Dynamics and Control of Soft Robots With Implicit Strain Parametrization

Federico Renda, *Member, IEEE*, Anup Mathew, *Member, IEEE*, and Daniel Feliu Talegon

***Abstract*—Efficient modeling of soft robots for design optimization and control is an active area of research. When representing soft robots as an assembly of rods and rigid bodies, strain-based parametrization has proved to be able to reduce the required number of degrees of freedom drastically. However, this reduction ability strongly depends on the choice of strain basis employed to describe the system. In this letter, we proposed a new implicit strain parametrization, and we showed its use to represent the system's configuration manifold with no more degrees of freedom than the number of actuators. Coupled with standard strain bases, this parametrization is applied to the dynamic simulation and control of soft robots employing a handful of generalized coordinates, drastically simplifying the control design problem. The approach is validated against a high-order model and exploited for shape (configuration space) and tip-pose (task-space) control of a complex tendon-driven soft robot.**

***Index Terms*—Modeling, Control, and Learning for Soft Robots.**

I. Introduction

BY DEFINITION, the motion of soft and continuum robots is predominantly defined by the large deformation of the elastic materials that constitute the main part of the robot's body. Their intrinsic adaptability makes them ideal candidates in applications that require negotiating intricate and confined spaces with extended contact interactions, such as aerospace engine inspection and minimally invasive surgery [1]. The quest for efficient modeling of such virtually infinite Degrees of Freedom (DOF) systems is an active area of research, which not only allows model-based design of the soft robots' mechanism and control but also contributes to the general understanding of the mechanics and numerical computation of highly deformable dynamical systems [2].

Manuscript received 12 October 2023; accepted 18 January 2024. Date of publication 31 January 2024; date of current version 9 February 2024. This letter was recommended for publication by Associate Editor D. Shin and Editor C. Laschi upon evaluation of the reviewers' comments. This work was supported in part by the US Office of Naval Research Global under Grant N62909-21-1-2033 and in part by Khalifa University under Awards RIG-2023-048 and RC1-2018-KUCARS. *(Corresponding author: Federico Renda.)*

Federico Renda is with the Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi 127788, UAE, and also with the Department of Mechanical and Nuclear Engineering, Khalifa University of Science and Technology, Abu Dhabi 127788, UAE (e-mail: federico.renda@ku.ac.ae).

Anup Mathew is with the Khalifa University Center for Autonomous Robotic Systems (KUCARS), Khalifa University of Science and Technology, Abu Dhabi 127788, UAE (e-mail: anup.mathew@ku.ac.ae).

Daniel Feliu Talegon is with the Department of Mechanical and Nuclear Engineering, Khalifa University of Science and Technology, Abu Dhabi 127788, UAE (e-mail: danielfeliu91@hotmail.com).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3360813, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3360813

Depending on the topology of the soft robot, recent approaches have leveraged consolidated theories of 3D and rod continuum mechanics, adapting them to the unique requirements of soft robotics applications. For general soft robots with significant volumetric deformation, the Finite Element Method (FEM) is successfully used to discretize and solve the equation of motion of the system [3]. The FEM approach employs a large number of DOFs to discretize the system. Thus, it requires sophisticated numerical architectures and substantial simplification to speed up the computation for control and design optimization purposes, especially in a highly dynamic regime [4]. On the other hand, it can be applied to a wide variety of soft robots, although it can struggle to handle effectively articulated systems composed of rigid and soft parts due to numerous mechanical constraints. If volumetric deformation is negligible with respect to bending and torsion deformation, a soft robot can be modeled as an assembly of elastic rods and rigid bodies. The Cosserat rod model is the most used in soft robotics to handle these kinds of systems [5]. A shooting method was initially proposed to solve the soft robot static equilibrium [6]. However, intrinsic instability issues have been highlighted for dynamic simulations [7]. Recently, a strain-based discretization of the Cosserat rod was proposed, named the Geometric Variable Strain (GVS) approach [8], [9], which can represent complex space deformations with a relatively limited number of DOFs in a Lagrangian form. Remarkably, the GVS approach generalized other popular modeling methods, such as the Piecewise Constant Curvature (PCC) model [10], as well as the screw-based approach for rigid robots [11], through the selection of the strain basis representing the system. Other models for soft robots include the discrete Cosserat approach [12] and the data-driven learning methods [13].

As mentioned earlier, the strain basis selection profoundly characterizes the identity of the GVS approach. By default, up to fifth or sixth-order Legendre polynomials are used for each strain modality with significant variations (shears may or may not be included, for instance) to accurately model a soft robot in a typical scenario [9]. Otherwise, if some a priori information about the robot shapes is known, that can be used to craft a custom basis tailored to the specific situation. For example, this is the case of multi-segment, parallel-actuator manipulators with negligible interaction with the environment, which leads to the PCC approximation [14]. In this manuscript, we introduce a new class of strain parametrization for the GVS approach, which employs implicit functions of the generalized coordinates instead of explicit ones. In particular, this new class of strain parametrization allows for a drastic reduction in the number of DOFs involved by exploiting the static equilibrium of the system. A similar use of the static equilibrium was also applied in the inverse kinematic control of [15]. Here, this

approach is systematized and extended to dynamic modeling and control.

The manuscript is organized as follows. Section II extends the GVS approach to the implicit strain parametrization with particular attention to threadlike-actuated systems. Section III presents the dynamic modeling and simulation of soft manipulators based on the implicit static equilibrium. Section IV describes a shape and tip-pose controller for the same system. The manuscript ends with a conclusive remark in Section V.

II. Implicit Strain Dynamics

In this section, we present a general formulation of the dynamics of soft manipulators parametrized by an implicit description of their strains, which extends the recently proposed GVS to this new class of strain definition.

A. Kinematics

We start with a brief description of the Cosserat rod kinematics on which the GVS approach is based. The position and orientation of the rigid cross-section of a Cosserat rod are defined by the curve:

$$\boldsymbol{g}(\cdot) : X \mapsto \boldsymbol{g}(X) = \begin{pmatrix} \mathbf{R}(X) & \mathbf{r}(X) \\ \mathbf{0} & 1 \end{pmatrix} \in SE(3) \,, \quad (1)$$

where $\mathbf{r}(X) \in \mathbb{R}^3$ is the position, $\mathbf{R}(X) \in SO(3)$ represents the orientation of the frame attached to the cross-section, and $X \in [0, L]$ is a material curvilinear abscissa. The time ($\dot{(\cdot)}$) and space ($(\cdot)'$) derivative of (1) define the vector field of velocity twist $\boldsymbol{\eta}$ and the one of strain twist $\boldsymbol{\xi}$ (upon comparison to the reference field of the stress-free configuration):

$$\dot{\boldsymbol{g}}(X) = \boldsymbol{g}\widehat{\boldsymbol{\eta}}(X) \,, \quad (2)$$

$$\boldsymbol{g}'(X) = \boldsymbol{g}\widehat{\boldsymbol{\xi}}(X) \,, \quad (3)$$

$$\widehat{\boldsymbol{\eta}} = \begin{pmatrix} \tilde{\mathbf{w}} & \mathbf{v} \\ \mathbf{0} & 0 \end{pmatrix} \in \mathfrak{se}(3) \,, \quad \boldsymbol{\eta} = (\widehat{\boldsymbol{\eta}})^{\vee} = \left(\mathbf{w}^T, \mathbf{v}^T\right)^T \in \mathbb{R}^6 \,. \quad (4)$$

$$\widehat{\boldsymbol{\xi}} = \begin{pmatrix} \tilde{\mathbf{k}} & \mathbf{p} \\ \mathbf{0} & 0 \end{pmatrix} \in \mathfrak{se}(3), \boldsymbol{\xi} = \left(\widehat{\boldsymbol{\xi}}\right)^{\vee} = \left(\mathbf{k}^T, \mathbf{p}^T\right)^T \in \mathbb{R}^6 \,, \quad (5)$$

In (4), $\tilde{\mathbf{w}}(X) \in \mathfrak{so}(3)$,$\mathbf{w}(X) \in \mathbb{R}^3$ and $\mathbf{v}(X) \in \mathbb{R}^3$ are the angular and linear velocity in the local frame at X. In (5), $\tilde{\mathbf{k}}(X) \in \mathfrak{so}(3)$,$\mathbf{k}(X) \in \mathbb{R}^3$ and $\mathbf{p}(X) \in \mathbb{R}^3$ are the angular and linear strains in the local frame at X. Equating the mixed time and space derivatives of (1) gives the compatibility equation of the velocity and strain fields.

$$\frac{\partial}{\partial t}\boldsymbol{g}' = \frac{\partial}{\partial X}\dot{\boldsymbol{g}} \Rightarrow \boldsymbol{\eta}' = \dot{\boldsymbol{\xi}} - \mathrm{ad}_{\boldsymbol{\xi}}\boldsymbol{\eta} \,. \quad (6)$$

Taking another time derivative yields the compatibility equation at the acceleration level.

$$\dot{\boldsymbol{\eta}}' = \ddot{\boldsymbol{\xi}} - \mathrm{ad}_{\dot{\boldsymbol{\xi}}}\boldsymbol{\eta} - \mathrm{ad}_{\boldsymbol{\xi}}\dot{\boldsymbol{\eta}} \,. \quad (7)$$

The adjoint operator ad in $\mathfrak{se}(3)$ is defined in Appendix A.

