Reinforcement Learning: States Representation, Morphological Computation, and Robustness.

Lekan Molu

New York City, NY 10012

Presented by **Lekan Molu** (Lay-con Mo-lu)

August 13, 2025

Talk Outline

- System Identification in Reinforcement Learning (RL);
- Robustness of Deep RL Policies:
  - Iterative Dynamic Game;
  - Convergence and Robustness in Deep RL Policy Optimization.
- Reduced-order modeling and morphological control of emergent robot configurations;
  - Soft robots – reduced order differential equation models;
  - Hierarchical adaptive nonlinear control schemes.
- (Abundant details in Appendices)

Technical Overview

This page is left blank intentionally.

Credits

S. Chen

A. Koul

Y. Efroni

D. Misra

D. Foster

R. Islam

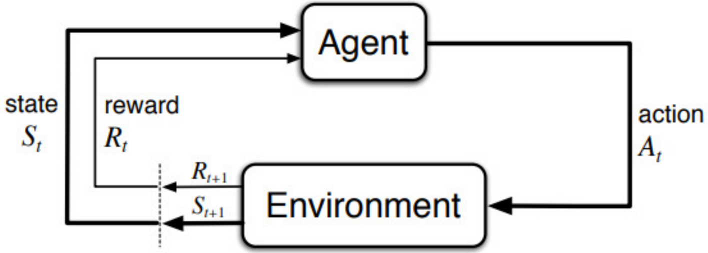
A. Lamb

M. Dudik

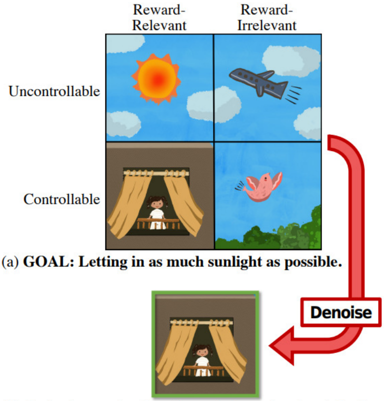
A. Krish.

J. Langford

Standard Reinforcement Learning



Compact States without Exogenous Distractors in RL

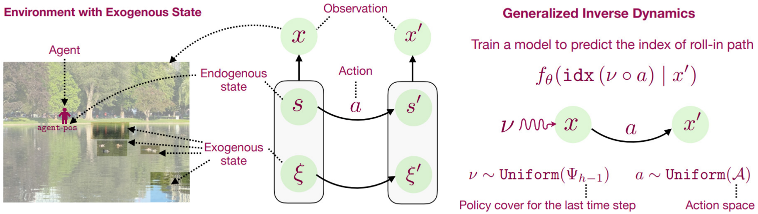


(a) **GOAL: Letting in as much sunlight as possible.**

(b) Optimal control only relies on information that is **both controllable and reward-relevant**. Good world models should ignore other factors as noisy distractors.

Denoised MDPs: Learning World Models Better Than the World Itself [5]

Compact States without Exogenous Distractors in RL



Learning s with [S] whilst ignoring temporally correlated ξ? Source: [3, Fig. 1].

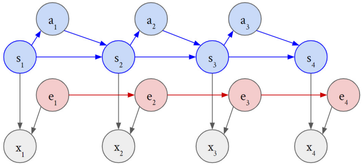
Literature comparison

Algorithms	PPE	OSSR	DBC	CDL	Denoised-MDP	1-Step Inverse	AC-State (Ours)
Exogenous Invariant State	✓	✓	✓	✓	✓	✓	✓
Exogenous Invariant Learning	✓	✓	✗	✗	✗	✓	✓
Flexible Encoder	✓	✗	✓	✗	✓	✓	✓
YOLO (No Resets) Setting	✗	✓	✓	✓	✓	✓	✓
Reward Free	✓	✓	✗	✓	✓	✓	✓
Control-Endogenous Rep.	✓	✓	✗	✓	✓	✗	✓

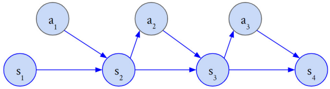
Emphasis on robustness to exogenous information. Comparison with baselines including PPE [3], OSSR [2], DBC [6] , Denoised MDP [5] and One-Step Inverse Models [4].

Rewards-agnostic Exogenous State Invariance in RL

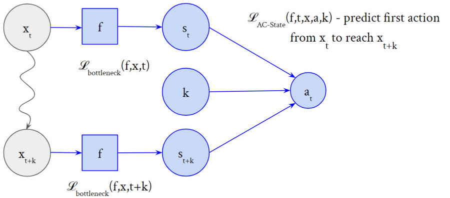
AC-State Discovers the smallest control-endogenous state s assuming factorized dynamics



AC-State collects data with a single random action followed by a high-coverage endogenous policy for k-1 steps



AC-State learns an encoder f for s = f(x) by optimizing a multi-step inverse model with a bottleneck

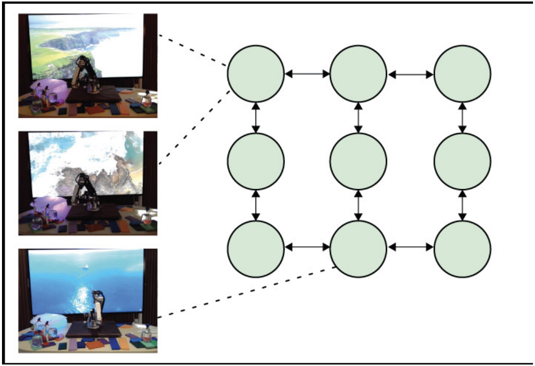


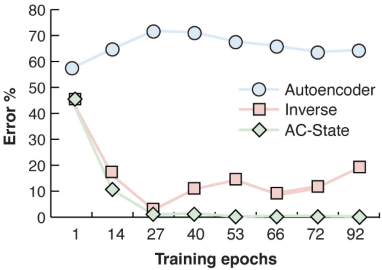
Latent States Discovery – Multi-step Inverse Dynamics

- $\hat{f} \approx \arg\min_{f \in \mathcal{F}} \mathbb{E}_{t,k} \Big[\mathcal{L}_{\mathrm{ACS}}\left(f, x, a, t, k\right) + \mathcal{L}_{\mathrm{B}}(f, x_t) + \mathcal{L}_{\mathrm{B}}(f, x_{t+k}) \Big]$

$$\mathcal{L}_{\mathrm{ACS}}\left(f, x, a, t; k\right) = -\log(\mathbb{P}\left(a_t | f(x_t), f(x_{t+k}); k\right)) \quad (1)$$

AC State in Action

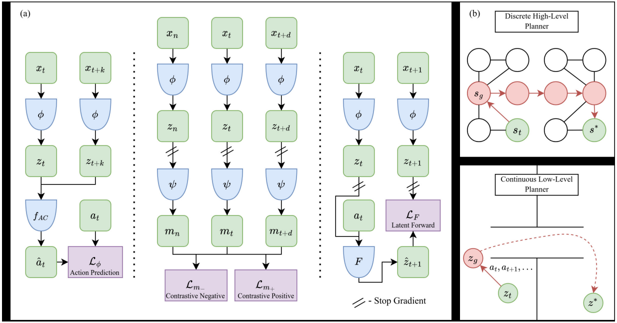




PCLAST: Agent Plannable Continuous Latent States

PcLast: Discovering Plannable Continuous Latent States

Anurag Koul [*1] **Shivakanth Sujit** [*234] **Shaoru Chen** [1] **Ben Evans** [5] **Lili Wu** [1] **Byron Xu** [1] **Rajan Chari** [1]
Riashat Islam [36] **Raihan Seraj** [36] **Yonathan Efroni** [7] **Lekan Molu** [1] **Miro Dudik** [1] **John Langford** [1] **Alex Lamb** [1]



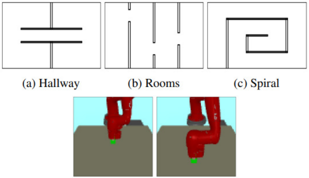
PCLAST Algorithm

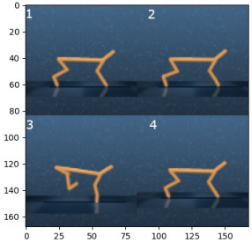
Algorithm 1 n-Level Planner

Require:
- Current observation x_t
- Goal observation x_{goal}
- Planning horizon H
- Encoder $\phi(\cdot)$
- PCLAST map $\psi(\cdot)$
- Latent forward dynamics $\delta(\cdot,\cdot)$
- Multi-Level discrete transition graphs $\{\mathcal{G}_i\}_{i=2}^{n}$

Ensure: Action sequence $\{a_i\}_{i=0}^{H-1}$

1: Compute current continuous latent state $\hat{s}_t = \phi(x_t)$ and target latent state $\hat{s}^* = \phi(x_{goal})$.
{See Appendix E for details of high-level planner and low-level planner.}
2: **for** $i = n, n-1, \ldots, 2$ **do**
3: $\quad \hat{s}^*$ = high-level planner($\hat{s}_t$, $\hat{s}^*$, $\mathcal{G}_i$)
{Update waypoint using a hierarchy of abstraction.}
4: **end for**
5: $\{a_i\}_{i=0}^{H-1}$ = low-level planner($\hat{s}_t, \hat{s}^*, H, \delta, \psi$)
{Solve the trajectory optimization problem.}


(a) Hallway (b) Rooms (c) Spiral
(d) Sawyer Reach Environment



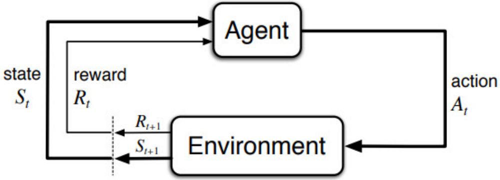
PCLAST Results

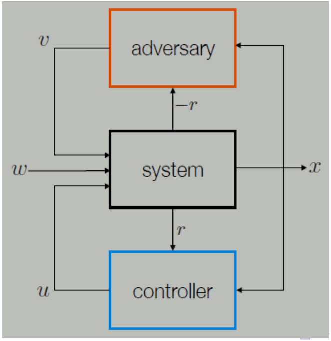
Method	Reward type	Hallway	Rooms	Spiral	Sawyer-Reach
PPO	Dense	6.7 ± 0.6	7.5 ± 7.1	11.2 ± 7.7	$\mathbf{86.00 \pm 5.367}$
PPO + ACRO	Dense	10.0 ± 4.1	23.3 ± 9.4	23.3 ± 11.8	84.00 ± 6.066
PPO + PCLaSt	Dense	$\mathbf{66.7 \pm 18.9}$	$\mathbf{43.3 \pm 19.3}$	$\mathbf{61.7 \pm 6.2}$	78.00 ± 3.347
PPO	Sparse	1.7 ± 2.4	0.0 ± 0.0	0.0 ± 0.0	68.00 ± 8.198
PPO + ACRO	Sparse	21.7 ± 8.5	5.0 ± 4.1	11.7 ± 8.5	$\mathbf{92.00 \pm 4.382}$
PPO + PCLaSt	Sparse	$\mathbf{50.0 \pm 18.7}$	$\mathbf{6.7 \pm 6.2}$	$\mathbf{46.7 \pm 26.2}$	82.00 ± 5.933
CQL	Sparse	3.3 ± 4.7	0.0 ± 0.0	0.0 ± 0.0	32.00 ± 5.93
CQL + ACRO	Sparse	15.0 ± 7.1	$\mathbf{33.3 \pm 12.5}$	$\mathbf{21.7 \pm 10.3}$	68.00 ± 5.22
CQL + PCLaSt	Sparse	$\mathbf{40.0 \pm 0.5}$	23.3 ± 12.5	20.0 ± 8.2	$\mathbf{74.00 \pm 4.56}$
RIG	None	0.0 ± 0.0	0.0 ± 0.0	3.0 ± 0.2	$\mathbf{100.0 \pm 0.0}$
RIG + ACRO	None	$\mathbf{15.0 \pm 3.5}$	$4.0 \pm 1.$	$\mathbf{12.0 \pm 0.2}$	100.0 ± 0.0
RIG + PCLaSt	None	10.0 ± 0.5	4.0 ± 1.8	10.0 ± 0.1	90.0 ± 5
Low-Level Planner + PCLaSt	None	86.7 ± 3.4	69.3 ± 3.4	50.0 ± 4.3	$\pm$
n-Level Planner + PCLaSt	None	$\mathbf{97.78 \pm 4.91}$	$\mathbf{89.52 \pm 10.21}$	$\mathbf{89.11 \pm 10.38}$	95.0 ± 1.54

Iterative Dynamic Game in RL

This page is left blank intentionally.

