Fast Whole-Body Strain Regulation in Continuum Robots

Lekan Molu

***Abstract*— We propose reaching steps towards the real-time strain control of multiphysics, multiscale continuum soft robots. To study this problem fundamentally, we ground ourselves in a model-based control setting enabled by mathematically precise dynamics of a soft robot prototype. Poised to integrate, rather than reject, inherent mechanical nonlinearity for embodied compliance, we first separate the original robot dynamics into two separate subdynamics — aided by a perturbing time-scale separation parameter. Second, we prescribe a set of stabilizing nonlinear backstepping controllers for regulating the resulting subsystems' strain dynamics. Third, we study the interconnected singularly perturbed system by analyzing and establishing its stability. Fourth, our theories are backed up by fast numerical results on a single arm of the Octopus robot arm. We demonstrate strain regulation to equilibrium, in a significantly reduced time, of the whole-body reduced-order dynamics of infinite degrees-of-freedom soft robots. This paper communicates our thinking within the backdrop of embodied intelligence: it informs our conceptualization, formulation, computational setup, and yields improved control performance for the nonlinear control of infinite degrees-of-freedom soft robots.**

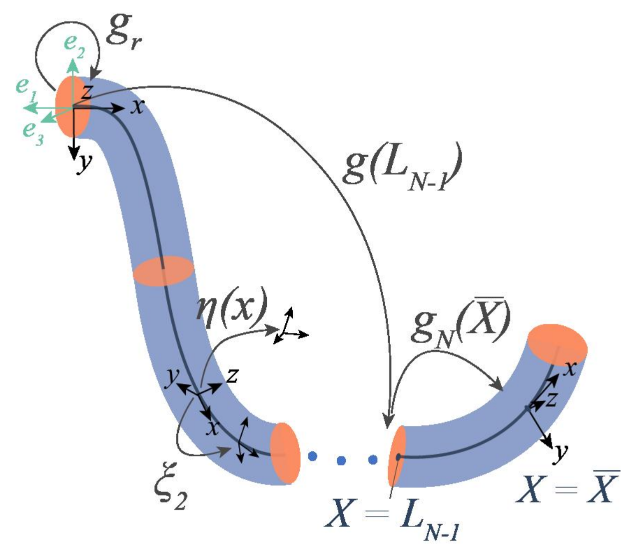

Fig. 1. Simplified configuration of an Octopus arm, reprinted from Molu and Chen [8].

I. Introduction

Soft manipulators, inspired by the functional role of living organisms' soft tissues, provide better compliance and configurability compared to their rigid counterparts. In proof-of-concept studies and in certain real-world cases, they have found applications in delicate 6D dexterous bending and whole-arm manipulation tasks [3], minimally invasive surgery in tight spaces [11, 12], inspection [7], and assistive rehabilitation [14, 10] tasks, where otherwise stiff and rigid robot configurations possess worse stiffness-to-weight ratios and manipulability. Despite their attractiveness, rigid robots are still the go-to mechanism in many automation tasks today. How can we bridge this divide for soft robot adoption in automation? We argue a sustained research effort for developing real-time computational tools for interaction modeling and control will be the key to wide adoption.

Soft robots are multiphysics systems that generate physically heterogeneous interactions from muscle activations to contact and adhesion with the environment in an embodied intelligence fashion [23]. Embodied intelligence stipulates that rather than reject external mechanical processes that impede performance, a robot should leverage its shape, bending, and twisting capabilities along with constraints in the external environment in achieving its desired configuration. The morphing characteristics of a soft robots occur at multiple scales from millimeters (in their continuum deformation characterization) to meters (in their overarching compliance strategy). We are poised with the fast and precise control of soft robots. To systematically dissect the problem, we focus on model-based control methods. This is attractive since the long time scales required to computationally compute models and control has been a drawback for their ubiquitous adoption in automation tasks. We take a holistic approach from modeling, applied mathematics and control, to fast scientific computing schemes to solve the multiscale problem constrained by the robot's multiphysics.

The author is with Microsoft Research NYC lekanmolu@microsoft.com.

Being a continuum physical phenomenon, the default machinery for analyses are nonlinear partial differential equations (PDEs). However, nonlinear PDE theory is tedious and computationally intensive for realizing computationally fast and compliant behavior in soft robots. There are notable strides in reduced-order, finite-dimensional mathematical models that induce tractability in computtaional models. A non-exhaustive list range from morphoelastic filament theory [9, 4, 2], to generalized Cosserat rod theory [20, 1], the constant curvature model [3], the piecewise constant curvature model [22, 16], and ordinary differential equations-based discrete Cosserat model [18, 19].

To study the problem at hand, we leverage [19]'s kinetic model in groundeing the layered multirate control scheme [6] we propose to manage the feedback control of various multiphysics components of a prototype soft robot. In this sentiment, we take the view of reduced order modeling and control with singular perturbation techniques [5]. Discretizing the continuum into piecewise constant strain sections [19], we consider regions where the robot's activation influences its mass density the most to be the principal or fast subsystem to be controlled on a finer scale. The remaining microstructures on the robot are considered the slower subsystem which can be solved at a much coarser resolution. Our goal is to devise a tractable mathematical scheme for separating the system dynamics into two subdynamical systems, controlled at different time scales, to improve computational time and accuracy. To encourage resilience in control, we sidestep linear control methods [15, 8] and opt for nonlinear control. The motivation is for the robot to utilize, not discard, its inherent mechanical nonlinearity feedback in achieving control compliance whilst improving computational time. Computational time is improved by leveraging a time-scale separation singular perturbation scheme and exploiting interprocess communication on a modern GPU and its host CPU.

Contributions: Our contributions are as follows:

- we separate the robot dynamics into separate time scales by manipulating the governing dynamics equations with a perturbation parameter;
- we then devise separate nonlinear controllers for either subdynamics, each operating at different time resolutions on separate GPU and host CPU threads;
- between the two separated subdynamics, an asynchronous communication scheme enables passing dynamics and control computational data from one thread to the other – the subdynamics and controller of the other system are "frozen" within the other subsystem's control and dynamics thread – we do not freeze the other process itself;
- a multi-rate sampling of state measurements asynchronously controls each subsystem: a fast sampling of the fast state variable is employed in a fast nonlinear backstepping controller and a slow-sampling of the slow state variable is employed in a slow backstepping controller. There is not a stringent requirement for communication between both subsystems so that the overall controller takes the form of a decentralized one;
- we achieve a faster computational time for control compared to previously reported results [21, 8].

Our formulation avoids the empirical hierarchical computational schemes typically employed on soft robot bodies such as Shih et al. [21]. While in a way our contribution adheres to this bio-inspired hierarchical computational scheme,a layered modeling and control scheme from a rigorous dynamical systems viewpoint enables us to preserve stability guarantees to the computational scheme. This allows the negligence of (i) parasitic parameters which otherwise complicate system model; (ii) extraneous minute time constants, and mass densities etc; and (iii) the overparameterization caused by sensitive neural network (and hence non-interpretability of) models used for the high-level controllers in bio-inspired models such as [21].

The rest of this paper is structured as follows: background and theoretical machinery are described in §II; §III introduces the singularly perturbed dynamics framework and in §IV, we prescribe the layered dynamics and backstepping controllers for the separated system including stability analyses; numerical simulations are presented in §V, and we conclude the paper in §VI.

