

# Contribution to the Kinematic Modeling and Control of Soft Manipulators using Computational Mechanics

PHD THESIS  
to obtain the title of

Doctor from Université des Sciences et  
Technologies de Lille  
Specialty: Robotics

Defended by  
Thor MORALES BIEZE

prepared at INRIA Lille, Defrost Team  
defended on October 24th, 2017

Jury

Supervisors: Pr. Rochdi MERZOUKI    - Université Lille 1  
Pr. Christian DURIEZ    - Inria Lille - Nord Europe

Reviewers: Pr. Véronique PERDEREAU    - Université Pierre et Marie Curie  
Pr. Kaspar ALTHOEFER    - Queen Mary University of London

Examiners: Pr. Pushparaj Mani PATHAK    - Indian Institute of Technology, Roorkee  
Pr. Abdelaziz BENALLEGUE    - Université de Versailles Saint-Quentin



---

## Contribution to Kinematic Modeling and Control of Soft Manipulators using Computational Mechanics

**Abstract:** This work provides new methods for the kinematic modeling and control of soft, continuum manipulators based on the Finite Element Method. Contrary to the case of rigid manipulators, soft and continuum manipulators generate their motion by deformation, therefore, the proposed methodology accounts for the deformation mechanics to better describe the kinematics of these type of robots. This methodology does not produce analytic solutions, instead, a numerical approximation is provided by methods derived from Computational Mechanics. The methodology is applied to a continuum manipulator, namely, the Compact Bionic Handling Assistant (CBHA). A closed-loop control scheme based on control allocation is also presented. The models and controller are validated experimentally.

**Keywords:** Continuum manipulators, Finite Element Method, Continuum kinematics, Continuum manipulator control, Soft robots, Continuum Mechanics, Computational Mechanics

---





*To my parents, Jessica and Jose Luis, for teaching me that a man,  
with love, can fly.*



*All models are wrong, but some are useful.*  
**George E. P. Box**



## Acknowledgments

First, I would like to dedicate this manuscript to my parents, Jessica Bieze and Jose Luis Morales, who never forget to send me their love every day, and to my three sisters Luisa, Jessica and Melissa.

I would like to thank my two supervisors, Rochdi Merzouki and Christian Duriez for believing in me. Thank you for your guidance and for sharing your knowledge and wisdom with me. I could not have better mentors and for that, I will be forever grateful.

I would like to thank also Jeremie Dequidt for being my friend and making my stay in the team so nice. Thanks also for being an example of how to dress well and promoting acceptable fashion in the team.

Many thanks to Alexandre Kruszewski for all the help he provided during the writing of this manuscript despite the crazy deadlines and for making me feel dumb every time we talk about automatic control because that pushes me to keep learning.

Many thanks to Pr. Gilberto Gonzales for pushing me to go out of my comfort zone and for caring always for my well being. Thanks to him for getting me involved in research in the beginning.

Many thanks to the members of DEFROST team, Frederick Largilliere, Mario Sanz Lopez, Eulalie Coevoet, Olivier Goury, Maxime Thieffry, Zhongkai Zhang, Bruno Carrez, Erwan Douaille, Marwa ElDiwiny, Piyush Jain, Damien Marchal, Sébastien Nelissen and Félix Vanneste for being alongside me in this adventure, thank you for your support, your patience and for laughing at my bad jokes. I will carry you all in my heart. Many thanks to Anne Rejl, for all the help with the administrative nightmare.

Many thanks to Sandra Hage Chehade and Camille Lihouck for giving me the opportunity to supervise their internships during their stay at INRIA and many thanks to all the students of the Bio-Inspired robotics lectures at Polytech Lille for being so receptive and patient towards me as a first time lecturer.

Many thanks to all the members of MOCIS team, in particular to Othman Lakhel and Achille Melingui for all their help in the experimental validation of my work.

Many thanks to the reviewers of my manuscript and jury members.

I would like to thank my dear friends Moises Vazquez and Missrraim Alvarez for all the years of friendship and for sending me the "good vibes" all the way across the Atlantic.

Last but not least, I would like to thank my girlfriend Marion Blasquez for her infinite love and patience and for becoming my family and giving me a loving home. She made this work possible and therefore, it is also hers.

Finally, I want to thank my cat Brisby for keeping me company during the nights while writing this manuscript.

## Acronyms

<b>AM</b>	- Additive Manufacturing
<b>API</b>	- Application Programming Interface
<b>CAD</b>	- Computer Assisted Design
<b>CBHA</b>	- Compact Bionic Handling Assistant
<b>CC</b>	- Constant Curvature (model)
<b>CGAL</b>	- Computational Geometry Algorithms Library
<b>CPU</b>	- Central Processing Unit
<b>DoF</b>	- Degrees of Freedom
<b>FBG</b>	- Fiber Bragg Grating
<b>FEM</b>	- Finite Element Method
<b>FKM</b>	- Forward Kinematic Model
<b>HA</b>	- Hybrid Approach (model)
<b>IKM</b>	- Inverse Kinematic Model
<b>KSI</b>	- Kinetic Sciences Inc.
<b>LMI</b>	- Linear Matrix Inequality
<b>ME</b>	- Maximum Error
<b>PDE</b>	- Partial differential Equation
<b>PI</b>	- Proportional Integrative (controller)
<b>QP</b>	- Quadratic Programming
<b>RMSD</b>	- Root-Mean-Square Deviation
<b>SOFA</b>	- Simulation Open Framework Architecture
<b>W-LAN</b>	- Wireless Local Area Network





## Glossary

$a_t$	-	Area of triangle
$\mathbf{C}$	-	Stiffness tensor
$\mathbf{d}_a$	-	Direction after cable guide
$\mathbf{d}_b$	-	Direction before cable guide
$e$	-	System error
$E$	-	Young's modulus
$E_{def}$	-	Energy of deformation
$\mathbf{F}$	-	Vector of internal forces
$\mathbf{F}_{ext}$	-	Vector of external forces
$\mathbf{F}^e$	-	Vector of element loads
$\mathbf{g}$	-	Gravity field
$G$	-	Shear modulus
$h$	-	Integrative term
$\mathbf{H}$	-	Matrix of constraints directions
$\mathbf{H}_a$	-	Matrix of actuators constraint directions
$\mathbf{H}_e$	-	Matrix of end-effector constraint directions
$\mathbf{H}_s$	-	Matrix of sensors constraint directions
$\mathbf{J}_{ea}$	-	Jacobian matrix between end-effector and actuators
$\mathbf{J}_{sa}$	-	Jacobian matrix between sensors and actuators
$\hat{\mathbf{J}}_{sa}$	-	Estimated Jacobian matrix between sensors and actuators
$\hat{\mathbf{J}}_{sa}^+$	-	Pseudo-inverse of the estimated Jacobian matrix between sensors and actuators
$k$	-	Spatial configuration
$k_i$	-	Integrative gain
$k_p$	-	Proportional gain
$\mathbf{K}$	-	Tangent stiffness matrix
$\mathbf{K}^e$	-	Element stiffness matrix
$m$	-	Constant total mass
$M$	-	Mapping function
$\mathbf{M}$	-	Inertia matrix
$\mathbf{n}$	-	Vector normal to a surface
$\mathbf{p}_{des}$	-	Desired position
$\mathbf{p}_{eff}$	-	End-effector position in free configuration
$\mathbf{P}$	-	Position vector
$S$	-	Set of triangles composing a cavity
$\mathbf{u}$	-	Displacement field
$\mathbf{u}^e$	-	Element displacement field

$v$	-	Poisson's ratio
$\mathbf{v}$	-	Closed-loop system control vector
$V$	-	Lyapunov candidate function
$\mathbf{W}$	-	Compliance matrix
$\mathbf{W}_{aa}$	-	Matrix of compliance (mechanical coupling) between actuators
$\mathbf{W}_{ea}$	-	Matrix of compliance (mechanical coupling) between end-effector and actuators
$\mathbf{W}_{sa}$	-	Matrix of compliance (mechanical coupling) between sensors and actuators
$\mathbf{x}$	-	Vector of generalized states
$\dot{\mathbf{x}}$	-	Velocity vector
$\hat{\mathbf{x}}$	-	Vector of estimated states
$\gamma$	-	Bounding parameter
$\Gamma$	-	Boundary surface
$\delta_a$	-	Cable length (cable actuator) or cavity volume (pneumatic actuator)
$\delta_e$	-	Vector representing the difference between current and desired position of end-effector
$\delta_s$	-	Sensor string length
$\delta_a^{free}$	-	Cable length (cable actuator) or cavity volume (pneumatic actuator) in free configuration
$\delta_e^{free}$	-	Vector representing the difference between current and desired position of end-effector in free configuration
$\delta_s^{free}$	-	Sensor string length in free configuration
$\Delta V$	-	Variation of the Lyapunov function
$\Delta \lambda_a$	-	Increment of actuators contribution
$\varepsilon$	-	Infinitesimal strain tensor
$\zeta_i$	-	Tetrahedron natural coordinates
$\lambda$	-	Vector of constraints contributions
$\lambda_a$	-	Vector of actuator constraint contributions
$\lambda_e$	-	Vector of end-effector constraint contributions
$\lambda_s$	-	Vector of sensor constraint contributions
$\phi_i$	-	Shape (interpolation) function associated to node $i$
$\Phi$	-	Total potential energy functional
$\sigma$	-	Cauchy stress tensor
$\omega$	-	Vector of inversion error
$\Omega$	-	Three-dimensional domain

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	General Introduction . . . . .	1
1.2	Framework and context . . . . .	2
1.3	Motivation . . . . .	2
1.4	Chapter summary . . . . .	3
1.5	List of Publications . . . . .	4
<b>2</b>	<b>State of the Art</b>	<b>7</b>
2.1	Introduction . . . . .	7
2.2	Continuum manipulator definition . . . . .	7
2.2.1	Soft continuum robot applications . . . . .	8
2.2.2	Bio-inspiration . . . . .	9
2.2.3	Classification . . . . .	10
2.3	State of the art in soft, continuum manipulators . . . . .	13
2.3.1	Design of continuum manipulators . . . . .	13
2.3.2	Modeling of continuum robots . . . . .	18
2.3.3	Dynamics and control of continuum robots . . . . .	27
2.4	Work contextualization and contributions . . . . .	29
<b>3</b>	<b>FEM-based model of Continuum Manipulators</b>	<b>31</b>
3.1	Introduction . . . . .	32
3.2	Continuum mechanics framework . . . . .	33
3.2.1	Constitutive material law . . . . .	34
3.2.2	Forces in the continuum manipulator . . . . .	35
3.3	Finite Element Method . . . . .	36
3.4	FEM-based kinematics of soft manipulators . . . . .	38
3.4.1	Constraint for the end-effector . . . . .	39
3.4.2	Actuator constraint model . . . . .	41
3.4.3	Sensor constraint model . . . . .	44
3.5	Reduced model in the constraint space . . . . .	45
3.5.1	Reduced compliance on the constraint space . . . . .	45
3.5.2	Coupled Kinematic Equations . . . . .	47
3.5.3	Inverse kinematic model solution by convex optimization . . . . .	48
3.6	Method implementation . . . . .	49
3.6.1	Simulation framework . . . . .	50
3.6.2	Corotational FEM . . . . .	50
3.6.3	Mesh generation . . . . .	51

3.6.4	Description of the Compact Bionic Handling Assistant	51
3.6.5	Simulation of the CBHA . . . . .	53
3.7	Kinematic models . . . . .	55
3.7.1	Forward kinematic models . . . . .	56
3.7.2	Inverse kinematic model . . . . .	62
3.8	Conclusion of the chapter . . . . .	63
<b>4</b>	<b>Closed loop control of soft, continuum manipulators</b>	<b>67</b>
4.1	Introduction . . . . .	67
4.2	Feed-forward control of continuum manipulators . . . . .	68
4.3	Closed-loop control of continuum manipulators . . . . .	71
4.3.1	Closed-loop control law design . . . . .	72
4.3.2	Robustness analysis . . . . .	76
4.4	Conclusions of the chapter . . . . .	80
<b>5</b>	<b>Conclusion and Perspectives</b>	<b>83</b>
5.1	Summary of conclusions . . . . .	83
5.2	The FeTCh manipulator . . . . .	85
5.3	Perspectives . . . . .	88
<b>A</b>	<b>Introduction to Continuum Mechanics and FEM</b>	<b>89</b>
A.1	Introduction . . . . .	89
A.1.1	Continuum Mechanics . . . . .	89
A.2	Finite Element Method for linear elastic bodies . . . . .	98
A.2.1	Discretization of the domain . . . . .	99
A.2.2	Element solution . . . . .	100
A.2.3	Assembly . . . . .	103
A.2.4	Displacement boundary conditions . . . . .	105
A.2.5	Solution method . . . . .	106
<b>B</b>	<b>Domain decomposition of continuum manipulators</b>	<b>109</b>
B.1	Domain decomposition of the CBHA . . . . .	109
	<b>Bibliography</b>	<b>111</b>

# Introduction

---

## Contents

---

<b>1.1</b>	<b>General Introduction</b>	<b>1</b>
<b>1.2</b>	<b>Framework and context</b>	<b>2</b>
<b>1.3</b>	<b>Motivation</b>	<b>2</b>
<b>1.4</b>	<b>Chapter summary</b>	<b>3</b>
<b>1.5</b>	<b>List of Publications</b>	<b>4</b>

---

## 1.1 General Introduction

50 years after the construction of the first continuum manipulator, researchers and engineers still consider the modeling and control of this type of robot an open problem. The development of a generic modeling approach for continuum manipulators that can capture the main behavior characteristics regardless of its design seems like a far fetched idea. From our point of view, this issue is linked to some key factors. First, the bio-inspiration behind continuum manipulators: nature offers a myriad of shapes and motion mechanisms from which we can gather inspiration for the design of soft, continuum manipulators. However, most of these principles require a great insight of the biological object of inspiration. As we close the gap between the more traditional rigid designs of manipulators currently populating the factories and the organic bodies of snakes and tentacles, the need to trespass the knowledge boundaries of robotics as a field arises.

On the other hand, the tools used to study this type of robots usually aim at the over-simplification of the problem; as for the writing of this manuscript, the most popular approach towards the modeling of continuum manipulators is based on conventional rigid robotic methodologies that consider only the structural geometry of the robot as an important feature in the modeling. In reality, the geometrical description of the manipulator is only one part of

the solution to the kinematic model. The other part being the continuum deformation.

In the study of deformations of continuum bodies, the field of continuum mechanics appears as a natural choice of analysis framework. In our opinion, a generic modeling methodology for continuum manipulators should be based on continuum mechanics since it allows for an accurate approximation of the kinematic solution based on a constitutive material law. If the material of the manipulator changes, it is only this constitutive law that changes in the model, but not its entire formulation.

## 1.2 Framework and context

This Ph.D. thesis was developed within the research groups, *Deformable Robotic Software* (Defrost)<sup>1</sup> from Inria Lille-Nord Europe and *Méthodes et Outils pour la Conception Intégrée de Systèmes* (MOCIS), of the *Centre de Recherche en Informatique, Signal et Automatique de Lille* (CRISAL) (UMR CNRS 9189)<sup>2</sup>. The implementation of the Ph.D. results has been realized as part of the framework of technological research with Festo-Didactic company from Esslingen, Germany<sup>3</sup>. This Ph.D. was funded by the *Mexican National Council of Science and Technology* (CONACYT)<sup>4</sup>.

## 1.3 Motivation

A generic modeling and control approach needs to have modularity. We need to be able to include, for example, the contact computation or the environment interaction with the robot as a module in our model to give it re-usability. It is simply not practical to think about a re-calibration of the model every time the conditions of operation change.

To this end, this Ph.D. work proposes a modeling and control methodology for soft, continuum manipulators based on computational mechanics. At the core of this methodology is the Finite Element Method (FEM), a technique for the approximation of differential equations with boundary conditions. FEM is used in this context to discretize the theoretical infinite number of degrees of

---

<sup>1</sup><https://team.inria.fr/defrost/>

<sup>2</sup><https://www.cristal.univ-lille.fr/>

<sup>3</sup><http://www.festo-didactic.com/int-en/>

<sup>4</sup><http://conacyt.gob.mx/>

freedom of continuum manipulators. The imposition of constraints in the form of Lagrange Multipliers allows for the model of sensors, actuators and end-effector. In this way, the forces applied to the manipulator by the actuators are related to the end-effector and sensors spaces. Forward and inverse kinematic models are derived from the FEM model of the manipulators. To account for non-modeled nonlinear behaviors of the robot, a closed-loop control strategy based on the implementation of the simulation as a state estimator is proposed.

## 1.4 Chapter summary

The manuscript is organized in 4 chapters as follows:

Chapter 2 introduces the definition of the kinematic model of continuum manipulators. To this end, the challenges of modelization are identified. The current state of the art in kinematic models and the methods used are given, along with the most recent efforts towards the formalization of a design methodology and control of continuum manipulators. The proposed methodology is then positioned in the context of the literature.

Chapter 3 formally introduces the methodology. First, a brief introduction to Continuum Mechanics is given and the fundamental steps of the Finite Element method are presented. Then, the method of modeling of continuum manipulators based on Computational Mechanics is explained. The concept of constraints, fundamental in the development and application of the method, is introduced.

Also, in this chapter we talk about the simulation framework used for the implementation of the method and the general steps required to obtain a robot simulation. Then, the method is applied to a continuum manipulator, namely, the Compact Bionic Handling Assistant (CBHA). The kinematic relationships between 3 different spaces are explained and the experimental validation for forward and inverse kinematics is shown.

The objective of Chapter 4 is to present a closed-loop control strategy for continuum manipulators based on the methodology given in the previous chapters. The feed-forward control strategy based on real-time simulation is showcased and, based on the experimental results, we explain the need for more accurate controllers. The closed-loop control based on PI control law and control allocation based on Quadratic Programming (QP) formulation is proposed. The implementation of the controller is validated with experiments

using the CBHA and the robustness analysis for the close-loop system is also performed.

Finally, Chapter 5 gives the conclusions for this work and discuss the perspectives of future work, in particular, the use of the methodology towards the design of soft, continuum manipulators based on simulation and possible improvements to the model of this type of robots.

## 1.5 List of Publications

The results obtained during the development of this work have made the topic of the following list of publications:

### Journal Publication

- Bieze T., Largilliere F., Merzouki R., Duriez C. (Under review). FEM-based forward and inverse kinematics of soft, continuum manipulators. *Soft Robotics Journal*. Submitted on June 2017.
- Coevoet E., Bieze T., Largilliere F., Zhang Z., Thieffry M., Carrez B., Marchal D., Goury O., Dequidt J., Duriez C. (Under review) Software toolkit for modeling, simulation and control of soft robots. *Advanced Robotics Journal*. Submitted on March 2017.

### International Conference

- Zhang Z., Bieze T., Dequidt J., Kruszewski A., Duriez C., Visual Servoing Control of Soft Robots based on Finite Element Model. Accepted for publication in 2017 IEEE/RSJ *International Conference on Intelligent Robots and Systems (IROS 2017)*
- Duriez, C., Coevoet, E., Largilliere, F., Bieze T., Zhang, Z., Sanz-Lopez, M., Dequidt, J. (2016, December). Framework for online simulation of soft robots with optimization-based inverse model. In *Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN)*, IEEE International Conference on (pp. 111-118). IEEE.
- Lakhal, O., Melingui, A., Bieze, T. M., Escande, C., Conrard, B., Merzouki, R. (2014, December). On the kinematic modeling of a class of continuum manipulators. In *Robotics and Biomimetics (ROBIO)*, 2014 IEEE International Conference on (pp. 368-373). IEEE.



- Bosman, J., Bieze, T. M., Lakhali, O., Sanz, M., Merzouki, R., Duriez, C. (2015, May). Domain decomposition approach for FEM quasistatic modeling and control of Continuum Robots with rigid vertebrae. In *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on (pp. 4373-4378). IEEE.

#### Book Chapter

- Duriez, C., Bieze, T. (2017). Soft Robot Modeling, Simulation and Control in Real-Time. In *Soft Robotics: Trends, Applications and Challenges* (pp. 103-109). Springer International Publishing.

#### Other contributions

- Runner up Prize at the Soft Robotics Toolkit Design Competition (2015). FeTCH Mark I Manipulator. Contributors: Bieze T., Largilliere F., Hage S., Sanz-Lopez M., Duriez C., Project page at <https://softroboticstoolkit.com/fetch>
- Bieze, T. (2015). On the Kinematic Modeling Methodology of Soft Manipulator Robots. Poster presentation. In *Robotics and Automation (ICRA)*, 2015 IEEE International Conference on. IEEE.



CHAPTER 2

# State of the Art

---

## Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>7</b>
<b>2.2</b>	<b>Continuum manipulator definition</b>	<b>7</b>
2.2.1	Soft continuum robot applications	8
2.2.2	Bio-inspiration	9
2.2.3	Classification	10
<b>2.3</b>	<b>State of the art in soft, continuum manipulators</b>	<b>13</b>
2.3.1	Design of continuum manipulators	13
2.3.2	Modeling of continuum robots	18
2.3.3	Dynamics and control of continuum robots	27
<b>2.4</b>	<b>Work contextualization and contributions</b>	<b>29</b>

---

## 2.1 Introduction

In this chapter, we will, first, give a formal definition of a soft, continuum manipulator in the context of this work. The State of the Art related to the modeling, design and control of this type of robots is provided in order to position our work in the framework of the current literature.

## 2.2 Continuum manipulator definition

To grasp the concept of continuum manipulator it is always helpful to visualize a conventional manipulator composed, in general, by a set of joints, rotational or spherical, serially connected by rigid links. In this class of manipulators, the links provide a kinematic relationship between the joints; in other words, by knowing the lengths of the links, one can always know the relative position of the joints with respect to a common reference frame [Siciliano 2016]. Now, let us place additional joints along the rigid links of our imaginary manipulator.

Soon enough, our manipulator will have more active<sup>1</sup> degrees of freedom than the ones required to position an object in space. In the literature, these type of manipulators are known as hyper-redundant [Hirose 2004]. If we keep repeating the process of adding joints until their number approaches infinity and the link lengths to zero, our robot will eventually approach to what is known as a continuum manipulator.

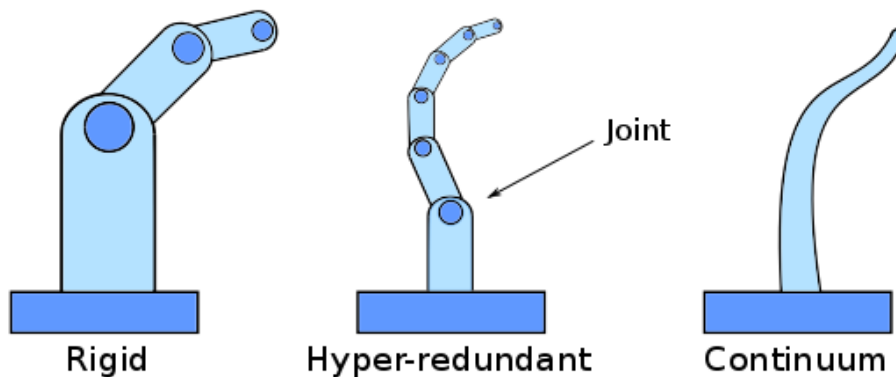


Figure 2.1: Left) A classic industrial manipulator, (Center) A hyper-redundant robot, the meeting ground between a rigid and a continuum manipulator, (Right) A continuum manipulator

The previous definition may imply that all the degrees of freedom in a continuum manipulator are controllable, which is in general, not the case. Often bio-inspired by the morphology and functionality of biological limbs and appendices like octopus tentacles [Zheng 2014], elephant trunks [Neppalli 2007] [McMahan 2005] [Zhao 2010], plant tendrils [Mehling 2006] [Yamada 2014] and other mammal tongues, continuum manipulators are composed by an elongated structure with no identifiable joints, which is continuously bending via elastic deformation [Robinson 1999].

### 2.2.1 Soft continuum robot applications

The characteristics of soft continuum manipulators, like their natural compliance, high power to weight ratio, and reduced dimensions compared to their

<sup>1</sup>In robotics, the degrees of freedom can be of two types: passive degrees of freedom are the ones that cannot be directly controlled and conform passively to the movements of the robot, while active degrees of freedom are the ones that are controlled to provide the required forces to move the manipulator.

rigid counterparts, make them particularly suitable for applications in which the contact with humans is unavoidable or even desired. Applications such as skeletal trauma treatment [Wilkening 2017] [Alambeigi 2017], endoscopy [Conrad 2013] [Cianchetti 2013] [Fraś 2015] and minimally invasive surgery [Qu 2016] [Orekhov 2016] [Mahoney 2016] have proved the great potential of application of continuum robots in the medical field, as extensively reviewed in [Burgner-Kahrs 2015].

As suggested by the very first prototypes of continuum manipulators [Anderson 1967]. The slender shape and high dexterity of continuum manipulators can be exploited in tasks such as minimally invasive inspection [Mehling 2006] [Tonapi 2014] and search and rescue [Bajo 2010] [Li 2017].

Continuum robots have been studied with the goal of exploiting their locomotive capabilities [Godage 2012] [Kang 2012] [Arienti 2013], although the review presented in the following of this manuscript is concerned mainly on manipulation.

### 2.2.2 Bio-inspiration

If one sees for the first time a continuum manipulator, without any previous knowledge on the concept behind it, one can immediately identify the remarkable morphological similarities that this type of devices have with some soft-bodied animals, particularly with the muscular hydrostats. Muscular hydrostats, commonly found in elephant trunks, mammal tongues and octopus tentacles are soft muscular structures that can bend, extend and twist and provide the force required for movement and skeletal support to animals (or limbs) that lack a rigid skeleton, see Fig. 2.2.

Muscular hydrostats are typically composed by a fluid-filled cavity surrounded by a muscular wall reinforced with connective tissue fibres. The arrangement of the muscle fibres in a hydrostatic muscle include both circular and longitudinal muscle fibres. These two muscle fibres can antagonize one another to produce a variety of shape changes including elongation and bending [Kier 1992]. The fluid inside the cavity of a hydrostatic limb is mainly a liquid which resists to volume change, thus, to create an elongation of the limb, the circular muscles contract to decrease the diameter while increasing the length to allow for a constant volume inside the cavity. The study of the biomechanics of hydrostatic structures have shown also additional fibres with more intricate configuration patterns that allow for more complex shape changes like twisting, present in octopus tentacles and mammals and reptile

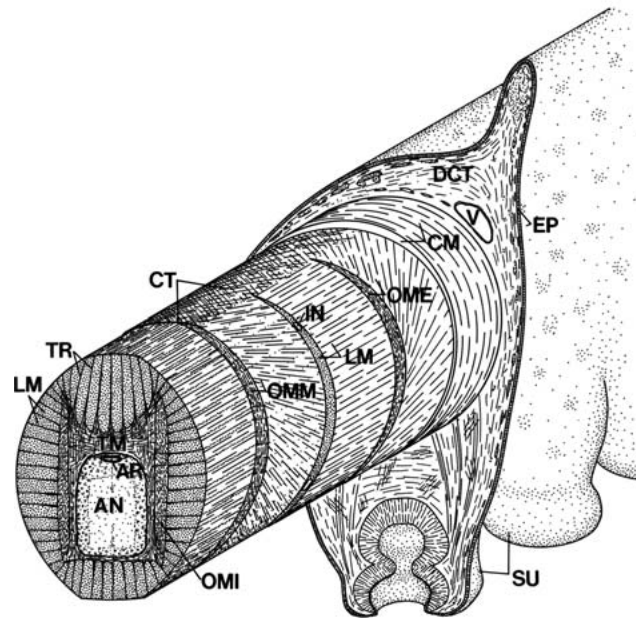


Figure 2.2: Diagram of the hydrostatic muscle in the arm of *Octopus* from [Trivedi 2008b]. AN, axial nerve cord; AR, artery; CM, circumferential muscle layer; CT, connective tissue; DCT, dermal connective tissue; EP, epidermis; IN, intramuscular nerve; LM, longitudinal muscle fibres; OME, external oblique muscle layer; OMI, internal oblique muscle layer; OMM, median oblique muscle layer; SU, sucker; TM, transverse muscle fibres; TR, trabeculae; and V, vein.

tongues [Kier 1985]. The complete replication of hydrostats is very complex, but the study of their underlying function principles have given roboticists an interesting insight and a solid starting point in the design of soft, continuum manipulators.

### 2.2.3 Classification

Continuum manipulators can be broadly classified with respect to the type of backbone they possess. *Single backbone* manipulators (see Fig. 2.3 (left)) have a central elongated structure that supports the passage of actuation/transmission elements along the body of the manipulator [Burgner-Kahrs 2015]. Many single-backbone designs utilize tendons routed along the structure which are spaced by *discs* attached to the backbone as a way of transmission. The termination points of the tendons in this type of design dictate the lengths of a bending section. *Multibackbone* (Fig. 2.3 (right))

continuum manipulators are typically composed by a parallel arrangement of elastic elements which are constrained with respect to each other in some way.

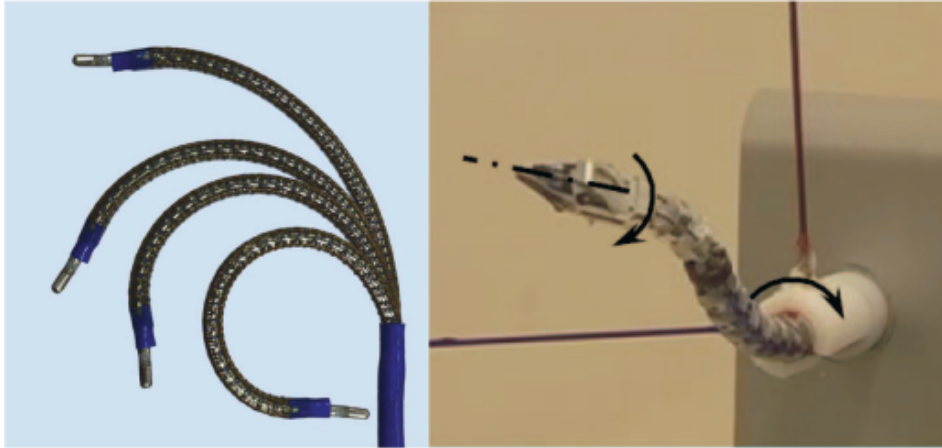


Figure 2.3: Single and Multibackbone continuum robots:(left) A single backbone tendon-driven steerable cardiac catheter from [Camarillo 2008].(Right) The DDU multibackbone manipulator designed by Simaan et al. [Simaan 2009].

Another classification of continuum manipulators is based on the actuation system implemented to apply forces and torques to the backbone. When the actuators are embedded in the structure and apply forces directly to the backbone, the actuation scheme is called intrinsic. Most continuum robot designs with multi-backbones use pneumatic cavities to conform the structure, essentially making a backbone composed of actuators that can be reshaped by applying pressure to the cavities. Extrinsic actuation systems place the actuators outside of the structure. These apply torques and forces at localized points via a mechanical link to the backbone, which is made by an incompressible elastic rod. A hybrid actuation scheme replaces the central elastic rod in an extrinsic design by an actively controlled actuator, which is in many designs, a pneumatic cavity [Chirikjian 2015] [Ataollahi 2017].

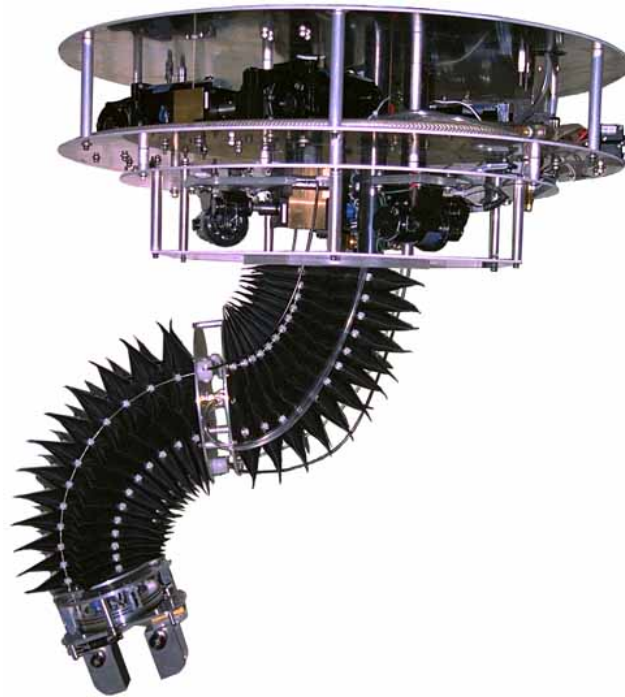


Figure 2.4: The KSI hybrid actuated manipulator [Immega 1995]

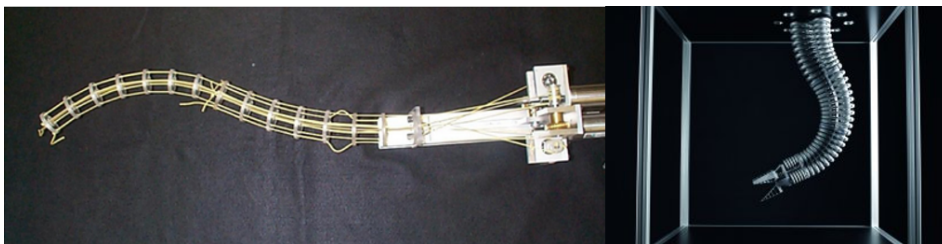


Figure 2.5: (Left) The Clemson tentacle, an extrinsic actuated manipulator [Gravagne 2001]. (Right) The BHA, an intrinsic actuated manipulator [Rolf 2012].



## 2.3 State of the art in soft, continuum manipulators

This section describes the research developed in the last 20 years, towards the design, modeling and control of soft, continuum manipulators. The review is divided into each of the subtopics mentioned in order to provide a clear overview of the field.

### 2.3.1 Design of continuum manipulators

As stated in [Walker 2013a], the first prototype of a continuum robot reported in the literature was the Tensor Arm [Anderson 1967], designed by Anderson and Horn in 1967. Conceived to be used under water, the prototype was able to achieve a wide range of shapes; however, the relationship between the shapes and inputs was highly complex and challenging for the computational resources of that time. Based on extrinsic actuation, the robot used nylon filaments routed along the structure through spacer discs that apply torques directly to the backbone to produce bending, see Fig. 2.6. As is usual in tendon-based designs, the termination points of the cables define the bending sections. This prototype has inspired since then, a significant number of designs based on the same principle.