Inculcating robustness into multistage decision policies





Problem Setup

- To quantify the brittleness, we optimize the stage cost

$$\max_{\boldsymbol{v}_t \sim \psi \in \Psi} \left[\sum_{t=0}^{T} \underbrace{c(\boldsymbol{x}_t, \boldsymbol{u}_t)}_{\text{nominal}} - \gamma \underbrace{g(\boldsymbol{v}_t)}_{\text{adversarial}} \right]$$

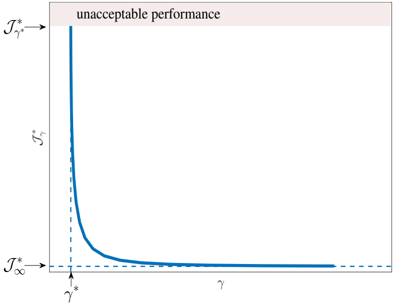
- To mitigate lack of robustness, we optimize the *cost-to-go*

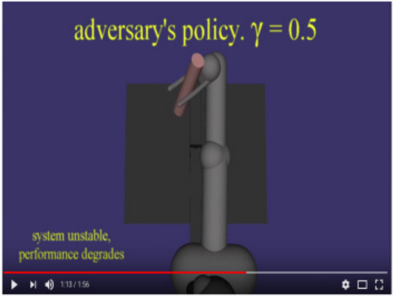
$$c_t(\boldsymbol{x}_t, \pi, \psi) = \min_{\boldsymbol{u}_t \sim \boldsymbol{\pi}} \max_{\boldsymbol{v}_t \sim \psi} \left(\sum_{t=0}^{T-1} \ell_t(\boldsymbol{x}_t, \boldsymbol{u}_t, \boldsymbol{v}_t) + L_T(\boldsymbol{x}_T) \right),$$

- and seek a saddle point equilibrium policy that satisfies

$$c_t(\boldsymbol{x}_t, \pi^*, \psi) \leq c_t(\boldsymbol{x}_t, \pi^*, \psi^*) \leq c_t(\boldsymbol{x}_t, \pi, \psi^*),$$

Results: Brittleness Quantification

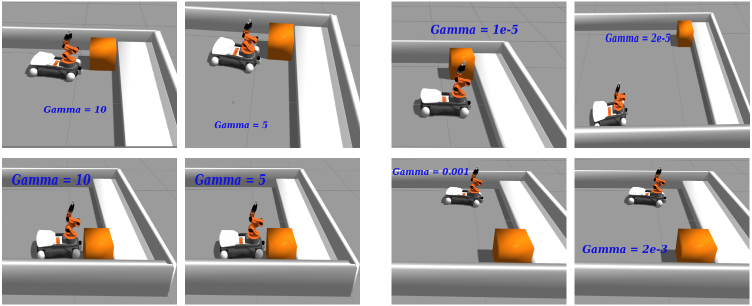




Results: Iterative Dynamic Game

$x_1^\star$

$x_2^\star$



End pose of the KUKA platform with our iDG formulation given different goal states and γ-values.

Mixed H_2/H_∞ Policy Optimization in RL

This page is left blank intentionally.

Talk Outline and Overview

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview
Risk-sensitive control
Contributions

Setup
Assumptions
Optimal Gain

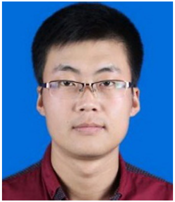
Model-based PO
Outer loop
Stabilization and Convergence

Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

- Policy Optimization and Stochastic Linear Control
  - Connections to risk-sensitive control;
  - Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ control theory.
- The case for convergence analysis in stochastic PO.
  - Kleinman's algorithm, *redux.*
  - Kleiman's algorithm in an iterative best response setting;
  - PO Convergence in best response settings.
- Robustness margins in model- and sampling- settings.
  - PO as a discrete-time nonlinear system;
  - Kleiman and input-to-state-stability;
  - Robust policy optimization as a small-input stable state optimization algorithm

Credits

Leilei Cui



Postdoc, MIT

Zhong-Ping Jiang

Professor, NYU

Research Significance

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview
Risk-sensitive control
Contributions

Setup
Assumptions
Optimal Gain

Model-based PO
Outer loop
Stabilization and Convergence

Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

- (Deep) RL and modern AI
  - Robotic manipulation (Levine et al., 2016), text-to-visual processing (DALL-E), Atari games (Mnih et al., 2013), e.t.c.
  - Policy optimization (PO) is fundamental to modern AI algorithms' success.
  - Major success story: functional mapping of observations to policies.
  - But how does it work?

Policy Optimization – General Framework

Continuous-Time Stochastic Policy Optimization
Lekan Molu
Outline and Overview
Risk-sensitive control
Contributions
Setup
Assumptions
Optimal Gain
Model-based PO
Outer loop
Stabilization and Convergence
Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

- PO encapsulates policy gradients (Kakade, 2001) or PG, actor-critic methods (Vrabie and Lewis, 2011), trust region PO Schulman et al. (2015), and proximal PO methods (Schulman et al., 2017).
- PG particularly suitable for complex systems.

$$\begin{aligned} &\min J(K) \\ &\text{subject to } K \in \mathcal{K} \end{aligned} \tag{1}$$

where $\mathcal{K} = \{K_1, K_2, \cdots, K_n\}$.

- $J(K)$ could be tracking error, safety assurance, goal-reaching measure of performance e.t.c. required to be satisfied.

Policy Optimization – Open questions

- Gradient-based data-driven methods: prone to divergence from true system gradients.
  - Challenge I: Optimization occurs in non-convex objective landscapes.
    - Get performance certificates as a mainstay for control design: Coerciveness property (Hu et al., 2023).
  - Challenge II: Taming PG's characteristic high-variance gradient estimates (REINFORCE, NPG, Zeroth-order approx.).
    - Hello, (linear) robust ($\mathcal{H}_\infty$-synthesis) control!

Policy Optimization – Open questions

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview
Risk-sensitive control
Contributions
Setup
Assumptions
Optimal Gain
Model-based PO
Outer loop
Stabilization and Convergence
Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

- Challenge III: Under what circumstances do we have convergence to a desired equilibrium in RL settings?
- Challenge IV: Stochastic control, not deterministic control settings.
  - models involving round-off error computations in floating point arithmetic calculations; the stock market; protein kinetics.
- Challenge V: Continuous-time RL control.
  - Very little theory. Lots of potential applications encompassing rigid and soft robotics, aerospace or finance engineering, protein kinetics.

(Non-exhaustive) Lit. Landscape on PO Theory

Literature landscape	Cont. time (Kalman '61, Luenberger '63)	Stochastic. LQR (Kalman '60)	Cont. Phase	LEQG or Mixed H_2/H_∞	Finite/Infinite Horizon
Fazel (2018)	No	No	Yes	No	Finite-horizon
Mohammadi (TAC -- 2020)	Yes	No	Yes	No	Finite-Horizon
Zhang (2019)	Yes	Yes (Gaussian)	Yes	Yes	Inf-horizon
Gravell (2021)	No	Multiplicative	Yes	No	Inf-horizon
Zhang (2020)	No	No	Yes	Yes	Rand-horizon
Molu (2022)	Yes	Yes (Brownian)	Yes	Yes	Inf-Horizon
Cui & Molu (2023)	Yes	Yes (Brownian)	Yes	Yes	Inf-Horizon

Outline and Overview
Risk-sensitive control
Contributions
Setup
Assumptions
Optimal Gain
Model-based PO
Outer loop
Stabilization and Convergence
Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

Model-based Policy Iteration

Algorithm 1: (Model-Based) PO via Policy Iteration

Input: Max. outer iteration $\bar{p}$, $q = 0$, and an $\epsilon > 0$;
Input: Desired risk attenuation level $\gamma > 0$;
Input: Minimizing player's control matrix $R \succ 0$.

Compute $(K_0, L_0) \in \mathcal{K}$; ▷ From [24, Alg. 1];
Set $P_{K,L}^{0,0} = Q_K^0$; ▷ See equation (9);
for $p = 0, \ldots, \bar{p}$ **do**
  Compute Q_K^p and A_K^p ▷ See equation (9);
  Obtain P_K^p by evaluating K_p on (10);
  while $\|P_K^p - P_{K,L}^{p,q}\|_F \le \epsilon$ **do**
    Compute $L_{q+1}(K_p) := \gamma^{-2} D^\top P_{K,L}^{p,q}$;
    Solve (11) until $\|P_K^p - P_{K,L}^{p,q}\|_F \le \epsilon$;
    $\bar{q} \leftarrow q + 1$
  end
  Compute $K_{p+1} = R^{-1} B^\top P_{K,L}^{p,\bar{q}}$ ▷ See (11b) ;
end

Convergence of the Inner Loop Iteration

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview
Risk-sensitive control
Contributions
Setup
Assumptions
Optimal Gain
Model-based PO
Outer loop
Stabilization and Convergence
Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

Theorem 3

For a $K \in \check{\mathcal{K}}$, and for any $(p, q) \in \mathbb{N}_+$, there exists $\beta(K) \in \mathbb{R}$ such that

$$Tr(P_K^p - P_{K,L}^{p,q+1}) \leq \beta(K) Tr(P_K^p - P_{K,L}^{p,q}). \tag{24}$$

Remark 2

*As seen from Lemma 5, $P_K^p - P_{K,L}^{p,q} \succeq 0$. By the norm on a matrix trace (**?**, Lemma 13) and the result of Theorem 3, we have $\|P_K - P_{K,L}^{p,q}\|_F \leq Tr(P_K - P_{K,L}^{p,q}) \leq \beta(K) Tr(P_K)$, i.e. $P_{K,L}^{p,q}$ exponentially converges to P_K in the Frobenius norm.*

Robustness Analyses

- Define $\tilde{P} = P_K - \hat{P}_K$ and $\tilde{K} = K - \hat{K}$.
- Keep $|\tilde{K}| < \epsilon$, start with a $K \in \mathcal{K}$: iterates stay in $\mathcal{K}$.

