

# Design and Stiffness control for soft manipulators

A bio-inspired approach to soft robots control

Francesco Stella

Thesis Report



# Design and Cartesian stiffness control for soft manipulators

Master thesis of Francesco Stella<sup>a</sup> - 09/09/2021

**Abstract**—Many interesting designs of soft robots with variable stiffness capabilities have been presented in the literature. However, little attention has been given on the control of their embedded physical intelligence.

In this work we present an algorithm that exploits the variable joint stiffness capabilities and the redundancy of a soft manipulator to achieve Cartesian stiffness control at the end effector, thanks to model-based and optimal control techniques. The algorithm is validated both analytically and in the real world. In particular, we present a tendon-driven soft manipulator, equipped with variable-stiffness segments and proprioceptive sensing. The robot is used as a platform to test the algorithm in real tasks, such as fitting a peg in the hole. Thanks to accurate modeling, the soft manipulator is able to obtain the desired stiffness at the end effector over the workspace.

## I. INTRODUCTION

Numerous designs of continuum soft robots, defined as robotic systems with purposefully designed compliant elements embedded into their mechanical structure [1], have been equipped with variable stiffness mechanisms (VSMs) [2],[3],[4]. For example, in [5], a variable-stiffness manipulator based on antagonistic pneumatic actuators is presented. Similarly, [2] proposes a VS assistive manipulator that can automate the bathing task for elderly people, while in [4], a 1 Degree of Freedom (DOF) robot with VS capabilities is presented. The VSMs are usually introduced in the designs to generically tune the grade of compliance with an unknown environment, allowing the robot to be robust or compliant with respect to external disturbances. To the best of the authors' knowledge, in soft continuum robots, the stiffness is usually controlled at the joint level, without any model on the resulting end effector stiffness in Cartesian space. Indeed, in order to define the relation between the VSM and the end effector Cartesian stiffness, an accurate model of the structure of the manipulator is needed. While controlling the stiffness at the joint level may be sufficient for some applications, a specification of Cartesian stiffness is required for various tasks, such as assembly of parts or sliding along surfaces [6][7]. Therefore, we aim to realize a desired Cartesian stiffness at the tool tip by exploiting the compliant structure of the robot itself [8], augmented with a VSM. In order to do so, we propose a soft manipulator modellable by the Piecewise Constant Curvature (PCC) assumption and equipped with a VSM, together with an optimization-based algorithm for end effector Cartesian stiffness control.

### Contribution

With this paper we propose:

- the design of a soft manipulator with VS, proprioceptive and tactile capabilities, able to control its end effector Cartesian stiffness.
- a general optimal control algorithm for Cartesian stiffness control of the end effector using model-based techniques for soft manipulators.

## II. FUNCTIONAL DESCRIPTION OF THE ROBOT

The robot is a soft manipulator composed of three flexible segments, connected in series and actuated by 9 tendons. Each segment is composed of an elastic chamber able to vary its stiffness and by a rigid platform, in which the motors and Inertial Measurement Units (IMUs) are embedded. The control of each segment is independent, as the actuated tendons are attached on two consecutive platforms.

The soft robot is able:

- to reconstruct the posture of the whole body when externally deformed thanks to the IMUs spread along the structure.
- to detect the contact with external objects along all its body, providing it of a tactile skin, thanks to pressure sensors attached to each of the chambers composing the segments.
- to vary the stiffness of each joint independently, thanks to a variable stiffness pneumatic mechanism.

### A. Hardware design

The robot is built using 3d printing, molding and casting techniques.

1) *Connecting plates and tendon actuation*: Each 3d-printed platform is composed of two concentric circles and three external housings for the motors and IMUs, as shown in Figure 2. The central circle is used as an attachment for the air chamber, while the second circle keeps the electronic wires and the air tubes in place.

Each tendon is actuated by a Micro Metal DC motor with a 398:1 gearbox. This gear ratio has been chosen to provide sufficient torque with a reasonable speed range. Each motor is equipped with a quadrature encoder, to enable both the direction and position to be determined with high precision. The pulleys, connected to the motors, were 3d printed to get an optimal transmission ratio between motor and tendon turns. The motors are controlled by H-bridges, to provide speed and direction control. An Arduino Due microcontroller is used to control the H-bridges. The encoders are connected to digital interrupts on the microcontroller to allow continuous monitoring. A serial protocol has been developed to allow

<sup>a</sup>: Faculty of Mechanical Engineering, Department of Cognitive Robotics, 3mE, Delft University of Technology, Delft, The Netherlands

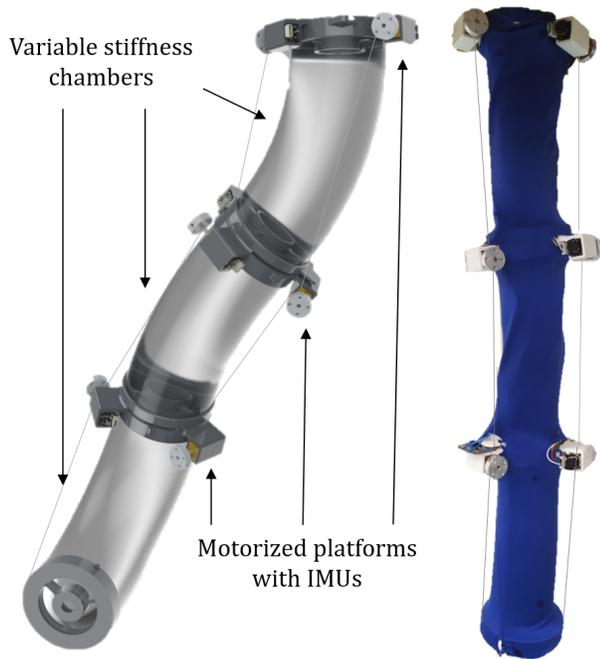


Fig. 1: Rendered and real images of the soft manipulator. A fabric sleeve is placed on top of the manipulator to keep the wiring and tubing in place.

communication between the high level control program running on Matlab in the PC and the Arduino. This allows a demand position to be set for each motor. The speed of each motor is controlled with a PID controller, integrated in the Matlab script running on the PC so to converge to the required encoder position in closed loop.

2) *Chambers and pneumatic control*: As shown in Figure 2, each segment is composed of an inflatable chamber. The chambers are built by casting a silicon material (Dragonskin-20) in four stages, as described in Figure 4. The molds are 3D printed so to be easily assemblable and allow a simple removal of the cured chamber<sup>1</sup>. The chamber is composed of a central beam with three longitudinal communicating cavities around. The inner beam withstands most of the loads in the uninflated configuration and limits the deformation in the axial direction, while the chambers, connected trough a tube to a pressure source, allow the segment to change its bending stiffness. Indeed, thanks to the nonlinear behaviour of the compressed air in the chamber, the bending stiffness can be controlled as a function of the internal pressure, with an increase in stiffness up to the 300% in the operational range.

Indeed, in the uninflated state, the external wall is free to bend, without any energy penalty other than its own structural stiffness. On the contrary, when inflated, the external wall of the chamber goes under tension, following the Young-Lagrange equation [9]. The nonlinear air behaviour, together with the nonlinear stress-strain relation of Dragonskin-20 [3] makes the bending stiffness of the whole structure increase

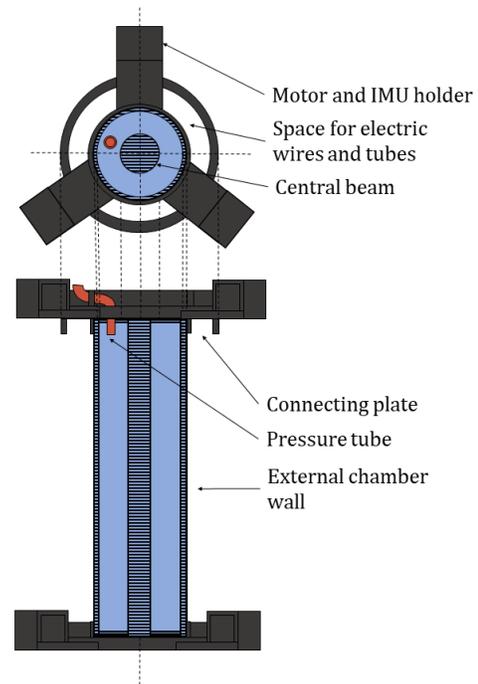


Fig. 2: Technical drawing of a single segment, composed of a chamber and two platforms. Each platform is built in a concentric fashion, with an external hole for connecting wires and tubes and an internal attachment for the chamber, composed of a central flexible beam and an elastic wall.