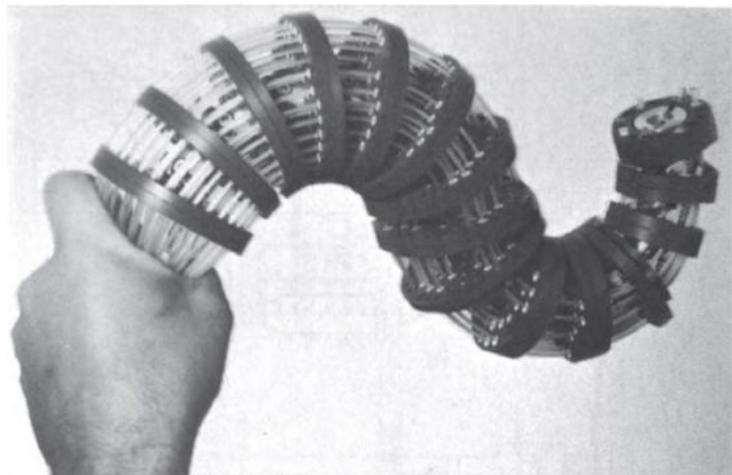


Figure 2.6: The "Tensor Arm" by Anderson and Horn [Anderson 1967].

Walker and Hanna presented the Elephant's Trunk Robot [Walker 1999], a continuum manipulator with extrinsic actuation composed of 4 main bending sections. The robot uses a pulley system outside the structure to pull cables

and bend the sections. Spacer disks are used to give the robot its characteristic shape but they also provide a mechanical coupling; out of the 32 DoF only 8 are actively controllable, leaving the rest to be coupled by passive springs positioned between adjacent segments of the manipulator at a 90 degrees angle with respect to the tendons. Thanks to this work the identification of critical hardware necessities in the field was possible, in particular the need for elastic structures. Later work in design of continuum robots replaced the spring system by an incompressible elastic rod [Gravagne 2003]. This choice of backbone has been, since then, the most used in tendon-based designs of continuum manipulators [Hemami 1984] [Simaan 2004] [Bardou 2010] [Zhao 2010]. Another choice of backbone is the use of a central spring [Mehling 2006] that provides the desired compliance to the manipulator but makes the operation of this type of manipulators a difficult task, since the forces applied by the tendons tend to be absorbed by the spring. However, spring-backbone manipulators have been implemented as actuated endoscopes with great success [Yoon 2011].

At the same time, research in the area of artificial muscle design facilitated the conception of locally actuated designs, such as the Oct-Arm [Walker 2005] [McMahan 2006], the "European Octopus" [Calisti 2012] [Guglielmino 2010] [Laschi 2009], and the Bionic Handling Assistant [Rolf 2012], which is inspired by the trunk of an elephant, with possible appealing advantages like full-body manipulation. Intrinsic design manipulators proved to be very dexterous by reproducing the way longitudinal muscles work in an animal trunk or tentacles [McMahan 2006]. Originally developed to provide actuation to a prosthesis device for severely paralyzed hand patients, the McKibben artificial muscle [Nickel 1963] has been used in continuum robot designs with great success [McMahan 2006] [Pritts 2004]. This type of pneumatic muscle is composed by a soft cavity constrained in its deformation by a braided sleeve. The muscle is considered contractor if the cavity expands radially when the pressure is increased in the cavity, or extensor if the cavity expands longitudinally. This operation is analogous to the longitudinal muscles. Experiments conducted on the McKibben muscle [Klute 1999] show that it provides a first order approximation to a biological muscle when contracting.

Another venue for researchers to design extrinsic actuation is the implementation of concentric tubes as the structure of the manipulator [Sears 2006] [Gilbert 2016]. The body of the robot is composed by a telescopic array of concentric tubes that are free to rotate and translate with respect to each other. To provide bending to the robot, the tubes are often pre-curved, so that when a distal tube is translated outside of the preceding one, it will nat-

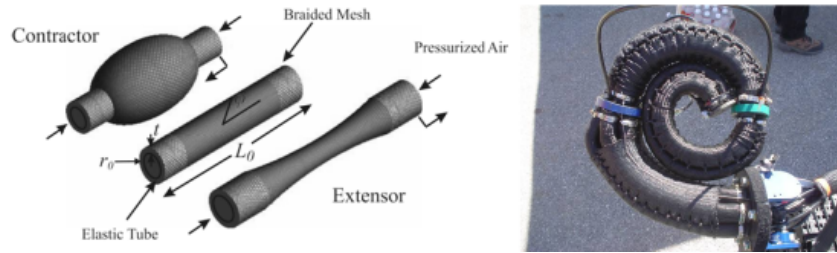


Figure 2.7: (Left) Extensor and contractor pneumatic muscles, (Right) An example of a locally actuated design, the Oct-Arm manipulator composed by pneumatic muscles, from [McMahan 2006]

usually bend [Furusho 2006]. Pre-curved concentric tubes designs have found a niche of applications in the medical field [Wu 2017], where their slim and clean design is well suited to act as actuated endoscopes in minimally invasive surgery [Furusho 2006] [Su 2012].



Figure 2.8: A concentric tube robot or *Active Cannula* composed by 3 pre-curved tubes, from [Wu 2017]

Given the advantages of intrinsic actuation in terms of dexterity, particularly the ability to extend the backbone longitudinally and in turn, improve its

ability to bend [Walker 2006], robot designers started to experiment with hybrid actuation schemes that incorporate both pneumatic cavities and tendons. An example of this type of design is the Air-Octor [Jones 2004a] [Jones 2004b] depicted in Fig. 2.9, a manipulator composed by a soft pneumatic hose that acts as the backbone, which is actuated by 3 cables to achieve bending. While the prototype proved to be capable of full-body manipulation, it suffered from sagging and kinking of the soft backbone cavity [McMahan 2005], a problem that is common to this type of designs. Another example of a realized hybrid actuation scheme is the KSI manipulator [Immega 1994], developed by Kinetic Sciences Inc. in 1994 [Immega 1995].

### 2.3.1.1 Backbone variable stiffness

In order to exploit to the maximum the capabilities of continuum robots, the ability to variate the stiffness of its structure is needed, particularly in medical applications where excessive contact forces at the tip of the manipulator can cause severe trauma to the surrounding tissue. In the literature, one can find different approaches to the variable stiffness of the backbone. In [Mahvash 2010], a stiffness control approach is presented. The method is based on the derivation of the kinematic model of the manipulator as the product of two transformations: the first transformation describes the non contact kinematics of the manipulator and it is specific to the robot itself, while the second transformation calculates the tip deflection due to the applied forces and is derived using the Cosserat rod formulation [Mahvash 2011]. To implement a desired tip stiffness, the two transformations are used to solve for the actuator positions that deform the manipulator so as to generate the required tip force at the measured tip position. This method is applied to a concentric tube design intended for minimally invasive surgery. In [Goldman 2011] a method for the stiffness control of intrinsic actuated manipulators is presented. The proposed algorithm relies on the measurement of actuation forces, in conjunction with the stiffness of the backbone, to move in a direction to minimize the environment interaction.

The combination of pneumatic backbones and tendons provide the means to increase the apparent stiffness of the structure by antagonizing the set of redundant actuators that compose the robot [Walker 2013b]. In [Maghooa 2015] the pneumatic backbone and tendons of a hybrid manipulator are controlled simultaneously to achieve a certain position of the tip of the arm while actively changing the stiffness of the robot applying the principle of antagonistic actuation. This functionality is also bio-inspired [Shiva 2016], being present in a significant number of animal arms, trunks and even the human tongue.

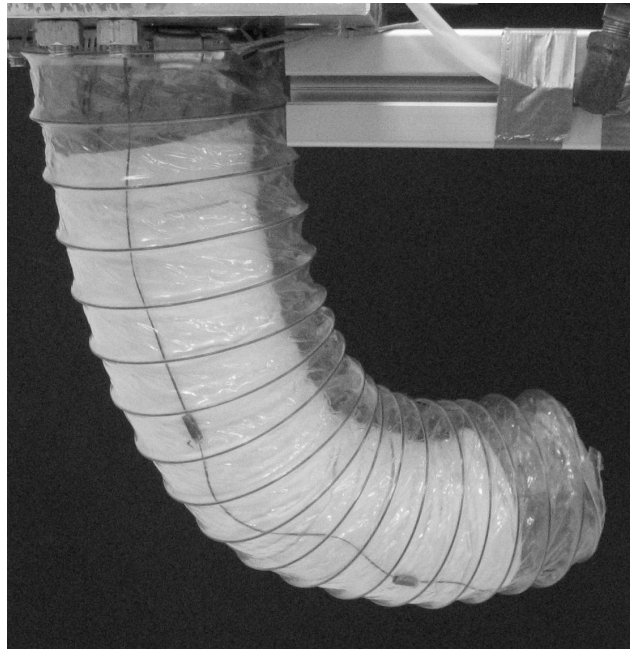


Figure 2.9: The Air-Octor hybrid manipulator composed by a pneumatic backbone which is bent by tendons, as presented in [Jones 2004b]

Some continuum manipulator prototypes utilize local rigidification mechanisms to simplify the stiffness control; such is the case of the STIFF-FLOP manipulator [Cianchetti 2013] [Fraś 2015], that utilizes granular jamming [Steltz 2010], in conjunction with pneumatic cavities to achieve a certain degree of variable stiffness of the structure. Each section of the robot is composed by an elastomeric cylinder that hosts three pneumatic cavities to provide bending and extension as well as a central chamber that holds the granular media. The section is surrounded by a braided sleeve to prevent the radial deformation of the elastomeric cavities. The granular media utilized is coffee powder, which *jams* when the central cylinder is under negative pressure causing drastic stiffness changes of the structure [Jiang 2012]. The STIFF-FLOP is one of the most advanced continuum manipulators prototypes currently under development, having force [Noh 2014] and tactile [Sareh 2014] capabilities provided by sensors embedded in the structure of the robot.

More advanced actuation systems have been studied with the intention of stiffness control. In [Sadeghi 2012], Sadeghi et al. demonstrated a method to control the stiffness of a worm-like soft robot utilizing electro-rheological materials that can change phase, from fluid to solid and vice versa, under the presence of magnetic fields.

### 2.3.1.2 Design methodologies

The key challenge for creating soft machines that achieve their full potential is the development of controllable soft bodies using materials that integrate sensors, actuators and computation, and that together enable the body to deliver the desired behaviour [Rus 2015]. The conception of a design methodology for continuum robots is not a trivial task due to the wide variety of possible backbone-actuation schemes, the applications to which the design is aimed at and more importantly, due to the interdisciplinary nature of the design tasks; very often, electronics, mechanics and computer science come into play during the integration of the system. Nevertheless, some design guidelines can be found in the literature. For example, a pragmatic approach to the design of continuum arms inspired by the octopus tentacle is presented in [Guglielmino 2013]. In [Bedell 2011] an algorithm for the optimization of concentric tube robot design is proposed; given the description of the task space and the number of sections in the robot, the algorithm solves for a design with the desired workspace.

The fabrication of artificial pneumatic muscles is also a daunting enterprise that requires, in most cases, multiple design iterations to achieve a desired behavior. A direct<sup>2</sup> finite element simulation has been used in [Connolly 2015] for the programming of desired deformations of pneumatic muscles with braided sleeves.

The work of Hauser et al. in [Hauser 2011] presents the concept of morphological computation. This concept aims at the relegation of difficult control tasks in soft bodied robots to the body itself, making use of inherent characteristics of the robot, like the natural compliance of its structure [Paul 2006]. While it is difficult to directly design *smart bodied* robots, this work provides useful design guidelines compatible with any kind of bio-inspired robot [Pfeifer 2009].

### 2.3.2 Modeling of continuum robots

In order to control the movement of continuum robots, kinematic models, which relate the configuration (pose) of the backbone and task space (end-effector position) and actuator variables must be established. These models are fundamental for the development of control strategies and path planning

---

<sup>2</sup>In this work, we refer to the direct finite element when the unknown nodal displacement is computed given the known external forces



for continuum manipulator robots. Equation (2.1) depicts the general form of a kinematic model for conventional manipulators

$$\mathbf{x} = f(q) \quad (2.1)$$

where  $\mathbf{x}$  is the position of the end-effector inside the task space of the robot which is, more often than not, expressed in cartesian coordinates,  $q$  is the set of configuration variables related to the joint variables (which are, in general, observable and directly controlled) and  $f$  is the unknown function that relates task space variables to the configuration of the robot. For soft, continuum robots however,  $f$  depends on the configuration variables and also on the mechanics of the deformable material. We can then rewrite Eq. 2.1 for continuum manipulators as

$$\mathbf{x} = f(q, \vartheta) \quad (2.2)$$

where  $\vartheta$  represents the mechanics of the soft material. The modeling task deals with finding this relationship and the assumptions used in order to accomplish this goal dictates the type or *flavor* of the modeling approach.

In general, kinematic modeling approaches of continuum manipulators can be broadly classified as quantitative models, which deal with the description of the robots in a mathematical fashion making use of geometry and elasticity theories, and qualitative models, which are numerical abstractions of the more complex model and often make use of experimental data to find *the closest* solution to the kinematic problem.

**Qualitative models** Due to the complexity involved in the modeling of continuum robots, qualitative approaches, which aim at *learning* the kinematic problem by the use of learning algorithms, have a significant level of popularity due to their ability to by-pass the modeling task. These approaches are based on previous knowledge of input-output data derived from experimentation and can provide accurate fast approximations to the kinematic solutions. Rolf and Stein proposed in [Rolf 2014] a control scheme for the Bionic Handling Assistant based on goal babbling [Rolf 2010]. The control scheme is capable of dealing with highly non-linear behavior like hyperelasticity, plasticity and non-stationarities, which are very complex to address analytically.

Control schemes based on Neural Networks have been also investigated. Giorelli et al. [Giorelli 2013] used a feedforward neural network to learn the in-

verse kinematics of a tendon driven manipulator. In this approach, a geometrical model of the manipulator is used for data generation and in [Giorelli 2015] a controller based on the static model Jacobian is used to control a non-constant curvature manipulator. Another feedforward neural network controller was proposed by Braganza et al. [Braganza 2006] [Braganza 2007]. In this work, no model is required; instead, a low level controller uses the neural network component to compensate for the dynamic uncertainties of the manipulator. More recently, the work of Melingui et al. [Melingui 2014] employed a neural network to solve the inverse kinematics based on the measurements of the end-effector position. In [Melingui 2017] an adaptive algorithm is implemented to improve the performance of the controller by allowing rapid position convergence of the end-effector. While qualitative controllers have shown great success in real world scenarios, the fact that the learning base changes when the operation conditions of the robot change make data-driven models limited in their application. This is the main reason as to why quantitative approaches still dominate the literature.

**Quantitative models** Current quantitative models in the literature can be classified into 2 different main approaches: those that describe the behavior of the continuum backbone curve with respect to geometry parameters, which we will call Geometric Models, and those that use classical elasticity theories, such as strings or rods, to describe the manipulator with respect to the way its structure behaves under external loading, which we will call Elasticity Models. In this section, a review of these modeling approaches is given in order to highlight the fundamental differences between these models and the approach presented in this work and to position clearly the contribution of this work in the current literature.

### 2.3.2.1 Geometric models

We refer to the first type of quantitative models as Geometric models. These approaches compute the kinematic relationships making use of the geometry of the manipulators, in particular, the curvature that the robot exhibit under the actuation forces; however, the forces themselves are not accounted for in the model.

The first kinematic analysis of continuum robot backbones was presented by Chirikjian [Chirikjian 1992] [Chirikjian 1994b]. In this approach, also called *Non-Constant curvature approach*, the Cartesian position of backbone curve points  $\mathbf{x}(s, t)$  is given by the integral representation:



$$\mathbf{x}(s, t) = \int_0^s l(\sigma, t) \mathbf{u}(\sigma, t) d\sigma \quad (2.3)$$

where  $\mathbf{u}(s, t) = [0 \ 1 \ 0]^T$  is the unit vector tangent to the curve at  $s$  and  $l(s, t)$  is a scaling factor that controls the length of the curve tangent according to local backbone extension or contraction. In [Chirikjian 1992], the following backbone curve representation is given:

$$\mathbf{x}(s, t) = [x_1(s, t) \ x_2(s, t) \ x_3(s, t)]^T = \begin{pmatrix} \int_0^s l(\sigma, t) \sin(K(\sigma, t)) \cos(T(\sigma, t)) d\sigma \\ \int_0^s l(\sigma, t) \cos(K(\sigma, t)) \cos(T(\sigma, t)) d\sigma \\ \int_0^s l(\sigma, t) \sin(T(\sigma, t)) d\sigma \end{pmatrix} \quad (2.4)$$

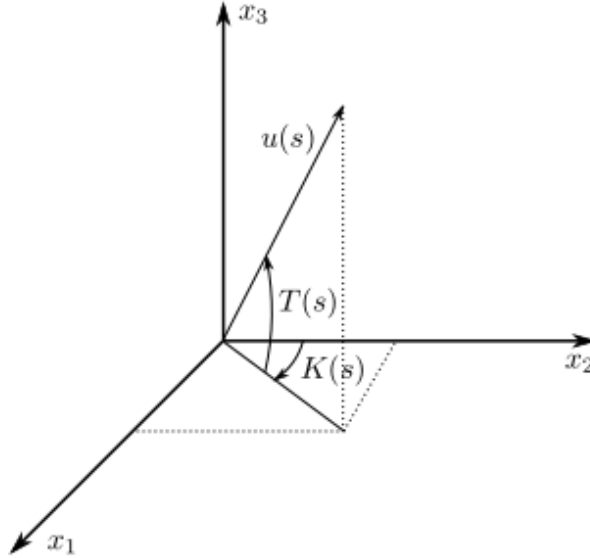


Figure 2.10: Backbone curve parametrization

with  $K$  and  $T$  being the curve parameters as illustrated in Fig. 2.10. The backbone reference frame  $\Phi(s, t)$  attached to the curve at  $s$  and which origin coincides with  $\mathbf{x}(s, t)$  has its orientation described by

$$\Phi(s, t) = \begin{pmatrix} \cos(K(s, t)) & \sin(K(s, t)) \cos(T(s, t)) & -\sin(K(s, t)) \sin(T(s, t)) \\ -\sin(K(s, t)) & \cos(K(s, t)) \cos(T(s, t)) & -\cos(K(s, t)) \sin(T(s, t)) \\ 0 & \sin(T(s, t)) & \cos(T(s, t)) \end{pmatrix} \quad (2.5)$$

The inverse kinematics of the continuous curve, that is, the problem of finding the backbone position and orientation given a specific task constraint,

is then solved by finding the values of the independent functions  $l(s, t)$ ,  $T(s, t)$  and  $K(s, t)$ . A modal approach for the selection of these values is developed in [Chirikjian 1992]. The disadvantage of this approach is that the set of backbone shapes available in the model is restricted by the combination of modal functions, and further analysis of the hardware is required to *tune* the selection of the basis functions to be able to model arbitrary backbone shapes. In [Mochiyama 2003], Mochiyama and Suzuki used the same approach to derive the kinematics and dynamics of a cable-like flexible manipulator.

In [Hannan 2000], Hannan and Walker presented what would later become the most used approach towards the kinematic modeling of continuum manipulators. This work makes the assumption that, after actuation, the shape of the backbone of the manipulator can be assimilated as a piecewise constant curvature curve and its evolution, from base to end, is then described by a set of 3 discrete transformations for the planar curve. The first transformation is a rotation to *point* the tangent direction at the curve beginning to the end of the curve, followed by a translation along the new direction, from the curve starting point to the curve end point, and finally, a second rotation of same magnitude as the first to realign with the tangent at the curve end point. See Fig. 2.11

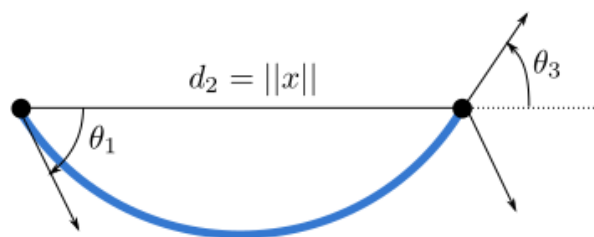


Figure 2.11: The geometry of a planar constant curvature curve segment

Thus, a virtual rigid link manipulator with two rotational and one prismatic joint can be used to derive the kinematic model of the planar constant curvature backbone. The corresponding kinematic model is found using standard Denavit-Hartenberg notation [Hartenberg 1955] for the virtual rigid manipulator. The Denavit-Hartenberg parameters are shown in the following table:

Link	$\theta$	d	a	$\alpha$
1	*	0	0	$-90^\circ$
2	0	*	0	$90^\circ$
3	*	0	0	$0^\circ$

where  $\theta$  and  $d$  are the joint parameters for rotational and prismatic joints, respectively,  $\alpha$  is the link twist and  $a$  is the length of the common normal vector between consecutive joints. Using the previous parameters, the homogeneous transformation matrix, which relates the Cartesian position of the end of the curve with respect to the base is formed as:

$$\mathbf{A}_0^3 = \begin{pmatrix} \cos(\theta_1 + \theta_3) & -\sin(\theta_1 + \theta_3) & 0 & -d_2 \sin(\theta_1) \\ \sin(\theta_1 + \theta_3) & \cos(\theta_1 + \theta_3) & 0 & d_2 \cos(\theta_1) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

Further manipulation of the derived homogeneous transformation matrix is required to relate the joint variables of the virtual robot to the configuration space variables of the continuous curve. Noting that

$$\theta_1 = \theta_3 = \frac{ks}{2} \quad (2.7)$$

and also

$$d_2 = \|\mathbf{x}(s)\| = \frac{2 \sin \theta}{k} \quad (2.8)$$

with  $k$  being the inverse of the radius of the arc and  $s$  being the arc length, the previous transformation matrix turns into

$$\mathbf{A}_0^3 = \begin{pmatrix} \cos(sk) & -\sin(sk) & 0 & \frac{1}{k}(\cos(sk) - 1) \\ \sin(sk) & \cos(sk) & 0 & \frac{1}{k} \sin(sk) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

which relates the continuum curve arc length and curvature to the task space. It is important to note that the arc length  $s$  can be chosen arbitrarily to model the transformation from the curve shape to any point along the backbone curve.

Spatial continuum kinematics can also be modelled by adding an additional pair of rotational joints to the virtual manipulator that rotates the curvature plane as shown in Fig. 2.12

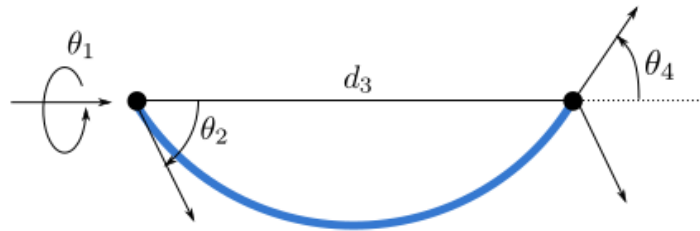


Figure 2.12: The geometry of a spatial constant curvature curve segment

The resulting Denavit-Hartenberg parameters are shown in the following table:

Link	$\theta$	d	a	$\alpha$
1	*	0	0	$90^\circ$
2	*	0	0	$-90^\circ$
3	0	*	0	$90^\circ$
4	*	0	0	$-90^\circ$

As reviewed in [Walker 2013a], the constant curvature approach produces the same kinematic relationships as the non-constant curvature approach. This approach has been used subsequently. Mahl et al. implemented a constant curvature continuum kinematics as *fast approximate* model for the Bionic Handling Assistant [Mahl 2014] and Escande et al. used the same approach to model the smaller version of the robot, the Compact Bionic Handling Assistant [Escande 2012] [Escande 2015].

Continuum manipulators composed by several serially connected sections can be modelled using the same approach by concatenating the transformation matrices of each section, as presented in [Jones 2006] and [Bardou 2010].

As is common when using geometric approaches, further modeling is required to find the relationships between the curve parameters and the actuator variables (displacement of tendons or pressure inside pneumatic muscles are the most common). This transformation is specific to each manipulator hardware and therefore, cannot be generalized. Nevertheless, geometric approaches provide a fast and accurate approximation to the kinematics of continuum manipulators in scenarios where gravity can be countered by a clever orientation of the manipulator or when the weight of the robot is small

compared to its structural stiffness. More important, they provide close form kinematic equations which can be easily implemented in control algorithms.

### 2.3.2.2 Elasticity models

The second class of quantitative kinematic models are, what we will call, Elasticity based models. These models substitute the backbone of the robot by continuum mechanic objects, like rods and strings, to better capture the elastic behavior of the backbone. In contrast with geometric models, mechanic-based models provide a description of the deformation of the backbone based on the constitutive law of the material from which the backbone is made, by establishing static equilibrium between external loading, such as gravity, and internal forces.

The Cosserat rod theory [Antman 1973] has become a great resource in the research of kinematics for continuum manipulators. In Cosserat rod theory, a homogeneous transformation matrix  $g(s)$  is used to describe the rod with  $s \in [0 \ l]$  being the reference parameter. The evolution of  $g(s)$  along  $s$  is defined by

$$\dot{\mathbf{R}}(s) = R(s)\hat{u}(s), \quad \dot{\mathbf{p}}(s) = R(s)v(s) \quad (2.10)$$

where the dot denotes derivative with respect to  $s$ ,  $\mathbf{R}$  and  $\mathbf{p}$  are the rotation matrix and position vector of  $g$  at  $s$  and  $\mathbf{R}(s)$  and  $\mathbf{u}(s)$  are the kinematic variables that represent the linear and angular rates of change of  $g(s)$ . The  $\hat{\cdot}$  operator is a linear mapping that converts  $\mathfrak{R}$  to the correspondent skew symmetric matrix as

$$\hat{u} = \begin{pmatrix} 0 & -u_z & u_y \\ u_z & 0 & -u_x \\ -u_y & u_x & 0 \end{pmatrix} \quad (2.11)$$

Given an undeformed reference configuration of the rod  $g^*(s)$ , the reference kinematic variables  $\mathbf{v}^*$  and  $\mathbf{u}^*$  can be obtained by

$$[\mathbf{v}^* \quad \mathbf{u}^*]^T = (g^{(*-1)}(s)\dot{g}^*(s))\mathcal{V} \quad (2.12)$$

where the operator  $\mathcal{V}$  represents the inverse operation of  $\hat{\cdot}$ . The internal force and moment vectors (in global coordinates) are denoted by  $\mathbf{n}$  and  $\mathbf{m}$ , the applied force distribution per unit of  $s$  is  $\mathbf{f}$ , and the applied moment distribution per unit of  $s$  is  $\mathbf{l}$ . The classical form of equilibrium differential equations for a Cosserat rod are

$$\dot{\mathbf{n}}(s) + \mathbf{f}(s) = 0 \quad (2.13)$$

$$\dot{\mathbf{m}}(s) + \dot{\mathbf{p}}(s) \times \mathbf{n}(s) + \mathbf{l}(s) = 0 \quad (2.14)$$

Using the constitutive law of the rod to map the kinematic variables to the internal forces and moments we have

$$\mathbf{n}(s) = \mathbf{R}(s)K_{se}(s)(\mathbf{v}(s) - \dot{\mathbf{v}}^*(s)) \quad (2.15)$$

$$\mathbf{m}(s) = \mathbf{R}(s)K_{bt}(s)(\mathbf{u}(s) - \dot{\mathbf{u}}^*(s)) \quad (2.16)$$

where  $K_{se}$  is the stiffness matrix for shear and extension and  $K_{bt}$  is the stiffness matrix for bending and torsion. Given the assumption that the stiffness matrices are constant with respect to  $s$ , a full set of explicit model equations that provide the values of  $\dot{\mathbf{p}}(s)$ ,  $\mathbf{R}(s)$ ,  $\dot{\mathbf{v}}(s)$  and  $\dot{\mathbf{u}}(s)$  can be derived. Usually, a linear constitutive law is used, but the approach can work under nonlinear constitutive laws as well.

The work of Davis and Hirschorn in 1994 [Davis 1994] on the modeling of flexible robot links with tendon control introduced this method in the context of robotics, although it was Trivedi et al. in 2008 [Trivedi 2008a] who conducted the first work directly in the field of soft manipulators using this approach, as reviewed in [Burgner-Kahrs 2015].

One of the advantages of mechanic-based modeling methods is the great variety of elements from which one can conform the model. As an example, one of the special cases of Cosserat rod, the Kirchhoff rod, has been used to derive models for concentric tube robots [Rucker 2010b]. The Kirchhoff theory assumes inextensibility and neglects transverse shear strain, which are generally regarded as good assumptions for long thin rods like the tubes that make up active cannulas [Rucker 2010a]. Wenlong et al. used Timoshenko beam theory, which accounts for shear deformation and beam twisting, to map a driven load to the pose of a continuum manipulator [Wenlong 2013]. Euler-Bernoulli beam theory, a special case of Timoshenko beam theory that only considers lateral external loading, has been used to simplify the computation of the mechanics of concentric tubes as well [Webster 2006].

While elasticity models can be fast to compute (up to a certain number of elements), they can be quite complex to parametrize and implement, also, as

explained before, since close form solutions for continuum mechanic objects exist only for a very limited number of cases, numerical approximations of the solutions of the partial differential equations that arise in their development must be performed.

### 2.3.3 Dynamics and control of continuum robots

The use of geometric models to describe the pose of a continuum manipulator is hardly sufficient due to the fact that internal and external loading in the manipulators play an important role in the configuration of the robot. Even in free space, gravity can cause significant deflections in the curvature of the robot even when no payload is being carried by the manipulator. Dynamic models, which aim at the description between the backbone shape and the forces that cause the deformation of it, have been the objective of intense research in the field since the very beginning [Chirikjian 1993] [Chirikjian 1994a].

As in the case with the modeling of kinematics, researchers have tried to apply methodologies commonly used for rigid manipulators to describe the dynamics of continuum ones. In [Mochiyama 2002], Mochiyama and Suzuki used the Lagrangian approach to describe the dynamics of a string-like manipulator. In this development, a discretization stage in the modeling is performed in which the backbone is assumed to be formed by circular *slices* located at  $\sigma$  along the backbone. Each slice has a mass. After calculating the kinetic and potential energy for each slice, the total energies are found by integration along the backbone and substituted in the Lagrangian  $L = K - P$ . Once the Lagrangian is calculated, the dynamic model is found as

$$\frac{\partial}{\partial t} \left( \frac{\partial L}{\partial \dot{\theta}_i(\sigma, t)} \right) - \frac{\partial L}{\partial \theta_i(\sigma, t)} = \tau_i(\sigma, t) \quad (2.17)$$

where  $(\theta_i, \tau_i)$  are the configuration variables and the forces that change them, respectively. In the initial study [Mochiyama 2002], non-extensible backbones were considered. Later work by Tatlicioglu [Tatlicioglu 2007] considered extensible backbones for a planar manipulator; however, the complexity of the closed form models generated renders the application to non-planar backbones too computationally expensive to be implemented in the control of manipulators.

More computationally efficient formulations of the dynamics of continuum backbones have been developed. In [Kang 2011], the Newton-Euler formula-

tion is applied to derive the dynamics of a manipulator for underwater applications. In [Giri 2011], Giri and Walker presented an approximation to the dynamics of a continuum arm using lumped parameters (mass, spring, and structural damping) and the Lagrangian formulation. This approach was implemented to compute the dynamics of an octopus inspired arm in [Zheng 2012] and in [Falkenhahn 2014] to describe the dynamics of the Bionic Handling Assistant manipulator.

The elasticity models discussed in section 2.3.2.2 provide a formulation that explicitly takes the internal and external forces into account. However, for more complex applications in which the external loading of the manipulator comes from the contact with its environment, or the velocity of the manipulator induces inertial effects in its behavior, these models are not sufficient. In [Yu 2015] inverse dynamics and sliding mode control schemes for a continuum manipulator are proposed based on the Euler-Lagrange formulation and in [Falkenhahn 2017] the dynamic control of the Bionic Handling Assistant based on a lumped mass model is presented.

Given the complexity involved in the derivation of the dynamics of continuum manipulators, model-less approaches have been proposed. Ivanescu et al [Ivanescu 2003] avoid the difficulties of solving the complex PDE system that describes the dynamics of continuum manipulators by proposing controllers based on an energy formulation. A model-less controller based on the estimation of the Jacobian under spatial constraints is presented in [Yip 2014] and [Yip 2016]. In this regard, controllers based on fuzzy-logic [Qi 2014] [Qi 2016] and adaptive algorithms [Melingui 2017] have been presented.

In medical applications, the use of *in-vivo* feedback information gathered by radio-graphic and electromagnetic images [Arai 1994] [Bertocchi 2006] have been used to control continuum manipulators with great success. Camera vision systems [Chitrakaran 2007] [Boudjabi 2003] have also been employed to provide feedback information for the controller. However, all these techniques rely on external sensors to close the loop. When the environment in which the manipulator works makes impossible the use of external sensors, controllers based on embedded sensors in closed-loop [Penning 2011] [Penning 2012] [Bajo 2011] can be used instead.



## 2.4 Work contextualization and contributions

Given the classification of the approaches towards the modeling and control of continuum manipulators presented in section 2.3.2, it is now possible to position the methodology presented in this work, in the context of the current literature. Fig. 2.13 provides a helpful visualization of the position of our method with respect of other previously proposed methods and their classification.

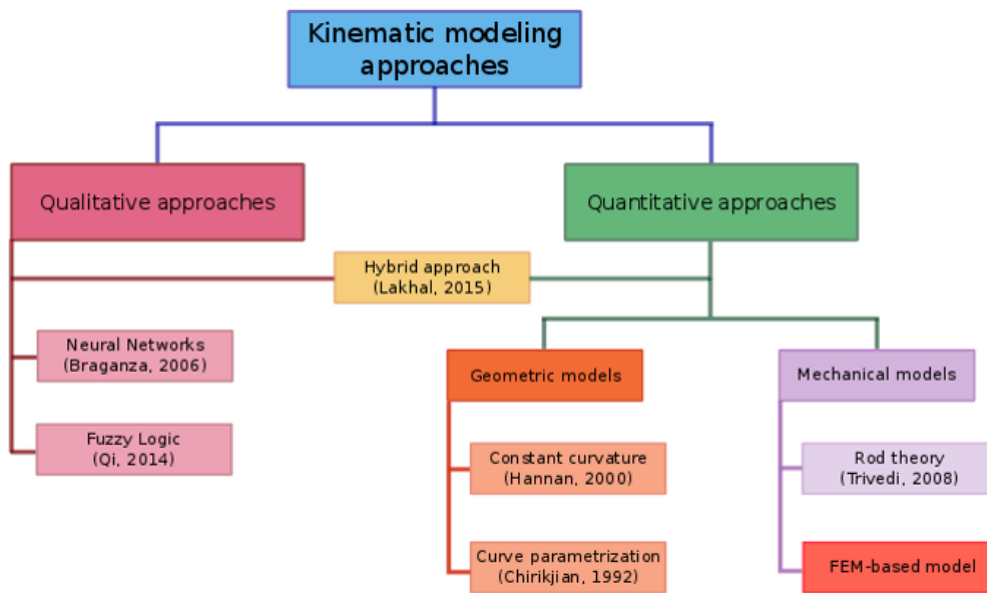


Figure 2.13: The context of the proposed method in the current state of the art

As it is illustrated in the figure, the proposed method is based on the mechanics of the materials, which means that no geometrical assumptions are done in order for the method to be applicable. Moreover, being a mechanics-based approach, the description provided by the method explicitly accounts for the internal and external loading of the continuum backbone, which means that the deformation caused by gravity or the load of a payload will be reflected in the model. Nevertheless, there are a few important differences between the method here discussed and other mechanic-based methods: First, while most methods are based on Cosserat rod and other elasticity theories, the proposed approach is based on Finite Element Analysis which provides geometrically accurate descriptions of the backbone. Second, the modeling

of actuators, sensors and end-effector is done implicitly by the formulation of the constraints, so no further modeling is required.

