An obstacle-interaction planning method for navigation of actuated vine robots

M. Selvaggio[1], L. A. Ramirez[2], N. D. Naclerio[3], B. Siciliano[1] and E. W. Hawkes[3]

***Abstract*— The field of soft robotics is grounded on the idea that, due to their inherent compliance, soft robots can safely interact with the environment. Thus, the development of effective planning and control pipelines for soft robots should incorporate reliable robot-environment interaction models. This strategy enables soft robots to effectively exploit contacts to autonomously navigate and accomplish tasks in the environment. However, for a class of soft robots, namely vine-inspired, tip-extending or "vine" robots, such interaction models and the resulting planning and control strategies do not exist. In this paper, we analyze the behavior of vine robots interacting with their environment and propose an obstacle-interaction model that characterizes the bending and wrinkling deformation induced by the environment. Starting from this, we devise a novel obstacle-interaction planning method for these robots. We show how obstacle interactions can be effectively leveraged to enlarge the set of reachable workspace for the robot tip, and verify our findings with both simulated and real experiments. Our work improves the capabilities of this new class of soft robot, helping to advance the field of soft robotics.**

I. INTRODUCTION

Thanks to their continuum bodies, soft robots can exhibit large-scale deformations and high compliance. Compared to their traditional rigid-bodied counterparts, soft robots bring benefits in constrained environment applications, requiring unavoidable interaction with the surroundings [1]. For example, the advantages of soft bodies have been shown for continuum manipulators [2], [3], [4], soft grippers [5], [6], [7], medical robots [8], [9], [10], [11], and mobile robots [12], [13]. Soft robots introduced a paradigm shift in the way obstacles are considered in robotic applications: interaction and exploitation of the environment may be advantageously used for navigation [14] and control [15], [16].

However, the interaction with the environment causes the robot to significantly change its shape, thus modifying and complicating its response to actuation commands. This must be accounted for to properly plan and execute effective movements with interacting soft robots. The above challenge has not yet been tackled for a new class of soft robot, vine-inspired, tip-extending, or "vine" robots [17], [18], [19]. When actuated, vine robots are capable of extension from the tip and steering with a variety of mechanisms that enables the realization of various geometrical shapes with the body [20], [21], [22]. This class of robots has been used to design proof-of-concept soft catheters for low-force interactions in constrained surgery [23], re-configurable and deployable antennas [24], and inspection devices deployed in archaeological sites in South America [25].

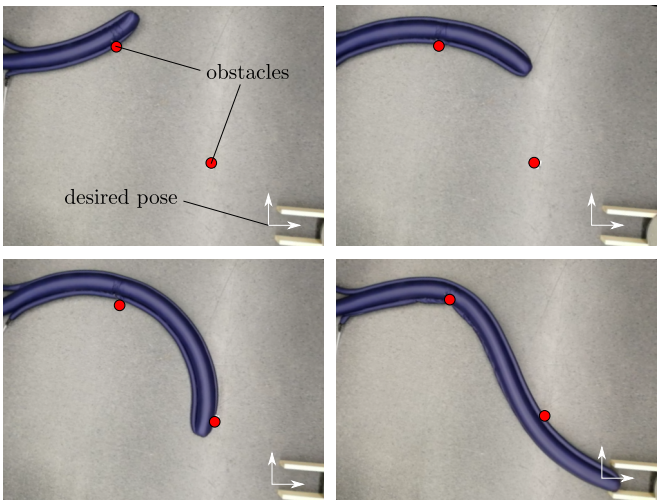


Fig. 1. Leveraging the proposed obstacle-interaction planner, a vine robot is steered toward a goal pose by means of pneumatic artificial muscles while exploiting environmental contacts with a set of obstacles.

To reliably accomplish tasks with vine robots, teleoperation was used in [26]; only very recently, interaction modeling and motion planning algorithms have started to be devised: in [14] the authors used obstacles interaction inside a cluttered environment to perform navigation with a non-steerable robot while in [27] they added a passive turning mechanisms. However, these models did not consider environmental forces and did not incorporate active steering.

To fill this gap in the literature, the main contribution of this paper is the development and the experimental verification of an obstacle interaction model that incorporates environmental forces and its use for interaction planning. As such, we extend the methods presented in [14], [27] by replacing the lumped parameter model with a more accurate deformation model of the robot and including the active steering capabilities, introduced in [22], to develop a novel planner for actuated vine robots (Fig. 1). Additionally, we analytically characterize the vine robot reachable workspace and propose algorithms for efficient implementation.

This paper is organized as follows. In Section II, we present the kinematic and interaction models as well as the

[1]Department of Electrical Engineering and Information Technology, University of Naples Federico II, 80125 Naples, Italy.

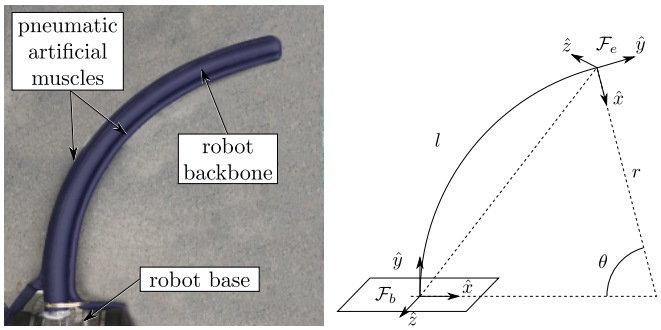
[2]Department of Mechanical Engineering, University of California Irvine, CA 92697, USA.

[3]Department of Mechanical Engineering, University of California Santa Barbara, CA 93106, USA.

Corresponding author e-mail: mario.selvaggio@unina.it

The stay of M. Selvaggio was financially supported by UniNA and Compagnia di San Paolo, in the frame of Programme STAR.

This work was supported by the NSF (grant no. 1637446), a NASA Space Technology Research Fellowship and by the PON RI 2014-2020 ICOSAF.



