2025 American Control Conference (ACC)
July 8-10, 2025. Denver, CO, USA

Fast Whole-Body Strain Regulation in Continuum Robots

Lekan Molu
https://github.com/robotsorcerer/DCM/tree/SPT

***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 nonlinearities for embodied compliance, we first separate the original robot dynamics into 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 an infinite degrees-of-freedom soft robot. This paper communicates our thinking within the backdrop of embodied intelligence: it informs our conceptualization, formulation, computational setup, and yields improved control performance for infinite degrees-of-freedom soft robots.**

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 [12, 13], inspection [7], and assistive rehabilitation [15, 11] 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 everyday automation? In this paper, we argue that 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 activation 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, components' geometry, along with constraints in the external environment to achieve a desired configuration. Given that soft robots' nonlinear deformation occur at multiple scales: from millimeters (in their continuum deformation state) 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 scale required to computationally resolve models and control has been a drawback for their ubiquitous adoption. We take a holistic approach that includes modeling, applied mathematics and control, and fast scientific computing schemes to solve the multiscale problem constrained by the robot's multiphysics.

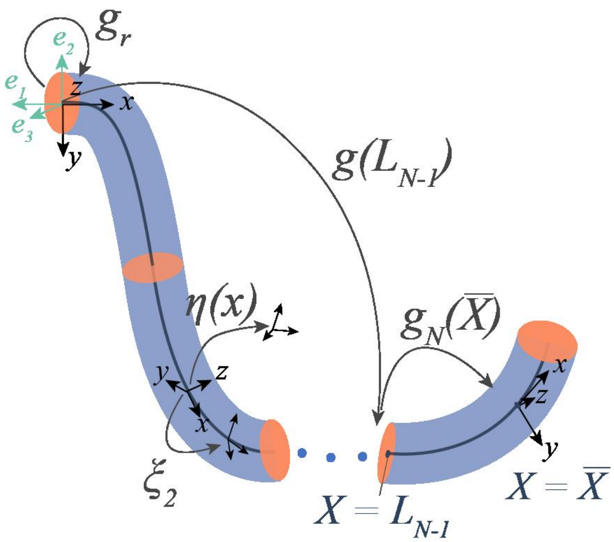

Fig. 1. Simplified configuration of an Octopus arm, reprinted from Molu and Chen [9].

Being a continuum phenomenon, the default machinery for soft robot 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 continuum models. A non-exhaustive list range from morphoelastic filament theory [10, 4, 2], to generalized Cosserat rod theory [20, 1], the constant curvature model [3], the piecewise constant curvature model [22, 17], and ordinary differential equations-based discrete Cosserat model [18, 19].

To study the problem at hand, we leverage [19]'s kinetic model in grounding the layered multirate control scheme [6] of the various interconnected physics components of a soft robot prototype. In this sentiment, we take the view of re-

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

979-8-3315-6937-2/$31.00 ©2025 AACC

duced 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 as the 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. This enables us to devise a tractable mathematical scheme for separating the system dynamics into two separate sub-dynamical systems, controllable at different time scales, to improve computational time and accurate strain regulation. To encourage resilience and improve runtime, we sidestep linear control methods [16, 9] and opt for nonlinear control whilst exploiting interprocess communication on a modern GPU and its host CPU. The motivation is for the robot to utilize, not discard, its inherent mechanical nonlinear feedback in achieving control compliance whilst improving computational time.

Contributions:

- 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 one 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 scheme and a slow-sampling of the slow state variable is employed in a slow backstepping controller framework. 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, 9].

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 and minute time constants as well as 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 upper- and lower-case bold-faced letters, respectively. 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 $\mathbb{SE}(3)$ Lie group is written as $\mathfrak{se}(3)$. The special orthogonal group consisting of corkscrew rotations is denoted $SO(3)$. 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 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 $\boldsymbol{g}_{L_n}$. 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 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 $g(\cdot) : \boldsymbol{X} \to \mathbb{SE}(3)$, 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. $\breve{\xi}(\boldsymbol{X}) = \boldsymbol{g}^{-1}\partial g/\partial \boldsymbol{X} \triangleq \boldsymbol{g}^{-1}\partial_x \boldsymbol{g}$. 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 $(\xi_i, \eta_i)_{i=1,\ldots,N}$ 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]

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

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

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

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, $\dot{\boldsymbol{z}}_2$) and slow-changing (i.e. $\dot{\boldsymbol{z}}_1$) 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.