II. Notations and Preliminaries

Matrices and vectors are respectively upper- and lowercase bold-faced letters. The strain field and strain twist vectors are $\xi \in \mathbb{R}^6$ and $\eta \in \mathbb{R}^3$, respectively. Sets, screw stiffness, wrench tensors, and the gravitational vector are upper-case Calligraphic bold-faced characters. Distributed wrench tensors are signified by an overbar, e.g. $\bar{\mathcal{F}}$. For a curve $\boldsymbol{X} : [0, L]$, where L is the curve's length at time t, the robot's configuration is denoted as $\mathcal{X}_t(\boldsymbol{X})$. The matrix $\boldsymbol{A}$'s Frobenius norm is denoted $\|\boldsymbol{A}\|$ while its Euclidean norm is $\|\boldsymbol{A}\|_2$. The Lie algebra of the Lie group $\mathbb{SE}(3)$ is $\mathfrak{se}(3)$. The special orthogonal group consisting of corkscrew rotations is $SO(3)$. The structure's configuration $\boldsymbol{g}(\boldsymbol{X})$ is a member of the Lie group $\mathbb{SE}(3)$, whose adjoint and coadjoint are respectively denoted $\text{Ad}_{\boldsymbol{g}}$, $\text{Ad}^{\star}_{\boldsymbol{g}}$. We remark that these are parameterized by the curve, $\boldsymbol{X}$. In generalized coordinate, the joint vector of a soft structure is denoted $\boldsymbol{q} = [\xi_1^\top, \ldots, \xi_{n_\xi}^\top]^\top \in \mathbb{R}^{6n_\xi}$ and $\dot{\boldsymbol{q}} = [\eta_1^\top, \ldots, \eta_{n_\xi}^\top]^\top \in \mathbb{R}^{6n_\xi}$. For a roll, pitch and yaw angles θ, ϕ, ψ, a typical strain ξ_i or strain twist vector η_i takes the forms $[\theta, \phi, \psi, x, y, z]^\top$ and $[\dot{\theta}, \dot{\phi}, \dot{\psi}, \dot{x}, \dot{y}, \dot{z}]^\top$ in our notation.

A. SoRo Configuration

Our analysis is amenable to many soft robots with one predominantly longer dimension than the other two (see Fig. 1) so that "thin" Cosserat rod theory [20] applies. Shown in Fig. 1, the inertial frame is the basis triad $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$ and $\boldsymbol{g}_r$ is the inertial to base frame transformation. For a cable-driven arm, actuation occurs through the central axis of the robot and at the point $\bar{\boldsymbol{X}}$ per section. The configuration matrix that parameterizes curve L_n in X is denoted $\boldsymbol{g}_{L_n}$. The robot's z-axis is offset in orientation from the inertial frame by $-90°$ so that a transformation from the base to inertial frames is

$$\boldsymbol{g}_r = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{1}$$

B. Continuous Strain Vector and Twist Velocity Fields

Suppose that $p(\boldsymbol{X}) \in \mathbb{R}^6$ describes a microsolid's position on the soft body at t and let $R(\boldsymbol{X})$ be the corresponding orientation matrix. Let the pose be $[p(\boldsymbol{X}), R(\boldsymbol{X})]$. Then, the robot's C-space, parameterized by a curve $g(\cdot)$: $\boldsymbol{X} \rightarrow \mathbb{SE}(3)$, is $g(\boldsymbol{X}) = \begin{pmatrix} R(\boldsymbol{X}) & p(\boldsymbol{X}) \\ \boldsymbol{0}^\top & 1 \end{pmatrix}$. Suppose that $\varepsilon(\boldsymbol{X}) \in \mathbb{R}^3$ and $\gamma(\boldsymbol{X}) \in \mathbb{R}^3$ respectively denote the linear and angular strain components of the soft arm. Then, the arm's strain field is a state vector, $\breve{\xi}(\boldsymbol{X}) \in \mathfrak{se}(3)$, along the curve $\boldsymbol{g}(\boldsymbol{X})$ i.e. $\breve{\xi}(\boldsymbol{X}) = \boldsymbol{g}^{-1}\partial g/\partial \boldsymbol{X} \triangleq$

$g^{-1}\partial_x g$. In the microsolid frame, the matrix and vector representation of the strain state are respectively $\breve{\xi}(\boldsymbol{X}) = \begin{pmatrix} \hat{\gamma} & \varepsilon \\ \mathbf{0} & 0 \end{pmatrix} \in \mathfrak{se}(3), \quad \xi(\boldsymbol{X}) = \begin{pmatrix} \gamma^\top & \varepsilon^\top \end{pmatrix}^\top \in \mathbb{R}^6$. Read $\hat{\gamma}$: the anti-symmetric matrix representation of γ. Read $\breve{\xi}$: the isomorphism mapping the twist vector, $\xi \in \mathbb{R}^6$, to its matrix representation in $\mathfrak{se}(3)$. Furthermore, let $\nu(\boldsymbol{X}), \omega(\boldsymbol{X})$ respectively denote the linear and angular velocities of the curve $g(\boldsymbol{X})$. Then, the velocity of $\boldsymbol{g}(\boldsymbol{X})$ is the twist vector field $\breve{\eta}(\boldsymbol{X}) = \boldsymbol{g}^{-1}\partial \boldsymbol{g}/\partial t \triangleq \boldsymbol{g}^{-1}\partial_t \boldsymbol{g}$. In the microsolid frame, $\breve{\eta}(\boldsymbol{X}) = \begin{pmatrix} \hat{\omega} & \nu \\ \mathbf{0} & 0 \end{pmatrix} \in \mathfrak{se}(3), \quad \eta(\boldsymbol{X}) = \begin{pmatrix} \omega^\top & \nu^\top \end{pmatrix}^\top \in \mathbb{R}^6$.

C. Discrete Cosserat-Constitutive PDEs

The PCS model assumes that (ξ_i, η_i) $i = 1, \ldots, N$ robot sections are constant. Spatially spliced along sectional boundaries, the overall strain position and velocity of the entire soft robot is a piecewise sum of the sectional strain field parameters.

Using d'Alembert's principle, the generalized dynamics for PCS model Fig. 1 under external and actuation loads admits the form [19]

$$\underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}_a \boldsymbol{J} d\boldsymbol{X}\right]}_{\boldsymbol{M}(\boldsymbol{q})} \ddot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} J^\top \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \boldsymbol{\mathcal{M}}_a \boldsymbol{J} d\boldsymbol{X}\right]}_{\boldsymbol{C}_1(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} +$$
$$\underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}_a \dot{\boldsymbol{J}} d\boldsymbol{X}\right]}_{\boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{D}} \boldsymbol{J} \|\boldsymbol{J}\dot{\boldsymbol{q}}\|_p d\boldsymbol{X}\right]}_{\boldsymbol{D}(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}}$$
$$- \underbrace{(1 - \rho_f/\rho) \left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}} \mathrm{Ad}_{\boldsymbol{g}}^{-1} d\boldsymbol{X}\right]}_{\boldsymbol{N}(q)} \mathrm{Ad}_{\boldsymbol{g}_r}^{-1} \boldsymbol{\mathcal{G}} - \underbrace{\boldsymbol{J}^\top(\bar{\boldsymbol{X}}) \boldsymbol{\mathcal{F}}_p}_{\boldsymbol{F}(\boldsymbol{q})}$$
$$- \underbrace{\int_0^{L_N} \boldsymbol{J}^\top \left[\nabla_x \boldsymbol{\mathcal{F}}_i - \nabla_x \boldsymbol{\mathcal{F}}_a + \mathrm{ad}^\star_{\eta_n} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)\right] d\boldsymbol{X}}_{\boldsymbol{u}(\boldsymbol{q})} = 0, \tag{2}$$

for a Jacobian $\boldsymbol{J}(\boldsymbol{X})$ (see definition in [19]), wrench of internal forces $\boldsymbol{\mathcal{F}}_i(\boldsymbol{X})$, distributed wrench of actuation loads $\bar{\boldsymbol{\mathcal{F}}}_a(\boldsymbol{X})$, and *distributed* wrench of the applied external forces $\bar{\mathcal{F}}_e(\boldsymbol{X})$. The torque and (internal) force are respectively $\boldsymbol{M}_k, \boldsymbol{F}_k$ for sections k; and $\boldsymbol{\mathcal{M}}(X)$ is the screw mass inertia matrix, given as $\boldsymbol{\mathcal{M}}(X) = \mathrm{diag}\left(\boldsymbol{I}_x, \boldsymbol{I}_y, \boldsymbol{I}_z, \boldsymbol{A}, \boldsymbol{A}, \boldsymbol{A}\right) \rho$ for a body density ρ, sectional area A, bending, torsion, and second inertia operator I_x, I_y, I_z respectively. In (2), $\boldsymbol{\mathcal{M}}_a = \boldsymbol{\mathcal{M}} + \boldsymbol{\mathcal{M}}_f$ is a lumped sum of the microsolid mass inertia operator, $\boldsymbol{\mathcal{M}}$, and that of the added mass fluid, $\boldsymbol{\mathcal{M}}_f$; dX is the length of each section of the multi-robot arm; $\mathcal{D}(\boldsymbol{X})$ is the drag fluid mass matrix; $\boldsymbol{J}(\boldsymbol{X})$ is the Jacobian operator; $\|\cdot\|_p$ is the translation norm of the expression contained therein; ρ_f is the density of the fluid in which the material moves; ρ is the body density; $\mathcal{G}$ is the gravitational vector defined as $\mathcal{G} = [0, 0, 0, -9.81, 0, 0]^T$; and $\mathcal{F}_p$ is the applied wrench at the point of actuation $\bar{\boldsymbol{X}}$.

