Finite element method-based kinematics and closed-loop control of soft, continuum manipulators

Thor Morales Bieze, Frederick Largilliere, Alexandre Kruszewski, Zhongkai Zhang, Rochdi Merzouki, Christian Duriez

FEM-based kinematics and closed-loop control of soft, continuum manipulators

Thor Morales Bieze, Frederick Largilliere, Alexandre Kruszewski, Zhongkai Zhang, Rochdi Merzouki and Christian Duriez

***Abstract*—This paper presents a modeling methodology and experimental validation for soft[1] manipulators to obtain forward and inverse kinematic models under quasistatic conditions. It offers a way to obtain the kinematic characteristics of this type of soft robots that is suitable for offline path planning and position control. The modeling methodology presented relies on continuum mechanics which does not provide analytic solutions in the general case. Our approach proposes a real-time numerical integration strategy based on Finite Element Method (FEM) with a numerical optimization based on Lagrangian Multipliers to obtain forward and inverse models. To reduce the dimension of the problem, at each step, a projection of the model to the constraint space (gathering actuators, sensors and end-effector) is performed to obtain the smallest number possible of mathematical equations to be solved. This methodology is applied to obtain the kinematics of two different manipulators with complex structural geometry. An experimental comparison is also performed in one of the robots, between two other geometric approaches and the approach that is showcased in this paper. A closed-loop controller based on a state estimator is proposed. The controller is experimentally validated and its robustness is evaluated using Lypunov stability method.**

***Index Terms*—Soft manipulators, Continuum robots, Soft robots, Finite Element Method and Robotic control**

I. Introduction

For the past four decades, rigid-link manipulators have been successfully deployed in the industrial environment. Their rigid bodies and high-torque joints are perfectly fitted to perform tasks that involve accurate positioning while carrying considerable payloads. However, as the applications for these systems move away from this structured environment, traditional rigid manipulators have been less successful. Indeed, their rigid and bulky bodies is a problem for adaptation to dynamic environments.

Roboticists, trying to cope with the new applications for manipulators, have turned their attention to nature, seeking for inspiration to design new robot manipulators. Soft manipulators are robots often inspired by the morphology and functionality of biological agents like octopus tentacles and elephant trunks [1] [2] [3] [4] or tendrils [5] [6]. This type of manipulator deforms continuously to achieve a certain pose and can exhibit key advantages over their rigid counterpart with suitable design: they are lighter and therefore have less energy consumption, present a bigger power-to-weight ratio as well as a natural compliance because of their material properties. This compliance also gives the manipulators the ability to better adapt themselves to dynamic work environments and to work side by side with humans, without the concern of hazardous collisions. Because of these characteristics, soft manipulators have found a niche of applications in the medical field [7] [8] [9] [10] [11].

The compliant nature of soft continuum manipulators comes with the issues of modeling and control of their behavior, which is highly non-linear. A popular approach to model continuum robots has been the modification of methods already established to model rigid manipulators. In [12], Hannan and Walker presented the development of the kinematic model for a trunk robot. The model considers that each section bends with constant curvature. This approach has been used to express the kinematics of continuum trunks [13] and tendril-like continuum robots [14]. The constant curvature models can be used even when the shape of the continuum robot does not conform to a circular arc. In [15], the kinematics of the bionic handling assistant are obtained by modeling each section of the robot as a finite number of serially connected circular arcs with different parameters each. The models mentioned, while producing close-form analytic models, are only based on the geometry of the robot, without consideration for the mechanics of the structure, necessary to properly describe the deformation of this type of robot.

A. Model requirements

In contrast with rigid robots, soft manipulator kinematics not only depend on the geometry of the robot, but also on its mechanical properties, in particular the stiffness of the structure. While rigid manipulator kinematics can be used to solve positioning problems with the assumption of resistance/counter-actuation to gravity or load effects, soft manipulators easily comply to these forces and deform. To answer the same problems of positioning, it is then necessary to take into account the current deformation (ie change of geometry) induced by these forces to obtain a kinematic relation between position of end-effector and position of actuators. (Fig. 1).

The degrees of freedom in a rigid manipulator are determined by the joints of the manipulator and are often all actuated. In soft manipulators, the degrees of freedom are generated by the deformation of the continuum and their number is infinite. (It can be noted that it is disconnected from

The authors are affiliated to Univ. Lille, INRIA, CNRS, Centrale Lille, UMR 9189 - CRIStAL - Centre de Recherche en Informatique, Signal et Automatique de Lille, F-59000 Lille, France.
Corresponding authors: thor.bieze@polytech-lille.fr, christian.duriez@inria.fr

[1]In the literature, these manipulators are usually classified as continuum robots. However, their main characteristic of interest in this paper is that they create motion by deformation, as opposed to the classical use of articulations.

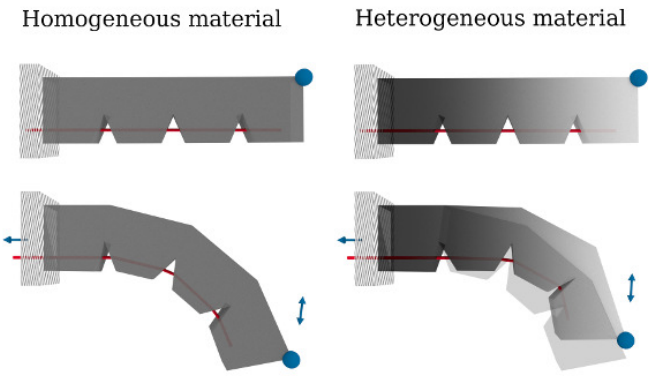

Fig. 1: In this example a tendon is pulled to create the motion of an elastic soft robot. Starting with the same geometry, the material stiffness has an influence on the kinematics (output vs input displacements).

the number of actuators). Usually this problem is addressed by a discretization of the degrees of freedom of the continuum, through methods provided by computational mechanics. Another difference in soft manipulators, compared to rigid ones, is that if a load is carried by a soft manipulator, this load will deform the robot and modify its kinematics.

B. Related Work

This recapitulation of previous work is centralized in the use of continuum mechanics in the modeling of soft manipulators. A discussion on the use of mechanics-based methods to describe soft continuum robots can start by mentioning the work of Chirikjian [16] [17], who used continuum mechanics to define a curve that describes the pose of hyper-redundant robots without considering actuation. This work laid the basis for subsequent work on continuum manipulator models.

Given the uprising trend in the design of manipulators composed by a single elongated backbone actuated by tendons, researchers have explored beam theory-based approaches to describe the pose of this particular type of robots. Rucker et al. present in [18] the potential of this theory applied to the modeling of manipulators with tendon routing. In [19] the compliance model of continuum robots is obtained by considering the robot as a single section of Cosserat rod. In [20], Jones presents a static model in three-dimensional space and in [21], this theory was implemented to compute the inverse kinematic model of a tendon-driven tentacle manipulator in two dimensions under Euler-Bernoulli beam hypothesis. While providing models suitable for fast computing, beam theory is limited in its application by the shape of the backbone, i.e. when the backbone cannot be assimilated as a uniform beam, this modeling approach loses relevance. In particular, when the body of the robot is actuated intrinsically by pneumatic or hydraulic actuators it tends to have a more complex shape, and its behavior cannot be described by beam theory methods. Finite Element Analysis is increasingly used in the field of soft robots. In [22], a direct finite element simulation is used to observe the behavior of soft pneumatic actuators.

In this article, the development of a method to compute the kinematic model of soft manipulators that relies on the finite element model of the structure of the robot is presented. By using different types of elements (tetrahedral, beam, or shell elements), the methodology can be used on robots of very complex shapes. Contrary to the majority of models currently available in the literature, this approach also models two types of actuators, which enables this technique to be used as part of the control of the robot as well as in off-line analysis. Moreover, gravity and payload carried by the end-effector can be accounted for by this approach.

This article presents the following contribution towards the kinematic modeling of soft manipulators:

- A FEM-based modeling approach that accounts for complex structural shapes and the mechanics of the employed material.
- The model of two actuation systems (i.e. pulling on cables and pneumatics) currently implemented in the majority of designs of soft manipulators.
- The integration of sensors in the simulation that allows for an observer of the manipulator in the configuration space.
- The validation of the modeling approach using two completely different deformable manipulators.
- The experimental comparison of this approach to two other geometric-based models.
- An experimental study on the robustness of the model under external loading.
- A closed-loop controller based on a state estimator and the robustness analysis of the closed-loop system.

In section II, the formulation of the static equilibrium and the constraints for end-effector, actuators and sensors is explained. Section III shows the projection of the model in the constraint space and the convex optimization process used to solve the reduced model. Section IV presents the experimental validation of the forward and inverse kinematic models and finally, in Section V, a discussion about the results and limitations of the model, as well as the perspectives for future work are presented.

II. Methods

In this section, the development of the Finite Element Method (FEM) of soft manipulators is presented. The method relies on the constitutive law of the material from which the robot is made. This constitutive law can be directly measured by conducting stress/strain mechanical tests to a material sample in the ideal case. When the strain/stress tests cannot be done, an approximation of the constitutive law can be obtained in the simulation. The main idea is to tune these material parameters qualitatively by approximating the deformation seen in reality by that observed on simulation in which the deformation of the real robot is matched given a known static load. A similar approach is presented in [23] in the context of

radiotherapy. After measuring the constitutive law, a volume-based approach is used with tetrahedral elements. Then, the mathematical formulation of the constraints is introduced using Lagrange multipliers. In this method, the end-effector, actuators, and sensors use constraint models.

A. FEM model of soft and continuum robots

Depending on the shape of the robot, one could use volume, surface or linear elements to compute the non-linear deformation of the structure. In this paper, volumetric elements[2] are used. A non-linear formulation accounts for the large displacements and rotations of the structure. In continuum mechanics, this is considered as the case of large strain but small stress. More sophisticated FEM models can be proposed in the future, according to the constitutive law and the solicitation of the employed material (i.e. large stress). The computation in real-time with such models will be even more challenging, but the principles of the method described in this paper would still apply.

The corotational implementation of volume FEM, presented in [24], is particularly suitable for linear elasticity under the hypothesis of large displacements. The shape of the robot is meshed using linear tetrahedral elements, but the same method could be used with other elements, shape functions and more advanced material laws.

In the FEM, the nodes at the vertices of the elements represent the degrees of freedom of the manipulator. Even for a considerable amount of nodes, the approach is fast to compute, numerically stable and a free implementation in C++ is available in the open-source framework SOFA [25]. During each step i of the simulation, the following linearization of the internal forces is updated:

$$\mathbf{f}(\mathbf{x_i}) \approx \mathbf{f}(\mathbf{x_{i-1}}) + \mathbf{K}(\mathbf{x_{i-1}})d\mathbf{x} \tag{1}$$

where $\mathbf{f}$ provides the volumetric internal stiffness forces at a given position $\mathbf{x}$ of the nodes, $\mathbf{K}(\mathbf{x})$ is the tangent stiffness matrix that depends on the actual positions of the nodes and $d\mathbf{x}$ is the difference between positions $d\mathbf{x} = \mathbf{x_i} - \mathbf{x_{i-1}}$. This linearization is valid as long as the displacement of the nodes $d\mathbf{x}$ is small. The lines and columns that correspond to fixed nodes are removed from the system to get a full rank for matrix $\mathbf{K}$. In $\mathbf{f}$ and $\mathbf{K}$, the rows (and columns for $\mathbf{K}$) contain the component of the internal forces (x, y, z) for the nodes, in the order corresponding to their numbering in the mesh.

