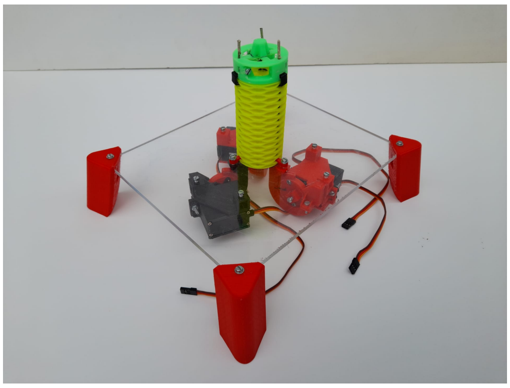


Design and experiments on a continuous deformation soft manipulator module

João Maria Cruz Azevedo Bastos Salgueiro

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisor: Prof. João Carlos Prata dos Reis

Examination Committee

Chairperson: Prof. Carlos Baptista Cardeira
Supervisor: Prof. João Carlos Prata dos Reis
Members of the Comittee: Prof. António Manuel Relógio Ribeiro
Prof. Jorge Manuel Mateus Martins

January 2021

”Valeu a pena? Tudo vale a pena
Se a alma não é pequena.
Quem quer passar além do Bojador
Tem que passar além da dor”
Fernando Pessoa, in Mensagem

Acknowledgments

First, I would like to thank Professor João Reis for all the invaluable help and guidance provided throughout this work and beyond. For offering me this challenge and allowing me to dive in a developing technology, that I believe will have a role in different aspects of human activities.

To IST that for the last few years prepared me, put me through tough situations and sleepless nights but provided me with a playground where I could put in practice and developed most of my skills.

To my friends that direct or indirectly helped my journey. The ones who made my daily IST routine, the likes of: Pedro Ferreira, Miguel Santos, João Castro, Duarte Valente, Pedro Rodrigues. And the long lasting friendships: Daniel Delgado, Diogo Carneiro, Rita Ferreira, Diogo Pinto, Pedro Pereira.

Finally I couldn't be the person that I am today without my family, more precisely my parents, Vera and João Salgueiro. They provided me with a solid foundation to grow and were always supporting me whether I knew it or not. They taught me important values and provided me with so many opportunities that I will never be able to make up to them.

Wanted to also thank NTNU for welcoming me in Norway, the professor Jorge Martins and the researchers João Oliveira and João Gouveia for the help given in collecting the necessary data.

Resumo

Neste trabalho desenvolveu-se e testou-se um manipulador macio com uma estrutura com resistência torsional que permite resistir a cargas excêntricas.

Começou-se por projetar e desenvolver um manipulador com uma geometria de *wave spring* que permite um aumento da rigidez torsional enquanto mantém a sua capacidade de extensão e compressão. O valor do número de ondas por nível foi obtido a partir de uma análise de elementos finitos estática e dinâmica. Após a definição dos parâmetros os elementos foram modelados e impressos em 3D usando filamentos rígidos e flexíveis.

Foram realizados vários testes experimentais, num primeiro teste foi obtido o espaço de ação 2D, num segundo conjunto de testes foi avaliada a repetibilidade do manipulador. Por fim foram realizados testes estáticos e dinâmicos com diferentes cargas.

Finalmente foram aplicados aos modelos obtidos a partir do treino de redes neuronais três trajetórias. Os modelos podem ser divididos em: modelos treinados com um número variado de dados do modelo teórico; modelos também treinados com um número variado de dados mas recolhidos do manipulador; e modelos treinados com a totalidade dos dados do manipulador no entanto com um número variado de neurónios.

Este estudo permitiu concluir que é possível projetar uma estrutura para um robô contínuo macio (soft) de forma a maximizar a rigidez associada à torção, sem sacrificar significativamente os graus de liberdade que se pretendem complacentes. Também se demonstrou que é possível a um robô contínuo realizar manipulação ”off-axis” com um valor de carga útil interessante do ponto de vista prático.

Palavras-chave: Robôs Macios, Robôs continuos, Mola ondulada, Manufatura aditiva, Redes neuronais

Abstract

In this work it was developed and tested a soft robot with an improved torsional stiffness capable to withstand torsional loads.

The design of the manipulator was based on a wave spring that allows an increment of its torsional stiffness while maintaining its capacity to extend and compress. The number of waves per level was obtained from a static and dynamic finite element analysis. With all the parameters defined the elements were modelled and 3D printed with rigid and flexible filaments.

Several experimental tests were performed. The first test obtained the work volume, the second test evaluated the repeatability of the manipulator, and lastly, a static and dynamic loading test was performed.

In the end three trajectories were applied to the inverse kinematics models obtained from the training of neural networks. These models can be divided into: models trained with a varied number of data points from the theoretical model; models trained also with a varied number of data but from the manipulator; and models trained with all the manipulator data but with a variable number of neurons.

From this study it was possible to design a structure for a continuum soft robot able to maximize its torsional stiffness without a significant loss of compliance in the degrees of freedom where it is desirable. It was also demonstrated that a soft robot is capable to have an off-axis loaded motion while carrying at least a small but useful payload.

Keywords: Soft Robots, Continuum robots, Wave spring, Additive manufacturing, Neural Networks

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	Rotation angle.
β	Angle of actuator motor.
Δ	Difference.
ϵ	Strain.
ϕ	Orientation angle.
σ	Stress.
θ	Leaning angle.
$Oz\varphi$	Plane made with the z axis and the orientation angle ϕ.

Roman symbols

ANN/NN	Artificial Neural Networks.
b	Bias.
b_r	Actuation radius of the actuators.
C	Curvature center coordinates.
deg	Degrees.
E	Young's Modulus.
FDM^{TM}	Fused Deposition Modelling.
FFF	Fused Filament Fabrication.
l_w	Wire or actuator length.
N_j^k	Hidden Neuron.
o'	Origin vector for an offset frame.
O_j^k	Ouput Neuron.

p	Displacement vector.
R	Rotation Matrix.
r	Neutral line curvature radius.
R_i	Actuator curvature radius.
TPE	Thermoplastic Elastomer.
ML	Machine Learning.
AI	Artificial Intelligence.
DOF	Degrees of freedom.

Subscripts

$i = 1 - 3$	Actuator numbering.
j	Neuron number.
k	Hidden layer number.
n	Neutral position.
vm	von Mises.
x, y, z	Cartesian components.

Superscripts

-1	Inverse.
T	Transpose.

Chapter 1

Introduction

1.1 Motivation

Nowadays, robots, machines that on demand fulfill tasks autonomously, play a significant role in the world around us. They are present in the most diverse areas of work from medicine, [1], [2], [3], [4], to the manufacturing industry, [5].

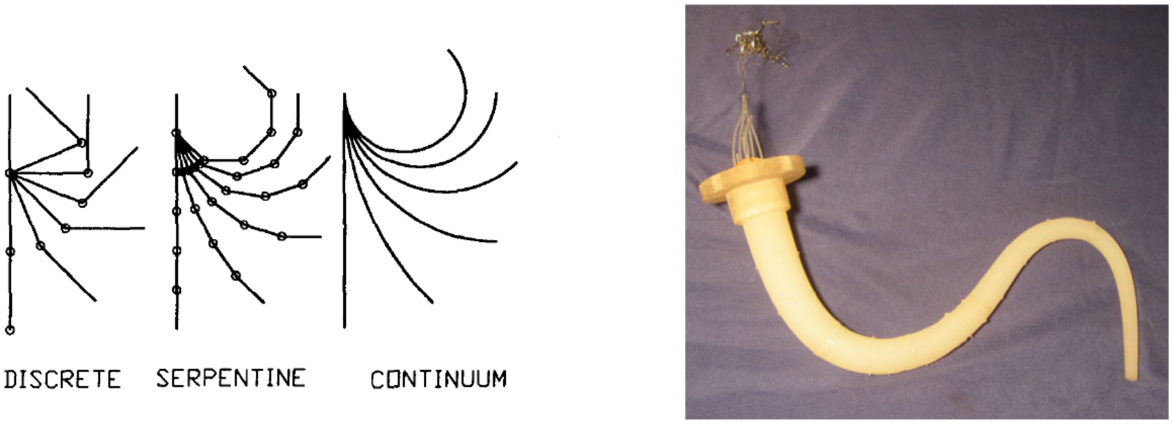
A robot, according to the dictionary of Cambridge, is defined as *a machine controlled by a computer that is used to perform jobs automatically*. With this definition it is more clear to identify the images of figure 1.1 as examples of different types of robots.

(a) OctArm - [6] (b) Industrial Robot - UR3 (c) Da Vinci Surgical robot

Figure 1.1: Example of different types of hard robots

As one can observe, in the previous figures, the majority of robots is made of rigid structures. This feature is of significant importance to obtain precise, strong, reliable and durable robots. The manufacturing industry is no exception to the use of robots that have gained an increased importance in the most diverse assembly lines. These industrial robots are characterized by their precision and accuracy, their discrete rigid links, their considerable weight and required clearance volume to operate. Nonetheless, if they happen to operate alongside humans or any other living form some of these characteristics may raise security concerns. Another setback resultant from the typical discrete rigid robots is their limitation to interact with the surrounding environment, [7].

The constraints previously mentioned can be overcome if a different approach is considered based on continuum robots. This type of robots can be defined as a continuously bending robot without any rigid links and identifiable rotational joints, [8] and [9]. Figure 1.2, from [9], displays the main difference between continuum and discrete robots. Continuum robots, due to its architecture, are able to exploit situations where neither the robots with rigid structure nor the humans are capable to operate, these include scenarios in unstructured and constrained environments. One of the reasons is that these robots are considered to have an infinite number of degrees of freedom (DOF) and since not all DOF are actuated, the robot will exhibit some form of passive compliance. Despite this passive compliance displayed appearing to be a disadvantage when tasks with precision and accuracy are required, it becomes a valuable asset for safer interactions and in fragile environments, [9].



(a) Continuum Robots - [9]

(b) Soft Continuum robot inspired by an octopus tentacle. - [10]

Figure 1.2: Continuum robots and soft robots

Like many things that humans build, biology has always been a source of inspiration and the field of continuum robots follows suit. The inspiration to this field can be drawn from animals and plants as they result from an evolution with millions of years. Biology may then be considered an important source of intellectual material to be used in engineering, from the design of a bullet train mimicking a kingfisher's peak to a tentacle inspired soft robot, [11], [12].

An alternative to the usual rigid structures for the continuum robots, with inspiration drawn from biology, can be accomplished with soft materials. Soft robots, as they are called, use soft materials as its structure and main body instead of the usual discrete rigid links. With these softer materials the robot inherits several of its characteristics, namely its softness and flexibility. These inherited properties allow the soft robot to have a safe interaction when handling fragile and soft elements, and when operating in a delicate environment without the need to have a range of sensors and a complex controller, [13]. These materials are able to absorb external forces and deform accordingly with the contact surface, [9].

1.2 Robots Classification

In the field of robotics, the robots can be mainly classified by two criteria: the materials used and the number of DOF. An overview of the several classes can be seen in the figure 1.3. It is possible to understand that two of the major classifications for a robot are: either a hard or soft robot and either a discrete or continuum robot.

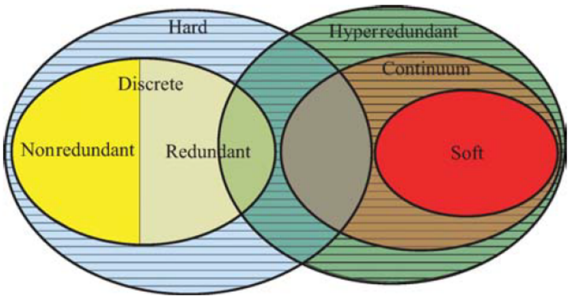


Figure 1.3: Robot classification - [7]

The more clear and intuitive differentiation is between hard and soft robots. An example of a discrete hard robot and of a soft robot can be, respectively seen in the figure 1.1 b) and figure 1.4. Hard robots are the most common type of robots and are made of rigid links, while soft robots are made of more compliant, flexible and stretchable materials. Both of these types of robots have their unique advantages. Hard robots, due to the characteristics of the rigid links, are able to have more precise movements, carry larger loads and are more easily controlled. Soft robots, on the other hand, have more flexibility, dexterity, ability to maneuver complex pathways and are more susceptible to external forces, [5]. The resultant advantages from the soft materials, such as a safer workspace manipulation, will be discussed further down this work. Nevertheless, in certain circumstances this type of robot may require a more complex control when comparing to the typical discrete hard robots.

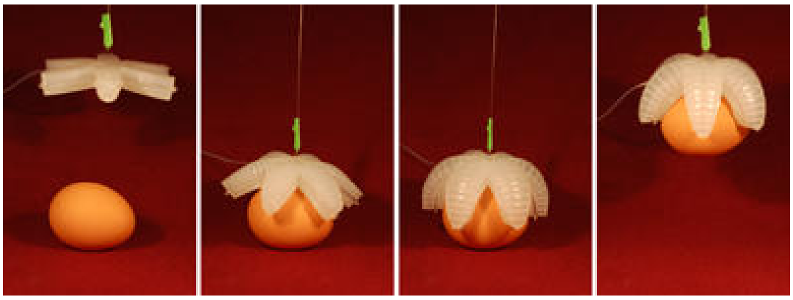


Figure 1.4: Example of a soft robot - [14]

The lack of rigid structures in the soft robots results in a continuous deformation hence in an infinite number of DOF. This system can be considered under-actuated since it does not have an actuator for every degree of freedom.

The most common type of hard robots, used in well-defined environments, are formed by discrete and rigid links connected by one degree of freedom joints, [7]. The movement of these robots is generated in the joints, they can either be rotational or translational, and each provides one DOF, [9]. On the other side, if a robot is not considered discrete it can be classified as continuum. A continuum robot can be defined as a continuously bending structure, [8]. These robots are becoming increasingly studied due to its complementary characteristics to hard robots. These include dexterity in a constrained and unstructured environment and a compliant interaction with external contacts, in contrast to the rigidity and precision from the latter.

A last term that, in the figure 1.3, defines a robot is its redundancy. A non-redundant robot is considered to be a manipulator with the minimum number of DOF, joints, for the fulfillment of a certain task. A redundant robot, differently, can be defined as a robot where the number of DOF is bigger than the number required. When the number of joints is largely superior to the required, the robot is considered to be hyperredundant with high dexterity. However, if the joints are placed in order that it forms a continuously bending structure it can be classified as a continuous hyperredundant hard robot. These hard continuum robots, very often inspired by biological soft structures, do not provide the most efficient and smoothest continuous motion (a hard robot inspired by an elephant trunk can be seen in figure 1.5).

Figure 1.5: Elephant trunk manipulator - [7]

A soft robot is automatically considered to be a continuum hyperredundant robot due to the type of material used, the method of actuation and the infinite number of DOF. As written before, a soft continuum robot does not have rigid links, and so bends continuously producing a smooth curvature. When a robot has infinite number of DOF it also means that it can take an infinite number of poses to reach the same position in space, [7]. Some examples of continuum robots can be found in [8], with several applications such as in medicine, material transportation, search and rescue, and more.

A further classification of the continuum robots is made regarding the actuation method and location of the mechanical actuators. They can be separated in three categories: Intrinsic, Extrinsic and Hybrid. In an intrinsic robot, the actuators are part of the moving mechanism. The simplest form of this actuator can be achieved by a single pneumatic or hydraulic chamber that operates within a single plane. This planar movement is accomplished by the different thickness of the walls around the chamber. To obtain a spatial movement one can gather several individual chambers in different orientations. For an extrinsic robot, the actuator is not part of the animated mechanism. The action is transferred, from the actuators to the flexible module, through a mechanical linkage such as tendons. This type of robots results in a lightweight and, if desired, hollow device. The hollowness of such robot is an advantage that can be taken for namely protect the electric cables inside the robot and, transport either liquids or bulk materials.

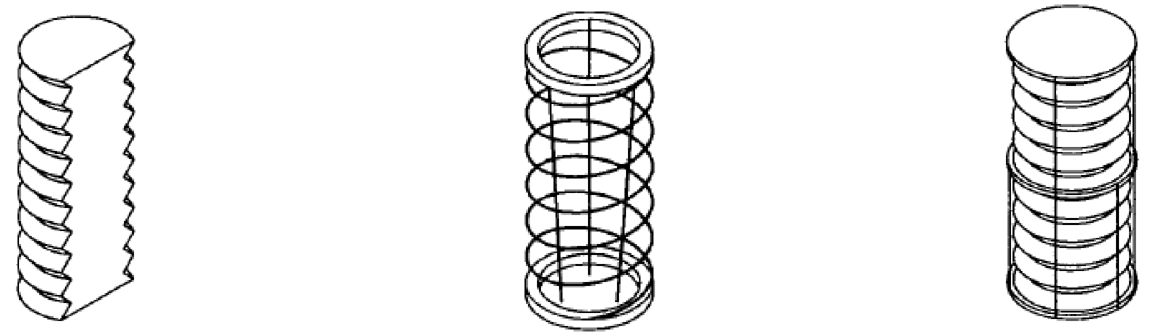

(a) Intrinsic Actuator (b) Extrinsic Actuator (c) Hybrid Actuator

Figure 1.6: Continuum Robot Classification according to the actuation method - [9]

As for the hybrid devices they are a combination of both extrinsic and intrinsic devices, [9].

The actuation method of the hard robots are less complex as they are applied, in a joint, to a rigid link that, due to its properties, propagates the motion to the remaining elements. For the soft robots, however, the method of actuation should differ from the previous methods as the materials used as the main body are not able to propagate the action when it is applied to one extremity. Some of the most common methods to actuate a flexible link are the use of a pneumatic or hydraulic chamber and the use of cables actuated remotely by servo motors, [15]. The latter actuation method can also be applied to an hyperredundant continuum robot. Continuum robots, especially soft robots, due to its large inspiration by nature, are able to better interact with the surrounding environment when compared to discrete hard robots. Nonetheless, soft robots also suffer more with the environment's loads.

The last parameter that can be modified is the overall length of the robot. A manipulator can have a fixed length, based on a backbone or similar structure, or a variable length, which is typically not supported by a backbone, [15]. Depending on the option chosen it will result in consequences on the behaviour of the robot.

References [5], [9], [16] and [17] present a complete survey of the literature published so far in regards to continuum robots. It explores the advantages and drawbacks, as well as the classification according to its inspiration, structure, material and actuators alongside several examples for each.

1.3 Objectives

The previous section briefly introduced the reader to the field of the proposed work. From that, the first important step is to define the research question and the goals that one will try to answer and achieve.

In the spectrum of continuous robots one could either choose a rigid hyper-redundant robot or a soft robot approach to obtain such type of manipulators. In this thesis the approach followed was with soft robots. This category of robots are made out of malleable materials that do not have a significant torsional stiffness, [15], [18]. In an off-axis scenario this reduced torsional rigidity may result in undesirably large deformations. Such deformations pose as one of the main challenges when developing these types of manipulators.

A research question therefore arises, can one design a soft continuum manipulator structure that can exhibit a desirable level of passive torsional stiffness, while keeping the bending and extensional stiffness sufficiently low to allow for practical actuation?

In the present work, it was proposed to develop a soft robot with a geometry that aims to positively answer this question. In that context the geometry used and optimized was inspired by the design of a wave spring.

Figure 1.7: Wave Spring [19]

To answer the research question, several goals were outlined. First it is required to design and assemble the manipulator in question. In this regard the objectives are:

- Design a 3D Model of the manipulator with all the necessary components;
- Perform a finite element analysis, to evaluate and choose one of the several candidate geometries;
- Manufacture and assemble the robot.

Afterwords, one must study and assess the capabilities of the proposed manipulator module itself. With this aim the following objectives were devised:

- Test the properties of the manipulator module;
- Obtain a kinematic model of the working prototype;
- Test the obtained model and the prototype with several trajectories;
- Analyse the overall performance of the manipulator module.

1.4 Thesis outline and contributions

The main contributions from this work are as follows: the design and construction of a soft continuum manipulator module capable to sustain torsional loads; a lightweight and fully working prototype actuated by semi-rigid elements (cables); an optimization of the number of waves per level of the wave spring based on simulation results, with an off-axis manipulation capable to sustain loads with more than 4 times its own weight; the use of an additive manufacturing process, 3D printing, to produce a soft continuous module; the characterization of the prototype repeatability and of motions in off-axis scenarios with several loads; and finally, the kinematic modelling of the prototype using neural networks with a variable number of training points and neural networks with a variable number of neurons, with resultant mean errors in the same order of magnitude as the repeatability radius.

Some of the previous achievements were submitted in an article to an international conference which was later accepted for publication, Salgueiro and Reis [20].

The present work is structured in:

- A chapter 2 where it is analysed the state of the art and researched relevant topics for the development of the prototype;
- Chapter 3 describes the process of development of the manipulator, it includes the challenges that have been overcome, a finite element analysis and the process of 3D printing;
- The next chapter, chapter 4, describes several kinematics studied and developed, namely theoretical models and models computed with neural networks;
- Chapter 5 focuses in exploring the manipulator properties with several experiments. It is studied the work volume of the manipulator, the repeatability, the static and dynamic performance when loaded. The last experimental work features the behaviour of the manipulator when submitted to different trajectories with different inverse kinematics models.
- In the last chapter, 6, it is made a summary of the experimental work, the achievements and future works.

Chapter 2

Background

2.1 Theoretical Overview

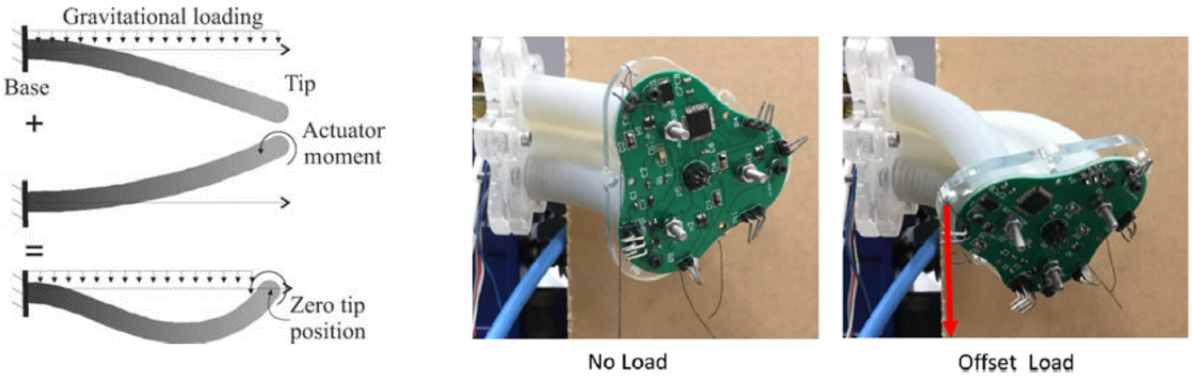
2.1.1 Soft Continuum Robots

In this section the focus will be directed to the guidelines that need attention when modelling soft continuum robots. This process requires additional attention to details that otherwise would be considered negligible. A model of a common rigid robotic manipulator assumes that its structure is infinitely rigid, and therefore all displacements are generated by the controlled motion of the joints. For a soft robot, on the other hand, one needs to ensure the desired deformation while considering the external forces that the robot should withstand. For example, if a robot is actuated by a pneumatic or hydraulic actuators the structure in rest won't need to withstand much load as this will be supported by the actuators. If the robot is a tendon based actuator the flexible structure should withstand the majority of the external forces as the tendon do not have the same structural stiffness.

One of the most important property inherited by the soft robots is their passive compliance. This behaviour is useful because when an external force or contact is applied, the robot will deform under that opposing force rather than generate a large contact force. Such behaviour is extremely important when considering a safe manipulation of a fragile or unknown object and when interacting with a constrained, unstructured and delicate environment without any feedback control, [4], [9]. A demonstration of the importance of this feature is the addition of a compliance to the traditional rigid manipulators by control methods, [21], while the soft manipulators inherit this property.

Nonetheless the importance of such characteristic, it results in a deviation from the desired position when external forces are applied, namely gravity. If one is able to model or predict these forces, the accuracy of the manipulator will improve. Figure 2.1 a) exemplifies a gravitational loading applied on a soft robot and its desired response to obtain the desired position. The compliance of the manipulator is defined by the stiffness of the material used and the pose of the manipulator, [4].

Contrary to the case of figure 2.1 a), when the load is applied in an offset scenario, figure 2.1 b), most soft manipulators tend to give in since their material and structure do not have torsional resistance, and it can not be counteracted with the action of the actuators. As a consequence of the previous flaw,



(a) Steps needed to counteract the gravitational displacement - [7]

(b) Example of an offset loading - [18]

Figure 2.1: Examples of the passive compliance from the soft robots

the attempt to realize a planned load motion will likely result in some problematic scenarios and so an alternative approach is required to solve it. These scenarios are not the most exploited in the literature, [18].

2.1.2 3D Printing

When designing a new prototype one should consider the manufacturing process and its limitations. To develop a prototype it is very often necessary to iterate several times the design as some parameters need to be changed or improved. For this, a manufacturing process that reduces the cost and production time is extremely welcomed. For that reason 3D printing, an additive manufacturing technology, is gaining an increasing interest as it allows the fabrication of complex architectures without the need for expensive processes and long delays. FDM$^{\text{TM}}$ (fused deposition modeling or FFF - fused filament fabrication) relies on the deposition of thin layers of melted material to build the model layer by layer where the coming layer is fused together with the previous. The material used for 3D printing is able to switch between a melted and solid state without loosing its mechanical properties. The filament of this material is then fed into the 3D printer extruder that melts the material and then deposit it in the desired position using a tridimensional mechanical system.

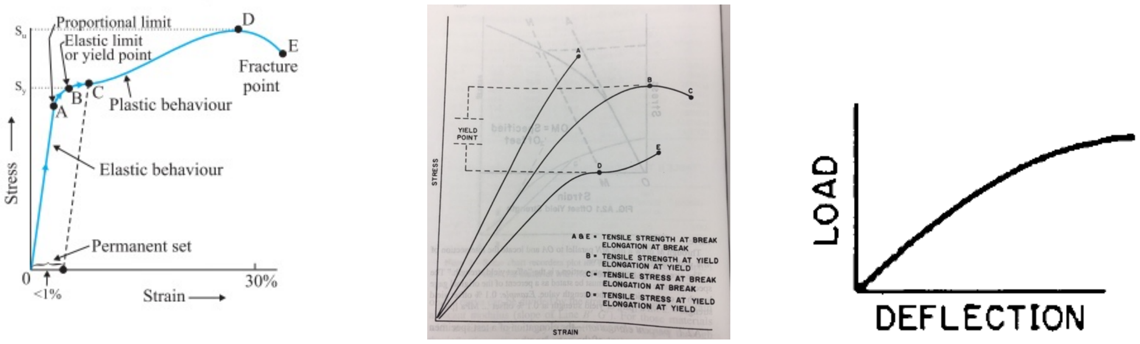
The mechanical properties of the final model produced by 3D printing will be poorer than those of its original material as they will depend on the most varied parameters, like the extrusion temperature, the layer parameters, orientation and many more. Several materials are available on the market, from rigid thermoplastics to flexible filaments. Some examples of available materials are: PLA (Polyactic Acid), PETG (Polyethylene Terephthalate Glycol-modified), ABS (Acrylonitrile Butadiene Styrene), ASA, Nylon, Filaflex and more, [22].

In Skorina and Onal [18] and Yap et al. [23] is 3D printed a soft robots and soft pneumatic actuators with a flexible material, for different applications, [23] also proposes guidelines for the printing parameters of a flexible filament.

2.1.3 Material Analysis

A key point in the process of designing and testing new prototypes is the choice of the most suitable materials. A commonly used test to obtain the properties of a material is the uni-axial tension test. The data acquired can be plotted in a stress-strain graph, figure 2.2. The first graphic of this figure is the typical response of a metal while the second depicts several of the possible behaviours that a polymer may have and the last reflects the reaction of an elastomer. In these graphs two different types of behaviour may be identified, an elastic and a plastic. An elastic behaviour occurs when a material after the stress ceases to be applied is able to recover to its initial shape without any remaining deformation. On the other hand, a plastic behaviour is a behaviour that will result in permanent deformations and the material is not capable to recover its original shape.

An elastic behaviour can be divided in a linear and non-linear trend. Figure 2.2 a) have a linear trend, from 0 to A, and follows the Hooke's law, $\sigma = E \times \epsilon$, where the stress is proportional to the strain by a factor of E. The constant E can be labeled as Modulus of Elasticity, Tensile Modulus or, as it is better known, Young's Modulus, and is a property that allows to make comparisons between stiffness from different material. Continuing in the same graph, the stress limit of the elastic behaviour is in B, the elastic limit or yield point, with similar points in 2.2 b), after this the material will enter in a plastic behaviour where the original shape is no longer recovered naturally. From this point onwards the specimen will continue to deform until it reaches the point D, from which the deformations ceases to be uniform to be concentrated on its weakest point originating a necking phenomenon. This maximum stress applied to the material, before it starts to fail, is designated as the ultimate tensile strength For the third plot of figure 2.2, we have a non-linear relation between the stress and strain of an elastomer.



(a) Typical metal - [24]   (b) Polymers - [25]   (c) Elastomers - [26]

Figure 2.2: Example of Stress-Strain curves for different materials

The polymers can behave in several different ways and figure 2.2 b) illustrates three examples. On the three curves it is possible to see an initial evolution of the graph following Hooke's law. The curve A is a result of a rigid and brittle material. From the two remaining curves it is possible to draw some similarities from the stress-strain curve of a typical metal. In these, both have a proportional limit and a yield point, however can evolve in to two different scenarios. One very similar to the behaviour of a typical metal, figure 2.2 a), and the other where the stress continues to increase until a fracture is obtained, [25], [27].

Depending on the type of curve obtained the variables to be taken into account will be different. If one gets curve of A, the important value would be the tensile strength at break. If one gets on of the remaining curves, probably, the most suitable value can be the tensile strength at yield.

Table 2.1: Definition of variables

Hooke's Law	$\sigma = E \times \epsilon$
Stress	$\sigma = \frac{Force}{A_{Initial}}$
Strain	$\epsilon_{ij} = (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$
Young's Modulus	E
Poisson's Ratio	$\vartheta = \frac{d\epsilon_{transversal}}{d\epsilon_{axial}}$

Another important property is the Poisson's ratio, and it can be defined as the ratio between the strain normal to the applied load, and the strain in the direction of the applied load.

Before the production of any structural element it is required to do a finite element analysis to simulate how the elements would react to the loads applied. For this several computer softwares are available to simulate such scenarios. The results can be shown in the form of Von Mises stress. This data and format allows one to know if the designed structure will withstand the applied loads. The von Mises criteria states that the material will start yielding, or entering in a plastic behaviour, when the von Mises stress is greater then the uni-axial yielding stress. This criteria allows the use of the data from a uni-axial test to be used in a multi-axial scenario and is extremely crucial as the majority of loads applied won't be in a single axis but rather as a combination of several loads. The von Mises value can be obtained from equation 2.1.

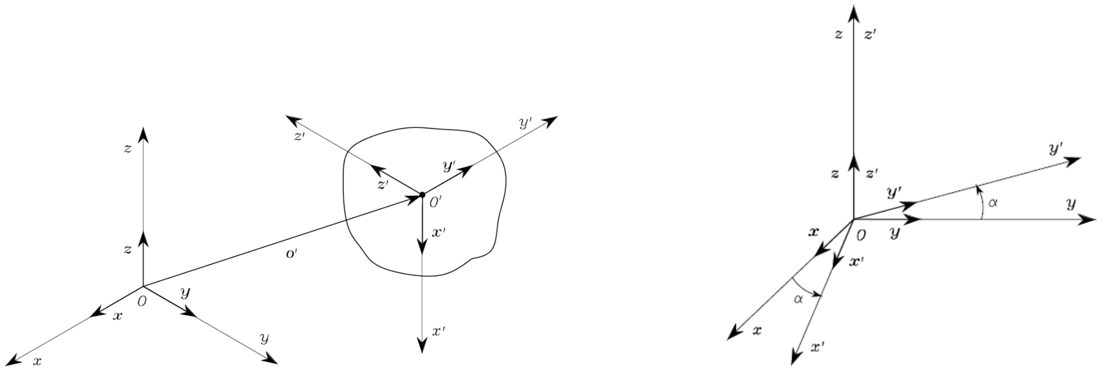
$$\sigma_{vm} = \sqrt{\frac{(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 + 6(\sigma_{12}^2 + \sigma_{23}^2 + \sigma_{13}^2)}{2}} \tag{2.1}$$

2.1.4 Kinematics

The functional use of robots is dependent of an appropriate modeling of their kinematic behaviour. For that it was used the forward and inverse kinematics. The aim of the first is to calculate the position and orientation of the manipulator end-effector, by defining the values of its joints. While for the second the goal is to obtain the required joints values that result in the desired end-effector position.

The end-effector pose can be expressed in terms of a vector and the orientation, that can be expressed as a rotation matrix with respect to the reference frame. The approach presented in this section follows closelly the one in Siciliano, Sciavico, Villani, and Oriolo [28].

Figure 2.3a), illustrates the position and orientation of a point in space, *O'* and *O'x'y'z'*. The position vector can be easily obtained from the point coordinates with respect to the base frame, *O-xyz*, as suggested in equation (2.2). The orientation vector, obtained from the models, can be transformed into a rotated frame if it is considered as one of the frame axis. This rotated frame, *O'-x'y'z'*, can also be defined with respect to the base frame as its axis can be described as vectors in the base frame, system of equations (2.3).



(a) Position and Orientation

(b) Rotation by α along the z axis

Figure 2.3: Position and Orientation; Rotation along the z axis - [28]

$$\vec{o'} = o'_x\vec{x} + o'_y\vec{y} + o'_z\vec{z} \Rightarrow \vec{o'} = \begin{bmatrix} o'_x \\ o'_y \\ o'_z \end{bmatrix} \tag{2.2}$$

$$\begin{cases} \vec{x'} = x'_x\vec{x} + x'_y\vec{y} + x'_z\vec{z} \\ \vec{y'} = y'_x\vec{x} + y'_y\vec{y} + y'_z\vec{z} \\ \vec{z'} = z'_x\vec{x} + z'_y\vec{y} + z'_z\vec{z} \end{cases} \tag{2.3}$$

The vectors $\vec{x'}$, $\vec{y'}$ and $\vec{z'}$, that represent the orientation of the rotated frame can be arranged into a 3x3 rotation matrix, equation (2.4).

$$R = \begin{bmatrix} | & | & | \\ \vec{x'} & \vec{y'} & \vec{z'} \\ | & | & | \end{bmatrix} = \begin{bmatrix} \vec{x'}_x & \vec{y'}_x & \vec{z'}_x \\ \vec{x'}_y & \vec{y'}_y & \vec{z'}_y \\ \vec{x'}_z & \vec{y'}_z & \vec{z'}_z \end{bmatrix} \tag{2.4}$$

The columns of this matrix are the coordinates of the three axes that compose the new frame in the base frame. On the other hand, the rows of this matrix represent the direction of the base frame, *O-xyz*, with respect to the frame*O'-x'y'z'*. The rotation matrix is an orthogonal matrix as a consequence of the characteristics of the vectors that compose it, this leads to the following relation, $R^TR = I_3 \Rightarrow R^T = R^{-1}$.

