Diploma Thesis

Experimental and numerical investigations to quantify the elastic deformation behavior of steel-reinforced hydraulic hoses

submitted for acquiring the academic degree
"Diplom-Ingenieur" (equivalent to Master of Science)
at the Vienna University of Technology, Institute of Process Engineering

Diplomarbeit

Experimentelle und numerische Untersuchungen zur Quantifizierung des elastischen Deformationsverhaltens von stahlverstärkten Hydraulikschläuchen

eingereicht zur Erlangung des akademischen Grades
„Diplom-Ingenieur"
an der Technischen Universität Wien, Institut für Verfahrenstechnik

von

Mario Trimmel, BSc

Matr.Nr.: 01228750

unter der Anleitung von

Associate Prof. Priv.-Doz. Dipl.-Ing. Dr.techn. **Stefan Scheiner**

Institut für Mechanik der Werkstoffe und Strukturen
Technische Universität Wien
Karlsplatz 13/202, 1040 Wien, Österreich

Wien, im Februar 2019

Acknowledgment

I would first like to thank my supervisor Dipl.-Ing. Dr. techn. Stefan Scheiner of the Vienna University of Technology (TU Wien). Dr. Scheiner's door was always open whenever I ran into a trouble spot or had a question about my research or writing. He consistently allowed this thesis to be my own work, but steered me in the right direction whenever he thought this was required.

Special thanks go to the company Semperit Technische Produkte GesmbH, which provided the topic of this thesis. I would like to especially mention Dipl.-Phys. Stefan Robin, Dipl.-Ing. Florian Arthofer and Dipl.-Ing. Alfred Wondracek, who were always available and supported me with the necessary input, be it sample material, ideas or their experience in the field of hydraulic hoses.

Furthermore, I would like to thank the staff of the laboratory for macroscopic material testing of the Institute for Mechanics of Materials and Structures (TU Wien), in particular Dr. phil. Roland Reihsner, and the staff at the physics laboratory of the Semperit Technische Produkte GesmbH — all of them always actively supported me in carrying out the experiments.

Finally, I must express my very profound gratitude to my parents and to my wife Karin for providing me with inexhaustible support and continuous encouragement throughout my years of studying and through the process of working on and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Kurzfassung

Hydraulikschläuche werden in vielen technischen Anwendungen als kraftführende Leitungen eingesetzt. Bei deren Konstruktion und Produktion muss eine ausreichende Belastbarkeit gegenüber diversen chemischen und mechanischen Einwirkungen sichergestellt werden. Hydraulikschläuche setzen sich üblicherweise aus mehreren Gummilagen und Verstärkungsschichten zusammen, wobei die Verstärkungsschicht die mechanische Tragfähigkeit bereitstellt und die Gummilagen die Verstärkungsschicht vor äußeren Einflüssen schützt.

Die vorliegende Masterarbeit beinhaltet experimentelle und numerische Untersuchungen an Hydraulikschläuchen, um deren mechanisches Verhalten zu quantifizieren. Für die experimentelle Untersuchung der Schläuche wurden neue Prüfprotokolle implementiert, welche verschiedenste Belastungsarten simulieren. Die Durchführung und die Bewertung der numerischen Untersuchungen wurde in Form einer Machbarkeitsstudie durchgeführt.

Die experimentellen Untersuchungen beinhalten im Wesentlichen uniaxiale Zugversuche ohne bzw. mit Druckbeaufschlagung sowie 3-Punkt-Biegeversuche. Um die auftretenden Verformungen zu erfassen, wurde zusätzlich eine Bildkorrelationsmethode entwickelt. Für die Durchführung der numerischen Untersuchungen, auf Basis einer Finite-Elemente-Methode, wurde auf der makroskopischen Ebene die sogenannte „Submodelltechnik“ angewendet. Zusätzlich sollten CT-Erfassungen als Basis dienen für die Erstellung von Modellen auf der mikroskopischen Ebene. Es ist zu erwähnen, dass sich diese Arbeit auf die Untersuchung von einfach und doppelt-geflechtverstärkten Hydraulikschläuchen beschränkt. Des Weiteren wird das Verformungsverhalten der Schläuche lediglich im elastischen Regime untersucht. Alterungseffekte, thermische oder chemische Einflüsse werden nicht behandelt.

Die experimentellen Untersuchungen lieferten für die Zug- und 3-Punkt Biegeversuche konsistente Ergebnisse und entsprachen im Wesentlichen den Erwartungen. Hingegen waren die Ergebnisse für Zugversuche mit zusätzlicher Druckbeaufschlagung äußerst inkonsistent, was auf eine zu geringe Druckbelastung zurückgeführt wurde — für zukünftige Studien wird eine leistungsfähigere Hydraulikeinheit empfohlen. Die numerischen Untersuchungen lieferten für das auf der makroskopischen Ebene eingesetzte Balkentheorie-Modell plausible Ergebnisse. Die Strategien zur Modellierung auf mikroskopischer Ebene waren zwar generell durchführbar, in dieser Form aber viel zu rechenaufwendig und liefern außerdem unzufriedenstellende Ergebnisse. Um die identifizierten Nachteile der derzeit verfügbaren Modellierungsstrategien auszumerzen, wird ein grundlegendes Überdenken der zur Anwendung kommenden Methoden empfohlen. Ein rigoroser Multiskalenansatz, der die Hochskalierung der mechanischen Eigenschaften von der Mikro- zur Makroebene ermöglicht, scheint hierfür ein vielversprechender Ansatz zu sein.

Abstract

Hydraulic hoses are essential components of many different industrial devices and machines. In order to assure sufficient load-carrying capacity, hydraulic hoses are mostly made of steel-reinforced rubber. One of the main difficulties in designing the exact composition and structure of steel-reinforced rubber hoses is finding the required balance between the actual loading of the hoses and the provided resistance against rupture or cracking. Classically, this was done by testing representative samples experimentally. Given the nowadays available, powerful computer systems it seems reasonable to complement experiments by computer simulations, in order to inter- and extrapolate experimental data points, in efficient manner.

This master thesis presents a two-fold approach to contributing to resolving the aforementioned issue, that is to quantify the mechanical behavior of hydraulic hoses by means of both experimental and computer simulation campaigns. Thereby, existing experimental modalities were extended by new protocols, while numerical studies were, due to the involved intricacies, carried out in terms of a feasibility study.

As for the experiments, tensile tests without and with additional pressurization, as well as three-point bending tests were carried out. For measuring the thereby occurring displacements, a new image correlation method was implemented. Computational studies, on the other hand, included several techniques, with the so-called submodeling technique turning out to be the most reasonable strategy on the macroscopic level. On the microscopic level, on the other hand, CT imaging served as basis for more detailed models which were supposed to provide additional, localized information. It should be stressed that this work was restricted to single- and double-reinforced hoses and to purely elastic deformations, while ageing effects, and thermal as well as chemical influences were not taken into consideration.

The results obtained for tensile and three-point bending tests were consistent and as expected, while the pressurization tests delivered substantially inconsistent results. It is believed that these inconsistencies were caused by the fact that the hydraulic unit used in the tests was not powerful enough, hence the applied pressure was too low. Thus, in future studies, it is recommended to use a more powerful hydraulic unit. The numerical studies yielded plausible results when using beam-type representations of the reinforcement steel on the macroscopic level. On the microscopic level, the studied methods appeared to be, in principle, feasible, but turned out to be both inaccurate and computationally extremely expensive. It is thus recommended to reconsider the modeling strategy from scratch. In this regard, a rigorous multiscale modeling strategy, allowing for upscaling the mechanical properties from the micro- to the macroscale, seems to be a promising approach.

Contents

Chapter 1

Introduction

1.1 Motivation

Hydraulic hoses are essential components of many different industrial devices and machines. They have to endure working pressures of more than 500 bar, are installed in lengths of up to 25 meters and must allow for substantial bending, with bending radii depending primarily on the hose diameter, ranging from 6 mm to 51 mm [1]. Nowadays, in order to increase their load-carrying capacity, or, in other words, to decrease their proneness to failure, hydraulic hoses are mostly made of steel-reinforced rubber.

One of the main difficulties in designing the exact composition and structure of steel-reinforced rubber hoses is finding the required balance between the actual loading of the hoses and the provided resistance against rupture or cracking. Thus, the design goal is to condition the hoses such that technologically relevant peak loads can be safely endured. To that end, the classical approach involves preparing representative samples of steel-reinforced rubber hoses, and testing these samples mechanically; e.g., by means of hydraulic impulse loading with flexing [2], finding ratios of proof and burst pressure to design working pressure [3], or burst pressure tests [4].

While experimental tests are absolutely essential in order to gain insights concerning the general characteristics of any material, they provide, from a quantitative point of view, information only in terms of snapshots. This means that, e.g., some mechanical constants (such as the stiffness) are revealed experimentally for a material featuring a specific composition, and tested under specific loading conditions. Extrapolation of such data to similar yet different materials is, however, often problematic. Hence, relying on experimental testing alone entails either substantial uncertainties, or tedious experimental campaigns which are, however, in a purely experimental approach necessary for understanding the material comprehensively. Owing to the substantial improvements of computer systems (in terms of computing power) over the last decades, harnessing computer simulations for studying the mechanical behavior of materials and structures has been identified as promising complement to experimental testing.

For many kinds of materials, material systems, and structures, computer simulations have indeed turned out as useful tool when it comes to simulating the response of a specific material (or structures made thereof) to specifically defined loading scenarios; see, e.g., works on cementitious material [5, 6], bone [7–10], or wood [11, 12], just to mention a few (out of many) examples. However, the material system that is dealt with in this work, namely that of steel-reinforced rubber hoses, is a particularly tricky one. This concerns both the materials involved (soft/compliant rubber is combined with hard/stiff steel) and the length scales of the structural elements (the diameters and wall thicknesses of the hoses are on the order of magnitude of a few centimeters, while the diameter of the steel strands is in the sub-millimeter range).

Considering a well-defined kind of steel-reinforced rubber hoses, this thesis aims at shedding light on two specific aspects: (i) studying the mechanical behavior of such hoses experimentally, based on meaningful application of already existing experimental methods, by implementing novel protocols; and (ii) performing feasibility studies concerning the numerical simulation of those tests. Before presenting the particular objectives of this work, the results of a corresponding literature survey are presented (in Section 1.2), focusing on the state of the art in both experimental and modeling approaches which already were, or at least could be applied to steel-reinforced rubber hoses.

1.2 Literature survey

The subsequently presented results of literature reviews on both experimental and computational techniques applied in the context of steel-reinforced rubber hoses show exemplarily the state of the art in these two fields. However, it should be stressed that this literature survey is most likely not exhaustive. The interested reader is hence referred to pertinent review articles and textbooks, see e.g. [13–15].

1.2.1 Experimental techniques for studying the mechanical behavior of steel-reinforced rubber hoses

Hydraulic-pressure impulse test without flexing:

The standard DIN EN ISO 6803 [2] defines pressure pulse tests for hydraulic hose lines which are carried out at low and high pulse pressures, without consideration of the bending [2]. The test specimens, which are equipped with suitable fittings, are connected to the test device. Test specimens with nominal internal diameters of less than 22 mm are implemented in a 180° bent form in the test setup, see Fig. 1.1(a). Test specimens with larger nominal internal diameters are installed with a 90° bend. Fig. 1.1(b) shows the test equipment used. The protocol of this (destructive) test is defined as follows: First, the used liquid (such as hydraulic fluids) is heated to the desired temperature, after which the pulsating pressure is applied [2]. The applied frequency of the pressure pulse depends on the pressure range of interest – high pressure-testing involves frequencies ranging from 0.5 to 1.3 Hz, whereas low pressure-testing involves frequencies ranging from 0.2 to 1 Hz [2]. Throughout the test implementation, the pressure pulse frequencies are recorded. The test is carried out until the hose line fails (e.g. in the form of leakage), or until the predefined maximum number of cycles has been reached.

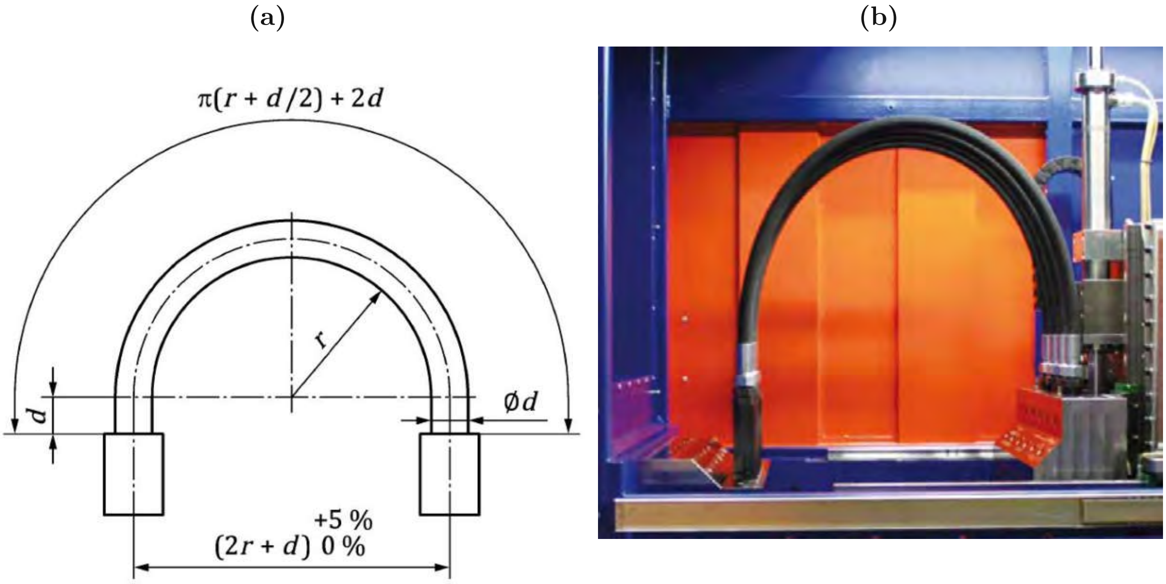


Fig. 1.1: Test setup according to DIN EN ISO 6803 [2]: (a) installation position of the test specimen (for hoses with nominal internal diameters of less than 22 mm) [2]; (b) test device for hydraulic hoses [16]

Hydrostatic testing:

The standard DIN EN ISO 1402 [4] describes tests involving the subjection of hoses to hydrostatic pressure. During the test (which is again destructive), the specimens are mounted in fixed position, see Fig. 1.2. Several aspects can be investigated, listed below:

- *Leak testing under constant test pressure:* For that purpose, the specimen is pressurized to the specified leak testing pressure (typically ranging from 0.7 to 1120 bar) and maintained at the chosen pressure level between 30 seconds and 60 seconds [4]. In this time window, the specimen is examined for defects, such as cracks, or leaks.
- *Measurement of deformation under pressure:* This test deals with determining the deformation of a hose under pressurization. When subjected to pressures ranging from 0.7 to 800 bar, torsional deformations, warping, as well as changes of the length and of the outer diameter are measured [4], based on references marks and rotating tape measure, respectively, according to standard DIN EN ISO 4671 [17].
- *Combined leak/burst resistance testing*: First, the hose is subjected to 70% of the expected burst pressure, and held at this pressure for 5 min. Subsequently, the pressure is increased, at a rate ranging from 320 to 800 bar/min, until the hose bursts. In addition to the pressure at bursting, the location and type of failure is recorded [4].

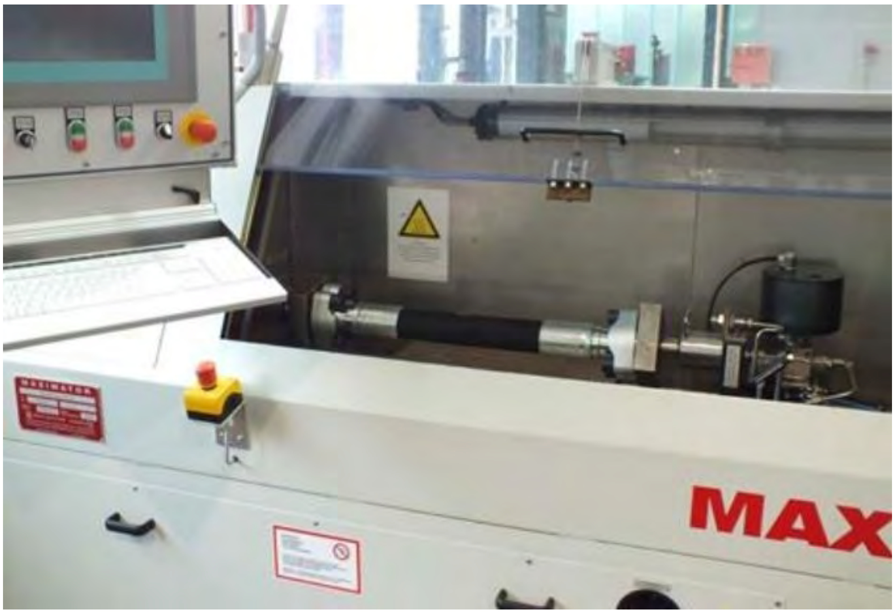


Fig. 1.2: Photograph of the device used for hydraulic testing of hoses [16]

Ratios of proof and burst pressure to maximum working pressure:

The standard DIN EN ISO 7751 [3] describes the protocol for determining the ratios of test and minimum burst pressure to the maximum operating pressure for different operating modes of a hose [3]. The procedure for carrying out the pressurization tests are specified in the previously described standard DIN EN ISO 1402 [3].

Further tests:

In addition to the previously described standard tests, a range of further standard testing modalities are available (which are, however, not described in this thesis). These include tests concerning the diameter and concentricity, see EN 24671 [18]; the temperature flexibility, see EN 24672 [19]; the adhesion between the layers, see EN 28033 [20]; the vacuum resistance, see EN ISO 7233 [21]; the oil resistance, see ISO 1817 [22]; the water resistance / resistance to aqueous liquids, see ISO 1817 [22]; and the ozone resistance, see EN 27326 [23].

1.2.2 Prediction of the mechanical behavior of steel-reinforced rubber hoses by means of computer simulations

Blurred layer method:

As mentioned before, modeling steel-reinforced rubber hoses in minute detail is neither reasonable nor feasible. Szabó and co-workers [24] addressed this problem as follows: The cross section of the studied hose was divided into three layers, see Fig. 1.3. The outer and the inner layers of the hose were considered to be isotropic and homogeneous. The middle layer, comprising both steel reinforcement and rubber matrix was considered to be a kind of blurred layer, which was assumed to be homogeneous but anisotropic.

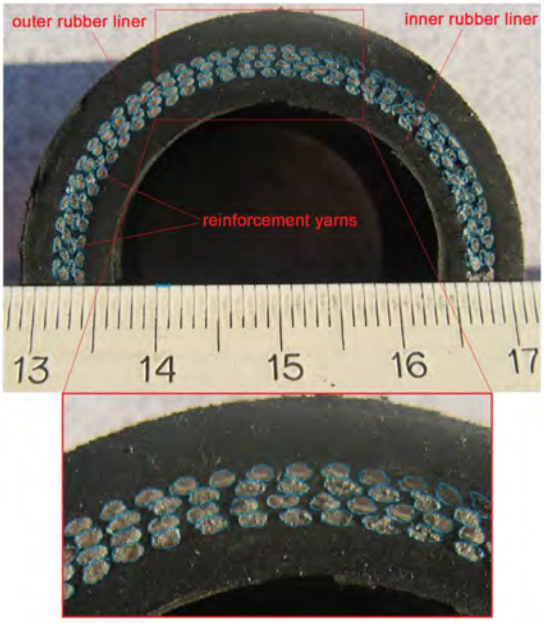


Fig. 1.3: Cross section of the material modeled by Szabó et al. [24], showing two rubber layers with one steel-rubber matrix in-between; reproduced from [24]

Image analysis software was used to capture the characteristics of selected cross sections of the material, such as the diameters defining the rubber layers, or the distribution and the slope of the steel yarns, allowing to estimate the volume fractions of the constitutents. Furthermore, tensile tests were considered for assigning material properties to the constituents. Notably, a Mooney-Rivlin-type constitutive law was considered for the rubber layers, assuming thereby incompressibility of rubber. The material properties of the composite material between the pure rubber layers were found based on simplistic mixture rules, see Fig. 1.4 for a corresponding flow chart followed in the course of such analysis.

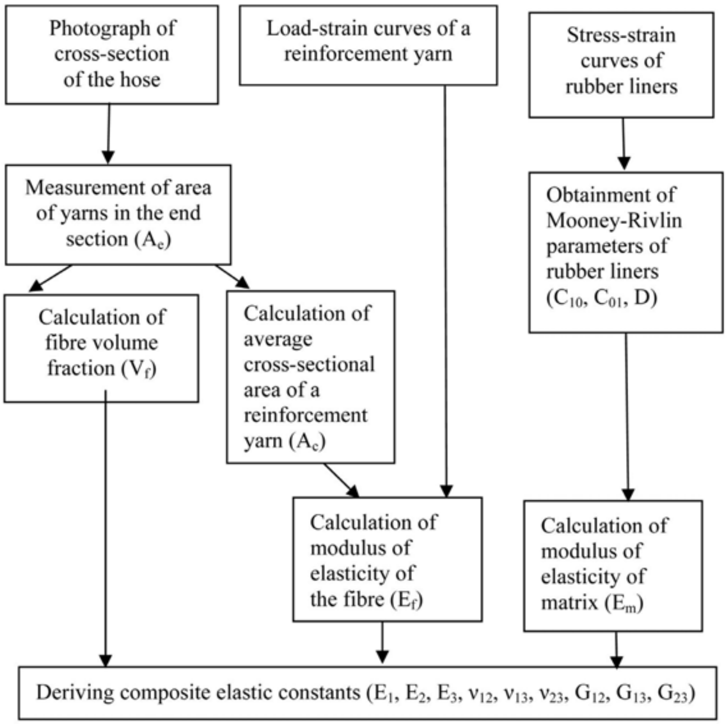

Photograph of cross-section of the hose
Load-strain curves of a reinforcement yarn
Stress-strain curves of rubber liners
Measurement of area of yarns in the end section (A_e)
Obtainment of Mooney-Rivlin parameters of rubber liners (C_{10}, C_{01}, D)
Calculation of fibre volume fraction (V_f)
Calculation of average cross-sectional area of a reinforcement yarn (A_c)
Calculation of modulus of elasticity of the fibre (E_f)
Calculation of modulus of elasticity of matrix (E_m)
Deriving composite elastic constants (E_1, E_2, E_3, ν_{12}, ν_{13}, ν_{23}, G_{12}, G_{13}, G_{23})

Fig. 1.4: Flow diagram for determination of the material properties of the reinforcement layers, which serve as input parameters for computer simulations in the framework of the blurred layer method [24]

The main advantage of this methods is clearly its simplicity, which comes, however, at a price. Firstly, the complex, braided structure of the steel reinforcement used to construct the material studied in this thesis cannot be adequately considered, as key features (such as the so-called form locking behavior due to the interactions between the individual interwoven wire bundels) are then not taken properly into account. A further, even more fundamental problem is the simplistic averaging of the elastic behavior of soft and hard constituents. Most likely, more sophisticated strategies are necessary for reliably predicting the overall mechanical behavior of steel-rubber composite materials.

Submodeling method:

A two-fold approach is pursued in the framework of the so-called submodeling method, see e.g. [6, 25]. On the macroscopic scale, the exact geometry is considered, using, e.g., the data acquired from computed tomography scanning. The exact material behavior of the macroscopic material follows from numerical analysis of a microscopic model, where specific microstructural features can be resolved, allowing hence for considering hierarchically organized materials, see Fig. 1.5. On the other hand, the submodeling technique also comprises the possibility to compute, say, stress distributions in areas which require very fine meshes separately, and to consider the derived stresses as boundary conditions for the coarser mesh used to represent the remaining part of the structure [26], see Fig. 1.6.

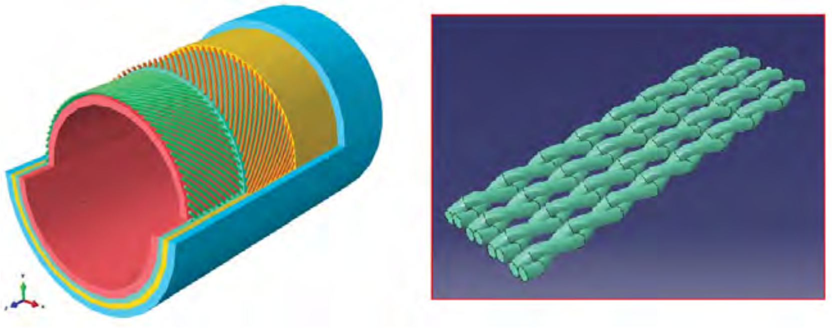


Fig. 1.5: Macroscopic scale model with reinforcement elements (left), and microscopic model of the reinforcement layers [25]

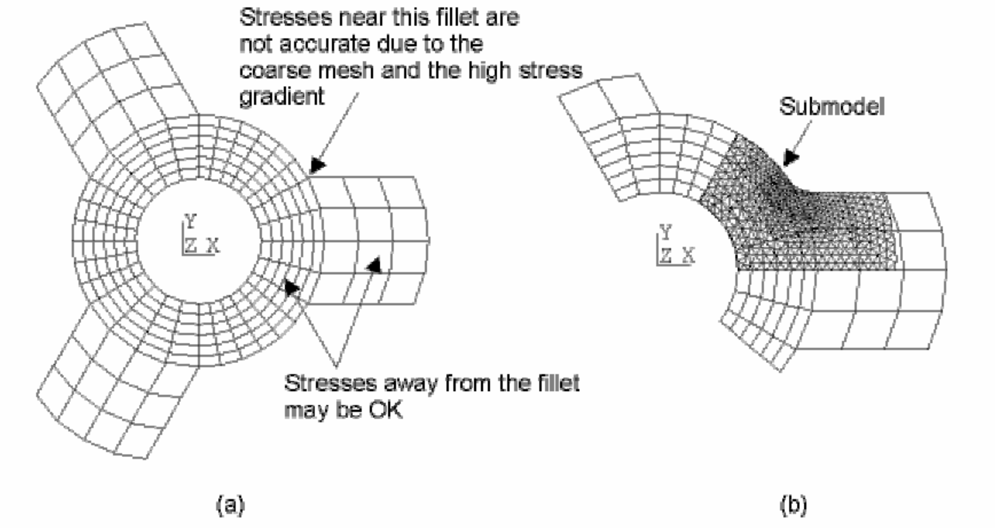

Stresses near this fillet are not accurate due to the coarse mesh and the high stress gradient
Y
Z X
Stresses away from the fillet may be OK
(a)
Submodel
Y
Z X
(b)

Fig. 1.6: Finer resolution of detailed areas within a coarser mesh [27]

While the submodeling technique offers a number of obvious advantages, some drawbacks should be emphasized as well. The main problem concerns the need of establishing accurate relations between micro- and macroscopic models. Furthermore, strictly speaking,the coupling of fine and coarse meshes requires an iterative approach, which may even compensate the strived-after improved numerical efficiency of the submodeling technique.

Domain superposition technique (DST):

The domain superposition technique was developed especially for composite materials requiring non-linear material models [28]. Models developed in the framework of the DST involve a global domain geometry which is superimposed with domains relating to reinforcement phases, see Fig. 1.7. Essentially, two strategies are pursued in the framework of the DST, the 2-, and the 3-region DST. Subsequently, both approaches are briefly compared to conventional Finite Element Analysis (FEA):

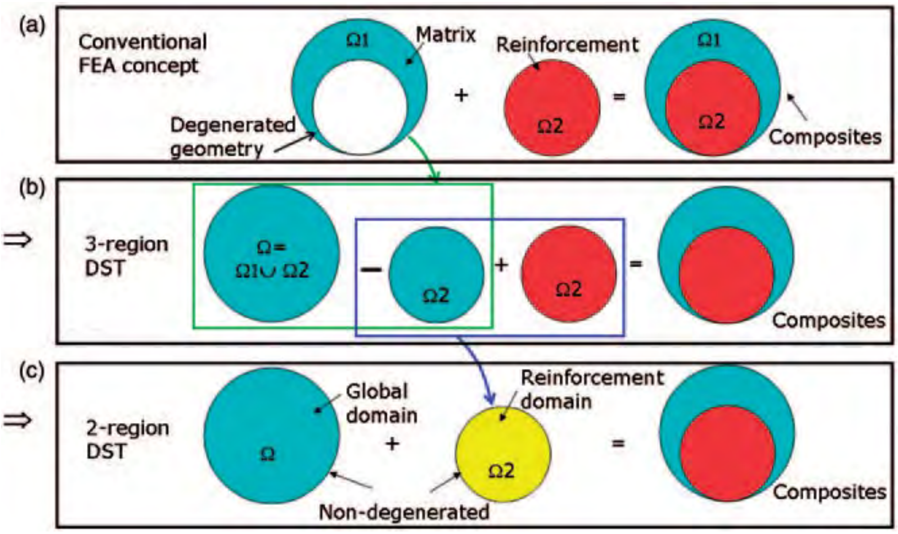


Fig. 1.7: Schematic overview of the domain superposition technique (DST), taken from [28]; implying of the reinforcement domain into the global domain by means of: (a) the conventional FEA concept, (b) the 3-region DST concept, (c) the 2-region DST concept

- *Conventional FEA concept:* The matrix domain and the reinforcement domain are modeled and connected to each other, implying the disadvantages as described before [28], see Fig. 1.7(a).
- *3-region DST:* Three sets of geometric domains are used. One domain related to the material of the matrix. The reinforcement geometric domains are considered twice. One of the reinforcement domains contains matrix material, however with properties of negative magnitude, and the other represents the reinforcement material. Combination of all three domains yields the desired material representation, see Fig. 1.7(b) [28].
- *2-region DST:* In this method, see Fig. 1.7(c), the meshes of the aforementioned two reinforcement domains are merged, yielding a completely new material on the reinforcement domain [28].

One key advantage of this method is that the individual components of the composite body can be individually meshed and then inserted at the desired locations of the matrix. This way, tedious partitioning of complex internal geometries as is usually needed in conventional FEA [28] can be avoided. The main disadvantage of the DST method, as compared to conventional FEA, is that the components are only embedded in the matrix layer, meaning that contact conditions which may exist between the reinforcement elements and the global domain cannot be taken into account.

1.3 Objectives of this thesis

From the review presented in Section 1.2 one can easily conclude that a reliable analysis of the material under investigation, that is steel-reinforced rubber hoses, is not possible by simply utilizing existing methods, neither experimentally nor numerically. For the sake of improving this unsatisfactory situation, this thesis aims at better understanding the mechanical behavior of steel-reinforced rubber hoses, see section 2 for a detailed description of the studied material. For that purpose, a two-fold strategy was pursued:

1. On the one hand, uniaxial tensile tests, without and with concurrent pressurization of the hoses, as well as three-point bending tests were performed, see Section 3.
2. On the other hand, the feasibility was fathomed whether computer simulations are a meaningful method in order to study the mechanical behavior of steel-reinforced rubber hoses, see Section 4.

The results and insights gained in the presented project are comprehensively discussed in terms of their significance in Sections 3.3 and 4.3, and respective conclusions are presented, together with a corresponding outlook to potential future activities, in Section 5.

Chapter 2

Description of the studied material

In this thesis, steel-reinforced rubber hoses are studied, see Fig. 2.1. During operation, the steel braiding is responsible for providing mechanical resistance against excessive radial widening, while the rubber is supposed to prevent the exposure of the steel braiding to chemical influences or damages.

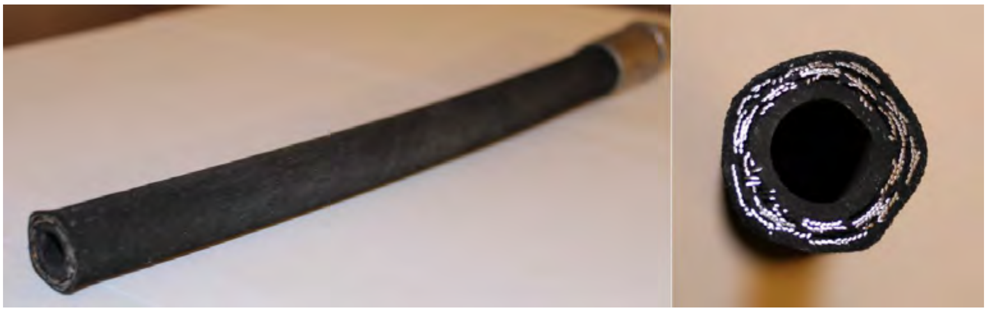


Fig. 2.1: Investigated material: steel-reinforced rubber hoses; photographs showing the investigated specimens (left) and one exemplary of the cross sections of the latter (right)

2.1 Structure and production of hydraulic hoses

Conventional, steel-reinforced hydraulic hoses typically consist of several layers, see standard EN 853 [29] for corresponding recommendations. The layers comprise an inner tube, the reinforcement domain, and an outer cover layer [30]. If there is more than one reinforcement layer, an intermediate layer is inserted between the reinforcement layers, in order to ensure an appropriate spacing. Fig. 2.2 shows, schematically, the structure of a double-reinforced rubber hose. In this thesis, both single- and double-reinforced hoses were tested.

Hydraulic hoses are often manufactured by means of the mandrel method [31]. This method involves spraying the inner tube onto a mandrel by means of extrusion. Then, the reinforcement is put in place by means of a braiding machine with rotating coil carriers. The optional intermediate layers are attached helically in the form of calendered tapes. Finally, the cover layer is again applied helically, beforehand prepared by calendering. In order to make sure that the shape of hose is not perturbed in any way by the subsequent vulcanization process, diagonal bands of tissue are wrapped around it, which are eventually removed. These tapes are also responsible for the typical fabric pattern on the outside of the cover layer.

In the present case, the inner layer is 2 mm thick, and consists of nitrile butadiene rubber (NBR), cross-linked by means of sulfur, while soot is used as active filler. The intermediate layer is also made of NBR, and exhibits a thickness of 0.2 mm. The outer layer, in turn, is 1 mm thick, and consists of styrene butadiene rubber (SBR), whereas cross-linking was also carried out by means of sulfur, and again soot was used as filler. As for the steel braiding, the wires consist of brass-coated steel, with diameters ranging from 0.25 mm to 0.71 mm [30].

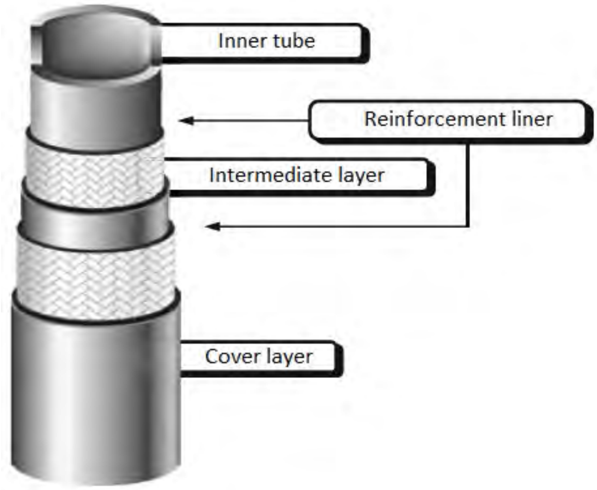


Fig. 2.2: Basic structure of a double-reinforced hydraulic hose [30]

As regards the deformation behavior of the hose when being pressurized, the braiding angle (that is the angle by which the wires are rotated with respect to each other), according to which the wires are applied, is key. Another important feature of such hoses is that their length and diameter should remain constant under loading [30]. Such isochoric behavior is only granted if the ratio of the radial force F_R and the axial force F_A is equal to the tangent of the braiding angle α [30]. This relation allows to deduce the so-called neutral angle α_N, see [30] for details. If $\alpha > \alpha_N$, pressurization causes an volume increase, and if $\alpha < \alpha_N$, pressurization causes volume decrease.

2.2 Fundamental aspects of rubber materials

2.2.1 Styrene-butadiene rubber (SBR)

SBR is the most prevalent synthetic rubber. It is a co-polymer, derived from a styrene monomer and a butadiene monomer [30]. SBR belongs to the R-group, which exhibits an unsaturated main chain (double bonds). Styrene and butadiene polymers can be derived by two processes, the emulsion polymerization (E-SBR) and the solution polymerization (S-SBR). They also feature a good resistance against heat and aging. On the other hand, SBR vulcanizates have many beneficial chemical properties in terms of resistance against water, glycol, alcohol, as well as, dilute and non-oxidizing alkalis and acids [30]. SBR vulcanizates are not resistant against oil, fats, as well as aromatic and aliphatic hydrocarbons. In terms of their mechanical properties, SBR vulcanizates have a lower strength, as compared to the vulcanizates made of natural rubber. SBR also has a lower tear resistance, as compared to the natural rubber vulcanizates. Advantageous for their use is the good compromise between abrasion and wet grip and additionally good resistance to aging, which is key for its application as tire material [30]. Furthermore, SBRs are applied in technical rubber products (such as conveyer belts or hoses), in the field of paper coating (in the form of SBR latex), and in the construction industry – in particular as carpet backing or foams.

2.2.2 Nitrile-butadiene rubber (NBR)

NBR is also a synthetic rubber, with the co-polymer consisting of acrylonitrile and butadiene [32]. For the production of NBR, butadiene and acrylonitrile are emulsified in water and then polymerized through free radical initiators [32]. The acrylonitrile content of the copolymer determines the glass transition temperature (increasing with increasing acrylonitrile content), and thus the viscoelastic properties (e.g., damping, elasticity), as well as the resistance to fuels, oil and hydrocarbons, of the vulcanized rubber [30], NBRs exhibit an improved heat resistance, as compared to SBRs and natural rubber [30]. Due to their low oxygen content, hoses made of SBRs are well suited for the use of hot fluids, as this slows down the aging processes, making them more stable for these liquid media. Given the low permeability of NBRs, they are well-suited for applications where any form of liquids are involved [30]. As concerns the mechanical properties of NBRs, they are characterized by a lower tensile strength than natural rubber (which further decreases with increasing temperature) [30].

2.2.3 Filler

In the presently studied material, soot is used as filler material. Soot is a complex solid that consists mainly of carbon, and is used in the rubber industry as an active filler. Active soots are defined as chain-shaped aggregates, consisting of spherical primary particles [30]. The production of soot is based on thermal decomposition of carbons and incomplete combustion of organic compounds [30]. Soot is added to mixtures in order to improve the resistance of the eventually emerging material against abrasion, its UV resistance, and its tear strength. Remarkably, the use of soot as filler material causes black coloring of such vulcanizates.

