



HAL
open science

Design, implementation and control of a deformable manipulator robot based on a compliant spine

Thor Morales Bieze, Alexandre Kruszewski, Bruno Carrez, Christian Duriez

► To cite this version:

Thor Morales Bieze, Alexandre Kruszewski, Bruno Carrez, Christian Duriez. Design, implementation and control of a deformable manipulator robot based on a compliant spine. The International Journal of Robotics Research, 2020, 10.1177/ToBeAssigned . hal-03028723

HAL Id: hal-03028723

<https://hal.science/hal-03028723v1>

Submitted on 11 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Design, implementation and control of a deformable manipulator robot based on a compliant spine

January 11, 2021

Thor Morales Bieze¹, Alexandre Kruszewski¹, Bruno Carrez¹ and Christian Duriez¹

¹ DEFROST team (Inria, Université de Lille, Ecole Centrale Lille, CNRS) Lille France

Corresponding author : Alexandre Kruszewski, 40 avenue Halley, 59650 Villeneuve d'Ascq, France
alexandre.kruszewski@centralelille.fr

Abstract

This paper presents the conception, the numerical modeling and the control of a dexterous, deformable manipulator bio-inspired by the skeletal spine found in vertebrate animals. Through the implementation of this new manipulator, we show a methodology based on numerical models and simulations, that goes from design to control of continuum and soft robots. The manipulator is modeled using Finite Element Method (FEM), using a set of beam elements that reproduce the lattice structure of the robot. The model is computed and inverted in real-time using optimisation methods. A closed-loop control strategy is implemented to account for the disparities between the model and the robot. This control strategy allows for accurate positioning, not only of the tip of the manipulator, but also the positioning of selected middle points along its backbone. In a scenario where the robot is piloted by a human operator, the command of the robot is enhanced by a haptic loop that renders the boundaries of its task space as well as the contact with its environment. The experimental validation of the model and control strategies is also presented in the form of an inspection task use case. ‘

Soft robots, Continuum robots, Deformable manipulators, Finite Element Method

1 Introduction

Industrial inspection tasks very often involve the transportation of instrumentation equipment to hard-to-reach areas and the manipulation of this equipment

in constraint spaces. The requirements of such tasks present a challenge to conventional rigid robots which are very often heavy and bulky due to their design specifications. A possible solution to such applications would be a deformable continuum manipulator robot with a high degree of dexterity which is able to negotiate its way around delicate equipment in narrow spaces. Some of the desired capabilities for such robot are:

- Large workspace: Ideally, the robot would be deployed away from the area to be inspected. It is then necessary for the robot to have a considerable reach and a large task space.
- Dexterity: Due to the constraints of the task, it is desired that the robot is able to reach a position inside its task space with more than one orientation of its end-effector.
- Ability to sneak between obstacles: The robot should have a high number of degrees of freedom in order to avoid obstacles while still being able to reach the desired position/orientation with its end-effector.
- Very compliant: In some cases, the contact with the environment cannot be avoided, therefore, in order to avoid or minimize the damage to the surroundings, it is necessary that the robot is compliant to external forces.

In order to design such robot, one can turn to nature to seek for inspiration, however, the lack of design methodologies for bio-inspired continuum robots pose certain issues: First, using the same principle of functionality in different environments can cause the robot to under-perform. For example, the idea of mimicking the morphology of an octopus tentacle may seem attractive at first glance, however, in this particular case, the medium (water) in which the source of inspiration normally operates provides a key support component (buoyancy) that enables the octopus to reach and grab objects with its boneless tentacles. However, outside of the water and without a backbone, a robotic tentacle may lack the structural support to perform the required task. Second, the lack of simulation tools to validate those principles of functionality may translate in a considerable amount of iterations in the process of design and implementation. And third, the conceptual stages of a continuum manipulator are often disjointed from its modeling and control methodologies, which may result in a device that is too complex to control. In the work presented in this paper, a simulation tool is at the very core of the robot implementation, that allows for a reduced number of design iterations (reduced cost of time and materials due to failed prototyping) and that also provides a suitable control strategy for the robot for both, direct piloting and path planning.

By a rapid shortcut, one might be tempted to equate *rigid* robots with vertebrates and *soft* robots with invertebrates. Yet the skeleton is structured around a spine that is mechanically designed to ensure flexibility. The spine is an important mechanical support and the vertebra repetition interspersed with soft inter-vertebral discs allows great freedom of movement, while absorbing shocks.

Additionally, the spine protects the transmission of information through the spinal cord.

All these elements have led us to consider the idea of drawing inspiration from the skeletal spine to design a compliant manipulator robot, capable of great freedom of movement, especially for inspection tasks in confined environments. We called this manipulator Echelon 3 (see Figure 1). To the best of our knowledge, this continuum robot is the very first to be designed, modeled and controlled in a single modeling framework: the finite element method. The methodology used is generic and can be used to design other robots. The simulation-assisted design, modeling and control strategies of the robot are explained in the following sections.

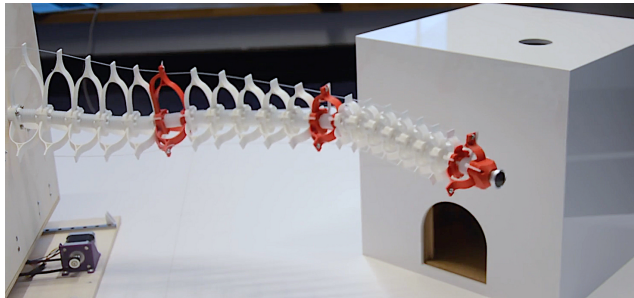


Figure 1: The Echelon 3 deformable manipulator.

2 Related Work

Due to their compliance to external forces and their dexterity (the variety of tasks that the robot can complete, and also how well it can perform those tasks ([Ma and Dollar(2011)]), continuum manipulators have become a suitable alternative to rigid manipulators in applications that involve dynamically changing environments or the direct contact with humans. Such applications include skeletal trauma treatment ([Wilkening et al(2011)], [Alambeigi et al(2017)]), endoscopy ([Cianchetti et al(2013)], [Fraś et al(2015)]), minimally invasive surgery ([Orehov et al(2016)], [Mahoney et al(2016)]), minimally invasive inspection ([Tonapi et al(2014)], [Greer et al (2018)]) and search and rescue ([Bajo and Simaan(2010)]).

Continuum manipulators can be broadly classified with respect to the actuation system implemented to apply forces and torques to the backbone. Intrinsic or locally actuated designs utilize pneumatic or hydraulic actuators made of elastomers, embedded in the structure of the robot to apply forces directly to the backbone. These actuators, referred to as pneumatic artificial muscles (PAM's) reproduce the way longitudinal muscles work in an animal trunks or tentacles. Some examples of locally actuated continuum robots include the Oct-Arm ([Walker et al(2005)], [McMahan et al(2006)]), the "European Octopus" ([Laschi et al(2009)], [Calisti et al(2012)]) and the Bionic Handling Assistant

([Rolf and Steil(2012)]). A disadvantage of using pneumatic actuators is that they tend to exhibit non-linear behaviors such as hysteresis, which makes the control of the robot a complex task.

Extrinsic actuation designs employ filaments or tendons routed along the structure of the manipulator. The tendons are pulled by motors positioned at the base of the robot to apply torques to the backbone to produce bending ([Cojocaru et al(2010)], [Ataka et al(2016)]). Walker and Hanna presented a study on a soft manipulator inspired by the elephant trunk ([Walker and Hannan(1999)]), that allowed the identification of critical hardware necessities, in particular the need for elastic structures and backbone support. In the work presented by Gravagne et al. ([Gravagne and Walker(2003)]), an incompressible elastic rod is used as a backbone to guide the bending of the manipulator. This choice of backbone has been the most used in tendon based designs ([Simaan and Flint(2004)], [Bardou et al(2010)], [Zhao and Gao(2010)]). When the application requires more compliance on the robot, a central spring can be used as a backbone ([Mehling et al(2006)], [Yoon et al(2011)]), however, these designs tend to be more difficult to control due to the spring absorbing a part of the actuation forces.

2.1 Modeling and Control of Soft Manipulators

The advantages of soft continuum robots come with the trade off of complexity in their modeling and control. In order to control a soft manipulator, a transformation between the low level degrees of freedom (actuator space) and the high level degrees of freedom which are the position and orientation of the tip of the robot (task space) needs to be formulated. A very common approach towards this model is to pass through an intermediate transformation involving the geometry of the robot. Using the assumption that, after actuation is applied, each section of the robot follows a constant curvature, the state of the tip of each section of robot can be related to its curvature ([Webster and Jones(2010)]), this approach have been implemented several times in the literature ([Rolf and Steil(2012)], [Escande et al(2012)], [Escande et al(2015)]). When the manipulator is composed of several serially connected constant curvature sections, the same approach can be used by concatenating the transformation matrices of each section, as presented in ([Jones and Walker(2006)]), [this approach is also compatible with closed-loop controllers where the state of the robot is acquired by external sensors or cameras](#) ([Marchese and Rus(2016)]). When it is assumed a deformation with constant curvature between sections, the system can then be considered within a framework approaching that of a rigid articulated model. However, these models ignore the notion of internal forces equilibrium which is very important when studying deformations: material properties, self-collision and deformations caused by the environment could all have an influence on the kinematics of a soft manipulator.

Due to their ability to by-pass the complexity of modeling the highly non-linear behavior of soft continuum manipulators, data driven approaches have gained popularity in the literature. These approaches are based on previous

knowledge of input-output data derived from experimentation and can provide accurate fast approximations to the kinematic model ([Rolf et al(2010)], [Rolf and Steil(2014)]). Neural network approaches have been also proposed in the context of learning the kinematics of continuum manipulators using geometry models for data recollection ([Giorelli et al(2015)]), or as part of low level controllers to compensate for the dynamic uncertainties of the robots ([Braganza et al(2007)]). While data-driven models have shown great success in real world scenarios, the fact that the learning base changes when the operation conditions of the robot change make these models limited in their application.

The need of accurately modelling the physical behavior of continuum manipulator has led to the use of the theory of Cosserat rods ([Renda et al(2014)]). Based on such model, the control of the position of the remote end of these robots has been implemented in [Torres and Alterovitz(2011)]. Most proposed models only consider the deformation of the central line of the robot. However, given the structure chosen for the robot presented in this paper, it is important to take into account the deformations created by the lattice structure. Yet, the deformation of lattice structure has been widely modeled in FEM using a combination of beam elements ([Przemieniecki (1985)]).

2.2 Outline of the paper

The paper presents new methodological steps for the conception, model and control design of a continuum manipulator, implemented for a compliant spine robot, actuated by tendons.

