Workspace Estimation and Design Optimization of Soft Robots

Walid Amehri

CENTRALE LILLE

THESE

Présentée en vue
d'obtenir le grade de

DOCTEUR

En

Spécialité : Automatique, Génie Informatique,

Traitement du Signal et des Images

Par

Walid AMEHRI

DOCTORAT DELIVRE PAR CENTRALE LILLE

Titre de la thèse :

Estimation de l'Espace de Travail et l'Optimisation de la Conception des Robots Déformables.

Workspace Estimation and Design Optimization of Soft Robots.

Soutenue le 28 Février 2022 devant le jury d'examen :

Président	Philippe Fraisse	Professeur, Université de Montpellier
Rapporteur	Sébastien Briot	Chargé de Recherche CNRS, HDR, Centrale Nantes
Rapporteur	Guillaume Laurent	Maître de Conférences, HDR, ENSMM
Examinateur	Philippe Fraisse	Professeur, Université de Montpellier
Examinateur	Jessica Burgner-Kahrs	Associate Professor, Université de Toronto
Examinateur	Pierre Renaud	Professeur, INSA Strasbourg
Directeur de thèse	Gang Zheng	Chargé de Recherche INRIA, HDR, INRIA Lille
Co-Directeur de thèse	Alexandre Kruszewski	Professeur, Centrale Lille

Thèse préparée dans le Centre de Recherche en Informatique, Signal et Automatique de Lille,
CRIStAL, CNRS UMR 9189

Ecole Doctorale MADIS-631

Contents

Acknowledgments

First of all, I would like to express my thanks to my two supervisors, Gang Zheng and Alexandre Kruszewski for presenting me with the opportunity to work on such an interesting subject, and also for their guidance, help, and sharing their experience with me. I am grateful.

Thanks to the Region Hauts-de-France, the Project Inventor (I-SITE ULNE, le programme d'Investissements d'Avenir, Métropole Européenne de Lille), and the Project ROBOCOP [ANR PRCE 19 CE19] for the financial supporting of this thesis.

Many thanks to all members of DEFROST team, and especially Ke Wu for the help with the platform. Thanks to Shijie Li for his help with the validation part.

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

I would like also to express my gratitude to my family and friends for their care and support.

Nomenclature

Symbol \ Model	**Piece-wise Constant Strain**	**Finite Element Method**
DoFs	Degrees of Freedom	
$\setminus$	Exclusion	
$\otimes$	Kronecker product	
$\dot{}$	Derivative with respect to time	
$'$	Derivative with respect to space	—
$\widehat{}$	Converts $\mathbb{R}^6$ in $\mathfrak{se}(3)$	—
$\widetilde{}$	Converts $\mathbb{R}^3$ in $\mathfrak{so}(3)$	—
n	Total number of DoFs	
N	Total number of segments/bodies	Dimension of the finite element mesh
n_τ	Actuation force vector (Input) dimension	
$n_u = 3$	End-effector (Output) dimension	
q	$= \left[q_1^T \cdots q_N^T\right]^T \in \mathbb{R}^n$ Generalized strain vector	$\in \mathbb{R}^{3n}$ Displacement vector
τ	$= \left[\tau_1 \cdots \tau_{n_\tau}\right]^T \in \mathbb{R}^{n_\tau}$ Soft robot actuators force vector.	
K	$= K \in \mathbb{R}^{n \times n}$ Generalized stiffness matrix	$= K(q) \in \mathbb{R}^{3n \times 3n}$ Generalized stiffness matrix
$H(q)$	$\in \mathbb{R}^{n \times n_\tau}$ Generalized actuation matrix	$\in \mathbb{R}^{3n \times n_\tau}$ Generalized actuation matrix
$F(q)$	$\in \mathbb{R}^n$ Generalized external force vector	$\in \mathbb{R}^{3n}$ Generalized external force vector
$\delta\tau$	Actuation vector with respect to prior configuration.	
δq	Generalized strain vector with respect to prior configuration	Displacement vector with respect to prior configuration
$\underline{\tau}$	$= \left[\underline{\tau}_1 \cdots \underline{\tau}_{n_\tau}\right]^T$ Minimum actuators bound vector	
$\overline{\tau}$	$= \left[\overline{\tau}_1 \cdots \overline{\tau}_{n_\tau}\right]^T$ Maximum actuators bound vector	
$\mathcal{T}$	$= \left[\underline{\tau}_1, \overline{\tau}_1\right] \times \cdots \times \left[\underline{\tau}_{n_\tau}, \overline{\tau}_{n_\tau}\right]$ Actuators bounds	
u	$\in \mathbb{R}^{n_u}$ End-effector position vector in the inertial frame.	
$\mathcal{W}_E$	$\in \mathbb{R}^{n_u}$ Workspace of a soft robot	
$\partial\mathcal{W}_E$	$\in \mathbb{R}^{n_u - 1}$ Soft robot workspace boundary	
$\mathcal{W}_{E_d}$	Discretized space of the current configuration	
$\mathcal{W}_{E_s}$	Feasible small displacement neighborhood space around the current configuration	
$\mathcal{B}$	$= [\underline{\mathcal{B}}, \overline{\mathcal{B}}]$ Lower $\underline{\mathcal{B}}$ and upper $\overline{\mathcal{B}}$ bounds of a domain $\mathcal{B}$	
R	$\in SO(3)$ Orientation matrix	—
X	$\in [0, L] \subset \mathbb{R}$ Abscissa along the robot arm	—

Continued on next page

Table 1 – continued from previous page

Symbol \ Model	Piece-wise Constant Strain	Finite Element Method
g	$(X) = \begin{pmatrix} R & u \\ 0^T & 1 \end{pmatrix} \in SE(3)$: Configuration matrix	—
v, w	$(X) \in \mathbb{R}^3$ Linear and angular velocity	—
p, k	$(X) \in \mathbb{R}^3$ Linear and angular strain	—
$\hat{\eta}$	$(X) = \begin{pmatrix} \tilde{\mathsf{w}} & \mathsf{v} \\ 0^T & 0 \end{pmatrix} \in \mathfrak{se}(3)$: Velocity twist matrix	—
η	$(X) = \left(\mathsf{v}^T, \mathsf{w}^T\right)^T \in \mathbb{R}^6$ Velocity vector	—
$\hat{\xi}$	$(X) = \begin{pmatrix} \tilde{\mathsf{k}} & \mathsf{p} \\ 0^T & 0 \end{pmatrix} \in \mathfrak{se}(3)$: Strain twist matrix	—
ξ	$(X) = \left(\mathsf{k}^T, \mathsf{p}^T\right)^T \in \mathbb{R}^6$ Strain vector	—
$\mathrm{ad}_{\xi,\eta}$	$(X) = \begin{pmatrix} \widetilde{\mathsf{k}, \mathsf{w}} & 0_{3\times3} \\ \widetilde{\mathsf{q}, \mathsf{v}} & \widetilde{\mathsf{k}, \mathsf{w}} \end{pmatrix} \in \mathbb{R}^{6\times6}$: Adjoint representation of the strain	—
$\mathrm{ad}^*_{\xi,\eta}$	$(X) = \begin{pmatrix} \widetilde{\mathsf{k}, \mathsf{w}} & \widetilde{\mathsf{q}, \mathsf{v}} \\ 0_{3\times3} & \widetilde{\mathsf{k}, \mathsf{w}} \end{pmatrix} \in \mathbb{R}^{6\times6}$: coAdjoint representation of the strain	—
Ad_g	$(X) = \begin{pmatrix} R & 0_{3\times3} \\ \tilde{\mathsf{u}}R & R \end{pmatrix} \in \mathbb{R}^{6\times6}$: Adjoint representation of g	—
Ad^*_g	$(X) = \begin{pmatrix} R & \tilde{\mathsf{u}}R \\ 0_{3\times3} & R \end{pmatrix} \in \mathbb{R}^{6\times6}$: coAdjoint representation of g	—
T_g	$(X) = \int_0^X \mathrm{Ad}_g$: Tangent operator of the exponential map	—
$e^{X\hat{\xi}}$	Exponential map of $X\widehat{\xi} \in SE(3)$ with $\theta^2 = \mathsf{k}^T\mathsf{k}$	—
α	$= \left[\alpha_1 \cdots \alpha_{n_\tau}\right]^T \in \mathbb{R}^{n_\tau}$: Actuators length parameter	—
$\underline{\alpha}$	$= \left[\underline{\alpha}_1 \cdots \underline{\alpha}_{n_\tau}\right]^T$: Minimum actuators length bounds	—
$\overline{\alpha}$	$= \left[\overline{\alpha}_1 \cdots \overline{\alpha}_{n_\tau}\right]^T$: Maximum actuators length bounds	—
$\mathcal{A}$	$= [\underline{\alpha}_1, \overline{\alpha}_1] \times \cdots \times \left[\underline{\alpha}_{n_\tau}, \overline{\alpha}_{n_\tau}\right]$: Actuators length bounds	—
L	$= \left[L_1 \cdots L_N\right]^T \in \mathbb{R}^N$: Segments length parameter	—
$\underline{L}$	$= \left[\underline{L}_1 \cdots \underline{L}_N\right]^T$: Minimum segments length bounds	—

Continued on next page

Table 1 – continued from previous page

Symbol \ Model	**Piece-wise Constant Strain**	**Finite Element Method**
$\overline{L}$	$= \left[\overline{L}_1 \cdots \overline{L}_N\right]^T$: Maximum segments length bounds	—
$\mathcal{L}$	$= \left[\underline{L}_1, \overline{L}_1\right] \times \cdots \times \left[\underline{L}_N, \overline{L}_N\right]$: Segments length bounds	—

Table 1: Acronyms and Glossary for the adopted mathematical modeling methods.

Chapter 1

Introduction

1.1 General Introduction

1.1.1 Robotics: Definitions and Classes

In an etymological sense, and according to [137], the word robotics originated from the word robot, which is derived from the Czech noun robota meaning labor. Its original appearance was in the RUR play of the first fictional automatons called robots published in 1920 [137]. Scientifically, robotics is a multidisciplinary field that aims at the design, modeling, control, and applications of robots for the purpose of increasing productivity and assisting human beings [32].

In accordance with [44, 52, 121], robots can be classified based on their fabrication material and their degrees of freedom. Consistently, there are two main classes of robots (as shown by Fig. 1.1): rigid and hyper-redundant (it is also important to state that some rigid robots can also be hyper-redundant, e.g., serpentine robots).

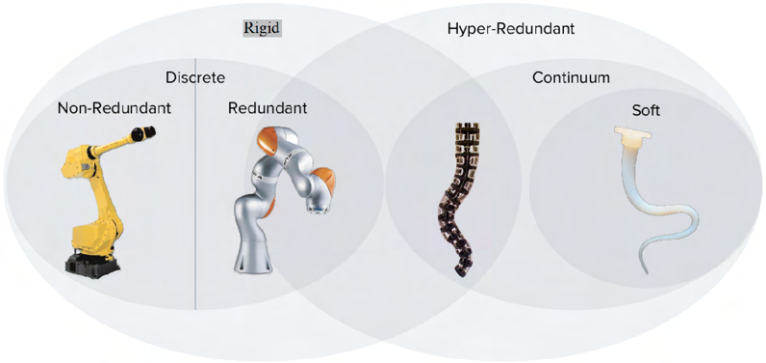


Figure 1.1: Classification of robots [44, 52, 121].

1.1.2 Rigid Robotics and the Rise of a New Robot Class

Rigid robots are a class of robots made of rigid materials with a Young's modulus greater than $10^9 Pa$ [72]

The first appearance of a rigid robot can be traced back to the third century B.C. and earlier by the mechanical engineer Yan Shi that created the first human-shaped figure of his mechanical

handiwork [85]. The first mechanical humanoid robot Mechanical Knight was designed by Leonardo da Vinci (see A in Fig. 1.2). Mechanical robots have then continued to emerge until 1738 where a mechanical duck called Digesting Duck that was able to eat and flap its wings was created by the inventor and artist Jacques de Vaucanson (see B in Fig. 1.2). Following the birth of electricity, robotics have evolved first via the Teleautomaton device that was created in 1898 by Nicola Tesla, which was a radio-controlled vessel (see C in Fig. 1.2), followed by the first electronic humanoid robot Elektro in 1930 and created by the Westinghouse Electric Corporation (see D in Fig. 1.2). The first appearance of rigid robots imitating biological behavior is traced back to 1948 by William Grey Walter [112] (see E in Fig. 1.2). Next, the first commercial robot Unimate was created in 1956 by the Unimation company [126], the same company that introduced the first installed industrial robot in 1961 (see F in Fig. 1.2). After that, the first full-scale humanoid robot WABOT-1 [67] was created by the Waseda University (see G in Fig. 1.2). Then, the famous six electro-mechanically driven axis robot Famulus [115] (see H in Fig. 1.2) was developed by the KUKA Robot Group in 1973. Next, in 1975, a programmable universal manipulation arm PUMA was created by Victor Scheinman (see I in Fig. 1.2). In 1983, the creation of the first parallel programming language used for the robot's control was performed by Stevo Bozinovski and Mihail Sestakov.

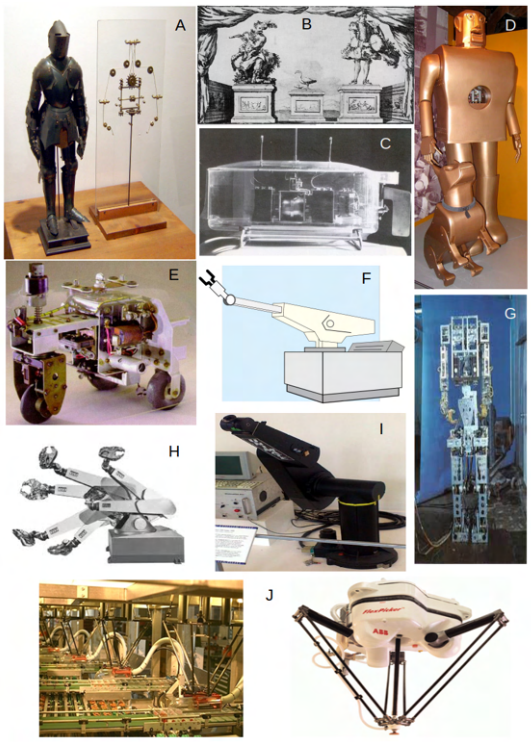


Figure 1.2: Rigid Robots Evolution: A- Mechanical Knight. B- Digesting Duck. C- Teleautomaton. D- Elektro. E- Elsie and Elmer [112]. F- Unimate [126]. G- WABOT-1 [67]. H- Famulus [115]. I- PUMA. J- Delta and ABB Flex-Pitcher robots [41, 43, 51].

Since then, researchers made a huge progress on rigid robotics, and owing to their stiff material and high-torque joints which make them exceptionally stable and accurate, rigid robots have become widely and successfully deployed especially in the industrial environment [120] and

have been proved to increase productivity (e.g., Delta and ABB Flex-Pitcher robots in J of Fig. 1.2) [41, 43, 51].

However, as new and more complex applications arise, traditional rigid robots have shown many drawbacks due to their high stiffness, lack of compliance in conventional actuation mechanisms, and limited degrees of freedom [42]. Moreover, rigid robots have been considered less practical when performing operations in dynamic environments [30] and not safe when interacting in fragile environments and also with humans [32, 131].

In the last decades, as a consequence of these drawbacks, many researchers have been motivated to seek a novel design of robots that can be flexible and safe in order to cope with new applications, leading to the rise of soft robotics.

1.1.3 Soft Robotics

Hyper-redundant robots consist of a large number of DoFs [121]. Continuum robots are capable of continuous deformations, but not all continuum robots are necessarily soft [121]. Soft robots are a sub-class of continuum robots that are hyper-redundant robots [82].

In a general sense, soft robotics is a branch of robotics that consists of the design, modeling, and control of robots that are fabricated from flexible and compliant materials equivalent to those found in living organisms such as silicone rubber and other materials that can be readily found in nature [65, 70, 111, 121].

The history of soft robotics is provided in detail in [59, 70, 88, 111, 125]. In the following, we present only the main points of the history of soft robotics. The first appearance of soft robots is traced back to the late 1970s where robot grippers based on granular materials were first published in [28, 90, 113]. After, a continuously-deforming body using elastomers was controlled by a pneumatic actuation in 1984 [128]. Next, robot grippers using electrorheological fluid were proposed in 1989 [64]. In 1991, [118] developed a flexible micro-actuator and later extended its applications to soft robotic grasping and also mobile mechanisms [117, 119] (see A in Fig. 1.3).

Ever since the field of soft robotics continued to make outstanding progress and has gained impressive attention at the beginning of the 21st century. Consequently, a huge portion of new soft material has been synthesized and made commercially accessible [13, 59, 65, 68, 70, 88, 111, 121], diverse fabrication techniques for soft robots have been developed and made available [13, 59, 65, 68, 70, 88, 111, 121], a growing number of works showing the use of soft robotics has been published in high-profile scientific journals [59, 65, 68, 70, 111, 121], and researchers generally agree that soft robots should be used in robotic applications in the future as they are safer and more adaptive in dynamic environments compared to the conventional rigid robots.

Being made of soft and deformable material, soft robots provide many benefits such as high adaptability and dexterity, and safe collisions [35], which are very useful for various applications, especially the manipulation of fragile objects [53, 59, 69, 76] (see B, C, and D in Fig. 1.3), environment exploration [11, 37, 46, 80, 87, 136] (see E, F, G, and H in Fig. 1.3), and medical operations [23, 25, 33, 34, 54, 74, 81, 114, 122] (see I, J, K in Fig. 1.3).

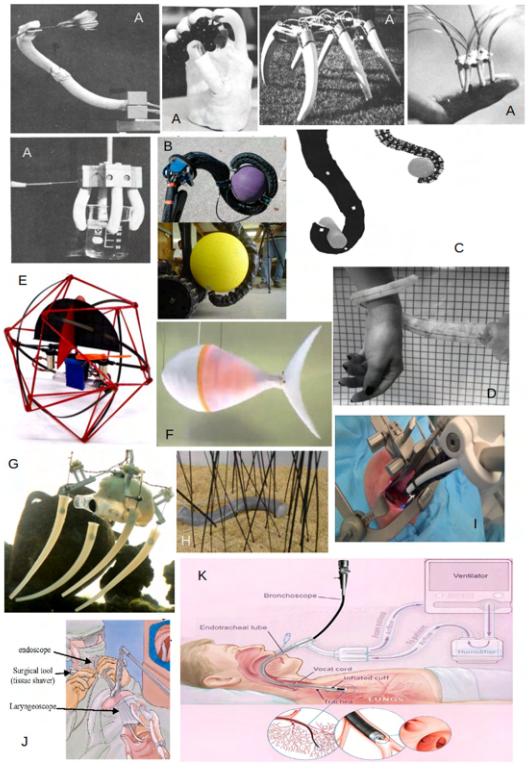


Figure 1.3: Soft Robots Applications: A- grasping and mobile mechanisms [117,119]. B- octArm [76]. C- elephant's trunk [53]. D- octopus arm [69]. E- terrestrial exploration [80]. F- tuna robotics [136]. G- multi-modal underwater [46]. H- planetary explorations [87]. I- transoral robotic surgery [74]. J- laryngeal surgery [114]. K- distal lung sampling [81].

1.1.4 Challenges and Motivations

Even though soft robots have made an interesting progress during the last few years, the actual process of designing a soft robot is still mainly inspired by a range of biological systems [111] including the structure and behavior of animal species such as the trunk of elephants [53], and the arms of an octopus [69]. Such a design procedure can be useful for initial tests and experimentation of the designed soft robot. However, when faced with specific performance objectives such as path planning and control [35, 129], it is possible that such a soft robot's design may not be able to achieve its objectives, especially because its range of reachability (i.e., its workspace) may be restricted.

Therefore, it is of benefit to the soft robotics community to propose generic methodologies in order to evaluate the workspace of soft robots, and also to assist and guide the systematic design of soft robots for the purpose of optimizing specific performance objectives.

The workspace estimation in soft robotics is still an open subject, and its importance is due to the numerous benefits that it can provide in order to solve different soft robotics' scientific challenges, mainly related to the robot's mechanical design and controller synthesis. On the one hand, the result of workspace evaluation can provide information about the accessibility of the object to be controlled [35, 38, 129], i.e. by identifying if the object's position belongs to the soft robot's workspace, sparing the controller's design time that can be wasted in trying

to control an unreachable object outside a robot's workspace. On the other hand, workspace information is crucial as well when planning feasible trajectory to be tracked by soft robots since the start and finish positions of a feasible trajectory should belong to a robot's workspace [129]. Besides, workspace analysis is also a necessary step to assist an effective, purpose-driven, and optimal design of soft robots [58], improving their stability, and increasing the range of their manipulability and reachability.

To achieve this aim, the actual thesis adopts two different mathematical models to describe the deformation of soft robots both with a slender shape and general form. Based on those adopted mathematical models, several approaches to estimate the workspace of soft robots are proposed. Finally, an effective method to optimize the design of soft robots for the purpose of achieving specific performance objectives is presented.

1.2 State of the Art

This work is concerned with the workspace evaluation and design optimization of soft robots based on their adopted mathematical models. Accordingly, a concise state of the art targeting the modeling, workspace evaluation, and design optimization of soft robots will be presented.

1.2.1 Mathematical Modeling of Soft Robots

Owing to their natural conformity and compliance, soft robots consists of a large number of DoFs and are characterized by different mechanical laws [50] compared to that of rigid robots, which makes their kinematic and dynamic modeling highly nonlinear and more complex.

In the literature, several modeling methods were proposed to describe the deformation of soft robots, and the most important ones are the Piece-wise Constant Curvature (PCC) [127], the Finite Element Method (FEM) [39], and the Piece-wise Constant Strain (PCS) based Discrete Cosserat method [103].

In the following, we will briefly investigate each modeling method.

1.2.1.1 Piece-wise Constant Curvature

The PCC approach [127] was initially established for kinematics studies and later extended to the dynamics of soft robots. It describes the investigated soft robot via a fixed number of arcs characterized by the curvature's radius, the arc's angle, and the bending plan. The robot is kinematically described through n_s successive transformations T_i^j (as shown in Fig. 1.4), with n_s being the total number of segments that constitutes the investigated robot.

The PCC has been widely and successfully applied to several continuum robots controlled by various methods of actuation [36, 63, 109, 127] . However, when the studied soft robot is subject to external loads, the assumption of a constant curvature does not always hold.

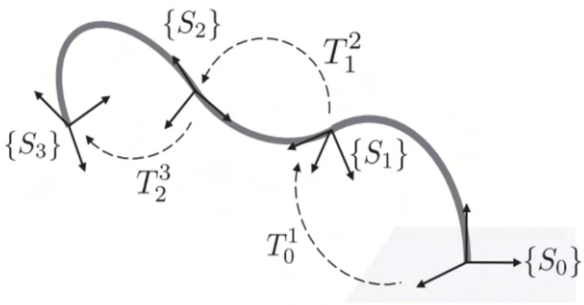


Figure 1.4: PCC modeling approach [127].

1.2.1.2 Finite Element Method

In [39], a modeling approach based on FEM was developed to describe the deformation of soft robots with general geometries (as shown by Fig. 1.5). In this method, the deformable domain of the structure is discretized into smaller finite elements using a specific meshing geometry, composed of a finite number of DoFs, which is then used to interpolate the behavior of the deformable domain by measuring the variation of the associated nodal displacements.

This modeling technique was successfully applied to different geometries of soft robots controlled by different means of actuation [15, 39, 47, 107]. However, to obtain a good modeling precision, this approach requires a fine mesh when discretizing the robot's spatial space, which in turn increases the number of DoFs and results in high computation complexity.

To solve this problem, model-order reduction (MOR) techniques have been introduced into the FEM approach [47] in order to reduce the number of DoFs and eventually decrease the computation complexity, but this improvement slightly affects the modeling precision.

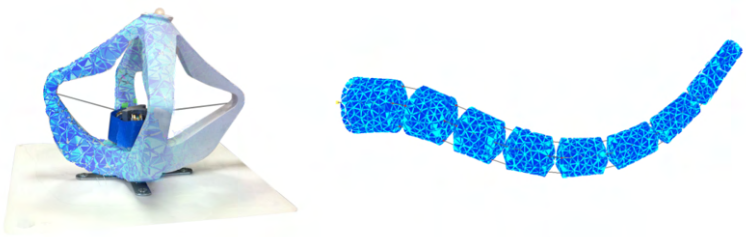


Figure 1.5: FEM modeling approach [39, 47].

1.2.1.3 Piece-wise Constant Strain

The Discrete Cosserat method [97, 98, 102–105] is a Piece-wise Constant Strain (PCS) method that can be viewed as an extension of the PCC technique.

This approach (see Fig. 1.6) describes the designed soft robot with a finite set of stain vectors (linear and angular) and provides the benefit of achieving a higher modeling precision with a lower number of DoFs. Besides, this modeling technique considers the material properties and geometric nonlinearity, thus taking into account large deformations. However, like the PCC, this approach can only be applied to slender-shaped soft robots. Moreover, the PCS assumes that all cross-sections (which are perpendicular to the centroidal line of slender soft robots) are

rigid and undeformable. Such an assumption is imposed for the sake of modeling simplicity, but inevitably it may affect the modeling precision. To solve this problem, a generalization of the PCS method was recently investigated in [20].

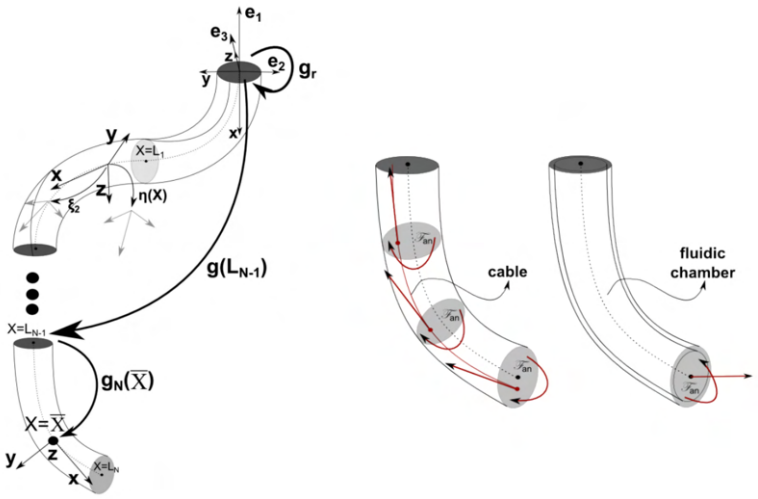


Figure 1.6: PCS modeling approach [103, 104].

1.2.1.4 Modeling Methods Comparison

The PCC modeling approach provides an effective and simple method for the modeling of soft robots. This method was largely deployed in the community of soft robotics, however, the adopted assumptions are not always valid which is very important for modeling the dynamics of soft robots.

Alternatively, the PCS method takes into account both the material properties of the studied robot and geometric nonlinearity, but with the limitation that it can only be applied to slender-shaped soft robots. Accordingly, the PCS modeling method is adopted by the present thesis in order to achieve the mathematical model of soft robots with slender forms.

On the other hand, and although it is computationally expensive, the FEM modeling method is adopted by this thesis in order to provide the mathematical model of soft robots with general form.

1.2.2 Workspace Evaluation

By definition, the workspace (also known as the accessible output set) of a robot is an area/volume that contains the whole of its achievable equilibrium configurations.

1.2.2.1 Rigid Robotics Workspace Methodologies

The subject of workspace determination has been widely investigated by the community of rigid robotics. In the literature, we can find many contributions to this topic using different methods to represent the workspace of rigid serial/parallel manipulators.

As to analytical Jacobian-based methods, the method proposed by [2, 3] (see A in Fig. 1.7) uses an explicit input-output function and consists of analytically determining singularities that make the sub-Jacobian row-rank deficient in order to map the boundaries of rigid manipulators.

This method was indeed successful in determining the singularities for a wide variety of rigid mechanisms [1]. However, this method has its limitations [2]: it requires that the output is an explicit function of the input (i.e., $x = \Omega(q)$), thus making it impossible to be used for soft robots. Moreover, a higher dimension of inputs increases the complexity of the analytical formulation of the sub-Jacobian which as a result makes it near impossible to analytically determine all singularities of such mechanisms.

In [55, 56] a Jacobian-based numerical method was presented for mapping the boundaries of rigid manipulator's workspaces using a continuation method based on the theory of bifurcation (see B in Fig. 1.7). To demonstrate the broadness of the applicability of this method, it was used to determine the workspace of planar and spatial Stewart platforms [73]. However, while seeking an initial boundary point, starting from a feasible initial point inside the workspace, this method may fail to identify enclosure curves (namely voids/holes) inside the workspace due to the discretization [2]. Moreover, it may also not be possible to distinguish an internal and an external boundary to the workspace when the ray emanating from an initial configuration passes through a bifurcation point [2]. These problems were properly handled in [18] where the methods presented in [55] were extended to treat more complex workspaces. Moreover, continuation methods were also used in [17] for the computation of tensegrity mechanism workspace, and a general high-order continuation method [57] with automatic differentiation was applied to a planar RRR mechanism and a three-dimensional Orthoglide parallel mechanism.

Regarding numerical methods, an optimization approach to determine the boundaries of rigid manipulator's workspaces has been presented in [116] (see C in Fig. 1.7). This numerical method consists of finding a suitable radiating point in the output coordinate space and then determining the points of intersection of a representative pencil of rays, emanating from the radiating point, with the boundary of the accessible set. This method is based on an optimization approach in which the accessibility of the workspace boundary is formulated in terms of a constrained optimization problem. However, the difficulties of this method lie in defining an appropriate objective function, non-convexity issues, and the inability to identify interior knowledge of the workspace.

Also, the wrench-feasible workspace of parallel rigid robots driven by cables has been analyzed in [48] by applying an interval analysis bisection method. Such a method has been then extended to analyze the workspace of cable-driven parallel mechanisms in [14, 79] (see D in Fig. 1.7). The advantage of this approach lies in the fact that it explores all feasible configurations in the workspace. However, this approach is in a way exhaustive since it consists of bisecting the whole space in order to find the feasible configurations which represent the workspace.

Finally, a numerical probabilistic method, Monte Carlo, was presented in [4, 27] to find the 2D and 3D workspace of a 3-DOF serial manipulator (see E in Fig. 1.7). This numerical method yields a robust result since it explores all actuators combinations, and has the advantage that it does not rely on the inverse problem of the robot. However, this technique is computationally expensive for high DoFs mechanisms.

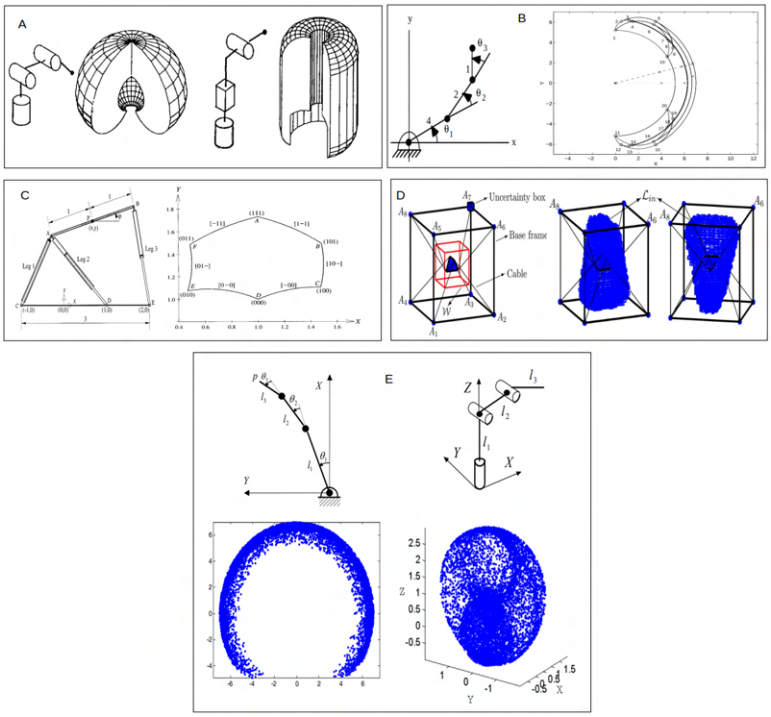


Figure 1.7: Rigid workspace methods: A- Jacobian-based analytical methods [3]. B- Jacobian-based numerical methods [55]. C- optimization methods [116]. D- interval analysis methods [48]. E- Monte-Carlo methods [27].

Although the above methods have been successfully implemented in determining the workspace of several types of rigid robots. However, they cannot be directly extended to analyze soft robots because of the modeling complexity for soft robots. In fact, the input-output relation for open-loop rigid chains (also called serial-link rigid robots) can normally be explicitly defined through its direct kinematic model ($u = f(q)$). However, such a relation for soft robots is generally defined in an implicit way ($f(u, \tau) = 0$) through a complex static/kinematic model (which will be explained in Chapter 2) for the purpose of modeling soft material deformations. On the other hand, closed-loop rigid chains (also called parallel rigid robots) necessitate the use of numerical schemes in order to solve the direct kinematics of the robot [12, 21, 22, 78], which is the same problem for soft robots. Hence, the workspace methodologies explored for the different kinds of rigid mechanisms need to be adapted in order to treat the same problem for soft robots since they consist of a large number of DoFs due to the compliance and the flexibility of the used material, and also because the modeling of soft robots is established through different mechanical laws, i.e., principles of continuum mechanics [50] (which will be explained in Section 2.5.2 for the PCS model, and Section 2.5.3 for the FEM model).

1.2.2.2 Continuum Robotics Workspace Approaches

On the other side, in the community of continuum robotics, the workspace subject has been rarely investigated. To our best knowledge, three main contributions were provided and all of them consist of discretizing the inputs (actuators) space in different fashions.

The paper of [24] (see A in Fig. 1.8) presents the workspace characterization for concentric

tube continuum robots by extending Monte-Carlo algorithms developed in [27]. Also, in [132] (see B in Fig. 1.8), the workspace analysis of cable-driven continuum manipulators based on the static model was investigated using an approach that consists of discretizing the input domain and finding the set of all reachable equilibrium positions of the manipulator's end-effector. The limitations of the above-stated approaches are: firstly, they consist of discretizing the inputs space, which varies depending on the configuration of the investigated continuum robot; secondly, the computational complexity (which is equal to $n_d^{n_\tau}$) of these approaches can explode exponentially when increasing the dimension of the actuators n_τ or when we want to get higher precision, i.e., increasing n_d; thirdly, valuable information about the robot's singularities and interior/exterior boundaries cannot be identified using these classical approaches since they only focus on solving the direct model of the investigated continuum robots in order to find the set of poses that constitutes the workspace.

In [26] (see C in Fig. 1.8), an approximation method was used to determine the workspace boundaries of continuum robots by sweeping the robot from the neutral configuration to the fully curved configuration for the purpose of deducing the workspace surface from the robot's tip trajectory. The proposed approximation method consists of estimating the workspace boundary by searching only the points distributed on the surface of the workspace. However, this method still consists of exploring the inputs space in a different manner by varying one configuration from its minimum to its maximum value while keeping the other configurations at a fixed neutral value. This strategy can map the workspace boundaries of the investigated robot and at the same time reduce the computational time of the pre-proposed scanning method from $n_d^{n_\tau}$ combinations to 2^{n_τ} combinations. Nevertheless, for robots controlled by a higher number of actuators, the approximation method will still be computationally expensive. Another limitation of this approach lies in failing to identify some workspace boundaries because singularities result not only from actuators saturation, but also due to physical limitations [18, 55] such as the mechanism's length, self-collision, or actuation instabilities, because this approach doesn't apply the idea that the workspace interior/exterior boundaries of the investigated continuum robots are extracted from the set of their output singularities ignoring the row-rank deficiency of the Jacobian of the generalized mechanical constraints which doesn't comply with what was studied in [3, 18, 55].

1.2.2.3 Soft Robotics Workspace Methods

For a given soft robot's configuration, till the starting date (1 October 2018) of the present thesis, no work has yet been investigated about how to comprehensively evaluate its workspace.

1.2.2.4 Workspace Methods Summary

As stated above, different methodologies were investigated for the workspace evaluation of rigid robots, few for continuum robots, and almost none for soft robots.

The methods investigated for continuum robots are not efficient since they are based on the discretization of the inputs space, whose dimension depends on the studied configuration of the robot, but such methods can still be used as a comparative reference since they exhaustively explore all possible achievable configurations of the investigated robot's workspace.

On the other hand, it is more appropriate to extend the applicability of the main workspace approaches developed in the rigid robotics community in order to treat the soft robotics case. However, this cannot be done in a direct way since soft robots are modeled in a more complex manner, yielding an implicit input-output relation.

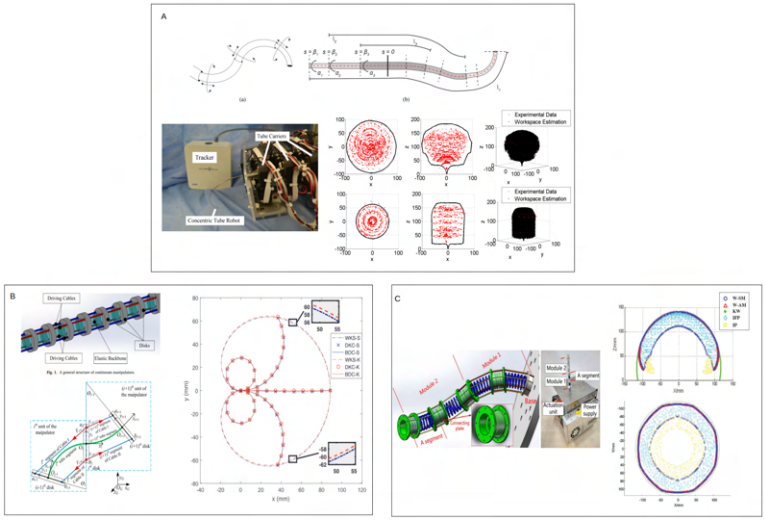


Figure 1.8: Continuum workspace methods: A- Monte-Carlo algorithms [24]. B- direct method [132]. C- approximation method [26].

1.2.3 Design Optimization of Soft Robots

Several design optimization methods were proposed for specific soft robotics applications, which can be classified into three main categories: heuristic methods, model-based, and hybrid.

1.2.3.1 Heuristic Methods

Genetic algorithms were explored in [31, 58] for the design optimization of soft robots (see A in Fig. 1.9). These methods are particularly useful for treating highly nonlinear problems since they don't require the calculation of costly gradients. However, defining a proper fitness function is difficult.

Evolutionary algorithms were also used in [16] (see B in Fig. 1.9) where it was proposed to optimize a small number of size parameters of pneumatic-actuated inflatable manipulators, and in [110] (see C in Fig. 1.9) for the design optimization of soft pneumatic actuators. In [10] (see D in Fig. 1.9), the central concept was to develop an assistive manipulator that can automate the bathing task for elderly citizens where the development of a novel algorithm based on cooperative multi-agent reinforcement learning that simultaneously optimizes stiffness and position was proposed.

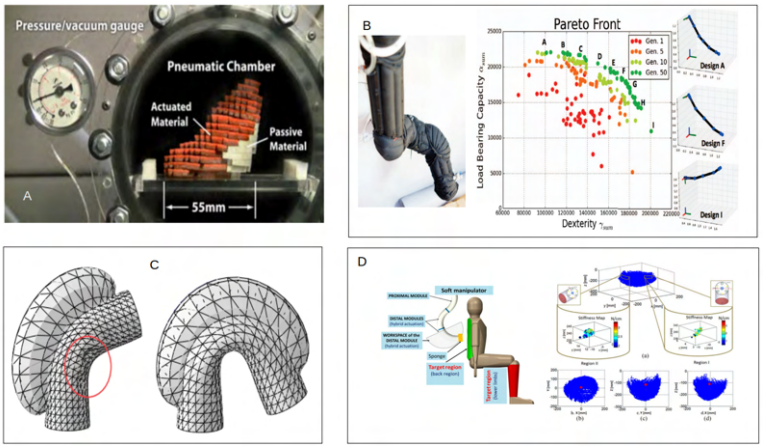


Figure 1.9: Heuristic Methods: A- Automatic design of soft robots [58]. B- Multi-objective design optimization of a soft pneumatic robot [16]. C- Design optimization of soft pneumatic actuators [110]. D- Multi-objective optimization for stiffness and position control in a soft robot arm [10].

1.2.3.2 Model-Based Approaches

As to model-based approaches, in [133] (see A in Fig. 1.10), a density-based topology optimization method was employed for the design of the exterior shell of a soft pneumatic gripper with a given shape and position in order to grasp a wide range of objects using the FEM model. Next, improving the mechanical performance of soft pneumatic robots through parameterized layout and shape optimization of air chambers was studied in [49] (see B in Fig. 1.10). In [84] (see C in Fig. 1.10), a model-based optimization for the design of a rotational soft robotic system driven by double cone dielectric elastomer actuators was investigated. The work of [29] (see D in Fig. 1.10) studied the design optimization of a soft robotic device for neurorehabilitation by computing a set of optimal parameters which, in turn, could be used to customize the device according to the patient's need. In [40] (see E in Fig. 1.10), the design of a pneumatically actuated silicone module with three pneumatic chambers is considered and optimized with the aim of using it in a soft robot arm for robotic surgery applications. In [45] (see F in Fig. 1.10), the design optimization of soft crawling robots (inchworm and quadrupedal) for maximum bending motion with respect to input pressure was evaluated using FEM. The work of [75] (see G in Fig. 1.10) developed a methodology for converting caterpillar-like soft-bodied robot from a lumped-parameter form into a Computer-Aided Design (CAD) model that could be easily manufactured using the results of coarse dynamic simulations. Also, the subject of optimal actuator location was investigated in [83], where it is proposed to minimize a linear-quadratic (LQ) function to find the optimal actuators location. This method was then extended in [130] (see H in Fig. 1.10) to achieve the optimal actuator location for a medical endoscope controlled by electro-active polymers.

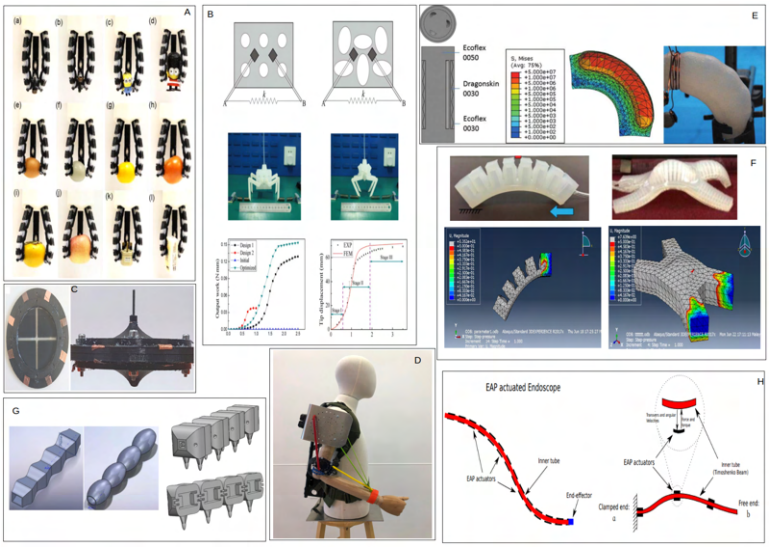


Figure 1.10: Model-based approaches: A- topology optimized design of a multimaterial soft gripper [133]. B- soft pneumatic robots chamber layout design optimization [49]. C- rotational soft robotic system design [84]. D- soft robotic device design optimization [29]. E- design optimization of a pneumatically actuating silicone module [40]. F- design optimization of a soft crawling robot [45]. G- dynamics-based design of soft robots [75]. H- optimal actuator location for electro-active polymer actuated endoscope [130].

1.2.3.3 Hybrid Approaches

In [94] (see Fig. 1.11), two frameworks, a model-based optimization, and deep reinforcement learning were proposed for the design optimization of a multi-chamber pneumatic-driven soft actuator in order to optimize its mechanical performance. The design goal was to achieve maximal horizontal motion of the top surface of the actuator with a minimum effect on its vertical motion.

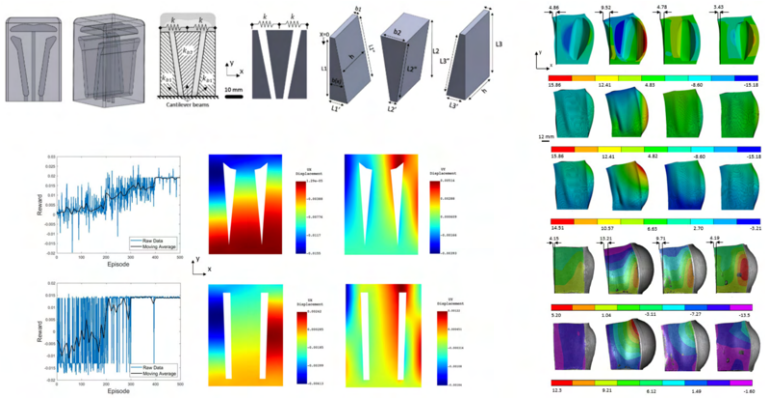


Figure 1.11: Hybrid approaches [94].

1.2.3.4 Conclusion

As stated above, different strategies (heuristic, model-based, and hybrid) were fully investigated for the design optimization of different soft robots. Those methods have helped achieve the optimal design of the investigated soft robots which were successfully deployed in real-life scenarios.

However, despite the significant progress in the conceptual design of soft robots, the mentioned methods are in a sense specific to a particular type of robot with a predefined concept.
On the other hand, generic methods focusing on the systematic design optimization of soft robots that are controlled via a generic number of actuators for the purpose of optimizing specific performance objectives have rarely been addressed in the literature.
This is due to the difficulty and complexity that lies in establishing an appropriate cost function combining multiple desired design objectives, particularly when these objectives are competing with each other.

To our best knowledge, no former work has been investigated on the optimization of the design of soft robots for the purpose of optimizing their workspace attainability and reachability which is of high importance and benefit to the design optimization of soft robots.

1.3 Contributions of the Thesis

1.3.1 Contributions and Organization of the Manuscript

For soft robotic applications such as design optimization, path planning, and pick-place, it is necessary and useful to evaluate the workspace of the studied soft robot. In this thesis, we investigate such a problem from two views, a direct and an inverse one.

On the one hand, the direct problem of the workspace can be formulated by the following question: given the designed configuration of the investigated soft robot, how should we efficiently estimate its workspace? (which will be studied in Chapter 3, Chapter 4, and Chapter 5). Estimating the workspace of a given soft robot's prototype is a crucial step for assessing the reachability and accessibility of particular locations which relates to the controller synthesis of the robot. Also, solving this problem helps in determining the robot's singularities and impediments of motion which relates to the design of the robot.

On the other hand, the following question formulates the inverse problem: how should we optimize the design of a soft robot in order to achieve specific performance objectives? (which will be investigated in Chapter 6). Given specific performance objectives (e.g., reaching specific locations), it is valuable for many reasons (both economic and scientific) to determine the soft robot's optimal design such that its workspace contains those desired locations in a virtual environment before proceeding to the physical design of the final prototype of the robot.

Accordingly, to answer these two questions, the manuscript was organized into 7 chapters as follows (See Fig. 1.12):

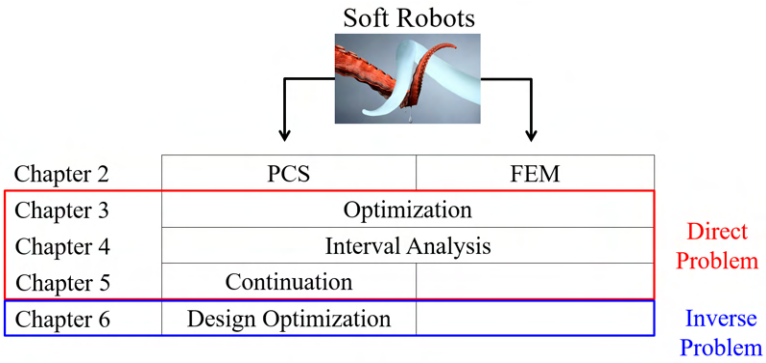


Figure 1.12: Manuscript organization.

Chapter 2 introduces the dynamic model of the adopted PCS and FEM modeling methods. Also, in this chapter, we establish the workspace definition of soft robots. Finally, we present a classic approach (named the forward approach) that consists of discretizing the actuators space in order to provide the workspace of soft robots. This approach was then validated via a trunk-like soft robot for both the PCS and FEM cases.

In Chapter 3, we propose an optimization-based approach [9, 123] that consists of estimating the workspace of soft robots by mapping their workspace's exterior boundaries. This approach was applied to both the PCS [123] and FEM [9] modeling methods.

Next, an interval analysis-based method [7] (named the forward-backward approach) is outlined in Chapter 4 in order to achieve the workspace of soft robots by exploring all feasible configurations. The proposed approach was applied to both the PCS and FEM modeling methods.

Chapter 5 presents a continuation method [8] based on the theory of bifurcation in order to map the interior and exterior boundaries of soft robots. This approach was only applied to the PCS modeling method.

After, an optimization approach based on the adopted PCS model was established in Chapter 6 in order to optimize the design of soft robots for the purpose of achieving specific performance objectives.

Finally, Chapter 7 provides the conclusions of the present thesis and presents the perspective of future works, especially, the orientation aspect of the workspace, the generalization of the continuation approach, and the possibility of its extension to treat the FEM case, and also the design optimization of soft robots based on the FEM method.

1.3.2 Publications

The contributions of the present thesis are outlined in the following:

International Conference:

1. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. "Fem based workspace estimation for soft robots: A forward-backward interval analysis approach." In 2020 3rd IEEE International Conference on Soft Robotics (RoboSoft), pp. 170-175. IEEE, 2020.
   Summary: In this paper, the preliminary result of an interval analysis-based approach in order to estimate the workspace of soft robots using the FEM modeling method was submitted.

Journal Articles:

1. Walid, Amehri, Gang Zheng, Alexandre Kruszewski, and Federico Renda. "Discrete Cosserat Method for Soft Manipulators Workspace Estimation: An Optimization-Based Approach." Journal of Mechanisms and Robotics 14, no. 1 (2021): 011012.
   Summary: In this article, we presented an optimization-based approach in order to estimate the exterior workspace boundary of slender-shaped soft robots using the PCS modeling method.

2. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. "Workspace Boundary Estimation for Soft Manipulators Using a Continuation Approach." IEEE Robotics and Automation Letters 6, no. 4 (2021): 7169-7176.
   Summary: In this letter, we provided a continuation approach based on the theory of bifurcation in order to estimate the interior and exterior boundaries slender-shaped soft robots using the PCS modeling method.

3. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. "FEM-based Exterior Workspace Boundary Estimation for Soft Robots via Optimization." IEEE Robotics and Automation Letters (2022).
   Summary: In this paper, we extended the applicability of the optimization-based approach by applying it to the FEM modeling method in order to estimate the exterior workspace boundary of soft robots.

4. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. Soft Robotics SORO, FEM-based Reachable Workspace Estimation of Soft Robots using an Interval Analysis approach.
   Summary: In this journal, the full version of the proposed interval analysis approach was presented which contains new results for FEM model with external forces, self-contained explanation and results on interval analysis, novel uniform spatial grid discretization strategy, a detailed algorithm with an improved stop condition, and comprehensive configuration simulation scenarios to emphasize the effectiveness of the proposed interval analysis based approach.

Submitted Articles:

1. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. ASME JMR, Position-Access Workspace of Soft Manipulators using an Interval Analysis Method.
   Summary: In this article, we extended the applicability of the interval analysis-based approach by applying it to the PCS modeling method in order to estimate the workspace of soft robots with slender shape.

2. Amehri, Walid, Gang Zheng, and Alexandre Kruszewski. IJRR, Cosserat-based Optimization Design for Slender Soft Manipulators.
   Summary: In this article, we proposed an optimization approach based on the adopted PCS model in order to optimize the design of soft robots with slender shape for the purpose of achieving specific performance objectives.

Chapter 2

Modeling and Forward Methods

2.1 Introduction

As stated in Chapter 1, the PCS method provides a good modeling precision with a lower number of DoFs which can only be applied to slender-shaped soft robots. On the other hand, the FEM method provides a precise model of soft robots with the general form, but it is computationally expensive.

Accordingly, the first step of this chapter is to present the mathematical models of the PCS and FEM methods for the purpose of modeling soft robots with a slender shape and general form, respectively. Next, we establish the definition of a soft robot's workspace. Finally, we propose a forward approach to estimate the workspace of soft robots. This approach was then validated using a trunk-like soft robot [129] for both the PCS and FEM cases.

In the following, we do not consider the material's nonlinearity, instead, we consider that the material is isotropic, and the deformation is entirely elastic.

2.2 Piece-wise Constant Strain (PCS)

The dynamic model based on the Piece-wise Constant Strain (PCS) approach was developed in [101] for a soft-rigid multibody system. In the following, all the quantities are expressed in the local (body) coordinate frame if not specified. The superscript $'$ represents the partial derivative with respect to the space variable.

In the following, we illustrate the mathematical model of the PCS model using the example of a tendon-driven soft robot. The method could be as well applied to other manners of actuation [102].

2.2.1 Kinematics

The configuration of a soft body i with respect to its predecessor in the chain (as illustrated by Fig. 2.1) is defined as a curve $g_i(\cdot) : X \in [0, L_i] \mapsto g_i(X) \in SE(3)$ with:

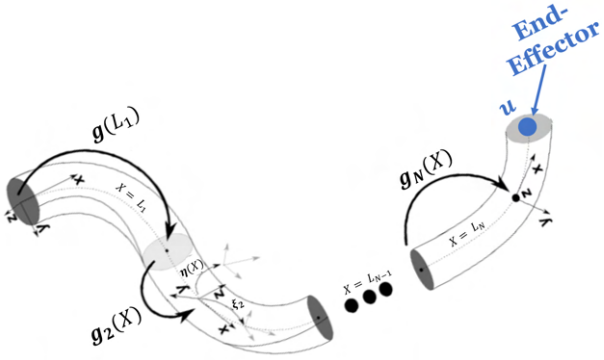


Figure 2.1: Schematic of the PCS kinematics.

$$g_i(X) = \begin{pmatrix} R_i & u_i \\ 0^T & 1 \end{pmatrix} \tag{2.1a}$$

where R_i is the orientation matrix, u_i is the end-effector position, L_i is the length, $\hat{\xi}_i$ is the strain matrix, of a body i, and $X \in [0, L]$ is the abscissa along the robot arm.

