TÉCNICO LISBOA

Experiments on Control of a Soft Continuum Manipulator Robot

Clara Mayumi Bertolino Hamada

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof. João Carlos Prata dos Reis
Prof. Egidio Falotico

Examination Committee

Chairperson: Prof. Duarte Pedro Mata de Oliveira Valério
Supervisor: Prof. João Carlos Prata dos Reis
Member of the Committee: Prof. Susana Margarida da Silva Vieira

July 2021

Acknowledgments

I would like to thank my parents for their encouragement and caring over all these years, for always being there for me and without whom my exchange program would not be possible.

I would also like to acknowledge my dissertation supervisors Prof. João Reis and Prof. Egidio Falotico, Prof. Flávio Trigo and the PhD student Francesco Pique for their insight, support and sharing of knowledge that has made this Thesis possible even during the COVID-19 pandemic.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me during the good and bad times in my life. Thank you.

To each and every one of you – Thank you.

Abstract

Continuum soft robots exhibit characteristics, such as compliance and dexterity, that offer greater security in human-robot interaction when compared to traditional robots. As a relatively recent field, continuum robotics has a potential growth not only in control but also in design. This study presents the construction, testing and control of a continuum soft robot to be used in the I-SUPPORT system, replacing the McKibben-based actuators used, with silicone ones. After it was detailed the prototype's manufacturing, the mechanical and kinematics characterization were made comparing both actuators. The mechanical characterization was done by measuring the elongation of the single actuators by supplying pressure inputs. The kinematics characterization was made with the actuators assembled in the robot, including elongation, bending, reachable workspace and repeatability. The silicone actuators showed to be a good alternative, offering more reliability and easy maintenance to the manipulator. It was also proposed a control concept capable of solving the inverse kinematics using polynomial regression, which requires no prior knowledge about the robot. The effectiveness of the control was demonstrated by simulation data and experimentally, by an open-loop task space control using straight line, circle, and 8-shaped point-to-point trajectories. The results presented a promising control for the continuum soft robot proposed.

Keywords

Continuum robotics; Machine learning; Robot control; Polynomial regression; Soft actuators.

Resumo

Os robôs contínuos macios exibem características, como conformidade e destreza, que oferecem maior segurança na interação homem-robô quando comparados aos robôs tradicionais. Como um campo relativamente recente, a robótica contínua tem um crescimento potencial não apenas em controle, mas também em design. Este estudo apresenta a construção, teste e controle de um robô soft contínuo para ser usado no sistema I-SUPPORT, substituindo os atuadores baseados em McKibben usados, por outros de silicone. Depois de detalhada a fabricação do protótipo, foi feita a caracterização mecânica e cinemática comparando os dois atuadores. A caracterização mecânica foi feita medindo o alongamento de atuadores individuais mediante o fornecimento de entradas de pressão. A caracterização cinemática foi feita com os atuadores montados no robô, incluindo alongamento, flexão, área de trabalho alcançável e repetibilidade. Os atuadores de silicone mostraram-se uma boa alternativa, oferecendo mais confiabilidade e fácil manutenção ao manipulador. Também foi proposto um conceito de controle capaz de resolver a cinemática inversa por meio de regressão polinomial, que não requer conhecimento prévio sobre o robô. A eficácia do controle foi demonstrada por dados de simulação e experimentalmente, por um controle de espaço de tarefa de loop aberto usando trajetórias ponto a ponto de linha reta, círculo e em forma de 8. Os resultados apresentaram um controle promissor para o robô macio contínuo proposto.

Palavras Chave

Robótica contínua; Aprendizado de máquina; Controle do robô; Regressão polinomial; Actuadores macios.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms

DoF	Degrees of Freedom
SQP	Sequential Quadratic Programming
BHA	Bionic Handling Assistant
CC	Constant Curvature

1

Introduction

Contents

1.1 Motivation

Conventional robotics are composed of a chain of rigid bodies, or links, connected by joints in which the motion results from a composition of elementary motions of each link with respect to the previous one [1]. This rigid materials structure allows accuracy, precise and fast motion, which are desirable characteristics in many applications, specially in well-structured environments, like in industrial tasks.

Although the rigid-link robots are widely used, there are situations where different attributes may provide better performance. Continuum soft robots, unlike rigid-link robots, are made of soft materials. Considered a new wave of robotics technology, they are inspired on biological structures, such as octopus tentacles [2], caterpillars [3], elephant trunks [4–6] and worm bodies [7]. Their movements result from the continuous bending along their length thanks to nonlinear stiffness and viscosity properties of the soft materials that generate smooth curves [8].

Their high compliance, omnidirectional mobility and adaptive behaviour allowed the continuum robots to recently perform a significant role in several areas, mainly where unexpected interactions with unstructured environments or humans are required [9]. Applications include endoscopy [10] and surgery [11], hot-cell decontamination [12], agricultural harvesting [13], search and rescue operations [14] and much more.

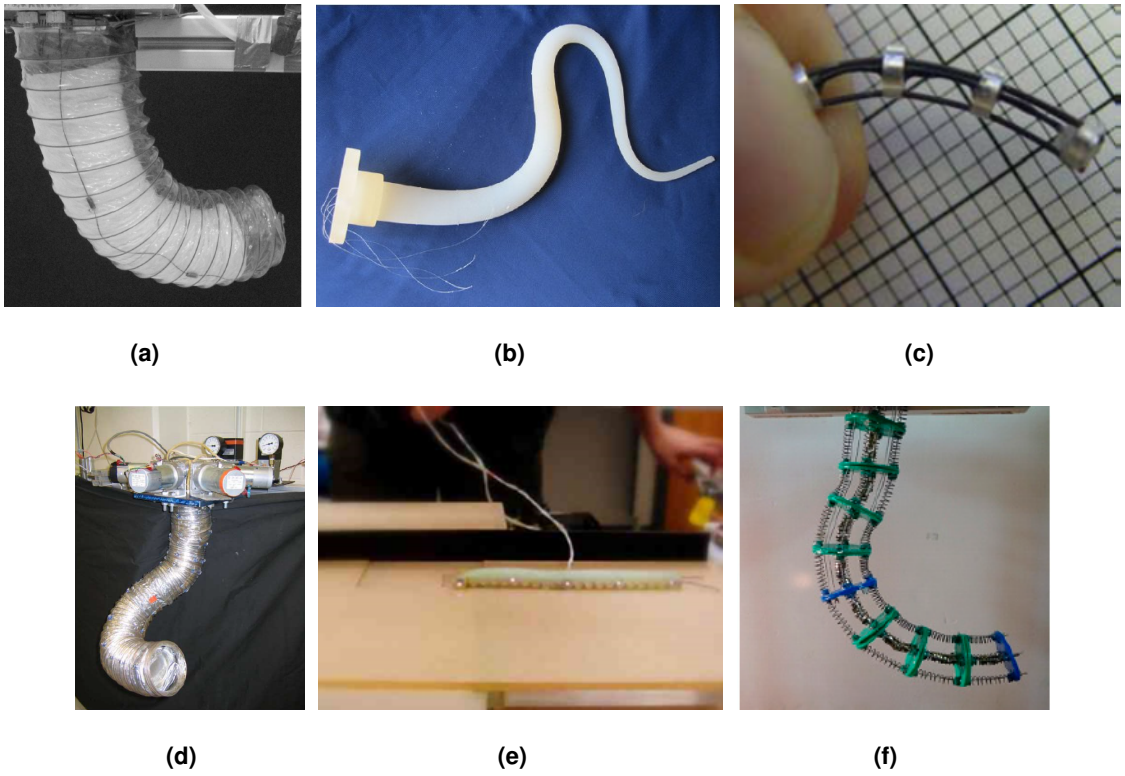
As a relatively recent field in robotics, continuum robots' study has a potential growth not only in control but also in design. The challenges of using them rely on finding proper materials that ensure the integration of actuation, powering and logic. Obstacles are also find in design the manipulator prototype to allow the performance of the goal task [15] and in modelling and controlling the robot's movements with positional accuracy. This is due to the fact that analytical or numerical methods are very complex to be developed because of their virtually infinite Degrees of Freedom (DoF) motions and the restriction of using high-frequency controllers imposed by the nonlinear characteristics of the materials [16].

1.2 State of the Art

1.2.1 Continuum robots

Robinson and Davies in [8] divide the robots in three classifications based on links and joints. The discrete robots are constructed by a series of rigid links connected by discrete joints, which corresponds to the traditional robots mechanism. The serpentine robots also have several discrete joints but they are connect by short rigid links, which results in smoother movements when compared to the discrete robots. The continuum robots do not contain any rigid links and joints, which increases their flexibility and dexterity in relation to the other two types of robot and produces curves in space even smoother than the serpentine robots.

Continuum robots can be categorized in terms of their structural design, materials and actuation strategy. In terms of mechanical structure these robots are divided with respect of their segments and the presence of discs in their structure being classified in four categories: single segment, multi-segment, single segment-multi-disc and multi-segment-multi-disc.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 1.1: Continuum robots classified by its mechanical structure: single segment in (a) [4], (b) [2] and (c) [11] with multiple discs, multi-segment in (d) [5], (e) [3] and (f) [13] with multiple discs.

A single segment robot's body consists in a single backbone-like structure, is the simplest structure of continuum robots. Jones et al. in [4], inspired by an elephant trunk, proposed a single segment manipulator (Figure 1.1(a)) with a central hose pneumatically actuated fitted over with an outer tube tendon actuated. Cianchetti et al. in [2] got inspiration from the octopus arm and designed a single conical segment manipulator (Figure 1.1(b)) made of silicone actuated by several cables arranged inside its structure.

When the continuum robot's body consists of multiple segments, each one behaving as a single segment, with independent movement from the other, it is classified as a multi-segment structure. Lin et al. in [3] proposed the GoQBot (Figure 1.1(e)), a robot that simulates caterpillar locomotion constituted by two segments working independently of each other by tensile actuators that produce the displacement between body contact points. McMahan et al. in [5] also inspired by an elephant trunk designed the Air-Octor (Figure 1.1(d)), an extension of the work developed by Jones et al. in [4]. It consists in a

two-section manipulator with hybrid actuation of pneumatic chambers and tendon-cable servo system with additional DoF and capability when compared to the single section robot.

In order to enhance functionalities, some robots have in their structure multiple discs inserted, working as a backbone for them. When the previous structures described are added with discs of the same or varying radii arranged equidistant from each other they are classified as two new categories based on the segments: single segment multi-disc or multi-segment multi-disc. Simaan in [11] proposed a snake-like single segment with several discs (Figure 1.1(c)) that are responsible to maintain a fixed radial distance between the flexible super-elastic hollow tubes that form the robot's structure. Inspired by an octopus, Yeshmukhametov et al. in [13] proposed the TakoBot 2, actuated by tendon-drive mechanism is a two sections robot with multiple sliding spacer discs, that improve the robot's dexterity and bending features maintaining its accuracy (Figure 1.1(f)).

There is an wide range of materials used for the design and assembly of the continuum robots [17]. The bioinspiration of this type of robot is not limited to its shape, but also extends to the choice of materials they are made from, usually soft materials. Because of that, the continuum robots and the soft robots are often treated as synonyms. Soft materials offer numerous advantages at potentially lower cost, such as lower impact force, which makes human-robot interaction safer; high flexibility and deformability, allowing adaptability to complex environments and energy absorption to maintain stability [9, 15].

Polyamide additive manufactured was used in the Bionic Handling Assistant (BHA) robot designed by Rolf and Steil in [6], resulting in a flexible and lightweight manipulator. Silicone was used in the GoQBot [3] and in the octopus robotic arm proposed by Cianchetti et al. in [2], offering lost cost and flexibility, resistance to abrasion, acidic and basic environment. A braided surface was used in the worm body soft robot with peristaltic movement developed by Seok et al. in [7], providing high strength and lightweight. Simaan et al. in [11] used super-elastic hollow tubes, which provides flexibility and could be used for secondary purposes in the surgical field. Providing greater torque and larger payload capacity, Hannan and Walker [18] developed their soft manipulators using springs and elastic connection between the segments and Yeshmukhametov et al. [13], with springs and plastic spacer discs.

The actuator is the component of the robot that allows its physical movement. As well as the variety of materials adopted for its manufacture, the range of actuators used in continuous robots is also wide. According to the investigative survey of Burgner-Kars et al. in [19] only in medical applications it can be found tendon-driven mechanisms, multibackbone structures, hydraulic and pneumatic chambers, shape memory effect, embedded micromotors, fluidic elastomers and McKibben muscles. However, two of them are most common to be used: pneumatic actuators and tendon actuators [17]. The pneumatic actuators work based on the pressurization of a chamber, which allows extension, bending and sometimes the contraction of the manipulator, it is usually low-cost. The tendon actuators are driven mechanisms

activated by electrical motors (DC, servo or stepper motors), it offers a better stiffness.

According to the method and location of the mechanical actuation the continuum robots could be classified as intrinsic, extrinsic and hybrid. In an intrinsic robot the actuators are within the moving manipulator structure itself. In the extrinsic devices the actuators are outside of the main structure, and work with a mechanical transmission of the forces. In the hybrid manipulators the actuation is based on a combination of intrinsic and extrinsic actuation [8].

The Active Hose developed by Tsukagoshi et al. in [14] and inspired by an elephant trunk uses as intrinsic actuator a wounded tube, a spiral tube surrounding the unit like a coil that when pressurized inside, pushes each other to its radius direction. Ranzani et al. in [20] inspired by the motion capabilities of the octopus arm, developed the STIFF-FLOP, a manipulator that uses a combination of two intrinsic actuators to produces its movement: a granular jamming stiffening system and radially arranged pneumatic chambers. The BHA [6] is also another example of intrinsic actuation in continuum robots, it works using three radially arranged bellow-shaped pneumatic chambers in each one of its modules.

The octopus robotic arm [2] of Cianchetti et al. using several cables arranged inside the conic body of the manipulator is an example of extrinsic actuation. The continuum robot developed by Su et al. in [21] is based on a backbone formed by concentric compliant tubes, that can move freely with respect to each other. These tubes are activated by a timing belt mechanism which drives the motion produced by piezoelectric motors, also characterizing an extrinsic actuation. In medical field, a steerable cardiac catheter is proposed by Camarillo et al. in [22] using tendon-driven mechanisms.

The hybrid actuation is found in the Air-Octor [5] and the single module from Jones et al. [4], both combining a single central pneumatic actuator with radially arranged cables. Immega and Antonelli in [23] proposed the KSI tentacle, in which a central pneumatic bellows works together with electrical motors that drives six tendons, three in each one of its module. Stilli et al. in [24], also inspired by an octopus arm, proposed a shrinkable, stiffness-controllable soft robot consisting of modules constructed by an internal bladder with an outer fabric sleeve pneumatically actuated and tendons attached to the outer sleeve allowing each module to bend.

1.2.2 Applied Control in Continuum Robots

As mentioned in Section 1.1, the control of the continuum robots is a nontrivial task, since they have infinite DoF and they are made of materials that exhibit highly nonlinear characteristics. The choice of the controller approach depends on the robot's application, design, actuator and sensor availability. In this project, the controllers will be described according to their modelling approach, being classified as model-based and model-free controllers. Model-based controllers are preferable for compact manipulators acting in structured environments, presenting high accuracy and reliability. Model-free controllers are more used in manipulators that present high non-linearity and are not uniform, characteristics that

offer great challenges to the modeling.

Model-based controllers correspond to controllers that rely on developing analytical models, they are the currently most widely used and studied strategy of control of the soft continuum robots [16]. The simplest modelling method is the Constant Curvature (CC) developed by Hannan and Walker in [18], which models the complex motion of the continuum robots, approximating it by a series of simple coupled motions. This method, although ignores a large portion of the manipulator dynamics, showed to be a very good approximation when its curvature is approximately constant, its actuators are symmetric distributed and the torsion and the external forces can be neglected.

A more complex modelling technique was developed by Gravagne et al. in [25]. It was studied the dynamics of a planar continuum robot, approximating it of a large-deflection beam, which considers the torsional effects acting in the manipulator but not the distributed loading. This approach results in large position error of the end-effector when the effects of gravitational loading should be considered.

Considering the self-weight and other external forces, the Cosserat rod theory was first applied in soft robots modelling by Trivedi et al. in [26]. This approach divides the continuum robot's body in infinitesimally small rigid bodies which can rotate independently from its neighbors. Although more complex models showed to be more accurate in modelling the continuum robots, their computational and sensing cost increased dramatically, which ended up limiting their usage.

After modelling the direct kinematics of the manipulator, in order to achieve a task-space (position, orientation or force) control goal, the inverse kinematics problem should be solved. Hannan and Walker in [18], Bailly and Amirat in [10] and Jones and Walker in [27] obtained the inverse kinematics using differential inverse kinematics through the computation of the Jacobian. Camarillo et al. in [28] obtained it by direct inversion, using the Moore-Penrose pseudoinverse. The inverse kinematics problem was also solved by optimization by Camarillo et al. in [29], approaching the problem using nonlinear programming solver with constrains.

With the inverse kinematic problem solved it could be applied different control strategies from open-loop to closed-loop. Giorelli et al. in [30] applied an open-loop control, using the target position as input to the inverse model, which predicts the actuators inputs, that were supplied to the robotic manipulator. Camarillo et al. in [29] used vision feedback as a sensor for closed-loop control of the catheter robot's tip. Gravagne et al. in [25] applied a PD with cable tension feedback for setpoint tracking, minimizing the incidental vibrations excited in the backbone of the spring-steel manipulator studied.

Model-free controllers use machine learning techniques and empirical methods. These approaches are data dependent, which means that in order to obtain the trained models, real world data of the continuum robot should be acquired first. Model-free, compared to model-based controllers, have the advantage of being independent of the manipulator shape, but require high quality and large amount of data for training the models [16].

Giorelli et al. in [30] proposed a feedforward neural network for learning the inverse kinematics. Their approach uses a nonredundant manipulator driven by cables in an open-loop control to compare the results obtained through model-based and learning method. The study indicates that model-free controllers are effective in solving the inverse kinematics problem due to its ability to take into account uncertainties.

Thuruthel et al. in [31] formulated the inverse kinematics problem like the differential inverse kinematics using local mappings, functions of the current actuator configuration and the current and the next end-effector positions to the next actuator configuration. The problem, which consists in a closed-loop control, was solved using neural network. This approach allows the generation of multiple global solutions to the inverse kinematic problem through an iterative method.

Another model-free approach was developed by Ansari et al. in [32, 33], solving the inverse kinematics problem using multiagent reinforcement learning, in which each actuator of the manipulator is considered as an agent that behaves autonomously within the environment. Through a reward system, the agents are motivated to select actions that lead the manipulator towards the goal. The solution represents the low-level controller, that in [33] was incorporated to a hierarchical controller.

1.3 Research Goals

This thesis was inspired by the I-SUPPORT service robotics system developed in the framework of the EU Horizon2020 Program applying soft robotics. The system envisions to support and enhance senior people to bathing, one of the activities classified as basic in the daily living of people and that can offer difficult tasks for elderly, such as reaching their back and lower limbs to wash [34]. The system (Figure 1.2) is composed by a motorized chair and two soft robotic arms: one for provide pouring water and the other for scrubbing [35]. Thanks to the use of soft robotic technology the system offers: safety, reliability and adaptability for human-robot interaction; high dexterity with a large reachable workspace, being able to reach difficult and sensitive regions of the body.

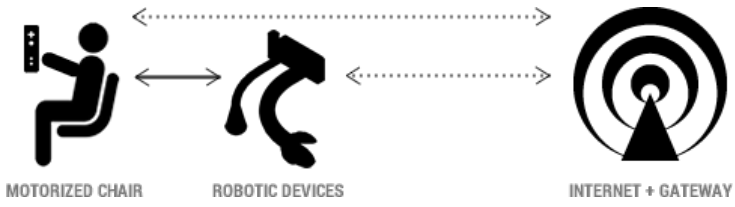


Figure 1.2: I-SUPPORT Robotic System. Adapted from [34].

However, since the McKibben-based actuators used in the I-SUPPORT robotic arm are manually manufactured, they could not be easily replicated with the same dimensions and symmetry, which make their maintenance time-consuming. This lack of manufacturing precision together with the rapid degra-

dation of the internal latex chamber of the actuators reduces their reliability, since each maintenance requires a new identification of the system, which means training the models all over again. Aiming to address these challenges, this thesis project proposes a soft robotic manipulator with actuators made from commercially available silicone bellows to be used in the I-SUPPORT system.

On the other hand, one of the most difficult tasks when using soft robots is to control their motion. The choice of the controller method depends on several characteristics of the robot, like its design, application and actuator used. Holsten et al. [36] proposes a new controller method that requires no prior knowledge about the actuators or the robot. This approach is a model-free controller that allows adaptation to erroneous manufacturing, which makes it suitable for controlling the I-SUPPORT manipulator, considering the challenges faced. Given these characteristics, this thesis project proposes a controller based on the approach used by Holsten et al. in [36] to be applied in the presented manipulator.

The goals of this thesis project could be summarized in:

- use of silicone actuators in the I-SUPPORT manipulator, offering a fast and easy maintenance and making them less susceptible to manufacturing errors;
- the kinematic characterization of the actuators assembled in the robot, which includes elongation, bending, reachable workspace and repeatability;
- the mechanical characterization of the single actuators measuring the elongation of them under different pressures inputs;
- implementation of open-loop controller method based on Holsten et. al [36], which uses polynomial regression to obtain the direct kinematic model of the robot;
- validation of the proposed controller using tracking tests with straight line, circle and 8-shaped point-to-point trajectories.

1.4 Thesis Outline

This introductory chapter provided the motivation of the thesis and the framework in which the work developed is inserted, as well as the goals to be achieved.

Chapter 2 presents the theoretical background that is needed to fully analyze the problem and explain the control method applied. And also reveal the materials and simulations adopted to obtain the results of the work.

Chapter 3 describes each component of the prototype, including the manufacturing method and materials used.

The results were divided in two steps. Section 4.1 presents the pre-experimental results using simulation data and previous experimental data provided by The BioRobotics Institute from Sant'Anna School

of Advanced Studies. Section 4.2 presents the experimental results using the robots assembled with the McKibben-based actuators and the proposed actuator.

Chapter 5 discusses the conclusions of the present work, together with future directions for further investigation on the subject at hand.

2

Materials and Methods

Contents

Before explaining in details the control method applied in this research, it will be provided the explanation of theoretical background necessary to fully understand it. Then, the control method approach is presented dividing it in main steps performed by the algorithm and it will be state some considerations of the method to ensure its proper functioning. In this Chapter, it will also be described the simulation model and the robotic arms used during the Pre-Experimental and Experimental phases of the project, as well as the experimental setup.

2.1 Theoretical Background

2.1.1 Direct and Inverse Kinematic Problems

All robotic applications require the execution of a specific motion to accomplish the robot's tasks. The correct execution of the tasks depends on the controller, that aims to provide the correct commands to the robot's actuators to produce the desired motion. Learning the relationship between the inputs and the outputs that characterizes the robot's motion is the key to control the robot.

In conventional robots, direct kinematic (Equation (2.1)) establishes the relationship (f) between the joint variables (q) and the end-effector position and orientation (x_e).

$$x_e = f(q). \tag{2.1}$$

The inverse kinematics (Equation (2.2)) corresponds to the inverse: to determine the joint variables (q) that corresponds to a given end-effector position and orientation (x_e).

$$q = f^{-1}(x_e). \tag{2.2}$$

The solution of the inverse kinematic problem is essential to transform the desired end effector-motion in the operational space to the corresponding joint space motions [1].

In continuum robotics, the joint variables are substituted by the actuator space, that concerns the actuation parameters, and can be, for example, the pressures in the pneumatic actuators or the length of the cables. Therefore, the direct kinematics maps the actuator space (q) into the task space (x_e). The task space concerns to the position and orientation of the end-effector [30].

2.1.2 Polynomial Regression

The regression analysis consists of finding the relationship between one or more variables, aiming to get information about one of them through knowing values of the other(s). Mathematically speaking, in a regression problem, given a sample of Y values with their associated values of x_i, $i = 1, 2, ..., k$, the

goal is to estimate on the basis of this sample the relationship between Y and the independent variables $x_1, x_2, ..., x_k$. Equation (2.3) illustrates the problem, where ϵ is a random error, with constant variance and zero mean value.

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k + \epsilon \tag{2.3}$$

The data in a multiple regression model with k predictors consists of $n(k+1)$ tuples of the form $(x_{1n}, x_{2n}, ..., x_{kn}, y_n)$, where x_{ij} is the value of the ith predictor x_i associated with the observed value y_j.

Given the trial function $y = b_0 + b_1 x + ... + b_k x_k$, the least squared principle is used to estimate the β_k parameters of the Equation (2.3). According to the principle of the least squares, the estimated βs are the values b_i that minimize the sum of squared deviations of the observed y_js from the trial function y given by f on Equation (2.4).

$$f(b_0, b_1, ..., b_k) = \sum_{j=1}^{n} [y_j - (b_0 + b_1 x_{1j} + b_2 x_{2j} + ... + b_k x_{kj})]^2 \tag{2.4}$$

Taking the $k+1$ partial derivatives with respect to b_0,b_1,..b_k and then equating them to zero, it is possible to obtain a system of normal equations given by Equation (2.5).

$$\begin{aligned} b_0 n + b_1 \sum x_{1j} + b_2 \sum x_{2j} + ... + b_k \sum x_{kj} &= \sum y_j \\ b_0 \sum x_{1j} + b_1 \sum x_{1j}^2 + b_2 \sum x_{1j} x_{2j} + ... + b_k \sum x_{1j} x_{kj} &= \sum x_{1j} y_j \\ &\vdots \\ b_0 \sum x_{kj} + b_1 \sum x_{1j} x_{kj} + b_2 \sum x_{2j} x_{kj} + ... + b_k \sum x_{kj}^2 &= \sum x_{kj} y_j \end{aligned} \tag{2.5}$$

The solution of the system Equation (2.5) using the principle of least squares gives the unknown parameters b_i, which corresponds to the estimated parameters $\hat{\beta}_0$,$\hat{\beta}_1$,...,$\hat{\beta}_k$.

