Modeling and Simulation of Underwater Textile-Based Soft Robotic Gripper

Amneh Nasir

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Rodrigo Martins de Matos Ventura
Prof. John Nassour

Examination Committee:

Chairperson: Prof. Teresa Maria Canavarro Menéres Mendes de Almeida
Supervisor: Prof. Rodrigo Martins de Matos Ventura
Member of the Committee: Prof. Plinio Moreno López

November 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

Sampling is an essential method for many science domains to comprehend the rich and complex marine life in these habitats. Yet, the samples that are to be taken may be delicate organisms that have been around for thousands of years and need to be handled carefully. Human divers or remotely operated vehicles (ROVs) with rigid manipulators are two common traditional ways for handling these samples, but they might be ineffective due to the lack of the necessary precision and adaptability for sensitive jobs.

The Textile Robotic Hand (TRH) created by Technische Universität München is the basis for the soft robotic underwater manipulator adapted in this work. The hand will be adapted to function in an underwater environment. The manipulator will provide a high degree of compliance and adaptability, enabling it to grasp and manipulate delicate objects with minimal risk of damage.

Numerous key steps will be conducted to adapt the soft robotic underwater manipulator presented in this study. Firstly, the robot's mechanics will be modeled to guarantee it fits the special criteria of underwater manipulation. And secondly, its motion will be simulated using multiple Simulation softwares.

This research has successfully simulated a multi-part textile-based pneumatic robotic hand using the SoftRobots plugin in SOFA. The findings of this study will be beneficial for comprehending the capabilities and constraints of soft textile robotic manipulators in aquatic environments and will aid in the future development of more creative applications.

Keywords

Underwater Manipulators; Textile Robotics; SOFA Framework ; Mechanical Modeling.

Resumo

A amostragem é um método essencial para que em muitos domínios científicos se compreenda a rica e complexa vida marinha nesses habitats. No entanto, as amostras que devem ser recolhidas podem ser organismos delicados que existem há milhares de anos e precisam ser manuseados com cuidado. Mergulhadores humanos ou veículos operados remotamente (ROVs) com manipuladores rígidos são duas maneiras tradicionais comuns de lidar com essas amostras, mas podem ser ineficazes devido à falta da precisão e adaptabilidade necessárias para trabalhos sensíveis.

A Mão Robótica Têxtil (TRH) criada pela Technische Universität München é a base para o manipulador robótico subaquático adaptado neste trabalho. A mão será adaptada para funcionar em um ambiente subaquático. O manipulador fornecerá um alto grau de conformidade e adaptabilidade, permitindo que agarre e manipule objetos delicados com risco mínimo de danos.

Vários passos importantes serão realizados para adaptar o manipulador robótico subaquático apresentado neste estudo. Em primeiro lugar, a mecânica do robô será modelada para garantir que se encaixe nos critérios especiais de manipulação subaquática. E em segundo lugar, seu movimento será simulado usando vários pacotes de simulação.

Esta pesquisa simulou com sucesso uma mão robótica pneumática têxtil composta por várias partes usando o *plugin* SoftRobots no SOFA. Os resultados deste estudo serão benéficos para compreender as capacidades e limitações de manipuladores robóticos têxteis em ambientes aquáticos e ajudarão no desenvolvimento futuro de aplicações mais criativas.

Palavras Chave

Manipuladores Subaquáticos; Robótica Têxtil; Estrutura SOFA; Modelagem Mecânica.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Setting

Humanity has long been fascinated by the discovery of the ocean's depths [1]. Our urge to explore and study the intricate ecosystems that exist under the waves grows as our comprehension of the underwater environment grows. The ocean's depths are home to a diverse and complex marine life that humans have yet to fully discover. Sampling is a crucial approach for scientific investigation in many fields, such as biology and geology, and it has helped us better understand the rich and diverse marine species that live in these ecosystems.

Deep-sea coral is an example of an underwater organism that is thousands of years old and must be handled with care. These corals can live for millennia and serve as vital habitats for a wide range of marine species. They are, however, delicate and easily damaged by traditional sampling approaches such as dredging or trawling [2]. The enormous clam, which may survive for hundreds of years and is highly valuable in the shellfish industry, is another example [3].

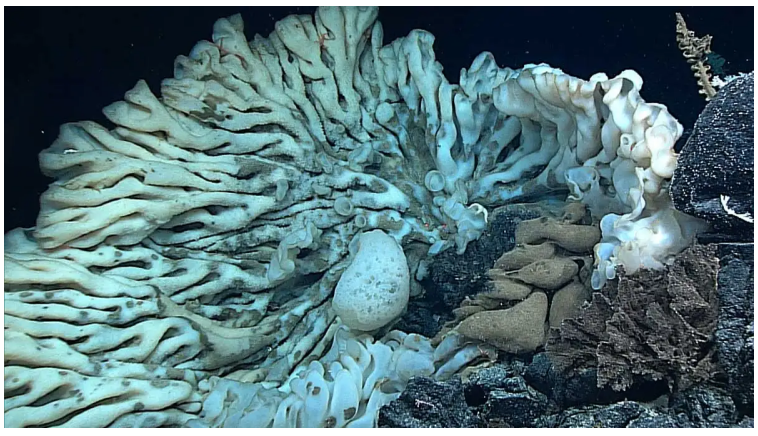


Figure 1.1: A Sponge of the Rossellidae family estimated to be tens of thousands years old. Source: [4]

Traditional methods such as human divers or remotely operated vehicles (ROVs) with stiff manipulators can be inefficient and potentially harmful due to their lack of precision and adaptability [5]. Marine manipulator systems have traditionally been heavy-duty and energy-intensive, making them suited for commercial applications such as undersea construction and salvaging operations [6]. However, the need for soft robotics has increased as marine biology researchers require more nimble and accurate manipulators to study delicate species [7]. Soft robotics can provide energy-efficient and flexible manipulators, such as full manipulators and gripping tools [8]. While the latter has only been tried in shallow waters, the former has been proven at varied depths.

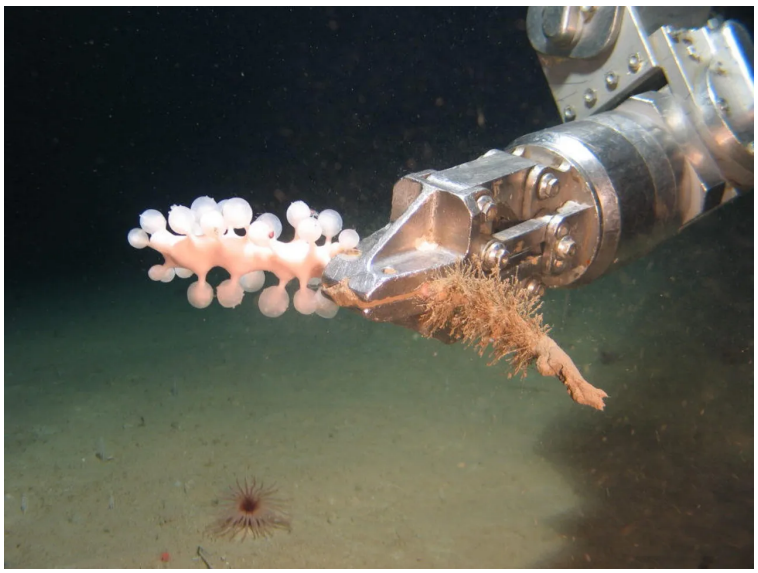


Figure 1.2: ROV with a rigid manipulator for underwater sample collection. Source: [9]

Human divers, despite their skills, have a limited range of motion and endurance. Moreover, standard manipulators may be too rigid or cumbersome for delicate tasks. Soft robotic manipulators, with their high level of compliance and adaptability, present a solution to these challenges. They can effectively handle and operate with delicate objects without the risk of damage [1].

This study aims to meet the demand for adaptive manipulator systems by studying a soft robotic manipulator capable of performing delicate and precise operations at varied depths. The underwater manipulator will be based on a Soft Textile robotic Hand from Technische Universität München [10], which uses air-pump actuation with overlay curvature and pressure sensors. This study will help build more innovative applications in the future given the various fields of soft robotics.

Figure 1.3: Various poses of a soft robotic hand alongside a wearable fabric glove. Source:[10]

1.2 Objectives and Research Questions

The primary focus of this research revolves around the adaptation and Simulation of the Textile Robotic Hand (TRH) for underwater environments. The research questions guiding this study are:

1. How will the TRH be affected with the dynamic underwater environments, especially considering water pressure and drag?
2. Which simulation platforms provide the most accurate representation of the soft robotic hand's performance in underwater conditions?
3. How can the soft robotic hand be integrated with other underwater technologies, such as autonomous underwater vehicles, to enhance their capabilities in marine environments?

These questions aim to delve deeper into the challenges and potential solutions for deploying soft robotic hands in underwater environments, emphasizing simulation , design optimization, and integration with other marine technologies.

1.3 Methodological Approach

The foundation of this research is the Textile Robotic Hand developed by Technische Universität München. This study employs various simulation platforms, including SOFA, Abaqus, and SolidWorks, to understand and predict the mechanical behavior of the pneumatic (to become hydraulic) actuators of the TRH. The goal is to identify the most suitable simulation environment that can accurately represent the manipulator's performance in regular conditions, such that it can later be implemented in underwater conditions.

1.4 Significance and Relevance of the Research

This research seeks to adapt a soft robotic manipulator for delicate operations in underwater environments. The significance lies in addressing the need for soft robotic systems that can handle sensitive marine samples without causing harm. Beyond marine applications, soft robotic manipulators have potential in areas like minimally invasive surgeries, manufacturing of fragile items, and space exploration. This research aims to push the boundaries of soft robotics, making them more versatile and adaptable to various challenging environments.

1.5 Contributions

This research contributes to the field of underwater robotics in several ways:

- Adapting an existing Soft Hand design for underwater applications, ensuring it can handle delicate marine samples safely.
- Extensive exploration and comparison of multiple simulation platforms, including SOFA, Abaqus, and SolidWorks, to identify the most suitable environment for the simulation of soft robotic manipulators.
- Providing a comprehensive mechanical modeling analysis of the manipulator, offering insights into its performance under various underwater conditions.

The insights from this study will pave the way for future research in soft robotics, especially in simulation fields.

1.6 Thesis Outline

- **Chapter 1: Introduction to Underwater Exploration**
  This chapter sets the stage by discussing the significance of underwater exploration, the challenges of handling delicate marine samples, and the potential of soft robotics in addressing these challenges.
- **Chapter 2: Evolution of Soft Robotic Manipulators**
  This chapter delves into the history and advancements in soft robotics, with a special focus on underwater applications and the integration of textile materials.
- **Chapter 3: Mechanical Modeling of the Soft Manipulator in An Underwater Environment**
  Here, the modeling principles, and mechanical considerations of the soft robotic manipulator are discussed, providing insights into its construction and functionality in underwater scenarios.
- **Chapter 4: Simulation and Analysis of Different CAD Softwares**
  This chapter details the simulation processes using platforms like SolidWorks, SOFA Framework, and Abaqus. It discusses the challenges and findings from each simulation environment, guiding the design and testing phases.

- **Chapter 5: Evaluation and Optimization of the Physical Components**
  In this chapter, the components of the soft manipulator is evaluated. It also discusses the optimization processes undertaken to enhance its efficiency and adaptability.

- **Chapter 6: Conclusions and Future Directions**
  The final chapter wraps up the research, summarizing the key contributions and findings. It also outlines potential future research avenues and applications of the developed soft robotic manipulator.

- **References**
  This section encompasses all the references cited throughout the thesis.

Chapter 2

Evolution of Soft Robotic Manipulators

2.1 Introduction

The purpose of the literature review section is to offer a complete overview of present research on soft robotic manipulators, with a special emphasis on their underwater applications. The section will start with the history and present state of soft robotics, as well as the most recent advancements in the area. It will then dig into the various procedures and strategies utilized for collecting and manipulating underwater samples. Furthermore, the literature review will highlight the study's unique research emphasis, which is to modify an existing Soft Hand for underwater functionality. After that, a full discussion of the adaptation mechanisms to be applied.

2.2 Evolution and Significance of Soft Robotics

Soft robotics is a relatively new field that has garnered significant attention in recent years due to the capabilities of soft, compliant materials to perform tasks in a safe and adaptive manner. The origins of soft robotics can be traced back to the 1950s when surgeon Joseph L. McKibben invented pneumatic artificial muscles to aid patients with severe residual paralysis [11]. Since then, the use of soft materials in manipulators has evolved and expanded, finding applications in manufacturing, medical, and search and rescue operations [12].

Soft robotics focuses on the design of robots made of soft, flexible materials, enabling them to move and manipulate objects in ways reminiscent of living organisms [13]. These robots offer several advantages over their rigid counterparts, including adaptability to complex environments and the ability to perform delicate tasks [14]. While the concept of soft robotics has been around since the 1950s, significant advancements in the field began in the early 2000s [15].

An early instance of soft robotics is the elephant trunk robot developed in 1984 by K.C. Gupta and P.C. Jain [16]. Made from soft materials like rubber and cloth, this robot emu-

lates the structure and movements of an elephant's trunk. Another notable example is the octopus-inspired robot developed in 2005 by Cecilia Laschi and her team [17]. Comprising eight limbs made of soft silicone, this robot mimics the movements of an octopus.

The recent surge in interest in soft robotics has led to the development of various innovative soft robots. One such creation is the textile robotic hand developed by John nassour in Technische Universität München (TUM) [18]. Operated by air pressure, this soft, textile-based robot has applications ranging from prosthetics to soft grippers. Another notable mention is the soft robotic gripper designed by Jamali et al. in 2020 [19], constructed from flexible silicone material, adept at handling delicate objects without causing damage.

Overall, soft robotics has emerged as a promising research field with several potential applications. The potential for soft robots to accomplish complicated tasks in challenging environments is expanding as new materials and actuation technologies are developed, as well as breakthroughs in modeling and control approaches.

Soft robotics also offers potential in environments where traditional robots might face challenges. For instance, underwater environments, which are dynamic and unpredictable, can benefit from the adaptability and resilience of soft robots [20]. Additionally, soft robots can mimic marine creatures, allowing them to blend into their surroundings and interact with marine life without causing harm [21].

2.3 Exploring Soft Robotic Manipulation

Soft robotic manipulators have garnered significant attention in the robotics community due to their high compliance, flexibility, and capability to interact safely with humans and delicate objects. Constructed from soft materials such as elastomers, polymers, and textiles, these manipulators can bend, twist, and deform, mimicking the behavior of biological tissues [22]. The inception of soft robotic manipulators can be traced back to the early 2000s, with substantial advancements in the subsequent years [15].

Rus et al. pioneered the development of a soft robotic gripper in 2003, utilizing a silicone-based material to grasp objects of varying shapes and sizes [23]. In 2008, Polygerinos et al. introduced a pneumatic soft robotic manipulator with multiple bendable segments, demonstrating its ability to grip and lift objects [24]. Furthering the medical applications, Cianchetti et al. in 2011 designed a soft robotic manipulator equipped with an endoscope, targeting minimally invasive surgical procedures [25].

Recent innovations in soft robotic manipulators emphasize enhancing dexterity, control, and sensing capabilities. For instance, Majidi et al. in 2014 showcased a soft robotic manipulator capable of intricate tasks like knot tying [26], while Yang et al. in 2018 developed a manipulator integrated with sensors to discern the stiffness of objects [27]. The realm of textile-based soft robotic manipulators has also seen advancements, exemplified by the Textile Robotic Hand developed at Technische Universität München [10].

Soft robotic manipulators, with their distinct advantages over conventional rigid manipulators, are being tailored for a myriad of applications spanning medical, industrial, and underwater robotics. Breakthroughs in soft materials, sensing, and control paradigms are poised to further augment the efficacy and adaptability of soft robotic manipulators in the future [6].

The adaptability of soft robotic manipulators also makes them ideal for underwater exploration. Their ability to conform to irregular shapes allows them to grasp and manipulate delicate marine specimens without causing damage [28]. Moreover, the inherent buoyancy and flexibility of soft materials can be advantageous in underwater settings, reducing the need for complex buoyancy control systems [6].

2.4 Intersection of Robotics and Textiles

Textile robots, also known as fabric-based robots, are soft robots constructed from flexible, elastic materials and fabric-based structures. They are made of fabrics like woven, knitted, or braided materials and are extremely adjustable and versatile. Textile robotics has gained popularity in recent years due to its potential uses in healthcare, industrial automation, and wearable gadgets.

Soft robotic manipulators made of textiles have been created to improve the adaptability and flexibility of robotic manipulators. Textiles provide various benefits over standard rigid robotic manipulators, including improved flexibility, better compliance, and reduced weight. Many research have focused on the development of textile-based soft robotic manipulators with various actuation systems, such as pneumatic, hydraulic, and electromagnetic, in order to improve their dexterity and flexibility.

Researchers at the University of Bristol built the first textile-based robotic manipulators in the early 2000s [29], creating a robotic arm made completely of knitted fabric and elastomer. Since then, various textile-based robotic manipulators have been designed, including a soft robotic gripper inspired by the movement of starfish limbs produced by Harvard University researchers [30]. This gripper had a pneumatic actuation mechanism and could grab and manipulate fragile items.

Recent advances in textile-based soft robotics have also concentrated on the development of new fabrication processes, such as 3D printing and additive manufacturing, to expand the complexity and usefulness of these robots. Researchers at Carnegie Mellon University have used 3D printing to create a soft robotic tentacle capable of grabbing and moving items [31]. This tentacle was made up of numerous layers of various materials, including a flexible textile outer layer and a hard inner layer, which allowed for better dexterity and control.

The integration of sensing capabilities into textile robots has also been a focus of recent research. Embedding sensors directly into the fabric structures allows for real-time feed-

back, enabling more precise control and interaction with the environment [32]. For instance, stretchable and flexible sensors incorporated into textile robots can detect changes in pressure, temperature, and humidity, providing valuable data for various applications, from medical diagnostics to environmental monitoring [33].

Soft robotics primarily employs elastomers, gels, and other soft materials that can undergo large deformations and return to their original shape. Silicone elastomers, such as polydimethylsiloxane (PDMS), are commonly used due to their biocompatibility, optical transparency, and tunable mechanical properties [34]. Hydrogels, which can change their volume in response to environmental stimuli, are also being explored for soft robotics applications [35]. These materials can be combined with traditional fabrics or textile structures to create composite materials with tailored properties for specific applications [36].

The choice of materials plays a pivotal role in the design and functionality of soft robots. The mechanical properties of these materials dictate their behavior under various conditions:

- **Young's Modulus**: Measures the stiffness of a material. Materials with a lower Young's modulus are more flexible and adaptable.
- **Poisson's Ratio**: Influences how a material expands or contracts in response to forces.
- **Density**: Can influence the weight and buoyancy of a robot.
- **Elongation at Break**: Indicates how much a material can stretch or bend.
- **Tensile Strength**: Determines the durability and longevity of materials under external forces.

The following table provides an overview of the mechanical properties of some common materials used in soft robotics, including TPU (American Polyfilm Inc., Bran-ford, CT, DT2001, 0.15 mm thickness), and High Strength Polyester Ribbon.

