POLITECNICO DI TORINO

Master's Degree
in Mechatronic Engineering

Master's Degree Thesis

Dynamical model for Parallel Continuum Robots using the Cosserat rod theory

Supervisor
Prof. Marina Indri
Supervisors at Femto-st
Postdoc. Benjamin Mauzé
MCF Redwan Dahmouche

Candidate
Nicola Lombardi

October 2021

Abstract

Parallel continuum robots are new structures that have enormous advantages in terms of safe human-machine interaction, possibility of miniaturization, or lower weight, but there are still areas that need to be explored further. One of these is dynamics. There are plenty of static models of parallel continuum robots, but few are studying their dynamics. The purpose of this work therefore is to create a dynamic model for parallel continuum robots that will be fast and accurate, so that it can be used for real-time applications.

To create a model of the robot, the flexible elements need to be modelled. To do this, it was decided to apply Cosserat's theory of elasticity. From this model, partial derivative equations are obtained. To solve numerically the equation an algorithm formed by three loops one inside the other is used. The most internal is used for integration in space. The central one contains the optimization algorithm. The algorithm starts from a guessed result, iterating it changes the value of the guessed until the boundary conditions are met. Finally, the outermost loop is used to advance in time.

Three different simulations have been performed, with a cantilever rod, with a Stewart-Gough continuum robot and with the Triskele-bot, a continuum parallel robot developed inside the Femto-st laboratory. They are performed in order to understand how the simulation parameters affect the results, how big is the noise due to the simulation, and if the model is valid for both forward and inverse dynamics.

Finally, trying to validate the model two experiments are performed with the Triskele-bot. In the first one an external force was applied to the robot and after it was released. In the second one a step input is given with the actuators. They are made to observe the free oscillation of the robot.

Keywords: Parallel continuum robots, Dynamic, Cosserat rod model

Contents

Chapter 1

Introduction

A new branch of robotics is continuous robots. These robots are characterized by having infinite degrees of freedom and number of joints, so they have the ability to take continuous curved shapes.

These robots are characterized by higher work volume, higher operating speeds, lower weight, lower power consumption, better maneuverability and better transportability; that allow safer operations due to lower inertia, and can be low cost if desired.

The applications can be very numerous. In the medical field the most famous application is certainly the endoscope, which allows to look inside the body of patients through the natural orifices. But it is also possible to perform brain, lung, endovascular, gastroenterological, and urogenital interventions.

Moreover it is possible to use them for industrial applications, thanks to the great scalability, it is possible to use them for micro-assembly, micro-manipulation, or thanks to the lower inertia it is possible to use them for collaborative applications.

They can reach hazardous places, such as for the inspection and control of wastewater or they can be used for space or military applications.

A new subcategory of continuum robots are Parallel Continuum Robots (PCRs). They consist of a set of flexible elements connected in parallel to a platform and actuated independently.

Compared to the polyarticulated robots, these structures have a lower weight, and therefore the human-machine interaction is safer. Thanks to the possibility of replacing the joints with flexible elements they have a better chance of miniaturization and a greater workspace.

Moreover, compared to continuous robots in series, they have a better stiffness of the structure, thus improving the accuracy of the robot.

But despite the many pros there are still challenges to be faced.The biggest one is to create a model that can predict the shape and behavior of a parallel continuum robot.

Modeling continuum robots can be done in several ways, the most basic being the study of kinematics used to represent the geometry of the robot. Another method is to study the statics of the robot, thus also investigating the forces, i.e. the causes of motion. These two approaches have been well analyzed in the scientific literature, with several studies. One approach not yet well analyzed is Dynamics. And this is precisely the topic that will be discussed in this work.

In particular, the objective of this work is to develop a model capable of simulating the dynamics of a parallel continuum robot. The model must be applicable for different geometries and structures of parallel continuum robots. It must also be potentially usable in a real-time simulation, so in the future it could be used for a controller. A goal we want to achieve is also the possibility to have a good scalability of the model and a good accuracy. So that it can be used for both large robots and precision structures.

To obtain these characteristics, it was decided to create a model based on Cosserat elasticity theory. Cosserat's theory considers the flexible element as a slender rod capable of undergoing bending, twisting, stretching, and shearing. Using this method we obtain Partial Derivative Equations (PDE) for which it is not possible to obtain an analytical solution for Parallel Continuum Robots.

It is therefore necessary to obtain a numerical solution. In order to obtain an Ordinary Differential Equation (ODE) from the PDE with one spatial dimension and a time dimension it is necessary to discretize the time partial derivative. To do that the BDF-α method is used. So once obtained the ODEs the equations have been integrated in space using the Runge-Kutta 4 (RK4) method.

Moreover, not all boundary conditions are available for such model. To resolve this problem therefore it is necessary to use an optimization algorithm. The algorithm starts from a guessed boundary condition, iterating it changes the value of the guessed until some geometrical constraints or balance of the forces are met.

With the model some simulations are performed. The first one is to understand how the different parameters are influencing the simulation.

The second simulation is with the continuum version of the Stewart-Gough platform. It is composed by 6 rods, connected in parallel on a platform and actuated by linear actuators. Starting from a static equilibrium the simulations are performed to understand the order of magnitude of the noise obtained by the algorithm.

The third simulation is with the Triskele-bot. A planar parallel continuum robot composed by 3 rods connected in parallel to a platform. For this simulation the robot must follow a trajectory with the inverse dynamic, and the results are used in the forward dynamic.

So, an experiment is performed in order to validate the model. For this purpose the

Triskele-bot is used, a prototype of Femto-st designed for precision applications.

The experiment is performed pushing and releasing the platform. To understand the free movement of the robot and compare it with the simulation.

In conclusion, regarding the simulations the results are quite satisfactory, having understood how to adjust the various parameters, obtaining a noise of 10 nm, and in the third simulation, an error of 1.76% compared to the simulated displacement.

Instead, as regards the comparison with the experiment, the result is not fully satisfactory. As there would be a need to analyze experimentally or through calibration of some parameters such as the damping matrix, and the length of the rods.

In spite of this, there are good hopes of improvement, and the use of the Cosserat model seems to be the most suitable for the intended purposes.

Context: Femto-ST

The work presented in this thesis is part of an internship carried out in the institute FEMTO-st "Franche-Comté Electronics Mechanics Thermal Science and Optics - Sciences and Technologies". This research center is under the authority of the University of Franche-Comté (UFC), the French National Centre for Scientific Research (CNRS), the Ecole Nationale Supérieure de Mécanique et Microtechniques (ENSMM) and the University of Technology Belfort-Montbéliard (UTBM).

The internship was carried out in the department AS2M, which deals with robotics, automatic control, mechatronic and artificial intelligence. The department is internationally recognized for the design and control of systems for micro and nano positioning and assembly and for micro robotics for minimal invasive surgery. Within the department there are 3 teams:

- The MACS team (Methodologies for Automation and for the Design of Mechatronic Systems).
- The PHM team (Prognostics and Health Management).
- The Micro and Nanorobotics team.

This internship was developed in the Micro and Nanorobotics team. The purpose of this team is to study the design, perception, control and metrology issues for micro and nanorobotic systems.

Thesis Structure

The thesis is structured as follows:

- In chapter 2 there is a state of the art for continuous parallel robots. The structure is explained, some PCR examples are given and the different approaches in modeling are described.
- In chapter 3 the theory and the algorithms behind the Lumped model and Cosserat model are presented.
- In chapter 4 are described the simulations carried out on the model of a cantilever rod, on the model of a Continuum Stewart-Gough Robot and finally on the Triskelebot.
- In chapter 5 there is a description of the setting of the experiment, on the Triskelebot, and how the measurements of its pose are carried out. Finally two experiments are described, the first in which there is a free motion, and the second in which the robot is actuated.
- In chapter 6 there are conclusions and possible future developments of this work.

Chapter 2

Parallel Continuum Robots: State of the art

A robot is an artificial device that performs certain actions based on commands given to it, either under direct human supervision, or autonomously based on general guidelines, perhaps using artificial intelligence processes. The tasks typically should be performed in order to replace or assist humans, such as in manufacturing, construction, manipulation of heavy and dangerous materials, or in prohibitive environments or those not compatible with the human condition, or simply to free humans from tasks.

Robots can be used in many different applications, so we can classify them according to their use.

The most common category is industrial robots. These robots are similar to human arms with a wrist and an end-effector that can hold tools. These robots are generally attached to a base

Humanoid robots are robots with human-like features. They can have arms, legs, and a torso, or replicate only one part of the human body, such as facial expressions. They are used for the study of bipedal locomotion, for human-machine interaction and in various research areas.

Biomimetic robots are mechanisms that take inspiration from natural systems. The purpose of these robots is to exploit a mechanism found in nature to address specific tasks. So these robots can be shaped like a fish to swim, shaped like a snake to crawl or take inspiration from nature to jump, fly, climb or walk.

Mobile robots are robots that can move and are not fixed in one position. Movement can be through wheels, legs or they can fly. They can move according to a predetermined path or even autonomously. To do this they are equipped with sensors such as cameras, lidar scanners or proximity sensors.

These different categories can overlap with each other.

2.1 Types of Robots

Another way to define robots is to analyze the mechanical structure of the elements. In particular, we can recognize three categories of robots. Polyarticulated robots, compliant robots and continuum robots.

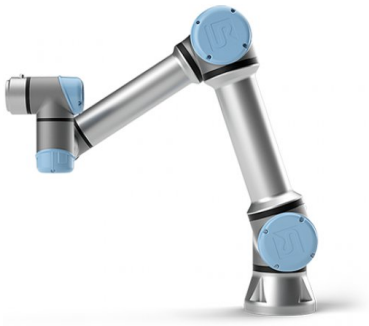
2.1.1 Polyarticulated robots

Polyarticulated robots are composed of rigid elements connected by joints. They are the most widely used robots.

These robots are generally robust to external perturbations and have significant masses. Their modeling is based on solid mechanics, so it is very simple and has an analytical solution.

These robots are therefore used as a reference for other types of robots. In figure 2.1 you can see two examples of industrial robots.

(a) Stewart-Gough robot



(b) Universal Robot UR3

Figure 2.1: Examples of industrial polyarticulated robots

2.1.2 Flexible Articulation Robots

The principle of Flexible Articulation Robots (or Compliant Robots) is to replace the joints with a flexible element. It is possible to create a flexible joint in two ways. The first is to use a flexible material clamped between two links. The second is called local slimming and consist to use a single link with a reduced cross section at the point of the joint.

With this solution it is possible to eliminate friction and backlash in the joints, such robots do not need lubrication, have greater accuracy and can be much easier to manufacture. These features make them suitable for miniaturization and thus the best solution for precision applications in the micro- and nano-scale.

However, these structures generally have a small workspace due to the small deformation ranges of the elements used, which quickly reach their mechanical limit due to the concentration of stresses in their joints. For this reason they can also be subject to fatigue wear along the joints.

In figure 2.4 is possible to observe the MilliDelta robot of the Harvard University [15]. This robot is composed of a local slimming and a piezo-electric beam that deforms in bending. It is capable of reaching the speed of 0.45 meters per second with a precision of 5 micrometers.

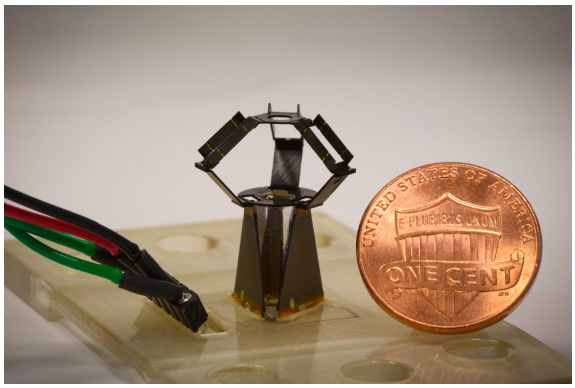


Figure 2.2: MilliDelta robot of the Harvard University

2.1.3 Continuum Robots

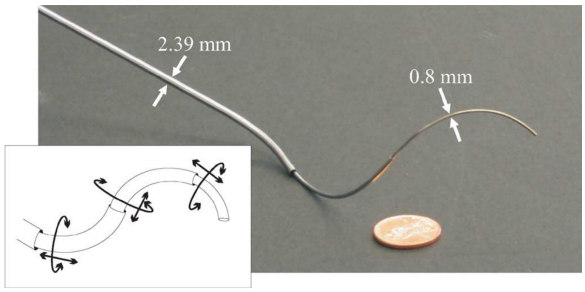
Continuum robots are composed entirely or only partially of flexible elements that deform continuously. These robots can deform, stretch, twist, warp. So they can be made of a wide range of materials, from very soft materials to very flexible materials to shape memory materials.

They have theoretically an infinite number of degrees of freedom (DoF), can perform a curvilinear path-way to reach positions impossible for rigid robots. They have a smaller weight compared to the polyarticulated robot, so they are safe in human-machine operations.

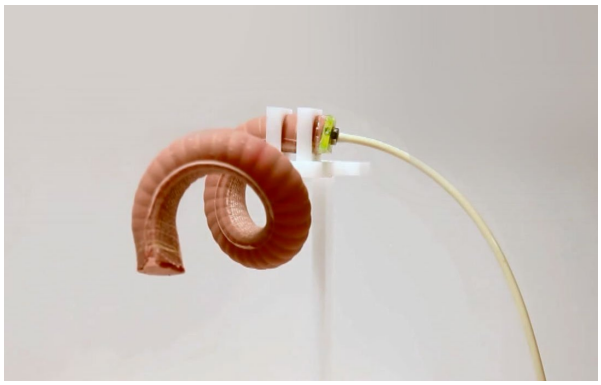
It is possible to define a subcategory of continuous robots that are soft robots [8]. They differ because they have a Young's modulus of less than 1 GPa [9] and are composed of materials such as silicone, polymers, or innovative materials.

An example of a continuous series robot is the concentric tube robot used for Colonoscopy (Figure 2.3) presented by Webster *et al.* [23]. It consists of several precurved concentric tubes. By rotating between them, the curvature of the tubes generates new robot geometries.

The robot in figure 2.3b instead is an example of soft robot. It is composed of a flexible multi-material and is actuated by pressurized fluid.



(a) Continuum robot, image from [23]



(b) Soft robot, image from [11]

Figure 2.3: Examples of continuum robots

2.2 Robotic Architectures

There are three major types of possible architectures: series, parallel, and hybrid robots.

Series robots are composed of an open kinematic chain consisting of unitary elements linked together by mechanical connections between the base and the end-effector.

Parallel robots are closed kinematic chain mechanisms connecting the base to the end effector by several kinematic chains.

Hybrid structures are a combination of series and parallel architectures.

The breakdown of architectures also applies to Continuum robots. In fact, the type of robots that will be analyzed in this paper will be Parallel Continuum Robots.