Fig. 3: The chamber deflection due to the 50g load at the tip decrease when the internal pressure increase. Interestingly, the tip displacement is smaller when the chamber is fully inflated, i.e. the stiffness of the chamber increases with the internal pressure.

when the chamber is inflated. In Figure 3 it is possible to notice how the chamber, loaded at the tip, deforms as a function of the internal pressures.

The bending stiffness of the chamber has been evaluated as a function of the internal pressure, as in Figure 6. A measurement setup, shown in Figure 5, has been developed to evaluate the stiffness of each chamber as a function of the internal pressure. The chamber is deformed by pushing the tip through a load cell (Futek LSB200) embedded on an actuated platform. The position of the platform is measured with a laser sensor (optoNCDT 1420). The internal pressure is measured with a pressure sensor (Gems Pressure Sensor350) connected to the chamber via a tube.

<sup>1</sup>\*: for the interested reader, the full CAD model of the mold, together with the rigid platforms constituting the robot, is available at "<https://grabcad.com/library/mold-chamber-softmanipulator-1>"

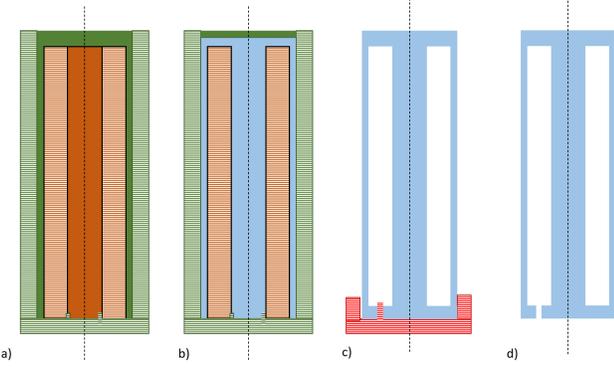


Fig. 4: **Molding process:** a) The first mold, composed of two parts, in green and orange, is assembled, b) the Dragonskin-20 is poured and it is left to cure for 4 hours, c) the cured Dragonskin-20 is removed from the first mold, and put in a second mold, represented in red, in which some more Dragonskin-20 has been poured. d) Finally, the cured chamber can be removed from the mold and attached to the air tube and platforms.

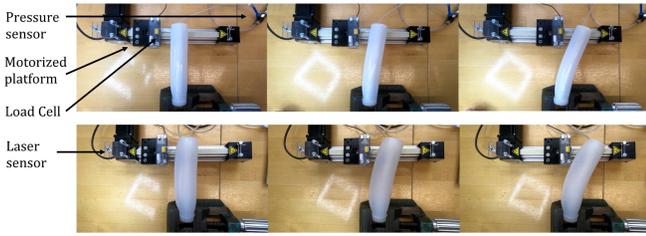


Fig. 5: the bending stiffness of the chamber has been evaluated from displacement vs force curves for several internal pressures. The experiment has been repeated five times for each pressure level  $P = 1, 1.05, 1.1, 1.15, 1.2 \text{ bar}$ . Finally, the stiffness is evaluated as the slope of the first order regression of the data.

Hence the position, pressure data and the force exerted by the chamber on the load cell are collected with a data acquisition board (NI Multi-channel data acquisition module USB-6000). The setup is controlled with a LabView programme that saves the data for further processing and controls the position of the platform.

Thanks to this experiment, it is possible to approximate at the first order the stiffness of the  $i$ -th chamber as a function of the internal pressure  $s$ . To do so, we have to transform the force exerted by the tip  $F_x$  and the displacement of the tip  $\Delta x$  data into the bending stiffness  $K_{\text{bending}}$  and the compression stiffness  $K_{\text{compression}}$ , later used to model the stiffness of the whole robot. The length of the segment  $l$ , the radius of curvature of the segment  $r$  and the PCC curvature angle  $\theta$  are linked by  $l = r\theta$ . Therefore, the kinematic relation between

the displacement of the tip and the bending angle  $\theta$  is:

$$\Delta x = r(1 - \cos(\theta)) = \frac{l}{\theta}(1 - \cos(\theta)) \quad (1)$$

For the small displacements used in the experiment ( $\theta \approx 0$ ), this relation can be approximated with a McLaurin expansion truncated at the second order, which brings to the result

$$\Delta x = \frac{l\theta}{2!} \quad (2)$$

We can then write the bending stiffness as:

$$K_{\text{bending}} = \frac{M}{\theta} = \frac{F_x l^2}{2! \Delta x} \quad (3)$$

where  $M$  represents the moment applied to the chamber and  $\theta$  the bending angle of the chamber. By performing a least squares regression on the  $\frac{F_x}{\Delta x}$  we are hence able to find the first order approximation of the stiffness-pressure relation as a function of the internal pressure  $s$ :

$$K_{\text{bending}_i}(s) = k_{p_0} + g(s_i - p_0) \quad (4)$$

where  $k_{p_0}$  is the stiffness at the atmospheric pressure  $p_0$  and  $g$  is the slope of the first order approximation of the stiffness/pressure relation as in Figure 6.

Hence, using a structural model of slender beams, we can find the relation between the bending and compression stiffness by solving the system:

$$\begin{cases} K_{\text{compression}} = \frac{3EI_c}{l^3} \\ K_{\text{bending}} = \frac{EI_{xx}}{l} \end{cases} \quad (5)$$

where  $E$  is the elastic modulus of the material, and  $I_c$  and  $I_{xx}$  represents the inertia respect to the central and radial axes respectively. Note that thanks to the pressure sensors connected to the chamber, it is also possible to identify external loads on the chamber. Indeed, when the internal pressure deviates from the model by a statistically significant amount, we can infer that it is due to an external pressure on the robot. This phenomenon can be developed to equip the robot of tactile-like sensing along the whole body

3) *Electronic setup:* Two Arduino Due microcontrollers are used to control the H-bridges and to record encoders, IMUs and pressure sensors data. The encoders and the pressure sensors are connected to digital interrupts on the microcontroller to allow continuous monitoring, while the IMUs data are connected to the Arduinos via an i2c communication. A serial protocol has been developed, to allow communication between the control program running on the PC and the Arduinos. This allows a demand position to be set for each motor. The PC, then sets the direction and speed of the motor following a PID control scheme so to move to the desired encoder position, as described by the algorithm in Section V. Similarly, the IMUs and pressure data are sent and postprocessed on the PC in real time with the algorithms described in Section IV. A block diagram of the electronic architecture is shown in Figure 7.

