Implementation of a Cosserat Rod-Based Configuration Tracking Controller on a Multi-Segment Soft Robotic Arm

Azadeh Doroudchi*[1], Zhi Qiao*[2], Wenlong Zhang[3], and Spring Berman[2]

***Abstract*— Controlling soft continuum robotic arms is challenging due to their hyper-redundancy and dexterity. In this paper we experimentally demonstrate, for the first time, closed-loop control of the configuration space variables of a soft robotic arm, composed of independently controllable segments, using a Cosserat rod model of the robot and the distributed sensing and actuation capabilities of the segments. Our controller solves the inverse dynamic problem by simulating the Cosserat rod model in MATLAB using a computationally efficient numerical solution scheme, and it applies the computed control output to the actual robot in real time. The position and orientation of the tip of each segment are measured in real time, while the remaining unknown variables that are needed to solve the inverse dynamics are estimated simultaneously in the simulation. We implement the controller on a multi-segment silicone robotic arm with pneumatic actuation, using a motion capture system to measure the segments' positions and orientations. The controller is used to reshape the arm into configurations that are achieved through combinations of bending and extension deformations in 3D space. Although the possible deformations are limited for this robot platform, our study demonstrates the potential for implementing the control approach on a wide range of continuum robots in practice. The resulting tracking performance indicates the effectiveness of the controller and the accuracy of the simulated Cosserat rod model.**

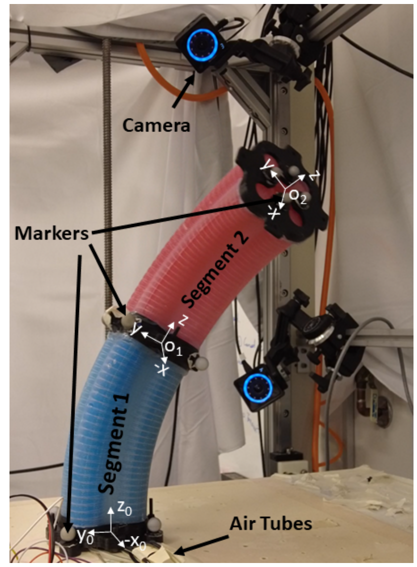


Fig. 1: Multi-segment silicone robotic arm used in this work. The segments were fabricated using a method similar to [1].

I. Introduction

Soft continuum robots have potential uses in manipulation and locomotion tasks that require high dexterity and compliance, and have often been inspired by soft biological structures with these properties, e.g., octopus arms and elephant trunks [2], [3]. The control of soft robotic arms is challenging due to the robots' inherent passive compliance, infinite degrees of freedom, and nonlinear material characteristics [4], [5]. Substantial progress has been made on this problem in recent years [6], [7], [8], and the development of soft segments with independent actuation [9] and soft grippers with integrated sensors [10] are expanding the autonomous capabilities of soft robots in unstructured environments.

Both model-based and model-free approaches have been used to design soft robot arm controllers [4]. Of the model-based approaches, dynamic controllers are generally more accurate than kinematic controllers; however, the high computational complexity and sensing requirements of existing dynamic controllers have limited their use [11]. The first closed-loop dynamic controller for a soft continuum robot was developed in [12] and validated on a 2D soft robot arm, composed of multiple segments with pneumatic actuation, for trajectory tracking and surface following tasks. The inverse dynamics solution was obtained by adding a mixed feedforward–feedback term to the closed-loop controller based on the Lagrangian formulation of the robot dynamics. Another model-based dynamic controller was designed in [13] to control a simulated octopus arm with realistic muscular actuation for reaching and grasping tasks. The octopus arm was modeled as a Cosserat rod, which can describe large deformations due to bending, torsion, shear, and extension [14]. In [15], continuum robots inspired by elongated-body animals were modeled as Kirchhoff rods, a special case of the Cosserat rod model, and used to solve the inverse dynamics problem to simulate terrestrial locomotion gaits. However, there are a limited number of works on implementing the Cosserat rod model in practice on an underactuated continuum robot.

In this paper, we adapt the controller from our previous work [16] to a soft robot arm composed of pneumatically-actuated silicone segments, shown in Fig. 1, and experimentally validate the controller on this platform. The controller in [16] is a model-based inverse dynamic controller that is designed to reshape a multi-segment soft robot arm, modeled

This work was supported by Office of Naval Research Award N00014-17-1-2117 and National Science Foundation Award CMMI-1800940.

*These authors contributed equally to the paper.

[1]Azadeh Doroudchi is with School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ, 85287, USA adoroudc@asu.edu

[2]Zhi Qiao and Spring Berman are with the School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe, AZ, 85287, USA zqiao7@asu.edu, spring.berman@asu.edu

[3]Wenlong Zhang is with the School of Manufacturing Systems and Networks, Fulton Schools of Engineering, Arizona State University, Mesa, AZ, 85212, USA wenlong.zhang@asu.edu

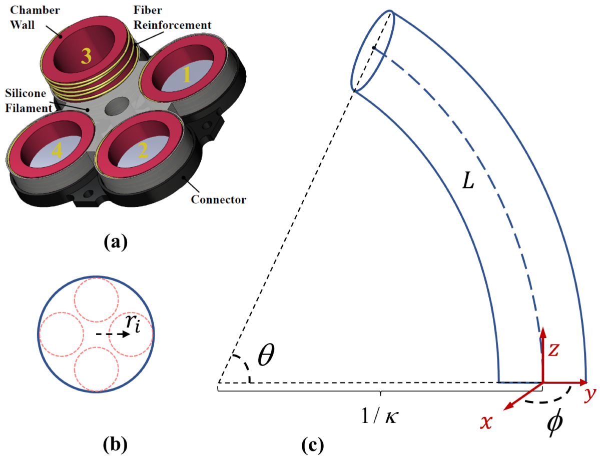


Fig. 2: (a) Design and (b) cross-section of a silicone segment. $\boldsymbol{r}_i$ is the vector from the backbone to the center of the i-th FRA chamber, which has cap area A_i, in the local frame of the cross-section, which has area A. (c) Kinematic representation of a constant-curvature segment with respect to a local frame fixed at the base of the segment.

as a Cosserat rod, into target 3D configurations. To enable the implementation of the controller, we define mappings from the robot's joint space to its actuator space (desired air pressures) and from its task space to its configuration space (curvatures, extension). The inverse dynamics of the robot are solved using real-time measurements of the positions and orientations of its segments, which are obtained by a motion capture system, and estimates of unmeasured variables from the solution to the forward dynamics of the Cosserat rod model. In summary, the main contributions of this paper are:

- Adaptation of our Cosserat rod model-based inverse dynamic controller for 3D configuration tracking [16] to a soft robotic arm composed of independently-controllable segments with pneumatic actuation.
- Estimation of unmeasured robot variables in the inverse dynamics solution using real-time simulation of the Cosserat rod model.
- Experimental validation of the controller on a multi-segment silicone robot that is capable of bending and extending in 3D space, demonstrating that the controller can be implemented on real soft robotic platforms.