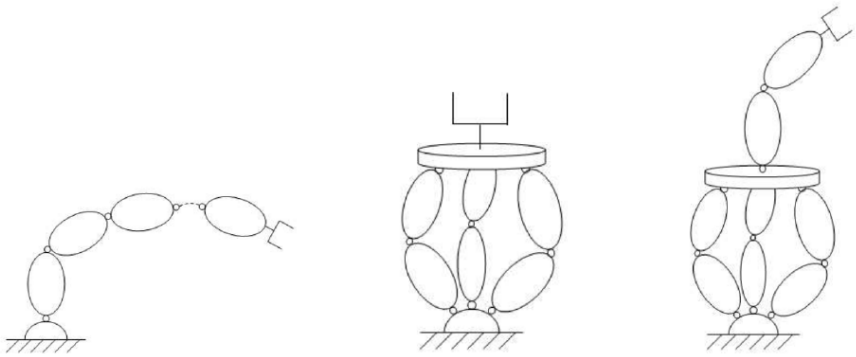
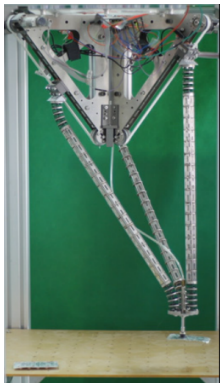
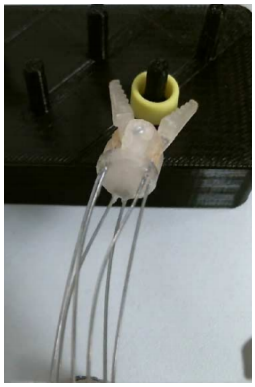


Figure 2.4: From left to right: a series architecture, a parallel architecture, an hybrid architecture. Image from [10]



(a) Continuum version of the Delta robot, image from [25]



(b) Continuum Stewart-Gough robot, image from [17]

(c) Tendon Actuated Planar Parallel Continuum Robot, image from [16]

Figure 2.5: Different structures of PCR

2.3 Parallel Continuum Robots

Parallel continuum robots (PCR) are composed by several flexible elements connected in parallel and each one must be operated independently [5]. These types of robots tend to have good loading capacity, exert greater forces, and have better accuracy and repeatability [4].

Parallel continuum robots have a larger workspace than their rigid counterparts, due to the deformation of the flexible elements [19] [7]. For example, Yang *et al.*[25] proposed a version of the Delta robot in which they replaced the rotational joints with continuous joints (Figures 2.5a).

A much analyzed structure is the continuum version of the Stewart-Gough platform (Figure 2.5b), analyzed in several papers such as [17], [22], or [4]. The Stewart-Gough platform is a parallel robot with six linear actuators, attached between the base plate and top plate by universal joints. The top plate is therefore able to move in 6 DoF: three translations and three rotations. It is possible to observe an example of this robot in Figure 2.1a. The continuum version of the Stewart-Gough robot (Figure 2.5b) consists of 6 continuum links connected in parallel to an end-effector platform. The rods pass through holes in the platform at the base and then join the end-effector. The actuation of the platform is obtained by lengthening or reducing the length of the rod.

A Planar Parallel Continuum Robot (Figure 2.5c) was created by Nuelle *et al.* [16]. It is a tendon actuated robot where the tendons are substituting the actuators in order to reach a bigger workspace. This robot obtained a positioning repeatability of 1.0% in relation to one continuum segment length of the robot, and positioning accuracy of 1.4%. These results are comparable to commonly used kineto-static modeling approaches for PCR.

The Triskele-bot is a continuum parallel robot (PCR) designed for planar positioning applications at the micrometer scale.

The Triskele-bot is an alternative to the 3-$\underline{\text{P}}$RR. This structure is a rigid parallel robot constituted of three independent kinematic chains. Each chain is composed of 3 joints: prismatic (P), rotational (R) and rotational(R). The three chains are constrained to the platform so that they have an angular distance of 120°. The robot platform has 3 degrees of freedom, translations along x and y, and a rotation around the z axis.

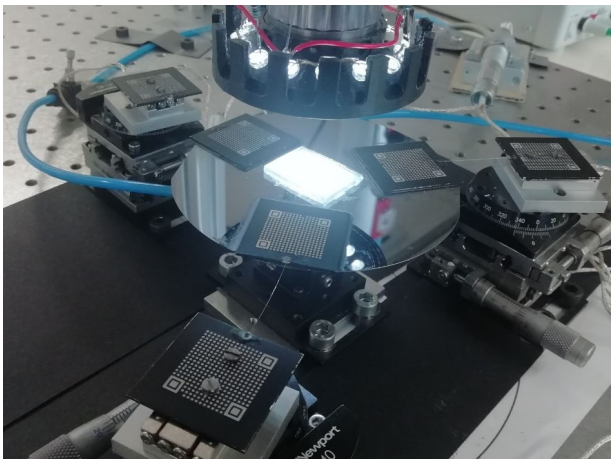


Figure 2.6: Triskele-bot

In the Triskele-bot the revolute joints and the intermediate rigid bodies have been replaced by a flexible body (F). This PCR can be abbreviated as 3-$\underline{\text{P}}$F, and is called Triskele-bot because the geometry resembles to a Triskele symbol. The degrees of freedom of the platform remain 3 just like in the rigid configuration. Figure 2.6 represents a kinematic diagram of the Triskele-bot.

Therefore the goal of the robot is to have a good accuracy and repeatability. These characteristics have been well analyzed in [14], [13] and [12], regarding statics, but not yet in dynamical analysis. So, this project will be focused on the design and development of a Parallel Continuum Robot.

2.4 Modellization of Parallel Continuum Robots

For years, polyarticulated robots have been studied and analyzed. Their modeling is based on solid mechanics and admits an analytical solution. This is because their number of degrees of freedom is well defined.

For continuous robots this is not the case. They have an infinite number of degrees of freedom. This means that each configuration of the robot can be defined by an infinite number of independent coordinates. This makes modeling flexible bodies difficult.

There are many techniques for modeling continuum robots. The first way to break them down is to figure out if the model is based on physics or is a black-box model.

For example Wu *et al.* [24] and Yip *et al.*[26] used an artificial neural network to simulate the behaviour of a PCR. With these models, it is possible to understand the condition of the robot's endeffector, i.e., position and force, without knowing the kinematics or mechanics and with unknown contraints along the flexible body.

For models that are based on physics, different approaches can be used, as can be seen in Figure 2.7 it is possible to make a distinction regarding kinematics and mechanics, and a distinction between discrete and continuous models.

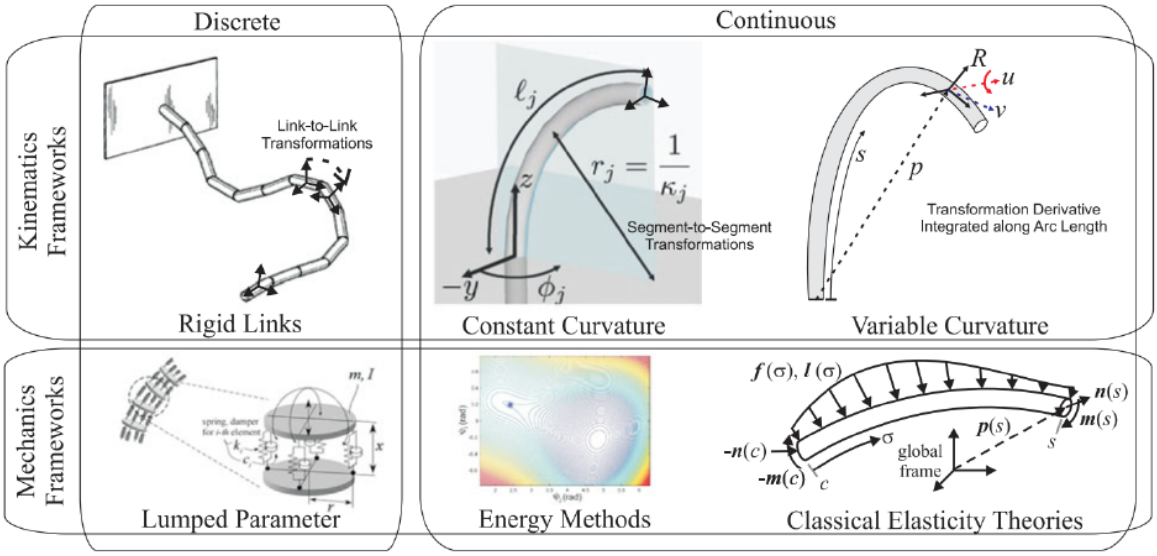


Figure 2.7: Different type of models for Continuum Robots [6]

A discrete model considers a flexible element as composed of a discrete number of rigid bodies. For discrete kinematic models the flexible elements are considered as a discretization of rigid transformations. In contrast, for the Lumped model it is possible to consider a finite number of masses connected by joints, springs and dampers.

There are two ways to model in the kinematic framework with a continuous approach: using constant curvature models, which are based on a decomposition into a set of sections

whose neutral fiber is of constant curvature; and variable curvature models, where the neutral fiber is parametrized by a curvilinear abscissa and a local reference frame is attached to any point of the neutral fiber.

To model in the mechanics frameworks with a continuous approach it is possible to use energetic approaches (Euler Lagrange) or approaches that follow theories of elasticity. This last method allows to have more complete models. An example of this is the Kirchhoff model [20] where the slender rod is incorporating bending and twisting or the Cosserat model [21] that is a generalization of the Kirchhoff theory accounting also for stretching and shearing.

In addition, it is possible to make a classification according to the fidelity of the model. There are models with low fidelity that reduce the number of parameters in order to simplify the model and to have lighter simulations, and models with high fidelity that have the objective to have high accuracy at the price of low speed of simulation.

Considering the current state-of-the-art, most models are studying continuum robots in static or quasi-static condition and the field of dynamics is little explored. The main dynamical models were developed by Till *et al.* [22] and Renda *et al.* [18]. Both models are based on Cosserat's theory of elasticity, which will be precisely the approach explored and analyzed in this report.

According to Cosserat's theory the flexible element is considered as a one-dimensional rod, which corresponds to the neutral fiber of the flexible element. Such a rod is able to undergo bend, twist, stretch, and shear; allowing all possible modes of deformation to be considered under a wide range of boundary conditions.

In particular in this work we will analyze the model of Cosserat because according to the studies carried out by Till *et al.* [22] would have the following characteristics:

- numerical consistency with continuous theory;
- real-time computation;
- good spatial scalability;
- stability in a large time-step;
- accuracy at steady state;
- low numerical damping.

These characteristics would make it the suitable model for a future applications as a basis for a real-time controller.

Chapter 3

Dynamic models for Parallel Continuum Robots

In this chapter the theory and the algorithms behind the Lumped model and Cosserat model will be presented. As it is possible to see in figure 2.7 the first is a discrete model, and the second is a continuous model that follows the classical elasticity theory.

3.1 Lumbed model

The first and simplest method for describing the dynamics of a continuum structure is the lumped element model. This model consists in describing a flexible body by a finite number of rigid bodies that approximate the behavior of the distributed system. In this way there is a transition from partial differential equations (PDEs) to a system of ordinary differential equations (ODEs), which can be solved analytically.

A constraint that depends on the type of deformation is applied between the adjacent bodies.

Interactions between bodies are also described by springs and dampers. They generate forces and torques between the bodies. Next paragraphs will present the spring and damping coefficient need to define those interactions.

Spring Coefficient

Hooke's law is used to describe a spring, so:

$$f = k\delta \tag{3.1}$$

where f is the spring force, k is the spring constant, and δ is the elongation of the spring. From the classical beam theory, the force on a continuous beam unit is:

$$F = \frac{EA}{l}\delta \tag{3.2}$$

where F is the force, E is Young's modulus of elasticity, A is the cross-sectional area of the beam and l is the length of an undeformed flexible unit, given by the formula:

$$l = \frac{L}{N} \tag{3.3}$$

where L is the length of the beam and N is the number of discretization for the rod. So finally it is possible to compare Hooke's law and the theory of elasticity obtaining the coefficient of elasticity as:

$$k = \frac{EA}{l} \tag{3.4}$$

For a rotational spring it is possible to follow the same procedure, obtaining the rotational spring coefficient k_T as:

$$k_T = \frac{GJ}{l} \tag{3.5}$$

where G is the shear modulus and J is the torsional constant.

Damping Coefficient

Damping of materials is a complex subject. The simplest model consists of considering the damping coefficient b proportional to the spring coefficient k:

$$b = \tau k \tag{3.6}$$

Where the constant τ is set from benchmark data, or calculated following a reliable model.

3.1.1 Model

The Simscape Multibody Flexible Body Library for MATLAB Simulink was used to create this model. In such environment, it is possible to create a Multibody model. Using black-boxes that represent bodies, constraints and reference systems. To such bodies it is possible to apply forces or request trajectories, and obtain outputs. In Figure 3.1 it is possible to observe the model of the Triskele-bot.

On the left there is the global reference system to which everything refers. Proceeding to the right we find in parallel the masses that represent the actuators. These are connected to the global reference system through a geometric transformation. Continuing to

the right it is possible to find the prismatic joints, which can be given a law of motion or forces; they represent the movement of the actuators.

Next we find the blocks that represent the flexible bodies. They are black boxes, whose parameters are to be set.

Finally there is the platform represented by a cylinder. It is connected to the global reference by a planar joint.

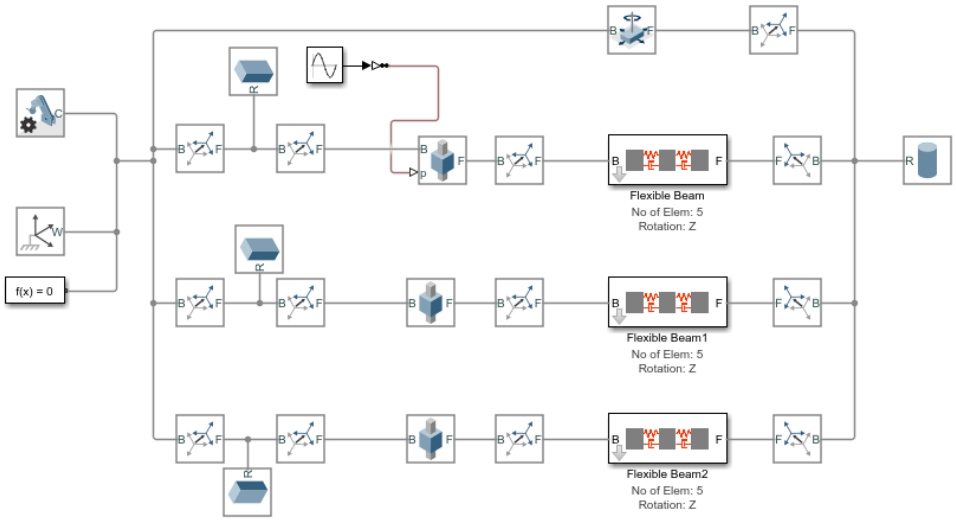


Figure 3.1: Simulink model of the Triskele-bot

3.1.2 Analysis

The advantage of this model is its simplicity to create it. The problem is that its fidelity depends on the number of discretization for each beam, and increasing the number of bodies exponentially increases the simulation time. This is due to the fact that the number of variables and equations to be solved is increased.

Since we want to apply this model to precise robots and possibly in a real-time simulation, the lumped parameter model has been set aside, in favor of the Cosserat model that will be analyzed in the following section.

3.2 Cosserat model

This chapter will initially present and analyze the equations of statics according to Cosserat-based model. Subsequently, these equations are derived with respect to time to obtain the equations of dynamics. Finally the numerical techniques that are used to solve the equations are presented.

Table 3.1 introduces the notation that will be presented in this chapter.

Symbol	Definition
s	Reference arclength
t	Time
$\boldsymbol{p}$	Position vector in cartesian coordinates
$\mathbf{R}$	Rotation matrix of material orientation
$\boldsymbol{n}$	Internal force
$\boldsymbol{m}$	Internal moment
$\boldsymbol{f}$	Distribuited force
$\boldsymbol{l}$	Distribuited moment
$\boldsymbol{l}$	Rate of change of the position vector
$\boldsymbol{u}$	Curvature vector
$\boldsymbol{q}$	Velocity vector
$\boldsymbol{\omega}$	Angular velocity
A	Cross section area
$\mathbf{J}$	Moment of inertia tensor
$\mathbf{K}_{se}$	Stiffness matrix for shear and extension
$\mathbf{K}_{bt}$	Stiffness matrix for bending and twisting
$\mathbf{B}_{se}$	Damping matrix for shear and extension
$\mathbf{B}_{bt}$	Damping matrix for bending and twisting
E	Young modulus
G	Shear modulus
$\mathbf{C}$	Drag coefficient matrix
$\boldsymbol{g}$	Gravitational acceleration vector
$\hat{(\cdot)}$ or $(\cdot)^{\wedge}$	Skew matrix operator
$\check{(\cdot)}$ or $(\cdot)^{\vee}$	Inverse skew matrix operator
$(\cdot)^h$	Historical part of the function for the time discretization

Table 3.1: Notation used for the Cosserat theory

3.2.1 Static equations

In order to create a mechanical model of a rod whose length is much larger than the other dimensions, a good approximation is to consider it as a one-dimensional object. As showed in figure 3.2 the rod is defined as a curve discretized spatially by the curvilinear abscissa $s \in [0, L]$, where L is the length of the rod.