The GVS approach utilizes the strain field $\boldsymbol{\xi}$ to represent the soft robot configuration. The integration of (3) provides the kinematic relation, while the integration of (6) and (7) give the velocity and acceleration model. Equations (3), (6), and (7) are linear matrix differential equations that can be formally solved for the fields $\boldsymbol{g}$, $\boldsymbol{\eta}$, and $\dot{\boldsymbol{\eta}}$ as follows:

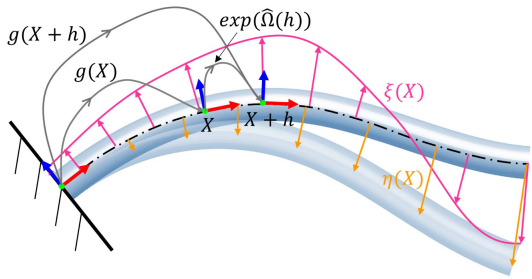

Fig. 1. Schematic of the GVS kinematics approach illustrating the strain and velocity fields along with the recursive formulation of the robot pose $\boldsymbol{g}(X)$ (11).

$$\boldsymbol{g}(X) = \exp\left(\widehat{\boldsymbol{\Omega}}(X)\right) \,, \quad (8)$$

$$\boldsymbol{\eta}(X) = \mathrm{Ad}^{-1}_{\boldsymbol{g}(X)} \int_0^X \mathrm{Ad}_{\boldsymbol{g}}\dot{\boldsymbol{\xi}} ds \,, \quad (9)$$

$$\dot{\boldsymbol{\eta}}(X) = \mathrm{Ad}^{-1}_{\boldsymbol{g}(X)} \left(\int_0^X \left(\dot{\mathrm{Ad}}_{\boldsymbol{g}}\dot{\boldsymbol{\xi}} + \mathrm{Ad}_{\boldsymbol{g}}\ddot{\boldsymbol{\xi}} \right) ds \right) , \quad (10)$$

where the exponential exp and adjoint Ad maps in $SE(3)$ are defined in Appendix B and Appendix A respectively. In (8), $\boldsymbol{\Omega}(X) \in \mathfrak{se}(3)$ represents the Magnus expansion of $\boldsymbol{\xi}$ [16]. Although exact, the preceding equations cannot be computed explicitly. Thus, we use a quadrature approximation of the Magnus expansion $\boldsymbol{\Omega}(h)$ that depends only on the value of the strain at a few points in an interval $[X, X + h]$. With that, we obtain a recursive formulation of the kinematic equations.

$$\boldsymbol{g}(X + h) = \boldsymbol{g}(X) \exp\left(\widehat{\boldsymbol{\Omega}}(h)\right) \,, \quad (11)$$

$$\boldsymbol{\eta}(X + h) = \mathrm{Ad}^{-1}_{\exp\left(\widehat{\boldsymbol{\Omega}}(h)\right)} \left(\boldsymbol{\eta}(X) + \mathrm{T}_{\boldsymbol{\Omega}(h)}\dot{\boldsymbol{\Omega}}(h) \right) \,, \quad (12)$$

$$\begin{aligned} \dot{\boldsymbol{\eta}}(X + h) = {} & \mathrm{Ad}^{-1}_{\exp\left(\widehat{\boldsymbol{\Omega}}(h)\right)} \left(\dot{\boldsymbol{\eta}}(X) + \mathrm{ad}_{\boldsymbol{\eta}(X)}\mathrm{T}_{\boldsymbol{\Omega}(h)}\dot{\boldsymbol{\Omega}}(h) \right) \\ & + \mathrm{Ad}^{-1}_{\exp\left(\widehat{\boldsymbol{\Omega}}(h)\right)} \left(\mathrm{T}_{\boldsymbol{\Omega}(h)}\ddot{\boldsymbol{\Omega}}(h) + \dot{\mathrm{T}}_{\boldsymbol{\Omega}(h)}\dot{\boldsymbol{\Omega}}(h) \right) , \end{aligned} \quad (13)$$

where $\mathrm{T}_{\boldsymbol{\Omega}}$ is the tangent operator of the exponential map and $\dot{\mathrm{T}}_{\boldsymbol{\Omega}}$ is its time derivative. Both operators are given in closed form in Appendix B [17]. The main components of the GVS kinematics approach are summarized in Fig. 1.

B. Implicit Strain Parametrization

Now that the relationship between the strain and the pose configuration spaces is established, we proceed by parametrizing the strain field. In particular, we define the strain at any point along the arm with implicit equations of the form:

$$\boldsymbol{f}\left(\boldsymbol{\xi}(X), \boldsymbol{q}, X\right) = \mathbf{0} \in \mathbb{R}^6 \,, \quad (14)$$

where $\boldsymbol{q} \in \mathbb{R}^n$ collects the set of n generalized coordinates chosen to parameterize the system. According to the implicit function theorem [18], as long as the Jacobian $\partial \boldsymbol{f}/\partial \boldsymbol{\xi}(X)$ is not singular, (14) defines $\boldsymbol{\xi}(X)$ as a function of $\boldsymbol{q}$ and X. Furthermore, taking the time derivative of (14) and rearranging

the terms, we obtain the Jacobian of this implicit map, denoted by $\boldsymbol{\Phi}_\xi(\boldsymbol{q}, X) \in \mathbb{R}^{6\times n}$.

$$\dot{\boldsymbol{\xi}}(X) = -\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\xi}(X)}\right)^{-1} \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \dot{\boldsymbol{q}} = \boldsymbol{\Phi}_\xi \dot{\boldsymbol{q}} \,. \tag{15}$$

Taking another time derivative provides the strain acceleration model in the form:

$$\ddot{\boldsymbol{\xi}}(X) = \boldsymbol{\Phi}_\xi(\boldsymbol{q}, X)\ddot{\boldsymbol{q}} + \dot{\boldsymbol{\Phi}}_\xi(\boldsymbol{q}, \dot{\boldsymbol{q}}, X)\dot{\boldsymbol{q}} \,. \tag{16}$$

The foregoing strain parametrization can be used in the computation of the approximated Magnus expansion $\boldsymbol{\Omega}(h)$ and its time derivatives, which lead to an expression of the kinematic equations (11), (12), and (13) in terms of the generalized coordinates $\boldsymbol{q}$. In particular, the velocity and acceleration twists take the form:

$$\boldsymbol{\eta}(X) = \mathrm{Ad}^{-1}_{\boldsymbol{g}(X)} \int_0^X \mathrm{Ad}_{\boldsymbol{g}} \boldsymbol{\Phi}_\xi ds \dot{\boldsymbol{q}} = \boldsymbol{J}(\boldsymbol{q}, X)\dot{\boldsymbol{q}} \,, \tag{17}$$

$$\begin{aligned}\dot{\boldsymbol{\eta}}(X) &= \mathrm{Ad}^{-1}_{\boldsymbol{g}(X)} \int_0^X \mathrm{Ad}_{\boldsymbol{g}} \left(\boldsymbol{\Phi}_\xi \ddot{\boldsymbol{q}} + \dot{\boldsymbol{\Phi}}_\xi \dot{\boldsymbol{q}}\right) + \dot{\mathrm{Ad}}_{\boldsymbol{g}} \boldsymbol{\Phi}_\xi \dot{\boldsymbol{q}} ds \\ &= \boldsymbol{J}(\boldsymbol{q}, X)\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}(\boldsymbol{q}, \dot{\boldsymbol{q}}, X)\dot{\boldsymbol{q}} \,. \end{aligned} \tag{18}$$

An explicit recursive formula for the system Jacobian $\boldsymbol{J}$ and its time derivative $\dot{\boldsymbol{J}}$ in terms of $\boldsymbol{\Phi}_\xi$ and $\dot{\boldsymbol{\Phi}}_\xi$ is given in Appendix C for a specific fourth-order quadrature approximation $\boldsymbol{\Omega}(h)$.