2.2.4 Mullins effect

One very distinctive mechanical feature of rubber materials is the so-called Mullins effect, also referred to as stress softening [33, 34], describing the kind of hysteresis-type shapes of loading/unloading-curves of such materials, representing progressive softening as the load is increased in the course of several loading cycles. As can be seen in Fig. 2.3, the Mullins effect implies that some kind of softening occurs during the first loading cycle, which may be caused by breaking mechanically unstable cross-links, desorption of adsorbed chain sections from the filler surface and adsorption in the low stress state, collapse of locally available agglomerates, displacement or orientation of filler particles in the direction of stretching, diffusion of adsorbed polymer molecules along the soot surface, or sliding entanglements along the chain ends or between cross-linking sites [33].

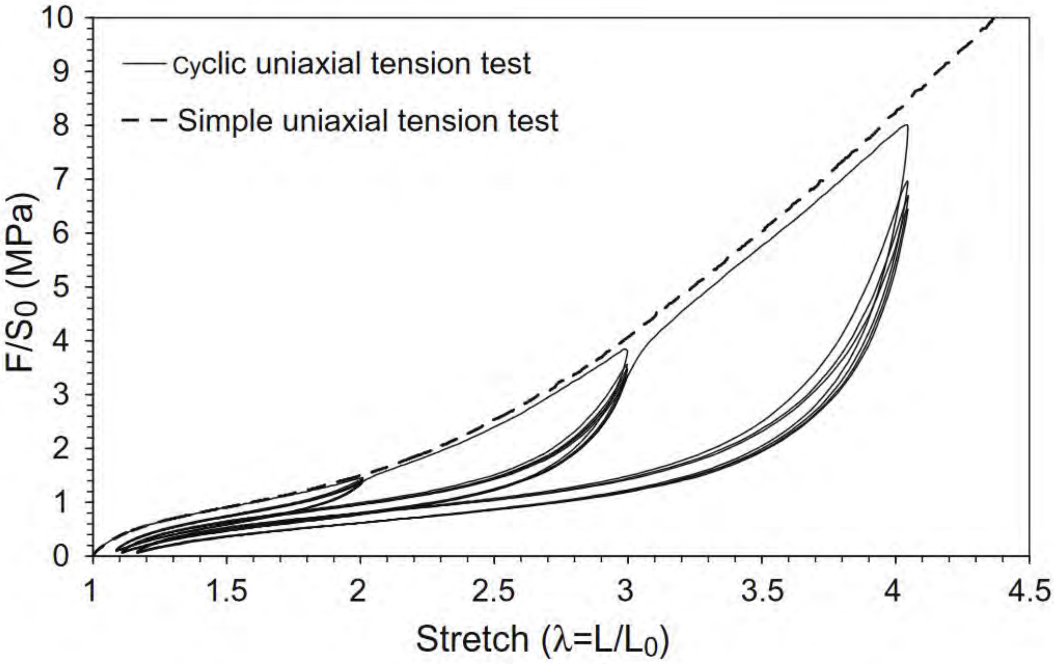


Fig. 2.3: Schematic illustration of the Mullins effect [34]

Chapter 3

Experimental campaign

3.1 Description of the experimental methods

The experimental campaign which was performed in the here-presented project aims at collecting information going beyond standardly applied testing modalities. The latter involve tests on the manufactured hoses, such as hydraulic impulse tests without flexing [2], determination of ratios of proof and burst pressure to maximum working pressure [3], and burst pressure tests [4], as well as tests on the materials of which the hoses are composed, including determination of the adhesion between the components [20], determination of the material hardness [35], determination of the yield strength in a tensile test and determination of tensile stress-strain properties [36]. In this project, tests were performed both on the single components of the hoses (see Section 3.2.1), as well as the on the hoses on a structural scale (see Sections 3.2.2 to 3.2.4).

3.1.1 Single component tests

For material characterization, material tests were performed, in particular on the rubber layers of which the studied steel-reinforced rubber hoses are composed. For that purpose, tensile and tensile/compression tests, respectively, were conducted using a universal testing machine, characterized by a maximum force of 5 kN, on both cylindrical and flat dumbbell-shaped specimens, see Fig. 3.1 for the specimen shapes and Tab. 3.1 for the specifications of the tests. Both types of tests were evaluated in terms of comparing the applied forces to the correspondingly occurring displacements, see Section 4.1 for details.

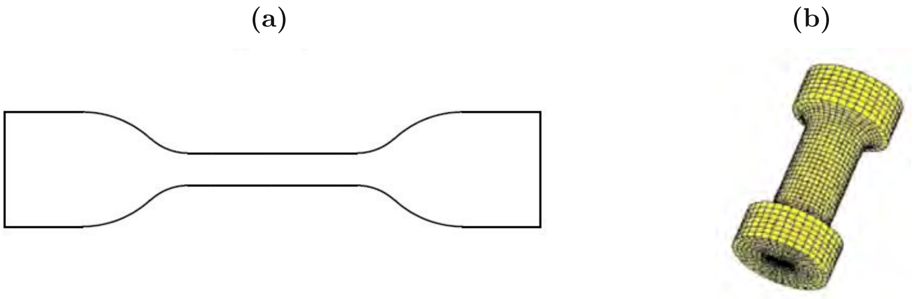


Fig. 3.1: Shapes of specimens subjected to single component tests; (a) flat dumbbell-shaped specimen [36], and (b) cylindrical dumbbell-shaped specimen [37]

Tab. 3.1: Specifications of the material tests

Specimen shape	Flat dumbbell	Cylindrical dumbbell
Test specification	Five cycles each from 0-100 % (l_0=20 mm); tensile load is increased until failure	Five cycles each from -10-60 % (l_0=20 mm); tensile/compressive load is increased until failure
Test velocity	20 and 200 mm/min	20 and 200 mm/min
Criterion for termination	Specimen failure (crack)	Specimen failure (crack)
Load cell	500 N, accuracy class 1	500 N, accuracy class 1
Displacement measurement	Image-correlation (two cameras)	Image-correlation (two cameras)
Fixation of specimens	Symmetrical clamping jaws	Form fit over the shoulder region of the dog bone-shaped test body with the clamping device
Sample preparation	Punching out of a plate with cutting tool according to ISO 23529 [38]	Manufactured by means of injection molding

3.1.2 Hose tests on universal testing machine

The main contribution of this thesis in terms of collecting new experimental insights was based on tensile tests conducted on hose sections, both excluding and including pressurization of the hoses.

3.1.2.1 Hose specimen

The specimens were sections of steel-reinforced rubber hoses of approximately 400 mm length (for the exact definition of the composition and constituents, see Section 2.2). For appropriate mounting of the specimens, female straight steel-crimped hose fittings (3/8 inch) were attached, according to the standard ISO/TS 17165-2 [39], to both ends of the hoses, see Fig. 3.2. For the sake of comparison, both single- and double-reinforced hoses were tested.

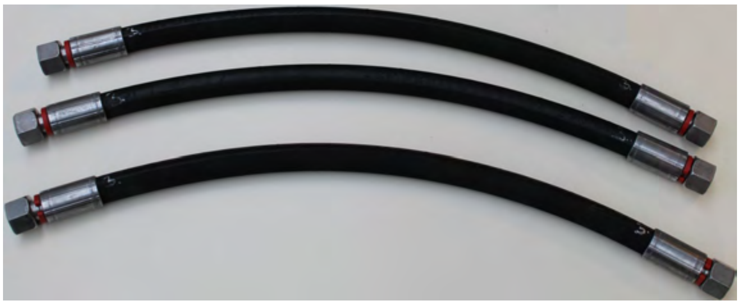


Fig. 3.2: Photograph of exemplary specimens

3.1.2.2 Testing devices

The central testing device was a standard, universal testing machine, see Fig. 3.3(a), including a hydraulic unit, namely a Zwick/Roell Z050. In more detail, the spindle-driven loading device was able to apply a maximum force of 50 kN, while the hydraulic unit (Wille Geotechnik DV10/150-1), see Fig. 3.3(b), was capable of generating a pressure of up to 150 bar.

By means of the fittings defined in Section 3.1.2.1, the hose specimens were mounted into the machine on the machine base, on the one hand, and to the vertically moving crossbeam, see Fig. 3.4(a), on the other hand. For that purpose, steel blocks were custom-built, which could be force-fitted into the machine, and which included screw connections precisely compatible to the hose fittings, see Fig. 3.4(b).

Several sensors were used for continuously measuring the test parameters. On the one hand, two Gefran pressure sensors (able to measure up to 250 bar) were used, one close to the hydraulic unit, where the pressurization was generated, and one was attached to the steel block at the crossbeam of the testing machine. Furthermore, the forces exerted by the testing machine were recorded directly by the load cell (Zwick/Roell Xforce HP for up to 50 kN). In order to link the force cell force values to a desktop computer used for data processing, the computer was coupled to the universal testing machine controller via an analog-to-digital converter (Type NI 9215).

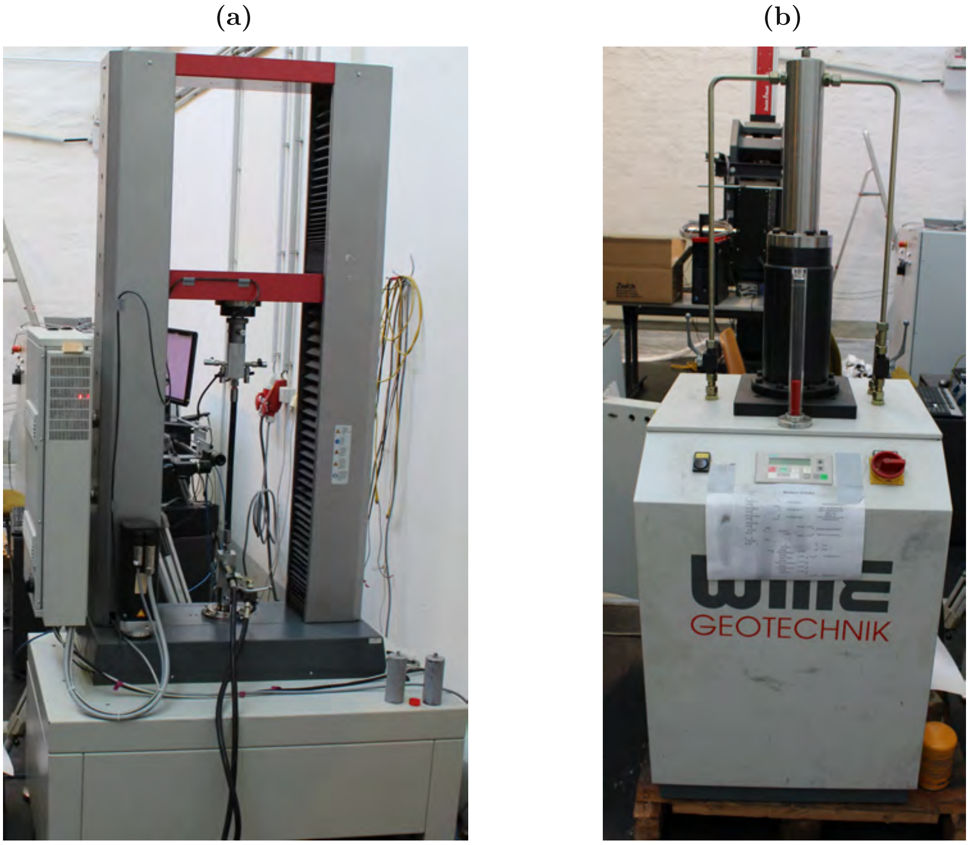


Fig. 3.3: Measurement setup: (a) universal testing machine; (b) hydraulic unit

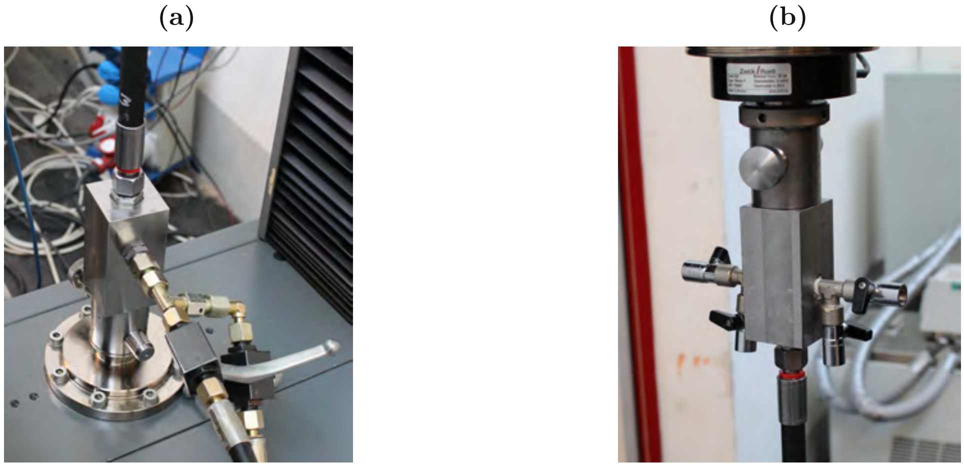


Fig. 3.4: Measurement setup: (a) connection on the machine base; (b) connection on the moving crossbeam

In order to monitor the displacements of the hose during tensile loading and pressurization, not only the movement of the crossbeam was recorded, but also an optical displacement measurement system, based on the principle of image correlation, was used. In particular, two cameras were used, both mounted on tripods in order to place the cameras as close to the hose as possible, see Fig. 3.5(a). One camera was positioned in front of the testing machine, for capturing displacements in longitudinal and transversal directions (with respect to the main axis of the hose). The second camera was rotated by 90° (with respect to the main axis of the mounted hose), for the purpose of checking the displacements recorded by the first camera, and for recording the transversal displacements in an additional direction, see Fig. 3.5(b).

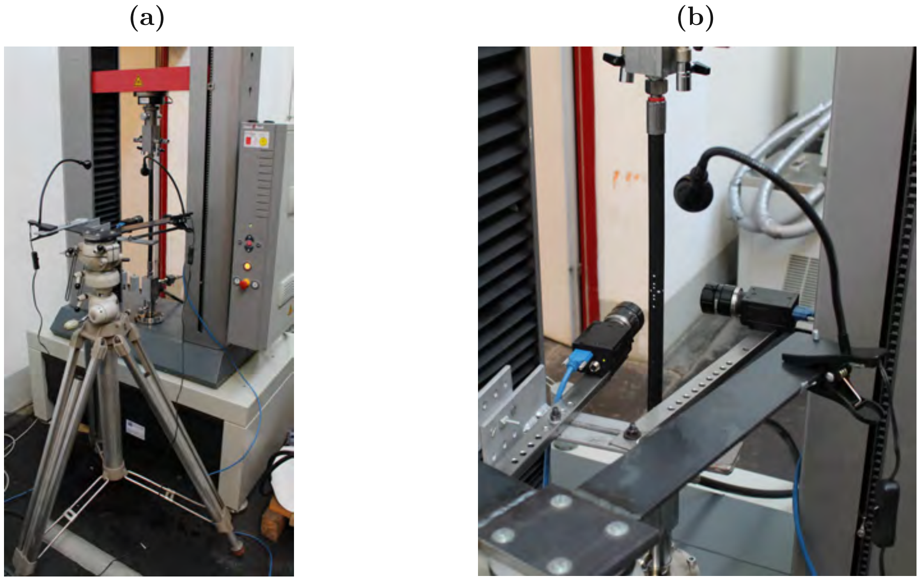


Fig. 3.5: Photographs showing (a) the arrangement of the cameras, and (b) a close-up of the tripod on which the cameras were mounted

3.1.2.3 Processing of recorded data

The image data was recorded by means of the commercial software LabVIEW (Version 2013) [40] which provides a platform for processing experimentally acquired data. Based on a previous work [41], a specifically designed workflow was established in the course of the presented project.

For recording the displacements based on image data, the tested hoses were provided with distinctive marks, see Fig. 3.6(a). In order to determine the displacements in a specific direction, the relative changes of the distances between the marks in the course of the hose deformation were measured. The respective analysis was implemented by means of the mathematical software Matlab (Version R2017b) [42]. The analysis included the following tasks:

1. Selection of the area to be analyzed.
2. Converting the image into gray values; Specification of the desired gray value areas and binarization of the image.
3. Detection of the marks: Enveloping the marks by rectangles and finding the (white-based, area-weighted) midpoint of the area within the rectangle, see Fig. 3.6(b).
4. Identification of the displacements of the marks across the image stacks.
5. Conversion and smoothing of the recorded force and pressure data.
6. Synchronizing displacement, force, and pressure data, and inclusion of all data in a solution matrix.

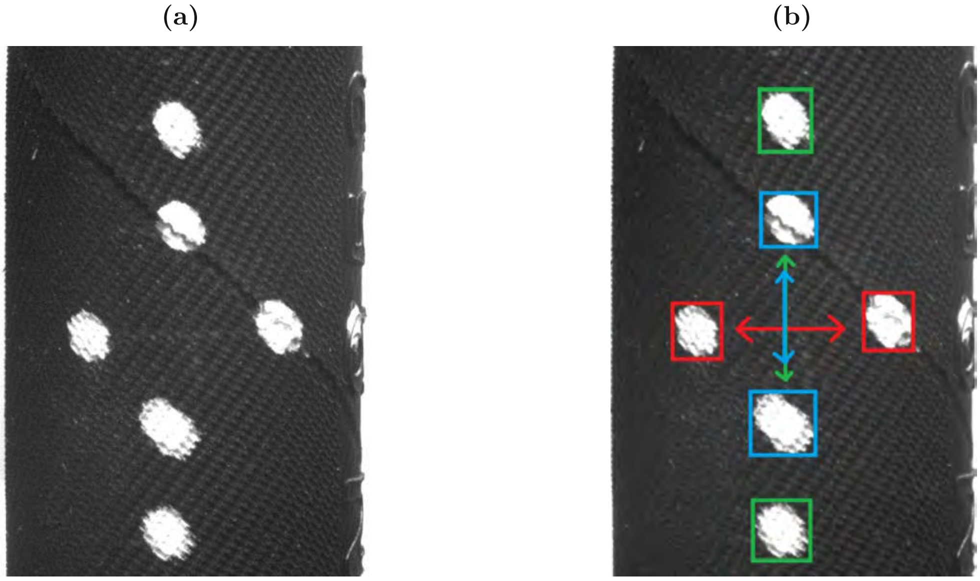


Fig. 3.6: Displacements were determined based on reference markings, see (a), and measurement of the distance changes while the hose is deforming, see (b)

The Matlab-script related to the above-sketched sequences of analysis steps is included in this thesis, see Appendix A.1.

3.1.2.4 Experimental protocols

Tensile tests were performed according to the following protocol:

1. Application of a initial load.
2. Position-controlled stretching of the specimen up to eventual elongations of 5 mm, 10 mm, and 15 mm, respectively, using stretching rates of 20 mm/min and 200 mm/min, respectively.
3. Complete unloading of the specimen.
4. Each test comprised 5 loading/unloading cycles.

Pressurization tests with fixed crossbeam were performed according to the following protocol:

1. Application of a initial load; the such obtained position was held constant subsequently.
2. Pressurization to 150 bar, using a pressurization rate of 150 bar/min.
3. Holding of pressure for two minute.
4. Complete de-pressurization.
5. Each test comprised two pressurization/depressurization cycles.

Pressurization tests with concurrently applied constant load were performed according to the following protocol:

1. Application of a initial load; this load was held constant subsequently (implying a correspondingly moving crossbeam).
2. to 5. As above.

3.1.3 Three-point bending tests

Additionally, the hose specimens were subjected to classical three-point bending tests, by means of a universal testing machine (Zwick/Roell Type Z005), using a 5 kN load cell (Zwick/Roell Type Xforce HP), and a standard three-point bending mounting device. The tests were performed in position-controlled fashion, up to a displacement of 20 mm at rates of 20 mm/min and 200 mm/min, respectively. During all three-point bending tests performed in this study, each specimen was loaded only once.

3.2 Experimental results

3.2.1 Single component tests

The stress values in the following plots are illustrated as normalized stress values back-calculated from the maximum obained stress value for all tensile tests. However, these values cannot be included in this thesis, due to confidentiality.

Tensile tests on flat dumbbell-shaped specimens:

The flat dumbbell-shaped specimens were produced by punching them out from the respective hose layers. Thereby, the specimens exhibited varying orientations; namely, in longitudinal direction, in transversal direction, and in diagonal direction (with respect to the main hose axis). However, no significant influence of the specimen orientation on the stiffness could be observed, for neither of the three hose layers, see Figs. 3.7(a) - (c). Layer 2 turned out to be more compliant than layer 1 and layer 3, see Fig. 3.7(d).

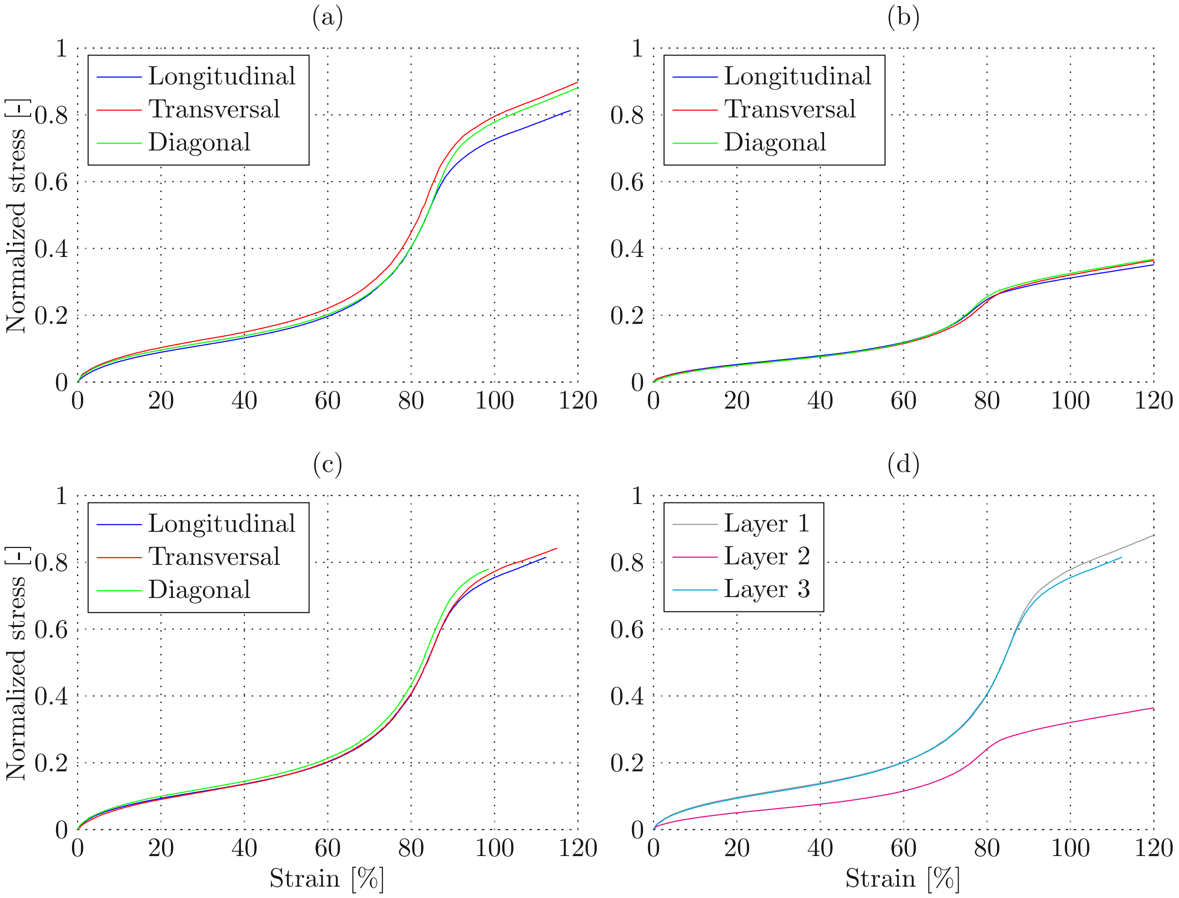


Fig. 3.7: Results of tensile tests on flat dumbbell-shaped specimens made out of the three layers of the hydraulic hoses: average values over five tests each, in (a) layer 1; (b) layer 2, (c) layer 3; (d) comparison of all layers in terms of average values over all five tests per layer, all punching directions, and loading rates

Tensile tests on cylindrical dumbbell-shaped specimens:

The stiffness of the layer 2 has turned out to be much lower than the stiffnesses of layer 1 and layer 3, while the latter two exhibit a more or less similar stiffness, see Fig. 3.8. No influence of the loading rate on the resulting load-displacement curves was observed.

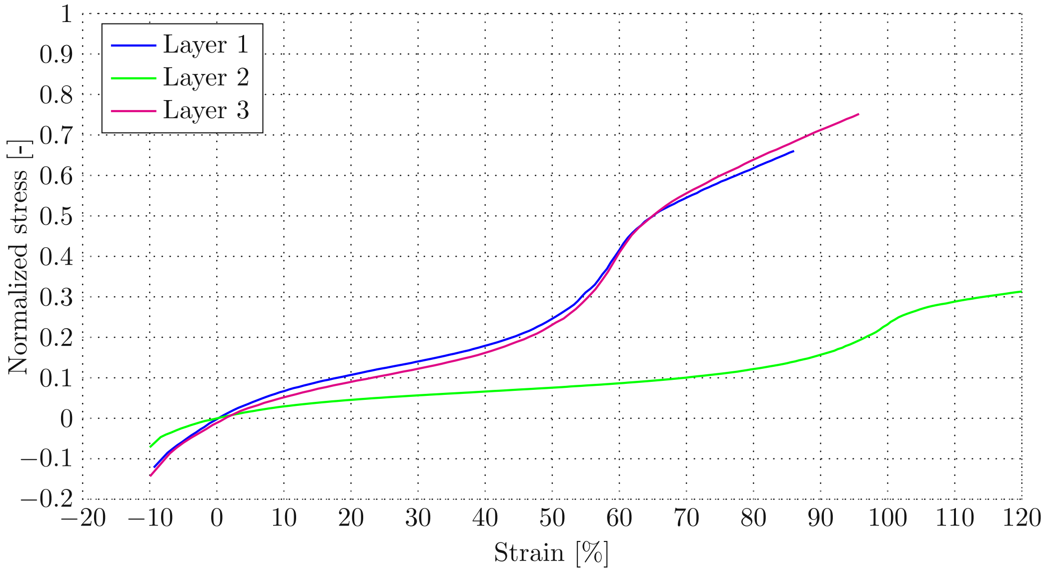


Fig. 3.8: Results of tensile / compression tests on dog cylindrical dumbbell-shaped specimens made out of the three layers of the hydraulic hoses; average values of five tests per layer, carried out at loading rates of 20 and 200 mm/min

3.2.2 Tensile tests on hydraulic hoses

In the subsequently presented tests, the displacements were measured both with the image correlation method described in Section 3.1.2.3, as well as based on the displacement measuring system of the crossbeam. Normalized force values, back-calculated from the maximum obtained pullforce for all tensile tests, were used for the following evaluation. However, these values cannot be included in this thesis, due to confidentiality.

Double reinforcement:

As expected, subjecting the specimen to a tensile load led to reduction of the cross section dimensions, see the load-displacement curves depicted, e.g. for sample 1, in Fig. 3.9 and 3.10. Thereby, the displacements were normalized with respect to the total length of the specimen (in case of the displacement measurement at the crossbeam), and with respect to the distance between the measurement marks in the middle section of the hose (in case of displacement measurement by means of image correlation), respectively. It seems like the difference between the two displacement measurement methods is negligible. From the strain characteristics in longitudinal direction, the Mullins effect is recognizable, given that the hose at the unloading does not return to the original length – of course, this could be also due to viscoelastic effects. Obviously, the image correlation-based measurements in transversal direction are considerably "noisy", which owes to the fact that the marks were quite close to each other, making the measurements more prone to irregularities. Fig. 3.10 shows comparisons of force versus displacements between the two measurement methods. Clearly, the resulting stiffnesses (proportional to the slopes of the curves) are again quite similar. It should be noted however that analysis of the results in terms of the overall stiffness of the hoses was not within the scope of this thesis, see also the discussion of results in Section 3.3. The differences between the tested samples were mostly insignificant. The difference between the measurement methods was slightly more significant in sample 3, which was caused by the more pronounced curvature of this sample in the unloaded state. All further results relating to samples 2 and 3 are illustrated in Figs. 3.11-3.14.

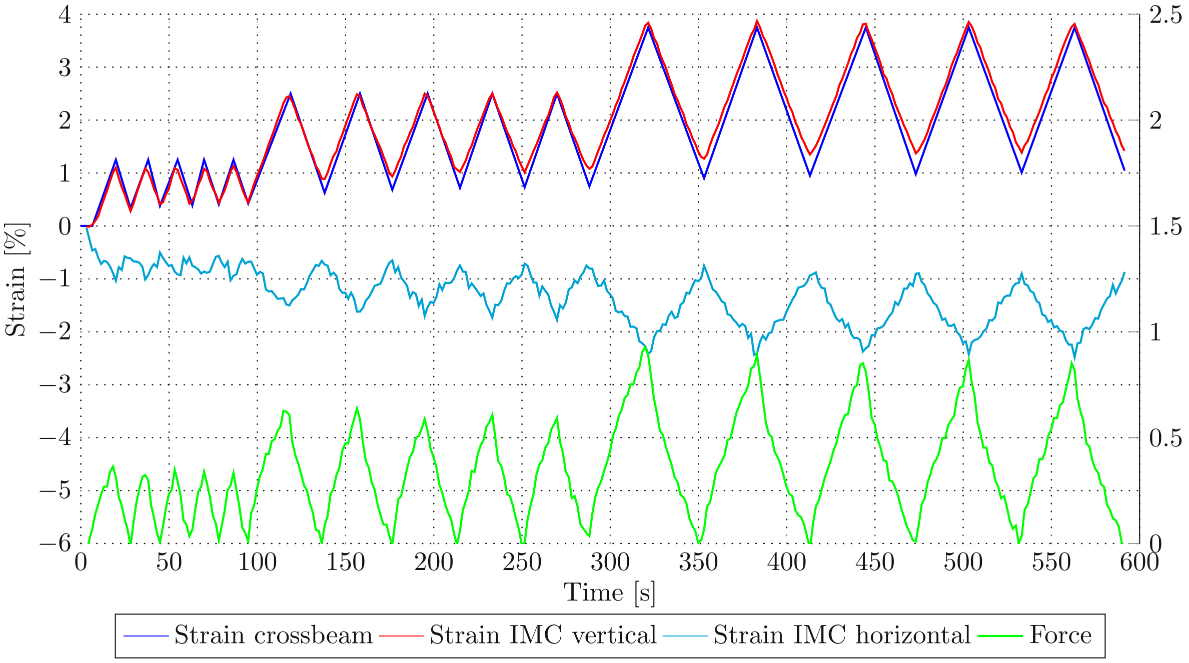


Fig. 3.9: Comparison of the force and displacements measured for sample 1 / double-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

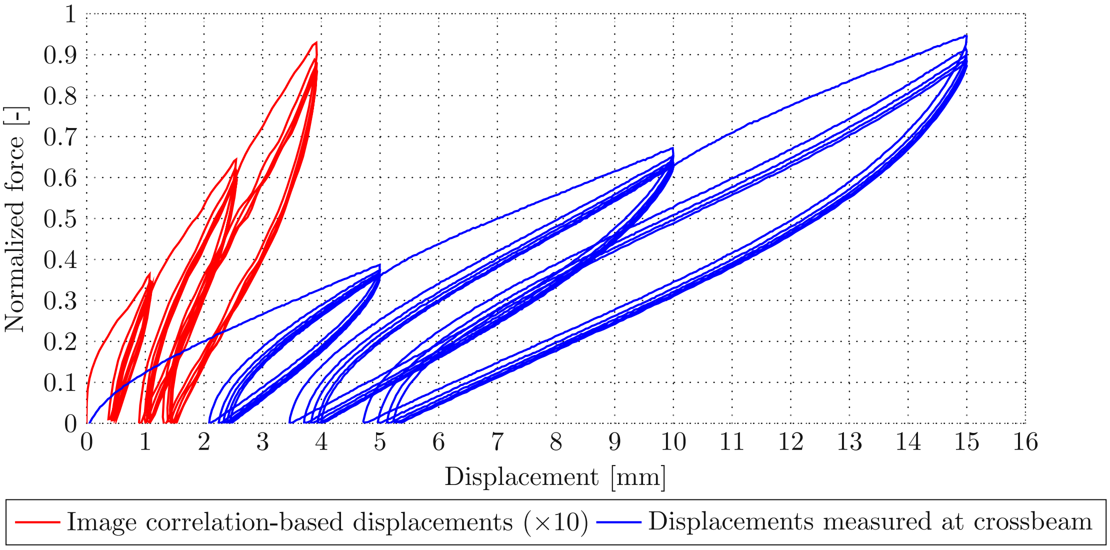


Fig. 3.10: Force versus displacements for sample 1 / double-reinforced

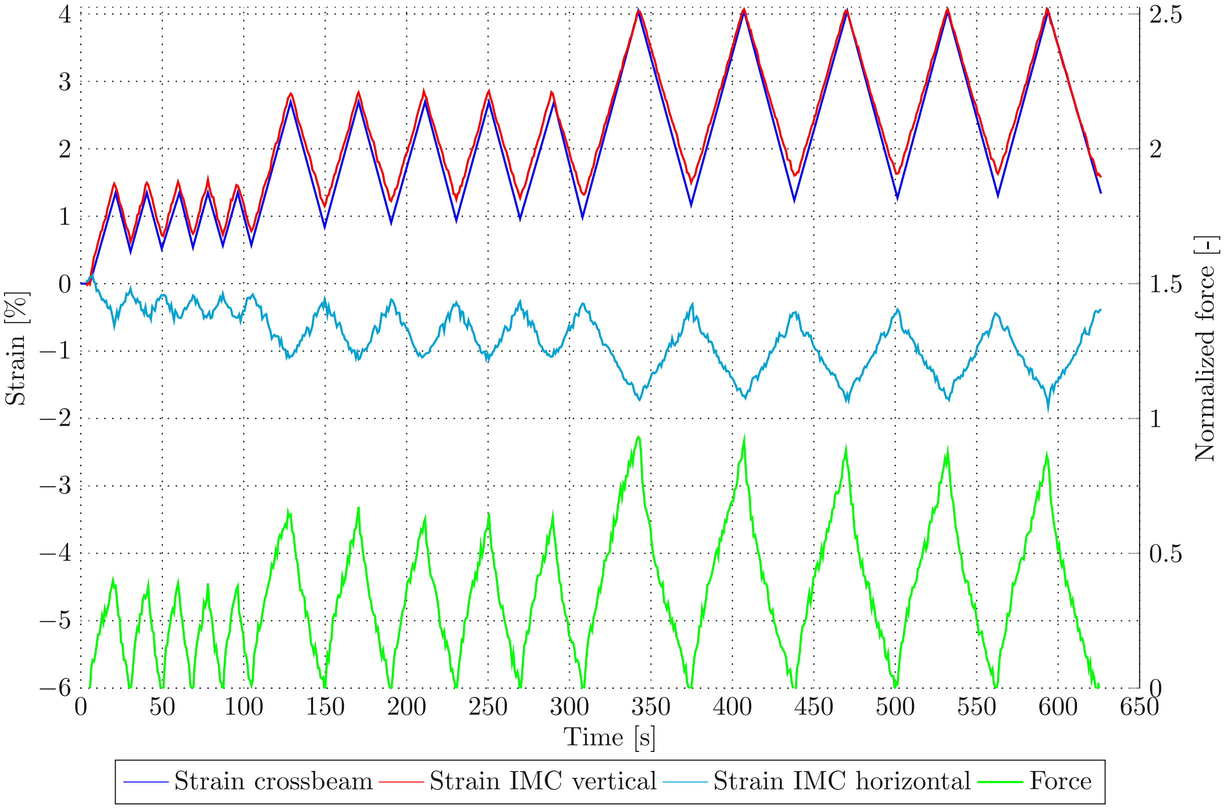


Fig. 3.11: Comparison of the force and displacements measured for sample 2 / double-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

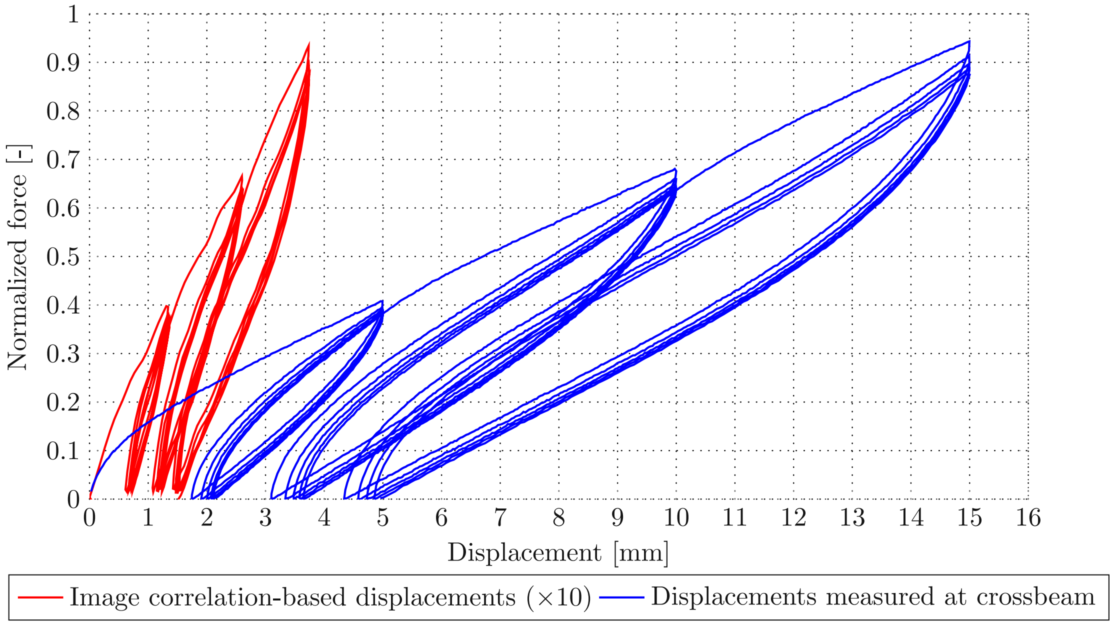


Fig. 3.12: Force versus displacements for sample 2 / double-reinforced

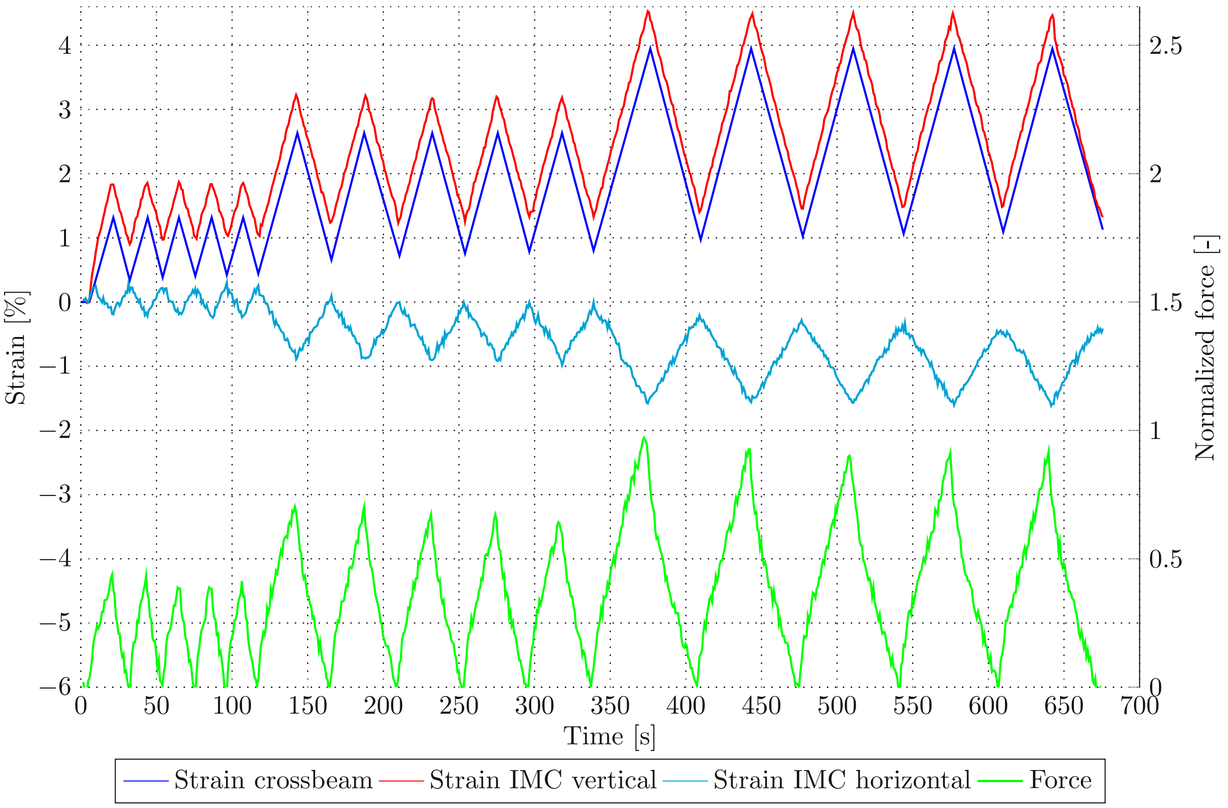


Fig. 3.13: Comparison of the force and displacements measured for sample 3 / double-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

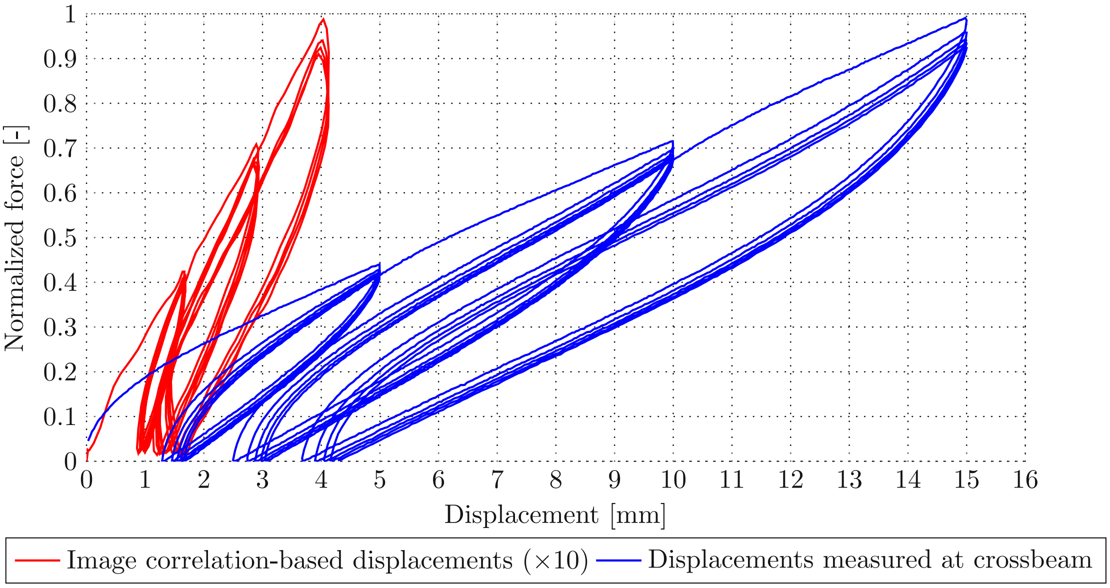


Fig. 3.14: Force versus displacements for sample 3 / double-reinforced

Single reinforcement:
As expected, single-reinforced samples turned out to be less stiff than the double-reinforced ones, while the influence of the curvature of the unloaded sample was more prominent. All related results are illustrated in Figs. 3.15-3.20.