(a) (b)

Fig. 2. (2a) Planar vine-robot steered in free space by two Pneumatic Artificial Muscles (PAMs) which create a constant curvature deformation. (2b) Kinematic representation of the robot. Two local frames, $\mathcal{F}_b$ and $\mathcal{F}_e$, are placed at the two ends of the robot backbone. The robot length is l, while r represents its curvature radius.

workspace analysis. In Section III, we describe our novel planning method. Section IV shows the experimental setup and achieved results while Section V concludes the paper.

II. MODELING

In this section, we briefly describe our vine robot, develop a description of the kinematics (Sect. II-A), the obstacle interaction model (Sect. II-B), and analyze the reachable workspace (Sect. II-C).

The vine robot used in this work is composed of an inflated fabric backbone and two Pneumatic Artificial Muscles (PAMs) glued lengthwise along the backbone (Fig. 2a). While two PAMs are used here for planar motion, three PAMs could be used for full 3-D movement. The underlying working mechanism is largely described in [17], [22], [28]. When pressurized, the backbone can lengthen at its tip by everting new material that is unspooled from a reel at the base. The lengthening occurs in the direction that the tip is pointing. Independently, the PAMs can be made to contract. This causes the whole robot to bend reversibly, and the position and orientation of the tip to change, accordingly. Because of the axially uniform nature of the robot, it is reasonable to assume a constant contraction of the PAMs along the entire length in free space. Assuming negligible frictional loss and neglecting the gravitational energy, this results in a nearly Constant Curvature Deformation (CCD) of the backbone.

A. Kinematics

The above-introduced CCD assumption allows reducing the dimension of the configuration space from infinite to finite. In this case, the robot kinematics can be expressed using two concatenate mappings [29], [30], i.e., (i) the mapping from actuator space (air pressure) to configuration space (length and curvature), and (ii) the mapping between the configuration space and the task space (tip position and orientation). In this section, we focus on the latter. The planar robot configuration is fully described through the vector of arc parameters $\boldsymbol{q} = [\kappa, l]^{\mathrm{T}}$, where $\kappa = 1/r \in \mathbb{R}$ denotes the robot curvature and $l \in \mathbb{R}^+$ its current length (Fig. 2b). We denote by $\mathcal{F}_b = \{O_b; \hat{x}, \hat{y}, \hat{z}\}$ and $\mathcal{F}_e = \{O_e; \hat{x}, \hat{y}, \hat{z}\}$ right-handed reference frames attached to the robot base and tip, respectively. The pose of $\mathcal{F}_e$ can be expressed in $\mathcal{F}_b$ through the homogeneous matrix[1] ${}^b\boldsymbol{T}_e({}^b\boldsymbol{R}_e, {}^b\boldsymbol{t}_e) \in \mathrm{SE}(2)$

$$ {}^b\boldsymbol{T}_e\left({}^b\boldsymbol{R}_e, {}^b\boldsymbol{t}_e\right) = \boldsymbol{T}\left(\boldsymbol{R}_{\hat{z}}\left(-\theta/2\right), \boldsymbol{0}\right) \boldsymbol{T}\left(\boldsymbol{I}, \boldsymbol{t}\right) \boldsymbol{T}\left(\boldsymbol{R}_{\hat{z}}\left(-\theta/2\right), \boldsymbol{0}\right), \tag{1} $$

where $\boldsymbol{R}_{\hat{z}}(\cdot) \in \mathrm{SO}(2)$ denotes the rotation matrix around the $\hat{z}$ axis, $\boldsymbol{t} = [0, 2l/\theta \sin(\theta/2)]^{\mathrm{T}} \in \mathbb{R}^2$ and $\boldsymbol{I} \in \mathbb{R}^{2\times 2}$ denotes the identity matrix (Fig. 2b). Given any point in the configuration space, one can calculate the corresponding task space position of the tip ${}^b\boldsymbol{t}_e = [x, y]^{\mathrm{T}} \in \mathbb{R}^2$ through the forward kinematics mapping (1), where x and y are coordinates of the robot tip in the base reference frame. The tip orientation $\theta \in [-\pi, \pi]$ can be trivially calculated considering the relation between the arc parameters $l = r\theta$.

As for the inverse kinematics, we found the following closed-form relationship exploiting the CCD assumption

$$ \boldsymbol{q} = \mathrm{IK}\left({}^b\boldsymbol{t}_e\right) = \begin{bmatrix} 2y/(x^2+y^2) \\ (x^2+y^2)(n\pi + \mathrm{atan}(y,x))/y \end{bmatrix}. \tag{2} $$

While infinite solutions exist to the IK problem (corresponding to periodic lengths, parametrized by $n \in \mathbb{Z}^+$), choosing $n = 0$ retrieves the shortest length solution. Moreover, singularities occur for $x^2 + y^2 = 0$ and $y = 0$. The former condition corresponds to the zero length solution $(l = 0)$, while the latter to the zero curvature $(r = \infty)$ of the straight configuration.

B. Obstacle interaction model

When a soft robot interacts with the environment, the interaction force modifies the robot shape. Accurate (non-linear) models, accounting for such deformation, include Cosserat rod theory and the finite-element method [31], [32]. However, the complexity and the computational burden of these methods have limited their use in real-time planning and control strategies for soft robots.