II. Silicone Segment Fabrication and Properties

The multi-segment silicone arm was fabricated using a process similar to the one described in [1]. Figure 2a illustrates the design of a silicone segment, Fig. 2b shows its cross-section, and Fig. 2c depicts its kinematic representation as a constant-curvature segment. As shown in Fig. 2a, each segment contains four fiber-reinforced actuators (FRAs), indexed $i = 1, 2, 3, 4$, which are symmetrically arranged with respect to the central axis of the segment and embedded in the segment using silicone filament. The helical fiber reinforcements are wound clockwise and counterclockwise in such a way that the FRAs elongate along the axial direction instead of expanding in the radial direction. The two segments are dyed either blue or red and are connected by black 3D-printed parts, which are fastened with bolts and nuts (see Fig. 1). Compressed air is applied to each segment through four air tubes, each connected to an FRA and a digital pressure regulator. The pressure regulator measures the air pressure p_m in the FRA and drives it to a desired pressure set-point p_d.

TABLE I: Silicone segment properties.

Param.	Description	Value	Units
N	Number of segments	2	–
r_i	Distance from backbone to each FRA	0.3	cm
A_i	FRA cap area	3.1	cm^2
r_0	Undeformed radius of the segment	5.3	cm
L_0	Undeformed length of the segment	18.5	cm
M	Mass of the segment	0.825	kg
ρ	Density of the segment	792.8	kg/m^3
G	Shear modulus of the segment	0.1	MPa
E	Young's modulus of the segment	0.28	MPa

The physical properties of each segment are listed in Table I. To estimate the Young's modulus of a segment, assuming that its value remains constant during actuation, an Instron 5944 universal testing machine was used to elongate the unactuated segment under three loading speeds (1 mm/s, 3 mm/s, and 5 mm/s) and measure its corresponding change in length and the applied force. Three trials were conducted for each loading speed, and the Young's modulus was estimated as the average value over all nine trials.

III. Dynamics and Kinematics of the Robot Arm Based on the Cosserat Rod Model

Here, we derive the forward and inverse dynamics of the robot arm based on a Cosserat rod model, which is accurate under the assumptions of a sufficiently large length-to-radius ratio, material incompressibility, and linear elasticity.

A. Background: Cosserat rod model

The robot arm is modeled as an elastic Cosserat rod of length L and density ρ. Each of its cross-sections has area A and second mass moment of inertia tensor $\boldsymbol{J}$. The cross-section at arc length s in the global frame G has position ${}^G\boldsymbol{p}(t,s) \in \mathbb{R}^3$ and orientation matrix ${}^G\boldsymbol{R}(t,s) \in SO(3)$ at time t (see Fig. 1 in [16] for a schematic of a Cosserat rod in Cartesian coordinates). From here on, a variable without the annotation G is defined with respect to a local frame that is fixed to the cross-section with which it is associated.

The configuration space variables of the rod, given that its neutral axis is in the z direction, are defined as the curvature vector, $\boldsymbol{u}(t,s) = [u_x, u_y, u_z]^T$, and the rate of change of position, $\boldsymbol{v}(t,s) = [v_x, v_y, v_z]^T$. The components u_x and u_y produce bending about the x and y axes; u_z produces torsion about the z-axis; v_x and v_y cause shear effects that change the size of the cross-section; and v_z produces extension along the z-axis. The vectors $\boldsymbol{q}(t,s)$ and $\boldsymbol{w}(t,s)$ denote the translational and angular velocities, respectively, of the cross-section at arc length s. The force and moment that the material at $\boldsymbol{p}(t,s+ds)$ exerts on the material at $\boldsymbol{p}(t,s-ds)$, for infinitesimal ds, are called the *internal* force and moment, ${}^G\boldsymbol{n}(t,s) \in \mathbb{R}^3$ and ${}^G\boldsymbol{m}(t,s) \in \mathbb{R}^3$. Any force and moment that are applied to the backbone are called an *external* force and moment, ${}^{G}\boldsymbol{f}(t,s) \in \mathbb{R}^3$ and ${}^{G}\boldsymbol{l}(t,s) \in \mathbb{R}^3$.

The deformation of each cross-section of the rod is governed by a set of partial differential equations, differentiated with respect to s and t. The spatial derivatives of the state variables are calculated at each cross-section. Defining $\widehat{(\cdot)}$ as the cross product matrix of a vector, the internal force and moment evolve according to the equations:

$$\begin{aligned} {}^{G}\boldsymbol{n}_s &= {}^{G}\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^{G}\boldsymbol{f}, \\ {}^{G}\boldsymbol{m}_s &= {}^{G}\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^{G}\widehat{\boldsymbol{p}}_s {}^{G}\boldsymbol{n} - {}^{G}\boldsymbol{l}, \end{aligned} \tag{1}$$

and the kinematic variables evolve according to:

$$\begin{aligned} &{}^{G}\boldsymbol{p}_s = {}^{G}\boldsymbol{R}\boldsymbol{v}, \quad {}^{G}\boldsymbol{p}_t = \boldsymbol{R}\boldsymbol{q}, \quad \boldsymbol{q}_s = \boldsymbol{v}_t - \widehat{\boldsymbol{u}}\boldsymbol{q} + \widehat{\boldsymbol{w}}\boldsymbol{v}, \\ &{}^{G}\boldsymbol{R}_s = {}^{G}\boldsymbol{R}\widehat{\boldsymbol{u}}, \quad {}^{G}\boldsymbol{R}_t = \boldsymbol{R}\widehat{\boldsymbol{w}}, \quad \boldsymbol{w}_s = \boldsymbol{u}_t - \widehat{\boldsymbol{u}}\boldsymbol{w}. \end{aligned} \tag{2}$$

The time derivatives are computed using the Backward Differentiation Formula (BDF) [17], [18].