For the Cosserat model the position of each point of the curve that represents the center line of the rod is described by the function $\boldsymbol{p}(s)$. The position of the curve, however, is not sufficient to describe the beam precisely, because orientation must also be considered. Orientation is represented by a rotation matrix $\mathbf{R}$.

The variable $\boldsymbol{v}$ is introduced as the derivative of the position in the local frame. It is described by the equation $\boldsymbol{v} = \mathbf{R}^T \boldsymbol{p}_s$. Similarly, curvature in the local frame $\boldsymbol{u}$ is introduced, described by the equation $\boldsymbol{u} = (\mathbf{R}^T \mathbf{R})^{\vee}$, where the operator $^{\vee}$ is the inverse

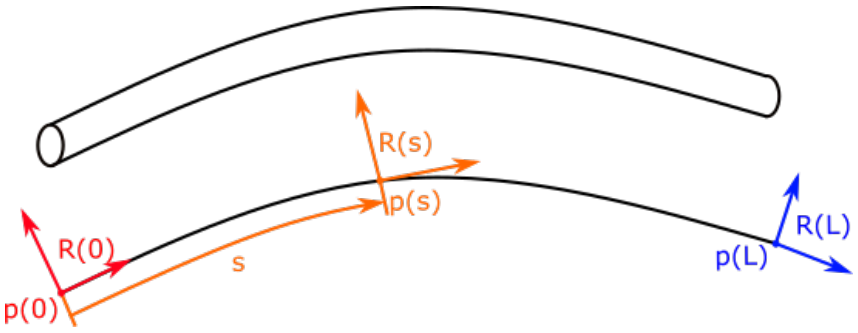


Figure 3.2: The rod is represented as a curve. Each point of the curve have a position described by the function $p(s)$, and an orientation described by $R(s)$

of the skew matrix.

Internal forces are described by $\boldsymbol{n}(s)$ and internal moments by $\boldsymbol{m}(s)$. As showed in Figure 3.3 the sign convention is chosen so that the force $\boldsymbol{n}(s)$ is the force that $s+\delta$ material exerts on the $s-\delta$ material, the same applies for the moment.

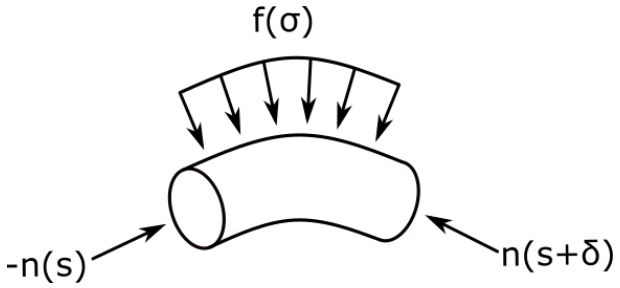


Figure 3.3: Force balance on an infinitesimal slide of the rod

The external forces and moments distributed on the beam are represented respectively by $\boldsymbol{f}(s)$ and $\boldsymbol{l}(s)$. A differential equation for $\boldsymbol{n}_s(s)$ can be derived by considering the force balance for an infinitesimal slide of the rod from s to $s+\delta$. Therefore the balance is given by the equation:

$$\boldsymbol{n}(s+\delta) - \boldsymbol{n}(s) + \int_s^{s+\delta} \boldsymbol{f}(\sigma) d\sigma = 0 \tag{3.7}$$

The equation is differentiated obtaining the following equation.

$$\boldsymbol{n}_s = -\boldsymbol{f} \tag{3.8}$$

The differential equation for $\boldsymbol{m}_s$ is found balancing the moments for an infinitesimal section

$$\boldsymbol{m}(s+\delta) - \boldsymbol{m}(s) + \boldsymbol{p}(s+\delta) \times \boldsymbol{n}(s+\delta) - \boldsymbol{p}(s) \times \boldsymbol{n}(s) + \int_s^{s+\delta} [\boldsymbol{l}(\sigma) + \boldsymbol{p}(\sigma) \times \boldsymbol{f}(\sigma)] d\sigma = 0 \tag{3.9}$$

So, differentiating it is obtained:

$$\boldsymbol{m}_s = -\boldsymbol{l} - \boldsymbol{p} \times \boldsymbol{f} - \frac{\partial}{\partial s}(\boldsymbol{p} \times \boldsymbol{n}) \tag{3.10}$$

Differentiating and simplifying it is obtained:

$$\begin{aligned} \boldsymbol{m}_s &= -\boldsymbol{l} - \boldsymbol{p} \times \boldsymbol{f} - \boldsymbol{p}_s \times \boldsymbol{n} - \boldsymbol{p} \times \boldsymbol{n}_s \\ &= -\boldsymbol{l} - \boldsymbol{p} \times \boldsymbol{f} - \boldsymbol{p}_s \times \boldsymbol{n} + \boldsymbol{p} \times \boldsymbol{f} \\ &= -\boldsymbol{l} - \boldsymbol{p}_s \times \boldsymbol{n} \end{aligned} \tag{3.11}$$

Therefore the statics of the rod according to the Cosserat model is described by the following set of equations:

$$\begin{aligned} \boldsymbol{p}_s &= \mathbf{R}\boldsymbol{v} \\ \mathbf{R}_s &= \mathbf{R}\hat{\boldsymbol{u}} \\ \boldsymbol{n}_s &= -\boldsymbol{f} \\ \boldsymbol{m}_s &= -\boldsymbol{l} - \boldsymbol{p}_s \times \boldsymbol{n} \end{aligned} \tag{3.12}$$

To completely constrain the ODE system, constitutive equations are used that relate the internal loads to the strain. It is used the linear elastic relation:

$$\begin{aligned} \boldsymbol{n} &= \mathbf{R}\mathbf{K}_{se}(\boldsymbol{v} - \boldsymbol{v}^*) \\ \boldsymbol{m} &= \mathbf{R}\mathbf{K}_{bt}(\boldsymbol{u} - \boldsymbol{u}^*) \end{aligned} \tag{3.13}$$

where the variable $\boldsymbol{v}^*$ and $\boldsymbol{u}^*$ are the shape of the rod in stress-free situation. The stiffness matrices indicate shear and extension with "se" and bending and torsion with "bt". For an omogeneous material they are:

$$\mathbf{K}_{se} = \begin{bmatrix} GA & 0 & 0 \\ 0 & GA & 0 \\ 0 & 0 & EA \end{bmatrix}; \mathbf{K}_{bt} = \begin{bmatrix} EI_{xx} & 0 & 0 \\ 0 & EI_{yy} & 0 \\ 0 & 0 & GI_{zz} \end{bmatrix} \tag{3.14}$$

To summarize the complete set of the Cosserat equation for the static of a continuum

rod are:

$$\begin{aligned} \boldsymbol{p}_s &= \mathbf{R}\boldsymbol{v} \\ \mathbf{R}_s &= \mathbf{R}\hat{\boldsymbol{u}} \\ \boldsymbol{n}_s &= -\boldsymbol{f} \\ \boldsymbol{m}_s &= -\boldsymbol{l} - \boldsymbol{p}_s \times \boldsymbol{n} \end{aligned} \tag{3.15}$$

$$\begin{aligned} \boldsymbol{n} &= \mathbf{R}\mathbf{K}_{se}(\boldsymbol{v} - \boldsymbol{v}^*) \\ \boldsymbol{m} &= \mathbf{R}\mathbf{K}_{bt}(\boldsymbol{u} - \boldsymbol{u}^*) \end{aligned}$$

3.2.2 Dynamic equations

To apply the equation for the Cosserat Rod in the dynamics we have to recognize the time dependence. For this reason the variables we considered earlier are now time-dependent. For example, the center line is now described by $\boldsymbol{p}(t, s)$.

It is possible to define a variable for the speed of the local frame $\boldsymbol{q}$ described by the equation $\boldsymbol{q} = \mathbf{R}^T\boldsymbol{p}_t$. In the same way a variable for the angular velocity $\boldsymbol{\omega}$ is defined, which is described by the equation $\boldsymbol{\omega} = (\mathbf{R}^T\mathbf{R}_t)^\vee$. Deriving along the space it is possible to obtain the equation of $\boldsymbol{q}_s$ which can be written as:

$$\boldsymbol{q}_s = \mathbf{R}_s^T\boldsymbol{p}_t + \mathbf{R}^T\boldsymbol{p}_{ts} \tag{3.16}$$

To obtain $\boldsymbol{p}_{ts}$ it is sufficient to derive the equation $\boldsymbol{p}_s = \mathbf{R}\boldsymbol{v}$, that becomes:

$$\boldsymbol{p}_{ts} = \mathbf{R}_t\boldsymbol{v} + \mathbf{R}\boldsymbol{v}_t \tag{3.17}$$

So, substituting $\boldsymbol{p}_{ts}$ and $\boldsymbol{p}_t = \mathbf{R}\boldsymbol{q}$, $\mathbf{R}_t = \mathbf{R}\hat{\boldsymbol{\omega}}$ and $\mathbf{R}_s = \mathbf{R}\hat{\boldsymbol{u}}$, where the $\hat{\cdot}$ is the skew-matrix operator, is obtained the following equation:

$$\begin{aligned} \boldsymbol{q}_s &= \mathbf{R}_s^T\boldsymbol{p}_t + \mathbf{R}^T(\mathbf{R}_t\boldsymbol{v} + \mathbf{R}\boldsymbol{v}_t) \\ &= (\mathbf{R}\hat{\boldsymbol{u}})^T\mathbf{R}\boldsymbol{q} + \mathbf{R}^T(\mathbf{R}\hat{\boldsymbol{\omega}}\boldsymbol{v} + \mathbf{R}\boldsymbol{v}_t) \\ &= -\hat{\boldsymbol{u}}\boldsymbol{q} + \hat{\boldsymbol{\omega}}\boldsymbol{v} + \boldsymbol{v}_t \end{aligned} \tag{3.18}$$

To find $\boldsymbol{\omega}_s$ it is used the same process, but working with the skew-symmetric representation to simplify the solution process. Therefore:

$$\boldsymbol{\omega}_s = \mathbf{R}_s^T\mathbf{R}_t + \mathbf{R}^T\mathbf{R}_{ts} \tag{3.19}$$

To obtain $\mathbf{R}_{ts}$, the partial derivative of the equation $\mathbf{R}_s = \mathbf{R}\hat{\boldsymbol{u}}$ is solved to obtain:

$$\mathbf{R}_{ts} = \mathbf{R}_t\hat{\boldsymbol{u}} + \mathbf{R}\hat{\boldsymbol{u}}_t \tag{3.20}$$

So by replacing and simplifying:

$$\begin{aligned}\hat{\boldsymbol{\omega}}_s &= \mathbf{R}_s^T\mathbf{R}_t + \mathbf{R}^T(\mathbf{R}_t\hat{\boldsymbol{u}} + \mathbf{R}\hat{\boldsymbol{u}}_t) \\ &= (\mathbf{R}\hat{\boldsymbol{u}})^T\mathbf{R}\hat{\boldsymbol{\omega}} + \mathbf{R}^T(\mathbf{R}\hat{\boldsymbol{\omega}}\hat{\boldsymbol{u}} + \mathbf{R}\hat{\boldsymbol{u}}_t) \\ &= -\hat{\boldsymbol{u}}\hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\omega}}\hat{\boldsymbol{u}} + \hat{\boldsymbol{u}}_t\end{aligned} \tag{3.21}$$

Knowing that $(-\hat{\boldsymbol{u}}\hat{\boldsymbol{\omega}} + \hat{\boldsymbol{\omega}}\hat{\boldsymbol{u}})^\vee = -\hat{\boldsymbol{u}}\boldsymbol{\omega}$, finally it is obtained:

$$\boldsymbol{\omega}_s = -\hat{\boldsymbol{u}}\boldsymbol{\omega} + \boldsymbol{u}_t \tag{3.22}$$

Taking into account the dependence on time, the balance equations of force and moment must also be modified. The force balance along a discrete piece of rod will therefore be expressed by the equation:

$$\boldsymbol{n}(s+\delta) - \boldsymbol{n}(s) + \int_s^{s+\delta} \boldsymbol{f}(\sigma)d\sigma = \int_s^{s+\delta} \rho A\boldsymbol{p}_{tt}d\sigma \tag{3.23}$$

And deriving it is obtained:

$$\boldsymbol{n}_s = \rho A\boldsymbol{p}_{tt} - \boldsymbol{f} \tag{3.24}$$

where $\boldsymbol{p}_{tt}$ can be expressed as $\boldsymbol{p}_{tt} = \frac{\partial}{\partial t}(\mathbf{R}\boldsymbol{q}) = \mathbf{R}_t\boldsymbol{q} + \mathbf{R}\boldsymbol{q}_t = \mathbf{R}(\hat{\boldsymbol{\omega}}\boldsymbol{q} + \boldsymbol{q})$, obtaining:

$$\boldsymbol{n}_s = \rho A\mathbf{R}(\hat{\boldsymbol{\omega}}\boldsymbol{q} + \boldsymbol{q}_t) - \boldsymbol{f} \tag{3.25}$$

Following the same procedure for the momentum balance, the following equation is obtained:

$$\begin{aligned}\boldsymbol{m}(s+\delta) - \boldsymbol{m}(s) + \boldsymbol{p}(s+\delta)\times\boldsymbol{n}(s+\delta) - \boldsymbol{p}(s)\times\boldsymbol{n}(s) + \int_s^{s+\delta}[\boldsymbol{l}(\sigma) + \boldsymbol{p}(\sigma)\times\boldsymbol{f}(\sigma)]d\sigma \\ = \int_s^{s+\delta}\frac{\partial}{\partial t}(\rho A\boldsymbol{p}\times\boldsymbol{p}_t + \mathbf{R}\rho\mathbf{J}\boldsymbol{\omega})d\sigma\end{aligned} \tag{3.26}$$

Deriving:

$$\boldsymbol{m}_s = \frac{\partial}{\partial t}(\rho\mathbf{R}\mathbf{J}\boldsymbol{\omega}) - \hat{\boldsymbol{p}}_s\boldsymbol{n} - \boldsymbol{l} \tag{3.27}$$

Explaining the derivative with respect to time

$$\frac{\partial}{\partial t}(\rho \mathbf{R}\mathbf{J}\boldsymbol{\omega}) = \rho \mathbf{R}_t \mathbf{J}\boldsymbol{\omega} + \rho \mathbf{R}\mathbf{J}\boldsymbol{\omega}_t = \rho \mathbf{R}(\hat{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\omega}_t)$$

Thus, substituting:

$$\boldsymbol{m}_s = \rho \mathbf{R}(\hat{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega} + \mathbf{J}\boldsymbol{\omega}_t) - \hat{\boldsymbol{p}}_s \boldsymbol{n} - \boldsymbol{l} \tag{3.28}$$

Therefore, it is also necessary to make the constitutive equation of the material dependent on time, to do this we add not only the elastic properties of the material, but also those of damping. Using the Kelvin-Voigt type viscous damping, the equations become:

$$\begin{aligned} \boldsymbol{n} &= \mathbf{R}[\mathbf{K}_{se}(\boldsymbol{v} - \boldsymbol{v}^*) + \mathbf{B}_{se}\mathbf{v}_t] \\ \boldsymbol{m} &= \mathbf{R}[\mathbf{K}_{bt}(\boldsymbol{u} - \boldsymbol{u}^*) + \mathbf{B}_{bt}\mathbf{u}_t] \end{aligned} \tag{3.29}$$

Furthermore, it is possible to define the external distributed forces as the sum given by the term of the drag air resistance law and the term of the weight.

$$\boldsymbol{f} = -\mathbf{R}\mathbf{C}\boldsymbol{q} \odot |\boldsymbol{q}| + \rho A \boldsymbol{g} \tag{3.30}$$

where the symbol $\odot$ stands for the Hadamard product that is:

$$\boldsymbol{q} \odot |\boldsymbol{q}| = [q_1^2 sign(q_1) \quad q_1^2 sign(q_1) \quad q_1^2 sign(q_1)]^T \tag{3.31}$$

3.2.3 Semi-discretization in time

In order to obtain an ODE from the PDE with one spatial dimension and a time dimension it is necessary to discretize the time partial derivative. One of the ways to discretize is to use the BDFα method. So the derivative of a generic function f at the i-th time step can be written as:

$$f_t(t_i) = c_0 f(t_i) + c_1 f(t_{i-1}) + c_2 f(t_{i-2}) + d_1 f_t(t_{i-1}) \tag{3.32}$$

which can be rewritten as:

$$f_t(t_i) = c_0 f(t_i) + f^h(t_i) \tag{3.33}$$

where the term f^h represents the previous history of the function. In particular, for

the BDFα method the constants are given by the equations:

$$\begin{aligned} c_0 &= (1.5+\alpha)/[\delta(1+\alpha)] \\ c_1 &= -2/\delta \\ c_2 &= (0.5+\alpha)/[\delta(1+\alpha)] \\ d_1 &= \alpha/(1+\alpha) \end{aligned} \tag{3.34}$$

Where the variable α is in the range of $-0.5 \leq \alpha \leq 0$. By modifying the α value, we pass from the trapezoidal method with $\alpha = 0$, to the second order backward differentiation method formula BDF2 with $\alpha = 0$.