We assume that the fast feedback law is asymptotically stable (formalized in Assumption 1) such that it does not modify the open-loop equilibrium manifold of the fast dynamics. Thus, setting $\epsilon = 0$ to extract the slow subdynamics (here $\boldsymbol{u}_f = 0$) the system dynamics becomes

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{f}(\boldsymbol{z}_1, \boldsymbol{z}_2, 0, \boldsymbol{u}_s, t), \; \boldsymbol{z}_1(t_0) = \boldsymbol{z}_1(0), \tag{5a}$$

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

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

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

Assumption 1 is a standard assumption in singular perturbation theory [5] and it allows us to isolate the equilibrium manifold of the fast dynamics such that the slow subdynamics takes the form of an algebraic expression. 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

$$\frac{d\boldsymbol{z}_1}{dT} = \epsilon \boldsymbol{f}(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_2 + \boldsymbol{\phi}(\boldsymbol{z}_1, t), \epsilon, \boldsymbol{u}_s, t), \tag{8a}$$

$$\frac{d\tilde{\boldsymbol{z}}_2}{dT} = \epsilon \frac{d\boldsymbol{z}_2}{dt} - \epsilon \frac{\partial \boldsymbol{\phi}}{\partial \boldsymbol{z}_1} \dot{\boldsymbol{z}}_1, \tag{8b}$$

$$= \boldsymbol{g}(\boldsymbol{z}_1, \tilde{\boldsymbol{z}}_2 + \boldsymbol{\phi}(\boldsymbol{z}_1, t), \epsilon, \boldsymbol{u}_f, t) - \epsilon \frac{\partial \boldsymbol{\phi}(\boldsymbol{z}_1, t)}{\partial \boldsymbol{z}_1} \dot{\boldsymbol{z}}_1. \tag{8c}$$

[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.

Setting $\epsilon = 0$, we obtain the algebraic equation

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

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

A. Soft Robots' Dynamics Separation

The robot's motion can be decomposed into those along the discretized sections' barycenter and those relative to the barycenter based on the discretized Cosserat constant strain assumption. Denote the composite mass distribution as a result of microsolid i's barycenter motion as $\boldsymbol{\mathcal{M}}_i^{\text{core}}$. Motions relative to $\boldsymbol{\mathcal{M}}_i^{\text{core}}$ can be considered a perturbation, $\boldsymbol{\mathcal{M}}^{\text{pert}}$, so that $\boldsymbol{\mathcal{M}}^{\text{pert}} = \boldsymbol{\mathcal{M}} \setminus \boldsymbol{\mathcal{M}}^{\text{core}}$. Examining (3), suppose that the perturbation and core microsolids' indices are $(L^p_{\text{min}}, L^p_{\text{max}})$ and $(L^c_{\text{min}}, L^c_{\text{max}})$, respectively, where $0 \leq L^p_{\text{min}} < L^c_{\text{min}}$, $L^c_{\text{max}} < L^p_{\text{max}} \leq L$, and $(L^c_{\text{max}} > L^c_{\text{min}}), (L^p_{\text{max}} > L^p_{\text{min}})$. Then, we can write

$$\boldsymbol{M}(\boldsymbol{q}) = (\boldsymbol{M}^c + \boldsymbol{M}^p)(\boldsymbol{q}),\ \boldsymbol{N} = (\boldsymbol{N}^c + \boldsymbol{N}^p)(\boldsymbol{q}), \tag{10a}$$

$$\boldsymbol{F}(\boldsymbol{q}) = (\boldsymbol{F}^c + \boldsymbol{F}^p)(\boldsymbol{q}), \quad \boldsymbol{D}(\boldsymbol{q}) = (\boldsymbol{D}^c + \boldsymbol{D}^p)(\boldsymbol{q}) \tag{10b}$$