The rotation matrix of can be a consequence of several successive rotations due to, for example, several sequential joints. This rotation matrix hence can be obtained and described as a sequence of elementary rotation about a single axis using the composition rule of rotation matrices. Figure 2.3b), demonstrates an elementary rotation along the z axis with a resultant rotation matrix in (2.5). If a rotation by α is applied to the frame, the new z axis, z', will be collinear to the original. Each of the new axis, x' and y' will have components in the two original x and y axis. The remaining rotation matrices along the remaining axis can be found in [28] and follow the same reasoning.

$$R_z = \begin{bmatrix} cos(\alpha) & -sen(\alpha) & 0 \\ sen(\alpha) & cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.5}$$

The rotation matrix also represents the coordinate transformation of a point between two frames with the same origin, resulting in the equation $p = Rp' \leftrightarrow p' = R^T p$. Where P represents the coordinates of the point in the base frame and, P' represents the same point but with respect to the rotated frame. An alternative meaning of the rotation matrix is the rotation of a vector with respect to the base frame by a given angle about an axis.

If one numerates the frames instead of using $'$, the previous equations will result in $p_0 = R_1^0 p^1 \leftrightarrow p_1 = R_0^1 p^0$. p^1 and p^0 are the point vector with respect to the frame 1 and frame 0 respectively. While the rotation matrix, R_1^0, represents the rotation of the frame 1 in relation with the frame 0, in other words, it is the rotation necessary to apply to the base frame, $O - x_0 y_0 z_0$, in order to obtain the second frame, $O - x_1 y_1 z_1$.

The composition rule, $R_2^0 = R_1^0 R_2^1$, states that to obtain a rotation from the frame 0 to the frame 2 one needs to apply successive elementary rotations about the previous frame. For this example one starts with the rotation from 0 to 1 and then a rotation from 1 to 2. The rotation matrices exposed are with respect to the last frame obtained, or current frame, however it is possible to represent the previous rotations always with respect to the base frame. To obtain the rotation matrix of such case one should compute the rotation along the initial axis before the remaining rotations. For example, if from the two rotations that lead to the matrix R_2^0 one wanted to rotate along one of the initial axis with a rotation of $\overline{R}_3^2$ the resultant rotation matrix should be accomplished by $\overline{R}_3^0 = \overline{R}_3^2 R_1^0 R_2^1$, [28].

Up to now all the positions and orientations had the same origin, however, it is not always the case. When the position is obtained in a different frame, p^1 in $O - x_1 y_1 z_1$, it can be transformed into the coordinates of the base frame p^0, equation (2.6) from figure 2.4.

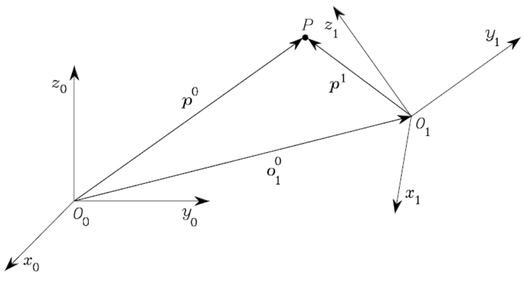


Figure 2.4: "Representation of a point P in different coordinate frames" - [28]

$$p^0 = o_1^0 + R_1^0 p^1 \tag{2.6}$$

This transformation is very useful to compute the direct kinematics of a manipulator composed as a series of rigid links and is helpful for the interpretation of results

2.1.5 Constant Curvature Kinematics

The constant curvature kinematics is a theoretical model that assumes the deformation of the manipulator to be a constant curvature. This constant curvature, in its turn, is defined by a smaller set of simpler parameters and equations. Figure 2.5 exemplifies the approximation of a manipulator with three actuators to a constant curvature, [8], [29].

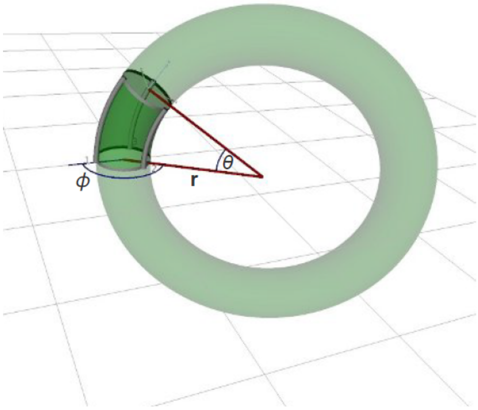


Figure 2.5: Torus Segment as a constant curvature continuum robot - [29]

This theory, in its present formulation, is not valid for scenarios without bending, in other words, scenarios where the manipulator has only a elongation or compression along its main axis. This situation represents a singularity. To overcome such hurdle, Rolf and Steil [29] provides a small modification to this theory in order for it to be able to compute the pure elongation and compression scenarios. A second setback is the inability for this theory to model the passive compliance present on such type of under actuated manipulators. Most continuous robot modules, to perform a spatial motion, rely only on three actuators, a number far inferior to the number of DOF present. This lack of actuation may lead to a considerable deviation from the desired position, [7]. Contrary to a rigid robot that is a fully defined system, [8].

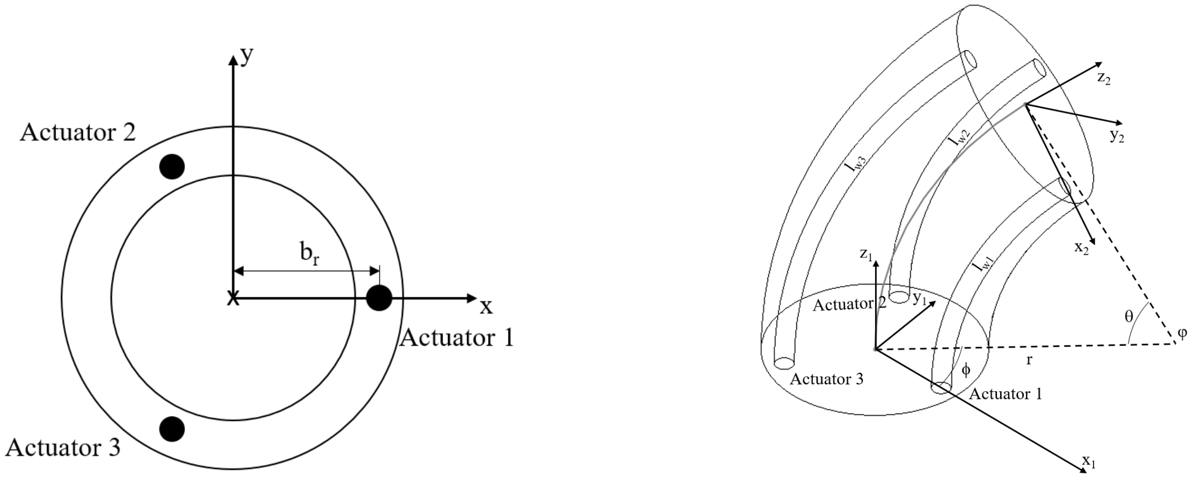
The improved constant curvature theory, from Rolf and Steil [29], shows a stable result around the singularity of the previous model while maintaining it simple and parameterless with improved and reliable results.

The constant curvature model, that will be presented next, has constant radius along its height and it was based on the work of Webster and Jones [8], Boccolato, Manta, Dumitru, and Cojocaru [12] and Rolf and Steil [29]. The forward kinematics was based on [29], while [12] and [8] were used for the inverse kinematics. In [30] and [31] the constant curvature assumption is applied for a truncated cone, with a varying radius along its height.

Forward Kinematics

From Rolf and Steil [29] we get the forward kinematics where, with the known length of the actuators on the outer sides of the module, figure 2.6, it is possible to calculate the end position and orientation of the top frame, 2.6 b). In this paper however, the axis orientation is rotated by 180º from the defined axis in figure 2.6a) along the z axis, while maintaining the same actuators numeration and ϕ. Equations (2.7f) and (2.8) result from the manipulation of the article equations in order to get the desired frame. The coordinates of the end-effector origin can be obtained by changing the signal of its x and y values, the equivalent of making a rotation along the z axis by π. And the elements of the matrix can be obtained as the multiplication of the original rotation matrix by a rotation along the new z axis by π ($R = R_{CC}R_{z(\pi)}$). Then it also necessary to change the values of the x and y components from the three vectors that compose the resultant frame, the equivalent of a rotation along the original z axis, $\overline{R}_{z(\pi)}R$.

The input variables of this model are: l_{w1} - length wire 1; l_{w2} - length wire 2; l_{w3} - length wire 3; and b_r - the actuation radius of the cables from the module axis, figure 2.6. The resultant variables are: r - the radius of the curvature made by the neutral line; the angles θ ans ϕ; and the coordinates of the origin of the top face. All variables can be seen in figure 2.6b).



(a) Axis orientation relative to the actuator

(b) Axis and angles orientation

Figure 2.6: Axis and angles orientation definition

The resultant equations and rotation matrix used for the constant curvature are therefore:

$$\bar{l} = \frac{l_{w1} + l_{w2} + l_{w3}}{3} \tag{2.7a}$$

$$g = \sqrt{l_{w1}^2 + l_{w2}^2 + l_{w3}^2 - (l_{w1}l_{w2} + l_{w1}l_{w3} + l_{w2}l_{w3})} \tag{2.7b}$$

$$\theta = \frac{2g}{3b_r} \tag{2.7c}$$

$$r = \frac{\bar{l}}{\theta} \tag{2.7d}$$

$$\phi = atan2\left(\sqrt{3}(l_{w3} - l_{w2}), l_{w2} + l_{w3} - 2l_{w1}\right) \tag{2.7e}$$

$$d_1^0 = \begin{cases} x = -\bar{l} * sinX\left(\dfrac{\theta}{2}\right) * cos\left(\dfrac{\pi-\theta}{2}\right) * sin\left(\phi - \dfrac{\pi}{2}\right) \\ y = \bar{l} * sinX\left(\dfrac{\theta}{2}\right) * cos\left(\dfrac{\pi-\theta}{2}\right) * cos\left(\phi - \dfrac{\pi}{2}\right) \\ z = \bar{l} * sinX\left(\dfrac{\theta}{2}\right) * sin\left(\dfrac{\pi-\theta}{2}\right) \end{cases} \tag{2.7f}$$

$$sinX = \begin{cases} 1 \quad , if\ sin(\alpha) = 0 \wedge |\alpha| < \pi \\ \dfrac{sin(\alpha)}{\alpha},\ else \end{cases} \tag{2.7g}$$

$$R_{CC} = \begin{bmatrix} cos(\phi)cos(\theta)cos(\phi) + sin(\phi)sin(\phi) & cos(\phi)cos(\theta)sin(\phi) - sin(\phi)cos(\phi) & cos(\phi)sin(\theta) \\ sin(\phi)cos(\theta)cos(\phi) - cos(\phi)sin(\phi) & sin(\phi)cos(\theta)sin(\phi) + cos(\phi)cos(\phi) & sin(\phi)sin(\theta) \\ -sin(\theta)cos(\phi) & -sin(\theta)sin(\phi) & cos(\theta) \end{bmatrix} \tag{2.8}$$

In the equations (2.7f) it is possible to observe a variant of the sin function, sinX. This modification of the sin function enables the model to evaluate pure elongation or compression scenarios. Previously, such case would result in θ equal to 0, r equal to infinite and an undefined ϕ. The proof for this formulation can be examined on [29].

To achieve the offset on the forward kinematics one can apply the equation (2.6) with the rotation matrix in (2.8) and o_1^0 as the vector obtained in equations (2.7f). This prototype only has an offset, p^1, along the z axis from the top face of the flexible manipulator, to the top face of the top element.

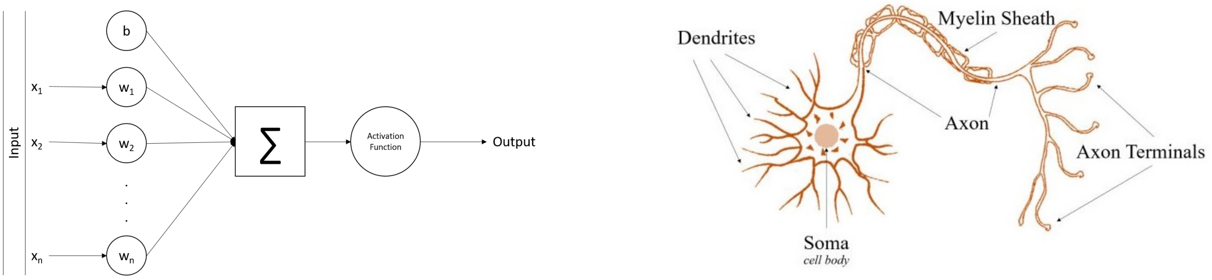
2.1.6 Artificial Neural Networks

For some tasks and some robots obtaining the theoretical model is not feasible or precise due to its complexity and non-linearities. For these cases, it is possible to use a model that learns the targeted behaviour by example using artificial intelligence. Artificial Intelligence, is defined as a computer or machine that is able to mimic natural intelligence, from humans or other living forms, and perform tasks using this replicated intelligence.

Machine learning (ML) is the first step to obtain an artificial Intelligence model. ML is said to be performed by an algorithm that is able to, without being explicitly programmed, learn from experience and make predictions. This method is especially useful when the goal is to replicate a complex system subjected to non-linearities that are not easily classified. These algorithms are able to learn with three different methods: Supervised, Unsupervised and Semi-supervised learning, [32], trained respectively with: labeled, unlabeled and both labeled and unlabeled data. The type of algorithm is dependent of the type of data used to train the model. The difference between labeled and unlabeled data is that on the first, the algorithm knows a priori its desired output; hence, supervised. With unlabeled data the algorithm will attempt to infer the properties from the data set.

Artificial Neural Networks (ANN) are mathematical structures that use ML algorithms to adapt to the training data, its simplest form is illustrated in figure 2.7 a), as a perceptron with a single layer and output.

In this figure we have the input layer, with the different values of x, a bias value, b figure 2.7a), a node where the sum is made and finally, an activation function that will give an output depending on its sum. These models of ANN are inspired by the human brain. In the same way that the dendrites receive the input signals through synapses from other neurons, the soma processes the information and, converts into an output that is sent by the axon terminals to the connected neurons; the perceptron follows suit with an approximated artificial model. The similarities are evident on figure 2.7.



(a) Perceptron - Single Layer Neural Network

(b) Schematic of a Neuron - [33]

Figure 2.7: Brain neuron and artificial neuron compared

Several activation functions, that determine the output value, can be used. They can range from linear, for example a binary or linear functions, to non-linear functions, like sigmoid and hyperbolic tangent. For a more complex set of data, that requires a creation of a more complex model, the choice must fall on non-linear functions and it should be tested which of the activation functions works best, [34]. The modification of the activation function won't be of a great difference if the architecture of the neural network is a single perceptron. In this case even if the activation function is non-linear the neural network would only be able to classify linearly separable patterns so to have a more complex model it is necessary to use a multi-layer neural network.

The last step to obtain a model using NN is its training method. Similarly to the increasing strength of the biologically neuron synapses every time they are fired, the connection weights of the ANN also change with training. The training goal is to minimize the model error with the provided data set. The training algorithms can use a first-order gradient, like error backpropagation, or a second-order grandient, like Newton's Methods or Levenberg-Marquardt. Generally the latter methods are more efficient with a greater convergence rate than the former, [35].

Chapter 3

Conceptual Development

To develop a soft continuum robot, and any product, one should go through several steps. First, a research should be done about the state of the art, in the previous and current chapter. Then, requirements and the use scenarios should be set in order to define the project goals. In this chapter such requirements will be defined as well as the design obtained. Next, a finite element analysis will be made and the manufacturing and assembling methods will be presented.

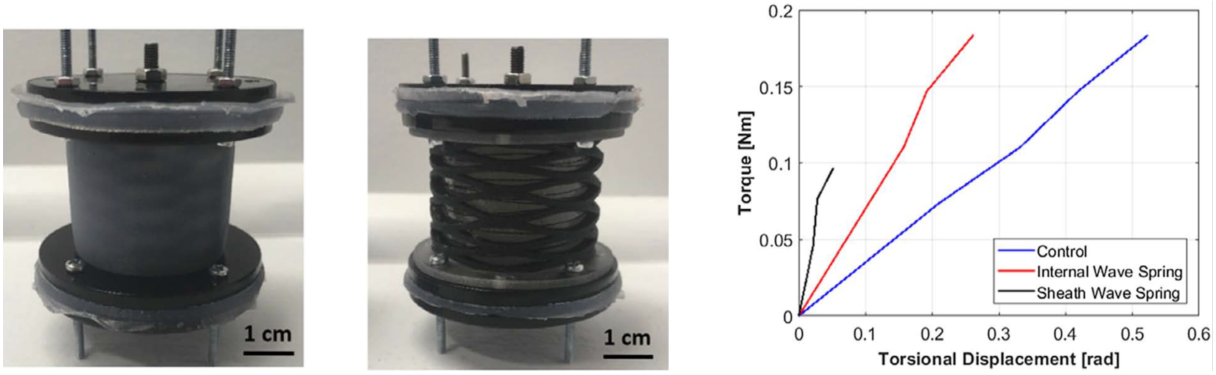
3.1 Design Parameters

The central point of the present work is to obtain a the soft manipulator capable to sustain torsion for an off-axis manipulation. To accomplish this higher torsional stiffness, the geometry of the manipulator should be able to behave as a structure while also being able to extend, compress and bend as a continuum manipulator. Considering these factors the geometry chosen was a wave-spring shape, figure 1.7. In [18] it is made an analysis of the behaviour of such geometry and its evolution when several parameters are modified. The module developed was a result of a pneumatic actuator reinforced with a wave spring. Figure 3.1 shows the advantages of using an external wave spring comparatively to an internal wave spring and to a control actuator without the wave spring, only with a silicone actuator. The graphic of figure 3.1 c) shows a significant increment on its stiffness.

This study found that the configuration for an amplitude of 1.905 mm has the highest torsional stiffness for a inner diameter with 30 mm, a beam width of 2.5 mm and that this stiffness increases with the number of waves (with a constant perimeter). The material used to print was NinjaFlex^{TM}.

The work of [18] could be closely followed, however this 3D printed manipulator is the result of combining a pneumatic actuator with the external wave-spring.

For the present work we have as project goals, the requirement for the prototype to have a spatial manipulation and to have a lightweight structure. The use of wires as the actuation method allows one to have a spatial manipulation by shortening and extending its length while not undermining the malleable characteristics of the material as it follows the manipulator curvature. An example of a cable driven with torsional stiffness robot can be found for instance in [15]. An important point to remember is the need



(a) Internal Wave Spring

(b) External (Sheath) Wave Spring

(c) Torsion test results

Figure 3.1: Analysis from [18]

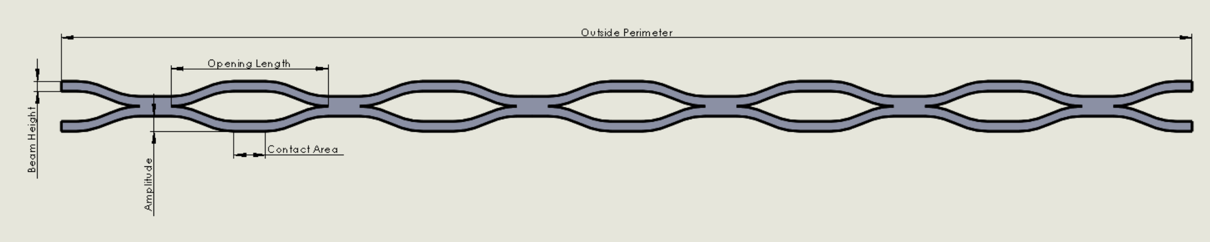


Figure 3.2: Side view measures of the joint

for cable guides to prevent it from buckling and closely follow the shape of the joint.

Another added value for the use of a cable driven actuator lies in the fact that they are more easily controlled and apply larger forces in specific scenarios, like in a traction situation, when compared to the pneumatic actuators, which are more commonly used. Although the force applied and the precision obtained are not as great as for the hard robots, the usage of soft materials allows the manipulator to have a greater softness and adaptability to the surroundings, [13].

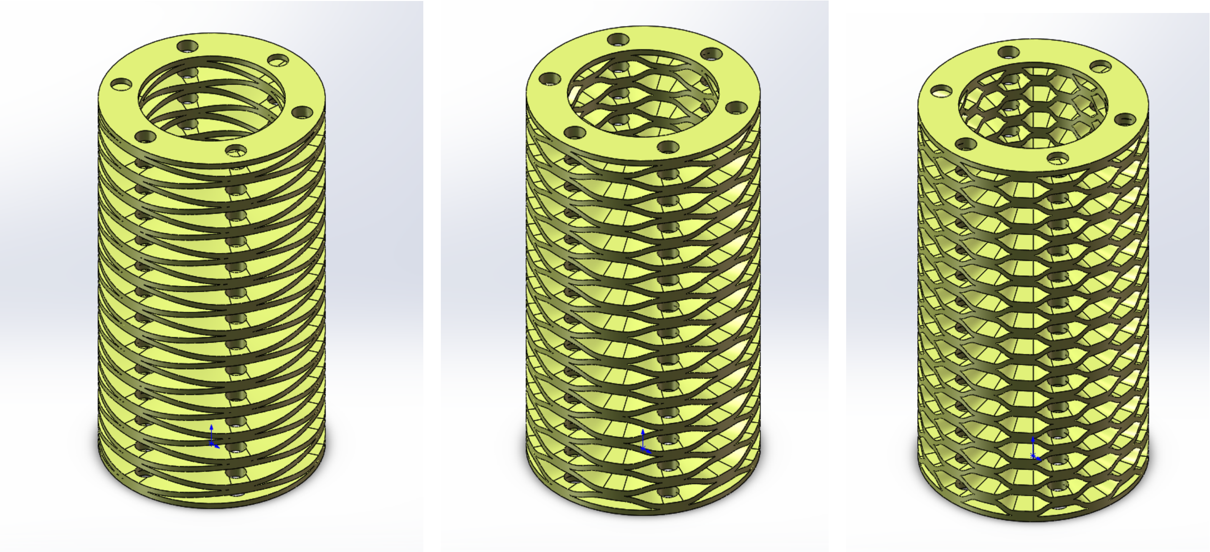
To obtain a lightweight prototype and to take fully advantage of the spring geometry, the manipulator was defined to have a hollow center. This feature also allows the passage of cables and other objects inside the structure. Such characteristic allows an increment on safety as it hides the otherwise suspended cables and maintains the properties of the soft robot that prevents large contact forces. The empty center can also be a route for external bodies to safely flow through.

Lastly, to obtain a good manipulation dexterity the manipulator must bend at least 90°.

If the previous requirements are met, they will result in a hyperredundant continuum extrinsic soft robot.

3.2 Finite Element Study

The majority of the parameters were defined either from functional constraints or from [18], these can be found on the table of figure 3.10. However, it is still required to define the number of waves per level. In this section, it will be analysed the simulated response of the manipulator to external forces, the impact that the number of waves per level have in its longitudinal, torsional and bending stiffness with a static and dynamic analysis, and it will be selected the number of waves that have the best results. For these analyzes it was used three different models where the only parameters modified were the number of waves per level and the wave length. The number of waves should be a multiple of the number of actuators to maintain symmetry, these models are shown in figure 3.3 with 3, 6 and 9 waves per level respectively. With the evolution of stiffness according to the number of waves and with the experience of handling the 3D printed tests, it is possible to choose the most appropriate and suitable model. This analysis was conducted using the software NX^{TM} from Siemens^{TM}.



(a) Manipulator with 3 waves per level (b) Manipulator with 6 waves per level (c) Manipulator with 9 waves per level

Figure 3.3: Models used for simulation

Data for simulation

To run a finite element analysis it is necessary to first characterize the properties of the materials used, *PLA* and *FilaFlex*. The only element made of the flexible filament, *FilaFlex*, was the manipulator module of figure 3.3. The remaining elements are made of *PLA*.

PLA has the following properties, table 3.1:

Table 3.1: PLA Properties. * Injection Moulded test pieces; ** Printed pieces and stretched parallel to layers; *** Printed pieces and stretched perpendicular to layers

Young's Modulus	1320 MPa*/1230 MPa**/1120 MPa***
Poisson's Ratio	0.33
Mass Density	1240 kg/m^{3}
Tensile Strength at break	52 MPa*/50 MPa**/39 MPa***

From PLA's datasheet, [36], it is possible to withdraw all the previously data, and others, with the exception of its Poisson's coefficient. For the simulations it was used the lowest value of tensile modulus to simulate the scenario where the forces are applied perpendicular to the layers. The Poisson's coefficient is obtained from Ferreira, Amatte, Dutra, and Burger [22]. In this article it is attempted to characterize the mechanical properties of two 3D printed materials in different directions. The tested materials were two versions of *PLA*, a basic *PLA* and a reinforced with short carbon fibres. In the work of [22] the 3D printer and material used are from the same manufacturer as the materials used here, in the present thesis. The tensile modulus obtained on table 3 of [22] is significantly higher than the one provided in its datasheet. However, the tensile strength in both directions, parallel and perpendicular to the layers, are very similar. Nonetheless, with the datasheets and the experiments performed it is possible to, reliably, extract the Poisson's coefficient from [22] and obtain a global coefficient as the mean of the average values in both directions, table 3.2 is copied from the paper's results. The directions 1 and 2 are in the printing plane and the axis 1 is aligned with the direction of the filament while axis 2 is perpendicular.

Table 3.2: PLA tests results from [22]

Property	Direction	PLA		
		Max	Avg	Dev
Poisson	ν_{12}	0.349	0.331	0.011
Coefficient	ν_{21}	0.336	0.325	0.014

Regarding the properties of *FilaFlex*, the information seen in table 3.3, with the exception of the Poisson's coefficient, is taken from the datasheet, [37]. This flexible material is a thermoplastic elastomer (TPE) also known as a thermoplastic rubber. The elastomeric properties of this filament are a result of blending two separate materials, a polymer as a base, plastic, and rubber as an alloy, [38]. Without any additional information from the datasheet the Poisson's coefficient was then assumed to be similar to rubber. Since *Filaflex* is a material that has a large elongation to break value, 700%, and the elongation and compression of the manipulator is mainly due to its geometry and not its material, it is possible to state that the strain developed will be very low and that Hooke's law will provide a good approximation for the present analysis. The Young's modulus was considered to be equal to the supplied tensile storage modulus.

Table 3.3: Filaflex Properties

Tensile Storage Modulus (20ºC)	48 MPa
Poisson's Ratio	0.49
Mass Density	1200 kg/m^3
Ultimate Tensile Strength	54 MPa

3.2.1 Results from simulation

The first results presented here refer to the static simulation of the three different models, figure 3.3, with the same boundary conditions and loads. In each test the base was fixed and on the top section of the module it was applied four types of loads: extension and compression forces, axial torque, and a transverse force. The calculated stiffness can be examined in the graphs of figure 3.6. After the analysis of the results obtained it was performed a modal analysis to a chosen geometry from the static loading.

The value for the extension and compression force is obtained from the maximum torque that the three servo motors (Tower Pro^{TM} MG996R) are capable to deliver. Considering that the motors will work at the voltage that produces the highest value of torque, 11kgcm at 6.0V with a radius of 22.5 mm, equation 3.3a, the applied force will be $4.88kgf = 47.85N \approx 50N$.

To calculate the extension and compression stiffness it was used the maximum displacement of the module, the applied force and the Hooke's law. For the torsional stiffness it was necessary to calculate the rotation angle suffered by the top section, for that it was used the displacement of the inner and outer radius. The displacement obtained is the magnitude movement and not the path followed and so it was necessary to estimate the angle. For the last simulation, bending scenario, the force was applied on the x direction hence the top section movement will be on the plane xz. With the displacement of x and z it is also possible to calculated the bending stiffness.

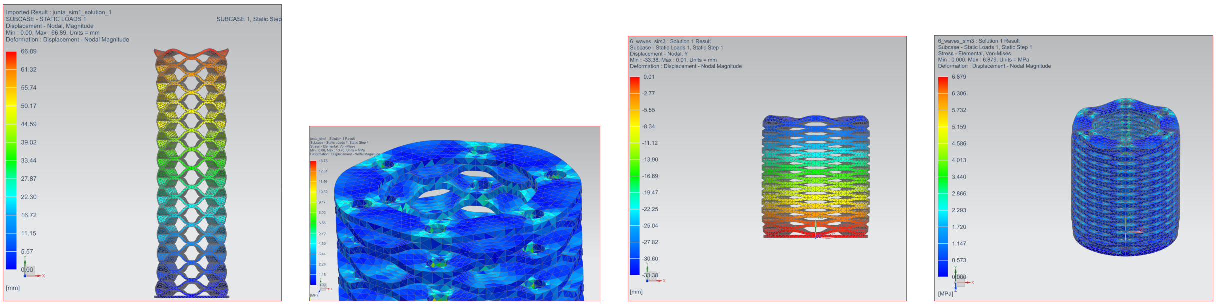
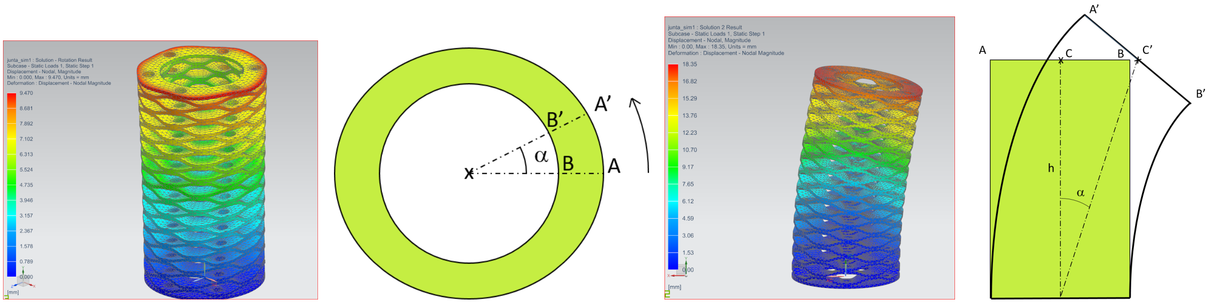

(a) Extension - Displacement 1:1 (b) Extension - Stress (c) Compression - Displacement 1:1 (d) Compression - Stress

Figure 3.4: Simulation - Extension and Compression



(a) Torsion - Displacement 1:1 (b) Torsion - Displacement interpretation (c) Bending - Displacement 1:1 (d) Bending - Displacement interpretation

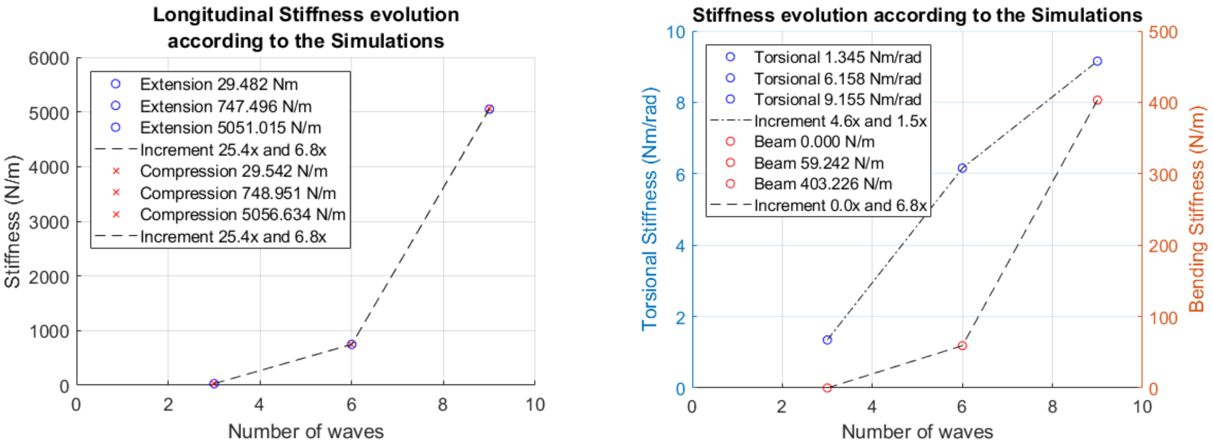
Figure 3.5: Simulation - Torsion and Bending

Using both displacement magnitudes of the points A and B, $\overline{AA'}$ and $\overline{BB'}$, and A', B' and C' presented in the table 3.4, for both torsion and bending simulations, figure 3.5, one is able to compute the degrees of rotation. The torque was applied to the outer diameter thus creating a small angle difference between the inner and the outer diameter, 0.8 deg. The results obtained for the four simulations of the three models can be analysed in figure 3.6. In these graphs it is possible to observe a growing tendency of the torsional stiffness as the number of waves increases, similarly to [18], although in this reference it is

Table 3.4: Displacement from the torsional and bending simulations for a model with a height, a inner and a outer diameter of 84.5 mm, 29.5 mm and 45.5 mm respectively

Torsional Simulation		Bending Simulation	
$\overline{AA'}$	9.33 mm	A’	(16.88 mm; 7.19 mm)
$\overline{BB'}$	5.85 mm	B’	(16.88 mm; -7.18 mm)
Outer Diameter (D_1)	45.5 mm	C’	(16.88 mm; 0 mm)
Inner Diameter (D_2)	29.5 mm	-	-

only tested levels with 3 to 5 waves.



(a) Longitudinal Stiffness

(b) Torsional and Bending like stiffness

Figure 3.6: Evolution of the Stiffness with respect to the number of waves per level

Table 3.5: Various Results of the joint rigidity

Model vs rigidity	Extension	Compression	Torsion	Bending
3 Waves	29.482 N/m	29.542 N/m	1.345 Nm/rad	-
6 Waves	747.496 N/m	748.951 N/m	6.158 Nm/rad	59.242 N/m
9 Waves	5051.015 N/m	5056.634 N/m	9.155 Nm/rad	403.226 N/m

As seen in the figure 3.6 it is possible to see the evolution of the various stiffness according to the number of waves. In the graphics both the compression and extension curves are very similar, this was expected as the force applied is on the same direction, with symmetric values, for both simulations. It is interesting that the bending simulation follows the same pattern as the previously mentioned curves. In this curve the increment from the 6 wave model to the 9 wave model is also equivalent to the extension/compression values, 6.8 times. This similarity can be due to the fact that in this simulation we continue to have only extension and compression behaviour on the model, with one side being compressed while the other is in extension. The torsional stiffness, on the other hand, follow a more linear curve.

From these results it is possible to state that the model with 6 waves per level maximizes its torsional stiffness in relation to the extension/compression and bending stiffness. This model has a torsional stiffness only 1.5 times inferior to the model with 9 waves, while the remaining stiffness are inferior 6.8

times. From the model with 3 waves it is possible to foresee a low performance as it won't have a stiffness able to support external loads by its own without the help of the actuators. On the opposite spectrum, the model with 9 waves has a very high stiffness for either compression or extension. Considering that the model with 6 waves maximizes the torsional stiffness with relation to the remaining stiffness and that the 3 and 9 waves' models wouldn't be as functional for its main purpose, the choice of the model with 6 waves to be used in the continuum soft robot is rather clear.

