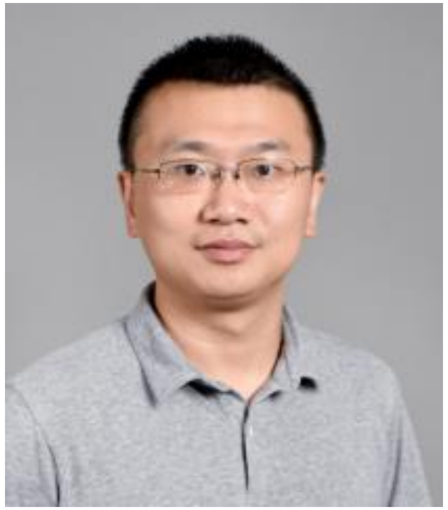
Safe Distributional-Reinforcement Learning-Enabled Systems

Presenter: Xian Yu @ OSU

NSF Safe RL Workshop 2025

Lei Ying @ U of Michigan



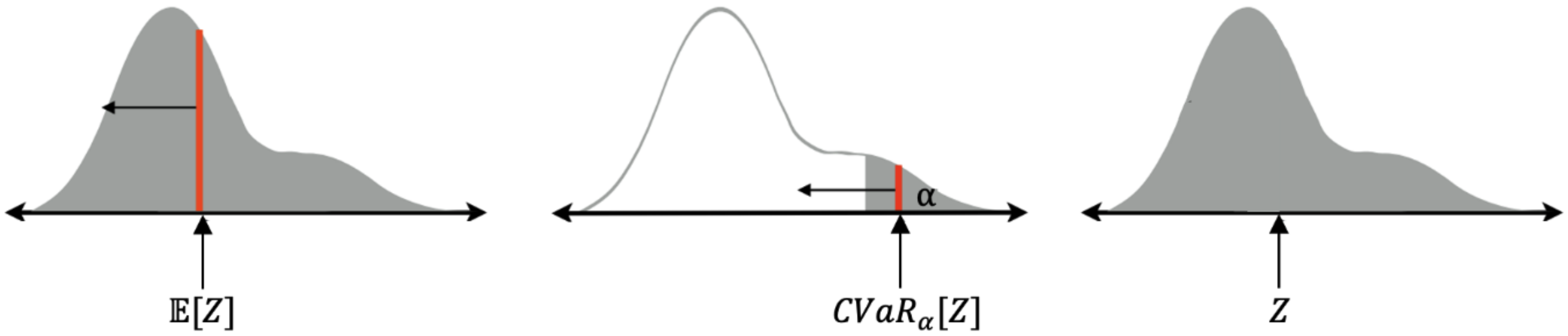
Wenlong Zhang @ ASU

Yongming Liu @ ASU

Xian Yu @ OSU

Risk-Sensitive DRL

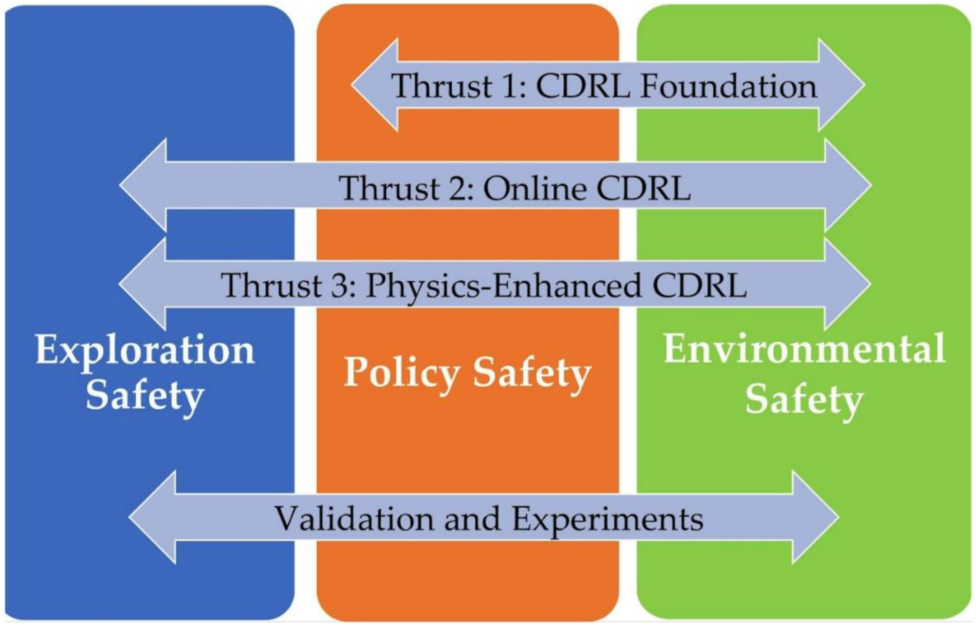
- Traditional (risk-neutral) RL
  - Only preserves the expectation: not enough information to make safe decisions.
- Risk-sensitive RL
  - Maintains a point estimate of a certain risk measure: reduces the possibility of experiencing adverse rewards but not applicable to other risk measures.
- Distributional RL
  - Estimates the entire reward distribution: provides a unified framework for integrating different risk measures.

