

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

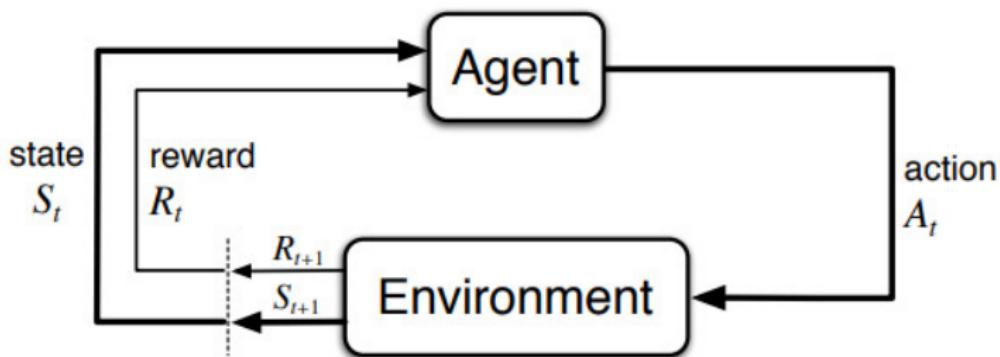
Lekan Molu

New York City, NY 10012

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

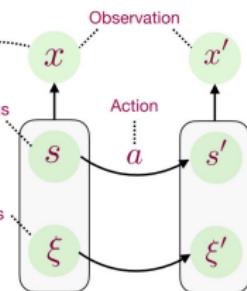
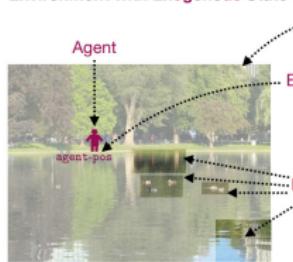
August 13, 2025

# Standard Reinforcement Learning



# Compact States without Exogenous Distractors in RL

Environment with Exogenous State



Generalized Inverse Dynamics

Train a model to predict the index of roll-in path

$$f_{\theta}(\text{idx}(\nu \circ a) | x')$$



$$\nu \sim \text{Uniform}(\Psi_{h-1}) \quad a \sim \text{Uniform}(\mathcal{A})$$

Policy cover for the last time step

Action space

Learning  $s$  with  $[S]$  whilst ignoring temporally correlated  $\xi$ ?

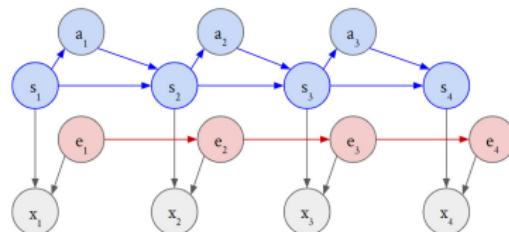
# Literature comparison

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.	✓	✓	✗	✓	✓	✗	✓

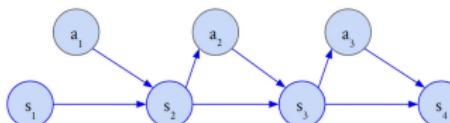
Emphasis on robustness to exogenous information. Comparison with baselines including PPE [3], OSSR [2], DBC [6] , Denoised MDP [5] and One-Step Inverse Models [4].

# Rewards-agnostic Exogenous State Invariance in RL

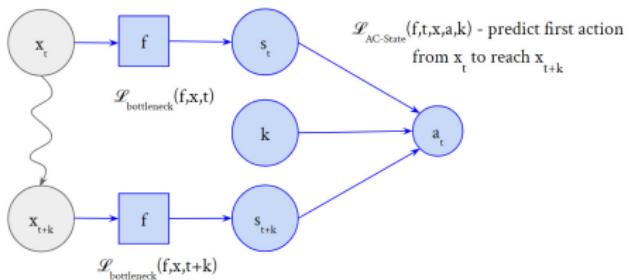
AC-State Discovers the smallest control-endogenous state  $s$  assuming factorized dynamics



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 for  $s = f(x)$  by optimizing a multi-step inverse model with a bottleneck



# Latent States Discovery – Multi-step Inverse Dynamics

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

- Latent state,  $e$ , is chosen from the lowest capacity encoder,  $f$ , in the set of optimal solutions to (1)

$$G = \arg \min_{f \in \mathcal{F}} \mathbb{E}_{t,k} [\mathcal{L}_{\text{ACS}}(f, x, a, t; k)]$$
$$\hat{f} \in \arg \min_{f \in G} \text{Range}(f). \quad (2)$$

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

# AC State Algorithm

---

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

---

- 1: Initialize observation trajectory  $x$  and action trajectory  $a$ . Initialize encoder  $f_\theta$ . 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  $N$ .
- 2:  $x_1 \sim U(\mu(x))$
- 3: **for**  $t = 1, 2, \dots, T$  **do**
- 4:    $a_t \sim U(\mathcal{A})$
- 5:    $x_{t+1} \sim \mathbb{P}(x'|x_t, a_t)$
- 6: **for**  $n = 1, 2, \dots, N$  **do**
- 7:    $t \sim U(1, T)$  and  $k \sim U(1, K)$
- 8:    $\mathcal{L} = \mathcal{L}_{\text{AC-State}}(f_\theta, t, x, a, k) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_t) + \mathcal{L}_{\text{Bottleneck}}(f_\theta, x_{t+k})$
- 9:   Update  $\theta$  to minimize  $\mathcal{L}$  by gradient descent.

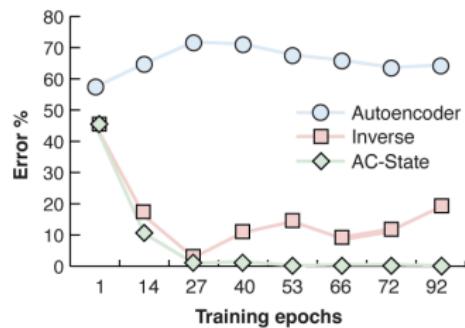
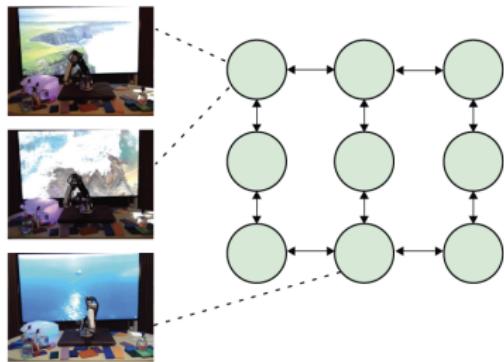
---

# AC State in Action

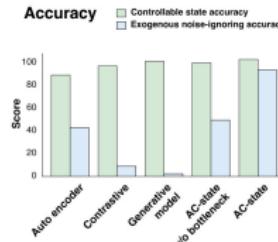
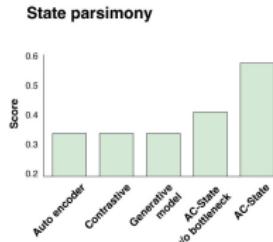
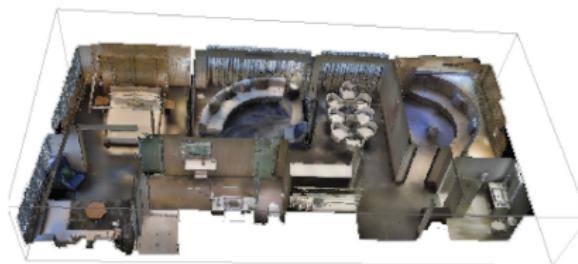


Exogenous distractors riddance.

# AC State in Action



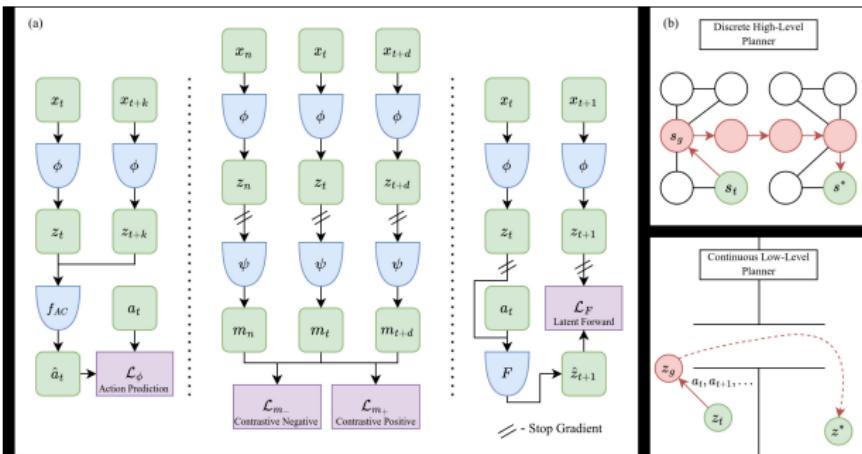
# AC State in Matterport



# PCLAST: Agent Plannable Continuous Latent States

## PcLast: Discovering Plannable Continuous Latent States

Anurag Kouli<sup>\*1</sup> Shivakanth Sujit<sup>\*234</sup> Shaoru Chen<sup>1</sup> Ben Evans<sup>5</sup> Lili Wu<sup>1</sup> Byron Xu<sup>1</sup> Rajan Chari<sup>1</sup>  
Riahat Islam<sup>3,6</sup> Raihan Seraj<sup>3,6</sup> Yonathan Efroni<sup>7</sup> Lekan Molu<sup>1</sup> Miro Dudik<sup>1</sup> John Langford<sup>1</sup> Alex Lamb<sup>1</sup>



# PCLAST Algorithm

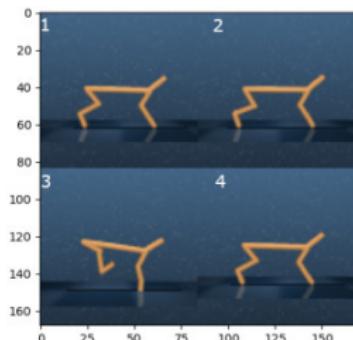
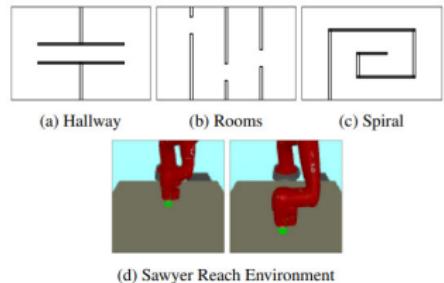
---

**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**  $i = n, n - 1, \dots, 2$  **do**
    - 3:    $\hat{s}^*$  = 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.}
- 



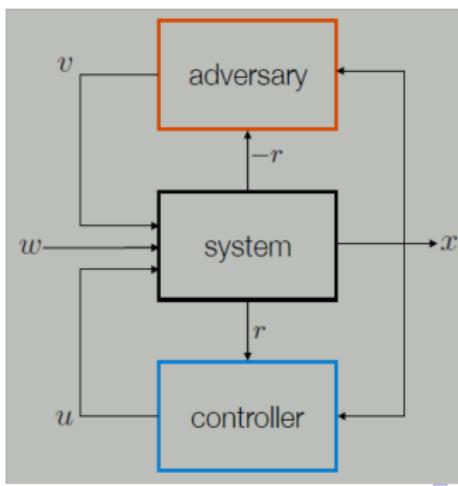
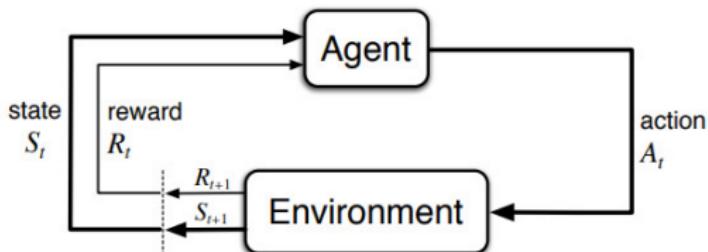
# PCLAST Results