Lemma 7 (Lemma 10, C&M, '23)

For any $K \in \mathcal{K}$, there exists an $e(K) > 0$ such that for a perturbation $\tilde{K}$, $K + \tilde{K} \in \mathcal{K}$, as long as $\|\tilde{K}\| < e(K)$.

Theorem 6

The inexact outer loop is small-disturbance ISS. That is, for any $h > 0$ and $\hat{K}_0 \in \mathcal{K}_h$, if $\|\tilde{K}\| < f(h)$, there exist a $\mathcal{KL}$-function $\beta_1(\cdot,\cdot)$ and a $\mathcal{K}_\infty$-function $\gamma_1(\cdot)$ such that

$$\|P^p_{\hat{K}} - P^\star\| \leq \beta_1(\|P^0_{\hat{K}} - P^\star\|, p) + \gamma_1(\|\tilde{K}\|). \tag{37}$$

Inner Loop Robustness

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview
Risk-sensitive control
Contributions

Setup
Assumptions
Optimal Gain

Model-based PO
Outer loop
Stabilization and Convergence

Sampling-based PO
Discrete-time system
Sampling-based nonlinear system
Robustness Analyses

Theorem 7

Assume $\|\tilde{L}_q(K_p)\| < e$ for all $q \in \mathbb{N}_+$. There exists $\hat{\beta}(K) \in [0, 1)$, and $\lambda(\cdot) \in \check{\mathcal{K}}_\infty$, such that

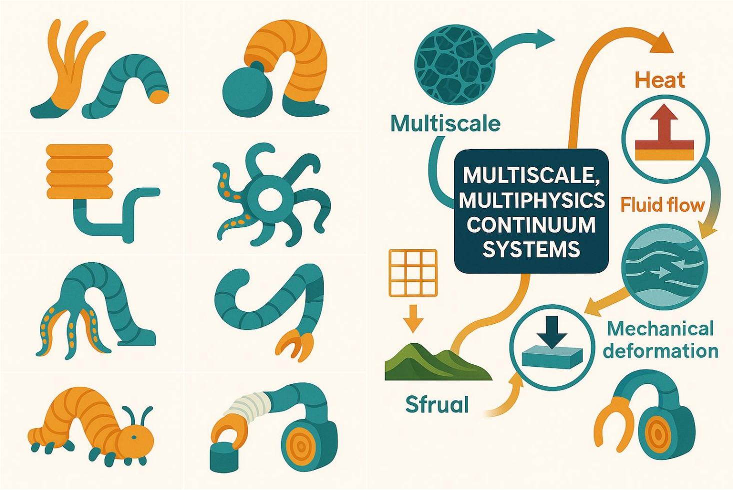
$$\|\hat{P}_{K,L}^{p,q} - P_{K,L}^{p,q}\|_F \le \hat{\beta}^{q-1}(K) Tr(P_{K,L}^{p,q}) + \lambda(\|\tilde{L}\|_\infty). \tag{42}$$

- From Theorem 7, as $q \to \infty$, $\hat{P}_{K,L}^{p,q}$ approaches the solution P_K and enters the ball centered at $P_{K,L}^{p,q}$ with radius proportional to $\|\tilde{L}\|_\infty$.
- The proposed inner-loop iterative algorithm well approximates $P_{K,L}^{p,q}$.

Morphological Computation in Emergent Robotic Systems

This page is left blank intentionally.

Soft Robotic Systems



Credit: Microsoft CoPilot.

The Piecewise Constant Strain (PCS) Cosserat Model

Octopus robot. Courtesy: IEEE Spectrum

Picture generated by Google Gemini

$$\underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}_a \boldsymbol{J} d\boldsymbol{X}\right]}_{\boldsymbol{M}(\boldsymbol{q})} \ddot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} J^\top \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \boldsymbol{\mathcal{M}}_a \boldsymbol{J} d\boldsymbol{X}\right]}_{\boldsymbol{C}_1(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} +$$

$$\underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}_a \dot{\boldsymbol{J}} d\boldsymbol{X}\right]}_{\boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{D}} \boldsymbol{J} \|\boldsymbol{J}\dot{\boldsymbol{q}}\|_p d\boldsymbol{X}\right]}_{\boldsymbol{D}(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} -$$

$$\underbrace{(1-\rho_f/\rho)\left[\int_0^{L_N} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}} \mathrm{Ad}_{\boldsymbol{g}}^{-1} d\boldsymbol{X}\right]}_{\boldsymbol{N}(q)} \mathrm{Ad}_{\boldsymbol{g}_r}^{-1} \boldsymbol{\mathcal{G}} - \underbrace{\boldsymbol{J}^\top(\bar{\boldsymbol{X}}) \boldsymbol{\mathcal{F}}_p}_{\boldsymbol{F}(\boldsymbol{q})}$$

$$\underbrace{-\int_0^{L_N} \boldsymbol{J}^\top \left[\nabla_x \boldsymbol{\mathcal{F}}_i - \nabla_x \boldsymbol{\mathcal{F}}_a + \mathrm{ad}^\star_{\eta_n} (\boldsymbol{\mathcal{F}}_i - \boldsymbol{\mathcal{F}}_a)\right] d\boldsymbol{X}}_{\boldsymbol{u}(\boldsymbol{q})} = 0,$$

Renda et al. T-RO 2016

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

SoRo's control computational complexity is hard!

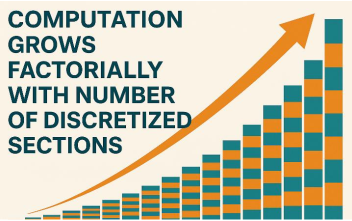
Structural Properties and Control of Soft Robots Modeled as Discrete Cosserat Rods

Lekan Molu and Shaoru Chen

***Abstract*—Soft robots featuring approximate finite-dimensional reduced-order models (undergoing small deformations) are increasingly becoming paramount in literature and applications. In this paper, we consider the piecewise constant strain (PCS) discrete Cosserat model whose dynamics admit the standard Newton-Euler dynamics for a kinetic model. Contrary to popular convention that soft robots under these modeling assumptions admit similar mechanical characteristics to rigid robots, the schemes employed to arrive at the properties for soft robots under finite deformation show a far dissimilarity to those for rigid robots. We set out to first correct the false premise behind this syllogism: from first principles, we established the structural properties of soft slender robots undergoing finite deformation under a discretized PCS assumption; we then utilized these properties to prove the stability of designed proportional-derivative controllers for manipulating the strain states of a prototypical soft robot under finite deformation. Our newly derived results are illustrated by numerical examples on a single arm of the Octopus robot and demonstrate the efficacy of our designed controller based on the derived kinetic properties. This work rectifies previously disseminated kinetic properties of discrete Cosserat-based soft robot models with greater accuracy in proofs and clarity.**

Nonlinear partial differential equations (PDEs) are the standard mathematical machinery for modeling continuum structures with distributed mass. And for soft robots exhibiting infinite degrees-of-freedom (DoF), nonlinear PDEs readily come in handy. However, scanty theory exists for nonlinear PDE analyses. To circumvent the complexity of PDE analyses, researchers have so far exploited approximate finite-dimensional ordinary differential equations (ODEs) [7] for analysis on spatially reduced models.

Tractable reduced-order mathematical models are typically formulated by restricting the range of shapes of the continuum robot to a finite-dimensional functional space over a curve that parameterizes the robot. This is equivalent to taking finite nodal points on the soft robot's body and approximating the dynamics along discretized nodal sections by an ODE. An aggregated ODE of all discretized sections can then be used to model the dynamics of the entire discretized continuum robot. A paramount example is the discrete Cosserat model of Renda et al. [18] whereupon the nonlinear PDE that describes the robot's kinetics in exact form is abstracted to standard Newton-Euler ODEs via



Enter Singularly Perturbed Systems

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

General SPT formulation.

Multiphysics, multiscale soft system.

Picture credit: Google Gemini.

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{f}(\boldsymbol{z}_1, \boldsymbol{z}_2, 0, \boldsymbol{u}_s, t),\ \ \boldsymbol{z}_1(t_0) = \boldsymbol{z}_1(0),$$
$$0 = \boldsymbol{g}(\boldsymbol{z}_1, \boldsymbol{z}_2, 0, 0, t).$$

Set ϵ to 0 → Slow subsystem

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

Fast subsystem on time scale: $T = t/\epsilon$

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

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

Singularly Perturbed Soft Cosserat Robot

Aggregate the robot's distributed mass, $\mathcal{M}$, inertia into a core active component, $\mathcal{M}_i^{\text{core}}$, and set the passive components as $\mathcal{M}^{\text{pert}} = \mathcal{M} \setminus \mathcal{M}^{\text{core}}$

Then the mass and Coriolis forces adopts the following representation

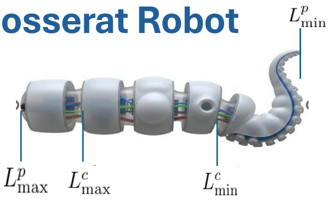
$$\text{where} \quad \boldsymbol{M}^p = \int_{L^p_{\min}}^{L^p_{\max}} \boldsymbol{J}^\top \mathcal{M}^{pert} \boldsymbol{J} dX$$

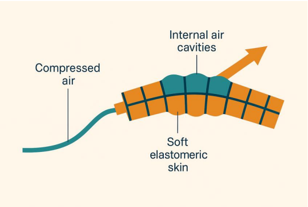
$$\boldsymbol{M}(\boldsymbol{q}) = (\boldsymbol{M}^c + \boldsymbol{M}^p)(\boldsymbol{q}), \ \boldsymbol{N} = (\boldsymbol{N}^c + \boldsymbol{N}^p)(\boldsymbol{q}),$$

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

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

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





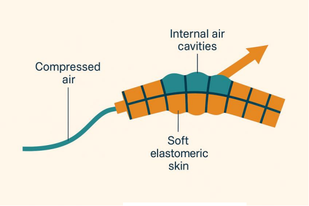
Picture credit: Google Gemini.

Dynamics Separation with Perturbation Parameter

The mass matrix then decomposes as

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

$M^c(q)$ and $M^p(q)$ are invertible (Molu & Chen, CDC 2024)



Introducing the perturbation parameter, $\epsilon = \|M^p\| / \|M^c\|$ We may define the matrix, $\bar{M}^p = M^p/\epsilon$

So that we can write,

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

where

$$s = \begin{bmatrix} s_{\text{fast}} \\ s_{\text{slow}} \end{bmatrix} = \begin{bmatrix} F^c + N^c \text{Ad}_{g_r}^{-1} \mathcal{G} - [C_1^c + C_2^c + D^c] z_{\text{fast}} \\ F^p + N^p \text{Ad}_{g_r}^{-1} \mathcal{G} - [C_1^p + C_2^p + D^p] z_{\text{slow}} \end{bmatrix}. \tag{13}$$

Singularly perturbed soft robot form

Suppose that

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

Then, we may write

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

Fast subdynamics extraction

Set $T = t/\epsilon$, with $dT/dt = 1/\epsilon$

Then, $\dot{\boldsymbol{z}}_{\text{fast}} = \dfrac{d\boldsymbol{z}_{\text{fast}}}{dt} \equiv \dfrac{1}{\epsilon}\dfrac{d\boldsymbol{z}_{\text{fast}}}{dT} \triangleq \dfrac{1}{\epsilon}\boldsymbol{z}'_{\text{fast}}$

and $\epsilon \dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{z}'_{\text{slow}}$.