The continuous models of the position, velocity and acceleration of a soft body can be derived from the Cosserat rod theory, which gives [19]:

$$g_i' = g_i \hat{\xi}_i \tag{2.2a}$$
$$\eta_i' = \dot{\xi}_i - \mathrm{ad}_{\xi_i} \eta_i \tag{2.2b}$$
$$\dot{\eta}_i' = \ddot{\xi}_i - \mathrm{ad}_{\dot{\xi}_i} \eta_i - \mathrm{ad}_{\xi_i} \dot{\eta}_i \tag{2.2c}$$

where $\mathrm{ad}_{(.)} \in \mathbb{R}^{6\times 6}$ is the adjoint operator of the Lie algebra, and

$$\hat{\xi}_i(X) = \begin{pmatrix} \tilde{\mathsf{k}}_i & \mathsf{p}_i \\ 0^T & 0 \end{pmatrix} \in \mathfrak{se}(3) \quad \xi_i(X) = \left(\mathsf{k}_i^T, {\mathsf{p}_i}^T\right)^T \in \mathbb{R}^6 \tag{2.3}$$

defines the strain state with $\tilde{\mathsf{k}}_i(X) \in \mathfrak{so}(3)$, $\mathsf{k}_i(X) \in \mathbb{R}^3$, and $\mathsf{p}_i(X) \in \mathbb{R}^3$ respectively the angular and linear strain (when compared to the reference values k_i^* and p_i^*);

$$\hat{\eta}_i(X) = \begin{pmatrix} \tilde{\mathsf{w}}_i & \mathsf{v}_i \\ 0^T & 0 \end{pmatrix} \in \mathfrak{se}(3) \quad \eta_i(X) = \left(\mathsf{w}_i^T, \mathsf{v}_i^T\right)^T \in \mathbb{R}^6 \ . \tag{2.4}$$

is the cross-section velocity twist with $\tilde{\mathsf{w}}_i(X) \in \mathfrak{so}(3)$, $\mathsf{w}_i(X) \in \mathbb{R}^3$, and $\mathsf{v}_i(X) \in \mathbb{R}^3$ respectively the angular and linear velocity. To model constrained rod, such as the Kirchhoff-Love case with angular strain only, the strain field is specified as:

$$\xi_i = B_{q_i} q_i + \xi_i^* \tag{2.5}$$

where $B_{q_i} \in \mathbb{R}^{6\times n_i}$ forms a basis for the allowed motion subspace, $q_i \in \mathbb{R}^{n_i}$ contains the values of the allowed strains and, $\xi_i^* \in \mathbb{R}^6$ is the initial twist modeling the initial shape.

Assuming piece-wise constant strains [98], (2.2) can be analytically integrated using the matrix exponential method, leading to:

$$g_i(X) = e^{X\hat{\xi}_i} \tag{2.6a}$$

$$\eta_i(X) = \mathrm{Ad}_{g_i}^{-1}\eta_h + \mathrm{Ad}_{g_i}^{-1}\mathrm{T}_{g_i}B_{q_i}\dot{q}_i \tag{2.6b}$$

$$\dot{\eta}_i(X) = \mathrm{Ad}_{g_i}^{-1}\dot{\eta}_h + \mathrm{Ad}_{g_i}^{-1}\int_0^X \mathrm{Ad}_{g_i(s)}\mathrm{ad}_{\eta_i(s)}dsB_{q_i}\dot{q}_i + \mathrm{Ad}_{g_i}^{-1}\mathrm{T}_{g_i}B_{q_i}\ddot{q}_i \tag{2.6c}$$

where $\mathrm{Ad}_{g_i}(X) \in \mathbb{R}^{6\times 6}$ is the Adjoint operator of $SE(3)$, and $\mathrm{T}_{g_i}(X)$ is the tangent operator of the exponential map.

Successive applications of the kinematics (2.6) for all the bodies of the system yields the definition of the geometric Jacobian $J_i(q,X) \in \mathbb{R}^{6\times n}$ and its derivative $\dot{J}_i(q,\dot{q},X) \in \mathbb{R}^{6\times n}$ (n being the total number of DOFs), which links the generalized strain vector $q = \begin{bmatrix} q_1^T & q_2^T & \cdots & q_N^T \end{bmatrix}^T \in \mathbb{R}^n$ (N being the total number of bodies) and the velocity twist $\eta_i(X)$, for each soft body i, as shown below:

$$\begin{aligned}\eta_i(X) &= \sum_{h=0}^{i} \mathrm{Ad}_{g_h\cdots g_i}^{-1}\mathrm{T}_{g_h}B_{q_h}\dot{q}_h \\ &= \sum_{h=0}^{i} {}^iS_h\dot{q}_h = J_i(q,X)\dot{q}\end{aligned} \tag{2.7a}$$

$$\begin{aligned}\dot{\eta}_i(X) &= \sum_{h=0}^{i} {}^iS_h\ddot{q}_h + \mathrm{Ad}_{g_h\cdots g_i}^{-1}\int_0^X \mathrm{Ad}_{g_h(s)}\mathrm{ad}_{\eta_h(s)}dsB_{q_h}\dot{q}_h \\ &= \sum_{h=0}^{i} {}^iS_h\ddot{q}_h + {}^i\dot{S}_h\dot{q}_h = J_i(q,X)\ddot{q} + \dot{J}_i(q,\dot{q},X)\dot{q}\end{aligned} \tag{2.7b}$$

$$\hat{\eta}_i(X) = g_i^{-1}\,\dot{g}_i \tag{2.7c}$$

where the block elements of the i^{th} Jacobian ${}^iS_{(\cdot)} \in \mathbb{R}^{6\times n_{(\cdot)}}$ and its derivative ${}^i\dot{S}_{(\cdot)} \in \mathbb{R}^{6\times n_{(\cdot)}}$ have been defined in [103].

2.2.2 Continuous Dynamics

Once a Jacobian is found, the generalized dynamics of the system can be obtained by projecting the free dynamics of each soft body by virtue of the D'Alembert's principle. The free dynamic equation of a soft body, with its boundary conditions, is given by [97]:

$$\begin{aligned}&\mathcal{M}_i\dot{\eta}_i + \mathrm{ad}_{\eta_i}^*\mathcal{M}_i\eta_i = \left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)' + \mathrm{ad}_{\xi_i}^*\left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right) + \bar{\mathcal{F}}_{e_i} \\ &\left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)(0) = -\mathcal{F}_{J_i} \quad \left(\mathcal{F}_{i_i} - \mathcal{F}_{a_i}\right)(L_i) = -\mathrm{Ad}_{g_{ij}}^*\mathcal{F}_{J_j}\ ;\end{aligned} \tag{2.8}$$

where $\mathcal{M}_i(X) = \mathrm{diag}(J_{x_i}, J_{y_i}, J_{z_i}, A_i, A_i, A_i)\rho_i \in \mathbb{R}^{6\times 6}$ is the screw inertia matrix of the cross-section ($J_{\cdot_i}(X)$ being the second moment of the area about the axis $\cdot$ and $A_i(X)$ being the area of the cross-section); $\bar{\mathcal{F}}_{e_i}(X) \in \mathbb{R}^6$ is the distributed external load; $\mathcal{F}_{a_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the distributed actuation [99]; $\mathcal{F}_{i_i}(X) \in \mathbb{R}^6$ is the internal wrench due to the elasticity of the soft body; $\mathcal{F}_{J_{(\cdot)}} \in \mathbb{R}^6$ is the wrench transmitted across joint $(\cdot)$ and $\mathrm{ad}_{(\cdot)}^*$ (respectively $\mathrm{Ad}_{(\cdot)}^*$) $\in \mathbb{R}^{6\times 6}$ is the co-adjoint (respectively co-Adjoint) map of the Lie algebra (respectively Lie group).

2.2.3 Elasticity and Actuation Forces

Regarding the internal elastic force, a linear viscoelastic constitutive model is usually chosen:

$$\mathcal{F}_{i_i}(X) = \Sigma_i\left(\xi_i - \xi^*\right) + \Upsilon_i\dot{\xi}_i = \Sigma_i B_{q_i}q_i + \Upsilon_i B_{q_i}\dot{q}_i \tag{2.9}$$

where

$$\Sigma_i(X) = \text{diag}(G_i J_{x_i}, E_i J_{y_i}, E_i J_{z_i}, E_i A_i, G_i A_i, G_i A_i)$$
$$\Upsilon_i(X) = \text{diag}(J_{x_i}, 3J_{y_i}, 3J_{z_i}, 3A_i, A_i, A_i)\nu_i \in \mathbb{R}^{6\times 6}$$

are the screw stiffness and viscosity matrix (E_i being the Young's modulus, G_i the shear modulus and ν_i the shear viscosity).

With regards to the actuation load (as depicted in Fig 2.2), by computing the force and moment exerted by the internal tendon on the mid-line of the rod [99].

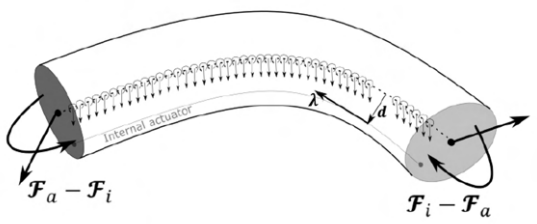


Figure 2.2: Schematic of the tendon-driven actuation of one body.

This is given by:

$$\mathcal{F}_a(X) = \sum_{i=1}^{n_\tau} \begin{bmatrix} \tilde{d}_i(X)\lambda_i(X) \\ \lambda_i(X) \end{bmatrix} \tau_i = H_\tau(q, X)\tau \tag{2.10}$$

where $d_i(X) \in \mathbb{R}^3$ represents the distance from the mid-line to the internal i^{th} actuator, $\tilde{d}_i$ denotes d_i in $\mathfrak{so}(3)$, $\lambda_i(X) \in \mathbb{R}^3$ is the unit vector tangent to the actuator path, and $\tau \in \mathbb{R}^{n_\tau}$ is the vector of magnitude of the actuators force given by the negative of the tendons tension (n_τ being the total number of actuators). While the distance $d_i(X)$ is fixed by design, the tangent vector $\lambda_i(X)$ depends, in general, on the deformation $\xi_i(X)$ of the soft robot body [108], [100]. Going further into details, the unit tangent vector $\lambda_i(X)$ can be obtained by spatial differentiation of the position vector of the actuator, thus:

$$\lambda_i(X) = \frac{\left[g_i^{-1}\left(g_i d_i\right)'\right]_3}{\|g_i^{-1}\left(g_i d_i\right)'\|} = \frac{\left[\hat{\xi}_i d_i(X) + d_i'(X)\right]_3}{\|\hat{\xi}_i d_i(X) + d_i'(X)\|} \,. \tag{2.11}$$

where $d_i(X)$ is expressed in homogeneous coordinates, $[\cdot]_3$ extracts the first three rows of a homogeneous vector and $\|\cdot\|$ takes the Euclidean norm.

Remark 1. *Considering d_i as a small parameter, a useful approximation can be obtained, which allows for a unit tangent vector independent of the deformation with a slight loss of accuracy [97]. In this case, one can consider $\lambda_i = [1\ 0\ 0]^T$. As a result, the generalized actuation matrix $H(q)$ may be approximated by $H(q) \approx H$.*

2.2.4 Discrete Dynamics

By Jacobian projection of the free dynamics (2.8), we obtain the generalized dynamics in its classical form:

$$M\left(q\right)\ddot{q} + \left(C\left(q, \dot{q}\right) + D\right)\dot{q} + Kq = H\left(q\right)\tau + F\left(q\right) \tag{2.12}$$

where $M \in \mathbb{R}^{n\times n}$ is the generalized mass matrix, $C \in \mathbb{R}^{n\times n}$ is the generalized Coriolis matrix, $D \in \mathbb{R}^{n\times n}$ is the block-diagonal generalized damping matrix, $K \in \mathbb{R}^{n\times n}$ is the block-diagonal generalized stiffness matrix, $H \in \mathbb{R}^{n\times n_\tau}$ is the generalized actuation matrix, $F \in \mathbb{R}^n$ is the vector of generalized position-dependent external force and $\tau \in \mathbb{R}^{n_\tau}$ is the vector of applied actuators force. Going further into details, the coefficient matrices take the form:

$$M(q) = \sum_{i=1}^{N} \int_0^{L_i} J_i^T \mathcal{M}_i J_i dX \tag{2.13a}$$

$$C(q,\dot{q}) = \sum_{i=1}^{N} \int_0^{L_i} J_i^T \left(\mathrm{ad}^*_{J_i\dot{q}} \mathcal{M}_i J_i + \mathcal{M}_i \dot{J}_i \right) dX \tag{2.13b}$$

$$D = \mathrm{diag}\left(B_{q_1}^T \int_0^{L_1} \Upsilon_1 dX B_{q_1}, \cdots, B_{q_N}^T \int_0^{L_N} \Upsilon_N dX B_{q_N} \right) \tag{2.13c}$$

$$K = \mathrm{diag}\left(B_{q_1}^T \int_0^{L_1} \Sigma_1 dX B_{q_1}, \cdots, B_{q_N}^T \int_0^{L_N} \Sigma_N dX B_{q_N} \right) \tag{2.13d}$$

$$H(q) = \left[\left(B_{q_1}^T \int_0^{L_1} H_\tau dX \right)^T, \cdots, \left(B_{q_N}^T \int_0^{L_N} H_\tau dX \right)^T \right]^T \tag{2.13e}$$

$$F(q) = \sum_{i=1}^{N} \int_0^{L_i} J_i^T \bar{\mathcal{F}}_{e_i} \tag{2.13f}$$

2.3 Finite Element Method (FEM)

2.3.1 FEM Introduction

The dynamic model of soft robots can be obtained based on Galerkin method of FEM and the Euler Lagrange equation [95].

FEM consists firstly of spatially discretizing the deformable domain of the structure into smaller finite elements N through the creation of a corresponding mesh which contains a finite number of DoFs n to interpolate the behavior of the deformable domain by measuring the variation of the associated nodal displacements. Then, the Euler-Lagrange method is applied to deduce the dynamics of soft robots.

In the literature, various geometries are proposed to discretize the spatial domain of a given structure, such as prisms, beams, hexahedrons and tetrahedrons [96, p. 732]. For the purpose of generality and convenience for element matrix integration, we select the linear tetrahedron element (shown by Fig. 2.3) as the appropriate meshing structure in this thesis.

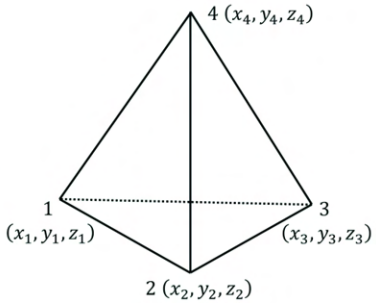


Figure 2.3: Linear tetrahedron element.

Take the trunk-like soft robot as an example, Fig. 2.4 shows the corresponding discretized version in the domain.

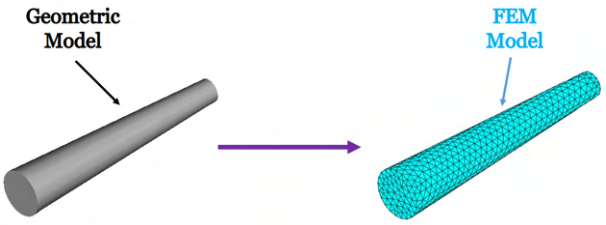


Figure 2.4: Discretized mesh model for trunk-like soft robot using the linear tetrahedron element.

Based on the discretized mesh model, the deduction of its dynamics can be realized element by element. In the following, we apply the modeling approach on a single linear tetrahedron element to get its motion equation, and then we perform an assembly process of all elements to obtain the motion equation for the whole structure.

2.3.2 Displacement Model

After having created the mesh, we have to define the primary variables of a single element, which corresponds to the displacement of the nodes. According to [96, p. 93], the displacement vector can be defined by the following formula:

$$\delta_e(x, y, z, t) = N_e q_e \tag{2.14}$$

where the index e denotes the investigated element, N_e is the basis functions matrix, and q_e represents the nodal (1, 2, 3, 4) displacement vector for a single element, with:

$$N_e = \begin{bmatrix} N_1 & 0 & 0 & \cdots & N_4 & 0 & 0 \\ 0 & N_1 & 0 & \cdots & 0 & N_4 & 0 \\ 0 & 0 & N_1 & \cdots & 0 & 0 & N_4 \end{bmatrix}_e \tag{2.15}$$

And

$$q_e = \begin{bmatrix} q_{1x} & q_{1y} & q_{1z} & q_{2x} & q_{2y} & q_{2z} & q_{3x} & q_{3y} & q_{3z} & q_{4x} & q_{4y} & q_{4z} \end{bmatrix}_e^T \tag{2.16}$$

The basis functions N_e represents the mathematical model of the displacement over space, which needs to satisfy some requirements such as the continuity over a specific order and the Kronecker property where each basis function has a value of one at its own node and zero at the other nodes in the corresponding element [95, p. 82].

In the literature, we can find different polynomial families to define the basis functions. In the following, we present the Lagrange family [96] because it is accurate and straightforward. It has been stated in [71, p. 252] that the order p of the polynomial is defined via the number of nodes of the element n_e by the following equation:

$$n_e = \frac{1}{6}(p+1)(p+2)(p+3) = 4 \tag{2.17}$$

Using the linear tetrahedron element which contains four nodes, the polynomial order for the basis functions should be equal to 1. According to this, the basis functions can be written in the following polynomial form:

$$N_i(x, y, z) = \left[\alpha_{(1,i)} + \alpha_{(2,i)}x + \alpha_{(3,i)}y + \alpha_{(4,i)}z\right]_e \tag{2.18}$$

where α_i for $1 \leqslant i \leqslant 4$ are the coefficients to be determined. For this, we define V_{equ} as the volume of the element in an equilibrium state, then the basis functions of the linear tetrahedron element can be formulated as:

$$N_i(x, y, z) = \left[\frac{1}{6V_{equ}} (a_i + b_i x + c_i y + d_i z) \right]_e \tag{2.19}$$

with $\alpha_{(1,i)} = \dfrac{a_i}{6V_{equ}}$, $\alpha_{(2,i)} = \dfrac{b_i}{6V_{equ}}$, $\alpha_{(3,i)} = \dfrac{c_i}{6V_{equ}}$, $\alpha_{(4,i)} = \dfrac{d_i}{6V_{equ}}$, and

$$V_{equ} = \frac{1}{6} \begin{vmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ 1 & x_3 & y_3 & z_3 \\ 1 & x_4 & y_4 & z_4 \end{vmatrix}_e$$

where (x_i, y_i, z_i) are the coordinates of the node i in the corresponding element. The variables a_i, b_i, c_i and d_i in (2.19) are constant coefficients which are computed when the element is in equilibrium point, with the following form:

$$a_i = (-1)^{m+1} \begin{vmatrix} x_j & y_j & z_j \\ x_k & y_k & z_k \\ x_l & y_l & z_l \end{vmatrix}_e \quad , b_i = (-1)^m \begin{vmatrix} 1 & y_j & z_j \\ 1 & y_k & z_k \\ 1 & y_l & z_l \end{vmatrix}_e$$

$$c_i = (-1)^m \begin{vmatrix} x_j & 1 & z_j \\ x_k & 1 & z_k \\ x_l & 1 & z_l \end{vmatrix}_e \quad , d_i = (-1)^m \begin{vmatrix} x_j & y_j & 1 \\ x_k & y_k & 1 \\ x_l & y_l & 1 \end{vmatrix}_e$$

where $m = j + k + l$, and $i, j, k, l \in \{1, 2, 3, 4\}$.

2.3.3 Strain-Displacement Model

The linear strain-displacement relation is defined by the following form [95]:

$$\varepsilon_e = B_e q_e \tag{2.20}$$

where ε_e is the strain vector and B_e is the strain matrix calculated by:

$$B_e = \begin{bmatrix} \dfrac{\partial N_1}{\partial x} & 0 & 0 & \cdots & \dfrac{\partial N_4}{\partial x} & 0 & 0 \\ 0 & \dfrac{\partial N_1}{\partial y} & 0 & \cdots & 0 & \dfrac{\partial N_4}{\partial y} & 0 \\ 0 & 0 & \dfrac{\partial N_1}{\partial z} & \cdots & 0 & 0 & \dfrac{\partial N_4}{\partial z} \\ \dfrac{\partial N_1}{\partial y} & \dfrac{\partial N_1}{\partial x} & 0 & \cdots & \dfrac{\partial N_4}{\partial y} & \dfrac{\partial N_4}{\partial x} & 0 \\ 0 & \dfrac{\partial N_1}{\partial z} & \dfrac{\partial N_1}{\partial y} & \cdots & 0 & \dfrac{\partial N_4}{\partial z} & \dfrac{\partial N_4}{\partial y} \\ \dfrac{\partial N_1}{\partial z} & 0 & \dfrac{\partial N_1}{\partial x} & \cdots & \dfrac{\partial N_4}{\partial z} & 0 & \dfrac{\partial N_4}{\partial x} \end{bmatrix}_e \tag{2.21}$$

By substituting (2.19) into the above equation, we get:

$$B_e = \frac{1}{6V_{equ}} \begin{bmatrix} b_1 & 0 & 0 & \cdots & b_4 & 0 & 0 \\ 0 & c_1 & 0 & \cdots & 0 & c_4 & 0 \\ 0 & 0 & d_1 & \cdots & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & \cdots & c_4 & b_4 & 0 \\ 0 & d_1 & c_1 & \cdots & 0 & d_4 & c_4 \\ d_1 & 0 & b_1 & \cdots & d_4 & 0 & b_4 \end{bmatrix}_e \tag{2.22}$$

2.3.4 Stress-Displacement Model

The relation between the stress and the nodal displacement vector is defined using the constitutive relation by the following representation [95]:

$$\sigma_e = CB_e q_e \tag{2.23}$$

where σ_e represents the stress vector.

For an isotropic material with given Young's Modulus E and Poisson's ratio ν, the elasticity matrix is defined by:

$$C = \begin{bmatrix} c_{11} & c_{12} & c_{13} & 0 & 0 & 0 \\ c_{12} & c_{22} & c_{23} & 0 & 0 & 0 \\ c_{13} & c_{23} & c_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & c_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{66} \end{bmatrix} \tag{2.24}$$

where

$$\begin{aligned} c_{11} = c_{22} = c_{33} &= \frac{E\,(1-\nu)}{(1+\nu)\,(1-2\nu)} \\ c_{12} = c_{13} = c_{23} &= \frac{E\,\nu}{(1+\nu)\,(1-2\nu)} \\ c_{44} = c_{55} = c_{66} &= \frac{E}{2\,(1+\nu)} \end{aligned}$$

With the deduced displacement relation, strain-displacement relation and stress-displacement relation, we are ready to apply the Euler-Lagrange method in order to derive the motion equation of the corresponding element.

2.3.5 Euler-Lagrange Equation

The Euler-Lagrange method is based on energy formulation, and it allows us to find the motion equation of the corresponding element of the soft robot's mesh. In general, its form can be written as follows [95]:

$$\frac{d}{dt}\left[\frac{\partial \mathfrak{L}}{\partial \dot{q_e}}\right] + \frac{\partial E_D}{\partial \dot{q_e}} = \frac{\partial \mathfrak{L}}{\partial q_e} \tag{2.25}$$

where E_D is the dissipation energy, and $\mathfrak{L}$ is the Lagrangian defined by [95]

$$\mathfrak{L}(q_e, \dot{q_e}, t) = E_K - E_P \tag{2.26}$$

with E_K and E_P representing the kinetic energy and the potential energy, respectively.

2.3.5.1 Potential Energy

The potential energy is defined by the difference between the strain energy and the potential work [95]:

$$E_P = E_S - W_P \tag{2.27}$$

2.3.5.2 Strain Energy

It is known that the strain energy is the energy stored in the body under loading, and it can be modeled as follows [95]:

$$E_S = \frac{1}{2} q_e^T K_e(q_e) q_e \tag{2.28}$$

where K_e represents the stiffness matrix for the analyzed element.

The integral form of the strain energy has the following form [95]:

$$E_S = \frac{1}{2} \int_{V_e} \varepsilon_e^T \sigma_e dv \tag{2.29}$$

where ε and σ are the strain and the stress defined in (2.20) and (2.23), respectively.

By substituting (2.20) and (2.23) into (2.29), we get:

$$E_S = \frac{1}{2} q_e^T \left[\int_{V_e} B_e^T C B_e dv \right] q_e \tag{2.30}$$

Then, from (2.28) and (2.30) we can directly get the stiffness matrix formula as follows:

$$K_e(q_e) = \int_{V_e} B_e^T C B_e dv \tag{2.31}$$

2.3.5.3 Potential Work

The potential work [95] represents the sum of the work done by the actuator forces F_e^b and the work done by the external (surface) forces F_e^s.

$$W_P = \left[F_e^b + F_e^s \right] q_e \tag{2.32}$$

with:

$$F_e^b = \int_{V_e} N_e^T f_e^b \, dv = H_e(q_e) \tau \tag{2.33a}$$

$$F_e^s = \int_{S_e} N_e^T f_e^s \, ds = F_e(q_e) \tag{2.33b}$$

where f_e^b and f_e^s are the unit body and surface forces, $H_e(q_e)$ is the actuation matrix, τ is the applied actuators force, and $F_e(q_e)$ is the external applied forces, for the corresponding element.

2.3.5.4 Kinetic Energy

The kinetic energy is the energy of the body resulting from its motion, it can be modeled as follows [95]:

$$E_K = \frac{1}{2} \dot{q_e}^T M_e(q_e) \dot{q_e} \tag{2.34}$$

where M_e represents the mass matrix for the related element.

The integral form of this kinetic energy has the following form [95]:

$$E_K = \frac{1}{2}\int_{V_e} \rho_e \dot{\delta_e}^T \dot{\delta_e} dv \tag{2.35}$$

where δ_e is the displacement defined in (2.14).

By substituting (2.14) in (2.35), we get:

$$E_K = \frac{1}{2}\dot{q_e}^T \left[\int_{V_e} \rho_e N_e^T N_e dv\right] \dot{q_e} \tag{2.36}$$

Then, from (2.34) and (2.36) we can directly get the mass matrix formula as follows:

$$M_e(q_e) = \int_{V_e} \rho_e N_e^T N_e dv \tag{2.37}$$

where ρ_e is the volumetric mass density.

2.3.5.5 Dissipation Energy

The dissipation energy is the energy due to the phenomena of damping, it can be modeled as follows [95]:

$$E_D = \frac{1}{2}\dot{q_e}^T D_e(q) \dot{q_e} \tag{2.38}$$

where D_e represents the damping matrix for the analyzed element.

The integral form of this damping energy has the following form [95]:

$$E_D = \frac{1}{2}\int_{V_e} \mu_e \dot{\delta_e}^T \dot{\delta_e} dv \tag{2.39}$$

where μ_e represents the damping coefficient, and δ_e is the displacement defined in (2.14).

By substituting (2.14) in (2.39), we get:

$$E_D = \frac{1}{2}\dot{q_e}^T \left[\int_{V_e} \mu_e N_e^T N_e dv\right] \dot{q_e} \tag{2.40}$$

Then, from (2.38) and (2.40) we can directly get the damping matrix formula as follows:

$$D_e(q_e) = \int_{V_e} \mu_e N_e^T N_e dv \tag{2.41}$$

2.3.5.6 Motion Equation

To obtain the dynamics of a single element via FEM, we replace the equations of the potential energy (2.27) and the kinetic energy (2.34) in the Lagrangian (2.26), and using (2.33), we obtain:

$$\mathfrak{L}(q_e, \dot{q_e}, t) = \frac{1}{2}\dot{q_e}^T M_e(q_e) \dot{q_e} - \frac{1}{2} q_e^T K_e(q_e) q_e + \left[H_e(q_e)\tau + F_e(q_e)\right] q_e \tag{2.42}$$

Since K_e is a symmetric matrix, thus the right term of (2.25) becomes:

$$\frac{\partial \mathfrak{L}}{\partial q_e} = -K_e(q_e) q_e + H_e(q_e)\tau + F_e(q_e) \tag{2.43}$$

The terms on the left of (2.25) depends only on $\dot{q}$, since the mass matrix M_e and the damping matrix D_e are symmetric, therefore we obtain:

$$\begin{aligned} \frac{d}{dt}\left[\frac{\partial \mathfrak{L}}{\partial \dot{q}_e}\right] &= M_e(q_e)\ddot{q}_e \\ \frac{\partial E_D}{\partial \dot{q}_e} &= D_e(q_e)\dot{q}_e \end{aligned} \tag{2.44}$$

Finally, the motion equation of the analyzed element can be written as follows:

$$M_e(q_e)\ddot{q}_e + D_e(q_e)\dot{q}_e + K_e(q)q = H_e(q_e)\tau + F_e(q_e) \tag{2.45}$$

Similarly, we apply the same procedure for each element of the mesh model, then we can assemble all elements using the connectivity matrix [96] to obtain the motion equation for the whole structure:

$$M(q)\ddot{q} + D(q)\dot{q} + K(q)q = H(q)\tau + F(q) \tag{2.46}$$

where $M(q)$ is the global mass matrix, $K(q)$ is the global stiffness matrix and $D(q)$ is the global damping matrix, and formulated below:

$$M\left(q\right) = \sum_{e=1}^{N} M_e(q) \tag{2.47a}$$

$$D\left(q\right) = \sum_{e=1}^{N} D_e(q) \tag{2.47b}$$

$$K\left(q\right) = \sum_{e=1}^{N} K_e(q) \tag{2.47c}$$

$$H\left(q\right) = \sum_{e=1}^{N} H_e(q) \tag{2.47d}$$

$$F\left(q\right) = \sum_{e=1}^{N} F_e(q) \tag{2.47e}$$

with $M_e(q)$, $D_e(q)$, $K_e(q)$, $H_e(q)$, and $F_e(q)$ are defined respectively in (2.37), (2.41), (2.31), (2.41), (2.33a), and (2.33b).

2.4 Workspace Definition

The workspace of a soft robot contains all its achievable equilibrium configurations where the end-effector can reach and stay there. In this thesis, we focus on the position aspect of the workspace, and we discuss the orientation aspect of the workspace in Chapter 7, Section 7.2.1.

For the purpose of defining the workspace of soft robots, let us note $u \in \mathbb{R}^{n_u}$ as the position of the end-effector in the inertial frame, and $q \in \mathbb{R}^n$ (n being the total number of DoFs) has a different meaning according to the appropriate modeling approach (it denotes the generalized strain vector in PCS, and the displacement vector in FEM). Then, we can define its workspace as follows:

Definition 1. *Consider a soft robot dynamically modeled by* (2.12) *for the PCS case or* (2.46) *for the FEM case, controlled by bounded actuators* $\tau \in \mathcal{T}$*, the workspace* $\mathcal{W}_{_E}$ *of the end-effector* u*, is a subspace of* $\mathbb{R}^{n_u}$*, defined below:*

$$\mathcal{W}_{_E} = \{u = \varphi(q) \mid \exists (q \in \mathbb{R}^n, \tau \in \mathcal{T}) \ s.t. \ \Phi(\tau, q) = 0\}$$

where φ *represents the geometric model of the studied soft robot,* $\Phi(\tau, q) = Kq - H(q)\tau - F(q)$ *for PCS, and* $\Phi(\tau, q) = K(q)q - H(q)\tau - F(q)$ *for FEM.*

Remark 2. *Using the above definition, It is important to state that the workspace contains all equilibrium positions where the soft robot's end-effector can reach and stay there. Moreover, the stability of a calculated configuration in the workspace can be asserted by verifying the eigenvalues of the stiffness matrix [134]. Finally, the elastic limit can be asserted by verifying that the corresponding constraint* $\Phi(\tau, q) = 0$ *is valid for the corresponding* $\tau \in \mathcal{T}$*.*

In this thesis, we adopt two strategies to estimate the workspace of soft robots;

- Discretizing the inputs (actuators) space $\mathcal{T}$ [4, 24, 26, 27, 132].
- Discretizing the outputs (end-effector) space [3, 48, 55, 78, 116].

The first strategy is manifested through the so-called forward approach [4, 24, 27, 132] (see Section 2.5), and the second strategy is manifested through three different approaches; optimization [123] (see Chapter 3), interval analysis (see Chapter 4) [7], and continuation [8] (see Chapter 5), which will be established in what follows.

2.5 Forward Approach

2.5.1 Forward Approach Introduction

The most basic approach to determine the workspace of soft robots is to iteratively approximate the input-output map $\phi : \mathcal{T} \rightarrow \mathcal{W}_{_E}$ through the discretization of the actuators bound $\mathcal{T}$ with a prescribed precision n_d, and then determine for each actuator value its corresponding end-effector position, as illustrated by Fig. 2.5.

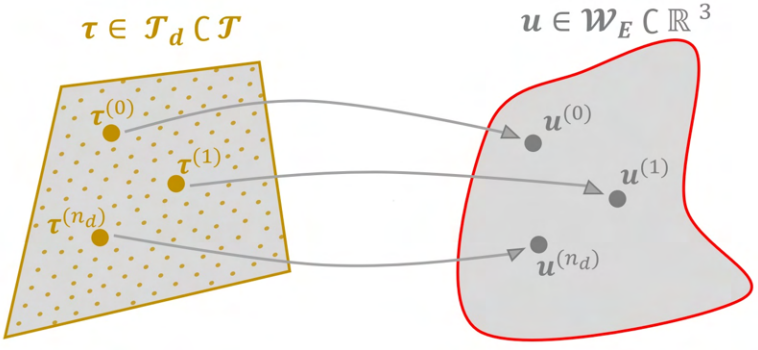


Figure 2.5: Forward Approach

In the following, we will explain the details of the forward approach for both the PCS and the FEM modeling methods.

2.5.2 Forward Approach for Workspace Estimation: PCS case

According to Definition 1, the workspace of a soft robot contains all equilibrium points. Applying this property to equation (2.12), the workspace of the studied soft robot is determined by the following equations:

$$K\, q = H(q)\tau + F\,(q) \tag{2.48a}$$

$$u(q) = \mathfrak{E}_1\; g(q)\; \mathfrak{E}_2^T \tag{2.48b}$$

with

$$g(q) = e^{L_1\hat{\xi}_1} \cdots e^{L_N\hat{\xi}_N} \tag{2.49a}$$

$$\hat{\xi}_i = \widehat{B_{q_i} q_i}\;;\;\; i \in 1 \cdots N \tag{2.49b}$$

The purpose of $\mathfrak{E}_1$ and $\mathfrak{E}_2$ in (2.48b) is to extract the end-effector position u from the configuration matrix g, where $\mathfrak{E}_2$ is a constant elementary vector defined as $\mathfrak{E}_2 = [0_{1\times 3}1]$, and $\mathfrak{E}_1$ is a constant elementary matrix defined by the following: when $n_u = 2$, it may take the form of $\mathfrak{E}_1 = [I_2, 0_2]$, or $\mathfrak{E}_1 = [[1,0,0,0]^T, [0,0,1,0]^T]^T$ (depending on the actuators location); when $n_u = 3$, $\mathfrak{E}_1 = [I_3\ 0_{3\times 1}]$ (with I and 0 are respectively the identity and zero matrices).

Using the PCS framework, the problem of estimating $\mathcal{W}_{_E}$ is equivalent to, for all admissible $\tau \in \mathcal{T}$, measure all the possible configurations of its end-effector u defined in (2.48b). However, this is not trivial, mainly due to the reason that the mapping $\phi : \mathcal{T} \to \mathcal{W}_{_E}$ is implicit and nonlinear, thus no analytic ϕ can be found such that $u = \phi(\tau)$.

In practice, for the purpose of estimating $\mathcal{W}_{_E}$ from the bounded actuators $\tau \in \mathcal{T}$, we might use numerical methods to approximate the mapping ϕ. In the following, we may use the gradient of the generalized strain vector q with respect to the actuators force vector τ, denoted as $\nabla_\tau(q)$, which will be analyzed hereafter.

By definition, the gradient $\nabla_\tau(q)$ is calculated by:

$$\nabla_\tau(q) = \Big[\frac{\partial q}{\partial \tau}\Big]^T$$

Thus, by calculating the partial derivative of (2.48a) with respect to the actuators vector τ, we have:

$$K\frac{\partial q}{\partial \tau} = \frac{\partial [H(q)\tau]}{\partial \tau} + \frac{\partial [F(q)]}{\partial \tau}$$

Using the principle of variable separation, the above equation is equivalent to:

$$K\frac{\partial q}{\partial \tau} = \frac{\partial [H(q)]}{\partial q}(I_n \otimes \tau)\frac{\partial q}{\partial \tau} + H(q) + \frac{\partial [F(q)]}{\partial q}\frac{\partial q}{\partial \tau}$$

From this we get:

$$\left(K - \frac{\partial [H(q)]}{\partial q}(I_n \otimes \tau) - \frac{\partial [F(q)]}{\partial q}\right)\frac{\partial q}{\partial \tau} = B(q)$$

where the symbol $\otimes$ is the Kronecker tensor product. Let us define the matrix $\mathfrak{B}$ as follows for the sake of simplicity:

$$\mathfrak{B}(\tau, q) = K - H_q(q)(I_n \otimes \tau) - F_q(q) \tag{2.50}$$

where

$$H_q\,(q) = \frac{\partial [H(q)]}{\partial q}\;,\; F_q\,(q) = \frac{\partial [F(q)]}{\partial q}$$

Obviously, the calculation of $\nabla_\tau(q)$ depends on the invertibility of the matrix $\mathfrak{B}$ in (2.50), which in turns is indirectly determined by the generalized actuation matrix H in (2.10), and the vector of generalized position-dependent external force F in (2.13f).

Therefore, the following assumption has been imposed in this thesis.

Assumption 1. *For the studied soft robot, it is supposed that the actuators are installed in such a way ensuring that the matrix $\mathfrak{B}$ defined in (2.50) is invertible for all $q \in \mathcal{W}_{_E}$.*

Remark 3. *Physically, the above assumption means that the actuators of the investigated soft robots are arranged in a way that a small actuators variation for a feasible configuration $q \in \mathcal{W}_{_E}$ might locally and uniquely determine the relative variation of position. In other words, it implies that the mounted actuators are installed such that they are not mounted on each other. If such an assumption is violated, i.e., several actuators are mounted in a redundant way, then it is necessary to keep only* 1 *actuator and remove other ones.*

Hence, if Assumption 1 is satisfied, we can then obtain:

$$\frac{\partial q}{\partial \tau} = \left(K - H_q(q)(I_n \otimes \tau) - \frac{\partial [F(q)]}{\partial q} \right)^{-1} H(q)$$

which means $\nabla_\tau(q) = H^T(q) \left[\mathfrak{B}^{-1}(\tau, q) \right]^T$.

As stated earlier in the introduction of this chapter (Section 2.5.1), the straightforward approach to estimate the workspace via (2.48) is to firstly discretize the actuators bounds $\mathcal{T}$ (as indicated in Fig. 2.5), and then approximate the mapping ϕ in an iterative manner.

Precisely, after discretizing the actuators force bounds $\mathcal{T}$, noted as $\mathcal{T}_d$, then for each $\tau \in \mathcal{T}_d$, with the knowledge of the end-effector position $u(q^{(j-1)})$ and the generalized strain vector $q^{(j-1)}$ (for the case $j = 1$, $u(q^{(0)})$, $q^{(0)}$ represents the end-effector's initial position and the initial generalized strain vector of the studied soft robot respectively), the following proposition enables us to calculate the next generalized strain vector $q^{(j)}$.

Proposition 1. *If Assumption 1 is satisfied, then for a given $\tau \in \mathcal{T}_d$ the following dynamics:*

$$q^{(j)} = q^{(j-1)} - \mathfrak{B}^{-1}(\tau, q^{(j-1)})\ \Phi(\tau, q^{(j-1)}) \tag{2.51}$$

exponentially converges to the corresponding solution of q in (2.48a)*, with $\mathfrak{B}$ defined in* (2.50) *and*

$$\Phi(\tau, q^{(j-1)}) = Kq^{(j-1)} - H(q^{(j-1)})\tau - F(q^{(j-1)}).$$

Remark 4. *It is important to state that convergence towards the closest solution with reference to the initial solution can only be guaranteed if the conditions resulting from Kantorovich's theorem [89] are respected.*

Proof. Solving (2.48a) numerically can be achieved by minimizing the following quadratic cost function $S(\tau, q^*)$:

$$\begin{aligned} S(\tau, q^*) &= \Phi^T(\tau, q^*)\Phi(\tau, q^*), \text{ where} \\ \Phi(\tau, q^*) &= Kq^* - H(q^*)\tau - F(q^*) \end{aligned}$$

First, we compute the time derivative of the cost function:

$$\dot{S}(\tau, q^*) = 2\Phi^T(\tau, q^*)\dot{\Phi}(\tau, q^*)$$

Next, we apply the principle of variable separation, and we obtain:

$$\begin{aligned}\dot{S}(\tau, q^*) &= 2\Phi^T(\tau, q^*)\frac{\partial[\Phi(\tau, q^*)]}{\partial q^*}\dot{q}^* \\ &= 2\Phi^T(\tau, q^*)\mathfrak{B}(\tau, q^*)\dot{q}^*\end{aligned}$$

with $\mathfrak{B}$ defined in (2.50).

In order to minimize the cost function $S(\tau, q^*)$, we need to define an adaptive law $\dot{q}^*$ such that $\dot{S}(\tau, q^*)$ is negative. For this, and in accordance with Assumption 1, if the following adaptive law is chosen:

$$\dot{q}^* = -\gamma\mathfrak{B}^{-1}(\tau, q^*)\Phi(\tau, q^*) \tag{2.52}$$

where γ is a positive parameter, then we can obtain:

$$\dot{S}(\tau, q^*) = -2\gamma S(\tau, q^*)$$

which implies that $S(\tau, q^*)$ exponentially converges to 0. Equivalently, it means that q^* exponentially converges to q with the proposed continuous-time adaptive law (2.52). Finally, the discrete-time version of (2.52) can be written as:

$$q^{(j)} = q^{(j-1)} - \Delta t\gamma\mathfrak{B}^{-1}(\tau, q^{(j-1)})\ \Phi(\tau, q^{(j-1)})$$

and we prove Proposition 1 by choosing $\gamma = \dfrac{1}{\Delta t}$. □

After, with $q^{(j)}$ calculated via (2.51), we compute the strain twist $\hat{\xi}_i^{(j)}$ related to the strain vector $q_i^{(j)}$ of each body i as follows:

$$\hat{\xi}_i^{(j)} = \widehat{B_{q_i} q_i^{(j)}}\ ;\ \ i \in 1 \cdots N$$

with which the configuration matrix $g(q^{(j)})$ associated to the generalized strain vector $q^{(j)}$ can be calculated as:

$$g(q^{(j)}) = e^{L_1\hat{\xi}_1^{(j)}} \cdots e^{L_N\hat{\xi}_N^{(j)}}$$

Then, we can derive the corresponding end-effector position as follows:

$$u(q^{(j)}) = \mathfrak{E}_1\ g(q^{(j)})\ \mathfrak{E}_2^T$$

Finally the estimation of $\mathcal{W}_{_E}$ can be successfully accomplished by iterating the above procedure for all feasible inputs $\tau \in \mathcal{T}_d$ from the initial state $j = 1$ to the last iteration $j = n_d$.

2.5.3 Forward Approach for Workspace Estimation: FEM case

In accordance with the workspace definition (see Definition 1), the workspace of a soft robot can finally be defined by the following set of equations:

$$K(q)\ q = H(q)\tau + F(q) \tag{2.53a}$$

$$q_{_E} = \mathfrak{C}q \tag{2.53b}$$

$$u = q_{_E} + u^{(0)} \tag{2.53c}$$

where $q_{_E}$ is the displacement vector of the end-effector with respect to its known initial position vector $u^{(0)}$, u represents the position vector of the end-effector, and $\mathfrak{C} \in \mathbb{R}^{3\times 3n}$ is a selection matrix associated with the end-effector node coordinates.

To achieve the straightforward approach, given the information of the end-effector's position vector $u^{(j-1)}$ and the vector of nodal displacements $q^{(j-1)}$ (for the case $j = 1$, $u^{(0)}$ and $q^{(0)}$ are respectively the initial position of the end-effector and the initial nodal displacement vectors of the investigated soft robot), we apply Newton-Raphson method to obtain the next nodal displacement $q^{(j)}$ corresponding to its actuators value $\tau \in \mathcal{T}_d$, with $\mathcal{T}_d$ being the discretized set of $\mathcal{T}$, which can be realized as follows:

$$q^{(j)} = q^{(j-1)} - \Phi_q(u, q^{(j-1)})\ \Phi(\tau, q^{(j-1)}), \forall \tau \in \mathcal{T}_d$$

where $\Phi_q(\tau, q)$ is the gradient of $\Phi(\tau, q)$ with respect to q, and $\Phi(\tau, q^{(j-1)}) = K(q^{(j-1)})q^{(j-1)} - H(q^{(j-1)})\tau - F(q^{(j-1)})$. Then, the corresponding end-effector position vector can be deduced by:

$$u^{(j)} = \mathfrak{C}q^{(j)} + u^{(0)}$$

Finally, as depicted by Fig. 2.5, the estimation of the reachable workspace $\mathcal{W}_E$ can be achieved by iterating the above process for all $\tau \in \mathcal{T}_d$.

2.5.4 Experimental Validation

The goal of this section is to validate the workspace estimation obtained by applying the forward approach on the PCS and FEM models with the real workspace obtained from the experimental setup using a trunk-like soft robot [130] (as depicted by Fig. 2.6) for both the PCS and FEM models.

The exact geometric parameters of the investigated trunk-like soft robot are illustrated in Fig. 2.6. The studied soft robot is controlled by 4 independent symmetric tendons (2 upward and 2 lateral) mounted through it from the base to the tip, and the tendons are actuated by 4 independent stepper motors. To obtain the value of the end-effector position, a sensor position was positioned on the tip of the robot, and for this purpose, a long and uniform hole was made along the whole length of the robot. Moreover, 3D-printed rigid rings (blue rings in Fig. 2.6) are mounted along the soft robot to minimize friction between the tendons and the soft rubber. Finally, the casting material of the investigated trunk-like robot is an isotropic silicone rubber with its Young's modulus $E = 6.5 \times 10^5 Pa$ and a Poisson's ration $\nu = 0.45$.

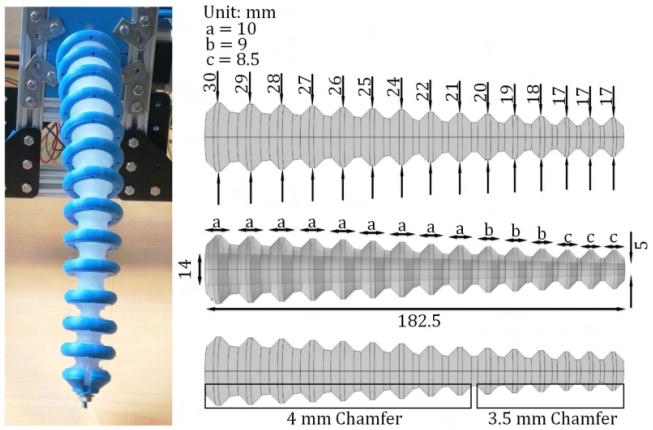
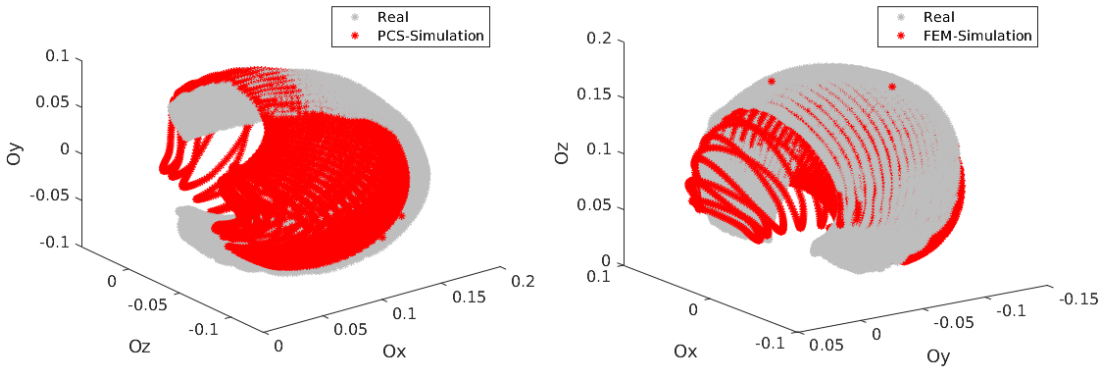


Figure 2.6: Trunk-like soft robot [129].

Using the forward approach, the real workspace can be achieved by reading the value of the position of the end-effector for each value of τ. The actuators bounds were within a magnitude $\mathcal{T} = [0, 5]_1 \times \cdots \times [0, 5]_4 N$, and the obtained experimentation result of the workspace is depicted by gray-colored points in Fig. 2.7.

On the other hand, in a simulation environment, we apply both the PCS and FEM methods to model the investigated trunk-like soft robot. The studied soft robot was modeled by the PCS method using $N = 15$ cylindrical bodies with equal lengths and decreasing radius from the base to the tip of the robot. Using the FEM method with $N = 5024$ tetrahedral elements and $n = 1484$ nodes, the investigated soft robot was modeled. The tendons in the simulation environment were approximately positioned using their real coordinates. Based on the adopted mathematical models, we apply the forward approach to get the corresponding workspace estimation. The obtained simulation results of the workspace for both the PCS and FEM cases are depicted by red-colored points in Figs. 2.7a and 2.7b, respectively. The result of the workspace obtained from the real experimentation took approximately 360 seconds, while the results of the workspace obtained via the simulations took approximately 50 seconds.



(a) $\mathcal{W}_E$: Real (gray) and PCS-simulation (red). (b) $\mathcal{W}_E$: Real (gray) and FEM-simulation (red).

Figure 2.7: Forward approach - $\mathcal{W}_E$ obtained from the real experimentation (gray) and $\mathcal{W}_E$ achieved by the simulations of the adopted mathematical models (red).

Visually, we can observe from Fig .2.7, the workspace estimation results estimated by applying the forward approach on the PCS and FEM case approximately matches the workspace result obtained from the real experimentation. Numerically, we compare the volume of the set of points obtained from the real experiment V_r with the set of points obtained via the PCS (noted V_s^{PCS}) and FEM (noted V_s^{FEM}) simulation results. The volume calculated from the set of points obtained by the real robot is $V_r \approx 0.0021m^3$, and the volumes obtained the PCS and FEM simulation results are $V_s^{PCS} \approx 0.0028m^3$ and $V_s^{FEM} \approx 0.0025m^3$, respectively. Then, by comparing both results, we find $|V_r - V_s^{PCS}| \approx 7.3 \times 10^{-4}m^3$ for the accuracy of the PCS result and $|V_r - V_s^{FEM}| \approx 4.8 \times 10^{-4}m^3$ for the accuracy of the FEM result.

Based on this, and in the following chapters, the workspace obtained via the forward approach will be considered as a reference. Moreover, the workspace estimation achieved by the approaches that will be presented in the next chapters (Chapter 3, Chapter 4, and Chapter 5) will be superimposed and compared with the workspace estimation obtained from the forward approach

in order to highlight their validity and efficiency.

2.6 Conclusion

In this chapter, we have presented the mathematical models of two adopted modeling methods, PCS and FEM. Next, we have presented the definition of the workspace of a soft robot. Finally, we have proposed a forward approach to estimate the workspace of soft robots based on PCS and FEM, and we have used a trunk-like soft robot to validate the workspace estimation obtained via the forward approach for the PCS and FEM with the real workspace result.

However, the main disadvantage of the forward approach lies in its computation complexity which can be very high.

Given n_d subsets ($n_d \in \mathbb{N}$), since $\tau \in \mathcal{T} \subset \mathbb{R}^{n_\tau}$ ($n_\tau \in \mathbb{N}$), for each entry of τ we will obtain $n_d^{n_\tau}$ possible combinations of input for τ. Based on this perspective, the computation complexity of the forward approach depends on two parameters, n_d and n_τ. To obtain a correct estimation of $\mathcal{W}_{_E}$, the discretization precision n_d needs to be sufficiently big, which will result in a significant increase in the computation complexity. Moreover, it will be exponentially exploded when the number of inputs n_τ becomes larger.

Note that the base of the computation complexity for the forward approach equals n_τ (the inputs' dimension), which varies according to different configurations of soft robots. Whereas, the dimension of the end-effector space is constant ($n_u \leqslant 3$; for the position case) because it is independent of any configuration of the investigated soft robot. Based on this observation, to reduce the computation complexity when estimating the workspace of soft robots, it is therefore, reasonable to propose novel methods that are based on discretizing the output space, i.e. the end-effector space u, which is invariant in dimension, instead of discretizing the input space, i.e., the actuators space $\mathcal{T}$ with variant dimension.

As stated in Section 2.5.4, the forward approach will be used as a reference to validate the workspace methodologies presented in the following chapters.

Chapter 3

Optimization-Based Approach for Workspace Estimation

3.1 Introduction

Due to the high computation complexity of the forward approach when estimating the workspace of soft robots, it is logical to propose an approach that can discretize the end-effector space since its dimension is smaller and constant compared to the actuators space for hyper-redundant robots, such as soft robots. In addition, it is more efficient to only map the exterior boundary of the workspace and avoid the heavy computation of its interior points.

Such a method is called the optimization-based approach [116] in this thesis and it consists of discretizing the end-effector space in order to map the exterior boundary of $\mathcal{W}_{_E}$, noted as $\partial\mathcal{W}_{_E}$, and deducing that the area/volume enclosed by $\partial\mathcal{W}_{_E}$ is the reachable workspace $\mathcal{W}_{_E}$.

Generally speaking, this method consists of numerical algorithms used to map the surface enveloping soft robots' workspaces [9, 123], which will be realized by the following steps:

1. In the first step, we establish the definition of a soft robot's workspace boundary, denoted as $\partial\mathcal{W}_{_E}$.

2. Next, consistent with the workspace boundary definition, we seek to find a boundary point u^b on $\partial\mathcal{W}_{_E}$. The strategy proposed consists of selecting a radiating point v, then from this point, a ray is emanated along a certain direction until $\partial\mathcal{W}_{_E}$ is met (as illustrated by Fig 3.1).

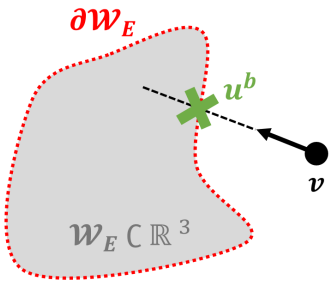


Figure 3.1: For a radiating vector v determine its associated boundary point u^b.

3. Finally, a map of $\partial\mathcal{W}_{_E}$ can be achieved by solving the procedure that consists of finding u^b for multiple successive rays with respective directions emanating at angular intervals.

Unlike the forward method (Section 2.5), the computation complexity of the proposed optimization-based approach depends on two factors: the dimension of the space to be discretized, and the discretization precision. By managing these two parameters, the optimization-based approach can reduce the complexity of the workspace estimation.