C. Dynamics

The dynamics equations of the system in the new generalized coordinates $\boldsymbol{q}$ are obtained by D'Alembert's projection of the original Cosserat rod dynamic model through the system Jacobian $\boldsymbol{J}$. The Cosserat rod equilibrium is given by [19]:

$$\overline{\boldsymbol{\mathcal{M}}}\,\dot{\boldsymbol{\eta}} + \mathrm{ad}^*_{\boldsymbol{\eta}} \overline{\boldsymbol{\mathcal{M}}}\,\boldsymbol{\eta} = (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)' + \mathrm{ad}^*_{\boldsymbol{\xi}} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a) + \overline{\boldsymbol{\mathcal{F}}}_e \,, \tag{19}$$

where $\overline{\boldsymbol{\mathcal{M}}}(X) \in \mathbb{R}^{6\times 6}$ is the screw inertia density matrix of the cross-section, $\overline{\boldsymbol{\mathcal{F}}}_e(X) \in \mathbb{R}^6$ is the distributed external load, and $\boldsymbol{\mathcal{F}}_i(X), \boldsymbol{\mathcal{F}}_a(X) \in \mathbb{R}^6$ are the internal wrenches due to the material elasticity and distributed actuation respectively. For simplicity, we use a Hook-like linear visco-elastic constitutive law as follows:

$$\boldsymbol{\mathcal{F}}_i(X) = \boldsymbol{\Sigma}(\boldsymbol{\xi} - \boldsymbol{\xi}^*) + \boldsymbol{\Upsilon}\dot{\boldsymbol{\xi}} \,, \tag{20}$$

where $\boldsymbol{\Sigma}(X) \in \mathbb{R}^{6\times 6}$ is the screw elasticity matrix, $\boldsymbol{\Upsilon}(X) \in \mathbb{R}^{6\times 6}$ is the screw damping matrix, and $\boldsymbol{\xi}^*$ is defined by (3) for the stress-free reference configuration $\boldsymbol{g}^*$. Regarding the distributed actuation wrench, for a threadlike actuator, it can be modeled as [14]:

$$\boldsymbol{\mathcal{F}}_a(X) = \sum_{i=1}^{n_a} \begin{bmatrix} \tilde{\boldsymbol{d}}_i \boldsymbol{t}_i \\ \boldsymbol{t}_i \end{bmatrix} u_i = \boldsymbol{\Phi}_a \boldsymbol{u} \,, \tag{21}$$

where $\boldsymbol{\Phi}_a(\boldsymbol{q}, X) \in \mathbb{R}^{6\times n_a}$ is the actuation matrix, $\boldsymbol{d}_i(X) \in \mathbb{R}^3$ represents the distance of the i^{th} actuator to the robot arm mid-line, $\boldsymbol{t}_i(\boldsymbol{q}, X) \in \mathbb{R}^3$ is the unit vector tangent to the i^{th} actuator routing, and $\boldsymbol{u} \in \mathbb{R}^{n_a}$ is the vector of actuation inputs. The tangent vector $\boldsymbol{t}_i$ depends on the deformation of the soft manipulator as follows:

$$\boldsymbol{t}_i(\boldsymbol{q}, X) = \frac{\boldsymbol{t}_{c_i}}{\|\boldsymbol{t}_{c_i}\|} = \frac{\tilde{\mathbf{k}}\boldsymbol{d}_i + \mathbf{p} + \boldsymbol{d}_i'}{\|\tilde{\mathbf{k}}\boldsymbol{d}_i + \mathbf{p} + \boldsymbol{d}_i'\|} \,. \tag{22}$$

Applying D'Alembert's principle to the Cosserat dynamics (19) leads to the system's equations of motion in the standard Lagrangian form:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + (\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{D}(\boldsymbol{q}))\dot{\boldsymbol{q}} + \boldsymbol{K}(\boldsymbol{q}) = \boldsymbol{B}(\boldsymbol{q})\boldsymbol{u} + \boldsymbol{F}(\boldsymbol{q}) \,. \tag{23}$$

The full expression for each of the terms of the preceding equation is given in the Appendix D. Here, we propose a new formulation of the generalized external force $\boldsymbol{F}(\boldsymbol{q})$ that is interesting on its own and will be used in the next section. Applying the expression of the system's Jacobian as it appears in (17) to (40f), we obtain:

$$\begin{aligned}\boldsymbol{F}(\boldsymbol{q}) &= \int_0^L \boldsymbol{J}^T \overline{\boldsymbol{\mathcal{F}}}_e dX = \int_0^L \int_0^X \boldsymbol{\Phi}_\xi^T \mathrm{Ad}^{-*}_{\boldsymbol{g}} ds \mathrm{Ad}^*_{\boldsymbol{g}} \overline{\boldsymbol{\mathcal{F}}}_e dX \\ &= \int_0^L \boldsymbol{\Phi}_\xi^T \mathrm{Ad}^{-*}_{\boldsymbol{g}} \int_X^L \mathrm{Ad}^*_{\boldsymbol{g}} \overline{\boldsymbol{\mathcal{F}}}_e ds dX = \int_0^L \boldsymbol{\Phi}_\xi^T \boldsymbol{\Phi}_e dX \,, \end{aligned} \tag{24}$$

where we have used the following integrals property obtained from the integration by part $\int_0^L \int_0^X f(s) ds g(X) dX = \int_0^L f(X) \int_X^L g(s) ds dX$. In (24), we have defined

$$\boldsymbol{\Phi}_e(\boldsymbol{q}, X) = \mathrm{Ad}^{-*}_{\boldsymbol{g}(X)} \int_X^L \mathrm{Ad}^*_{\boldsymbol{g}} \overline{\boldsymbol{\mathcal{F}}}_e ds \,, \tag{25}$$

which represents the total external wrench applied to the cross-section at X.

III. Statics-Based Dynamics

In this section, we investigate a specific implicit strain parametrization arising from the manipulator's static equilibrium. A needed preliminary result is the integral form of the manipulator's static equilibrium. The robot's static equilibrium is governed by the following equations extracted from (19) and its boundary conditions.

$$(\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)' + \mathrm{ad}^*_{\boldsymbol{\xi}} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a) + \overline{\boldsymbol{\mathcal{F}}}_e = \mathbf{0}, \tag{26a}$$

$$\boldsymbol{\mathcal{F}}_i(L) - \boldsymbol{\mathcal{F}}_a(L) = \mathbf{0} \,. \tag{26b}$$

Exploiting the identity $(\mathrm{Ad}^*_{\boldsymbol{g}})' = \mathrm{Ad}^*_{\boldsymbol{g}} \mathrm{ad}^*_{\boldsymbol{\xi}}$, (26a) can be rewritten as

$$\left(\mathrm{Ad}^*_{\boldsymbol{g}} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)\right)' + \mathrm{Ad}^*_{\boldsymbol{g}} \overline{\boldsymbol{\mathcal{F}}}_e = \mathbf{0}. \tag{27}$$

Integrating from X to L and accounting for the boundary condition (26b), we obtain

$$\mathrm{Ad}^*_{\boldsymbol{g}} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)(X) = \int_X^L \mathrm{Ad}^*_{\boldsymbol{g}} \overline{\boldsymbol{\mathcal{F}}}_e ds. \tag{28}$$

Finally, making use of equations (20), (21), (25), and partially solving for $\boldsymbol{\xi}(X)$, yields:

$$\boldsymbol{\xi}(X) = \boldsymbol{\Sigma}^{-1} (\boldsymbol{\Phi}_a(\boldsymbol{\xi}(X))\boldsymbol{u} + \boldsymbol{\Phi}_e(\boldsymbol{\xi})) + \boldsymbol{\xi}^*. \tag{29}$$

Equation (29) is used in the following as implicit strain definition. We start with the actuators dominance condition, for which all the explicit computations are provided, and finally, we extend the results to include external forces.