In the following, the contributions to the field derived from the development of this Ph.D. work are enlisted:

- FEM-based quasi-static modeling approach that accounts for complex structural shapes of single and multi-backbones.
- The model implementation of two types of actuation systems (i.e. tendons and pneumatic muscles) currently implemented in the majority of designs of soft manipulators.
- The integration of sensors in the model that allows for a description of the manipulator in the configuration space.
- A closed-loop control strategy for continuum manipulators based on the state estimation of the robot.

In the next chapter, the assumptions considered during modeling and the study framework of this methodology (namely Continuum Mechanics and Finite Element Method) are introduced.

# FEM-based model of Continuum Manipulators

---

## Contents

---

<b>3.1</b>	<b>Introduction</b>	<b>32</b>
<b>3.2</b>	<b>Continuum mechanics framework</b>	<b>33</b>
3.2.1	Constitutive material law	34
3.2.2	Forces in the continuum manipulator	35
<b>3.3</b>	<b>Finite Element Method</b>	<b>36</b>
<b>3.4</b>	<b>FEM-based kinematics of soft manipulators</b>	<b>38</b>
3.4.1	Constraint for the end-effector	39
3.4.2	Actuator constraint model	41
3.4.3	Sensor constraint model	44
<b>3.5</b>	<b>Reduced model in the constraint space</b>	<b>45</b>
3.5.1	Reduced compliance on the constraint space	45
3.5.2	Coupled Kinematic Equations	47
3.5.3	Inverse kinematic model solution by convex optimization	48
<b>3.6</b>	<b>Method implementation</b>	<b>49</b>
3.6.1	Simulation framework	50
3.6.2	Corotational FEM	50
3.6.3	Mesh generation	51
3.6.4	Description of the Compact Bionic Handling Assistant	51
3.6.5	Simulation of the CBHA	53
<b>3.7</b>	<b>Kinematic models</b>	<b>55</b>
3.7.1	Forward kinematic models	56
3.7.2	Inverse kinematic model	62
<b>3.8</b>	<b>Conclusion of the chapter</b>	<b>63</b>

---

### 3.1 Introduction

In contrast with rigid robots, soft manipulator kinematics not only depend on the geometry of the robot, but also on the mechanical properties of its material, in particular the stiffness, that dictates the net configuration of the robot under a set of external forces. **This statement represents the core of the modeling principle considered in this manuscript.** As an example, consider the manipulator in Fig. 3.1 (a). A tendon routed through the structure is pulled to achieve a desired end-effector position. In Fig. 3.1 (b), a manipulator with the same geometry but with inhomogeneous material is actuated in the same manner by the same amount, however, the resulting configuration differs from (a). This is because of the influence of the material stiffness in the kinematics of the manipulator.

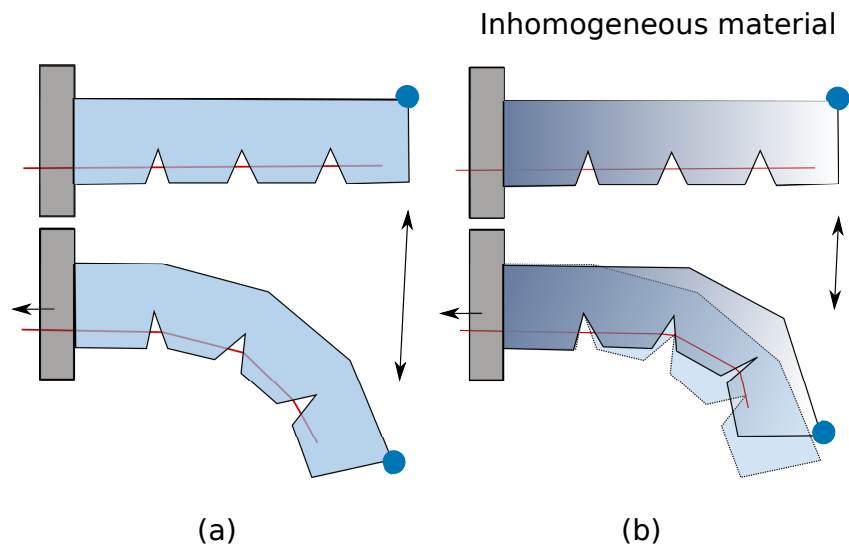


Figure 3.1: In this example a tendon is pulled to create the motion of an elastic soft robot. Starting with the same geometry, the material stiffness has an influence on the kinematics (output vs input displacements).

While rigid manipulator kinematics can be used to solve positioning problems with the assumption of resistance/counter-actuation to gravity or load effects, soft manipulators easily comply to these forces and deform. To answer the same problems of positioning, it is then necessary to take into account the current deformation (i.e. change of geometry) induced by these forces to obtain a kinematic relation between the po-

sition of the end-effector and the position of the actuators.

Another difference between soft continuum manipulators and rigid ones is the impact that external forces (particularly gravity) have on the final pose of the manipulator [Trivedi 2008b]. As mentioned before, in a continuum manipulator, there can be only a finite number of actuated degrees of freedom<sup>1</sup>. The state of the remainder of the (infinite) degrees of freedom in a continuum robot backbone will be determined by both the constraints of the controlled degrees of freedom and internal and external forces [Walker 2013a].

In this work, this problem is addressed by a discretization of the degrees of freedom of the continuum manipulator, through numerical methods provided by Computational Mechanics. In the following, the Continuum Mechanics framework is briefly introduced to help us describe the motion of the continuum manipulator based on its deformation and to help the reader to understand the assumptions made in this modeling methodology. This work also explains the methodology of kinematics estimation on a continuum manipulator based on FEM.

## 3.2 Continuum mechanics framework

From a Continuum Mechanics perspective, we will consider the structure of a continuum manipulator at a macroscopic level, that is, there is no space between the particles that compose the manipulator, the body of the manipulator fills the entire space it occupies. In this section, we present the basic notions of Continuum Mechanics with the example of a continuum manipulator. The interested reader may refer to the Appendix A.1 for a more formal introduction.

Consider the manipulator in Fig. 3.2. with all its material particles occupying a 3-dimensional domain with a starting configuration  $k_0$ . Given a set of external loads, the manipulator will *deform*, changing its geometry. The new *deformed* configuration of the manipulator is called  $k_i$ . A material particle of the manipulator will have a position  $\mathbf{P}$  in  $k_0$  and a position  $\bar{\mathbf{P}}$  in  $k_i$ . Then, we can always find a function  $M$  that *maps*  $\bar{\mathbf{P}}$  from  $\mathbf{P}$ .

---

<sup>1</sup>This statement can be extended to any physically realizable system.

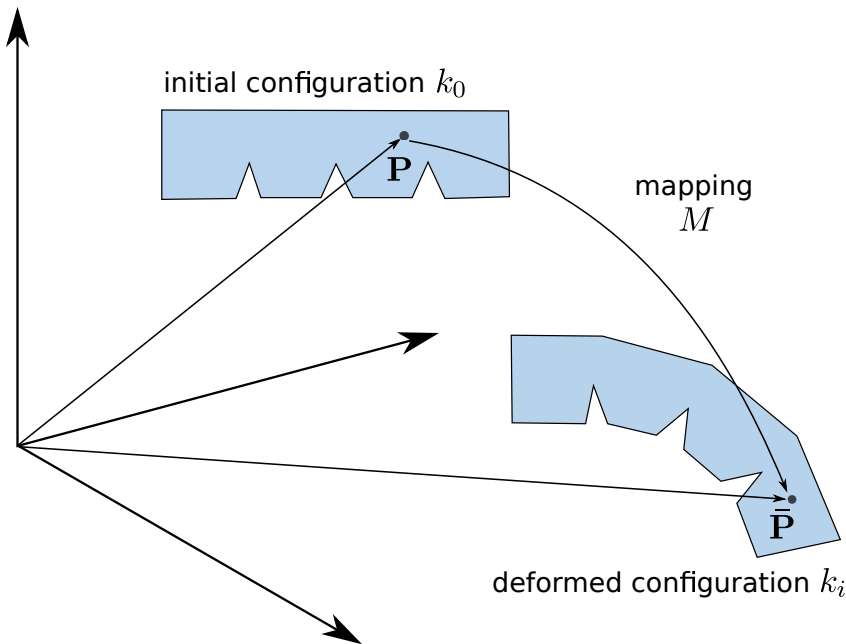


Figure 3.2: Illustration of the initial and deformed configurations of the continuum manipulator.

In the study of deformation, two important physical quantities arise: the stress and the strain.

**Stress** The deformation of the manipulator will cause internal body reaction forces. These forces are called *stress* which is defined as the force per unit area.

**Strain** The displacement of the particles in a continuum manipulator can be considered, in the general case, as a result of a rigid body motion and a deformation. A measure of *strain* is the computation of the deformation part of the displacement.

### 3.2.1 Constitutive material law

As stated before, the internal forces in a continuum manipulator are caused by its deformation. Therefore, it is of interest to find a relation between the stress and the strain. The stress-strain relationship is given by the *Constitutive law* and it depends on the material from which the manipulator is build. **We assume that the manipulator**

**is built from linear elastic materials.** For linear elastic materials, the constitutive law is represented by *Hook's law*.

In the experimental characterization of linear elastic materials, Hook's law is expressed in terms of the *Young's modulus*, which is a measure of the linear stiffness, the *Poisson's ratio*, which expresses a measure of the change in volume of the material, and the *shear modulus*. We will also make the assumption that the material of the manipulator is *isotropic*, which means that the material properties are independent from the direction they are measured. It means that we do not consider material made of fibers; however, this assumption does not prevent the use of different materials in the structure of the robot. In this case, only one Young's modulus, one Poisson's ratio, and one shear modulus characterize the material properties.

### 3.2.2 Forces in the continuum manipulator

In order to compute the response of the continuum manipulator to a set of known forces, the *Principle of Conservation of Linear Momenta*, commonly known as Newton's second law of motion is applied. This principle states that the rate of linear momentum of a collection of particles is always proportional to the forces acting on the particles.

The forces acting on the particles are the sum of internal and external forces. Internal forces are of two types, volumetric and boundary forces. The boundary forces are related to the stress by a tensor called *Cauchy's stress tensor* and they need to be integrated along the entire manipulator. This integration produces a differential equation describing the motion of the continuum manipulator. In general, this equation cannot be solved analytically because it involves an infinite number of degrees of freedom. Instead, numerical methods provided by the field of Computational Mechanics are used to approximate the solution. In the following, we will introduce the Finite Element Method as means of discretization of continuum manipulators.

### 3.3 Finite Element Method

In order to model the kinematics of continuum manipulators, the equation of motion provided by Continuum Mechanics needs to be solved. This equation however, cannot be solved analytically, hence the need of numerical methods to approximate its solution. One of these numerical methods is the Finite Element Method (FEM). For a more formal introduction to FEM, the interested reader can refer to the Appendix A.2

The FEM can be conceptually divided in 3 main steps: *discretization*, *discrete solution* and *assembly*. The main idea behind the FEM is the consideration of the continuum body, in this case the continuum manipulator, as being comprised of a set of subdomains, called *finite elements*. In this step, the theoretical infinite number of degrees of freedom in a continuum manipulator is reduced to a finite number, which makes the approximation of the solution attainable. The collection of finite elements that compose the structure under study is called *mesh*, as shown in Fig. 3.3. The generation of the mesh is an important process in FEM, since it defines the fidelity of the model. As a general rule, the model fidelity increases as we increase the number of elements in the mesh. In practice, the mesh is generated from a surface model of the manipulator, like those provided by Computer Assisted Design (CAD) software.



Figure 3.3: Illustration of a mesh. The volumetric mesh is created from a surface model of the manipulator.

Many choices exist for selecting the type of element that will compose the mesh. One of the considerations to make in this regard is the dimensions of the problem. For example, for 1-dimensional problems, the typical choice is the two-node truss; for 2-dimensional problems plate or shell elements are often selected, and for 3-dimensional problems the hexahedron or tetrahedron element are a good example of volumetric elements (see Fig. 3.4).



The type of element selected for the mesh will determine the *nodal interpolation*, which is basically the expression of any physical quantity inside the mesh depending only on the values at the nodes. The accuracy of the interpolation depends on the amount of nodes an element has and on the choice of interpolation or *shape* functions. These functions can be linear or higher order polynomials. Once the mesh is generated, the solution to the differential equations is approximated over each element.

For the solution of the problem at the element level, each element is disassembled from the mesh and is referred to a *reference element* in which the computations are made easier. Because of this process, the nodes in each element will have a local numeration (at the element level) and a global numeration (at the mesh level). The final step in the FEM deals with the re-assembly of the elements using its global numeration. To find the global equation system we must assemble all the element equations. In other words, we must combine local element equations for all the elements used in the discretization.

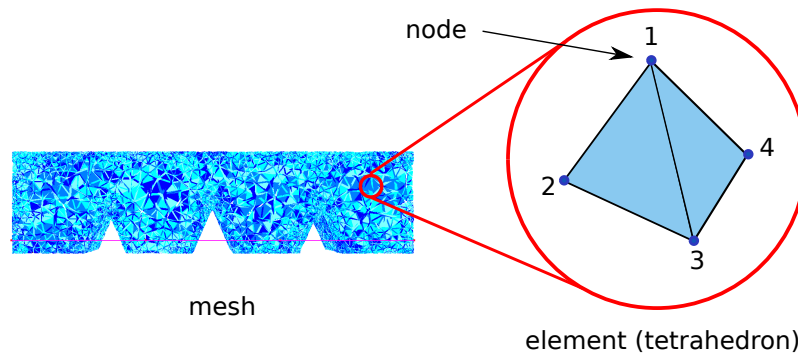


Figure 3.4: Illustration of an element. One of the many (tetrahedra) elements composing the mesh.

Before the solution of the global system, boundary conditions should be imposed. Boundary conditions in the most simple case, occur at the fixation points of the manipulator (where the manipulator is attached to its base). In this case, the imposition of this boundary in the global system is simple; the equations for the nodes that are fixed to the base are eliminated from the global system. Another, more complex boundary condition is the one that represents the actuators, end-effector and sensors in our soft, continuum manipulator. In our work, these conditions are imposed as constraints in the model and allow for the kinematic descriptions of the manipulator. In the next part of this chapter, the

derivation of the kinematic models for continuum manipulators based on the aforementioned constraints will be explained.

### 3.4 FEM-based kinematics of soft manipulators

The computation of the kinematic model of continuum manipulators developed in this section depend on the simulation of the FEM model explained in 3.3. To arrive at the models of continuum manipulators, we shall start from the formulation given by Newton's second law that models the dynamic behavior of the robot as

$$\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}} = \mathbf{F}_{\text{ext}} - \mathbf{F}(\mathbf{x}, \dot{\mathbf{x}}) + \mathbf{H}^T \boldsymbol{\lambda} \quad (3.1)$$

where  $\mathbf{x} \in \mathbf{R}^n$  is the vector of generalized degrees of freedom (in this case, the position of the nodes),  $\mathbf{M}(\mathbf{x}) : \mathbf{R}^n \mapsto \mathcal{M}^{n \times n}$  is the inertia matrix of the manipulator,  $\dot{\mathbf{x}} \in \mathbf{R}^n$  is the velocity vector,  $\ddot{\mathbf{x}} \in \mathbf{R}^n$  is the acceleration vector,  $\mathbf{F}_{\text{ext}}$  represents the external forces (like gravity),  $\mathbf{F}(\mathbf{x}, \dot{\mathbf{x}})$  represents the non-linear internal forces that depends on the generalized state of the manipulator and  $\mathbf{H}^T \boldsymbol{\lambda} \in \mathbf{R}^n$  gathers the contribution of the constraints forces (actuators, sensors and end-effector) as *Lagrange multipliers*.

In this work, **the study is limited to quasi-static behavior on purpose**. Thus, in a first approach, the assumption is that the control of the robot is performed at low velocities. As such, the term  $\mathbf{M}(\mathbf{x})\ddot{\mathbf{x}}$  in Eq. 3.1 that corresponds to the inertia effects is removed. On this basis, Eq. 3.1 for the quasi-static analysis of the continuum manipulator becomes

$$\mathbf{F}_{\text{ext}} - \mathbf{F}(\mathbf{x}) + \mathbf{H}^T \boldsymbol{\lambda} = 0 \quad (3.2)$$

The internal forces  $\mathbf{F}(\mathbf{x})$ , which are obtained from the FEM of the manipulator are non-linear. We use the Taylor series expansion to compute a unique linearization of  $\mathbf{F}(\mathbf{x})$  per simulation step. During each step  $i$  of the simulation, the internal forces are updated as

$$\mathbf{F}(\mathbf{x}_i) \approx \mathbf{F}(\mathbf{x}_{i-1}) + \mathbf{K}(\mathbf{x}_{i-1})d\mathbf{x} \quad (3.3)$$

where  $\mathbf{K}(\mathbf{x})$  is the tangent stiffness matrix that depends on the actual positions of the nodes and  $d\mathbf{x}$  is the difference between positions  $d\mathbf{x} = \mathbf{x}_i - \mathbf{x}_{i-1}$ . This linearization is valid as long as the displacement of the nodes  $d\mathbf{x}$  is small.

As mentioned before, the lines and columns that correspond to fixed nodes are removed from the system to get a full rank for matrix  $\mathbf{K}$ . In  $\mathbf{F}$  and  $\mathbf{K}$ , the rows (and columns for  $\mathbf{K}$ ) contain the components of the internal forces  $(x, y, z)$  for the nodes, in the order corresponding to their global node numbering in the mesh.

The way  $\mathbf{H}$  is obtained is explained in the following, but it is important to mention that its computation is performed with the values obtained from the previous simulation step. We then use the expression  $\mathbf{H}(\mathbf{x}_{i-1})$  and through the linearization explained in Eq. 3.3, we obtain the following formulation :

$$-\mathbf{K}(\mathbf{x}_{i-1})d\mathbf{x} = \mathbf{F}_{\text{ext}} + \mathbf{F}(\mathbf{x}_{i-1}) + \mathbf{H}(\mathbf{x}_{i-1})^T \boldsymbol{\lambda} \quad (3.4)$$

The variables  $d\mathbf{x}$  and  $\boldsymbol{\lambda}$  are both unknown and are found during the optimization process. It should also be noticed that the matrix  $\mathbf{K}$  is highly sparse. In the implementation, a conjugate gradient solver is used and preconditioned by a sparse  $\mathbf{LDL}^T$  decomposition. For a mesh composed of about 1000 nodes and about 3000 tetrahedral elements, a refresh rate of 60Hz is obtained with the implementation available in the simulation framework.

### 3.4.1 Constraint for the end-effector

To set the Lagrange multiplier on the end-effector, a point or a set of points of the robot need to be considered as the end-effector; these are the points we seek to control in position. It could be any points mapped on the FE mesh. For each point, the constraint objective is to reduce the difference between the end-effector position and its desired position  $\mathbf{p}_{des}$ . Thus, a function  $\boldsymbol{\delta}_e(\mathbf{x}) : \mathbf{R}^{3n} \rightarrow \mathbf{R}^3$  with  $n$  being the number of nodes, evaluates this difference along  $x$ ,  $y$  and  $z$  (see Fig. 3.5). If the end-effector corresponds to a node  $i$  of the mesh, the function is  $\boldsymbol{\delta}_e(\mathbf{x}) = \mathbf{x}_i - \mathbf{p}_{des}$ , where  $\mathbf{x}_i$  is the position of node  $i$ . If the effector is set inside an element, we use

$$\delta_e(\mathbf{x}) = \sum_{i=0}^n \phi_i(\mathbf{p}_{eff}) \mathbf{x}_i \quad (3.5)$$

where  $\mathbf{p}_{eff}$  is the position of the end-effector in the rest configuration of the FEM model and  $\phi_i$  is the shape (interpolation) function associated to node  $i$ .

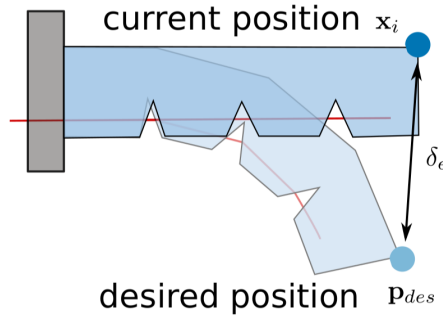


Figure 3.5: Illustration of the end-effector constraint.  $\delta_e$  represents the difference between current and desired position.

If several points are used for the end-effector position, the vector  $\delta_e(\mathbf{x})$  gathers the evaluation of the difference for all the points. The function is then  $\mathbf{R}^{3n} \rightarrow \mathbf{R}^{3m}$ , where  $m$  is the number of end-effector points.

The matrix  $\mathbf{H}$  used for the end-effectors corresponds to  $\mathbf{H}_e(\mathbf{x}) = \frac{\partial \delta_e(\mathbf{x})}{\partial \mathbf{x}}$ .

The matrix  $\mathbf{H}_e$  is highly sparse: A row, that corresponds to a component of a point of the end-effector, will contain non null values on a very small number of columns. As the point is mapped on a single tetrahedral element, there is a maximum of 4 non-null values per row. Of course, the column should match with the components of the nodes, given the fact that the non-null values are gathered in 3x3 diagonal block matrices.

Finally, an important point is the effort value that is put on the Lagrange multiplier that corresponds to the terminal effector. The value of  $\lambda_e$  will depend on the load applied on the end-effector. Two cases can be considered:

- I If the points defined as end-effector move freely in the space, there is no physical interaction, so the contribution of the constraint vanishes  $\lambda_e = \mathbf{0}$ .

- II If one or several points of the end-effector carries one object  $l$  which mass creates a load that could deform the structure. In such cases, the corresponding load should be set on  $\boldsymbol{\lambda}_e = m_l \mathbf{g}$  with  $m_l$  being the mass of the object and  $\mathbf{g}$  the gravity field.

In both cases, it is assumed that the value of the Lagrange multiplier  $\boldsymbol{\lambda}_e$  is known, whether by experimentation or identification.

### 3.4.2 Actuator constraint model

One can find different actuation schemes for soft, continuum manipulators in the literature. From piezo electric actuation [Su 2012], electro-rheological fluids [Sadeghi 2012], to electro-active polymers [Chikhaoui 2014]; however, as implied in 2.3.1, the two most prominent actuation schemes for continuum robots remain the pneumatic actuators, and cables or tendons. As such, these two types of actuation are considered in this work.

The model of the actuators considers their physical characteristics. The contributions of these actuator constraints are unknown before the optimization process. However, given the type of actuation, the constraint is not set in the same way.

#### 3.4.2.1 Cable actuator

In a first case (Fig 3.6), a cable is used to actuate the manipulator. The cable can simply be attached at one point of the structure, but it can also go through several other points or guides; in that case, **frictionless guides are considered**. In all cases, the unknown  $\boldsymbol{\lambda}_a$  is the stretching force inside the cable. It is evident that this force is unilateral ( $\boldsymbol{\lambda}_a \geq 0$ ) since we cannot *push* with the cable.

Let's suppose now that the points are numbered starting from the extremity where the cable is being pulled. The matrix  $\mathbf{H}$  is computed this way: At each point  $p$ , representing one of the cable guides (Fig. 3.6), we take the direction of the cable *before*  $p$

$$\mathbf{d}_b = \frac{\mathbf{x}_p - \mathbf{x}_{p-1}}{\|\mathbf{x}_p - \mathbf{x}_{p-1}\|} \quad (3.6)$$

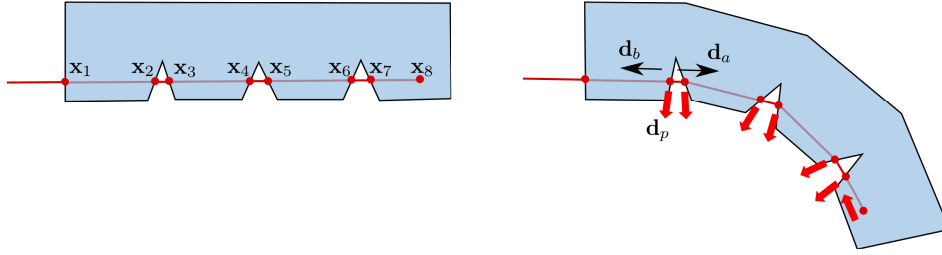


Figure 3.6: Cable actuation.  $\mathbf{d}_b$  and  $\mathbf{d}_a$  on the figure, represent the direction of the tendon before and after the cable guide, respectively, which are used to compute the normal forces at the guides.

and *after*  $p$

$$\mathbf{d}_a = \frac{\mathbf{x}_{p+1} - \mathbf{x}_p}{\|\mathbf{x}_{p+1} - \mathbf{x}_p\|} \quad (3.7)$$

To obtain the constraint direction that is applied to the point, we use

$$\mathbf{d}_p = \mathbf{d}_a - \mathbf{d}_b \quad (3.8)$$

Note that the direction of the final point is equal to the direction "before" as  $\mathbf{d}_a$  does not exist. These constraint directions are mapped on the nodes using the interpolation

$$\begin{bmatrix} \vdots \\ \mathbf{F}_n \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \phi_n(\alpha, \beta, \gamma) \mathbf{d}_p \\ \vdots \end{bmatrix} \lambda_a = \mathbf{H}_a^T \lambda_a \quad (3.9)$$

A function  $\delta_a(\mathbf{x})$  is defined to provide the length of the cable, given the position of the constrained nodes. This constraint formulation allows also to consider the physical limitations of the actuators, which in this case is the cable stroke. The actuator stroke can also be included by imposing

$$\delta_a(\mathbf{x}) \in [\delta_{min} \ \delta_{max}] \quad (3.10)$$

Through the use of this function, we get

$$\mathbf{H}_a = \frac{\partial \delta_a(\mathbf{x})}{\partial \mathbf{x}} \quad (3.11)$$

### 3.4.2.2 Pneumatic actuator

The formulation is compatible with pressure-based actuation of cavities that are placed in the structure. In that case, the effort  $\lambda_a$  is the uniform pressure exerted on the wall of the cavity, while  $\delta_a$  represents the volume. As the pressure is uniform inside the cavity, a single constraint can be set for each pneumatic actuator; however, all triangles of the cavity wall will be involved.

For each triangle  $t$ , the area  $a_t$  and the normal direction  $\mathbf{n}_t$  are computed. If this result is multiplied by the pressure, we obtain the force applied by the pneumatic actuator on the triangle  $t$ . We distribute this contribution to each of its nodes by dividing the resulting vector by 3. We sum up the results of each triangle in the corresponding column of  $\mathbf{H}_a^T$ . This gives us the corresponding force acting on each of the nodes of all the triangles that compose the cavity

$$\mathbf{F}_i = \sum_{t \in S, i \in t} \frac{a_t}{3} \mathbf{n}_t \lambda_a = (\mathbf{H}_a^T)_i \lambda_a \quad (3.12)$$

where  $\mathbf{F}_i$  is the pressure force assigned to the node  $i$  and  $S$  is the set of triangles composing the cavity.

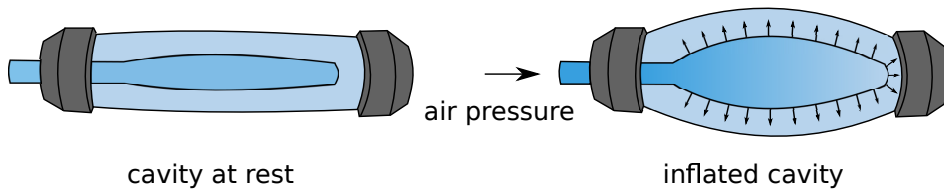


Figure 3.7: A pneumatic actuator. The cavity at rest (left) is then pressurized with air (right). The pressure applies a normal force to the walls of the cavity.

Usually, pneumatic actuators only provide positive pressure so  $\lambda_a \geq 0$ . However, in some cases, it is also possible to create both negative and positive pressure using vacuum/pressure actuation. In that case, there is no particular constraint on the unknown value of  $\lambda_a$ , despite an eventual limit (max / min) of pressure that can be achieved by the actuator.

### 3.4.3 Sensor constraint model

In order to relate the end-effector position and the geometry of the manipulator, one needs sensors that can measure the geometrical state or *shape* of the robot. In the literature, this task is done either by using external sensors, such as stereo-vision systems and x-ray cameras [Otake 2014], or by sensors embedded in the body of the manipulator. **In this study, we assume that the robot has embedded sensors.** Various studies have been conducted to evaluate the feasibility of different types of embedded sensors for shape reconstruction, including electromagnetic and piezoelectric polymers [Cianchetti 2012], cable length measurement [Segreti 2012] [Murphy 2014] and Fiber Bragg grating (FBG) [Farvardin 2016] [Searle 2013]. We can summarize the common aim of these techniques as the estimation of the configuration of the manipulator by measuring the relative position of a set of points along a *string*.

In this study, we implement the sensors in the simulation, using the same string concept. As in the case of the cable actuator, the string of the sensor is routed through several friction-less guides, at  $n$  points  $\mathbf{x}_n$ , as shown in Fig. 3.8. In the model, the measure of the lengths read by the sensor will be

$$\sum_{i=1}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\| \quad (3.13)$$

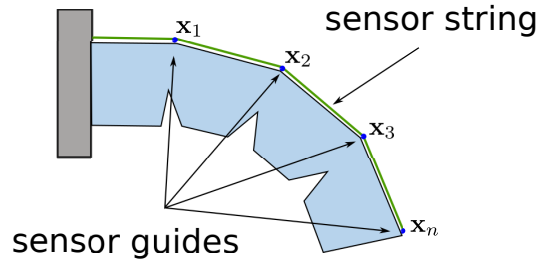


Figure 3.8: Example of an embedded sensor. The sensor string is routed through  $n$  frictionless guides.

which evaluates the distance between each sensor guide after the position of the nodes has been updated. A function  $\delta_s$  is defined to represent the current length of the sensor string given the position of the constrained



nodes. The matrix  $\mathbf{H}_s$  that gathers the directions of the sensor constraint is obtained in the same way as for the cable actuator. Since the sensor does not apply any forces to the manipulator,  $\boldsymbol{\lambda}_s$  is equal to 0.

## 3.5 Reduced model in the constraint space

The classical resolution of a deformable body motion using FEM (like solving the static equilibrium of the structure described at equation 3.4) provides a forward model: it allows to compute the displacements of the structure, given the values of the efforts put on the actuator  $\boldsymbol{\lambda}_a$ . However, in the case of position control, the actuation  $\boldsymbol{\lambda}_a$  is the unknown. Consequently, for controlling the motion of the manipulator, an inverse model is needed, which is challenging to compute in real-time as the size of the system is in the range of several thousands degrees of freedom. In this work, another approach is used, based on the projection of the mechanics in the constraint space that drastically reduces the size of the optimization problem. This approach, initially developed in [Duriez 2013], is generalized. A new formulation of the inverse problem in the form of a quadratic programming (QP) optimization (developed in [Largilliere 2015]) is used.

### 3.5.1 Reduced compliance on the constraint space

As stated above, the optimization process relies on a projection of the mechanics in the constraint space. Each constraint has a direction that is set by a line of the matrices  $\mathbf{H}_e$ ,  $\mathbf{H}_a$  and  $\mathbf{H}_s$ . This matrix is usually sparse, as the direction of the constraints is mapped on few nodes of the FE mesh. The values of the effort applied by the actuators  $\boldsymbol{\lambda}_a$  are not known at the beginning of the optimization process, whereas the value of  $\boldsymbol{\lambda}_e$  is supposed to be known and the value of  $\boldsymbol{\lambda}_s$  is 0.

The first step consists of obtaining a free configuration  $\mathbf{x}_{\text{free}}$  of the robot which is found by solving the equation 3.4 while considering that there is no actuation applied to the deformable structure. In practice, the known value of  $\boldsymbol{\lambda}_e$  is used and  $\boldsymbol{\lambda}_a = 0$  is imposed.

The linear equation 3.4 is solved using a  $\mathbf{LDL}^T$  factorization of the matrix  $\mathbf{K}$ . Given this new *free* position  $\mathbf{x}_{\text{free}}$  for all the nodes of the mesh, one can evaluate the values of

$$\boldsymbol{\delta}_e^{\text{free}} = \boldsymbol{\delta}_e(\mathbf{x}_{\text{free}}) \quad (3.14)$$

that represent the shift between the effector point(s) position and the desired position introduced in section 3.4.1. One can also evaluate

$$\boldsymbol{\delta}_a^{\text{free}} = \boldsymbol{\delta}_a(\mathbf{x}_{\text{free}}) \quad (3.15)$$

which represents the position of the actuated points without actuation effort.

From the FEM formulation of the problem that uses the global matrix  $\mathbf{K}$  (Eq. A.55), a formulation that accounts for the directions of the constraints placed for actuators and end-effectors is derived. Using the Schur complement of matrix  $\mathbf{K}$  in the Lagrange multiplier-augmented system [Przemieniecki 1985], a small formulation of  $\boldsymbol{\delta}_e$  is obtained. This variable expresses the difference between the desired position for the end-effector and its current position in terms of the actuators contributions  $\boldsymbol{\lambda}_a$ :

$$\boldsymbol{\delta}_e = \underbrace{[\mathbf{H}_e \mathbf{K}^{-1} \mathbf{H}_a^T]}_{\mathbf{W}_{ea}} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_e^{\text{free}} \quad (3.16)$$

The Schur complement also provides similar formulations for the difference between a desired sensor or actuator position and its current position:

$$\boldsymbol{\delta}_a = \underbrace{[\mathbf{H}_a \mathbf{K}^{-1} \mathbf{H}_a^T]}_{\mathbf{W}_{aa}} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_a^{\text{free}} \quad (3.17)$$

$$\boldsymbol{\delta}_s = \underbrace{[\mathbf{H}_s \mathbf{K}^{-1} \mathbf{H}_a^T]}_{\mathbf{W}_{sa}} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_s^{\text{free}} \quad (3.18)$$

This step is central in the method. It consists of projecting the mechanics into the constraint space. As the constraints are the inputs (effector position shift and sensor length shift) and outputs (effort to apply on the actuators) of the inverse problem, the smallest possible projection space for the inverse problem is obtained. It allows for a projection that drastically reduces the size of the search space without loss of information. Indeed, the following subsection shows how the matrices  $\mathbf{W}_{ea}$  and

$\mathbf{W}_{aa}$  provide the mechanical coupling equations between actuators and effector point(s).

After this projection, the optimization is processed in the reduced constraint space to get the values of  $\boldsymbol{\lambda}_a$ . This part is described in 3.5.3.

The final configuration of the soft robot, at the end of the time step, is obtained as

$$\mathbf{x} = \mathbf{x}_{\text{free}} + \mathbf{K}^{-1} \mathbf{H}_a^T \boldsymbol{\lambda}_a \quad (3.19)$$

It should be emphasized that one of the main difficulties in this implementation is the computation  $\mathbf{W}_{ea}$  and  $\mathbf{W}_{aa}$  in a fast manner. No pre-computation is possible as their value changes at each iteration. However, this type of projection problem is frequent when solving friction contact on deformable objects. Thus, several strategies are already implemented in SOFA.