The contributions of this paper, with respect to the current state-of-the-art, are as follows:

- In the design of the manipulator, we proved that it is possible to obtain large global deformations while limiting the amplitude of the local deformations in the structure of the robot. The robot is made of 3D printed vertebrae, however, it retains a high compliance and its motion is created by deformation.
- We proved that taking the design inspiration from vertebrate animals does not undermine the soft capabilities of the continuum manipulator.
- We showed how a simulation of the FEM model of the robot is used to unify the different stages of its implementation. This allows for a reduction of iterations on the design stage and also provides a suitable modeling and control strategies for the continuum manipulator.
- We demonstrated teleoperation with haptic feedback for a more intuitive manner of control, in the case where the robot is controlled by a human operator.

The paper is organized as follows: first, the mechanical design of the deformable continuum robot is presented in the next section, then the mechanical model with 3D simulation and the control design are explained. The control is

further extended for tele-operation with haptic rendering and the experimental results are discussed before the conclusion.

3 Design

On the skeletal spine, ligaments make the connection between the vertebrae and stabilize them. Their deformation allows for a good overall range of motion but they have also a limiting movement role to protect the link. Their local deformation is relatively small. We have kept these principles in our design to create large amplitudes of global movement from relatively small local deformations which are constrained to remain small by the design.

Another source of inspiration in the design of the manipulator is the engineering truss (i.e. a framework consisting of beams and rods that is often used in civil engineering to support a roof, bridge, or other structures.)

3.1 Hardware description

The manipulator is composed by 3 sections with a total length of 78.3cm. Each section is actuated extrinsically by 3 tendons attached at the section end points (see Figure 2 and Extension 1). By pulling the tendons in different configurations, one can change the orientation and position of the tip of each section. To provide the tension in the tendons, a Pololu 3215 DC motor actuates a timing belt based system that pulls the tendon mounted at the back end of the robot. This choice of transmission is to avoid using a pulley system where the tendon can slip out of the spool when not in tension. The actuator setup is presented in Figure 3.

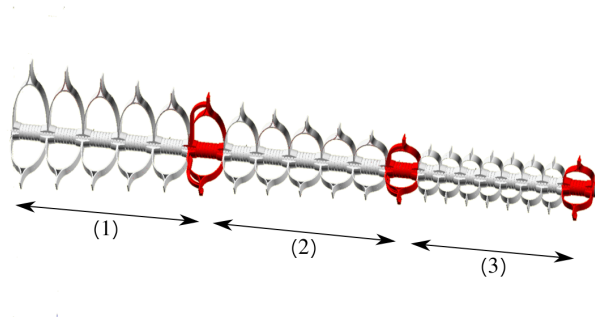


Figure 2: The manipulator assembly. The 3 sections of the manipulator: (1) base section, (2) middle section and (3) the distal section. The red vertebrae in the arm are the section end points where the tendons are fixed.

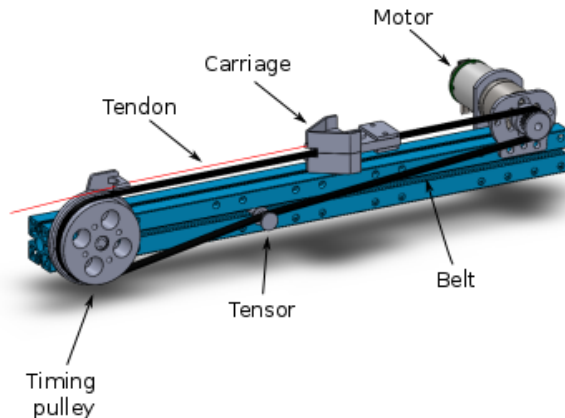


Figure 3: Actuator setup. Each tendon is attached to a carriage mounted on top of a timing belt. The belt is moved by a Pololu 3215 DC motor.

The backbone of the robot is made out of corrugated high density Polyethylene with an elastic module of 0.8 GPa and a tensile strength of 15 MPa. The role of the backbone in the design is twofold: to give support to the structure of the robot while preventing the shearing between the vertebrae, and to provide housing for transmission and instrumentation elements (tendon guides and camera and sensor wires).

As illustrated in Figure 4, a transmission tube made of Teflon-PTFE is allocated inside the backbone. This tube connects the actuators at the back end of the robot to the middle and distal sections of the manipulator. The internal routing of the tendons reduces the required actuator stroke and decouples the deformation of each section.

The vertebrae that form the sections of the manipulator are 3D printed using regular Polylactic Acid (PLA) with an elastic module of 3.5GPa and a tensile strength of 50MPa and the tendons are made of regular Nylon fishing line rated for 36kg of tension. As mentioned before, the main role of the vertebrae is to guide the tendons along the backbone and to limit its local deformation by means of self-collision. In the base and middle sections, the dorsal part of the vertebrae is longer in comparison to the lateral parts (see figure 5). The objective of this feature is to increase the torque applied by the tendons at the termination of the sections to help the manipulator cope with gravitational forces.

The manipulator lacks extension since the backbone is passive. To compensate the lack of extension, the manipulator is mounted on top of a sliding platform which is actuated simultaneously by 2 Nema 17 stepper motors. This provides an additional degree of freedom along the longitudinal axis of the manipulator.

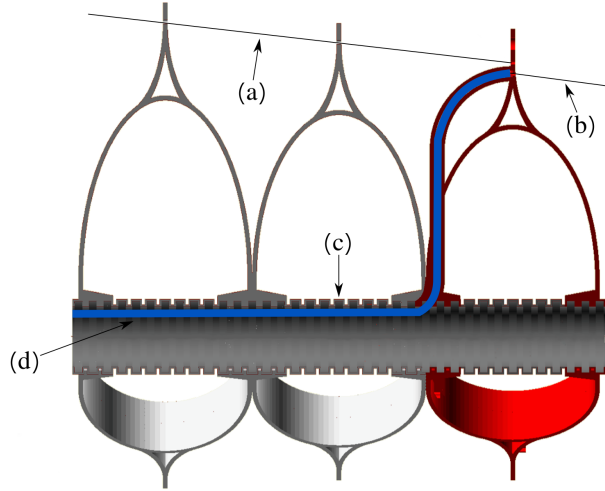


Figure 4: Longitudinal cut of the backbone and dorsal vertebrae. (a) Upper tendon of base section (b) Upper tendon of middle section (c) Backbone (d) Teflon-PTFE transmission tube allocated inside the backbone.

4 Modeling and simulation

In this section, we describe the mechanical model used to calculate the kinematic model of the robot. We use the Finite Element Method (FEM) with curved beam type elements. These beams are then assembled by applying the rigid transformations corresponding to the geometry of the vertebra structure. We reproduce the internal forces of the structure, the mass, the forces that apply to the passage of cables, as well as the external forces applied to the robot. From this mechanical model we derive the direct model as well as the inverse model.

4.1 Curved beam Model

The robot structure is based on repetitive pattern (the trellis). Each element is composed of tubes and curved rods that can be modeled in a generic way by FEM using curved beam elements. The main assumption is that the beam elements have a constant section along their main axis. The sections are circular (to model the central tube like in Figure 7 (d) and (e)) or rectangle (to model the vertebrae like in Figure 7 (b) and (c)). Each beam is parameterized by 2 frames \mathbf{X}_1 and \mathbf{X}_2 , i.e. 2 nodes that have 6 Degree Of Freedom (DOF): 3 translations, 3 rotations. Thus $\mathbf{X}_1, \mathbf{X}_2 \in SE(3)$ A precise interpolation along the main axis of the element is obtained by using cubic spline curve (see Figure 6).

Mechanically, the beam model integrates on the chosen geometry an elastic law of linear behavior with common parameters such as the Young's modu-

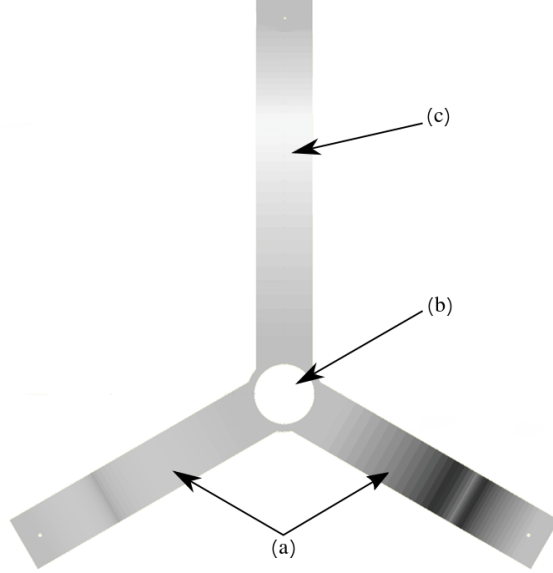


Figure 5: Frontal view of the vertebra. (a) Vertebra lateral segments (b) Backbone attachment point (c) Vertebra dorsal segment.

lus and the Poisson ratio. After computing the area moment of inertia of the section geometry, a stiffness matrix \mathbf{K}_b can be obtained in the local frame of each beam (a full description of how this matrix is obtained can be found in [Przemieniecki (1985)]). This 12×12 matrix provides the internal forces and moments \mathbf{f}_b observed on the two nodes when it is multiplied by \mathbf{u}_{loc} the displacement of the nodes in the local frame of the beam.

$$\mathbf{f}_b = \mathbf{K}_b \mathbf{u}_{loc} \quad (1)$$

It accounts for the energies of deformation in the 6 possible directions: 1 direction of stretching, 2 directions of bending, 1 direction of torsion, 2 directions of shearing.

In our formulation, the local frame corresponds to the frame interpolated in the middle of the beam $\mathbf{X}_b \in SE(3)$. To compute this frame, we use the spline interpolation of the element as explained in Figure 6.

Once \mathbf{X}_b has been computed, it provides the transformation from the world coordinate to the local frame of the beam. The rotation \mathbf{T}_b defined by this transformation will be used to *rotate* the matrix of the beam \mathbf{K}_b in the global frame, to obtain a corotational approach, like in [Crisfield (1990)].

$\mathbf{u}_{loc} \in R^{12}$ is composed by two displacement twists $\mathbf{u}_{1,loc} \in se(3)$ and

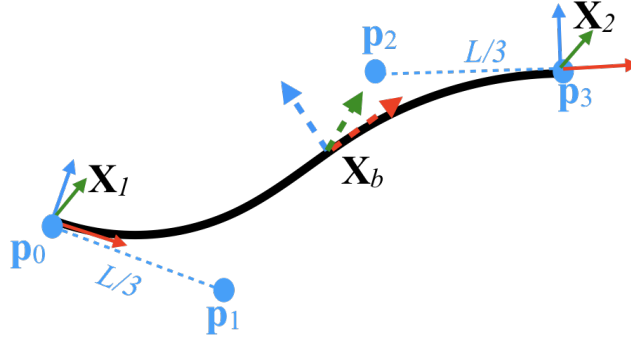


Figure 6: The beam position is defined by two frames \mathbf{X}_1 and \mathbf{X}_2 placed at the extremities. These frames are the Nodes of the beam. Based on the position of these two frame, we place the 4 control points of the spline $\mathbf{p}_0, \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$, using the initial length L of the beam. Vectors $\mathbf{p}_0\mathbf{p}_1$ and $\mathbf{p}_2\mathbf{p}_3$ are respectively aligned with the \mathbf{x} axis (in red in the picture) of \mathbf{X}_1 and \mathbf{X}_2 . After construction of the spline, the frame of the beam \mathbf{X}_b is interpolated so that its \mathbf{x} axis is aligned with the spline tangent. The torsion (rotation along \mathbf{x}) between \mathbf{X}_1 and \mathbf{X}_2 is also interpolated (not shown on the picture)

$\mathbf{u}_{2,loc} \in se(3)$. Their computation is based on the motion of frames \mathbf{X}_1 and \mathbf{X}_2 in the reference frame of the beam \mathbf{X}_b from the initial position to the current one.