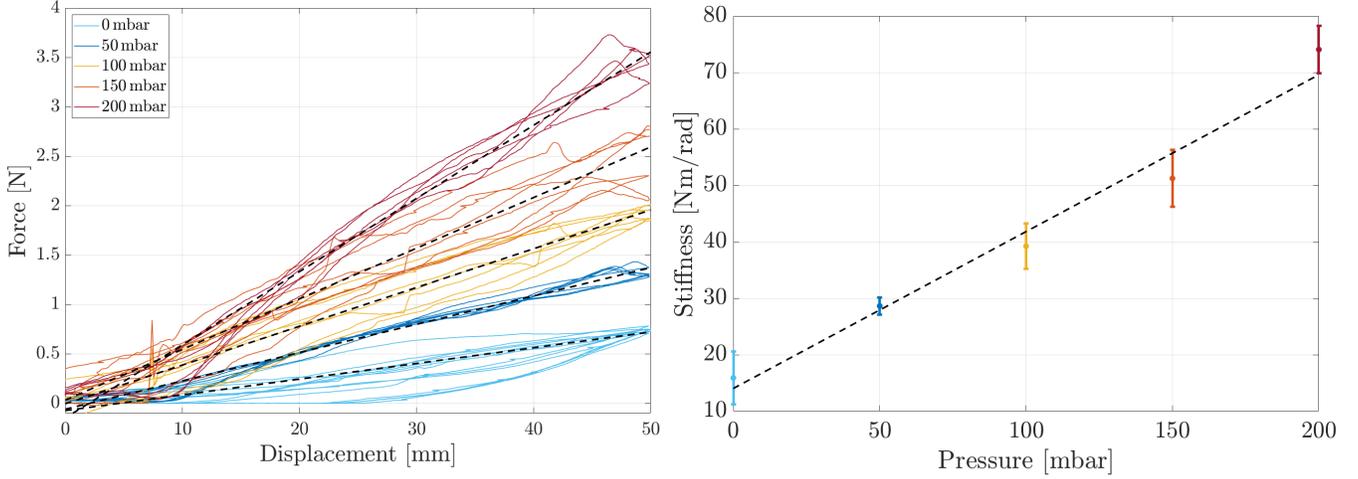


Fig. 6: Pressure-stiffness relation as a function of 5 internal pressures. **Left:** force displacement curve over 5 experiments for each pressure level, together with the least square regression. The slope of the curve represents the bending stiffness, increases with the internal pressure. **Right:** The regression line as presented in Equation 4, where the data points are the stiffnesses for each pressure level.

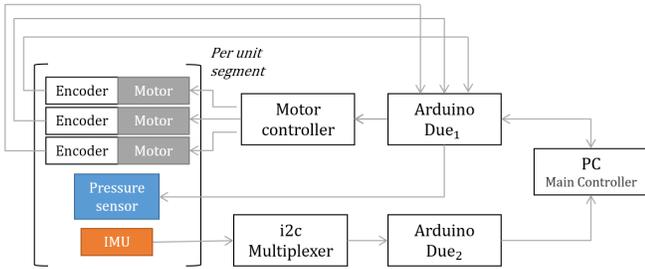


Fig. 7: System architecture of the electronics, showing the connections between the key components and the flow of data through the control system.

### III. ROBOT KINEMATIC MODEL

As presented in [10], in cable driven robots, the tendon action on the beam can be approximated as a constant moment. If the assumption of constant moment is inserted in the Euler Bernoulli equation, the result is a beam with constant curvature. Note that this assumption is valid only in a steady state condition. Indeed, when some power is exchanged from the environment to the robot, this assumption is not valid anymore and the soft robot will be eventually deformed out of the Constant Curvature (CC) kinematic model [11]. Nonetheless, this assumption has been demonstrated to approximate accurately the behavior of many continuum robots [12], [13], [14], [15], [16], [17]. We can then take this hypothesis as a basis for a kinematic model with a computationally treatable number of DOFs. Therefore, it is reasonable to describe the robot as a series of three arcs with constant curvature, i.e. by adopting a PCC model. For each segment the arc of constant curvature can be univocally defined by just three Lagrangian variables, as shown in Figure 8: the angle of circumference spanned by the segment  $\theta$ , the

angle of the plane containing the arc  $\phi$ , and arc length  $L$ .

As presented in [18], the full pose of the  $i$ -th segment can be completely described by the rotation matrix  $\mathbf{R}_{i-1}^i(\theta, \phi)$  and translation mapping  $\mathbf{t}_{i-1}^i(\theta, \phi, L)$ :

$$\mathbf{R}_{i-1}^i = \begin{bmatrix} C_{\phi_i}^2(C_{\theta_i} - 1) + 1 & S_{\phi_i}C_{\phi_i}(C_{\theta_i} - 1) & C_{\phi_i}S_{\theta_i} \\ S_{\phi_i}C_{\phi_i}(C_{\theta_i} - 1) & S_{\phi_i}^2(C_{\theta_i} - 1) + 1 & S_{\phi_i}S_{\theta_i} \\ -C_{\phi_i}S_{\theta_i} & -S_{\phi_i}S_{\theta_i} & C_{\theta_i} \end{bmatrix} \quad (6)$$

$$\mathbf{t}_{i-1}^i = \frac{L}{\theta} [C_{\phi_i}(C_{\theta_i} - 1) \quad S_{\phi_i}(C_{\theta_i} - 1) \quad S_{\theta_i}]^T \quad (7)$$

with  $C_{\phi_i}$ ,  $C_{\theta_i}$ ,  $S_{\phi_i}$ ,  $S_{\theta_i}$  being  $\cos(\phi_i)$ ,  $\cos(\theta_i)$ ,  $\sin(\phi_i)$ ,  $\sin(\theta_i)$  respectively.

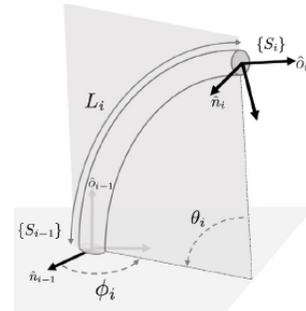


Fig. 8: Under the assumption of piecewise constant curvature, each section of a soft robot can be defined by three parameters: length  $L$ , curvature  $\theta = \frac{k}{L}$  and angle of orientation of the plane of bending  $\phi$ . Image from [19].

Moreover, we also use the equivalent description of a CC arc presented in [20] for pose reconstruction. In this model, the coordinates  $\phi$  and  $\theta$  are reparametrized into  $\Delta_{x,i}, \Delta_{y,i} \in \mathbb{R}$ . Hence the rotation matrix can be written as

$$\mathbf{R}_{i-1}^i(\Delta_{x,i}, \Delta_{y,i}, s_i) = \begin{bmatrix} 1 + \frac{\Delta_{x,i}^2}{\Delta^2} \xi & \frac{\Delta_{x,i} \Delta_{y,i}}{\Delta^2} \xi & -\frac{\Delta_{x,i}}{\Delta} \sin(s_i \Delta_i) \\ \frac{\Delta_{x,i} \Delta_{y,i}}{\Delta^2} \xi & 1 + \frac{\Delta_{y,i}^2}{\Delta^2} \xi & -\frac{\Delta_{y,i}}{\Delta} \sin(s_i \Delta_i) \\ \frac{\Delta_{x,i}}{\Delta} \sin(s_i \Delta_i) & \frac{\Delta_{y,i}}{\Delta} \sin(s_i \Delta_i) & 1 + \xi \end{bmatrix} \quad (8)$$

and the position can be modeled by the vector:

$$\mathbf{t}_{i-1}^i(\Delta_{x,i}, \Delta_{y,i}, L_i, s_i) = \frac{L_i}{\Delta_i^2} \begin{bmatrix} (1 - \cos(s_i \Delta_i)) \Delta_{x,i} \\ (1 - \cos(s_i \Delta_i)) \Delta_{y,i} \\ \sin(s_i \Delta_i) \Delta_i \end{bmatrix} \quad (9)$$

where  $\Delta_i = \sqrt{\Delta_{x,i}^2 + \Delta_{y,i}^2}$  and  $\xi = (1 - \cos(s_i \Delta_i))$ .  $s_i \in [0, 1]$  is the local coordinate along the segment, with 0 referring to the base, and 1 to the tip. In the following, both models are used, as the first model allows an intuitive interpretation of the deformations, which is pivotal for modeling the stiffnesses, while the second model is free of the singularity in  $\theta = 0$  and is therefore more robust. It is then used in the pose reconstruction algorithm.