So that,

$$\begin{aligned} \boldsymbol{z}'_{\text{fast}} &= \epsilon \boldsymbol{\mathcal{H}}^{-1}_{\text{fast}} (\boldsymbol{s}_{\text{fast}} + \boldsymbol{u}_{\text{fast}}) - \boldsymbol{\mathcal{H}}^{-1}_{\text{fast}} \boldsymbol{\mathcal{H}}^{\text{fast}}_{\text{slow}} \boldsymbol{z}'_{\text{slow}} \\ \boldsymbol{z}'_{\text{slow}} &= \boldsymbol{\mathcal{H}}^{-1}_{\text{slow}} (\boldsymbol{s}_{\text{slow}} - \boldsymbol{u}_{\text{slow}}) - \boldsymbol{\mathcal{H}}^{-1}_{\text{fast}} (\boldsymbol{s}_{\text{fast}} - \boldsymbol{u}_{\text{fast}}) \end{aligned}$$

A backstepping nonlinear multi-scale controller

Theorem 1: The control law

$$q_{\text{fast}}^{d}(t_f) - q_{\text{fast}}(t_f) + q_{\text{fast}}^{\prime d}(t_f)$$

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

Where,

$$[\theta^\top, \phi^\top]^\top = [q_{\text{fast}}^\top, z_{\text{fast}}^{\prime\top}]^\top \text{ where } \theta' = \epsilon z_{\text{fast}}.$$

Theorem 2: Under the tracking error $e_2 = \phi - \nu$ and matrices $(K_p, K_q) = (K_p^\top, K_q^\top) > 0$, the control input

$$u_{\text{fast}} = \frac{1}{\epsilon}\mathcal{H}_{\text{fast}}[q_{\text{fast}}^{\prime\prime d} + e_1 - 2e_2 - K_q^\top (K_q K_q^\top)^{-1} K_p e_1] + \frac{1}{\epsilon}\mathcal{H}_{\text{slow}}^{\text{fast}} z_{\text{slow}}' - s_{\text{fast}} \quad (24)$$

exponentially stabilizes the fast subdynamics (18).

Theorem 3: The control law

$$u_{\text{slow}} = \mathcal{H}_{\text{slow}}(e_1 - e_2 - e_3 + \ddot{q}_{\text{fast}}^{d}) - s_{\text{slow}}$$

exponentially stabilizes the slow subdynamics.

A backstepping nonlinear multi-scale controller

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,

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

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.

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

Numerical Results

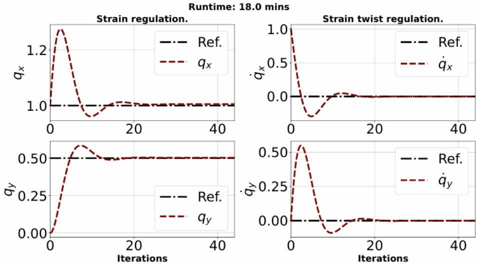


Fig. 2. Backstepping control on the singularly perturbed soft robot system with 10 discretized pieces, divided into 6 fast and 4 slow pieces. For a tip load of $\mathcal{F}_p^y = 10\,N$, the backstepping gains were set as $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ for a desired joint configuration $\xi^d = [0, 0, 0, 1, 0.5, 0]^\top$ and $\eta^d = \mathbf{0}_{6\times1}$ 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 STEADY STATE.

Numerical Results – System Setup

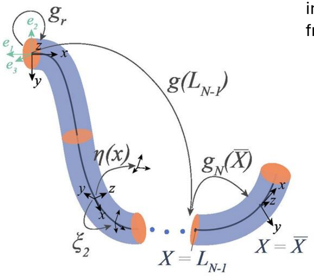


Fig. 1. Simplified configuration of an Octopus arm, reprinted from Molu and Chen [9].

The robot's z-axis is offset in orientation from the inertial frame by -90 deg so that a transformation from the base to inertial frames is

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

Tip wrench at $\bar{X} = L$ is ,

$$\boldsymbol{\mathcal{F}}_p = \text{diag}\left(\boldsymbol{R}^\top(L), \boldsymbol{R}^\top(L)\right) \begin{pmatrix} \boldsymbol{0}_{3\times1} & 0 & 10 & 0 \end{pmatrix}^\top$$

Param	Symbol	Value
Reynold's #		0.82
Young's Mod.	**E**	110*kPa*
Shear visc.	**J**	3 *kPa*

Numerical Results – System Setup

Param	Symbol	Value
Reynold's #		0.82
Young's Mod.	**E**	110*kPa*
Shear visc.	**J**	3 *kPa*
Bending 2nd Inertia	$I_y = I_z = \pi r^4/4$	
Torsion 2nd Inert	$I_x = \pi r^4/2$	
Material abscissa	$L = 2m$	
Poisson ratio	ρ	0.45
Mass density	$\mathcal{M} = \rho \cdot \mathrm{diag}([I_x, I_y, I_z, A, A, A])$	
Drag stiffness matrix	$\boldsymbol{D} = -\rho_w \nu^T \nu \check{\boldsymbol{D}} \nu / \lvert\nu\rvert$	

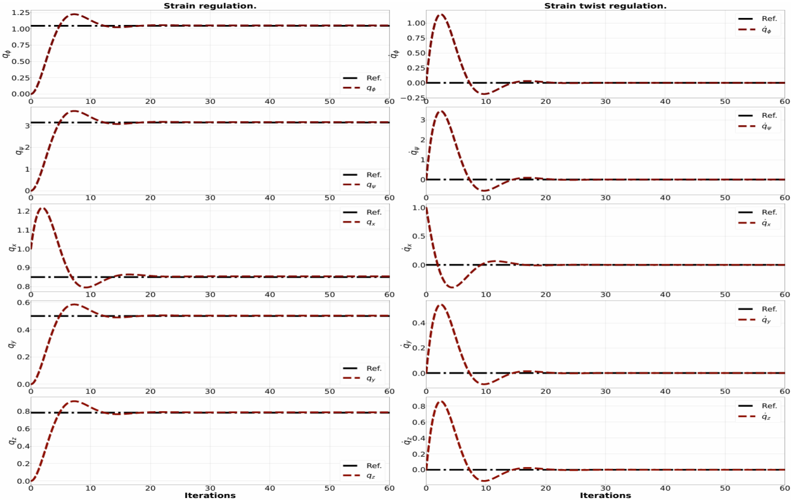


Fig. 3. Backstepping control on the singularly perturbed soft robot system with 10 pieces 4 slow and 6 fast sections.

This page is left blank intentionally.

Exogenous Markov Decision Process (Exo-MDP) Machinery

- Consider the tuple $\mathcal{M} := (\mathcal{X}, \mathcal{Z}, \mathcal{A}, T, R, H)$
  - Starting distribution $\mu \in \Delta(\mathcal{Z})$;
  - Agent receives observations $\{x_h\}_{h=1}^{H} \in \mathcal{X}$ from the emission function $q : \mathcal{Z} \to \Delta(\mathcal{X})$;
  - Agent transitions between latent states via $T : \mathcal{Z} \times \mathcal{A} \to \Delta(\mathcal{S})$;
    - And rewards by $R : \mathcal{X} \times \mathcal{A} \to \Delta([0, 1])$
- Trajectories: $(z_1, x_1, a_1, r_1, \cdots, z_H, a_H, r_H)$ from repeated interactions;
  - $z_1 \sim \mu_1(\cdot)$, $z_{h+1} \sim T(\cdot|z_h, a_h)$, $x_h \sim q(\cdot|z_h)$ and $r_h \sim R(x_h, a_h, x_{h+1})$ for all $h \in [H]$.
- Define $supp(q(\cdot|z)) = \{x \in \mathcal{X} | q(x|z) > 0\}$ for any z.

Exo-MDP Machinery

Block MDP assumption $supp(q(\cdot|z_1)) \cap sup(q(\cdot|z_2)) = \emptyset$ for all $z_1 \neq z_2$.

- Agent chooses $a \sim \pi(z_h|x_h)$
- There exists non-stationary episodic policies $\Pi_{NS} := \Pi^H \supseteq (\pi_1, \cdots, \pi_H)$;
- Optimal policy $\pi^\star = \text{argmax} V_{\pi \in \Pi_{NS}}(\pi)$;
  - For $V_{\pi \in \Pi_{NS}} = \sum_h = 1^H r_h$.
- EXO-BMDP: Essentially a Block MDP [1] such that the latent states admits the form $z = (s, e)$, where $s \in \mathcal{S}$, $e \in E$.
- $\mu(z) = \mu(s)\mu\xi$ and $T(z'|z, a) = T(s'|s, a)T_e(e'|e)$

AC State Algorithm

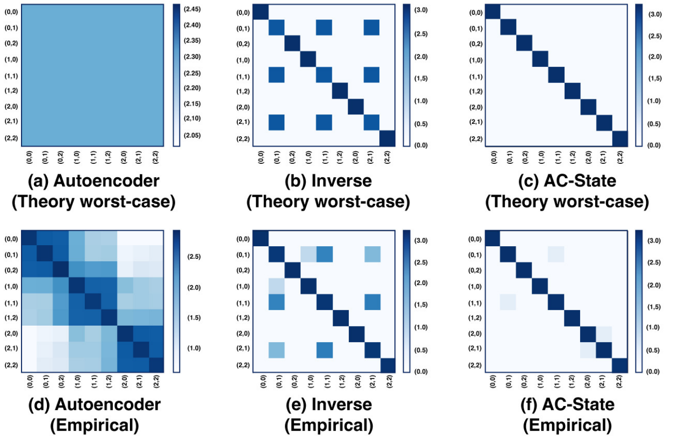
Algorithm 1 AC-State Algorithm for Latent State Discovery Using a Uniform Random Policy

Initialize observation trajectory x and action trajectory a. Initialize encoder f_θ. Assume any pair of states are reachable within exactly K steps and a number of samples to collect T, and a set of actions $\mathcal{A}$, and a number of training iterations N.
$x_1 \sim U(\mu(x))$
for t = 1, 2, ..., T **do**
    $a_t \sim U(\mathcal{A})$
    $x_{t+1} \sim \mathbb{P}(x'|x_t, a_t)$
for n = 1, 2, ..., N **do**
    $t \sim U(1, T)$ and $k \sim U(1, K)$
    $\mathcal{L} = \mathcal{L}_{\texttt{AC-State}}(f_\theta, t, x, a, k) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_t) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_{t+k})$
    Update θ to minimize $\mathcal{L}$ by gradient descent.

AC State in Action

Exogenous distractors riddance.