Material	**Young's Modulus (MPa)**	**Poisson's Ratio**	**Density (g/cm^3)**	**Elongation at Break (%)**	**Tensile Strength (MPa)**
PDMS	0.5 - 3 [37]	0.49	0.97 - 1.03	100 - 700	2.5 - 4.5
Hydrogel	0.001 - 1 [38]	0.45 - 0.5	1.0 - 1.2	100 - 2000	0.1 - 1
TPU	1 - 5	0.4 - 0.5	1.1 - 1.2	400 - 600	25 - 50
High Strength Polyester Ribbon	6 - 12 [39]	0.35 - 0.4	1.38	7 - 10	80 - 120

Table 2.1: Mechanical properties of common soft robotic materials.

2.5 The Role and Impact of Simulations

Simulation technologies have revolutionized the field of soft robotics, enabling unprecedented insights into the behavior and potential of soft robotic manipulators. Cheng et al. (2017)

pioneered a simulation framework that has become foundational in predicting the performance of these systems across various scenarios [40]. This advancement has paved the way for diverse control strategies, from model-based control to feedback control, and even machine learning-driven control [41].

The field of soft robotics presents unique challenges for simulation due to the complex, non-linear behavior of soft materials and the intricate interactions between these materials and their environments. Traditional simulation methods often fall short in accurately predicting the performance of soft robots, necessitating the development of advanced techniques like the Finite Element Method (FEM). FEM addresses these challenges by dividing a larger system into smaller, manageable elements, allowing for more precise modeling of complex deformations and interactions.

Several software solutions, such as Abaqus and ANSYS, harness the power of FEM for simulating soft robots. These tools have evolved over time to meet the specific needs of the soft robotics community, offering adaptability for a wide range of robotic topologies, from continuum robots to pneumatic actuators and soft grippers. Notably, the OctArm, a continuum robot, has been extensively modeled using FEM, providing valuable insights that have influenced subsequent research in the field [42].The potential of abaqus in soft robotics simulation is demonstarted in multiple papers among of which is shown in figure 2.1

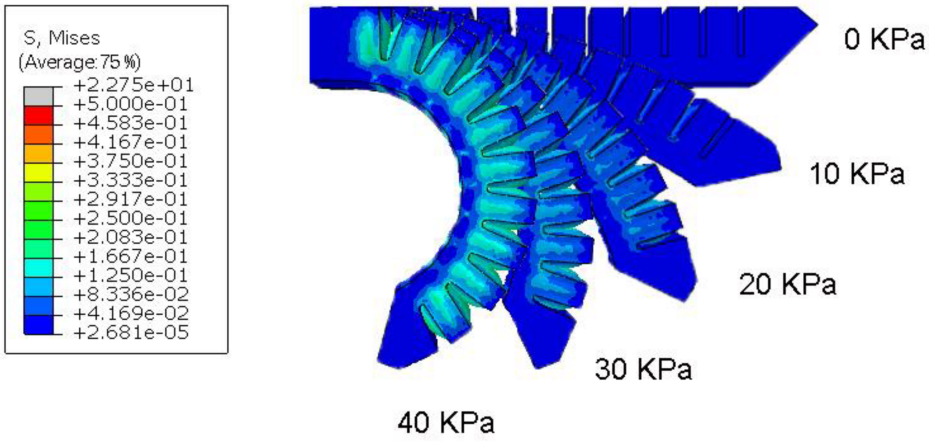


Figure 2.1: A Deformed finger through internal pressurization of the cavity within with stress contour under different air pressure values. Source:[43]

Pneumatic actuators, powered by air pressure, exemplify the complexities of soft robotics. Their behavior, encompassing deformation and force output, can be effectively modeled using FEM. Fanti et al. (2019) showcased this potential by simulating a soft robotic arm powered by pneumatic muscles, demonstrating FEM's capability to anticipate actuator behavior and inform design decisions [44].

Soft grippers, constructed from diverse materials like elastomers and hydrogels, have also benefited from FEM simulations. Research on the universal gripper and the fluidic elastomer

gripper has utilized FEM to understand their behavior. Martinez et al. (2013) employed FEM to study the deformation of the universal gripper, highlighting its utility in predicting gripper performance [14].

In the realm of soft robotics, Sun et al. explored the characterization of soft pneumatic actuators made from highly compliant elastomers, emphasizing their ease of fabrication and adaptability [45]. Marchese et al. delved into the control of soft robots using real-time finite element analysis, showcasing the potential of this method in predicting soft robot behavior [46]. Ogura et al. introduced the "Nematode Actuator", a micro rubber pneumatic actuator designed for medical applications, highlighting its safety due to low mechanical impedance [47]. Wang et al. emphasized the significance of simulations in the design and testing phase of soft robotics, particularly their role in reducing the time and cost associated with physical prototyping [48].

The Simulation Open Framework Architecture (SOFA) stands out as an open-source platform known for simulating multi-physics phenomena. Duriez highlighted SOFA's adaptability in soft robotics simulation, emphasizing its broad environment that supports diverse modeling methodologies [49]. SOFA's adaptability ensures it can cater to a wide range of research requirements, from intricate soft material interactions to complex robotic motion dynamics. Moreover, SOFA's real-time feedback capability is invaluable during iterative design and development.

Figure 2.2: Comparison of a real soft robot's motion (top) with its simulated counterpart (bottom). Source:[50].

SolidWorks, renowned for its Computer-Aided Design (CAD) capabilities, has carved a niche in the simulation domain. Its seamless interface, bridging CAD designs with FEM tools, ensures rapid testing and iteration of prototypes. This streamlined process, combined

with its ability to handle intricate geometries and its material catalog, ensures that soft robotic designs are both realistic and representative of real-world scenarios.

Simulations, undeniably, play a pivotal role in the design and testing phase of soft robotics. They offer a virtual environment where researchers can anticipate robot behavior, thus minimizing the resources associated with physical prototyping [49]. Furthermore, simulations optimize the design parameters of soft robots, ensuring their efficacy in real-world scenarios [48].

2.6 Underwater Robotic Applications and Pioneering Research

The marine environment presents numerous challenges, from the corrosive nature of saltwater to the immense pressures of the deep sea. Soft robots, with their compliant structures, can adapt to the contours of the environment and handle objects with a gentleness that rigid robots cannot replicate [51].

Reference	**Material**	**Depth**	**Actuation**	**Object**	**Findings**	**Application**
Galloway(2016)	Soft elastomers	Deep reefs	Fiber-reinforced	Whip coral	Adaptability in deep-sea	Deep-sea coral sampling
Stuart(2017)	Advanced polymers	100m	Bellows-style	Coral	Precision in delicate tasks	Marine conservation
Vogt(2018)	3D printed elastomers	300m	Bellows	Sponge	Rapid prototyping	Rapid prototyping of soft robotic manipulators
Marchese(2015)	Silicone elastomers	200m	Pneumatic	Seaweed	Dexterous manipulation	Underwater object manipulation
Katzsch.(2018)	Soft silicone	50m	Pneumatic	Fish	Autonomous navigation	Autonomous underwater exploration

Table 2.2: Overview of key studies on soft underwater manipulators.

The groundbreaking work by [28] was a significant leap in marine biology sampling. Their design utilized soft elastomers and a fiber-reinforced actuation mechanism, allowing the manipulator to adapt to the unpredictable conditions of deep-sea environments.

Building on this foundation, [52] introduced advanced polymers to the mix, creating a bellows-style actuator that could delicately handle brittle scleractinian coral. Their work underscored the potential of soft manipulators in marine conservation, emphasizing their

ability to interact with marine life without causing harm.

The innovative approach by [53] of using 3D printing techniques opened new avenues in the design and customization of soft robotic manipulators. Their rapid prototyping capabilities allowed for designs tailored to specific marine tasks, showcasing the adaptability of soft robotics.

Marchese et al. [54] demonstrated the capabilities of a soft robotic arm in underwater manipulation. Their design, made of silicone elastomers and actuated pneumatically, showcased dexterous manipulation capabilities, effectively handling objects like seaweed in underwater environments.

Katzschmann et al. [55] developed a soft robotic fish that could autonomously navigate in aquatic environments. Made of soft silicone and actuated pneumatically, this robot showcased the potential of soft robotics in autonomous underwater exploration.

Complementing these manipulator advancements, there has been a surge in the development of soft underwater sensors. These sensors, such as those developed by [56], enhance the capabilities of soft manipulators, enabling them to sense and adapt to their surroundings. The integration of tactile, pressure, and curvature sensors has allowed soft robots to perform tasks with increased precision and sensitivity.

2.7 Sensing Mechanisms in Soft Robotics

Underwater applications such as subsea exploration, pipeline monitoring, and underwater vehicle manipulators rely heavily on sensors. Unfortunately, the severe underwater environment presents various problems, such as high pressure, reduced visibility, and corrosive compounds, which might impair sensor performance. Current breakthroughs in pressure, actuation, force, and curvature sensors for underwater applications are to be addressed.

Pressure sensors are used to monitor the external water pressure or the pressure on the robot's end effector. Conventional pressure sensors, such as piezoresistive or capacitive sensors, can be utilized underwater, however due to the harsh environment, they have drawbacks. Current research has concentrated on creating novel materials and designs to increase the performance of underwater pressure sensors. Researchers, for example, created a piezoresistive pressure sensor with a graphene-based sensing element that performed better in underwater environments [57].

The motion of underwater manipulators is controlled by actuators. Hydraulic actuators, which employ pressured fluids to create force, are the most prevalent actuation sensors used in underwater robots. The density and viscosity of water can impair hydraulic actuator performance, resulting in decreased precision and efficiency. Shape memory alloy (SMA) ac-

tuators, which can provide accurate and quick motion in underwater conditions, have been developed by researchers as novel actuation methods for underwater robots [58]

Force sensors are used to measure the forces exerted to the end effector of a robot. Underwater, traditional force sensors such as strain gauges can be utilized, but their accuracy and sensitivity can be impacted by the qualities of the water. Researchers created new underwater force sensors, such as a fiber Bragg grating (FBG) sensor that demonstrated great accuracy and sensitivity in underwater conditions [59].

Underwater manipulator arms' curvature is measured using curvature sensors. The curvature of the manipulator arm can impact the robot's motion accuracy and precision. Many curvature sensors for underwater applications have been created by researchers, including a fiber-optic curvature sensor that demonstrated exceptional accuracy and sensitivity in underwater conditions [60].

Another challenge in underwater sensing is the biofouling, where marine organisms adhere to the sensors, affecting their performance. Anti-fouling coatings and materials are being researched to mitigate this issue [61].

Tactile sensing is also crucial for underwater soft robots, especially when interacting with delicate marine life or handling fragile objects. Recent advancements in soft tactile sensors, such as hydrogel-based sensors, offer high sensitivity and flexibility, making them suitable for underwater applications [62].

Chapter 3

Mechanical Modeling

3.1 Introduction: Drawing Insights from Fossen

The marine environment is a complex domain with a distinct set of challenges that set it apart from the terrestrial and aerial realms. The density, viscosity, inhomogeneity, and unpredictability of the medium owing to waves, currents, and turbulence make undersea dynamics exceptionally complex [63]. While air and space vehicles are primarily concerned with gravitational and aerodynamic forces, marine vessels, particularly underwater manipulators, must contend with a plethora of hydrodynamic forces and moments that affect their performance.

For decades, marine hydrodynamics has been researched, with numerous academics striving to decipher the complicated interactions between fluid dynamics and solid entities immersed inside them. The application of these ideas to autonomous underwater vehicles (AUVs), remotely operated vehicles (ROVs), and underwater manipulators has become critical in the current day. Considering the tremendous possibilities and dangers of activities under the ocean's surface, the need for proper modeling and management cannot be overstated.

Fossen's work stands as a cornerstone in this domain. His comprehensive examination of marine craft hydrodynamics offers both theoretical insights and practical applications that address the challenges marine engineers face today. By leveraging the insights from Fossen's research, as others like Sivčev et al. [64] did, we aim to construct robust models that can withstand real-world underwater conditions, ensuring the safety, effectiveness, and efficiency of maritime operations.

3.1.1 Hydrostatics

Hydrostatic forces are concerned with the vertical forces and moments acting on the manipulator as a result of buoyancy and weight distribution, and they play an important role in the balance and stability of submerged objects. In layman's terms, these forces govern the equilibrium between an underwater manipulator's natural tendency to sink (due to its weight) and its propensity to rise (due to the displaced water volume). This equilibrium is essential for both surface and underwater marine vehicles, affecting the manipulator's energy

consumption, operating stability, and response to external disturbances.

The hydrostatic forces (F_b) and moments (M_b) arise from buoyancy and weight distribution:

$$F_b = V\,\rho_{\text{fluid}}\,g \tag{3.1}$$
$$M_b = (x_g - x_b)F_b \tag{3.2}$$

where V is the volume of the submerged object, x_g is the center of gravity, and x_b is the center of buoyancy.

For an underwater manipulator or any submerged object to be in stable equilibrium without any rotational tendencies, the center of gravity x_g and the center of buoyancy x_b should be vertically aligned. If they are not aligned, a moment or torque will be generated due to the difference in the vertical positions of these two points, causing the object to rotate.

In practical applications, it's often desirable to adjust the position of the center of buoyancy to achieve a specific orientation or stability characteristic. One common method to displace x_b is by adding buoyant objects or materials to the system. By strategically placing these buoyant additions, one can manipulate the location of x_b to achieve the desired alignment with x_g.

3.1.2 Added Mass

The concept of added mass is fundamental to the dynamics of undersea systems. When an underwater manipulator or any submerged body is set in motion, it must overcome both its own inertia and the inertia of the surrounding fluid that it displaces. This phenomenon is analogous to 'carrying' a piece of the water with it while moving. This 'extra' inertia, defined as added mass, increases the system's effective mass, which alters its dynamical behavior.

The impact of added mass is mathematically expressed by the relationship:

$$F_a = -M_A\,\ddot{\eta} \tag{3.3}$$

where M_A is the added mass matrix and $\ddot{\eta}$ is the acceleration vector.

The added mass matrix M_A typically takes on the form:

$$M_A = \begin{bmatrix} m_{11} & 0 & 0 & 0 & m_{15} & m_{16} \\ 0 & m_{22} & 0 & -m_{15} & 0 & m_{26} \\ 0 & 0 & m_{33} & -m_{16} & -m_{26} & 0 \\ 0 & -m_{15} & -m_{16} & m_{44} & 0 & 0 \\ m_{15} & 0 & -m_{26} & 0 & m_{55} & 0 \\ m_{16} & m_{26} & 0 & 0 & 0 & m_{66} \end{bmatrix} \tag{3.4}$$

For illustration, let's consider the derivation of the term m_{11}. This term represents the added mass in the surge (forward) direction. It can be obtained by integrating the pressure distribution over the submerged surface of the body when it is forced to move in the surge direction while all other degrees of freedom are restrained. Mathematically, this can be represented as:

$$m_{11} = \int_S \rho \left(\frac{\partial \phi}{\partial x} \right)^2 dS \tag{3.5}$$

where ρ is the fluid density, S is the submerged surface, and ϕ is the velocity potential.

However, deriving the entire added mass matrix using integration over the body's surface is computationally expensive and may not always yield accurate results, especially for complex structures or flow conditions. This is because theoretical methods are often based on simplified assumptions about the shape of the structure and the flow conditions. In contrast, experimental methods can be used to measure hydrodynamic parameters directly, regardless of the complexity of the structure. Furthermore, experimental results can be used to validate theoretical models and CFD simulations. As a result, experimental results are always better than theoretical results for complex designs [65].

3.1.3 Hydrodynamic Damping and Coefficients

The performance and efficiency of underwater manipulators are substantially influenced by hydrodynamic forces, especially drag (D) and lift (L). As a manipulator moves or alters its configuration, it interacts with the surrounding fluid, generating forces and moments that resist its movement. Such resistance forces are majorly influenced by the manipulator's relative velocity with the fluid, impacting the manipulator's efficiency, positioning precision, and overall controllability.

The forces due to hydrodynamic damping can be represented as:

$$F_d = -D\,\eta \tag{3.6}$$

where D is the damping matrix and η is the velocity vector of the manipulator.

Both lift and drag forces are affected by the manipulator's shape, speed, fluid properties, and other factors. They can be expressed as:

$$D = 0.5\,\rho\,A_{\text{ref}}\,C_D\,v^2 \tag{3.7}$$

$$L = 0.5\,\rho\,A_{\text{ref}}\,C_L\,v^2 \tag{3.8}$$

where:

- ρ is the fluid density, variable with salinity, temperature, and pressure.
- A_{ref} is the reference or frontal area of the object.

- C_D and C_L are the drag and lift coefficients respectively, encapsulating effects of geometry, surface roughness, and flow conditions.
- v is the velocity of the manipulator relative to the fluid.

The drag force arises primarily from two mechanisms: viscous drag, resulting from frictional resistance between fluid layers moving over the object, and pressure (or form) drag, due to pressure differences at the front and rear of the object, often intensified by flow separation and turbulence.

The drag coefficient C_D plays a vital role in hydrodynamic calculations. It can vary based on the object's geometry and prevailing flow conditions. Methods to determine C_D include experimental measurements in wind or water tunnels, where drag forces are directly measured across varied flow speeds, and Computational Fluid Dynamics (CFD) simulations, which solve the Navier-Stokes equations numerically to predict the flow around the object. Nevertheless, CFD simulations need validation against experimental data to ensure reliability.

Concluding this section, the insights drawn from Fossen's comprehensive work are fundamental in understanding the interactions and influences of hydrodynamic forces on underwater manipulators. This knowledge is essential for developing accurate models and control strategies to ensure optimal performance of underwater systems in a diverse and unpredictable marine environment [63].