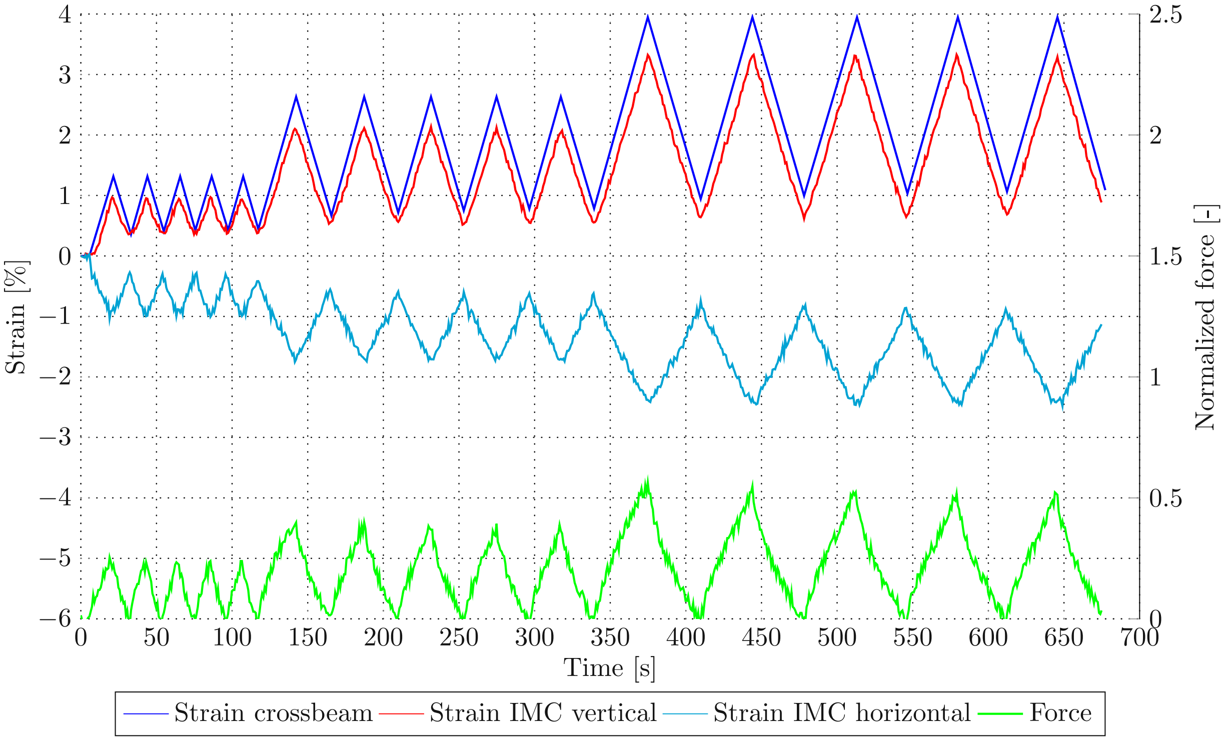


Fig. 3.15: Comparison of the force and displacements measured for sample 1 / single-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

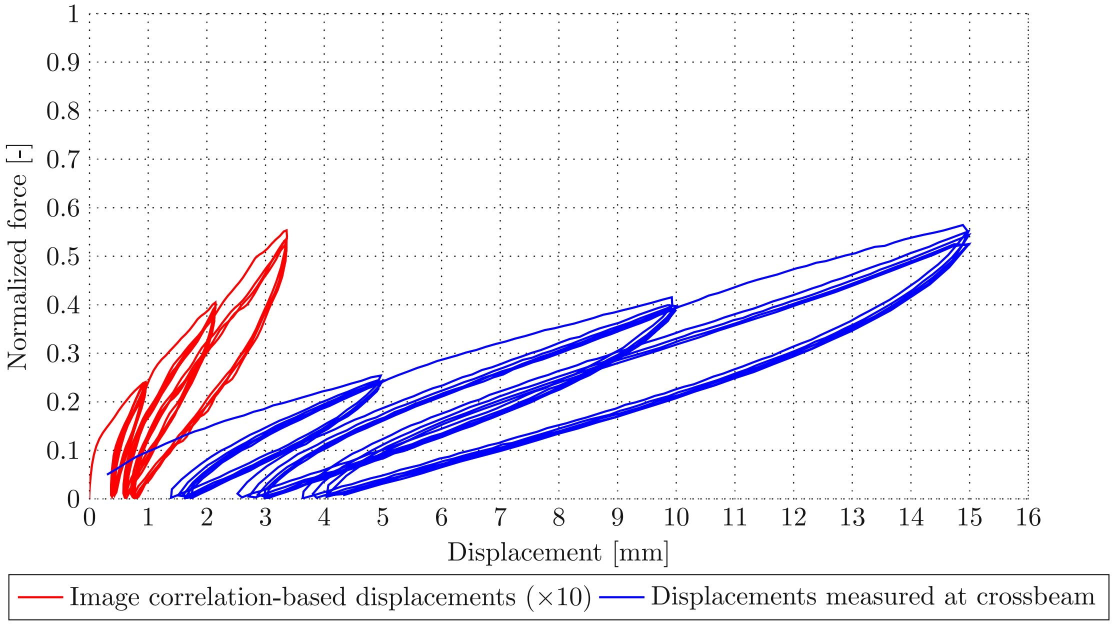


Fig. 3.16: Force versus displacements for sample 1 / single-reinforced

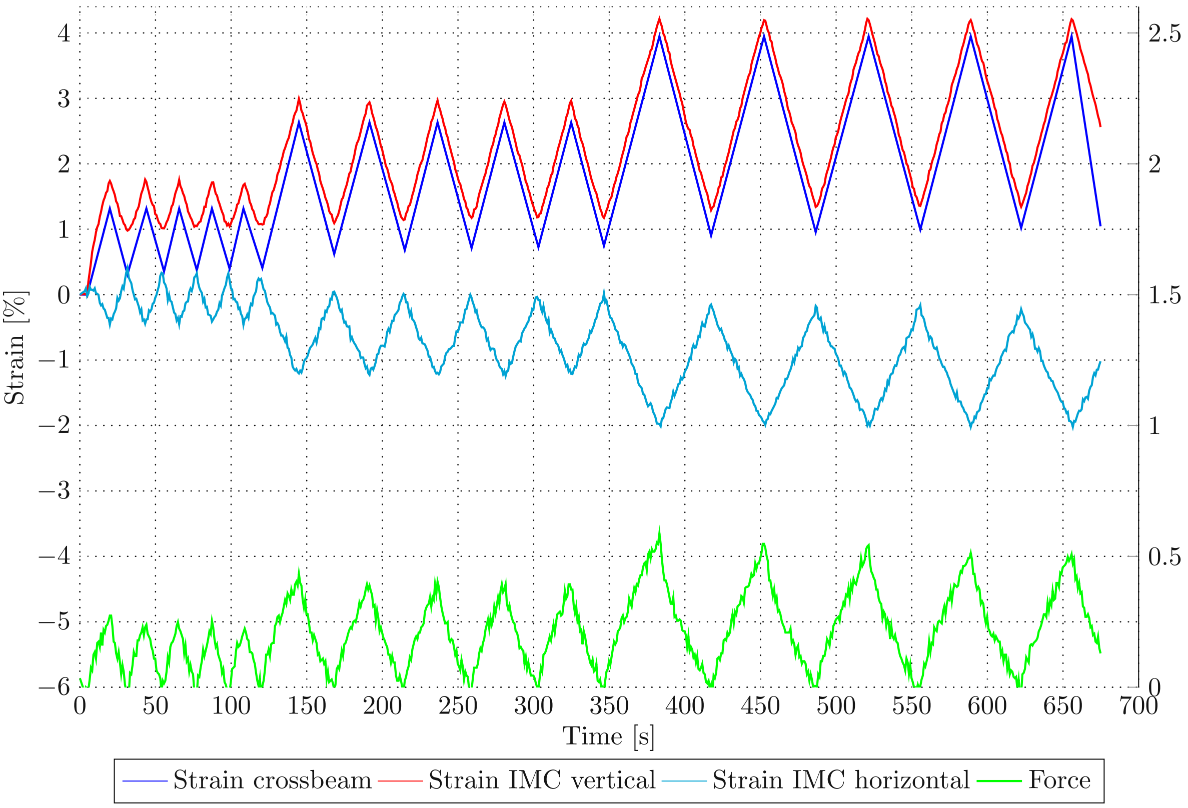


Fig. 3.17: Comparison of the force and displacements measured for sample 2 / single-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

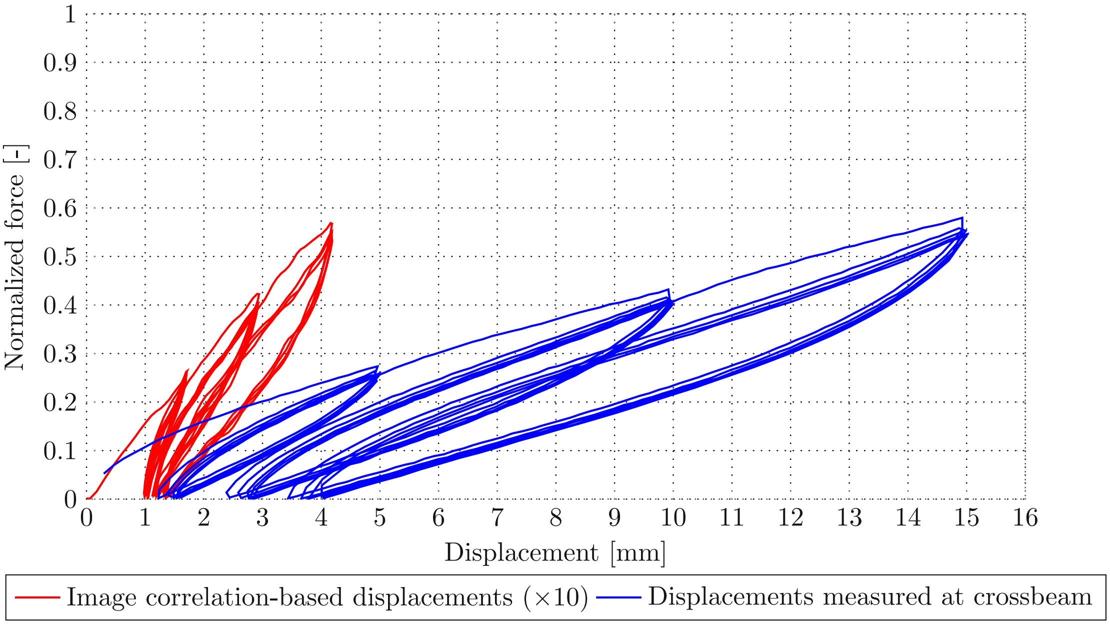


Fig. 3.18: Force versus displacements for sample 2 / single-reinforced

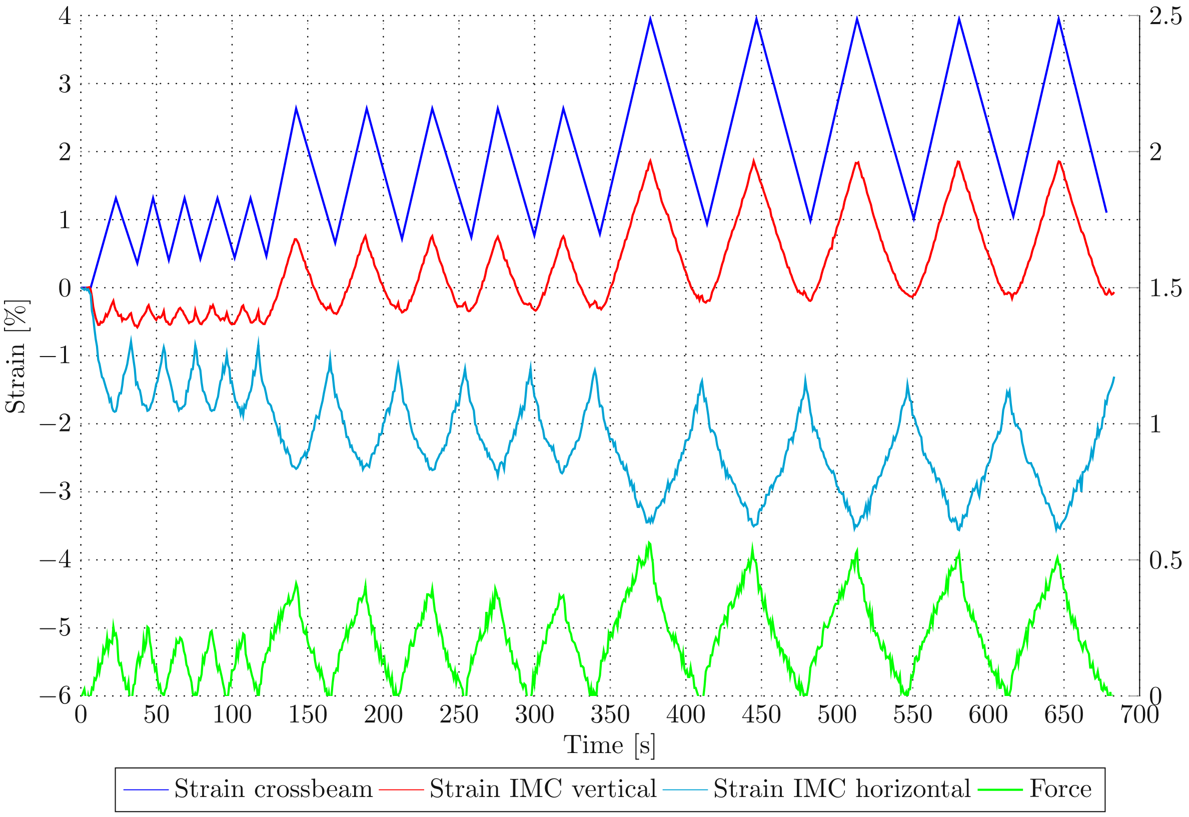


Fig. 3.19: Comparison of the force and displacements measured for sample 3 / single-reinforced; strains were computed based on the displacement measurement at the crossbeam, and also based on the displacement measurement through image correlation (IMC)

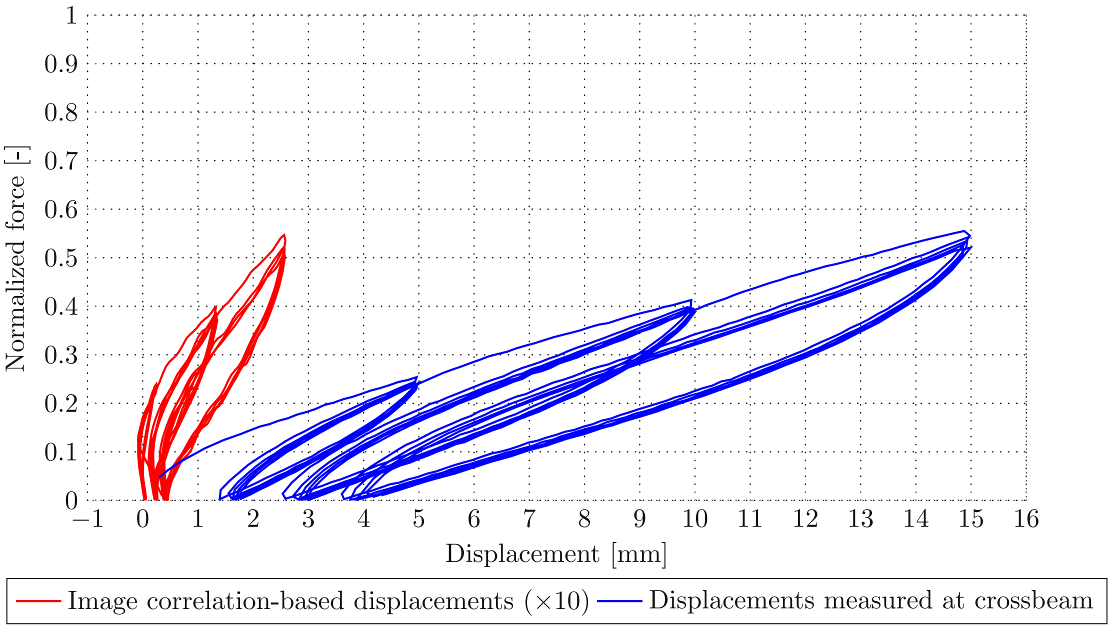


Fig. 3.20: Force versus displacements for sample 1 / single-reinforced

Comparison single- versus double-reinforced samples

Fig. 3.21 shows a direct comparison of the test results recorded for single-reinforced samples to the results recorded for double-reinforced samples. As stressed before, and as expected, double-reinforced specimens turned out to be much stiffer than the single-reinforced specimens. Hence, the forces required to reach the pre-defined maximum displacements was much lower for single-reinforced specimens.

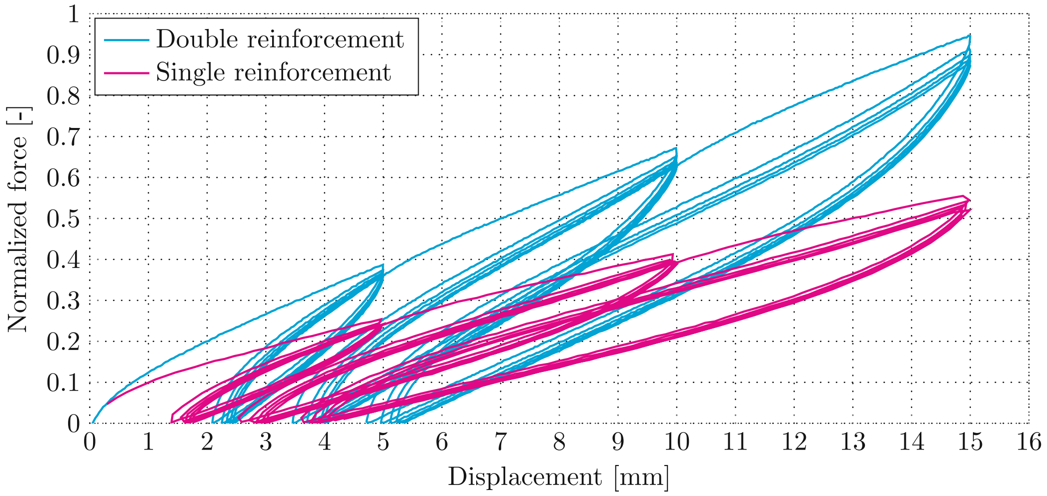


Fig. 3.21: Force-displacement curves for both single- and double-reinforced specimens; the depicted curves were acquired by means of the displacement measurement of the crossbeam

Influence of loading rates:

Fig. 3.22 confirms the previously made statement that loading rates did not affect the obtained results significantly. While this figure only contains the results acquired for single-reinforced specimens, also no dependence of the force-displacement curves on the underlying loading rate could be observed for double-reinforced samples.

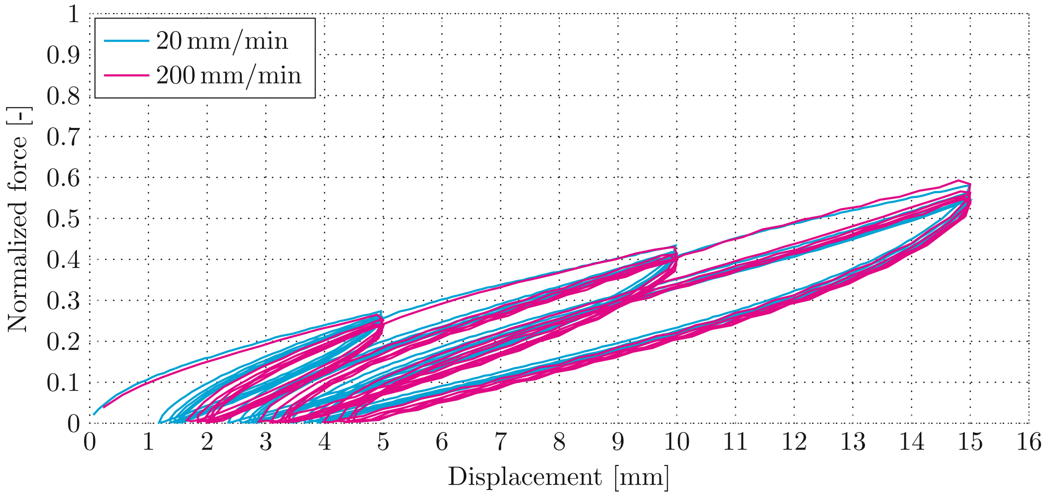


Fig. 3.22: Comparison of loading rates on the force-displacement curves for single-reinforced samples; the depicted curves were acquired by means of of the displacement measurement of the crossbeam

3.2.3 Pressurization tests

The measured and predetermined force values are represented as normalized force values and can be back-calculated to a fixed force value. However, these values cannot be included in this thesis, due to confidentiality.

3.2.3.1 Constant initial load

First, force-displacement curves were compiled, considering a constant initial load, and a pressurization rate of 150 bar/min, for both single- and double-reinforced samples. In Fig. 3.23, the initial path of the force-displacement curves confirms again the higher stiffness of the double-reinforced samples. Since the tensile testing machine was operated in force-controlled mode, the force stayed more or less constant at the initial load level, while pressurization led to shortening of the sample and unloading to elongation.

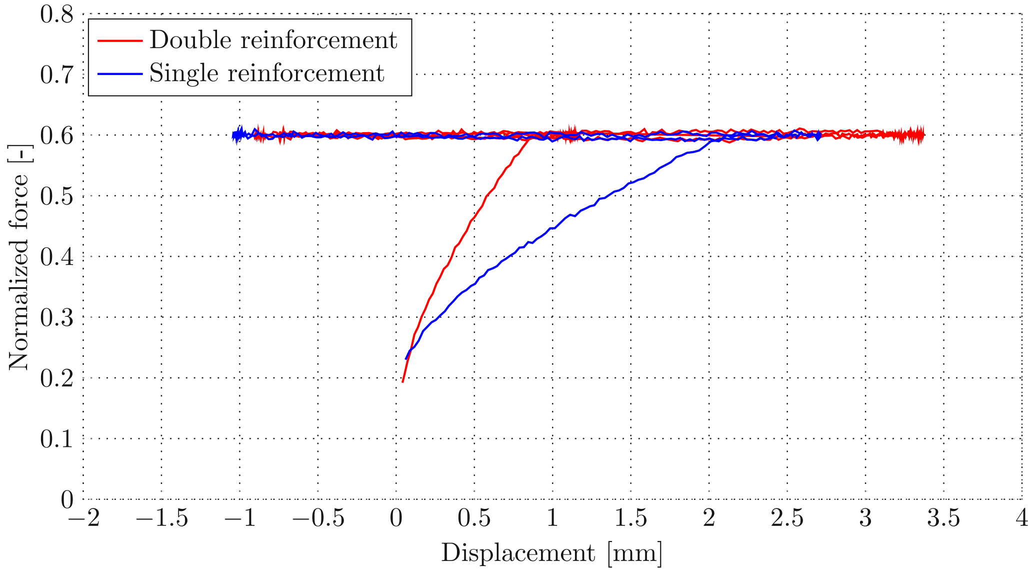


Fig. 3.23: Force-displacement curves for single- and double-reinforced samples during pressurization tests

Double reinforcement

All double-reinforced samples exhibited a very similar behavior. Pressurization led to immediate vertical shortening and to horizontal elongation of the hose, in both directions however to a small magnitude only (less than 1 %). A kind of stress softening effect could be observed in all directions, and for both displacement measurement methods: after the first pressurization cycle, the original dimensions of the hose sample could no longer be reached. Unlike for simple tensile tests, see Section 3.2.2, in the pressurization tests, a distinctive difference between the measurements based on image correlation, on the one hand, and on the crossbeam displacement, on the other hand, was observed; the image correlation-based displacements were consistently higher. Furthermore, comparison of Figs. 3.24, 3.25, and 3.26 reveals that the obtained results are actually not consistent, meaning that significant deviations between the samples were observed.

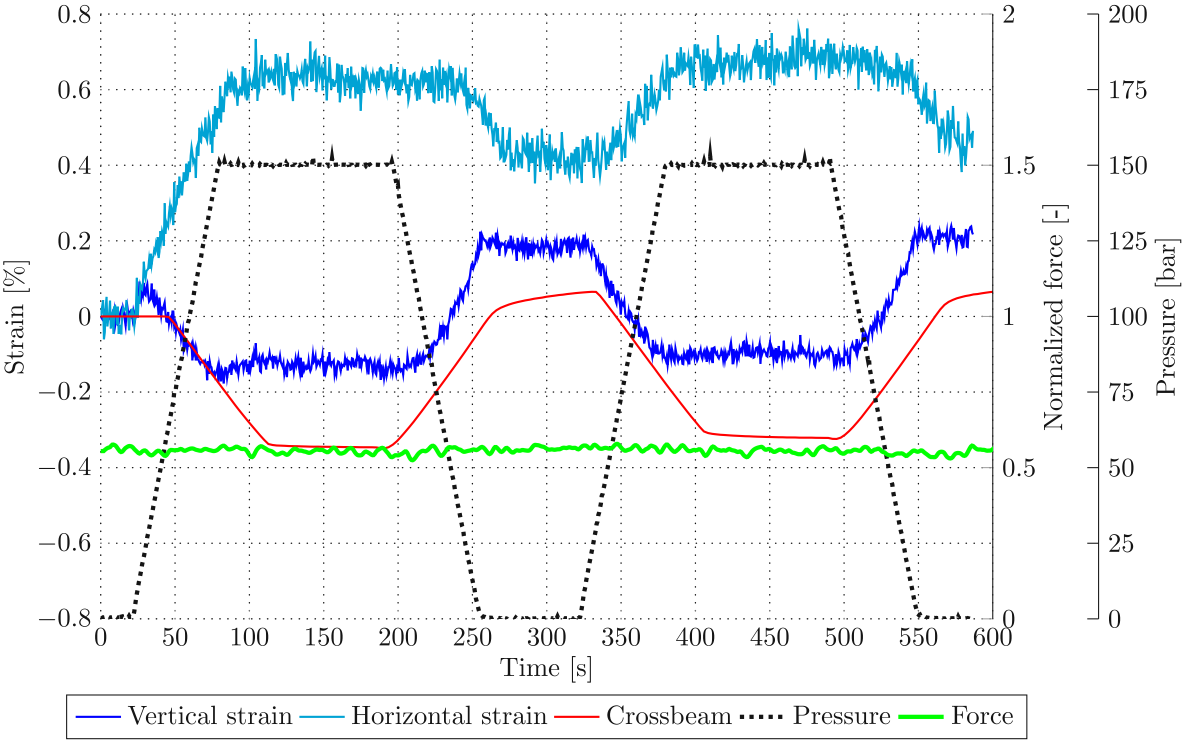


Fig. 3.24: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 1 / double reinforcement

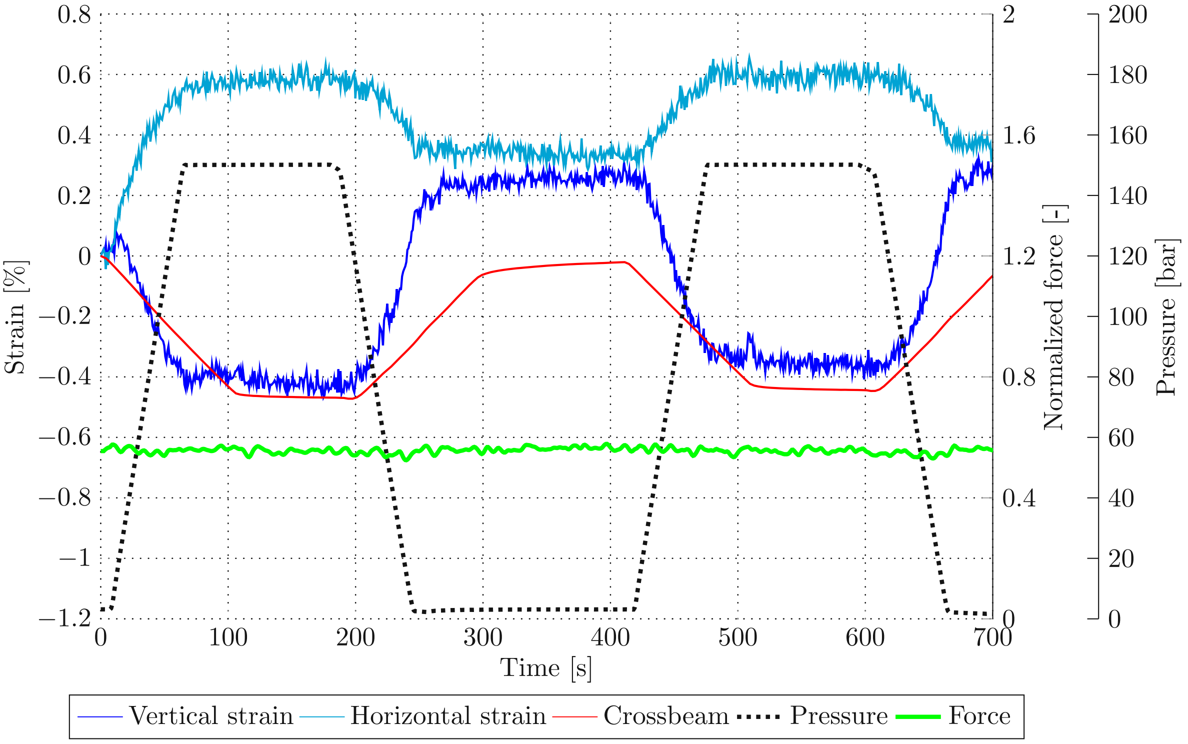


Fig. 3.25: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 2 / double reinforcement

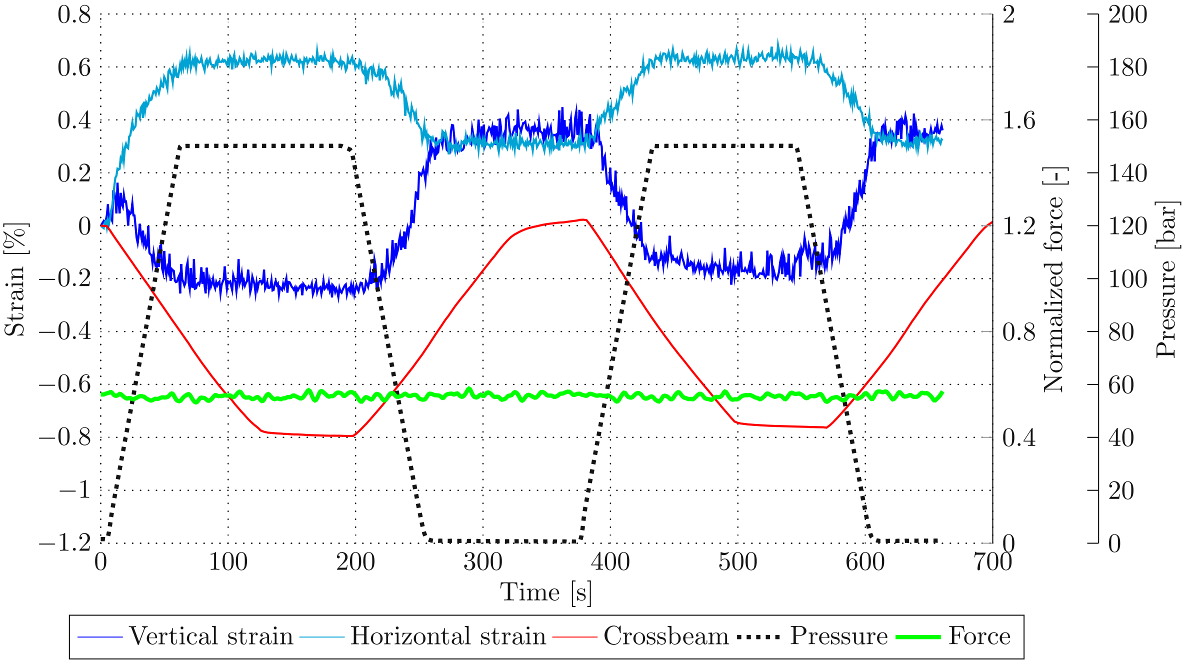


Fig. 3.26: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 3 / double reinforcement

Single reinforcement:

Qualitatively, the results obtained for the single-reinforced samples do not differ significantly from the double-reinforced ones. However, due to the reduced stiffness of the former, the displacement magnitudes are higher, see Figs. 3.27, 3.28, and 3.29..