METHOD	Reward Type	HALLWAY	ROOMS	SPIRAL	SAWYER-REACH
PPO	DENSE	$6.7 \pm 0.6$	$7.5 \pm 7.1$	$11.2 \pm 7.7$	<b><math>86.00 \pm 5.367</math></b>
PPO + ACRO	DENSE	$10.0 \pm 4.1$	$23.3 \pm 9.4$	$23.3 \pm 11.8$	$84.00 \pm 6.066$
PPO + PCLAST	DENSE	<b><math>66.7 \pm 18.9</math></b>	<b><math>43.3 \pm 19.3</math></b>	<b><math>61.7 \pm 6.2</math></b>	$78.00 \pm 3.347$
PPO	SPARSE	$1.7 \pm 2.4$	$0.0 \pm 0.0$	$0.0 \pm 0.0$	$68.00 \pm 8.198$
PPO + ACRO	SPARSE	$21.7 \pm 8.5$	$5.0 \pm 4.1$	$11.7 \pm 8.5$	<b><math>92.00 \pm 4.382</math></b>
PPO + PCLAST	SPARSE	<b><math>50.0 \pm 18.7</math></b>	<b><math>6.7 \pm 6.2</math></b>	<b><math>46.7 \pm 26.2</math></b>	$82.00 \pm 5.933$
CQL	SPARSE	$3.3 \pm 4.7$	$0.0 \pm 0.0$	$0.0 \pm 0.0$	$32.00 \pm 5.93$
CQL + ACRO	SPARSE	$15.0 \pm 7.1$	<b><math>33.3 \pm 12.5</math></b>	<b><math>21.7 \pm 10.3</math></b>	$68.00 \pm 5.22$
CQL + PCLAST	SPARSE	<b><math>40.0 \pm 0.5</math></b>	$23.3 \pm 12.5$	$20.0 \pm 8.2$	<b><math>74.00 \pm 4.56</math></b>
RIG	NONE	$0.0 \pm 0.0$	$0.0 \pm 0.0$	$3.0 \pm 0.2$	<b><math>100.0 \pm 0.0</math></b>
RIG + ACRO	NONE	<b><math>15.0 \pm 3.5</math></b>	$4.0 \pm 1.$	<b><math>12.0 \pm 0.2</math></b>	$100.0 \pm 0.0$
RIG + PCLAST	NONE	$10.0 \pm 0.5$	$4.0 \pm 1.8$	$10.0 \pm 0.1$	$90.0 \pm 5$
LOW-LEVEL PLANNER + PCLAST	NONE	$86.7 \pm 3.4$	$69.3 \pm 3.4$	$50.0 \pm 4.3$	$\pm$
$n$ -LEVEL PLANNER + PCLAST	NONE	<b><math>97.78 \pm 4.91</math></b>	<b><math>89.52 \pm 10.21</math></b>	<b><math>89.11 \pm 10.38</math></b>	$95.0 \pm 1.54$

# 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

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

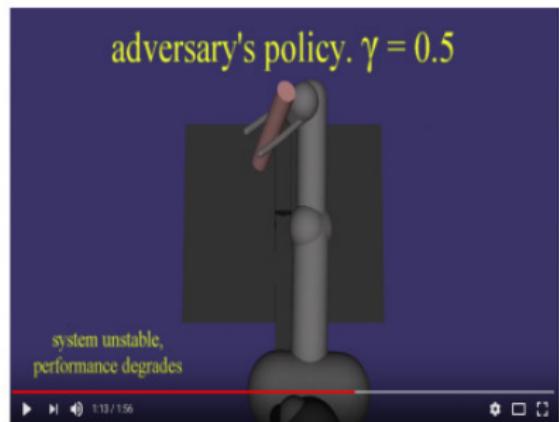
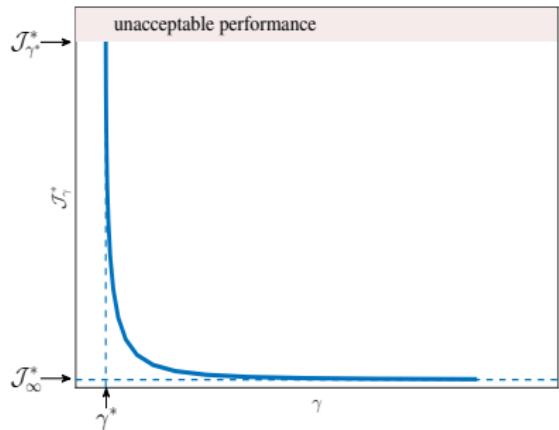
- To mitigate lack of robustness, we optimize the *cost-to-go*

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

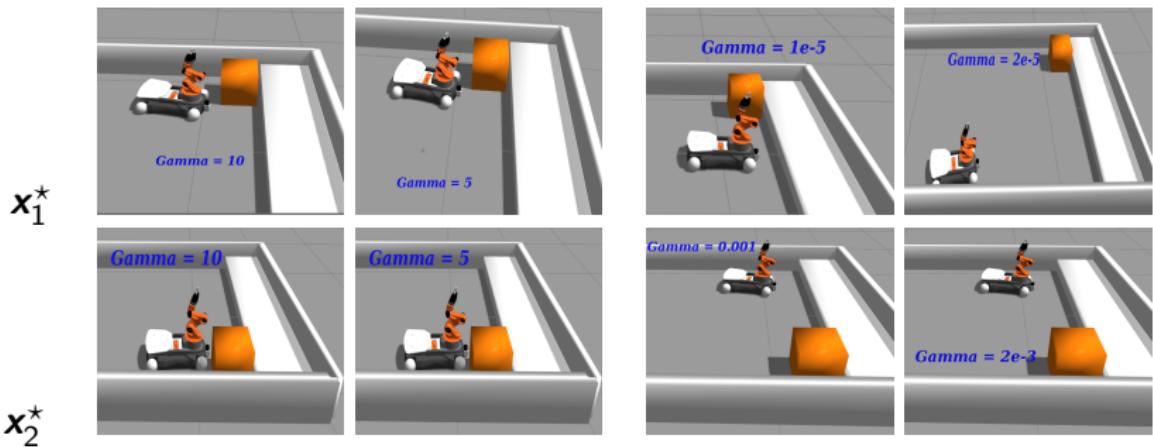
- and seek a saddle point equilibrium policy that satisfies

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

# Results: Brittleness Quantification



## Results: Iterative Dynamic Game



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

# Mixed $H_2/H_\infty$ 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*.
  - Kleinman'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;
  - Kleinman and input-to-state-stability;
  - Robust policy optimization as a small-input stable state optimization algorithm

# Credits

Continuous-  
Time  
Stochastic  
Policy  
Optimization

Lekan Molu

Outline and  
Overview

Risk-sensitive  
control

Contributions

Setup

Assumptions

Optimal Gain

Model-based  
PO

Outer loop

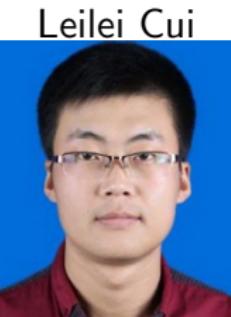
Stabilization and  
Convergence

Sampling-  
based PO

Discrete-time  
system

Sampling-based  
nonlinear system

Robustness Analysis



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 Analysis

- (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

- 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  $\mathcal{K} = \{K_1, K_2, \dots, K_n\}$ .

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

# Policy Optimization – Open questions

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

- 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.

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

## (Non-exhaustive) Lit. Landscape on PO Theory

Literature landscape	Cont. time (Kalman '61, Luenberger '63)	Stochastic. LQR (Kalman '60)	Cont. Phase	LEQG or Mixed $H_2/H_\infty$	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

# Mainstay

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

- Continuous-time infinite-dimensional linear systems.
  - Disturbances enter additively as random stochastic Wiener processes.
  - Many natural systems admit uncertain additive Brownian noise as diffusion processes.
    - Theoretical analysis machinery: Ito's stochastic calculus.
- Goal: keep controlled process,  $z$ , small i.e.

$$\|z\|_2 = \left( \int |z(t)|^2 dt \right)^{1/2},$$

- Under a minimizing  $u(x(t)) \in \mathcal{U}$  in spite of unforeseen  $w(t) \in \mathcal{W} \subseteq \mathbb{R}^q$ .

# Contributions

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

- A two-loop iterative alternating best-response procedure for computing the optimal mixed-design policy;
- Rigorous convergence analyses follow for the model-based loop updates;
- In the absence of exact system models, we provide an input-to-state-stable hybrid robust stabilization scheme.

# Nonconvexity and Coercivity in PG

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

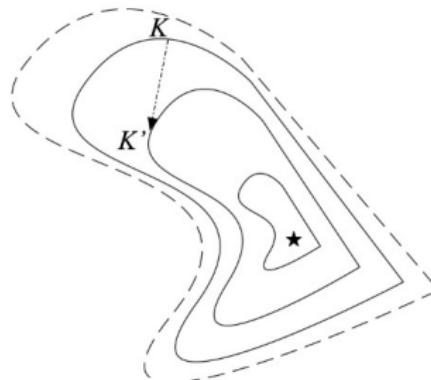
Sampling-based PO

Discrete-time system

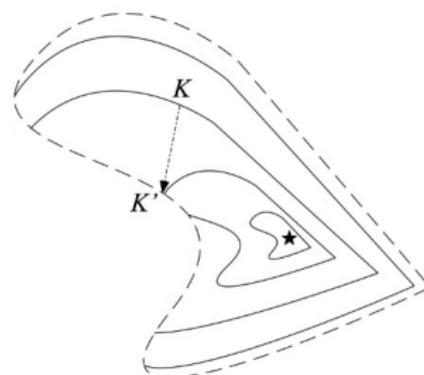
Sampling-based nonlinear system

Robustness Analyses

- Coercivity: iterates remain feasible and strictly separated from the infeasible set as the cost decreases.



(a) Landscape of LQR



(b) Landscape of Mixed  $\mathcal{H}_2/\mathcal{H}_{\infty}$  Control

Figure: Coercivity property of PG on LQR and in mixed-design settings.  
Credit: (Zhang et al., 2019).

# Model-based Policy 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

---

## 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  $R \succ 0$ .

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

# Convergence Analyses: Outer Loops

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

## Lemma 1

*Under our assumptions and for the ARE (14), if  $K_0 \in \mathcal{K}$ , then for any  $p \in \mathbb{N}_+$ , we must have the following conditions for the optimal  $K^*$  and  $P^*$ ,*

- (1)  $K_p \in \mathcal{K}$ ;
- (2)  $P_K^0 \succeq P_K^1 \succeq \cdots P_K^p \succeq \cdots \succeq P^*$ ;
- (3)  $\lim_{p \rightarrow \infty} \|K_p - K^*\|_F = 0, \lim_{p \rightarrow \infty} \|P_K^p - P^*\|_F = 0$ .

# Optimality of the 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

## Lemma 3 (Optimality of the iteration)

Consider any  $K \in \mathcal{K}$ , let  $K' = R^{-1}B^\top P_K$  (where  $P_K$  is the solution to (18), and  $\Psi_K = (K - K')^\top R(K - K')$ . If  $\Psi_K = 0$ , then  $K = K^*$ .