A polynomial regression model is a special type of multiple regression model, where $x_n = x^n$. Substituting the values of x in Equation (2.3), one obtains the Equation (2.6).

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + ... + \beta_k x^k + \epsilon \tag{2.6}$$

To estimate the β_k parameters of Equation (2.6), the same approach of Equations (2.4) and (2.5) should be followed [37, 38].

2.1.3 K-means Clustering

Clustering corresponds to partition data set into groups, called clusters, in which the similarity of the elements inside the group are larger than among groups. The partition is made without a defined target, this means that it is an unsupervised learning problem. The clustering problem has been used in several applications like image segmentation, object and character recognition, information retrieval and data mining.

The K-means clustering is a partitional clustering technique that uses the squared error criteria to produce the clusters. The K-means is considered one of the simplest and most popular algorithms for clustering data [39].

Given a data set X consisting of n vectors x_j, $j = 1, ..., n$, the K-means algorithm divides it in c clusters C_i, $i = 1, ..., c$, and finds in each cluster the center o that minimizes the cost function J given by Equation (2.7).

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{j, x_j \in C_i} d\left(x_j - o_i\right) \right) \tag{2.7}$$

Often the distance d is the Euclidean distance and could be substituted in the cost function J by Equation (2.8).

$$d\left(x_j - o_i\right) = \| x_j - o_i \|^2 \tag{2.8}$$

The partitioned clusters are defined by the membership matrix U, a binary matrix in which the elements are equal to 1 if x_j belongs to the C_i cluster and equal to 0 otherwise, following the relationship given by Equation (2.9).

$$U = \begin{bmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{c1} & \cdots & u_{cn} \end{bmatrix}_{c \times n} \tag{2.9}$$

where:

$$u_{ij} = \begin{cases} 1, & \text{if } \| x_j - o_i \|^2 \leq \| x_j - o_k \|^2 \text{, for each } k \neq i \\ 0, & \text{otherwise} \end{cases}$$

After finding the elements of the matrix U, the optimal centers o_i of each cluster can be calculated by Equation (2.10).

$$o_i = \frac{1}{|C_i|} \sum_{j, x_j \in C_i} x_j \tag{2.10}$$

The algorithm could be summarized by Algorithm 2.1.

Algorithm 2.1: K-means Algorithm

Initialize the clusters centers o_i, randomly chosen from data set X
Compute the membership function U
Calculate the cost function J
Update the centers o_i
while $J < threshold$ **or** J stopped to improve **do**
    Compute the membership function U
    Calculate the cost function J
    Update the centers o_i

Although the k-means is the most popular clustering algorithm, it has some drawbacks that include the sensitivity to the initialization of the clusters' centers and the possibility of convergence to a local minimum [40]. Arthur, D. and Vassilvitskii, S. in [41] proposed a simple randomized seeding technique called k-means++ to initialize the centers of the k-means improving the speed and the accuracy of the clustering method.

Considering $D(x_j)$ the shortest distance from a data point x_j to the closest center of c_i already chosen. The k-means++ chooses an initial center o_1 uniformly at random from data set X. Then the next center o_i is chosen randomly from X with a probability given by Equation (2.11).

$$P = \frac{D(o_i)^2}{\sum_{x_j \in X} D(x_j)^2} \tag{2.11}$$

The center are chosen repeatedly using Equation (2.11) until the c centers were obtained. Therefore the Algorithm 2.1 becomes the Algorithm 2.2.

Algorithm 2.2: K-means++ Algorithm

Initialize the clusters centers o_1, uniformly and randomly chosen from data set X
for i=2:c **do**
    Chose center o_i with probability $P = \frac{D(o_i)^2}{\sum_{x_j \in X} D(x_j)^2}$
Compute the membership function U
Calculate the cost function J
Update the centers o_i
while $J < threshold$ **or** J stopped to improve **do**
    Compute the membership function U
    Calculate the cost function J
    Update the centers o_i

2.1.4 Classification Tree

A decision tree partitions recursively the input variables of a data set (also known as predictors, features or attributes) into regions, each of which is assigned a label that characterizes its data points. The

decision tree algorithm can be used to classification or regression problems.

The tree structure that represents the recursive partition has tree types of nodes: the root node, the internal nodes and the terminal nodes or leaves. The root node is where the algorithm starts and does the first partition into subsets. The internal nodes are the subsequent decisions made to split the data. Each leaf is assigned to one class representing the most appropriate label [40].

Starting from the root node, to construct a decision tree is necessary to make it grow. The tree growing means to find the appropriate split that best reduces an error measure to create an internal node. This procedure is repeated recursively until a stopping criteria is achieved: the error measured is bellow a certain tolerance value or the improvement in further splitting do not exceed a certain threshold value. For binary trees, each internal node generates two nodes [42].

Classification trees try to predict discrete classes, determining from which class the input variable belongs to. As said before, the split is chosen to minimize the error measure, often called impurity function. Considering p_i, $i = 1, 2, ..., k$, the probability of a variable from the data set belongs to the class k, the impurity function ϕ is a function that maps each input argument from the probability vector $P = (p_1, p_2, ..., p_k)$, where $\sum_{i=1}^{k} p_i = 1$, into a non-negative real number satisfying the conditions:

- $\phi(P) = 0$ if exists i such that $p_i = 1$;
- $\phi(P)$ is maximum if for all i $p_i = \frac{1}{k}$;
- $\phi(P)$ is symmetric with respect to the components of P;
- $\phi(P)$ is smooth in its range.

Therefore, the impurity function is largest when data are split evenly in all classes and is smallest when all the data belong to the same class. There are many types of impurity functions, but the best known and used in this project is the Gini diversity index [43] given by Equation (2.12).

$$\phi(P) = \sum_{i \neq j} p_i p_j = 1 - \sum_{i=1}^{k} p_i^2 \tag{2.12}$$

Reducing the error measure can result on large decision trees that are over-fitted and over-specialized to the training data set. This means that the created tree may split and classify with high accuracy the training set, but do not generalize well for new cases. It could be avoided by using pruning methods, like the impurity function, there are many techniques for pruning decision trees, it will be explained in this chapter just the one used in this project. A simple pruning technique proposed by Quinlan in [44] is the reduced-error pruning. It consists in inspecting the tree from bottom to top, considering each internal node as a candidate for pruning. For each node it checks whether replacing it by the most frequent class does not reduce the tree's accuracy. If so, the node is pruned. The procedure continues until any other node cannot be pruned.

2.1.5 Sequential Quadratic Programming

Nonlinear programming is an optimization problem formulated by Equation (2.13) in which some or all the constrains $g(x)$, $h(x)$ or the objective function $f(x)$ are nonlinear.

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & f(x) \\ \text{subject to:} \quad & g(x) \leq 0 \\ & h(x) = 0 \end{aligned} \tag{2.13}$$

The Sequential Quadratic Programming (SQP) is a method of nonlinear programming. It is an iterative procedure that models the kth iteration's solution x^k of Equation (2.13) by a quadratic programming sub-problem, and then uses this solution to find a better approximation x^{k+1} to the original problem, aiming to achieve an optimal solution x^* [45].

It is assumed that the quadratic sub-problem has linear constrains and reflects somehow the local properties of the original problem. Although it does not have the nonlinear constrains, the objective function of the sub-problem is quadratic, and thus nonlinear, reflecting the nonlinearities of the original problem.

Considering the Lagrangian function L defined by Equation (2.14), where u and v are multiplier vectors; and the Hessian H of the ψ function defined by a symmetric matrix in which the elements are given by Equation (2.15), where (i, j) represents the position of the element in the matrix. The quadratic sub-problem could be given by Equation (2.16), where B_k is the Hessian of the Lagrange function.

$$L(x, u, v) = f(x) + u^t h(x) + v^t g(x) \tag{2.14}$$

$$H\psi(x)_{i,j} = \frac{\partial^2 \psi(x)}{\partial x_i \partial x_j} \tag{2.15}$$

$$\begin{aligned} \underset{d_x}{\text{minimize}} \quad & \nabla f(x^k)^t d_x + \frac{1}{2} d_x^t B_k d_k \\ \text{subject to:} \quad & \nabla g(x^k)^t d_x + g(x^k) \leq 0 \\ & \nabla h(x^k)^t d_x + h(x^k) = 0 \end{aligned} \tag{2.16}$$

The solution $d_x = x - x^k$ of the Equation (2.16) is used to make the next iteration x^{k+1}. At each iteration, besides the update of x, a new estimation of the multiplier vectors is also calculated using Equation (2.17), where α is the step-length parameter chosen to guarantee the global convergence of the algorithm using the merit function ϕ that forces the solution $x(k)$ to approximate from the optimal solution x^* [46].

$$
\begin{aligned}
x^{k+1} &= x^k + \alpha d_x \\
u^{k+1} &= u^k + \alpha \left(u_{qp} - u^k\right) \\
v^{k+1} &= v^k + \alpha \left(v_{qp} - v^k\right)
\end{aligned}
\tag{2.17}
$$

The algorithm could be summarized by Algorithm 2.3.

Algorithm 2.3: Sequential Quadratic Programming Algorithm

Initialize the algorithm, giving values of $\left(x^0, u^0, v^0\right)$, B_0
Define the merit function ϕ
Formulate the sub-problem (Equation (2.16))
Solve the sub-problem obtaining (d_x, d_u, d_v)
Choose the step-length α satisfying $\phi\left(x^k + \alpha d_x\right) < \phi\left(x^k\right)$
Calculate $\left(x^{k+1}, u^{k+1}, v^{k+1}\right)$ using Equation (2.17). **while** x^{k+1} is different from x^k **do**
    Compute B_{k+1}
    Increment k the iteration: $k = k + 1$
    Define the merit function ϕ
    Formulate the sub-problem (Equation (2.16))
    Solve the sub-problem obtaining (d_x, d_u, d_v)
    Choose the step-length α satisfying $\phi\left(x^k + \alpha d_x\right) < \phi\left(x^k\right)$
    Calculate $\left(x^{k+1}, u^{k+1}, v^{k+1}\right)$ using Equation (2.17).

2.2 Control Method Applied

To control the soft robot presented in this project a method based on the data driven approach proposed by Holsten et al. in [36] was applied. Holsten et al. divided the workspace of the robot in sub-domains, creating a model for each one of them based on the hypothesis that the workspace could be approximated by a system of linear equations.

The method consists in five steps illustrated in Figure 2.1. It initializes with the data acquirement, collecting the shape vector parameters with the correspondent actuators inputs; with this data will be performed a clustering, diving the workspace in small regions and for each region will be trained a polynomial model that will be used to achieve desired configurations of the robot with a pre-trained tree classifier. Each one of these steps will be explained in details on the next subsections.

2.2.1 Data Acquirement

The data acquirement activity diagram is illustrated in the Figure 2.2. Although it is a not complex process, since the data collected will be used to create the models that are going to describe the robot's movements, this step should be done carefully to guarantee the high-quality measurement data. The

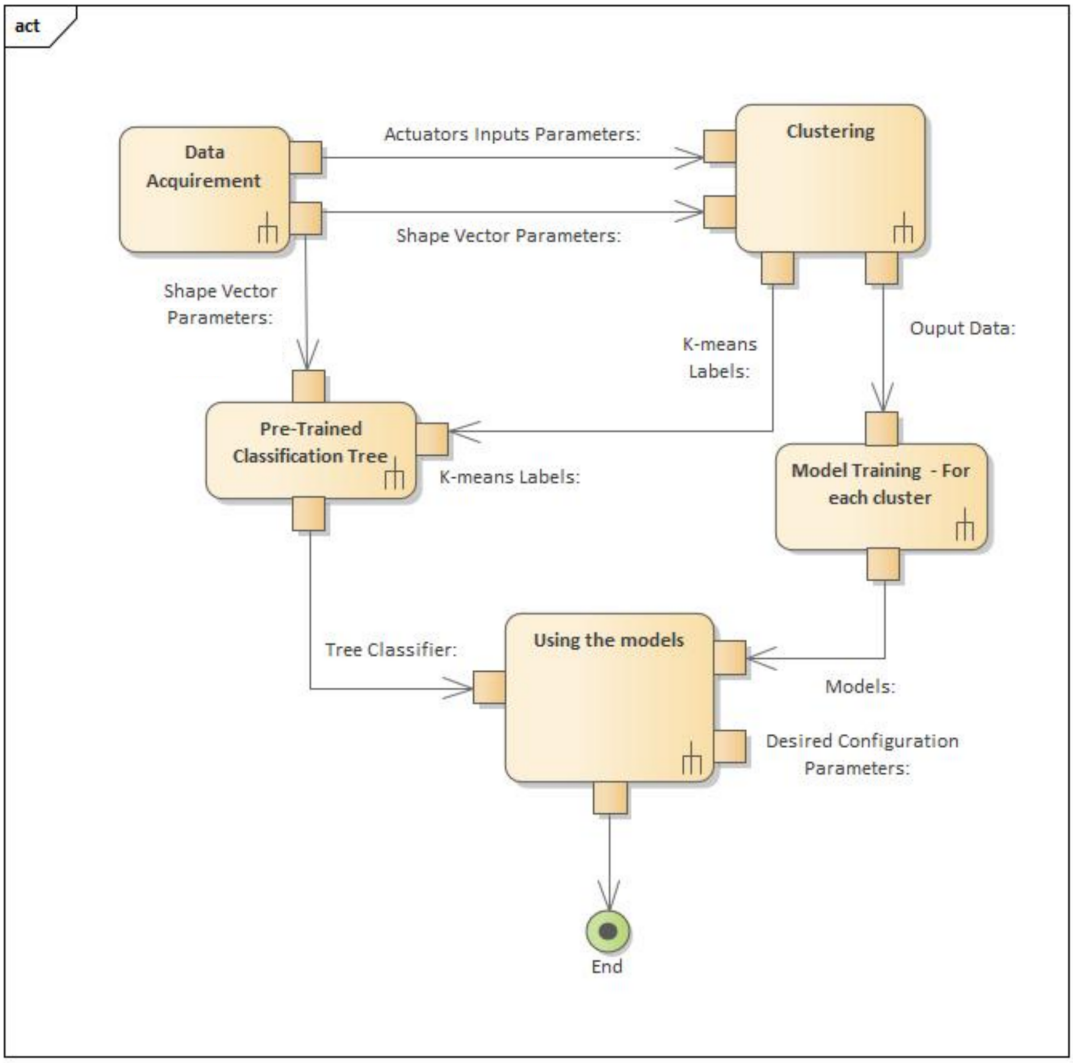


Figure 2.1: Workflow of the Method Applied

sensor used should be able to acquire the robot's position (x, y, z) in P points, called here as shape vector, given to known actuators' input parameters.

To create a large data set for training and validation of the models, the actuation space will be sampled to cover the whole space of the shape parameters, describing the workspace of the robot. Hence, for the k^{th} robot's configuration it will be extracted a shape vector s given by Equation (2.18) corresponding to the inputs α given by Equation (2.19) of the N actuators.

$$s^k \equiv \begin{bmatrix} x_0^k & y_0^k & z_0^k & \dots & x_{P-1}^k & y_{P-1}^k & z_{P-1}^k \end{bmatrix}^T \tag{2.18}$$

$$\alpha^k \equiv \begin{bmatrix} \alpha_0^k & \alpha_1^k & \cdots & \alpha_{N-1}^k \end{bmatrix}^T \tag{2.19}$$

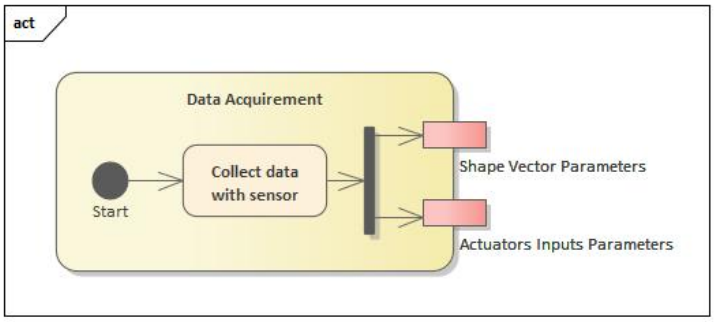


Figure 2.2: Data Acquirement Activity Diagram

2.2.2 Clustering

The clustering step will create sub-domains of the workspace allowing to create local low-ordered models instead of a global higher order model. As demonstrated by Holsten et al. in their experiments, the division is more efficient than using just one model to describe the robot's configurations, since it reduces the time complexity, the amount of data needed and increases the accuracy of the models obtained.

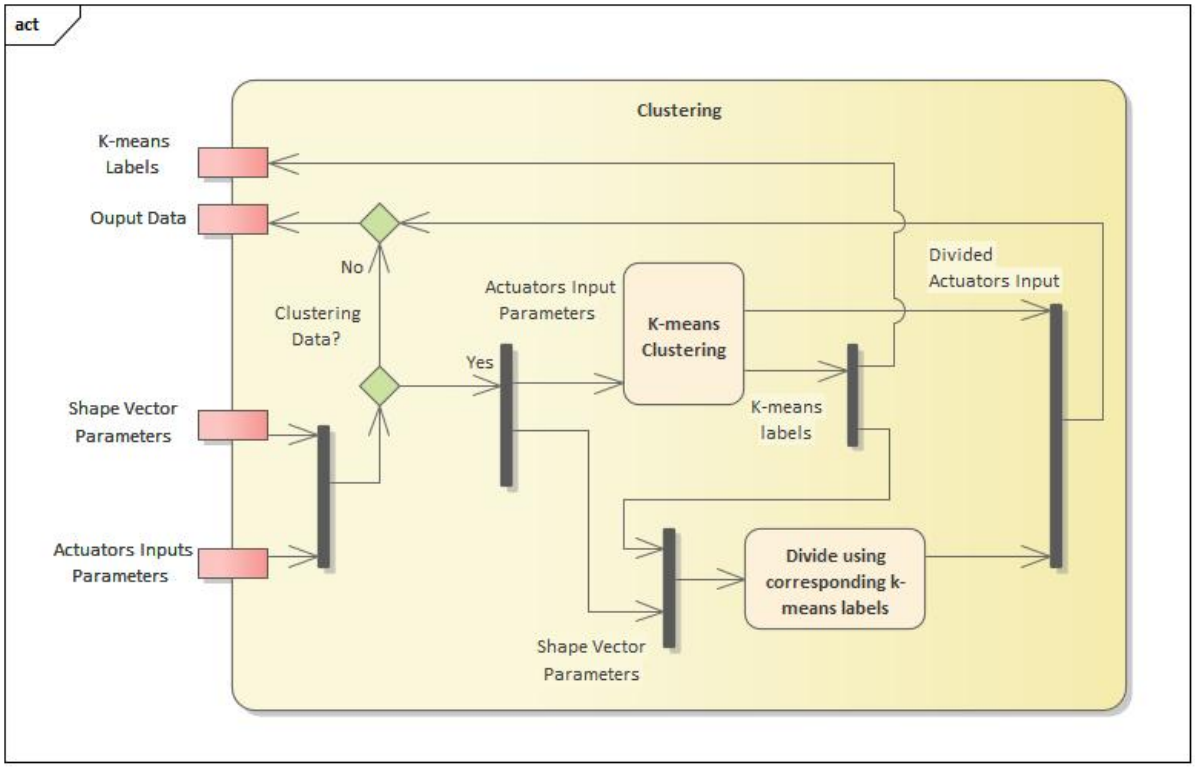


Figure 2.3: Clustering Activity Diagram

The clustering method used here was the k-means explained in Section 2.1.3. The actuators inputs of the acquired data were used as data set to apply the clustering. Then the labels obtained, that define

the partitioned clusters, were used to divide also the shape vector parameters. It was assumed that the inputs (actuators parameters) and outputs (shape vector parameters) of the soft robot were correlated to each other, i.e. for a given input the robot would move to approximately the same position regardless of where it was before. To this assumption to be valid the data acquired must be static, it means that to give the next input the robot should be stable in its current position, which defines the steady state condition.

The clustering activity diagram is illustrated in the Figure 2.3. It was used the MATLAB® function *kmeans* evaluating the distances using the Euclidean distance and the k-means++ algorithm for cluster center initialization [47].

2.2.3 Models Training

After the partition of the data, obtaining the clusters' labels, the models were built. The regression polynomial method proposed by Holsten et al. in [36] results in the direct kinematic model of the robot. As explained in Section 2.1.1, the direct kinematic model give us the relationship between the actuator space and the end-effector position of the soft robot. Using the direct model the inverse kinematic model is obtained using Monroe-Penroe inverse or quadratic programming, depending on the order of the model. In Figure 2.4 it is possible to see this procedure.

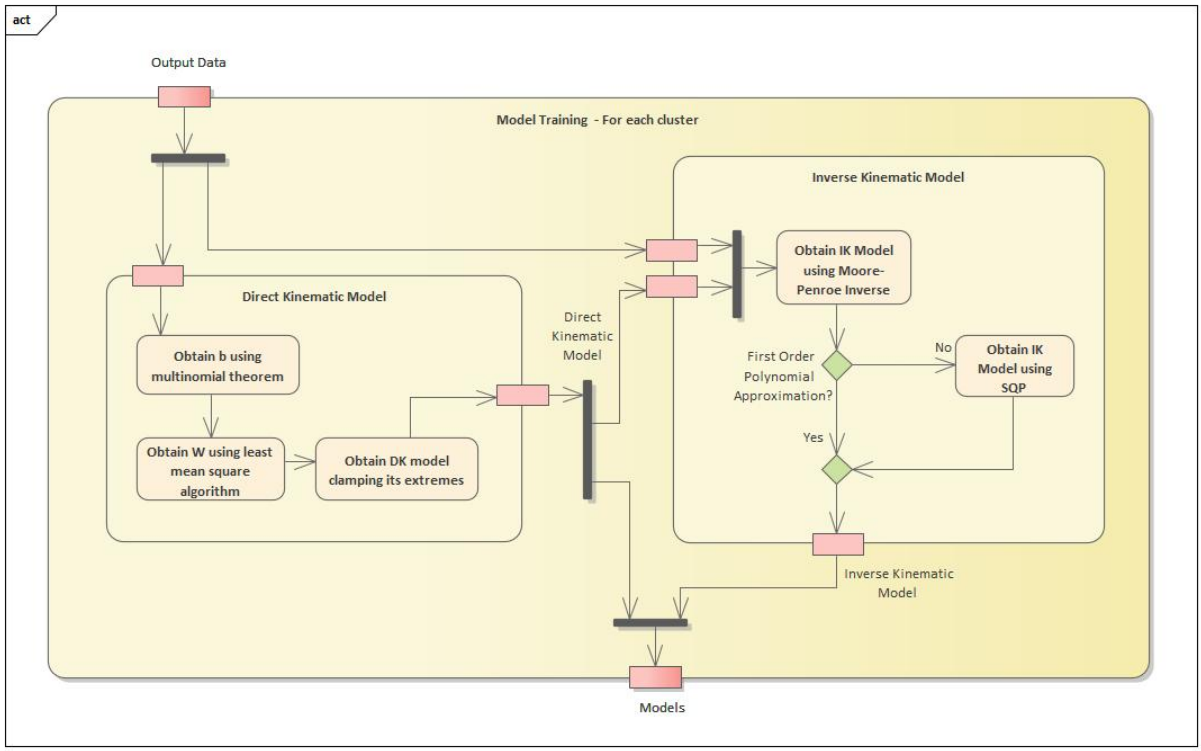


Figure 2.4: Model Training Activity Diagram

2.2.3.A Direct Kinematic Model

The direct kinematic model is built assuming that the workspace could be described by a system of linear equations given by Equation (2.20), where S is the matrix of the shape vectors s (Equation (2.18)) for the k configurations, B is the matrix of the b vectors containing all the monomials up to degree equal to the order of the polynomial approximation and W is the weight coefficients matrix.

$$S = WB \tag{2.20}$$

where:

$$S \equiv \begin{bmatrix} s^0 & s^1 & \cdots & s^{k-1} \end{bmatrix} \tag{2.21}$$

$$B \equiv \begin{bmatrix} b\left(\alpha^0\right) & b\left(\alpha^1\right) & \cdots & b\left(\alpha^{k-1}\right) \end{bmatrix} \tag{2.22}$$

Each $b\left(\alpha^k\right)$ of the Equation (2.22) is obtained using the multinomial theorem [48]. Considering the N actuators inputs α, according to the multinomial theorem, the regression polynomial of order d is given by Equation (2.23).

$$\left(\alpha_0 + \alpha_1 + \cdots + \alpha_{N-1}\right)^d = \sum \binom{d}{d_0, d_1, \cdots, d_{N-1}} \alpha_0^{d^0} \alpha_1^{d^1} \cdots \alpha_{N-1}^{d^{N-1}} \tag{2.23}$$

where:

$$d_0 + d_1 + \cdots + d_{N-1} = \sum_{n=0}^{N-1} d_n = d \tag{2.24}$$

The mathematical combination of Equation (2.23) represents the coefficients of each monomial of the regression polynomial and could be written as Equation (2.25).

$$\begin{aligned} \binom{d}{d_0, d_1, \cdots, d_{N-1}} &= \frac{d!}{d_0! d_1! \cdots d_{N-1}!} = \\ &= d! \prod_{n=0}^{N-1} \frac{1}{d_n!} \end{aligned} \tag{2.25}$$

The monomials are given by the product of α in the Equation (2.23), where d_n represents all the possible numbers of non-negative integers adding up to the polynomial order approximation d and could be rewritten as Equation (2.26).

$$\alpha_0^{d^0} \alpha_1^{d^1} \cdots \alpha_{N-1}^{d^{N-1}} = \prod_{n=0}^{N-1} \alpha_n^{d_n} \tag{2.26}$$

Using Equations (2.25) and (2.26) it is possible to rewrite the Equation (2.23) as Equation (2.27).

$$(\alpha_0 + \alpha_1 + \cdots + \alpha_{N-1})^d = \sum_{d_0+d_1+\cdots+d_{N-1}=d} d! \prod_{n=0}^{N-1} \frac{\alpha_n^{d_n}}{d_n!} \tag{2.27}$$

Dividing the Equation (2.27) by $d!$ it can be obtained the coefficients of the monomials written between 0 and 1. Calling its result as $M_{\alpha^k}^d$ and according to Equation (2.6), it is possible to write $b\left(\alpha^k\right)$ as a vector given by Equation (2.28).

$$b\left(\alpha^k\right) = \begin{bmatrix} 1 \\ M_{\alpha^k}^1 \\ M_{\alpha^k}^2 \\ \vdots \\ M_{\alpha^k}^d \end{bmatrix} \tag{2.28}$$

where

$$M_{\alpha^k}^d = (\alpha_0 + \alpha_1 + \cdots + \alpha_{N-1})^d = \sum_{d_0+d_1+\cdots+d_{N-1}=d} \prod_{n=0}^{N-1} \frac{\alpha_n^{d_n}}{d_n!} \tag{2.29}$$

Using the data acquired in the step described in Section 2.2.1, it is solved the linear system given by Equation (2.20), obtaining the weight matrix W that minimizes the value of the norm $\|(S - WB)\|$ and gives the relationship between the actuators inputs and the shape vectors.