A. Actuators Dominance Dynamics

Assuming $\boldsymbol{\Phi}_e \approx \mathbf{0}$, we propose the static-based implicit strain definition in the form (14) specified below.

$$\boldsymbol{\xi}(X) = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Phi}_a(\boldsymbol{\xi}(X))\boldsymbol{q}_a + \boldsymbol{\Phi}_p(X)\boldsymbol{q}_p\right) + \boldsymbol{\xi}^*, \tag{30}$$

where $\boldsymbol{\Phi}_p(X) \in \mathbb{R}^{6\times n_p}$. The preceding strain description ensures a perfect representation of the static equilibrium configurations while the degrees of freedom spanned by $\boldsymbol{\Phi}_p(X)$ approximate the deformation of the soft robot due to its dynamics. Applying (30) to the differential implicit strain formula (15), yields

$$\dot{\boldsymbol{\xi}}(X) = \left(\boldsymbol{I} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi}_a\right)^{-1}\boldsymbol{\Sigma}^{-1}\left[\boldsymbol{\Phi}_a\ \boldsymbol{\Phi}_p\right]\begin{bmatrix}\dot{\boldsymbol{q}}_a\\ \dot{\boldsymbol{q}}_p\end{bmatrix} = \boldsymbol{\Phi}_\xi\dot{\boldsymbol{q}}\ , \tag{31}$$

where $\boldsymbol{q} = [\boldsymbol{q}_a^T\ \boldsymbol{q}_p^T]^T \in \mathbb{R}^{n_a+n_p}$ and

$$\boldsymbol{\Psi}_a(\boldsymbol{\xi}(X)) = \frac{\partial \boldsymbol{\Phi}_a\boldsymbol{q}_a}{\partial \boldsymbol{\xi}(X)} = \sum_{i=1}^{n_a}\frac{q_{a_i}}{\|\boldsymbol{t}_{c_i}\|}\begin{bmatrix}\tilde{\boldsymbol{d}}_i\tilde{\boldsymbol{t}}_i^2\tilde{\boldsymbol{d}}_i & -\tilde{\boldsymbol{d}}_i\tilde{\boldsymbol{t}}_i^2\\ \tilde{\boldsymbol{t}}_i^2\tilde{\boldsymbol{d}}_i & -\tilde{\boldsymbol{t}}_i^2\end{bmatrix}. \tag{32}$$

Taking another time derivative, we obtain

$$\begin{aligned}\ddot{\boldsymbol{\xi}}(X) &= \boldsymbol{\Phi}_\xi\ddot{\boldsymbol{q}} + \left(\boldsymbol{I} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi}_a\right)^{-1}\boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Psi}_{\dot{a}} + \dot{\boldsymbol{\Psi}}_a\right)\boldsymbol{\Phi}_\xi\dot{\boldsymbol{q}}\\ &= \boldsymbol{\Phi}_\xi\ddot{\boldsymbol{q}} + \dot{\boldsymbol{\Phi}}_\xi\dot{\boldsymbol{q}}\ .\end{aligned} \tag{33}$$

The expressions for $\boldsymbol{\Psi}_{\dot{a}}$ and $\dot{\boldsymbol{\Psi}}_a$ are given in Appendix E. Once the Jacobians $\boldsymbol{\Phi}_\xi$ and $\dot{\boldsymbol{\Phi}}_\xi$ are found, we can follow the procedure outlined in Section II to establish the dynamic model of the soft robot parametrized by the static-based implicit strain definition (30). It is worth noticing that at the steady state, we obtain

$$\int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_a dX\boldsymbol{q}_a + \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_p dX\boldsymbol{q}_p = \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_a dX\boldsymbol{u} \tag{34}$$

whose solution is, as expected, $\boldsymbol{q} = [\boldsymbol{u}^T\ \mathbf{0}^T]^T$, while, if $n_p = 0$, then $\boldsymbol{q} = \boldsymbol{u}$ as in [15].

B. External Forces Extension

The static-based implicit strain definition (30) can be extended to the case of non-negligible external forces by including $\boldsymbol{\Phi}_e$ from (29).

$$\boldsymbol{\xi}(X) = \boldsymbol{\Sigma}^{-1}\left(\boldsymbol{\Phi}_a(\boldsymbol{\xi}(X))\boldsymbol{q}_a + \boldsymbol{\Phi}_p(X)\boldsymbol{q}_p + \boldsymbol{\Phi}_e(\boldsymbol{\xi})\right) + \boldsymbol{\xi}^*. \tag{35}$$

Taking the time derivatives of the foregoing strain parametrization, we formally obtain the strain rates as in (15) and (16). Unfortunately, for general external forces, it is hard to analytically compute the Jacobians $\boldsymbol{\Phi}_\xi$ and $\dot{\boldsymbol{\Phi}}_\xi$ as we did for the actuators dominance condition. This would involve the computation of the Jacobian of the static equilibrium (29). In the literature, similar quantities have been approximated through machine-learning approaches, as, for example, in [20] and [21]. Once $\boldsymbol{\Phi}_\xi$ and $\dot{\boldsymbol{\Phi}}_\xi$ are found, the dynamic model of the soft robot can be established following the procedure in Section II. It is worth highlighting that the resulting dynamical system is forced to live on the static equilibrium manifold generated by $\boldsymbol{\Phi}_\xi$, including some extra freedom due to $\boldsymbol{\Phi}_p$. Indeed, using (35) in (40d) and (24), at the steady state we obtain

$$\int_0^L \boldsymbol{\Phi}_\xi^T\left[\boldsymbol{\Phi}_a\ \boldsymbol{\Phi}_p\right]dX\boldsymbol{q} + \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_e dX$$

TABLE I
ACTUATOR ROUTING PARAMETERS. $\lambda = 4\pi\ m^{-1}$ AND $r_m = (r_b + r_t)/2$

Number	Actuator range (m)	y-Coordinate (m)	z-Coordinate (m)
1	$[0,\ 1]$	$r(X)$	0
2	$[0,\ 1]$	$-r(X)\sin(\pi/6)$	$r(X)\cos(\pi/6)$
3	$[0,\ 1]$	$-r(X)\sin(\pi/6)$	$-r(X)\cos(\pi/6)$
4	$[0,\ 1]$	$r_t\sin(\lambda X)$	$r_t\cos(\lambda X)$
5	$[0,\ 1]$	$r_t\sin(\lambda X)$	$-r_t\cos(\lambda X)$
6	$[0,\ 0.5]$	$-r(X)$	0
7	$[0,\ 0.5]$	$r(X)\sin(\pi/6)$	$-r(X)\cos(\pi/6)$
8	$[0,\ 0.5]$	$r(X)\sin(\pi/6)$	$r(X)\cos(\pi/6)$
9	$[0,\ 0.5]$	$r_m\sin(\pi/2+\lambda X)$	$r_m\cos(\pi/2+\lambda X)$
10	$[0,\ 0.5]$	$-r_m\sin(\pi/2+\lambda X)$	$r_m\cos(\pi/2+\lambda X)$

$$= \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_a dX\boldsymbol{u} + \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_e dX \tag{36}$$

The terms involving $\boldsymbol{\Phi}_e$ cancel out on both sides of the equation, leading, once again, to $\boldsymbol{q} = [\boldsymbol{u}^T\ \mathbf{0}^T]^T$, or $\boldsymbol{q} = \boldsymbol{u}$ if $n_p = 0$.

C. Validation

We constructed a 1-meter-long soft manipulator (Fig. 2(a)), which consists of two 0.5-meter-long segments. The material properties of both segments are: a Young modulus of 1 MPa, a density of 100 kg/m^3, a poison ratio of 0.5, and an elastic damping factor of 10 KPas. The manipulator's radius $r(X)$ linearly decreases from $r_b = 3$ cm at the base to $r_t = 2$ cm at the tip. The manipulator is equipped with 10 linearly independent actuators, with 5 of them extending from the base to the tip of the manipulator, while the remaining 5 terminate at the midpoint of the rod. The threadlike actuators routing parameters are defined in Table I and consist of 6 linear and 4 spiral actuators (the y- and z-axis span the cross-sectional plane).

We analyze two cases using the implicit strain parameterization defined in (30): one considering $\boldsymbol{\Phi}_p(X) = \mathbf{0}$, and another one we defined a piece-wise constant angular strain basis (torsion and bending), $\boldsymbol{\Phi}_p(X) = [\boldsymbol{I}^{\mathbf{3\times 3}}, \mathbf{0}^{\mathbf{3\times 3}}]^T$, to capture remaining dynamics of the system. This additional basis can introduce constant angular strains in the first and the second half of the manipulator. We selected simple constant angular strain bases in order to maintain a dominant role of the implicit parameterization in the motion prediction while improving the accuracy during the transient dynamics. Other more complex functions could be selected instead. To summarize, we compare a fully actuated system with 10 DOFs and 10 actuators and an underactuated system with 16 DoOFs. We validate the proposed formulation against explicit cubic Legendre polynomial basis for each strain and for each division (very high order model) for a total of 48 DOFs. This standard model is implemented and solved using the MATLAB toolbox SoRoSim [22].