Agent Controllable States Representation



(a) Autoencoder (Theory worst-case)

(b) Inverse (Theory worst-case)

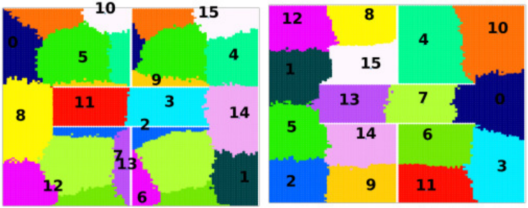
(c) AC-State (Theory worst-case)

(d) Autoencoder (Empirical)

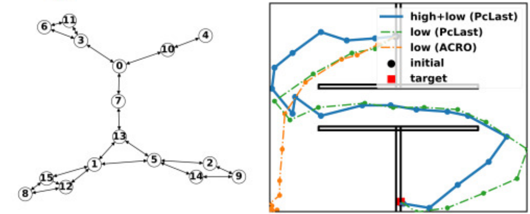
(e) Inverse (Empirical)

(f) AC-State (Empirical)

PCLAST Segmentation Results

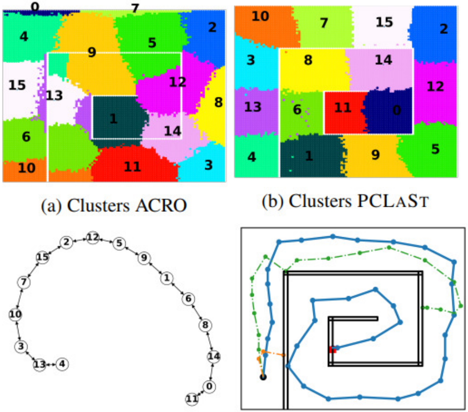


(a) Clusters ACRO (b) Clusters PCLAST



(c) State-transitions PCLAST (d) Planning Trajectories

PCLAST Segmentation Results

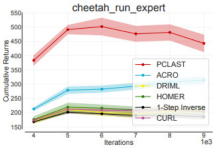


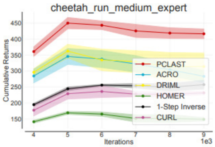
(a) Clusters ACRO (b) Clusters PCLAST

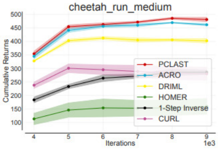
(c) State-transitions PCLAST (d) Planning Trajectories

Figure 6. Clustering, Abstract-MDP, and Planning are shown for *Maze-Spiral* environment. Details same as Figure 5.

PCLAST – Cheetah Environment







Morphological Computation

This page is left blank intentionally.

Appendix – SoRos

This page is left blank intentionally.

Morphological Computation – Overview

- The principle of morphological computation in nature
  - Morphology: shape, geometry, and mechanical properties.
  - Computation: sensorimotor information transmission among geometrical components.
- Morphology and computation in artificial robots
  - Cosserat Continua and reduced soft robot models.
  - **Reductions**: Structural Lagrangian properties and control.
- Towards real-time strain regulation and control
  - **Simplexity**: Hierarchical and fast versatile control with reduced variables.

Credits

Shaoru Chen

Postdoc, MSR

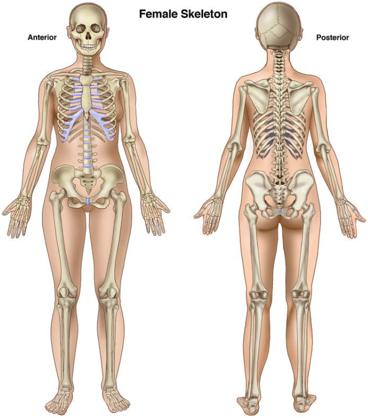
Lekan Molu

Senior Researcher, MSR

Morphology and computation

- Morphology: Emergent behaviors of natural organisms from complex sensorimotor nonlinear mechanical feedback from the environment.
  - Shape affecting behavioral response.
  - Geometrical Arrangement of motors such that processing and perception affect computational characteristics.
  - Mechanical properties that allow the engineering of emergent behaviors via adaptive environmental interaction.
- Computation: The information transformation among the system geometrical units, upon environmental perception, that effect morphological changes in shape and material properties.

MC in vertebrates – a case for soft designs



An adult human skeleton $\approxeq$ 11% of the body mass. ©Brittanica

- The arrangement and compliance of body parts, perception, and computation creates emergence of complex interactive behavior.
- Soft bodies seem critical to the emergence of adaptive natural behaviors.
- Morphological computation is crucial in the design of robots that execute adaptive natural behavior.

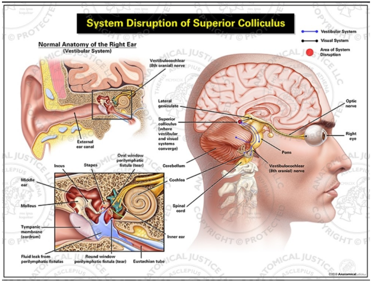
Simplexity in Morphological Computation

- Simplexity: Exploiting structure for effective control.
  - The geometrical tuning of the morphology and neural circuitry in the brain of mammals that simplify the perception and control of complex natural phenomena.
  - Not exactly simplified models or reduced complexity.
  - But rather, sparse connections and finite variables to execute adaptive sensorimotor strategies!
- Example: Saccades (focused eye movements) are controlled by (small) Superior Colliculus in the human brain.
  - Plug: Complex neural circuitry; simple control systems!

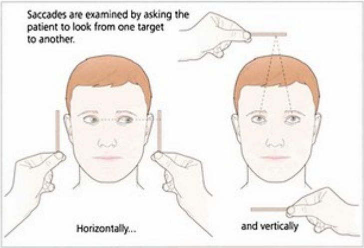
Simplexity: The Central Pattern Generator

- A neural mechanism (in vertebrates) that generates motor control with minimal parameters.
- CPG: Neurons and synapses couple to generate effective motor activation for rhythmic environmental motion.
  - In Lampreys, only two signals trigger swimming motion, for example!
  - This CPG enables indirect use of brain computational power via nonlinear feedback from stretch receptor neurons on Lamprey's skin.

Saccades and the Superior Colliculus



©Anatomical Justice.



Credit: Vision and Learning Center.

Morphing in Invertebrates: Cephalopods

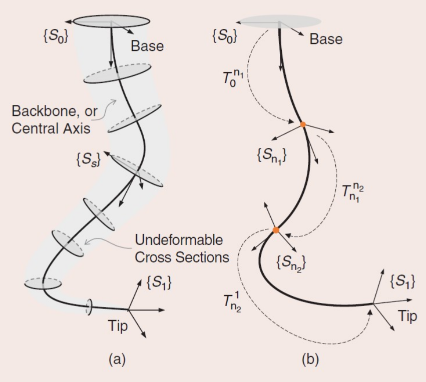
Cuttlefish. ©Monterey Bay Museum

Octopus. ©Smithsonian Magazine

The Octopus and Cuttlefish

- No exoskeleton, or spinal cord.
- A muscular hydrostat: transversal, longitudinal, and oblique muscles along richly innervated arms and mechanoreceptors:
  - Allows for bending, stretching, stiffening, and retraction.
  - Diverse compliance across eight arms imply sophisticated motion strategies in the wild!
- Simplexity enhanced by a peripheral nervous system and a central nervous system.

Soft Robot Mechanism in Focus

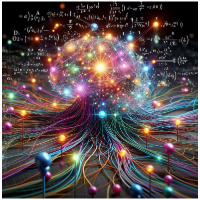


A continuum soft robot whose mechanics can be well-described with Cosserat rod theory. Reprinted from (**(author?)** [2])

- One dimension is quintessentially longer than the other two.
- Characterized by a central axis with undeformable discs that characterize deformable cross-sectional segments.
- Strain and deformation, via e.g. Cosserat rod theory, enables precise finite-dimensional mathematical models.

A Finite and Reliable Model

- A soft robot's usefulness is informed by control system that melds its body deformation with internal actuators.
- By design, this calls for a high-fidelity model or a delicate balancing of complex morphology and data-driven methods.



- Non-interpretable; non-reliable.
- $\times$Continuous coupled interaction between the material, actuators, and external affordances.

The case for model-based control

- Soft robots are infinite degrees-of-freedom continua i.e., PDEs are the main tools for analysis.
- Nonlinear PDE theory is tedious and computationally intensive.
- Notable strides in reduced-order, finite-dimensional mathematical models that induce tractability in continuum models.

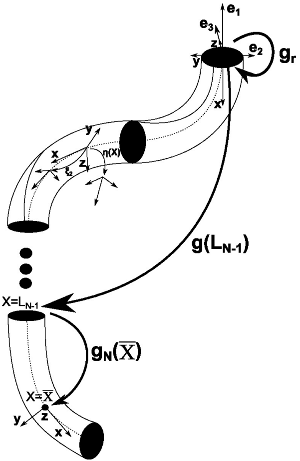
Tractable reduced-order models

- Morphoelastic filament theory: [8; 5; 3];
- Generalized Cosserat rod theory: [14; 1];
- The constant curvature model: [4];
- The piecewise constant curvature model: [15; 9]; and
- Ordinary differential equations-based discrete Cosserat model: [11; 10].

Cosserat-based piecewise constant strain model

- A discrete Cosserat model: **(author?)** [10].
  - Shapes defined by a finite-dimensional functional space, parameterized by a curve, $X : [0, L]$..
  - Assumes constant strains between finite nodal points on robot's body.
  - Strain-parameterized dynamics on a reduced special Euclidean-3 group (SE(3)).

The piecewise constant strain model



Credit: [10].

- C-space: $g(X) : X \to \mathbb{SE}(3) = \begin{pmatrix} \boldsymbol{R}(X) & \boldsymbol{p}(X) \\ 0^\top & 1 \end{pmatrix}$.
- Strain and twist vectors: $\{\boldsymbol{\eta}, \boldsymbol{\xi}\} \in \mathbb{R}^6$.
  - $\{\eta, \xi\} := \{\boldsymbol{q}, \dot{\boldsymbol{q}}\}$
- Strain field: $\breve{\eta}(X) = g^{-1}\partial g / \partial X$.
- Twist field: $\breve{\xi}(X) = g^{-1}\partial g / \partial t$.

The piecewise constant strain model

- $X \in [0, L]$ is divided into N intervals: $[0, L_1], \cdots, [L_{N-1}, L_N]$.
- In [10]'s proposition, the robot's mass divides into N discrete sections $\{\mathcal{M}_n\}_{n=1}^N$;
- Each with constant strain $\boldsymbol{\eta}_n$
- Strain field: $\breve{\boldsymbol{\eta}}(X) = g^{-1}\partial g/\partial X$.
- Twist field: $\breve{\boldsymbol{\xi}}(X) = g^{-1}\partial g/\partial t$.