Suppose that $\boldsymbol{z} = \dot{\boldsymbol{q}}$ and the robot's state at a configuration $\boldsymbol{g}$ is $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{z}^\top]^\top$, then equation (2) can be appropriately written in standard Newton-Euler (N-E) form as

$$\begin{aligned} \boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{z}} + \left[\boldsymbol{C}_1(\boldsymbol{q}, \boldsymbol{z}) + \boldsymbol{C}_2(\boldsymbol{q}, \boldsymbol{z}) + \boldsymbol{D}(\boldsymbol{q}, \boldsymbol{z})\right] \boldsymbol{z} = \\ \tau(\boldsymbol{q}) + \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{N}(\boldsymbol{q}) \mathrm{Ad}_{\boldsymbol{g}_r}^{-1} \mathcal{G}. \end{aligned} \tag{3}$$

III. Singularly Perturbed Dynamics

Seeking a robust response to parametric variations, noise sensitivity, and parasitic small time constants in the dynamics that increase model order, we separate system (3) into a standard two-time-scale singularly perturbed system consisting of fast-changing (here, $\dot{z}_2$) and slow-changing (i.e. $\dot{z}_1$) subdynamics. Thus, we write

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{f}(\boldsymbol{z}_1, \boldsymbol{z}_2, \epsilon, \boldsymbol{u}_s, t), \;\; \boldsymbol{z}_1(t_0) = \boldsymbol{z}_1(0), \;\; \boldsymbol{z}_1 \in \mathbb{R}^{6N}, \tag{4a}$$
$$\epsilon \dot{\boldsymbol{z}}_2 = \boldsymbol{g}(\boldsymbol{z}_1, \boldsymbol{z}_2, \epsilon, \boldsymbol{u}_f, t), \;\; \boldsymbol{z}_2(t_0) = \boldsymbol{z}_2(0), \;\; \boldsymbol{z}_2 \in \mathbb{R}^{6N} \tag{4b}$$

where $\boldsymbol{f}$ and $\boldsymbol{g}$ are $\mathcal{C}^n (n \gg 0)$ differentiable functions of their arguments, $\epsilon > 0$ denotes all small parameters to be ignored[1], $\boldsymbol{u}_s$ is the slow sub-dynamics' control law, and $\boldsymbol{u}_f$ is the fast sub-dynamics' controller.

Set $\epsilon = 0$ for the slow subsystem $\boldsymbol{u}_f = 0$ so that (4b) becomes the algebraic equation

$$0 = \boldsymbol{g}(\boldsymbol{z}_1, \boldsymbol{z}_2, 0, 0, t). \tag{5}$$

To ensure that the fast subsystem has a distinct equilibrium manifold, we proceed with the following standard assumption from singular perturbation theory [5].

Assumption 1 (Real and distinct root): Equation (5) has the unique and distinct root $\boldsymbol{z}_2 = \boldsymbol{\phi}(\boldsymbol{z}_1, t)$ (for a sufficiently smooth $\boldsymbol{\phi}(\cdot)$) so that

$$0 = \boldsymbol{g}(\boldsymbol{z}_1, \boldsymbol{\phi}(\boldsymbol{z}_1, t), 0, 0, t) \triangleq \bar{\boldsymbol{g}}(\boldsymbol{z}_1, 0, t), \;\; \boldsymbol{z}_1(t_0) = \boldsymbol{z}_1(0). \tag{6}$$

The slow subsystem therefore becomes

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{f}(\boldsymbol{z}_1, \boldsymbol{\phi}(\boldsymbol{z}_1, t), 0, \boldsymbol{u}_s, t) \triangleq \boldsymbol{f}_s(\boldsymbol{z}_1, \boldsymbol{u}_s, t). \tag{7}$$

For the fast subdynamics, let us introduce the time scale $T = t/\epsilon$, and write the deviation of $\boldsymbol{z}_2$ from its isolated equilibrium manifold, $\boldsymbol{\phi}(\boldsymbol{z}_1, t)$ as $\tilde{\boldsymbol{z}}_2 = \boldsymbol{z}_2 - \boldsymbol{\phi}(\boldsymbol{z}_1, t)$. Then, (4) becomes

$$\frac{d\boldsymbol{z}_1}{dT} = \epsilon \boldsymbol{f}(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_2 + \boldsymbol{\phi}(\bar{\boldsymbol{z}}_1, t), \epsilon, \boldsymbol{u}_s, t), \tag{8a}$$
$$\frac{d\tilde{\boldsymbol{z}}_2}{dT} = \epsilon \frac{d\boldsymbol{z}_2}{dt} - \epsilon \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{z}_1} \dot{\boldsymbol{z}}_1, \tag{8b}$$
$$= \boldsymbol{g}(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_2 + \boldsymbol{\phi}(\boldsymbol{z}_1, t), \epsilon, \boldsymbol{u}_f, t) - \epsilon \frac{\partial \boldsymbol{\phi}(\boldsymbol{z}_1, t)}{\partial \boldsymbol{z}_1} \dot{\boldsymbol{z}}_1. \tag{8c}$$

Setting $\epsilon = 0$, we obtain the fast subdynamics

$$\frac{d\tilde{\boldsymbol{z}}_2}{dT} = \boldsymbol{g}(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_2 + \boldsymbol{\phi}(z_1, t), 0, \boldsymbol{u}_f, t) \tag{9}$$

with $\boldsymbol{z}_1$ frozen to its initial values.

[1] Restriction to a two-time-scale is not binding and one can choose to expand the system into multiple sub-dynamics across multiple time scales.

A. Soft Robots' Dynamics Separation

Since we take a discretized Cosserat approach in the robot's analytical dynamics, the total energy of the robot can be decomposed into those motions along the sections' barycenter or center of mass (as it were for a rigid body) and those relative to the barycenter motions. Denote the composite mass distribution as a result of the microsolid i's barycenter motion as $\boldsymbol{\mathcal{M}}_i^{\text{core}}$. The relative motion w.r.t to $\boldsymbol{\mathcal{M}}_i^{\text{core}}$ can be considered a perturbation from that of $\boldsymbol{\mathcal{M}}_i^{\text{core}}$, denoted $\boldsymbol{\mathcal{M}}^{\text{pert}}$, so that altogether $\boldsymbol{\mathcal{M}}^{\text{pert}} = \boldsymbol{\mathcal{M}} \setminus \boldsymbol{\mathcal{M}}^{\text{core}}$.

