

Dynamic Control of Soft Robots with Internal Constraints in the Presence of Obstacles

Cosimo Della Santina¹, Antonio Bicchi^{2,3}, Daniela Rus¹

Abstract—The development of effective reduced order models for soft robots is paving the way toward the development of a new generation of model based techniques, which leverage classic rigid robot control. However, several soft robot features differentiate the soft-bodied case from the rigid-bodied one. First, soft robots are built to work in the environment, so the presence of obstacles in their path should always be explicitly accounted by their control systems. Second, due to the complex kinematics, the actuation of soft robots is mapped to the state space nonlinearly resulting in spaces with different sizes. Moreover, soft robots often include internal constraints and thus actuation is typically limited in the range of action and it is often unidirectional. This paper proposes a control pipeline to tackle the challenge of controlling soft robots with internal constraints in environments with obstacles. We show how the constraints on actuation can be propagated and integrated with geometrical constraints, taking into account physical limits imposed by the presence of obstacles. We present a hierarchical control architecture capable of handling these constraints, with which we are able to regulate the position in space of the tip of a soft robot with the discussed characteristics.

I. INTRODUCTION

Soft-bodied robots are robotic systems made of continuously deformable elastic elements [1]. Thanks to their inherently safe behavior, and their ability of changing their shape in ways not possible for their rigid counterparts, soft robots promise to strongly expand the range of applicability of robotic systems.

Given their very innovative nature, the large part of the research in the field has been devoted to establishing novel mechanical design principles. Only more recently attention has been focused on developing effective brains - i.e. controllers - and the problem have revealed to be a very challenging one. For an extensive review of current methods in soft robots control, the interested reader can refer to [2]. A main issue preventing the translation of the many results already available for the classic rigid bodied case has been the theoretically infinite dimensionality of the soft body. This issue has been tackled in the last few years, by the introduction of several effective finite dimensional models [3]–[5]. Among them, piecewise constant curvature models

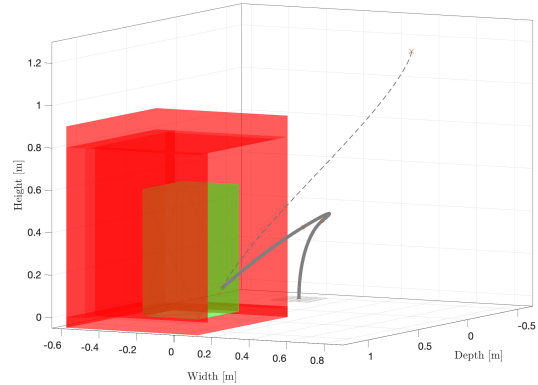


Fig. 1. An example of application of the proposed framework. A soft bodied robot is controlled to reach the green area, while avoiding the red one. The robot is composed of three segments. Each one is actuated with four chambers, able to produce only negative internal forces - modeling the behavior of a vacuum actuated robot. The initial position of the robot's tip is highlighted with a red cross in the up right of the picture, and the trajectory followed so far with a dashed gray line. The target is reached in few seconds.

have affirmed themselves as a simple yet effective way of describing the behavior of soft robots. Their many applications range from design [6], to sensing [7], to kinematic control [8], to dynamic control [9]–[11]. However, the difficulties in developing reliable models are not the only challenge that differentiate control of rigid and soft robots. Others include the necessity of coping with strongly under actuated input spaces [12], and the need of building controllers that do not sensibly alter the natural softness of the robot [13].

The purpose of this paper is to tackle another main challenge setting soft-bodied robots apart from the more classic rigid-bodied ones. In the latter, actuators are connected directly to the physical components defining the state - i.e. the robot's joints - effectively providing to the controller direct access to the wrenches acting on the state variables. This is not the case for soft-bodied robots. Indeed, the state of a soft robot has no localized counterpart in the physical system. The challenge is strongly exacerbated by the physical characteristics of soft actuators. Soft robots are indeed typically actuated with components that can generate forces only in a limited range, often even in only one direction. Fig. 2(c) depicts a classic configuration of actuators in a fluid actuated systems [11], [14], [15]. Each actuated segment of the soft robot has a certain number of chambers embedded into it. Inflating or de-pressurizing one produces a force acting on the internal walls of the segment, bending it. However, only positive or negative actions can be produced in this way for a given system, depending on if the robot is pressure or vacuum powered. This results into a unilateral actuation characteristics, connected through non linear non

This work was supported by the National Science Foundation (grant NSF 1830901 and NSF 1226883). We are grateful for this support.

¹C. Della Santina, and D. Rus are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 32 Vassar St., Cambridge, MA 02139, USA, cosimodellasantina@gmail.com, rus@csail.mit.edu

²A. Bicchi is with Research Center "E. Piaggio" and the Dept. of Information Engineering, University of Pisa, Italy

³A. Bicchi is with "Soft Robotics for Human Cooperation and Rehabilitation" Lab, Istituto Italiano di Tecnologia, Genoa, Italy, antonio.bicchi@iit.it

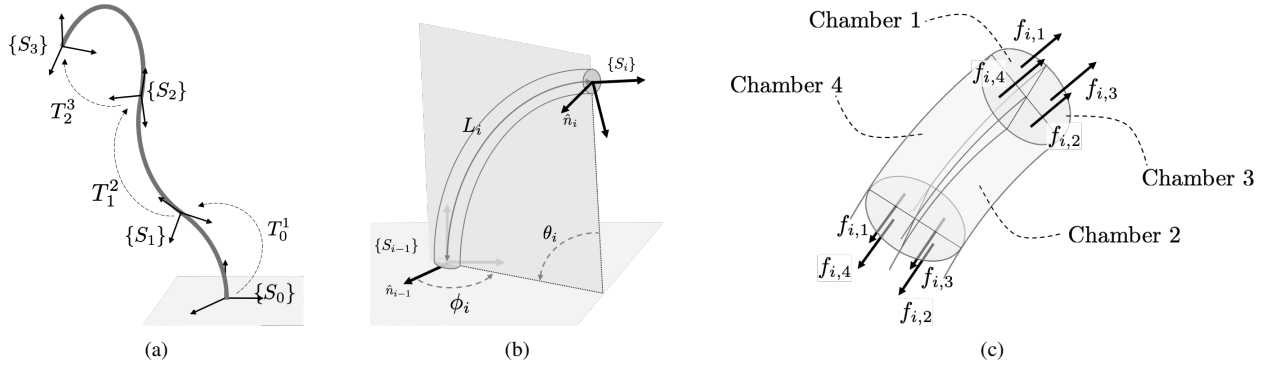


Fig. 2. Three views of a piecewise constant curvature robot. Panel (a) shows a schematic representation of a PCC robot composed of three segments. The frames $\{S_i\}$ are reported, with the corresponding transformations T_{i-1}^i . Panel (b) depicts a single segment, and defines the three variables specifying the configuration, i.e. curvature angle θ_i , angle of bending ϕ_i , and segment length $L_i = L_{0,i} + \delta L_i$. Panel (c) shows a scheme of the actuation mechanism for a single segment. Four chambers actuate it, generating four independent internal forces $f_{i,1}, f_{i,2}, f_{i,3}, f_{i,4}$. These elements can only be inflated or deflated, thus the forces are or all positive or all negative, depending on the design.

square applications to the actual wrenches that a model-based controller would specify.