Dynamic equations

From the continuum equations for a cable-driven soft arm [[12]], we can derive the following dynamic equation [[10]]:

$$\underbrace{\left[\int_0^{L_N} J^T \mathcal{M}_a \boldsymbol{J} dX\right]}_{\boldsymbol{M}(\boldsymbol{q})} \ddot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} \boldsymbol{J}^T \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \mathcal{M}_a \boldsymbol{J} dX\right]}_{C_1(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} + \underbrace{\left[\int_0^{L_N} J^T \mathcal{M}_a \dot{\boldsymbol{J}} dX\right]}_{C_2(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}}$$

$$+ \underbrace{\left[\int_0^{L_N} \boldsymbol{J}^T \mathcal{D} \boldsymbol{J} \|\boldsymbol{J}\dot{\boldsymbol{q}}\|_p dX\right]}_{D(\boldsymbol{q},\dot{\boldsymbol{q}})} \dot{\boldsymbol{q}} - \underbrace{(1 - \rho_f/\rho) \left[\int_0^{L_N} \boldsymbol{J}^T \mathcal{M} \mathrm{Ad}_{\boldsymbol{g}}^{-1} dX\right]}_{\boldsymbol{N}(\boldsymbol{q})} \mathrm{Ad}_{\boldsymbol{g}_r}^{-1} \mathcal{G}$$

$$- \underbrace{\boldsymbol{J}(\bar{X})^T \mathcal{F}_p}_{\boldsymbol{F}(\boldsymbol{q})} - \underbrace{\int_0^{L_N} \boldsymbol{J}^T \left[\nabla_x \mathcal{F}_i - \nabla_x \mathcal{F}_a + \mathrm{ad}^\star_{\boldsymbol{\xi}_n} (\mathcal{F}_i - \mathcal{F}_a)\right] dX}_{\boldsymbol{\tau}(\boldsymbol{q})} = 0, \tag{1}$$

Structural properties – mass inertia operator

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \left[\boldsymbol{C}_1(q,\dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\dot{\boldsymbol{q}} = \boldsymbol{F}(\boldsymbol{q}) + \boldsymbol{N}(\boldsymbol{q})\mathrm{Ad}_{\boldsymbol{g}_r}^{-1}\mathcal{G} + \tau(\boldsymbol{q}) - \boldsymbol{D}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}. \tag{2}$$

Property 1 (Boundedness of the Mass Matrix)

The mass inertial matrix $\boldsymbol{M}(\boldsymbol{q})$ is uniformly bounded from below by $m\boldsymbol{I}$ where m is a positive constant and $\boldsymbol{I}$ is the identity matrix.

Proof of Property 1.

This is a restatement of the lower boundedness of $\boldsymbol{M}(\boldsymbol{q})$ for fully actuated n-degrees of freedom manipulators [[13]]. □

Structural properties – parameters Identification

Property 2 (Linearity-in-the-parameters)

There exists a constant vector $\boldsymbol{\Theta} \in \mathbb{R}^{l}$ *and a regressor function* $Y(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}) \in \mathbb{R}^{N \times l}$ *such that*

$$\boldsymbol{M}(\boldsymbol{q})\ddot{(\boldsymbol{q})} + \left[\boldsymbol{C}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})\right] \dot{\boldsymbol{q}} - \boldsymbol{F}(\boldsymbol{q})\boldsymbol{N}(\boldsymbol{q})Ad_{\boldsymbol{g}_r}^{-1}\mathcal{G} = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\Theta. \tag{3}$$

Structural properties – skew symmetry of system inertial forces

Property 3 (Skew symmetric property)

The matrix $\dot{M}(\boldsymbol{q}) - 2\left[C_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + C_2(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]$ *is skew-symmetric.*

Skew-symmetric of robot's mass and Coriolis forces

By Leibniz's rule, we have

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) = \frac{d}{dt}\left(\int_0^{L_N} \boldsymbol{J}^T \mathcal{M}_a \boldsymbol{J} dX\right) = \int_0^{L_N} \frac{\partial}{\partial t}\left(\boldsymbol{J}^T \mathcal{M}_a \boldsymbol{J}\right) dX$$
$$\triangleq \int_0^{L_N} \left(\dot{\boldsymbol{J}}^T \mathcal{M}_a \boldsymbol{J} + \boldsymbol{J}^T \dot{\mathcal{M}}_a \boldsymbol{J} + \boldsymbol{J}^T \mathcal{M}_a \dot{\boldsymbol{J}}\right) dX. \quad (4)$$

Therefore, $\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\left[C_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + C_2(\boldsymbol{q}, \dot{\boldsymbol{q}})\right]$ becomes

$$\int_0^{L_N} \left(\dot{\boldsymbol{J}}^\top \mathcal{M}_a \boldsymbol{J} + \boldsymbol{J}^\top \dot{\mathcal{M}}_a \boldsymbol{J} + \boldsymbol{J}^\top \mathcal{M}_a \dot{\boldsymbol{J}}\right) dX - 2\int_0^{L_N} \left(\boldsymbol{J}^\top \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \mathcal{M}_a \boldsymbol{J} + \boldsymbol{J}^\top \mathcal{M}_a \dot{\boldsymbol{J}}\right) dX \quad (5)$$

$$\triangleq \int_0^{L_N} \left(\dot{\boldsymbol{J}}^\top \mathcal{M}_a \boldsymbol{J} + \boldsymbol{J}^\top \dot{\mathcal{M}}_a \boldsymbol{J} - \boldsymbol{J}^\top \mathcal{M}_a \dot{\boldsymbol{J}}\right) dX - 2\int_0^{L_N} \boldsymbol{J}^\top \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \mathcal{M}_a \boldsymbol{J} dX. \quad (6)$$

Skew-Symmetric Property Proof

Similarly, $-\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\left[\boldsymbol{C}_1(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\right]^\top$ expands as

$$-\dot{\boldsymbol{M}}^\top(\boldsymbol{q}) + 2\left[\boldsymbol{C}_1^\top(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}_2^\top(\boldsymbol{q},\dot{\boldsymbol{q}})\right] =$$

$$\int_0^{L_N} dX^\top \left(-\boldsymbol{J}^\top \mathcal{M}_a \dot{\boldsymbol{J}} - \boldsymbol{J}^\top \dot{\mathcal{M}}_a \boldsymbol{J} - \dot{\boldsymbol{J}}^\top \mathcal{M}_a \boldsymbol{J}\right) + 2\int_0^{L_N} dX^\top \left(\boldsymbol{J}^\top \mathcal{M}_a \mathrm{ad}_{\boldsymbol{J}\dot{\boldsymbol{q}}} \boldsymbol{J} + \dot{\boldsymbol{J}}^\top \mathcal{M}_a \boldsymbol{J}\right)$$

$$\triangleq \int_0^{L_N} \left(\boldsymbol{J}^\top \mathcal{M}_a \dot{\boldsymbol{J}} - \dot{\boldsymbol{J}}^\top \mathcal{M}_a \boldsymbol{J} - \boldsymbol{J}^\top \dot{\mathcal{M}}_a \boldsymbol{J}\right) dX - 2\int_0^{L_N} J^\top \mathrm{ad}^\star_{\boldsymbol{J}\dot{\boldsymbol{q}}} \mathcal{M}_a \boldsymbol{J} dX \tag{7}$$

which satisfies the identity:

$$\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\left[\boldsymbol{C}_1(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})\right] = -\left[\dot{\boldsymbol{M}}(\boldsymbol{q}) - 2\left[\boldsymbol{C}_1(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\right]^\top. \tag{8}$$

A fortiori, the skew symmetric property follows.

MC Takeaways: Simplexity

- Simplexity: Reliance on a few parameters to model an infinite-DoF system:

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

- Simplexity: From PDE to ODE, i.e. inifinite-dimensional analysis (Continuum PDE) to finite-dimensional ODE!

Control exploiting structural properties

Regarding the generalized torque $\boldsymbol{\tau}(\boldsymbol{q})$ as a control input, $\boldsymbol{u}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, feedback laws are sufficient for attaining a desired soft body configuration.

Theorem 1 (Cable-driven Actuation)

For positive definite diagonal matrix gains $\boldsymbol{K}_D$ and $\boldsymbol{K}_p$, without gravity/buoyancy compensation, the control law

$$\boldsymbol{u}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -\boldsymbol{K}_p \tilde{\boldsymbol{q}} - \boldsymbol{K}_D \dot{\boldsymbol{q}} - \boldsymbol{F}(\boldsymbol{q}) \tag{9}$$

under a cable-driven actuation globally asymptotically stabilizes system (2), where $\tilde{\boldsymbol{q}}(t) = \boldsymbol{q}(t) - \boldsymbol{q}^d$ is the joint error vector for a desired equilibrium point $\boldsymbol{q}^d$.

Computational Control exploiting structural properties

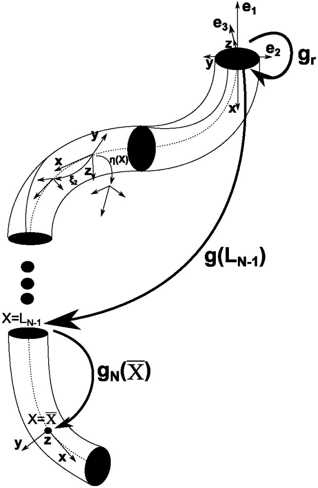
Corollary 2 (Fluid-driven actuation)

If the robot is operated without cables, and is driven with a dense medium such as pressurized air or water, then the term $F(\boldsymbol{q}) = 0$ *so that the control law* $\boldsymbol{u}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = -\boldsymbol{K}_p \tilde{\boldsymbol{q}} - \boldsymbol{K}_D \dot{\boldsymbol{q}}$ *globally asymptotically stabilizes the system.*

Proof.

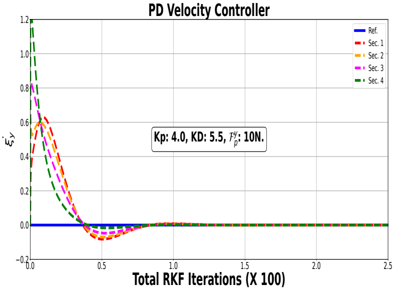
Proofs in Section V of **(author?)** [7]. □

Robot parameters

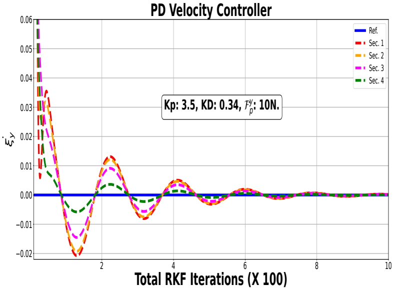


- Tip load in the $+y$ direction in the robot's base frame.
- Poisson ratio: 0.45; $\mathcal{M} = \rho[I_x, I_y, I_z, A, A, A]$ with $\rho = 2,000 kgm^{-3}$;
- $\boldsymbol{D} = -\rho_w \nu^T \nu \check{\boldsymbol{D}} \nu / |\nu|$.
- $X \in [0, L]$ discretized into 41 segments.

Computational Control exploiting structural properties

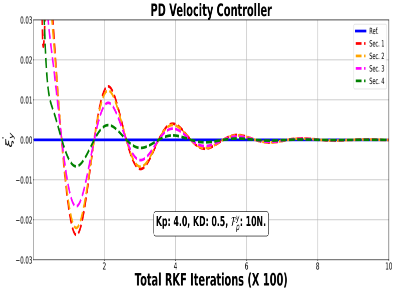


Cable-driven, strain twist setpoint terrestrial control.

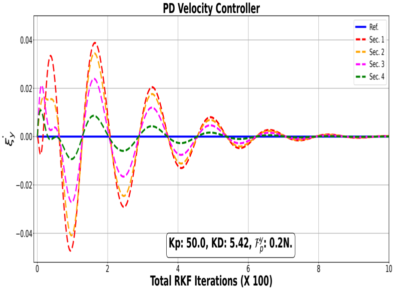


Fluid-actuated, strain twist setpoint terrestrial control.