### 3.5.2 Coupled Kinematic Equations

Using the compliance operator  $\mathbf{W}_{ea}$ , one can get a measure of the mechanical coupling between effector and actuator, and with  $\mathbf{W}_{aa}$ , the coupling between actuators. For instance, the displacement  $\delta_e^i$  created on the end-effector (along a direction stored on the line  $i$  of matrix  $\mathbf{H}_e$ ) by a unitary force  $\lambda_a^j$  applied by the actuator (which is stored at the line  $j$  of matrix  $\mathbf{H}_a$ ) is directly obtained by

$$\Delta \delta_e^i = w_{ea}^{ij} \lambda_a^j + \delta_e^{i,\text{free}} \quad (3.20)$$

As the motion is created by deformation, the motion of the actuator  $j$  is influenced by actuator  $k$ . Through the same principle, actuator  $k$  also influences the displacement of the end-effector. To get a kinematic link between actuators and effector, the method needs to account for the mechanical coupling that can exist between actuators. It is captured by  $\mathbf{W}_{aa}$  that can be inverted if actuators are defined on independent degrees of freedom. Thus one can get a kinematic link by rewriting equation (3.17) as:

$$\boldsymbol{\delta}_e = \mathbf{W}_{ea} \mathbf{W}_{aa}^{-1} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_a^{\text{free}}) + \boldsymbol{\delta}_e^{\text{free}} \quad (3.21)$$

Equation (3.21) is composed of a reduced number of linear equations that relate the displacement of the actuators to the displacement of the effector. Consequently, matrix  $\mathbf{W}_{ea} \mathbf{W}_{aa}^{-1}$  is equivalent to the Jacobian matrix of a rigid manipulator. This matrix is a local linearization provided by the FEM model on a given position and it needs to be updated for deformations with large displacements.

In conclusion, with basic manipulation of the matrices  $\mathbf{W}_{ea}$  and  $\mathbf{W}_{aa}$ , a condensed direct model of the soft manipulator can be obtained.

### 3.5.3 Inverse kinematic model solution by convex optimization

The goal of the optimization is to find how to actuate the structure so that the end-effector of the robot reaches a desired position. This was initially proposed in [Largilliere 2015]. It consists of reducing the norm of  $\delta_e$  which actually measures the shift between the end-effector and its desired position. Thus, computing  $\min(\frac{1}{2} \delta_e^T \delta_e)$  leads to a Quadratic Programming (QP) problem

$$\min \left( \frac{1}{2} \lambda_a^T \mathbf{W}_{ea}^T \mathbf{W}_{ea} \lambda_a + \lambda_a^T \mathbf{W}_{ea}^T \delta_e^{\text{free}} \right) \quad (3.22)$$

*subject to* (course of actuators) :

$$\delta_{\min} \leq \delta_a = \mathbf{W}_{aa} \lambda_a + \delta_a^{\text{free}} \leq \delta_{\max} \quad (3.23)$$

*and* (case of unilateral effort actuation) :

$$\lambda_a \geq 0$$

The use of a minimization allows to find a solution even when the desired position is out of the workspace of the robot. In such a case, the algorithm will find the point that minimizes the distance to the desired position while staying in the limits introduced by the course of the actuators.

In practice, the QP solver available in the Computational Geometry Algorithms Library (CGAL) [Fabri 1998] is used. The matrix of the QP  $\mathbf{W}_{ea}^T \mathbf{W}_{ea}$  is symmetric.

If the number of actuators is equal or inferior to the size of the end-effector space, the matrix  $\mathbf{W}_{ea}^T \mathbf{W}_{ea}$  is also definite. In such a case, the solution of the minimization is unique. In the case when the number of actuators is greater than the degrees of freedom of the effector points, the matrix of the QP is only semi-definite. Consequently, the solution could be non-unique.

A new criterion for the minimization can be introduced, based on the deformation energy. Indeed, this energy  $E_{def}$  is linked to the mechanical work of the forces exerted by the actuators.  $E_{def}$  can be computed by evaluating the dot product between  $\boldsymbol{\lambda}_a$  and the displacements of the actuators

$$\Delta \boldsymbol{\delta}_a = \boldsymbol{\delta}_a - \boldsymbol{\delta}_a^{\text{free}} \quad (3.24)$$

due to the actuator forces

$$E_{def} = \boldsymbol{\lambda}_a^T \Delta \boldsymbol{\delta}_a = \boldsymbol{\lambda}_a^T \mathbf{W}_{aa} \boldsymbol{\lambda}_a \quad (3.25)$$

Matrix  $\mathbf{W}_{aa}$  is definite positive if the actuators are placed on different nodes of the FEM or with different directions (i.e. if there is no linear dependencies between lines of  $\mathbf{H}_a$ ). Thus, one can add this energy in the minimization process by replacing (3.22) with:

$$\min \left( \frac{1}{2} \boldsymbol{\lambda}_a^T (\mathbf{W}_{ea}^T \mathbf{W}_{ea} + \nu \mathbf{W}_{aa}) \boldsymbol{\lambda}_a + \boldsymbol{\lambda}_a^T \mathbf{W}_{ea}^T \boldsymbol{\delta}_e^{\text{free}} \right) \quad (3.26)$$

with  $\nu$  chosen sufficiently small so that the deformation energy does not disrupt the quality of the effector positioning. In practice,

$$\nu = \frac{\text{tr}(\mathbf{W}_{ea}^T \mathbf{W}_{ea})}{\text{tr}(\mathbf{W}_{aa})} * 10^{-3} \quad (3.27)$$

is chosen so that the term  $\nu \mathbf{W}_{aa}$  does not alter the value of the QP matrix. Thanks to this modification, the QP matrix becomes definite positive and a unique solution of the problem can be found.

## 3.6 Method implementation

In this section, the inverse FEM method is implemented in order to obtain forward and inverse kinematic relationships for a geometrically com-

plex continuum manipulator. Two different descriptions are obtained for the inverse kinematics; one that relates the position of the end-effector to the actuator space, and another that relates the position of the end-effector to the geometric configuration of the robot. The kinematic models are used to pilot the robot in feedforward configuration. The forward kinematic model obtained from this approach is compared in terms of position accuracy to two different geometric models, developed for the same robot. Finally, the results are discussed.

### 3.6.1 Simulation framework

SOFA [Faure 2012] is a simulation framework first released in 2007, originally intended for the interactive computation of biomechanical models for soft tissue. The framework is an open source C++ library with a modular architecture in which the simulations are built as scenes. The scenes are composed by various components that encapsulate one of the aspects of a simulation, such as the mechanical objects representing the degrees of freedom, the forces and constraints, the differential equations, the main loop algorithms, the linear solvers, the collision detection algorithms or the interaction devices.

At the present time, an auxiliary module, or plug-in for SOFA, dedicated entirely towards the modeling and control of soft robots and its environment is in development. This plug-in allows for a rapid generation of mechanic-based models of soft devices and their interaction with the environment. The algorithm proposed by Duriez in [Duriez 2013] is at the core of this plug-in and allows for the direct control of soft robots based on the simulation of the inverse FEM of the robots.

In order to solve non-linear deformations, while providing a computationally economic method that is able to simulate the robots in real time, a *corotational* formulation of volume FEM, like the one developed in [Felippa 2005], [Felippa 2000] is implemented in SOFA.

### 3.6.2 Corotational FEM

In physical simulation of deformable bodies with small strains, the non-linear part of the deformation of the body is in most cases neglected. However, when the elements undergo a rigid transformation, this non-linear deformation is no longer small compared with the linear part and can no longer be neglected. Felippa and Haugen [Felippa 2005]

proposed the computation of the rigid transformation by adding an additional frame of reference, called *corotated configuration*, attached to the elements that moves with it like a "shadow". Element deformations are measured with respect to the corotated configuration. The element stress field and tangent stiffness matrix are corrected using the element rotations. A significant feature of this implementation is that it allows for large rigid motions, as long as the strains are small. The implementation is element-independent and compatible with linear finite elements, which makes it particularly suitable for real-time FEM simulation.

### 3.6.3 Mesh generation

The mesh generation is a critical part in the simulation process. In our case, volumetric meshes (volume elements) are required. The Computational Geometry Algorithms Library (CGAL) is a software written in C++ that allows for the generation of surface and volume meshes. The input to the CGAL software are surface objects in stereolithography format (.stl), like those generated by a Computer Assisted Design Software (CAD). In practice, we use only one type of element for a given mesh; however, in some cases, different mesh subdomains are required, for example, to represent heterogeneous materials. CGAL is able to generate meshes composed of different subdomains while respecting the physical interfaces; elements do not cross interfaces.

### 3.6.4 Description of the Compact Bionic Handling Assistant

The CBHA is the bionic continuum manipulator component of the RobotinoXT, a didactic mobile platform designed by Festo Robotics [Festo 2012]. The system is shown in Fig. 3.9 (a). The bionic continuum manipulator is formed by 2 serially connected sections of pneumatic actuators, an axially rotating wrist and a compliant gripper. Without actuation, the manipulator has a length of 206mm, with each section having a length of 103mm. The width at the base of the manipulator is 100mm long and the top has 80mm of width. In our study, the end of the second section is considered as the end-effector.

Each section of the manipulator is composed of an array of pneumatic actuators or bellows, connected in parallel configuration, as shown in

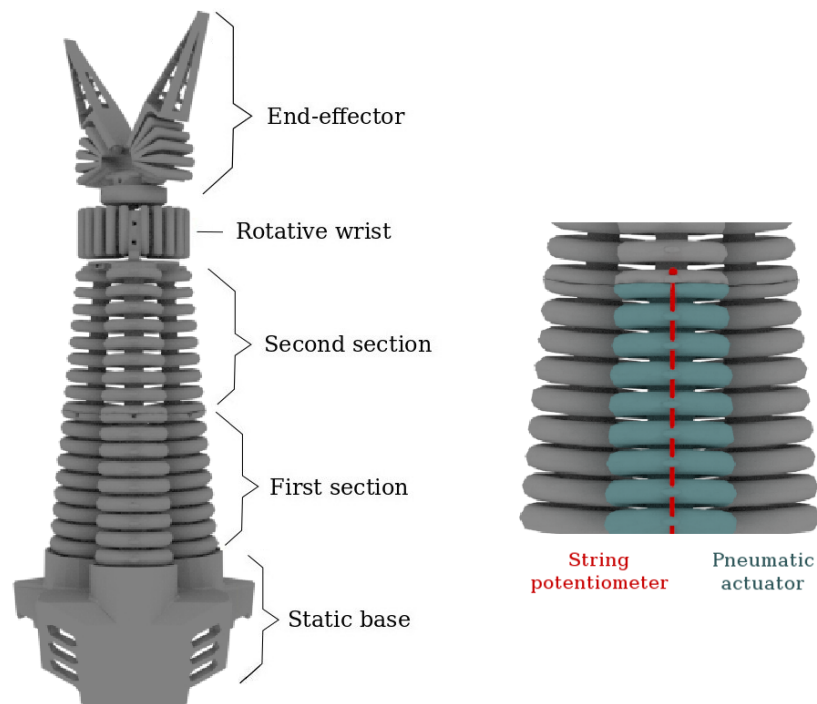


Figure 3.9: The CBHA by Festo Robotics. (a) The anatomy of the Compact Bionic Handling Assistant. (b) A section of the manipulator, composed by 3 pneumatic actuators and their correspondent length sensor.

Fig. 3.9 (b). By applying different pressures to the bellows, each section can bend independently (Fig. 3.10).

The pressure inside the cavities is regulated by a piezo proportional valve terminal, which includes its own pressure regulators. The air is compressed by two small membrane pumps that can attain a pressure of 2.5 to 2.7 bars. However, the nominal operation pressure for a single cavity in practice is in the range of 0.3 to 1 bar. The pose of the manipulator is obtained as the contribution of the poses of the 2 sections.

In order to sense the state of the robot, string potentiometers measure the lengths of the actuators. The control unit in the RobotinoXT is a EA09 control board that handles the sensor readings and drive units. An Application Programming Interface (API) embedded in the system provides communication with external applications via ethernet connection



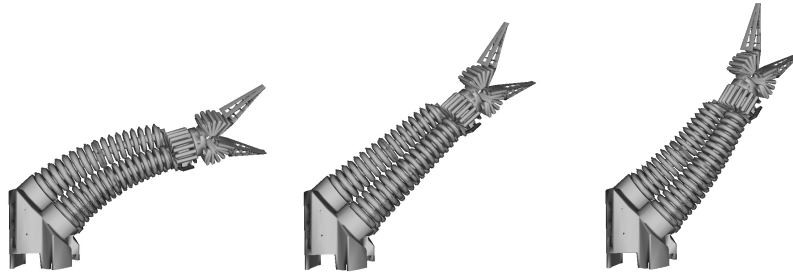


Figure 3.10: Examples of configurations of the CBHA based on the pressure inside the pneumatic actuators.

and W-LAN.

The manipulator is built by a process of Additive Manufacturing (AM), which is the name to describe the technologies that build 3D objects in a *layer-upon-layer* fashion. The material employed in the CBHA construction is Polyamide-Nylon, which has a Young's Modulus in the range of 1.6 to 2.3 GPa and a Poisson's ratio of 0.39.

### 3.6.5 Simulation of the CBHA

In order to simulate the CBHA and kinematically control it, a mesh, composed by 3528 tetrahedra is created from the surface representation of its structure (Fig.3.11). To allow for a fast computation of such a big model, a model reduction technique, based, on a domain decomposition is performed (see Appendix B). We select the end-effector point to correspond to the tip of the second section and each pneumatic actuator is modeled following the procedure described in 3.4.2.

The implementation of the simulation that allows for the interactive control of the robot is handled by an external computer featuring an Intel Xeon CPU W3690 at 3.47GHz x6 with a performance, in terms of simulation steps, of 18.4 frames per second. The communication between the simulation and the robot is done via W-LAN with a latency of 3ms, and when actuation commands are sent, the pressure regulators in the CBHA have a response time of 10ms.

As implied in 3.4.1, one of the features that sets apart this modeling approach is the ability to predict yielding of the structure under external

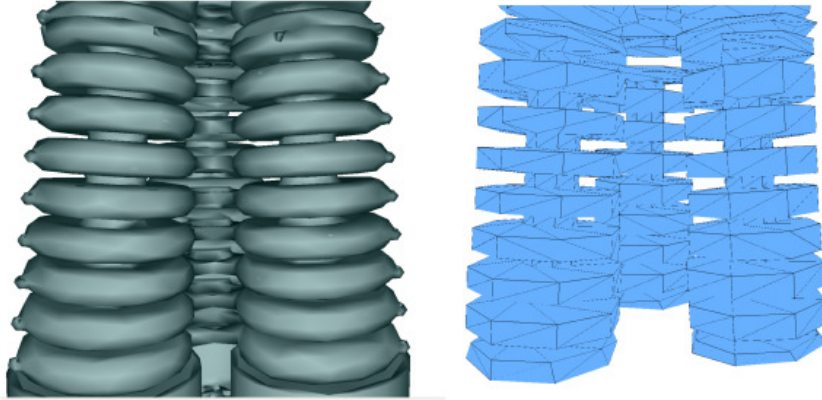


Figure 3.11: Visual surface model of the CBHA (left) and the FEM mesh used in simulation (right)

loading given an accurate parametrization of the material mechanics. This feature is of special importance when the robot is large in comparison to its thickness or when the orientation of the structure as a whole cannot be changed to counteract the effects of gravity. When the loads are known *a priori*, the magnitude of the force acting on the end-effector  $\lambda_e$  is used in Eq. 3.4 to compute the configuration of the manipulator that accounts for said force.

As a preliminary experiment to assess the predictive capabilities of the model, we observe the deflection of the robot when a set of known loads are applied at the end-effector position. First, an initial configuration of the manipulator without loading is selected and the position of the end-effector is measured with respect to a reference frame; then, the load is applied and the new position of the end-effector is recovered. The loads are applied statically, so only the stationary state of the manipulator is observed. The same load is applied to the model of the manipulator using the same initial pose and the resulting end-effector position is also recovered. The model of the sensors presented in 3.4.3 is used to attain the initial configuration of the robot in the simulation. This configuration is measured by the string potentiometers.

A vector that connects the initial and final end-effector positions represents the deflection caused by the load. In order to assess the repeatability of the measurements, the loading sequence described is performed 40 times for each loading value and the average value is then used for the model validation. The results are presented in Fig. 3.12

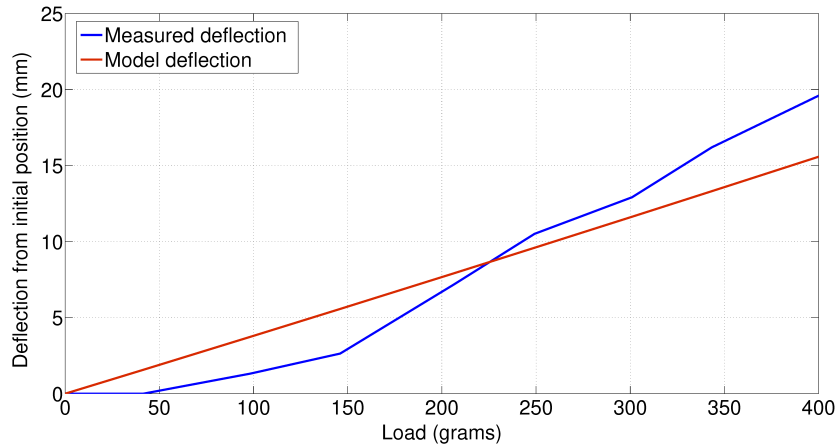


Figure 3.12: Comparison between measured and predicted deflections caused by external loading on the CBHA manipulator

The results illustrate the compliance to loading profile of the manipulator, with a maximum deflection error of 4.107mm with an average error of 2.104mm between the measured values (blue line) and the prediction of the model (red line). The results also show that the CBHA presents strain hardening/necking stages of plastic behavior which corresponds to the compliance of the plastic material used to make the manipulator. This non-linear behavior is not considered in our model and therefore, the model prediction is accurate only for a small region of the profile. This behavior can be accounted for, in order to improve the predictive capabilities of the model, by implementing two constitutive laws with a switching condition, depending on the region of compliance in which the manipulator is operating. However, this would modify significantly the way the inverse FEM is formulated.

### 3.7 Kinematic models

In this section, the methodology developed in this chapter is applied to obtain the forward and inverse kinematic relationships of the CBHA based on the FEM simulation. An experiment is performed to assess the accuracy of the forward model using the real robot. The results are then compared to those obtained using 2 different modeling approaches. The inverse kinematic model, derived from the inverse FEM simulation of the robot, is also shown and validated experimentally.

### 3.7.1 Forward kinematic models

The forward kinematic model of a soft manipulator deals with the problem of finding the end-effector position, given a defined configuration of the manipulator. For a rigid manipulator, this configuration is simply the set of variables associated with the joints of the robot. In contrast with the rigid robots, the variables that express the configuration of a soft manipulator change with respect to the morphology of the robot and its type of actuation, and therefore, cannot be obtained in a straight forward manner. The FEM-based methodology presented provides a formulation to obtain the kinematic relation between the end-effector and the configuration of the manipulator.

Given the intrinsic nature of the CBHA, the configuration of the robot is represented by the lengths of the sensor strings, that correspond to a position of the end-effector, as illustrated in Fig. 3.13.

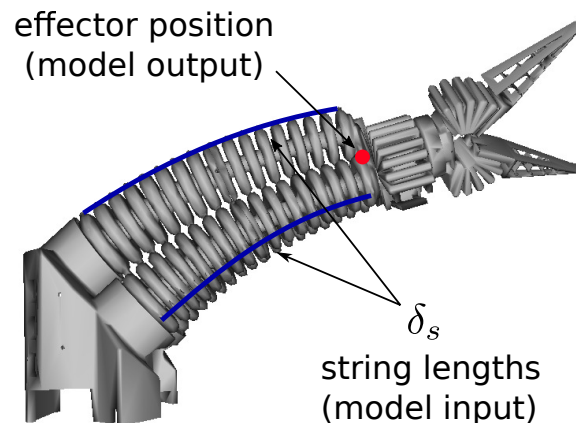


Figure 3.13: Illustration of the forward kinematic model (inputs and output)

The input to our model becomes the configuration of the robot, and the output will be the actuation required to achieve said configuration. When the actuation is computed and applied to the FEM model, the position of the end-effector that corresponds to the given configuration is retrieved. Eq. 3.18, which is reproduced here for clarification, is used to relate the configuration of the manipulator to the actuation space. The lengths are recovered by the string potentiometers of the CBHA.

$$\boldsymbol{\delta}_s = \underbrace{[\mathbf{H}_s \mathbf{K}^{-1} \mathbf{H}_a^T]}_{\mathbf{W}_{sa}} \boldsymbol{\lambda}_a + \boldsymbol{\delta}_s^{\text{free}} \quad (3.28)$$

In this approach, no geometrical assumptions are needed. Each part of the manipulator is modeled in detail as described in 3.6.5. Once the constraints have been incorporated in the model, the convex optimization finds each actuator contribution required to achieve the desired sensor lengths. The position of the end-effector is not represented explicitly in Eq. 3.18. Instead, the corresponding position is recovered after the position of the nodes is updated.

In the following, we briefly present 2 different forward models of the CBHA previously introduced in the literature, in order to compare the results obtained experimentally.

### 3.7.1.1 Constant curvature model

This approach follows the development shown in 2.3.2.1. It was tailored to the case of the CBHA in [Escande 2011] and experimentally validated in [Escande 2012]. In this approach, the evolution from end-to-end of a section  $i$  is described, in terms of backbone parameters, by 2 coupled rotations and one translation, given by the homogeneous transformation

$${}^i_j\mathbf{T} = \begin{bmatrix} \cos^2 \phi_i \cos \theta_i + \sin^2 \phi_i & \cos \phi_i \sin \phi_i (\cos \theta_i - 1) & \cos \phi_i \sin \theta_i & x_i \\ \cos \phi_i \sin \phi_i (\cos \theta_i - 1) & \sin^2 \phi_i \cos \theta_i + \cos^2 \phi_i & \sin \phi_i \sin \theta_i & y_i \\ -\cos \phi_i \sin \theta_i & -\sin \phi_i \sin \theta_i & \cos \theta_i & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.29)$$

The end-effector cartesian coordinates of the bending section  $i$  are given by  $(x_i, y_i, z_i)$ , where

$$x_i = r_i \cos \phi_i (1 - \cos \theta_i) \quad (3.30)$$

$$y_i = r_i \sin \phi_i (1 - \cos \theta_i) \quad (3.31)$$

$$z_i = r_i \sin \phi_i \quad (3.32)$$

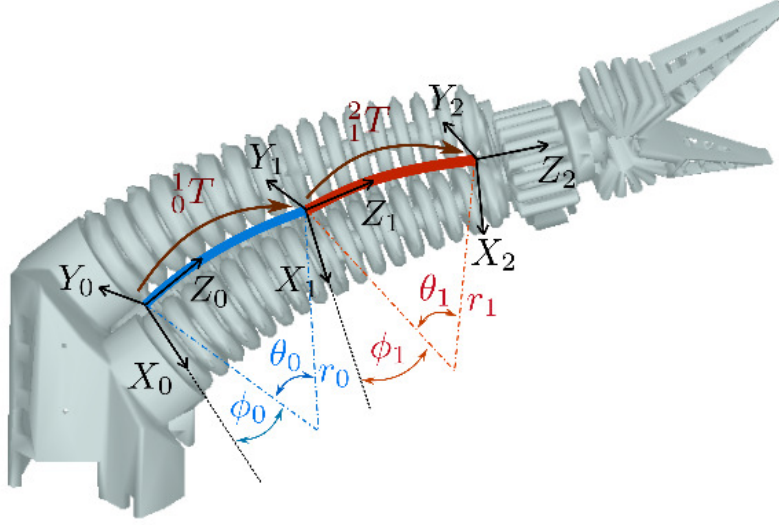


Figure 3.14: Parametrization of the Constant curvature model of the CBHA

The backbone variables, namely the rotation angles  $\phi_i$  and  $\theta_i$  and the curvature radius  $r_i$ , are illustrated in Fig. 3.14.

In order to have the kinematic model relating the configuration of the manipulator to the end-effector, the backbone variables need to be expressed in terms of the actuator lengths. These relationships are

$$\phi_i = \tan^{-1} \frac{\sqrt{3}(l_3 - l_1)}{2l_1 - l_2 - l_3} \quad (3.33)$$

$$\theta_i = \frac{D_i}{3d_i} \quad (3.34)$$

$$r_i = \frac{(l_1 + l_2 + l_3)d_i}{D_i} \quad (3.35)$$

where

$$D_i = 2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1l_2 - l_1l_3 - l_2l_3} \quad (3.36)$$

The lengths of each actuator in section  $i$  are represented by  $l_1$ ,  $l_2$  and  $l_3$ . The parameter  $d_i$  represents the diameter of section  $i$ . In this model, each section is considered to be a cylinder with constant radius.

### 3.7.1.2 Hybrid model

In the Hybrid approach to the modeling of the CBHA [Lakhal 2016], the structure of the robot is considered as 17 vertebrae serially connected. Between each pair of vertebrae, an inter-vertebra section is modeled as a 3UPS-1UP joint. The behavior of a sub-structure composed by 2 vertebrae and the inter-vertebra is represented by a parallel robot with 3 DoF, as depicted in Fig. 3.15.

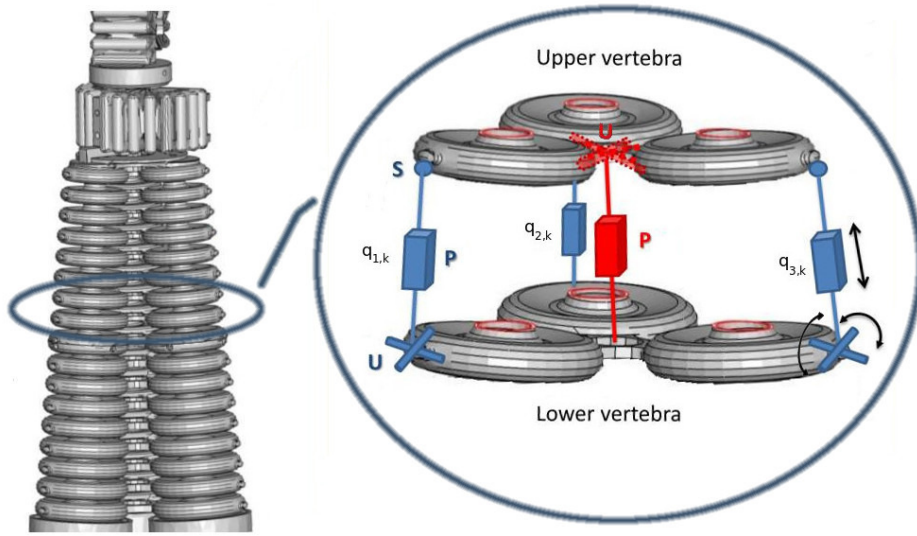


Figure 3.15: Sub-structure of the CBHA modeled as a parallel robot

The position and orientation of the upper vertebra  $k + 1$  with respect to the lower vertebra  $k$  is given by the transformation matrix

$${}^k_{k+1}\mathbf{T} = \begin{bmatrix} \cos \theta_k & \sin \theta_k \sin \Psi_k & \sin \theta_k \cos \Psi_k & 0 \\ 0 & \cos \Psi_k & -\sin \Psi_k & 0 \\ -\sin \theta_k & \sin \Psi_k \cos \theta_k & \cos \theta_k \cos \Psi_k & z_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.37)$$

where the angles  $\theta_k$  and  $\Psi_k$  represent the pitch and roll angles of the vertebra, respectively. The jaw of vertebra  $k + 1$  with respect to the vertebra  $k$  is constrained by a passive prismatic joint. The position of the center of vertebra  $k + 1$  is given by the longitudinal translation  $z_k$ . The prismatic variable  $q_{n,k}$ , shown in Fig. 3.15, represents the length of the inter-vertebra, which is a percentage of the total length of the

pneumatic actuator. The value of  $q_{n,k}$  is computed by considering the minimum and maximum elongation of each inter-vertebra.

### 3.7.1.3 Forward kinematics experimental validation

In order to validate the position tracking capabilities of the model, a trajectory inside the task space of the robot is defined. The trajectory is composed of a set of *via points* for which the position can be measured in the real robot. The cartesian position of the end-effector is measured using a stereo-vision system composed by a set of 6 infrared cameras. For each position, the corresponding set of actuator lengths was recorded using the string potentiometers. The set of lengths was then used as an input for the forward kinematic model in the simulation. After imposing the sensor constraints, the configuration of the robot is updated and the position of the end-effector is then recovered. In this experiment we assume that there is no payload being carried by the end-effector. Fig. 3.16 shows the error between the measured trajectory and the trajectory estimated by the FEM simulation. Fig. 3.17 and 3.18 show the estimated trajectory for the constant curvature and hybrid approaches, respectively.

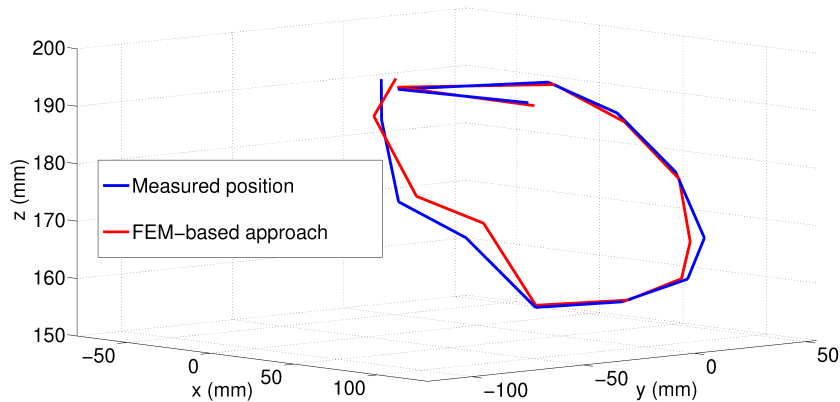


Figure 3.16: Estimated trajectory of the FEM-based approach compared to the measured trajectory

The euclidean error comparison between the 3 approaches is summarized in Table 3.1, where RMSD stands for the root-mean-square deviation, ME is the maximum error, CC is the constant curvature model, HA is the hybrid model and FEM is the finite element-based model.



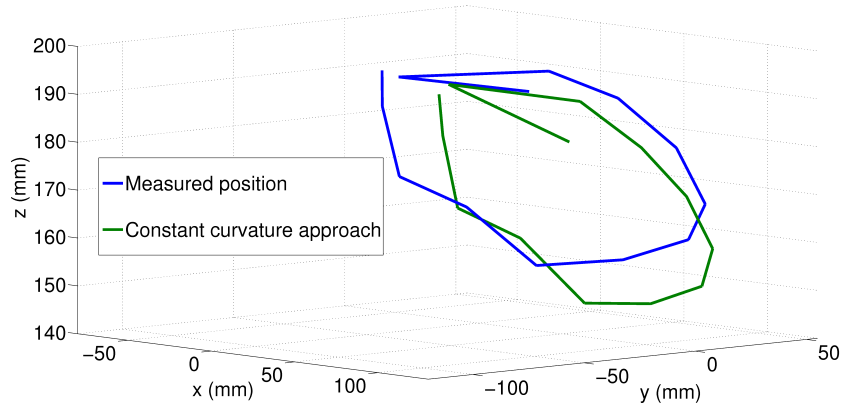


Figure 3.17: Estimated trajectory of the Constant curvature approach compared to the measured trajectory

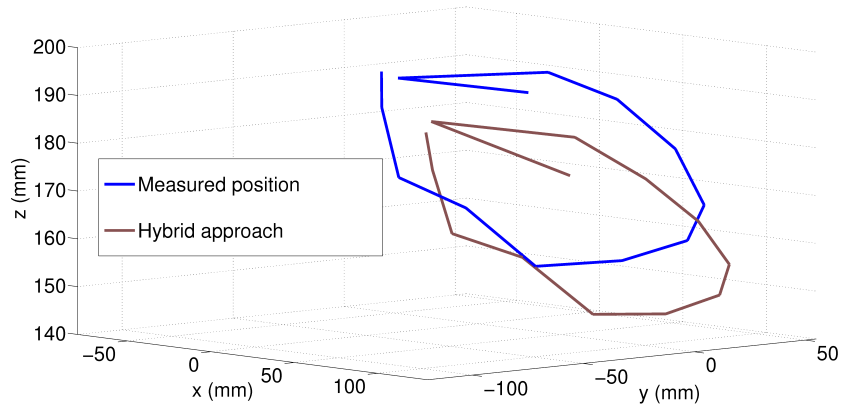


Figure 3.18: Estimated trajectory of the Hybrid approach compared to the measured trajectory

Table 3.1: Error metrics for the 3 modeling approaches for the CBHA in free load configuration

	CC model	HA model	FEM model
RMSD(mm)	12.87	17.09	4.66
ME(mm)	16.35	22.4	7.71

From Fig. 3.18, we can observe that the Hybrid approach is the least accurate. This can result from the fact that one cannot precisely measure the displacement of each vertebra (only 6 string potentiometers are available). Moreover, this model was initially developed to be able to inverse it, more than for the attainable precision of the forward kinematic model. The constant curvature model has a standard deviation of 12.87mm, which represents 6.43% of the total length of the manipulator; however, for longer manipulators with bigger variations in their diameter with respect to their length, the error increases considerably. Moreover, as soon as a deformation in the manipulator is induced by an external force, both of these approaches lose relevance.

### 3.7.2 Inverse kinematic model

Inverse kinematics deal with the problem of finding the correct configuration of the robot, given a desired position of the end-effector. This configuration, as for the forward kinematic model, is represented by the set of sensor lengths that describe the state of the manipulator. Of course, the description of the robot can be given in the actuator space directly [Bosman 2015], using in this case the relationship between the end-effector position and the actuators contribution. This provides a pressure-to-position model that requires a precise control over the actuation (pressure inside the pneumatic actuators). Due to the embedded control of the CBHA, the precise control cannot be attained however.

Instead, the relationship given by Eq. 3.16 is used to find the contribution of the actuators required to achieve the desired end-effector position. After these forces are computed and applied to the model in the simulation, the reading from the sensor model given in Eq. 3.13 will be taken as the solution of the inverse kinematic problem. Fig. 3.19 illustrates the input and output relationships in this model.