We propose here a control pipeline solving this challenge in the piecewise constant curvature case. Using the proposed framework, a soft robot with the discussed characteristics can reach areas of interest with its tip, by exploiting its ability of continuously deforming to avoid obstacles, and always remaining into configurations that can be sustained by its actuation system. See Fig. 1 for an example of application. More specifically, the main contributions of this work are

- A strategy for propagating constraints on the actuation, and integrating them with constraints on configuration;
- A control architecture dealing with these constraints;
- The definition and analytical solution of the optimal control allocation problem;
- A planning algorithm, regulating the tip position while guaranteeing the feasibility of the solution;
- Simulations validating the theoretical results.

II. SOFT ROBOT WITH PIECEWISE CONSTANT CURVATURE AND VARIABLE LENGTH

A Piecewise Constant Curvature (PCC) soft robot, is a mechanical system composed by a sequence of continuously deformable segments, with curvature constant in space (CC) but variable in time, merged so that the resulting curve is everywhere differentiable. Fig. 2(a) presents an example of a soft robot composed by three CC segments. We consider here PCC soft robots for which also the length of each segment can change independently. Given the discussed importance in the soft robotics literature, it is beyond the scope of the present paper to further discuss the ability of PCC models to describe soft robots' behavior. See instead [6]–[11], just to cite a few. Note that of the following subsections, only the one about kinematics describes state of the art results. The rest are a novel contribution of this work.

A. Kinematics

Consider a PCC robot composed by n CC segments connected in series. We introduce n reference frames $\{S_1\}, \dots, \{S_n\}$ attached at the ends of each segment, plus one fixed base frame $\{S_0\}$. The pair $\{S_{i-1}\}$ and $\{S_i\}$ fully defines the configuration of the i -th segment. Fig. 2(b)

shows the kinematics of a single CC segment. The configuration of the segment can be described through three variables; i) the angle ϕ_i between the plane $\hat{n}_{i-1} - \hat{o}_{i-1}$ and the plane on which the bending occurs, ii) the relative rotation θ_i between the two reference systems expressed on that plane, iii) and the change in length δL of the central arch.

We call T_{i-1}^i the homogeneous transformations mapping S_{i-1} into S_i . It can be evaluated using geometrical considerations to be

$$T_{i-1}^i(\phi_i, \theta_i, \delta L_i) = \begin{bmatrix} R_{i-1}^i(\phi_i, \theta_i) & t_{i-1}^i(\phi_i, \theta_i, \delta L_i) \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

where

$$R_{i-1}^i = \begin{bmatrix} c_{\phi_i}^2(c_{\theta_i} - 1) + 1 & s_{\phi_i}c_{\phi_i}(c_{\theta_i} - 1) & -c_{\phi_i}s_{\theta_i} \\ s_{\phi_i}c_{\phi_i}(c_{\theta_i} - 1) & c_{\phi_i}^2(1 - c_{\theta_i}) + c_{\theta_i} & -s_{\phi_i}s_{\theta_i} \\ c_{\phi_i}s_{\theta_i} & s_{\phi_i}s_{\theta_i} & c_{\theta_i} \end{bmatrix}$$

$$t_{i-1}^i = \frac{L_{0,i} + \delta L_i}{\theta_i} [c_{\phi_i}(c_{\theta_i} - 1) \quad s_{\phi_i}(c_{\theta_i} - 1) \quad s_{\theta_i}]^T,$$

with c_{ϕ_i} , s_{ϕ_i} , c_{θ_i} , s_{θ_i} being $\cos(\phi_i)$, $\sin(\phi_i)$, $\cos(\theta_i)$, $\sin(\theta_i)$ respectively. In the following we refer to $q_i = [\phi_i \quad \theta_i \quad \delta L_i]^T \in \mathbb{R}^3$ as configuration of the i -th segment, and to $L_{0,i}$ as its rest length. $q \in \mathbb{R}^n$ is the configuration of the soft robot, and it collects q_i for all the segments.

B. Dynamics

We introduce the following model to describe the dynamics of a PCC robot with non constant length

$$B(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + Kq + D(q)\dot{q} = \tau,$$

$$\tau = g(q)f, \quad (1)$$

$$f \in \mathbb{I}_f,$$

where $q \in \mathbb{R}^n$ is the configuration of the robot, with its time derivatives $\dot{q}$, $\ddot{q}$. $B(q) \in \mathbb{R}^{n \times n}$ is the inertia matrix, $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ collects Coriolis and centrifugal terms, $G(q) \in \mathbb{R}^n$ is the gravity force, Kq is the elastic force, with $K \in \mathbb{R}^{n \times n}$ stiffness matrix, and $D(q)\dot{q}$ is the friction force, with $D(q)$ damping. For the sake of space, we do not go into the details of the analytical derivation of these terms. Since the robot has not constant length the necessary steps differ from the ones we presented in previous publications. However, they can be obtained by using the augmented formulation in [11],

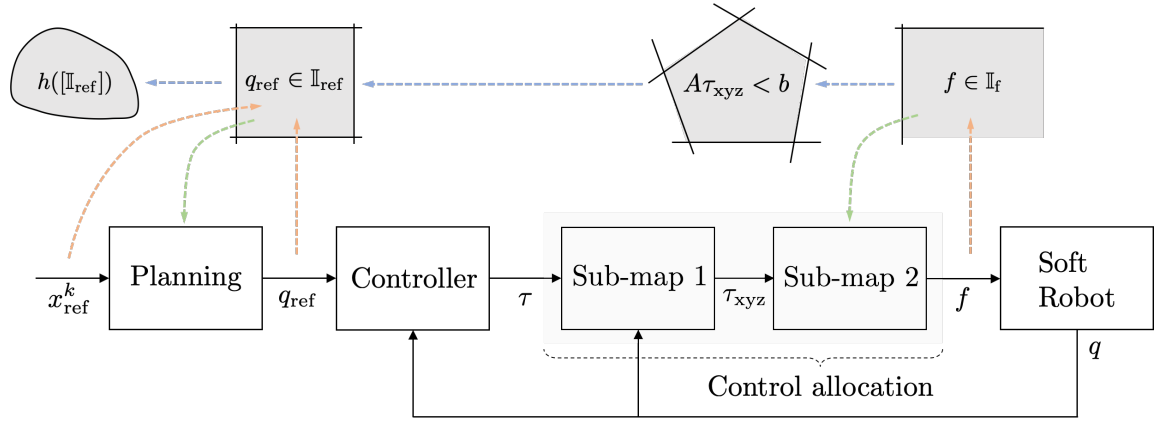


Fig. 3. Scheme summarizing the proposed control pipeline. The bottom of the figure presents the control architecture as a block scheme. The planning algorithm produces a trajectory q_{ref} producing the desired end effector position x_{ref}^k . The controller generates the ideal control action τ necessary to converge on q_{ref} , and the control allocation translates this action in the actual inputs f . The upper part depicts instead the propagation and integration of the constraints, from the actuation f towards the reference x_{ref}^k . The dashed lines highlight this flow. Blue arrows indicate propagation, orange arrows indicate integration, and red arrows underline which part of the control architecture is taking care of guaranteeing the fulfillment of the constraint.

and by substituting $L_{0,i} + \delta L_i$ to any initial occurrence of the there constant length L_i .