In this paper, the study is limited to quasi-static behavior on purpose, since the simulation step required to capture high frequency vibrations is not feasible. Thus, in a first approach, the assumption is that the control of the robot is performed at low velocities, so that the inertia effects in the motion of the robot can be neglected.

One seeks to establish static equilibrium at each step from the first law of Newton:

$$\mathbf{f_{ext}} + \mathbf{f}(\mathbf{x_i}) + \mathbf{H}^T\lambda = 0 \tag{2}$$

[2]The method has also been tested with beam elements.

where $\mathbf{f_{ext}}$ represents the external forces (e.g. gravity) and $\mathbf{H}^T\lambda$ gathers the contributions of the end-effector, actuators and the contact forces as Lagrange multipliers (see the following sections). The way $\mathbf{H}$ is obtained is explained in sections II-B and II-C but its computation is performed with the values obtained from the previous step. We then use the expression $\mathbf{H}(\mathbf{x_{i-1}})$ and through the linearization explained in (1), we obtain the following formulation :

$$-\mathbf{K}(\mathbf{x_{i-1}})d\mathbf{x} = \mathbf{f_{ext}} + \mathbf{f}(\mathbf{x_{i-1}}) + \mathbf{H}(\mathbf{x_{i-1}})^T\lambda \tag{3}$$

The variables $d\mathbf{x}$ and λ are both unknown and are found during the optimization process.

It is noted that the matrix $\mathbf{K}$ is highly sparse. In the implementation, a conjugate gradient solver is used and preconditioned by a sparse $\mathbf{LDL}^T$ decomposition. For a mesh composed of about 1000 nodes and about 3000 tetrahedral elements, a refresh rate of 60Hz is obtained with the implementation available in SOFA.

B. Constraint for the end-effector

To set the Lagrange multiplier on the end-effector, a point or a set of points of the robot needs to be considered as the end-effector. It could be any point(s) mapped on the finite element mesh. For each point, the constraint objective is to reduce the difference between the end-effector position and its desired position $\mathbf{p}_{des}$. Thus, a function $\delta_e(\mathbf{x}) : \mathbb{R}^{3n} \to \mathbb{R}^3$ with n being the number of nodes, evaluates this difference along x, y and z. If the end-effector corresponds to a node i of the mesh, the function is: $\delta_e(\mathbf{x}) = \mathbf{x}_i - \mathbf{p}_{des}$, where $\mathbf{x}_i$ is the position of node i. If the effector is set inside an element, we use:

$$\delta_e(\mathbf{x}) = \sum_{i=0}^{n} \phi_i(\mathbf{p}_{eff})\mathbf{x}_i \tag{4}$$

where $\mathbf{p}_{eff}$ is the position of the end-effector in the rest configuration of the FEM model and ϕ_i is the shape (interpolation) function associated to node i.

If several points are used for end-effector position, the vector $\delta_e(\mathbf{x})$ gathers the evaluation of the difference for all the points. The function is then $\mathbb{R}^{3n} \to \mathbb{R}^{3m}$, where m is the number of end-effector points.

The matrix $\mathbf{H}$ used for the end-effectors corresponds to $\mathbf{H}_e(\mathbf{x}) = \frac{\partial \delta_e(\mathbf{x})}{\partial \mathbf{x}}$.

The matrix $\mathbf{H}_e$ is highly sparse: A row, that corresponds to a component of a point of the end-effector, will contain non null values on a very small number of columns: As the point is mapped on a single tetrahedral element, there is a maximum of 4 non-null value per row. Of course, the column should match with the components of the nodes, given the fact that the non-null values are gathered in 3x3 diagonal block matrices.

Finally, an important point is the effort value that is put on the Lagrange multiplier that corresponds to the terminal effector. The value of λ_e will depend on the load applied on the end-effector. Two cases can be considered: (I) if the points defined as end-effector move freely in the space, there

is no physical interaction, so the contribution of the constraint vanishes $\lambda_e = \mathbf{0}$. (II) if one or several points of the end-effector carries one object l which mass creates a load that could deform the structure. In such cases, the corresponding load should be set on $\lambda_e = m_l \mathbf{g}$. with m_l the mass of the object and $\mathbf{g}$ the gravity field.

C. Actuator constraint model

In this work, the actuator model takes into account its physical characteristics. Two types of actuators have been implemented in the framework: Tendon (cable) and pneumatic actuators. The contributions of these actuator constraints are unknown before the optimization process. However, given the type of actuation, the constraint is not set the same way.

a) Cable actuator: In a first case (Fig 2), a cable is used to actuate the structure. The cable can simply be attached at one point of the structure, but it can also go through several other points (frictionless guides are considered) In all cases, the unknown λ_a is the stretching force inside the cable. This force is unilateral ($\lambda_a \geq 0$).

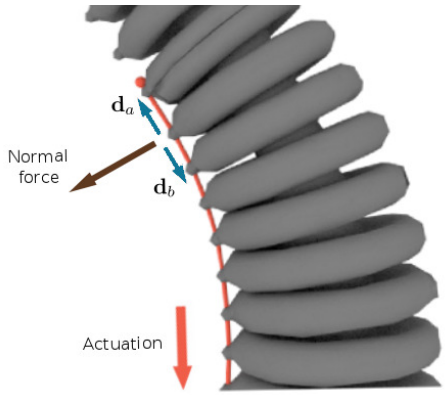

Fig. 2: Tendon actuation. $\mathbf{d}_b$ and $\mathbf{d}_a$ on the figure, represent the direction of the tendon before and after the cable guide, respectively, which are used to compute the normal forces at the guides.

Let's suppose now that the points are numbered starting from the extremity where the cable is being pulled. The matrix $\mathbf{H}$ is computed this way: At each point p, we take the direction of the cable *before* $\mathbf{d}_b = \frac{\mathbf{x}_p - \mathbf{x}_{p-1}}{\|\mathbf{x}_p - \mathbf{x}_{p-1}\|}$ and *after* $\mathbf{d}_a = \frac{\mathbf{x}_{p+1} - \mathbf{x}_p}{\|\mathbf{x}_{p+1} - \mathbf{x}_p\|}$. To obtain the constraint direction that is applied to the point, we use $\mathbf{d}_p = \mathbf{d}_a - \mathbf{d}_b$. Note that the direction of the final point is equal to the direction "before" as $\mathbf{d}_a$ does not exist. These constraint directions are mapped on the nodes using the interpolation:

$$\begin{bmatrix} \vdots \\ \mathbf{f}_n \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots \\ \phi_n(\alpha,\beta,\gamma)\ \mathbf{d}_p \\ \vdots \end{bmatrix} \lambda_a = \mathbf{H}_a^T \lambda_a \tag{5}$$

A function $\delta_a(\mathbf{x})$ is defined to provide the length of the cable, given the position of the constrained node(s). The actuator stroke can also be included by imposing $\delta_a(\mathbf{x}) \in [\delta_{min} \quad \delta_{max}]$). Through the use of this function, we get $\mathbf{H}_a = \frac{\partial \delta_a(\mathbf{x})}{\partial \mathbf{x}}$.

b) Pneumatic actuator: The formulation is compatible with pressure-based actuation of cavities that are placed on the structure, as seen in Fig. 3. In that case, the effort λ_a is the pressure exerted on the wall of the cavity. As the pressure is uniform inside the cavity, a single constraint can be set for each pneumatic actuator. All triangles of the cavity wall will be involved: For each triangle t, the area and the normal direction are computed. If this result is multiplied by the pressure, one obtains the force applied by the pneumatic actuator on the nodes t of this triangle. Consequently, the contribution of each triangle is added in the corresponding column of $\mathbf{H}_a^T$.

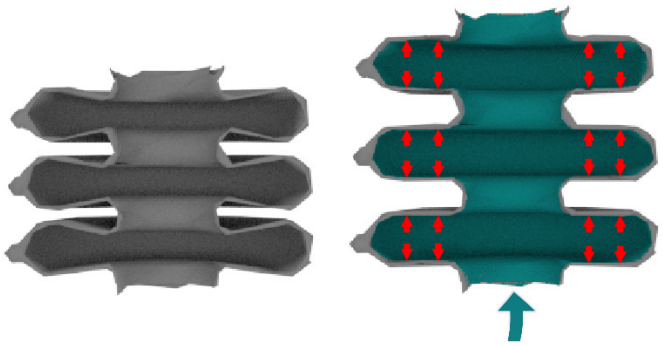

Fig. 3: Pressure actuation

In the particular case of a pneumatic actuation, λ_a provides the difference of pressure inside the cavity compared to the atmospheric pressure. Usually, pneumatic actuators only provide positive pressure so $\lambda_a \geq 0$. However, in some cases, it is also possible to create both negative and positive pressure using vacuum/pressure actuation. In that case, there is no particular constraint on the unknown value of λ_a, despite an eventual limit (max / min) of pressure that can be achieved by the actuator.

The approach to the modeling of fluidic actuators can also account for hydraulic actuators, by accounting for the weight distribution of the liquid at any given configuration, as presented in [26].

D. Sensor constraint model

In order to relate the end-effector position and the geometry of the manipulator, one needs sensors that can measure the geometric state of the robot. In this study, the sensors used can measure the lengths of the sections that compose the manipulator, but also that can be easily integrated in the design. String potentiometers offer a good solution to acquire information on the real geometric state of the robot. As in the case of the cable actuator, the string of the sensor is routed through several frictionless guides, at n points $\mathbf{x}_n$. In the model, the measure of the lengths read by the sensor will be

$$\sum_{i=1}^{n-1} \|\mathbf{x}_{i+1} - \mathbf{x}_i\| \tag{6}$$

which evaluates the distance between each sensor guide after the position of the nodes have been updated. A function

δ_s is defined to represent the difference between the current lengths of the sensors and the desired lengths. The matrix $\mathbf{H}_s$ that gathers the directions of the sensor constraint is obtained in the same way as for the cable actuator, shown in section II-C.

III. Reduced model in the constraint space

The classical resolution of a FEM problem (like solving the static equilibrium of the structure described at equation 3) provides a forward model: it allows to compute the displacements of the structure, given the values of the efforts put on the actuator λ_a. However, in the case of position control, the actuation λ_a is the unknown. Yet, for controlling the motion of the soft robot, an inverse model is needed, which is challenging to compute in real-time as the size of the system is in the range of several thousands degrees of freedom. In this work, another approach is used, based on the projection of the mechanics in the constraint space that drastically reduces the size of the optimization problem. This approach, initially developed in [27], is generalized. A new formulation of the inverse problem in the form of a quadratic programming (QP) optimization (developed in [28]) is used.