To validate the method, a set of 50 end-effector positions are selected inside the task space of the robot and the corresponding set of lengths for each position is recorded by the sensors of the robot. The same set of positions is used as inputs for the inverse model simulation, and the resultant length of each sensor is computed after the model is updated. This study is summarized in Table 3.2, where  $l_1, \dots, l_6$  represent the lengths of the sensors and their values are in mm,  $\mu$  represents the mean

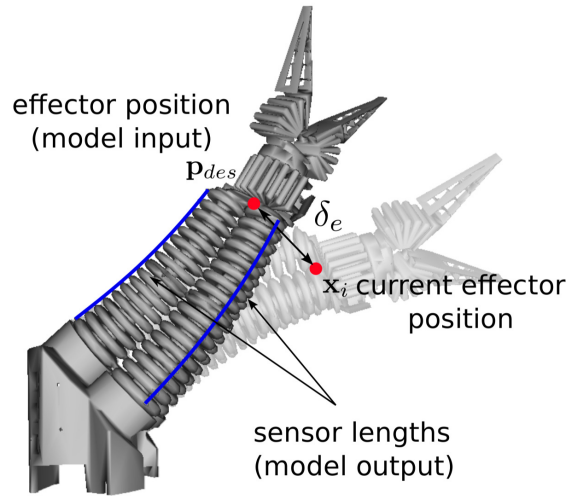


Figure 3.19: Illustration of the inverse kinematic model (input and output)

error and  $\sigma$  is the standard deviation. The results are presented in Fig. 3.20.

Table 3.2: Error between measured and estimated lengths for the CBHA

$l(\text{mm})$	$l_1$	$l_2$	$l_3$	$l_4$	$l_5$	$l_6$
$\mu(\text{mm})$	3.2	2.43	3.86	4.08	3.6	3.69
$\sigma(\text{mm})$	1.55	1.76	2.05	2.12	2.56	2.06

The results show a mean error between 2.43mm and 4.08mm across all lengths, which represents between 1.21% and 2.04% of the total length of the manipulator. Again, the set of actuator contributions (in this case the pressures applied to the cavities) obtained from the optimization process can be used as input for the real robot to obtain a feed-forward control signal, as will be explained in the next chapter.

### 3.8 Conclusion of the chapter

The results of the experimental comparison show that the FEM-based forward kinematic model has greater accuracy in terms of position tracking, compared to the geometric models. The model of the sensors in the

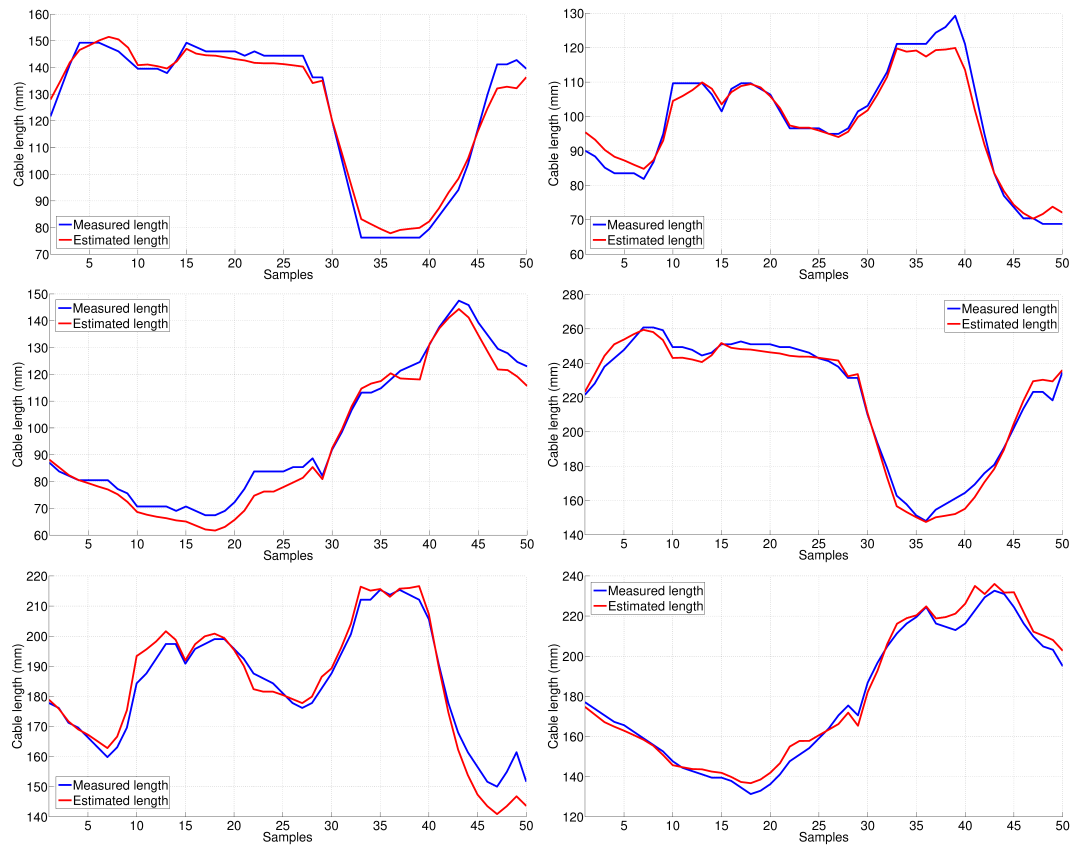


Figure 3.20: Comparison of the 6 measured and estimated lengths for a predefined set of end-effector positions for the CBHA

manipulator allows for a representation of the kinematics in the configuration space that is less sensitive to model uncertainties, compared to a description in the actuator space. Nevertheless, given an accurate model of the constitutive law of the material, the method can provide a description in the actuator space that can be used to control the robot directly. Unlike other methods, the kinematic model involving the actuator space can be derived without any additional transformations.

The FEM model has a few limitations in its development, however. These limitations represent the main source of error in the experimental results: for the moment, the constitutive law used to model the material of the trunk is only an approximation, as shown in 3.6.5. Moreover, non-linear effects like the plasticity of the material are not yet implemented in the model.

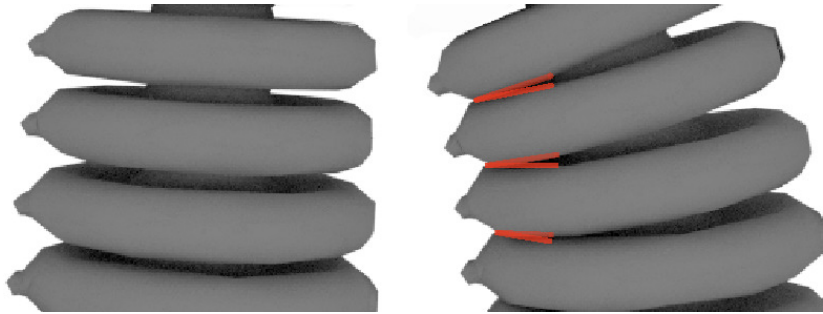


Figure 3.21: Collision of the outer walls of the cavities when bent at maximum angles

Another source of error comes from the geometry of the trunk itself. When the trunk is bent at a maximum angle, the outer walls of the pneumatic cavities collide with each other, as shown in Fig. 3.21. The consideration of these collisions is not yet implemented in the simulation; however, the formulation of the contact forces is already developed and implemented in SOFA.

Additionally, the extension of the CBHA is limited by a steel cable that runs through its backbone. This constraint is not considered in the model and so, the extension of the robot in simulation is constrained only by the stiffness of the structure and load mass. Thus, this modeling uncertainty can be compensated by adding a closed-loop controller, which is introduced in the following chapter.



# Closed loop control of soft, continuum manipulators

---

## Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>67</b>
<b>4.2</b>	<b>Feed-forward control of continuum manipulators</b>	<b>68</b>
<b>4.3</b>	<b>Closed-loop control of continuum manipulators</b>	<b>71</b>
4.3.1	Closed-loop control law design	72
4.3.2	Robustness analysis	76
<b>4.4</b>	<b>Conclusions of the chapter</b>	<b>80</b>

---

## 4.1 Introduction

In the previous chapter, the kinematics of continuum manipulators based on the inverse FEM simulation were presented and the limitations of the model in terms of model uncertainties were discussed. The aim now, is to compensate for the limitations of the kinematic modeling given in the previous chapter in terms of accuracy of the position tracking and rapidity by using a kinematic controller.

Controllers for soft manipulators have been investigated in the past with the intention of rejecting non-linear behaviours and model uncertainties that result from the complex dynamics of the manipulators (see 2.3.3). Control based on energy formulations [Ivanescu 2003], model-less approaches [Yip 2014] and feedback controllers [Penning 2011] [Penning 2012] have been proposed before with the intention of achieving accurate positioning of the manipulators in presence of non-modeled dynamics. In this chapter we propose a feedback control scheme for soft, continuum manipulators. The control is based on the Jacobian estimation of the robot, obtained from the FEM model simulation and

the implementation of a state estimator. We have tested the closed-loop controller on the CBHA, but the same kind of control can be used with any soft, continuum manipulator as long as it can be modelled using the FEM-based method explained in the previous chapter.

In the next section, the feed-forward control based on the inverse kinematic model of the CBHA is presented in order to highlight the need for closed-loop controllers.

## 4.2 Feed-forward control of continuum manipulators

Based on the inverse kinematic model of the CBHA presented in the previous chapter, it is possible to use a real-time simulation of the robot to directly control it. A feed-forward controller which, in the case of the CBHA, computes the pressures required to reach a desired position for the end-effector is implemented using the relationship given in Eq. 3.16 and 3.17 (see Fig. 4.1).

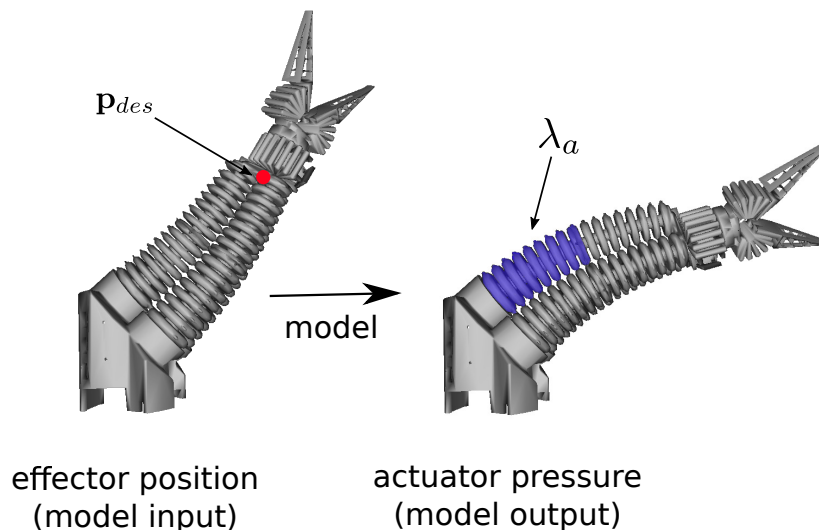


Figure 4.1: Illustration of the feedforward controller based on the pressure-to-position model

We call this type of relationship a *pressure-to-position* model, since it relates these two spaces. In order to describe accurately the implemen-



tation of this type of controller, let us consider the discrete form of Eq. 3.16. Assuming the external forces to be constant, we have

$$\boldsymbol{\delta}_{e,k+1} = \boldsymbol{\delta}_{e,k} + \mathbf{W}_{ea}(\mathbf{x}_k)\Delta\boldsymbol{\lambda}_{a,k+1} \quad (4.1)$$

$$\boldsymbol{\delta}_{a,k+1} = \boldsymbol{\delta}_{a,k} + \mathbf{W}_{aa}(\mathbf{x}_k)\Delta\boldsymbol{\lambda}_{a,k+1} \quad (4.2)$$

where  $\boldsymbol{\delta}_{e,k+1}$  is the current end-effector position and  $\Delta\boldsymbol{\lambda}_{a,k+1}$  is the increment of the actuators force contribution at time  $k + 1$ ,  $\mathbf{x}_k$  is the position of the nodes at time  $k$ , also referred as the state of the robot and  $\mathbf{W}_{ea}$  is the coupling between end-effector and actuators. In order to describe the system in a language that is more familiar to robotic control theory, in the following, the matrix  $\mathbf{W}_{ea}$  will be recalled as  $\mathbf{J}_{ea}$ , which is the Jacobian matrix between the inputs of the system (the pressure in the cavities) and the output error. In this way, Eq. 4.1 can be re-written as

$$\boldsymbol{\delta}_{e,k+1} = \boldsymbol{\delta}_{e,k} + \mathbf{J}_{ea}(\mathbf{x}_k)\Delta\boldsymbol{\lambda}_{a,k+1} \quad (4.3)$$

The error is defined as

$$e_{k+1} = \boldsymbol{\delta}_{e,k+1} - \boldsymbol{\delta}_{e,k+1}^d \quad (4.4)$$

where  $\boldsymbol{\delta}_{e,k+1}^d$  is the desired position. The control signal sent to the robot (pressure commands) computed at each simulation time is

$$\boldsymbol{\lambda}_{a,k+1} = \boldsymbol{\lambda}_{a,k} + \Delta\boldsymbol{\lambda}_{a,k+1} \quad (4.5)$$

The configuration of the robot at time  $k + 1$  is updated by

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{K}_k^{-1}\mathbf{H}_a^T\Delta\boldsymbol{\lambda}_{a,k+1} \quad (4.6)$$

where the matrix  $\mathbf{H}_a$  results from the imposition of the model constraints as Lagrange multipliers, as explained in 3.4. In this controller implementation, the estimation of the Jacobian matrix for the real robot performed by the simulation is inaccurate. This inaccuracy is due to the fact that a model will never match perfectly the real system because of unmodelled phenomenons, parameter uncertainties and disturbances. This results in a significant disparity between the Jacobian of the robot and the Jacobian estimated by the simulation.

To show the performance of this type of control, a semicircular trajectory formed by via points inside the task space of the robot is computed. Each point of the trajectory is introduced in the simulation sequentially as a desired position for the end-effector and the actuation computed by the simulation is then used as set point for the pressure regulator of the robot. The results from this test are shown in Fig. 4.2.

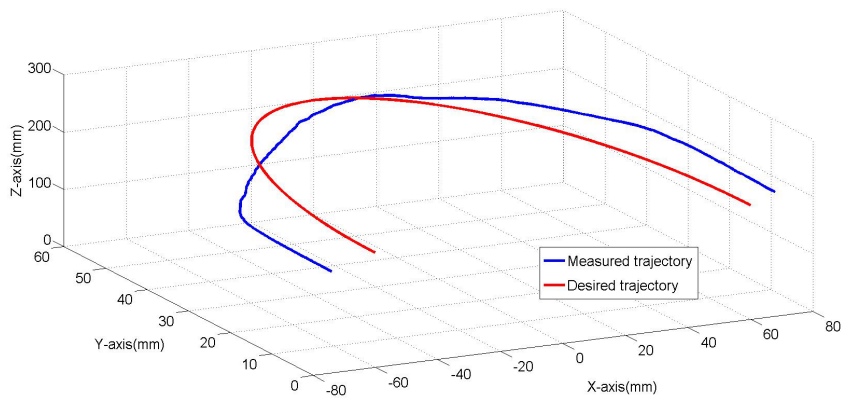


Figure 4.2: Comparison between desired and measured trajectories using the simulation of the CBHA as a feed-forward controller

The results show a maximum error of 23mm, which represents the 11% of the total length of the manipulator. The pressure-to-position model used in this controller is highly sensible to model disparities, since the pressure constraints are no longer modelled as Dirichlet boundary conditions [Reddy 2013] in the FEM simulation. Moreover, in the case of the CBHA, because of the lack of sensors, we cannot be certain that the pressures computed by the simulation are the ones applied to the cavities.

One way to correct this type of model is the use of a closed-loop control strategy. To this end, some questions arise in the development of the controller: First, what type of controller could be used? and more importantly, Is the closed-loop system robustly stable?

## 4.3 Closed-loop control of continuum manipulators

In the development of the closed-loop controller for the CBHA, a fundamental change in the nature of the model used in simulation is performed. As explained previously, the pressure-to-position model used in the feed-forward control required an accurate control of the pressures applied to the cavities. Instead, the pressure-to-length model given in 3.7.2 is used. In this way, the inputs of our model become the lengths of the sensors that correspond to a desired position. Eq. 4.1 is replaced by

$$\delta_{s,k+1} = \delta_{s,k} + \mathbf{J}_{sa}(\mathbf{x}_k)\Delta\lambda_{a,k+1} \quad (4.7)$$

where  $\mathbf{J}_{sa}$  is the Jacobian matrix between sensors and actuators.

Instead of imposing a desired end-effector position  $\delta_e^d$ , a reference computation is performed in order to obtain the sensor lengths  $\delta_s^d$  required to achieve said position (Fig. 4.3).

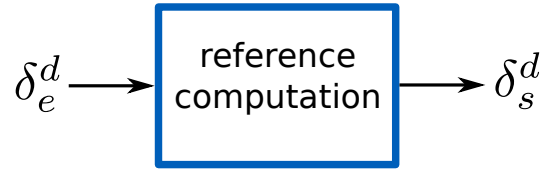


Figure 4.3: Reference computation that is performed before the feedback system.

The reference computation provides the desired motion in terms of desired sensor lengths. The desired trajectory can also be set by the user by imposing the motion to the trunk and recording the sensor lengths. When these desired sensor lengths are provided, we can propose a closed-loop approach, as shown in Fig. 4.4.

In Fig. 4.4 the blue blocks represent the computations performed by simulation. The output of the reference computation (Fig. 4.3) is the input of the closed-loop. Two simulations executing simultaneously are implemented in the closed-loop system: One main simulation that computes the Inverse kinematic model and a second simulation that acts as a state estimator for the system. The state estimator is the Forward kinematic model simulation of the robot that computes an estimated

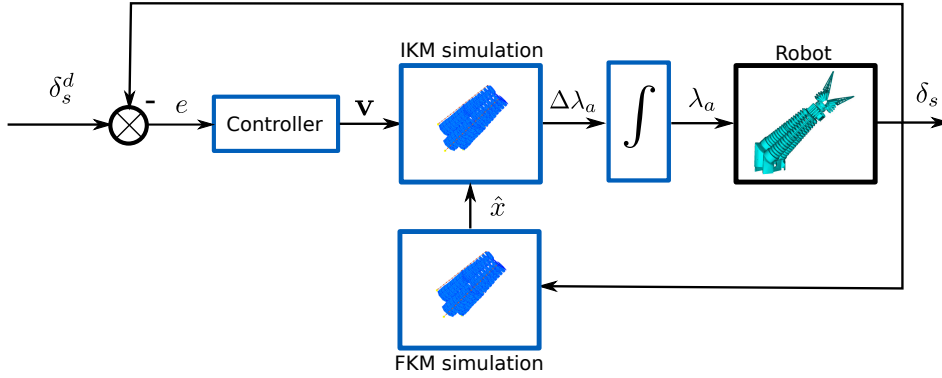


Figure 4.4: Closed-loop control of the CBHA based on IKM and FKM simultaneous simulations and the controller

configuration for the robot based on the lengths of the actuators. This configuration is used to update the state of the Inverse kinematic model at each simulation step. In this way, we make sure that the configurations of both simulation model and the manipulator are the same before the estimation of the Jacobian is computed.

### 4.3.1 Closed-loop control law design

The tracking error  $e_k$  in the closed-loop system is computed as

$$e_k = \delta_{s,k} - \delta_{s,k}^d \quad (4.8)$$

with  $\delta_{s,k}^d$  represents the desired lengths of the sensors and  $\delta_{s,k}$  represents the current lengths in the robot. We define the control vector  $\mathbf{v}_k$  as

$$\mathbf{v}_k = \hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k) \mathbf{r}_k \quad (4.9)$$

where  $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$  is the estimated Jacobian matrix between the sensors and actuators and  $\mathbf{r}_k = \Delta \boldsymbol{\lambda}_{a,k+1}$ . Using Eq. 4.9, the kinematic model can be rewritten as

$$\boldsymbol{\delta}_{s,k+1} = \boldsymbol{\delta}_{s,k} + \mathbf{v}_k \quad (4.10)$$

The control law is based on proportional integrative strategy, therefore, the control vector  $\mathbf{v}_k$  is designed in the sensor space as

$$\mathbf{v}_k = -k_p e_k - k_i h_k \quad (4.11)$$

with  $k_p$  and  $k_i$  being the proportional and integrative gains of the controller, respectively. The integrative term  $h$  at time  $k + 1$  is computed as

$$h_{k+1} = h_k + e_k \quad (4.12)$$

Then, the control allocation based on a Quadratic Programming (QP) formulation [Johansen 2013] is employed to find a unique solution to

$$\mathbf{r}_k = \hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)\mathbf{v}_k \quad (4.13)$$

where  $\hat{\mathbf{J}}_{sa}^+$  is the pseudo-inverse of the estimated Jacobian. In practice, as  $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$  may not be fully invertible, we introduce a variable  $O$  defined as

$$O = \hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)\mathbf{r}_k - \mathbf{v}_k \quad (4.14)$$

Using  $O$ , the QP problem formulation (3.5.3) becomes

$$\min_{\mathbf{v}_k} (O^T O) \quad (4.15)$$

the resulting  $\mathbf{r}_k$  will be the best possible inversion of Eq. 4.9 in the least square sense. In addition, the QP formulation allows to define constraints like actuator saturation or positive direction of actuation. Using Eq. 4.11 in Eq. 4.13,  $\mathbf{r}_k$  is rewritten as

$$\mathbf{r}_k = -\hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)(k_p e_k + k_i h_k) \quad (4.16)$$

Using Eq. 4.16 in Eq. 4.7, the closed-loop system is defined as

$$e_{k+1} = e_k + \mathbf{J}_{sa}(\mathbf{x}_k)\mathbf{r}_k \quad (4.17)$$

which in the ideal case in which  $\hat{\mathbf{J}}_{sa}$  is invertible, can be written as

$$e_{k+1} = e_k + \mathbf{v}_k \quad (4.18)$$

The system of Eq. 4.18 is a first order discrete model that can be controlled with any standard controller. We choose the control strategy to be based on proportional-integrative control law because we want to

improve the convergence rate and remove any steady state error (in the sensor space at least).

The gains of the controller are tuned by extensive testing using a simulated version of the closed-loop; we use a direct FEM simulation of the manipulator that takes any external force as input and computes the displacement field based on these forces. No computation of the inverse of the stiffness matrix  $\mathbf{K}^{-1}$  is required. This direct simulation is put into the place of the real robot in Fig. 4.4. The control input computed by the simulation is applied to the direct simulation and the performance of the robot is observed.

In order to communicate the different simulations, a couple of client-server scripts are written in Python based on the sockets provided by ZeroMQ communication protocol [Hintjens 2013], that allows for a fast communication between the simulations with an average latency of  $75\mu s$ .

After testing, the selected gain values are  $k_p = 0.14$  and  $k_i = 0.0003$  as a compromise between the rise time of the signal and its overshoot. Fig 4.5 shows the lengths of the simulated robot and the real robot, both in closed-loop, given a pre-computed set points corresponding to an end-effector position inside the task space of the robot. The position is chosen so that the actuators are far from their saturation points. The model simulation and the real robot have different initial conditions. After 1000 simulation steps, the set points are changed in both the simulation and the real robot.

Some noticeable features in Fig 4.5 can be observed. First, the results show that both, the simulation of the robot and the robot itself have the same settling time  $t_s \approx 400$  simulation steps. We can also see that the curve that represents the measured value of the lengths in the robot *jumps* between two values. This behavior is a consequence of the poor resolution of the string potentiometers. While the precision of the sensors cannot be improved, the signal of the sensors can be filtered by using a numerical implementation of an Infinite Impulse Response filter [Rabiner 1975]. However, after implementing the filter, a considerable delay in the measuring signal was noticed. This delay hampers considerably the performance of the closed-loop system and, therefore, we opted to not implement it in the final version of the controller.

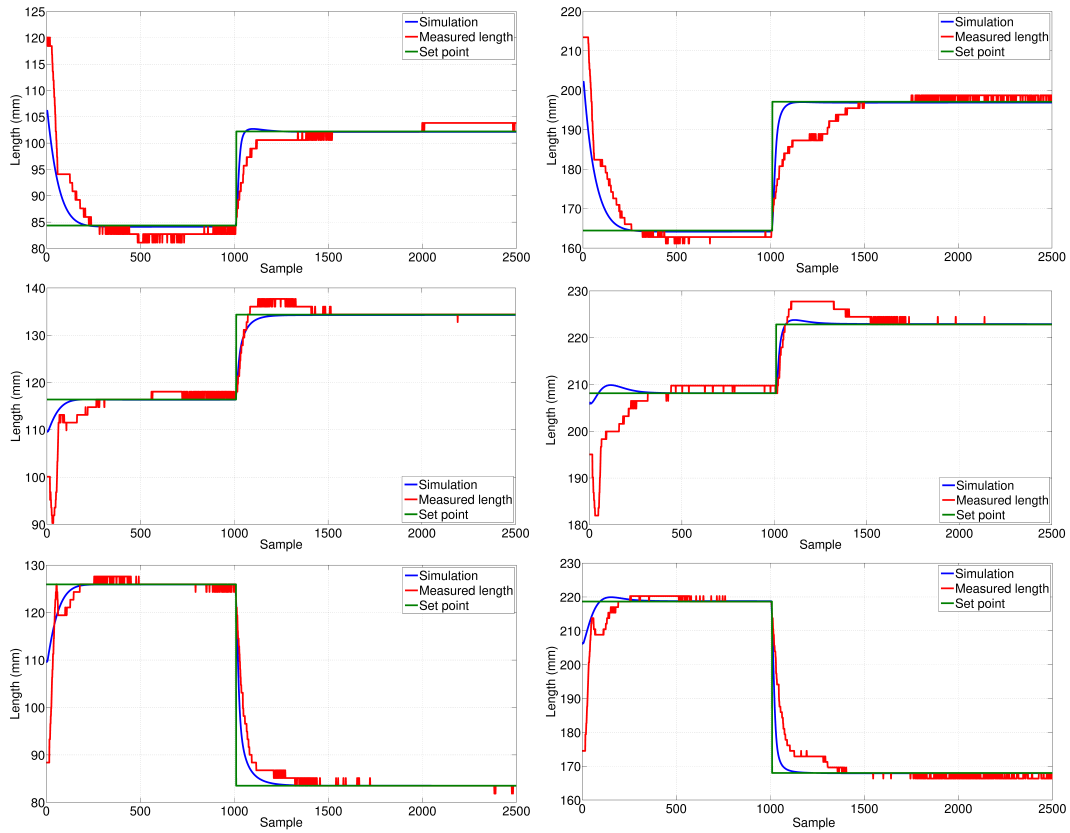


Figure 4.5: Comparison of real and estimated actuator lengths of the CBHA. A second set point is applied to the system after 1000 simulation steps in order to observe the performance of the controller. The time step is 0.1s for the experiment.

Fig 4.5 also shows a different response in the transitory stage of the curve of the measured lengths. This response can be attributed to different factors; first, there is the time required to compute the configuration of the manipulator from the measured sensor lengths; second, there is a time delay for the desired pressure to be applied to the robot, and finally, the plasticity of the material from which the manipulator is built (polyamide-nylon), that introduces a dampening effect at large deformations, which is not represented in our FEM model. There is hysteresis in the behavior of the pneumatic cavities. On the other hand, the pneumatic valves that control the pressure inside the actuators have a small dead zone, so, when the manipulator starts its motion from a zero-pressure condition, very small increments in the pressure do not produce any motion until this dead zone is surpassed, which is not considered in the FEM model.

A second experiment was performed with the real robot in the loop. In this experiment an external unknown force was applied to the manipulator in order to see the uncertainty rejection capabilities of the controller. Fig. 4.6 shows the results of this experiment.

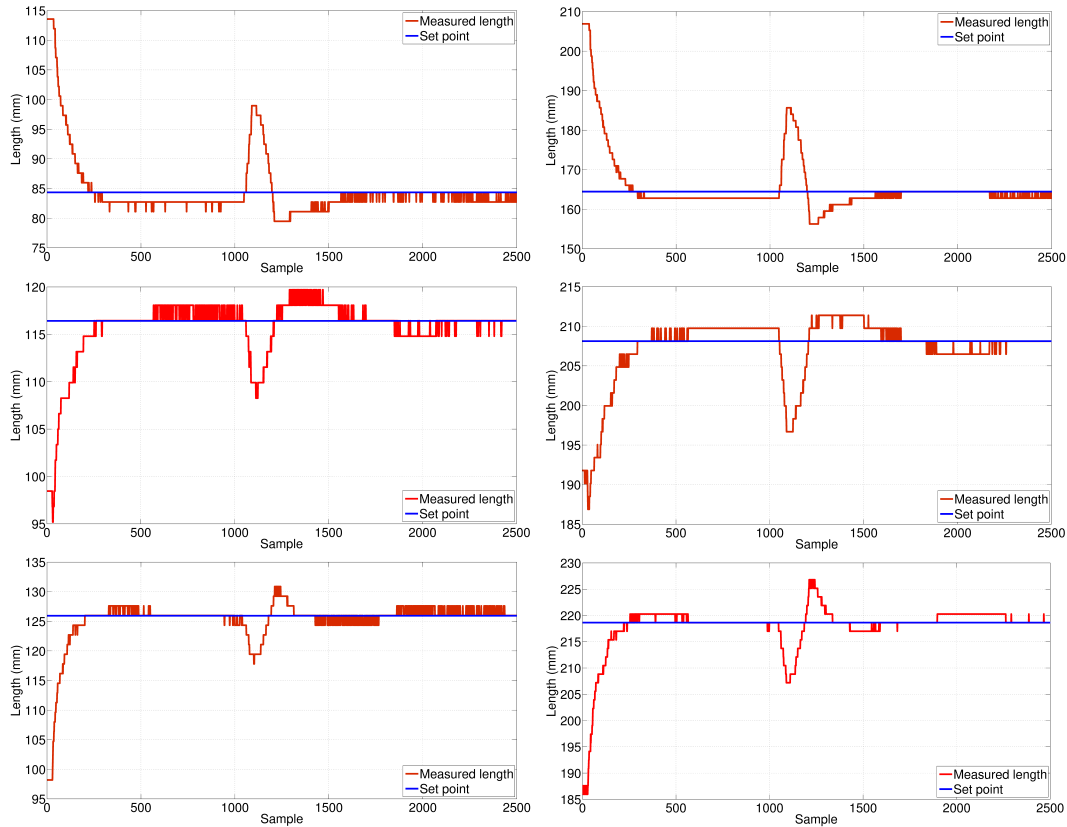


Figure 4.6: Measured lengths of the CBHA in closed-loop. An external force is applied to the manipulator after 1050 time steps. The time step is 0.1s for the experiment.

Fig. 4.6 shows the controller disturbance rejection capabilities. After the external force is applied, the manipulator is able to return to the desired configuration.

### 4.3.2 Robustness analysis

Because of modeling uncertainties, the estimated Jacobian matrix  $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$  is, in general, different from the Jacobian of the robot  $\mathbf{J}_{sa}(\mathbf{x}_k)$ . We introduce the vector  $\omega_k$  that represents the disparities between the



real Jacobian and the estimated Jacobian. We call this vector the *inversion error* and is defined as

$$\omega_k = [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k)\hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)]\mathbf{v}_k \quad (4.19)$$

Then, the closed-loop system is re-written as

$$e_{k+1} = e_k + \mathbf{v}_k + \omega_k = e_k - k_p e_k - k_i h_k + \omega_k \quad (4.20)$$

The disturbed closed-loop system is

$$\begin{bmatrix} e_{k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - k_p \mathbf{I} & -k_i \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} e_k \\ h_k \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \omega_k \quad (4.21)$$

It can be disturbing that we end up with such a simple linear system. We emphasize to the reader that the non-linearities are taken into account by the two simulation blocks (FKM and IKM in Fig. 4.4) in the closed-loop control. In Eq. 4.21, we are writing the system in terms of  $e_k$  and  $h_k$  and if the model was perfect, the system would be trivial. However, we can have modeling errors, that is why, in the following, we will prove that the control is robust to these modeling uncertainties  $\omega_k$ .