1. It can reduce the dimension of the space to be discretized in the forward approach. In the forward approach, we discretize the actuators space whose dimension n_τ is varying according to the soft robot structure, but the space to be discretized in the optimization-based approach is always less than or equal to 3 for the position case;

2. When discretizing the space with a prescribed precision (with n_d sub-spaces), the forward method cannot reduce the value of n_d, while the optimization-based approach, which consists only in finding then mapping the boundary points u^b, results in reducing the total number of iterations.

Briefly, the proposed optimization-based approach can be summarized into the following procedure:

[Step 1] Defining the workspace boundary $\partial\mathcal{W}_{_E}$.

[Step 2] Finding a point on $\partial\mathcal{W}_{_E}$.

[Step 3] Mapping the workspace boundary $\partial\mathcal{W}_{_E}$.

In the following, we present the detailed procedure of the optimization-based approach for both the PCS [123] and the FEM [9] modeling methods.

3.2 Optimization Approach for the PCS case

3.2.1 Workspace Boundary $\partial\mathcal{W}_{_E}$ Definition

Given a configuration of a soft robot, with its workspace $\mathcal{W}_{_E}$ defined in Definition 1, and $\tau \in \mathcal{T} = [\underline{\tau}, \overline{\tau}]$, $\forall u \in \mathcal{W}_{_E}$, we can then denote the neighborhood of u with a specified positive radius ε by:

$$B_\varepsilon(u) = \{\ u_\varepsilon \in \mathbb{R}^{n_u} \mid \|u - u_\varepsilon\| < \varepsilon\ \} \tag{3.1}$$

based on which the workspace boundary $\partial\mathcal{W}_{_E}$ according to [116] can be defined below.

Definition 2. *The workspace boundary $\partial\mathcal{W}_{_E}$ is a subset of its workspace $\mathcal{W}_{_E}$, and defined by the following:*

$$\partial\mathcal{W}_{_E} = \{u \in \mathcal{W}_{_E} \mid \forall\varepsilon > 0, \exists u_\varepsilon \in B_\varepsilon(u), s.t.\ u_\varepsilon \notin \mathcal{W}_{_E}\}$$

Remark 5. *To address the mapping of $\partial\mathcal{W}_{_E}$, the strategy proposed is based on solving a constrained minimization problem defined in Section 3.2.2 of the optimization-based approach procedure. For this, we first need to calculate the gradient of the end-effector position $u(q)$ with respect to the actuators vector τ, denoted as $\nabla_\tau\big[u(q)\big]$, and established in the following theorem.*

Theorem 1. *For a given controllable configuration of the soft robot, with $\nabla_\tau(q)$ defined in Assumption 1, the geometric Jacobian $J(q)$ defined in (2.7a), and the orientation matrix $R(q)$ computed via the definition of the configuration matrix $g(q)$ through the following:*

$$R(q) = \mathfrak{E}_1 \; g(q) \; \mathfrak{E}_1^T \tag{3.2}$$

then the gradient of the end-effector position $u(q)$ with respect to the actuators vector τ, denoted as $\nabla_\tau\big[u(q)\big]$, is established by:

$$\nabla_\tau\big[u(q)\big] = \nabla_\tau(q) J^T(q) [0_3 \; R(q)]^T \tag{3.3}$$

with $\nabla_\tau(q) = H^T(q)\left[\mathfrak{B}^{-1}(\tau, q)\right]^T$ being defined in Assumption 1, and $\mathfrak{B}(\tau, q, L, \alpha) = K(L, \alpha) - H_q(q, L, \alpha)(I_n \otimes \tau) - F_q(q, L, \alpha)$.

Proof. Using the principle of variable separation, the gradient of the end-effector position $\nabla_\tau\big[u(q)\big]$ can be written as follows:

$$\nabla_\tau\Big[u(q)\Big] = \Big[\frac{\partial u(q)}{\partial \tau}\Big]^T = \Big[u_q(q) \; \big[\nabla_\tau(q)\big]^T\Big]^T$$

Based on the velocity twist vector $\hat{\eta}(q)$ defined in (2.7c), we can derive the linear velocity $\mathsf{v}(q)$ via the following:

$$\mathsf{v}(q) = \dot{u}(q) = \mathfrak{E}_1 \; \hat{\eta}(q) \; \mathfrak{E}_2^T, \text{ with } \hat{\eta}(q) = g^{-1}(q) \; \dot{g}(q)$$

Next, we develop the term $\hat{\eta}(q)\mathfrak{E}_2^T$:

$$\hat{\eta}(q) \; \mathfrak{E}_2^T = g^{-1}(q) \; \dot{g}(q) \; \mathfrak{E}_2^T = g^{-1}(q) \; \frac{\partial g(q)}{\partial q} \; (I_4 \otimes \dot{q}) \; \mathfrak{E}_2^T$$

Using the definition of the configuration matrix $g(q)$, we develop the above equation, and we obtain:

$$\begin{aligned}\hat{\eta}(q) \; \mathfrak{E}_2^T &= \begin{bmatrix} R^{-1}(q) & R^{-1}(q)u(q) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{\partial R(q)}{\partial q} & \frac{\partial u(q)}{\partial q} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ \dot{q} \end{bmatrix} \\ &= \Big[R^{-1}(q) \frac{\partial u(q)}{\partial q} \dot{q} \quad 0\Big]^T\end{aligned}$$

Substituting the above equation of $\hat{\eta}(q)\mathfrak{E}_2^T$ in the linear velocity $\dot{u}(q)$, we achieve the following expression of $\dot{u}(q)$:

$$\dot{u}(q) = \mathfrak{E}_1 \; \hat{\eta}(q) \; \mathfrak{E}_2^T = R^{-1}(q) \; \frac{\partial u(q)}{\partial q} \; \dot{q} \tag{3.4}$$

Using the relation between the velocity vector $\eta(q)$ and the geometric Jacobian $J(q)$ defined by (2.7a), we derive another form of the linear velocity $\dot{u}(q)$, which can be formulated as follows:

$$\eta(q) = \big[w(q) \; \dot{u}(q)\big]^T = J(q) \; \dot{q}$$

which gives:

$$\dot{u}(q) = [0_3 \; I_3] \; J(q) \; \dot{q} \tag{3.5}$$

Using the expressions (3.4) and (3.5), we deduce the following equation:

$$R^{-1}(q) \; \frac{\partial u(q)}{\partial q} \; \dot{q} = [0_3 \; I_3] \; J(q) \; \dot{q}$$

Then, we get:

$$u_q(q) = \frac{\partial u(q)}{\partial q} = R(q)\ [0_3\ I_3]\ J(q) = [0_3\ R(q)]\ J(q)$$

Finally, by substituting $u_q(q)$ in $\nabla_\tau\big[u(q)\big]$, we deduce the following:

$$\nabla_\tau\big[u(q)\big] = \nabla_\tau(q)\big[u_q(q)\big]^T = \nabla_\tau(q) J^T(q)[0_3\ R(q)]^T$$

□

3.2.2 Finding a point on $\partial\mathcal{W}_E$

In order to map the surface enveloping the workspace $\mathcal{W}_E$, we need first to figure out how to determine a boundary point u^b. For concrete applications, it is a nontrivial task to find a point on the workspace boundary $\partial\mathcal{W}_E$ (as shown in Fig. 3.2). To clearly visualize this problem, we present the following simplified illustration (see Fig. 3.2).

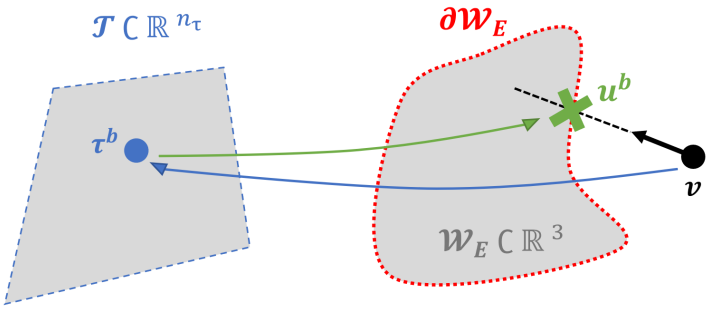


Figure 3.2: For a radiating vector v, calculate its corresponding actuators τ^b via (3.6), and deduce its associated boundary point u^b via (2.48).

Assume that a radiating point v is selected, and that v is exterior to the workspace $\mathcal{W}_E$. It is now proposed that, consistent with Definition 2, a boundary point u^b in a particular direction emanating from the radiating point v (as depicted by Fig. 3.2), can be determined by solving the following constrained optimization problem [116]:

$$\begin{aligned} \tau^b = \arg\min_{\tau} \quad & \|u(q) - v\|_2^2 = f(q) \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K\ q = H(q)\tau + F\,(q) \\ & u(q) = \mathfrak{E}_1\ g(q)\ \mathfrak{E}_2^T \end{aligned} \tag{3.6}$$

where τ^b is the optimal and feasible actuators vector (since it satisfies the bounded constraint) for the purpose of minimizing the nonlinear cost function $f(q) = \|u(q) - v\|_2^2$, which represents the distance between v and $u(q)$.

Based on the solution of τ^b, and in accordance with Proposition 1, we calculate the corresponding generalized strain vector q^b through (2.51). Then, we compute the strain twist $\hat{\xi}_i^b$ related to the strain vector q_i^b of each segment i using (2.49b). Next, we derive the configuration matrix $g(q^b)$ associated to the generalized strain vector q^b from (2.49a). Finally, using (2.48b), we deduce the boundary point position u^b.

Regarding the implementation of the methodology, we need to compute the gradient of the objective function $f(q)$ with respect to τ, denoted by $\nabla_\tau\big(f(q)\big)$, which can be formulated as follows:

$$\nabla_\tau\big(f(q)\big) = 2\nabla_\tau\big[u(q)\big]\ \big(u(q) - v\big) \tag{3.7}$$

with $\nabla_\tau\big[u(q)\big]$ defined in (3.3) .

Another additional question linked to the implementation of the proposed methodology is how a radiating point v may be generated. As depicted by Fig. 3.2, the radiating point v must be exterior to the workspace. In practice, since the length L of a soft robot is bounded ($L \leqslant \overline{L}$), we only need to choose v such that $||\vec{Ov}|| > \overline{L}$, where $O = [0, 0, 0]^T$ is the origin of the inertial frame.

3.2.3 Mapping the Workspace Boundary $\partial\mathcal{W}_{_E}$

In this section, we present a numerical methodology used for mapping the workspace boundary $\partial\mathcal{W}_{_E}$. Solving this problem is equivalent to finding a boundary point u^b for multiple successive directions (as indicated in Fig. 3.3). In the following, we represent a simplified illustration of mapping the workspace boundary (see Fig. 3.3).

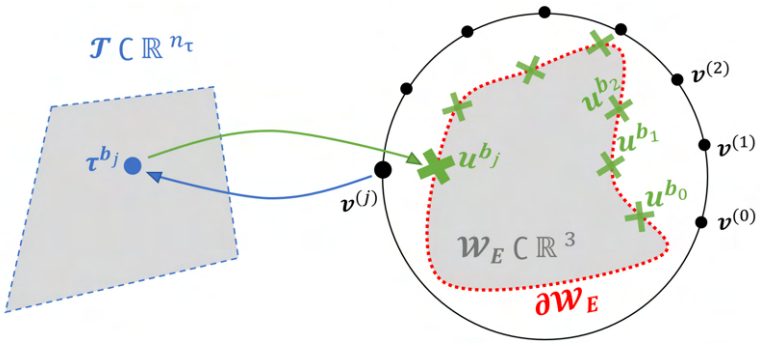


Figure 3.3: Numerical map of $\partial\mathcal{W}_{_E}$ - for each radiating vector $v^{(j)}$, compute its corresponding actuators τ^{b_j} via (3.8), and deduce its associated boundary point u^{b_j} via (2.48).

The workspace boundary $\partial\mathcal{W}_{_E}$ may be numerically mapped by solving the optimization problem (3.8) for n_d successive rays, with respective directions $v^{(j)}$, $j = 0, 1, 2, \ldots, n_d$ emanating at angular intervals of the angles δ_z, δ_x and δ_y (with $\delta_{z,x,y} = 2\pi/n_d$) from a radiating point v_0 [116]:

$$\begin{aligned} \tau^{b_j} = \min_{\tau} \quad & \|u(q) - v^{(j)}\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K\ q = B(q)\tau + F\,(q) \\ & u(q) = \mathfrak{E}_1\ g(q)\ \mathfrak{E}_2^T \end{aligned} \tag{3.8}$$

where τ^{b_j} is the optimal and feasible actuators force vector in order to minimize the distance between $v^{(j)}$ and $u(q)$.

From the solution τ^{b_j}, and according to Proposition 1, we calculate the corresponding generalized strain vector q^{b_j} (2.51). Then, we compute the strain twist $\hat{\xi}_i^{b_j}$ related to the strain vector $q_i^{b_j}$ of each segment i using (2.49b), and based on which we derive the configuration

matrix $g(q^{b_j})$ associated to the generalized strain vector q^{b_j} from (2.49a). Finally, we deduce the boundary point position u^{b_j} using (2.48b).

The direction vector $v^{(j)}$ is expressed by the following:

$$v^{(j)} = E_{\delta}^{(j)}\, v_0$$

where the initial radiating point v_0 is exterior to the workspace and can be calculated using the same technique suggested in Section 3.2.2. The rotation matrix $E_{\delta}^{(j)}$ is defined as: $E_{\delta}^{(j)} = E_{\delta_z^{(j)}} E_{\delta_x^{(j)}} E_{\delta_y^{(j)}}$, and $E_{\delta_{z,x,y}^{(j)}}$ represents the basic rotation matrices about the z, x and y axis respectively [52].

Up to this point, the procedure of the optimization-based approach has been completely described in each step with its related specifics.

3.2.4 Convexity Issues during the Mapping of $\partial\mathcal{W}_E$

Going a step further, difficulties may now arise in relation to some details of the methodology implementation, and it concerns the strategy to be adopted if non-convexity of the workspace $\mathcal{W}_E$ interferes with the mapping of the workspace boundary $\partial\mathcal{W}_E$ as illustrated in Fig. 3.4, where as a result of the non-convexity problem, the workspace may be over-estimated.

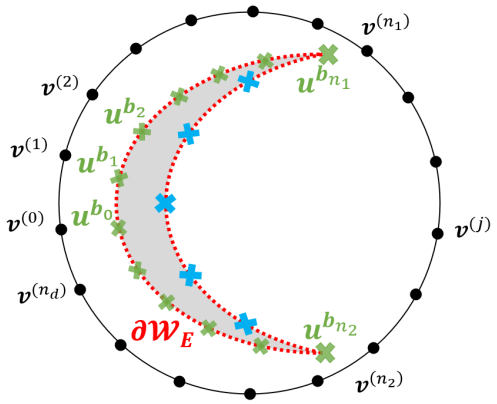


Figure 3.4: Non-convexity problem of $\partial\mathcal{W}_E$ - sudden leap between the deduced boundary points $u^{b_{n_1}}$ and $u^{b_{n_2}}$.

Eventually, in this particular situation, mapping the boundary $\partial\mathcal{W}_E$ may result in an over-estimation of the workspace since all cyan colored boundary points in Fig. 3.4 between the successfully determined boundary points $u^{b_{n_1}}$ and $u^{b_{n_2}}$, may not be identified, due to fact that, for all admissible direction vectors $v^{(j)}$ between $v^{(n_1)}$ and $v^{(n_2)}$, the only possible solutions that can be achieved by the minimization problem (3.8) will be either $u^{b_{n_1}}$ or $u^{b_{n_2}}$, i.e.,

$$\forall v^{(j)} \in [v^{(n_1)}\ v^{(n_2)}],\ (3.8) \implies u^{b_{n_1}} \vee u^{b_{n_2}}$$

To solve this specific problem, we need first to understand the phenomenon leading to the occurrence of this situation. Actually, this particular behavior is mainly due to a sudden leap between two successive iterations, either gradual or declining, of at least one actuators force $\tau_k^{b_j} \in \tau^{b_j}$ of the optimization problem (3.8).

In this thesis, and consistent with the above observation, the strategy proposed to solve this particular problem is firstly based on discretizing the space between the boundary points $u^{b_{n_1}}$ and $u^{b_{n_2}}$, based on which we obtain a list of radiating points v_l. In the next step, for each radiating point $v_l \in [u^{b_{n_1}}\ u^{b_{n_2}}]$ we proceed by solving the minimization problem (3.6), which allows us to obtain the feasible actuators force vector τ^b corresponding to each radiating point v_l. Next, by solving (2.48), we can identify the boundary points u^b on the non-convex part of the workspace (See the cyan colored boundary points in Fig. 3.4). Finally, by mapping all boundary points, we obtain the correct estimation of the workspace (as shown in Fig. 3.5).

Briefly, the proposed strategy to solve the non-convexity problem can be summarized into the following procedure:

[Step. A] Calculate v_l by discretizing the space between the boundary points $u^{b_{n_1}}$ and $u^{b_{n_2}}$.

[Step. B] For each discretized point v_l, find its related feasible actuators vector by solving (3.6), then deduce its corresponding boundary point from (2.48).

In the following illustration (see Fig. 3.5), we present a simplified figure describing two iterations of the proposed strategy for solving the non-convexity problem.

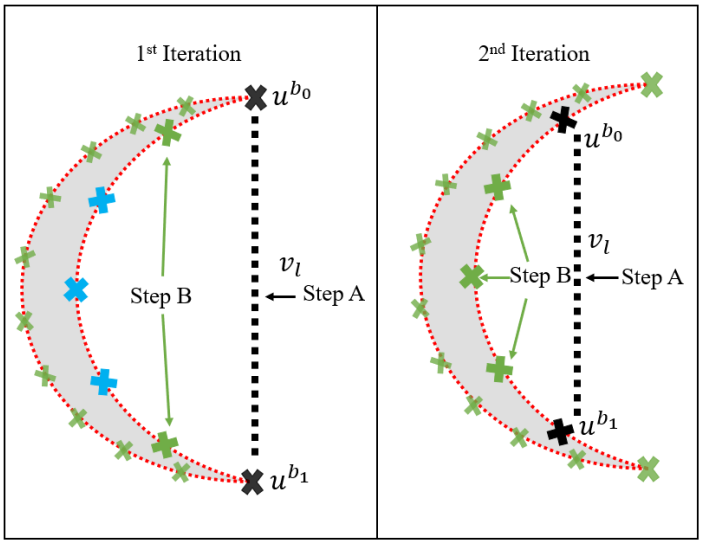


Figure 3.5: Non-convexity solution - for each point v_l, calculate its related actuators via (3.6), and deduce its associated boundary point via (2.48).

In the following, a brief algorithm is presented to describe the main steps of the optimization-based approach along with the adopted strategy to solve the non-convexity problem (See Algorithm 1).

3.2.5 Validation and Analysis

In this section, we will illustrate the presented results by evaluating the workspace for different configurations of tendon-actuated soft robots. We consider various scenarios, and we implement both the forward and optimization-based approaches to get $\mathcal{W}_E$ and $\partial\mathcal{W}_E$ respectively.

Algorithm 1 Calculate $\partial\mathcal{W}_{_E}$

Require: $v_0, \mathcal{T}, \bar{\mathcal{F}}_{e_i}, F^{(0)}, \mathfrak{E}_1, \mathfrak{E}_2, K, E_i, G_i, A_i, L_i, n_d, n_s, r$
  $\delta \leftarrow 2\pi/n_d$; $\partial\mathcal{W}_{_E} \leftarrow \varnothing$; ▷ initialization
  $Y \leftarrow [\mathcal{T}, \mathfrak{E}_1, \mathfrak{E}_2, K, E_i, G_i, A_i, L_i]$; $F \leftarrow F^{(0)}$
  for $j \leftarrow 1$ to n_d **do**
    $v^{(j)} \leftarrow E_\delta^{(j)}\, v_0$ ▷ direction vector
    $\tau_b^{(j)} \leftarrow solve_{(3.8)}(v^{(j)}, Y)$ ▷ optimal solution
    $[u_b^{(j)}, q_b^{(j)}] \leftarrow solve_{(2.48)}(\tau_b^{(j)}, Y, F)$ ▷ boundary point
    $F \leftarrow solve_{(2.13\mathrm{f})}(q_b^{(j)}, \bar{\mathcal{F}}_{e_i})$
    $\partial\mathcal{W}_{_E} \leftarrow \partial\mathcal{W}_{_E} \oplus u_b^{(j)}$ ▷ append
    if $\|u_b^{(j-1)} - u_b^{(j)}\|_2 > r$ **then** ▷ non-convexity
      $v_l \leftarrow dis(u_b^{(j)}, u_b^{(j-1)}, n_s)$ ▷ discretize
      for $s \leftarrow 1$ to n_s **do**
        $\tau_b^{(s)} \leftarrow solve_{(3.6)}(v_l^{(s)}, Y)$ ▷ optimal solution
        $[u_b^{(s)}, q_b^{(s)}] \leftarrow solve_{(2.48)}(\tau_b^{(s)}, Y, F)$
        $F \leftarrow solve_{(2.13\mathrm{f})}(q_b^{(s)}, \bar{\mathcal{F}}_{e_i})$
        $\partial\mathcal{W}_{_E} \leftarrow \partial\mathcal{W}_{_E} \oplus u_b^{(s)}$ ▷ append
      end for
    end if
  end for

In the following table (see Table 3.1), we give the numerical values of the Young's modulus E_i, shear modulus G_i, the Poisson's ratio ν, the inertial length L_i, the radius r_i, the second moments of area $J_{(x,y,z)_i}$ and the cross-section area A_i for each segment $i \in \{1, 2, 3\}$.

	Soft robot segments		
Physical parameters	First	Second	Third
L_i (m)	0.3	0.2	0.1
r_i (m)	0.06	0.04	0.02
J_{x_i} (m^4)	2.03×10^{-5}	4.02×10^{-6}	2.51×10^{-7}
$J_{(y,z)_i}$ (m^4)	1.07×10^{-5}	2.01×10^{-6}	1.25×10^{-7}
A_i (m^2)	0.0113	0.0050	0.0013
E (Pa)	18×10^4		
ν	0.45		
G (Pa)	62.069×10^3		

Table 3.1: Numerical values of the soft robot physical parameters.

For the optimization-based approach we propose to discretize the angles with a discretization step size of 0.05 Radian (unit).

In the following scenarios and also for the scenarios of the following workspace estimation chapters (Chapter 4 and Chapter 5), the obtained results of the workspace estimation via the proposed approaches are superimposed with the forward method results, and the 3D results are presented via 2D-views (Oxz, Oyx, and Oyz). Also, since in this thesis we only focus on the

position aspect of the workspace (see Definition 1), the obtained simulation results represent only the feasible positions of the workspace but not the orientation aspect. Besides, please note that one operation is a complete iteration that contains all the necessary steps to achieve one single feasible point in the workspace. Also, the presented simulation scenarios were implemented using MATLAB (In the forward approach, we used the function "fsolve" to solve the forward statics, while the function "fmincon" was used to solve the optimization problems formulated in this thesis) on an Intel Xeon(R) with a 16-GB RAM and a 3.50 GHz processor.

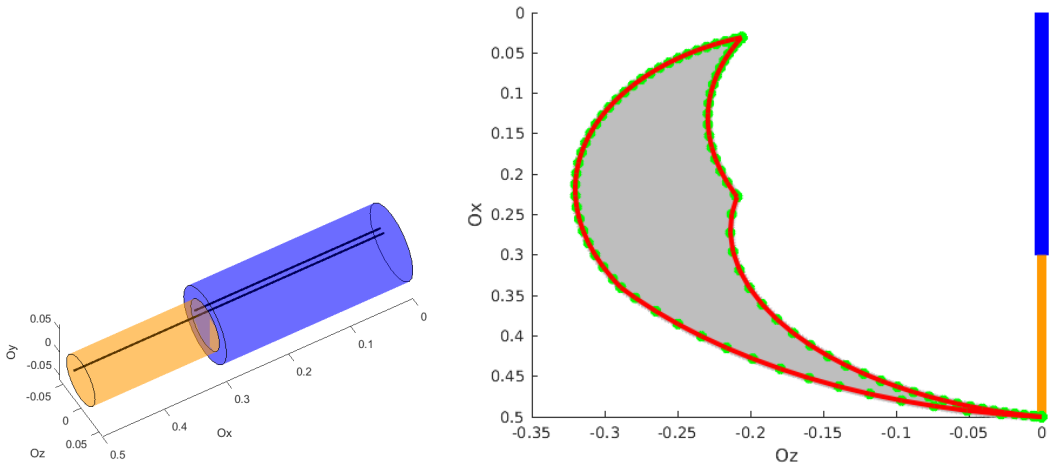
Finally, for the forward approach, we propose to discretize the actuators force vector with a discretization step size of 1 Newton (unit). In addition, since we know from simulation the cost of a single operation of the forward approach in order to find a feasible configuration in the workspace (0.00165 second for a single operation), then, we only gave an approximation to the computational time for the scenarios where the investigated soft robot is actuated by more than 4 actuators since the workspace estimation of these scenarios via the forward approach requires a long time.

3.2.5.1 Scenario 1: planar 2-segments 2-tendons soft robot

In the first scenario, we consider a soft robot composed of two segments and actuated by two tendons (See Fig. 3.6a). The two tendons are installed on the Ozx-plan, and parallel to each other. The first tendon is fixed at the position $(0, 0, -r_1/2)$ and extends along the first segment length L_1. The second tendon is fixed at the position $(0, 0, -r_2/2)$ and extends to $(L_1 + L_2, 0, -r_2/2)$. Such a chosen structure allows us to obtain a 2D workspace. We choose a tension magnitude within $\mathcal{T}_{1,2} = [0, 200]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0, 6]N \cdot m$ and $\mathcal{C}_{\mathcal{T}_2} = [0, 4]N \cdot m$.

First, we use the forward approach to estimate the workspace $\mathcal{W}_{_E}$ of this soft robot. We choose $n_d = 200$, and we obtain the following estimation of $\mathcal{W}_{_E}$ (See Fig. 3.6b). The computation cost of the forward approach for the two segments - two tendons soft robot is equal to 200^2 operations, and the workspace estimation took 66.35 seconds to obtain the full estimation of its workspace $\mathcal{W}_{_E}$.

Next, we apply the optimization-based approach with $n_d = 120$, and based on (3.8) we calculate all boundary points on the surface enveloping the workspace (See Fig. 3.6b). Finally, as illustrated by Fig. 3.6b, we achieve $\partial\mathcal{W}_{_E}$ of this particular soft robot configuration. The computation cost of the optimization-based approach for the studied soft robot is equal to 168 operations, and it required nearly 1.82 seconds to fully estimate its $\partial\mathcal{W}_{_E}$.



(a) Soft robot structure.

(b) Forward and Optimization approach.

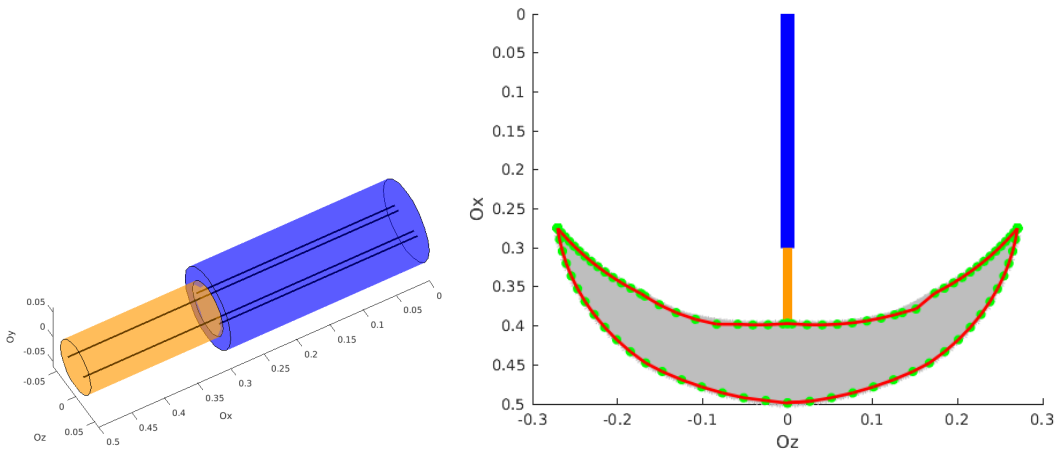
Figure 3.6: Scenario 1 - workspace (gray) $\mathcal{W}_{E}$, boundary points (green), and workspace boundary (red) $\partial\mathcal{W}_{E}$.

3.2.5.2 Scenario 2: planar 2-segments 4-tendons soft robot

In the second scenario, we consider the soft robot of the first scenario and we add two symmetric tendons (See Fig. 3.7a). We choose a tension magnitude within $\mathcal{T}_{1,\cdots,4} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,3}} = [0, 3]N \cdot m$ and $\mathcal{C}_{\mathcal{T}_{2,4}} = [0, 2]N \cdot m$.

First, we use the forward approach to estimate the workspace $\mathcal{W}_{E}$ of this soft robot. We choose $n_d = 100$, and we obtain the following estimation of $\mathcal{W}_{E}$ (See Fig. 3.7b). The computation cost of the forward approach for the two segments - four tendons soft robot is equal to 100^4 operations, and the workspace estimation will take 1.65×10^5 seconds to obtain the full estimation of its workspace $\mathcal{W}_{E}$.

Next, we apply the optimization-based approach with $n_d = 120$, and based on (3.8) we calculate all boundary points on the surface enveloping the workspace (See Fig. 3.7b). Finally, as illustrated by Fig. 3.7b, we achieve $\partial\mathcal{W}_{E}$ of this particular soft robot configuration. The computation cost of the optimization-based approach for the studied soft robot is equal to 168 operations, and it required nearly 3.92 seconds to fully estimate its $\partial\mathcal{W}_{E}$.



(a) Soft robot structure.

(b) Forward and Optimization approach.

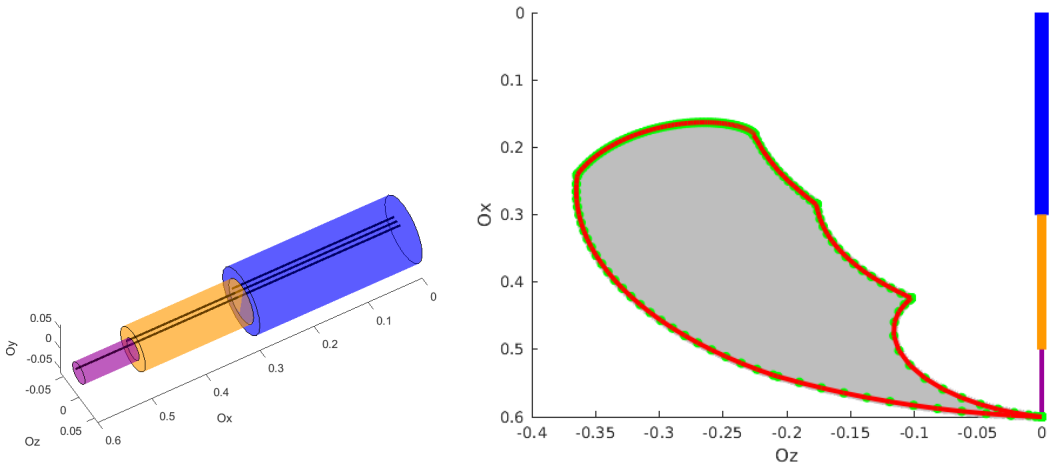
Figure 3.7: Scenario 2 - Workspace (gray) $\mathcal{W}_{E}$, boundary points (green), and workspace boundary (red) $\partial\mathcal{W}_{E}$.

3.2.5.3 Scenario 3: planar 3-segments 3-tendons soft robot

In the third scenario, we consider adding another segment and a tendon to the first scenario (See Fig. 3.8a). The additional tendon is mounted in a parallel manner to the tendons in the first scenario: it is fixed at the position $(0, 0, -r_3/2)$ and extends along the soft robot length $L_1 + L_2 + L_3$. Also, this configuration enables us to achieve a larger 2D workspace estimation compared to the first scenario. We choose a tension magnitude within $\mathcal{T}_{1,2,3} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0, 3]N \cdot m$, $\mathcal{C}_{\mathcal{T}_2} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_1} = [0, 1]N \cdot m$.

First, we use the forward approach to estimate the workspace $\mathcal{W}_{E}$ of this soft robot. We choose $n_d = 100$, and we obtain the following estimation of $\mathcal{W}_{E}$ (See Fig. 3.8b). The computation cost of the forward approach for the three segments - three tendons soft robot is equal to 100^3 operations, and the workspace estimation took 2962 seconds to obtain the full estimation of its workspace $\mathcal{W}_{E}$.

Next, we apply the optimization-based approach with $n_d = 120$, and based on (3.8) we calculate all boundary points on the surface enveloping the workspace (See Fig. 3.8b). Finally, as illustrated by Fig. 3.8b, we achieve $\partial\mathcal{W}_{E}$ of this particular soft robot configuration. The computation cost of the optimization-based approach for the studied soft robot is equal to 220 operations, and it required nearly 3.52 seconds to fully estimate its $\partial\mathcal{W}_{E}$.



(a) Soft robot structure.

(b) Forward and Optimization approach.

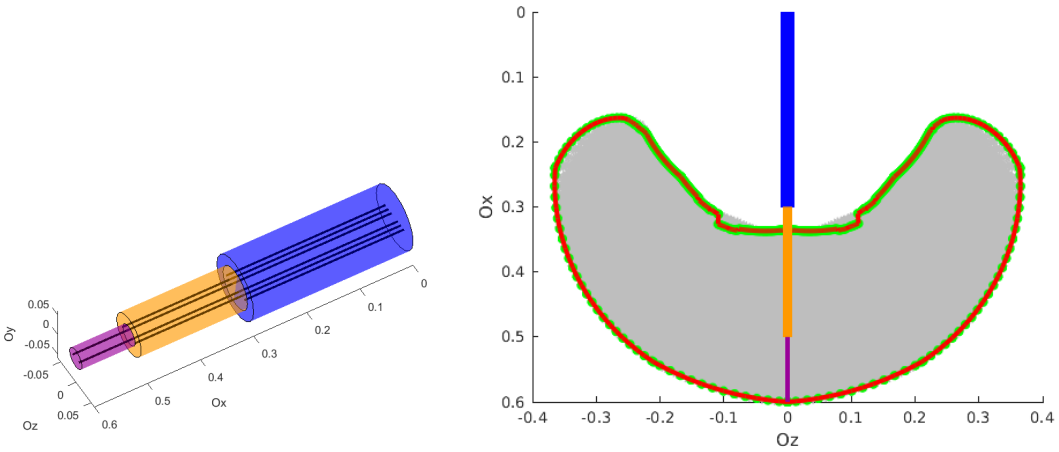
Figure 3.8: Scenario 3 - Workspace (gray) $\mathcal{W}_{E}$, boundary points (green), and workspace boundary (red) $\partial\mathcal{W}_{E}$.

3.2.5.4 Scenario 4: planar 3-segments 6-tendons soft robot

In the fourth scenario, three additional tendons are routed in a fashion where they are symmetric to the tendon configuration in the third scenario (See Fig. 3.9a). We choose a tension magnitude within $\mathcal{T}_{1,\cdots,6} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4}} = [0, 3]N{\cdot}m$, $\mathcal{C}_{\mathcal{T}_{2,5}} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6}} = [0, 1]N \cdot m$.

First, we use the forward approach to estimate the workspace $\mathcal{W}_{E}$ of this soft robot. We choose $n_d = 100$, and we obtain the following estimation of $\mathcal{W}_{E}$ (See Fig. 3.9b). The computation cost of the forward approach for the three segments - six tendons soft robot is equal to 100^6 operations, and the workspace estimation will take 1.65×10^9 seconds to obtain the full estimation of its workspace $\mathcal{W}_{E}$.

Next, we apply the optimization-based approach with $n_d = 120$, and based on (3.8) we calculate all boundary points on the surface enveloping the workspace (See Fig. 3.9b). Finally, as illustrated by Fig. 3.9b, we achieve $\partial\mathcal{W}_{E}$ of this particular soft robot configuration. The computation cost of the optimization-based approach for the studied soft robot is equal to 220 operations, and it required nearly 13.58 seconds to fully estimate its $\partial\mathcal{W}_{E}$.



(a) Soft robot structure.

(b) Forward and Optimization approach.

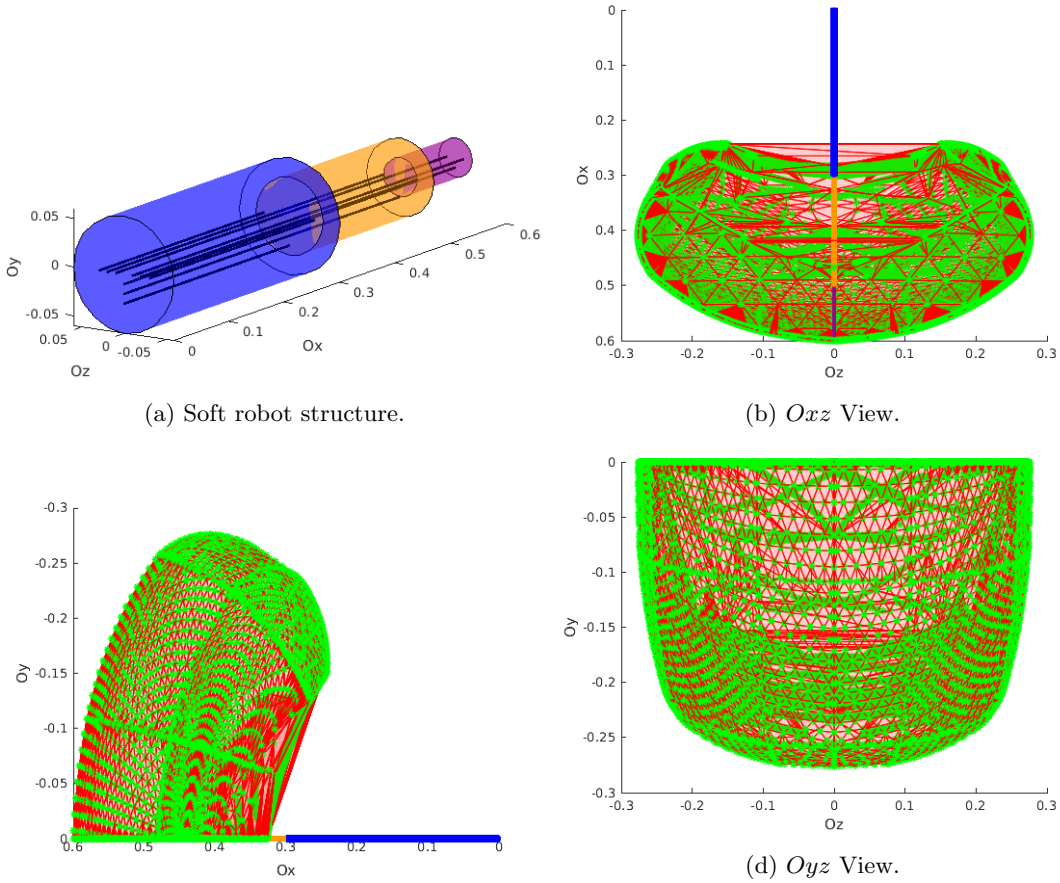
Figure 3.9: Scenario 4 - Workspace (gray) $\mathcal{W}_{E}$, boundary points (green), and workspace boundary (red) $\partial\mathcal{W}_{E}$.

3.2.5.5 Scenario 5: spatial 3-segments 9-tendons soft robot

In the final scenario, three additional tendons are added to the configurations of the above scenario in order to obtain a 3D workspace. The three additional tendons are mounted in the following way; the seventh, eighth and ninth tendons are routed in a manner where the seventh tendon is fixed at the position $(0, -r_1/2, 0)$ and extends along the first segment length L_1, the eighth tendon is fixed at the position $(0, -r_2/2, 0)$ and extends to $(L_1 + L_2, -r_2/2, 0)$, and the ninth tendon is fixed at the position $(0, -r_3/2, 0)$ and extends along the soft robot length $L_1 + L_2 + L_3$ (See Fig. 3.10a). With the tendons being mounted in this fashion, the actual configuration allows us to have a 3D workspace. We choose a tension magnitude within $\mathcal{T}_{1,\dots,9} = [0, 50]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4,7}} = [0, 1.5]N \cdot m$, $\mathcal{C}_{\mathcal{T}_{2,5,8}} = [0, 1]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6,9}} = [0, 0.5]N \cdot m$.

First, we use the forward approach to estimate the workspace $\mathcal{W}_{E}$ of this soft robot. We choose $n_d = 100$, and we obtain the following estimation of $\mathcal{W}_{E}$ (See Figs. 3.10b, 3.10c, 3.10d). The computation cost of the forward approach for the three segments - nine tendons soft robot is equal to 100^9 operations, and the workspace estimation will take 1.65×10^{12} seconds to obtain the full estimation of its workspace $\mathcal{W}_{E}$.

Next, we apply the optimization-based approach with $n_{d_y} = 120$ and $n_{d_z} = 72$, and based on (3.8) we calculate all boundary points on the surface enveloping the workspace (See Figs. 3.10b, 3.10c, 3.10d), and we finally achieve $\partial\mathcal{W}_{E}$ of this particular soft robot configuration. The computation cost of the optimization-based approach for the studied soft robot is equal to 8640 operations, and it required nearly 510 seconds to fully estimate its $\partial\mathcal{W}_{E}$.



(a) Soft robot structure.

(b) Oxz View.

(c) Oyx View.

(d) Oyz View.

Figure 3.10: Scenario 5 - Workspace (gray) $\mathcal{W}_{E}$, boundary points (green), and workspace boundary (red) $\partial\mathcal{W}_{E}$.

3.2.5.6 Notes on scenarios results

Based on the achieved results, we can clearly observe that the workspace boundary estimation result achieved via the optimization-based method encloses the estimated workspace obtained via the forward approach.

In addition, the proposed optimization-based approach can yield equivalent workspace estimation precision compared to the forward method, and simultaneously reduce both the number of operations and the computational time required for the workspace estimation for each scenario.

3.2.5.7 Notes on computational efficiency

For each particular configuration of a tendon-actuated soft robot, we showed the estimation of its workspace using both the forward and optimization-based approach. From the simulation results, we observe that the proposed optimization-based approach consisting of mapping the

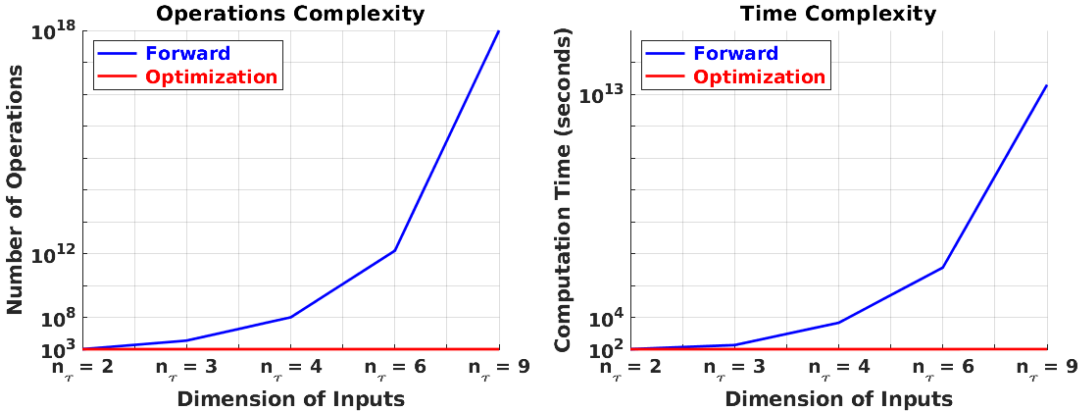
exterior boundaries of soft robots' workspaces has significantly reduced the computation cost of the workspace estimation for each soft robot configuration.

In the following table (see Table. 3.2), we summarize the operations complexity and the computational time for each scenario.

Table 3.2: Operations Complexity and computational time: forward vs optimization.

	Operations		Time (seconds)	
Approach / Inputs	Forward	Optimization	Forward	Optimization
$n_\tau = 2$	200^2	168	66.35	1.82
$n_\tau = 3$	100^3	220	2962	3.52
$n_\tau = 4$	100^4	168	$\approx 1.65 \times 10^5$	3.92
$n_\tau = 6$	100^6	220	$\approx 1.65 \times 10^9$	13.58
$n_\tau = 9$	100^9	8640	$\approx 1.65 \times 10^{12}$	510

Visually, the results of the above table are depicted in Figs. 3.11a and 3.11b, and we can clearly observe that, when the size of the actuators, (i.e., the number of tendons) increases, the computation complexity of the forward approach explodes exponentially, whereas the computation complexity of the optimization-based approach remains almost linearly stable.



(a) Operations computation complexity.

(b) Time computation complexity

Figure 3.11: Operations complexity and computational time of the forward and optimization approach for the workspace estimation of soft robots using the PCS model.

3.3 Optimization Approach for the FEM case

3.3.1 Implementation of the Methodology

The implementation of the optimization-based approach on the FEM model [9] follows the same steps described for the PCS model and detailed in Section 3.2.

In the first step, the definition of the workspace boundary for the FEM model is similar to that of the PCS method (see Definition 2 in Section 3.2.1).

In the second step, the goal is the same, which is to find a point on $\partial\mathcal{W}_{_E}$. However, the optimal problem to find a boundary point is formulated under different constraints for the FEM model, and established as follows:

$$\begin{aligned} [\tau^b, q^b] = \arg\min_{\tau,q} \quad & \|u(q) - v\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K(q)q = H(q)\tau + F(q) \\ & u = \mathfrak{C}q + u^{(0)} \end{aligned} \tag{3.9}$$

where the obtained actuators vector τ^b with its associated displacement vector q^b represents the optimal and feasible solution of (3.9).

Finally, we deduce the related end-effector boundary position using the following:

$$u^b = \mathfrak{C}q^b + u^{(0)}$$

Similar to the PCS case, the third step consists of mapping the workspace boundary $\partial\mathcal{W}_{_E}$. However, the optimal problem needs to be adapted for the FEM model, and formulated as follows:

$$\begin{aligned} [\tau^{b_j}, q^{b_j}] = \arg\min_{\tau,q} \quad & \|u(q) - v^{(j)}\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K(q)q = H(q)\tau + F(q) \\ & u = \mathfrak{C}q + u^{(0)} \end{aligned} \tag{3.10}$$

where the achieved actuators vector u^{b_j} with its associated displacement vector q^{b_j} is the optimal and feasible solution to minimize the distance between $v^{(j)}$ and u .

Accordingly, the associated end-effector boundary position can be deduced as:

$$u^{b_j} = \mathfrak{C}q^{b_j} + u^{(0)}$$

Finally, the solution to non-convexity problems that may arise using the FEM model can be solved following the same procedure described in Section 3.2.4 for the PCS case. However, we need to use the corresponding optimization equations (3.9) and the related FEM static model established in (2.53) during Step. B of the proposed strategy.

The algorithm of the optimization-based approach for the FEM case is presented in the following (See Algorithm 2):

3.3.2 Validation and Analysis

In this section, we illustrate the presented results by evaluating the reachable workspace of a trunk-like soft robot with a Young's modulus $E = 1.8 \times 10^7 Pa$, and a Poisson's ratio $\nu = 0.45$.

We consider three different scenarios where the trunk-like soft robot is actuated by different cable routing configurations, then we implement both the forward and optimization-based approach in order to estimate the reachable workspace $\mathcal{W}_{_E}$ and the workspace boundary $\partial\mathcal{W}_{_E}$, respectively, for each scenario.

For the optimization-based approach we propose to discretize the angles with a discretization step size of 0.08 Radian (unit).

Algorithm 2 Calculate $\partial\mathcal{W}_{_E}$

Require: $v_0, \mathfrak{C}, \mathcal{T}, n_d, n_s, r, u^{(0)}$
  $\delta \leftarrow 2\pi/n_d$; $\partial\mathcal{W}_{_E} \leftarrow \varnothing$; ▷ Initialization
  for $j \leftarrow 1$ to n_d **do**
    $v^{(j)} \leftarrow R_{\delta}^{(j)}\, v_0$ ▷ direction vector
    $[\tau^{b_j}, q^{b_j}] \leftarrow solve_{(3.10)}(v^{(j)}, C, \mathcal{T}, u^{(0)})$ ▷ optimal solution
    $u^{b_j} \leftarrow \mathfrak{C} q^{b_j} + u^{(0)}$ ▷ boundary point
    $\partial\mathcal{W}_{_E} \leftarrow \partial\mathcal{W}_{_E} \oplus u^{b_j}$ ▷ append
    if $\|u^{b_{j-1}} - u^{b_j}\|_2 > r$ **then** ▷ non-convexity
      $v_l \leftarrow dis(u^{b_{j-1}}, u^{b_j}, n_s)$ ▷ discretize
      for $s \leftarrow 1$ to n_s **do**
        $[\tau^{b_s}, q^{b_s}] \leftarrow solve_{(3.9)}(v_l^{(s)}, \mathfrak{C}, \mathcal{T}, u^{(0)})$
        $u^{b_s} \leftarrow \mathfrak{C} q^{b_s} + u^{(0)}$ ▷ boundary point
        $\partial\mathcal{W}_{_E} \leftarrow \partial\mathcal{W}_{_E} \oplus u^{b_s}$ ▷ append
      end for
    end if
  end for

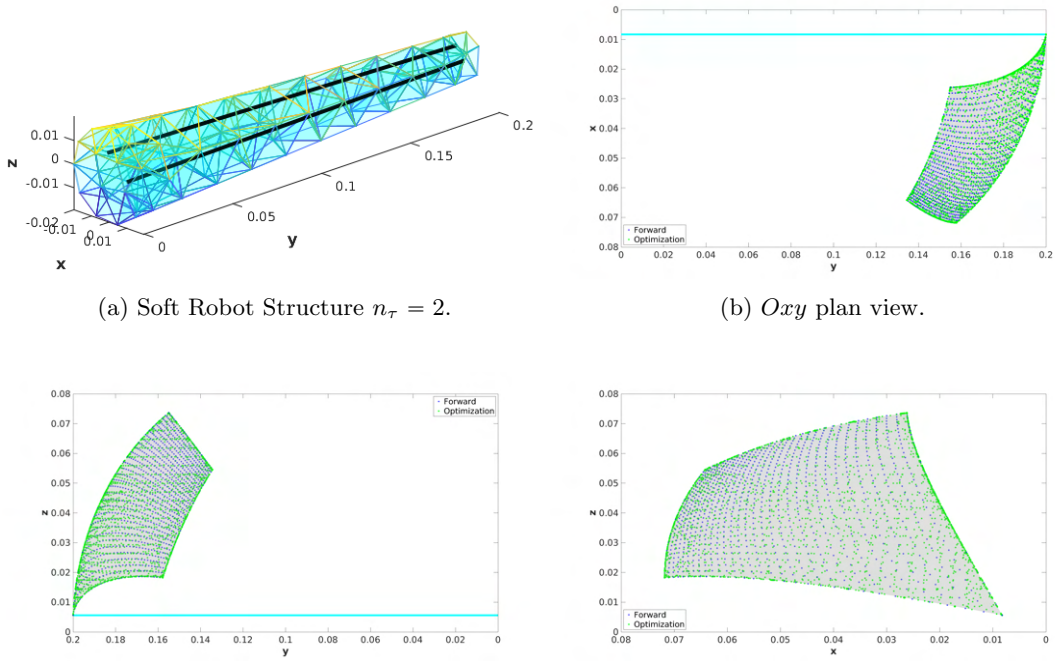
3.3.2.1 Scenario 1: trunk-like soft robot actuated by two tendons.

In the first scenario, we consider a cable routing configuration where the trunk-like soft robot is actuated by two tendons (as shown by Fig. 3.12a). We apply a force whose magnitude is within a specific range $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100] \times [0\ 100]$.

Following the procedure presented in Section 2.5.3, we firstly apply the forward approach with a discretization precision $n_d = 100$, and we obtain the corresponding estimation of $\mathcal{W}_{_E}$ (illustrated by blue colored points in Figs. 3.12b, 3.12c and 3.12d) for this soft robot's configuration. In term of operations' computation complexity, the forward approach requires $n_d^{n_\tau} = 100^2$ operations, and in term of time's computation, the workspace estimation took 1463 seconds to obtain the full estimation of this scenario's workspace.

Next, we use (3.10) (See Fig. 3.3) of the optimization-based approach with $n_{d_z} = 72$ and $n_{d_x} = 12$ to map the boundary points of this particular configuration.

For this scenario, the optimization-based approach required a total of 864 operations with a time computation of 984 seconds. The obtained boundary points are depicted by green colored points in Figs. 3.12b, 3.12c and 3.12d, which represent the workspace boundary $\partial\mathcal{W}_{_E}$ of this soft robot's configuration.



(a) Soft Robot Structure $n_\tau = 2$.

(b) Oxy plan view.

(c) Ozy plan view.

(d) Ozx plan view.

Figure 3.12: Scenario 1 - $\mathcal{W}_{_E}$ estimation via forward approach (blue points) and $\partial\mathcal{W}_{_E}$ estimated via optimization approach (green points).

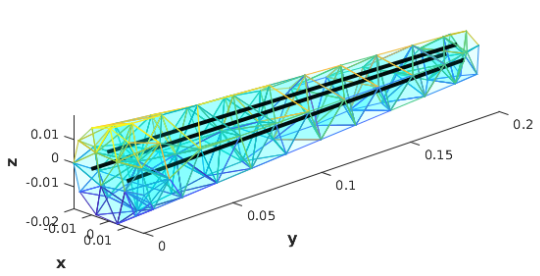
3.3.2.2 Scenario 2: trunk-like soft robot actuated by three tendons.

In this scenario, we consider a cable routing configuration where the trunk-like soft robot is actuated by three tendons (as shown by Fig. 3.13a). We apply a force whose magnitude is within a specific range $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100] \times [0\ 100] \times [0\ 100]$.

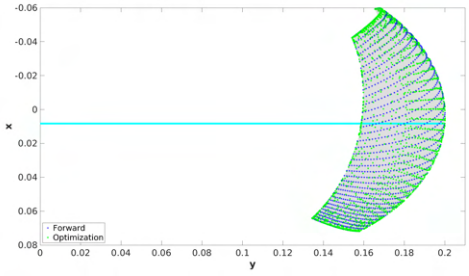
First, we apply the forward approach with a discretization number equal $n_d = 100$, and to achieve the workspace $\mathcal{W}_{_E}$ estimation (illustrated by blue colored points in Figs. 3.13b, 3.13c and 3.13d) of this particular configuration. In term of operations' computation complexity, the forward approach requires $n_d^{n_\tau} = 100^3$ operations, and in term of time's computation, the workspace estimation will take 1.4×10^5 seconds to obtain the full estimation of this scenario's workspace.