B. Joint space to actuator space mapping

The sources of the external forces and moments are the control inputs applied by the robot's pneumatic actuators, which produce force ${}^{G}\boldsymbol{f}_p$ and moment ${}^{G}\boldsymbol{l}_p$, and the gravitational force per unit length in the global frame, ${}^{G}\boldsymbol{f}_e$:

$${}^{G}\boldsymbol{f} = {}^{G}\boldsymbol{f}_p + {}^{G}\boldsymbol{f}_e, \quad {}^{G}\boldsymbol{l} = {}^{G}\boldsymbol{l}_p, \tag{3}$$

$$\text{where} \quad {}^{G}\boldsymbol{f}_e = \rho A {}^{G}\boldsymbol{g}, \quad {}^{G}\boldsymbol{g} = [0\ 0\ -9.81]^T\ m/s^2. \tag{4}$$

The forces and moments applied by the pneumatic actuators to the backbone are given by:

$$\begin{aligned} {}^{G}\boldsymbol{f}_p &= \sum_{i=1}^{4} g_i A_i [{}^{G}\boldsymbol{R}_s \boldsymbol{e}_3] - \rho A {}^{G}\boldsymbol{g}, \\ {}^{G}\boldsymbol{l}_p &= \sum_{i=1}^{4} g_i A_i \frac{\partial}{\partial s}[({}^{G}\boldsymbol{p} + {}^{G}\boldsymbol{R}\boldsymbol{r}_i) \times {}^{G}\boldsymbol{R}\boldsymbol{e}_3], \end{aligned} \tag{5}$$

where g_i is the chamber air pressure of the i-th FRA and, as depicted in Fig. 2b, A_i is the corresponding chamber cap area, $\boldsymbol{r}_i$ is the vector from the center of the backbone to the center of the i-th FRA in the local frame attached to a segment cross-section, and $\boldsymbol{e}_3$ is the unit vector along the z-axis. The gravitational effect in (4) is subtracted from the actuation force to cancel out its effect on the backbone in (3).

Each silicone segment has three DOFs: bending about the x-axis, bending about the y-axis, and elongation along the z-axis. Hence, the equivalent actuation for each segment, $\boldsymbol{P} = [P_x, P_y, P_z]$, can be computed using the mapping below:

$$\begin{aligned} &P_x = {}^{G}\boldsymbol{l}_p(y) \cdot {}^{G}\boldsymbol{l}_p^T(y)/A_i, \ P_z = {}^{G}\boldsymbol{f}_p(z) \cdot {}^{G}\boldsymbol{f}_p^T(z)/A_i, \\ &P_y = {}^{G}\boldsymbol{l}_p(x) \cdot {}^{G}\boldsymbol{l}_p^T(x)/A_i, \end{aligned} \tag{6}$$

where A_i is equal for all chambers. The equivalent actuation is related to the real actuator pressures, g_1, g_2, g_3, g_4, as:

$$\begin{aligned} &P_x = -g_1 + g_2 - g_3 + g_4, \quad P_z = g_1 + g_2 + g_3 + g_4. \\ &P_y = g_1 + g_2 - g_3 - g_4, \end{aligned} \tag{7}$$

In segment 1, two of the four air tubes are connected to one pressure controller and the other two to another pressure controller such that $g_1 = g_2$ and $g_3 = g_4$. The reverse combination is used for segment 2, such that $g_1 = g_3$ and $g_2 = g_4$. The constraint $g_1 + g_4 = g_2 + g_3$ captures both sets of pressure relations. Given this constraint, the real actuator pressures are computed as:

$$\begin{aligned} &g_1 = (-P_x + P_y + P_z)/4, \quad g_2 = (P_x + P_y + P_z)/4, \\ &g_3 = (-P_x - P_y + P_z)/4, \quad g_4 = (P_x - P_y + P_z)/4. \end{aligned} \tag{8}$$

C. Forward dynamics solution

The solution to the forward dynamics of the rod is obtained by substituting (8) into (5), (4) and (5) into (3), and (3) into (1), and then spatially integrating (1) over the length of the rod to calculate the configuration space variables as follows:

$$\begin{aligned} \boldsymbol{v} &= (\boldsymbol{K}_{se} + c_0\boldsymbol{B}_{se})^{-1}[{}^{G}\boldsymbol{R}^T\ {}^{G}\boldsymbol{n} + \boldsymbol{K}_{se}\boldsymbol{v}^* - \boldsymbol{B}_{se}\boldsymbol{v}_h], \\ \boldsymbol{u} &= (\boldsymbol{K}_{bt} + c_0\boldsymbol{B}_{bt})^{-1}[{}^{G}\boldsymbol{R}^T\ {}^{G}\boldsymbol{m} + \boldsymbol{K}_{bt}\boldsymbol{u}^* - \boldsymbol{B}_{bt}\boldsymbol{u}_h]. \end{aligned} \tag{9}$$

Here, $\boldsymbol{v}^*$ and $\boldsymbol{u}^*$ are the vectors $\boldsymbol{v}$ and $\boldsymbol{u}$ at the undeformed reference shape, and the history elements $\boldsymbol{v}_h$ and $\boldsymbol{u}_h$ are calculated from the values of $\boldsymbol{v}$ and $\boldsymbol{u}$ at the previous two time steps [19]. The vectors $\boldsymbol{v}_h$ and $\boldsymbol{u}_h$ are obtained from experimental data on the position and orientation of the robot in its task space, as discussed in Section IV. The matrices $\boldsymbol{K}_{se}$, $\boldsymbol{K}_{bt}$, $\boldsymbol{B}_{se}$, and $\boldsymbol{B}_{bt}$ in (9) are defined as:

$$\begin{aligned} \boldsymbol{K}_{se} &= \begin{bmatrix} \alpha_c G & 0 & 0 \\ 0 & \alpha_c G & 0 \\ 0 & 0 & E \end{bmatrix} A, \quad \boldsymbol{K}_{bt} = \begin{bmatrix} E & 0 & 0 \\ 0 & E & 0 \\ 0 & 0 & G \end{bmatrix} \boldsymbol{J}, \\ \boldsymbol{B}_{se} &= \tau\boldsymbol{K}_{se}, \quad \boldsymbol{B}_{bt} = \tau\boldsymbol{K}_{bt}, \end{aligned} \tag{10}$$

where G and E are the shear modulus and Young's modulus, respectively, of the segment (given in Table I) and $\alpha_c = 4/3$ for circular cross-sections. The damping matrices $\boldsymbol{B}_{se}$ and $\boldsymbol{B}_{bt}$ are calculated from vibration tests [20], in which τ is twice the period of vibrations exhibited by the robot arm's tip. Due to the high stiffness of the segments, we set $\tau \approx 0$.