Therefore, using this method it is possible to rewrite the constitutive equations of the material as:

$$\begin{aligned} \boldsymbol{n} &= \mathbf{R}[\mathbf{K}_{se}(\boldsymbol{v}-\boldsymbol{v}^*) + \mathbf{B}_{se}(c_0\boldsymbol{v}+\boldsymbol{v}^h)] \\ \boldsymbol{m} &= \mathbf{R}[\mathbf{K}_{bt}(\boldsymbol{u}-\boldsymbol{u}^*) + \mathbf{B}_{bt}(c_0\boldsymbol{u}+\boldsymbol{u}^h)] \end{aligned} \tag{3.35}$$

So to use u and v in PDE equations we invert the terms, obtaining:

$$\begin{aligned} \boldsymbol{v} &= (\mathbf{K}_{se}+c_0\mathbf{B}_{se})^{-1}(\mathbf{R}^T\boldsymbol{n} + \mathbf{K}_{se}\boldsymbol{v}^* - \mathbf{B}_{se}\boldsymbol{v}^h) \\ \boldsymbol{u} &= (\mathbf{K}_{bt}+c_0\mathbf{B}_{bt})^{-1}(\mathbf{R}^T\boldsymbol{m} + \mathbf{K}_{bt}\boldsymbol{u}^* - \mathbf{B}_{bt}\boldsymbol{u}^h) \end{aligned} \tag{3.36}$$

Finally, therefore, it is possible to write the set of PDE equations of the Cosserat rod as an ODE as follows:

$$\begin{aligned} \boldsymbol{p}_s &= \mathbf{R}\boldsymbol{v} \\ \mathbf{R}_s &= \mathbf{R}\hat{\boldsymbol{u}} \\ \boldsymbol{n}_s &= \rho A\mathbf{R}(\hat{\boldsymbol{\omega}}\boldsymbol{q}+\boldsymbol{q}_t) - \boldsymbol{f} \\ \boldsymbol{m}_s &= \rho\mathbf{R}(\hat{\boldsymbol{\omega}}\mathbf{J}\boldsymbol{\omega}+\mathbf{J}\boldsymbol{\omega}_t) - \hat{\boldsymbol{p}}_s\boldsymbol{n} - \boldsymbol{l} \\ \boldsymbol{q}_s &= \boldsymbol{v}_t - \hat{\boldsymbol{u}}\boldsymbol{q} + \hat{\boldsymbol{\omega}}\boldsymbol{v} \\ \boldsymbol{\omega}_s &= \boldsymbol{u}_t - \hat{\boldsymbol{u}}\boldsymbol{\omega} \end{aligned} \tag{3.37}$$

$$
\begin{aligned}
\boldsymbol{v} &= (\mathbf{K}_{se} + c_0\mathbf{B}_{se})^{-1}(\mathbf{R}^T\boldsymbol{n} + \mathbf{K}_{se}\boldsymbol{v}^* - \mathbf{B}_{se}\boldsymbol{v}^h) \\
\boldsymbol{u} &= (\mathbf{K}_{bt} + c_0\mathbf{B}_{bt})^{-1}(\mathbf{R}^T\boldsymbol{m} + \mathbf{K}_{bt}\boldsymbol{u}^* - \mathbf{B}_{bt}\boldsymbol{u}^h) \\
\boldsymbol{v}_t &= c_0\boldsymbol{v} + \boldsymbol{v}^h \\
\boldsymbol{u}_t &= c_0\boldsymbol{u} + \boldsymbol{u}^h \\
\boldsymbol{q}_t &= c_0\boldsymbol{q} + \boldsymbol{q}^h \\
\boldsymbol{\omega}_t &= c_0\boldsymbol{\omega} + \boldsymbol{\omega}^h \\
\boldsymbol{f} &= -\mathbf{RC}\boldsymbol{q} \odot |\boldsymbol{q}| + \rho A\boldsymbol{g}
\end{aligned}
\tag{3.38}
$$

3.2.4 Spatial Integration

To integrate the ODE equation into the space dimension, it is possible to use the Runge-Kutta method. In this case the fourth order method is used, abbreviated as RK4.

In particular, for the Cosserat ODE the integration is in the space domain. The general equation assume the form $f_s(s_i) = f(f(s_i), f^h(s_i))$, where f^h is the term that represents the condition of f occurred in the past with respect to time, as shown in (3.32) and (3.33). Therefore, the integration occurs according to the following algorithm:

$$
\begin{aligned}
k_1 &= f\left(f(s_i), f^h(s_i)\right) \\
k_2 &= f\left(f(s_i) + k_1\frac{ds}{2}, \frac{f^h(s_i) + f^h(s_{i+1})}{2}\right) \\
k_3 &= f\left(f(s_i) + k_2\frac{ds}{2}, \frac{f^h(s_i) + f^h(s_{i+1})}{2}\right) \\
k_4 &= f\left(f(s_i) + k_3 ds, f^h(s_{i+1})\right) \\
f(s_{i+1}) &= f(s_i) + \frac{ds}{6}[k_1 + 2(k_2 + k_3) + k_4]
\end{aligned}
\tag{3.39}
$$

3.2.5 Optimization Method

The Levenberg-Marquardt algorithm (LMA) is an optimization algorithm used for solving problems in the form of nonlinear least squares. LMA is an iterative algorithm, in which the solution update vector at each iteration is given by an interpolation between the Gauss-Newton algorithm and the gradient descent method. LMA can be regarded as a trust region version of the Gauss-Newton algorithm, compared to which it is more robust but, in general, slightly slower.

This algorithm was not developed for this application, but the matlab @fsolve function was simply used. This function is very sophisticated and efficient. It allows you to use

different algorithms, and set the desired constraints.

3.2.6 Shooting Method

The shooting method is used to solve the problem with boundary conditions. To solve the nonlinear system,the vector of guessed values, and the function to be solved must be provided to the function @fsolve.

Usually, for one end of the rod is known the position, orientation, velocity and angular velocity, while forces and moments are not known.

The purpose of the spatial integration will be then to know all these quantities for the opposite end of the rod.

To solve the problem of unknown forces and moments these values are initially guessed and given to the optimization algorithm.

To obtain convergence more conditions must be given to the optimization algorithm, they are called residuals. For a PCR generally are the balance of forces, the balance of moments, and the geometric relationships between the rods and the end-effector.

For example for a cantilever rod, the boundary equations are $\boldsymbol{p}(t_i, 0) = \boldsymbol{p}_0$, $\mathbf{R}(t_i, 0) = \mathbf{R}_0$, $\boldsymbol{q}(t_i, 0) = \mathbf{0}$ and $\boldsymbol{\omega}(t_i, 0) = \mathbf{0}$, while the guessing values are $\boldsymbol{n}(t_i, 0)$ and $\boldsymbol{m}(t_i, 0)$. The residuals are given by the balance of the forces and moments on the tip of the rod.

The algorithm therefore is schematized like shown in Figure 3.4. In this algorithm are present three different loops each one inside of the others. The most internal loop is the integration in space; to do this the Runge-Kutta 4th algorithm is used to determinate the state of the rod along the space. The intermediate loop is handled by the optimization algorithm; in order to find the unknown parameters, that in this case are the forces and moments. Finally, the outermost loop is used to advance in time.

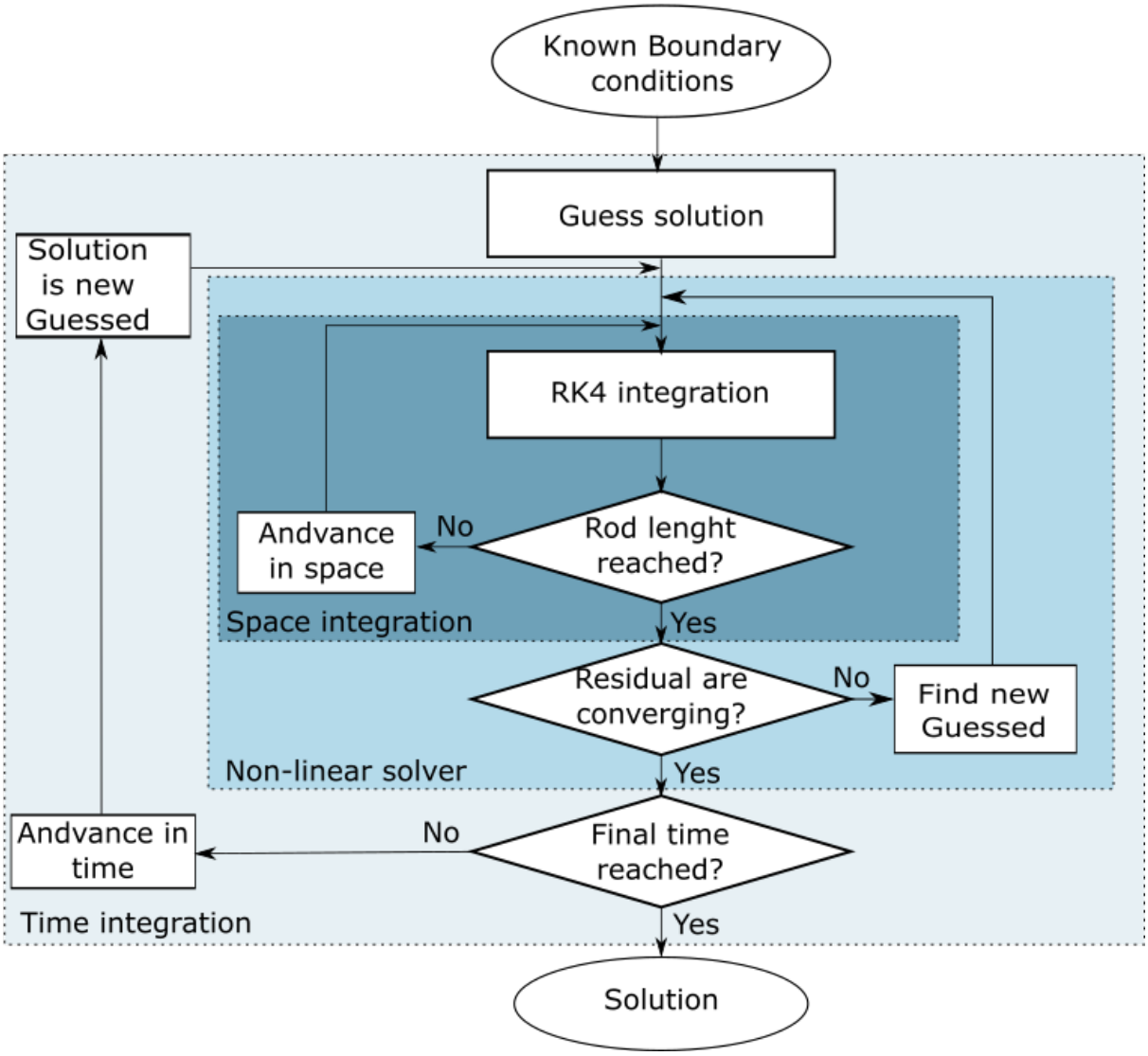


Figure 3.4: Shooting algorithm schema

Chapter 4

Simulations

4.1 Cantilever Rod

4.1.1 Model

Figure 4.1: Diagram of the cantilever rod simulation

The simplest case that can be used to verify dynamics for continuum bodies is a cantilever rod. In particular the case where a rod is clamped at its proximal end and free at its distal end. At instant zero, a vertical force F_e with downward direction is applied on the rod tip. At the next instant the force is released, so that the free oscillation of the rod can be observed. It is possible to see the diagram of the cantilever rod in Figure 4.1

The first boundary condition of the problem is that the right side of the rod is clamped. So, the proximal end's position and orientation are fixed, and the velocity and the angular velocity are null: $\boldsymbol{p}(s=0,t)=\boldsymbol{p}_0$, $\mathbf{R}(s=0,t)=\mathbf{R}_0$, $\boldsymbol{q}(s=0,t)=\mathbf{0}$ and $\boldsymbol{\omega}(s=0,t)=\mathbf{0}$.

The second boundary condition is the force $\boldsymbol{F}_e=-2\boldsymbol{z}$ N applied on the tip of the rod at the instant $t=0$, and released at the instant $t=0^+$. This simulation is performed in order to observe the free motion.

To have a convergence of the optimization algorithm the guessed are the values of the force and moment on the clamped side of the rod, so $\boldsymbol{n}(0,t)$ and $\boldsymbol{m}(0,t)$. The residuals are the balance equations on the tip of the rod, so $\boldsymbol{n}(s=L,t)=\boldsymbol{F}_e$ and $\boldsymbol{m}(s=L,t)=0$.

The values used for the simulations can be found in Table 4.1, where the rod has length L, a circular cross section of radius r, Young modulus E, shear modulus G, and density ρ. And the simulation is performed with null damping matrices.

Data	**Value**	**Unit**
L	400	mm
r	1	mm
E	207	GPa
G	79.61	GPa
ρ	8000	$kg * m^{-3}$
F_{tip}	2	N

Table 4.1: Numerical data used for the cantilever rod

4.1.2 Simulations

For the first simulation of the cantilever beam, spacial and time integration parameters need to be set. The duration of the simulation is set at 1 s with a sampling time of $dt = 5\ ms$. The beam is discretized in space (N) with 150 points. For the time discretization BDF-α the constant is $\alpha = -0.4$.

Since the damping of the rods is zero, what we would expect from the simulation would be a constant rod tip oscillation. However, as can be seen in Figure 4.2 we have a convergence in the oscillations. Since there is no physical damping, it can be deduced that this behavior is due to a numerical damping obtained by the semi-discretizations in time.