Denoting the indices of the perturbation and core components of the soft microsolids as $(L^p_{\text{min}}, L^p_{\text{max}})$ and $(L^c_{\text{min}}, L^c_{\text{max}})$ in (10), respectively, we find that $\boldsymbol{M}_p = \int_{L^p_{\text{min}}}^{L^p_{\text{max}}} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}_p \boldsymbol{J} dX$, $M_c = \int_{L^c_{\text{min}}}^{L^c_{\text{max}}} \boldsymbol{J}^\top \mathcal{M}_c \boldsymbol{J} dX$ and every other matrix in (10) is similarly defined. Given the robot configuration in Fig. 1, we choose $0 \leq L^p_{\text{min}} < L^c_{\text{min}}$ and $L^c_{\text{max}} < L^p_{\text{max}} \leq L$, where $(L^c_{\text{max}} > L^c_{\text{min}}), (L^p_{\text{max}} > L^p_{\text{min}})$. Let $\bar{\boldsymbol{M}}^p = \boldsymbol{M}^p/\epsilon$, $\bar{\boldsymbol{C}}_1^p = \boldsymbol{C}_1^p/\epsilon$, $\bar{\boldsymbol{C}}_2^p = \boldsymbol{C}_2^p/\epsilon$, and $\bar{\boldsymbol{N}}^p = \boldsymbol{N}^p/\epsilon$. As a result, $\boldsymbol{M}^p(q)$ will be small and can be considered as a perturbation of $\boldsymbol{M}^c(q)$. The same argument holds for the $\boldsymbol{N}, \boldsymbol{C}$, and $\boldsymbol{D}$ matrices. The matrices are therefore separable as

$$\boldsymbol{M}(\boldsymbol{q}) = (\boldsymbol{M}^c + \boldsymbol{M}^p)(\boldsymbol{q}),\ \boldsymbol{N} = (\boldsymbol{N}^c + \boldsymbol{N}^p)(\boldsymbol{q}), \tag{10a}$$
$$\boldsymbol{F}(\boldsymbol{q}) = (\boldsymbol{F}^c + \boldsymbol{F}^p)(\boldsymbol{q}), \quad \boldsymbol{D}(\boldsymbol{q}) = (\boldsymbol{D}^c + \boldsymbol{D}^p)(\boldsymbol{q}) \tag{10b}$$
$$\boldsymbol{C}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (\boldsymbol{C}_1^c + \boldsymbol{C}_1^p)(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{10c}$$
$$\boldsymbol{C}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (\boldsymbol{C}_2^c + \boldsymbol{C}_2^p)(\boldsymbol{q}, \dot{\boldsymbol{q}}). \tag{10d}$$

where each matrix has been diagonalized. Dropping the templated arguments for easy readability, each separate matrix in (10) takes the form

$$\boldsymbol{M} = \underbrace{\begin{bmatrix} \boldsymbol{\mathcal{H}}_{\text{fast}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix}}_{\boldsymbol{M}^c(\boldsymbol{q})} + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}} \\ \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}\ \top} & \boldsymbol{\mathcal{H}}_{\text{slow}} \end{bmatrix}}_{\boldsymbol{M}^p(\boldsymbol{q})}, \tag{11}$$

where each block in the matrices $\boldsymbol{M}^c(\boldsymbol{q})$ and $\boldsymbol{M}^p(\boldsymbol{q})$ are invertible (see [8]), and by extension $\boldsymbol{\mathcal{H}}_{\text{fast}}$ is also invertible; $\boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}$ denotes the decomposed mass of the perturbed sections of the robot relative to the core sections. Let the components of the robot state $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{z}^\top]^\top$ decompose as $\boldsymbol{q} = [\boldsymbol{q}_{\text{fast}}^\top, \boldsymbol{q}_{\text{slow}}^\top]^\top$, $\boldsymbol{z} = [\boldsymbol{z}_{\text{fast}}^\top, \boldsymbol{z}_{\text{slow}}^\top]^\top$, where $\boldsymbol{x}_{\text{fast}}$ denotes the components of $\boldsymbol{x}$ belonging to the fast subsystem and $\boldsymbol{x}_{\text{slow}}$ denotes the components of $\boldsymbol{a}$ belonging to the slow subsystem. Furthermore, let $\bar{\boldsymbol{M}}^p = \boldsymbol{M}^p/\epsilon$, and let $\boldsymbol{u} = [\boldsymbol{u}_{\text{fast}}^\top, \boldsymbol{u}_{\text{slow}}^\top]^\top$ be the applied torque (control law to be designed). Rewriting (3) with the singular perturbation parameter $\epsilon = \|\boldsymbol{M}^p\|/\|\boldsymbol{M}^c\|$, we have

$$(\boldsymbol{M}^c + \epsilon \bar{\boldsymbol{M}}^p)\dot{\boldsymbol{z}} = \boldsymbol{s} + \boldsymbol{u}, \tag{12}$$

where

$$\boldsymbol{s} = \begin{bmatrix} \boldsymbol{s}_{\text{fast}} \\ \boldsymbol{s}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^c + \boldsymbol{N}^c \text{Ad}_{\boldsymbol{g}_r}^{-1} \boldsymbol{\mathcal{G}} - [\boldsymbol{C}_1^c + \boldsymbol{C}_2^c + \boldsymbol{D}^c]\boldsymbol{z}_{\text{fast}} \\ \boldsymbol{F}^p + \boldsymbol{N}^p \text{Ad}_{\boldsymbol{g}_r}^{-1} \boldsymbol{\mathcal{G}} - [\boldsymbol{C}_1^p + \boldsymbol{C}_2^p + \boldsymbol{D}^p]\boldsymbol{z}_{\text{slow}} \end{bmatrix}. \tag{13}$$

Since $\boldsymbol{\mathcal{H}}_{\text{fast}}$ is invertible, let

$$\bar{\boldsymbol{M}}^p = \begin{bmatrix} \bar{\boldsymbol{M}}_{11}^p & \bar{\boldsymbol{M}}_{12}^p \\ \bar{\boldsymbol{M}}_{21}^p & \bar{\boldsymbol{M}}_{22}^p \end{bmatrix} \text{ and } \boldsymbol{\Delta} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} & \boldsymbol{0} \end{bmatrix}, \tag{14}$$

then premultiplying both sides of (12) by $\boldsymbol{I} - \epsilon\boldsymbol{\Delta}$, and ignoring the squared term in ϵ, it can be verified that

$$\begin{bmatrix} \boldsymbol{\mathcal{H}}_{\text{fast}} & \epsilon \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}} \\ \boldsymbol{0} & \epsilon \boldsymbol{\mathcal{H}}_{\text{slow}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}_{\text{fast}} \\ \dot{\boldsymbol{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{\text{fast}} \\ \boldsymbol{s}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{s}_{\text{fast}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_{\text{fast}} \\ \boldsymbol{u}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{u}_{\text{fast}} \end{bmatrix}. \tag{15}$$

Rearranging,

$$\begin{bmatrix} \boldsymbol{\mathcal{H}}_{\text{fast}} & \bar{\boldsymbol{M}}_{12}^p \\ \boldsymbol{0} & \bar{\boldsymbol{M}}_{22}^p \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}_{\text{fast}} \\ \epsilon \dot{\boldsymbol{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{\text{fast}} \\ \boldsymbol{s}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{s}_{\text{fast}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_{\text{fast}} \\ \boldsymbol{u}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{u}_{\text{fast}} \end{bmatrix} \tag{16}$$

which is in the standard singularly perturbed form (4).

1) Fast subsystem dynamics extraction: Consider the fast time scale $T = t/\epsilon$, with $dT/dt = 1/\epsilon$. It follows that the dynamics on this time scale is $\dot{\boldsymbol{z}}_{\text{fast}} = \frac{d\boldsymbol{z}_{\text{fast}}}{dt} \equiv \frac{1}{\epsilon}\frac{d\boldsymbol{z}_{\text{fast}}}{dT} \triangleq \frac{1}{\epsilon}\boldsymbol{z}'_{\text{fast}}$ and $\epsilon \dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{z}'_{\text{slow}}$.

Hence, rewriting (16), we have

$$\begin{bmatrix} \boldsymbol{\mathcal{H}}_{\text{fast}} & \epsilon \bar{\boldsymbol{M}}_{12}^p \\ \boldsymbol{0} & \bar{\boldsymbol{M}}_{22}^p \end{bmatrix} \begin{bmatrix} \boldsymbol{z}'_{\text{fast}} \\ \boldsymbol{z}'_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \epsilon \boldsymbol{s}_{\text{fast}} \\ \boldsymbol{s}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{s}_{\text{fast}} \end{bmatrix} + \begin{bmatrix} \epsilon \boldsymbol{u}_{\text{fast}} \\ \boldsymbol{u}_{\text{slow}} - \epsilon \bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{u}_{\text{fast}} \end{bmatrix}, \tag{17}$$

or,

$$\boldsymbol{z}'_{\text{fast}} = \epsilon \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1}(\boldsymbol{s}_{\text{fast}} + \boldsymbol{u}_{\text{fast}}) - \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}} \boldsymbol{z}'_{\text{slow}} \tag{18a}$$
$$\boldsymbol{\mathcal{H}}_{\text{slow}} \boldsymbol{z}'_{\text{slow}} = \boldsymbol{s}_{\text{slow}} - \boldsymbol{u}_{\text{slow}} - \boldsymbol{\mathcal{H}}_{\text{slow}} \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1}(\boldsymbol{s}_{\text{fast}} - \boldsymbol{u}_{\text{fast}}) \tag{18b}$$

where the perturbed variables are frozen on this fast time scale.