D. Task space to configuration space mapping

Since the configuration space variables (u_x, u_y, v_z) cannot be directly measured from the robot's task space, the solution to the inverse dynamics problem requires a mapping from the robot's bending and extension deformations in the task space to the corresponding values of u_x, u_y, and v_z. Since this mapping is robot-independent [11], we use the Piecewise Constant Curvature (PCC) configuration space variables, $(\kappa(t,s), \phi(t,s), L(t,s))$, to complete the mapping. These variables are defined for a constant-curvature segment that approximates the silicone segment, illustrated in Fig. 2c: κ is the curvature of the segment's backbone, L is its length, and ϕ is its angle of rotation about the z-axis with respect to the x-axis (positive counterclockwise). The position (x, y, z) of the tip of the segment with respect to the local frame attached to its base is related to the corresponding PCC configuration space variables as follows, defined as in [5]:

$$\begin{aligned} &\kappa = 2x/(x^2+z^2), \ L = (1/\kappa)\tan^{-1}(x/z) \ \text{for } \phi = 0 \text{ rad} \\ &\kappa = 2y/(y^2+z^2), \ L = (1/\kappa)\tan^{-1}(y/z) \ \text{for } \phi = \frac{\pi}{2} \text{ rad} \\ &\kappa = 0, \ \ L = z \quad \text{for } x = 0, \ y = 0. \end{aligned} \tag{11}$$

The segment's backbone lies in the x–z plane at $\phi = 0$ rad and in the y–z plane at $\phi = \pi/2$ rad. The curvature variable κ in each plane equals the component of the curvature vector $\boldsymbol{u}$ in the same plane for bending deformations, and the extension variable v_z is the extension ratio of the segment:

$$\begin{aligned} &u_y = \kappa \ \text{ for } \ \phi = 0 \text{ rad}, \quad v_z = 1 + (L - L_0)/L_0, \\ &u_x = -\kappa \ \text{ for } \ \phi = \pi/2 \text{ rad}, \end{aligned} \tag{12}$$

where L_0 is the undeformed length of the segment. The inverse dynamics solution can be obtained once the control input is computed, which is discussed in the next section.

IV. Inverse Dynamic Control of the Robot Arm

We design a control approach that drives the robot arm to track a time-varying reference configuration $(\bar{\boldsymbol{u}}(t), \bar{\boldsymbol{v}}(t))$, as well as its first and second time derivatives, that achieves desired bending and extension deformations in 3D space. We specify a *high-level controller* that is similar to the dynamic controller in our prior work [16], with proportional-derivative gain matrices that are defined in terms of the physical and material properties of distinct cross-sections of the robot arm. The outputs of the high-level controller are defined as:

$$\begin{aligned} {}^G\boldsymbol{f}_p =& {}^G\boldsymbol{R}[\boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v}) + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt}] \\ & - \rho A \ {}^G\boldsymbol{g}, \\ {}^G\boldsymbol{l}_p =& {}^G\boldsymbol{R}[\boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u}) + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt}], \end{aligned} \tag{13}$$

where $\boldsymbol{K}_{m_1}$ and $\boldsymbol{K}_{m_2}$ are 3×3 diagonal matrices whose diagonal entries are proportional to ρA and $\rho \boldsymbol{J}$, respectively, and $\boldsymbol{K}_{p_1} = a_1 \boldsymbol{K}_{se},\ \boldsymbol{K}_{v_1} = b_1 \boldsymbol{B}_{se},\ \boldsymbol{K}_{p_2} = a_2 \boldsymbol{K}_{bt},\ \boldsymbol{K}_{v_2} = b_2 \boldsymbol{B}_{bt}$ are diagonal gain matrices with manually tuned coefficients $b_1 = 1, b_2 = 1$ for both segments, $a_1 = 100, a_2 = 50$ for segment 1, and $a_1 = 37.5, a_2 = 18.75$ for segment 2. The closed-loop system is proven to be globally asymptotically stable in the Appendix.

Figure 3 shows a block diagram of our controller. The reference configuration $(\bar{\boldsymbol{u}}(t), \bar{\boldsymbol{v}}(t))$ and its first and second time derivatives are sent to the control loop at each time step. The high-level controller computes the control outputs (the desired forces and moments $(\boldsymbol{f}_p, \boldsymbol{l}_p)$ applied by the actuators) according to (13) based on the difference between these reference values and the current configuration $(\boldsymbol{u}(t), \boldsymbol{v}(t))$ and their time derivatives, computed numerically for the simulated robot. The control outputs are mapped to the desired air pressure p_d in each FRA using (6)-(8). Then, the *low-level controller* drives the measured air pressure p_m in each FRA of the real robot to the corresponding desired set-point p_d via the robot's pneumatic actuators. After the arm deforms, the position and orientation of the tip of each segment $(\boldsymbol{p}, \boldsymbol{R})$ are measured by an Optitrack motion capture system with Motive software. Using mapping (11)-(12), the measured task space variables $(\boldsymbol{p}, \boldsymbol{R})$ are used to compute the configuration space variables $(\boldsymbol{u}, \boldsymbol{v})$, which are stored as the history elements $(\boldsymbol{u}_h, \boldsymbol{v}_h)$ for the next two time steps. To close the loop, the values of $(\boldsymbol{u}, \boldsymbol{v})$ and their first time derivatives at the next time step are approximated by simulating the Cosserat model forward dynamics based on (1)-(5), (9)-(10), using high-level controller output (13).

Algorithm 1: Configuration tracking controller

1: Given $\bar{\boldsymbol{u}}^i_j, \bar{\boldsymbol{v}}^i_j,\ \bar{\boldsymbol{u}}^i_{t,j}, \bar{\boldsymbol{v}}^i_{t,j},\ \bar{\boldsymbol{u}}^i_{tt,j}, \bar{\boldsymbol{v}}^i_{tt,j}$,
$\quad i = 1, ..., T/dt,\ j = 1, ..., L/ds$
2: **for** $i \leftarrow 1$ to T/dt **do**
3: $\quad \boldsymbol{n}^i_0, \boldsymbol{m}^i_0 \leftarrow$ SSM $(\boldsymbol{n}^{i-1}_L = \boldsymbol{m}^{i-1}_L = \boldsymbol{0})$
4: $\quad$ **for** $j \leftarrow 1$ to L/ds **do**
5: $\quad\quad \boldsymbol{n}^i_j, \boldsymbol{m}^i_j \leftarrow$ RK4 using $(\boldsymbol{n}^i_{j-1}, \boldsymbol{m}^i_{j-1})$ and $(\boldsymbol{n}^i_{s,j-1}, \boldsymbol{m}^i_{s,j-1})$
6: $\quad\quad \boldsymbol{u}^i_j, \boldsymbol{v}^i_j \leftarrow$ Forward dynamics (9) using $\boldsymbol{n}^i_j, \boldsymbol{m}^i_j$ and $\boldsymbol{u}^i_{h,j}, \boldsymbol{v}^i_{h,j}$
7: $\quad\quad \boldsymbol{f}^i_{p,j}, \boldsymbol{l}^i_{p,j} \leftarrow$ Control law (13) using $\boldsymbol{v}^i_j, \boldsymbol{u}^i_j$ and numerical approximations of their time derivatives
8: $\quad\quad \boldsymbol{P}^i_j \leftarrow$ Eq. (6) using $\boldsymbol{f}^i_{p,j}, \boldsymbol{l}^i_{p,j}$
9: $\quad\quad \boldsymbol{n}^i_{s,j}, \boldsymbol{m}^i_{s,j} \leftarrow$ Eq. (1) using $\boldsymbol{f}^i_{p,j}, \boldsymbol{l}^i_{p,j}$
10: $\quad$ **end for**
11: **end for**
Note: $(\boldsymbol{n}, \boldsymbol{m}, \boldsymbol{f}, \boldsymbol{l})$ are defined in the global frame, and $(\boldsymbol{v}, \boldsymbol{u}, \boldsymbol{P})$ in the local frame.