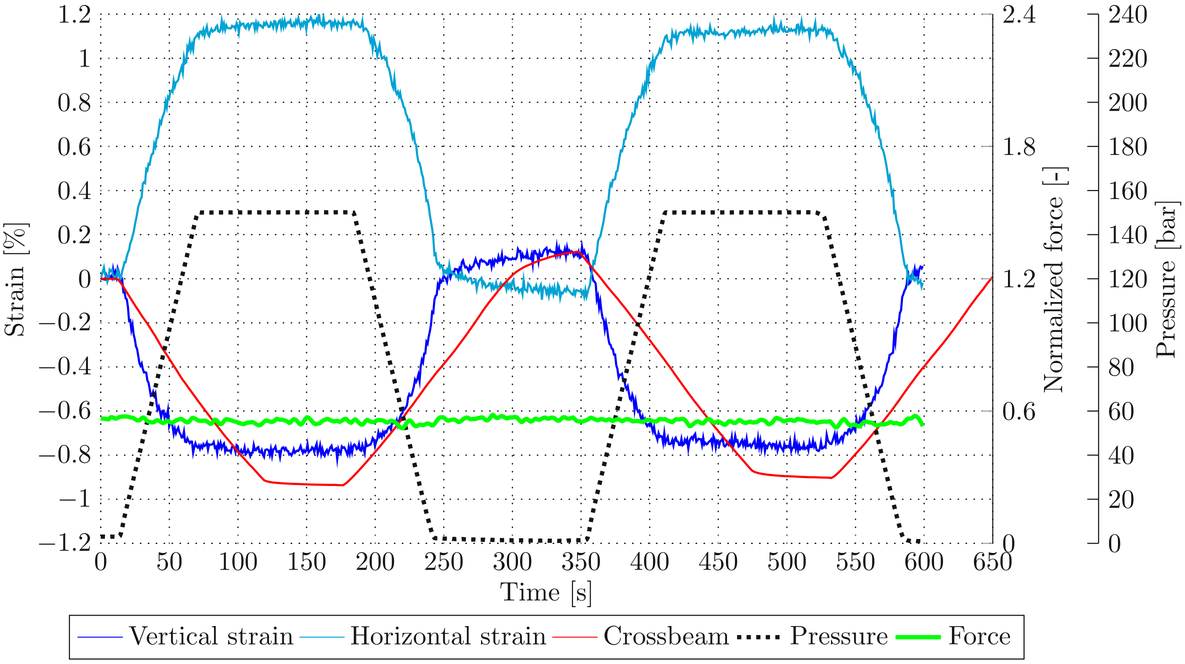


Fig. 3.27: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 1 / single reinforcement

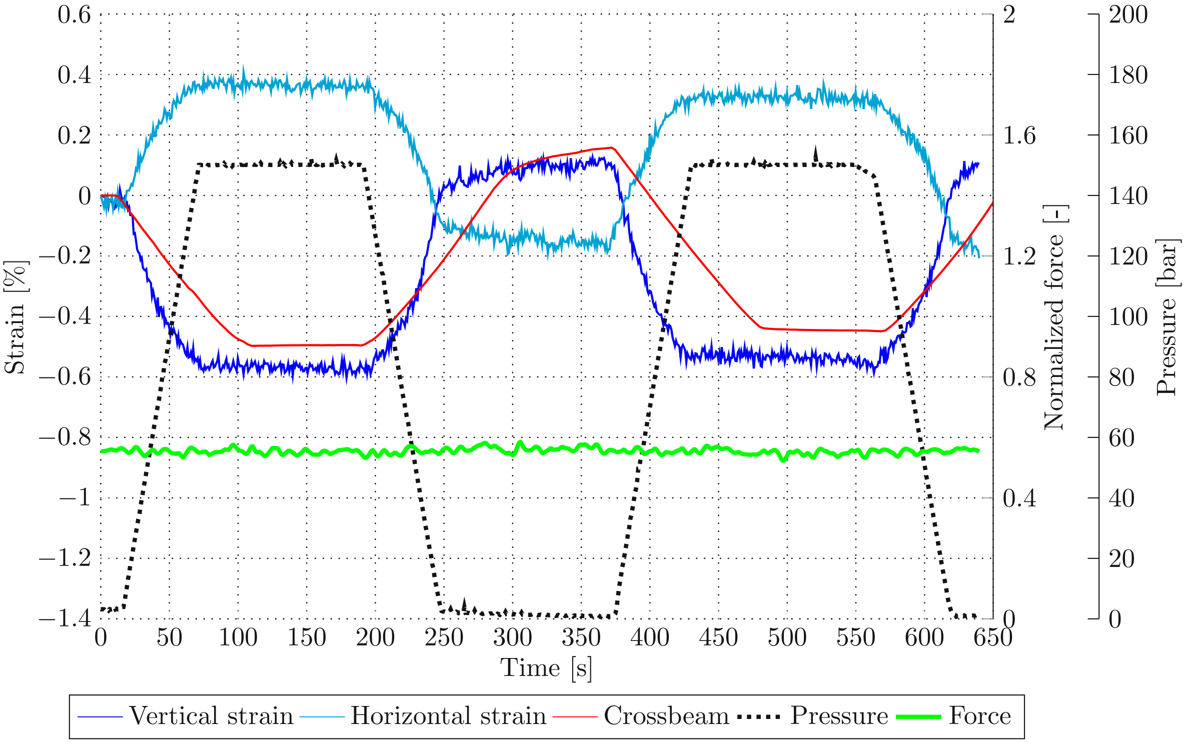


Fig. 3.28: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 2 / single reinforcement

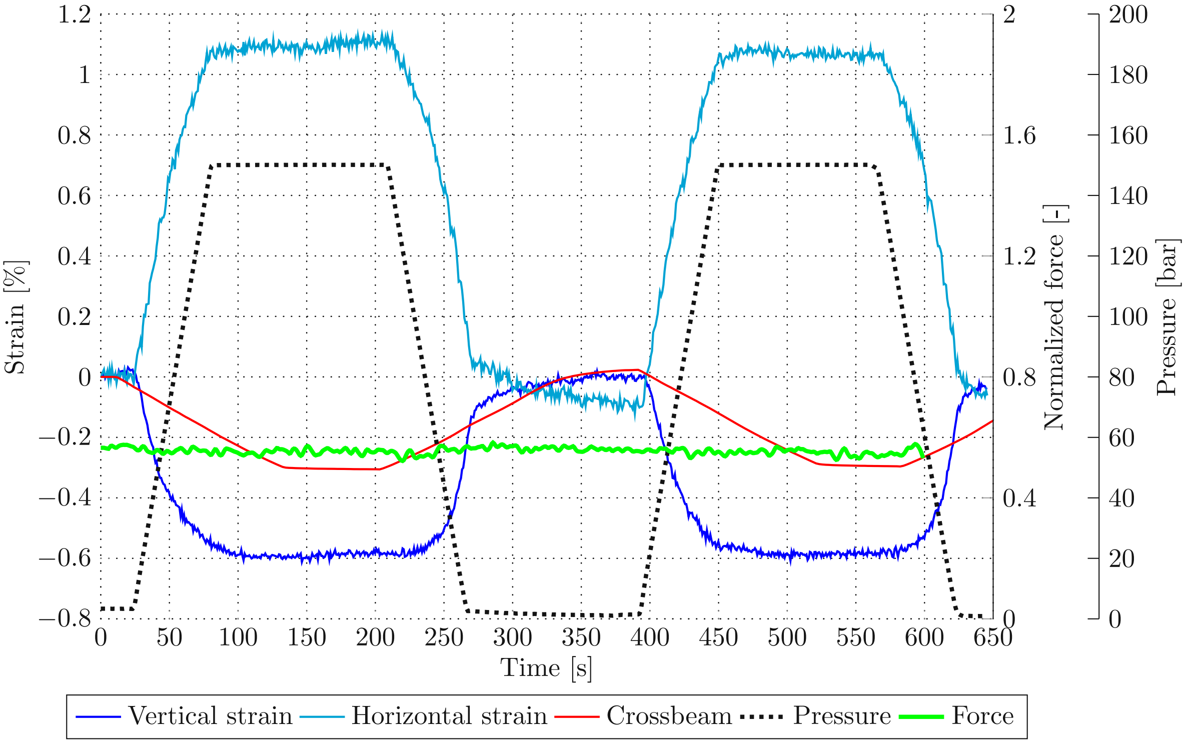


Fig. 3.29: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 3 / single reinforcement

3.2.3.2 Constant position of crossbeam

When keeping the position of the crossbeam constant, while recording displacements and the force to which the crossbeam is subjected during pressurization, the measured displacements turned out to be competely inconsistent, both quantitatively and qualitatively, irrespective of the measurement system, and irrespective of whether single- or double-reinforced samples were tested, see Figs. 3.30 to 3.35. While consequently no valid interpretation is possible, it should be noted that at least the recorded crossbeam forces were somewhat consistent

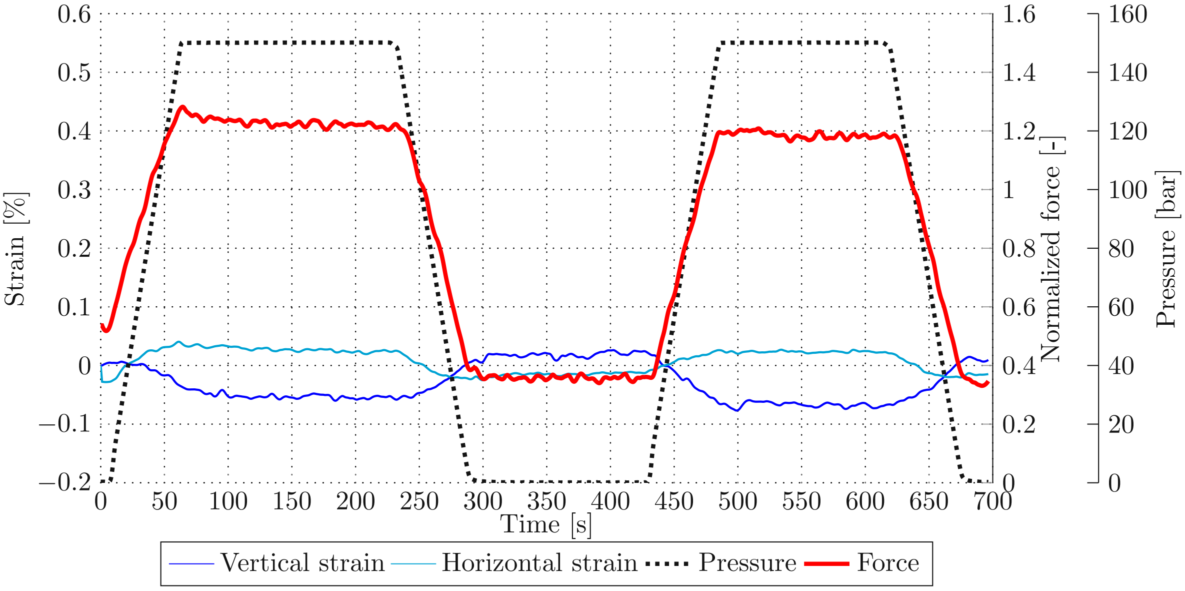


Fig. 3.30: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 1 / double reinforcement

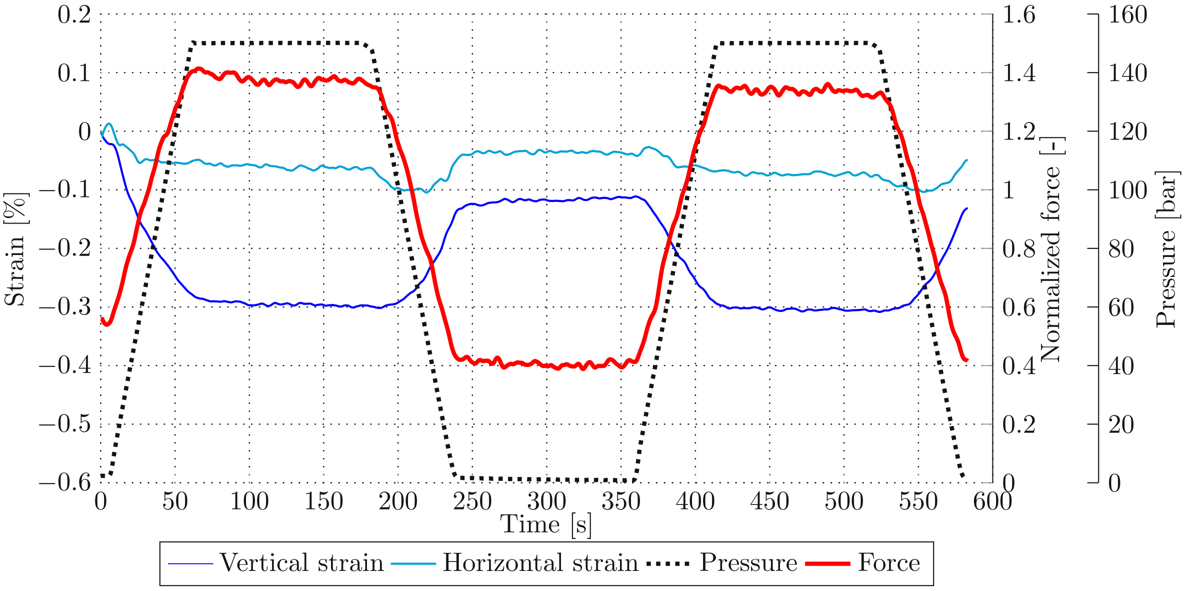


Fig. 3.31: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 2 / double reinforcement

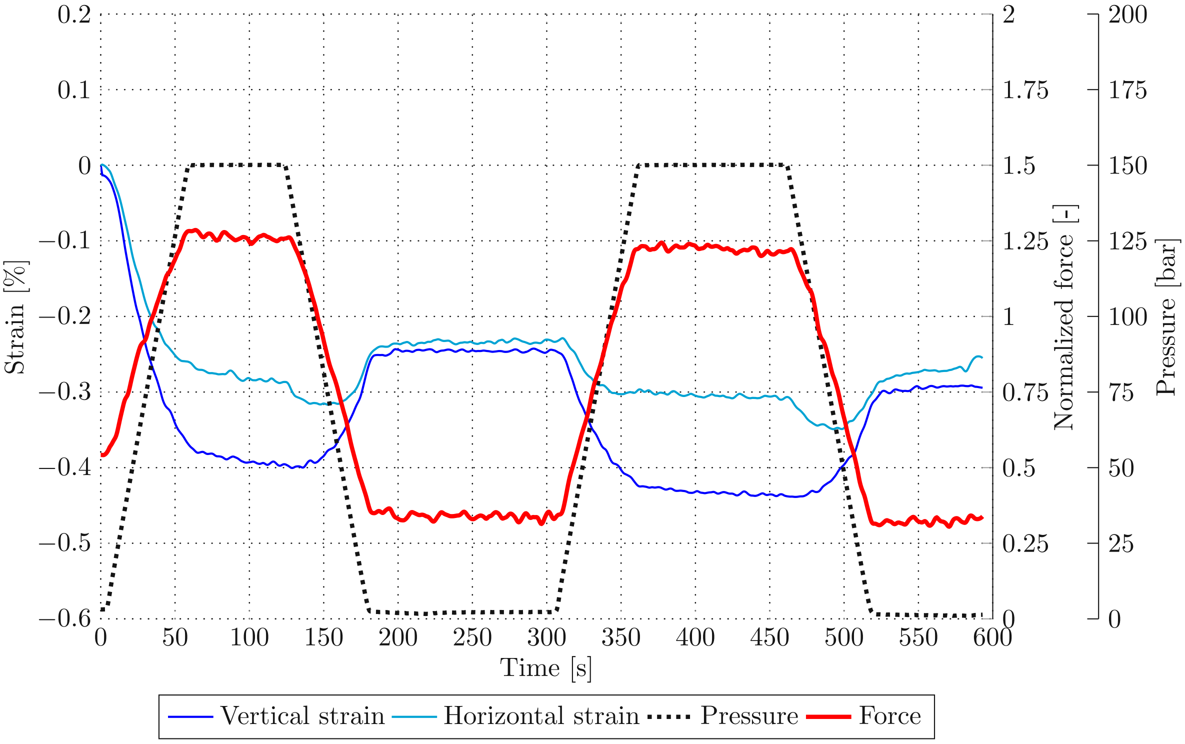


Fig. 3.32: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 3 / double reinforcement

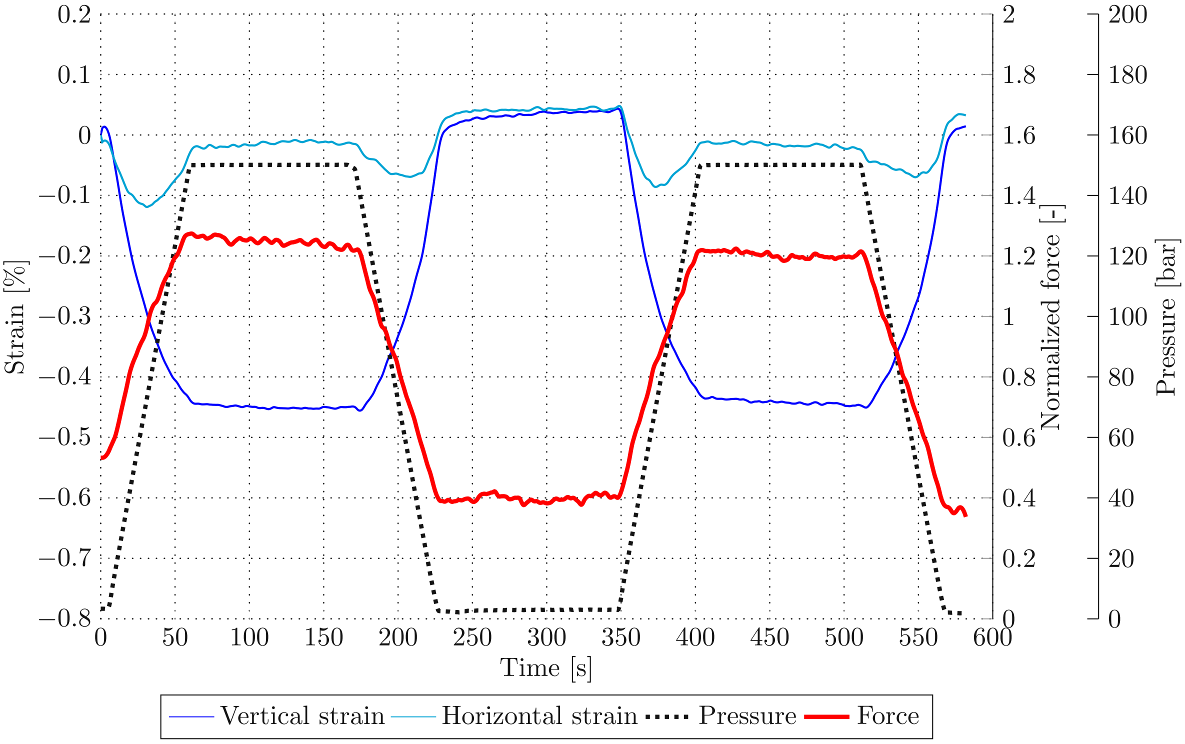


Fig. 3.33: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 1 / single reinforcement

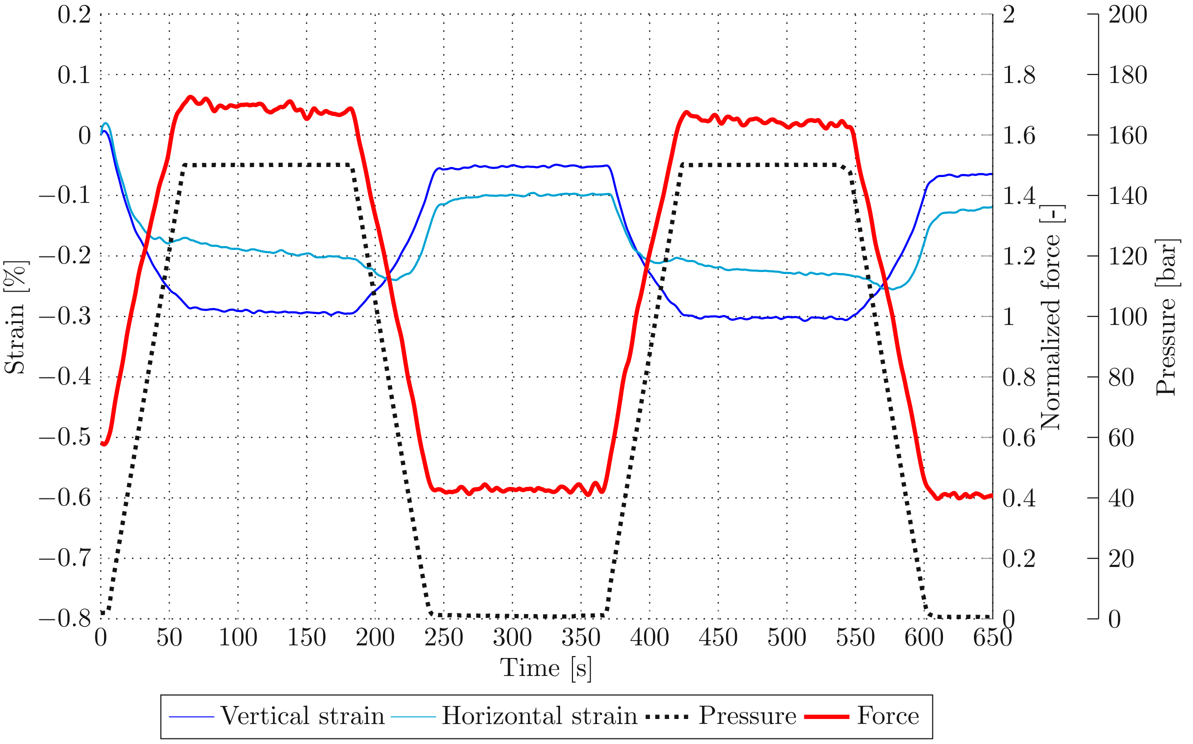


Fig. 3.34: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 2 / single reinforcement

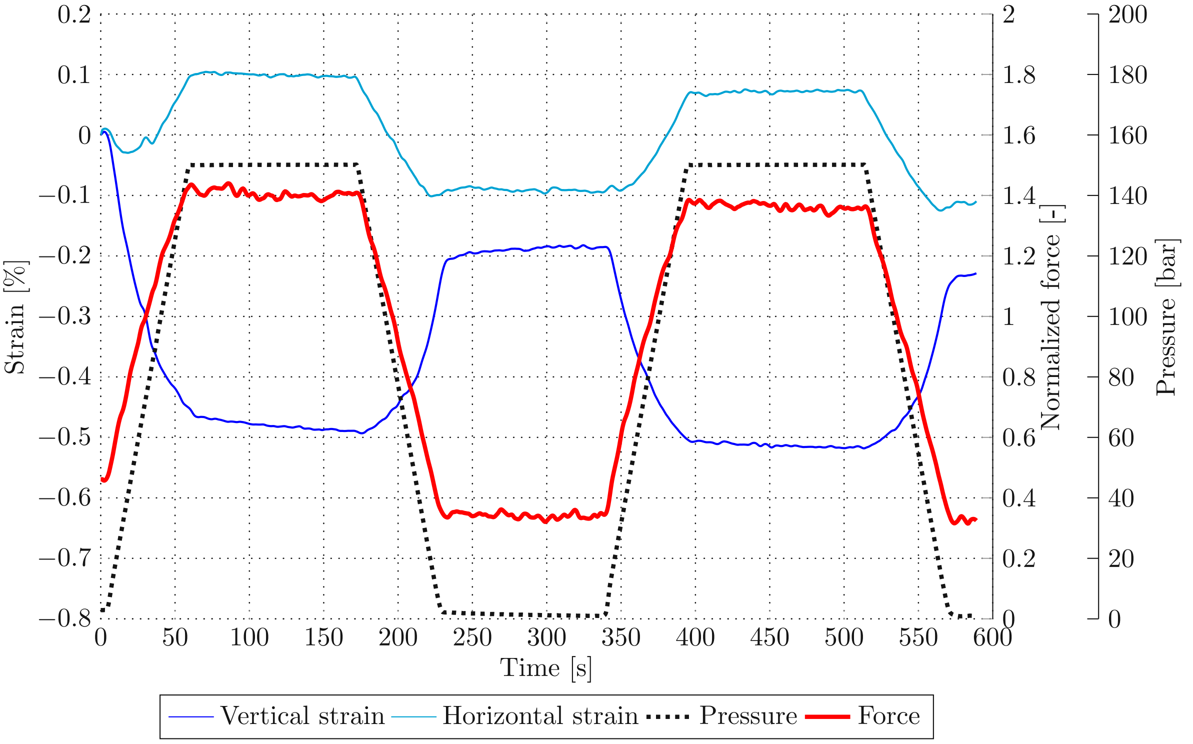


Fig. 3.35: Time course of force, pressure, and displacement in longitudinal and transverse direction, for sample 3 / single reinforcement

3.2.4 Three-point bending tests

The three-point bending tests were performed in and against the direction of the natural curvature of the samples. While subsequently only results for a loading rate of 20 mm/min are shown, also tests with a loading rate of 200 mm/min were performed. However, no differences between the results related to different loading rates could be observed. Each sample underwent five load cycles, see Figure 4.31. Since the only the first and last load cycle seem relevant, only these two parts of the curve are presented in the remaining plots. Figs. 3.36 to 3.38 show the results obtained for single-reinforced samples, loaded in direction of the natural curvature, Figs. 3.39 to 3.41 show the results obtained for single-reinforced samples, loaded against the direction of the natural curvature, Figs. 3.42 to 3.44 show the results obtained for double-reinforced samples, loaded in direction of the natural curvature, and Figs. 3.45 to 3.47 show the results obtained for single-reinforced samples, loaded against the direction of the natural curvature. As highlighted by the comparison shown on Fig. 3.48, the differences between loading directions (in versus against the natural curvature) are rather insignificant, while double-reinforced samples exhibit a significantly higher bending stiffness than the single-reinforced ones.

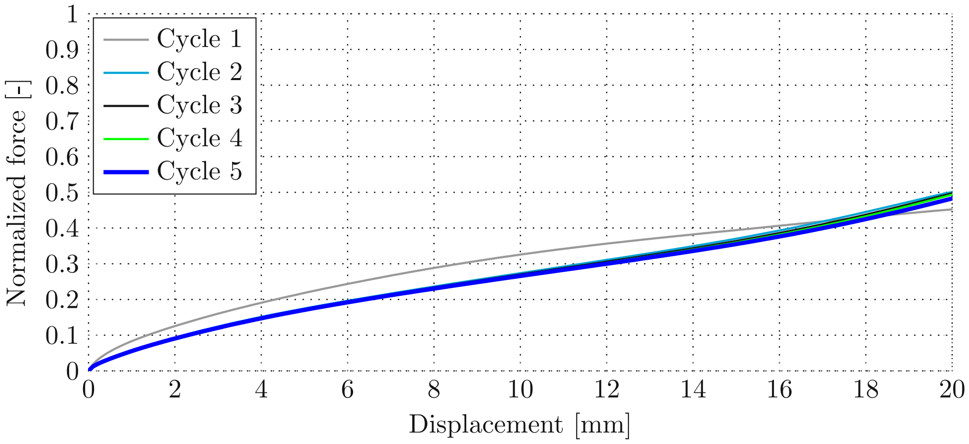


Fig. 3.36: Force-displacement curves obtained from three-point bending tests performed on sample 1 (single-reinforced, loaded in direction of the natural curvature)

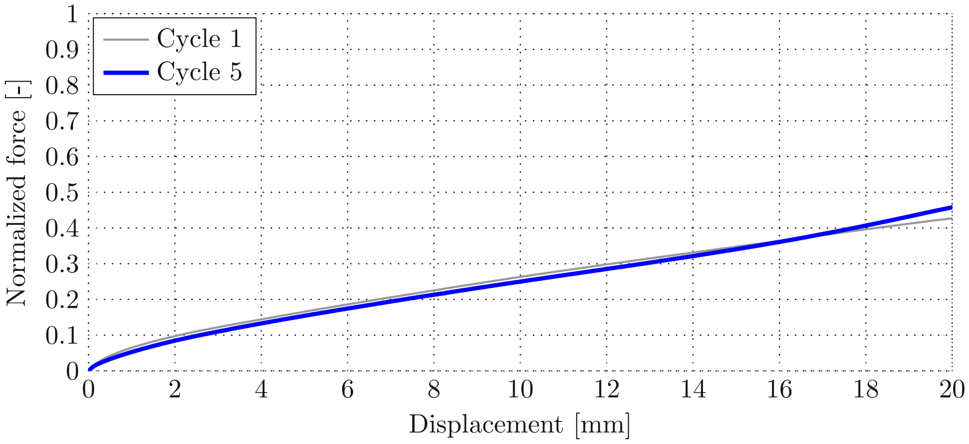


Fig. 3.37: Force-displacement curves obtained from three-point bending tests performed on sample 2 (single-reinforced, loaded in direction of the natural curvature)

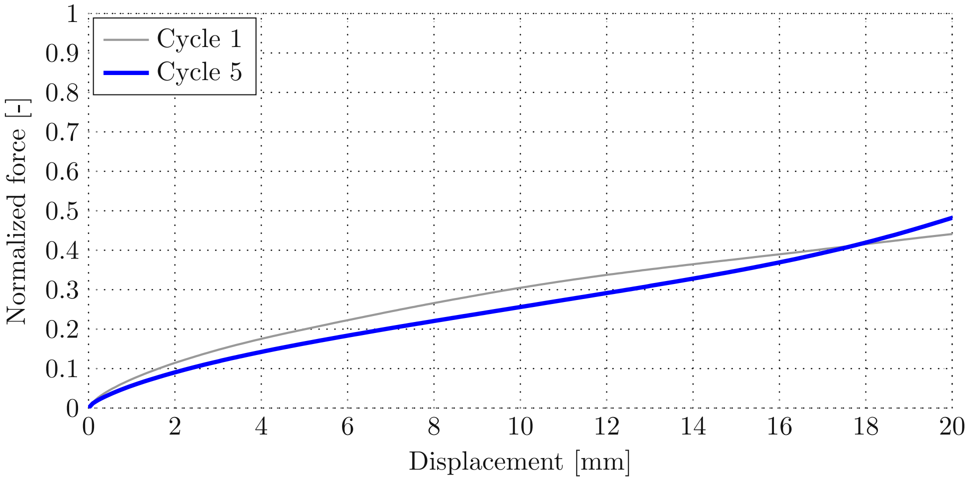


Fig. 3.38: Force-displacement curves obtained from three-point bending tests performed on sample 3 (single-reinforced, loaded in direction of the natural curvature)

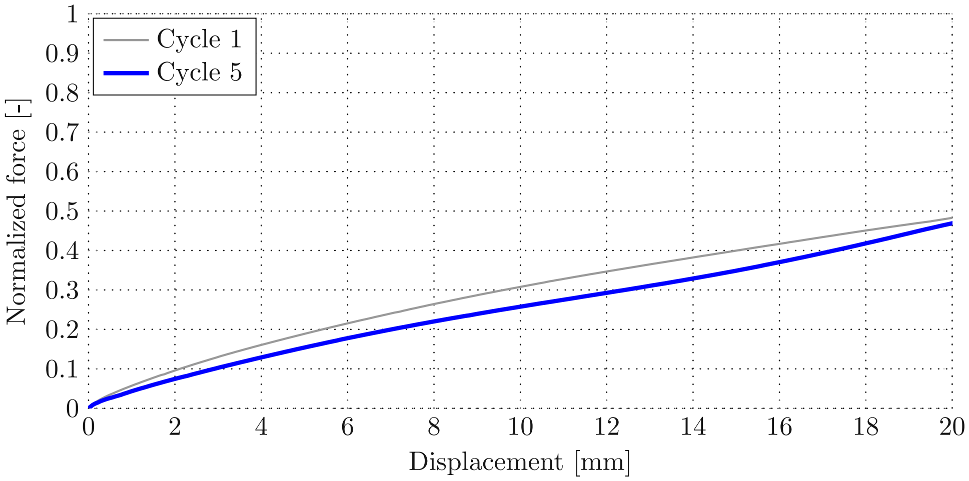


Fig. 3.39: Force-displacement curves obtained from three-point bending tests performed on sample 4 (single-reinforced, loaded against the direction of the natural curvature)

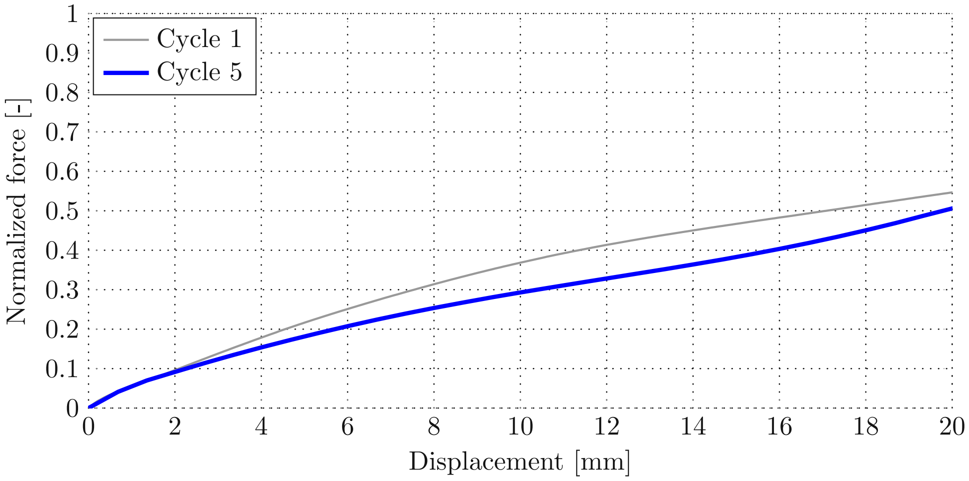


Fig. 3.40: Force-displacement curves obtained from three-point bending tests performed on sample 5 (single-reinforced, loaded against the direction of the natural curvature)

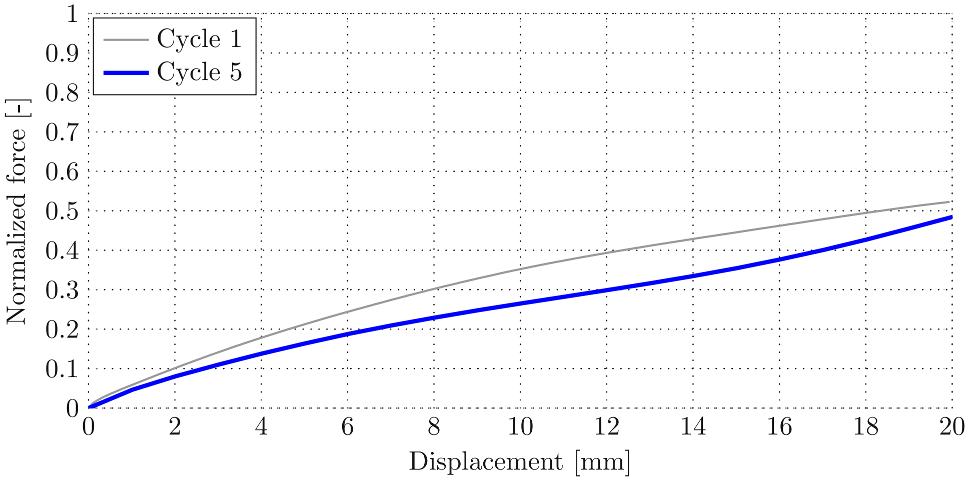


Fig. 3.41: Force-displacement curves obtained from three-point bending tests performed on sample 6 (single-reinforced, loaded against the direction of the natural curvature)

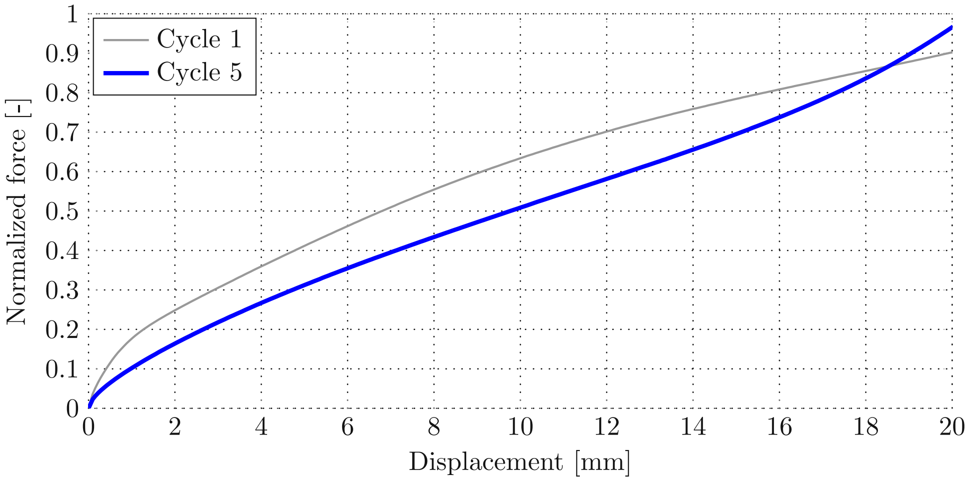


Fig. 3.42: Force-displacement curves obtained from three-point bending tests performed on sample 7 (double-reinforced, loaded in direction of the natural curvature)

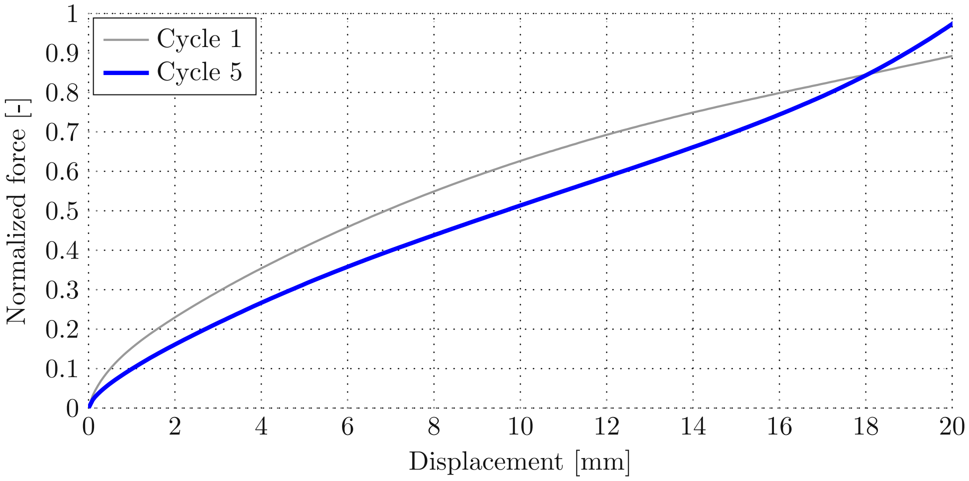


Fig. 3.43: Force-displacement curves obtained from three-point bending tests performed on sample 8 (double-reinforced, loaded in direction of the natural curvature)

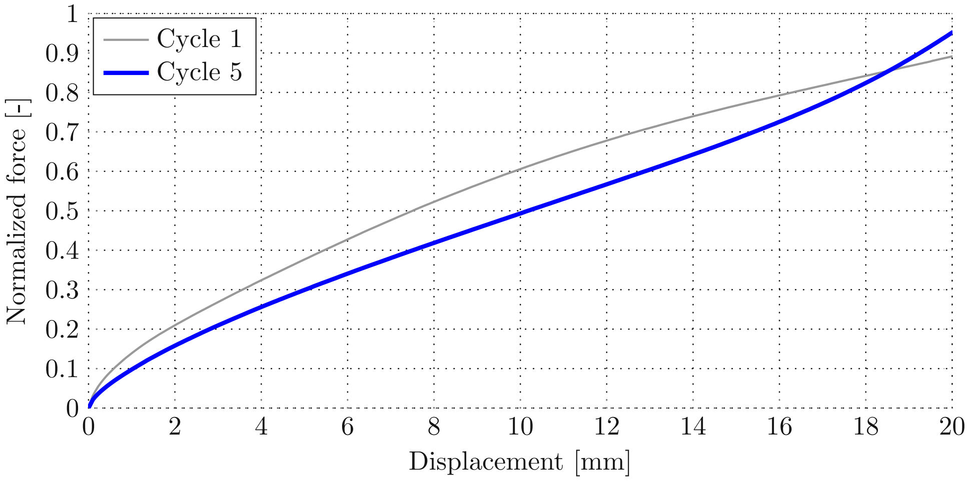


Fig. 3.44: Force-displacement curves obtained from three-point bending tests performed on sample 9 (double-reinforced, loaded in direction of the natural curvature)