To simplify the notation of the problem, we define the following vectors

$$\mathbf{X}_{k+1} = \begin{bmatrix} e_{k+1} \\ h_{k+1} \end{bmatrix}, \quad \mathbf{X}_k = \begin{bmatrix} e_k \\ h_k \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \quad \text{and} \quad \mathbf{F} = [k_p \quad k_i] \quad (4.22)$$

Also

$$\begin{bmatrix} \mathbf{I} - k_p \mathbf{I} & -k_i \mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} = \mathbf{A} - \mathbf{BF} \quad (4.23)$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \quad \text{and} \quad \mathbf{B} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \quad (4.24)$$

Using this notation, matrix  $\omega_k$  is written as

$$\omega_k = [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k)\hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)]\mathbf{F}\mathbf{X}_k \quad (4.25)$$

## 78 Chapter 4. Closed loop control of soft, continuum manipulators

If the Jacobian is bounded and invertible, then there always exist a parameter  $\gamma$  such that

$$\omega_k^T \omega_k = \mathbf{X}_k^T \mathbf{F}^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k) \hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)]^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k) \hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)] \mathbf{F} \mathbf{X}_k \leq \gamma^2 \mathbf{X}_k^T \mathbf{F}^T \mathbf{F} \mathbf{X}_k \quad (4.26)$$

with

$$[\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k) \hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)]^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x}_k) \hat{\mathbf{J}}_{sa}^+(\hat{\mathbf{x}}_k)] \leq \gamma^2 \mathbf{I} \quad (4.27)$$

For the proof of stability, we use Lyapunov's second method of stability [Lyapunov 1992]. We define the Lyapunov candidate function as

$$V = \mathbf{X}_k^T \mathbf{P} \mathbf{X}_k \quad (4.28)$$

where  $\mathbf{P}$  is the Lyapunov matrix with the properties

$$\mathbf{P}^T = \mathbf{P} > 0 \quad (4.29)$$

From Eq. 4.28 and the notation given in Eq. 4.22, the variation of the Lyapunov function is defined as

$$\Delta V = \mathbf{X}_{k+1}^T \mathbf{P} \mathbf{X}_{k+1} - \mathbf{X}_k^T \mathbf{P} \mathbf{X}_k \quad (4.30)$$

Using Eq. 4.24, Eq. 4.30 is re-defined as

$$\Delta V = ((\mathbf{A} - \mathbf{B}\mathbf{F})\mathbf{X}_k + \mathbf{D}\omega_k)^T \mathbf{P} ((\mathbf{A} - \mathbf{B}\mathbf{F})\mathbf{X}_k + \mathbf{D}\omega_k) - \mathbf{X}_k^T \mathbf{P} \mathbf{X}_k \quad (4.31)$$

By making

$$\mathbf{A} - \mathbf{B}\mathbf{F} = \mathbf{C} \quad (4.32)$$

Eq. 4.31 is written as

$$\begin{aligned} \Delta V &= (\mathbf{C}\mathbf{X}_k + \mathbf{D}\omega_k)^T \mathbf{P} (\mathbf{C}\mathbf{X}_k + \mathbf{D}\omega_k) - \mathbf{X}_k^T \mathbf{P} \mathbf{X}_k \\ &= \mathbf{X}_k^T \mathbf{C}^T \mathbf{P} \mathbf{C} \mathbf{X}_k + \mathbf{X}_k^T \mathbf{C}^T \mathbf{P} \mathbf{D} \omega_k + \omega_k^T \mathbf{D}^T \mathbf{P} \mathbf{C} \mathbf{X}_k + \omega_k^T \mathbf{D}^T \mathbf{P} \mathbf{D} \omega_k - \mathbf{X}_k^T \mathbf{P} \mathbf{X}_k \end{aligned} \quad (4.33)$$

Reverting the notation in Eq. 4.24, Eq. 4.33 can be written in matrix form as

$$\Delta V = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \begin{bmatrix} (\mathbf{A} - \mathbf{BF})^T \mathbf{P} (\mathbf{A} - \mathbf{BF}) - \mathbf{P} & (\mathbf{A} - \mathbf{BF})^T \mathbf{P} \mathbf{D} \\ \mathbf{D}^T \mathbf{P} (\mathbf{A} - \mathbf{BF}) & \mathbf{D}^T \mathbf{P} \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} \quad (4.34)$$

For the proof, we introduce an accessory parameter  $\alpha \geq 0$  in Eq. 4.26, such that

$$\Upsilon = \alpha \omega^T \omega - \alpha \gamma^2 \mathbf{X}_k^T \mathbf{F}^T \mathbf{F} \mathbf{X}_k < 0 \quad (4.35)$$

From Eq. 4.35, the left hand side of the inequality is written in matrix form as

$$\Upsilon = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \begin{bmatrix} -\alpha \gamma^2 \mathbf{F}^T \mathbf{F} & 0 \\ 0 & \alpha \mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} < 0 \quad (4.36)$$

Adding and subtracting this term to Eq. 4.34 allows us to find a bounding for  $\Delta V$  as

$$\Delta V - \Upsilon + \Upsilon = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \mathbf{Q} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} + \Upsilon \quad (4.37)$$

with

$$\mathbf{Q} = \begin{bmatrix} (\mathbf{A} - \mathbf{BF})^T \mathbf{P} (\mathbf{A} - \mathbf{BF}) - \mathbf{P} + \alpha \gamma^2 \mathbf{F}^T \mathbf{F} & (\mathbf{A} - \mathbf{BF})^T \mathbf{P} \mathbf{D} \\ \mathbf{D}^T \mathbf{P} (\mathbf{A} - \mathbf{BF}) & \mathbf{D}^T \mathbf{P} \mathbf{D} - \alpha \mathbf{I} \end{bmatrix} \quad (4.38)$$

We know from Eq. 4.35 that  $\Upsilon < 0$ . Therefore, if  $\mathbf{Q}$  is definite negative, then  $\Delta V < 0$ . To prove the closed-loop system to be stable, the values for matrix  $P > 0$  and  $\alpha \geq 0$  need to be found such as matrix  $\mathbf{Q}$  is definite negative, given the predefined values of the boundary parameter  $\gamma$  and the tuned controller parameter  $k_p$  and  $k_i$ . To this end, a Linear Matrix Inequality [Boyd 1994] Solver called SeDuMi [Sturm 1999] is used in the software Matlab. In order to describe the LMI given by Eq. 4.32, Yalmip [Lofberg 2004], a toolbox for optimization that is compatible with Matlab is employed. Given a value of  $\gamma = 0.98$  and the gain values  $k_p = 0.14$  and  $k_i = 0.0003$ , the LMI was solved successfully. The resulting matrix  $P$  and the parameter  $\alpha$  that make matrix  $Q$  negative definite are

$$\mathbf{P} = \begin{bmatrix} 646.4512 & 1.2983 \\ 1.2983 & 0.0087 \end{bmatrix} \quad \text{and} \quad \alpha = 4655 \quad (4.39)$$

Using the LMI solver, we can also compute the maximum value of  $\gamma$ , which provides an insight on the robustness of the closed-loop system. After some iterations we have

$$\max \gamma = 0.98685 < 1 \quad (4.40)$$

One can see that if  $\gamma > 1$  the sign in the control input could potentially change which would lead to instability, so  $\gamma = 0.98685$  is close to the limit case. The proposed closed-loop system is robustly stable and can handle high Jacobian inversion errors in the change of control variables.

## 4.4 Conclusions of the chapter

In this chapter, feed-forward and closed-loop controllers based on the FEM model for the CBHA were studied. Given the fact that the estimation of the Jacobian matrix depends on the configuration of the robot, the feed-forward control was ineffective in terms of precise positioning. Based on this observation, a closed-loop control strategy that relies on the forward kinematic model as a state estimator was proposed. The state estimator is able to apply the configuration of the robot to the simulation. The closed-loop was tuned using a direct FEM simulation of the CBHA and then implemented in the real robot. The performance of the controller was showed and the proof of stability was given.

As explained in previous chapters, the model employed in the controller works under quasi-static conditions, which means that it is assumed that once the robot reaches a desired position, a period of stabilization is required before moving to the next position. The natural next step in the control of continuum manipulators would be to extend the controller to the dynamic case in which the inertial effects of the manipulator are considered. This would allow for the control of the acceleration of the robot, required for more complex applications. In general however, in order to capture high frequency vibrations of the structure, a higher sampling rate is required.

In 4.3, we mentioned that a reference computation was performed in order to change the spaces related by our models. In this case, the robot is actuated directly to achieve a certain cartesian position, then, a sensor reading is performed to obtain the lengths that correspond to said position. However, thanks to the implementation of the state estimator, an end-effector trajectory can be set manually by manipulating the trunk directly while recording the lengths with the estimator. This manner of pre-computing the reference is more intuitive and can be used when the control commands for a desired position are not known.

In this chapter, the closed-loop control strategy for continuum manipulators was presented and the proof of stability was given. However, in the robustness analysis of the controller, the only assumption imposed on the inversion uncertainty  $\omega_k$  was its norm boundedness. No structure was assumed on  $\omega_k$ . This simplification of the uncertainties may lead to extreme conservative results. For example, consider a system composed by two actuators with a proportional gain  $k_p$  described by

$$e_{1k+1} = (1 - kp)e_{1k} + \omega_{1k} \quad (4.41)$$

$$e_{2k+1} = (1 - kp)e_{2k} + \omega_{2k} \quad (4.42)$$

given the values of uncertainties

$$\omega_{1k} = 0, \quad \omega_{2k} = 10^5 e_{1k} \quad (4.43)$$

The LMI proposed before is unfeasible with these values but if we study this model it can be proved to be stable. The system described in Eq. 4.41 and 4.42 is not realistic in the context of soft continuum manipulators, since more often than not the actuators are coupled. Nevertheless, the example helps to highlight the need of structured system uncertainties [Doyle 1982]. For a structured system, the inversion uncertainty takes the form

$$\omega_k = \mathbf{E}\Delta\mathbf{F}\mathbf{X} \quad (4.44)$$

where  $\mathbf{E}$  and  $\mathbf{F}$  are called *uncertainty weighting filters*. The boundedness is now applied to the uncertainty structure  $\Delta \leq \gamma\mathbf{I}$ . Special procedures have been developed to deal with structured bounded uncertainties [Morari 1989].



# Conclusion and Perspectives

---

## Contents

---

<b>5.1</b>	<b>Summary of conclusions</b>	<b>83</b>
<b>5.2</b>	<b>The FeTCh manipulator</b>	<b>85</b>
<b>5.3</b>	<b>Perspectives</b>	<b>88</b>

---

The main objective of this work was to provide a methodology to study and model soft, continuum manipulators based on computational mechanics. The methodology was used in the kinematic modelization of a continuum manipulator, the CBHA. We provided new control methods based on such models. We have validated the methodology with experiments that involve the real-time simulation of the FEM model. In the following, the conclusions for each chapter are given.

## 5.1 Summary of conclusions

In chapter 2, a concise and complete view of the field as it is was given. The main problem on the modelization of continuum manipulators was introduced and a brief study on the most popular approaches towards the modeling in the literature was given and a qualitative evaluation of these approaches was presented. In order to make room for our contribution in the field, the role that external forces play in the motion of continuum manipulators was also presented. Also, a taxonomy of the different types of designs was presented in order to highlight the main actuation differences. In our opinion, the most important part of the state of the art was the introduction of the main issue that most of the modeling approaches fail to capture: the deformation mechanics! Continuum manipulators produce their motion by deformation, and yet, very few approaches make emphasis on this feature.

Chapter 3 started as an introduction on continuum mechanics in an attempt to show that the study of deformations seems like a more natural and compatible tool to model continuum manipulators. The Finite Element Method was presented as a method to solve differential equations with boundary conditions; a problem that is encountered when studying continuum media. Special emphasis was put in the concept of discretization, that allow us to reduce the infinite number of degrees of freedom of continuum manipulators. The methodology we use towards the modeling and control of soft, continuum manipulators was introduced and the model reduction based on the projection of the FEM model was explained. The model of sensors, actuators and end-effector as constraints in the FEM was explained. The mechanical coupling between the different constraints, provided by the compliance operators, was shown to be a necessity in the computation of the Jacobian matrix for continuum manipulators. The model of actuators and sensors based on Lagrange multipliers was used in the implementation of the kinematic models for the CBHA. The CBHA manipulator was introduced as a case of study to exemplify the implementation of the methodology. While our methodology does not produce analytic solutions for the kinematic model, the numerical approximation of the kinematic solution was shown superior to those obtained by 2 other geometric methods in terms of tracking position.

As a conclusion for chapter 3, some of the uncertainties in the model of the CBHA were discussed. This gave room for the implementation of the closed-loop control presented in chapter 4. The simulation of the FEM model was used to control the manipulator in real time. The discussion about the feed-forward control of the CBHA provided a good example of the need for closed-loop controllers. While there are a lot of non-linear behaviors that are currently not included in the model, the closed-loop control is still able to provide good performance in terms of the positioning of the manipulator. The proposed controller was proved to be robustly stable.

As a conclusion to this work, we have shown the feasibility of using FEM simulation as a generic methodology for modeling and control of soft, continuum manipulators, based on the premise that the study of deformations is fundamental if one seeks to characterize this type of robots.



One interesting venue for future research of the methodology presented in this manuscript is its application to the design of soft, continuum manipulators. Since the simulation of the backbone of continuum robots can be done in a fast manner. The simulation can be used to reduce the number of design iterations performed to achieve a certain behaviour. The fabrication of soft actuators, in particular fluidic actuators, is a complex task. Being able to observe the behavior that a certain design has, prior to its fabrication, is of great help when undertaking the design of a new prototype. This approach towards the design and integration of soft, continuum manipulators has been explored in the past, with the design and construction of the FeTCh manipulator.

## 5.2 The FeTCh manipulator

The prototype of the FeTCh manipulator was designed with the intention of using the principle of antagonistic actuation to achieve the rigidification of its structure [Shiva 2016]. The backbone of the robot is composed by actuators in an intrinsic fashion, while including extrinsic tendon actuators. Each section of the robot is formed by three pneumatic actuators arranged in parallel configuration between two rigid platforms that guide the deformation of the actuators. This configuration gives the section 2 axis bending, and also extension. The arrangement of the actuators also helps to counter the lack of shearing stiffness that is characteristic to pneumatic actuators. The described section of the manipulator is shown in Fig. 5.1.

The extrinsic actuation of the manipulator is composed by tendons that are pulled by a servomotor-pulley system. These tendons are routed through the rigid platforms at  $120^\circ$  around the rigid platform at the end of each section and apply a force that is regulated. The inclusion of the tendons provides more stability to the manipulator when it moves, but also provide a degree of selective rigidification of the structure.

The idea behind the actuator design was to create an extensor that can elongate when pressurized, as opposed to the more traditional contractor artificial muscles. The extension of the actuator is caused by the unfolding of its walls, rather than pure deformation caused by the pressure. The simulation of the actuator was used in order to observe

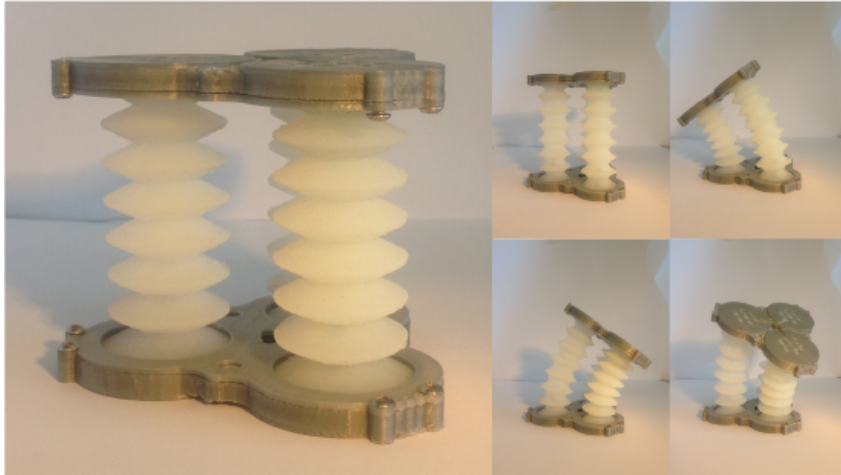


Figure 5.1: Single section of the FeTCh manipulator showing different configurations depending on the pressure inside the actuators

its behaviour and obtain the actuation values (in this case the pressure inside the cavity) for a desired elongation.

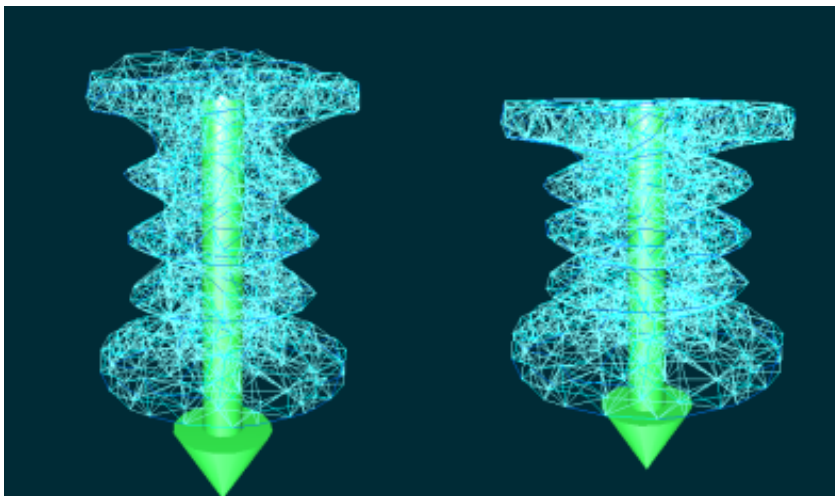


Figure 5.2: Simulation of the soft actuator.

After the validation of the design by the simulation, the actuator was built and tested. The actuator has a maximum extension of 81.25% when pressurized with 0.41bars. The final design of the manipulator is achieved by serially connecting 3 sections like the one shown in Fig. 5.1.

The design allows for the positioning of the final rigid platform, but also for its orientation. Further testing on the rigidification effect based on

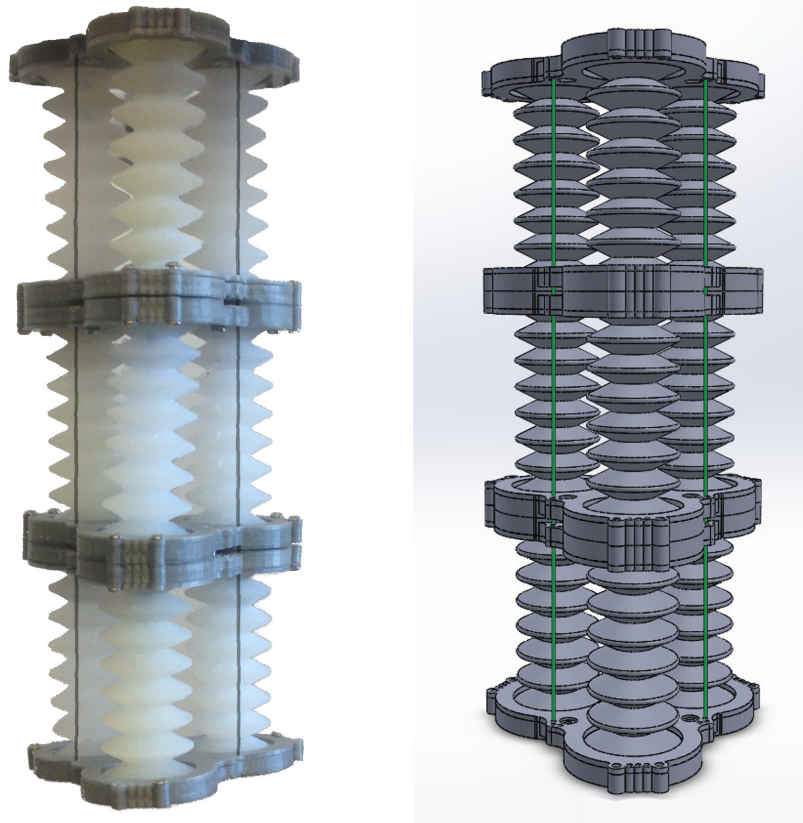


Figure 5.3: The FeTch manipulator.

the FEM model is still required but the preliminary results are promising.

The simulation of soft structures prior its fabrication allows also for the testing of the design robustness. Since the integration of soft robots in general is not a perfected process, defects in the soft parts are often encountered. For example, the most common flaw in soft pneumatic actuators are artificial heterogenous regions formed by air bubbles. In the simulation, the inclusion of these regions is easily done. This allows the validation of designs when fabrication flaws are expected. In addition, the range of actuation can also be determined, this allows for the selection of key components like the servos for the tendons and the type of pneumatic manifolds to be used in the back-end of the manipulators. Once the final assembly for the manipulator has been established, the identification of the task space of the robot can also be performed.

### 5.3 Perspectives

The possible improvements in the current work regarding the modeling of continuum manipulators are numerous and it is difficult to mention them all here. However, based on the experience we have with the CBHA, some immediate modifications can be performed to improve the model without making it too complex for fast computation. For example, it is observed that at different temperatures, the manipulator behaves slightly differently; at higher temperature, the plastic material is more ductile. The inclusion of the model of thermal expansion in the element potential energy functional  $\Pi$  (Eq. A.52), from the FEM, can be done without changing its general formulation. By adding the term

$$h = \int_V \mathbf{B}^T \mathbf{E} \varepsilon^T dV \quad (5.1)$$

where  $\mathbf{B}$  is the relation between strains and nodal displacements and  $\mathbf{E}$  is the elasticity matrix, we can represent the fictitious forces that correspond to the thermal expansion of the material. Of course, in small scale manipulators, this term can be completely neglected, but for applications that require bigger manipulators, the effect of temperature in their behavior can be considerable. It is also observed that material fatigue affects the motion of the manipulator. After prolonged use of the robot, the pneumatic actuators expand more easily. Creeping and relaxation stages of plastic materials should also be considered in the model.

In terms of design, there is a great need for novel actuation technologies to be developed with the purpose of providing continuum robots with bigger force generation capabilities. At the present time, it is clear that soft manipulators have a niche of applications in the medical field. However, this *narrow* niche can be expanded provided that this type of robots can compete with rigid ones in terms of force generation and payload handling.

As a final perspective, much work needs to be done in terms of the formalization of the theory behind continuum manipulators. In order to expand and teach soft robotics as a field, a unified and comprehensive terminology and methodology of modeling and control needs to be firmly established. It is our hope that the work presented in this brief manuscript contributes towards that important objective.

# Introduction to Continuum Mechanics and FEM

---

## A.1 Introduction

The objective of this appendix is to give the reader a brief introduction to continuum mechanics and FEM. The interested reader should refer to [Constantinescu 2007], [Reddy 2013] for a more detailed explanation of the concepts presented in the sequel.

### A.1.1 Continuum Mechanics

When analyzing the effects of forces and energy on bodies from a mechanical point of view, we can choose to perform the analysis at different physical scales. Continuum Mechanics performs this study at the macroscopic scale, by assuming that the body is a continuous mass. By making this assumption, the molecular substructures can be disregarded and the body is then considered a *continuum*. In the following, the description of motion in continuum bodies is presented.

#### A.1.1.1 Concept of configuration and description of motion

To arrive at a description of motion in a continuum media, let us consider a body  $\mathbf{B}$  with all its material particles occupying a three dimensional domain  $\Omega \in \mathbf{R}$ . For a given set of loads (external forces), the body will *deform* changing its geometry and this deformation will be accompanied by internal body reaction forces, called *stress*. The new region  $\kappa_i$  occupied by the body after the  $i$ -th deformation is called the  $i$ -th *configuration* of the continuum. In this case,  $\Omega = \kappa_0$  is the initial or underformed configuration and it is used as a reference to express the deformation of the continuum body.

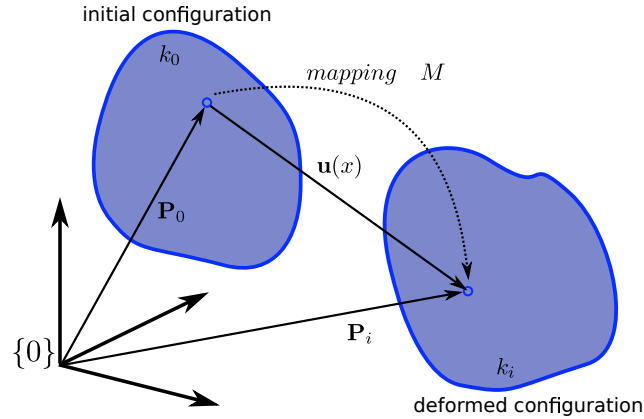


Figure A.1: Illustration of the Lagrangian description of motion between initial and deformed configuration for an elastic body

A material particle of  $\mathbf{B}$ , with a position vector  $\mathbf{P}_0$  in the initial configuration  $\kappa_0$  will have a new position vector  $\mathbf{P}_i$  in  $\kappa_i$  after deformation given by

$$\mathbf{P}_i = M(\mathbf{P}_0, t) \quad (\text{A.1})$$

The function  $M$  is called a *deformation mapping* and describes a 1-to-1 relation between particle positions in the initial configuration and the new position in the deformed configuration. The displacement of a particle in a continuum can be defined as

$$\mathbf{u}(\mathbf{P}_i) = \mathbf{P}_i - \mathbf{P}_0 = M(\mathbf{P}_0, t) - \mathbf{P}_0 \quad (\text{A.2})$$

This description of motion of continuum is referred as the *Lagrangian* description (Fig. A.1).

Now consider an infinitely small line segment in the body  $\mathbf{B}$  given by  $d\mathbf{P}_0$  and  $d\mathbf{P}_i$  before and after deformation, respectively, as shown in Fig. A.2. The relationship between these two lines is given by the gradient of deformation  $\mathbf{F}$ , defined as

$$\mathbf{F} = \left( \frac{\partial M}{\partial P_i} \right)^\top = \nabla \mathbf{u} + \mathbf{I} \quad (\text{A.3})$$

where  $\nabla$  is the gradient operator with respect to  $\mathbf{P}_i$  and  $\mathbf{I}$  is the identity matrix. The determinant of  $\mathbf{F}$ ,  $J = \det \mathbf{F}$  is referred to as the *Jacobian of motion*.

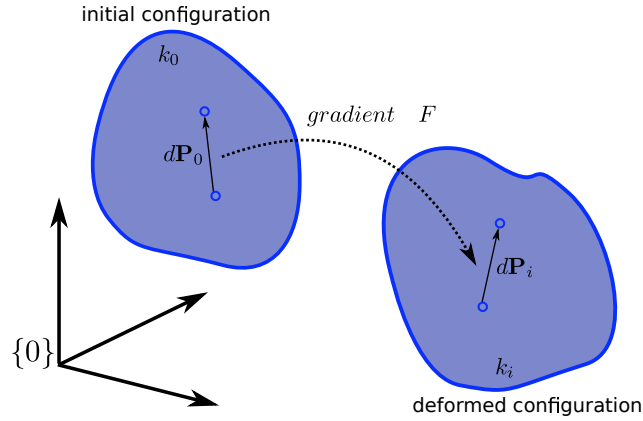


Figure A.2: Illustration of the gradient of deformation in the Lagrangian description of motion

### A.1.1.2 Stress and Strain

The displacement of particles in a continuum media can be considered, in the general case, as a result of a rigid body motion (a translation and a rotation) and a deformation. A measure of strain is the computation of the deformation part of the displacement, independent of the rigid body motion.

In order to arrive at the definition of the strain tensor, we define the *right Cauchy-Green tensor* as

$$\mathbf{C} = \mathbf{F} \cdot \mathbf{F}^\top \quad (\text{A.4})$$

which gives us the square of local displacement due to the deformation. Similarly, the *left Cauchy-Green tensor* is defined as

$$\mathbf{B} = \mathbf{F}^\top \cdot \mathbf{F} \quad (\text{A.5})$$

Using Eq. A.4, we define the *Green - Lagrange strain tensor* in vector form as

$$\mathbf{E} = \frac{1}{2} (\mathbf{F} \cdot \mathbf{F}^\top - \mathbf{I}) = \frac{1}{2} (\mathbf{C} - \mathbf{I}) \quad (\text{A.6})$$

with  $\mathbf{E} = 0$  when the displacement is a result of a rigid body motion. Eq. A.6 is expressed in Cartesian coordinates  $(X_1, X_2, X_3)$ , as

$$E_{i,j} = \frac{1}{2} \left( \frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \frac{\partial u_k}{\partial X_j} \right), \forall \quad 1 \leq (i, j) \leq 3 \quad (\text{A.7})$$

## 92 Appendix A. Introduction to Continuum Mechanics and FEM

where  $u_i$  are the cartesian components of the displacement,  $E_{11}, E_{22}$  and  $E_{33}$  are called normal strains and  $E_{12}, E_{23}$  and  $E_{13}$  are called shear strains. The tensor can also be written in terms of the gradient of the displacement

$$\mathbf{E} = \frac{1}{2} [(\mathbf{I} + \nabla \mathbf{u}) \cdot (\mathbf{I} + \nabla \mathbf{u})^\top - \mathbf{I}] = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^\top + (\nabla \mathbf{u}) \cdot (\nabla \mathbf{u})^\top] \quad (\text{A.8})$$

In the case of infinitesimal strains, the displacement gradients are small and the nonlinear terms in Eq. A.8 can be neglected. The infinitesimal strain tensor becomes

$$\varepsilon = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^\top] \quad (\text{A.9})$$

Also important in the study of continuum motion is the measure of stress. As mentioned before, when a body deforms, the deformation causes reaction forces within the body. These internal forces are called stress and are defined as the force per unit area.

Consider a small element of area  $a$  in a continuum media. The stress vector  $\mathbf{m}$  acting on the surface of the element is defined as

$$\mathbf{m}(\hat{\mathbf{n}}) = \lim_{\Delta a \rightarrow 0} \frac{\Delta \mathbf{f}(\hat{\mathbf{n}})}{\Delta a} \quad (\text{A.10})$$

where  $\Delta \mathbf{f}(\hat{\mathbf{n}})$  is the force acting on a small surface of area  $\Delta a$  and  $\hat{\mathbf{n}}$  is the unit vector normal to that area. We can also write  $\mathbf{m}$  in terms of the *Cauchy stress tensor*  $\boldsymbol{\sigma}$  as

$$\mathbf{m}(\hat{\mathbf{n}}) = \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} \quad (\text{A.11})$$

The Cauchy stress tensor  $\boldsymbol{\sigma}$  is defined as the current force per unit of deformed area. The *first Piola-Kirchhoff stress tensor* or *Lagrangian stress tensor*

$$\mathbf{G} = J \boldsymbol{\sigma} \cdot \mathbf{F}^\top \quad (\text{A.12})$$

is defined as the current force per unit of undeformed area. In a similar way, we introduce the *second Piola-Kirchhoff stress tensor*  $\mathbf{S}$ , defined as the transformed current force per unit of undeformed area. The second



Piola-Kirchhoff stress tensor is associated to the first Piola-Kirchhoff stress tensor by the equation

$$\mathbf{S} = \mathbf{F}^\top \cdot \mathbf{G} \quad (\text{A.13})$$

In practice, the second Piola-Kirchhoff stress tensor is used in the study of large deformations, useful for the simulation of elastic materials.

### A.1.1.3 Constitutive equations

In order to reconstruct the deformed configuration of a body under a set of external forces, a relationship between stress and strain is necessary. In our study, we will continue to consider the case of small strains. For linear elastic materials, the relationship between the components of stress and strain tensors at each point of the body is given by *Hooke's law*, which is written in tensor form as

$$\sigma_{ij} = \sum_{1 \leq (k,l) \leq 3} C_{ijkl} \varepsilon_{kl}, \quad \forall \quad 1 \leq (i, j) \leq 3 \quad (\text{A.14})$$

where the scalar coefficients  $C_{ijkl}$  are called stiffness coefficients of the *stiffness* tensor  $\mathbf{C}$ . In the general case,  $\mathbf{C}$  is formed by 81 coefficients, but due to the symmetry of the stress and strain tensors, the coefficients are reduced to 21. Using single subscript notation for stresses and strains and two subscripts for the stiffness coefficients

$$\begin{aligned} \sigma_1 &= \sigma_{11}, & \sigma_2 &= \sigma_{22}, & \sigma_3 &= \sigma_{33}, & \sigma_4 &= \sigma_{23}, & \sigma_5 &= \sigma_{13}, & \sigma_6 &= \sigma_{12} \\ \varepsilon_1 &= \varepsilon_{11}, & \varepsilon_2 &= \varepsilon_{22}, & \varepsilon_3 &= \varepsilon_{33}, & \varepsilon_4 &= \varepsilon_{23}, & \varepsilon_5 &= \varepsilon_{13}, & \varepsilon_6 &= \varepsilon_{12} \\ 11 &\rightarrow 1 & 22 &\rightarrow 2 & 33 &\rightarrow 3 & 23 &\rightarrow 4 & 13 &\rightarrow 5 & 12 &\rightarrow 6 \end{aligned} \quad (\text{A.15})$$

Eq. A.14 takes the form

$$\sigma_i = C_{ij} \varepsilon_j \quad (\text{A.16})$$

which is referred as the *Voigt-Kelvin* notation [Dellinger 1998]. In matrix notation, Eq. A.16 is written as

$$\begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & C_{14} & C_{15} & C_{16} \\ C_{21} & C_{22} & C_{23} & C_{24} & C_{25} & C_{26} \\ C_{31} & C_{32} & C_{33} & C_{34} & C_{35} & C_{36} \\ C_{41} & C_{42} & C_{43} & C_{44} & C_{45} & C_{46} \\ C_{51} & C_{52} & C_{53} & C_{54} & C_{55} & C_{56} \\ C_{61} & C_{62} & C_{63} & C_{64} & C_{65} & C_{66} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ 2\varepsilon_4 \\ 2\varepsilon_5 \\ 2\varepsilon_6 \end{bmatrix} \quad (\text{A.17})$$

with  $C_{ij} = C_{ji}$ . The components of strain are related to the components of stress by

$$\varepsilon_i = W_{ij}\sigma_j \quad (\text{A.18})$$

where  $W_{ij}$  are the coefficients of the *compliance* tensor  $\mathbf{W} = \mathbf{C}^{-1}$ . Eq. A.18 is written in matrix notation as

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ 2\varepsilon_4 \\ 2\varepsilon_5 \\ 2\varepsilon_6 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & W_{14} & W_{15} & W_{16} \\ W_{21} & W_{22} & W_{23} & W_{24} & W_{25} & W_{26} \\ W_{31} & W_{32} & W_{33} & W_{34} & W_{35} & W_{36} \\ W_{41} & W_{42} & W_{43} & W_{44} & W_{45} & W_{46} \\ W_{51} & W_{52} & W_{53} & W_{54} & W_{55} & W_{56} \\ W_{61} & W_{62} & W_{63} & W_{64} & W_{65} & W_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} \quad (\text{A.19})$$

**Orthotropic materials** Orthotropic materials have three mutually orthogonal planes of material symmetry, meaning that their material properties differ when measured from different directions. In this case, the elastic coefficients are reduced to 9. Most of the time, experimental characterization of material is performed using known loads; it is then convenient to express the relation of stress and strain in terms of the compliance. Eq. A.19 for orthotropic materials has the form

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ 2\varepsilon_4 \\ 2\varepsilon_5 \\ 2\varepsilon_6 \end{bmatrix} = \begin{bmatrix} W_{11} & W_{12} & W_{13} & 0 & 0 & 0 \\ W_{21} & W_{22} & W_{23} & 0 & 0 & 0 \\ W_{31} & W_{32} & W_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & W_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & W_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & W_{66} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} \quad (\text{A.20})$$

In the experimental characterization of linear elastic materials, the coefficients of the compliance tensor are computed using the *Young's modulus*, the *shear modulus* and the *Poisson ratio* [Wortman 1965]. After introducing these engineering constants, Eq. A.20 takes the form

$$\begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ 2\varepsilon_4 \\ 2\varepsilon_5 \\ 2\varepsilon_6 \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & -\frac{\nu_{21}}{E_2} & -\frac{\nu_{31}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{12}}{E_1} & \frac{1}{E_2} & -\frac{\nu_{32}}{E_3} & 0 & 0 & 0 \\ -\frac{\nu_{13}}{E_1} & -\frac{\nu_{23}}{E_2} & \frac{1}{E_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{23}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{13}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{12}} \end{bmatrix} \begin{bmatrix} \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \\ \sigma_5 \\ \sigma_6 \end{bmatrix} \quad (\text{A.21})$$

where  $E_1$ ,  $E_2$  and  $E_3$  are the Young's moduli in the three main material directions,  $\nu_{ij}$  is Poisson's ratio, defined as the ratio of traverse strain in the  $j$ th direction to the axial strain in the  $i$ th direction when stressed in the  $i$ -direction, and  $G_{23}$ ,  $G_{13}$  and  $G_{12}$  are the shear moduli in the 2-3, 1-3 and 1-2 planes respectively.

When computed experimentally, the Young's modulus, which is a measure of linear stiffness, is represented by the slope of the linear part in the stress-strain relationship, while the Poisson's ratio is a measure of the change in volume of a material.