2) Slow sub-dynamics: To extract the slow subdynamics, we let $\epsilon \to 0$ in (17), so that what is left

$$\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{\mathcal{H}}_{\text{slow}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}_{\text{fast}} \\ \dot{\boldsymbol{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{s}_{\text{slow}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{u}_{\text{slow}} \end{bmatrix} \tag{19}$$

or

$$\dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{\mathcal{H}}_{\text{slow}}^{-1}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) \tag{20}$$

constitutes the system's slow dynamics, where the fast components are frozen.

IV. Hierarchical Controller Synthesis

We seek a *multi-rate feedback backstepping controller* which steer an arbitrary point in $\boldsymbol{q}(t)$ at time t, to a target point $\boldsymbol{q}^d = (\boldsymbol{q}_1^d, \ldots, \boldsymbol{q}_N^d)^\top$. Owing to the long computational times required to realize effective control [8], we transform the Cosserat system into a singularly perturbed system. Under standard singular perturbation theory (SPT) assumptions, we take a composite control system viewpoint – systematically separating the fast and slow dynamics of (3) into a nonlinear two time-scale system comprising separate fast and slow controllers. We proceed to design nonlinear backstepping controllers for the two separate time-scale problems developed in §III-A.

1) Stability analysis of the fast velocity subdynamics: We now conduct a stability analysis of the velocity component of the fast subdynamics in (18) on the time scale $t_f \equiv T \triangleq t/\epsilon$. Consider the transformation $[\boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top]^\top = [\boldsymbol{q}_{\text{fast}}^\top, \boldsymbol{z}_{\text{fast}}^\top]^\top$ where $\boldsymbol{\theta}' = \epsilon \boldsymbol{z}_{\text{fast}}$. Suppose that we choose the virtual input $\boldsymbol{\nu}_1$ such that $\boldsymbol{\theta}' = \boldsymbol{\nu}_1$ and let $\boldsymbol{q}_{\text{fast}}^d = [\boldsymbol{\xi}_1^d, \ldots, \boldsymbol{\xi}_{n_\xi}^d]_{\text{fast}}^\top$ be the desired joint space configuration

Theorem 1: The control law

$$\boldsymbol{q}_{\text{fast}}^d(t_f) - \boldsymbol{q}_{\text{fast}}(t_f) + \boldsymbol{q}_{\text{fast}}'^d(t_f)$$

guarantees an exponential stability of the origin of the subsystem $\boldsymbol{\theta}' = \boldsymbol{\nu}_1$ such that for all $t_f \geq 0$, $\boldsymbol{q}_{\text{fast}}(t_f) \in S$ for a compact set $S \subset \mathbb{R}^{6N}$. That is, $\boldsymbol{q}_{\text{fast}}(t_f)$ remains bounded as $t_f \to \infty$.

Proof: Define the tracking error and corresponding error dynamics as

$$\boldsymbol{e}_1 = \boldsymbol{\theta} - \boldsymbol{q}_{\text{fast}}^d, \implies \boldsymbol{e}_1' = \boldsymbol{\theta}' - \boldsymbol{q}_{\text{fast}}'^{d} \triangleq \boldsymbol{\nu}_1 - \boldsymbol{q}_{\text{fast}}'^{d}. \tag{21a}$$

Consider the following candidate Lyapunov function,

$$\boldsymbol{V}_1(\boldsymbol{e}_1) = \frac{1}{2}\boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1 \tag{22}$$

where $\boldsymbol{K}_p$ is a diagonal matrix of positive damping (gains). Ignoring the templated arguments for ease of readability, for a constant $\boldsymbol{q}_{\text{fast}}^d$, we must have

$$\boldsymbol{V}_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p(\boldsymbol{\nu}_1 - \boldsymbol{q}_{\text{fast}}'^d). \tag{23}$$

Set $\boldsymbol{\nu}_1 = \boldsymbol{q}_{\text{fast}}'^d - \boldsymbol{e}_1$, then

$$\boldsymbol{V}_1' = -\boldsymbol{e}_1 \boldsymbol{K}_p \boldsymbol{e}_1 \leq 2\boldsymbol{V}_1. \tag{24}$$

That is for, $\lim_{t\to\infty} \boldsymbol{e}_1(t) = 0$ the control law $\boldsymbol{q}_{\text{fast}}'^d - \boldsymbol{e}_1 \triangleq \boldsymbol{q}_{\text{fast}}^d - \boldsymbol{q}_{\text{fast}} + \boldsymbol{q}'^d$ implies an exponentially stable origin of the subsystem hence satisfying Assumption 1. ∎

2) Stability analysis of the fast acceleration subdynamics:

Theorem 2: Under the tracking error $\boldsymbol{e}_2 = \boldsymbol{\phi} - \boldsymbol{\nu}_1$ and matrices $(\boldsymbol{K}_p, \boldsymbol{K}_q) = (\boldsymbol{K}_p^\top, \boldsymbol{K}_q^\top) > 0$, the control input

$$\begin{aligned}\boldsymbol{u}_{\text{fast}} = \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}_{\text{fast}}[\boldsymbol{q}_{\text{fast}}''^d + \boldsymbol{e}_1 - 2\boldsymbol{e}_2 - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\boldsymbol{e}_1] \\ + \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}\boldsymbol{z}_{\text{slow}}' - \boldsymbol{s}_{\text{fast}}\end{aligned} \tag{25}$$

exponentially stabilizes the fast subdynamics (18).

Proof: First recall that

$$\begin{aligned}\boldsymbol{e}_1' &= \boldsymbol{\theta}' - \boldsymbol{q}_{\text{fast}}'^d \triangleq \boldsymbol{z}_{\text{fast}} - \boldsymbol{q}_{\text{fast}}'^d + (\boldsymbol{\nu}_1 - \boldsymbol{\nu}_1) && (26a)\\ &= (\boldsymbol{\phi} - \boldsymbol{\nu}_1) + (\boldsymbol{\nu}_1 - \boldsymbol{q}_{\text{fast}}'^d) \triangleq \boldsymbol{e}_2 - \boldsymbol{e}_1. && (26b)\end{aligned}$$

Now, consider the whole nonlinear fast subsystem (18). It follows that

$$\begin{aligned}\boldsymbol{e}_2' &= \boldsymbol{\phi}' - \boldsymbol{\nu}_1' = \boldsymbol{z}_{\text{fast}}' + \boldsymbol{e}_1' - \boldsymbol{q}_{\text{fast}}''^d && (27)\\ &= \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1}\left[\epsilon\boldsymbol{u}_{\text{fast}} + \epsilon\boldsymbol{s}_{\text{fast}} - \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}\boldsymbol{z}_{\text{slow}}'\right] + (\boldsymbol{e}_2 - \boldsymbol{e}_1) - \boldsymbol{q}_{\text{fast}}''^d.\end{aligned}$$

Suppose that we choose the Lyapunov candidate function

$$\boldsymbol{V}_2(\boldsymbol{e}_1, \boldsymbol{e}_2) = \boldsymbol{V}_1 + \frac{1}{2}\boldsymbol{e}_2^\top\boldsymbol{K}_q\boldsymbol{e}_2 = \frac{1}{2}[\boldsymbol{e}_1 \ \boldsymbol{e}_2]\begin{bmatrix}\boldsymbol{K}_p & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_q\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_1\\ \boldsymbol{e}_2\end{bmatrix},$$

it can be verified that

$$\begin{aligned}\boldsymbol{V}_2'(\boldsymbol{e}_1, \boldsymbol{e}_2) &= \boldsymbol{e}_1^\top\boldsymbol{K}_p\boldsymbol{e}_1' + \boldsymbol{e}_2^\top\boldsymbol{K}_q\boldsymbol{e}_2' && (28a)\\ &= \boldsymbol{e}_1^\top\boldsymbol{K}_p(\boldsymbol{e}_2 - \boldsymbol{e}_1) + \boldsymbol{e}_2^\top\boldsymbol{K}_q[\boldsymbol{\mathcal{H}}_{\text{fast}}^{-1}(\epsilon\boldsymbol{u}_{\text{fast}} + \epsilon\boldsymbol{s}_{\text{fast}} - \\ &\quad \boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}\boldsymbol{z}_{\text{slow}}') + (\boldsymbol{e}_2 - \boldsymbol{e}_1) - \boldsymbol{q}_{\text{fast}}''^d]. && (28b)\end{aligned}$$