$$\boldsymbol{C}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (\boldsymbol{C}_1^c + \boldsymbol{C}_1^p)(\boldsymbol{q}, \dot{\boldsymbol{q}}), \tag{10c}$$

$$\boldsymbol{C}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) = (\boldsymbol{C}_2^c + \boldsymbol{C}_2^p)(\boldsymbol{q}, \dot{\boldsymbol{q}}) \tag{10d}$$

where $\boldsymbol{M}^p = \int_{L^p_{\text{min}}}^{L^p_{\text{max}}} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}^{pert} \boldsymbol{J} dX$, and $\boldsymbol{M}^c = \int_{L^c_{\text{min}}}^{L^c_{\text{max}}} \boldsymbol{J}^\top \mathcal{M}^{core} \boldsymbol{J} dX$, and every other matrix in (10) is similarly defined.

Suppose that the respective matrices are in diagonal block form, the mass inertia matrix in (10) can be decomposed as (dropping the joint space arguments for ease of readability)

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

where each block $\boldsymbol{M}^c(\boldsymbol{q})$ and $\boldsymbol{M}^p(\boldsymbol{q})$ are invertible (see [9]), and by extension $\boldsymbol{\mathcal{H}}_{\text{fast}}$ is invertible; $\boldsymbol{\mathcal{H}}^{\text{fast}}_{\text{slow}}$ denotes the decomposed mass of the perturbed sections of the robot relative to the core sections.

Introducing the change of variables $[\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top = [\boldsymbol{q}^\top, \boldsymbol{z}^\top]^\top$, so that the robot's state, $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{z}^\top]^\top$ decomposes as $\boldsymbol{q} = [\boldsymbol{q}^\top_{\text{fast}}, \boldsymbol{q}^\top_{\text{slow}}]^\top$, $\boldsymbol{z} = [\boldsymbol{z}^\top_{\text{fast}}, \boldsymbol{z}^\top_{\text{slow}}]^\top$, 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 $\boldsymbol{x}$ belonging to the slow subsystem. Furthermore, let $\bar{\boldsymbol{M}}^p = \boldsymbol{M}^p/\epsilon$, and let $\boldsymbol{u} = [\boldsymbol{u}^\top_{\text{fast}}, \boldsymbol{u}^\top_{\text{slow}}]^\top$ 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

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

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

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

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

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

Rearranging, we must have

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

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

1) Fast subsystem dynamics extraction: Consider the fast time scale $T = t/\epsilon$, with $dT/dt = 1/\epsilon$. The dynamics on this time scale is $\dot{\boldsymbol{z}}_{\text{fast}} = \dfrac{d\boldsymbol{z}_{\text{fast}}}{dt} \equiv \dfrac{1}{\epsilon}\dfrac{d\boldsymbol{z}_{\text{fast}}}{dT} \triangleq \dfrac{1}{\epsilon}\boldsymbol{z}'_{\text{fast}}$ and $\epsilon\dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{z}'_{\text{slow}}$. Rewriting (16), we find

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

or,

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

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

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

2) Slow sub-dynamics: To extract the slow subdynamics, we let $\epsilon \to 0$ in (16), so that what is left, i.e,

$$\dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{\mathcal{H}}^{-1}_{\text{slow}}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) \tag{19}$$

constitutes the system's slow dynamics, where the fast components are frozen on this slow time scale.

IV. Hierarchical Controller Synthesis

We seek a *multi-rate feedback backstepping controller* which steer an arbitrary strain configuration $[\boldsymbol{q}(t)^\top, \dot{\boldsymbol{q}}(t)^\top]^\top$ at time $t < t_f$, to a target point $[\boldsymbol{q}^{d\top}(t_f), \dot{\boldsymbol{q}}^{d\top}]^\top(t_f)$. We now design nonlinear backstepping controllers for the separate subsystems in §III-A.