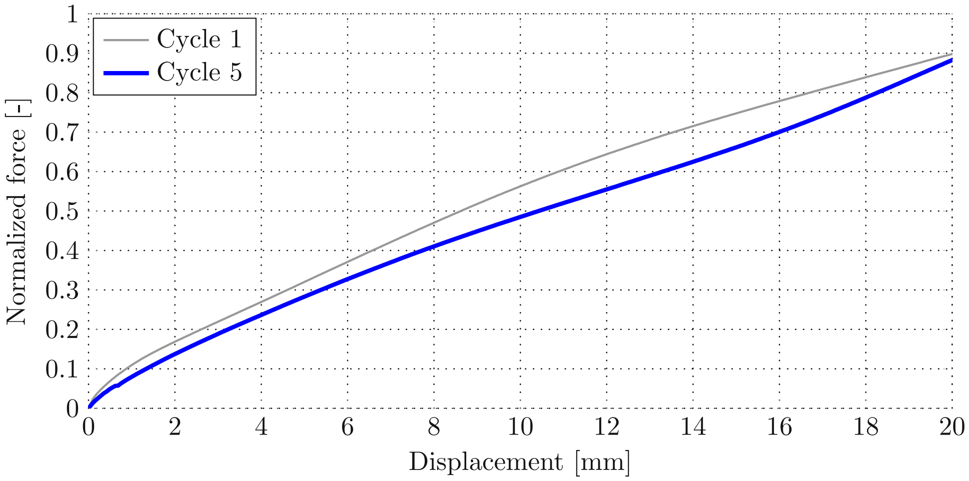


Fig. 3.45: Force-displacement curves obtained from three-point bending tests performed on sample 10 (double-reinforced, loaded against the direction of the natural curvature)

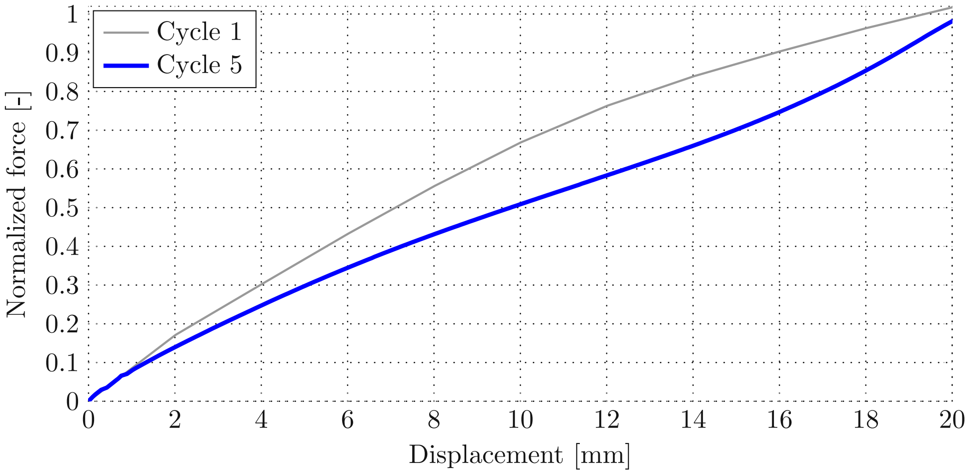


Fig. 3.46: Force-displacement curves obtained from three-point bending tests performed on sample 11 (double-reinforced, loaded against the direction of the natural curvature)

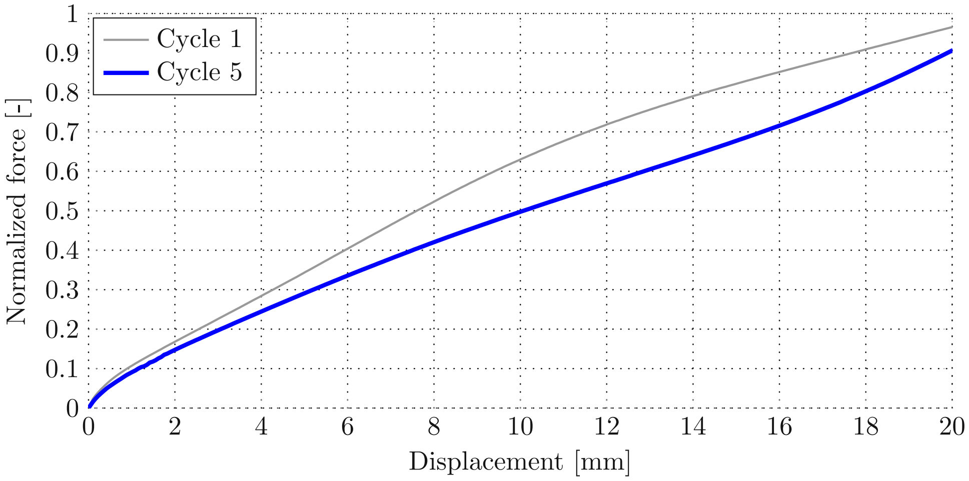


Fig. 3.47: Force-displacement curves obtained from three-point bending tests performed on sample 12 (double-reinforced, loaded against the direction of the natural curvature)

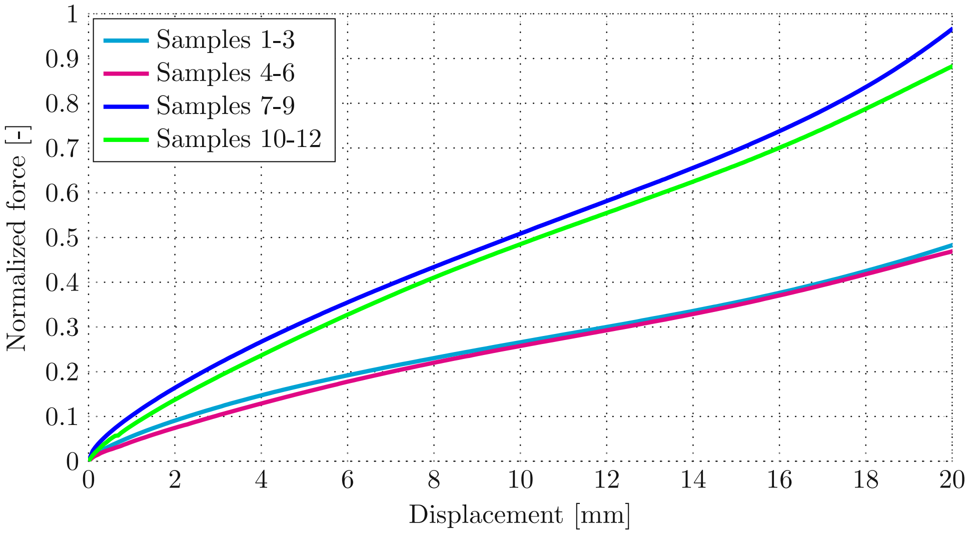


Fig. 3.48: Comparison of the average force-displacement curves obtained for four sample groups: samples 1 to 3 (single-reinforced, loaded in direction of the natural curvature), samples 4 to 6 (single-reinforced, loaded against the direction of the natural curvature), samples 7 to 9 (double-reinforced, loaded in direction of the natural curvature), and samples 10 to 12 (double-reinforced, loaded against the direction of the natural curvature)

3.3 Discussion of results

In summary, the experimental campaign presented in this thesis has provided interesting insights, summarized next:

- From a methodological point of view, some new modalities have been successfully tried out. This concerns a new image correlation approach allowing for quantification of the deformation behavior during tensile and pressurization tests performed on hydraulic hose samples. Furthermore, a new way of loading the latter has been introduced, involving pressurization, on the one hand, and keeping a constant overall stretching or a constant force, respectively, on the other hand.
- As for the tensile tests, the observed behavior was as expected. Single-reinforced samples were less stiff than double-reinforced ones. Comparing the tensile tests performed on hose samples with those performed on rubber samples, it could be confirmed that the load-bearing is mostly provided by the steel braiding, whereas the rubber, and the de-bonding of the rubber and the steel reinforcement, probably introduce some kind of non-linearity right after the start of the tensile tests. The fact that the embedded steel carries most of the applied load is also corroborated by the lack of significant viscoelastic effects, as has turned out through comparison of the results obtained for different loading rates – no significant differences could be observed.
- The pressurization tests with constant initial load and constant crossbeam position, respectively, were, in principle, successful. It seems like this type of load application can indeed lead to enriching the overall picture gained from an experimental campaign. On the other hand, it turned out quite clearly that the available hydraulic unit was by far not powerful enough. The maximum pressure used in the experiments was 150 bar, which is, however, by factor ≈ 3.3 smaller than the maximum working pressure of the studied hose samples. Especially when keeping the crossbeam at a fixed position during pressurization, the corresponding displacement data was completely inconsistent. It is believed that for higher pressures the data would have to become much more consistent, and, eventually, significant in terms of deducible interpretations. When keeping the applied force constant during pressurization, the obtained displacement data was certainly more significant but nevertheless, the too low loading does rather not allow for drawing trustworthy conclusions from the experimental data.
- The three-point bending tests yielded consistent results, showing a minor but clearly reproducible non-linear bending rigidity.
- Interestingly (but not surprisingly), more or less all experimental results exhibit some kind of non-linearity. While steel, being the main load-bearing component in the composite material, usually behaves linearly, at least when loaded in a purely elastic regime, several additional effects obviously introduce non-linearities. Firstly, this concerns of course the second component of the composite material, rubber. Secondly, steel is not contributing as bulk material but is embedded in a rubber matrix in the form of a complex braided

structure. And, thirdly, de- and re-bonding effects between steel and rubber during loading and unloading may also play a role.

Overall, the experimental results provide valuable data for validation of the computational models presented next. Furthermore, when considering the experimental studies presented in this thesis as kind of pilot study for a comprehensive, non-standard experimental campaign for revealing the mechanical behavior of steel-reinforced rubber hoses, the obtained results, and in particular the obtained inconsistencies therein, provide an extremely useful basis for improvement and extension of the applied methods. This concerns, e.g., the needed magnitude of pressurization, the image correlation system, the sample preparation, just to some of the discussed aspects.

One important aspect which has not been discussed so far concerns the computation of strains from the measured displacements. It should be noted here that, purely for the sake of comparability of the two displacement measurement systems, strains were computed by assuming tacitly that the studied material undergoes small displacement gradients, hence a linearized strain tensor can be assumed. This is not necessarily true. However, finding the correct strain measure was suspected to be a lengthy process, and would, at this point of the study, not necessarily yield additional insights. Hence, resolving this aspect was postponed to follow-up studies.

Chapter 4

Computational analyses of hydraulic hoses

As for computational analyses, it should be emphasized that the objective of this thesis is not to present a fully functioning computational strategy for simulating the mechanical behaviour of hydraulic hoses under operation conditions, but to perform a feasibility study as to which computational strategies seem to be promising in this regard, and should be further pursued in future studies. For that purpose, implementation of a two-scale approach called the submodeling technique is presented in the following subsections. But first, the fundamental concept of this technique is briefly explained.

The main idea of the submodeling technique is to study a complex, hierarchically organized material on the relevant length scales. In the present case, two analysis steps were implemented:

1. First, the structure under investigation, that is a steel-reinforced rubber hose, was studied on the macroscopic length scale. For construction of a respective model, typical macrostructural features were considered, see Section 4.1 for details.
2. Then, the output of the first analysis step served as input for the second step, which involved the analysis of a subregion of the macroscopic structure. The microstructural organization of this subregion was available in high resolution, acquired by means of micro-computed tomography (μCT). In particular, the stress and/or deformation data computed based on the macroscopic structure was utilized in terms of the boundary conditions imposed onto the microstructural domain, see Section 4.2 for details.

4.1 Macroscopic model

For the sake of comparability, two macroscopic modeling approaches were implemented, both in the framework of the Finite Element method (FEM). On the one hand, the steel reinforcement was considered in terms of beam elements, interacting with each other, and embedded in a rubber matrix, see Section 4.1.1, while alternatively, the so-called rebar method was applied, where the reinforcement layer was considered in smeared fashion, see Section 4.1.2.

4.1.1 Beam representation of reinforcement

Beam theory is a classical concept in structural mechanics. Without going into the respective details, see e.g. [43], beam theory involves considering three-dimensional beam-like structures only in terms of their axis – classically, it is required that the cross-sectional dimensions of such beams are small as compared to their longitudinal length, and it is assumed that cross sections which are, when mechanically not loaded, plane and oriented perpendicular to the beam axis, remain plane and oriented perpendicular to the beam axis upon exertion to mechanical loading. In the present case, when considering the steel braiding embedded in the hoses as beam, the beam axes certainly cannot be assumed to be straight but exhibit some sort of helical structure, see Fig. 5.1. Furthermore, the braiding does not consist of only one steel beam, but out of many, which are interacting with each other.

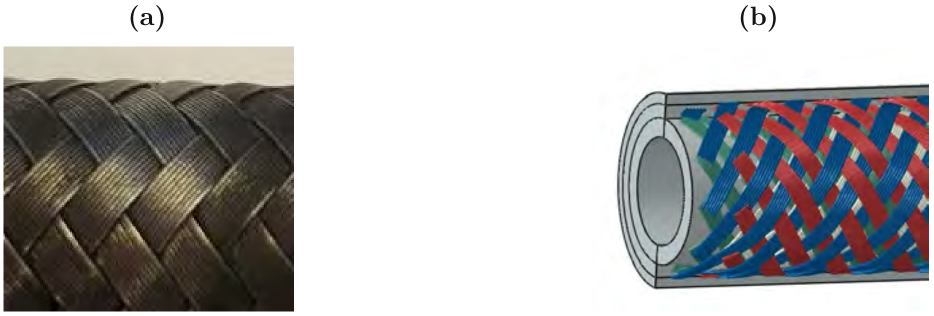


Fig. 4.1: Steel braiding embedded in the hydraulic hoses studied in this thesis; (a) photograph of an exemplary braiding structure, and (b) reconstruction of the overall structure of the hoses

In order to take the steel braiding adequately into account, a Python script was developed, see Appendix A.2, allowing to construct interacting, possibly also interwoven beams representing the actually applied braiding. Fig. 4.2 shows an exemplary output of this Python script, and Fig. 4.3 shows some exemplary beam-like representations of the steel braidings in Abaqus (version 2018). The latter is the commercial Finite Element software which was used for numerical computations in this thesis, and which allows for straightforward import of the braiding structures established by the aforementioned Python script.

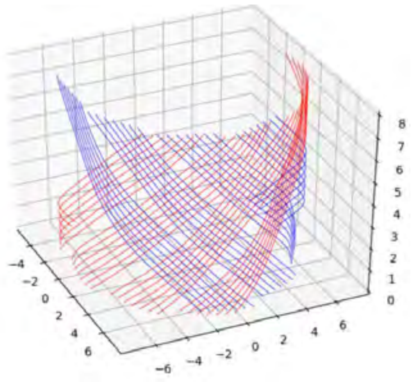


Fig. 4.2: Output of Python script which was specifically developed for replicating the (partly complex) braiding structures used in hydraulic hoses

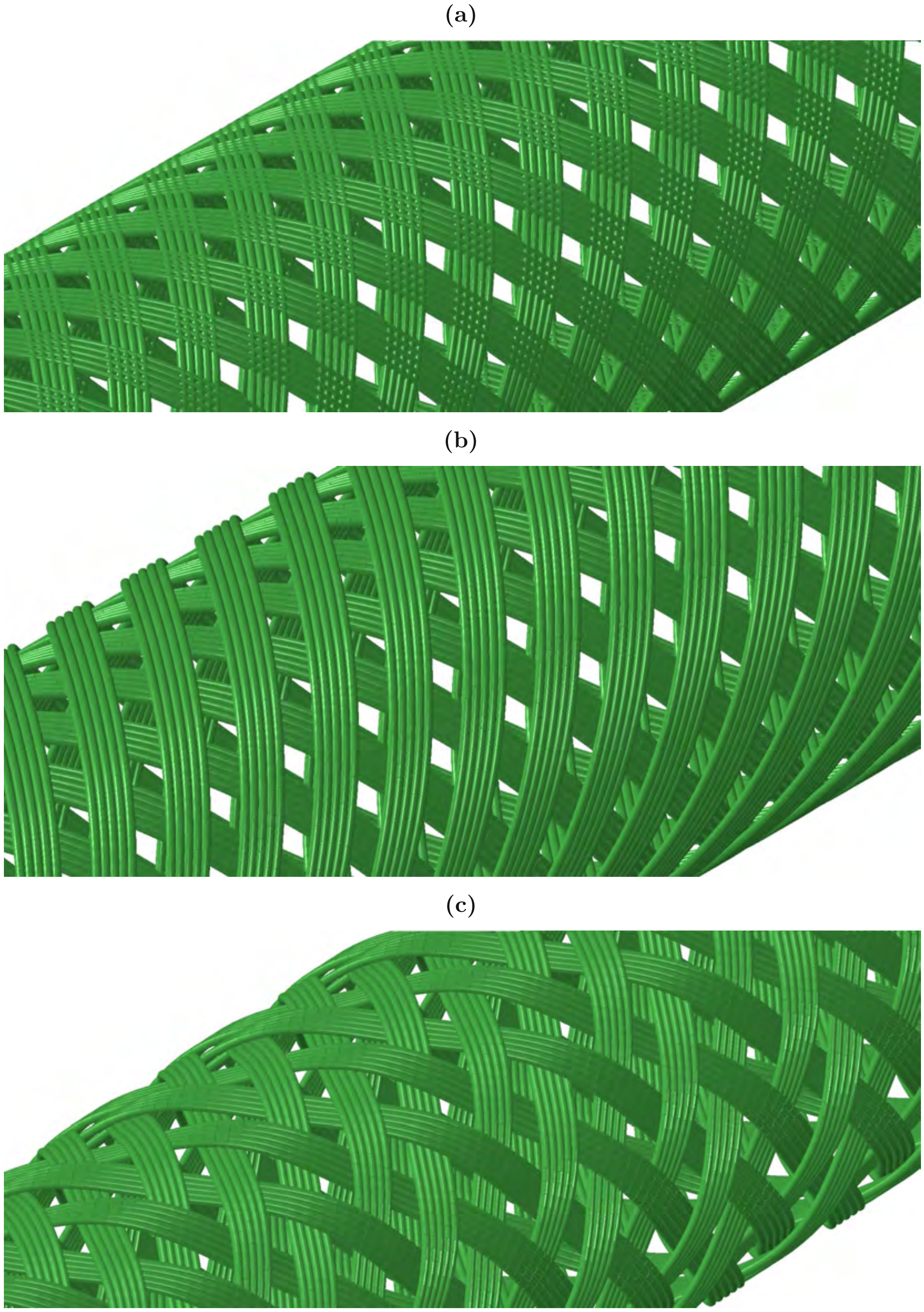


Fig. 4.3: Different braiding structures realized in Abaqus, involving (a) intersecting beam elements, (b) beam element layers separated from each by a offset of the wire diameter, and (c) an interwoven internal structure

For the computation, material models must be assigned to the respective components. For the reinforcing wires, linear elastic material behavior according to the classical Hooke's law was considered. Based on previously carried out tensile tests performed on the reinforcement steel that is used in the studied hydraulic hoses, a Young's modulus was chosen. Furthermore, a Poisson's ratio of 0.3 was assumed. The rubber layers were assumed to follow a Neo-Hookian constitutive law, reading as $W = C_{10}(\overline{I}_1 - 3) + \frac{1}{D_1}(J_{el} - 1)^2$ where W is the strain energy density, C_{10} is a material constant which relates to the shear behavior, D_1 is a material constant which relates to the bulk compressibility and is set to zero if the rubber is considered as fully incompressible, $\overline{I}_1$ is the first invariant of the isochoric part of the right Cauchy-Green deformation tensor, and J_{el} is the elastic volume ratio [44]. The material parameters C_{10} and D_1 are chosen based on fitting the tensile tests presented in Section 3.2.1.

The interaction behavior between the rubber matrix and the steel wires was considered based on using so-called embedded elements. In this technique, the translational degrees of freedom are limited when an embedded element lies within in a host element.

For the simulation, only a hose section of 150 mm was used. The model consisted of a total of 53400 elements, 39000 of which were beam elements, and 14400 hexahedral elements representing the rubber layers.

4.1.2 Rebar representation of reinforcement layer

The reinforcing bar (or, in short, rebar) method involves reducing the reinforcement layer to a corresponding, representative reinforcement shell, see Fig. 4.4(a). This shell has to match the mechanical properties of the actual layer it is supposed to represent as well as possible. When implementing such approximative representation of the reinforcement layer in the commercial Finite Element software Abaqus (version 2018), the following parameters must be defined: the thickness of the shell, the internal material orientation, relating to the orientation of the steel braiding, see Fig. 4.4(b) and (c), the cross-sectional area of bar element, the offset of the bars to the surface right in the middle of the wire layer, and the number of embedded bar elements. Notably, this method assumes regular spacing between the wires.

Rebar elements itself act as a kind of smeared elements which consist of the reinforcement and the host matrix. The rebar elements which represent the reinforcement are located in the reinforcing layer. The latter is represented by continuum elements, whose (anisotropic) stiffness follows from the interaction of the matrix and the steel wires, considering thereby the orientation and volume fraction of the latter. This method, previously applied, e.g., for simulation of car tires [45], allows to be computationally much more efficient, since the reinforcement wires do not have to considered explicitly.

The chosen material parameters were the same as for the beam-reinforced model. The model was created at a lenth of 75 mm, and, due to symmetry, only one quarter of the hose was considered, using for that purpose 6000 elements.

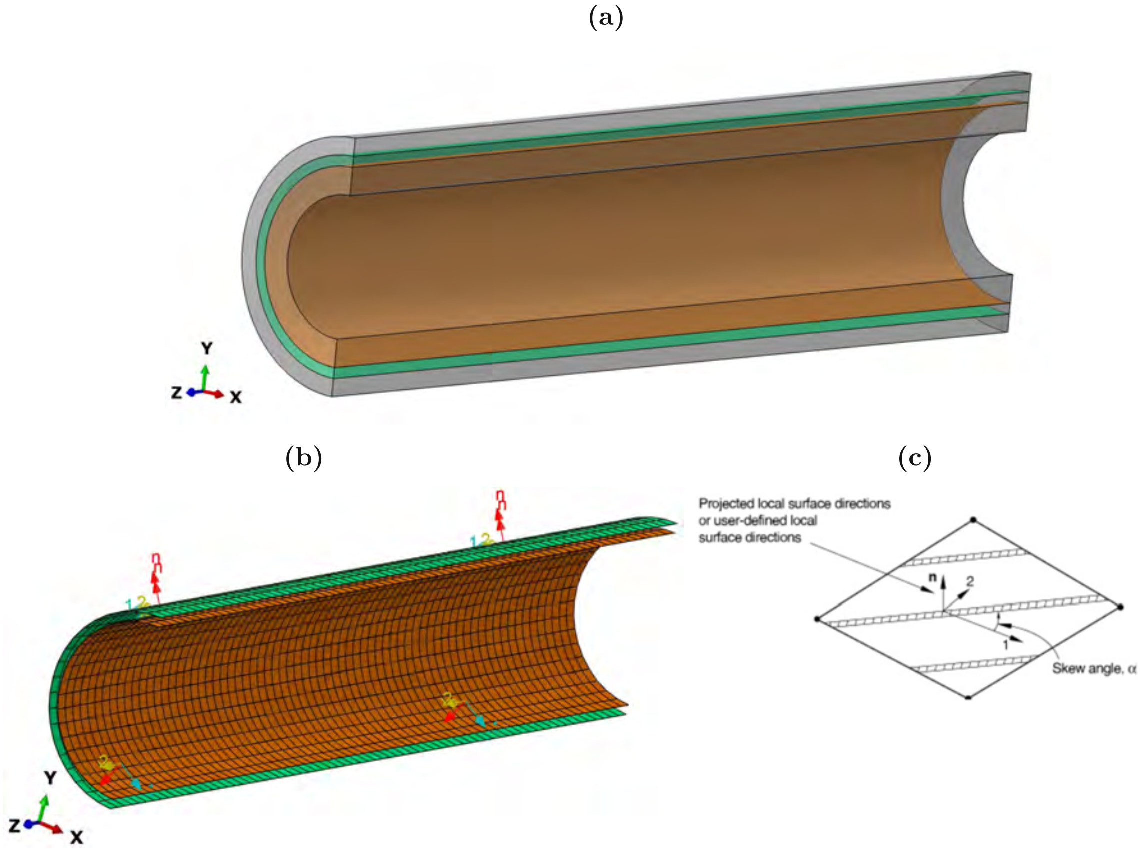

(a)
Y
Z
X
(b)
n
1
2
Y
Z
X
(c)
Projected local surface directions or user-defined local surface directions
n
2
1
Skew angle, α

Fig. 4.4: Rebar method: (a) embedded rebar shell, (b) rebar orientation, (c) definition rebar orientation [46]

4.1.3 Numerical studies and comparison with experimental data

In order to find out whether the chosen model representations make sense or not, the aim of the subsequently presented set of numerical studies was to replicate the experimental results presented in Section 3. Thereby, it should be noted that only tensile and three-point bending tests were considered, as the results of the pressurization tests were not consistent.

As compared to the laboratory tests on hose specimens, see Sections 3 and 4, several simplifications were made:

- The natural curvature of the specimens was neglected; hence, in the simulations unloaded specimens were assumed to be perfectly straight.
- The steel wires were assumed to be perfectly embedded in the rubber matrix, neglecting possibly occurring irregularities.

4.1.3.1 Three-point bending tests

Here, the aim was to replicate the displacement-driven three-point bending test described in Section 3.1.3, see Fig. 4.5 for the respective boundary conditions. The compression die and the support elements were considered in the form of infinitely stiff shell elements, owing to the fact that these components are by orders of magnitude stiffer than the hose. Furthermore, symmetry of the considered hose section was pre-defined with respect to the XY-plane and the YZ-plane. All translations and rotations were locked for the compression die and the support elements, except for the compression die the pre-defined movement in Y-direction was allowed.

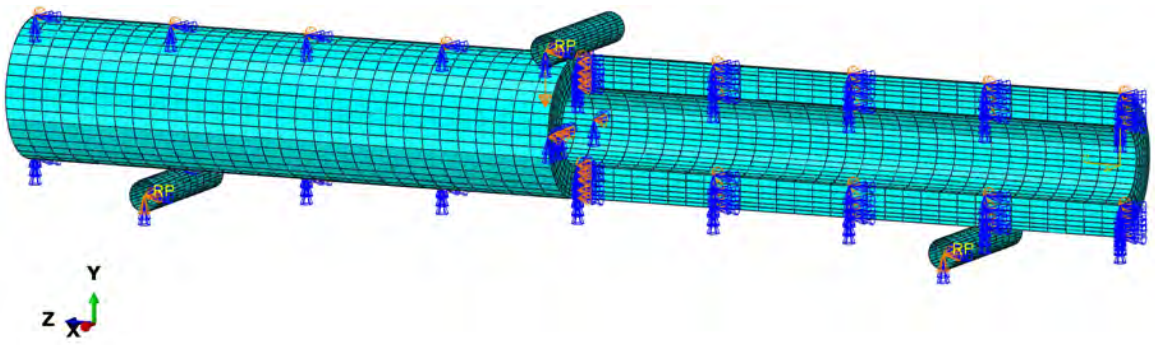


Fig. 4.5: Model for 3-point-bending-test on double-reinforced hose section

An exemplary result of the simulated three-point bending tests is depicted in Fig. 4.6, with displacements in Y-directed indicated by a suitable color map.

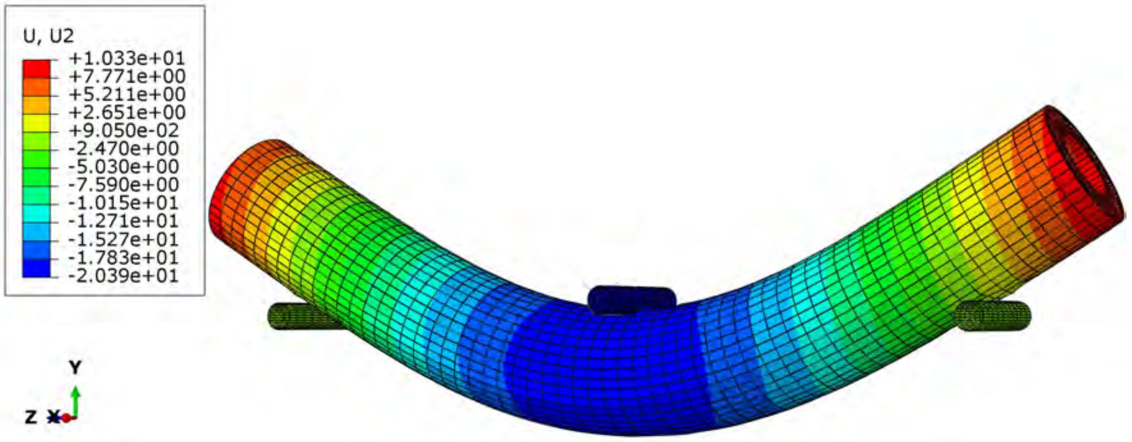


Fig. 4.6: Displacements in Y-direction resulting from numerical simulation of three-point bending tests, considering beam element-reinforced (double reinforcement) hose

Furthermore, Fig. 4.7 shows a comparison of all model representations for a double reinforced hose with the experimental data, in terms of applied force versus displacement. Obviously, the overall stiffness of the rebar model is much too low, as the force needed to induce the pre-defined displacements is much lower than the experimentally applied forces. In fact, the force resulting for the rebar model is just by approximately 50 % higher than the force needed to displace a pure rubber model as required. On the other hand, the beam model representations are much closer to the experimental data than the rebar model, while the differences between different braiding structures turned out to be not overly significant.

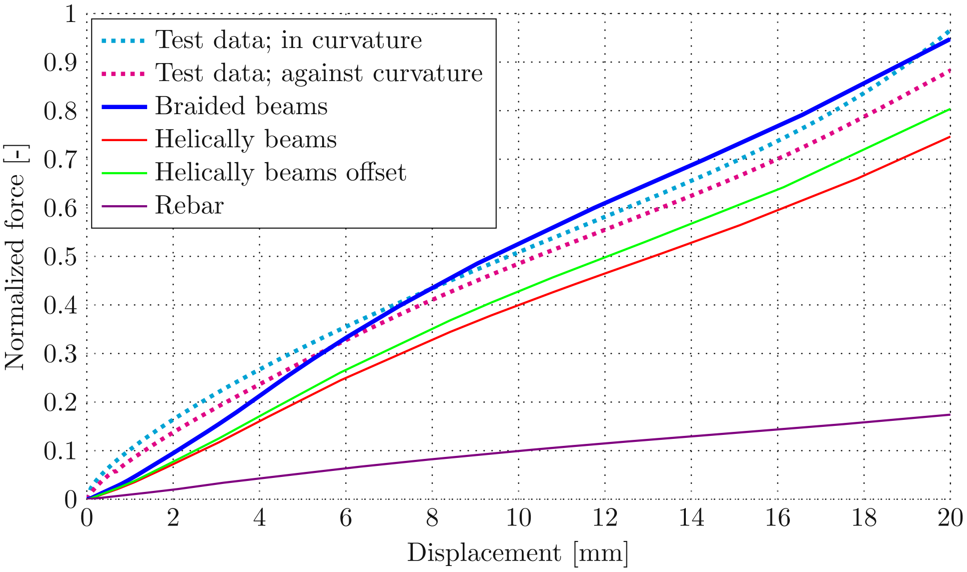


Fig. 4.7: Comparison of all tested methods with the experimental data (double reinforcement)

Fig. 4.8 shows a comparison of the experimental data with simulated data for a single-reinforced hose. The simulation was carried out here only with a beam-reinforced model which took into account the braiding structure. The plot shows that the maximum force reached at a displacement of 20 mm was approximately 15% lower than the corresponding experimental values. In general, the model-predicted hose stiffnesses turned out to be a bit lower than the experimentally revealed ones, particularly right after the start of the tests.

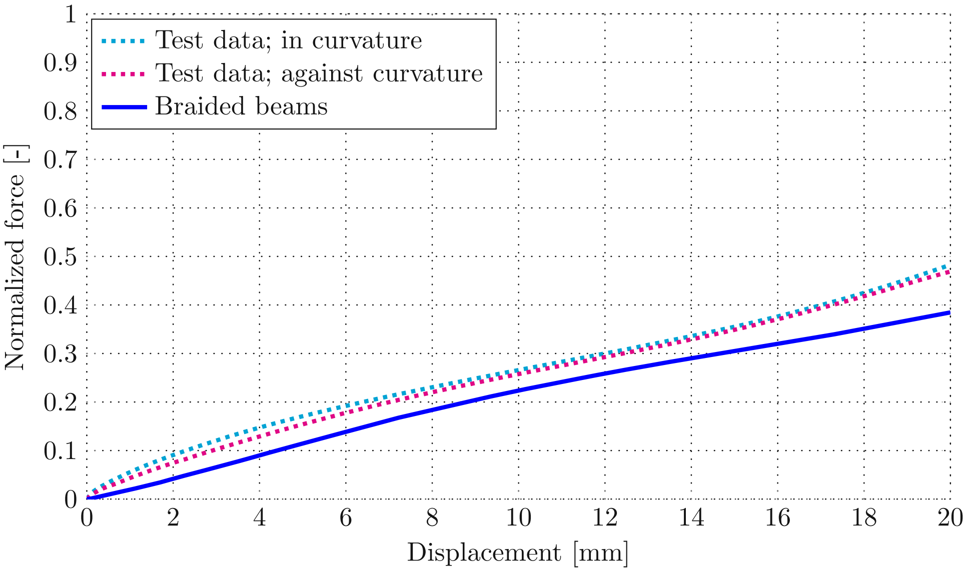


Fig. 4.8: Comparison of the model with the braided beams with the experimental data (single reinforcement)

Furthermore, some exemplary results for a three-point bending test of a double-reinforced hose section are shown for the interwoven braiding structure, namely in terms of von Mises stresses occurring in the steel reinforcement, see Fig. 4.9. Notably, the von Mises stresses in the outer wire layer turned out be higher than in the inner layer, reaching up to approximately 1.85 GPa.

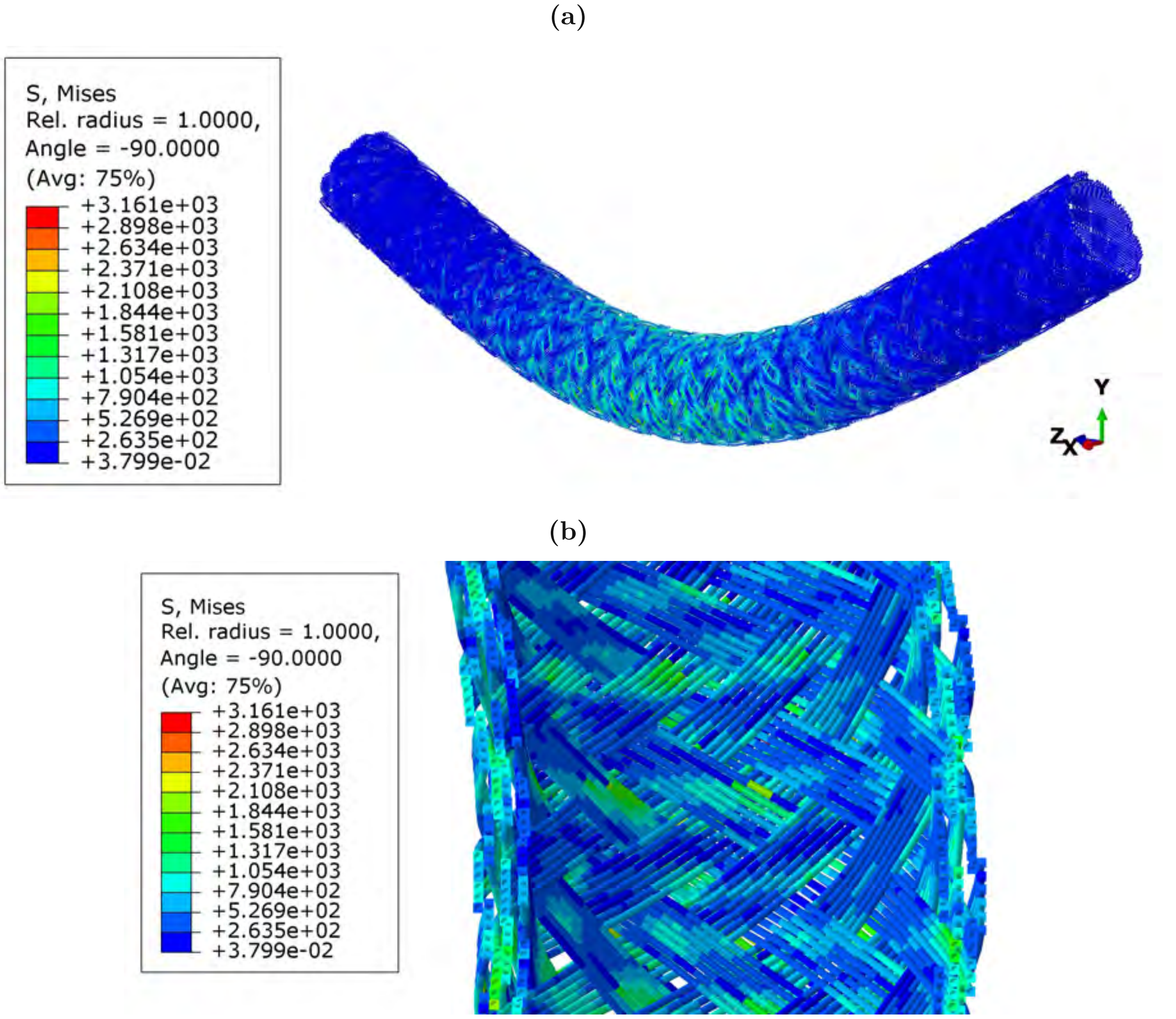


Fig. 4.9: Simulation results (double-reinforced hose): (a) von Mises stress on the whole steel braiding, (b) von Mises stress for a detailed area of the steel braiding

4.1.3.2 Tensile tests

Simulation of tensile tests comprised application of a pre-defined tensile force and recording the corresponding displacements. The corresponding boundary conditions are depicted in Fig. 4.10. In addition to the displacement and loading boundary conditions, coupling constraints were considered, in order to allow the use of concentrated forces.