For the proposed implicit strain parametrization (30), the first step is to solve for the values of strains at certain values of X determined by the quadrature approximations of the Magnus expansion and the integration scheme. For a given value of X, $\boldsymbol{q}_a$, and $\boldsymbol{q}_p$, the equation can be solved using a root-finding method. We employed the Newton-Raphson method utilizing the Jacobian of the equation with respect to $\boldsymbol{\xi}(X)$, given by: $\boldsymbol{I} - \boldsymbol{\Sigma}^{-1}\boldsymbol{\Psi}_a$. With $\boldsymbol{\xi}(X)$, we obtain the values of $\boldsymbol{\Phi}_\xi$ and $\dot{\boldsymbol{\Phi}}_\xi$ needed for the recursive computation of the geometric Jacobian $\boldsymbol{J}$ and its derivative $\dot{\boldsymbol{J}}$. Using $\boldsymbol{J}$ and $\dot{\boldsymbol{J}}$, we compute the coefficients of the Lagrangian form (23) by a numerical integration

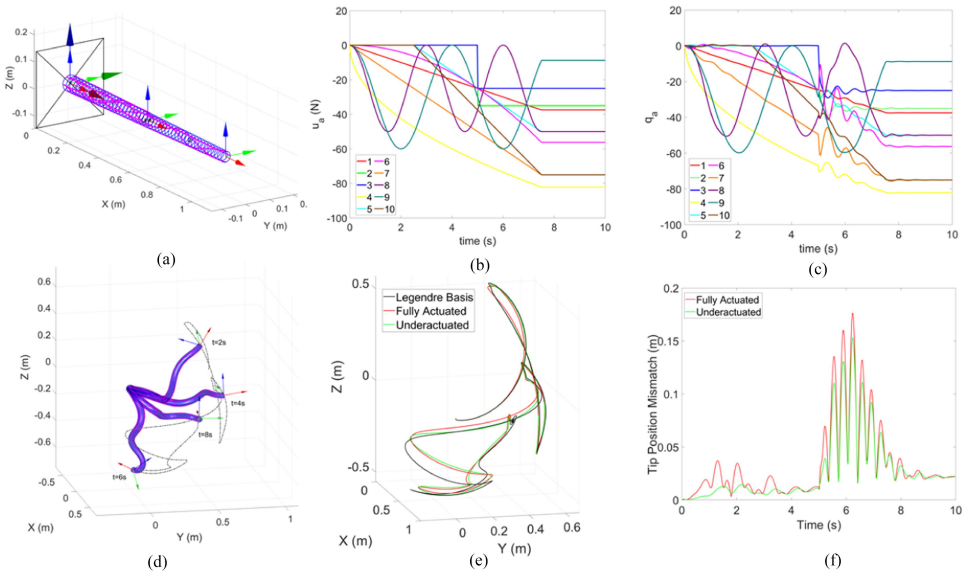

Fig. 2. (a) Soft manipulator with 10 actuators. (b) Arbitrary input actuator strengths. (c) Time evolution of the states of the fully actuated system. (d) Snapshots of corresponding shapes with the tip trajectory in dotted lines. (e) Tip trajectory comparison with respect to a standard model with a Legendre polynomial base. (f) Tip position mismatch of fully actuated and underactuated system with respect to that of the standard model.

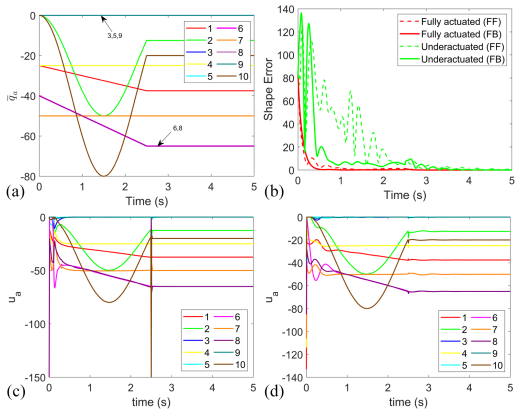

Fig. 3. (a) Desired shape of the manipulator. (b) Shape error vs time. Here FF and FB indicate feedforward and feedback control laws. (c) and (d) show the input actuator strengths for the fully actuated and the underactuated systems, respectively.

along the soft body. We used the Gauss quadrature scheme for this space integration. Finally, the system dynamics are solved using explicit numerical integrators implemented by $ode45$ or $ode15s$ MATLAB functions.

We applied arbitrary actuator strengths for each actuator as shown in Fig. 2(b). From 0 to 7.5 s, the cable strengths are varied, and after 7.5 s, it is held constant. To introduce a significant level of dynamic activity, we implemented a step load on actuators 2 and 3 at the 5-second mark. The dynamic evolution of the state of the fully actuated system ($\boldsymbol{q}_a$) is shown in Fig. 2(c), and the corresponding soft robot motion is shown in Fig. 2(d). The results are similar compared to the high-order standard model as highlighted in Fig. 2(e), where the tip positions are plotted. This demonstrates the ability of the proposed approach to satisfactorily represent the system configuration manifold using no more degrees of freedom than the number of actuators, with an almost 5x dimension reduction for this specific example. The accuracy is further improved with the additional DOFs spanned by $\boldsymbol{\Phi}_p$ as shown by comparing the tip position error in Fig. 2(f), especially during the step load at around 5 s. The reader may also refer to the supplementary video to compare the simulation results.

IV. Statics-Based Dynamic Control

In this section, we develop a shape and tip-pose controller for the static-based reduced-order model presented in Section III-A. We present the results obtained through employing the two implicit strain parametrizations defined earlier in the validation Section III-C.

A. Shape Control

The equilibrium configurations are defined by the solution of equation (34). The control input in the equilibrium is defined as $\boldsymbol{q} = [\boldsymbol{u}^T \ \boldsymbol{0}^T]^T$, while, if $n_p = 0$, then $\boldsymbol{q} = \boldsymbol{u}$. Then, following the procedure in [23], we propose a controller that combines a feedforward term ($\boldsymbol{u} = \overline{\boldsymbol{q}}_a$) with a PD controller to reject disturbances or parameter uncertainties. In our special scenario with full actuation, the following controller is obtained:

$$\boldsymbol{u} = \overline{\boldsymbol{q}} + k_d(\dot{\overline{\boldsymbol{q}}} - \dot{\boldsymbol{q}}) + k_p(\overline{\boldsymbol{q}} - \boldsymbol{q}). \tag{37}$$

Whereas, for the under-actuated case $n_p > 0$, the above equation is modified as follows:

$$\boldsymbol{u} = \overline{\boldsymbol{q}}_a + k_d \boldsymbol{B}^T \left(\begin{bmatrix} \dot{\overline{\boldsymbol{q}}}_a \\ \boldsymbol{0} \end{bmatrix} - \dot{\boldsymbol{q}} \right) + k_p \boldsymbol{B}^T \left(\begin{bmatrix} \overline{\boldsymbol{q}}_a \\ \boldsymbol{0} \end{bmatrix} - \boldsymbol{q} \right). \tag{38}$$

Fig. 3(a) shows the desired statically actuated DOFs where $\overline{\boldsymbol{q}}_a$ is varied from 0 to 2.5 s and afterward kept constant. We performed two sets of simulations for the underactuated and the fully actuated systems: one with the feedforward (FF) control law ($\boldsymbol{u} = \overline{\boldsymbol{q}}_a$) and one with the feedback controllers specified by equations (37) and (38). We used a value of $k_p = 1000$ and $k_d = 100$ for both controllers. The values of

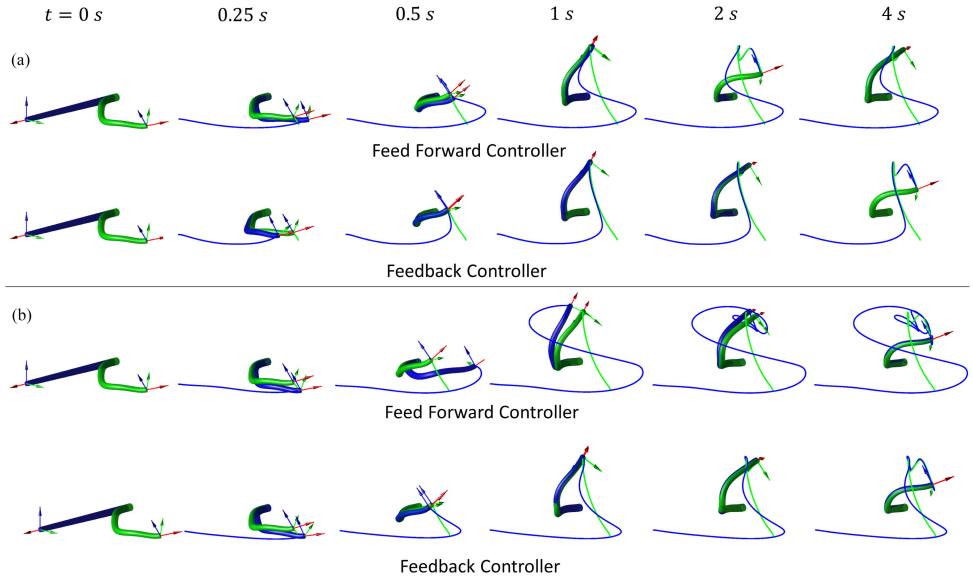

Fig. 4. Snapshots of dynamics simulation for the fully actuated (a) and the underactuated (b) systems. The green body indicates the desired shape, while the blue body is the actual shape of the manipulator. The green and the blue lines indicate the corresponding tip trajectories.