In both of these Lagrangian parametrizations of the arcs,  $\mathbf{t}_{i-1}^i$  and  $\mathbf{R}_{i-1}^i$  represent the rotation matrix and the translation mapping the reference frame at the base of the  $i$ -th segment to the one attached to its tip. The two can be combined in a homogeneous transformation mapping from the frame at the bottom of the segment to the frame at the top of the segment as:

$$\mathbf{T}_{i-1}^i = \begin{bmatrix} \mathbf{R}_{i-1}^i & \mathbf{t}_{i-1}^i \\ [0 \ 0 \ 0] & 1 \end{bmatrix} \quad (10)$$

Finally, as schematized in Figure 9, another advantage of the PCC assumption is that the whole kinematics can be decomposed into two mappings [21]: a first mapping  $q_{PCC} = m(\mathbf{h})$  from actuator coordinates  $\mathbf{h}$ , i.e. the length of the tendons to configuration space parameters that describe CC arcs  $\mathbf{q}$  and a second mapping  $\mathbf{x} = f(\mathbf{q}) = f(m(\mathbf{h}))$  that maps the PCC coordinates into Cartesian coordinates.

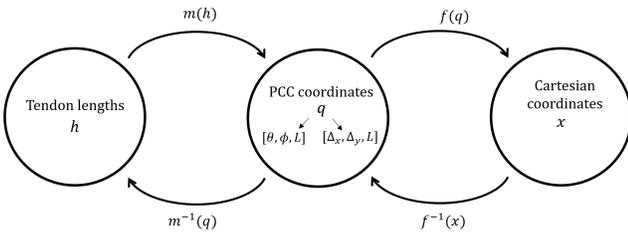


Fig. 9: the three spaces and mappings between them which define the kinematics of constant-curvature robots. The mapping  $m(\mathbf{h})$  transforms actuator space variables  $\mathbf{h}$  to configuration space variables  $\mathbf{q}$ . Next, the mapping  $f(\mathbf{q})$  takes these configuration space variables to the task space, in this case, the Cartesian space.

#### IV. POSE RECONSTRUCTION ALGORITHM

Here we propose an algorithm that allows the robot to reconstruct its pose when deformed by external forces thanks to the IMUs embedded at the end of each segment. Since an IMU is embedded in each platform, it is possible to use the posture provided by a stack of Mahony filters [22] to extract the global posture of each reference frame  $\hat{\mathbf{R}}_{i-1}^i$ . Then the

relative orientation of two consecutive platforms is evaluated as  $\hat{\mathbf{R}}_{i-1}^i = (\hat{\mathbf{R}}_0^i)^{-1} \hat{\mathbf{R}}_0^i$ . Finally, the parameterization of each segment is extracted by properly inverting the rotation matrix defined in Equation (8) for  $s_i = 1$ . This is done by evaluating the configuration of the segment as the solution of the optimization problem

$$\arg \min_{\Delta_{x,i}, \Delta_{y,i} \in \mathbb{R}^2} \|\mathbf{R}_{i-1}^i(\Delta_{x,i}, \Delta_{y,i}, 1) - \hat{\mathbf{R}}_{i-1}^i\|_F^2, \quad (11)$$

where  $\|\cdot\|_F$  is the Frobenius norm, and  $\mathbf{R}_{i-1}^i(\Delta_{x,i}, \Delta_{y,i}, 1)$  is given by Equation (8). This is evaluated by starting from the initial guess

$$\begin{aligned} \Delta_{x,i} &= \frac{1}{2} \frac{(\hat{R}_{i-1}^i[3, 1] - \hat{R}_{i-1}^i[1, 3]) \arccos(\hat{R}_{i-1}^i[3, 3])}{\sin(\arccos(\hat{R}_{i-1}^i[3, 3]))}, \\ \Delta_{y,i} &= \frac{1}{2} \frac{(\hat{R}_{i-1}^i[3, 2] - \hat{R}_{i-1}^i[2, 3]) \arccos(\hat{R}_{i-1}^i[3, 3])}{\sin(\arccos(\hat{R}_{i-1}^i[3, 3]))}, \end{aligned} \quad (12)$$

where  $\hat{R}_{i-1}^i[j, k]$  is the element of  $\hat{R}_{i-1}^i$  positioned at row  $j$  and column  $k$ . Note that if the PCC hypothesis were perfectly fulfilled, then  $\mathbf{R}_{i-1}^i(\Delta_{x,i}, \Delta_{y,i}) = \hat{\mathbf{R}}_{i-1}^i$  - i.e. this would be the global minimum of Equation (11). Yet the real system will likely have a non perfectly constant bending. Therefore, a standard gradient descent method is used to locally refine the guess. In this way the residual drift is tamed by the constraints imposed by the kinematic model. Finally, full postures (i.e. both orientation and position) of each point along the segment can be retrieved by substituting back the parameterization into Equations (8) and (9). A more thorough explanation of the method can be found in our previous work [23].

#### V. STIFFNESS CONTROL ALGORITHM

With a soft robot, we would like to realize a desired Cartesian stiffness at the end effector. To realize this behavior, we can use the mechanically variable joint stiffness and - in case of a redundant robot - the nullspace of the robot.

For a general robotic application it is most natural to specify the desired stiffness behaviour of the robot in Cartesian coordinates. The desired values would result from a task description with respect to the ee. The user may specify a desired constant stiffness matrix  $\mathbf{K}_x = -\frac{\delta \mathbf{f}}{\delta \mathbf{x}} \in \mathbb{R}^{r \times r}$  as the relation between the Cartesian wrench  $\mathbf{f}$  and the Cartesian displacement  $\mathbf{x}$ . Here,  $r$  is the number of Cartesian DOF. On the other hand, the robot will be able to adjust the stiffness  $\mathbf{K}_J = -\frac{\delta \tau}{\delta \mathbf{q}}$  of its  $n$  joints, where  $\tau$  is the joint torque and  $\mathbf{q}$  is the joint position. In particular, we propose to model the joint stiffness matrix  $\mathbf{K}_J$  of the manipulator presented above by computing the Hessian of the elastic potential

$$U = \sum_i^n \left[ \frac{1}{2} k_{\text{bending}} \theta_i^2 + \frac{1}{2} k_{\text{compression}} L_i^2 \right] \quad (13)$$

Therefore  $\mathbf{K}_J(\mathbf{k}_J, \mathbf{q})$  is a function of the three values of stiffness of the chamber

$$\mathbf{k}_J(\mathbf{s}) = [k_{1\text{bending}}(s_1), k_{2\text{bending}}(s_2), k_{3\text{bending}}(s_3)]^T \quad (14)$$

which can be remapped in terms of the internal pressures  $s_i$ .

With this section we propose a novel algorithm<sup>2</sup> that, given a goal Cartesian coordinate  $\mathbf{x}_d \in \mathbb{R}^r$ , and a desired Cartesian stiffness matrix  $\mathbf{K}_x \in \mathbb{R}^{r \times r}$ , finds the optimal joint position  $\mathbf{q}_{\text{opt}} \in \mathbb{R}^n$  and the joint stiffness matrix  $\mathbf{K}_J \in \mathbb{R}^{n \times n}$  which provides the best approximation of  $\mathbf{K}_x$  under the constraints  $\mathbf{x}_d = f(\mathbf{q}_{\text{opt}})$  and lower and upper bounds on joint stiffness and  $\mathbf{q}$ . In particular, the algorithm is developed for redundant robots, i.e. robots in which  $r > n$ , so to exploit the nullspace  $\mathcal{N} \in \mathbb{R}^{r-n}$  of configurations that satisfy the constraint on the end effector position.