Substituting the value of $\boldsymbol{u}_{\text{fast}}$ in (25) into the foregoing (and ignoring the templated arguments for ease of readability), we have

$$\begin{aligned}\boldsymbol{V}_2' &= \boldsymbol{e}_1^\top\boldsymbol{K}_p(\boldsymbol{e}_2 - \boldsymbol{e}_1)\\ &\quad - \boldsymbol{e}_2^\top\boldsymbol{K}_q\left(\boldsymbol{e}_2 - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\boldsymbol{e}_1\right) && (29a)\\ &= -\boldsymbol{e}_1^\top\boldsymbol{K}_p\boldsymbol{e}_1 - \boldsymbol{e}_2^\top\boldsymbol{K}_q\boldsymbol{e}_2 \triangleq -2\boldsymbol{V}_2 \leq 0. && (29b)\end{aligned}$$

Since $\boldsymbol{V}_2'$ is negative definite, the equilibrium point $\boldsymbol{e}_{12} = [\boldsymbol{e}_1^\top, \boldsymbol{e}_2^\top]^\top = \boldsymbol{0}$ is exponentially stable. And the controller that satisfies the equilibrium points $[e_1^\top, e_2^\top]^\top = \boldsymbol{0}$ is given by (25) or in simplified form

$$\begin{aligned}\boldsymbol{u}_{\text{fast}} = \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}_{\text{fast}}[\boldsymbol{q}_{\text{fast}}''^d - \tilde{\boldsymbol{q}}_{\text{fast}} - 2\tilde{\boldsymbol{q}}_{\text{fast}}' - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\tilde{\boldsymbol{q}}_{\text{fast}}]\\ + \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}\boldsymbol{z}_{\text{slow}}' - \boldsymbol{s}_{\text{fast}},\end{aligned}$$

where $\tilde{\boldsymbol{q}}_{\text{fast}} = \boldsymbol{q}_{\text{fast}} - \boldsymbol{q}_{\text{fast}}^d$ and $\tilde{\boldsymbol{q}}_{\text{fast}}' = \boldsymbol{q}_{\text{fast}}' - \boldsymbol{q}_{\text{fast}}'^d$. On the fast subsystem, the control input value when the perturbed parameters are frozen is

$$\boldsymbol{u}_{\text{slow}} = \boldsymbol{s}_{\text{slow}} - \boldsymbol{\mathcal{H}}_{\text{slow}}\boldsymbol{z}_{\text{slow}}' - \boldsymbol{\mathcal{H}}_{\text{slow}}\boldsymbol{\mathcal{H}}_{\text{fast}}^{-1}(\boldsymbol{s}_{\text{fast}} - \boldsymbol{u}_{\text{fast}}) \tag{30}$$

where the variables $\boldsymbol{s}_{\text{slow}}, \boldsymbol{\mathcal{H}}_{\text{slow}}, \boldsymbol{z}_{\text{slow}}'$ are frozen. ∎

3) Stability analysis of the slow subsystem: For the slow subsystem (20), let $[\boldsymbol{\alpha}^\top, \boldsymbol{\beta}^\top]^\top = [\boldsymbol{q}_{\text{slow}}^\top, \boldsymbol{z}_{\text{slow}}^\top]^\top$ so that we have the following dynamics

$$\dot{\boldsymbol{\alpha}} = \boldsymbol{z}_{\text{slow}} \triangleq \boldsymbol{\nu}_2, \tag{31}$$

where $\boldsymbol{\nu}_2$ is a virtual input. The tracking error is $\boldsymbol{e}_3 = \boldsymbol{\alpha} - \boldsymbol{q}_{\text{slow}}^d$ with error dynamics $\dot{\boldsymbol{e}}_3 = \dot{\boldsymbol{\alpha}} - \dot{\boldsymbol{q}}_{\text{slow}}^d = \boldsymbol{\nu}_2 - \dot{\boldsymbol{q}}_{\text{slow}}^d$. Consider the Lyapunov function candidate

$$\boldsymbol{V}_3(\boldsymbol{e}_3) = \frac{1}{2}\boldsymbol{e}_3^\top\boldsymbol{K}_r\boldsymbol{e}_3 \text{ where } \boldsymbol{K}_r = \boldsymbol{K}_r^\top > 0. \tag{32}$$

It follows that

$$\dot{\boldsymbol{V}}_3(\boldsymbol{e}_3) = \boldsymbol{e}_3^\top\boldsymbol{K}_r\dot{\boldsymbol{e}}_3 = \boldsymbol{e}_3^\top\boldsymbol{K}_r(\boldsymbol{\nu}_2 - \dot{\boldsymbol{q}}_{\text{slow}}^d). \tag{33}$$

Setting $\boldsymbol{\nu}_2 = \dot{\boldsymbol{q}}_{\text{slow}}^d - \boldsymbol{e}_3$. It follows that $\dot{\boldsymbol{V}}_3 = -\boldsymbol{e}_3^\top\boldsymbol{K}_r\boldsymbol{e}_3 = -2\dot{\boldsymbol{V}}_3 \leq 0$. This implies $\lim_{t\to\infty}\boldsymbol{e}_3 = 0$, so that we have exponential stability on the slow subdynamics. The control law that stabilizes the system is therefore

$$\boldsymbol{u}_{\text{slow}} = \dot{\boldsymbol{q}}_{\text{slow}}^d - \tilde{\boldsymbol{q}}_{\text{slow}}, \tag{34}$$

where $\tilde{\boldsymbol{q}}_{\text{slow}} = \boldsymbol{q}_{\text{slow}} - \boldsymbol{q}_{\text{slow}}^d$.

4) Stability of the singularly perturbed interconnected system: Let $\varepsilon = (0, 1)$ and consider the composite Lyapunov function candidate $\Sigma(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}})$ as a weighted combination of $\boldsymbol{V}_2$ and $\boldsymbol{V}_3$ i.e. ,

$$\boldsymbol{\Sigma}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) = (1-\varepsilon)\boldsymbol{V}_2(\boldsymbol{z}_{\text{fast}}) + \varepsilon \boldsymbol{V}_3(\boldsymbol{z}_{\text{slow}}),\ 0 < \varepsilon < 1. \tag{35}$$

It follows that,

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) &= (1-\varepsilon)[\boldsymbol{e}_1^\top \boldsymbol{K}_p \dot{\boldsymbol{e}}_1 + \boldsymbol{e}_2^\top \boldsymbol{K}_q \dot{\boldsymbol{e}}_2] + \varepsilon \boldsymbol{e}_3^\top \boldsymbol{K}_r \dot{\boldsymbol{e}}_3, \\ &= -2(\boldsymbol{V}_2 + \boldsymbol{V}_3) + 2\varepsilon \boldsymbol{V}_2 \le 0 \end{aligned} \tag{36}$$

which is clearly negative definite for any $\varepsilon \in (0, 1)$. Therefore, we conclude that the origin of the singularly perturbed system is asymptotically stable under the control laws.

$$\boldsymbol{u}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) = (1-\varepsilon)\boldsymbol{u}_{\text{fast}} + \varepsilon \boldsymbol{u}_{\text{slow}}. \tag{37}$$