Next, using (3.10) of the optimization-based approach with $n_{d_z} = 72$ and $n_{d_x} = 12$, we map the boundary points of this particular configuration.

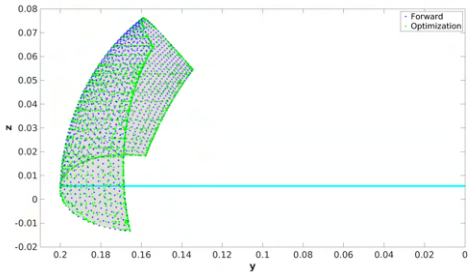
For this scenario, the proposed optimization-based approach required a total of 864 operations with a time computation of 1034 seconds. The achieved boundary points are illustrated by green colored points in Figs. 3.13b, 3.13c and 3.13d, which represent the workspace boundary $\partial\mathcal{W}_{_E}$ of this particular configuration.



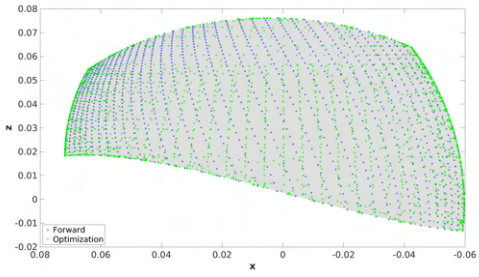
(a) Soft Robot Structure $n_\tau = 3$.



(b) Oxy plan view.



(c) Ozy plan view.



(d) Ozx plan view.

Figure 3.13: Scenario 2 - $\mathcal{W}_{_E}$ estimation via forward approach (blue points) and $\partial\mathcal{W}_{_E}$ estimated via optimization approach (green points).

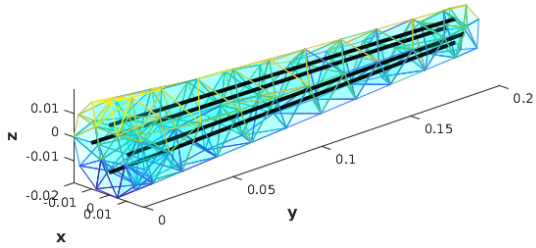
3.3.2.3 Scenario 3: trunk-like soft robot actuated by four tendons.

In the final scenario, we consider a cable routing configuration where the trunk-like soft robot is actuated by four symmetric tendons (as shown by Fig. 3.14a). We apply a force whose magnitude is within a specific range $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100]_1 \times \cdots \times [0\ 100]_4$.

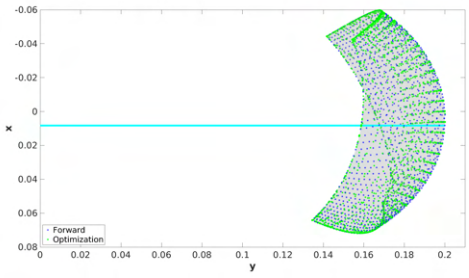
First, we apply the forward approach with a discretization precision equal $n_d = 100$, and we obtain the corresponding estimation of $\mathcal{W}_{_E}$ (illustrated by blue colored points in Figs. 3.14b, 3.14c and 3.14d) for this soft robot's configuration. In term of operations' computation complexity, the forward approach requires $n_d^{n_\tau} = 100^4$ operations, and in term of time's computation, the workspace estimation will take 1.4×10^7 seconds to achieve the full estimation of this scenario's workspace.

Next, we use (3.10) of the optimization-based approach with $n_{d_z} = 72$ and $n_{d_x} = 12$ to map the boundary points of this particular configuration.

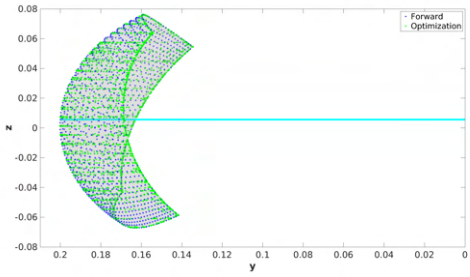
For this scenario, the optimization-based approach required a total of 864 operations with a time computation of 1039 seconds. The obtained boundary points are depicted by green colored points in Figs. 3.14b, 3.14c and 3.14d, which represent the workspace boundary $\partial\mathcal{W}_{_E}$ of the studied soft robot's configuration.



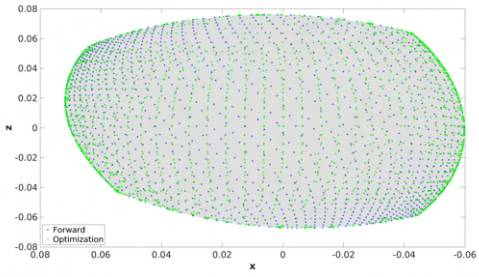
(a) Soft Robot Structure $n_\tau = 4$.



(b) Oxy plan view.



(c) Ozy plan view.



(d) Ozx plan view.

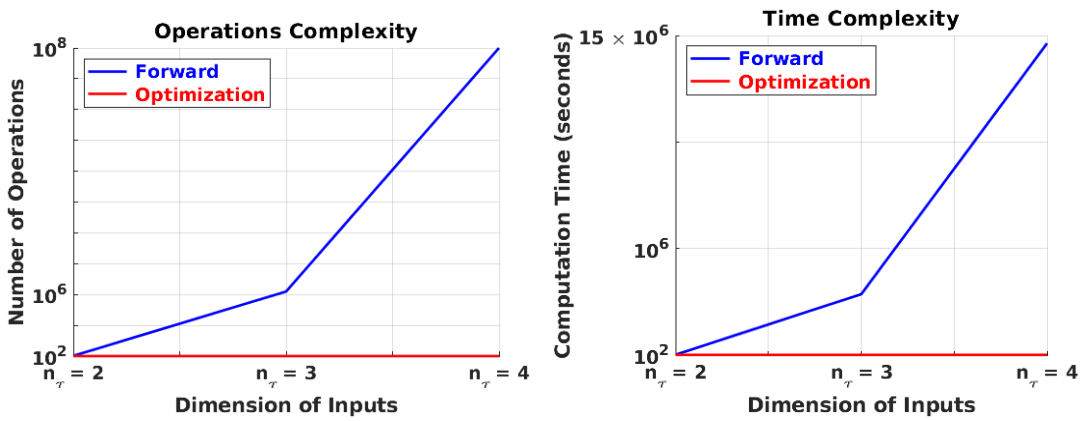
Figure 3.14: Scenario 3 - $\mathcal{W}_E$ estimation via forward approach (blue points) and $\partial\mathcal{W}_E$ estimated via optimization approach (green points).

3.3.2.4 Notes on computational efficiency

In the following table (see Table 3.3), and Figures. 3.15a and 3.15b, we summarize the operations complexity and time computation for each scenario.

Table 3.3: Operations and time computational complexity for the FEM case: forward vs optimization-based.

	Operations		Time (seconds)	
Approach / Inputs	Forward	Optimization	Forward	Optimization
$n_\tau = 2$	100^2	864	1463	984
$n_\tau = 3$	100^3	864	$\approx 1.4 \times 10^5$	1034
$n_\tau = 4$	100^4	864	$\approx 1.4 \times 10^7$	1039



(a) Operations computation complexity.

(b) Time computation complexity

Figure 3.15: Operations complexity and computational time of the forward and optimization approach for the workspace estimation of soft robots using the FEM model.

3.4 Conclusion

In this chapter, we have proposed an optimization-based approach that consists of mapping the exterior boundaries of the workspace of soft robots.

The proposed approach was successfully implemented to both the PCS and the FEM models, where we have shown its efficiency in reducing the computation complexity and computational time necessary to estimate the workspace of soft robots in contrast to the forward approach which explodes exponentially when increasing the dimension of the actuators.

On the other hand, since this approach consists only in mapping the exterior boundary of the workspace by determining the boundary points lying on the surface enclosing the workspace, it surely will not be able to identify inside information of the workspace such as interior boundaries.

Chapter 4

Interval Analysis Approach for Workspace Estimation

Due to the limitations encountered by the optimization approach that cannot provide knowledge on the interior configurations of the workspace, it is therefore necessary to provide a method that can overcome this limitation, but is also based on the second strategy, i.e., of discretizing the end-effector's space.

Such a method is called the forward-backward approach [7] that is based on interval analysis techniques [48, 61, 77–79], and it consists of discretizing the end-effector's space by starting from an initial reachable configuration then exploring the whole possible reachable space in order to finally estimate the workspace of a soft robot.

In the following, we will first provide a brief introduction about interval analysis techniques used by this approach, and then an introduction of the approach itself followed by its realization for both the PCS and FEM models.

4.1 Interval Analysis Introduction

The proposed forward-backward approach uses interval analysis techniques in order to estimate the workspace of soft robots. Thus, a brief introduction to some interval analysis techniques used in this thesis is presented in the following. More details can be found in [61].

Consider two real intervals $\mathcal{X}_1 = [\underline{\mathcal{X}}_1, \overline{\mathcal{X}}_1]$ and $\mathcal{X}_2 = [\underline{\mathcal{X}}_2, \overline{\mathcal{X}}_2]$, then according to [61], the product of $\mathcal{X}_1$ and $\mathcal{X}_2$ can be calculated by the following:

$$\mathcal{X}_1\mathcal{X}_2 = \big[min\big(\mathcal{C}(\mathcal{X}_1, \mathcal{X}_2)\big), max\big(\mathcal{C}(\mathcal{X}_1, \mathcal{X}_2)\big)\big]$$

with $\mathcal{C}(\mathcal{X}_1, \mathcal{X}_2) = \{\underline{\mathcal{X}}_1\underline{\mathcal{X}}_2, \underline{\mathcal{X}}_1\overline{\mathcal{X}}_2, \overline{\mathcal{X}}_1\underline{\mathcal{X}}_2, \overline{\mathcal{X}}_1\overline{\mathcal{X}}_2\}$.

The product of a real matrix $\mathcal{A} \in \mathbb{R}$ by a real interval $\mathcal{X} = [\underline{\mathcal{X}}, \overline{\mathcal{X}}]$ with appropriate dimension is calculated as follows:

$$\mathcal{A}\mathcal{X} = \begin{cases} [\mathcal{A}\underline{\mathcal{X}}, \mathcal{A}\overline{\mathcal{X}}], & \mathcal{A} > 0 \\ [\mathcal{A}\overline{\mathcal{X}}, \mathcal{A}\underline{\mathcal{X}}], & \mathcal{A} < 0 \end{cases}$$

However, due to the wrapping effect ($\mathcal{A}\mathcal{X} \supset \{\mathcal{A}x \mid x \in \mathcal{X}\}$) [61], this technique yields an over-estimation of the resulted space which can be explained by considering the following example:

$$\mathcal{A} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix},\ \mathcal{X} = \begin{pmatrix} [-1, 0] \\ [1, 2] \end{pmatrix}$$

then

$$\mathcal{AX} = \begin{pmatrix} [0,2] \\ [1,2] \end{pmatrix}$$

which implies that $(0,2)^T$ belongs to $\mathcal{AX}$, whereas it does not belong to the actual value set $\mathbb{B} = \{\mathcal{A}x \mid x \in \mathcal{X}\}$, as shown by Fig. 4.1.

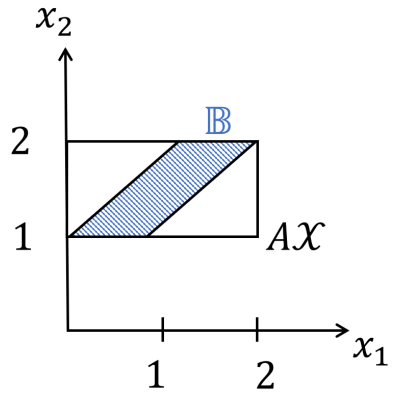


Figure 4.1: Over-estimation introduced by the wrapping effect.

Therefore, it is necessary to propose a technique to eliminate the resulted over-estimation in order to achieve the correct and accurate space, and this will be implemented in the proposed forward-backward approach and explained in the following.

4.2 Forward-Backward Approach Introduction

In this section, we present a brief overview of how the proposed forward-backward approach operates.

Firstly, we need to discretize the end-effector space. To realize this, we need to have an idea about its bounds. In this approach, we approximate the relation between q and τ at the $(j-1)$th iteration as follows:

$$K\delta q^{(j)} = H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)}), \text{ PCS}$$

and

$$K(q^{(j-1)})\delta q^{(j)} = H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)}), \text{ FEM}$$

with $\delta q^{(j)} = q - q^{(j-1)}$ and $\delta\tau^{(j-1)} = \tau - \tau^{(j-1)}$, where $\delta q^{(j)}$ and $\delta\tau^{(j-1)}$ represent respectively the generalized strain and the actuators vectors with respect to the prior configuration (for the initial state $j = 1$, $u(q^{(0)})$ and $q^{(0)}$ are the end-effector's initial position and the initial generalized strain vector respectively, and $\tau^{(0)}$ is the initial actuators force vector).

Based on the above result, a forward estimation of the end-effector workspace related to $\delta q^{(j)}$ can be obtained (see Step 1 in Fig. 4.2), noted as $\mathcal{W}_{E}^{(j)}$, by using interval analysis techniques [61] (which will be detailed in Section 4.3.1 for the PCS case and Section 4.4 for the FEM case). Once it is estimated, we can discretize $\mathcal{W}_{E}^{(j)} \subset \mathbb{R}^3$ with a prescribed precision and obtain, as shown by Step 2 in Figs. 4.2 and 4.3, the associated discretized space $\mathcal{W}_{E_d}^{(j)}$ (subscript "d" stands for discretization). Next, based on the discretized space $\mathcal{W}_{E_d}^{(j)}$, we calculate a feasible

small neighborhood, noted as $\mathcal{W}_{E_s}^{(j)}$ (subscript "s" stands for small neighborhood), around the current end-effector position $u^{(j-1)}$ (see Step 2 in Figs. 4.2 and 4.3). After that, we proceed backwardly to the actuators domain $\mathcal{T}$ in order to compute the corresponding actuators $\tau^{(j)}$ (see Step 3 in Figs. 4.2 and 4.4), with which the investigated soft robot may reach $\mathcal{W}_{E_s}^{(j)}$. This step, i.e. Step 3, is necessary to obtain the accurate corresponding end-effector position $u^{(j)}$ which is approximately equal to the associated point in the set $\mathcal{W}_{E_s}^{(j)}$. In addition, this step is also necessary for the computation of the new actuators bounds $\mathcal{T}^{(j)}$ in Step 4. After, by utilizing the actuators vector $\tau^{(j)}$ inferred from Step 3, we calculate the new actuators bound $\mathcal{T}^{(j)}$ (shown as Step 4 in Figs. 4.2 and 4.5), allowing us to forwardly construct the new end-effector bound $\mathcal{W}_{E}^{(j+1)}$ which will be required for the next iteration of the proposed approach.

However, since interval analysis techniques lead to an over-estimation of $\mathcal{W}_{E}^{(j)}$ (due to the wrapping effect [61]), this thesis proposes a simple strategy to accurately reduce the dimension of $\mathcal{W}_{E}^{(j)}$. The strategy proposed to reduce the overestimation lies within two steps of the forward-backward approach. In part, Step 2 indeed allows us to reduce the over-estimation by discarding certain unfeasible small neighborhoods because the feasible small neighborhoods $\mathcal{W}_{E_s}^{(j)}$ must belong to the discretized space $\mathcal{W}_{E_d}^{(j)}$ and the possible small neighborhoods (called the η-neighborhood in Step 2) around the current configuration $u^{(j-1)}$ (which is illustrated by Fig. 4.3, and will be explained in detail in Step 2, Section 4.3.2). On the other hand, the over-estimation correction occurs mainly at the level of Step 5 of the approach where we dismiss pre-explored feasible small neighborhoods, and therefore efficiently and precisely reduce the over-estimated workspace (which is illustrated by Fig. 4.6, and will be explained in detail in Step 5, Section 4.3.5).

Eventually, to achieve the whole estimation of $\mathcal{W}_{E}$ using the forward-backward method, we iterate the above procedure starting from the initial state $j = 1$ until the stop condition (which will be detailed in Step 5 and depicted by Fig. 4.7) is verified.

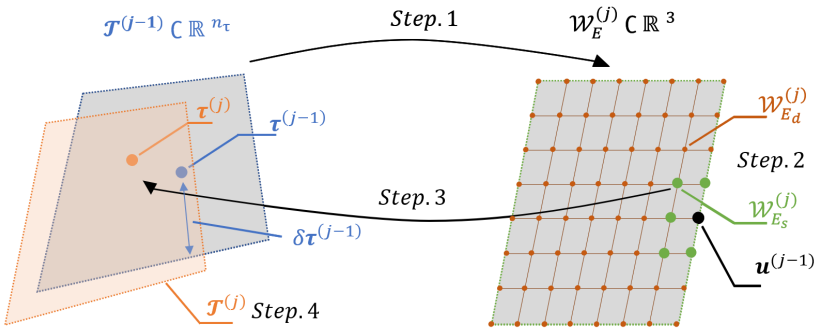


Figure 4.2: Forward-Backward approach.

Compared to the forward method, the forward-backward approach has the following two advantages:

1. It reduces the space's dimension that must be discretized (3 for the position case);
2. It eliminates wrong sub-spaces by the interval estimation technique, and discards pre-explored configurations, yielding a significant decrease of the necessary iteration number to explore the whole workspace, which results in reducing the required computational time to achieve $\mathcal{W}_{E}$.

Compared to the optimization-based approach, the forward-backward approach is exhaustive since it seeks to explore the whole workspace providing its interior and exterior points.

4.3 Forward-Backward Approach for the PCS case

In this section, we present the detailed realization of the forward-backward approach applied to the PCS model following the portrayed scheme in Fig. 4.2.

4.3.1 Step 1: Forward estimation of $\mathcal{W}_E^{(j)}$

In the first step, we use interval analysis techniques to forwardly estimate the workspace of the end-effector.

Precisely, suppose that at the $(j-1)$th iteration we have the information of the end-effector position $u^{(j-1)}$, the generalized strain vector $q^{(j-1)}$ and the corresponding actuators vector $\tau^{(j-1)} \in \mathcal{T}^{(j-1)}$ (for the case $j=1$, it corresponds to the initial configuration of the studied soft robot). Then, it is now proposed that (2.48) can be approximated by the following:

$$K\delta q^{(j)} = H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)}) \tag{4.1a}$$

$$u^{(j)} = \mathfrak{E}_1\, g(\delta q^{(j)})\, \mathfrak{E}_2^T \tag{4.1b}$$

with $\delta q^{(j)} = q - q^{(j-1)}$ and $\delta\tau^{(j-1)} = \tau - \tau^{(j-1)}$.

Let us note the matrix A as:

$$A(q^{(j-1)}) = K^{-1}H(q^{(j-1)})$$

Define now

$$A^+ = \max\{0, A\}$$

where the $\max\{\cdot\}$ operation is applied element-wise [135], and

$$A^- = A - A^+$$

then using the following theorem, we achieve an over-estimation of the end-effector workspace $\mathcal{W}_E^{(j)}$ corresponding to the given actuators bound $\mathcal{T}^{(j-1)}$.

Theorem 2. *For a given controllable configuration of the soft robot, with the actuators bound $\mathcal{T}^{(j-1)}$ defined as*

$$\mathcal{T}^{(j-1)} = [\underline{\delta\tau}^{(j-1)}, \overline{\delta\tau}^{(j-1)}]$$

where $\underline{\delta\tau}^{(j-1)}$ and $\overline{\delta\tau}^{(j-1)}$ represent the lower and upper bounds of the actuators vector with respect to the prior configuration respectively.
An over-estimated end-effector workspace $\mathcal{W}_E^{(j)}$ is given by

$$\mathcal{W}_E^{(j)} = \left\{ u^{(j)} \in \mathbb{R}^{n_u} | u^{(j)} \in \left[\underline{u}^{(j)}, \overline{u}^{(j)}\right] \right\} \tag{4.2}$$

with

$$\underline{u}^{(j)} = \mathfrak{E}_1 \underline{g}(\underline{\delta q}^{(j)})\mathfrak{E}_2^T, \text{ and } \overline{u}^{(j)} = \mathfrak{E}_1 \overline{g}(\overline{\delta q}^{(j)})\mathfrak{E}_2^T$$

the configuration matrix bounds $g(\delta q^{(j)}) \in [\underline{g}(\underline{\delta q}^{(j)}), \overline{g}(\overline{\delta q}^{(j)})]$, and the generalized strain vector bounds $\delta q^{(j)} \in [\underline{\delta q}^{(j)}, \overline{\delta q}^{(j)}]$ are determined by:

$$\underline{\delta q}^{(j)} = A^+\underline{\delta\tau}^{(j-1)} + A^-\overline{\delta\tau}^{(j-1)} + K^{-1}F(q^{(j-1)})$$

and

$$\overline{\delta q}^{(j)} = A^{+}\overline{\delta\tau}^{(j-1)} + A^{-}\underline{\delta\tau}^{(j-1)} + K^{-1}F(q^{(j-1)})$$

Proof. From (4.1a) we can get the generalized strain vector as follows:

$$\begin{aligned}\delta q^{(j)} &= K^{-1}H(q^{(j-1)})\delta\tau^{(j-1)} + K^{-1}F(q^{(j-1)}) \\ &= A(q^{(j-1)})\delta\tau^{(j-1)} + K^{-1}F(q^{(j-1)})\end{aligned}$$

Using Lemma 3 in [135], we get the following:

$$A^{+}\underline{\delta\tau}^{(j-1)} + A^{-}\overline{\delta\tau}^{(j-1)} \leqslant A\delta\tau^{(j-1)} \leqslant A^{+}\overline{\delta\tau}^{(j-1)} + A^{-}\underline{\delta\tau}^{(j-1)}$$

and deduce the generalized strain vector bounds as:

$$\underline{\delta q}^{(j)} \leqslant \delta q^{(j)} \leqslant \overline{\delta q}^{(j)}$$

with:

$$\underline{\delta q}^{(j)} = A^{+}\underline{\delta\tau}^{(j-1)} + A^{-}\overline{\delta\tau}^{(j-1)} + K^{-1}F(q^{(j-1)})$$

and

$$\overline{\delta q}^{(j)} = A^{+}\overline{\delta\tau}^{(j-1)} + A^{-}\underline{\delta\tau}^{(j-1)} + K^{-1}F(q^{(j-1)})$$

Next, applying the same lemma to (2.49b), and since $B_{q_i} \geqslant 0$, we obtain the strain twist bounds as:

$$\underline{\hat{\xi}}_i^{(j)} \leqslant \hat{\xi}_i^{(j)} \leqslant \overline{\hat{\xi}}_i^{(j)}$$

After, by implementing the above result in (2.49a), and according to [61], we get the configuration matrix bounds:

$$\underline{g}(\underline{\delta q}^{(j)}) \leqslant g(\delta q^{(j)}) \leqslant \overline{g}(\overline{\delta q}^{(j)})$$

Finally, we deduce the bounds of the end-effector position:

$$\underline{u}^{(j)} \leqslant u^{(j)} \leqslant \overline{u}^{(j)}$$

with

$$\underline{u}^{(j)} = \mathfrak{E}_1\underline{g}(\underline{\delta q}^{(j)})\mathfrak{E}_2^T \text{ and } \overline{u}^{(j)} = \mathfrak{E}_1\overline{g}(\overline{\delta q}^{(j)})\mathfrak{E}_2^T$$

which defines the over-estimated workspace of the end-effector $\mathcal{W}_{E}^{(j)}$ in (4.2). □

4.3.2 Step 2: Computing the Feasible Small Neighborhood $\mathcal{W}_{E_s}^{(j)}$

With $\mathcal{W}_{E}^{(j)}$ being estimated in Step 1, the second step consists of discretizing $\mathcal{W}_{E}^{(j)}$ with a prescribed precision, and then selecting a feasible small neighborhood $\mathcal{W}_{E_s}^{(j)}$ around the current end-effector position $u^{(j-1)}$.

The proposed strategy is to discretize $\mathcal{W}_{E}^{(j)}$ using a globally uniform spatial grid with a global small pre-defined precision s_{E_d} in each direction x, y and z, i.e.,

$$\mathcal{W}_{E_d}^{(j)} = \left\{ u^{(j)} \in \mathcal{W}_{E}^{(j)} | u^{(j)} = u^{(j-1)} + s_{E_d}\Gamma, \text{ for } \Gamma \in \mathbb{Z}^3 \right\} \tag{4.3}$$

Next, an η-neighborhood of the configuration $u^{(j-1)}$ can be defined as:

$$\mathcal{S}_\eta(u^{(j-1)}) = \left\{ \tilde{u} \in \mathbb{R}^3 | \|u^{(j)} - u^{(j-1)}\|_\infty = \eta \right\}$$

with $\eta = s_{E_d}$. Consequently, with the discretized space $\mathcal{W}_{E_d}^{(j)}$ being established, we can then define the feasible small neighborhood of the end-effector position $u^{(j-1)}$ as follows:

$$\mathcal{W}_{E_s}^{(j)} = \mathcal{W}_{E_d}^{(j)} \cap \mathcal{S}_\eta(u^{(j-1)}) \tag{4.4}$$

The relationship between the estimated workspace $\mathcal{W}_E^{(j)}$, its discretization space $\mathcal{W}_{E_d}^{(j)}$ and the relative feasible small neighborhood space $\mathcal{W}_{E_s}^{(j)}$ are illustrated by Fig. 4.3.

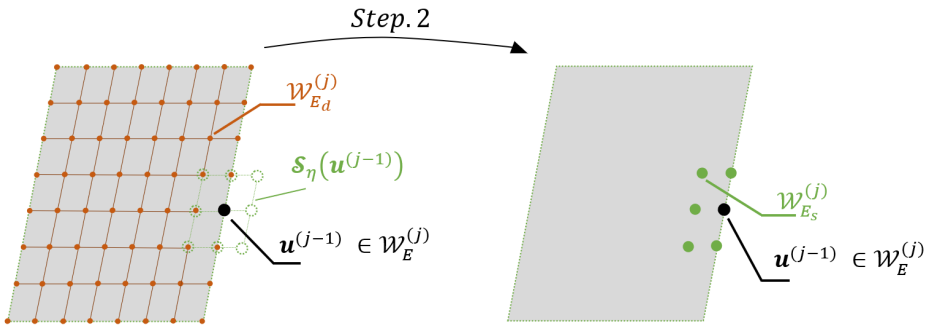


Figure 4.3: Feasible small neighborhood $\mathcal{W}_{E_s}^{(j)}$.

4.3.3 Step 3: Backward Estimation of the Corresponding Configuration

The third step's purpose is to choose the discretized points in the feasible small neighborhood space $\mathcal{W}_{E_s}^{(j)}$, except the point $u^{(j-1)}$, and then backwardly calculate the feasible actuators vector $\tau^{(j)}$ which can drive the soft robot's end-effector to reach those points. The corresponding scheme is depicted by Fig. 4.4.

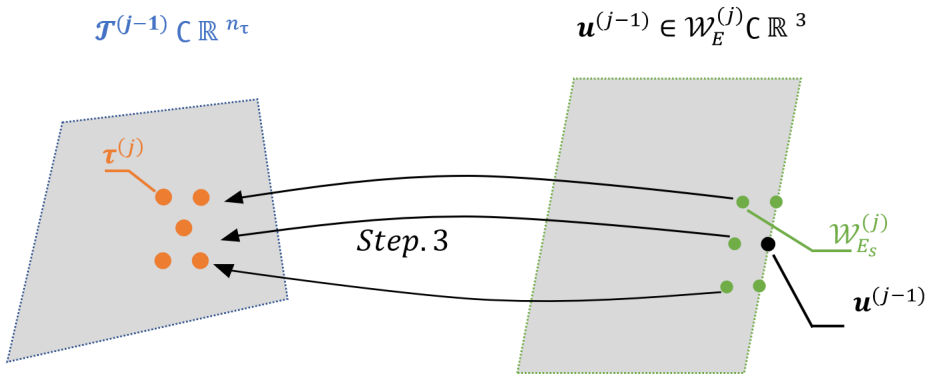


Figure 4.4: Backward estimation of $\tau^{(j)}$.

Precisely, since any point $\hat{u}^{(j)} \in \mathcal{W}_{E_s}^{(j)} \backslash u^{(j-1)}$ is located inside the feasible small region of $u^{(j-1)}$, then we can use this value to compute the associated configuration by solving the

following nonlinear constrained optimization problem:

$$\begin{aligned} \tau^{(j)} = \arg\min_{\tau} \quad & \|u(q) - \hat{u}^{(j)}\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K\ q = H(q)\tau + F\,(q) \\ & u(q) = \mathfrak{E}_1\ g(\delta q^{(j)})\ \mathfrak{E}_2^T \end{aligned} \tag{4.5}$$

where $\tau^{(j)}$ is the optimal and feasible solution (since it satisfies the bounded and nonlinear constraints) for the purpose of minimizing the nonlinear cost function $\|u(q) - \hat{u}^{(j)}\|_2^2$, which represents the distance between $\hat{u}^{(j)}$ and $u(q)$.

With the calculated $\tau^{(j)}$, and in accordance with Proposition 1, we can calculate the corresponding generalized strain vector $q^{(j)}$ through (2.51). Then, we compute its related strain twist $\hat{\xi}_i^{(j)}$ using (2.49b). Next, we derive the associated configuration matrix $g(q^{(j)})$ via (2.49a). Finally, using (2.48b), we may deduce the corresponding position $u^{(j)}$.

Regarding the implementation of the methodology, the gradient of $f(q)$ with respect to τ, denoted by $\nabla_\tau\big(f(q)\big)$ can be analytically formulated as:

$$\nabla_\tau\big(f(q)\big) = 2\nabla_\tau\big[u(q)\big]\ \big(u(q) - \hat{u}^{(j)}\big)$$

with $\nabla_\tau\big[u(q)\big] = \nabla_\tau(q)J^T(q)[0_3\ R(q)]^T$ being already established in Theorem 1, and $\nabla_\tau(q) = H^T\,(q)\left[\mathfrak{B}^{-1}(\tau, q)\right]^T$ being defined in Assumption 1, with $\mathfrak{B}(\tau, q, L, \alpha) = K(L, \alpha) - H_q(q, L, \alpha)(I_n \otimes \tau) - F_q(q, L, \alpha)$.

4.3.4 Step 4: Calculation of New Actuators Bound

In this step, we use the value of $\tau^{(j)}$ to compute the associated new bound, noted as $\mathcal{T}^{(j)}$, from $\mathcal{T}^{(j-1)}$.

For this, denote $\tau_k^{(j)}$ for $1 \leqslant k \leqslant n_\tau$ as the kth value of $\tau^{(j)}$, and $\mathrm{diag}(\tau^{(j)})$ as the related $n_\tau \times n_\tau$ diagonal matrix whose entries are the n_τ elements of the vector $\tau^{(j)}$, i.e.,

$$\mathrm{diag}(\tau^{(j)}) = \mathrm{diag}\left\{\tau_1^{(j)}, \cdots, \tau_{n_\tau}^{(j)}\right\}$$

Then, the new bound $\mathcal{T}^{(j)}$, illustrated by Fig. 4.5, can be determined using the value of $\tau^{(j)}$ from Step 3 and the value of $\mathcal{T}^{(j-1)}$ as follows:

$$\mathcal{T}^{(j)} = \mathcal{T}^{(j-1)} - \mathrm{diag}(\tau^{(j)}) \times \mathbf{1}_{n_\tau \times 2} \tag{4.6}$$

where $\mathbf{1}_{n_\tau \times 2}$ represents the $n_\tau \times 2$ matrix with all entries equal to 1.

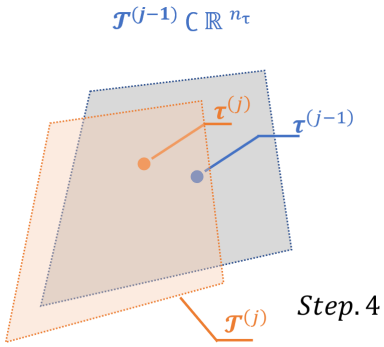


Figure 4.5: New actuators bound $\mathcal{T}^{(j)}$.

4.3.5 Step 5: Iteration and Stop Condition

For a particular configuration $u^{(j-1)}$ with its corresponding over-estimated workspace $\mathcal{W}_E^{(j)}$ calculated via Step 1, its related discretized space $\mathcal{W}_{E_d}^{(j)}$ computed via Step 2, and its associated feasible small neighborhood $\mathcal{W}_{E_s}^{(j)}$ deduced from (4.4). An important property of Step 2 allows us to not take into account certain non-feasible neighborhoods around $u^{(j-1)}$, which results in diminishing the number of feasible small neighborhoods $\mathcal{W}_{E_s}^{(j)}$ to be explored for the next iterations of the forward-backward approach, as illustrated in Fig. 4.3, by verifying the condition (4.4).

At the same time, given the information of $\mathcal{W}_{E_s}^{(j-1)}$, and the newly calculated feasible small neighborhood $\mathcal{W}_{E_s}^{(j)}$ from (4.4), an additional characteristics allowing us to further shrink the space to be explored is that some of the points belonging to $\mathcal{W}_{E_s}^{(j)}$ have already been explored by the former iterations, noted as $\mathcal{W}_{E_\cup}^{(j-1)}$, and defined by recurrence as follows:

$$\mathcal{W}_{E_\cup}^{(j-1)} = \cup_{k=1}^{j-1} \mathcal{W}_{E_s}^{(k)}$$

The above elimination procedure is executed in Step 5, as depicted by Fig. 4.6, by verifying the following condition:

$$\mathcal{W}_{E_s}^{(j)} \leftarrow \mathcal{S}_\eta(u^{(j-1)}) \backslash \left(\mathcal{W}_{E_\cup}^{(j-1)} \cap \mathcal{S}_\eta(u^{(j-1)}) \right) \tag{4.7}$$

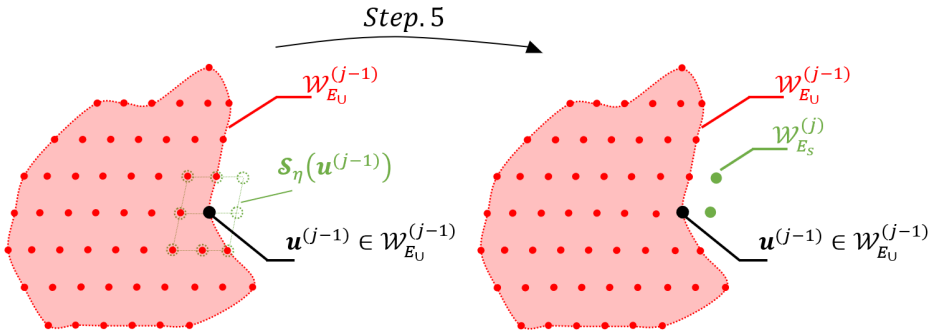


Figure 4.6: Discard pre-explored feasible neighborhoods.

Combining these two conditions (Eq. (4.4) in Step 2 and Eq. (4.7) in Step 5) enables us to shrink the space to be explored efficiently and accurately, resulting in a significant decrease of the computation complexity of the proposed forward-backward approach (as it will be shown in the simulation results in Section 4.3.6).

Consequently, the stop condition of the proposed forward-backward method can be formulated as follows:

$$\exists j \in \mathbb{N}, \text{ such that } \tilde{u} \in \mathcal{W}_{E_\cup}^{(j-1)}, \quad \forall \tilde{u} \in \mathcal{W}_{E_s}^{(j)}$$

or equivalently

$$\exists j \in \mathbb{N}, \text{ such that } \mathcal{W}_{E_\cup}^{(j)} = \mathcal{W}_{E_\cup}^{(j-1)}$$

implying that when backwardly estimating the related actuators vector $\tau^{(j-1)}$ of each point $\tilde{u} \in \mathcal{W}_{E_s}^{(j)}$, and then, if the associated end-effector position $u^{(j)}$ belongs to the union of all feasible small neighborhood set of the last iterations $\mathcal{W}_{E_\cup}^{(j-1)}$ (as shown in Fig. 4.7), we can then conclude that these configurations have already been explored.

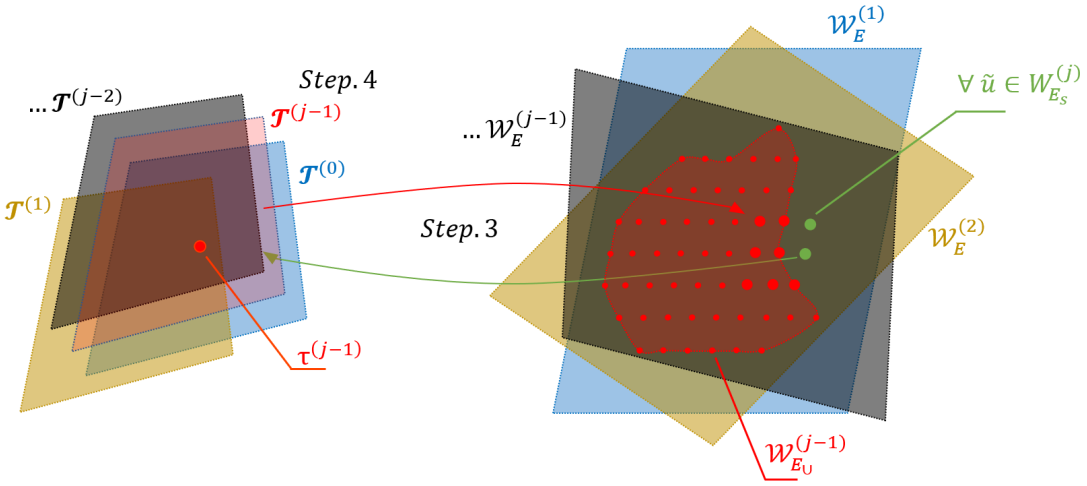


Figure 4.7: Stop condition.

Finally, we deduce that all feasible small neighborhoods were explored and no more reduction of the over-estimated workspace is possible. At this level, the estimated workspace is:

$$\mathcal{W}_E = \mathcal{W}_{E_\cup}^{(j-1)}.$$

In the following, a brief algorithm is presented in the table below to describe the main steps of the forward-backward approach.

Algorithm 3 Calculate $\{u, q, \tau, \mathcal{T}, \mathcal{W}_{E_s}, \mathcal{W}_{E_\cup}\}^{(j)}, \mathcal{W}_E$

Require: $\Gamma, s_{E_d}, \{u, q, \tau, \mathcal{T}, \mathcal{W}_{E_s}, \mathcal{W}_{E_\cup}\}^{(j-1)}$
  while $\mathcal{W}_{E_s}^{(j)} \notin \mathcal{W}_{E_\cup}^{(j-1)}$ **do** ▷ Stop Condition.
    $\mathcal{W}_E^{(j)} \leftarrow f_1(\{u, q, \mathcal{T}\}^{(j-1)})$ ▷ Step 1: Forward estimation of $\mathcal{W}_E^{(j)}$, eq. (4.2)
    $\mathcal{W}_{E_d}^{(j)} \leftarrow f_2(u^{(j-1)}, \Gamma, s_{E_d})$ ▷ Step 2: Discretized space $\mathcal{W}_{E_d}^{(j)}$, eq. (4.3)
    $\mathcal{W}_{E_s}^{(j)} \leftarrow \mathcal{W}_{E_d}^{(j)} \cap \mathcal{S}_\eta(u^{(j-1)})$ ▷ Step 2: Feasible neighborhood $\mathcal{W}_{E_s}^{(j)}$, eq. (4.4)
    $\hat{u}^{(j)} \leftarrow \mathcal{W}_{E_s}^{(j)} \backslash u^{(j-1)}$ ▷ Exclusion.
    $\{u, q, \tau\}^{(j)} \leftarrow f_3(\hat{u}^{(j)}, q^{(j-1)})$ ▷ Step 3: Backward estimation of $u^{(j)}$, eq. (4.5)
    $\mathcal{T}^{(j)} \leftarrow f_4(\tau^{(j)}, \mathcal{T}^{(j-1)})$ ▷ Step 4: New actuators bound $\mathcal{T}^{(j)}$, eq. (4.6)
    $\mathcal{W}_{E_s}^{(j)} \leftarrow \mathcal{S}_\eta(u^{(j-1)}) \backslash (\mathcal{W}_{E_\cup}^{(j-1)} \cap \mathcal{S}_\eta(u^{(j-1)}))$ ▷ Step 5: Dismiss explored area, eq. (4.7)
    $\mathcal{W}_{E_\cup}^{(j)} \leftarrow \mathcal{W}_{E_\cup}^{(j-1)} \cup \mathcal{W}_{E_s}^{(j)}$ ▷ Append.
    $\{u, q, \tau, \mathcal{T}, \mathcal{W}_{E_\cup}\}^{(j-1)} \leftarrow \{u, q, \tau, \mathcal{T}, \mathcal{W}_{E_\cup}\}^{(j)}$ ▷ Update Values.
    $j \leftarrow j + 1$ ▷ Next iteration.
  end while
  $\mathcal{W}_E \leftarrow \mathcal{W}_{E_\cup}^{(j-1)}$ ▷ The Estimated Reachable Workspace.

4.3.6 Validation and Analysis

In this section, the same soft robots' configurations presented in the scenarios of Section 3.2.5 in Chapter 3 are considered, where we implement both the forward and forward-backward approaches to deduce the workspace estimation $\mathcal{W}_E$. Table 3.1 presents the numerical values of the investigated soft robot.

In the following scenarios, the workspace obtained from the forward approach is depicted by black-colored points. We suggest discretizing the end-effector space with a discretization step size of $s_{E_d} = 0.02$ Meter (unit).

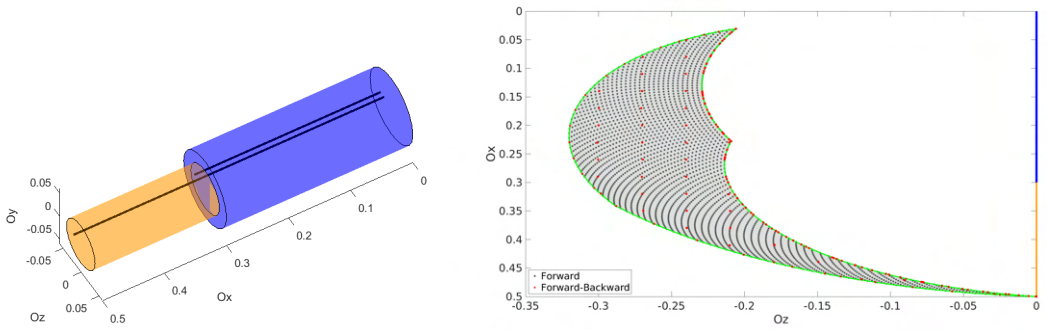
4.3.6.1 Scenario 1: planar 2-segments 2-tendons soft robot

In the first scenario, we consider a soft robot composed of two segments and actuated by two tendons (See Fig. 4.8a). The two tendons are installed as the first scenario of Section 3.2.5.1 in Chapter 3. We choose a tension magnitude within $\mathcal{T}_{1,2} = [0, 200]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0, 6]N \cdot m$ and $\mathcal{C}_{\mathcal{T}_2} = [0, 4]N \cdot m$.

Next, we apply the proposed forward-backward interval analysis approach by following the procedure presented in Section 4.3. Using $\mathcal{T}^{(0)}$ we can forwardly estimate $\mathcal{W}_E^{(1)}$ as stated in Step. 1 of Fig. 4.2, based on which a feasible small neighborhood $\mathcal{W}_{E_s}^{(1)}$ is determined as mentioned in Step. 2 of Fig. 4.2. Next, as indicated in Step. 3 of Fig. 4.2, we backwardly compute the feasible actuators vector $\tau^{(1)}$ of each configuration in the feasible small neighborhood $\mathcal{W}_{E_s}^{(1)}$. Finally, we find the new actuators bound $\mathcal{T}^{(1)}$ as demonstrated in Step. 4 of Fig. 4.2. The same process is pursued in the next iterations until we meet the stop condition as stated in Step. 5 and shown by Fig. 4.7.

For this scenario, the forward-backward approach took 36 complete iterations and found a total of 1820 feasible points in the workspace with a time computation of 17.5 seconds. Those obtained feasible points are illustrated by red-colored points in Fig. 4.8b, and the union of those feasible points represents the workspace $\mathcal{W}_E$ estimation, which is depicted by the gray zone with green contour. Moreover, for a complete 36 iterations it is expected to have $\mathcal{W}_{E_d} = 6^{36} \approx 10^{28}$

points to explore. However, what we actually explored was a total of $\mathcal{W}_{E_d} = 6732$ points, and a final total of $\mathcal{W}_{E_s} = 1820$ feasible points.



(a) Soft robot structure.

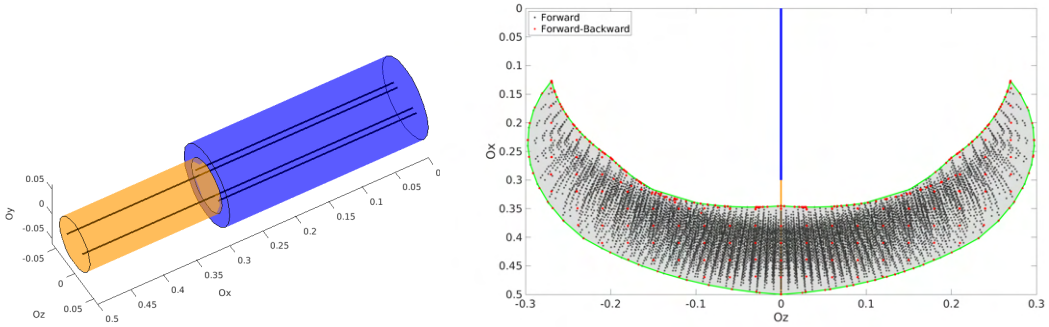
(b) Oxz plan view of $\mathcal{W}_E$.

Figure 4.8: Scenario 1 - $\mathcal{W}_E$ estimation via forward approach (black points) and forward-backward approach (gray area).

4.3.6.2 Scenario 2: planar 2-segments 4-tendons soft robot

In the second scenario, we consider the soft robot of the first scenario and we add two symmetric tendons (See Fig. 4.9a). We choose a tension magnitude within $\mathcal{T}_{1,2} = [0, 150]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\tau_{1,3}} = [0, 4.5]N \cdot m$ and $\mathcal{C}_{\tau_{2,4}} = [0, 3]N \cdot m$.

Next, we apply the proposed forward-backward interval analysis approach by following the procedure presented in Section 4.3. The forward-backward approach in this scenario took 47 complete iterations and computed a total of 3640 feasible points in 187.63 seconds. Those obtained feasible points are illustrated by red-colored points in Fig. 4.9b, and their union represents the workspace $\mathcal{W}_E$ estimation, which is illustrated by the gray zone with green contour. In addition, we actually explored a total of $\mathcal{W}_{E_d} = 13464$ points, and a final total of $\mathcal{W}_{E_s} = 3640$ feasible points, instead of an expected $\mathcal{W}_{E_d} = 6^{47} \approx 3 \times 10^{36}$ points to explore.



(a) Soft robot structure.

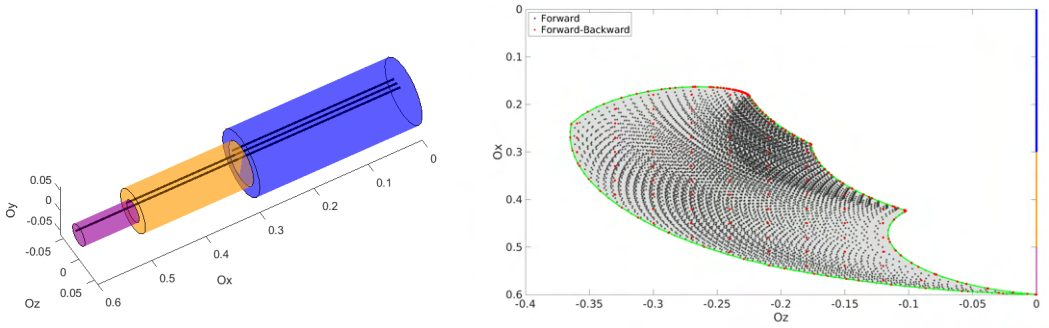
(b) Oxz plan view of $\mathcal{W}_E$.

Figure 4.9: Scenario 2 - $\mathcal{W}_E$ estimation via forward approach (black points) and forward-backward approach (gray area).

4.3.6.3 Scenario 3: planar 3-segments 3-tendons soft robot

In the third scenario (See Fig. 4.10a), we consider the same tendons routing of the third scenario in Section 3.2.5.3 from Chapter 3. We choose a tension magnitude within $\mathcal{T}_{1,2,3} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0, 3]N \cdot m$, $\mathcal{C}_{\mathcal{T}_2} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_1} = [0, 1]N \cdot m$.

Next, following the procedure presented in Section 4.3, we apply the proposed interval analysis approach. For this scenario, the forward-backward approach took 32 complete iterations and computed a total of 2275 feasible points in a duration of 72.84 seconds. Those obtained feasible points are depicted by red-colored points in Fig. 4.10b, and their union is illustrated by the gray zone with green contour, which gives the workspace $\mathcal{W}_E$ estimation. Besides, we actually explored a total of $\mathcal{W}_{E_d} = 8415$ points, and a final total of $\mathcal{W}_{E_s} = 2275$ feasible points, instead of an expected $\mathcal{W}_{E_d} = 6^{32} \approx 7 \times 10^{24}$ points to explore.



(a) Soft robot structure.

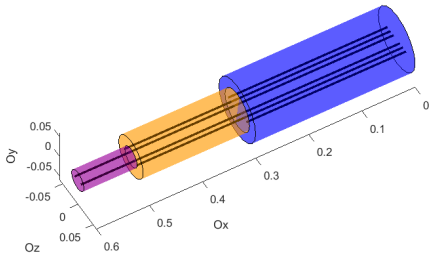
(b) Oxz plan view of $\mathcal{W}_E$.

Figure 4.10: Scenario 3 - $\mathcal{W}_E$ estimation via forward approach (black points) and forward-backward approach (gray area).

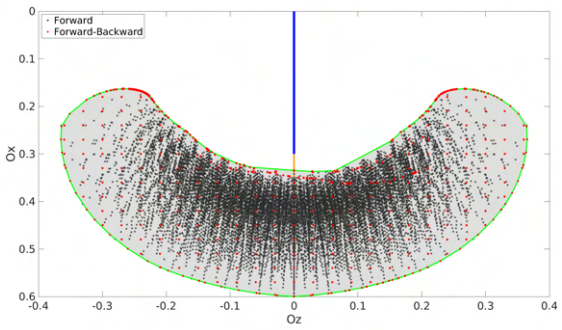
4.3.6.4 Scenario 4: planar 3-segments 6-tendons soft robot

In this scenario, three additional tendons are routed in a fashion where they are symmetric to the tendons configuration in the third scenario (See Fig. 3.9a). We choose a tension magnitude within $\mathcal{T}_{1,\cdots,6} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4}} = [0, 3]N \cdot m$, $\mathcal{C}_{\mathcal{T}_{2,5}} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6}} = [0, 1]N \cdot m$.

Next, the proposed interval analysis approach is applied by following the procedure presented in Section 4.3. The forward-backward approach of this scenario took 55 complete iterations and computed a total of 4550 feasible points in the workspace within 2436 seconds. The red-colored points in Fig. 4.11b depict those obtained feasible points, where their union represents the workspace $\mathcal{W}_E$ estimation, which is illustrated by the gray zone with green contour. Furthermore, for a complete 55 iterations it is expected to have $\mathcal{W}_{E_d} = 6^{55} \approx 6 \times 10^{42}$ points to explore. However, we actually explored a total of $\mathcal{W}_{E_d} = 16830$ points, and a final total of $\mathcal{W}_{E_s} = 4550$ feasible points.



(a) Soft robot structure.



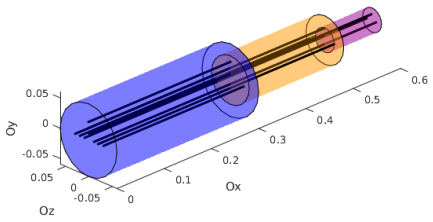
(b) Oxz plan view of $\mathcal{W}_E$.

Figure 4.11: Scenario 4 - $\mathcal{W}_E$ estimation via forward approach (black points) and forward-backward approach (gray area).

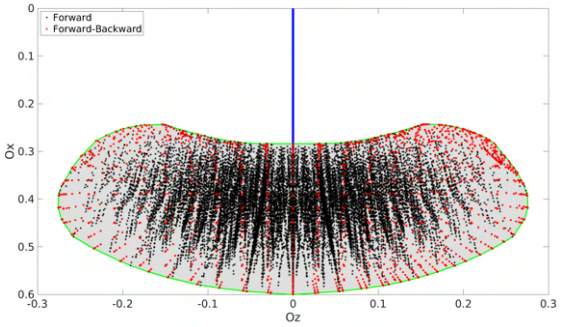
4.3.6.5 Scenario 5: spatial 3-segments 9-tendons soft robot

In the final scenario (See Fig. 4.12a), we consider the same tendons routing of the last scenario in Section 3.2.5.5 from Chapter 3, but in this scenario, the three additional tendons are mounted upward (i.e., on the positive level of the Oy axis). We choose a tension magnitude within $\mathcal{T}_{1,\cdots,9} = [0, 50]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4,7}} = [0, 1.5]N \cdot m$, $\mathcal{C}_{\mathcal{T}_{2,5,8}} = [0, 1]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6,9}} = [0, 0.5]N \cdot m$.

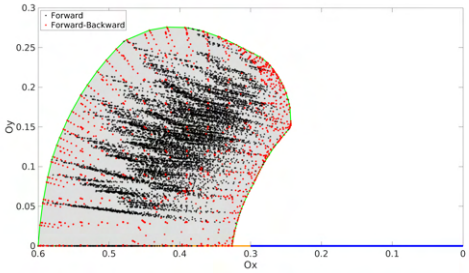
Next, we apply the proposed forward-backward interval analysis approach by following the procedure presented in Section 4.3. This approach took 35 complete iterations and computed a total of 13350 feasible points with a computational time of 4679 seconds. Those obtained feasible points are depicted by red-colored points in Figs.4.12b, 4.12c and 4.12d, and their union of represents the workspace $\mathcal{W}_E$ estimation, which is depicted by the gray area with green contour. Furthermore, we actually explored is a total of $\mathcal{W}_{E_d} = 50490$ points, and a final total of $\mathcal{W}_{E_s} = 13350$ feasible points, instead of the expected $\mathcal{W}_{E_d} = 6^{35} \approx 10^{27}$ points to explore.