First, in order to develop a Cartesian stiffness controller, it is pivotal to define a relation between the Cartesian stiffness and the joint space, i.e. a transformation  $\Gamma: \mathbf{K}_J = \Gamma(\mathbf{K}_x)$ . As presented in [24], the  $\Gamma$  mapping can be written as:

$$\mathbf{K}_J = -\frac{\delta\tau}{\delta\mathbf{q}} = -\frac{\delta(\mathbf{J}(\mathbf{q})^T \mathbf{K}_x \Delta\mathbf{x})}{\delta\mathbf{q}} = \quad (15)$$

$$= \mathbf{J}(\mathbf{q})^T \mathbf{K}_x \mathbf{J}(\mathbf{q}) - \frac{\delta\mathbf{J}(\mathbf{q})^T}{\delta\mathbf{q}} \mathbf{K}_x \Delta\mathbf{x} \quad (16)$$

where  $\mathbf{J}$  represents the manipulator Jacobian  $\mathbf{J}(\mathbf{q}) = \frac{\delta f(\mathbf{q})}{\delta\mathbf{q}}$ , where  $f(\mathbf{q})$  is the forward kinematics mapping.  $\Delta\mathbf{x} = \mathbf{x}_d - \mathbf{x}$  is the Cartesian displacement between the desired and the actual position. If the stiffness is computed around the equilibrium position (i.e.  $\Delta\mathbf{x} = \mathbf{0}$ ), Equation 16 reduces to:

$$\mathbf{K}_J = \mathbf{J}(\mathbf{q})^T \mathbf{K}_x \mathbf{J}(\mathbf{q}) \quad (17)$$

In this work, we will focus on the reduced form in Equation 17 as we desire to find the optimal pose that in equilibrium satisfies the constraint on the end effector position.

Under this simplification, we are able to solve the inverse stiffness problem  $\Gamma^{-1}$ , i.e. computing the resulting Cartesian stiffness matrix  $\mathbf{K}_x = \Gamma^{-1} \mathbf{K}_J$  for a given  $\mathbf{K}_J$  by:

$$\mathbf{K}_x = \mathbf{J}(\mathbf{q})^{+T} \mathbf{K}_J \mathbf{J}(\mathbf{q})^+ \quad (18)$$

where  $+$  denotes the pseudoinverse operator. However, as computing  $\mathbf{J}(\mathbf{q})^+$  can be analytically untreatable and computationally heavy, the inverse problem is better solved in terms of compliances through the relations  $\mathbf{C}_x = \mathbf{K}_x^{-1}$ ,  $\mathbf{C}_J = \mathbf{K}_J^{-1}$  [25], i.e.

$$\mathbf{C}_x = \mathbf{J}(\mathbf{q}) \mathbf{C}_J \mathbf{J}(\mathbf{q})^T \quad (19)$$

As presented in [6], it is highly improbable that the desired Cartesian stiffness can be achieved as a perfect solution due to the non surjective mapping from joint stiffnesses to Cartesian stiffnesses. Therefore we can search for a good approximation of it via the nonconvex optimization problem:

$$\min \|\mathbf{C}_{x_{\text{desired}}} - \mathbf{J}(\mathbf{q}) \mathbf{C}_J(\mathbf{k}) \mathbf{J}(\mathbf{q})^T\|_{Fr} \quad (20)$$

$$\text{s.t. } \mathbf{x}_{\text{desired}} - f(\mathbf{q}) = 0 \quad (21)$$

$$\mathbf{l}_{\text{bound}} \leq \mathbf{q} \leq \mathbf{u}_{\text{bound}} \quad (22)$$

$$\mathbf{l}_{\text{bound}} \leq \mathbf{k}_J \leq \mathbf{u}_{\text{bound}} \quad (23)$$

where  $\|\cdot\|_{Fr}$  is the Frobenius norm.

However, it is possible to notice that finding the optimal stiffness for a fixed configuration  $\mathbf{q}_{\text{fix}}$  is a convex problem in

the Riemannian manifold. Indeed, the optimal stiffness value for a given pose can be found by solving the more efficient convex optimization problem of for the compliance vector of the chambers  $\mathbf{c}_J(\mathbf{s})$  i.e. a vector created from the element-wise inversion of  $\mathbf{k}_J(\mathbf{s})$ , at the extremes of the cost function  $\|\mathbf{C}_{x_{\text{desired}}} - \mathbf{J}(\mathbf{q}) \mathbf{C}_J(\mathbf{k}_J) \mathbf{J}(\mathbf{q})^T\|_{Fr}$  i.e.

$$\min \mathbf{C}_q^T \mathbf{H}(\mathbf{q}, \mathbf{C}_{x_{\text{desired}}}) \mathbf{C}_J + \mathbf{b}(\mathbf{q}, \mathbf{C}_{x_{\text{desired}}})^T \mathbf{C}_q \quad (24)$$

$$\text{s.t. } \mathbf{l}_{\text{bound}} \leq \vec{k}_J \leq \mathbf{u}_{\text{bound}} \quad (25)$$

where

$$\mathbf{H} = \frac{\delta^2}{\delta \mathbf{c}_J^2} |\mathbf{C}_x - \mathbf{J} \mathbf{C}_J \mathbf{J}^T|_{Fr} \quad (26)$$

$$\mathbf{b} = \frac{\delta}{\delta \mathbf{c}_J} |\mathbf{C}_x - \mathbf{J} \mathbf{C}_J \mathbf{J}^T|_{Fr, \mathbf{c}_J=0} \quad (27)$$

This extremely efficient joint stiffness optimization step can now be used as a sub-procedure in the following, unfortunately much more complicated, non convex optimization step for the nullspace of the robot.

In order to move efficiently in the manifold defined by  $\mathbf{x}_{\text{desired}} - f(\mathbf{q}) = 0$ , i.e. the nullspace  $\mathcal{N}$  of the robot, it is important to develop a tool to project a random configuration on the manifold in an efficient manner. This can be achieved through Algorithm 1, which using a gradient descent with the update  $\Delta \mathbf{q}_{i+1} = -\mathbf{J}^T (\mathbf{J} \mathbf{J}^T)^{-1} \Delta \mathbf{x}(\mathbf{q}_i)$  projects a random configuration  $\mathbf{q}_0$  into a configuration  $\mathbf{q} \in \mathcal{N}$ . Note that in [26], it is proved that the projection operator covers the whole constraint manifold. That is, for any configuration on the manifold, there exist configurations within the ambient space that will be projected onto the configuration.

---

#### Algorithm 1 Projection on manifold

---

**while true do**

$\Delta x \leftarrow$  Displacement From Constraint( $q$ )

**if**  $|\Delta x| \leq \epsilon$  **then**

        return  $q$

**end**

$J \leftarrow$  EvaluateJacobian( $q$ )

$\Delta q_{\text{error}} \leftarrow J^T (J J^T)^{-1} \Delta x$

$q \leftarrow (q - \Delta q_{\text{error}})$

**end**

---

Once the initial random position is projected on the manifold  $\mathcal{N}$  it is possible to create a local orthonormal basis  $\Phi(q)$  for the tangent space to the manifold, i.e. reconstructing the Atlas of the manifold. From [27], an orthonormal basis can be created by finding a solution to:

$$\begin{pmatrix} \mathbf{J}(\mathbf{q}) \\ \Phi(\mathbf{q})^T \end{pmatrix} \Phi(\mathbf{q}) = \begin{pmatrix} 0 \\ \mathbf{I} \end{pmatrix} \quad (28)$$

that is finding the kernel of the Jacobian. Hence a random point  $q_u$  can be computed starting from the chart  $C_q = (q, \Phi(q))$  as:

$$\mathbf{q}_u = \mathbf{q} + \Phi(\mathbf{q}) \mathbf{u}_q \quad (29)$$

<sup>2</sup>\*: for the interested reader, please refer to the "https://github.com/fstella97/SoftManipulator" repository for full the implementation of the algorithm in Matlab