V. Numerical Results

A. System Setup

We replicate the robot parameters used in Molu and Chen [8] with tweaks to accommodate our layered control method. As seen in Fig. 1, the tip load acts on the $+y$-axis in the robot's base frame so that the tip wrench applied at $\bar{X} = L$, can be expressed as

$$\boldsymbol{\mathcal{F}}_p = \text{diag}\left(\boldsymbol{R}^\top(L), \boldsymbol{R}^\top(L)\right)\begin{pmatrix} \boldsymbol{0}_{3\times 1} & 0 & 10 & 0 \end{pmatrix}^\top \tag{38}$$

where $\boldsymbol{R}(L)$ is the first 3×3 block submatrix of (1). We use $\boldsymbol{\mathcal{F}}_p^y$ to represent the tip load acting along the $+y$ direction in what follows. Given the geometry of the robot, we chose a drag coefficient of 0.82 (a Reynolds number of order 104) for underwater operations. We set the Young's modulus as $\boldsymbol{E} = 110kPa$ and the shear viscosity modulus to $3kPa$. The bending second inertia momenta are $I_y = I_z = \pi r^4/4$ while the torsion second moment of inertia is $I_x = \pi r^4/2$ for $r = 0.1m$, the arm's radius – uniform across sections. The arm length is $L = 2m$. We assume a (near-incompressible) rubber material makes up the robot's body with Poisson ratio 0.45; the mass is chosen as $\boldsymbol{\mathcal{M}} = \rho \cdot \text{diag}([I_x, I_y, I_z, A, A, A])$ for a cylindrical soft shell's nominal density of $\rho = 2,000kgm^{-3}$ as used in [19]; the cross-sectional area $A = \pi r^2$ so that $I_x = \pi r^4/2$. The drag screw stiffness matrix $\boldsymbol{D}$ in (3) is a function of each section's geometry and hydrodynamics so that $\boldsymbol{D} = -\rho_w \nu^T \nu \check{\boldsymbol{D}} \nu / |\nu|$ where ρ_w is the water density set to $997kg/m^3$, and $\check{\boldsymbol{D}}$ is the tensor that models the geometry and hydrodynamics factors in the viscosity model (see [19, §II.B, eq. 6]). The curvilinear abscissa, $X \in [0, L]$ was discretized into 41 microsolids per section.

B. Asynchronous deployment

We asynchronously deployed the slow and fast controllers on both subsystems using two separate threads: the slow controller (34) was deployed on the CPU while the fast controller (25) was deployed on a CUDA-capable GPU thread running PyTorch [13]. The results from the slow subsystem namely $\boldsymbol{z}_{\text{slow}}$ and $\boldsymbol{u}_{\text{slow}}$ are retrieved from a Linux named pipe within the faster subsystem's thread. These frozen values are then used in computing $\boldsymbol{z}_{\text{fast}}$ and $\boldsymbol{u}_{\text{fast}}$ in the fast subsystem thread. We test the capability of the controller in various settings as we did in [8]. We ran a host of experiments to ascertain the veracity of our results. For the sake of conciseness, we report only two results here[2].

In our first experiment, we discretized the robot into 6 pieces. The fast subdynamics consisted of 4 out of the 6 pieces while the remaining two pieces constituted the slow subdynamics. Every piece in the robot was further divided into 13 segments. We found that this coarse segmentation scheme does not affect the quality of the strain regulation equilibrium that we strive for. We considered a desired strain along the $+y$ direction of 0.5 and loaded the tip of the robot with a force of 10 Newtons. For the controller gains, we set $\boldsymbol{K}_p = 5$ and $\boldsymbol{K}_d = 0.5$. The stabilization results are illustrated in Fig. 2 showing strain and strain twist regulations along the x and y directions of the robot. The total runtime for realizing accurate strain states regulation was 18 minutes as shown.

In a second experiment, we chose a robot with finer resolution – 10 pieces. And we set six of the pieces to the fast sub-dynamics and four to the slow subdynamics. Every other parameter of the robot remained the same as in the previous experiment. We found the strain states reached steady state within 25 minutes. The results are shown in Fig. 3.

We further compared the time it takes for this hierarchical control scheme to regulate strain states to equilibrium against our previous work Molu and Chen [8] that employed a PD

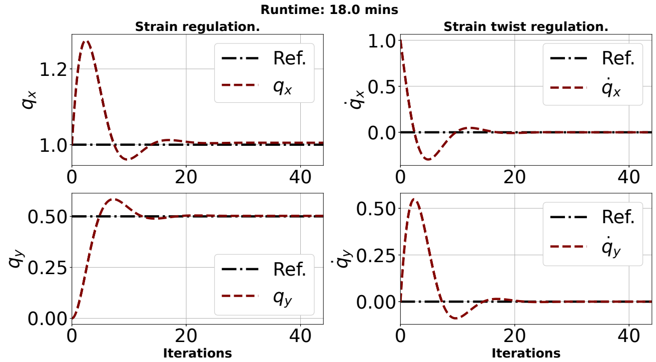

Fig. 2. Backstepping control on the singularly perturbed soft robot system with 10 discretized pieces, divided into 6 fast and 4 slow pieces. For a tip load of $\boldsymbol{\mathcal{F}}_p^y = 10\,N$, the backstepping gains were set as $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ for a desired joint configuration $\xi^d = [0, 0, 0, 1, 0.5, 0]^\top$ and $\eta^d = \boldsymbol{0}_{6\times 1}$ that is uniform throughout the robot sections.

Pieces			Runtime (mins)	
Total	Fast	Slow	Hierarchical SPT (mins)	Single-layer PD control (hours)
6	4	2	18.01	51.46
8	5	3	30.87	68.29
10	7	3	32.39	107.43

TABLE I

TIME TO REACH STRAIN STEADY STATE.

[2] Users can download the online code from the link in the bottom of the title of this paper for further testing and evaluation.