A. Reduced compliance in constraint space

As stated above, the optimization process relies on a projection of the mechanics in the constraint space. Each constraint has a direction that is set by a line of the matrices $\mathbf{H}_e$ and $\mathbf{H}_a$

This matrix is sparse, as the direction of the constraints is mapped on few nodes of the FE mesh. The values of the effort applied by the actuators λ_a are not known at the beginning of the optimization process, whereas the value of λ_e is supposed to be known.

The first step consists of obtaining a free configuration $\mathbf{x}_{\text{free}}$ of the robot which is found by solving the equation 3 while considering that there is no actuation applied to the deformable structure. In practice, the known value of λ_e is used and $\lambda_a = 0$ is imposed.

The linear equation 3 is solved using a LDL^T factorization of the matrix $\mathbf{K}$. Given this new *free* position $\mathbf{x}_{\text{free}}$ for all the nodes of the mesh, one can evaluate the values of $\delta_e^{\text{free}} = \delta_e(\mathbf{x}_{\text{free}})$, the shift between the effector point(s) position and the desired position introduced in section II-B. One can also evaluate $\delta_a^{\text{free}} = \delta_a(\mathbf{x}_{\text{free}})$ the position of the actuated points without actuation effort.

From the FEM formulation of the problem that uses a large matrix K, a formulation that accounts for the directions of the constraints placed for actuators and end-effectors is derived. Using the Schur complement of matrix K in the Lagrange multiplier-augmented system [29], a small formulation of δ_e is obtained. This variable expresses the difference between the desired position for the end-effector and its current position in terms of the actuators contributions λ_a:

$$\delta_e = \underbrace{\left[\mathbf{H}_e\mathbf{K}^{-1}\mathbf{H}_a^T\right]}_{\mathbf{W}_{ea}}\lambda_a + \delta_e^{\text{free}} \tag{7}$$

The Schur complement also provides similar formulations for the difference between a desired sensor or actuator position and its current position:

$$\delta_a = \underbrace{\left[\mathbf{H}_a\mathbf{K}^{-1}\mathbf{H}_a^T\right]}_{\mathbf{W}_{aa}}\lambda_a + \delta_a^{\text{free}} \tag{8}$$

$$\delta_s = \underbrace{\left[\mathbf{H}_s\mathbf{K}^{-1}\mathbf{H}_a^T\right]}_{\mathbf{W}_{sa}}\lambda_a + \delta_s^{\text{free}} \tag{9}$$

This step is central in the method. It consists in projecting the mechanics into the constraint space. As the constraints are the inputs (effector position shift and sensor length shift) and outputs (effort to apply on the actuators) of the inverse problem, the smallest possible projection space for the inverse problem is obtained. It allows for a projection that drastically reduces the size of the search space without loss of information. Indeed, section III-B shows how the matrices $\mathbf{W}_{ea}$ and $\mathbf{W}_{aa}$ provides the mechanical coupling equations between actuators and effector point(s).

After this projection, the optimization is processed in the reduced constraint space to get the values of λ_a. This part is described in the section III-C.

The final configuration of the soft robot, at the end of the time step, is obtained as :

$$\mathbf{x}_t = \mathbf{x}_{\text{free}} + \mathbf{K}^{-1}\mathbf{H}_a^T\lambda_a. \tag{10}$$

It should be emphasized that one of the main difficulties is to compute $\mathbf{W}_{ea}$ and $\mathbf{W}_{aa}$ in a fast manner. No pre-computation is possible as their value changes at each iteration. However, this type of projection problem is frequent when solving friction contact on deformable objects. Thus, several strategies are already implemented in SOFA [25].

B. Coupled Kinematic Equations

Using the compliance operator $\mathbf{W}_{ea}$, one can get a measure of the mechanical coupling between effector and actuator, and with $\mathbf{W}_{aa}$, the coupling between actuators.

For instance, the displacement δ_e^i created on the end-effector (along a direction stored on the line i of matrix $\mathbf{H}_e$) by a unitary force λ_a^j applied by the actuator (which is stored at the line j of matrix $\mathbf{H}_a$) is directly obtained by $\Delta\delta_e^i = w_{ea}^{ij}\lambda_a^j + \delta_e^{i,\text{free}}$.

As the motion is created by deformation, the motion of actuator j is influenced by actuator k.

Through the same principle, actuator k also influences the displacement of the effector. To get a kinematic link between actuators and effector, the method needs to account for the mechanical coupling that can exist between actuators. It is captured by $\mathbf{W}_{aa}$ that can be inverted if actuators are defined on independent degrees of freedom. Thus one can get a kinematic link by rewriting equation (8) as:

$$\delta_e = \mathbf{W}_{ea}\mathbf{W}_{aa}^{-1}(\delta_a - \delta_a^{\text{free}}) + \delta_e^{\text{free}} \tag{11}$$

Equation (11) is composed of a reduced number of linear equations that relate the displacement of the actuators to the displacement of the effector. Consequently, matrix $\mathbf{W}_{ea}\mathbf{W}_{aa}^{-1}$ is equivalent to the Jacobian matrix of a rigid manipulator. This matrix is a local linearization provided by the FEM model on a given position. It needs to be updated for deformations with large displacements.

C. Inverse model by convex optimization

The goal of the optimization is to find how to actuate the structure so that the end-effector of the robot reaches a desired position. This was initially proposed in [28]. It consists in reducing the norm of δ_e which actually measures the shift between the end-effector and its desired position. Thus, computing $min(\frac{1}{2}\delta_e^T\delta_e)$ leads to a Quadratic Programming (QP) problem:

$$min\left(\frac{1}{2}{\lambda_a}^T\mathbf{W}_{ea}^T\mathbf{W}_{ea}\lambda_a+{\lambda_a}^T\mathbf{W}_{ea}^T\delta_e^{\text{free}}\right) \tag{12}$$

$$\begin{array}{l} subject\ to\ (\text{course of actuators}):\\ \delta_{min}\leq\delta_a=\mathbf{W}_{aa}\lambda_a+\delta_a^{\text{free}}\leq\delta_{max}\\ and\ (\text{case of unilateral effort actuation}):\\ \lambda_a\geq 0 \end{array} \tag{13}$$

The use of a minimization allows to find a solution even when the desired position is out of the workspace of the robot. In such a case, the algorithm will find the point that minimizes the distance to the desired position while staying in the limits introduced by the course of the actuators.

In practice, the QP solver available in the Computational Geometry Algorithms Library (CGAL) [30] is used. The matrix of the QP $\mathbf{W}_{ea}^T\mathbf{W}_{ea}$ is symmetric. If the number of actuators is equal or inferior to the size of the end-effector space, the matrix is also definite. In such a case, the solution of the minimization is unique.

In the case when the number of actuators is greater than the degrees of freedom of the effector points, the matrix of the QP is only semi-definite. Consequently, the solution could be non-unique.

A new criterion for the minimization can be introduced, based on the deformation energy. Indeed, this energy E_{def} is linked to the mechanical work of the forces exerted by the actuators. E_{def} can be computed by evaluating the dot product between λ_a and the displacements of the actuators $\Delta\delta_a=\delta_a-\delta_a^{\text{free}}$ due to the actuator forces $E_{def}={\lambda_a}^T\Delta\delta_a={\lambda_a}^T\mathbf{W}_{aa}\lambda_a$. Yet, matrix $\mathbf{W}_{aa}$ is positive-definite if the actuators are placed on different nodes of the FEM or with different directions (i.e. if there is no linear dependencies between lines of $\mathbf{H}_a$. Thus, one can add this energy in the minimization process by replacing (12) with:

$$min\left(\frac{1}{2}{\lambda_a}^T(\mathbf{W}_{ea}^T\mathbf{W}_{ea}+\varepsilon\mathbf{W}_{aa})\lambda_a+{\lambda_a}^T\mathbf{W}_{ea}^T\delta_e^{\text{free}}\right) \tag{14}$$

with ε chosen sufficiently small so that the deformation energy does not disrupt the quality of the effector positioning. In practice, $\varepsilon=\frac{tr(\mathbf{W}_{ea}^T\mathbf{W}_{ea})}{tr(\mathbf{W}_{aa})}*10^{-3}$ so that the term $\varepsilon\mathbf{W}_{aa}$ does not alter the value of the QP matrix. Thanks to this modification, the QP matrix becomes positive-definite and a unique solution of the problem can be found.

IV. Kinematic models of a soft manipulator

In this section, the proposed methodology is validated experimentally by obtaining forward and inverse kinematic models of the Compact Bionic Handling Assistant (CBHA). The forward model is compared to two geometric models developed for the same robot. The simulation in real time of the inverse kinematic model is used in open loop to control the position of the end-effector of the manipulator. A proof of genericness is given by modeling a second soft manipulator inspired by parallel robots actuated by tendons with clear geometric differences and material characteristics.

A. Description of the CBHA

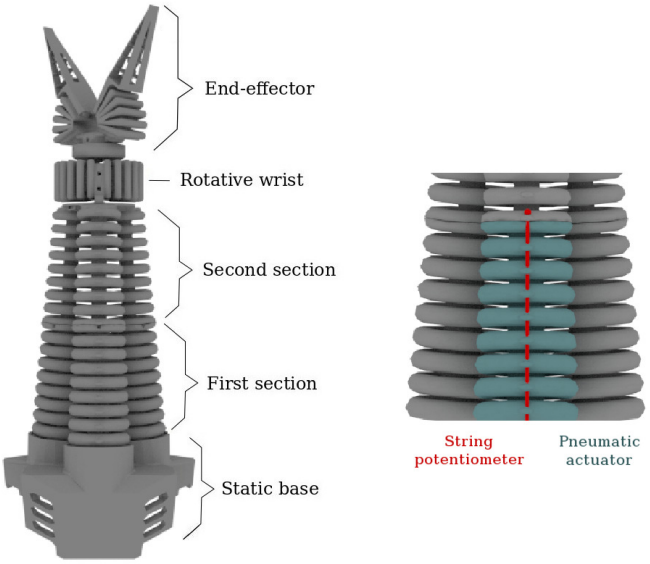

Fig. 4: The RobotinoXT by Festo Robotics. (left) The anatomy of the Compact Bionic Handling Assistant. (right) A section of the manipulator, composed by 3 pneumatic actuators and their correspondent length sensor.

The CBHA is the bionic continuum manipulator component of the RobotinoXT, a didactic mobile platform designed by Festo Robotics. The system is shown in Fig. 4 (a). The bionic continuum manipulator is formed by 2 serially connected sections of pneumatic actuators, an axially rotating wrist and a compliant gripper. Without actuation, the manipulator has a length of 206mm, with each section having a length of 103mm. The width at the base of the manipulator is 100mm long and the top has 80mm of width. In our study, the end of the second section is considered as the end-effector.

Each section of the manipulator is composed of a parallel array of pneumatic actuators, as shown in Fig. 4 (b). By

applying different pressures to the bellows, each section can bend or extend independently. The pose of the manipulator is obtained as the contribution of the poses of the 2 sections. In order to sense the state of the robot, string potentiometers measure the lengths of the actuators.