Hence, knowing the weight matrix W and given the actuators inputs α, the shape vector $s^*(\alpha)$ can be easily estimated using the Equation (2.30), that represents the direct kinematic model of the robot.

$$s^*(\alpha) = Wb(\alpha) \tag{2.30}$$

Since it is an approximation and to guarantee security, if $s(\alpha)$ obtained by Equation (2.30) is out of the bounds formed by the maximum and the minimum values of the cluster data used for training, the $s(\alpha)$ is substituted by the nearest value between the range. This means that the function is clamped to be in the interval $[lb, ub]$, where lb is the minimum value of the shape vector inside the cluster and ub, the maximum value, following the relationship given by Equation (2.31).

$$clamp\left(s^*(\alpha), lb, ub\right) = \begin{cases} lb, & \textit{if } s^*(\alpha) \leq lb \\ ub, & \textit{if } s^*(\alpha) \geq ub \\ s^*(\alpha), & \textit{otherwise} \end{cases} \tag{2.31}$$

2.2.3.B Inverse Kinematic Model

In order to reach a certain desired position in the workspace of the robot it is necessary to obtain the estimated actuators inputs that make the robot move to this position, in other words, it is necessary to obtain the inverse kinematic model. Considering the direct kinematic model obtained in Section 2.2.3.A, the problem becomes to find the estimated actuators inputs α^* that minimizes the Euclidean distance

between the desired position s_d and the one estimated using the direct kinematic model $s^*(\alpha^*)$.

$$\alpha^* \equiv \arg\min \|s_d - s^*(\alpha^*)\|^2 \tag{2.32}$$

Observing b given by Equation (2.28), since the first element is always 1, it is possible to rewrite it as Equation (2.33). Considering ω as the zeroth ordered weights, the weight matrix W could also be rewritten as Equation (2.34).

$$b = \begin{bmatrix} 1 \\ \hat{b} \end{bmatrix} \tag{2.33}$$

$$W = \begin{bmatrix} \omega & \hat{W} \end{bmatrix} \tag{2.34}$$

Then, we could rewrite Equation (2.30) as Equation (2.35), and substitute it in Equation (2.32), resulting in Equation (2.36) where just the second term depend on the estimated actuators inputs α^*.

$$\begin{aligned} s^* &= \begin{bmatrix} \omega & \hat{W} \end{bmatrix} \begin{bmatrix} 1 \\ \hat{b} \end{bmatrix} \\ s^* &= \omega + \hat{W}\hat{b} \end{aligned} \tag{2.35}$$

$$\alpha^* \equiv \arg\min \left\| (s_d - \omega) - \left(\hat{W}\hat{b}(\alpha^*) \right) \right\|^2 \tag{2.36}$$

For first order approximations, according to Equation (2.29), $\hat{b}$ corresponds simply to the actuators estimated inputs, therefore Equation (2.36) becomes a linear least square problem that could be solved through the Moore-Penrose pseudoinverse of $\hat{W}$ ($\hat{W}^+$), giving an unique solution to the linear system [49] given by Equation (2.37).

$$\alpha^* = \hat{W}^+ (s_d - \omega) \tag{2.37}$$

Again, a clamping function, as the one used to obtain the direct kinematic model (Equation (2.31)), was used. The lower and upper bounds were now defined as the minimum and maximum values of the actuators inputs inside the cluster that the model is being created.

For higher order approximations, the Equation (2.36) is nonlinear and an optimization method is convenient to be used. Since the function is quadratic, following the approach used in [36] by Holsten et al. and considering that, according to Boggs and Tolle in [46] the algorithm is the most successful method for solving nonlinear constrained problems, it was used a sequential quadratic programming described in Section 2.1.5 to solve it.

Using the function from MATLAB®, *fmincon*, the constrains were defined as the lower and upper

bounds previous described in this section; and as initial conditions it was given the model obtained using Equation (2.37). The merit function used by the MATLAB® function was proposed by Han and Powell [50] and it is given by Equation (2.38).

$$\phi(x) = f(x) + \sum r \cdot g(x) + \sum r \cdot \max[0, h(x)] \tag{2.38}$$

where,

$$r = r_{k+1} = \max\left\{u, \frac{r_k + u}{2}\right\} \tag{2.39}$$

2.2.4 Pre-Trained Classification Tree

With the labels obtained in the clustering step described in Section 2.2.2 and the acquired shape vector parameters (Section 2.2.1), a classification tree is trained to predict the labels of the tested shape parameters (Figure 2.5).

This procedure was made using the MATLAB® function *fitctree*, training a binary tree classifier using the Gini diversity index as impurity function and the reduced-error pruning, both of them explained in Section 2.1.4 [47].

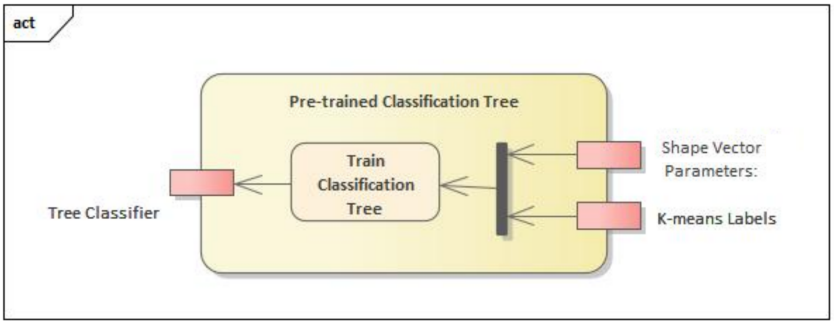


Figure 2.5: Classification Tree Description

2.2.5 Using the Models

In order to use the models obtained in Section 2.2.3, Holsten et al. in [36] adopted two methods: a regression tree classifier and a minimum solution and compared both of them.

The minimum solution method consists in, given the desired shape vector parameters, obtaining the predicted actuator inputs for all the inverse kinematic models trained, this means that the results will be obtained for all the clusters. With the predicted actuators inputs, using the corresponding direct kinematic models, the estimated shape vector parameter is obtained. Finally, comparing the estimated shape vector parameters with the desired ones, the model that minimizes the differences between them

is chosen. This method is time consuming, since it has to test all the models to choose the one that best describes the desired position of the robot and for higher order models it could became really slow.

The second method proposed uses the pre-trained classification tree described in Section 2.1.4. The classification tree will choose one optimal local model to predict the actuators inputs. Although, this method is faster, according to Holsten et al. it is less accurate due to the possibility of misclassifications of the tree classifier.

The description is showed as scheme in Figure 2.6.

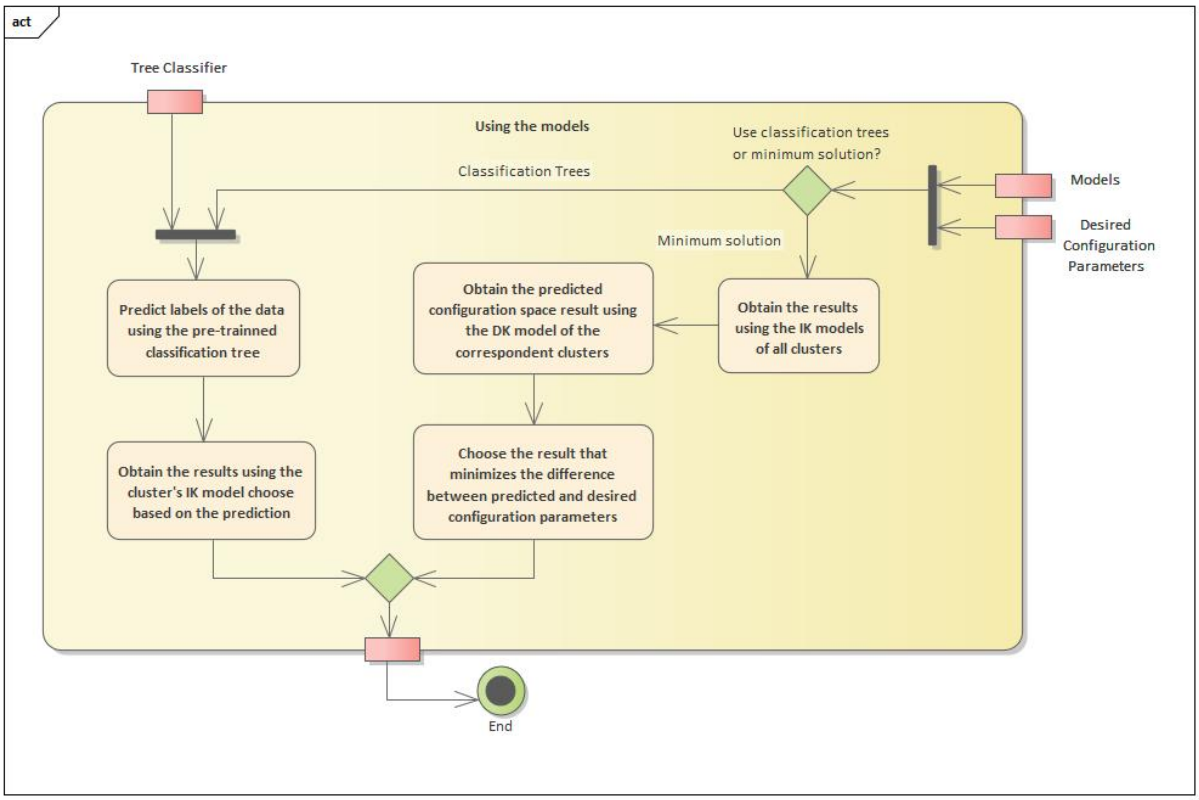


Figure 2.6: Scheme of how the models are used

2.2.6 Considerations of the Method

To ensure that the method works properly, the B matrix in the linear system of the Equation (2.20) must be a full rank matrix. This condition makes it possible to obtain the weight matrix W, one of the primary steps to get the direct kinematic model. In order to have a full rank matrix, the number of linearly independent columns of B must be greater than or equal to the number of its rows.

The number of columns of B is determined by the number of acquired robot's configurations k. The number of rows is given by the sum of the number of terms of each multinomial that composes the vector $b\left(\alpha^k\right)$ given by Equation (2.28) plus one, which represents the first term 1 of the vector. The number

of terms of the multinomials are determined by the Equation (2.24) that corresponds to all the possible combinations of N integer numbers whose sum is equal to the degree of approximation d, which can be calculated by the combination of Equation (2.40) following the stars and bars theorem [51].

$$\textit{Number of terms of } M^d_{\alpha^k} = \binom{N+d-1}{d} \tag{2.40}$$

Therefore the minimum number of acquired data k_{min} is given by Equation (2.41), where N is the number of actuators.

$$k_{min} \geq \left[\sum_{i=1}^{d} \binom{N+i-1}{i}\right] + 1 \tag{2.41}$$

In polynomial regression, increasing the variability of the data makes it necessary to use polynomials of higher orders to approximate them. Since polynomial regression is used here to obtain the direct kinematic model of the robot, if the shape configurations have great variability, to describe the workspace of the robot it is necessary to use more complex polynomials. According to Equation (2.41), increasing d means to increase a lot the number of samples that must be acquired.

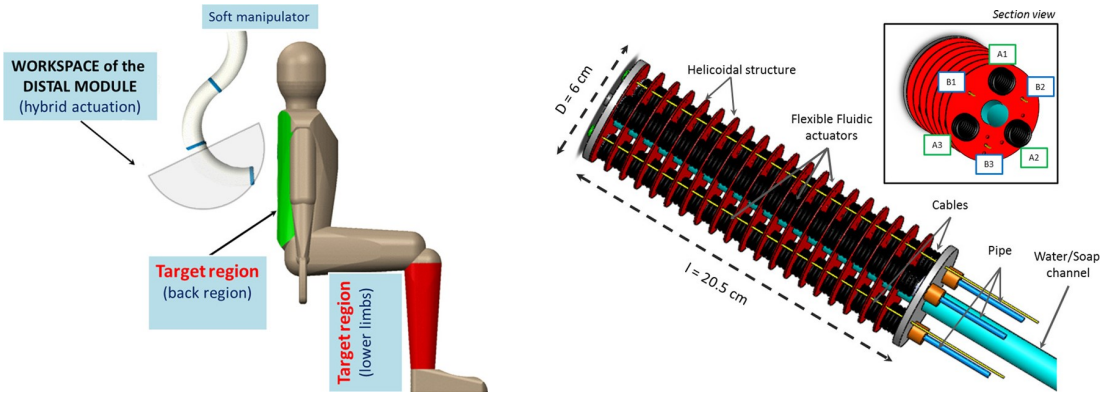
Since collecting data is time consuming, following the same approach used by Holsten et al., it was divided the workspace of the robot using clusters which partitions the workspace into regions with smallest variability allowing to have lower order polynomials describing it. This approach is possible if every cluster have at least the minimum number of observations given by Equation (2.41).

As explained in Section 2.2.2, the results of the k-means clustering method used in this project depends on the initialization of the clusters, knowing that, if one of the clusters do not satisfied the minimum number of samples, the algorithm divides the data again. It was established that, if after 30 times trying to find a configuration that allows every cluster to have the minimum number of samples, the algorithm reveals that it is not possible to divide the data in the clusters with that specific polynomial order.

2.3 I-SUPPORT System

As said before, in Section 1.3, the I-SUPPORT system is composed by a motorized chair and two soft robotic arms: one for provide pouring water and the other for scrubbing [35]. In order to achieve the desirable workspace, that should reach the target regions of the users' body during the bath tasks, the robotic arms are made of three interconnected modules (Figure 2.7(a)). Although the modules have the same design, the proximal one is only actuated by cables and the middle and distal ones are identical and based on hybrid actuation, a combination of cables and fluidic actuators, more specifically, pneumatic ones [52].

The combined actuation ends up increasing the overall functionality of the complete module, because it allows the actuators to complement each other regarding their limitations. The use of the cables permits multiple functionalities, such as the compensation of gravity effects and shortening, but it requires a higher number of modules to cover the same workspace; on the other hand, the use of the fluidic actuators enables elongation performances, maximizing the total length of the arm and reducing the number of modules, although it is unstable due to gravitational effects.



(a) Overview of the soft robotic arm of the I-SUPPORT System.

(b) CAD design of the single module. In the section view is possible to see the fluidic actuators (A1, A2 and A3) and the cables (B1, B2 and B3).

Figure 2.7: I-SUPPORT [32].

Since the success of the manipulator is very dependent on the functionality of the hybrid modules, they are the focus of the works in [32, 33, 52]. The most recently developed module prototype [33] is composed of three pneumatic actuators and three cables alternatively displaced at an angle of 60° in a circle of 60mm diameter. The 140mm diameter central channel in the circle is responsible for providing water and/or soap (Figure 2.7(b)). All the actuators are decoupled, meaning that the cables have dedicated lines for tension regulation as the pneumatic actuators have for pressure. The total length of the module is 150mm and the total weight is 120g.

The McKibben actuators are the artificial pneumatic muscles most frequently used at present, their design is characterized by a inner tube cover by a braid and connected to fittings at both ends [53]. The pneumatic actuators used by [32, 33] are McKibben-based. The braid has a bellow-shaped surface, increasing the elongation performances, and the inner tube is a balloon made of latex silicone rubber.

To guide the bending and, consequently, the application force of the soft robotic arm during its motion, in [32], a flexible helicoidal structure has been inserted along the module, hosting the six actuators and also the water/soap channel. However this structure produced undesirable torsional movements, limiting its bending capabilities and in [33] it was replaced by a layer-by-layer reinforcement structure, in which

the layers are distanced by 10mm from each other.

The module built by [32] accomplished satisfactorily the reachability necessary to the user workspace, covering larger distances with a reduced number of modules, and showed a stiffness variation, guaranteed by the hybrid actuation, to compensate internal/external loading in the bathing environment.

However, since the actuator are manufactured in a bellow shape using manual mechanical deformation, this procedure may result in an asymmetric shape not easily identified when the module is at rest. The friction between the internal chamber and the external braid during the actuator function may cause easily the rupture of the latex balloon, the replacement of it imply in the entire construction of a new actuator again, representing a time consuming maintenance.

The McKibben-based I-SUPPORT was used to generate data for the Pre-Experimental (Section 4.1) and Experimental (Section 4.2) phases of this project. For the Experimental tests, the data was acquired using one module of the I-SUPPORT. Its inputs are digit numbers from 0 to 255 that corresponds to a given pressure for the pneumatic actuators [54] and as output the end-effector position in the sensor reference frame.

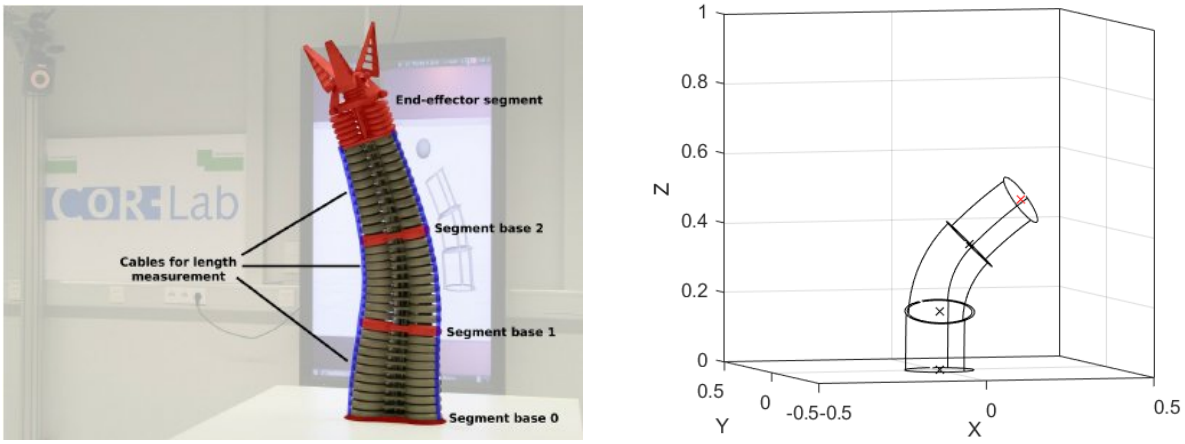
The Pre-Experimental data was provided by The BioRobotics Institute from Sant'Anna School of Advances Studies. It was provided data sets from two different assemblies of the types of I-SUPPORT: using just one module and another using two modules of the robot. The inputs of the single module robot are random digit numbers from 0 to 255, representing the pressures supplied to the actuators, and the output is the end-effector position in the used sensor reference frame. The set comprises 9996 samples. In the two module version of the robot, the inputs are cable lengths and the outputs are the four end-effector markers position in 3D space with respect to the used sensor reference frame. The set comprises 3064 samples.

2.4 Bionic Handling Assistant Model

The Bionic Handling Assistant (Figure 2.8(a)) is a continuum platform developed by Festo. Its design was inspired by an elephant trunk and consists of three segments with a slightly conic form, each one actuated by three pneumatic chambers supplied with compressed air, and three further actuators arranged around a ball-joint wrist constitute the gripper. The BHA has, in total, 11 DoF [55].

Rolf and Steil in [6] proposed a kinematic model for the BHA. Using the CC approximation for modeling the robot's deformations as modified toroidal deformations, they achieved stable results even close to the model's singularities. The model (Figure 2.8(b)) is composed of three cylindrical segments, which from the base have their radii reduced. Although the model's geometry assumes that each segment has an equal radius along its length, the model showed to be a good representation of the BHA robot, presenting only 1% of error comparing the model's prediction with real-world motion data recorded with

a tracking system, including circular bendings and elongations.



(a) Actuation and sensing of the BHA [6]

(b) BHA model proposed by Rolf et al. in [6]

Figure 2.8: Bionic Handling Assistant

The kinematic model includes the three segments of the BHA, taking in as input the length, from 0.1 to 0.3m, of each one of the nine cable potentiometers connected to the actuators. With an offset of 0.148m in z-direction from the last segment, the end-effector position with respect to a reference frame fixed at the origin is given as output.

Due to the similarities with the I-SUPPORT prototype proposed in this project, regarding the number of segments of the robotic arm, the pneumatic actuators and the fact that both are continuum robots, the BHA model developed by Rolf and Steil in [6] was chosen to obtain the training data during the Pre-Experimental phase (Section 4.1). Since in this project it was used just one module of the I-SUPPORT, the BHA model was reduced to only one segment, to be applied the polynomial method described in Section 2.2.

2.5 Experimental Setup

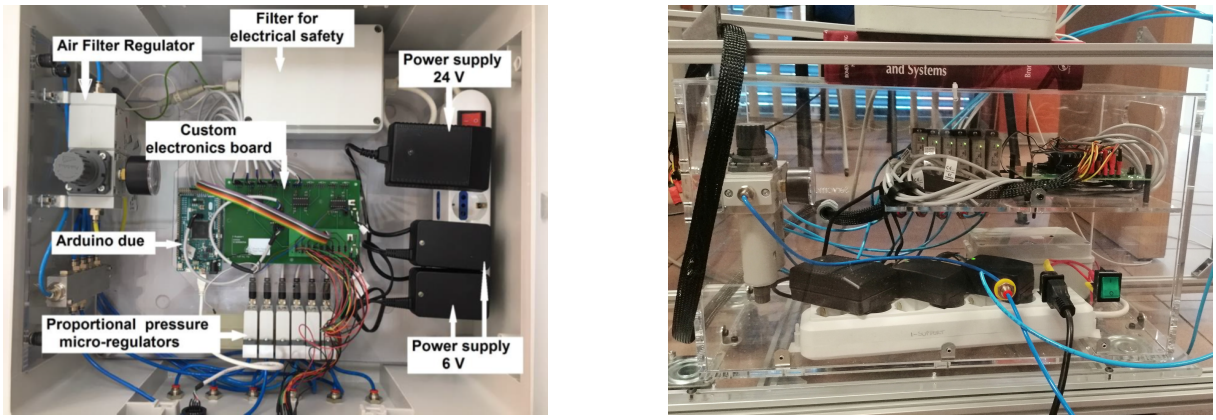
2.5.1 Pneumatic Set

The experiments described in Section 4.2 were developed using the pneumatic set available in The BioRobotics Institute of Sant'Anna School of Advanced Studies [54]. It consists in three main parts that can be seen in Figure 2.9(a): the pneumatic actuators, the power supply and the control unit.

Inside the box there are six proportional pressure micro-regulators (Camozzi, mod. K8P-0-E522, 0-3bar, 0-10V) and one filter regulator to set pressure input at 4bar, that consist the pneumatic actuation. For the following experiments it was used only three micro-regulators, since only one module of

the I-SUPPORT was actuated. The air source is a silent compressor (Figure 2.10) with performance of 0.19kW and intake capacity of 25L/min and maximum pressure of 8bar. During the experiments performed in this project the air pressure of the compressor was set at 3bar.

The power supply is constituted by one class II power supply (24V) for the six micro-regulators, two class II power suppliers (6V) for the six servomotors, which were not used in this project, and a filter box which guarantees electrical safety consisting of a filter (6EHT1 CORCOM) and a fuse (FUSES FSF 6.3x32 - Non resettable fuses).



(a) Components of the pneumatic set [54]

(b) Pneumatic acrylic box

Figure 2.9: Pneumatic Set

The control unit is composed by an Arduino Due and a custom electronics board to connect the Arduino with the pressure micro-regulators and the servomotors. The board and the Arduino are connect via SPI communication for the micro-regulators and directly to Arduino digital pins for the servomotors. The board also host a DAC (TLC5628) and 3 quad operational amplifiers (LM324-N).