We instead discuss more in the details the actuation model - i.e. the second and the third equations in (1) - which is at the core of the present paper. $\tau \in \mathbb{R}^n$ is an hypothetical input that would produce a direct acceleration in the space of the configuration variables. We call $\tau_i \in \mathbb{R}^3$ the vector collecting the $3i-2, 3i-1, 3i$ -th elements of τ . We do not have direct access to τ for the reasons discussed in the introduction. We instead can access $f \in \mathbb{I}_f \subset \mathbb{R}^h$, mapped in τ through $g(q) \in \mathbb{R}^{n \times h}$. $\mathbb{I}_f$ is a set with interval geometry, describing the constraints on the actuation. More specifically, it contains only vectors with positive elements for pressure actuated systems, and with only negative ones for vacuum actuated systems. As discussed in the introduction, the presence of these constraints, together with $g(q)$, strongly increases the complexity of the control problem.

We introduce now some assumptions on the actuation system. It is worth underlying that they are not vital to build a solution, and the results can be easily generalized to the general case. They are instead made for the sake of conciseness and clarity. We assume each segment to have its set of chambers, that actuates it separately from the other segments, as in Fig. 2(c). We consider four chambers per segment. We call $f_i \in \mathbb{R}^4$ the vector collecting the $4i-3, 4i-2, 4i-1, 4i$ -th elements of f . As a consequence, $g(q)$ is block diagonal, with blocks $g_i(q_i) \in \mathbb{R}^{3 \times 4}$. Its general form is $g_i(q_i) = J_i^T(q)R_0^i(q) - J_{i-1}^T(q)R_0^{i-1}(q)$, where J_i is the Jacobian matrix mapping $\dot{q}_i$ into linear velocities aligned to the directions of the forces f_i . R_0^i is the rotation matrix mapping $\{S_i\}$ to $\{S_0\}$. Interestingly, it can be shown that g_i depends only on the configuration of the i -th segment q_i .

C. Constraints on configuration

Since the main application of soft robots is to act into the environment, we introduce a set of constraints on the configuration modeling the presence of areas in which the robot can and cannot move. We call $\mathbb{T}_i \subseteq \mathbb{R}^3$ with $i \in \{1, \dots, N_{\text{target}}\}$ the target sets. We want at least one point of each target set to be reachable by the robot with its tip. $\mathbb{O} \subseteq \mathbb{R}^3$ is the set of points that the robot must avoid,

i.e. the obstacles. $\mathbb{I}_q \subseteq \mathbb{R}^n$ is the interval set of admitted configurations.

III. PROPOSED SOLUTION

To solve the control problem we propose a framework based on two main components; a control architecture, and a procedure for integrating the constraints so that the architecture can manage them. Fig. 3 graphically represents both the flows.

The control architecture starts with *planning*. The role of this algorithm is to produce a path in configuration space $q_{\text{ref}} \in \mathbb{R}^n$ moving the tip of the soft robot from $x_{\text{ref}}^{k-1} \in \mathbb{R}^3$ to $x_{\text{ref}}^k \in \mathbb{R}^3$, while always satisfying the constraints. The *controller* has the role of generating an action $\tau \in \mathbb{R}^n$ such that the state $q, \dot{q}$ converges to the desired one $q_{\text{ref}}, \dot{q}_{\text{ref}}$. This part of the architecture works as if a set of inputs directly acting on the states were available - see (1). This latter level of abstraction is implemented by the *control allocation*, which evaluates on-line the forces f necessary to obtain τ , while always satisfying the constraints.

The procedure for constraints integration and propagation works instead backward. First the interval constraints $f \in \mathbb{I}_f$ are propagated back, into polyhedral constraints on τ_{xyz} . Then these constraints are integrated with constraints on q and x , resulting in a lower bound of the feasible region with interval geometry $\mathbb{I}_{\text{ref}}$.

We describe in details each component of the framework in the following sections.

IV. OPTIMAL CONTROL ALLOCATION

A. Allocation problem and proposed solution

We perform this analysis for $\mathbb{I}_f = \{f \text{ s.t. } f \leq 0\}$. As discussed above, this is the case typically encountered when the robot is vacuum actuated, since a de-pressurization can only produce inward forces (see Fig. 2(c)). This will allow a direct inspection of the results which would not be possible in the general case. The solution for $\mathbb{I}_f = \{f \text{ s.t. } f \geq 0\}$ - i.e. pressure powered system - is discussed at the end of the section. Upper and lower bounds can also be considered with similar results, that we can not report for the sake of space.

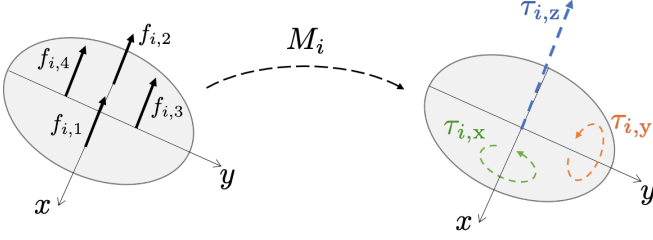


Fig. 4. To solve the optimal allocation problem, we introduce an intermediate representation $\tau_{i,xyz} \in \mathbb{R}^3$ of the control action - on the right of the figure. It stays in between $f_i \in \mathbb{R}^4$ - left of the figure - and $\tau_i \in \mathbb{R}^3$. The latter is not shown in figure since it does not have a direct physical interpretation due to the complex parametrization of the soft robot. The components of τ_{xyz} are the torque around the local x axis $\tau_{i,x}$, around the local y axis $\tau_{i,y}$, and the force in the direction of the local z axis. A constant matrix $M_i \in \mathbb{R}^{3 \times 4}$ maps f_i into $\tau_{i,xyz}$.

The proposed optimal control allocation problem is

$$\min_{f_i} \|f_i\|^2 \quad \text{s.t. } f_i \leq 0 \quad \text{and} \quad \tau_i = g_i(\phi_i, \theta_i, \delta L_i) f_i. \quad (2)$$

Solving it assures at the same time that f_i satisfies the constraints - i.e. it generates the desired wrench τ_i and it is physically achievable - and that unnecessary internal forces are never produced. The direct analytical solution of (2) is hard due to the complex dependency of the problem from f_i and $\phi_i, \theta_i, \delta L_i$. To deal with this issue we introduce an intermediate representation of the actuation, presented in Fig. 4. In this way the mapping $g_i(\phi_i, \theta_i, \delta L_i) f_i$ is expressed as cascade combination of two simpler maps

$$\tau_i = \alpha_i(\phi_i, \theta_i, \delta L_i) \tau_{xyz,i} \quad \text{and} \quad \tau_{xyz,i} = M_i f_i, \quad (3)$$

where $\tau_{xyz,i} = [\tau_{x,i}, \tau_{y,i}, \tau_{z,i}]^T \in \mathbb{R}^3$, the first two elements begin the torque around the local x and y axes, and the third being the force in the direction of the local z axis. M_i is

$$M_i = \begin{bmatrix} 0 & 0 & \Delta_i & -\Delta_i \\ -\Delta_i & \Delta_i & 0 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad (4)$$

where $\Delta_i \in \mathbb{R}^+$ is the distance of center of each chamber from the central axis. $\alpha_i(\phi_i, \theta_i, \delta L_i)$ is kinematic dependent, and can be evaluated for a generic CC segment as

$$\alpha_i = \begin{bmatrix} -c\phi_i s\theta_i & -s\phi_i s\theta_i & 0 \\ -s\phi_i & c\phi_i & (L_0 + \delta L_i) \frac{\theta_i - s\theta_i}{\theta_i^2} \\ 0 & 0 & \frac{s\theta_i}{\theta_i} \end{bmatrix}. \quad (5)$$

Note that the limit in $\theta_i \rightarrow 0$ is always well defined. So we complete A_i in $\theta_i = 0$ with the limit value

$$\alpha_i(\phi_i, 0, \delta L_i) = \begin{bmatrix} 0 & 0 & 0 \\ -s\phi_i & c\phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

In this way, the control allocation problem is clearly split in two sub-problems. The mapping from τ to τ_{xyz} is square but state dependent. The mapping from τ_{xyz} to f is not square, but state independent. This enables a neat two-steps solution of (2), that we describe in the next subsections.