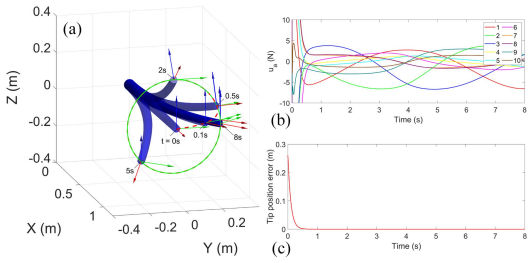

Fig. 5. Task space control of fully actuated system: (a) Snapshots of dynamics, (b) controlled actuator strengths, and (c) Tip position error vs. time.

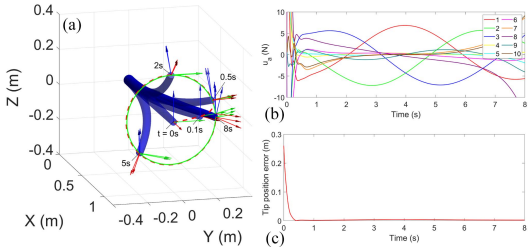

Fig. 6. Task space control of underactuated system: (a) Snapshots of dynamics, (b) controlled actuator strengths, and (c) Tip position error vs. time.

actuator strengths for fully actuated and underactuated systems are shown in Fig. 3(c) and (d). The snapshots of the dynamic simulation results are shown in Fig. 4. The specified feedback controllers lead to fast convergence. To quantify this, we plot the shape error over time in Fig. 3(b) where the shape error is calculated as $\|[\overline{\boldsymbol{q}}_a^T \ \boldsymbol{0}^{n_p\times 1}]^T - \boldsymbol{q}\|$. To further appreciate the controller's performance, the reader may refer to the supplementary video.

B. Tip-Pose Control

The tip-pose controller can be obtained by computing the generalized force corresponding to a feedback linearization controller designed on the task-space dynamic model. Following the procedure in [11], we obtain

$$\boldsymbol{u} = \boldsymbol{B}^{\dagger}\boldsymbol{J}_t^T\boldsymbol{\Omega}\cdot\Big[\dot{\overline{\boldsymbol{\eta}}}_t + k_d(\overline{\boldsymbol{\eta}}_t - \boldsymbol{\eta}_t) + k_p(\log(\boldsymbol{g}_t^{-1}\overline{\boldsymbol{g}}_t))^{\vee} + \boldsymbol{J}_t\boldsymbol{M}^{-1}((\boldsymbol{C}+\boldsymbol{D})\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} - \boldsymbol{F}) - \dot{\boldsymbol{J}}_t\dot{\boldsymbol{q}}\Big]. \quad (39)$$

where $\boldsymbol{J}_t = \boldsymbol{J}(L)$, $\boldsymbol{\Omega} = (\boldsymbol{J}_t\boldsymbol{M}^{-1}\boldsymbol{J}_t^T)^{-1}$ is the task-space mass matrix, $\overline{\boldsymbol{g}}_t(t)$ and $\boldsymbol{g}_t(t)$ represent the desired and the actual tip pose, and $k_p = 10,000$ and $k_d = 1,000$ are the proportional and derivative gains used in the control law. The logarithmic map (log) in $SE(3)$ is defined in Appendix B. For the fully actuated system, the pseudoinverse of $\boldsymbol{B}$ becomes simply $\boldsymbol{B}^{-1}$. $\overline{\boldsymbol{\eta}}_t$ and $\dot{\overline{\boldsymbol{\eta}}}_t$ are the desired tip velocity and acceleration expressed in the actual tip frame, given by:

$$\widehat{\overline{\boldsymbol{\eta}}} = \overline{\boldsymbol{g}}_t^{-1}\dot{\overline{\boldsymbol{g}}}_t \qquad \overline{\boldsymbol{\eta}}_t = \mathrm{Ad}_{\boldsymbol{g}_t^{-1}\overline{\boldsymbol{g}}_t}\overline{\boldsymbol{\eta}}$$

$$\dot{\overline{\boldsymbol{\eta}}}_t = \mathrm{ad}_{(\overline{\boldsymbol{\eta}}_t - \boldsymbol{\eta}_t)}\mathrm{Ad}_{\boldsymbol{g}_t^{-1}\overline{\boldsymbol{g}}_t}\overline{\boldsymbol{\eta}} + \mathrm{Ad}_{\boldsymbol{g}_t^{-1}\overline{\boldsymbol{g}}_t}\dot{\overline{\boldsymbol{\eta}}}$$

We defined the desired tip pose trajectory as follows. The desired tip frame origin follows a circular path with a radius of 0.25 m, parallel to the yz-plane. The center of the circle is located at 0.95 m along the x-axis. The x-axis of the reference frame is oriented at 37.8° degrees relative to the global x-axis. Figs. 5 and 6 detail the simulation results of the fully actuated and underactuated systems, respectively. These figures also show the tension of the actuator strengths (control inputs) needed to perform the desired trajectory tracking and the error at the tip. The dynamics of the system are effectively canceled out by the controller, leading to a rapid convergence of the error between the actual and reference frames. The initial high and fluctuating actuator strengths during the simulation highlight the

process of counteracting the system's dynamic response. After this, the actuator strengths smoothly evolve to track the circular trajectory. For both the fully actuated and underactuated cases, the tracking error is very small, with slightly better performance in the former, as can be seen from Figs. 5(c) and 6(c) at around 4 seconds. The results for both designs are reported in the video companion to this manuscript.

V. Conclusions and Future Work

In conclusion, an implicit parametrization of the geometric strain-based approach to soft robotic modeling has been proposed, allowing to reduce the number of DOFs without compromising accuracy. In particular, we employed an integral form of the soft robot static equilibrium as implicit strain parametrization, which forces the system to evolve on the reduced manifold of static configuration or a slightly enlarged one if needed. Such a reduced model was proven to be accurate against high-order models despite having only as many DOFs as the number of actuators. Finally, a shape controller and a tip-pose (task-space) controller were designed and adapted to the static-based implicit parametrization model. The results show an excellent error convergence for both the fully actuated and the underactuated cases. It is worth highlighting that, in a physical system, the controller implementation would be slightly different with respect to the simulated case. In fact, instead of reconstructing the strain values from the generalized coordinates $\boldsymbol{q}$, the strain field $\boldsymbol{\xi}$ and its time derivative $\dot{\boldsymbol{\xi}}$ can be directly measured or estimated. With that, all the matrices and vectors required to implement the controllers (38), (39) can be obtained. Also, for a given soft robot design, the integral form of the static strain equilibrium and its Jacobian could be learned from data, making the proposed approach more efficient and accurate for real applications. This will be the subject of future works.