Figure 2.10: Compressor used as air supplier

The design of the pneumatic set were thought to avoid electric issues and respect minimal safety requirements, this also includes to host all the components inside the acrylic box of Figure 2.9(b) ensuring

electric insulation. The pneumatic box is placed in the support frame structure that will also uphold the robot through its base (Section 3.1.1).

2.5.2 Sensors

During the experiments described in Section 4.2 it was used two sensors to capture the end-effector position of the robot. The characteristics of them and how the data should be acquired in each one will be described in Sections 2.5.2.A and 2.5.2.B.

2.5.2.A SMART-DX System

Figure 2.11: SMART-DX motion system used during the experimental tests

From BTS Bioengineering company, the SMART-DX motion tracking system [56] is a passive optical motion capture, this means that it uses retroreflective markers that are tracked by infrared cameras. Using epipolar geometry between the cameras, the markers are reconstructed in 3D space. According to the manufacturer, the system could reach 0.1mm of accuracy, however the actual accuracy depends on location of the markers, cameras and the respective calibration. The data is acquired with a frequency of 250Hz.

The setup used in this project (Figure 2.11) consists of six cameras placed around the support frame

structure where the robot was uphold with three markers attached to its tip. The data acquired with this setup was used to perform the analysis regarding the control described in Section 4.2.3.B. Since the data is acquired with a high frequency rate, each input result in a step response of the system, thus to use the data for the analysis performed it was necessary to pre-processing the data, capturing just the steady state position of the robot.

2.5.2.B Aurora System

The Aurora System [57] is an advanced electromagnetic spatial measurement system that calculates the position and the orientation of a six DoF embedded sensor within a defined volume. It presents a high degree of accuracy in an environment free of electromagnetic disturbances of 0.70mm for position and 0.30 degrees for orientation (Root Mean Square). The system is composed of three main parts: the planar field generator, which generates the magnetic fields; the system control unit, which controls the operation of the Aurora systems, calculates the position and orientation of the sensor and connects the field generator with the computer through an USB cable; and the sensor interface unit, connecting the embedded sensor and the control unit and responsible to convert the analog signals to digital ones.

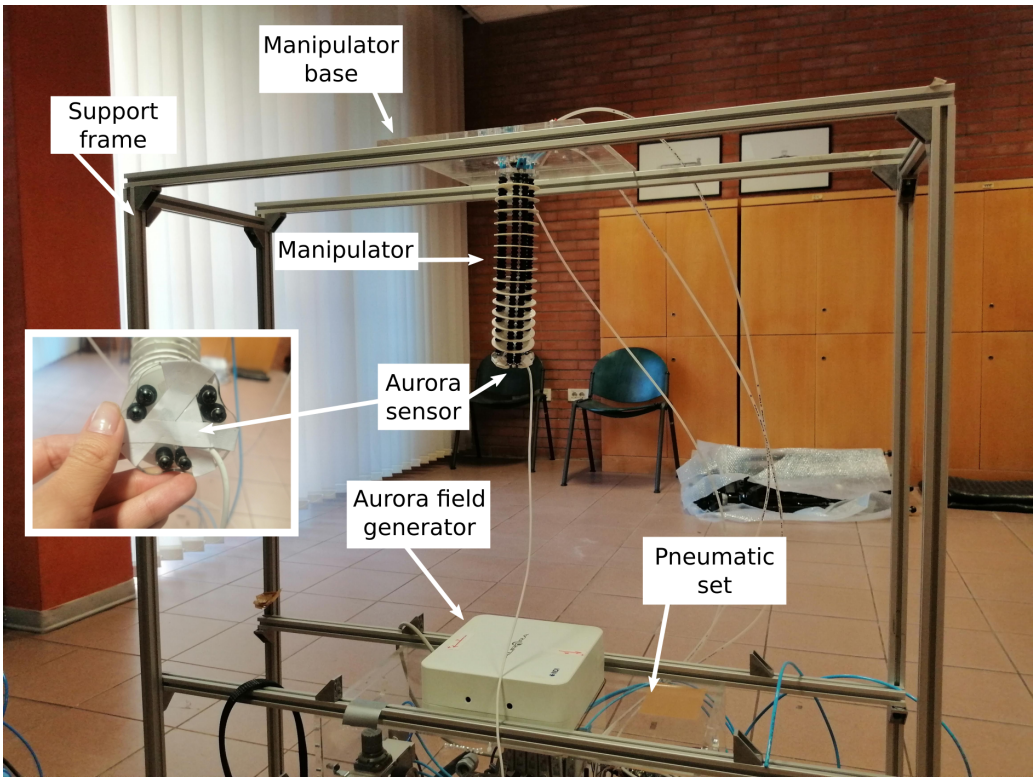


Figure 2.12: Aurora system used during the experimental tests

The sensor was attached at the end of the manipulator as it is possible to see in Figure 2.12. It was created a MATLAB script to acquire the robot's position and orientation after some seconds after the

given input, so different from the data acquired from Smart-DX System, the outputs were the positions of the robot already in the steady state. All the experiments, except the ones described in Section 4.2.3.B, were performed using the setup of Figure 2.12.

3

Prototype Description

Contents

The proposed robotic (Figure 3.1) arm follow the same structure of the most recently developed module prototype of the I-SUPPORT [33] described in Section 2.3, being composed of three distinct parts: the base, the guiding structure and the actuators. These parts will be described in more details in this Chapter. The final assembly of the proposed robot in this project will be seen at the end of it.

Figure 3.1: Proposed robotic arm in a bending configuration

3.1 Components

3.1.1 Base

A rigid rectangular base (Figure 3.2) made of Plexiglas is used to place the robot in the support frame structure, allowing its movements, and to uphold the pneumatic connectors and the cable set-up.

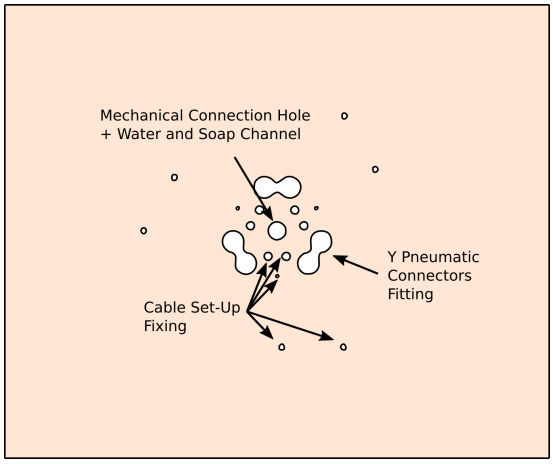


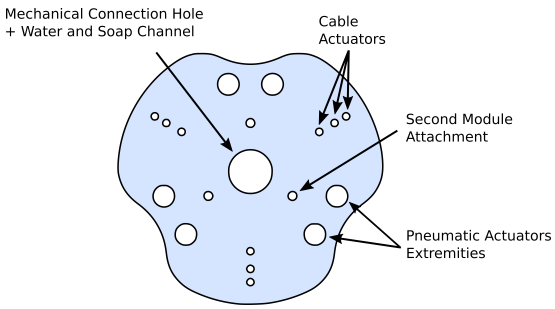
Figure 3.2: CAD design of the manipulator base

In this work the hybrid actuation will not be used, this means that only pneumatic actuators will be adopted, but the base has the holes to fix the cable set-up. These holes leave open the possibility for future works to add the cable actuators without changing the base. The cable set-up fixing measurements were based on the cable actuators used by Ansari et al. in [33].

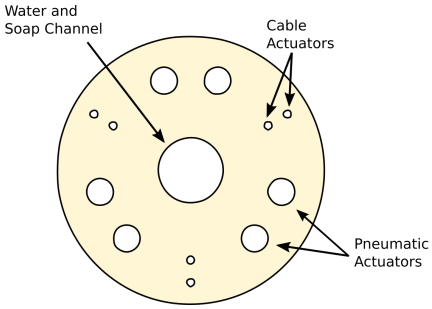
A central hole of 10mm was inserted to facilitate the flow of water and soap, but it will also be used as a mechanical connection between the manipulator and its base, that will be detailed on Section 3.2.

3.1.2 Discs

There are two types of discs in the robot manipulator performing different roles. On the proximal and distal extremities of the pneumatic actuators, a Plexiglas disc of 5mm thickness (Figure 3.3(a)) is attached. In the proximal, the central hole, besides of working as a water and soap channel, will hold the mechanical connection with the base. Aiming for future works, the holes for the cable actuators and the holes to add a new module of the manipulator were kept. Therefore, these discs act mainly as connectors between the parts of the prototype.



(a) CAD design of the proximal and distal discs



(b) CAD design of the central discs

Figure 3.3: CAD Discs

Additionally, fifteen central discs, as the ones showed in Figure 3.3(b), of 1.2mm thickness polypropylene were placed equally spaced between the actuators act as a reinforcement structure. They are responsible to lead the application of the force of the manipulator through the guiding of the bending during its motion.

3.1.3 Actuators

The main difference between the proposed robot and the more recently one developed by Ansari et al. in [33] rely on the pneumatic actuators. Instead of McKibben-based, this work proposes the use of silicone bellows as actuation system, trying to increase the reliability and facilitate the maintenance of the robot.

It was tried to substitute the actuators without making substantial changes regarding the dimensions of the robot. In the retail market, the only silicone bellows found available with 10mm of external diameter and uniform shape was 10cm length, approximately half of the length of the one module I-SUPPORT robot. An unit of the silicone bellows used can be seen in Figure 3.4.

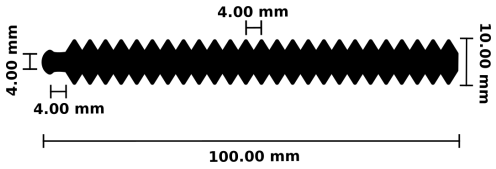


Figure 3.4: Silicone Bellows Dimensions

Due to the size of an unit of the silicone bellows, to build the robot's actuators, it was necessary to join two units of them with Sil-Poxy^{TM} silicone glue (Figure 3.5). In order to increase the glued surface area, one small cut was made in each one of the silicone units: one of them was cut in the middle of the first bellow, right after the first bigger diameter; and the other one, was cut in the extremity, allowing the properly fit between them.

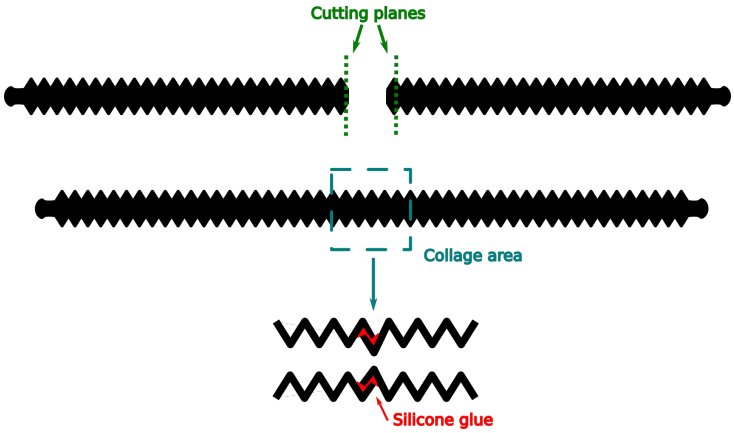


Figure 3.5: Actuator building scheme

The silicone glue provides a strong, flexible bond between silicone parts with high elongation. Besides the fast cure and durability, it makes the connection between the bellows units uniform and smooth. It can also be used for repair of the silicone actuators, which makes the maintenance really easy.

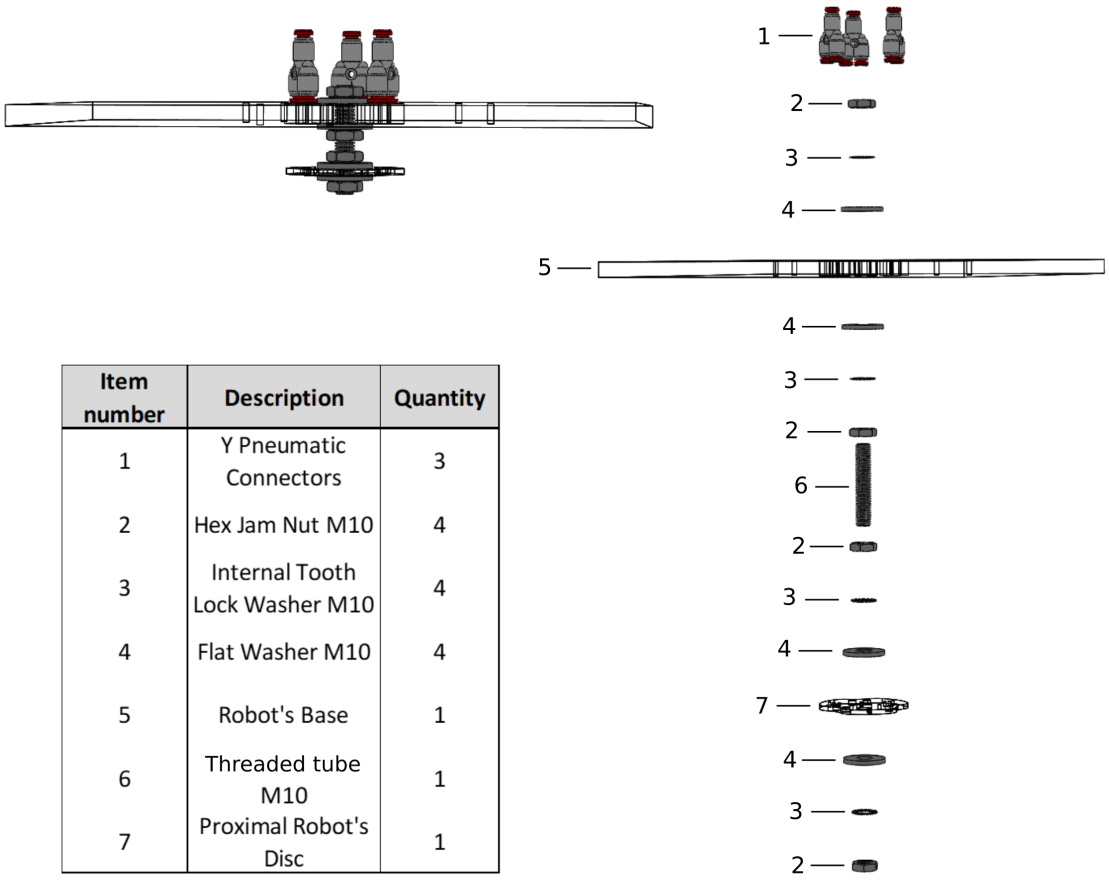
3.2 Assembly

3.2.1 Mechanical Connection

The connection, before made only with the Y connectors and pneumatic tubes, will be made using a threaded tube, locked with washers and nuts. The mechanical connection offers more reliability since

the robot's weight will be hold by it and not by the linkage between the tubes inside the Y connectors.

In Figure 3.6 is possible to see how the connection was made. The threaded tube is inserted in the central holes of the base and the proximal disc and hold by a nut - tooth washer - flat washer set, where the tooth washer prevent the nuts from backing out and the flat washer distribute the load in the base and the disc.



Item number	Description	Quantity
1	Y Pneumatic Connectors	3
2	Hex Jam Nut M10	4
3	Internal Tooth Lock Washer M10	4
4	Flat Washer M10	4
5	Robot's Base	1
6	Threaded tube M10	1
7	Proximal Robot's Disc	1

Figure 3.6: Details of the Mechanical Connection

3.2.2 Final Assembly

The actuators are inserted in the holes designed in the polypropylene discs, letting them equally spaced. The extremities of the chambers are inserted in the acrylic discs. The distal extremity's holes are closed and in the proximal extremity's ones the pneumatic tubes are inserted and connected to the Y connectors.

The acrylic pneumatic box described in Section 2.5.1 has four outside connections. The power supply plug is connected into an outlet. To each of the three proportional micro regulators a pneumatic tube is attached. The other extremity of the tube is coupled to a pair of silicone actuator by the Y connectors. The USB connection communicates with the computer, which will make it possible to control the manipulator. The compressor is connected to the air filter regulator by a pneumatic tube, and also is plugged into an outlet. The final assembly described can be seen in Figure 3.7, for simplicity, it is only showed one proportional micro regulator connected to a pair of actuators, the other two are equally connected to the filter, the custom board and a pair of actuators.

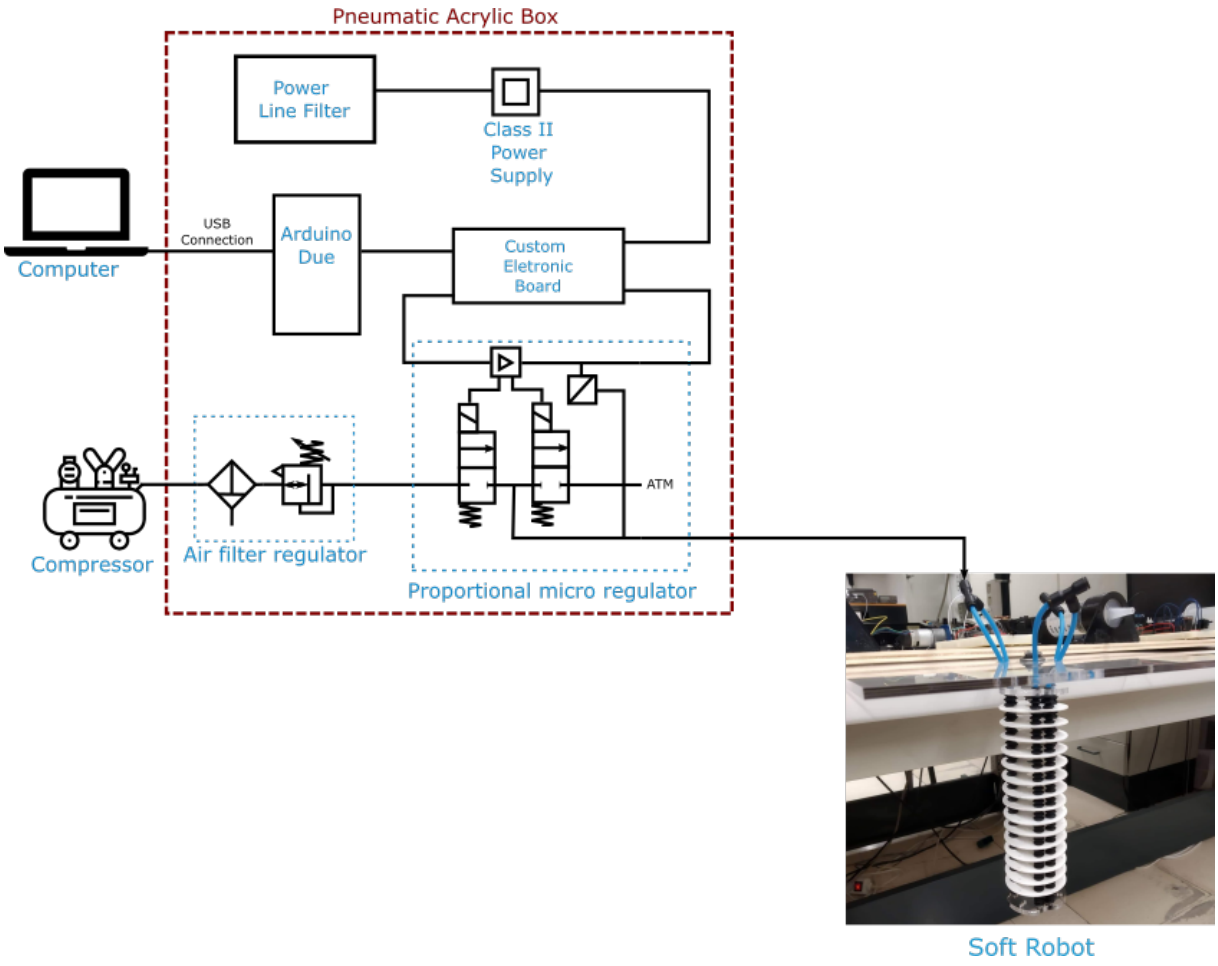


Figure 3.7: Final Assembly

4

Results

Contents

4.1 Pre-Experimental Results

The method to obtain the inverse and the direct kinematics models of the robot described on Chapter 2 was based on the approach proposed by Holsten et al. in [36]. Although they claim that the data-driven method needs no prior knowledge about the robot, it was made a pre-experimental phase using two types of data that are similar to the proposed robot: using a simulation model of the BHA [6] robot and using data from the McKibben-based I-SUPPORT robot provided by The BioRobotics Institute from Sant'Anna School of Advanced Studies. In this pre-experimental tests the goal was to acquire some sensibility of the method and observe its behaviour in I-SUPPORT like robots.

4.1.1 Bionic Handling Assistant Model

As explained in Section 2.2.1, the amount of data acquired should be sufficient to describe the workspace of the robot. Ansari et al. in [32] described the reachable workspace of their I-SUPPORT prototype using 8000 samples, considering that the main difference between their robot and the one proposed in this project rely on the actuators, it was chosen to use the same amount of data.

The inputs and outputs obtained with the BHA model were used to generate the inverse and direct kinematics models of the robot. It was created 2 types of inputs to give to the BHA model:

- randomly generated from 0.1 to 0.3m;
- permutation of 20 points equally spaced between 0.1 and 0.3m;

Since the selection of data for training has direct influence on the results, it was used as pattern to all the tests performed in the Section 4.1 to do 10 times the same test, selecting randomly the percentage of data determined for training. Therefore the results presented are the mean error between the ten results using the same parameters and different data for training. In this way, it can be analysed the combination of number of cluster and degree of the polynomial that gives the best approximation to describe the robot independently of the data selected.

Figure 4.1 shows the direct kinematics model errors between the predicted and the real lengths for the training and validation set of data using one cluster, varying the polynomial order. It was used 70% of data for training, achieving errors with magnitude of 10^{-5} using one polynomial of fourth order and even lower increasing the degree of the polynomial as one may see in Figure 4.1. It was noticed that although the training error keeps decreasing for higher orders, the validation error stops to decrease at 15th order, probably indicating data overfitting. The standard deviation between the 10 rounds of training, almost zero before, begin to increase with a fifth order model. This results were similar for all the three types of input data tested.

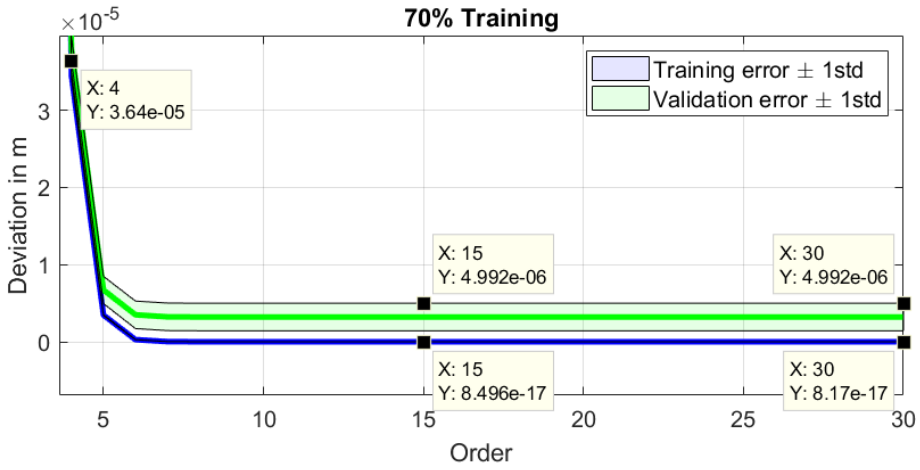


Figure 4.1: Direct kinematics model errors of BHA permutation data without noise with one cluster

The extremely low errors with magnitude of $10^{-5} \sim 10^{-6}$ with almost no variation between the rounds of training happened because the BHA model is obtained from toroidal approximations of the robot's movements, a relatively simple mathematical approximation, which reduces an infinite-dimensional structure into a 3D structure and could be easily described by polynomials. In order to have a better simulation of the behaviour of the algorithm with real world acquired data, it was added a Gaussian white noise in the output of the BHA model as illustrated in Figure 4.2, estimating that 94% of the amount of data acquired is within 6mm error due to noise.