In this work, we adopt the analytic solutions for a loaded cantilever beam derived through Cosserat rod theory under small displacements in the SE(2) assumption [33], [34]. Denoting by s the material abscissa which parametrizes a straight beam (Fig. 3a), by a the distance along s at which the contact occurs, by ${}^b\boldsymbol{t}(s)$ and $\theta(s)$ the generic position and orientation of the cross section at s, we can express the mapping $\forall s \in \mathbb{R}^+ \mapsto {}^b\boldsymbol{T}(s)\left({}^b\boldsymbol{R}(\theta(s)), {}^b\boldsymbol{t}(s)\right)$ as

$$ {}^b\boldsymbol{t}(s) = \begin{bmatrix} s \\ \dfrac{fs^2(3a-s)}{6EI} \end{bmatrix}, \quad \boldsymbol{\theta}(s) = \begin{bmatrix} \dfrac{fs(2a-s)}{2EI} \end{bmatrix}, \quad s \le a, \tag{3} $$

$$ {}^b\boldsymbol{t}(s) = \begin{bmatrix} s \\ \dfrac{fa^2(3s-a)}{6EI} \end{bmatrix}, \quad \boldsymbol{\theta}(s) = \begin{bmatrix} \dfrac{fa^2}{2EI} \end{bmatrix}, \quad s > a, \tag{4} $$

where EI is the flexural rigidity while f denotes the intensity of the interaction force applied to the robot orthogonal to its centerline. f can be computed by resorting to the geometric model of the robot-obstacle interaction. In this model, the

[1] Here we use ${}^a\boldsymbol{T}_b({}^a\boldsymbol{R}_b, {}^a\boldsymbol{t}_b)$ to express the pose of b in a where ${}^a\boldsymbol{R}_b \in \mathrm{SO}(2)$ denotes the rotation matrix and ${}^a\boldsymbol{t}_b \in \mathbb{R}^2$ the translation vector.

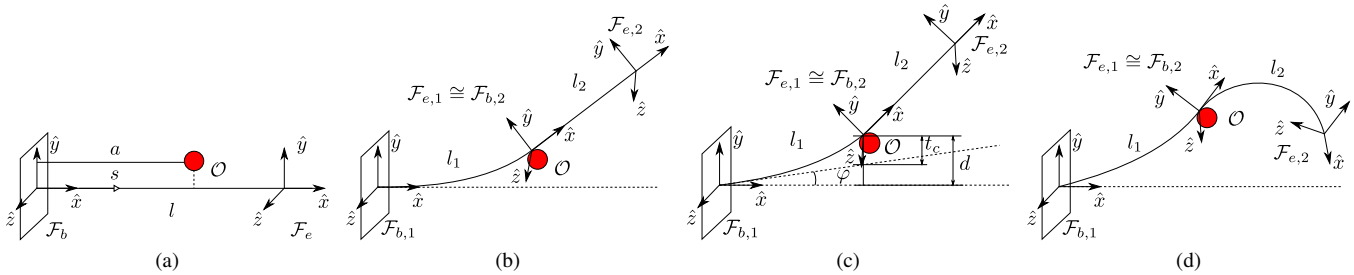


Fig. 3. Vine robot obstacle interaction model. (3a) Undeformed robot configuration; (3b) the robot interacts with the obstacle $\mathcal{O}$ and it is split into two sections: one having length l_1 which deforms according to a point-loaded cantilever beam model, and one having length l_2 exhibiting constant rotation. (3c) If the deflection caused by the obstacle interaction is greater the maximum deflection the robot wrinkles about the base; (3d) when actuated the unconstrained part continues to behave according to the CCD model.

robot can be decomposed into two parts r_1 and r_2: r_1 is the portion at $s \leq a$, with length l_1 and constitutes the part undergoing deformations induced by the interaction; r_2 is the portion at $s > a$ between the contact point and robot tip, with length l_2, exhibiting constant rotation of the cross-section (Fig. 3b). We assume that, when deformed, the former portion of the robot is fully described by the deformation model (3)-(4), while the latter part can be still freely actuated, thus behaving according to the CCD model (Fig. 3d) [35]. This assumption is introduced to simplify the reachable workspace calculation and the solution to the planning problem described later in Sect. III.

The interaction force f produces an internal moment m in the robot body. For thin-walled pressurized beams, there exists a threshold value for the internal moment m_c after which a roughly constant moment bends the robot through an arbitrary angle by wrinkling the beam at the point of highest moment (the base for a point-loaded cantilever beam) [36]; m_c is function of the radius, the area moment of inertia and the internal pressure of the beam. When interacting with an obstacle, if the moment about the base (m) is lower than the critical one (m_c) the robot bends according to (3)-(4), while if the moment is higher, it also wrinkles (Fig. 3c).

The geometric reasoning applied to calculate the bending/wrinkling amount is as follows: the contact point along the robot curvilinear abscissa a can be retrieved by projecting the obstacle on the undeformed robot; for each a there is a unique critical force causing wrinkling at the base $f_c = m_c/a$. Denoting by ${}^b\boldsymbol{t}^0(a)$ and ${}^b\boldsymbol{t}^c(a)$ the position of the cross section in the undeformed and the critically deformed configurations, respectively, ${}^b\boldsymbol{t}_o$ the position of the obstacle, we can calculate $t_c(a) = ||{}^b\boldsymbol{t}^c(a) - {}^b\boldsymbol{t}^0(a)||$, i.e. the amount of deflection induced by f_c at $s = a$, and $d = ||{}^b\boldsymbol{t}_o - {}^b\boldsymbol{t}^0(a)||$, i.e. the displacement the robot should undergo at a to steer around the obstacle. If $d \leq t_c$, the robot only bends when interacting with the obstacle and the value of f is calculated by inverting (3) substituting ${}^b\boldsymbol{t}(s = a) = {}^b\boldsymbol{t}_o$; otherwise t_c is considered as bending deflection produced by f_c and the wrinkling rotation is calculated as $\varphi = \arctan((d - t_c)/a)$. The forward kinematics model is then updated accordingly by pre-multiplying the ${}^b\boldsymbol{T}_e$ in (1) by a matrix ${}^0\boldsymbol{T}_b\left(\boldsymbol{R}_z\left(\varphi\right), \boldsymbol{0}\right)$ expressing the base frame pose into the inertial frame.

We note that the deformation model given in (3) and (4) is valid when the undeformed robot is in the straight configuration. However, a vine robot can interact with an obstacle when already curved by the PAMs actuation. In this case, each point in the curved configuration can be uniquely mapped through a rigid transformation to its corresponding pose in the straight configuration and vice versa. Thus strain-producing deformations can be computed at the local level in the straight configuration and mapped back to the curved one. At the kinematic level, this method is known in literature as co-rotational approach [37].