Appendix A

$$\mathrm{Ad}_{\boldsymbol{g}} = \begin{pmatrix} \boldsymbol{R} & \boldsymbol{0} \\ \widetilde{\mathbf{r}}\boldsymbol{R} & \boldsymbol{R} \end{pmatrix}, \mathrm{Ad}^*_{\boldsymbol{g}} = \begin{pmatrix} \boldsymbol{R} & \widetilde{\mathbf{r}}\boldsymbol{R} \\ \boldsymbol{0} & \boldsymbol{R} \end{pmatrix},$$
$$\mathrm{ad}_{\boldsymbol{\xi}} = \begin{pmatrix} \widetilde{\mathbf{k}} & \boldsymbol{0} \\ \widetilde{\mathbf{u}} & \widetilde{\mathbf{k}} \end{pmatrix}, \mathrm{ad}^*_{\boldsymbol{\xi}} = \begin{pmatrix} \widetilde{\mathbf{k}} & \widetilde{\mathbf{u}} \\ \boldsymbol{0} & \widetilde{\mathbf{k}} \end{pmatrix}$$

Appendix B

Exponential map:

$$\exp\left(\widehat{\boldsymbol{\Omega}}\right) = \boldsymbol{I}_4 + \widehat{\boldsymbol{\Omega}} + \frac{(1-\cos(\theta))}{\theta^2}\widehat{\boldsymbol{\Omega}}^2 + \frac{(\theta-\sin(\theta))}{\theta^3}\widehat{\boldsymbol{\Omega}}^3.$$

where, θ is $\|\mathbf{k}\|$.

Logarithmic map:

$$\begin{aligned}\log(\boldsymbol{g}) = \frac{1}{8}\csc^3\left(\frac{\theta}{2}\right)\sec\left(\frac{\theta}{2}\right)\big[&(\theta\cos(2\theta)-\sin(X\theta))\,\boldsymbol{I}_4 \\ &-(\theta\cos(\theta)+2\theta\cos(2\theta)-\sin(\theta)-\sin(2\theta))\,\boldsymbol{g} \\ &+(2\theta\cos(\theta)+\theta\cos(2\theta)-\sin(\theta)-\sin(2\theta))\,\boldsymbol{g}^2 \\ &-(\theta\cos(\theta)-\sin(\theta))\,\boldsymbol{g}^3\big]\end{aligned}$$

where θ satisfies $Tr(\boldsymbol{g}) = 2(1+\cos(\theta))$ for $-\pi < \theta < \pi$.

Tangent operator of the exponential map:

$$\begin{aligned}\mathrm{T}_{\boldsymbol{\Omega}} &= \int_0^1 \exp\left(s\cdot\mathrm{ad}_{\boldsymbol{\Omega}}\right) ds \\ &= \boldsymbol{I}_6 + \frac{1}{2\theta^2}\left(4-4\cos(\theta)-\theta\sin(\theta)\right)\mathrm{ad}_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^3}\left(4\theta-5\sin(\theta)+\theta\cos(\theta)\right)\mathrm{ad}^2_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^4}\left(2-2\cos(\theta)-\theta\sin(\theta)\right)\mathrm{ad}^3_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^5}\left(2\theta-3\sin(\theta)+\theta\cos(\theta)\right)\mathrm{ad}^4_{\boldsymbol{\Omega}},\end{aligned}$$

Time derivative of the tangent operator:

$$\begin{aligned}\dot{\mathrm{T}}_{\boldsymbol{\Omega}} &= \int_0^1 \frac{\partial \exp\left(s\cdot\mathrm{ad}_{\boldsymbol{\Omega}}\right)}{\partial t} ds \\ &= \frac{\dot\theta}{2\theta^3}\left(-8+(8-\theta^2)\cos(\theta)+5\theta\sin(\theta)\right)\mathrm{ad}_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^2}\left(4-4\cos(\theta)-\theta\sin(\theta)\right)\left(\dot{\mathrm{ad}}_{\boldsymbol{\Omega}}\right) \\ &\quad + \frac{\dot\theta}{2\theta^4}\left(-8\theta+(15-\theta^2)\sin(\theta)-7\theta\cos(\theta)\right)\mathrm{ad}^2_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^3}\left(4\theta-5\sin(\theta)+\theta\cos(\theta)\right)\left(\dot{\mathrm{ad}}^2_{\boldsymbol{\Omega}}\right) \\ &\quad + \frac{\dot\theta}{2\theta^5}\left(-8+(8-\theta^2)\cos(\theta)+5\theta\sin(\theta)\right)\mathrm{ad}^3_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^4}\left(2-2\cos(\theta)-\theta\sin(\theta)\right)\left(\dot{\mathrm{ad}}^3_{\boldsymbol{\Omega}}\right) \\ &\quad + \frac{\dot\theta}{2\theta^6}\left(-8\theta+(15-\theta^2)\sin(\theta)-7\theta\cos(\theta)\right)\mathrm{ad}^4_{\boldsymbol{\Omega}} \\ &\quad + \frac{1}{2\theta^5}\left(2\theta-3\sin(\theta)+\theta\sin(\theta)\right)\left(\dot{\mathrm{ad}}^4_{\boldsymbol{\Omega}}\right)\end{aligned}$$

Appendix C

Considering the strain parametrization (14), (15), (16), and a fourth-order Zanna's quadrature [24] we obtain:

$$\begin{aligned}\boldsymbol{\Omega}(h) &= \frac{h}{2}\left(\boldsymbol{\xi}_{z1}+\boldsymbol{\xi}_{z2}\right) + \frac{\sqrt{3}h^2}{12}\mathrm{ad}_{\boldsymbol{\xi}_{z1}}\boldsymbol{\xi}_{z2}\,, \\ \dot{\boldsymbol{\Omega}}(h) &= \mathcal{Z}_h\left(\boldsymbol{\Phi}_\xi\right)\dot{\boldsymbol{q}}\,, \\ \ddot{\boldsymbol{\Omega}}(h) &= \mathcal{Z}_h\left(\boldsymbol{\Phi}_\xi\right)\ddot{\boldsymbol{q}} + \mathcal{Z}_h\left(\dot{\boldsymbol{\Phi}}_\xi\right)\dot{\boldsymbol{q}} + \frac{\sqrt{3}h^2}{6}\mathrm{ad}_{\boldsymbol{\Phi}_{\xi_{z1}}\dot{\boldsymbol{q}}}\boldsymbol{\Phi}_{\xi_{z2}}\dot{\boldsymbol{q}}\,,\end{aligned}$$

where the subscripts $_{z1}$, $_{z2}$ refer to the quadrature point in the interval $[X, X+h]$, and the operator $\mathcal{Z}_h$ is defined as:

$$\mathcal{Z}_h\left(\boldsymbol{Q}\right) = \frac{h}{2}\left(\boldsymbol{Q}_{z1}+\boldsymbol{Q}_{z2}\right) + \frac{\sqrt{3}h^2}{12}\left(\mathrm{ad}_{\boldsymbol{\xi}_{z1}}\boldsymbol{Q}_{z2} - \mathrm{ad}_{\boldsymbol{\xi}_{z2}}\boldsymbol{Q}_{z1}\right)$$

The recursive computation of the Jacobian and its derivative become:

$$\boldsymbol{J}(X+h) = \mathrm{Ad}^{-1}_{\exp\left(\widehat{\boldsymbol{\Omega}}(h)\right)}\left(\boldsymbol{J}(X) + \mathrm{T}_{\boldsymbol{\Omega}(h)}\mathcal{Z}_h\left(\boldsymbol{\Phi}_\xi\right)\right),$$

$$\dot{\boldsymbol{J}}(X+h) = \mathrm{Ad}^{-1}_{\exp\left(\widehat{\boldsymbol{\Omega}}(h)\right)}\Bigg(\dot{\boldsymbol{J}}(X) + \left(\mathrm{ad}_{\boldsymbol{\eta}(X)}\mathrm{T}_{\boldsymbol{\Omega}(h)} + \dot{\mathrm{T}}_{\boldsymbol{\Omega}(h)}\right)\mathcal{Z}_h\left(\boldsymbol{\Phi}_\xi\right) + \mathrm{T}_{\boldsymbol{\Omega}(h)}\Bigg(\mathcal{Z}_h\left(\dot{\boldsymbol{\Phi}}_\xi\right) + \frac{\sqrt{3}h^2}{6}\mathrm{ad}_{\boldsymbol{\Phi}_{\xi_{z1}}\dot{\boldsymbol{q}}}\boldsymbol{\Phi}_{\xi_{z2}}\Bigg)\Bigg),$$

APPENDIX D

Terms and coefficients of the Lagrangian dynamics equation.

