Contributions to the Concept of Embodied Intelligence in Soft Robotics through Control and Design Co-Optimization

Tanguy Navez

Université de Lille

Université de Lille

Doctoral School **MADIS**

Research unit **DEFROST (Deformable Robotics Software)**

Thesis defended by **Tanguy Navez**

Defended on **October 4, 2024**

Thesis submitted in order to become a Doctor of Philosophy from Université de Lille

Academic Field **Computer Science**

Contributions to the Concept of Embodied Intelligence in Soft Robotics through Control and Design Co-Optimization

Thesis supervised by	Christian Duriez	Supervisor
	Olivier Goury	Co-Supervisor

Committee members

Referees	Guillaume Laurent	Professor at ENSMM	
	Vincent Lebastard	Associate Professor at IMT Atlantique	
Examiners	Sinan Haliyo	Professor at Sorbonne Université	Committee President
	Audrey Sedal	Associate Professor at McGill University	
	Pierre Renaud	Professor at INSA Strasbourg	
Supervisors	Christian Duriez	Senior Researcher at Centre Inria de l'Université de Lille	
	Olivier Goury	Junior Researcher at Centre Inria de l'Université Grenoble Alpes	

Université de Lille

Université de Lille

École doctorale **MADIS**

Unité de recherche **DEFROST (Deformable Robotics Software)**

Thèse présentée par **Tanguy Navez**

Soutenue le **4 octobre 2024**

Thèse soumise en vue de l'obtention du grade de Docteur de l'Université de Lille

Discipline **Informatique**

Contribution au Concept d'Intelligence Incarnée en Robotique Souple via la Co-Optimisation du Contrôle et de la Conception

Thèse dirigée par	Christian Duriez	directeur
	Olivier Goury	co-directeur

Composition du jury

Rapporteurs	Guillaume Laurent	professeur à l'ENSMM	
	Vincent Lebastard	mcf à l'IMT Atlantique	
Examinateurs	Sinan Haliyo	professeur au Sorbonne Université	président du jury
	Audrey Sedal	mcf au McGill University	
	Pierre Renaud	professeur à l'INSA Strasbourg	
Directeurs de thèse	Christian Duriez	directeur de recherche au Centre Inria de l'Université de Lille	
	Olivier Goury	chargé de recherche au Centre Inria de l'Université Grenoble Alpes	

Contributions to the Concept of Embodied Intelligence in Soft Robotics through Control and Design Co-Optimization

Abstract

Soft robots utilize compliant materials to interact in a flexible way with complex and uncertain environments, enabling the manipulation of fragile objects and safe interaction with living beings. Their adaptability drives innovations in fields such as medicine and manufacturing.

Designing soft robots is challenging, even for experienced designers, due to their nonlinear materials, multiphysics coupling, the complex interaction between multiple bodies and the environment, and their many degrees of freedom. This explained why the first designs in soft robotics were inspired by nature, mimicking soft animals such as worms or octopuses. Soft bodies have the ability to conform to hard object and reconfigure for different tasks, then delegating an important part of the control to the body. Unlike rigid robots, embodied intelligence is still an emerging topic in soft robotics. However, it is clear that intelligent behaviors can be quickly learned by agents whose morphology is well-adapted to their environment. Moving away from the traditional focus on training control and dexterity, this thesis aims to address control challenges by linking artificial intelligence with soft robot design.

The field of soft robotics presents numerous challenges in modeling, control, and design. The DEFROST team at Inria Lille has developed several Finite Element Method (FEM)-based tools to address these challenges, enabling accurate simulation of soft robots. These tools have been used for low-level control and to evaluate soft robot designs before manufacturing. In this work, various FEM-based simulation and numerical optimization tools are applied to explore the computational design of soft robots.

This exploration requires addressing several challenges. The design space must be sufficiently large to explore relevant designs but also constrained enough to make the optimization problem tractable. Developing relevant mathematical fitness functions is crucial for accurately evaluating the performance and effectiveness of soft robot designs. Given the significant data requirements of computational design algorithms and computational expense of accurate simulations, we aim to speed up the simulation either by choosing models that balance computation time and accuracy or by employing learning techniques to accelerate FEM simulations.

This thesis explores the computational design of soft robots, with a focus on the simulation-to-reality transfer of numerical results. The design optimization of two parametric soft manipulators, one with embedded sensors and the other with self-contact capabilities, was addressed. As control tasks, environments, and design spaces become more complex, the computational burden increases. This motivates the development of a surrogate model learned from FEM simulations to characterize both soft robot design and control. The model's applicability is demonstrated through various scenarios, notably the embedded control of a pneumatic manipulator and the computational design of a soft manipulator. In addition, a key objective of this work is to develop tools towards the co-optimization of soft robot design and control.

Keywords: soft robotics, computational design, reduced order modeling

DEFROST (Deformable Robotics Software)
CRIStAL, Centre Inria de l'Université de Lille

Contribution au Concept d'Intelligence Incarnée en Robotique Souple via la Co-Optimisation du Contrôle et de la Conception

Résumé

Les robots souples utilisent des matériaux conpliants pour interagir de manière flexible avec des environnements complexes et incertains, permettant ainsi la manipulation d'objets fragiles et une interaction sécurisée avec les êtres vivants. Leur adaptabilité favorise des innovations dans des domaines tels que la médecine et l'industrie.

La conception de robots souples est un défi, même pour les concepteurs expérimentés, en raison de leurs matériaux non linéaires, du couplage multiphysique, de l'interaction complexe entre plusieurs corps et l'environnement, et de leurs nombreux degrés de liberté. Cela explique pourquoi les premières conceptions en robotique souple s'inspiraient de la nature, en imitant des animaux souples comme les vers ou les pieuvres. Les corps souples ont la capacité de se conformer à des objets durs et de se reconfigurer pour différentes tâches, déléguant ainsi une partie importante du contrôle au corps lui-même. Contrairement aux robots rigides, l'intelligence incarnée est encore un sujet émergent en robotique souple. Cependant, il est évident que des comportements intelligents peuvent être rapidement appris par des agents dont la morphologie est bien adaptée à leur environnement. En s'éloignant de la vision traditionnelle de l'intelligence artificielle en robotique, qui se concentre sur l'entraînement au contrôle et à la dextérité du robot sur diverses tâches, cette thèse vise à aborder les défis de contrôle en établissant un lien entre l'intelligence artificielle et la conception des robots souples.

Le domaine de la robotique souple présente de nombreux défis en matière de modélisation, de contrôle et de conception. L'équipe DEFROST d'Inria Lille a développé plusieurs outils basés sur la méthode des éléments finis pour relever ces défis, permettant une simulation précise des robots souples. Ces outils ont été utilisés pour le contrôle bas niveau et pour évaluer les conceptions de robots souples avant leur fabrication. Dans ce travail, divers outils de simulation basés sur la méthode des éléments finis et d'optimisation numérique sont appliqués pour explorer la conception computationnelle des robots souples.

Cette exploration nécessite de relever plusieurs défis. L'espace de conception doit être suffisamment vaste pour explorer des conceptions pertinentes, mais aussi suffisamment contraint pour rendre le problème d'optimisation abordable. Développer des fonctions d'optimisation mathématiques pertinentes est crucial pour évaluer avec précision la performance et l'efficacité des conceptions de robots souples. Étant donné les exigences importantes en matière de données des algorithmes de conception computationnelle et le coût élevé des simulations précises, nous visons à accélérer la simulation soit en choisissant judicieusement des modèles qui équilibrent le temps de calcul et la précision, soit en employant des techniques d'apprentissage pour accélérer les simulations éléments finies.

Cette thèse explore la conception computationnelle des robots souples, en mettant l'accent sur le transfert des résultats numériques de la simulation à la réalité. L'optimisation de la conception de deux manipulateurs souples paramétriques, l'un avec des capteurs intégrés et l'autre avec des capacités d'auto-contact, a été abordée. À mesure que les tâches de contrôle, les environnements et les espaces de conception deviennent plus complexes, la charge computationnelle augmente. Cela motive le développement d'un modèle de substitution appris à partir des simulations éléments finis pour caractériser à la fois la conception et le contrôle des robots souples. L'applicabilité du modèle est démontrée à travers divers scénarios, notamment le contrôle embarqué d'un manipulateur pneumatique et la conception computationnelle d'un manipulateur souple. En outre, un objectif clé de ce travail est de développer des outils pour la co-optimisation de la conception et du contrôle des robots souples.

Mots clés : robotique souple, conception numérique, réduction de modèle

Remerciements

Ce doctorat a été co-financé par la région Hauts-de-France.

Table of Contents

Chapter 1

Overall Introduction

1.1 Introduction and Chapter Overview

The objective of this first chapter is to provide context and give an overall introduction to the present work. We begin this chapter by an introduction of a short history and the motivations behind the soft robotics field, followed by a discussion about the main characteristics and properties of soft robots. Then, the second section explore the concept of Embodied Intelligence through the observation of biological systems characteristics. We show that part of the intelligence classically attributed to the brain can be shared with the body, and the growing interest and attempts to apply these principles for building efficient soft robots. The third section is then dedicated to the design of soft robots in this framework. Although the soft robotics community historically took inspiration from biological organisms for building soft robots, a systemic approach based on computational design framework need to be developed. After introducing the computational design general methodology, we show that the scope of this thesis is then extensive, encompassing numerical simulation, reduced modeling, control, and computational design in the context of soft robotics. Background and related works relative to the aforementioned topics will be in-depth introduced in Chapter 2. Finally, the key contributions of this work are highlighted and the manuscript outlines are given.

1.2 Soft Robotics

1.2.1 History and Motivations

Robotics initially transformed industries by automating challenging and repetitive tasks. Traditionally, robots were built with rigid components, which simplified their design, modeling, and control. While this rigidity allowed for high precision, it also posed risks during interactions with delicate environments, such as handling fragile objects or working alongside humans.

As robots began to extend their applications beyond factories into the medical field, particularly as tools to assist surgeons with minimally invasive procedures, researchers turned their attention to soft materials. Soft robotics emerged as a means to achieve safe and compliant interactions with human organs. The probably first soft robot [SIT91], introduced in the 1990s and inspired by the need for medical micro-robots, marked the beginning of the field. In the subsequent decades, soft robotics evolved into an active multidisciplinary area of research,

focusing on various aspects such as material sciences, design, fabrication, modeling, and control of soft bodies.

Soft robots are increasingly being integrated into various applications, as illustrated in Figure 1.1. In the medical field, they are utilized for exoskeletons in rehabilitation and for minimally invasive surgery as endoscopes or catheters [Cia+18]. In industrial settings, they are used as robotic grippers, taking advantage of their deformable bodies to adapt to the shapes of fragile objects. Additionally, soft robots are emerging in search and rescue operations, where their flexible bodies allow them to navigate through challenging environments that rigid robots cannot easily access. This thesis will concentrate mostly on soft robots for manipulation.

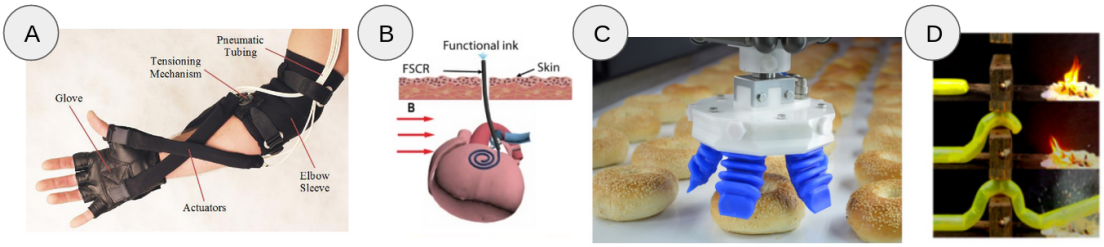

Figure 1.1: Example of soft robots. A) Soft robot orthosis for wrist rehabilitation [Bar+15a]. B) Ferromagnetic soft catheter for minimally invasive bio-printing, i.e. in vivo fabrication of artificial tissues and medical devices [Zho+21a]. C) Soft gripper for the food industry, launched by the start-up SoftRobotics [Lab]. D) A pneumatic soft robot for exploration that navigate constrained environments through growth [Haw+17].

1.2.2 Definition

The definition of soft robotics is not entirely stable. Initially, soft robots were classified based on the material they are made of and its corresponding Young's modulus. Specifically, soft robots are considered to be those made from materials with a Young modulus between $10^4 Pa$ and $10^9 Pa$ [Maj14]. However, this classification has its limitations, as materials with a high Young modulus can still be compliant depending on their structural organization and density. Additionally, many systems identified as soft robots are actually hybrids, comprising both rigid and deformable components. For instance, passive compliance can be reached using compliant elements, such as springs or dampers. In this thesis, we investigate soft robots that demonstrate passive compliant behavior by using flexible materials or rigid materials with specific infill patterns and densities for their bodies.

While classical rigid robots have a finite number of degrees of freedom, soft bodies are theoretically characterized by an infinite number of degrees of freedom, meaning an infinite number of independent relative motions. Consequently, soft robots have under-actuated and redundant structures, as they lack an actuator for each degree of freedom. Thus, multiple configurations may achieve the same goal through deformation. These characteristics make it challenging to develop efficient controllers for soft robots and to design them intuitively, which underscores the importance of the use of computational algorithms in this thesis work.

1.3 Embodied Intelligence in Soft Robotics

The concept of Embodied Intelligence is broad and encompasses different topics in various fields. This section narrows the scope of this thesis and explores the diverse challenges associated with Embodied Intelligence in soft robotics.

1.3.1 From Conventional Rigid Robots to Biological Systems

When the general public thinks of artificial intelligence, they typically envision a robot. However, a robot is not necessarily intelligent in the same way humans are. Artificial intelligence is more accurately described as a set of mathematical techniques that, at best, can imitate human intelligence in very specific tasks. Robots are programmed systems designed to perform particular tasks, and their intelligence is confined to the context of these tasks. Traditionally, intelligence has been associated with cognitive abilities and problem-solving, often focused on the brain and information processing. However, this perspective is incomplete; it is also possible to discuss intelligence at the level of the body, as we will explore in this section.

When comparing conventional rigid robots against biological systems, three main key differences are observed [HNF23].

Conventional robots rely on simplified rigid dynamics, which simplifies modeling and control. In the industrial context, it enables to obtain reliable and precise robots. In contrast, biological systems are soft and exhibit non-linear dynamics. The use of soft materials in complex morphological systems can introduce interesting features, as seen in nature. For example, the flexibility of an octopus's tentacles allows for intricate manipulation and exploration, while the soft body of a worm enables it to navigate through confined spaces. These non-linearities directly contribute to both perception and task execution.

In contrast to biological organisms that seamlessly interact with their environment, rigid robots are typically designed for controlled and simplified environments, often avoiding direct interactions. For example, industrial robots are frequently enclosed in cages to prevent hazardous contact with humans. Biological evolution has shaped organisms to thrive within specific environments, such as fish, which possess streamlined bodies and flexible fins enabling efficient movement through water. However, traditional controllers often struggle in open, dynamic scenarios.

In conventional rigid robots, the controller is considered separately from the body. It dictates what the body should do. In biological organisms, the boundary is much more blurred: the controller and the controlled body are not explicitly detached from one another. Biological organisms can solve problems efficiently in real-time without relying on a central controller. They utilize their bodies to perform computations typically attributed to the brain, a concept known as morphological computation. This synergy between the brain and body allows for efficient and adaptive control. In geckos, for instance, the passive adhesive properties of their feet enable them to climb smooth surfaces without continuous active control [Wan+15]. Mountain goats can navigate steep, rocky terrains with agility and precision thanks to the specialized structure of their hooves and legs [Aba+19], which contribute significantly to stability and movement, allowing for efficient and adaptive control with minimal active brain involvement.

These three principles are typically actively avoided in conventional rigid robotics, while biological organisms, shaped by long evolutionary processes, seem to favor these phenomena. They help identify the key challenges to be addressed in designing effective soft robot bodies, simplifying robot control in terms of control parameters [Las22]. These principles are commonly encapsulated within the concept of Embodied Intelligence.

1.3.2 Embodied Intelligence in the Soft Robotics and Artificial Intelligence Communities

Embodied intelligence is a concept specifying the need for intelligence to be developed in the physical world, seeing sensory and action systems as fully integrated with cognitive processing. The concept was formulated to make the interaction of agents more immersive with their environment, based on lessons learned from observing the psychological development of human babies [SG05]. In order to be able to learn as humans do, embodied agents able to perceive the world in the first person along with a simulation of the agent environment are needed. Embodied intelligence is nowadays widely studied in the robotics field especially for learning high level control policies for solving complex tasks using reinforcement learning [KBP13]. Unlike its rigid counterpart, Embodied Intelligence is still an emerging topic in soft robotics. However, it is clear that intelligent behaviors can be rapidly learned by agents whose morphology are well adapted to their environment [Bro91].

Within the soft robotics community, the term Embodied Intelligence is gathering increasing attention. It is a profoundly interdisciplinary field at the intersection of materials science, artificial intelligence, cognitive science, and robotics. Figure 1.2 summarizes the most common aspects of Embodied Intelligence identified through interviews conducted with 200 researchers across different fields [Sad+22].

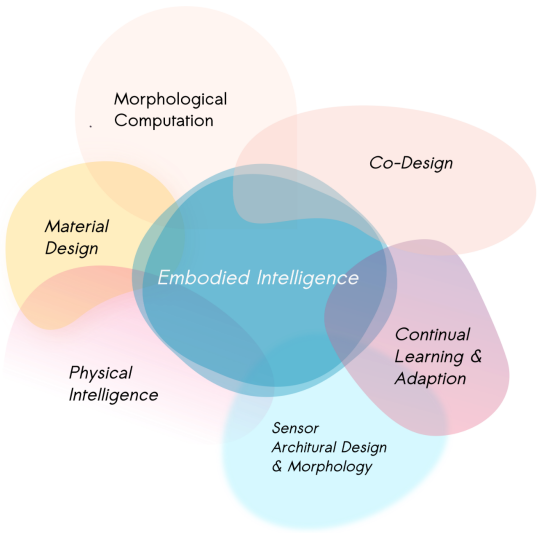

Figure 1.2: Key aspects of Embodied Intelligence, reproduced from [Sad+22].

Various concepts such as Morphological Computation and Physical Intelligence appear in this Figure. We provide insights about them below. The authors of [Sad+22] highlight the ongoing challenge of co-designing the body and controller of soft robots to leverage material properties and dynamics for achieving desired functionalities with minimal control input. Despite numerous isolated studies on Embodied Artificial Intelligence, a common theoretical framework remains lacking, despite recent efforts in this direction [Zha+24].

Among them, Morphological Computation frameworks involve studying how the inherent physical properties of a system can aid in solving computational problems, a functionality typically attributed to the controller [Gha+21]. The emphasis is then on simplifying control and

data processing. General Morphological Computation frameworks use reservoir computing, a machine learning technique for managing complex time-series data. This approach converts input data into higher-dimensional dynamic representations, enhancing the data nonlinear characteristics and complexity. Research has shown that incorporating sensor feedback mechanisms into the design of soft robots enables for the generation of adaptive motion patterns without the need for complex controllers [HNF23], as illustrated in Figure 1.3.

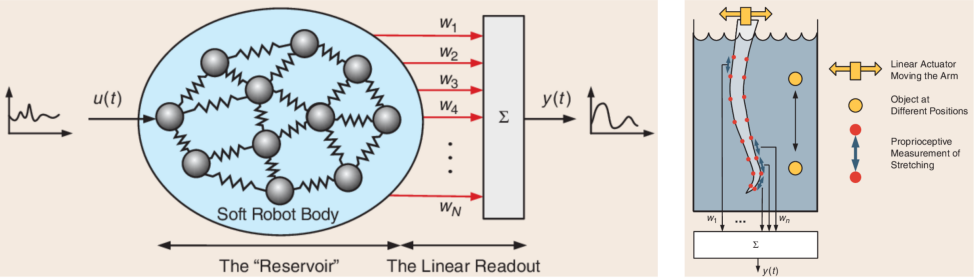

Figure 1.3: Illustration of Morphological Computation framework from 1.3. The soft body is used as a reservoir and a single readout layer is trained using linear regression to map actuation input u to target output y (left), enabling a soft octopus silicone arm to detect objects by only observing the distance between red points on its body (right).

Most efforts from the soft robotics community are focused on Physical Intelligence, which involves directly encoding sensing, actuation, control, memory, logic, computation, adaptation, learning, and decision-making into the body of an agent [Sit21], thereby facilitating control through the body. The goal of Physical Intelligence is then to lower costs, improve response times, and enhance the resilience of robotic systems. Although the terms Morphological Computing and Physical Intelligence are often used interchangeably, it can be misleading. Most of robot designs aim to leverage the non-linear dynamics of soft bodies to facilitate control functions rather than directly using them as computational resources. A few notable example of Physical Intelligence are described as illustration. The beach animals developed by the artist Theo Jansen [lin+], are able to perform complex walking motions on the sand from a simple wind actuation (see Figure 1.4.A). In [Dro+21], researchers propose a mechanical method to control the gaits of soft-legged robots using simple pneumatic circuits without any electronic components, reducing the controller to a constant source of pressurized air (see Figure 1.4.B). Instead of using complex controllers for soft grippers with multiple grasping appendages, researchers [Bro+10] have developed an efficient gripper that conforms to various object shapes using granular jamming (see Figure 1.4.C).

In practice, robots require complex and adapted controllers to interact effectively with their environments, which emphasizes the integration of the body, brain, and their interactions with the physical world. A significant portion of the artificial intelligence community investigates the principles of environmental complexity, evolved morphology, and the learnability of suitable controllers through evolutionary processes. These approaches utilize evolutionary algorithms, inspired by Darwinian evolution, for evolving robot morphologies. They are carried out in simulation due to the challenges of conducting large-scale physical experiments on evolution and learning.

For example, studies from [Gup+21] and [Bha+22] demonstrate the intricate connection between environment and morphology to facilitate learning the controller on new tasks respectively on limbs-made and soft creatures, as shown in Figure 1.5. However, while these methods

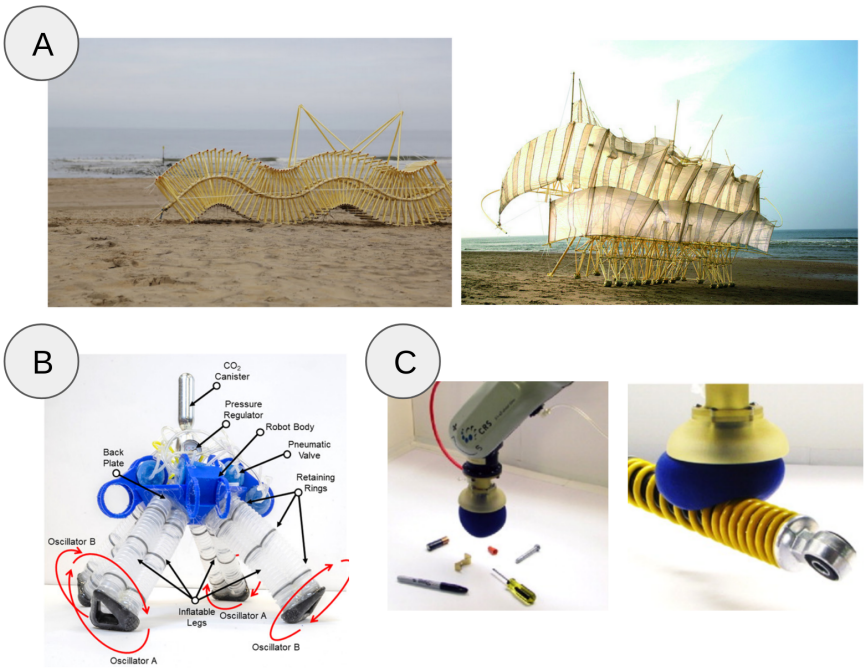

Figure 1.4: Illustration of Physical and Embodied Intelligence. A) Beach animals made with PVC tubes, zip ties and recycled plastic bottles [lin+]. B) Quadruped robot with onboard soft valves powered by a constant airflow and pneumatic oscillators used for controlling the motions of each diagonal leg pair for forward walking [Dro+21]. C) Universal gripper using granular jamming to conform to any object shape [Bro+10].

overcome human subjectivity in design, fabricating the resulting designs remains a challenge.

In this thesis, we primarily focus on applications involving Physical Intelligence and introduce tools to explore Embodied Intelligence in soft robotics while adhering to strict fabrication constraints. Specifically, we ensure that computationally generated designs are manufacturable using known fabrication techniques.

1.4 Soft Robot Design

1.4.1 Soft Robots are Primarily Designed Through Bio-Inspiration and Intuition

The design of soft robots has been primarily influenced by biology through bio-mimetic, which involves replicating the capabilities of natural systems by imitating their motions, appearance, and behaviors [SH23]. Well known soft robot designs were thus inspired by the elephant trunk [HW03] (see Figure 1.7.A) or the tentacle of the octopus [Las+12] (see Figure 1.7.B).

There is still no universally adopted systematic methodology for the simulation or design of soft robots. Unlike conventional rigid robotics, where well-established design frameworks and common modeling techniques exist, the field of soft robotics remains in an exploratory phase, lacking standardized practices and components. Soft robot design is thus still primarily investigated both from the observation of nature and human intuition.

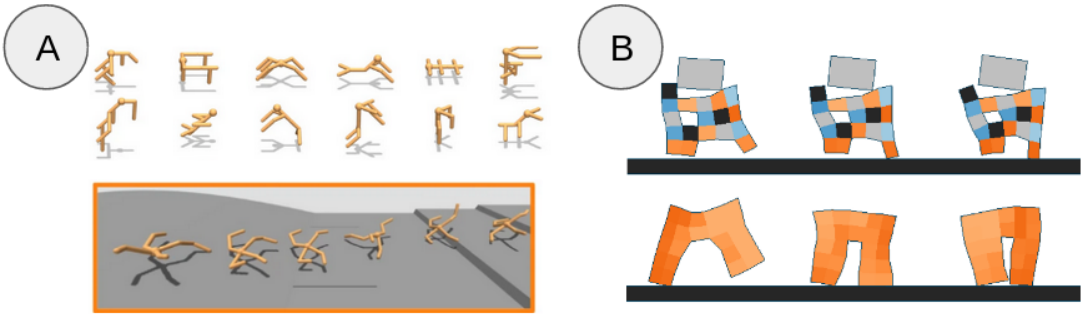

Figure 1.5: Examples of embodied artificial intelligence. A) Example of evolved morphologies (up) and trained morphology on a locomotion task (down) obtained using the framework for co-evolving animals morphology and controller from [Gup+21]. B) Example of evolved morphology and controller for locomotion with (up) or without (down) carrying an object from [Bha+22].

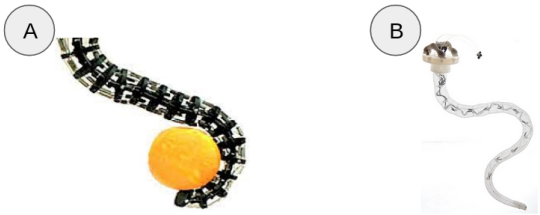

Figure 1.6: Bio-Inspired robots. A) Soft manipulator inspired by the elephant trunk [HW03]. B) Soft manipulator inspired by the octopus tentacle [Las+12].

Bio-inspiration is often favored due to the evolution of biological organisms to survive in specific environments and adapt to particular tasks. However, directly drawing inspiration from biological organisms for robot design presents limitations. Fabricating robots at the complexity level of biological organisms is currently unachievable, and the environments and tasks considered for our robots may not mirror those found in nature.

The ultimate goal is to identify fundamental concepts governing bio-inspired organisms and apply them to robot design using available manufacturing methods. Such designs may initially rely on intuition and evolve to adapt to tasks and environments. Given the vast design spaces encountered in soft robotics, empirical design is impractical, highlighting the need for computational design in soft robotics.

1.4.2 Computational Design Methodology

The computational design of soft robots involves the intricate interplay between several components, making it a deeply interdisciplinary field. The interplay between these components is displayed in Figure 1.7. These components are introduced below and further discussed in the background section:

- **Simulation Environment and Models**: This entails having simulation environments and models for the robot body, its controller, and its environment. It requires access to models that characterize both control and design, as well as the ability to account for interactions with the external environment such as contacts. Algorithms for optimization are typically

simulation-intensive, so finding a balance between simulation speed and accuracy for the simulation-to-reality transfer is crucial. More information on the different simulation tools available is provided in the Background section.

- **Fitness Functions**: Designing objective functions to evaluate the performance of a robot is essential. These functions should be linked to the robot's controller ability to perform its task in its environment. This involves either decomposing the task into multiple sets of mechanical constraints to be met or testing the controller on the task in simulation for a given set of design parameters. More information on control models is provided in the Background section.

- **Optimization Parameters**: Choosing which parameters to optimize involves a trade-off between the size of the design space, allowing for the emergence of novel shapes, and maintaining a feasible optimization problem. More information on common materials, actuators, and sensors in soft robotics is provided in the Background section. Fabrication constraints must also be taken into account.

- **Optimization Method**: A method of optimization is needed to guide the search for the best design. The choice of method (e.g., gradient, stochastic, ...) is partly dictated by the choices made for the previous three ingredients.

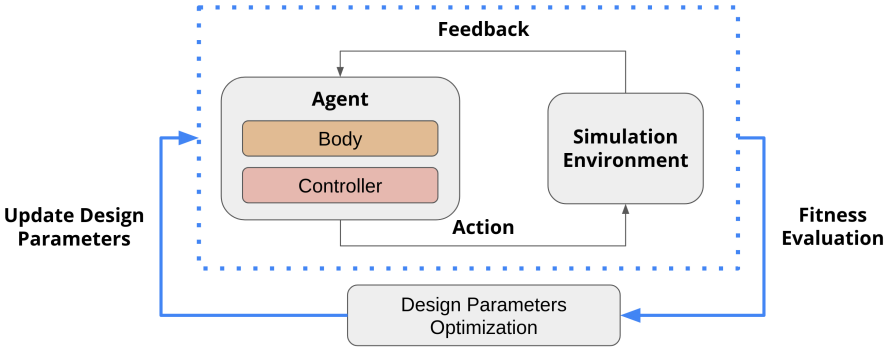

Figure 1.7: Control and design co-optimization workflow in the framework of soft robotics.

This thesis work establish different computation design methods for evolving controller and design, and apply them to diverse scenarios.

1.5 Manuscript Outline and Contributions

In this introductory chapter, it is emphasized that the link between the robot design, controller, and environment cannot be overlooked, as they are typically considered separately in a classical intuitive soft robot design process. Yet, they are interconnected.

One challenge lies in transferring design and controller optimization results obtained using numerical simulations that may lack precision or realism, in order to make the optimization tractable. We justify the use of the Simulation Open Framework Architecture (SOFA) simulator based on its proven precision in previous studies on physical robot control, which aligns with our goal of achieving sim-to-real transfer. This is further discussed in the Background section of

this manuscript, which introduces several problematics in term of computation time, leading to the need for developing surrogate of the simulation.

The objective of this thesis is then to explore the optimization of robot design to facilitate its control strategy and enhance its performance in specific tasks and environments. This work introduces novel general tools for exploring designs of soft robots with regards to their performance and addresses some challenges for co-optimizing control and design for soft robots. The organisation of the manuscript and the different contributions of this thesis work are the following:

- **Chapter 2: Background and Related Work.** This chapter introduces the various mathematical tools and models that can be used for modelling, control and design in soft robotics. In addition, it introduces the current state of the art on computational design optimization of soft robots.
- **Chapter 3: Optimization of the Parametric Design of Soft Robots Assisted by Surrogate**. Development of a generic framework for optimizing parametric design of soft robots. It has been applied in several scenarios with an emphasis on the simulation to reality transfer. Particular attention has been put on the design of fitness functions for assessing the performance of soft grippers and manipulators.
- **Chapter 4: Condensed FEM Modeling for Embedded Control and Design Optimization**. Development of a single framework for learning a compact mechanical representation of a robot based on mechanical matrices projected in the constraint space. It enables to mitigate the computational burden of the simulation while keeping a precision close to the one offered by FEM simulation. The relevance of using this single differentiable description for both design and embedded control is highlighted in several applications.
- **Chapter 5: Towards a General Framework for the Emergence of Embodied Intelligence in Soft Robots**. Discussion of this thesis results and introduction of ongoing work aimed at leveraging the condensed FEM model to co-optimize design and high-level controllers. Additionally, methods for obtaining compact, easily fabricable representations of more free-form designs are investigated.

These topics have been covered in multiple scientific publications to conferences or journal, as either the principal author or co-author:

- Stefan Escaida Navarro, Tanguy Navez [1], Olivier Goury, Luis Molina and Christian Duriez, "An Open Source Design Optimization Toolbox Evaluated on a Soft Finger", RAL, 2023 [Nav+23]
- Etienne Ménager, Tanguy Navez [1], Olivier Goury and Christian Duriez, "Direct and inverse modeling of soft robots by learning a condensed FEM model", ICRA, 2023 [Mén+23]
- Tanguy Navez, Baptiste Liévin, Quentin Peyron, Stefan Escaida Navarro, Olivier Goury and Christian Duriez, "Design Optimization of a Soft Gripper using Self-Contact", Robosoft, 2024 [Nav+24]
- Tanguy Navez, Etienne Ménager, Paul Chaillou, Olivier Goury and Christian Duriez, "Modeling, Embedded Control and Design of Soft Robots using a Learned Condensed FEM Model', Transactions on Robotics, under review

[1] equal contribution to first author

Along with these projects, two open-source plugin for the simulation software SOFA have been released:

- The "SoftRobots.DesignOptimization" plugin, implementing an interface for design optimization and calibration of soft robots simulation asissted by surrogate [Navb]
- The "CondensedFEMModel" plugin, implementing components for learning a condensed model of a soft robot and leverage it in several applications, co-developed with Etienne Ménager [Tan]

Chapter 2

Background and Related Work

The computational design of soft robots within the framework of Embodied Intelligence is challenging, as it encompasses various topics related to soft robot modeling and simulation, control, and design. This chapter introduces the state of the art for each of these three topics.

2.1 Soft Robot Modeling and Numerical Simulation

Numerical simulation consists in approximating the analytical solution of a physical model, which is an abstract representation of the reality. This representation captures the essential aspects of a system to make understanding and analysis more accessible.

Whether for control or computer-assisted design applications, having an accurate numerical simulation of soft robots is a crucial component for achieving successful applications. This Section introduces the specific features of the soft robot simulation followed by an overview of both the different theoretical models and simulators used for modeling and simulating soft robots. Then, both the continuous mechanical models and the SOFA Framework implementing it, which are mainly used in this thesis, are further developed. Finally, we see how reduced order modeling framework are used for reducing the simulation computation time which is critical in applications leveraging simulation which require real-time interactions.

2.1.1 Soft Robot Simulation Features

In robotics, simulation primarily enables to control a physical robot by predicting the best actuation strategy to achieve certain user-defined goals. For this purpose, a digital twin, i.e. a model calibrated to be an accurate representation of the physical prototype, is needed. We are then interested in models linking the robot's actuation a to its state description X. Direct models, described using the equation $X = f(a)$, compute the robot state depending on the constraints imposed on the robot. In the inverse models described as $a = f^{-1}(X)$, we are interested in how the constraints evolve based on the robot state description. Each of these models have their own set of applications. Direct models are mostly used for interactive simulations, while inverse models are mostly used for control or robot state estimation. Simulation is also widely used as an assisting tools for design. Under the condition of having a model able to capture variation in

design parameters, simulation enables to evaluate design performances.

Modeling and simulating soft robots pose unique challenges compared to classical rigid robots due to the intrinsic complexity associated with soft materials and deformable structures. In rigid robotics, standardized efficient and quick to compute analytical models are available. This is explained by their structural assumptions, fixed robot shape and limited deformations, that remain verified at all time. In comparison, soft robots lack such standardized model. They have in theory an infinite number of degrees of freedom, due to the inherent complexity associated with deformable structure made with non-linear materials. Variable stiffness, a common feature in soft robotics, further complicates the modeling process, as the robot behavior depends on the changing stiffness of its materials. Interactions with fluids introduce fluid-structure coupling and hydrodynamic challenges. Additional complexity also comes from the frequent operation of soft robots in unstructured environments as their interactions with objects or surfaces involve intricate contact mechanics.

The nature of soft robot implies the lack of analytical solutions in most case. One solution for addressing these challenges is to implement numerical models that use numerical solvers to solve for the robot deformations on a discretized geometry by making approximations. The diversity and complexity in soft robot designs, materials, and actuation methods makes it challenging to develop generalized modeling framework across various soft robot platforms. Then, the most efficient method can vary from one system to another. From shape-specific methods to the more generic ones, we describe the different modeling classes in the next Section.

2.1.2 Classification of Soft Robot Modeling and Approaches

As developing new soft robot models exceed the scope of this thesis, only a short review of the main soft robot modeling approaches is considered. Complete state of the arts reviews of soft robot modeling strategies are available in [Arm+22; SD22; Qin+23; Men+22]. These strategies can be roughly classified in 4 main categories although they may sometimes overlap.

A - Geometrical Models

Geometrical models are based on strong geometrical assumptions on the deformed shapes undertaken by the soft body. The Piece-wise Constant Curvature (PCC) Model is one of the most popular ones, available under the assumption of constant curvature for soft robot shapes composed of beam elements. As it is an analytical model, it is fast to compute and easy to obtain gradients for control. Thus, it has historically been one the most used method in the soft robotics community [WJ10]. Examples of applications include soft manipulators [Mar+14] [CFW18], as shown in Figure 2.1. However, the constant curvature assumption becomes invalid when external forces are applied on the robot, making it unsuitable to handle gravity and interactions with the environment through contacts.

B - Continuum Mechanical Models

Continuum mechanical models are obtained from classical elasticity theory and characterized by a continuous robot state space i.e. the robot is modelled as a continuum where the properties of interest (stress, strain, ..) vary continuously. By establishing the connection between particles motion and deformation tensors, these models are derived based on the constitutive laws of materials. For an introduction of the basic notions related to continuum mechanics, we refer the reader to the following handbook [Sal01].

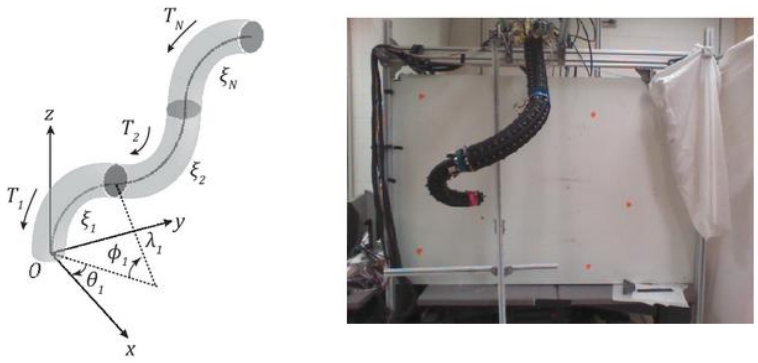

Figure 2.1: Schematic of Piecewise Constant Curvature forward kinematics modeling applied to the forward kinematics (left) of the OctArm, a multisection continuum manipulator (right) from [CFW18]. The soft structure is represented with a fixed number of arcs described by three parameters, namely the curvature radius, the arc angle and the bending plane.

In soft robotics, a significant obstacle lies in enhancing the efficiency of these methods to enable real-time usage without compromising realism. Thus, the development of specific numerical resolution schemes targeted at continuum mechanical models have an important place in the soft robot modeling literature. This partly explains why *modeling* and *numerical resolution* are terms used regularly interchangeably in the soft robotics community while both of them are well differentiated in the continuum mechanics community. Given their importance in this thesis work, soft robot continuum mechanical models and associated numerical integration methods are described in more details compared to other modeling categories.

As simulating an infinite number of degrees of freedom is not conceivable, there is a necessary step of discretization of the continuous structure either in particles or in a spatial mesh. We start by introducing below the generic 3D modeling and joint numerical solvers. A common way to reduce the resolution space being to consider geometrical assumptions when possible, we then introduce the main modeling strategies for this purpose.

B.1 - Generic Continuum Mechanical Modeling and Numerical Resolution

The Finite Element Method (FEM) is the most known generic 3D numerical technique, applicable without any assumption on the shape of the structure. It consists in discretizing the robot geometry in a mesh of finite elements and solving the underlying non-linear equations of the continuum mechanics for each of these elements. This is why FEM is also commonly referred as a mesh-based method. Within each element, approximate solutions to the governing partial differential equations are formulated using polynomial interpolation functions. By assembling the contributions from all elements, a system of algebraic equations is derived, which is then solved numerically to obtain the solution for the entire domain. In practice, FEM models have been widely used by the soft robot community and shown to be effective to address challenging modeling and simulation tasks, involving for instance factors such as gravity forces, important load constraints and contact forces. For instance, it has been applied to very different shapes and scenarios, such as to the modeling of a parallel robot actuated by cables [Dur13a], soft tentacles robots [CED17; Mor+20a] interacting with their environment, a pneumatically actuated crawling robot for locomotion [GD18], or a parallel continuum haptic device [Koe+23]. Some examples are shown in Figure 2.2. Nevertheless, one drawback of these methods is that they are computationally expensive because of the large number of discrete degrees of freedom to

handle. They also rely on a lot of hyper-parameters (mechanical and geometric parameters, mesh discretization, boundary conditions on forces and contacts), requiring extensive experience from the user to be used effectively. Hence, continuum mechanical models with geometrical assumptions are commonly preferred when possible.

Figure 2.2: Examples of soft robots controlled using FEM: Diamond robot made in silicone and actuated by four cables [Dur13a] (left), a soft tentacle robot actuated by cables [CED17] (middle) and a crawling robot actuated by four pneumatic cavities [GD18] (right).

Another well-known model is the Material Point Method (MPM) [Jia+16], where a continuum body is described by a set of small Lagrangian elements called material points. Material points store all the information relative to the robot state and a fixed background mesh is used to compute gradients of deformations. It is a direct competitor to FEM modeling as a generic method to simulate any robot shape. ChainQueen is an open-source simulator implementing a variant of MPM specifically targeted at soft robot simulation [Spi+23]. They propose a hybrid Eulerian-Lagrangian approach where particles and grid representation are used interchangeably. It is particularly effective for handling large deformations and material interfaces, but is more computationally expensive than FEM as the background grid must be reset at the end of each MPM calculation steps. This makes it particularly suitable for modeling locomoting soft robots. One interesting characteristics is that computing gradients is fairly easy relative to the fixed background grid, opening the path for gradient-based control and design co-optimization applications. Compared to FEM, there are however less examples of physical soft robot controller derived from MPM in the bibliography. Some examples of applications are a soft beam actuated by magnetic field gradients [Dav+23], or pneumatically actuated locomoting soft robots shapes generated and controlled using MPM [Kob+23] (see Figure 2.3).

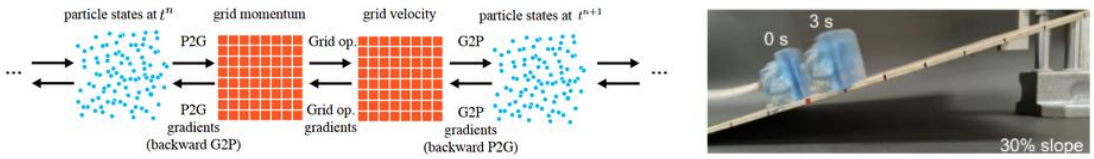

Figure 2.3: Illustration of the Material Point Method for simulating soft robots: particle-to-grid scheme and grid-to-particle transfers implemented in the ChainQueen simulator [Spi+23] (left), and an example of a locomoting soft robot with a MPM-based controller [Kob+23] (right).

B.2 - Continuum Mechanical Models with Geometrical Assumptions and Numerical Resolution

B.2.a - Modeling

A large set of continuum mechanical models consider specific assumptions on the shape of the robot. Some approaches have been specifically proposed for 1D slender structures, such as beams and rods, and 2D surface structures such as membranes, shells or plates. Although the former are extensively utilized as the geometrical hypothesis are suitable practical applications (catheters, manipulators), there are relatively few examples of applications for the latter in the bibliography.

The 1D slender structure category include widely used models such as the Kirchhoff [Dil92] and the Euler-Bernoulli [Sha97] rod theories, as well as the Cosserat beams [**renda_discrete_2016**], and have been shown to be the most efficient techniques in most cases for modeling systems that can be described as combinations of slender-shaped structures. Both these models are expressed as systems of partial differential equations with one spatial dimension.

The Kirchhoff and Euler-Bernoulli theories both focus on global deformations like bending and torsion. The modeled structure is then simplified to a uni-dimensional element that primarily flexes within the plane of deformation. Kirchhoff rod theory neglects transverse deformations, by assuming that cross-section remains plane and normal to the deformed beam axis during deformation, while Euler-Bernoulli beam theory considers these deformations along with longitudinal bending, as long as transverse deformations stay small. Kirchhoff theory is then applicable primarily to thin rods where transverse deformations are negligible, as it is the case for catheters for instance [Kra+17] (see Figure 2.4.A). On the other hand, Euler-Bernoulli theory is more versatile, accommodating thicker beams and situations where transverse deformations cannot be disregarded. It has successfully been applied to the modeling of a cylindrical soft arm [Ols+20] bent by external load (see Figure 2.4.B). This approach however lacks to capture local deformation in the structure.

An alternative for accurate modeling of such slender structures is the Cosserat rod theory. It is a generalization of Kirchhoff rod theory considering structures as a line along which is continuously stacked a set of rigid cross Sections. For each of these Sections, local deformations and rotations are computed. The ability of Cosserat models to handle kinematics and dynamics of slender shaped objects undergoing important deformations has been demonstrated on a wide range of soft robots and actuated objects, such as an octopus arm [Ren+18b] (see Figure 2.4.C), underwater soft robots [Ren+18a], or threads in surgical simulation [Pai02]. Although it is considered as a geometrically exact modeling approach, it is however often used jointly with complex numerical solvers in real-time applications for computational time reason.

In all cases, a common limitation of these kind of approaches is that all deformations are not taken into account as cross-sections are assumed to be (or almost) rigid. In certain cases, this may not precisely reflect the behaviour of some materials, large transverse deformations implied by some actuation strategy such as pneumatic cavities, or of specific robot configurations.

2D surface structures modeling methods are targeted at thin objects which motion can be abstracted through the motion of a surface, such as membranes, plates or shells. In these cases, the thickness of the soft structure is considered constant under loading.

For instance, a combination of the Cosserat model and the Reissner shell model [SF89] is used to model the shell of a soft underwater vehicle inspired by cephalopods in [Ren+15] (see Figure 2.5). Exploiting axis symmetry enables reducing the general configuration space of the shell which is modeled as a continuous assemble of rigid fibers along the median of the shell. Other example include the shell of a growing soft robot [PRZ+23].

Although this style of models has been widely applied in medical simulation, for example to model organ membranes, there are few examples of applications in soft robotics. This is

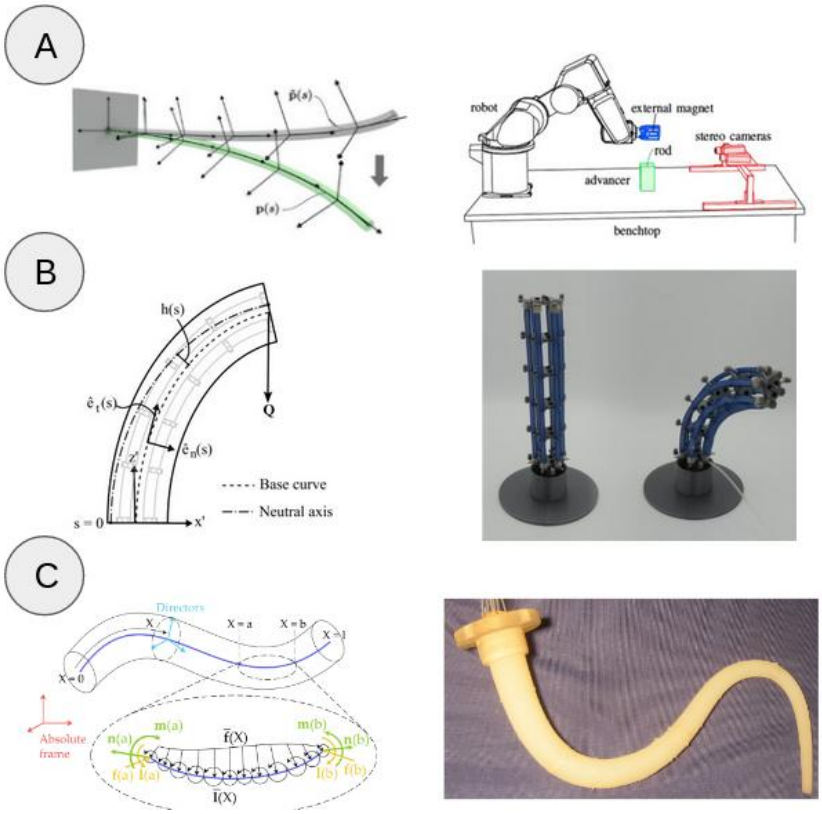

Figure 2.4: Illustration of wire-like soft structures modeling. A) Schematic of a rod modeled with Kirchhoff theory as a space curve with a local coordinate frame attached to each point (left), and illustration of the setup proposed in [Kra+17] for guiding the magnetic rod with a robot-manipulated magnet (right). B) Schematic of Euler-Bernoulli beam modeling (left) for a soft robot arms (right) from [Ols+20]. C) Schematic of Cosserat modeling from [Arm+22] (left) alongside an octopus arm controlled via forward kinematics derived from Cosserat theory (right) from [Ren+18b].

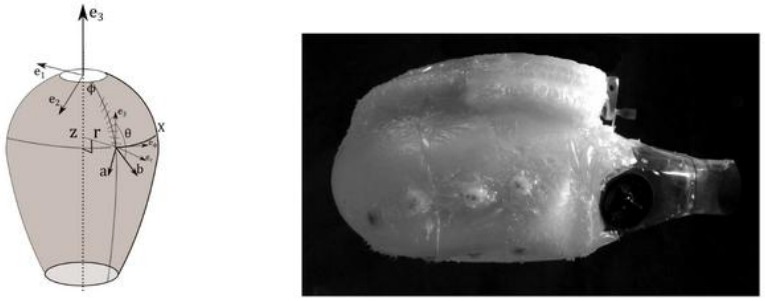

Figure 2.5: Schematic of a shell modeled using Reissner modeling (left) and a cephalopod-inspired pulsed-jet soft robot for locomotion (right) from [Ren+15].

explained by the low number of soft robot designs where 2D structure hypotheses are valid.

B.2.b - Numerical Resolution

Analytical solutions are available only in some very specific conditions. 1D slender structures subject to concentrated external loads are one example. In most cases, analytical solutions are not available, and numerical solvers are used.

Approaches for solving continuum mechanical models with geometrical assumptions are classified into non-energetic and energetic methods. A complete review of the numerical resolution of director-based formulations is proposed in [Arm+22]. For non-energetic methods, the modeling formulation is reshaped as a set of partial differential equations with boundary conditions that are directly discretized on a spatial grid. Then standard numerical analysis techniques are employed for the resolution, such as finite differences or shooting methods. Energetic methods rely on the principle of energy conservation whatever the chosen coordinate system, as stipulated by Lagrangian mechanics. The configuration is parameterized by some vector fields, reduced on a truncated functional basis of space-dependant vectors. This defines a set of Lagrange differential equations in time. One powerful method is the geometrically exact FEM (GE-FEM) [SV88] where material points of the classical medium are replaced by the rigid Sections of the Cosserat model, and the obtained system is solved with FEM.

C - Particle Models

As for continuum mechanical models, particle models are also obtained by deriving physical laws. However, they consider a finite discretization of the body in discrete materials components. This is why they are also sometimes referred as mesh-less methods.

For instance, lumped-mass models consists in representing the structure as a set of discrete masses connected by a network of springs. It has been applied to model the large deformations of a continuum surface actuated by robot arms [Hab+20] (see Figure 2.6.A) or the modeling of tendon-driven medical robotics catheter in [Jun+11]. Although it provides a fast solution regarding computation time, it requires specific optimization for calibrating the model parameters to the robot mechanical properties.

Other examples are pseudo-rigid models represents soft bodies as series of rigid links which are connected by joints. These kind of models are usually convenient for modeling hybrid robots combining both soft and rigid parts. Relevant applications are to be found in dexterous catheter manipulators [Kut+11](see Figure 2.6.B) [KP13] in the early stage of the soft robotics fields.

Although easier to implement than continuum mechanical methods, particle models also suffer from the computational complexity related to handling a large number of degrees of freedom, and unlike the aforementioned methods, they struggle to model certain aspects such as boundary conditions because of their inconsistent discretization.

D - Learning-Based Models

Learning-based models rely on machine learning. Complete reviews are available in [WLK21] [Kim+21]. These models are trained from large sets of data. These data can be extracted from different sources, such as numerical simulation results harvested using any other aforementioned modeling strategy, or measures extracted from sensors on a physical prototype. Most framework consists in training shallow neural networks, such as a MultiLayer Perceptron, for using it directly as a direct ($X = f(a)$) or an inverse model ($a = g(X)$) (see Figure 2.7.A).

More complex deep architectures, such as Recurrent Neural Networks (RNNs), have also been applied for capturing complex functions involving intricate time and mechanical coupling such as when considering soft robot dynamics. However, the recurrent nature of RNNs makes

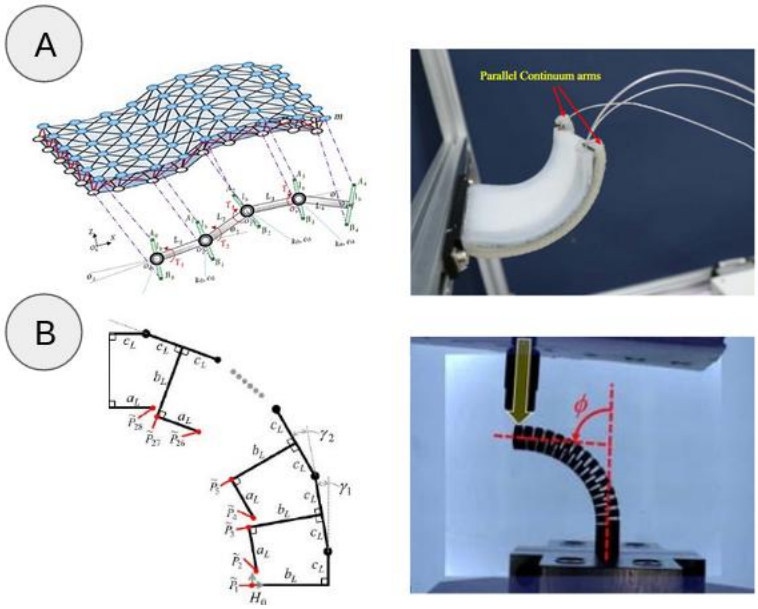

Figure 2.6: Illustration of two particle models. A) Schematic of a lumped-mass discrete model integrated with an arm model along one side (left) and the controlled curved-shape soft surface actuated by two continuum arms (right) from [Hab+20]. B) Schematic of a pseudo-rigid model (left) applied to the control of a dexterous catheter designed for minimally-invasive surgery (right) from [Kut+11].

training on long sequences challenging due to the gradient propagation across many time steps, which can lead to vanishing or exploding gradient issues, slowing down or even hindering learning convergence. Thus there are few working examples in the bibliography [Kim+21]. For instance, a RNN is applied to learning the forward dynamics of both a tendon-driven and a pneumatically-driven soft manipulators in [Thu+17] from simulation data, and authors showed that the open-loop controller built from this learned RNN is not robust to external disturbances. A model-based learning method, this time learned from real-world data, enabled to reach a better stability in [Thu+19] and was validated on the the closed-loop control of a pneumatically actuated soft manipulator (see Figure 2.7.B).

Learning-based models can be applied to any robot shape and scenario. Once trained, the high prediction speed of these models is suitable for real-time control. Nevertheless, their training is tailored to the specific scenario and shape met during the data generation, limiting their applicability to other contexts.

Some data-driven approaches employ data jointly with modeling tools for efficiently approximating the physical model. In this case, these methods are referred as model order reduction. These methods are further discussed in Section 2.1.5.

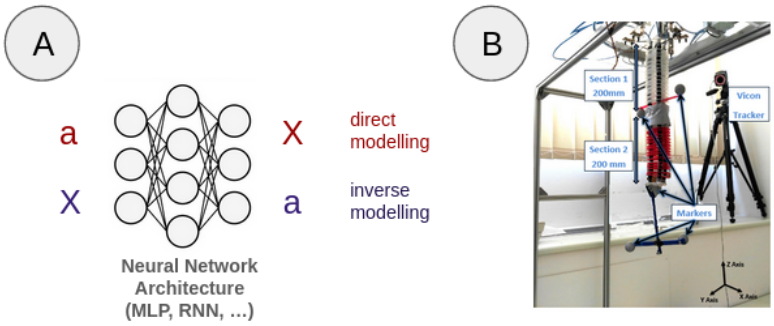

Figure 2.7: Illustration of learning based models. A) Neural networks are trained to learn direct or inverse models. B) Pneumatically-driven soft manipulator controlled using a trained RNN as the forward dynamics model from [Thu+19].

2.1.3 Physics-Based Simulators and SOFA Framework

In this thesis work, we rely SOFA for the simulation of soft robots. This open-source software is collaboratively developed by several research teams. The DEFROST team, where I conducted this work, has been its main contributor these last years. After a short introduction of the main alternatives physics-based simulators used in soft robotics, this Section describes the SOFA framework and its features.

Main features of SOFA framework:

- FEM simulation
- Multi-model representation and mappings
- Libraries of actuator and sensor models
- Inverse models for soft robot control

Available Physics-Based Simulators

SOFA has an original positioning between real-time simulators for robotics, such as MuJoCo [TET12] or Gazebo [KH04], and FEM computation software like ANSYS or COMSOL.

The FEM is a standard for many popular multi-physics simulation software, such as Abaqus, ANSYS, COMSOL. These software are user-friendly frameworks providing a wide spectrum of tools for tackling different physical problems, including heat transfer, electric conduction, magnetism and fluid flow [Col+21]. They have also been demonstrated for modeling soft robots and for soft robot design studies. However they lack features necessary for robotics applications, like actuators and sensors modeling or controllers. Moreover, as they are not specifically tailored for soft body simulation, their computational cost is a non affordable burden in real-time applications.

ChainQueen [Hu+18b] is another alternative simulator. Implementing a variant of MPM, it is fully differentiable, but it is limited in its capabilities, proposing only limited actuator models and material constitutive laws.