Computational Control exploiting structural properties

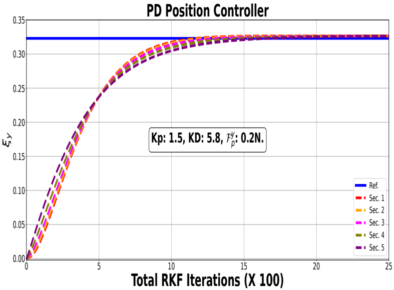


Fluid-actuated, strain twist setpoint underwater control.

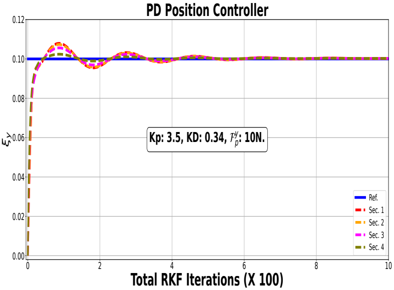


Cable-driven, strain twist setpoint regulation.

Computational Control exploiting structural properties



Cable-based position control with a small tip load, 0.2N.



Terrestrial position control.

Exploiting Mechanical Nonlinearity for Feedback!

This page is left blank intentionally.

Hierarchical Dynamics and Control

- Reaching steps towards the real-time strain control of multiphysics, multiscale continuum soft robots.
- Separate subdynamics — aided by a perturbing time-scale separation parameter.
- Respective stabilizing nonlinear backstepping controllers.
- Stability of the interconnected singularly perturbed. system.
- Fast numerical results on a single arm of the Octopus robot arm.

Decomposition of SoRo Rod Dynamics

- $\mathcal{M}_i^{\text{core}}$: composite mass distribution as a result of microsolid i's barycenter motion;
- $\mathcal{M}^{\text{pert}}$: motions relative to $\mathcal{M}_i^{\text{core}}$, considered as a perturbation;
- $\mathcal{M} = \mathcal{M}^{\text{pert}} \cup \mathcal{M}^{\text{core}}$.
- Introduce the transformation: $[\boldsymbol{q}, \dot{\boldsymbol{q}}] = [\boldsymbol{q}, \boldsymbol{z}]$, rewrite (2):

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

Dynamics separation

Suppose that $\boldsymbol{M}^p = \int_{L^p_{\text{min}}}^{L^p_{\text{max}}} \boldsymbol{J}^\top \boldsymbol{\mathcal{M}}^{pert} \boldsymbol{J} dX$, and $\boldsymbol{M}^c = \int_{L^c_{\text{min}}}^{L^c_{\text{max}}} \boldsymbol{J}^\top \mathcal{M}^{core} \boldsymbol{J} dX$, then,

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

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

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

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

Dynamics Separation

Furthermore, let

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

where $\boldsymbol{\mathcal{H}}_{\text{slow}}^{\text{fast}}$ denotes the decomposed mass of the perturbed sections of the robot relative to the core sections.

- Let robot's state, $\boldsymbol{x} = [\boldsymbol{q}^\top, \boldsymbol{z}^\top]^\top$ decompose as $\boldsymbol{q} = [\boldsymbol{q}_{\text{fast}}^\top, \boldsymbol{q}_{\text{slow}}^\top]^\top$ and $\boldsymbol{z} = [\boldsymbol{z}_{\text{fast}}^\top, \boldsymbol{z}_{\text{slow}}^\top]^\top$,
- Define $\bar{\boldsymbol{M}}^p = \boldsymbol{M}^p/\epsilon$, and let $\boldsymbol{u} = [\boldsymbol{u}_{\text{fast}}^\top, \boldsymbol{u}_{\text{slow}}^\top]^\top$ be the applied torque.

SoRo Dynamics Separation

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

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

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

SoRo Dynamics Separation

Premultiplying both sides by $\boldsymbol{I} - \epsilon\boldsymbol{\Delta}$, it can be verified that

$$\begin{bmatrix} \boldsymbol{\mathcal{H}}_{\text{fast}} & \bar{\boldsymbol{M}}_{12}^p \\ \boldsymbol{0} & \bar{\boldsymbol{M}}_{22}^p \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{z}}_{\text{fast}} \\ \epsilon\dot{\boldsymbol{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{s}_{\text{fast}} \\ \boldsymbol{s}_{\text{slow}} - \epsilon\bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{s}_{\text{fast}} \end{bmatrix} + \begin{bmatrix} \boldsymbol{u}_{\text{fast}} \\ \boldsymbol{u}_{\text{slow}} - \epsilon\bar{\boldsymbol{M}}_{21}^p \boldsymbol{\mathcal{H}}_{\text{fast}}^{-1} \boldsymbol{u}_{\text{fast}} \end{bmatrix} \tag{15}$$

which is in the standard singularly perturbed form (**??**):

$$\dot{\boldsymbol{z}}_1 = \boldsymbol{f}(\boldsymbol{z}_1, \boldsymbol{z}_2, \epsilon, \boldsymbol{u}_s, t), \;\; \boldsymbol{z}_1(t_0) = \boldsymbol{z}_1(0), \;\; \boldsymbol{z}_1 \in \mathbb{R}^{6N}, \tag{16a}$$

$$\epsilon\dot{\boldsymbol{z}}_2 = \boldsymbol{g}(\boldsymbol{z}_1, \boldsymbol{z}_2, \epsilon, \boldsymbol{u}_f, t), \;\; \boldsymbol{z}_2(t_0) = \boldsymbol{z}_2(0), \;\; \boldsymbol{z}_2 \in \mathbb{R}^{6N} \tag{16b}$$

SoRo Fast Subsystem Extraction

On the fast time scale $T = t/\epsilon$, with $dT/dt = 1/\epsilon$ so that,

$$\dot{\boldsymbol{z}}_{\text{fast}} = \frac{d\boldsymbol{z}_{\text{fast}}}{dt} \equiv \frac{1}{\epsilon}\frac{d\boldsymbol{z}_{\text{fast}}}{dT} \triangleq \frac{1}{\epsilon}\boldsymbol{z}'_{\text{fast}}$$

; and

$$\epsilon\dot{\boldsymbol{z}}_{\text{slow}} = \boldsymbol{z}'_{\text{slow}}.$$

Fast subdynamics:

$$\boldsymbol{z}'_{\text{fast}} = \epsilon\mathcal{H}^{-1}_{\text{fast}}(\boldsymbol{s}_{\text{fast}} + \boldsymbol{u}_{\text{fast}}) - \mathcal{H}^{-1}_{\text{fast}}\mathcal{H}^{\text{fast}}_{\text{slow}}\boldsymbol{z}'_{\text{slow}}, \tag{17a}$$

$$\boldsymbol{z}'_{\text{slow}} = \mathcal{H}^{-1}_{\text{slow}}(\boldsymbol{s}_{\text{slow}} - \boldsymbol{u}_{\text{slow}}) - \mathcal{H}^{-1}_{\text{fast}}(\boldsymbol{s}_{\text{fast}} - \boldsymbol{u}_{\text{fast}}) \tag{17b}$$

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

SoRo Slow Subsystem Extraction

- We let $\epsilon \to 0$ in (15), so that what is left, i.e.,

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

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

This page is left blank intentionally

Control of the Fast Strain Subdynamics

- Consider the transformation: $\begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{\phi} \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_{\text{fast}} \\ \boldsymbol{z}_{\text{fast}} \end{bmatrix}$ so that $\boldsymbol{\theta}' = \epsilon \boldsymbol{z}_{\text{fast}} \triangleq \boldsymbol{\nu} :=$ A virtual input.
- Let $\{\boldsymbol{q}^d_{\text{fast}}, \dot{\boldsymbol{q}}^d_{\text{fast}}\} = \{\boldsymbol{\xi}^d_1, \ldots, \boldsymbol{\xi}^d_{n_\xi}, \boldsymbol{\eta}^d_1, \ldots, \boldsymbol{\eta}^d_{n_\xi}\}_{\text{fast}}$ be the desired joint space configuration for the fast subsystem.

Theorem 3 ([6])

The control law

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

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

Control of the Fast Strain Subdynamics

Proof Sketch 1 (Proof of Theorem 3)

$$\boldsymbol{e}_1 = \boldsymbol{\theta} - \boldsymbol{q}_{fast}^{d}, \implies \boldsymbol{e}_1' = \boldsymbol{\theta}' - \boldsymbol{q}_{fast}^{\prime d} \triangleq \boldsymbol{\nu} - \boldsymbol{q}_{fast}^{\prime d}. \tag{19}$$

$$\text{Choose } \boldsymbol{V}_1(\boldsymbol{e}_1) = \frac{1}{2}\boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1 \tag{20}$$

$$\text{Then, } \boldsymbol{V}_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1' = \boldsymbol{e}_1^\top \boldsymbol{K}_p (\boldsymbol{\nu} - \boldsymbol{q}_{fast}^{\prime d}). \tag{21}$$

For $\boldsymbol{\nu} = \boldsymbol{q}_{fast}^{\prime d} - \boldsymbol{e}_1, \ \boldsymbol{V}_1' = -\boldsymbol{e}_1 \boldsymbol{K}_p \boldsymbol{e}_1 \leq 2\boldsymbol{V}_1.$

Stability Analysis of the Fast Velocity Subdynamics

Theorem 4 ([6])

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

$$\boldsymbol{u}_{fvel} = \frac{1}{\epsilon}\mathcal{H}_{fast}[\boldsymbol{q}_{fast}^{\prime\prime d} + \boldsymbol{e}_1 - 2\boldsymbol{e}_2 - \boldsymbol{K}_q^\top(\boldsymbol{K}_q\boldsymbol{K}_q^\top)^{-1}\boldsymbol{K}_p\boldsymbol{e}_1] + \frac{1}{\epsilon}\mathcal{H}_{slow}^{fast}\boldsymbol{z}_{slow}^{\prime} - \boldsymbol{s}_{fast} \tag{22}$$

exponentially stabilizes the fast subdynamics (17).