Algorithm 1 outlines the numerical solution of the forward and inverse dynamic problems in the simulation. The solution of the forward dynamics is found using the computationally inexpensive approach in [19]. First, the time-varying reference configuration and its first and second time derivatives are defined (line 1). An *outer loop* iterates over time steps i (lines 2 to 11), and an *inner loop* iterates over discretized spatial locations (nodes) j along the backbone of the robot (lines 4 to 10). In the outer loop, the boundary conditions $\boldsymbol{n}^i_0 = {}^G\boldsymbol{n}(i, 0)$ and $\boldsymbol{m}^i_0 = {}^G\boldsymbol{m}(i, 0)$ of the fixed end of the arm are guessed using the standard shooting method (SSM) (line 3), which converts a two-point boundary value problem (BVP) to an initial value problem (IVP), as described in [21], given that $\boldsymbol{n}^{i-1}_L = {}^G\boldsymbol{n}(i-1, L) = \boldsymbol{0}$ and $\boldsymbol{m}^{i-1}_L = {}^G\boldsymbol{m}(i-1, L) = \boldsymbol{0}$ at the free end of the arm and $\boldsymbol{n}^0_0$ and $\boldsymbol{m}^0_0$ are set to $\boldsymbol{0}$. The implicit fourth-order Runge-Kutta (RK4) method is applied to compute the internal force and moment $\boldsymbol{n}^i_j$, $\boldsymbol{m}^i_j$ from their current values and spatial derivatives at spatial node $j-1$ (line 5). Next, the values of $\boldsymbol{v}^i_j$ and $\boldsymbol{u}^i_j$ are computed according to (9) using the numerically integrated $\boldsymbol{n}^i_j, \boldsymbol{m}^i_j$ and the history elements $\boldsymbol{v}^i_{h,j}, \boldsymbol{u}^i_{h,j}$ from the past two time steps (line 6). Using the error between the configuration space variables and their desired values and the error between their actual and desired time derivatives, control law (13) is used to calculate the force and moment $(\boldsymbol{f}^i_{p,j}, \boldsymbol{l}^i_{p,j})$ that the pneumatic actuators must apply to the corresponding backbone section (line 7). These forces and moments are mapped to the equivalent actuation $\boldsymbol{P}$ (line 8), which is converted using (8) into the desired actuator pressures that are sent to the actual robot. Lastly, $\boldsymbol{n}^i_{s,j}$ and $\boldsymbol{m}^i_{s,j}$ are found for the next iteration of the inner loop (line 9).

V. Simulation and Experimental Results

We compare the controller's performance at tracking reference configurations in the actual robot arm (Fig. 1) and its simulation, which runs simultaneously in real time.

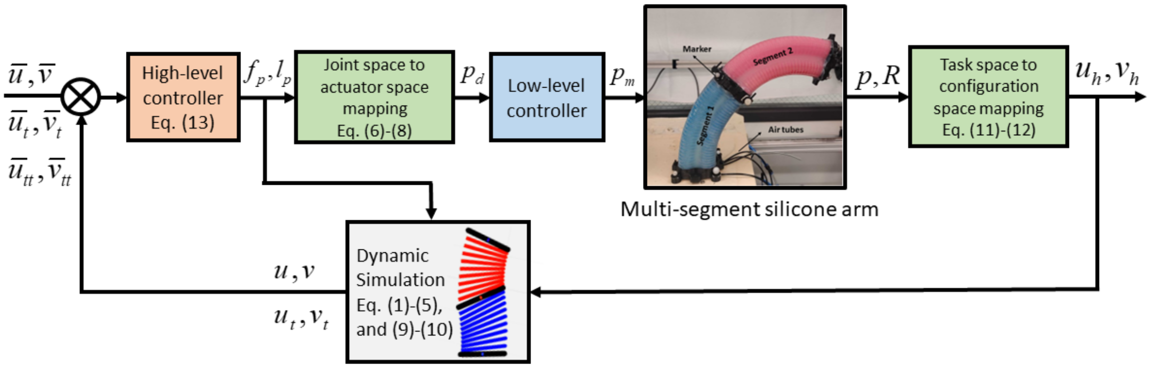
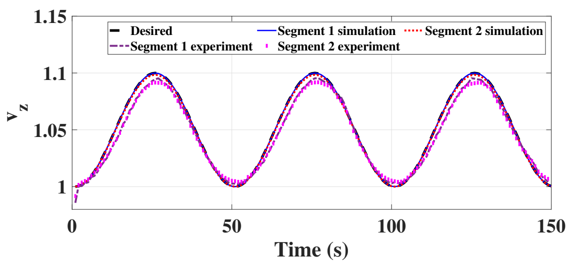
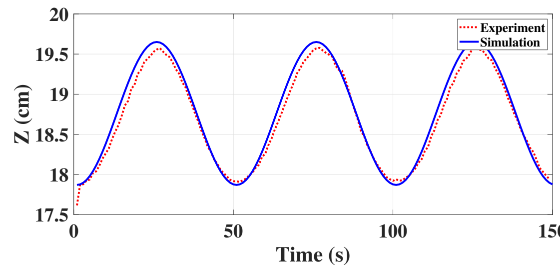


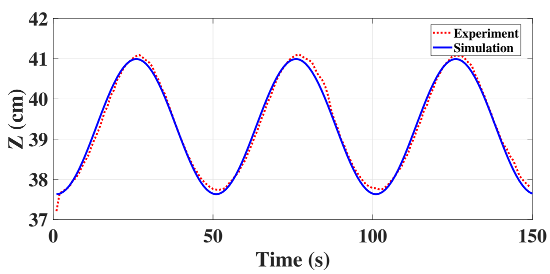
Fig. 3: Schematic block diagram of the configuration tracking controller.