## Proof.

Since  $R \succ 0$ ,  $\Psi_K = 0$  implies  $K = K'$ . Therefore at  $\Psi_K = 0$ , we must have  $K = K'$  which implies that  $P_K = P'_K$ . If  $K = K'$  and  $P_K = P'_K$ , it suffices to conclude that  $K' = K \triangleq K^*$  where  $K^* = R^{-1}B^\top P^*$ . Hence,  $\Psi_K = 0$  is tantamount to  $P_K = P^*$  and  $K = K^*$ . □

# Convergence Analysis: Inner Loop I

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

## Lemma 5

Suppose that  $L_0(K_0)$  is stabilizing, then for any  $q \in \mathbb{N}_+$  (with  $P_{K,L}^{p,\bar{q}}$  as the solution to (15)), i.e.

$$A_{K,L}^{(p,q)\top} P_{K,L}^{p,q} + P_{K,L}^{p,q} A_{K,L}^{p,q} + Q_K^p - \gamma^2 L_q^\top(K_p) L_q(K_p) = 0 \quad (22a)$$

$$K_{p+1} = R^{-1} B^\top P_K^{p,q}, \quad L_{q+1}(K_p) = \gamma^{-2} D^\top P_{K,L}^{p,q}. \quad (22b)$$

Then, the following statements hold

- 1  $A_{K,L}^{p,q}$  is Hurwitz;
- 2  $P_{K,L}^{p,\bar{q}} \succeq \dots \succeq P_K^{(p,q+1)} \succeq P_K^{p,q} \succeq \dots \succeq P_{K,L}^{p,0}$ ; and
- 3  $\lim_{q \rightarrow \infty} \|P_{K,L}^{p,q} - P_{K,L}^{p,\bar{q}}\|_F = 0$ .

# Convergence Rate – Inner Loop

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

## Lemma 6 (Monotonic Convergence of the Inner-Loop)

For any  $K \in \mathcal{K}$ , let  $L(K)$  be the control gain for the player  $w$  such that  $A_K + DL(K)$  is Hurwitz. Let  $P_K^L$  be the solution of

$$(A_K + DL(K))^T P_K^L + P_K^L (A_K + DL(K)) + Q_K - \gamma^2 L(K)^T L(K) = 0. \quad (23)$$

Let  $L'(K) = \gamma^{-2} D^T P_K^L$  and

$\Psi_K^L = \gamma^{-2} (L'(K) - L(K))^T (L'(K) - L(K))$ . Then, for a  $c(K) = \text{Tr} \left( \int_0^\infty e^{(A_K + DL(K^*))t} e^{(A_K + DL(K^*))^T t} dt \right)$ , the following inequality holds  $\text{Tr}(P_K - P_K^L) \leq \|\Psi_K^L\| c(K)$ .

# 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 Analysis

## Theorem 3

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

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

## 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  $\|P_K - P_{K,L}^{p,q}\|_F \leq \text{Tr}(P_K - P_{K,L}^{p,q}) \leq \beta(K) \text{Tr}(P_K)$ , i.e.  $P_{K,L}^{p,q}$  exponentially converges to  $P_K$  in the Frobenius norm.

# Sampling-based PO Scheme

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

- $A, B, C, D, E$  are often unavailable so that the policy evaluation step will result in biased estimates.
- There is the possibility for a divergence from the stability-robustness feasibility set  $\tilde{\mathcal{K}}$ :
  - When errors are present from I/O or state data;
  - Residuals from early termination of numerically solving Riccati equations;
  - Using an approximate cost function owing to inexact values of  $Q$  and  $R$ ;
  - Since the inner loop is computed in a finite number of steps;
  - In a data sampling scheme, we must guarantee the stability and robustness of the closed-loop system.

# Sampling-based PO: Statement of the Problem

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

## Problem 4 (Sampling-based Policy Optimization)

If  $A, B, C, D, E, P$  are all replaced by approximate matrices  $\hat{A}, \hat{B}, \hat{C}, \hat{D}, \hat{E}, \hat{P}$ , under what conditions will the sequences  $\{\hat{P}_{K,L}^{p,q}\}_{(p,q)=1}^{\infty}$ ,  $\{\hat{K}_p\}_{p=0}^{\infty}$ ,  $\{\hat{L}_q\}_{q=0}^{\infty}$  converge to a small neighborhood of the optimal values  $\{P_{K,L}^*\}_{(p,q)=0}^{\infty}$ ,  $\{K_p^*\}_{p=0}^{\infty}$ , and  $\{L_q^*\}_{q=0}^{\infty}$ ?

# Robustness Analyses

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

- 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

$$\|P_{\hat{K}}^p - P^*\| \leq \beta_1(\|P_{\hat{K}}^0 - P^*\|, p) + \gamma_1(\|\tilde{K}\|). \quad (37)$$

# ISS Outer Loop Robustness Proof

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

- Prelim result (Lemma 12, C&M, '23): For any  $h > 0$  and  $K \in \mathcal{K}_h$ , let  $K' = R^{-1}B^\top P_K$ , where  $P_K$  is the solution of (18), and  $\hat{K}' = K' + \tilde{K}$ . Then, there exists  $f(h) > 0$ , such that  $\hat{K}' \in \mathcal{K}_h$  as long as  $\|\tilde{K}\| < f(h)$ .
- Therefore,  $\hat{K}_K^p \in \mathcal{K}_h$  for any  $p \in \mathbb{N}_+$ .
- Let

$$f_1(\hat{K}') = \frac{\log(5/4)b(h)}{2n\|A_{\hat{K}'}^*\|}, f_2(\hat{K}') = \text{Tr} \left( \int_0^\infty e^{A_{\hat{K}'}^* \top t} e^{A_{\hat{K}'}^* t} dt \right).$$

# ISS Outer Loop Robustness Proof

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



$$\underline{f}_1(h) = \inf_{\hat{K}' \in \mathcal{K}_h} f_1(\hat{K}') > 0, \bar{f}_2(h) = \sup_{\hat{K}' \in \mathcal{K}_h} f_2(\hat{K}') < \infty. \quad (38)$$

- This implies

$$Tr(P_{\hat{K}}^p - P^*) \leq [1 - \underline{f}_1(h)] Tr(P_{\hat{K}}^{p-1} - P^*) + \bar{f}_2(h) \|R\| \|\tilde{K}_K^p\|^2. \quad (39)$$

- Repeating (39) for  $p, p-1, \dots, 1$ ,

$$Tr[P_{\hat{K}}^p - P^*] \leq (1 - \underline{f}_1)^p Tr(P_{\hat{K}}^1 - P^*) + \frac{\bar{f}_2 \|R\| \|\tilde{K}\|_\infty^2}{\underline{f}_1(h)}. \quad (40)$$

# Outer 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

It follows from (40) and (Mori, 1988, Theorem 2) that

$$\|P_{\hat{K}}^p - P^*\|_F \leq (1 - \underline{f}_1)^p \sqrt{n} \|P_{\hat{K}}^1 - P^*\|_F + \frac{\bar{f}_2 \|R\| \|\tilde{K}\|_\infty^2}{\underline{f}_1}. \quad (41)$$

As  $p \rightarrow \infty$ ,  $P_{\hat{K}}^p \rightarrow P^*$ . Whence, a radius of  $P^*$ 's neighbor is proportional to  $\|\tilde{K}\|_\infty^2$ .

# 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

The perturbed inner-loop iteration (26) has inexact matrix  $\hat{A}_{K,L}^{p,q}$ , and sequences  $\{\hat{L}_{q+1}(K_p)\}_{q=0}^{\infty}$ , and  $\{\hat{P}_{K,L}^{p,q}\}_{q=0}^{\infty}$ .

## Lemma 8 (Stability of the Inner-Loop's System Matrix)

Given  $K \in \check{\mathcal{K}}$ , there exists a  $g \in \mathbb{R}_+$ , such that if

$\|\tilde{L}_{q+1}(K_p)\|_F \leq g$ ,  $\hat{A}_{K,L}^{p,q}$  is Hurwitz for all  $q \in \mathbb{N}_+$ .

# 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

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

- From Theorem 7, as  $q \rightarrow \infty$ ,  $\hat{P}_{K,L}^{p,q}$  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  $P_{K,L}^{p,q}$ .

# Numerical Results – Car Cruise Control System

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

- Well-suited to our robust control formulation because
  - the disturbances and state variables are separable and can be lumped into the form of the stochastic differential equations;
  - it is a multiple-input (throttle, gear, vehicle speed) single-output (vehicle acceleration) system that introduces modeling challenges;
  - the entire operating range of the system is nonlinear though there is a reasonable linear bandwidth that characterize the input/output (I/O) system as we will see shortly.

# Road (Disturbance) Profile

Continuous-  
Time  
Stochastic  
Policy  
Optimization

Lekan Molu

Outline and  
Overview

Risk-sensitive  
control

Contributions

Setup

Assumptions

Optimal Gain

Model-based  
PO

Outer loop

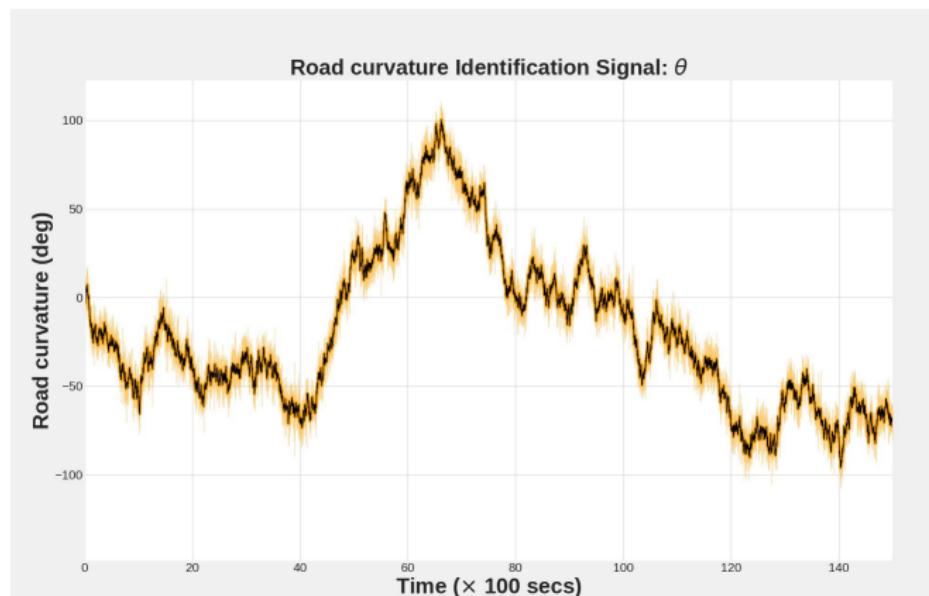
Stabilization and  
Convergence

Sampling-  
based PO

Discrete-time  
system

Sampling-based  
nonlinear system

Robustness Analyses



# Cost Matrix and Gains Convergence

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview

Risk-sensitive control

Contributions

Setup

Assumptions

Optimal Gain

Model-based PO

Outer loop

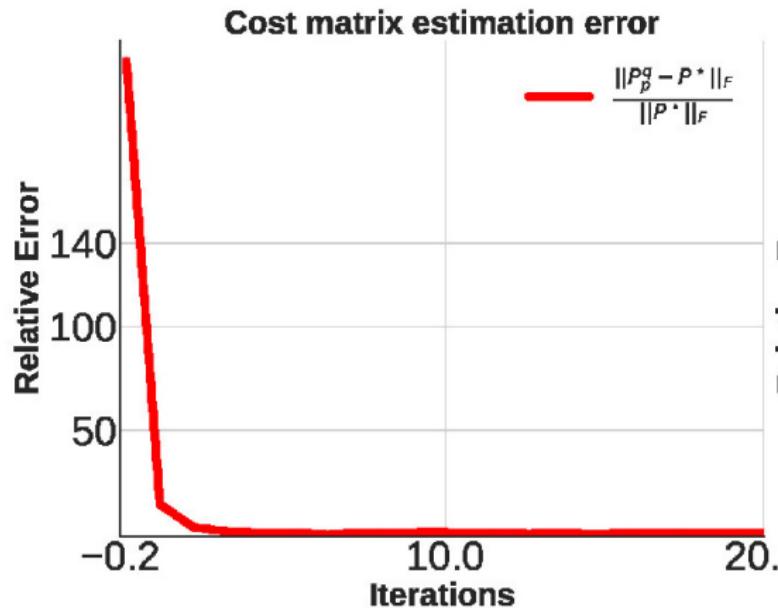
Stabilization and Convergence

Sampling-based PO

Discrete-time system

Sampling-based nonlinear system

Robustness Analysis



# Pendulums Experiment – Comparison to NPG

Continuous-  
Time  
Stochastic  
Policy  
Optimization

Lekan Molu

Outline and  
Overview

Risk-sensitive  
control

Contributions

Setup

Assumptions

Optimal Gain

Model-based  
PO

Outer loop

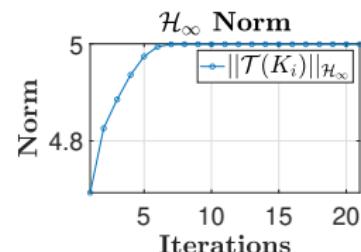
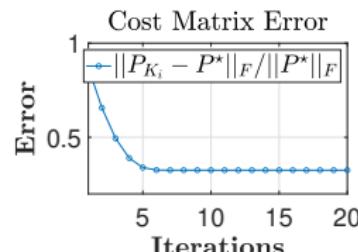
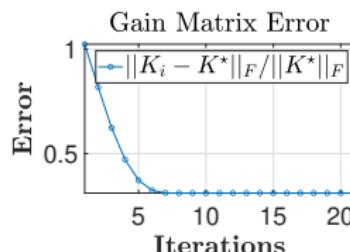
Stabilization and  
Convergence

Sampling-  
based PO

Discrete-time  
system

Sampling-based  
nonlinear system

Robustness Analyses



Model-free design:  $\|\tilde{K}\|_\infty = 0.15$ .