Other well-known state of the art simulators are specifically targeted at simulating the dynamics of deformable objects for control purpose, and rely on different approaches for the modeling and solving the dynamics. Among them, some are using particle-based modeling (MuJoCo [TET12], Flex [Mac+14]) or continuum-based modeling solved with FEM (Bullet [Bai], SOFA [Fau+12]). Although simulation is widely used as a mean to evaluate designs or learning controllers, few works consider an in-depth analysis and comparison of the simulation to reality transfer. Recently, these simulators have been compared on the cloth manipulation tasks [Bla+24]. On their benchmark tasks, SOFA showed to present lower errors than the other aforementioned simulators.

Recent significant release include the Sorotoki toolkit [CPN24], implementing FEM simulator and algorithms for control and design of soft robots. In their work, they also propose a complete comparative review of the current simulation software available. Both SOFA and Sorotoki are the more complete platforms regarding the range of models (hyper-elastic, tendon, fluidic), considered tasks (locomotion, manipulation) and control algorithms available. Both simulators have also been widely validated on physical prototypes. Another interesting feature of Sorotoki is that it natively includes algorithms for inverse design of soft robots (more details on this aspect are given in Section 2.3.2). However, unlike SOFA, the focus of the Sorotoki package is not on real-time control.

SOFA Framework

The Simulation Open Framework Architecture (SOFA) software is an open-source C++ library that has been initially presented as biomedical and surgical simulator [Fau+12]. It enables to simulate a large choice of different mechanical models and geometrical descriptions (from curve to 3D meshes), handling multi-physics coupling and precise contact management including self-collision. SOFA implements a scene-based architecture that can be described in a Python script. Robots and environments objects are described with elementary components that are easily interchanged. Components are provided for encoding topologies and contacts, numerical integration, mechanical modeling or rendering. This architecture is especially well-suited for exploring and experimenting with various ideas. It turned out to be particularly convenient for constructing automated and parameterized pipelines to evaluate soft robot designs in this thesis work. Being targeted at real-time simulation of real robots, SOFA also provide tools for interacting with sensing hardware and haptic devices [Fau+12]. The software is built to allow users to develop new components as plugins on top of the core software. Across the years, the software has been updated with several plugins specifically targeted at soft robotics: the SoftRobot plugin [Dur13a] [Coe+17] targeted at soft robot simulation and control which provides a wide range of actuator and sensors models, and its pendant the SoftRobot.Inverse plugin [Coe19] implementing inverse model for soft robot control. Although initially targeted at FEM simulation, SOFA also implement other continuum mechanics based models such as Euler-Bernoulli beam model[Tal+16] and a model combining FEM simulation and discrete Cosserat [ARD21]. Example of soft robots controlled using SOFA for the simulation are shown in Figure 2.2.

One of the strength of SOFA lies in a multiple model representation, where a simulated object can be simultaneously represented by several meshes linked together through mappings. Mappings are non-linear functions enabling to transfer positions, velocities and forces between different representations. For instance, it enables to consider different representation for 2D contact surfaces and simulation meshes, as illustrated in Figure 2.8. It is also convenient to model hybrid robots, as rigid parts can be easily mapped to soft parts.

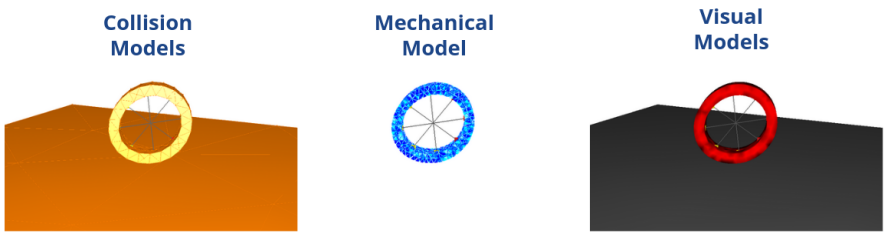

Figure 2.8: Illustration of multi-modal mapping in SOFA, for a circular rolling robot actuated by 4 cables. On the left, models for detecting collisions between the robot and the surface it moves on. In the middle, mechanical model for computing the deformations of the rolling robot. This model is usually more detailed than the collision model. It controls the update of the other representations through different mappings. On the right, models for visualisation purpose.

2.1.4 Modeling and Simulation based on Continuum Mechanics in SOFA

In this thesis work, continuum mechanical models and FEM numerical solvers are used for the simulation of soft robots. This choice is justified mostly by the genericity of the methods and numerous successful applications addressing the simulation to reality transfer in the state of the art. In this Section, the governing discretized equations and numerical integration schemes are introduced.

Usefull concepts:

- Conversion of Lagrangian mechanics equations to a system $Ax = \mathbf{b}$
- Definition of constraints as Lagrangian multipliers
- Resolution for a free configuration $\mathbf{x}^{free}$ and then for the constrained system

Discrete Equations of Motion

The mathematical description of the behavior of soft bodies is obtained from deriving equations based on continuous media principles. As previously explained, various modeling approaches stem from these equations. In each approach, a set of coordinates describing any configuration X of the soft body is required. These coordinates can either be relatives, pertaining to a frame moving with the system (such as in the Euler-Bernoulli beam model or the Cosserat model), or absolutes, referencing a fixed frame (as for 3D volumes). We refer to these coordinates as generalized coordinates and denote them $\mathbf{q}$.

As explained in the previous Section, all the introduced continuum mechanical models can be solved in the framework of FEM. In this case, the type of elements used will differ, depending on the considered geometrical assumptions, as shown in Figure 2.9. In all cases, a mesh defines how the kinematics of each node will be linked to the kinematics of the other nodes through an interpolation function. The number of degrees of freedom considered for the nodes, i.e. the generalized coordinates q, depends on the chosen geometrical assumptions. For instance, frame nodes with 3 translations and 3 rotations are usually considered for 1D beam elements, some rotations elements are included for 2D elements, and most often only translation are considered in the generic case.

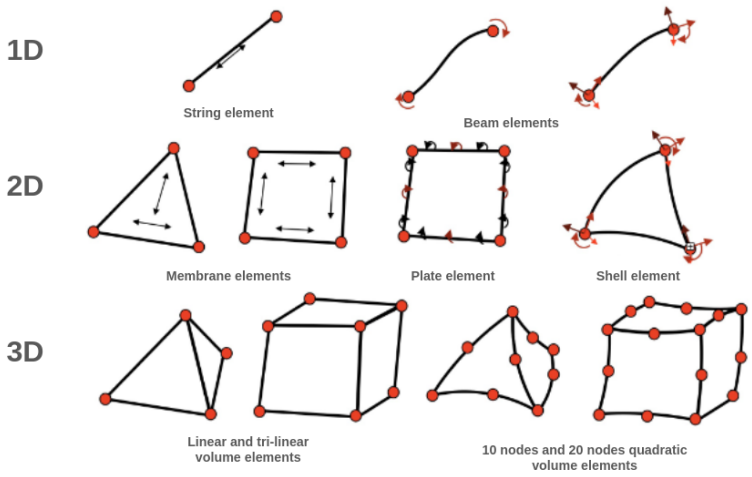

Figure 2.9: Example of elements used in FEM.

In their primal form, the discrete dynamic equations describing a soft body are then written as:

$$\mathsf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{f}_{int}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{f}_{ext} \tag{2.1}$$

where M is the mass matrix, $\mathbf{f}_{int}$ are non-linear internal forces computed from the material laws chosen for the robot that can include viscosity effects, $\mathbf{f}_{ext}$ are external forces such as the gravitational force. All these dimensions are projected in the chosen coordinate space $\mathbf{q}$. Finally, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are respectively the velocities and acceleration of the generalized coordinates $\mathbf{q}$.

When possible, these equations are further simplified under both the assumptions that some nodes of the mesh are fixed and that the velocities of the nodes are low so that inertial forces vanish. This hypothesis is often valid, as it is common to actuate soft robots with low actuation frequency to avoid dynamics effects such as vibrations. However, the first assumption is not valid when considering soft robots for locomotion for instance. We are then in the quasi-static case, and the system of equations to solve is simplified as:

$$\mathbf{f}_{int}(\mathbf{q}) + \mathbf{f}_{ext} = 0 \tag{2.2}$$

Numerical Resolution with an Implicit Euler Scheme

To be solved, the dynamics equations are commonly discretized in time with an integration scheme, over a chosen time step $h = t_{i+1} - t_i$. At each time step, numerical solvers are used to obtain a good approximation to these equations.

Explicit time-stepping schemes, computing the current position $\mathbf{q}_{i+1}$ and velocities $\dot{\mathbf{q}}_{i+1}$ from the values at the previous steps $\mathbf{q}_i$ and $\dot{\mathbf{q}}_i$, enable an easy computation of internal forces. However their stability is conditioned by the chosen time steps h [Dur13b]. On the other hand, implicit time-stepping schemes compute the current dimensions from internal forces defined at the next time step. This kind of approach, at the cost of the inversion of complex matrices, ensures unconditional stability and is adapted for interactive simulations with contact handling. Indeed, when a contact occurs, the velocity of a node change and the acceleration is not defined anymore. This issue can be addressed in an Euler implicit time-stepping scheme by replacing

the acceleration by a change of velocity $d\dot{\mathbf{q}} = \dot{\mathbf{q}}_{i+1} - \dot{\mathbf{q}}_i$. These reasons explain why implicit integration scheme are usually privileged over explicit schemes.

In SOFA, an implicit time-stepping Euler scheme is implemented. Given the current state $(\mathbf{q}_i, \dot{\mathbf{q}}_i)$, it is written as:

$$\begin{cases} \mathsf{M}(\mathbf{q}_i)d\dot{\mathbf{q}} = h\mathbf{f}_{int}(\mathbf{q}_{i+1}, \dot{\mathbf{q}}_{i+1}) + h\mathbf{f}_{ext} \\ \dot{\mathbf{q}}_{i+1} = \dot{\mathbf{q}}_i + hd\dot{\mathbf{q}} \\ \mathbf{q}_{i+1} = \mathbf{q}_i + h\dot{\mathbf{q}}_{i+1} \end{cases} \tag{2.3}$$

At this step, as positions q_{i+1} and velocities $\dot{q}_{i+1}$ are not yet known at the end of the time step, internal forces are computed using a first-order Taylor series expansion for linearizing them:

$$\mathbf{f}_{int}(\mathbf{q}_{i+1}, \dot{\mathbf{q}}_{i+1}) \approx \mathbf{f}_{int}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + \frac{\partial \mathbf{f}_{int}(\mathbf{q}_i, \dot{\mathbf{q}}_i)}{\partial \mathbf{q}} d\mathbf{q} + \frac{\partial \mathbf{f}_{int}(\mathbf{q}_i, \dot{\mathbf{q}}_i)}{\partial \dot{\mathbf{q}}} d\dot{\mathbf{q}} \tag{2.4}$$

In the most generic case, we denote $\frac{\partial \mathbf{f}_{int}}{\partial \mathbf{q}} = \mathsf{K}$ the stiffness matrix and $\frac{\partial \mathbf{f}_{int}}{\partial \dot{\mathbf{q}}} = \mathsf{D}$ the damping matrix. In SOFA, viscous material are simulated using Rayleigh damping which define D as a linear combination of the mass and stiffness matrices $\mathsf{D} = \alpha\mathsf{M} + \beta\mathsf{K}$ with α the Rayleigh mass and β the Rayleigh stiffness.

This leads to the following system to solve at each time step:

$$\underbrace{(\mathsf{M}(\mathbf{q}_i) - h\mathsf{D}(\mathbf{q}_i, \dot{\mathbf{q}}_i) - h^2\mathsf{K}(\mathbf{q}_i, \dot{\mathbf{q}}_i))}_{\mathsf{A}} \underbrace{d\dot{\mathbf{q}}}_{\mathbf{x}} = \underbrace{h\mathbf{f}_{int}(\mathbf{q}_i, \dot{\mathbf{q}}_i) + h^2\mathsf{K}(\mathbf{q}_i, \dot{\mathbf{q}}_i)\dot{\mathbf{q}}_i + h\mathbf{f}_{ext}}_{b} \tag{2.5}$$

We obtain a sparse matrix system $\mathsf{A}\mathbf{x} = \mathbf{b}$ solved to compute the value of $d\dot{\mathbf{q}}$. Then the values of the new positions $\mathbf{q}_{i+1}$ and the new velocities $\dot{\mathbf{q}}_{i+1}$ are respectively updated using the Euler scheme as described in Eq. 2.3. As K, D and M are symmetric, positive definite and sparse by construction, the matrix A is then invertible and the system can be solved at each time step.

Similarly, in the quasi static case, we obtain the following system to solve at each time step:

$$\underbrace{(-\mathsf{K}(\mathbf{q}_i, \dot{\mathbf{q}}_i)}_{\mathsf{A}} \underbrace{d\mathbf{q}}_{\mathbf{x}} = \underbrace{\mathbf{f}_{int}(\mathbf{q}_i) + \mathbf{f}_{ext}}_{b} \tag{2.6}$$

Lagrangian Formulation for Actuation and Contact Constraints in Forward Simulation

Soft robots movements are generated by a change in the equilibrium of forces induced either by actuation or interaction with the environment. In the previous Section, we have considered that the external forces $\mathbf{f}_{ext}$ do not depend on the position of the nodes, which is not always the case for actuator and contact forces. We separate external forces, such as gravity forces or constant forces applied in one point, and forces due to constraints, i.e. actuator and contact forces. Therefore, $\mathbf{b}$ includes external forces and constraints are model using Lagrange multipliers, leading to the following system to solve:

$$\begin{cases} \mathsf{A}\mathbf{x} = \mathbf{b} + h\mathsf{H}_a^T\lambda_a + h\mathsf{H}_c^T\lambda_c & \text{in the dynamic case} \\ \mathsf{A}\mathbf{x} = \mathbf{b} + \mathsf{H}_a^T\lambda_a + \mathsf{H}_c^T\lambda_c & \text{in the quasi-static case} \end{cases} \tag{2.7}$$

λ_a and λ_c are respectively the constraint forces relative to actuation or contacts.

H_i^T for $i \in \{a, c\}$ can be seen as the first derivatives (i.e. the Jacobian matrices in the mathematical sens) of the constraints. They are matrices of size $n \times n_i$, n being the number of DoFs in

the FEM mesh and n_i being the number of DoFs for the considered constraint, which directly contains the direction of the considered forces on the set of involved nodes.

For each of these constraints, a characteristic distance δ_i , $i \in \{a, c\}$ is defined. This quantity has different meaning depending on the type of constraint considered. For instance, δ_a is the length of the cable for a cable actuator or the volume change for a pneumatic actuator. For a complete mathematical description of most soft robot models available in SOFA, the reader is referred to this work [Coe19]. In SOFA, colliding points and surfaces are detected at each simulation step. The responses are computed using Signorini's and Coulomb law as described in [Dur13b]. For contacts, δ_c represents signed distance between two colliding objects in the contact zone. Illustrations of these distances are provided in Figure 2.10. We will introduce further the mathematical expressions of these characteristic distances in this manuscript along with the inverse control of soft robots in Section 2.2.1.

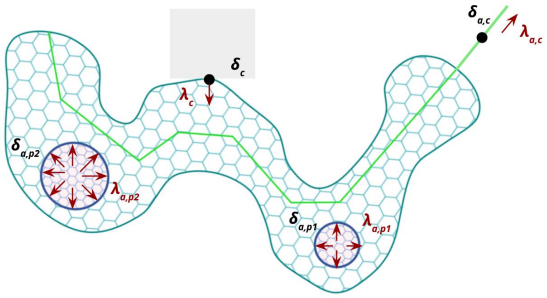

Figure 2.10: Illustration of the characteristic dimensions for a soft robot, actuated by a cable c (in green) and two pneumatic cavities $p1$ and $p2$ (in dark blue) while colliding with an object (grey square).

For solving continuum mechanics equations with a numerical method, the number of degrees of freedom must be sufficiently high for ensuring accurate results. This result in obtaining systems with many DoFs, which makes the system impossible to solve in the computation time required in robotics. However, whereas FEM can guarantee good prediction of the deformation of the soft robot, what is really useful in robotics is to parameterize the space of configurations the robot can take, i.e. the relationship between actuators, contacts and effectors. The motion of the robot can then be described in a sub-space composed of the actuators and external constraints such as contacts.

The constraints are then solved in two steps. First, the free configuration $\mathbf{x}^{free}$ in which there is no actuation or contacts is obtained by solving:

$$\mathsf{A}\mathbf{x}_{free} = \mathbf{b} \tag{2.8}$$

Next, these values are corrected by incorporating the effect of the constraints in the considered sub-spaces. The final solution, obtained from both Equation 2.8 and Equation 2.7, is then given by:

$$\begin{cases} \mathbf{x} = \mathbf{x}^{free} + h\mathsf{A}^{-1}\mathsf{H}_a^T\lambda_a + h\mathsf{A}^{-1}\mathsf{H}_c^T\lambda_c & \text{in the dynamic case} \\ \mathbf{x} = \mathbf{x}^{free} + \mathsf{A}^{-1}\mathsf{H}_a^T\lambda_a + \mathsf{A}^{-1}\mathsf{H}_c^T\lambda_c & \text{in the quasi-static case} \end{cases} \tag{2.9}$$

2.1.5 Model Order Reduction of Soft Robots

In soft robotics, model order reduction algorithms have been widely studied for reaching real-time performances. Having reliable and quick to compute reduced models is necessary to efficiently tackle data-driven co-optimization algorithms for both control and design. In this thesis work, we aim at being generic and we do not make any assumption regarding the geometry of the robot. Moreover, we are interested in models that can capture design variations. This Section explores current reduced order applications in soft robotics.

In classical continuum mechanics, reduced order modeling or model order reduction (MOR) techniques consist in reducing the computational complexity of a model while maintaining acceptable prediction errors. In the machine learning community, these methods are referred as state representation learning [Les+18]. The general objective is to find a model with few variables that describe the full order model by keeping only relevant information for the task at hand. These small dimensional model are mostly targeted at control applications, but, in our case, we are also interested in design applications.

Reduced Order Modeling of Continuum Mechanical Models with Geometrical Assumptions

For models with geometrical assumptions, any approaches targeting at approximating evolution of states of a soft robot is called a MOR approach. The objective is to build a reduced order kinematics of a soft robot. As already discussed in Section 2.1.2, energetic methods for solving continuum mechanical models with geometrical assumptions require finding a truncated functional basis of vectors, on which are defined a set of decoupled Lagrange differential equations in time or in space. Two main methods are available to find or further reduce this basis: modal approaches, where the system kinematics is approximated by a weighted sum of the dominant deformation modes, and fitting approaches, where the system kinematics is based on a basis function that is calibrated to the considered system [Sad+23]. In both cases, the modes shape or pre-designed basis functions are obtained with generic methods (spectral analysis or series expansion), or by fitting some empirically chosen function. For instance, Legendre polynomials are used for approximating a beam in [Liu+18b]. For a more in-depth reviews of reduced modeling techniques applied to continuum mechanical models with geometrical assumptions, the reader can refer to [Sad+23] [Qin+23] .

Reduced Order Modeling of Generic Continuum Mechanical Models

The computational complexity associated with high dimensional models is a major drawback to FEM simulation. A first step towards real-time simulation is to replace complex matrix computations with approximations. This is the case in SOFA, where the linear elasticity hypothesis is adopted, which enable local pre-computations (see Equation 2.9) for reaching real-time performances.

Popular methods, like the principal orthogonal decomposition, aim at compressing the state space dimension. This technique consists in reducing the size of the model by projecting the FEM equations into a reduced space. The principal modes of deformations are computed using a singular value decomposition on a pre-computed dataset of soft robot configurations. This dataset should be representative of the workspace of the robot. This technique enables to find a reduced basis Φ which, once integrated in the Equations 2.7 of motions, gives the following

equations:

$$\begin{cases} \Phi^T A \Phi d\ddot{\alpha} = \Phi^T \mathbf{b}(\Phi\alpha, \Phi\dot{\alpha}) + h(H_a\Phi)^T \lambda_a + h(H_c\Phi)^T \lambda_c & \text{in the dynamic case} \\ \Phi^T A \Phi d\alpha = \Phi^T \mathbf{b}(\Phi\alpha) + (H_a\Phi)^T \lambda_a + (H_c\Phi)^T \lambda_c & \text{in the quasi-static case} \end{cases} \tag{2.10}$$

where $\Phi\alpha = \mathbf{q}$ is the reduced expression for the vector of positions. This methods enable to drastically reduce the number of degrees of freedom of the model, and can account for contacts and even self-contacts [GCD21]. For instance, in [GD18], a basis of 60 modes is enough for controlling a complex robot (see Figure 2.11).

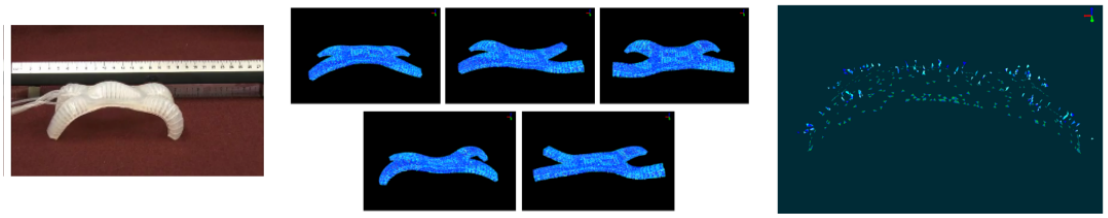

Figure 2.11: Example of MOR for a multigait soft robot (left) actuated by 5 pneumatic actuators from [GD18]. The 5 first basis vectors (middle) and the reduced integration domain (right) are displayed. A given state of the multigait is approximated by a weighted sum of the basis vectors.

Note that these kind of methods present the same drawbacks than learning-based models introduced in Section 2.1.2. They are dependant on the data used for computing the reduced basis, and thus are not able to generalize to configurations unmet in the dataset. It is challenging for most soft robots having strong interactions with the environment, where it is difficult to determine where the contacts will occur in advance. These methods are also design/mesh dependant, meaning that changing the geometry of the robot involves recomputing the entire pipeline.

2.2 Soft Robot Control

Controllers leverage soft robot models to generate control inputs that manipulate the soft robot effectively. Having a trustworthy simulation is necessary to account for the non-linearities and large deformations behaviors of soft robots, but this simulation must be at least real-time for being used in practical applications. Additionally, a robust controller must incorporate a feedback component to accommodate external disturbances as well as disparities between the model and the physical robot.

One question that may be asked is : what is meant by controlling a robot? When the task involves to follow a trajectory in space, optimisation methods for continuum-mechanics based inverse models have been developed [Coe19] [Mén+24]. It consists in finding the actuation strategy to match some target locations and velocities with the effectors of the robot. However, the most apparent answer to our question is to focus on the task the robot needs to accomplish. For instance, it could be grasping and manipulating objects or locomoting to a specific far-away location. With our tools, controlling a robot on these kinds of task then involves breaking down the task in a set of trajectories that can be solved with optimization. Optimization approaches are not sufficient in this case and need to be coupled with other algorithms such as path planning algorithms. For this part, we rely on Reinforcement Learning (RL) algorithms to explore the space of trajectories from simulation data. In this manuscript, we differentiate between low-level control, which involves executing predefined trajectories, and high-level sequential control schemes, which generate a series of trajectories to accomplish a specific task. The two main algorithms employed for each class of controllers are described below.

2.2.1 Low Level Control using Mechanical Modeling

This Section introduces the intricate coupling between simulation based on mechanical modeling and control, highlighting that control methods both benefits from accurate and fast simulations. Moreover, optimization methods and mathematical notations for inverse control of soft robots based on a condensation of the FEM model are presented. Low-level inverse control of soft robots using quadratic programming is widely used in this work.

Usefull concepts:

- Condensation of the FEM in the constraint space W_{ij} for $i, j \in \{a, c, e\}$
- Characteristic distance δ_i^{free} for $i \in \{a, c, e\}$
- Quadratic programming for control

Simulation and Model-Based Control

Optimization algorithms used in low level control strategies require mechanical quantities that are computed in the simulation.

When considering physical prototypes, having good open-loop algorithms is not always sufficient. Algorithms based on continuum mechanical models are sensitive to lot of numerical parameters such as mechanical parameters, mesh discretization and boundary conditions. Moreover, they are sensitive to modeling uncertainties, like the variations in geometry and

materials properties of the soft robot, or external perturbations that are not accounted for in the model. For guarantying that the system reach a desired trajectory, closed-loop controllers are set. They rely on sensors to provide feedback about the robot state (actuation, contacts, deformation). Closed-loop control can be divided into two levels of details [WC22]. In the first one, the feedback enable controlling the deformation state of actuators. For instance, measures of the pressure applied in the pneumatic cavities of a soft robot can be used to ensure the actuator delivers the wanted actuation [Bes+16]. The second-level builds a feedback part from measures in the configuration or task space. For instance, it could be the location of the end-effectors of the robot [Del+20; BS16].

Building working closed-loop algorithms for soft robot is an active area of research. One limitation of closed-loop strategies is that they are mostly model-based, meaning that algorithms not developed with generic mechanical model are applicable only to particular soft robot systems. Another issue is related to the small size of most soft robots. Making small sensors integrated with actuators is then a limiting factor for small soft robots, but also for soft robots with many actuators. In these cases, open-loop algorithms are preferred. Complete review of closed-loop algorithms used in soft robotics is available in [WC22].

Quadratic Programming

A method has been proposed for one-time step control of soft robots modeled with Lagrangian mechanics [Coe19]. This method extend the modeling and numerical scheme introduced in Section 2.1.4.

In addition to actuation and contact constraints, the effectors sub-space is considered. Effectors are points at which the robot's movement is controlled. Note that, as soft robots are usually under-actuated, only a sub-part of the total number of degrees of freedom can be controlled. Effectors are classically end-tip of the robots or where sensors are located. The general idea is to minimize some characteristic distance $\delta_e = \mathbf{q}_e - \mathbf{q}_{goal}$ between the 3D locations of effectors $\mathbf{q}_e$ and some goals $\mathbf{q}_{goal}$. As for the actuators and the contacts, the Jacobian matrix (in the mathematical sense) of the effectors H_e is defined to obtain the motion of the effectors given the motion of the degrees of freedom of the FEM model:

$$\Delta\delta_e = \mathsf{H}_e d\mathbf{q} \tag{2.11}$$

Practically, some nodes of the mesh are defined as effectors such that

$$\mathsf{H}_e = \begin{bmatrix} 0 & 0 & 0 & 0 & \mathsf{I} & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.12}$$

where I is the identity matrix located at the effector position indices. I is of the same dimensions as the number of degrees of freedoms attached to a node (3 for a position node, or 6 for a position and orientation node when using beam of shell elements for instance). The generalization to several effectors is straightforward.

For reducing the equations of equilibrium and motion on the loaded degrees of freedom, the compliance is projected in the space of constraint. The unloaded degrees of freedoms are then slaves of the loaded ones. This technique is called condensation of the FEM. As with the forward modelling described in Section 2.1.4, the solution of the inverse problem is computed in two steps. First, a free configuration is solved for $\mathbf{q}^{free}$ and the violation of the constraints in the free configuration of the robots are evaluated as δ_i^{free} for $i \in \{a, c, e\}$. When the equilibrium is disturbed, the distance of effectors to their goal position is updated by a linear approximation of the nodes displacement using FEM interpolation. For this purpose, characteristic distances are

built from the projection of the integrated admittance matrix A^{-1} from Equation 2.5 as follows:

$$\begin{aligned}\delta_a &= \mathsf{W}_{aa}\lambda_a + \mathsf{W}_{ac}\lambda_c + \delta_a^{free}\\ \delta_c &= \mathsf{W}_{ca}\lambda_a + \mathsf{W}_{cc}\lambda_c + \delta_c^{free}\\ \delta_e &= \mathsf{W}_{ea}\lambda_a + \mathsf{W}_{ec}\lambda_c + \delta_e^{free}\end{aligned} \tag{2.13}$$

when $\mathsf{W}_{ij} = \mathsf{H}_i\mathsf{A}^{-1}(\mathbf{q},\dot{\mathbf{q}})\mathsf{H}_j^T$ for $i,j \in \{a,c,e\}$ the projection of A in the constraints spaces. These matrices are homogeneous to a compliance and gather the mechanical coupling between the different constraints. The values of these matrices are not constant and depend on the configuration q of the robot. Moreover, their computation in real-time can be challenging as the size of A is proportional to the size of the FEM mesh.

Note that this projection on constraints is in fact also used for forward simulation. The difference is that, at this stage, user-chosen actuation is directly applied to the actuators. Here, the values of actuation and contacts are not known in advance, which leads to solve the following optimization problem:

$$\begin{aligned} arg_{\lambda_a,\lambda_c} min \quad & \|\delta_e\|^2 \\ s.t. & \delta_{min} \le \delta_a \le \delta_{max} \text{ (Actuators course constraint)} \\ & \lambda_{min} \le \lambda_a \le \lambda_{max} \text{ (Actuation effort constraint)} \\ & 0 \le \lambda_c \perp \delta_c \ge 0 \text{ (Contact constraint)} \end{aligned} \tag{2.14}$$

This optimization problem minimizes the distance δ_e while respecting the constraints on actuation and contacts. We face a quadratic problem (QP) when no contacts are considered, and a quadratic problem with complementary conditions (QPCC) when contacts are considered. In the last case, λ_c is part of the problem for enabling the robot to use contact to reach certain positions. Dedicated solvers are available to solve both types of optimization problems, and are further introduced in [Coe19].

2.2.2 High Level Sequential Control using Machine Learning

This Section introduces reinforcement learning methods to address sequential decision problems. While these concepts are not directly applied in this thesis, our goal is to leverage reinforcement learning for the co-optimization of both controller and design for soft robots. Challenges related to integrating the condensed FEM model into a reinforcement learning loop are further discussed in Chapter 5.

Usefull concepts:

- Soft Actor Critic
- Sample efficiency and sim-to-real transfer in reinforcement learning

High level sequential controllers such as RL algorithms may also benefit from simulation. They require large amount of data for training, which are not easy to obtain on a physical prototype. Simulation provides a reproducible environment where experiments can be precisely replicated, facilitating comparison and validation of different algorithms or approaches. Simulation environments are safer, as RL algorithms usually require large amounts of trial and error,

which can be risky and potentially damaging both for the prototype and its environment. Finally, simulation enables parallelization and computation time that can be faster than real-time. This last point needs to be mitigated. When an accurate simulator is used, the complexity of the simulation may lead to very high computation times, limiting the use of machine learning algorithms. Control policies learned in less accurate simulators, on the other hand, may struggle to transfer to physical prototypes.

Reinforcement Learning (RL) algorithms have been applied to optimize soft robot controllers for high level tasks. This is the case, for example, with Chainqueen [**Hu2018ChainQueenAR**] and DiffTaichi [**Difftaichi**], which have been used to simulate soft robots to learn control strategies. However, these simulators rely on simplified models and do not offer a wide variety of actuations, deformable structures or constitutive laws. SofaGym [**SofaGym**], a RL framework relying on FEM simulation, can be used to solve complex tasks with accurate robot models. However, learning the control strategy requires a lot of interaction between the robot and its environment. This process is time-consuming when the simulated model is slow.

Solving a sequential decision problem involves determining a sequence of decisions to accomplish a specific task. A common modeling approach for decision-making is the Markov Decision Process (MDP), defined by four key components:

- **State Space**: The robot's state space is represented by $S = \{\mathbf{s}_0, \mathbf{s}_1, ..., \mathbf{s}_{t-1}, \mathbf{s}_t, \mathbf{s}_{t+1}, ...\mathbf{s}_n\}$ where $\mathbf{s}_t$ denotes the successive states of the robot.
- **Action Space**: A represents the set of possible actions, with A_t being the set of actions available from state $\mathbf{s}_t$.
- **Transition Function**: The transition function P, such that $\mathbf{s}_t \approx P(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$, describes the probability that taking action $\mathbf{a}_{t-1}$ in state $\mathbf{s}_{t-1}$ will lead to state s_t.
- **Reward Function**: The reward function $R_t \approx R(\mathbf{s}_{t-1}, \mathbf{a}_{t-1})$ computes the expected reward received after transitioning from state $\mathbf{s}_{t-1}$ to $\mathbf{s}_t$ using action $\mathbf{a}_{t-1}$.

In robotics, the action space is defined by the various values applied to actuators, while the robot states are described by sensor data. Solving an MDP involves finding the optimal policy $\pi(\mathbf{a}t|\mathbf{s}t-1)$, which represents the probability of taking action a_t given the state $\mathbf{s}_{t-1}$. The policy, denoted as π, is to be optimized to maximize the expected cumulative reward J such that

$$J_\pi = \lim_{t \to +\infty} E_\pi[G_\pi(t)] = \lim_{t \to +\infty} E_\pi[\sum_{k=0}^{t} \gamma^t R(\mathbf{s}_k, \mathbf{a}_k)] \tag{2.15}$$

where $G_\pi(t)$ is the expected reward in step t applying the strategy π, and $\gamma \in [0,1]$ is the discount factor which determines the importance of future rewards.

In the context of robotics, we know the sets of states S, actions A and the J function, but not the transition and reward functions. Reinforcement learning rely on the interactions of the robot with its environment, whether physically or using simulation, to obtain information regarding both these functions through trial and errors. Much of the reinforcement learning literature rely on a value function V to solve this problem:

$$V_\pi(\mathbf{s}_k) = E_\pi[G_\pi(t)\mathbf{s}_0 = \mathbf{s_k}] \quad k \leq t \tag{2.16}$$

This function differs from the expected cumulative reward function J as it defines the expected outcome of starting from state $\mathbf{s}_k$ and following the policy π. Specifically, it quantifies

the expected reward from state $\mathbf{s}_k$. According to MDP theory, a control strategy is optimal if it maximizes the value function V for any initial state. However, in robotics, it is often impractical to directly control the state the robot will reach after one action, as evaluating the transition function P is not always possible. Instead, we use the Q-value function $Q_\pi(\mathbf{s}_k, \mathbf{a}_k)$, which is defined as the expected return starting from state $\mathbf{s}_k$, taking action $\mathbf{a}_k$, and subsequently following policy π:

$$Q_\pi(\mathbf{s}_k, \mathbf{a}_k) = E_\pi[G_\pi(t)\mathbf{s}_0 = \mathbf{s_k}, \mathbf{a}_0 = \mathbf{a_k}] \quad k \leq t \tag{2.17}$$

Finally, maximizing the value function V for any initial state is equivalent to consistently choosing the action in the current state that leads to the state with the highest value. In other words, the optimal policy is obtained as follows:

$$\pi*(\mathbf{s}_k, \mathbf{a}_k) = argmax_{Q_\pi}(Q(\mathbf{s}_k, \mathbf{a}_k)) \tag{2.18}$$

where $\pi*$ is the optimal strategy.

The Q-value function exact value is typically unknown, prompting the development of various approximation methods [KBP13]. One notable approach is the Soft Actor-Critic (SAC) algorithm [Haa+18], an off-policy reinforcement learning method designed to balance exploration and exploitation by maximizing both the expected reward and the policy entropy. SAC employs a stochastic policy to encourage exploration through random action selection. It utilizes two Q-value functions to reduce positive bias during policy improvement and a separate policy network, the actor, to update the policy towards actions that optimize the soft Q-values. The SAC algorithm has proven effective in learning control policies for soft robots simulated in SOFA, and an open-source plugin called SofaGym is available for this purpose [Sch+23].

Reinforcement learning faces several challenges, particularly with robots, including sample efficiency. While actions are typically sampled discontinuously, most robots operate in continuous state and action spaces. For example, adjusting a cable actuator on a soft robot in 1mm increments requires fine control but results in a vast sampling space. On physical systems, obtaining samples is costly due to the time needed for complex task execution and robot maintenance. Using a simulation environment for training control policies can mitigate these issues, but it introduces challenges related to the simulation-to-reality transfer. The simulation must be robust against environmental and model inaccuracies. Therefore, reinforcement learning benefits significantly from simulators that are both accurate and computationally efficient. This last point especially motivates the development of a surrogate learned from the condensed FEM model in this thesis work.

2.3 Soft Robot Design

Soft robot designs benefit from a variety of new materials, embedded sensors for smooth integration in the robot body, and actuators for making the structure able to deform itself. The rise of soft robotics is strongly linked with the widespread adoption of additive manufacturing (AM) due to its capacity for rapid prototyping and customization.

Soft-robots are still primarily designed by hand through replacing hard parts by soft ones in rigid robots or reproducing what can be found in nature. Using soft materials in robots indeed poses unprecedented challenges by increasing the complexity of robotics systems because of the intricate coupling between geometry, material and actuation, rendering intuitive design extremely challenging. We discussed more in-depth the limitations of these approaches in the introductory Section 1.4, and the need of specific tools and simulation for efficiently exploring the design space. A handful of attempt to automate the design of soft robots has been carried out, but there is no general framework widely adopted by the community yet.

This Section provides an overview of the vast design possibilities and the accompanying fabrication limitations. It then delves into the primary classes of frameworks used for optimizing the design of soft robots.

2.3.1 Design Space and Fabrication

This Section highlights the complexity of designing soft robots regarding the large available design space and the lack of standardized components. The computational design of soft robots implies refining these possibilities to build a space of possible designs. This involves considering the materials, electronic components, and manufacturing capabilities we have access to. Across this thesis work, several types of robot morphologies, materials (silicone, TPU), metamaterials (stochastically generated micro-structures), actuators (pneumatic chambers, cables, servo-motors), sensors (motion tracking, pneumatic chambers, capacitive sensors) have been considered in several soft robot prototypes.

Available Materials

In nature, soft materials such as rubber, sponge and leather are prevalent [Li+19]. The emergence of soft robotics is greatly linked to the advancements in fabricating synthetic materials that share analogous characteristics.

Silicone-based elastomers are the primary choice for designing soft robots. Typically, they are casted in rigid molds, which are 3D printed beforehand [Sac+21]. However, complexities arise when dealing with cavities, requiring multiple separate casts and subsequent reassembly. This process is prone to minor asymmetries and challenges in ensuring a tight seal for the cavity. Moreover, it tends to be slow and cumbersome to manage.

Advancements in additive manufacturing have paved the way for direct 3D printing of flexible materials. For a comprehensive overview of the primary commercial materials and associated printing techniques, refer to [YSY20]. In the context of this thesis, a notable example of a widely used elastomer for 3D printing filament is thermoplastic polyurethane (TPU). Despite these strides, achieving the same level of flexibility with 3D printed silicone as with traditional silicone casting and molding methods remains challenging. However, recent developments offer promising alternatives. For instance, [Spa+21] demonstrated the 3D printing of silicone

pneumatic actuators (see Figure 2.12.A), while commercial printers capable of directly 3D printing silicone objects are now available [23].

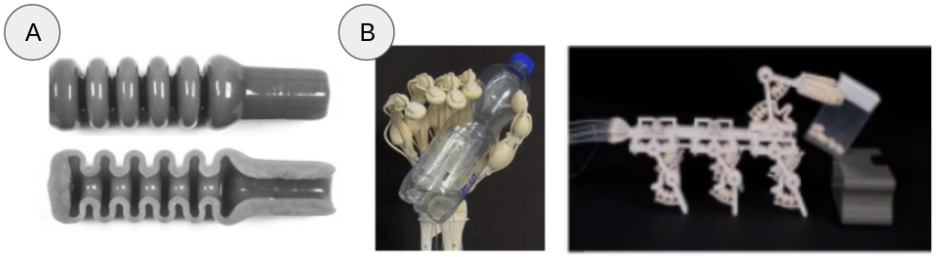

Figure 2.12: Example of a few soft robot 3D printed design. A) 3D printed silicone pneumatic actuator with the method from [Spa+21]. B) 3D printer multi-materials tendon-driven gripper and locomoting gripper obtained with the method from [Buc+23].

Composite materials, including shape memory alloys, shape memory polymers, and smart composites, as well as multi-material 3D printing, both enable to obtain complex engineered materials made from several constituent materials, offering a wide array of interesting properties. However, utilizing these kind of techniques remains challenging and costly with current commercially available materials and 3D printers, thus lying beyond the scope of this thesis. We however highlight one recent notable advancement, which is the vision-based multi-material 3D printing method pioneered in [Buc+23], facilitating the creation of structures featuring intricate arrangements of elastic and rigid materials, along with integrated sensing and actuation channels. The authors showcase their approach by directly 3D printing soft-rigid hybrid robots like a tendon-driven five-fingered gripper and a locomoting gripper (see Figure 2.12.B). One advantage of some advanced multi-material 3D printers is the ability to obtain continuous gradients in material properties during printing. This suggests that we will be able to automatically manufacture much more complex soft robots by overcoming numerous manufacturing constraints in a near future.

When it comes to FEM simulation, modeling plain structures made of soft elastomers is straightforward. However, dealing with non-periodic composite structures presents greater complexity. This is because accurately defining the material boundaries within the FEM mesh requires a thorough understanding of the distinct material interfaces. In this thesis, we focus exclusively on mono-material soft robots.

Metamaterials

In an ideal scenario, soft bodies should exhibit mechanical properties heterogeneity, i.e. comprising various materials with differing mechanical properties across different regions. Figure 2.13 provides an overview of the mechanical properties offered by synthetic materials and compares them with those of organic materials. As discussed earlier, multi-material 3D printing offers a solution for achieving such heterogeneous bodies.

An alternative approach involves working with flexible mechanical metamaterials [Ber+17]. These are engineered materials featuring structures ranging from microscopic to macroscopic scales, deliberately designed to exhibit unconventional mechanical properties. Of particular interest is the ability to manipulate material patterns to directly program mechanical behavior within soft bodies. Three primary categories of flexible mechanical metamaterials have been

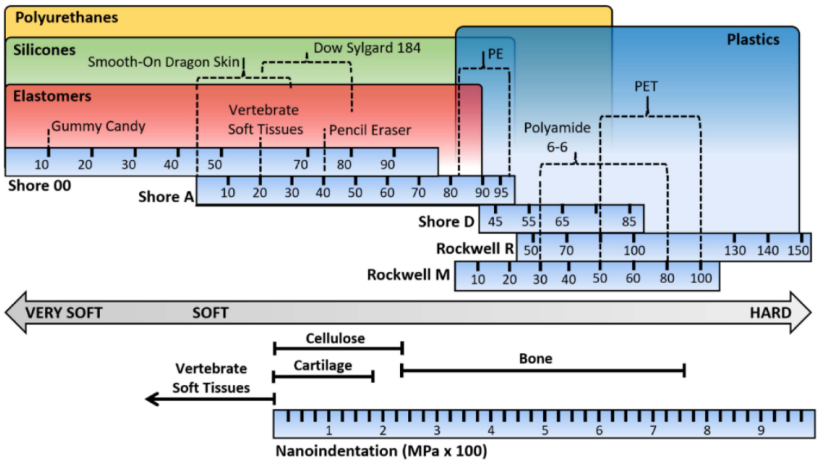

Figure 2.13: Comparison of hardness and softness of synthetic materials used in soft robotics and some organic materials, reproduced from [Zhu+22].

utilized to encode intriguing behaviors in soft robots [RBS19].

Linear beam-based structures are frequently employed for programming the patterns of flexible metamaterials due to their ease of manufacture using 3D printing. Among these structures, auxetic materials stand out, featuring a negative Poisson's ratio achieved through well-known patterns. These materials thicken when stretched and become thinner when compressed. For instance, auxetic structures have been combined with non-auxetic clutches to simplify locomotion in soft worm robots, reducing the need for multiple independent actuators to just one [Mar+16] (see Figure 2.14.A), and have also been applied in the design of gripper fingers [TCR21]. Elastic beams not only bend but also buckle under axial compression, enabling the triggering of homogeneous and reversible configurations in structures composed of regular arrays of such beams. These configurations can be activated by applying negative pressure, as demonstrated in the development of a locomoting soft robot composed of several buckling material units [Gro+21] (see Figure 2.14.B).

Furthermore, recent work introduces the use of anisotropic foams, which involves flexible materials with procedurally and **stochastically generated micro-structures**. Unlike other aforementioned methods, this approach allows for a wide and smooth continuous range of mechanical properties, without relying on an abrupt mechanical behaviors. These foams have been demonstrated to enable the control of a new degree of freedom in a parallel robot [VGD21] (see Figure 2.14.C). Regarding FEM simulation, when metamaterials patterns are at the micro-scale, employing a mesh with sufficient precision to simulate their complexity becomes impractical. In such scenarios, metamaterials are approximated using homogenization theory. This involves employing a coarser macroscopic discretization, which provides an equivalent behavior at the macroscopic scale through a local averaging of the microscopic behavior, replacing the need for micro-scale discretization. Such approaches have been successfully demonstrated within the SOFA simulator.

Reinforced systems, also known as lattice structures in the literature, involve the assembly of 3D building blocks comprised of interconnected beams. These unit blocks often exhibit specific anisotropic behaviors and are assembled according to user requirements. Various techniques can

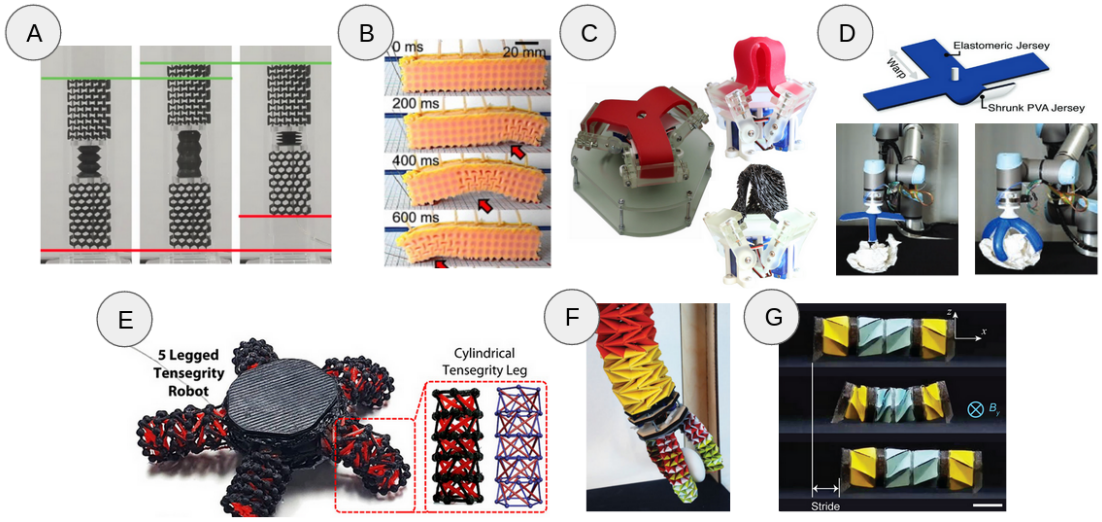

Figure 2.14: Highlight of a few soft robotics systems using metamaterials. A) Locomoting soft worm robot based on complementary auxetic components acting as a passive clutches 2.14. B) Metarpillar robot leveraging actuated buckling material units for locomotion [Gro+21]. C) Using stochastic foam enable to obtain a new degree of freedom on a parallel soft robot [VGD21]. D) Soft pneumatic gripper making use of two different knitted material. When actuated, the softer elastomeric yarn expand and all legs bend simultaneously [San+23]. E) Multi-directional locomoting soft robot with legs composed of tensegrity patterns [Lee+20]. F) Origami robotic manipulator from [TSR20]. G) Magnetically actuated origami crawler robot [Ze+22].

be employed to fabricate these blocks, including knitting or weaving. For example, both knitting and weaving have been utilized to construct artificial muscles using specialized filaments as actuators [Maz+17]. However, such approaches often necessitate additional assembly post-processing steps alongside casting and/or printing techniques. Addressing this challenge, [San+23] present a solution in the form of a 3D knitting machine designed for pneumatic soft robots (see Figure 2.14.D). Another noteworthy work introduces a strategy for additive manufacturing of tensegrity patterns, which combine stiff beams and flexible tendons. This method was demonstrated on a tensegrity robot capable of omnidirectional walking [Lee+20] (see Figure 2.14.E).

Origami and Kirigami, the Japanese arts of folding and cutting material sheets, represent the most active field in the soft robotics community for utilizing metamaterials. Their primary feature lies in their ability to create programmable, morphable shapes for soft robots. However, while designs achieved through these methods can exhibit high deformability, they often lack load-bearing capacity. This limitation can be addressed through specific designs involving arrangements of different materials [FAS18], leveraging multi-material 3D printing. Numerous exemplary applications abound, including a three-fingered origami manipulator [JL18], a soft robotics arm segment integrating kirigami conductive sensors [TSR20] (see Figure 2.14.F), and a small magnetically actuated origami crawler robot [Ze+22] (see Figure 2.14.G). Thanks to the maturity of the field, several specialized tools for simulation or design of origami- and kirigami-based mechanisms are readily available [RT18].

Actuation

Actuators enable soft robots to deform and move. Transitioning from rigid to soft robotics necessitates the development of novel actuation systems that are lightweight and compact to accommodate the characteristics of soft bodies. Various classes of actuators have been categorized based on the stimuli that trigger the actuation response [El-+20]. Each class possesses its own set of advantages and disadvantages, extensively discussed in [BRR18].

Many of these actuation systems rely on specialized active materials. Examples include magnetically responsive actuators [Kim+19], thermally responsive actuators utilizing shape memory alloys [Wan+14], and even photo-responsive actuators [Liu+17] or explosive-based actuators relying on internal combustion [Bar+15b]. Such actuation systems necessitate specific and unconventional methods for generating stimuli to initiate movement and rely on materials not easily accessible and challenging to manufacture.

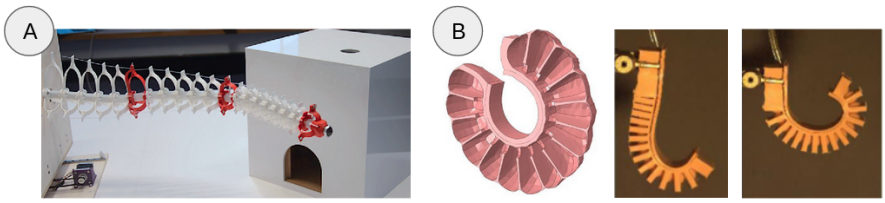

Figure 2.15: Highlight of a few actuation technologies used with soft robotics systems. A) Soft manipulator actuated by cables going through its compliant splines [Mor+20a]. B) Schematic and prototype of the PneuNet soft robot actuated by inflation of a pneumatic network [Mos+14].

The predominant actuation methods observed in existing literature for soft systems involve direct mechanical stimuli, achieved either through servomotors or cables attached to soft parts [Mor+20a] (see Figure 2.15.A), or by pressurizing cavities to induce deformation [Mos+14](see Figure 2.15.B). These actuation approaches are both used in the present thesis work. Utilizing cables involves managing friction effects between the cable and the structure, but it facilitates significant deformations. Conversely, pneumatic actuation allows for more distributed deformations but exerts greater stress on the structure compared to cable actuation. Regarding simulation, numerical models for such actuators have been extensively explored, with their accuracy validated on physical prototypes [Coe19]. It is worth noting that simulating vacuum actuation, where negative pressure is applied to a pneumatic cavity, poses computational challenges due to the occurrence of collisions across large surfaces.

Sensing

As outlined in Section 2.2 regarding control, sensors are necessary for gathering real-world data for model calibration or establishing feedback loops to address model uncertainties.

These sensors may be external, as seen in motion capture systems. In such systems, reflective markers are placed on the soft body and tracked in real-time by infrared cameras, as it is the case with the continuous soft robot by [Del+20] (see Figure 2.16.A). However, this approach is typically suited for specific constrained environments due to its reliance on an external setup and susceptibility to perturbations like occlusion.

In scenarios where external conditions can not be guaranteed, sensors must be embedded within soft structures. However, conventional sensors, primarily designed for rigid structures, pose challenges when integrating them into soft bodies. They must exhibit flexibility to preserve

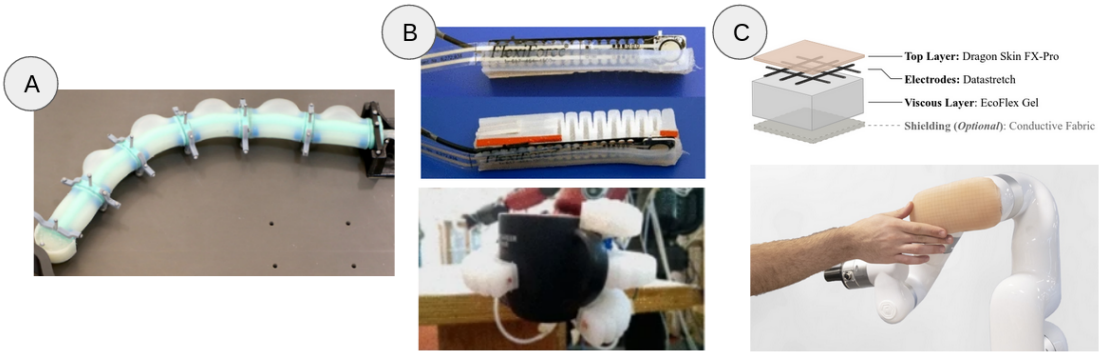

Figure 2.16: Highlight of a few sensor technologies used with soft robotics systems. A) Continuous soft robot with reflective markers for trajectory tracking with a motion capture system. [Del+20]. B) View of a soft finger with internal air channels, sealed Sections and inserted resistive sensors (up), and a soft gripper grasping a mug with force contact sensing (bottom) [Hom+19]. C) Artificial skin made of an electrode matrix embedded within different silicone elastomers layers for touch sensing (up), that can be mounted on a robotics arm (bottom) [Tey+21].

the soft robot's behavior. Additionally, given that soft robots theoretically possess an infinite number of degrees of freedom, sensors with enhanced range, accuracy, and resolution are needed.

Embedded sensors for soft robot is thus an active area of research. Here, we introduce several well-know classes of embedded sensors. Resistive sensors exploit the correlation between changes in the resistance of elastic conductive materials and their deformation state to estimate the robot's actuation state or interactions with the environment [Hom+19] (see Figure 2.16.B) [Tia+23]. Capacitive sensors discern the interactions of soft robots with their surroundings by detecting changes in the electric fields they generate [Tey+21] (see Figure 2.16.C) [Nav+20; Als+23]. Lastly, magnetic sensors ascertain the deformation state of soft robots by measuring changes in magnetic flux while a permanent magnet embedded within the soft body moves [Kha+12; Baa+22]. For instance, such sensors have even been employed to locate the tips of catheters in in-vivo applications [Kha+12].

Morphology and Geometry

The morphology and geometry of both the soft robot body and its individual components are crucial in the design process. For example, the exploration of the actuator design space extends beyond selecting suitable actuators and materials. In the context of pneumatic actuators, it also includes decisions about the placement and configuration of internal cavities.

Identifying the optimal morphology for a soft robot is particularly challenging because of the complex deformations possible with the materials or metamaterials in use. This complexity makes it difficult to rely solely on intuition when making design decisions.

2.3.2 Automated Design in Soft Robotics

Faced with the complexity of the encountered design spaces, and with the emergence of increasingly reliable numerical simulators, a growing interest in computational design optimization of soft robots has emerged in the soft robotics community.
In this Section, the main optimization frameworks are described and classified depending on their level of automation. It is shown that reducing user-induced bias in these frameworks, through choice of design parameters and fitness functions for instance, is usually to the detriment of the quality of the simulation and therefore to the transfer of design results into reality.
Generating novel designs while following embodied intelligence principles requires a general framework for evolving soft robot designs and controllers altogether. We show that state-of-the-art works in design and control co-optimization, introduced in this Section, struggle to integrate the two aspects.

A Classification of Soft Robot Computational Design Methods

Robotics design process starts with the articulation of high-level task objectives. For instance, it could be to quickly and stably navigate a terrain for a locomoting robot. Task-specific requirements are iteratively translated into a morphology, a mechanical structure, and finally refined in a comprehensive design encompassing detailed geometry, selection of materials and/or metamaterials, and the integration of actuators, sensors, and control systems. This process is summarized in Figure 2.17.

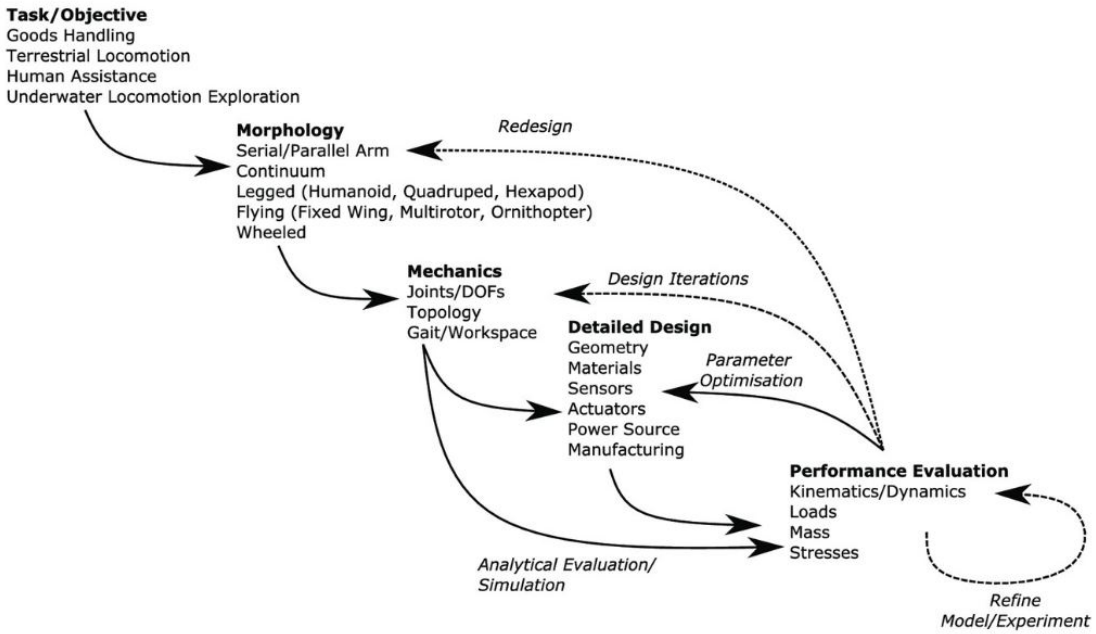

Figure 2.17: Iterative workflow for robotics design, reproduced from [PH22].

Given the infinite degrees of freedom in soft robotics and the lack of standardized compo-

nents, transforming task-specific requirements into an effective morphology still relies heavily on intuition and bio-inspired principles [Coy+18]. Due to the absence of a standardized process, researchers have increasingly turned to computational methods to accelerate the design exploration phase.