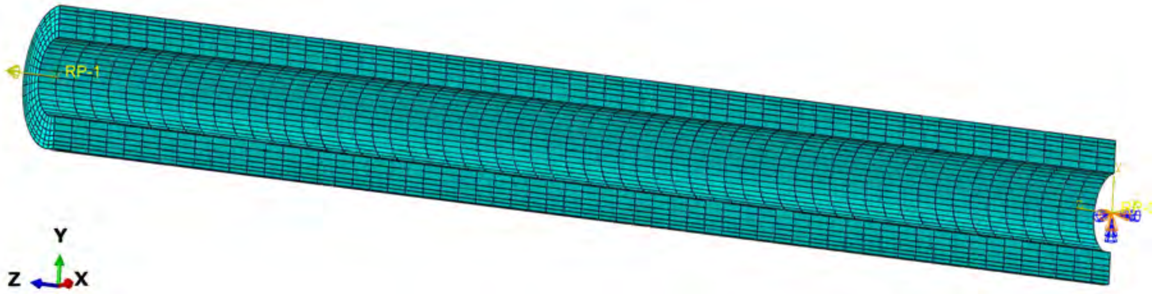


Fig. 4.10: Hose model with boundary conditions applied to simulate tensile tests (double-reinforced hose section)

Results of these simulations are depicted in terms of a displacement-related color code in the three-dimensional model, see Fig. 4.11, and in terms of a comparison of simulation results and experimental data, see Fig. 4.12. It turned out that, as for the three-point bending tests, the stiffness of the rebar model was much lower than of the beam model, while the differences between the different braiding structures was marginal. Comparing model predictions and experimental results, it seems that the beam representation of the steel wires yield a good agreement, while the experimentally tested hoses were much stiffer during the first 2 mm of displacement.

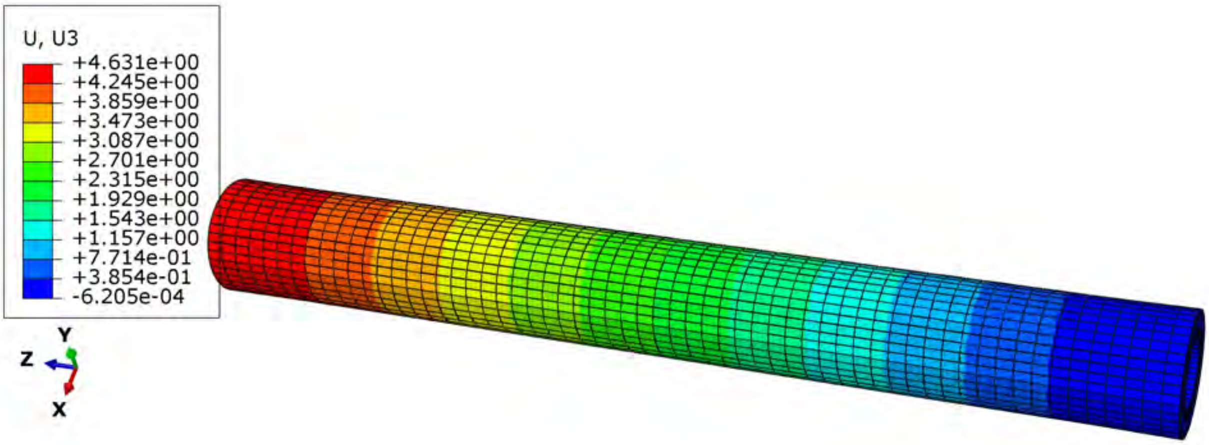


Fig. 4.11: Displacements in Z-directions

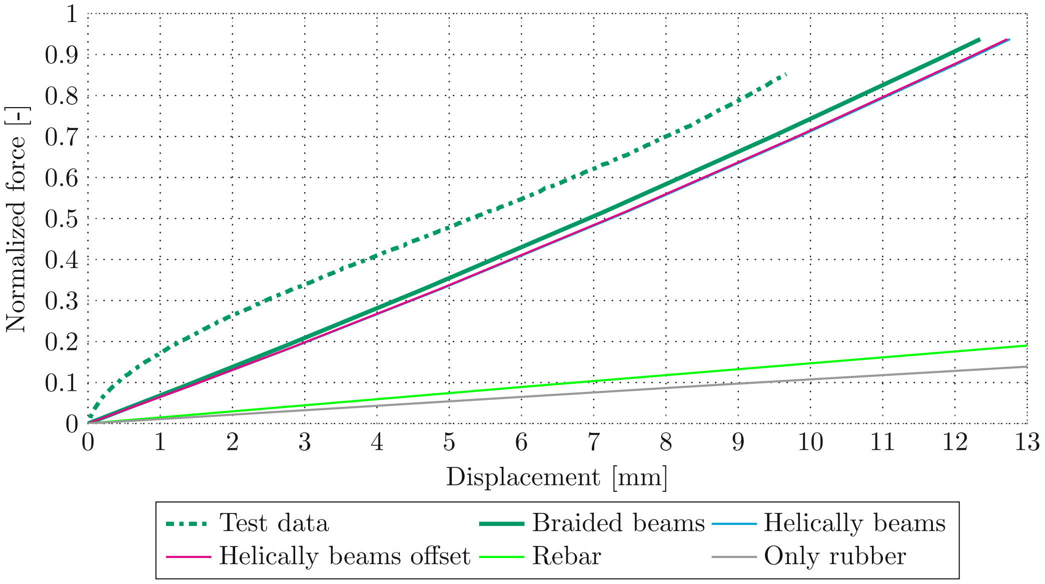


Fig. 4.12: Comparison of the model with the braided beams with the experimental data (double reinforcement)

Fig. 4.13 shows the results for a single-reinforced hose, considering again braided beams. As for the double-reinforced hose, the model underestimates the initial stiffness, directly after load application, while the agreement between the model-predicted and experimentally obtained force-displacement curves is satisfying at a larger stage of the experiments.

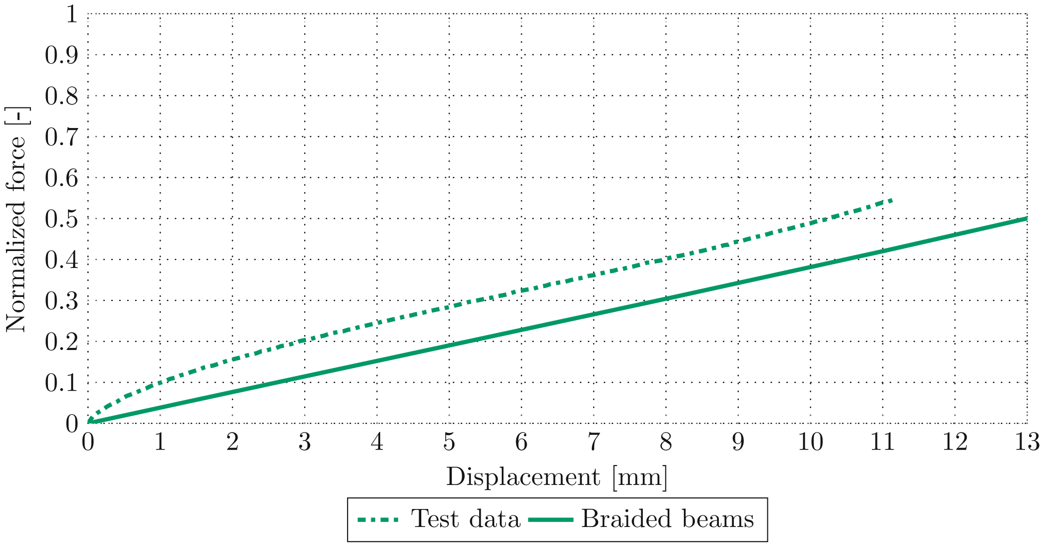


Fig. 4.13: Comparison of the model with the braided beams with the experimental data (simple reinforcement)

Furthermore, some exemplary results are shown for the interwoven braiding structure for a double-reinforced hose section, namely in terms of strains in Z-directions and in X-direction, as well as in terms of the von Mises stresses occurring in the steel reinforcement, see Fig. 4.14. While the experimentally recorded strain in Z-direction was a bit higher than the simulated one (3.84% versus 3.48%), the transversal strains were also close to each other – 2% according to the simulation and 2.2% according to the experimental data. Notably, the von Mises stresses in the outer wire layer turned out be higher than in the inner layer, reaching up to approximately 500 MPa.

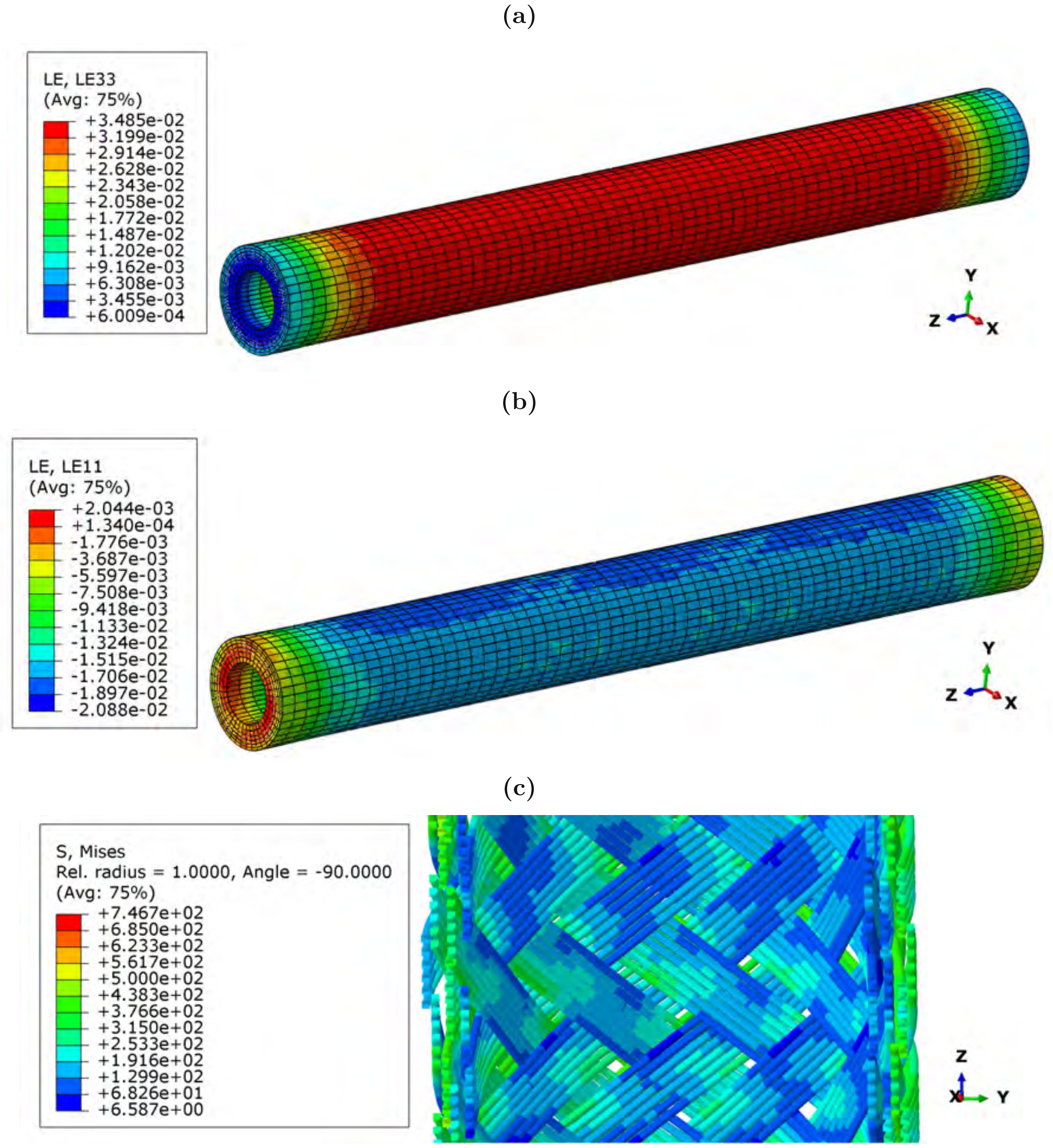


Fig. 4.14: Simulation results (double reinforced hose): (a) nominal strain in Z-direction, (b) nominal strain in X-direction, (c) von Mises stress of the wire layers

4.2 Micro model

4.2.1 Acquisition of image data based on micro-computed tomography

The microstructure of single- and double-reinforced hydraulic hoses was acquired by means of a micro-computed tomography (μCT). In the course of the project summarized in the present thesis, two types of scanners were used. On the one hand, a micro-computer tomograph (Scanco Medical μCT 100) which is commonly used in the biomedical field. However, it turned out that the X-ray energy (45-90 kVp) limitations of this scanner implied non-removable artefacts in the gained image data, see Fig. 4.15(a); also using a chopper filter did not yield the rquired improvement of the images, see 4.15(b), thus rendering the use of such images questionable. Hence, on the other hand, a industrial computer tomograph, commonly used in industrial applications, was used, allowing for application of higher X-ray energies (up to 120 kVp). The such captured CT data exhibited much less artefacts, see Fig. 4.15(c) and (d), at a resolution of 20.83 μm. On this basis, 3D reconstructions of the steel braiding could be achieved, see Fig. 4.16(a). Some details needed however particular attention; namely contact regions of similar materials, such as steel-steel contact, see Fig. 4.16(b).

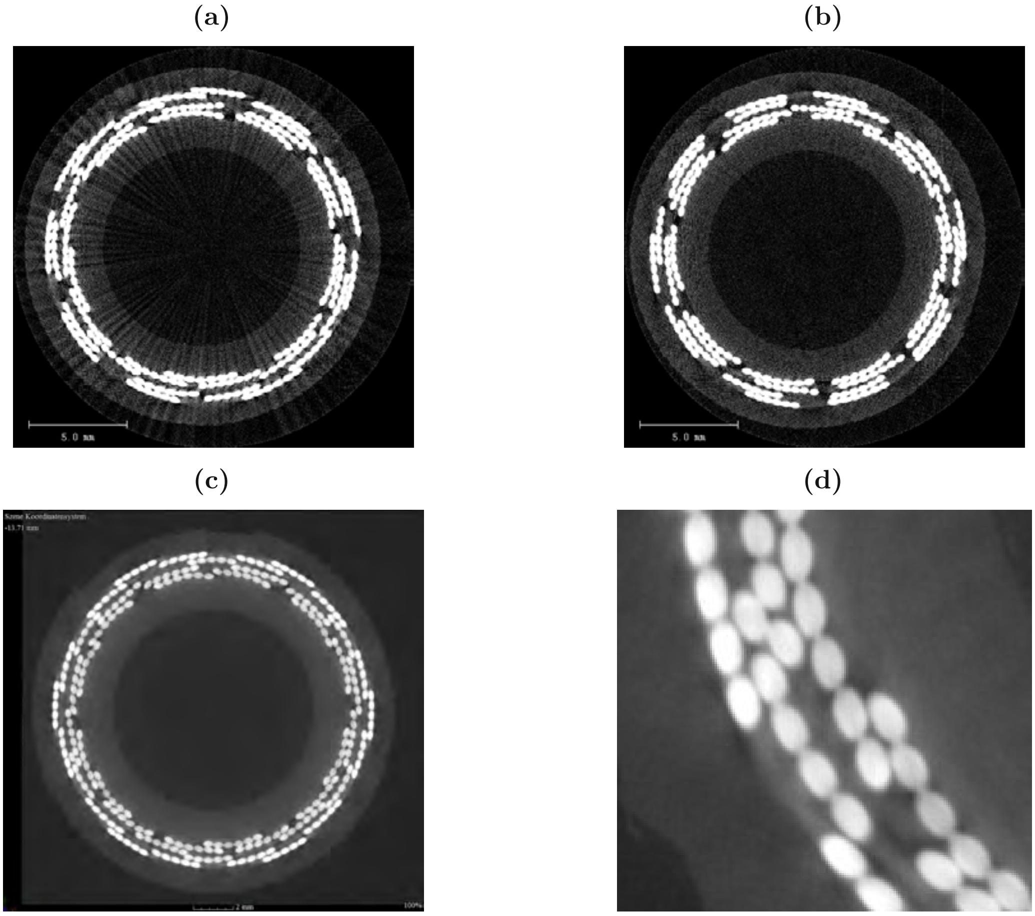


Fig. 4.15: CT measurments: (a) metal artifact, (b) reduced metal artifacts (chopper filter); industrial CT measurements: (c) image of a industrial CT measurement, (d) detected boarders of the rubber layer

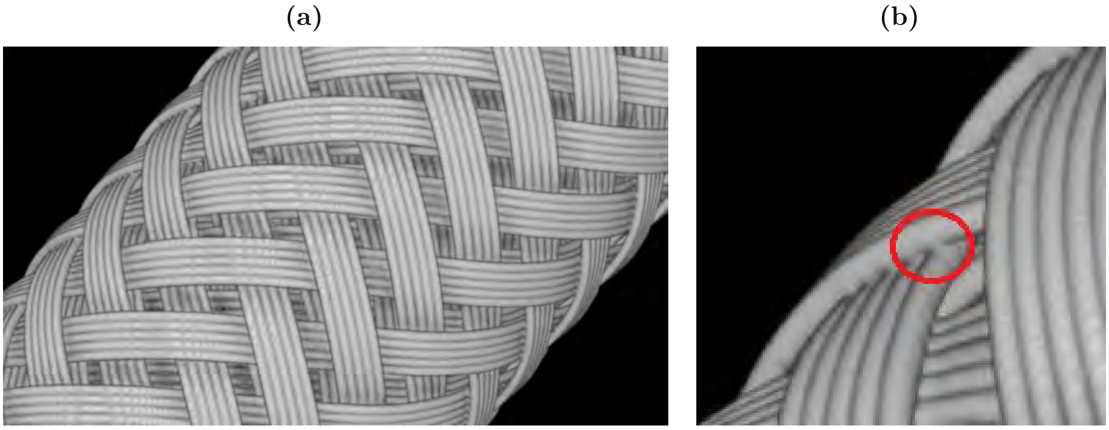


Fig. 4.16: CT Measurments: (a) 3D-model of the internal steel architecture, (b) intersection

4.2.2 Generation of voxel-based Finite Element models

Corresponding FE meshes were created on a voxel-by-voxel basis, see Fig. 4.17 for an exemplary example. The associated Python code is attached in Appendix A.3.

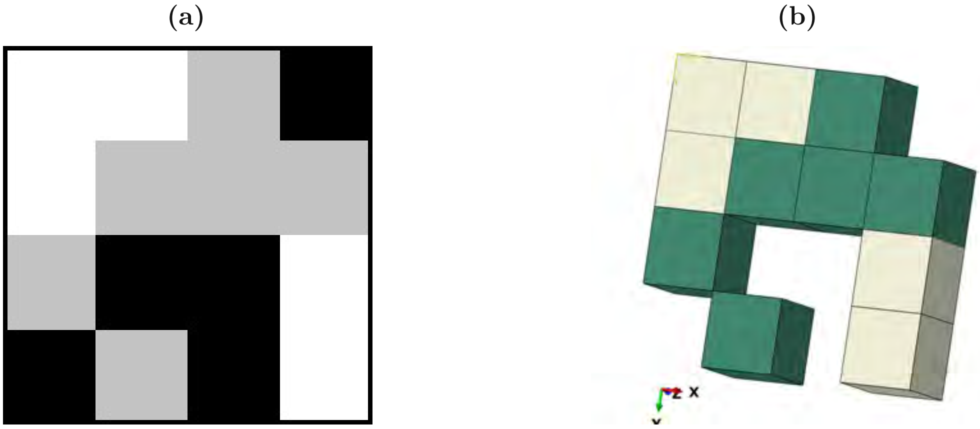


Fig. 4.17: Voxel-based mesh example: (a) example picture (gray value picture), (b) voxel-based mesh created from the gray-scale image; black corresponds to air, gray corresponds to rubber, white corresponds to steel

1. First, the image was converted into an array containing gray values, representing the density of the respective voxels – significantly differing grey values indicate different materials. In the present case, the materials of interest were air, rubber, and steel. E.g., the grey value array relating to the voxels depicted in Fig. 4.17(a) reads as:

$$\text{Grey Values (raw image)} \begin{bmatrix} 254 & 254 & 100 & 0 \\ 254 & 100 & 100 & 100 \\ 100 & 0 & 0 & 254 \\ 0 & 100 & 0 & 254 \end{bmatrix}$$

2. For processing reasons, the array was then extended by one row and one column, yielding the following array:

$$\text{Origin plane (extended)} \begin{bmatrix} 254 & 254 & 100 & 0 & 0 \\ 254 & 100 & 100 & 100 & 0 \\ 100 & 0 & 0 & 254 & 0 \\ 0 & 100 & 0 & 254 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad \text{Extended plane} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Then, each entry in the array was numbered, in order to be able to unambigiously process the information contained in each and every voxel:

$$\text{Origin plane (extended)} \begin{bmatrix} 1 & 2 & 3 & 4 & 0 \\ 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 \\ 15 & 16 & 17 & 18 & 19 \\ 0 & 20 & 21 & 22 & 23 \end{bmatrix} \quad \text{Extended plane} \begin{bmatrix} 24 & 25 & 26 & 27 & 0 \\ 28 & 29 & 30 & 31 & 32 \\ 33 & 34 & 35 & 36 & 37 \\ 38 & 39 & 40 & 41 & 42 \\ 0 & 43 & 44 & 45 & 46 \end{bmatrix}$$

4. Based on the numbered array, the respective entries (i.e., all nodes) were assigned to corresponding eight-node brick elements. E.g., the first element of the array shown in Fig. 4.17(a) consisted of nodes 1,2,6,5,24,25,29,28. The element size was determined based on the resolution of the CT image.

5. Finally, material properties were assigned to the voxels based on the grey values. In the present example, a grey value of 100 related to rubber, and grey value of 254 to steel, and a grey value of 0 to air.

Applying this procedure to the entire CT data eventually yields a corresponding FE model, see e.g. Fig. 4.17(b), for the above-explained example, and Fig. 4.18 for a the FE model represented created for the steel braiding in the hydraulic hoses.

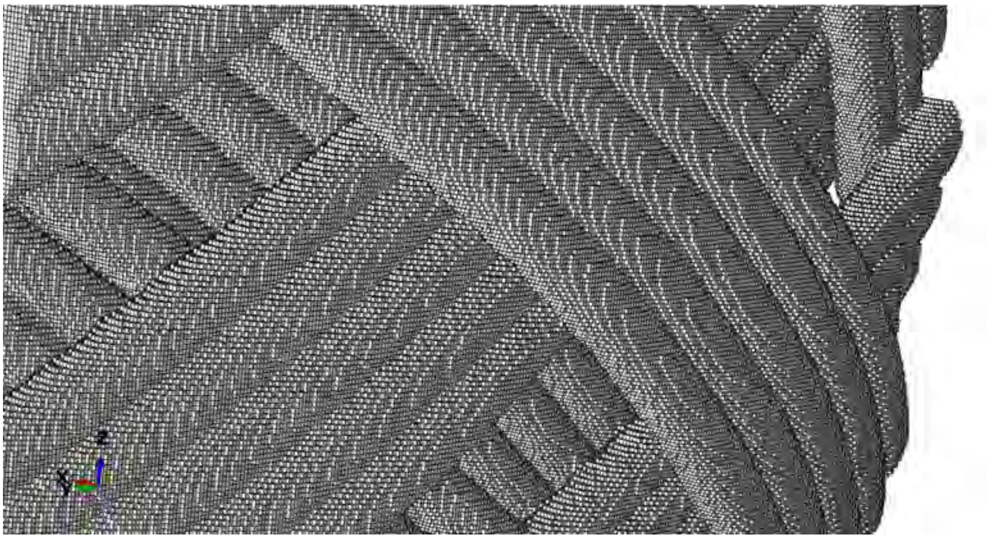


Fig. 4.18: CT data-based voxel representation of the steel braiding in double-reinforced hydraulic hoses

4.2.3 Extraction of microstructural features from micro-computed tomography data

Furthermore, a second strategy was implemented in the course of the presented work, aiming at utilizing the information provided by the CT data for generating highly realistic microscale Finite Element models. This concerns in particular the detection of the exact paths of the reinforcement steel wires, as well as of imperfections (such as deviations from the originally designed arrangement of the wires, or voids in the rubber). In the literature, this strategy is sometimes referred to as “reversed engineering“. An in-house subroutine, developed and implemented by means of Python, was used to extract this information from the CT data, for the further use for the model building in Abaqus (version 2018). To that end, also the image-editing software ImageJ was used, for preparing the raw images.

First, the images relating to the raw data were adjusted in terms of the brightness, in such way that only the steel wires were displayed, compare Fig. 4.19(a) and (b). Then, the images were binarized, hence, depending on the actual grey level of a specific voxel, this grey value was changed to either black (relating to air or rubber), or white (relating to steel), see Fig. 4.19(c). ImageJ furthermore offered the option to separate the wires from each other, and to get rid of unwanted intersections, by means of the program-internal function “watershed“, see Fig. 4.19(d).

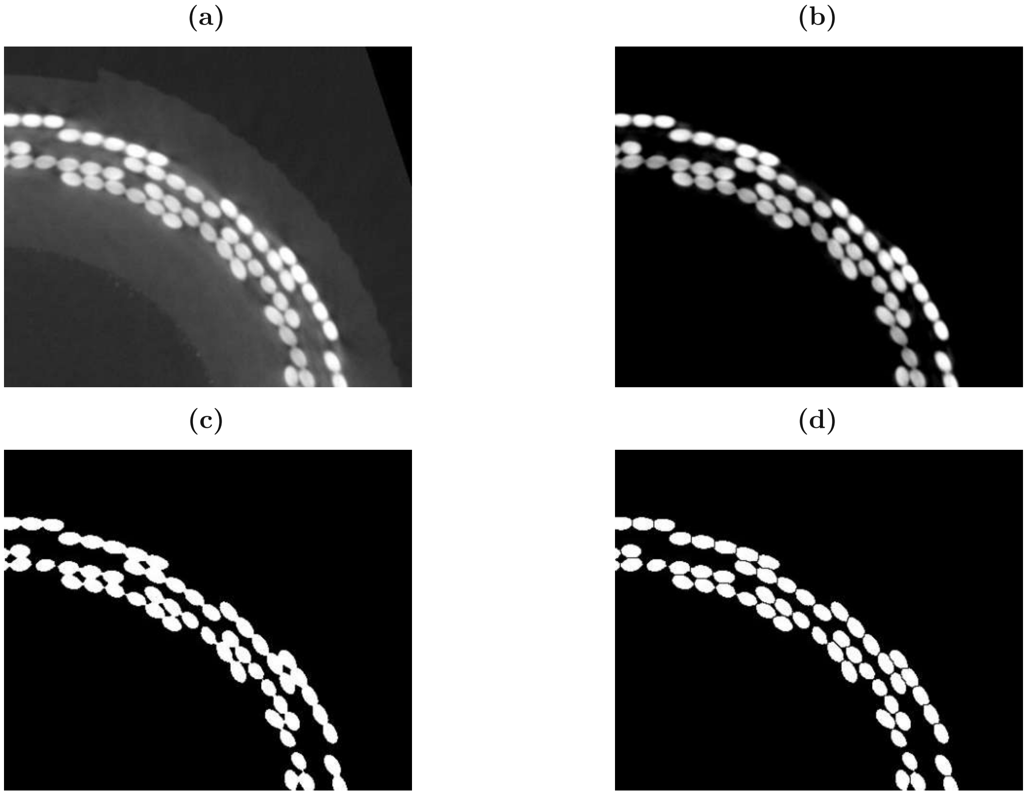


Fig. 4.19: Sequence of image processing steps realized by means of the software ImageJ, starting from (a) raw image data, over (b) images with adjusted brightness, (c) binarized images, eventually leading to (d) separated wires while intersections have been eradicated

Then, the exact path of each wire had to be detected, through the whole stack of CT slices. For that purpose, a reference point was chosen, right in the center of the hose. This required again some manipulation of the raw CT data. Namely, through brightness adjustment, the inner part of the hose was emphasised allowing to find the center of the resulting circle, see Fig. 4.20(a). Following this midpoint from slice to slice gave access to the curvature of the hose, see Fig. 4.20(b).

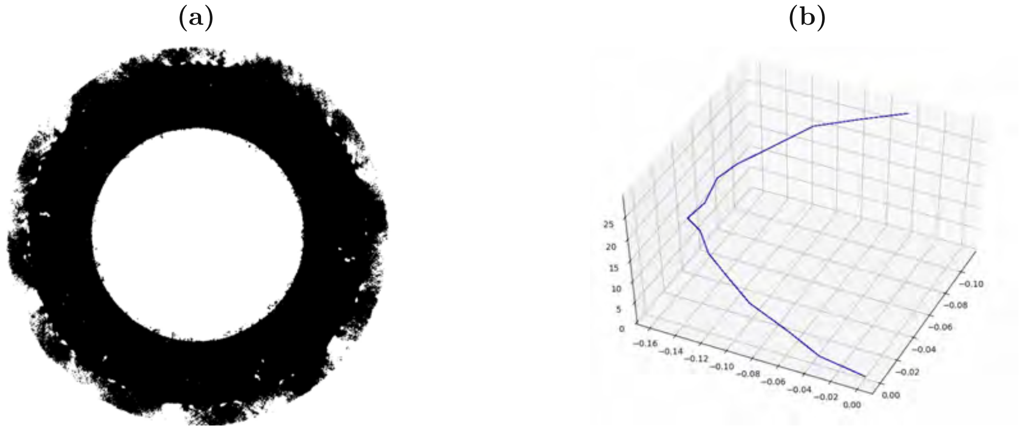


Fig. 4.20: Determination of the curvature of the scanned hoses, based on (a) determination of the hose midpoint, leading to (b) the spatial trace of this midpoint indicating the corresponding hose curvature

For tracing the single wires, each wire in a specific CT slice was detected, and the coordinates of their midpoints were written into an array. Repeating this procedure slice by slice, and considering that the resolution of CT scans is much smaller than the diameter of a single wire, allows for following the position of all wire midpoints through the stack of CT slices unambiguously, see Figure 4.21. This information, together with the previously determined hose curvature, allows for creation of a corresponding CAD-assembly which was used for the further Finite Element representation of the steel braiding, see Fig. 4.22, based on a so-called solid works-macro (Abaqus, version 2018).

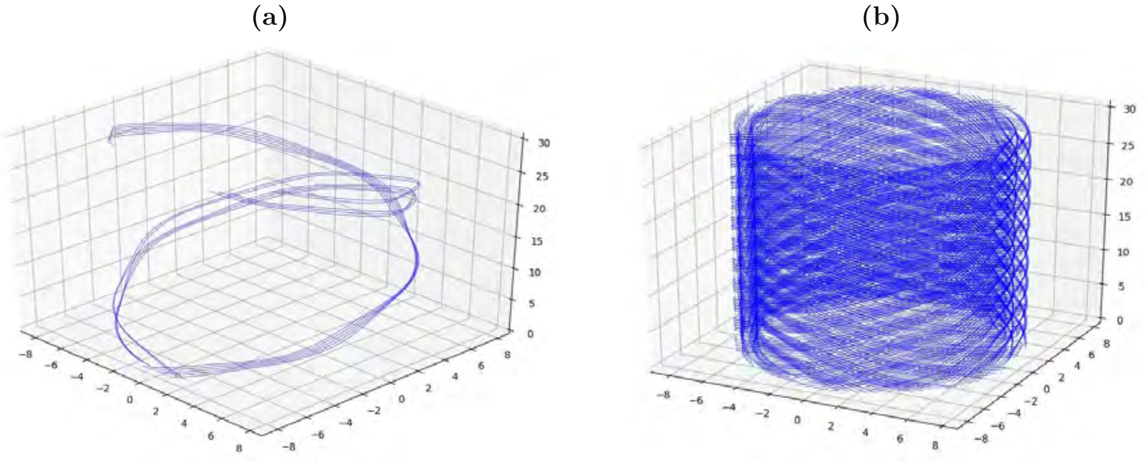


Fig. 4.21: Tracing the steel wires through the stack of CT slices, showing (a) a few wires, and (b) all wires included in the scanned specimen

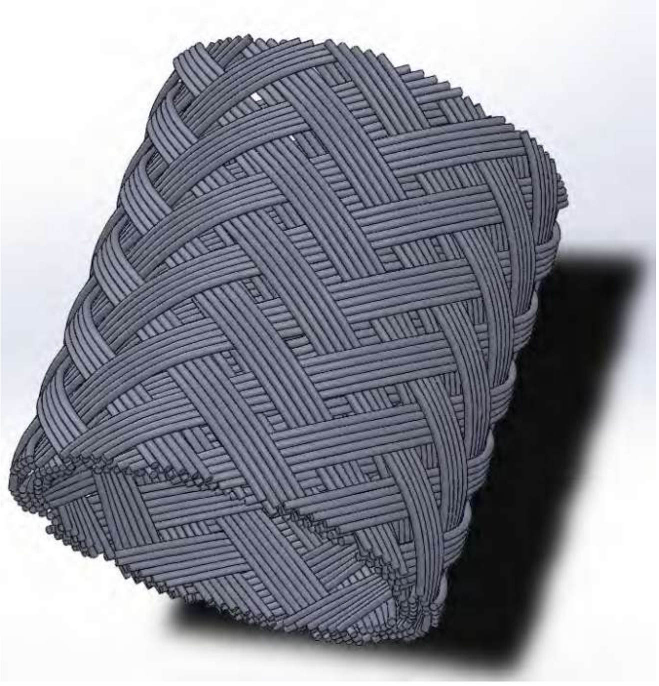


Fig. 4.22: CAD-assembly of the steel braiding

Another aim of the presently described processing of CT data was detection of voids. Therefore, first the CT image was binarized with the previously used program imageJ so that only the air (black) and the hose material (gray), including the reinforcing material, was recognizable, see Fig. 4.23 (a). With the previously created script, which is based on the voxel-by-voxel method, a voxel-based mesh was rebuilt by means of the edited image stack, with the difference to that before that the voids are meshed as negatives, which are later subtracted from the host matrix. The created void negatives consist of an orphan mesh, see Fig. 4.23(b). In order to convert the orphan mesh into geometries, a free Abaqus plug-in was used (“3D mesh to geometry“). The result of this script, see Appendix A.5, are negatives of the detected voids, which can be subtracted from the host matrix, see Fig. 4.23(c).

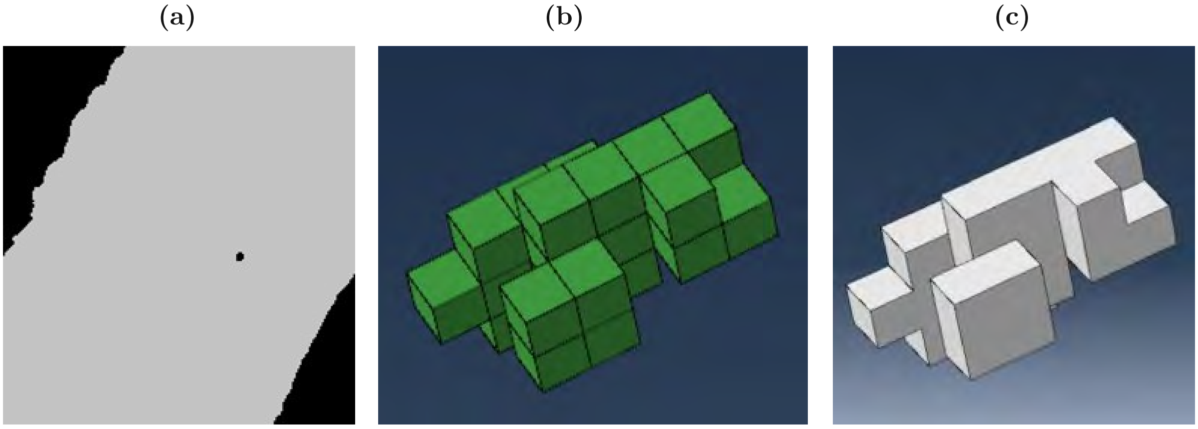


Fig. 4.23: Void editing: (a) binarized CT image, (b) void negative (orphan mesh), (c) void negative converted with the abaqus plugin “3D mesh to geometry“

Finally, a microscale model of the scanned hose could be created, using the information concerning wire and void detection. For that purpose, a rubber matrix layer was created manually which allows to insert afterwards the reconstructed steel braiding, see Fig. 4.27(a). Also, the matrix contained the detected voids. Via subtraction of the aformentioned steel braiding assembly and the generated void negatives, see Fig. 4.27(b) to 4.27(d), the final FE model could be generated , see Fig. 4.27(e) and (f), using standard tetrahedral elements. On the interface between steel and rubber, any movement between neighbouring nodes was locked, by means of a tie constraint condition.