B. Step 1: square inversion

The first equation of (3) can be inverted by using the following generalized inverse matrix

$$\alpha_i^{-1} \doteq \begin{cases} \begin{bmatrix} -\frac{c\phi_i}{s\theta_i} & -s\phi_i & \frac{s\phi_i(\theta_i - s\theta_i)(L_0 + \delta L_i)}{\theta_i s\theta_i} \\ -\frac{s\phi_i}{s\theta_i} & c\phi_i & -\frac{c\phi_i(\theta_i - s\theta_i)(L_0 + \delta L_i)}{\theta_i s\theta_i} \\ 0 & 0 & \frac{\theta_i}{s\theta_i} \end{bmatrix} & \text{if } \theta_i \neq 0 \\ \begin{bmatrix} 0 & -s\phi_i & 0 \\ 0 & c\phi_i & 0 \\ 0 & 0 & 1 \end{bmatrix} & \text{otherwise.} \end{cases} \quad (7)$$

Note that a generalized inverse so defined assures that $\alpha_i \alpha_i^{-1} = I$ - with I identity matrix - for all $\theta_i \neq 0$. Furthermore, it prevents the production of torques around ϕ when $\theta_i = 0$, being

$$\alpha_i(\phi_i, 0, \delta L_i) \alpha_i^{-1}(\phi_i, 0, \delta L_i) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (8)$$

This is consistent with the fact that the robot has null inertia around ϕ_i in this configuration. We thus define $\tau_{xyz} = \alpha^{-1}(q) \tau$, where $\alpha^{-1} \in \mathbb{R}^{n \times n}$ is the block diagonal matrix having as i -th block α_i^{-1} . This action is implemented by *sub-map 1* in Fig. 3.

C. Step 2: explicit solution of the allocation problem

Using (3) and (7), Eq. (2) can be reformulated as follows

$$\min_{f_i} \|f_i\|^2 \quad \text{s.t. } f_i \leq 0 \quad \text{and} \quad \tau_{xyz,i} = M_i f_i \quad (9)$$

The problem can be further simplified by explicitly solving the equality constraints

$$\begin{bmatrix} f_{i,1} \\ f_{i,2} \\ f_{i,3} \end{bmatrix} = \left(M_i \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right)^{-1} \left(\tau_{xyz,i} - M_i \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \right). \quad (10)$$

Through (10), standard algebraic transformations yield to the equivalent optimization problem

$$\min_{z_i} \frac{1}{2} z_i^2 \quad \text{s.t. } G z_i \leq S_i \tau_{xyz,i} \quad (11)$$

where $z_i = f_{i,4} - \frac{\tau_{x,i}}{4} + \frac{\tau_{y,i}}{2\Delta_i}$, $G = [-1 \quad -1 \quad 1 \quad 1]^T$, and

$$S_i = \begin{bmatrix} 0 & \frac{1}{2\Delta_i} & -\frac{1}{4} \\ 0 & -\frac{1}{2\Delta_i} & -\frac{1}{4} \\ -\frac{1}{2\Delta_i} & 0 & -\frac{1}{4} \\ \frac{1}{2\Delta_i} & 0 & -\frac{1}{4} \end{bmatrix}. \quad (12)$$

The Karush-Kuhn-Tucker optimality conditions for (11) are

$$\begin{aligned} z_i + G^T \lambda &= 0, & \Lambda(G z_i - S_i \tau_{xyz,i}) &= 0, \\ \lambda &\geq 0, & G z_i &\leq S_i \tau_{xyz,i} \end{aligned} \quad (13)$$

where $\lambda \in \mathbb{R}^4$ are the Lagrange multipliers, and $\Lambda \in \mathbb{R}^{4 \times 4}$ is the diagonal matrix having λ as diagonal elements. The first equation in (13) directly yields $z_i = -G^T \lambda$, which in turn allows to rewrite the optimality conditions as

$$\begin{aligned} \Lambda(G G^T \lambda + S_i \tau_{xyz,i}) &= 0 \\ \lambda &\geq 0 & G G^T \lambda + S_i \tau_{xyz,i} &\geq 0. \end{aligned} \quad (14)$$

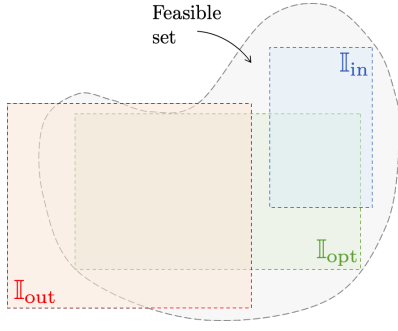


Fig. 5. Three examples of candidate interval sets that could be produced by lines 4 and 5 of algorithm 1. $\mathbb{I}_{\text{out}}$ is outside the feasible set, so it is rejected by lines 7 and 8. The other two intervals would instead pass the check. Among the two, $\mathbb{I}_{\text{opt}}$ would be maintained, while $\mathbb{I}_{\text{in}}$ discarded, since the first has a larger perimeter.

The solution of (14) can be found explicitly by solving the first equation for all the possible combinations of active and inactive constraints, i.e. for all the possible combinations of null and not null elements of λ . The results are in the form $f_{i,4} = \tilde{G}^R \tilde{S}_i \tau_{xyz,i}$ where $\tilde{G}$, $\tilde{S}$ are the matrices obtained by selecting the rows of G , S corresponding the non null elements of λ , and $\tilde{G}^R$ is the right inverse of $\tilde{G}$. The active set of this solution can be evaluated by substituting $f_{i,4}$ into the second and third equations in (14). For more details on this latter step please refer to [16, Sec. 4.1], where the solution of a problem in the form of (14) is derived for being used in explicit linear model predictive control.

Combining (10) and $f_{i,4}$ leads - through some simple algebraic derivations - to the piece-wise linear closed form solution of the problem (9)

$$f_i = I_a^j(\Delta_i) \tau_{xyz,i}, \quad \text{where } j \text{ s.t. } A^j(\Delta_i) \tau_{xyz,i} \leq 0, \quad (15)$$

with A^j and I_a^j defined in (21) and (22). The subscript $j \in \{0, 1, \dots, 4\}$ refers to the constraint which is active. Indeed, when $j = 0$ all four forces are used to generate actuation, i.e. only strictly negative forces are produced. It is worth noticing that in this case the mapping is very straightforward; a torque around x is produced through a pair of equal and opposite forces applied on y , and vice versa for a torque around y . The vertical force is equally distributed to all the four chambers. When an inequality with $j \neq 0$ holds, it means that the j -th constraint is active, and $f_{i,j} = 0$. This is clearly reflected by the null rows in (22).