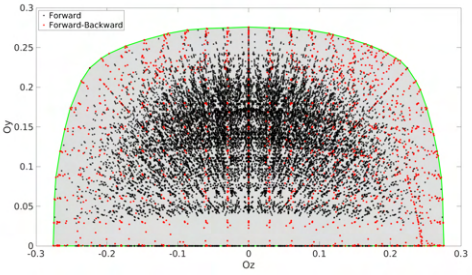
(a) Soft robot structure.



(b) Oxz plan view of $\mathcal{W}_E$.



(c) Oyx plan view of $\mathcal{W}_E$.



(d) Oyz plan view of $\mathcal{W}_E$.

Figure 4.12: Scenario 5 - $\mathcal{W}_E$ estimation via forward approach (black points) and forward-backward approach (gray area).

4.3.6.6 Notes on scenarios results

The proposed forward-backward approach yields comparable workspace estimation precision compared to the forward method, and at the same time reduces the complexity required to estimate the workspace of soft robots.

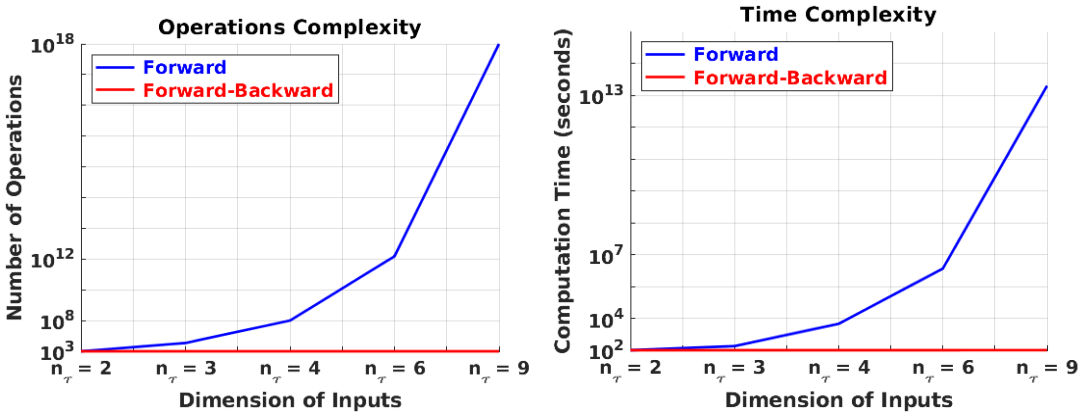
In fact, this significant reduction introduced by the proposed forward-backward approach was partially due to the elimination process of inappropriate configurations from Step. 2 (See Fig. 4.3), but mainly due to the dismissal of all feasible configurations that have already been explored during Step. 5 (See Fig. 4.6).

4.3.6.7 Notes on computational efficiency

In Table 4.1, and Figures. 4.13a and 4.13b, we summarize the operations' complexity and computational time of each scenario.

Table 4.1: Operations Complexity and computational time: forward vs forward-backward.

Approach / Inputs	Operations		Time (seconds)	
	Forward	Forward-Backward	Forward	Forward-Backward
$n_\tau = 2$	200^2	1820	66.35	17
$n_\tau = 3$	100^3	2275	2962	72
$n_\tau = 4$	100^4	3640	$\approx 1.65 \times 10^5$	187
$n_\tau = 6$	100^6	4550	$\approx 1.65 \times 10^9$	2436
$n_\tau = 9$	100^9	13350	$\approx 1.65 \times 10^{12}$	4679



(a) Operations.

(b) Time

Figure 4.13: Operations complexity and computational time of the forward and forward-backward approaches for the workspace estimation of the investigated scenarios using the PCS model.

4.4 Forward-Backward Approach for the FEM case

4.4.1 Implementation of the Methodology

The implementation of the forward-backward approach for the FEM model follows the same steps described for the PCS model (which was detailed in Section 4.3, and illustrated by Fig. 4.2).

However, Step 1 and Step 3 of the forward-backward needs to be adapted for the FEM case.

For the first step of the this approach, we firstly need to linearly approximate the workspace defined by (2.53) with the knowledge of the position vector of the end-effector $u^{(j-1)}$, the displacement vector $q^{(j-1)}$, and the corresponding actuators vector $\tau^{(j-1)} \in \mathcal{T}^{(j-1)}$, as follows:

$$\begin{aligned} K(q^{(j-1)})\delta q^{(j)} &= H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)}) \\ u^{(j)} &= \mathfrak{C}\delta q^{(j)} + u^{(j-1)} \end{aligned} \tag{4.8}$$

with $\delta q^{(j)} = q - q^{(j-1)}$ and $\delta u^{(j-1)} = u - u^{(j-1)}$.

Next, suppose that the actuators are well installed in a manner that the investigated soft robot

is always controllable [134], then for any given value of $q^{(j-1)}$ the matrix $K(q^{(j-1)})$ is always invertible. Let us introduce the matrix A as:

$$A(q^{(j-1)}) = \mathfrak{C}K^{-1}(q^{(j-1)})H(q^{(j-1)})$$

and define the matrix A^+ as before by

$$A^+ = \max\{0, A\}$$

and $A^- = A - A^+$

With the above necessary elements, for the related displacement $\delta q^{(j)}$, the over-estimated workspace $\mathcal{W}_E^{(j)}$ corresponding to the actuators bound $\mathcal{T}^{(j-1)}$ can be calculated via the following theorem:

Theorem 3. *Given a soft robot's controllable configuration, with the actuators bound $\mathcal{T}^{(j-1)}$ defined as*

$$\mathcal{T}^{(j-1)} = [\underline{\delta\tau}^{(j-1)}, \overline{\delta\tau}^{(j-1)}]$$

with $\underline{\delta\tau}^{(j-1)}$ and $\overline{\delta\tau}^{(j-1)}$ represents the lower and upper bounds of the actuators vector with respect to the prior configuration respectively. Then the over-estimated end-effector workspace $\mathcal{W}_E^{(j)}$ is defined as

$$\mathcal{W}_E^{(j)} = \left\{ u^{(j)} \in \mathbb{R}^{n_u} | u^{(j)} \in \left[\underline{u}^{(j)}, \overline{u}^{(j)} \right] \right\} \tag{4.9}$$

with

$$\underline{u}^{(j)} = A^+\underline{\delta\tau}^{(j-1)} + A^-\overline{\delta\tau}^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)}$$

and

$$\overline{u}^{(j)} = A^+\overline{\delta\tau}^{(j-1)} + A^-\underline{\delta\tau}^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)}.$$

Proof. Using the linear approximation defined in (4.8), the nodal displacement vector can be directly deduced as:

$$\delta q^{(j)} = K^{-1}(q^{(j-1)})\big[H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)})\big]$$

With the defined matrix A, we can then use the above equation to write the end-effector displacement in function of the matrix A, which yields:

$$\begin{aligned} u^{(j)} &= \mathfrak{C}\delta q^{(j)} + u^{(j-1)} \\ &= \mathfrak{C}K^{-1}(q^{(j-1)})\big[H(q^{(j-1)})\delta\tau^{(j-1)} + F(q^{(j-1)})\big] + u^{(j-1)} \\ &= \mathfrak{C}K^{-1}(q^{(j-1)})H(q^{(j-1)})\delta\tau^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)} \\ &= A(q^{(j-1)})\delta\tau^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)} \end{aligned}$$

Next, according to Lemma 3 in [135], the following over-estimation is obtained:

$$A^+\underline{\delta\tau}^{(j-1)} + A^-\overline{\delta\tau}^{(j-1)} \leqslant A\delta\tau \leqslant A^+\overline{\delta\tau}^{(j-1)} + A^-\underline{\delta\tau}^{(j-1)}$$

By simply defining

$$\underline{u}^{(j)} = A^+\underline{\delta\tau}^{(j-1)} + A^-\overline{\delta\tau}^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)}$$

and

$$\overline{u}^{(j)} = A^+\overline{\delta\tau}^{(j-1)} + A^-\underline{\delta\tau}^{(j-1)} + \mathfrak{C}K^{-1}(q^{(j-1)})F(q^{(j-1)}) + u^{(j-1)}$$

that leads to:

$$\underline{u}^{(j)} \leqslant u^{(j)} \leqslant \overline{u}^{(j)}$$

Finally, we obtain the over-estimated workspace $\mathcal{W}_E^{(j)}$ defined by (4.9). □

The third step of the forward-backward approach needs also to be adapted in order to treat the FEM case. Therefore, as explained in Step 3 of Section 4.3.3 for the PCS model, we follow the same procedure for the FEM model, and we determine the associated displacement vector of the end-effector as:

$$\hat{q}_{E}^{(j)} = \hat{u}^{(j)} - u^{(j-1)}$$

Consequently, we can calculate the equivalent actuators and displacement vector by solving the optimization problem formulated below:

$$\begin{aligned} [\tau^{(j)}, q^{(j)}] = \arg\min_{\tau,q} \quad & \|q_{E} - \hat{q}_{E}^{(j)}\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K(q)q = H(q)\tau + F(q) \\ & q_{E} = \mathfrak{C}q \end{aligned} \tag{4.10}$$

Solving the backward mechanism (4.10) yields the feasible actuators vector $\tau^{(j)}$ with its related feasible displacement vector $q^{(j)}$, which will eventually drive the soft robot's end-effector to reach $\hat{u}^{(j)}$.

Finally, we deduce the end-effector position associated with the optimal displacement vector $q^{(j)}$ as

$$u^{(j)} = \mathfrak{C}q^{(j)} + u^{(j-1)}.$$

The algorithm of the forward-backward approach for the FEM case consists of the same main steps that was described for the PCS model in Table 3. However, the functions of Steps 1 and 3 need to be adapted for the FEM case as explained above.

4.4.2 Validation and Analysis

This section presents the comparative results between the forward and forward-backward approaches to the same soft robots' configurations presented in the scenarios of Section 3.3.2 in Chapter 3.

In the following scenarios, the workspace obtained from the forward approach is depicted by blue-colored points. We propose to discretize the end-effector space with a discretization step size of $s_{E_d} = 0.005$ Meter (unit).

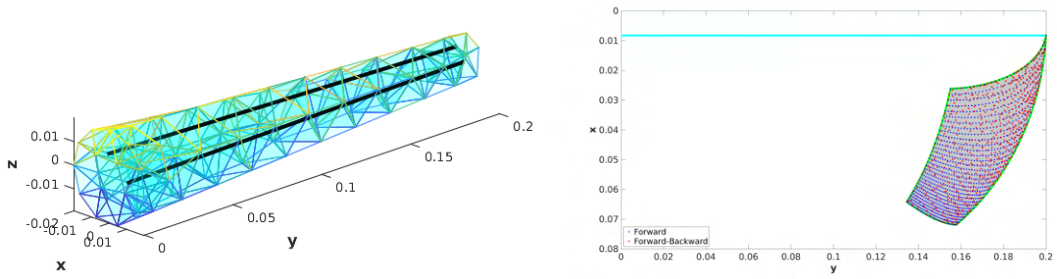
4.4.2.1 Scenario 1: trunk-like soft robot actuated by two cables.

In the first scenario (as shown by Fig. 4.14a), we consider the same configuration of the first scenario in Section 3.3.2.1 from Chapter 3. The applied force is of the following range: $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100] \times [0\ 100]$.

For this scenario, the forward-backward approach took 37 complete iterations and found a total of 635 feasible points in the workspace with a time computation complexity of 757 seconds. Those obtained feasible points are depicted by red colored points in Figs. 4.14b, 4.14c and 4.14d), and the union of all small neighborhood of those feasible points gives the estimation of workspace $\mathcal{W}_{E}$, which is represented by the gray zone with green contour. It is clear to see from the three-view drawing that the workspace estimation result obtained via forward-backward method coincides with that obtained via forward approach.

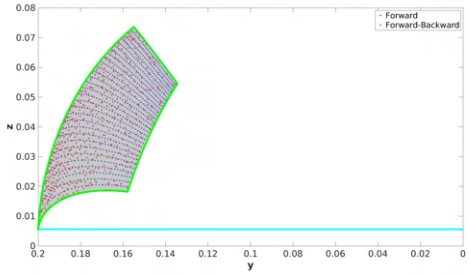
Moreover, for a complete 37 iterations it is expected to have $\mathcal{W}_{E_d} = 6^{37} = 6.18 \times 10^{28}$ points to explore. However, what we actually explored is a total of $\mathcal{W}_{E_d} = 1680$ points, and a final total of $\mathcal{W}_{E_s} = 635$ feasible points. This significant reduction was partially due to the elimination

process of inappropriate configurations from Step. 2 (See Fig. 4.3), but mainly due to the elimination of all feasible configurations that were already explored during Step. 5 (See Fig. 4.6).

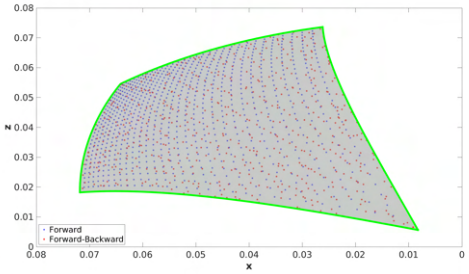


(a) Soft Robot Structure $n_\tau = 2$.

(b) Oxy plan view.



(c) Ozy plan view.



(d) Ozx plan view.

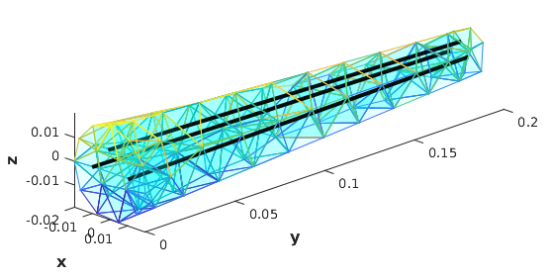
Figure 4.14: Scenario 1 - $\mathcal{W}_{E}$ estimation via forward approach (blue points) and forward-backward (gray area) approach.

4.4.2.2 Scenario 2: trunk-like soft robot actuated by three cables.

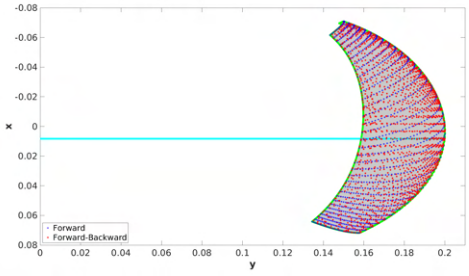
In this scenario (as shown by Fig. 4.15a), we consider the same configuration of the second scenario in Section 3.3.2.2 from Chapter 3. The applied force is of the following range: $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100] \times [0\ 100] \times [0\ 150]$.

Next, we apply the forward-backward interval analysis approach using the equations presented in Section 4.4. For this scenario, the forward-backward approach took 44 complete iterations and calculated a total of 1941 feasible points in the workspace with a time computation complexity of 2063 seconds. Those achieved feasible points are illustrated with red colored points in Figs. 4.15b, 4.15c and 4.15d), and the union of all small neighborhood of those feasible points yield the estimated workspace $\mathcal{W}_{E}$, which is represented by the gray zone within the green contour. Using the three-view illustration, we can clearly observe that the estimated workspace calculated via the proposed forward-backward approach conforms with that obtained through the forward approach.

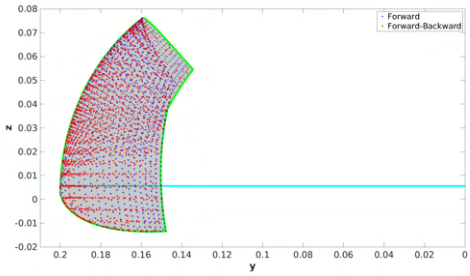
Furthermore, for a complete 44 iterations it is expected to get $\mathcal{W}_{E_d} = 6^{44} = 1.17 \times 10^{34}$ points to investigate. Yet, the total of points that we actually explored is $\mathcal{W}_{E_d} = 5466$, with a final total of $\mathcal{W}_{E_s} = 1941$ feasible points. This substantial reduction was slightly due to Step. 2 (refer to Fig. 4.3) elimination process of irrelevant configurations, but considerably due to the exclusion of already explored feasible configurations throughout Step. 5 (See Fig. 4.6).



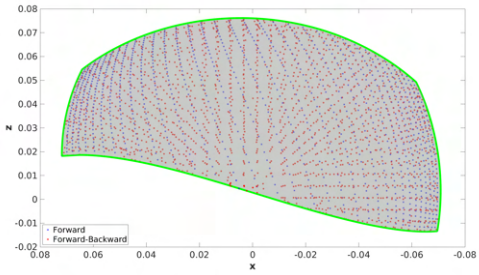
(a) Soft Robot Structure $n_\tau = 3$.



(b) Oxy plan view.



(c) Ozy plan view.



(d) Ozx plan view.

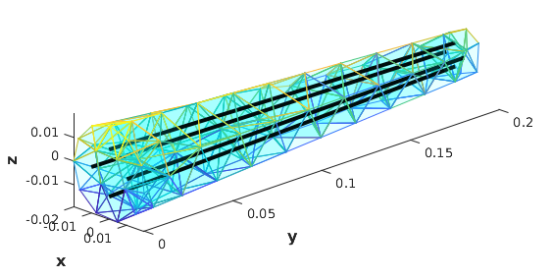
Figure 4.15: Scenario 2 - $\mathcal{W}_{E}$ estimation via forward approach (blue points) and forward-backward (gray area) approach.

4.4.2.3 Scenario 3: trunk-like soft robot actuated by four cables.

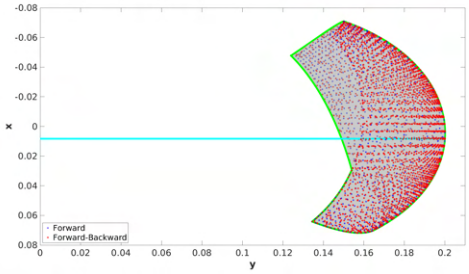
In this scenario (as shown by Fig. 4.16a), we consider the same configuration of the last scenario in Section 3.3.2.3 from Chapter 3. The applied force is of the following range: $\mathcal{T}^{(0)} = \mathcal{T} = [0\ 100] \times [0\ 100] \times [0\ 150] \times [0\ 150]$.

Next, we apply the proposed forward-backward interval analysis approach by following the procedure presented in Section 4.4. For this scenario, the forward-backward approach took 42 complete iterations and computed a total 3667 feasible points in the workspace with a time computation complexity of 3733 seconds. Those obtained feasible points are depicted by red colored points in Figs. 4.16b, 4.16c and 4.16d, and the union of all small neighborhood of those feasible points gives the estimation of the workspace $\mathcal{W}_{E}$, which is represented by the gray zone with green contour. It is clear to see from the three-view drawing that the workspace estimation result obtained via forward-backward method coincides with that obtained via forward approach.

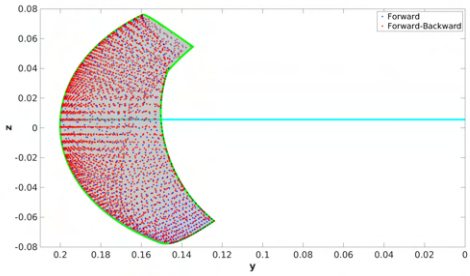
In addition, for a complete 42 iterations it is expected to have $\mathcal{W}_{E_d} = 6^{42} = 4.81 \times 10^{32}$ points to explore. However, what we actually explored is a total of $\mathcal{W}_{E_d} = 10626$ points, and a final total of $\mathcal{W}_{E_s} = 3667$ feasible points. This significant reduction was partially due to the elimination process of inappropriate configurations from Step. 2 (See Fig. 4.3), but mainly due to the elimination of all feasible configurations that were already explored during Step. 5 (See Fig. 4.6).



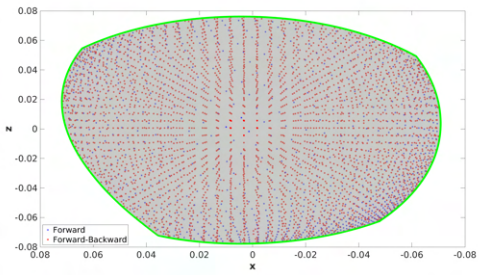
(a) Soft Robot Structure $n_\tau = 4$.



(b) Oxy plan view.



(c) Ozy plan view.



(d) Ozx plan view.

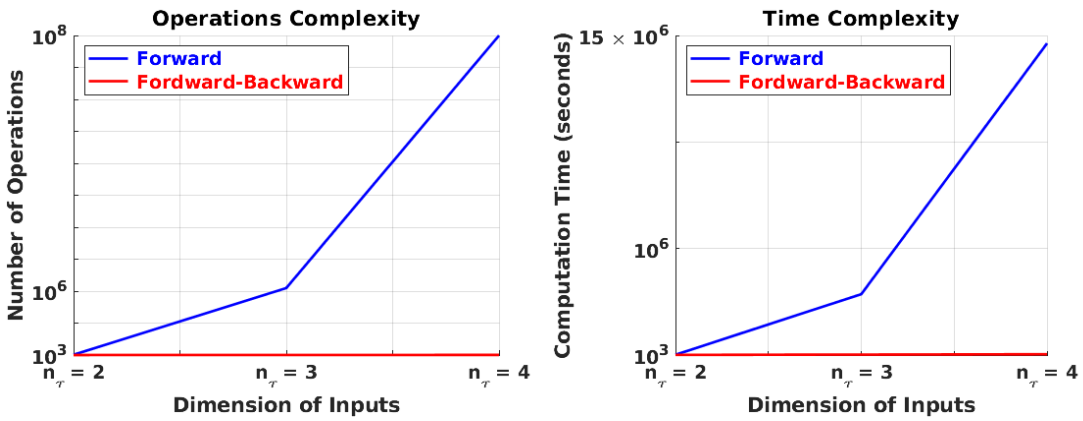
Figure 4.16: Scenario 3 - $\mathcal{W}_E$ estimation via forward approach (blue points) and forward-backward (gray area) approach.

4.4.2.4 Notes on computational efficiency

In the following table (see Table 4.2) and illustrations (see Fig. 4.17a and Fig. 4.17b), we summarize the operations and time computation complexity of each scenario.

Table 4.2: Operations and time computational complexity: forward vs forward-backward.

	Operations		Time (seconds)	
Approach / Inputs	Forward	Forward - Backward	Forward	Forward - Backward
$n_\tau = 2$	100^2	635	1463	757
$n_\tau = 3$	100^3	1941	$\approx 1.4 \times 10^5$	2063
$n_\tau = 4$	100^4	3667	$\approx 1.4 \times 10^7$	3733



(a) Operations.

(b) Time

Figure 4.17: Operations complexity and computational time of the forward and forward-backward approaches for the workspace estimation of the investigated scenarios using the FEM model.

4.5 Conclusion

In this chapter, we have proposed an interval analysis approach, named the forward-backward approach after its procedure, which consists of exploring the whole feasible configurations of the workspace by discretizing the end-effector's domain.

The proposed approach was implemented for both the PCS and the FEM models, where we have shown its effectiveness in reducing the computation complexity and computational time necessary to estimate the workspace of soft robots in contrast to the forward approach which explodes exponentially when increasing the dimension of the actuators.

On the other hand, since this approach consists of determining all the possible configurations that are feasible, it is useful in identifying interior and exterior information about the workspace. However, this approach is exhaustive in the sense that it explores all feasible configurations of the workspace instead of mapping only its interior and exterior boundaries.

Chapter 5

Continuation Approach for Workspace Estimation: PCS case

5.1 Introduction

Due to the limitations encountered by the optimization approach in Chapter 3 that cannot provide knowledge of the interior configurations and boundaries of the workspace of soft robots, and also due to the limitations of the interval analysis based forward-backward approach in Chapter 4 is exhaustive by exploring the whole workspace, it is logical to provide an alternative methodology allowing us to determine only but all boundaries of soft robots workspaces, both interior and exterior.

Such a method is called the continuation approach [8] that is based on the theory of bifurcation. This method is founded on the mathematical theory of bifurcation that was originally introduced by the French mathematician Henri Poincaré [91], and it consists of studying the solutions of nonlinear differential equations, in addition to the study of possible variations in the topological structure of a vector field family. Subsequently, Henri Poincaré classified various types of bifurcation points [92]. Accordingly, this approach takes into account possible bifurcation behavior in continuation calculation while mapping the interior and exterior boundaries of soft robots workspaces.

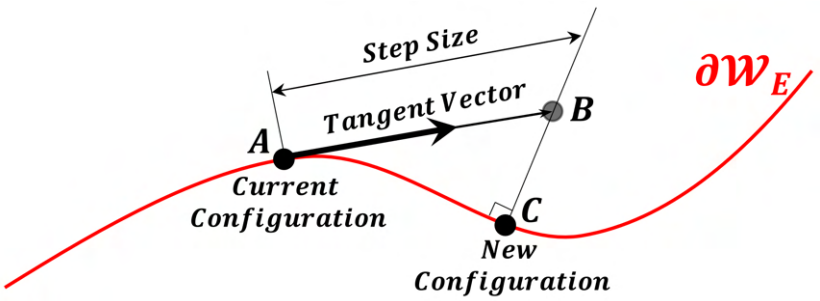


Figure 5.1: Overall view of the continuation method.

The continuation method [55] applies the idea that the workspace boundary (noted as $\partial \mathcal{W}_E$) of a soft robot can be extracted from the set of its output singularities. Starting from an initial

point on the boundary (i.e., point A as shown in Fig. 5.1), this method firstly classifies the status of the boundary point, based on which it calculates the associated tangent vectors, and then numerically predicts its evolution tendency which gives point B. Note that this predicted point B might not be still on $\partial\mathcal{W}_{_E}$, thus a correction procedure is applied to seek a point of $\partial\mathcal{W}_{_E}$ by minimizing the distance to B which yields point $C \in \partial\mathcal{W}_{_E}$. Such a procedure is for the purpose of mapping the point B to one-dimensional solution curves.

Briefly speaking, the proposed continuation approach can be summarized in the following procedure:

1. Formulating the analytical criteria of the workspace boundary.

2. Finding an initial point on the workspace boundary.

3. Calculating tangent vectors to continuation curves at boundary points.

4. Mapping one-dimensional solution curves.

In the following, we will outline the realization of the continuation approach for the PCS model and later discuss the possibility of its extension to the FEM model.

5.2 Implementation of the Methodology

In this section, we present the specific details of the continuation approach applied to the PCS model.

5.2.1 Formulating the Analytical Criteria of the Workspace Boundary

Intuitively, a workspace boundary might be caused by the physical constraint (such as bounded length) of the soft robots, or by the boundedness of actuators, i.e., $\tau_k \in [\underline{\tau}_k, \overline{\tau}_k]$ where $\underline{\tau}_k$ and $\overline{\tau}_k$ represent respectively the lower and upper bound of the kth actuator. In order to take into account actuator boundedness, we introduce two intermediate variables $s_{2k-1} \in \mathbb{R}$ and $s_{2k} \in \mathbb{R}$ satisfying $\underline{\tau}_k - \tau_k + \frac{1}{2}s_{2k-1}^2 = 0$ (equivalent to $\tau_k \geqslant \underline{\tau}_k$), and $-\overline{\tau}_k + \tau_k + \frac{1}{2}s_{2k}^2 = 0$ (equivalent to $\tau_k \leqslant \overline{\tau}_k$). The purpose of the introduction of the variable $s \in \mathbb{R}^{2n_\tau}$ is to treat the boundedness of the actuators in the same way as the mathematical model (e.g., (2.48)) via algebraic equations.

Following the above thoughts, in order to formulate the analytical criteria of the workspace boundary by taking into account the saturation of actuators, let us introduce the following generalized coordinates vector:

$$Q = \left[u^T, \tau^T, s^T\right]^T \equiv \left[u^T, Z^T\right]^T \tag{5.1}$$

where the actuators vector $\tau \in \mathbb{R}^{n_\tau}$ and the intermediate variable vector $s \in \mathbb{R}^{n_s}$ are combined in the vector $Z = \left[\tau^T, s^T\right]^T \in \mathbb{R}^{n_Z}$, $Q \in \mathbb{R}^{n_Q}$ with $n_{_Q} = n_u + n_{_Z}$, $n_{_Z} = n_\tau + n_s$, and $n_s = 2n_\tau$.

With the introduced intermediate variable vector s, the bounded inequality constraints of each actuator can be then transformed into equality constraints, and this enables us to achieve a generalized formulation of the studied soft robot with the following mechanical constraints

$\Psi(Q)$:

$$\Psi(Q) = \Psi(u, Z) = \begin{bmatrix} u - \mathfrak{E}_1 g\big(q(\tau)\big)\mathfrak{E}_2^T \\ \underline{\tau}_1 - \tau_1 + \dfrac{1}{2}s_1^2 \\ -\overline{\tau}_1 + \tau_1 + \dfrac{1}{2}s_2^2 \\ \vdots \\ \underline{\tau}_k - \tau_k + \dfrac{1}{2}s_{2k-1}^2 \\ -\overline{\tau}_k + \tau_k + \dfrac{1}{2}s_{2k}^2 \\ \vdots \\ \underline{\tau}_{n_\tau} - \tau_{n_\tau} + \dfrac{1}{2}s_{n_s-1}^2 \\ -\overline{\tau}_{n_\tau} + \tau_{n_\tau} + \dfrac{1}{2}s_{n_s}^2 \end{bmatrix} = 0 \tag{5.2}$$

where $\Psi(u, Z) \in \mathbb{R}^m$ is the generalized mechanical constraints, with $m = n_u + n_s$, and the generalized strain vector q corresponding to the actuators vector τ is calculated via (2.51).

Remark 6. *Note that for a given actuator configuration $\tau = [\tau_1, \cdots, \tau_{n_\tau}]^T$ with $\tau_k = [\underline{\tau}_k, \overline{\tau}_k]$ for $1 \leqslant k \leqslant n_\tau$, its associated intermediate variables s_{2k-1} and s_{2k} can be uniquely determined by:*

$$s_{2k-1} = \sqrt{2|\tau_k - \underline{\tau}_k|},\ s_{2k} = \sqrt{2|\overline{\tau}_k - \tau_k|} \tag{5.3}$$

i.e., $Z = \big[\tau^T, s^T\big]^T$ can be uniquely determined by τ. Furthermore, by using numerical approach to solve $\Psi(u, Z) = 0$, like Newton method, we can then obtain the associated value of u.

Considering the above formulation of the generalized mechanical constraints (5.2), the generalized form of a soft robot $\mathcal{W}_{_E}$ can therefore be explicitly reformulated as follows:

$$\mathcal{W}_{_E} = \{\ u \in \mathbb{R}^{n_u} \mid \Psi(u, Z) = 0,\ \text{for some } Z\ \} \tag{5.4}$$

Now, consider a soft robot configuration in which its end-effector position lies on the workspace boundary of (5.4). Physically, this situation is equivalent to a singular behavior in which the soft robot end-effector cannot move in certain directions. Mathematically, we can deduce that the soft robot's workspace boundary is a subset of its workspace at which the sub-Jacobian matrix of the generalized mechanical constraints $\Psi(u, Z)$ with respect to the actuators vector τ and the intermediate variable s, denoted as $\Psi_{_Z}(u, Z) \in \mathbb{R}^{m \times n_z}$, is row-rank deficient [56], and it can be defined by the following:

Definition 3. *The workspace boundary $\partial\mathcal{W}_{_E}$ of a soft robot is a subset of its workspace $\mathcal{W}_{_E}$, at which the sub-Jacobian $\Psi_{_Z}(u, Z) \in \mathbb{R}^{m \times n_z}$ is row-rank deficient, i.e.,*

$$\partial\mathcal{W}_{_E} = \{\ u \in \mathcal{W}_{_E} \mid \textit{rank}\ \Psi_{_Z}(u, Z) < m,\ \textit{for } Z \in \mathbb{R}^{n_z}\ \textit{with } \Psi(u, Z) = 0\ \}$$

Equivalently, the row-rank deficient of $\Psi_{_Z}(u, Z)$ implies that there exists an orthonormal vector $\gamma \in \mathbb{R}^m$ forming the nullspace of $\Psi_{_Z}^T$, i.e., $\Psi_{_Z}^T(u, Z)\gamma = 0$. Thus, an analytical condition [62] for a soft robot's workspace boundary can be reformulated as:

$$\partial\mathcal{W}_{_E} = \{\ u \in \mathcal{W}_{_E} \mid \Psi_{_Z}^T(u, Z)\gamma = 0, \gamma^T\gamma = 1,\ \Psi(u, Z) = 0\} \tag{5.5}$$

In accordance with (5.5), a composite vector is introduced:

$$Y = \left[u^T, Z^T, \gamma^T\right]^T \equiv \left[Q^T, \gamma^T\right]^T \tag{5.6}$$

where $Y \in \mathbb{R}^{n_Y}$, $n_Y = n_u + n_Z + m = n_Q + m$.

Since boundary points must be characterized by the generalized vector Q, that satisfy (5.5), then, stated in terms of the composite vector Y, points on $\partial\mathcal{W}_E$ can be characterized by the following composite set of equations:

$$G(Y) = \begin{bmatrix} \Psi(u, Z) \\ \Psi_z^T(u, Z)\gamma \\ \frac{1}{2}(\gamma^T\gamma - 1) \end{bmatrix} = 0 \tag{5.7}$$

Clearly, the above equation is a system of $m + n_Z + 1$ equations with $n_u + n_Z + m$ variables. Thus the dimension of its solutions, equivalently the dimension of $\partial\mathcal{W}_E$, is equal to $n_u - 1$.

Accordingly, when the end-effector position is two-dimensional $u \in \mathbb{R}^2$, then $\partial\mathcal{W}_E$ is a one-dimensional curve. On the other hand, in the case where soft robots are used in spatial applications, the output vector is three-dimensional $u \in \mathbb{R}^3$, implying that $\partial\mathcal{W}_E$ is two-dimensional.

Remark 7. *In spatial applications, this thesis proposes an approach to calculating sets that are of dimension two by mapping the achievable one-dimensional $\partial\mathcal{W}_E$ curves on the Oxy, Oyz, and Oxz plans of the workspace boundary solution set (5.7).*

Moreover, a practical approach [62] to map two-dimensional workspace boundary sets has been proposed, where the purpose is to define a linear relation within the output coordinates by introducing a plan intersecting with the workspace boundary in a one-dimensional curve.

5.2.2 Finding an Initial Point on the Workspace Boundary

In order to map the workspace boundary, we need first to figure out how to determine an initial boundary point u^{b_0}. For concrete applications, it is a nontrivial task to find a point on the workspace boundary $\partial\mathcal{W}_E$ (as shown in Fig. 5.2).

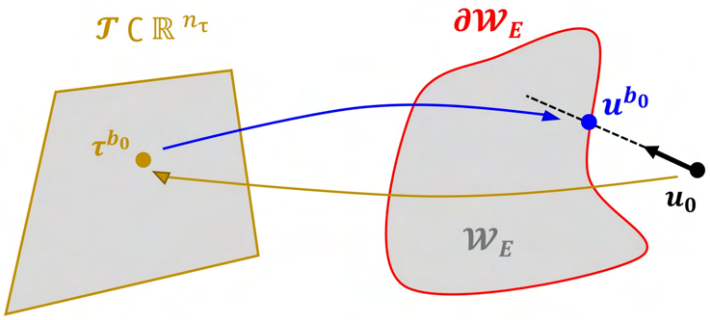


Figure 5.2: A boundary point u^{b_0}.

Assume that a radiating point u_0 is selected, and that u_0 is exterior to $\mathcal{W}_E$. It is now proposed that, consistent with Definition 3, a boundary point u^{b_0} in a particular direction emanating from the radiating point u_0 (as depicted by Fig. 5.2), can be determined by solving

the following constrained optimization problem:

$$\begin{aligned} \tau^{b_0} = \min_{\tau} \quad & \|u(q) - u_0\|_2^2 \\ \text{s.t.} \quad & \tau \in \mathcal{T} \\ & K\ q = H(q)\tau + F\,(q) \\ & u(q) = \mathfrak{E}_1 g\big(q(\tau)\big)\mathfrak{E}_2^T \end{aligned} \tag{5.8}$$

where τ^{b_0} is the optimal and feasible actuators vector (since it satisfies the bounded constraint) for the purpose of minimizing the nonlinear cost function $f(q) = \|u(q) - u_0\|_2^2$, which represents the distance between u_0 and $u(q)$.

From the solution τ^{b_0}, and according to Proposition 1, we calculate the corresponding generalized strain vector q^{b_0} (2.51). Then, we compute the strain twist $\hat{\xi}_i^{b_0}$ related to the strain vector $q_i^{b_j}$ of each body i using (2.49b), and based on which we derive the configuration matrix $g(q^{b_0})$ associated to the generalized strain vector q^{b_0} from (2.49a). Finally, we deduce the boundary point position u^{b_0} using (2.48b).

Using the values of u^{b_0}, τ^{b_0}, and s^{b_0}, we establish the corresponding generalized vector Q^{b_0} using (5.2). Since the obtained solution verifies the workspace boundary definition (see Definition 3), what remains now is to find a vector γ^{b_0} that satisfies (5.5), which can be done by calculating the nullspace of $\Psi_{_Z}^T(Q^{b_0})$.

Finally, the obtained vector γ^{b_0} with the generalized vector Q^{b_0} yields the desired initial composite vector Y^{b_0} (5.6) from which the algorithm that consists of mapping the workspace boundary solution curves (5.7) will start.

Regarding the implementation of the methodology, we need to calculate the sub-Jacobian $\Psi_{_Z}(u, Z)$, which can be formulated as follows:

$$\Psi_{_Z}(u, Z) = \begin{bmatrix} -\nabla_\tau\big[u(q)\big] & 0_{n_u \times n_s} \\ \Delta_{\pm 1} & \Delta_s \end{bmatrix} \tag{5.9}$$

with $\nabla_\tau\big[u(q)\big] = \nabla_\tau(q) J^T(q)[0_3\ R(q)]^T$ being already established in Theorem 1, and $\nabla_\tau(q) = H^T\,(q)\left[\mathfrak{B}^{-1}(\tau, q)\right]^T$ being defined in Assumption 1, with $\mathfrak{B}(\tau, q, L, \alpha) = K(L, \alpha) - H_q(q, L, \alpha)(I_n \otimes \tau) - F_q(q, L, \alpha)$.

The matrices $\Delta_{\pm 1}$ and Δ_s are equal to the following:

$$\Delta_{\pm 1} = \begin{bmatrix} -1 & \cdots & 0 \\ +1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \\ 0 & \cdots & +1 \end{bmatrix}, \quad \Delta_s = \begin{bmatrix} s_1 & & \\ & \ddots & \\ & & s_{n_s} \end{bmatrix} \tag{5.10}$$

The initial radiating point u_0 in 5.2, may be generated using the same approach explained in Section 3.2.2 of Chapter 3.

Alternatively, we could also directly select a limit point, i.e., a point for which the related actuators vector is equal to a combination set of upper and lower values of each actuator, as an initial boundary point. Such a point is always located on $\partial\mathcal{W}_{_E}$, according to Definition 3, since its associated sub-Jacobian is row-rank deficient due to the actuators saturation (because one or multiple variables s will be zero when some actuators reach their upper or lower bounds, and thus the matrix Δ_s in (5.10) will cause the sub-Jacobian $\Psi_{_Z}$ to be row-rank deficient).

5.2.3 Calculating Tangent Vectors to Continuation Curves at Boundary Points

As pointed out by Fig. 5.1, mapping $\partial\mathcal{W}_{_E}$ consists of finding tangent vectors $h(Y)$ to solution curves at each boundary point met during the continuation process. But before tangent calculation, we need to identify the status of each boundary point met along the mapping of the workspace boundaries.

Actually, there exist two different types of boundary points [55], a regular boundary point, and a bifurcation point. A boundary point can be classified as regular [55] if the Jacobian matrix of the composite set that characterizes candidate points on the workspace boundary $G(Y)$ defined in (5.7), denoted as $G_{_Y}(Y) \in \mathbb{R}^{n_Y-1\times n_Y}$, has full row-rank, and $\Psi_{_Z}(u,Z)$ is row-rank deficient 1.

On the other hand, a bifurcation point is where at least two solution curves can cross, and it can be classified into two different categories, simple and multiple [55]. A bifurcation point may be classified as simple [124] if both $G_{_Y}(Y)$ and $\Psi_{_Z}(u,Z)$ are row-rank deficient 1. Finally, a multiple bifurcation point [93] can be identified by two conditions, the first is that $G_{_Y}(Y)$ is row-rank deficient of degree 1, and the second is that $\Psi_{_Z}(u,Z)$ is row-rank deficient of degree 2.

These possible cases are summarized in Table 5.1.

Boundary point \ Row-Rank		$\Psi_{_Z}(u,Z)$	$G_{_Y}(Y)$
Regular		$m-1$	$n_{_Y}-1$
Bifurcation	Simple	$m-1$	$n_{_Y}-2$
	Multiple	$m-2$	$n_{_Y}-2$

Table 5.1: Classification of the possible categories of boundary points.

In the following, we will explain in detail how, in each case, tangent vectors to continuation curves may be calculated.

5.2.3.1 Regular boundary points

Within the solution curve (5.7), a regular point u^{b_r} characterized by the composite vector Y^{b_r} (as shown by Fig. 5.3), where at most a single solution curve is passing through, may be met. At such a point, the row-rank deficiency of $\Psi_{_Z}$ is 1, and $G_{_Y}(Y^{b_r})$ has full row-rank.

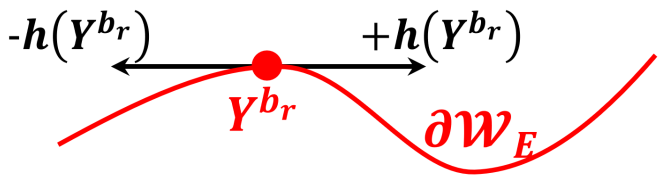


Figure 5.3: Regular boundary point - tangent vector.

At such a point, a tangent vector $h(Y^{b_r})$ (as illustrated by Fig. 5.3) to the set (5.7) may be calculated based on the following theorem [6, 106]:

Theorem 4. *Let G: $S \subset \mathbb{R}^{n_Y} \mapsto \mathbb{R}^{n_Y-1}$, be of class C^r, $r \geqslant 1$, on an open set $S \subset \mathbb{R}^{n_Y}$. Thus, at a point Y^{b_r}, if $G_{_Y}$ has full row-rank, then a unit tangent vector $h(Y^{b_r}) \in \mathbb{R}^{n_Y}$ is uniquely*

defined by:

$$\begin{aligned} G_{_Y}(Y^{b_r})h(Y^{b_r}) &= 0 \\ h^T(Y^{b_r})h(Y^{b_r}) &= 1 \\ \begin{vmatrix} G_{_Y}(Y^{b_r}) \\ h^T(Y^{b_r}) \end{vmatrix} &> 0 \end{aligned} \tag{5.11}$$

The tangent vector $h(Y^{b_r})$ determined by (5.11) can be calculated at each point along the workspace boundary solution set (5.7) as long as $G_{_Y}(Y^{b_r})$ has full row-rank.

However, difficulties may arise when multiple solution curves cross at bifurcation points of (5.7). According to Table 5.1, $G_{_Y}$ is row-rank deficient at such configurations, and (5.11) is unable to determine a unique tangent vector. Therefore, new techniques must be adopted [93, 124] in order to compute tangent vectors $h(Y)$ to continuation curves at both simple and multiple bifurcation points.

5.2.3.2 Bifurcation points - Simple bifurcation point

While proceeding along the workspace boundary solution set defined by (5.7), a point u^{b_s} characterized by its composite vector Y^{b_s} (as depicted by Fig. 5.4) may be found at which the Jacobian $G_{_Y}(Y^{b_s})$ becomes row-rank deficient of degree 1, and the rank deficiency of $\Psi_{_Z}$ is 1. Such a point is classified as a simple bifurcation point [124], where a pair of solution curves (See Fig. 5.4) characterized by a pair of tangent vectors $h(Y^{b_s})$ satisfying (5.11) are traversing it.

For the purpose of identifying the nature of a singular behavior at such a point Y^{b_s}, we perform an elementary row decomposition, e.g. singular value decomposition (SVD) or QR decomposition, to simplify $G_{_Y}(Y^{b_s})$ into the following form:

$$\mathfrak{E}G_{_Y}(Y^{b_s}) = \begin{bmatrix} \mathfrak{F}_{_Y}(Y^{b_s}) \\ 0 \end{bmatrix} \tag{5.12}$$

where $\mathfrak{E}$ is an elementary non-singular matrix, and $\mathfrak{F}_{_Y}(Y^{b_s})$ is an $(n_{_Y}-2)\times n_{_Y}$ matrix with full row-rank. Applying elementary row operation to the workspace boundary solution set (5.7) using $\mathfrak{E}$, we get the following equivalent equation:

$$\breve{G}(Y^{b_s}) \equiv \begin{bmatrix} \mathfrak{F}(Y^{b_s}) \\ \mathfrak{f}(Y^{b_s}) \end{bmatrix} = \mathfrak{E}G(Y^{b_s}) = 0 \tag{5.13}$$

Using (5.12) and (5.13), we observe that $\mathfrak{F}_{_Y}(Y^{b_s})$ is the Jacobian matrix of $\mathfrak{F}(Y^{b_s})$, and the Jacobian of the function $\mathfrak{f}(Y^{b_s})$ is $\mathfrak{f}_{_Y}(Y^{b_s}) = 0$.

Since $\mathfrak{F}_{_Y}(Y^{b_s})$ is full row-rank, i.e., $rank\ \mathfrak{F}_{_Y}(Y^{b_s}) = n_{_Y}-2$, then the nullspace of $\mathfrak{F}_{_Y}(Y^{b_s})$ (equivalently, the nullspace of $G_{_Y}(Y^{b_s})$) is of dimension 2, which generates two different non-zero solutions of the following equation:

$$\mathfrak{F}_{_Y}(Y^{b_s})\zeta^i = 0,\ i = 1, 2 \tag{5.14}$$

that can be orthonormalized; $\zeta^i\zeta^j = \delta_{ij},\ i, j = 1, 2$, where δ is the Dirac delta function.

The purpose now is to determine the pair of tangent vectors $h(Y^{b_s})$ to continuation curves at Y^{b_s} by writing it in the form of $h(Y^{b_s}) = \sum_j \alpha_j\zeta^j$. With ζ being calculated via (5.14), the remaining question now is about how to compute the vector α.

Using Taylor's expansion, the tangent vectors of the solution curve can be calculated. Taylor's expansion for functions of $n_{_Y}$ variables is stated in [5] as follows:

Theorem 5. *According to [5], let G: $S \subset \mathbb{R}^{n_Y} \mapsto \mathbb{R}^{n_Y-1}$, $n_Y \geqslant 2$, be of class C^r, $r \geqslant 1$, on the open set $S \subset \mathbb{R}^{n_Y}$. Then, the quadratic Taylor approximation is defined by:*

$$\breve{G}(Y^{b_s} + \Delta Y) \approx \breve{G}(Y^{b_s}) + \begin{bmatrix} \mathfrak{F}_Y(Y^{b_s})\alpha_i\zeta^i \\ \frac{1}{2}(\alpha_j\zeta^j)\mathfrak{f}_{YY}(Y^{b_s})(\alpha_k\zeta^l) \end{bmatrix} \tag{5.15}$$

where the Hessian matrix of $\mathfrak{f}(Y)$, evaluated at Y^{b_s}, is the matrix:

$$\mathfrak{f}_{YY}(Y^{b_s}) \equiv \left[\frac{\partial \mathfrak{f}(Y)}{\partial Y_i Y_j}\right]_{Y=Y^{b_s}} \tag{5.16}$$

Since $Y^{b_s} + \Delta Y$ and Y^{b_s} must satisfy the boundary workspace solution set (5.7), then we have:

$$\breve{G}(Y^{b_s} + \Delta Y) = \breve{G}(Y^{b_s}) = 0$$

Taking that into consideration, (5.15) becomes:

$$\begin{bmatrix} \mathfrak{F}_Y(Y^{b_s})\alpha_i\zeta^i \\ \frac{1}{2}(\alpha_j\zeta^j)\mathfrak{f}_{YY}(Y^{b_s})(\alpha_k\zeta^k) \end{bmatrix} = 0 \tag{5.17}$$

Noting that, for all α_i, the upper sub-vector term $\mathfrak{F}_Y(Y^{b_s})\alpha_i\zeta^i = 0$, since ζ^i represents the nullspace of $\mathfrak{F}_Y(Y^{b_s})$ as stated in (5.14). Finally, (5.17) is reduced to the following:

$$\alpha^T\left[\zeta^i\mathfrak{f}_{YY}(Y^{b_s})\zeta^j\right]\alpha = 0 \tag{5.18}$$

where $\alpha = [\alpha_1, \alpha_2]^T$.

Using elementary row decomposition, e.g. eigenvalue decomposition, the above quadratic equation may be solved. Another way is to transform (5.18) to a quadratic equation in terms of α_1 and α_2 and solving it using the quadratic formula. No matter which way is applied, solving (5.18) yields two normalized solution vectors α^1 and α^2. Finally, with ζ^1 and ζ^2 being calculated via (5.14), we obtain a pair of tangent vectors $h(Y^{b_s}) = [h^1(Y^{b_s})^T\ h^2(Y^{b_s})^T]^T$ depicted by Fig. 5.4, and formulated by the following:

$$\begin{aligned} h^1(Y^{b_s}) &= \alpha_1^1\zeta^1 + \alpha_2^1\zeta^2 \\ h^2(Y^{b_s}) &= \alpha_1^2\zeta^1 + \alpha_2^2\zeta^2 \end{aligned} \tag{5.19}$$

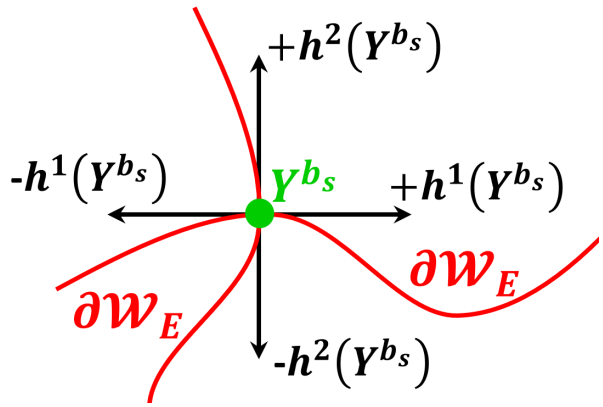


Figure 5.4: Simple bifurcation point - tangent vector.

5.2.3.3 Bifurcation points - Multiple bifurcation point

During the continuation calculation, a point u^{b_m} characterized by the composite vector Y^{b_m} (as depicted by Fig. 5.5) may be encountered at which the Jacobian $\Psi_{_Z}$ becomes row-rank deficient of degree 2, and the rank deficiency of $G_{_Y}(Y^{b_m})$ is 1. Such a point is classified as a multiple bifurcation point [93], where numerous branches may occur (See Fig. 5.5) depending on the design of the soft robot.

This thesis uses a numerical method [93] that consists of finding, for all branches intersecting at Q^{b_m}, all vectors γ such that $G_{_Y}(Q^{b_m}, \gamma)$ for each branch is row-rank deficient.

The adopted strategy can be separated into two major steps:

[Step 1] Calculating all solutions for γ at a multiple bifurcation point.

[Step 2] Calculating all tangent vectors $h(Q^{b_m}, \gamma)$ to continuation curves.

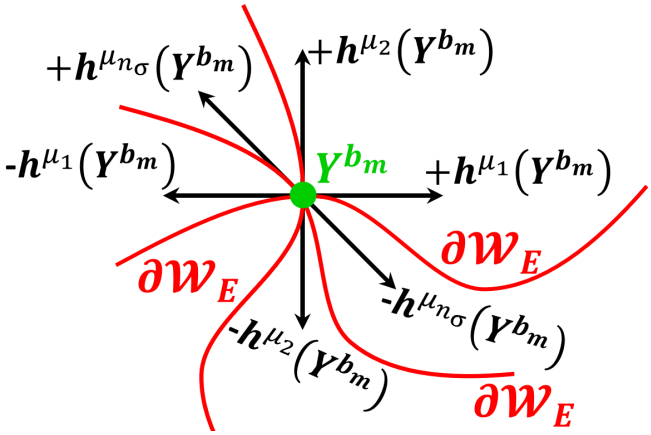


Figure 5.5: Multiple bifurcation point - tangent vector.

Step 1: Calculating all solutions for γ at a multiple bifurcation point

In the first step, vector γ for which $G_{_Y}(Q^{b_m}, \gamma)$ is row-rank deficient will be determined. Since the row-rank deficiency of $\Psi_{_Z}$ is 2, thus, we have 2 orthonormal solutions γ_1 and γ_2 such that:

$$\begin{aligned} \Psi_{_Z}^T \gamma_i &= 0 \\ \gamma_i^T \gamma_j &= \delta_{ij},\ i,j = 1,2 \end{aligned} \tag{5.20}$$

where γ can be represented as a linear combination of γ_1 and γ_2 as:

$$\gamma \equiv \gamma(\beta) = \tan(\beta)\gamma_1 + \gamma_2 \tag{5.21}$$

with $\beta \in [0\ \pi]$. Hence, the purpose of this step is to solve the following problem:

Problem 1. *Calculate all γ for which each corresponding $G_{_Y}(Q^{b_m}, \gamma)$ is row-rank deficient, where:*

$$G_{_Y}(Q^{b_m}, \gamma) = \begin{bmatrix} \Psi_{_u} & \Psi_{_\tau} & \Psi_{_s} & 0 \\ 0 & \left[\Psi_{_Z}^T \gamma\right]_{_\tau} & \left[\Psi_{_Z}^T \gamma\right]_{_s} & \Psi_{_Z}^T \\ 0 & 0 & 0 & \gamma^T \end{bmatrix} \tag{5.22}$$

Since the number of γ is finite, it can be assumed that neither γ_1 nor γ_2 are solutions. Therefore, the solution of (5.21) must verify the condition $\sin(\beta)\cos(\beta) \neq 0$.