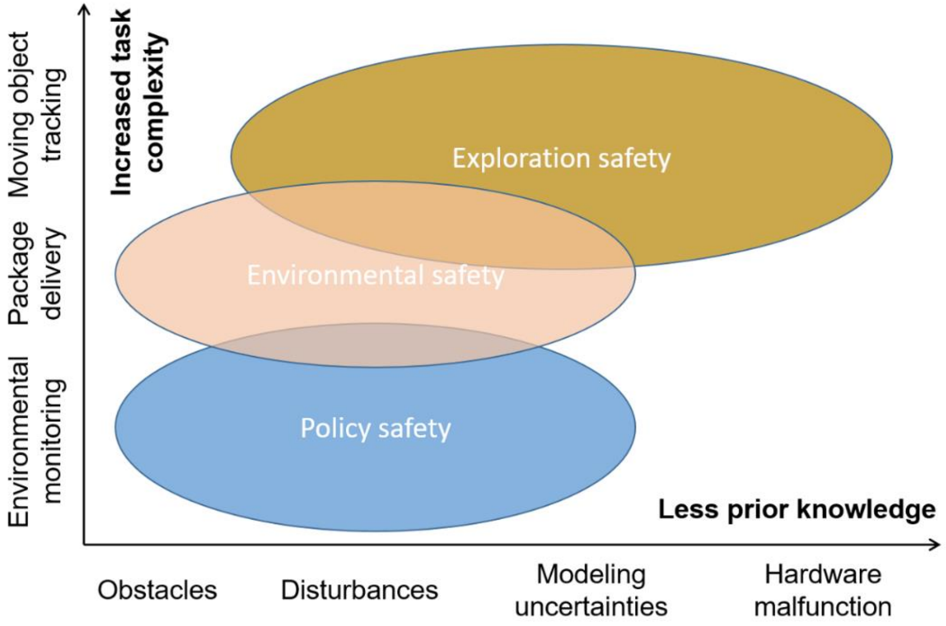


(a) Risk-neutral RL (b) Risk-sensitive RL (c) Distributional RL

End-to-End Safety

- **Policy Safety** concerns solving a risk-sensitive Constrained DRL (CDRL) problem.
- **Exploration Safety** concerns the safety when learning the safe policy.
- **Environmental Safety** concerns model misspecification and nonstationarity when solving the problem.





Thrust 1: Policy Gradient Methods for Risk-Sensitive DRL with Provable Convergence

Minheng Xiao[1], Xian Yu[1], Lei Ying[2], [1]The Ohio State University, [2]University of Michigan

Background and Objective

We propose a distributional policy gradient algorithm that aims to solve a risk-sensitive RL problem with any coherent risk measure:

$$\min_{\theta} \rho(Z_\theta^s)$$

Compared to other NN-based policy gradient algorithms (e.g., D4PG, SDPG), our approach CDPG

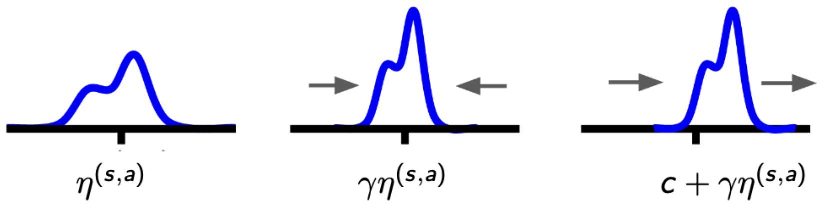
- Leverages distributional Bellman equation to derive an **analytical gradient form**;
- Has **finite-time convergence guarantee** under both exact and inexact policy evaluation.