B. Forward kinematic models

The forward kinematic model of a soft manipulator deals with the problem of finding the end-effector position, given a defined configuration of the manipulator. For a rigid manipulator, this configuration is simply the set of variables associated with the joints of the robot. In contrast with the rigid robots, the variables that express the configuration of a soft manipulator change with respect to the structure of the robot and its type of actuation, and therefore, cannot be obtained in a straightforward manner. The FEM-based methods explained before provide an easy way to obtain the kinematic relation between the end-effector and the configuration of the manipulator.

- FEM-based model

Given the intrinsic nature of the CBHA, the configuration of the robot is represented by the lengths of the pneumatic actuators that correspond to an end-effector position. Of course, the description of the robot could be given in the actuator space directly, using in this case Equation 7, to attain a pressure-to-position model, but that requires a precise control over the actuation (in this case the pressure inside the cavities) in order to obtain a good estimation of the position of the end-effector. Instead, Equation 9, which is reproduced here for clarification, is used to relate the end-effector position to the configuration of the manipulator represented by the lengths of each pneumatic actuator, given by the sensors. This representation is clearer in the context of kinematic modeling, and allows for a position-to-position model which is less sensitive to unknown hardware parameters.

$$\delta_s = \underbrace{\left[\mathbf{H}_s\mathbf{K}^{-1}\mathbf{H}_a^T\right]}_{\mathbf{W}_{sa}}\lambda_a + \delta_s^{\text{free}} \tag{15}$$

In this approach, no geometrical assumptions are needed. Each part of the robot is modeled in detail using shell and tetrahedral elements, as presented in Fig. 5. The mesh used in the model of the pneumatic cavities is composed by 3528 elements.

Once the constraints have been incorporated in the model, the convex optimization finds each actuator contribution required to have the desired sensor lengths. The final position of the end-effector is obtained after the position of the nodes of the mesh is updated.

- Constant curvature model

This model of the CBHA, which was developed in [31] and [32] and validated in [33], works under the assumption that, after actuation, the resulting pose of each section in the robot can be represented by an arc section with constant curvature (Fig. 6).

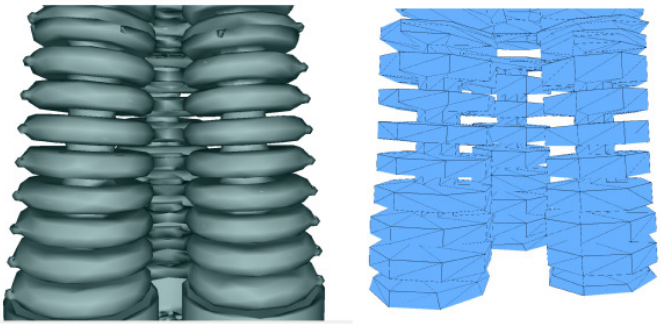

Fig. 5: Visual model of the trunk and the underlying finite element model.

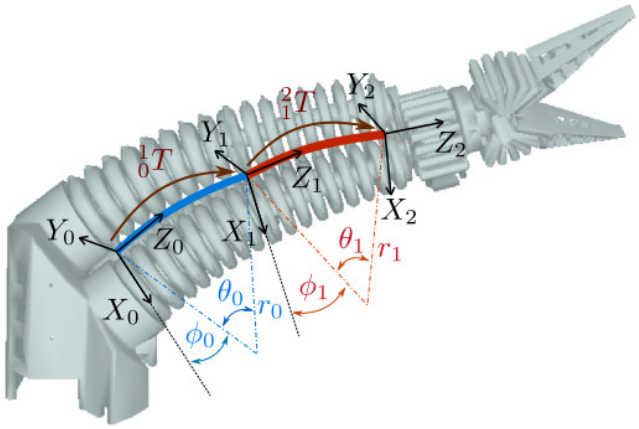

Fig. 6: Constant curvature model

The evolution from end-to-end of a section i is described, in terms of backbone parameters, by 2 coupled rotations and one translation in the homogeneous transformations:

$${}^{\text{i}}_{\text{j}}\text{T} = \begin{bmatrix} c^2\phi_i c\theta_i + s^2\phi_i & c\phi_i s\phi_i(c\theta_i - 1) & c\phi_i s\theta_i & x_i \\ c\phi_i s\phi_i(c\theta_i - 1) & s^2\phi_i c\theta_i + c^2\phi_i & s\phi_i s\theta_i & y_i \\ -c\phi_i s\theta_i & -s\phi_i s\theta_i & c\theta_i & z_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

where the notations s and c mean sine and cosine respectively. The cartesian coordinates of the end of the bending section i are given by (x_i, y_i, z_i), where $x_i = r_i c\phi_i(1 - c\theta_i)$, $y_i = r_i s\phi_i(1 - c\theta_i)$ and $z_i = r_i s\theta_i$. The backbone variables ϕ_i, θ_i and r_i can be expressed in terms of the actuator lengths in order to have the correct kinematic relations :

$$\begin{aligned} \phi_i &= tan^{-1}\left(\tfrac{\sqrt{3}(l_3 - l_1)}{2l_1 - l_2 - l_3}\right) \\ \theta_i &= \tfrac{D_i}{3d_i} \\ r_i &= \tfrac{(l_1 + l_2 + l_3)d_i}{D_i} \end{aligned} \tag{17}$$

with

$$D_i = 2\sqrt{l_1^2 + l_2^2 + l_3^2 - l_1 l_2 - l_1 l_3 - l_2 l_3} \tag{18}$$

The parameter d_i represents the diameter of section i. In this model, each section is considered to be a cylinder with constant radius. The lengths of each actuator in the section i are represented by l_1, l_2 and l_3.

- Hybrid model

In this approach, developed in detail in [34], the CBHA is considered as 17 vertebrae serially connected. Between each pair of vertebrae, an inter-vertebra section is modeled as a 3UPS-1UP joint (3 universal-prismatic-spherical joints and one universal-prismatic joint). The behavior of a sub-structure composed by 2 vertebrae and an inter-vertebra is represented by a parallel robot with 3 DoF, as shown in Fig. 7.

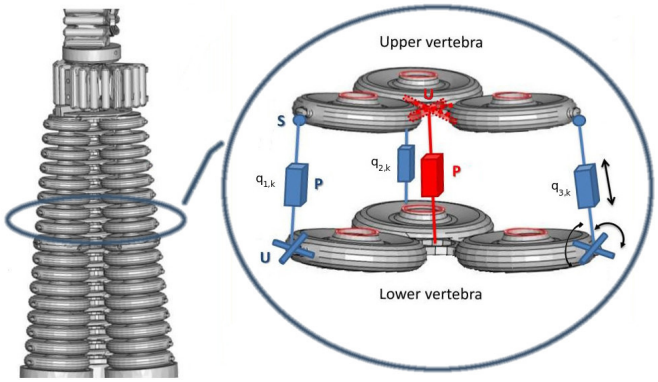

Fig. 7: Sub-structure of the CBHA modeled as a parallel robot.

The parallel robot consists of an upper and a lower platform connected by 3 limbs and a central leg. The limbs are modeled by a UPS joint in which only the prismatic part is active allowing the control of the position and orientation of the upper vertebra, with respect to the lower vertebra. The central leg is modeled as a passive UP joint and is used to constraint the rotation about the longitudinal axis of the parallel robot, as well as any shearing motion between the vertebrae.

The position and orientation of the upper vertebra $k+1$, with respect to the lower vertebra k is given by the transformation matrix

$$ {}^{k}_{k+1}\mathrm{T} = \begin{bmatrix} c\theta_k & s\theta_k s\psi_k & s\theta_k c\psi_k & 0 \\ 0 & c\psi_k & -s\psi_k & 0 \\ -s\theta_k & s\psi_k c\theta_k & c\theta_k c\psi_k & z_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{19} $$

where the angles θ_k and ψ_k represent pitch and roll angles, respectively, and the notations s and c mean sine and cosine respectively. In this model, the prismatic variable $q_{n,k}$ shown in Fig. 7 represents the length of each inter-vertebra, which is a percentage of the total length of the actuator. This percentage can be obtained by considering the minimum and maximum elongation of each inter-vertebra. This development is presented in detail in [34].

C. Experimental validation and model comparison

In order to validate the model, a set of 50 end-effector positions distributed inside the task space of the manipulator were selected. For each position, the correspondent configuration of the robot was recorded using the string potentiometers that are placed along the structure of the robot. The set of lengths recorded were used as an input for the forward kinematic model. The experiment assumes zero-end-effector payload. The results are compared to those of the Constant Curvature and also the Hybrid approach. This comparison is presented in Fig. 8 and 9.

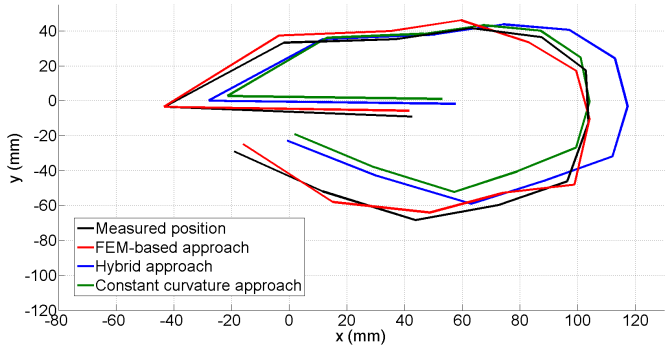

Fig. 8: X/Y view of the results from the model comparison.

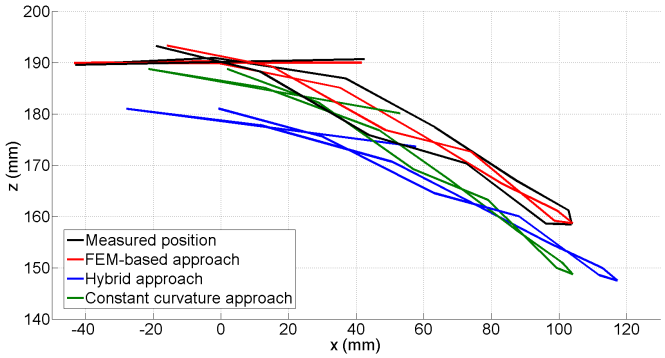

Fig. 9: X/Z view of the results from the model comparison.

The results show that the constraint approach is more accurate in estimating the position of the end-effector, with a Root-Mean-Square (RMS) error of 4.66mm, compared to 12.87mm and 17.09mm of error for the Constant curvature and the Hybrid approaches, respectively. We hypothesize that the imprecise measurement of displacement for each vertebra may be the cause of the hybrid approach being less precise than ours, as there were only 6 string potentiometers available to chart the displacements. Moreover, this model was initially developed to be able to inverse it, more than for the pure precision of the forward kinematic model.

Nevertheless, the FEM model still has a few limitations in its development. These limitations represent the main source of error in the results: for the moment, the constitutive law used to model the material of the trunk is only an approximation.