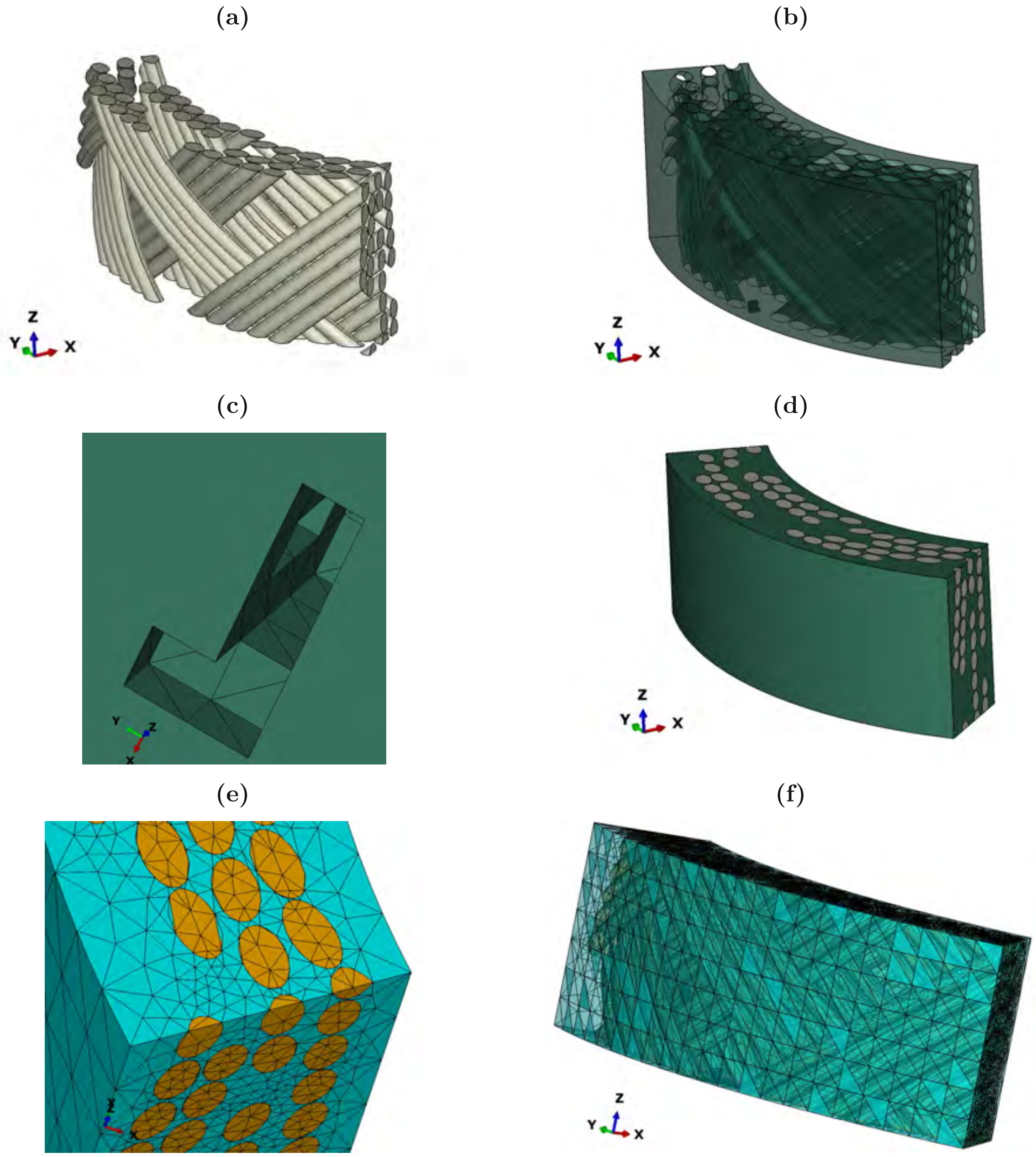
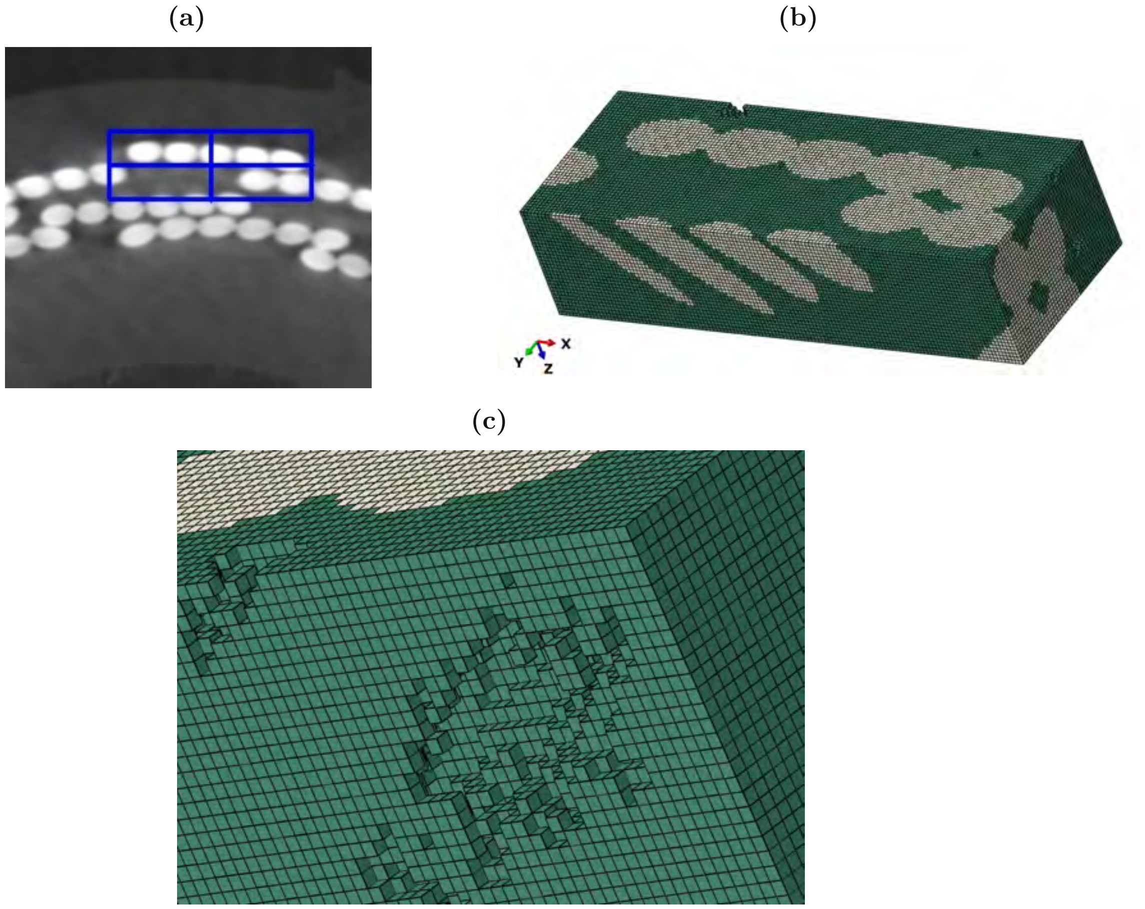


Fig. 4.24: Mesh generation of a representative part of the scanned specimen: (a) rebuild wires, (b) subtracting wires from the matrix, (c) subtracted voids, (d) assembly, (e) meshed assembly, (f) meshed assembly

4.2.4 Exemplary microscale computations

First, a subregion of the scanned domain was selected, see Figure 4.25(a) to (c), containing rubber, steel wires, and air voids. From the corresponding macroscopic model, see Section 4.1.3.2 and in particular Figure 4.10, boundary conditions for submodeling were imposed onto the outer surfaces of the microscopic model. The resulting displacements and von Mises stresses are depicted in Figure 4.26(a) and (b). As expected, the irregular (serrated) contact surfaces between steel and rubber cause local stress peaks, see Figure 4.26, which are probably unrealistic and somewhat adulterate the results. A remedy could be the use of a smoothing routine in the affected regions. Furthermore, calculations were complicated by the non-uniform stress distributions resulting from the macroscopic (beam) model, on the boundaries of the microscopic model. A reasonable simplification could be some smearing leading to quasi-uniform boundary conditions. And, importantly, for the sake of being computationally manageable on standard desktop computers, only a very small microscopic model could be used, consisting of 176416 finite elements. Increasing the size of the studied domain would increase the significance of the results, but also lead to dramatically increasing computational costs.


(a)

(b)

(c)

Fig. 4.25: Voxel-based-method: (a) desired area to compute, (b) meshed micro model, (c) voids of the matrix which are automatically taken into account

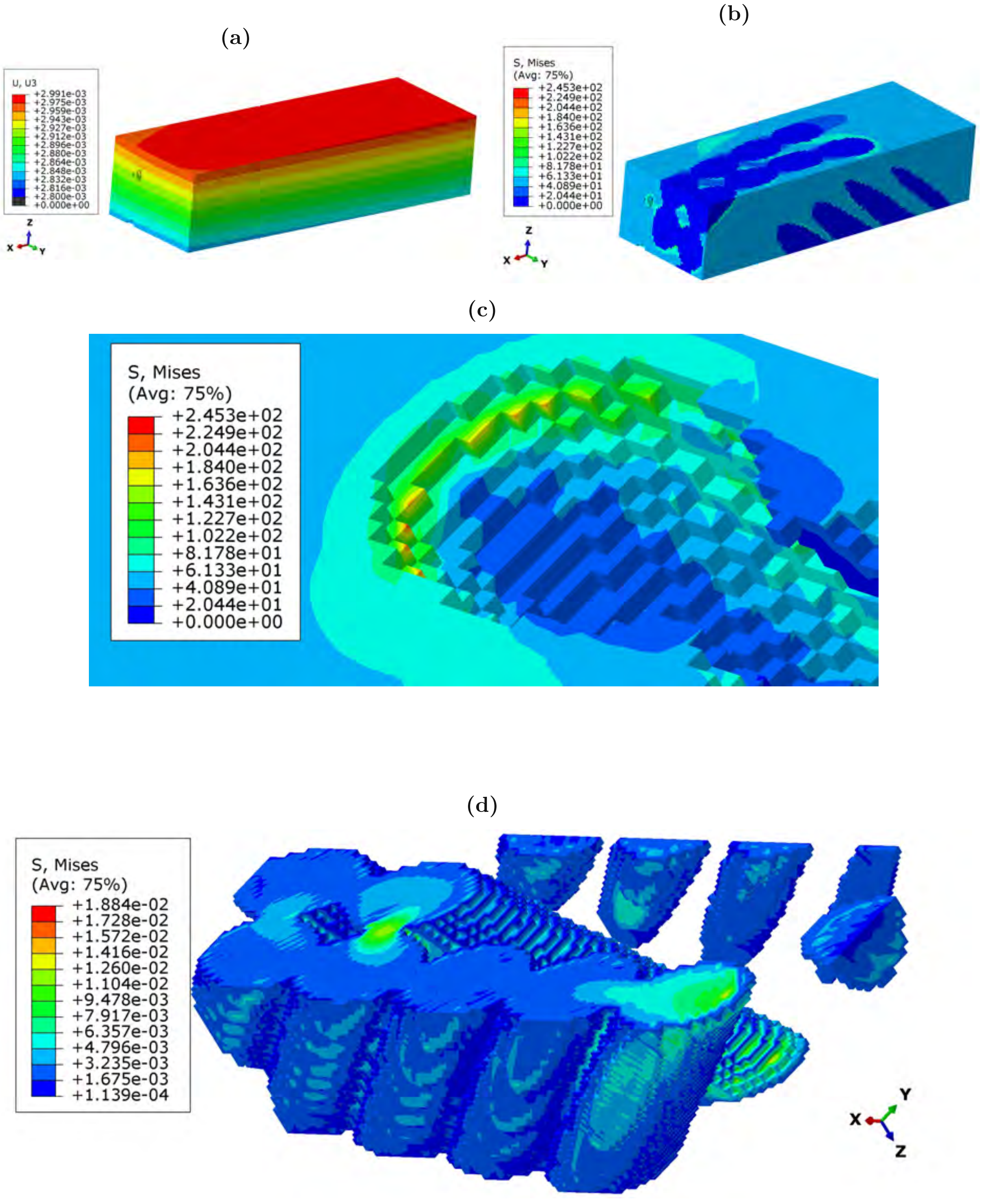
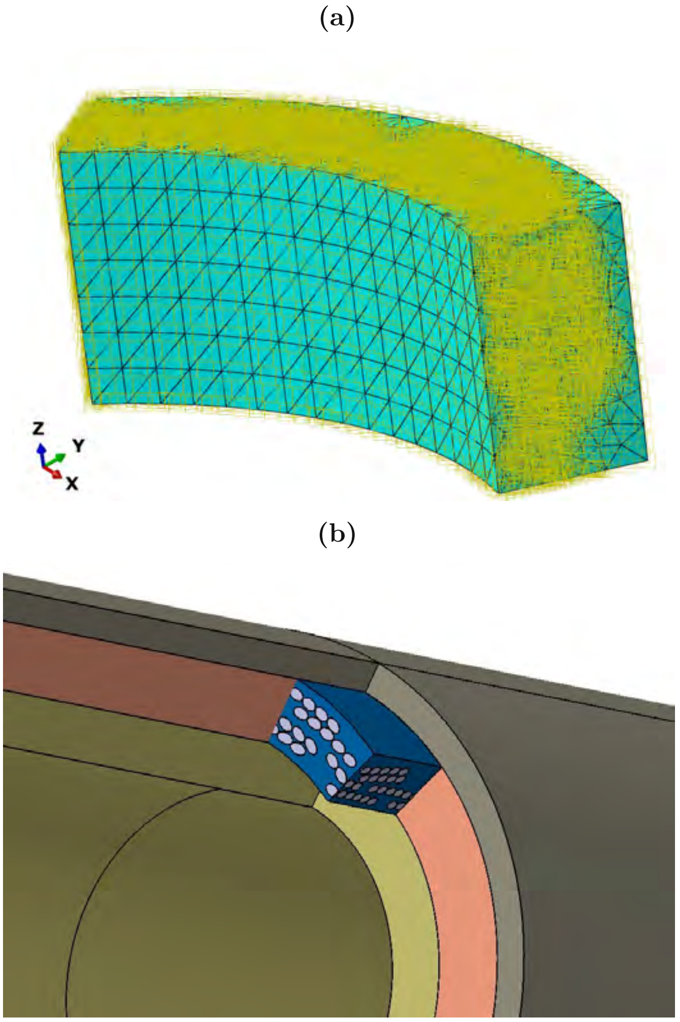


Fig. 4.26: Results from a FE calculation based on a voxel-by-voxel reconstruction of a representative microscopic domain, illustrated in terms of (a) displacements in Z-direction, and (b) von Mises stresses, (c) stress increasings at the serrated interface between rubber and steel, (d) von Mises stresses in the steel

Finally, a, compared to the above presented voxel approach, computationally more efficient approach based on standard tetrahedral elements was implemented, based on wire and void detection, as explained in Section 4.2.3, see Fig. 4.21 and 4.23. The simulated load case related to the tensile tests described in Section 4.1.3.2, see also Fig. 4.10. Hence, the displacements computed by means of this macroscopic model were used on the submodeling boundary conditions for the microscopic model depicted in Figure 4.27(a). The micro model was placed in the middle of the length of the hose section in the Z-direction and additionally exactly in the matrix layer, see 4.27(b).


(a)

(b)

Fig. 4.27: (a) Submodeling boundary conditions and (b) placement of the micro model in the macro model

The resulting displacements and von Mises stresses are depicted in Figs. 4.28(a), (c), and (e), while the results within the same area computed by means of the macroscopic model are depicted in Figs. 4.28(b), (d), and (f). In the visualizations for the displacements, it can be seen that at the edge nodes or submodeling boundary conditions, the respective displacements that occur at the macro level also occur exactly at this point on the micro level. Furthermore, the differences, in terms of wire position, between macro- and microscopic model, and consequently potentially not quite correctly assigned boundary conditions lead to the differences between the results of micro- and macroscopic computations. This issues should be addressed before drawing definite conclusions from the computations.

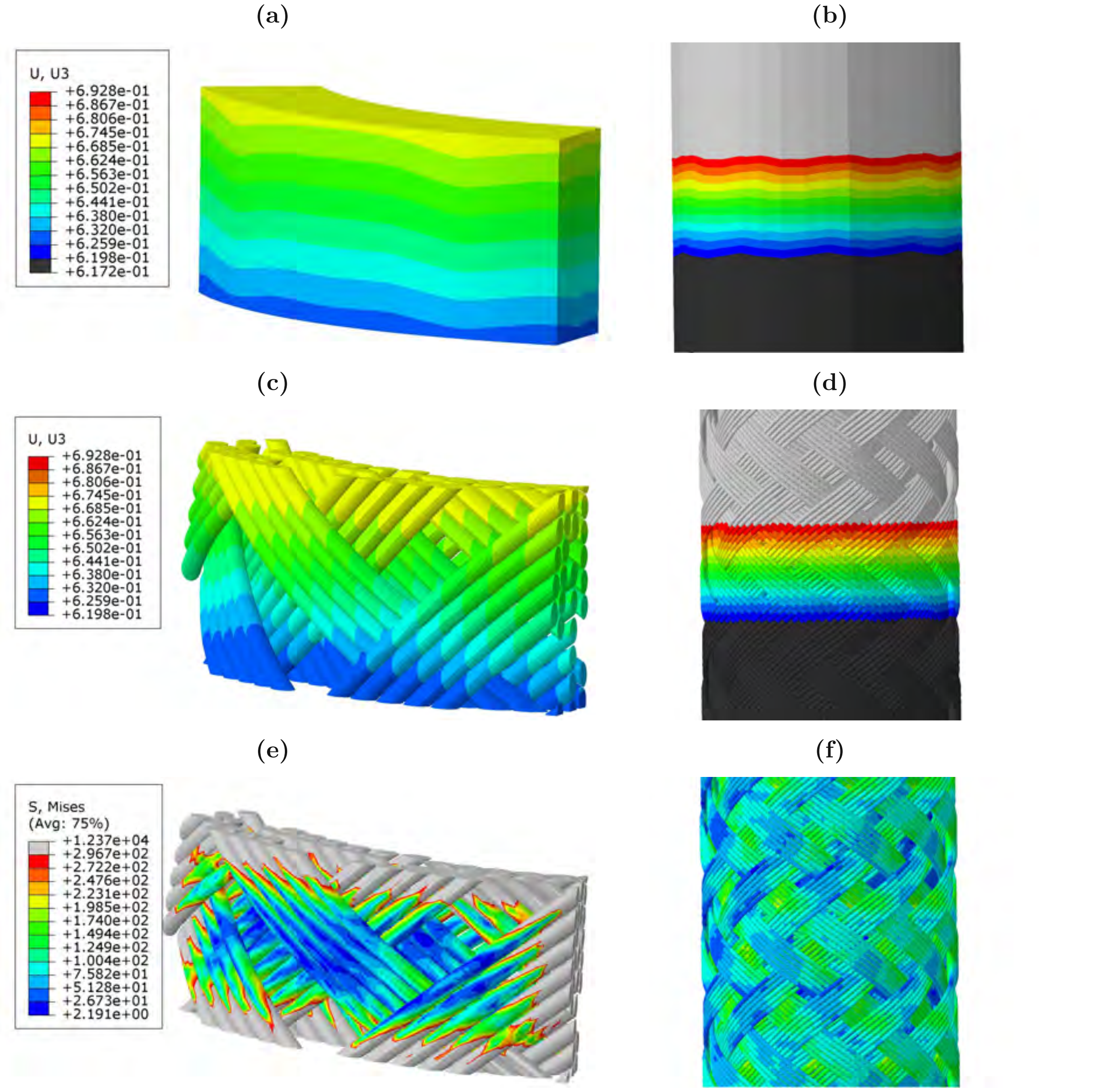


Fig. 4.28: Visualizations for the computations on the micro level and macro level: (a) Z-displacement (micro model), (b) Z-displacement (macro model), (c) Z-displacement steel (micro model), (d) Z-displacement steel (macro model), (e) von Mises stresses steel (micro model), (f) von Mises stresses steel (macro model)

4.3 Discussion

Considering the results presented in previous Sections 4.1 and 4.2, the following conclusions can be drawn, discussed in terms of the different types of computational analysis tested in this thesis:

- Beam-element based macroscopic models are actually capable of reproducing tensile and three-point bending tests. Also from a qualitative point of view, the results are indeed reasonable. However, it must be stressed that the underlying material parameters had to be fitted based on the experimental data, which gives rise to the suspicion that the such calibrated material models may not work for other compositions or other materials. Furthermore, the used beam elements turned out to be too stiff, as compared to corresponding solid element representations of the reinforcement wires. This was taken into account by reduction of the beam diameter such that the increased stiffness was compensated. Clearly, this approach must be subjected to utmost scrutiny, and the general validity of such reduction is probably questionable.
- The results for rebar layer-based macroscopic models showed that such model representations cannot yield the required stiffness increase, as compared to hoses consisting of rubber only. Hence, it seems like rebar layer-based models are not suitable for the composite structure under investigation at all.
- As for the voxel-based reconstruction of the microstructure, this method is applicable without any conceptual restriction. Clearly, adequate results require an extremely high resolution of the underlying image data, in order to take into account the curved shapes that are encountered throughout the braiding structure. At lower resolutions, unwanted stress peaks follow. On the other hand, high-resolution micro-Finite Element models are, as precise as they may be, computationally very demanding. Even for resolution considered in this work, only a small part of the image data could be considered for computational simulations, as otherwise huge computer clusters would have been necessary, in order to provide the needed hardware requirements.
- The reconstruction of microstructural features from CT data was actually successful, both in terms of void detection and in terms of considering the actual braiding structure. However, using this information for computing microstructural stress and strain distributions relating consistently to the macroscopic computations turned out be not straightforward. On the one hand, little geometric differences, e.g. in terms of steel wire positions, between micro- and macrostructural model configurations require some manual manipulations. On the other hand, strictly speaking, an iterative approach would have to be applied, since micro- and macrostructural stress and strain distributions are per se not consistent for the macroscopic representation an idealized wire geometry is used, whereas the microscopic representation complies with the real situation which deviates from the idealized one.

Chapter 5

Conclusions and outlook

This thesis comprises both experimental and computational studies performed on steel-reinforced rubber hoses produced for high pressure applications. A detailed discussion of the obtained results can be found in Sections 3.3 and 4.3, and is not reiterated here. In the following, the main conclusions related to the experimental campaign and to the computational analyses are presented. And, finally, reasonable future research directions tying in with the methods and results presented in this thesis are suggested.

Starting with the experimental results, it can be concluded that all experimental methods could be implemented as planned. However, it turned out that the used hydraulic unit was not powerful enough to induce consistent results – the maximum pressure applied in the experiments was significantly lower than the maximum pressure the hoses are subjected to during targeted operation. It is believed that increasing the magnitude of pressurization would have to induce consistent results, since then the steel braiding would be activated to the same extent in all samples. Apart from that, the tensile tests and the three-point bending tests yielded consistent results which complied more or less with the expectations.

The computational analyses somewhat corroborated the initial apprehension as to the immense difficulty of such endeavor. It has turned out that modeling such hoses on the macroscopic scale based on considering the steel braiding as beam elements allows for reasonably exact simulation of tensile and three-point bending tests. However, for some reason, these beam elements are too stiff in bending mode, as compared to using solid element representations of steel. This was counteracted by reducing the diameter of the beam elements for simulating the three-point bending tests. While this strategy led to a good agreement between model predictions and experimental results, it is certainly questionable. Representing the steel reinforcement based on the rebar strategy led to much too compliant behavior of the simulated hoses, both for tensile and three-point bending loading. On the microscopic scale, the voxel-based modeling strategy was feasible from a conceptual point of view. However, due to the enormous computational demands for a large high-resolution micro-Finite Element model, only a small region could be simulated, yielding probably not representative results. Furthermore, despite the already substantial computational costs of the voxel-based model presented in this thesis, the resolution was probably still too low, as stress concentrations could be observed which were most likely caused by not considering the curved shapes of the steel wires exactly enough. Utilizing CT data for detection of voids and the actual steel wire arrangement, on the other hand, worked nicely.

However, the common disadvantage of all microscopic models was the geometric inconsistency between the idealized geometry of the steel wires assumed on the macroscopic scale, and the actual one on the microscopic scale. Strictly speaking, this problem would have to be addressed based on an iterative approach, where the macroscopic model is adapted to the real geometry in step-wise fashion. Clearly, such approach would lead to a substantial increase of computational costs.

Finally, the above arguments are considered as basis for future research tying in with the here-presented project. From an experimental point of view, the next steps are quite clear. Equipping the universal testing machine with a more powerful hydraulic unit and re-performing (and probably extending) the combined tensile/pressurization loading promises to yield interesting new insights. From a theoretical point of view, it is suggested to revisit the modeling strategy from scratch. In fact, it is believed that instead of the here-pursued top-down strategy, development of a rigorous bottom-up approach would be advisable. This way, the main weaknesses of the current methods could be probably eradicated, including the so far missing (i) reliable, and physically well-substantiated material models for rubber, (ii) suitable interface conditions between rubber and steel which take into account de- and rebonding mechanisms upon loading and unloading, and (iii) homogenization techniques capturing then highly complex interactions between the (hard) steel wires which are arranged in a braided structure and the (soft) rubber matrix.

While there is no doubt that mastering the latter three challenges is very challenging, the benefits would be groundbreaking, allowing, for the first time, for reliable and computationally efficient computer simulations as to the deformation and failure mechanisms of hydraulic hoses.

Bibliography

[1] Semperflex Hydraulics. *Hydraulic Hose Catalogue 2017*. Ed. by Semperflex Hydraulics. May 2017. URL: `https://tinyurl.com/Hydraulikschlauchkatalog-2017` (Zugriff am 12/13/2018).

[2] *DIN EN ISO 6803:2017 07: Rubber or plastics hoses and hose assemblies – Hydraulic-pressure impulse test without flexing (ISO 6803:2017);* Berlin: DIN Deutsches Institut für Normung e. V., July 2017.

[3] *DIN EN ISO 7751:2016: Rubber and plastics hoses and hose assemblies – Ratios of proof and burst pressure to maximum working pressure (ISO 7751:2016)*. Berlin: DIN Deutsches Institut für Normung e. V., Nov. 2016.

[4] *DIN EN ISO 1402:2009 10: Rubber and plastics hoses and hose assemblies – Hydrostatic testing (ISO 1402:2009);* Berlin: DIN Deutsches Institut für Normung e. V., Oct. 2009.

[5] M. Königsberger, M. Hlobil, B. Delsaute, S. Staquet, C. Hellmich, and B. Pichler. "Hydrate failure in ITZ governs concrete strength: A micro-to-macro validated engineering mechanics model". In: *Cement and Concrete Research* 103 (Nov. 2017). DOI: `10.1016/j.cemconres.2017.10.002`.

[6] C. Dunant and K. Scrivener. "Micro-mechanical modelling of alkali–silica-reaction-induced degradation using the AMIE framework". In: *Cement and Concrete Research* 40 (Apr. 2010), pp. 517–525. DOI: `10.1016/j.cemconres.2009.07.024`.

[7] M.-I. Pastrama, S. Scheiner, P. Pivonka, and C. Hellmich. "A mathematical multiscale model of bone remodeling, accounting for pore space-specific mechanosensation". In: *Bone* 107 (Nov. 2017). DOI: `10.1016/j.bone.2017.11.009`.

[8] A. Fritsch, C. Hellmich, and L. Dormieux. "Ductile sliding between mineral crystals followed by rupture of collagen crosslinks: Experimentally supported micromechanical explanation of bone strength". In: *Journal of Theoretical Biology* 260 (July 2009), pp. 230–52. DOI: `10.1016/j.jtbi.2009.05.021`.

[9] S. D. Badilatti, P. Christen, I. Parkinson, and R. Müller. "Load-adaptive bone remodeling simulations reveal osteoporotic microstructural and mechanical changes in whole human vertebrae". In: *Journal of Biomechanics* 49 (Oct. 2016). DOI: `10.1016/j.jbiomech.2016.10.002`.

[10] A. Julián Arias Moreno, K. Ito, and B. van Rietbergen. "Micro-Finite Element analysis will overestimate the compressive stiffness of fractured cancellous bone". In: *Journal of Biomechanics* 49 (May 2016). DOI: `10.1016/j.jbiomech.2016.05.021`.

[11] M. Li, J. Füssl, M. Lukacevic, J. Eberhardsteiner, and C. Martin. “Strength predictions of clear wood at multiple scales using numerical limit analysis approaches”. In: *Computers and Structures* 196 (Dec. 2017). DOI: `10.1016/j.compstruc.2017.11.005`.

[12] K. De Borst, C. Hellmich, and J. Eberhardsteiner. “Development and experimental validation of a continuum micromechanics model for the elasticity of wood”. In: *European Journal of Mechanics A-Solids* 24 (Nov. 2005), pp. 1030–1053. DOI: `10.1016/j.euromechsol.2005.05.006`.

[13] J. Cho, Y. Jee, W. Kim, S. Han, and S. Lee. “Homogenization of braided fabric composite for reliable large deformation analysis of reinforced rubber hose”. In: *Composites Part B: Engineering* 53 (Oct. 2013), pp. 112–120. DOI: `10.1016/j.compositesb.2013.04.045`.

[14] G. Drumond, I. P Pasqualino, and M. Costa. “IBP121815 Study of an alternitive material for the manufacture of subsea umbilical hydraulic hoses”. In: Sept. 2015.

[15] J. Cho, Y. Yoon, C. Seo, and Y. Kim. “Fatigue life assessment of fabric braided composite rubber hose in complicated large deformation cyclic motion”. In: *Finite Elements in Analysis and Design* 100 (Aug. 2015). DOI: `10.1016/j.finel.2015.03.002`.

[16] Deutsche Gesetzliche Unvallversicherung e.V. (DGUV). *Sachgebiet Maschinen, Robotik und Fertigungsautomation; Hydraulische Prüfstände; FB HM-086; Ausgabe 08/2018*. URL: `https://www.dguv.de/medien/fb-holzundmetall/publikationen/infoblaetter/infobl_deutsch/086_hydraulik_pruef.pdf` (Zugriff am 12/13/2018).

[17] *DIN EN ISO 4671:2012 02: Rubber and plastics hoses and hose assemblies - Methods of measurement of the dimensions of hoses and the lengths of hose assemblies (ISO 4671:2007 + Amd 1:2011);* Berlin: DIN Deutsches Institut für Normung e. V., Feb. 2012.

[18] *DIN EN 24671:1993 04: Rubber and plastics hoses and hose assemblies; methods of measurement of dimensions (ISO 4671:1984);* Berlin: DIN Deutsches Institut für Normung e. V., Apr. 1993.

[19] *DIN EN 24671:1993 04: Rubber and plastics hoses; sub-ambiant temperature flexibility tests (ISO 4672:1988);* Berlin: DIN Deutsches Institut für Normung e. V., Apr. 1993.

[20] *DIN EN 28033:1993 07: Rubber and plastic hose; determination of adhesion between components (ISO 8033:1991);* Berlin: DIN Deutsches Institut für Normung e. V., July 1993.

[21] *DIN EN ISO 7233:2017 03: Rubber and plastics hoses and hose assemblies - Determination of resistance to vacuum (ISO 7233:2016);* Berlin: DIN Deutsches Institut für Normung e. V., Mar. 2017.

[22] *DIN ISO 1817:2016 11: Rubber, vulcanized or thermoplastic - Determination of the effect of liquids (ISO 1817:2015)*. Berlin: DIN Deutsches Institut für Normung e. V., Nov. 2016.

[23] *DIN EN ISO 27326:1993 07: Rubber and plastics hoses; assessment of ozone resistance under static conditions (ISO 7326:1991);* Berlin: DIN Deutsches Institut für Normung e. V., July 1993.

[24] G. Szabó, K. Varadi, and D. Felhős. “Finite element model of a filament-wound composite tube subjected to uniaxial tension”. In: *Modern Mechanical Engineering* 07 (Jan. 2017), pp. 91–112. DOI: 10.4236/mme.2017.74007.

[25] Veryst Engineering. *Reinforced Hose Analysis [online]*. Needham: Veryst Engineering, LLC. URL: https://www.veryst.com/Papers/Reinforced_Hose_Analysis.pdf (Zugriff am 12/13/2018).

[26] C. Groth. *Submodelltechnik*. Dec. 2015. URL: http://www.cae-wiki.info/wikiplus/index.php/Submodelltechnik (Zugriff am 12/13/2018).

[27] Ansys. *Understanding Submodeling*. Apr. 2016. URL: https://www.sharcnet.ca/Software/Ansys/17.0/en-us/help/ans_adv/Hlp_G_ADV4UN.html (Zugriff am 12/13/2018).

[28] W.-G. Jiang. “Implementation of domain superposition technique for the nonlinear analysis of composite materials”. In: *Journal of Composite Materials* 47 (Jan. 2013), pp. 243–249. DOI: 10.1177/0021998312439220. (Zugriff am 12/13/2018).

[29] *DIN EN 853:2011 11: Rubber hoses and hose assemblies - Wire braid reinforced hydraulic type - Specification;* Berlin: DIN Deutsches Institut für Normung e. V., Nov. 2011.

[30] F. S. Fritz Röthemeyer. *Kautschuktechnologie - Werkstoffe - Verarbeitung - Produkte*. M: Carl Hanser Verlag GmbH Co KG, 2013. ISBN: 978-3-446-43760-9.

[31] K. J. Schmidt. *Kompendium Schlauchtechnik*. Essen: Vulkan-Verlag GmbH, 1999. ISBN: 978-3-802-72715-3.

[32] E. Britannica. *Nitrile rubber; Encyclopædia Britannica, inc. ; Ausgabe 04/2016*. URL: https://www.britannica.com/technology/nitrile-rubber (Zugriff am 12/21/2018).

[33] R. Kreiselmaier. *Definition von Mullins und Payne Effekt*. May 21, 2014. URL: https://www.kgk-rubberpoint.de/9465/wie-elastisch-ist-gummi/# (Zugriff am 12/13/2018).

[34] J. Diani, B. Fayolle, and P. Gilormini. “A review on the Mullins effect”. In: *European Polymer Journal* 45.3 (2009), pp. 601–612. ISSN: 0014-3057. DOI: https://doi.org/10.1016/j.eurpolymj.2008.11.017. URL: http://www.sciencedirect.com/science/article/pii/S0014305708006332.

[35] *ISO 48:2010 09: Rubber, vulcanized or thermoplastic - Determination of hardness (hardness between 10 IRHD and 100 IRHD);* Genf: Internationale Organisation für Normung, Sept. 2010.

[36] *ISO 37:2011 12: Rubber, vulcanized or thermoplastic - Determination of tensile stress-strain properties;* Genf: Internationale Organisation für Normung, Dec. 2011.

[37] F. Abraham, T. Alshuth, and S. Jerrams. “Dependence on mean stress and stress amplitude of fatigue life of EPDM elastomers”. In: *Plastics, Rubber and Composites* 30 (Sept. 2001), pp. 421–425. DOI: 10.1179/146580101101541822.

[38] *DIN ISO 23529:2012 10: Rubber – General procedures for preparing and conditioning test pieces for physical test methods (ISO 23529:2010);* Berlin: DIN Deutsches Institut für Normung e. V., Oct. 2012.

[39] *ISO/TS 17165-2:2018 04: Hydraulic fluid power - Hose assemblies - Part 2: Practices for hydraulic hose assemblies*. Genf: Internationale Organisation für Normung, Apr. 2018.

[40] National Instruments. *LabVIEW*. Version 13.0f2. Apr. 10, 2013. URL: `http://www.ni.com`.

[41] L. Scalise, B. Lonzi, and N. Bernacchia. "A camera-based experimental method for mechanical test on patellar tendons". In: *Conference Proceedings of the Society for Experimental Mechanics Series* 63 (July 2015), pp. 7–17. DOI: `10.1007/978-3-319-06974-6_2`.

[42] The MathWorks, Inc. *Matlab R2017b*. Version 9.3.0.713579. Feb. 19, 2017. URL: `https://www.mathworks.com`.

[43] O. A. Bauchau and J. I. Craig. "Euler-Bernoulli beam theory". In: *Structural Analysis*. Ed. by O. A. Bauchau and J. I. Craig. Dordrecht: Springer Netherlands, 2009, pp. 173–221. ISBN: 978-90-481-2516-6. DOI: `10.1007/978-90-481-2516-6_5`. URL: `https://doi.org/10.1007/978-90-481-2516-6_5`.

[44] M. Shahzad, A. Kamran, M. Z. Siddiqui, and M. Farhan. "Mechanical Characterization and FE Modelling of a Hyperelastic Material". In: *Materials Research* 18 (Oct. 2015). DOI: `10.1590/1516-1439.320414`.

[45] X. Yang, M. Behroozi, and O. Olatunbosun. "A neural network approach to predicting car tyre micro-scale and macro-scale behaviour". In: *Journal of Intelligent Learning Systems and Applications* 6 (Jan. 2014), pp. 11–20. DOI: `10.4236/jilsa.2014.61002`.

[46] A. 6. O. Documentation. *Local rebar orientation system*. Abaqus 6.10. Dassault Systèmes. Vélizy-Villacoublay, France, 2010. DOI: `https://www.sharcnet.ca/Software/Abaqus610/Documentation/docs/v6.10/books/usb/default.htm?startat=pt01ch02s02aus12.html`.

Appendix A

Appendix

A.1 Image correlation

The subsequently presented Matlab code was developed to determine local strains on the hose surface in the vertical and horizontal direction via white reference marks which were placed on the hose surface. Another task of the code was to synchronize the obtained strain curves with the force and pressure curves.

Program Code A.1: Image correlation (177 Lines of Matlab)

```
% This code deals with the evaluation of the local strains on the hose
% surface and the following linkage with the force and pressure signals;
% Before this program can be used however, the size of the reference
% marks must be determined by measuring the image; This can easily be done
% using a conventional image processing program.

% Number of pictures of the respective camera
Anzahl_Bilder=7719;

% Outer hose diameter (in mm)
Schlauchdurchmesser=17;

% Size of the marker (number of pixels in side length)
g_M=50;

% Shade_Faktor (0-255)
Shade_Faktor=15;

% Camera Number (considered camera)
cam_nr='cam1_';

% File Name of Recorded Force Values       (in LabView)
name_force='force_1.txt';
% File Name of Recorded Pressure Values       (in LabView)
name_pressure='pressure_1.txt';

% First image ==> serves to cut out the area to be viewed
```

```
first_image=strcat(cam_nr,'1.tif');

% Manually cutting out the area to be examined
% (includes the measurement marks);
% Contains coordinate points of the created region of interest
img2 = imread(first_image);
[crop, rect] = imcrop(img2);
sizecrop2=size(crop);
height2=sizecrop2(1,1);
width2=sizecrop2(1,2);

% Apply the predefined shade factor in the image
% stack and display the cropped area
subtraction_matrix2=ones(height2,width2)*Shade_Faktor;
subtraction_matrix2=uint8(subtraction_matrix2);
imshow(crop-subtraction_matrix2);

% Create a matrix for caching the various shifts
data=zeros(Anzahl_Bilder,11);

% Squares search loop in image ==> returns center points and
% Page coordinates of found squares
for k=1:length(data);
%Filename with variable and file extension
image=strcat(cam_nr, num2str(k), '.tif');
% Reading the current picture
imageData = imread(image);
% Trim the recalled image ==> same coordinates as before
crop_image2=imcrop(imageData, rect);
filter_matrix = crop_image2-subtraction_matrix2;
% Convert the image to black and white
bw2 = im2bw(filter_matrix);
stats2 = [regionprops(bw2)];
% Search function of the squares; A rectangle is placed
% around each mark and the white-based area centroid is
% determined in absolute coordinates
list=zeros(length(stats2),1);
for i = 1:numel(stats2)
    % BoundingBox(4) & (3) describes the side lengths of the
    % rectangle in x- and y- direction
    if and(stats2(i).BoundingBox(4)>g_M,stats2(i).BoundingBox(3)>g_M);
        % the index of all matching and recognized objects
        % is written in a list
        list(i)=i;
    end
    rectangle('Position', stats2(i).BoundingBox);
end

% Empty list entries (for objects having geometries smaller
% than required) had index "0"; these values    are    deleted with
% this function
ls=list(list ~= 0);
```

```

% Position of the individual measuring marks
% Marking to determine:
% the horizontal stretching (west)
a=ls(1);
% the vertical stretching (north 2)
b=ls(2);
% the vertical stretching (north 1)
c=ls(3);
% the vertical stretching (south 1)
d=ls(4);
% the vertical stretching (south 2)
e=ls(5);
% the horizontal stretching (east)
f=ls(6);

% Determination of the horizontal displacement
% (position(west)-postion(east))
data(k,1)=abs(stats2(a).Centroid(2)-stats2(f).Centroid(2));
% (position(north 2)-postion(south 2))
data(k,2)=abs(stats2(b).Centroid(1)-stats2(e).Centroid(1));
% (position(north 1)-postion(south 1))
data(k,3)=abs(stats2(c).Centroid(1)-stats2(d).Centroid(1));
end

% Creation of a solution matrix;
% Here, the determined absolute expansions
% and the smoothed and corrected force and
% pressure values    are    stored
Solution_matrix=zeros(Anzahl_Bilder,11);

for u=1:length(Solution_matrix);
% Horicontal strain
data(u,4)=data(u,1)-data(1,1);
Solution_matrix(u,1)=100*data(u,4)/data(1,1);
% Vertical strain (north 2 - south 2)
data(u,5)=data(u,2)-data(1,2);
Solution_matrix(u,2)=100*data(u,5)/data(1,2);
% Vertical strain (north 1 - south 1)
data(u,6)=data(u,3)-data(1,3);
Solution_matrix(u,3)=100*data(u,6)/data(1,3);
end

% Filename of the raw data of the force signal (in volts)
FileName = name_force

% In the text file, commas are replaced
% by points to be compatible with Matlab.
Data = fileread(FileName);
Data = strrep(Data, ',', '.');
FID = fopen(NewFileName, 'w');
NewFileName = 'force_1_p.txt'
```

```
fwrite(FID, Data, 'char');
fclose(FID);

% Reading the force signal
[delete1,delete2,volt,time]=textread(NewFileName, '%s%f%f%f','delimiter',';');

clear delete1 delete2;

% Convert and smooth from Volt to Newton; In addition,
% the offset between the number of images and the measured
% force values    is    adjusted to remain consistent when creating the plots
Anzahl_Kraftwerte = length(volt);
Offset = Anzahl_Kraftwerte - Anzahl_Bilder;
Delay = round(Anzahl_Kraftwerte/Offset);
force_factor=ones(length(volt),1)*5000 ;
newton_uncalibrated=times(force_factor,volt);
newton_corrected=
newton_uncalibrated(find(rem(1:length(newton_uncalibrated),Delay)>0));
newton_corr_2 = (-1)*min(newton_corrected)+newton_corrected
Solution_matrix(1:length(newton_corrected),4)=smooth(newton_corrected,0.02,'rloess');

% Filename of the raw data of the force signal (in volts)
FileName = name_pressure

% In the text file, commas are replaced
% by points to be compatible with Matlab.
Data = fileread(FileName);
Data = strrep(Data, ',', '.');
NewFileName = 'pressure_1_p.txt'
FID = fopen(NewFileName, 'w');
fwrite(FID, Data, 'char');
fclose(FID);

% Reading the pressure signal
[delete1,delete2,volt,time]=textread(NewFileName, '%s%f%f%f','delimiter',';');

clear delete1 delete2;

% Convert and smooth from Volt to bar; In addition,
% the offset between the number of images and the measured
% force values    is    adjusted to remain consistent when creating the plots
force_factor=ones(length(volt),1)*-15 ;
pressure_uncalibrated=times(force_factor,volt);
pressure_corrected=
pressure_uncalibrated(find(rem(1:length(pressure_uncalibrated),Delay)>0));
Solution_matrix(1:length(newton_corrected),5)=smooth(pressure_corrected,0.005,'rloess');
```

A.2 Steel braiding

The subsequently introduced Python code was created to provide the steel braiding structure for the Macro model. The individual wires will be constructed as beam profiles and written into an Abaqus compatible input file, which contains the assembly.