Two possible factors that may contribute to these evolutions are respective the added volume of material and the maximum angle made by the wave with the horizontal plane. Regarding the second factor, as the number of waves increases its periodicity will decrease hence the angle of the wave with the horizontal plane will increase. Due to this geometry a model with less waves will have a lower extension/compression stiffness, as its angle will be lower and its direction will be almost perpendicular to the direction of the applied force. Conversely, a model with a greater number of waves will have a more vertical wave and so it is able to better resist the applied force on the same direction. On the other hand, when we have a torsional force, the force direction will be perpendicular to the wave and so resulting only in a mild increment of torsional stiffness.

Table 3.6: Maximum von Mises stress for each simulation - * Not comparable with the remaining values

Model vs Max von-Mises stress	Extension	Compression	Torsion	Bending
3 Waves	24.13 MPa*	2.41 MPa*	8.93 MPa*	1.02 MPa*
6 Waves	13.76 MPa	6.87 MPa	11.11 MPa	1.40 MPa
9 Waves	4.14 MPa	1.56 MPa	5.65 MPa	0.41 MPa

None of the maximum von Mises stress reach a troublesome value, as the ultimate tensile strength for Filaflex^{TM} is 54 MPa.

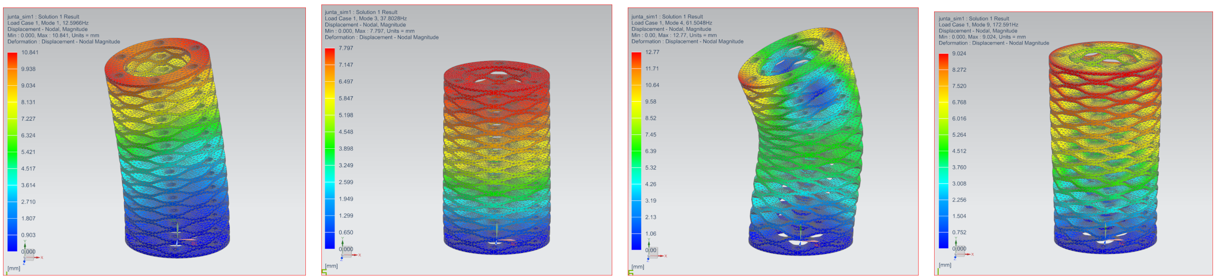
The values of stress for the model with 3 waves are not comparable with the remaining models since it was necessary to apply a lower force to the first model.

3.2.2 Dynamic Analysis

A modal analysis was performed to the flexible module with 6 waves per level. Figure 3.7 illustrates the first vibration modes for bending, compression and torsion, and table 3.7 presents the corresponding frequency and deformation type.

The examination of the natural frequencies show that the torsional stiffness is higher when compared to the remaining modes, as the natural frequency of the torsional movement is higher than for the remaining. The results so far accomplished are in agreement with the research question since they provide a geometry with an increased torsional stiffness for a soft robot. However, the question is not completely answered, it is still needed to verify experimentally that, first and foremost, this geometry is capable to behave like a soft robot, then and lastly to verify if the properties are maintained.

With all the necessary features defined, one could start modeling the actuator. However, the design of the model is also dependent of the properties of the material to be used. The material used was a soft



(a) Modes 1,2 - 1st Bending - 12 Hz (b) Mode 3 - Longitudinal - 37 Hz (c) Modes 4,5 - 2nd Bending - 61 Hz (d) Mode 9 - Torsion - 172 Hz

Figure 3.7: Mode shapes

Table 3.7: Modal Modes and Frequencies

Mode	Type	Frequency
Modes 1,2	1st Bending Mode	12 Hz
Mode 3	Extension	37 Hz
Modes 4,5	2nd Bending Mode	61 Hz
...	...	...
Mode 9	Torsion	172 Hz

filament called *Filaflex*, table 3.3. For the cables, it was used stranded steel wires with a diameter of 2 millimeters which fit very well not only in terms of measures but also in terms of physical characteristics. These allow to apply forces, to extend or compress, without buckling. The manipulator will have support points close enough to prevent any buckling. For last, the material used to build rigid parts of the mechanism was *PLA*, a 3D rigid filament, which is commonly used in 3D printing, table 3.1.

3.3 Evolution of the 3D Model of the Joint

As the geometry parameters were modified, due to functional constraints, several printing test were conducted in order to guaranty that the 3D printer is capable to print the final model and to refine the 3D printer parameters. With the printing tests it was also possible to compare the printed module with simulation results. Figure 3.8 and table 3.8 depict the evolution of the models, from left to right and top to bottom.

Table 3.8: Evolution of the parameters of the Wave Spring in mm

Model	Inner Diameter	Outer Diameter	Amplitude	Number of Waves	Beam Height	Beam Width	Material
2	30	40	1.9	6	1.3	5	PLA
5.2	30	45	1.9	6	1.3	7.5	FilaFlex
5.3 V2	29.5	45.5	1.9	6	1.3	8	FilaFlex

The starting point to print was the model 1 with the parameters from [18] in a rigid filament in order to grasp the scale and measures of the model. The goal is to have a structure that is able to support its

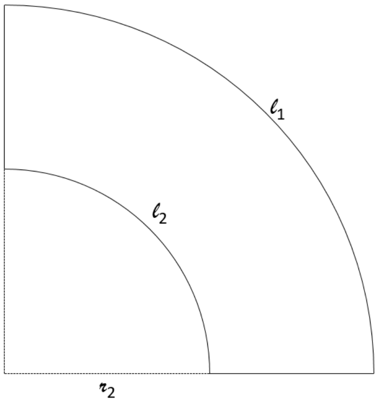
Figure 3.8: Evolution of the 3D model with the different parameters. The parameters can be seen in the table 3.8

own weight and encapsulate the steel wires along its height, and not only as an external support against torsion as in the article. In light of this, the number of waves was increased to six. The numbers of waves should be a multiple of the number of actuators to maintain symmetry. Since the minimum number of actuators to obtain a spatial motion is three, the number of waves per level should, therefore, be a multiple of three. The increment to six waves per level of the manipulator will increase the torsional stiffness and provide an angle closer to 45° resulting in a more balanced ratio between extension/compression and torsional stiffness. The beam width was also increased to allow the wires to pass trough.

Having a satisfactory result in terms of measures with the rigid material, the filament was changed to the flexible one. After obtaining a good printing setup and a good prototype, it was necessary to add holes for the passage of the 3 steel wires plus an extra sets of holes for a possible additional module. Initially, the diameter was 2.5mm but later in the process it was increased to 4.25mm in order to add a nylon ring to each contact surface to reduce the friction between the elastomer and the stranded steel wires. An important note to remember when modeling foldable structures is to make sure that there is enough space for the top and bottom layers to fold. During the printing tests a model with only three waves per level was printed, however it did not show favorable results in terms of supporting its own weight. And accordingly to [18] and with the simulations performed the torsional stiffness would be lower. This can be confirmed when handling the prototype as it showed a more easily deformable module.

With the basic parameters of the wave defined it is necessary to calculate the minimum height of the module in order to fulfill the last requirement, to bend at least 90 deg.

Initially it was estimated a reasonable height for the manipulator to be from 80 to 90 mm. To obtain the height required, one should take into consideration that this height will be a multiple of the wave level height. In figure 3.9 it is possible to follow the reasoning for the height obtained, as a result of the 26 waves, which correspond to a power of compression of 49.53 mm. Equations (3.1a) show the calculation of the model height, the possible compression and the resultant minimal height. In the first equation 1.905 mm is the amplitude and 1.25 mm is the height of each beam, for the 26 layers, the top and bottom added layers make up the 28 layers of 1.25 mm.



$$h = 1.25 \times 28 + 26 \times 1.905 = 84.53mm\,, \tag{3.1a}$$

$$\Delta h_c = 26 \times 1.905 = 49.53mm\,, \tag{3.1b}$$

$$l_2 = h - \Delta h_c = 35mm\,. \tag{3.1c}$$

Figure 3.9: Picture and equations used to calculate the length of compression and extension. h - height; Δh_c - compression; l_2 - minimal height

With figure 3.9 and the minimal height of the manipulator, l_2, it is possible to obtain the necessary maximum height following the set of equations (3.2). Δh_e - extension; Δh_{total} - maximum height variation.

$$2\pi r_2 = 4 \times l_2 \Rightarrow r_2 = 22mm\,, \tag{3.2a}$$

$$2\pi(r_2 + 45,5) = 4 \times l_1 \Rightarrow l_1 = 106mm\,, \tag{3.2b}$$

$$\Delta h_e = 106 - 84.53 = 21.5mm\,, \tag{3.2c}$$

$$\Delta h_{total} = 49.53 + 21.5 = 71mm\,. \tag{3.2d}$$

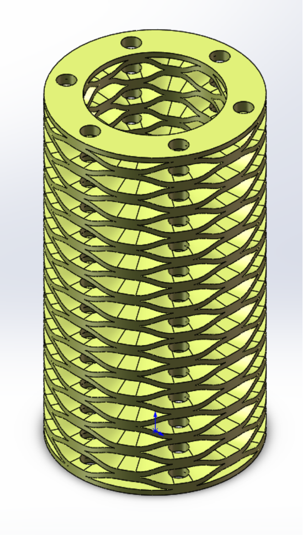
With an extension delta inferior to the compression delta, it is possible to guarantee that the requirement to bend at least 90 deg is met. The obtained extension delta is achievable since it is shorter than the compression delta and after a visual analysis to the model, it is clear that the manipulator is able to extend more than it is able to compress (the wave has a lower angle than 45 deg).

3.3.1 Other 3D Elements

With the structure of the manipulator already defined and modeled, it is then important to model the rest of the elements that complete the soft robot. The most important parts to model are as follow:

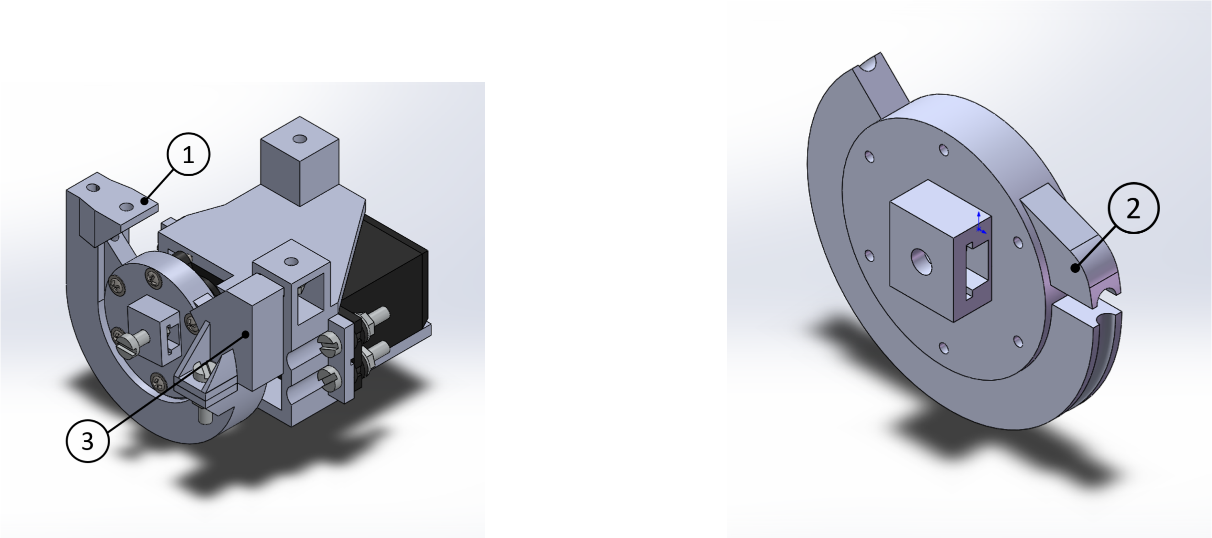
1. The actuators, figure 3.11;
2. The extremity part (top), figure 3.13 a);
3. And the pieces that attach all the parts together, figure 3.13 b),c).

To design the actuators it was necessary to design a system that transforms the rotational motion to a linear movement, for pushing and pulling of the wires. For that, it is required a motor that is able



Final model measures	
Height	84.5 mm
Minimal height	35 mm
Outer Diameter	45.5 mm
Inner Diameter	29.5 mm
Amplitude	1.9 mm
Number of Waves per level	6
Number of levels of waves	26
Number of holes	6
Diameter Holes	4.2 mm

Figure 3.10: Final model parameters



(a) Actuator

(b) Actuator's wheel

Figure 3.11: Actuators

to receive as input the desired rotation for the actuator. The motor used was a servo motor with an amplitude of movement of 180º, servo motor Tower Pro^{TM} MG996R. Having this limitation and knowing the extent of compression and elongation needed for the module, one can compute the necessary radius of the turning wheel r_{act}, equations 3.3. This diameter refers to the diameter of the center of the wire.

$$\frac{1}{2}(2\pi r_{act}) = \Delta h_{total} \Rightarrow r_{act} = 22.6mm\,, \tag{3.3a}$$

$$diam_{act} = 22.6 \times 2 = 45.2mm \Rightarrow 45mm\,. \tag{3.3b}$$

The final version of the actuator can be seen in the figure 3.11. In this figure it is possible to visualize that for taking advantage of the empty core of the manipulator, figure 3.12, the anchor point on figure

3.11a), element 1, should be on the opposite side of the axis. It was also added to the actuators several extensions and supports to prevent the wires from derail. On the figure 3.11 b) we have a support element 2, that when the actuator pushes the wire (rotates clock-wise) it prevents from moving out of its position. On figure 3.11 a) the small added piece 3 prevents the wire to be pinched and jam the actuator.

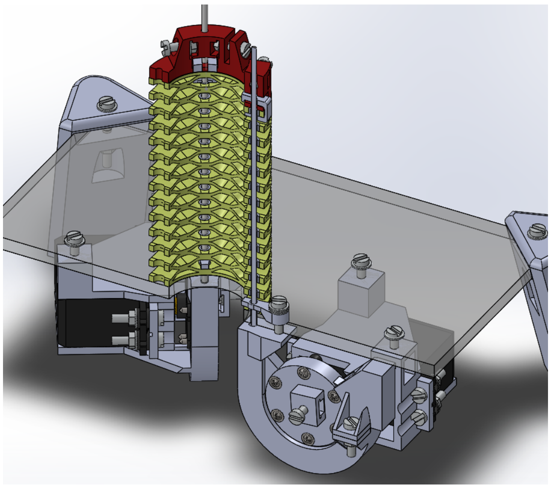
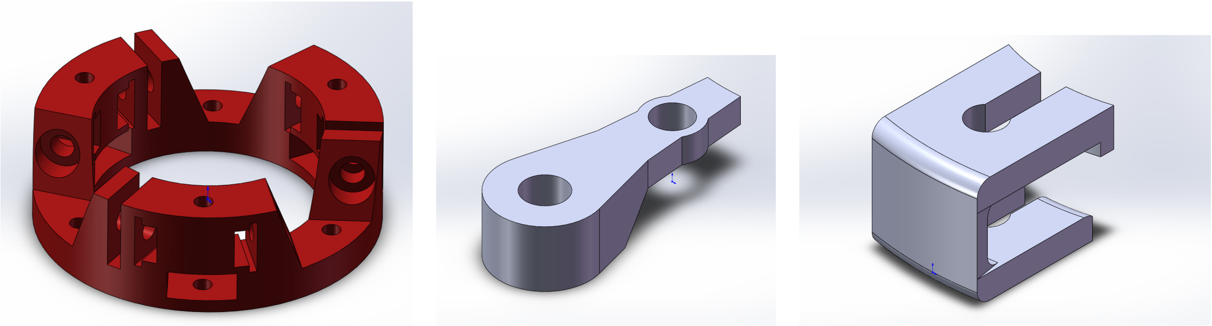


Figure 3.12: Manipulator's mechanics



(a) Top part

(b) Base Holder

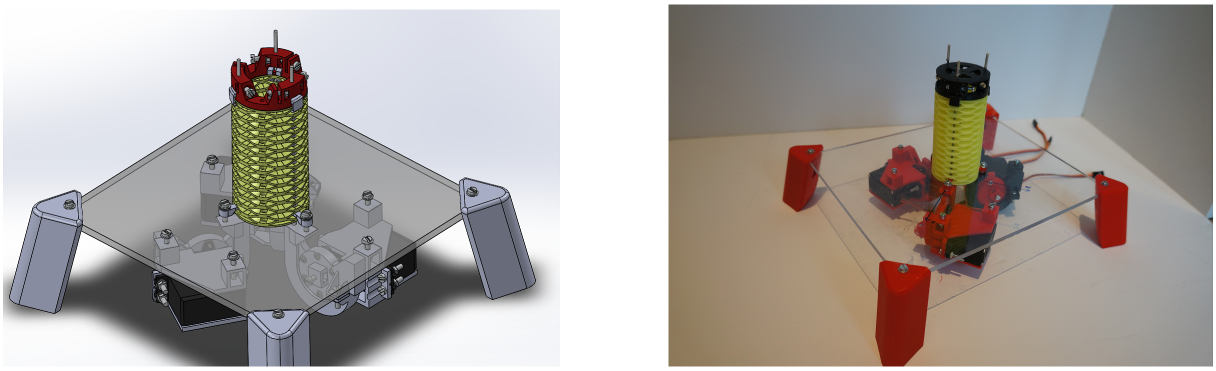
(c) Top Holder

Figure 3.13: Holders to unite the different parts

The top element of the prototype, 3.13 a), is probably the most complex model to be 3D printed, besides the flexible module. In this model it is also implemented the necessary holes to guide the extra wires for the extension/tool of this joint; space for the screws and nuts to lock the position of the wires; and space for the extension or tool to be attached, vertical gaps. The main guideline when modelling this part was to avoid that any element exceeds the boundaries of the top element, defined by its inner and outer radius, while maintaining a compact model with a minimal influence on the flexible module. The assembly of the top part with the top holder on the flexible module can be demonstrated in figures 3.12 and 3.14, and the final functional and totally assembled model can be seen seen in figure 3.14.

3.4 Prototype production and assembling

With all the elements fully defined the prototype is then prepared to be produced. For that it was used an additive technology called 3D printing.

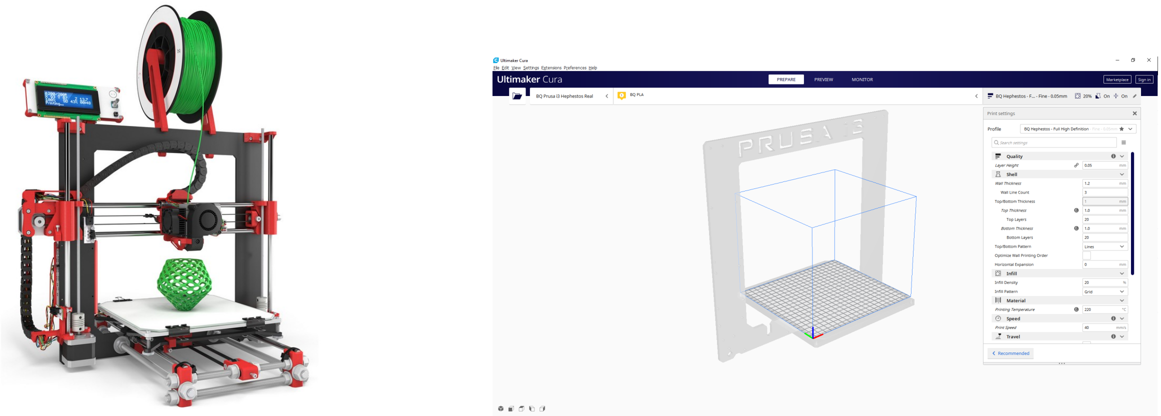


(a) Final Model
(b) Assembled Prototype

Figure 3.14: Final Model and Assembled Prototype

3.4.1 3D Printing

All components where printed using a 3D printer from *BQ*, *Hephestos* figure 3.15 a), and an open-source software from *Ultimaker*, *Ultimaker Cura 4.5* figure 3.15 b). The *PLA* filament, as already mentioned, is from the same producer as the 3D printer, *BQ*, while the flexible filament, FilaFLex^{TM}, is from Recreus Industries S.L.^{TM}.

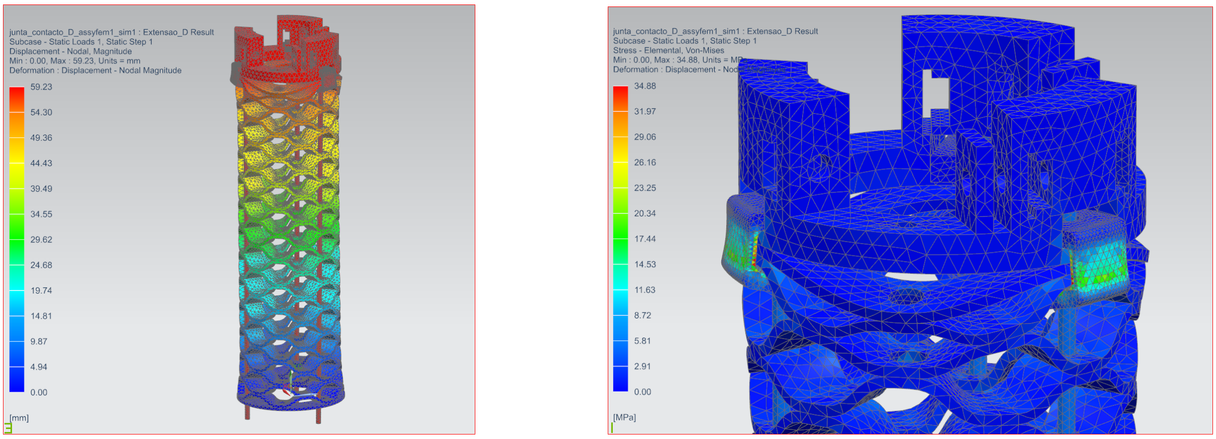


(a) 3D Printer - BQ Hephestos
(b) 3D Printer Software - Ultimaker Cura 4.5

Figure 3.15: 3D Printer and Software used

PLA is a very easily printed material and the printing profiles used were made available by the manufacturer. However, to improve the adhesion of the first layer to the printing bed, the building plate where the material is deposited, it was added a layer of painter tape to the printing bed. Almost no rigid element needed to be printed in a specific orientation except the top holders, figure 3.13 c). This element, due to its geometry and the fact that the applied forces tend to open this part, has a potential fatal flaw as it will concentrate the forces. For that reason a simulation was made, with an extension force, to the flexible module with the top elements of the assembled prototype.

From the resulting figure 3.16, it is possible to notice a stress concentration on the inner face of the holder with a maximum stress of 34.88 MPa. If one checks the material properties of *PLA*, table 3.1, the lowest tensile strength at break is about 39 MPa which is close to the maximum stress obtained. However this value is only for the case where the force is applied perpendicularly to the printing layers.



(a) Simulation - Real Displacement

(b) Simulation - Detailed stress

Figure 3.16: Simulation - Assembled prototype extension

When the force is applied in the direction of the layers this value increases to 50 MPa. In light of this, this element should be printed sideways in order to prevent any mechanical failure due to bad layer adhesion or excessive force.

The remaining elements were printed either in its working position or with the largest face in the building plate and with printing support if deemed necessary.

If the majority of the parts were easily printed, FilaFLex$^{\text{TM}}$ provides a more tricky printing process. For this material due to its nature is more susceptible to buckle when subjected to compression loads and so, some adjustment to the settings of the printing profiles and printer were required.

One of the greatest difficulties while printing a flexible filament is its tendency to buckle after the driving gear and possibly jam the extruder. Some advices can be seen in [39], namely to decrease the gap between the driving gears and the hot end, exchange the nozzle for a bigger one, reduce the printing speed, and cooling the extruder to avoid a precocious softening of the filament. The adjustments made in this work to the settings were: 1) Reducing the printing speed, to allow a greater cooling of the printed layers and prevent the buckling, 2) the addition of an extra fan to cool the layers and the extruder, 3) and the disabling of the features *retraction*, *support material* and *build plate adhesion*. The function to generate support material was not used as the extra soft material would be extremely difficult to remove.

A key aspect when 3D printing without support material are the overhang faces of the model, this aspect should be taken into account in order to minimize the bad printed areas. The overhang issue is a result of the material not being able to support its own weight when in a melted state without any support material. When considering printing rigid materials one is able to generate support that can be easily removed, however when it comes to a flexible filament it is not possible to easily break the fine bond that connects the support material to the model. Its physical characteristics allows it to deform and not break the connection. The lack of support can result in some falling material from its intended position, an exaggerated example can be seen in figure 3.17. Here the lack of support for the last layer results in a deformation due to the lack of support underneath, in contrast with the lower layers that even without support can be printed with good results. When the overhang is not considerable, as one might see on the fourth and fifth layer, there is no observable deformation.

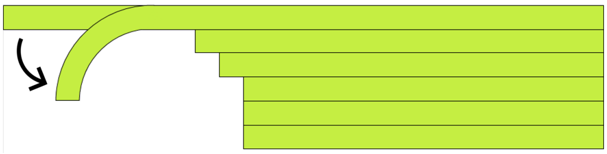


Figure 3.17: Overhang example

To prevent the falling of the overhang layers, for the rigid filaments, one could reduce the layer height hence reducing the overhang area. Such solution was attempted for FilaFlex^{TM}, from 0.2mm to 0.1mm, however it did not produce a better outcome. Another solution to prevent the falling material is by increasing its cooling rate as soon as it is extruded from the nozzle. For that it was added an additional fan, attached to the building plate seen in figure 3.18, to help cooling the overhang areas. This allows a faster cooling which leads to the material to harden faster and not to fall as much.

Table 3.9: FilaFlex Printing Parameters

Layer Height	0.2 mm
Printing Temperature	225ºC
Print Speed	15 mm/s
Enable Retraction	No
Generate Support	No
Built Plate Adhesion	No

In addition to the previous changes it is possible that the top layers of the flexible element may oscillate while it is printing, resulting in a misalignment of the top layers. The tendency to oscillate the non-fixed, and last printed extremity can be due to two factors. One of the factors is the movement of the base, in a backward and forward movement, that is then propagated to the extremity. The other factor is the movement of the extruder itself. While it deposits the material, and it adheres to the previous layer, the extruder, if it is not well calibrated, can start pulling the material and so the 3D piece.

All of these issues can be minimized. The first factor can be minimized by having a 3D printer where the base plan only moves in the vertical direction. And the second factor can also be minimized with proper calibration. Since the calibration was good, the factor that would contribute more to the oscillating of the part would be the first. To oppose such behaviour, since the available printer has a base plate that moves in the horizontal direction, it was modeled and printed, with *PLA*, a lateral support, seen in figure 3.18a). Each level of the support is separated in two halves so they can be placed sideways to the flexible as it is printed.

The last aspect to take into account when manufacturing any piece is the tolerances needed and the 3D printing process is no exception. However, one should have in mind that different printers may need different tolerances and different calibrations. For this printer and its current set up the tolerances used can be seen in the table 3.10. It is also important to notice that the tolerances will vary with the type and purpose of the part, for some parts it will still be necessary to sand for a better fit. In some elements the aim is to have a tight fit while others should have clearance more to move. Table 3.10 present the

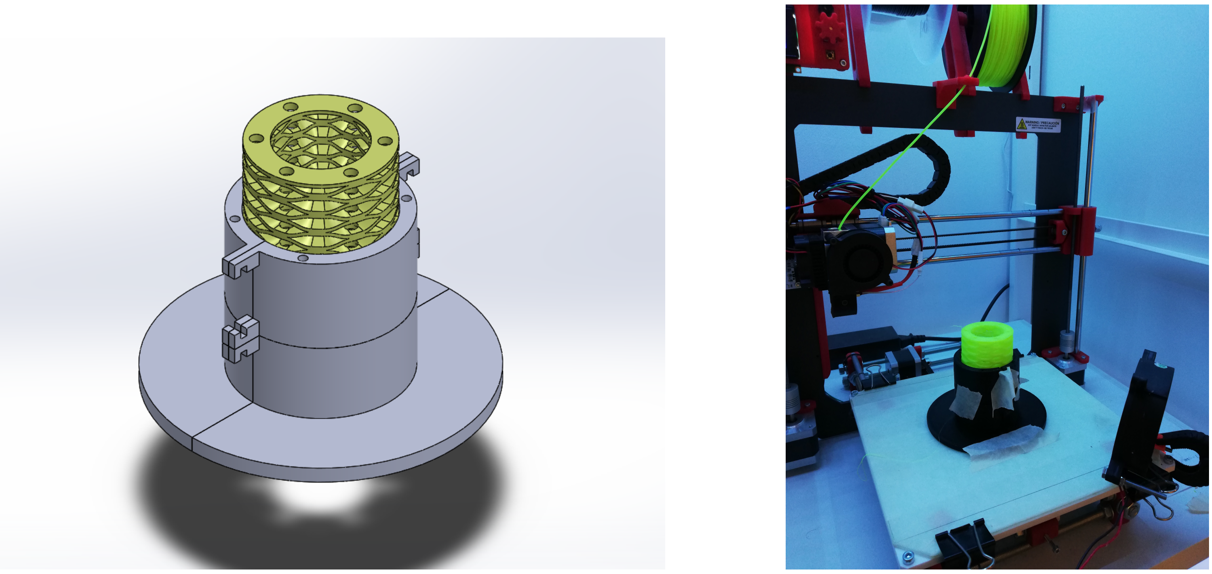

(a) 3D model of the printing support (b) Printing support used with the extra fan

Figure 3.18: Printing support for the joint

tolerances used.

Table 3.10: Printing Clearances

Part type	Material	Clearances	Example
Fitting piece with pressure	PLA	0.3 mm	Top holders
Fitting piece with pressure	FilaFlex	0.3 mm	Holes in the joint for the inner tubes
Part without movement without support material	PLA	0.2 mm	Nuts and Washers
Part without movement with support	PLA	1 mm	Motor fit in the actuator
Parts with movement	PLA	1 mm	Actuator Wheel and Actuator body

3.4.2 Other Materials

Since not every material used could be 3D printed, other materials were gathered in order to fully build the soft robot. The first important material to gather and retrieve information was that of the wires that were going to be used. These wires should fulfill certain characteristics such as:

- Be flexible, in order to bend accordingly with the joint and with the actuator's curvature;
- Be rigid enough to prevent buckling while in compression.

A stranded steel wire, with 2 millimeters of diameter fulfilled these requirements. During the tests, that were realized throughout the several prints, it was noted that the interaction between the steel wire movement and the flexible part was not the best due to friction. In light of this, it was fitted a nylon tube between the steel wire and the module with a low friction coefficient with an inner diameter of 2 mm and an outer diameter of 4mm.

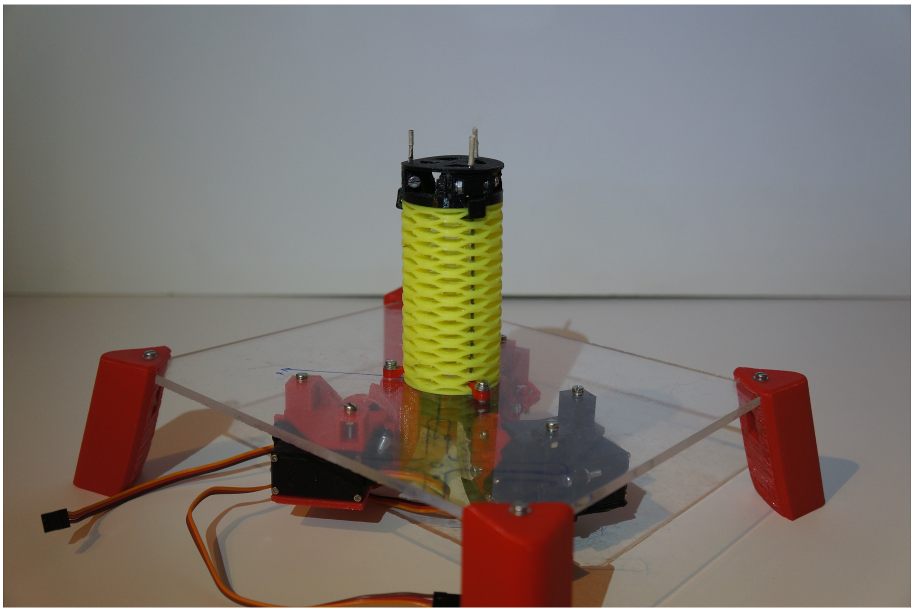


Figure 3.19: Fully assembled Soft Robot Prototype

3.5 Control

The current state of the prototype does not provide any feedback about its positioning and shape. The open-loop controller will therefore only depend of the correct modulation of the inverse kinematics of the manipulator to obtain the desired position. This lack of feedback reinforces the necessity to use a neural network to take into account all the possible nonlinearities and minimize the errors that could have been reduced with a feedback controller.

Figure 3.20 outlines the circuit used to control the actuators. The servo motors are powered by a power supply with 6.5 Volts.

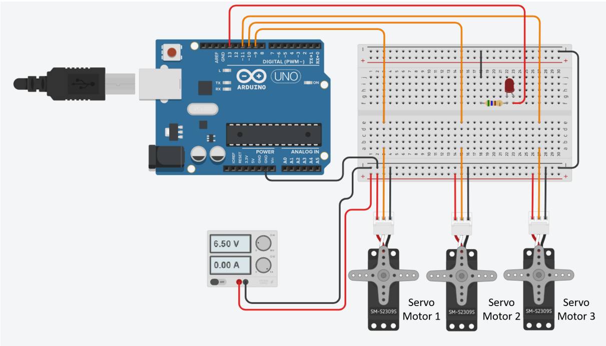


Figure 3.20: Experimental Arduino circuit

Chapter 4

Kinematics

With the prototype already assembled, the next step is to compute its kinematics. Two approaches were taken, a first one with a simplified theoretical model and a second one with an approximated model using neural networks. The first approach, with the theoretical model, follows the constant curvature assumption for the forward and inverse kinematics. Later, it was added an offset to the kinematics to approximate better the assembled prototype with all the elements. The forward kinematics can be easily adapted to this addition, however for the inverse kinematics it is a more complex process. Because of this, NN are going to be used for the computation of the inverse kinematics. These theoretical models will provide basic information about the properties of the prototype before moving on to the experimental model. It will provide an expected action space and a starting point to the training of the neural networks. The second approach will consider a set of collected data from the prototype to train the neural networks. With the collected data it is expected that the nonlinearities from the manipulator are learned, resulting in a better model. The nonlinearities of the prototype will not be modeled with the theoretical approach since it considers the manipulator to be perfect.

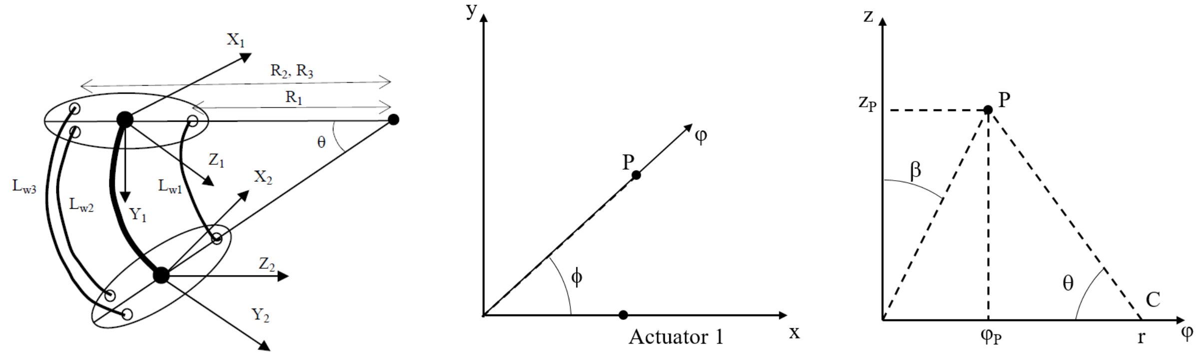
4.1 Constant Curvature Model

4.1.1 Inverse Kinematics for the flexible module

With the forward kinematics computed with the references on 2.1.5, it is now necessary to obtain the inverse kinematics models. These next equations will present how to calculate the inverse kinematics of the flexible module.