1) Stability analysis of the fast strain subdynamics: Let us first consider the velocity component of the fast subdynamics in (18); this exists on the time scale $t_f \triangleq T \triangleq$. Consider the transformation $[\boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top]^\top = [\boldsymbol{q}^\top_{\text{fast}}, \boldsymbol{z}^\top_{\text{fast}}]^\top$ where $\boldsymbol{\theta}' = \epsilon\boldsymbol{z}_{\text{fast}}$. Suppose that we choose the virtual input $\boldsymbol{\nu}$ such that $\boldsymbol{\theta}' = \boldsymbol{\nu}$ and let $\boldsymbol{q}^d_{\text{fast}} = [\boldsymbol{\xi}^d_1, \ldots, \boldsymbol{\xi}^d_{n_\xi}]^\top_{\text{fast}}$ be the desired joint space configuration

Theorem 1: The control law

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

is sufficient to guarantee an exponential stability of the origin of $\boldsymbol{\theta}' = \boldsymbol{\nu}$ 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

$$\boldsymbol{e}_1 = \boldsymbol{\theta} - \boldsymbol{q}^d_{\text{fast}}, \implies \boldsymbol{e}_1' = \boldsymbol{\theta}' - \boldsymbol{q}'^d_{\text{fast}} \triangleq \boldsymbol{\nu} - \boldsymbol{q}'^d_{\text{fast}}. \tag{20a}$$

Consider the following candidate Lyapunov function,

$$V_1(\boldsymbol{e}_1) = \frac{1}{2}\boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1 \tag{21}$$

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

$$V_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p(\boldsymbol{\nu} - \boldsymbol{q}'^d_{\text{fast}}). \tag{22}$$

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

$$V_1' = -\boldsymbol{e}_1 \boldsymbol{K}_p \boldsymbol{e}_1 \le 2V_1. \tag{23}$$

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

2) Stability analysis of the strain twist subdynamics:

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

$$\begin{aligned}\boldsymbol{u}_{\text{fast}} = \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}_{\text{fast}}[\boldsymbol{q}''^d_{\text{fast}} + \boldsymbol{e}_1 - 2\boldsymbol{e}_2 - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\boldsymbol{e}_1]\\ + \frac{1}{\epsilon}\boldsymbol{\mathcal{H}}^{\text{fast}}_{\text{slow}}\boldsymbol{z}'_{\text{slow}} - \boldsymbol{s}_{\text{fast}}\end{aligned} \tag{24}$$

exponentially stabilizes the fast subdynamics (18).

Proof: First recall that

$$\boldsymbol{e}_1' = \boldsymbol{\theta}' - \boldsymbol{q}'^d_{\text{fast}} \triangleq \boldsymbol{z}_{\text{fast}} - \boldsymbol{q}'^d_{\text{fast}} + (\boldsymbol{\nu} - \boldsymbol{\nu}) \tag{25a}$$

$$= (\boldsymbol{\phi} - \boldsymbol{\nu}) + (\boldsymbol{\nu} - \boldsymbol{q}'^d_{\text{fast}}) \triangleq \boldsymbol{e}_2 - \boldsymbol{e}_1. \tag{25b}$$

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

$$\begin{aligned}\boldsymbol{e}_2' &= \boldsymbol{\phi}' - \boldsymbol{\nu}' = \boldsymbol{z}'_{\text{fast}} + \boldsymbol{e}_1' - \boldsymbol{q}''^d_{\text{fast}} \\ &= \boldsymbol{\mathcal{H}}^{-1}_{\text{fast}}\left[\epsilon\boldsymbol{u}_{\text{fast}} + \epsilon\boldsymbol{s}_{\text{fast}} - \boldsymbol{\mathcal{H}}^{\text{fast}}_{\text{slow}}\boldsymbol{z}'_{\text{slow}}\right] + (\boldsymbol{e}_2 - \boldsymbol{e}_1) - \boldsymbol{q}''^d_{\text{fast}}.\end{aligned} \tag{26}$$