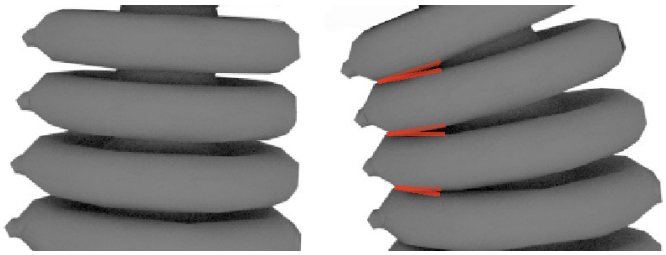

Fig. 10: Collision of the outer wall of the cavities. The collisions occur on the orange edges depicted in the bent actuator (right).

Moreover, non-linear effects like the viscosity of the material are not yet implemented in the model.

Another source of error comes from the geometry of the trunk itself. When the trunk is bent at a maximum angle, the outer walls of the pneumatic cavities collide with each other, as shown in Fig. 10. The consideration of these collisions is not yet implemented in the simulation.

The generic nature of the approach showcased in this article is illustrated by obtaining the inverse kinematics of two different soft manipulators. The simulation of the inverse model provides the position control of the robots in open loop that can be used to pilot directly the robot, as in the case of the parallel soft robot.

D. Inverse kinematic model

In this section, an experimental validation of the modeling methodology is conducted using two different soft robots:

- A parallel soft robot made of silicone, actuated with tendons (cables) controlled in position,
- The Compact Bionic Handling Assistant (CBHA).

The inverse model provided by convex optimization in real-time allows to teleoperate the robots in open-loop: Given a desired input position of the effector, the desired output for the actuators is computed. For the soft parallel robot, a desired position of the tendons is provided.

1) Modeling and feed-forward control of a parallel soft robot: This experiment is based on a 3D soft robot, made of silicone, which design is inspired by parallel robots with closed kinematic chains (Fig. 11). In its rest position, the dimensions of the robot are $180 \times 180 \times 130mm$ The robot naturally deforms and sinks under the action of gravity, but 4 unilateral actuators (servo-motors that are connected to the structure of the robot with cables) are placed on each leg to prevent and pilot the deformation. The effector position is placed on the upper part of the robot. Its trajectory is defined in 3D (and can interactively be changed by a user) and the algorithm provides the position to apply to the servomotors. The Young modulus of the silicone, measured experimentally, is used to parametrize the robot. The FEM model of the robot is composed of 4147 Tetrahedrons and 1628 Nodes. When projected in the constraint space, the size of the system is highly reduced: 3 equations for the effector, and 4 equations for the actuators. The convex optimization that leads to the

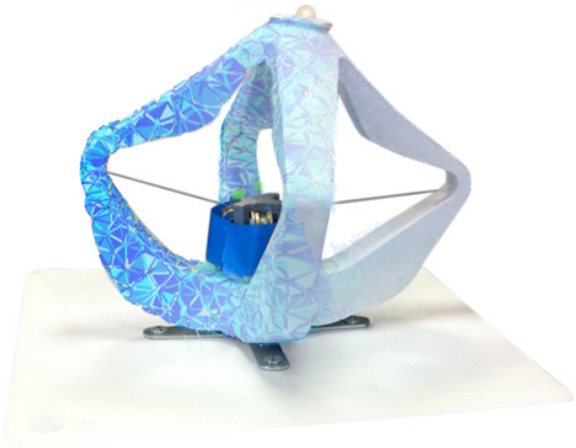

Fig. 11: Deformable parallel manipulator.

inverse model can be performed in real-time. The most time-expensive part of the computation is the projection expressed in equations (7) to (8) (50 ms on a Core i7, 2.8GHz), but when using the Graphics Processing Unit (GPU) method described in [35], it significantly reduces the computation time of the projection (15ms in this case).

To validate the method, a study of the discrepancy between the desired positions and the obtained positions is conducted on static positions distributed across a workspace of $25mm \times 25mm \times 50mm$ around the rest-position of the robot (see Fig. 12). The measurements are performed using a motion capture system based on infrared cameras[3]. On a sample of 28 positions, a mean error of $1.4mm$ is obtained with a standard deviation of 0.6 mm and a maximum error of 2.9 mm. This illustrates the precision that can be achieved using such FEM approaches.

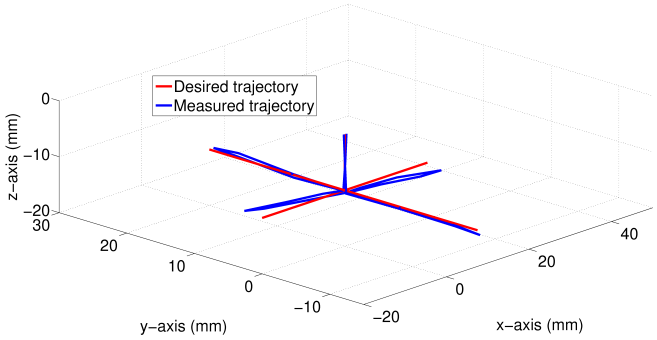

Fig. 12: Comparison of desired trajectory and measured trajectory of the parallel manipulator

It can be noted that these results are obtained using an open-loop and with a *position to position* control: for a given position of the effector, the algorithm finds a position for the

[3]The positioning precision provided by the motion capture system is less than $0.1mm$

actuator cables. This is a favorable case for FEM precision because the partial differential equations are enforced with Dirichlet boundary conditions.

2) Inverse kinematics of the CBHA: Considering the kinematic relationship for the CBHA given in section IV-D, that is the link between the actuator lengths and the position of the end-effector, the inverse kinematic model, solved by the convex optimization will give the actuator lengths that result from a predefined end-effector position. The FEM analysis applied to model this soft robot is detailed in [36]. A domain decomposition strategy is applied in order to perform the computation of the model (Equation (3)) and the projection in the constraint space (Equation (7)). After the actuator contribution required to achieve the desired position of the end-effector is applied to the model, and the position of the nodes is updated, the readings from the sensors in the simulation, given by Equation 6, will give the lengths of the pneumatic actuators that represent the output of the inverse kinematic model.

To validate the method, a set of 50 end-effector positions are selected inside the task space of the robot and the corresponding set of lengths for each position is recorded by the sensors of the robot. The same set of positions is used as inputs for the inverse model, and the resultant length of each actuator is estimated. This study is summarized in Table 1, where $l_1,\ldots,l_6$ represent the lengths of the actuators and their values are in mm, μ represents the mean error and σ is the standard deviation. The results are presented in Fig. 13.

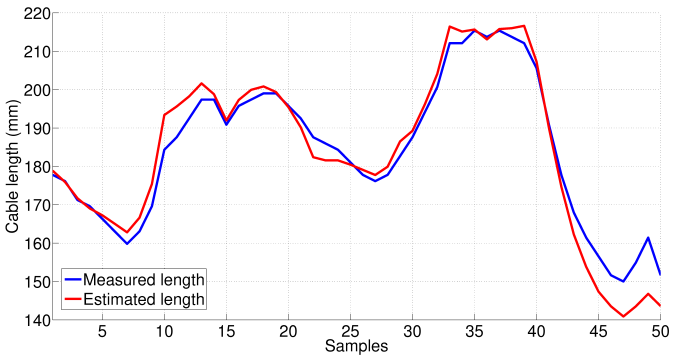

Fig. 13: Comparison of measured and estimated lengths of one of the sensors given a predefined set of end-effector positions for the CBHA

TABLE 1: Statistical analysis of the error between measured and estimated lengths for the CBHA

l(mm)	l_1	l_2	l_3	l_4	l_5	l_6
$\mu(mm)$	3.2	2.43	3.86	4.08	3.6	3.69
$\sigma(mm)$	1.55	1.76	2.05	2.12	2.56	2.06

The results show a mean error between 2.43mm and 4.08mm across all lengths, which represents between 1.21% and 2.04% of the total length of the manipulator. As in the case of the parallel robot, the set of actuator contributions (in this case the pressures applied to the cavities) obtained from the optimization process can be used as input for the real robot to obtain a feed-forward control. However, as explained before, there are some considerable discrepancies between the pressure calculated by the simulation and the pressure applied to the cavities, mainly caused by the way the pressure is regulated in the robot. This leads to less accurate results.

In order to improve the results in terms of efforts (pressure,force) in the inverse model, one can use more advanced constitutive models for the materials. One of these models is the St Venant-Kirchhoff hyper-elastic model. The stress/strain relationship in the St Venant-Kirchhoff model is represented by the Second Piola-Kirchhoff stress tensor $\mathbf{S}$ that has the form:

$$\mathbf{S} = \lambda \mathbf{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E} \tag{20}$$

where $\mathbf{E}$ is the Lagrange-Green strain tensor and λ and μ are the Lamé constants that can be approximated from the Young's modulus and Poisson's ratio of the material in question. We have conducted tests on the parallel soft robot using the St. Venant-Kirchhoff model to compare the results to those obtained using the corotational formulation. In the tests we observed very small errors in the displacement output (3.24% of a total cable stroke of 50mm). In the case of the force output we observed bigger errors (16.02% of the tension in the cable) related to the errors made by the corotational formulation in the stress computation.

E. Deflection of end-effector under external loading

As explained in section II, one of the advantages of this modeling approach is the ability to predict the deflection of the robot under external loading, given a good representation of the material mechanics. If the load is known *a priori*, the value of the force acting on the end-effector λ_e can be used in equation (3) to compute the position that accounts for said force. In order to validate this modeling feature, a set of experiments were conducted on both manipulators using known loads.

First, an initial configuration for the manipulator without loading is selected and the position of the end-effector is measured, then the load is applied and the new position of the end-effector is recovered after the robot achieves static equilibrium. The same load is then applied to the model of the manipulator using the same initial pose and the resulting end-effector position is also recovered. In the case of the CBHA, the model of the sensors presented in section II-D is used to apply the configuration of the real robot measured by the string potentiometers to the simulation model. A vector that connects initial and final end-effector positions represents the deflection caused by the loading.

In order to assess the repeatability of the measurements, the loading sequence described before is performed 40 times for each loading value and the average value is then used for the model validation. A standard deviation of 0.4838mm is obtained across all the measurements. The comparison between measured and model deflections for both manipulators is presented in Fig. 14 and 15.