# Pendulums Experiment – Comparison to NPG

Continuous-Time Stochastic Policy Optimization

Lekan Molu

Outline and Overview

Risk-sensitive control

Contributions

Setup

Assumptions

Optimal Gain

Model-based PO

Outer loop

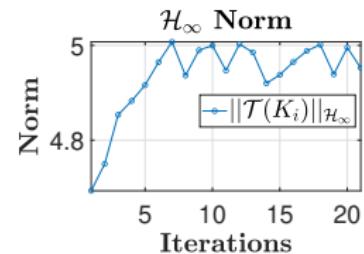
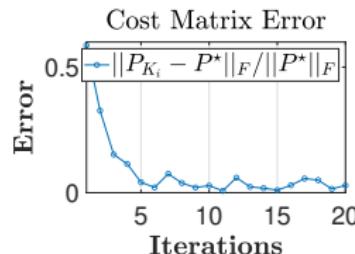
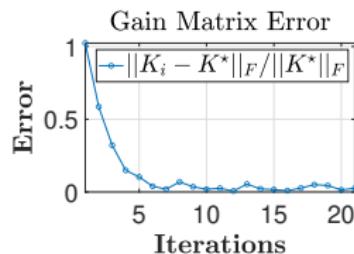
Stabilization and Convergence

Sampling-based PO

Discrete-time system

Sampling-based nonlinear system

Robustness Analysis



Model-based design:  $\|\tilde{K}\|_\infty = 0.15$ .

# Double Pendulum and Acrobot Experiment – Comparison to NPG

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

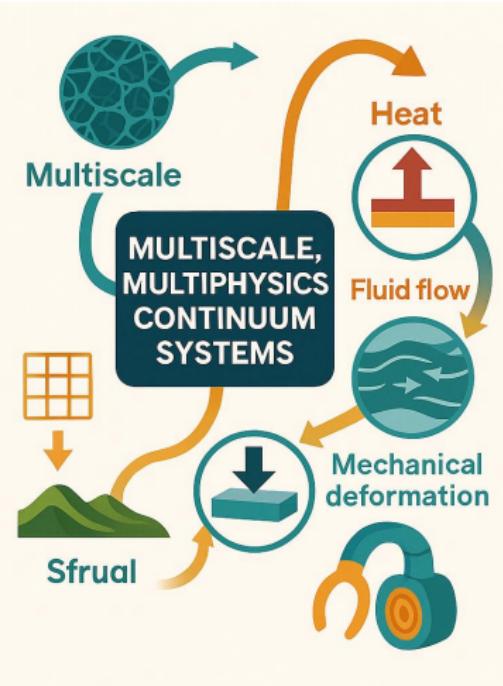
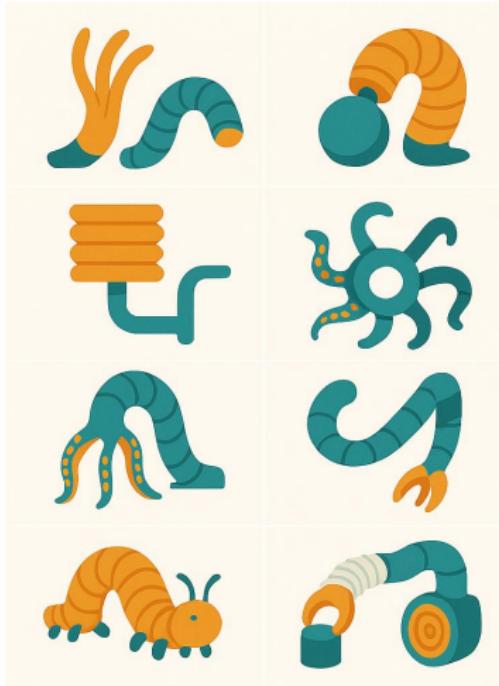
Table: Computational Time: Model-based PO vs. Model-free PO vs. NPG.

Policy Optimization			Computational time (secs)		
Double Inverted Pendulum			Triple Inverted Pendulum		
Model-based	Model-free	NPG	Model-based	Model-free	NPG
0.0901	0.3061	2.1649	0.1455	0.7829	2.3209

# Morphological Computation

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

Renda et al.  
T-RO 2016

$$\begin{aligned} & \underbrace{\left[ \int_0^{L_N} \mathbf{J}^\top \mathcal{M}_a \mathbf{J} d\mathbf{X} \right] \dot{\mathbf{q}} + \left[ \int_0^{L_N} \mathbf{J}^\top \text{ad}_{\mathbf{J}\dot{\mathbf{q}}}^* \mathcal{M}_a \mathbf{J} d\mathbf{X} \right] \dot{\mathbf{q}} }_{M(\mathbf{q})} + \\ & \underbrace{\left[ \int_0^{L_N} \mathbf{J}^\top \mathcal{M}_a \mathbf{J} d\mathbf{X} \right] \dot{\mathbf{q}} + \left[ \int_0^{L_N} \mathbf{J}^\top \mathcal{D}\mathbf{J} \| \mathbf{J}\dot{\mathbf{q}} \|_p d\mathbf{X} \right] \dot{\mathbf{q}} }_{C_1(\mathbf{q}, \dot{\mathbf{q}})} - \\ & \underbrace{\left( 1 - \rho_f / \rho \right) \left[ \int_0^{L_N} \mathbf{J}^\top \mathcal{M} \text{Ad}_{\mathbf{g}}^{-1} d\mathbf{X} \right]}_{N(\mathbf{q})} \text{Ad}_{\mathbf{g}_r}^{-1} \mathcal{G} - \underbrace{\mathbf{J}^\top (\bar{\mathbf{X}}) \mathcal{F}_p}_{F(\mathbf{q})} \\ & - \underbrace{\int_0^{L_N} \mathbf{J}^\top [\nabla_x \mathcal{F}_i - \nabla_x \mathcal{F}_a + \text{ad}_{\eta_n}^* (\mathcal{F}_i - \mathcal{F}_a)] d\mathbf{X} }_{u(\mathbf{q})} = 0, \end{aligned}$$

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

# SoRo's control computational complexity is hard!

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

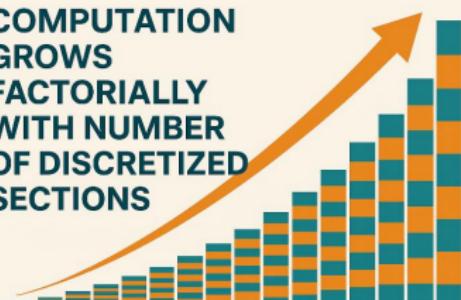
Lekan Molu and Shaoru Chen

**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

COMPUTATION  
GROWS  
FACTORIALLY  
WITH NUMBER  
OF DISCRETIZED  
SECTIONS



# Enter Singularly Perturbed Systems

$$\begin{aligned}\dot{\mathbf{z}}_1 &= \mathbf{f}(\mathbf{z}_1, \mathbf{z}_2, \epsilon, \mathbf{u}_s, t), \quad \mathbf{z}_1(t_0) = \mathbf{z}_1(0), \quad \mathbf{z}_1 \in \mathbb{R}^{6N}, \\ \epsilon \dot{\mathbf{z}}_2 &= \mathbf{g}(\mathbf{z}_1, \mathbf{z}_2, \epsilon, \mathbf{u}_f, t), \quad \mathbf{z}_2(t_0) = \mathbf{z}_2(0), \quad \mathbf{z}_2 \in \mathbb{R}^{6N}\end{aligned}$$

General SPT formulation.

$$\begin{aligned}\dot{\mathbf{z}}_1 &= \mathbf{f}(\mathbf{z}_1, \mathbf{z}_2, 0, \mathbf{u}_s, t), \quad \mathbf{z}_1(t_0) = \mathbf{z}_1(0), \\ 0 &= \mathbf{g}(\mathbf{z}_1, \mathbf{z}_2, 0, 0, t).\end{aligned}$$

Set  $\epsilon$  to 0 → Slow subsystem

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

$$\frac{d\tilde{\mathbf{z}}_2}{dT} = \epsilon \frac{d\mathbf{z}_2}{dt} - \epsilon \frac{\partial \phi}{\partial \mathbf{z}_1} \dot{\mathbf{z}}_1, \quad (8b)$$

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

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



Multiphysics, multiscale soft system.

Picture credit: Google Gemini.

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

$$0 = \mathbf{g}(\mathbf{z}_1, \phi(\mathbf{z}_1, t), 0, 0, t) \triangleq \bar{\mathbf{g}}(\mathbf{z}_1, 0, t), \quad \mathbf{z}_1(t_0) = \mathbf{z}_1(0). \quad (6)$$

The slow subsystem therefore becomes

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

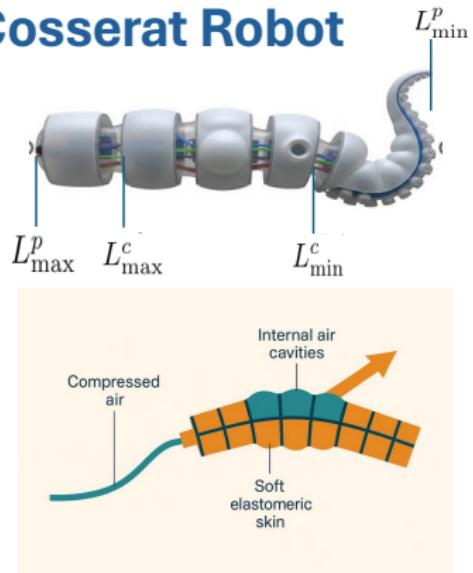
# Singularly Perturbed Soft Cosserat Robot

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

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

$$\begin{aligned} M(\mathbf{q}) &= (M^c + M^p)(\mathbf{q}), \quad N = (N^c + N^p)(\mathbf{q}), \\ F(\mathbf{q}) &= (F^c + F^p)(\mathbf{q}), \quad D(\mathbf{q}) = (D^c + D^p)(\mathbf{q}) \\ C_1(\mathbf{q}, \dot{\mathbf{q}}) &= (C_1^c + C_1^p)(\mathbf{q}, \dot{\mathbf{q}}), \\ C_2(\mathbf{q}, \dot{\mathbf{q}}) &= (C_2^c + C_2^p)(\mathbf{q}, \dot{\mathbf{q}}) \end{aligned}$$



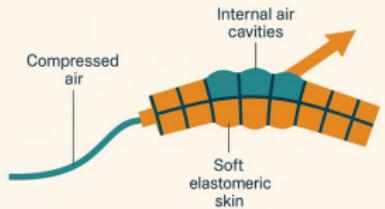
Picture credit: Google Gemini.