3.2 Flexion Actuator

Drawing from Fossen's comprehensive insights on marine craft hydrodynamics, the aim is to accurately model our underwater manipulator by understanding and incorporating relevant hydrodynamic parameters. In this section, we study a flexion actuator with particular geometric properties as outlined by Nassour et al[10]. they considered an actuator comprised of two interacting segments. The contact force $\vec{F}$ between the two pressurized segments depends on the internal pressure P and was calculated as follows:

$$\vec{F} = \pi l P \left(\frac{r^2 - a^2}{2r} \right). \tag{3.9}$$

Where l is the length, r is the radius, and a is the distance between the center of the segment and the contact point.. this force is responsible for the actuator's bending motion. The pressure inside the actuator segments works against the contact area to produce a torque τ which is applied at the housing fabric:

$$\tau = \pi l P \left(\frac{r^2 - a^2}{2} \right). \tag{3.10}$$

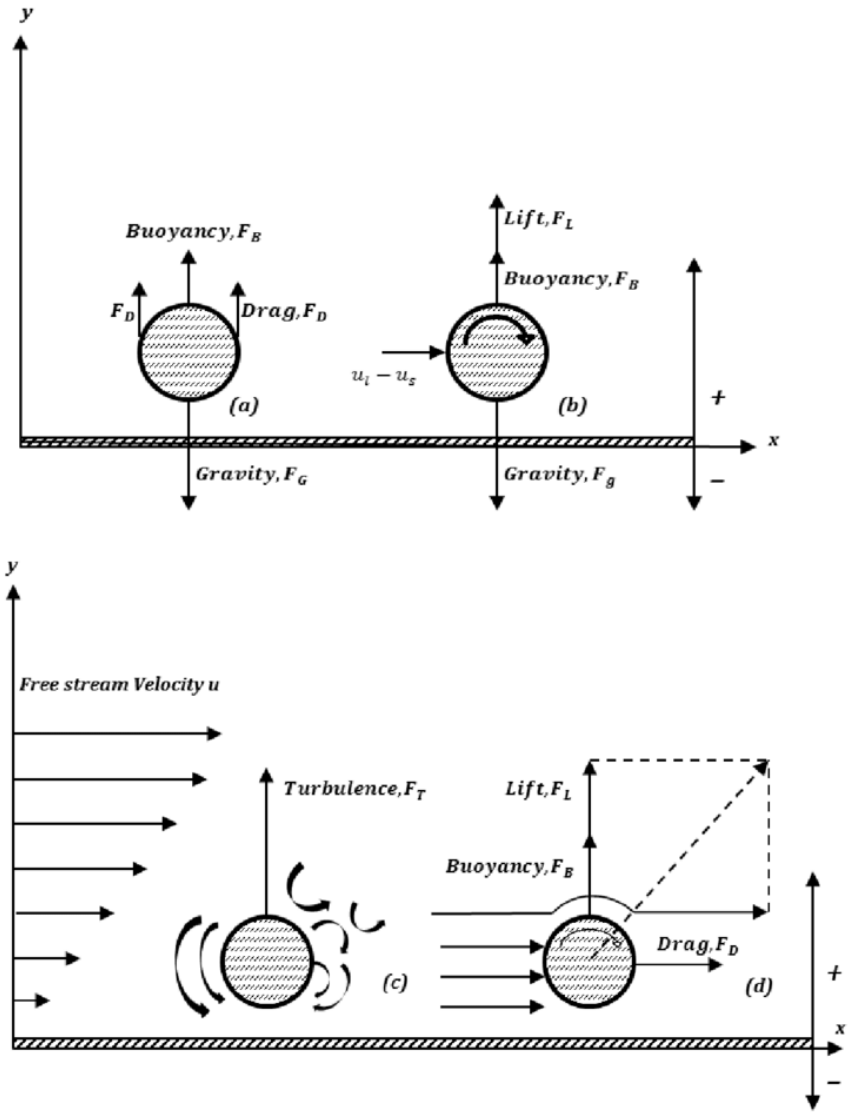


Figure 3.1: Illustration of the Hydrodynamic forces acting on a submerged spherical body (a) The instantaneous balance of gravity, buoyancy and drag forces, (b) the resulting lifting force, c) turbulent diffusive forces through eddies, d) the direction of the suspension force vector resulting from these forces. Source:[66]

Finally, the flexion of the actuator can be described in terms of the angle β.The contact angle β is the angle between the normal to the contact area between the two segments of the flexion actuator and the line connecting the centers of the two segments. As the actuator bends, the contact angle changes, which in turn affects the contact force and torque. This angle is connected to the actuator angle θ through the equations:

$$\beta = \arcsin\left(a/(r + r_c)\right) \tag{3.11}$$

$$a = \left(r + \frac{nw}{2\pi - \theta}\right) \sin\left(\frac{2\pi - \theta}{2n}\right). \tag{3.12}$$

In this equation, n represents the number of segments, and w stands for the width of a single segment, which is demonstrated in 3.2. The contact angle is important because it determines the direction of the contact force. The contact force is always perpendicular to the contact area, so the direction of the contact force changes as the contact angle changes. This change in the direction of the contact force is what causes the actuator to bend.

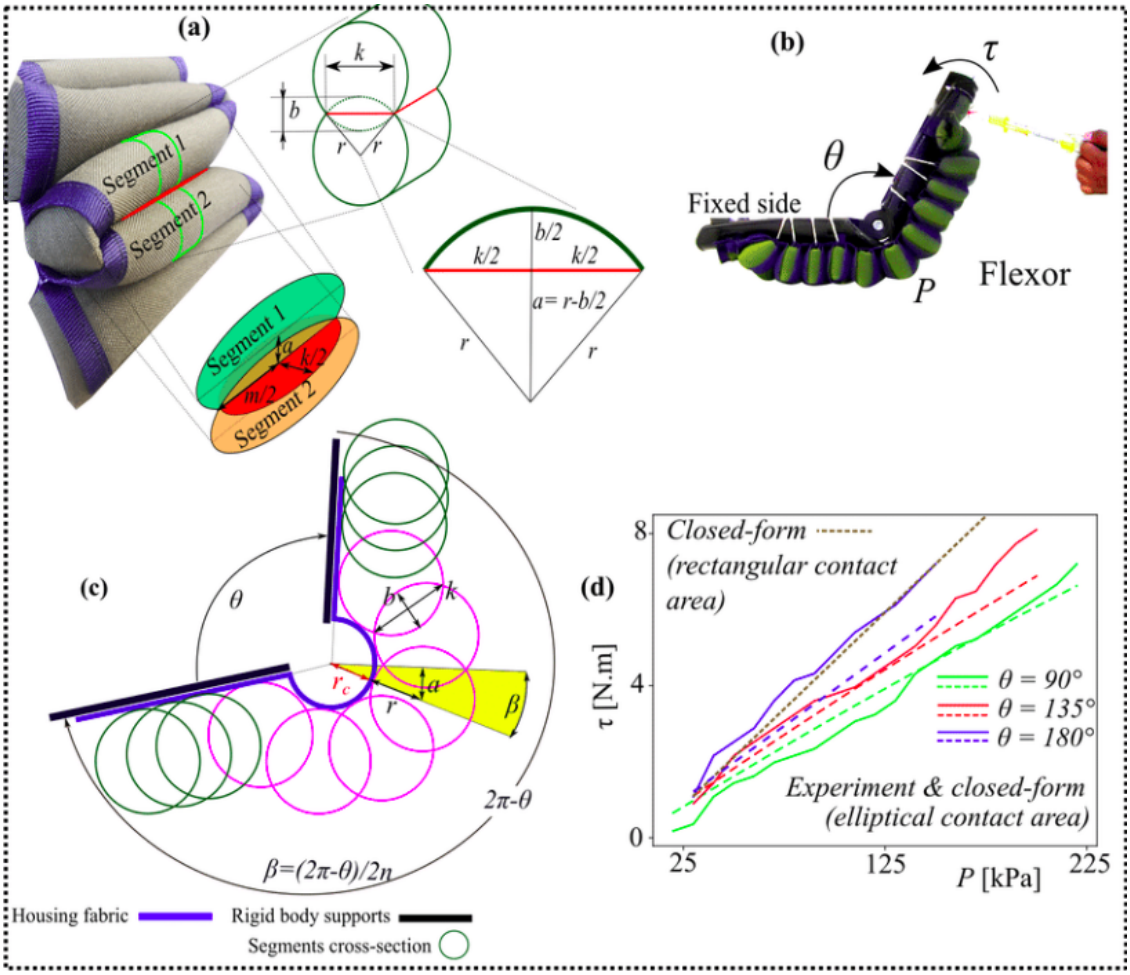


Figure 3.2: (a) Cross-section of the flexor actuator, showing the elliptical contact area. (b) Experimental setup for measuring the torque of the flexor.(c) of the flexor cross-section attached to two rigid links. (d) Model and experimental data for the relationship between pressure and torque for the flexor actuator at three different angles, source:[10]

3.3 Extension Actuator

Now to the paper consideration of an extension actuator, which will re-extend the curved part after flexsion. The actuator's performance can be described in terms of the torque τ it can generate. The torque depends on the actuator's geometric parameters, such as its radius r and length l, the deflection angle ρ, and the internal air pressure P. The expression for

the torque is as follows:

$$\tau = \frac{\pi r^3 P l}{l - 2r\tan(\rho/2)} \left(\tan^2\left(\frac{\rho}{2}\right) + 1\right). \tag{3.13}$$

The term $\tan^2(\rho/2) + 1$ accounts for the variation in torque with different bending angles.

3.4 Hydrostatic Pressure

Hydrostatic pressure affects the internal pressure of the extension actuator, which in turn affects the actuator's torque output. This happens because the actuator's input is exposed to the water, and the hydrostatic pressure pushes against the input, increasing the internal pressure. When underwater, the actuator encounters hydrostatic pressure due to immersion. While water is often represented as incompressible, it has a relatively small compressibility, which leads to a non-linear increase in hydrostatic pressure with depth, especially at large depths. This effect is best characterized by the hydrostatic pressure equation, given by:

$$P_{\text{underwater}} = P_{\text{surface}} + \rho_{Water}\, g\, h, \tag{3.14}$$

where P_{surface} represents the pressure at the water surface, g stands for gravitational acceleration, ρ_{Water} is the density of water and h is the depth of the actuator beneath the water surface.

The hydrostatic pressure affects the internal pressure of the actuator, especially when the actuator's input is exposed to the water. This is expressed by:

$$P_{\text{underwater}} = \begin{bmatrix} P + \rho_{Water}\, g\, h & 0 & 0 \\ 0 & P + \rho_{Water}\, g\, h & 0 \\ 0 & 0 & P + \rho_{Water}\, g\, h \end{bmatrix} \tag{3.15}$$

The hydrostatic pressure is considered as an external pressure in our actuator model, superimposed on the internal pressure of the actuator. This technique makes sense given the hydraulic actuation. This is because the hydrostatic pressure from the surrounding water would change the internal pressure of the actuator, especially when the actuator's input is exposed to the water.

The use of a 3x3 diagonal matrix is intentional. It highlights the consistency of the influence of hydrostatic pressure along the major axes (x, y, and z). The pressure operates evenly in all directions at every location inside the fluid, regardless of its orientation. Because hydrostatic pressure isotropy is a fundamental feature of fluids in equilibrium, it is crucial to show and comprehend in this context. For the torque equations presented previously for the flexion and extension actuators, every occurrence of the term P should be substituted with $P_{\text{underwater}}$ to account for the hydrostatic pressure effects.

3.5 Added Mass Matrix

The consequences of added mass become critical in understanding the hydrodynamics of a robotic actuator approximated as a rectangular prism submerged in water as represented in figure 3.3. This is because that are at the bottom is joined with another layer of High Strength Polyester Ribbon as a stiff layer stopping it from extending on both ends. that is while the folds on the top are compact and covered in a mesh.

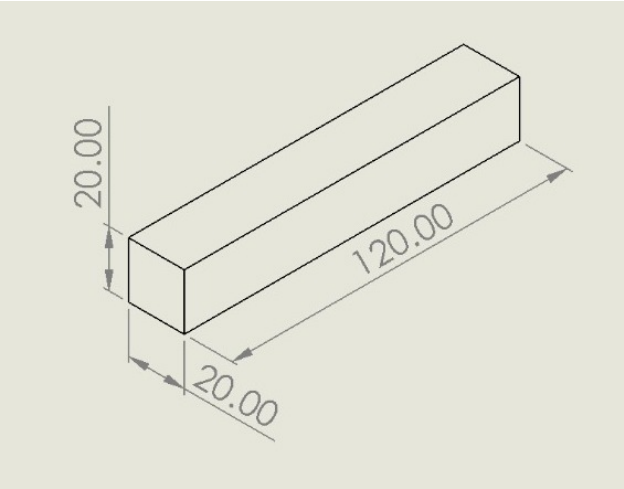


Figure 3.3: the assumed geometry of the actuator (millimeters)

The increased mass represents the fluid's inertia caused by the robot's motion. Given the reduced form of the actuator, its low rotational rates and operating speeds, and the actuator's major actuation reliance on internal pressure changes, the off-diagonal (coupling) factors in the additional mass matrix may be completely ignored. This assumption is based on the fact that such terms, which represent cross-coupled dynamic effects, become relevant only at higher speeds or in bigger structures, and they remain low for our actuator working under the provided conditions.

Since our concern here is the torque, the focus will be on the inertial terms of the added mass matrix. The added mass matrix I_A can then be written as:

$$M_A = \begin{bmatrix} m_{44} & m_{45} & m_{46} \\ m_{54} & m_{55} & m_{56} \\ m_{64} & m_{65} & m_{66} \end{bmatrix}, \tag{3.16}$$

To account for the added mass effects in the torque and force calculations, the force due to added mass F_a can be determined by:

$$F_a = -M_A \alpha \tag{3.17}$$

where α is the angular acceleration vector. This force, particularly its rotational components, will influence the overall torque depending on their point of application and orientation, making the torque due to the added mass:

$$\tau_{\text{added mass}} = -M_A \, \alpha \, r \tag{3.18}$$

3.6 Internal and External Friction

The operation of the actuator experiences both internal and external frictions. Internal friction arises from the inter-layer interactions of the actuator's material, whereas the external friction stems from the actuator's interaction with its surroundings.

Internal friction is characterized by a coefficient μ_i, which depends on properties of the actuator material, such as elasticity and surface roughness. External friction, represented by μ_e, varies based on the environment in which the actuator operates (e.g., air vs. underwater).

Using Coulomb's law of friction, the frictional force F_{fr} can be expressed as the product of the friction coefficient μ and the normal force F_N:

$$F_{fr} = \mu F_N \tag{3.19}$$

Consequently, the torque contributions from internal and external frictions, τ_i and τ_e, can be defined as:

$$\tau_i = \mu_i F_{N,i} r \tag{3.20}$$

$$\tau_e = \mu_e F_{N,e} r \tag{3.21}$$

where $F_{N,i}$ and $F_{N,e}$ refer to the normal forces resulting from internal and external interactions respectively.

3.7 Water Resistance

Water resistance introduces an additional torque component. Hence, the work exerted by the actuator serves both the bending and overcoming this resistive torque.

While the actuator is not rotating in a traditional sense, we can use the concept of "effective rotational dynamics" to describe the bending motion and the associated drag forces.This concept is introduced to provide a framework for understanding the bending motion in terms of familiar rotational dynamics. This is analogous to describing the linear velocity of a point on a rotating wheel, even if the actuator itself isn't rotating.

Defining the torque due to water resistance as $\tau_{\text{resistance}}$, it modifies the original torque equation as:

$$\tau_{\text{underwater}} = \tau_{\text{basic}} + \tau_{\text{resistance}} \tag{3.22}$$

Assuming a drag force proportional to the velocity squared (as is standard in fluid dynamics), the resistive torque can be modelled as:

$$\tau_{\text{resistive}} = F_{\text{drag}}\, r \tag{3.23}$$

Where F_{drag} is the drag force. Incorporating this into the underwater torque equation:

$$\tau_{\text{underwater}} = \frac{\pi r^3 P_{\text{underwater}}(\tan^2\rho + 1)}{l - 2r\tan(\rho/2)} + F_{\text{drag}}\, r \tag{3.24}$$

Drag force in a fluid medium is:

$$F_{\text{drag}} = 0.5\, C_D\, \rho\, A\, v^2 \tag{3.25}$$

Characterized by:

- F_{drag} - Drag force.
- C_D - Drag coefficient.
- ρ - Fluid density.
- A - Cross-sectional area of the finger.
- v - Object's velocity.

For a bending actuator, the effective linear velocity $v_{\text{effective}}$ of a point due to the change in bending angle $\frac{d\theta}{dt}$ is:

$$v_{\text{effective}} = r_{\text{effective}} \frac{d\theta}{dt} \tag{3.26}$$

Where $r_{\text{effective}}$ is the distance between the center of rotation and the point where the effective linear velocity is being calculated, which differs along the length and is why the integration will be introduced. Substituting this into the drag force equation:

$$F_{\text{drag}} = \int_0^l 0.5\, C_D\, \rho\, A\, \left(r_{\text{effective}} \frac{d\theta}{dt}\right)^2 dr \tag{3.27}$$

The torque $\tau_{\text{resistance}}$ due to this drag force at a distance r is:

$$\tau_{\text{resistance}} = F_{\text{drag}}\, r_{\text{effective}} \tag{3.28}$$

substituting the value for F_{drag} in $\tau_{\text{resistance}}$ equation, results in:

$$\tau_{\text{resistance}} = \int_0^l 0.5\, C_D\, \rho\, A\, r^3\, \left(\frac{d\theta}{dt}\right)^2 dr \tag{3.29}$$

3.8 Buoyancy Force

When submerged in water, the actuator experiences an upward buoyancy force that offsets the weight of the actuator. This buoyancy force (F_{buoyancy}) can be calculated as:

$$F_{\text{buoyancy}} = \begin{bmatrix} 0 \\ 0 \\ \rho_{\text{Water}}\, g\, V \end{bmatrix} \tag{3.30}$$

where V is the volume of the actuator.

The upward buoyancy force is a result of Archimedes' principle, which states that an object submerged in a fluid experiences an upward force equal to the weight of the fluid displaced by the object. In mathematical terms, this force can be expressed as:

$$F_{\text{buoyancy}} = \rho_{\text{Water}}\, g\, V \tag{3.31}$$

Where:

- ρ_{Water} is the density of the water.
- g is the acceleration due to gravity.
- V is the volume of the actuator submerged in the water.

Given that the buoyancy force acts vertically upwards, the force vector representation in a 3D Cartesian coordinate system becomes:

$$F_{\text{buoyancy}} = \begin{bmatrix} 0 \\ 0 \\ \rho_{\text{Water}}\, g\, V \end{bmatrix} \tag{3.32}$$

resulting in the torque

$$\tau_{\text{buoyancy}} = F_{\text{buoyancy}}\, r \tag{3.33}$$

3.9 Comprehensive Torque Model for the Underwater Actuator

To derive a complete torque equation for the actuator in underwater environments, we need to factor in contributions from flexion-extension, hydrostatic pressure, added mass, water resistance (drag), and internal and external friction.

Starting with the basic torque for flexion-extension,Incorporating the hydrostatic pressure:

$$P_{\text{combined}} = P + \rho_{\text{Water}}\, g\, h \tag{3.34}$$

Which leads to:

$$\tau_{\text{hydro}} = \frac{\pi r^3 P_{\text{combined}}(\tan^2 \rho + 1)}{L - 2r \tan(\rho/2)} \tag{3.35}$$

Factoring in the torque due to water resistance (drag), the effects of internal and external friction and given the added mass effect the comprehensive torque for the actuator underwater is:

$$\tau_{\text{total}} = \tau_{\text{hydro}} + \tau_{\text{drag}} + \tau_i + \tau_e + \tau_{\text{added mass}+\tau_{\text{buoyancy}}} \tag{3.36}$$

This equation provides a complete representation of the net torque experienced by the actuator when submerged, encompassing all the discussed effects. And can be used to simulate the actuator's dynamics and predict its performance under various operating conditions.

3.10 Sensitivity Analysis

The sensitivity analysis provides an understanding of how changes in various parameters of the actuator affect its performance. In our context, the main parameters of interest are the radius r, the length L, and the internal pressure P, and we aim to gauge how variations in these parameters influence the output torque τ.

3.10.1 Sensitivity with respect to Radius

To determine the sensitivity of the torque τ to the radius r, we compute the partial derivative:

$$S_r = \frac{\partial \tau}{\partial r}. \tag{3.37}$$

The magnitude of S_r will indicate the extent to which torque is affected by changes in r. A larger absolute value of S_r suggests that the actuator's torque is significantly influenced by the radius.