The previous references of the constant curvatures models did not provide a clear inverse kinematics for the wires length of a continuum manipulator with extension degree of freedom, for example [12] provides the inverse kinematics but for a continuum manipulator with a constant neutral line. Nonetheless, both [12], [8] and [30] provide some helpful equations to compute the wire lengths depending on the computed r and the angles θ and ϕ, equations (4.1), (4.3) and (4.7). The remaining equations were based on the assumption of a constant curvature of the module and the wires, figure 4.1 a). Some new variables have been introduced: lw_i - the length of the wire i, and ϕ_i - angle made by the desired point with the actuator i. (x_1,y_1,z_1) are the origin coordinates of the base frame, hence are equal to (0,0,0), and (x_2,y_2,z_2) are the origin coordinates of the end-effector position according to figure 2.6b). To obtain the length of the actuator, or wire, we follow the next equation.

$$lw_i = R_i\theta \tag{4.1}$$



(a) Geometry to find the wire's length from [12]

(b) Geometry to find ϕ

(c) Geometry to find θ

Figure 4.1: Geometry to find theta and phi using trigonometry

Where the equation $\theta = 2\beta$, from [12], and figure 4.1 result in the next relation.

$$\theta = 2\beta \Rightarrow \theta = 2atan2(\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, z_2 - z_1) \tag{4.2}$$

From equation (4.1) the last term that hasn't been defined is R_i. This term is the curvature radius of the actuator i, it is similar to the neutral line curvature radius, equation 2.7d.

$$R_i = r - bcos(\phi_i) \tag{4.3}$$

To obtain the radius r it is necessary to calculate the coordinates of its center. This center must be at the same distance as the origin and the desired point, and also in the same direction. The coordinates can be then calculated by the intersection of 3 planes, the first plane is the bisector plane between the origin and the desired point P, as any point on this plane will be equally distant from either point. The base plane, xy plane, since the center of the arc will be on the same plane as the base of the manipulator, and finally, the plane $Oz\varphi$ that is formed by the z axis and the point P. The plane $Oz\varphi$,is illustrated in the figure 4.1. From the equations of the previous planes it is possible to get the system of equations (4.4), which computes the coordinates x and y of the center. C_z will be equal to zero.

$$C = \begin{cases} C_x = \dfrac{x_1^2 + y_1^2 + (z_1 - C_z)^2 - x_2^2 - y_2^2 - (z_2 - C_z)^2}{2x_1 + \frac{2y_1y_2}{x_2} - 2x_2 - \frac{2y_2^2}{x_2}} \\ C_y = \dfrac{y_2}{x_2}C_x \end{cases} \tag{4.4}$$

The previous system of equations, (4.4), was also computed for the case where C_x is in function of C_y. This two set of equations are important because it takes into account the values where either

x_2 or y_2 are equal to zero and, therefore preventing a division by zero. The specific case, when both x_2 and y_2 are equal to zero the module is either in compression or extension, hence the wire length will automatically be equal to the height of the desired point. From the next equations it is possible to complete equation (4.3) and compute the wire length, equation (4.1).

$$r = \sqrt{(x_2 - C_x)^2 + (y_2 - C_y)^2 + (z_2 - C_z)^2} \tag{4.5}$$

$$\phi = atan2(y_2 - y_1, x_2 - x_1) \tag{4.6}$$

$$\begin{cases} \phi_1 = -\phi, \\ \phi_2 = \dfrac{2\pi}{3} - \phi, \\ \phi_3 = \dfrac{4\pi}{3} - \phi, \end{cases} \tag{4.7}$$

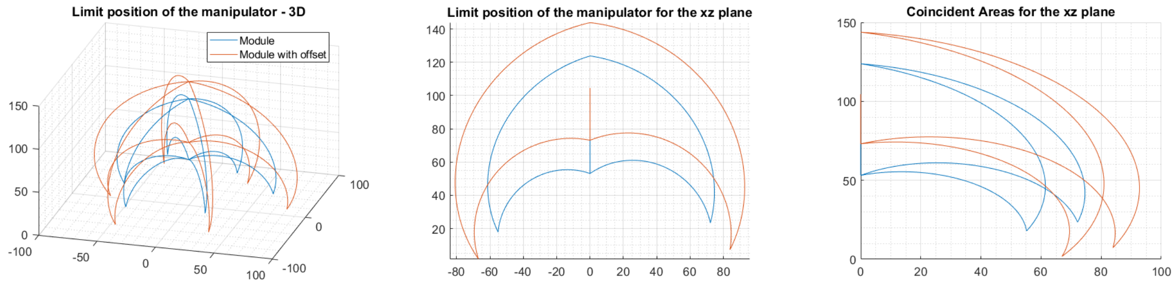
After obtaining the wires length for the desired position one is able to compute the angular position for each motor according to its geometry. Initially, it was necessary to define a base position for the actuators from which all the positions and angles are defined. This base, or neutral position, as the following parameters: β_n and l_n - where the first is the actuator angle and the second is the defined neutral length of the wires. To calculate β_i, the angle for each actuator, one must first evaluate the necessary increment of the wire length with $\Delta lw_i = lw_i - l_n$. With the increment calculated it is now necessary to evaluate the respective angle increment with, $\Delta\beta_i = \Delta lw_i/arc$, where $\Delta\beta_i$ is the angle variation needed to apply to the motors, and arc is the displacement applied to the wire when the actuator varies 1º. Finally β_i can be obtained as $\beta_i = \beta_n + \Delta\beta_i$.

As explained before, during the modelling and assembling of the manipulator it was necessary to add structural elements in order to control the manipulator. When these elements are added to the manipulator, the end-effector acquires an offset from the flexible module. The inverse kinematics for this constant curvature model with offset is a more complex system, [40] and so the use of neural networks. The training of neural networks with data from a simulated model will also provide a starting point for the training of the networks for the assembled prototype with regards to the minimum number of data points and to the method followed.

4.1.2 Theoretical Working Volume

From the forward kinematics it is possible to obtain the theoretical action space with an offset of 20mm relative to the rigid part at the top of the module, figure 4.2. The subtitle of the first plot is valid for the remaining graphics in this figure.

As examined in the previous figure, the limits in the two bending directions are not symmetric, however such difference was expected since the actuators are not symmetrically positioned either. If one remembers figure 2.6, the positive part of the x axis has the actuator position at the rightmost point, while the two remaining actuators are in between the leftmost point and the center of the manipulator.



(a) 3D Action Space (b) 2D Action Space - xz plane (c) 2D Coincident Areas - xz plane

Figure 4.2: Theoretical Action Space with offset

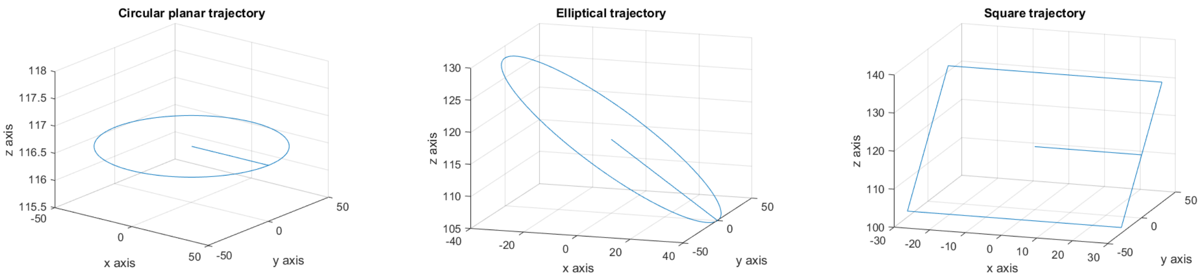
Such difference will produce different results in both bending directions. When bending to the right, the second and third actuator are closer to the neutral line thus producing a smaller rotation arm which leads to a greater extension of the leftmost side of the manipulator. The actuator number 1 is on the right edge and thus its minimum length will be the minimum length of this side of the manipulator. This positioning will result in a extension capable of reaching further to the side and less to the downwards. In contrast, when bending to the left, the rightmost side of the manipulator will only be as great as the actuator number 1, and the leftmost side of the manipulator will be smaller than the minimum wire length, as the actuators two and three will behave as a anchor point for the rotation. These properties will allow the manipulator to have a more pronounced curvature and less sideways reach. It is possible to perceive that the manipulator will bend more than 90 deg. This is accomplished since the previous calculations made were regarding the minimum height of the sides of the manipulator, and not for intermediate locations where the actuators 2 and 3, in this example, are positioned.

4.1.3 Trajectories used for Inverse Kinematics Validation

For a method of consistency it was used the same three different trajectories: a circular trajectory with a constant height, an elliptical trajectory, which is a circular trajectory with a variable height. And finally a square trajectory, in the base frame, with variable height. This trajectories can be seen in figure 4.3. The circular trajectory has a radius of 50 mm and maintains the same height, 116.53 mm. This neutral height is obtained when the manipulator is in its neutral position. The elliptical trajectory has a smaller radius with 40 millimeters and a minimum and maximum height of 105 and 128.06 mm, respectively ($\Delta = \pm 11.53mm$). The square trajectory has a side length of 60 mm with a minimum and maximum height of 101.53 and 131.53 mm ($\Delta = \pm 15mm$).

4.2 Artificial Neural Network Model

In Duka [40] the author obtains a model of a simulated planar three-link manipulator using neural networks. This paper, although only being applied to a simulated manipulator, provides a good base for the inverse kinematics of the present continuum soft robot. The number of inputs and outputs in this paper is identical to the proposed in this work, both have 3 inputs and 3 outputs. In this section the neural



(a) Circular Trajectory (b) Elliptical Trajectory (c) Square Trajectory

Figure 4.3: Trajectories used for the Inverse Kinematics

network form [40] will first be validated by accurately obtaining the inverse kinematics of a constant curvature model with offset. Next, it was attempted to obtain the minimum number of data points required for the training of the neural network while still obtaining a reliable model. This step also allows to know the minimum theoretical number of points required to be acquired during the experimental process.

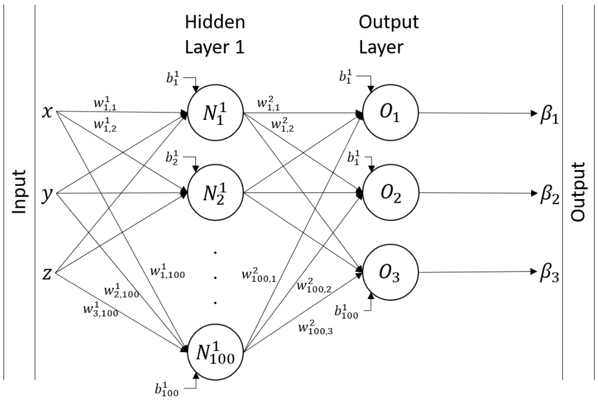


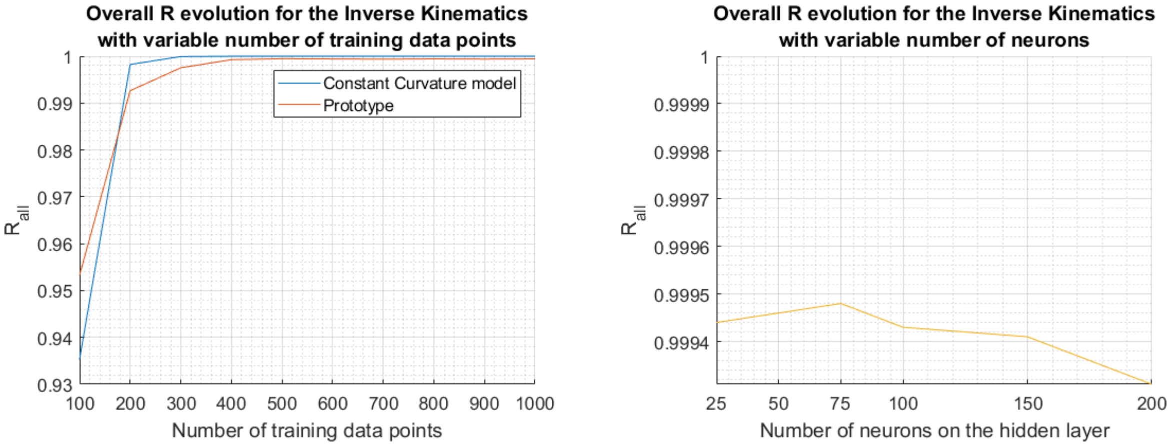
Figure 4.4: Neural Network Architecture

With the architecture of the network defined and the number of necessary points evaluated, it is now possible to, with the prototype data, train the neural network and obtain several models. With the data obtained it was possible to evaluate the impact of the number of training points. Later, it was also analysed the impact that the number of neurons used in the NN has on the performance of this prototype.

The architecture followed is a feed-forward network and is illustrated in figure 4.4. It contains 3 inputs, the coordinates of the desired position; 3 outputs, the angle position of the desired position; a hidden layer with 100 neurons and sigmoid activation functions; an input layer and an output layer with a linear activation function. The neural network was created with a toolbox from *Matlab*, trained using the Levenberg-Marquardt backpropagation algorithm, and evaluated with the mean-squared error. The samples for its training were divided in to three categories, training, validation and testing, each one with respectively 70%, 15% and 15% of the total number of samples.

To gather the information to train the neural networks it was created a set of 1000 random coordinates for the actuators angles. The output of these coordinates is then obtained from the constant curvature model and the prototype. The input-output pair (angles-position) will be switched to train the NN and obtain the inverse kinematics. The position will be the input of the NN and the actuators angles will be the desired target for the respective position. The use of the actuators angles as the output, rather than the length of the wires, is to minimize the errors that may be originated from these and evaluate the nonlinearites present with the NN.

After the neural network is trained a regression value, R, that measures the correlation between the inputs (desired position) and the outputs (actuators angles) is computed. A value closer to 1 imply a close relationship, while a value closer to 0 imply a more random relationship. In the next figure it is possible to analyse the evolution of R for the different situations.



(a) Variable number of training data points (b) Variable number of neurons on the hidden layer

Figure 4.5: Overall R evolution for the different Inverse Kinematics

After examining the evolution of R from the constant curvature model, one is able to assume that the minimum number required to properly model a theoretical model is 300. However, a final decision should only be made after obtaining the trajectories errors in 4.2.1. For the prototype model, if following the same reasoning, the values of R appear to only stabilize after the model trained with 400 data points, so it is likely that this will be the minimum number. However, it is still required to analyse the trajectories errors.

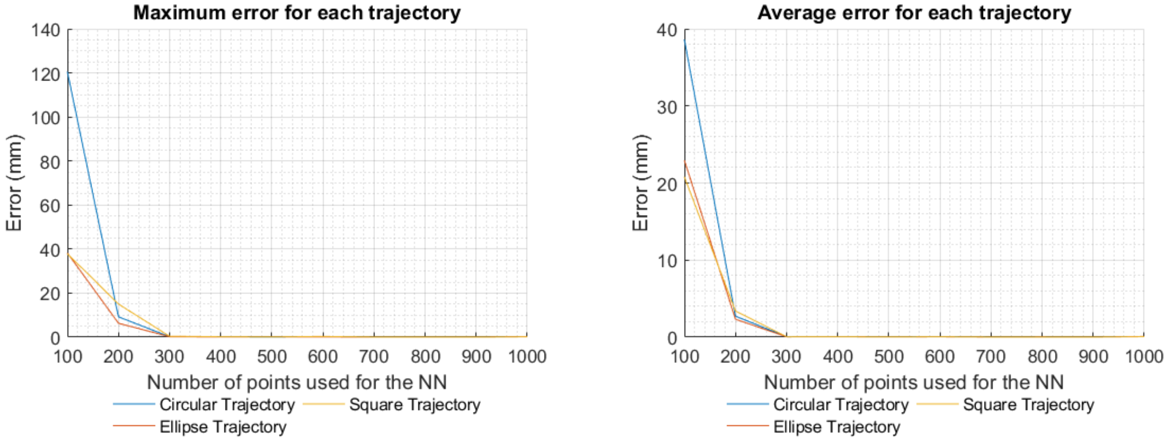
After the analysis of the results obtained with the previous models, in the experimental work, it was attempted to improve the results by varying the number of neurons on the hidden layer of the previous architecture. To train these new models five new networks were created with respectively 25, 50, 75, 150 and 200 neurons on the hidden layer and all the data previously collected was used. The model with 100 neurons was used from the previous models. For these new models, with a variable number of neurons, it is noted a somewhat descending trend as the number of neurons is increased. This can be due to two scenarios; one due to the number of neurons used require an increased number of training points, and the other due to over complicating of a rather simpler modulation.

4.2.1 Inverse Kinematics from a Constant Curvature model

With the inverse kinematics for the constant curvature model it is now possible to apply the trajectories previously mentioned, figure 4.3, and obtain the input angles. Images in the figure 4.6 represent the maximum and average error for the different trajectories of the different models.

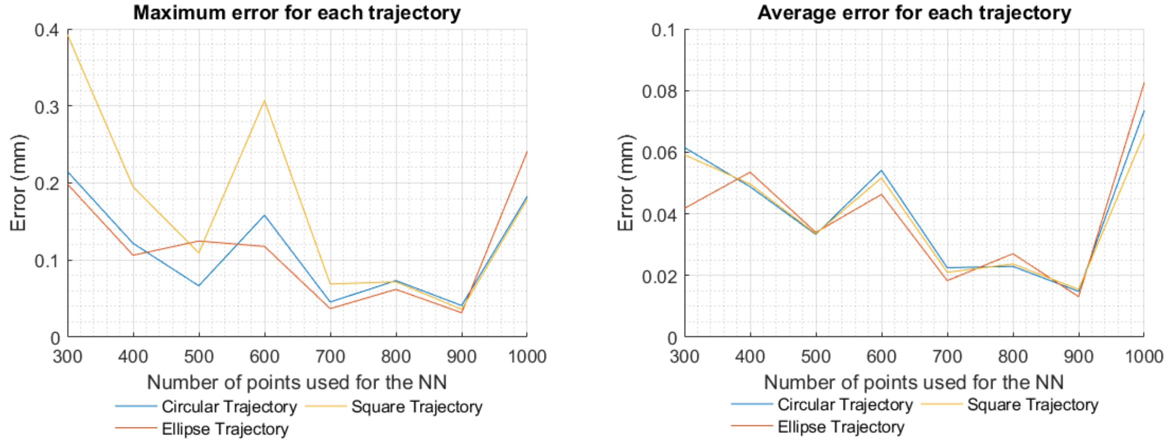
The definition of error used was the norm between desired point coordinates, ($[x_{des}, y_{des}, z_{des}]$), and the coordinates obtained, ($[x_{obt}, y_{obt}, z_{obt}]$), this value will always be positive and represents the 3D displacement.

$$error = \sqrt{(x_{obt} - x_{des})^2 + (y_{obt} - y_{des})^2 + (z_{obt} - z_{des})^2} \tag{4.8}$$



(a) Maximum error for each trajectory

(b) Medium error for each trajectory



(c) Maximum error for each trajectory zoomed

(d) Medium error for each trajectory zoomed

Figure 4.6: Maximum and average error with IK1 FK1

The error evolution confirms the assumptions made about the minimum number of training data points. Considering that both maximum and average errors, for the three trajectories, stabilize after the model with 300 data points. With this conclusion it is possible to assume that the best results for the prototype will occur after the model with 400 points. The results obtained for the constant curvature model in a simulated environment, validate the assumption that the method of [40] was suited for this manipulator and that a higher number of either neurons or data points was not necessary.

Chapter 5

Experiments and Discussion

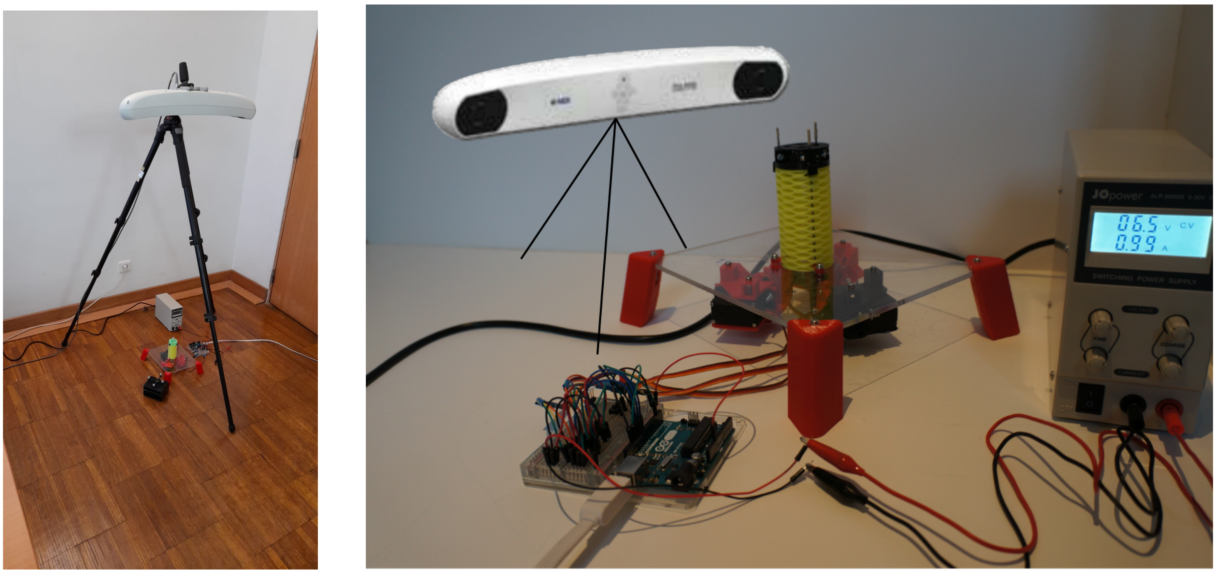
This chapter presents the tests that the manipulator was subjected to and the respective results. It begins with the characterization of the experimental setup, followed by an analysis of the properties of the manipulator; and then tests. These tests include: 1) tests to obtain the working volume; 2) tests with static and dynamic loading; 3) tests to compute the repeatability of the manipulator; 4) the procedure followed to collect the data for the neural networks and lastly 5) the validation of the several models with different trajectories.

5.1 Experimental Setup

The hardware required to preform the experiments and obtain a fully functional manipulator can be seen in figure 5.1 and consist of:

- Embedded controller - Arduino$^{\text{TM}}$ UNO Rev3;
- Power Supply - Jopower$^{\text{TM}}$ ALP-3005M 0-30V 0-5A;
- Servo Motors - Tower Pro$^{\text{TM}}$ MG996R;
- 3D Tracking System - Polaris Spectra$^{\text{TM}}$;
- Markers - Part Number 8700339 and single markers;
- 560 Ω resistor;
- Red LED.

From figure 5.2 it is possible to identify the relation between the two sets of markers used with the 3D Tracking system. Using the manufacturer's own software it is possible to designate a global reference frame on the rigid body, $O_P - X_P Y_P Z_P$ (frame in blue), with a part number of 8700339. In this figure the rigid body is attached to the base plate however in the latest experiments this reference frame will be moved outside the plate in order to better suit the test, figure 5.1a). The single marker at the tip of the robot module is used to track its position in the 3D space relative to the base frame, $(O - XYZ)$. The polaris software saves the data in a *.csv* file, comma separated values, that can be read by a *Matlab* function.



(a) Experimental Setup

(b) Experimental Setup close up

Figure 5.1: Experimental Setup

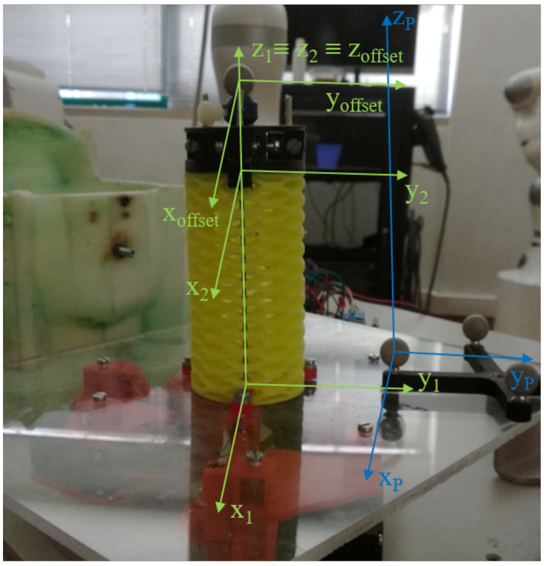
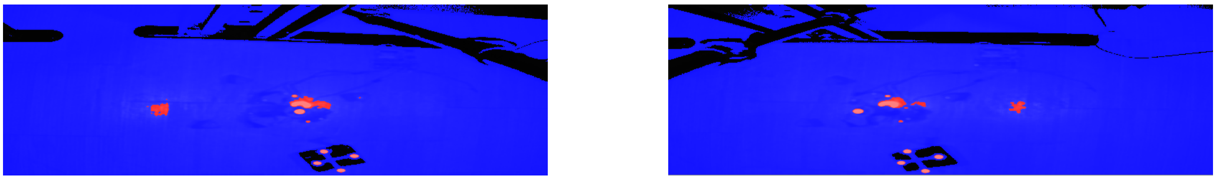


Figure 5.2: Definition of the frames

The pose presented, in the previous figure, shows the manipulator in its neutral position, in other words, the manipulator is in a state where there are no applied forces. This position in the remaining text is sometimes be referred as home, neutral or base position.

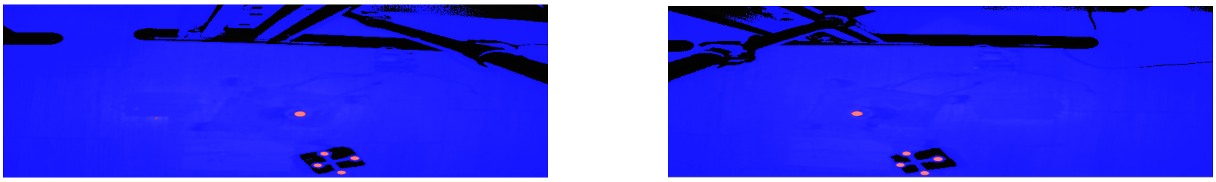
The schematics of the Arduino^{TM} circuit can be seen in 3.5.

In order for the Polaris^{TM} cameras to better detect the individual marker and in the largest possible volume, from the section 5.5 on, it was necessary to position them as represented in the figure 5.1 a), in a vertical position. The resultant perpendicular orientation of the cameras to the floor will result in some interferences due to the reflection of the infrared radiation on some surfaces. These interferences can be easily solved by covering them with some painter's tape, figure 5.3 displays the results. The vertical position of the system also produced some situations where the marker was hidden by the manipulator's own body. To surpass this setback an additional offset was added with 10 mm, as it can be observed on figure 5.2.



(a) Left Camera with interences

(b) Right Camera with interences



(c) Left Camera without interences

(d) Right Camera without interences

Figure 5.3: Software images with highlighted markers before and after the addition of painter's tape

The 3D tracking system does not send information back to the controller it only records the position of the manipulator's tip.

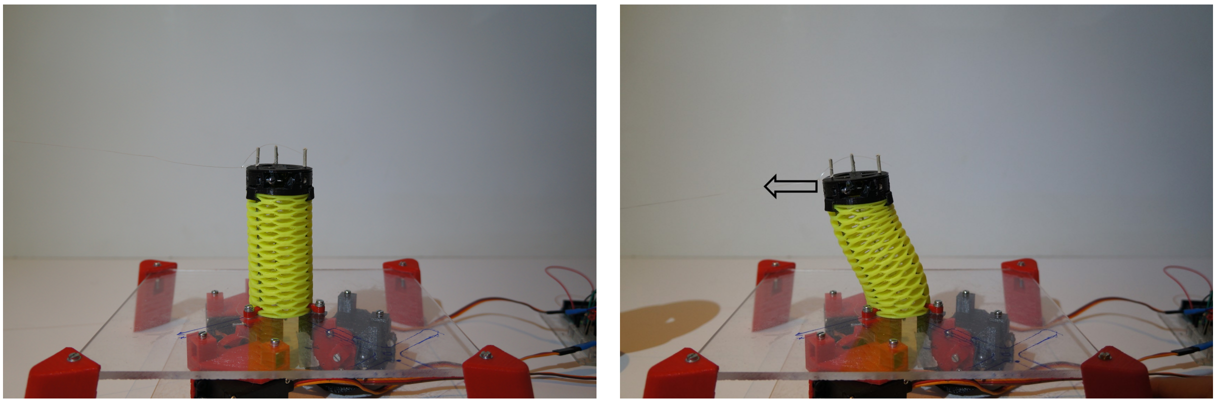
5.2 Manipulator properties

5.2.1 Passive Compliance

The passive compliance is more perceivable in this manipulator when a lateral force is applied to the tip of the module, figure 5.4. This figure displays two states of two different scenarios, a neutral and a bent pose, in both scenarios the applied force is planar with the top face of the top element. The resultant plane, after the force is applied, is approximately parallel to its initial plane.

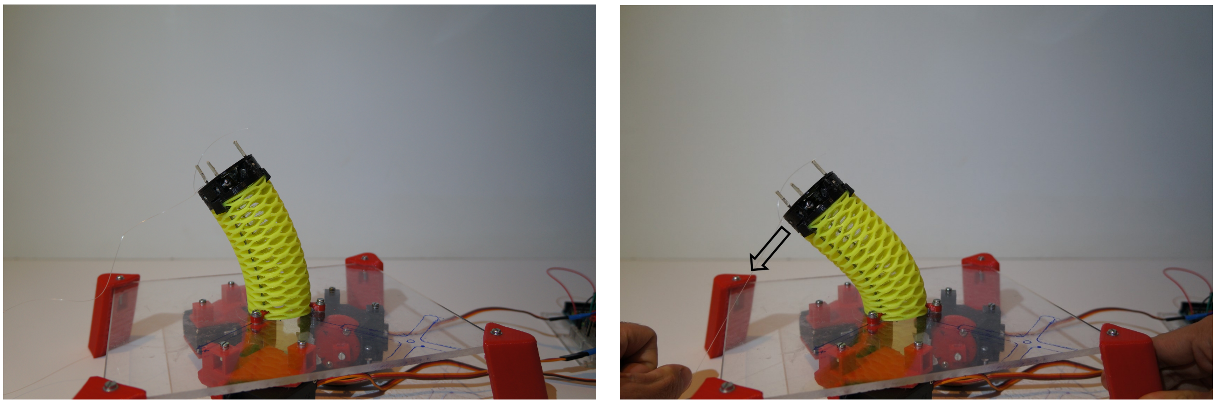
To better understand this property, Kim, Cheng, Kim, and Iagnemma [4] makes an analysis of a similar manipulator with a flexible backbone. This passive compliance is a consequence of the flexible material used to build the manipulator and of its actuation method. The actuation method relies solely on stranded steel wires that have no significant stiffness to prevent these forces. Since the length of the wires were maintained constant during this experiment it is possible to assume that the neutral line length, as in [4], is invariable throughout the experiments. When a lateral force is applied to the tip of the manipulator, it bends into an s-shape independently of the applied tension to the wires. Such behaviour is a consequence of the multiple configurations that the manipulator is able to acquire while maintaining the length of the wires, figure 5.5.

An analysis to the scenarios here presented and the theoretical behaviour of such manipulator, it is possible to distinguish a pattern when a force is applied. This pattern, s-shape, results in a part of



(a) Before force applied

(b) After force applied



(c) Before force applied

(d) After force applied

Figure 5.4: Passive compliance displacement. The force is applied to a wire connected to the manipulator tip

the manipulator to have a positive curvature, to one side, while the remaining of the manipulator have a negative curvature with the same magnitude, same curvature but on the other side. This behaviour is clearer when the manipulator is in the neutral position; however, when it is in a bent position, as observed in 5.4 c), it results in an overlap of the already present curvature with the forced curvature. In this scenario the manipulator is bent to the same side from where the force is pulling to, this applied force causes an increased curvature, decreased radius, on the bottom segment of the manipulator. On the other hand, the top segment of the manipulator is expected to bend to the opposite direction, leading this segment to be straighten as a result of the merging of two symmetric curvatures.

The bending resistance of the manipulator is a result of the stiffness of the flexible material and its geometry. The present topic will be studied in further experiments where two different positions, with approximately the same neutral line length, result in two different bending stiffness, section 5.4.1.

5.2.2 Motor - Sensibility Test

In this section, two properties will be studied, the theoretical displacement of the extremity of the manipulator and a simple visual experiment to test if the motor has good repeatability. As seen in section 3.3.1 and in equation 3.3b the diameter that the center of the wire makes is 45 mm. So the displacement that a degree has in the wires can be obtained as:

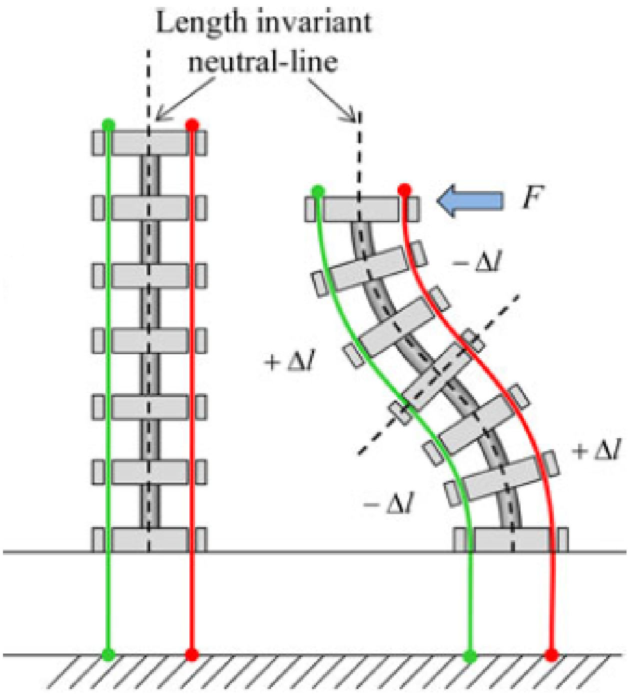


Figure 5.5: Passive compliance example from Kim, Cheng, Kim, and Iagnemma [4]

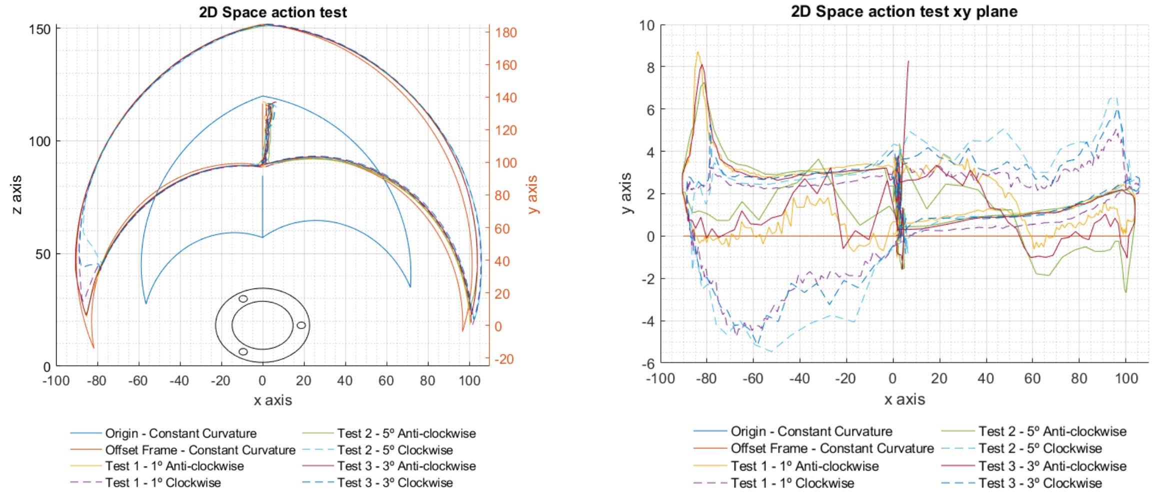
$$arc(\alpha) = 2\pi r_{actuator}\frac{\alpha}{360} \Rightarrow arc(\alpha) = 0.01745 * 22.5 * \alpha = 0.3927 * \alpha(mm)\,. \tag{5.1}$$

As seen in the equation 5.1, for every degree moved by the motor the wire will move 0.3927 mm.