# Dynamics Separation with Perturbation Parameter

The mass matrix then decomposes as

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

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



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

So that we can write,

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

where

$$\mathbf{s} = \begin{bmatrix} \mathbf{s}_{\text{fast}} \\ \mathbf{s}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \mathbf{F}^c + \mathbf{N}^c \text{Ad}_{g_r}^{-1} \mathcal{G} - [\mathbf{C}_1^c + \mathbf{C}_2^c + \mathbf{D}^c] \mathbf{z}_{\text{fast}} \\ \mathbf{F}^p + \mathbf{N}^p \text{Ad}_{g_r}^{-1} \mathcal{G} - [\mathbf{C}_1^p + \mathbf{C}_2^p + \mathbf{D}^p] \mathbf{z}_{\text{slow}} \end{bmatrix}. \quad (13)$$

# Singularly perturbed soft robot form

Suppose that

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

Then, we may write

$$\begin{bmatrix} \mathcal{H}_{\text{fast}} & \bar{M}_{12}^p \\ 0 & \bar{M}_{22}^p \end{bmatrix} \begin{bmatrix} \dot{\mathbf{z}}_{\text{fast}} \\ \epsilon \dot{\mathbf{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} s_{\text{fast}} \\ s_{\text{slow}} - \epsilon \bar{M}_{21}^p \mathcal{H}_{\text{fast}}^{-1} s_{\text{fast}} \end{bmatrix} + \begin{bmatrix} u_{\text{fast}} \\ u_{\text{slow}} - \epsilon \bar{M}_{21}^p \mathcal{H}_{\text{fast}}^{-1} u_{\text{fast}} \end{bmatrix} \quad (16)$$

Fast subdynamics extraction

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

Then,  $\dot{\mathbf{z}}_{\text{fast}} = \frac{d\mathbf{z}_{\text{fast}}}{dt} \equiv \frac{1}{\epsilon} \frac{d\mathbf{z}_{\text{fast}}}{dT} \triangleq \frac{1}{\epsilon} \mathbf{z}'_{\text{fast}}$   
and  $\epsilon \dot{\mathbf{z}}_{\text{slow}} = \mathbf{z}'_{\text{slow}}$ .

So that,

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

# A backstepping nonlinear multi-scale controller

*Theorem 1:* The control law

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

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

Where,

$$[\boldsymbol{\theta}^\top, \boldsymbol{\phi}^\top]^\top = [\mathbf{q}_{\text{fast}}^\top, \mathbf{z}_{\text{fast}}^\top]^\top \text{ where } \boldsymbol{\theta}' = \epsilon \mathbf{z}_{\text{fast}}$$

*Theorem 2:* Under the tracking error  $e_2 = \boldsymbol{\phi} - \boldsymbol{\nu}$  and matrices  $(\mathbf{K}_p, \mathbf{K}_q) = (\mathbf{K}_p^\top, \mathbf{K}_q^\top) > 0$ , the control input

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

exponentially stabilizes the fast subdynamics (18).

*Theorem 3:* The control law

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

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  $\Sigma(z_{\text{fast}}, z_{\text{slow}})$  as a weighted combination of  $\mathbf{V}_2$  and  $\mathbf{V}_3$  i.e. ,

$$\Sigma(z_{\text{fast}}, z_{\text{slow}}) = (1 - \varepsilon)\mathbf{V}_2(z_{\text{fast}}) + \varepsilon\mathbf{V}_3(z_{\text{slow}}), \quad 0 < \varepsilon < 1. \quad (35)$$

It follows that,

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

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.

$$\mathbf{u}(z_{\text{fast}}, z_{\text{slow}}) = (1 - \varepsilon)\mathbf{u}_{\text{fast}} + \varepsilon\mathbf{u}_{\text{slow}}. \quad (37)$$

# Numerical Results

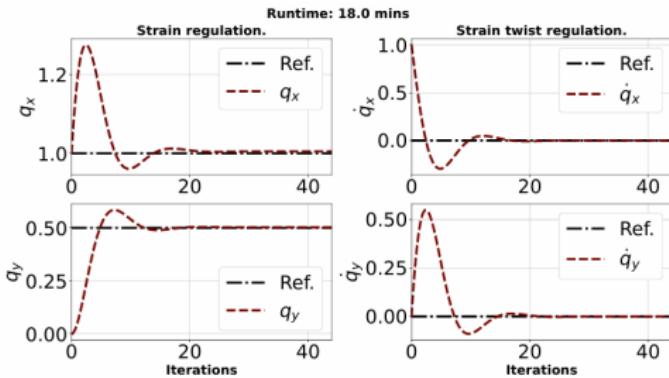


Fig. 2. Backstepping control on the singularly perturbed soft robot system with 10 discretized pieces, divided into 6 fast and 4 slow pieces. For a tip load of  $\mathcal{F}_p^y = 10 \text{ N}$ , the backstepping gains were set as  $K_p = 10$ ,  $K_d = 2.0$  for a desired joint configuration  $\xi^d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

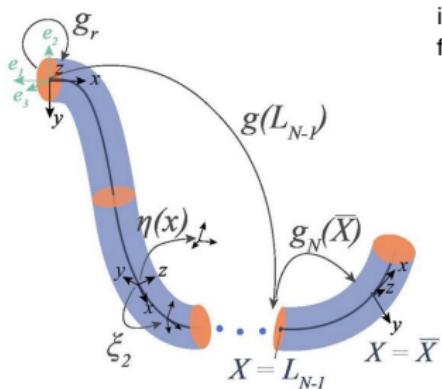
Total	Pieces		Runtime (mins)	
	Fast	Slow	Hierarchical SPT (mins)	Single-layer PD control (hours)
6	4	2	18.01	51.46
8	5	3	30.87	68.29
10	7	3	32.39	107.43

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

$$g_r = \begin{pmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

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

$$\mathcal{F}_p = \text{diag} (\mathbf{R}^T(L), \mathbf{R}^T(L)) \left( \begin{array}{cccc} \mathbf{0}_{3 \times 1} & 0 & 10 & 0 \end{array} \right)^T$$

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

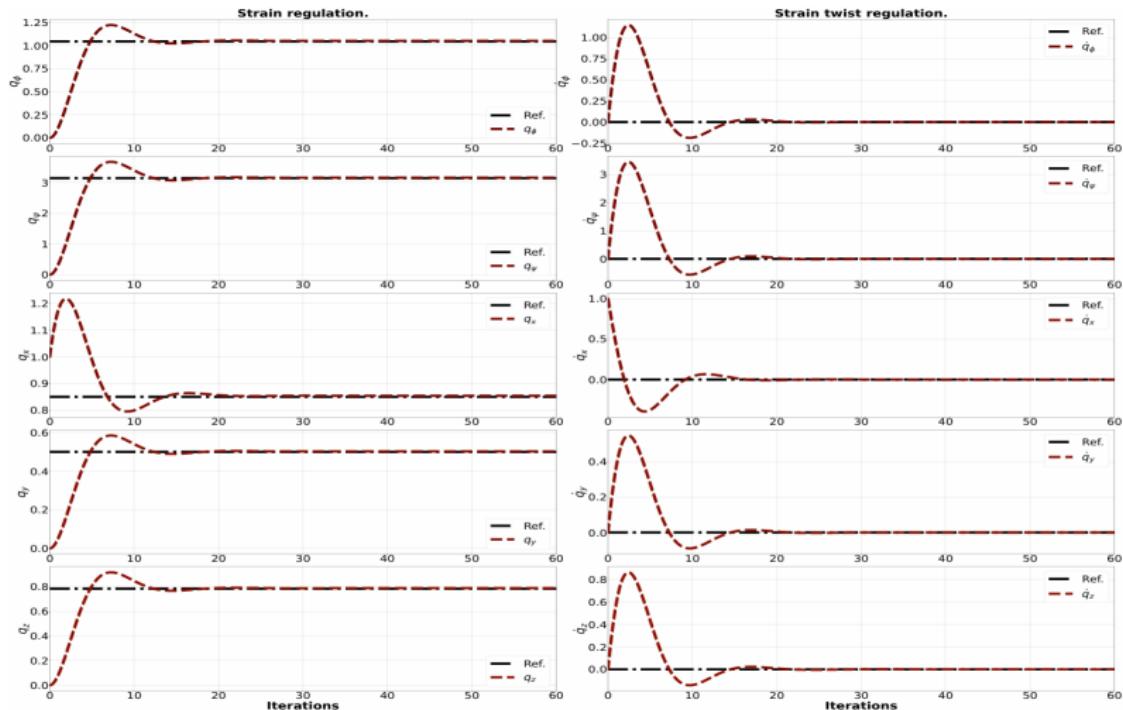


Fig. 3. Backstepping control on the singularly perturbed soft robot system with 10 pieces 4 slow and 6 fast sections.

# Appendix – AC States Algorithm and Results

This page is left blank intentionally.

# Exogenous Markov Decision Process (Exo-MDP) Machinery

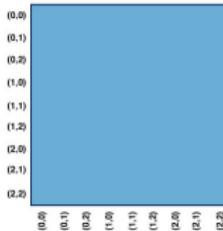
- Consider the tuple  $\mathcal{M} := (\mathcal{X}, \mathcal{Z}, \mathcal{A}, T, R, H)$ 
  - Starting distribution  $\mu \in \Delta(\mathcal{Z})$ ;
  - Agent receives observations  $\{x_h\}_{h=1}^H \in \mathcal{X}$  from the emission function  $q : \mathcal{Z} \rightarrow \Delta(\mathcal{X})$ ;
  - Agent transitions between latent states via  $T : \mathcal{Z} \times \mathcal{A} \rightarrow \Delta(\mathcal{S})$ ;
    - And rewards by  $R : \mathcal{X} \times \mathcal{A} \rightarrow \Delta([0, 1])$
- Trajectories:  $(z_1, x_1, a_1, r_1, \dots, z_H, a_H, r_H)$  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

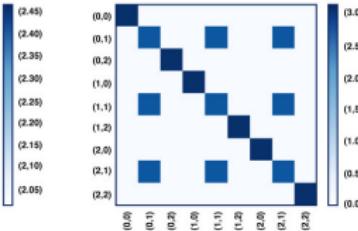
Block MDP assumption  $\text{supp}(q(\cdot|z_1)) \cap \text{supp}(q(\cdot|z_2)) = \emptyset$  for all  $z_1 \neq z_2$ .