In this Section, we endeavor to present a classification of computational design frameworks tailored specifically for soft robots. We commence by acknowledging that these frameworks typically rely on four fundamental components, as described in the introductory Section 1.4.2: a simulation environment and objective function(s) for assessing design performance, a description of the design space, and an optimization method for efficiently exploring the design space.

In line with the suggestion from [PH22], we opt to prioritize the level of automation provided by the method as a classification parameter, characterizing them based on the required level of user involvement. This classification is summarized in Figure 2.18.

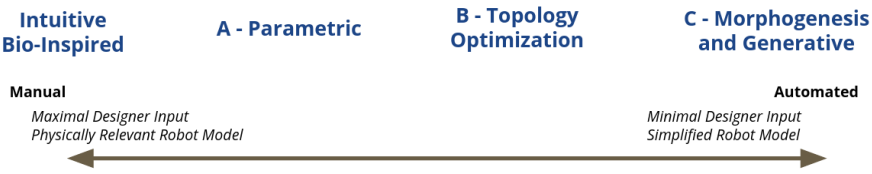

Figure 2.18: A classification of methods for exploring soft robot design based on the level of user's involvement they require.

A - Parametric Design

Beginning with a user-defined design representation, these methods focus on optimizing high-level parameters. They leverage the user's intuition to construct a well-parameterized design, enabling the creation of compact yet expressive design spaces which encompasses fabrication constraints. This approach enables the use of accurate and computationally intensive simulations, as shown by numerous examples in the literature spanning from numerical optimization to physical prototyping. However, these parameterizations typically revolve around variations of a baseline design, inherently influenced by human biases. Consequently, they are incremental, with users often refining variables of interest throughout the design variables exploration process. Parametric design methods can be broadly categorized into two groups: model-free and model-based methods.

Model-free Methods

These methods employ simulation as a black-box to evaluate design performances and rely on algorithms such as sensitivity analysis or stochastic algorithms to navigate the design space. Such approaches are particularly well-suited for computational design of soft robots, where complex mathematical fitness functions abound due to non-linearities in materials, contacts, and actuator/effectors couplings. Furthermore, design variables often interact in intricate ways, and disparate designs can yield similar performance outcomes. These methods facilitate comprehensive exploration of the design space and the discovery of multiple viable solutions. However, they suffer from low sample efficiency, lack proof of convergence to an optimal design and only yield solutions that are close to optimal.

Model-free methods have found widespread application in diverse shape optimization problems. For instance, investigations into soft pneumatic actuator geometries have employed various techniques, including sensitivity analysis [Hu+18a], multi-objective evolutionary algorithms

[Pag+20], genetic algorithms [RPR17], and sequential programming with finite difference approximations [Däm+19] (see Figure 2.19.A). In [MMD19], evolutionary algorithms were utilized to optimize the leg shapes of locomoting soft robots (see Figure 2.19.B). In this work, leg shapes were represented using splines, with their control points serving as design variables. The aforementioned work rely on FEM simulations for evaluating design performances. However, when the design space involves numerous variables, conducting many FEM simulations can be computationally intensive. Consequently, the expense of accurate simulations may hinder the incorporation of detailed environmental modeling or exploration of large design spaces.

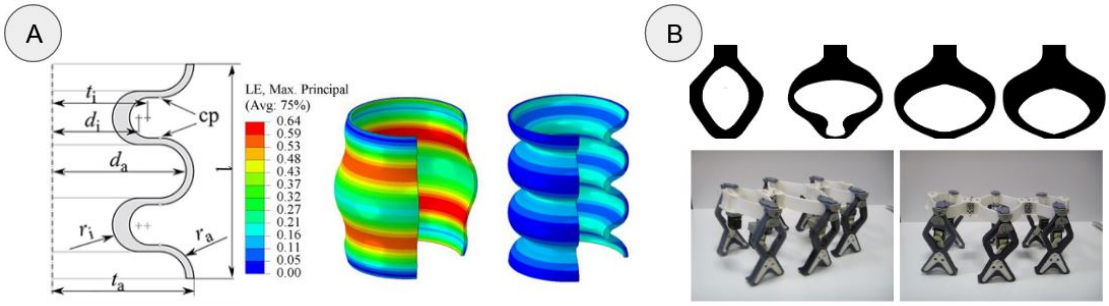

Figure 2.19: Example of frameworks for black box design optimization of parametric soft robot designs. A) Design parameterization of a linear bellow-type actuator optimized to minimize the induced maximal principal strain [Däm+19]. B) Example of parametric leg shapes and optimized leg robot design from [MMD19].

Model-Based Methods

These methods rely on specific strategies for leveraging information extracted from soft robot models. They need rewriting of specific objectives function and analytical gradients from soft robot models relative to considered continuous design variables.

Complete frameworks derived from continuum mechanics modeling have been developed for automatically designing actuated objects and soft actuators. Given an input shape, authors of [Sko+13] develop a framework for optimizing the location of cable actuators along with the input shape's material distribution for being able to reach a set of target deformation poses (see Figure 2.20.A). Mostly 2D cases are considered and there is little information on how the framework would scale to the 3D world. A similar objective of matching target deformation shapes is addressed in [Ma+17] [Din+19], where pneumatic chamber locations and rigid frame materials are optimized to constrain pneumatic chambers expansion (see Figure 2.20.B). Pneumatic chambers locations are found by iterative clustering, and rigid frames materials are optimized through gradient-based optimization.

Specific analytical models have been developed to optimize the position and geometrical parameters of elliptic pneumatic actuators for rigidification performance of a soft gripper [GK20], or the fibers orientation of a fiber-reinforced pneumatic actuator for performing different deformation states (extension, contraction, bending, twisting) [CWB17] (see Figure 2.20.C).

In all cases, authors show the results transfer qualitatively to physical prototypes.

The efficacy of these methods relies on the user's proficiency in crafting mathematical formulations that effectively reduce the feature space while maintaining high accuracy. These approaches are tailored to particular scenarios with specific constraints, posing challenges for extension to different design variables and scenarios, particularly for non-specialists in

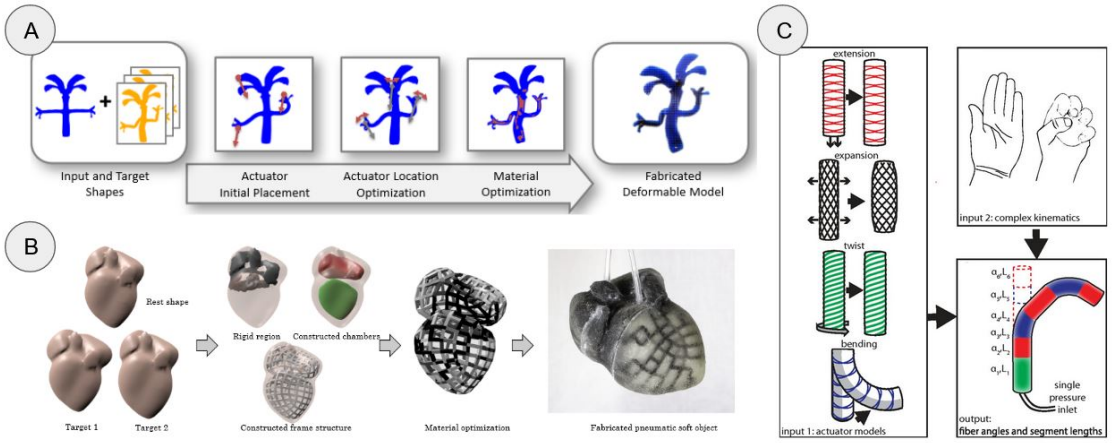

Figure 2.20: Example of frameworks for gradient-based design optimization of parametric soft robot designs. A) Workflow for optimizing cable location and material distribution of an actuated object for reaching target deformations, from [Sko+13]. B) Pipeline for determining pneumatic chambers location and shape, and optimizing frame structures for a pneumatic object to reach target deformations [Ma+17]. C) Fiber-reinforced pneumatic actuator design parameters are optimized for replicating a prescribed motion in [CWB17].

mechanical modeling and mathematical programming. Furthermore, as these methods do not model interactions with the environment, they are confined to soft components or soft robots without direct interaction with their surroundings. Consequently, tasks such as locomotion, grasping, or manipulation remain beyond their scope.

B - Topology Optimization

Topology optimization is a widely used method in the industry for structural optimization. Unlike parametric design approaches applied to optimizing shapes of soft robots, it enables for the generation of more free forms, reducing dependence on initial user intuition. Relying on the FEM for design performance evaluation, it is a numerical technique that redistributes material within specified design bounds. The problem is described by a set of constraints, loads and boundary conditions. The objective function is formulated from the mechanical equations describing the equilibrium of the optimized structure and mathematical programming is applied to explore the design space.

Different formulations and dedicated algorithms exist for topology optimization [SM13], accommodating discrete, or continuous variables like material densities or geometric boundaries. Continuous variable methods are widespread adopted as they facilitate efficient exploration of the design space through gradient-based algorithms.

Density-based methods iteratively redistribute material within a design space to optimize structural performance. This design space is typically represented as a grid of regular elements (such as voxels or hexahedrons). Each element is assigned a continuous density of material, ranging from void to solid, with penalties applied to encourage convergence toward a final refined design. This approach has found applications in designing grippers actuated by cables [Che+18a] [Wan+20] or motors [Liu+18a] (see Figure 2.21.A), as well as legged robots actuated by servomotors [Sun+23] or pneumatic actuators [CPN20] (see Figure 2.21.B).

On the other hand, level-set methods iteratively enhance the design's shape using a level-set

function delineating the interface between material and void regions. Despite being less prevalent than density-based methods, level-set approaches offer the advantage of precisely defining boundaries. This is particularly crucial in designs optimization cases featuring pneumatic chambers, as it prevents the appearance of disconnected elements [Pin+23]. Level-set methods have been employed in designing skeletons to achieve desired deformations with soft pneumatic actuators [Che+21] (see Figure 2.21.C) or ferromagnetic soft robots [Tia+20].

All the cited works directly prototype their designs using additive manufacturing and validate them experimentally. As additive manufacturing becomes more widespread, it opens up new possibilities. Recent advancements extend conventional topology optimization algorithms to generate structures with multiple materials and varying properties [Pin+23].

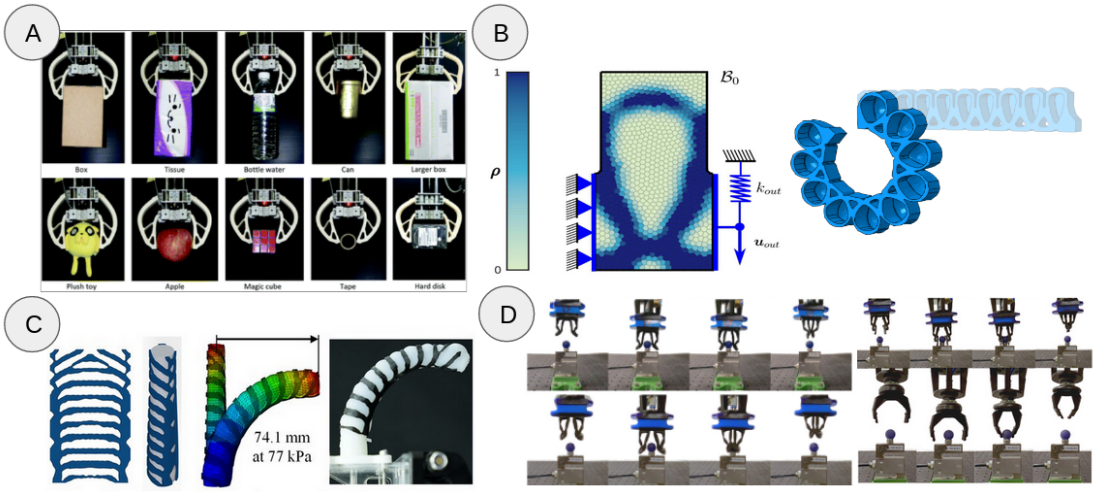

Figure 2.21: Soft robot components generated using topology optimization. A) Universal soft gripper design optimized with density-based topology optimization [Liu+18a]. B) Soft actuator design optimized with density-based topology optimization [CPN20]. C) Soft actuator skeleton design optimized with level-set topology optimization [Che+21]. D) Soft grippers generated with evolutionary algorithms coupled with topology optimization [Pin+24].

While there is significant enthusiasm for optimizing compliant mechanisms using topology optimization, conventional algorithms [Sig97] encounter several challenges in effectively addressing the requirements of soft robot design.

Primarily, these algorithms must be extended to accommodate the unique characteristics of soft robots, such as nonlinear materials and contacts. Recent efforts address this need: in [Che+17] and [CPN20], level-set and density-based methods are respectively adapted to handle hyper-elastic structures. Moreover, in [FSP23], density-based topology optimization is expanded to manage self-contacting structures. However, many examples of topology optimization in soft robot design rely on gradient-based algorithms, assuming linear behavior for soft materials and no discontinuous contact scenarios. Yet, gradient-based solvers can easily get trapped in local optima for non smooth problems. To efficiently explore soft robot design spaces, one solution involves coupling gradient-based topology optimization with stochastic algorithms to vary initial material distributions. This approach is exemplified in [Pin+24] where it's applied to generate soft grippers (see Figure 2.21.D).

Secondly, conventional topology optimization struggles to effectively capture soft robot

actuation methods due to its requirement for fixed constraints and loading scenarios. However, soft robot actuation typically involves internal mechanisms with spreaded constraints, such as the internal routing of cables or pneumatic chambers within the robot's body. This necessitates the development of specific formulations for modeling actuators alongside with material parameters in the framework of topology optimization, as further discussed in [PH22].

Lastly, conventional topology optimization algorithms typically focus on optimizing a configuration to static equilibrium under given constraints. Yet, selecting relevant configurations for specific tasks is challenging and often relies on user expertise to specify mechanical specifications. This complexity is particularly evident in tasks like grasping and manipulation, where various interaction scenarios between the robot and manipulated objects are possible. To address this issue, authors of [Che+18a] employ stochastic optimization algorithms to quantify the uncertainty associated with each potential contact scenario, determined experimentally beforehand. Others approaches consider expanding existing topology optimization frameworks to incorporate control strategies, as seen in [CBS23] and [Kob+23]. To facilitate obtaining gradients of the FEM with respect to actuation variables, these studies utilize Material Point Method (MPM) simulation instead of FEM simulation to evaluate optimized robot performance. These approaches are further explored alongside morphogenesis methods below.

C - Morphogenesis and Generative Methods

Morphogenesis frameworks, often referred to as evolutionary or generative design, employ discretized design spaces such as voxels grids. Material properties can evolve independently for each element within this space, while the actuation strategy is co-evolved, typically represented as voxels whose volumes cyclically expand and contract.

Many studies utilize simulation to evaluate design performance, employing gradient-free algorithms such as genetic algorithms to explore the design space. This approach originated from Karl Sims' pioneering work [Sim94], which introduced a method for generating numerical creatures capable of performing simple tasks like jumping or running. This methodology has since been extended to soft structures [Che+14] [Che+18b], encoding the morphology and control policies of soft robots within compositional pattern-producing networks (CPPN). Genetic algorithms, such as the neuro-evolution of augmenting topologies (NEAT) method, are then employed to evolve the neural architecture based on performance metrics evaluated in simulation.

Examples of soft robots generated using these frameworks are depicted in Figure 2.22.

The generation of comprehensive morphologies for soft robots remains an active research field, with numerous studies [PH22] exploring diverse fitness functions, morphology encodings, and heuristic algorithms. These approaches excel in producing both robot behaviors and morphologies that outperform traditional design optimization methods within their simulated environments.

However, heuristic algorithms commonly employed in such approaches often face challenges related to low sample efficiency, imposing significant constraints that are further discussed below.

First, researchers are often stuck with limited design spaces, typically confined to small grids comprising around $20 \times 20 \times 20$ voxels. However, recent advancements, such as those presented in [CBS23], have introduced fully differentiable pipelines capable of addressing expansive design spaces. Their study showcases the optimization of soft robots at a resolution of 100^3 voxels with over 2 million parameters. In contrast to previous methodologies reliant on discrete design spaces and stochastic solvers, their approach challenges the notion of soft

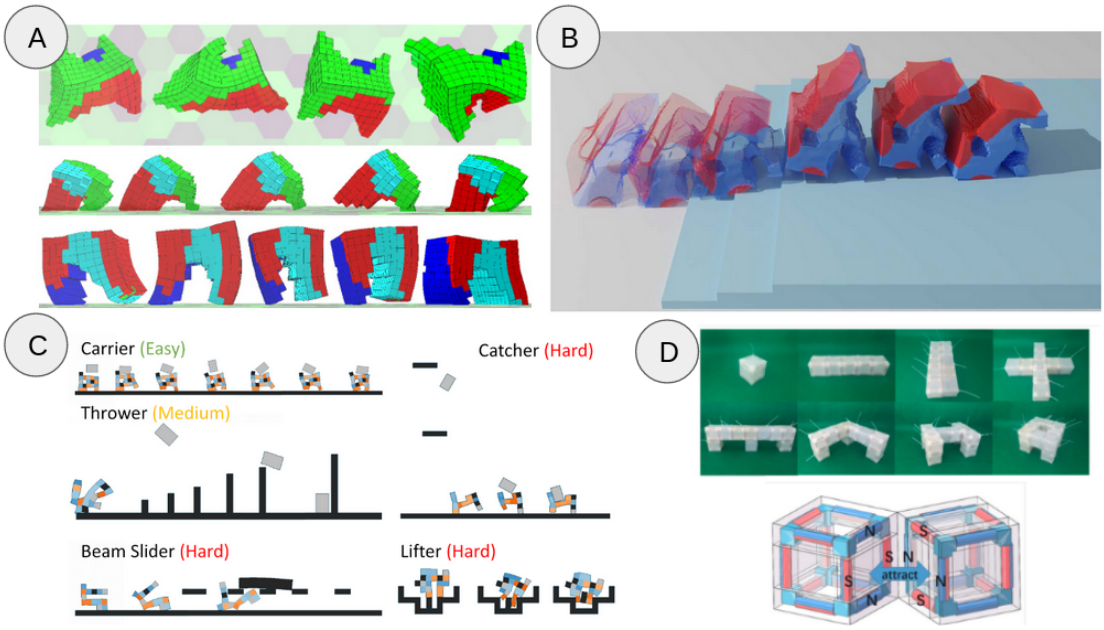

Figure 2.22: Highlights of several works using morphogenesis methods. A) Soft robots traversing a flat surface, created through CPPN-NEAT [Che+14]. B) Soft robots ascending stairs, generated via a fully differentiable pipeline within an extensive design space [CBS23]. C) Several 2D designs and non periodic control strategies for object manipulation tasks, obtained using CPPN-NEAT jointly with RL [Bha+22]. D) Magnetic modules for building voxelized soft robots [Sui+20].

robot design spaces as inherently discrete, advocating instead for their continuity. By leveraging differentiable MPM simulations, they successfully optimized a robot's geometry for climbing stairs (see Figure 2.22.B), offering promising avenues for scaling morphogenesis frameworks to larger resolution design spaces. Another significant contribution in this direction is the framework introduced in [Yuh+23]. This framework employs a density-based topology optimization algorithm coupled with MPM simulation instead of traditional FEM approaches. Additionally, they integrate a sinusoidal actuation component. They demonstrate optimization of the shape of pneumatically actuated climbing soft robots [Kob+23], and evaluate the transfer to a physical prototype, but only qualitatively. Ultimately, the auto-encoder approach, as introduced in [Spi+19], empowers the utilization of gradient information to explore expansive design spaces characterized by numerous voxels, all without necessitating prior knowledge regarding the quantity and positioning of design variables.

Secondly, while the controllers' parameters are optimized in certain studies, such as being encoded in the genotype of a soft robot and jointly optimized with design parameters as demonstrated in [Che+18b], they often remain relatively simplistic in their periodic phase changes. Employing systematically periodic actuation patterns inhibits generated robots from mastering complex non-periodic tasks, such as navigating uneven terrains. To address this challenge, the Evolution Gym framework [Bha+22] advocates for employing RL to train the control policy of the designs. They demonstrate the method's efficacy in optimizing soft robot designs for intricate control scenarios (see Figure 2.22.C), yet the majority of their illustrative examples are confined to 2D to ensure the optimization process remains manageable. Another systemic limitation

of designs obtained from morphogenesis lies in the framework's exclusive consideration of pneumatic chambers as actuators, which narrows the spectrum of available options for control strategies.

Finally, the majority of morphogenesis frameworks rely on computationally efficient but somewhat less accurate models of materials and actuators to ensure tractability. For instance, the widely utilized simulator VoxCAD [HL14], which employs a mass-spring based particle model, is prevalent in many studies. Despite its computational efficiency compared to generic continuum mechanics models, there remains a notable absence of successful transitions from designs optimized using morphogenesis methods to functional physical prototypes. For example, researchers in [Kri+20] demonstrate discrepancies between the quantitative and qualitative behaviors of physical models and VoxCAD simulations. Additionally, implementing simulated actuation strategies, such as voxels cyclically expanding and contracting, poses significant challenges in the physical world. Addressing this, modular assemblies featuring magnetic soft voxels with air supply have been proposed to streamline the prototyping process for evolved soft robots [Sui+20] (see Figure 2.22.D).

Summary

In the previous Sections, we have explored various strategies for automating the optimization of soft robot designs. In Table 2.2, we summarize the key attributes of each method category. It is worth noting that this classification is broad, and some works may intersect across multiple categories.

In the next Section, we discuss the link between design and control optimization.

Method	Needed Intuition	Optimized features	Design Space and Optimization Convergence	Transfer to Physical Prototypes
Parametric Design	**Medium to Strong**: The design space consists of variations of design parameters around an initial baseline design conceived by the user.	Case-dependent: Geometrical variations, materials, actuator or sensor locations/shapes. Discovered designs are bound to the user's initial intuition.	In model-based methods, differentiable models linking fitness functions to design parameters are available, and a **large number of design variables** can be considered. Model-free methods rely on algorithms with low sample efficiency, leading to consideration of **small to medium numbers of design variables**.	Demonstrated in many works. It relies on the user experience to build both a well-calibrated simulation for performance evaluation and a well-constrained design space to ensure fabrication constraints.
Topology Optimization	**Medium**: The user has to provide a set of loads and boundary constraints describing their problem. This is difficult for complex design tasks involving several different configurations, such as grasping tasks.	Shape of soft robot components.	The design space is composed of whether materials distribution distributed over a voxel grid or parameters of level-set functions describing shapes. Algorithm convergence is quick regarding **large numbers of design variables** as it benefits from the gradient.	Demonstrated in many works, but restricted to cases where material laws can be assumed to be linear and where the dynamics of the soft body can be neglected.
Morphogenesis Methods	**Low**	Complete robot morphology, including material distribution.	The design space is a mix of material properties, actuator location, and controller variables, all within a voxel grid. **Small design spaces** are considered due to computationally expensive simulation and the low sample efficiency of stochastic algorithms.	Few works have managed to build prototypes, and in the best case, they match the numerical simulations only qualitatively.

Table 2.2: Summary of the main classes of design optimization algorithms and their fundamental properties.

2.3.3 Design and Control Co-Optimization with Sim-to-Real Transfer

Soft robot control and design are typically treated as separate challenges due to their computational complexities. However, there exists a profound interplay between them, as the inherent compliance of soft robot bodies allows them to passively exhibit behaviors that are challenging to program directly. Co-optimization of soft robot control and design should capitalize on this interplay to generate efficient soft robots.

Among the various approaches to soft robot design previously discussed, morphogenesis frameworks align most closely with this mindset. Nevertheless, many of these methods are primarily evaluated in simulation, leaving uncertainty regarding their translation to physical prototypes.

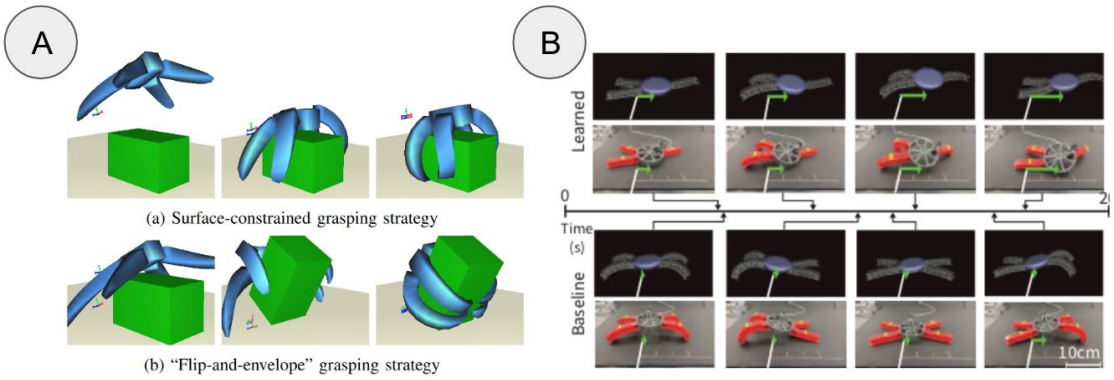

Figure 2.23: Highlights of two works targeting control and design co-optimization of soft robots. A) Two soft hand design and matching control signals obtained with the pipeline from [Dei+17]. B) Comparison of temporal behaviours of a baseline and optimized soft robots for locomotion from [SSW22].

Few works have explored the co-optimization of soft robot design and controller with an assessment of the simulation-to-reality transfer. The common thread among these methods is their reliance on FEM simulation for evaluating design performance, given its widely validated precision in the literature. In their study, authors of [Dei+17] employ particle filter-based optimization to co-optimize finger angles and grasp strategies of a soft pneumatic gripper. While they do not construct a physical prototype, they utilize the SOFA simulator, which is well validated in the literature. However, their method is computationally intensive, involving the simulation of 5.5 million soft gripper design and control strategies.

Recognizing the significant cost of precise simulation, a paradigm shift is underway towards leveraging reduced modeling or surrogates of simulations. For instance, MOR techniques has been applied to mitigate computational complexity of FEM simulation in [SSW22]. This work introduce reconfigurable reduced-order models of soft robot modules blended together to fully describe a soft robot design. They employ multi-task reinforcement learning to jointly optimize the placement of pneumatically actuated soft legs on the soft robot body, as well as control policies for locomotion task. Their approach generates crawling robots that surpass expert designs, with successful simulation-to-reality transfer demonstrated. Recently, the design and control of auxetic lattice structures for soft robots has been adressed in [ZSZ23]. A neural network is trained on FEM simulation data, and employed for gradient-based optimization of design and trajectory control for soft robots with auxetic structures.

These works rely on carefully chosen parametric representation to describe the joint design and control parameters space, ensuring tractable optimization. This is the mindset we adopt in our work.

2.4 Conclusion

This state-of-the-art review explores the fundamental components required to advance towards the co-optimization of design and control in soft robotics. This endeavor hinges on the integration of numerical simulations with a balance between accuracy and computational efficiency, alongside the implementation of efficient controllers and tools for exploring the vast design space of soft robots.

We started by delving into soft robot modeling and simulation, with a particular emphasis on continuum mechanical models. These models have attracted considerable attention due to their ability, under appropriate modeling choices, to accurately depict deformations, a characteristic extensively validated by the soft robotics community. We introduce the discretization of the dynamic equations derived from continuum mechanics and their resolution. These methods are however computationally demanding. While geometrical assumptions can sometimes mitigate this computational burden, they are not always usable, as we target generic shape of soft robots in this thesis work. Furthermore, since our objectives extend beyond control to encompass design optimization applications, the justification for simulations with computational speed that surpass real-time becomes apparent. This supports the exploration of model order reduction techniques. In particular, we introduce a prominent method based on principal orthogonal decomposition, which has been demonstrated in control applications. However, it falls short in capturing design parameters such as geometrical dimensions. This underscores the necessity to develop a new reduced model capable of encompassing both controller and design parameters.

Secondly, distinct methods for controlling soft robots are introduced, categorized based on the degree to which they necessitate breaking down the robot's task into a series of trajectories. These methods include single time step trajectory control utilizing quadratic programming and task-level control using techniques such as reinforcement learning. The overarching aim is to progress towards the latter category, which circumvents the inclusion of user bias in the initial task decomposition into less evident sub-tasks. Nevertheless, this progression also entails an additional computational burden, which we aim to mitigate through the speed and differentiability offered by the reduced model developed in this work.

Lastly, the complexity associated with designing soft robots is developed, particularly in light of the absence of standardized components. We explore the spectrum of available materials, metamaterials, actuators, and sensors, and justify the components used in this work based on factors such as accessibility and fabrication feasibility. Additionally, the main classes of framework for design optimization of soft robots are introduced. We highlight that methods generating more free-form robots often rely on less precise simulations and design specifications that fail to account for fabrication constraints. Given our emphasis on translating numerical findings into physical prototypes, we primarily focus on hand-crafted parametric representations of soft robot designs.

Chapter 3

Optimization of the Parametric Design of Soft Robots Assisted by Surrogate

3.1 Introduction and Chapter Overview

In this chapter, the potential of computational design is explored for facilitating effective soft robot control and improving performances on a designated task, with a specific emphasis on ensuring the transferability between numerical results and reality. The scope of this study is refined given the aforementioned requirements. Exploring the design space of a soft robot requires both a mathematical representation of the soft robot design and a numerical simulation to assess its performance. In this section we choose to work with a parametric representation of the design space and we will rely on FEM to evaluate the quality of the design. On the one hand, the choice to work with a parametric representation of the design space is motivated to ensure both manufacturing and design reproducibility constraints. On the other hand, the utilization of Finite Element Method (FEM) is justified due to its proven precision, reliability, and generic applicability to robots of diverse shapes. Finally, acknowledging both computational and transferability constraints, the focus is intentionally put on low-level tasks.

With all these elements set, the second Section 3.2 of this chapter is dedicated to the resulting generic computational framework for exploring soft robot parametric design. This work gave rise to the development of distributed open-source software [1] [Navc], built as an overlay of the SOFA simulation software. An overview of the proposed framework is given, including mathematical descriptions of the considered parametric design representations, optimization problems, and stochastic solvers. We also introduce automatic mesh generation tools for addressing FEM simulation requirements and components for exploring the solution space in multi-objective settings.

The subsequent sections leverage the developed tools in two distinct design optimization scenarios. These applications showcase the ability of the proposed framework to apply to diverse soft robotic design challenges. Each application involves defining the parametric design of a robot, implementing task-specific objective functions to be assessed in a parametric simulation, and analyzing the transfer of numerical results to a built prototype. Part of these results have

[1] *SoftRobots.DesignOptimization* plugin for SOFA

been presented in two articles [Nav+23; Nav+24].
In section 3.3, a cable-driven soft finger design geometry is optimized for improving integrated sensorization and kinematics. An emphasis is put on showing the method's adaptability to tailor designs to specific user needs, including sensor types. In depth-analysis of the sim-to-real transfer behavior is conducted regarding factors such as mesh density in simulation, mechanical parameters, and fabrication tolerances are discussed.
In section 3.4, a servo-motor actuated soft gripper parametric design is explored to analyze the potential of using self-contacts to enhance grasping quality and energy consumption. Additionally, the influence of friction coefficients at contact points and the shape of the objects to be grasped are examined. It is also demonstrated how to use stochastic optimization for calibrating simulation mechanical parameters, which is particularly useful in highly nonlinear cases such as those involving contacts, where traditional gradient-based methods struggle.

In section 3.5, the advantages and limitations of the parametric design framework are discussed.

3.2 Framework for the Soft Robots Parametric Design Optimization

3.2.1 Context for a Generic Framework for Computational Design Optimization of Soft Robots and with Sim-to-Real Transfer

Within the domain of Soft Robotics, many of novel designs for actuators, mechanisms, and sensors using a variety of materials have been proposed over the last few years. The dissemination of these designs happens through a variety of channels: scientific literature, lab-to-lab sharing, and online platforms. Examples of such platforms are the Soft Robotics Toolkit[2], Instructables[3], and the website for the Soft Robots plugin for SOFA featuring both designs and simulation files[4]. Some of the designs and fabrication methods have since been widely adopted, such as the PneuNet actuator [She+11] [Mos+14] or the Vine Robot [Haw+17].

The available resources allow users to replicate the designs, oftentimes with step-by-step tutorials. Despite this, if they want to adapt a design for their own specific application, they need to get back to the metaphorical drawing board to redesign the parts and everything that is needed for fabrication (molds, laser-cut parts, etc.). However, considering the trend towards computational design in Soft Robotics, this is no longer an up-to-date mindset. The mindset that is starting to emerge is to think of families of designs that are given by a parameterized representation of the devices of interest. Integrated into this way of thinking are simulation tools that are fed by the parametric designs allowing to find optimal ones computationally. This is the type of architecture that we propose in the open source *SoftRobots.DesignOptimization* plugin for *SOFA*, illustrated in Figure 3.1.

Soft robots are still primarily designed by hand by replacing hard parts with soft ones in rigid robots or reproducing what can be found in nature. Thus, there is a growing interest from the soft robotics community for tools enabling efficient design exploration. However, most of the available tools are still restricted to a few soft robot modeling strategies or to specific design applications. In [YHM22], a toolbox for exploring the design of a standardized soft pneumatic actuator based on FEM simulation is introduced. However, this toolbox is limited to a single parametric design and lacks the integration of optimization tools for efficiently exploring the design space. A more complete toolbox for simulating, optimizing the design, and controlling soft robots has been proposed in [Mat+23]. However, it is based on the Geometric Variable-Strain model of Cosserat rod which constrains its application to robots with rod-like shapes. There is a need for a generalized modeling platform enabling to tackle applications in design optimization and model-based control without the need to develop several specific scripts in different software frameworks.

One challenge in the domain of design optimization is the choice of mathematical representation of the design variables. On the one hand, some works are interested in design parameters encoding in discretized spaces, for instance on a voxel grid. Design variables such as materials are then optimized at the scale of each voxel. Although these kinds of approaches enable the generation of almost free geometries, they still lack efficient soft actuator modeling and struggle to integrate manufacturability constraints. A comprehensive review of works in both topology optimization and generative approaches applied to soft robot design optimization is proposed in [PH22]. On the other hand, high-level design parameters encodings consider the robot as an association of modular components [SSW22] or geometrical components of

[2] https://softroboticstoolkit.com/

[3] https://www.instructables.com/Soft-Robotics/

[4] https://project.inria.fr/softrobot/

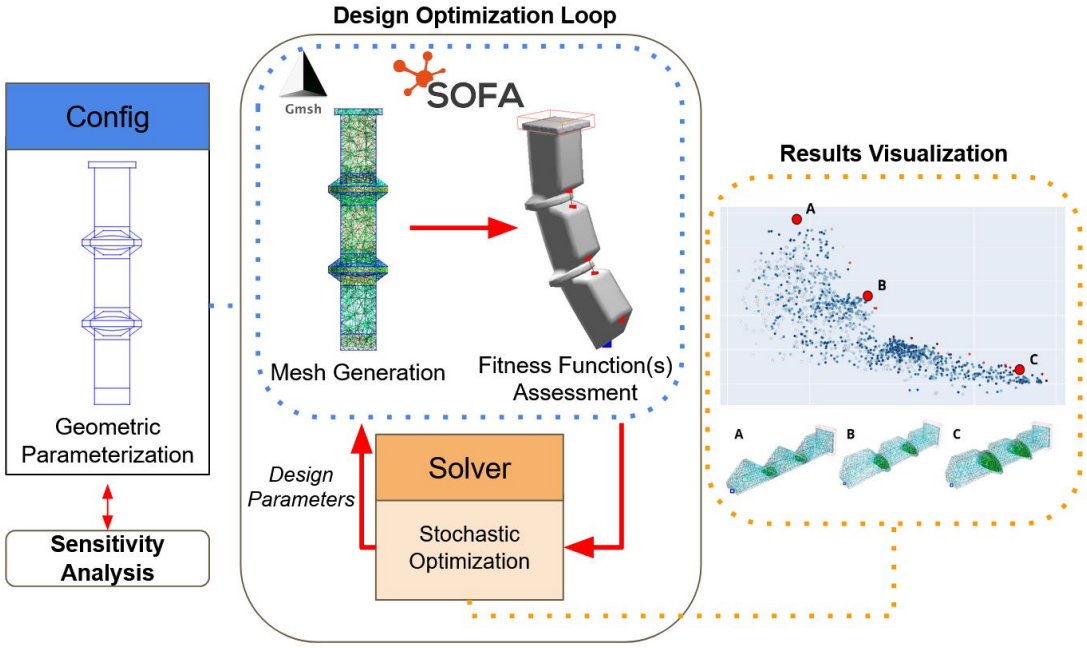

Figure 3.1: The Design Optimization Toolbox is based on fully parametric description of soft robotics devices coupled to the simulation framework SOFA. Users can implement their own *Config* class describing a generic design and multiple fitness functions. Volumetric meshes are generated using the Python API of *Gmsh*, enabling evaluation of the optimization objectives within a SOFA simulation. A sensitivity analysis feature enables exploring relationships between objectives and design parameters. Heuristic-based algorithm are implemented through the *Solver* class for efficiently exploring the Pareto Front in the design space.

variable sizes [YHM22] [MMD19]. These types of encoding rely on the designer's expertise in designing an initial geometric modeling of the robot. Then, optimization is performed for variation around this initial model. The result is a more restricted design space, but one that can more easily ensure compliance with physical fabrication constraints as well as an acceptable optimization time. In this work, we consider the optimization of generic design with a focus on both manufacturability and simulation-to-reality transfer.

In [SSW22], the authors propose a co-optimization of the control policy and the design of a crawling robot, using SOFA. They discuss that the optimization outcome outperforms an expert-designed device. The robot is a disc with eight ports where pneumatic legs can be attached. The design space is thus discrete. The authors chose to employ model-order-reduction [GD18] to make the exploration of the control/design space feasible in terms of time and effort. These works highlight the feasibility of using SOFA for optimization tasks that successfully transfer to real robot platforms. In [Ma+21], Ma et al. introduce their work on optimizing the shape and control of swimmers using a differentiable simulation, which is leveraged for a gradient-based optimization approach. Their work considers both co-optimization of shape and control as well as multi-objective optimization of swimming speed and energy efficiency. An optimal design is found by interpolating between previously established shapes. Thus, their design space is continuous, but limited to barycentric combinations of the initial designs. In [Tap+20], Tapia et al.

present an approach for optimal sensorization of soft robots using stretch sensors to reconstruct their configuration. The robot's mechanics are modeled using the FEM and a model for the resistive type sensors is established, allowing to relate deformations to sensor values and vice-versa. They also simulate the actuation of the robot using a pneumatic chamber. However, the authors do not report optimizing for both actuation and sensing. Finally, in [Spi+21] Spielberg et al. report on a method for co-learning task and sensor placement for soft robots. Their work highlights the importance of optimizing for sensor placement, as this impacts the ability to properly reconstruct the state of the robot from the sparse sensor data. However, their work does not address the challenge of optimizing the shape of the robots and they do not evaluate the simulation to reality transfer.

In summary, a variety of methods exist in literature to optimize the design of soft robots, oftentimes co-optimizing control policies. The choice of design variables (discrete, continuous, interpolated, etc.) as well as the considerations of fabrication constraints are important. Challenges such as sensorization or interaction with the environment are also addressed in a few cases but never all together. However, to our knowledge, a multi-objective optimization for actuation, sensing, and with contact modeling has not been presented yet. Also, while some of the approaches include useful features for optimization, such as differentiable simulations, the design space is often discrete or reduced to combinations of existing designs. We aim to manage completely generic designs using the whole feature set of SOFA available for exploring different modes of actuation, sensing, including contacts and functionality such as model-order reduction. We also provide additional insight for the sim-to-real behavior compared to previous works.

3.2.2 Overview of the Design Optimization Toolbox

The introduced framework is built as an interface between the *SOFA Framework* that provides the simulation and Python libraries for scriptable design and automatic meshing with *Gmsh* as well as heuristic-based optimization. Design optimization is done through the exploration of design parameters with regard to user-defined objective functions. The mechanical behavior of soft robot deformations can be described through continuum mechanics for which there are no analytical solutions in the general case. The toolbox enables the use of many simulation functionalities developed around the *SOFA Framework* to simulate and evaluate a given design, including actuation [Coe+17] and sensor [Nav+20] models, different modeling strategies (Finite Element Method (FEM), Cosserat Rod [ARD21]) for computing numerical solutions, inverse solvers and contact modeling [CED17]. SOFA has in fact already been used for design optimization tasks [SSW22; MMD19], as discussed in the previous section. In order to be able to use the FEM for modeling a soft robot, tools for automatically regenerating meshes when the design parameters evolve are introduced in Section 3.2.3.

The proposed toolbox for design optimization is developed in an object-oriented programming fashion. The user can easily define his own components inheriting from base classes in a well-structured manner. For each class, one to several component extensions are provided as an example. A design optimization problem is represented through the *Config* class. It describes the design parameters and their bounds. This class makes the link with user-defined scripts both for generating new design meshes and for evaluating their fitness function(s) in *SOFA* simulation scene(s). The *Solver* class manages the design optimization and results visualization. Thanks to the modularity of the proposed toolbox, new scripts taking advantage of Python computation libraries are easily implemented. Although the primary focus of the developed framework is to target applications in soft robot design calibration and optimization, it can also be used for non-robotics-related applications such as calibrating design and simulation parameters of rigid or deformable structures.

3.2.3 Design Space Parameterization

The design parameters considered are diverse. One of the prerequisites for the solver algorithms, introduced later, is that the values taken by these parameters are continuous and bounded by box constraints. From a practical point of view, the parameters considered can be parameters of the robot model and its environment (parameters defining the characteristics of the materials, parameters characterizing the contacts, etc.), or even geometric dimensions defining shape variations of the robot, sensors or actuators, around an initial morphology

When the geometrical dimensions of the soft robot evolve, it is necessary to generate the meshes needed for assessing its performance in FEM simulation. The free software Gmsh implements a Python API for one of its backends, the *Open Cascade Technology* (OCCT). This means the user can create arbitrary 3D shapes by building them in a bottom-up fashion starting with elementary entities, i. e. from points to curves to surfaces and finally volumes. In addition, 2D and 3D meshes can be generated from the resulting 3D shape with Gmsh. They are thus readily available for simulation within SOFA or other platforms. Furthermore, the fabrication process can be considered as a part of the computational design phase. For instance, in the case of the sensorized soft finger designs from Section 3.3, the fabrication of the devices based on casting involves several steps. The corresponding molds can therefore be generated automatically from the given finger design to ease the fabrication process. In the case of the contact-aided gripper design from Section 3.4, meshes needed for 3D printing are automatically generated. In both cases, this automation aids in the fabrication process and the dissemination of the designs.

3.2.4 Sensitivity Analysis

Sensitivity analysis methods are statistical and mathematical techniques employed to quantify the variation in model performance in response to changes in parameter values. In our work, it is used systematically as a pre-processing step to reduce the complexity of the optimization process by eliminating design variables having little impact on the considered fitness functions. It computes the gradient of the fitness functions for the design variables. They are normalized and visualized as a graph so that we can easily select the most relevant design variables during optimization.

Another application, as illustrated in one of the experiments of the work on the design of the sensorized soft finger in Section 3.3, is to obtain a better understanding of the design optimization process by assessing the impact of parameter variations on the optimized designs.

For this purpose, a one-at-a-time strategy [BVL16] has been adopted. Starting from a baseline design, optimization objectives are evaluated for bounding values of each design variable. In other words, only one design parameters vary at a time, meaning only variations around the baseline designs are considered. This evaluation lacks the consideration of interactions between variables but has the advantage of being quick to evaluate. The following normalized metric is computed for each variable i, fixing values of all other variables $J = I \setminus \{\mathbf{p_i}\}$ where I is the ensemble of considered design parameters:

$$100 \times \frac{(max_i(f(J, \mathbf{p_i}) - min_i(f(J, \mathbf{p_i})))_{local}}{(max_I(f(J) - min_I(f(J)))_{global}} \tag{3.1}$$

where f is the considered fitness function.

3.2.5 Optimization Problems and Solvers

Both single-objective and multi-objective optimization problems of the following form are tackled:

$$\begin{aligned} \underset{\mathbf{x}}{\arg\min} \quad & (f_1(\mathbf{p}), f_2(\mathbf{p}), ..., f_n(\mathbf{p})) \\ s.t. \quad & a_i < p_i < b_i \quad \forall p_i \in \boldsymbol{p} \end{aligned} \tag{3.2}$$

where the integer n is the number of objectives and p_i are continuous design variables bound by box constraints.

We chose to rely on stochastic algorithms for addressing these optimization problems, for several reasons that are now listed. These algorithms leverage randomness and probabilistic components to explore solution spaces. They do not need to compute gradients for the fitness functions relative to the design parameters, which are not available especially when considering geometrical parameters, as building explicit meshes for FEM simulation is not a differentiable operation. For instance, having the same number of nodes when regenerating a mesh is not guaranteed. Stochastic algorithms usually offer a diverse set of techniques for escaping local optima and enabling them to navigate complex landscapes. However, they do not provide any guarantee to find the optimal solution but instead converge toward near-optimal solutions. In the context of multi-objective optimization, it is usually not possible to find a solution minimizing all the objective functions. The stochastic nature of these algorithms enables the exploration of diverse regions of the Pareto front, facilitating the discovery of solutions that represent optimal compromises between conflicting objectives. In that case, we are interested in Pareto optimal solutions i. e. solutions that cannot be improved regarding any objective function f_i without deteriorating their result concerning at least one of the other objective functions.

Stochastic solvers implementation is provided through the open source library *Optuna* [Aki+19] targeted at hyperparameters optimization. In our work, evolutionary and bayesian algorithms are both considered. One additional interest of these classes of algorithms is that they are fully parallelizable, i.e. evaluations of design in simulation can be distributed on several different processes for reducing computation time. The general principles of these two classes of algorithms are described below.

Evolutionary algorithms

Rather than optimizing a single candidate, evolutionary algorithms evaluate a population of candidates evolving over generations, mimicking the natural selection principles met in nature. They consist of four overall steps, corresponding each to a particular facet of natural selection. There are many variants of these steps, which makes it easy to modulate implementations of such algorithms. A typical workflow for an evolutionary algorithm is shown in Figure 3.2.

Candidate solutions to the optimization problem play the role of individuals in the population. During the selection phase, the best individuals regarding the objective to evaluate are selected. Then, the population is replaced through the breeding of the selected individuals, crossover, and random mutations.

Covariance Matrix Adaptation Evolution Strategy (CMA-ES) [Han05] and Non-dominated Sorting Genetic Algorithm (NSGA-II) [Deb+02] variants are respectively used for mono and multi objectives optimization.

Bayesian algorithms

Bayesian optimization is typically used to optimize expensive-to-evaluate functions that can have up to a hundred parameters. However, few applications consider Bayesian algorithms for

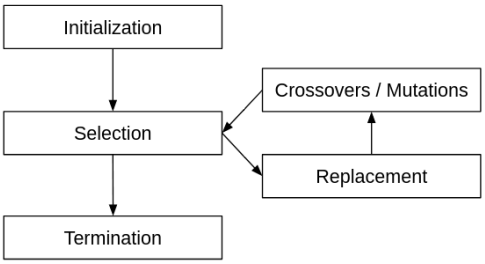

Figure 3.2: Workflow for an evolutionary algorithm.

the design optimization of soft robots. In general, these optimization works consider quick-to-evaluate objectives and rely on evolutionary algorithms. Nevertheless, Bayesian methods have the advantage of converging faster than evolutionary methods by making assumptions on the underlying fitness landscape, at the cost of a smaller exploration of the design space. This feature is all the more interesting when the evolution of the cost function is particularly time-consuming, or when the user is not interested in exploring the entire design space but rather in quickly finding a satisfactory solution. An example of such a scenario is when considering the calibration of mechanical parameters of a soft robot.

In Bayesian optimization [Fra18], the objective function is considered a random function. A probability distribution called a surrogate is iteratively trained for giving a belief about the quantity of this function. The Bayesian strategy alternates between quantifying the uncertainty in the surrogate, building an acquisition function to determine the next query data point, evaluating its objective, and updating the surrogate.

Both the Tree-structured Parzen Estimator (TPE) [Ber+11] algorithm and its extension for multi-objective optimization (MOTPE) [Oza+20] are implemented.

3.3 Design Optimization of a Sensorized Soft Finger

3.3.1 General Description of the Soft Finger

In this work, we develop the *SoftRobots.DesignOptimization* plugin to experiment the methods proposed in the previous Section. We apply these methods to optimize a soft finger that we wish to use to build a soft gripper in the future. The current design is based on a previous work [Nav+20]. The first step is to present a completely parametric version of the soft finger that can readily be simulated with *SOFA*, as shown in Figure 3.1. For describing the parametric designs and generating the meshes needed for simulation, the Python API for the free software *Gmsh* [GR09] is used. The 3D-printed molds necessary for the silicone casting steps during the fabrication are derived automatically from the design parameters.
Considering the action of displacing the cable by a unit amount of 10 mm, we want to find the design achieving the largest variation of the finger's cavities volume, for sensing purposes, and at the same time the largest bending angle, for dexterity purposes. These objectives are called deformation volume and angular displacement, respectively. We consider this task to be a proof of concept for the use of the *SoftRobots.DesignOptimization* plugin that showcases the optimization and trade-off of performance between sensor efficiency and finger dexterity. Then, instances of the soft fingers developed with the presented tools are fabricated and we take an in-depth look at the question of how well the optimization results predict performance in reality, i. e. an analysis of the sim-to-real transfer. Finally, it is shown how the current design and numerical optimization framework could be extended for configuration estimation under perturbation with the environment i. e. external contacts.

3.3.2 Numerical Implementation

Parametric Design

The geometrical dimensions of the finger are considered as design parameters. The ones subject to change during optimization of the finger are shown in Figure 3.3 on the left, on the example of our initial design *finger baseline*. All other parameters remain constant, in particular, the thickness, height, and length of the finger. The cable is at a fixed distance from the outer radius of the cavity. Thus, the cable passes closer to the center of the finger when a small outer radius is chosen for the cavities.

We furthermore consider the fabrication process as a part of the computational design phase. The fabrication of the devices based on casting involves several steps. The corresponding molds can therefore be generated automatically from the given finger design to ease the fabrication process, as is shown in the example of the Soft Finger (see Figure 3.4).

Simulation and Fitness Functions

For simulation, we use the FEM model together with the constraint-based cable and pneumatic cavities models from the Soft Robots plugin [Coe+17]. The assessment of the fitness functions is made for the model at static equilibrium. The fitness functions for the quantifying the deformation volume, f_1, and the angular displacement, f_2 are, respectively, given by:

$$\begin{aligned} f_1(\mathbf{p}) &= |Vol(\mathbf{p}, \delta_a^{10}) - Vol(\mathbf{p}, \delta_a^0)| \\ f_2(\mathbf{p}) &= \frac{|\arccos(|P_z^{Tip}(\mathbf{p}, \delta_a^{10})|)}{\|P^{Tip}(\mathbf{p}, \delta_a^{10})\|} \end{aligned} \tag{3.3}$$

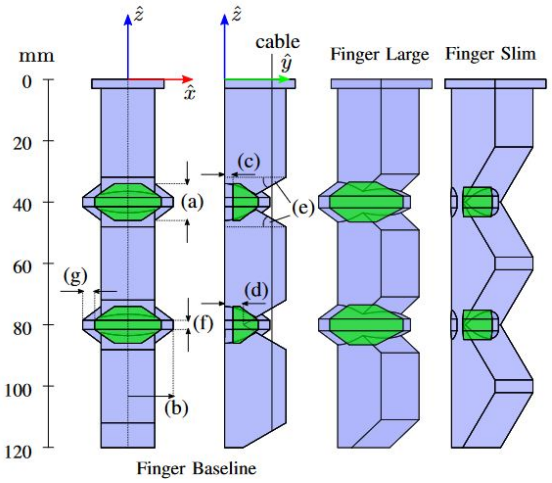

Figure 3.3: Left: The parameters subject to optimization are the following: (a) Cavity Height, (b) Outer Radius, (c) Cork Thickness, (d) Joint Height, (e) Joint Slope Angle, (f) Plateau Height, (g) Wall Thickness. Right: The two instances of the finger selected for evaluation, the *finger large* and *finger slim*.

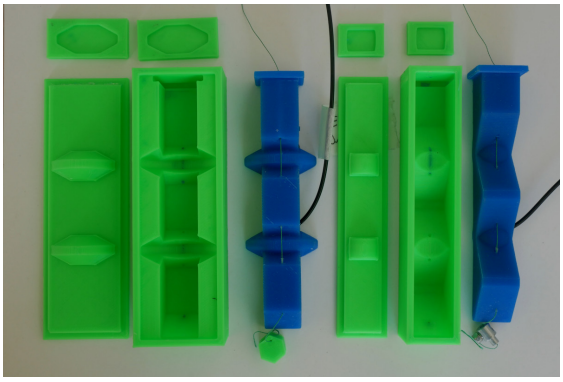

Figure 3.4: Both fingers, large and slim, fabricated using the corresponding automatically generated molds.

where $\mathbf{p}$ is the set of design variables. Here, $\mathbf{p}$ is of dimension 7 and is displayed in Figure 3.3, δ_a^0 and δ_a^{10} are the actuation displacements imposed on the cable actuators of respectively 0 mm and 10 mm. The operator *Vol()* gives the volume of a cavity and P^{Tip} is the position of the tip of the finger. The function f_1 then computes the deformation volume (the variation of volume) in the cavity for a controlled actuator displacement, which corresponds to what a flow sensor can measure (see Section 3.3.4), whereas the function f_2 is the angular displacement reached by the fingertip under the same actuator displacement constraints. Notations are introduced in Figure 3.5.

Simulating the sensorized finger is performed using an Euler implicit time integration method with a time step of 0.01 seconds. Other detailed parameters are available in the open source code on GitHub [Navc].

For the optimization, we made assumptions about the mechanical properties of the finger.

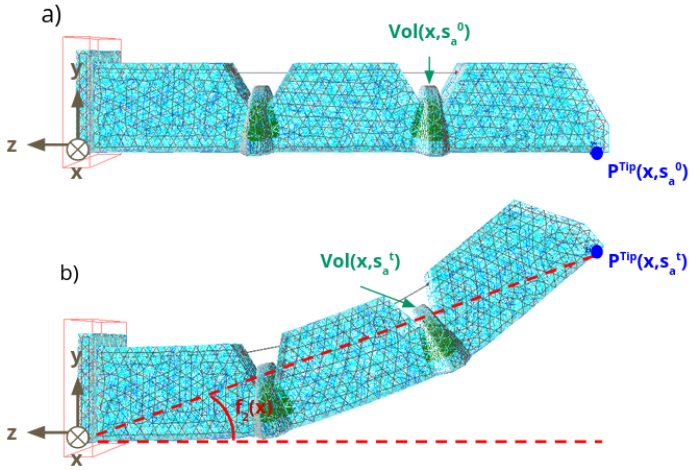

Figure 3.5: Illustration of the optimized fitness functions. a) Initial equilibrium of the soft finger. b) Equilibrium of the soft finger for an imposed actuation displacement s_a^{10}.

We used a Young's Modulus of 3 MPa and a Poisson's Ratio of 0.30, as well as damping coefficient parameters related to mass and stiffness of 0.1. For a given set of design parameters, the objectives are assessed using a 3D mesh having about 500 nodes. With this settings, the simulation takes around 2 seconds to reach equilibrium for each design on our setup[5].

3.3.3 Numerical Results

Optimization Results

We use the implementation of the NSGA-II multi-objective evolutionary algorithm from the *Optuna* library as a solver. Algorithm hyper-parameters and results are displayed in Figure 3.6. Results design variables can be explored in the GitHub repository.

The best performing results in Figure 3.6 with respect to the angular displacement objective are geometries with less material and cavities flattened along the finger length such as design B. The volume of the cavities is then minimal, thus facilitating a more significant deflection of the finger because the cable can pass closer to the joints. Conversely, the best results regarding the sensitivity to deformation volume consider the largest possible cavities having thin walls, as in design E. The Pareto front enables efficient exploration of the design space and to find compromises like that of design C.

Later, in Section 3.3.5, it is discussed more in detail that the results obtained seem to generalize well to other mechanical parameters (Young's Modulus and Poisson's Ratio) and discretization sizes.

Considerations on the Choice of the Fitness Functions

The choice of fitness function has a big impact on the optimization outcome. If pressure measurements are used rather than air-flow sensors, the optimal geometries are quite different.

[5]Laptop with height cores 2.70 GHz Intel Core i7 - 6820

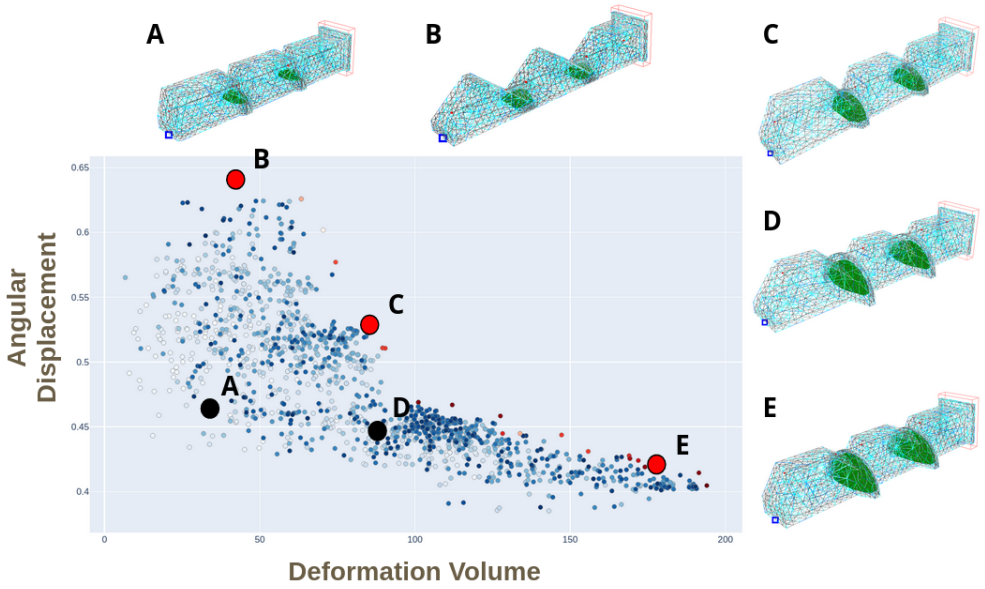

Figure 3.6: Pareto Front obtained as well as geometries of a few sampled designs for the Sensorized Finger design optimization. Results are generated using NSGA-II algorithms with an initial population of 50 design candidates, a probability of crossover of 0.9, and a probability of swapping parameters between parents of 0.5. Each point is the evaluation of a design in simulation. A total of 1500 designs were considered. The Pareto optimal solutions are represented by red dots.