Suppose that we choose the Lyapunov candidate function

$$V_2(\boldsymbol{e}_1, \boldsymbol{e}_2) = V_1 + \frac{1}{2}\boldsymbol{e}_2^\top \boldsymbol{K}_q \boldsymbol{e}_2 = \frac{1}{2}[\boldsymbol{e}_1\ \boldsymbol{e}_2]\begin{bmatrix}\boldsymbol{K}_p & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_q\end{bmatrix}\begin{bmatrix}\boldsymbol{e}_1\\ \boldsymbol{e}_2\end{bmatrix},$$

it can be verified that

$$V_2'(\boldsymbol{e}_1, \boldsymbol{e}_2) = \boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1' + \boldsymbol{e}_2^\top \boldsymbol{K}_q \boldsymbol{e}_2' \tag{27a}$$

$$\begin{aligned}= \boldsymbol{e}_1^\top \boldsymbol{K}_p(\boldsymbol{e}_2 - \boldsymbol{e}_1) + \boldsymbol{e}_2^\top \boldsymbol{K}_q[\boldsymbol{\mathcal{H}}^{-1}_{\text{fast}}(\epsilon\boldsymbol{u}_{\text{fast}} + \epsilon\boldsymbol{s}_{\text{fast}} - \\ \boldsymbol{\mathcal{H}}^{\text{fast}}_{\text{slow}}\boldsymbol{z}'_{\text{slow}}) + (\boldsymbol{e}_2 - \boldsymbol{e}_1) - \boldsymbol{q}''^d_{\text{fast}}].\end{aligned} \tag{27b}$$

Substituting the value of $\boldsymbol{u}_{\text{fast}}$ in (24) into the foregoing (and ignoring the templated arguments for ease of readability), we have

$$\begin{aligned}V_2' = \boldsymbol{e}_1^\top \boldsymbol{K}_p(\boldsymbol{e}_2 - \boldsymbol{e}_1)\\ - \boldsymbol{e}_2^\top \boldsymbol{K}_q\left(\boldsymbol{e}_2 - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\boldsymbol{e}_1\right)\end{aligned} \tag{28a}$$

$$= -\boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1 - \boldsymbol{e}_2^\top \boldsymbol{K}_q \boldsymbol{e}_2 \triangleq -2V_2 \le 0. \tag{28b}$$

Since 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 $[\boldsymbol{e}_1^\top, \boldsymbol{e}_2^\top]^\top = \boldsymbol{0}$ is given by (24) or in simplified form

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

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

$$\boldsymbol{u}_{\text{slow}} = \boldsymbol{s}_{\text{slow}} - \boldsymbol{\mathcal{H}}_{\text{slow}}\boldsymbol{z}'_{\text{slow}} - \boldsymbol{\mathcal{H}}_{\text{slow}}\boldsymbol{\mathcal{H}}^{-1}_{\text{fast}}(\boldsymbol{s}_{\text{fast}} - \boldsymbol{u}_{\text{fast}}) \tag{29}$$

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

3) Stability analysis of the slow strain subdynamics: For the slow subsystem (19), let $\boldsymbol{e}_3 = \boldsymbol{z}_{\text{slow}} - \boldsymbol{\nu}$ so that $\dot{\boldsymbol{e}}_3 = \dot{\boldsymbol{z}}_{\text{slow}} - \dot{\boldsymbol{\nu}}$. It follows that

$$\dot{\boldsymbol{e}}_3 = \dot{\boldsymbol{z}}_{\text{slow}} - \ddot{\boldsymbol{q}}^d_{\text{fast}} + (\boldsymbol{e}_2 - \boldsymbol{e}_1), \tag{30a}$$

$$= \boldsymbol{\mathcal{H}}^{-1}_{\text{slow}}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) - \ddot{\boldsymbol{q}}^d_{\text{fast}} + (\boldsymbol{e}_2 - \boldsymbol{e}_1). \tag{30b}$$

Theorem 3: The control law

$$\boldsymbol{u}_{\text{slow}} = \boldsymbol{\mathcal{H}}_{\text{slow}}(\boldsymbol{e}_1 - \boldsymbol{e}_2 - \boldsymbol{e}_3 + \ddot{\boldsymbol{q}}^d_{\text{fast}}) - \boldsymbol{s}_{\text{slow}} \tag{31}$$

exponentially stabilizes the slow subdynamics.