- Agent chooses  $a \sim \pi(z_h|x_h)$
- There exists non-stationary episodic policies  
 $\Pi_{NS} := \Pi^H \supseteq (\pi_1, \dots, \pi_H);$
- Optimal policy  
 $\pi^* = \operatorname{argmax}_{\pi \in \Pi_{NS}} V_{\pi \in \Pi_{NS}}(\pi);$ 
  - 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  $s \in \mathcal{S}$ ,  $e \in \mathcal{E}$ .
- $\mu(z) = \mu(s)\mu\xi$  and  
 $T(z'|z, a) = T(s'|s, a)T_e(e'|e)$

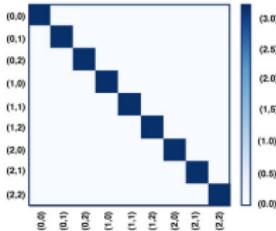
# Agent Controllable States Representation



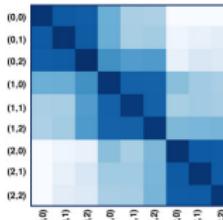
**(a) Autoencoder  
(Theory worst-case)**



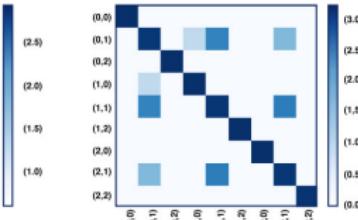
**(b) Inverse  
(Theory worst-case)**



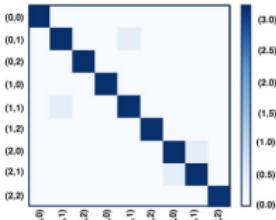
**(c) AC-State  
(Theory worst-case)**



**(d) Autoencoder  
(Empirical)**

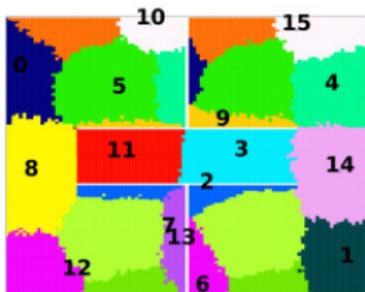


**(e) Inverse  
(Empirical)**



**(f) AC-State  
(Empirical)**

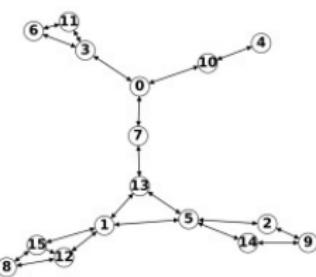
# PCLAST Segmentation Results



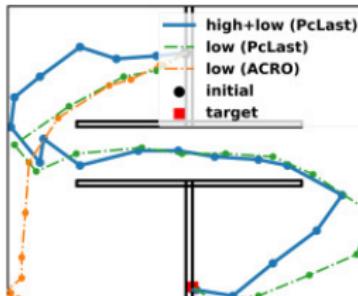
(a) Clusters ACRO



(b) Clusters PCLAST

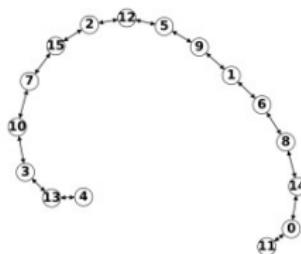
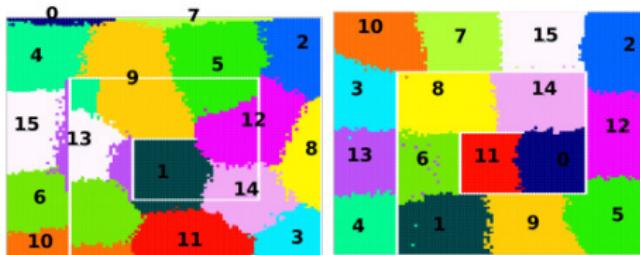


(c) State-transitions PCLAST

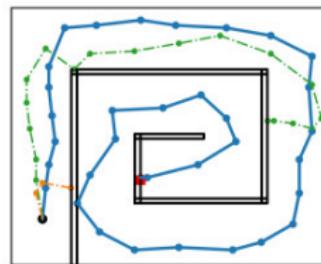


(d) Planning Trajectories

# PCLAST Segmentation Results



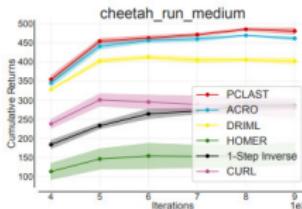
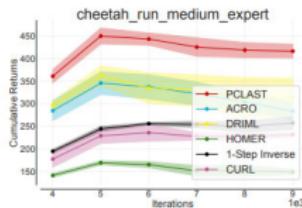
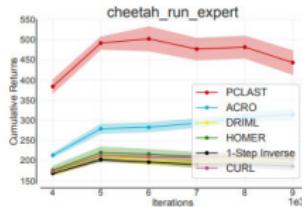
(c) State-transitions PCLAST



(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.

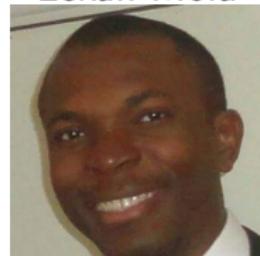
# Credits

Shaoru Chen



Postdoc, MSR

Lekan Molu

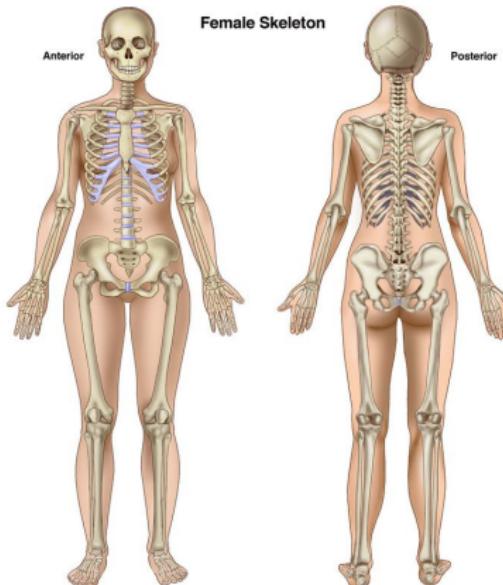


Senior Researcher, MSR

# 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  $\approx 11\%$  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.

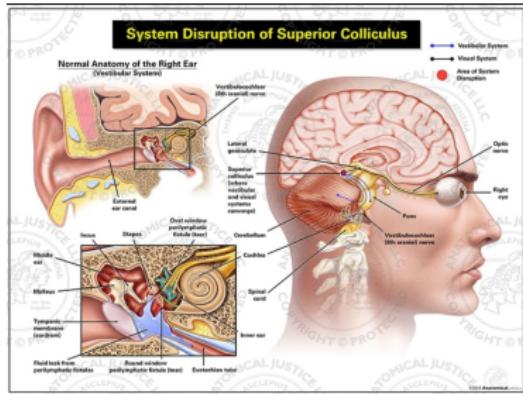
# Simplexity in Morphological Computation

- **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!**

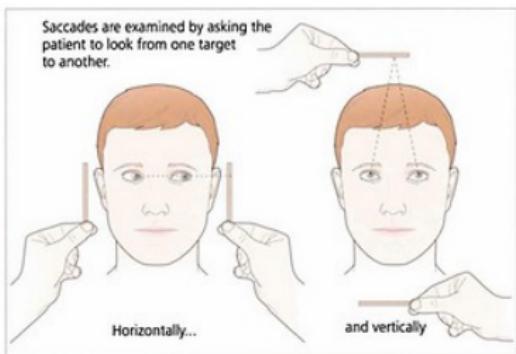
# Simplexity: The Central Pattern Generator

- 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

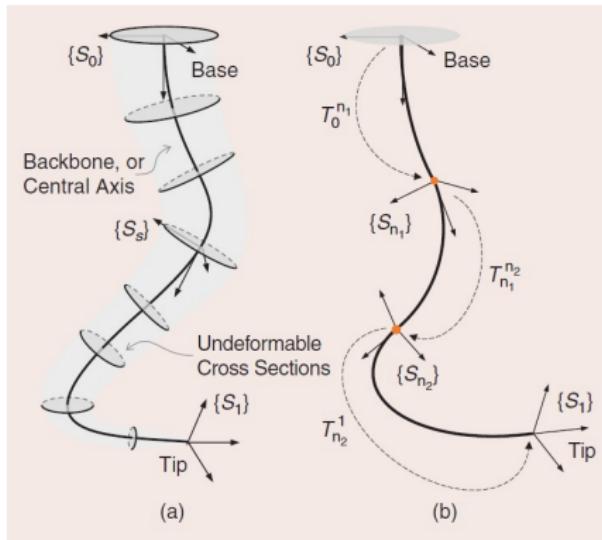


Octopus. ©Smithsonian Magazine

# 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!
- Simplicity 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.

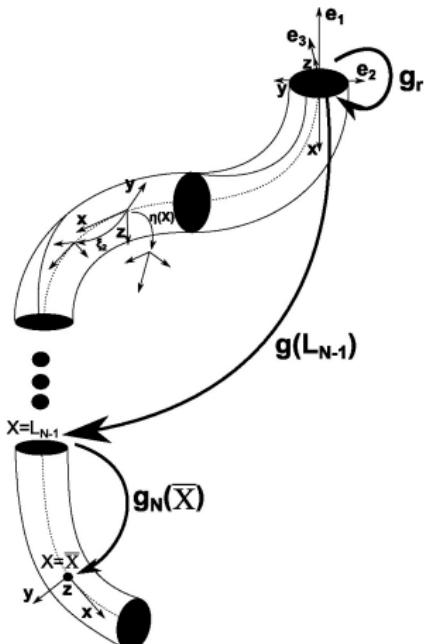
## Tractable reduced-order models

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

# The piecewise constant strain model

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

# Dynamic equations

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

$$\begin{aligned}
 & \underbrace{\left[ \int_0^{L_N} \mathbf{J}^T \mathcal{M}_a \mathbf{J} dX \right]}_{\mathbf{M}(\mathbf{q})} \ddot{\mathbf{q}} + \underbrace{\left[ \int_0^{L_N} \mathbf{J}^T \text{ad}_{\dot{\mathbf{q}}}^* \mathcal{M}_a \mathbf{J} dX \right]}_{\mathcal{C}_1(\mathbf{q}, \dot{\mathbf{q}})} \dot{\mathbf{q}} + \underbrace{\left[ \int_0^{L_N} \mathbf{J}^T \mathcal{M}_a \mathbf{J} dX \right]}_{\mathcal{C}_2(\mathbf{q}, \dot{\mathbf{q}})} \dot{\mathbf{q}} \\
 & + \underbrace{\left[ \int_0^{L_N} \mathbf{J}^T \mathcal{D} \mathbf{J} \| \mathbf{J} \dot{\mathbf{q}} \|_p dX \right]}_{\mathcal{D}(\mathbf{q}, \dot{\mathbf{q}})} \dot{\mathbf{q}} - (1 - \rho_f / \rho) \underbrace{\left[ \int_0^{L_N} \mathbf{J}^T \mathcal{M} \text{Ad}_{\mathbf{g}}^{-1} dX \right]}_{\mathbf{N}(\mathbf{q})} \text{Ad}_{\mathbf{g}_r}^{-1} \mathcal{G} \\
 & - \underbrace{\mathbf{J}(\bar{X})^T \mathcal{F}_p}_{\mathcal{F}(\mathbf{q})} - \underbrace{\int_0^{L_N} \mathbf{J}^T [\nabla_x \mathcal{F}_i - \nabla_x \mathcal{F}_a + \text{ad}_{\xi_n}^* (\mathcal{F}_i - \mathcal{F}_a)] dX}_{\tau(\mathbf{q})} = 0, \quad (1)
 \end{aligned}$$

## Structural properties – mass inertia operator

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

### Property 1 (Boundedness of the Mass Matrix)

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

### Proof of Property 1.

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



# Structural properties – parameters Identification

## Property 2 (Linearity-in-the-parameters)

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

$$\begin{aligned}\ddot{\boldsymbol{M}(\boldsymbol{q})\boldsymbol{(\ddot{\boldsymbol{q}})} + [\boldsymbol{C}_1(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}_2(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{D}(\boldsymbol{q}, \dot{\boldsymbol{q}})]\dot{\boldsymbol{q}} - \boldsymbol{F}(\boldsymbol{q})\boldsymbol{N}(\boldsymbol{q})\boldsymbol{Ad}_{\boldsymbol{g}_r}^{-1}\boldsymbol{\mathcal{G}}} \\ = \boldsymbol{Y}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}})\boldsymbol{\Theta}.\end{aligned}\quad (3)$$