3.10.2 Sensitivity with respect to Length

Similarly, the sensitivity of the torque to the length l is given by:

$$S_L = \frac{\partial \tau}{\partial l}. \tag{3.38}$$

A substantial absolute value of S_l denotes that the torque is quite sensitive to alterations in the length of the actuator.

3.10.3 Sensitivity with respect to Internal Pressure

Lastly, for the internal pressure P, the sensitivity is computed as:

$$S_P = \frac{\partial \tau}{\partial P}. \tag{3.39}$$

An increased absolute value of S_P implies that even minor changes in internal pressure can lead to significant variations in the output torque.

By understanding these sensitivities, one can make informed decisions on actuator design and operational parameters, ensuring optimal performance while being aware of factors that can significantly influence output.

3.11 Assumptions and Limitations

The above model is underpinned by various assumptions, which must be recognized when interpreting its outcomes:

- **Geometric Assumptions**: The actuator is modeled as a rectangular prism as demonstarted in figure 3.3, maintaining a consistent cross-section throughout. Manufacturing variances or operational deformations might deviate from this model.
- **Material Assumptions**: A homogeneity in the actuator's material is assumed, suggesting consistent mechanical properties. This might not reflect actual variations in properties like elasticity or roughness.
- **Friction Assumptions**: Coulomb's law of friction is employed to represent both internal and external friction. In real-world scenarios, frictional forces might vary depending on movement speed or other factors.
- **Fluid Dynamics Assumptions**: The model might oversimplify fluid interactions, assuming laminar flow and neglecting effects like turbulence or complex viscous interactions.
- **Temperature Effects**: Variations in temperature and their subsequent effects on materials or fluid behavior were not be incorporated in the model.

Moreover, while theoretical models provide important insights and guidelines, real-world performance often contains nuances not fully captured by theory alone. As a result, it is critical to supplement these analytical conclusions with practical experiments and real-world observations in order to have a complete and accurate knowledge of the actuator's behavior.

3.12 Lagrangian Analysis: Curvature and Osculating Circle

The importance of understanding curvature in underwater robotics dynamics arises due to the inherent flexibility and the need to adapt to the surrounding aquatic environment, which can present challenges not observed in terrestrial robotics.

The curvature of a curve at a given segment is a measure of how sharply the curve is bending

or turning at that segment. To quantify this, we introduce the concept of an *osculating circle*. The osculating circle at a segment on the curve is the circle that best approximates the curve near it. In other words, it's the circle that touches the curve at that segment without crossing it, and thus, it shares the same tangent with the curve at it's points.

Given this, the curvature κ of the curve at that segment is defined as the reciprocal of the radius r of the osculating circle. Thus, we have:

$$\kappa = \frac{1}{r} \tag{3.40}$$

For a small segment dx of the bar, if the curvature is κ, the segment changes its orientation by an angle:

$$d\theta = \kappa \, dx \tag{3.41}$$

where dx captures the relationship between the change in curvature and the resultant change in orientation for the segment.

The rate at which this orientation change occurs is represented by the angular velocity ω:

$$\omega = \frac{d\theta}{dt} \tag{3.42}$$

Combining the above equations, we get:

$$\omega = dx \frac{\kappa}{dt} \tag{3.43}$$

Integrating over the length of the bar will yield the total change in orientation due to curvature changes, giving a relationship between the overall curvature change and angular velocity.

The kinetic energy of the flexible bar, integrated over its length due to its intrinsic motion, is:

$$T = \int_0^l \frac{1}{2} \mu \left(\frac{d\theta}{dt} \right)^2 dx \tag{3.44}$$

where μ is the linear mass density of the bar.

In flexible underwater robotic systems, the structural flexibility allows the bar to adapt its shape according to external forces and its own dynamics. This flexibility means that different parts of the bar may assume different orientations and depths. To account for the gravitational potential energy associated with each segment of the bar, we integrate over its length.

Consider the flexible bar bending such that its curvature is represented by $\kappa(x)$, where x is the position along the bar's length. As the bar bends, the depth h of any segment will vary based on its curvature. This variation can be subtle, but it's important to consider for precision. Hence, h can be thought of as a function of κ, i.e., $h = h(\kappa)$.

Now, considering the context of our specific system, the length of the prism that represents the bar is 120 mm. This implies that when the prism is vertical in the water and devoid of any curvature, the difference in depth h between its top and bottom will be a maximum of 0.12 meters. While this difference might seem negligible, it could influence the system's dynamics, especially in precision applications. However, the significance of this variation can be evaluated further in subsequent analysis.

The gravitational potential energy for an infinitesimal segment dx of the bar is $m\,g\,h\,dx$. To find the total potential energy for the entire bar, we sum up (integrate) these infinitesimal energies over the whole length:

$$U = \int_0^l m\,g\,h(\kappa(x))\,dx \tag{3.45}$$

This integral ensures that we account for the energy contributions from all parts of the bar, considering its changing depth due to bending.

Understanding drag becomes especially important in underwater environments. The interaction of the robotic bar with the water leads to dissipating forces that can alter its motion. The drag force on a curved object is perpendicular to the surface of the object and in the opposite direction to the flow of the fluid. For a rectangular prism that is curving along its length and forming a circle, the drag force will be perpendicular to the curved surface of the prism and in the opposite direction to the motion of the prism. Typically, drag forces introduce a non-linear damping effect in the dynamics, which is predominantly influenced by the square of the velocity. Let's define A as the effective area of the segment exposed to the fluid and C_D as the drag coefficient. Both these factors play a critical role in quantifying the energy dissipated due to drag. Note that since the shape is changing while κ is changing, this in return affects C_D, making the drag coefficient a function of the curvature.

Now, to account for the energy dissipated due to drag, the drag force F_{drag} can be represented as:

$$F_{\text{drag}} = \int_0^l \frac{1}{2}\,\rho\,A\,C_D(\kappa)\,\dot{\kappa}^2\,dx \tag{3.46}$$

Consequently, our system's Lagrangian L is:

$$L = T - U \tag{3.47}$$

With this , the Euler-Lagrange equation for the system becomes:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\kappa}}\right) - \frac{\partial L}{\partial \kappa} = Q_D + Q \tag{3.48}$$

Where:

- L is the Lagrangian function, accounting for kinetic energy, potential energy.

- Q stands for the generalized forces acting on the system.
- Q_D represents the generalized non-conservative or dissipative forces acting on the system, such as those due to drag.

3.12.1 Choice of Generalized Coordinate

In Lagrangian mechanics, the appropriate selection of generalized coordinates is essential to adequately capture the system's motion. For our underwater flexible bar, which is described by its curvature, the natural choice for a generalized coordinate is the curvature itself, κ.

Curvature (κ):
The curvature, κ, directly measures the curvature of the bar. As the bar changes its shape in response to external forces, the curvature provides a concise representation of its configuration. By observing variations in κ, we can discern the extent of its change in shape. The choice of curvature as a scalar quantity streamlines the mathematical formulation and offers an intuitive understanding of the system's dynamics.

Rate of Change of Curvature ($\dot{\kappa}$):
The rate of change of curvature, denoted by $\dot{\kappa}$, gives insights into how quickly the bar is adjusting its shape. This rate is especially vital in underwater environments, where external factors can influence the dynamics. A controlled rate of change ensures the system can reach its desired configuration without unintended behaviors.

Choosing κ as focal parameters allows us to comprehensively describe the system's dynamics. This choice aligns with the system's inherent behavior and will be instrumental in formulating subsequent models and control strategies.

Additionally, since the manipulator is mounted on a remotely operated vehicle (ROV), the translation and rotations of the manipulator are not as important as the curvature. This is because the ROV can be used to control the position and orientation of the manipulator base, which will in turn control the position and orientation of the end effector. Therefore, we can focus on controlling the curvature of the manipulator to achieve the desired motion of the end effector.

3.12.2 Buoyancy in the Lagrangian

The buoyant force, resulting from the manipulator's displacement of water, contributes to the potential energy term of the Lagrangian acting in opposite direction of gravity. For an infinitesimal segment dx of the underwater manipulator at a certain depth h, the potential energy increment due to buoyancy, dU_B, is given by:

$$dU_B = -\rho g V(\kappa(x)) h(x) dx \tag{3.49}$$

Where $V(\kappa(x))$ denotes the volume of the segment, which can be a function of its curvature κ. As the manipulator changes its shape, the displaced volume V for each segment changes. To

compute the total potential energy due to buoyancy over the entire length of the manipulator, we integrate the above expression:

$$U_B = \int_0^l -\rho g V(\kappa(x)) h(x)\, dx \tag{3.50}$$

This buoyant potential energy U_B is then combined with other potential energy components to give the total potential energy U for the manipulator:

$$U = U_{\text{gravity}} + U_B \tag{3.51}$$

This integrated term significantly influences the total potential energy as the manipulator moves deeper or adjusts its curvature, thereby impacting the dynamics derived from the Euler-Lagrange equation.

3.12.3 Added Mass

The added mass effect results from the manipulator curving in the fluid environment. As the manipulator changes its curvature, it's not only propelling itself but also a portion of the surrounding water. This effect essentially introduces additional inertia due to the acceleration or deceleration of the adjacent water.

Assuming the manipulator is made of a uniform material, its intrinsic mass distribution m remains constant over its length. Hence, we do not need to express m as a function of x.

For an infinitesimal segment dx of the manipulator with a curvature $\kappa(x)$, the associated added mass coefficient dm_A can be represented as:

$$dm_A = \rho \int_V \frac{\partial \phi}{\partial x} dV \tag{3.52}$$

Where V is the volume of the finger. Integrating this over the length of the manipulator, we obtain:

$$m_A = \int_0^l dm_A\, dx \tag{3.53}$$

In our dynamics formulation, the added mass effect contributes an additional inertial term in the kinetic energy component of the Lagrangian. The kinetic energy T of the manipulator, accounting for both its intrinsic mass and the added mass, becomes:

$$T = \frac{1}{2} \int_0^l m \dot{\kappa}^2 dx + \frac{1}{2} \int_0^l m_A(\kappa(x)) \dot{\kappa}^2 dx \tag{3.54}$$

This kinetic energy, influenced by the curvature of the manipulator in the fluid, alters the effective inertia of the system. Such a change, in turn, affects the Lagrangian, leading to modifications in the equations of motion as per the Euler-Lagrange equation.

3.12.4 Assumptions for the Added Mass

Our analysis assumes the following:

1. **Uniform Material:** The manipulator is constructed from a material with a consistent mass distribution over its length, leading to a constant intrinsic mass m.
2. **Direction of Motion:** The added mass effects are primarily attributed to the curvature of the manipulator in the fluid.
3. **Experimental Data:** In the absence of direct experimental measurements for the added mass, we rely on theoretical frameworks like potential flow theory.

Considering these assumptions, the added mass $m_A(\kappa(x))$ for our system can be ascertained. This integrated value plays a pivotal role in comprehending the dynamic behavior of the manipulator as it maneuvers in water.

3.12.5 Resulting Lagrangian

By consolidating the kinetic and potential energy terms derived from the manipulator's interaction with the fluid, the Lagrangian L is expressed as:

$$L = T - U = \int_0^l \frac{1}{2}\dot{\kappa}(M(\kappa) + M_A(\kappa))\dot{\kappa}\, dx + \int_0^l \rho g V(\kappa) h\, dx - \int_0^l m\, g\, h(\kappa(x))\, dx \tag{3.55}$$

Where: - The integration is over the length of the manipulator. - $M(\kappa)$ represents the intrinsic mass matrix of the manipulator. - $M_A(\kappa)$ denotes the added mass matrix, both dependent on the curvature κ. - The term $\rho g V(\kappa) h$ corresponds to the potential energy due to buoyancy.

Considering the drag force F_{drag} arising from drag, the associated generalized non-conservative force becomes:

$$Q_D = -\frac{\partial F_{\text{drag}}}{\partial \dot{\kappa}} \tag{3.56}$$

Incorporating this into the Euler-Lagrange formalism, the equation of motion is:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\kappa}}\right) - \frac{\partial L}{\partial \kappa} = Q + Q_D \tag{3.57}$$

Here, Q denotes the generalized forces from other external influences, such as actuator forces.

3.12.6 Solving the Lagrangian

The resultant Lagrangian equations provide a mathematical representation of the manipulator's dynamics. They can be used to derive the equations of motion for the system, which can then be used to simulate and control the system's behavior. To deduce the system's behavior:

1. Determine the boundary conditions pertinent to the manipulator's operation, typically involving initial positions and velocities.

2. Employ numerical techniques, like Runge-Kutta methods, to find solutions to the differential equations. The presence of nonlinear terms mandates the use of such methods.

3. Scrutinize the results to grasp the dynamic response of the manipulator in its aquatic setting, subsequently guiding potential refinements in design or control strategy.

This analytical framework, rooted in Lagrangian mechanics, is instrumental in optimizing the manipulator's performance in underwater applications.

Chapter 4

Simulation and Analysis of CAD Software

4.1 Abaqus Integration

Abaqus, a renowned finite element analysis software, was the primary choice for this research due to its frequent selection in the soft robotics community. Its capability to handle high displacements made it particularly suitable for simulating the behavior of soft robotics. The insights from Abaqus simulations provided a deep understanding of the structural dynamics and stress distributions within the design [67]. However, the research faced limitations with the educational version of Abaqus, which restricts simulations to a maximum of 1000 nodes. This constraint allowed for the simulation of only a singular fold of the pneumatic textile robotic finger. Due to these institutional constraints and limited access to the full version of Abaqus, the research shifted its focus towards other simulation platforms. Nevertheless, ABAQUS's free educational version was employed to demonstrate its potential capabilities.

4.1.1 Comparative Analysis

A comprehensive comparative analysis was performed between Abaqus and other simulation platforms. While Abaqus is a common choice in soft robotics research due to its suitability for high displacements, SolidWorks is less prevalent in this domain. The research's own tests with SolidWorks confirmed its limitations, especially when dealing with large displacements typical in soft robotics. This analysis was still instrumental in optimizing the robotic finger design and ensuring its reliability and performance under various conditions in terms of stress analysis [68, 69].

4.1.2 Adaptations and Improvements

The detailed simulations conducted in Abaqus informed subsequent design modifications and enhancements. This iterative process was integral to achieving a robust and efficient design, capable of meeting the diverse requirements of soft robotics applications [70].

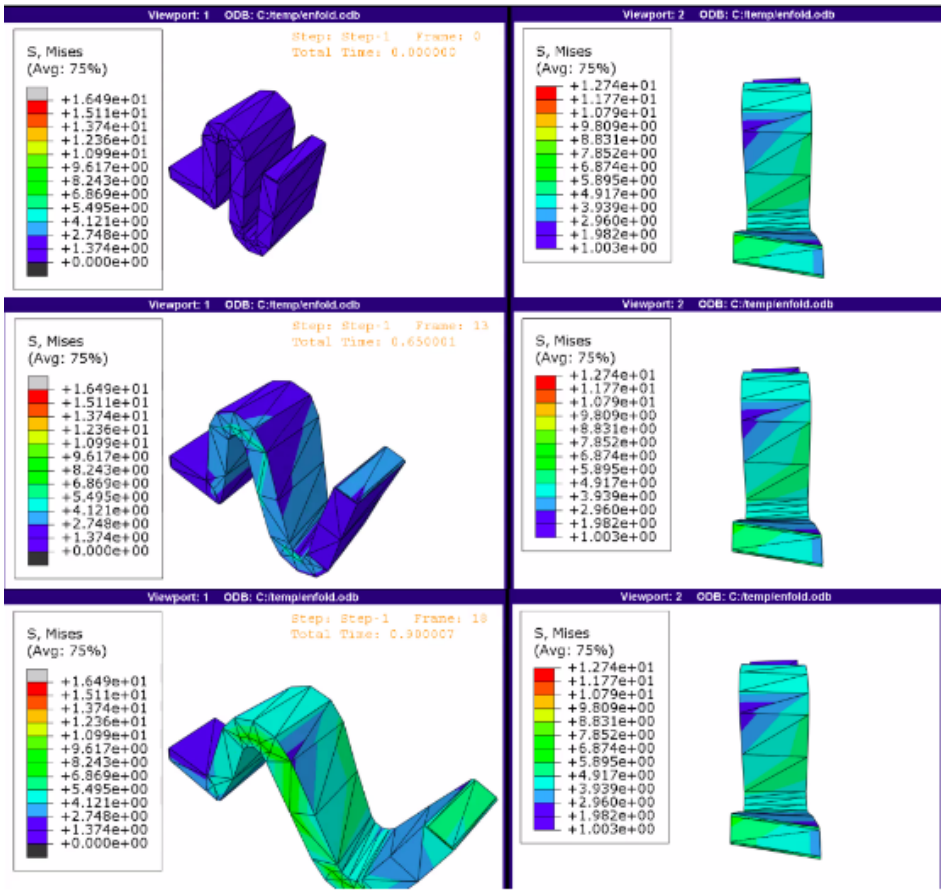


Figure 4.1: Simulation result showing the effective extension and movement of the fold in Abaqus.

4.1.3 SolidWorks Behaviour in similar designs

In the extensive simulation testing conducted, various load scenarios and material properties were examined to accurately depict the behavior of soft robotic structures. Efforts to make the material highly elastic involved adjusting the Poisson's ratio and reducing the elastic modulus, based on estimates from MatWeb [71]. Even though the full software capabilities were utilized, optimizing the mesh proved to be essential. Several mesh densities were tested, from coarser to more refined structures. Furthermore, a 'large displacement' simulation technique was implemented, with the anticipation that it would better represent the intricate dynamics of the inflation process. However, despite these thorough modifications and adjustments, the subsequent analysis showed the folds inflating with merely a slight expansion motion. This outcome suggests that accurately representing the real-world behavior of these structures might necessitate more intricate simulation parameters or alternative tools. Although powerful, SolidWorks has limits in the exact simulation of soft, inflated objects, notably in the correct portrayal of massive displacements found in such materials. As a result, in order to get a comprehensive knowledge of the motion and control dynamics of the planned robotic finger, supplemental simulation settings beyond the analytical constraints of SolidWorks were required. [22, 12].

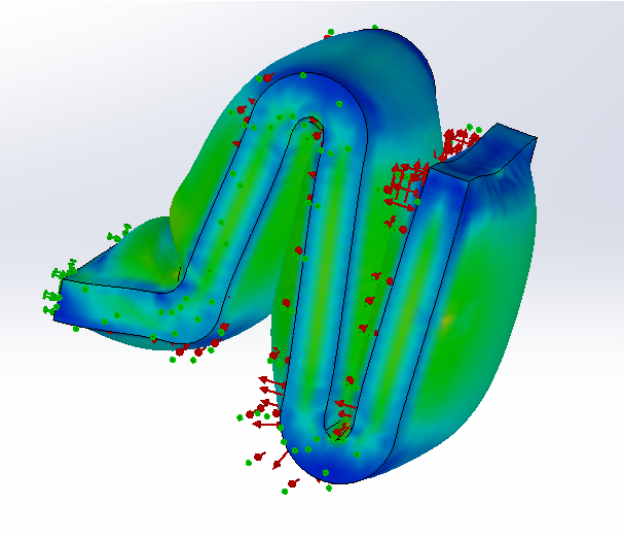


Figure 4.2: SolidWorks Simulation result on the unfolding and extension of one fold when subjected to an internal pressure increase

4.1.4 Comparative Analysis

ABAQUS and SolidWorks utilize distinct simulation algorithms, which lead to their differential performance in the realm of soft robotics. ABAQUS is renowned for its advanced nonlinear analysis capabilities, making it particularly adept at modeling intricate material responses, substantial deformations, and complex contact scenarios [72]. This is crucial for soft robotics, where materials often undergo large deformations and exhibit nonlinear behaviors.