Stability Analysis of Fast Velocity Subdynamics

Proof Sketch 2 (Sketch Proof of Theorem 4)

Recall from the position dynamics controller:

$$\begin{aligned} \boldsymbol{e}_1' &= \boldsymbol{\theta}' - \boldsymbol{q}_{fast}'^{d} \triangleq \boldsymbol{z}_{fast} - \boldsymbol{q}_{fast}'^{d} + (\boldsymbol{\nu} - \boldsymbol{\nu}) && \text{(23a)} \\ &= (\boldsymbol{\phi} - \boldsymbol{\nu}) + (\boldsymbol{\nu} - \boldsymbol{q}_{fast}'^{d}) \triangleq \boldsymbol{e}_2 - \boldsymbol{e}_1. && \text{(23b)} \end{aligned}$$

It follows that

$$\begin{aligned} \boldsymbol{e}_2' &= \boldsymbol{\phi}' - \boldsymbol{\nu}' = \boldsymbol{z}_{fast}' + \boldsymbol{e}_1' - \boldsymbol{q}_{fast}''^{d} && \text{(24)} \\ &= \boldsymbol{\mathcal{H}}_{fast}^{-1} \left[\epsilon \boldsymbol{u}_{fast} + \epsilon \boldsymbol{s}_{fast} - \boldsymbol{\mathcal{H}}_{slow}^{fast} \boldsymbol{z}_{slow}' \right] + (\boldsymbol{e}_2 - \boldsymbol{e}_1) - \boldsymbol{q}_{fast}''^{d}. \end{aligned}$$

Stability Analysis of the Fast Velocity Subdynamics

Proof Sketch 3 (Sketch Proof of Theorem 4)

For diagonal matrices $\boldsymbol{K}_p, \boldsymbol{K}_q$ with positive damping, let us choose the Lyapunov candidate function

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

If $\tilde{\boldsymbol{q}}_{fast} = \boldsymbol{q}_{fast} - \boldsymbol{q}_{fast}^{d}$ and $\tilde{\boldsymbol{q}}'_{fast} = \boldsymbol{q}'_{fast} - \boldsymbol{q}'^{d}_{fast}$, then the controller

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

exponentially stabilizes the system;

Stability Analysis of the Fast Velocity Subdynamics

Proof Sketch 4 (Sketch Proof of Theorem 4)

since it can be verified that

$$
\begin{aligned}
\boldsymbol{V}_2' &= \boldsymbol{e}_1^\top \boldsymbol{K}_p(\boldsymbol{e}_2 - \boldsymbol{e}_1) \\
&\quad - \boldsymbol{e}_2^\top \boldsymbol{K}_q \left(\boldsymbol{e}_2 - \boldsymbol{K}_q^\top (\boldsymbol{K}_q \boldsymbol{K}_q^\top)^{-1} \boldsymbol{K}_p \boldsymbol{e}_1\right) && \text{(25a)} \\
&= -\boldsymbol{e}_1^\top \boldsymbol{K}_p \boldsymbol{e}_1 - \boldsymbol{e}_2^\top \boldsymbol{K}_q \boldsymbol{e}_2 && \text{(25b)} \\
&\triangleq -2\boldsymbol{V}_2 \le 0. && \text{(25c)}
\end{aligned}
$$

Stability analysis of the slow subdynamics

Set $\boldsymbol{e}_3 = \boldsymbol{z}_{\text{slow}} - \boldsymbol{\nu}$ so that $\dot{\boldsymbol{e}}_3 = \dot{\boldsymbol{z}}_{\text{slow}} - \dot{\boldsymbol{\nu}}$. Then,

$$\dot{\boldsymbol{e}}_3 = \dot{\boldsymbol{z}}_{\text{slow}} - \ddot{\boldsymbol{q}}^d_{\text{fast}} + (\boldsymbol{e}_2 - \boldsymbol{e}_1), \tag{26a}$$

$$= \mathcal{H}^{-1}_{\text{slow}}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) - \ddot{\boldsymbol{q}}^d_{\text{fast}} + (\boldsymbol{e}_2 - \boldsymbol{e}_1). \tag{26b}$$

Theorem 5

The control law

$$\boldsymbol{u}_{slow} = \mathcal{H}_{slow}(\boldsymbol{e}_1 - \boldsymbol{e}_2 - \boldsymbol{e}_3 + \ddot{\boldsymbol{q}}^d_{fast}) - \boldsymbol{s}_{slow} \tag{27}$$

exponentially stabilizes the slow subdynamics.

Stability analysis of the slow subdynamics

Proof.

Consider the Lyapunov function candidate

$$\boldsymbol{V}_3(\boldsymbol{e}_3) = \frac{1}{2}\boldsymbol{e}_3^\top \boldsymbol{K}_r \boldsymbol{e}_3 \text{ where } \boldsymbol{K}_r = \boldsymbol{K}_r^\top > 0. \tag{28}$$

It follows that

$$\dot{\boldsymbol{V}}_3(\boldsymbol{e}_3) = \boldsymbol{e}_3^\top \boldsymbol{K}_r \dot{\boldsymbol{e}}_3 \tag{29a}$$

$$= \boldsymbol{e}_3^\top \boldsymbol{K}_r \left[\boldsymbol{\mathcal{H}}_{\text{slow}}^{-1}(\boldsymbol{s}_{\text{slow}} + \boldsymbol{u}_{\text{slow}}) - \ddot{\boldsymbol{q}}_{\text{fast}}^d + \boldsymbol{e}_2 - \boldsymbol{e}_1\right]. \tag{29b}$$

Substituting $\boldsymbol{u}_{\text{slow}}$ in (27), it can be verified that

$$\dot{\boldsymbol{V}}_3(\boldsymbol{e}_3) = \boldsymbol{e}_3^\top \boldsymbol{K}_r \boldsymbol{e}_3 \triangleq -2\boldsymbol{V}_3(\boldsymbol{e}_3) \le 0. \tag{30}$$

Hence, the controller (27) stabilizes the slow subsystem. □

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. ,

$$\boldsymbol{\Sigma}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) = (1-\varepsilon)\boldsymbol{V}_2(\boldsymbol{z}_{\text{fast}}) + \varepsilon \boldsymbol{V}_3(\boldsymbol{z}_{\text{slow}}),\; 0 < \varepsilon < 1. \tag{31}$$

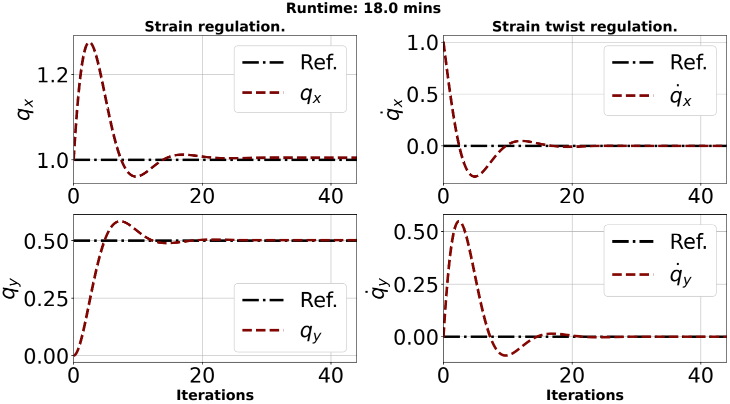
It follows that,

$$\begin{aligned}\dot{\boldsymbol{\Sigma}}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) &= (1-\varepsilon)[\boldsymbol{e}_1^\top \boldsymbol{K}_p \dot{\boldsymbol{e}}_1 + \boldsymbol{e}_2^\top \boldsymbol{K}_q \dot{\boldsymbol{e}}_2] + \varepsilon \boldsymbol{e}_3^\top \boldsymbol{K}_r \dot{\boldsymbol{e}}_3, \\ &= -2(\boldsymbol{V}_2 + \boldsymbol{V}_3) + 2\varepsilon \boldsymbol{V}_2 \le 0\end{aligned} \tag{32}$$

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.

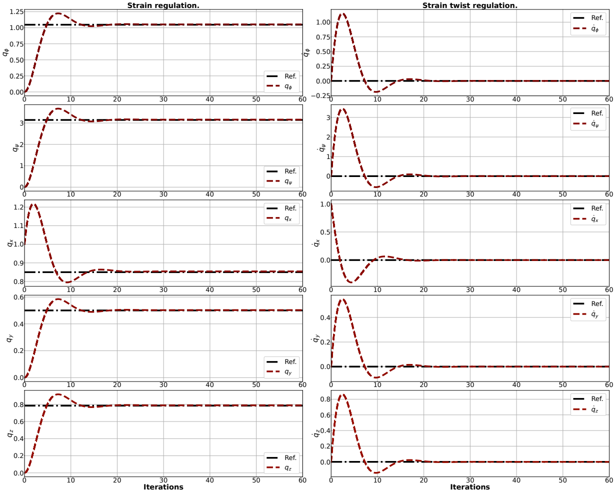
$$\boldsymbol{u}(\boldsymbol{z}_{\text{fast}}, \boldsymbol{z}_{\text{slow}}) = (1-\varepsilon)\boldsymbol{u}_{\text{fast}} + \varepsilon \boldsymbol{u}_{\text{slow}}. \tag{33}$$

Asynchronous, time-separated control



Ten discretized PCS sections: 6 fast, 4 slow subsections. $\mathcal{F}_p^y = 10\,N$, with $\boldsymbol{K}_p = 10$, $\boldsymbol{K}_d = 2.0$ for $\boldsymbol{\eta}^d = [0, 0, 0, 1, 0.5, 0]^\top$ and $\boldsymbol{\xi}^d = \mathbf{0}_{6\times 1}$.

Five-axes control



Time Response Comparison with Non-hierarchical Controller

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: Time to Reach Steady State.

Contributions

- Layered singularly perturbed techniques for decomposing system dynamics to multiple timescales.
- Stabilizing nonlinear backstepping controllers were introduced to the respective subdynamics for fast strain regulation.

Discussions

- Leverage the *multiphysics of (often) heterogeneous soft material components*;
- Neat manipulation strategies for motion is a *multiscale problem* that requires imbuing geometric mathematical reasoning into the control strategies for desired movements.
- Challenge: Merging the long-term planning horizon of spatial perception tasks with the *fast time-constant* (typically milliseconds or microseconds) requirements of the precise control of soft, compliant pneumatic/mechanical systems across multiple time-scales;

Discussions

- Process spatial information (Lagrangian) often within a long-time horizon context (Eulerian) for the real-time control or planning across multiple time-scales.

References I

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

[2] Cosimo Della Santina, Christian Duriez, and Daniela Rus. Model-based control of soft robots: A survey of the state of the art and open challenges. *IEEE Control Systems Magazine*, 43(3):30–65, 2023.

[3] 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.

[4] 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.

[5] 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.

[6] Lekan Molu. Fast Whole-Body Strain Regulation in Continuum Robots. *(submitted to) American Control Conference*, 2024.

[7] 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.

[8] 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.

[9] 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.

[10] Federico Renda, Frédéric Boyer, Jorge Dias, and Lakmal Seneviratne. Discrete cosserat approach for multisection soft manipulator dynamics. *IEEE Transactions on Robotics*, 34(6):1518–1533, 2018.

References II

[11] 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.

[12] 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.

[13] José Guadalupe Romero, Romeo Ortega, and Ioannis Sarras. A globally exponentially stable tracking controller for mechanical systems using position feedback. *IEEE Transactions on Automatic Control*, 60(3):818–823, 2014.

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

[15] 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.

References I

Simon Du, Akshay Krishnamurthy, Nan Jiang, Alekh Agarwal, Miroslav Dudik, and John Langford.
Provably efficient rl with rich observations via latent state decoding.
In *International Conference on Machine Learning*, pages 1665–1674. PMLR, 2019.

Yonathan Efroni, Dylan J Foster, Dipendra Misra, Akshay Krishnamurthy, and John Langford.
Sample-efficient reinforcement learning in the presence of exogenous information.
(Accepted for publication at) Conference on Learning Theory, 2022.

Yonathan Efroni, Dipendra Misra, Akshay Krishnamurthy, Alekh Agarwal, and John Langford.
Provably filtering exogenous distractors using multistep inverse dynamics.
In *International Conference on Learning Representations*, 2022.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell.
Curiosity-driven exploration by self-supervised prediction.
In *International conference on machine learning*, pages 2778–2787. PMLR, 2017.

Tongzhou Wang, Simon S Du, Antonio Torralba, Phillip Isola, Amy Zhang, and Yuandong Tian.
Denoised mdps: Learning world models better than the world itself.
arXiv preprint arXiv:2206.15477, 2022.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine.
Learning invariant representations for reinforcement learning without reconstruction.
arXiv preprint arXiv:2006.10742, 2020.