# Structural properties – skew symmetry of system inertial forces

## Property 3 (Skew symmetric property)

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

# Skew-symmetric of robot's mass and Coriolis forces

By Leibniz's rule, we have

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

Therefore,  $\dot{\mathcal{M}}(\mathbf{q}) - 2 [C_1(\mathbf{q}, \dot{\mathbf{q}}) + C_2(\mathbf{q}, \dot{\mathbf{q}})]$  becomes

$$\int_0^{L_N} \left( \mathbf{J}^T \mathcal{M}_a \mathbf{J} + \mathbf{J}^T \dot{\mathcal{M}}_a \mathbf{J} + \mathbf{J}^T \mathcal{M}_a \dot{\mathbf{J}} \right) dX - 2 \int_0^{L_N} \left( \mathbf{J}^T \text{ad}_{\dot{\mathbf{q}}}^* \mathcal{M}_a \mathbf{J} + \mathbf{J}^T \mathcal{M}_a \dot{\mathbf{J}} \right) dX\quad (5)$$

$$\triangleq \int_0^{L_N} \left( \mathbf{J}^T \mathcal{M}_a \mathbf{J} + \mathbf{J}^T \dot{\mathcal{M}}_a \mathbf{J} - \mathbf{J}^T \mathcal{M}_a \dot{\mathbf{J}} \right) dX - 2 \int_0^{L_N} \mathbf{J}^T \text{ad}_{\dot{\mathbf{q}}}^* \mathcal{M}_a \mathbf{J} dX.\quad (6)$$

## Skew-Symmetric Property Proof

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

$$\begin{aligned}
 & -\dot{\mathcal{M}}^\top(\mathbf{q}) + 2\left[\mathcal{C}_1^\top(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{C}_2^\top(\mathbf{q}, \dot{\mathbf{q}})\right] = \\
 & \int_0^{L_N} dX^\top \left(-\mathbf{J}^\top \mathcal{M}_a \mathbf{j} - \mathbf{J}^\top \dot{\mathcal{M}}_a \mathbf{J} - \mathbf{j}^\top \mathcal{M}_a \mathbf{J}\right) + 2 \int_0^{L_N} dX^\top \left(\mathbf{J}^\top \mathcal{M}_a \text{ad}_{\mathbf{J}\dot{\mathbf{q}}} \mathbf{J} + \mathbf{j}^\top \mathcal{M}_a \mathbf{J}\right) \\
 & \triangleq \int_0^{L_N} \left(\mathbf{J}^\top \mathcal{M}_a \mathbf{j} - \mathbf{j}^\top \mathcal{M}_a \mathbf{J} - \mathbf{J}^\top \dot{\mathcal{M}}_a \mathbf{J}\right) dX - 2 \int_0^{L_N} \mathbf{J}^\top \text{ad}_{\mathbf{J}\dot{\mathbf{q}}}^* \mathcal{M}_a \mathbf{J} dX \quad (7)
 \end{aligned}$$

which satisfies the identity:

$$\begin{aligned}
 & \dot{\mathcal{M}}(\mathbf{q}) - 2[\mathcal{C}_1(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{C}_2(\mathbf{q}, \dot{\mathbf{q}})] = \\
 & -\left[\dot{\mathcal{M}}(\mathbf{q}) - 2[\mathcal{C}_1(\mathbf{q}, \dot{\mathbf{q}}) + \mathcal{C}_2(\mathbf{q}, \dot{\mathbf{q}})]\right]^\top. \quad (8)
 \end{aligned}$$

*A fortiori*, the skew symmetric property follows.

## MC Takeaways: Simplexity

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

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

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

## Control exploiting structural properties

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

### Theorem 1 (Cable-driven Actuation)

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

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

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

# Computational Control exploiting structural properties

## Corollary 2 (Fluid-driven actuation)

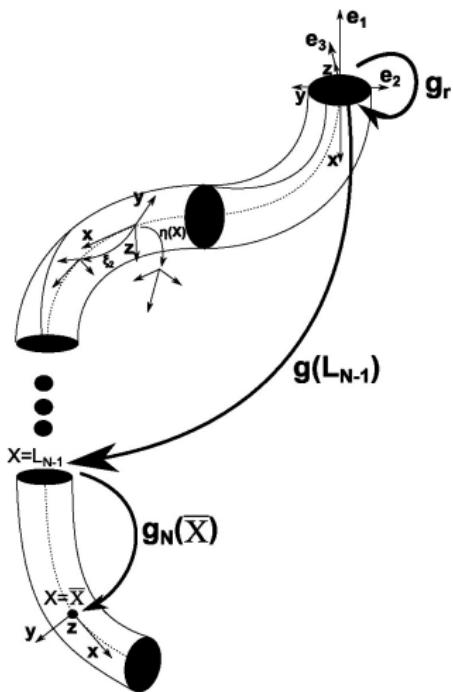
*If the robot is operated without cables, and is driven with a dense medium such as pressurized air or water, then the term  $F(\mathbf{q}) = 0$  so that the control law  $\mathbf{u}(\mathbf{q}, \dot{\mathbf{q}}) = -\mathbf{K}_p \tilde{\mathbf{q}} - \mathbf{K}_D \dot{\mathbf{q}}$  globally asymptotically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

## Proof.

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

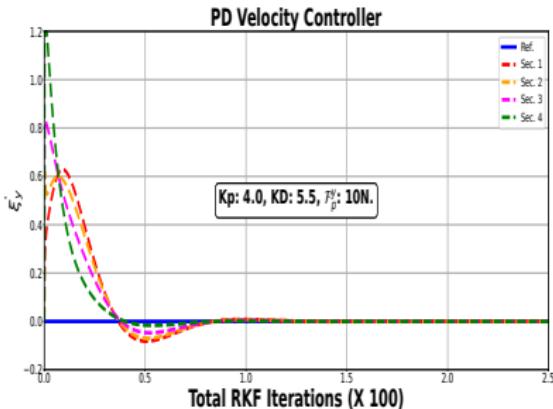


# Robot parameters

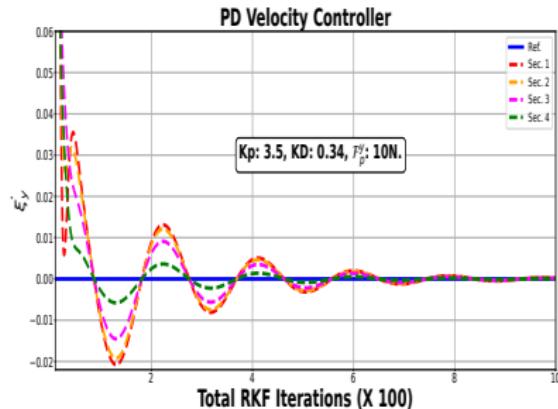


- Tip load in the  $+y$  direction in the robot's base frame.
- Poisson ratio: 0.45;  
 $\mathcal{M} = \rho[I_x, I_y, I_z, A, A, A]$  with  
 $\rho = 2,000 \text{ kg m}^{-3}$ ;
- $\mathbf{D} = -\rho_w \nu^T \check{\mathbf{D}} \nu / |\nu|$ .
- $X \in [0, L]$  discretized into 41 segments.

# Computational Control exploiting structural properties

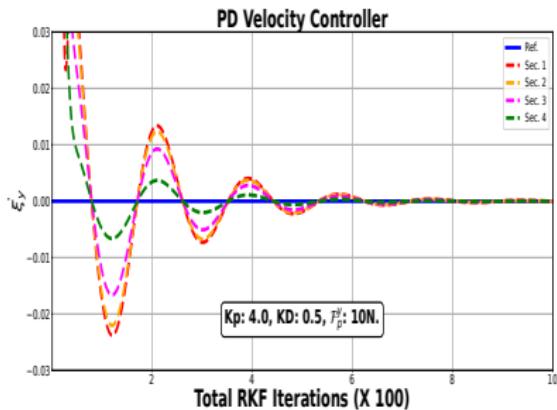


Cable-driven, strain twist setpoint  
 terrestrial control.

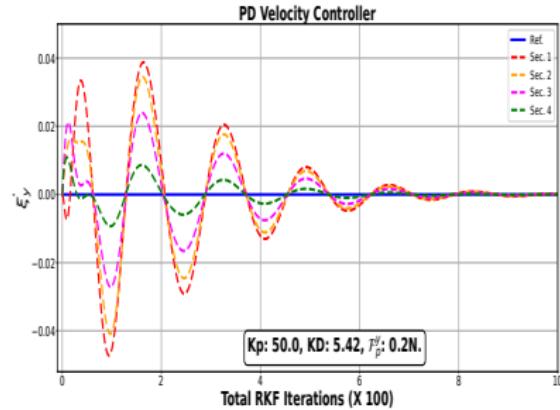


Fluid-actuated, strain twist setpoint  
 terrestrial control.

# Computational Control exploiting structural properties

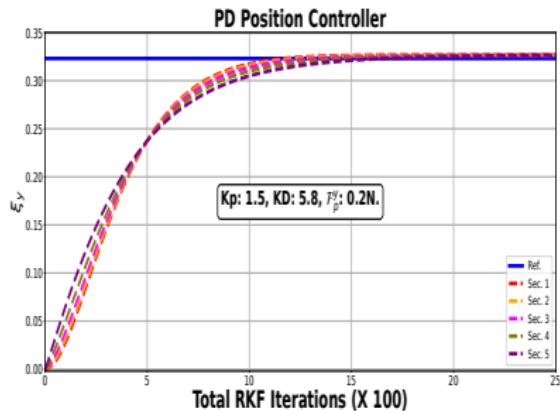


Fluid-actuated, strain twist setpoint  
 underwater control.

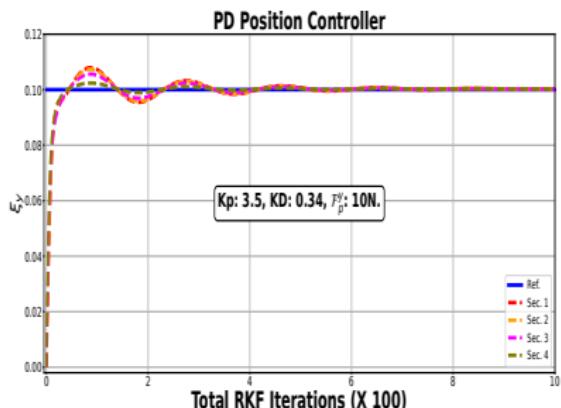


Cable-driven, strain twist setpoint  
 regulation.

# 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:  $[\mathbf{q}, \dot{\mathbf{q}}] = [\mathbf{q}, \mathbf{z}]$ , rewrite (2):

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

# Dynamics separation

Suppose that  $\mathbf{M}^p = \int_{L_{\min}^p}^{L_{\max}^p} \mathbf{J}^\top \mathcal{M}^{pert} \mathbf{J} dX$ , and  $\mathbf{M}^c = \int_{L_{\min}^c}^{L_{\max}^c} \mathbf{J}^\top \mathcal{M}^{core} \mathbf{J} dX$ , then,

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

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

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

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

# Dynamics Separation

Furthermore, let

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

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

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

# SoRo Dynamics Separation

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

where

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

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

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

# SoRo Dynamics Separation

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

$$\begin{bmatrix} \mathcal{H}_{\text{fast}} & \bar{\mathbf{M}}_{12}^P \\ \mathbf{0} & \bar{\mathbf{M}}_{22}^P \end{bmatrix} \begin{bmatrix} \dot{\mathbf{z}}_{\text{fast}} \\ \epsilon \dot{\mathbf{z}}_{\text{slow}} \end{bmatrix} = \begin{bmatrix} \mathbf{s}_{\text{fast}} \\ \mathbf{s}_{\text{slow}} - \epsilon \bar{\mathbf{M}}_{21}^P \mathcal{H}_{\text{fast}}^{-1} \mathbf{s}_{\text{fast}} \end{bmatrix} + \begin{bmatrix} \mathbf{u}_{\text{fast}} \\ \mathbf{u}_{\text{slow}} - \epsilon \bar{\mathbf{M}}_{21}^P \mathcal{H}_{\text{fast}}^{-1} \mathbf{u}_{\text{fast}} \end{bmatrix} \quad (15)$$

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

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

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

## SoRo Fast Subsystem Extraction

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

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

; and

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

Fast subdynamics:

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

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

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

## SoRo Slow Subsystem Extraction

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

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

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:  $\begin{bmatrix} \theta \\ \phi \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{\text{fast}} \\ \mathbf{z}_{\text{fast}} \end{bmatrix}$  so that  $\theta' = \epsilon \mathbf{z}_{\text{fast}} \triangleq \nu :=$  A virtual input.
- Let  $\{\mathbf{q}_{\text{fast}}^d, \dot{\mathbf{q}}_{\text{fast}}^d\} = \{\xi_1^d, \dots, \xi_{n_\xi}^d, \eta_1^d, \dots, \eta_{n_\xi}^d\}_{\text{fast}}$  be the desired joint space configuration for the fast subsystem.