Next, in the visual experiment, every motor is sent to several consecutive positions in order to test if it reaches the same position. The order of the consecutive positions start in the neutral position (corresponding to the natural length of the module), followed by the maximum extension (0°), minimum extension (180°) and finally back to the neutral position. In this experiment it was not noticed any measurable error. Nevertheless, a different phenomenon was observed when the desired angle is close to the last position of the motor. In these situations the motor may skip adjacent angles, for example, if the motor is sent to the positions 100°, 101°, 102° and 103°, it may move from 100° to 101°, not move from 101° to 102° and then move to 103°. This could be a result from the non linearity due to friction.

5.3 Experimental work volume

Discovering the action space is a necessary experiment in order to understand the limits of the manipulator. This test does not only serves to discover its limits but also to verify the theoretical model. To test the action space of the robot the motors were sent to their most extreme positions (0º and 180º). In this test the action space obtained was in the plane Oxz, meaning that the actuators 2 and 3 will move simultaneously and independently from actuator 1. The action space plane will have a periodicity of three and it will repeat itself every 120º. In figure 5.19 it is possible to observe such behaviour in the data collected for the training of the neural network. Figure 5.6 a) shows the location of the actuators in the plane xy for a friendlier analysis.



(a) 2D Action Space Obtained in the plane xz and actuators on the xy plane

(b) Error in the xy plane

Figure 5.6: Obtained 2D action space with constant curvature model

In the graphs presented here, figure 5.6, it is possible to analyse several lines. Two of them represent the perfect constant curvature model and are a theoretical result from the kinematics computed in the sections above. One of them represents the origin of the extremity of the flexible module while the other represents the offset added. This added offset accounts to, besides the pieces needed to assemble the prototype, an offset added to prevent the marker from being hidden by the manipulator own body. The added elements do not have a curvature and for that reason they maintain the same direction of the base they are attached to.

For this test, six different trajectories were analysed, three of them follow an anti-clockwise rotation and the other three a clockwise. The difference between the three trajectories in the clockwise trajectory, or anti-clockwise, is the input interval between angles of consecutive points. All trajectories start and end in the neutral position with coordinates (0;0;116,53) from the motors input (100º,100º,100º).

The perfect model obtained from the direct kinematics was based on the range of the motors being from 0º to 180º. However, after comparing the trajectory obtained and the perfect trajectory it was noticed that the trajectory obtained had less compression and extension displacement than that of the perfect module. In light of this, an analysis was conducted to the free rotation of the motors without being constrained by the wires. As expected, the obtained working range was inferior to 180º, approximately 165º-170º, which results in a lower compression and extension capability. Considering this result the perfect kinematic model was modified to fit this interval. The adaptation gave a better outcome as the trajectories obtained fit better in this theoretical trajectories, figure 5.6 a).

The greatest errors in the trajectories occurs in the two lowest points of both sides, where a set of motors is completely compressed and the remaining set are completely extended. This difference can be a result of a non perfect curvature area in both the top and base of the module due to the added elements, both on top and on the base. Of these two, the biggest error occurs in the left side of the graph, when the actuators 2 and 3 are completely retracted and the actuator 1 is fully extended, in the clockwise rotation. When the manipulator is fully compressed there are some clearance between the layers, however they may not be enough when the manipulator is sent to this position. Since the actuators 2 and 3 are not at the leftmost position of the module, and behave as a pivot point, the leftmost point tend to be even more compressed than the actuators and possibly compressing the material once there is no gap. As written before, the biggest error occurs in the clockwise rotation, such performance can be a consequence of the number of motors that are actuated simultaneously. In the clockwise manner all the actuators are compressed and only one of them extends, while in the anti-clockwise manner all actuators are extended and two of them compress. In the first scenario only one actuator forces the compression of the material at the left most point, while on the second scenario the applied force is from two actuators instead of one. This difference might be the reason for such error.

5.4 Loading tests

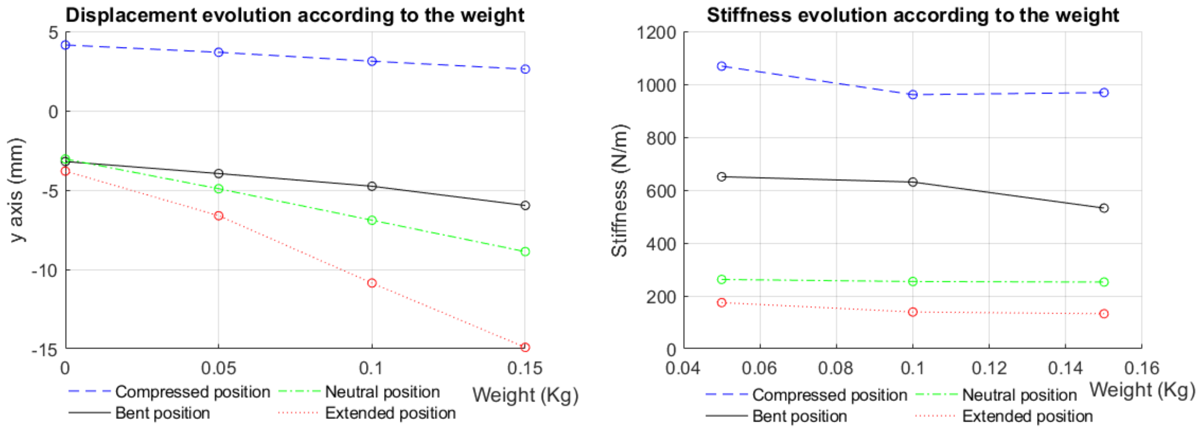
To better understand the behaviour of a manipulator it is indispensable to understand how it responds to load. In view of this, two loading tests were performed, a static and a dynamic in four different loading scenarios, a loadless test and tests with 50g, 100g and 150g placed at the extremity of the manipulator.

5.4.1 Static Loading

(a) Experimental Setup without load (b) Experimental Setup (c) Experimental setup

Figure 5.7: Static and Dynamic Loading Setup

In the static loading test several loads were applied to the manipulator in four different poses and each pose considered reflect a situation that the manipulator may encounter during its work. The chosen positions are: the neutral pose, which corresponds to the originally undeformed shape; the fully contracted (compressed) posture, with a tip displacement of 28.4 mm, with the smallest bending arm;



(a) Displacement from the static loading test (b) Experimental setup

Figure 5.8: Displacement and Stiffness from the loading test

the fully extended pose, which contrary to the previous posture has the biggest possible bending arm, with a tip displacement of 35.4 mm; and a bent pose. The bent position aims to compare an off-axis with an on-axis loading with roughly the same neutral line length. The considered on-axis pose was the neutral and the off-axis position results from the input of (170º,30º,30º). This input, accordingly to the theoretical models, leads to an increment in the neutral line of 8.1 millimeters and a rotation along the vertical direction of 99.5º. Figure 5.7 displays the bent position with the off-axis configuration.

Table 5.1: Static Loading test Displacement

Pose	Displacement (mm)			Color
	50g	100g	150g	
Contracted	0.5 mm	1.0 mm	1.5 mm	Blue
Bent	0.8 mm	1.6 mm	2.8 mm	Black
Neutral	1.9 mm	3.9 mm	5.8 mm	Green
Extended	2.8 mm	7.1 mm	11.1 mm	Red

Table 5.2: Static Loading test Stiffness

Pose	Stiffness (N/m)			Color
	50g	100g	150g	
Contracted	1068.3 N/m	960.5 N/m	968.4 N/m	Blue
Bent	650.3 N/m	630.2 N/m	531.8 N/m	Black
Neutral	262.1 N/m	254.3 N/m	252.2 N/m	Green
Extended	174.6 N/m	139.0 N/m	132.5 N/m	Red

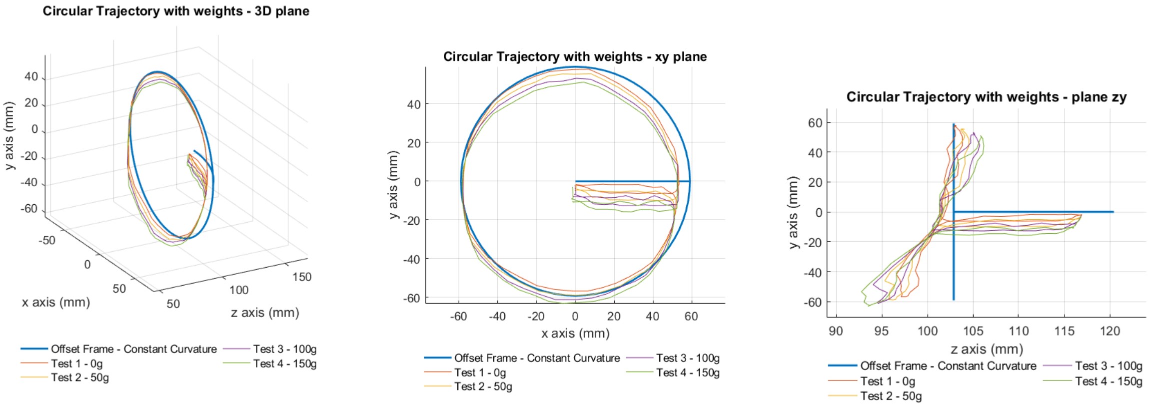
From the obtained results and displayed in figure 5.8, it is noted that even though the neutral line is longer in the bent position the tip displacement is smaller and therefore, a greater stiffness. In the remaining poses, the manipulator is subject to a lateral force that exploits the passive compliance of this type of continuum manipulators, section 5.2.1. Variously, when in the bent pose, the prototype seems to better counteract this passive compliance due to its geometry. Such behaviour can be attributable to the fact that the force, despite being lateral, is not applied in the plane formed by the neutral line curvature and its center. In such case, when a force is applied in this plane the geometry does not have impact to prevent the passive compliance. This deduction should be further investigated in future works to better understand this relation. Nonetheless, this is the desired result since it shows that this design is able to sustain torsional forces. Table 5.1 presents the displacement for the different weights in different poses. The displacement is measured from the initial loadless position of each pose. From these values it was also created a stiffness table 5.2. Both tables are illustrated on the graphs of figure 5.8.

As the displacement values suggest the stiffness in the bent pose is more than two times greater than for the neutral pose. The biggest value of stiffness is, however, from the compressed pose as it has almost no gaps in between the several wave levels resulting in almost solid element.

From the previous stiffness values it is possible to make a comparison between the finite element analysis, figure 3.6b), and the experimental work in this section. From these two experiments it is only possible to compare the bending stiffness of the manipulator with 6 waves on the neutral position. The bending stiffness of the prototype is more than 4 times greater than the simulated stiffness, a result likely derived from the added structural elements namely the stranded steel wires, as explained in 5.2.1.

5.4.2 Dynamic loading

The second loading test is dynamic and its goal is to verify if the results so far accomplished resonate in the expected performance of the module. In this experiment the manipulator is subjected to two trajectories, each one realized with the same 4 different weights as the previous test set (0g, 50g, 100g, 150g). From the previous static loading test it was also maintained the prototype position and the starting position was maintained throughout the different trajectories as figures 5.9 and 5.10 indicate.



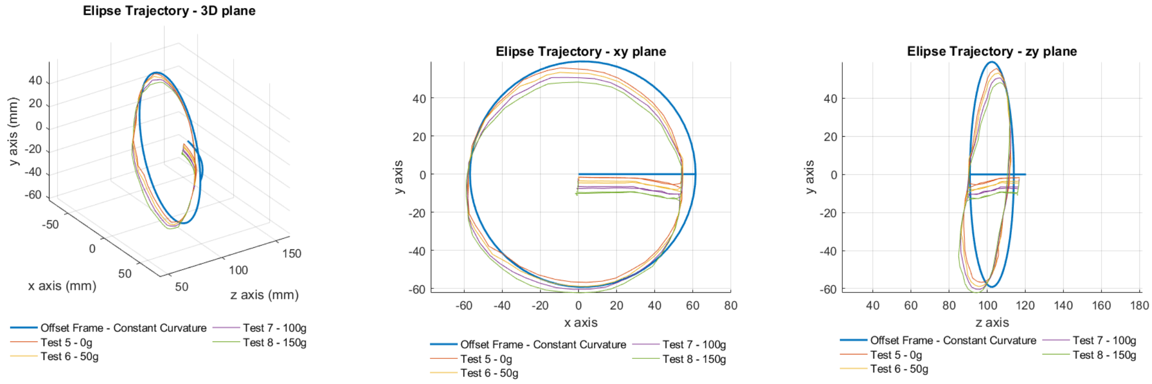
(a) Circular Trajectory planned and obtained with weights in the 3D plane

(b) Circular Trajectory planned and obtained with weights in the plane XY

(c) Circular Trajectory planned and obtained with weights in the plane ZY.

Figure 5.9: Circular Trajectory for the constant curvature model with weights

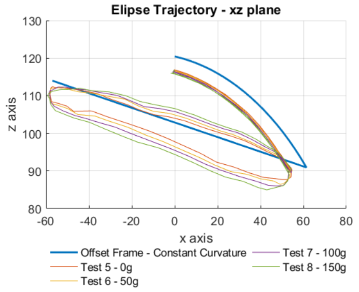
From figures 5.9 and 5.10 one is able to identify the planned trajectory for both circular and elliptical paths in light blue. In this test we do not have yet a Neural Network to compute the inverse kinematics of the module, instead the theoretical forward and inverse kinematics were used. The inverse kinematics model only contemplates the flexible module of the prototype, however the forward kinematic model is



(a) Elliptical Trajectory planned and obtained with weights in the 3D plane

(b) Elliptical Trajectory planned and obtained with weights in the plane XY

(c) Elliptical Trajectory planned and obtained with weights in the plane YZ



(d) Elliptical Trajectory planned and obtained with weights in the plane XZ.

Figure 5.10: Elliptical Trajectory for the constant curvature model with weights

able to incorporate the remaining elements. Taking this into account, some steps were required to obtain the desired trajectory of the marker located at the tip of the manipulator. First, it was chosen the desired trajectory of the flexible module. Then, with the inverse kinematics, it was calculated the motors input for the trajectory. In the final step, the previously calculated inputs are used to compute the desired trajectory with the forward kinematics with the necessary offset. Since the goal of the present test is to analyse its performance in a dynamic scenario, the previously obtained trajectories are enough to analyse such behaviour.

The present results don't show any considerable flaws in the prototype performance. It is shown, in figures 5.9 and 5.10, an expected displacement in the vertical direction, y axis, as a consequence of the different weights added. A second detail present in both plots is the fact that there isn't just a displacement in the y axis but also in its perpendicular direction, z axis. Taking as an example the circular trajectory, the manipulator tip does not only describe a trajectory with a vertical error due to the gravity force but rather describes a diagonal one, figure 5.9. As the paths are inspected it is realized that as the weight increases the diagonal angle increases. This diagonal path results from a downward and forward or backward displacement depending on the tip orientation. If the manipulator has a slightly upwards orientation the movement will be downwards and forward, however if the orientation is slightly downwards the movement will be also downwards but backwards. This behaviour can be linked to the passive compliance of these types of tendon driven continuum manipulators, as explained in 5.2.1 and visible in figure 5.5, and the orientation of the extremity of the manipulator.

5.5 Repeatability Test

Repeatability is an important parameter to characterize the behaviour of a manipulator and its appropriate type of work. It is defined as the ability of the manipulator to reach the same exact location several times. To analyse the manipulator repeatability it was randomly generated 5 different inputs for each actuator to conduct various tests.

The repeatability definition was obtained from [41] where it was applied to an industrial robot and was defined as a sphere with a center located on the average coordinates of the individual points, and a radius equal to the value of repeatability RP_l, equation 5.2a. Repeatability, from [41], can be obtained as follows:

$$RP_l = \bar{l} + 3S_l \, , \tag{5.2a}$$

$$\bar{l} = \frac{1}{n}\sum_{j=1}^{n} l_j \, , \tag{5.2b}$$

$$l_j = \sqrt{(x_j + \overline{x})^2 + (y_j + \overline{y})^2 + (z_j + \overline{z})^2} \, , \tag{5.2c}$$

$$S_l = \sqrt{\frac{\sum_{j=1}^{n}(l_j - \bar{l})^2}{n-1}} \, . \tag{5.2d}$$

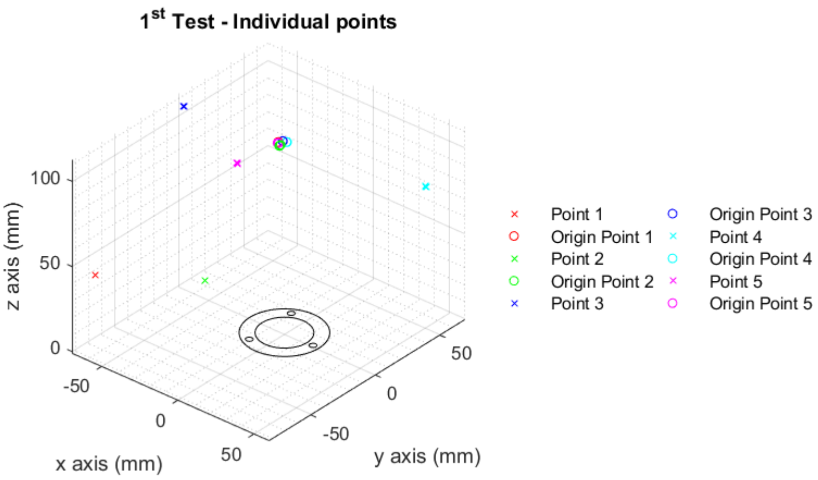
To study the repeatability of the prototype three different tests were delineated, 5.5.1, 5.5.2 and 5.5.3, with five sets of randomly generated coordinates. Table 5.3 displays the generated coordinates. The first test, 5.5.1, consists of sending the manipulator to one position and then back to the neutral position 10 times. This cycle is repeated for every set of coordinates and the data is recorded when the destination point is reached. The second test, 5.5.2, the manipulator is sent to the several positions following the numerical order of the points. In other words the manipulator starts in the neutral position, then is sent to the first position and so on until it reaches the last position. At this point the manipulator is sent back to the neutral position. This cycle is also repeated 10 times, in this experiment the manipulator is not sent to the neutral position after each point . Finally the third test, 5.5.3, is very similar to the previous experiment however it follows a random order of points for each cycle.

Table 5.3: Input coordinates of each point

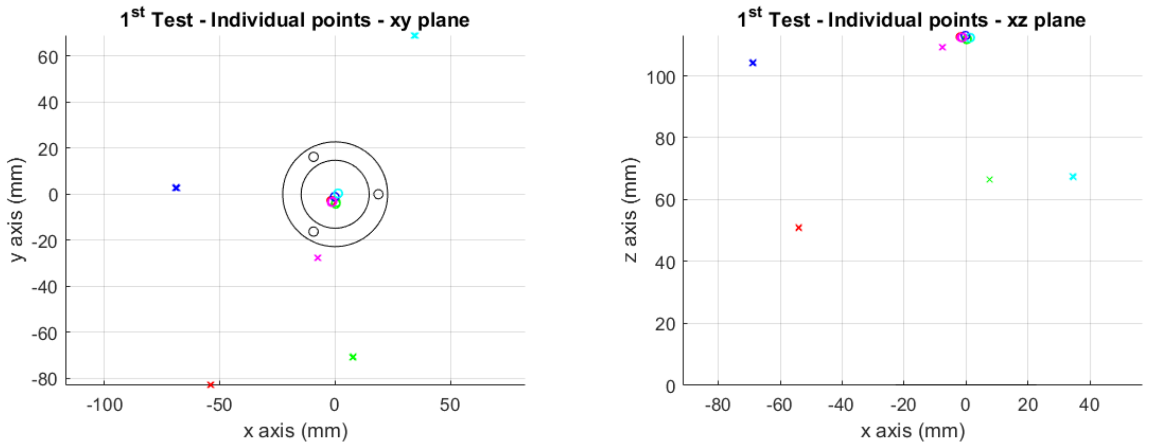
Point	Motor 1	Motor 2	Motor 3	Color
1	8º	17º	148º	Red
2	125º	57º	171º	Green
3	6º	79º	69º	Blue
4	138º	143º	34º	Cyan
5	88º	80º	116º	Mangenta

Table 5.3, show the colors association for the graphics presented.

The points with the input coordinates from the table 5.3 result in the coordinates seen in the figure 5.11 from the first test that was performed.



(a) First Test - 3D space



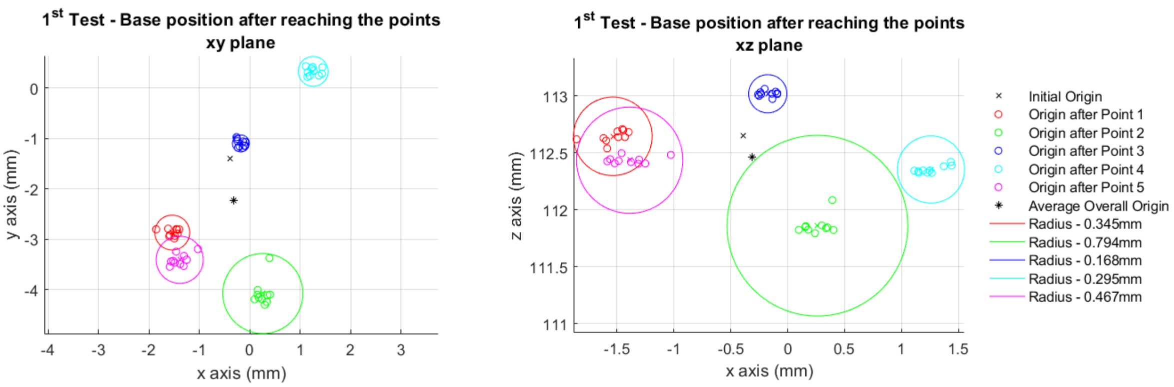
(b) First Test - xy plane (c) First Test - xz plane

Figure 5.11: Repeatability Test - 1

5.5.1 Test 1 - Individual Points

As explained before, the first experiment performed is to better understand the repeatability of the several destination points when originated from the neutral position. The procedure followed a cyclical behaviour for each data point. The manipulator starts in the home position before it is sent to the desired position and then back to the home position. This process is repeated 10 times for each of the five designated points, table 5.3, (N → P → N).

The first analysis of the results obtained from this experiment, figure 5.11, will focus on the positioning of the neutral or home coordinates, later it will be discussed the repeatability of the destination points. The results from the neutral position can be seen in 5.12, this position is both the origin position and the destination position from the off-axis position. The first property that is observable is the clustering of the origin points depending on where the manipulator returns from. This behaviour can be linked to the hysteresis of the flexible module as, when the manipulator is bent in a direction, it tends to pull its origin into the same direction of the movement, figure 5.12 a). For each cluster of points it is possibly, from the equation 5.2, to calculate the repeatability of each. In the legend of the figure and in the table 5.4 it is



(a) First Test - Neutral Position in the xy plane

(b) First Test - Neutral Position in the xz plane

Figure 5.12: Test 1 - Repeatability test from the neutral position

possible to read these values. This table also displays the repeatability values of the destination points and both the average repeatability.

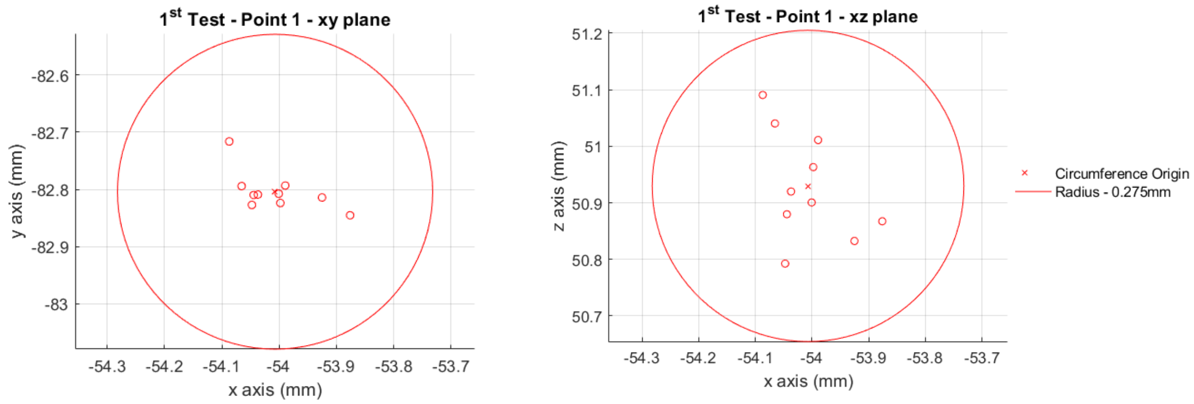
Table 5.4: Repeatability radius for test 1

Point	Repeatability Radius	Origin Repeatability Radius
1	0.275mm	0.345mm
2	0.192mm	0.794mm
3	0.528mm	0.168mm
4	0.402mm	0.295mm
5	0.354mm	0.467mm
Average	0.3501mm	0.4139mm

From the observed values, regarding the origin repeatability, there is one value of repeatability that is in contrast with the rest. This radius, from the origin of the second point, is the only value that is clearly larger than the average, the remaining radius are smaller or close to the average. These results show a discrepancy in a single point, also seen in figure 5.12. Without this point the repeatability radius would be certainly lower and so would be its average. The mean coordinates of the neutral position were also computed, as seen in the previous graphs of figure 5.12. The coordinates of this point came close to the initial origin, with a distance of 0.84 mm between the two of them, however it is understandable that if an experiment with more points was performed the average position would be closer to its real origin.

Similarly, the destination points have a repeatability value in the same magnitude as the origin. Such results may lead to a conclusions that if the manipulator follows the same exact path with the same input, the repeatability would be very good, as seen in the table 5.4. In figure 5.13 it is possible to analyse the coordinates of the several coordinates for the point number 1 with the repeatability sphere and mean position.

The graphs of the remaining points can be seen in the annexes C.1.1. After analysing these graphs it is possible to notice an outlier in the coordinates of point 2, figure C.1a) and b), however this outlier does not cause a radius of repeatability to be bigger than the average but rather a much smaller one. In the end this point does not have a meaningful impact on the results contrary to the outlier on the origins.



(a) First Test - Point 1 in the xy plane (b) First Test - Point 1 in the xz plane

Figure 5.13: Test 1 - Repeatability for Point 1

5.5.2 Test 2 - Points in a defined order

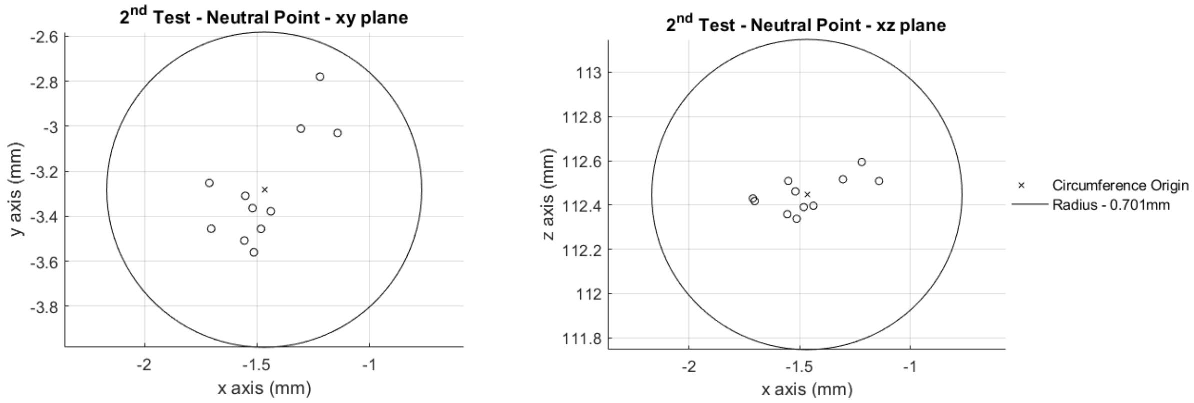
In this second test the manipulator is sent to the different points following the numerical order, proceeded by the neutral position. This experiment is similar to the previous one due to both tests maintain, throughout the time, the initial position that precedes each point. In the first test the initial position is always the neutral one while in this test it depends on the numerical order and is maintained during the 10 cycles that the manipulator goes trough (N → P1 → P2 → P3 → P4 → P5 → N).

Table 5.5: Repeatability values for test 2

Point	Repeatability Radius
1	0.7088mm
2	0.3172mm
3	0.5567mm
4	0.3910mm
5	0.5472mm
Origin	0.7007mm
Average	0.5369mm

The values obtained for the repeatability for each point and for the average are very similar to the values of the previous experiment.

In figure 5.14 it is possible to analyse the origin coordinates throughout the test when originated from the fifth position. The remaining graphs can be also seen in appendix C.1.2 as well as the overall position, in a 2D and 3D graph. Some interesting coordinates to analyse are from point 1, figure C.4a) and b) where in the plane xy the coordinates could form a compact sphere but when the xz plane is visualized it can be seen that the points form a vertical column.



(a) Second Test - Neutral Point in the xy plane (b) Second Test - Neutral Point in the xz plane

Figure 5.14: Test 2 - Repeatability for the Neutral Point from the point 5

5.5.3 Test 3 - Points in a random order

For the third and final experiment in this subject, the structure of the previous test was maintained however the order of each cycle was randomly generated. The random orders followed for each cycle are presented in table 5.6.

Table 5.6: Random Generated Order for the third test

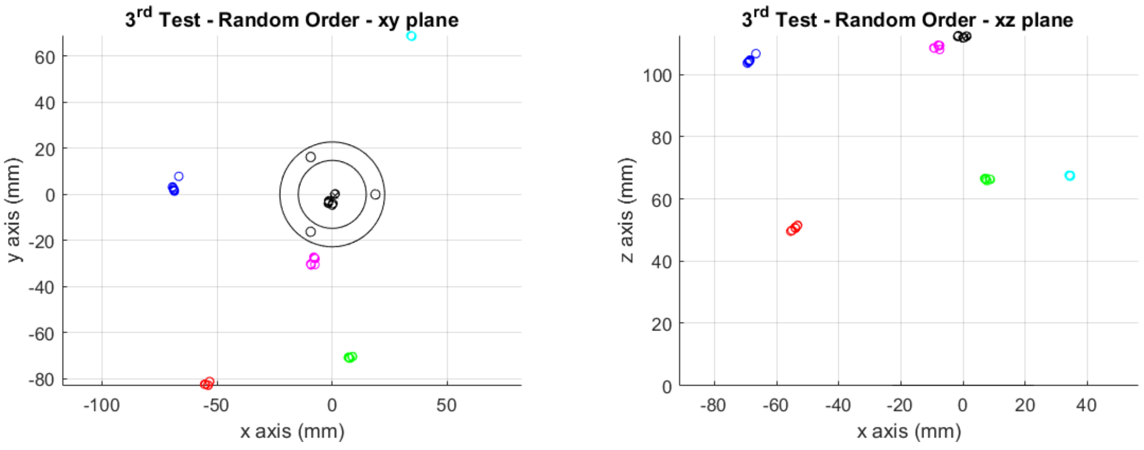
Cycle	Point 1	Point 2	Point 3	Point 4	Point 5	Point 6
1	5	2	1	3	4	N
2	2	1	3	4	5	N
3	5	4	1	3	2	N
4	5	3	1	4	2	N
5	2	3	4	1	5	N
6	3	4	5	2	1	N
7	4	2	3	1	5	N
8	2	3	1	5	4	N
9	1	4	3	2	5	N
10	1	4	5	3	2	N

Table 5.7 presents the repeatability radius obtained from the several points. From the figures 5.15, 5.16 and the remaining figures in the appendix C.1.3 one is able to notice that the data points are more scattered in space, when compared to the exact same input from the previous experiments, due to the random order followed and the resultant hysteresis.