To test this, we designed alternative fitness functions as follows:

$$\begin{aligned} f_3(\mathbf{p}) &= \frac{f_1(\mathbf{p})}{Vol(\mathbf{p}, \delta_a^{10})} \\ f_4(\mathbf{p}) &= Vol(\mathbf{p}, \delta_a^{0}) \end{aligned} \tag{3.4}$$

where f_3 characterizes the change of pressure of the pneumatic chamber under imposed displacement and f_4 its initial volume. The resulting Pareto Front is shown in Figure 3.7.

This time, contrary to the results obtained when evaluating the deformation volume, the flatter the pneumatic cavity, the more effective is the pressure change. The f_4 objective on the initial volume of the cavity was added to avoid having pneumatic cavities that are too small and impossible to manufacture. This example demonstrates the relevance of using the provided toolbox for generating designs in accordance with different physical specifications.

3.3.4 Experimental Validation

The test bench used to validate the performance of the fingers is shown in Figure 3.8. For all experiments, we are interested in the effect of a displacement of 10 mm of the cable, as discussed in Section 3.3.2. To evaluate deformation volume, we have attached an air-flow sensor D6F-P0001A1 by the OMRON Corporation[6] to the distal cavity. The analog signal of the sensors

[6] https://omronfs.omron.com/en_US/ecb/products/pdf/en-D6F_series_users_manual.pdf

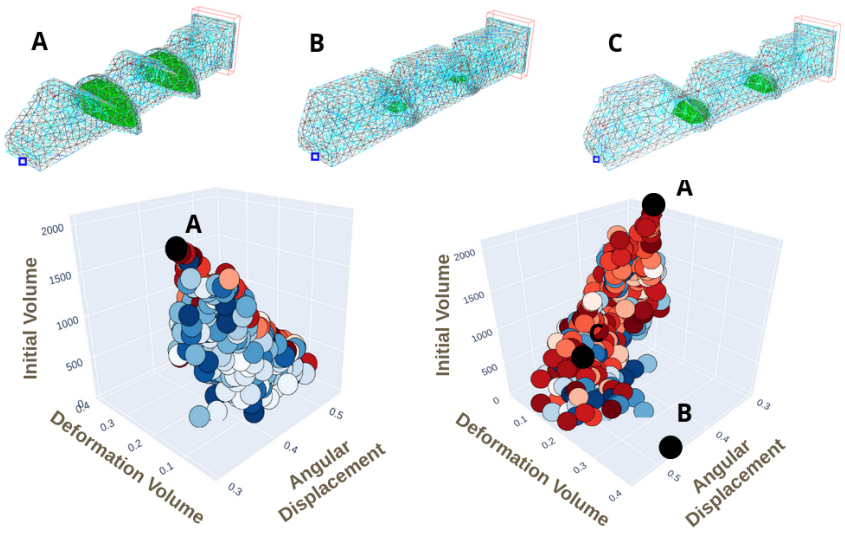

Figure 3.7: Pareto Front obtained from the design optimization of the Sensorized Finger for using a pressure sensor. Geometries of a few sampled design are also shown: **A)** $f_2(\mathbf{p}_A) = 17.2°$, $f_3(\mathbf{p}_A) = 0.085$, $f_4(\mathbf{p}_A) = 2043\,\mu L$, **B)** $f_2(\mathbf{p}_B) = 26.9°$, $f_3(\mathbf{p}_B) = 0.395$, $f_4(\mathbf{p}_B) = 47\,\mu L$, **C)** $f_3(\mathbf{p}_C) = 26.9°$, $f_3(\mathbf{p}_C) = 0.132$, $f_4(\mathbf{p}_C) = 572\,\mu L$.

is digitized using an Arduino UNO board. On the same board, the time integration is done to obtain the volume measurements from the flow. This technique enables measuring very small volumes and exhibits high repeatability, as previously reported in [Nav+19]. We did not look into using pressure sensors here, because the tubing and the sensor itself represent a rigid volume that needs to be taken into account to obtain volume estimates. However, determining this rigid volume is a more involved process whereas the measurements of the air-flow sensors can be interpreted directly as is. The angular displacement is measured from pictures, as shown in Figure 3.9.

For validating the results obtained by running the optimization, we have fabricated three instances of the finger. We created one *finger slim* and two identical *finger large* (#1 and #2), as illustrated in Figure 3.3. The idea of fabricating the same design twice was to assess the variability introduced by the fabrication process. In particular, sealing the cavities involves a manual gluing process that we assumed could impact the deformation volume. We first present the results of what is predicted by the simulation in Table 3.1. It is shown that a large difference of 515% exists between the predicted deformation volume of the slim and large fingers. The change in angular displacement is not quite as dramatic, but still clearly noticeable, i. e. an increase of 9.03° or 147% from large to slim.

Device	Deform. Vol.	Ang. Disp. (deg.)
Finger Baseline	41.5 µL	21.1
Finger Slim	29.7 µL	28.13
Finger Large	153.1 µL	19.1

Table 3.1: Deformation Volumes in SOFA

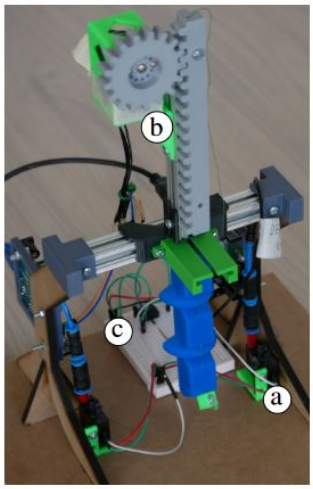

Figure 3.8: Test bench for evaluating the finger's performances: (a) Air-flow sensor, (b) linear actuator and (c) reset button for integration.

Deformation Volume

For assessing the deformation volume, each experiment was repeated 5 times. The mean value, together with the standard deviation is reported for all experiments in Table 3.2. In Table 3.3, the metrics provided enable a comparison of the experiments. We first note that the experiments demonstrate high repeatability, with a series of 5 repetitions showing a deviation of less than 1%, except for the slim finger, where the deviation is slightly higher. This is probably because the overall measured volume is small and thus the measurements are more affected by the noise floor.

Device	Deform. Vol. (µL)	Std (µL)	Std (%)
Finger Slim T1	45.7	0.50	1.10
Finger Slim T2	45.5	0.81	1.78
Finger large #1	300.7	2.24	0.75
Finger large #2 T1	304.7	1.46	0.48
Finger large #2 T2	307.9	1.75	0.57

Table 3.2: Sensitiviy of Deformation Volume Measurements

Metric	Value
Gain slim to large (sim.)	5.15 (515%)
Gain slim to large #2	6.74 (674%)
Diff finger large #1 vs #2	7.21 µL (2.40 %)
Diff slim T1 vs T2	0.22 µL (0.49 %)
Diff large T1 vs T2	3.25 µL (1.06 %)

Table 3.3: Metrics for comparisons

When comparing the slim finger with the large finger, we observe in Table 3.3 that the gain in Deformation Volume is 6.74 (674%), which is a quite significant increase and is also a figure

similar to what was predicted by SOFA, which is a gain of 5.15 (515%). When we compare the two identical large fingers, we observe a difference of 7.21 µL or 2.4%. This indicates that the variability in deformation volume introduced by fabrication is much lower than the variability due to design. Finally, when comparing the results of the exact same fingers (slim and large #2) after remounting the fingers into the test bench (trial 1 and trial 2), we observed a variability of 0.22 µL (0.49%) and 3.25 µL (1.06%), respectively. We therefore conclude that variability due to fabrication and experimental conditions is small compared to the variability due to design. Thus, the optimization process run in simulation has in fact predicted a significant change in performance for the deformation volume on real-world devices.

Angular Displacement Performance

To evaluate the performance of the designs in terms of angular displacement, we have manually labeled pictures of the angular displacements with respect to the rest position of both the slim and large fingers, as seen in Figure 3.9. The results have been summarized in Table 3.4. The angular displacement is measured with respect to the vertical line (the $\hat{z}$-axis) and the reference point on the finger is at the back of its tip, as shown in Figure 3.9. There is a very good agreement between the predicted difference in angular displacement of both designs. However, the absolute values differ slightly. Overall these results confirm that the simulation can predict the difference in performance between the designs.

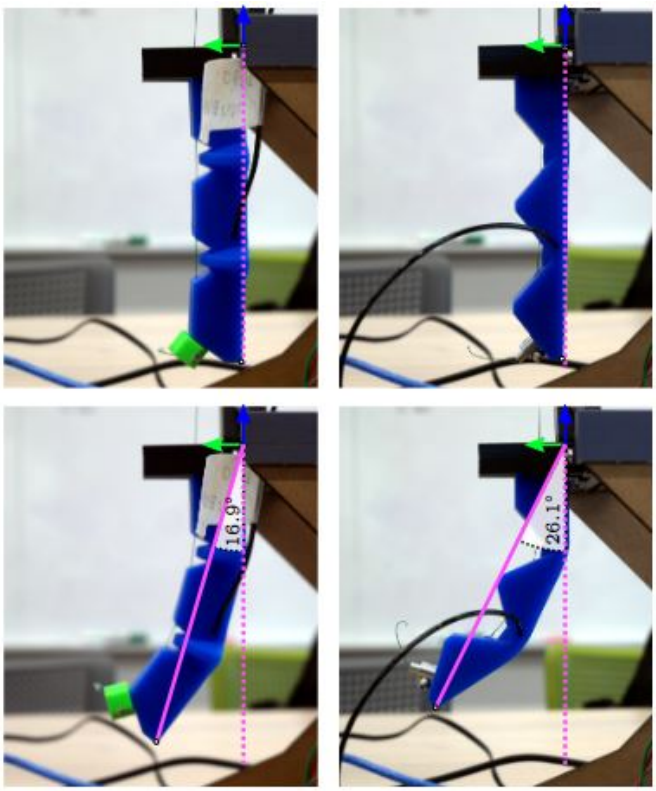

Figure 3.9: Comparison of angular displacement between the large finger (left) and the slim finger (right).

Angular Displacement (deg.)	Reality	SOFA
Finger Slim	26.1	28.13
Finger large	16.9	19.1
Diff	9.2	9.03

Table 3.4: Comparison Angular Displacement

3.3.5 Considerations on Sim-to-Real Transfer

In this sub-section, we dig deeper into the question of how well the results obtained with the help of the simulation transfer to reality. The main points addressed are the mesh density employed, the mechanical parameters chosen (Poisson's Ratio and Young's Modulus) as well as the influence of tolerances due to fabrication.

Mesh Density and Deformation Volumes

We used Gmsh's ability to locally define mesh sizes to selectively increase the mesh density around the cavities, as shown in Figure 3.10. We observed that increasing the mesh density can affect the estimate of volume change significantly. We therefore repeated the same simulation increasing each time the mesh density to observe this effect. The results are shown in Figure 3.11. A higher mesh density captures more detail of the deformation around the cavity and the overall trend is for the sensitivity to increase. However, as the number of nodes employed increases, the estimate converges, as the variation in sensitivity per increase in number of nodes, i. e. $\Delta V/\Delta \#N$ converges to 0.

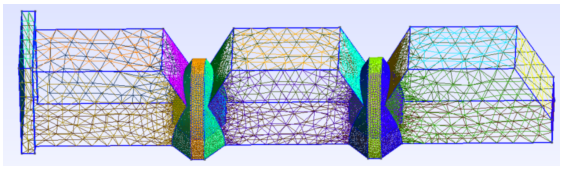

Figure 3.10: Local mesh size

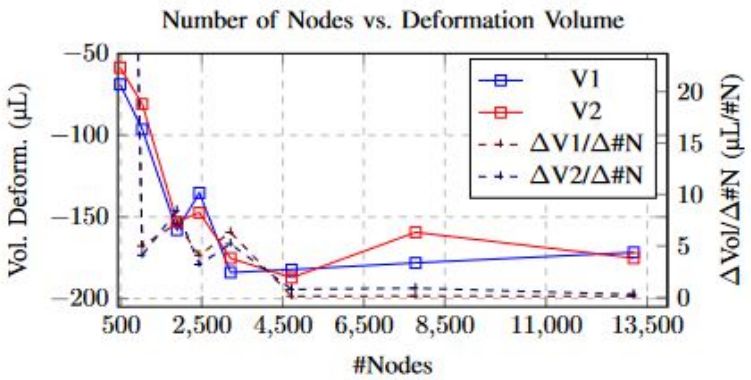

Figure 3.11: The effect of the number of nodes chosen for representing the finger in SOFA.

As discussed later, in Section 3.3.5, the Pareto Front for our optimization problem is not very

sensitive to the mesh size. Therefore, lower resolutions can be chosen here. Furthermore, this analysis points to the issue of the sim-to-real gap as deformation volumes are not predicted exactly, even for high mesh densities. Possible explanations include that we are ignoring local self-contacts at the folds and tolerances in the fabrication process (see Section 3.3.5). It should also be considered that the estimated deformation volumes are less than 10% of the total cavity volume. In this specific case, our manufacturing process and the current tolerances for these soft robotics fingers do not require models more precise than those we will be using subsequently.

Pareto Front Variability

At first, when exploring the optimal design of the soft finger in Section 3.3.3, we made two strong assumptions without prior calibration with a physical prototype as stated there, which were a Young's Modulus of 3 MPa, a Poisson's Ratio of 0.3 and a mesh density of 500 nodes (see also Figure 3.12). To investigate the variability of the Pareto Front, we ran two optimization procedures, once with a Poisson's Ratio of 0.45, having a mesh density of 2500, and once with a Poisson's Ratio of 0.495 and a mesh density of 500 nodes. Here, we show that the Pareto Fronts obtained, shown in Figure 3.12, are not very dependent on these two parameters. Points on the extremes of the compromises are basically identical designs on either Pareto Front and correspond to the finger slim and finger large. This demonstrates that using a lower-resolution mesh and non-calibrated mechanical parameters to explore our Pareto Front is feasible. Nevertheless, this should not be taken for granted and would need to be verified for other optimization tasks.

Note that we do not consider here the Young's Modulus because it has little impact on the result. This is due to the performance metrics, the deformation volume of the cavities, and angular displacement, which are geometric in nature. An assessment of the finger, for example taking into account contacts in the optimization problem, would have required a prior calibration of this parameter. The same is true for an actuation based on forces on the cable rather than displacement, which would also have required accurate mechanical parameters calibration.

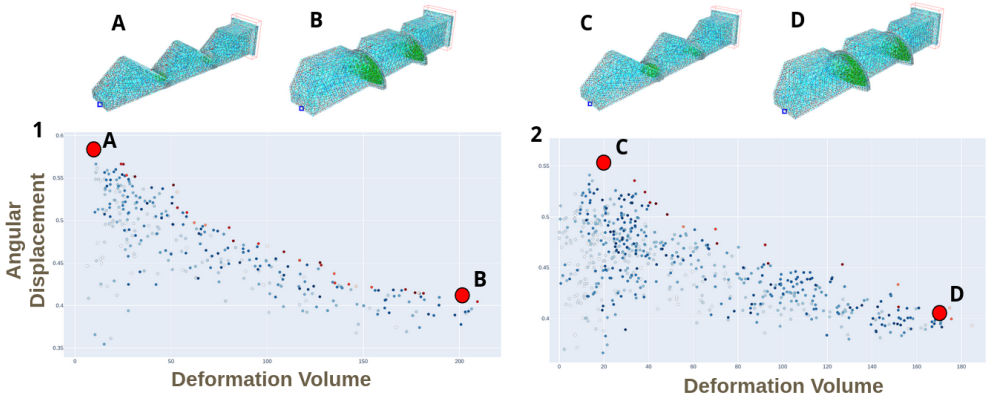

Figure 3.12: Pareto Fronts as well as optimal geometries obtained regarding both deflection and Deformation Volume for refined meshes and Poissons's Ratios values of 1) 0.45 with 2500 nodes and 2) 0.495 with 500 nodes. Simulation times range from 2 seconds per design with meshes of 500 nodes to 15 seconds for meshes of 2500 nodes on our computational setup.

Fabrication Tolerances

We also are interested in studying the effect that fabrication tolerances can have on performance metrics. Since our process of fabrication is based on 3D-printed molds, we expect tolerances in the fabricated devices related to this process, such as larger than intended borders. These in turn can result in thinner-than-expected features in the silicone casting process. We have looked at the parameter *Wall Thicknes*, (g) in Figure 3.3, of the large finger and we measured that it exhibited a variation of up to 0.4 mm, going down from the intended 3 mm to 2.6 mm. As shown in Table 3.5, when simulating the device with the adjusted *Wall Thickness* parameter, using 3209 nodes (see Figure 3.11), there is a significant impact on the deformation volume, an increase of about 24%. This is an important reduction of the sim-to-real gap. We have provided a *sensitivity analysis* for the deformation volume with respect to the free design parameters shown in Figure 3.13. We used a One-At-a-Time strategy. Starting from the finger large design (see Figure 3.3), optimization objectives are evaluated for the bounding values of each design variable. This evaluation lacks the consideration of interactions between variables but has the advantage of being quick to evaluate. It is confirmed that the Wall Thickness is indeed a very sensitive parameter for the deformation volume optimization objective. This gives another clue for the calibration procedure: a model calibration could consider fabrication tolerances, for instance by finding the design parameters within a tolerance interval that minimizes the sim-to-real difference.

Wall Thickness	Deform. Vol.
3 mm	−175.24 µL
2.6 mm	−217.00 µL

Table 3.5: Analysis Fabrication Tolerances Effect

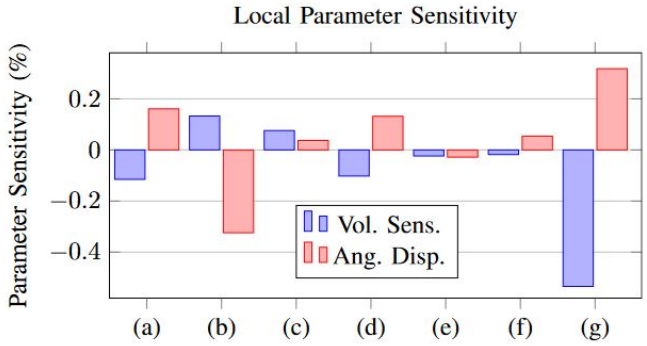

Figure 3.13: Analysis of local parameter changes around the large finger design using the One-At-a-Time strategy. The labels correspond to the parameters described in Figure 3.3. The values for each label with respect to each objective correspond to the maximum objective differentials encountered by separately varying each parameter with respect to the initial parameters of the large Finger. These values are then normalized by the maximum objective differential encountered by varying all design variables. Both objectives are mostly sensitive to a) Cavity Height, (b) Outer Radius and (g) Wall Thickness parameters.

3.3.6 Extension to Configuration Estimation with Capacitive Touch Location Sensing

Introduction and Motivations

As a contribution to the *Soft Robotics Toolkit Competition 2023*, we close the bridge between the soft finger design proposed in [Nav+20] and the *SoftRobots.DesignOptimization* plugin. This section focuses on the reproducibility of the method. Instead of sharing static CAD files, it provides both an in-depth fabrication tutorial and parametric designs that can be used in optimization frameworks, while automating part of the fabrication process by generating the corresponding molds automatically.

Compared to the previously introduced version of the soft finger design, the main difference consists in the addition of capacitive sensing functionalities. The objective is to perform configuration estimation using model-based sensing. The mechanical model of the device is leveraged to interpret the sensor values. The interpretation happens through an inverse model implemented in *SOFA* that yields the forces/deformations of the device that best explains observed sensor values.

Another slight difference is that the pneumatic cavities are now in two parts at the level of each section (see results in Figure 3.15). This slightly complicates the manufacturing process but generates less opposition to the cable closing course compared to the previous design.

The capacitive touch location sensing enables to determination of where the interactions are happening on the finger, whereas the sensing of changes in air pressure inside of the airtight cavities embedded in the finger enables the quantification of deformation. The touch location sensors provide the model in *SOFA* with crucial information that helps to correctly attribute the observed effects, i. e. deformation sensed by the air-pressure sensors, to the causes, i. e. cable displacement and external forces. Furthermore, touch sensing is only used to determine the contact location, not deformation. This makes the design of the capacitive sensing element relatively simple. This design is very much like the one found in touch screens of consumer electronic devices. Illustrations of the capacitive sensing operating and configuration estimation pipeline are provided in Figure3.14.

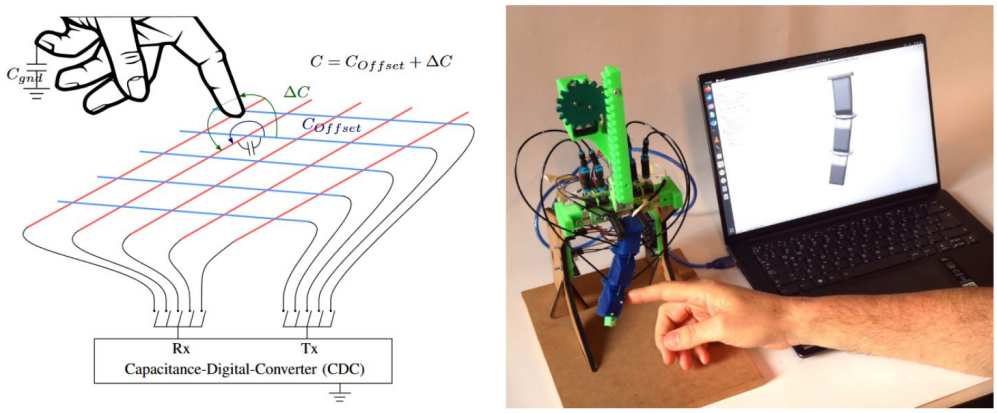

Figure 3.14: On the left, illustration of multi-point mutual capacitance. On the right, an example of interaction with the soft finger with configuration estimation in *SOFA*.

Design Optimization

As air-pressure sensors are used instead of volume sensors, we consider the same fitness functions as for the experiment from Section 3.3.3. First, the kinematics of the Soft Finger is evaluated as the *angular displacement* of the tip under a fixed cable displacement of 10 mm. Secondly, the *pressure sensibility* is computed as the ratio between the change in volume of the cavity under deformation versus its volume at rest. Here, the minimum pressure change in a cavity under two interaction scenarios is taken. The first one consists of a fixed cable displacement of 10 mm and the second one of a fixed force applied on the lateral edge of the Soft Finger. High-pressure sensitivities are reached for flat cavities, which are not feasible to fabricate. To avoid having a non-feasible design, the *initial volume* of the cavities is computed as an additional fitness function.

The assessment of the fitness functions is made for the model at static equilibrium. Numerical results obtained with the provided tools are displayed in Figure 3.15. The best designs in terms of kinematics tend to favor flat cavities in the direction of the length of the finger (see design **D**) and less material (see design **C**). However, best designs regarding pressure sensibility tend to privilege flatter cavities in the direction of the height of the Soft Finger (see design **A**), as this direction is subjected to more deformation when the Soft Finger is subject to a lateral force. A good compromise between all considered fitness metrics is therefore design **E**.

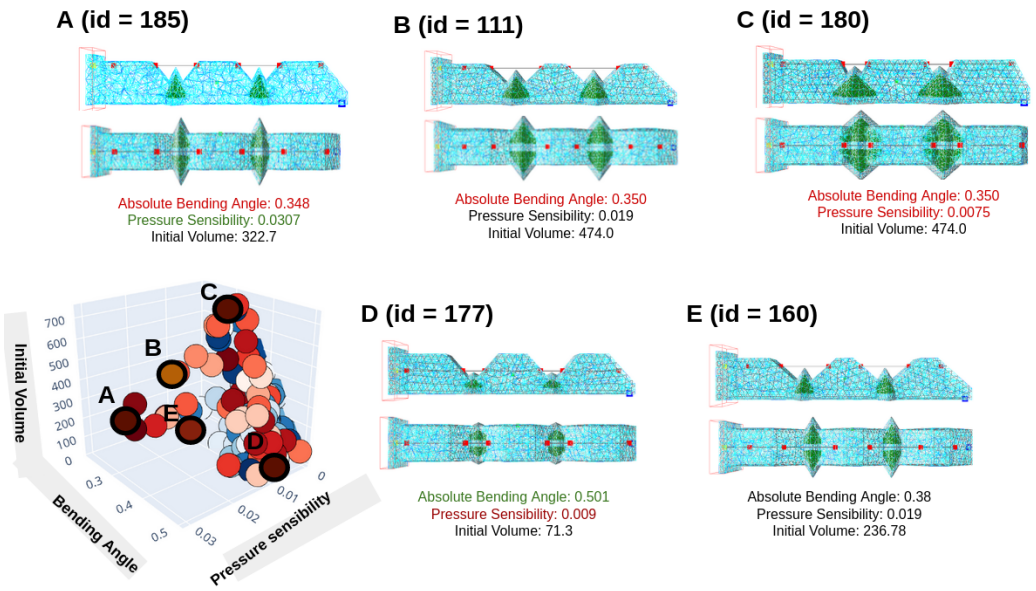

Figure 3.15: Pareto Front and a few sampled geometries obtained for 185 trials of the Soft Finger design optimization. Pareto optimal solutions are represented by red dots. Results are generated using the NSGA-II algorithm with an initial population of 50 design candidates, probabilities of crossover, and swapping parameters between parents of 0.9 and 0.5, respectively. The best scores and worst scores regarding each fitness function are respectively written in green and red.

This work represents a proof of concept, as an in-depth study of the simulation-to-reality transfer has not yet been carried out.

3.3.7 Conclusion

These contributions illustrate how to use the computational plugin *SoftRobots.DesignOptimization* and evaluate its performance on the example of a cable-driven soft finger featuring embedded flow sensors. The finger design task highlights the need for a multi-objective optimization framework because optimizing at the same time for actuation capabilities and for sensing capabilities can be conflicting goals. Thus, a trade-off needs to be found. We provided an experimental validation of the proposed optimization procedure. For a given displacement of the cable inside the finger, we chose to optimize the angular deflection achieved as well as the deformation volume measured inside the embedded cavities. It was shown with the help of two fabricated devices on the extremes of the Pareto Front, called finger slim and finger large, that the toolbox correctly predicts significant variations of both optimization objectives. We furthermore provide a more in-depth analysis of how the simulation parameters chosen affect –or not– the real-world outcome of the optimization for our task. As an extension, we also considered optimizing for the sensitivity to interactions with the environment.

In the future, we would like to explore further devices and design objectives. In particular, we would like to explore more complex devices, such as a gripper, whose basic building block could be the finger presented here. Objectives on the kinematics and sensing of the sensorized finger were considered. These objectives were chosen for the ease of measuring their performance on a test bench. This work could be extended to optimizing the design of an entire soft gripper for manipulation by considering additional design parameters, such as the location of the fingers in relation to each other, as well as additional fitness functions evaluating the contact forces and manipulation dexterity of the robot. Although on a different design, we consider optimizing a soft manipulator while taking into account contacts in the next section.

3.4 Design Optimization of a Soft Gripper using Self-Contacts

In this section, we aim to optimize the design of a soft gripper for manipulation, emphasizing the role of contact interactions. Our objective is to explore how self-contacts influence the robot's behavior and explore their potential in designing more effective soft grippers and manipulators.

3.4.1 Background on the Design of Contact-Aided Soft Robots

Designing and assessing both soft grippers and soft manipulators is a broad area of research getting more and more attention from the soft robotics community [Hug+16] [Tee+20].

A soft gripper design is classically optimized in a multi-objective setting. However, the design task is difficult regarding the stiffness of the deformable part. The latter has to be able to easily reach the object to be handled while transmitting significant forces for grasping and handling. On the one hand, a consistently soft manipulator will easily reach the object but will eventually fail to pick it up. On the other hand, a constantly stiffer manipulator will require a lot of effort and energy just to reach the object. Similarly, developing soft grippers capable of generating a desired grasping force requires them to be soft enough to conform to the object shape and maximize the contact area, and rigid enough to transfer loads. Therefore a compromise has to be found between the abilities of the gripper both to deform during the grasping phase and to stiffen during object manipulation.

Satisfying these requirements has been addressed through active stiffness modulation [MCC16]. Another interesting approach is to leverage self-contacts. These can be used to create closed parallel chains for specific actuation values and therefore stiffen the structure. Few works consider using self-contacts in their soft robot design. One famous example is the design of a fast PneuNet actuator developed in [Mos+14]. The actuator is composed of an inextensible layer deformed by a series of cube-like chambers, whose walls collide upon inflation to create additional forces. In [Mor+20a], a bio-inspired soft manipulator robot based on a compliant spine uses self-collision of its vertebrae to limit local deformations. Finally, a catheter made with several segments with joint stops is introduced in [Gao+20] that can stiffen locally and achieve a desired shape for the same actuation force. These soft robot designs exploit a-posteriori self-collisions and there is no general methodology to design soft robots using self-contacts. Moreover, there is, to the best of our knowledge, no analysis of the influence of self-contacts on the robot behavior and their potential for designing grippers and manipulators.

Given the trend toward computational design in soft robotics, there is a growing interest for sharing parametric families of designs coupled with simulation tools, that can be easily adapted to the user needs through computational optimization. Few works on design optimization frameworks targeting contact-aided soft robot design have been proposed. In [Ros+19], design optimization of contact-aided continuum robots is addressed. In the field of compliant mechanisms synthesis, a computational framework has been developed for optimizing the topology of a contact-aided compliant mechanism [KSS18]. However, we are not aware of works considering a general methodology for the optimization of grippers and flexible manipulators leveraging self-contacts.

3.4.2 Contributions

In this section, we present two main contributions: the demonstration of an optimization framework for soft robot design with both self-contacts and external contacts, as well as the analysis of the potential of using self-contacts for soft robot design. The aim is not to design an advanced gripper but to work with a design space parameterization that enables highlighting

the contribution of self-contacts to the gripper's performance. This practical design is easily manufactured. Both the simulation and numerical optimization results are experimentally validated.

Additionally, an environment targeted toward soft gripper optimization for grasping tasks is introduced. Although a case study based on a particular parameterization is analyzed in this work, both simulation and grasping quality metrics generalize to other soft grippers. All the code is available as another part of the open source plugin *SoftRobots.DesignOptimization* for *SOFA*. It also shows how heuristic-based algorithms implemented in the plugin can be applied to the calibration of mechanical parameters of soft robots in the practical case of the soft grippers. It is especially relevant in cases with non-smooth events such as contact, in which gradient-based methods might struggle with highly non-convex fitness function landscapes. Finally, the flexibility of the proposed framework is highlighted in the design optimization of soft finger shapes for handling different objects.

3.4.3 Parameterization and Numerical Implementation

Parametric Design

The design topology is chosen to enable the creation of self-contacts for specific actuation values during optimization, while being simple enough to understand their effect on the finger performances, and simple to fabricate and actuate. The finger design is depicted in Figure 3.16. It consists of two legs attached to a top wall, one being fixed in space at its base and intended to come into contact with the object to grasp, and the other being actuated using a servo motor.

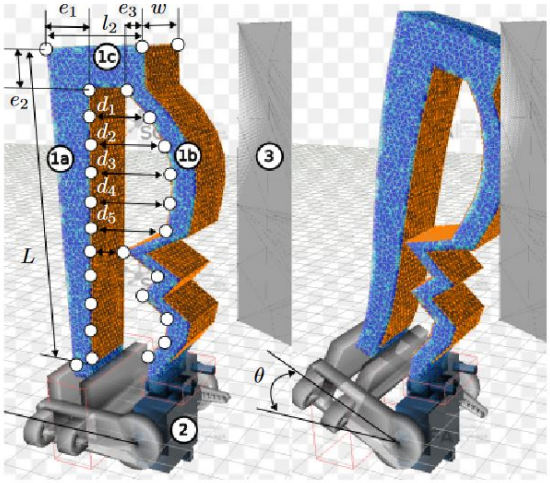

Figure 3.16: Illustration of the finger design parameterization. Finger composed of left (1a), top (1c), and right (1b) walls. (2) Servomotor. (3) Example of object to grasp, fixed in space. Left: rest configuration. Right: closed configuration.

The design parameters to optimize are the thickness of each leg and the top wall (e_1, e_2, e_3) and the distances $(d_1, ..., d_n)$ between the left leg and n intermediate points along the right leg. Varying these distance values enables the production of zigzag patterns on the right leg to create self-contacts and adapt to the shape of the object to grasp. Additionally, the finger shape is an extruded planar pattern and is therefore easy to fabricate using additive manufacturing techniques. The design parameters are constrained in intervals with fixed boundaries to prevent

interpenetration, limit the finger size, and respect the minimum thickness that can be obtained using the 3D printer considered in this work. The total finger thickness w, the finger width l_2, and the length L are fixed for simplicity.

Simulation

A quasi-static simulation of the soft finger is used to assess its performance. The mechanical behavior of the soft gripper deformation is described using continuum mechanics for which there are no analytical solutions in the general case, justifying the use of non-linear FEM for obtaining an approximate solution. Under the assumptions of both low robot acceleration and velocity, using the quasi-static equilibrium is enough for accurately modeling soft robot behavior. It is written as follows:

$$\begin{bmatrix} \mathsf{K}_{DD}(\mathbf{x}_i) & \mathsf{K}_{DR}(\mathbf{x}_i)J & 0 & \mathsf{H}_c^T \\ J^T\mathsf{K}_{RD}(\mathbf{x}_i) & J^T\mathsf{K}_{RR}(\mathbf{x}_i)J & \mathsf{H}_a^T & 0 \\ 0 & \mathsf{H}_a & 0 & 0 \\ \mathsf{H}_c & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x}_i \\ \delta_a \\ \lambda_a \\ \lambda_c \end{bmatrix} \begin{matrix} = \\ = \\ = \\ \perp \end{matrix} \begin{bmatrix} \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{x}_{i-1}) \\ J^T(\mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{x}_{i-1})) \\ \theta \\ \delta_c \end{bmatrix} \tag{3.5}$$

where $\Delta\mathbf{x}_i$ are small displacement of nodes around their current position $\mathbf{x}_i$, J is the Jacobian, K_{ij} are the tangent stiffness matrices depending on the current positions of the FEM nodes where D and R refer respectively to deformable and rigidified degrees of freedom. Rigidified nodes are the ones fixed to the servomotor. f_{ext} are gravity forces and $\mathbf{f}_{int}$ are nonlinear internal forces of the deformable structure computed from the material constitutive laws. Considering both the low thickness of the finger's leg and the low angle values achieved with the servo-motor, the deformation inside the material is assumed to be small and the material to be elastic linear. Its behavior is characterized by a Poisson ratio ν and a Young's Modulus E. Both actuation and contact constraints effects are integrated into the model as Lagrange multipliers, denoted respectively λ_a and λ_c, whereas the corresponding constraints displacements are δ_a and δ_c. They respectively depict the angular displacement of the servomotor and the distance between contact points computed with the model. The angle θ is the desired servomotor angular displacement specified by the user. Finally, the symbol $\perp$ is used for referring to complementary constraints such as contacts.

The servo motor is modeled using stiffness projection of the node elements of the soft finger parts attached to the actuation device. The degrees of freedom of the rigid part are then directly given by the angular displacement δ_a imposed on the servomotors.

Specific meshes of the soft finger and the manipulated object are provided for modeling collisions. Contact points and surfaces are detected at each time step using a collision detection algorithm. The constraint response is then computed using Signorini's law and friction is modeled using the Coulomb law with a friction coefficient μ. The formulation of the complementary problem using Signorini's law is as follows:

$$0 \leq \delta_c \perp \lambda_c \geq 0 \tag{3.6}$$

According to Coulomb's friction law illustrated in Figure 3.17, the reaction force lies within a cone defined by the normal force's magnitude and direction. The contact force is thus given by the composition of a normal force $\mathsf{H}_n^T\lambda_n$ and a tangential force $\mathsf{H}_t^T\lambda_t$ such that $\mathsf{H}_c^T\lambda_c = \mathsf{H}_n^T\lambda_n + \mathsf{H}_t^T\lambda_t$. When the reaction force is strictly within the cone, the objects stick together. If the reaction force reaches the cone's boundary, the objects slip along the tangential direction. In this slipping

scenario, the friction force aligns with the direction of motion. The three cases are summarized below:

$$\begin{cases} \lambda_n = \mathbf{0} \quad \text{and} \quad \delta_n \geq \mathbf{0} \quad \text{(inactive)} \\ \|\lambda_t\| < \mu\|\lambda_n\| \quad \text{and} \quad \delta_c = \mathbf{0} \quad \text{(stick)} \\ \|\lambda_t\| = \mu\|\lambda_n\| \quad \text{and} \quad \delta_n = \mathbf{0} \quad \text{(slip)} \end{cases} \tag{3.7}$$

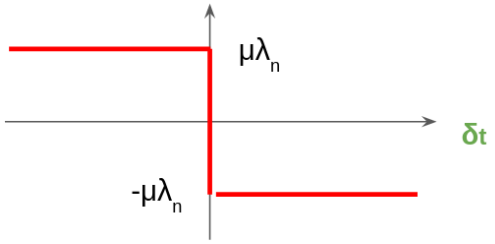

Figure 3.17: Illustration of Coulomb's friction law.

Solving the constraints is done in two steps by first solving a free configuration of the robot without any constraint:

$$\mathsf{K}(x)\Delta\mathbf{x}_i^{free} = \mathbf{f}_{ext} - \mathbf{f}_{int}(\mathbf{x}_{i-1}) \tag{3.8}$$

and then computing the new shape of the robot from δ_a. The motor torque λ_a is then computed.

Fitness functions for calibration

Calibration of mechanical parameters is a crucial step for ensuring that the simulation is reliable enough for the targeted applications. Due to the non-linearities and non-smooth events induced by self-contacts, stochastic algorithms are well-suited for exploring the space of mechanical parameters. Unlike gradient-based algorithms, stochastic methods are effective at avoiding local minima, making them more robust for the task at hand.

For this purpose, the general idea is to build a fitness function as a distance between sensor measurements on the physical prototypes and the same measurements obtained from the simulation. For the considered soft finger, the servomotor torque is measured and used for calibration. This gives the following fitness function:

$$F^{calib\_torque}(\mathbf{p}) = \sqrt{\frac{1}{N}\sum_{\theta_i}(\lambda_a(p,\theta_i) - \lambda_a^*(\theta_i))^2} \tag{3.9}$$

where $\mathbf{p}$ is a vector containing the optimized calibration parameters, θ_i are the sampled angular displacements imposed on the servomotor, N is the total number of measures, λ_a and λ_a^* are respectively the servomotor torques computed through simulation and measured on the physical prototype.

Fitness functions for design optimization

Several metrics have previously been used for assessing soft grippers and manipulators, considering whether object shape matching [DL21], grasping forces [Don+18], workspace [AZK21] or sensing [Nav+23]. When designing a soft gripper finger, the best compromise between deformability and ability to apply force has to be found. One objective is also to reduce the

necessary energy needed for both actions. Finally, actuation limits in terms of maximum force and torque are taken into account to avoid oversizing the actuation system. These compromises are expressed in the form of three fitness functions to be optimized for the design parameters $\mathbf{p} = (e_1, e_2, e_3, d_1, ..., d_n)$: F^{str}, F^{grasp_res} and F^{grasp_ener}.

In order to evaluate these fitness functions, a direct simulation where an angular displacement θ^{max} is imposed at the base of the finger is implemented. The angular displacement θ is gradually increased and the resulting torque is monitored at each simulation time step. The angular displacement samples are denoted θ_i. When the maximum torque is reached, the fitness functions are then computed. The expressions of these three functions are introduced below.

The first function to optimize is the maximum contact force applied by the soft finger on the object, denoted F^{str}. Only the component along the x-axis i.e. the direction from the finger towards the object, is considered. In simulation, this objective is computed as being the sum of the forces $\lambda^j_{c,x}$ exerted on each node of the finger mesh colliding with the nodes of the object at equilibrium:

$$F^{str}(\mathbf{p}, \theta^{max}) = \sum \lambda^j_{c,x} \tag{3.10}$$

The second function directly relates to the necessary energy for maintaining the object during the grasp. It is computed as a resistance F^{grasp_res} to be minimized. To encourage the creation of designs with an effective grasp at the smallest actuation cost, the minimum of the computed resistance for each angular displacement is kept. This gives:

$$F^{grasp\_res}(\mathbf{p}) = min_{\theta_i}(\frac{\lambda_a(\mathbf{p}, \theta_i)}{F^{str}(\mathbf{p}, \theta_i)}) \tag{3.11}$$

Finally, the F^{grasp_ener} is a metric proportional to the electric energy needed for grasping the object. The torque $\lambda_a(p, \theta_i)$ is measured for each angular step θ_i. For a specific angular displacement, the mechanical work is:

$$E(\mathbf{p}, \theta_i) = \lambda_a(\mathbf{p}, \theta_i)\theta_i \tag{3.12}$$

This mechanical work is directly proportional to the electric energy absorbed by the actuator by a constant transmission ratio. Under the hypothesis of constant torque in a sampled angular interval, the total electric energy F^{grasp_ener} absorbed during the grasping cycle can be represented by:

$$F^{grasp\_ener} = \sum_{\theta_i} \lambda_a(\mathbf{p}, \theta_i)\theta_i \tag{3.13}$$

Choice of Optimization algorithm

The design system we defined is a multi-objective problem with highly non-linear fitness functions regarding design parameters and without available analytical gradients. Stochastic algorithms are thus good candidates for solving this kind of problem. In our experience, Bayesian algorithms are usually better than Evolutionary algorithms at giving good solutions in a few optimization iterations as they privilege exploitation over exploration. Nevertheless, evolutionary algorithms are mainly used for generating the results in this work for their didactic purpose. Indeed, they enable exploring more in-depth what are good and bad results regarding the fitness metrics. Unless mentioned otherwise, they are use in all the following experiments.

3.4.4 Numerical Results

Design Optimization Results and Analysis

The considered scenario consists of grasping a vertical rigid object. For the optimization, a Young's Modulus of 3.62 MPa and a Poisson's Ratio of 0.452 are selected. They are obtained through torque calibration as latter described in Section 3.4.5. Moreover, strong hardware constraints come from the maximum torque exerted by the chosen servomotors. Preliminary experimental studies have shown it to be a limiting factor in the angular displacement of the finger when interacting with objects. A maximum torque of $\lambda^{max} = 1.2\,\mathrm{N\,m}$ is thus considered. A single friction coefficient is taken equal to 0.85 for all contacts. These hypotheses are further discussed in section 3.4.4. For a given set of design parameters, the three objectives are assessed using a 3D mesh with a uniform node density, resulting in a number of nodes varying from 2000 to 3000 with the design. A total of 40 intermediate angular positions θ_i are sampled in the range [0°,30°]. With these settings, evaluating a design takes between 30 s to 90 s with our setup [7], depending on the number of contact surfaces met during the grasping trajectory.

Both algorithm's hyper-parameters and results obtained for the three fitness functions introduced in section 3.4.3 are displayed in Figure 3.18.

For generating high contact forces, the best performing results, such as designs **B** and **C**, are the ones with thicker walls and self-contacts, leading to a higher rigidity when contacting the object. The respective designs **D** and **A** are examples of ineffective designs.
The relevance of both the grasping resistance and the grasping energy metrics is now highlighted. On the one hand, poor grasping resistance scores reflect designs where keeping the grasped position is expensive in terms of actuation effort. It is the case for design **A** where the absence of self-contacts does not enable to rigidify the finger enough. It is also the case for design **E** where self-contacts appear too close to the base of the soft finger structure, then directly opposing to the servomotor course. On the other hand, minimizing the grasping energy enables the exclusion of designs where self-contacts appear too soon when the soft finger is closing on an object, as it is the case for **E**. Finally, efficient designs also feature large contact surfaces shared between the soft finger and the considered object. Good compromises between all metrics are designs **B** and **C**.

Influence of Relative Object Pose and Shape

The potential of self-contacts to adapt to objects with different poses and shapes, as well as the modularity of the design optimization framework, are here demonstrated. Indeed, changing both these parameters has a strong impact on the optimization outcomes.

First, two different object-shape scenarios are considered: one with a hexagon-shaped object and one with a spherical object. The choice of these object shapes is to explore two different scenarios: one involving contacts with a plane, which is one of the simplest cases, and another one featuring contacts with a small convex surface. This last scenario aims to favor the emergence of designs that wrap around the object to develop more grasping forces. The approach is fairly general and can be extended to any other object shape. In order to favor the appearance of contact surfaces with a specific object shape, a higher value of 40° for the maximum angle command θ of the servomotor is considered. The objective is to privilege the emergence of more self-contacts. This angle is illustrated in Figure 3.16. Design optimization results are displayed in Figure 3.19.

Compared with the results in Figure 3.18 for the baseline finger design **A**, the considered actuation angle leads to self-contacts during grasping. However, this design still gives poor

[7] Laptop with eight cores 2.70 GHz Intel Core i7 - 6820

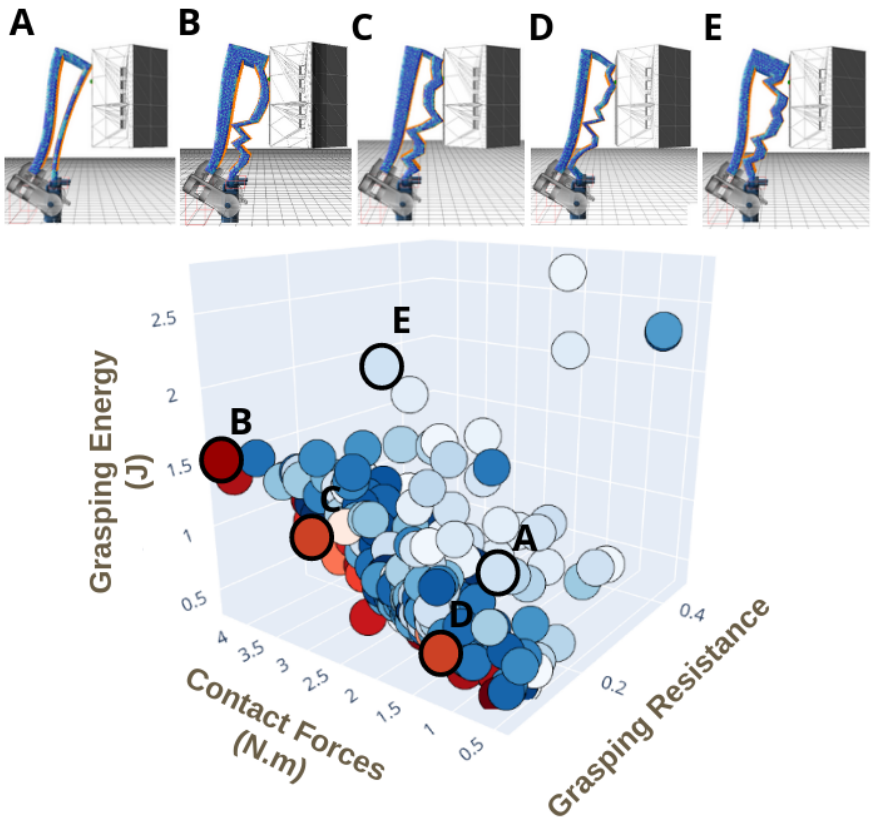

Figure 3.18: Pareto Front and a few sampled geometries at the grasping pose obtained for 300 trials of the Soft Finger design optimization. Pareto optimal solutions are represented by red dots. Results are generated using the NSGA-II algorithm with an initial population of 50 design candidates, probabilities of crossover, and swapping parameters between parents of 0.9 and 0.5, respectively.

performances for both object shapes, due to the finger slenderness and the proximity of the contact to the base.

In the hexagon-shaped scenario, because of the object layout, there is little room for generating large contact areas shared between the soft finger and the object unlike in the previous experiment. Only one or two contact lines along the width of the finger are possible at most. This is why the best-performing designs feature double closed kinematic chains for transmitting more contact forces, as shown in both designs **B** and **C**.

For the spherical object, Pareto designs such as designs **D** and **E** both comply with the shape of the object and feature an internal closed chain far away from the base of the finger and close to the object for an enhanced stiffening. However, **E** is also an example of a design where a second internal closed chain appears too close to the finger base and opposes its course: because of the maximum servomotor torque, only a maximum angular displacement of 35° is reached.

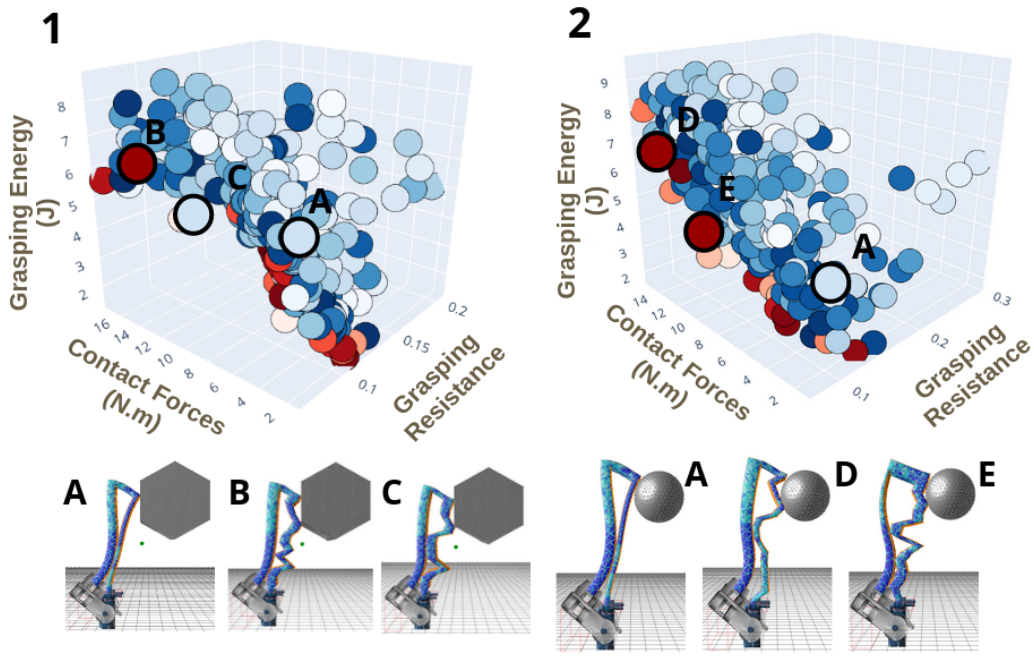

Figure 3.19: Pareto Front and a few sampled geometries at the grasping pose obtained for 300 trials of the Soft Finger design optimization for different objects scenarios. Scenarios are 1) a hexagon-shaped object and 2) a spherical object. Pareto optimal solutions are represented by red dots. Results are generated using the NSGA-II algorithm with an initial population of 50 design candidates, probabilities of crossover, and swapping parameters between parents of respectively 0.9 and 0.5.

Influence of Friction

The impact of the chosen friction coefficients μ for both soft finger self-contacts and contacts with the object is now discussed. Numerical values of the fitness functions for different designs and friction coefficients are displayed in Table 3.6.

Design	μ	F^{str}	F^{grasp_res}	F^{grasp_ener}
B from Figure3.18	0.85	4.32	0.075	1.49
B from Figure3.18	0.30	3.09	0.086	1.40
E from Figure3.19	0.85	11.08	0.090	4.32
E from Figure3.19	0.30	8.97	0.094	7.14

Table 3.6: Fitness functions values for a two soft finger design and different friction coefficient μ. Although the magnitudes of the different fitness functions can differ by a factor of 100, using a Pareto front to explore the design space prevents fitness functions with higher values from dominating the optimization process.

Decreasing the friction coefficient results in reducing the generated contact force and increasing the grasping energy as illustrated in the cases of both designs Figure3.18.B and Figure3.19.E. This can be explained by the fact that generated forces are lost in sliding motion. However,

some designs may take advantage of an increased sliding motion to generate additional contact surfaces. This is particularly the case in manipulation tasks considering small objects such as the sphere. Without necessarily decreasing the contact force, considering altering the μ coefficient and maximizing this contact surface with an additional fitness function would ensure more stable grasping. In practice, these coefficients may be easily altered on the physical prototype by installing pads of different materials located at the expected contact areas on the soft finger. It is interesting to notice that the optimization tools are able to find solutions that go against normal intuition: one would think that increasing friction is always beneficial for stable grasping, as suggested by Table. 3.6, but we show here that this is not always true as it impacts the number of contact surfaces.

3.4.5 Experimental Study

Experimental Setup

The experimental test bench used to characterize the finger design is represented in Figure 3.20. This setup incorporates a Herkulex DRS-0101 servomotor (DFRobot) with a stall torque of 1.18 N m and a resolution of 0.321° to actuate the finger. The actuation torque is evaluated using a strain gauge placed on the servo motor arm and by multiplying the measured torque by the arm length. The gauge is capable of measuring forces up to 1 kg with a measurement accuracy of ±0.02%, connected to a high-precision 24-bit hx711 analog-to-digital converter. As a result, torques up to 0.36 N m can be measured with an accuracy of $\pm 7.2\times10^{-3}$ N m. The servo motor control and torque estimation are performed on an Arduino Mega acquisition board connected to a PC. This test bench incorporates also a Velleman precision balance enabling precise readings of the contact force with a precision of 9.81 mN.

The rest of the components making up the test bench were produced on a Prusa MK3 3D printer using PLA as the material. All the fingers tested were 3D printed (Prusa mk3) using FilaFlex flexible filament with a shore hardness of 60A and a stretch capacity of around 950%.

As the ultimate aim of the test bench was to check the gripping capacities of the various finger designs, a complete gripper was developed, shown in Figure 3.21. This gripper uses the same servomotor-finger assembly concept as in Figure 3.20, which is repeated 2 times with a 120° offset. Its modular design makes it easy to change the design of the fingers using the various fixing and locking screws and to adjust their distance from the gripper center. Different distance values are indicated using 3D-printed graduations for accurate positioning. In order to carry out grasping operations, this gripper was assembled on the end-effector of a UR3 robot arm (Universal Robot). The integration of the gripper enables various 'pick and place' processes to be carried out with objects of different sizes, shapes, and weights.

Simulation Calibration

We demonstrate now how to use the toolbox for calibrating the model of a soft robotic system involving contacts. In this work, the mechanical properties of the material composing the finger, i.e. E and ν, as well as the distance from the finger d_o to the object and the torque offset τ_0 must be estimated. The Young's modulus is computed from the shore hardness given by the filament manufacturer using the relation in [Gen58], giving, $E = 3.62$ MPa. The Poisson ratio and the two other parameters are obtained through optimization, using the fitness function (3.9). The Poisson ratio is typically close to 0.5 for rubber material, and small changes around these values can lead to large changes in the simulated robot behavior. Therefore, the algorithm is forced to explore values close to 0.5 by optimizing a modified Poisson ratio ν_o such as:

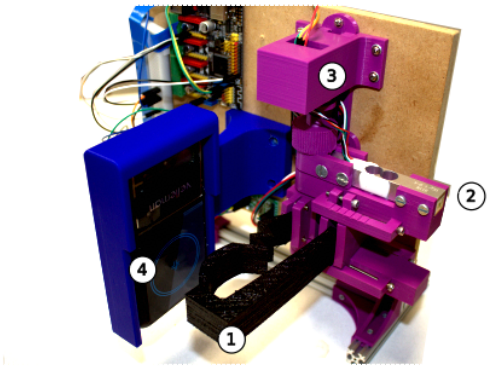

Figure 3.20: Experimental setup for the finger design (1). Test bench consisting of a strain gauge (2), a Herkulex drs-0101 servomotor (3) and a Velleman precision balance (4).

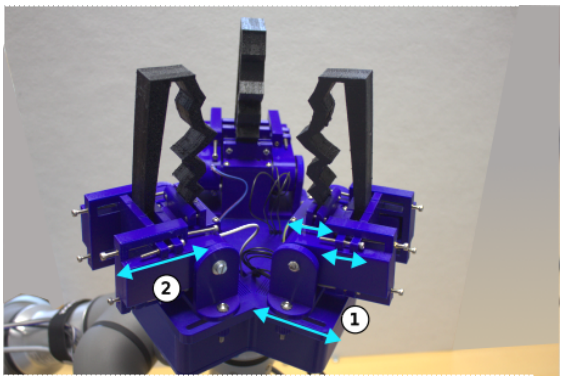

Figure 3.21: Prototype of gripper composed of three fingers. Screws and graduation are used to adjust the distance between the finger and the gripper center (1) and the finger width (2). The gripper is assembled on the UR3 collaborative robot in a pick-and-place situation.

$$\begin{aligned} \nu &= (\nu_\infty - \nu_m)(1 - e^{-\frac{\nu_o}{\tau}}) + \nu_m \\ \tau &= -1/\ln\left(1 - \frac{\nu_M - \nu_m}{\nu_\infty - \nu_m}\right) \end{aligned} \tag{3.14}$$

where $[\nu_m, \nu_M, \nu_\infty] = [0.4, 0.499, 0.501]$ are the values of ν for $\nu_o = [0, 1, \infty]$.

Experimental data was acquired on both the baseline and finger design **C** from Figure3.18. On the test bench, the servomotor's angle was increased and then decreased in the range $[0°, 40°]$ with steps of $1°$. Each experiment was repeated 3 times, leading to point clouds with hysteresis patterns as shown in Fig 3.22. The centerline of these two point clouds, one for each finger, was extracted using a recursive barycentric method and evaluated at 8 angular displacement values in $[0°, 30°]$, giving the experimental torque values $\lambda_a^\star$.