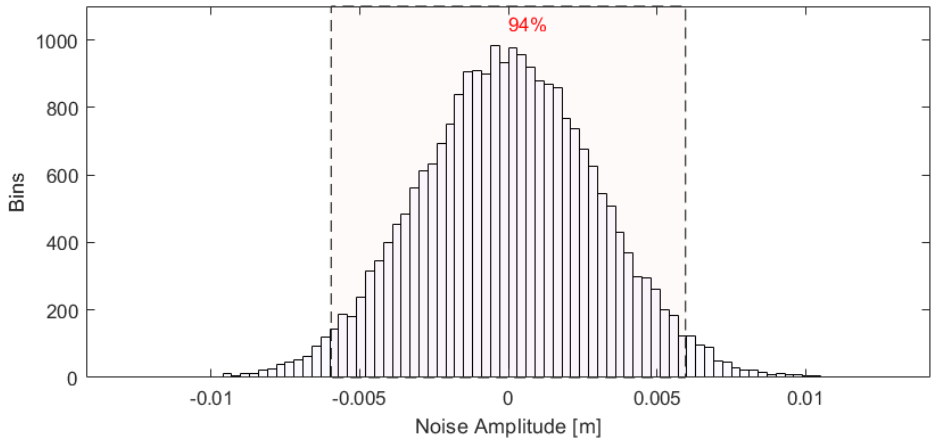
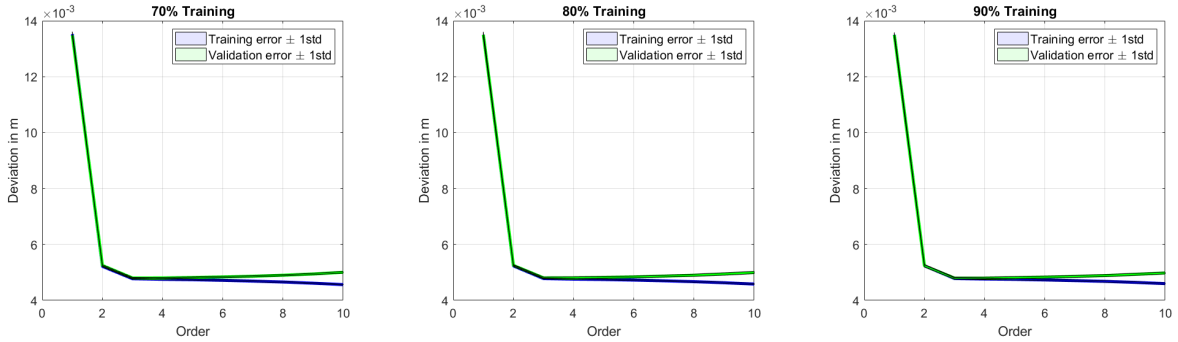


Figure 4.2: Gaussian white noise added to the BHA output data

The first analysis was done to assess the percentage of data for training and validation that should be used. In Figure 4.3 it is possible to see the results of the test performed to obtain the direct kinematics model using only one cluster and the random data with noise added varying the percentage of data for training using. The variation of the training percentage does not decrease the minimum error, obtained

with a polynomial of third order in all tests. The standard deviation also does not present significant changes regarding the validation and the training errors. However there was an increase of computation time with the increase of data for training and therefore it was chosen 70% of data for training in all following tests.



(a) 70% training **(b)** 80% training **(c)** 90% training

Figure 4.3: Variation of the percentage of data for testing and training

The second analysis was carried out to verify what happen when the model is inverted, if any information could be lost in the operation. It was tested several combinations of order and clusters for both types of inputs created. In Table 4.1 one may see some validation error results for the direct and inverse kinematics model with some of the combinations tested.

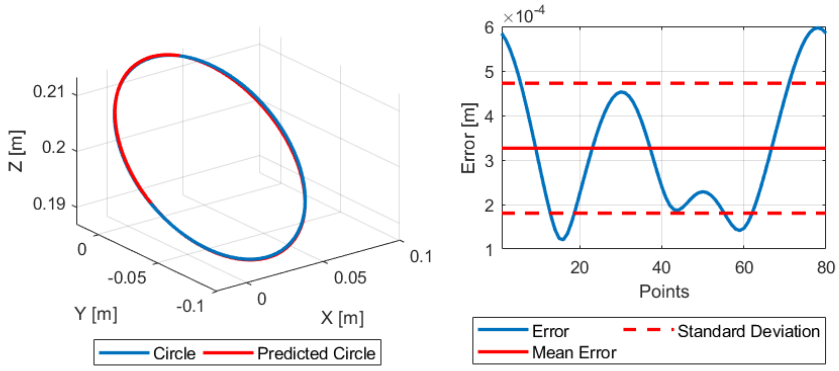
Table 4.1: Validation mean error and standard deviation for the direct and inverse kinematics models using some combinations of order and clusters

		Direct Kinematics		Inverse Kinematics	
	Clusters	**Order**	**Error [mm]**	**Order**	**Error [mm]**
Random	1	3	4.81 ± 0.01	3	6.96 ± 0.04
Random	2	3	4.81 ± 0.01	2	7.04 ± 0.05
Permutation	1	4	4.77 ± 0.02	4	6.84 ± 0.04
Permutation	2	3	4.77 ± 0.02	3	7.07 ± 0.13
Permutation	3	3	4.79 ± 0.02	3	7.00 ± 0.08

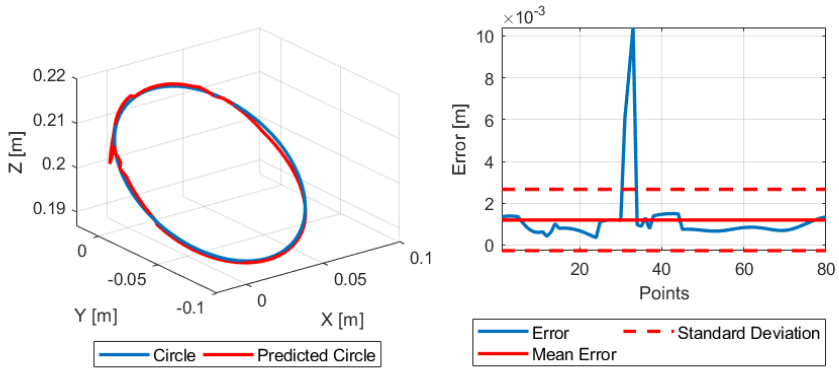
It was noticed that for a given number of clusters, the minimum error for both kinematic models were given by the same order of polynomial. For example, using the random input data, if the workspace of the robot is divided by two sub-spaces (or two clusters), the polynomial that better describes the regions for the space vector and the actuators parameters is given by a 3rd order polynomial. This means that a straight relationship between orders and clusters in both kinematic models exists. This is reasonable to assume, since, as it was explained in Section 2.2.3.B, the inverse kinematics model is obtained through the direct kinematics model. So, for a chosen number of clusters, if an obtained direct kinematics model presented lower validation error, consequently, when inverted, it would also result in a lower error for the

inverse model.

In Table 4.1 it was also observed that the permutation data gives slightly lower errors compared with the random data for the direct kinematics model in all the showed combinations. The same did not happen for the inverse kinematics as it is possible to see in the model composed by two clusters of 3rd order, although the algorithm performed better for the direct kinematics using the permutation input, the inverse kinematics model for this input presented higher standard deviation and mean error compared to the random data.



(a) One polynomial of 3rd order with error of 0.3264 ± 0.1459mm



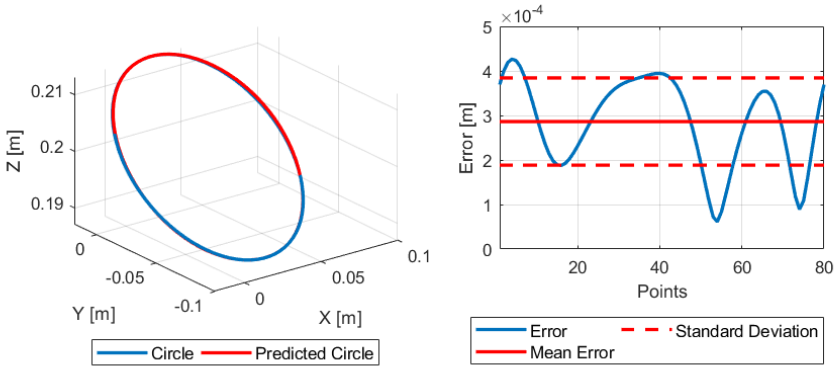
(b) Two polynomials of 3rd order with error of 1.1987 ± 1.4657mm

Figure 4.4: Point-to-point circle trajectory with random data

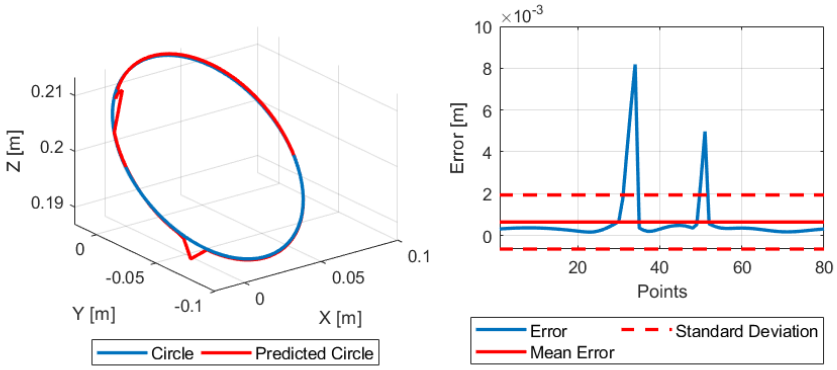
Finally, to test the models obtained it was used an open-loop control. It was created a point-to-point trajectory that was given as input to the inverse kinematics model obtained, which gives as output the lengths of each one of the potentiometers. These outputs were used as inputs to the Bionic Handling Assistant simulation, expecting that the tip of the robot would describe the chosen trajectory.

It was created a circle trajectory, with its limits inside the workspace of the robot. Analysing the

possible combinations of order and cluster that gives the overall lower error for the direct kinematics model, and for the inverse kinematics model, it was verified that for both types of data the overall lower error model is given by one polynomial. Since it was desired to see also the behaviour of the method using clusters, it was chosen to test the trajectory also for the clustered model that gives the lower error for the direct kinematics.



(a) One polynomial of 4th order with error of 0.2863 ± 0.0980mm



(b) Two polynomials of 3nd order with error of 0.6454 ± 1.2924mm

Figure 4.5: Point-to-point circle trajectory with permutation data

Following the criteria described, for the random data it was picked a polynomial of 3rd order, for the permutation data, a polynomial of 4th order and a clustered-model of two polynomials of 3rd order for both of them. The inverse kinematics models were trained and then their outputs (actuators parameters) were given as inputs to the BHA model resulting in Figures 4.4 and 4.5.

In Table 4.1 it is possible to see that the errors for the permutation input using one polynomial of 4th order is slightly smaller than the errors obtained using a 3rd order polynomial for the random data. So, as was expected, the circle trajectory for the permutation also performed better. For the clustered model,

one may see in Table 4.1 that for the permutation data, the direct kinematics error is lower than the one obtained with the random data, and the opposite occurs with the inverse kinematics error. Performing the circle trajectory, it was obtained not only lower mean errors but also lower standard deviation for the permutation data.

It is worth to remember here, as mentioned before in this Chapter, that the model obtained depends on the data selected for training. The results obtained in Table 4.1 has come out from ten different sets for training, the results obtained in the trajectories of Figures 4.4 and 4.5 were obtained using just one set of data. During the tests, it was noticed that, specially for the clustered results, some sets of data performed better with the random and some with permutation input, which indicates that for the real robot experiments, it should be selected to test more than one set of data.

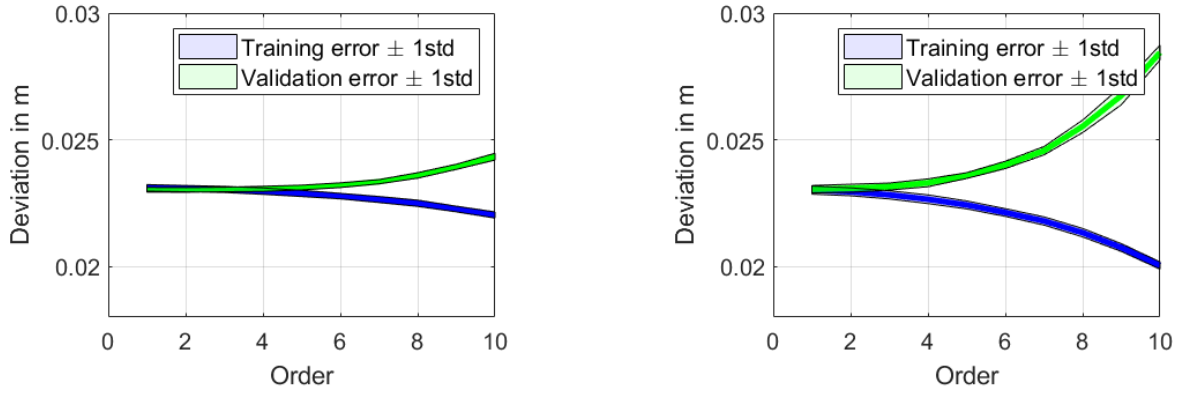
In both clustered trajectories (Figures 4.4(b) and 4.5(b)) it is observed some peaks giving by one or two points that give higher errors and consequently make the trajectory not smooth. One possible reason for this to happen is the misclassification of this points inside the clusters' models, since the classification is not done evaluating the error of each model but based on a pre-trained classification tree explained in Section 2.2.4 some misclassified points were already expected.

4.1.2 McKibben-based I-SUPPORT

Using the set of data provided from the single I-SUPPORT module, it was obtained the error of the direct kinematics model to variable orders of the polynomials. It was performed the same test with one (Figure 4.6(a)) and three clusters (Figure 4.6(b)), in which it was possible to see an increase of the overfitting of data with the increase of order and number of clusters, indicating that the best model to describe the robot's workspace would be the one using one polynomial of first order. Considering that one of the characteristic of soft robots are their infinite DoF motion, to conclude that the workspace of the I-SUPPORT could be described by a straight line seemed very counter-intuitive, so the reasons behind this result needed some further investigation.

In Figure 4.7 one may see the results of the clustering part from the algorithm described in Section 2.2.2. The partition of the actuators' input data made in three clusters and the correspondent mapping of the configuration space do not show a clear separation between the clusters in the configuration space, instead the points appear all over the workspace, which indicates that giving the same input twice could make the robot go to totally different positions in the space. In fact the data acquired was a motor babbling data sampled with 40Hz, this means that the samples are generated randomly with almost no time between the inputs. Since the robot is still moving in the moment that the next input is given, the previous input influences the next one, which makes the data unsuitable for obtaining kinematic models with the algorithm, as it describes not only the static but also the dynamics of the robot.

Using the set of data from two modules of the I-SUPPORT, it was again obtained the direct kinematics



(a) One cluster and variable order of the polynomial

(b) Three clusters and variable order of the polynomials

Figure 4.6: Direct kinematics error for validation and training data of one module of the I-SUPPORT. Data provided by The Biorobotics Institute from Sant'Anna School of Advances Studies.

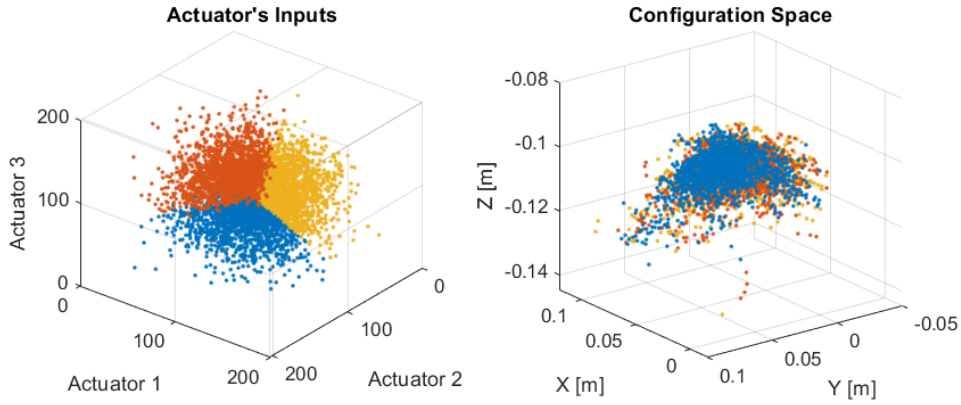


Figure 4.7: Results of the clustering step of the control algorithm performed

model error to variable orders of the polynomial which can be seen in Figure 4.8. The minimum mean error and standard deviation of 85.3470 ± 1.1928mm is obtained with a polynomial of 5th order, with magnitude ten times bigger than the other tests performed in this Chapter, including the one with I-SUPPORT non-static real data.

In order to try to reduce the error, it was performed a test increasing the number of clusters and varying the order of the polynomials. Due to the limitation of the method regarding the minimum number of samples for cluster explained in Section 2.2.6 and considering the number of samples available, the change in number of clusters was very limited and not sufficient to have a significant decrease in the error obtained in Figure 4.8. For two modules of the I-SUPPORT it was expected a higher error in the model compared to one module of the I-SUPPORT, due to increase of the gravitational effects in the robot's movements caused by the mass added by the extra module. However, to verify if the method

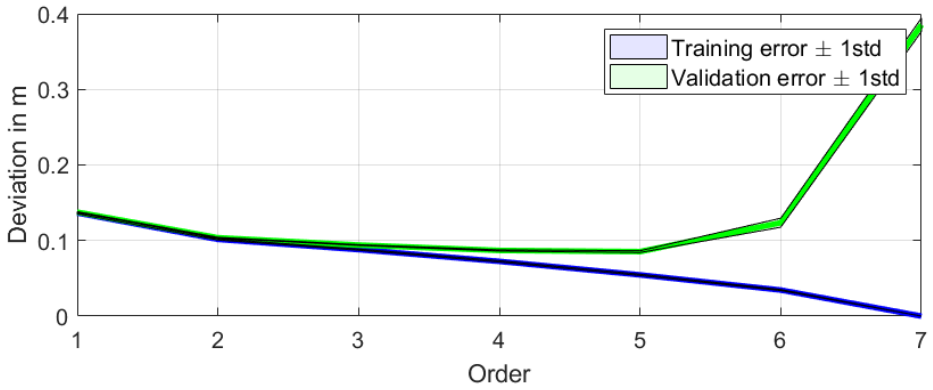


Figure 4.8: Direct kinematics model error of the data provided from two modules of the I-SUPPORT

could provide a good kinematic model for the robot, it would be necessary more data samples allowing to increase the number of clusters and hopefully reduce the error of the model obtained.

4.1.3 Conclusion

The pre-experimental tests revealed a direct relationship between the inverse kinematics and the direct kinematics model, in which a specific number of clusters gives the lower error for both models at the same polynomial order. Since the errors of the model are very dependent on the set of data selected for training and validation it should be selected more than one set of data for training the models of the real robot to perform the trajectories.

The variation of the percentage of data did not show significant changes in the error of the model, but increased the computational time for training it, therefore it was chosen to perform all the tests in the real robot experiments using 70% of training data.

In these pre-experimental tests, the inputs generated by permutation of points presented lower errors for the direct kinematics model and for this reason it was chosen to generate the I-SUPPORT inputs in the experiments described in Section 4.2. Considering the fact that the random generation input data can provide inputs too close to each other, combined with the robot natural oscillation it could result in the algorithm struggling to devise meaningful clusters of data and this also contributed to select the permutation generated input to the experiments.

In the experiments that will be performed the data acquired should be static, this means that one should be wait until the robot reaches the steady state to give the next input, otherwise the algorithm will not work properly. For future works, with a second module coupled in the I-SUPPORT it should be assured that the number of samples acquired allows a sufficient variation of polynomial order and clusters to find the model that best describes the kinematics of the robot.

4.2 Experimental Results

In this Section it will be described the experiments performed using the real world robots. First it will be made a mechanical characterization using the McKibben-based and the silicone actuators. Followed by a kinematic characterization comparing the I-SUPPORT robots assembled with both of the actuators. And finally, the control method explained in Chapter 2 will be applied in the I-SUPPORT with silicone actuators and tested through three different point-to-point trajectories.

4.2.1 Single Actuators Pressure vs Elongation

The first experiment was made using a single actuator without being assembled in the manipulator, the goal is to compare the silicone actuator and the McKibben-based one characterizing the behaviour of them without load. To simulate the constrains imposed by the discs that work as reinforcement structure, it was placed mechanical washers equally spaced along the actuators. The elongation was measured with a measuring tape that was fixed on the bench. The workbench used in the experiment is showed in Figure 4.9.

Initially, both of the actuators were measured without any pneumatic input, the second input was only registered when the actuators presented some visual elongation movement and from there on the pressure was gradually increased. Ideally in this experiment the pressure supplied to the actuators should be increased until it causes failure. But since it was not desirable to make the future use of this actuators impossible, the ultimate input was registered when no further elongation was observed in the actuators and the surface showed some evidences that the volume capacity was almost at its limits.

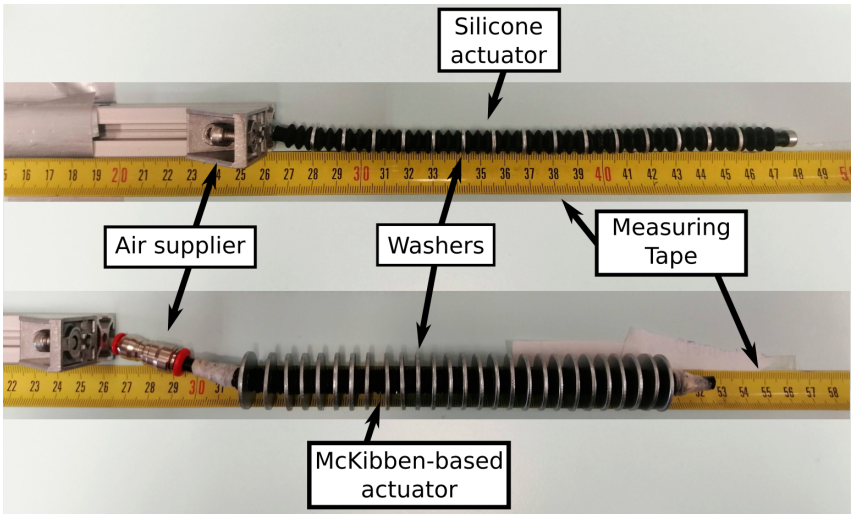


Figure 4.9: Workbench during the characterization without load for the McKibben-based and the silicone actuator

According to the User Manual of the pneumatic set used in this project [54], setting the pressure at

4bar for the air supplier the relationship between the digit and the pressure applied to the pneumatic actuators can be described as linear, as it is possible to see in Figure 4.10. Although in this project was used a pressure of 3bar for air supplier, assuming the valves with absolute pressure sensor and considering that it was worked with pressures under 2bar inside the actuators, one may assume that the linear relationship remained the same and given by Equation (4.1), where d is the digit number.

$$FittedCurve(d) = 0.00586d - 8.224 \times 10^{-5} \tag{4.1}$$

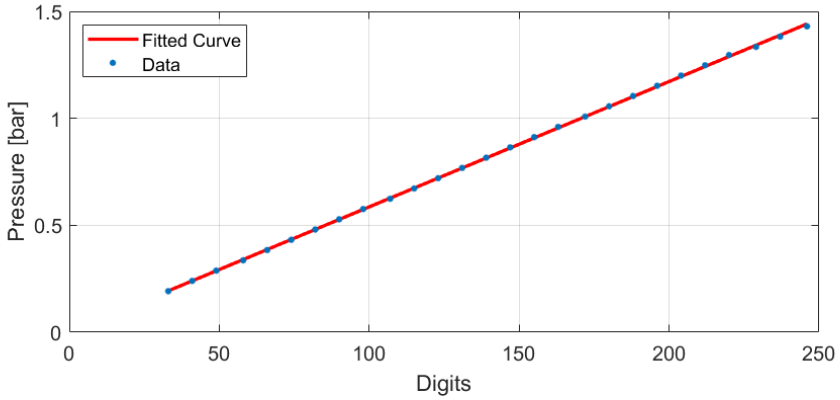


Figure 4.10: Relationship between digit and pressure applied to the pneumatic actuators

In Figure 4.11 it is possible to see the results of the experiment, since the initial length of the actuators differ in 1.6cm, the plot was made with respect to the initial length. The McKibben-based actuator showed at the limit a slightly higher elongation, 0.5cm bigger than the Silicone one. The Silicone actuator presented a linear behaviour from the beginning to the end of the test. On the other hand, the McKibben-based actuator presented a parabolic behaviour, probably due to the mechanical resistance imposed by the braid of the McKibben-based actuator which allows a higher stiffness compared to the silicone actuator.

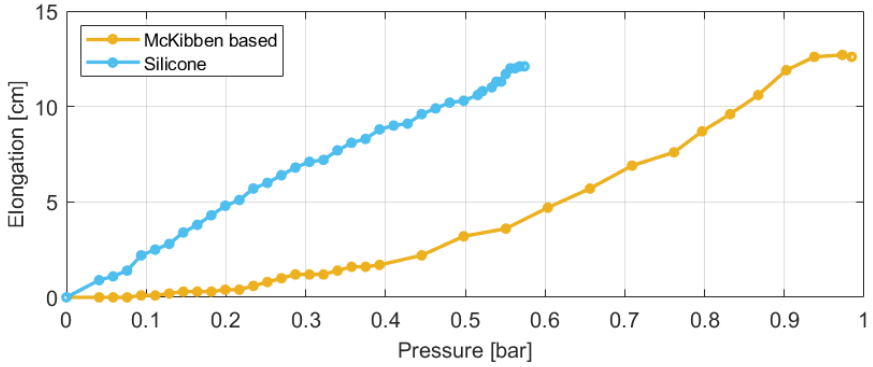


Figure 4.11: Elongation per pressure of the McKibben-based and the silicone actuators

4.2.2 Kinematic Characterization

In this section it will be provided an experimental characterization of the functionalities of the manipulator with the silicone actuator, which includes elongation, omnidirectional bending, the complete reachable workspace and the repeatability. In all the mentioned experiments it will be provided a comparison between the proposed robot and the one using the McKibben-based actuator.

Based on the maximum elongation of the single actuators obtained in Section 4.2.1, it was selected the maximum pressures or, essentially the valve set-point reference value coded as an 8 bit integer, that would be work with during the following experiments. To guarantee security and prevent the damage in the actuators it was chosen initially to work 15% bellow the limit pressure as upper bound for both actuators, resulting in maximum 0.48bar (Digit $= 82$) for the silicone actuator and 0.83bar (Digit $= 142$) for the McKibben-based one.

During the first experiment it was notice that for the McKibben-based actuator robot the defined bounds did not produce the expected results: the upper bound was too high resulting in some ruptures in the balloon chambers inside the actuators. This probably happened due to natural asymmetry between the actuators, since they are deformed manually in their manufacture, which increases the mechanical resistance in the robot's movement. In Figure 4.12 it can be noticed a slightly inclination in the robot assembled with the McKibben-based actuators, not present in the silicone one, which although has also a manual step, this step is easily reproduced many times without major differences between them. So it was defined a new maximum for the McKibben based robot of 0.69bar (Digit $= 117$).