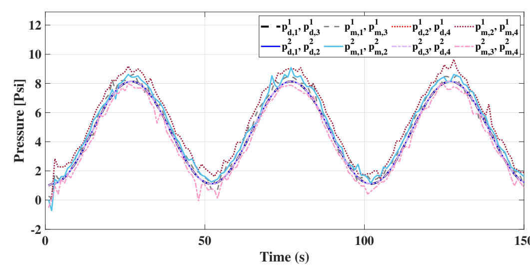
(a) Desired, simulated, and experimental $v_z(t)$ for both segments.



(b) Height $z(t)$ of segment 1's tip in simulation and experiment.



(c) Height $z(t)$ of segment 2's tip in simulation and experiment.



(d) Desired pressure $p^i_{d,j}$ and measured pressure $p^i_{m,j}$ for actuator j of segment i during the experiment.

Fig. 4: Controller performance for tracking the reference extension $\bar{v}_z(t)$ in (14) in an experimental trial with the soft robotic arm and the corresponding simulation of its dynamical model.

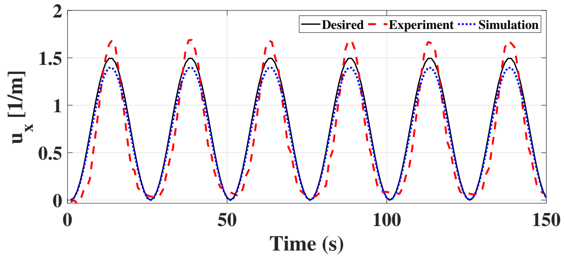
As mentioned, the controller can produce all four main deformations; however, due to the robot's physical design restrictions, we can only validate the controller for bending and extension. Also, while there are no theoretical limitations imposed by our modeling and control approach on the number of segments, we used only two segments due to space constraints on the motion capture system. The simulated arm is slender, uniform, and symmetric about the z-axis and has isotropic material properties and a circular cross-section with radius r_0. Our controller can also be applied to multi-segment robots with other cross-section geometries if they satisfy the assumptions required for using the Cosserat rod model.

We tested sinusoidal reference inputs with various amplitudes and frequencies that produce extension and bending under open-loop actuation. The position and orientation of each segment's tip were recorded using motion capture cameras. The dead-zone of the electric valve of each FRA was avoided by pre-loading the FRAs at a pressure of 1 psi. We describe results for reference inputs that kept the robot within the tracked space of the motion capture system: one that produced extension of both segments, and four that produced simultaneous bending of segment 1 about the $(+x)$- or $(-x)$-axis (i.e., toward the $(-y)$- or $(+y)$-axis) and bending of segment 2 about the $(+y)$- or $(-y)$-axis (i.e., toward the $(+x)$- or $(-x)$-axis). Videos of tests with these inputs, including ones not discussed here due to space limitations, are shown in the supplementary attachment.

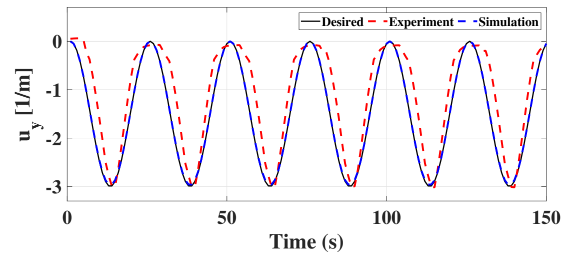
The reference input that produced extension was given by:

$$\bar{v}_z(t) = 1 + a\sin^2(\omega t), \quad \text{(segments 1 and 2)} \qquad (14)$$

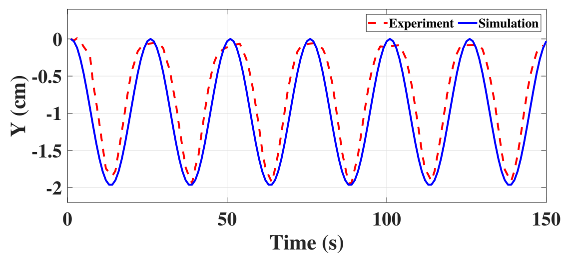
with $a = 0.1$ cm, $\omega = 2\pi/100$ rad/s. Figures 4a-d compare the desired, simulated, and actual extensions $v_z(t)$, the z coordinates of the segment tips, and the desired and measured actuator pressures. The figures show that the robot closely tracks the reference input $\bar{v}_z(t)$, and that the simulation accurately predicts the robot's deformation over time.



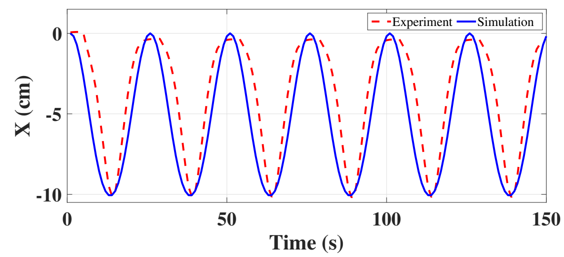
(a) Desired, simulated, and experimental $u_x(t)$ for segment 1.



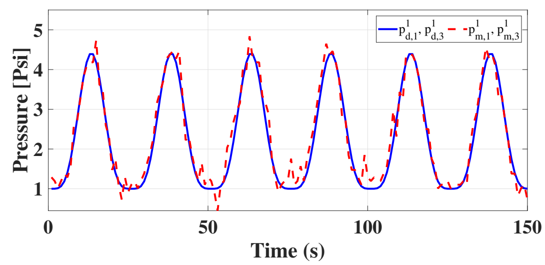
(b) Desired, simulated, and experimental $u_y(t)$ for segment 2.



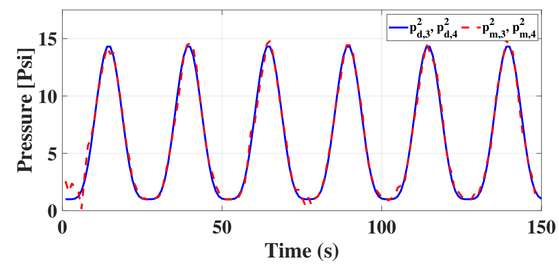
(c) y coordinate of segment 1's tip in simulation and experiment.



(d) x coordinate of segment 2's tip in simulation and experiment.