To reach the function minimum as fast as possible, the Tree-Structured Parzen Estimator Bayesian algorithm was used. After 250 iterations, the cost function reaches an acceptable value of $4.3{\times}10^{-3}$ N m with the parameters $(\nu, d_o, \tau_0) = (0.451, 35.3\,\text{mm}, -3.2 \times 10^{-3}\,\text{N m})$ for the base

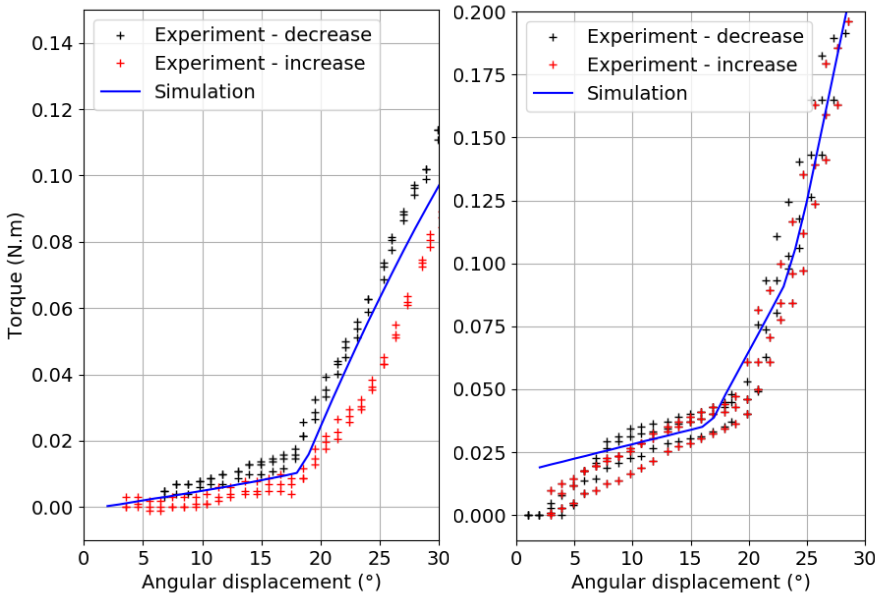

Figure 3.22: Measured and simulated relations between angular displacement and torque for the base finger (left) and the C finger (right).

finger design, and $(0.451, 30.3\,\text{mm}, 1.5\times10^{-2}\,\text{N m})$ for the C finger design. The obtained value of the Poisson ratio is consistent with the usual values for elastomer materials. The relatively high torque offset observed during the finger C calibration is due to the amplifier of the hx711, whose offset was observed to increase with its temperature.

Moreover, the distance to the object varies reasonably around the nominal value of 30 mm, considering the possible assembly and manufacturing defaults on the test bench and the 3D printing errors on the finger. This validates the ability of the model to predict the finger's behavior both when colliding with the object and when self-contact occurs.

Performance Validation

To validate the performances of the finger designs, the fitness functions values obtained experimentally and numerically are compared. For the comparison, the numerical values of the performance indicators are obtained in simulation with the calibrated parameters determined in section 3.4.3 for the baseline design. The results are presented in Table 3.7. There is a good agreement between the simulated values of grasping energy and the reality, as they depend on the calculated torques that were calibrated. There is a significant difference in contact force for the two designs with self-contacts, leading to discrepancies in the values of grasping resistance as well. After investigation, several sources of error were detected. There are variations of the effective finger length and the relative finger-to-object distance due to manual assembly, leading self-contacts to happen for higher angular displacements and decreasing the final contact force. Also, simulation parameters such as the friction coefficient μ and the density of the surface contact meshes were not optimized. A lower value of μ and a denser mesh significantly reduce the simulated contact force, decreasing further the gap of measured contact forces between

simulation and reality. Furthermore, the assumption that both internal and object contacts share the same friction coefficient creates errors as well. However, despite this difference in contact force, finger designs **A**, **C**, and **B** in this order have increasing contact forces and decreasing grasping energy as expected from the Pareto front in Figure 3.18. In conclusion, the experiments confirm that the optimization process yields correct sim-to-real behavior in terms of the relative performance of the designs. However, the current modeling assumptions do not allow for quantitatively accurate results.

Design	F^{str}		F^{grasp_res}		F^{grasp_ener}	
	Sim.	Exp.	Sim.	Exp.	Sim.	Exp.
A	0.65	0.63	0.154	0.155	0.296	0.308
B	4.32	0.87	0.075	0.320	1.49	1.247
C	1.88	0.69	0.142	0.430	0.654	0.769

Table 3.7: Fitness functions values for the three soft finger designs obtained experimentally (Exp.) and in simulation (Sim.).

Grasping Demonstration

Designs are demonstrated on a pick-and-place task corresponding to grasping a weighted cylinder and moving it to a given location. In order to have friction coefficients consistent with those used in simulation, tabs of the same material as the fingers are glued to the cylinder in the places where it is gripped. The final angle command imposed to all servomotors is 37°. The video of the grasping experiments is available here [Nava]. The gripper is tested with the designs **A**, **B** and **C** from Figure3.18, and manages to lift respectively 150g, 650g, and 750g. The fact that the gripper performs better with design **C** rather than design **B** is explained by the considered greater angle control than the one considered during the optimization. The results obtained validate the superior performances of the optimized designs with self-contacts. Pictures of the gripper design **C** taken at different times during the grasping scenario are provided in Figure 3.23.

3.4.6 Conclusion

This work presents the computational design and calibration of a parametric soft finger with contact modeling. We show that the most performing designs make use of self-contacts, both for improving grasping metrics and for adapting to the object shape. The optimization results are assessed experimentally, validating the Pareto front given by the FEM-based design framework.

Several perspectives are considered to push this work further. The considered design parameterization leads to sharp line-shaped contact areas. This strengthens the influence of friction at both the Finger-Finger and Finger-Objects interfaces. In future iterations, we plan to consider building smooth elbow-shaped surfaces between the control points instead of lines, by using Bezier curves for instance.

Also, the design optimization of a soft finger shape is done for a single specific object shape. In future work, we plan to extend the introduced tools for designing a more universal soft gripper, by considering the evaluation of fitness functions averaged on a dataset of different objects. Moreover, other fitness functions may be considered for accounting whether for manufacturing constraints such as minimizing the material volume needed for building the finger, or functional constraints such as the grasping stability i.e. maximizing the shared surface area between the soft finger and object.

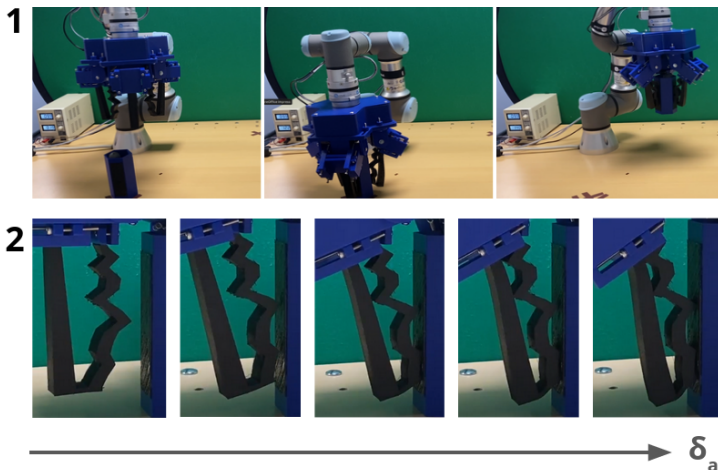

Figure 3.23: 1) Gripper design **C** at different moments of the considered grasping scenario, respectively at the initial position, once the object is grasped, and during object displacement. 2) Gripper design **C** during the grasping phase for different angular displacements.

3.5 Discussion and Conclusion

In this Chapter, a framework targeted at the computational design optimization of parametric soft robot designs is introduced. This framework is fairly general. Indeed, the chosen stochastic solvers are not required to compute analytical gradients of design objectives relative to the design variables, meaning the method is compatible with any modeling choice for performance assessment of soft robot designs. In particular, we coupled it with FEM simulation which enables the simulation of a wide range of soft robots without prior assumptions about their shape and which results transfer well to soft robots in the physical world. We can differentiate between scenarios where there is no contact, resulting in smooth transfer, and scenarios with frictional contacts, where the transfer is qualitatively correct but not quantitatively accurate.

Both a soft sensorized finger and contact-aided soft gripper design spaces have been explored using the aforementioned framework. In each case, the proposed framework enables exploring design performances relative to user-chosen metrics. They feature how good design choices, such as optimized embedded sensor geometry or structures making use of self-contacts, facilitate task-specific effective control and sensing. We also show that the numerical results transfer well to physical prototypes in terms of relative performances obtained numerically.

However, the proposed framework has some disadvantages. First, the used stochastic solvers do not scale well on large problems with many design variables and are limited to continuous variables. Moreover, the proposed framework relies on many simulations for soft robot assessment that may be computationally expensive. In our experiments, design spaces of at most 20 design parameters and simulation evaluation taking between 30 seconds to 1 minute were considered. Simulation computation time constraints induce considering relatively simple control strategies during a design evaluation. The designer then needs to break down the task they want their robot to perform into a set of easy-to-evaluate performance metrics with low-level controllers.

With the desire to scale to more complex models and control strategy for design assessment, we develop a reduced model based on the FEM in the next Chapter. The objective is to keep a precision close to that of the FEM while drastically reducing the simulation time. Compared to this

framework, the idea is not to use the simulation as a black box anymore, but to extract relevant information to be used for control, design and interaction with the environment dynamically.

Finally, the considered design spaces are still heavily reliant on the user intuition. We also considered moving to more expressive design spaces, using modular components or metamaterials. Preliminary works about this matter are further discussed in Chapter 5.

Chapter 4

Condensed FEM Modeling for Embedded Control and Design Optimization

4.1 Introduction and Chapter Overview

Although the FEM is a powerful tool for modeling and predicting the behavior of soft robots, its computation time can be a significant limitation in robotics applications. Using FEM in control applications poses challenges for those not specialized in numerical computation, as achieving real-time performance requires optimization of computation time. Additionally, design optimization can benefit from simulations that run faster than real-time, facilitating quicker design space exploration.

This chapter introduces a learning-based approach to obtain a compact but sufficiently rich mechanical representation for both control and design optimization applications. Our choice is based on non- linear compliance data in the space of robot constraints (actuator, effector and contact) provided by a condensation of the FEM model. The proposed method is able to handle all kinds of constraints and in particular contacts. We demonstrate that this compact model can be learned with a reasonable amount of data and remain very efficient in terms of modeling, since we can deduce the direct and inverse kinematics of the robot. We envision this condensed FEM modeling as a general framework for modeling, control, and design of soft robots. This work have been jointly developed with Etienne Menager and presented in two articles [Mén+23], one being under review.

First, the learned condensed FEM model is contextualized within the model reduction bibliography and the proposed mechanical representation is described in section 4.2. Then several applications leveraging the proposed compact mechanical representation of a soft robot are described. These applications are summarized in Figure 4.1.

Section 4.3 highlights the use of the condensed FEM model in control scenarios, demonstrating the derivation of both direct and inverse kinematics. Rather than directly learning a direct or inverse kinematic link between effectors movement and actuators movement, a mechanical representation is learned. This approach enables the development of various control schemes from a single learned model, as illustrated in two manipulation scenarios with a soft gripper.

Additionally, this section emphasizes the modularity of the proposed framework by showcasing the control of a complete gripper in a manipulation task based on the condensed FEM model learned from a single soft finger.

In section 4.4, both the low-memory consumption and the high prediction speed of the learned condensed model are leveraged for real-time embedding control without relying on expensive online FEM simulation .

In section 4.5, it is demonstrated how both the expressivity and the differentiability of the learned condensed model relative to soft robot design variations are used in calibration and design optimization applications.

Finally, section 4.6 discuss the advantages and limitations of the proposed framework.

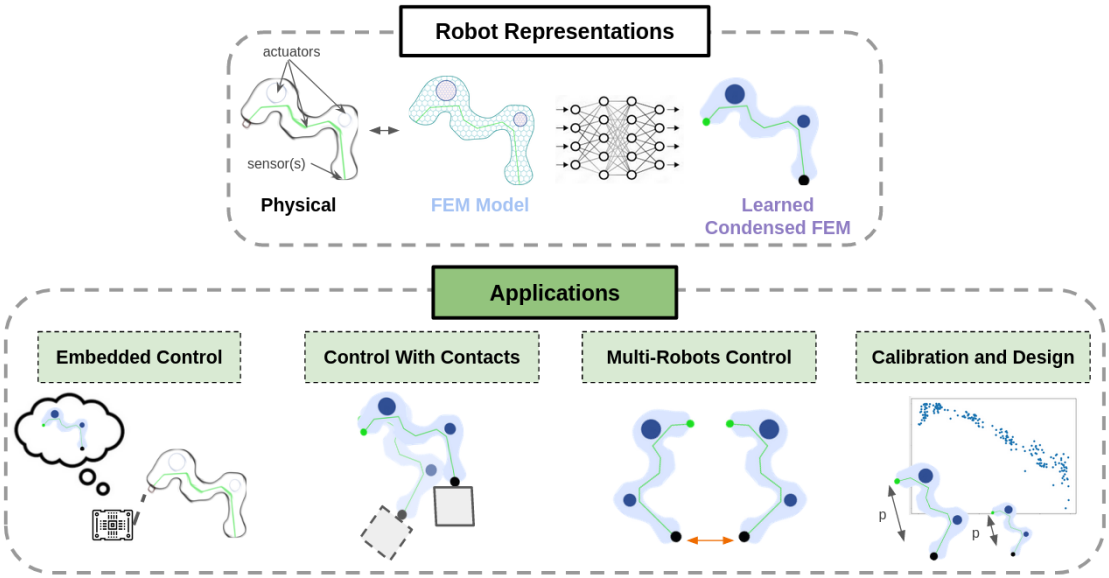

Figure 4.1: Illustration of the proposed framework and its applications. A physical soft robot is modeled using the FEM. The condensed modeling is obtained by projecting the FEM characteristic matrices in the constraint space. A neural network is trained offline for predicting the condensed FEM modeling from both the initial robot condensed state and applied actuation. For a given robot, a single learned condensed FEM modeling may be used in several applications i.e. for real-time embedded control, for inverse control involving a set of predefined contact points, for controlling several coupled similar robots altogether as well as for calibration and design optimization.

4.2 Learned Condensed FEM Modeling

In this section, the learned condensed FEM model is introduced. After contextualizing the chosen mechanical representation, we justify the choice of the model to be learned and explain how it is built and applied in control applications.

4.2.1 Model Order Reduction based on Soft Robot Simulation for Embedded Control and Design

As described in section 2.1.5 of the state of the art, model order reduction techniques aim to lower the computational complexity of computationally expensive and large-dimension mathematical models in numerical simulations. In this work, we target a new method for building a reduced model of soft robots. This model should capture the inverse and direct kinematics of the robot under quasi-static assumptions, and accounts for contact configurations. In addition, this reduced model has to be sufficiently small to be embedded in a micro-controller. It should also be sufficiently generic to capture design variations for calibration and design optimization applications.

Modeling and Control

Models chosen for describing the behaviour of soft robots influence the way they are controlled. For example, thanks to their small dimension, beam models can be used for quick control loops [LXZ23] or predictive control [SK22]. However, the 1D aspect of the underlying model does not enable considering complex robot geometries. Other highly simplified models, such as the voxel-based mass-spring model used in EvolutionGym [Bha+22], enable to easily explore the resolution of complex tasks thanks to their low computational cost, but they do not transfer well to physical robots. Several works propose to use neural networks to learn robot model from experimental data [Thu+17; Gil+18; Ber+20]. However, they can require large amounts of training data to be accurate, which can limit their use in situations where data is either rare or expensive to obtain. Learned models have been applied in low-level adaptive controllers [Thu+17; Thu+18], but their capabilities are limited to the training data sets. Finally, models based on continuum mechanics can be used to build controllers based on fine control of robot deformation. For example, the FEM model can be condensed through projection in a smaller space [Dur13a]. However, this still requires to compute the entire FEM model for computing the condensation. The linearization of the command and the size of the involved mathematical systems reduce the control time horizon and imposes low-level control.

Reduced Order Modeling for Control

Using reduced models is a common approach in soft robotics to simplify the modeling and control of these complex systems. They are obtained using different methods, including projection, simplification, and learning. In [GD18], Proper Orthogonal Decomposition is applied to reconstruct the state of a robot based on deformation modes. The general idea is to reduce the number of degrees of freedom in the simulation, keeping only the modes that faithfully simulate the robot. This method requires an offline learning phase and is coupled with additional control tools from [CED17] for real-time control of soft robots. However, leveraging this reduced model for embedded control applications is still too computationally intensive. Simplification of the robot's mechanical behaviour can be achieved by using beam models such as Euler-Bernoulli or Cosserat to represent the behaviour of more complex robots in a smaller

resolution space [MPD23]. As the model is simplified, it only represents the robot's behaviour under a given set of assumptions. Learning methods enable the estimation of reduced models from experimental data. Various learning methods can be used. For example, data-driven approach has been applied to learn a differentiable inverse model of a soft robot's quasi-static physics linking actuation to effector positions [Ber+20]. However, since this direct link depends on the specific task at hand, the learned reduced model is only applicable to the conditions under which the data was collected. Each scenario requires specific training. In addition, learned models may not be able to capture the transient dynamic effects of the system, which may affect the performance of the associated controller. Other learning based methods facilitate the implementation of multi-time-step control. For example, a spectral sub-manifold can be used to learn the robot's dynamics around an equilibrium point [Alo+22]. The method has to rely on approximate dynamics and applications are limited to robots with equilibrium points and no contact points. Using reduced models significantly impacts system controllability. Simplifying the model reduces the complexity of the controller, making its design more straightforward and efficient. In addition, reduced models can enable faster and more efficient simulation, which can be useful for the design and optimization of reactive controllers [Thu+17; LXZ23]. However, using reduced models can also pose challenges for controllability. By reducing the number of degrees of freedom of the system, it is possible to lose important information on system behaviour. This can affect model accuracy and controller performance. For example, in the proxies method [MPD23], a simplified model has to be combined with an FEM-based control loop to correct the errors of the beam-based model. In addition, reduced models may not be able to capture non-linear effects and complex interactions with the environment, which may limit their use in more complex situations. This is the case, for example, with models learned for one set of contacts [GD18], which are difficult to generalize to a different set of contacts.

Reduced Order Modeling for Design

A trend towards using surrogates to facilitate soft robot design exploration is starting to emerge. These models are approximate representations of system behaviour under different working conditions. Surrogate models have been applied to optimize robot design according to specific performance criteria. In [YHM24], a surrogate model is learned to predict angular deflection of pneumatic modules as a function of mechanical and geometric parameters. However, in this approach, changing the cost function requires generating new data and relearning the surrogate model. In [ZSZ23], an FEM-based differentiable surrogate model is used to perform joint optimization of the design and trajectory of an auxetic robot. However, this model seeks to predict the behaviour of each lattice node and is limited to cylindrical geometries, which restricts its applicability to other design parameters. Finally, a soft crawler robot legs placement has been optimized to maximize its locomotion capacity [SSW22]. It relies on pre-learned POD-based reduced models of the legs as presented in [GD18] to speed up the simulation. This application is particularly compelling as it co-optimizes both the morphology and control of the soft robot, enabling the emergence of Embodied Intelligence.

However, not all reduced models are suitable for design optimization. For example, the POD-based model presented in [GD18] is a mesh and design-dependent model, meaning it cannot be explicitly linked to geometric design parameters. In the above-mentioned optimization case of the soft crawler robot [SSW22], it does not enable to consider variations of the legs geometry. Moreover, other learning-based models presented in [Thu+17; Alo+22] do not depend explicitly on the design of the soft-robots.

4.2.2 Mechanical Representation Learning

In this work, a high-dimensional state of the robot is available through simulation. For reducing the dimension of this state, the global compliance matrix projected in the constraint space W is a good candidate because it is a small dimensional matrix, independent of the size of the FEM mesh and it expresses a direct relationship between actuator and effector constraint spaces. This matrix is obtained through condensation of the FEM model in the space of constraint, as described in this manuscript state of the art in sections 2.1.4 and 2.2.1. We recall that W is defined as $\mathsf{W}_{ij} = \mathsf{H}_i \mathsf{A}^{-1}(\mathbf{q}, \dot{\mathbf{q}}) \mathsf{H}_j^T$ for $i, j \in \{a, c, e\}$, q the degrees of freedom and H_i^T for $i \in \{a, c, c\}$ are the first derivatives of the constraints. When the model is dynamic, A is analogous to an impedance matrix, so A^{-1} represents an admittance. In contrast, when the model is quasi-static, A is analogous to a stiffness matrix, making A^{-1} represents compliance. This suggests that the condensed model could be applicable in dynamic contexts. However, for this study, we have primarily focused on quasi-static conditions, which is why we emphasize compliance. The mechanical matrices W_{ij} and δ_j for $i, j \in \{a, c, e\}$, respectively the compliance and characteristic distances both projected in the constraint space, can be seen as a reduced state of a soft robot. It has been shown that both direct and inverse models of soft robot can be derived, as long as an approximation of the δ_j for $j \in \{a, c, e\}$ is available. Combining W with the free displacement vector δ^{free} enable to obtain this approximation using the following equations:

$$\begin{aligned} \delta_a &= \mathsf{W}_{aa}\lambda_a + \mathsf{W}_{ac}\lambda_c + \delta_a^{free} \\ \delta_c &= \mathsf{W}_{ca}\lambda_a + \mathsf{W}_{cc}\lambda_c + \delta_c^{free} \\ \delta_e &= \mathsf{W}_{ea}\lambda_a + \mathsf{W}_{ec}\lambda_c + \delta_e^{free} \end{aligned} \tag{4.1}$$

In the FEM framework, the computation of W is expensive because it involves the multiplication and inversion of several large matrices and this needs to be computed at every simulation step. This is an incentive for learning it rather than computing it from mechanical simulation. Learning these mechanical quantities rather than non-supervised ones makes it possible to obtain interpretative mechanical robot representations useful for simulating, controlling, and even designing the robot. The method is illustrated in Figure 4.2.

For building the condensed FEM model, we consider the following hypotheses:

- The robot's behaviour is described using the quasi-static mechanical equations. This quasi-static assumption holds true for robots that either move slowly or have negligible mass. If this assumption no longer holds, dynamic effects will emerge that are not taken into account by the model. This assumption is valid for many robots (catheters, trunks, etc.) [Coe19; Kar+20]. For other applications, such as high-frequency pick-and-place [ZHD21] or locomotors that are not attached to a fixed base [Coe19], dynamic effects often have to be taken into account. In this work we consider fixed-base robots with low speeds.

- The location of the contact points on the robot is known in advance, even though these points are not necessarily active throughout the simulation. This assumption means that contacts only exist at these locations. In more complex contact scenarios, such as dexterous manipulation [Coe19] or catheter locomotion for minimally invasive surgery [Kar+20], the position and number of contact points is not known in advance.

- Only actuation and contact forces modify the state of the robot. This means that even if W directly depends on the global FEM state, the knowledge of both actuators and contacts states is enough to evaluate W in the quasi-static case.

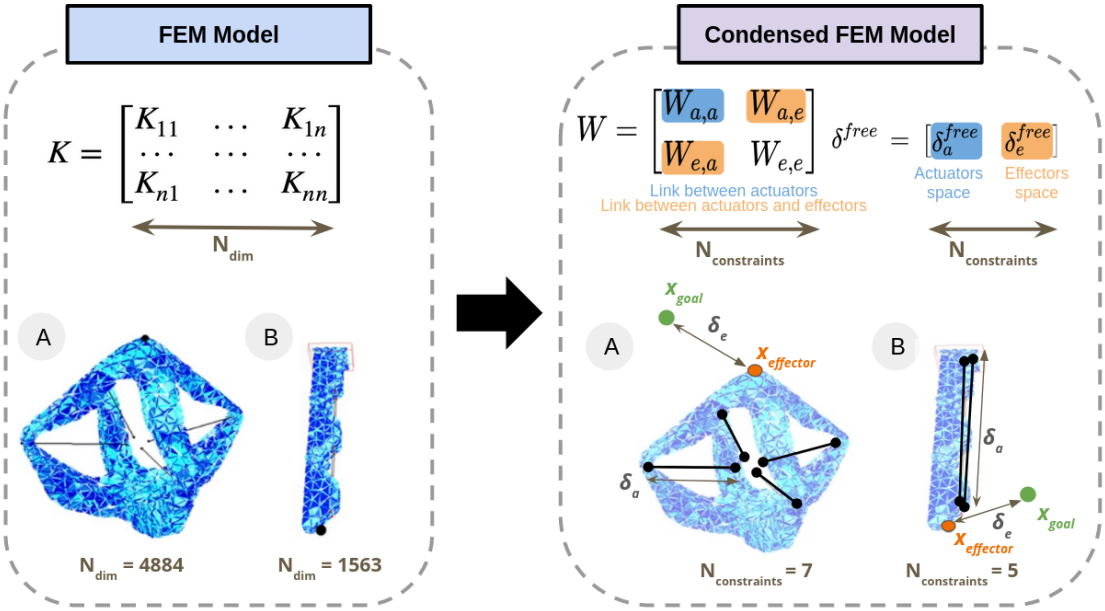

Figure 4.2: Illustration of the condensed FEM model for both the Diamond parallel and Soft Finger robots. They are respectively actuated by 4 and 2 cables. In both cases, a FEM model of the robot (left) is used to learn a condensed model (right). The reduced model is based on the projection of the compliance matrix into the constraint space, resulting in drastically reducing the number of characteristic dimensions describing a robot state.

- As we are not using highly non-linear material models or considering non-linear phenomenons such as buckling, there are no cases where the robot can reach a similar configuration for a same actuation.

- The data space is dense enough to represent the quasi-static state of the robot in many configurations. This assumption leads to significant sampling, which can be difficult to achieve when there are many different actuators or contacts.

The learning problem can be formulated in the following way. Let δ_a and δ_c be the effective states reached by both actuation and contact constraints when applying a specific active constraint state, W and δ^{free} be the corresponding compliance matrix and free displacement of the robot projected in the constraint space. The goal is to reconstruct $(\mathsf{W}, \delta^{\text{free}})$ from initial values $(\mathsf{W}_0, \delta_0^{\text{free}})$ for efforts $\lambda_a = 0$, $\lambda_c = 0$ and current active constraint state δ_a, δ_c. This leads to learn a function F such that:

$$\widetilde{\mathsf{W}}, \widetilde{\delta}^{\text{free}} = F(\delta_a, \delta_c, \mathsf{W}_0, \delta_0^{\text{free}}) \tag{4.2}$$

W_0 is a good compliance initialization since it already captures the mechanical coupling between constraints and the compliance matrix varies smoothly from that initial position in the robot workspace.

4.2.3 Learning Model

The compliance matrix W is a symmetric matrix that represents the compliance in the space of defined constraints: actuator, effector and contact. It is formed by sub mechanical matrices as follows:

$$\mathsf{W} = \begin{bmatrix} \mathsf{W}_{ee} & \mathsf{W}_{ea} & \mathsf{W}_{ec} \\ \mathsf{W}_{ae} & \mathsf{W}_{aa} & \mathsf{W}_{ac} \\ \mathsf{W}_{ce} & \mathsf{W}_{ca} & \mathsf{W}_{cc} \end{bmatrix} \tag{4.3}$$

The magnitude of both the values of W and δ^{free} depends on both the type of constraints and the measurement units chosen for the simulation. To avoid vanishing gradients during the learning process, both the input and output data are element-wise standardized. Standardization involves transforming each feature in the dataset so that it has a mean of 0 and a standard deviation of 1. This is done by subtracting the mean of the feature and then dividing by its standard deviation, computed across the entire training set. This standardization is computed once using all the training set.

As the matrix is symmetric, the networks are trained in a supervised manner by minimizing the reconstruction loss:

$$L = \sum_{\delta_a, \delta_c} [\|\widetilde{\mathsf{W}}^{\text{tri}} - \mathsf{W}^{\text{tri}}\|^2 + \|\widetilde{\delta}^{\text{free}} - \delta^{\text{free}}\|^2] \tag{4.4}$$

where W^{tri} is the upper triangular part of the matrix W, $\widetilde{\mathsf{W}}^{\text{tri}}$ and $\widetilde{\delta}^{\text{free}}$ are the predicted values of W^{tri} and δ^{free} as given by the network for actuation displacement δ_a and contact displacement δ_c. As all data are standardized, the computed loss functions obtained on different training data are somewhat comparable.

Well-known neural network architectures such as the MultiLayer Perceptron (MLP) are sufficient for learning a mechanical representation for control and design tasks. This kind of network is chosen because it is one of the simplest one. As it has few weights, it is very quick to train and produce results. In our implementation, the input is the raw data vectorized and concatenated to form a single vector and the output is the vectorized couple $(\widetilde{\mathsf{W}}^{\text{tri}}, \widetilde{\delta}^{\text{free}})$. Rectified Linear Unit (ReLu) activation functions are used.

As the framework is a supervised learning framework, pre-computation steps where data are collected and the learning model is trained are first performed. The data $(\delta_a, \delta_c, \mathsf{W}^{\text{tri}}, \delta^{\text{free}})$ are obtained through simulation and stored in a database. Each data point is associated with a robot configuration after applying an active constraint displacement (δ_a, δ_c). The range of possible values of both δ_a and δ_c are bounded in continuous intervals. Sampling the active constraint space is done using Scrambled Halton sampling algorithm [MC04] for building the training dataset and random search for the test dataset. Compared to homogeneous grid search strategies, it enables a better filling of a high dimensional sampling space and therefore a better scalability to robot having numerous actuators and contacts. For example, while 15000 configurations sampled with a grid search strategy were used for the Soft Finger robot without contact, we reduce the number of needed samples using the SH sampling algorithm to 6500 (see section 4.3). Reducing the dataset by 60% does not compromise the accuracy of the resulting matrices. Moreover, the number of sampled configurations is less or equal to those required for other methods. For example, the POD-based model presented in [GD18] requires 10000 samples to create a reduced model of a cable-driven parallel robot, where the learned condensed FEM model rely on only 6561 samples for the same robot with a grid search strategy. We do not consider optimizing the number of samples in this work, but we highlight that there is potential

for improvement in this area.

The number of points of the test set is chosen to amount to 25% of the number of points of the training set. Once the data is acquired, learning consists in training the network to predict the quantities $(\mathsf{W}^{\text{tri}}, \delta^{\text{free}})$ from the quantities $(\delta_a, \delta_c, \mathsf{W}_0^{\text{tri}}, \delta_0^{\text{free}})$ using MLP predictions and Equation 4.4. Unless mentioned otherwise, the used network in the following experiments of this chapter is a fully connected MLP with 3 hidden layers of 450 nodes each. The learning rate is initialized to 10^{-3} and incrementally decreased with an adaptive scheduler.

For all the learning experiments, training stops after a maximum of 50000 epochs. We retain the model from the epoch that achieves the lowest loss on the test set. The initial learning rate is set to 10^{-4}. An adaptive scheduler is consistently used in all experiments to automatically reduce the learning rate when the optimizer encounters a local optimum. We employ a discrete hyperparameter search to determine the best set of hyperparameters. Various combinations are tested, and models are compared using our training strategy on the same dataset. The hyperparameters explored include the number of hidden layers, which ranges from 3 to 5, and the number of nodes per hidden layer, which ranges from 400 to 900.

4.2.4 Learned Condensed FEM Model for Control Applications

Once learned, the condensed FEM model can be used in a control loop scheme. We suppose that $\mathsf{W}_0^{\text{tri}}$ and δ_0^{free} are known. δ_a and δ_c are obtained through measurements on the robot. Then, the trained network is used to predict $(\widetilde{\mathsf{W}}^{\text{tri}}, \widetilde{\delta}^{\text{free}}) = F(\delta_a, \delta_c, \mathsf{W}_0^{\text{tri}}, \delta_0^{\text{free}})$, where $\widetilde{\mathsf{W}}$ is reconstructed from $\widetilde{\mathsf{W}}^{\text{tri}}$.

Even if the condensed FEM model is learned on a predefined set of fixed goals $\mathbf{x}_{\text{goal}}^{\text{train}}$, one for each of the effectors according to $\widetilde{\delta}_e^{\text{free}} = \mathbf{x}_{\text{effector}} - \mathbf{x}_{\text{goal}}^{\text{train}}$, it can be easily adapted to other goals $\mathbf{x}_{\text{goal}}$ during the control phase with the formula:

$$\delta_e^{\text{free}} = \widetilde{\delta}_e^{\text{free}} + \mathbf{x}_{\text{goal}}^{\text{train}} - \mathbf{x}_{\text{goal}} \tag{4.5}$$

The learned quantities are used in an inverse optimization problem using a quadratic optimization solver. The aim is to compute the actuation force vector λ_a necessary for controlling the robot. The C++ library Prox-QP [Bam+22] is used for implementing the solver in our work. Finally, the actuation effort λ_a is applied on the robot, and the respective new actuation and contact states δ_a and δ_c are recovered. At this step, corrective motion can be applied to improve performance, stability, and precision. In particular, it enables to palliate the two main sources of errors which are due to both the learning and the simulation-to-reality gap. The control loop starts again with a new prediction of the value until convergence, i.e. until an accuracy or stability criterion is satisfied.

In section 4.3, this pipeline is applied to simulated robots. As the simulation-to-reality transfer of the robots considered in our experiments has already been studied in previous works, the simulation could be considered as a ground truth for assessing our algorithms. For a FEM model, this transfer is possible when the mechanical and geometric parameters of the simulated model are well calibrated, the mesh discretization is precise enough and when the boundary conditions on forces and contacts are respected. Constraints states could then be directly retrieved as results from the simulation. In section 4.4, this workflow is applied on a physical prototype. Constraint states are retrieved from a set of sensors, thus enabling to avoid doing FEM simulation of the robot in the control loop.

4.3 Soft Robot Control using a Condensed FEM Model

4.3.1 Open Loop Control of Soft Robots without Contacts

Following the introduction of the condensed FEM model, a set of experiments are introduced below for highlighting control applications without taking contacts into account. After introducing the two considered robots, open loop control and analysis of mesh influence are performed.

Diamond and Soft Finger Robot without Contacts

The open-loop control is illustrated and validated using two FEM models of different soft robots driven by cables. These robots are shown in Figure 4.2.

- The Diamond robot is a soft parallel silicone robot driven by 4 cables. The FEM model of this robot has been validated in [Dur13a]. The simulation mesh is made of 4147 tetrahedrons.
- The Soft Finger robot is validated in [Nav+20]. It is made up of three segments connected by accordion-shaped joints and actuated by 2 cables. The simulation mesh is made of 1557 tetrahedrons.

These two robots are simulated in SOFA using components from the SoftRobot plugin [Dur13a] [Coe+17]. FEM-based control has already been applied and validated on real robots in the above-mentioned work. Consequently, in the following, simulation results are considered as ground truth.

For each robot, the data are retrieved from the simulation as described in subsection 4.2.3. Statistics about the data are given in Table 4.1. During data acquisition, the space of displacements i.e. the space of relative lengths δ_a imposed on the cables is discretized and the actuation state is observed as the reached length of the cables at equilibrium.

Table 4.1: Data acquisition parameters for each robot for $n_l = 6500$ sample points in the workspace. Final loss and best epoch after 9000 learning iterations.

Robot	Cable workspace		Final Loss	Best Epoch
	$\delta_{a,min}$	$\delta_{a,max}$		
Diamond	0	30	6.52e-5	8960
Finger	0	20	3.79e-6	8850

Learning Results

For evaluating the learning process, the average loss is computed on the test set of random robots configurations. Learning results for the condensed FEM model are given in Table 4.1.

Although there are more constraints in the Diamond problem, it takes the same amount of data samples for training the MLP for the Diamond and the Soft Finger.

This is because more mechanical non-linearities are observed in the Soft Finger case as it implies more element rotations. In this example, the influence of the forces exerted on the actuators on the end effector changes a lot depending on the configuration of the robot, i.e. depending on how bent the robot is.

Control Results

To evaluate the use of the learned condensed FEM model for inverse modeling, a trajectory is considered in the form of several goal positions to be reached by the effector for each robot. As the soft robot dynamics is neglected in this study, the results are only valid in the quasi-static regime. The results are shown in Figure 4.3.

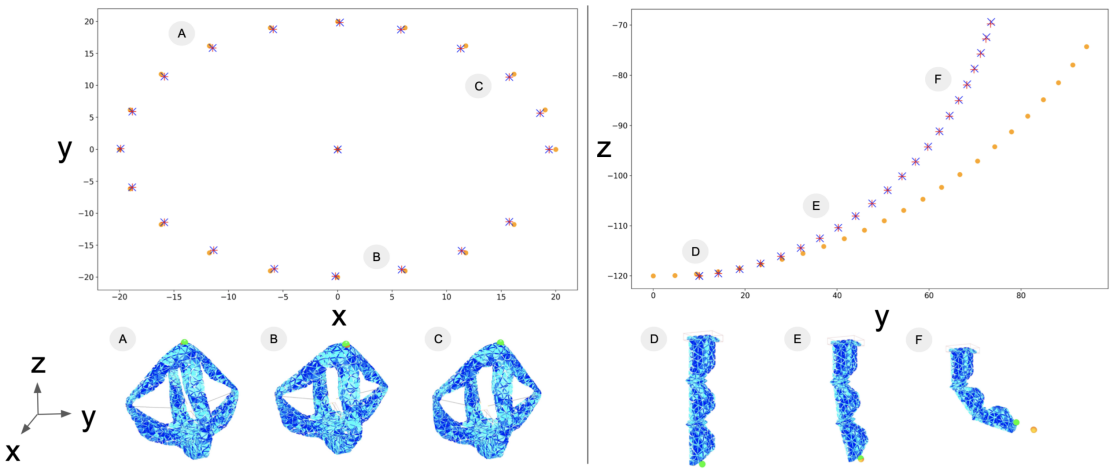

Figure 4.3: Control results for the Diamond and Soft Finger robots without contact. On the left, the trajectory of the end effector of the Diamond robot in the (x, y) plane (mm) for 30 different goals (orange dots) when using learned mechanical matrices (blue cross) or computed FEM mechanical matrices (red cross). 3 different positions of the Diamond robot along this trajectory are shown as examples. The effector is represented by a green dot, and the goal by an orange dot. On the right, the trajectory of the end effector of the Soft Finger robot in the (y, z) plane (mm) for 25 different goals (orange dots) when using learned mechanical matrices (blue cross) or computed mechanical matrices (red cross). 3 different positions of the Soft Finger robot along this trajectory are shown as examples.

A circular trajectory of the effector in a horizontal plane is applied on the Diamond robot. This trajectory is chosen for being convenient for two dimensions visualization although the robot end effector can be controlled in all three directions of space. Then the positions obtained for each goal using the condensed FEM model learning-based control pipeline and the classical inverse model are compared. In most configurations, the positions reached by the tips in both cases are very similar. This shows that the compact model is a good representation of the system. In some positions, small-scale errors between the learned model and the full model can occur: this comes from the learning errors specific to the method. This error can be further decreased by increasing the discretization of the actuation space during data collection and by judicious choices of network hyper-parameters.

For the Soft Finger robot, a trajectory is chosen to force the robot to bend. Both control pipelines using the learned condensed FEM model and classical FEM model achieve similar results. Training errors can be explained by the same reasons as for the Diamond robot. In the case of the Soft Finger robot, the desired trajectory is not in the robot's workspace. This highlights that our framework relies on an optimization algorithm seeking to respect the soft robot's physical constraints.

Analysis of Mesh Refinement Influence

Mesh refinement generally results in more accurate simulations at the cost of significant computation time. A compromise between speed and precision is needed for FEM-based control of a soft robot in real time.

As the condensed FEM model is built from the projection of mechanical matrices in constraint space, its size is independent of the mesh size. With a control based on a learned condensed model, the additional computation costs of mesh refinement are transferred to the offline phase of learning. This is experimented with the example of the three condensed FEM models of the Soft Finger, with different mesh sizes (see Figure 4.4).

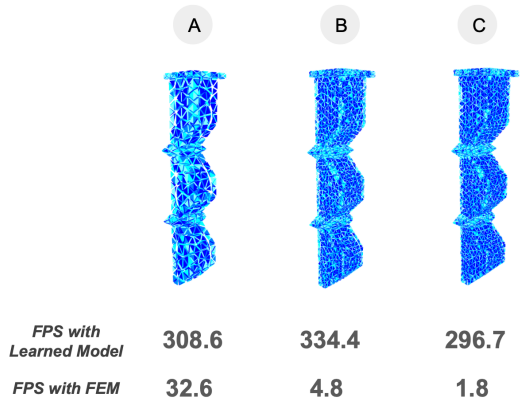

Figure 4.4: Soft Fingers with different mesh resolutions: A) 1557 tetrahedrons, B) 8895 tetrahedrons, C) 17852 tetrahedrons. For each of them, the average FPS of the simulation when using the learned condensed or doing online FEM simulation is displayed.

6500 Soft Finger states are sampled for building the training set, and a fixed number of 9000 training iterations is performed for each Soft Finger. The magnitude order of final training losses is similar, being respectively 3.79×10^{-6}, 4.26×10^{-6} and 3.60×10^{-6} for meshes resolution A, B and C. Moreover, simulations FPS are similar with the three discretizations as shown in Figure 4.4.

There is no variation in FPS when using the learned condensed model. Whatever the size of the mesh, the learned neural network queries allowing to get the mechanical state of the robot takes about the same amount of time and results in similar prediction performances. The learned condensed model complexity does not depend on the mesh size but on the number of actuators, effectors, and contacts in the problem. As the mesh resolution increases, the FEM simulation is more accurate. Thus, training on a more faithful data set leads to a more faithful result, without resulting in additional computational expense for using the trained neural network.

Illustrating the Modularity of the Condensed FEM Model regarding Different Inverse Problems and Redundant Soft Robots

Mechanical matrices learned for a single soft robot can be used to formulate a single inverse problem merging several identical soft robots. Moreover, both the data acquisition and training phases of the condensed FEM model can be efficiently reduced for redundant robots.

This is illustrated with the example of a 2D Gripper robot made of two identical Soft Finger robots as described in the previous section. The mechanical matrices of the system can be built

from the matrices computed for a single Soft Finger. The learned mechanical representations of the two Soft Fingers are mechanically coupled by adding a force between the effectors. This force corresponds to a configuration in which the two Fingers catch an object by enabling one Finger to reach a position while forcing the other Finger to follow it. To take this coupling into account, the optimization problem is reformulated according to:

$$\begin{aligned} arg_{\lambda^t_{a,1},\lambda^t_{a,2},\lambda^t_e} min \quad & \|\mathbf{P}^t_{e_1} - \mathbf{P}_{goal}\|^2 \\ s.t. & \mathbf{P}^t_{e_1} + \beta = \mathbf{P}^t_{e_2} \end{aligned} \tag{4.6}$$

where $\mathbf{P}^t_{e_1}$ and $\mathbf{P}^t_{e_2}$ are the respective positions of the effectors of Fingers 1 and 2 at time t, $\mathbf{P}_{goal}$ the target position of $\mathbf{P}^t_{e_1}$, β a distance describing the size of the object, $\lambda^t_{a,1}$ and $\lambda^t_{a,2}$ the forces applied on the cable actuators of respectively Fingers 1 and 2 at time t and λ^t_e the coupling force shared on the two Fingers. Both the quantities $\mathbf{P}^t_{e_1}$ and $\mathbf{P}^t_{e_2}$ can be written with the compliance matrices projected in the constraint space as

$$\begin{aligned} \mathbf{P}^t_{e_1} &= \mathbf{P}^{t-1}_{e_1} + \widetilde{\mathbf{W}}^t_{ea,1}(\lambda^t_{a,1} - \lambda^{t-1}_{a,1}) + \widetilde{\mathbf{W}}^t_{ee,1}(\lambda^t_e - \lambda^{t-1}_{e,1}) \\ \mathbf{P}^t_{e_2} &= \mathbf{P}^{t-1}_{e_2} + \widetilde{\mathbf{W}}^t_{ea,2}(\lambda^t_{a,2} - \lambda^{t-1}_{a,2}) - \widetilde{\mathbf{W}}^t_{ee,2}(\lambda^t_e - \lambda^{t-1}_e) \end{aligned} \tag{4.7}$$

where the $\widetilde{\mathbf{W}}_{ea}$ and $\widetilde{\mathbf{W}}_{ee}$ values are the predicted projected compliance values. The quantities at time $t-1$ are known quantities that are not involved in the optimization.

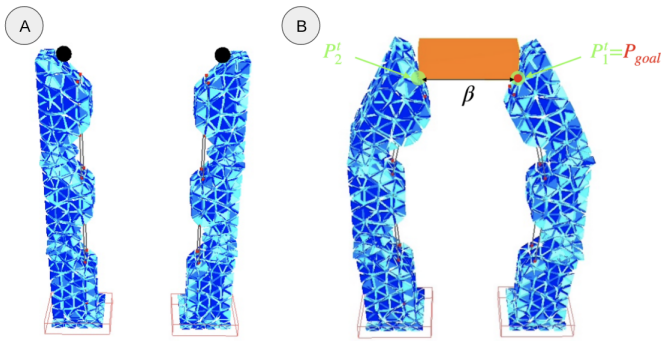

Figure 4.5: A) 2D Gripper robot controlled using effectors located at the tip of each finger. B) Position obtained using mechanical coupling between Soft Fingers 1 and 2 and learned Finger mechanical representation. The position of the effector of Finger 1 $\mathbf{P}^t_{e_1}$ is given by a goal point while the position of the effector of Finger 2 $\mathbf{P}^t_{e_2}$ is imposed by the coupling. This coupling is represented in the picture by an orange object to grasp.

This optimization problem enables to mechanically couple the two Fingers using the values learned for a single Finger. Indeed it is possible to build $\widetilde{\mathbf{W}}_2$ from the same network used to predict $\widetilde{\mathbf{W}}_1$ by carefully considering the orientation of the fingers. $\widetilde{\mathbf{W}}_2$ is thus obtained from a rotation of $\widetilde{\mathbf{W}}_1$. Solving this optimization problem applied to the Gripper robot enables to catch an object. Starting from the position given in the Figure 4.5.A, the resolution of Equation 4.6 leads to the result given in Figure 4.5.B. The movement of the gripped object between the two Fingers can be controlled. However, there is a limitation when the force applied to the effectors of the Fingers becomes important for extreme actuation positions. To solve this problem, it is necessary to integrate the possibility of putting forces on the effector during the learning stage, as described in the following section.

4.3.2 Open Loop Control with Contacts

In this section, we introduce contact modeling using the condensed FEM model, considering mostly manipulation control tasks.

Soft Gripper Robot with Contacts

To evaluate the condensed learned model with contact, we use the Soft Finger. First, it is used to press a button. Then we combine the same soft finger three times to create a gripper and manipulate an object. We made several simplifying assumptions: (i) the gripped object is always rigid, (ii) the location of the contacts on the actuated flexible finger is fixed and placed on the surface of the last phalanx, (iii) the contact points are always active and friction-less.

The condensed FEM model and the two considered application cases are illustrated in Figure 4.6:

- A soft finger presses a cube-shaped button. The resistance of the button is modeled as a constant horizontal force. This example is used to validate the learning of the different mechanical quantities of the condensed FEM model related to contacts, namely W_{cc}, W_{ca}, and δ_c^{free}.
- A Soft Gripper manipulates a cube. The gripper consists of 3 identical Soft Finger robots. This example also illustrates that a single condensed FEM model learned for one Soft Finger robot can be used in another inverse optimization scheme coupling several Soft Fingers.

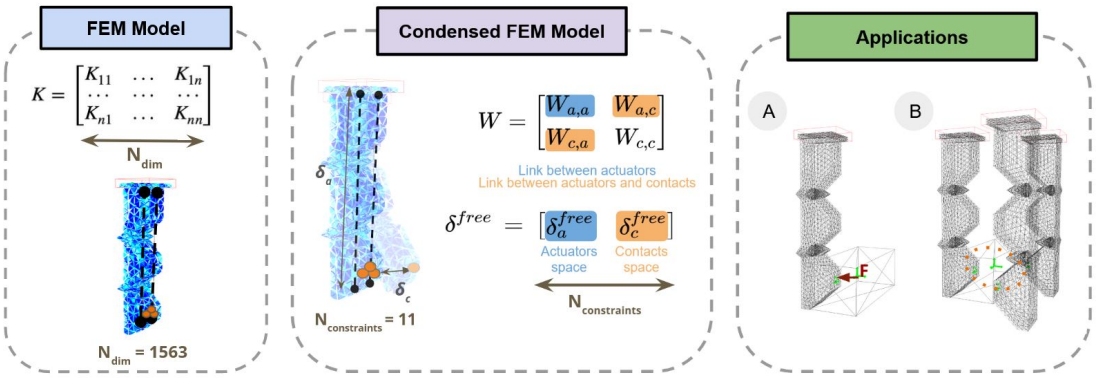

Figure 4.6: Illustration of the condensed FEM model for a Soft Finger with contacts and the two considered control scenarios (right). The Soft Finger is actuated by 2 cables and has 3 fixed contact points located at the tip. The FEM model of the robot (left) is used to learn a condensed model (middle). The condensed model is based on the projection of the compliance matrix into the constraint space, resulting in drastically reducing the number of characteristic dimensions describing a robot state. The scenarios are A) a Soft Finger robot pushing a button, and B) a Soft Gripper robot manipulating a cube. The cube effector is represented by a green frame.

When generating the dataset for learning the condensed FEM model for both scenarios, the sampling space is directly proportional to the number of both actuation and contact constraints. One encountered issue is that exploring the constraint space has exponential complexity with the number of constraints. In the case of the Soft Finger, there are 2 constraints relative to actuation

and 9 constraints relative to contacts, since we fixed the number of contact points to 3. Moreover, using a box sampling on both actuation and contact constraint spaces would result in many robot states that are irrelevant to our applications. That is why in both cases, the dataset is generated by sampling in the cube effector displacement space and using an inverse model to find the corresponding robot configuration. This means that the position of the cube is sampled first, and then an inverse model is used to find the active constraints to reach it. The sampling space size is then reduced from 11 constraints to 3. As the two scenarios involve different configurations of the cube, two different training sets are considered, each time for a single Soft Finger. Respective final test loss for training the condensed FEM model for the Soft Finger and one Soft Finger of the Soft Gripper over 50000 training iterations are 1.67e-6 and 1.24e-3.

Inverse Problem for Soft Finger/Object Interaction

Let $X_{\text{cube}} \in \mathbb{R}^6$ be the position of the cube in both translation and rotation. Let $^{\text{prev}}$ be the notation to refer to quantities calculated at the previous time step in an iterative process. In a manipulation task, we want to control the translation of the cube. It is then equivalent to minimizing the following quantity:

$$\min_{\lambda_a} \|\mathsf{X}_{\text{cube,trans}} - \mathsf{X}^{\text{goal}}_{\text{cube,trans}}\|^2 \tag{4.8}$$

where $\mathsf{X}^{\text{goal}}_{\text{cube,trans}}$ is the translation component of the goal position to reach, $\mathsf{X}_{\text{cube,trans}}$ the translation component of X and $\|.\|$ is the euclidean norm. To solve this problem, X_{cube} has to be written as a function depending on the actuation force λ_a. We use condensed mechanics and cube kinematics to obtain this function.

Kinematics of Points Placed on a Rigid Object From the hypothesis that contact points location on both the Soft Finger and the object are shared, it results that $\lambda_c = \lambda_{c,\text{finger}} = -\lambda_{c,\text{cube}}$ and $\mathbf{x}_{c,\text{finger}} = \mathbf{x}_{c,\text{cube}}$, where $\mathbf{x}_{c,\text{finger}}$ and $\mathbf{x}_{c,\text{cube}}$ are the euclidean positions of the contact points between the Soft Finger and the cube.

For an infinitesimal displacement of X_{cube}, the positions of X_{cube} and $\mathbf{x}_{c,\text{cube}}$ are linked as:

$$\mathbf{x}_{c,\text{cube}} - \mathbf{x}^{\text{prev}}_{c,\text{cube}} = \mathsf{J}_c(\mathsf{X}_{\text{cube}} - \mathsf{X}^{\text{prev}}_{\text{cube}}) \tag{4.9}$$

where $\mathsf{J}_c \in \mathbb{R}^{N\times 6}$ is the Jacobian matrix linking the kinematics of the manipulated object to the contact points and N the number of contact points.

From this Jacobian matrix, the resulting contact force on the cube is then expressed as $\mathbf{f}_{\text{cube}} = \mathsf{J}_c^T \lambda_c$.

Condensed mechanics applied on a Soft Manipulator

Knowing the contact forces $\mathbf{f}_{\text{cube}}$ imposed on the cube, and multiplying by $\mathsf{D} = \mathsf{J}_c^T \mathsf{W}_{cc}^{-1}$, the following relation is obtained:

$$\begin{aligned} \mathsf{D}\mathbf{x}^{\text{prev}}_{c,\text{cube}} + \mathsf{D}\mathsf{J}_c(\mathsf{X}_{\text{cube}} - \mathsf{X}^{\text{prev}}_{\text{cube}}) = \\ \mathsf{D}\delta_c^{\text{free}} + \mathsf{D}\mathsf{W}_{cc}\lambda_c + \mathsf{D}\mathsf{W}_{ca}\lambda_a \end{aligned} \tag{4.10}$$

Manipulating the equation gives a function $\mathsf{X}_{\text{cube}} = \mathsf{A}\lambda_a + \mathsf{B}$ as $\mathsf{D}\mathsf{W}_{cc}\lambda_c = \mathbf{f}_{\text{cube}}$ is known. The optimization objective from Equation 4.8 is now well-posed.

Both the Soft Finger robot and the Soft Gripper robot differ in the way the force $\mathbf{f}_{\text{cube}}$ is computed. From the new values of X_{cube} and λ_a, it is then possible to compute a new value of λ_c

using Equation 4.10.

Application: Soft Finger pushing a Press-button.

We aim to control the position of a cube through its contact with the soft finger. A known constant horizontal force **F** is opposed to the orientation of the applied bending movement. The contact force $\mathbf{f}_{\text{cube}}$ can be deduced from **F**. A trajectory of positions of the center of the cube X_{cube} following a horizontal line is given as the target. As the main direction of deformation of the Soft Finger is in its bending direction, only a single degree of freedom of the cube is controlled. The results are shown in Figure 4.7.

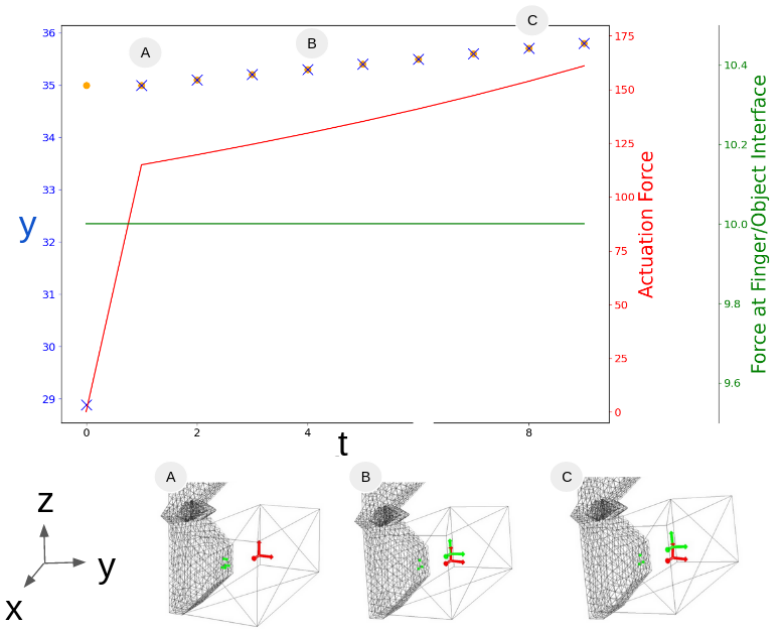

Figure 4.7: Control results for the Soft Finger robot pressing a button. The reached horizontal position (mm) of the cube is compared on a trajectory of 9 successive goals (orange dots) when using learned mechanical matrices (blue cross). The contact force exerted at the interface between the Soft Finger and the button as well as the sum of actuation forces exerted on the two cables during the trajectory are also displayed. 3 different positions of the Soft Finger robot met during this trajectory are shown as examples. In this representation, the effector is a green frame, and the goal is a red frame.

In this application, the soft robot open-loop control using the condensed FEM model shows a similar degree of precision to the simulation-based control.

Application: Soft Gripper Manipulating a Cube.

This experiment aims at manipulating a cube using three Soft Fingers.

Mechanical Coupling Using Contacts for Manipulation

The mechanical matrices of the Soft Gripper robot are built from the mechanical matrices computed on three single Soft Finger. Compared to the scenario of the Soft Finger pushing a button, applied forces on the cube are not known this time.

After solving the same problem as for a single Soft Finger, Equation 4.10 enables to express the contact forces λ_c with the new values of the actuation forces λ_a and the 6D position of the cube X_{cube}. The corresponding algorithm to control the robot is written as follows:

Algorithm 1 Control algorithm for object manipulation scenario.

Ensure: Start from $\lambda_c = 0$.
for each time step **do**
1. Compute λ_a from Equation 4.8 using $\mathsf{X}_{\text{cube}} = \mathsf{A}\lambda_a + \mathsf{B}$ and the previous evaluation of λ_c. λ_c is used for computing B.
2. Evaluate the new value of λ_c using Equation 4.1 and the new value of λ_a.

end for

In this algorithm, both λ_a and λ_c are updated once per time step using previous values. Experimentations showed that this simple scheme is sufficient to control the robot. Another valid choice would be to consider using an internal iterative procedure for gradually updating λ_a and λ_c until their respective convergence at each time step. With the design choices for the Soft Gripper robot, only small movements of the cube are enabled around its resting position. Indeed, the initial configuration of the Soft Fingers in contact with the cube consists of fully relaxed cable actuation. Thus, the motion is limited in the opposite direction of the Soft Fingers bending.

Results

For the experiment, a circular trajectory of the object center is considered. The results are shown in Figure 4.8.

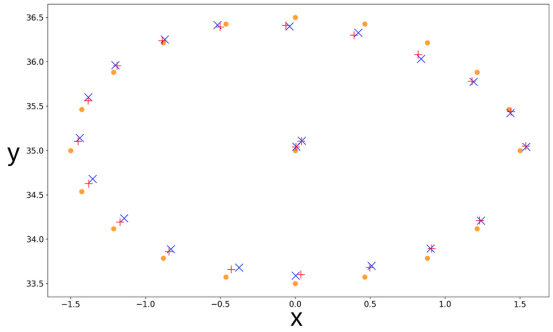

Figure 4.8: Control results for the Soft Gripper robot. The positions of the manipulated cube (mm) are compared on a trajectory for 20 different goals (orange dots) when using learned mechanical matrices (blue cross) or mechanical matrices computed from the full simulation (red cross).

On the considered trajectory, slight differences between goals and reached effector positions are observed both with the simulation-based and learning-based controllers. In addition, a small effector-goal gap of about 0.04 mm is observed along the vertical z-axes in both cases. This is explained by the modeling of the manipulator which considers Soft Fingers contact points to always be at the same location on the manipulated object. These errors could be reduced using a closed-loop controller.

Moreover, small differences between the reached positions of both the simulation-based and learning-based controllers are observed. Compared to the case of the Soft Finger pushing a

button, training results are less good. This is explained by the complexity of the dataset, as various forces can be applied to the cube.