C. Workspace analysis

The tip of a PAMs-actuated vine robot moving into a planar Cartesian space can reach a set of poses given by the physical limits of the length and curvature (Fig. 4). Mathematically, the reachable workspace $\mathcal{W} \subset \mathbb{R}^2$ when no obstacles are present can be described by the set

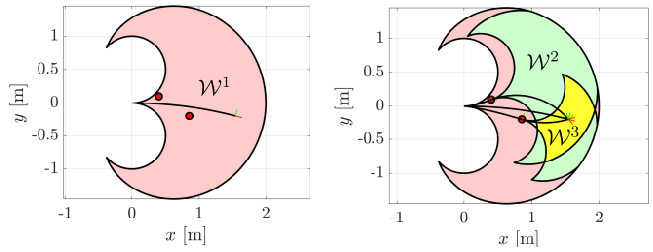
$$\mathcal{W} = \{\boldsymbol{t} = FK(\boldsymbol{q}) : \boldsymbol{q} \in [\boldsymbol{q}^-, \boldsymbol{q}^+]\}, \tag{5}$$

where $\boldsymbol{q}^-$ and $\boldsymbol{q}^+$ are vectors of configuration space lower and upper limits, respectively, and $FK(\cdot)$ is the analytical forward kinematics function for the position (derived from (1)). Obviously, $\mathcal{W}$ is not dexterous since every position $\boldsymbol{t} \in \mathcal{W}$ can be reached only with a unique orientation $\theta\left(\boldsymbol{t}\right)$.

Next, let us consider the presence of a set of obstacles $\{\mathcal{O}_1, \ldots, \mathcal{O}_n\}$ fixed to the workspace. The workspace representation of obstacles is the set

$$\mathbb{O} = \bigcup_{i=1}^{n} \mathcal{O}_i, \quad \mathcal{O}_i = \{\boldsymbol{t}(s) : \boldsymbol{t}(s) \cap \mathcal{O}_i \neq \varnothing \ \forall \ s\}, \tag{6}$$

i.e., the union of workspace regions in which the robot body is forbidden to enter. In this work, we consider cylindrical obstacles with radius $r_o << r$, and we neglect the robot body radius such that we can assume that the contact area reduces to a point at the mid-line of the backbone. By interacting with the i-th (sub)set of obstacles, the robot can change its shape, and thus reach a new set of positions $\mathcal{W}_i$ with a unique orientation (possibly different from the previous). In view of the assumption made in Sect. II-B, $\mathcal{W}_i$ can be easily calculated by applying the definition (5) to the last unconstrained portion of the robot (recalculating physical limits after the interaction occurs).



(a) (b)

Fig. 4. Workspaces with different number of possible orientations for vine robots with length $l \in [0, 2]$ m, curvature $\kappa \in [-2, 2]$ 1/m and two obstacles (red circles) placed at $[0.4, 0.1]^{\mathrm{T}}$ and $[0.85, -0.2]^{\mathrm{T}}$, respectively. (4a) Zero obstacle interaction case: one orientation is achievable for each position in the set $\mathcal{W}^1$. (4b) One obstacle interaction cases: $\mathcal{W}^2$ and $\mathcal{W}^3$ denote the sets of poses reachable with two and three orientations, respectively.

By opportunely considering the union and intersections between $\mathcal{W}_i$ and $\mathcal{W}$, one can recursively calculate workspaces with increasing number of possible orientations p. Denoting by $\mathcal{W}^p$ the workspace with number of possible orientations equal to p, the recursion writes as follows

$$\begin{aligned}
&\mathcal{W}^1 = \mathcal{W} \setminus \mathcal{O} \\
&\mathcal{W}^2 = \bigcup_{i=1}^{o} \mathcal{W}_i^2 \qquad\qquad \mathcal{W}_i^2 = \mathcal{W}_i \cap \mathcal{W}^1, \\
&\mathcal{W}^3 = \bigcup_{i=1}^{o} \mathcal{W}_i^3 \qquad \mathcal{W}_i^3 = \bigcup_{j=i+1}^{o} \mathcal{W}_i^2 \cap \mathcal{W}_j^2, \\
&\quad\vdots \qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
&\mathcal{W}^o = \bigcup_{i=1}^{o} \mathcal{W}_i^o \quad \mathcal{W}_i^o = \bigcup_{j=i+1}^{o} \mathcal{W}_i^{o-1} \cap \mathcal{W}_j^{o-1},
\end{aligned} \tag{7}$$

where o denotes the number of sequences in which it is possible to exploit the (sub)set of obstacles (Fig. 4 contains the workspaces representation up to $\mathcal{W}^3$).

To calculate the maximum number of possible orientations $o = \max(p)$ for a given pose $\boldsymbol{t} \in \mathcal{W}^o$ and n obstacles, we begin with zero, one, and two obstacles (Fig. 5a–5c), then construct the abstract graph representation of the obstacles, i.e., a complete graph with n nodes (Fig. 5d), and consider the number of possible paths with no node repetitions given by

$$o = 1 + \sum_{k=1}^{n} 2^k \frac{n!}{(n-k)!} \tag{8}$$

where 1 denotes the *base* solution (position reachable without interacting with any obstacle), 2^k represents the fact the robot can steer around an obstacle in a clockwise or counterclockwise direction, the fraction gives the number of possible paths visiting k nodes (obstacles), the summation accounts for paths consisting of any number of segments between 1 and n [38].

Obviously, not all the possible orientations are realizable given the physical robot limits, such as total robot length, minimum achievable curvature, and self-intersections. In the following section, we devise a method to calculate obstacle-interaction solutions and check their feasibility.