**Isotropic materials** If the material properties are independent from the direction, meaning that

$$E_1 = E_2 = E_3 = E \quad , \quad G_{23} = G_{13} = G_{12} = G \quad , \quad \nu_{12} = \nu_{23} = \nu_{13} = \nu \quad (\text{A.22})$$

the material is said to be isotropic. For isotropic materials, the stress-strain relationship is reduced to

$$\sigma_{ij} = \frac{E}{1+\nu} \varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)} \varepsilon_{kk} \delta_{ij} \quad (\text{A.23})$$

where  $\delta_{ij}$  is called the *Kronecker Delta* and it is used as a simple representation of the scalar product

$$\delta_{ij} = \hat{\mathbf{e}}_i \cdot \hat{\mathbf{e}}_j \quad (\text{A.24})$$

that takes the values  $\delta_{ij} = 1$  if  $i = j$  and  $\delta_{ij} = 0$  otherwise. The vectors  $\hat{\mathbf{e}}_i$  and  $\hat{\mathbf{e}}_j$  are the orthonormal unit vectors in a right-handed basis system. Introducing the Lamé coefficients [Akamatsu 1991]

$$\mu = \frac{E}{2(1 + \nu)} \tag{A.25}$$

$$\lambda = \frac{E\nu}{(1 + \nu)(1 - 2\nu)}$$

the constitutive equation for isotropic linearly elastic material takes the form

$$\sigma_{ij} = 2\mu\varepsilon_{ij} + \sum_{k=1}^3 \varepsilon_{kk}\delta_{ij} \tag{A.26}$$

#### A.1.1.4 Equations of motion

The evolution of the motion in continuum bodies is derived from the *Principle of Conservation of Linear Momenta* commonly known as *Newton's second law of motion* that can be stated as the time rate of linear momentum of a collection of particles equals the net force exerted on the collection. This principle allow us to express the relationships between the primary unknown displacement field of the particles of a continuum body in terms of stresses and strains. The solution of these equations represent the response of the system to a set of input (known forces) data. In vector form, the principle is represented by

$$m \frac{d\mathbf{v}}{dt} = \mathbf{F} \tag{A.27}$$

where  $m$  is the constant total mass,  $\mathbf{v}$  is the velocity, and  $\mathbf{F}$  is the net force applied on the particles. For a body over the control volume  $\Omega$ , Eq. A.27 takes the form

$$\mathbf{F} = \frac{\partial}{\partial t} \int_{\Omega} \rho d\mathbf{v} + \int_{\Gamma} \rho \mathbf{v} \cdot d\mathbf{s} \tag{A.28}$$

where  $d\mathbf{s}$  is the vectorial representation of an area element,  $\rho$  is the mass density and  $\Gamma$  is the boundary surface of the body. Let  $\mathbf{f}$  be the external body force per unit of mass, which acts on the distribution of mass inside

the body. Considering an elemental volume  $d\Omega$  inside  $\Omega$ , the body force of  $d\Omega$  is equal to  $\rho d\Omega \mathbf{f}$ . The total body force in the control volume is

$$\int_{\Omega} \rho \mathbf{f} d\Omega \quad (\text{A.29})$$

Let  $\mathbf{m}$  be the surface force per unit area. The surface force acting on an elemental surface  $ds$  of the volume is  $\mathbf{m} ds$ . The total surface force acting on the closed surface of  $\Omega$  is

$$\oint_{\Gamma} \mathbf{m} ds \quad (\text{A.30})$$

rewriting  $\mathbf{m}$  in terms of Cauchy's stress tensor (Eq. A.11), and using the divergence theorem, we have

$$\oint_{\Gamma} \hat{\mathbf{n}} \cdot \boldsymbol{\sigma} ds = \int_{\Omega} \nabla \cdot \boldsymbol{\sigma} d\Omega \quad (\text{A.31})$$

Using the Reynolds transport theorem [Cengel 2006], the global form of the equation of motion of the body occupying the region  $\Omega$  with bounding surface  $\Gamma$  and acted upon the surface force  $\mathbf{m}$  and body force  $\mathbf{f}$  is given by

$$0 = \int_{\Omega} \left[ \nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} - \rho \frac{D\mathbf{v}}{Dt} \right] d\mathbf{x} \quad (\text{A.32})$$

where  $\frac{D}{Dt}$  denotes the *material time derivative*. The local form is

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} = \rho \frac{D\mathbf{v}}{Dt} \quad (\text{A.33})$$

or

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} = \rho \left( \frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} \right) \quad (\text{A.34})$$

For infinitesimal deformation of solid bodies in static equilibrium, Eq. A.33 becomes

$$\nabla \cdot \boldsymbol{\sigma} + \rho \mathbf{f} = \mathbf{0} \quad (\text{A.35})$$

In Cartesian rectangular coordinates, we have

$$\frac{\partial \sigma_{ji}}{\partial x_j} + \rho f_i = 0, \forall \quad 1 \leq (i, j) \leq 3 \quad (\text{A.36})$$

In the general case, Eq. A.34 is the partial differential equation that describes the motion of a deformable body. However, an analytical solution to Eq. A.34 is intractable for most cases. In turn, an approximation of the solution is computed using numerical methods.

## A.2 Finite Element Method for linear elastic bodies

In order to reconstruct the sequence of configurations that an elastic body will take, given a set of forces acting on it, Eq. A.36 remains to be solved. This equation, however, cannot be solved analytically, since it involves an infinite number of degrees of freedom. In general, finding the solution for continuum elastic solids is done by making use of numerical methods provided by the field of Computational Mechanics, since analytical solutions exist only for trivial problems. One of these numerical methods is the Finite Element Method (FEM).

In the finite element method, a geometrically complex domain  $\Omega$  is viewed as being comprised by a finite set of subdomains (elements)  $\Omega_e$  over which the solution of the governing equation is approximated using variational methods<sup>1</sup>. The division of the domain in subdomains allows for an accurate representation of complex geometry.

We can summarize the Finite Element Method in three main steps:

1. The division of the domain of interest in smaller subdomains. We call this step the *discretization* step
2. The approximation of the solution at the element level as a linear combination of *nodal values* and *shape functions*
3. The *assembly* of the elements that allows to approximate the global solution over the whole original domain

In the following, these three steps are briefly explained for the formulation of the displacement-based FEM [Bathe 2006], which is used in

---

<sup>1</sup>In fact, FEM can be viewed as a piecewise variational method

the manuscript to derive the kinematics of soft manipulators. If more detailed explanation is required regarding the general formulation of the problem using variational methods, the reader is advised to refer to [Reddy 1993].

### A.2.1 Discretization of the domain

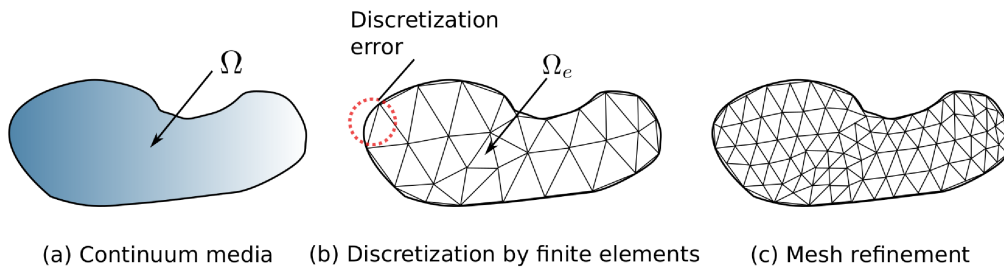


Figure A.3: Discretization of the domain. A continuum domain (a) is approximated by triangle elements (b) which produces a discretization error. The mesh is refined (c) by increasing the number of elements. This in turn reduces the error.

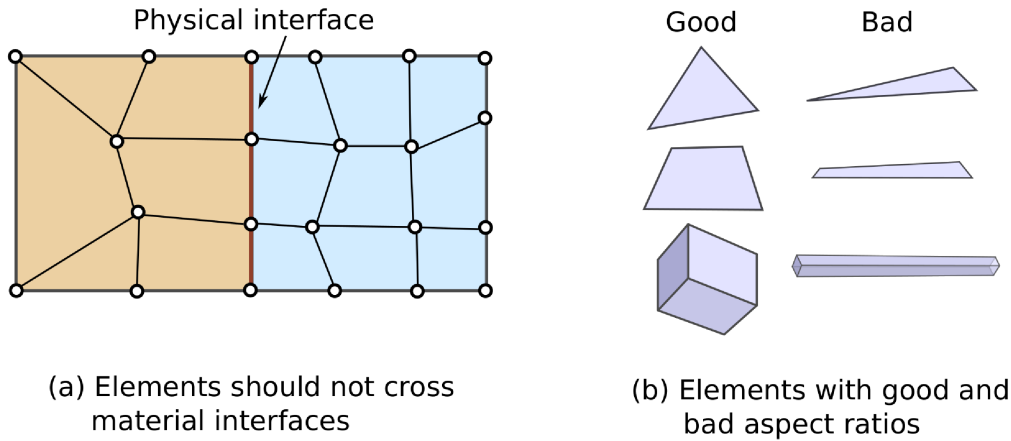
The first step in FEM is the partition of the domain of interest  $\Omega$  into smaller, geometrically independent domains called finite elements  $\Omega_e$ . The collection of finite elements that form the discrete domain is called *mesh* (Fig. A.3). In this step, the first source of error in a finite element analysis is introduced: In most cases, the discrete representation will differ geometrically from the original domain. It is evident that this error can be diminished by reducing the dimensions of the elements to better conform to  $\Omega$ , but this results in an increased number of elements involved in the computation which, in turn, require more computational resources. A compromise between an accurate representation of the complex geometry and the number of elements is needed. In fact, mesh refinement and convergence is still an active research topic.

While there are some commercially available applications for automatic mesh generation, it is important to mention the general set of rules that a finite element mesh must follow. This set of rules are:

Finite elements in a mesh do not overlap in space. This property goes by the name of *local support*

Elements with high aspect ratios should be avoided. The aspect ratio of a three-dimensional element is the ratio between its largest and smallest dimensions

A physical interface (Fig A.4 (a)), resulting from a change in material, should be an interelement boundary. Elements should not cross an interface



(a) Elements should not cross material interfaces

(b) Elements with good and bad aspect ratios

Figure A.4: Illustration of the mesh rules. Elements should not cross physical interfaces formed by a change of material (a). Elements with high aspect ratios (b) should be avoided.

In the following, the treatment given to an element is explained.

### A.2.2 Element solution

At the element level, the physical quantities of interest, which in our case is the displacement field  $u$  at any point inside the element, is approximated by a linear combination of  $u$  at the nodes of the element. This approximation is expressed as

$$u^e \approx \sum_{i=1}^n u_i^e \phi_i^e(x) \tag{A.37}$$

where  $u^e$  is the displacement inside the element,  $u_i^e$  is the displacement of node  $i$  and  $\phi_i^e$  are the interpolation functions that depend on the element geometry. In the following, we introduce the linear tetrahedron element and derive its interpolation models. In this case  $\phi_i^e$  are taken to be linear polynomials.



A.2.2.1 The linear Tetrahedron

The simplest solid element is the four-node tetrahedron, also known as the linear tetrahedron [Felippa 2001], depicted in fig A.5. The tetrahedron geometry is defined by the position of its nodes in Cartesian coordinates  $x_i, y_i$  and  $z_i$  with  $i = 1, 2, 3, 4$  or in the *tetrahedral natural coordinates* denoted by

$$\zeta_1, \zeta_2, \zeta_3, \zeta_4 \tag{A.38}$$

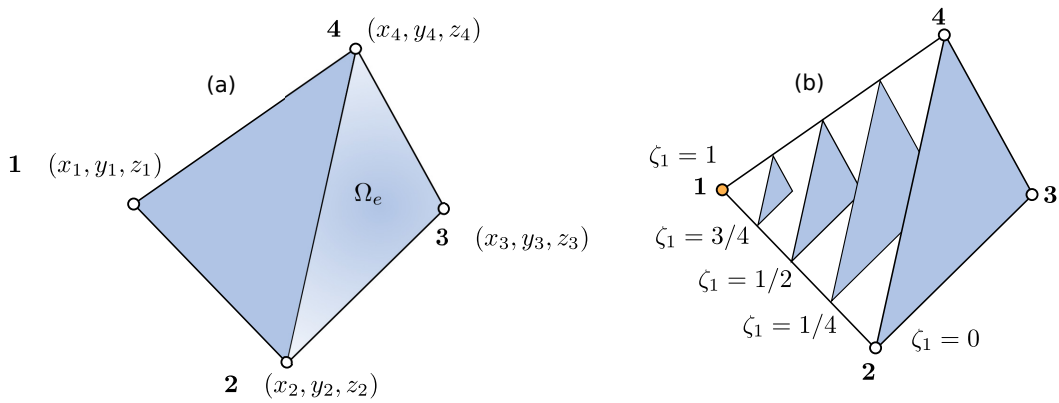


Figure A.5: Illustration of the tetrahedron element. (a) Element picture. (b) Visualization of  $\zeta_1$  as planes parallel to face 234. Figure adapted from [Felippa 2001]

The value of  $\zeta_i$  is one at corner  $i$  and zero at the other 3 corners, including the entire opposite face and it varies linearly with distance as one traverses the distance from the corner to that face.

Any function  $f(x, y, z)$  linear in  $x, y, z$  that takes values  $f_i (i = 1, 2, 3, 4)$  at the corners of the tetrahedron may be interpolated in terms of the coordinates of the tetrahedron by

$$f(\zeta_1, \zeta_2, \zeta_3, \zeta_4) = f_1\zeta_1 + f_2\zeta_2 + f_3\zeta_3 + f_4\zeta_4 \tag{A.39}$$

The interpolation functions are expressed in terms of the tetrahedron coordinates because they depend on the geometry of the element, however, other quantities like stress, strain and displacements are expressed in Cartesian coordinates. Therefore, a transformation between these two coordinate systems is needed. This transformation is given by

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \quad (\text{A.40})$$

For linear elastic material, the interpolation functions are simply the tetrahedron coordinates  $\zeta_i$ . The displacement field over the tetrahedron  $\mathbf{u}^e$  is defined by the three components  $u_x$ ,  $u_y$  and  $u_z$ . These are linearly interpolated from their node values

$$\begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} u_{x1} & u_{x2} & u_{x3} & u_{x4} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} \\ u_{z1} & u_{z2} & u_{z3} & u_{z4} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \quad (\text{A.41})$$

combining Eq. A.40 and A.41, we obtain the *isoparametric definition* of the tetrahedron as a displacement model

$$\begin{bmatrix} 1 \\ x \\ y \\ z \\ u_x \\ u_y \\ u_z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \\ u_{x1} & u_{x2} & u_{x3} & u_{x4} \\ u_{y1} & u_{y2} & u_{y3} & u_{y4} \\ u_{z1} & u_{z2} & u_{z3} & u_{z4} \end{bmatrix} \begin{bmatrix} \zeta_1 \\ \zeta_2 \\ \zeta_3 \\ \zeta_4 \end{bmatrix} \quad (\text{A.42})$$

The relation between the strains and the nodal displacements is given by

$$\varepsilon = \mathbf{B}\mathbf{u}^e \quad (\text{A.43})$$

where matrix  $\mathbf{B}$  is constant over the element, and has the form

$$\mathbf{B} = \begin{bmatrix} a_1 & 0 & 0 & a_2 & 0 & 0 & a_3 & 0 & 0 & a_4 & 0 & 0 \\ 0 & b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 \\ 0 & 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 \\ b_1 & a_1 & 0 & b_2 & a_2 & 0 & b_3 & a_3 & 0 & b_4 & a_4 & 0 \\ 0 & c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 \\ c_1 & 0 & a_1 & c_2 & 0 & a_2 & c_3 & 0 & a_3 & c_4 & 0 & a_4 \end{bmatrix} \quad (\text{A.44})$$

where

$$6V \frac{\partial \zeta_i}{\partial x} = a_i, \quad 6V \frac{\partial \zeta_i}{\partial y} = b_i, \quad 6V \frac{\partial \zeta_i}{\partial z} = c_i, \quad i = 1, 2, 3, 4. \quad (\text{A.45})$$

with  $V$  being the volume of the element. Similarly to Eq. A.20, the stress field is related to the strain field by

$$\sigma = \mathbf{C}\varepsilon \quad (\text{A.46})$$

for isotropic materials the relation in expanded form simplifies to

$$\begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{zz} \\ \sigma_{xy} \\ \sigma_{yz} \\ \sigma_{zx} \end{bmatrix} = \frac{E}{(1+v)(1-2v)} \begin{bmatrix} 1-v & 0 & 0 & 0 & 0 & 0 \\ 0 & 1-v & 0 & 0 & 0 & 0 \\ 0 & 0 & 1-v & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2}-v & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{2}-v & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2}-v \end{bmatrix} \quad (\text{A.47})$$

where  $E$  is the Youngs modulus and  $v$  is the Poisson ratio. Introducing Eq. A.43 and A.46 into the *Total Potential Energy* functional restricted to the element volume and assuming that the elasticity modulus do not vary over the element, we can compute the *element stiffness matrix* as

$$\mathbf{K}^e = V\mathbf{B}^T\mathbf{C}\mathbf{B} \quad (\text{A.48})$$

### A.2.3 Assembly

The purpose of the finite element solution of the elastic problem is to find the displacement field  $\mathbf{u}$  that provides minimum to the total potential energy functional  $\Pi$

$$\Pi = \int_V \frac{1}{2} \varepsilon^T \sigma dv - \int_V \mathbf{u}^T \mathbf{f} dV - \int_S \mathbf{u}^T \mathbf{m} dS \quad (\text{A.49})$$

with the second term corresponding to the volume forces, where  $\mathbf{f}$  is the body force vector and the third term corresponding to the surface forces where  $\mathbf{m}$  is the surface force vector. The displacement boundary conditions do not appear in the functional  $\Pi$ ; boundary conditions are implemented after the assembly of the finite element equations.

Using the relationships for stress and strain, derived in A.2.2.1, we can express Eq. A.49 in terms of the displacements of the nodes

$$\Pi = \int_V \frac{1}{2} (\mathbf{B}\mathbf{u}^e)^T \mathbf{C} (\mathbf{B}\mathbf{u}^e) dv - \int_V (\boldsymbol{\zeta}\mathbf{u}^e)^T \mathbf{f} dV - \int_S (\boldsymbol{\zeta}\mathbf{u}^e)^T \mathbf{t} dS \quad (\text{A.50})$$

where the superindex  $e$  denotes a value at the element level. Nodal displacement  $\mathbf{u}^e$  which corresponds to the minimum of the functional  $\Pi$  are determined by the condition

$$\frac{\partial \Pi}{\partial \mathbf{u}^e} = 0 \quad (\text{A.51})$$

Differentiating  $\Pi$  with respect to  $\mathbf{u}^e$  produces the following equilibrium equations for the finite element

$$\Pi = \int_V \mathbf{B}^T \mathbf{C} \mathbf{B} d\mathbf{u}^e - \int_V \boldsymbol{\zeta}^T \mathbf{f} dV - \int_S \boldsymbol{\zeta}^T \mathbf{m} dS = 0 \quad (\text{A.52})$$

Eq. A.52 is usually represented in the equivalent form

$$\mathbf{K}^e \mathbf{u}^e = \mathbf{F}^e \quad (\text{A.53})$$

with

$$\begin{aligned} \mathbf{K}^e &= \int_V \mathbf{B}^T \mathbf{W} \mathbf{B} dv \\ \mathbf{F}^e &= \int_V \boldsymbol{\zeta}^T \mathbf{f} dV - \int_S \boldsymbol{\zeta}^T \mathbf{t} dS \end{aligned} \quad (\text{A.54})$$

where matrix  $\mathbf{K}^e$  is the element stiffness matrix and  $\mathbf{F}^e$  is the vector of loads. The aim of the assembly is to form the global equation system

$$\mathbf{K} \mathbf{u} = \mathbf{F} \quad (\text{A.55})$$

using the element equations of the  $i$ th finite element

$$\mathbf{K}_i^e \mathbf{u}_i^e = \mathbf{F}_i^e \quad (\text{A.56})$$

Let us introduce the matrix  $\mathbf{K}_d$  and vectors  $\mathbf{u}_d$  and  $\mathbf{F}_d$  formed by the concatenation of Eq. A.56, such as

$$\begin{aligned}
\mathbf{K}_d &= \begin{bmatrix} \mathbf{K}_1^e & 0 & 0 \\ 0 & \mathbf{K}_2^e & 0 \\ 0 & 0 & \cdots \end{bmatrix} \\
\mathbf{u}_d &= (\mathbf{u}_1^e \quad \mathbf{u}_2^e \quad \cdots) \\
\mathbf{F}_d &= (\mathbf{F}_1^e \quad \mathbf{F}_2^e \quad \cdots)
\end{aligned} \tag{A.57}$$

with the relations

$$\begin{aligned}
\mathbf{K} &= \mathbf{A}^T \mathbf{K}_d \mathbf{A} \\
\mathbf{F} &= \mathbf{A}^T \mathbf{F}_d
\end{aligned} \tag{A.58}$$

where matrix  $\mathbf{K}$  is the global stiffness matrix. Here, matrix  $\mathbf{A}$  provides the transformation from global to local node enumeration. In practice, one node can be shared by several elements. The nodes have a local index (at the element level) and a global index (at the level of the mesh). Finally, we arrive at the following global equation system

$$\mathbf{A}^T \mathbf{K}_d \mathbf{A} \mathbf{u} - \mathbf{A}^T \mathbf{F}_d = 0 \tag{A.59}$$

#### A.2.4 Displacement boundary conditions

Displacement boundary conditions were not present in the total potential energy functional presented in Eq. A.49. They are usually applied to the global equation system, given by Eq. A.55, after its assembly for convenience. In the general case, before the application of boundary conditions, matrix  $\mathbf{K}$  is singular, which means that its rows and columns are linearly dependent and, therefore,  $\mathbf{K}$  is not a full rank matrix.

Let us consider the application of the displacement boundary condition

$$\mathbf{u}_m^e = d \tag{A.60}$$

to the global system Eq. A.55. We will use *explicit* method for this purpose.

In the explicit method, we substitute the known value of the displacement  $\mathbf{u}_m^e = d$  in the  $m$ th column of matrix  $\mathbf{K}$ , of dimensions  $n \times n$ , and move this column to the right-hand side of Eq. A.55

$$\begin{aligned}
\mathbf{F}_i &= \mathbf{F}_i - \mathbf{K}_{im} d, \quad i = 1 \dots n, \quad i \neq m \\
\mathbf{F}_m &= d
\end{aligned} \tag{A.61}$$

Then, we fill the  $m$ th row and  $m$ th column of matrix  $K$  with zeros, except for the main diagonal term, which is replaced by 1

$$\begin{aligned}\mathbf{K}_{mj} &= 0, & j &= 1..n, \\ \mathbf{K}_{im} &= 0, & i &= 1..n, \\ \mathbf{K}_{mm} &= 1\end{aligned}\tag{A.62}$$

After boundary conditions are applied, matrix  $\mathbf{K}$  has full rank.

### A.2.5 Solution method

For practical applications, the system Eq. A.55 results in a large number of simultaneous linear algebraic equations. Fortunately, matrix  $\mathbf{K}$  possesses some properties that enables an easy computation of the system of equations. For instance,  $\mathbf{K}$  is symmetric, which means that it is enough to store only half of the matrix entries including those in the diagonal. Also,  $\mathbf{K}$  is definite positive after setting the boundary conditions (Eq. A.62) and highly sparse, which makes storage and computations economic.

Solution methods for the simultaneous linear system of equations come in two different types: direct methods, such as those based on the  $\mathbf{LDL}^T$  decomposition or  $\mathbf{LDU}$  method with profile matrix [Bunch 1971], which are usually used for problems of moderated size, and iterative methods that require less computations for large scale problems. In practice, we use the iterative method of the *preconditioned conjugate gradient method* or PCG, since we often deal with problems involving thousands of degrees of freedom.

Let us consider the example system given by

$$\mathbf{A}x = b\tag{A.63}$$

In the PCG, the convergence of the conjugate gradient is improved by the preconditioning of the equation system

$$\mathbf{M}^{-1}\mathbf{A}x = \mathbf{M}^{-1}b\tag{A.64}$$

where  $\mathbf{M}^{-1}$  is the preconditioning matrix which in some sense approximates  $\mathbf{A}^{-1}$ . The simplest preconditioning is diagonal preconditioning, in which  $\mathbf{M}$  contains only diagonal entries of the matrix  $\mathbf{A}$ . There are

many existing solvers and libraries developed to solve large sparse linear matrix systems.





# Domain decomposition of continuum manipulators

---

In some continuum manipulator designs, the structure of the robot is composed by a sequence of serially connected substructures that have a vertebra-inter-vertebra pattern. In such designs, the inter-vertebra section shows a deformation that is greater than the vertebra under actuation. In the context of FEM modelization, the model dimensions of such manipulators can be significantly decreased by means of a domain decomposition technique.

## B.1 Domain decomposition of the CBHA

The body of the CBHA is composed by a set of substructures formed by the aforementioned vertebra-inter-vertebra pattern. The main idea of the domain decomposition is to discretize the deformation of the structure using a set of frames with 6 DoF, attached to the structure at the position of the vertebrae. In the simulation of the robot, the frames are nodes with 6 DoF which are kinematically linked to the inter-vertebra section. A non-linear stiffness matrix that correspond to the inter-vertebra section is pre-computed and condensed as springs with equivalent stiffness between the frames.

To pre-compute the equivalent stiffness, a substructure of the manipulator is isolated. The substructure is composed by 2 consecutive frames and the inter-vertebra between them. The nodes at the upper and lower boundary of the substructure are attached to the frames and have 6 DoF (pure rigid motion) while the nodes in-between the vertebrae have only 3 DoF (deformation). These 2 types of nodes need to be merged in the same mechanical system. The stiffness matrix  $\mathbf{K}$  of the inter-vertebra can be computed using a conventional FEM approach that takes into account the nodes kinematically linked to the rigid frames. Given the

## 11 Appendix B. Domain decomposition of continuum manipulators

---

computed matrix  $K$ , the compliance that corresponds to the rigid frames can also be computed as

$$\mathbf{C} = \mathbf{J}\mathbf{K}^{-1}\mathbf{J}^T \quad (\text{B.1})$$

where

$$\mathbf{J} = \begin{bmatrix} \mathbf{0}^{6 \times n} & \mathbf{I}^{6 \times 6} \end{bmatrix} \quad (\text{B.2})$$

with  $n$  being the number of independent DoF. The identity matrix in  $\mathbf{J}$  corresponds to the constraints of each DoF in the upper frame. Given two consecutive frames  $i$  and  $j$ , the equivalent stiffness matrix  $\mathbf{K}_{ij}^{eq}$  can be expressed as

$$\mathbf{K}_{ij}^{eq} = \mathbf{R}_i \begin{bmatrix} \mathbf{H}_{ij}\mathbf{C}_{ij}^{-1}\mathbf{H}_{ij}^T & \mathbf{H}_{ij}\mathbf{C}_{ij}^{-1} \\ \mathbf{C}_{ij}^{-1}\mathbf{H}_{ij}^T & \mathbf{C}_{ij}^{-1} \end{bmatrix} \mathbf{R}_i^T \quad (\text{B.3})$$

where  $\mathbf{R}_i$  is the rotational matrix of frame  $i$ . Matrix  $\mathbf{H}_{ij}$  describes the relationship between linear displacement on frame  $j$  and angular displacement on frame  $i$  and has the form

$$\mathbf{H}_{ij} = \begin{bmatrix} \mathbf{H}^{3 \times 3} & \mathbf{0}^{3 \times 3} \\ (\mathbf{x}_j - \mathbf{x}_i)^\wedge & \mathbf{H}^{3 \times 3} \end{bmatrix} \quad (\text{B.4})$$

with  $\mathbf{x}_i$  and  $\mathbf{x}_j$  are the position of frames  $i$  and  $j$  respectively and the symbol  $\wedge$  represents the skew symmetric matrix representation of a vector.

Using this approach, the FE model of a continuum manipulator can be replaced by a chain of frames linked by equivalent springs.

# Bibliography

- [Akamatsu 1991] Masayuki Akamatsu, Gen Nakamura and Stanley Steinberg. *Identification of Lamé coefficients from boundary observations*. Inverse Problems, vol. 7, no. 3, page 335, 1991. (Cited on page 96.)
- [Alambeigi 2017] Farshid Alambeigi, Yu Wang, Shahriar Sefati, Cong Gao, Ryan J Murphy, Iulian Iordachita, Russel H Taylor, Harpal Khanuja and Mehran Armand. *A curved-drilling approach in core decompression of the femoral head osteonecrosis using a continuum manipulator*. IEEE Robotics and Automation Letters, vol. 2, no. 3, pages 1480–1487, 2017. (Cited on page 9.)
- [Anderson 1967] Victor C Anderson and Ronald C Horn. *Tensor arm manipulator design*. Transactions of the ASME, vol. 67, no. 57, pages 1–12, 1967. (Cited on pages 9 and 13.)
- [Antman 1973] Stuart S Antman. *The theory of rods*. In Linear Theories of Elasticity and Thermoelasticity, pages 641–703. Springer, 1973. (Cited on page 25.)
- [Arai 1994] Fumihito Arai, M Ito, T Fukuda, M Negoro and T Naito. *Intelligent assistance in operation of active catheter for minimum invasive surgery*. In Robot and Human Communication, 1994. RO-MAN’94 Nagoya, Proceedings., 3rd IEEE International Workshop on, pages 192–197. IEEE, 1994. (Cited on page 28.)
- [Arienti 2013] Andrea Arienti, Marcello Calisti, Francesco Giorgio-Serchi and Cecilia Laschi. *Poseidrone: design of a soft-bodied rov with crawling, swimming and manipulation ability*. In Oceans-San Diego, 2013, pages 1–7. IEEE, 2013. (Cited on page 9.)
- [Ataollahi 2017] Asghar Ataollahi, Kaspar Althoefer, Tobias Richard Schaeffter and Kawaldeep Rhode. *Continuum manipulator*, June 13 2017. US Patent 9,675,781. (Cited on page 11.)
- [Bajo 2010] Andrea Bajo and Nabil Simaan. *Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 3666–3673. IEEE, 2010. (Cited on page 9.)
- [Bajo 2011] Andrea Bajo, Roger E Goldman and Nabil Simaan. *Configuration and joint feedback for enhanced performance of multi-segment continuum robots*. In Robotics and Automation (ICRA),

- 2011 IEEE International Conference on, pages 2905–2912. IEEE, 2011. (Cited on page 28.)
- [Bardou 2010] Berengere Bardou, Philippe Zanne, Florent Nageotte and Michel de Mathelin. *Control of a multiple sections flexible endoscopic system*. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 2345–2350. IEEE, 2010. (Cited on pages 14 and 24.)
- [Bathe 2006] Klaus-Jürgen Bathe. Finite element procedures. Klaus-Jürgen Bathe, 2006. (Cited on page 98.)
- [Bedell 2011] Chris Bedell, Jesse Lock, Andrew Gosline and Pierre E Dupont. *Design optimization of concentric tube robots based on task and anatomical constraints*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 398–403. IEEE, 2011. (Cited on page 18.)
- [Bertocchi 2006] Ulisse Bertocchi, Luca Ascari, Cesare Stefanini, Cecilia Laschi and Paolo Dario. *Sensory feedback exploitation for robot-assisted exploration of the spinal cord*. In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 601–606. IEEE, 2006. (Cited on page 28.)
- [Bosman 2015] Julien Bosman, Thor Morales Bieze, Othman Lakhall, Mario Sanz, Rochdi Merzouki and Christian Duriez. *Domain decomposition approach for FEM quasistatic modeling and control of Continuum Robots with rigid vertebrae*. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 4373–4378. IEEE, 2015. (Cited on page 62.)
- [Boudjabi 2003] Saliha Boudjabi, Antoine Ferreira and Alexandre Krupa. *Modeling and vision-based control of a micro catheter head for teleoperated in-pipe inspection*. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, volume 3, pages 4282–4287. IEEE, 2003. (Cited on page 28.)
- [Boyd 1994] Stephen Boyd, Laurent El Ghaoui, Eric Feron and Venkataramanan Balakrishnan. Linear matrix inequalities in system and control theory. SIAM, 1994. (Cited on page 79.)
- [Braganza 2006] D Braganza, DM Dawson, ID Walker and N Nath. *Neural network grasping controller for continuum robots*. In Decision and Control, 2006 45th IEEE Conference on, pages 6445–6449. IEEE, 2006. (Cited on page 20.)

- [Braganza 2007] David Braganza, Darren M Dawson, Ian D Walker and Nitendra Nath. *A neural network controller for continuum robots*. IEEE transactions on robotics, vol. 23, no. 6, pages 1270–1277, 2007. (Cited on page 20.)
- [Bunch 1971] James R Bunch and Beresford N Parlett. *Direct methods for solving symmetric indefinite systems of linear equations*. SIAM Journal on Numerical Analysis, vol. 8, no. 4, pages 639–655, 1971. (Cited on page 106.)
- [Burgner-Kahrs 2015] Jessica Burgner-Kahrs, D Caleb Rucker and Howie Choset. *Continuum robots for medical applications: A survey*. IEEE Transactions on Robotics, vol. 31, no. 6, pages 1261–1280, 2015. (Cited on pages 9, 10 and 26.)
- [Calisti 2012] Marcello Calisti, Andrea Arienti, Federico Renda, Guy Levy, Binyamin Hochner, Barbara Mazzolai, Paolo Dario and Cecilia Laschi. *Design and development of a soft robot with crawling and grasping capabilities*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 4950–4955. IEEE, 2012. (Cited on page 14.)
- [Camarillo 2008] David B Camarillo, Christopher F Milne, Christopher R Carlson, Michael R Zinn and J Kenneth Salisbury. *Mechanics modeling of tendon-driven continuum manipulators*. IEEE Transactions on Robotics, vol. 24, no. 6, pages 1262–1273, 2008. (Cited on page 11.)
- [Cengel 2006] Y Cengel and J Cimbala. Fundamentals and application. McGraw-Hill, USA, 2006. (Cited on page 97.)
- [Chikhaoui 2014] Mohamed Taha Chikhaoui, Kanty Rabenorosoa and Nicolas Andreff. *Kinematic modeling of an eap actuated continuum robot for active micro-endoscopy*. In Advances in Robot Kinematics, pages 457–465. Springer, 2014. (Cited on page 41.)
- [Chirikjian 1992] GS Chirikjian. *Theory and applications of hyper-redundant robotic mechanisms*. Ph. D thesis, Dept. of Applied Mechanics, California Institute of Technology, 1992. (Cited on pages 20, 21 and 22.)
- [Chirikjian 1993] Gregory S Chirikjian. *A continuum approach to hyper-redundant manipulator dynamics*. In Intelligent Robots and Systems’ 93, IROS’93. Proceedings of the 1993 IEEE/RSJ International Conference on, volume 2, pages 1059–1066. IEEE, 1993. (Cited on page 27.)