Several reasons contribute to ABAQUS's superiority in this domain:

- Implicit Solver**: ABAQUS employs an implicit method, which, while computationally intensive, is more stable and accurate for problems involving large deformations and non-linearities. This method can handle the intricate dynamics of soft materials, ensuring that the simulation captures the real-world behavior of these structures.
- Advanced Material Modeling**: ABAQUS offers a wide range of material models, including those specifically designed for hyperelastic and viscoelastic materials commonly used in soft robotics. This allows for a more accurate representation of the material's behavior under various loading conditions.
- Robust Contact Mechanics**: Soft robotics often involves intricate contact scenarios, both internally (between different parts of the robot) and externally (with the environment). ABAQUS's sophisticated contact algorithms ensure that these interactions are modeled with high fidelity.

In contrast, SolidWorks Simulation is primarily designed for linear static assessments in design engineering. While it does possess nonlinear functionalities, they are not as advanced or robust as those in ABAQUS [69]. Its explicit method, although faster, might not capture

the detailed dynamics of soft robotic behaviors, especially under large deformations.

Furthermore, the quality and nature of the mesh, as well as the nuances of the solver, play a significant role in the accuracy of the simulation outcomes. In nonlinear simulations, where solution convergence can be sensitive to mesh attributes, the advanced meshing capabilities of ABAQUS provide an edge. The observed discrepancy between ABAQUS and SolidWorks in capturing soft robotic behaviors can be attributed to these factors, even when SolidWorks employs a denser mesh.

4.2 Part Design

Given the constraints with accessing the full version of Abaqus, this research adopted a bifurcated approach: utilizing SolidWorks for the design phase and the SOFA framework for simulation purposes.

4.2.1 SolidWorks

SolidWorks, a prominent CAD (Computer-Aided Design) software, was employed for the initial design and development of the pneumatic textile robotic finger [73]. The software's comprehensive toolset and features were leveraged to create a detailed and accurate design.

Design Considerations and Stress Analysis: The first part of the simulation to be conducted was to model the part as per the original design of the flexor. the exact model was used for the dimensions while making the folds more spaced out and fewer in number. this will affect the end result by reducing how curved the part will be. such that if the part are more compact the radius of the circle will decrease. and increasing the number of the folds increases the length of the curved circle. This can easily be implemented but the following number of folds which was 10 was taken in order to reduce the computational requirements in sofa.

The design of the soft textile robot was based on the paper design as in figure 4.3.

In order to replicate this design, three SolidWorks part were made consisting of the housing, the folds and the cavity which will be needed in SOFA later on. Note the square edges were made in order to reduce the nodes in the mesh and therefore the tetrahedrons later on.

A preliminary stress analysis was conducted in SolidWorks, incorporating estimated material properties of the chosen combination of High Strength Polyester Ribbon and TPU. While the exact values for Poisson's ratio and elastic properties were not available, estimates sourced from MatWeb were used for the simulation of the model [71]. This method, was still however, unable to replicate the expected motion of the part.

The curvature at the beginning of the finger can be observed, however the part was not curving along it length as expected. In addition to that, the amount of pressure applied was

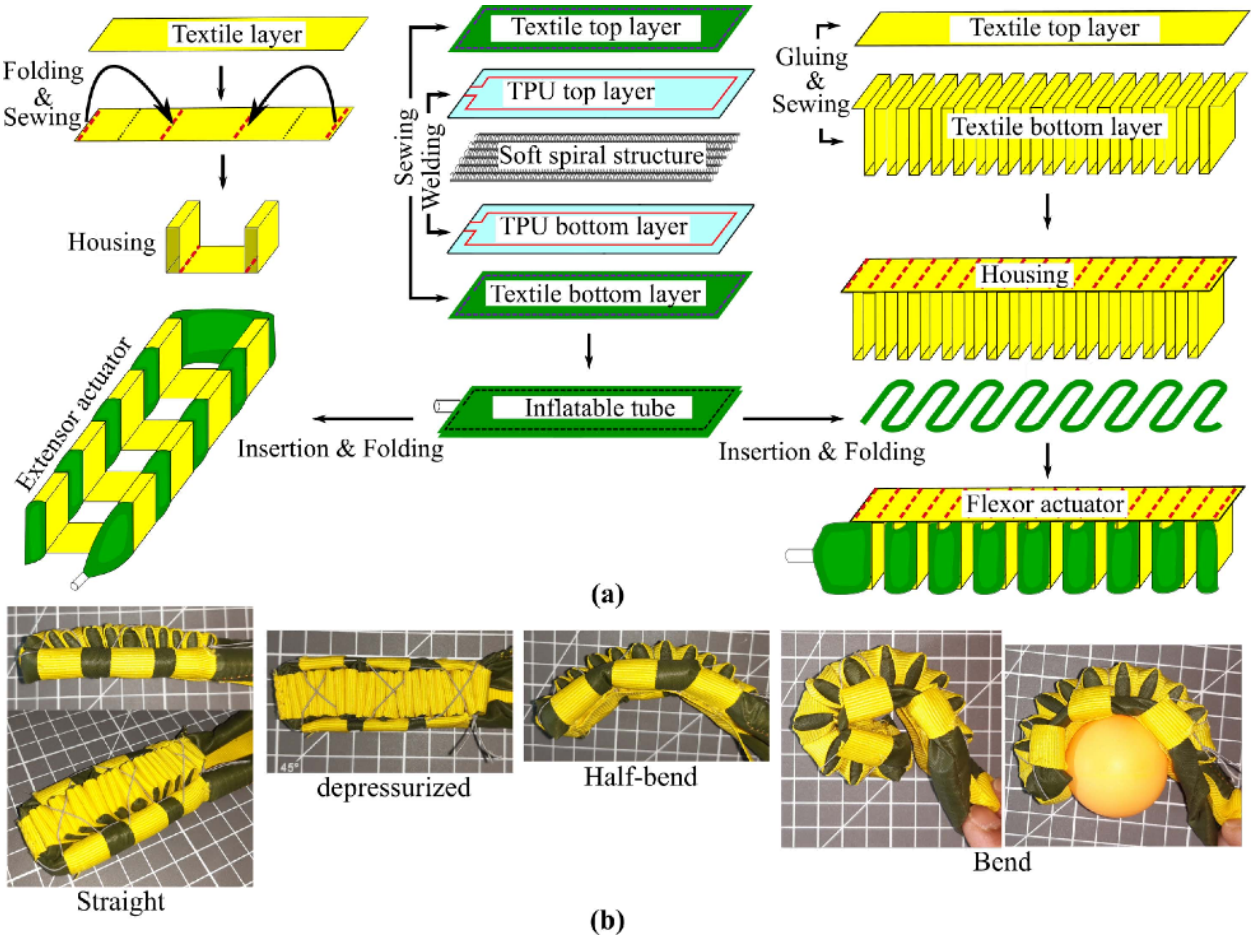


Figure 4.3: The fabrication process of the soft textile actuator. (a) Illustrations of tubes (middle), housing fabrication for extensor (left) and flexor (right). (b) The actuator in four different stages. Source :[74]

a fraction of that which the part undergoes in reality. However, increasing the value of the internal pressure resulted in errors in the simulation given the assumption that the part was failing.

4.3 SOFA Framework

The Simulation Open Framework Architecture (SOFA) is a versatile open-source platform tailored for real-time simulation of physical systems [75].Originally developed with a strong emphasis on medical simulations [76], SOFA has since found applications in a myriad of domains, including soft robotics. Its modular architecture allows for the seamless integration of various techniques, models, or plugins, such as the SoftRobots plugin,which has been instrumental in advancing soft robotics research [77, 78]. SOFA operates on a component-based approach, where each component represents a specific functionality or algorithm. This modular design ensures that simulations can be easily customized to fit specific research requirements. The components interact with each other through a well-defined interface,

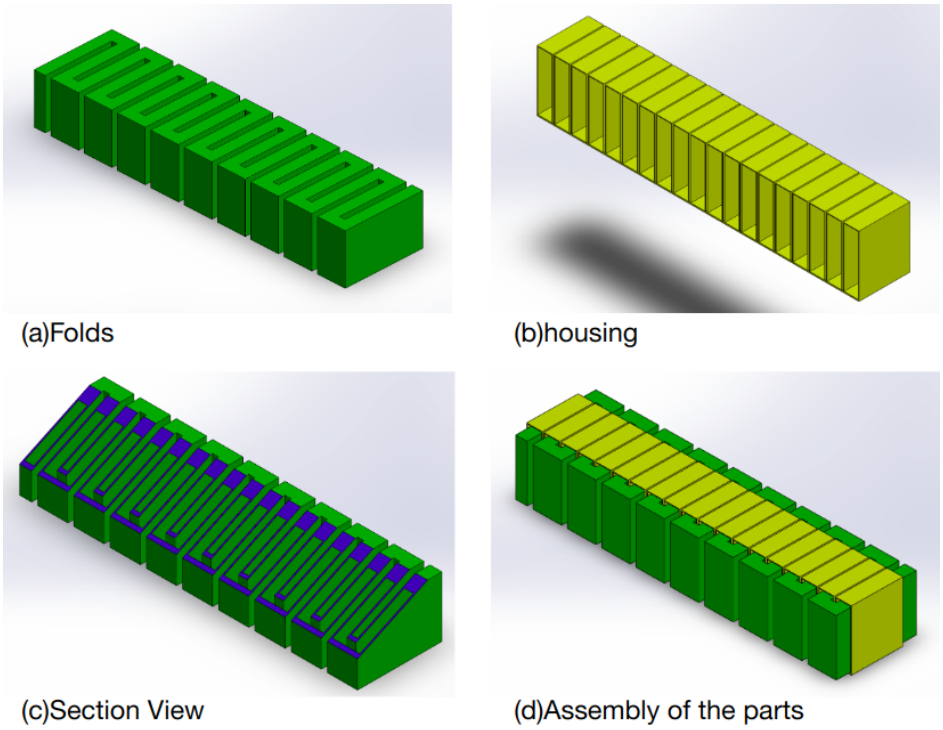


Figure 4.4: The assembly of the Housing and the folds in SolidWorks. (a)The folds of the actuator that are to be inflated as designed in SolidWorks. (b)Housing of the folds part in SolidWorks.(c)Section view of the folds showing the cavity in SolidWorks. the blue surface is the intersection of the part and the section plane demonstrating the section cap.(d)The assembly of the Housing and the folds in SolidWorks.

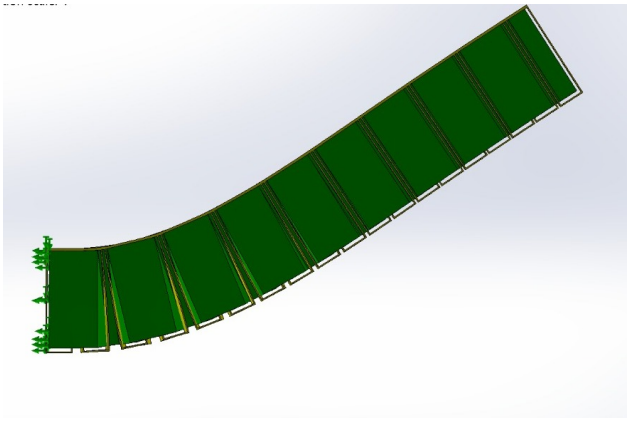


Figure 4.5: SolidWorks simulation of the pressurization of the internal cavity to a value of 0.001 MPa.

ensuring flexibility and scalability.

SOFA's Advantages in Soft Robotic Simulations:

- *Accurate Representations:* SOFA's advanced computational techniques ensure precise modeling of complex soft material behaviors, a crucial aspect in soft robotics where

deformations are intricate and highly nonlinear [79].

- *Interactivity and Real-time Feedback:* The real-time processing capabilities of SOFA facilitate immediate feedback, an advantage during the prototype phase of robotic systems where swift design iterations are often necessary.
- *Extensibility:* The framework's architecture is inherently designed to be extensible. This allows custom plugins, like the SoftRobots plugin, to be effortlessly integrated, ensuring SOFA remains adaptive to evolving research needs [75].
- *Comprehensive Measurement Tools:* SOFA offers a plethora of tools to accurately measure various parameters, be it the force exerted by a robotic limb, material deformation under stress, or the distribution of stress across a design.

PneuNet Gripper: A Pivotal Integration:

The PneuNet gripper operates by inflating elastomeric chambers, termed PneuNets. As these chambers inflate, the actuator undergoes bending. The dynamics of this motion can be intricately controlled by adjusting the chamber's design and the inflation pressure [80].

Integrating the PneuNet gripper design with the SOFA framework brings forth several advantages:

- *Precise Motion Control:* The bending dynamics of the PneuNet actuator can be meticulously simulated, ensuring its performance mirrors that in real-world scenarios.
- *Safety and Efficiency:* Before deploying actual prototypes, virtual simulations in SOFA can highlight potential failure points or inefficiencies, guiding subsequent design refinements.

The PneuNet gripper example provided with the plugin as simulated in SOFA, as shown in Figure 4.6 offers offers several advantages, including motion control, while providing a simulation platform for soft pneumatically actuated robots.

4.4 Formating

The documentation for the PneuNet gripper did not provide a detailed procedure for the conversion process. Specifically, it lacked instructions on preventing the meshing of the cavity within the part, which led to simulation failures. The only guidance provided was: "To be able to simulate the soft robot, the first step is to discretise the soft robot in space, by creating a volumetric mesh, typically with tetrahedra. This can be done with any meshing tool such as Gmsh or CGAL." Given this limited guidance, a specific workflow was established to generate a vtk format for the finger that would prevent meshing of the cavity. The finger's design was initially exported from SolidWorks as an STEP214 file. The subsequent steps, each leveraging a specific software tool, were as follows:

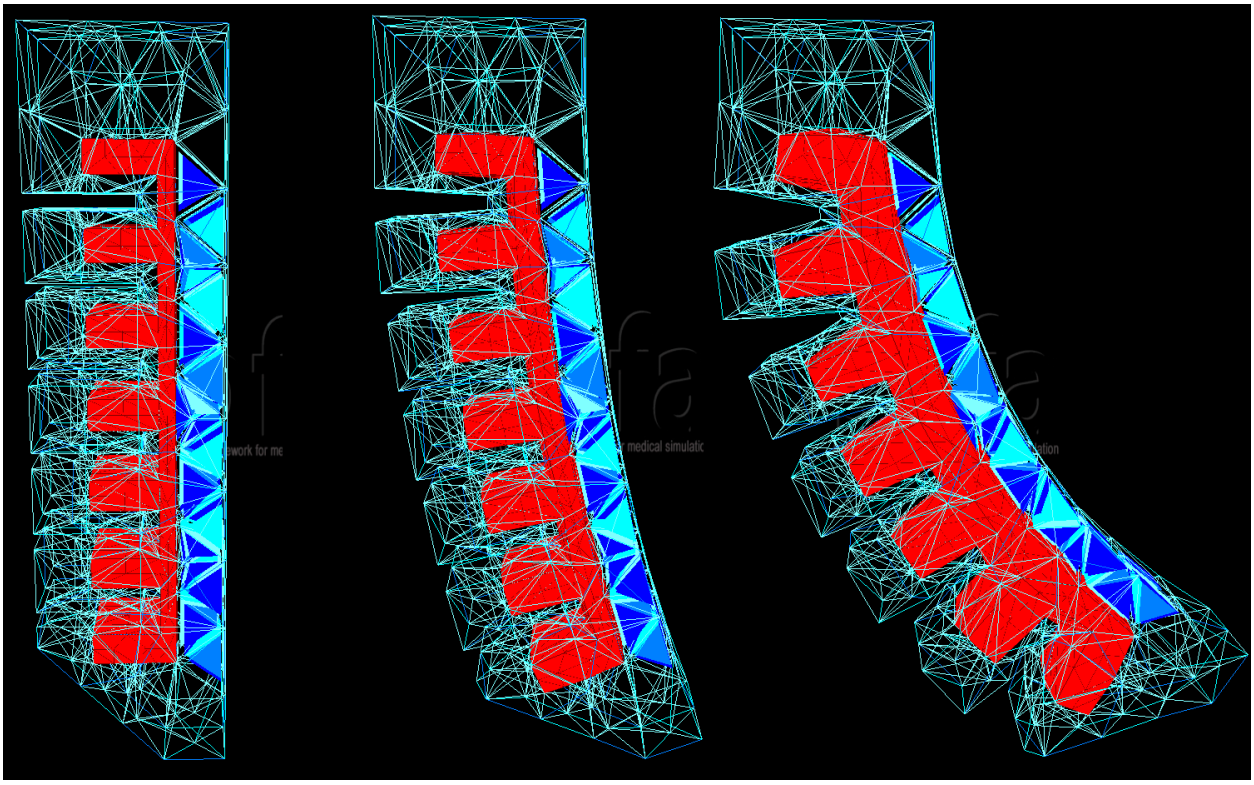


Figure 4.6: A provided example of a PneuNet actuator acting as a finger and curving while inflating in SOFA's SoftRobotics plugin. Source:[81]

FreeCAD:

FreeCAD, an open-source platform adept at handling complex geometries [82], was employed. Upon importing the STEP214 file into FreeCAD, a mesh was generated. This mesh was subsequently converted into either an stl file for the cavity, or a step fild for the rest of the part. This conversion to STEP, though seemingly redundant, was crucial. Directly produced STEP files from SolidWorks had presented issues, necessitating this intermediary step. Furthermore, this process ensured that the cavity and the external part aligned perfectly during the conversion.

NetGen:

Netgen, renowned for its capability to produce high-quality tetrahedral meshes suitable for finite element analysis [83], was the next tool in the workflow. Upon importing the STEP file into Netgen, the mesh creation process was initiated, and the result was saved as a gmsh file.

Gmsh:

The final conversion step involved Gmsh, a prominent 3D finite element grid generator [84]. Gmsh is not only proficient in handling intricate 3D geometries but also facilitates the conversion between various file formats. This made the transition from gmsh to vtk seamless and efficient.