### Theorem 3 ([6])

*The control law*

$$\mathbf{u}_{\text{fpos}} = \mathbf{q}_{\text{fast}}^d(t_f) - \mathbf{q}_{\text{fast}}(t_f) + \dot{\mathbf{q}}_{\text{fast}}^d(t_f)$$

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

# Control of the Fast Strain Subdynamics

Proof Sketch 1 (Proof of Theorem 3)

$$\mathbf{e}_1 = \theta - \mathbf{q}_{fast}^d, \implies \mathbf{e}'_1 = \theta' - \mathbf{q}'^d_{fast} \triangleq \nu - \mathbf{q}'^d_{fast}. \quad (19)$$

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

Then,  $\mathbf{V}'_1 = \mathbf{e}'_1^\top \mathbf{K}_p \mathbf{e}'_1 = \mathbf{e}_1^\top \mathbf{K}_p (\nu - \mathbf{q}'^d_{fast}).$  (21)

For  $\nu = \mathbf{q}'^d_{fast} - \mathbf{e}_1$ ,  $\mathbf{V}'_1 = -\mathbf{e}_1^\top \mathbf{K}_p \mathbf{e}_1 \leq 2\mathbf{V}_1.$

# Stability Analysis of the Fast Velocity Subdynamics

## Theorem 4 ([6])

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

$$\begin{aligned}\mathbf{u}_{fvel} = & \frac{1}{\epsilon} \mathcal{H}_{fast} [\mathbf{q}_{fast}^{\prime\prime d} + \mathbf{e}_1 - 2\mathbf{e}_2 - \mathbf{K}_q^\top (\mathbf{K}_q \mathbf{K}_q^\top)^{-1} \mathbf{K}_p \mathbf{e}_1] \\ & + \frac{1}{\epsilon} \mathcal{H}_{slow}^{\text{fast}} \mathbf{z}'_{slow} - \mathbf{s}_{fast}\end{aligned}\quad (22)$$

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:*

$$\mathbf{e}'_1 = \theta' - \mathbf{q}'^d_{fast} \triangleq \mathbf{z}_{fast} - \mathbf{q}'^d_{fast} + (\boldsymbol{\nu} - \boldsymbol{\nu}) \quad (23a)$$

$$= (\phi - \boldsymbol{\nu}) + (\boldsymbol{\nu} - \mathbf{q}'^d_{fast}) \triangleq \mathbf{e}_2 - \mathbf{e}_1. \quad (23b)$$

*It follows that*

$$\mathbf{e}'_2 = \phi' - \boldsymbol{\nu}' = \mathbf{z}'_{fast} + \mathbf{e}'_1 - \mathbf{q}''^d_{fast} \quad (24)$$

$$= \mathcal{H}_{fast}^{-1} \left[ \epsilon \mathbf{u}_{fast} + \epsilon \mathbf{s}_{fast} - \mathcal{H}_{slow}^{fast} \mathbf{z}'_{slow} \right] + (\mathbf{e}_2 - \mathbf{e}_1) - \mathbf{q}''^d_{fast}.$$

# Stability Analysis of the Fast Velocity Subdynamics

## Proof Sketch 3 (Sketch Proof of Theorem 4)

For diagonal matrices  $K_p, K_q$  with positive damping, let us choose the Lyapunov candidate function

$$V_2(\mathbf{e}_1, \mathbf{e}_2) = V_1 + \frac{1}{2} \mathbf{e}_2^\top K_q \mathbf{e}_2 = \frac{1}{2} [\mathbf{e}_1 \ \mathbf{e}_2] \begin{bmatrix} K_p & \mathbf{0} \\ \mathbf{0} & K_q \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{bmatrix}.$$

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

$$\begin{aligned} u_{fvel} = & \frac{1}{\epsilon} \mathcal{H}_{fast} [\mathbf{q}''^d_{fast} - \tilde{\mathbf{q}}_{fast} - 2\tilde{\mathbf{q}}'_{fast} - K_q^\top (K_q K_q^\top)^{-1} K_p \tilde{\mathbf{q}}_{fast}] \\ & + \frac{1}{\epsilon} \mathcal{H}_{slow}^{\text{fast}} \mathbf{z}'_{slow} - s_{fast}, \end{aligned}$$

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*

$$\begin{aligned} \mathbf{V}'_2 &= \mathbf{e}_1^\top \mathbf{K}_p (\mathbf{e}_2 - \mathbf{e}_1) \\ &\quad - \mathbf{e}_2^\top \mathbf{K}_q \left( \mathbf{e}_2 - \mathbf{K}_q^\top (\mathbf{K}_q \mathbf{K}_q^\top)^{-1} \mathbf{K}_p \mathbf{e}_1 \right) \end{aligned} \tag{25a}$$

$$= -\mathbf{e}_1^\top \mathbf{K}_p \mathbf{e}_1 - \mathbf{e}_2^\top \mathbf{K}_q \mathbf{e}_2 \tag{25b}$$

$$\triangleq -2\mathbf{V}_2 \leq 0. \tag{25c}$$

# Stability analysis of the slow subdynamics

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

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

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

## Theorem 5

*The control law*

$$\mathbf{u}_{\text{slow}} = \mathcal{H}_{\text{slow}}(\mathbf{e}_1 - \mathbf{e}_2 - \mathbf{e}_3 + \ddot{\mathbf{q}}_{\text{fast}}^d) - \mathbf{s}_{\text{slow}} \quad (27)$$

*exponentially stabilizes the slow subdynamics.*

# Stability analysis of the slow subdynamics

Proof.

Consider the Lyapunov function candidate

$$V_3(\mathbf{e}_3) = \frac{1}{2} \mathbf{e}_3^\top \mathbf{K}_r \mathbf{e}_3 \text{ where } \mathbf{K}_r = \mathbf{K}_r^\top > 0. \quad (28)$$

It follows that

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

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

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

$$\dot{V}_3(\mathbf{e}_3) = \mathbf{e}_3^\top \mathbf{K}_r \mathbf{e}_3 \triangleq -2V_3(\mathbf{e}_3) \leq 0. \quad (30)$$

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  $\Sigma(z_{\text{fast}}, z_{\text{slow}})$  as a weighted combination of  $V_2$  and  $V_3$  i.e. ,

$$\Sigma(z_{\text{fast}}, z_{\text{slow}}) = (1 - \varepsilon)V_2(z_{\text{fast}}) + \varepsilon V_3(z_{\text{slow}}), \quad 0 < \varepsilon < 1. \quad (31)$$

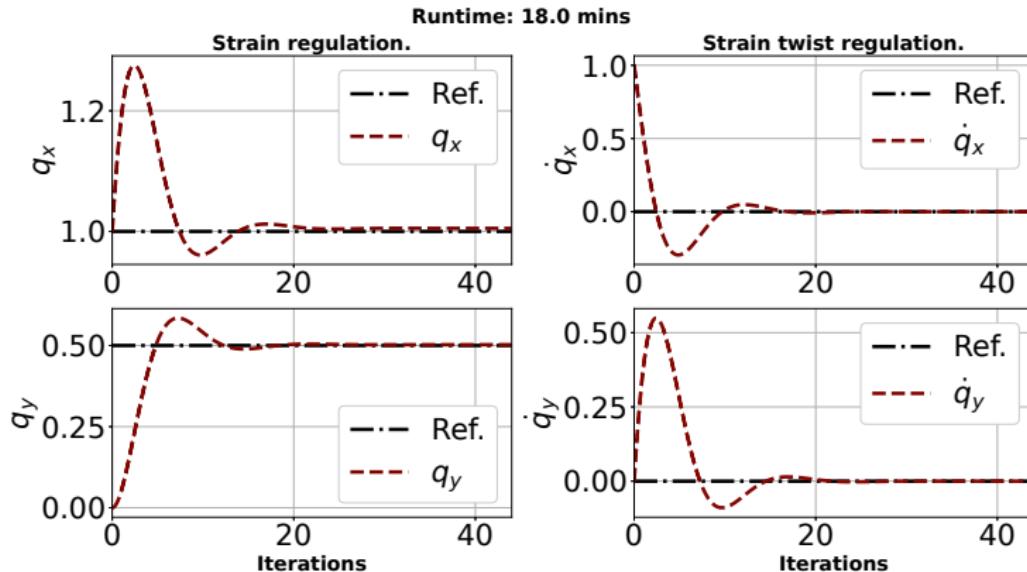
It follows that,

$$\begin{aligned} \dot{\Sigma}(z_{\text{fast}}, z_{\text{slow}}) &= (1 - \varepsilon)[\mathbf{e}_1^\top K_p \dot{\mathbf{e}}_1 + \mathbf{e}_2^\top K_q \dot{\mathbf{e}}_2] + \varepsilon \mathbf{e}_3^\top K_r \dot{\mathbf{e}}_3, \\ &= -2(V_2 + V_3) + 2\varepsilon V_2 \leq 0 \end{aligned} \quad (32)$$

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.

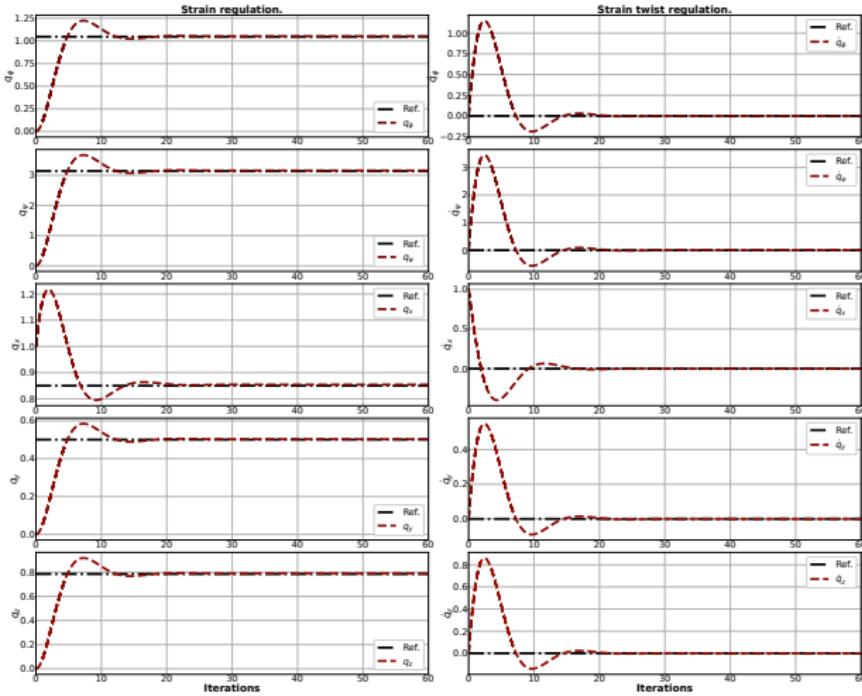
$$\mathbf{u}(z_{\text{fast}}, z_{\text{slow}}) = (1 - \varepsilon)\mathbf{u}_{\text{fast}} + \varepsilon \mathbf{u}_{\text{slow}}. \quad (33)$$

# Asynchronous, time-separated control



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

# 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.