III. PLANNING

Generally, given a desired pose to accomplish a task and the current state of the robot, the planning problem reduces to finding a free set of poses that brings the robot from the initial to the final configuration (if such a set exists). This set of poses is traditionally found such that the poses avoid any obstacle in the workspace [39]. In this paper, we are instead interested in finding the path, intended as the sequence and the number of obstacles to be exploited, that navigates the robot to the desired position with minimal orientation error (as described in Sect. II-C, only discrete orientations are achievable at the desired position). To accomplish this, we devise a planner that generates paths leveraging obstacle interactions for navigating the soft growing robot to its destination. Given a desired pose $\boldsymbol{T}_d \in \mathrm{SE}(2)$, the proposed planner solves for all the possible paths that brings the robot tip to $\boldsymbol{t}_d \subset \mathbb{R}^2$ and returns the path with minimal orientation error in that point. The planner is guaranteed to return an optimal solution whenever a solution exists.

As explained in Sect. II-C, there are o ways to reach $\boldsymbol{t}_d \in \mathcal{W}^o$. The base case is simply calculated resorting to the inverse kinematics routine (2). The other solutions are generated by encoding the permutations of obstacles as a matrix $\boldsymbol{P}$ in which each row represents a different sequence of n obstacles. The solutions involving $k < n$ obstacles are calculated by storing the solutions corresponding to k columns of the matrix row P_i according to a dynamic programming approach. For each obstacle j in the sequence P_i, given the current robot state, we compute the solution of the interaction as explained in Sect. II-B. Thanks to the assumption made in Sect. II, the solution at j is not influenced by the one computed in the previous $j-1$ interaction steps. In favor of speed, we avoid calculating self-intersecting paths at the obstacle j by applying the following reasoning: the steering modality around the obstacle j is uniquely determined by the successive target position (desired position or next obstacle $j+1$). This is simply done by calculating the inverse kinematic solutions for reaching the obstacle j and $j+1$ (or the target position) and comparing the corresponding curvatures (Fig. 6).

At this point, from the obstacle j we compute the inverse kinematics to the target, check path feasibility and store it in a look up table. When the next permutation P_{i+1} is considered, all the partial solutions computed up to P_i are possibly reused. Once all the solutions are computed, the planner ranks them evaluating the cost $\|\theta_d - \theta\| \,(\mathrm{mod} 2\pi)$ and returns the optimal solution. Algorithms 1 and 2 contain the described procedures.

IV. EXPERIMENTS AND RESULTS

In this section, we briefly describe the experimental setup used to validate our model/algorithms (Sect. IV-A) and present the experimental results achieved (Sect. IV-B).

A. Experimental methods

To validate our findings and algorithms we carried out experiments using a self-contained muscle-actuated vine-

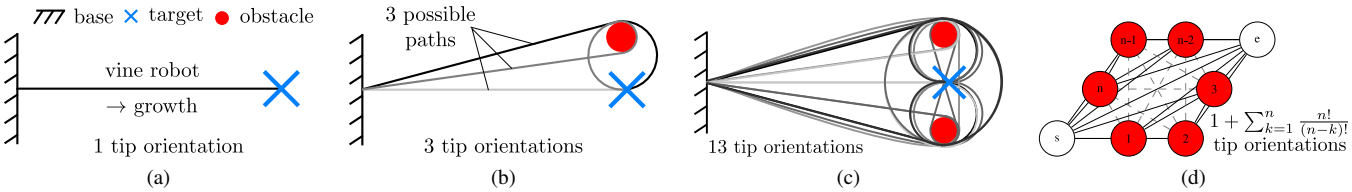


Fig. 5. Illustration of the possible paths realizable by the robot given an increasing number of obstacles n. (5a) The robot reaches the desired target with the unique possible orientation ($n = 0$); (5b) 3 possible paths to the target are sketched given 1 obstacle ($n = 1$); (5c) 13 possible paths to the target are sketched given two obstacles ($n = 2$); (5d) contains the graph abstracting the generic n obstacles case. The number of paths is given by (8).

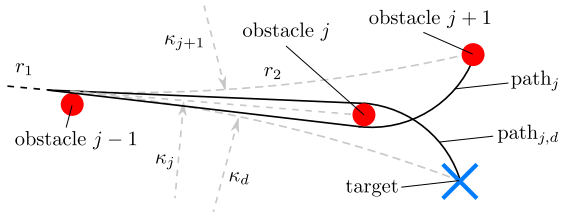


Fig. 6. Excluding self-intersecting paths at the j-th obstacle involves calculating the curvature to reach the j-th and the $j+1$-th obstacle (or the target) from $j-1$: if $\kappa_{j+1} > \kappa_j$ the robot steers around the obstacle j in the counterclockwise direction (path$_j$). For each obstacle j the solution to reach the target is also computed (path$_{j,d}$).

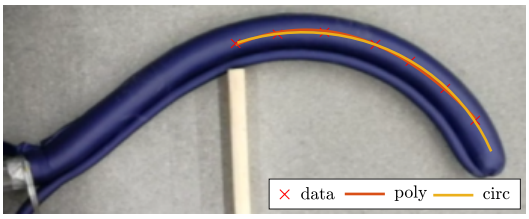


Fig. 7. Experimental validation of the CCD assumption: the shape of the robot's free section is accurately described by a circular arc.

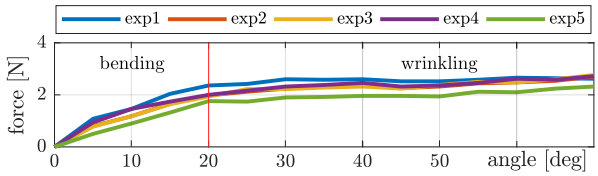


Fig. 8. Example force-displacement test for flexural rigidity evaluation. Bending/wrinkling areas are separated by the red line. Beam flexural rigidity was calculated from the derivative of the mean curve in the bending area.

robot (Fig. 2a). This is composed of a 0.064 m diameter, 2 m long robot backbone equipped with two 0.024 m diameter radially attached muscles. The robot body is wrapped on a reel in the base station that controllably spools out the material to allow growth from the robot tip. The air pressure in the backbone, which causes growth, and in the two radially attached muscles, which causes bending, is manually controlled by means of three pressure regulators.