Proof: Consider the Lyapunov function candidate

$$V_3(\boldsymbol{e}_3) = \frac{1}{2}\boldsymbol{e}_3^\top \boldsymbol{K}_r \boldsymbol{e}_3 \text{ where } \boldsymbol{K}_r = \boldsymbol{K}_r^\top > 0. \tag{32}$$

It follows that

$$\dot{V}_3(\boldsymbol{e}_3) = \boldsymbol{e}_3^\top \boldsymbol{K}_r \dot{\boldsymbol{e}}_3 \tag{33a}$$

$$= \boldsymbol{e}_3^\top \boldsymbol{K}_r\left[\boldsymbol{\mathcal{H}}^{-1}_{\text{slow}}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) - \ddot{\boldsymbol{q}}^d_{\text{fast}} + \boldsymbol{e}_2 - \boldsymbol{e}_1\right]. \tag{33b}$$

Substituting $\boldsymbol{u}_{\text{slow}}$ in (31), it can be verified that

$$\dot{V}_3(\boldsymbol{e}_3) = \boldsymbol{e}_3^\top \boldsymbol{K}_r \boldsymbol{e}_3 \triangleq -2V_3(\boldsymbol{e}_3) \le 0. \tag{34}$$

Hence, the controller (31) stabilizes the slow subsystem. ■

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 V_2 and V_3 i.e. ,

$$\boldsymbol{\Sigma}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) = (1 - \varepsilon)V_2(\boldsymbol{z}_{\text{fast}}) + \varepsilon V_3(\boldsymbol{z}_{\text{slow}}),\ 0 < \varepsilon < 1. \tag{35}$$

It follows that,

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) &= (1 - \varepsilon)[\boldsymbol{e}_1^\top \boldsymbol{K}_p \dot{\boldsymbol{e}}_1 + \boldsymbol{e}_2^\top \boldsymbol{K}_q \dot{\boldsymbol{e}}_2] + \varepsilon \boldsymbol{e}_3^\top \boldsymbol{K}_r \dot{\boldsymbol{e}}_3, \\ &= -2(V_2 + V_3) + 2\varepsilon V_2 \le 0\end{aligned} \tag{36}$$

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 parameters of [9] 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

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

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 $\boldsymbol{D} = -\rho_w \nu^T \nu \breve{\boldsymbol{D}} \nu / |\nu|$ where ρ_w is the water density set to $997kg/m^3$, and $\breve{\boldsymbol{D}}$ 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. For integrating the system duynamics, we adopt a Runge-Kutta-Fehlberg (RKF) integrator implemented in PyTorch. For every discretized Cosserat piece in our evaluations, we further divided each piece into 13 segments to accommodate the PCS algorithm's modeling precision. Unlike the extremely fine resolution of segments discretization ($\approx$ 64) in [9], we found that this coarse segmentation scheme does not diminish simulation fidelity in all our testing.

B. Deployment and Discussion

We asynchronously deployed the slow and fast controllers on both subsystems using two separate threads: the slow controller (34) was deployed on a host CPU while the fast controller (24) was deployed in parallel PyTorch [14] thread on a CUDA-capable GPU. The slow subsystem state, $\boldsymbol{z}_{\text{slow}}$, and control $\boldsymbol{u}_{\text{slow}}$ are retrieved from a Linux named pipe within the faster subsystem's thread. Computation on the slow subsystem are frozen when computing $\boldsymbol{z}_{\text{fast}}$ and $\boldsymbol{u}_{\text{fast}}$ in the fast subsystem thread. We now report two numerical experiments (for the sake of conciseness) to validate our new scheme. Further testing and evaluation are available in the online code repository.