As stated before, the single actuators do not have the same length and consequently the robot assembled with them also do not have the same length, in order to make the comparison possible, the results will be shown with respect to the starting configuration.

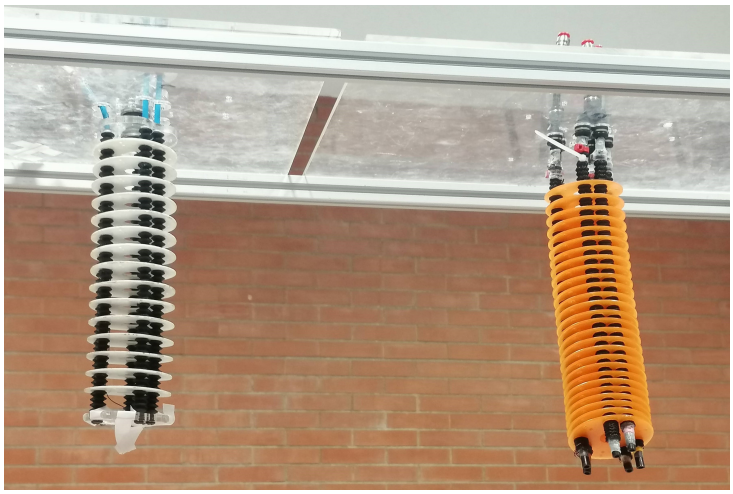
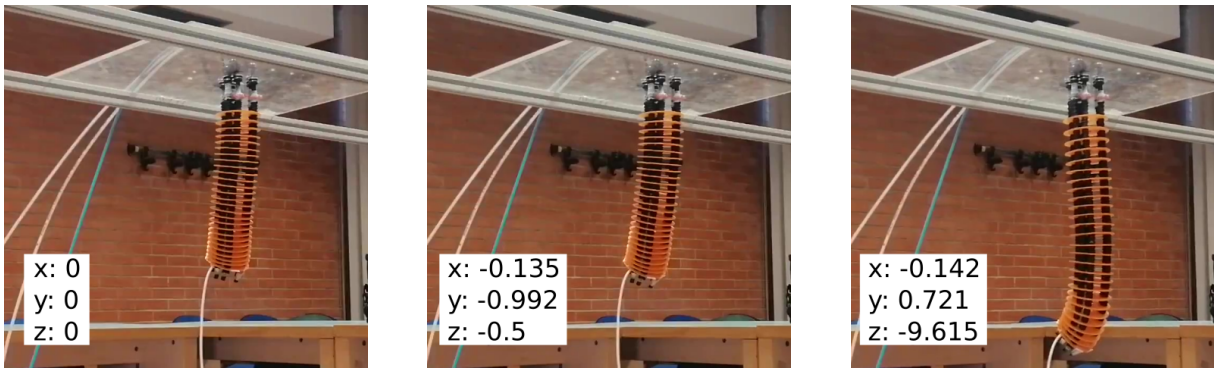


Figure 4.12: I-SUPPORT robots assembled with the two types of actuators. On the left, in white, with the silicone actuators and on the right, in orange, with the McKibben-base actuators.

The data acquired during the experiments were all static, but the time it takes for the robots to reach the steady state position is different: the McKibben-based actuators robot takes 2s against 5s of the silicone actuators robot. This may be due the fact that the braid of the McKibben-based introduces a higher level of friction, consequently the same level of disturbance in both actuators takes more time to be stabilized in the silicone manipulator. Due to this difference, the experiments performed with the silicone I-SUPPORT are more time consuming.

4.2.2.A Elongation

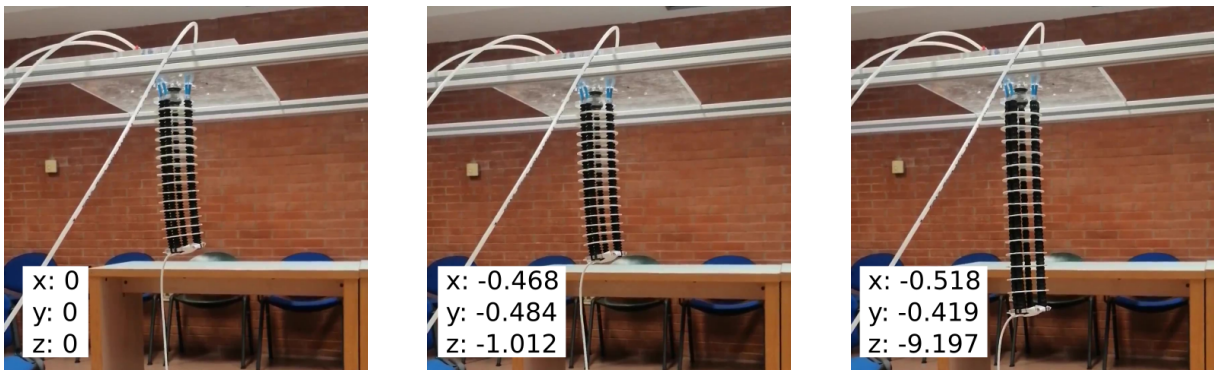
The elongation corresponds to the activation of all the three chambers at the same time with the same pressure. It was used three values of pressure: 0, 0.06bar and 0.48bar (0, 10 and 82 in digits) for the robot assembled with silicone actuators and 0, 0.16bar and 0.69bar (0, 27 and 117 in digits) for the robot assembled the McKibben-based one. In Figures 4.13 and 4.14 is possible to see the results of the experiment with the position of tip of the robots indicated in centimeters on the left corner. Figures 4.13(b) and 4.14(b) represent the lowest input that provided a significant change on the length of the manipulators. Although the silicone robotic arm presented a higher initial elongation of 1.012cm with respect to the initial position against just 0.5cm for the McKibben-based, the maximum elongation was higher in the latter. Considering the single actuators results (Section 4.2.1), the elongation of 9.615cm presented by the robot assembled with the McKibben-based actuators be slightly higher than the elongation of 9.197cm from the silicone robot was already expected.



(a) Rest Position **(b)** Minimum Elongation **(c)** Maximum Elongation

Figure 4.13: Elongation of the robot assembled with the McKibben-based actuators with the indication of the position (x, y, z) of the tip of the manipulator in centimeters.

It is also possible to observe in Figures 4.13 and 4.14 that although all the actuators were activated at the same time with the same input pressure, the robots did not follow a strictly straight line towards the ground. In fact, both robots presented a deviation from the origin, more pronounced in the silicone manipulator in XZ plane and in the YZ plane for the McKibben-based one.



(a) Rest Position **(b)** Minimum Elongation **(c)** Maximum Elongation

Figure 4.14: Elongation of the robot assembled with the silicone actuators with the indication of the position (x, y, z) of the tip of the manipulator in centimeters.

4.2.2.B Bending due to a single chamber

The bending will be evaluated in two different activation patterns, in which the robot will be subject to two conditions: with a tendon constrain and without it. First it was performed a test activating only one pneumatic chamber, supplied with pressures varying from 0bar to the maximum pressure previously defined for each robot. It was given 11 inputs, that were equally spaced from 0 to 0.48bar (0 to 82 digits) for the silicone manipulator and from 0 to 0.69bar (0 to 117 digits) for the McKibben-based one.

Figures 4.15 and 4.16 illustrate the resulting behaviour of this activation pattern for both robots without the tendon constraint. The z-axis displacement of the McKibben-based manipulator is almost twice of the one showed by the robot with silicone actuators, although the x-axis displacement was the same as one may see in the xz-view of the Figures 4.15 and 4.16. In y-axis, the maximum displacement for the silicone actuator do not occur at the final position, which means that the robot's tip describes parabolic-like trajectory in the xy-plane while the McKibben-based manipulator describes a more linear trajectory.

It is also possible to notice that the McKibben-based manipulator showed a pure bending movement, from the initial position it can be seen that increasing the chamber pressure results in the robot always going up and right. The same is not verified in the silicone manipulator, before observing the bending, with the first three pressure inputs the robot goes down and right, after that, the following inputs result in the bending movement as the one observed in the manipulator with the other actuator model. Which means that the behaviour of the silicone robot when activating one chamber is characterized by a diagonal elongation movement followed by a smooth bending.

Although the robots present different behaviours with the activation of one chamber, the final orientation of tip during the bending movement is similar in both of them: the z-direction is pointed to the

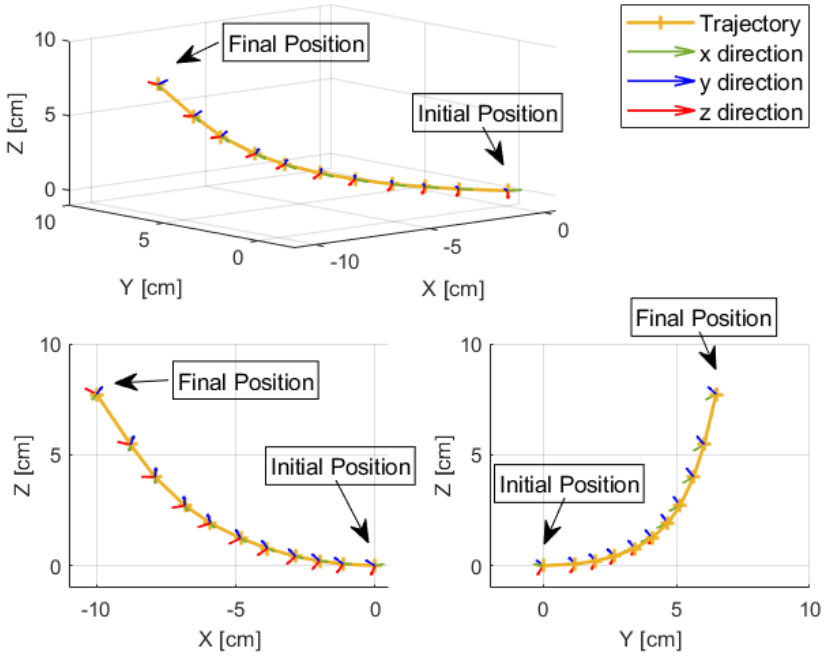


Figure 4.15: Global bending movement with the activation of one pneumatic chamber of the robot assembled with the McKibben-based actuators.

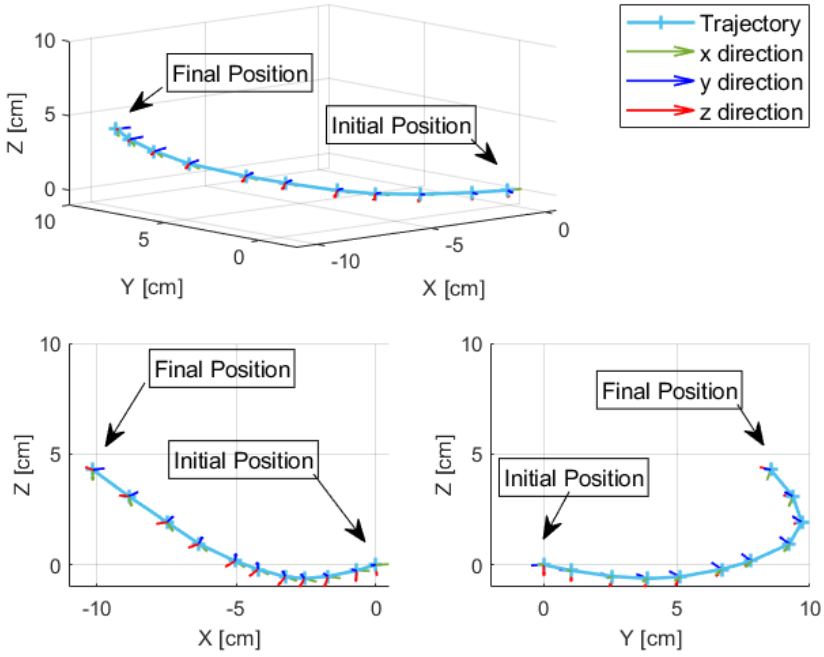
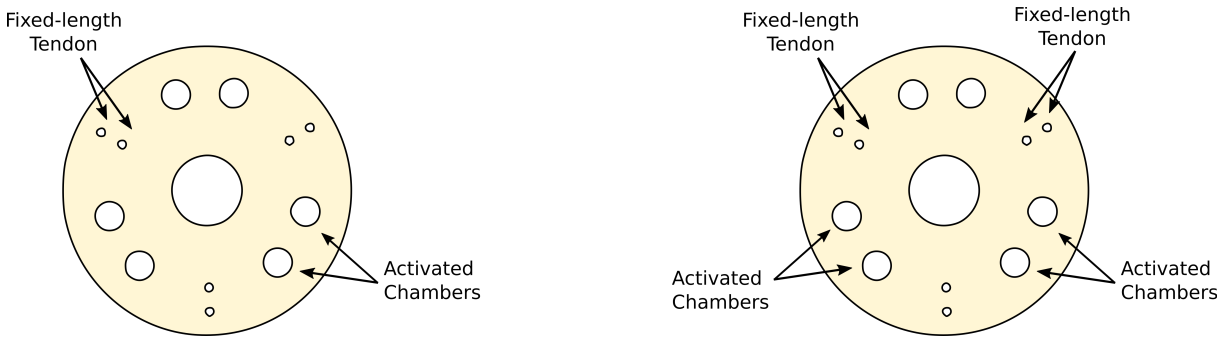


Figure 4.16: Global bending movement with the activation of one pneumatic chamber of the robot assembled with the silicone actuators.

right and slightly tilted upwards. The natural inclination of the McKibben-based manipulator mentioned before can be seen when comparing the z-direction of both robots at the initial position, in Figure 4.16 in xz-view and in yz-view the red arrow is pointed down almost parallel to the z-axis, observing the same views in Figure 4.15, it is possible to notice that the red arrow forms an angle with the z-axis.

To add the tendon constrain in the robots, one of the holes made in the distal disc of reinforcement structure was chosen to host the tendon. The tendon was passed through all the consecutive and parallel discs of the structure until it reaches the base, where it was fixed ensuring that the robot's natural length was maintained. To carry out the experiments, the activated pneumatic chamber was the one that was diametrically opposite as illustrated in Figure 4.17(a).



(a) Activation of one chamber

(b) Activation of two chambers

Figure 4.17: Arrangement of the tendon and the activated pneumatic chambers.

Figures 4.18 and 4.19 illustrate the resulting behaviour of this activation pattern for both robots with the tendon constraint. Comparing them to Figures 4.15 and 4.16 one may affirm that the tendon did not have influence in the global bending movement for both robots, since the overall final position and the shape of the trajectory are almost equal in both conditions. The main difference can be seen in the silicone manipulator which had its bending movement delayed, compared to the non-tendon behaviour it requires a higher input pressure to stop the diagonal elongation and starts the bending.

4.2.2.C Bending due to two chambers

The second activation pattern for evaluating the bending was performed activating two adjacent pneumatic chambers, both supplied with the same pressures, varying from 0bar to the maximum pressure previously defined for each robot. It was given 11 inputs, that were equally spaced from 0 to 0.48bar (0 to 82 digits) for the silicone manipulator and from 0 to 0.69bar (0 to 117 digits) for the McKibben-based one. Again, it was performed the test with and without the tendon constraint.

Figures 4.20 and 4.21 illustrate the resulting behaviour of this activation pattern for both robots without the tendon constraint. The resultant movement of the activation of two pneumatic chambers for both

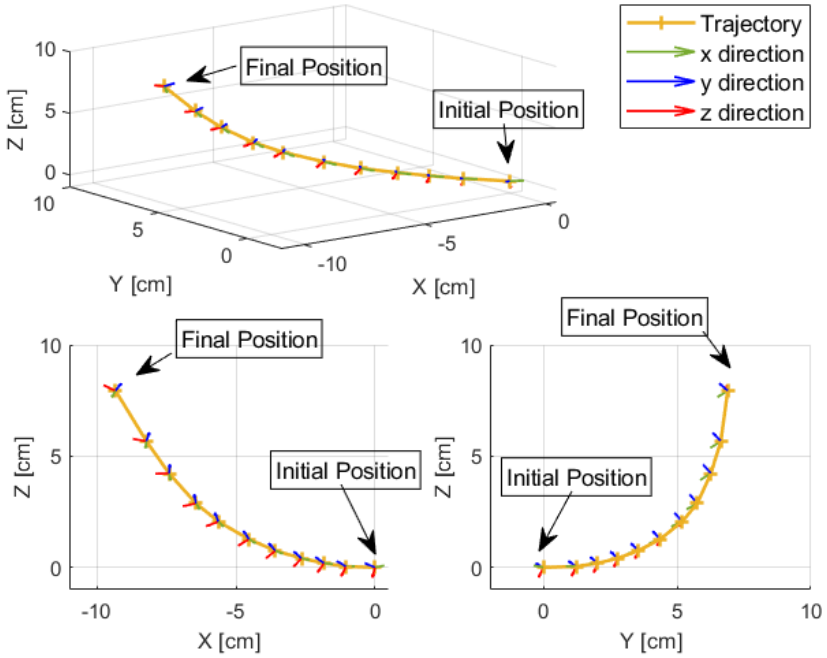


Figure 4.18: Global bending movement with the activation of one pneumatic chamber restrict by an opposite tendon of the robot assembled with the McKibben-based actuators.

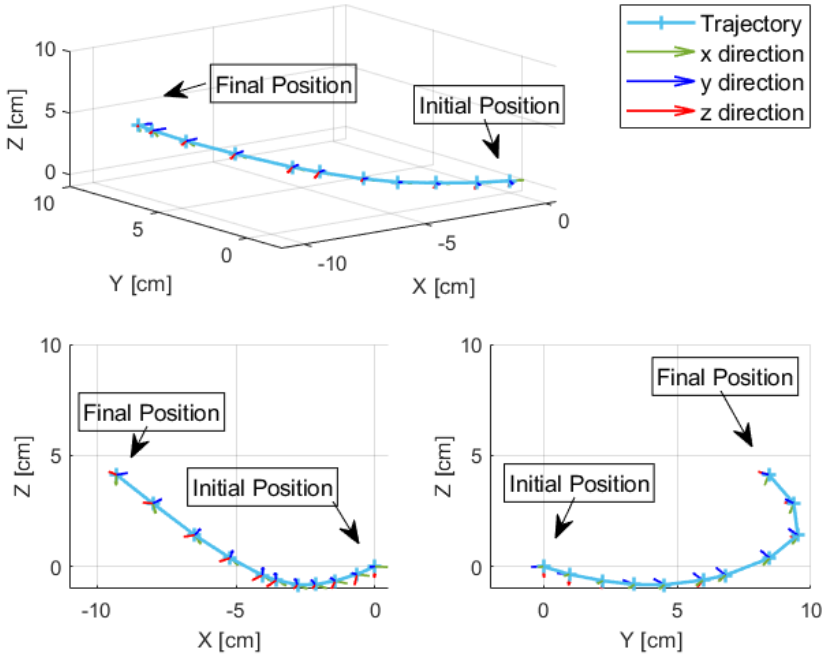


Figure 4.19: Global bending movement with the activation of one pneumatic chamber restrict by an opposite tendon of the robot assembled with the silicone actuators.

robot presented an accentuated diagonal elongation first, followed by a bending started just in the last three input pressures for both robots, which is different from the global movement obtained when activating one chamber in which bending movement predominated. Due to this behaviour the final position of the end-effector for the silicone and the McKibben-based manipulators is lower than the initial position, but with changed orientation.

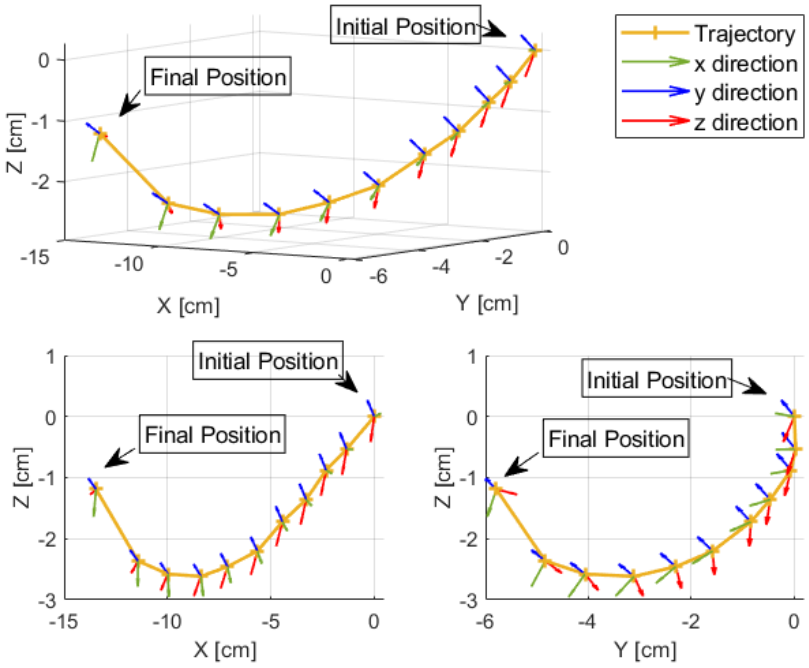


Figure 4.20: Global bending movement with the activation of two adjacent pneumatic chambers of the robot assembled with the McKibben-based actuators.

In the xz-view of Figures 4.20 and 4.21 one may see that both robots presented an diagonal elongation almost equal, until approximately 10cm in x-direction and 2.5cm in z-direction, after that when the bending movement starts, the McKibben-based robot goes upward 0.5cm more and around 1cm more to the side than the silicone one. On the other side, in yz-view the behaviour of the robots is totally different, the McKibben-based robot presented a displacement of 6cm in y-direction while the silicone manipulator presented less than 1cm of displacement in the same direction, therefore its movement was almost entirely in the xz-plane. For the silicone robot, it is also possible to noticed the same behaviour observed in the global bending of Figure 4.16, with the highest pressure inputs the robot starts to approximates to the initial position in the y-direction.

To add the tendon constrain in the robots, it was followed the same procedure explained in Section 4.2.2.B, but instead of choosing only one pair of holes it was chosen two and the respective chambers, diametrically opposite of them as illustrated in Figure 4.17(b). Figures 4.22 and 4.23 illustrate the

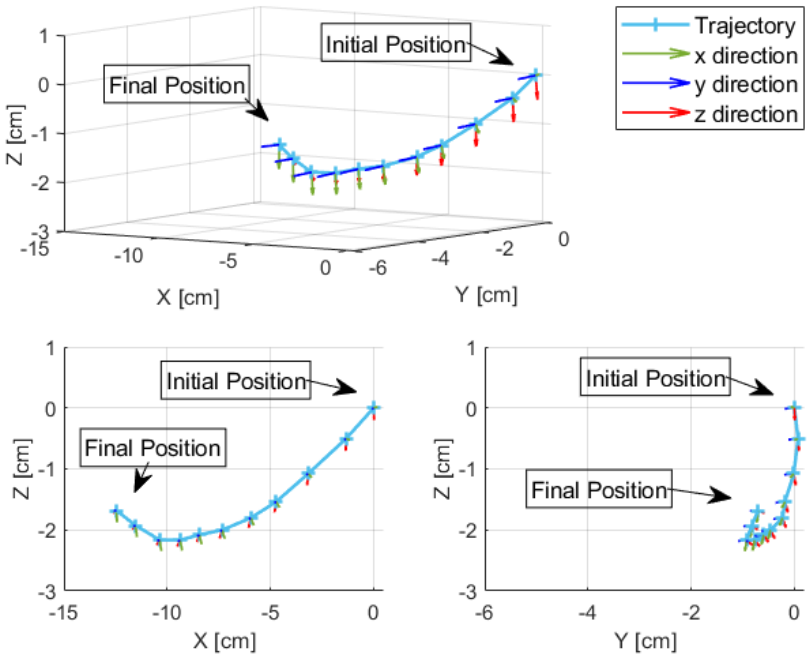


Figure 4.21: Global bending movement with the activation of two adjacent pneumatic chambers of the robot assembled with the silicone actuators.

resulting behaviour of this activation pattern for both robots with the tendon constraint. Unlike the results obtained using one chamber (Figures 4.18 and 4.19), the added constrains and the activation of the two chambers come out in a completely different behaviour of the manipulators when compared to the unrestricted movement.

The McKibben-based manipulator (Figure 4.22) presented a slightly diagonal elongation with the three lower pressures, but right after that it starts to bending reaching approximately 10cm in the z-direction, the highest position recorded in the tests carried out in this project. Like the previous tests, it showed a displacement in x and y directions following the same pattern: the increase of the pressure makes the robot's tip to move away from the initial position in x and y directions.

The silicone manipulator (Figure 4.23) presented a pure bending, in which the increase of pressure makes the robot's tip goes always upper in z-direction, reaching an even higher position than the McKibben-based constrained manipulator (Figure 4.22), 15cm from the initial position. It is possible to see that the silicone robot starts to bending over itself with the highest input pressures, when it stops to increase the displacement in x-direction and begins to return to the initial position in x, continuously increasing the z displacement. This behaviour is also reflected in the yz-view, where one may see the S-shape curve described, which means that the robot's tip started to go in one direction and at some point it begins to go in the opposite direction exceeding the initial position when it bends over itself.