The procedure to identify the vine robot flexural rigidity consisted in incrementally applying known displacements to the robot tip while measuring the applied force through a force sensor (MARK-10 Series 3). The wrinkling point was evaluated as the point at which the applied force started flattening. The critical moment was calculated as the moment after wrinkling occurred (Fig. (8)).

All the methods introduced in Sect. II and III were implemented in MATLAB. Reachability of the target points was tested at each calculation step by encoding the workspaces as a polyshape and using the in-built MATLAB methods.

The experimental setup for testing navigation is shown in Fig. 1 and in Fig. 9. It contains two obstacles placed at $\mathcal{O}_{1,t} = [0.7, 0.2]^{\mathrm{T}}\ m$ and $\mathcal{O}_{2,t} = [1.2, -0.4]^{\mathrm{T}}\ m$ with respect to the robot base frame. The obstacles are steel shafts with diameter equal to 0.005 m taped to the ground through a $0.04 \times 0.04 \times 0.005$ m acrylic base. The target location is at $\boldsymbol{t}_d = [1.7, -0.7]^{\mathrm{T}}$ m and the desired orientation is chosen as $\theta_d = 0$ rad.

B. Results and discussion

In this section, we present the results of experiments carried out to characterize our vine robot, validate the kinematic and deformation models, and test the planning method developed in this work.

Algorithm 1: Compute all feasible paths given a set of obstacles and return the optimal path

```
path = optimalPath(T_d)
    Data: obstacle set 𝕆, robot limits q⁺, q⁻
    Result: optimal path
    base ← computeIK(t_d);
    P ← permutations(𝕆);
    for (every row P_i in P) do
        for (every obj j in P_i) do
            if (!solutionExists(j,P_i))
                path ← plan(t_{j−1}, t_j, t_d);
                if (isFeasible(path, q⁺, q⁻))
                    paths = storeSolution(path)
    path = argMinOrientationError(T_d, paths, base)
```

Algorithm 2: Calculate paths though obstacle P_j (Fig. 6)

```
path = plan(t_{j−1}, t_j, t_d)
    Result: paths through j: path_j, path_{j,d}
    κ_j, κ_{j+1}, κ_d ← computeCurvatures(t_j, t_{j+1}, t_d);
    mode ← compare(κ_j, κ_d);
    path_j ← computeInteraction(P_j, mode);
    path_{j,d} ← computeIK(t_d);
    mode ← compare(κ_j, κ_{j+1});
    path_j ← computeInteraction(P_j, mode);
    r_1 ← updateRobot(path_j)
```

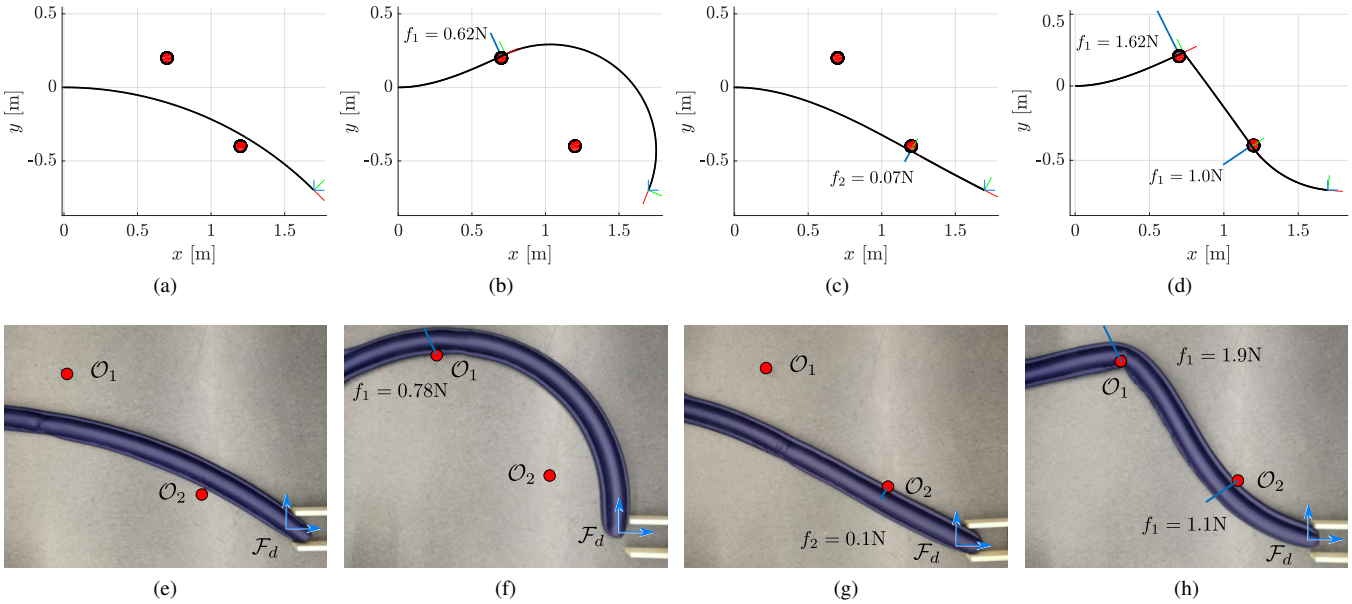


Fig. 9. Paths realizable by the vine robot given a set of two obstacles $\mathbb{O} = \{\mathcal{O}_1, \mathcal{O}_2\}$ and a desired pose $\mathcal{F}_d$. (9e),(9a) Base solution (inverse kinematics); (9f),(9b) one obstacle interaction ($\mathcal{O}_1$); (9g),(9c) one obstacle interaction ($\mathcal{O}_2$); (9h),(9d) two obstacles interaction.

The CCD in free space assumption made in Sect. II-B was validated by evaluating the error between a best-fit (least-square) 7 order polynomial curve and a circular arc fitting points of the backbone in a calibrated image setup (see Fig. 7). The mean error was $e_m = 0.024$ m while its standard deviation was $e_s = 0.032$ m on a $r \approx 0.6$ m radius of curvature robot. This result validates our assumption.