To evaluate this damping and to determine how the different parameters are influencing the simulation we carried out two sets of simulation. The first set is with the values of the sampling time dt ranging from 0.1 ms up to 10 ms, for values of α of 0, -0.2, -0.4, -0.5, and a constant number of spatial discretization N of 150 points per rod. A second set of simulations is carried out to understand how the damping varies as N varies. So, a range of N varying from 10 to 400 points per rod is used, using the same α, and a constant sampling time of 10 ms.

Before proceeding, we tried to define the damping numerically. So for the simulation we identify the maxima peaks of the function. We will call the value of the first maximum X_0 and the value of the last one as X_n. The number of oscillations between the extreme maxima is n. So we obtain the Log-dec rate as

$$\delta = \frac{1}{n} ln \left(\frac{X_0}{X_n} \right) \tag{4.1}$$

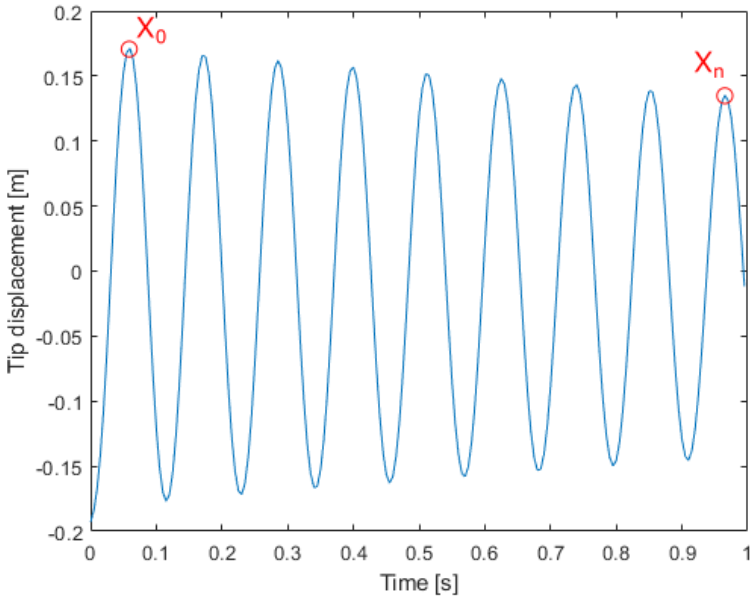


Figure 4.2: Oscillation obtained from the simulation for the tip of the cantilever rod

So the value of the damping is given by

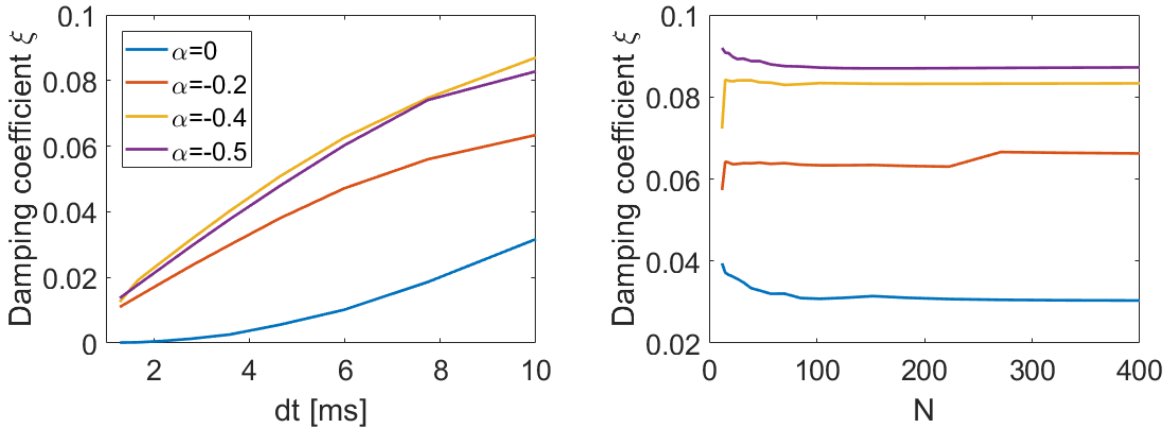
$$\xi = \frac{\delta}{(4\pi^2 + \delta^2)^{0.5}} \tag{4.2}$$

As can be seen in Figure 4.3a the damping increases as dt increases, this can be explained by the fact that the discretization has a larger error as the sampling time increases. In addition, for values of dt equal to 1 ms or less, the simulation stops or gives results with high errors; for example in Figure 4.3a the yellow curve undergoes a sharp deviation, just due to computational problem. This phenomenon occurs because the Levenberg-Marquardt algorithm reaches the maximum number of iterations set. To obtain a better result therefore it would be necessary to request a higher value from the algorithm, but it would be obtained a simulation time too high.

From the plots in Figure 4.3b, it can be seen that by overcoming an initial transition phase the algorithm is not influenced by the value of discretization in space (N). So, this is a proof that the numerical damping is due to the temporal semi-discretization and not the spatial discretization.

For the same sets of simulations, the simulation time was measured. From Figure 4.4a it is possible to observe how the simulation time is proportional to the inverse of an exponential ($T_s = n^{1/dt}$) to the sampling time (dt), while it is directly proportional to the number of discretizations (N), as we can see in Figure 4.4b.

So to have a good compromise between speed of simulation and the residual of the

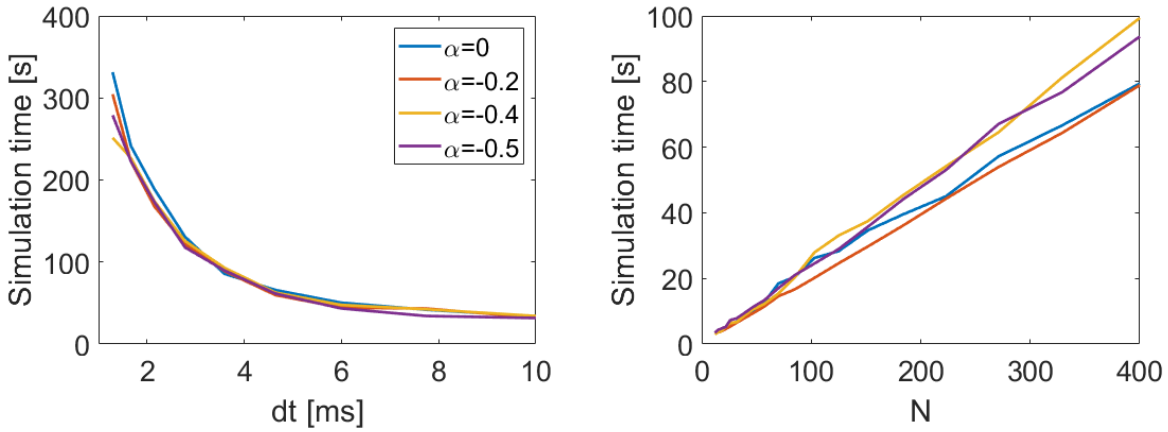


(a) Damping values for different dt and α, using N=150

(b) Damping values for different N and α, using dt=10ms

Figure 4.3: Damping variation with different values of dt and N

shooting algorithm, it is not advisable to use too low values of dt. Regarding N, it is sufficient to use a value not too small, because after a certain value the result is rather constant, but we are going to influence the simulation time in a directly proportional way.



(a) Simulation time values for different dt and α, using $N = 150$

(b) Simulation time values for different N and α, using $dt = 10ms$

Figure 4.4: Simulation time variation with different values of dt and N

4.1.3 Conclusion

For a good simulation the value of dt cannot be too small in order to avoid a too long time of simulation; and at the same time it cannot be too high in order to not increase the numerical damping. Regarding N it is suggested to use a value not too high, because after a certain value it does not influence the damping, but it increases the simulation

time. Regarding α it is suggested to use values close to 0, because a lower damping is obtained.

4.2 Stewart-Gough platform

The first robot analyzed is the continuum version of the Stewart-Gough platform presented in the chapter 2. This structure is the only one that have been studied regarding to its dynamic behavior using a Cosserat-based model [22]. We decided to study in a first time this structure in order to compare the results and understand if our proposed model is indeed effective comparing to theirs.

4.2.1 Model

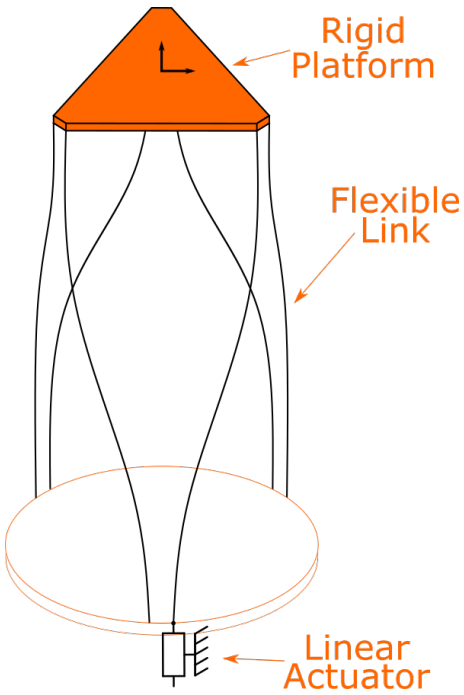


Figure 4.5: Schema of the Continuum Stewar-Gough robot

The continuum version of the Stewart-Gough robot consists of 6 continuum links connected in parallel to an end-effector platform. The rods pass through holes in the platform at the base and then join the end-effector. The actuation of the platform is obtained by lengthening or reducing the length of the rod. All the coordinates used can be referenced inside a global frame that corresponds to the center of the base plate. It is possible to observe the schema of this robot in Figure 4.5.

To solve the dynamics problem of a PCR therefore, it is necessary to model the robot rods with the dynamic equations (3.37), and solve the boundary conditions using equations

of dynamics and geometric relations that contraint the rods to the end-effector.

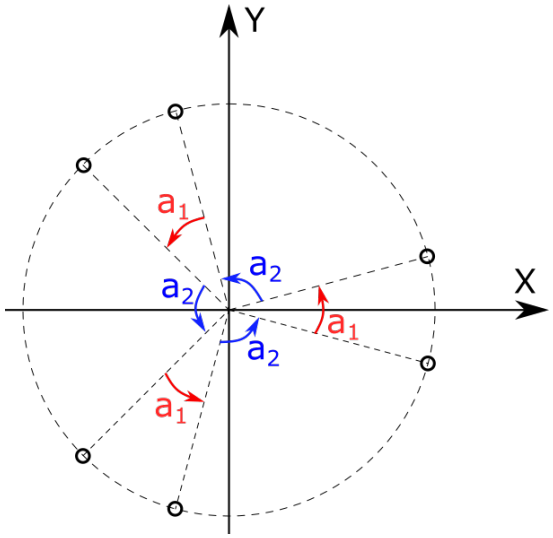


Figure 4.6: Schema of the hole of the base for a Continuum Stewar-Gough robot

Therefore the holes where the rods pass can be described by cylindrical coordinates using the angles showed in the schema of Figure 4.6. The radius r_e is constant , and the height is zero. So we just need to express the angle θ for each hole as:

$$\theta_i^B = \begin{cases} a_1/2 & i = 1 \\ \theta_{i-1}^B + a_1 & i \in \{2,4,6\} \\ \theta_{i-1}^B + 120° - a_1 & i \in \{3,5\} \end{cases} \tag{4.3}$$

Therefore by transforming into Cartesian coordinates is is obtained the position as:

$$\boldsymbol{p}_i(0) = r_e \begin{bmatrix} cos\theta_i^B & sin\theta_i^B & 0 \end{bmatrix}^T \tag{4.4}$$

To represent the points of attachment of the rods to the end-effector, the same method is used, considering the center of the end effector as the origin. In this case the angles will be:

$$\theta_i^B = \begin{cases} (120° - a_1)/2 & i = 1 \\ \theta_{i-1}^B + 120° - a_1 & i \in \{2,4,6\} \\ \theta_{i-1}^B + a_1 & i \in \{3,5\} \end{cases} \tag{4.5}$$

Therefore in Cartesian coordinates:

$$\boldsymbol{r}_i = r_e \begin{bmatrix} cos\theta_i^B & sin\theta_i^B & 0 \end{bmatrix}^T \tag{4.6}$$

In both cases to describe the robot we simply need the radius r_e, and the angle a_1.

It is therefore possible to find all the parameters of the robot as in Table 4.2, where L is the length of the robot rods, r is the radius of their the circular sections, E is Young's modulus, G is the shear modulus, and ρ is the density. Next, the parameters of the two identical cylindrical platforms are listed: mass m_e, inertia along their respective axes I_x, I_y, and I_z, height h_e, and radius r_e.

Data	**Value**	**Unit**
L	170	mm
r	1	mm
E	207	GPa
G	79.61	GPa
ρ	8000	$kg * m^{-3}$
m_e	92.1	g
h_e	5	mm
r_e	8.7	mm
a_1	20	deg
I_x, I_y	1.91e-4	$kg * m^2$
I_z	3.81e-4	$kg * m^2$

Table 4.2: Data used for the Stewart-Gough Robot

Residual equations

To solve the boundary conditions for PCR, the equations that give a relationship between the rods and the end effector must be found. Specifically, the equations are:

- a balance of the forces acting on the end effector.
- a balance of the moments acting on the end effector
- a geometric relation between the position of each tip of the rod and the position of the end effector
- a geometric relation between the orientation of each rod and the orientation of the end effector.

Therefore, in solving the boundary condition problem for each rod, the geometric constraint that relates the tip of the rod with the end effector is:

$$\boldsymbol{p}_e(t) + \mathbf{R}_e(t)\boldsymbol{r}_i = \boldsymbol{p}_i(t, L_i) \quad \text{for} \quad i = 1...6 \tag{4.7}$$

Furthermore it is assumed that the orientation of the rod at the tip must be the same as the end-effector, therefore:

$$\mathbf{R}_e(t) = \mathbf{R}_i(t, L_i) \quad \text{for} \quad i = 1...6 \tag{4.8}$$

Furthermore, the force balance on the end effector is used:

$$\boldsymbol{F}_e(t) + m_e\boldsymbol{g} - \sum_{i=1}^{6} \boldsymbol{n}_i(t, L_i) = m_e\boldsymbol{a}_e(t) \tag{4.9}$$

where $\boldsymbol{F}_e$ is the vector of external forces, m_e is the mass of the end effector, $\boldsymbol{g}$ the gravity, and $\boldsymbol{a}_e$ is the acceleration of the end effector.

The moment balance equation in the global frame is:

$$\boldsymbol{M}_e(t) + \sum_{i=1}^{6} \boldsymbol{m}_i(t, L_i) + [\mathbf{R}_e(t)\boldsymbol{r}_i] \times \boldsymbol{n}_i(t, L_i) = \tag{4.10}$$

$$\mathbf{R}_e(t)\mathbf{J}_e\boldsymbol{R}_e^T(t)\boldsymbol{\omega}_{et}(t) + \hat{\omega}_e(t)\mathbf{R}_e(t)\mathbf{J}_e\boldsymbol{R}_e^T(t)\boldsymbol{\omega}_e(t) \tag{4.11}$$

where $\boldsymbol{M}_e$ are the external moments, $\mathbf{J}_e$ is the inertia tensor of the end effector, $\boldsymbol{\omega}_e$ is the angular velocity in the global frame, and $\boldsymbol{\omega}_{te}$ is the angular acceleration in the global frame.

For this problem our guessed are the position and orientation of the end effector, and the forces and moments inside the rods. So the guessed vector will be:

$$G = \begin{bmatrix} \boldsymbol{p}_e & \boldsymbol{k}_e & \boldsymbol{n}_1 & \boldsymbol{m}_1 & \dots & \boldsymbol{n}_6 & \boldsymbol{m}_6 \end{bmatrix} \tag{4.12}$$

where $\boldsymbol{k}_e$ is the vector describing the rotation obtained with the Rodrigues formula. In total, therefore, there are 42 unknowns, which are solved using the following vector as residual error, obtained from the equation previously founded.