Lemma 1. *$G_{_Y}(Q^{b_m}, \gamma)$ is row-rank deficient if and only if there exists a non-zero vector $c = [c_1^T, c_2^T, c_3]^T$, where $c_1 \in \mathbb{R}^m$, $c_2 \in \mathbb{R}^{n_Z}$, and $c_1 \in \mathbb{R}$:*

$$c^T G_{_Y}(Q^{b_m}, \gamma) = 0 \tag{5.23}$$

By developing (5.23) through (5.22), we obtain the following set of equations:

$$c_1^T \Psi_{_\tau} + c_2^T \left[\Psi_{_Z}^T \gamma\right]_{_\tau} = 0 \tag{5.24a}$$

$$c_1^T \Psi_{_s} + c_2^T \left[\Psi_{_Z}^T \gamma\right]_{_s} = 0 \tag{5.24b}$$

$$c_2^T \Psi_{_Z}^T + c_3 \gamma^T = 0 \tag{5.24c}$$

Introduce now μ as: $\mu \equiv \mu(\beta) = \tan(\beta)$, then, the problem may be equivalently reformulated as follows:

Problem 2. *Assume that neither $G_{_Y}(Q^{b_m}, \gamma_1)$ nor $G_{_Y}(Q^{b_m}, \gamma_2)$ is row-rank deficient, the goal now would be to find all $\mu(\beta)$ such that $G_{_Y}(Q^{b_m}, \gamma(\beta))$ is row-rank deficient.*

A necessary condition for $G_{_Y}(Q^{b_m}, \gamma(\beta))$ has already been outlined by the set of equations (5.24a), (5.24b), and (5.24c). Multiplying (5.24c) by γ, we obtain:

$$c_2^T \Psi_{_Z}^T \gamma + c_3 \gamma^T \gamma = 0$$

Using (5.20), we deduce:

$$\Psi_{_Z}^T \gamma = 0 \implies c_3 = 0$$

Taking that into consideration, (5.24c) finally becomes:

$$\Psi_{_Z} c_2 = 0$$

And since the row-rank deficiency of $\Psi_{_Z} : \mathbb{R}^{n_Z} \mapsto \mathbb{R}^m$ is 2, then there exists an orthonormal basis σ of the nullspace of $\Psi_{_Z}^T$ such that:

$$c_2 = d_1\sigma_1 + d_2\sigma_2 + \cdots + d_{n_\sigma}\sigma_{n_\sigma} \equiv \sigma d \tag{5.25}$$

where $\sigma = \{\sigma_1, \sigma_2, \cdots, \sigma_{n_\sigma}\}$, $d = \{d_1, d_2, \cdots, d_{n_\sigma}\}$, and $n_\sigma = n_{_Z} - m + 2$.

From (5.24a), and (5.24b), we obtain the following:

$$c_1^T \left[\Psi_{_\tau}, \Psi_{_s}\right] = -c_2^T \left[\left[\Psi_{_Z}^T \gamma\right]_{_\tau}, \left[\Psi_{_Z}^T \gamma\right]_{_s}\right]$$

For the purpose of formulating c_1 in function of c_2, and since $\left[\Psi_{_\tau}, \Psi_{_s}\right]$ is right invertible, we can write the above equation in the following form:

$$c_1^T = c_2^T \mathfrak{M}(\gamma) \tag{5.26}$$

with:

$$\mathfrak{M}(\gamma) = -\left[\left[\Psi_{_Z}^T \gamma\right]_{_\tau}, \left[\Psi_{_Z}^T \gamma\right]_{_s}\right] \left[\Psi_{_\tau}, \Psi_{_s}\right]^+ \tag{5.27}$$

After, by substituting c_1^T from (5.26) into (5.24b), we get:

$$c_2^T\left(\mathfrak{M}(\gamma)\Psi_{_s}+\left[\Psi_{_Z}^T\gamma\right]_{_s}\right)=0$$

Next, we replace (5.25) into the above equation, we obtain:

$$d^T\mathfrak{H}(\gamma(\beta))=0$$

with

$$\mathfrak{H}(\gamma(\beta))\equiv\sigma^T\left(\mathfrak{M}(\gamma)\Psi_{_s}+\left[\Psi_{_Z}^T\gamma\right]_{_s}\right) \tag{5.28}$$

As stated in Lemma 2.3 of [93], $G_{_Y}(Q^{b_m},\gamma)$ is row-rank deficient if and only if $\mathfrak{H}(\gamma(\beta))$ is row-rank deficient. By substituting (5.21) into (5.28), we achieve:

$$\mathfrak{H}(\gamma(\beta))=\tan(\beta)\mathfrak{A}+\mathfrak{B} \tag{5.29}$$

where:

$$\mathfrak{A}=\sigma^T\left(\mathfrak{M}(\gamma_1)\Psi_{_s}+\left[\Psi_{_Z}^T\gamma_1\right]_{_s}\right) \tag{5.30a}$$

$$\mathfrak{B}=\sigma^T\left(\mathfrak{M}(\gamma_2)\Psi_{_s}+\left[\Psi_{_Z}^T\gamma_2\right]_{_s}\right) \tag{5.30b}$$

Based on Lemma 2.4 in [93], if both $G_{_Y}(Q^{b_m},\gamma_1)$ and $G_{_Y}(Q^{b_m},\gamma_2)$ have full row-rank, then $\mathfrak{A}$ is right invertible. Equivalently, according to Theorem 2.5 in [93], $G_{_Y}(Q^{b_m},\gamma)$ is row-rank deficient if only and if μ is a real eigenvalue of $-\mathfrak{B}\mathfrak{A}^+$.

Step 2: Calculating all tangent vectors $h(Q^{b_m},\gamma)$ to continuation curves.

In this step, we will describe how to calculate the tangent vectors $h(Q^{b_m},\gamma)$ to all solution curves passing through a multiple bifurcation point.

Consistent with Lemma 2.4 in [93], denote χ as the eigenvector corresponding to the eigenvalue μ of the matrix $-\mathfrak{B}\mathfrak{A}^+$. By definition, we obtain:

$$-\mathfrak{B}\mathfrak{A}^+\chi=\mu\chi \tag{5.31}$$

Let us define $\rho_2\in\mathbb{R}^{n_s}$ as the following [93]:

$$\rho_2=\mathfrak{A}^+\chi \tag{5.32}$$

From the above equation, we can easily deduce that $\chi=\mathfrak{A}\rho_2$. Using this, and substituting (5.32) in (5.31), we obtain:

$$-\mathfrak{B}\rho_2=\mu\mathfrak{A}\rho_2$$

which can be reformulated as:

$$(\mu\mathfrak{A}+\mathfrak{B})\rho_2=0$$

Since $\mu=\tan(\beta)$, we get:

$$(\tan(\beta)\mathfrak{A}+\mathfrak{B})\rho_2=0$$

Using (5.29), then (5.28), the above equation can finally be rewritten as follows:

$$\sigma^T\mathfrak{S}=0 \tag{5.33}$$

with $\mathfrak{S} = \left(\mathfrak{M}(\gamma)\Psi_{_s} + \left[\Psi_{_Z}^T \gamma \right]_{_s} \right) \rho_2$.

Since σ is an orthonormal basis of the nullspace of $\Psi_{_Z}^T$, and from (5.33) $\mathfrak{S}$ is orthogonal to σ, then $\mathfrak{S}$ represents the column space of $\Psi_{_Z}^T$ [86]. Therefore, there exists a unique vector $\rho_3 \in \mathbb{R}^m$ in the column space of $\Psi_{_Z}^T$, for which the following is deduced:

$$\mathfrak{S} = -\Psi_{_Z}^T \rho_3 \tag{5.34}$$

Given that ρ_3 and γ are respectively the column space and the nullspace of $\Psi_{_Z}^T$. Therefore, using the first formula of $\mathfrak{S}$ from (5.33), its second formula from (5.34), we obtain the following set of equations:

$$\left(\mathfrak{M}(\gamma)\Psi_{_s} + \left[\Psi_{_Z}^T \gamma \right]_{_s} \right) \rho_2 = -\Psi_{_Z}^T \rho_3 \tag{5.35a}$$

$$0 = \gamma^T \rho_3 \tag{5.35b}$$

which finally allows us to obtain the final form of $\rho_3 \in \mathbb{R}^m$ as follows:

$$\rho_3 = \begin{bmatrix} -\Psi_{_Z}^T \\ \gamma^T \end{bmatrix}^+ \begin{bmatrix} \mathfrak{M}(\gamma)\Psi_{_s} + \left[\Psi_{_Z}^T \gamma \right]_{_s} \\ 0 \end{bmatrix} \rho_2 \tag{5.36}$$

Next, we define $\rho_1 \in \mathbb{R}^{n_u + n_\tau}$ by the following:

$$\rho_1 = -\left[\Psi_{_u}, \Psi_{_\tau} \right]^+ \Psi_{_s} \rho_2 \tag{5.37}$$

Substituting both $\mathfrak{M}(\gamma)$ and ρ_1 respectively from (5.26) and (5.37) into (5.35a), and using (5.35b), we can deduce the following system of equations:

$$\left[\Psi_{_u}, \Psi_{_\tau} \right] \rho_1 + \Psi_{_s} \rho_2 = 0 \tag{5.38a}$$

$$\left[0, \left[\Psi_{_Z}^T \gamma \right]_{_\tau} \right] \rho_1 + \left[\Psi_{_Z}^T \gamma \right]_{_s} \rho_2 + \Psi_{_Z}^T \rho_3 = 0 \tag{5.38b}$$

$$\gamma^T \rho_3 = 0 \tag{5.38c}$$

Eventually, the above set can be reformulated into the following form:

$$\begin{aligned} 0 &= \begin{bmatrix} \left[\Psi_{_u}, \Psi_{_\tau} \right] & \Psi_{_s} & 0 \\ \left[0, \left[\Psi_{_Z}^T \gamma \right]_{_\tau} \right] & \left[\Psi_{_Z}^T \gamma \right]_{_s} & \Psi_{_Z}^T \\ 0 & 0 & \gamma^T \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \end{bmatrix} \\ 0 &= G_{_Y}(Q^{b_m}, \gamma)\ h(Q^{b_m}, \gamma) \end{aligned} \tag{5.39}$$

Finally, it is clear that:

$$h(Q^{b_m}, \gamma) = [\rho_1^T, \rho_2^T, \rho_3^T]^T \tag{5.40}$$

is the tangent vector of the continuation curve corresponding to its associated eigenvalue μ and eigenvector χ.

5.2.4 Mapping One-Dimensional Solution Curves

With tangent vectors being calculated for each possible boundary point situation (as shown in Fig. 5.3, Fig. 5.4, and Fig. 5.5), the final step of the continuation method is to map the solution curves of (5.7).

Given a boundary point u^{b_0} characterized by its composite vector Y^{b_0}, where Y^{b_0} is a solution of the workspace boundary set (5.7), i.e. $G(Y^{b_0}) = 0$, and its tangent vector $h(Y^{b_0})$ being calculated depending on its category (regular boundary point, simple or multiple bifurcation point), a ray is emanated from Y^{b_0}, with an infinitesimally small discretization precision ε, on which a prediction Y^p (as depicted by Fig. 5.6) of the appropriate solution is calculated as follows:

$$Y^p = Y^{b_0} \pm \varepsilon h(Y^{b_0}) \tag{5.41}$$

Next, based on the predicted state Y^p, its associated solution Y^b (as shown in Fig. 5.6) must verify the nonlinear constraint condition (5.7) of the workspace boundary solution set.

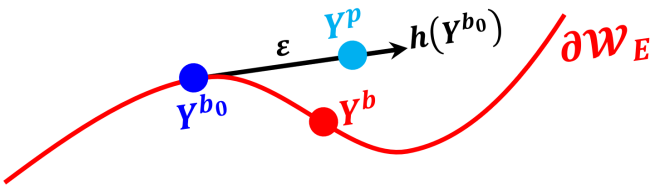


Figure 5.6: Mapping one-dimensional solution curves.

In this thesis, we propose to calculate Y^b (as depicted in Fig. 5.6) using a constrained optimization problem, subject to nonlinear constraints, and formulated by the following:

$$\begin{aligned} Y^b = \min_{Y} \quad & \|Y - Y^p\| \\ \text{s.t.} \quad & G(Y) = 0 \end{aligned} \tag{5.42}$$

where $G(Y)$ defined in (5.7) to characterize the workspace boundary set.

With Y^b being computed, we proceed by repeating the same process for each boundary point met during the continuation calculation until all solution curves of the workspace boundary are mapped.

An equivalent technique to (5.42) presented in [6] has also been proven to be an efficient approach to find the next possible solution on the continuation curve.

The stop condition can be evaluated by verifying whether all limit points have been explored. When it is the case, then the stop condition is true, and the continuation process is terminated by yielding the inner and outer boundaries of the soft robot. If not, there might exist some isolated boundaries which cannot be reached from the chosen initial point. In this situation, we can choose another combination set of upper and lower values of each actuator, which has not yet been explored by the former process, as a new initial point and restart the same process.

In following, we present a brief algorithm (see Algorithm 4) explaining the major steps of the proposed continuation approach in order to map the interior and exterior boundaries of soft robots.

Algorithm 4 Calculate $\partial\mathcal{W}_{_E}$

Require: $u_0, \mathcal{T}, \varepsilon$
  $\partial\mathcal{W}_{_E} \leftarrow \varnothing,\ stop \leftarrow 0$ ▷ Initialization.
  $[\tau^{b_0}, u^{b_0}] \leftarrow solve_{(5.8)}(u_0, \mathcal{T})$ ▷ Initial boundary point.
  $Q^{b_0} \leftarrow solve_{(5.1)}(u_0, \mathcal{T})$ ▷ Initial boundary configuration.
  $\gamma^{b_0} \leftarrow null(\Psi_{_Z}^{T}(Q^{b_0}))$
  $Y^{b_0} \leftarrow [{Q^{b_0}}^{T}, {\gamma^{b_0}}^{T}]^{T}$ ▷ Initial composite vector.
  while $!stop$ **do**
    $flag \leftarrow Table_{5.1}(Y^{b_0})$ ▷ Boundary point class.
    switch $flag$ **do** ▷ Tangent vector.
      case (Regular boundary point)
        $h(Y^{b_0}) \leftarrow solve_{(5.11)}(Q^{b_0}, Y^{b_0})$
      case (Simple bifurcation point)
        $h(Y^{b_0}) \leftarrow solve_{(5.19)}(Q^{b_0}, Y^{b_0})$
      case (Multiple bifurcation point)
        $h(Y^{b_0}) \leftarrow solve_{(5.40)}(Q^{b_0}, Y^{b_0})$
    $Y^{p} \leftarrow Y^{b_0} \pm \varepsilon h(Y^{b_0})$ ▷ Predict.
    $Y^{b} \leftarrow solve_{(5.42)}(Y^{b_0}, Y^{p})$ ▷ Correct.
    $\partial\mathcal{W}_{_E} \leftarrow \partial\mathcal{W}_{_E} \oplus Y^{b}$ ▷ Append.
    $Y^{b_0} \leftarrow Y^{b}$ ▷ Update initial composite vector.
    $stop \leftarrow update(\partial\mathcal{W}_{_E}, Y^{b_0})$ ▷ Update stop condition.
  end while

5.3 Validation and Analysis

We implement both the forward and the continuation approaches on planar and spatial soft robots to deduce the workspace estimation $\mathcal{W}_{_E}$, and the workspace boundary $\partial\mathcal{W}_{_E}$. Table 3.1 presents the numerical values of the investigated soft robot.

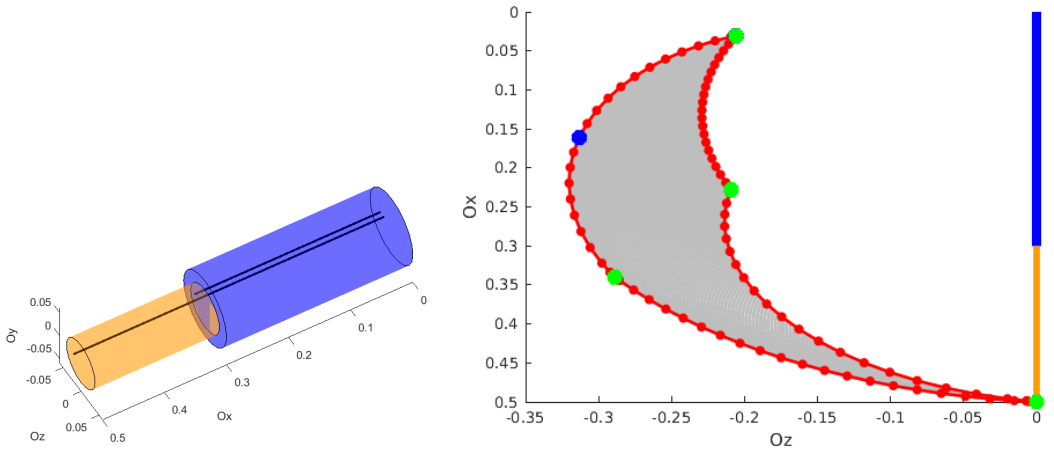
In the following scenarios, the workspace estimation obtained from the forward approach is depicted by gray-colored points. For the continuation approach, we suggest to proceed with a step $\varepsilon = 0.01$.

5.3.0.1 Scenario 1: planar 2-segments 2-tendons soft robot

In the first scenario, we consider the same configuration of the first scenario in Section 3.2.5.1 of Chapter 3. We choose a tension magnitude within $\mathcal{T}_{_{1,2}} = [0, 200]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{_{\mathcal{T}_1}} = [0, 6]N \cdot m$ and $\mathcal{C}_{_{\mathcal{T}_2}} = [0, 4]N \cdot m$.

Starting from an initial boundary configuration calculated via (5.8), which is a regular boundary point in this case (blue point on Fig. 5.7b), we use (5.11) to calculate its corresponding tangent vectors, which yields one tangent vector. Based on this, we calculate its associated prediction via (5.41), then deduce its corresponding solution using (5.42). Proceeding along the solution curve (5.7), we meet four bifurcation points (green points on Fig. 5.7b), between which regular boundary points (red points on Fig. 5.7b) are met. Finally, workspace boundary curves are mapped as shown in Fig. 5.7b.

For this scenario, the continuation approach computed a total of 87 boundary points and took only 9.54 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

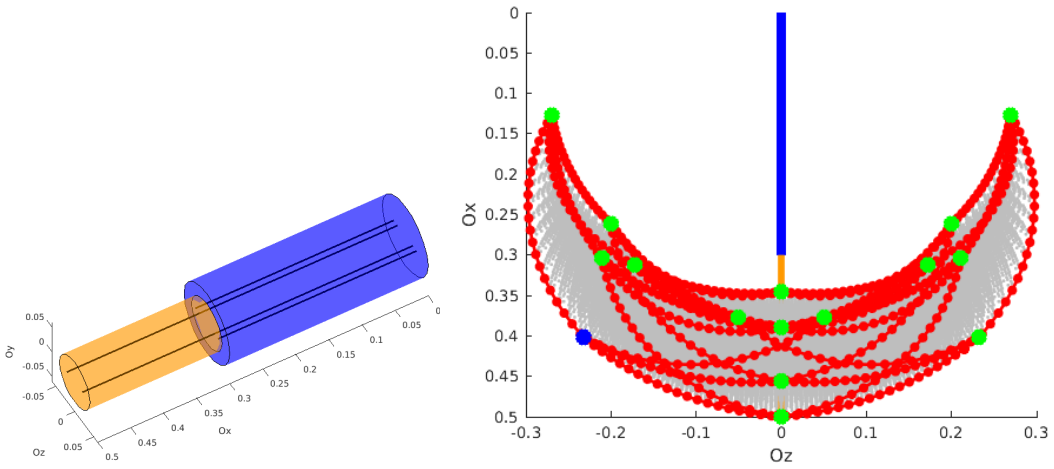
Figure 5.7: Scenario 1 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.2 Scenario 2: planar 2-segments 4-tendons soft robot

In the second scenario, we consider the soft robot of the first scenario and we add two symmetric tendons (See Fig. 5.8a). We choose a tension magnitude within $\mathcal{T}_{1,2} = [0, 150]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,3}} = [0, 4.5]N \cdot m$ and $\mathcal{C}_{\mathcal{T}_{2,4}} = [0, 3]N \cdot m$.

The numerical continuation algorithm starts from the blue point on Fig. 5.8b, which is a limit configuration in this case. Next, using (5.40) we obtain 4 distinct tangent vectors, each one corresponding to its associated solution curve since we have 4 that crosses through this point. After, we calculate the prediction corresponding to each tangent vector (5.41), and we obtain for each curve its associated solution (5.42). Regular boundary points (red points in Fig. 5.8b) are computed between each of the sixteen total bifurcation points, including the initial point (green points in Fig. 5.8b) found for this particular soft robot structure. Finally, interior and exterior boundaries to the planar two segments - four tendons soft robot's workspace are mapped and illustrated by Fig. 5.8b.

For this scenario, the continuation approach computed a total of 687 boundary points and took only 96.35 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

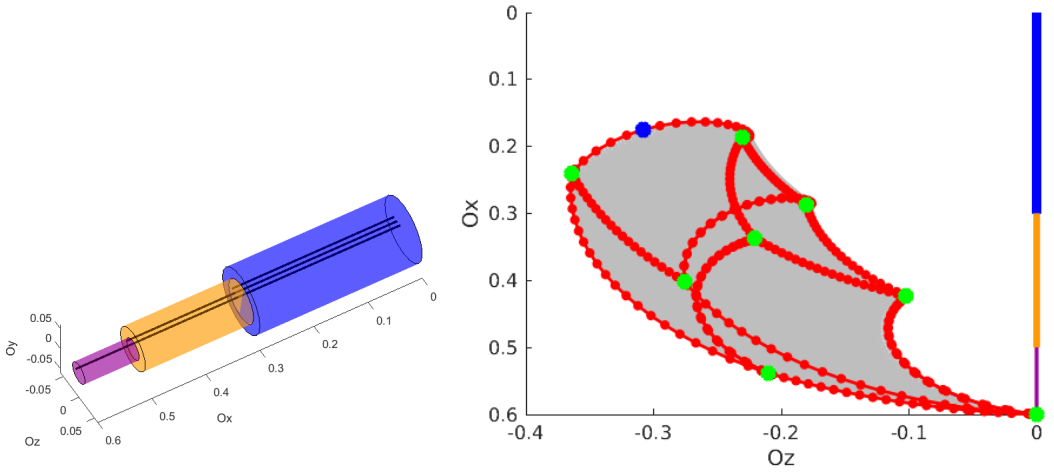
Figure 5.8: Scenario 2 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.3 Scenario 3: planar 3-segments 3-tendons soft robot

In the third scenario (See Fig. 4.10a), we consider the same tendons routing of the third scenario in Section 3.2.5.3 from Chapter 3. We choose a tension magnitude within $\mathcal{T}_{1,2,3} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0, 3]N \cdot m$, $\mathcal{C}_{\mathcal{T}_2} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_1} = [0, 1]N \cdot m$.

Interior and exterior workspace boundaries of this particular soft robot structure are mapped (as shown by Fig. 5.9b), using (5.8) we find an initial boundary configuration (blue point on Fig. 5.9b) that is a regular boundary point, based on which, we calculate its unique corresponding tangent vector through (5.11). Next, through (5.41) we predict the possible solution, and we determine its associated solution (5.42), proceeding all the way along solution curves we identify 8 bifurcation points (green points in Fig. 5.9b), with regular boundary points (magenta points in Fig. 5.9b) lying between them.

For this scenario, the continuation approach computed a total of 327 boundary points and took only 40.62 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

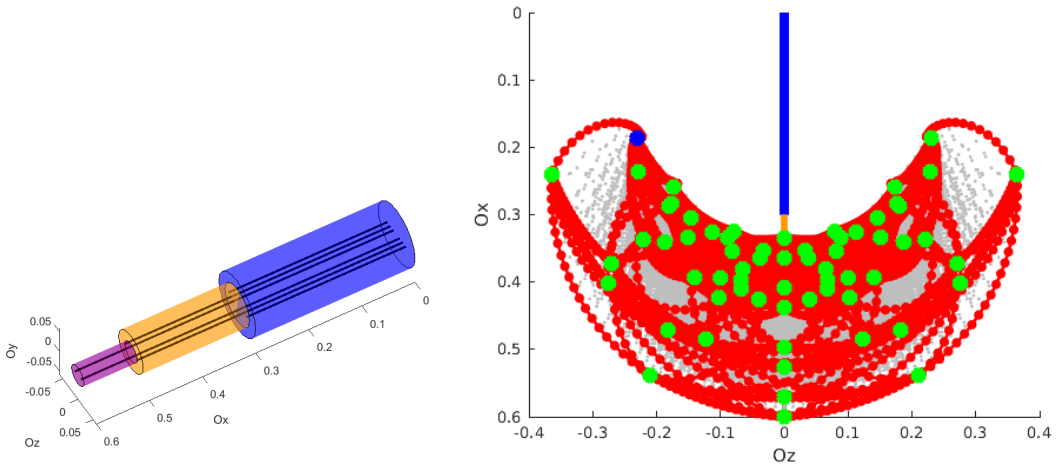
Figure 5.9: Scenario 3: $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.4 Scenario 4: planar 3-segments 6-tendons soft robot

In this scenario, three additional tendons are routed in a fashion where they are symmetric to the tendons configuration in the third scenario (See Fig. 3.9a). We choose a tension magnitude within $\mathcal{T}_{1,\cdots,6} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4}} = [0, 3]N \cdot m$, $\mathcal{C}_{\mathcal{T}_{2,5}} = [0, 2]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6}} = [0, 1]N \cdot m$.

Solution curves of the workspace boundary set (5.7) are mapped beginning from an initial boundary point corresponding to a limit point calculated via a random combination set of lower and upper values of each actuators vector bounds and is located in the interior boundary. From this initial point, relevant tangent vectors are calculated (5.40), yielding 6 possible directions to explore. Next, we predict the solution corresponding to each boundary curve from (5.41), and deduce their appropriate boundary configuration (5.42) (see Fig. 5.10b). Finally, solution sets are calculated and exterior and interior workspace boundaries are mapped (as shown by Fig. 5.10b).

For this scenario, the continuation approach computed a total of 4126 boundary points and took only 1096 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

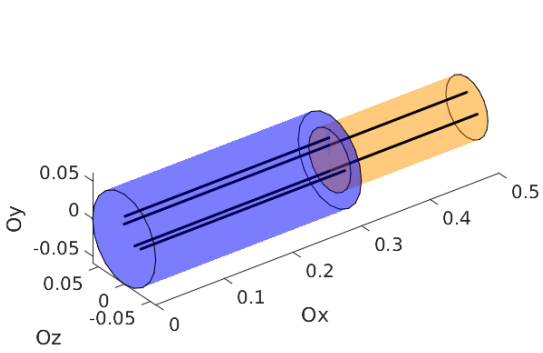
Figure 5.10: Scenario 4 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.5 Scenario 5: spatial 2-segments 4-tendons soft robot

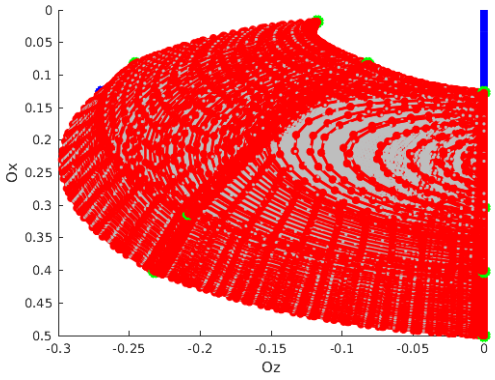
In this scenario, we consider a soft robot composed of two segments and actuated by four tendons (See Fig. 5.11a). The two lateral tendons are installed as in the first scenario. The two upward tendons are installed on the Oxy plan, and parallel to each other. The first tendon is fixed at the position $(0, r_1/2, 0)$ and extends along the first segment length L_1. The second tendon is fixed at the position $(0, r_2/2, 0)$ and extends to the end-effector position. We choose the bounded tension magnitude $\mathcal{T}_{1,2,3,4} = [0, 150]N$ enabling us to apply a couple magnitude $\mathcal{C}_{\mathcal{T}_{1,3}} = [0, 4.5]$, and $\mathcal{C}_{\mathcal{T}_{2,4}} = [0, 3]N \cdot m$.

Interior and exterior workspace boundaries of this particular structure are mapped starting from an initial limit point. Next, using (5.40) we compute all possible tangent vectors for each plan Oxy, Oyz and Oxz. Based on this, we calculate a prediction corresponding to each tangent vector and we obtain each associated solution (5.42). The obtained result is depicted by red curves in a three-view drawing (Oxz, Oyx, and Oyz plans, in Figs. 5.11b, 5.11c, and 5.11d respectively).

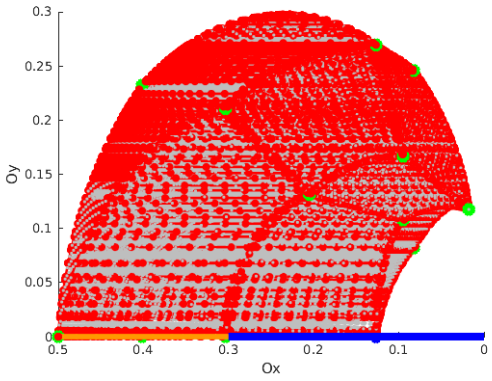
For this scenario, the continuation approach computed a total of 688 boundary points and took only 107 seconds to map the workspace boundaries for this scenario.



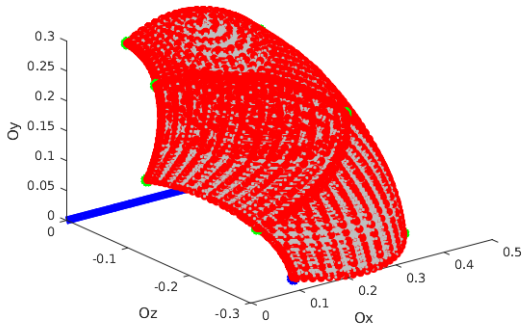
(a) Soft robot structure.



(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.



(c) Oyx plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.



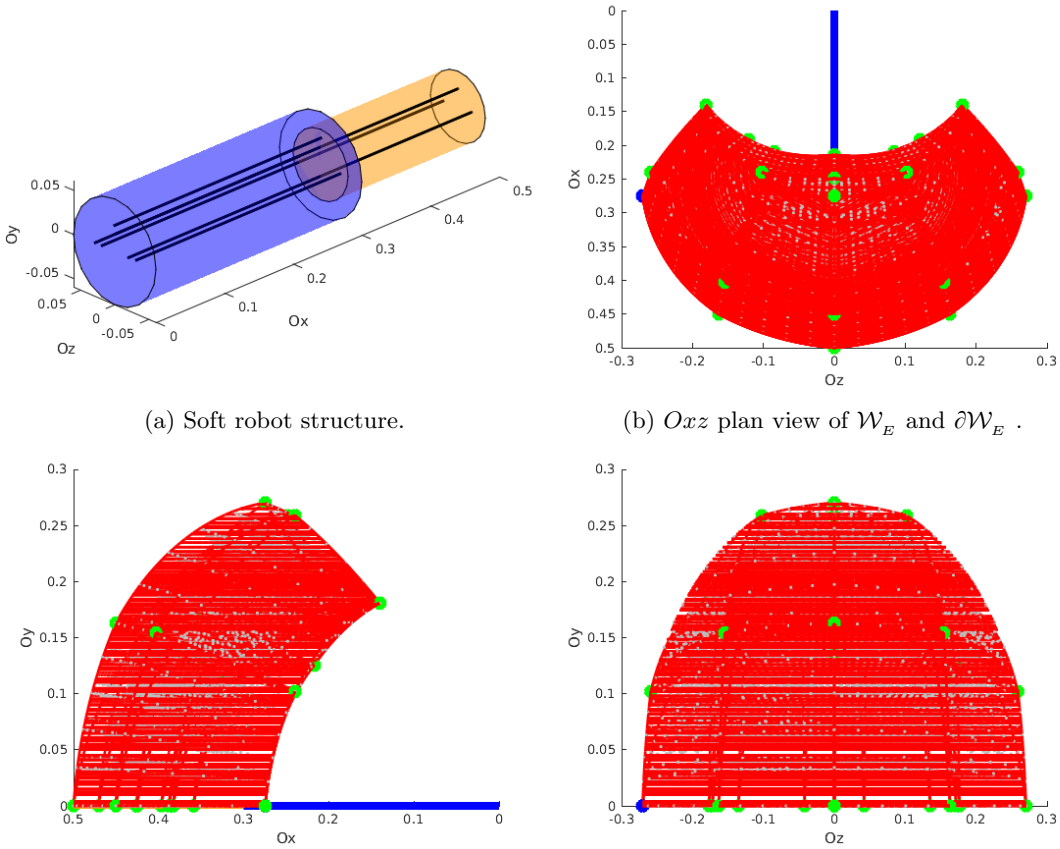
(d) Oyz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

Figure 5.11: Scenario 5 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.6 Scenario 6: spatial 2-segments 6-tendons soft robot

In this scenario, we consider a soft robot composed of two segments and actuated by six tendons (See Fig. 5.12a). There are four lateral symmetric tendons (as in Fig. 5.8a), and two upward tendons as in the above scenario. We choose the bounded tension magnitude $\mathcal{T}_{_{1,\cdots,6}} = [0, 100]N$ enabling us to apply a couple magnitude $\mathcal{C}_{_{\mathcal{T}_{1,2,3}}} = [0, 3]$, and $\mathcal{C}_{_{\mathcal{T}_{4,5,6}}} = [0, 2]N \cdot m$.

The obtained result is depicted by red curves in a three-view drawing (Oxz, Oyx, and Oyz plans, in Figs. 5.12b, 5.12c, and 5.12d respectively). For this scenario, the continuation approach computed a total of 4107 boundary points and took only 980 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_{E}$ and $\partial\mathcal{W}_{E}$.

(c) Oyx plan view of $\mathcal{W}_{E}$ and $\partial\mathcal{W}_{E}$.

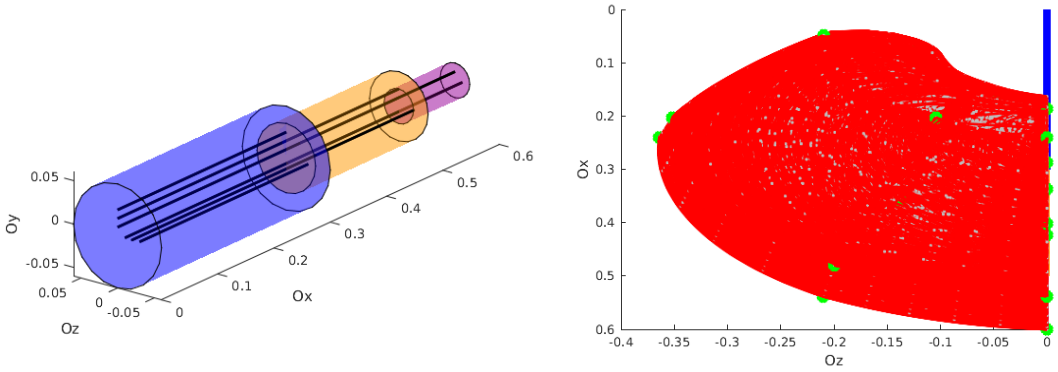
(d) Oyz plan view of $\mathcal{W}_{E}$ and $\partial\mathcal{W}_{E}$.

Figure 5.12: Scenario 6 - $\mathcal{W}_{E}$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_{E}$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.7 Scenario 7: spatial 3-segments 6-tendons soft robot

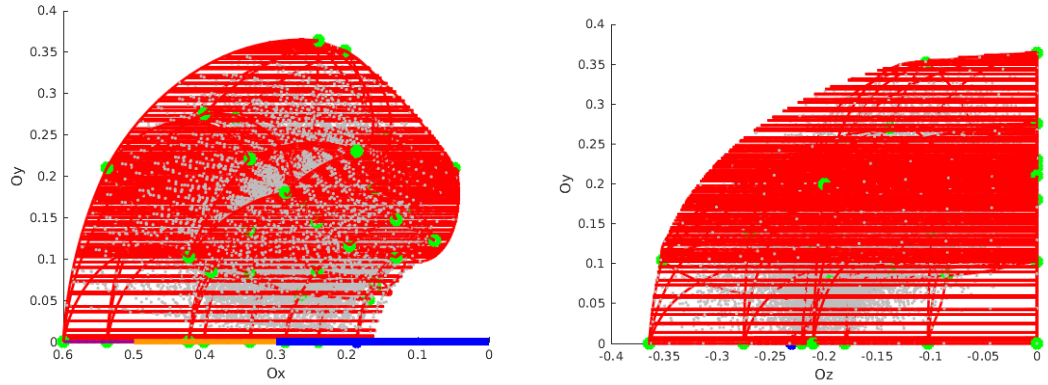
In this scenario, we consider a soft robot composed of three segments and actuated by four tendons (See Fig. 5.13a). The three lateral tendons are installed as in the third scenario (see Fig. 5.9a). The three upward tendons are installed on the Oxy plan, and are parallel to each other. The first tendon is fixed at the position $(0, r_1/2, 0)$ and extends along the first segment length L_1. The second tendon is fixed at the position $(0, r_2/2, 0)$ and extends to $(L_1 + L_2, r_2/2, 0)$, and the third tendon is fixed at the position $(0, r_3/2, 0)$ and extends all along to the end-effector position. We choose a tension magnitude within $\mathcal{T}_{_{1,\cdots,6}} = [0, 100]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{_{\mathcal{T}_{1,4}}} = [0, 3]N \cdot m$, $\mathcal{C}_{_{\mathcal{T}_{2,5}}} = [0, 2]N \cdot m$, and $\mathcal{C}_{_{\mathcal{T}_{3,6}}} = [0, 1]N \cdot m$.

The obtained result is depicted by red curves in a three-view drawing (Oxz, Oyx, and Oyz plans, in Figs. 5.13b, 5.13c, and 5.13d respectively). For this scenario, the continuation approach computed a total of 4122 boundary points and took only 1392 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.



(c) Oyx plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

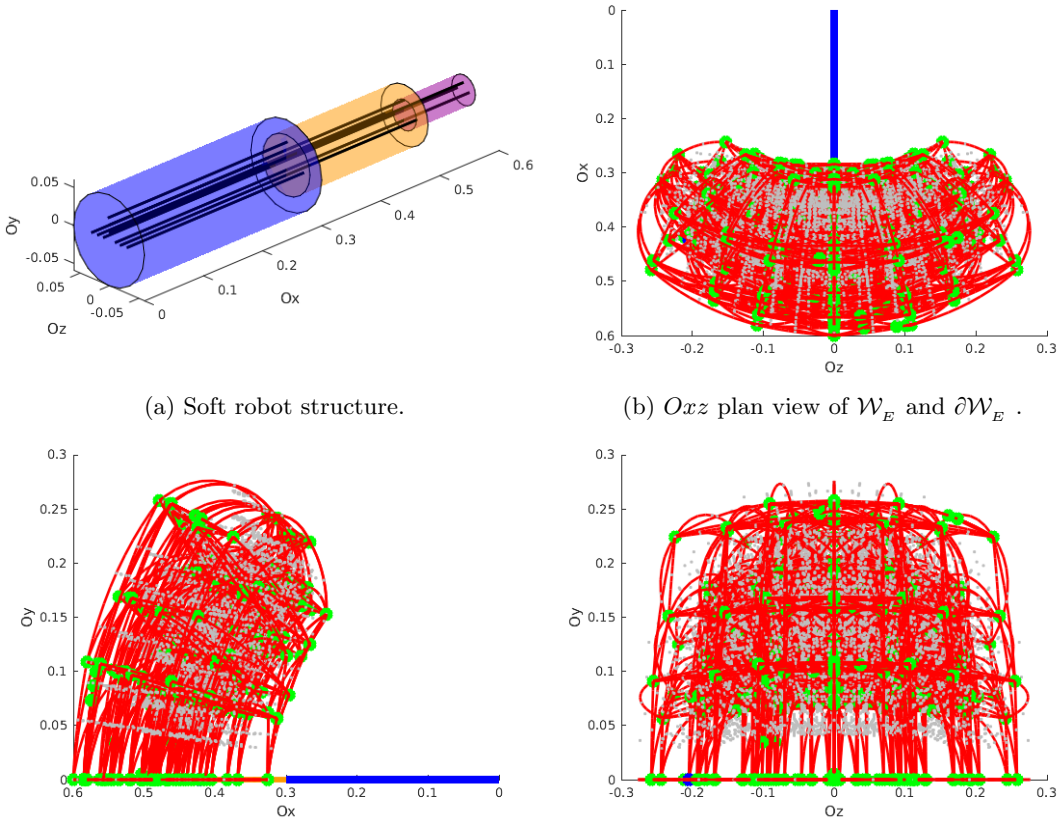
(d) Oyz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

Figure 5.13: Scenario 7 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.8 Scenario 8: spatial 3-segments 9-tendons soft robot

In the final scenario (See Fig. 4.12a), we consider the same tendons routing of the last scenario in Section 4.3.6.5 from Chapter 4. We choose a tension magnitude within $\mathcal{T}_{1,\dots,9} = [0, 50]N$ that will allow us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_{1,4,7}} = [0, 1.5]N \cdot m$, $\mathcal{C}_{\mathcal{T}_{2,5,8}} = [0, 1]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_{3,6,9}} = [0, 0.5]N \cdot m$.

The obtained result is depicted by red curves in a three-view drawing (Oxz, Oyx, and Oyz plans, in Figs. 5.14b, 5.14c, and 5.14d respectively). For this scenario, the continuation approach computed a total of 7621 boundary points and took only 2674 seconds to map the workspace boundaries for this scenario.



(a) Soft robot structure.

(b) Oxz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

(c) Oyx plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

(d) Oyz plan view of $\mathcal{W}_E$ and $\partial\mathcal{W}_E$.

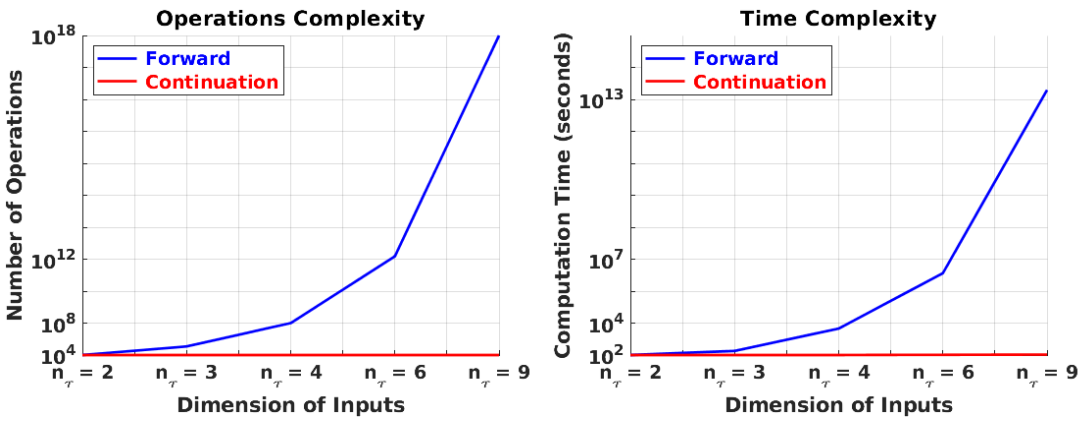
Figure 5.14: Scenario 8 - $\mathcal{W}_E$ estimation via forward approach (gray points) and workspace boundaries $\partial\mathcal{W}_E$ computed via the continuation approach (red curves and green bifurcation points).

5.3.0.9 Notes on computational efficiency

In Table 5.2, and Figures 5.15a and 5.15b, we summarize the operations' complexity and computational time of each scenario.

Table 5.2: Operations Complexity and computational time: forward vs continuation.

	Operations		Time (seconds)	
Approach / Inputs	Forward	Continuation	Forward	Continuation
$n_\tau = 2$ (Sc1)	200^2	87	66.35	9.54
$n_\tau = 3$ (Sc3)	100^3	327	2962.92	40.62
$n_\tau = 4$ (Sc2, Sc5)	100^4	$(687, 688)$	$\approx 1.65 \times 10^5$	$(96, 107)$
$n_\tau = 6$ (Sc4, Sc6, Sc7)	100^6	$(4126, 4107, 4122)$	$\approx 1.65 \times 10^9$	$(1096, 980, 1392)$
$n_\tau = 9$ (Sc8)	100^9	7621	$\approx 1.65 \times 10^{12}$	2674

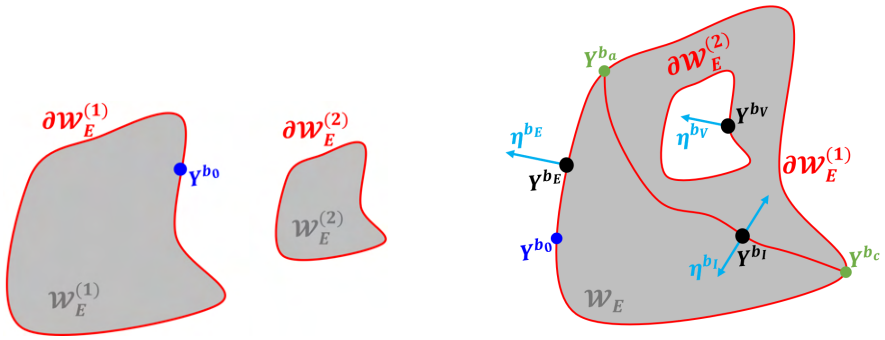


(a) Operations. (b) Time

Figure 5.15: Operations complexity and computational time of the forward and continuation approaches for the workspace and workspace boundaries estimation of the investigated scenarios using the PCS model.

5.3.1 Discussion on Multi-Component Workspaces, Voids/Holes, and Boundaries Characterization:

In this thesis, starting from one boundary point, the proposed continuation approach enables us to map the exterior and interior workspace boundaries whether such boundaries result from actuators saturation, or physical limitations such as the mechanism's length, self-collision, or elastic instabilities. However, as stated at the end of Section 5.2.4, isolated boundaries (see Fig. 5.16a) and voids (see Fig. 5.16b) might arise in complex soft robot configurations. In either case, both $\partial\mathcal{W}_E^{(1)}$ and $\partial\mathcal{W}_E^{(2)}$ verify the workspace boundary Definition 3. Suppose that in the first instance, the initial boundary point from which the continuation algorithm starts is on $\partial\mathcal{W}_E^{(1)}$ as shown in Fig. 5.16. In this case, using this point as a starting configuration of the continuation algorithm, we can surely map $\partial\mathcal{W}_E^{(1)}$ in the first process. Then, to map $\partial\mathcal{W}_E^{(2)}$, we need to verify if all limit points have been explored by the first process, and since we have a second workspace boundary, it is evident that there exists at least one limit configuration that was not explored by the former process. Then, using this configuration as a new initial point, we can run a second continuation algorithm process allowing us to map $\partial\mathcal{W}_E^{(2)}$.



(a) (b)

Figure 5.16: Multi-component workspaces (a); Holes/Voids (b).

Another important issue concerns the characterization of different types of boundaries: exterior boundaries, interior boundaries, and voids. Firstly, we would like to emphasize that the proposed continuation approach contains necessary information to calculate the normal vector of each point on the obtained boundaries. In fact, for any boundary point $Y^b = [Q^b, \gamma^b] \in \partial\mathcal{W}_E$, its normal vector can be calculated as [55]: $\eta^b \equiv \eta(Y^b) = \Psi_u^T(Q^b)\gamma^b$. To explain the main idea, we take Fig. 5.16 as an example. Consider, as depicted by Fig. 5.16b, a boundary point Y^{b_E} lying on the exterior boundary $\partial\mathcal{W}_E^{(1)}$, with its normal vector η^{b_E}, therefore in the normal direction we can find a point (noted as u^{p_E}) which will not belong to the workspace. In summary, this judgment can be formulated as the following condition:

$$\exists u^{p_E} = u^{b_E} + \varepsilon\eta^{b_E}, \text{ s.t. } u^{p_E} \notin \mathcal{W}_E$$

where $\varepsilon \in \mathbb{R}$ is a pre-defined small value. Conversely, for a boundary point Y^{b_I} lying on the interior curve, with its normal vector η^{b_I} as shown by Fig. 5.16b, an interior boundary may be sorted by verifying:

$$\forall u^{p_I} = u^{b_I} + \varepsilon\eta^{b_I}, \text{ s.t. } u^{p_I} \in \mathcal{W}_E$$

Finally, for Y^{b_V} lying on $\partial\mathcal{W}_E^{(2)}$, with its normal vector η^{b_V} as illustrated by Fig. 5.16b, a void may be distinguished by verifying the following condition:

$$\exists u^{p_V} = u^{b_V} + \varepsilon\eta^{b_V}, \text{ s.t. } u^{p_V} \notin \mathcal{W}_E$$

Note that the above condition is identical to that of the exterior boundary. However, the fact that $\partial\mathcal{W}_E^{(2)}$ is enveloped by $\partial\mathcal{W}_E^{(1)}$ enables us to assert that $\partial\mathcal{W}_E^{(2)}$ is the void and $\partial\mathcal{W}_E^{(1)}$ is the exterior boundary.

5.4 Conclusion

The continuation approach allows the mapping of interior and exterior boundaries of soft robots with slender shapes using the PCS model.

However, it is important to state that this approach cannot provide information on all points in the workspace, but only on boundary points. Moreover, this approach was only applied to slender-shaped soft robots via the PCS model since this modeling method provides the analytical form of all necessary parameters.

Thus, it is of interest to extend the applicability of this approach in order to also treat soft robots with general shape which can be modeled by FEM. However, since the FEM model is established differently from the PCS approach, some steps of the proposed continuation approach will need to be adapted in order to consider the FEM case (which will be outlined in Chapter 7).

Chapter 6

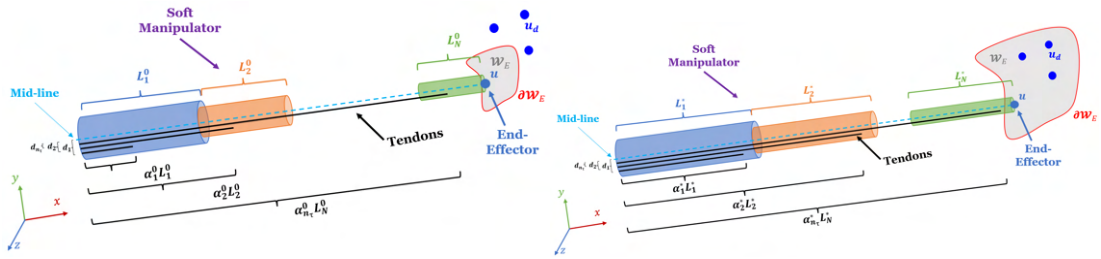
Design Optimization: PCS case

6.1 Introduction

In Chapter 3, Chapter 4, and Chapter 5, we have proposed different approaches in order to answer the direct problem of soft robots, i.e., to estimate the workspace of a given designed configuration of a soft robot. On the other hand, this chapter treats the inverse problem, i.e., to optimize the design of a soft robot in order to guarantee that certain points belong to its workspace. Solving such a problem is useful for robotic applications where we don't have an idea about the exact geometric parameters of the soft robot, but we do know what task is the soft robot trying to achieve, such as Pick-and-Place, and trajectory planning.

The actual process of a soft robot's design still follows intuition and the procedure of trial-and-error. Considering specific performance objectives, the classical design approach of soft robots is coupled with the uncertainty of achieving such objectives and also with the substantial economical expenditures necessary for the trial-and-error endeavors. For the purpose of accomplishing the desired objectives, it is therefore logical, for both economic and scientific reasons, to optimize the design of soft robots in a virtual environment before proceeding to its final physical conception.

This thesis follows the above thought and seeks to answer the following question: Given a soft robot (illustrated by Fig. 6.1a), composed of a finite number of segments with bounded length $L \in \mathcal{L}$, and driven by mounted actuators (i.e., tendons) with bounded actuators length (i.e., tendons' length) $\alpha \in \mathcal{A}$, and bounded actuators magnitude (i.e., tendons' tension) $\tau \in \mathcal{T}$. How should we optimize the design of a soft robot in order to reach the desired position (e.g., blue-colored points u_d in Fig. 6.1a).



(a) Initial design. (b) Optimal Design.

Figure 6.1: Soft robot's design.

In other words, we seek to determine the optimal length parameter L of each segment, and the optimal actuators length parameter α such that the objective points (e.g., blue-colored points u_d in Figs. 6.1a) must be located inside the workspace of the optimally designed soft robot, as depicted by Fig. 6.1b.

In the present work, we assume that the segments radius r and the distance of the actuators to the mid-line d are both pre-determined and fixed parameters. However, the proposed method can be generalized to treat the case where r and d are varying (which will be explained in Chapter 7).

The reachable workspace of a soft robot has been defined in Definition 1 for the case that L and α are constant parameters. However, since we now seek to achieve the optimal values of L and α, the definition of the reachable workspace of a soft robot needs to be adapted, which should be a function of L and α.

Definition 4. *The workspace $\mathcal{W}_E$ of the end-effector u of a soft robot modeled by $\Psi(\tau, q, L, \alpha) = 0$ (with $q \in \mathbb{R}^n$ being the generalized strain vector), composed of bounded length of segments $L \in \mathcal{L}$, and driven via bounded actuators $\tau \in \mathcal{T}$, with bounded actuators length $\alpha \in \mathcal{A}$, is a subspace of $\mathbb{R}^3$, defined by the following:*

$$\mathcal{W}_E(L, \alpha) = \{u = \psi(q, L, \alpha) \mid \exists (q \in \mathbb{R}^n, \tau \in \mathcal{T}),\ s.t.\ \Psi(\tau, q, L, \alpha) = 0\}$$

where ψ represents the geometric model of the soft robot which depends not only on q, but also on the parameters to be optimized L and α (more details will be discussed in 6.2.1), and $\Psi(\tau, q, L, \alpha) = K(L, \alpha)\, q - H(q, L, \alpha)\tau - F(q, L, \alpha)$.

To accomplish the vision depicted by Fig 6.1b, we propose an optimization approach, which will be implemented on the PCS model and explained in the following.

6.2 Implementation of the Methodology

6.2.1 Influence of the Parameters to be Optimized on the PCS Static Model

Before we step into the details of the proposed optimization approach, we need to clarify the influence of the parameters to be optimized on the adopted mathematical model of soft robots. In this work, we seek to optimize two parameters; the length L_i of each segment $i \in 1 \cdots N$, and the actuators location, particularly the length parameter α_k of each actuator $k \in 1 \cdots n_\tau$.