- [Chirikjian 1994a] Gregory S Chirikjian. *Hyper-redundant manipulator dynamics: A continuum approximation*. *Advanced Robotics*, vol. 9, no. 3, pages 217–243, 1994. (Cited on page 27.)
- [Chirikjian 1994b] Gregory S Chirikjian and Joel W Burdick. *A modal approach to hyper-redundant manipulator kinematics*. *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pages 343–354, 1994. (Cited on page 20.)
- [Chirikjian 2015] Gregory S Chirikjian. *Conformational modeling of continuum structures in robotics and structural biology: A review*. *Advanced Robotics*, no. 13, pages 817–829, 2015. (Cited on page 11.)
- [Chitrakaran 2007] Vilas K Chitrakaran, Aman Behal, Darren M Dawson and Ian D Walker. *Setpoint regulation of continuum robots using a fixed camera*. *Robotica*, vol. 25, no. 5, pages 581–586, 2007. (Cited on page 28.)
- [Cianchetti 2012] M Cianchetti, F Renda, A Licofonte and C Laschi. *Sensorization of continuum soft robots for reconstructing their spatial configuration*. In *Biomedical Robotics and Biomechanics (BioRob)*, 2012 4th IEEE RAS & EMBS International Conference on, pages 634–639. IEEE, 2012. (Cited on page 44.)
- [Cianchetti 2013] Matteo Cianchetti, Tommaso Ranzani, Giada Gerboni, Iris De Falco, Cecilia Laschi and Arianna Menciassi. *STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module*. In *Intelligent Robots and Systems (IROS)*, 2013 IEEE/RSJ International Conference on, pages 3576–3581. IEEE, 2013. (Cited on pages 9 and 17.)
- [Connolly 2015] Fionnuala Connolly, Panagiotis Polygerinos, Conor J Walsh and Katia Bertoldi. *Mechanical programming of soft actuators by varying fiber angle*. *Soft Robotics*, vol. 2, no. 1, pages 26–32, 2015. (Cited on page 18.)
- [Conrad 2013] Benjamin L Conrad, Jinwoo Jung, Ryan S Penning and Michael R Zinn. *Interleaved continuum-rigid manipulation: An augmented approach for robotic minimally-invasive flexible catheter-based procedures*. In *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, pages 718–724. IEEE, 2013. (Cited on page 9.)
- [Constantinescu 2007] Andrei Constantinescu and Alexander Korsunsky. *Elasticity with mathematica®: An introduction to con-*

- tinuum mechanics and linear elasticity. Cambridge University Press, 2007. (Cited on page 89.)
- [Davis 1994] JH Davis and RM Hirschorn. *A model for the embedded tendon control of a slender three-dimensional flexible robot link*. Dynamics and Control, vol. 4, no. 2, pages 185–208, 1994. (Cited on page 26.)
- [Dellinger 1998] Joe Dellinger, Dan Vasicek and Carl Sondergeld. *Kelvin notation for stabilizing elastic-constant inversion*. Revue de l’Institut Français du Pétrole, vol. 53, no. 5, pages 709–719, 1998. (Cited on page 93.)
- [Doyle 1982] John Doyle. *Analysis of feedback systems with structured uncertainties*. In IEE Proceedings D-Control Theory and Applications, volume 129, pages 242–250. IET, 1982. (Cited on page 81.)
- [Duriez 2013] Christian Duriez. *Control of elastic soft robots based on real-time finite element method*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 3982–3987. IEEE, 2013. (Cited on pages 45 and 50.)
- [Escande 2011] Coralie Escande, Pushparaj Mani Pathak, Rochdi Merzouki and Vincent Coelen. *Modelling of multisection bionic manipulator: Application to robotinoxt*. In Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on, pages 92–97. IEEE, 2011. (Cited on page 57.)
- [Escande 2012] Coralie Escande, Rochdi Merzouki, Pushparaj Mani Pathak and Vincent Coelen. *Geometric modelling of multisection bionic manipulator: Experimental validation on robotinoxt*. In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on, pages 2006–2011. IEEE, 2012. (Cited on pages 24 and 57.)
- [Escande 2015] Coralie Escande, Taha Chettibi, Rochdi Merzouki, Vincent Coelen and Pushparaj Mani Pathak. *Kinematic calibration of a multisection bionic manipulator*. IEEE/ASME transactions on mechatronics, vol. 20, no. 2, pages 663–674, 2015. (Cited on page 24.)
- [Fabri 1998] Andreas Fabri, Geert-Jan Giezmann, Lutz Kettner *et al.* *On the design of cgal the computational geometry algorithms library*. 1998. (Cited on page 48.)
- [Falkenhahn 2014] Valentin Falkenhahn, Tobias Mahl, Alexander Hildebrandt, Rüdiger Neumann and Oliver Sawodny. *Dynamic mod-*

- eling of constant curvature continuum robots using the Euler-Lagrange formalism*. In Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on, pages 2428–2433. IEEE, 2014. (Cited on page 28.)
- [Falkenhahn 2017] Valentin Falkenhahn, Alexander Hildebrandt, Rüdiger Neumann and Oliver Sawodny. *Dynamic control of the bionic handling assistant*. IEEE/ASME Transactions on Mechatronics, vol. 22, no. 1, pages 6–17, 2017. (Cited on page 28.)
- [Farvardin 2016] Amirhossein Farvardin, Ryan J Murphy, Robert B Grupp, Iulian Iordachita and Mehran Armand. *Towards real-time shape sensing of continuum manipulators utilizing fiber Bragg grating sensors*. In Biomedical Robotics and Biomechanics (BioRob), 2016 6th IEEE International Conference on, pages 1180–1185. IEEE, 2016. (Cited on page 44.)
- [Faure 2012] François Faure, Christian Duriez, Hervé Delingette, Jérémie Allard, Benjamin Gilles, Stéphanie Marchesseau, Hugo Talbot, Hadrien Courtecuisse, Guillaume Bousquet, Igor Peterlik et al. *Sofa: A multi-model framework for interactive physical simulation*. In Soft Tissue Biomechanical Modeling for Computer Assisted Surgery, pages 283–321. Springer, 2012. (Cited on page 50.)
- [Felippa 2000] Carlos A Felippa. *A systematic approach to the element-independent corotational dynamics of finite elements*. Technical report, Technical Report CU-CAS-00-03, Center for Aerospace Structures, 2000. (Cited on page 50.)
- [Felippa 2001] Carlos A Felippa. *Nonlinear finite element methods*. University of Colorado, Boulder, Colorado, USA, 2001. (Cited on page 101.)
- [Felippa 2005] Carlos A Felippa and Bjorn Haugen. *A unified formulation of small-strain corotational finite elements: I. Theory*. Computer Methods in Applied Mechanics and Engineering, vol. 194, no. 21, pages 2285–2335, 2005. (Cited on page 50.)
- [Festo 2012] Robotics Festo. Bionic handling assistant. <https://www.festo.com/group/en/cms/10239.htm>, 2012. (Cited on page 51.)
- [Fraś 2015] J Fraś, Jan Czarnowski, M Maciaś, J Główka, Matteo Cianchetti and Arianna Menciassi. *New STIFF-FLOP module construction idea for improved actuation and sensing*. In



- Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 2901–2906. IEEE, 2015. (Cited on pages 9 and 17.)
- [Furusho 2006] J Furusho. *Curved multi-tube systems for fetal blood sampling and treatments of organs like brain and breast*. International Journal of Computer Assisted Radiology and Surgery, vol. 1, no. 1, pages 223–226, 2006. (Cited on page 15.)
- [Gilbert 2016] Hunter B Gilbert, D Caleb Rucker and Robert J Webster III. *Concentric tube robots: The state of the art and future directions*. In Robotics Research, pages 253–269. Springer, 2016. (Cited on page 14.)
- [Giorelli 2013] Michele Giorelli, Federico Renda, Gabriele Ferri and Cecilia Laschi. *A feed-forward neural network learning the inverse kinetics of a soft cable-driven manipulator moving in three-dimensional space*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 5033–5039. IEEE, 2013. (Cited on page 19.)
- [Giorelli 2015] Michele Giorelli, Federico Renda, Marcello Calisti, Andrea Arienti, Gabriele Ferri and Cecilia Laschi. *Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature*. IEEE Transactions on Robotics, vol. 31, no. 4, pages 823–834, 2015. (Cited on page 20.)
- [Giri 2011] Nivedhitha Giri and Ian D Walker. *Three module lumped element model of a continuum arm section*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4060–4065. IEEE, 2011. (Cited on page 28.)
- [Godage 2012] Isuru S Godage, Thrishantha Nanayakkara and Darwin G Caldwell. *Locomotion with continuum limbs*. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 293–298. IEEE, 2012. (Cited on page 9.)
- [Goldman 2011] Roger E Goldman, Andrea Bajo and Nabil Simaan. *Compliant motion control for continuum robots with intrinsic actuation sensing*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 1126–1132. IEEE, 2011. (Cited on page 16.)
- [Gravagne 2001] Ian A Gravagne, Christopher D Rahn and Ian D Walker. *Good vibrations: a vibration damping setpoint controller for continuum robots*. In Robotics and Automation, 2001.

- Proceedings 2001 ICRA. IEEE International Conference on, volume 4, pages 3877–3884. IEEE, 2001. (Cited on page 12.)
- [Gravagne 2003] Ian A Gravagne, Christopher D Rahn and Ian D Walker. *Large deflection dynamics and control for planar continuum robots*. IEEE/ASME transactions on mechatronics, vol. 8, no. 2, pages 299–307, 2003. (Cited on page 14.)
- [Guglielmino 2010] Emanuele Guglielmino, Nikos Tsagarakis and Darwin G Caldwell. *An octopus anatomy-inspired robotic arm*. In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 3091–3096. IEEE, 2010. (Cited on page 14.)
- [Guglielmino 2013] Emanuele Guglielmino, Isuru Godage, Letizia Zullo and Darwin G Caldwell. *A pragmatic bio-inspired approach to the design of octopus-inspired arms*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 4577–4582. IEEE, 2013. (Cited on page 18.)
- [Hannan 2000] MW Hannan and ID Walker. *Novel kinematics for continuum robots*. In Advances in Robot Kinematics, pages 227–238. Springer, 2000. (Cited on page 22.)
- [Hartenberg 1955] Richard S Hartenberg and Jacques Denavit. *A kinematic notation for lower pair mechanisms based on matrices*. Journal of applied mechanics, vol. 77, no. 2, pages 215–221, 1955. (Cited on page 22.)
- [Hauser 2011] Helmut Hauser, Auke J Ijspeert, Rudolf M Fuchslin, Rolf Pfeifer and Wolfgang Maass. *Towards a theoretical foundation for morphological computation with compliant bodies*. Biological cybernetics, vol. 105, no. 5, pages 355–370, 2011. (Cited on page 18.)
- [Hemami 1984] Ahmad Hemami. *Design of light weight flexible robot arm*. In Robots, volume 8, pages 1623–1640, 1984. (Cited on page 14.)
- [Hintjens 2013] Pieter Hintjens. Zeromq: messaging for many applications. " O'Reilly Media, Inc.", 2013. (Cited on page 74.)
- [Hirose 2004] Shigeo Hirose and Makoto Mori. *Biologically inspired snake-like robots*. In Robotics and Biomimetics, 2004. ROBIO 2004. IEEE International Conference on, pages 1–7. IEEE, 2004. (Cited on page 8.)
- [Immega 1994] Guy Immega. *Tentacle-like manipulators with adjustable tension lines*, 1994. US Patent 5,317,952. (Cited on page 16.)

- [Immega 1995] Guy Immega and Keith Antonelli. *The KSI tentacle manipulator*. In Robotics and Automation, 1995. Proceedings., 1995 IEEE International Conference on, volume 3, pages 3149–3154. IEEE, 1995. (Cited on pages 12 and 16.)
- [Ivanescu 2003] Mircea Ivanescu, Nicu Bizdoaca and Deniela Pana. *Dynamic control for a tentacle manipulator with SMA actuators*. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, volume 2, pages 2079–2084. IEEE, 2003. (Cited on pages 28 and 67.)
- [Jiang 2012] Allen Jiang, Asghar Ataollahi, Kaspar Althoefer, Prokar Dasgupta and Thrishantha Nanayakkara. *A variable stiffness joint by granular jamming*. In ASME 2012 international design engineering technical conferences and computers and information in engineering conference, pages 267–275. American Society of Mechanical Engineers, 2012. (Cited on page 17.)
- [Johansen 2013] Tor A Johansen and Thor I Fossen. *Control allocation survey*. Automatica, vol. 49, no. 5, pages 1087–1103, 2013. (Cited on page 73.)
- [Jones 2004a] BA Jones. *Kinematics of a novel pneumatic manipulator*. Technical Report, Dept, of ECE, Clemson University, 2004. (Cited on page 16.)
- [Jones 2004b] Bryan A Jones, William McMahan and Ian Walker. *Design and analysis of a novel pneumatic manipulator*. In IFAC Symposium Advances in Automotive Control, 2004, 2004. (Cited on pages 16 and 17.)
- [Jones 2006] Bryan A Jones and Ian D Walker. *Kinematics for multisection continuum robots*. IEEE Transactions on Robotics, vol. 22, no. 1, pages 43–55, 2006. (Cited on page 24.)
- [Kang 2011] Rongjie Kang, Asimina Kazakidi, Emanuele Guglielmino, David T Branson, Dimitris P Tsakiris, John A Ekaterinaris and Darwin G Caldwell. *Dynamic model of a hyper-redundant, octopus-like manipulator for underwater applications*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 4054–4059. IEEE, 2011. (Cited on page 27.)
- [Kang 2012] Rongjie Kang, Emanuele Guglielmino, David T Branson and Darwin G Caldwell. *Bio-inspired crawling locomotion of a multi-arm octopus-like continuum system*. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 145–150. IEEE, 2012. (Cited on page 9.)

- [Kier 1985] William M Kier and Kathleen K Smith. *Tongues, tentacles and trunks: the biomechanics of movement in muscular-hydrostats*. Zoological Journal of the Linnean Society, vol. 83, no. 4, pages 307–324, 1985. (Cited on page 10.)
- [Kier 1992] William M Kier. *Hydrostatic skeletons and muscular hydrostats*. nautilus, vol. 8, no. 11, 1992. (Cited on page 9.)
- [Klute 1999] Glenn K Klute, Joseph M Czerniecki and Blake Hanaford. *McKibben artificial muscles: pneumatic actuators with biomechanical intelligence*. In Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on, pages 221–226. IEEE, 1999. (Cited on page 14.)
- [Lakhal 2016] Othman Lakhal, Achille Melingui and Rochdi Merzouki. *Hybrid approach for modeling and solving of kinematics of a compact bionic handling assistant manipulator*. IEEE/ASME Transactions on Mechatronics, vol. 21, no. 3, pages 1326–1335, 2016. (Cited on page 59.)
- [Largilliere 2015] Frederick Largilliere, Valerian Verona, Eulalie Coevoet, Mario Sanz-Lopez, Jeremie Dequidt and Christian Duriez. *Real-time control of soft-robots using asynchronous finite element modeling*. In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 2550–2555. IEEE, 2015. (Cited on pages 45 and 48.)
- [Laschi 2009] Cecilia Laschi, Barbara Mazzolai, V Mattoli, M Cianchetti and P Dario. *Design of a biomimetic robotic octopus arm*. Bioinspiration & Biomimetics, vol. 4, no. 1, page 015006, 2009. (Cited on page 14.)
- [Li 2017] Jinglin Li, Zhou Teng and Jing Xiao. *Can a continuum manipulator fetch an object in an unknown cluttered space?* IEEE Robotics and Automation Letters, vol. 2, no. 1, pages 2–9, 2017. (Cited on page 9.)
- [Lofberg 2004] Johan Lofberg. *YALMIP: A toolbox for modeling and optimization in MATLAB*. In Computer Aided Control Systems Design, 2004 IEEE International Symposium on, pages 284–289. IEEE, 2004. (Cited on page 79.)
- [Lyapunov 1992] Aleksandr Mikhailovich Lyapunov. *The general problem of the stability of motion*. International Journal of Control, vol. 55, no. 3, pages 531–534, 1992. (Cited on page 78.)
- [Maghooa 2015] Farahnaz Maghooa, Agostino Stilli, Yohan Noh, Kaspar Althoefer and Helge A Wurdemann. *Tendon and pressure*

- actuation for a bio-inspired manipulator based on an antagonistic principle.* In Robotics and Automation (ICRA), 2015 IEEE International Conference on, pages 2556–2561. IEEE, 2015. (Cited on page 16.)
- [Mahl 2014] Tobias Mahl, Alexander Hildebrandt and Oliver Sawodny. *A variable curvature continuum kinematics for kinematic control of the bionic handling assistant.* IEEE transactions on robotics, vol. 30, no. 4, pages 935–949, 2014. (Cited on page 24.)
- [Mahoney 2016] Arthur W Mahoney, Patrick L Anderson, Philip J Swaney, Fabien Maldonado and Robert J Webster. *Reconfigurable parallel continuum robots for incisionless surgery.* In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on, pages 4330–4336. IEEE, 2016. (Cited on page 9.)
- [Mahvash 2010] Mohsen Mahvash and Pierre E Dupont. *Stiffness control of a continuum manipulator in contact with a soft environment.* In Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on, pages 863–870. IEEE, 2010. (Cited on page 16.)
- [Mahvash 2011] Mohsen Mahvash and Pierre E Dupont. *Stiffness control of surgical continuum manipulators.* IEEE Transactions on Robotics, vol. 27, no. 2, pages 334–345, 2011. (Cited on page 16.)
- [McMahan 2005] William McMahan, Bryan A Jones and Ian D Walker. *Design and implementation of a multi-section continuum robot: Air-Octor.* In Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on, pages 2578–2585. IEEE, 2005. (Cited on pages 8 and 16.)
- [McMahan 2006] William McMahan, V Chitrakaran, M Csencsits, D Dawson, Ian D Walker, Bryan A Jones, M Pritts, D Dienno, M Grissom and Christopher D Rahn. *Field trials and testing of the OctArm continuum manipulator.* In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on, pages 2336–2341. IEEE, 2006. (Cited on pages 14 and 15.)
- [Mehling 2006] Joshua S Mehling, Myron A Diftler, Mars Chu and Michael Valvo. *A minimally invasive tendril robot for in-space inspection.* In Biomedical Robotics and Biomechatronics, 2006. BioRob 2006. The First IEEE/RAS-EMBS International Con-

- ference on, pages 690–695. IEEE, 2006. (Cited on pages 8, 9 and 14.)
- [Melingui 2014] Achille Melingui, Rochdi Merzouki, Jean Bosco Mbede, Coralie Escande, Boubaker Daachi and Nabil Benoudjit. *Qualitative approach for inverse kinematic modeling of a compact bionic handling assistant trunk*. In Neural Networks (IJCNN), 2014 International Joint Conference on, pages 754–761. IEEE, 2014. (Cited on page 20.)
- [Melingui 2017] Achille Melingui, Joseph Jean-Baptiste Mvogo Ahanda, Othman Lakhal, Jean Bosco Mbede and Rochdi Merzouki. *Adaptive Algorithms for Performance Improvement of a Class of Continuum Manipulators*. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2017. (Cited on pages 20 and 28.)
- [Mochiyama 2002] Hiromi Mochiyama and Takahiro Suzuki. *Dynamical modelling of a hyper-flexible manipulator*. In SICE 2002. Proceedings of the 41st SICE Annual Conference, volume 3, pages 1505–1510. IEEE, 2002. (Cited on page 27.)
- [Mochiyama 2003] Hiromi Mochiyama and Takahiro Suzuki. *Kinematics and dynamics of a cable-like hyper-flexible manipulator*. In Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on, volume 3, pages 3672–3677. IEEE, 2003. (Cited on page 22.)
- [Morari 1989] Manfred Morari and Evangelos Zafriou. Robust process control. Prentice hall Englewood Cliffs, NJ, 1989. (Cited on page 81.)
- [Murphy 2014] Ryan J Murphy, Michael DM Kutzer, Sean M Segreti, Blake C Lucas and Mehran Armand. *Design and kinematic characterization of a surgical manipulator with a focus on treating osteolysis*. Robotica, vol. 32, no. 6, pages 835–850, 2014. (Cited on page 44.)
- [Neppalli 2007] Srinivas Neppalli, B Jones, William McMahan, V Chitrakaran, I Walker, M Pritts, M Csencsits, C Rahn and M Grisom. *OctArm-A soft robotic manipulator*. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2569–2569, 2007. (Cited on page 8.)
- [Nickel 1963] Vernon L Nickel, Jacquelin Perry and Alice L Garrett. *Development of Useful Function in the Severely Paralyzed Hand*. JBJS, vol. 45, no. 5, pages 933–952, 1963. (Cited on page 14.)



- [Noh 2014] Yohan Noh, Emanuele Lindo Secco, Sina Sareh, Helge Würdemann, Angela Faragasso, Junghwan Back, Hongbin Liu, Elizabeth Sklar and Kaspar Althoefer. *A continuum body force sensor designed for flexible surgical robotics devices*. In Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE, pages 3711–3714. IEEE, 2014. (Cited on page 17.)
- [Orekhov 2016] Andrew L Orekhov, Caroline B Black, John Till, Scotty Chung and D Caleb Rucker. *Analysis and validation of a teleoperated surgical parallel continuum manipulator*. IEEE Robotics and Automation Letters, vol. 1, no. 2, pages 828–835, 2016. (Cited on page 9.)
- [Otake 2014] Y Otake, RJ Murphy, MD Kutzer, RH Taylor and M Armand. *Piecewise-rigid 2D-3D registration for pose estimation of snake-like manipulator using an intraoperative x-ray projection*. In Proc. SPIE Medical Imaging, volume 9036, pages 90360Q–90360Q, 2014. (Cited on page 44.)
- [Paul 2006] Chandana Paul. *Morphological computation: A basis for the analysis of morphology and control requirements*. Robotics and Autonomous Systems, vol. 54, no. 8, pages 619–630, 2006. (Cited on page 18.)
- [Penning 2011] Ryan S Penning, Jinwoo Jung, Justin A Borgstadt, Nicola J Ferrier and Michael R Zinn. *Towards closed loop control of a continuum robotic manipulator for medical applications*. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 4822–4827. IEEE, 2011. (Cited on pages 28 and 67.)
- [Penning 2012] Ryan S Penning, Jinwoo Jung, Nicola J Ferrier and Michael R Zinn. *An evaluation of closed-loop control options for continuum manipulators*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 5392–5397. IEEE, 2012. (Cited on pages 28 and 67.)
- [Pfeifer 2009] Rolf Pfeifer and Gabriel Gómez. *Morphological computation—connecting brain, body, and environment*. Creating Brain-Like Intelligence, pages 66–83, 2009. (Cited on page 18.)
- [Pritts 2004] Michael B Pritts and Christopher D Rahn. *Design of an artificial muscle continuum robot*. In Robotics and Automation, 2004. Proceedings. ICRA’04. 2004 IEEE International Confer-

- ence on, volume 5, pages 4742–4746. IEEE, 2004. (Cited on page 14.)
- [Przemieniecki 1985] Janusz S Przemieniecki. *Theory of matrix structural analysis*. Courier Corporation, 1985. (Cited on page 46.)
- [Qi 2014] Peng Qi, Chuang Liu, Linan Zhang, Shuxin Wang, Hak-Keung Lam and Kaspar Althoefer. *Fuzzy logic control of a continuum manipulator for surgical applications*. In Robotics and Biomimetics (ROBIO), 2014 IEEE International Conference on, pages 413–418. IEEE, 2014. (Cited on page 28.)
- [Qi 2016] Peng Qi, Chuang Liu, Ahmad Ataka, Hak-Keung Lam and Kaspar Althoefer. *Kinematic control of continuum manipulators using a fuzzy-model-based approach*. *IEEE Transactions on Industrial Electronics*, vol. 63, no. 8, pages 5022–5035, 2016. (Cited on page 28.)
- [Qu 2016] Tingyu Qu, Jie Chen, Shen Shen, Zhen Xiao, Zhe Yue and Henry YK Lau. *Motion control of a bio-inspired wire-driven multi-backbone continuum minimally invasive surgical manipulator*. In Robotics and Biomimetics (ROBIO), 2016 IEEE International Conference on, pages 1989–1995. IEEE, 2016. (Cited on page 9.)
- [Rabiner 1975] Lawrence R Rabiner and Bernard Gold. *Theory and application of digital signal processing*. Englewood Cliffs, NJ, Prentice-Hall, Inc., 1975. 777 p., 1975. (Cited on page 74.)
- [Reddy 1993] Junuthula Narasimha Reddy. *An introduction to the finite element method*, volume 2. McGraw-Hill New York, 1993. (Cited on page 99.)
- [Reddy 2013] Junuthula Narasimha Reddy. *An introduction to continuum mechanics*. Cambridge university press, 2013. (Cited on pages 70 and 89.)
- [Robinson 1999] G Robinson and J Bruce C Davies. *Continuum robots-a state of the art*. In Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on, volume 4, pages 2849–2854, 1999. (Cited on page 8.)
- [Rolf 2010] Matthias Rolf, Jochen J Steil and Michael Gienger. *Goal babbling permits direct learning of inverse kinematics*. *IEEE Transactions on Autonomous Mental Development*, vol. 2, no. 3, pages 216–229, 2010. (Cited on page 19.)



- [Rolf 2012] Matthias Rolf and Jochen J Steil. *Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant*. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 3440–3446. IEEE, 2012. (Cited on pages 12 and 14.)
- [Rolf 2014] Matthias Rolf and Jochen J Steil. *Efficient exploratory learning of inverse kinematics on a bionic elephant trunk*. IEEE transactions on neural networks and learning systems, vol. 25, no. 6, pages 1147–1160, 2014. (Cited on page 19.)
- [Rucker 2010a] D Caleb Rucker, Bryan A Jones and Robert J Webster. *A model for concentric tube continuum robots under applied wrenches*. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 1047–1052. IEEE, 2010. (Cited on page 26.)
- [Rucker 2010b] D Caleb Rucker, Robert J Webster III, Gregory S Chirikjian and Noah J Cowan. *Equilibrium conformations of concentric-tube continuum robots*. The International journal of robotics research, vol. 29, no. 10, pages 1263–1280, 2010. (Cited on page 26.)
- [Rus 2015] Daniela Rus and Michael T Tolley. *Design, fabrication and control of soft robots*. Nature, vol. 521, no. 7553, pages 467–475, 2015. (Cited on page 18.)
- [Sadeghi 2012] Ali Sadeghi, Lucia Beccai and Barbara Mazzolai. *Innovative soft robots based on electro-rheological fluids*. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on, pages 4237–4242. IEEE, 2012. (Cited on pages 17 and 41.)
- [Sareh 2014] Sina Sareh, Allen Jiang, Angela Faragasso, Yohan Noh, Thrishantha Nanayakkara, Prokar Dasgupta, Lakmal D Seneviratne, Helge A Wurdemann and Kaspar Althoefer. *Bio-inspired tactile sensor sleeve for surgical soft manipulators*. In Robotics and Automation (ICRA), 2014 IEEE International Conference on, pages 1454–1459. IEEE, 2014. (Cited on page 17.)
- [Searle 2013] Thomas C Searle, Kaspar Althoefer, Lakmal Seneviratne and Hongbin Liu. *An optical curvature sensor for flexible manipulators*. In Robotics and Automation (ICRA), 2013 IEEE International Conference on, pages 4415–4420. IEEE, 2013. (Cited on page 44.)

- [Sears 2006] Patrick Sears and Pierre Dupont. *A steerable needle technology using curved concentric tubes*. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 2850–2856. IEEE, 2006. (Cited on page 14.)
- [Segreti 2012] Sean M Segreti, Michael DM Kutzer, Ryan J Murphy and Mehran Armand. *Cable length estimation for a compliant surgical manipulator*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 701–708. IEEE, 2012. (Cited on page 44.)
- [Shiva 2016] Ali Shiva, Agostino Stilli, Yohan Noh, Angela Faragasso, Iris De Falco, Giada Gerboni, Matteo Cianchetti, Arianna Menciassi, Kaspar Althoefer and Helge A Wurdemann. *Tendon-based stiffening for a pneumatically actuated soft manipulator*. IEEE Robotics and Automation Letters, vol. 1, no. 2, pages 632–637, 2016. (Cited on pages 16 and 85.)
- [Siciliano 2016] Bruno Siciliano and Oussama Khatib. Springer handbook of robotics. Springer, 2016. (Cited on page 7.)
- [Simaan 2004] Nabil Simaan, Russell Taylor and Paul Flint. *A dexterous system for laryngeal surgery*. In Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on, volume 1, pages 351–357. IEEE, 2004. (Cited on page 14.)
- [Simaan 2009] Nabil Simaan, Kai Xu, Wei Wei, Ankur Kapoor, Peter Kazanzides, Russell Taylor and Paul Flint. *Design and integration of a telerobotic system for minimally invasive surgery of the throat*. The International journal of robotics research, vol. 28, no. 9, pages 1134–1153, 2009. (Cited on page 11.)
- [Steltz 2010] E Steltz, A Mozeika, J Rembisz, N Corson and HM Jaeger. *Jamming as an enabling technology for soft robotics*. In SPIE Smart Structures and Materials+ Nondestructive Evaluation and Health Monitoring, pages 764225–764225. International Society for Optics and Photonics, 2010. (Cited on page 17.)
- [Sturm 1999] Jos F Sturm. *Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones*. Optimization methods and software, vol. 11, no. 1-4, pages 625–653, 1999. (Cited on page 79.)
- [Su 2012] Hao Su, Diana C Cardona, Weijian Shang, Alexander Camilo, Gregory A Cole, D Caleb Rucker, Robert J Webster and Gregory S Fischer. *A MRI-guided concentric tube continuum robot*

- with piezoelectric actuation: a feasibility study.* In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 1939–1945. IEEE, 2012. (Cited on pages 15 and 41.)
- [Tatlicioglu 2007] Enver Tatlicioglu, Ian D Walker and Darren M Dawson. *New dynamic models for planar extensible continuum robot manipulators.* In Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, pages 1485–1490. IEEE, 2007. (Cited on page 27.)
- [Tonapi 2014] Manas M Tonapi, Isuru S Godage and Ian D Walker. *Next generation rope-like robot for in-space inspection.* In Aerospace Conference, 2014 IEEE, pages 1–13. IEEE, 2014. (Cited on page 9.)
- [Trivedi 2008a] Deepak Trivedi, Amir Lotfi and Christopher D Rahn. *Geometrically exact models for soft robotic manipulators.* IEEE Transactions on Robotics, vol. 24, no. 4, pages 773–780, 2008. (Cited on page 26.)
- [Trivedi 2008b] Deepak Trivedi, Christopher D Rahn, William M Kier and Ian D Walker. *Soft robotics: Biological inspiration, state of the art, and future research.* Applied Bionics and Biomechanics, vol. 5, no. 3, pages 99–117, 2008. (Cited on pages 10 and 33.)
- [Walker 1999] Ian D Walker and Michael W Hannan. *A novel ‘elephant’s trunk’ robot.* In Advanced Intelligent Mechatronics, 1999. Proceedings. 1999 IEEE/ASME International Conference on, pages 410–415. IEEE, 1999. (Cited on page 13.)
- [Walker 2005] Ian D Walker, Darren M Dawson, Tamar Flash, Frank W Grasso, Roger T Hanlon, Binyamin Hochner, William M Kier, Christopher C Pagano, Christopher D Rahn and Qiming M Zhang. *Continuum robot arms inspired by cephalopods.* In Defense and Security, pages 303–314. International Society for Optics and Photonics, 2005. (Cited on page 14.)
- [Walker 2006] Ian D Walker, Carlos Carreras, Robin McDonnell and George Grimes. *Extension versus bending for continuum robots.* International Journal of Advanced Robotic Systems, vol. 3, no. 2, page 26, 2006. (Cited on page 16.)
- [Walker 2013a] Ian D Walker. *Continuous backbone continuum robot manipulators.* ISRN Robotics, vol. 2013, 2013. (Cited on pages 13, 24 and 33.)

- [Walker 2013b] Ian D Walker. *Robot strings: long, thin continuum robots*. In Aerospace Conference, 2013 IEEE, pages 1–12. Ieee, 2013. (Cited on page 16.)
- [Webster 2006] Robert J Webster, Allison M Okamura and Nah J Cowan. *Toward active cannulas: Miniature snake-like surgical robots*. In Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on, pages 2857–2863. IEEE, 2006. (Cited on page 26.)
- [Wenlong 2013] Yang Wenlong, Dong Wei and Du Zhijiang. *Mechanics-based kinematic modeling of a continuum manipulator*. In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on, pages 5052–5058. IEEE, 2013. (Cited on page 26.)
- [Wilkening 2017] Paul Wilkening, Farshid Alambeigi, Ryan J Murphy, Russell H Taylor and Mehran Armand. *Development and experimental evaluation of concurrent control of a robotic arm and continuum manipulator for osteolytic lesion treatment*. IEEE Robotics and Automation Letters, vol. 2, no. 3, pages 1625–1631, 2017. (Cited on page 9.)
- [Wortman 1965] JJ Wortman and RA Evans. *Young’s modulus, shear modulus, and Poisson’s ratio in silicon and germanium*. Journal of applied physics, vol. 36, no. 1, pages 153–156, 1965. (Cited on page 95.)
- [Wu 2017] Liao Wu, Ross Crawford and Jonathan Roberts. *Dexterity Analysis of Three 6-DOF Continuum Robots Combining Concentric Tube Mechanisms and Cable-Driven Mechanisms*. IEEE Robotics and Automation Letters, vol. 2, no. 2, pages 514–521, 2017. (Cited on page 15.)
- [Yamada 2014] Akimasa Yamada, Shigeyuki Naka, Shigehiro Morikawa and Tomoyuki Tani. *MRI compatible continuum robot based on closed elastica with bending and twisting*. In Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3187–3192, 2014. (Cited on page 8.)
- [Yip 2014] Michael C Yip and David B Camarillo. *Model-less feedback control of continuum manipulators in constrained environments*. IEEE Transactions on Robotics, vol. 30, no. 4, pages 880–889, 2014. (Cited on pages 28 and 67.)
- [Yip 2016] Michael C Yip and David B Camarillo. *Model-less hybrid position/force control: a minimalist approach for continuum manipulators in unknown, constrained environments*. IEEE

- Robotics and Automation Letters, vol. 1, no. 2, pages 844–851, 2016. (Cited on page 28.)
- [Yoon 2011] Hyun-Soo Yoon, Se Min Oh, Jin Hyeok Jeong, Seung Hwan Lee, Kyung Tae, Kyoung-Chul Koh and Byung-Ju Yi. *Active bending endoscope robot system for navigation through sinus area*. In Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on, pages 967–972. IEEE, 2011. (Cited on page 14.)
- [Yu 2015] Yan Yu, Peng Qi, Kaspar Althoefer and Hak-Keung Lam. *Lagrangian dynamics and nonlinear control of a continuum manipulator*. In Robotics and Biomimetics (ROBIO), 2015 IEEE International Conference on, pages 1912–1917. IEEE, 2015. (Cited on page 28.)
- [Zhao 2010] Qiang Zhao and Fang Gao. *Design and analysis of a kind of biomimetic continuum robot*. In Robotics and Biomimetics (ROBIO), 2010 IEEE International Conference on, pages 1316–1320. IEEE, 2010. (Cited on pages 8 and 14.)
- [Zheng 2012] Tianjiang Zheng, David T Branson, Rongjie Kang, Matteo Cianchetti, Emanuele Guglielmino, Maurizio Follador, Gustavo A Medrano-Cerda, Isuru S Godage and Darwin G Caldwell. *Dynamic continuum arm model for use with underwater robotic manipulators inspired by octopus vulgaris*. In Robotics and Automation (ICRA), 2012 IEEE International Conference on, pages 5289–5294. IEEE, 2012. (Cited on page 28.)
- [Zheng 2014] Tianjiang Zheng, Yawei Yang, David T. Branson, Rongjie Kang, Emanuele Guglielmino, Matteo Cianchetti, Darwin G. Caldwell and Guilin Yang. *Control design of shape memory alloy based multi-arm continuum robot inspired by octopus*. In Proc. IEEE 9th Conference on Industrial Electronics and Applications, pages 1108–1113, 2014. (Cited on page 8.)