The flexural rigidity value (EI) was evaluated from the force-displacement test through inversion of (4). A plot showing the results from a series of force-displacement tests for a 6.89 kPa inflated vine robot backbone is given in Fig. 8. The mean value between 0 and 20 deg (wrinkling point) of the EI parameter was adopted in the following planning experiments, i.e., $EI = 0.1$ Nm^2. The critical moment attained a value of $m_c = 0.32$ Nm. The accuracy of the static identification procedure is critical to obtain reliable planning results. In our experiments, we noticed that the static friction between the robot body and the ground greatly influences the measurements and thus must be minimized. However, better static and dynamic characterization of vine robots would be beneficial for future work.

In the environment for testing navigation (Sect. IV-A), the planning algorithm found the four feasible plans depicted in Fig. 9. There is one solution with no obstacle interactions (Fig. 9a), two with one obstacle interaction (Fig. 9b and Fig. 9c), and one with the two obstacles interaction (Fig. 9d). The solutions were ranked evaluating the orientation error cost proposed in Sect. III. The minimum orientation error was found for the two obstacles interaction case: only thanks to the simultaneous interaction with two obstacles, the robot can assume a complex shape which allows it to accomplish the reaching task successfully. This strengthens the motivation behind this work. The planned paths were then validated using the real vine robot: as it is possible to see in Fig. 9e–9h, the planned paths are able to correctly predict the robot shape. During the tests, interaction forces between the vine robot and the obstacles were also measured and compared to the planned ones. The planner is able to accurately predict interaction forces, the discrepancy to be attributed to static friction and stiffening phenomenon caused by PAMs inflation.

Finally, we briefly discuss our results. As it is possible to notice, the assumption made in Sect. II-B influences the accuracy of the planned paths: when the free portion of the robot is steered, the whole robot changes its shape. This causes the robot cross section at the interaction to rotate hinged at the contact point. Moreover, the variation of air pressure in the muscle actuators changes the robot's flexural rigidity and possibly transforms wrinkling into bending or vice versa. However, removing our assumptions will require solving the deformation, and thus the planning problem, iteratively, thus dramatically slowing down the solution search.

V. CONCLUSION

In this work, we presented an obstacle-interaction planning method suitable for navigation of soft-growing robots within cluttered environments. First, the robot kinematic and the static deformation models including wrinkling were derived. Then, these models were used to characterize how the vine robot behaves when interacting with obstacles in the workspace. The interaction with obstacles was shown to enlarge the robot workspace, allowing it to reach a set of positions with multiple orientations. Finally a planning algorithm was devised which finds the sequence of obstacles that brings the vine robot to a desired position with a minimal orientation error. Experiments performed in the real environment are in accordance with the theoretical findings and provides us with guidelines for future works.

References

[1] D. Rus and M. T. Tolley, "Design, fabrication and control of soft robots," *Nature*, vol. 521, no. 7553, pp. 467–475, May 2015.

[2] W. McMahan, B. A. Jones, and I. D. Walker, "Design and implementation of a multi-section continuum robot: Air-octor," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Aug 2005, pp. 2578–2585.

[3] S. Neppalli, B. Jones, W. McMahan, V. Chitrakaran, I. Walker, M. Pritts, M. Csencsits, C. Rahn, and M. Grissom, "Octarm - a soft robotic manipulator," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Oct 2007, pp. 2569–2569.

[4] J. Li, Z. Teng, J. Xiao, A. Kapadia, A. Bartow, and I. Walker, "Autonomous continuum grasping," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Nov 2013, pp. 4569–4576.

[5] R. K. Katzschmann, A. D. Marchese, and D. Rus, "Autonomous object manipulation using a soft planar grasping manipulator," *So Ro*, vol. 2, no. 4, pp. 155–164, 2015.

[6] K. C. Galloway, K. P. Becker, B. Phillips, J. Kirby, S. Licht, D. Tchernov, R. J. Wood, and D. F. Gruber, "Soft robotic grippers for biological sampling on deep reefs," *So Ro*, vol. 3, no. 1, pp. 23–33, 2016.

[7] L. Chin, M. C. Yuen, J. Lipton, L. H. Trueba, R. Kramer-Bottiglio, and D. Rus, "A simple electric soft robotic gripper with high-deformation haptic feedback," in *Int. Conf. Rob. Autom.*, May 2019, pp. 2765–2771.

[8] R. J. Webster, III, J. M. Romano, and N. J. Cowan, "Mechanics of precurved-tube continuum robots," *IEEE Trans. Robot.*, vol. 25, no. 1, pp. 67–78, Feb 2009.

[9] J. Burgner-Kahrs, D. C. Rucker, and H. Choset, "Continuum robots for medical applications: A survey," *IEEE Trans. Robot.*, vol. 31, no. 6, pp. 1261–1280, Dec 2015.

[10] J. O. Alcaide, L. Pearson, and M. E. Rentschler, "Design, modeling and control of a sma-actuated biomimetic robot with novel functional skin," in *IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 4338–4345.

[11] H. Abidi, G. Gerboni, M. Brancadoro, J. Fras, A. Diodato, M. Cianchetti, H. Wurdemann, K. Althoefer, and A. Menciassi, "Highly dexterous 2-module soft robot for intra-organ navigation in minimally invasive surgery," *Int J Med Robot*, vol. 14, no. 1, 2 2018.

[12] M. T. Tolley, R. F. Shepherd, B. Mosadegh, K. C. Galloway, M. Wehner, M. Karpelson, R. J. Wood, and G. M. Whitesides, "A resilient, untethered soft robot," *So Ro*, vol. 1, no. 3, pp. 213–223, 2014.

[13] C. D. Onal and D. Rus, "Autonomous undulatory serpentine locomotion utilizing body dynamics of a fluidic soft robot," *Bioinsp Biomim*, vol. 8, no. 2, p. 026003, mar 2013.