Distributional Policy Gradient Theorem

Gradient of the reward probability measure:

$$\nabla_\theta \eta_\theta^s = \mathbb{E}_{\tau_\theta}\left[g(s_0) + \sum_{t=1}^{|\tau_\theta|} \mathcal{B}^{\tau_\theta(s_0,s_t)} g(s_t) \right] \qquad (4)$$

where $g(s) := \sum_{a\in\mathcal{A}} \nabla_\theta \pi_\theta(a|s)\eta_\theta^{(s,a)}$ *and* $\mathcal{B}^{\tau_\theta(s_0,s_t)}$ *is the t-step pushforward operator, defined as* $\mathcal{B}^{\tau_\theta(s_0,s_t)} := (b_{c_0,\gamma})_\# \dots (b_{c_{t-1},\gamma})_\# = (b_{c_{t-1}+\gamma c_{t-2}+\dots+\gamma^{t-1}c_0,\gamma^t})_\#$.



Distributional Policy Gradient Algorithm

Algorithm 1 Distributional Policy Gradient Algorithm

Require: Initial Parameter θ_1, Stepsize δ

for $t = 1, \dots, T$ **do**
  if $\|\nabla_\theta \rho(Z_{\theta_t}^s)\| < \epsilon$ **then**
    Return θ_t
  end if
  # Distributional Policy Evaluation
  while *not converged* **do**
    $\eta_{\theta_t} \leftarrow \mathcal{T}^{\theta_t}\eta_{\theta_t}$
  end while
  # Distributional Policy Improvement
  Compute policy gradient $\nabla_\theta \rho(Z_{\theta_t}^s)$ based on $\nabla_\theta \eta_{\theta_t}^s$.
  Update $\theta_{t+1} \leftarrow \theta_t - \delta \cdot \nabla_\theta \rho(Z_{\theta_t}^s)$.
end for

Finite-Time Convergence Guarantee

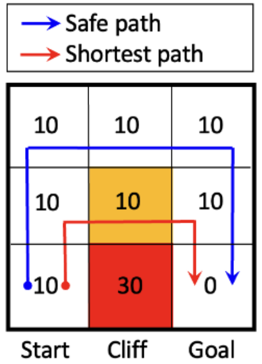
Theorem 4.11 (CDPG Convergence). *Suppose Assumption 4.10 holds. Let* $\epsilon_\alpha = \min\{\sum_{i=1}^{j} p_i^{N,\infty} - \alpha, \alpha - \sum_{i=1}^{j-1} p_i^{N,\infty}\}$. *In Algorithm 2, let the stepsize* $\delta = 1/\beta$ *and the number of* $\Pi_{\mathcal{C}}\mathcal{T}^\pi$ *oracle calls* $k(N, |\tau_\theta|) = \kappa N|\tau_\theta + 1|$. *For any* $\epsilon > 0$, *we have* $\min_{t=1,\dots,T} \|\nabla_\theta \rho(Z_{\theta_t,N})\|_2^2 \le \epsilon$, *whenever*

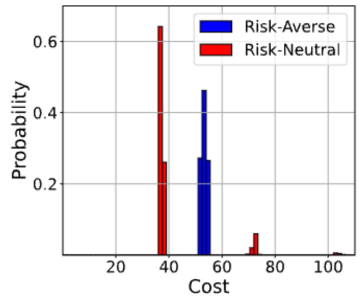
$$T \ge \frac{4\beta(\rho(Z_{\theta_1,N}) - \min_{\theta\in\Theta}\rho(Z_{\theta,N}))}{\epsilon} \quad \textit{and}$$

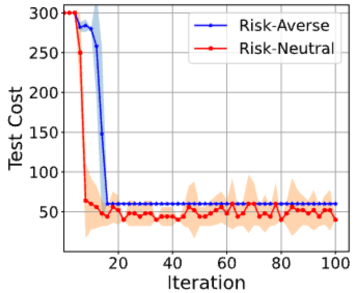
$$\kappa \ge \max\left\{ \mathcal{O}\left(\frac{\log(N^{1.5}\epsilon^{-0.5})}{N}\right), \mathcal{O}\left(\frac{\log(N\epsilon_\alpha^{-2})}{N}\right) \right\}.$$