In a two-axes strain regulation control experiment, we discretized the continuum robot described in §V-A into 6 pieces. The fast and slow subdynamics were divided up as 4 and 2 pieces, respectively. The goal is to have the continuum strain along the $+x$ and $+y$ directions as 1.0 and 0.5 respectively whilst every other axis is kept at zero under a tip load 10 Newtons. We set gains $\boldsymbol{K}_p = 5$ and $\boldsymbol{K}_d = 0.5$. Fig. 2 shows the strain and strain twist stabilization results under a total runtime of 18 minutes. As seen, the system reaches steady state across all axes of interest. We remark that this whole body control scheme takes tens of hours for a typical soft robot (later reported in Table I). This experiment confirms our hypothesis that dynamics decomposition and nonlinear control aids fast strain regulation.

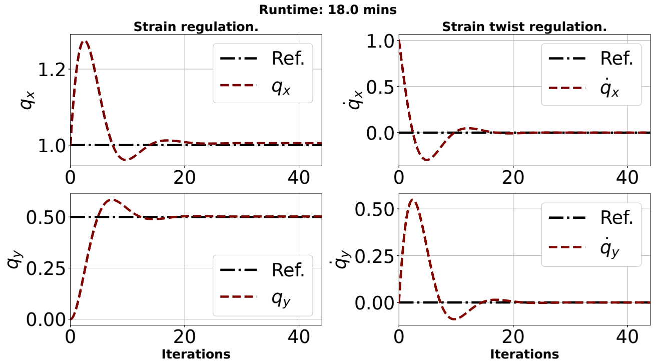

Fig. 2. Backstepping control on the singularly perturbed soft robot system with 10 discretized pieces, divided into 6 fast and 4 slow pieces. For a tip load of $\boldsymbol{\mathcal{F}}_p^y = 10\,N$, the backstepping gains were set as $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ for a desired joint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

Our second experiment employs a PCS scheme with 10 discretized Cosserat sections — with six fast and flow subdynamics, respectively. Under a tip load $\boldsymbol{\mathcal{F}}_p^y = 10\,N$, and backstepping gains $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ we aim for desired strain states $\xi^d = [0, \pi/3, \pi, 0.85, 0.5, \pi/4]^\top$ and twist states $\eta^d = \mathbf{0}_{6\times 1}$. Fig. 3 shows we reached equilibrium in less than 20 iterations of running the RKF scheme within 25 minutes. We found the strain states reached steady state within 25 minutes. The results are shown in Fig. 3.

We further compared the time to reach steady state in our hierarchical control scheme versus a previous work [9] that employed a PD single-layer control law scheme. Here, we employ a similar amount of discretized Cosserat sections and segments in the hierarchical controller (13 segments per sections); while the PD controller employed 41 segments per section. An equal amount of tip load, i.e. $10N$ was employed in all experiments. Computations were carried out on an 80GB A100 CUDA-capable NVIDIA GPU for the single layer PD, and fast subdynamics' controllers. The slow subsystem was executed on the host CPU thread. Table I elucidates our results. In all experiments, we found that the hierarchical scheme was significantly faster in reaching equilibrium whilst preserving whole-body strain regulation compared against the PD strain regulation law.