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{E}^F = \boldsymbol{F}_e + m_e(\boldsymbol{g} - \boldsymbol{a}_e) - \sum_{i=1}^{6} \boldsymbol{n}_i \\ \boldsymbol{E}^M = \boldsymbol{M}_e(t) - \mathbf{R}_e(\mathbf{J}_e\boldsymbol{\omega}_{et} + \hat{\boldsymbol{\omega}}_e\mathbf{J}_e\boldsymbol{\omega}_e) + \sum_{i=1}^{6} \{\boldsymbol{m}_i(t, L_i) + [\mathbf{R}_e(t)\boldsymbol{r}_i] \times \boldsymbol{n}_i(t, L_i)\} \\ \boldsymbol{E}_1^p = \boldsymbol{p}_e + \mathbf{R}_e\boldsymbol{r}_1 - \boldsymbol{p}_1 \\ \boldsymbol{E}_1^R = \left(\mathbf{R}_e^T\mathbf{R}_1 - \mathbf{R}_e\mathbf{R}_1^T\right)^V \\ \vdots \\ \boldsymbol{E}_6^p = \boldsymbol{p}_e + \mathbf{R}_e\boldsymbol{r}_6 - \boldsymbol{p}_6 \\ \boldsymbol{E}_6^R = \left(\mathbf{R}_e^T\mathbf{R}_6 - \mathbf{R}_e\mathbf{R}_6^T\right)^V \end{bmatrix} \tag{4.13}$$

The dynamic terms of the end-effector are found using the same discretization method illustrated for the Cosserat equations, therefore:

$$\begin{aligned} \boldsymbol{v}_e &= c_0\boldsymbol{p}_e + \boldsymbol{p}_e^h \\ \boldsymbol{a}_e &= c_0\boldsymbol{v}_e + \boldsymbol{v}_e^h \\ \mathbf{R}_e &= c_0\mathbf{R}_e + \mathbf{R}_e^h \\ \boldsymbol{v}_e &= c_0\boldsymbol{p}_e + \boldsymbol{p}_e^h \\ \boldsymbol{\omega}_{et} &= c_0\boldsymbol{\omega}_e + \boldsymbol{\omega}_e^h \end{aligned} \tag{4.14}$$

4.2.2 Simulations

For this robot a simulation without external perturbation was performed in order to understand the reliability of the algorithm with the different values of the sampling time dt, of the constant of the temporal discretization α, and the number N of spacial discretization.

A set of simulations with α equal to 0, -0.2, -0.4 and -0.5, dt equal to 5ms and 10ms, and N equal to 5, 10, 50, 85, 170 were performed for this purpose.

For these simulations we analyzed the position of the end effector, and the noise that is obtained despite there are no external perturbations

However, to be able to compare these data, we have identified a parameter that quantifies the noise present in the simulation.

The first step to find the parameter is to find the Root Mean Square Error (RMSE) for each coordinate (x,y,z) as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{T}\left(\bar{f} - f_i\right)^2}{T}} \tag{4.15}$$

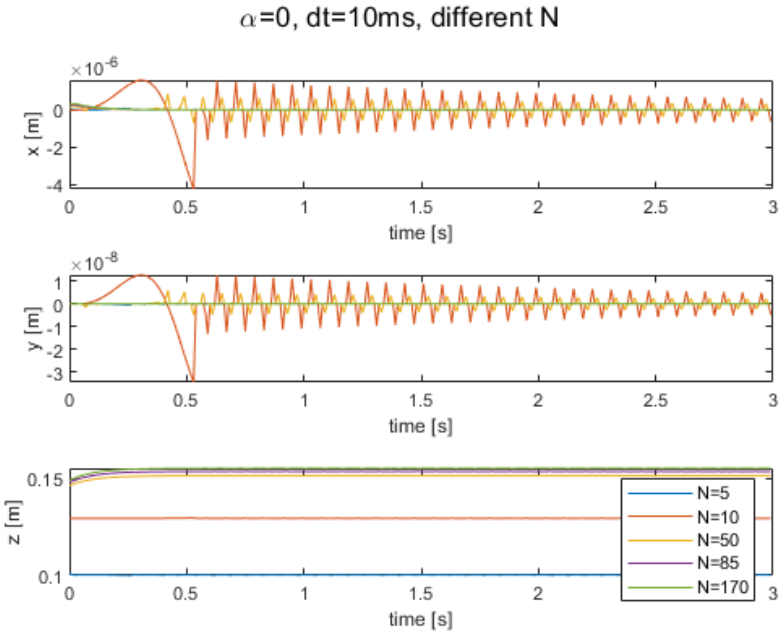


Figure 4.7: End-effector position assumed for $\alpha = -0.2$, $dt = 10\ ms$ and different values of N

where f_i is the value of the position of the end effector at the i-th instant, T is the number of instants in which the position was calculated and $\bar{f}$ is the average value of the position taken by the end effector, calculated as:

$$\bar{f} = \frac{\sum_{i=1}^{T} f_i}{T} \tag{4.16}$$

So, calculating the $RMSE$ for each axis we obtain $RMSE_x$, $RMSE_y$, $RMSE_z$. From these parameters we obtain the parameter that we will call "Noise" using the equation:

$$Noise = \frac{log_{10}(RMSE_x) + log_{10}(RMSE_y) + log_{10}(RMSE_z)}{3} \tag{4.17}$$

The results obtained are reported in Table 4.3 and table 4.4. What can be observed is that the average noise decreases as N increases, but increases as dt increases. This can be deduced since when we are reducing the temporal discretization the number of iterations required to have good convergence of the algorithm is too high, just as we went into detail in the simulation of the cantilever rod, and as can be observed in the Figure 4.4a. Also if we look at Figure 4.7 it is possible to notice different values for the platform along the z axis. This happens because for too small values of the spatial discretization, such as N=5 or N=10, the approximation is too large, and thus the results have a large error. However, by increasing the value of N it is possible to observe how the values tend to converge.

α\N	5	10	50
0	-3.80	-3.61	-5.91
-0.2	-4.27	-3.60	-4.15
-0.4	-5.71	-4.28	-3.76
-0.5	-4.36	-4.29	-3.53

Table 4.3: Noise values for the different simulations with $dt = 5ms$. Red means high noise and blue means low noise

α\N	5	10	50	85	170
0	-8.10	-6.42	-6.71	-8.54	-8.34
-0.2	-6.27	-5.85	-6.56	-8.42	-8.24
-0.4	-6.39	-5.63	-6.53	-8.35	-8.00
-0.5	-5.84	-5.40	-5.86	-5.96	-6.32

Table 4.4: Noise values for the different simulations with $dt = 10ms$. Red means high noise and blue means low noise

4.2.3 Conclusion

For a parallel continuum robot the noise due to the simulation residual depends strongly on the number of spacial discretizations, therefore it is advisable to choose a high value. It is advisable to use a larger sampling time to avoid a high number of iterations for the optimization algorithm. The parameter α does not influence the simulation too much, but the best results are obtained for $\alpha = 0$, or $\alpha = -0.2$.

4.3 Triskèle bot

4.3.1 Model

In the Triskele-bot the revolute joints and the intermediate rigid bodies have been replaced by a flexible body (F). This PCR can be abbreviated as 3-$\underline{\text{P}}$F, and is called Triskele-bot because the geometry resembles to a Triskele symbol. The degrees of freedom of the platform remain 3 just like in the rigid configuration. Figure 4.8 represents a kinematic diagram of the Triskele-bot.

Going into detail for each link in the Figure 4.9, the flexible element is considered as a 1-dimensional rod defined by the curvilinear abscissa s along the rod, and for each point it is possible to identify a reference system with position $p(s)$ and rotation $\theta(s)$, referred to the global reference system with center in O.

- Q_i represents the position of the actuator, and ϕ_i its orientation with respect to the

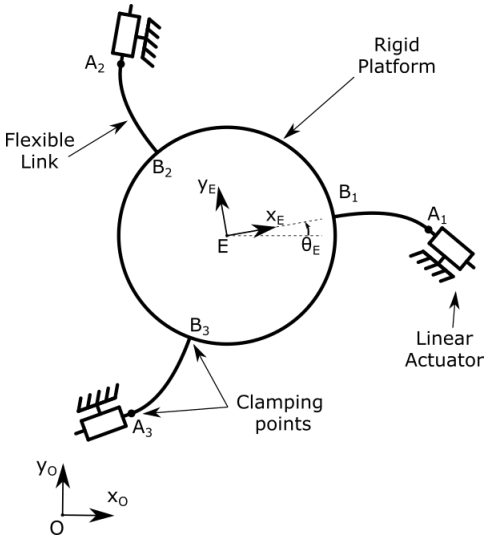


Figure 4.8: Kinematic diagram of the Triskele-bot

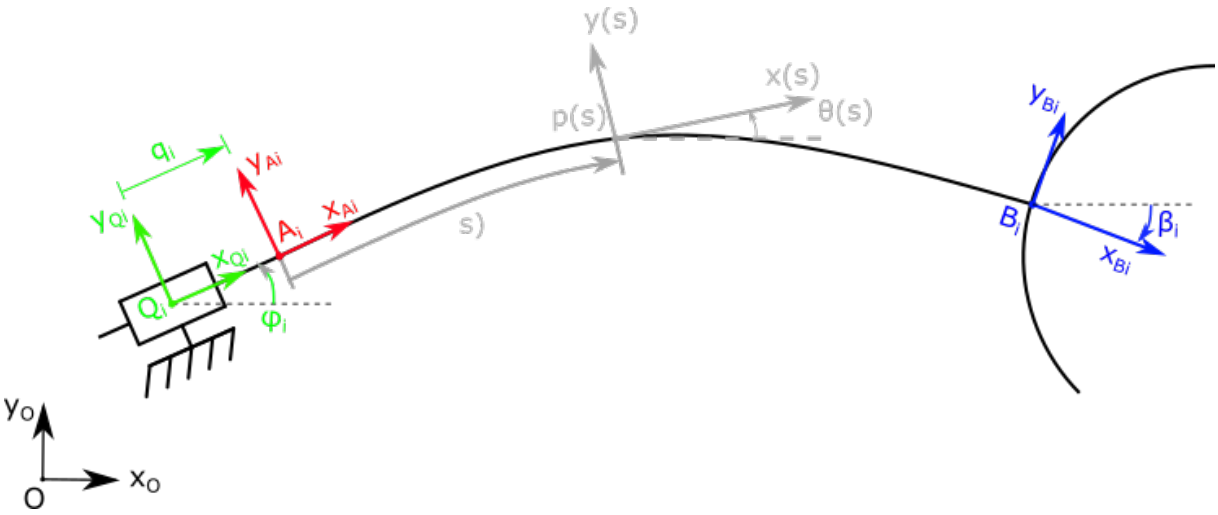


Figure 4.9: Detailed schema for the flexible rod

global reference system. To find the position of Q_i it is possible to use the cylindrical coordinates with a constant radius r_a and an angle α_i.

- A_i is the point where the flexible element is linked with the actuator. It is translated by q_i with respect to the point Q_i, in the direction Q_{xi}. So, the angle of rotation of its orientation is ϕ_i. This point corresponds to be in the position $p(s = 0)$.

- B_i is the point where the flexible element is linked to the platform, β_i is the rotation angle with respect to the global reference system. This point corresponds to $s = L$, where L is the length of the rod.

- E is the center point of the platform. θ_E is the rotation of the platform with respect to the global reference system.

The numerical values used for the simulation are shown in Table 4.5, where L is the length of the rod, r is its radius, E is the Young's modulus, G is the shear modulus and ρ is the density. τ is the parameter that represents the relation between the stiffness matrix and the damping matrix as explained for (3.6). The properties of the end-effector are: mass m_e, radius r_e, and thickness l_e.

The initial orientation of the end-effector is found from the angle θ_e. r_a as explained before is radius to position the actuators with the cylindrical coordinates, with the angles γ_i as in Table 4.6, and the angle of orientation of the actuators ϕ_i. In the same table it is possible to find the angle of orientation β_i of the rod in the point of clamping with the end effector.

Data	**Value**	**Unit**
L	32	mm
r	0.0625	mm
E	69.2	GPa
G	29.58	GPa
ρ	2203	$kg * m^{-3}$
τ	0	-
m_e	40	g
l_e	0.1	mm
r_e	50.8	mm
θ_e	10	deg
r_a	80	mm

Table 4.5: Numerical data used for the Triskele-bot

i	$\boldsymbol{\phi_i}$	$\boldsymbol{\gamma_i}$	$\boldsymbol{\beta_i}$
1	140°	-30°	10°
2	20°	210°	130°
3	-100°	90°	-120°

Table 4.6: Angle of calibration for the Triskele-bot

Residual equations

To solve the rod boundary conditions, the procedure is the same as we used previously for the Stewart-Gough robot.

So the first equation is the geometric constraint that relates the tip of the rods to the position of the end-effector. Therefore:

$$\boldsymbol{p}_e(t) + \mathbf{R}_e(t)\boldsymbol{r}_i = \boldsymbol{p}_i(t, L_i) \quad \text{for} \quad i = 1...3 \tag{4.18}$$

The end point of the rod is rotated according to a rotation $\mathbf{R}_{fi}(t)$ with respect to the orientation of the end-effector. So the orientation equation will be:

$$\mathbf{R}_e(t) = \mathbf{R}_{fi}(t)\mathbf{R}_i(t, L_i) \quad \text{for} \quad i = 1...3 \tag{4.19}$$

Furthermore, the force balance on the end-effector is:

$$\boldsymbol{F}_e(t) + m_e\boldsymbol{g} - \sum_{i=1}^{3} \boldsymbol{n}_i(t, L_i) = m_e\boldsymbol{a}_e(t) \tag{4.20}$$

where $\boldsymbol{F}_e$ is the vector of external forces, m_e is the mass of the end effector, $\boldsymbol{g}$ the gravity, and $\boldsymbol{a}_e$ is the acceleration of the end effector.

The moment balance equation in the global frame is:

$$\boldsymbol{M}_e(t) + \sum_{i=1}^{3} \boldsymbol{m}_i(t, L_i) + [\mathbf{R}_e(t)\boldsymbol{r}_i]^{\wedge}\boldsymbol{n}_i(t, L_i) = \mathbf{R}_e(t)\mathbf{J}_e\boldsymbol{R}_e^T(t)\boldsymbol{\omega}_{et}(t) + \hat{\omega}_e(t)\mathbf{R}_e(t)\mathbf{J}_e\boldsymbol{R}_e^T(t)\boldsymbol{\omega}_e(t) \tag{4.21}$$

.

where $\boldsymbol{M}_e$ are the external moments, $\mathbf{J}_e$ is the inertia tensor of the end effector, $\boldsymbol{\omega}_e$ is the angular velocity in the global frame, and $\boldsymbol{\omega}_{te}$ is the angular acceleration in the global frame.