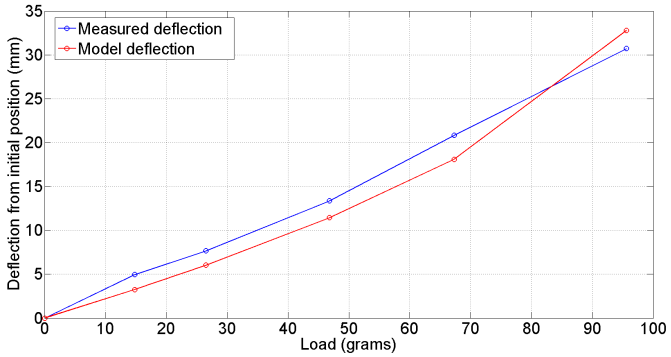

Fig. 14: Comparison between measured and predicted deflections caused by external loading on the parallel manipulator

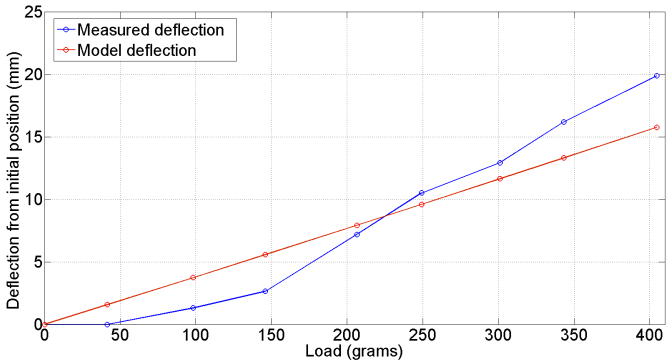

Fig. 15: Comparison between measured and predicted deflections caused by external loading on the CBHA manipulator

In the figures, the blue line represents the compliance to loading of the manipulators and the red line is the prediction of the model. In the case of the CBHA, the maximum error is 4.107mm with an average error of 2.1047mm. Nevertheless, Fig. 15 shows that the CBHA presents a strain hardening/necking stages of plastic behavior at the beginning of the loading profile which corresponds to the compliance of the plastic material from which the manipulator is made of (polyamide nylon), and therefore the model prediction is accurate only for a small region of the profile. In order to improve the model predictive capabilities for the CBHA in particular, two constitutive laws could be implemented to account for the different behaviors, but this would modify the way the inverse FEM is formulated. In contrast, the maximum error in the case of the parallel manipulator is 2.06mm with an average error of 2.01mm. The reason we obtain better results is because the material of the parallel robot conforms better to the assumption of high deformation and low stress, while also being an elastic material with no plastic behavior.

V. FEM-BASED CLOSED-LOOP CONTROL OF CONTINUUM ROBOTS

In section IV-D2, the relationship between the sensor lengths and the end-effector position of the CBHA was obtained based on the FEM simulation of the robot, however, in order to control the motion of the robot, the set of pressures applied to the actuators is to be computed. Indeed, the relationship given by equation (7) can be used to control directly the robot in open-loop, but as explained in IV-B this requires an accurate control over the pressures applied to the robot. Moreover, non-linear behaviors like the hysteresis and strain-rate dependency of the material (which is not considered in the model) render the feedforward control of the manipulators unusable in real applications.

Controllers for soft manipulators have been investigated in the past with the intention of rejecting non-linear behaviours and model uncertainties that result from the complex dynamics of the manipulators. Control based on energy formulations [37], model-less approaches [38] and feedback controllers [39] [40] have been proposed before with the intention of achieving accurate positioning of the manipulators in presence of non-modeled dynamics. In this section, a closed-loop control strategy based on a state estimator is proposed.

A. Closed-loop control design

The closed-loop controller is designed to ensure the correct configuration of the robot, given a desired end-effector position. A reference computation is performed to transform the desired position to the correspondent configuration. Assuming that the external forces are constant, the discrete model of the system, derived form of equation (9), takes the form:

$$\delta_{s,k+1} = \delta_{s,k} + \mathbf{J}_{sa}(\mathbf{x}_k)\Delta\lambda_{a,k+1} \tag{21}$$

where $\mathbf{J}_{sa} = \mathbf{W}_{sa}$ is the Jacobian matrix between sensors and actuators. When the desired sensor lengths are provided by the reference computation, we can propose the closed-loop control approach shown in Fig. 16

In Fig. 16 the blue blocks represents the computations performed by simulation. Two simulations executing simultaneously are implemented in the closed-loop system: One main simulation that computes the Inverse kinematic model and a second simulation that acts as a state estimator for the system. The state estimator is the Forward kinematic model simulation of the robot that computes an estimated configuration for the robot based on the lengths of the sensors. This configuration is used to update the state of the Inverse kinematic model at each simulation step. In this way, we make sure that the configurations of both simulation model and the manipulator are similar before the estimation of the Jacobian is computed. The tracking error e_k in the closed-loop system is computed as:

$$e_k = \delta_{s,k} - \delta^d_{s,k} \tag{22}$$

with $\delta^d_{s,k}$ represents the desired lengths of the sensors and $\delta_{s,k}$ represents the current lengths in the robot. We define the control vector $\mathbf{v}_k$ as:

$$\mathbf{v}_k = \hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)\mathbf{r}_k \tag{23}$$

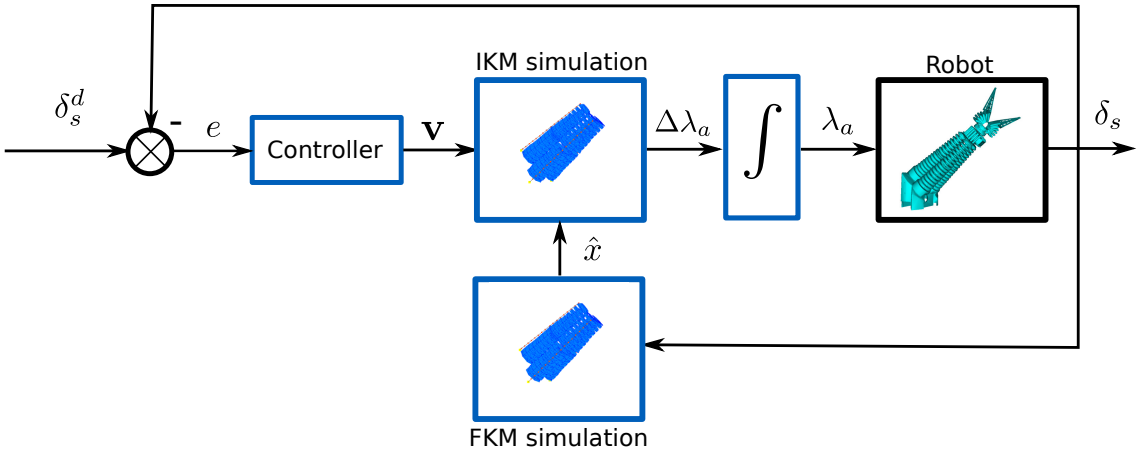

Fig. 16: Closed-loop control of the CBHA based on IKM and FKM simultaneous simulations and the controller

where $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$ is the estimated Jacobian matrix between the sensors and actuators and $\mathbf{r}_k = \Delta\lambda_{a,k+1}$. Using Eq. 23, the kinematic model can be rewritten as:

$$\delta_{s,k+1} = \delta_{s,k} + \mathbf{v}_k \tag{24}$$

The control law is based on proportional integrative strategy, therefore, the control vector $\mathbf{v}_k$ is designed in the sensor space as:

$$\mathbf{v}_k = -k_p e_k - k_i h_k \tag{25}$$

with k_p and k_i being the proportional and integrative gains of the controller, respectively. The integrative term h at time $k+1$ is computed as:

$$h_{k+1} = h_k + e_k \tag{26}$$

Then, the control allocation based on a Quadratic Programming (QP) formulation [41] is employed to find a unique solution to:

$$\mathbf{r}_k = \hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)\mathbf{v}_k \tag{27}$$

where $\hat{\mathbf{J}}^{+}_{sa}$ is the pseudo-inverse of the estimated Jacobian. In practice, as $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$ may not be fully invertible, we introduce a variable O defined as

$$O = \hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)\mathbf{r}_k - \mathbf{v}_k \tag{28}$$

Using O, the QP problem formulation (III-C) becomes:

$$\min_{\mathbf{u}_k} \quad (O^T O) \tag{29}$$

the resulting $\mathbf{r}_k$ will be the best possible inversion of Eq. 23 in the least square sense. In addition, the QP formulation allows to define constraints like actuator saturation or positive direction of actuation. Using Eq. 25 in Eq. 27, $\mathbf{r}_k$ is rewritten as

$$\mathbf{r}_k = -\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)(k_p e_k + k_i h_k) \tag{30}$$

Using Eq. 30 in Eq. 21, the closed-loop system is defined as:

$$e_{k+1} = e_k + \mathbf{J}_{sa}(\mathbf{x}_k)\mathbf{r}_k \tag{31}$$

which in the ideal case in which $\hat{\mathbf{J}}_{sa}$ is invertible, can be written as:

$$e_{k+1} = e_k + \mathbf{v}_k \tag{32}$$

The system of Eq. 32 is a simple first order discrete model that can be controlled with any standard controller. We choose the control strategy to be based on proportional-integrative control law because we want to improve the convergence rate and remove any steady state error (in the sensor space at least). After testing, the selected gain values are $k_p = 0.14$ and $k_i = 0.0003$ as a compromise between the rise time of the signal and its overshot. Fig 17 shows the response of one actuator length of the simulated robot and the real robot given a pre-computed set points corresponding to an end-effector position inside the task space of the robot. The position is chosen so that the actuators are far from their saturation points. The model simulation and the real robot have different initial condition. The experiment was performed for 2500 simulation steps with a simulation step of 0.1 seconds. After 1000 simulation steps, the set points are changed in both the simulation and the real robot.

The results show that both, the simulation of the robot and the robot itself have the same settling time $t_s \approx 400$ simulation steps. We can also see that the curve that represents the measured value of the lengths in the robot *jumps* between two values. This behavior is a consequence of the poor resolution of the string potentiometers. Fig 17 also shows a different behavior in the transitory stage of the curve of the measured lengths. This behavior can be attributed to different factors; first, there is the time required to compute the configuration of the manipulator from the measured sensor lengths; second, there is a time delay for the desired pressure to be applied to the robot, and finally, the plasticity of the material from

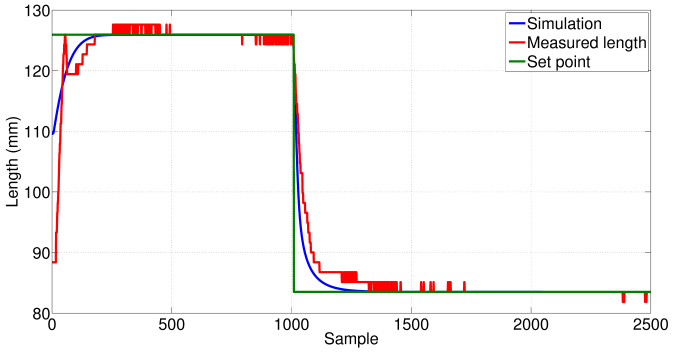

Fig. 17: Comparison of real and estimated actuator length of the CBHA. A second set point is applied to the system after 1000 simulation steps in order to observe the performance of the controller. The time step is 0.1s for the experiment.