For the moment, the application case considers only small deformations. However, results show that contacts information can be taken into account in the learned condensed FEM model, which is promising. Future works may consider application cases with larger deformation and collision detection/resolution so to avoid having the fixed contact points location hypothesis. This may open the path to locomotion applications.

4.4 Embedded Control of Soft Robot

In this section, it is demonstrated how the condensed FEM model can be used to control a physical pneumatic robot ongoing large displacements without relying on online FEM simulation, leveraging both the small scale and prediction speed of the neural network for application in embedded control.

4.4.1 Pneumatically Actuated Continuum Robot.

The considered soft robot is a pneumatic trunk robot initially introduced in [Abi+18]. The robot is a highly flexible and pneumatically driven soft robot composed of two modules including three fully fiber-reinforced chamber pairs each. This robot was developed for medical applications as an in-vivo cancer diagnostic tool during minimally invasive interventions. The physical robot and the associated simulation are shown in Figure 4.9.

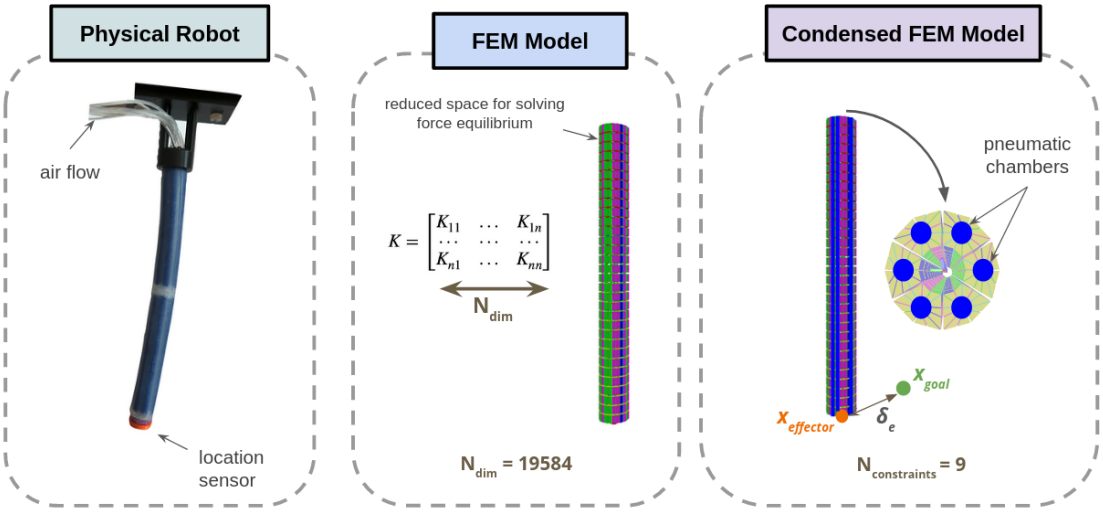

Figure 4.9: Illustration of the physical prototype and the condensed FEM model for the PAC robot. The robot is actuated by 6 pneumatic cavities represented by blue cylinders. A location sensor is located at its tip. The reduced FEM-based modeling from [Cha+23] is displayed as the FEM model.

Physical prototype: The chamber pressure is regulated and monitored by solenoid valves (Rexroth ED02 R414002399). This pressure regulators are voltage-actuated, and the proportional voltage is generated thanks to a dedicated voltage output board (PhidgetAnalog 4-Output

1000_0b). Pressurized air is supplied to the regulators by a compressor (Fengda FD-186). An electromagnetic tracking system (Polhemus Liberty) monitors the end-effector position and orientation. The system may be connected both to a computer or Raspberry Pi, synchronized and connected with the learned model through Python scripts. An illustration of the hardware setup with a Raspberry Pi is provided in Figure 4.12.

Simulation: The Stiff-Flop robot is built using a beam-based reduced finite element method model as described in [Cha+23]. Compared to [Cha+23], the sensor stiffness cannot be neglected on our physical prototype. We consequently adapt the stiffness of the modules leading to a Young Modulus of 150kPa and 90kPa respectively for the base and the top modules.

Learned condensed FEM model: The condensed FEM model is learned upon the beam-based reduced model and not on the full-FEM simulation to speed up the sampling phase and to highlight the capacity of the condensed FEM model to further decrease computation cost, even for already optimized simulation. Data are collected in the space of the volume change of the cavities. The dataset is composed of 30000 configurations of the robot. The final test loss for training the condensed FEM model for this soft robot over 50000 training episodes is $6.05e-5$.

4.4.2 Control using the Condensed FEM Model with Minimal Sensor Feedback

A strong assumption of the control loop used in previous control applications is that both the actual displacements of the actuators and the actual positions of the effectors are observable. Practically, this requires the systematic use of a set of adapted sensors, introducing an additional constraint on the physical prototype. This issue is overcome by the following framework targeting both embedded open and closed-loop control of a robot from absolute position sensor feedback.

Concerning the Stiff-Flop robot, the system is pressure-controlled. There is no direct access to the position of the actuator. In our case, it would require a sensor of the volume of each cavity and there is no trivial way to build such a sensor. Instead, the condensed FEM model is used as a state observer. The value of the volume of the cavities is computed based on the actuation λ_a and the projected mechanical matrices from $\delta_a(x) = \mathsf{W}_{aa}^{\text{prev}} \lambda_a + \delta_a^{\text{free,prev}}$ where ^{prev} stands for the evaluation of the mechanical quantities in the previous iteration of the control loop. This strategy results in a time-step shift on the predicted mechanical matrices used for computing the actuation displacement. This shift may be corrected using a suitable corrector in a closed-loop setting.

The general embedded control loop framework is as described in Alg. 2.

To close the loop, a measurement taken on the physical robot must be compared with the values predicted by the used control scheme. In the proposed experiment, a location sensor is placed on the tip of the robot, and a PI controller is set on the effector displacement (see Equation 4.5). The same proportional and integral coefficients are considered for all the experiments, independently of the frequency of the control loop.

4.4.3 Experiments in a Virtual Environment

To evaluate the use of the learned condensed model for inverse control of the Stiff-Flop robot, a 40mm radius circular trajectory around the robot tip is considered. This trajectory takes the form of a succession of 200 goal positions to reach the circle from the initial rest position followed by 600 successive goal positions describing the circle itself.

The objective of the following experiments is to evaluate the introduced control scheme used with the learned condensed FEM model in open-loop. Initially, the experiments are carried out

Algorithm 2 Embedded control loop with end-effector location feedback only.

Ensure: Access to W_0 and δ_0^{free} from the training dataset. Set initial actuation $\lambda_a = 0$ and initial actuators displacement $\delta_a = \delta_{a,0}^{\text{free}}$.
for each time step **do**
1. If considering to close the loop, recover the position of the effectors and compute the corrected goal position.
2. Predict the mechanical quantities $\widetilde{\mathsf{W}}$ and $\widetilde{\delta}^{\text{free}}$ according to Equation 4.2 without contact, using the trained neural network.
3. Solve the optimization problem to find the force λ_a.
4. Compute the new actuator state from mechanical matrices and forces as $\delta_a = \mathsf{W}_{aa}\lambda_a + \delta_a^{\text{free}}$ for querying the learned condensed FEM model at the next iterations.
5. Apply the forces λ_a on the robot.
end for

in a virtual environment to limit the artifacts associated with the simulation to reality transfer. The positions errors are compared when using either the computed state of the actuation δ_a or the predicted state of the actuation $\delta_a = \mathsf{W}_{aa}^{\text{prev}}\lambda_a + \delta_a^{\text{free,prev}}$ as explained in the previous section.

In both cases, the trajectory is closely matched in open-loop, with a maximum error of approximately 1.8 mm using the computed state and 5 mm using the predicted state. These results show that the robot is capable of following the circular trajectory, even if no actuation states are measured in the simulation as in the second case. When the predicted state is used, a constant shift depending on the discretization step used for the integration is nevertheless observable. Indeed, increasing the number of intermediate goals sampled on the circular trajectory results in decreasing the error on the robot state estimation between two successive points. This shift can also be compensated by setting up a closed-loop control scheme based on minimal sensor feedback, as shown in the following experiments on the physical prototype.

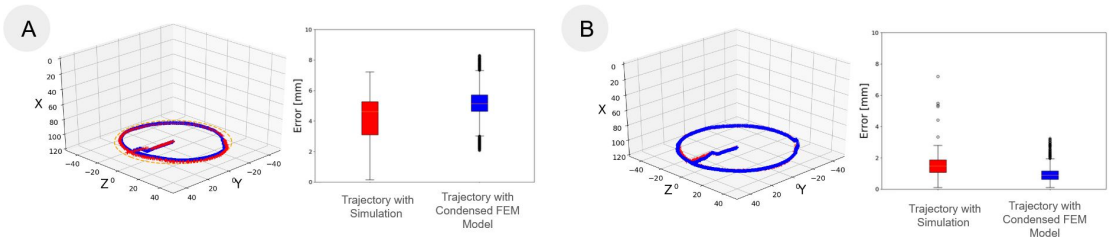

Figure 4.10: Performed trajectory and tracking errors obtained for performing a circular trajectory (orange dots) using condensed FEM model-based control schemes with the physical prototype of the robot. Errors are computed as the absolute distance between the desired and the reached positions. Experiments are A) Comparison of full order simulation (red) based and condensed FEM model with minimal sensor feedback (blue) based open-loop control, B) Comparison of simulation (red) based and condensed FEM model with minimal sensor feedback (blue) based closed-loop control.

4.4.4 Experiments with the Physical Prototype

Hardware Setup Description

The chamber pressure is regulated and monitored by solenoid valves (Rexroth ED02 R414002399). This pressure regulators are voltage-actuated, and the proportional voltage is generated thanks to a dedicated voltage output board (PhidgetAnalog 4-Output 1000_0b). Pressurized air is supplied to the regulators by a compressor (Fengda FD-186). An electromagnetic tracking system (Polhemus Liberty) monitors the end-effector position and orientation. The system may be connected both to a computer or Raspberry Pi, synchronized and connected with the learned model through Python scripts. An illustration of the hardware setup with a Raspberry Pi is provided in Figure 4.12.

Evaluation of the Transfer to Reality

The control is tested on both the simulated and the physical robot. Results are shown in Figure 4.10. Both open and closed-loop scenarios follow the same 40mm circle trajectory. This validates the use of this model for robot control, under the same conditions as the physical prototype, and also demonstrates the benefits and drawbacks of this system for positioning purposes.

The computational cost cutback lets the learned model run at 10e3 Hz. The communication with the hardware setup limits the feedback loop at 90.9 Hz which is limited for example by the Polhemus sensor, able to communicate at 100 Hz. With the classic beam-based reduced model, the simulation runs at 5 Hz, so the communication system does not limit the control loop frequency.

In the rest of this section, we will detail and discuss the results obtained with the physical prototype in terms of open loop, closed loop, convergence speed of the closed loop control and embedded control.

Open-Loop Control

As computing a robot state using the learned condensed FEM model is way faster than when using the reduced FEM-based simulation [Cha+23], the actuation pressures are also applied faster. This may lead to dynamical effects that are not taken into account in the quasi-static modeling used for learning the condensed FEM model. Reducing these dynamical effects is then performed by increasing the number of intermediate goals sampled on the trajectory. At each time step, the robot's inverse problem is solved using optimisation tools, and the actuation to move from one point to another is imposed. By reducing the distance between two successive points, the variation in actuation to reach the next point in one time step is reduced. Acceleration is also reduced, which reduces dynamic effects. The greater the number of points on the circle, the less the dynamic effects affect the model's performance. The quasi-static assumption is thus performing better if the number of points on the trajectory is increased for a given time budget. This criterion is used to compare the temporal performance of the models over several goals, as well as the accuracy they achieve within a given time budget. In practice, we consider that we have the same calculation time budget for both models to achieve a circular trajectory. This budget is set to 80 seconds.

Concerning the computational performance of the model, the idea is to know how many points can be considered in the allocated time. As the model has to be calculated for each point to be reached, this value gives the average calculation time for both models. For 80 seconds of calculation, 3300 points can be reached with the learned condensed model. Only 500 points can

be reached with the beam-based model. The open-loop built from the learned condensed FEM model is 7 times faster on average than a reduced model that uses beam theory to be simulated.

Concerning the accuracy of the model under time budget constraint, results are displayed in Fig 4.10.A. It demonstrates the ability to transfer to a physical prototype using the learned condensed FEM model without actuation sensors. In fact, the trajectory errors when using the learned condensed FEM model have the same magnitude as when using the beam-based FEM model. The little difference between the mean error could be explained by the use of predicted actuation states from previous mechanical state, leading to a one-time step shift between the mechanical state values evaluated in the two methods.

Thus, using the condensed FEM model does not enable increasing the speed of doing the entire circle trajectory under strong accuracy constraints for the Stiff-Flop robot. Indeed, increasing the speed will directly increase the positioning error due to the dynamic effect. However, it is possible to consider more points in the trajectory for the same time budget with the learned condensed model. This makes it possible to compensate more quickly for errors in a closed loop, as demonstrated in the next experiment.

Closed-Loop Control

Assessing the closed-loop trajectory is performed with the same iteration distribution and under a time budget as for the open-loop experiment. The proportional and integral error weights are respectively $K_P = 0.01$ and $K_I = 0.002$. The choice of the control approach is not the focus of this work. The objective is to show the interest of the high frequency of the condensed model in a closed control loop scenario, without trying to have the best closed loop for our task. It is certain that other more efficient control schemes could be designed.

Results are displayed in Fig 4.10.B. The positioning accuracy improvement is noteworthy when using the learned condensed FEM model-based closed-loop. This accuracy improvement is mainly due to the PI correction. Indeed, the correction frequency is way higher with the learned model, so the system compensates better both modeling and control errors compared to the closed-loop using the beam-based reduced modeling. Indeed, for the same time budget, the learned condensed FEM model may use higher iterations, so it will apply more successive corrections to reach more accurate positions. The high computing frequency related to using the learned condensed FEM model overcomes the inaccuracies for closed-loop control applications.

Convergence Speed of the Closed Loop Control

In this experiment, the convergence speed of the closed loop is evaluated on the task of reaching a point in space with a given precision, i.e. when the error to the target is less than a given threshold. Practically, we consider that the error threshold is reached when the error remains below $\epsilon = 0.5$mm for at least 2.4 seconds which corresponds to around 10 simulation steps of the FEM model.

The results are plot in Figure 4.11. The error threshold is reached in more than 100 seconds for the beam-based model, compared with around ten seconds for the learned condensed model. Due to its high correction speed, the closed loop based on the learned condensed FEM model reaches the desired position in space faster than with the beam-based reduced model control loop. It shows that a PID corrector can be used to correct iteratively any errors faster with the learned condensed model, as the learned condensed FEM model prediction is faster than the beam-based model prediction, and the error between the two models remains small.

This speed improvement demonstrates the utility of the learned condensed FEM model for accurate positioning control. This demonstrates very interesting properties for accurate medical applications like biopsy retrieval where it is needed to reach point-per-point positions

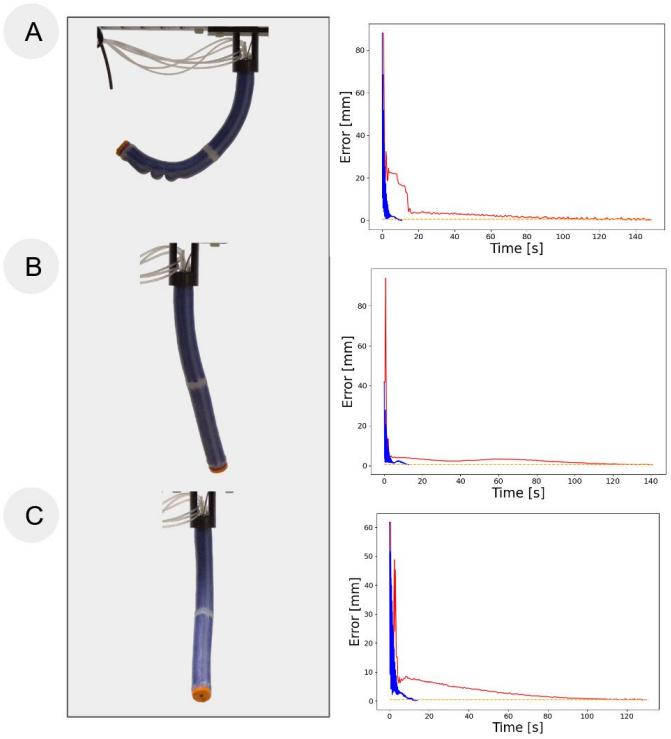

Figure 4.11: Convergence histories obtained in a closed loop setting for reaching a point, and pictures of the associated reached state by the physical prototype. Log-scale error histories are compared respectively for simulation-based (red) and condensed FEM-based (blue) closed-loop control. The aimed error is 0.5mm to the target point (orange dot line). Considered target points are A) [57, -5, -70], B) [106, 26, 32.5], and C) [92.5, -58.5, 9].

with accuracy criterion. In addition, as shown in the supplementary video, this high-frequency correction helps to react faster to external disturbances such as contact without modeling them. The minimal sensor configuration is considered. The value of δ_a (and therefore the predictions of W and δ^{free}) depend on the λ_a and the position of the end effector. When the robot is disturbed, the position of the effector no longer corresponds to the position given by the condensed model. Indirectly, this disturbance information is taken into account in the δ_a. The closed loop is then used to quickly react to this kind of error, even if no disturbance effect has been taken into account during training. Depending on the considered task, this feature could be an asset to ensure a good trade-off between precision and speed over a considered trajectory.

Embedded Control

The control loops developed with the learned condensed FEM model have low computation and memory resource consumption. Indeed, the memory consumed by the main controller loop is estimated at 660 MiB on average.

The full algorithm has been directly embedded in a Raspberry PI 4. It has been tested on the different scenarios and features comparable results, demonstrating using the embedded control loop based on the condensed model on microprocessors. The results in terms of positioning

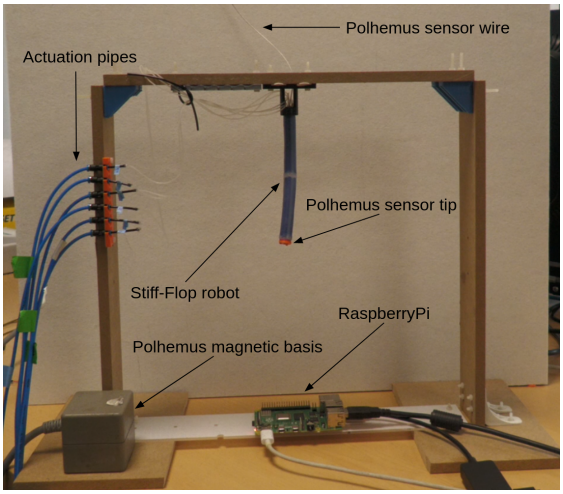

Figure 4.12: Picture of the setup with the Raspberry Pi.

are identical to those presented in Figure 4.10. The difference is on the calculation time, which is longer on a Raspberry PI 4 than on a computer with a high-performance processor as the calculation frequency is not the same.

This implementation demonstrates the clear advantage of using the learned condensed FEM model for embedded robot control, which may run with good performances on light hardware. This could be useful for control in contexts of robot autonomy, such as flying robots [RK22] or aquatic exploration applications [SNW17].

4.5 Condensed FEM Modeling Extended to Calibration and Design

In classical design optimization loops, designs are evaluated in simulation for iteratively updating design parameters. The computation time of the entire process is directly proportional to the computation time needed for simulating a design. Recent works in the community consider training a surrogate to directly learn the values of the fitness function relative to the design variables, as we did in Chapter 3.

Changing the fitness function then requires to regenerate data and re-train a model.

The specificity of our model is to learn mechanical matrices instead of fitness functions, and account for controller variables on all the considered workspace. In fact, both the W and δ^{free} mechanical matrices directly capture the link between the robot's actuators, contact points, and effectors. This link depends directly on the robot's body and its mechanical properties. It is then possible to write multiple fitness functions characterizing design performance and control based on the condensed FEM matrices.

We make the hypothesis that a design can be described by the mechanical matrices characteristics of its rest state, namely W_0 and δ_0^{free}. This hypothesis will be challenged in the applications in the next sections. However, W_0 and δ_0^{free} does not enable directly to generate this design. This is why we introduce some explicit design parameters $\mathbf{p}$ which makes sense for the user.

The idea is therefore to build a differentiable link as follows:

$$\widetilde{\mathsf{W}}_0, \widetilde{\delta}_0^{\text{free}} = G(\mathbf{p}) \tag{4.11}$$

where G is a MLP network. Data are collected in the same way as for the network F and the training loss is similar to Equation 4.4. Then applications such as calibration and design optimization can be explored. When design parameters are relevant, the condensed FEM model is extended following:

$$\widetilde{\mathsf{W}}, \widetilde{\delta}^{\text{free}} = F(\delta_a, \delta_c, G(\mathbf{p})) \tag{4.12}$$

This enables building a differentiable link between design parameters and mechanical quantities given active constraint states as illustrated in Figure 4.13. Differentiable fitness functions relative to $\mathbf{p}$ can then be designed from W and δ^{free} for assessing a design performance.

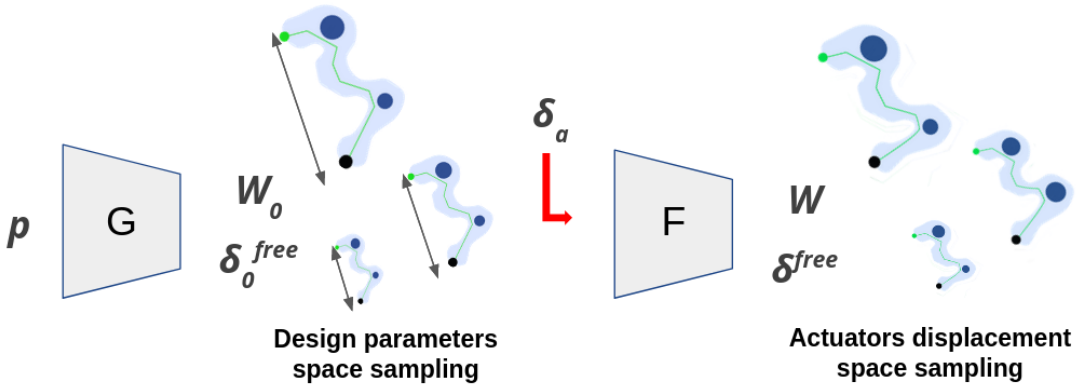

Figure 4.13: Illustration of the pipeline for learning mechanical states of Soft Robot. G predicts the mechanical state of the robot in its rest position from design parameters, and F predicts the mechanical state of the robot from both G outputs and a given actuation displacement.

Using the condensed FEM model for design applications presents some advantages. In conventional design optimization frameworks, once a design is optimized, all computed simulation data are lost in the process. Training a condensed FEM model on a data library containing variations of a robot design still requires many simulations to generate the training set. However, once the condensed FEM model is learned, using it both in the design optimization loop and for control tasks does not require performing any more simulations. Another interesting aspect related to learning for both actuation states and design parameters altogether is that it may help to limit the number of needed samples in the training dataset compared to learning a condensed FEM model separately for each design. Finally, in the case of the condensed FEM model, since fitness functions are directly formulated from the learned matrices, a single condensed model can be used to optimize a robot across multiple tasks.

After introducing the considered Soft Finger parametric design, the performance and expressivity of the condensed FEM model relative to geometrical and mechanical design variables is analyzed. Then, Soft Finger design calibration and optimization scenarios are explored, leveraging the formulation of 3 different fitness functions from learned mechanical matrices.

4.5.1 The Soft Finger Parametric Design

In our numerical experiments, a parametric design of Soft Finger is considered, building upon the one proposed chapter 3. Compared to the one previously considered in this work, there is only one cable actuator going from the base to the tip. The considered parametric Soft Finger and its geometrical design variables are shown in Figure 4.14.

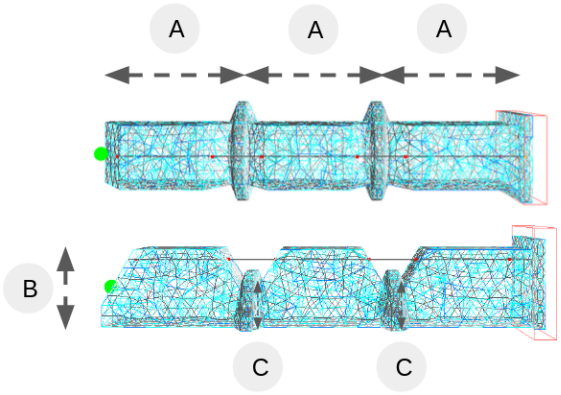

Figure 4.14: Soft Finger parametric design. Design parameters are A) Length, B) Height and C) Joint Height.

4.5.2 Parametric Design Captured by a Single Condensed FEM Model

For demonstrating the expressivity of the condensed FEM model relative to both geometrical and mechanical design variables, two condensed FEM models are learned from two different datasets with the following features:

- The **mechanical parameters dataset** considers 10000 Soft Finger states with varying Young's Modulus (range $[1000kPa, 10000kPa]$), Poisson Ratio (range $[0.400, 0.499]$), and cables displacement (range $[0.0mm, 20.0mm]$) sampled using a Scrambled Halton sequence.
- The **geometrical dimensions dataset** considers 50000 Soft Finger states with varying geometrical dimensions around a baseline design of the Soft Finger as shown in Figure 4.14. The sampling spaces are 38.0mm to 42.0mm for the Length, 20.0mm to 22.0mm for the Height, and 5.0mm to 8.0mm for the Joint Height, and 0mm to 20mm for the cable displacement. The open-source software Gmsh [GR09], enabling parametric CAD by scripting, is used for generating the meshes needed for simulation for each set of geometrical parameters.

This model integrates both actuation and design, rather than providing a separate control model for each design. Since the neural network shares information between different designs, the dataset size is smaller than if each design were considered separately.

Training results are displayed in Table.4.2 for both datasets. For learning the condensed FEM model of geometrical variations of the Soft Finger, a more complex strategy is adopted. A neural network of 5 fully connected layers with 800 nodes each is chosen.

The dataset features robot states that are highly different compared to the datasets previously encountered in section 4.3. Indeed, the geometrical dimension parameters have a greater

influence on the learned condensed FEM matrices. This results in under-sampling the training set. As an answer, a dropout strategy is applied for regularizing the network. The orders of magnitude of the final learning loss for the learned condensed FEM model with the geometrical dimensions dataset displayed in Table 4.2 is significantly higher than the learning losses previously encountered. However, the generalization remains adequate for position-based control applications. Design variations are harder to learn than actuation variations. Indeed, mechanical matrix values that link actuation to effectors, specifically in the direction where the geometrical dimensions (Length and Height) of the parametric soft finger evolve, are harder to learn due to strong non-linearities. However, these error values correspond to a small difference between predicted and simulated effector positions of around 3mm in the worst case met on the entire validation dataset.

Table 4.2: Test loss values for each parametric Soft Finger dataset for both G and F neural networks.

Dataset	G		F	
	Training Loss	Best Epoch	Final Loss	Best Epoch
Mechanical Parameters	3.82e-7	18420	5.04e-5	12530
Geometrical Dimensions	0.19	4400	0.13	270

Several designs are evaluated on the same trajectory using the condensed FEM model-based open loop inverse controller. The trajectory is described by 25 intermediate goals and consists of closing the Soft Finger on itself. Comparative trajectory errors between positions reached by the ground truth simulation and the corresponding positions obtained with the condensed FEM model are displayed in Table 4.3. Whatever the sampled design, it shows that both the trained condensed FEM models enable the control of different designs from their respective validation datasets. In studied examples, the maximal positioning error is about 1.2 mm which is approximately less than 1% of the total length of the parametric Soft Finger.

Table 4.3: Relative Euclidean positioning error for different values of geometrical and mechanical parameters.

Trajectory errors with mechanical parameters dataset:

Young Modulus (GPa)	Poisson Ratio	Error
5000	0.47	0.91%
1000	0.45	0.29%
7000	0.4	0.90%
3000	0.47	1.03%

Trajectory errors with geometrical dimensions dataset:

Length (mm)	Height (mm)	Joint Height (mm)	Error
40	20	6	1.03%
38.5	21.5	6	3.11%
40	20.5	7.5	7.02%
41	21	5.5	1.62%
39.5	20	7	5.10%

4.5.3 Parameters Optimization Using a Condensed FEM Model

Application to Calibration of the Condensed FEM Model of a Parametric Soft Finger

Mechanical parameters influence the behavior of the robot in simulation. Since the condensed FEM model is learned from simulations, one of the drawbacks is that it is learned for a fixed set of numerical parameters. Even if the simulation is initially calibrated using a physical prototype, discrepancies may arise during the manufacturing of other similar prototypes or due to silicone degradation over time. In such cases, it would be necessary to re-calibrate the simulation and develop a new condensed FEM model for effective control. In the example of the Soft Finger, the appearance of air bubbles or an incorrect dosage of silicone during the molding process can lead to different corresponding mechanical parameters.

For dealing with this issue, the extended condensed FEM model is learned from the mechanical parameters dataset. The aim is to address learning the condensed FEM model with zero-shot transfer to reality.

Once the condensed FEM model is learned, it is calibrated using data collected from the physical robot. The general workflow to calibrate a condensed FEM model of any soft robot is as follows. First, positions of the effector(s) are measured for several actuation displacement states δ_a. Increasing the number of actuation states enables better convergence. Then, a fitness function $O_{\text{cal}}(\mathbf{p})$, with $\mathbf{p}$ the design parameters, computing the difference between the relative position of the effectors δ_e^* evaluated on the physical prototype and those estimated by the condensed FEM model $\widetilde{\delta_e}$ is built as:

$$O_{\text{cal}}(\mathbf{p}) = \sum_{\delta_a} |\widetilde{\delta_e}(\delta_a, \mathbf{p}) - \delta_e^*(\delta_a)| \tag{4.13}$$

To build a differentiable link with the design parameters $\mathbf{p}$, δ_e is expressed from the predicted mechanical matrices. Using the the expression of the coupling between W and δ without contacts, it gives:

$$\widetilde{\delta_e}(\delta_a, \mathbf{p}) = \widetilde{\mathsf{W}}_{ea}(\delta_a, \mathbf{p})\widetilde{\mathsf{W}}_{aa}^{-1}(\delta_a, \mathbf{p})(\delta_a - \widetilde{\delta_a^{free}}) + \widetilde{\delta_e^{free}} \tag{4.14}$$

The optimal design parameters $\mathbf{p}$ are obtained by minimizing O_{cal} through gradient descent.

The Poisson's Ratio is optimized because the Soft Finger is controlled through displacement. In this control mode, the Poisson's Ratio is crucial for accurately modeling how the material deforms laterally in response to an applied longitudinal strain. On the other hand, if we were controlling the Soft Finger by applying force, the Young's Modulus would be more important, as it directly affects the stiffness and how the material resists deformation under load. Therefore, optimizing the Poisson's Ratio is more relevant for displacement control, while the Young's Modulus would be more critical for force control. As the simulation to reality transfer of the Soft Finger has already been validated in previous work, simulation results are taken as ground truth. The states δ_e^* of the robot are sampled in simulation for 3 given cable displacement values δ_a. Poisson's Ratio ground truth value is taken equal to 0.47. Then, the fitness function from Equation 4.13 is minimized for retrieving the corresponding parameters. Obtained convergence histories are displayed in Fig 4.15 for three different initializations of the mechanical parameters.

In each case, the algorithm succeeds in finding a good Poisson Ratio for reducing the distance error on effector location. The optimization converges in a few ms. Changing for a servoing in force exerted on the cables instead of cable displacements would require taking into account the Young's Modulus during the optimization.

With this experiment, it is demonstrated that it is possible to learn a single condensed FEM model that can be directly calibrated to a soft robot. By relying on the differentiability of the learned condensed FEM model, the calibration method only takes a few seconds on

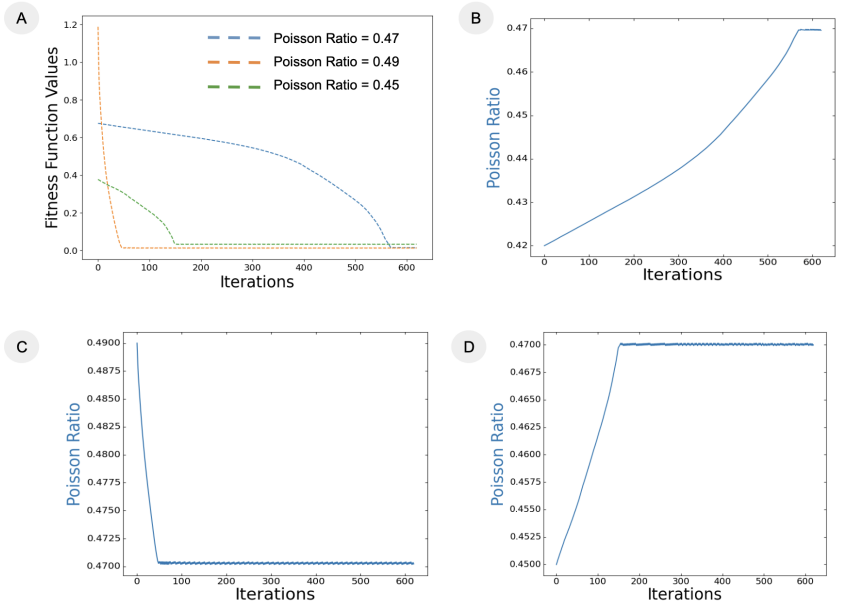

Figure 4.15: Optimization history for the calibration of the Soft Finger robot. Results have been obtained with an initial learning rate of 0.01 used conjointly with an adaptive scheme as described in section 4.2.3. Both fitness function (A) and mechanical parameters histories (B,C,D) across optimization iterations are displayed for different initial guesses of the mechanical parameters. Initial mechanical parameters are a fixed Young Modulus of 3000 GPa as well as B) Poisson Ratio = 0.47, C) Poisson Ratio = 0.49, and D) Poisson Ratio = 0.45.

a pre-learned model, compared to other models relying on non-gradient solvers requiring a lot of simulations. Moreover, including design parameters such as geometrical dimensions in addition to mechanical parameters would enable to calibrate the condensed FEM model to design variations that may appear during the manufacturing process.

Application to Design Optimization of a Parametric Soft Finger

For this experiment, the parametric Soft Finger design introduced in Figure 4.14 is optimized successively for its dexterity and precision grasping.

Design optimization objectives are formulated from the predicted mechanical matrices. The two considered objectives are both illustrated in Figure 4.16.

The first fitness function O_{dext} characterizes the kinematics of the parametric Soft Finger, i.e. the reached bending angle for a fixed cable displacement δ_a of 10mm. Minimizing this function is equivalent to maximizing the workspace of the robot. O_{dext} is expressed as follows:

$$\begin{aligned} O_{\text{dext}}(\mathbf{p}) &= |\alpha^{\max} - \alpha(\delta_a, \mathbf{p})| \\ \alpha(\delta_a, \mathbf{p}) &= Arccos\left(\frac{\widetilde{\delta_e^z}(\delta_a, \mathbf{p})}{|\delta_e(\mathbf{0}, \mathbf{p})|}\right) \end{aligned} \tag{4.15}$$

where α is the function giving the bending angle of the parametric Soft Finger in the z-plane and $\alpha^{\max} = 1.57$ rad is the maximum reached angle.

The second fitness function O_{str} measures the contact force generated at the tip of the

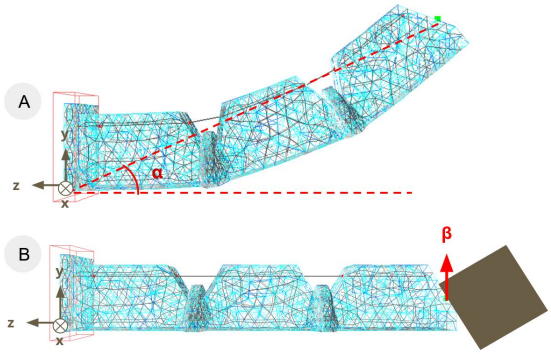

Figure 4.16: Illustration of design objectives for the parametric Soft Finger. A) Bending angle $\alpha(\delta_a, \mathbf{p})$ reached for a fixed cable displacement δ_a. B) Contact force $\beta(\delta_a, \mathbf{p})$ generated for a fixed cable displacement δ_a.

Soft Finger for a fixed cable displacement δ_a of 10mm. Minimizing this function amounts to maximizing the performance of using the parametric Soft Finger for precision grasping. Again, for building a differentiable link with the design parameters $\mathbf{p}$, the contact force λ_c is expressed from the predicted mechanical matrices.

With the assumption of a fixed object i.e. $\delta_c = 0$, and considering the tip effector as a contact point, the system expressed from both expressions of the coupling of δ_a and δ_c with the compliance projected in the constraint space, enables to obtain:

$$\lambda_c(\mathbf{p}, \delta_a) = B(\mathbf{p}, \delta_a)^{-1}[\tilde{\mathrm{W}}_{ca}\tilde{\mathrm{W}}_{aa}^{-1}(\delta_a - \tilde{\delta}_a^{\text{free}}) + \tilde{\delta}_c^{\text{free}}] \tag{4.16}$$

with $B(\mathbf{p}, \delta_a) = \tilde{\mathrm{W}}_{ca}\tilde{\mathrm{W}}_{aa}^{-1}\tilde{\mathrm{W}}_{ac} - \tilde{\mathrm{W}}_{cc}$

Finally, the fitness function O_{str} is expressed as follows:

$$\begin{aligned} O_{\text{str}}(\mathbf{p}) &= |\beta^{max} - \beta(\delta_a, \mathbf{p})| \\ \beta(\delta_a, \mathbf{p}) &= |\lambda_c(\mathbf{p}, \delta_a)^y| \end{aligned} \tag{4.17}$$

where β is the function giving the force generated at the tip of the parametric Soft Finger on the y-axis and $\beta^{max} = 10000$ N is an unreachable force.

The two fitness functions are written from the mechanical matrices learned with the condensed FEM model. These matrices connect forces to displacements in crucial areas of the design, while accounting for its mechanical structure. This allows for the creation of fitness functions derived from this compact representation. Design optimization can then benefit from this pre-learned model to quickly travel the design space. In the following, results are first generated and analyzed separately for each fitness function. Then, it is shown how a multi-objective scheme can be implemented based on the condensed FEM modeling.

Single objective optimization: As the considered fitness functions are highly non-convex, a simple gradient descent optimizer may fall in a local optimum. To avoid this behavior, a grid search is first performed to determine a good starting point for the optimization. As the design space is of only three dimensions, the computation is a simple matter of seconds with the condensed FEM model. A total of 600 designs are evaluated for both fitness functions. In high dimensional cases, an alternative for avoiding the dependency on the initialization would have been to use stochastic gradient descent instead of a simple gradient descent scheme as a solver.

For each fitness function, the objective function landscape is computed. For this purpose, a regular grid sampling strategy is applied around the chosen initialization starter design. Landscapes computed for both fitness functions are displayed in Figure 4.17. They show that the length has a higher impact on both objectives than two other design variables.

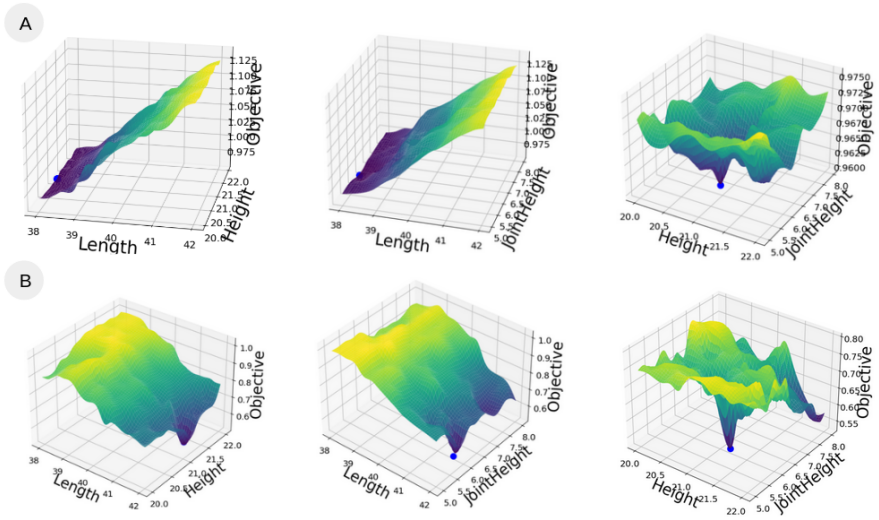

Figure 4.17: Fitness landscapes around the initialized design selected with grid search strategy, for A) dexterity fitness function (Length = 38.0mm, Height = 20.57mm, Joint Height = 6.28mm) and B) strength fitness function (Length = 42mm, Height = 22mm, Joint Height = 8mm). The best design after optimization is shown as a blue dot.

Gradient descent is performed from the initial design. Regarding dexterity, the best design (Length=38.0 mm, Height=20.6mm, Joint Height=6.3mm) privileged a shorter length. In the case of strength optimization, a longer length helps (Length=42.0 mm, Height=22.0mm, Joint Height=7.9mm) to generate more force on the object thanks to the lever arm effect. The best design regarding contact forces generated at the tip is also a Soft Finger with more material, enabling it to rigidify better around the contact location.

Multi-objective optimization: In this experiment, it is highlighted how to determine the optimal design of a Soft Finger regarding several metrics computed using the condensed FEM model. Figure 4.18 shows the fitness functions values for 600 Soft Finger designs sampled with a grid search strategy. For having comparative values between the two losses, they are both normalized using $O_i^{\text{norm}}(\mathbf{p}) = \frac{O_i(\mathbf{p}) - \min_{\mathbf{p}*}(O_i(\mathbf{p}*))}{\max_{\mathbf{p}*}(O_i(\mathbf{p}*)) - \min_{\mathbf{p}*}(O_i(\mathbf{p}*))}$ where $\mathbf{p}*$ are design parameters encountered during initial grid search.

A Pareto front is clearly visible, showing that the two considered metrics are antagonistic. A compromise has therefore to be found. This can be done using the gradients calculated via the condensed model, by formulating a single fitness function as an aggregation of the considered fitness functions:

$$O_{\text{mult}}(\mathbf{p}) = \gamma_1 O_{\text{dext}}^{\text{norm}}(\mathbf{p}) + \gamma_2 O_{\text{str}}^{\text{norm}}(\mathbf{p}) \tag{4.18}$$

where both γ_1 and γ_1 are user-chosen weighting factors between 0 and 1 for selecting the priority to give to each fitness function. This kind of approach enables to generate the optimal design of a Soft Robot given user-chosen requirements. Setting these weighting factors is usually

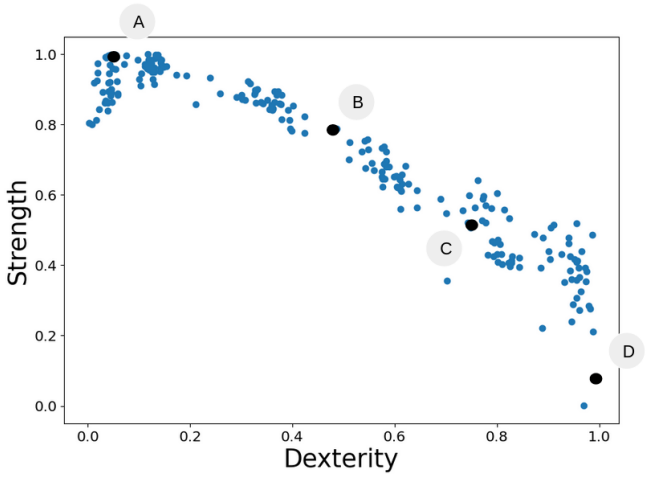

Figure 4.18: Pareto Front for the design optimization of the Soft Finger regarding Dexterity and Strength metrics and generated using a Grid Search strategy. Design parameters for several design sampled: A) Length = 38.0mm, Height = 21.6mm, Joint Height = 5.0mm, B) Length = 40.4mm, Height = 20.0mm, Joint Height = 5.6mm, C) Length = 40.2mm, Height = 22.0mm, Joint Height = 7.4mm, D) Length = 42.0mm, Height = 22.0mm, Joint Height = 8.0mm

challenging. They are commonly chosen by trial and error, each trial requiring many simulations to be performed for evaluating fitness functions at each iteration of the optimization process. Using a pre-learned condensed FEM model enables to discard the computation time related to simulation.

4.6 Discussion and Conclusion

In this chapter, a method for modeling a soft robot in a compact way has been introduced. The proposed representation takes the form of learned mechanical quantities. The particularity of this method is that it provides a link between actuation, effector, contacts and design parameters. Although we consider entirely soft robots, the current condensed formalism enables to consider robots with both rigid and soft parts. We summarize the main contributions, limits, and future avenues to explore below.

We showed that this method enables to control soft robots with quadratic optimisation. Although we only consider pneumatic and cable actuators, nothing prevents the method to be used with a wider variety of actuators. The method can also handle contacts, which are considered as constraints in the same way as actuators and effectors. We show how to formulate inverse control problems both for single and coupled soft robots. In the current framework, we made the assumption that the number and the location of the contact points on the robot are fixed. This assumption can be overcome by considering multi-point contact. The approach involves accounting for a variable number of contact points without prior knowledge of their quantity and locations. This could be done using models that can predict matrices of variable size, like Transformers [Vas+23] or GNN [Zho+21b], or by increasing the robot's state to include many potential contact points.

Secondly, embedding of the condensed FEM model for open-loop and closed-loop control is demonstrated on a pneumatic soft robot. Once learned, we demonstrate that the model

can run on a very simple embedded controller. The embedded control could be applied to other robots. The more complex the robot, the more speed-up can be obtained. So far, only quasi-static assumption has been considered. This assumption could be overcome by considering learning the dynamics in order to take into account speed and position information. Neural architectures developed for sequential data, such as LSTMs [SM19] or Transformers [Vas+23], are good candidates for this task. The condensed FEM model extension to dynamics may unlock applications in using more robust controllers such as Model Predictive Control [GPM89].

In this chapter, we show that the condensed FEM model formalism can be used to jointly learn the design and control of a soft robot. Applications in the direct calibration of the condensed FEM model and design optimization of soft robots are demonstrated in the example of a parametric Soft Finger. So far, our model has been demonstrated with few design parameters. The use of this type of model for more complex parameterization, in terms of number of parameters or expressivity, is an interesting prospect involving in-depth rethinking of network scaling and sampling strategies. Moreover, this type of model could provide ideal conditions for the development of co-optimization techniques. For the moment, the proposed control is low-level, based on inverse robot modeling and Quadratic Programming optimization tools. However, it is possible to imagine using the learned condensed FEM model as a surrogate model in an RL approach, where the control strategy takes into account design information to both solve high-level control tasks and optimize the robot design.

Finally, to overcome the assumption on the sampling process, one envisioned extension is to use online learning techniques. Indeed, one of the drawbacks of the method is its reliability on a huge amount of data collected offline. However, many robot states may be redundant or not carry relevant information for the task to be performed. This would enable a case-dependent exploration of the actuation space. This is particularly necessary when there are several points of contact, leading to a much larger sampling space. Physics Informed Neural Network [RPK19] could be implemented to facilitate learning generalization. It necessitates formulating the problem in another way for integrating first-order derivatives for learned variables.

Chapter 5

Towards a General Framework for the Emergence of Embodied Intelligence in Soft Robots

5.1 Introduction and Chapter Overview

In the previous chapter, we demonstrated the learning of a condensed FEM model, with design parameters as inputs, which allows it to function as a differentiable surrogate for the simulation regarding both soft robot design parameters and state. This surrogate is quick to evaluate while maintaining precision comparable to FEM-based simulation. However, there are still several challenges to address when it comes to co-evolving the design and controller of soft robots. First, how can we develop generic parametric designs that accurately describe diverse morphologies, rather than being restricted to variations around a designer-provided baseline? These design parameterizations should also include manufacturing constraints to ensure applicability in real-world scenarios. Moreover, we seek parametric representations that are highly expressive, capable of generating a wide variety of shapes, yet rely on a minimal number of parameters. Indeed, the design parametric space should remain small to ensure that co-optimization remains tractable. Second, how can we optimize soft robot controllers for high-level tasks without having to decompose these tasks into a set of mechanical requirements? These questions are further constrained by available computational time and power, which must be addressed in any proposed solutions.

This chapter presents preliminary and ongoing work aimed at leveraging this surrogate to develop a generic framework for the co-evolution of soft robot design and control. The work presented here has not yet been published in a scientific journal or conference. This chapter is shorter than the others because it is limited to a presentation of ongoing work. It also discusses future works based on the results obtained in this thesis.

In Section 5.2, we explore how to build design parameterizations that are both more expressive and less influenced by human subjectivity than the parametric representations introduced in chapter 3. This exploration includes investigating new types of parameterizations based on generative design and constructing a library of modular soft components.

Section 5.3 then introduces methods for efficiently learning controllers for high-level tasks

using reinforcement learning and the condensed FEM model as a simulator. One of the problems with reinforcement learning-based co-optimization frameworks is the computational time required to jointly explore design and control spaces. On one hand, it is necessary to have expressive design spaces with few parameters for exploration. On the other hand, the simulations needed to evaluate the performance of a design and a controller are costly and generally the bottleneck in this type of framework. We analyze strategies to reduce the amount of data required to train the condensed FEM model and extend it to account for dynamics. Finally, we discuss considerations for incorporating moving contact points in the condensed FEM model to better explore interactions between soft robots and their environments.

5.2 Toward more Expressive Design Parameterizations

In the previous works introduced in this thesis, parametric representation of soft robot geometries were considered as a design space for computational design. This type of parameterization is constructed on a case-by-case basis for a specific family of designs, relying heavily on the expertise and intuition of the designer. Consequently, each new design requires the development of a specific parametric model, limiting the ability to generate morphologies that deviate significantly from the baseline envisioned by the designer. However, parametric descriptions have the advantage of easily incorporating manufacturability constraints, facilitating the transfer to real-world applications. Moreover, the number of design variables is kept small, making it practical for use with simple numerical optimization algorithms.

In the literature discussed in the Background section 2.3.2, methods that enable the generation of significantly different morphologies rely on very abstract representations of the design space, such as grid of voxels with varying material properties. These approaches enable for the creation of designs that diverge greatly from what intuition might suggest, but they present challenges when transitioning from optimized numerical designs to real-world implementations.

Two different approaches are presented to explore these design spaces.

The first approach, described in section 5.2.1, relies on the use of Growing Cellular Automata to construct an implicit representation of a family of designs. The idea is to provide examples of successful designs and automatically create a parameterization of the design space based on these examples.

The second approach, detailed in section 5.2.2, takes a modular perspective. The goal is to construct the body of a soft robot by juxtaposing soft modules with known mechanical behaviors. Design optimization can then be performed in a grid space where each squared-element is populated with easily manufacturable modules from this library. This approach ensures manufacturability of the optimized robot body, as opposed to using abstract spaces that may not guarantee it, such as those involving discontinuous material distributions along the soft body or multiple pneumatic cavities.

5.2.1 Generative Design of Soft Robots using Growing Cellular Automata

Generative models [RNA22] [Bon+22] are currently popular in the machine learning community. These models aim to understand and replicate the underlying data distribution of a given dataset.

The advantage of these methods is their ability to achieve a compact parameterization of a robot's geometry using a database of functional robot geometries. Provided access to this database, such parameterizations offer a richer representation without the need for explicit, case-by-case parameterizations created by the designer, as explored in Chapter 3.

This approach is illustrated on the example of the parameterization of the shape of a pneumatic cavity. Applying the method proposed in Chapter 3 involves manually constructing

complex design parameterizations by defining control points on the cavity's surface, as illustrated with the optimization of the soft manipulator using self-contacts in Chapter 3. These control points are treated as design parameters. To accommodate a wide variety of cavity shapes, it is necessary to include a substantial number of points in the parameterization, which significantly expands the design space. Furthermore, enforcing manufacturability constraints proves to be challenging due to the complex inter dependencies among the design parameters. Our proposed approach involves learning a generative model from different cavity geometries, which yields a lower-dimensional design space capable of generating complex shapes. This is illustrated in Figure 5.1. The idea is to rely on the ability of the model to infer constraints on manufacturability and a complex design space.

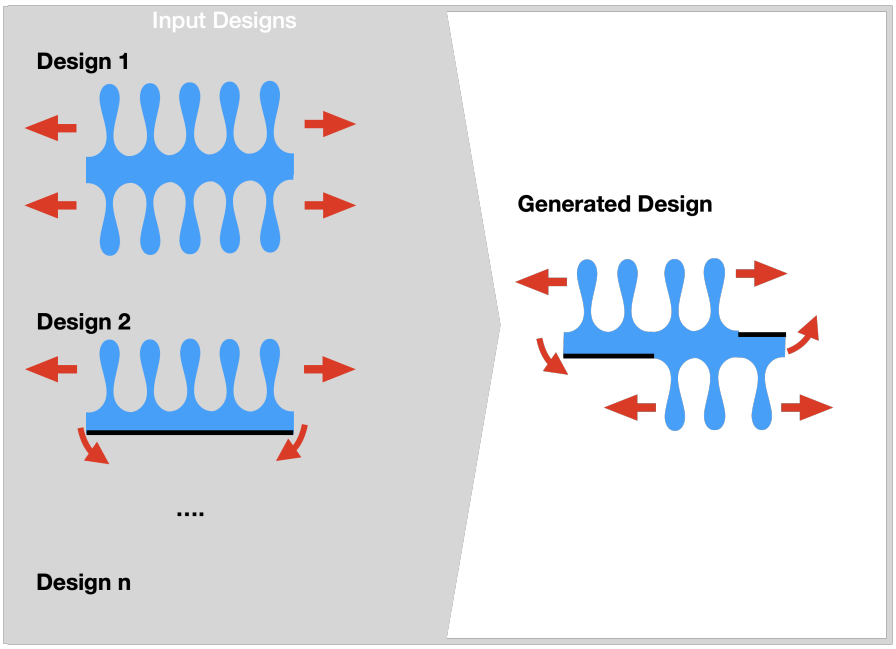

Figure 5.1: Illustration of the proposed generative framework for the generation of pneumatic cavities. The generative model is trained using a library of pneumatic actuators (left). In the training dataset, Design 1 tends to elongate, while Design 2 is a Pneunet-style design [Mos+14] that tends to bend. By utilizing the generative design framework, we can infer from the distribution of input data to generate an optimal shape tailored for the desired deformation (right).

Recently, we have started to see examples of generative models emerging in the field of soft robotics design [Wan+23] [CWY24]. Both of these works rely on machine learning models that are computationally intensive to train. However they still lack to integrate manufacturing constraints, and the designer needs to post-process the results.

Design Description from Realistic Soft Robot Design Sub-Parts

The idea is to develop a framework that generates realistic patches of soft robots using low-dimensional networks. The design of the robot is then described implicitly through a collection of patches extracted from CAD models of working soft robot designs. A patch is then a sub-part of a CAD model. Specifically, we consider small cubes composed of cubic cells, each with an associated continuous state ranging from 0 (representing an empty cell) to 1 (representing a cell

filled with material). A complete soft robot design is then described by an association of patches.

Growing Shapes from a Neural Cellular Automata

For this purpose, we implemented a framework based on Neural Cellular Automata (NCA) [Mor+20b], an extension of traditional Cellular Automata models. These models operate on a grid of cells that iteratively evolve through a set of local rules based on their current state and the states of neighboring cells. Unlike fixed-rule models, NCA dynamically learns and adapts these rules from provided data.

When growing a Cellular Automata, the design space is discretized into a Euclidean grid of elements, typically a voxel grid, where each element is associated with continuous state from 0 to 1. Growing a Cellular Automata usually begins with a single cell, whose continuous state can be chosen or initialized at random, and iteratively updates local cells based on their neighbors' states. NCA takes as input the states of the current cell and its neighbors, encoded as hidden channels, and provides the cell's update. The NCA model is trained by initializing an empty grid with a single living cell, running the NCA for a certain number of steps, comparing the output with the target design, and then backpropagating the mean square error loss. An illustration of an NCA update step is provided in Fig. 5.2.

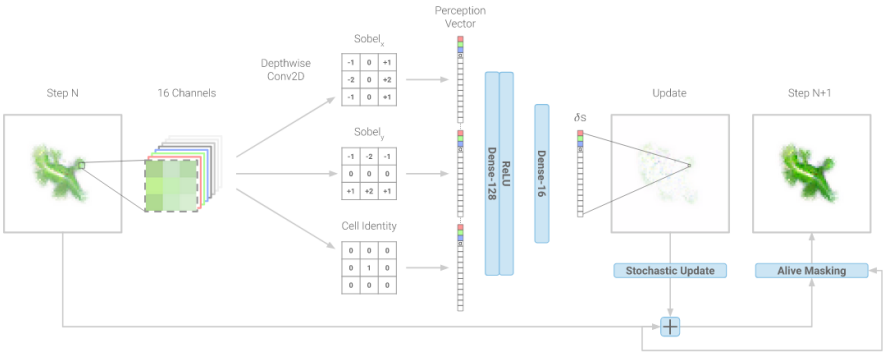

Figure 5.2: Illustration of an update step for growing a lizard shape using a NCA from [Mor+20b]

Handling 3D Resolution for Design Applications

As most of the examples in the machine learning community were about generating 2D images, we faced several issues to scale NCA to 3D. Indeed, the computational cost is directly related to the number of element in the grid, as NCA involve convolution operations on the entire design space. Furthermore, as we are interested in generating detailed 3D design, this is an even bigger constraint. To address this issue, we adopt a multi-resolution strategy [LYH11]. We begin with an initial grid scale and apply the NCA for several steps, then up-sample the grid to a higher scale by dividing cells and applying the NCA again. This process is repeated until the target scale is achieved. During training, we compare the final scale with the target or each intermediate scale with scaled versions of the target, providing flexibility to use any intermediate scale as needed. We evaluate and demonstrate our method on the example of the iterative reconstruction of a high resolution 2D picture, as shown in Figure 5.3. This examples that we are able to handle large design spaces compared to the small resolution cases proposed in [Mor+20b].

We demonstrated our framework on regenerating 3D designs from examples (see Fig 5.4.A). We also show that we can learn implicit parameterization of local patches of provided mechanical designs (see Fig 5.4.B).