Program Code A.2: Writing Inputfile (Steel braiding) (199 Lines of Python)

```python
# Required packages
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import matplotlib.pyplot as plt
import math

# The aim of this script is to generate an input file of the wire assemblies consisting of beam elements.

# Name of the output file (.inp)
filename = "DL_2_hellically_offset"
# Hose data according to specification
Schlauchlaenge = 150
Schlaglaenge = 50
Drahtdurchmesser = 0.6
Abstand_zw_Draehten = 0.33
Mittlerer_R_DL = (24+20.1)/4
Winkel = XXX
# Wire material characteristics
E_Modul = XXXXX
Poissonzahl = XXXX
# Interweaving; Interweaving Yes ==> "J" / No ==> "N"
Flechten ="J"
AB_ZW_DL = Drahtdurchmesser/2
# ==> Packages pointing in the same direction
Pakete = 7
# ==> Wires per package (wires 1 are pitched to the left / wires 2 are pitched to the right)
Draehte1 = 5
Draehte2 = 5

# Adjustment parameter
# ==> Starting angle of the right-pitched wire packets (optionally changeable if there are collisions between the wire packets with each other)
# ==> Should always be multiples of the offset angle between each package
Startwinkel = 18
# ==> Intensity of the amplitudes along the wire length (radius function) ==> Normalization of the angle function
Intensitaet = 0.31
```

```
# ==> Wenn nicht alle Pakete erstellt werden sollen (zur Fehlersuche) (0 < SP < Pakete) (Normal immer auf Wert der Maximalen Paketmenge pro Richtung)
SP = Pakete
# ==> Incremente (defines how many points are created per wire)
Incremente = 300

# Calculation of the distances and angles between the respective wires
alpha = Winkel
beta = 90 - alpha
r = Mittlerer_R_DL
h = Abstand_zw_Draehten * math.cos(beta*np.pi/180)
v = Abstand_zw_Draehten * math.sin(beta*np.pi/180)
gamma = 2*math.asin((h/r))*180/np.pi
Offset = gamma
Offset2 = gamma
Winkel_offset = 360/Pakete

# Convertion start angle to radians
Start_offset = (Startwinkel*np.pi/180)

# Data array for wire packages which are pitched to the left
array_2Dl = []
# Data array for wire packages which are pitched to the right
array_2Dr = []

# Construction of the individual wires in succession, taking into account the interweaving
# Determining the length of the wire based on the determined revolutions (in radians)
theta = np.linspace(0, (2*Schlauchlaenge/Schlaglaenge)*np.pi, Incremente)
for n in range(0,360,Winkel_offset):
    for m in range(Draehte1):
        # Pitched to the left
        # Adjust the offsets between the wires and the packages
        Draht_Versatzl = np.linspace(0-m*(np.pi/180)*Offset+(n*np.pi/180), (2*Schlauchlaenge/Schlaglaenge)*np.pi-m*(np.pi/180)*Offset+(n*np.pi/180), Incremente)
        zl = np.linspace(0, Schlauchlaenge, Incremente)
        # Radius function that takes into account the interweavings
        if Flechten == "J":
            rl = Mittlerer_R_DL+Intensitaet*np.sin((theta-m*(np.pi/180)*Offset2)*(Pakete/2)-(0.5*np.pi))
        else:
            rl = Mittlerer_R_DL + AB_ZW_DL
        xl = rl * np.sin(Draht_Versatzl)
        yl = rl * np.cos(Draht_Versatzl)
        array_2Dl.append([xl,yl,zl])
    for m1 in range(Draehte2):
        # Pitched to the left
        # Adjust the offsets between the wires and the packages
        Draht_Versatzr = np.linspace((2*Schlauchlaenge/Schlaglaenge)*np.pi+m1*(np.pi/180)*Offset+((n*np.pi/180)+Start_offset), 0+m1*(np.pi/180)*Offset+((n*np.pi/180)+Start_offset), Incremente)
        zr = np.linspace(0, Schlauchlaenge, Incremente)
        # Radius function that takes into account the interweavings
        if Flechten == "J":
            rr = Mittlerer_R_DL+Intensitaet*np.sin((theta-m1*(np.pi/180)*Offset2)*(Pakete/2)-(1.5*np.pi))
        else:
            rr = Mittlerer_R_DL - AB_ZW_DL
        xr = rr * np.sin(Draht_Versatzr)
        yr = rr * np.cos(Draht_Versatzr)
        # Array; 1.column x-coordinates / 2.column y-coordinates / 3.column z-coordinates
        array_2Dr.append([xr,yr,zr])

# Coordinate matrix
list =[]
list = [array_2Dl,array_2Dr]

# Plotting curves for both directions (optional out commentable to get only one direction)
fig = plt.figure()
ax = fig.gca(projection='3d')
for m in range(Draehte1*SP):
    ax.plot(list[0][m][0],list[0][m][1],list[0][m][2],'b',linewidth=0.5)
for m in range(Draehte2*SP):
   ax.plot(list[1][m][0],list[1][m][1],list[1][m][2],'r',linewidth=0.5)
ax.legend()
plt.show()

# Inputfile creation (Abaqus)

# Beam profiles created according to standardized rules and conventions; Result file with wire assembly
a=""
a += ("*HEADING" + " \n" +
      "**" + " \n" +
      "** PARTS" + " \n" +
      "**" + " \n" +
      "*Part, name=Draht" + " \n" +
      "*End Part" + " \n" +
      "**" + " \n" +
      "** ASSEMBLY" + " \n" +
      "**" + " \n" +
      "*Assembly, name=Assembly" + " \n" +
      "**" + " \n")
cv = 0
cv1 = 0
for m in range(len(list)):
    for n in range(len(list[m])):
        cv1 += 1
        a += "*Instance, name=Draht-" + str(cv1) + ", part=Draht" + " \n" + "          0.,           0.,        0." + " \n"
        a += "*Node" + " \n"
        for o in range(len(list[0][0][0])):
            a += str(o+1) + ", " + str(list[m][n][0][o]) + ", " + str(list[m][n][1][o]) + ", " + str(list[m][n][2][o]) + " \n"
        a += "*Element, type=B31" + " \n"
        for p in range(1,len(list[0][0][0])):
            a += str(p) + ", " + str(p) + ", " + str(p+1) + " \n"
            cv += 1
        a += ("*Nset, nset=_PickedSet2, internal, generate" + " \n" +
            "1,  " + str(cv+1) + ",    1" + " \n" +
            "*Elset, elset=_PickedSet2, internal, generate" + " \n" +
            "1,  " + str(cv) + ",    1" + " \n" +
            "** Section: Draht_1  Profile: Profile-1" + " \n" +
            "*Beam Section, elset=_PickedSet2, material=draht, temperature=GRADIENTS, section=CIRC" + " \n" +
# Diameter is taken into account in beam profile
            str(Drahtdurchmesser/2) + " \n" +
            "1.,0.,0." + " \n" +
            "*End Instance" + " \n")
# Enter material characteristics
a+= ("*End Assembly" + " \n" +
    "**" + " \n" +
    "** MATERIALS" + " \n" +
    "**" + " \n" +
    "*Material, name=draht" + " \n" +
    "*Elastic" + " \n" +
    str(E_Modul) + "., " + str(Poissonzahl) + " \n" +
    "**" + " \n")
fobj = open(filename + ".inp", "w")
fobj.write(a)
fobj.close()
```

A.3 VBM-Method

The subsequently introduced Python code was developed to create an Abaqus-compatible input file that uses the available CT images to create voxel-based mesh models.

Program Code A.3: VBM method (216 Lines of Python)

```python
# Required packages
import cv2
import numpy as np
from skimage.morphology import label
from skimage.measure import regionprops

# Input data
input_name="VBM_Sub.inp"
Anzahl =10
Start_pic = 50
filename_raw="2SN"
file_ending=".tif"
Reference_pic = "MIDPOINT_DATA_0000"
# Minimum Shadefaktor to determine which gray area is declared as air
minimum_shade=60
# Length per pixel (in mm)
lpp = 0.020833

# Center of the 1st image ==> Serves as a reference
MIDPOINT=[]
firstimage=cv2.imread(str(Reference_pic) + ".tif")
firstimage2=cv2.cvtColor(firstimage, cv2.COLOR_BGR2GRAY)
label_img2=label(firstimage2)
regions2=regionprops(label_img2)
for props in regions2:
    if props.area > 160000 and props.area < 190000:
        MIDPOINT = [(props.centroid[0]),(props.centroid[1])]

# First picture used to select the ROI
firstimage=filename_raw + "00" + str(Start_pic) + file_ending
# Read image
im = cv2.imread(firstimage)
# Select ROI
bbox = cv2.selectROI(im)
# Crop image
imCrop = im[int(bbox[1]):int(bbox[1]+bbox[3]), int(bbox[0]):int(bbox[0]+
    bbox[2])]
# Display cropped image
resize = cv2.resize(imCrop, (0,0), fx=1, fy=1)
cv2.imshow("Image", resize)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```
# Corner coordinates of the ROI
y1=int(bbox[1])
y2=int(bbox[3]+bbox[1])
x1=int(bbox[0])
x2=int(bbox[2]+bbox[0])

# Creation of a 3D array with the gray values    of    the image stack
# Reading the defined area to be created by the VBM method from the CT image stack
list=[]
for u in range(Start_pic,Start_pic+Anzahl):
    filename = filename_raw + (4-len(str(u)))*"0" + str(u) + file_ending
    im = cv2.imread(filename)
    im2 = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    imCrop = im2[y1:y2, x1:x2]
    list.append(imCrop)

# Convert the 3D list with the gray values    of    the ROI into a 3D array
pic_arr=np.array(list, dtype=int)

# Creation of auxiliary arrays that should be used to create node numbering  and element assignment
# Basic array in the length and size of the detected image stack (ROI)
num_arr=np.zeros((len(pic_arr),len(pic_arr[0]),len(pic_arr[0][0])))
# Creation of empty arrays which have been extended by one unit in each direction (XYZ); Otherwise information would be lost in Meshing
num_arr_nodes=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_nodes2=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_nodes3=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_colors=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))

# Registration of the hose material; If the currently checked voxel has a gray value above the "minimum_shade" value, it is registered as hose material (steel or rubber)
cv_element=0
for h in range(len(pic_arr)):
    for i in range(len(pic_arr[0])):
        for j in range(len(pic_arr[0][0])):
            if pic_arr[h][i][j]>minimum_shade:
                cv_element+=1
                num_arr[h][i][j]=cv_element
            if pic_arr[h][i][j]>minimum_shade:
                num_arr_colors[h][i][j]=pic_arr[h][i][j]

# There are currently only two types of tubing (steel and rubber). If a voxel is read in with a gray value below the defined limit (166); the element number is written to the rubber list; about it for steel
cv_colors=0
```

```
rubber = []
steel = []
for h2 in range(len(num_arr_colors)):
    for i2 in range(len(num_arr_colors[0])):
        for j2 in range(len(num_arr_colors[0][0])):
            if num_arr_colors[h2][i2][j2]!=0:
                cv_colors += 1
                if num_arr_colors[h2][i2][j2] < 166 and num_arr_colors[h2][i2][j2] > 0:
                    rubber.append(cv_colors)
                if num_arr_colors[h2][i2][j2] >= 166:
                    steel.append(cv_colors)

# Numbering of the node elments in the 3D-Array; will be needed for the subsequently creation of the node lists for each element
cv=0
for m in range(len(num_arr)):
    for n in range(len(num_arr[0])):
        for o in range(len(num_arr[0][0])):
            if num_arr[m][n][o]!=0:
                cv+=1
                num_arr_nodes[m][n][o]=cv

# These loops are used to generate voxels from the corner pixels; For this purpose, an extension is introduced for each last specific pixel, in x y or z direction, so that this coordinate point also has the required 8 corner nodes
for u in range(len(num_arr_nodes)):
    for v in range(len(num_arr_nodes[0])):
        for w in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes[u][v][w]!=0:
                num_arr_nodes2[u][v][w]=1
                if num_arr_nodes[u][v][w+1]==0:
                    num_arr_nodes2[u][v][w+1]=1
                if num_arr_nodes[u][v+1][w]==0:
                    num_arr_nodes2[u][v+1][w]=1
                if num_arr_nodes[u][v+1][w+1]==0:
                    num_arr_nodes2[u][v+1][w+1]=1
                if num_arr_nodes[u+1][v][w]==0:
                    num_arr_nodes2[u+1][v][w+1]=1
                if num_arr_nodes[u][v+1][w]==0:
                    num_arr_nodes2[u+1][v+1][w]=1
                if num_arr_nodes[u+1][v+1][w+1]==0:
                    num_arr_nodes2[u+1][v+1][w+1]=1
                if num_arr_nodes[u+1][v][w]==0:
                    num_arr_nodes2[u+1][v][w]=1

# Here, the complete numbering of all relevant nodes, for the complete 3D array, which are subsequently required to form the element
cv_nodes=0
for x in range(len(num_arr_nodes)):
    for y in range(len(num_arr_nodes[0])):
```

```python
        for z in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes2[x][y][z]!=0:
                cv_nodes += 1
                num_arr_nodes3[x][y][z] = cv_nodes

# For the input file to be used, will be string elements created which contains the coordinates of each node, and one that contains the elements with the associated nodes; In addition, the correction of the hose curvature and the cut-out area is carried out here ==> one thus obtains a model which always refers to the center of the hose
cv1=0
Nodes=""
cv2=0
Elements=""
for q in range(len(num_arr_nodes)):
    for e in range(len(num_arr_nodes[0])):
        for r in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes3[q][e][r]!=0:
                cv1+=1
                Nodes += str(cv1) + ", " + str((r+x1-MIDPOINT[0])*lpp) + ", " + str((y1+e-MIDPOINT[1])*lpp) + ", " + str(q*lpp) + "\n"
            if num_arr_nodes[q][e][r]!=0:
                cv2+=1
                p1=int(num_arr_nodes3[q][e][r])
                p2=int(num_arr_nodes3[q][e][r+1])
                p3=int(num_arr_nodes3[q][e+1][r+1])
                p4=int(num_arr_nodes3[q][e+1][r])
                p5=int(num_arr_nodes3[q+1][e][r])
                p6=int(num_arr_nodes3[q+1][e][r+1])
                p7=int(num_arr_nodes3[q+1][e+1][r+1])
                p8=int(num_arr_nodes3[q+1][e+1][r])
                Elements += str(cv2) + ", " + str(p1) + ", " + str(p2) + ", " + str(p3) + ", " + str(p4) + ", " + str(p5) + ", " + str(p6) + ", " + str(p7) + ", " + str(p8) + "\n"

# This function executes the material assignment. The respective elements are assigned the appropriate materials based on their gray values. The result of this function is the output of a string containing the materials, constructed according to the rules required for the input file
def Get_Solid_sections(gray, mat):
    Material=""
    a=""
    for o in range(0,len(mat)):
        b=o+1
        if b==0:
            a += str(mat[o]) + ", "
        elif b%16 == 0:
            a += str(mat[o]) + "\n"
        elif b==len(mat):
            a += str(mat[o])
        else:
            a += str(mat[o]) + ", "
    Material += a
    material = gray
    text = " \n*Solid Section, elset=_"+gray+", material = "+material+" \n1., \n*Elset, elset=_"+gray+", internal"+" \n"
    return text + Material + " \n"

Solid_Section_Mat100 = Get_Solid_sections("Gray100",rubber)
Solid_Section_Mat254 = Get_Solid_sections("Gray254",steel)

# Creation of the inputfile
text1 = ("*Heading"+
         " \n" +
         "** Job name: Job_CT_Messung"+
         " Model name: Model-1"+
         " \n"+
         "** Generated by: Abaqus/CAE 6.14-1"+
         " \n"+
         "*Preprint, echo=NO, model=NO, history=NO, contact=NO"+
         "\n"+
         "** ----------------------------------------------------------------"+
         " \n"+
         "**"+
         " \n"+
         "** PART INSTANCE: Part-1-1"+
         " \n"+
         "** "+
         " \n"+
         "*Node")
text2 = "*Element, type=C3D8H"
# Assignement of the material properties
text3 = (" \n**"+
        " \n** MATERIALS"+
        " \n** "+
        " \n*Material, name=Gray100  \n"+
        "*Elastic \n"+
        " 220000., 0.3 \n"+
        "*Material, name=Gray254  \n"+
        "*Hyperelastic, neo hooke \n"+
        " 1.2839,0. \n"+
        "** \n")
fobj = open(input_name,"w")
fobj.write(text1+" \n"+Nodes+2*" \n"+
           text2+" \n"+Elements +" \n"+
           Solid_Section_Mat100 +" \n"+
           Solid_Section_Mat254 +" \n"+
           text3)
fobj.close()
```

A.4 Wire detection

The subsequently introduced Python code was developed in order to rebuild its steelbraiding structure by means of a CT image stack of a hose. The result is a macro code compatible with common CAD programs that can be used to create solid parts that can be assembled into an assembly.

Program Code A.4: Wire detection (RE method) (197 Lines of Python)

```python
# Required Packages
from skimage.morphology import label
from skimage.measure import regionprops
import cv2
import math
import matplotlib.pyplot as plt
import numpy as np
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
import os

# Working directory
path="C:\Master_Thesis\CT-Bilder-31102018\multiple_reinforced\
    Wire_detection"
os.chdir(path)

# Input Data
Anzahl = 3
# Length per pixel
Lpp=0.0208333333
# Name of the wire images processed in ImageJ
filename_raw="2SN_"
# Name of hose inside diameter images processed in ImageJ
filename_raw2="MIDPOINT_DATA_"
file_ending=".tif"
# Wire diameter (in m)
Diameter=0.00029

# Center point detection for the complete hose
MIDPOINT=[]
MIDPOINT_1=[]
# Center of the 1st image ==> Serves as a reference
firstimage=cv2.imread(filename_raw2 + "0000" + ".tif")
firstimage2=cv2.cvtColor(firstimage, cv2.COLOR_BGR2GRAY)
label_img2=label(firstimage2)
regions2=regionprops(label_img2)
for props in regions2:
    if props.area > 160000 and props.area < 190000:
        MIDPOINT_1 = [(props.centroid[0]),(props.centroid[1])]

```

```python
# Centers of all images
for u in range(0,Anzahl):
    filename = filename_raw2 + (4-len(str(u)))*"0" + str(u) + file_ending
    im = cv2.imread(filename)
    im2 = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    label_img = label(im2)
    regions = regionprops(label_img)
    for props in regions:
        if props.area > 160000 and props.area < 190000:
            MIDPOINT.append([(props.centroid[0]),(props.centroid[1])])

# Calculation of the offset ==> for all images
Offset=[]
for m in range(len(MIDPOINT)):
    y = -(MIDPOINT[m][0]-MIDPOINT_1[0])-MIDPOINT_1[0]
    x = -(MIDPOINT[m][1]-MIDPOINT_1[1])-MIDPOINT_1[1]
    Offset.append([y,x])

# Detection of the wire paths
# The wire center point of each wire cross section of each image of the image stack is determined and provided with an identifier
# At the beginning of each new image, the count value of the identifier is always increased by 1000; every detected midpoint raises the mark again by one ==> this numbering is used as the key for the used Dictionary
coords_list=[]
cv_1=0
for u in range(0,Anzahl):
    cv_1+=1000
    filename = filename_raw + (4-len(str(u)))*"0" + str(u) + file_ending
    im = cv2.imread(filename)
    im2 = cv2.cvtColor(im, cv2.COLOR_BGR2GRAY)
    label_img = label(im2)
    regions = regionprops(label_img)
    cv_2=0
    coords_list_pp=[]
    for props in regions:
        cv_2+=1
        coords_list_pp.append([[int(round(props.centroid[0])),int(round(props.centroid[1]))],[(props.centroid[0]+Offset[u][0])*Lpp,(props.centroid[1]+Offset[u][1])*Lpp,float(u)*Lpp],[cv_1+cv_2]])
    coords_list.append(coords_list_pp)

# Creation of an empty matrix in the size of the captured image stack
CT_Array=np.zeros((Anzahl,len(im2),len(im2[0])))

# Creation of the Dictionary, this is required to storage the precise midpoint positions of the detected wire cross-sections
# The respective identification numbers (numerical values) are entered in sequence at the detected wire centers
Keys_Coord = []
Coordinates = []
for m in range(Anzahl):
    for n in coords_list[m]:
        CT_Array[m][n[0][0]][n[0][1]]=n[2][0]
        Keys_Coord.append(str(int(n[2][0])))
        Coordinates.append(n[1])

CT_Array=np.array(CT_Array, dtype=int)
Key_and_Coord = zip(Keys_Coord,Coordinates)
Dictionary = dict(Key_and_Coord)

Keylist = Dictionary.keys()
Keylist.sort()

# Determination of the curve path of each individual wire
# This loop goes through the first picture only in the xy plane; If a wire midpoint is detected, the search area shifts by one count in the z direction and searches in a small predefined xy-field for the next possible wire midpoint. This happens until the wire has been determined over the entire length. In the list "Reinforcements", the detected center points are stored as counting characteristics e.g. "1001". The precise geometrical coordinates are saved in the previous described Dictionary. The counting characteristics serves as key values for the dictionary.
Reinforcement_List=[]
for m in coords_list[0]:
    Reinforcement=[str(m[2][0])]
    y_act=m[0][0]
    x_act=m[0][1]
    for z in range(1, Anzahl):
        cv=0
        for y in range(y_act-5,y_act+5):
            for x in range(x_act-5,x_act+5):
                if CT_Array[z][y][x]!=0:
                    x_act=x
                    y_act=y
                    Reinforcement.append(str(CT_Array[z][y][x]))
                #(optional) Error Detection ==> returns picture numbers which contains errors
                #else:
                #    cv+=1
                #    if cv == 100:
                #        print "Error in picture " + str(z) + " !!!!"
    Reinforcement_List.append(Reinforcement)

# Here the recognized wire keys are converted into a wire coordinate list which can be easily used to create input files or macro codes for conventional CAD programs.
Wire_List=[]
for m in range(len(Reinforcement_List)):
    Wire=[]
    for n in Reinforcement_List[m]:
        Wire.append(Dictionary[n])
    Wire = np.array(Wire)
    Wire = Wire.T
    Wire_List.append(Wire)

# This section allows the illustration of the respective wire paths in a curve plot.
mpl.rcParams['legend.fontsize'] = 10

fig = plt.figure()
ax = fig.gca(projection='3d')

for m in range(len(Wire_List)):
    ax.plot(Wire_List[m][1],Wire_List[m][0],Wire_List[m][2],'b',linewidth=0.4)

ax.legend()

plt.show()

# Creation of the Macro code for solidworks for the creation of Solidparts
fobj = open("Macro.txt", "w")

a = ""
a += ("Dim swApp As Object" + " \n" + " \n"
+ "Dim Part As Object" + " \n"
+ "Dim boolstatus As Boolean" + " \n"
+ "Dim longstatus As Long, longwarnings As Long" + " \n"+" \n")

for n in range(len(Wire_List)):
    a += ("Sub main" + str(n+1) + "()" + " \n" + " \n" +
    "Set swApp = Application.SldWorks" + " \n" + " \n" +
    "Set Part = swApp.ActiveDoc" + " \n" +
    "Dim COSMOSWORKSObj As Object" + " \n" +
    "Dim CWAddinCallBackObj As Object" + " \n" +
    "Set CWAddinCallBackObj = swApp.GetAddInObject(\"CosmosWorks.CosmosWorks\")" + " \n" + " \n" +

    "' New Document" + " \n" +
    "Dim swSheetWidth As Double" + " \n" +
    "swSheetWidth = 0" + " \n" +
    "Dim swSheetHeight As Double" + " \n" +
    "swSheetHeight = 0" + " \n" +
    "Set Part = swApp.NewDocument(\"C:\ProgramData\SolidWorks\SOLIDWORKS 2017\\templates\Teil.prtdot\", 0, swSheetWidth, swSheetHeight)" + " \n" +
    "Dim swPart As PartDoc" + " \n" +
    "Set swPart = Part" + " \n" +
    "swApp.ActivateDoc2 \"Teil"+str(n+1)+"\", False, longstatus" + " \n" +
    "Set Part = swApp.ActiveDoc" + " \n" +
    "Part.InsertCurveFileBegin" + " \n")
    for m in range(len(Wire_List[n][0])):
        if m%5==0:
            a +="boolstatus = Part.InsertCurveFilePoint("+ str(Wire_List[n
                ][0][m]/1000.0)+", "+ str(Wire_List[n][1][m]/1000.0)+", "+
                str(Wire_List[n][2][m]/1000.0)+")"+" \n"


    a +=( "boolstatus = Part.InsertCurveFileEnd()" + " \n" +
    "Part.ClearSelection2 True" + " \n" +
    "boolstatus = Part.Extension.SelectByID2(\"Kurve1\", \"REFERENCECURVES
        \", 0, 0, 0, True, 0, Nothing, 0)" + " \n" +
    "Part.ClearSelection2 True" + " \n" +
    "boolstatus = Part.Extension.SelectByID2(\"Kurve1\", \"REFERENCECURVES
        \", 0, 0, 0, False, 4, Nothing, 0)" + " \n" +
    "Dim myFeature As Object" + " \n" +
    "Set myFeature = Part.FeatureManager.InsertProtrusionSwept4(False,
        False, 0, False, False, 0, 0, False, 0, 0, 0, 10, True, True, True,
         0, True, True, "+str(Diameter)+", False)" + " \n" + " \n" +
    "' Save As" + " \n" +
    "longstatus = Part.SaveAs3(\"C:\Master_Thesis\CT-Bilder-31102018\
        Parts_Multiple\Teil"+str(n+1)+".SLDPRT\", 0, 2)" + " \n" +" \n" +"
        \n" +" \n" +
    "Set CWAddinCallBackObj = Nothing" +" \n" +
    "Set COSMOSWORKSObj = Nothing" +" \n" +
    "End Sub" + " \n" + " \n")


a += "Sub Test()" + " \n"
for m in range(len(Wire_List)):
    a += "    main" + str(m+1) + " \n"

a += "End Sub"

fobj.write(a)
fobj.close()
```

A.5 Void detection

The subsequently introduced Python code was developed to convert occuring voids in the rubber matrix into negatives, which can subsequently be subtracted from the solid material. The creation of the void negatives was carried out according to the VBM method.

Program Code A.5: Void detection (RE method) (226 lines of Python)

```python
# Required packages
import math
import matplotlib.pyplot as plt
import numpy as np
from skimage import io
from skimage.draw import circle
from skimage.measure import label, regionprops
from skimage.transform import rotate
import cv2

# Input Data
filename_raw2="CT_"
Anzahl = 60
file_ending = ".tif"
filename_raw2="CT_"
# length per pixel
lpp = 0.020833

# Select ROI
bbox = cv2.selectROI(cv2.imread("CT_0000.tif"))

# Crop image
imCrop = cv2.imread("CT_0000.tif")[int(bbox[1]):int(bbox[1]+bbox[3]), int(bbox[0]):int(bbox[0]+bbox[2])]

# Display cropped image
resize = cv2.resize(imCrop, (0,0), fx=10, fy=10)
cv2.imshow("Image", resize)
cv2.waitKey(0)
cv2.destroyAllWindows()

y1=int(bbox[1])
y2=int(bbox[3]+bbox[1])
x1=int(bbox[0])
x2=int(bbox[2]+bbox[0])

# Correct the image so that only the voids are visible. For this purpose, the edge regions of the tube contour are uniformized to obtain ordered geometries.
list=[]
for m in range(44,Anzahl):
    filename = filename_raw2 + (4-len(str(m)))*"0" + str(m) + file_ending
    im_1 = cv2.imread(filename)
    im_2 = cv2.bitwise_not(cv2.cvtColor(im_1, cv2.COLOR_BGR2GRAY))
    im_0 = np.zeros((len(im_2),len(im_2[0])))
    im_3 =cv2.cvtColor(im_1, cv2.COLOR_BGR2GRAY)
    im_3[im_3 > 22]=50
    im_3[im_3 < 23]=0
    label_img = label(im_2)

    label_img=label(im_2)
    regions=regionprops(label_img)

    for props in regions:
        if props.area >450000 and props.area < 500000:

            im_3 = cv2.circle(im_3, (int(props.centroid[1]), int(props.centroid[0])), 565, 60, 50)
            im_3 = cv2.circle(im_3, (int(props.centroid[1]), int(props.centroid[0])), 385, 60, 22)
            im_0 = cv2.circle(im_0, (int(props.centroid[1]), int(props.centroid[0])), 475, 50, 170)

            im_4 = im_0+im_3
            im_4 = im_4[x1:x2, y1:y2]
            im_4[im_4==60]=0
            im_4[im_4>50]=0
            im_4[im_4==50]=200

            list.append(im_4)

# Creation of auxiliary arrays that should be used to create node numbering and element assignment
# Basic array in the length and size of the detected image stack (ROI)
pic_arr=np.array(list, dtype=int)
# Creation of empty arrays which have been extended by one unit in each direction (XYZ); Otherwise information would be lost in Meshing
num_arr=np.zeros((len(pic_arr),len(pic_arr[0]),len(pic_arr[0][0])))
num_arr_nodes=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_nodes2=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_nodes3=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))
num_arr_colors=np.zeros((len(pic_arr)+1,len(pic_arr[0])+1,len(pic_arr[0][0])+1))

# Registration of the hose material; If the currently checked voxel has a gray value above the "minimum_shade" value, it is registered as hose material (steel or rubber)
cv_element=0
for h in range(len(pic_arr)):
    for i in range(len(pic_arr[0])):
        for j in range(len(pic_arr[0][0])):
            if pic_arr[h][i][j]!=0:
                cv_element+=1
                num_arr[h][i][j]=cv_element
            if pic_arr[h][i][j]!=0:
                num_arr_colors[h][i][j]=pic_arr[h][i][j]
# There are currently only two types of tubing (steel and rubber). If a voxel is read in with a gray value below the defined limit (166); the element number is written to the rubber list; about it for steel
cv_colors=0
Seele = []
ZWP = []
Decke =[]
for h2 in range(len(num_arr_colors)):
    for i2 in range(len(num_arr_colors[0])):
        for j2 in range(len(num_arr_colors[0][0])):
            if num_arr_colors[h2][i2][j2]!=0:
                cv_colors += 1
                if num_arr_colors[h2][i2][j2] != 0:
                    Decke.append(cv_colors)

# Numbering of the node elments in the 3D-Array; will be needed for the subsequently creation of the node lists for each element
cv=0
for m in range(len(num_arr)):
    for n in range(len(num_arr[0])):
        for o in range(len(num_arr[0][0])):
            if num_arr[m][n][o]!=0:
                cv+=1
                num_arr_nodes[m][n][o]=cv

# These loops are used to generate voxels from the corner pixels; For this purpose, an extension is introduced for each last specific pixel, in x y or z direction, so that this coordinate point also has the required 8 corner nodes
for u in range(len(num_arr_nodes)):
    for v in range(len(num_arr_nodes[0])):
        for w in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes[u][v][w]!=0:
                num_arr_nodes2[u][v][w]=1
                if num_arr_nodes[u][v][w+1]==0:
                    num_arr_nodes2[u][v][w+1]=1
                if num_arr_nodes[u][v+1][w]==0:
                    num_arr_nodes2[u][v+1][w]=1
                if num_arr_nodes[u][v+1][w+1]==0:
                    num_arr_nodes2[u][v+1][w+1]=1
                if num_arr_nodes[u+1][v][w]==0:
                    num_arr_nodes2[u+1][v][w+1]=1
                if num_arr_nodes[u][v+1][w]==0:
                    num_arr_nodes2[u+1][v+1][w]=1
                if num_arr_nodes[u+1][v+1][w+1]==0:
                    num_arr_nodes2[u+1][v+1][w+1]=1
                if num_arr_nodes[u+1][v][w]==0:
                    num_arr_nodes2[u+1][v][w]=1

# Here, the complete numbering of all relevant nodes, for the complete 3D array, which are subsequently required to form the element
cv_nodes=0
for x in range(len(num_arr_nodes)):
    for y in range(len(num_arr_nodes[0])):
        for z in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes2[x][y][z]!=0:
                cv_nodes += 1
                num_arr_nodes3[x][y][z] = cv_nodes

# For the input file to be used, will be  string elements created which contains the coordinates of each node, and one that contains the elements with the associated nodes; In addition, the correction of the hose curvature and the cut-out area is carried out here ==> one thus obtains a model which always refers to the center of the hose
cv1=0
Nodes=""
cv2=0
Elements=""
for q in range(len(num_arr_nodes)):
    for e in range(len(num_arr_nodes[0])):
        for r in range(len(num_arr_nodes[0][0])):
            if num_arr_nodes3[q][e][r]!=0:
                cv1+=1
                Nodes += str(cv1) + ", " + str((float(r)+x1)*lpp) + ", " + str((float(e)+y1)*lpp) + ", " + str(float(q)*lpp) + " \n"
            if num_arr_nodes[q][e][r]!=0:
                cv2+=1
                p1=int(num_arr_nodes3[q][e][r])
                p2=int(num_arr_nodes3[q][e][r+1])
                p3=int(num_arr_nodes3[q][e+1][r+1])
                p4=int(num_arr_nodes3[q][e+1][r])
                p5=int(num_arr_nodes3[q+1][e][r])
                p6=int(num_arr_nodes3[q+1][e][r+1])
                p7=int(num_arr_nodes3[q+1][e+1][r+1])
                p8=int(num_arr_nodes3[q+1][e+1][r])
                Elements += str(cv2) + ", " + str(p1) + ", " + str(p2) + ", " + str(p3) + ", " + str(p4) + ", " + str(p5) + ", " + str(p6) + ", " + str(p7) + ", " + str(p8) + " \n"

# This function executes the material assignment. The respective elements are assigned the appropriate materials based on their gray values. The result of this function is the output of a string containing the materials, constructed according to the rules required for the input file
def Get_Solid_sections(gray, mat):
    Material=""
    a=""
```

```python
    for o in range(0,len(mat)):
        b=o+1
        if b==0:
            a += str(mat[o]) + ", "
        elif b%16 == 0:
            a += str(mat[o]) + " \n"
        elif b==len(mat):
            a += str(mat[o])
        else:
            a += str(mat[o]) + ", "
    Material += a
    material = gray
    text = " \n*Solid Section, elset=_"+gray+", material = "+material+" \n1., \n*Elset, elset=_"+gray+", internal"+" \n"
    return text + Material + " \n"

Solid_Section_Mat150 = Get_Solid_sections("Decke",Decke)

# Creation of the inputfile
text1 = ("*Heading"+
         " \n" +
         "** Job name: Job_CT_Messung"+
         " Model name: Model-1"+
         " \n"+
         "** Generated by: Abaqus/CAE 6.14-1"+
         " \n"+
         "*Preprint, echo=NO, model=NO, history=NO, contact=NO"+
         "\n"+
         "** ----------------------------------------------------------------"+
         " \n"+
         "**"+
         " \n"+
         "** PART INSTANCE: Part-1-1"+
         " \n"+
         "** "+
         " \n"+
         "*Node")

text2 = "*Element, type=C3D8H"

text3 = (" \n**"+
        " \n** MATERIALS"+
        " \n** "+
        " \n*Material, name=Decke  \n"+
        "*Elastic \n"+
        " 210000., 0.3 \n"+
        "*Material, name=ZWP    \n"+
        "*Elastic \n"+
        "70000., 0.38 \n"+
        " \n*Material, name=Seele  \n"+
          "*Elastic \n"+
          " 210000., 0.3 \n"+
          " \n** ")


fobj = open("job-1.inp","w")

fobj.write(text1+" \n"+Nodes+2*" \n"+
           text2+" \n"+Elements +" \n"+
           Solid_Section_Mat150 +" \n"+
           text3)

fobj.close()
```