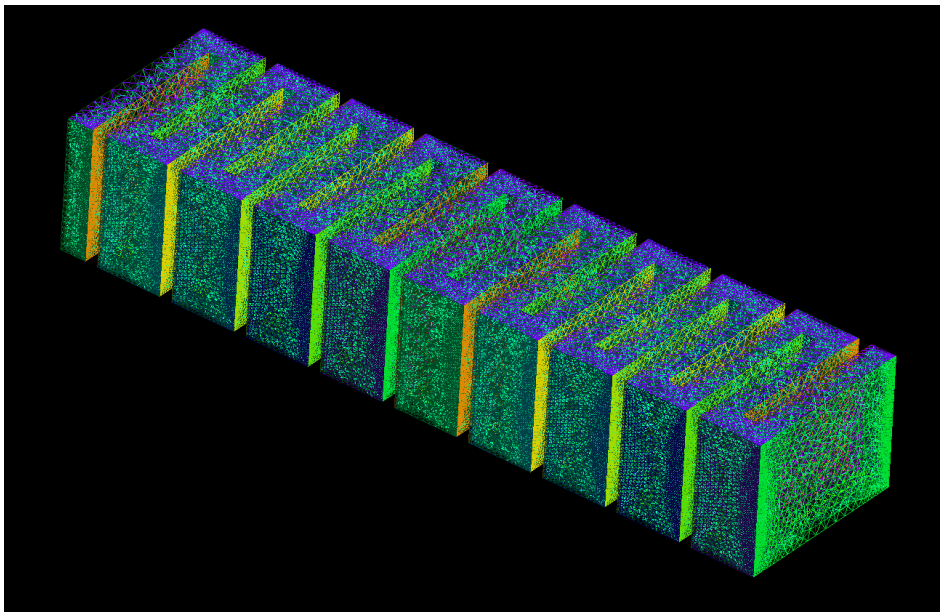


Figure 4.7: Resulting Gmsh meshed part of the folds

4.5 Resulting Models on SOFA

In order to design a more complex system, The first part was designed while taking inspiration from the shape provided in the example of the plugin 4.6 and ensure that the code can replicate the provided motion on the format conversion used above. after that the model was slowly morphed into a design more similar to the design of the TRH. first be making loops, then disjoining the common line among the loops and turning into one line, and finally by having the constrain line that is represented by the stiff layer into the housing part of the design.

The finger actuator model in SOFA is constructed using a hierarchical node-based approach. The primary node, termed as the 'finger', represents the main body of the actuator. Within this node, the geometry and mechanical properties of the finger are defined. A significant feature utilized in the model is the 'BoxROI', which defines a specific region of interest within the finger's geometry using a bounding box with specific coordinates. This is particularly useful for applying localized forces or constraints, in this case it is a constraint to one end of the finger and the exterior. The finger's geometry is loaded from a VTK file, providing

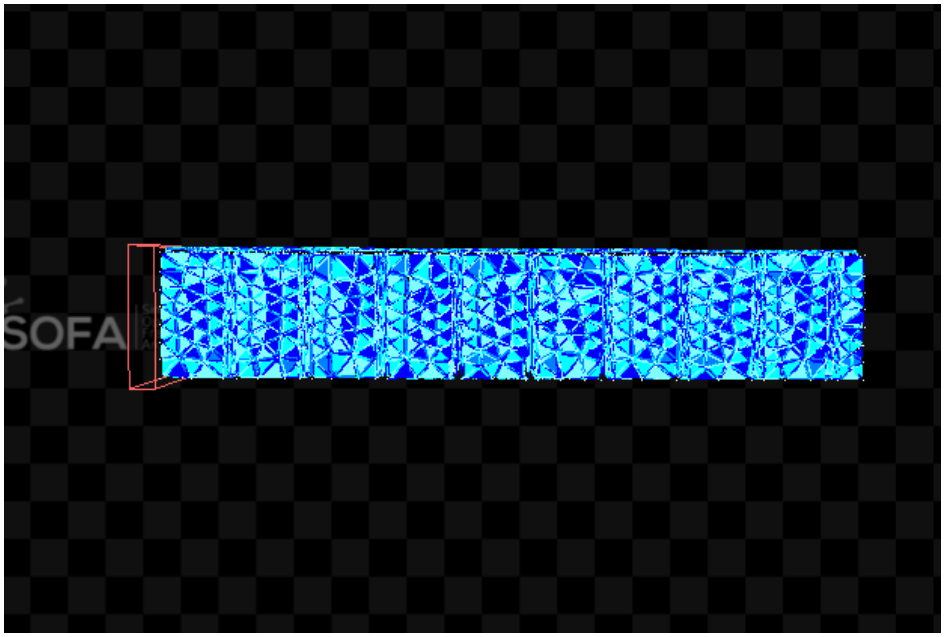


Figure 4.8: The finger model in SOFA before pressurization

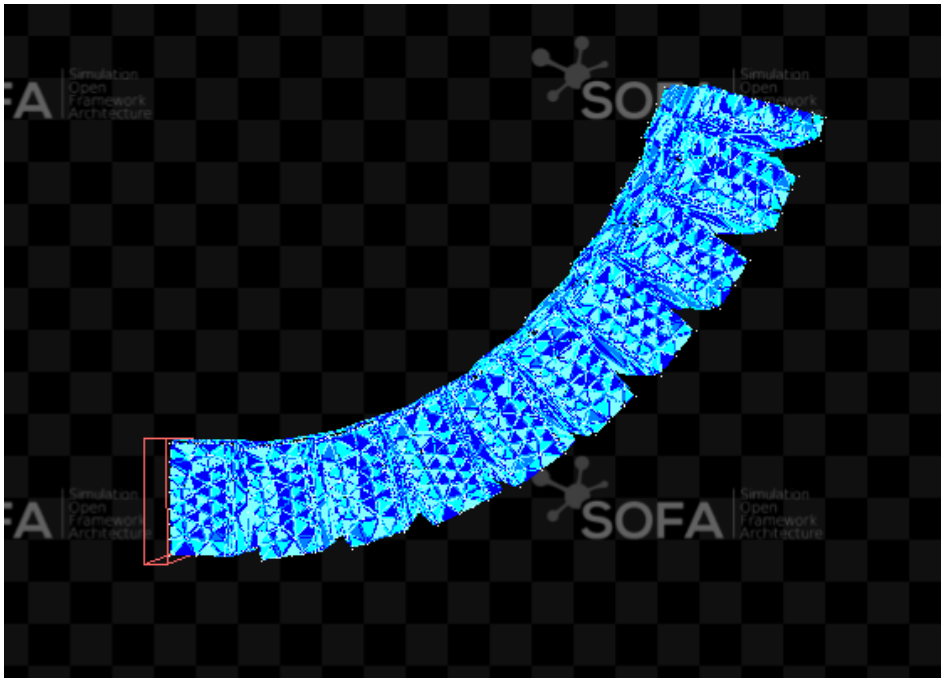


Figure 4.9: The finger model in SOFA curved after pressurization

a mesh representation composed of tetrahedral elements, often referred to as 'tetras'. These tetrahedra are crucial for the Finite Element Method (FEM) simulations, with the 'TetrahedronFEMForceField' object defining their elastic properties, including a Young's modulus of 2500 kPa and a Poisson's ratio of 0.3. Within the 'cavity' child node of the finger, a 'SurfacePressureConstraint' is applied to simulate the effects of internal or external pressures on the

actuator, distributed across the surface of the actuator using the triangles of the cavity mesh.

It is also important to note the lack of the extensor in the design. which in this case was not needed as the pressurization and depressurization was possible on SOFA. so the simulated part can go back to its initial state by simply depressurizing it. In addition to that, along the way, multiple altercations have been applied such that the SOFA model will be able to perform the calculations among the tetras without issue. Crashing was a prominent problem in the case where the mesh was fine due to an increased number of nodes and thus, calculation requirements. in order to solve this problem, the curves of the folds were represented with square edges rather than arches, and the figure 4.10 shows the prominent increase in nodes just by making this altercation. however, the results were not altered by this modification and held true to the expected results.

The pressurization process was occurring due to the FingerController code provided in the end. And it is important to cap the value of the pressure increase to a point that is realistic and can be withstood. During testing of the real model, the part could take up to 2 Bars before the connection of the finger with the pump had problems. and although this was not a failure of the finger itself, it was still taken into consideration and the pressure was caped at 1.5 Bars.

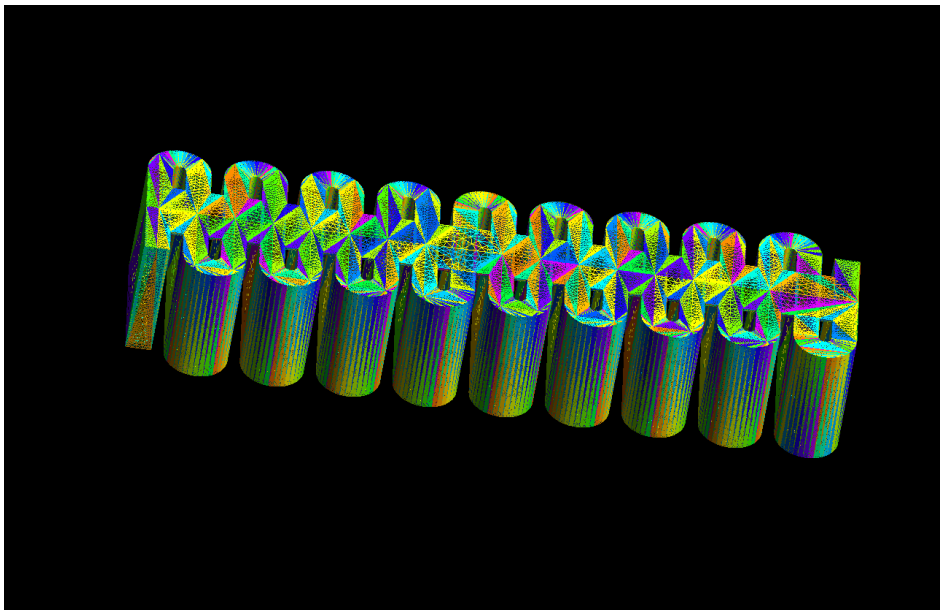
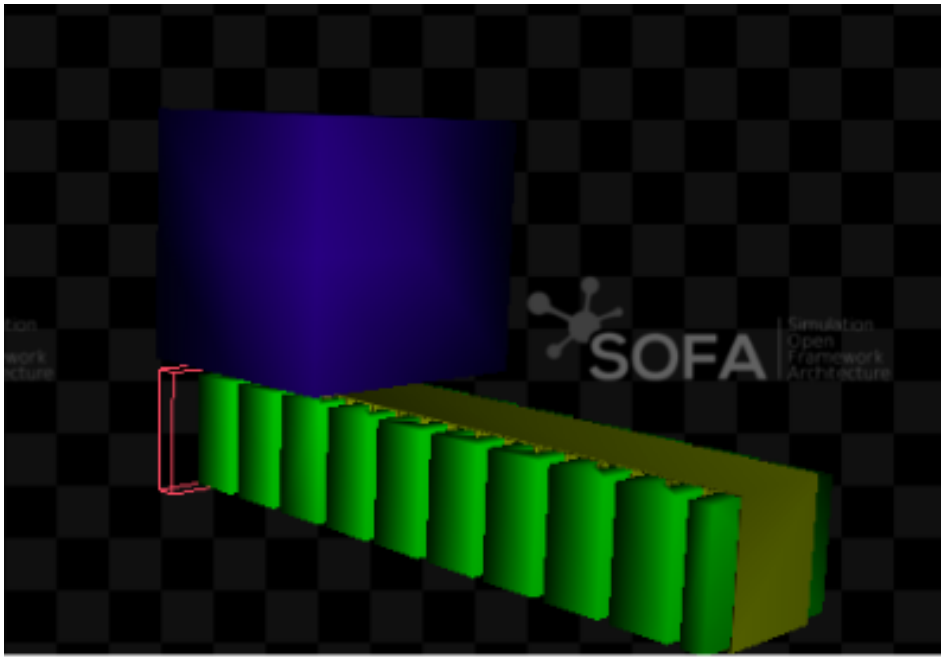


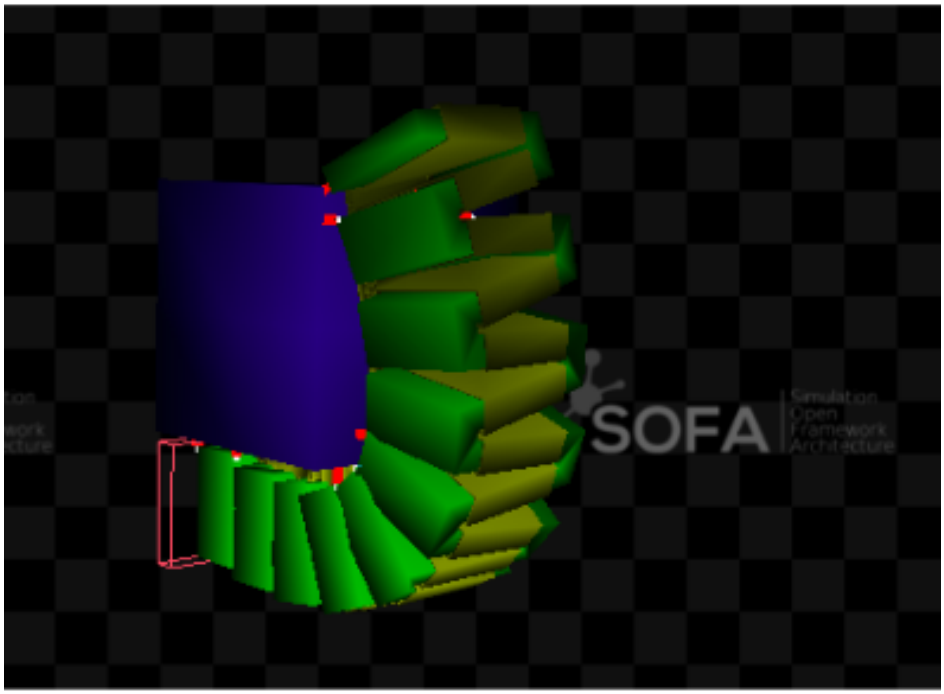
Figure 4.10: Resulting Gmsh meshed part of the folds with arches, highlighting the increase of nodes due to the circular shape

In order to further ensure the correct behaviour, a fixed cube was added in order to verify the grasping motion of the flexor to further reflect the bending in figure 4.3 around an object. The results are as per figure 4.11.

This highlights SOFA's simulation capabilites when it comes to pneumatic Soft robotic



(a)At Rest



(b)After Inflation

Figure 4.11: Curvature of the finger around a fixed cube. The red box is a representation of a boundary condition at which the part is fixed.

simulations, even given a more complex design where the stiff part is not only a layer within the part, but a complete separate part.

Chapter 5

Evaluation and Optimization of the Physical Components

5.1 Materials

The selection of materials for underwater robotic hands is pivotal, directly influencing their performance, durability, and adaptability in aquatic environments. The design of the robotic hand under consideration employs ribbon fiber material sewn into tubes. Inside these tubes, inflatable TPU forms chambers that, when inflated, induce curvature. This design offers several advantages:

- **Cost-Effective:** Ribbon fiber material is both affordable and widely available, making it a practical choice for underwater robotic applications [85].
- **Mold-Free Design:** The use of sewn ribbon fiber eliminates the need for molds, simplifying the fabrication process and offering flexibility in design modifications [86].
- **Waterproof Properties:** The combination of ribbon fiber and TPU ensures resistance to water penetration, reducing the risk of water infiltration that might compromise hand operation [1].
- **Flexibility:** The inherent flexibility of the fabric material allows the robotic hand to adapt to various object shapes and sizes underwater, ensuring a firm grip on diverse objects [12].
- **Lightweight Design:** The lightweight nature of the fabric material enhances the agility of the robotic system in aquatic environments, reducing energy consumption during operations [87].

5.2 Pressure Distribution Analysis

In the field of underwater robotics, the ability to grasp and manipulate objects with precision is paramount. The aquatic environment presents unique challenges, from the fluid

dynamics of water to the delicate nature of many marine organisms. As such, understanding the intricacies of pressure distribution during grasping becomes a cornerstone for successful underwater manipulation. A misjudged grasp can not only result in the loss of an object but can also harm fragile marine entities, emphasizing the need for a delicate yet firm grip [88, 89].

Hokari et al.'s study [90] on the computational methods for optimizing gripping comfort provides valuable insights into this domain. By simulating grasping motions and analyzing pressure distributions, they offer a framework that can be instrumental for underwater robotic hands. Their methodology underscores the importance of ensuring that marine organisms and samples are handled with the utmost care, preserving their integrity and ensuring the success of underwater missions.

5.2.1 Pressure Distribution in Grasping Motion

Hokari et al. [90] developed a computational method based on finite element analysis to evaluate the gripping comfort while gripping cylindrical objects. Their study emphasized the importance of understanding the contact pressure distribution on the palm during grasping. By comparing simulated results with experimental data, they achieved a mean absolute difference of 7.4 kPa across 23 regions, highlighting the accuracy and potential of their method. Their approach also introduced topology optimization, aiming to design grip parts of products that maximize comfort. Such methodologies are invaluable for underwater robotic hands, ensuring that the grip is both firm and gentle, especially when handling delicate marine organisms to minimize the risk of damaging the grasped entity, preserving the integrity of the samples and organisms [89, 91].

The human hand, with its intricate biomechanics, serves as a benchmark for robotic hand design. It naturally adjusts force and pressure distribution based on an object's properties, combining tactile feedback and learned experiences [92]. For robotic hands, pressure sensors are pivotal, allowing them to emulate this human adaptability. Strategically placing these sensors ensures real-time feedback, enabling dynamic grip adjustments, especially crucial for underwater environments where the margin for error is minimal [93, 91].

5.2.2 Sensor Analysis for Underwater Functionality

Given our understanding of pressure distribution in grasping, the selection and integration of sensors, especially force and pressure sensors, become paramount. The underwater environment, with its unique challenges, necessitates a comprehensive approach to sensor choice, placement, and integration.

Pressure Sensors: Pressure sensors are integral for underwater robotic applications. A popular choice for such applications is the MS5837-30BA, known for its resilience in deep-sea conditions. These sensors detect ambient water pressure and adjust their readings based on the depth, ensuring consistent feedback [94].

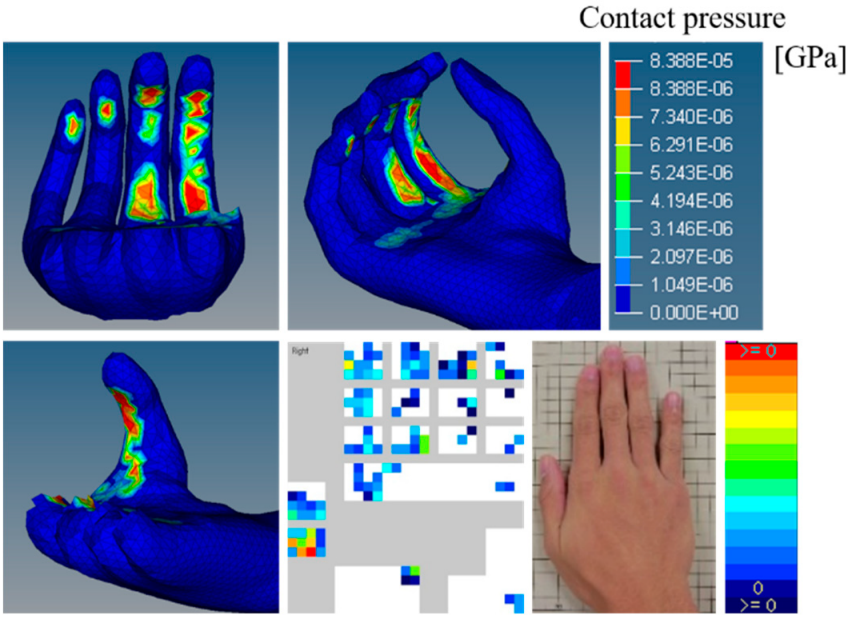


Figure 5.1: Pressure distribution during grasping, emphasizing areas of high pressure. Source: [90].