The formulation of the beam allows different types of geometrical sections and not straight initial shape. To model the robot, we used both rectangular and tubular sections as shown in Figure 7.

4.2 Assembling and definition of the degrees of freedom

The mechanical structure of the robot is a truss, which corresponds to the assembly of beams. The geometrical assembling of the truss follows a series of parameters that have been described in the Design section: number of sections, number of vertebrae per sections, size of the vertebrae, placement of the cables, etc... It has been implemented in a generic manner, allowing for testing, in simulation, different designs.

One of the key component in the modeling is the mechanical assembly of beam models. The assembly is based on the fact that the structure is composed of beam elements connected by nodes. Each node of the beam model described bellow is defined by rigid frame. However, we have distinguished the concept of a frame *node*, i.e. extremity of a beam element and the concept of frame *DOF*, (for Degree Of Freedom), i.e. number of independent unknowns that define the configuration of the robot. Between node and DOF frames, we can use rigid transformations, traditionally used in rigid robotics, to perform the assembly.

We can model that a frame node is rigidly attached to a frame DOF during

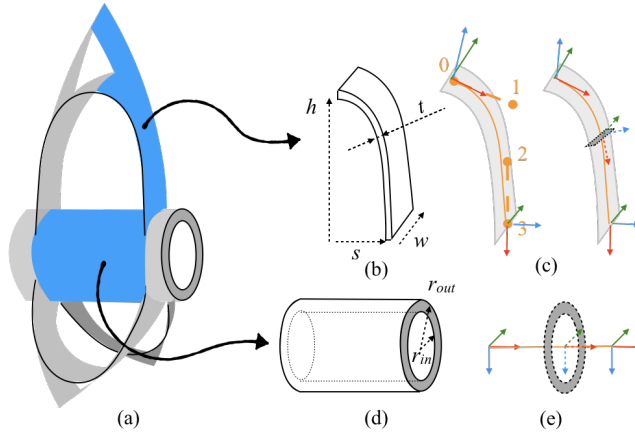


Figure 7: Mechanical model of the robot decomposed in beam elements. (a) element of the structure of the robot (b) geometrical parameters of the vertebrae (c) curved beam, left: 4 points of the splines, right: interpolation of the rectangular section between the two extremities of the beam (d) geometrical parameters of the central tube (e) beam model with the interpolation of the section of the tube between the two extremities of the beam.

assembly. Consequently, there is a rigid transformation between the DOF and the node. We can compute the rigid position of the node in the DOF frame: ${}_{\text{DOF}}\mathbf{X}_{\text{node}} \in SE(3)$. (We have ${}_{\text{DOF}}\mathbf{X}_{\text{node}} = ({}_{\text{node}}\mathbf{X}_{\text{DOF}})^{-1}$)

At the beam level, the force $\mathbf{f}_b \in R^{12}$ defined in equation 1 is applied on two rigid frames. So it is a composition of two wrenches $\mathbf{f}_{1,b}, \mathbf{f}_{2,b} \in se^*(3)$. In $\mathbf{f}_{1,b}$, the node of application of the wrench is 1 but the orientation is given by the beam frame \mathbf{X}_b .

If the node 1 is rigidly attached to a DOF i , we have to transport the point of application of the wrench and orientate the wrench in the world space. First, we rotate the wrench to obtain it the frame of node 1 $\mathbf{f}_{1,1} = {}_1R_b \mathbf{f}_{1,b}$. Then we can transpose the wrench in the node frame (both point of application and orientation): $\mathbf{f}_{i,1} = \mathbf{Ad}_{iX_1}^T * \mathbf{f}_{1,1}$. Where \mathbf{Ad}^T is the transposed Adjoint of the transformation ${}_iX_1$ between frame i and frame node 1. Finally, this wrench is rotated to obtain it the frame in the world coordinates $\mathbf{f}_{i,0} = \mathbf{R}_i^T \mathbf{f}_{i,1}$.

We gather and add all the wrenches defined on DOFs in the world coordinates to obtain the vector of internal elastic forces $\mathbf{F}(\mathbf{q})$, where $\mathbf{q} = \{\mathbf{X}_i\}$ gathers all the positions of the DOFs. The same transformations are used to compute the mass matrix $\mathbf{M}(\mathbf{q})$.

Figure 8, illustrates the placement of the Frame DOFs in the structure and the rigid transformations we have implemented between beam nodes and DOFs. To model the robot, we have used 41 beams to model the backbone and 123 beams to model the vertebrae. In total, we used 102 frame DOFs, so the size of the model is 612. The global system of equation follows the second law of

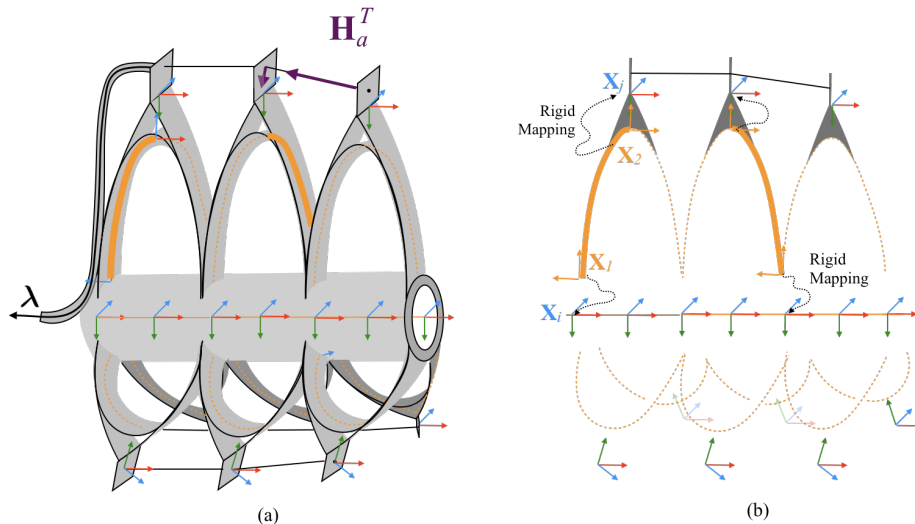


Figure 8: Model of a section with 3 vertebrae by assembling of beams. (a) shows the discretization of the structure in beam elements (in yellow). Two beams are highlighted. The actuation is modeled using constraints: \mathbf{H}_a^T provides the direction of the constraint and λ is the intensity of the pulling force. (b) illustrates the assembling of the two highlighted beams. The frame node \mathbf{X}_1 is attached to a frame DOF \mathbf{X}_i . Same for a frame node \mathbf{X}_2 rigidly attached to frame DOF \mathbf{X}_j .

Newton:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{R} - \mathbf{F}(\mathbf{q}) + \mathbf{H}_a^T(\mathbf{q})\lambda \quad (2)$$

where \mathbf{R} contains the known external forces like the gravity and $\mathbf{H}_a^T(\mathbf{q})\lambda$ represents the contribution of the actuators through cables. Like in [Largilliere et al(2015)], we use Lagrange multipliers to model the actuation. It allows to let as unknown λ the tension force applied on the cable. The distribution of this force on the DOFs depends on the geometrical configuration of the cable in the structure and the current position the robot \mathbf{q} . The way this distribution is computed is explained in more details in [Coevoet et al(2017)]. We can gather this distribution in a matrix $\mathbf{H}_a^T(\mathbf{q})$. Figure 8 illustrate the construction of the matrix \mathbf{H}_a for the robot.

4.3 Forward simulation

The range of the displacements applied on the beam during the deformation of the manipulator justifies the use of a non-linear corotational beam model. The model provides good results for large displacement but small strain. At each

time step of the simulation we update a linearisation of the internal forces:

$$\mathbf{F}(\mathbf{q}_{t+h}) = \mathbf{F}(\mathbf{q}_t) + \frac{\partial \mathbf{F}}{\partial \mathbf{q}}(\mathbf{q}_t) d\mathbf{q} \quad (3)$$

We use implicit Euler integration with these linearized forces, the time step is h and the current state is $(\mathbf{q}^t, \mathbf{v}^t)$ (with \mathbf{v} the velocities of the degrees of freedom). α and β are the two coefficients of Rayleigh (dynamic viscosity and static viscosity). We obtain the following system of equations:

$$\begin{aligned} & \underbrace{\left(\left(\frac{1}{h} + \alpha \right) \mathbf{M} + (\beta + h) \frac{\partial \mathbf{F}}{\partial \mathbf{q}} \right)}_{\mathbf{A}(\mathbf{q}_t)} d\mathbf{v} \\ & = \underbrace{-\mathbf{f}(\mathbf{q}_t) - h \frac{\partial \mathbf{F}}{\partial \mathbf{q}} \mathbf{v}_t + \mathbf{R} + \mathbf{H}_a^T(\mathbf{q}_t) \lambda}_{\mathbf{b}(\mathbf{q}_t, \mathbf{v}_t)} \end{aligned} \quad (4)$$

In practice, in the simulation, we used $\alpha = 1$ and $\beta = 0.1$. When solved, the value of the solution $d\mathbf{v}$ is used to update the Euler scheme: $\mathbf{v}_{t+h} = \mathbf{v}_t + d\mathbf{v}$ and $\mathbf{q}_{t+h} = \mathbf{q}_t + h\mathbf{v}_{t+h}$.