To be consistent with Definition 4, the workspace of the studied soft robot is now governed by:

$$K(L,\alpha)\ q = H(q,L,\alpha)\tau + F\left(q,L,\alpha\right) \tag{6.1a}$$

$$u(q,L,\alpha) = \mathfrak{E}_1\ g(q,L,\alpha)\ \mathfrak{E}_2^T \tag{6.1b}$$

with

$$g(q,L,\alpha) = e^{\alpha_1 L_1 \widehat{\xi}_1} e^{(1-\alpha_1)L_1\widehat{\xi}_1} \cdots e^{\alpha_{n_\tau} L_N \widehat{\xi}_N} e^{(1-\alpha_{n_\tau})L_N\widehat{\xi}_N} \tag{6.2a}$$

$$\widehat{\xi}_i = \widehat{B_{q_i} q_i}\ ;\ \ i \in 1\cdots N \tag{6.2b}$$

The block-diagonal generalized stiffness matrix K, the generalized actuation matrix H, and the vector of generalized position-dependent external forces F are now established as:

$$K(L,\alpha) = \mathrm{diag}\left(B_{q_1}^T\left[\int_{\mathcal{X}_1^1}\Sigma_1 dX, \int_{\mathcal{X}_1^2}\Sigma_1 dX\right]B_{q_1}, \cdots, B_{q_N}^T\left[\int_{\mathcal{X}_N^1}\Sigma_N dX, \int_{\mathcal{X}_N^2}\Sigma_N dX\right]B_{q_N}\right)\ , \tag{6.3a}$$

$$H\left(q,L,\alpha\right) = \left[\left(B_{q_1}^T\left[\int_{\mathcal{X}_1^1} H_\tau dX, \int_{\mathcal{X}_1^2} H_\tau dX\right]\right)^T, \cdots, \left(B_{q_N}^T\left[\int_{\mathcal{X}_N^1} H_\tau dX, \int_{\mathcal{X}_N^2} H_\tau dX\right]\right)^T\right]^T \tag{6.3b}$$

$$F\left(q,L,\alpha\right) = \sum_{i=1}^{N}\left[\int_{\mathcal{X}_i^1} J_i^T\bar{\mathcal{F}}_{e_i} dX, \int_{\mathcal{X}_i^2} J_i^T\bar{\mathcal{F}}_{e_i} dX\right] \tag{6.3c}$$

where $\mathcal{X}_i^1 = [0, \alpha_k L_i]$ and $\mathcal{X}_i^2 = [\alpha_k L_i, (1-\alpha_k)L_i]$, for $i \in 1\cdots N$, and $k \in 1\cdots n_\tau$.

The configuration matrix g_i of a body i and the strain twist $\widehat{\xi}_i$ of a body i can now be defined as follows:

$$g_i(X,L,\alpha) = \begin{cases} g_i^{s_1} = e^{\alpha_k L_i \widehat{\xi}_i}, & X \in \mathcal{X}_i^1 \\ g_i^{s_2} = e^{(1-\alpha_k)L_i\widehat{\xi}_i}, & X \in \mathcal{X}_i^2 \end{cases} \tag{6.4a}$$

$$\widehat{\xi}_i(X,L,\alpha) = g_i^{-1}\ g_i' \tag{6.4b}$$

The geometric Jacobian is now formulated as follows:

$$J_i(q,L,\alpha,X) = \sum_{l=0}^{i} \mathrm{Ad}_{g^{(l)}\cdots g_i}^{-1}\mathrm{T}_{g^{(l)}} B_{q^{(l)}} \tag{6.5}$$

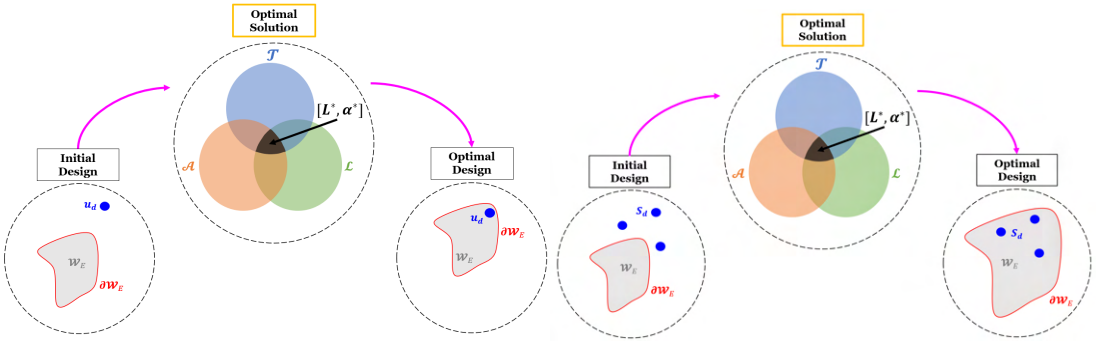
Remark 8. *It is important to mention that, in addition to optimizing the segment length parameter L and the actuators length parameter α, it is possible to optimize the parameter representing the distance of the actuator to the mid-line d and also the segment radius parameter r by following the same procedure that will be explained for the parameters L and α.*

6.2.2 Optimization-based Method

In this section, we present an optimization approach for the purpose of optimizing the design of the investigated slender soft robot in order to reach a single specific point and multiple target points, as depicted in Figs. 6.2a and 6.2b.

To achieve one single point (as shown in Fig. 6.2a), the procedure consists of finding the optimal parameters L and α in order that the soft robot design is optimized enough to the level that its workspace will contain this point.

However, when trying to reach multiple target points (as shown in Figs. 6.2b), the parameters must be optimized in order that the workspace of the designed soft robot contains all the desired points, which is clearly more difficult since the cost function will be more complex than the first case, because these objective points may be competing with each other.



(a) Optimal Design for a single point. (b) Optimal Design for multiple points.

Figure 6.2: Soft robot's design to achieve a single point (a), and multiple points (b).

6.2.3 Optimal Design to Achieve One Single Point

The proposed method consists of determining the optimal values of both the length parameter L of each segment denoted as L^*, and the actuators length parameter α designated by α^*, in order that, given these optimal values, the end-effector's workspace of the optimally designed soft robot will contain the position to be accessed (as illustrated in Fig. 6.2a).

Denote u and u_d as the end-effector's position and the desired position to be reached, according to the workspace Definition 4, we can then define the distance between u and u_d as

$$f(q, L, \alpha) = \tfrac{1}{2}\|u(q, L, \alpha) - u_d\|_2^2$$

Hence, an optimal design of a soft robot in order to reach u_d can be achieved through solving the following nonlinear constrained optimization problem:

$$\begin{aligned} [L^*, \alpha^*, \tau^*] = \underset{\{L,\alpha,\tau\}}{\arg\min} \quad & f(q, L, \alpha) \\ \text{s.t.} \quad & \tau \in \mathcal{T}, L \in \mathcal{L}, \alpha \in \mathcal{A} \\ & K(L, \alpha)q = H(q, L, \alpha)\tau + F(q, L, \alpha) \\ & u(q, L, \alpha) = \mathfrak{E}_1 g(q, L, \alpha)\mathfrak{E}_2^T \end{aligned} \tag{6.6}$$

where τ^* is the optimal actuators vector to reach the desired position u_d. The vectors L^* and α^* represent the optimal length of segments and the optimal actuators placement, respectively (since they satisfy the bounded constraints and the nonlinear constraints) to minimize the nonlinear cost function $f(q, L, \alpha)$.

To solve the above optimization problem, gradient-based method [60] can be used which depends on the knowledge of the gradient of $f(q, L, \alpha)$ with respect to the vectors L, α, and τ, noted as $\nabla_{[L,\alpha,\tau]} f$.

The gradient method to solve (6.6) can be established by the following [60]:

$$\begin{aligned}
&\mathfrak{Y}_{s+1} = [L,\alpha,\tau]_s - \lambda \nabla_{[L,\alpha,\tau]} f\left([L,\alpha,\tau]_s\right) \\
&\text{s.t.} \quad [L,\alpha,\tau]_{s+1} = \underset{[L,\alpha,\tau]}{\arg\min} \|\mathfrak{Y}_{s+1} - [L,\alpha,\tau]\| \\
&\qquad \tau \in \mathcal{T}, L \in \mathcal{L}, \alpha \in \mathcal{A} \\
&\qquad K(L,\alpha)q = H(q,L,\alpha)\tau + F\left(q,L,\alpha\right) \\
&\qquad u(q,L,\alpha) = \mathfrak{E}_1 g(q,L,\alpha)\mathfrak{E}_2^T
\end{aligned} \tag{6.7}$$

with: $[L,\alpha,\tau]_0$ is known, and λ is an infinitesimal small step size.

Or in an alternative form; The gradient method to solve (6.6) can be established by the following [60]:

$$\begin{aligned}
\mathfrak{Y}_{s+1} &= [L,\alpha,\tau]_s - \lambda \nabla_{[L,\alpha,\tau]} f\left([L,\alpha,\tau]_s\right) \\
[L,\alpha,\tau]_{s+1} &= Pr(\mathfrak{Y}_{s+1}),\ \text{Project } \mathfrak{Y}_{s+1} \text{ onto the feasible set}
\end{aligned} \tag{6.8}$$

where Pr is the projection operator.

Within the PCS framework, the following subsections are devoted to establishing the analytical form:

$$\nabla_{[L,\alpha,\tau]} f = \Big[\nabla_L f, \nabla_\alpha f, \nabla_\tau f\Big] \tag{6.9}$$

6.2.3.1 Calculation of $\nabla_\tau f$

First, the analytical form of the gradient of the cost function with respect to the actuator vector $\nabla_\tau f$ has already been demonstrated and established in (3.7) and can be written in the following form:

$$\nabla_\tau f = \Big[\frac{\partial f}{\partial \tau}\Big]^T = \nabla_\tau u\ \left[u(q,L,\alpha) - u_d\right] \tag{6.10}$$

with $\nabla_\tau u$ represents the gradient of the end-effector with respect to τ, and is defined according to Theorem 1 by:

$$\nabla_\tau u = \nabla_\tau q\ \nabla_u q = \nabla_\tau q \left[[0_3\ R(q,L,\alpha)]J(q,L,\alpha)\right]^T \tag{6.11}$$

where $J(q,L,\alpha)$ is the geometric Jacobian defined in (6.5), the orientation matrix $R(q,L,\alpha)$ is computed via the definition of the configuration matrix $g(q,L,\alpha)$ through: $R(q,L,\alpha) = \mathfrak{E}_1\ g(q,L,\alpha)\ \mathfrak{E}_1^T$, and $\nabla_\tau q$ is the gradient of the generalized strain vector with respect to the actuators force vector and formulated according to Assumption 1 as:

$$\nabla_\tau q = H^T(q,L,\alpha)\big[\mathfrak{B}^{-1}(\tau,q,L,\alpha)\big]^T \tag{6.12}$$

with

$$\mathfrak{B}(\tau,q,L,\alpha) = K(L,\alpha) - H_q(q,L,\alpha)(I_n \otimes \tau) - F_q(q,L,\alpha)$$

6.2.3.2 Calculation of $\nabla_L f$

The analytical form of the gradient of the cost function with respect to the segments length parameter $\nabla_L f$ can be calculated using the exponential map properties, but first let us write the formula of $\nabla_L f$ using the principle of variable separation as:

$$\nabla_L f = \Big[\frac{\partial f}{\partial L}\Big]^T = \Big[\frac{\partial f}{\partial u}\ \frac{\partial u}{\partial L}\Big]^T \tag{6.13}$$

By definition, the partial derivative of the cost function with respect to the end-effector position vector is established by:

$$\frac{\partial f}{\partial u} = \left[u(q, L, \alpha) - u_d\right]^T \tag{6.14}$$

Next, the partial derivative of the end-effector position vector with respect to the segment length parameter can be decomposed using (6.1b) as follows:

$$\frac{\partial u}{\partial L} = \mathfrak{E}_{1} \frac{\partial g}{\partial L} \mathfrak{E}_{N} + \frac{\partial u}{\partial q}\frac{\partial q}{\partial L}$$

with $\mathfrak{E}_{N} = I_N \otimes \mathfrak{E}_{2}^T$, which can be written as:

$$\frac{\partial u}{\partial L} = \mathfrak{E}_{1} \frac{\partial g}{\partial L} \mathfrak{E}_{N} + \left[\nabla_u q\right]^T \frac{\partial q}{\partial L} \tag{6.15}$$

with

$$\nabla_u q = \Big[\frac{\partial u}{\partial q}\Big]^T = \Big[\mathfrak{E}_{1} \frac{\partial g}{\partial q} \mathfrak{E}_{n}\Big]^T = \Big[[0_3\ R(q, L, \alpha)]J(q, L, \alpha)\Big]^T$$

which is deduced from (6.11).

Therefore, to formulate $\dfrac{\partial u}{\partial L}$ more explicitly, we need to find the analytical form of $\dfrac{\partial g}{\partial L}$ and $\dfrac{\partial q}{\partial L}$. For this, let us first calculate $\dfrac{\partial q}{\partial L}$. Applying the partial derivative with respect to L to (6.1a), we obtain:

$$\frac{\partial K}{\partial L}(I_N \otimes q) + K\frac{\partial q}{\partial L} = \frac{\partial H}{\partial L}\tau + \frac{\partial F}{\partial L}$$

with K being always invertible, the above gives:

$$\frac{\partial q}{\partial L} = K^{-1}\left(\frac{\partial H}{\partial L}\tau + \frac{\partial F}{\partial L} - \frac{\partial K}{\partial L}(I_N \otimes q)\right) \tag{6.16}$$

where the value of q corresponding to the value of τ can be calculated by solving (6.1a) using Proposition 1.

Next, we calculate $\dfrac{\partial g}{\partial L}$ using the exponential map properties. First, we use (6.4a) in order to reformulate the exponential map $g(q, L, \alpha)$ from (6.2a) in the following form:

$$g(q, L, \alpha) = (g_1^{s_1} g_1^{s_2}) \cdots (g_N^{s_1} g_N^{s_2}) \tag{6.17}$$

with $g_i^{s_1}$ and $g_i^{s_2}$ being defined in (6.4a). Using the above exponential map formula (6.17), $\dfrac{\partial g}{\partial L}$ can be established as follows [66]:

$$\begin{aligned}\frac{\partial g}{\partial L} = {} & \frac{\partial g_1^{s_1}}{\partial L}\left[I_N \otimes (g_1^{s_2} \cdots g_N^{s_1} g_N^{s_2})\right] + g_1^{s_1}\frac{\partial g_1^{s_2}}{\partial L}\left[I_N \otimes (g_2^{s_1} \cdots g_N^{s_1} g_N^{s_2})\right] + \cdots \\ & + g_1^{s_1} g_1^{s_2} \cdots \frac{\partial g_N^{s_1}}{\partial L}\left[I_N \otimes g_N^{s_2}\right] + g_1^{s_1} g_1^{s_2} \cdots g_N^{s_1}\frac{\partial g_N^{s_2}}{\partial L}\end{aligned} \tag{6.18}$$

where

$$\frac{\partial g_i^{s_1}}{\partial L} = \left[0_4, \cdots, \frac{\partial g_i^{s_1}}{\partial L_i}, \cdots, 0_4\right] = \left[0_4, \cdots, \alpha_k g_i \widehat{\xi}_i, \cdots, 0_4\right]$$

and

$$\frac{\partial g_i^{s_2}}{\partial L} = \left[0_4, \cdots, \frac{\partial g_i^{s_2}}{\partial L_i}, \cdots, 0_4\right] = \left[0_4, \cdots, (1-\alpha_k) g_i \widehat{\xi}_i, \cdots, 0_4\right]$$

With $\dfrac{\partial g}{\partial L}$ being calculated using (6.18), and $\dfrac{\partial q}{\partial L}$ from (6.16), we implement them into (6.15) to obtain the analytical form of $\dfrac{\partial u}{\partial L}$, and using $\dfrac{\partial f}{\partial u}$ from (6.14), we finally deduce $\nabla_L f$ in (6.13).

6.2.3.3 Calculation of $\nabla_\alpha f$

Last, with $\nabla_\tau f$ from (6.10), and $\nabla_L f$ being calculated from (6.13), the remaining gradient to be computed in order to complete (6.9) is $\nabla_\alpha f$, which will be treated hereafter.

The analytical form of the gradient of the cost function with respect to the actuators length parameter $\nabla_\alpha f$ can be calculated by following the same steps as those for $\nabla_L f$:

$$\nabla_\alpha f = \left[\frac{\partial f}{\partial \alpha}\right]^T = \left[\frac{\partial f}{\partial u}\ \frac{\partial u}{\partial \alpha}\right]^T \tag{6.19}$$

with $\dfrac{\partial f}{\partial u}$ being calculated from (6.14), thus, we only need to calculate the remaining term $\dfrac{\partial u}{\partial \alpha}$ which can be decomposed using (6.1b) as follows:

$$\frac{\partial u}{\partial \alpha} = \mathfrak{E}_1\ \frac{\partial g}{\partial \alpha}\ \mathfrak{E}_{n_\tau} + [\nabla_u q]^T \frac{\partial q}{\partial \alpha} \tag{6.20}$$

with $\mathfrak{E}_{n_\tau} = I_{n_\tau} \otimes \mathfrak{E}_2^T$. Then by applying the partial derivative with respect to α to (6.1a), we obtain:

$$\frac{\partial q}{\partial \alpha} = K^{-1}\left(\frac{\partial H}{\partial \alpha}\tau + \frac{\partial F}{\partial \alpha} - \frac{\partial K}{\partial \alpha}(I_{n_\tau} \otimes q)\right) \tag{6.21}$$

In the above equation, to calculate the partial derivative $\dfrac{\partial g}{\partial \alpha}$, we can use the properties of the exponential map [66] from (6.17), which yields:

$$\begin{aligned}\frac{\partial g}{\partial \alpha} = {} & \frac{\partial g_1^{s_1}}{\partial \alpha}\left[I_{n_\tau} \otimes (g_1^{s_2} \cdots g_N^{s_1} g_N^{s_2})\right] + g_1^{s_1}\frac{\partial g_1^{s_2}}{\partial \alpha}\left[I_{n_\tau} \otimes (g_2^{s_1} \cdots g_N^{s_1} g_N^{s_2})\right] + \cdots \\ & + g_1^{s_1} g_1^{s_2} \cdots \frac{\partial g_N^{s_1}}{\partial \alpha}\left[I_{n_\tau} \otimes g_N^{s_2}\right] + g_1^{s_1} g_1^{s_2} \cdots g_N^{s_1}\frac{\partial g_N^{s_2}}{\partial \alpha}\end{aligned} \tag{6.22}$$

where

$$\frac{\partial g_i^{s_1}}{\partial \alpha} = \left[0_4 \cdots \frac{\partial g_i^{s_1}}{\partial \alpha_k} \cdots 0_4\right] = \left[0_4 \cdots L_i g_i \widehat{\xi}_i \cdots 0_4\right]$$

and

$$\frac{\partial g_i^{s_2}}{\partial \alpha} = \left[0_4 \cdots \frac{\partial g_i^{s_2}}{\partial \alpha_k} \cdots 0_4\right] = \left[0_4 \cdots - L_i g_i \widehat{\xi}_i \cdots 0_4\right]$$

With $\dfrac{\partial g}{\partial \alpha}$ being calculated using (6.22), and $\dfrac{\partial q}{\partial \alpha}$ from (6.21), we implement them into (6.20) to obtain the analytical form of $\dfrac{\partial u}{\partial \alpha}$, and using $\dfrac{\partial f}{\partial u}$ from (6.14), we deduce $\nabla_\alpha f$ in (6.19).

Finally, with $\nabla_\tau f$ being calculated from (6.10), $\nabla_L f$ in (6.13), and $\nabla_\alpha f$ using (6.19), we deduce the value of $\nabla_{[L,\alpha,\tau]} f$ in (6.9), which is necessary for solving (6.6) via (6.7).

6.2.4 Optimal Design to Achieve Multiple Points

In a general sense, it is important to optimize the segments length parameter L and the actuators length parameter α so that the end-effector of the studied soft robot can reach multiple target points, which can be useful for various applications such as pick-and-place and trajectory planning. Equivalently, doing this will result in optimizing the end-effector's workspace in order to contain the target points (as depicted by Fig. 6.2b).

Given a set of target points, noted as $\mathcal{S}_d = \{u_d^{(1)}, \cdots, u_d^{(n_\mathcal{S})}\} \in \mathbb{R}^{3\times n_\mathcal{S}}$, for the obtained optimized soft robot with optimized structure, it will exist j different configurations $q^{(j)}$ with the corresponding end-effector position $u^{(j)}$ for $j = 1 \cdots n_\mathcal{S}$, such that $\|u^{(j)} - u_d^{(j)}\|_2^2 = 0$. Therefore, let us define the total distance of the end-effector's position with respect to all target points in $\mathcal{S}_d$ as:

$$h(q, L, \alpha) = \tfrac{1}{2} \sum_{j=1}^{n_\mathcal{S}} \|u^{(j)}(q^{(j)}, L, \alpha) - u_d^{(j)}\|_2^2$$

then the optimal design of the investigated soft robot for the purpose of reaching all target points in $\mathcal{S}_d$ can be obtained by solving the following nonlinear constrained optimization problem:

$$\begin{aligned} [L^*,\alpha^*, T^*] = &\arg\min_{\{L,\alpha,T\}} \quad h(q, L, \alpha) \\ \text{s.t.} \quad &\tau^{(j)} \in \mathcal{T}, L \in \mathcal{L}, \alpha \in \mathcal{A} \\ &K(L,\alpha)\ q^{(j)} = H(q^{(j)}, L, \alpha)\tau^{(j)} + F\left(q^{(j)}, L, \alpha\right) \\ &u^{(j)} = \mathfrak{E}_1\ g^{(j)}(q^{(j)}, L, \alpha)\ \mathfrak{E}_2^T \end{aligned} \tag{6.23}$$

with $q^{(j)} = \left[q_1^{(j)} \cdots q_N^{(j)}\right]^T \in \mathbb{R}^N$, $T^o = [\tau^{(1)} \cdots \tau^{(n_\mathcal{S})}]^o$ is the optimal actuators vector collection corresponding to each target point $u_d \in \mathcal{S}_d$, and $\tau^{(j)} = \left[\tau_1^{(j)} \cdots \tau_{n_\tau}^{(j)}\right]^T \in \mathbb{R}^{n_\tau}$, for $j \in 1, \cdots, n_\mathcal{S}$, is the actuators vector associated with the desired position u_{d_j}. The vectors L^* and α^* represent the optimal length of each segment of the robot and the optimal actuators length, respectively.

Similar to the case of a single point, the gradient-based method [60] can be used to solve the above optimization problem which needs the calculation of $h(q, L, \alpha)$ with respect to the vectors L, α, and τ, and formulated as:

$$\nabla_{[L,\alpha,T]} h = \left[\nabla_L h, \nabla_\alpha h, \nabla_T h\right] \tag{6.24}$$

which will be detailed hereafter.

6.2.4.1 Calculation of $\nabla_T h$

According to the principle of variable separation, we have:

$$\begin{aligned} \nabla_T h = \left[\frac{\partial h}{\partial T}\right]^T &= \left[\frac{\partial h}{\partial u}\frac{\partial u}{\partial T}\right]^T \\ &= \left[\frac{\partial h}{\partial u^{(1)}}\frac{\partial u^{(1)}}{\partial \tau^{(1)}}, \cdots, \frac{\partial h}{\partial u^{(n_\mathcal{S})}}\frac{\partial u^{(n_\mathcal{S})}}{\partial \tau^{(n_\mathcal{S})}}\right]^T \end{aligned} \tag{6.25}$$

Then by definition, the partial derivative of h with respect to $u^{(j)}$ can be written as:

$$\frac{\partial h}{\partial u^{(j)}} = \left[u^{(j)}(q^{(j)}, L, \alpha) - u_{d_j}\right]^T \tag{6.26}$$

Moreover, the partial derivative of the end-effector position $u^{(j)}$ with respect to its associated actuators vector $\tau^{(j)}$ can be deduced from (6.11) and formulated by:

$$\begin{aligned}\frac{\partial u^{(j)}}{\partial \tau^{(j)}} &= \frac{\partial u^{(j)}}{\partial q^{(j)}}\frac{\partial q^{(j)}}{\partial \tau^{(j)}} \\ &= \Big[[0_3\ R(q^{(j)}, L, \alpha)]J(q^{(j)}, L, \alpha)\Big]\frac{\partial q^{(j)}}{\partial \tau^{(j)}}\end{aligned} \tag{6.27}$$

with $\dfrac{\partial q^{(j)}}{\partial \tau^{(j)}}$ deduced from (2.50) as:

$$\frac{\partial q^{(j)}}{\partial \tau^{(j)}} = [\mathfrak{B}^{(j)}(\tau^{(j)}, q^{(j)}, L, \alpha)]^{-1} H^{(j)}(q^{(j)}, L, \alpha) \tag{6.28}$$

where

$$\mathfrak{B}^{(j)}(\tau^{(j)}, q^{(j)}, L, \alpha) = K(L, \alpha) - H_{q^{(j)}}(q^{(j)}, L, \alpha)(I_n \otimes \tau^{(j)}) - F_{q^{(j)}}(q^{(j)})$$

By substituting $\dfrac{\partial q^{(j)}}{\partial \tau^{(j)}}$ into (6.27), then (6.26) and (6.27) into (6.25), we finally obtain the analytical form of $\nabla_T h$.

6.2.4.2 Calculation of $\nabla_T L$

Similarly, using the principle of variable separation gives us the following formula:

$$\nabla_L h = \Big[\frac{\partial h}{\partial L}\Big]^T = \Big[\frac{\partial h}{\partial u}\frac{\partial u}{\partial L}\Big]^T = \sum_{j=1}^{j=n_{\mathcal{S}}} \left[\frac{\partial h}{\partial u^{(j)}}\frac{\partial u^{(j)}}{\partial L}\right]^T \tag{6.29}$$

where the partial derivative of h with respect to $u^{(j)}$ has already been established in (6.26). Therefore, we need only to find the analytical form of $\dfrac{\partial u^{(j)}}{\partial L}$. For this, using (6.15), we can write the following:

$$\frac{\partial u^{(j)}}{\partial L} = \mathfrak{E}_1\ \frac{\partial g^{(j)}}{\partial L}\ \mathfrak{E}_N + [\nabla_u q^{(j)}]^T \frac{\partial q^{(j)}}{\partial L} \tag{6.30}$$

with $\nabla_u q^{(j)} = \Big[[0_3\ R(q^{(j)}, L, \alpha)]J(q^{(j)}, L, \alpha)\Big]^T$ being deduced from (6.27). Moreover, using (6.16), we can get the following:

$$\frac{\partial q^{(j)}}{\partial L} = K^{-1}\left(\frac{\partial H^{(j)}}{\partial L}\tau^{(j)} + \frac{\partial F^{(j)}}{\partial L} - \frac{\partial K}{\partial L}(I_N \otimes q^{(j)})\right) \tag{6.31}$$

where the value of $q^{(j)}$ corresponding to the value of $\tau^{(j)}$ can be calculated by solving (6.1a) using Proposition 1.

Next, we need to find the analytical form of $\dfrac{\partial g}{\partial L}$ using the exponential map properties. For this, we reformulate the exponential map $g^{(j)}(q^{(j)}, L, \alpha)$ from (6.2a) as follows:

$$g^{(j)}(q^{(j)}, L, \alpha) = \Big(g_{1,s_1}^{(j)} g_{1,s_2}^{(j)}\Big) \cdots \Big(g_{N,s_1}^{(j)} g_{N,s_2}^{(j)}\Big) \tag{6.32}$$

with $g_{i,s_1}^{(j)} = e^{\alpha_k L_i \widehat{\xi}_i^{(j)}}$, and $g_{i,s_2}^{(j)} = e^{(1-\alpha_k) L_i \widehat{\xi}_i^{(j)}}$, for $i = 1 \cdots N$, $k = 1 \cdots n_\tau$, and $j = 1 \cdots n_{\mathcal{S}}$.

Using the above exponential map formula (6.32), $\dfrac{\partial g^{(j)}}{\partial L}$ can be calculated as follows [66]:

$$\begin{aligned}\frac{\partial g^{(j)}}{\partial L} &= \frac{\partial g^{(j)}_{1,s_1}}{\partial L}\left[I_N \otimes \left(g^{(j)}_{1,s_2}\cdots g^{(j)}_{N,s_1} g^{(j)}_{N,s_2}\right)\right] + g^{(j)}_{1,s_1}\frac{\partial g^{(j)}_{1,s_2}}{\partial L}\left[I_N \otimes \left(g^{(j)}_{2,s_1}\cdots g^{(j)}_{N,s_1} g^{(j)}_{N,s_2}\right)\right] + \cdots \\ &+ g^{(j)}_{1,s_1} g^{(j)}_{1,s_2}\cdots\frac{\partial g^{(j)}_{N,s_1}}{\partial L}\left[I_N \otimes g^{(j)}_{N,s_2}\right] + g^{(j)}_{1,s_1} g^{(j)}_{1,s_2}\cdots g^{(j)}_{N,s_1}\frac{\partial g^{(j)}_{N,s_2}}{\partial L}\end{aligned} \tag{6.33}$$

where

$$\frac{\partial g^{(j)}_{i,s_1}}{\partial L} = \left[0_4 \cdots \alpha_k g^{(j)}_i \widehat{\xi}^{(j)}_i \cdots 0_4\right]$$

and

$$\frac{\partial g^{(j)}_{i,s_2}}{\partial L} = \left[0_4 \cdots (1-\alpha_k) g^{(j)}_i \widehat{\xi}^{(j)}_i \cdots 0_4\right]$$

With $\dfrac{\partial g^{(j)}}{\partial L}$ being calculated using (6.33), and $\dfrac{\partial q^{(j)}}{\partial L}$ from (6.31), we implement both of them into (6.30) to obtain $\dfrac{\partial u^{(j)}}{\partial L}$, then, using (6.26), we finally deduce $\nabla_L h$ in (6.29).

6.2.4.3 Calculation of $\nabla_T \alpha$

As before, $\nabla_\alpha h$ can be formulated as:

$$\nabla_\alpha h = \left[\frac{\partial h}{\partial \alpha}\right]^T = \left[\frac{\partial h}{\partial u}\frac{\partial u}{\partial \alpha}\right]^T = \sum_{j=1}^{j=n_{\mathcal{S}}}\left[\frac{\partial h}{\partial u^{(j)}}\frac{\partial u^{(j)}}{\partial \alpha}\right]^T \tag{6.34}$$

with $\dfrac{\partial h}{\partial u}$ being calculated from (6.26), thus, we only need to find the analytical form of $\dfrac{\partial u}{\partial \alpha}$. For this, using (6.20), we can write the following:

$$\frac{\partial u^{(j)}}{\partial \alpha} = \mathfrak{E}_1 \frac{\partial g^{(j)}}{\partial \alpha}\mathfrak{E}_{n_\tau} + \left[\nabla_u q^{(j)}\right]^T \frac{\partial q^{(j)}}{\partial \alpha} \tag{6.35}$$

Moreover, using (6.21), we can then have:

$$\frac{\partial q^{(j)}}{\partial \alpha} = K^{-1}\left(\frac{\partial H^{(j)}}{\partial \alpha}\tau^{(j)} + \frac{\partial F^{(j)}}{\partial \alpha} - \frac{\partial K}{\partial \alpha}(I_{n_\tau} \otimes q^{(j)})\right) \tag{6.36}$$

Next, we have to find the analytical form of $\dfrac{\partial g}{\partial \alpha}$ using the properties of the exponential map. For this, using (6.32), $\dfrac{\partial g^{(j)}}{\partial \alpha}$ can be written as follows [66]:

$$\begin{aligned}\frac{\partial g^{(j)}}{\partial \alpha} &= \frac{\partial g^{(j)}_{1,s_1}}{\partial \alpha}\left[I_{n_\tau} \otimes \left(g^{(j)}_{1,s_2}\cdots g^{(j)}_{N,s_1} g^{(j)}_{N,s_2}\right)\right] + g^{(j)}_{1,s_1}\frac{\partial g^{(j)}_{1,s_2}}{\partial \alpha}\left[I_{n_\tau} \otimes \left(g^{(j)}_{2,s_1}\cdots g^{(j)}_{N,s_1} g^{(j)}_{N,s_2}\right)\right] + \cdots \\ &+ g^{(j)}_{1,s_1} g^{(j)}_{1,s_2}\cdots\frac{\partial g^{(j)}_{N,s_1}}{\partial \alpha}\left[I_{n_\tau} \otimes g^{(j)}_{N,s_2}\right] + g^{(j)}_{1,s_1} g^{(j)}_{1,s_2}\cdots g^{(j)}_{N,s_1}\frac{\partial g^{(j)}_{N,s_2}}{\partial \alpha}\end{aligned} \tag{6.37}$$

where

$$\frac{\partial g^{(j)}_{i,s_1}}{\partial \alpha} = \left[0_4 \cdots L_i g^{(j)}_i \widehat{\xi}^{(j)}_i \cdots 0_4\right]$$

and

$$\frac{\partial g_{i,s_2}^{(j)}}{\partial \alpha} = \left[0_4 \cdots - L_i g_i^{(j)} \widehat{\xi}_i^{(j)} \cdots 0_4\right]$$

With $\frac{\partial g^{(j)}}{\partial \alpha}$ being calculated using (6.37), and $\frac{\partial q^{(j)}}{\partial \alpha}$ from (6.36), we implement both of them into (6.35) to obtain $\frac{\partial u^{(j)}}{\partial \alpha}$, then, using (6.26), we finally deduce $\nabla_\alpha h$ in (6.34).

Finally, with $\nabla_T h$ being calculated from (6.25), $\nabla_L h$ in (6.29), and $\nabla_\alpha h$ using (6.34), we deduce the value of $\nabla_{[L,\alpha,T]} h$ in (6.24), which is necessary for solving (6.23).

6.3 Validation and Analysis

In this section, we illustrate the presented results by implementing the proposed optimization approach on tendon-driven soft robots (as depicted by Figs. 6.1a). Table 3.1 presents the numerical values of the investigated soft robot.

In the examined scenarios, $\mathcal{W}_{_E}$ is estimated using the forward approach (See Section 2.5.2) and illustrated by the gray area. The workspace boundary $\partial\mathcal{W}_{_E}$ is estimated using the continuation approach (See Section 5.2) and depicted by red curves and green bifurcation points. The mounted tendons are depicted by black lines, and the desired points to reach are blue-colored.

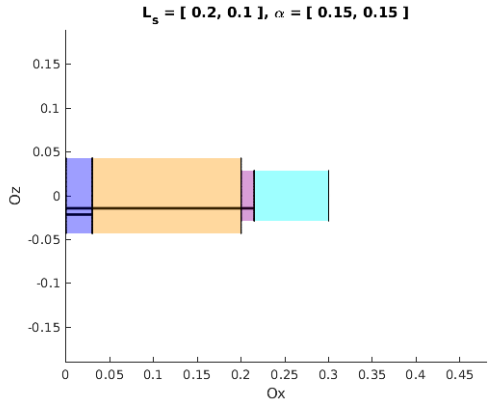
Also, the presented simulation scenarios were implemented using MATLAB (the function "fmincon" was used to solve the optimization problems formulated in this chapter) on an Intel Xeon(R) with a 16-GB RAM and a 3.50 GHz processor.

6.3.0.1 Scenario 1: 2-tendons driven planar soft robot

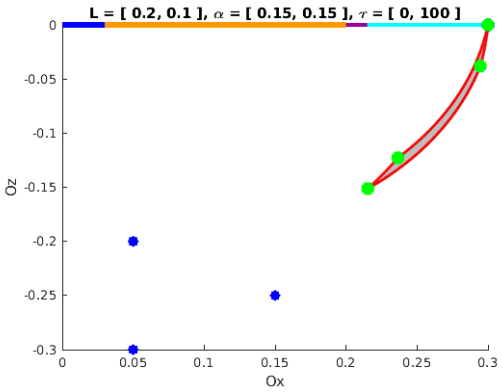
In the first scenario, we investigate a soft robot consisting of two segments and driven by two tendons installed on the Ozx plan (as shown in Fig. 6.3a).

In the initial state, the segments initial length vector is given $L^0 = [0.2,\ 0.1]^T m$, and the initial actuators length vector is $\alpha^0 = [0.15,\ 0.15]^T$. We select the bounded tension magnitude $\mathcal{T}_{1,2} = [0,\ 100]N$ allowing us to apply a couple magnitude within $\mathcal{C}_{\mathcal{T}_1} = [0,\ 2.14]N \cdot m$, and $\mathcal{C}_{\mathcal{T}_2} = [0,\ 1.42]N \cdot m$. The segments length bounds are $\mathcal{L} = [0,\ 0.3] \times [0,\ 0.2]m$, and the actuators length bounds are $\mathcal{A} = [0,\ 1]^2$.

The obtained results of $\mathcal{W}_{_E}$ and $\partial\mathcal{W}_{_E}$ for the initial state of the designed soft robot with the given initial values of the parameters L^0 and α^0 are depicted by Fig. 6.3b where we can clearly observe that the workspace does not contain the desired points.



(a) Initial design.

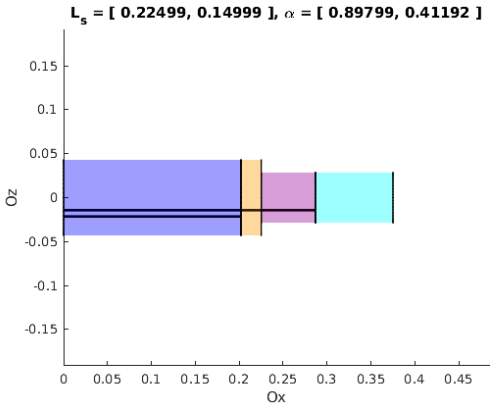


(b) $\mathcal{W}_E$ (gray) and $\partial\mathcal{W}_E$ (red).

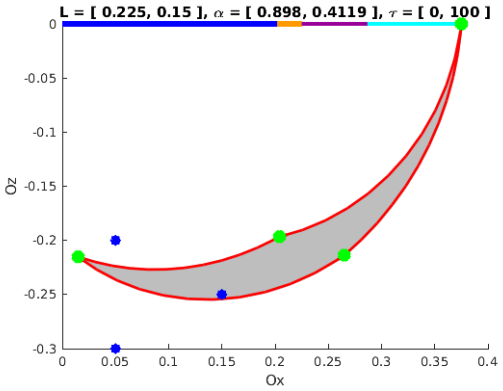
Figure 6.3: Soft robot's initial design.

Firstly, we want to show that there are some cases where the optimal solution can only reach a desired point due to the length constraint. For this, we impose the segments length bounds in $\mathcal{L} = [0,\ 0.24] \times [0,\ 0.16]m$.

Using the proposed optimization approach, we apply (6.23) on the designed soft robot in order to achieve the optimal values of L^* and α^* in order to reach the desired points. By doing this, we find the optimal segments length parameter $L^* = [0.225,\ 0.15]^T$, and the optimal actuators length parameter $\alpha^* = [0.89,\ 0.41]^T$, which gives the optimal design shown by Fig. 6.4a. As illustrated by Fig. 6.4b, we can observe that the optimal solution allows only the reachability of one single point due to the restrictions imposed on the segments length parameter $\mathcal{L}$.



(a) Optimal design.



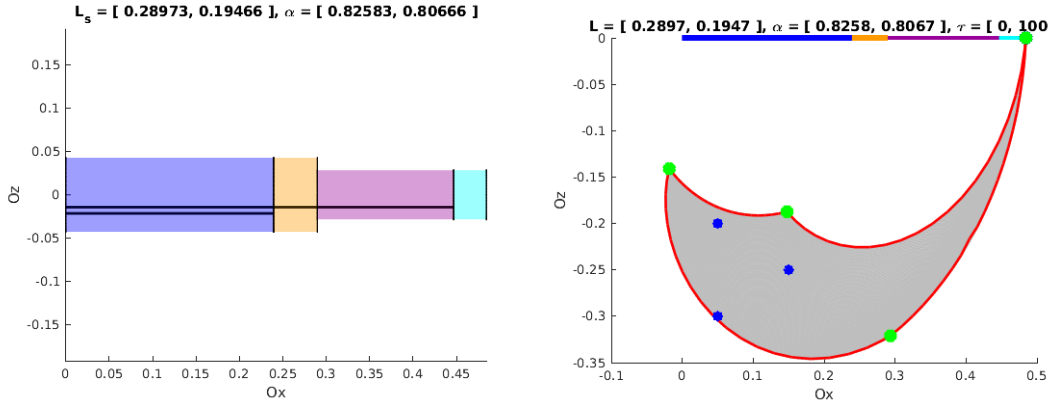
(b) $\mathcal{W}_E$ (gray) and $\partial\mathcal{W}_E$ (red).

Figure 6.4: Soft robot's optimal design - achieving only one point.

To achieve the accessibility to multiple points, we can select a larger possibility for the length bounds of each segment as $\mathcal{L} = [0,\ 0.3] \times [0,\ 0.2]m$.

Using the proposed optimization approach, we apply (6.23) on the designed soft robot. By doing this, we find the optimal segments length parameter $L^* = [0.28,\ 0.19]^T$, and the optimal

actuators length parameter $\alpha^* = [0.82,\ 0.80]^T$, which gives the optimal design shown by Fig. 6.5a. Accordingly, the workspace of the optimally designed soft robot now encloses the desired points as illustrated by Fig. 6.5b, allowing the end-effector to reach these target locations.



(a) Optimal design.

(b) $\mathcal{W}_{_E}$ (gray) and $\partial\mathcal{W}_{_E}$ (red).

Figure 6.5: Soft robot's optimal design - achieving multiple points.

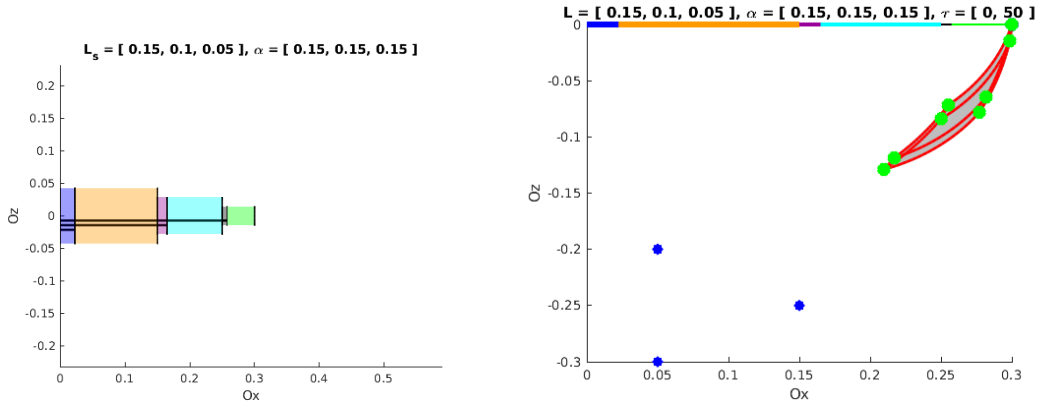
The computational time necessary to obtain the optimal results took approximately 0.445 seconds for this scenario.

6.3.0.2 Scenario 2: 3-tendons driven planar soft robot

In the second scenario, we investigate a soft robot comprised of three segments and controlled by three tendons installed on the Ozx plan (as shown in Fig. 6.6a).

In the initial state, the segments initial length vector is given $L^0 = [0.15,\ 0.1,\ 0.05]^T m$, and the initial actuators length vector is $\alpha^0 = [0.15,\ 0.15,\ 0.15]^T$. We choose the bounded tension magnitude $\mathcal{T}_{_{1,2,3}} = [0,\ 50]N$ allowing us to apply a couple magnitude within $\mathcal{C}_{_{\mathcal{T}_1}} = [0,\ 1.07]N \cdot m$, and $\mathcal{C}_{_{\mathcal{T}_2}} = [0,\ 0.71]N \cdot m$, and $\mathcal{C}_{_{\mathcal{T}_3}} = [0,\ 0.35]$. The segments length bounds are $\mathcal{L} = [0,\ 0.3] \times [0,\ 0.2] \times [0,\ 0.1]m$, and the actuators length bounds are $\mathcal{A} = [0,\ 1]^3$.

The obtained results of $\mathcal{W}_{_E}$ and $\partial\mathcal{W}_{_E}$ for the initial state of the designed soft robot with the given initial values of the parameters L^0 and α^0 are depicted by Fig. 6.6b where we can clearly observe that the workspace does not contain the desired points.



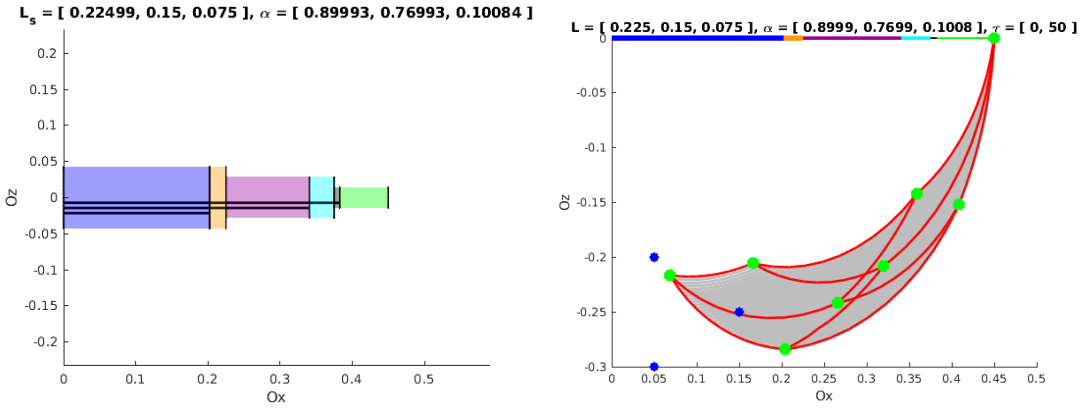
(a) Initial design.

(b) $\mathcal{W}_E$ (gray) and $\partial\mathcal{W}_E$ (red).

Figure 6.6: Soft robot's initial design.

Firstly, we want to show that there are some cases where the optimal solution can only reach a desired point due to the length constraint. For this, we impose the segments length bounds in $\mathcal{L} = [0,\ 0.24] \times [0,\ 0.16] \times [0,\ 0.08]m$.

Using the proposed optimization approach, we find the optimal segments length parameter $L^* = [0.225,\ 0.15,\ 0.075]^T$, and the optimal actuators length parameter $\alpha^* = [0.89,\ 0.76,\ 0.10]^T$, which gives the optimal design shown by Fig. 6.7a. As illustrated by Fig. 6.7b, we can observe that the optimal solution allows only the reachability of one single point due to the restrictions imposed on the segments length parameter $\mathcal{L}$.



(a) Optimal design.

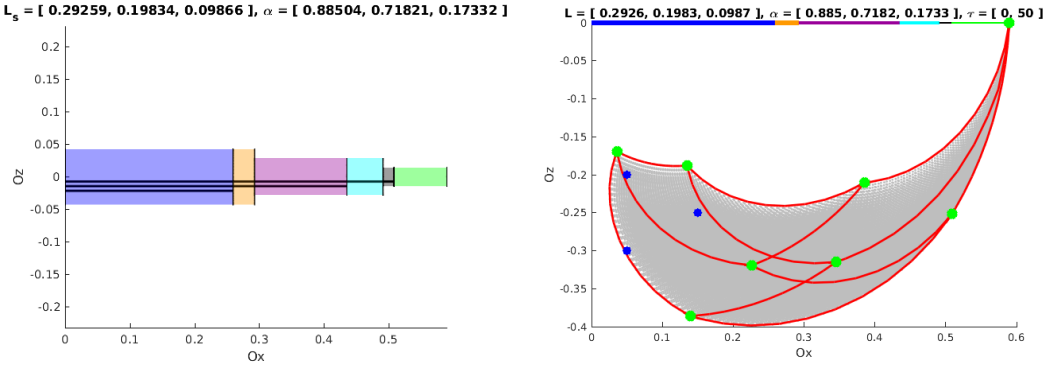
(b) $\mathcal{W}_E$ (gray) and $\partial\mathcal{W}_E$ (red).

Figure 6.7: Soft robot's optimal design - achieving only one point.

To achieve the accessibility to multiple points, we can select a larger possibility for the length bounds of each segment as $\mathcal{L} = [0,\ 0.3] \times [0,\ 0.2]m$.

Using the proposed optimization approach, we find the optimal segments length parameter $L^* = [0.29,\ 0.19,\ 0.09]^T$, and the optimal actuators length parameter $\alpha^* = [0.88,\ 0.71,\ 0.17]^T$, which gives the optimal design as shown by Fig. 6.8a. Accordingly, the workspace of the

optimally designed soft robot now encompass the desired points as illustrated by Fig. 6.8b, allowing the end-effector to reach these target locations.



(a) Optimal design.

(b) $\mathcal{W}_E$ (gray) and $\partial\mathcal{W}_E$ (red).

Figure 6.8: Soft robot's optimal design - achieving multiple points.

The computational time necessary to obtain the optimal results took approximately 0.612 seconds for this scenario.

6.4 Conclusion

In this chapter, we have proposed an optimization approach based on the adopted PCS model for the purpose of optimizing the design of a soft robot with a slender shape. The approach was then validated on different tendon-driven soft robot configurations.

However, it is important to mention that we can extend the proposed approach to also optimize the radius of the segments and the distance of the actuator to the mid-line (which will be detailed in Chapter 7). Moreover, the optimization approach can be extended to treat different cases of actuation since the PCS method was already applied to different methods of actuation, e.g., fluidic actuation [102].

On the other hand, to extend the applicability of the optimization approach in order to treat the FEM case, we need to consider what parameters should be optimized in addition to adapting the optimal problem for the FEM case (which will be explained in Chapter 7).

Chapter 7

Conclusions and Perspectives

7.1 Conclusions

In this thesis, we have investigated the workspace estimation and design optimization of soft robots. On the one hand, the workspace estimation of soft robots provides crucial information about the space where they can operate, which is useful for control applications (e.g., Pick-and-Place, trajectory planning, etc). Moreover, evaluating the workspace also helps in determining the singularities and impediments of motion to the robot which can be used as valuable information for the design optimization of soft robots. On the other hand, the design optimization of soft robots is a useful strategy to avoid the procedure of trial-and-error when designing soft robots. Given specific performance objectives, it is more logical and interesting for both economic and scientific reasons to determine the optimal design of a soft robot in order to achieve those objectives in a virtual environment before designing the final prototype of the robot.

In Chapter 2, we have adopted two mathematical methods, the first is PCS which focuses on slender-shaped soft robots, and the second is FEM which treats soft robots with general form.

To answer the direct problem of this thesis, we have firstly presented a classic approach (forward approach), which consists of discretizing the actuators space, in order to estimate the workspace of soft robots. However, due to its high complexity, alternative approaches were proposed to efficiently estimate the workspace of soft robots.

Accordingly, we have proposed a first approach in Chapter 3 (optimization-based) that consists of estimating the workspace of soft robots by mapping the exterior boundaries of the workspace. This method successfully reduced the complexity of the workspace estimation compared to the forward approach.

To address the challenge of estimating the interior/exterior points in the workspace by discretizing the end-effector space, we have proposed a new approach based on interval analysis (forward-backward) in Chapter 4 that consists of exploring all the feasible configurations in the workspace of soft robots. Compared to the forward approach, this method successfully decreased the complexity of the workspace estimation.

In a final approach to the workspace estimation problem, we have proposed in Chapter 5 a continuation approach, based on the PCS model, that consists of mapping the interior and exterior boundaries of the workspace of soft robots. Notably, it provides information about the singularities of soft robots.

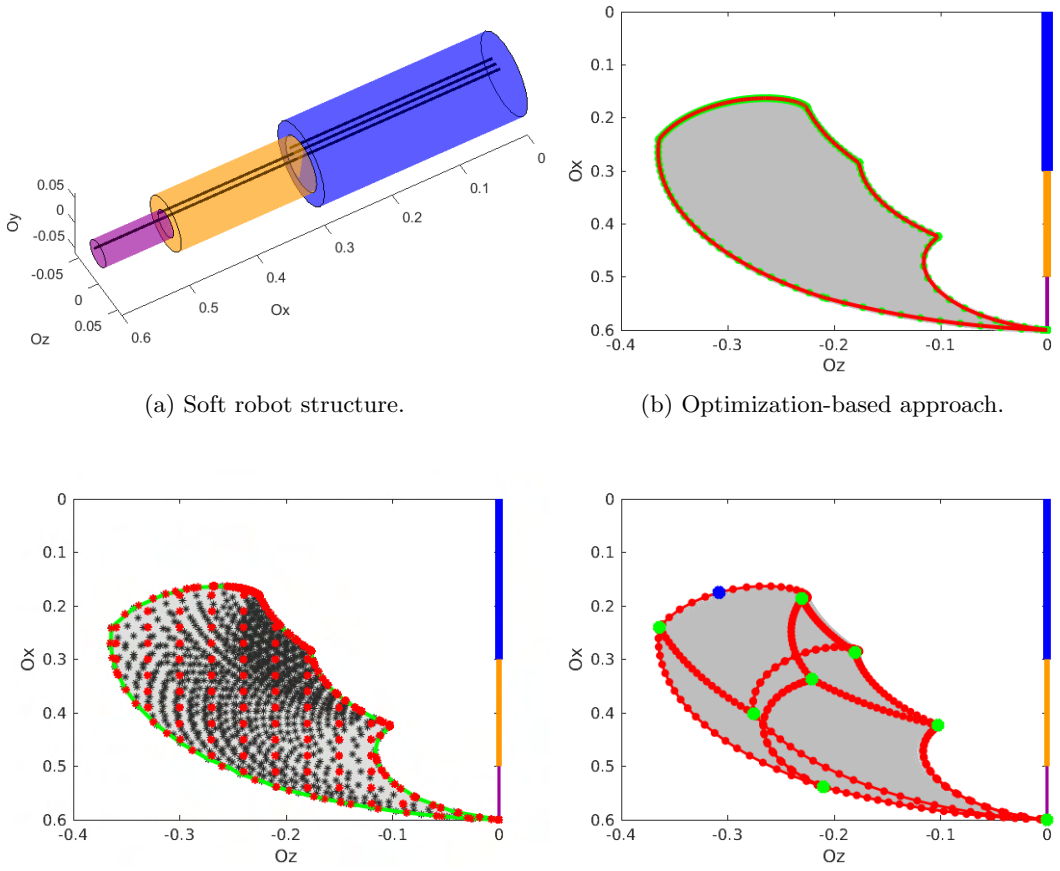
To treat the inverse problem of this thesis, we have proposed an optimization method based on the PCS model in Chapter 6 that consists of optimizing the design of soft robots in order to achieve specific desired points.

In the following, we present a brief comparison of the proposed workspace estimation approaches for both the PCS and FEM models.

7.1.1 PCS-based Workspace Estimation Approaches Comparison

Based on the PCS modeling method, three different approaches have been proposed to evaluate the workspace of soft robots and were validated through the forward approach. Each approach has its advantages and disadvantages, and the following summarizes their characteristics.

For the sake of clarity and simplicity, we consider one specific scenario (e.g., the planar soft robot composed of three segments and actuated by three tendons). The structure of the robot and its corresponding results are given in Fig. 7.1. Then, we summarize the advantages, disadvantages, operations complexity, and computational time of each investigated workspace approach in Table 7.1.



(a) Soft robot structure.

(b) Optimization-based approach.

(c) Interval Analysis approach.

(d) Continuation Approach.

Figure 7.1: Different workspace methodologies applied to the investigated soft robot modeled through the PCS model.

Table 7.1: Advantages and limitations of the investigated workspace estimation approaches based on the PCS model.