The robot's end-effector has planar motion so there is no movement along z, or rotations around the x and y axes. Therefore it is possible to simplify the model by using the force balance only along x and y, and the momentum balance only along z. So, it is possible to write the residuals as:

$$\boldsymbol{E} = \begin{bmatrix} (\%x, y)\boldsymbol{E}^F = \boldsymbol{F}_e + m_e(\boldsymbol{g} - \boldsymbol{a}_e) - \sum_{i=1}^{3} \boldsymbol{n}_i \\ (\%z)\boldsymbol{E}^M = \boldsymbol{M}_e(t) - \mathbf{R}_e(\mathbf{J}_e\boldsymbol{\omega}_{et} + \hat{\boldsymbol{\omega}}_e\mathbf{J}_e\boldsymbol{\omega}_e) + \sum_{i=1}^{3} \{\boldsymbol{m}_i(t, L_i) + [\mathbf{R}_e(t)\boldsymbol{r}_i]\hat{}\,\boldsymbol{n}_i(t, L_i)\} \\ \boldsymbol{E}_1^p = \boldsymbol{p}_e + \mathbf{R}_e\boldsymbol{r}_1 - \boldsymbol{p}_1 \\ \boldsymbol{E}_1^R = \left(\mathbf{R}_e^T(\mathbf{R}_{f1}(t)\mathbf{R}_1(t, L_1)) - \mathbf{R}_e(\mathbf{R}_{f1}(t)\mathbf{R}_1(t, L_1))^T\right)^V \\ \vdots \\ \boldsymbol{E}_3^p = \boldsymbol{p}_e + \mathbf{R}_e\boldsymbol{r}_3 - \boldsymbol{p}_3 \\ \boldsymbol{E}_3^R = \left(\mathbf{R}_e^T(\mathbf{R}_{f3}(t)\mathbf{R}_3(t, L_3)) - \mathbf{R}_e(\mathbf{R}_{f3}(t)\mathbf{R}_3(t, L_3))^T\right)^V \end{bmatrix} \tag{4.22}$$

For this problem the guessed are the position and orientation of the end effector, and the forces and moments inside the rods. So the guessed vector will be:

$$G = \begin{bmatrix} \boldsymbol{p}_e & \boldsymbol{k}_e & \boldsymbol{n}_1 & \boldsymbol{m}_1 & \boldsymbol{n}_2 & \boldsymbol{m}_2 & \boldsymbol{n}_3 & \boldsymbol{m}_3 \end{bmatrix} \tag{4.23}$$

4.3.2 Simulations

The simulation carried out with the Triskele-bot, consists in imposing a trajectory to the end-effector and elaborate the inverse dynamics, to find the trajectory to impose to the actuators. Then, the result with the forward dynamics to observe if the end-effector actually follows the same path previously given .

Therefore, the goal of these simulations is to understand if the algorithm is valid for a forward dynamics problem and for a inverse dynamics problem.

This simulation is performed with the sampling time dt equal to $1ms$, of the constant of the temporal discretization α equal to -0.2, and the number N of spacial discretization of 50. The trajectory performed is a circular shape with an initial linear path. The radius of the cyrcle is 5 mm, and the velocity for the linear path is 15.7 $mm \cdot s^{-1}$, while for the circular part the angular velocity was 3.14 $rad \cdot s^{-1}$.

The purpose of this simulation is to use the direct and inverse dynamics. From Figure 4.10 it is possible to observe the plot on the x-y axis at each point of the performed trajectory.

The trajectory calculated for the actuators can be observed in Figure 4.11a.

In Figure 4.11b it is possible to observe the trajectories for both x and y axis. The trajectory imposed to the inverse dynamics is plotted in blue, while the trajectory calculated following the forward dynamics is reported in red. It is also plotted the error between the two curves calculated for each instant of time as the distance in the plane between the

two trajectories. It results that during the straight motion the error grows linearly, while during the circular motion it variate only a little. Therefore the error seems to depend on the distance covered, so for a displacement of 5 mm was obtained an error of 0.088 mm which is equivalent to 1.76% of the movement.

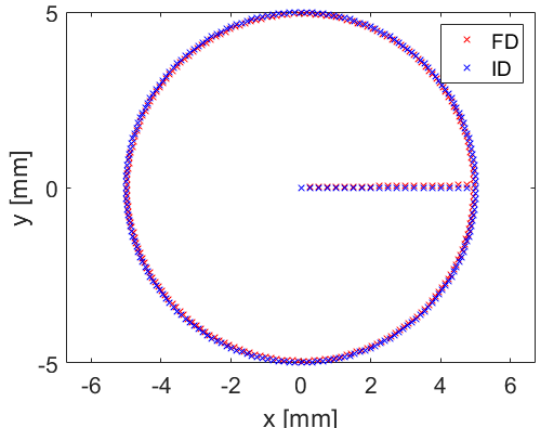
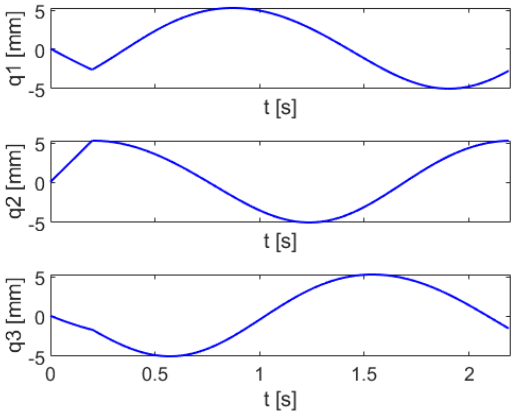
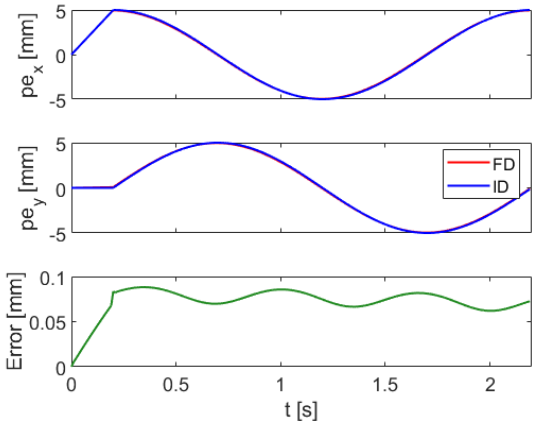


Figure 4.10: Trajectory of the end-effector performed by the inverse dynamic (red) and by the forward dynamic (blue)



(a) Calculated trajectories of the actuators



(b) Displacement of the end-effector performed by the inverse dynamic (red) and by the forward dynamic (blue), and error (green) between the two curves

Figure 4.11: Inverse and Forward Simulation results

Chapter 5

Experiment

The experiments were performed using the Triskele-bot. This chapter will initially describe the hardware structure of the robot and the instrumentation to obtain the measurements. Then the experiments performed are illustrated. The first one is done by applying a force to the platform, and observing the free oscillations. The second is done by moving the actuators.

5.1 Experimental Setup

This section describes the robot and the measurement setup that are both on an anti-vibration table which isolates the experiment from the vibrations coming from the ground.

5.1.1 Robot

The Triskele-bot consists of a central platform to which are attached flexible elements that transmit forces and displacements provided by actuators. As explained before, the resulting planar displacements are two translations and a rotation. The height of the platform is kept constant thanks to a vacuum preloaded air bearing.

Platform

The platform is the central element of the robot and thus the end-effector. The platform is pushed and pulled by the flexible elements in order to perform planar movements. The platform is a silicon wafer with a diameter of 101.6 mm. The diameter was chosen so as to have a large enough surface area on which the air bearing can act.

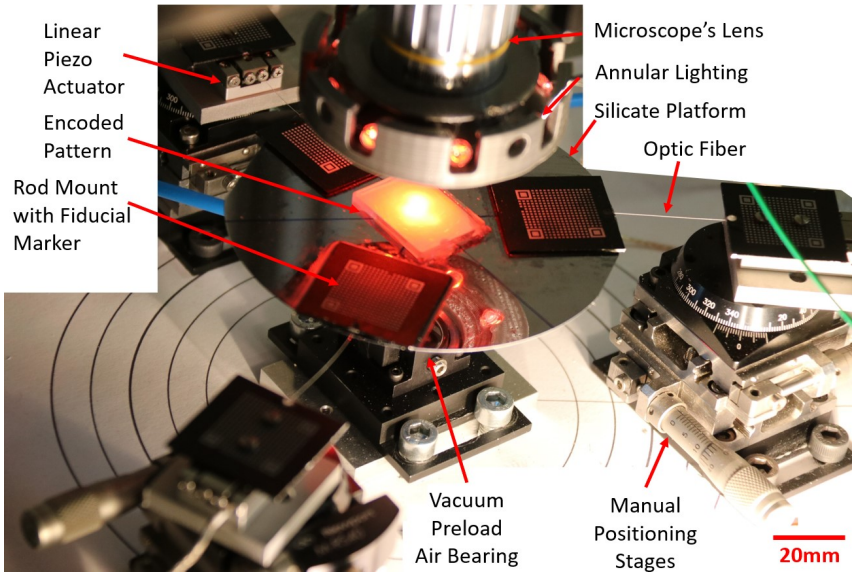


Figure 5.1: Prototype of the Triskele-bot

Rods

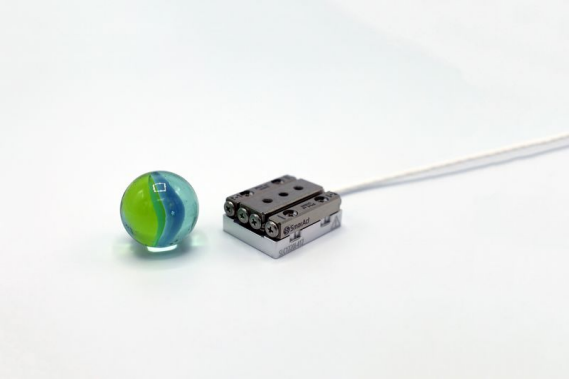
The flexible element is the fundamental part of a continuous robot. It must transmit the forces, moments, and displacements from the actuators to the platform. In particular, they must be able to both push and pull. The characteristics of the chosen material and the rod's size influence have an important impact on the behavior of the robot. In order to obtain an accurate and repeatable robot, optical fibers have been considered. This material is purely elastic without vicious property. Such material has a Young's Modulus of 69 GPa and a Shear Modulus of 29.58 GPa, according to articles [3]. The rods have a circular cross section of a diameter of 125 μm and a length of 30 mm.

Actuators

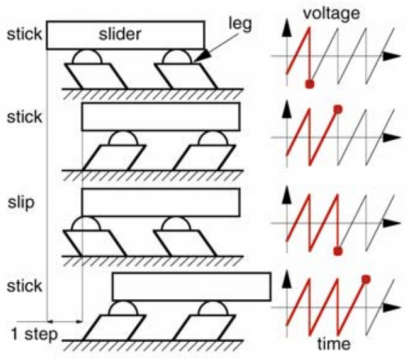
The actuators used in the Triskele-bot are piezo-electric linear actuators. The model SLC-1730 showed in Figure 5.2a was used. It is produced by the SmartAct.

They operate using the stick-slip principle. As showed in Figure 5.2b, during the stick phase, the piezoelectric material expands slowly, so the moving part moves following the extension of the piezoelectric part. In the slip phase, the piezoelectric material retracts very quickly, so the movable part slips and does not move. The cycle is then repeated.

The advantage of this working way is that the accuracy of the actuators is very high. Furthermore, the actuators are equipped with sensors, which allow the feedback control. Thanks to such characteristics the actuators on a displacement of 100 μm have a repeatability of 18.02 nm. Moreover they have a maximum range of 21 mm, and a maximum speed of 2 $mm.s^{-1}$.



(a) Linear piezo-electric actuator of the SmarAct SLC-1730



(b) Stick-slip actuator working diagram

Figure 5.2: Stick-slip actuation

Air bearing

For frictionless planar motion, the platform is supported by a vacuum preload air bearing. It was used the IBS bearing S205001, 50 mm in diameter as showed in Figure 5.3. This bearing has a circular shaped surface. In the periphery it pushes the platform creating a suppression, while in the central part it creates a depression. In this way, the distance to the platform is made constant at about 5 μm.

Figure 5.3: IBS airbearing s205001

5.1.2 Measurement

The experimental setup uses two different measurement systems. The first one uses a camera that reads the position of QR-codes-like pattern, placed at crucial points of the robot. This method is used to identify the position and orientation of the ends of the flexible elements.

The second method is to measure the position of the end-effector using an encoded pattern. Using this pattern, it is possible to measure with a high resolution the position

of the robot at any instant.

Both pattern and their algorithms have been developed inside the laboratory. Further information can be found inside the articles [1, 2].

QR Code measurement setup

As it is possible to observe in Figure 5.4, an IDS U3-3070CP camera mounted with a 50 mm lens and positioned at a distance of 1.5 m from the robot plane focuses on the robot and can see all Qr-codes at once. Knowing the position and orientation of those patterns and thanks to the fabrication process, the position and orientation of the flexible element's ends can be deduced. It is possible to observe in Figure 5.5a how the QR codes appear from the camera.

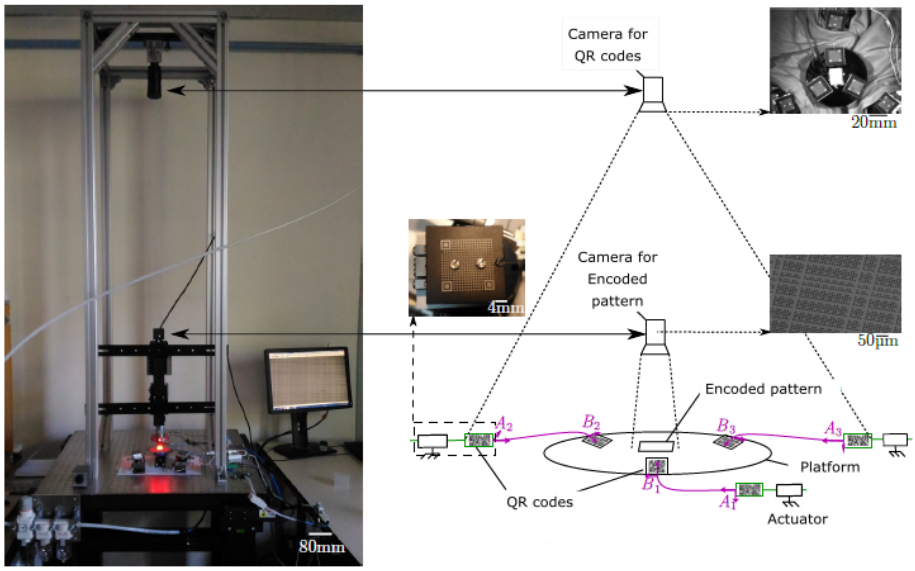
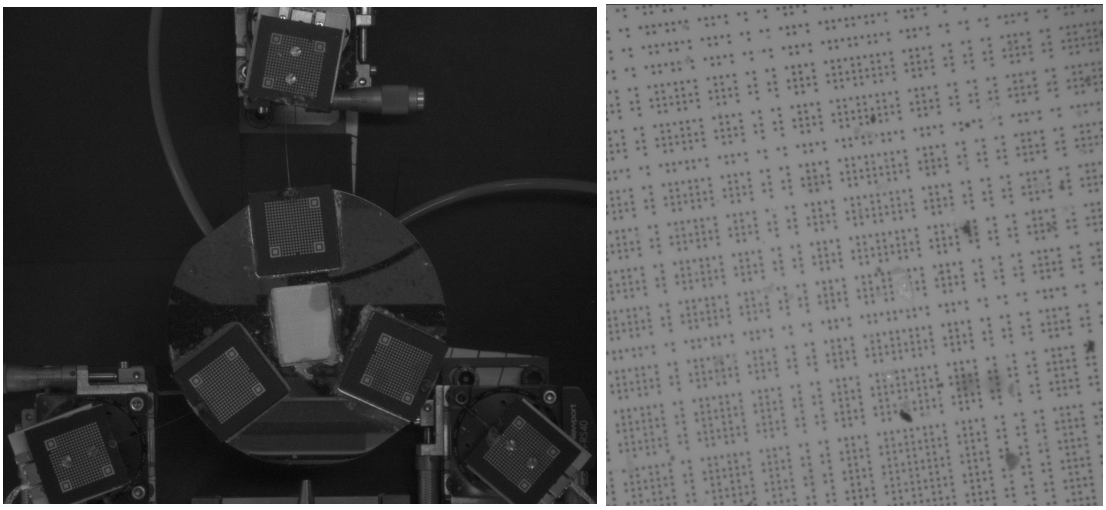


Figure 5.4: Camera setup

Encoded Pattern measurement setup

The measurement setup consists of a camera with an optical microscope and an objective, the environment is illuminated with LEDs that focus on a encoded pattern. The pattern was manufactured by etching a chromium layer onto a transparent glass wafer. The periodicity between features is 9 μm along the two directions. This pattern is glued on the platform of the robot. What is possible to see from the camera is shown in figure 5.5b.