(e) Desired pressure $p^1_{d,j}$ and measured pressure $p^1_{m,j}$ for actuator j of segment 1 during the experiment.



(f) Desired pressure $p^2_{d,j}$ and measured pressure $p^2_{m,j}$ for actuator j of segment 2 during the experiment.

Fig. 5: Controller performance for tracking the reference curvatures $\bar{u}_x(t)$ (segment 1) and $\bar{u}_y(t)$ (segment 2) in (15) in an experimental trial with the soft robotic arm and the corresponding simulation of its dynamical model.

Of the reference inputs that produced bending, here we present the results for the input that caused segment 1 to bend about the $(+x)$-axis and segment 2 to bend about the $(-y)$-axis. This input was defined as the following curvatures:

$$\begin{aligned} \bar{u}_x(t) &= b \sin^2(\omega t), \quad \text{(segment 1)} \\ \bar{u}_y(t) &= c \sin^2(\omega t), \quad \text{(segment 2)} \end{aligned} \tag{15}$$

where $b = 1.5$ cm, $c = -3$ cm, $\omega = 2\pi/50$ rad/s. Figure 5 compares the simulated and actual curvatures $u_x(t)$ and $u_y(t)$, and x or y coordinate of the segment tips, as well as the desired and measured actuator pressures. The robot tracks the reference input fairly well (Figs. 5a,b) and generally follows the prediction of the simulation (Figs. 5a-d). The main source of the robot's tracking error and its discrepancy from the simulation is the delay between the pneumatic actuation and the resulting elongation or shortening of the FRAs, which was not included in the Cosserat model forward dynamics. This delay could be reduced in practice with modifications to the actuation mechanism, such as adding a vacuum to speed up deflation of the FRAs.

Table II lists the root-mean-square errors (RMSEs) between the reference inputs and the corresponding extension or curvature of each segment during the simulations and experiments. We computed the RMSEs from the time series plotted over 150 s in Figs. 4a, 5a, and 5b. The highest RMSEs are the errors between the experimental and reference curvatures, which are largely due to the previously described delay between the actuation and resulting robot deformation.

TABLE II: RMSEs with respect to reference deformations.

Deformation	Segment	Simulation	Experiment
v_z	1	0.000	0.008
v_z	2	0.000	0.009
u_x	1	0.061	0.318
u_y	2	0.014	0.662

VI. Conclusions and Future Work

We have implemented a computationally efficient control approach, using distributed sensing and actuation, on a multi-segment soft robot arm to track desired 3D configurations. The position and orientation of each segment's tip are measured externally and fed back to a simulated Cosserat rod model of the robot. This model is used to estimate the configuration space variables and their first time derivatives, which precludes the need for force and torque sensors on the robot, and to compute the control outputs. The segments' local sensing and actuation capabilities allow them to be controlled independently and facilitate tracking of diverse configurations that could be used for complex manipulation tasks. A feasible next step toward expanding the robot's autonomous operations is to add more segments and include embedded force sensors to enable estimation and control of the robot's interactions with its environment, similar to the proprioceptive grasping method in [10] for a soft robot hand.

APPENDIX

To assess the stability of the closed-loop system, we substitute expressions (13) for the high-level controller outputs ${}^G\boldsymbol{f}_p, {}^G\boldsymbol{l}_p$ into (3), which we then substitute into (1) to obtain:

$$\begin{aligned}
&{}^G\boldsymbol{R}[\boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt} + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v})] \\
&\quad = {}^G\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^G\boldsymbol{n}_s, \\
&{}^G\boldsymbol{R}[\boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt} + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u})] \\
&\quad = {}^G\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^G\widehat{\boldsymbol{p}}_s{}^G\boldsymbol{n} - {}^G\boldsymbol{m}_s.
\end{aligned} \tag{16}$$

The right-hand sides of these equations are the sums of the internal forces and moments with respect to the arc length. By defining $\boldsymbol{n'}_s$ and $\boldsymbol{m'}_s$ as the following expressions,

$$\begin{aligned}
{}^G\boldsymbol{n'}_s &= {}^G\boldsymbol{R}\rho A(\widehat{\boldsymbol{w}}\boldsymbol{q} + \boldsymbol{q}_t) - {}^G\boldsymbol{n}_s, \\
{}^G\boldsymbol{m'}_s &= {}^G\boldsymbol{R}\rho(\widehat{\boldsymbol{w}}\boldsymbol{J}\boldsymbol{w} + \boldsymbol{J}\boldsymbol{w}_t) - {}^G\widehat{\boldsymbol{p}}_s{}^G\boldsymbol{n} - {}^G\boldsymbol{m}_s,
\end{aligned} \tag{17}$$

and rewriting them in terms of the second time derivatives of the configuration space variables, ${}^G\boldsymbol{n'}_s = {}^G\boldsymbol{R}\boldsymbol{K}_{m_1}\boldsymbol{v}_{tt}$ and ${}^G\boldsymbol{m'}_s = {}^G\boldsymbol{R}\boldsymbol{K}_{m_2}\boldsymbol{u}_{tt}$, the closed-loop configuration dynamics of the robot can be expressed as:

$$\begin{aligned}
\boldsymbol{K}_{m_1}\bar{\boldsymbol{v}}_{tt} + \boldsymbol{K}_{v_1}(\bar{\boldsymbol{v}}_t - \boldsymbol{v}_t) + \boldsymbol{K}_{p_1}(\bar{\boldsymbol{v}} - \boldsymbol{v}) &= \boldsymbol{K}_{m_1}\boldsymbol{v}_{tt}, \\
\boldsymbol{K}_{m_2}\bar{\boldsymbol{u}}_{tt} + \boldsymbol{K}_{v_2}(\bar{\boldsymbol{u}}_t - \boldsymbol{u}_t) + \boldsymbol{K}_{p_2}(\bar{\boldsymbol{u}} - \boldsymbol{u}) &= \boldsymbol{K}_{m_2}\boldsymbol{u}_{tt}.
\end{aligned} \tag{18}$$