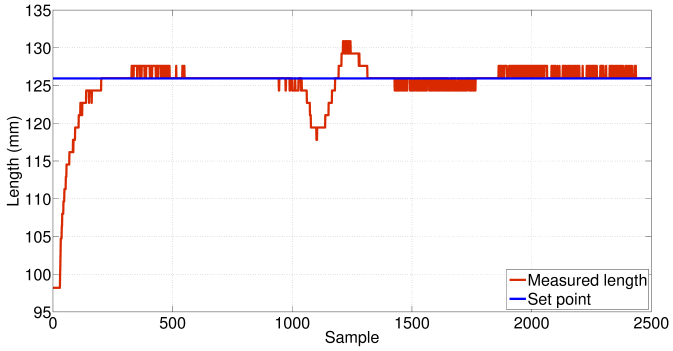

Fig. 18: Measured lengths of the CBHA in closed-loop. An external force is applied to the manipulator after 1050 time steps. The time step is 0.1s for the experiment.

which the manipulator is built (polyamide-nylon) which is not accounted for in our FEM model. On the other hand, the pneumatic valves that control the pressure inside the actuators have a small dead zone, so, when the manipulator starts its motion from a zero-pressure condition, very small increments in the pressure do not produce any motion until this dead zone is surpassed, which is not considered in the FEM.

A second experiment was performed using the real robot in the loop. In this experiment an external unknown force was applied to the manipulator in order to see the uncertainty rejection capabilities of the controller. Fig. 18 shows the results of this experiment.

B. Robustness analysis

Because of modeling uncertainties, the estimated Jacobian matrix $\hat{\mathbf{J}}_{sa}(\hat{\mathbf{x}}_k)$ is, in general, different from the Jacobian of the robot $\mathbf{J}_{sa}(\mathbf{x}_k)$. We introduce the vector ω_k that represents the disparities between the real Jacobian and the estimated Jacobian. We call this vector the *inversion error* and is defined as:

$$\omega_k = [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)]\mathbf{v}_k \tag{33}$$

Then, the closed-loop system is re-written as:

$$e_{k+1} = e_k + \mathbf{v}_k + \omega_k = e_k - k_p e_k - k_i h_k + \omega_k \tag{34}$$

The disturbed closed-loop system is:

$$\begin{bmatrix} e_{k+1} \\ h_{k+1} \end{bmatrix} = \begin{bmatrix} \mathbf{I} - k_p\mathbf{I} & -k_i\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} \begin{bmatrix} e_k \\ h_k \end{bmatrix} + \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \omega_k \tag{35}$$

It can be disturbing that we end up with such a simple linear system. We emphasize to the reader that the non-linearities are taken into account by the two simulation blocks (FKM and IKM in Fig. 16) in the closed-loop control. In Eq. 35, we are writing the system in terms of e_k and h_k and if the model was perfect, the system would be trivial. However, we can have modeling errors, that is why, in the following, we will prove that the control is robust to these modeling uncertainties ω_k.

To simplify the notation of the problem, we define the following vectors:

$$\mathbf{X}_{k+1} = \begin{bmatrix} e_{k+1} \\ h_{k+1} \end{bmatrix}, \mathbf{X}_k = \begin{bmatrix} e_k \\ h_k \end{bmatrix}, \mathbf{D} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix}, \mathbf{F} = \begin{bmatrix} k_p & k_i \end{bmatrix} \tag{36}$$

Also

$$\begin{bmatrix} \mathbf{I} - k_p\mathbf{I} & -k_i\mathbf{I} \\ \mathbf{I} & \mathbf{I} \end{bmatrix} = \mathbf{A} - \mathbf{BF} \tag{37}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{I} & \mathbf{I} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \mathbf{I} \\ 0 \end{bmatrix} \tag{38}$$

Using this notation, matrix $\omega_\mathbf{k}$ is written as:

$$\omega_k = [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)]\mathbf{F}\mathbf{X}_k \tag{39}$$

We assume that the error in the Jacobian estimation is bounded by a bounding parameter γ such that:

$$\omega_k^T \omega_k = \mathbf{X}_k^T \mathbf{F}^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)]^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)]\mathbf{F}\mathbf{X}_k \leq \gamma^2 \mathbf{X}_k^T \mathbf{F}^T \mathbf{F}\mathbf{X}_k \tag{40}$$

with

$$[\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)]^T [\mathbf{I} - \mathbf{J}_{sa}(\mathbf{x_k})\hat{\mathbf{J}}^{+}_{sa}(\hat{\mathbf{x}}_k)] \leq \gamma^2 \mathbf{I} \tag{41}$$

For the proof of stability, we use Lyapunov's second method of stability [42]. We define the Lyapunov candidate function as:

$$V = \mathbf{X}_k^T \mathbf{P}\mathbf{X}_k \tag{42}$$

where $\mathbf{P}$ is an unknown Lyapunov matrix with the properties

$$\mathbf{P}^T = \mathbf{P} > 0 \tag{43}$$

From Eq. 42 and the notation given in Eq. 36, the variation of the Lyapunov function is defined as:

$$\Delta V = \mathbf{X}_{k+1}^T \mathbf{P}\mathbf{X}_{k+1} - \mathbf{X}_k^T \mathbf{P}\mathbf{X}_k \tag{44}$$

Using Eq. 38, Eq. 44 is re-defined as:

$$\Delta V = ((\mathbf{A}-\mathbf{BF})\mathbf{X}_k + \mathbf{D}\omega_k)^T \mathbf{P}((\mathbf{A}-\mathbf{BF})\mathbf{X}_k + \mathbf{D}\omega_k) - \mathbf{X}_k^T \mathbf{P}\mathbf{X}_k \tag{45}$$

By making

$$\mathbf{A}-\mathbf{BF} = \mathbf{C} \tag{46}$$

Eq. 45 is written as:

$$\begin{aligned}\Delta V &= (\mathbf{C}\mathbf{X}_k + \mathbf{D}\omega_k)^T \mathbf{P}(\mathbf{C}\mathbf{X}_k + \mathbf{D}\omega_k) - \mathbf{X}_k^T \mathbf{P}\mathbf{X}_k \\ &= \mathbf{X}_k^T \mathbf{C}^T \mathbf{P}\mathbf{C}\mathbf{X}_k + \mathbf{X}_k^T \mathbf{C}^T \mathbf{P}\mathbf{D}\omega_k + \omega_k^T \mathbf{D}^T \mathbf{P}\mathbf{C}\mathbf{X}_k + \omega_k^T \mathbf{D}^T \mathbf{P}\mathbf{D}\omega_k - \mathbf{X}_k^T \mathbf{P}\mathbf{X}_k\end{aligned} \tag{47}$$

Reverting the notation in Eq. 38, Eq. 47 can be written in matrix form as:

$$\Delta V = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \begin{bmatrix} (\mathbf{A}-\mathbf{BF})^T \mathbf{P}(\mathbf{A}-\mathbf{BF}) - \mathbf{P} & (\mathbf{A}-\mathbf{BF})^T \mathbf{P}\mathbf{D} \\ \mathbf{D}^T \mathbf{P}(\mathbf{A}-\mathbf{BF}) & \mathbf{D}^T \mathbf{P}\mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} \tag{48}$$

For the proof, we introduce an accessory parameter $\alpha \geq 0$ in Eq. 40, such that:

$$\Upsilon = \alpha\omega^T\omega - \alpha\gamma^2 \mathbf{X}_k^T \mathbf{F}^T \mathbf{F}\mathbf{X}_k < 0 \tag{49}$$

From Eq. 49, the left hand side of the inequality is written in matrix form as:

$$\Upsilon = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \begin{bmatrix} -\alpha\gamma^2 \mathbf{F}^T\mathbf{F} & 0 \\ 0 & \alpha\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} < 0 \tag{50}$$

Adding and subtracting this term to Eq. 48 allow us to find a bounding for ΔV as:

$$\Delta V - \Upsilon + \Upsilon = \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix}^T \mathbf{Q} \begin{bmatrix} \mathbf{X}_k \\ \omega_k \end{bmatrix} + \Upsilon \tag{51}$$

with

$$\mathbf{Q} = \begin{bmatrix} (\mathbf{A}-\mathbf{BF})^T \mathbf{P}(\mathbf{A}-\mathbf{BF}) - \mathbf{P} + \alpha\gamma^2 \mathbf{F}^T\mathbf{F} & (\mathbf{A}-\mathbf{BF})^T \mathbf{P}\mathbf{D} \\ \mathbf{D}^T \mathbf{P}(\mathbf{A}-\mathbf{BF}) & \mathbf{D}\mathbf{P}\mathbf{D}^T - \alpha\mathbf{I} \end{bmatrix} \tag{52}$$

We know from Eq. 49 that $\Upsilon < 0$. Therefore, if $\mathbf{Q}$ is definite negative, then $\Delta V < 0$. To prove the closed-loop system to be stable, the values for matrix $P > 0$ and $\alpha \geq 0$ need to be found such as matrix $\mathbf{Q}$ is definite negative, given the predefined values of the boundary parameter γ and the tuned controller parameter k_p and k_i. To this end, a Linear Matrix Inequality [43] Solver called SeDuMi [44] is used in the software Matlab. In order to describe the LMI given by Eq. 46, Yalmip [45], a toolbox for optimization that is compatible with Matlab is employed. Given a value of $\gamma = 0.98$ and the gain values $k_p = 0.14$ and $k_i = 0.0003$, the LMI was solved successfully. The resulting matrix P and the parameter α that make matrix Q negative definite are:

$$\mathbf{P} = \begin{bmatrix} 646.4512 & 1.2983 \\ 1.2983 & 0.0087 \end{bmatrix} \quad and \quad \alpha = 4655 \tag{53}$$

Using the LMI solver, we can also compute the maximum value of γ, which provides an insight on the robustness of the closed-loop system. After some iterations we have:

$$max \quad \gamma = 0.98685 < 1 \tag{54}$$

Recalling Eq. 39, if $\omega^T\omega > 1$, then matrices $\mathbf{J}_{sa}(\mathbf{x_k})$ and $\hat{\mathbf{J}}_{sa}^{+}(\hat{\mathbf{x}}_k)$ do not have the same sign, which means that the robot Jacobian and the estimated Jacobian indicate opposite directions. In our case, $\gamma = 0.98685$ is close to the limit case. The proposed closed-loop system is robustly stable and can handle high Jacobian inversion errors in the change of control variables.

VI. Conclusions and future work

This paper presents a modeling methodology to obtain the kinematic relationships of soft manipulators. The kinematic equations are derived from a FEM model (or any equivalent physics based model) that can be obtained from the geometry and the material properties of a soft manipulator. After a projection in a small constraint space, a set of coupled equations relate the position of the end-effector to the contribution of actuators and displacement of sensors. The validity of the method is demonstrated in two different manipulators with complex geometry. In the case of the CBHA, the results were compared to those obtained with two geometric models developed for the same robot. While the model of the material used does not take into account the properties of viscosity, this consideration is only due to the absence of knowledge of these specific properties for the material used. Indeed, the framework used allows for modeling viscoelasticity with Prony series [46]. In general, a viscoelastic model is characterized by a rate-independent term, which in this case is the shear modulus representing the elastic behavior, and a rate-dependent modulus. The rate-dependent modulus of the material is defined by the Prony series based on time; faster strain rates will induce higher modulus than static loads. The limitations on the use of Prony series come with the determination of the required coefficients, since it involves stress relaxation tests performed under controlled temperature and loading speed. Another way to model viscoelasticity behaviour is to introduce a rate-dependent damping effect using Rayleigh equation. Rayleigh damping is a viscous damping that is proportional to a linear combination of mass and stiffness. Using Rayleigh damping, The internal forces in the robot (equation 1) takes the form:

$$\mathbf{f}(\mathbf{x_i}) \approx \mathbf{f}(\mathbf{x_{i-1}}) + \mathbf{K}(\mathbf{x_{i-1}})d\mathbf{x} + \mathbf{B}(\mathbf{x_{i-1}})d\mathbf{x} \tag{55}$$

where the Rayleigh damping matrix is computed as:

$$\mathbf{B} = \alpha\mathbf{M} + \beta\mathbf{K} \tag{56}$$

where $\mathbf{M}$ and $\mathbf{K}$ are the mass and stiffness matrices, respectively, and α and β are the coefficients of proportionality.