Note that  $\mathbf{q}_u$  will be close to the manifold but still must be projected to  $\mathcal{N}$ . Hence  $\mathbf{q}_u$  is projected on the manifold  $\mathcal{N}$  *orthogonally* to the tangent space of  $\mathbf{C}_J$  in which it is embedded. The projected configuration  $\mathbf{q}_m$  is found by solving the system:

$$\begin{cases} \mathbf{x}_{\text{desired}} - f(\mathbf{q}_m) = 0 \\ \Phi(\mathbf{q})^T(\mathbf{q}_m - \mathbf{q}) = 0 \end{cases} \quad (30)$$

with a gradient descent method. In Algorithm 2, the following equations are used in the descent:

$$\mathbf{A}_u(\mathbf{q}_m) = \begin{pmatrix} \mathbf{J}(\mathbf{q}_m) \\ \Phi(\mathbf{q})^T \end{pmatrix},$$

$$\mathbf{b}_u(\mathbf{q}_m) = \begin{pmatrix} \mathbf{x}_{\text{desired}} - f(\mathbf{q}_m) \\ \Phi(\mathbf{q})^T(\mathbf{q}_m - \mathbf{q}_u) \end{pmatrix}$$

---

#### Algorithm 2 Perpendicular Projection algorithm

---

**Input:**  $q_u$   
**Output:**  $q_m$

$q_m \leftarrow q_u$   
 $b \leftarrow b_u(q_m)$   
**while**  $|b|_2 > \epsilon$  **do**  
     $q_m \leftarrow q_m - A_u(q_m)^{-1}b$   
     $b \leftarrow b_u(q_m)$   
    **if**  $|b|_2 \leq \epsilon$  **then**  
        **return**  $q_m$   
    **end**  
**end**

---

By iterating Algorithm 2 it is possible to walk on the manifold i.e. moving from adjacent configurations which satisfy the constrain on the end effector position.

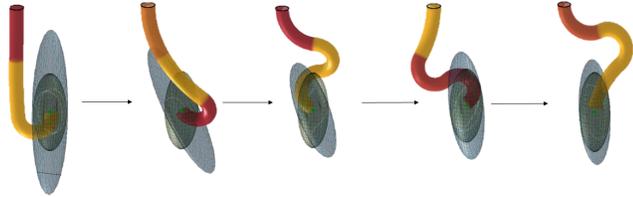


Fig. 11: best performing poses and stiffnesses over the optimization, where the color of the chamber indicate the stiffnesses  $k_J$ . Thanks to the optimization procedure it is possible to select the configuration and joint stiffness  $[\mathbf{q}, \mathbf{s}]$  so that the modelled stiffness, represented as a blue ellipsoid minimize the error with respect to the desired stiffness, displayed as a green ellipsoid. In the figure it is possible to see how the optimization algorithm progressively converges toward best performing poses.

Given Algorithms 1 and 2, we propose to optimize on the manifold by developing a genetic algorithm that moves only on the nullspace  $\mathcal{N}$ . Therefore we implemented a genetic algorithm in which populations generated over the whole joint space are projected to  $\mathcal{N}$  through Algorithm 1. Hence, for each of these configurations  $\in \mathcal{N}$ , it is possible to efficiently compute the optimal stiffness through the convex optimization problem

described in Equation 24. Finally, the performance of each configuration is evaluated as the Frobenius norm between the desired stiffness and the resulting stiffness ellipsoid. Hence, in the next generation, new individuals are added through Algorithm 1, in order to better cover  $\mathcal{N}$  and the zones with the best performing configurations are further explored by generating children through Algorithm 2.

In Figure 12 it is possible to notice that the population, on average, converges toward better performing individuals over the computation. Finally, the pose with the minimal cost on the stiffness difference is chosen as presented in Figure 11. Finally the desired pose is translated into motor coordinates through the mapping  $m^{-1}(\mathbf{q})$  and the desired stiffness in the chambers is translated into the desired internal pressure  $s$  with Equation 4. Each motor is controlled to the desired position in closed loop with a PID controller based on the encoder readings. The chambers are inflated with an external pressure source based on the optimal pressure state.

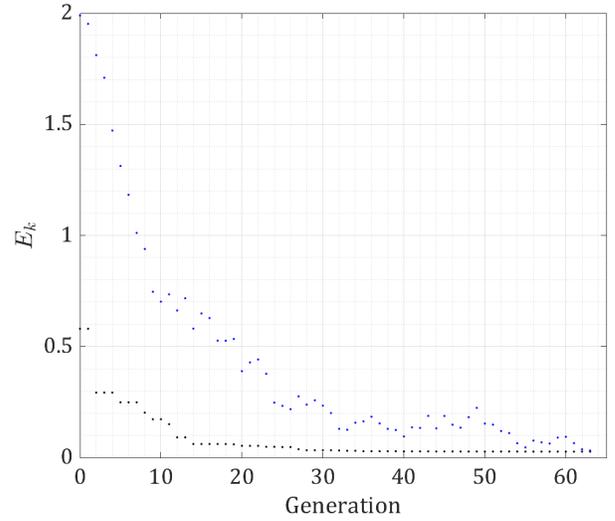


Fig. 12: convergence of the genetic algorithm over the generations. The best performer is reported in black, while the mean fitness value of the whole population at the  $i$ -th generation is shown in blue.

## VI. VALIDATION OF THE ALGORITHM

### A. Analytical validation

In [25], the concept of stiffnessity is introduced as a tool to understand the effects of the optimization parameters on the reachable stiffness for varying positions in the workspace. The high dimensionality (6 DOF) of the robotic arm in the 3d Cartesian space does not allow for an intuitive visualization. However, given the symmetry about the plane defined by  $\phi$ , it is possible to capture the whole Cartesian workspace just by looking at one plane. Moreover, we represent here each stiffness matrix as an ellipsoid, with the principal axes being the eigenvectors of the  $K$  matrix. We aim to quantify the performance of the achieved stiffness optimized via Equation 20. To be able to compare results with different stiffness amplitudes and orientations, we use the normalized Frobenius norm

$$E_k = \frac{\|K_{\text{desired}} - K_q\|_{\text{Fr}}}{\|K_{\text{desired}}\|_{\text{Fr}}} \quad (31)$$

The normalization provides the relative deviation with respect to the desired stiffness value. A value of  $E_k = 0$  means perfect tracking of the desired stiffness value. A value of  $E_k = 1$  means errors of the magnitude of the desired stiffness values arise. Firstly, in Figure 13, it is possible to notice that, with the optimization method described in Section V, which combines the VSM and Nullspace optimization, we achieve good performances over the workspace (mean( $E_k$ ) = 0.33, std( $E_k$ ) = 0.46). Moreover, it is possible to evaluate the effects of the VSM and of the nullspace search on the performance individually. Interestingly, when optimizing only between the configurations  $\in \mathcal{N}$ , we perform better than if we optimize only the VSM for fixed configurations. Moreover, if we optimize only on the configurations  $\in \mathcal{N}$  the optimization performance is worst in zones in which the nullspace is smaller, i.e. the reachability map [28], shown in Figure 14 and stiffnessity map are positively correlated.

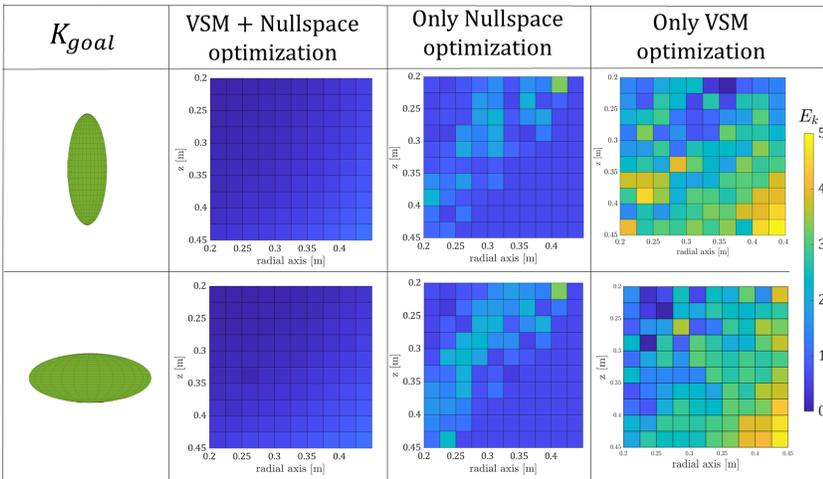


Fig. 13: stiffability map for different desired stiffness orientations and optimization criteria. **Left:** the result of the optimization with both variable stiffness and nullspace freedom. **Center:** the error heatmap in which only nullspace optimization is considered, and the stiffness of each chamber is kept constant. **Right:** the heatmap in which only nullspace optimization is considered, and the stiffness of each chamber is kept constant.