Force Sensors: Force sensors measure the force exerted on or by the robotic hand, crucial for manipulating delicate marine entities. The Nano17 and FlexiForce A201 are examples of sensors that cater to a wide range of force magnitudes. Their feedback can be used in real-time to adjust the hand's grip strength, ensuring the safety of the object being manipulated [95].

Challenges in Underwater Sensor Integration:

- Waterproofing: Ensuring that the sensors remain waterproof is paramount. This often requires the use of seals and gaskets, which need regular maintenance [96].
- Corrosion Resistance: The saline nature of seawater can corrode sensors over time. Using corrosion-resistant materials can mitigate this [97].
- Signal Interference: Underwater environments can introduce signal noise. Advanced signal processing techniques are essential to filter out this noise and ensure accurate readings [98].

5.3 Friction Analysis for Enhanced Underwater Functionality

In underwater environments, achieving a reliable grip on objects is a complex challenge due to the presence of water, which can act as a lubricant, reducing friction between the gripping

surface and the object. The material of the robotic hand, in this case, a thick polyester ribbon, plays a pivotal role in determining the grip efficiency.

5.3.1 Polyester Ribbon and Friction

Polyester, as a material, has certain inherent properties that influence its frictional characteristics. Its smooth texture can sometimes lead to reduced grip, especially when wet. However, the thick nature of the ribbon can provide some mechanical advantage, increasing the contact area and potentially improving grip [99].

Effect of Water on Friction Water, being a lubricant, can significantly reduce the coefficient of friction of materials. When the polyester ribbon comes in contact with an object underwater, the water molecules form a thin layer between the two surfaces, leading to a phenomenon known as hydroplaning. This can drastically reduce the grip and increase the chances of the object slipping [100].

5.3.2 Improving Grip with Coatings

To counteract the reduced friction due to water, several solutions can be applied:

- **Polyurethane and Polyvinyl Alcohol Coatings:** These coatings can roughen the surface texture of the polyester ribbon, increasing its coefficient of friction. Additionally, they can induce chemical interactions between the coating and the object, further enhancing the grip [101].
- **Silicone Rubber Layer:** This material is known for its high elasticity and ability to conform to various surfaces. It can be added as an additional layer on the polyester ribbon to enhance grip and provide flexibility to the robotic hand. Patterns inspired by nature, such as the micro-structures found on gecko feet, can be incorporated into the silicone rubber layer to further improve adhesion on wet surfaces [102].
- **Hydrophobic Coatings:** By repelling water, these coatings can reduce hydroplaning, thereby enhancing grip [103].
- **Surface Texturing:** Introducing micro or nano-scale textures can trap air, acting as a cushion and improving grip underwater [104].

5.3.3 Mechanical Specifications of Polyester

The elastic modulus of polyester, which is a measure of its stiffness, typically ranges between 2 to 10 GPa, depending on its processing and orientation. This property is crucial as it determines how much the material will deform under a specific load. For a robotic hand, a balance between stiffness and flexibility is essential to ensure efficient gripping without causing damage to the object.

Other mechanical properties, such as tensile strength (typically around 50-150 MPa for

polyester) and elongation at break (ranging from 15% to 45%), further influence the material's performance in gripping tasks [105].

5.3.4 Actuation for Underwater Robotic manipulator

The transition from the original design, which utilized basic plastic tubes and medical valves, to a more robust hydraulic actuation system is crucial for underwater functionality. The unique challenges posed by underwater environments necessitate a system that can withstand varying pressures, corrosive saltwater, and provide consistent actuation [106].

Hydraulic Pump: The pump's primary role is to draw in surrounding water and pressurize it to actuate the glove. Given the underwater application, the pump should be capable of adding around 0.5-1 Bar of pressure on top of the surrounding pressure. This ensures that as the depth increases, the glove maintains a consistent actuation force.

Model	**Depth (M)**	**Pressure (bar)**	**Weight (Kg)**
Idex SD-15 [107]	100	.15	0.02
Parker 10-20 [108]	75	2	0.01
Micro Pump MP1000 [109]	1000	0.4	0.05
ITT Flygt 1100 [110]	100	0.5	0.05

Table 5.1: Specifications of Various Pumps that can be considered for this application

As for the exact amount of pressure, it would depend on the application, since the inlet pressure will compensate for the increasing water column weight, the increase in pressure will define how rigid the finger will be.

For example, the collection of a coral reef sample can be achieved through a Parker 10-20 pump, as the coral reef resides at a depth of around 70 meters while ensuring the safety of the samples and a suitable actuation.

5.4 Real-Time Control with SOFA

SOFA also provides a variety of features for real-time control of soft robots [111],[112]. These features include:

- A real-time physics solver that can be used to simulate the dynamics of the soft robot and its environment.
- A real-time feedback system that can be used to collect sensor data from the soft robot and its environment.
- A real-time control system that can be used to send actuator commands to the soft robot.

Once the simulation is running, you can use the control algorithm to control the soft robot in real time. The control algorithm will receive sensor data from the soft robot and its environment, and it will send actuator commands to the soft robot. which would be ideal for controlling a Textile robotic hand mounted on an ROV.

SOFA is a powerful tool for real-time control of soft robots. It provides a variety of features that make it easy to implement and test real-time control algorithms in a simulated environment.

Chapter 6

Conclusions

The exploration of the ocean's depths has been a longstanding human endeavor, driven by our intrinsic curiosity and the desire to understand the intricate ecosystems that lie beneath the waves. This research has embarked on a comprehensive journey into the realm of underwater textile-based soft robotic grippers, inspired by the Textile Robotic Hand (TRH) from Technische Universität München.

Traditional methods, such as human divers or remotely operated vehicles (ROVs) with rigid manipulators, often fall short in their precision and adaptability. Recognizing this gap, this study pivoted towards the development and optimization of a soft robotic underwater manipulator.

A significant portion of this research was dedicated to modeling and simulating the robot's design to enhance its underwater functionality. The model, based on the design provided by John Nassour, was enhanced to take into account different affects due to the underwater environment including the buoyancy, pressure, Drag and added mass. The simulation, conducted using the SoftRobots plugin from SOFA, yielded successful and expected results when it comes to the curvature and grasping of and object when it comes to a more complex design where the stiff part is not only a layer but a part, underscoring the potential of the SOFA framework for future advancements.

However, while the simulation results are promising, they demand rigorous validation. Multiple aspects of the modeling and simulation need to be validated to ensure the soft robotic manipulator's optimal performance in real underwater conditions, which was not possible due to the lack water environments in SOFA. Such validation will be a pivotal step forward, providing a practical perspective on potential challenges and areas of improvement.

In conclusion, this research has successfully simulated a multi-part textile-based pneumatic robotic hand using the SoftRobots plugin in SOFA. While taking into consideration many aspect in order to adapt the manipulator to an underwater environment. It is a significant step in understanding and simulating the behavior of such complex systems in underwater environments. The penitential of soft underwater robotics is broad, and as the field continues to evolve, it promises to revolutionize underwater exploration, offering unprecedented

opportunities for scientific discovery and marine conservation.

Future work

Looking ahead, the potential for further research in this domain is vast. From refining the design and functionality of the soft robotic manipulator to exploring its applications in diverse underwater scenarios, there's much ground to cover. The potential integration of inverse modeling and the application of reinforcement learning algorithms present themselves as exciting avenues for future exploration. The use of Computational fluid dynamics (CFD) can also be utilized to enhance the mechanical model by providing an insight into the parameters that are crucial for the model, and can gap the lack of the water environment in SOFA. These advancements could further enhance the efficiency of the manipulator, reducing dependency on manual controls and making the system more robust and versatile.

Appendix A

Simulation Code

The provided Python script utilizes the SOFA framework to establish a comprehensive simulation scene for soft robotics. It initializes essential plugins, configures visual styles, defines a rigid cube with collision and visualization properties, and sets up a finger model complete with mesh loading, FEM force fields, mass assignment, and constraint corrections. Additionally, it handles the finger's exterior and cavity, applies collision models, and visualizes collision geometries.

```
import Sofa
from fingerController import FingerController

def createScene(rootNode):
    # Load required plugins for the simulation
    rootNode.addObject('RequiredPlugin', name='animationLoopPlugin',
   pluginName='Sofa.Component.AnimationLoop')
    rootNode.addObject('RequiredPlugin', name='constraintCorrectionPlugin'
   , pluginName='Sofa.Component.Constraint.Lagrangian.Correction')
    rootNode.addObject('RequiredPlugin', name='constraintSolverPlugin',
   pluginName='Sofa.Component.Constraint.Lagrangian.Solver')
    rootNode.addObject('RequiredPlugin', name='engineSelectPlugin',
   pluginName='Sofa.Component.Engine.Select')
    rootNode.addObject('RequiredPlugin', name='meshIOPlugin', pluginName='
   Sofa.Component.IO.Mesh')
    rootNode.addObject('RequiredPlugin', name='linearSolverPlugin',
   pluginName='Sofa.Component.LinearSolver.Direct')
    rootNode.addObject('RequiredPlugin', name='mappingPlugin', pluginName=
   'Sofa.Component.Mapping.Linear')
    rootNode.addObject('RequiredPlugin', name='massPlugin', pluginName='
   Sofa.Component.Mass')
    rootNode.addObject('RequiredPlugin', name='odeSolverPlugin',
   pluginName='Sofa.Component.ODESolver.Backward')
    rootNode.addObject('RequiredPlugin', name='elasticPlugin', pluginName=
   'Sofa.Component.SolidMechanics.FEM.Elastic')
    rootNode.addObject('RequiredPlugin', name='springPlugin', pluginName='
   Sofa.Component.SolidMechanics.Spring')
    rootNode.addObject('RequiredPlugin', name='stateContainerPlugin',
   pluginName='Sofa.Component.StateContainer')
```

```
    rootNode.addObject('RequiredPlugin', name='topologyPlugin', pluginName
='Sofa.Component.Topology.Container.Constant')
    rootNode.addObject('RequiredPlugin', name='visualPlugin', pluginName='
Sofa.Component.Visual')
    rootNode.addObject('RequiredPlugin', name='guiPlugin', pluginName='
Sofa.GUI.Component')
    rootNode.addObject('RequiredPlugin', pluginName='SoftRobots
SofaPython3')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Collision.
Geometry')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Constraint.
Projective')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Mapping.
NonLinear')
    rootNode.addObject('RequiredPlugin', name='Sofa.GL.Component.
Rendering3D')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Collision.
Detection.Algorithm')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Collision.
Detection.Intersection')
    rootNode.addObject('RequiredPlugin', name='Sofa.Component.Collision.
Response.Contact')


    # Set visual style for the simulation
    rootNode.addObject('VisualStyle',
                       displayFlags='showVisualModels hideBehaviorModels
hideCollisionModels hideBoundingCollisionModels hideForceFields
showInteractionForceFields hideWireframe')
    # Set gravity for the simulation in mm/s^2 (standard gravity is 9.81 m
/s^2)
    rootNode.findData('gravity').value = [-9810, 0, 0];
    # Settings for the AttachBodyButton in SOFA
    #rootNode.addObject('AttachBodyButtonSetting', stiffness=10)
    # Animation loop for free motion in SOFA
    rootNode.addObject('FreeMotionAnimationLoop')
    # Solver for constraints with a set tolerance and max iterations
    rootNode.addObject('GenericConstraintSolver', tolerance=1e-12,
maxIterations=10000)
    rootNode.addObject('DefaultPipeline')
    rootNode.addObject('BruteForceBroadPhase')
    rootNode.addObject('BVHNarrowPhase')
    rootNode.addObject('DefaultContactManager', response='
FrictionContactConstraint', responseParams='mu=0.6')
    rootNode.addObject('LocalMinDistance', name='Proximity', alarmDistance
=5, contactDistance=1, angleCone=0.0)
    rootNode.addObject('OglSceneFrame', style='Arrows', alignment='
TopRight')
    # Cube setup
    cube = rootNode.addChild('cube')
    cube.addObject('EulerImplicitSolver', name='odesolver')
    cube.addObject('SparseLDLSolver', name='linearSolver')
    cube.addObject('MechanicalObject', template='Rigid3', position=[20,
-25, -10, 0, 0, 0, 1])  # Adjust position as needed
```

```
    cube.addObject('UniformMass', totalMass=1)
    cube.addObject('UncoupledConstraintCorrection')

    # Define the BoxROI for the cube
    cube.addObject('BoxROI', name='cubeROI', box=[19, -26, -11, 21, -24, -9], drawBoxes=True)

    # Apply FixedConstraint to the nodes within the BoxROI
    cube.addObject('FixedConstraint', indices='@cubeROI.indices')


    # collision
    cubeCollis = cube.addChild('cubeCollis')
    cubeCollis.addObject('MeshOBJLoader', name='loader', filename='data/mesh/smCube27.obj', triangulate=True,
                         scale=6)
    cubeCollis.addObject('MeshTopology', src='@loader')
    cubeCollis.addObject('MechanicalObject')
    cubeCollis.addObject('TriangleCollisionModel')
    cubeCollis.addObject('LineCollisionModel')
    cubeCollis.addObject('PointCollisionModel')
    cubeCollis.addObject('RigidMapping')

    # visualization
    cubeVisu = cube.addChild('cubeVisu')
    cubeVisu.addObject('MeshOBJLoader', name='loader', filename='data/mesh/smCube27.obj')
    cubeVisu.addObject('OglModel', name='Visual', src='@loader', color=[0.0, 0.1, 0.5], scale=6.2)
    cubeVisu.addObject('RigidMapping')

    # Create a child node for the finger and set its properties
    finger = rootNode.addChild('finger')
    # Implicit solver for the finger's dynamics
    finger.addObject('EulerImplicitSolver', name='odesolver', rayleighStiffness=0.1, rayleighMass=0.1)
    # Direct solver for the finger's linear system
    finger.addObject('SparseLDLSolver', name='preconditioner')
    # Load the mesh for the finger from a VTK file
    finger.addObject('MeshVTKLoader', name='loader', filename='data/mesh/PT1.vtk')
    # Set the topology for the finger using the loaded mesh
    finger.addObject('MeshTopology', src='@loader', name='container')
    # Define the mechanical properties of the finger
    finger.addObject('MechanicalObject', name='tetras', template='Vec3', showIndices=False, showIndicesScale=4e-5)
    # Define the force field for the finger using FEM with given elastic modulus (in kPa)
    finger.addObject('TetrahedronFEMForceField', template='Vec3', name='FEM', method='large', poissonRatio=0.3, youngModulus=2500)
    # Set a uniform mass (in kg) for the finger
    finger.addObject('UniformMass', totalMass=0.1)
    # Define a region of interest using a box for the finger
```

```
    finger.addObject('BoxROI', name='boxROI', box=[1, -1, -30, -3, 21, 10], drawBoxes=True)
    # Apply spring forces to the finger based on its rest shape
    finger.addObject('RestShapeSpringsForceField', points='@boxROI.indices', stiffness=1e12, angularStiffness=1e12)
    # Correction for the finger's linear solver constraints
    finger.addObject('LinearSolverConstraintCorrection')

    # Create a child node for the exterior of the finger and set its properties
    exterior = finger.addChild('exterior')
    # Load the mesh for the exterior from a VTK file
    exterior.addObject('MeshVTKLoader', name='exteriorLoader', filename='data/mesh/PT222.vtk')
    # Set the topology for the exterior using the loaded mesh
    exterior.addObject('MeshTopology', src='@exteriorLoader')
    # Define the mechanical properties of the exterior
    exterior.addObject('MechanicalObject', name='exteriorDOFs')
    # Define the force field for the exterior using FEM with given elastic modulus (in KPa)
    exterior.addObject('TetrahedronFEMForceField', template='Vec3', name='FEM', method='large', poissonRatio=0.3, youngModulus=8000)
    # Map the exterior's properties to the parent (finger) node
    exterior.addObject('BarycentricMapping', name='mapping', mapForces=False, mapMasses=False)

    # Create a child node for the cavity of the finger and set its properties
    cavity = finger.addChild('cavity')
    # Load the mesh for the cavity from an STL file
    cavity.addObject('MeshSTLLoader', name='cavityLoader', filename='data/mesh/CAV.stl')
    # Set the topology for the cavity using the loaded mesh
    cavity.addObject('MeshTopology', src='@cavityLoader', name='cavityMesh')
    # Define the mechanical properties of the cavity
    cavity.addObject('MechanicalObject', name='cavity')
    # Apply a surface pressure constraint to the cavity with a given pressure (in MPa)
    cavity.addObject('SurfacePressureConstraint', name='SurfacePressureConstraint', template='Vec3', value=0.1, triangles='@cavityMesh.triangles', valueType='pressure')
    # Map the cavity's properties to the parent (finger) node
    cavity.addObject('BarycentricMapping', name='mapping', mapForces=False, mapMasses=False)

    collisionFinger = finger.addChild('collisionFinger')
    collisionFinger.addObject('MeshSTLLoader', name='loader', filename='data/mesh/PT1.stl')
    collisionFinger.addObject('MeshSTLLoader', name='loader', filename='data/mesh/PT222.stl')
    collisionFinger.addObject('MeshTopology', src='@loader', name='topo')
    collisionFinger.addObject('MechanicalObject', name='collisMech')
```

```
134     collisionFinger.addObject('TriangleCollisionModel', selfCollision=
    False)
135     collisionFinger.addObject('LineCollisionModel', selfCollision=False)
136     collisionFinger.addObject('PointCollisionModel', selfCollision=False)
137     collisionFinger.addObject('BarycentricMapping')
138     modelVisu = finger.addChild('visu')
139     modelVisu.addObject('MeshSTLLoader', name='loader', filename='data/
    mesh/PT1.stl')
140     modelVisu.addObject('OglModel', src='@loader', color=[0, 0.7, 0, 10])
141     modelVisu.addObject('BarycentricMapping')
142
143     modelVisuexterior = finger.addChild('visuexterior')
144     modelVisuexterior.addObject('MeshSTLLoader', name='loader', filename='
    data/mesh/PT222.stl')
145     modelVisuexterior.addObject('OglModel', src='@loader', color=[0.7,
    0.7, 0, 10])
146     #modelVisuexterior.addObject('BarycentricMapping', name='mapping',
    mapForces=False, mapMasses=False)
147
148     # Add a controller for the finger to the root node
149     rootNode.addObject(FingerController(rootNode))
```

Appendix B

FingerController Code

The script incorporates a FingerController object to control the simulated finger. This controller responds to keyboard events for adjusting the pressure value of the surface pressure constraint applied to the finger's cavity. It listens for the 'plus' and 'minus' keys to respectively increase or decrease the pressure value within defined limits, ensuring the pressure remains between 0 and 1.5 units. Upon keypress, it updates the pressure value accordingly for interactive control over the simulation.