Table 5.7: Repeatability values for test 3

Point	Repeatability Radius
1	2.490mm
2	1.584mm
3	6.199mm
4	0.477mm
5	2.893mm
Neutral	4.285mm
Average	2.9881mm
Median	2.6915 mm

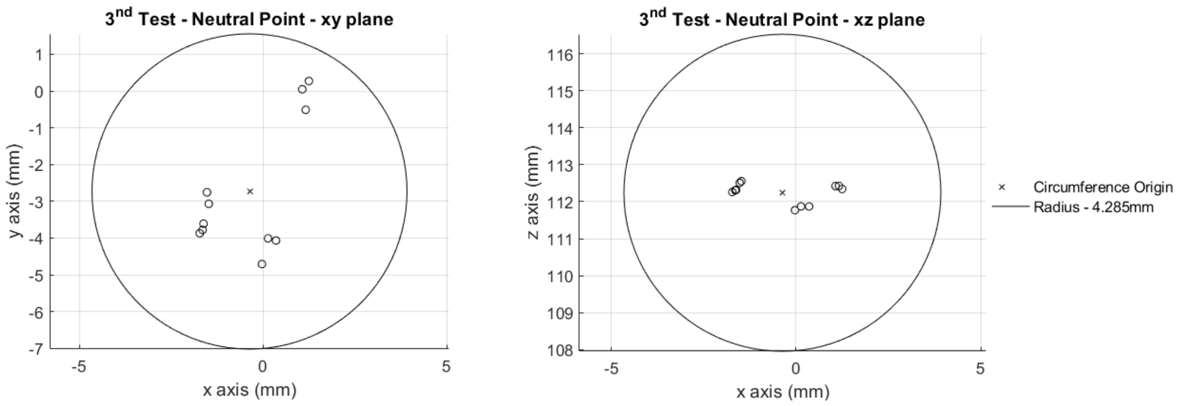
(a) Third Test - 3D space



(b) Third Test - xy plane

(c) Third Test - xz plane

Figure 5.15: Repeatability Test - 3



(a) Third Test - Neutral Point in the xy plane

(b) Third Test - Neutral Point in the xz plane

Figure 5.16: Test 3 - Repeatability for the Neutral Point

5.5.4 Comparison

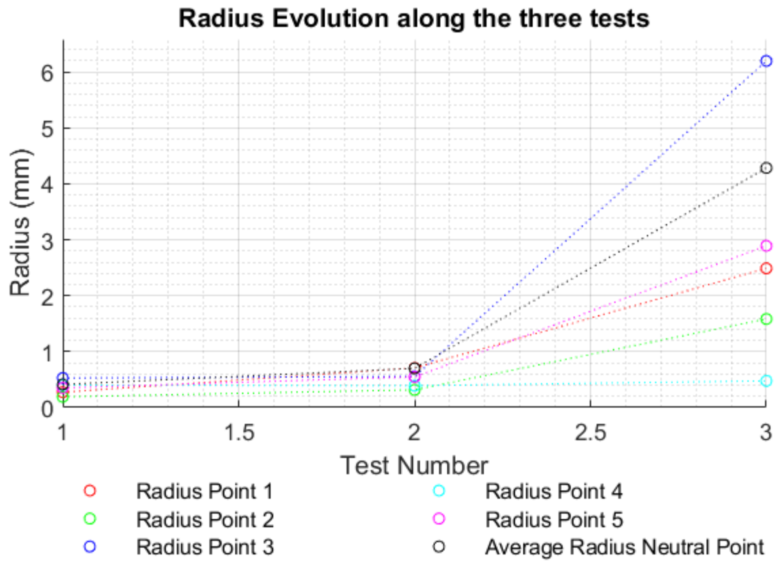


Figure 5.17: Repeatability evolution through the different tests

With the collected data from the previous three experiments the graphic on figure 5.17 can be plotted. As the graphic and the tables presented earlier show, experiment one, 5.5.1, and experiment two, 5.5.2, are in the approximately same order of magnitude, inferior to one millimeter. For the third test, however this order of magnitude is significantly increased due to the uncertainty of the path taken by the manipulator. If one computes the average radius for the points in the first experiment, including the neutral position, one would get a mean radius of 0.382 mm. The average value of the second and third tests have already been calculated in the previous tables. As we go from a simpler experiment, with a strict order, to a more complex experiment, with an aleatory order, the mean radius increases. From the first to the second experiment the increment is of 40.5%, 1.405 times over the previous radius value $(0.382mm \rightarrow 0.5369mm)$. The biggest increment occurs when it is added to the system a random order, this increment from the second to the third test is of 456.5%, 5.655 more than the result of the second experiment (0.5369 mm $\rightarrow$ 2.9881 mm). The repeatability radius, up to here studied, may be affected by not only some clearance present in the prototype actuators, in the gap between the wires and the flexible module, but also from a lack of precision from the motors that operate the actuators and mainly from the hysteresis from the flexible module.

5.6 Data collection for the training of the ANN

A fundamental step before the training of the neural networks is the process of data acquisition. In this step, the 1000 sets of motors inputs created for the constant curvature inverse kinematics were used and the manipulator response to each was then recorded using the Polaris Spectra^{TM}, using the setup visualized in 5.1. Nonetheless, three points weren't detected by the software as they were still hidden by the manipulator's own body. These undetected coordinates will not have a meaningful impact in the training of the neural networks as there are enough data points collected and the desired trajectories will not go through these extreme positions. The addition of a even longer offset would not result as it would collide with the base plate.

The cloud of all the points collected is seen in figures 5.18 and 5.19, the red marker indicates the neutral position. As figure 5.19 a) shows, the limits of the action space are not equidistant from the center of the manipulator but have a frequency of 120º, as expected from the theoretical space action from the constant curvature model. The limits that are further from the origin, in the xy plane, have the same direction as the actuator, where the limits that are closer to the origin have the opposite direction. The actuator positions are illustrated in both figures 5.18 and 5.19.

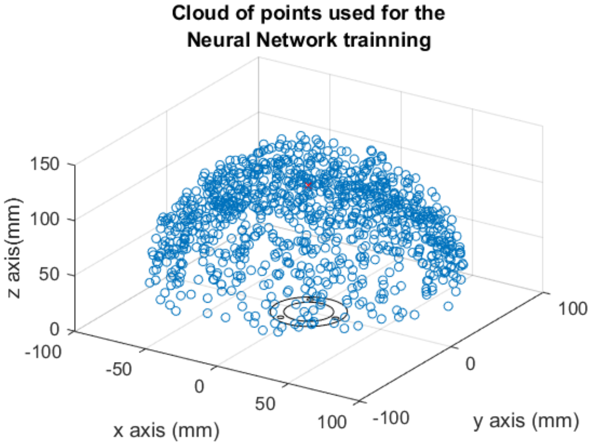
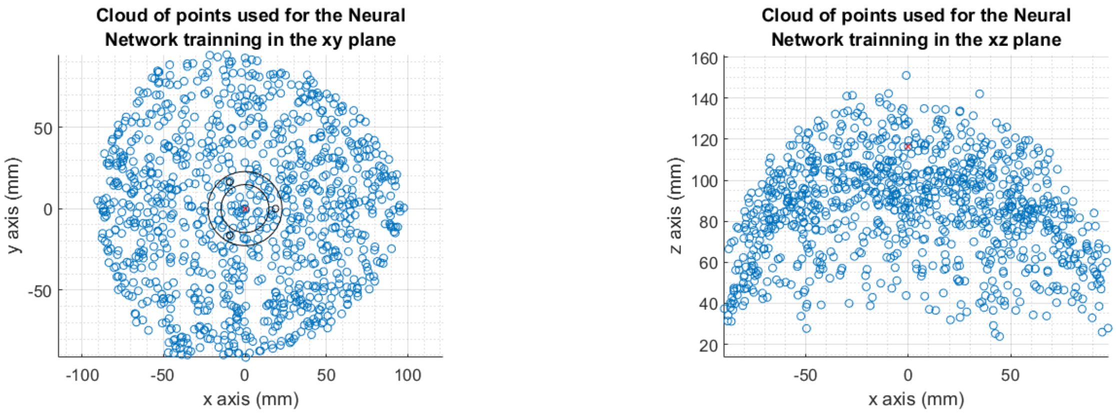


Figure 5.18: 3D cloud of the points collected



(a) Cloud of the points collected in the xy plane

(b) Cloud of the points collected in the xz plane

Figure 5.19: Cloud of the points collected in the plane xy and xz plane

5.7 Data Processing for the trajectories

Before the analysis of the trajectories obtained it is necessary to filter some undesired data. When recording the trajectories, part of the data acquired was obtained when the manipulator was not in movement, and if this data is not removed it will increase the weight of a single position on the mean error.

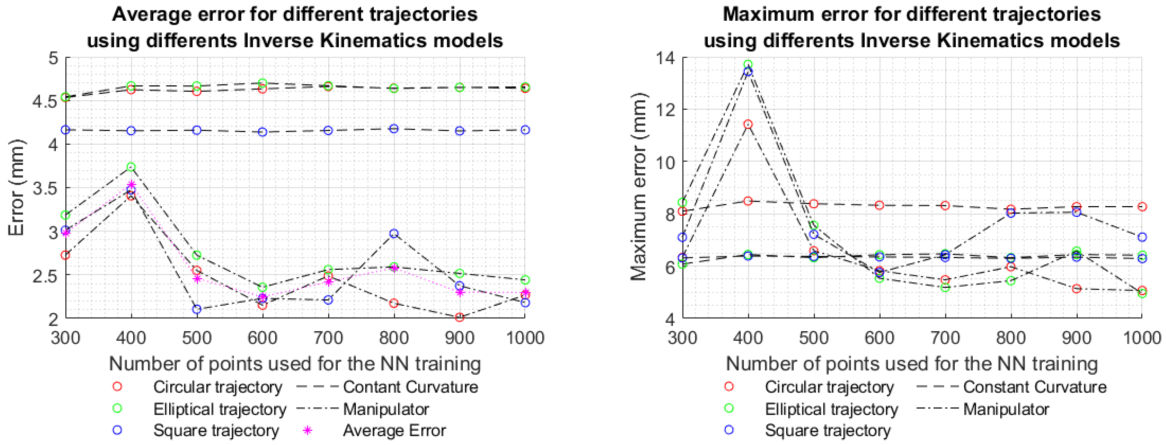
Anticipating the next sections, the trajectories errors for each path described by the manipulator will be calculated. However, the acquired data will not match with the inputs sent to the manipulator, therefore, it is important to have a solution. Without the direct correlation between the inputs, sent to the manipulator, and the outputs, acquired by the software, the error was assumed to be the norm distance between the coordinates obtained and the closest coordinate point of the desired trajectory.

5.8 Trajectories

With the training of several neural networks, in 4.2, with the previously collected data, the models will now be tested with different trajectories. The objective of testing the trajectories with various neural networks is to analyse its error evolution according to the number of data points used to train and reach a conclusion about the minimum number of data necessary to train the neural networks. In the paper [40] the number of points used were 1000 while for the theoretical model, trained previously, the minimum was 300. To have a comparison term with the NNs trained with the collected data, the NNs trained with the theoretical data were also put to the test. For every model created it was applied the same three trajectories of figure 4.3.

From the conclusions reached in 4.2.1 networks with less than 300 training data points do not give a good approximation of the theoretical constant curvature model. Considering this fact, only networks trained with a greater number of points were tested (300, 400, 500, 600, 700, 800, 900 and 998 data points, 998 points because it was added the neutral position).

In figures 5.20a) and 5.20b) it is presented the error evolution according to the number of data points used to train both types of learning models for the different trajectories.



(a) Mean error (b) Maximum error

Figure 5.20: Mean and Maximum error for the 3 trajectories using several Neural Networks trained with a different amount of data points with the same number of neurons

From figures 5.20a) and 5.20b) it is observed that the average and maximum error maintain a constant value for the models trained with simulated data. This performance can be explained due to the fact that these models only need 300 data points to, in a simulated world, create an almost perfect model of it. This means that models trained with an equal or greater number of points will result in a very similar or even identical models. Figure 4.6 shows the evolution of the average and maximum errors with negligible values in the simulated world.

On the other hand, when analysing the error evolution for the model trained with the collected data from the manipulator, figure 5.20a) shows that from the model trained with 500 points onwards, the mean error of the three trajectories is maintained in the range of 2.24 to 2.58 mm. And from the model with 600 training data points onwards the trajectories errors range is decreased to approximately 2-2.5 mm from 2-2.7 mm, with an outlier on the square trajectory with a value close to 3 mm. The models trained with 300 and 400 data points have an error that fluctuates between 2.7 and 3.7 mm which is higher than the range of the remaining models. The first model does not present a good R, in the graphs of figure 4.5, as the remaining models, where it is stabilized at its maximum. Hence, it was expected that between these two groups there would be a difference in performance.

Considering this error evolution, it is reasonable to assume that the minimum number of required points to train the neural network is from 500/600 upwards in contrast with the 300 points required to train with the constant curvature model. It is natural that the neural networks require more data from the prototype since it is more favourable to the appearance of errors either from the manipulator itself and/or from the remaining hardware and collecting software

Figure 5.20b) plots the evolution of the maximum error for the different models. The error for the circular and elliptical trajectories appears to stabilize, in the range of 5 to 6 mm, from the model with 600 training data points onwards. For the square trajectory the maximum error is the smallest for that model, however it increases more than 2 mm in the next iterations. For this trajectory the error range is slightly higher from 5.75 to 8 mm, if the model with 400 training data points is not considered. From the analysis made to these two error evolutions, figures 5.20a) and 5.20b), one can consider the network that was created with 400 data points as an outlier, or without proper training, and/or that the model with 300 data points for training described better the manipulator than the expected.

It is now important to compare the values obtained from the models trained with theoretical values to the models trained with the collected data. The maximum errors are very similar to all the models, however the model trained with 600 points, from the prototype data, displays a slightly advantage. The maximum error for the three trajectories in this model are one of the smallest and are on the same range. Up to this model, the maximum error of the three trajectories, for the model learned from the manipulator, followed the same pattern. When one of them increased the rest also increased, when one decreased the rest would also decrease. From this network onward there is one trajectory that doesn't follow the trend. The analysis of the maximum error is not the most important one since this error can be a result of an outlier point that corrupts this graphic.

From our observation, of figure 5.20a), it was noticed that the mean error from the theoretical learned model is bigger for all trajectories than the error from the models created using the manipulator data. When observing the trajectories from the model learned from the simulated manipulator, figures C.10, C.17 and C.24, it is possible to notice that there is a constant difference of heights. This systematic difference may be one of the reasons for the errors observed. Nonetheless, when these trajectories are analysed in the xy plane one can observe that the trajectories are somewhat similar to the desired one but still have a systematic errors due to the nonlinearities of the prototype.

All of the trajectories obtained are presented in the appendix C.2 and the best trajectories are presented in the next section 5.8.1.

5.8.1 Best trajectories with 100 neurons

From our observations of the average and maximum error graphics, figure 5.20a) and 5.20b), it is possible to choose the model with 600 training data points as approximately the best model. Such model has, when all trajectories are taken into account, a smaller average of the total error, table 5.8 and figure 5.20a), and a better approximation to the desired trajectory on the xy plane. This model is also a good choice when observing the maximum error, it has the smallest error for the square trajectory and for the remaining trajectories is in the same range for the best results. In this section it is displayed the trajectories of the selected model, the remaining trajectories can be seen in the appendix C.2. Nevertheless, the remaining trajectories, of the models that are not selected, also gave satisfactory results and almost every other could be chosen. But the one that is perceived to better suit all trajectories is the model with 600 number of data points, figures in 5.21, even though it is not the best for every trajectory.

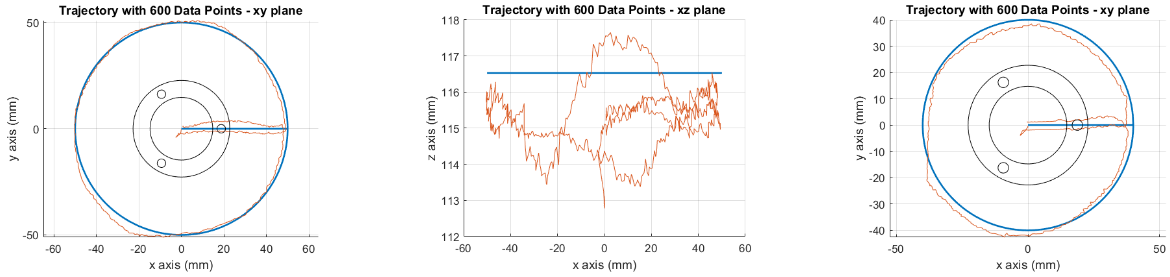
From the previous tables 5.7 and 5.8, it is noticed that the average error of the models, with similar R, have the same order of magnitude of the mean and median values of the repeatability radius, respectively, 2.98 and 2.69 mm. The median value was also chosen due to its capacity to overlook the presence of outliers on the measurements. In conclusion, it is understood that the neural networks were

Table 5.8: Average Error from all trajectories for the models trained with the experimental data

Number of points used for training	Average error for all trajectories
300	2.9744mm
400	3.5393mm
500	2.4611mm
600	2.2468mm
700	2.4199mm
800	2.5797mm
900	2.3019mm
1000	2.2977mm

trained correctly as the analysis of the error shows a similar value between tables 5.7 and 5.8.

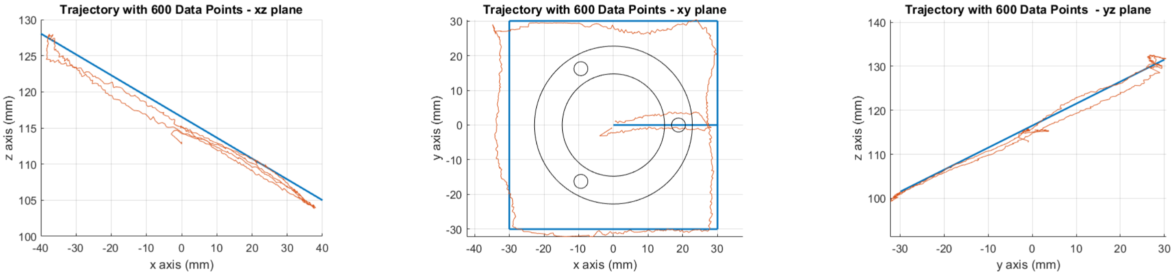
If the repeatability radius was to be calculated differently, for example as the radius between the average coordinate and the further point the results would lead to a lower and closer value to the average error of trajectories.



(a) Circular Trajectory from the model learned with 600 training points - xy plane

(b) Circular Trajectory from the model learned with 600 training points - xz plane

(c) Elliptical Trajectory from the model learned with 600 training points - xy plane



(d) Elliptical Trajectory from the model learned with 600 training points - xz plane

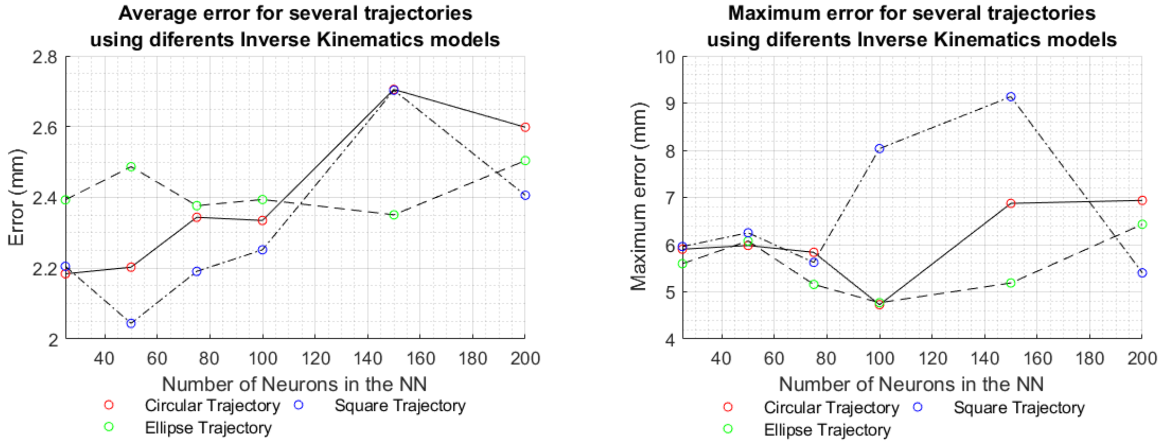
(e) Square Trajectory from the model learned with 600 training points - xy plane

(f) Square Trajectory from the model learned with 600 training points - yz plane

Figure 5.21: Circular, Elliptical and Square Trajectory from the model learned with 600 training points

5.8.2 Trajectories with variable number of neurons

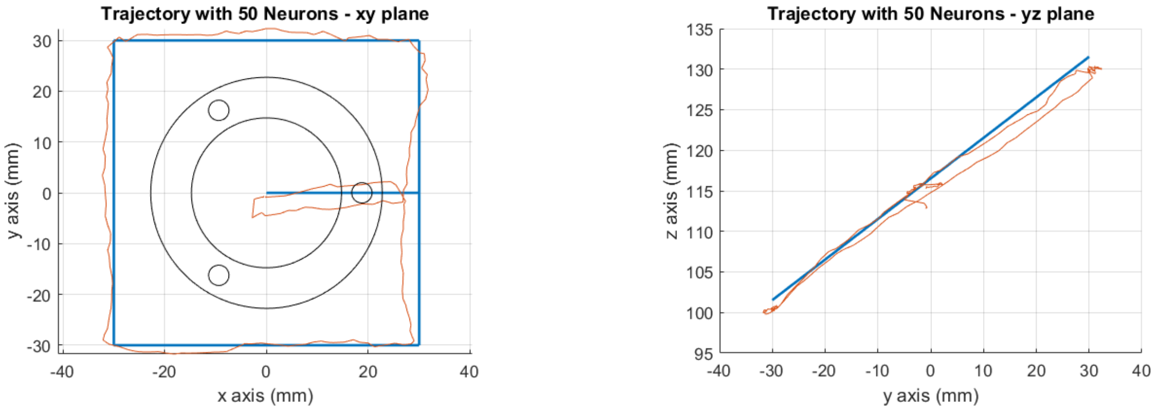
Up until now the neural network architecture was fixed as the results from the theoretical analysis showed a good performance, chapter 4.2. However, in an attempt to improve the results obtained so far in the experimental part, the architecture was modified. For the new architectures the only parameter modified was the number of neurons on its hidden layer, to 25, 50, 75, 100, 150 and 200 neurons. The training method was maintained from the previous neural networks and it used the totality of the collected points (998).



(a) Mean error

(b) Maximum error

Figure 5.22: Mean and Maximum error for the 3 trajectories using several Neural Networks trained with a different amount of neurons with a 1000 data points for training



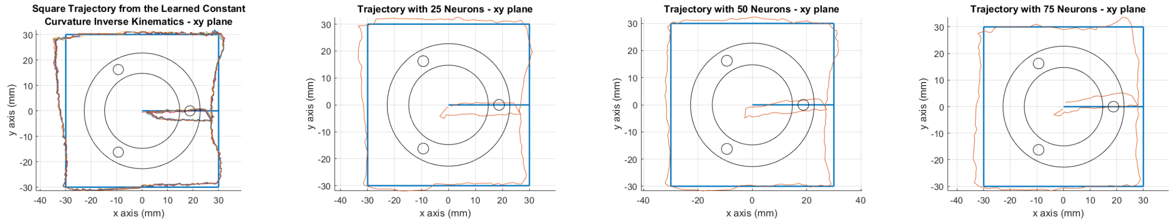
(a) Square Trajectory from the model learned with 50 Neurons - xy plane

(b) Square Trajectory from the model learned with 50 Neurons - yz plane

Figure 5.23: Square Trajectory from the model learned with 50 Neurons

When analysing the trajectories described by the manipulator, the models with more neurons do not increase its accuracy instead it seems to overfit the model. This behaviour is more visible on the left side of the square trajectory from the models created with 150 and 200 neurons, figures 5.24 f) and g).

A more deeper analysis of this edge in all the models results in a perceived evolution regarding its alignment to the desired trajectory. In the trajectory produced by the theoretical model it is noted that on the left upper half side of the manipulator path it exhibits a systematic positional error, figure 5.24a). With the trained models it is expected that this feature is modeled and opposed, in fact, the paths recorded show a compensation rate depending on the number of neurons used, figure 5.24 and in C.3.3. It is possible to state that as the number of neurons used for training is increased the compensation of this feature is also increased. The tipping point between a model being undercompensated to overcompen-

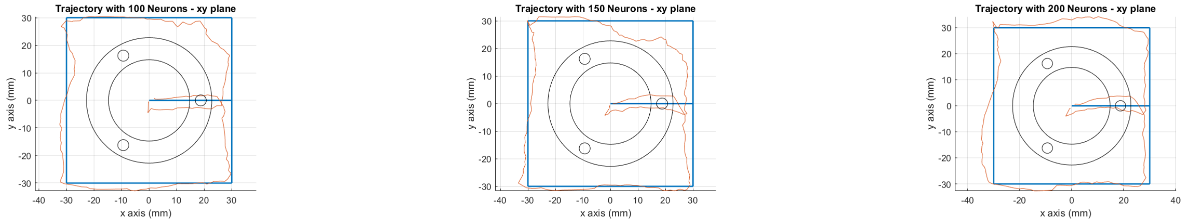


(a) Square Trajectory from the model learned with theoretical data - xy plane

(b) Square Trajectory from the model learned with 25 Neurons - xy plane

(c) Square Trajectory from the model learned with 50 Neurons - xy plane

(d) Square Trajectory from the model learned with 75 Neurons - xy plane



(e) Square Trajectory from the model learned with 100 Neurons - xy plane

(f) Square Trajectory from the model learned with 150 Neurons - xy plane

(g) Square Trajectory from the model learned with 200 Neurons - xy plane

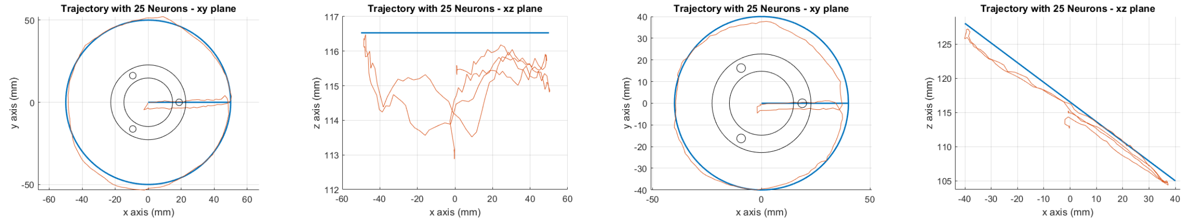
Figure 5.24: Square Trajectory from the model learned with variable number of neurons - xy plane

sated is with the model trained with 50 neurons, in which the manipulator produces a straighter line. The models with 25 and 75 neurons, respectively, undercompensate and overcompensate this systematic error. Another setback from the models trained with a higher number of neurons is the decreasing definition of the top right corner in the trajectory.

Nonetheless, not every aspect of the trajectory becomes worst when the number of neurons increase, an example of an improvement is in the right edge of the square trajectory. From the theoretical models the path followed by the manipulator creates a concave shape in this area, C.24. In the same way as for the left edge of the square, the increasing the number of neurons used for training tends to increment the compensation from the model. Therefore, the results obtained are somehow expected. The models with less than 50 neurons are slightly undercompensated and from this model onwards the trajectories become more straight and accurate. In order to choose a model that better fits this trajectory a middle ground should be accomplished between the several features of the manipulator. For that, the average error and the overall look of the path were taken into account. The selected model was trained with 50 neurons, figure 5.23, because not only it has the smallest average error but also because it fits better three of the four square edges than any other model. However, the right side is still partially undercompensated.

For the circular trajectory, as the number of neurons increases the average error also tends to increase. This can be verified when observing the path described by the manipulator on the xy plane since the trajectories tend to be worst as the number of neurons increase, C.3.1. In every path a systematic error can be observed in the third quadrant in the xy plane. Figure 5.25a) and b) shows the best fitted circular trajectory from a model created with a neural network with 25 neurons.

Lastly, the average error of the elliptical trajectory does not have a clear trend along the several networks, hence the analysis should be manly focused on its visual component. Nevertheless, the



(a) Circular Trajectory from the model learned with 25 Neurons - xy plane

(b) Circular Trajectory from the model learned with 25 Neurons - xz plane

(c) Elliptical Trajectory from the model learned with 25 Neurons - xy plane

(d) Elliptical Trajectory from the model learned with 25 Neurons - xz plane

Figure 5.25: Circular and Elliptical Trajectory from the model learned with 25 Neurons

trajectories with a lower and similar errors are the ones with 25, 75, 100 and 150 neurons. Following a visual examination the path that better fits the desired trajectory, in the xy plane, is the path resultant from the model trained with 25 neurons as seen in the figure 5.25c) and d). In the same way that the circular trajectories have a systematic error on the third quadrant, the elliptical trajectories also have the same systematic error. The trajectory from the model with 25 neurons follows more consistently the desired trajectory than the remaining models.

As seen in all trajectories, the neural networks with 150 and 200 neurons produce a more irregular path. It is possible that it results from an insufficient number of training points for such number of neurons.

The models trained with a fewer number of neurons appear to better model the prototype behaviour, even though the errors obtained are similar.

5.9 Material Response

In this final section it is examined the consequences, on the material, of the effort taken during the experimental work. During the work of the prototype it was observed that when the manipulator was fully compressed the actuators were in a constant effort to maintain the pose. This phenomenon can be explained by the opposition of the stranded steel wires to be bent into the enclosure of the actuators.

At the end of the experiments the prototype was also inspected in order to find anomalies. No fatal flaws were discovered, however some deformations were detected and should be taken into account if, in future works, it is desired to 3D print similar components for functional parts. The most important deformation was in the flexible module, figure A.1b), where there is a small layer separation. After the discovering of this separation between the layers it was also found other small deformations created during the 3D printing. A common point between these aspects is that all of them are at the same height as the printing support, figure 3.18. Therefore, it is possible to attribute this deformations to the pausing made while printing the module to add the lateral supports as it results in a greater cooling of the deposited layers and probably resulting in a not so good layer adhesion. The printing pause may not be necessary if the model to print has a lower height, the printing speed is reduced or if it is possible to print with rigid support material on a 3D printer with two extruders.

Another element that is subject to considerable forces and that was analysed in figure 3.16 are the clamps that attach the top piece to the flexible module, figure 3.13 c) and d). After the conclusion of the experimental tests, it is possible to visualize some deformation, figure A.1a), in these pieces. Although these deformations are small an analysis to alternative materials to *PLA* should be made.

Even though these not fatal flaws they should be analysed and improved in further developments both the design and the material used.

Chapter 6

Conclusions

The objective of the present thesis was to design, build and experiment on a continuous deformation soft robot module capable to sustain an acceptable level of torsional load, to allow for off-axis manipulation. This objective was successfully accomplished in all stages defined at the beginning of this document, by the use and exploration of several methods, materials and technologies. A new soft manipulator was then created from scratch by the use of several original ideas, first by the design and optimization of a theoretical model to cope with the challenges identified, and later by an experimental part where the mentioned ideas were translated into reality by the construction of a real and operational manipulator. Therefore the present work was divided in two major parts: a design one and an experimental section.

On the first part, it was investigated with a finite element analysis the design of a manipulator capable to sustain torsional forces while maintaining its ability to compress and extend. This analysis was conducted for three different models with a variable number of waves per level, 3, 6 and 9 waves. The first model shows a small extending and compressing stiffness crucial to support its own weight as the wave angle with the base plane is small. In contrast, the third model despite having the larger torsional stiffness has an extending and bending stiffness too great as a result of a more vertical wave. The model with 6 waves was the model that had the best relation between the torsional and the extending/bending stiffness. Its torsional stiffness is only 1.5 times smaller than the larger torsional stiffness for an extending and bending stiffness lower 6.8 times, the torsional stiffness is also 4.6 times greater than the stiffness of the model with 3 waves. A modal analysis of the model with 6 waves confirms the increased torsional stiffness as it has a greater natural frequency than the remaining studied modes. With the theoretical design defined, with 6 waves per level, it was possible to develop the remaining components to form a functional prototype. The majority of the elements that compose the manipulator were produced using additive manufacturing more commonly known process as 3D printing in order to take fully advantage of its capabilities.

Later, on the second part, with the manipulator fully assembled and operational it was possible to make a more detailed analysis to its behaviour with and without loads, such as tests to analyse its response to loading, experiments to analyse the repeatability of the manipulator, and a final experiment to test several kinematic models created with neural networks. The experimental work volume obtained allowed to validate the theoretical model. In these plots it is noted that the biggest errors occur in the lowest positions where the modulation errors are more pronounced, a result of a greater curvature. The work volume will vary according to the direction considered as the positioning of the actuators is not symmetric in every plane. This can be better visualized in the cloud of collected points for the training of the neural networks.

The loading tests provided an assurance that the module is capable to work with off-axis motions. The static experiment evaluated the displacement of the manipulator tip when loaded with 4 different weights (0g, 50g, 100g and 150g) in 4 different poses, 3 in on-axis positions and 1 in an off-axis position. From the results obtained, the off-axis pose showed to have a greater vertical stiffness than the on-axis poses with the exception of the fully compressed pose. This last pose has the smallest bending arm and behaves as a very dense module. The torsional stiffness of the bent position is more than two times greater than for the on-axis pose with approximately the same neutral line length (the on-axis neutral line length is 8.1 mm smaller). The dynamic loading experiment, aims to verify the results obtained previously for an off-axis manipulation. The same weights were applied and the prototype followed two different trajectories for each weight, a circular and an elliptical. In both trajectories the manipulator described a consistent and reliable motion in off-axis situations, without any critical deformations. In the experimental work it was noted that even though the torsional stiffness was increased it still exhibits some passive compliance in directions that the geometry cannot actuate as it depends on the extending and compressing stiffness.

To better analyse the behaviour of the manipulator three experiments were developed to evaluate its repeatability. The first two tests followed a determined order while the third a random order. In the first two tests the mean repeatability radius was very similar, 0.35 mm and 0.53 mm. It is possible to observe in the plots the presence of hysteresis on the neutral position due to the formation of cluster as a function of the position reached previously. On the third test, with a random order, the mean repeatability radius increased substantially to 2.98 mm, almost 6 times the previous values. Such increment may be a result of the hysteresis present in the assembly as it depends on the originated position of the motion and the order followed.

In the last experimental work several inverse kinematic models were used to compute the required inputs of three different trajectories, one with constant height and the remaining two with variable height. The mean and maximum error of the obtained trajectories were also calculated. The inverse kinematic models were obtained using neural networks trained with data acquired from the theoretical model and from the assembled prototype. The trajectories obtained from the first showed a constant error and the nonlinearities of the system. From the second the trajectories have a smaller mean error and a better modulation of the nonlinearities. From these, and from its training, it was possible to evaluate the error evolution according to the number of training points of the neural network. 500 data points is the minimum required number to successfully train a neural network with 100 neurons. The obtained mean error for this model and for the remaining models with a higher number of points has the same order of magnitudes as the repeatability test with random order. It is then understood that the neural networks were correctly trained since the data used for its training was randomly selected.

With the number of points studied and with some acquired trajectories, it was then studied the impact that the number of neurons have on the modulation of the manipulator. The number of neurons used varied from 25 to 200. From the trajectories described by each model it was possible to observe an evolution of the compensation of the nonlinearities, better seen on the square trajectory. The networks with a lower number of neurons appear to undercompensate the nonlinearities present, observed in the trajectories created with the theoretical model. While the networks with a higher number of neurons seem to overcompensate it. The mean error of the several trajectories also show a small increasing trend as the number of neurons increases. From this, the best square trajectory obtained was from the model trained with 50 neurons. For the remaining trajectories the decision is not as clear, however the neural networks with less neurons perform slightly better as the neural networks with higher number of neurons seem to overfit the training data.

6.1 Achievements

The main accomplishment of this work was the design and construction of a fully functional soft continuum manipulator module capable to sustain torsional loads. Several other findings can be listed next:

1. Evaluation of the optimal number of waves for the current wave spring geometry;
2. The design of a lightweight, with hollow center, soft continuous manipulator;
3. A simple actuation method with few moving parts;
4. The combination of a 3D printing production method, using a flexible filament, with the necessity of quickly prototyping a soft continuous deformation robot;
5. A manipulator capable to sustain, without significant deformation, an off-axis weight with 4,16 times the weight of the module in static and dynamic scenarios;
6. The characterization of the manipulator work space and repeatability with a mean radius of 2.98 mm;
7. The kinematic modelling of the prototype using neural networks, that include a study of the number of training points required to effectively model it (500) and the impact of the number of neurons used (from 25 to 200, with the first ones obtaining the best results);
8. Article Accepted for publication at an International Conference – ICARA 2021, [20].

This geometry is a step further for the soft robots to be commonly used in the more diverse areas. 3D printing can be extremely helpful in future developments of soft robots, however it may not be the answer if the manipulator will be subjected to extreme stress scenarios where the material can malfunction due to the method of printing by layers, or if the aim is to mass produce. Regarding neural networks and artificial intelligence, they are a valuable tool to model behaviours that may be too complex for detailed modelling by other means and were an important asset for the prototype here created.