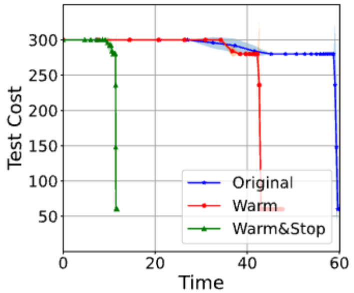
Numerical Experiments

Cliffwalking:

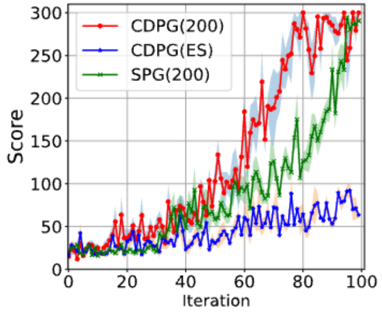


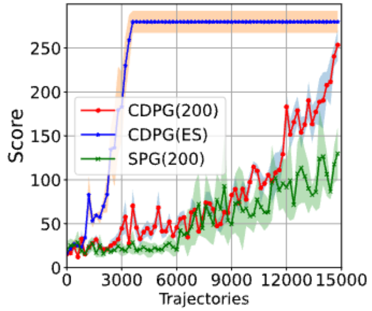






Cartpole:





Thrust 3: Physics-guided Distributional Reinforcement Learning

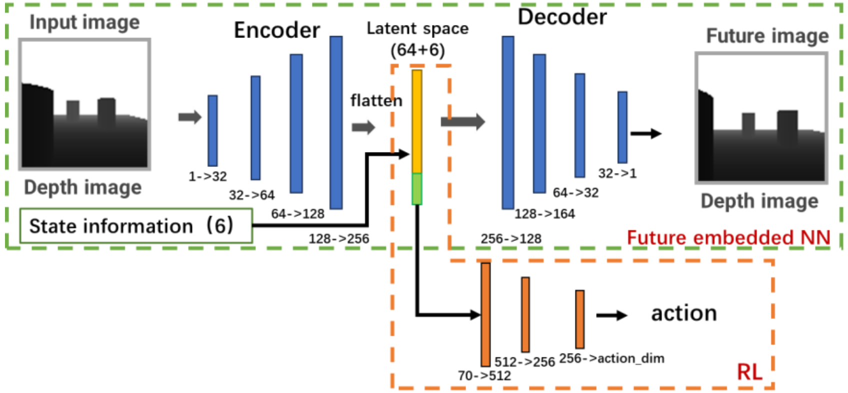
Yaoxin Shen, Rahul Rathnakumar, Yongming Liu**, Arizona State University

Arizona State University

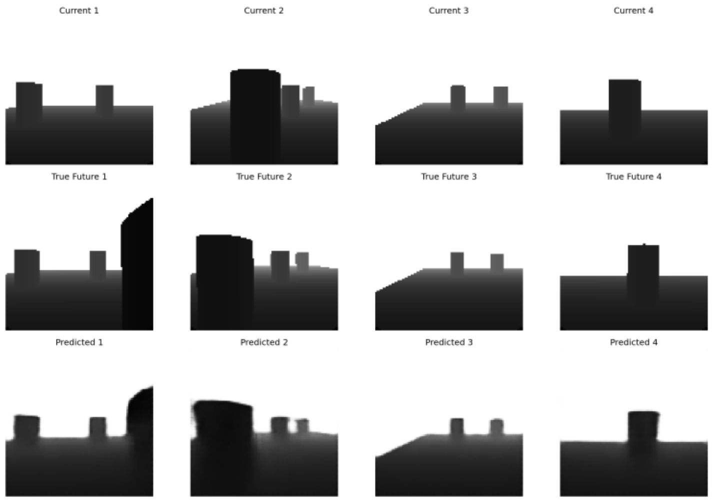
Background and Objective

This research aims to develop **physics-guided end-to-end learning** methods to tackle the challenge of lacking safety guarantees in reinforcement learning applications. This study seeks to improve policy optimization, exploration strategies, and safety by embedding physics knowledge into **input, output, and model structure**. A key contribution of this work is its ability to enhance predictive capabilities and risk awareness, particularly in unseen and uncertain scenarios, ensuring more robust and reliable decision-making in reinforcement learning systems.