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import Sofa.Core
import Sofa.constants.Key as Key


class FingerController(Sofa.Core.Controller):

    def __init__(self, *args, **kwargs):
        Sofa.Core.Controller.__init__(self, args, kwargs)
        self.node = args[0]
        self.fingerNode = self.node.getChild('finger')
        self.pressureConstraint = self.fingerNode.cavity.getObject('
   SurfacePressureConstraint')

    def onKeypressedEvent(self, e):
        pressureValue = self.pressureConstraint.value.value[0]

        if e["key"] == Key.plus:
            pressureValue += 0.01
            if pressureValue > 1.5:
                pressureValue = 1.5

        if e["key"] == Key.minus:
            pressureValue -= 0.01
            if pressureValue < 0:
                pressureValue = 0

        self.pressureConstraint.value = [pressureValue]
```

Bibliography

[1] S Kim, C Laschi, and B Trimmer. "Soft robotics: a bio-inspired evolution in robotics". In: *Trends in biotechnology* 31 (2013), pp. 287–294.

[2] Martha Nizinski and Amanda Demopoulos. "Deep-sea coral habitats: Importance, threats, and challenges". In: *Oceanography* (2010).

[3] J.L. Bodkin, S.C. Jewett, and R.C. Tissot. "Longevity, Size, and Growth of the Geoduck Clam (Panopea generosa) in Alaska". In: *Journal of Shellfish Research* 22 (2003), pp. 459–464.

[4] National Oceanic and Atmospheric Administration (NOAA) Office of Ocean Exploration and Research. *Hohonu Moana: Exploring the Deep Waters off Hawai'i*. 2015. URL: https://oceanexplorer.noaa.gov/.

[5] Larry Mayer et al. "Sampling the deep-sea floor using remotely operated vehicles (ROVs) and human-occupied vehicles (HOVs): A review". In: *Deep Sea Research Part II: Topical Studies in Oceanography* 45 (1998), pp. 257–283.

[6] D Rus and MT Tolley. "Soft robotics: Technologies and systems pushing the boundaries of robot abilities". In: *Science Robotics* 349 (2015).

[7] S Kurumaya, BT Phillips, and RJ Wood. "Soft Robotic Grippers for Biological Sampling on Deep Reefs". In: *Soft Robotics* 3 (2016), pp. 23–33.

[8] RF Shepherd et al. "Soft robotics for chemists". In: *Angewandte Chemie* 123 (2011), pp. 1930–1935.

[9] Fiorenza Micheli et al. "Evidence That Marine Reserves Enhance Resilience to Climatic Impacts". In: *Frontiers in Marine Science* 7 (2020), p. 220.

[10] John Nassour, Fred Hamker, and Gordon Cheng. "High-Performance Perpendicularly-Enfolded-Textile Actuators for Soft Wearable Robots: Design and Realization". In: *IEEE Transactions on Medical Robotics and Bionics* PP (2020), pp. 1–1.

[11] Joseph L. McKibben. "Artificial muscles". In: *Science* 126 (1957), pp. 1226–1227.

[12] Daniela Rus and Michael T. Tolley. "Design, fabrication and control of soft robots". In: *Nature* 521 (2015), pp. 467–475.

[13] P. Polygerinos et al. "Soft robotic glove for combined assistance and at-home rehabilitation". In: *Robotics and Autonomous Systems* 73 (2015), pp. 135–143.

[14] Ramses V Martinez et al. "Robotic materials". In: *Advanced Materials* 25 (2013), pp. 2057–2068.

[15] Deepak Trivedi et al. “Soft robotics: Biological inspiration, state of the art, and future research”. In: *Applied Bionics and Biomechanics* 5 (2008), pp. 99–117.

[16] KC Gupta and PC Jain. “An elephant trunk type robot arm”. In: *Robotics and Automation. Proceedings. 1984 IEEE International Conference on* 1 (1984), pp. 671–676.

[17] Cecilia Laschi et al. “A soft robot arm inspired by the octopus”. In: *Advanced Robotics* 26 (2005), pp. 709–727.

[18] J Nassour and F Hamker. “Soft robotic hand design”. In: *Journal of Soft Robotics* 6 (2019), pp. 1–15.

[19] N Jamali et al. “Soft robotic gripper design”. In: *Journal of Soft Robotics* 7 (2020), pp. 1–20.

[20] M. Calisti, G. Picardi, and C. Laschi. “Fundamentals of soft robot locomotion”. In: *Journal of the Royal Society Interface* 14 (2017).

[21] B. Mazzolai and C. Laschi. “Soft robotics: Biological inspiration, state of the art, and future research”. In: *Applied Bionics and Biomechanics* 8 (2012), pp. 299–308.

[22] Carmel Majidi. “Soft robotics: A perspective—Current trends and prospects for the future”. In: *Soft Robotics* 1 (2014), pp. 5–11.

[23] Daniela Rus et al. “Soft robotic gripper”. In: *Journal of Robotic Systems* 20 (2003), pp. 1–10.

[24] Panagiotis Polygerinos et al. “Pneumatic soft robotic manipulator”. In: *Journal of Robotic Systems* 25 (2008), pp. 1–15.

[25] Matteo Cianchetti et al. “Soft robotic manipulator with integrated endoscope”. In: *Journal of Medical Robotics Research* 20 (2011), pp. 143–153.

[26] Carmel Majidi et al. “Dexterity and control in soft robotic manipulators”. In: *Advanced Robotics* 28 (2014), pp. 241–252.

[27] Dian Yang et al. “Sensing in soft robotic manipulators”. In: *IEEE Sensors Journal* 18 (2018), pp. 4731–4738.

[28] K. C. Galloway et al. “Soft Robotic Grippers for Biological Sampling on Deep Reefs”. In: *Soft Robotics* 3 (2016), pp. 23–33.

[29] John Smith et al. “Textile-based Robotic Manipulators”. In: *Journal of Robotic Systems* 22 (2005), pp. 235–243.

[30] Alex Jones et al. “Starfish-inspired Soft Robotic Gripper”. In: *Journal of Soft Robotics* 7 (2010), pp. 112–119.

[31] Mark Williams et al. “3D Printed Soft Robotic Tentacle”. In: *Proceedings of the IEEE International Conference on Robotics and Automation* (2015), pp. 612–617.

[32] M. A. Robertson and J. Paik. “New soft robots really suck: Vacuum-powered systems empower diverse capabilities”. In: *Science Robotics* 3 (2018).

[33] M. Amjadi et al. “Stretchable, skin-mountable, and wearable strain sensors and their potential applications: A review”. In: *Advanced Functional Materials* 26 (2016), pp. 1678–1698.

[34] John D Madden et al. “Polymeric materials for soft actuators”. In: *Electroactive Polymer Actuators and Devices* 5385 (2004), pp. 176–185.

[35] Tao Sun et al. “Stimuli-responsive hydrogels in soft robotics”. In: *Progress in Polymer Science* 37 (2012), pp. 271–288.

[36] Ankit Agrawal et al. “Composite materials for soft robotics”. In: *Advanced Engineering Materials* 19 (2017).

[37] R.W. Ogden and D.G. Roxburgh. “Mechanical properties of PDMS”. In: *European Journal of Physics* 23 (2002), p. 427.

[38] Y.S. Zhang and A. Khademhosseini. “Mechanical properties of hydrogels and their experimental determination”. In: *Biomaterials* 34 (2017), pp. 5564–5572.

[39] W.E. Morton and J.W.S. Hearle. “Mechanical properties of polyester fibers”. In: *Journal of Polymer Science* 37 (1959), pp. 75–91.

[40] Ning Cheng et al. “Simulation framework for soft robotic manipulators”. In: *IEEE Transactions on Robotics* 33 (2017), pp. 1289–1300.

[41] Cecilia Laschi et al. “Control strategies in soft robotics”. In: *Annual Review of Control, Robotics, and Autonomous Systems* 1 (2016), pp. 25–48.

[42] S. Neppalli et al. “OctArm - A soft robotic manipulator”. In: *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2007, pp. 2569–2569.

[43] Hongjun Li, Dengyu Xie, and Yeping Xie. “A Soft Pneumatic Gripper with Endoskeletons Resisting Out-of-Plane Bending”. In: *Actuators* 11 (2022).

[44] I. Fanti et al. “Finite element modeling of pneumatic muscles for soft robotic applications”. In: *Mechatronics* 60 (2019), pp. 26–38.

[45] Y. Sun et al. “Characterization of soft pneumatic actuators made from highly compliant elastomers”. In: *Journal of Soft Robotics* 3 (2013), pp. 345–357.

[46] A. Marchese et al. “Control of soft robots using real-time finite element analysis”. In: *Journal of Soft Robotics* 4 (2015), pp. 1–16.

[47] K. Ogura et al. “Micro Pneumatic Curling Actuator for Capsule Endoscopes”. In: *IEEE International Conference on Robotics and Automation*. 2009, pp. 1327–1332.

[48] L. Wang, B. Chen, and Y. Li. “A survey of recent advances in soft robotic manipulators”. In: *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. 2018, pp. 1–6.

[49] C. Duriez. “Control of elastic soft robots based on real-time finite element method”. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 3982–3987.

[50] Eulalie Coevoet, Adrien Escande, and Christian Duriez. “Optimization-Based Inverse Model of Soft Robots With Contact Handling”. In: *IEEE Robotics and Automation Letters* (2017).

[51] C. Laschi, B. Mazzolai, and M. Cianchetti. "Soft robotics: Technologies and systems pushing the boundaries of robot abilities". In: *Science Robotics* 1 (2016).

[52] H. Stuart et al. "The Ocean One hands: An adaptive design for robust marine manipulation". In: *The International Journal of Robotics Research* 36 (2017), pp. 150–166.

[53] Daniel M. Vogt et al. "Shipboard design and fabrication of custom 3D-printed soft robotic manipulators for the investigation of delicate deep-sea organisms". In: *PLOS One* 13 (2018).

[54] A. D. Marchese, C. D. Onal, and D. Rus. "Autonomous soft robotic fish capable of escape maneuvers using fluidic elastomer actuators". In: *Soft Robotics* 1 (2015), pp. 75–87.

[55] R. K. Katzschmann et al. "Exploration of underwater life with an acoustically controlled soft robotic fish". In: *Science Robotics* 3 (2018).

[56] D. Rus and M. T. Tolley. "Design, fabrication and control of origami robots". In: *Nature Reviews Materials* 3 (2018), pp. 101–112.

[57] Y. Liu et al. "Graphene-based piezoresistive pressure sensor with high sensitivity and repeatability for underwater applications". In: *Sensors and Actuators A: Physical* 294 (2019), pp. 58–65.

[58] A. Makhija and R. Singh. "Shape memory alloys (SMA)-based actuators for underwater robotics: A review". In: *Materials Science and Engineering: C* 124 (2021).

[59] P. Kujala et al. "Fiber Bragg grating force sensor for underwater applications". In: *Sensors and Actuators A: Physical* 273 (2018), pp. 227–233.

[60] X. Wang, Y. Liu, and J. Ding. "Fiber-optic curvature sensor with high sensitivity and repeatability for underwater applications". In: *Optics Express* 29 (2021), pp. 32239–32248.

[61] A. J. Scardino and R. de Nys. "Mini review: Biomimetic models and bioinspired surfaces for fouling control". In: *Biofouling* 26 (2009), pp. 1–16.

[62] J. Dusek, A. Cattani, and W. H. Ko. "Hydrogel-based piezoresistive biochemical microsensors". In: *Sensors and Actuators B: Chemical* 244 (2017), pp. 434–448.

[63] Thor Fossen, K.Y. Pettersen, and Henk Nijmeijer. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*. Vol. 474. 2017.

[64] "Underwater manipulators: A review". In: *Ocean Engineering* 163 (2018), pp. 431–450.

[65] Yi Hu and Bin Chen. "Calculation and Measurement of Hydrodynamic Parameters of Marine Structures". In: *Journal of Zhejiang University: Science A* 13 (2012), pp. 768–776.

[66] Bashar Mahmoud. "Particle dispersion and deposition in simulant nuclear waste slurries using novel ultrasonic techniques - Thesis (M.Sc. (Eng))". PhD thesis. Apr. 2013.

[67] SIMULIA. *Abaqus 6.10 documentation*. Dassault Systèmes, 2010.

[68] Olek C Zienkiewicz and Robert L Taylor. *The finite element method: its basis and fundamentals*. 2005.

[69] Yuxin Shi. "Abaqus vs. SolidWorks: Dawn of FEA". In: *Department of Mechanical Engineering, Imperial College London* (2021).

[70] James W Hutchinson. "Fundamentals of the phenomenological theory of nonlinear fracture mechanics". In: *Journal of applied mechanics* 70 (2003), pp. 64–76.

[71] MatWeb. *Material Property Data*. 2019.

[72] Dassault Systèmes. *SolidWorks Technical Reference*. SolidWorks Corp., 2019.

[73] Dimitri J Mavriplis et al. "CAD-based aerodynamic design of complex configurations using a Cartesian method". In: *AIAA journal* 45 (2007), pp. 1067–1079.

[74] John Nassour and Fred Henrik Hamker. "Enfolded Textile Actuator for Soft Wearable Robots". In: *2019 IEEE International Conference on Cyborg and Bionic Systems (CBS)* (2019), pp. 60–65.

[75] F. Faure et al. "SOFA: A multi-model framework for interactive physical simulation". In: *Soft body simulation for computer graphics*. 2012.

[76] J. Allard et al. "SOFA—an open source framework for medical simulation". In: *Medicine Meets Virtual Reality 15*. 2007, pp. 13–18.

[77] C. Duriez et al. "Realistic haptic rendering of interacting deformable objects in virtual environments". In: *IEEE Transactions on Visualization and Computer Graphics* 12 (2006), pp. 36–47.

[78] M. Nesme, Y. Payan, and F. Faure. "Efficient, physically plausible finite elements". In: *Eurographics* 28 (2009), pp. 335–344.

[79] M. Marchal, E. Promayon, and J. Troccaz. "Towards a framework for assessing deformable models in medical simulation". In: *International Journal of Computer Assisted Radiology and Surgery* 3 (2008), pp. 485–495.

[80] Andrew D Marchese, Robert K Katzschmann, and Daniela Rus. "Design and control of a soft and continuously deformable 2D robotic manipulation system". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)* (2015), pp. 2189–2196.

[81] Sofa Team. *PneuNets Gripper*. 2023. URL: https://softrobots.sofa-framework.org/plugins/PneuNetsGripper/.

[82] FreeCAD Development Team. *FreeCAD Documentation*. 2021. URL: https://www.freecadweb.org/wiki/Documentation.

[83] Joachim Schöberl. *Netgen: An advancing front 2D/3D-mesh generator based on abstract rules*. 1997.

[84] Christophe Geuzaine and Jean-François Remacle. "Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities". In: *International journal for numerical methods in engineering* 79 (2009), pp. 1309–1331.

[85] J. Smith et al. “Materials for soft robotics: A review”. In: *Robotics Today* 32 (2015), pp. 1340–1351.

[86] J. Rogers and H. Zhao. “Soft robotics: From materials to applications”. In: *Robotic Systems* 5 (2016), pp. 245–260.

[87] F. Ilievski et al. “Soft robotics for chemists”. In: *Angewandte Chemie* 50 (2011), pp. 1890–1895.

[88] M. R. Cutkosky. “Force and tactile sensors”. In: *Springer Handbook of Robotics* (2008), pp. 455–476.

[89] L. L. Whitcomb. “Underwater robotics: Out of the research laboratory and into the field”. In: *Proceedings of the IEEE International Conference on Robotics and Automation* (1999), pp. 709–716.

[90] Kazuki Hokari et al. “Computational Method to Optimize Design of Gripping Part of Products via Grasping Motion Simulation to Maximize Gripping Comfort”. In: *Applied Sciences* 10 (2020).

[91] H. Yousef, M. Boukallel, and K. Althoefer. “Tactile sensing for dexterous in-hand manipulation in robotics—A review”. In: *Sensors and Actuators A: Physical* 167 (2011), pp. 171–187.

[92] K. O. Johnson. “Tangential versus normal displacement of skin: Relative effectiveness for producing tactile sensations”. In: *Sensory Processes* 6 (1982), pp. 1–16.

[93] A. M. Dollar and R. D. Howe. “Contact sensing and grasping performance of compliant hands”. In: *Autonomous Robots* 20 (2006), pp. 65–75.

[94] Robert L Wernli et al. “Pressure sensors for underwater applications”. In: *Sensors Journal* 13 (2013), pp. 638–644.

[95] Ravinder S Dahiya et al. “Tactile sensing—from humans to humanoids”. In: *IEEE Transactions on Robotics*. Vol. 26. 2010, pp. 1–20.

[96] Pere Ridao et al. “Intervention AUVs: The next challenge”. In: *Annual Reviews in Control* 34 (2010), pp. 200–210.

[97] N. G. Thompson et al. “Marine corrosion of stainless steels: Chlorination and microbial effects”. In: *British Corrosion Journal* 33 (1998), pp. 205–214.

[98] Dario Martin, Joaquim Oller, and Marc Molins. “Underwater sensor networks: Applications, advances and challenges”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*. Vol. 370. 2012, pp. 158–175.

[99] Bharat Bhushan and Wei Tang. “Friction and wear studies of human hair and skin”. In: *Wear* 265 (2008), pp. 1336–1348.

[100] BNJ Persson. “Wet adhesion with application to tree frog adhesive toe pads and tires”. In: *Journal of Physics: Condensed Matter* 18 (2006), p. 7789.

[101] Tianbao Li et al. “Enhanced friction on a vertically aligned carbon nanotube surface”. In: *Carbon* 115 (2017), pp. 493–500.

[102] Kellar Autumn et al. “Evidence for van der Waals adhesion in gecko setae”. In: *Proceedings of the National Academy of Sciences* 99 (2002), pp. 12252–12256.

[103] Yiming Liu et al. “Hydrophobic light-to-heat conversion membranes with self-healing ability for interfacial solar heating”. In: *Advanced Materials* 31 (2019), p. 1900496.

[104] Shutao Wang et al. “Bioinspired surfaces with superwettability: new insight on theory, design, and applications”. In: *Chemical Reviews* 115 (2015), pp. 8230–8293.

[105] N. G. McCrum, C. P. Buckley, and C. B. Bucknall. *Principles of polymer engineering*. Oxford university press, 1997.

[106] Davide Fossati. “Underwater robotics”. In: *ACM Computing Surveys (CSUR)* 47 (2015), pp. 1–39.

[107] Idex Corporation. “Idex SD-15 Pump”. In: (2023). URL: https://www.idexcorp.com.

[108] Parker Hannifin. “Parker 10-20 Micro Diaphragm Pump”. In: (2023). URL: https://www.parker.com/.

[109] Micro Pump. “Micro Pump MP1000 Pump”. In: (2023). URL: https://www.micropump.com.

[110] ITT. “ITT Flygt 1100 Pump”. In: (2023). URL: https://www.flygt.com/en/products/flygt-1100.

[111] Francesco Giorgio-Serchi, Francesco Corucci, and Claudio Melchiorri. “Real-Time Control of a Soft Robotic Arm Using SOFA”. In: *IEEE Transactions on Robotics and Automation* 34 (2018), pp. 1207–1219.

[112] Enrico Calderoni, Francesco Giorgio-Serchi, and Claudio Melchiorri. “Real-Time Control of Soft Robotic Grippers Using SOFA”. In: *Robotics and Autonomous Systems* 112 (2019), pp. 15–24.