The problem of position control for soft manipulators was solved by obtaining the inverse kinematic relationships of two different types of robots. The implementation of the simulation

of the model was then used to directly pilot one of the manipulators given a desired position of the end-effector in feed-forward control.

The feed-forward control of the robots relies entirely on its model. Because of the lack of material parameters, the open-loop system does not account for non-linear behaviors such as viscosity. The closed-loop controller proposed in this papers was proven to be able to reject these model uncertainties and improve the overall behavior of the manipulator. Moreover, the proposed controller can be used even when high Jacobian inversion errors are present.

It is important to remark that the method is no longer viable when we leave the quasi-static motion case, and for the moment, the sampling rate required to capture vibrations in the robot is not feasible. Nevertheless, this first approach to the kinematics and control for soft manipulator opens up some interesting perspectives for future work:

- The model of the tendons does not account for the friction between the cable and the guides it passes through. Including a term in the formulation of the direct model to account for the friction can be done, but the way it will change the inverse model should be investigated.
- Given the information provided by the FEM model, a study on the impedance control of the robot is feasible. The information regarding the compliance of the robot can be directly extracted from the FEM.

VII. Author Disclosure Statement

No competing financial interests exist.

VIII. Acknowledgments

This research was part of the project COMOROS supported by ANR (Tremplin-ERC) and the Conseil Régional Haut-de-France and the European Union through the European Regional Development Fund (ERDF).

References

[1] S. Neppalli, B. Jones, W. McMahan, V. Chitrakaran, I. Walker, M. Pritts, M. Csencsits, C. Rahn, and M. Grissom, "Octarm-a soft robotic manipulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 2569–2569.

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

[3] Q. Zhao and F. Gao, "Design and analysis of a kind of biomimetic continuum robot," in *Proc. IEEE International Conference on Robotics and Biomimetics*, 2010, pp. 1316–1320.

[4] T. Zheng, Y. Yang, D. T. Branson, R. Kang, E. Guglielmino, M. Cianchetti, D. G. Caldwell, and G. Yang, "Control design of shape memory alloy based multi-arm continuum robot inspired by octopus," in *Proc. IEEE 9th Conference on Industrial Electronics and Applications*, 2014, pp. 1108–1113.

[5] J. S. Mehling, M. A. Diftler, M. Chu, and M. Valvo, "A minimally invasive tendril robot for in-space inspection," in *Proc. IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, 2006, pp. 690–695.

[6] A. Yamada, S. Naka, S. Morikawa, and T. Tani, "Mri compatible continuum robot based on closed elastica with bending and twisting," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 3187–3192.

[7] A. Shiva, A. Stilli, Y. Noh, A. Faragasso, I. De Falco, G. Gerboni, M. Cianchetti, A. Menciassi, K. Althoefer, and H. A. Wurdemann, "Tendon-based stiffening for a pneumatically actuated soft manipulator," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 632–637, 2016.

[8] T. Kato, I. Okumura, S.-E. Song, A. J. Golby, and N. Hata, "Tendon-driven continuum robot for endoscopic surgery: Preclinical development and validation of a tension propagation model," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 5, pp. 2252–2263, 2015.

[9] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *Proc. IEEE International Conference on Robotics and Automation*, 2004, pp. 351–357.

[10] M. Cianchetti, T. Ranzani, G. Gerboni, T. Nanayakkara, K. Althoefer, P. Dasgupta, and A. Menciassi, "Soft robotics technologies to address shortcomings in today's minimally invasive surgery: the stiff-flop approach," *Soft Robotics*, vol. 1, no. 2, pp. 122–131, 2014.

[11] D. Haraguchi, T. Kanno, K. Tadano, and K. Kawashima, "A pneumatically driven surgical manipulator with a flexible distal joint capable of force sensing," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 6, pp. 2950–2961, 2015.

[12] M. W. Hannan and I. D. Walker, "Analysis and initial experiments for a novel elephant's trunk robot," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2000, pp. 330–337.

[13] B. Jones, I. D. Walker *et al.*, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006.

[14] B. Bardou, P. Zanne, F. Nageotte, and M. De Mathelin, "Control of a multiple sections flexible endoscopic system," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 2345–2350.

[15] T. Mahl, A. Hildebrandt, and O. Sawodny, "A variable curvature continuum kinematics for kinematic control of the bionic handling assistant," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 935–949, 2014.

[16] G. S. Chirikjian and J. W. Burdick, "A modal approach to hyper-redundant manipulator kinematics," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 343–354, 1994.

[17] ——, "Kinematically optimal hyper-redundant manipulator configurations," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 794–806, 1995.

[18] D. C. Rucker and R. J. Webster, "Statics and dynamics of continuum robots with general tendon routing and external loading," *IEEE Transactions on Robotics*, vol. 27, no. 6, pp. 1033–1044, 2011.

[19] G. Smoljkic, D. Reynaerts, J. Vander Sloten, and E. Vander Poorten, "Compliance computation for continuum types of robots," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2014, pp. 1066–1073.

[20] B. Jones, R. L. Gray, K. Turlapati *et al.*, "Three dimensional statics for continuum robotics," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2009, pp. 2659–2664.

[21] M. Giorelli, F. Renda, M. Calisti, A. Arienti, G. Ferri, and C. Laschi, "A two dimensional inverse kinetics model of a cable driven manipulator inspired by the octopus arm," in *Proc. IEEE International Conference on Robotics and Automation*, 2012, pp. 3819–3824.

[22] F. Connolly, P. Polygerinos, C. J. Walsh, and K. Bertoldi, "Mechanical programming of soft actuators by varying fiber angle," *Soft Robotics*, vol. 2, no. 1, pp. 26–32, 2015.

[23] E. Coevoet, N. Reynaert, E. Lartigau, L. Schiappacasse, J. Dequidt, and C. Duriez, "Registration by interactive inverse simulation: application for adaptive radiotherapy," *International journal of computer assisted radiology and surgery*, vol. 10, no. 8, pp. 1193–1200, 2015.

[24] C. A. Felippa, "A systematic approach to the element-independent corotational dynamics of finite elements," *Center for Aerospace Structures Document Number CU-CAS-00-03, College of Engineering, University of Colorado*, 2000.

[25] F. Faure, C. Duriez, H. Delingette, J. Allard, B. Gilles, S. Marchesseau, H. Talbot, H. Courtecuisse, G. Bousquet, I. Peterlik *et al.*, "Sofa: A multi-model framework for interactive physical simulation," in *Payan Y. (eds) Soft Tissue Biomechanical Modeling for Computer Assisted Surgery*. Springer, Berlin, Heidelberg, 2012, pp. 283–321.

[26] A. Rodríguez, E. Coevoet, and C. Duriez, "Real-time simulation of hydraulic components for interactive control of soft robots," in *The 2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[27] C. Duriez, "Control of elastic soft robots based on real-time finite element method," in *Proc. IEEE International Conference on Robotics and Automation*, 2013, pp. 3982–3987.

[28] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *Proc. IEEE International Conference on Robotics and Automation*, 2015, pp. 2550–2555.

[29] J. S. Przemieniecki, *Theory of matrix structural analysis*. Courier Corporation, United States, 1985.

[30] A. Fabri and S. Pion, "Cgal: The computational geometry algorithms library," in *Proc. ACM SIGSPATIAL international conference on advances in geographic information systems*, 2009, pp. 538–539.

[31] C. Escande, P. M. Pathak, R. Merzouki, and V. Coelen, "Modelling of multisection bionic manipulator: Application to robotinoxt," in *Proc. IEEE International Conference on Robotics and Biomimetics, Karon Beach, Thailand*, 2011, pp. 92–97.

[32] C. Escande, R. Merzouki, P. M. Pathak, and V. Coelen, "Geometric modelling of multisection bionic manipulator: Experimental validation on robotinoxt," in *Proc. IEEE International Conference on Robotics and Biomimetics*, pp. 2006–2011.

[33] C. Escande, T. Chettibi, R. Merzouki, V. Coelen, and P. M. Pathak, "Kinematic calibration of a multisection bionic manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 20, no. 2, pp. 663–674, 2015.

[34] O. Lakhal, A. Melingui, and R. Merzouki, "Hybrid approach for modeling and solving of kinematics of compact bionic handling assistant manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 3, pp. 1326–1335, 2016.

[35] H. Courtecuisse, J. Allard, C. Duriez, and S. Cotin, "Preconditioner-based contact response and application to cataract surgery," in *Medical Image Computing and Computer-Assisted Intervention*. Springer, 2011, pp. 315–322.

[36] J. Bosman, T. M. Bieze, O. Lakhal, M. Sanz, R. Merzouki, and C. Duriez, "Domain decomposition approach for fem quasistatic modeling and control of continuum robots with rigid vertebras," in *Proc. IEEE International Conference on Robotics and Automation*, 2015, pp. 4373–4378.

[37] M. Ivanescu, N. Bizdoaca, and D. Pana, "Dynamic control for a tentacle manipulator with sma actuators," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 2. IEEE, 2003, pp. 2079–2084.

[38] M. C. Yip and D. B. Camarillo, "Model-less feedback control of continuum manipulators in constrained environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 880–889, 2014.

[39] R. S. Penning, J. Jung, J. A. Borgstadt, N. J. Ferrier, and M. R. Zinn, "Towards closed loop control of a continuum robotic manipulator for medical applications," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4822–4827.

[40] R. S. Penning, J. Jung, N. J. Ferrier, and M. R. Zinn, "An evaluation of closed-loop control options for continuum manipulators," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5392–5397.

[41] T. A. Johansen and T. I. Fossen, "Control allocation—a survey," *Automatica*, vol. 49, no. 5, pp. 1087–1103, 2013.

[42] A. M. Lyapunov, "The general problem of the stability of motion," *International Journal of Control*, vol. 55, no. 3, pp. 531–534, 1992.

[43] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[44] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones," *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[45] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*. IEEE, 2004, pp. 284–289.

[46] S. Marchesseau, T. Heimann, S. Chatelin, R. Willinger, and H. Delingette, "Multiplicative jacobian energy decomposition method for fast porous visco-hyperelastic soft tissue model," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2010, pp. 235–242.