When solving the direct simulation problem, we want to impose reference lengths $\delta_a^*(t)$ to the cables and compute the new state of the robot as well as the new lengths of the cable. This is only a unilateral constraint, as we can only *pull* on the cable (not *push*). The effective length $\delta_a(\mathbf{q}_t)$ depends on the position of the degrees of freedom and can be written in the actuation space via a nonlinear mapping between motion space and actuation space. This nonlinear function can also be linearized at each time step using the expression of the matrix $\mathbf{H}_a(\mathbf{q}_t)$ at the current step:

$$\delta_a(\mathbf{q}_{t+h}) = \delta_a(\mathbf{q}_t) + \mathbf{H}_a(\mathbf{q}_t) d\mathbf{q} \quad (5)$$

Then we combine the equations of the deformable model and the cables:

$$\begin{cases} \mathbf{A} d\mathbf{v} - \mathbf{b} - h \mathbf{H}_a^T \lambda = 0 \\ \text{subject to :} \\ \lambda \geq 0 \\ \lambda \perp \delta_a + \mathbf{H}_a d\mathbf{q} - \delta_a^* \leq 0 \end{cases} \quad (6)$$

The symbol \perp states that each component of the vector λ that represents the forces in the cables is not null if and only if the cable is tight ($\delta_a + \mathbf{H}_a d\mathbf{q} - \delta_a^*(t) = 0$). On the other hand, when a cable is not tight ($\delta_a + \mathbf{H}_a d\mathbf{q} - \delta_a^*(t) < 0$), the cable cannot apply any forces $\lambda = 0$. To solve these equations, we first factorize the sparse matrix \mathbf{A} using a \mathbf{LDL}^T decomposition. Then we decompose $d\mathbf{v}$ in two elements $d\mathbf{v} = d\mathbf{v}^{\text{free}} + d\mathbf{v}^{\text{a}}$ where $d\mathbf{v}^{\text{free}} = \mathbf{A}^{-1} \mathbf{b}$ is the *free* motion (by fixing $\lambda = 0$ which represent a case where the cables are not pulling) and $d\mathbf{v}^{\text{a}} = h \mathbf{A}^{-1} \mathbf{H}_a^T \lambda$ is the contribution of the actuator. We integrate the velocity to obtain a *free* displacement $d\mathbf{q}^{\text{free}}$.

Then, we project the mechanics in the constraint space using the Schur complement of the equation (6):

$$0 \geq \underbrace{h^2 \mathbf{H}_a \mathbf{A}^{-1} \mathbf{H}_a^T}_{\mathbf{W}_{aa}} \lambda + \delta_a + \mathbf{H}_a d\mathbf{q}^{\text{free}} - \delta_a^* \perp \lambda \geq 0 \quad (7)$$

We can compute the value of λ using a Linear Complementarity Problem (LCP) solver. Once λ is computed, we can find the final value of $d\mathbf{v} = d\mathbf{v}^{\text{free}} + h\mathbf{A}^{-1}\mathbf{H}_a^T\lambda$ and equation (6) is solved for the current time step.

This approach, based on complementarity constraints, leads to realistic simulation of the cable/structure interaction. We could also use complementarity constraints to simulate dry friction but the level of friction was quite low on this robot and difficult to parameterize. Consequently, we suppose that we have frictionless interaction between the cable and the structure of the robot.

4.4 Inverse model

In our previous work [Largilliere et al(2015)] and [Coevoet et al(2017)], we have proposed to use convex optimization to find the inverse model of a soft robot modeled and simulated with finite element method. For the first time, we apply the approach for deformable models with frame nodes (i.e. nodes which position are defined in $SE(3)$), whereas previous approach were dealing only with node positions defined in R^3

For this robot, we have 10 actuators (9 cables and the translation of the base) and for the most simple case, we want to pilot only 5 degrees of freedom at the tip (3 translation and 2 rotation as the robot is not able to rotate by torsion along its main axis). For some inspection tasks, we may want to have more control on the posture of the robot, and add 2 additional directions on a node placed at the end of the second section, like explained in Figure 16. In both cases, the robot is redundant: there is more actuators than directions to be piloted in the task space. In some configurations, these redundancies decrease with cables at full-travel position and which can only act unilaterally.

Let \mathbf{X}_i^d be a desired position of a DOF frame i of the model, defined in the task space. This frame will be called *effector*. From the current position of the same frame in the FEM simulation \mathbf{X}_i , we deduce a twist $\mathbf{t}_i^e \in se(3)$ that represents the gap to the desired position. This twist is defined in the world coordinate. We suppose that the gap is not large and can be corrected in one step. In case of large distance between \mathbf{X}_i^d and \mathbf{X}_i , \mathbf{t}_i^e could represent a fraction of the gap and the desired position will be reached in several steps.

By selecting the directions that we want to pilot in the twists (translations along X, Y, Z and/or rotations along Y, Z), and gathering the effector frames we want to pilot, we define a gap of the effectors δ_e , that we want to minimize during the optimization. For instance for the configuration described in Figure

16, we have:

$$\delta_e = \begin{bmatrix} \mathbf{I}^{3 \times 3} & \begin{bmatrix} \mathbf{0}^{3 \times 3} \\ 0 \\ \gamma \\ 0 \end{bmatrix} \\ \mathbf{0}^{1 \times 3} & 0 \\ \mathbf{0}^{1 \times 3} & 0 \\ \mathbf{0}^{1 \times 3} & \begin{bmatrix} \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \\ \mathbf{0}^{1 \times 3} & \begin{bmatrix} \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \end{bmatrix} \mathbf{t}_3^e + \begin{bmatrix} \mathbf{0}^{3 \times 3} & \begin{bmatrix} \mathbf{0}^{3 \times 3} \\ \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \\ \mathbf{0}^{1 \times 3} & \begin{bmatrix} \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \\ \mathbf{0}^{1 \times 3} & \begin{bmatrix} \mathbf{0}^{1 \times 3} \\ \mathbf{0}^{1 \times 3} \end{bmatrix} \\ \mathbf{0}^{1 \times 3} & 0 \\ \mathbf{0}^{1 \times 3} & 0 \end{bmatrix} \mathbf{t}_2^e$$

where $\delta_e \in R^7$ represents the gap in the 7 directions that are defined in the task space, \mathbf{t}_3^e and \mathbf{t}_2^e stands for the twists defined the terminal points of section 3 and 2. The parameter γ is used to balance the errors in translation (defined in m) and in rotation (defined in rad).

The value of the twist \mathbf{t}_i^e linearly depends on $d\mathbf{q}$. So that at each step we can have the following linearized relationship:

$$\delta_e = \mathbf{H}_e d\mathbf{q} + \delta_e^0(\mathbf{q}_t) = h^2 \underbrace{\mathbf{H}_e \mathbf{A}^{-1} \mathbf{H}_a}_{\mathbf{W}_{ea}} \lambda + \underbrace{\mathbf{H}_e d\mathbf{q}^{\text{free}} + \delta_e^0}_{\delta_e^{\text{free}}} \quad (8)$$

Where \mathbf{H}_e and \mathbf{H}_a are constraint projection matrices from DOF space to effector space and from actuator space to DOF space.

The optimisation goal is to find the value of λ that minimizes the norm $\|\frac{1}{2}\delta_e\delta_e^T\|$. This optimisation is done by solving a QP (Quadratic Programming), instead of using the LCP presented in equation (7):

$$\begin{cases} \min \left(\frac{1}{2} \|h\mathbf{W}_{ea}\lambda + \delta_e^{\text{free}}\|^2 \right) \\ \text{subject to :} \\ \lambda \geq 0 \\ h\mathbf{W}_{aa}\lambda + \mathbf{H}_a d\mathbf{q}^{\text{free}} \leq 0 \end{cases} \quad (9)$$

The QP provides the value of λ and the following process is the same than for the direct case to obtain the final value of the motion $d\mathbf{v}$. As the number of actuators is greater than the size of the task space (i.e. the number of effector directions), the matrix of the QP is not positive-definite and the solution could be non-unique. To make the QP convex, we introduce in the minimization expression an additional energy term $E = h^2\lambda^T\mathbf{W}_{aa}\lambda$ which represents the mechanical work of the actuator forces. We then replace the equation (9) by:

$$\min \left(\left\| \frac{1}{2}\delta_e\delta_e^T \right\|^2 + eE \right) \quad (10)$$

with e chosen sufficiently small so that the deformation energy does not disrupt the quality of the effectors positioning. Note that the QP problem is based on a local model of the FEM thus to be able to achieve large deformation, the linearization point (and the associated matrices) are updated at each simulation

step. Moreover, since the model used by the QP is only valid locally, we bound the variation of the actuation ($-\delta\lambda \leq \lambda - \lambda_{prev} \leq \delta\lambda$).

5 Control

5.1 Description

The 9 tendons of the robot are actuated by 9 DC motors controlled in position with a sampling rate of 10ms. These inner loop controllers ensure displacements with an accuracy under the millimeter range. A full stroke displacement can be achieved in approximately 1 second. The sliding platform motion is achieved using stepper motors controlled in position.

The orientation and the position of both the tip of the robot and the end of the second section are sensed using a magnetic-based sensor (Polhemus Liberty <https://polhemus.com/motion-tracking/all-trackers/liberty>). The resolution of this sensor is around $0.1mm$ and $0.01deg$. This technology does not require a direct line of sight of the sensors (compared to vision based sensors) which allows the robot to inspect enclosed spaces as long as no magnetic disturbances are present in the working area (large iron objects, magnets, coils ...). These measurements can be updated with a rate up to 240Hz.

A desktop computer runs the main controller which is based on the FEM forward and inverse models. It communicates with the actuators and sensors via USB. The main controller updates the required tendon lengths and sliding position with a rate around 30Hz-40Hz depending on the simulation state.

5.2 Open loop control

A first way to control this robot is to consider the use of the inverse model as in [Duriez (2013)]. The references $r_i(t)$ are directly sent to the QP solver which computes the required actuation λ . During the QP solving step, we take into account the limitations of the actuators (both strokes $[\underline{\delta}_a, \overline{\delta}_a]$ and speed limits $[\underline{d\delta}_a, \overline{d\delta}_a]$), the solution is guaranteed to stay in the work space of the robot [and in the domain of validity of the QP problem](#) by adding the following constraints:

$$\begin{cases} \underline{\delta}_a \leq \delta_a(\mathbf{q}_{t+h}) \leq \overline{\delta}_a \\ \underline{d\delta}_a \leq \delta_a(\mathbf{q}_{t+h}) - \delta_a(\mathbf{q}_t) \leq \overline{d\delta}_a \end{cases} \quad (11)$$

Where $\delta_a(\mathbf{q}_{t+h}) = h\mathbf{W}_{aa}\lambda + \delta_a + \mathbf{H}_a d\mathbf{q}^{\text{free}}$ is the new cable lengths to apply and $\delta_a(\mathbf{q}_t)$ is the previous cable lengths applied.

Some experiments are performed in order to evaluate the quality of the inverse model. To do that, the objective of the QP problem is set to minimize the error between the references and the position of the tip of the robot as well as its orientation. Two test are performed: The first test consists in following step trajectories along the Y axis and remain still in the others; The second test consists in following a circle trajectory in the YZ plane.