Note that (15) also specifies the set of admissible $\tau_{xyz,i}$ as $\bigcup_{j \in \{0, 1, \dots, 4\}} \{\tau_{xyz,i} \text{ s.t. } A^j(\Delta_i) \tau_{xyz,i} \leq 0\}$. While unions of convex polyhedrons are not in general convex polyhedrons themselves, it can be proven that this holds true for our case by following the arguments in [16, Sec. 4.2]. We call $A_i \in \mathbb{R}^{3 \times 3}$ the matrix such that $A_i \tau_{xyz,i} \leq 0$ identifies the complete active set for the i -th segment.

Finally the map from τ_{xyz} to f is obtained by applying iteratively (15). This action is implemented by the block *sub-map 2* in Fig. 3.

D. Solutions with positivity constraints

In pressure driven systems the structure of the problem is the same, but with positive forces, $f_i \geq 0$. This produces only an inversion of the inequality sign, yielding

$$f_i = I_a^j(\Delta_i) \tau_{xyz,i}, \quad \text{where } j \text{ s.t. } A^j(\Delta_i) \tau_{xyz,i} \geq 0, \quad (16)$$

where I_a^j and A^j are defined as in (21) and (22).

V. CONTROLLER

As controller we use the PD with dynamics compensation that we proposed in [10], [11]

$$\begin{aligned} \tau = & K_P(q_{\text{ref}} - q) + K_D(\dot{q}_{\text{ref}} - \dot{q}) + K q_{\text{ref}} \\ & + D(q)\dot{q}_{\text{ref}} + B(q)\ddot{q}_{\text{ref}} + C(q, \dot{q})\dot{q}_{\text{ref}} + G(q). \end{aligned} \quad (17)$$

$K_P, K_D \in \mathbb{R}^{n \times n}$ are gain matrices. $q_{\text{ref}} \in \mathbb{R}^n$ is the desired state evolution, with its time derivatives $\dot{q}_{\text{ref}}, \ddot{q}_{\text{ref}}$. The remaining terms are as in (1). Please see [10] for a proof of global asymptotic stability in the unconstrained case. However, τ can be exerted only within certain limits, as specified by (1). Nonetheless, the asymptotic stability of the closed loop can still be proven for all the q_{ref} internal to the feasible set (see next section). Indeed, invoking continuity properties assures that a neighborhood of q_{ref} exists, such that all its elements satisfy the constraints. The region of asymptotic stability of (17) is always lower bounded by such neighborhood. This proves the local asymptotic stability of the equilibrium.

VI. PLANNING

A. Propagation and integration of the constraints

The set of configurations q_{ref} compatible with the actuation constraints $f \in \mathbb{I}_f$ can be described by propagating back these constraints, through the control allocation and the controller, as summarized by Fig. 3. The result is the following set of inequality constraints

$$A \alpha^{-1}(q)(Kq + G(q)) < b, \quad (18)$$

with $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$ defined as in Sec. IV-D. $\tau_{xyz}(q) = \alpha^{-1}(q)(Kq + G(q))$ is the steady state control action produced by controller (17) and *sub-map 1*.

Eq. (18) identifies a set with a complex geometry, which would be difficult to manage during on-line generation of trajectories. We solve this issue by considering as approximation a big enough interval set which is fully included in the actual feasible set. Contextually, we perform also the integration of (18) with the configuration constraints discussed in Sec. II-C. We formalize the problem as follows

$$\begin{aligned} \max_{q_{\text{lb}}, q_{\text{ub}}} & \quad \frac{1}{2} \|q_{\text{lb}} - q_{\text{ub}}\|_2^2 \\ \text{s.t.} & \quad \mathbb{I}(q_{\text{lb}}, q_{\text{ub}}) \subseteq \mathbb{I}(\bar{q}_{\text{lb}}, \bar{q}_{\text{ub}}) \\ & \quad h(\mathbb{I}(q_{\text{lb}}, q_{\text{ub}})) \in \mathbb{T}_i \quad \forall i \\ & \quad h_c(\mathbb{I}(q_{\text{lb}}, q_{\text{ub}})) \cap \{\mathbb{O} \times \dots \times \mathbb{O}\} = \emptyset \\ & \quad A \tau_{xyz}(\mathbb{I}(q_{\text{lb}}, q_{\text{ub}})) \leq b. \end{aligned} \quad (19)$$

where we use some basic interval analysis notation to describe sets in a more compact form; i) $\mathbb{I}(q_{\text{lb}}, q_{\text{ub}}) = \{q \text{ s.t. } q_{\text{lb}} \leq q \leq q_{\text{ub}}\}$, where the inequalities are to be intended element-wise; ii) given a function $F(\cdot)$, we call $F(\mathbb{I}) = \{F(x) \text{ s.t. } x \in \mathbb{I}\}$.

The first three constraints constitute the bounds introduced in Sec. II-C - i.e. staying into the configuration limits $\mathbb{I}(\bar{q}_{\text{lb}}, \bar{q}_{\text{ub}})$, reaching the targets $\mathbb{T}_i$, avoiding the obstacles $\mathbb{O}$. $h(\cdot)$ is the forward kinematics of the robot's tip. $h_c(\cdot)$ is the forward kinematics of a set of relevant points along the

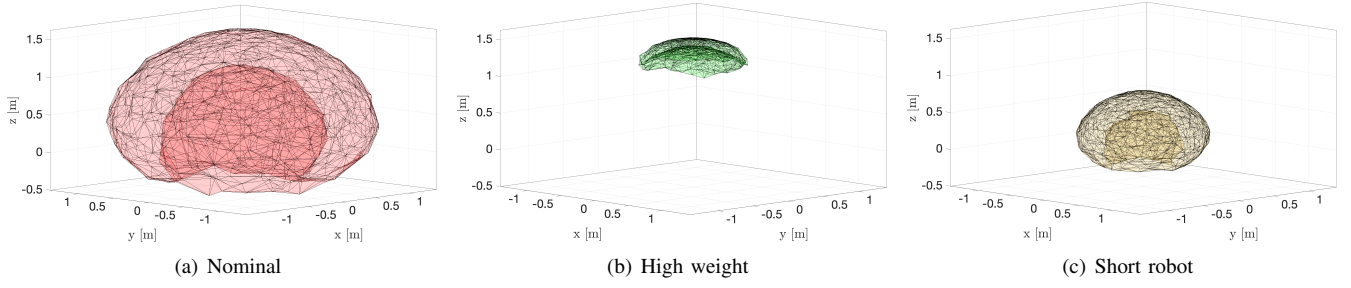


Fig. 6. Positions $h(\mathbb{I}_{\text{ref}})$ that the robot can reach with its tip while remaining in the feasible set $\mathbb{I}_{\text{ref}}$. Three soft robots with different physical characteristics are considered. Panel (a) shows the result for the nominal choice of physical parameters discussed in the text. Panel (b) reports the outcome when the weight of all the segments is increased of a factor ten. The robot can reach a much smaller area without failing to satisfy the constraints on actuation $f \leq 0$. Panel (c) shows the result for a robots with the same physical characteristics as the robot in panel (a), but with half the rest length.