VI. 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 sig-

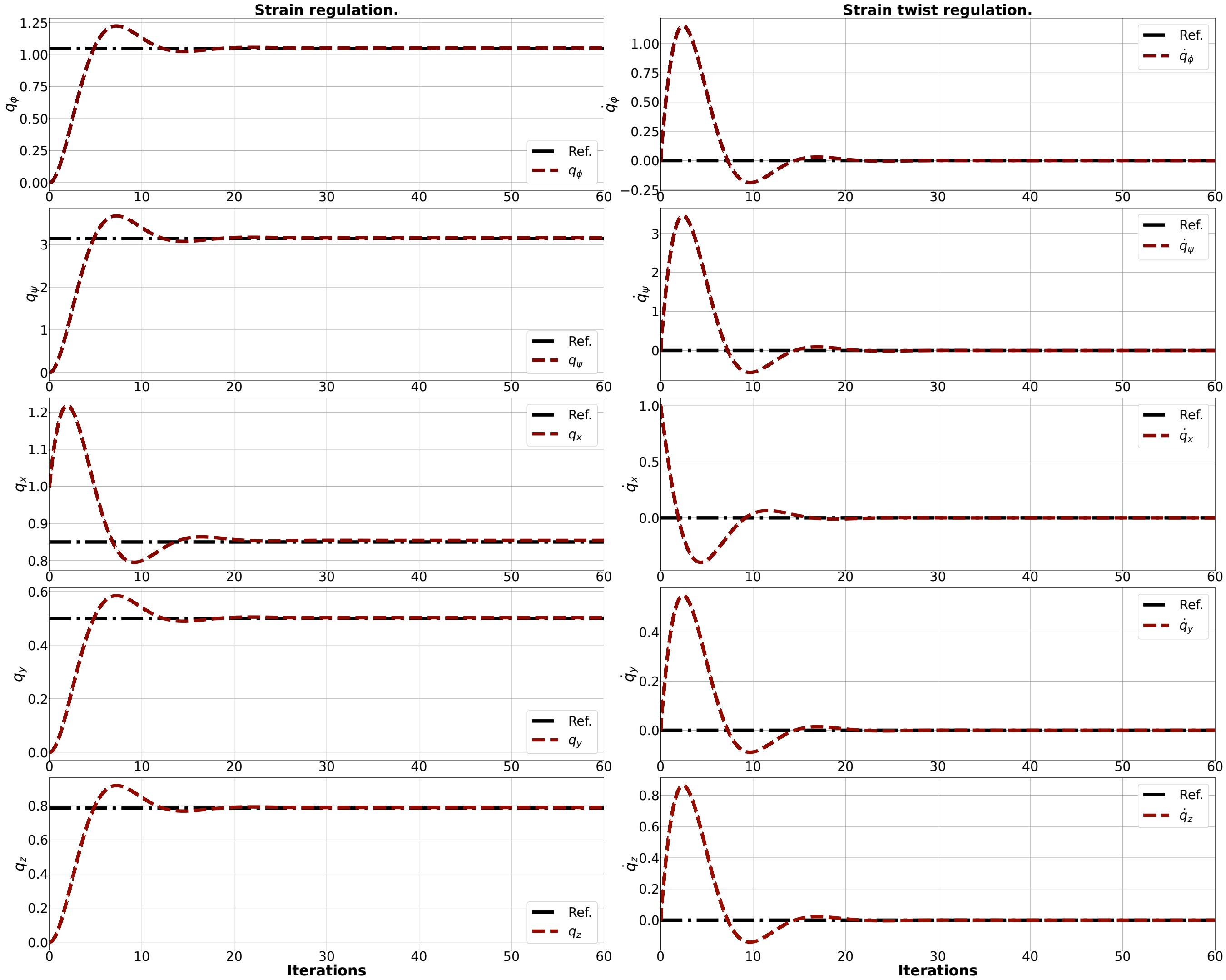

Fig. 3. Backstepping control on the singularly perturbed soft robot system with 10 pieces 4 slow and 6 fast sections.

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 Steady State.

nificant drawback. To circumvent this, we introduced a singularly perturbed technique for decomposing system dynamics to a fast and slower subdynamics, respectively. Stabilizing nonlinear backstepping controllers were introduced to the respective subdynamics to further improve computation times. The fast part was controlled at a finer resolution while the slower part was controlled at a more coarse resolution, with the overall scheme executed in a decentralized fashion. We found that our results do not merely regulate particulate strain states but also achieve desired equilibrium faster and better compared to other reported schemes. Our approach takes a further step towards replicating embodied intelligence [8] in soft robots that mimic the behavior of living matter by engrossing hierarchy layers in soft robots' dynamics and control computational schemes.

References

[1] Eugène Maurice Pierre Cosserat and François Cosserat. *Théorie des corps déformables*. A. Hermann et fils, 1909.

[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.

[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.

[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.

[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.

[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.

[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.

[8] Gianmarco Mengaldo, Federico Renda, Steven L Brunton, Moritz Bächer, Marcello Calisti, Christian Duriez, Gregory S Chirikjian, and Cecilia Laschi. A concise guide to modelling the physics of embodied intelligence in soft robotics. *Nature Reviews Physics*, 4(9):595–610, 2022.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[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.

[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.

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

[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.

[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.

[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.