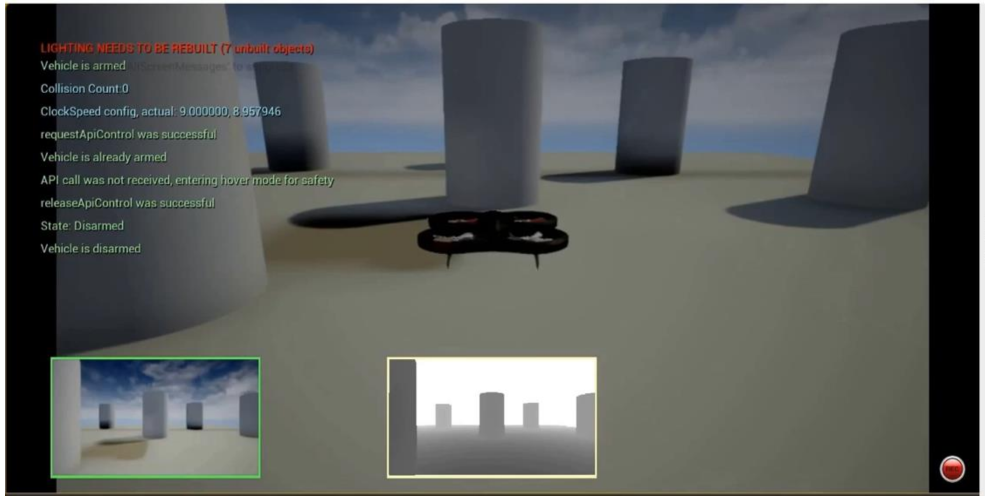
Approach 1: Future-Aware Embedded Neural Networks

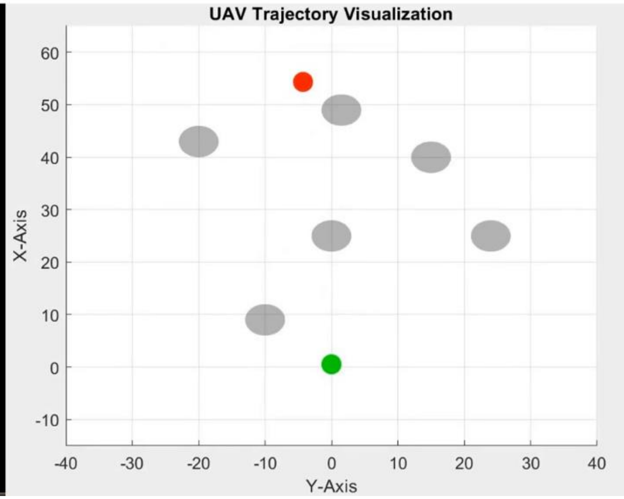


Results 1 :



With the current depth image and state as input and the future image as output, the network enhances future image reconstruction and embeds predictive information into the latent space.





Approach 2: Bayesian Entropy Neural Networks

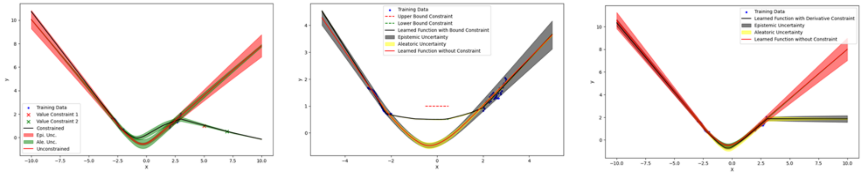
Bayesian Entropy – Constrained updating of probability distributions using data and expert knowledge.