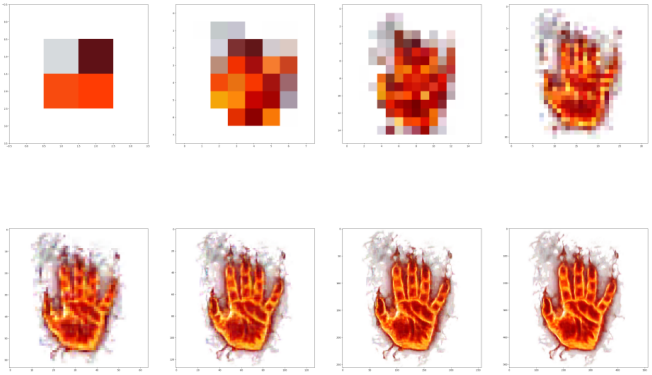

Figure 5.3: Iterative reconstruction of a HD image using our generative framework with multi-resolution NCA. Several intermediate reconstruction states are displayed, from top left to bottom right.

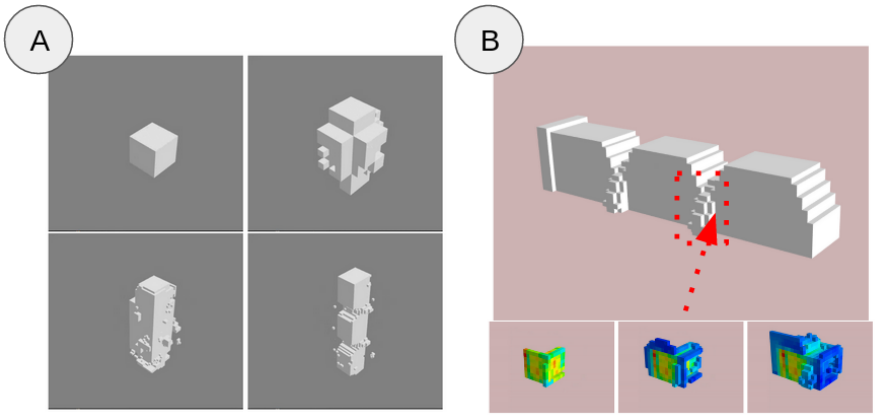

Figure 5.4: Illustration of several results obtained with our generative model. A) Iterative reconstruction of a Soft Finger design. In this experiment, the NCA is trained to reconstruct a Soft Finger design. Several intermediate shapes obtained during the growth process are displayed. B) Implicit parameterization of local patches guided by example, with the example of the iterative growth of one patch from the Soft Finger. Voxel colors correspond to continuous values between 0 and 1. In this experiment, the NCA is trained to reconstruct several 3D patches of the Soft Finger design. Given the initial values of one of the patches (see picture on the bottom left), we show that we we are able to reconstruct this patch through growth.

Reducing the Design Parameter Space

Training a network to learn an implicit parameterization for a family of designs involves training it to reconstruct multiple designs from a dataset. Once the NCA is trained, the reconstruction of a design depends solely on the initial seed values, which are associated with the value of voxels of the initial patch provided to the algorithm. For example, we use 5x5x5 patches as initial seeds to reconstruct larger 16x16x16 patches. Although this result in still a large design space of 125 parameters, it enables to compact an entire shape. To reduce the parameter space, we employ a Variational Autoencoder (VAE) [KW19] to create a compact parameterization of these

seeds. The VAE is trained to encode the initial seeds into a smaller, more manageable parameter space, allowing us to navigate and reconstruct designs efficiently based on these parameters.

Future Work: Adding Physical Constraints

Once trained, the network is expected to interpolate between the dataset designs by selecting various continuous values within the compact design space. However, the design domain is constrained only in several points of this parameterized space, making it challenging to generate viable designs on all the domain. As a result, many generated designs have disconnected parts or mechanically unfeasible structures. This occurs because convergence is enforced only on the known data at the parameter bounds. This limitation is a common challenge in generative design.

To address this issue, the conventional approach is to include numerous intermediate designs in the training dataset. However, this significantly increases the cost of training the generative model and can potentially reduce its expressiveness, as it may become overly constrained. We consider two other approaches. The first one is to add physical constraint in the training loss of the generative network. For instance, several points of the parametric design space may be randomly sampled during the training. After growth, they may be evaluated in simulation and physical constraints may be computed as a regularization term for the training loss of the network. Another idea is to use the imperfect learned implicit parameterization as a starting point to generate new plausible design patches, incorporating physical constraints during growth. For instance, an initial robot design can be described as patches that can be assembled and combined using evolutionary algorithms for sampling in the parametric design space, and topology optimization (see the Background section 2.3.2) for enforcing mechanical constraints. Topology optimization algorithms are gradient-based algorithms that are highly dependant on the initial conditions of the design problem (load and boundary conditions, initialization of design variables). Recent work [Pin+24] shows the relevance of including variability in the initialization of topology optimization design problem to generate different soft robot shapes. Compared to this work, our approach would enable to include constraints on the general shape of the generated robot designs.

5.2.2 A Library of Modular Components for Soft Robot Design

Soft robotics misses modular components that can be reused for having desired motions. Computational design optimization contributions to soft robotics mostly consider specific design parameterization to the problem at hand. On the other hand, classic rigid robots are usually designed through a set of degrees of freedom requirements, and the mapping between these requirements and the optimization of of generic geometrical parameters has been well explored [Pet08; Ha+17].

In the Background section on computational design for soft robotics 2.3.2, we showed that the most promising frameworks for generating totally new morphologies for soft robots rely on modular representations of soft robots. For instance, design spaces are often represented as aggregation of voxels able to expand and contract [Che+14; Bha+22; CBS23], but the optimized designs are not manufacturable. These methods also rely on overly simplified simulation, avoiding numerically optimized designs to transfer well to physical prototypes.

The idea is to build a library of easily manufacturable parametric modular components with characterized behaviours in term of degrees of freedoms. Problematic related to simulation computation time may be addressed by leveraging the condensed FEM model to learn reduced and differentiable representations of these parametric modular components.

For this purpose, we take inspiration from what is done in the field of compliant mechanisms, where many people have worked at designing and characterizing simple mechanisms made of soft materials to enable targeted degrees of freedoms [HMO13]. They take geometrical considerations and jointly use different soft materials with different mechanical behaviours. We have built a library of geometrical parameterizations and simulation for several different compliant mechanisms. Examples of simulation for two compliant mechanisms from the library are displayed in Figure 5.5.

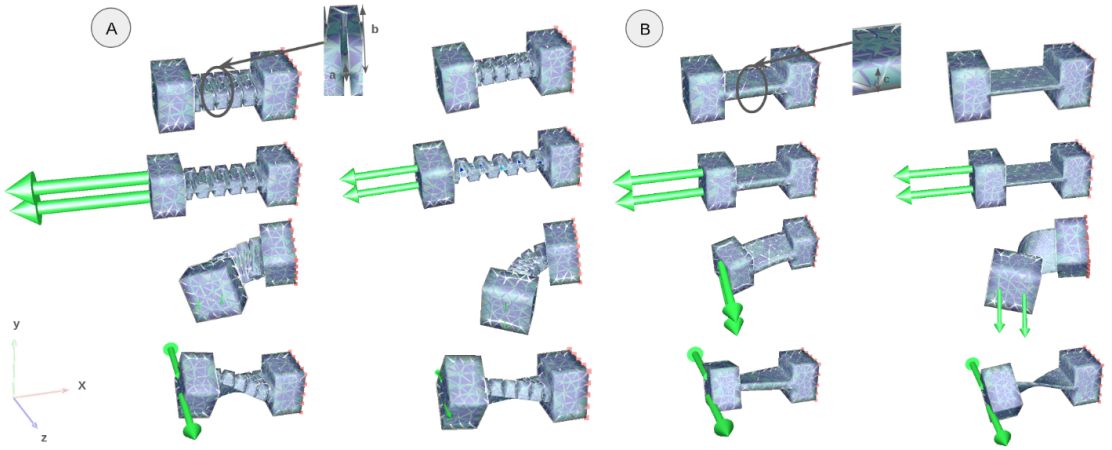

Figure 5.5: Example of simulation of two compliant mechanisms from the component library. For each of them, different loading cases are provided. The size of the green arrows is directly proportional to the force exerted at their base. A) Accordion pattern mechanism. Reducing the dimension of (a) the guide and (b) the width of the accordion pattern from the right column enable to obtain greater translation along the x-axis but less resistance to simulations along y and z-axes. B) Single leaf spring mechanism. Reducing the dimension of (c) of the single leaf spring mechanism from the right column enables to facilitate both the twisting and the torsion motions.

In our approach, we aim to avoid using multiple materials, which complicates the manufacturing process. Instead, we plan to use fused deposition modeling 3D printing parameters as design variables, such as material density and fiber orientation, to smoothly and locally alter the mechanical properties of printed components. We utilize the IceSL slicer [1], which allows for defining continuous infill fiber orientations using three simple parameters. This results in components with heterogeneous properties that are easily printable. Examples of several components printed with different fiber orientations and material density are provided in Figure 5.6.

Simulation models for soft robots exhibiting anisotropic behavior have already been developed using the finite element method (FEM) in *SOFA* [Van+20; VGD21]. These models rely on homogenization methods to calibrate the simulation, as simulating the detailed full mesh of the structure would be computationally intensive and would require handling numerous self-contacts. Currently, there is no inverse model yet directly linking the infill parameters and the numerical parameters of the printed material. This inverse model can be built by measuring the force/displacement ratios for several design variations of each compliant mechanism, calibrating a simulation scene to each design, and learning condensed FEM models from data extracted

[1] *IceSL*: https://icesl.loria.fr/about/

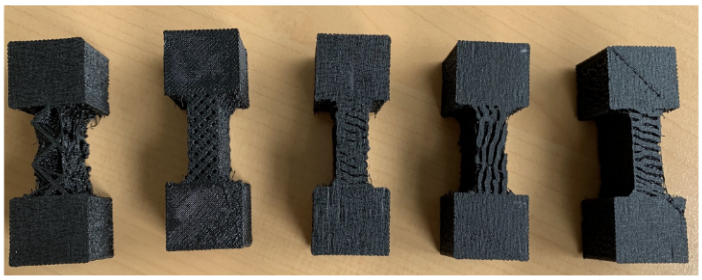

Figure 5.6: Example of several single leaf spring patterns obtained for different 3D printing parameters.

from the simulation. Acquiring the data necessary for this task requires to build a test bench for evaluating several variations for each compliant mechanisms.

Once this library of condensed FEM models is established, we can develop optimization frameworks to fine-tune the placement and geometric parameters of these models based on user-defined objectives and constraints. We envision the design space as a grid where each cell is populated with compliant mechanisms exhibiting varying infill and geometrical parameters properties. Both this design problem and the illustrative example of a leg constructed from several compliant mechanisms of our library are shown in Figure 5.7.

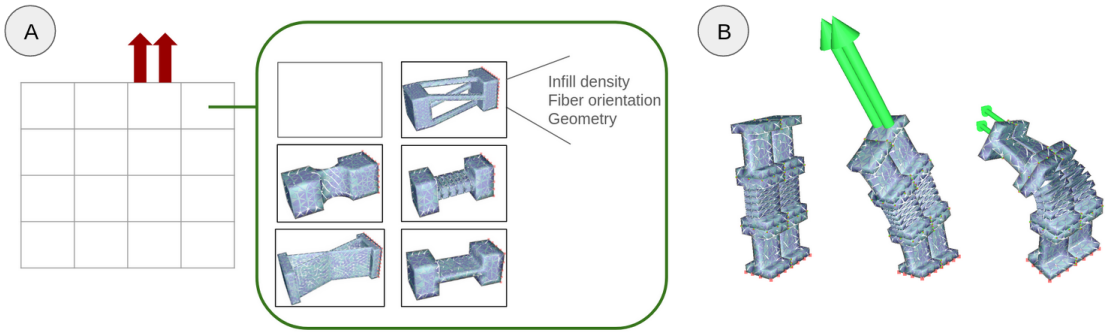

Figure 5.7: A) The proposed design space. Given a task and actuator locations, we want to optimize the placement and parameters of compliant mechanisms for each voxel of the grid. B) Example of simulation of a soft leg built from several accordion pattern and single leaf spring mechanisms. Different loading cases are provided. The size of the green arrows is directly proportional to the force exerted at their base. In this example, bilateral constraints are applied to attach the different modules together. In the future, we plan to speed up the simulation by building an equivalent condensed FEM model of the leg from the condensed FEM models of its different modules.

5.3 Toward a Generic Framework for Soft Robot Control and Design Co-Optimization

In this thesis, we primarily focus on optimizing robotic designs for tasks that have been broken down into multiple simple control tasks. These can be solved using model-based approaches, either directly or inversely, with a one-time-step scheme. It requires specific user intervention to decompose a complex task into a set of sub-tasks or mechanical constraints. For instance, a grasping task can be divided into several sub-tasks such as positioning the gripper at the level of the object, closing it on the object, or maintaining sufficient force on the object during manipulation. Decomposing tasks into sub-tasks is not straightforward, especially when considering complex tasks, and taking into account interactions with the environment make it even more challenging. Additionally, this decomposition into sub-tasks can be design-dependent. For example, in the case of a gripper, the best grasping position may vary depending on the shape of the gripper.

We choose to use reinforcement learning for exploring higher-level tasks, as detailed in Section 2.2.2 of the Background. However, a challenge with these methods is that the simulations required to evaluate the performance of a design and controller are costly.

To mitigate this problem, the idea is to use a surrogate to replace the simulation. The learned condensed FEM model is a good candidate because it enables the characterization of the robot design as well as the robot actuation and contact states. However, there are still some hurdles to overcome to use it smoothly, which are described below.

Learned Condensed FEM Model extended to Dynamics

The learned condensed FEM model needs to be extended to handle dynamics of soft bodies. To address this, we propose replacing the MLP used for learning the quasi-static condensed FEM model with a Long Short-Term Memory (LSTM) network, a type of recurrent neural network architecture designed to capture and retain long-term dependencies in sequential data [SM19].

For our first tests, we considered the Tetrapod, a four-legged robot, for trajectory control applications. The Tetrapod is powered by four servomotors, each located at the base of its TPU 3D-printed legs. Picture of the robot is available in Figure 5.8.A.

The motivation behind the choice of this platform is twofold. First, we demonstrated that using this platform in manipulation scenarios induces dynamic effects that influence the robot's behavior and must be considered when exploring an optimal control policy. Two RL policies were learned to reach the farthest point on the right side of the Tetrapod's workspace, as shown in Figure 5.8.B. The first policy does not involve carrying an object, while the second policy involves carrying an object modeled as a 1 kg mass at the tip of the robot. These results highlight the importance of considering the Tetrapod's dynamics in manipulation tasks, especially when dynamic effects are significant, as in the second scenario. The RL approach results in a more effective strategy compared to using simple quadratic programming, which would have directed the robot straight to the right without accounting for momentum. Secondly, the modular design of the Tetrapod's platform allows for easy replacement of its legs. The legs can be 3D printed with specific pre-curvatures, which significantly impact the Tetrapod's performance, including its stability under external forces and its operational workspace. This makes the platform an excellent candidate for design optimization applications. In the future, we plan to apply this model both for embedded control of the Tetrapod dynamics and design optimization.

Initial results for learning the dynamics of the Tetrapod robot show that we can successfully use it to follow a complex trajectory, as displayed in in Figure 5.8.C. However, this model needs to be scaled to larger datasets, and its generalization ability to unseen trajectories still needs to

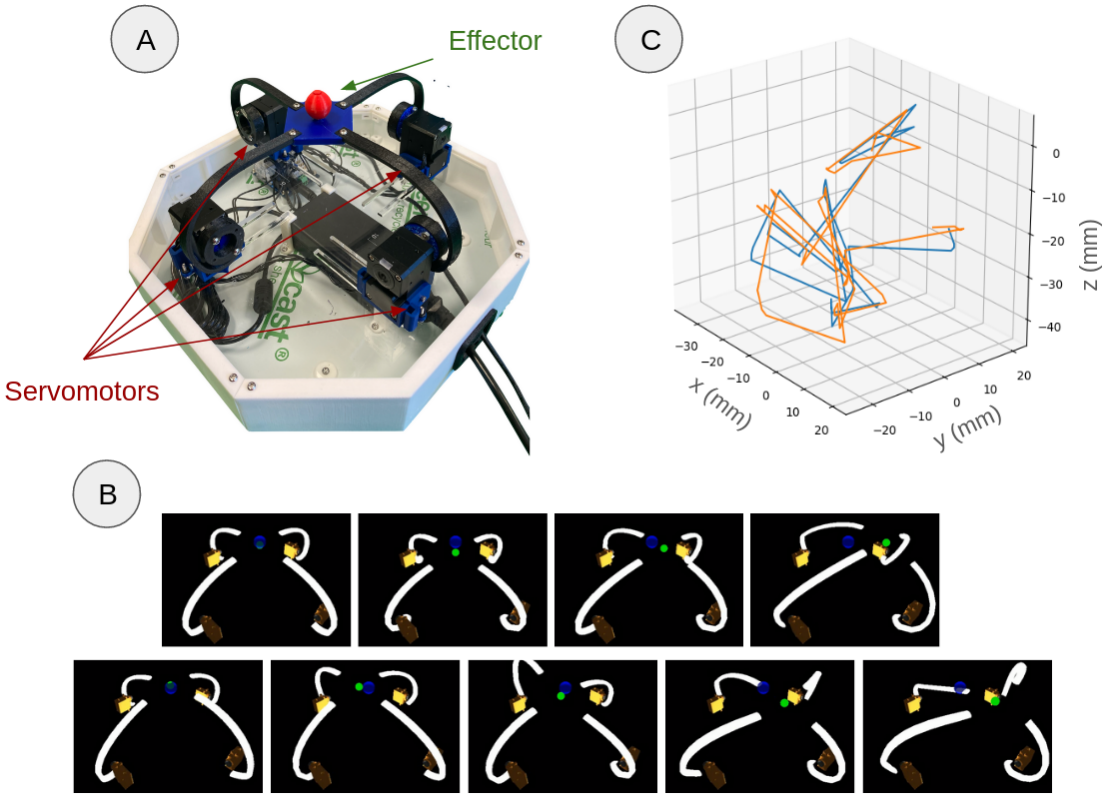

Figure 5.8: A) Picture of the Tetrapod robot. B) The two learned RL policies for reaching the farthest point on the right side of the Tetrapod's workspace while not carrying an object (top) or while carrying an object modeled as a 1 kg mass at the tip of the robot (bottom). C) Results for learning the dynamics of the Tetrapod: the original trajectory (blue) and the trajectory obtained from the learned condensed FEM model (orange).

be further explored.

Managing Variable-Size Robot States

For now, both the MLP and LSTM network architectures can not handle robot states with variable size. Although we showed that the learned condensed FEM model can manage contacts, exploring interactions of the soft robot with its environment requires to take into account the various contact points location on the robot that are not known in advance. For this purpose, we aim at integrating size-independent neural network such as Graph Neural Networks [SM19] or Transformers [Vas+23].

Online Learning

Another challenge is to efficiently integrate the learned condensed FEM model into a reinforcement learning training loop. Due to the transition to dynamics and the multiple contact scenarios that can occur during the learning of the control policy, it is very complex to anticipate and pre-train a condensed model in advance for all potential states to be met during co-design exploration. Indeed, there are far more possible states to describe the dynamics of a robot than

its quasi-statics, as each state depends on previous states. Furthermore, there are numerous potential contact points between the robot and its environment. In reality, reinforcement learning algorithms will likely quickly converge towards more plausible robot states during their operation, making it expensive and useless to learn all possible states of the robot, especially those that will be rarely or never encountered.

One idea is to implement online learning, which involves continuously learning the condensed FEM model based on the data encountered during the policy exploration of both controller and design.

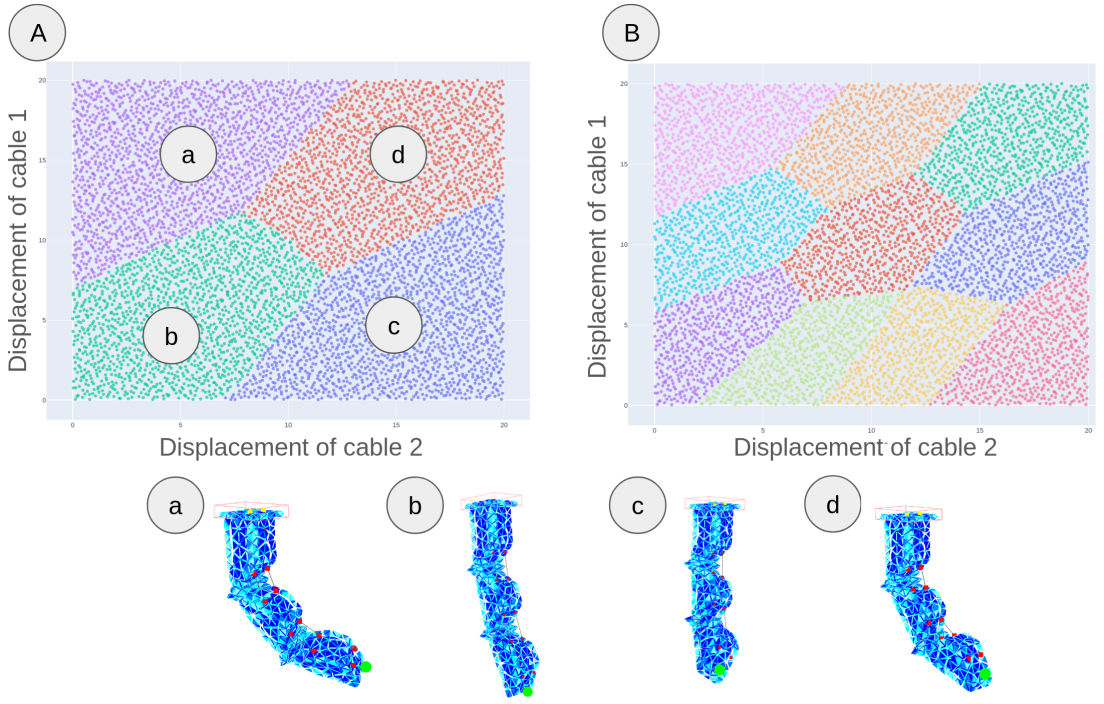

Figure 5.9: Illustrative of cluster areas obtained for different configuration of the Soft Finger respectively for A) 4 clusters and B) 10 clusters. A robot state is illustrated for each clustered area from case A).

Preliminary results have shown that it is possible to drastically reduce the number of samples required to learn the condensed models of the Diamond robot or the Soft Finger robot from Chapter 4, provided that these samples are chosen wisely. For example, applying a K-means clustering algorithm to the training sets used in previous mentioned experiments allows us to identify regions where the mechanical properties of the matrices are similar. The metrics we use for the clustering is a norm of normalized values of both W and δ^{free} quantities. These regions are actuation range where the condensed model varies little. The centroid of each cluster is then chosen as a representative of all the data within that cluster. We then build a reduced dataset from all centroids for learning the condensed FEM model. The proposed method has been applied to reduce 6500 data points previously used for training both Diamond and Soft Finger robots. It is demonstrated that surrogate models can be learned with similar performance for control using reduced datasets, i.e. with an error between each effector and target goal position being less than 0.1mm on the bending and circular trajectories from Fig. 4.3 . In other words, we obtain similar trajectory graphs for open-loop controller built from the learned condensed FEM

model with 6500 samples or from a subset of well-chosen samples. In the Diamond example, 200 samples are enough while 500 samples is a good choice in the case of the Soft Finger. Illustrative clustering results for the Soft Finger are presented in Fig. 5.9.

This demonstrates the possibility to greatly improve the sample efficiency of the learned condensed FEM model through carefully selecting the samples to include in the training set. However, in a typical reinforcement learning loop, data arrives continuously and there is no pre-existing database of the robot states. When a new robot state needs to be evaluated, it becomes necessary to decide whether to trust the learned condensed FEM model to predict this state, or to use simulation to obtain these data and incorporate them into the training set. Potential investigated solution to address this challenge includes building an uncertainty estimation using deep ensembles, i.e. an ensemble model of several deep neural network trained concurrently [LPB17].

5.4 Conclusion

In this Chapter, we introduced the current avenues being explored toward achieving co-optimization of soft robots design and control. Exploring both design and control spaces presents significant constraints regarding the balance between the size of both control and design parameters and the computational costs of the simulation used for design assessment.

We first proposed two methods to address the challenge of obtaining compact representations of the design space that can describe a wide variety of designs. The first method involves learning local rules to iteratively grow designs from a dataset of functional soft robot designs, resulting in compact and expressive shape parameterizations. The grown designs general morphology are inherently constrained by the data used to train the generative model. The second approach involves building a library of compliant mechanisms with parameterized shapes and 3D printing properties. Soft robot designs are described using these components. Condensed models relative to the design parameters of each component can be pre-learned, as described in Chapter 4. Since the complete condensed model of a soft robot can be constructed through the association of the condensed models of its components, this approach enables a modular representation suitable for both design and control.

Secondly, we introduced the use of the condensed FEM model as a surrogate to reduce the computational cost relative to the FEM simulation while exploring high-level control tasks with reinforcement learning. As many scenarios that may benefits from this kind of approach involve non negligible dynamical effects, such as locomotion and manipulation of weighted objects, we first propose an extension of the condensed FEM model to dynamics. Additionally, we introduce methods for including contacts through size-independent neural networks, and online learning strategies to increase the sample efficiency of the condensed FEM learning in a reinforcement learning loop.

Unfortunately, we were unable to finalize these works within the timeframe of this thesis. However, we hope to make significant progress on these topics in the near future.

Conclusion and Perspectives

As the field of soft robotics expands, there is an increasing need for methods to numerically explore the extensive solution space of soft robot designs. This thesis proposes to leverage FEM simulations to evaluate robot designs, ensuring that the results are accurate enough to transfer numerically optimized solutions to physical prototypes.

Although FEM simulation is commonly used to accelerate the design evaluation process, design exploration is still largely done through trial and error, requiring the designer to iteratively redesign CAD parts before re-evaluating their designs in simulation. We introduced a general methodology in Chapter 3 for emerging Physical Intelligence in soft robots. This framework relies on scripted parametric representation of geometric shapes and multi-objectives stochastic algorithms for exploring non-convex optimization landscapes. It is then generic, in the sense that the simulator is considered as a black-box, and the optimization algorithms used for design explorations do not make any assumption on the choice of the simulator. Although the optimized results cannot directly predict absolute real-world performance metrics precisely, we show that by carefully calibrating the simulation parameters used to assess the baseline design (such as mechanical parameters, mesh discretization, and boundary conditions) and making thoughtful decisions about the constraints applied within the design space, we can effectively explore and compare different design possibilities in term of relative performances. Using this framework, we demonstrated its effectiveness on two soft manipulator cases. The versatility of the method was shown through the optimization of various criteria related to the optimized robot kinematics, the sensitivity of its embedded sensors, its energy consumption, or its performance in terms of generated forces. Finally, we demonstrated that the developed optimization framework can be leveraged in calibrating the simulation parameters, a task that is challenging for traditional gradient-based algorithms in highly nonlinear scenarios involving contacts. We showed that these methods produce designs that outperform user-intuitive baselines, especially in highly unintuitive cases, such as those involving internal contacts. As results takes the shape of a Pareto Front, they enable to efficiently investigate compromise between several evaluation metrics. Furthermore, they offer an agile way to share designs. Indeed, the user does not need to replicate the design to adapt it to their own application, as they can directly regenerate an optimized design for their specific goals and constraints. We have thus shown that it is possible to easily re-optimize the shape of the manipulator using self-contacts to adapt to manipulation tasks for diverse object shapes.

The considered optimized robots in this thesis work are simple proofs of concepts, and the scaling to complex robotics systems with many design parameters still needs to be assessed. Although the proposed stochastic framework helps the design exploration, a significant constraint still arises from the definition of the optimized parametric design. This constraint limits generated designs to a baseline morphology chosen by the user. Preliminary work on exploring more expressive design spaces have been explored, and presented in Chapter 5.2. First, we

introduced a generative design framework using Neural Cellular Automata to create novel designs from a dataset of functional soft robot designs examples. By building a parameterization from examples, we can reduce the complexity in constructing an efficient geometric parameterization of the soft robot body. However, it is essential to incorporate both physical and manufacturing constraints into the current framework to ensure the plausibility of the generated designs. Secondly, we developed a library of modular parametric components for designing soft robots. These components are simple geometric shapes inspired by compliant mechanism research. By integrating 3D printing features such as infill density and anisotropic patterns into their parameterization, we can smoothly adjust their mechanical properties. The purpose of this library is to define soft robot designs based on the desired degrees of freedom, akin to rigid robotics. Utilizing pre-designed modules provides several benefits. By employing a finite set of known components, we can pre-characterize and develop reduced models for each of them, significantly improving computation time during a complete design evaluation. This approach also ensures the manufacturability of the final designs.

Besides problematics related to design parameterization, the computation time related to design assessment in simulation does not enable to consider high level controllers, but only simple pre-determined direct or inverse strategies. For instance, in the optimization of a manipulator, it is assumed that one must predefine either the contact zones between the object and the gripper, or its actuation strategy. However, it is clear that different control strategies are more suitable depending on the gripper design. Thus, emerging true Embodied Intelligence in soft robots requires to co-optimize for both controller and design.

For this purpose, we developed the learned condensed FEM model in Chapter 4, shared as an open-source plugin for the simulation software SOFA, as a surrogate accounting for both control and design particularities. Low dimensional neural network are used to rapidly predict mechanical matrices quantities that synthesize the link between mathematical constraints (actuation, effector, contacts) exerted on the robot under the quasi-static assumption. We leverage these mechanical matrices in applications such as embedded inverse control of soft robot or design optimization. We have demonstrated the usefulness of employing this method to replace simulation in embedded inverse control schemes. This significantly reduces the time and memory required for numerical simulation computations, a critical barrier for achieving real-time simulation of complex soft robots. Furthermore, we showcased the use of this framework for design optimization, leveraging neural network differentiability to effectively explore the design space. Further investigation of this capability may consist in the implementation of additional constraints to guarantee the smoothness of the neural network gradients. For instance, techniques such as Sobolev learning could be considered [Cza+17].

As discussed in Chapter 5.3, the condensed FEM model needs to be extended for use as a surrogate in optimizing high-level control policies with methods such as reinforcement learning. It is essential to account for the robot dynamics, particularly to exploit the non-linear dynamics of soft bodies in complex tasks like locomotion and rapid manipulation. Promising preliminary results have been obtained in learning the condensed FEM for dynamic trajectory predictions, but the framework needs to be scaled to handle large datasets. One current limitation is the neural network architecture inability to handle robot states with variable sizes. When exploring high-level controllers, new unplanned contact points may alter the size of the learned mechanical matrices. Additionally, as the design space expands, uniformly sampling the entire mathematical constraint space beforehand to learn the condensed FEM model is computationally impractical, as many states may never be encountered. Future work will focus on using size-independent neural networks and online learning strategies to address these challenges.

The soft robotics field still tends to prioritize the search for innovative solutions over standard and robust ones [Bai+24]. As a result, there are still few commercial soft robots, except for medical applications and soft grippers. This lack of standards and benchmarks is visible at multiple levels, including the available simulation tools, manufacturing methods, and means of comparing results.

There is a shortage of comprehensive, high-performance digital tools for the control and design of soft robots. Many simulators have emerged from university laboratories in recent years, but they remain small projects implementing only a few specific models [Mac+14; Bai; CPN24; Bha+22; Hu+18b]. This limits their use to simple, highly specialized prototypes. As discussed in the Background of this thesis, the numerical optimization of soft robot designs is still largely dependent on simulators that prioritize computation speed over accuracy [Bha+22; Hu+18b], making them less relevant for simulation-to-reality transfer. While the SOFA [Fau+12] simulator used in this thesis offers accurate models under correct assumptions, it is still limited in terms of available models. For example, the platform does not directly provide magnetic actuation models, despite their use in many soft robots [Kim+19; Zho+21a; Ze+22; Kha+12]. Additionally, SOFA remains challenging to use for users who are not comfortable with programming. In other industrial fields (automotive, aviation, traditional rigid industrial robotics), commercial solutions such as ANSYS, Abaqus or Comsol, are more comprehensive, standardized, and accessible from a user perspective. Although the tools used in these fields can be employed for the analysis of static designs, they do not implement the necessary components for robotic control. The distribution of models across various scattered platforms makes it particularly challenging to use simulation for the generic exploration of soft robots design and controller. The soft robotics fields would greatly benefit from such a platform in the future.

This lack of standardization also impacts the ease of sharing soft robot designs. Given the absence of commercially available building blocks, laboratories continue to develop their own custom designs and recipes for manufacturing actuators, sensors, and controllers. This makes designs difficult to replicate. Some efforts are being made to facilitate the sharing of these designs, for instance through sharing tutorials and open-source repositories that allow the replication of designs [Del+17; Hab+24], and platforms like the Soft Robotics Toolkit[2] that compile design fabrication tutorials. However, this effort is not consistent in the literature. In this thesis, we have aimed to contribute by providing open-source code for each of our contributions. Additionally, the development and democratization of more robust and accessible design methods, such as 3D printing of silicons [Spa+21] and composite materials [Buc+23], could further facilitate the sharing of complex designs in the near-future. Finally, there is a lack of clear metrics for comparing robotic designs. This issue is directly related to manufacturing challenges and the absence of standard evaluation platforms. For example, it is reported in [Gra+20] that approximately one out of every two robots built for a scientific publication on modeling and control of concentric tube robots is used only once. Working on common platforms would facilitate easier comparisons and accelerate progress on the main research challenges of the soft robotics fields. Such a platform has been proposed for the evaluation of pneumatic robots [Shi+23]. There is also a need for standardized indicators and clear shared quantitative metrics to compare the performance of soft robots. For instance, the authors of [Bai+24] advocate that terms like "robustness" are often mentioned in scientific contributions without quantitative evaluations. The soft robotics community has begun working on establishing standard for testing and metrics and methods. This was the subject of a workshop I attended in the 2024 RoboSoft conference. Ideas such as enforcing a systemic and comprehensive standardized report for replicating and comparing results start to emerge. This approach may facilitate the sharing and comparison of designs, leading to the development of standard components. Collaboration and

[2] https://softroboticstoolkit.com/home

transfer to industry will also be crucial for establishing and refining these standards.

We believe that the methods developed in this thesis will be particularly relevant in this context, with the spread of more generic numerical tools and when families of standard components will be clearly identified. Shareable parametric and learned condensed FEM models could be used for automatic design and control applications. This is the idea we would like to explore with the development of the modular library of compliant mechanisms from Chapter 5.2.2.

Bibliography

[23] *Soft Pneumatic Actuator Fabrication with Silicone 3D Printing and Robotics*. https://lynxter.fr/en/. Aug. 8, 2023. URL: https://lynxter.fr/en/blog/chronicles/soft-pneumatic-actuator-fabrication-with-silicone-3d-printing-and-robotics/ (visited on 04/04/2024).

[Aba+19] Sara-Adela Abad et al. "Significance of the Compliance of the Joints on the Dynamic Slip Resistance of a Bioinspired Hoof". In: *IEEE Transactions on Robotics* 35.6 (Dec. 2019), pp. 1450–1463. ISSN: 1941-0468. DOI: 10.1109/TRO.2019.2930864. URL: https://ieeexplore.ieee.org/document/8807297 (visited on 06/06/2024).

[Abi+18] Haider Abidi et al. "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery". In: *The international journal of medical robotics + computer assisted surgery: MRCAS* 14.1 (Feb. 2018). ISSN: 1478-596X. DOI: 10.1002/rcs.1875.

[Aki+19] Takuya Akiba et al. *Optuna: A Next-generation Hyperparameter Optimization Framework*. arXiv.org. July 25, 2019. URL: https://arxiv.org/abs/1907.10902v1 (visited on 01/12/2024).

[Alo+22] John Irvin Alora et al. *Data-Driven Spectral Submanifold Reduction for Nonlinear Optimal Control of High-Dimensional Robots*. Sept. 20, 2022. arXiv: 2209.05712[cs]. URL: http://arxiv.org/abs/2209.05712 (visited on 05/17/2024).

[Als+23] Mohammad Alshawabkeh et al. "Highly Stretchable Additively Manufactured Capacitive Proximity and Tactile Sensors for Soft Robotic Systems". In: *IEEE Transactions on Instrumentation and Measurement* 72 (2023), pp. 1–10. ISSN: 1557-9662. DOI: 10.1109/TIM.2023.3250232. URL: https://ieeexplore.ieee.org/document/10057215 (visited on 07/06/2024).

[ARD21] Yinoussa Adagolodjo, Federico Renda, and Christian Duriez. "Coupling Numerical Deformable Models in Global and Reduced Coordinates for the Simulation of the Direct and the Inverse Kinematics of Soft Robots". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 3910–3917. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3061977. URL: https://ieeexplore.ieee.org/abstract/document/9362217 (visited on 01/05/2024).

[Arm+22] Costanza Armanini et al. *Soft Robots Modeling: a Structured Overview*. Oct. 4, 2022. DOI: 10.48550/arXiv.2112.03645. arXiv: 2112.03645[cs]. URL: http://arxiv.org/abs/2112.03645 (visited on 01/24/2024).

[AZK21] Walid Amehri, Gang Zheng, and Alexandre Kruszewski. "Workspace Boundary Estimation for Soft Manipulators Using a Continuation Approach". In: *IEEE Robotics and Automation Letters* 6.4 (Oct. 2021), pp. 7169–7176. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3097662. URL: https://ieeexplore.ieee.org/document/9488225 (visited on 12/29/2023).

[Baa+22] Thomas Baaij et al. "Learning 3D shape proprioception for continuum soft robots with multiple magnetic sensors". In: *Soft Matter* 19.1 (Dec. 21, 2022), pp. 44–56. ISSN: 1744-6848. DOI: 10.1039/D2SM00914E. URL: https://pubs.rsc.org/en/content/articlelanding/2023/sm/d2sm00914e (visited on 07/06/2024).

[Bai] E. Coumans {and} Y. Bai. *Pybullet, a python module for physics simulation for games, robotics and machine learning*. URL: http://pybullet.xn--org,%2020162021-qn6g.

[Bai+24] Robert Baines et al. "The need for reproducible research in soft robotics". In: *Nature Machine Intelligence* 6.7 (July 2024), pp. 740–741. ISSN: 2522-5839. DOI: 10.1038/s42256-024-00869-9. URL: https://www.nature.com/articles/s42256-024-00869-9 (visited on 07/25/2024).

[Bam+22] Antoine Bambade et al. "PROX-QP: Yet another Quadratic Programming Solver for Robotics and beyond". In: RSS 2022 - Robotics: Science and Systems. June 27, 2022. URL: https://inria.hal.science/hal-03683733 (visited on 05/17/2024).

[Bar+15a] N. Bartlett et al. *A Soft Robotic Orthosis for Wrist Rehabilitation*. 2015.

[Bar+15b] Nicholas W. Bartlett et al. "A 3D-printed, functionally graded soft robot powered by combustion". In: *Science* 349.6244 (July 10, 2015), pp. 161–165. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aab0129. URL: https://www.science.org/doi/10.1126/science.aab0129 (visited on 04/10/2024).

[Ber+11] James Bergstra et al. "Algorithms for Hyper-Parameter Optimization". In: *Advances in Neural Information Processing Systems*. Vol. 24. Curran Associates, Inc., 2011. URL: https://papers.nips.cc/paper_files/paper/2011/hash/86e8f7ab32cfd12577bc2619bc635690-Abstract.html (visited on 01/12/2024).

[Ber+17] Katia Bertoldi et al. "Flexible mechanical metamaterials". In: *Nature Reviews Materials* 2.11 (Oct. 17, 2017), pp. 1–11. ISSN: 2058-8437. DOI: 10.1038/natrevmats.2017.66. URL: https://www.nature.com/articles/natrevmats201766 (visited on 04/05/2024).

[Ber+20] James M. Bern et al. "Soft Robot Control With a Learned Differentiable Model". In: *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). New Haven, CT, USA: IEEE, May 2020, pp. 417–423. ISBN: 9781728165707. DOI: 10.1109/RoboSoft48309.2020.9116011. URL: https://ieeexplore.ieee.org/document/9116011/ (visited on 05/17/2024).

[Bes+16] Charles M. Best et al. "A New Soft Robot Control Method: Using Model Predictive Control for a Pneumatically Actuated Humanoid". In: *IEEE Robotics & Automation Magazine* 23.3 (Sept. 2016), pp. 75–84. ISSN: 1558-223X. DOI: 10.1109/MRA.2016.2580591. URL: https://ieeexplore.ieee.org/document/7551190 (visited on 07/09/2024).

[Bha+22] Jagdeep Singh Bhatia et al. *Evolution Gym: A Large-Scale Benchmark for Evolving Soft Robots*. Jan. 24, 2022. DOI: 10.48550/arXiv.2201.09863. arXiv: 2201.09863[cs]. URL: http://arxiv.org/abs/2201.09863 (visited on 03/11/2024).

[Bla+24] David Blanco-Mulero et al. *Benchmarking the Sim-to-Real Gap in Cloth Manipulation*. Jan. 25, 2024. DOI: 10.48550/arXiv.2310.09543. arXiv: 2310.09543[cs]. URL: http://arxiv.org/abs/2310.09543 (visited on 02/15/2024).

[Bon+22] Sam Bond-Taylor et al. "Deep Generative Modelling: A Comparative Review of VAEs, GANs, Normalizing Flows, Energy-Based and Autoregressive Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.11 (Nov. 2022), pp. 7327–7347. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2021.3116668. URL: https://ieeexplore.ieee.org/document/9555209 (visited on 06/13/2024).

[Bro+10] Eric Brown et al. "Universal robotic gripper based on the jamming of granular material". In: *Proceedings of the National Academy of Sciences* 107.44 (Nov. 2, 2010), pp. 18809–18814. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1003250107. URL: https://pnas.org/doi/full/10.1073/pnas.1003250107 (visited on 06/06/2024).

[Bro91] Rodney A. Brooks. "New Approaches to Robotics". In: *Science* 253.5025 (Sept. 13, 1991), pp. 1227–1232. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.253.5025.1227. URL: https://www.science.org/doi/10.1126/science.253.5025.1227 (visited on 06/10/2024).

[BRR18] Pinar Boyraz, Gundula Runge, and Annika Raatz. "An Overview of Novel Actuators for Soft Robotics". In: *Actuators* 7.3 (Sept. 2018), p. 48. ISSN: 2076-0825. DOI: 10.3390/act7030048. URL: https://www.mdpi.com/2076-0825/7/3/48 (visited on 04/10/2024).

[BS16] Andrea Bajo and Nabil Simaan. "Hybrid motion/force control of multi-backbone continuum robots". In: *The International Journal of Robotics Research* 35.4 (Apr. 2016), pp. 422–434. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364915584806. URL: http://journals.sagepub.com/doi/10.1177/0278364915584806 (visited on 07/09/2024).

[Buc+23] Thomas J. K. Buchner et al. "Vision-controlled jetting for composite systems and robots". In: *Nature* 623.7987 (Nov. 2023), pp. 522–530. ISSN: 1476-4687. DOI: 10.1038/s41586-023-06684-3. URL: https://www.nature.com/articles/s41586-023-06684-3 (visited on 04/04/2024).

[BVL16] Guus ten Broeke, George van Voorn, and Arend Ligtenberg. "Which Sensitivity Analysis Method Should I Use for My Agent-Based Model?" In: *Journal of Artificial Societies and Social Simulation* 19.1 (2016), p. 5. ISSN: 1460-7425.

[CBS23] François Cochevelou, David Bonner, and Martin-Pierre Schmidt. "Differentiable Soft-Robot Generation". In: *Proceedings of the Genetic and Evolutionary Computation Conference*. GECCO '23: Genetic and Evolutionary Computation Conference. Lisbon Portugal: ACM, July 15, 2023, pp. 129–137. ISBN: 9798400701191. DOI: 10.1145/3583131.3590408. URL: https://dl.acm.org/doi/10.1145/3583131.3590408 (visited on 07/21/2023).

[CED17] Eulalie Coevoet, Adrien Escande, and Christian Duriez. "Optimization-Based Inverse Model of Soft Robots With Contact Handling". In: *IEEE Robotics and Automation Letters* 2.3 (July 2017), pp. 1413–1419. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2669367. URL: https://ieeexplore.ieee.org/abstract/document/7856996 (visited on 01/05/2024).

[CFW18] Anant Chawla, Chase Frazelle, and Ian Walker. "A Comparison of Constant Curvature Forward Kinematics for Multisection Continuum Manipulators". In: *2018 Second IEEE International Conference on Robotic Computing (IRC)*. 2018 Second IEEE International Conference on Robotic Computing (IRC). Jan. 2018, pp. 217–223. DOI: 10.1109/IRC.2018.00046. URL: https://ieeexplore.ieee.org/document/8329910 (visited on 02/08/2024).

[Cha+23] Paul Chaillou et al. "Reduced finite element modelling and closed-loop control of pneumatic-driven soft continuum robots". In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)*. 2023 IEEE International Conference on Soft Robotics (RoboSoft). ISSN: 2769-4534. Apr. 2023, pp. 1–8. DOI: 10.1109/RoboSoft55895.2023.10122081. URL: https://ieeexplore.ieee.org/document/10122081 (visited on 05/17/2024).

[Che+14] Nick Cheney et al. "Unshackling evolution: evolving soft robots with multiple materials and a powerful generative encoding". In: *ACM SIGEVOlution* 7.1 (Aug. 1, 2014), pp. 11–23. DOI: 10.1145/2661735.2661737. URL: https://doi.org/10.1145/2661735.2661737 (visited on 03/06/2024).

[Che+17] Feifei Chen et al. "Topology optimization of hyperelastic structures using a level set method". In: *Journal of Computational Physics* 351 (Dec. 2017), pp. 437–454. ISSN: 00219991. DOI: 10.1016/j.jcp.2017.09.040. URL: https://linkinghub.elsevier.com/retrieve/pii/S0021999117307003 (visited on 09/20/2021).

[Che+18a] Feifei Chen et al. "Topology Optimized Design, Fabrication, and Characterization of a Soft Cable-Driven Gripper". In: *IEEE Robotics and Automation Letters* 3.3 (July 2018), pp. 2463–2470. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2018.2800115. URL: https://ieeexplore.ieee.org/document/8276274/ (visited on 03/13/2024).

[Che+18b] Nick Cheney et al. "Scalable co-optimization of morphology and control in embodied machines". In: *Journal of The Royal Society Interface* 15.143 (June 2018), p. 20170937. ISSN: 1742-5689, 1742-5662. DOI: 10.1098/rsif.2017.0937. URL: https://royalsocietypublishing.org/doi/10.1098/rsif.2017.0937 (visited on 03/06/2024).

[Che+21] Shitong Chen et al. "Topology Optimization of Skeleton-Reinforced Soft Pneumatic Actuators for Desired Motions". In: *IEEE/ASME Transactions on Mechatronics* 26.4 (Aug. 2021), pp. 1745–1753. ISSN: 1083-4435, 1941-014X. DOI: 10.1109/TMECH.2021.3071394. URL: https://ieeexplore.ieee.org/document/9397352/ (visited on 04/01/2022).

[Cia+18] Matteo Cianchetti et al. "Biomedical applications of soft robotics". In: *Nature Reviews Materials* 3.6 (June 2018), pp. 143–153. ISSN: 2058-8437. DOI: 10.1038/s41578-018-0022-y. URL: https://www.nature.com/articles/s41578-018-0022-y (visited on 06/10/2024).

[Coe+17] E. Coevoet et al. "Software toolkit for modeling, simulation, and control of soft robots". In: *Advanced Robotics* 31.22 (Nov. 17, 2017), pp. 1208–1224. ISSN: 0169-1864, 1568-5535. DOI: 10.1080/01691864.2017.1395362. URL: https://www.tandfonline.com/doi/full/10.1080/01691864.2017.1395362 (visited on 08/03/2022).

[Coe19] Eulalie Coevoet. "Optimization-based inverse model of soft robots, with contact handling". PhD thesis. Université de Lille 1, Sciences et Technologies, Jan. 9, 2019. URL: https://hal.science/tel-02446416 (visited on 02/08/2024).

[Col+21] Jack Collins et al. “A Review of Physics Simulators for Robotic Applications”. In: *IEEE Access* 9 (2021), pp. 51416–51431. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2021.3068769. URL: https://ieeexplore.ieee.org/document/9386154/ (visited on 08/04/2022).

[Coy+18] Stephen Coyle et al. “Bio-inspired soft robotics: Material selection, actuation, and design”. In: *Extreme Mechanics Letters* 22 (July 1, 2018), pp. 51–59. ISSN: 2352-4316. DOI: 10.1016/j.eml.2018.05.003. URL: https://www.sciencedirect.com/science/article/pii/S2352431617302316 (visited on 03/06/2024).

[CPN20] Brandon Caasenbrood, Alexander Pogromsky, and Henk Nijmeijer. “A Computational Design Framework for Pressure-driven Soft Robots through Nonlinear Topology Optimization”. In: *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*. 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft). New Haven, CT, USA: IEEE, May 2020, pp. 633–638. ISBN: 9781728165707. DOI: 10.1109/RoboSoft48309.2020.9116010. URL: https://ieeexplore.ieee.org/document/9116010/ (visited on 06/03/2022).

[CPN24] Brandon J. Caasenbrood, Alexander Y. Pogromsky, and Henk Nijmeijer. “Sorotoki: A Matlab Toolkit for Design, Modeling, and Control of Soft Robots”. In: *IEEE Access* 12 (2024), pp. 17604–17638. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3357351. URL: https://ieeexplore.ieee.org/document/10411872 (visited on 03/06/2024).

[CWB17] Fionnuala Connolly, Conor J. Walsh, and Katia Bertoldi. “Automatic design of fiber-reinforced soft actuators for trajectory matching”. In: *Proceedings of the National Academy of Sciences* 114.1 (Jan. 3, 2017), pp. 51–56. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1615140114. URL: https://pnas.org/doi/full/10.1073/pnas.1615140114 (visited on 03/15/2024).

[CWY24] Wee Kiat Chan, PengWei Wang, and Raye Chen-Hua Yeow. *Creation of Novel Soft Robot Designs using Generative AI*. May 2, 2024. DOI: 10.48550/arXiv.2405.01824. arXiv: 2405.01824[cs]. URL: http://arxiv.org/abs/2405.01824 (visited on 06/12/2024).

[Cza+17] Wojciech Marian Czarnecki et al. *Sobolev Training for Neural Networks*. July 26, 2017. DOI: 10.48550/arXiv.1706.04859. arXiv: 1706.04859[cs]. URL: http://arxiv.org/abs/1706.04859 (visited on 07/02/2024).

[Däm+19] Gabriel Dämmer et al. “PolyJet-Printed Bellows Actuators: Design, Structural Optimization, and Experimental Investigation”. In: *Frontiers in Robotics and AI* 6 (May 14, 2019). ISSN: 2296-9144. DOI: 10.3389/frobt.2019.00034. URL: https://www.frontiersin.org/articles/10.3389/frobt.2019.00034 (visited on 03/15/2024).

[Dav+23] Joshua Davy et al. “A Framework for Simulation of Magnetic Soft Robots Using the Material Point Method”. In: *IEEE Robotics and Automation Letters* 8.6 (June 2023), pp. 3470–3477. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2023.3268016. URL: https://ieeexplore.ieee.org/document/10103621/ (visited on 02/08/2024).

[Deb+02] K. Deb et al. “A fast and elitist multiobjective genetic algorithm: NSGA-II”. In: *IEEE Transactions on Evolutionary Computation* 6.2 (Apr. 2002), pp. 182–197. ISSN: 1941-0026. DOI: 10.1109/4235.996017. URL: https://ieeexplore.ieee.org/document/996017 (visited on 01/12/2024).

[Dei+17] Raphael Deimel et al. “Automated co-design of soft hand morphology and control strategy for grasping”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN: 2153-0866. Sept. 2017, pp. 1213–1218. DOI: 10.1109/IROS.2017.8202294. URL: https://ieeexplore.ieee.org/document/8202294 (visited on 03/20/2024).

[Del+17] Cosimo Della Santina et al. “The Quest for Natural Machine Motion: An Open Platform to Fast-Prototyping Articulated Soft Robots”. In: *IEEE Robotics & Automation Magazine* 24.1 (Mar. 2017), pp. 48–56. ISSN: 1558-223X. DOI: 10.1109/MRA.2016.2636366. URL: https://ieeexplore.ieee.org/abstract/document/7857692 (visited on 07/25/2024).

[Del+20] Cosimo Della Santina et al. “Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment”. In: *The International Journal of Robotics Research* 39.4 (Mar. 2020), pp. 490–513. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364919897292. URL: http://journals.sagepub.com/doi/10.1177/0278364919897292 (visited on 04/10/2024).

[Dil92] Ellis Harold Dill. “Kirchhoff’s Theory of Rods”. In: *Archive for History of Exact Sciences* 44.1 (1992), pp. 1–23. ISSN: 0003-9519. URL: https://www.jstor.org/stable/41133926 (visited on 02/14/2024).

[Din+19] Longwei Ding et al. “Design of soft multi-material pneumatic actuators based on principal strain field”. In: *Materials & Design* 182 (Nov. 15, 2019), p. 108000. ISSN: 0264-1275. DOI: 10.1016/j.matdes.2019.108000. URL: https://www.sciencedirect.com/science/article/pii/S0264127519304381 (visited on 03/15/2024).

[DL21] Zhifeng Deng and Miao Li. “Learning Optimal Fin-Ray Finger Design for Soft Grasping”. In: *Frontiers in Robotics and AI* 7 (2021). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2020.590076 (visited on 12/29/2023).

[Don+18] Huixu Dong et al. “Geometric design optimization of an under-actuated tendon-driven robotic gripper”. In: *Robotics and Computer-Integrated Manufacturing* 50 (Apr. 1, 2018), pp. 80–89. ISSN: 0736-5845. DOI: 10.1016/j.rcim.2017.09.012. URL: https://www.sciencedirect.com/science/article/pii/S0736584517301643 (visited on 12/29/2023).

[Dro+21] Dylan Drotman et al. “Electronics-free pneumatic circuits for controlling soft-legged robots”. In: *Science Robotics* 6.51 (Feb. 24, 2021), eaay2627. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aay2627. URL: https://www.science.org/doi/10.1126/scirobotics.aay2627 (visited on 06/06/2024).

[Dur13a] Christian Duriez. “Control of Elastic Soft Robots based on Real-Time Finite Element Method”. In: ICRA 2013 IEEE International Conference on Robotics and Automation. 2013. URL: https://hal.science/hal-00823766 (visited on 02/14/2024).

[Dur13b] Christian Duriez. “Real-time haptic simulation of medical procedures involving deformations and device-tissue interactions”. thesis. Université des Sciences et Technologie de Lille - Lille I, Feb. 1, 2013. URL: https://theses.hal.science/tel-00785118 (visited on 02/19/2024).

[El-+20] Nazek El-Atab et al. “Soft Actuators for Soft Robotic Applications: A Review”. In: *Advanced Intelligent Systems* 2.10 (Oct. 2020), p. 2000128. ISSN: 2640-4567, 2640-4567. DOI: 10.1002/aisy.202000128. URL: https://onlinelibrary.wiley.com/doi/10.1002/aisy.202000128 (visited on 04/10/2024).

[FAS18] Jakob A. Faber, Andres F. Arrieta, and André R. Studart. “Bioinspired spring origami”. In: *Science* 359.6382 (Mar. 23, 2018), pp. 1386–1391. ISSN: 0036-8075, 1095-9203. DOI: 10.1126/science.aap7753. URL: https://www.science.org/doi/10.1126/science.aap7753 (visited on 04/05/2024).

[Fau+12] François Faure et al. “SOFA: A Multi-Model Framework for Interactive Physical Simulation”. In: *Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Ed. by Yohan Payan. Studies in Mechanobiology, Tissue Engineering and Biomaterials. Berlin, Heidelberg: Springer, 2012, pp. 283–321. ISBN: 9783642290145. DOI: 10.1007/8415_2012_125. URL: https://doi.org/10.1007/8415_2012_125 (visited on 02/08/2024).

[Fra18] Peter I. Frazier. *A Tutorial on Bayesian Optimization*. July 8, 2018. DOI: 10.48550/arXiv.1807.02811. arXiv: 1807.02811[cs,math,stat]. URL: http://arxiv.org/abs/1807.02811 (visited on 07/09/2024).

[FSP23] Andreas Henrik Frederiksen, Ole Sigmund, and Konstantinos Poulios. “Topology optimization of self-contacting structures”. In: *Computational Mechanics* (Oct. 7, 2023). ISSN: 1432-0924. DOI: 10.1007/s00466-023-02396-7. URL: https://doi.org/10.1007/s00466-023-02396-7 (visited on 03/13/2024).

[Gao+20] Anzhu Gao et al. “Modeling and Task-Oriented Optimization of Contact-Aided Continuum Robots”. In: *IEEE/ASME Transactions on Mechatronics* 25.3 (June 2020), pp. 1444–1455. ISSN: 1941-014X. DOI: 10.1109/TMECH.2020.2977107. URL: https://ieeexplore.ieee.org/document/9018009 (visited on 12/29/2023).

[GCD21] Olivier Goury, Bruno Carrez, and Christian Duriez. “Real-Time Simulation for Control of Soft Robots With Self-Collisions Using Model Order Reduction for Contact Forces”. In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 3752–3759. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3064247. URL: https://ieeexplore.ieee.org/document/9372815 (visited on 03/01/2024).

[GD18] Olivier Goury and Christian Duriez. “Fast, Generic, and Reliable Control and Simulation of Soft Robots Using Model Order Reduction”. In: *IEEE Transactions on Robotics* 34.6 (Dec. 2018), pp. 1565–1576. ISSN: 1941-0468. DOI: 10.1109/TRO.2018.2861900. URL: https://ieeexplore.ieee.org/document/8453914 (visited on 01/05/2024).

[Gen58] A. N. Gent. “On the Relation between Indentation Hardness and Young’s Modulus”. In: *Rubber Chemistry and Technology* 31.4 (Sept. 1, 1958), pp. 896–906. ISSN: 1943-4804, 0035-9475. DOI: 10.5254/1.3542351. URL: https://meridian.allenpress.com/rct/article/31/4/896/89954/On-the-Relation-between-Indentation-Hardness-and (visited on 12/29/2023).

[Gha+21] Keyan Ghazi-Zahedi et al. “Editorial: Recent Trends in Morphological Computation”. In: *Frontiers in Robotics and AI* 8 (May 31, 2021). ISSN: 2296-9144. DOI: 10.3389/frobt.2021.708206. URL: https://www.frontiersin.org/articles/10.3389/frobt.2021.708206 (visited on 06/06/2024).