Approach / Qualities	Optimization	Interval Analysis	Continuation
Advantages	Maps outer workspace boundaries. Avoids heavy calculation of interior points. Least expensive.	Explores the whole workspace. Provide interior as well as exterior knowledge.	Maps inner and outer workspace boundaries. Ability to classify configurations.
Disadvantages	Inability to provide interior knowledge. Inability to classify configurations.	Exhaustive. Inability to classify configurations.	Inability to provide information on every point in the workspace.
Operations Complexity	220	2275	327
Computational Time (seconds)	3	72	40

In this specific scenario, the optimization-based approach is clearly the most efficient way to estimate the workspace of soft robots since it consists only in mapping the exterior boundary and avoiding the heavy computation of interior points, which also comes with the drawback that it cannot provide interior information to the workspace.

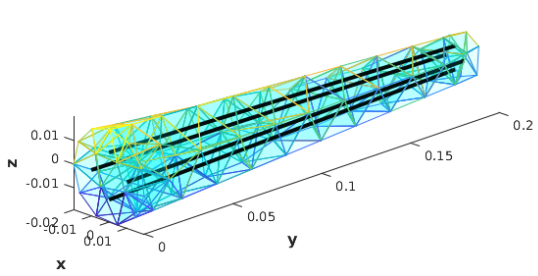
On the other hand, the interval analysis approach consists of exploring the whole workspace which allows us to have interior and exterior knowledge about the workspace, however, as a result, this approach is exhaustive.

Finally, the continuation approach allows the mapping of both interior and exterior boundaries to the workspace, this method allows us to identify the singularities of the robot which is useful for the design. However, this approach does not seek to find all achievable configurations of the workspace, but only the singularities of soft robots.

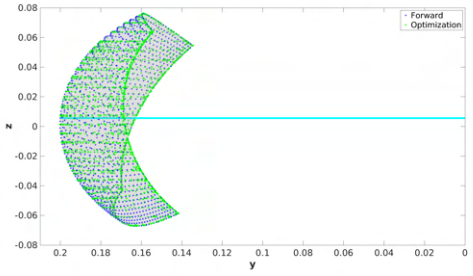
7.1.2 FEM-based Workspace Estimation Approaches Comparison

Based on the FEM modeling method, two different approaches have been proposed to evaluate the workspace of soft robots and were validated through the forward approach. Each approach has its advantages and disadvantages, and the following summarizes their characteristics.

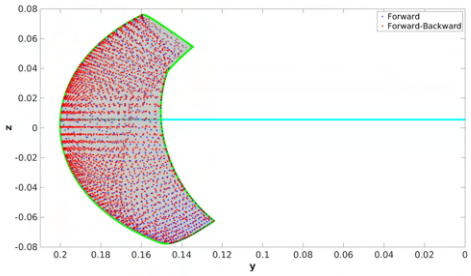
For the sake of clarity and simplicity, we consider one specific scenario (e.g., the trunk-like soft robot actuated by four symmetric tendons). The structure of the robot and its corresponding results are given in Fig. 7.2. Then, we summarize the advantages, disadvantages, operations complexity, and computational time of each investigated workspace approach in Table 7.2.



(a) Soft robot structure.



(b) Optimization-based approach.



(c) Interval Analysis approach.

Figure 7.2: Different workspace methodologies applied to the investigated soft robot modeled through the FEM model.

Table 7.2: Advantages and limitations of the investigated workspace estimation approaches based on the FEM model.

Approach / Qualities	Optimization	Interval Analysis
Advantages	Maps outer workspace boundaries. Avoids heavy calculation of interior points. Least expensive.	Explores the whole workspace . Provide interior as well as exterior knowledge.
Disadvantages	Inability to provide interior knowledge. Inability to classify configurations.	Inability to provide interior knowledge. Inability to classify configurations.
Operations Complexity	864	3667
Computational Time (seconds)	1039	3733

The interval analysis approach is computationally expensive compared to the optimization-based approach. However, the interval analysis approach provides valuable interior and exterior information of the workspace.

On the other hand, the optimization-based approach is more efficient. However, this approach only maps the exterior boundary of the workspace ignoring its interior configurations.

In the following, several possible extensions of the obtained results will be discussed.

7.2 Perspectives

7.2.1 From Position-Access to Orientation-Access Workspace

This thesis has proposed different methodologies to estimate the position-access workspace of soft robots. However, the proposed approaches can also be extended to treat the orientation aspect of the workspace. Such an extension will come with necessary adaptations of the formulation of the workspace definition and the workspace set.

For the PCS model, we can treat the orientation aspect of the workspace by taking into consideration the orientation matrix R defined in (3.2) and then adapting the algorithms established in Chapters 3, 4, and 5 in a proper way for the purpose of studying the orientation aspect.

In the FEM model, we can treat the orientation aspect of the workspace by modifying the selection matrix $\mathfrak{C}$ in (2.53). By considering 3 independent nodal positions around the end-effector $q_{_{E_1}}, q_{_{E_2}}$ and $q_{_{E_3}}$, which can be used to define a unique plan of the end-effector, we can always properly choose the matrix $\mathfrak{C} \in \mathbb{R}^{6\times 3n_p}$ such that $q_{_E}$ now is a function of those 3 points and contains both position and orientation information of the defined plan.

7.2.2 From Tendons-driven Soft Robots to other Methods of Actuation

In this thesis, we have focused on tendon-driven soft robots in both the direct problem (i.e., workspace estimation) and inverse problem (i.e., design optimization).

However, since the PCS method was applied to different methods of actuation (e.g., fluidic [102]), and that the FEM approach was also applied to different methods of actuation (e.g., pneumatic and hydraulic in [15, 107]), then, we can also use different methods of actuation by adapting the actuation matrix H in both modeling methods to the adopted actuation manner.

7.2.3 Continuation Approach: Toward a Generalization of Tangent Vectors Calculations

The continuation approach was explained in detail in Section 5.2 of Chapter 5, where tangent vectors to boundary points met along the solution curve (5.7) are calculated in Section 5.2.3 based on a classification of those boundary points into three categories (see Table 5.1): regular boundary points (see Fig. 5.3), simple bifurcation points (see Fig. 5.4), and multiple bifurcation points (see Fig. 5.5).

The tangent vectors to each boundary point class were calculated using a different approach. On the one hand, the tangent vector of regular boundary points were calculated using Theorem 4 in Section 5.2.3.1 which consists of calculating the nullspace of the workspace boundary set Jacobian $G_{_Y}$. On the other hand, tangent vectors to simple bifurcation points was calculated

using 2nd order Taylor expansion in Theorem 5 in Section 5.2.3.2 using the workspace boundary set Hessian G_{YY}. Finally, tangent vectors to multiple bifurcation points were calculated by formulating and solving the eigenvalue problem defined in Section 5.2.3.3 through the workspace boundary set Jacobian G_Y. For the purpose of generality, it is of interest to provide a single **general-purpose methodology** for calculating tangent vectors to all boundary point cases met along the solution curve.

From the first point of view, Taylor expansions can indeed be used to treat all cases. In fact, the 1st order Taylor expansion has already been used to calculate tangent vectors to regular boundary points (see Theorem 4), and the 2nd order Taylor expansion is also used in Theorem 5 to calculate tangent vectors to simple bifurcation points. However, to treat multiple bifurcation points, this methodology requires the calculation of partial derivatives of G up to the n_bth-order, with n_b being the number of branches passing through the multiple bifurcation point must be previously known. Thus, the main difficulty of this technique lies first in identifying n_b, and secondly in calculating partial derivatives of G up to the n_bth-order.

In an alternative way, the eigenvalue problem to calculate tangent vectors of multiple bifurcation points (see Section 5.2.3.3) can be extended to treat the simple bifurcation as well as regular boundary point cases. However, the extension of this approach cannot be directly implemented since it requires the existence and the knowledge of two orthonormal solutions γ_1 and γ_2 of Ψ_Z. However, this is difficult to obtain because for both simple bifurcation points and regular boundary points becaause the row-rank deficiency of Ψ_Z is 1, i.e., we can only have one orthonormal solution γ_1 that verifies the property $\Psi_Z \gamma_1 = 0$. Therefore, the main challenge using this proposed technique would be to determine the second solution γ_2 such that $\Psi_Z \gamma_2 = 0$ for multiple bifurcation points as well as regular boundary points.

7.2.4 Continuation Approach: an Extension to treat FEM

As shown in Section 5.2, the continuation approach requires the analytical representation of the parameters describing the static/kinematic model of soft robots. This is mainly due to the fact that tangent vector calculation in Section (5.2.3) requires the computation of the Jacobians Ψ_Z and G_Y, and also the Hessian matrix G_{YY}. With the PCS model, this is possible because we have the analytical form of these mathematical parameters.

However, the FEM static model uses numerical approximations to establish its parameters, therefore, in the FEM framework we don't have the analytical form of those parameters. Hence, extending the applicability of the continuation approach to treat the FEM model will not be easy, and we think that the only possible way to perform this would be to calculate the Jacobians Ψ_Z and G_Y, and also the Hessian matrix G_{YY}, via numerical techniques.

7.2.5 Design Optimization: Including more Parameters in PCS

In Chapter 6, Section 6.2, given specific performance objectives (e.g., achieving specific desired points), we have explained how to optimize the segments length parameter L and the actuators length parameter α, in order to achieve those objectives. What has been established for those parameters can also be extended in order to optimize the segments radius parameter r of the robot and the actuators distance to the mid-line d, as illustrated by Fig. 7.3.

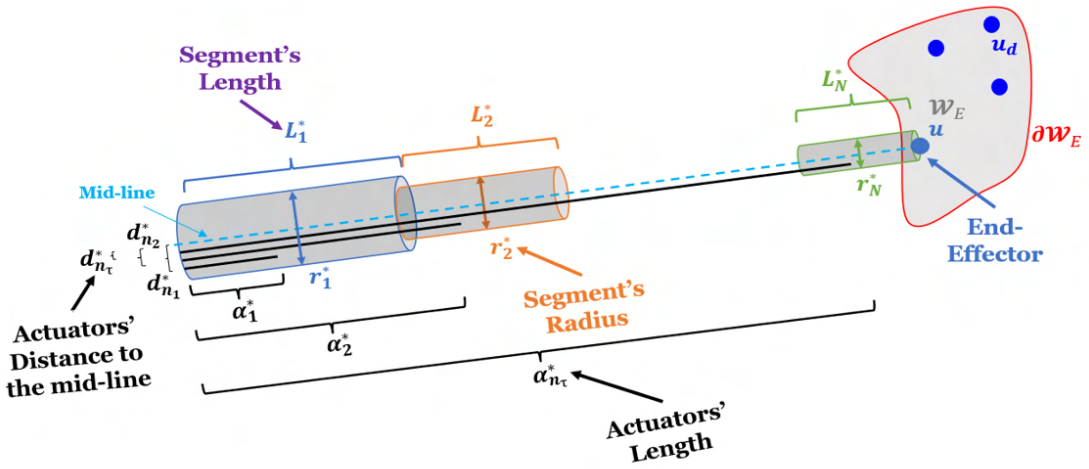


Figure 7.3: Design Optimization - Including more Parameters in PCS.

For this, the PCS static/kinematic model must consider the variations of the radius parameter r and the distance to the mid-line parameter d as what has been done for L and α in Section 6.2.1. With this, the next step is to take into account r and d in the optimal problem (6.23) by reformulating it into the following:

$$h(q, \mathfrak{X}) = \tfrac{1}{2} \sum_{j=1}^{n_S} \| u^{(j)}(q^{(j)}, \mathfrak{X}) - u_d^{(j)} \|_2^2$$

and

$$\begin{aligned} [\mathfrak{X}^*, T^*] = \underset{\{\mathfrak{X}, \tau\}}{\arg\min} \quad & h(q, \mathfrak{X}) \\ \text{s.t.} \quad & \tau^{(j)} \in \mathcal{T}, L \in \mathcal{L}, \alpha \in \mathcal{A}, r \in \mathcal{R}, d \in \mathcal{D} \\ & K(\mathfrak{X})\, q^{(j)} = H(q^{(j)}, \mathfrak{X}) \tau^{(j)} + F\left(q^{(j)}, \mathfrak{X}\right) \\ & u^{(j)} = \mathfrak{E}_1\, g^{(j)}(q^{(j)}, \mathfrak{X})\, \mathfrak{E}_2^T \end{aligned} \tag{7.1}$$

with $\mathfrak{X} = [L, \alpha, r, d]$.

One additional step that will be required for the implementation of the methodology is to calculate the gradient of the objective function with respect to r and d, i.e., $\nabla_r h$ and $\nabla_d h$.

Finally, solving (7.1) will yield the optimal segments length parameter L^*, the optimal actuators length parameter α^*, the optimal segments radius parameter r^*, and the optimal actuators distance to the mid-line d^*, allowing us to achieve a full design optimization of the investigated soft robot based on the PCS model.

7.2.6 Design Optimization: From PCS to FEM

The optimization approach proposed to achieve the optimal design of soft robots modeled by the PCS (see Chapter 6, section 6.2) can be extended to treat the FEM case as well.

For this, the same procedure followed for the PCS model might be applied to the FEM framework with necessary adaptation. The first step is to establish the static/kinematic model

in function of the parameters to be optimized, e.g., the robots' length L, the actuators length α, the radius of the robot r, and the distance of the actuators to the mid-line d, as follows:

$$K(q, \mathfrak{X})\ q = H(q, \mathfrak{X})\tau + F(q, \mathfrak{X}) \tag{7.2a}$$

$$q_{_E} = \mathfrak{C}q \tag{7.2b}$$

$$u = q_{_E} + u^{(0)} \tag{7.2c}$$

with $\mathfrak{X} = [L, \alpha, r, d]$.

The second step is to formulate the optimal problem in order to optimize the objective parameters $\mathfrak{X}$:

$$f(q, \mathfrak{X}) = \tfrac{1}{2} \sum_{j=1}^{n_{\mathcal{S}}} \| u^{(j)}(q^{(j)}, \mathfrak{X}) - u_d^{(j)} \|_2^2$$

and

$$\begin{aligned} [\mathfrak{X}^*, T^*] = \underset{\{\mathfrak{X}, \tau\}}{\arg\min} \quad & h(q, \mathfrak{X}) \\ \text{s.t.} \quad & \tau^{(j)} \in \mathcal{T}, L \in \mathcal{L}, \alpha \in \mathcal{A}, r \in \mathcal{R}, d \in \mathcal{D} \\ & K(q^{(j)}, \mathfrak{X})\ q^{(j)} = H(q^{(j)}, \mathfrak{X})\tau^{(j)} + F\left(q^{(j)}, \mathfrak{X}\right) \\ & u^{(j)} = \mathfrak{C}q^{(j)} + u^{(0)} \end{aligned} \tag{7.3}$$

Finally, solving (7.3) yields the optimal parameters to achieve the optimal design of the investigated soft robot based on the FEM model.

Bibliography

[1] K Abdel-Malek and Nada Khairallah. Serial 5dof manipulators: workspace, void, and volume determination. In *Proc. of the 25th Design Automation in Mechanisms and Robotics, Las Vegas*. Citeseer, 1999.

[2] Karim Abdel-Malek, Frederick Adkins, Harn-Jou Yeh, and Edward Haug. On the determination of boundaries to manipulator workspaces. *Robotics and Computer-Integrated Manufacturing*, 13(1):63–72, 1997.

[3] Karim Abdel-Malek and Harn-Jou Yeh. Analytical boundary of the workspace for general3-dof mechanisms. *The International Journal of Robotics Research*, 16(2):198–213, 1997.

[4] D Alciatore and C Ng. Determining manipulator workspace boundaries using the monte carlo method and least squares segmentation. *ASME Robotics: Kinematics, Dynamics and Controls*, 72:141–146, 1994.

[5] Carl Barnett Allendoerfer. *Calculus of several variables and differentiable manifolds*. Macmillan, 1974.

[6] Eugene L Allgower and Kurt Georg. *Introduction to numerical continuation methods*. SIAM, 2003.

[7] Walid Amehri, Gang Zheng, and Alexandre Kruszewski. Fem based workspace estimation for soft robots: a forward-backward interval analysis approach. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 170–175. IEEE, 2020.

[8] Walid Amehri, Gang Zheng, and Alexandre Kruszewski. Workspace boundary estimation for soft manipulators using a continuation approach. *IEEE Robotics and Automation Letters*, 6(4):7169–7176, 2021.

[9] Walid Amehri, Gang Zheng, and Alexandre Kruszewski. Fem-based exterior workspace boundary estimation for soft robots via optimization. *IEEE Robotics and Automation Letters*, 2022.

[10] Yasmin Ansari, Mariangela Manti, Egidio Falotico, Matteo Cianchetti, and Cecilia Laschi. Multiobjective optimization for stiffness and position control in a soft robot arm module. *IEEE Robotics and Automation Letters*, 3(1):108–115, 2017.

[11] Simona Aracri, Francesco Giorgio-Serchi, Giuseppe Suaria, Mohammed E Sayed, Markus P Nemitz, Stephen Mahon, and Adam A Stokes. Soft robots for ocean exploration and offshore operations: A perspective. *Soft Robotics*, 2021.

[12] Vigen Arakelian, Sébastien Briot, and Victor Glazunov. Increase of singularity-free zones in the workspace of parallel manipulators using mechanisms of variable structure. *Mechanism and Machine Theory*, 43(9):1129–1140, 2008.

[13] Siegfried Bauer, Simona Bauer-Gogonea, Ingrid Graz, Martin Kaltenbrunner, Christoph Keplinger, and Reinhard Schwödiauer. 25th anniversary article: a soft future: from robots and sensor skin to energy harvesters. *Advanced Materials*, 26(1):149–162, 2014.

[14] Alessandro Berti, Jean-Pierre Merlet, and Marco Carricato. Workspace analysis of redundant cable-suspended parallel robots. In *Cable-Driven Parallel Robots*, pages 41–53. Springer, 2015.

[15] Thor Morales Bieze, Frederick Largilliere, Alexandre Kruszewski, Zhongkai Zhang, Rochdi Merzouki, and Christian Duriez. Finite element method-based kinematics and closed-loop control of soft, continuum manipulators. *Soft robotics*, 5(3):348–364, 2018.

[16] Daniel M Bodily, Thomas F Allen, and Marc D Killpack. Multi-objective design optimization of a soft, pneumatic robot. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1864–1871. IEEE, 2017.

[17] Quentin Boehler, Isabelle Charpentier, Marc S Vedrines, and Pierre Renaud. Definition and computation of tensegrity mechanism workspace. *Journal of Mechanisms and Robotics*, 7(4):044502, 2015.

[18] Oriol Bohigas, Montserrat Manubens, and Lluís Ros. A complete method for workspace boundary determination on general structure manipulators. *IEEE Transactions on Robotics*, 28(5):993–1006, 2012.

[19] F. Boyer and F. Renda. Poincare's equations for cosserat media: Application to shells. *Journal of Nonlinear Science*, 2016.

[20] Frédéric Boyer, Vincent Lebastard, Fabien Candelier, and Federico Renda. Dynamics of continuum and soft robots: A strain parameterization based approach. *IEEE Transactions on Robotics*, 37(3):847–863, 2020.

[21] Sébastien Briot and Ilian A Bonev. Are parallel robots more accurate than serial robots? *Transactions of the Canadian Society for Mechanical Engineering*, 31(4):445–455, 2007.

[22] Sébastien Briot, Wisama Khalil, et al. Dynamics of parallel robots. *From rigid bodies to flexible elements. Springer*, 2015.

[23] Jessica Burgner, D Caleb Rucker, Hunter B Gilbert, Philip J Swaney, Paul T Russell, Kyle D Weaver, and Robert J Webster. A telerobotic system for transnasal surgery. *IEEE/ASME Transactions on Mechatronics*, 19(3):996–1006, 2013.

[24] Jessica Burgner-Kahrs, Hunter B Gilbert, Josephine Granna, Philip J Swaney, and Robert J Webster. Workspace characterization for concentric tube continuum robots. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1269–1275. IEEE, 2014.

[25] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. Continuum robots for medical applications: A survey. *IEEE Transactions on Robotics*, 31(6):1261–1280, 2015.

[26] Kun Cao, Rongjie Kang, David T Branson III, Shineng Geng, Zhibin Song, and Jian S Dai. Workspace analysis of tendon-driven continuum robots based on mechanical interference identification. *Journal of Mechanical Design*, 139(6):062303, 2017.

[27] Yi Cao, Ke Lu, Xiujuan Li, and Yi Zang. Accurate numerical methods for computing 2d and 3d robot workspace. *International Journal of Advanced Robotic Systems*, 8(6):76, 2011.

[28] U Cardaun. Stand der greiferentwicklung, fordern und heben. *Fuchleil Monlage Handhahngstechnik*, 28:40, 1978.

[29] TOMMASO Carella. Design optimization of a passive soft robotic device for neurorehabilitation. 2019.

[30] Sabri Cetinkunt and Wayne J Book. Performance limitations of joint variable-feedback controllers due to manipulator structural flexibility. *IEEE Transactions on Robotics and Automation*, 6(2):219–231, 1990.

[31] Frédéric Chapelle and Philippe Bidaud. Evaluation functions synthesis for optimal design of hyper-redundant robotic systems. *Mechanism and machine theory*, 41(10):1196–1212, 2006.

[32] Andrea Cherubini, Robin Passama, André Crosnier, Antoine Lasnier, and Philippe Fraisse. Collaborative manufacturing with physical human–robot interaction. *Robotics and Computer-Integrated Manufacturing*, 40:1–13, 2016.

[33] Matteo Cianchetti, Cecilia Laschi, Arianna Menciassi, and Paolo Dario. Biomedical applications of soft robotics. *Nature Reviews Materials*, 3(6):143–153, 2018.

[34] Matteo Cianchetti, Tommaso Ranzani, Giada Gerboni, Thrishantha Nanayakkara, Kaspar Althoefer, Prokar Dasgupta, and Arianna Menciassi. Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach. *Soft robotics*, 1(2):122–131, 2014.

[35] Anne Delettre, Guillaume J Laurent, Yassine Haddab, and Nadine Le Fort-Piat. Robust control of a planar manipulator for flexible and contactless handling. *Mechatronics*, 22(6):852–861, 2012.

[36] Cosimo Della Santina, Antonio Bicchi, and Daniela Rus. On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control. *IEEE Robotics and Automation Letters*, 5(2):1001–1008, 2020.

[37] Cosimo Della Santina, Manuel G Catalano, and Antonio Bicchi. Soft robots. *Encyclopedia of Robotics*, 2020.

[38] Cosimo Della Santina, Robert K Katzschmann, Antonio Biechi, and Daniela Rus. Dynamic control of soft robots interacting with the environment. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 46–53. IEEE, 2018.

[39] Christian Duriez. Control of elastic soft robots based on real-time finite element method. In *2013 IEEE International Conference on Robotics and Automation*, pages 3982–3987. IEEE, 2013.

[40] Yahya Elsayed, Augusto Vincensi, Constantina Lekakou, Tao Geng, CM Saaj, Tommaso Ranzani, Matteo Cianchetti, and Arianna Menciassi. Finite element analysis and design optimization of a pneumatically actuating silicone module for robotic surgery applications. *Soft Robotics*, 1(4):255–262, 2014.

[41] Douglas R Ewing, Alessio Pigazzi, Yulun Wang, and Garth H Ballantyne. Robots in the operating room—the history. In *Seminars in laparoscopic Surgery*, volume 11, pages 63–71. Sage Publications Sage CA: Thousand Oaks, CA, 2004.

[42] Nima Fazeli, Samuel Zapolsky, Evan Drumwright, and Alberto Rodriguez. Fundamental limitations in performance and interpretability of common planar rigid-body contact models. In *Robotics Research*, pages 555–571. Springer, 2020.

[43] A Gasparetto and L Scalera. A brief history of industrial robotics in the 20th century. *Advances in Historical Studies*, 8(1):24–35, 2019.

[44] Thomas George Thuruthel, Yasmin Ansari, Egidio Falotico, and Cecilia Laschi. Control strategies for soft robotic manipulators: A survey. *Soft robotics*, 5(2):149–163, 2018.

[45] Mariam Md Ghazaly, Siti Norazlin Mohd Basar, and Muhammad Shadiq Lagani. Design optimization & analysis of a soft crawling robot. *International Journal of Integrated Engineering*, 13(6):285–298, 2021.

[46] Francesco Giorgio-Serchi, Andrea Arienti, Francesco Corucci, Michele Giorelli, and Cecilia Laschi. Hybrid parameter identification of a multi-modal underwater soft robot. *Bioinspiration & biomimetics*, 12(2):025007, 2017.

[47] Olivier Goury and Christian Duriez. Fast, generic, and reliable control and simulation of soft robots using model order reduction. *IEEE Transactions on Robotics*, 34(6):1565–1576, 2018.

[48] Marc Gouttefarde, David Daney, and Jean-Pierre Merlet. Interval-analysis-based determination of the wrench-feasible workspace of parallel cable-driven robots. *IEEE Transactions on Robotics*, 27(1):1–13, 2010.

[49] Di Guo and Zhan Kang. Chamber layout design optimization of soft pneumatic robots. *Smart Materials and Structures*, 29(2):025017, 2020.

[50] Morton E Gurtin. *An introduction to continuum mechanics*. Academic press, 1982.

[51] Martin Hägele, Klas Nilsson, J Norberto Pires, and Rainer Bischoff. Industrial robotics. In *Springer handbook of robotics*, pages 1385–1422. Springer, 2016.

[52] Brian C Hall et al. *Lie groups, Lie algebras, and representations: an elementary introduction*, volume 10. Springer, 2003.

[53] Michael W Hannan and Ian D Walker. Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of robotic systems*, 20(2):45–63, 2003.

[54] Wissem Haouas, Redwan Dahmouche, Nadine Le Fort-Piat, and Guillaume J Laurent. 4-dof spherical parallel wrist with embedded grasping capability for minimally invasive surgery. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2363–2368. IEEE, 2016.

[55] Edward J Haug, Chi-Mei Luh, Frederick A Adkins, and Jia-Yi Wang. Numerical algorithms for mapping boundaries of manipulator workspaces. *Journal of Mechanical Design*, 118(2):228–234, 1996.

[56] EJ Haug, JY Wang, and JK Wu. Dextrous workspaces of manipulators. i. analytical criteria. *Journal of Structural Mechanics*, 20(3):321–361, 1992.

[57] Gauthier Hentz, Isabelle Charpentier, and Pierre Renaud. Higher-order continuation for the determination of robot workspace boundaries. *Comptes Rendus Mécanique*, 344(2):95–101, 2016.

[58] Jonathan Hiller and Hod Lipson. Automatic design and manufacture of soft robots. *IEEE Transactions on Robotics*, 28(2):457–466, 2011.

[59] Josie Hughes, Utku Culha, Fabio Giardina, Fabian Guenther, Andre Rosendo, and Fumiya Iida. Soft manipulators and grippers: a review. *Frontiers in Robotics and AI*, 3:69, 2016.

[60] Petros A Ioannou and Jing Sun. *Robust adaptive control*. Courier Corporation, 2012.

[61] Luc Jaulin, Michel Kieffer, Olivier Didrit, and Eric Walter. Interval analysis. In *Applied Interval Analysis*, pages 11–43. Springer, 2001.

[62] Doo-Yearn Jo and EJ Haug. Workspace analysis of multibody mechanical systems using continuation methods. 1989.

[63] Robert K Katzschmann, Cosimo Della Santina, Yasunori Toshimitsu, Antonio Bicchi, and Daniela Rus. Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model. In *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 454–461. IEEE, 2019.

[64] Gary L Kenaley and Mark R Cutkosky. Electrorheological fluid-based robotic fingers with tactile sensing. In *ICRA*, pages 132–136, 1989.

[65] Sangbae Kim, Cecilia Laschi, and Barry Trimmer. Soft robotics: a bioinspired evolution in robotics. *Trends in biotechnology*, 31(5):287–294, 2013.

[66] Pawel Kowal. A note on matrix differentiation. 2006.

[67] Chin-Hsing Kuo, Pei-Chun Lin, Terence Essomba, and Guan-Chen Chen. *Robotics and Mechatronics: Proceedings of the 6th IFToMM International Symposium on Robotics and Mechatronics (ISRM 2019)*, volume 78. Springer Nature, 2019.

[68] Cecilia Laschi and Matteo Cianchetti. Soft robotics: new perspectives for robot bodyware and control. *Frontiers in bioengineering and biotechnology*, 2:3, 2014.

[69] Cecilia Laschi, Matteo Cianchetti, Barbara Mazzolai, Laura Margheri, Maurizio Follador, and Paolo Dario. Soft robot arm inspired by the octopus. *Advanced Robotics*, 26(7):709–727, 2012.

[70] Cecilia Laschi, Barbara Mazzolai, and Matteo Cianchetti. Soft robotics: Technologies and systems pushing the boundaries of robot abilities. *Science robotics*, 1(1):eaah3690, 2016.

[71] Gui-Rong Liu and Siu S Quek. *The finite element method: a practical course*. Butterworth-Heinemann, 2013.

[72] Carmel Majidi. Soft robotics: a perspective—current trends and prospects for the future. *Soft robotics*, 1(1):5–11, 2014.

[73] Oren Masory and Jian Wang. Workspace evaluation of stewart platforms. *Advanced robotics*, 9(4):443–461, 1994.

[74] Stefan Mattheis, Pia Hasskamp, Laura Holtmann, Christina Schäfer, Urban Geisthoff, Nina Dominas, and Stephan Lang. Flex robotic system in transoral robotic surgery: the first 40 patients. *Head & neck*, 39(3):471–475, 2017.

[75] Steven E McHugh, Frank J Saunders, and Jason H Rife. Dynamics-based design of a soft robot. In *ASME International Mechanical Engineering Congress and Exposition*, volume 43833, pages 93–102, 2009.

[76] 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 *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 2336–2341. IEEE, 2006.

[77] J-P Merlet. Determination of 6d workspaces of gough-type parallel manipulator and comparison between different geometries. *The International Journal of Robotics Research*, 18(9):902–916, 1999.

[78] Jean-Pierre Merlet. *Parallel robots*, volume 128. Springer Science & Business Media, 2005.

[79] Jean-Pierre Merlet. On the workspace of suspended cable-driven parallel robots. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 841–846. IEEE, 2016.

[80] Stefano Mintchev, Davide Zappetti, Jerome Willemin, and Dario Floreano. A soft robot for random exploration of terrestrial environments. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7492–7497. IEEE, 2018.

[81] Zisos Mitros, Balint Thamo, Christos Bergeles, Lyndon Da Cruz, Kevin Dhaliwal, and Mohsen Khadem. Design and modelling of a continuum robot for distal lung sampling in mechanically ventilated patients in critical care. *Frontiers in Robotics and AI*, 8, 2021.

[82] Thor Morales Bieze. *Contribution to the kinematic modeling and control of soft manipulators using computational mechanics*. PhD thesis, Lille 1, 2017.

[83] Kirsten Morris. Linear-quadratic optimal actuator location. *IEEE Transactions on Automatic Control*, 56(1):113–124, 2010.

[84] Sophie Nalbach, Rukmini Manoz Banda, Sipontina Croce, Gianluca Rizzello, David Naso, and Stefan Seelecke. Modeling and design optimization of a rotational soft robotic system driven by double cone dielectric elastomer actuators. *Frontiers in Robotics and AI*, 6:150, 2020.

[85] J Needham. Science and civilisation in china: Volume 2, history of scientific thought [en línea]. cambridge, uk: Cambridge university press. isbn 0-521-05800-7, 2005.

[86] Evar D Nering. Linear algebra and matrix theory. Technical report, 1970.

[87] Chelsea Shan Xian Ng and Guo Zhan Lum. Untethered soft robots for future planetary explorations? *Advanced Intelligent Systems*, page 2100106, 2021.

[88] Surya G Nurzaman, Utku Culha, Luzius Brodbeck, Liyu Wang, and Fumiya Iida. Active sensing system with in situ adjustable sensor morphology. *PLoS One*, 8(12):e84090, 2013.

[89] James M Ortega. The newton-kantorovich theorem. *The American Mathematical Monthly*, 75(6):658–660, 1968.

[90] AP Perovskii. Universal grippers for industrial robots. *Russ Eng J*, 60(8):3–4, 1980.

[91] Henri Poincaré. Sur l'équilibre d'une masse fluide animée d'un mouvement de rotation. *Acta mathematica*, 7(1):259–380, 1885.

[92] Henri Poincaré. *New methods of celestial mechanics*, volume 13. Springer Science & Business Media, 1992.

[93] Chaoxin Charles Qiu, Chi-Mei Luh, and Edward J Haug. Dextrous workspaces of manipulators, part iii: Calculation of continuation curves at bifurcation points. *Journal of Structural Mechanics*, 23(1):115–130, 1995.

[94] Mahsa Raeisinezhad, Nicholas Grant Pagliocca, Behrad Koohbor, and Mitja Trkov. Design optimization of a pneumatic soft robotic actuator using model-based optimization and deep reinforcement learning. *Frontiers in Robotics and AI*, 8:107, 2021.

[95] Singiresu S Rao. *The finite element method in engineering*. Butterworth-heinemann, 2017.

[96] JN Reddy. *An introduction to the finite element method*, volume 1221. McGraw-Hill New York, USA, 2013.

[97] F. Renda, F. Boyer, J. Dias, and L. Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, Dec 2018.

[98] F. Renda, V. Cacucciolo, J. Dias, and L. Seneviratne. Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5495–5502, Oct 2016.

[99] F. Renda, M. Cianchetti, H. Abidi, J. Dias, and L. Seneviratne. Screw-based modeling of soft manipulators with tendon and fluidic actuation. *Journal of Mechanism and Robotics*, 2017. doi:10.1115/1.4036579.

[100] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi. Dynamic model of a multibending soft robot arm driven by cables. *Robotics, IEEE Transactions on*, 30(5):1109–1122, Oct 2014.

[101] F. Renda and L. Seneviratne. A geometric and unified approach for modeling soft-rigid multi-body systems with lumped and distributed degrees of freedom. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1567–1574, May 2018.

[102] Federico Renda, Costanza Armanini, Vincent Lebastard, Fabien Candelier, and Frederic Boyer. A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation. *IEEE Robotics and Automation Letters*, 5(3):4006–4013, 2020.

[103] Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018.

[104] Federico Renda, Vito Cacucciolo, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5495–5502. IEEE, 2016.

[105] Federico Renda, Michele Giorelli, Marcello Calisti, Matteo Cianchetti, and Cecilia Laschi. Dynamic model of a multibending soft robot arm driven by cables. *IEEE Transactions on Robotics*, 30(5):1109–1122, 2014.

[106] Werner C Rheinboldt. *Numerical analysis of parametrized nonlinear equations*. Wiley-Interscience, 1986.

[107] Alejandro Rodríguez, Eulalie Coevoet, and Christian Duriez. Real-time simulation of hydraulic components for interactive control of soft robots. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4953–4958. IEEE, 2017.

[108] D.C. Rucker and R.J. Webster. Statics and dynamics of continuum robots with general tendon routing and external loading. *Robotics, IEEE Transactions on*, 27(6):1033–1044, Dec 2011.

[109] G Runge, M Wiese, L Günther, and A Raatz. A framework for the kinematic modeling of soft material robots combining finite element analysis and piecewise constant curvature kinematics. In *2017 3rd International Conference on Control, Automation and Robotics (ICCAR)*, pages 7–14. IEEE, 2017.

[110] Gundula Runge, Jan Peters, and Annika Raatz. Design optimization of soft pneumatic actuators using genetic algorithms. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 393–400. IEEE, 2017.

[111] Daniela Rus and Michael T Tolley. Design, fabrication and control of soft robots. *Nature*, 521(7553):467–475, 2015.

[112] Renato ME Sabbatini. Imitation of life: A history of the first robots. *Brain and Mind. Electronic Magazine on Neuroscience*, (9), 1999.

[113] I Schmidt. Flexible moulding jaws for grippers. *Industrial Robot: An International Journal*, 1978.

[114] Nabil Simaan, Russell Taylor, and Paul Flint. A dexterous system for laryngeal surgery. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 1, pages 351–357. IEEE, 2004.

[115] Balkeshwar Singh, N Sellappan, and P Kumaradhas. Evolution of industrial robots and their applications. *International Journal of emerging technology and advanced engineering*, 3(5):763–768, 2013.

[116] JA Snyman, LJ Du Plessis, and Joseph Duffy. An optimization approach to the determination of the boundaries of manipulator workspaces. *Journal of Mechanical Design*, 122(4):447–456, 2000.

[117] Koichi Suzumori. Elastic materials producing compliant robots. *Robotics and Autonomous systems*, 18(1-2):135–140, 1996.

[118] Koichi Suzumori, Shoichi Iikura, and Hirohisa Tanaka. Development of flexible microactuator and its applications to robotic mechanisms. In *Proceedings. 1991 IEEE International Conference on Robotics and Automation*, pages 1622–1623. IEEE Computer Society, 1991.

[119] Koichi Suzumori, Shoichi Iikura, and Hiroshisa Tanaka. Applying a flexible microactuator to robotic mechanisms. *IEEE Control systems magazine*, 12(1):21–27, 1992.

[120] Nahema Sylla, Vincent Bonnet, Frédéric Colledani, and Philippe Fraisse. Ergonomic contribution of able exoskeleton in automotive industry. *International Journal of Industrial Ergonomics*, 44(4):475–481, 2014.

[121] 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*, 5(3):99–117, 2008.

[122] Zion Tsz Ho Tse, Yue Chen, Sierra Hovet, Hongliang Ren, Kevin Cleary, Sheng Xu, Bradford Wood, and Reza Monfaredi. Soft robotics in medical applications. *Journal of Medical Robotics Research*, 3(03n04):1841006, 2018.

[123] Amehri Walid, Gang Zheng, Alexandre Kruszewski, and Federico Renda. Discrete cosserat method for soft manipulators workspace estimation: An optimization-based approach. *Journal of Mechanisms and Robotics*, 14(1):011012, 2021.

[124] JY Wang and JK Wu. Dextrous workspaces of manipulators, part 2: Computational methods. *Journal of Structural Mechanics*, 21(4):471–506, 1993.

[125] Liyu Wang, Surya G Nurzaman, Fumiya Iida, et al. Soft-material robotics. *Foundations and Trends® in Robotics*, 5(3):191–259, 2017.

[126] Patrick Waurzyniak. Masters of manufacturing: Richard morley. *Manufacturing Engineering*, 131(1):35–35, 2003.

[127] Robert J Webster III and Bryan A Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.

[128] James F Wilson. Robotic mechanics and animal morphology. In *Robotics and Artificial Intelligence*, pages 419–443. Springer, 1984.

[129] Ke Wu and Gang Zheng. Fem-based gain-scheduling control of a soft trunk robot. *IEEE Robotics and Automation Letters*, 6(2):3081–3088, 2021.

[130] Yongxin Wu and Yann Le Gorrec. Optimal actuator location for electro-active polymer actuated endoscope. *IFAC-PapersOnLine*, 51(3):199–204, 2018.

[131] Holly A Yanco and Jill L Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, pages 111–119. sn, 2002.

[132] Han Yuan and Zheng Li. Workspace analysis of cable-driven continuum manipulators based on static model. *Robotics and Computer-Integrated Manufacturing*, 49:240–252, 2018.

[133] Hongying Zhang, A Senthil Kumar, Jerry YH Fuh, and Michael Yu Wang. Topology optimized design, fabrication and evaluation of a multimaterial soft gripper. In *2018 IEEE International Conference on Soft Robotics (RoboSoft)*, pages 424–430. IEEE, 2018.

[134] G Zheng, O Goury, M Thieffry, A Kruszewski, and C Duriez. Controllability pre-verification of silicone soft robots based on finite-element method. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7395–7400. IEEE, 2019.

[135] Gang Zheng, Denis Efimov, and Wilfrid Perruquetti. Design of interval observer for a class of uncertain unobservable nonlinear systems. *Automatica*, 63:167–174, 2016.

[136] Joseph Zhu, Carl White, Dylan K Wainwright, Valentina Di Santo, George V Lauder, and Hilary Bart-Smith. Tuna robotics: A high-frequency experimental platform exploring the performance space of swimming fishes. *Science Robotics*, 4(34), 2019.

[137] Dominik Zunt. Who did actually invent the word "robot" and what does it mean?". *The Karel Čapek website*, 2002.

Résumé Substantiel

Introduction

Les robots rigides présentent de nombreux inconvénients lorsqu'ils fonctionnent dans des environnements dynamiques et fragiles et par conséquent, les robots déformables ont été un engin émergent qui a été graduellement étudié par les chercheurs afin de surmonter ces limitations et de faire face à de nouvelles applications robotiques. Les robots déformables sont fabriqués à partir de matériaux souples et flexibles, ce qui leur permet d'avoir plusieurs caractéristiques telles qu'une grande dextérité, des collisions prudentes et sans danger ainsi qu'une flexibilité importante. Toutes ces fonctionnalités offrent de nombreux avantages pour différentes applications, notamment l'exploration de l'environnement [11, 37, 46, 80, 87, 136] et les opérations médicales [23, 25, 33, 34, 54, 74, 81, 114, 122].

Cependant, en raison de leur conformité naturelle, la modélisation des robots souples est plus complexe que celle des robots rigides, car les robots déformables comportent un nombre élevé de degrés de liberté, leur déformation est non linéaire et ils sont caractérisés par des lois mécaniques différentes de celles des robots rigides. Par conséquent, des problèmes scientifiques tels que la détermination de l'espace de travail et l'optimisation de la conception des robots déformables émergent et avec eux les possibilités de nouvelles contributions dans le domaine de la robotique souple.

Même si les robots mous ont fait des progrès intéressants au cours des dernières années, le processus de conception d'un robot mou s'inspire toujours principalement d'une série de systèmes biologiques [111], notamment la structure et le comportement d'espèces animales telles que la trompe des éléphants [53], et les bras d'une pieuvre [69]. Une telle procédure de conception peut être utile pour les tests initiaux et l'expérimentation du robot souple conçu. Cependant, lorsqu'il est confronté à des objectifs de performance spécifiques tels que la planification et le contrôle de la trajectoire [35, 129], il est possible que la conception d'un tel robot souple ne soit pas en mesure d'atteindre ses objectifs, notamment parce que sa plage d'accessibilité (c'est-à-dire son espace de travail) peut être restreinte.

Par conséquent, il est utile pour la communauté de la robotique douce de proposer des méthodologies génériques afin d'évaluer l'espace de travail des robots souples, et également d'aider et de guider la conception systématique de robots souples dans le but d'optimiser des objectifs de performance spécifiques.

L'estimation de l'espace de travail en robotique douce reste un sujet ouvert, et son importance est due aux nombreux avantages qu'elle peut apporter pour résoudre différents défis scientifiques de la robotique douce, principalement liés à la conception mécanique du robot et à la synthèse du contrôleur. D'une part, le résultat de l'évaluation de l'espace de travail peut fournir des informations sur l'accessibilité de l'objet à contrôler [35, 38, 129], c'est-à-dire en identifiant si la position de l'objet appartient à l'espace de travail du robot souple, épargnant ainsi au contrôleur

le temps de conception qui peut être perdu à essayer de contrôler un objet inaccessible en dehors de l'espace de travail du robot. D'autre part, les informations sur l'espace de travail sont également cruciales lors de la planification d'une trajectoire réalisable à suivre par des robots mous, car les positions de départ et d'arrivée d'une trajectoire réalisable doivent appartenir à l'espace de travail d'un robot [129]. En outre, l'analyse de l'espace de travail est également une étape nécessaire pour contribuer à une conception efficace, ciblée et optimale des robots mous [58], en améliorant leur stabilité et en augmentant la portée de leur manipulabilité et de leur accessibilité.

Pour atteindre cet objectif, la thèse actuelle adopte deux modèles mathématiques différents, la méthode de déformation constante par morceaux (PCS) et la méthode des éléments finis (FEM) pour décrire la déformation des robots mous de forme élancée et de forme générale, respectivement. Sur la base de ces modèles mathématiques adoptés, plusieurs approches sont proposées pour estimer l'espace de travail des robots mous. Enfin, une méthode efficace d'optimisation de la conception des robots mous en vue d'atteindre des objectifs de performance spécifiques est présentée.

Modélisation

Dans ce chapitre, nous avons présenté les modèles mathématiques des méthodes PCS et FEM dans le but de modéliser les robots mous ayant une forme élancée et une forme générale, respectivement.

Ensuite, nous avons établi la définition de l'espace de travail d'un robot souple.

Enfin, nous avons proposé une approche directe pour estimer l'espace de travail des robots mous. Cette approche a ensuite été validée en utilisant un robot souple de type tronc [129] pour les cas PCS et FEM.

Approche d'optimisation pour l'estimation de l'espace de travail

En raison de la complexité de calcul élevée de l'approche directe lors de l'estimation de l'espace de travail des robots mous, il est logique de proposer une approche qui peut discrétiser l'espace de l'effecteur final puisque sa dimension est plus petite et constante par rapport à l'espace des actionneurs pour les robots hyper-redondants, tels que les robots mous. De plus, il est plus efficace de cartographier uniquement la limite extérieure de l'espace de travail et d'éviter le calcul lourd de ses points intérieurs.

Une telle méthode [9,123] consiste à discrétiser l'espace de l'effecteur final afin de cartographier la limite extérieure de l'espace de travail, et à déduire que la zone/volume délimitée par la limite de l'espace de travail est l'espace de travail accessible.

L'approche proposée a été appliquée avec succès aux modèles PCS et FEM, où nous avons montré son efficacité à réduire la complexité et le temps de calcul nécessaires pour estimer l'espace de travail des robots mous, contrairement à l'approche directe qui explose exponentiellement lorsque la dimension des actionneurs augmente.

D'autre part, comme cette approche ne consiste qu'à cartographier la limite extérieure de l'espace de travail en déterminant les points limites situés sur la surface entourant l'espace de travail, elle ne sera certainement pas en mesure d'identifier les informations internes de l'espace de travail telles que les limites intérieures.

Approche d'analyse intervalle pour l'estimation de l'espace de travail

En raison des limitations rencontrées par l'approche d'optimisation qui ne peut pas fournir de connaissances sur les configurations intérieures de l'espace de travail, il est donc nécessaire de fournir une méthode qui peut surmonter cette limitation, mais qui est également basée sur la deuxième stratégie, c'est-à-dire la discrétisation de l'espace de l'effecteur.

Une telle méthode [7] est basée sur des techniques d'analyse d'intervalles [48, 61, 77–79], et consiste à discrétiser l'espace de l'effecteur en partant d'une configuration initiale atteignable puis en explorant l'ensemble de l'espace atteignable possible pour finalement estimer l'espace de travail d'un robot souple.

L'approche proposée a été appliquée avec succès aux modèles PCS et FEM, où nous avons montré son efficacité à réduire la complexité et le temps de calcul nécessaires pour estimer l'espace de travail des robots mous, contrairement à l'approche directe qui explose exponentiellement lorsque la dimension des actionneurs augmente.

D'autre part, comme cette approche consiste à déterminer toutes les configurations possibles qui sont réalisables, elle est utile pour identifier les informations intérieures et extérieures de l'espace de travail. Cependant, cette approche est exhaustive dans le sens où elle explore toutes les configurations réalisables de l'espace de travail au lieu de cartographier uniquement ses limites intérieures et extérieures.

Approche de continuité pour l'estimation de l'espace de travail

En raison des limites rencontrées par l'approche d'optimisation qui ne peut pas fournir la connaissance des configurations intérieures et des limites de l'espace de travail des robots mous, et également en raison des limites de l'approche d'analyse d'intervalle qui est exhaustive en explorant l'espace de travail entier, il est logique de fournir une méthodologie alternative nous permettant de déterminer seulement mais toutes les limites des espaces de travail des robots mous, à la fois intérieurs et extérieurs.

Une telle méthode est appelée l'approche de continuation [8] qui est basée sur la théorie de la bifurcation. Cette méthode est fondée sur la théorie mathématique de la bifurcation qui a été introduite à l'origine par le mathématicien français Henri Poincaré [91], et elle consiste à étudier les solutions d'équations différentielles non linéaires, en plus de l'étude des variations possibles de la structure topologique d'une famille de champs de vecteurs. Par la suite, Henri Poincaré a classé différents types de points de bifurcation [92]. En conséquence, cette approche prend en compte le comportement possible de bifurcation dans le calcul de continuation tout en cartographiant les limites intérieures et extérieures des espaces de travail des robots logiciels.

L'approche de continuation permet de cartographier les frontières intérieures et extérieures des robots mous aux formes élancées en utilisant le modèle PCS. Cependant, il est important de préciser que cette approche ne peut pas fournir d'informations sur tous les points de l'espace de travail, mais seulement sur les points limites.

Otpimisation de la conception des robots mous

Le processus actuel de conception d'un robot souple fait toujours appel à l'intuition et à la procédure d'essai-erreur. Si l'on considère des objectifs de performance spécifiques, l'approche

classique de la conception des robots mous est associée à l'incertitude d'atteindre de tels objectifs et aux dépenses économiques substantielles nécessaires pour les essais et les erreurs. Dans le but d'atteindre les objectifs souhaités, il est donc logique, pour des raisons économiques et scientifiques, d'optimiser la conception des robots mous dans un environnement virtuel avant de procéder à sa conception physique finale.

Étant donné un robot souple composé d'un nombre fini de segments de longueur limitée, et entraîné par des actionneurs montés de longueur limitée, et de magnitude limitée, ce chapitre présente une méthode d'optimisation permettant d'optimiser la conception d'un robot souple afin d'optimiser l'accessibilité de leur espace de travail.

L'approche proposée a ensuite été validée sur différentes configurations de robots souples entraînés par des tendons.

Estimation de l'Espace de Travail et l'Optimisation de la Conception des Robots Déformables

Résumé

Les robots rigides présentent de nombreux inconvénients lorsqu'ils fonctionnent dans des environnements dynamiques et fragiles et par conséquent, les robots déformables ont été un engin émergent qui a été graduellement étudié par les chercheurs afin de surmonter ces limitations et de faire face à de nouvelles applications robotiques.

Les robots déformables sont fabriqués à partir de matériaux souples et flexibles, ce qui leur permet d'avoir plusieurs caractéristiques telles qu'une grande dextérité, des collisions prudentes et sans danger ainsi qu'une flexibilité importante. Toutes ces fonctionnalités offrent de nombreux avantages pour différentes applications, notamment l'exploration de l'environnement et les opérations médicales. Cependant, en raison de leur conformité naturelle, la modélisation des robots souples est plus complexe que celle des robots rigides, car les robots déformables comportent un nombre élevé de degrés de liberté, leur déformation est non linéaire et ils sont caractérisés par des lois mécaniques différentes de celles des robots rigides.

Par conséquent, des problèmes scientifiques tels que la détermination de l'espace de travail et l'optimisation de la conception des robots déformables émergent et avec eux les possibilités de nouvelles contributions dans le domaine de la robotique souple. L'évaluation de l'espace de travail offre de nombreux avantages pour différentes applications de la robotique souple principalement liées à leur conception et à leur contrôle. En conséquence, cette thèse étudie l'estimation de l'espace de travail et l'optimisation de la conception des robots souples.

Afin d'accomplir cette tâche, on propose deux méthodes différentes pour la modélisation des robots déformables, la première est la méthode de déformation constante par morceaux (PCS), qui est utilisée pour la modélisation des robots déformables avec une géométrie continue et la deuxième est la méthode des éléments finis (FEM), qui est utilisé pour la modélisation des robots déformables avec une géométrie générale.

Ensuite, basée sur ces modèles mathématiques, cette thèse propose différentes méthodologies pour estimer l'espace de travail des robots souples. Deux stratégies ont été proposées pour l'estimation de l'espace de travail, la première consiste à discrétiser l'espace des entrées (actionneurs), et la deuxième consiste à discrétiser l'espace des sorties (l'effecteur du robot).

Cependant, la première stratégie est inefficace, car elle dépend de la dimension des actionneurs et celle-ci varie en fonction de la configuration du robot déformable étudiée.

En revanche, la seconde stratégie présente une méthodologie stable et efficace pour l'estimation de l'espace de travail puisque l'espace de l'effecteur est toujours constant (inférieur ou égal à 3, si nous nous concentrons sur l'aspect position de l'espace de travail), quelles que soient la configuration du robot souple étudié et la dimension des actionneurs.

Les approches proposées pour l'estimation de l'espace de travail ont ensuite été appliquées aux deux modèles mathématiques adoptés et validées à l'aide de différentes configurations de robots déformables.

Enfin, cette thèse propose une approche d'optimisation basée sur les modèles mathématiques adoptés pour optimiser la conception des robots souples afin d'atteindre certains objectifs spécifiques.

Mots-clefs : Robot Mou, Modèle Cinématique/Dynamique, Espace de Travail, Optimisation.

Workspace Estimation and Design Optimization of Soft Robots

Abstract

Soft robots are an emergent instrument that has gradually been investigated by researchers in the recent years to overcome limitations of traditional rigid robots as well as to propose novel robotic applications. Rigid robots are challenged when operating in restricted and dynamic environments. Being made from a soft and flexible material, soft robots provide many benefits such as high dexterity, safe interactions, and increased adaptability, which are very useful for various applications, especially the manipulation of fragile objects, environment exploration, and medical operations.

However, owing to their natural conformity and compliance, soft robots consist of a large number of degrees of freedom, their deformation is highly nonlinear, and they are characterized by different mechanical laws compared to that of rigid robots. All these aspects make their modeling more complex.

Consequently, scientific challenges such as workspace evaluation and design optimization of soft robots arise and with them the opportunities of new contributions in the field of soft robotics.

The workspace evaluation provides many benefits for different soft robotic applications mainly related to their design and control.

Accordingly, the present thesis investigates the workspace evaluation and design optimization of soft robots.

For this, two different methods are adopted for the mathematical modeling of soft robots. The first is the Piece-wise Constant Strain (PCS), which focuses on slender-shaped soft robots, and the second is the Finite Element Method (FEM), which concerns soft robots with a general shape. Based on the adopted mathematical models, this thesis proposes different methodologies to estimate the workspace of soft robots.

In fact, two strategies can be followed for the workspace estimation of soft robots: the first consists of discretizing the inputs (actuators) space, and the second consists of discretizing the outputs (end-effector) space.

However, the first strategy is not efficient as it depends on the dimension of the actuators which are used to control the investigated soft robot, and varies corresponding to the particular studied configuration.

Conversely, the second strategy presents a stable and efficient way for the workspace estimation since the end-effector's space is always constant (smaller or equal to 3, if we focus on the position access) regardless of the studied soft robot's configuration and the dimension of actuators.

Each proposed workspace estimation approach was then applied to both the PCS and the FEM models and validated via a variety of soft robots' configurations.

Finally, this thesis proposes a model-based optimization approach in order to optimize the design of soft robots for the purpose of achieving specific performance objectives.

Keywords: Soft Robots, Kinematic/Dynamic Modeling, Workspace Estimation, Optimization.