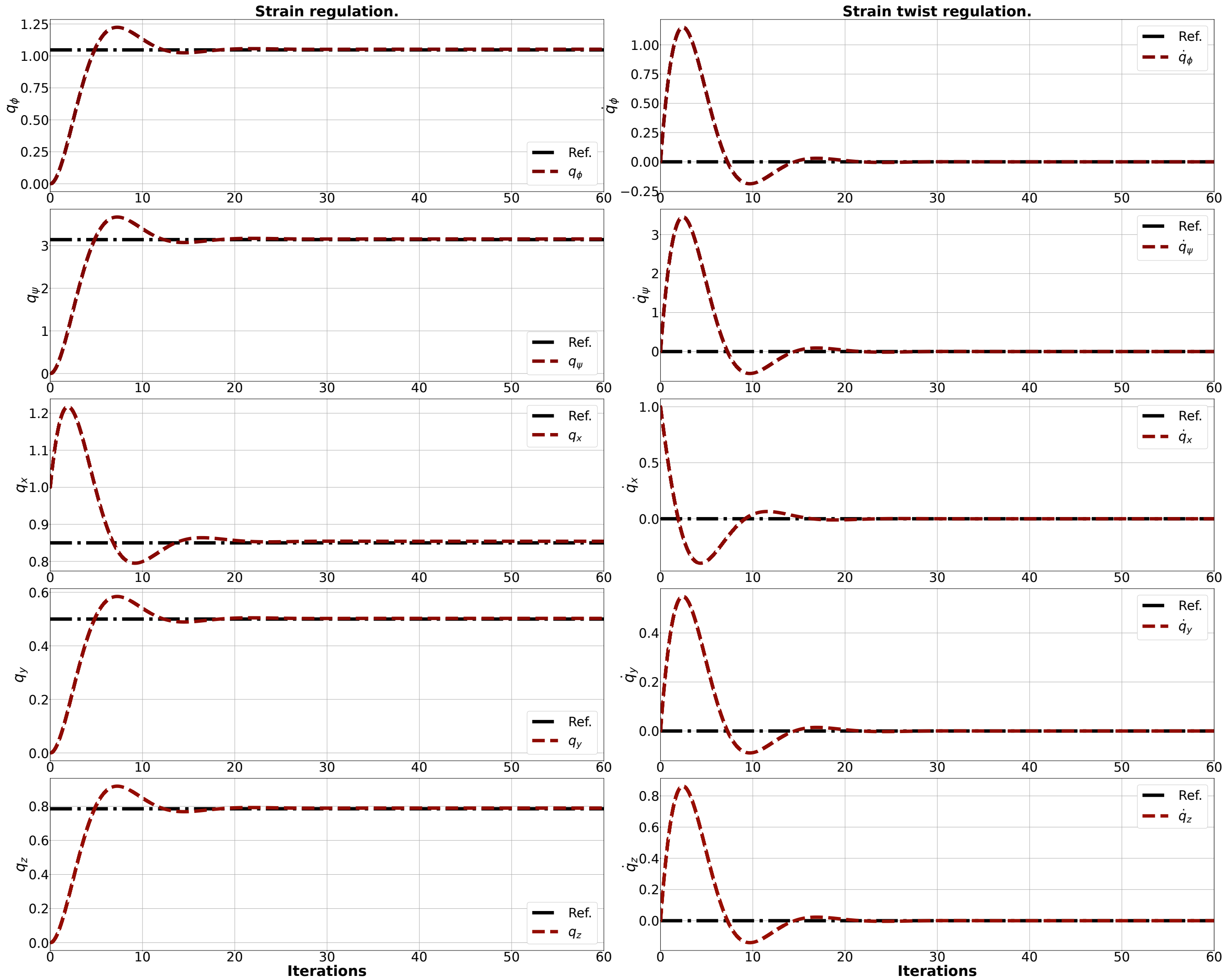

Fig. 3. Backstepping control on the singularly perturbed soft robot system with 10 pieces 4 slow and 6 fast sections. For a tip load of $\boldsymbol{\mathcal{F}}_p^y = 10\,N$, the backstepping gains were set as $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ for a desired joint configuration $\xi^d = [0, \pi/3, \pi, 0.85, 0.5, \pi/4]^\top$ and $\eta^d = \mathbf{0}_{6\times 1}$ that is uniform throughout the robot sections. Total runtime was 25 minutes.

single-layer control law in table V-B. All sections of he robot share a similar discretization parameter between the two schemes, with an equal amount of tip load in all experiments; only the control gains are adapted to stabilize each respective system. Computations were carried out on an 80GB A100 CUDA-capable NVIDIA GPU for the single layer PD, and fast subdynamics' controllers. Only the slow subdynamics and its controller are run in a separate CPU thread. In all experiments, we found our new method regulate the strain states to desired equilibrium in a shorter time compared to the PD strain regulation law that does not employ dynamics separation and control hierarchy.

VI. Discussions and Conclusion

In the quest towards the adoption of soft robots in everyday automation processes, we identified that the long processing times for computing models and controllers/policies is a significant drawback. This is demonstrated in table V-B where PD control laws for regulating strain states implies dozens of hours to regulate the robot's continuum configuration – even with a reduced-order model. To circumvent this, we devised a singularly perturbed dynamics by introducing a time-scale perturbation parameter. This allows decomposing the system to separate subdynamics that can be controlled in a decentralized fashion. For the control scheme, we devised a nonlinear backstepping controller that incorporates systemic nonlinearities in the system dynamics. We found that our results do not merely regulate particulate strain states but also achieve desired equilibrium in a matter of minutes. This is in stark contrast to the single-layer control scheme that takes hours before the continuum strain states reach equilibrium.

References

[1] Eugène Maurice Pierre Cosserat and François Cosserat. *Théorie des corps déformables*. A. Hermann et fils,

1909. 1

[2] Mattia Gazzola, LH Dudte, AG McCormick, and Lakshminarayanan Mahadevan. Forward and inverse problems in the mechanics of soft filaments. *Royal Society open science*, 5(6):171628, 2018. 1

[3] Isuru S Godage, David T Branson, Emanuele Guglielmino, Gustavo A Medrano-Cerda, and Darwin G Caldwell. Shape function-based kinematics and dynamics for variable length continuum robotic arms. In *2011 IEEE International Conference on Robotics and Automation*, pages 452–457. IEEE, 2011. 1

[4] Bartosz Kaczmarski, Alain Goriely, Ellen Kuhl, and Derek E Moulton. A Simulation Tool for Physics-informed Control of Biomimetic Soft Robotic Arms. *IEEE Robotics and Automation Letters*, 2023. 1

[5] Kokotovi'c, Petar and Khalil K., Hassan and O'Reilly, John. *Singular Perturbation Methods in Conteol: Analysis and Design*. Society for Industrial and Applied Mathematics, 1999. 1, 3

[6] Nikolai Matni, Aaron D Ames, and John C Doyle. A quantitative framework for layered multirate control: Toward a theory of control architecture. *IEEE Control Systems Magazine*, 44(3):52–94, 2024. 1

[7] Joshua S Mehling, Myron A Diftler, Mars Chu, and Michael Valvo. A minimally invasive tendril robot for in-space inspection. In *The First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, 2006. BioRob 2006.*, pages 690–695. IEEE, 2006. 1

[8] Lekan Molu and Shaoru Chen. Lagrangian Properties and Control of Soft Robots Modeled with Discrete Cosserat Rods. In *IEEE International Conference on Decision and Control, Milan, Italy*. IEEE, 2024. 1, 2, 4, 6

[9] Derek E Moulton, Thomas Lessinnes, and Alain Goriely. Morphoelastic Rods III: Differential Growth and Curvature Generation in Elastic Filaments. *Journal of the Mechanics and Physics of Solids*, 142:104022, 2020. 1

[10] R. W. Nuckols, S. Lee, K. Swaminathan, D. Orzel, R. D. Howe, and C. J. Walsh. Individualization of exosuit assistance based on measured muscle dynamics during versatile walking. *Science Robotics*, 6(60): eabj1362, 2021. 1

[11] Olalekan Ogunmolu, Adwait Kulkarni, Yonas Tadesse, Xuejun Gu, Steve Jiang, and Nicholas Gans. Soft-neuroadapt: A 3-dof neuro-adaptive patient pose correction system for frameless and maskless cancer radiotherapy. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3661–3668. IEEE, 2017. 1

[12] Olalekan Ogunmolu, Xinmin Liu, Nicholas Gans, and Rodney D Wiersma. Mechanism and model of a soft robot for head stabilization in cancer radiation therapy. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4609–4615. IEEE, 2020. 1

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 6

[14] Panagiotis Polygerinos, Zheng Wang, Kevin C Galloway, Robert J Wood, and Conor J Walsh. Soft robotic glove for combined assistance and at-home rehabilitation. *Robotics and Autonomous Systems*, 73: 135–143, 2015. 1

[15] Pietro Pustina, Pablo Borja, Cosimo Della Santina, and Alessandro De Luca. P-sati-d shape regulation of soft robots. *IEEE Robotics and Automation Letters*, 8(1): 1–8, 2022. 2

[16] Ke Qiu, Jingyu Zhang, Danying Sun, Rong Xiong, Haojian Lu, and Yue Wang. An efficient multi-solution solver for the inverse kinematics of 3-section constant-curvature robots. *arXiv preprint arXiv:2305.01458*, 2023. 1

[17] Federico Renda, Michele Giorelli, Marcello Calisti, Matteo Cianchetti, and Cecilia Laschi. Dynamic model of a multibending soft robot arm driven by cables. *IEEE Transactions on Robotics*, 30(5):1109–1122, 2014.

[18] Federico Renda, Vito Cacucciolo, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for soft robot dynamics: A new piece-wise constant strain model with torsion and shears. *IEEE International Conference on Intelligent Robots and Systems*, 2016-Novem:5495–5502, 2016. ISSN 21530866. 1

[19] Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multi-section soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018. 1, 2, 3, 6

[20] M. B. Rubin. *Cosserat Theories: Shells, Rods, and Points*. Springer-Science+Business Medis, B.V., 2000. 1, 2

[21] Chia-Hsien Shih, Noel Naughton, Udit Halder, Heng-Sheng Chang, Seung Hyun Kim, Rhanor Gillette, Prashant G Mehta, and Mattia Gazzola. Hierarchical control and learning of a foraging cyberoctopus. *Advanced Intelligent Systems*, page 2300088, 2023. 2

[22] Robert J. III Webster and Bryan A. Jones. Design and kinematic modeling of constant curvature continuum robots: A review. *The International Journal of Robotics Research*, 29(13):1661–1683, 2010. 1

[23] Davide Zambrano, Matteo Cianchetti, Cecilia Laschi, Helmut Hauser, Rudolf Fuchslin, and Rolf Pfeifer. The Morphological Computation Principles As A New Paradigm For Robotic Design. Opinions and Outlooks on Morphological Computation. *Opinions and Outlooks on Morphological Computation*, 16(25):214–225, 2014. 1