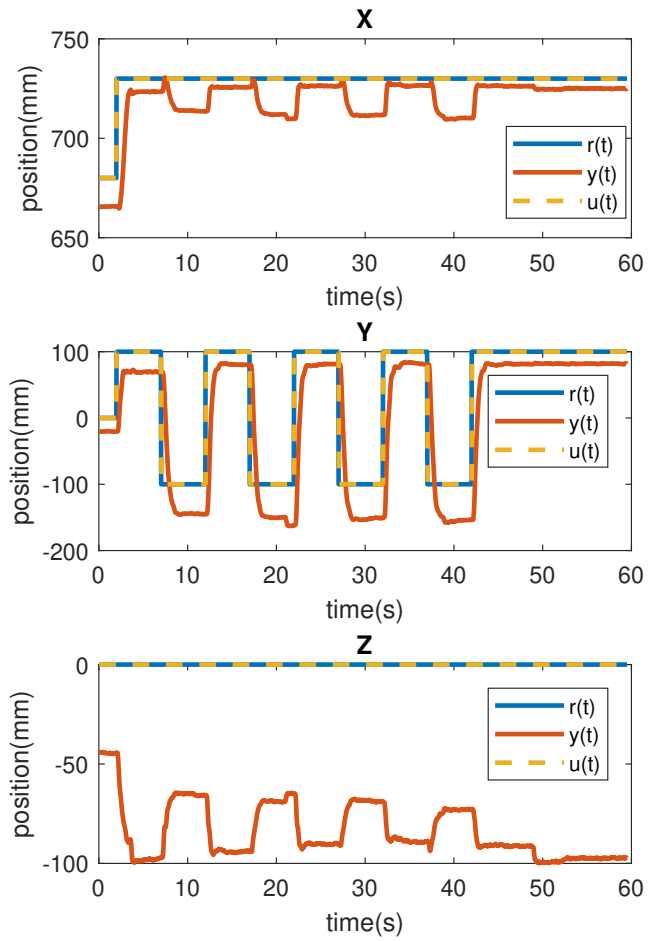


Figure 9: Open loop step response of the tip of the robot $y(t)$ and the associated QP inputs $u(t)$.

Figure 9 depicts the evolution of the position of the tip of the robot during the Y step response. The settling time is around 1s and the maximal steady state error around 50mm on the Y axis. Some oscillations and asymmetry in the behavior can be noticed which illustrate the non linear behavior of the system. Deflection is present on the X and Z axis which shows that the inverse model is not perfect due to unavoidable modeling inaccuracies. Even in the nominal case, i.e. without environment interactions like contacts, a robust output feedback controller is required to track the references more accurately.

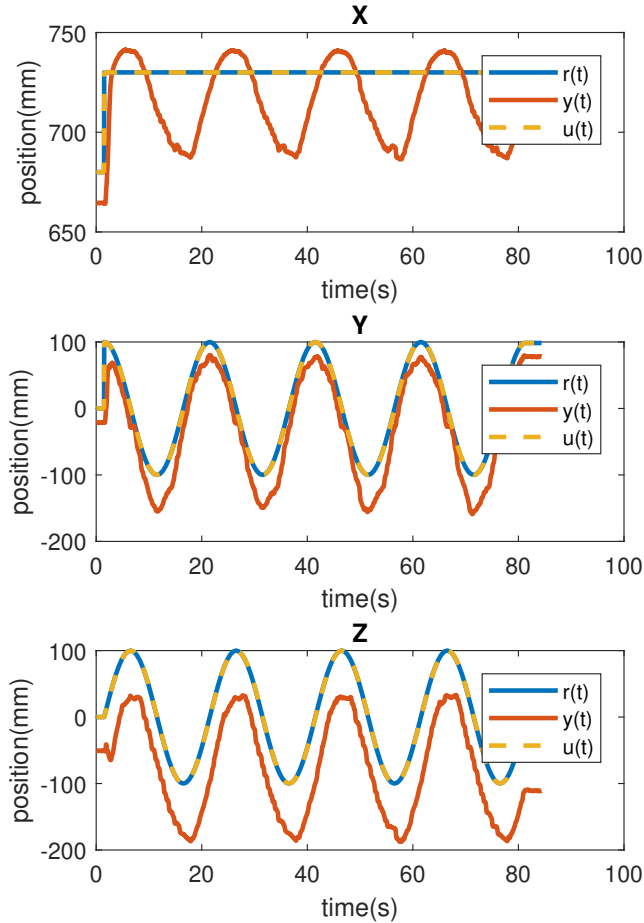


Figure 10: Open loop trajectory tracking of the tip of the robot $y(t)$ and the associated QP inputs $u(t)$.

5.3 Closed loop control

In order to improve the accuracy of the positioning of the robot, a closed loop control strategy is implemented. This strategy relies on the measurement coming from the Polhemus sensor and the use of the inverse model.

The main idea is to decouple the static input-output behavior of the robot by using the inverse model obtained in the previous section. Similar control schemes are used in [Morales-Bieze et al(2018), Morales-Bieze (2017), Largilliere et al(2015)] in which a QP-based inverse model and pure integral controllers are considered to control soft robots. The results presented in [Morales-Bieze et al(2018)] proved that, in the quasi-static case, this kind of control law have robustness properties towards the model inversion error. However these results are based on small speed assumptions which reduces the performances that can be obtained from this robot. Based on this knowledge and without any assumption on the speed magnitude during QP solving step, we implemented PI controllers in order to speedup the settling time while preserving some quality in the transient (limit the oscillations) and ensuring disturbance rejection.

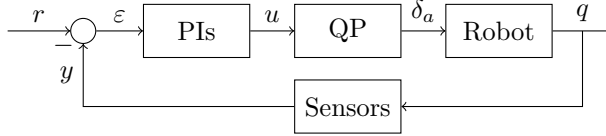


Figure 11: Control architecture. The block PIs represent a collection of PI controllers (one for each output), the QP block represent the inverse kinematic optimization under constraints. u is the vector of control signals, λ the actuation signals (tendon displacement), q the vector of coordinates of the robot, y are the measurements and r the references

The control strategy is the described by Figure 11. The outputs of the robot $q_i(t)$ (generalized coordinates) are measured by the magnetic sensor. These measurements $y_i(t)$ are compared to the references and send to the PI controllers (one for each coordinate). The control signals $u_i(t)$ computed by the PI controllers are send to the QP solver which computes the required actuation δ_a (tendon lengths and base position). Note that the number of actuators must be greater or equal than the number of outputs otherwise some PI controller may never converge and create unstable behavior.

$$\begin{aligned} u_i(t) &= K_{P_i} \varepsilon_i(t) + K_{I_i} \int \varepsilon_i(t) \\ \varepsilon_i(t) &= r_i(t) - y_i(t) \end{aligned} \quad (12)$$

Thanks to the decoupling provided by the inverse model, each PI controller controls a single-input/single-output system. Thus they can be tuned empirically or by identification-based method around a given set-point. In this paper, the second method is employed by considering second order transfer functions with input delay during the identification stage. Also, all these PI controllers

have clamping-based anti-windup scheme implemented to avoid unstable behavior around the limit of the working space of the robot.

5.4 Experiments

The efficiency of the closed loop is compared to the open loop strategy through 2 scenarios. These two cases consider the position and the orientation of the tip of the robot as the variables to be controlled. The first experiment consists in following a step reference signal on the Y axis (see also Extension 2). Figure 12 depicts the time response of the 3 controlled variables and shows how the control input $u(t)$ provided to the QP solver compensates the external disturbances and modeling error. The results show a steady state error close to the sensor sensitivity (few mm). Figure 13 and Table 1 provide a comparison of the performances between the open loop control scheme and the closed loop. It appears that the closed loop improves the accuracy and the settling time. The closed loop also reduces the uncompensated coupling effect which is the most noticeable on the z axis (see Figure 13).

Axis	control	mean error	std. dev.	set. time
X	Open loop	12.6	11.47	
	Closed loop	1.61	4.82	
Y	Open loop	30.44	57.1	0.96s
	Closed loop	0.62	55.35	0.53s
Z	Open loop	80.63	14.34	
	Closed loop	0.001	7.22	

Table 1: Open loop and closed loop performances in terms of mean error, error standard deviation and settling time (10%)

The second case consists in following a circle trajectory in the YZ plane, as shown in Extension 3. The frequency of the trajectory is set to 0.05Hz. Figures 14 and 15 depicts the evolution of the position of the tip of the robot with respect to time and in the YZ plane respectively. Table 2 shows the performance indexes of the open loop and the closed loop in terms of mean error and standard deviation of the error signal. As in the step case, the closed loop improves the accuracy and is able to track the sinusoidal signal with less deflection than the open loop.

A final experiment is performed with a more practical goal. The robot task is to inspect through holes of a cube (see Extension 1). To do so, the position of the second section and the orientation of sections 2 and 3 are considered as the variables to be controlled. This setup is showed in Figure 16. This choice allows to reach a given position for the second section and orientate the camera placed at the tip of the robot inside the cube. This experiment shows that the controlled variables can be changed according to the robot task, only the PI control gains need to be re-tuned.

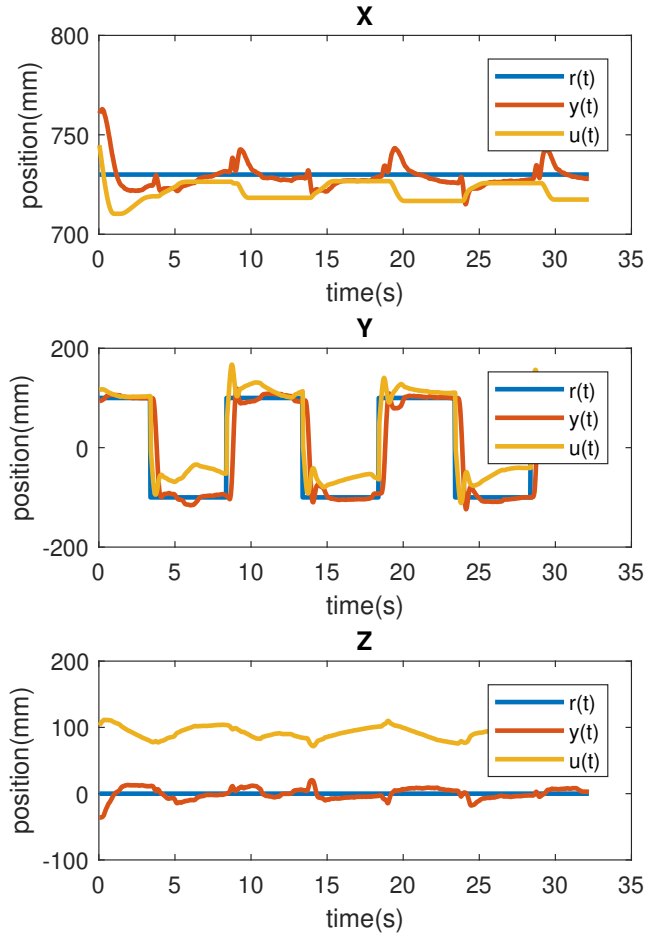


Figure 12: Evolution of the position of the tip of the robot $y(t)$ and the associated QP inputs $u(t)$ in closed loop

Axis	control	mean error	std. dev.
X	Open loop	14.9	19.72
	Closed loop	0.24	8.90
Y	Open loop	31.31	14.72
	Closed loop	0.35	11.70
Z	Open loop	76.59	13.06
	Closed loop	0.042	9.04