[14] J. D. Greer, L. H. Blumenschein, A. M. Okamura, and E. W. Hawkes, "Obstacle-aided navigation of a soft growing robot," in *IEEE Int. Conf. Robot. Autom.*, May 2018, pp. 1–8.

[15] C. Della Santina, R. K. Katzschmann, A. Bicchi, and D. Rus, "Dynamic control of soft robots interacting with the environment," in *IEEE Int. Conf. Soft Rob.*, Apr 2018, pp. 46–53.

[16] C. Della Santina, A. Bicchi, and D. Rus, "Dynamic control of soft robots with internal constraints in the presence of obstacles," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, Nov 2019, pp. 6622–6629.

[17] E. W. Hawkes, L. H. Blumenschein, J. D. Greer, and A. M. Okamura, "A soft robot that navigates its environment through growth," *Sci. Robot.*, vol. 2, no. 8, 2017.

[18] D. Mishima, T. Aoki, and S. Hirose, *Development of Pneumatically Controlled Expandable Arm for Search in the Environment with Tight Access*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 509–518.

[19] H. Tsukagoshi, N. Arai, I. Kiryu, and A. Kitagawa, "Smooth creeping actuator by tip growth movement aiming for search and rescue operation," in *IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1720–1725.

[20] L. H. Blumenschein, N. S. Usevitch, B. H. Do, E. W. Hawkes, and A. M. Okamura, "Helical actuation on a soft inflated robot body," in *IEEE Int. Conf. Soft Robot.*, 2018, pp. 245–252.

[21] L. H. Blumenschein, A. M. Okamura, and E. W. Hawkes, "Modeling of bioinspired apical extension in a soft robot," in *Biomimetic and Biohybrid Systems*. Cham: Springer International Publishing, 2017, pp. 522–531.

[22] J. D. Greer, T. K. Morimoto, A. M. Okamura, and E. W. Hawkes, "A soft, steerable continuum robot that grows via tip extension," *So Ro*, vol. 6, no. 1, pp. 95–108, 2019.

[23] P. Slade, A. Gruebele, Z. Hammond, M. Raitor, A. M. Okamura, and E. W. Hawkes, "Design of a soft catheter for low-force and constrained surgery," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2017, pp. 174–180.

[24] L. H. Blumenschein, L. T. Gan, J. A. Fan, A. M. Okamura, and E. W. Hawkes, "A tip-extending soft robot enables reconfigurable and deployable antennas," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 949–956, 2018.

[25] M. Coad, L. Blumenschein, S. Cutler, J. Reyna Zepeda, N. Naclerio, H. ElHussieny, U. Mehmood, J. Ryu, E. W. Hawkes, and A. Okamura, "Vine robots: Design, teleoperation, and deployment for navigation and exploration," *IEEE Robotics Automation Magazine*, pp. 0–0, 2019.

[26] H. El-Hussieny, U. Mehmood, Z. Mehdi, S.-G. Jeong, M. Usman, E. W. Hawkes, A. M. Okamura, and J.-H. Ryu, "Development and evaluation of an intuitive flexible interface for teleoperating soft growing robots," in *IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2018.

[27] J. D. Greer, L. H. Blumenschein, R. Alterovitz, E. W. Hawkes, and A. M. Okamura, "Robust Navigation of a Soft Growing Robot by Exploiting Contact with the Environment," no. X, pp. 1–13, 2019.

[28] N. D. Naclerio and E. W. Hawkes, "Simple, low-hysteresis, foldable, fabric pneumatic artificial muscle," *Robot. Autom. Lett.*, 2020, accepted.

[29] I. Robert J. Webster and B. A. Jones, "Design and kinematic modeling of constant curvature continuum robots: A review," *Int J Robot Res*, vol. 29, no. 13, pp. 1661–1683, 2010.

[30] L. Wang and N. Simaan, "Geometric calibration of continuum robots: Joint space and equilibrium shape deviations," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 387–402, Apr 2019.

[31] S. Antman, *Nonlinear Problems of Elasticity; 2nd ed.* Dordrecht: Springer, 2005.

[32] F. Largilliere, V. Verona, E. Coevoet, M. Sanz-Lopez, J. Dequidt, and C. Duriez, "Real-time control of soft-robots using asynchronous finite element modeling," in *IEEE Int. Conf. Robot. Autom.*, May 2015, pp. 2550–2555.

[33] S. Grazioso, G. Di Gironimo, and B. Siciliano, "Analytic solutions for the static equilibrium configurations of externally loaded cantilever soft robotic arms," in *IEEE Int. Conf. Soft Robot.*, Apr 2018, pp. 140–145.

[34] S. Grazioso, G. Di Gironimo, and B. Siciliano, "A geometrically exact model for soft continuum robots: The finite element deformation space formulation," *So Ro*, vol. 6, no. 6, pp. 790–811, 2019.

[35] Y. Chen, L. Wang, K. Galloway, I. Godage, N. Simaan, and E. Barth, "Modal-based kinematics and contact detection of soft robots," 2019.

[36] Y. P. Liu, C. G. Wang, H. F. Tan, and M. K. Wadee, "The interactive bending wrinkling behaviour of inflated beams," *P Roy Soc A-Math Phy*, vol. 472, no. 2193, p. 20160504, 2016.

[37] M. A. Crisfield and G. Cole, "Co-rotational beam elements for two- and three-dimensional non-linear analysis," in *Discretization Methods in Structural Mechanics*, G. Kuhn and H. Mang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1990, pp. 115–124.

[38] F. Bullo, *Lectures on Network Systems*, 1st ed. Kindle Direct Publishing, 2019.

[39] G. J. Vrooijink, M. Abayazid, S. Patil, R. Alterovitz, and S. Misra, "Needle path planning and steering in a three-dimensional non-static environment using two-dimensional ultrasound images," *Int J Robot Res*, vol. 33, no. 10, pp. 1361–1374, 2014.