6.2 Future Work

This work might be a starting point for future works that can follow multiple directions according to the selected project goals.

1. A first area that can be improved is the characterization of the materials used and a research into alternative materials to these. This study can also help choosing a material that minimizes the hysteresis of the flexible module. Next, it can be examined the resultant properties of the 3D printed elements;
2. Another possibility for improvement is the design of the actuation mechanism in order to reduce the overall volume of the robot module. For this, linear actuators, or miniature actuators embedded in the structure of the link itself can be investigated;
3. Another area to be studied is the addition of proprioceptive sensors to the module to have a better perception of the robot curvature. This inclusion will allow a better control, accuracy and also to reduce the dependency of external sensors like the Polaris system used in this work;
4. Following the development of the single module, a full robot structure comprised of several modules can be assembled and studied, in order to develop both low level, and also high level behavioural control algorithms.

Bibliography

[1] T. Ranzani, G. Gerboni, M. Cianchetti, and A. Menciassi. A bioinspired soft manipulator for minimally invasive surgery. *Bioinspiration & Biomimetics*, 10(3):035008, may 2015.

[2] M. Manti, A. Pratesi, E. Falocito, M. Cianchetti, and C. Laschi. Soft assistive robot for personal care of eldery people. *IEEE RAS/EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 833–838, June 2016.

[3] T. Branch, S. Stinton, M. Sterberg, W. Hutton, F. Lavoie, C. Guier, and P. Neyret. Robotic axial lower leg testing:repeatability and reproducibility. *Knee Surg Sports Traumatol Arthrosc*, 23:2892–2899, 2015.

[4] Y.-J. Kim, S. Cheng, S. Kim, and K. Iagnemma. A stiffness-adjustable hyperredundant manipulator using a variable neutral-line mechanism for minimally invasive surgery. *IEEE Transactions on Robotics*, 30:382–395, 2014.

[5] S. Kolachalama and S. Lakshmanan. Continum robots for manipulation applications: A survey. *Jornal of Robotics*, page 19, jul 2020. https://doi.org/10.1155/2020/4187048.

[6] W. McMahan, V. Chitrakaran, M. Csencsits, D. Dawson, I. Walker, B.A.Jones, M. Pritts, D. Dienno, M. Grissom, and C. Rahn. Field trials and testing of the octarm continuum manipulator. *IEEE International Conference on Robotics and Automation*, 2006.

[7] D. Trivedi, C. D. Rahn, W. M. Kier, and I. D. Walker. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 5(3):99–117, Sept. 2008.

[8] R. J. Webster and B. A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010.

[9] G. Robinson and J. B. C. Davies. Continuum robots - a state of the art. *IEEE International Conference on Robotics and Automation*, pages 2849–2854, 1999.

[10] F. Renda, M. Giorelli, M. Calisti, M. Cianchetti, and C. Laschi. Dynamic model of a multibending soft robot arm driven by cables. *IEEE TRANSACTIONS ON ROBOTICS*, 30(5):1109–1122, 2014.

[11] A. k. Goel. Biologically inspired design: A new program for computational sustainability. *IEEE Intelligent Systems*, pages 80–84, 2013.

[12] G. Boccolato, F. Manta, S. Dumitru, and D. Cojocaru. 3d kinematics of a tentacle robot. *International Journal of Systems Applications, Engineering and Development*, 4:1–8, 2010.

[13] J.-Y. Lee, E.-Y. Go, W.-Y. Choi, W.-B. Kim, and K.-J. Cho. Development of soft continuum manipulator with pneumatic and tendon driven actuations. *13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 377–379, Aug. 2016.

[14] H. University. Soft robotics. https://gmwgroup.harvard.edu/soft-robotics, -. (accessed: 10.12.2020).

[15] J. Santoso, E. H. Skorina, M. Luo, R. Yan, and C. D. Onal. Design and analysis of an origami continuum manipulation module with torsional strength. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2098–2104, 2017.

[16] I. D. Walker. Continuous backbone continuum robot manipulators. *International Scholarly Research Notices*, 2013.

[17] J. Burgner-Kahrs, D. C. Rucker, and H. Choset. Continuum robots for medical applications: A survey. *IEEE TRANSACTIONS ON ROBOTICS*, 31(6):1261–1280, 2015.

[18] E. H. Skorina and C. D. Onal. Soft hybrid wave spring actuators. *Advanced Intelligent Systems*, 1900097(2):1–8, Jan. 2020.

[19] T. Spring. Multi turn wave springs with shim ends. https://www.da-yi.com.tw/en/product/Multi-Turn-Wave-Springs-with-Shim-Ends/multi-turn-wave-springs-with-shim-ends.html, -. (accessed: 31.12.2020).

[20] J. M. Salgueiro and J. C. P. Reis. Towards a highly integrated 3d printed soft continuum manipulator. *ICARA - International Conference on Automation, Robotics and Applications*, Accepted for publication.

[21] M. Blessing and I. D. Walker. Novel continuum robots with variable-length sections. *Proceedings 3rd IFAC Symposium on Mechatronic Systems, Sydney, Australia*, pages 55–60, Sept. 2004.

[22] R. T. L. Ferreira, I. C. Amatte, T. A. Dutra, and D. Burger. Experimental characterization and micrography of 3d printed pla and pla reinforced with short cabon fibers. Sept. 2017.

[23] H. K. Yap, H. Y. Ng, and C.-H. Yeow. High-force soft printable pneumatics for soft robotic applications. *SoRo - Soft Robotics*, 3(3):144–158, 2016.

[24] toppr. Hooke's law and stress-strain curve. https://www.toppr.com/guides/physics/mechanical-properties-of-solids/hookes-law-and-stress-strain-curve/, -. (accessed: 10.12.2020).

[25] T. weekly Pellet. Tensile strength: Yield or break. https://weeklypellet.com/2017/04/25/tensile-stregth-yield-or-break/, 2017. (accessed: 10.12.2020).

[26] C. M. Harris and A. G. Piersol. *Harris' Shock and Vibration Handbook*. McGRAW-HILL, 5 edition, 2001.

[27] P. P. Database. Stress-strain behavior of polymers. https://polymerdatabase.com/polymer%20physics/Stress-Strain%20Behavior.html, -. (accessed: 10.12.2020).

[28] B. Siciliano, L. Sciavico, L. Villani, and G. Oriolo. *Robotics Modelling, Planning and Control*. Springer.

[29] M. Rolf and J. J. Steil. Constant curvature continuum kinematics as fast approximate model for the bionic handling assistant. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3440–3447, oct 2012.

[30] D. Cojocaru, S. Dumitru, F. Manta, and G. Boccolato. Experiments with a continuum robot structure. *ICINCO*, (2), 2010.

[31] G. Boccolato, I. Dinulescu, A. Predescu, and D. Cojocaru. Simulation of a hyper redundant arm. *University of Craiova*, 2, 2010. http://ace.ucv.ro/analele/2010_vol2/01GiusepeBoccolato.pdf.

[32] IBM. Machine learning. https://www.ibm.com/cloud/learn/machine-learning, 2020. (accessed: 10.12.2020).

[33] T. B. Talk. Neuron. https://teenbraintalk.wordpress.com/neurons/, -. (accessed: 10.12.2020).

[34] missinglink.ai. 7 types of neural network activation functions: How to choose? https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/, -. (accessed: 10.12.2020).

[35] C. Ye, Y. Yang, C. Fermuller, and Y. Aloimonos. On the importance of consistency in training deep neural networks. 2017.

[36] BQ. Technical specifications - pla 1.75 mm. https://cdn-3d.niceshops.com/upload/file/Technical_specifications _-_BQ_PLA.pdf. (accessed: 31.12.2020).

[37] Recreus. Technical specifications - filaflex 1.75 mm - the original elastic filament for 3d printing. https://cdn-3d.niceshops.com/upload/file/FILAFLEX_TECHNICAL_DATA_SHEET.pdf. (accessed: 31.12.2020).

[38] filament2print. Filaflex tpe 82a. https://filament2print.com/gb/flexible-tpe-tpu/1145-filaflex-tpe-82a.html, -. (accessed: 10.12.2020).

[39] Recreus. How to print with filaflex. https://recreus.com/en/how-to-print-with-filaflex/, -. (accessed: 10.12.2020).

[40] A.-V. Duka. Neural network based inverse kinematics solution for trajectory tracking of a robotic arm. *The 7th International Conference Interdisciplinarity in Engineering*, pages 20–27, 2013.

[41] M. PŁaczek and Łukasz Piszczek. Testing of an industrial robot's accuracy and repeatability in off and online environment. *Eksploatacja i Niezawodnosc - Maintenance and Reliability*, 20(3): 455–464, 2018. http://dx.doi.org/10.17531/ein.2018.3.15.

Appendix A

Material Response

(a) Clamp deformation after use

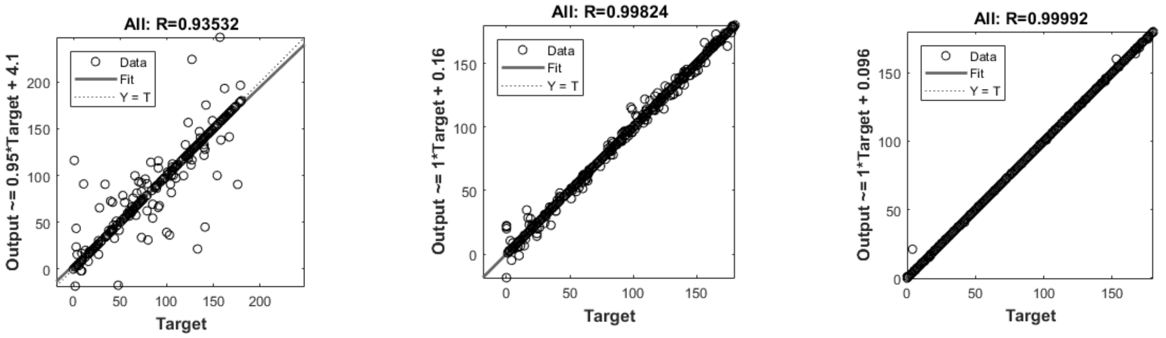
(b) Module layer separation

Figure A.1: Clamp deformation and layer separation

Appendix B

Neural Networks

Overall R evolution with the theoretical data up to its maximum.

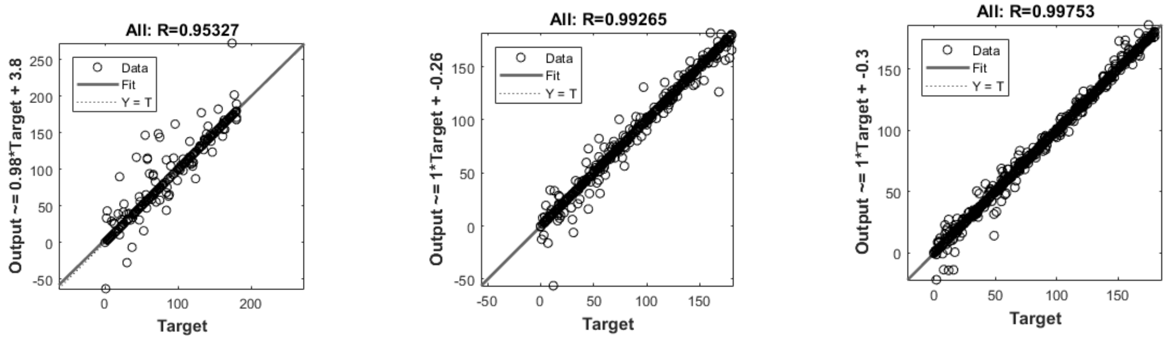


(a) With 100 training data points (b) With 200 training data points (c) With 300 training data points

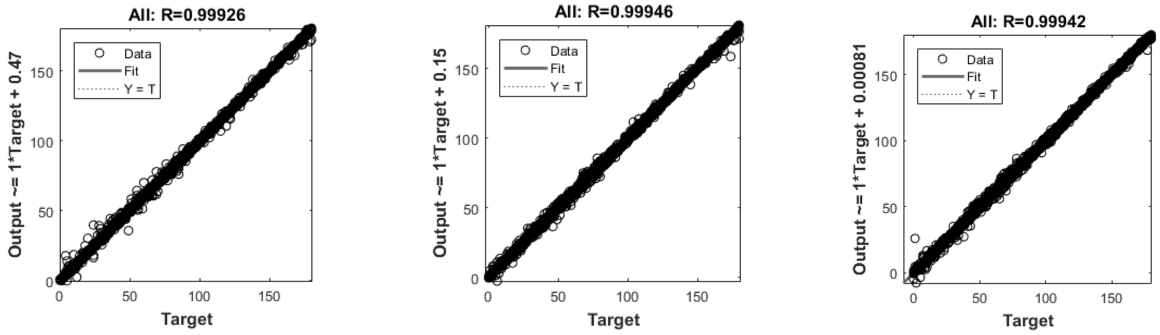
(d) With 400 training data points (e) With 500 training data points

Figure B.1: R evolution with the theoretical data

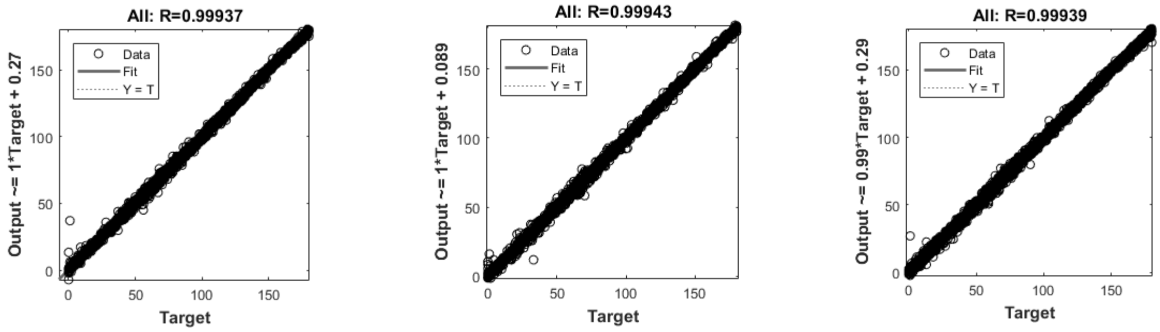
Overall R evolution with the data collected from the prototype and constant number of neurons, 100.



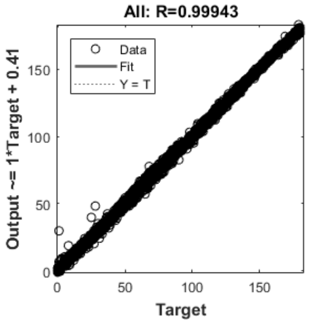
(a) With 100 training data points (b) With 200 training data points (c) With 300 training data points



(d) With 400 training data points (e) With 500 training data points (f) With 600 training data points



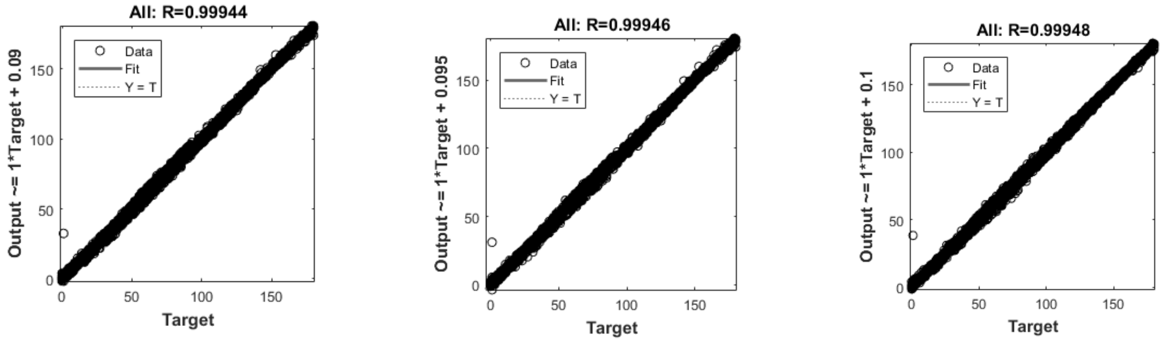
(g) With 700 training data points (h) With 800 training data points (i) With 900 training data points



(j) With 1000 training data points

Figure B.2: R evolution with the prototype data

Overall R evolution with the data collected from the prototype and a variable number of neurons.



(a) With 25 Neurons (b) With 50 Neurons (c) With 75 Neurons

(d) With 150 Neurons (e) With 200 Neurons

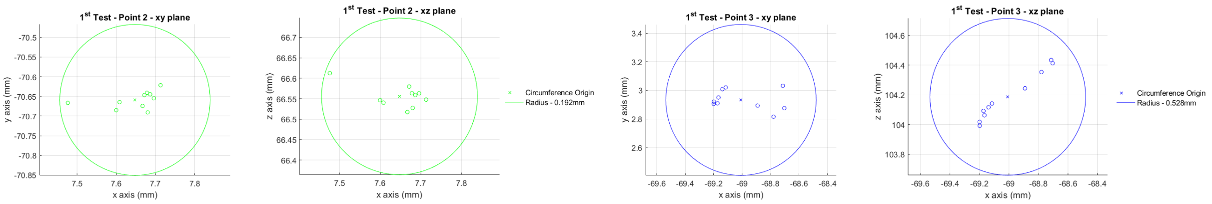
Figure B.3: R evolution with the prototype data and variable number of neurons

Appendix C

Results Appendixes

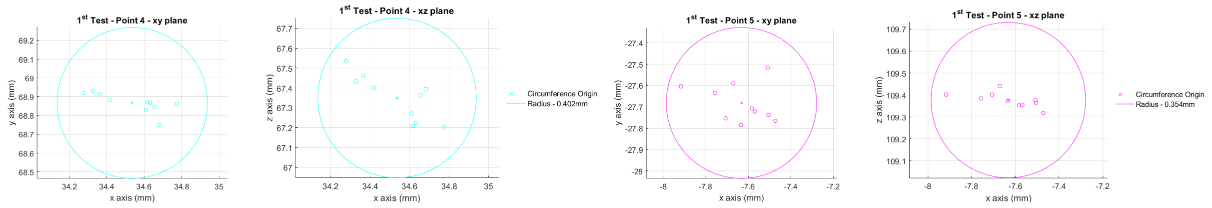
C.1 Repeatability Tests

C.1.1 Test 1 - Individual Points



(a) Point 2 in the xy plane (b) Point 2 in the xz plane (c) Point 3 in the xy plane (d) Point 3 in the xz plane

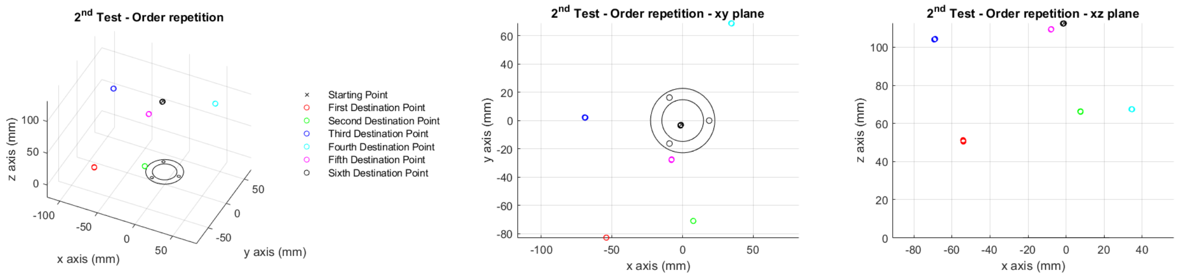
Figure C.1: Test 1 - Repeatability for Point 2 and 3



(a) Point 4 in the xy plane (b) Point 4 in the xz plane (c) Point 5 in the xy plane (d) Point 5 in the xz plane

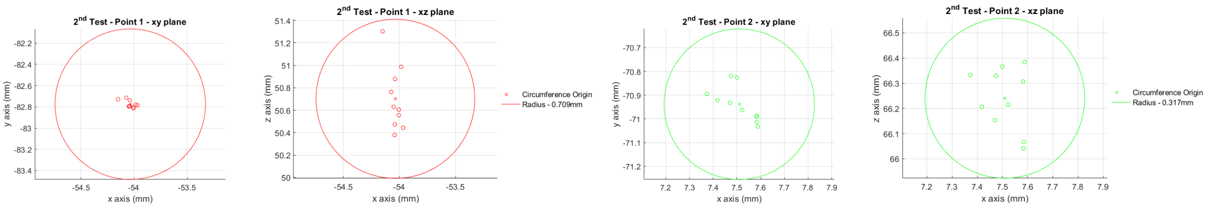
Figure C.2: Test 1 - Repeatability for Point 4 and 5

C.1.2 Test 2 - Individual Points



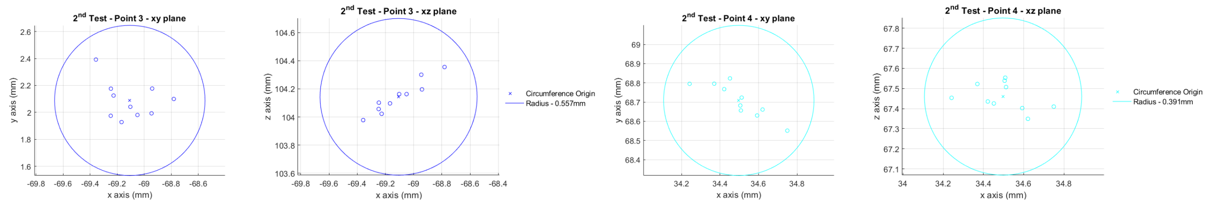
(a) Second Test - 3D space (b) Second Test - xy plane (c) Second Test - xz plane

Figure C.3: Repeatability Test - 2



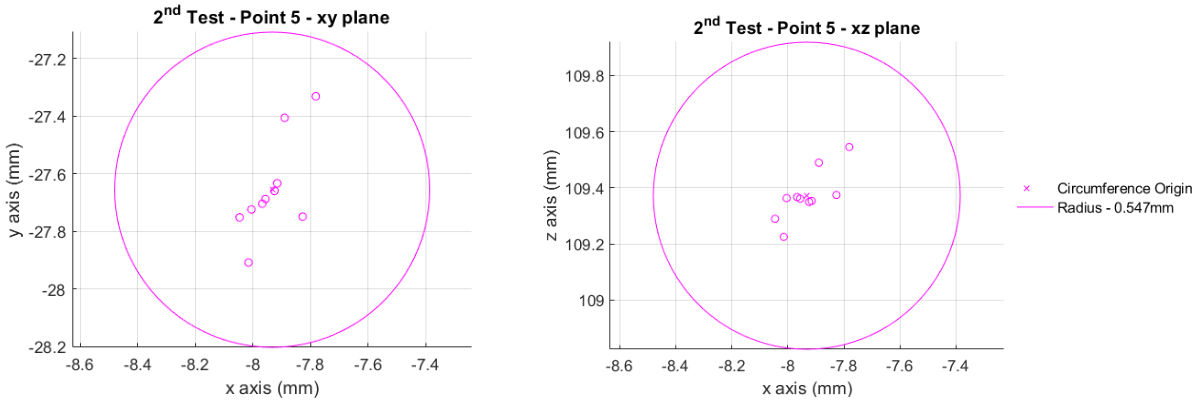
(a) Point 1 in the xy plane (b) Point 1 in the xz plane (c) Point 2 in the xy plane (d) Point 2 in the xz plane

Figure C.4: Test 2 - Repeatability for Point 1 and 2



(a) Second Test - Point 3 in the xy plane (b) Second Test - Point 3 in the xz plane (c) Second Test - Point 4 in the xy plane (d) Second Test - Point 4 in the xz plane

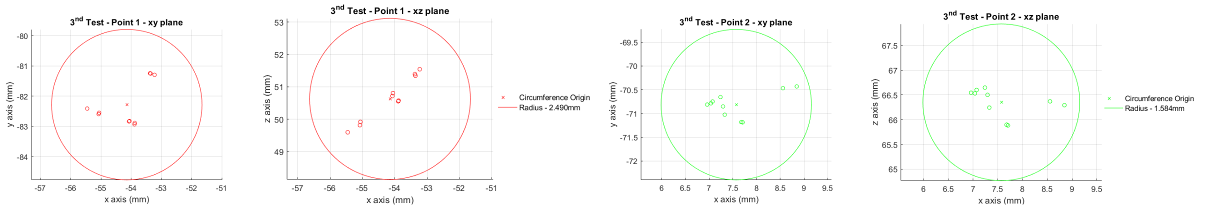
Figure C.5: Test 2 - Repeatability for Point 3 and 4



(a) Second Test - Point 5 in the xy plane (b) Second Test - Point 5 in the xz plane

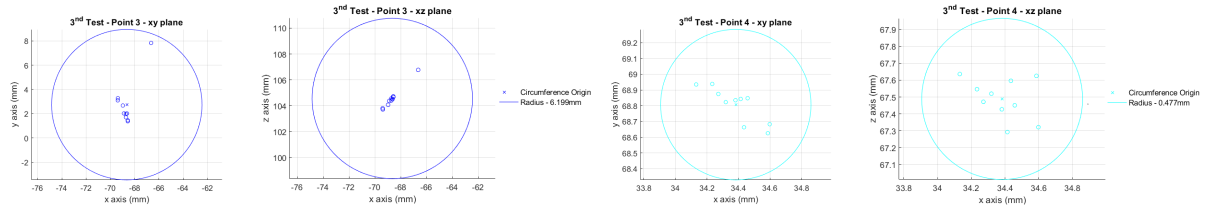
Figure C.6: Test 2 - Repeatability for Point 5 from Point 4

C.1.3 Test 3 - Individual Points



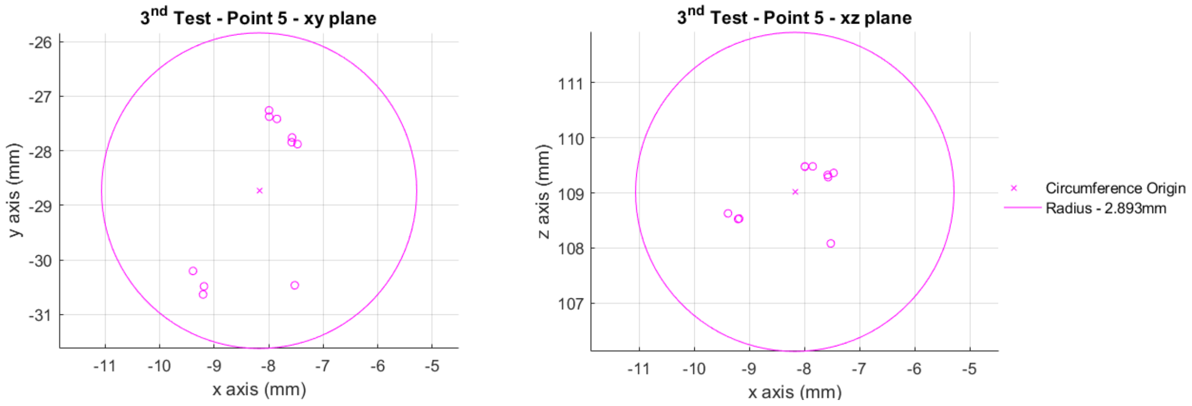
(a) Point 1 in the xy plane (b) Point 1 in the xz plane (c) Point 2 in the xy plane (d) Point 2 in the xz plane

Figure C.7: Test 3 - Repeatability for Point 1 and 2



(a) Point 3 in the xy plane (b) Point 3 in the xz plane (c) Point 4 in the xy plane (d) Point 4 in the xz plane

Figure C.8: Test 3 - Repeatability for Point 3 and 4



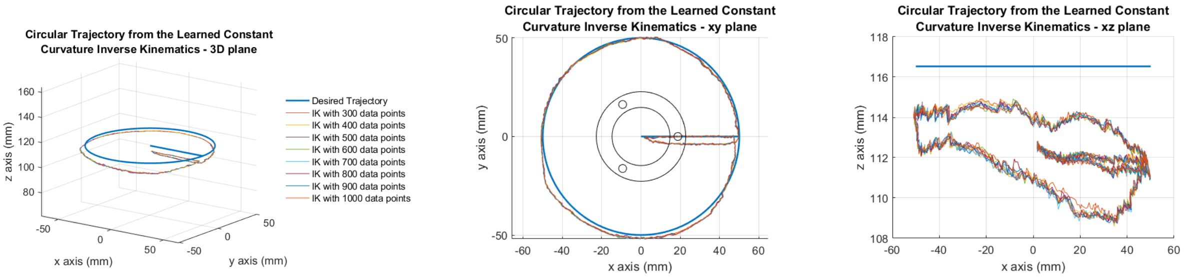
(a) Third Test - Point 5 in the xy plane (b) Third Test - Point 5 in the xz plane

Figure C.9: Test 3 - Repeatability for Point 5

C.2 Trajectories with 100 Neurons

C.2.1 Circular Trajectory

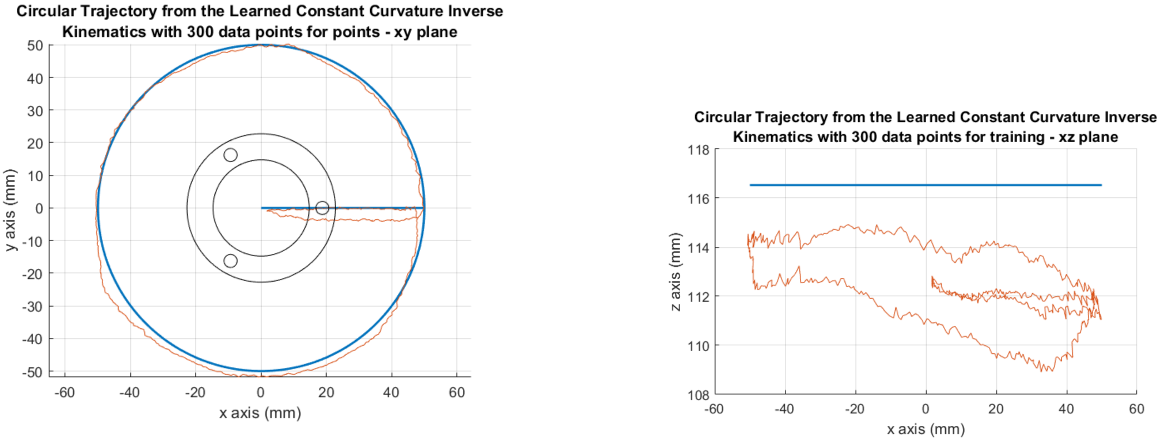
From the Constant Curvature Model



(a) Circular Trajectory - 3D space (b) Circular Trajectory - xy plane (c) Circular Trajectory - xz plane

Figure C.10: Circular Trajectory from the model learned from the constant curvature theory

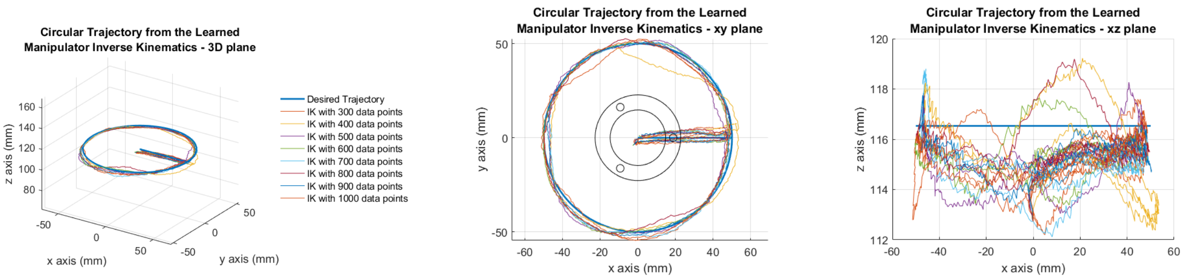
Since the trajectories, as seen in the previous figure, are very similar it is not necessary to show every individual trajectories instead it is going to be shown only one.