Table 2: Open loop and closed loop performances en terms of mean error and error standard deviation when the robot follows a circle trajectory in the YZ plane

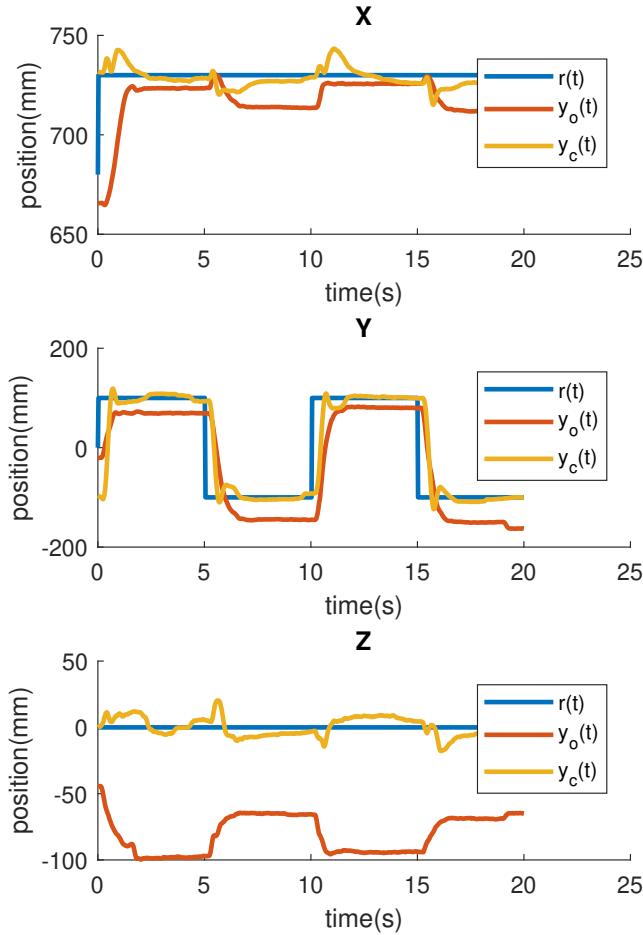


Figure 13: Evolution of the position of the tip of the robot in open loop y_o and closed loop y_c

6 Tele-operation and haptic rendering

The direct analogy that exist between the human arm and rigid manipulators provides an easy and obvious method of human operation for this type of robots. In contrast, human operation of continuum robots remains a non-intuitive task due to the need to coordinate their (infinite) degrees of freedom and their coupled actuation ([Censits et al. (2005)], [McMahan W and Walker (2009)]). In order to provide a method of tele-operation for the manipulator presented in this paper, a Touch Haptic device (3D Systems <https://www.3dsystems.com/haptics-devices/touch>) is linked to the degrees of freedom of the end-effector of the robot in the simulation. By manipulating the stylus of the haptic device, the user can control the position and orientation of the end-effector of the robot. Addition-

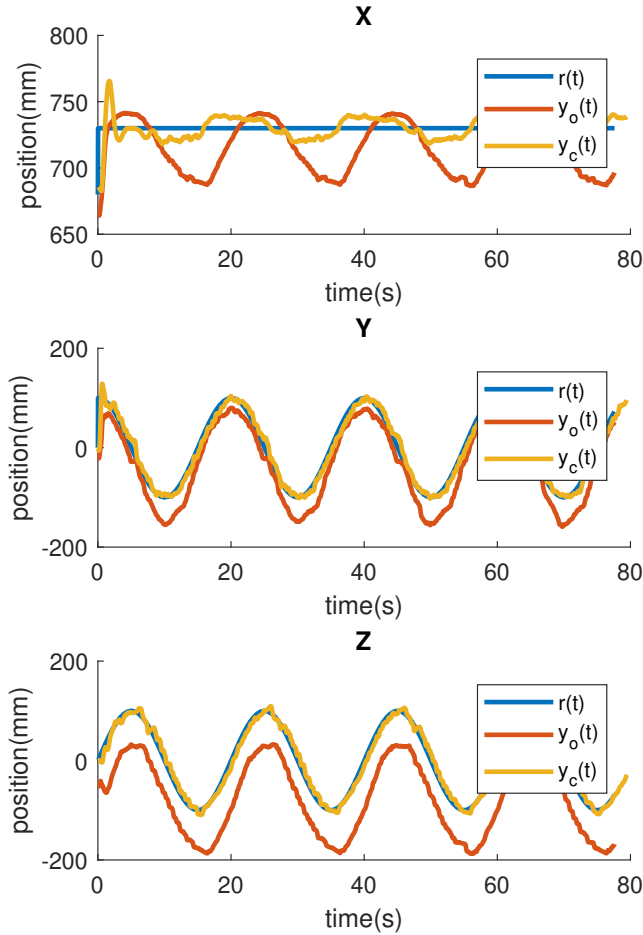


Figure 14: Evolution of the position of the tip of the robot in open loop y_o and closed loop y_c

ally, a haptic feedback loop provides the user with the feeling of the external forces acting on the robot. The implementation of the haptic feedback is showed in figure 17.

A basic solution to provide a haptic rendering is to apply a force on the stylus that is proportional to the difference between the reference position provided by the user and the measured position of the robot. However this strategy creates unwanted drag forces during fast movement due to the response time of the robot and its actuators. To overcome this problem, a filtering strategy is proposed based on the knowledge of the closed loop behavior of the robot. Instead of considering directly the reference position, the force feedback is computed from the predicted position if the robot was free to move. A gain G allows to tune the strength of the haptic feedback.

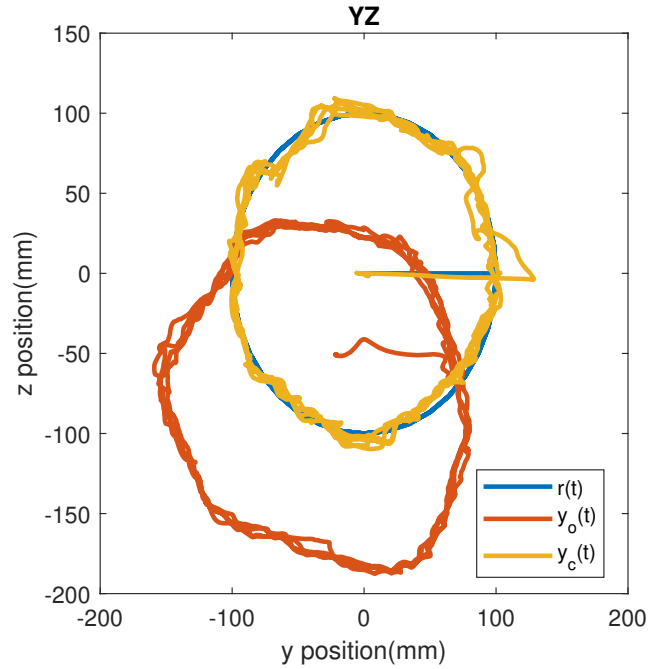


Figure 15: Evolution of the position of the tip of the robot in open loop y_o and closed loop y_c

To check the validity of this approach, we consider the following cases : slow and fast motions (resp. $< 5mm/s$ and $> 10mm/s$) in which the user should feel nearly zero force, contact with a wall in which the user should feel when the contact occurs and finally moving an object (see Extension 4) in which the user should feel a friction-like resistance. The results of these experiments are depicted in figures 18 and 19. During the experiments we can notice that the operator can feel the wall contact along the Z axis at $t = 92s$. When the operator moves away from the wall there is a stick effect which is typical when a PI controller is considered (the integrator tries to reject the wall). When the operator tries to move an object ($t = 138s$), he can feel an opposing force as long as the object is still as the force applied is not enough to make the object move. When the force applied by the robot on the object is high enough, the object starts to move and the object friction forces suddenly reduce, which generate an accompanying haptic force due to inertia of the robot. Finally, during the free movement phases ($t < 90s$ and $105s < t < 138s$), the operator can feel some small parasitic forces which are filtered by our model based approach strategy.

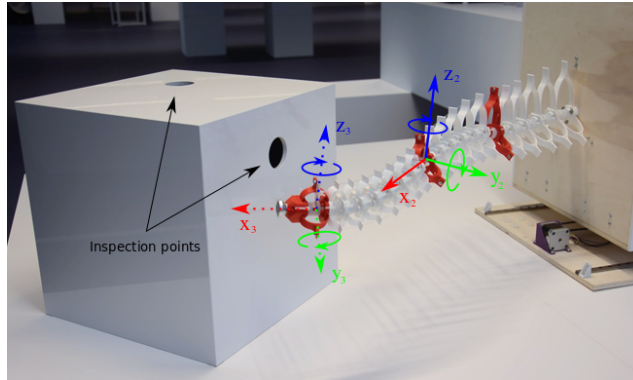


Figure 16: The inspection task. The inspection entry points and the degrees of freedom selected as variables to be controlled. For this particular task, the position of the terminal point of sections 2 and the orientation of both section 2 and 3 are the controlled variables.

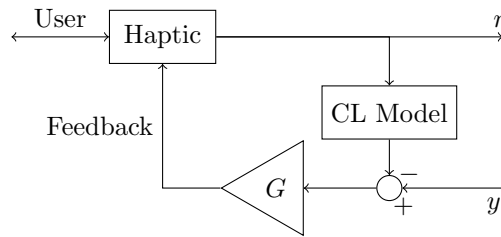


Figure 17: Haptic feedback architecture is based on the closed loop model

7 Discussion

Note that the proposed approach aims at being generic. The use of finite element methods permit the model of parallel structures and to include new structural parts incrementally without having to retrieve the new equations or parameters of the system. To illustrate this feature, consider the robotic arm to which we have added mechanical constrains (black trusses on figures 20) between some selected ribs to cause partial curvatures along a single section and asymmetric deformation. In order to include this modification using the methodology presented, it is enough to just add the new constraints into the model description (which is 3 lines of codes in the simulation framework).

In a framework like Coserat rods or constant curvature model, this would necessitate the addition of more sections and to take into account the coupling between non adjacent sections (the parallelism) and reconsider the controller design, all this probably by hand. Moreover, an important benefit of the use of the same model all along the design (modeling, simulation and control design) is the ability to define only once these new constraints as they will propagate all

these modifications automatically to the control law design. This reduces the development time and risk of error by a non negligible factor.

To illustrate the importance of taking into account these structural modifications we made another experiment. This experiment consists in trying to reach a given position using the inverse model in open loop. During this, we measured the error of the tip position between the model (which always reach the good position) and the real robot when the trusses are not taken into account (but present in the robot) and when they are. Table 3 and figures 20 and 21 show the obtained modelling errors. It appears that the positioning error in open loop is 4 times more important in the case where the trusses are not modeled.

Axis	w.o. trusses	with trusses
X	-187mm	33mm
Y	70mm	-44mm
Z	-18mm	-6mm
norm	200mm	55mm

Table 3: Open loop modelling error with and without taking into account the truss

Note that this feature could be used in the case of mechanical failure. For example if a tendon breaks and we detect it, the simulation and the QP problem could be updated to take this new situation into account.