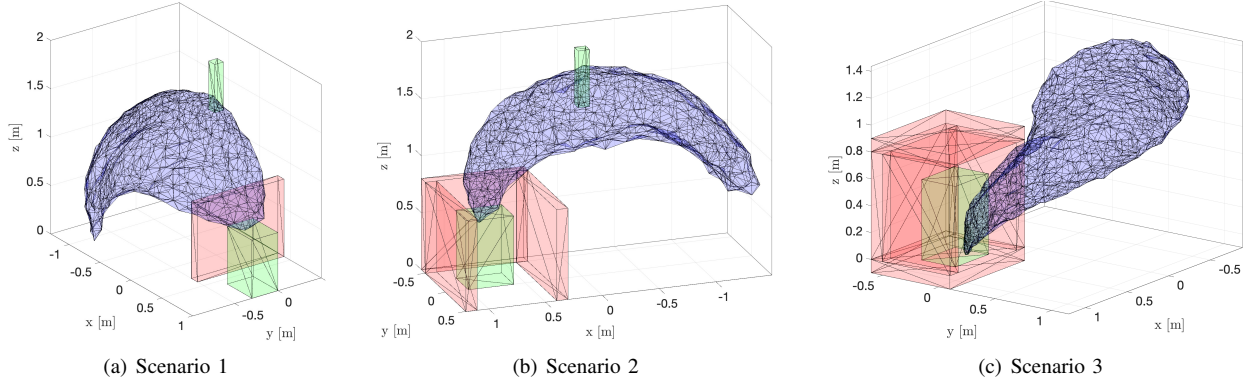


Fig. 7. The blue volumes show the positions $h(\mathbb{I}_{\text{ref}})$ that the robot can reach with its tip while remaining in the feasible set $\mathbb{I}_{\text{ref}}$. Three examples are shown for three different scenarios. The green rectangles are volumes that the robot must reach with its tip, i.e. $\mathbb{T}_i$ in (19). The red rectangles are areas that the robot body can not interact with, i.e. $\mathbb{O}$ in (19).

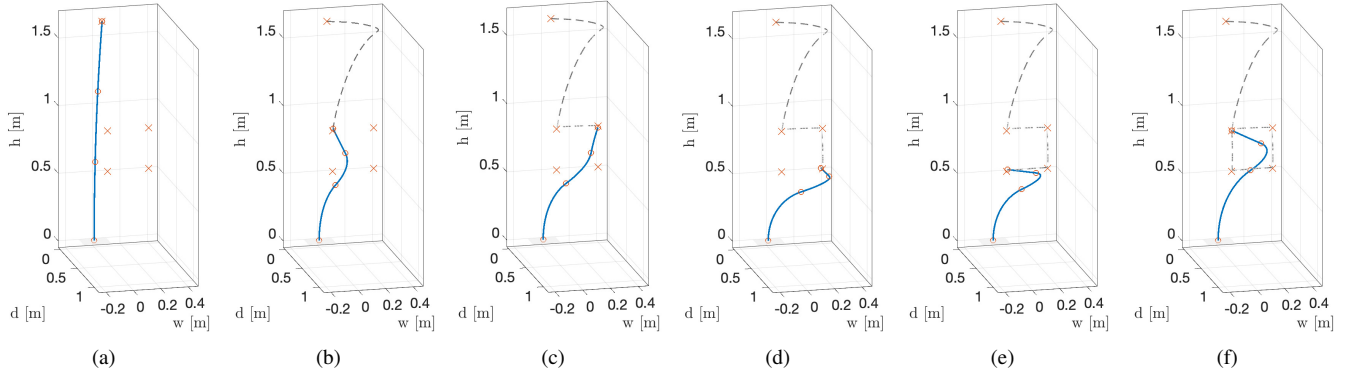


Fig. 8. Sequence showing the soft robot sequentially reaching for five targets (red crosses in figure). The robot has three actuated segments. The end of each segment is highlighted with a red circle. The trajectory followed by the robot tip is depicted as a gray dashed line. In all its motions the robot configuration q remains in the safe area $\mathbb{I}_{\text{ref}}$, where the actuators can sustain the robot despite the strict negativity of the inputs.

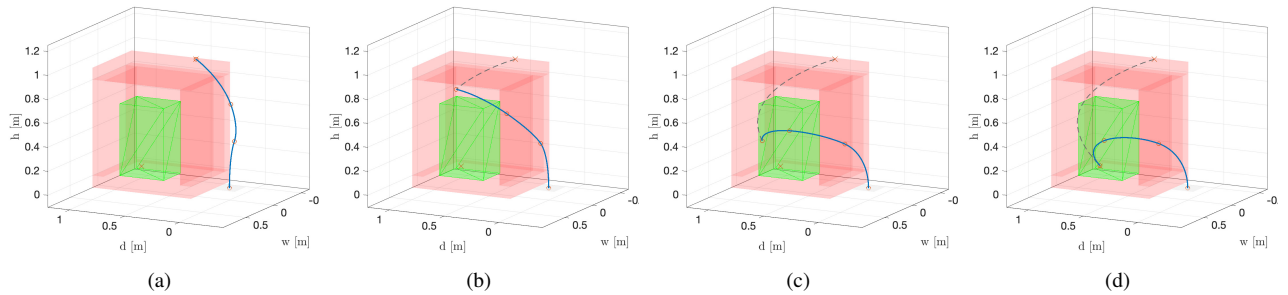


Fig. 9. Sequence showing the soft robot reaching a target (red cross in figure) placed inside an area of interest (green volume), while avoiding with all its body the obstacles surrounding the goal (red volumes). The robot has three actuated segments. The end of each segment is highlighted with a red circle. The trajectory followed by the robot tip is depicted as a gray dashed line. In all its movements the configuration q remains in the safe area $\mathbb{I}_{\text{ref}}$, where the actuators can sustain the robot despite the strict negativity of the inputs, and where no contact with the environment can occur.