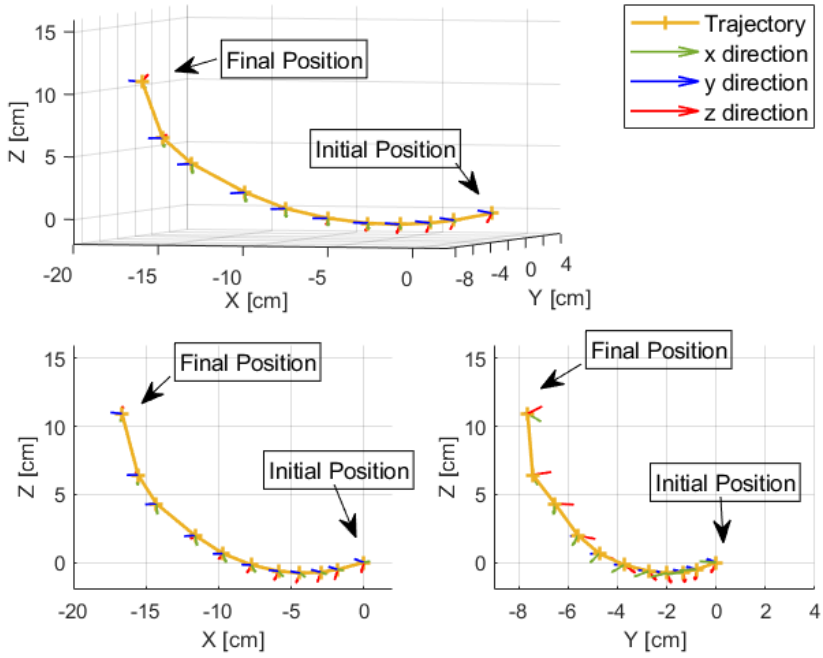


Figure 4.22: Global bending movement with the activation of two adjacent pneumatic chambers restrict by tendons of the robot assembled with the McKibben-based actuators.

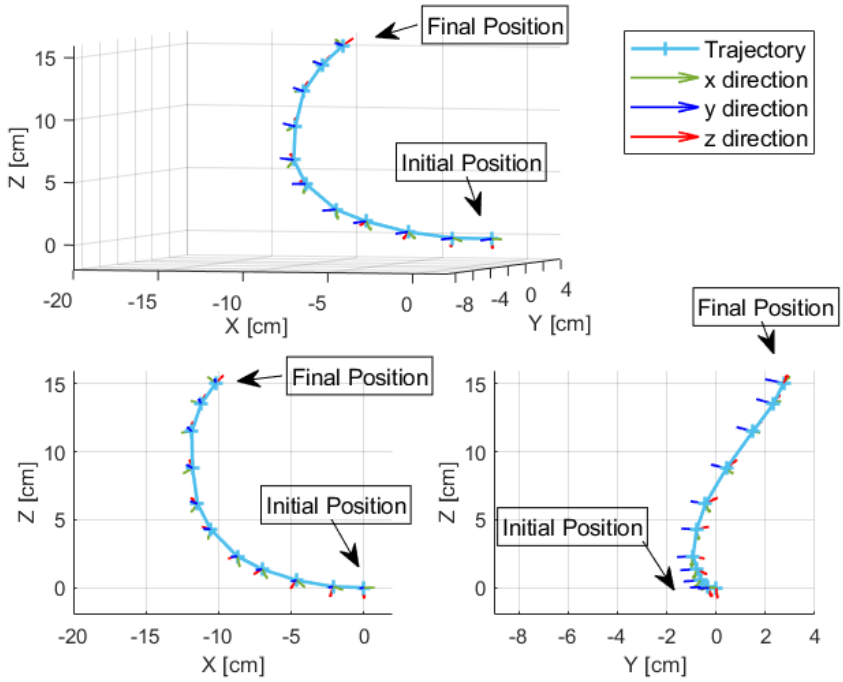


Figure 4.23: Global bending movement with the activation of two adjacent pneumatic chambers restrict by tendons of the robot assembled with the silicone actuators.

4.2.2.D Reachable workspace

The workspace measurement aims to measure all the reachable positions of the end-effector of the robot. In order to do that it was acquired a total of 8000 points, resulted from a permutation of the three pneumatic inputs using twenty values equally spaced from 0 to 0.48bar (0 to 82 digits) for the silicone manipulator and from 0 to 0.69bar (0 to 117 digits) for the McKibben-based one. The results of the experiments regarding the initial position can be seen in Figures 4.24 and 4.25.

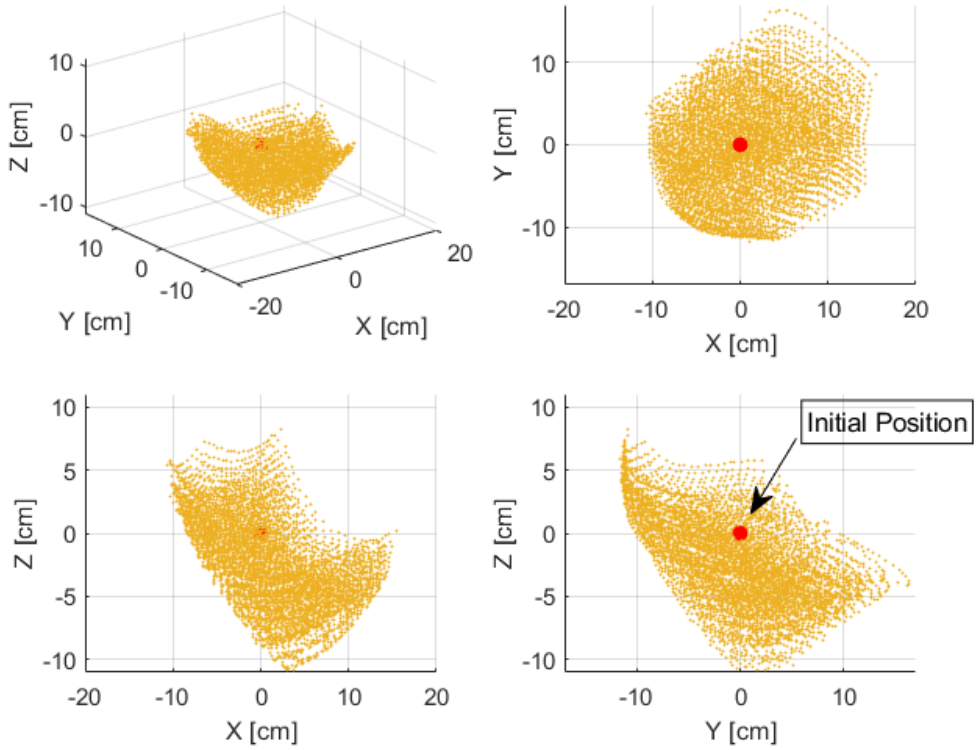


Figure 4.24: Workspace evaluation for the single module assembled with the McKibben-based actuators. The red point represents the initial position of the module in an unactivated state.

The overall shape of the workspace for both robots can be categorized as a volumetric convex with the limits defined by the maximum bending and elongation capabilities, and the xy-view showed a circular-like shape. The McKibben-based module presented a range of 26.0cm, 28.1cm and 19.1cm for the x, y and z directions respectively. A range slightly higher than the results presented by the silicone manipulator, 25.1cm, 25.5cm and 13.2cm for the x, y and z directions respectively.

Although the McKibben-based manipulator showed a better reachability it is possible to see in Figures 4.24 and 4.25, specially in the 2D views, that the silicone module offer a symmetrical motion in every direction. In the xz-view, for example, one may see that the lower point of the McKibben assembled robot has a positive offset from zero in x-axis and even assuming a plane parallel to z-axis passing

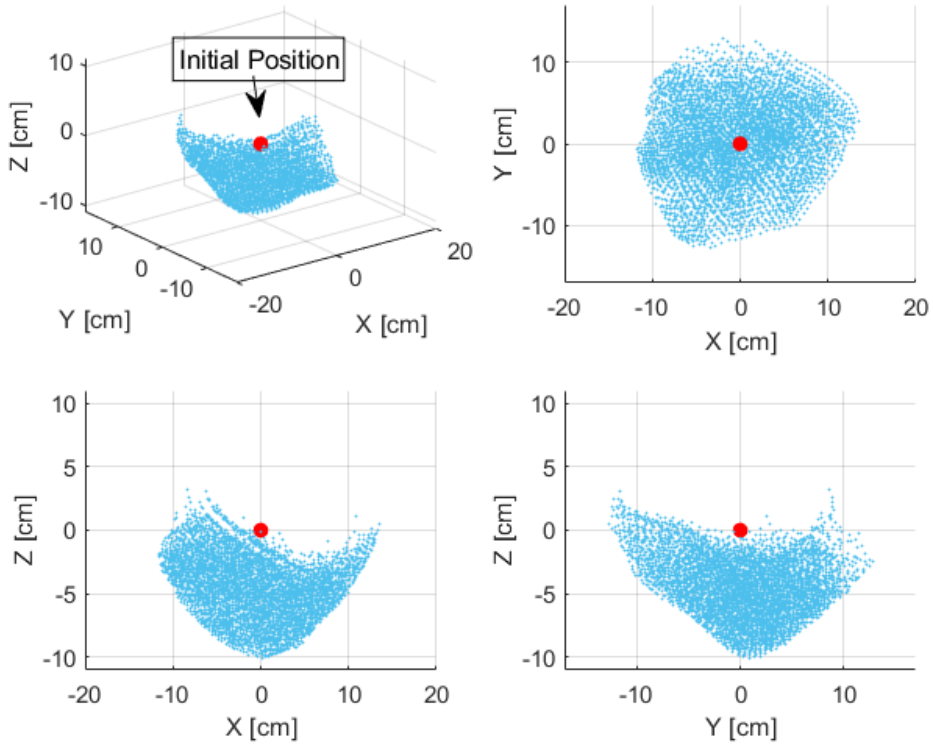


Figure 4.25: Workspace evaluation for the single module assembled with the silicone actuators. The red point represents the initial position of the module in an unactivated state.

through this offset, it presents a pronounced displacement in z-axis direction on the left side of the volume. In contrast to the behaviour of silicone manipulator, in which, observing the same view, the lower point occurs at zero in x-axis and a plane parallel to z-axis at $x = 0$ corresponds to the symmetrical plane, dividing the volume into two almost identical pieces.

The asymmetry of the shape in the McKibben-based module, as already identified by Ansari et al. in [32], can be attributed to the manufacturing procedure, in which the manual deformation of the braid could result in irregularities that end up being increased when the robot is assembled with the six actuators as one may see in Figure 4.12.

4.2.2.E Repeatability

The evaluation of the repeatability corresponds to a statistical characterization of the modules verifying how repeatable the positions of the workspace are. To perform this test it was chosen three random pneumatic inputs of each condition studied: bending due to one chamber, bending due to two chambers and elongation, that were reached from twenty different random input values from the set of data that resulted on the reachable workspace of Figures 4.24 and 4.25, starting at the rest position as illustrated

in Equation (4.2), where Point A represents the chosen point.

$$\text{Rest} \rightarrow \text{Point A} \rightarrow \text{Random Point} \rightarrow \text{Point A} \rightarrow \text{Random Point} \rightarrow \cdots \tag{4.2}$$

It was performed 9 tests with different goal points, each one with 20 samples, which sum 180 points. For each test, the 20 samples of the chosen point (Point A) acquired were normalized centering the data to have mean 0, allowing the analysis regarding the overall probability of a given input brings the robots to the same position in the space. The normalized data acquired is showed in the histogram of Figure 4.26, in which each bar width is 0.1cm.

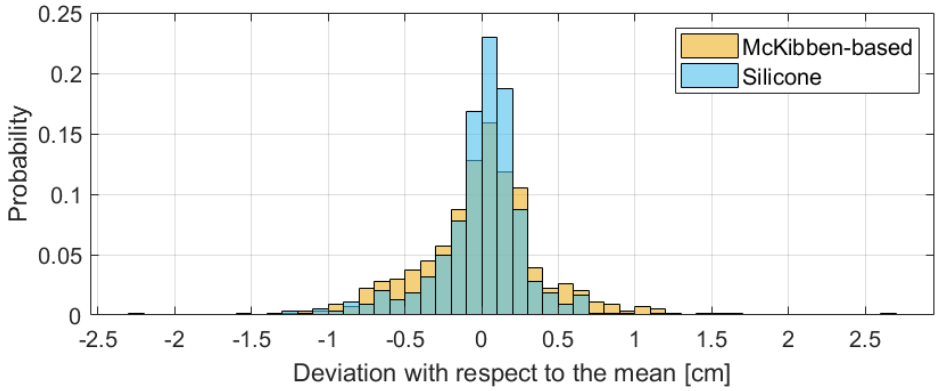


Figure 4.26: Histogram of the data acquired with nine different goal points that were normalized to zero mean for each manipulator assembled with the actuators.

Although both robots showed most samples around 0.5cm of deviation, the McKibben-based module presented as maximum deviation during the tests 2.7cm, while the silicone module presented 1.5cm, almost half of the value presented by the other manipulator. The silicone robot also exhibits 94% of the data within 0.61cm of deviation from the mean values, against 0.86cm showed by the McKibben-based robot. Regarding the repeatability the manipulator assembled with the silicone actuators demonstrates a better performance not only with respect to the maximum values but also to the value inside of the confidence interval that were both smaller.

4.2.3 Control

In this section the polynomial method to obtain the inverse and the direct kinematic models of the soft robot explained in Chapter 2 will be applied in the I-SUPPORT assembled with the silicone actuators. First it was performed a sensibility test to choose the models that will be used to a open-loop controller tested with three distinct point-to-point trajectories.

4.2.3.A Pre-process Data

As mentioned in Section 2.5.2.A the data collected with the SMART-DX system was used only to perform the sensibility tests of Section 4.2.3.B. In order to apply the polynomial method described in Chapter 2, it should be acquired samples to cover the whole space the robot could reach, which means the same 8000 points acquired in Section 4.2.2.D. Considering that the resultant data with the SMART-DX System were a series of step responses, to extract the steady state position to obtain the models the data should be pre-process.

It was chosen to apply a Gaussian-weighted moving average filter to smooth the oscillations of the step response aiming to obtain steady state. A moving average takes several sequential values of a time series, called windows, and computes the average of those points that will be the output value [58]. A simple average of the points for the step response would result in undesirable outputs, since the beginning and the end of each step would be influenced by the rising to the next step, so it was added a Gaussian-weight to the points inside the window. Based on trials, the window of the applied filter was composed of 300 points.

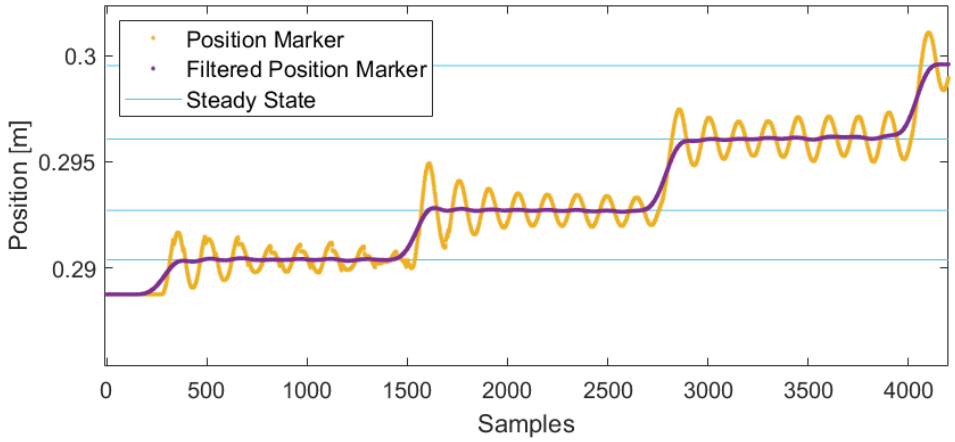


Figure 4.27: Example of the moving average filter applied in the data acquired by the SMART-DX System.

In Figure 4.27 it can be seen the data acquired in yellow and the filtered data in purple, to extract the steady state of each input it was taken the average of 1250 points, which corresponds to the 5s acquired at 250Hz after the input that was waited until the robot reduce its oscillations. Since it was attached to the tip of the robot three markers, this procedure had to be repeated for the x, y and z coordinates of the three markers.

To obtain the final position of the center of the tip of the robot it was taken the average of the x, y and z positions of the markers, obtaining the centroid of the triangle formed by the markers displaced around the tip of the robot. This operation was only needed because it was used two types of sensors, using the Aurora system the sensor was attached in the center of the tip, and since the sensibility test

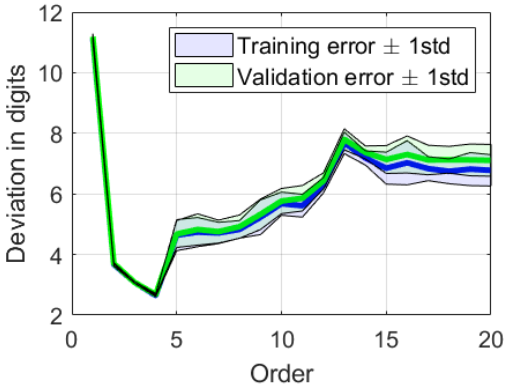
is a previous step to apply the control, it is mandatory to have the same input and output relationship: three inputs, one for each pneumatic actuator, to three outputs, the x, y and z coordinates of the center of the tip.

4.2.3.B Combination of order and cluster of the models

Using the data pre-processed with 8000 samples it was performed an analysis under different conditions to define which models describe the kinematics of the robot better. According to the results obtained during the studies in Section 4.1, 70% of the data was used for training the models, since increasing the data did not show considerable differences in the model errors, but it increased the computational time.

The analysis begins by observing Figure 4.28(a) in which is possible to see the evolution of the error of the direct kinematic model varying the polynomial order with one cluster. After 14th order, the training error stops to reduce and begins to increase for higher orders, which indicates that the order of the model should not be higher than 14th.

(a) Error of the direct kinematic model during the training and validation.



(b) Error of the inverse kinematic model during the training and validation.

Figure 4.28: Error presented by the models using one cluster and varying the order of the polynomial

During the pre-experiments it was stated that using the same number of clusters, the direct and the inverse kinematic models presented the lower error at the same polynomial order. However, using the real data acquired for the silicone manipulator this behaviour does not repeat, in Figure 4.28(b) one may see that the errors of training and validation for the inverse kinematic model decrease until polynomials of 4th orders, not at 14th order as was expected due to the direct kinematic model errors observed in Figure 4.28(b). The absence of the relationship between the models makes it necessary individual evaluation of the model errors varying the order of the polynomial and the number of clusters.

After testing several combinations of polynomial order and the clusters, the lower error amongst the direct kinematic models obtained were given by 8 clusters of 6th order and for the inverse kinematic

model it was reached using 14 clusters of 3rd order. Since for the real robot, the direct relationship between the models errors do not exist, it was also selected to perform the experiments of Section 4.2.3.C a combination of order and clusters that gives the lower mean square error between the direct and inverse kinematic errors. In order to do that, considering that the outputs of the models are positions of the tip and pressure, to make the evaluation possible, the data had to be normalized between 0 and 1, dividing it by the maximum value of each model's output. The values obtained can be seen in Table 4.2.

Table 4.2: Validation error for the direct and inverse kinematic models using selected combinations of order and clusters for the silicone manipulator

		Model Errors	
Cluster	**Order**	**Direct Kinematic**	**Inverse Kinematic**
14	3	2.49 ± 0.61 mm	1444.65 ± 30.51 Pa
8	6	2.05 ± 0.60 mm	1986.52 ± 77.27 Pa
13	4	2.32 ± 0.50 mm	1510.04 ± 39.36 Pa

4.2.3.C Point-to-Point Trajectories

The models selected during the tests described in Section 4.2.3.B were used to perform point-to-point trajectories in the space with the real manipulator. The positions (r) in the space that described a chosen straight line, a circle and a 8-trajectory were given as inputs to the inverse kinematic model, allowing to obtain the pressure (u) that would be supplied to the pneumatic actuators of the silicone manipulator and would result in the trajectories (y) characterizing an open-loop controller (Figure 4.29).

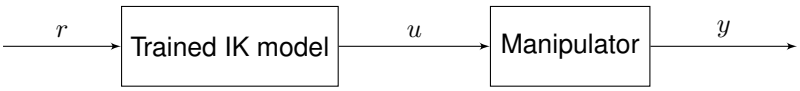


Figure 4.29: Block diagram of the open-loop controller applied

Before starting the tests to evaluate the performance of the proposed controller subjecting the robot to three different trajectories, it was necessary to define the resolution of the points given, since the time interval between them was already fixed at 5s, it means to define the distance between each point in the trajectory. In order to do that it was chosen one random model based on the combination of order and cluster from Table 4.2 and using a straight line trajectory vary the distance between the points and choose the one that gives the lower error to be used in the other trajectories.

Table 4.3: Resolution analysis between two consecutive points in the line trajectory.

Distance between two consecutive points [mm]	4.87	4.33	4.17	4.03	4.77	3.43	3.00
Mean Squared Error [mm]	9.94	10.01	10.51	9.67	10.11	10.07	10.09

According to Table 4.3, the lower error for the straight line trajectory was given using the points spaced with approximately 4mm and therefore, this value will be used to generate the circle and the 8-shaped trajectory. Using an optimization algorithm would probably give more accurate results of what distance should be taken between the points, but it would also be more time consuming, since it would be necessary to repeat the trajectory in several resolutions to perform the algorithm. Considering that the error does not change significantly, a deeper analysis was not carried out.

As mentioned before it was tested three different trajectories: a straight line, a circle and a 8-shaped trajectory. The parametric function of the straight line is given by Equation (4.3), where (x_0, y_0, z_0) represents the position vector of the initial point, (a, b, c) a vector parallel to the line and t a number between 0 and 1. The parametric functions of the circle and the 8-shaped trajectory, also called as Lemniscate of Bernoulli, are given respectively by Equations (4.4) and (4.5), where T is a rotation matrix, θ is an angle between 0 and 2π radians, r is the radius and (x_c, y_c, z_c) is the position vector of the center.

$$\vec{l} = (x_0, y_0, z_0) + t \cdot (a, b, c) \tag{4.3}$$

$$\vec{c} = T \cdot \begin{bmatrix} r\cos\theta \\ r\sin\theta \\ 0 \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{4.4}$$

$$\vec{e} = T \cdot \begin{bmatrix} \frac{r\cos\theta}{1+\sin^2\theta} \\ \frac{r\sin\theta\cos\theta}{1+\sin^2\theta} \\ 0 \end{bmatrix} + \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \tag{4.5}$$

To enable the manipulator to perform the trajectories, they should be within the limits of the workspace of the robot, but an attempt was also made to occupy a considerable area with the curves, varying all their coordinates. Figure 4.30 illustrates the position of the trajectories in the workspace, to better visualize them, it was plotted only 10% of the workspace points acquired. The total length of the straight line was set at 11.68cm, the circle and the 8-trajectory was created with 5cm radius (r) and rotated in 15° around the x-axis (T).

To perform the tests, the models with the combination of order and clusters given in Table 4.2 were trained with five different sets of data randomly selected from the 8000 points that describes the workspace of the manipulator, following the separation of training data previously set at 70%. Figures 4.31 to 4.33 show the performance of the second data set for the trajectories, the black point indicates the initial point of the trajectory and the arrow indicates the direction of movement.

In Figure 4.31 it is possible to see that the model with 13 cluster of 4th order polynomial performed worse due to the two points misclassified at the end of the line, which resulted in higher mean error, differing from the other models in almost 2mm. Although it performed also worse in the circle trajec-

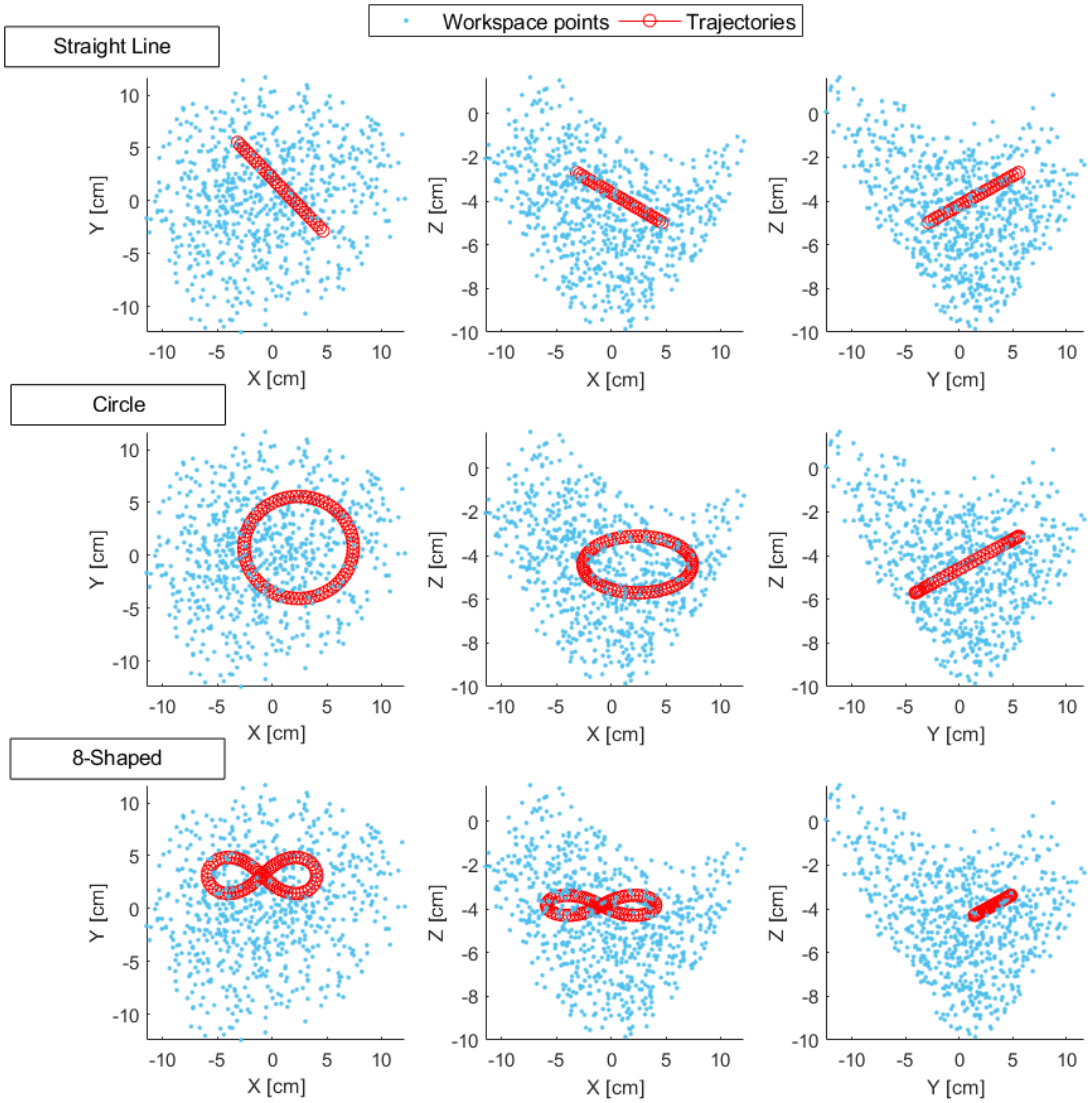


Figure 4.30: Proposed trajectories' position in xy, xz and yz views in the workspace of the robot.

tory (Figure 4.32), the difference between the errors of the models decreased and none of the models showed errors higher than 20mm. Since this combination of cluster and order of polynomial using the second data set presented the worst performance for the line and circle shape, it was expected that it would have resulted in higher errors also for the 8-shape trajectory, however it showed the best result. In Figure 4.33 one may see that the 8 clusters and the 14 clusters models, specially in the beginning of the trajectory struggle to reach the goal points, giving errors of more than 20mm and contrasting with the rest of the points.