(a) Circular Trajectory - xy plane (b) Circular Trajectory - xz plane

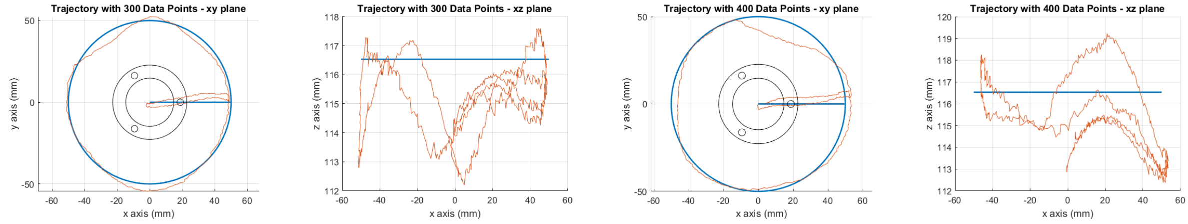
Figure C.11: Circular Trajectory from the model learned from the constant curvature theory with 300 training points

From the Manipulator Data



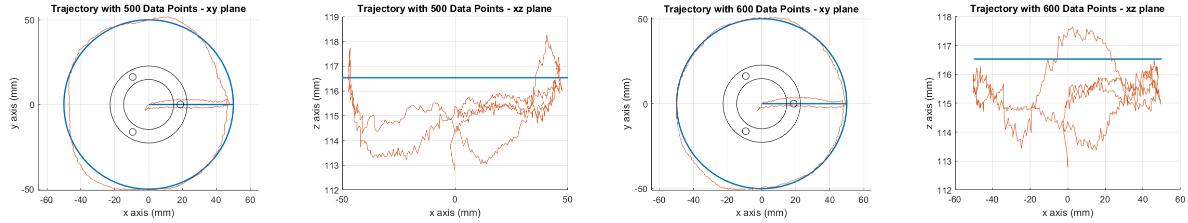
(a) Circular Trajectory- 3D space (b) Circular Trajectory - xy plane (c) Circular Trajectory - xz plane

Figure C.12: Circular Trajectory from the model learned from the Manipulator data



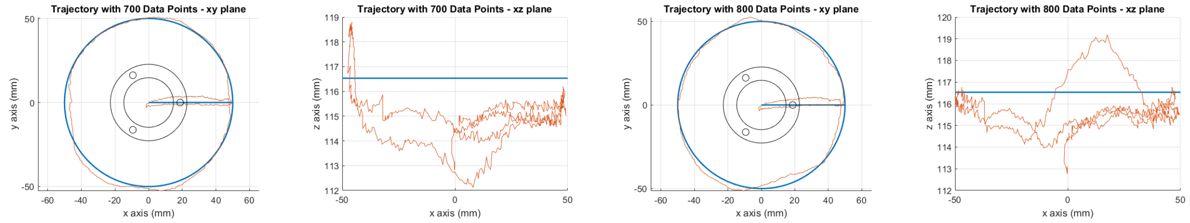
(a) Circular Trajectory from the model learned with 300 training points - xy plane (b) Circular Trajectory from the model learned with 300 training points - xz plane (c) Circular Trajectory from the model learned with 400 training points - xy plane (d) Circular Trajectory from the model learned with 400 training points - xz plane

Figure C.13: Circular Trajectory from the model learned with 300 and 400 training points



(a) Circular Trajectory from the model learned with 500 training points - xy plane (b) Circular Trajectory from the model learned with 500 training points - xz plane (c) Circular Trajectory from the model learned with 600 training points - xy plane (d) Circular Trajectory from the model learned with 600 training points - xz plane

Figure C.14: Circular Trajectory from the model learned with 500 and 600 training points



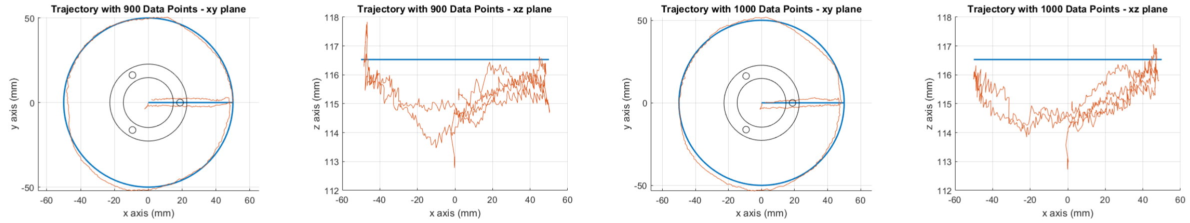
(a) Circular Trajectory from the model learned with 700 training points - xy plane

(b) Circular Trajectory from the model learned with 700 training points - xz plane

(c) Circular Trajectory from the model learned with 800 training points - xy plane

(d) Circular Trajectory from the model learned with 800 training points - xz plane

Figure C.15: Circular Trajectory from the model learned with 700 and 800 training points



(a) Circular Trajectory from the model learned with 900 training points - xy plane

(b) Circular Trajectory from the model learned with 900 training points - xz plane

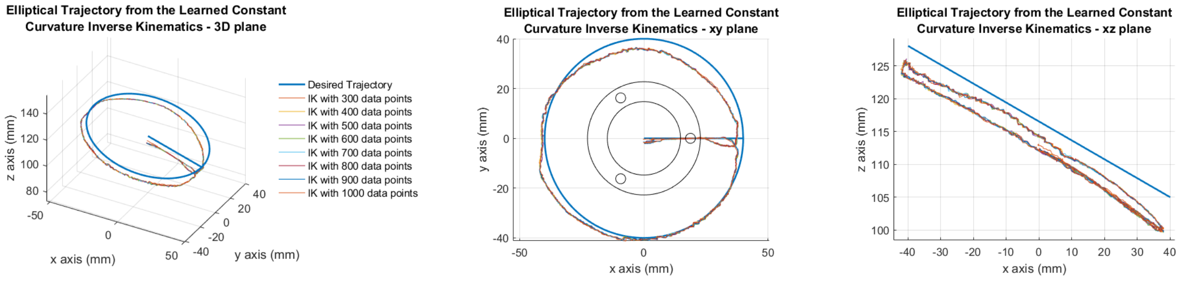
(c) Circular Trajectory from the model learned with 1000 training points - xy plane

(d) Circular Trajectory from the model learned with 1000 training points - xz plane

Figure C.16: Circular Trajectory from the model learned with 900 and 1000 training points

C.2.2 Elliptical Trajectory

From the Constant Curvature Model

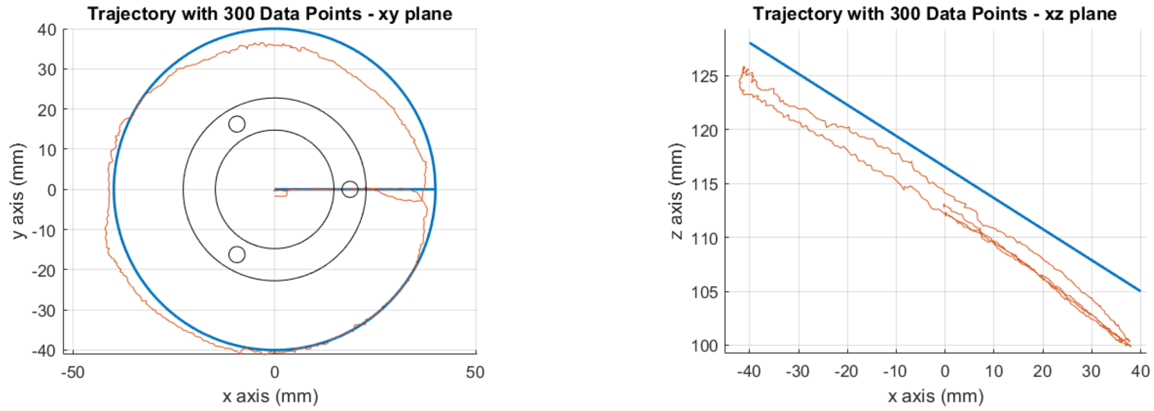


(a) Elliptical Trajectory - 3D space

(b) Elliptical Trajectory - xy plane

(c) Elliptical Trajectory - xz plane

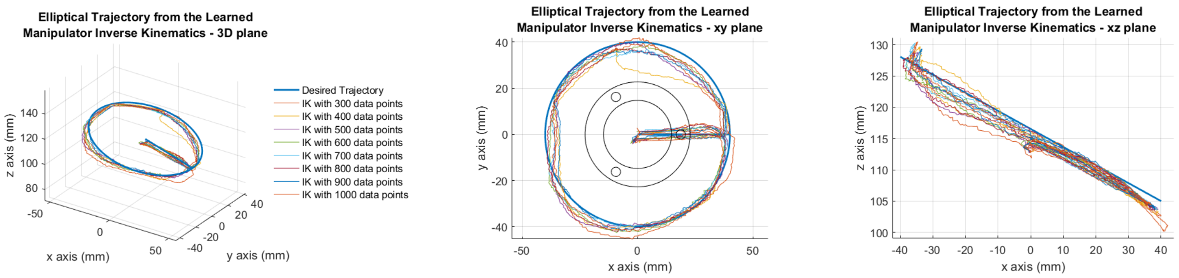
Figure C.17: Elliptical Trajectory from the model learned from the constant curvature theory



(a) Elliptical Trajectory - xy plane (b) Elliptical Trajectory - xz plane

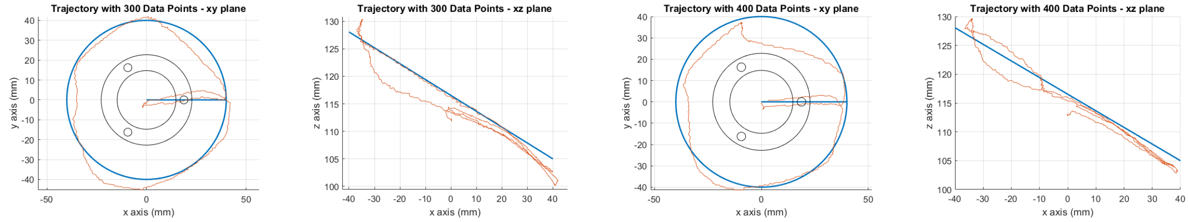
Figure C.18: Elliptical Trajectory from the model learned from the constant curvature theory with 300 training points

From the Manipulator Data



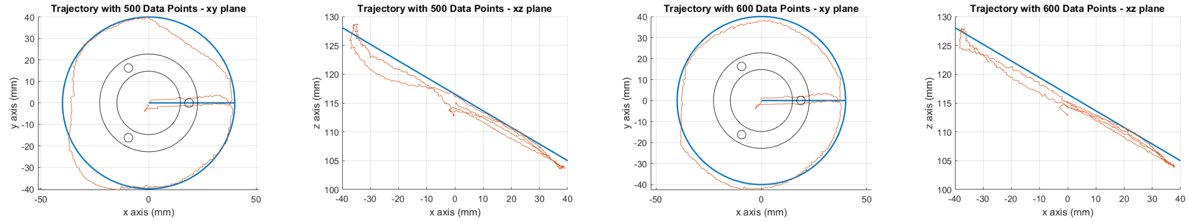
(a) Elliptical Trajectory- 3D space (b) Elliptical Trajectory - xy plane (c) Elliptical Trajectory - xz plane

Figure C.19: Elliptical Trajectory from the model learned from the Manipulator data



(a) Elliptical Trajectory from the model learned with 300 training points - xy plane (b) Elliptical Trajectory from the model learned with 300 training points - xz plane (c) Elliptical Trajectory from the model learned with 400 training points - xy plane (d) Elliptical Trajectory from the model learned with 400 training points - xz plane

Figure C.20: Elliptical Trajectory from the model learned with 300 and 400 training points



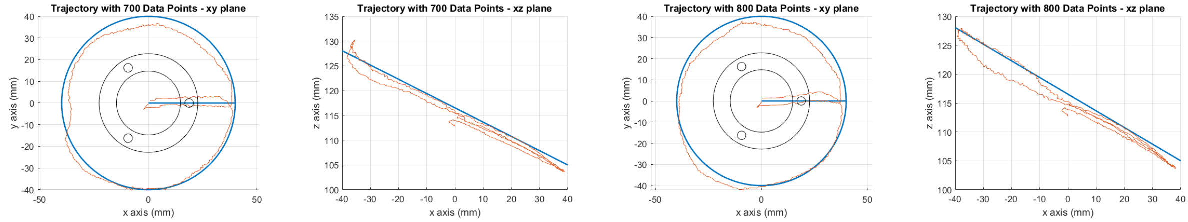
(a) Elliptical Trajectory from the model learned with 500 training points - xy plane

(b) Elliptical Trajectory from the model learned with 500 training points - xz plane

(c) Elliptical Trajectory from the model learned with 600 training points - xy plane

(d) Elliptical Trajectory from the model learned with 600 training points - xz plane

Figure C.21: Elliptical Trajectory from the model learned with 500 and 600 training points



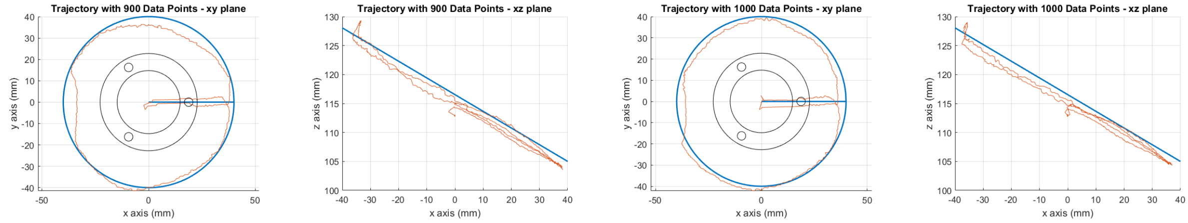
(a) Elliptical Trajectory from the model learned with 700 training points - xy plane

(b) Elliptical Trajectory from the model learned with 700 training points - xz plane

(c) Elliptical Trajectory from the model learned with 800 training points - xy plane

(d) Elliptical Trajectory from the model learned with 800 training points - xz plane

Figure C.22: Elliptical Trajectory from the model learned with 700 and 800training points



(a) Elliptical Trajectory from the model learned with 900 training points - xy plane

(b) Elliptical Trajectory from the model learned with 900 training points - xz plane

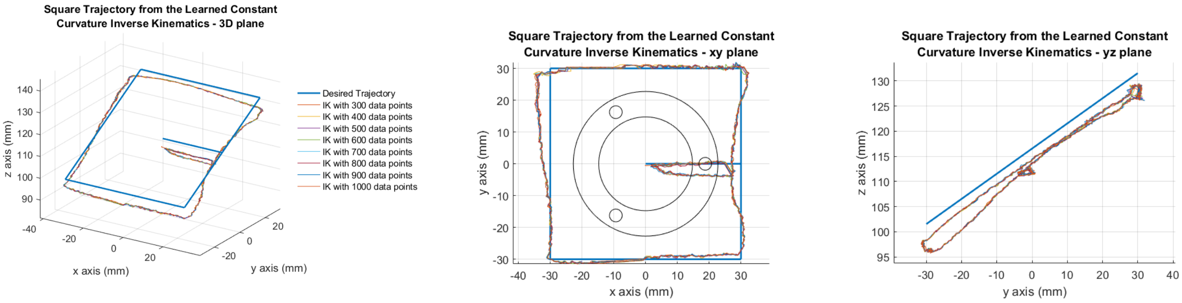
(c) Elliptical Trajectory from the model learned with 1000 training points - xy plane

(d) Elliptical Trajectory from the model learned with 1000 training points - xz plane

Figure C.23: Elliptical Trajectory from the model learned with 900 and 1000 training points

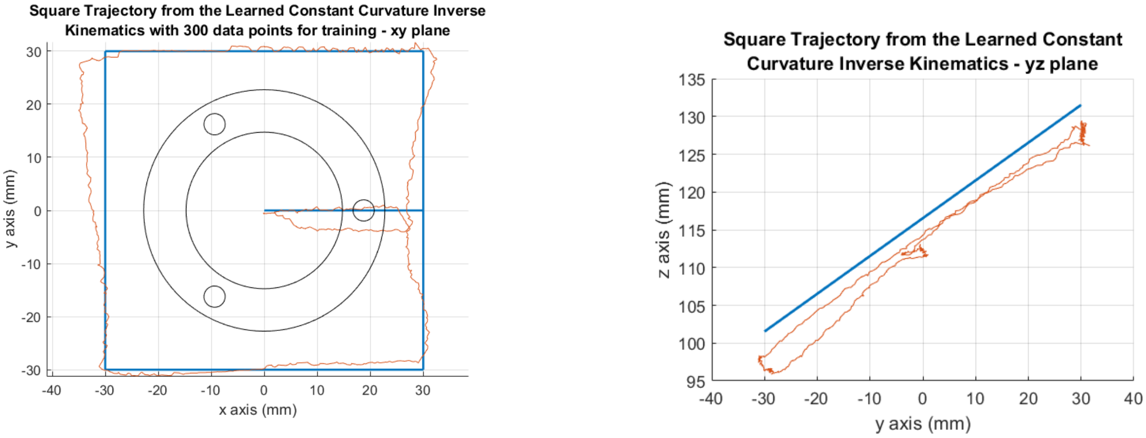
C.2.3 Square Trajectory

From the Constant Curvature Model



(a) Square Trajectory - 3D space (b) Square Trajectory - xy plane (c) Square Trajectory - xz plane

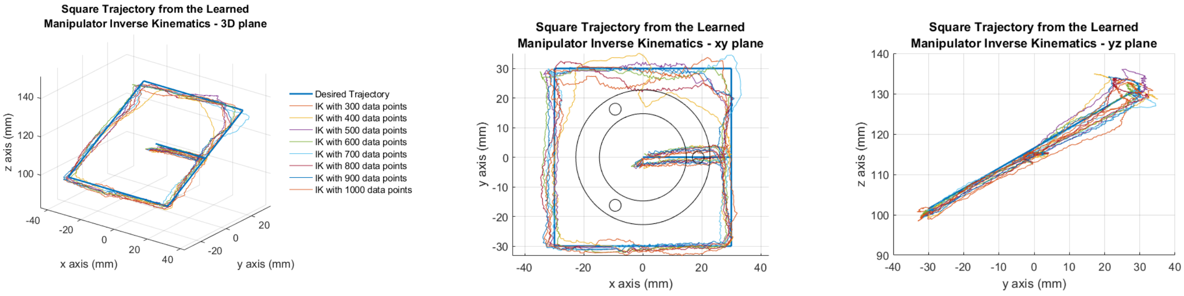
Figure C.24: Square Trajectory from the model learned from the constant curvature theory



(a) Square Trajectory - xy plane (b) Square Trajectory - xz plane

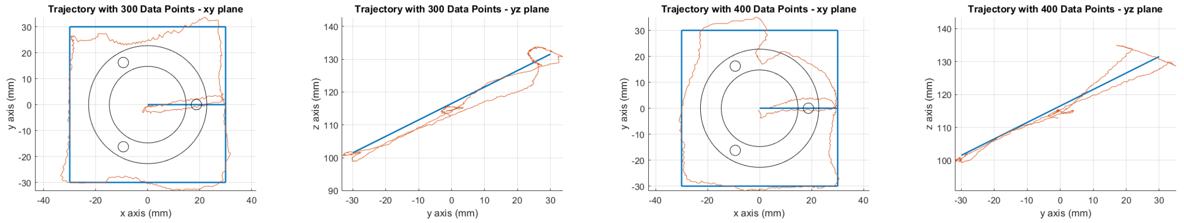
Figure C.25: Square Trajectory from the model learned from the constant curvature theory with 300 training points

From the Manipulator Data



(a) Square Trajectory- 3D space (b) Square Trajectory - xy plane (c) Square Trajectory - xz plane

Figure C.26: Square Trajectory from the model learned from the Manipulator data



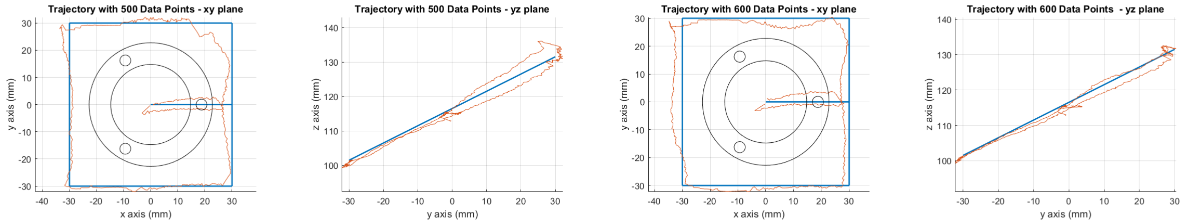
(a) Square Trajectory from the model learned with 300 training points - xy plane

(b) Square Trajectory from the model learned with 300 training points - yz plane

(c) Square Trajectory from the model learned with 400 training points - xy plane

(d) Square Trajectory from the model learned with 400 training points - yz plane

Figure C.27: Square Trajectory from the model learned with 300 and 400 training points



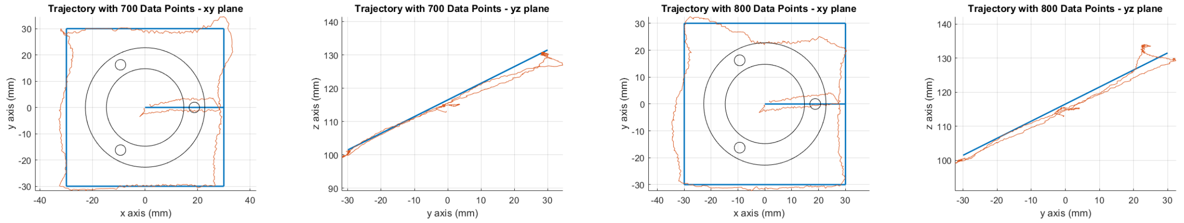
(a) Square Trajectory from the model learned with 500 training points - xy plane

(b) Square Trajectory from the model learned with 500 training points - yz plane

(c) Square Trajectory from the model learned with 600 training points - xy plane

(d) Square Trajectory from the model learned with 600 training points - yz plane

Figure C.28: Square Trajectory from the model learned with 500 and 600 training points



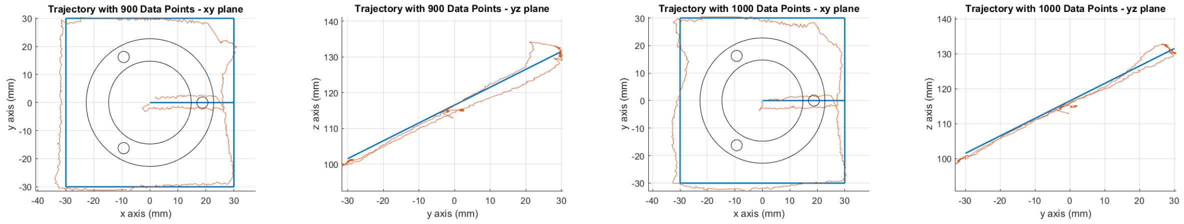
(a) Square Trajectory from the model learned with 700 training points - xy plane

(b) Square Trajectory from the model learned with 700 training points - yz plane

(c) Square Trajectory from the model learned with 800 training points - xy plane

(d) Square Trajectory from the model learned with 800 training points - yz plane

Figure C.29: Square Trajectory from the model learned with 700 and 800 training points



(a) Square Trajectory from the model learned with 900 training points - xy plane

(b) Square Trajectory from the model learned with 900 training points - yz plane

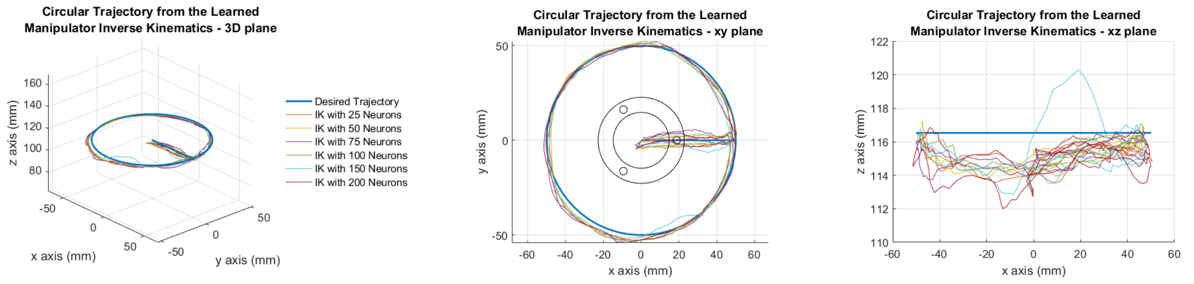
(c) Square Trajectory from the model learned with 1000 training points - xy plane

(d) Square Trajectory from the model learned with 1000 training points - yz plane

Figure C.30: Square Trajectory from the model learned with 900 and 1000 training points

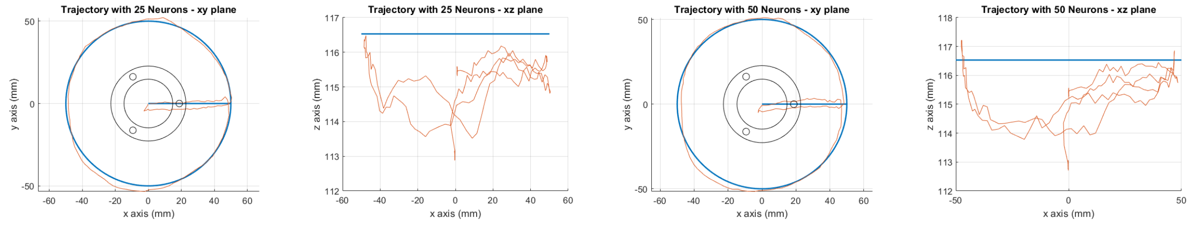
C.3 Trajectories with Variable Neurons

C.3.1 Circular Trajectory



(a) Circular Trajectory - 3D space (b) Circular Trajectory - xy plane (c) Circular Trajectory - xz plane

Figure C.31: Circular Trajectory from the model learned from the constant curvature theory with a variable number of neurons



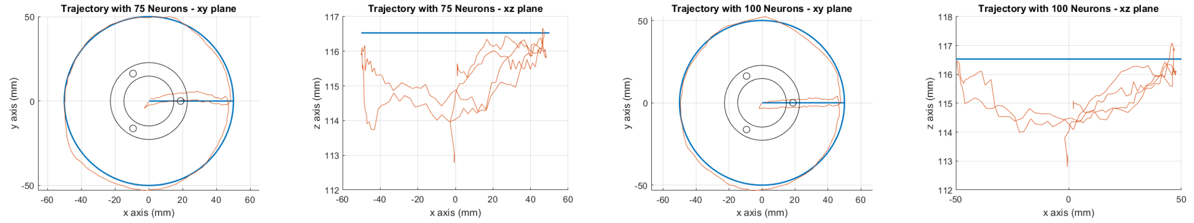
(a) Circular Trajectory from the model learned with 25 Neurons - xy plane

(b) Circular Trajectory from the model learned with 25 Neurons - xz plane

(c) Circular Trajectory from the model learned with 50 Neurons - xy plane

(d) Circular Trajectory from the model learned with 50 Neurons - xz plane

Figure C.32: Circular Trajectory from the model learned with 25 and 50 Neurons



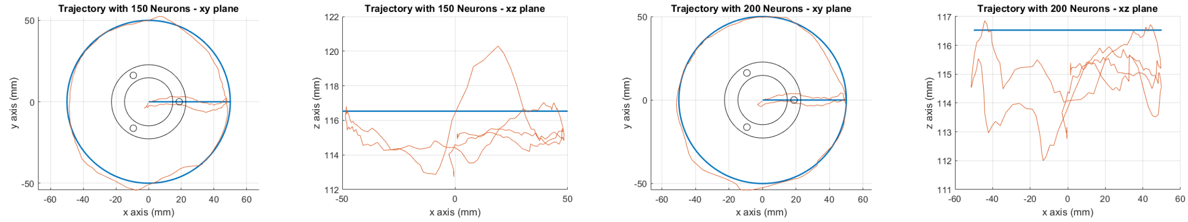
(a) Circular Trajectory from the model learned with 75 Neurons - xy plane

(b) Circular Trajectory from the model learned with 75 Neurons - xz plane

(c) Circular Trajectory from the model learned with 100 Neurons - xy plane

(d) Circular Trajectory from the model learned with 100 Neurons - xz plane

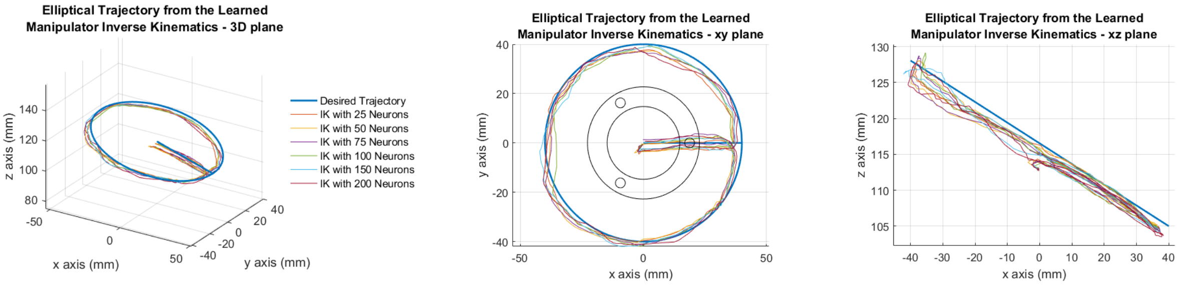
Figure C.33: Circular Trajectory from the model learned with 75 and 100 Neurons



(a) Circular Trajectory from the model learned with 150 Neurons - xy plane (b) Circular Trajectory from the model learned with 150 Neurons - xz plane (c) Circular Trajectory from the model learned with 200 Neurons - xy plane (d) Circular Trajectory from the model learned with 200 Neurons - xz plane

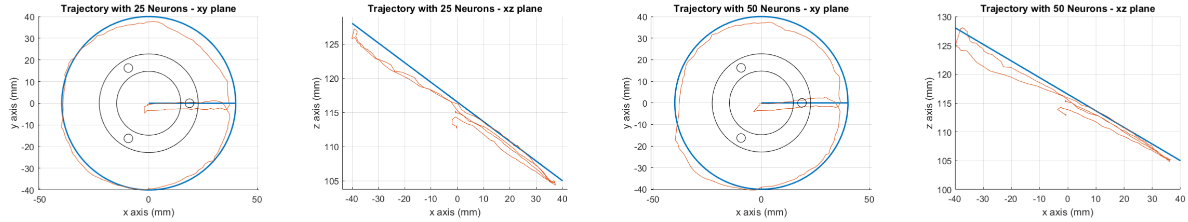
Figure C.34: Circular Trajectory from the model learned with 150 and 200 Neurons

C.3.2 Elliptical Trajectory



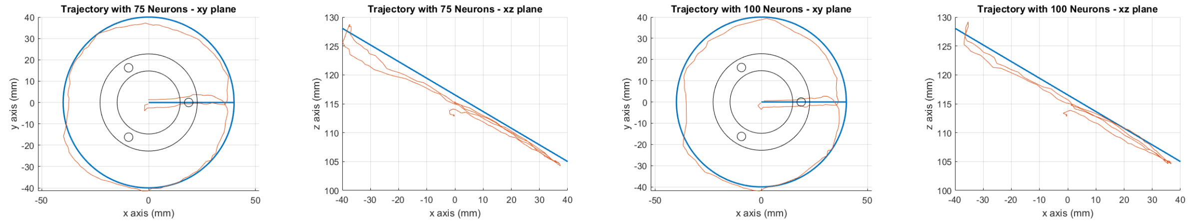
(a) Elliptical Trajectory - 3D space (b) Elliptical Trajectory - xy plane (c) Elliptical Trajectory - xz plane

Figure C.35: Elliptical Trajectory from the model learned from the constant curvature theory with a variable number of neurons



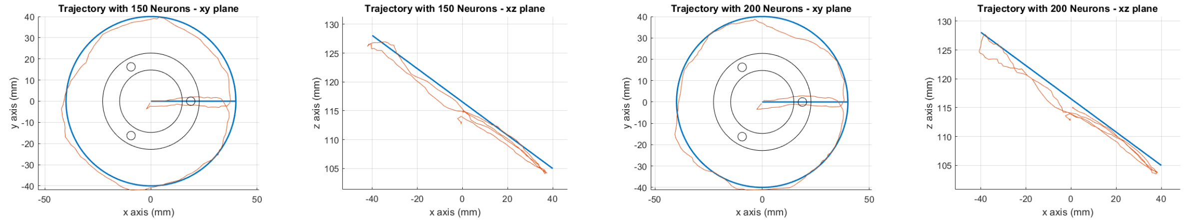
(a) Elliptical Trajectory from the model learned with 25 Neurons - xy plane (b) Elliptical Trajectory from the model learned with 25 Neurons - xz plane (c) Elliptical Trajectory from the model learned with 50 Neurons - xy plane (d) Elliptical Trajectory from the model learned with 50 Neurons - xz plane

Figure C.36: Elliptical Trajectory from the model learned with 25 and 50 Neurons



(a) Elliptical Trajectory from the model learned with 75 Neurons - xy plane (b) Elliptical Trajectory from the model learned with 75 Neurons - xz plane (c) Elliptical Trajectory from the model learned with 100 Neurons - xy plane (d) Elliptical Trajectory from the model learned with 100 Neurons - xz plane

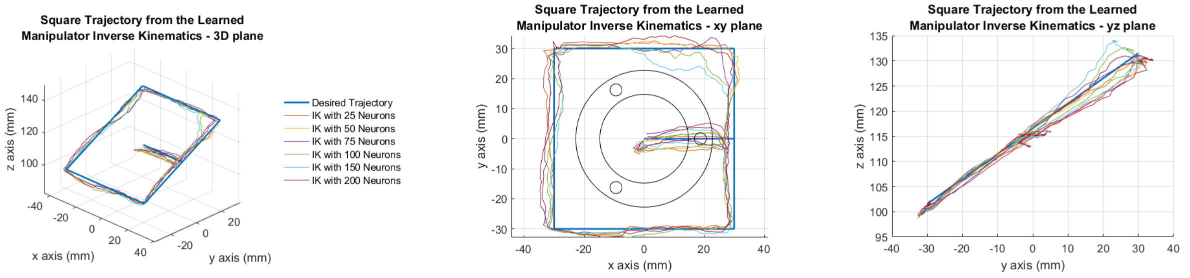
Figure C.37: Elliptical Trajectory from the model learned with 75 and 100 Neurons



(a) Elliptical Trajectory from the model learned with 150 Neurons - xy plane (b) Elliptical Trajectory from the model learned with 150 Neurons - xz plane (c) Elliptical Trajectory from the model learned with 200 Neurons - xy plane (d) Elliptical Trajectory from the model learned with 200 Neurons - xz plane

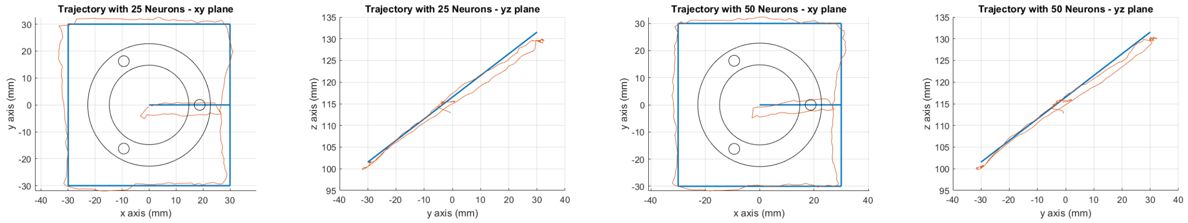
Figure C.38: Elliptical Trajectory from the model learned with 150 and 200 Neurons

C.3.3 Square Trajectory



(a) Square Trajectory - 3D space (b) Square Trajectory - xy plane (c) Square Trajectory - yz plane

Figure C.39: Square Trajectory from the model learned from the constant curvature theory with a variable number of neurons



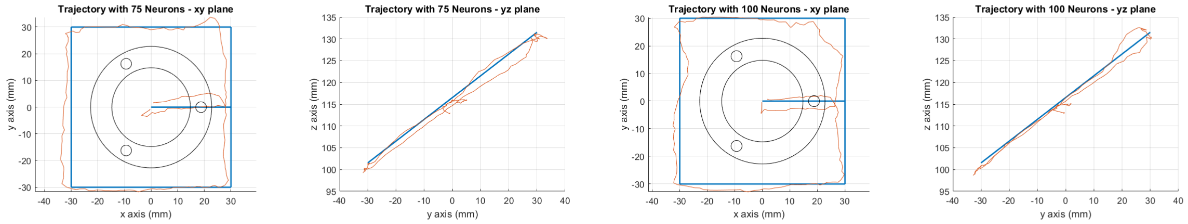
(a) Square Trajectory from the model learned with 25 Neurons - xy plane

(b) Square Trajectory from the model learned with 25 Neurons - yz plane

(c) Square Trajectory from the model learned with 50 Neurons - xy plane

(d) Square Trajectory from the model learned with 50 Neurons - yz plane

Figure C.40: Square Trajectory from the model learned with 25 and 50 Neurons



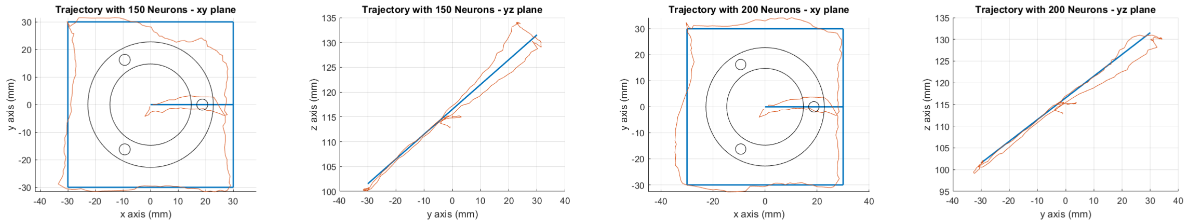
(a) Square Trajectory from the model learned with 75 Neurons - xy plane

(b) Square Trajectory from the model learned with 75 Neurons - yz plane

(c) Square Trajectory from the model learned with 100 Neurons - xy plane

(d) Square Trajectory from the model learned with 100 Neurons - yz plane

Figure C.41: Square Trajectory from the model learned with 75 and 100 Neurons



(a) Square Trajectory from the model learned with 150 Neurons - xy plane

(b) Square Trajectory from the model learned with 150 Neurons - yz plane

(c) Square Trajectory from the model learned with 200 Neurons - xy plane

(d) Square Trajectory from the model learned with 200 Neurons - xz plane

Figure C.42: Square Trajectory from the model learned with 150 and 200 Neurons