Defining the error vector $\boldsymbol{e}(t) = (\bar{\boldsymbol{v}} - \boldsymbol{v}, \bar{\boldsymbol{u}} - \boldsymbol{u})^T$ and writing (18) in terms of $\boldsymbol{e}(t)$, the closed-loop dynamics take the form of a homogeneous second-order differential equation, $\boldsymbol{e}_{tt} + \boldsymbol{K'}_v\boldsymbol{e}_t + \boldsymbol{K'}_p\boldsymbol{e} = \boldsymbol{0}$, with matrices $\boldsymbol{K'}_v$ and $\boldsymbol{K'}_p$ given by:

$$\begin{aligned}
\boldsymbol{K'}_v &= \begin{bmatrix} \boldsymbol{K}_{v_1} \oslash \boldsymbol{K}_{m_1} & 0 \\ 0 & \boldsymbol{K}_{v_2} \oslash \boldsymbol{K}_{m_2} \end{bmatrix}, \\
\boldsymbol{K'}_p &= \begin{bmatrix} \boldsymbol{K}_{p_1} \oslash \boldsymbol{K}_{m_1} & 0 \\ 0 & \boldsymbol{K}_{p_2} \oslash \boldsymbol{K}_{m_2} \end{bmatrix},
\end{aligned} \tag{19}$$

in which $\oslash$ denotes element-wise division of matrices (Hadamard division). These matrices are symmetric and positive definite. We choose the positive definite Lyapunov function $V = \frac{1}{2}(\boldsymbol{e}_t^T\boldsymbol{e}_t + \boldsymbol{e}^T\boldsymbol{K'}_p\boldsymbol{e})$, which has time derivative:

$$\begin{aligned}
V_t &= \frac{1}{2}\boldsymbol{e}_{tt}^T\boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T\boldsymbol{e}_{tt} + \frac{1}{2}\boldsymbol{e}_t^T\boldsymbol{K'}_p\boldsymbol{e} + \frac{1}{2}\boldsymbol{e}^T\boldsymbol{K'}_p\boldsymbol{e}_t \\
&= \frac{1}{2}(\boldsymbol{e}_{tt}^T + \boldsymbol{e}^T\boldsymbol{K'}_p)\boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T(\boldsymbol{e}_{tt} + \boldsymbol{K'}_p\boldsymbol{e}) \\
&= \frac{1}{2}(-\boldsymbol{e}_t^T\boldsymbol{K'}_v)\boldsymbol{e}_t + \frac{1}{2}\boldsymbol{e}_t^T(-\boldsymbol{K'}_v\boldsymbol{e}_t) = -\boldsymbol{e}_t^T\boldsymbol{K'}_v\boldsymbol{e}_t.
\end{aligned}$$

Since $\boldsymbol{K'}_v$ is positive definite, V_t is a negative definite function. By applying Lyapunov's direct method to the closed-loop system dynamics, we can prove that $\boldsymbol{e}(t) \to \boldsymbol{0}$ as $t \to \infty$ and the system is globally asymptotically stable [22].

REFERENCES

[1] P. H. Nguyen, S. Sridar, W. Zhang, and P. Polygerinos, "Design and control of a 3-chambered fiber reinforced soft actuator with off-the-shelf stretch sensors," *Int. Journal of Intelligent Robotics and Applications*, vol. 1, no. 3, pp. 342–351, 2017.

[2] A. Kapadia and I. D. Walker, "Task-space control of extensible continuum manipulators," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2011, pp. 1087–1092.

[3] M. Calisti, M. Giorelli, G. Levy, B. Mazzolai, B. Hochner, C. Laschi, and P. Dario, "An octopus-bioinspired solution to movement and manipulation for soft robots," *Bioinspiration & biomimetics*, vol. 6, no. 3, p. 036002, 2011.

[4] 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.

[5] H. Jiang, Z. Wang, Y. Jin, X. Chen, P. Li, Y. Gan, S. Lin, and X. Chen, "Hierarchical control of soft manipulators towards unstructured interactions," *Int. J. Robot. Res.*, vol. 40, no. 1, pp. 411–434, 2021.

[6] X. Chen, X. Zhang, Y. Huang, L. Cao, and J. Liu, "A review of soft manipulator research, applications, and opportunities," *J. Field Robot.*, vol. 39, no. 3, pp. 281–311, 2022.

[7] P. Schegg and C. Duriez, "Review on generic methods for mechanical modeling, simulation and control of soft robots," *Plos one*, vol. 17, no. 1, p. e0251059, 2022.

[8] G. Mengaldo, F. Renda, S. L. Brunton, M. Bächer, M. Calisti, C. Duriez, G. S. Chirikjian, and C. Laschi, "A concise guide to modelling the physics of embodied intelligence in soft robotics," *Nature Reviews Physics*, pp. 1–16, 2022.

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

[10] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Robust proprioceptive grasping with a soft robot hand," *Autonomous Robots*, vol. 43, no. 3, pp. 681–696, 2019.

[11] R. J. Webster III 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.

[12] R. K. Katzschmann, 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 *IEEE Int. Conf. on Soft Robotics*, 2019, pp. 454–461.

[13] H.-S. Chang, U. Halder, E. Gribkova, A. Tekinalp, N. Naughton, M. Gazzola, and P. G. Mehta, "Controlling a cyberoctopus soft arm with muscle-like actuation," in *IEEE Int. Conf. on Decision and Control*, 2021, pp. 1383–1390.

[14] S. S. Antman, "Problems in nonlinear elasticity," *Nonlinear Problems of Elasticity*, pp. 513–584, 2005.

[15] F. Boyer, S. Ali, and M. Porez, "Macrocontinuous dynamics for hyperredundant robots: application to kinematic locomotion bioinspired by elongated body animals," *IEEE Transactions on Robotics*, vol. 28, no. 2, pp. 303–317, 2011.

[16] A. Doroudchi and S. Berman, "Configuration tracking for soft continuum robotic arms using inverse dynamic control of a Cosserat rod model," in *IEEE Int. Conf. on Soft Robotics*, 2021, pp. 207–214.

[17] J. Stoer and R. Bulirsch, *Introduction to numerical analysis*. Springer Science & Business Media, 2013, vol. 12.

[18] M. Schaller, S. M. R. Sorkhabadi, and W. Zhang, "Robotic shoe: An ankle assistive device for gait plantar flexion assistance," in *2020 Design of Medical Devices Conference*. American Society of Mechanical Engineers Digital Collection, 2019.

[19] J. Till, V. Aloi, and C. Rucker, "Real-time dynamics of soft and continuum robots based on Cosserat rod models," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 723–746, 2019.

[20] J. Linn, H. Lang, and A. Tuganov, "Geometrically exact Cosserat rods with Kelvin–Voigt type viscous damping," *Mechanical Sciences*, vol. 4, no. 1, pp. 79–96, 2013.

[21] R. W. Holsapple, "A modified simple shooting method for solving two-point boundary value problems," Ph.D. dissertation, Texas Tech University, 2003.

[22] H. K. Khalil and J. W. Grizzle, *Nonlinear systems*. Prentice Hall Upper Saddle River, NJ, 2002, vol. 3.