Through a system of vision, image processing and decoding determine the planar Cartesian coordinates (x,y, θ) of the image obtained from the pattern with respect to the



(a) View on the QR codes (b) View on the encoded pattern

Figure 5.5: View from the two cameras

global position of the pattern.

As we want to study the dynamics of the robot, the camera and the setup need to be designed to have the highest possible acquisition speed with the lowest possible noise.

In order to have a good capture rate, the first bottleneck of the camera is the data rate of 5 Gbps. This is a problem because using the maximum resolution of 2048x1536 it is not possible to achieve a high frame rate.

So the first parameter modified was the resolution set to 600x600, which is the minimum resolution possible to use the decoding algorithm. With this resolution it was possible to use a frame rate of 114.44 fps, that is the maximum for this camera.

An additional parameter to adjust was the exposure time. The exposure time is the time the digital sensor is exposed to light. So the amount of light reaching the sensor is proportional to the exposure time and a large value of it will mean a very brightly illuminated image, and a small exposure time a dark image.

But exposure time has a big impact on the image especially when you want to record moving objects. In fact, a large exposure time will result in a blurred image and the algorithm will not be able to decode the pattern.

So, in order to get a sharp image in our case we needed to use a very small exposure time, and at the same time to get a sufficiently illuminated image we used a set of 15 LEDs with a power of 120 mW each and providing 40000 lm each.

With these precautions it was possible to use an exposure time of 0.2 ms.

What is possible to see from the camera is shown in Figure 5.5b.

5.2 Free motion

The experiment performed on the Triskele-bot involves applying an external force to the end-effector that is suddenly released. In this way it is possible to observe the free motion of the robot. To perform this experiment, the external force was applied through the mechanism of a pen. The pen was firmly anchored to the work table and it is mooved thanks to a manual positioning stages using a micrometer-screw. To apply the force the pen tip is pulled out to its maximum extension, then it is retracted with the snap of the spring. In this way the retraction of the pen is much faster than the movement of the robot, and therefore will not affect the free movement.

The purpose of this experiment was to get a comparison with the simulation, and then evaluate any issues with it.

5.2.1 Results

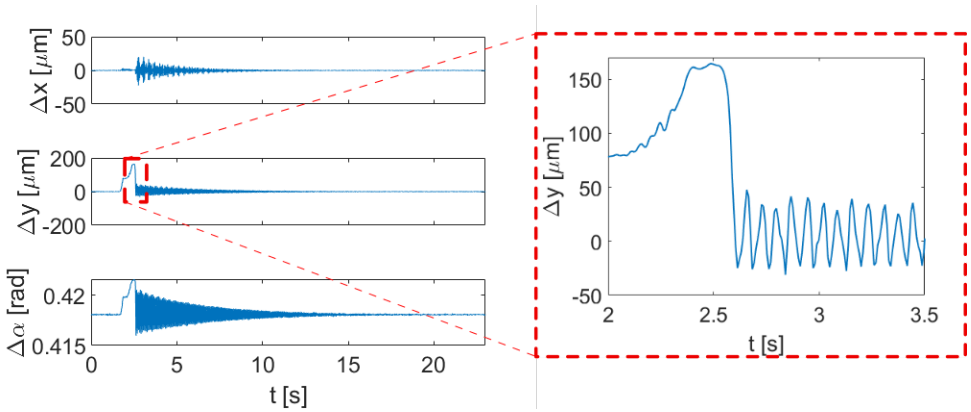


Figure 5.6: Displacement of the End-effector after applying and releasing a perturbation force

It is possible to observe the movement of the robot in Figures 5.6,5.7a and 5.7b. As can be seen in Figure 5.7b, when the force is released the robot movement is not in the same direction of the force, but becomes very chaotic. The reason of that is the asymmetry of the robot along the direction of the force.

From the experiment there is a degree of uncertainty due to the fact that the transformation between the global reference system and the local reference system is unknown. Therefore the data obtained must be considered as relative movements and not absolute movements.

Moreover from the zoom in Figure 5.6 it is possible to observe that at the moment of the force release there is a high damping of the motion.

A further problem is that the acquisition speed of the camera allows the recording of a maximum of 10 points for each period of oscillation. This certainly does not allow a proper motion analysis in all the frequency of the movement.

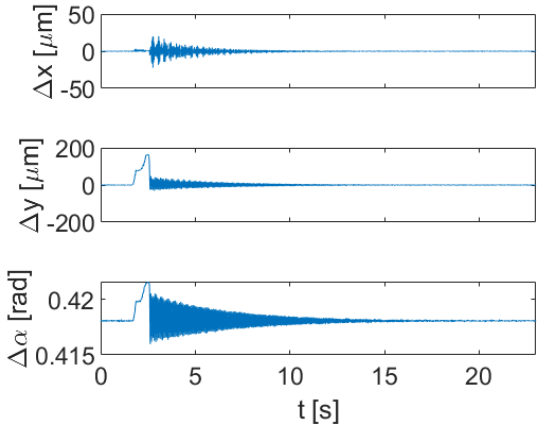
Trying to reproduce the experiment by simulation the results are not satisfactory. It is possible to compare the simulation with the experiment in Figure 5.7, and the experiment and the model are not converging.

To have good results some problems still need to be solved. The biggest problem is that a better model of the robot should be obtained, but some parameters can be identified only experimentally or by calibration.

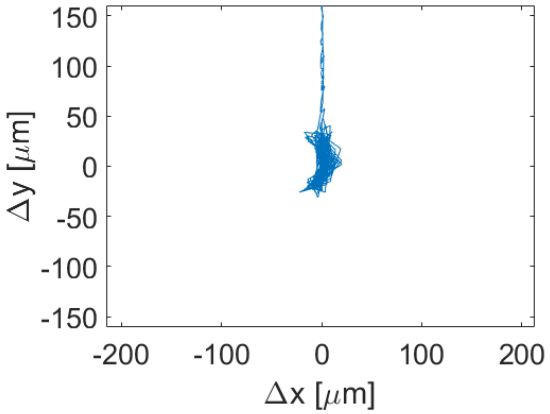
In particular, it is very important to be able to find the damping matrix, which it would allow to understand how quickly the robot tends to reach an equilibrium. In fact it can be observed that in the simulation and in the experiment the robot has a different converging behaviour.

So, what is possible to compare now is just the general behavior of the robot.

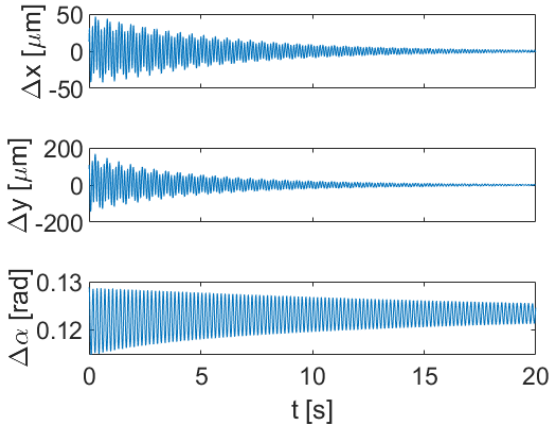
The analogies that can be found are that in both graphs there is an ovoid motion. In the experiment the movement converges and is very chaotic, and the main axis of the ovoid rotates slightly, creating a shape similar to a banana. Instead, in the simulation the robot have an ovoid movement with a diagonal direction, given by the asymmetry of the robot.



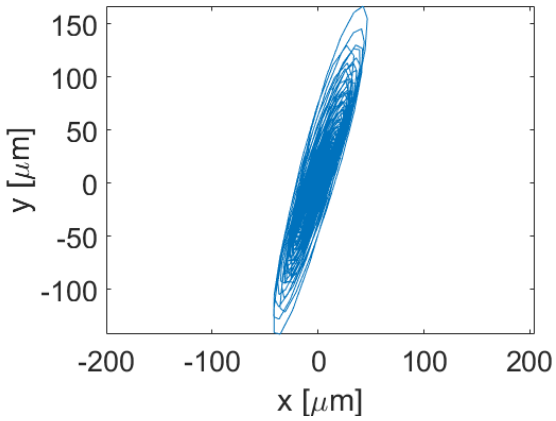
(a) Experiment: Position of the end-effector in x, y and θ over the time



(b) Experiment: Position of the end effector in the plane x, y



(c) Simulation: Position of the end-effector in x, y and θ over the time



(d) Simulation: Position of the end effector in the plane x, y

Figure 5.7: Displacement of the end-effector after an external force, in the experiment and in the simulation

Chapter 6

Conclusions and Future projects

Conclusions

Technologies are advancing and offering new opportunities. In robotics in recent years, one area that is gaining attention is continuuum robots. These robots offer possibilities that have not been explored yet, such as safer human-machine interaction, the possibility of miniaturization, and lower weight. But there is still the need to address several challenges. One of them is dynamics modeling. The purpose of this work is to address the study, then to try to create a model for the dynamics of parallel continuum robots that is adequately robust and accurate, and that is potentially usable in real-time simulations.

For this purpose it was decided to use a model based on the Cosserat elasticity theory, according to which it is possible to model flexible one-dimension rods accounting for bend, twist, stretch, and shear.

This model was therefore applied to three different simulated cases.

The first simulation is of a cantilever rod to which a force is applied and then released. This simulation was useful to understand how the different simulation parameters affect the model.

The second simulation is of a continuum version of Stewart-Gough's robot. It was useful to understand the size of the simulation noise.

The third simulation is of the Triskele-bot. Both forward and inverse dynamics were used for this simulation to understand if the model is effective in both directions.

Finally, an experiment was performed with the physical version of the Triskele-bot. An external force was applied to the robot and after quickly released, in order to observe the free oscillation of the robot. Finally it has been tried to replicate this experiment in simulation, with a result not fully satisfactory, but with good hopes for improvement given the very similar general behavior.

Future projects

A work to be done in the near future concerns the motion of the triskele-bot at constant speed through the actuators, and then the observation of its free movement after blocking the actuators. This experiment will also be replicated in the simulation in order to understand if more coherent results will be obtained compared to the first experiment. Currently it was not possible to perform this experiment due to a breakdown of the actuators.

Next we would like to implement a 7 DoFs robot, and proceed with further simulations of such a robot.

A further step would be to implement the code in C++ in order to verify the performance, and then if it is possible to make it working in real time.

A further progress would be the design of a controller that uses the dynamic model as an observer. To do this, however, the use of a microprocessor capable of working in real time would be necessary to control the actuators.

List of Figures

List of Tables

Bibliography

[1] A. N. Andre et al. "Sensing One Nanometer over Ten Centimeters: A Micro-Encoded Target for Visual In-Plane Position Measurement". In: *IEEE/ASME Transactions on Mechatronics* (2020), pp. 1193–1201. ISSN: 1941-014X. DOI: 10.1109/TMECH.2020.2965211.

[2] Antoine N André et al. "Robust phase-based decoding for absolute (X, Y, Θ) positioning by vision". In: *IEEE Transactions on Instrumentation and Measurement* 70 (2021), pp. 1–10. DOI: 10.1109/TIM.2020.3009353.

[3] Paulo Antunes et al. *Mechanical properties of optical fibers*. INTECH Open Access Publisher, 2012.

[4] Caroline B Black, John Till, and D Caleb Rucker. "Parallel continuum robots: Modeling, analysis, and actuation-based force sensing". In: *IEEE Transactions on Robotics* 34.1 (2017), pp. 29–47.

[5] Caroline E Bryson and D Caleb Rucker. "Toward parallel continuum manipulators". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 778–785.

[6] Jessica Burgner-Kahrs, D Caleb Rucker, and Howie Choset. "Continuum robots for medical applications: A survey". In: *IEEE Transactions on Robotics* 31.6 (2015), pp. 1261–1280.

[7] FJ Campa et al. "A 2 dof continuum parallel robot for pick & place collaborative tasks". In: *IFToMM World Congress on Mechanism and Machine Science*. Springer. 2019, pp. 1979–1988.

[8] Eulalie Coevoet, Adrien Escande, and Christian Duriez. "Optimization-based inverse model of soft robots with contact handling". In: *IEEE Robotics and Automation Letters* 2.3 (2017), pp. 1413–1419.

[9] Christian Duriez. "Control of elastic soft robots based on real-time finite element method". In: *2013 IEEE international conference on robotics and automation*. IEEE. 2013, pp. 3982–3987.

[10] Wisama Khalil and Etienne Dombre. *Modélisation identification et commande des robots*. Hermes, 1999.

[11] Harvard Biodesign Lab. *Soft Robotics*. URL: https://biodesign.seas.harvard.edu/soft-robotics.

[12] Benjamin Mauzé. "Triskèle-Bot: étude et développement d'un robot parallèle continu pour le micro-positionnement". PhD thesis. Université Bourgogne Franche-Comté, 2021.

[13] Benjamin Mauzé et al. "Micrometer Positioning Accuracy With a Planar Parallel Continuum Robot". In: *Frontiers in Robotics and AI* 8 (2021), p. 196.

[14] Benjamin Mauzé et al. "Nanometer precision with a planar parallel continuum robot". In: *IEEE Robotics and Automation Letters* 5.3 (2020), pp. 3806–3813.

[15] Hayley McClintock et al. "The milliDelta: A high-bandwidth, high-precision, millimeter-scale Delta robot". In: *Science Robotics* 3.14 (2018).

[16] Kathrin Nuelle et al. "Modeling, calibration, and evaluation of a tendon-actuated planar parallel continuum robot". In: *IEEE Robotics and Automation Letters* 5.4 (2020), pp. 5811–5818.

[17] Andrew L Orekhov et al. "Analysis and validation of a teleoperated surgical parallel continuum manipulator". In: *IEEE Robotics and Automation Letters* 1.2 (2016), pp. 828–835.

[18] Federico Renda et al. "Discrete Cosserat Approach for Multisection Soft Manipulator Dynamics". In: *IEEE Transactions on Robotics* 34.6 (2018), pp. 1518–1533. DOI: 10.1109/TRO.2018.2868815.

[19] Inderjeet Singh et al. "Optimal work space of parallel continuum manipulator consisting of compact bionic handling arms". In: *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2017, pp. 258–263.

[20] Ryo Takano, Hiromi Mochiyama, and Naoyuki Takesue. "Real-time shape estimation of Kirchhoff elastic rod based on force/torque sensor". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2508–2515.

[21] John Till and D Caleb Rucker. "Elastic rod dynamics: Validation of a real-time implicit approach". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 3013–3019.

[22] John Till and D Caleb Rucker. "Elastic stability of cosserat rods and parallel continuum robots". In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 718–733.

[23] Robert J Webster, Joseph M Romano, and Noah J Cowan. "Mechanics of precurved-tube continuum robots". In: *IEEE Transactions on Robotics* 25.1 (2008), pp. 67–78.

[24] Guanlun Wu and Guanglin Shi. "Experimental statics calibration of a multi-constraint parallel continuum robot". In: *Mechanism and Machine Theory* 136 (2019), pp. 72–85.

[25] Zhixiong Yang, Xiangyang Zhu, and Kai Xu. "Continuum delta robot: a novel translational parallel robot with continuum joints". In: *2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*. IEEE. 2018, pp. 748–755.

[26] Michael C Yip, Jake A Sganga, and David B Camarillo. "Autonomous control of continuum robot manipulators for complex cardiac ablation tasks". In: *Journal of Medical Robotics Research* 2.01 (2017), p. 1750002.