Although simple, the straight line trajectory offered difficulties for the performance of the algorithm,

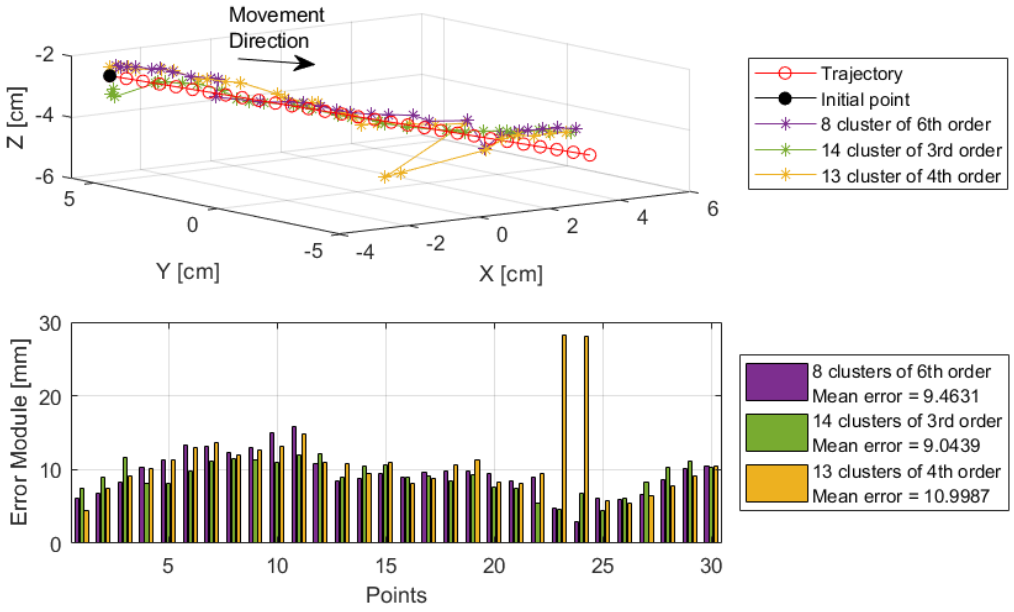


Figure 4.31: Straight line trajectory performed by the three selected models and its errors through the evolution of the path.

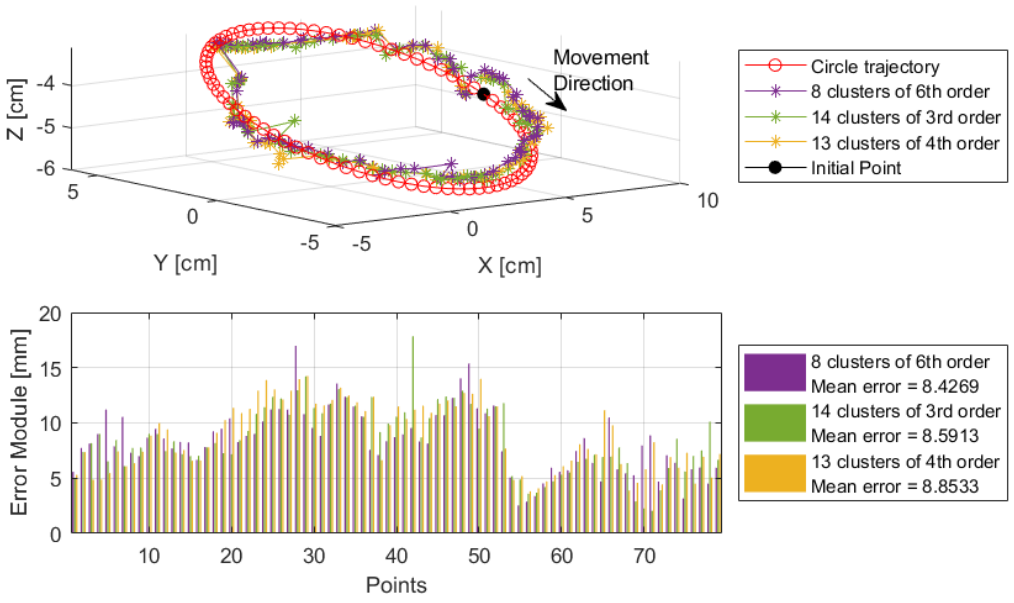


Figure 4.32: Circle trajectory performed by the three selected models and its errors through the evolution of the path.

presenting higher errors when compared to the circle and 8-shaped trajectories. This could be the result of the misclassification of the pre-trained classification tree. As explained in Chapter 2 it labels the points on clusters based on the data selected for training, which not necessarily give the best classification. The

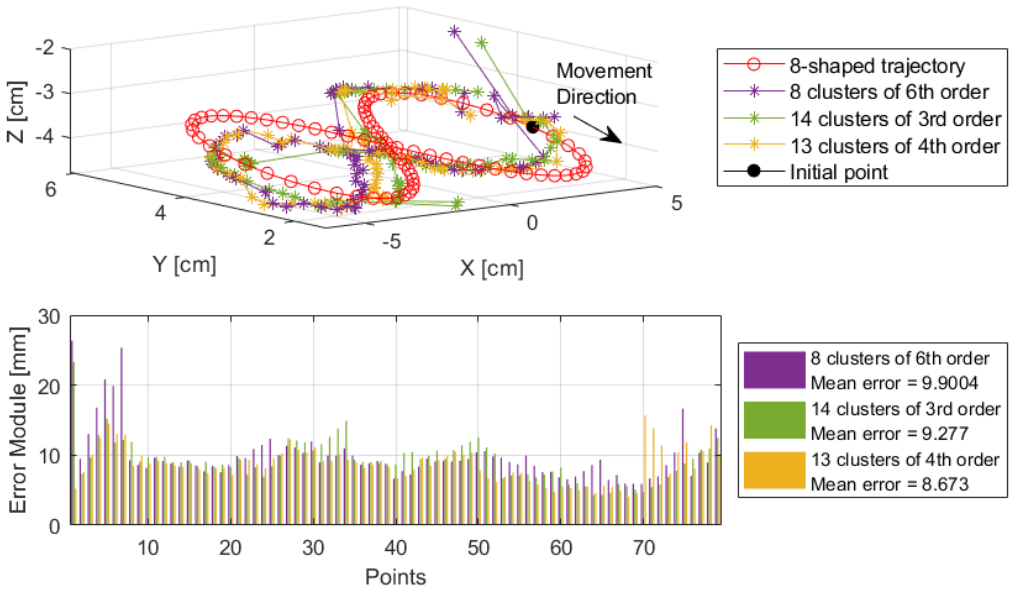


Figure 4.33: 8-shaped trajectory performed by the three selected models and its errors through the evolution of the path.

data random selected for training could also contribute to the higher errors, since the other trajectories do not pass through the same points of the straight line, as it can be seen in Figure 4.30, if the selection resulted in low density of the workspace points where is located the line, this could lead to greater prediction model errors and consequently worst performance.

The mean error and the standard deviation of the point-to-point trajectories for all the data sets tested can be seen in Table 4.4, the marked cells represent the lowest error for each model in each trajectory. Observing the marked cells, it cannot be stated that one data set was better than the others as well as the models. Although the same data set, the second one, resulted in the best performance for all the trajectories, the models with which it was obtained were different, for the straight line with the model with 14 clusters of 3rd order polynomial, for the circle, with the model with 8 clusters of 6th order polynomials and for the 8-trajectory with the 13 cluster of 4th order polynomials. Analysing the model with 8 cluster of 6th order polynomial, it can be seen that the second data set gave the best results for the straight line and the circle, but not for the 8-trajectory which it was obtained with the third data set. The same behaviour happened with the others models with different data sets.

Comparing the experimental results of Table 4.4 with the theoretical ones from Table 4.2, one may noticed that the experimental errors are around 4, 5 times bigger than the theoretical results of the direct kinematic model. It was already expected higher errors, since the pneumatic inputs for the robot are obtained from the inverse kinematic model that has an associated error, these inputs are then sent to the robot which, in addition to its own natural frequency oscillation, presents external noise, such as the

Table 4.4: Mean error and standard deviation for all the data sets randomly selected, models obtained and trajectories tested.

		Point-to-point Trajectory Error [mm]		
Model	**Data Set**	**Straight Line**	**Circle**	**8-Trajectory**
8 cluster of 6th order	**1**	12.70 ± 5.11	9.34 ± 3.50	11.46 ± 2.37
	2	9.46 ± 2.92	8.43 ± 2.81	9.90 ± 3.71
	3	10.22 ± 3.19	8.94 ± 2.94	9.18 ± 2.85
	4	10.41 ± 2.33	8.91 ± 3.40	9.79 ± 5.88
	5	10.28 ± 2.15	8.86 ± 3.44	9.96 ± 3.43
14 cluster of 3rd order	**1**	10.52 ± 3.05	8.80 ± 3.74	10.21 ± 3.74
	2	9.04 ± 2.16	8.59 ± 3.06	9.28 ± 2.93
	3	10.50 ± 2.49	9.80 ± 3.86	9.88 ± 3.21
	4	10.39 ± 2.42	8.97 ± 3.24	9.15 ± 1.98
	5	10.45 ± 2.89	8.88 ± 3.30	9.07 ± 2.40
13 cluster of 4th order	**1**	10.19 ± 2.61	8.53 ± 3.65	10.35 ± 1.78
	2	11.00 ± 5.27	8.85 ± 3.05	8.67 ± 2.49
	3	9.43 ± 2.02	9.22 ± 3.30	10.51 ± 3.48
	4	9.89 ± 2.45	8.73 ± 3.47	9.39 ± 2.56
	5	9.96 ± 2.52	9.12 ± 3.36	9.15 ± 2.34

oscillations introduced the operation of the pneumatic set. Besides, the direct kinematic errors of the Table 4.2 are obtained direct from a comparison between the models output and testing set of data, if the trajectories' points were used as inputs to inverse kinematic model and the output was used as input to the direct kinematic model, the position errors would be naturally bigger than the ones in the Table 4.2, since it would be considered two associated errors from the direct and from the inverse kinematic model.

5

Conclusion

This thesis presents the construction, testing and control of a continuum soft robot to be used in the I-SUPPORT system. This work proposes the use of silicone actuators instead of McKibben-based ones in the robotic arm of the system, aiming to offer more reliability and easy maintenance to the manipulator. By obtaining the kinematics models of the robot through polynomial regression method based on Holsten et al. [36], an open-loop controller is proposed that does not require prior knowledge about the robot.

The design was based on the I-SUPPORT manipulator developed by Ansari et al. in [33], trying to obtain similar length and structure when changing the actuators. The manufacturing of the proposed silicone actuators is easy, inexpensive and fast, it only needs the silicone bellows, scissors and the specific glue. In case it presents some small holes in the surface during use, it could be easily repaired with the glue to cover them. Due to this characteristics, a symmetrical shape for the robot is obtained without any major redesign effort.

The silicone offers less stiffness when compared to the McKibben-based actuators. Studying the single actuators, it was observed that this characteristic offers a more linear relationship between pressure and elongation, but it also implies in more time to reach the steady state condition for the silicone actuator.

The kinematic characterization of the robots assembled with the actuators showed some interesting results. Both robots presented a similar elongation, differing only by 0.418cm with respect to its initial length. When bending due to one chamber, the silicone assembled robot showed smaller upwards displacement, but when bending due to two chambers, the displacements are similar.

The added tendons allowed the silicone manipulator presented better results when compared to the McKibben-based. It reaches a maximum bending of 15cm upwards activating two chambers and showed a bending over itself. This offers promising results for future works with the hybrid actuation, combining the pneumatic and cable-driven actuators, which will increase the stiffness in the manipulator, probably reducing its time to stabilize.

It is important to highlight here the ability of the silicone actuator to work with negative pressures or vacuum, which is not possible for the McKibben-based actuators. This ability could enhance the kinematic characteristics observed during the experiments by the silicone manipulator, such as improving the bending capabilities and increasing the overall workspace. Therefore, the combination of negative and positive pressures, not used in this project due to availability, could be also addressed in future works.

The symmetrical shape achieved in the manufacturing of the actuators, reflected in the reachable workspace and the repeatability of the robot. The experiments showed for the silicone robot a symmetrical motion in every direction, although slightly lower reachability than the manipulator assembled with the McKibben-based actuators. The silicone assembled also presented better results regarding the

repeatability, offering more reliability, which makes it easier to modelling and control.

The model-free control was tested obtaining the inverse and direct kinematics models for the robot for different sets of training data and combinations of polynomial order and clusters. In all the experiments the controller presented good results in describing the point-to-point trajectories proposed with mean errors varying from 8.5 to 12mm. These results could be improved implementing a closed-loop control.

However, the experiments revealed the challenge of defining a unique combination of order and clusters that best describes the robot's kinematics, depending on the trajectory to be described, one combination stands out in relation to the others. Applying machine learning techniques to choose, between a restricted selection of models, the model depending on the task to be executed could offer a solution to this problem.

The thesis studied the behaviour of one module of the I-SUPPORT manipulator. Future works could couple the second module, assuring that the number of samples acquired allows a sufficient variation of polynomial order and clusters to be applied the control method based on polynomial regression. The hope is that using this work, a more structured approach can be taken to analyze and tackle the control problems related to the proposed silicone actuators in the I-SUPPORT robotic arm.

Bibliography

[1] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics - Modelling, Planning and Control*. Springer-Verlag London Limited, 2009.

[2] M. Cianchetti, A. Arienti, M. Follador, B. Mazzolai, P. Dario, and C. Laschi, "Design concept and validation of a robotic arm inspired by the octopus," *Materials Science and Engineering: C*, vol. 31, no. 6, pp. 1230–1239, 2011.

[3] H. T. Lin, G. G. Leisk, and B. Trimmer, "GoQBot: a caterpillar-inspired soft-bodied rolling robot," *Bioinspiration & Biomimetics*, vol. 6, no. 2, p. 026007, 2011.

[4] B. A. Jones, M. William, and I. W. Clemson, "Design and Analysis of a Novel Pneumatic Manipulator," *IFAC Proceedings Volumes*, vol. 37, no. 14, pp. 687–692, 2004.

[5] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-Octor," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 2578–2585.

[6] M. Rolf and J. J. Steil, "Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2012, pp. 3440–3446.

[7] S. Seok, C. D. Onal, R. Wood, D. Rus, and S. Kim, "Peristaltic locomotion with antagonistic actuators in soft robotics," in *2010 IEEE International Conference on Robotics and Automation*, 2010, pp. 1228–1233.

[8] G. Robinson and J. B. Davies, "Continuum robots - a state of the art," in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 4, 1999, pp. 2849–2854.

[9] S. Kim, C. Laschi, and B. Trimmer, "Soft robotics: A bioinspired evolution in robotics," *Trends in Biotechnology*, vol. 31, no. 5, pp. 287–294, 2013.

[10] Y. Bailly and Y. Amirat, “Modeling and control of a hybrid continuum active catheter for aortic aneurysm treatment,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 924–929.

[11] N. Simaan, “Snake-Like Units Using Flexible Backbones and Actuation Redundancy for Enhanced Miniaturization,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, 2005, pp. 3012–3017.

[12] G. Immega, K. Antonelli, and J. Ko, “Teleoperation of the KSI tentacle manipulator for hot cell decontamination,” in *1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century*, 1995, pp. 2133–2136 vol.3.

[13] A. Yeshmukhametov, K. Koganezawa, and Y. Yamamoto, “A Novel Discrete Wire-Driven Continuum Robot Arm with Passive Sliding Disc: Design, Kinematics and Passive Tension Control,” *Robotics*, vol. 8, no. 3:51, 2019.

[14] H. Tsukagoshi, A. Kitagawa, and M. Segawa, “Active Hose: an Artificial Elephant’s Nose with Maneuverability for Rescue Operation,” in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, 2001, pp. 2454–2459 vol.3.

[15] C. Laschi, B. Mazzolai, and M. Cianchetti, “Soft robotics: Technologies and systems pushing the boundaries of robot abilities,” *Science Robotics*, vol. 1, no. 1, pp. 1–12, 2016.

[16] T. G. Thuruthel, “Machine Learning Approaches for Control of Soft Robots,” Ph.D. dissertation, Scuola Superiore Sant’Anna, 2018.

[17] S. Kolachalama and S. Lakshmanan, “Continuum robots for manipulation applications: A survey,” *Journal of Robotics*, vol. 2020, 2020.

[18] M. W. Hannan and I. D. Walker, “Kinematics and the implementation of an elephant’s trunk manipulator and other continuum style robots,” *Journal of Robotic Systems*, vol. 20, no. 2, pp. 45–63, 2003.

[19] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, “Continuum Robots for Medical Applications: A Survey,” *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1261–1280, 2015.

[20] T. Ranzani, M. Cianchetti, G. Gerboni, I. D. Falco, and A. Menciassi, “A Soft Modular Manipulator for Minimally Invasive Surgery: Design and Characterization of a Single Module,” *IEEE Transactions on Robotics*, vol. 32, no. 1, pp. 187–200, 2016.

[21] H. Su, D. C. Cardona, W. Shang, A. Camilo, G. A. Cole, D. C. Rucker, R. J. Webster, and G. S. Fischer, “A MRI-guided concentric tube continuum robot with piezoelectric actuation: A feasibility

study," *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 1939–1945, 2012.

[22] D. B. Camarillo, C. F. Milne, C. R. Carlson, M. R. Zinn, and J. K. Salisbury, "Mechanics modeling of tendon-driven continuum manipulators," *IEEE Transactions on Robotics*, vol. 24, no. 6, pp. 1262–1273, 2008.

[23] G. Immega and K. Antonelli, "The KSI tentacle manipulator," in *Proceedings of 1995 IEEE International Conference on Robotics and Automation*, 1995, pp. 3149–3154 vol.3.

[24] A. Stilli, H. A. Wurdemann, and K. Althoefer, "Shrinkable, stiffness-controllable soft manipulator based on a bio-inspired antagonistic actuation principle," *IEEE International Conference on Intelligent Robots and Systems*, no. Iros, pp. 2476–2481, 2014.

[25] I. A. Gravagne, C. D. Rahn, and I. D. Walker, "Large deflection dynamics and control for planar continuum robots," *IEEE/ASME Transactions on Mechatronics*, vol. 8, no. 2, pp. 299–307, 2003.

[26] D. Trivedi, A. Lotfi, and C. D. Rahn, "Geometrically exact dynamic models for soft robotic manipulators," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1497–1502.

[27] B. A. Jones and I. D. Walker, "Kinematics for Multisection Continuum Robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–56, 2006.

[28] D. B. Camarillo, C. R. Carlson, and J. K. Salisbury, "Configuration Tracking for Continuum Manipulators With Coupled Tendon Drive," *IEEE Transactions on Robotics*, vol. 25, no. 4, pp. 798–808, 2009.

[29] ——, "Task-Space Control of Continuum Manipulators with Coupled Tendon Drive," in *Khatib O., Kumar V., Pappas G.J. (eds) Experimental Robotics. Springer Tracts in Advanced Robotics, vol 54. Springer, Berlin, Heidelberg.*, 2009, pp. 271–280.

[30] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "Neural Network and Jacobian Method for Solving the Inverse Statics of a Cable-Driven Soft Arm with Nonconstant Curvature," *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 823–834, 2015.

[31] T. G. Thuruthel, E. Falotico, M. Manti, A. Pratesi, M. Cianchetti, and C. Laschi, "Learning closed loop kinematic controllers for continuum manipulators in unstructured environments," *Soft Robotics*, vol. 4, no. 3, pp. 285–296, sep 2017.

[32] Y. Ansari, M. Manti, E. Falotico, Y. Mollard, M. Cianchetti, and C. Laschi, "Towards the development of a soft manipulator as an assistive robot for personal care of elderly people," *International Journal of Advanced Robotic Systems*, vol. 14, no. 2, mar 2017.

[33] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, "Multiobjective Optimization for Stiffness and Position Control in a Soft Robot Arm Module," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 108–115, jan 2018.

[34] European Union's Horizon 2020. I-support. (Accessed: 15.10.2019). [Online]. Available: http://www.i-support-project.eu/

[35] X. S. Papageorgiou, C. S. Tzafestas, and P. Vartholomeos, "ICT-Supported Bath Robots: Design Concepts," in *17th International Conference on Social Robotics (ICSR 2015)*, 2015.

[36] F. Holsten, M. P. Engell-Norregard, S. Darkner, and K. Erleben, "Data driven inverse kinematics of soft robots using local models," *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2019-May, pp. 6251–6257, 2019.

[37] J. L. Devore, *Probability and Statistics for Engineering and the Sciences*, 8th ed. Brooks/Cole, Cengage Learning, 2012.

[38] T. T. Soong, *Fundamentals of Probability and Statistics for Engineers*. John Wiley & Sons, Ltd, 2004.

[39] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[40] J. S. Jang, C. T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft: a Computational Approach to Learning and Machine Intelligence*. Upper Saddle River, NJ: Prentice-Hall, Inc, 1997.

[41] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, vol. 07-09-January-2007, pp. 1027–1035, 2007.

[42] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook*. Springer New York Dordrecht Heidelberg London, 2010.

[43] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. CRC Press, 1984.

[44] J. R. Quinlan, "Simplifying Decision Trees," *International Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 221–234, 1986.

[45] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, 7th ed. McGraw Hill, 2001.

[46] P. T. Boggs and J. W. Tolle, "Sequential Quadratic Programming," *Acta Numerica*, vol. 4, pp. 1–51, 1996.

[47] MATLAB, *Statistics and Machine Learning Toolbox (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[48] "*NIST Digital Library of Mathematical Functions*," http://dlmf.nist.gov/, Release 1.0.28 of 2020-09-15, f. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. [Online]. Available: http://dlmf.nist.gov/

[49] Laub, A. J. The moore-penrose pseudoinverse. (Accessed: 08.10.2020). [Online]. Available: https://www.math.ucla.edu/~laub/33a.2.12s/mppseudoinverse.pdf

[50] MATLAB, *Optimization Toolbox (R2020b)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[51] Brilliant.org. Integer equations - stars and bars. (Accessed: 19.10.2020). [Online]. Available: https://brilliant.org/wiki/integer-equations-star-and-bars/#stars-and-bars

[52] M. Manti, A. Pratesi, E. Falotico, M. Cianchetti, and C. Laschi, "Soft assistive robot for personal care of elderly people," *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, vol. 2016-July, pp. 833–838, 2016.

[53] F. Daerden, "Conception and realization of pleated pneumatic artificial muscles and their use as compliant actuation elements," Ph.D. dissertation, Vrije Universiteit Brussel, 1999.

[54] *Soft Manipulator - User Manual*, The BioRobotics Institute - Sant'Anna School of Advances Studies, Viale Rinaldo Piaggio, 34 - 56025 Pontedera(PI), Italy.

[55] A. Grzesiak, R. Becker, and A. Verl, "The Bionic Handling Assistant: A success story of additive manufacturing," *Assembly Automation*, vol. 31, no. 4, pp. 329–333, 2011.

[56] SMART-DX Motion Tracking Systems. (Accessed: 15.11.2020). [Online]. Available: https://www.btsbioengineering.com/products/smart-dx-motion-capture/

[57] *Aurora V2 User Guide - Revision 3*, Northern Digital Inc., Waterloo, Ontario, Canada N2V 1C5, July 2012.

[58] R. J. Hyndman, "Moving averages," *Real Analysis Exchange*, vol. 33, no. 1, pp. 29–30, 2009.