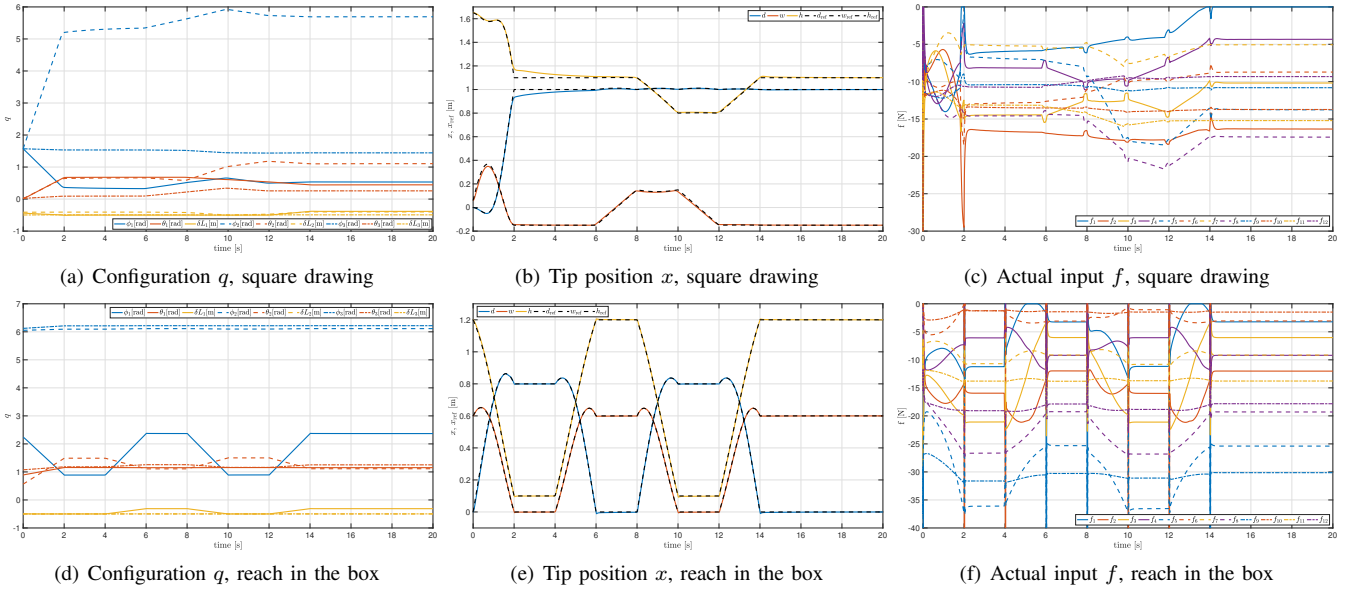


Fig. 10. Evolution in time of the main quantities describing the behavior of the soft robot and the control architecture during the execution of the *square drawing* task - panels (a,b,c) - and the *reach in the box* task - panels (d,e,f). Panels (b,e) account for the very precise tracking of the trajectory produced by the planner, and expressed here the end effector (desired positions depicted as black dashed lines). Panel (c,f) show that the actuation f is always strictly negative, fulfilling the constraints. The spikes in actuation in panel (f) are due to the high acceleration required to the system during the sudden slope changes.

Algorithm 1 Find large interval included in the feasible set.

```

1:  $q_{lb}^{opt} \leftarrow \bar{q}_{lb}$  ▷ Initialization
2:  $q_{ub}^{opt} \leftarrow \bar{q}_{ub}$ 
3: for  $i \leftarrow 1, N_{trials}$  do
4:    $\tilde{q}_{lb} \leftarrow \text{rand}(\bar{q}_{lb}, \bar{q}_{ub})$  ▷ Uniform random distribution
5:    $\tilde{q}_{ub} \leftarrow \text{rand}(\bar{q}_{lb}, \bar{q}_{ub})$ 
6:   if  $\|q_{lb}^{opt} - q_{ub}^{opt}\|_2^2 < \|\tilde{q}_{lb} - \tilde{q}_{ub}\|_2^2$  then
7:     Included  $\leftarrow \text{CheckInclusion}(\tilde{q}_{lb}, \tilde{q}_{ub})$ 
8:     if Included = True then
9:        $q_{lb}^{opt} \leftarrow \tilde{q}_{lb}$  ▷ New optimum found
10:       $q_{ub}^{opt} \leftarrow \tilde{q}_{ub}$ 

```

robot that we want not to be in contact with the environment. In this work we consider them to be the end and the middle of each segment.

To find a solution for (19) we propose algorithm 1, implementing a Monte Carlo approach. Fig. 5 shows an example of some core steps of the algorithm. It is worth underlying that, while potentially computationally onerous, this algorithm is executed off-line, and only once for a given robot and environment. Its output serves then as a compact representation of the constraints, that we use in the next subsection to implement a very efficient constrained kinematic inversion strategy.

B. Constrained inversion algorithm

Given a desired position of the tip x_{ref}^k , and the feasible interval set $\mathbb{I}_{ref}$, we evaluate the configuration $\bar{q}_{ref}^k$ such that $h(\bar{q}_{ref}^k) = x_{ref}^k$, with $h(\cdot)$ forward kinematics of the tip, as steady state of the following dynamical system

$$\dot{q}^k = J_{set}^+ e_{set} + (I - J_{set}^+ J_{set}) J^+ (x_{ref} - h(\bar{q}^k)). \quad (20)$$

J^+ is the Moore-Penrose pseudo-inverse of J . J_{set}, e_{set} are the Jacobian and the error of the constraints. J_{set} is obtained from the identity matrix by selecting all the i -th columns,

Algorithm 2 Check if the interval is included in the feasible set.

```

1: procedure CHECKINCLUSION( $\tilde{q}_{lb}, \tilde{q}_{ub}$ )
2:   Included  $\leftarrow$  True ▷ initialization
3:   DestinationReached $_j \leftarrow$  False
4:   for  $i \leftarrow 1, N_{check}$  do
5:     if Included = True then
6:        $q_{test} \leftarrow \text{rand}(\tilde{q}_{lb}, \tilde{q}_{ub})$  ▷ Random config.
7:        $\tau_{xyz} \leftarrow \alpha^{-1}(q_{test})(K q_{test} + G(q_{test}))$ 
8:       for  $j \leftarrow 1, N_T$  do
9:         if  $h(q_{test}) \in \mathbb{T}$  then
10:          DestinationReached $_j \leftarrow$  True
11:         if  $h_c(q_{test}) \in \{\mathbb{O} \times \dots \times \mathbb{O}\}$  then
12:           Included  $\leftarrow$  False
13:         if  $(A\tau_{xyz} < b) = \text{False}$  then
14:           Included  $\leftarrow$  False
15:       if  $\exists j$  s.t. DestinationReached $_j = \text{False}$  then
16:         Included  $\leftarrow$  False
17:   return Included

```

such that $\bar{q}_i^k$ does not satisfy the constraint specified by $\mathbb{I}_{ref}$. e_{set} is a column vector having as j -th element 1 if the variable violating the j -th violated constraint is below its lower bound, and -1 otherwise. Due to space limitations, we can not go deeper in the details of this algorithm, which was inspired by results provided in [17]. The convergence of (20) can be proven with arguments similar to the ones used in that paper.

Once $\bar{q}^k$ is calculated, the trajectory bringing from x_{ref}^{k-1} to x_{ref}^k in a time t_{fin} is $q_{ref}(t) = \frac{t_{fin}-t}{t_{fin}} \bar{q}^{k-1} + \frac{t}{t_{fin}} \bar{q}^k$ which is fully contained into $\mathbb{I}_{ref}$, since the set is connected by construction.