The current implementation of the controller does not include a correction of the simulated model (used to build the QP problem). But in one of our previous work [Morales-Bieze (2017)], we proved for a similar control that this kind of control scheme is robust to modeling error (expressed as an error on the actuator space to the effector space Jacobian matrix). This previous study showed that as long as the main entries of the Jacobians (of the model and of the real robot) have the right sign, there exist a controller that guarantee the convergence of the error to 0. Given the errors observed in open loop (10% of the length of the robot) we are always in this case. If the errors were greater, we have also presented a method to refresh the simulated state of the robot based on sensor information, thanks to a second inverse model. But that was not needed in this case.

8 Conclusion and future work

A novel design of an extrinsically actuated deformable continuum manipulator inspired by the skeletal spine and the engineering lattice is presented. The design makes use of a hollow backbone to prevent the shearing between the sections of the robot and the self-collision of the vertebrae to limit the local deformation. The simulation scheme used to unify the different stages in the implementation of the robot is showcased. Lastly the experimental validation of the modeling

and control strategies is presented, as well as a proposed haptic feedback loop to enhance the tele-operation capabilities of the controller.

We are currently designing the next generation of the manipulator. In this iteration, some improvements over the current hardware are planned: We plan to add two additional sections to the manipulator to further enhance its workspace. In order to optimize the dimensions of the back end of the robot, we are currently working on the design of a single pulley system with a tension sensor to ensure that the tendon is always under a minimum tension to avoid slacking and to improve the control of the manipulator. The magnetic sensors currently employed to measure the position and orientation of the robot proved to be very sensitive to the noise induced by the endoscopic camera. We will replace these sensors with string potentiometers and flex sensors to reconstruct the state of the robot and use the estimated states to improve the inverse model used to control the manipulator.

We plan to improve the control of the robot by accounting for the deformations of the cables in the FEM model as well as implementing a dynamic controller ([Thieffry et al. (2019)]) to reduce vibrations and allow for the imposition of the acceleration during the trajectory of the manipulator. The implementation of a dynamic controller, when based on a reduced order model, will also decrease the computational requirements of this methodology. We plan also to include an automatic control gain tuning procedure in our framework. Finally, we plan to complement the model of the manipulator by including the self collision of the vertebrae and the use of a non-linear constitutive material law to account for plasticity phenomenon.

9 Appendix A: Index to Multimedia Extensions

Extension	Type	Description
1	Video	Description of the manipulator and inspection scenario
2	Video	Point-to-point trajectory tracking experiment
3	Video	Circle trajectory tracking experiment
4	Video	Teleoperation with haptic feedback loop

10 Appendix B: Table of variables

The authors would like to thanks Fabien Verbrughe and Gilles Marguerite from Centrale Lille for their valuable advice.

Symbol	Description	Type
\mathbf{X}_i	Beam extremity frame	Computed
\mathbf{X}_b	Beam frame	Computed
\mathbf{X}_i^d	Frame desired position	User imposed
\mathbf{p}_i	Spline control point	Computed
\mathbf{K}_b	Beam stiffness matrix	Computed
\mathbf{f}_b	Beam internal forces	Computed
\mathbf{u}_{loc}	Beam nodal displacement	Computed
\mathbf{T}_b	Beam rotation matrix	Computed
\mathbf{t}_e	Effector twist	Computed
$\mathbf{A}\mathbf{d}^T$	Transposed adjoint matrix	Computed
\mathbf{F}	Internal elastic forces vector	Computed
\mathbf{q}	Frame positions vector	Computed
\mathbf{v}	Frame velocities vector	Computed
\mathbf{M}	Mass matrix	Computed
\mathbf{R}	External forces	User imposed
\mathbf{H}_a	Constraint direction matrix	User imposed
λ	Actuator force vector	Computed
h	Simulation time step	User imposed
α	Dynamic viscosity	User imposed
β	Static viscosity	User imposed
δ_a^*	Tendon reference length	User imposed
δ_a	Tendon effective length	Computed
δ_e	Effector gap	Computed
$d\mathbf{q}^{free}$	Frame free displacement	Computed
$d\mathbf{v}^{free}$	Frame free motion	Computed
E	Deformation energy	Computed
r	System reference	User imposed
ϵ	System error	Computed
y	System measured output	Measured
K_P	Controller proportional gain	User imposed
K_I	Controller integral gain	User imposed

References

- [Alambeigi et al(2017)] Alambeigi F, Wang Y, Sefati S, Gao C, Murphy RJ, Iordachita I, ... and Armand, M (2017) *A curved-drilling approach in core decompression of the femoral head osteonecrosis using a continuum manipulator* IEEE Robotics and Automation Letters, (pp. 1480-1487)..
- [Ataka et al(2016)] Ataka A, Qi P, Shiva A, Shafti A, Wurdemann H, Dasgupta P, and Althoefer K (2016) *Towards safer obstacle avoidance for continuum-style manipulator in dynamic environments* In Proceedings of the IEEE

RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics. Vol. 2016-July, pp. 600-605.

- [Bajo and Simaan(2010)] Bajo A, and Simaan N. (2010) *Finding lost wrenches: Using continuum robots for contact detection and estimation of contact location* In Robotics and Automation (ICRA), 2010 IEEE International Conference on (pp. 3666-3673).
- [Bardou et al(2010)] Bardou B, Zanne P, Nageotte F and de Mathelin M (2010) *Control of a multiple sections flexible endoscopic system* IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [Braganza et al(2007)] Braganza D, Dawson DM, Walker ID, and Nath N (2007) *A neural network controller for continuum robots* IEEE transactions on robotics, (pp. 1270-1277).
- [Calisti et al(2012)] Calisti M, Arienti A, Renda F, Levy G, Hochner B, Maz-zolai B ... and Laschi C (2012) *Design and development of a soft robot with crawling and grasping capabilities* In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 4950-4955).
- [Cianchetti et al(2013)] Cianchetti M, Ranzani T, Gerboni G, De Falco I, Laschi C, and Menciassi A (2013) *STIFF-FLOP surgical manipulator: mechanical design and experimental characterization of the single module* In Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on (pp. 3576-3581).
- [Cojocaru et al(2010)] Cojocaru D, Ivanescu M, Tanasie RT, Dumitru S and Manta F (2010) *Experiments with Tentacle Robots* In ISR/ROBOTIK, pp. 1-6.
- [Coevoet et al(2017)] Coevoet E, Morales-Bieze T, Largilliere F, Zhang Z, Thieffry M, Sanz-Lopez M, Carrez B, Marchal D, Goury O, Dequidt J and Duriez C (2017) *Software toolkit for modeling, simulation, and control of soft robots*, Advanced Robotics, vol 31, no. 22, pp. 1208-1224.
- [Crisfield (1990)] Crisfield MA (1990) *A consistent co-rotational formulation for non-linear, three-dimensional, beam-elements* Computer methods in applied mechanics and engineering volume 81 num 2 (pp 131-150).
- [Csencsits et al. (2005)] Csencsits M, Jones BA, McMahan W, Iyengar V and Walker ID *User interfaces for continuum robot arms* In 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp 3123-3130).
- [Duriez (2013)] Duriez C (2013) *Control of elastic soft robots based on real-time finite element method*, Robotics and Automation (ICRA), 2013 IEEE International Conference on, pp. 3982-3987.

- [Escande et al(2012)] Escande C, Merzouki R, Pathak PM, and Coelen V (2012) *Geometric modelling of multisection bionic manipulator: Experimental validation on robotinoxt* In Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on (pp. 2006-2011).
- [Escande et al(2015)] Escande C, Chettibi T, Merzouki R, Coelen V, and Pathak PM (2015) *Kinematic calibration of a multisection bionic manipulator* IEEE/ASME transactions on mechatronics, (pp. 663-674).
- [Fraś et al(2015)] Fraś J, Czarnowski J, Maciaś M, Główka J, Cianchetti M, and Menciassi A (2015) *New STIFF-FLOP module construction idea for improved actuation and sensing* In Robotics and Automation (ICRA), 2015 IEEE International Conference on (pp. 2901-2906).
- [Giorelli et al(2015)] Giorelli M, Renda F, Calisti M, Arienti A, Ferri G, and Laschi C (2015) *Neural network and jacobian method for solving the inverse statics of a cable-driven soft arm with nonconstant curvature* IEEE Transactions on Robotics, (pp. 823-834).
- [Gravagne and Walker(2003)] Gravagne IA, Rahn CD and Walker ID (2003) *Large deflection dynamics and control for planar continuum robots* IEEE/ASME Transactions on Mechatronics, volume 8, number 2.
- [Greer et al (2018)] Greer JD, Morimoto TK, Okamura AM and Hawkes EW (2018). *A Soft, Steerable Continuum Robot That Grows via Tip Extension* Soft robotics, volume 6, number 1, pp.95-108.
- [Jones and Walker(2006)] Jones BA and Walker ID (2006) *Kinematics for multisection continuum robots* IEEE Transactions on Robotics, (pp. 43-55).
- [Largilliere et al(2015)] Largilliere F, Verona V, Coevoet E, Sanz-Lopez M, Dequidt J and Duriez C (2015) *Real-time control of soft-robots using asynchronous finite element modeling* IEEE International Conference on Robotics and Automation, vol. 2015–June, no. June.
- [Laschi et al(2009)] Laschi C, Mazzolai B, Mattoli V, Cianchetti M, and Dario P. (2009) *Design of a biomimetic robotic octopus arm* Bioinspiration & biomimetics.
- [Ma and Dollar(2011)] Ma RR and Dollar AM (2011) *On dexterity and dexterous manipulation* In Advanced Robotics (ICAR), 2011 15th International Conference on (pp. 1-7).
- [Marchese and Rus(2016)] Marchese AD and Rus D (2016) *Design, kinematics, and control of a soft spatial fluidic elastomer manipulator* The International Journal of Robotics Research (IJRR), 35(7), (pp. 840–869).
- [McMahan W and Walker (2009)] McMahan W and Walker ID (2009) *Octopus-inspired grasp-synergies for continuum manipulators* In 2008 IEEE International Conference on Robotics and Biomimetics (pp 945-950).