### C. Task-based validation

Thanks to the algorithm presented in Section V, the robot is able to complete tasks in which the directional control of the stiffness ellipsoid is crucial, such as inserting a peg in the hole. Indeed, in assembly tasks that require a tight fitting, the robot will have to be stiff in the direction of the hole, so to be able to push the peg inside, and compliant in the perpendicular plane, so to adjust for misalignments. Therefore we can command the robot to execute a peg insertion just by providing the position of the hole and the desired direction as an input. In Figure 16, it is possible to see the convergence the robot toward the hole over two scenarios. Interestingly, the control scheme has no constraints on the end effector orientation. Moreover, thanks to the embedded softness the robot is able to comply with unmodeled obstacles, such as the wall in Figure 16.

### B. Experimental evaluation

An experiment is performed to merge the analytical results from the optimization with the physical performance of the robot. Given the solution of the algorithm described in section V The stiffness ellipsoid of the end effector is evaluated perturbing the soft robot with the robotic manipulator Franka Emika Panda [29] along 6 directions, shown in Figure 17. During the experiment length  $o$ , both the displacement of the end effector  $\Delta x \in R^{3 \times o}$  and the reaction forces  $F \in R^{3 \times o}$  are recorded. Hence, the experimental ellipsoid matrix  $K_{exp} \in R^{3 \times 3}$  evaluated by minimizing the least square error as:

$$K_{\text{Experiment}} = F \Delta x^+ \quad (32)$$

Over 8 experiments, the end-effector stiffness well match the the desired stiffness ellipsoid, with an average error of  $E_k = 0.43$ .

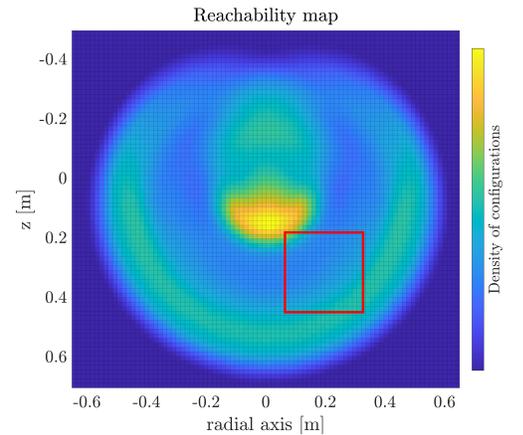


Fig. 14: Reachability map. Thanks to the reachability map we can have a empirical measure of the amplitude of the nullspace inside the workspace. The red rectangle highlight the section of the workspace used for the optimization evaluation

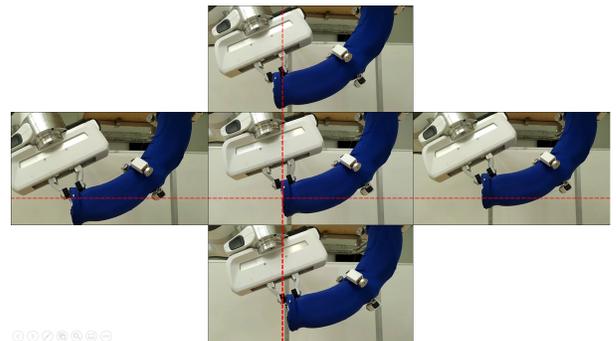


Fig. 17: experiment representation: the soft manipulator in  $q_{\text{opt}}, k_{\text{opt}}$  is perturbed along the  $x, y$  and  $z$  axes by a second manipulator, attached at the end effector. During the experiment the position and reaction force of the manipulator are recorded for later postprocessing.

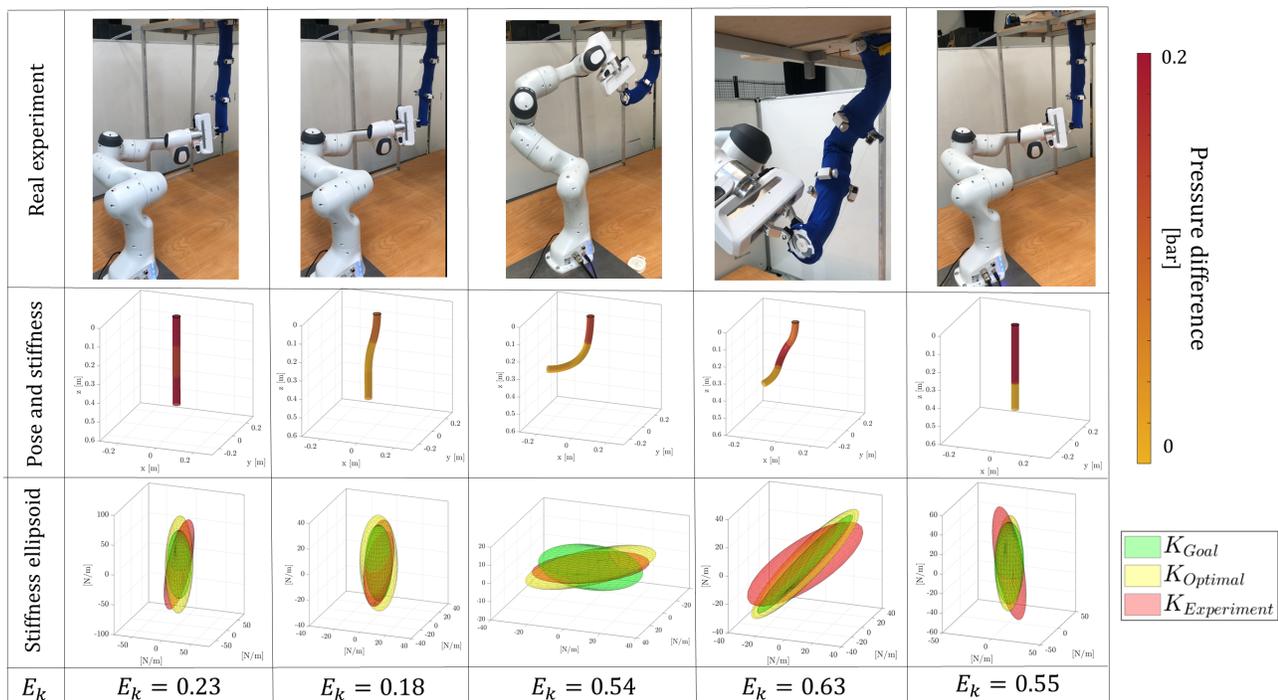


Fig. 15: **Top:** Images of the real experiments. **Center:**  $q_{opt}$  configuration and the optimal stiffness per chamber resulting from the optimization. **Bottom:** the respective the Goal, Optimal and Experimental stiffness ellipsoid are shown, together with the  $E_k$  metric. The robot is able to well approximate the desired stiffness at the end effector.