VII. SIMULATIONS

Consider a soft robot composed by three sequentially connected CC segments, as in Fig. 2(a). Each segment is 1m

$$A^0 = \begin{bmatrix} 0 & \frac{-1}{2\Delta_i} & \frac{1}{4} \\ 0 & \frac{1}{2\Delta_i} & \frac{1}{4} \\ \frac{1}{2\Delta_i} & 0 & \frac{1}{4} \\ \frac{-1}{2\Delta_i} & 0 & \frac{1}{4} \end{bmatrix}, A^1 = \begin{bmatrix} 0 & \frac{1}{\Delta_i} & 0 \\ \frac{1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ 0 & \frac{1}{\Delta_i} & -2 \end{bmatrix}, A^2 = \begin{bmatrix} 0 & \frac{-1}{\Delta_i} & 0 \\ \frac{1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ 0 & \frac{-1}{\Delta_i} & -2 \end{bmatrix}, A^3 = \begin{bmatrix} \frac{1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & 0 & 0 \\ \frac{-1}{\Delta_i} & 0 & -2 \end{bmatrix}, A^4 = \begin{bmatrix} \frac{-1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ \frac{1}{2\Delta_i} & 0 & 0 \\ \frac{1}{\Delta_i} & 0 & -2 \end{bmatrix}, \quad (21)$$

$$I_a^0 = \begin{bmatrix} 0 & \frac{-1}{2\Delta_i} & \frac{1}{4} \\ 0 & \frac{1}{2\Delta_i} & \frac{1}{4} \\ \frac{1}{2\Delta_i} & 0 & \frac{1}{4} \\ \frac{-1}{2\Delta_i} & 0 & \frac{1}{4} \end{bmatrix}, I_a^1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{\Delta_i} & 0 \\ \frac{1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \end{bmatrix}, I_a^2 = \begin{bmatrix} 0 & \frac{-1}{\Delta_i} & 0 \\ 0 & 0 & 0 \\ \frac{1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \end{bmatrix}, I_a^3 = \begin{bmatrix} \frac{1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ 0 & 0 & 0 \\ \frac{-1}{\Delta_i} & 0 & 0 \end{bmatrix}, I_a^4 = \begin{bmatrix} \frac{-1}{2\Delta_i} & \frac{-1}{2\Delta_i} & \frac{1}{2} \\ \frac{-1}{2\Delta_i} & \frac{1}{2\Delta_i} & \frac{1}{2} \\ \frac{1}{2\Delta_i} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (22)$$

long, and it weights 0.4Kg. Its bending rigidity is $0.1 \frac{\text{Nm}}{\text{rad}}$, and its compression stiffness is $0.1 \frac{\text{N}}{\text{mm}}$. Each segment hosts four actuators, as in Fig. 2(c). They simulate vacuum chambers, and thus they can produce only negative forces, $f \leq 0$. The axis of each chamber is $\Delta_i = 0.2\text{m}$ far from the central axis of the segment. All the segments share the same constraints on the configuration, $\phi_i \in [0, 2\pi]$, $\theta_i \in [0, 2\pi]$, $\delta L_i \in [-0.5, 0]\text{m}$. When $q = 0$ the robot is in a straight vertical configuration, with its tip pointing upwards. The gravity acceleration is the standard one, and points downwards.

As first step we evaluate $\mathbb{I}_{\text{ref}}$ through algorithm 1. We used $N_{\text{trials}} = 10^3$ and $N_{\text{check}} = 10^3$. The outcome of the algorithm was consistent across multiple executions. The algorithm has been implemented in MatLab 2018b, and its execution time on a standard laptop ranges from few seconds to ten minutes depending on the scenario. Note that this algorithm is intended to be executed off-line and just once for a given pair of robot and scenario. We report the results expressed as the volume of the operational space that can be reached by moving into the estimated set $\mathbb{I}_{\text{ref}}$, i.e. $h(\mathbb{I}_{\text{ref}})$. Fig. 6 shows the case of no obstacles, i.e. $\mathbb{O} = \emptyset$. The only target is the straight configuration, i.e. $\mathbb{T}_1 = \{q \text{ s.t. } \theta_i = 0, \forall i\}$. The three panels show the results for different choices of the physical parameters. Fig. 7 reports instead the outcomes for three different choices of $\mathbb{O}$ and $\mathbb{T}_i$. The first is highlighted as a red volume, the seconds as green ones.

We thus apply the control architecture to the execution of two tasks. The first one is the sequential acquisition of five targets with the tip of the robot, in a free space scenario (as for Fig. 6(a)). Targets are positioned so that the final tip trajectory is a square. Fig. 8 shows the resulting behavior. The second task is to go back and forth between two points, one in free space and one inside the area of interest contained in a box. The scenario is the same of Fig. 7(c). Results are shown in Fig. 9. Note that the robot must fully exploit its degrees of freedoms to bend around the obstacles and reach for the target. Fig. 10 shows the evolution in time of salient quantities for both tasks.

VIII. CONCLUSIONS AND FUTURE WORK

This paper proposed a control pipeline able to regulate the tip position of soft bodied robots, with unidirectional actuation connected to the state through a non linear characteristics, and constrains in configuration modeling the environment. Extensive simulations were provided to support the theoretical results. Future work will be devoted to experimentally validating the architecture on a real robot.

REFERENCES

- [1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, 2015.
- [2] T. G. Thuruthel, E. Falotico, M. Manti, and C. Laschi, "Stable open loop control of soft robotic manipulators," *IEEE Robotics and Automation Letters*, 2018.
- [3] F. Renda, F. Boyer, J. Dias, and L. Seneviratne, "Discrete cosserat approach for multisection soft manipulator dynamics," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1518–1533, 2018.
- [4] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *Soft robotics*, 2018.
- [5] S. H. Sadati, S. E. Naghibi, I. D. Walker, K. Althoefer, and T. Nanayakkara, "Control space reduction and real-time accurate modeling of continuum manipulators using ritz and ritz-galerkin methods," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 328–335, 2018.
- [6] G. Runge and A. Raatz, "A framework for the automated design and modelling of soft robotic systems," *CIRP Annals-Manufacturing Technology*, 2017.
- [7] B. Kim, J. Ha, F. C. Park, and P. E. Dupont, "Optimizing curvature sensor placement for fast, accurate shape sensing of continuum robots," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 5374–5379.
- [8] H. Wang, B. Yang, Y. Liu, W. Chen, X. Liang, and R. Pfeifer, "Visual servoing of soft robot manipulator in constrained environments with an adaptive controller," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 41–50, 2017.
- [9] C. Della Santina, R. K. Katzschmann, A. Bicchì, 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.
- [10] —, "Model-based dynamic feedback control of a planar soft robot: trajectory tracking and interaction with the environment," *Under review at International Journal of Robotics Research*, 2019.
- [11] R. K. Katzschmann, C. Della Santina, T. Yasunori, A. Bicchì, and D. Rus, "Dynamic motion control of multi-segment soft robots using piecewise constant curvature matched with an augmented rigid body model," in *2019 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019.
- [12] C. Della Santina, L. Pallottino, D. Rus, and A. Bicchì, "Exact task execution in highly under-actuated soft limbs: an operational space based approach," in *2019 IEEE International Conference on Soft Robotics (RoboSoft)*. IEEE, 2019.
- [13] C. Della Santina, M. Bianchi, G. Grioli, F. Angelini, M. Catalano, M. Garabini, and A. Bicchì, "Controlling soft robots: balancing feedback and feedforward elements," *IEEE Robotics & Automation Magazine*, vol. 24, no. 3, pp. 75–83, 2017.
- [14] M. A. Robertson and J. Paik, "New soft robots really suck: Vacuum-powered systems empower diverse capabilities," *Science Robotics*, vol. 2, no. 9, p. ean6357, 2017.
- [15] S. Li, D. M. Vogt, D. Rus, and R. J. Wood, "Fluid-driven origami-inspired artificial muscles," *Proceedings of the National Academy of Sciences*, vol. 114, no. 50, pp. 13 132–13 137, 2017.
- [16] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
- [17] N. Mansard, A. Remazeilles, and F. Chaumette, "Continuity of varying-feature-set control laws," *IEEE Transactions on Automatic Control*, vol. 54, no. 11, pp. 2493–2505, 2009.