- [McMahan et al(2006)] McMahan W, Chitrakaran V, Csencsits M, Dawson D, Walker ID Jones BA, ... and Rahn CD (2006) *Field trials and testing of the OctArm continuum manipulator* In Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on (pp. 2336-2341).
- [Mahoney et al(2016)] Mahoney AW, Anderson PL, Swaney PJ, Maldonado F, and Webster RJ (2016) *Reconfigurable parallel continuum robots for incisionless surgery* In Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on (pp. 4330-4336).
- [Mehling et al(2006)] Mehling JS, Diftler MA, Chu M and Valvo M (2006) *A minimally invasive tendril robot for in-space inspection* IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics.
- [Morales-Bieze (2017)] Morales-Bieze T (2017), *Contribution to Kinematic Modeling and Control of Soft Manipulators using Computational Mechanics* PhD defence, Lille University of Science and Technology.
- [Morales-Bieze et al(2018)] Morales-Bieze T, Largilliere F, Kruszewski A, Merzouki R and Duriez C (2018), *FEM-based kinematics and closed-loop control of soft, continuum manipulators* soft Robotics.
- [Orekhov et al(2016)] Orekhov AL, Black CB, Till J, Chung S, and Rucker DC (2016) *Analysis and validation of a teleoperated surgical parallel continuum manipulator* IEEE Robotics and Automation Letters, (pp. 828-835).
- [Przemieniecki (1985)] Przemieniecki JS (1985) *Theory of matrix structural analysis*, Courier Corporation publisher, 1985.
- [Renda et al(2014)] Renda F, Giorelli M, Calisti M, Cianchetti M and Laschi C (2014) *Dynamic model of a multibending soft robot arm driven by cables* IEEE Transactions on Robotics, vol. 30, (pp. 1109-1122).
- [Rolf et al(2010)] Rolf M, Steil JJ, and Gienger M (2010) *Goal babbling permits direct learning of inverse kinematics* IEEE Transactions on Autonomous Mental Development, (pp. 216-229).
- [Rolf and Steil(2012)] Rolf M, and Steil JJ (2012) *Constant curvature continuum kinematics as fast approximate model for the Bionic Handling Assistant* In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on (pp. 3440-3446).
- [Rolf and Steil(2014)] Rolf M and Steil JJ (2014) *Efficient exploratory learning of inverse kinematics on a bionic elephant trunk* IEEE transactions on neural networks and learning systems, (pp. 1147-1160).
- [Simaan and Flint(2004)] Simaan N, Taylor R and Flint P (2004) *A dexterous system for laryngeal surgery* IEEE International Conference on Robotics and Automation.

- [Thieffry et al. (2019)] Thieffry M, Kruszewski A, Duriez C, and Guerra TM *Control Design for Soft Robots Based on Reduced-Order Model* IEEE Robotics and Automation Letters volume 4 num 1 (pp 25-32).
- [Tonapi et al(2014)] Tonapi MM, Godage IS, and Walker ID (2014) *Next generation rope-like robot for in-space inspection* In Aerospace Conference, 2014 IEEE (pp. 1-13).
- [Torres and Alterovitz(2011)] Torres LG and Alterovitz R (2011) *Motion planning for concentric tube robots using mechanics-based models*. In International Conference on Intelligent Robots and Systems (IROS).
- [Yoon et al(2011)] Yoon H, Oh SM, Jeong JH, Lee SH, Tae K, Koh KC and Yi BJ (2011) *Active bending endoscope robot system for navigation through sinus area* IEEE/RSJ International Conference on Intelligent Robots and Systems.
- [Walker and Hannan(1999)] Walker ID and Hannan MW (1999) *A novel 'elephant's trunk' robot*, IEEE/ASME International Conference on Advanced Intelligent Mechatronics.
- [Walker et al(2005)] Walker ID, Dawson DM, Flash T, Grasso FW, Hanlon RT, Hochner B, ... and Zhang QM (2005) *Continuum robot arms inspired by cephalopods* In Unmanned Ground Vehicle Technology VII (Vol. 5804, pp. 303-315).
- [Webster and Jones(2010)] Webster RJ and Jones BA (2010) *Design and kinematic modeling of constant curvature continuum robots: A review* The International Journal of Robotics Research.
- [Wilkening et al(2011)] Wilkening P, Alambeigi F, Murphy RJ, Taylor RH, and Armand M (2017) *Development and experimental evaluation of concurrent control of a robotic arm and continuum manipulator for osteolytic lesion treatment* IEEE robotics and automation letters, (pp. 1625-1631).
- [Zhao and Gao(2010)] Zhao Q and Gao F (2010) *Design and analysis of a kind of biomimetic continuum robot* IEEE International Conference on Robotics and Biomimetics.

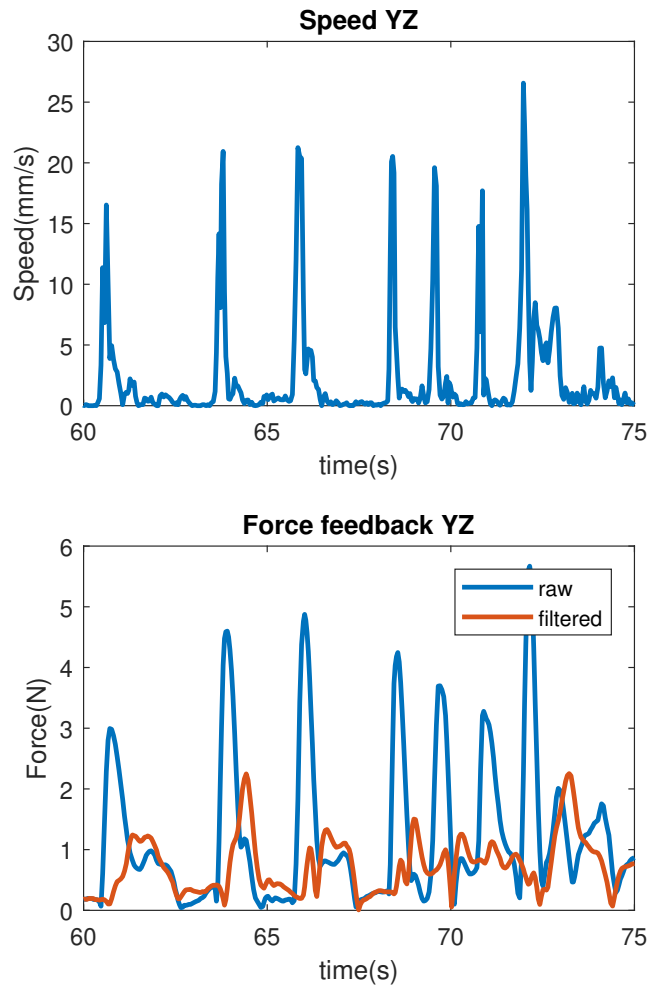


Figure 18: Evolution of the haptic device force feedback with (red) and without using the closed loop model (blue) in various situations

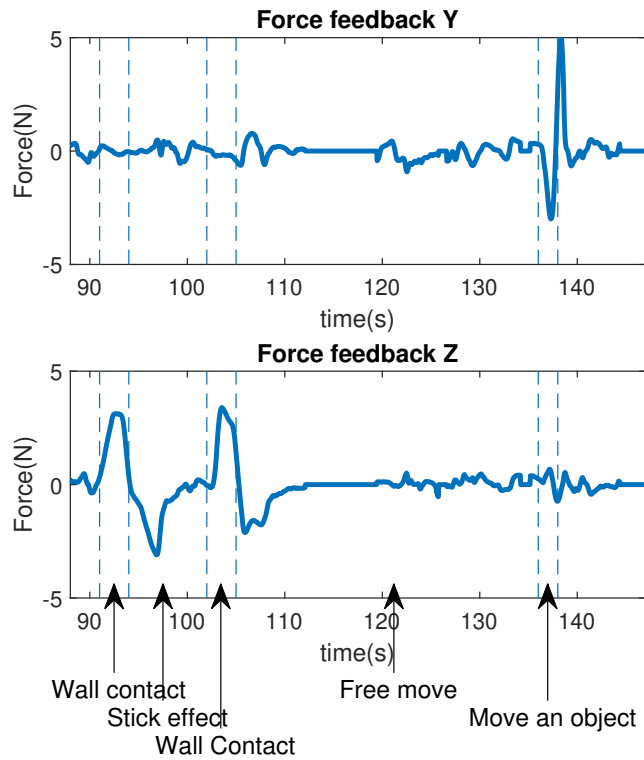


Figure 19: Evolution of the haptic device force feedback when contact with the environment occurs: Wall collision on the Z axis and moving an object along the Y axis (see also Extension 4)

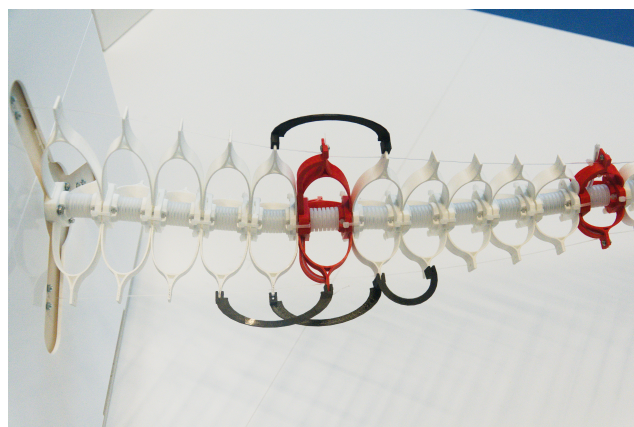


Figure 20: The robot with new structural constrains

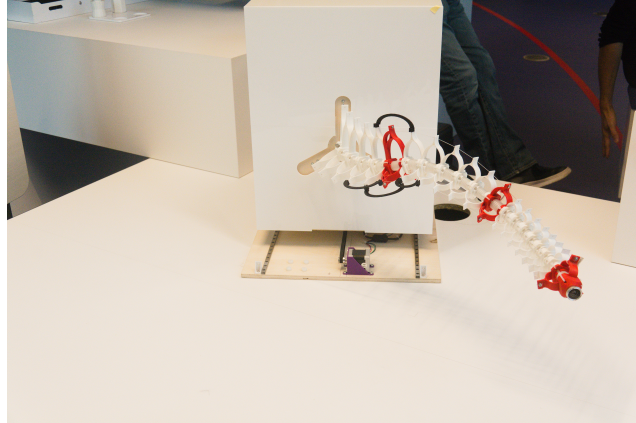


Figure 21: Inverse model open loop experiment when not taking into account the trusses

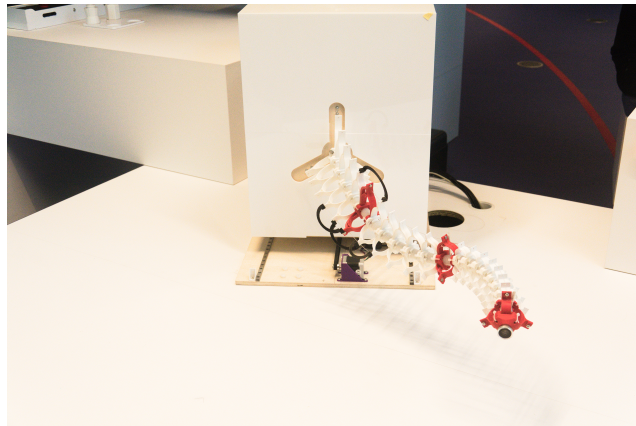


Figure 22: Inverse model open loop experiment when taking into account the trusses