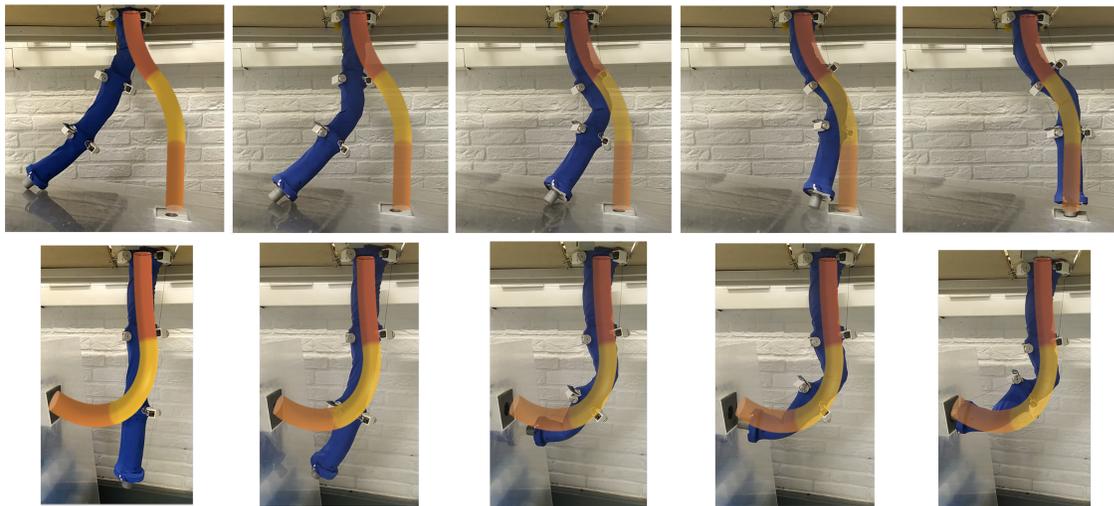


Fig. 16: Thanks to accurate control the robot was able to complete insert a peg in the hole just by knowing the position of the hole and the desired stiffness ellipsoid. Note that no constraints on the end-effector orientation have been given.

## VII. CONCLUSION

The work shows a general algorithm for Cartesian stiffness control on model-based soft robots. The algorithm has been validated on a carefully modeled soft manipulator equipped with a pressure based variable stiffness mechanism. We were able to obtain the desired Cartesian stiffness ellipsoid at the end effector with constant accuracy over the workspace. Moreover, with a computational analysis of the algorithm, the achievable theoretical possible accuracy on the stiffness over the workspace has been evaluated. Finally, analytical and experimental results show that being able to exploit

the redundancy of the robot and its own structural stiffness, i.e. the possibility of the manipulator to adjust its pose, is more efficient than equipping soft manipulators with variable stiffness mechanisms in order to achieve directional stiffness control. The presented methodology expands soft robot capabilities to tasks in which an accurate control on the stiffness is central, such as assembly, polishing tasks, or unknown environment exploration. Moreover, this work presents a step forward toward bringing well known control methods for human-machine interaction, such as impedance control, to soft robotics.

## REFERENCES

- [1] C. Della Santina, M. Catalano, and A. Bicchi, "Soft robots," *Springer Encyclopedia of robotics*, 2021.
- [2] Y. Ansari, M. Manti, E. Falotico, M. Cianchetti, and C. Laschi, "Multi-objective optimization for stiffness and position control in a soft robot arm module," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 108–115, 2017.
- [3] T. Ranzani, G. Gerboni, M. Cianchetti, and A. Menciassi, "A bioinspired soft manipulator for minimally invasive surgery," *Bioinspiration & biomimetics*, vol. 10, no. 3, p. 035008, 2015.
- [4] M. T. Gillespie, C. M. Best, and M. D. Killpack, "Simultaneous position and stiffness control for an inflatable soft robot," in *2016 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2016, pp. 1095–1101.
- [5] S. M. Babu, A. Sadeghi, A. Mondini, and B. Mazzolai, "Antagonistic pneumatic actuators with variable stiffness for soft robotic applications," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 283–288.
- [6] A. Albu-Schaffer and G. Hirzinger, "Cartesian impedance control techniques for torque controlled light-weight robots," in *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292)*, vol. 1. IEEE, 2002, pp. 657–663.
- [7] N. Hogan, "Impedance control: An approach to manipulation," in *1984 American control conference*. IEEE, 1984, pp. 304–313.
- [8] D. Zappetti, R. Arandes, E. Ajanic, and D. Floreano, "Variable-stiffness tensegrity spine," *Smart Materials and Structures*, vol. 29, no. 7, p. 075013, 2020.
- [9] I. Müller and P. Strehlow, *Rubber and rubber balloons: paradigms of thermodynamics*. Springer Science & Business Media, 2004, vol. 637.
- [10] C. Li and C. D. Rahn, "Design of continuous backbone, cable-driven robots," *J. Mech. Des.*, vol. 124, no. 2, pp. 265–271, 2002.
- [11] T. George Thuruthel, Y. Ansari, E. Falotico, and C. Laschi, "Control strategies for soft robotic manipulators: A survey," *Soft robotics*, vol. 5, no. 2, pp. 149–163, 2018.
- [12] M. W. Hannan and I. D. Walker, "Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots," *Journal of robotic systems*, vol. 20, no. 2, pp. 45–63, 2003.
- [13] N. Simaan, R. Taylor, and P. Flint, "A dexterous system for laryngeal surgery," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, vol. 1. IEEE, 2004, pp. 351–357.
- [14] I. S. Godage, D. T. Branson, E. Guglielmino, G. A. Medrano-Cerda, and D. G. Caldwell, "Shape function-based kinematics and dynamics for variable length continuum robotic arms," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 452–457.
- [15] V. Falkenhahn, T. Mahl, A. Hildebrandt, R. Neumann, and O. Sawodny, "Dynamic modeling of bellows-actuated continuum robots using the euler-lagrange formalism," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1483–1496, 2015.
- [16] R. K. Katschmann, C. Della Santina, Y. Toshimitsu, A. Bicchi, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *2019 2nd IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019, pp. 454–461.
- [17] C. Della Santina, A. Bicchi, and D. Rus, "Dynamic control of soft robots with internal constraints in the presence of obstacles," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6622–6629.
- [18] R. J. Webster III and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *The International Journal of Robotics Research*, vol. 29, no. 13, pp. 1661–1683, 2010.
- [19] C. Della Santina, R. K. Katschmann, A. Biechi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *2018 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2018, pp. 46–53.
- [20] C. Della Santina, A. Bicchi, and D. Rus, "On an improved state parametrization for soft robots with piecewise constant curvature and its use in model based control," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1001–1008, 2020.
- [21] B. A. Jones and I. D. Walker, "Kinematics for multisection continuum robots," *IEEE Transactions on Robotics*, vol. 22, no. 1, pp. 43–55, 2006.
- [22] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *IEEE Transactions on automatic control*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [23] J. Hughes, F. Stella, C. D. Santina, and D. Rus, "Sensing soft robot shape using imus: An experimental investigation," in *International Symposium on Experimental Robotics*. Springer, 2020, pp. 543–552.
- [24] J. K. Salisbury, "Active stiffness control of a manipulator in cartesian coordinates," in *1980 19th IEEE conference on decision and control including the symposium on adaptive processes*. IEEE, 1980, pp. 95–100.
- [25] F. Petit, "Analysis and control of variable stiffness robots," Ph.D. dissertation, ETH-Zürich, 2014.
- [26] Z. Kingston, M. Moll, and L. E. Kavraki, "Exploring implicit spaces for constrained sampling-based planning," *The International Journal of Robotics Research*, vol. 38, no. 10-11, pp. 1151–1178, 2019.
- [27] W. C. Rheinboldt, "Manpak: A set of algorithms for computations on implicitly defined manifolds," *Computers & Mathematics with Applications*, vol. 32, no. 12, pp. 15–28, 1996.
- [28] F. Zacharias, C. Borst, and G. Hirzinger, "Capturing robot workspace structure: representing robot capabilities," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Ieee, 2007, pp. 3229–3236.
- [29] F. Emika, "Panda," *tech. rep.*, 2018.