$$\boldsymbol{M}(\boldsymbol{q}) = \int_0^L \boldsymbol{J}^T\overline{\boldsymbol{\mathcal{M}}}\,\boldsymbol{J}dX \tag{40a}$$

$$\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}}) = \int_0^L \boldsymbol{J}^T\left(\mathrm{ad}^*_{\boldsymbol{\eta}}\overline{\boldsymbol{\mathcal{M}}}\,\boldsymbol{J} + \overline{\boldsymbol{\mathcal{M}}}\,\dot{\boldsymbol{J}}\right)dX \tag{40b}$$

$$\boldsymbol{D}(\boldsymbol{q}) = \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Upsilon}\boldsymbol{\Phi}_\xi dX \tag{40c}$$

$$\boldsymbol{K}(\boldsymbol{q}) = \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Sigma}\left(\boldsymbol{\xi}-\boldsymbol{\xi}^*\right)dX \tag{40d}$$

$$\boldsymbol{B}(\boldsymbol{q}) = \int_0^L \boldsymbol{\Phi}_\xi^T\boldsymbol{\Phi}_a dX \tag{40e}$$

$$\boldsymbol{F}(\boldsymbol{q}) = \int_0^L \boldsymbol{J}^T\overline{\boldsymbol{\mathcal{F}}}_e dX \tag{40f}$$

APPENDIX E

A lengthy but straightforward calculation yields

$$\boldsymbol{\Psi}_{\dot{a}}(\boldsymbol{\xi}(X)) = \frac{\partial\boldsymbol{\Phi}_a\dot{\boldsymbol{q}}_a}{\partial\boldsymbol{\xi}(X)} = \sum_{i=1}^{n_a}\frac{\dot{q}_{a_i}}{\|\boldsymbol{t}_{c_i}\|}\begin{bmatrix}\tilde{\boldsymbol{d}}_i\tilde{\boldsymbol{t}}_i^{\,2}\tilde{\boldsymbol{d}}_i & -\tilde{\boldsymbol{d}}_i\tilde{\boldsymbol{t}}_i^{\,2}\\ \tilde{\boldsymbol{t}}_i^{\,2}\tilde{\boldsymbol{d}}_i & -\tilde{\boldsymbol{t}}_i^{\,2}\end{bmatrix},$$

$$\dot{\boldsymbol{\Psi}}_a = \sum_{i=1}^{n_a}\begin{bmatrix}\tilde{\boldsymbol{d}}_i\\ \boldsymbol{I}\end{bmatrix}\left(\frac{\dot{q}_{a_i}\|\boldsymbol{t}_{c_i}\| - q_{a_i}\boldsymbol{t}_i^T\dot{\boldsymbol{t}}_{c_i}}{\|\boldsymbol{t}_{c_i}\|}\tilde{\boldsymbol{t}}_i^{\,2}\right)\begin{bmatrix}\tilde{\boldsymbol{d}}_i & -\boldsymbol{I}\end{bmatrix} + \sum_{i=1}^{n_a}\begin{bmatrix}\tilde{\boldsymbol{d}}_i\\ \boldsymbol{I}\end{bmatrix}\left(\frac{q_{a_i}}{\|\boldsymbol{t}_{c_i}\|}\frac{d\tilde{\boldsymbol{t}}_i^{\,2}}{dt}\right)\begin{bmatrix}\tilde{\boldsymbol{d}}_i & -\boldsymbol{I}\end{bmatrix}.$$

To complete the computation, we provide

$$\dot{\boldsymbol{t}}_i = -\frac{\tilde{\boldsymbol{t}}_i^{\,2}}{\|\boldsymbol{t}_{c_i}\|}\dot{\boldsymbol{t}}_{c_i}, \quad \dot{\boldsymbol{t}}_{c_i} = \begin{bmatrix}-\tilde{\boldsymbol{d}}_i & \boldsymbol{I}\end{bmatrix}\boldsymbol{\Phi}_\xi\dot{\boldsymbol{q}}.$$

REFERENCES

[1] C. Laschi, B. Mazzolai, and M. Cianchetti, "Soft robotics: Technologies and systems pushing the boundaries of robot abilities," *Sci. Robot.*, vol. 1, no. 1, 2016, Art. no. eaah3690.

[2] C. Armanini, F. Boyer, A. T. Mathew, C. Duriez, and F. Renda, "Soft robots modeling: A structured overview," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1728–1748, Jun. 2023.

[3] P. Schegg and C. Duriez, "Review on generic methods for mechanical modeling, simulation and control of soft robots," *PLos One*, vol. 17, no. 1, pp. 1–14, 2022.

[4] O. Goury and C. Duriez, "Fast, generic, and reliable control and simulation of soft robots using model order reduction," *IEEE Trans. Robot.*, vol. 34, no. 6, pp. 1565–1576, Dec. 2018.

[5] M. Tummers, V. Lebastard, F. Boyer, J. Troccaz, B. Rosa, and M. T. Chikhaoui, "Cosserat rod modeling of continuum robots from newtonian and lagrangian perspectives," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 2360–2378, Jun. 2023.

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

[7] F. Boyer, V. Lebastard, F. Candelier, F. Renda, and M. Alamir, "Statics and dynamics of continuum robots based on Cosserat rods and optimal control theories," *IEEE Trans. Robot.*, vol. 39, no. 2, pp. 1544–1562, Apr. 2023.

[8] F. Renda, C. Armanini, V. Lebastard, F. Candelier, and F. Boyer, "A geometric variable-strain approach for static modeling of soft manipulators with tendon and fluidic actuation," *IEEE Robot. Automat. Lett.*, vol. 5, no. 3, pp. 4006–4013, Jul. 2020.

[9] F. Boyer, V. Lebastard, F. Candelier, and F. Renda, "Dynamics of continuum and soft robots: A strain parameterization based approach," *IEEE Trans. Robot.*, vol. 37, no. 3, pp. 847–863, Jun. 2021.

[10] R. 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.

[11] K. M. Lynch and F. C. Park, *Modern Robotics: Mechanics, Planning, and Control*. Cambridge, U.K.: Cambridge Univ. Press, 2017.

[12] M. Gazzola, L. H. Dudte, A. G. McCormick, and L. Mahadevan, "Forward and inverse problems in the mechanics of soft filaments," *Roy. Soc. Open Sci.*, vol. 5, no. 6, Art. no. 171628, 2018.

[13] C. Laschi, T. G. Thuruthel, F. Lida, R. Merzouki, and E. Falotico, "Learning-based control strategies for soft robots: Theory, achievements, and future challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 100–113, Jun. 2023.

[14] F. Renda, M. Cianchetti, H. Abidi, J. Dias, and L. Seneviratne, "Screw-based modeling of soft manipulators with tendon and fluidic actuation," *J. Mechanism Robot.*, vol. 9, no. 4, 2017, doi: 10.1115/1.4036579.

[15] F. Renda, C. Armanini, A. Mathew, and F. Boyer, "Geometrically-exact inverse kinematic control of soft manipulators with general threadlike actuators' routing," *IEEE Robot. Automat. Lett.*, vol. 7, no. 3, pp. 7311–7318, Jul. 2022.

[16] E. Hairer, C. Lubich, and G. Wanner, *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations* (Springer Series in Computational Mathematics). Berlin, Germany: Springer, 2006.

[17] J. M. Selig, *Geometric Fundamentals of Robotics. Monographs in Computer Science*. Berlin, Germany: Springer, 2007.

[18] T. M. Apostol, *Mathematical Analysis* (Addison-Wesley Series in Mathematics). Boston, U.K.: Addison-Wesley, 1974.

[19] F. Boyer and F. Renda, "Poincare's equations for Cosserat media: Application to shells," *J. Nonlinear Sci.*, vol. 27, pp. 1–44, 2016.

[20] G. Fang, Y. Tian, Z.-X. Yang, J. M. P. Geraedts, and C. C. L. Wang, "Efficient jacobian-based inverse kinematics with sim-to-real transfer of soft robots by learning," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 6, pp. 5296–5306, Dec. 2022.

[21] J. M. Bern, Y. Schnider, P. Banzet, N. Kumar, and S. Coros, "Soft robot control with a learned differentiable model," in *Proc. 3rd IEEE Int. Conf. Soft Robot.*, 2020, pp. 417–423.

[22] A. T. Mathew, I. B. Hmida, C. Armanini, F. Boyer, and F. Renda, "Sorosim: A MATLAB toolbox for hybrid rigid–soft robots based on the geometric variable-strain approach," *IEEE Robot. Automat. Mag.*, vol. 30, no. 3, pp. 106–122, Sep. 2023.

[23] C. D. Santina, C. Duriez, and D. Rus, "Model-based control of soft robots: A survey of the state of the art and open challenges," *IEEE Control Syst. Mag.*, vol. 43, no. 3, pp. 30–65, Jun. 2023.

[24] A. Zanna, "Collocation and relaxed collocation for the FER and the magnus expansions," *SIAM J. Numer. Anal.*, vol. 36, no. 4, pp. 1145–1182, Apr. 1999.