[Gil+18] Morgan T. Gillespie et al. “Learning nonlinear dynamic models of soft robots for model predictive control with neural networks”. In: *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. 2018 IEEE International Conference on Soft Robotics (RoboSoft). Apr. 2018, pp. 39–45. DOI: 10.1109/ROBOSOFT.2018.8404894. URL: https://ieeexplore.ieee.org/document/8404894 (visited on 07/05/2024).

[GK20] Di Guo and Zhan Kang. “Chamber layout design optimization of soft pneumatic robots”. In: *Smart Material Structures* 29 (Feb. 1, 2020). ADS Bibcode: 2020SMaS...29b5017G, p. 025017. ISSN: 0964-1726. DOI: 10.1088/1361-665X/ab607b. URL: https://ui.adsabs.harvard.edu/abs/2020SMaS...29b5017G (visited on 07/06/2024).

[GPM89] Carlos E. García, David M. Prett, and Manfred Morari. “Model predictive control: Theory and practice—A survey”. In: *Automatica* 25.3 (May 1, 1989), pp. 335–348. ISSN: 0005-1098. DOI: 10.1016/0005-1098(89)90002-2. URL: https://www.sciencedirect.com/science/article/pii/0005109889900022 (visited on 05/17/2024).

[GR09] 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* 79.11 (Sept. 10, 2009), pp. 1309–1331. ISSN: 0029-5981, 1097-0207. DOI: 10.1002/nme.2579. URL: https://onlinelibrary.wiley.com/doi/10.1002/nme.2579 (visited on 01/05/2024).

[Gra+20] Reinhard M. Grassmann et al. “CTCR Prototype Development: An Obstacle in the Research Community?” In: Robotics Retrospectives - Workshop at RSS 2020. June 25, 2020. URL: https://openreview.net/forum?id=bYLFxFQPFtX (visited on 07/25/2024).

[Gro+21] B. Grossi et al. “Metarpillar: Soft robotic locomotion based on buckling-driven elastomeric metamaterials”. In: *Materials & Design* 212 (Dec. 15, 2021), p. 110285. ISSN: 0264-1275. DOI: 10.1016/j.matdes.2021.110285. URL: https://www.sciencedirect.com/science/article/pii/S0264127521008406 (visited on 04/05/2024).

[Gup+21] Agrim Gupta et al. “Embodied intelligence via learning and evolution”. In: *Nature Communications* 12.1 (Oct. 6, 2021), p. 5721. ISSN: 2041-1723. DOI: 10.1038/s41467-021-25874-z. URL: https://www.nature.com/articles/s41467-021-25874-z (visited on 06/10/2024).

[Ha+17] Sehoon Ha et al. “Joint optimization of robot design and motion parameters using the implicit function theorem: 2017 Robotics: Science and Systems, RSS 2017”. In: *Robotics*. Robotics: Science and Systems (2017). Ed. by Siddhartha Srinivasa et al. DOI: 10.15607/rss.2017.xiii.003. URL: http://www.scopus.com/inward/record.url?scp=85048806626&partnerID=8YFLogxK (visited on 07/11/2024).

[Haa+18] Tuomas Haarnoja et al. *Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor*. Aug. 8, 2018. DOI: 10.48550/arXiv.1801.01290. arXiv: 1801.01290[cs,stat]. URL: http://arxiv.org/abs/1801.01290 (visited on 06/20/2024).

[Hab+20] Hossein Habibi et al. “A Lumped-Mass Model for Large Deformation Continuum Surfaces Actuated by Continuum Robotic Arms”. In: *Journal of Mechanisms and Robotics* 12.1 (Feb. 1, 2020), pp. 1–12. ISSN: 1942-4302. DOI: 10.1115/1.4045037. URL: http://www.mendeley.com/research/lumpedmass-model-large-deformation-continuum-surfaces-actuated-continuum-robotic-arms (visited on 02/14/2024).

[Hab+24] Tim-Lukas Habich et al. "SPONGE: Open-Source Designs of Modular Articulated Soft Robots". In: *IEEE Robotics and Automation Letters* 9.6 (June 2024), pp. 5346–5353. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2024.3388855. arXiv: 2404.10734[cs]. URL: http://arxiv.org/abs/2404.10734 (visited on 07/25/2024).

[Han05] Nikolaus Hansen. "The CMA Evolution Strategy: A Tutorial". 2005. URL: https://inria.hal.science/hal-01297037 (visited on 01/12/2024).

[Haw+17] Elliot W. Hawkes et al. "A soft robot that navigates its environment through growth". In: *Science Robotics* 2.8 (July 19, 2017), eaan3028. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aan3028. URL: https://www.science.org/doi/10.1126/scirobotics.aan3028 (visited on 01/05/2024).

[HL14] Jonathan Hiller and Hod Lipson. "Dynamic Simulation of Soft Multimaterial 3D-Printed Objects". In: *Soft Robotics* 1.1 (Mar. 2014), pp. 88–101. ISSN: 2169-5172, 2169-5180. DOI: 10.1089/soro.2013.0010. URL: https://www.liebertpub.com/doi/10.1089/soro.2013.0010 (visited on 03/11/2024).

[HMO13] Larry L. Howell, Spencer P. Magleby, and Brian M. Olsen, eds. *Handbook of Compliant Mechanisms*. 1st ed. Wiley, Feb. 4, 2013. ISBN: 9781119953456 9781118516485. DOI: 10.1002/9781118516485. URL: https://onlinelibrary.wiley.com/doi/book/10.1002/9781118516485 (visited on 07/11/2024).

[HNF23] Helmut Hauser, Thrishantha Nanayakkara, and Fulvio Forni. "Leveraging Morphological Computation for Controlling Soft Robots: Learning from Nature to Control Soft Robots". In: *IEEE Control Systems* 43.3 (June 2023), pp. 114–129. ISSN: 1066-033X, 1941-000X. DOI: 10.1109/MCS.2023.3253422. URL: https://ieeexplore.ieee.org/document/10136438/ (visited on 06/06/2024).

[Hom+19] Bianca S. Homberg et al. "Robust proprioceptive grasping with a soft robot hand". In: *Autonomous Robots* 43.3 (Mar. 1, 2019), pp. 681–696. ISSN: 1573-7527. DOI: 10.1007/s10514-018-9754-1. URL: https://doi.org/10.1007/s10514-018-9754-1 (visited on 04/10/2024).

[Hu+18a] Weiping Hu et al. "A Structural Optimisation Method for a Soft Pneumatic Actuator". In: *Robotics* 7.2 (June 2018), p. 24. ISSN: 2218-6581. DOI: 10.3390/robotics7020024. URL: https://www.mdpi.com/2218-6581/7/2/24 (visited on 03/15/2024).

[Hu+18b] Yuanming Hu et al. *ChainQueen: A Real-Time Differentiable Physical Simulator for Soft Robotics*. Oct. 1, 2018. DOI: 10.48550/arXiv.1810.01054. arXiv: 1810.01054[cs]. URL: http://arxiv.org/abs/1810.01054 (visited on 07/22/2024).

[Hug+16] Josie Hughes et al. "Soft Manipulators and Grippers: A Review". In: *Frontiers in Robotics and AI* 3 (2016). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2016.00069 (visited on 12/29/2023).

[HW03] Michael W. Hannan and Ian D. Walker. "Kinematics and the Implementation of an Elephant's Trunk Manipulator and Other Continuum Style Robots". In: *Journal of Robotic Systems* 20.2 (Feb. 2003), pp. 45–63. ISSN: 0741-2223, 1097-4563. DOI: 10.1002/rob.10070. URL: https://onlinelibrary.wiley.com/doi/10.1002/rob.10070 (visited on 06/06/2024).

[Jia+16] Chenfanfu Jiang et al. "The material point method for simulating continuum materials". In: *ACM SIGGRAPH 2016 Courses*. SIGGRAPH '16. New York, NY, USA: Association for Computing Machinery, July 24, 2016, pp. 1–52. ISBN: 9781450342896. DOI: 10.1145/2897826.2927348. URL: https://doi.org/10.1145/2897826.2927348 (visited on 02/08/2024).

[JL18] Donghwa Jeong and Kiju Lee. "Design and analysis of an origami-based three-finger manipulator". In: *Robotica* 36.2 (Feb. 2018), pp. 261–274. ISSN: 0263-5747, 1469-8668. DOI: 10.1017/S0263574717000340. URL: https://www.cambridge.org/core/journals/robotica/article/design-and-analysis-of-an-origamibased-threefinger-manipulator/EE67BBCEB45E64C292E287A7F931CA59 (visited on 04/05/2024).

[Jun+11] Jinwoo Jung et al. "A modeling approach for continuum robotic manipulators: Effects of nonlinear internal device friction". In: *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. ISSN: 2153-0866. Sept. 2011, pp. 5139–5146. DOI: 10.1109/IROS.2011.6094941. URL: https://ieeexplore.ieee.org/document/6094941 (visited on 02/14/2024).

[Kar+20] Lennart Karstensen et al. "Autonomous guidewire navigation in a two dimensional vascular phantom". In: *Current Directions in Biomedical Engineering* 6.1 (May 1, 2020). ISSN: 2364-5504. DOI: 10.1515/cdbme-2020-0007. URL: https://www.degruyter.com/document/doi/10.1515/cdbme-2020-0007/html (visited on 07/05/2024).

[KBP13] Jens Kober, J. Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey". In: *The International Journal of Robotics Research* 32.11 (Sept. 2013), pp. 1238–1274. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364913495721. URL: http://journals.sagepub.com/doi/10.1177/0278364913495721 (visited on 06/10/2024).

[KH04] N. Koenig and A. Howard. "Design and use paradigms for Gazebo, an open-source multi-robot simulator". In: *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566). Vol. 3. Sept. 2004, 2149–2154 vol.3. DOI: 10.1109/IROS.2004.1389727. URL: https://ieeexplore.ieee.org/document/1389727 (visited on 07/05/2024).

[Kha+12] Yaariv Khaykin et al. "Point-by-point pulmonary vein antrum isolation guided by intracardiac echocardiography and 3D mapping and duty-cycled multipolar AF ablation: effect of multipolar ablation on procedure duration and fluoroscopy time". In: *Journal of Interventional Cardiac Electrophysiology* 34.3 (Sept. 1, 2012), pp. 303–310. ISSN: 1572-8595. DOI: 10.1007/s10840-012-9676-3. URL: https://doi.org/10.1007/s10840-012-9676-3 (visited on 04/10/2024).

[Kim+19] Yoonho Kim et al. "Ferromagnetic soft continuum robots". In: *Science Robotics* 4.33 (Aug. 21, 2019), eaax7329. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aax7329. URL: https://www.science.org/doi/10.1126/scirobotics.aax7329 (visited on 04/10/2024).

[Kim+21] Daekyum Kim et al. "Review of machine learning methods in soft robotics". In: *PLOS ONE* 16.2 (2021), e0246102. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0246102. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0246102 (visited on 02/08/2024).

[Kob+23] Hiroki Kobayashi et al. *Computational synthesis of locomoting soft robots by topology optimization*. Synthical. Oct. 17, 2023. URL: https://synthical.com/article/446ec90e-c896-486f-a02f-9fea505f6e4a (visited on 02/08/2024).

[Koe+23] Margaret Koehler et al. “Modeling and Control of a 5-DOF Parallel Continuum Haptic Device”. In: *IEEE Transactions on Robotics* 39.5 (Oct. 2023), pp. 3636–3654. ISSN: 1941-0468. DOI: 10.1109/TRO.2023.3277068. URL: https://ieeexplore.ieee.org/document/10145827 (visited on 07/05/2024).

[KP13] Mahta Khoshnam and Rajni V. Patel. “A pseudo-rigid-body 3R model for a steerable ablation catheter”. In: *2013 IEEE International Conference on Robotics and Automation*. 2013 IEEE International Conference on Robotics and Automation. ISSN: 1050-4729. May 2013, pp. 4427–4432. DOI: 10.1109/ICRA.2013.6631205. URL: https://ieeexplore.ieee.org/document/6631205 (visited on 02/08/2024).

[Kra+17] Louis B. Kratchman et al. “Guiding Elastic Rods With a Robot-Manipulated Magnet for Medical Applications”. In: *IEEE transactions on robotics: a publication of the IEEE Robotics and Automation Society* 33.1 (Feb. 2017), pp. 227–233. ISSN: 1552-3098. DOI: 10.1109/TRO.2016.2623339.

[Kri+20] Sam Kriegman et al. “Scalable sim-to-real transfer of soft robot designs”. In: *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)* (May 2020). DOI: 10.1109/RoboSoft48309.2020.9116004. URL: https://par.nsf.gov/biblio/10197965-scalable-sim-real-transfer-soft-robot-designs (visited on 03/11/2024).

[KSS18] Prabhat Kumar, Anupam Saxena, and Roger A. Sauer. *Computational synthesis of large deformation compliant mechanisms undergoing self and mutual contact*. ADS Bibcode: 2018arXiv180206049K. Feb. 1, 2018. DOI: 10.48550/arXiv.1802.06049. URL: https://ui.adsabs.harvard.edu/abs/2018arXiv180206049K (visited on 12/29/2023).

[Kut+11] Michael D.M. Kutzer et al. “Design of a new cable-driven manipulator with a large open lumen: Preliminary applications in the minimally-invasive removal of osteolysis”. In: *2011 IEEE International Conference on Robotics and Automation*. 2011 IEEE International Conference on Robotics and Automation. ISSN: 1050-4729. May 2011, pp. 2913–2920. DOI: 10.1109/ICRA.2011.5980285. URL: https://ieeexplore.ieee.org/document/5980285 (visited on 02/08/2024).

[KW19] Diederik P. Kingma and Max Welling. “An Introduction to Variational Autoencoders”. In: *Foundations and Trends® in Machine Learning* 12.4 (2019), pp. 307–392. ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000056. arXiv: 1906.02691[cs,stat]. URL: http://arxiv.org/abs/1906.02691 (visited on 07/19/2024).

[Lab] D. I. Labs. *Soft Robotics AI Enabled Automation Solutions Combing 3D Vision, Artificial Intelligence and Soft Grasping*. Soft Robotics. URL: https://www.softroboticsinc.com/ (visited on 06/10/2024).

[Las+12] Cecilia Laschi et al. “Soft Robot Arm Inspired by the Octopus”. In: *Advanced Robotics* 26.7 (Jan. 2012), pp. 709–727. ISSN: 0169-1864, 1568-5535. DOI: 10.1163/156855312X626343. URL: https://www.tandfonline.com/doi/full/10.1163/156855312X626343 (visited on 06/06/2024).

[Las22] Cecilia Laschi. “Embodied Intelligence in soft robotics: joys and sorrows”. In: *IOP Conference Series: Materials Science and Engineering* 1261.1 (Oct. 2022), p. 012002. ISSN: 1757-899X. DOI: 10.1088/1757-899X/1261/1/012002. URL: https://dx.doi.org/10.1088/1757-899X/1261/1/012002 (visited on 06/06/2024).

[Lee+20] Hajun Lee et al. “3D-printed programmable tensegrity for soft robotics”. In: *Science Robotics* 5.45 (Aug. 26, 2020), eaay9024. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aay9024. URL: https://www.science.org/doi/10.1126/scirobotics.aay9024 (visited on 04/05/2024).

[Les+18] Timothée Lesort et al. “State representation learning for control: An overview”. In: *Neural Networks* 108 (Dec. 1, 2018), pp. 379–392. ISSN: 0893-6080. DOI: 10.1016/j.neunet.2018.07.006. URL: https://www.sciencedirect.com/science/article/pii/S0893608018302053 (visited on 03/06/2024).

[Li+19] Shuo Li et al. “Bio-inspired Design and Additive Manufacturing of Soft Materials, Machines, Robots, and Haptic Interfaces”. In: *Angewandte Chemie International Edition* 58.33 (Aug. 12, 2019), pp. 11182–11204. ISSN: 1433-7851, 1521-3773. DOI: 10.1002/anie.201813402. URL: https://onlinelibrary.wiley.com/doi/10.1002/anie.201813402 (visited on 03/22/2024).

[lin+] Get link et al. *The Art of Evolution; Theo Jansen's Strandbeests*. URL: https://www.luxartasia.com/2018/07/theo-jansen-strandbeests-evolution-interview.html (visited on 06/06/2024).

[Liu+17] Ying Liu et al. “Sequential self-folding of polymer sheets”. In: *Science Advances* 3.3 (Mar. 2017), e1602417. ISSN: 2375-2548. DOI: 10.1126/sciadv.1602417.

[Liu+18a] Chih-Hsing Liu et al. “Optimal Design of a Soft Robotic Gripper for Grasping Unknown Objects”. In: *Soft Robotics* 5.4 (Aug. 28, 2018), pp. 452–465. ISSN: 2169-5172, 2169-5180. DOI: 10.1089/soro.2017.0121. URL: https://www.liebertpub.com/doi/10.1089/soro.2017.0121 (visited on 04/18/2023).

[Liu+18b] Yan 1 Liu et al. “Spectral Collocation Method in the Large Deformation Analysis of Flexible Beam”. In: (Nov. 2018), pp. 1–8. ISSN: 1992-9978. URL: https://www.proquest.com/docview/2317017248?fromopenview=true&pq-origsite=gscholar&sourcetype=Scholarly%20Journals (visited on 03/01/2024).

[LPB17] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. Nov. 3, 2017. DOI: 10.48550/arXiv.1612.01474. arXiv: 1612.01474[cs,stat]. URL: http://arxiv.org/abs/1612.01474 (visited on 06/19/2024).

[LXZ23] Haihong Li, Lingxiao Xun, and Gang Zheng. “Piecewise Linear Strain Cosserat Model for Soft Slender Manipulator”. In: *IEEE Transactions on Robotics* 39.3 (June 2023), pp. 2342–2359. ISSN: 1941-0468. DOI: 10.1109/TRO.2023.3236942. URL: https://ieeexplore.ieee.org/document/10027557 (visited on 07/05/2024).

[LYH11] Shutao Li, Bin Yang, and Jianwen Hu. “Performance comparison of different multi-resolution transforms for image fusion”. In: *Information Fusion* 12.2 (Apr. 1, 2011), pp. 74–84. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2010.03.002. URL: https://www.sciencedirect.com/science/article/pii/S1566253510000382 (visited on 07/19/2024).

[Ma+17] Li-Ke Ma et al. “Computational design and fabrication of soft pneumatic objects with desired deformations”. In: *ACM Transactions on Graphics* 36.6 (Nov. 20, 2017), pp. 1–12. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3130800.3130850. URL: https://dl.acm.org/doi/10.1145/3130800.3130850 (visited on 09/20/2021).

[Ma+21] Pingchuan Ma et al. "DiffAqua: A Differentiable Computational Design Pipeline for Soft Underwater Swimmers with Shape Interpolation". In: *ACM Transactions on Graphics* 40.4 (Aug. 31, 2021), pp. 1–14. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/3450626.3459832. arXiv: 2104.00837[cs]. URL: http://arxiv.org/abs/2104.00837 (visited on 01/05/2024).

[Mac+14] Miles Macklin et al. "Unified particle physics for real-time applications". In: *ACM Transactions on Graphics* 33.4 (July 27, 2014), 153:1–153:12. ISSN: 0730-0301. DOI: 10.1145/2601097.2601152. URL: https://doi.org/10.1145/2601097.2601152 (visited on 02/15/2024).

[Maj14] Carmel Majidi. "Soft Robotics: A Perspective—Current Trends and Prospects for the Future". In: *Soft Robotics* 1.1 (Mar. 2014), pp. 5–11. ISSN: 2169-5172, 2169-5180. DOI: 10.1089/soro.2013.0001. URL: https://www.liebertpub.com/doi/10.1089/soro.2013.0001 (visited on 06/07/2024).

[Mar+14] Andrew D. Marchese et al. "Design and control of a soft and continuously deformable 2D robotic manipulation system". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. 2014 IEEE International Conference on Robotics and Automation (ICRA). ISSN: 1050-4729. May 2014, pp. 2189–2196. DOI: 10.1109/ICRA.2014.6907161. URL: https://ieeexplore.ieee.org/document/6907161 (visited on 02/08/2024).

[Mar+16] Andrew G. Mark et al. "Auxetic metamaterial simplifies soft robot design". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)* (May 2016), pp. 4951–4956. DOI: 10.1109/ICRA.2016.7487701. URL: http://ieeexplore.ieee.org/document/7487701/ (visited on 04/05/2024).

[Mat+23] Anup Teejo Mathew et al. "SoRoSim: A MATLAB Toolbox for Hybrid Rigid–Soft Robots Based on the Geometric Variable-Strain Approach". In: *IEEE Robotics & Automation Magazine* 30.3 (Sept. 2023), pp. 106–122. ISSN: 1558-223X. DOI: 10.1109/MRA.2022.3202488. URL: https://ieeexplore.ieee.org/abstract/document/9895355 (visited on 01/05/2024).

[Maz+17] Ali Maziz et al. "Knitting and weaving artificial muscles". In: *Science Advances* 3.1 (Jan. 6, 2017), e1600327. ISSN: 2375-2548. DOI: 10.1126/sciadv.1600327. URL: https://www.science.org/doi/10.1126/sciadv.1600327 (visited on 04/05/2024).

[MC04] Michael Mascagni and Hongmei Chi. "On the Scrambled Halton Sequence". In: 10.3 (Dec. 1, 2004), pp. 435–442. ISSN: 1569-3961. DOI: 10.1515/mcma.2004.10.3-4.435. URL: https://www.degruyter.com/document/doi/10.1515/mcma.2004.10.3-4.435/html (visited on 05/17/2024).

[MCC16] Mariangela Manti, Vito Cacucciolo, and Matteo Cianchetti. "Stiffening in Soft Robotics: A Review of the State of the Art". In: *IEEE Robotics & Automation Magazine* 23.3 (Sept. 2016), pp. 93–106. ISSN: 1558-223X. DOI: 10.1109/MRA.2016.2582718. URL: https://ieeexplore.ieee.org/document/7565718 (visited on 12/29/2023).

[Men+22] Gianmarco Mengaldo et al. "A concise guide to modelling the physics of embodied intelligence in soft robotics". In: *Nature Reviews Physics* 4.9 (Sept. 2022), pp. 595–610. ISSN: 2522-5820. DOI: 10.1038/s42254-022-00481-z. URL: https://www.nature.com/articles/s42254-022-00481-z (visited on 02/09/2024).

[Mén+23] Etienne Ménager et al. “Direct and inverse modeling of soft robots by learning a condensed FEM model”. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. 2023 IEEE International Conference on Robotics and Automation (ICRA). London, United Kingdom: IEEE, May 29, 2023, pp. 530–536. ISBN: 9798350323658. DOI: 10.1109/ICRA48891.2023.10161537. URL: https://ieeexplore.ieee.org/document/10161537/ (visited on 05/17/2024).

[Mén+24] Etienne Ménager et al. “Condensed semi-implicit dynamics for trajectory optimization in soft robotics”. In: *IEEE International Conference on Soft Robotics (RoboSoft)*. San Diego (CA), United States: IEEE, Apr. 2024. URL: https://hal.science/hal-04466639 (visited on 02/29/2024).

[MMD19] Thomas Morzadec, Damien Marcha, and Christian Duriez. “Toward Shape Optimization of Soft Robots”. In: *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. 2019 2nd IEEE International Conference on Soft Robotics (RoboSoft). Apr. 2019, pp. 521–526. DOI: 10.1109/ROBOSOFT.2019.8722822. URL: https://ieeexplore.ieee.org/document/8722822 (visited on 01/05/2024).

[Mor+20a] Thor Morales Bieze et al. “Design, implementation, and control of a deformable manipulator robot based on a compliant spine”. In: *The International Journal of Robotics Research* 39.14 (Dec. 2020), pp. 1604–1619. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364920910487. URL: http://journals.sagepub.com/doi/10.1177/0278364920910487 (visited on 12/29/2023).

[Mor+20b] Alexander Mordvintsev et al. “Growing Neural Cellular Automata”. In: *Distill* 5.2 (Feb. 11, 2020), e23. ISSN: 2476-0757. DOI: 10.23915/distill.00023. URL: https://distill.pub/2020/growing-ca (visited on 06/13/2024).

[Mos+14] Bobak Mosadegh et al. “Pneumatic Networks for Soft Robotics that Actuate Rapidly”. In: *Advanced Functional Materials* 24.15 (Apr. 2014), pp. 2163–2170. ISSN: 1616-301X, 1616-3028. DOI: 10.1002/adfm.201303288. URL: https://onlinelibrary.wiley.com/doi/10.1002/adfm.201303288 (visited on 12/29/2023).

[MPD23] Etienne Ménager, Quentin Peyron, and Christian Duriez. *Toward the use of proxies for efficient learning manipulation and locomotion strategies on soft robots*. Oct. 25, 2023. DOI: 10.48550/arXiv.2310.17029. arXiv: 2310.17029[cs]. URL: http://arxiv.org/abs/2310.17029 (visited on 07/05/2024).

[Nava] Tanguy Navez. *Soft Gripper using Self-Contacts*. URL: https://www.youtube.com/watch?v=hq81VX2wWvA (visited on 12/29/2023).

[Navb] Tanguy Navez. *SoftRobots.DesignOptimization*. URL: https://github.com/SofaDefrost/SoftRobots.DesignOptimization.

[Navc] Tanguy Navez. *SoftRobots.DesignOptimization plugin for SOFA*. GitHub. URL: https://github.com/SofaDefrost/SoftRobots.DesignOptimization.

[Nav+19] Stefan Escaida Navarro et al. “Modeling Novel Soft Mechanosensors Based on Air-Flow Measurements”. In: *IEEE Robotics and Automation Letters* 4.4 (Oct. 2019), pp. 4338–4345. ISSN: 2377-3766. DOI: 10.1109/LRA.2019.2932604. URL: https://ieeexplore.ieee.org/abstract/document/8784163 (visited on 12/28/2023).

[Nav+20] Stefan Escaida Navarro et al. “A Model-Based Sensor Fusion Approach for Force and Shape Estimation in Soft Robotics”. In: *IEEE Robotics and Automation Letters* 5.4 (Oct. 2020), pp. 5621–5628. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.3008120. URL: https://ieeexplore.ieee.org/document/9137639 (visited on 12/28/2023).

[Nav+23] Stefan Escaida Navarro et al. “An Open Source Design Optimization Toolbox Evaluated on a Soft Finger”. In: *IEEE Robotics and Automation Letters* 8.9 (Sept. 2023), pp. 6044–6051. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2023.3301272. URL: https://ieeexplore.ieee.org/document/10202189/ (visited on 08/17/2023).

[Nav+24] Tanguy Navez et al. “Design Optimization of a Soft Gripper Using Self-Contacts”. In: *2024 IEEE 7th International Conference on Soft Robotics (RoboSoft)*. 2024 IEEE 7th International Conference on Soft Robotics (RoboSoft). ISSN: 2769-4534. Apr. 2024, pp. 1054–1060. DOI: 10.1109/RoboSoft60065.2024.10521925. URL: https://ieeexplore.ieee.org/abstract/document/10521925 (visited on 07/09/2024).

[Ols+20] Gina Olson et al. “An Euler–Bernoulli beam model for soft robot arms bent through self-stress and external loads”. In: *International Journal of Solids and Structures* 207 (Dec. 15, 2020), pp. 113–131. ISSN: 0020-7683. DOI: 10.1016/j.ijsolstr.2020.09.015. URL: https://www.sciencedirect.com/science/article/pii/S002076832030353X (visited on 02/08/2024).

[Oza+20] Yoshihiko Ozaki et al. “Multiobjective tree-structured parzen estimator for computationally expensive optimization problems”. In: *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. GECCO ’20. New York, NY, USA: Association for Computing Machinery, June 26, 2020, pp. 533–541. ISBN: 9781450371285. DOI: 10.1145/3377930.3389817. URL: https://dl.acm.org/doi/10.1145/3377930.3389817 (visited on 01/12/2024).

[Pag+20] Amir Pagoli et al. “Design and Optimization of a Dextrous Robotic Finger: Incorporating a Sliding, Rotating, and Soft-Bending Mechanism While Maximizing Dexterity and Minimizing Dimensions”. In: *IEEE Robotics & Automation Magazine* 27.4 (Dec. 2020), pp. 56–64. ISSN: 1558-223X. DOI: 10.1109/MRA.2020.3024283. URL: https://ieeexplore.ieee.org/document/9217429 (visited on 03/15/2024).

[Pai02] Dinesh K. Pai. “STRANDS: Interactive Simulation of Thin Solids using Cosserat Models”. In: *Computer Graphics Forum* 21.3 (Sept. 2002), pp. 347–352. ISSN: 01677055, 14678659. DOI: 10.1111/1467-8659.00594. URL: https://onlinelibrary.wiley.com/doi/10.1111/1467-8659.00594 (visited on 02/08/2024).

[Pet08] Marcus Pettersson. “Design Optimization in Industrial Robotics : Methods and Algorithms for Drive Train Design”. In: (2008). URL: https://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-11664 (visited on 07/11/2024).

[PH22] Joshua Pinskier and David Howard. “From Bioinspiration to Computer Generation: Developments in Autonomous Soft Robot Design”. In: *Advanced Intelligent Systems* 4.1 (Jan. 2022), p. 2100086. ISSN: 2640-4567, 2640-4567. DOI: 10.1002/aisy.202100086. URL: https://onlinelibrary.wiley.com/doi/10.1002/aisy.202100086 (visited on 01/05/2024).

[Pin+23] Josh Pinskier et al. “Automated design of pneumatic soft grippers through design-dependent multi-material topology optimization: 2023 IEEE International Conference on Soft Robotics, RoboSoft 2023”. In: *Proceedings of the IEEE International Conference on Soft Robotics, RoboSoft 2023* (2023). DOI: 10.1109/RoboSoft55895.2023.10122069. URL: http://www.scopus.com/inward/record.url?scp=85160555571&partnerID=8YFLogxK (visited on 03/13/2024).

[Pin+24] Josh Pinskier et al. “Diversity-Based Topology Optimization of Soft Robotic Grippers”. In: *Advanced Intelligent Systems* (Jan. 18, 2024), p. 2300505. ISSN: 2640-4567, 2640-4567. DOI: 10.1002/aisy.202300505. URL: https://onlinelibrary.wiley.com/doi/10.1002/aisy.202300505 (visited on 03/13/2024).

[PRZ+23] Flavie PRZYBYLSKI et al. “3D Kinematics and Quasi-Statics of a Growing Robot Eversion”. In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)*. 2023 IEEE International Conference on Soft Robotics (RoboSoft). ISSN: 2769-4534. Apr. 2023, pp. 1–6. DOI: 10.1109/RoboSoft55895.2023.10122073. URL: https://ieeexplore.ieee.org/document/10122073 (visited on 03/20/2024).

[Qin+23] Longhui Qin et al. “Modeling and Simulation of Dynamics in Soft Robotics: a Review of Numerical Approaches”. In: *Current Robotics Reports* (Aug. 19, 2023). ISSN: 2662-4087. DOI: 10.1007/s43154-023-00105-z. URL: https://doi.org/10.1007/s43154-023-00105-z (visited on 02/01/2024).

[RBS19] Ahmad Rafsanjani, Katia Bertoldi, and André R. Studart. “Programming soft robots with flexible mechanical metamaterials”. In: *Science Robotics* 4.29 (Apr. 10, 2019), eaav7874. ISSN: 2470-9476. DOI: 10.1126/scirobotics.aav7874. URL: https://www.science.org/doi/10.1126/scirobotics.aav7874 (visited on 04/05/2024).

[Ren+15] F Renda et al. “Modelling cephalopod-inspired pulsed-jet locomotion for underwater soft robots”. In: *Bioinspiration & Biomimetics* 10.5 (Sept. 28, 2015), p. 055005. ISSN: 1748-3190. DOI: 10.1088/1748-3190/10/5/055005. URL: https://iopscience.iop.org/article/10.1088/1748-3190/10/5/055005 (visited on 02/14/2024).

[Ren+18a] Federico Renda et al. “A unified multi-soft-body dynamic model for underwater soft robots”. In: *The International Journal of Robotics Research* 37.6 (May 2018), pp. 648–666. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364918769992. URL: http://journals.sagepub.com/doi/10.1177/0278364918769992 (visited on 02/08/2024).

[Ren+18b] Federico Renda et al. “Discrete Cosserat Approach for Multi-Section Soft Robots Dynamics”. In: *IEEE Transactions on Robotics* 34.6 (Dec. 2018), pp. 1518–1533. ISSN: 1552-3098, 1941-0468. DOI: 10.1109/TRO.2018.2868815. arXiv: 1702.03660[cs]. URL: http://arxiv.org/abs/1702.03660 (visited on 02/08/2024).

[RK22] Markus Ryll and Robert K. Katzschmann. “SMORS: A soft multirotor UAV for multimodal locomotion and robust interaction”. In: *2022 International Conference on Robotics and Automation (ICRA)*. 2022 International Conference on Robotics and Automation (ICRA). May 2022, pp. 2010–2016. DOI: 10.1109/ICRA46639.2022.9812044. URL: https://ieeexplore.ieee.org/document/9812044 (visited on 05/17/2024).

[RNA22] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. “Deep Generative Models in Engineering Design: A Review”. In: *Journal of Mechanical Design* 144.7 (July 1, 2022), p. 071704. ISSN: 1050-0472, 1528-9001. DOI: 10.1115/1.4053859. URL: https://asmedigitalcollection.asme.org/mechanicaldesign/article/144/7/071704/1136676/Deep-Generative-Models-in-Engineering-Design-A (visited on 06/13/2024).

[Ros+19] Laura Ros-Freixedes et al. “Design optimization of a contact-aided continuum robot for endobronchial interventions based on anatomical constraints”. In: *International Journal of Computer Assisted Radiology and Surgery* 14.7 (July 1, 2019), pp. 1137–1146. ISSN: 1861-6429. DOI: 10.1007/s11548-019-01972-8. URL: https://doi.org/10.1007/s11548-019-01972-8 (visited on 12/29/2023).

[RPK19] M. Raissi, P. Perdikaris, and G. E. Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (Feb. 1, 2019), pp. 686–707. ISSN: 0021-9991. DOI: 10.1016/j.jcp.2018.10.045. URL: https://www.sciencedirect.com/science/article/pii/S0021999118307125 (visited on 07/05/2024).

[RPR17] Gundula Runge, Jan Peters, and Annika Raatz. "Design optimization of soft pneumatic actuators using genetic algorithms". In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO). Dec. 2017, pp. 393–400. DOI: 10.1109/ROBIO.2017.8324449. URL: https://ieeexplore.ieee.org/document/8324449 (visited on 03/15/2024).

[RT18] Daniela Rus and Michael T. Tolley. "Design, fabrication and control of origami robots". In: *Nature Reviews Materials* 3.6 (June 2018), pp. 101–112. ISSN: 2058-8437. DOI: 10.1038/s41578-018-0009-8. URL: https://www.nature.com/articles/s41578-018-0009-8 (visited on 04/05/2024).

[Sac+21] Ela Sachyani Keneth et al. "3D Printing Materials for Soft Robotics". In: *Advanced Materials* 33.19 (May 2021), p. 2003387. ISSN: 0935-9648, 1521-4095. DOI: 10.1002/adma.202003387. URL: https://onlinelibrary.wiley.com/doi/10.1002/adma.202003387 (visited on 04/04/2024).

[Sad+22] S. M. Hadi Sadati et al. "Embodied Intelligence & Morphological Computation in Soft Robotics Community: Collaborations, Coordination, and Perspective". In: *IOP Conference Series: Materials Science and Engineering* 1261.1 (Oct. 1, 2022), p. 012005. ISSN: 1757-899X. DOI: 10.1088/1757-899X/1261/1/012005. URL: https://iopscience.iop.org/article/10.1088/1757-899X/1261/1/012005/meta (visited on 06/06/2024).

[Sad+23] S.M.H. Sadati et al. "Reduced order modeling and model order reduction for continuum manipulators: an overview". In: *Frontiers in Robotics and AI* 10 (2023). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2023.1094114 (visited on 03/01/2024).

[Sal01] Jean Salençon. *Handbook of Continuum Mechanics*. Springer, 2001. DOI: 10.1007/978-3-642-56542-7. URL: https://hal.science/hal-00112704 (visited on 07/05/2024).

[San+23] Vanessa Sanchez et al. "3D Knitting for Pneumatic Soft Robotics". In: *Advanced Functional Materials* 33.26 (June 2023), p. 2212541. ISSN: 1616-301X, 1616-3028. DOI: 10.1002/adfm.202212541. URL: https://onlinelibrary.wiley.com/doi/10.1002/adfm.202212541 (visited on 04/05/2024).

[Sch+23] Pierre Schegg et al. "SofaGym: An Open Platform for Reinforcement Learning Based on Soft Robot Simulations". In: *Soft Robotics* 10.2 (Apr. 1, 2023), pp. 410–430. ISSN: 2169-5172, 2169-5180. DOI: 10.1089/soro.2021.0123. URL: https://www.liebertpub.com/doi/10.1089/soro.2021.0123 (visited on 06/20/2024).

[SD22] Pierre Schegg and Christian Duriez. "Review on generic methods for mechanical modeling, simulation and control of soft robots". In: *PLOS ONE* 17.1 (Jan. 14, 2022), e0251059. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0251059. URL: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0251059 (visited on 01/24/2024).

[SF89] J. C. Simo and D. D. Fox. "On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parametrization". In: *Computer Methods in Applied Mechanics and Engineering* 72.3 (Mar. 1, 1989), pp. 267–304. ISSN: 0045-7825. DOI: 10.1016/0045-7825(89)90002-9. URL: https://www.sciencedirect.com/science/article/pii/0045782589900029 (visited on 02/28/2024).

[SG05] Linda Smith and Michael Gasser. "The development of embodied cognition: six lessons from babies". In: *Artificial Life* 11.1 (2005), pp. 13–29. ISSN: 1064-5462. DOI: 10.1162/1064546053278973.

[SH23] Francesco Stella and Josie Hughes. "The science of soft robot design: A review of motivations, methods and enabling technologies". In: *Frontiers in Robotics and AI* 9 (Jan. 18, 2023), p. 1059026. ISSN: 2296-9144. DOI: 10.3389/frobt.2022.1059026. URL: https://www.frontiersin.org/articles/10.3389/frobt.2022.1059026/full (visited on 05/17/2023).

[Sha97] A.A. Shabana. "Definition of the Slopes and the Finite Element Absolute Nodal Coordinate Formulation". In: *Multibody System Dynamics* 1.3 (Sept. 1, 1997), pp. 339–348. ISSN: 1573-272X. DOI: 10.1023/A:1009740800463. URL: https://doi.org/10.1023/A:1009740800463 (visited on 02/01/2024).

[She+11] Robert F. Shepherd et al. "Multigait soft robot". In: *Proceedings of the National Academy of Sciences* 108.51 (Dec. 20, 2011), pp. 20400–20403. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1116564108. URL: https://pnas.org/doi/full/10.1073/pnas.1116564108 (visited on 01/05/2024).

[Shi+23] Jialei Shi et al. "Characterisation and control platform for pneumatically driven soft robots: Design and applications". In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)*. 2023 IEEE International Conference on Soft Robotics (RoboSoft). ISSN: 2769-4534. Apr. 2023, pp. 1–8. DOI: 10.1109/RoboSoft55895.2023.10122041. URL: https://ieeexplore.ieee.org/document/10122041 (visited on 07/25/2024).

[Sig97] Ole Sigmund. "On the Design of Compliant Mechanisms Using Topology Optimization*". In: *Mechanics of Structures and Machines* 25.4 (Jan. 1997), pp. 493–524. ISSN: 0890-5452. DOI: 10.1080/08905459708945415. URL: http://www.tandfonline.com/doi/abs/10.1080/08905459708945415 (visited on 03/13/2024).

[Sim94] Karl Sims. "Evolving virtual creatures". In: *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*. SIGGRAPH '94. New York, NY, USA: Association for Computing Machinery, July 24, 1994, pp. 15–22. ISBN: 9780897916677. DOI: 10.1145/192161.192167. URL: https://dl.acm.org/doi/10.1145/192161.192167 (visited on 03/06/2024).

[Sit21] Metin Sitti. "Physical intelligence as a new paradigm". In: *Extreme Mechanics Letters* 46 (July 1, 2021), p. 101340. ISSN: 2352-4316. DOI: 10.1016/j.eml.2021.101340. URL: https://www.sciencedirect.com/science/article/pii/S2352431621001012 (visited on 06/06/2024).

[SIT91] K. Suzumori, S. Iikura, and H. Tanaka. "Development of flexible microactuator and its applications to robotic mechanisms". In: *1991 IEEE International Conference on Robotics and Automation Proceedings*. 1991 IEEE International Conference on Robotics and Automation Proceedings. Apr. 1991, 1622–1627 vol.2. DOI: 10.1109/ROBOT.1991.131850. URL: https://ieeexplore.ieee.org/document/131850 (visited on 06/07/2024).

[SK22] Filippo A. Spinelli and Robert K. Katzschmann. "A Unified and Modular Model Predictive Control Framework for Soft Continuum Manipulators under Internal and External Constraints". In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). ISSN: 2153-0866. Oct. 2022, pp. 9393–9400. DOI: 10.1109/IROS47612.2022.9981702. URL: https://ieeexplore.ieee.org/document/9981702 (visited on 07/05/2024).

[Sko+13] Mélina Skouras et al. "Computational design of actuated deformable characters". In: *ACM Transactions on Graphics* 32.4 (July 21, 2013), pp. 1–10. ISSN: 0730-0301, 1557-7368. DOI: 10.1145/2461912.2461979. URL: https://dl.acm.org/doi/10.1145/2461912.2461979 (visited on 09/20/2021).

[SM13] Ole Sigmund and Kurt Maute. "Topology optimization approaches". In: *Structural and Multidisciplinary Optimization* 48.6 (Dec. 1, 2013), pp. 1031–1055. ISSN: 1615-1488. DOI: 10.1007/s00158-013-0978-6. URL: https://doi.org/10.1007/s00158-013-0978-6 (visited on 03/13/2024).

[SM19] Ralf C. Staudemeyer and Eric Rothstein Morris. *Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks*. Sept. 12, 2019. DOI: 10.48550/arXiv.1909.09586. arXiv: 1909.09586[cs]. URL: http://arxiv.org/abs/1909.09586 (visited on 06/14/2024).

[SNW17] Zhong Shen, Junhan Na, and Zheng Wang. "A Biomimetic Underwater Soft Robot Inspired by Cephalopod Mollusc". In: *IEEE Robotics and Automation Letters* 2.4 (Oct. 2017), pp. 2217–2223. ISSN: 2377-3766. DOI: 10.1109/LRA.2017.2724760. URL: https://ieeexplore.ieee.org/document/7973024 (visited on 05/17/2024).

[Spa+21] Bjorn Sparrman et al. "Printed silicone pneumatic actuators for soft robotics". In: *Additive Manufacturing* 40 (Apr. 1, 2021), p. 101860. ISSN: 2214-8604. DOI: 10.1016/j.addma.2021.101860. URL: https://www.sciencedirect.com/science/article/pii/S2214860421000257 (visited on 04/04/2024).

[Spi+19] Andrew Spielberg et al. "Learning-In-The-Loop Optimization: End-To-End Control And Co-Design Of Soft Robots Through Learned Deep Latent Representations". In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/hash/438124b4c06f3a5caffab2c07863b617-Abstract.html (visited on 03/20/2024).

[Spi+21] Andrew Spielberg et al. "Co-Learning of Task and Sensor Placement for Soft Robotics". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 1208–1215. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3056369. URL: https://ieeexplore.ieee.org/document/9345345 (visited on 01/05/2024).

[Spi+23] Andrew Spielberg et al. "Advanced soft robot modeling in ChainQueen". In: *Robotica* 41.1 (Jan. 2023), pp. 74–104. ISSN: 0263-5747, 1469-8668. DOI: 10.1017/S0263574721000722. URL: https://www.cambridge.org/core/journals/robotica/article/advanced-soft-robot-modeling-in-chainqueen/FACC977AC73D95935AB7122362C3CB70 (visited on 02/08/2024).

[SSW22] Charles Schaff, Audrey Sedal, and Matthew R. Walter. "Soft Robots Learn to Crawl: Jointly Optimizing Design and Control with Sim-to-Real Transfer". In: *Robotics: Science and Systems XVIII* (June 27, 2022). DOI: 10.15607/RSS.2022.XVIII.062. URL: http://www.roboticsproceedings.org/rss18/p062.pdf (visited on 01/05/2024).

[Sui+20] Xin Sui et al. "Automatic Generation of Locomotion Patterns for Soft Modular Reconfigurable Robots". In: *Applied Sciences* 10.1 (Jan. 2020), p. 294. ISSN: 2076-3417. DOI: 10.3390/app10010294. URL: https://www.mdpi.com/2076-3417/10/1/294 (visited on 03/11/2024).

[Sun+23] Yilun Sun et al. "Design of topology optimized compliant legs for bio-inspired quadruped robots". In: *Scientific Reports* 13.1 (Mar. 25, 2023), p. 4875. ISSN: 2045-2322. DOI: 10.1038/s41598-023-32106-5. URL: https://www.nature.com/articles/s41598-023-32106-5 (visited on 03/13/2024).

[SV88] J. C. Simo and L. Vu-Quoc. "On the dynamics in space of rods undergoing large motions — A geometrically exact approach". In: *Computer Methods in Applied Mechanics and Engineering* 66.2 (Feb. 1, 1988), pp. 125–161. ISSN: 0045-7825. DOI: 10.1016/0045-7825(88)90073-4. URL: https://www.sciencedirect.com/science/article/pii/0045782588900734 (visited on 02/28/2024).

[Tal+16] Hugo Talbot et al. "Interactive Training System for Interventional Electrocardiology Procedures". In: *Medical Image Analysis* 8789 (2016), p. 11. DOI: 10.1007/978-3-319-12057-7_2. URL: https://inria.hal.science/hal-01338346 (visited on 02/08/2024).

[Tan] Etienne Ménager Tanguy Navez. *CondensedFEMModel*. URL: https://github.com/SofaDefrost/CondensedFEMModel.

[Tap+20] Javier Tapia et al. "MakeSense: Automated Sensor Design for Proprioceptive Soft Robots". In: *Soft Robotics* 7.3 (June 2020), pp. 332–345. ISSN: 2169-5172. DOI: 10.1089/soro.2018.0162. URL: https://www.liebertpub.com/doi/abs/10.1089/soro.2018.0162 (visited on 01/05/2024).

[TCR21] Ryan L. Truby, Lillian Chin, and Daniela Rus. "A Recipe for Electrically-Driven Soft Robots via 3D Printed Handed Shearing Auxetics". In: *IEEE Robotics and Automation Letters* 6.2 (Apr. 2021), pp. 795–802. ISSN: 2377-3766, 2377-3774. DOI: 10.1109/LRA.2021.3052422. URL: https://ieeexplore.ieee.org/document/9326362/ (visited on 04/05/2024).

[Tee+20] Clark B Teeple et al. "Multi-segment soft robotic fingers enable robust precision grasping". In: *The International Journal of Robotics Research* 39.14 (Dec. 2020), pp. 1647–1667. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364920910465. URL: http://journals.sagepub.com/doi/10.1177/0278364920910465 (visited on 12/29/2023).

[TET12] Emanuel Todorov, Tom Erez, and Yuval Tassa. "MuJoCo: A physics engine for model-based control". In: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. ISSN: 2153-0866. Oct. 2012, pp. 5026–5033. DOI: 10.1109/IROS.2012.6386109. URL: https://ieeexplore.ieee.org/document/6386109 (visited on 02/15/2024).

[Tey+21] Marc Teyssier et al. "Human-Like Artificial Skin Sensor for Physical Human-Robot Interaction". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. 2021 IEEE International Conference on Robotics and Automation (ICRA). ISSN: 2577-087X. May 2021, pp. 3626–3633. DOI: 10.1109/ICRA48506.2021.9561152. URL: https://ieeexplore.ieee.org/document/9561152 (visited on 04/10/2024).

[Thu+17] Thomas George Thuruthel et al. “Learning dynamic models for open loop predictive control of soft robotic manipulators”. In: *Bioinspiration & Biomimetics* 12.6 (Oct. 16, 2017), p. 066003. ISSN: 1748-3190. DOI: 10.1088/1748-3190/aa839f. URL: https://iopscience.iop.org/article/10.1088/1748-3190/aa839f (visited on 02/19/2024).

[Thu+18] Thomas George Thuruthel et al. “Stable Open Loop Control of Soft Robotic Manipulators”. In: *IEEE Robotics and Automation Letters* 3.2 (Apr. 2018), pp. 1292–1298. ISSN: 2377-3766. DOI: 10.1109/LRA.2018.2797241. URL: https://ieeexplore.ieee.org/document/8268050 (visited on 07/05/2024).

[Thu+19] Thomas George Thuruthel et al. “Model-Based Reinforcement Learning for Closed-Loop Dynamic Control of Soft Robotic Manipulators”. In: *IEEE Transactions on Robotics* 35.1 (Feb. 2019), pp. 124–134. ISSN: 1941-0468. DOI: 10.1109/TRO.2018.2878318. URL: https://ieeexplore.ieee.org/document/8531756 (visited on 02/19/2024).

[Tia+20] Jiawei Tian et al. “Designing Ferromagnetic Soft Robots (FerroSoRo) with Level-Set-Based Multiphysics Topology Optimization”. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2020 IEEE International Conference on Robotics and Automation (ICRA). ISSN: 2577-087X. May 2020, pp. 10067–10074. DOI: 10.1109/ICRA40945.2020.9197457. URL: https://ieeexplore.ieee.org/abstract/document/9197457 (visited on 03/13/2024).

[Tia+23] Sizhe Tian et al. *Multi-tap Resistive Sensing and FEM Modeling enables Shape and Force Estimation in Soft Robots*. Nov. 24, 2023. DOI: 10.48550/arXiv.2311.14566. arXiv: 2311.14566[cs]. URL: http://arxiv.org/abs/2311.14566 (visited on 07/06/2024).

[TSR20] Ryan L. Truby, Cosimo Della Santina, and Daniela Rus. “Distributed Proprioception of 3D Configuration in Soft, Sensorized Robots via Deep Learning”. In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 3299–3306. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.2976320. URL: https://ieeexplore.ieee.org/document/9013033 (visited on 04/05/2024).

[Van+20] Félix Vanneste et al. “Anisotropic Soft Robots Based on 3D Printed Meso-Structured Materials: Design, Modeling by Homogenization and Simulation”. In: *IEEE Robotics and Automation Letters* 5.2 (Apr. 2020), pp. 2380–2386. ISSN: 2377-3766. DOI: 10.1109/LRA.2020.2969926. URL: https://ieeexplore.ieee.org/abstract/document/8972411 (visited on 07/12/2024).

[Vas+23] Ashish Vaswani et al. *Attention Is All You Need*. Aug. 1, 2023. DOI: 10.48550/arXiv.1706.03762. arXiv: 1706.03762[cs]. URL: http://arxiv.org/abs/1706.03762 (visited on 05/17/2024).

[VGD21] Félix Vanneste, Olivier Goury, and Christian Duriez. “Enabling the control of a new degree of freedom by using anisotropic material on a 6-DOF parallel soft robot”. In: *2021 IEEE 4th International Conference on Soft Robotics (RoboSoft)*. 2021 IEEE 4th International Conference on Soft Robotics (RoboSoft). Apr. 2021, pp. 636–642. DOI: 10.1109/RoboSoft51838.2021.9479214. URL: https://ieeexplore.ieee.org/document/9479214 (visited on 04/05/2024).

[Wan+14] Wei Wang et al. “Locomotion of inchworm-inspired robot made of smart soft composite (SSC)”. In: *Bioinspiration & Biomimetics* 9.4 (Oct. 7, 2014), p. 046006. ISSN: 1748-3190. DOI: 10.1088/1748-3182/9/4/046006.

[Wan+15] Zhouyi Wang et al. “Biomechanics of gecko locomotion: the patterns of reaction forces on inverted, vertical and horizontal substrates”. In: *Bioinspiration & Biomimetics* 10.1 (Feb. 4, 2015), p. 016019. ISSN: 1748-3190. DOI: 10.1088/1748-3190/10/1/016019.

[Wan+20] Rixin Wang et al. “Topology optimization of a cable-driven soft robotic gripper”. In: *Structural and Multidisciplinary Optimization* 62.5 (Nov. 1, 2020), pp. 2749–2763. ISSN: 1615-1488. DOI: 10.1007/s00158-020-02619-y. URL: https://doi.org/10.1007/s00158-020-02619-y (visited on 10/30/2023).

[Wan+23] Tsun-Hsuan Wang et al. *DiffuseBot: Breeding Soft Robots With Physics-Augmented Generative Diffusion Models*. Nov. 28, 2023. DOI: 10.48550/arXiv.2311.17053. arXiv: 2311.17053[cs]. URL: http://arxiv.org/abs/2311.17053 (visited on 06/13/2024).

[WC22] Jue Wang and Alex Chortos. “Control Strategies for Soft Robot Systems”. In: *Advanced Intelligent Systems* 4.5 (May 2022), p. 2100165. ISSN: 2640-4567, 2640-4567. DOI: 10.1002/aisy.202100165. URL: https://onlinelibrary.wiley.com/doi/10.1002/aisy.202100165 (visited on 02/29/2024).

[WJ10] Robert J. Webster and Bryan A. Jones. “Design and Kinematic Modeling of Constant Curvature Continuum Robots: A Review”. In: *The International Journal of Robotics Research* 29.13 (Nov. 2010), pp. 1661–1683. ISSN: 0278-3649, 1741-3176. DOI: 10.1177/0278364910368147. URL: http://journals.sagepub.com/doi/10.1177/0278364910368147 (visited on 01/24/2024).

[WLK21] Xiaomei Wang, Yingqi Li, and Ka-Wai Kwok. “A Survey for Machine Learning-Based Control of Continuum Robots”. In: *Frontiers in Robotics and AI* 8 (2021). ISSN: 2296-9144. URL: https://www.frontiersin.org/articles/10.3389/frobt.2021.730330 (visited on 02/08/2024).

[YHM22] Yao Yao, Liang He, and Perla Maiolino. “A Simulation-Based Toolbox to Expedite the Digital Design of Bellow Soft Pneumatic Actuators”. In: *2022 IEEE 5th International Conference on Soft Robotics (RoboSoft)*. 2022 IEEE 5th International Conference on Soft Robotics (RoboSoft). Apr. 2022, pp. 29–34. DOI: 10.1109/RoboSoft54090.2022.9762153. URL: https://ieeexplore.ieee.org/document/9762153 (visited on 01/05/2024).

[YHM24] Yao Yao, Liang He, and Perla Maiolino. “SPADA: A Toolbox of Designing Soft Pneumatic Actuators for Shape Matching Based on Surrogate Modeling”. In: *Robotics Reports* 2.1 (Jan. 1, 2024), pp. 1–14. ISSN: 2835-0111. DOI: 10.1089/rorep.2023.0029. URL: https://www.liebertpub.com/doi/10.1089/rorep.2023.0029 (visited on 05/17/2024).

[YSY20] Yee Ling Yap, Swee Leong Sing, and Wai Yee Yeong. “A review of 3D printing processes and materials for soft robotics”. In: *Rapid Prototyping Journal* 26.8 (Jan. 1, 2020), pp. 1345–1361. ISSN: 1355-2546. DOI: 10.1108/RPJ-11-2019-0302. URL: https://doi.org/10.1108/RPJ-11-2019-0302 (visited on 04/04/2024).

[Yuh+23] Changyoung Yuhn et al. “4D topology optimization: Integrated optimization of the structure and self-actuation of soft bodies for dynamic motions”. In: (2023). DOI: 10.48550/ARXIV.2302.00905. URL: https://arxiv.org/abs/2302.00905 (visited on 12/12/2023).

[Ze+22] Qiji Ze et al. “Soft robotic origami crawler”. In: *Science Advances* 8.13 (Apr. 2022), eabm7834. ISSN: 2375-2548. DOI: 10.1126/sciadv.abm7834. URL: https://www.science.org/doi/10.1126/sciadv.abm7834 (visited on 04/05/2024).

[Zha+24] Zikai Zhao et al. “Exploring Embodied Intelligence in Soft Robotics: A Review”. In: *Biomimetics* 9.4 (Apr. 19, 2024), p. 248. ISSN: 2313-7673. DOI: 10.3390/biomimetics9040248. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11047907/ (visited on 06/06/2024).

[ZHD21] Jasan Zughaibi, Matthias Hofer, and Raffaello D’Andrea. *A Fast and Reliable Pick-and-Place Application with a Spherical Soft Robotic Arm*. Mar. 7, 2021. DOI: 10.48550/arXiv.2011.04624. arXiv: 2011.04624[cs,eess]. URL: http://arxiv.org/abs/2011.04624 (visited on 07/05/2024).

[Zho+21a] Cheng Zhou et al. “Ferromagnetic soft catheter robots for minimally invasive bioprinting”. In: *Nature Communications* 12.1 (Aug. 20, 2021), p. 5072. ISSN: 2041-1723. DOI: 10.1038/s41467-021-25386-w. URL: https://www.nature.com/articles/s41467-021-25386-w (visited on 06/10/2024).

[Zho+21b] Jie Zhou et al. *Graph Neural Networks: A Review of Methods and Applications*. Oct. 6, 2021. DOI: 10.48550/arXiv.1812.08434. arXiv: 1812.08434[cs,stat]. URL: http://arxiv.org/abs/1812.08434 (visited on 07/05/2024).

[Zhu+22] Mengjia Zhu et al. “Soft, Wearable Robotics and Haptics: Technologies, Trends, and Emerging Applications”. In: *Proceedings of the IEEE* 110.2 (Feb. 2022), pp. 246–272. ISSN: 1558-2256. DOI: 10.1109/JPROC.2021.3140049. URL: https://ieeexplore.ieee.org/document/9686045 (visited on 04/04/2024).

[ZSZ23] Chonghui Zhang, Audrey Sedal, and Yaoyao Fiona Zhao. “Differentiable Surrogate Models for Design and Trajectory Optimization of Auxetic Soft Robots”. In: *2023 IEEE International Conference on Soft Robotics (RoboSoft)*. 2023 IEEE International Conference on Soft Robotics (RoboSoft). ISSN: 2769-4534. Apr. 2023, pp. 1–8. DOI: 10.1109/RoboSoft55895.2023.10121968. URL: https://ieeexplore.ieee.org/document/10121968 (visited on 03/20/2024).

List of Tables

List of Figures