The entropy is measured between the posterior p and prior q of the joint distribution for θ and x as:

$$S[p,q] = \iint p(x,\theta)\log\left(\frac{p(x,\theta)}{q(x,\theta)}\right) dx d\theta$$

Results 2 :

Simultaneously optimize for the Lagrange multipliers using backprop using the Differential Method of Multipliers:



2 value constraints: $x = 5$, $x = 7.5$ Bound constraints: $[-0.5, 0.5]$ Derivative constraint: $x = 5.0$

Thrust 4: Simulation and Experimental Testing

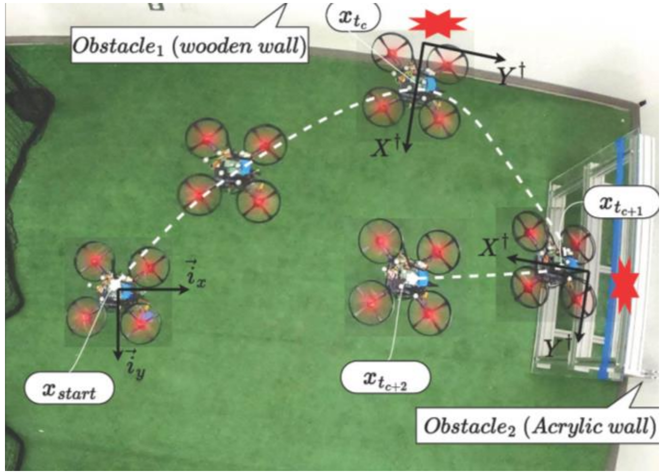
Valentin Gaucher, Yogesh Kumar, Amirali Abazari, Wenlong Zhang

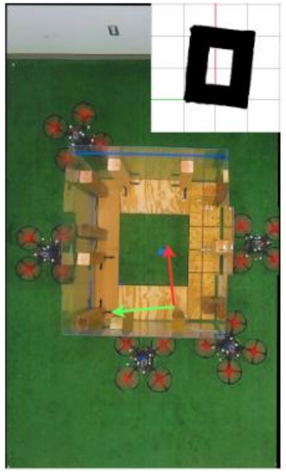
Development of Flexible UAV Platform

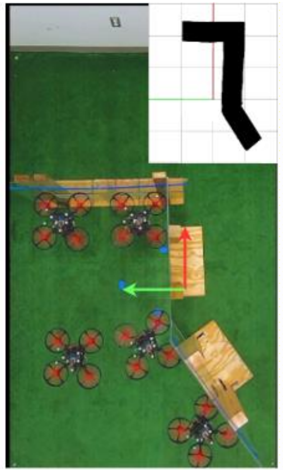
- Quadrotor UAV with passive foldable arms
- Wrench estimation to understand the environment
- Mechanical Intelligence: squeeze-and-fly, contact-based mapping and navigation, aggressive flights
- Next step: integration and testing with DRL algorithm

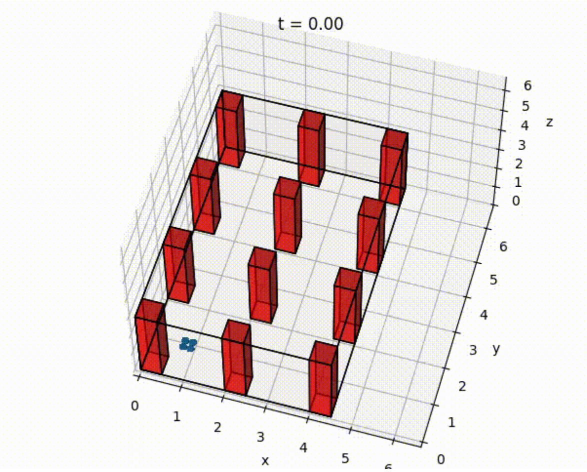
Exploration of UAV Simulators

- Surveyed and compared multiple UAV simulators
- Focused on identifying modular and open-source simulators ready for RL implementation
- Finalists: AirSim[1] and RotorPy[2]
- Next step: evaluation and integration of both simulators









[1] https://github.com/spencerfolk/rotorpy
[2] https://microsoft.github.io/AirSim/

Products

Y. Kumar et al., “Design, Contact Modeling, and Collision-inclusive Planning of a Dual-stiffness AerialRoboT (DART)”, 2025 ICRA, accepted.
A. Abazari, et al., “Dynamic Collision-Inclusive Modeling of a Multi-rotor Aerial Vehicle using Linear Complementarity Systems”, 2025 ACC, accepted.
K. Patnaik et al., “Tactile-based Exploration, Mapping and Navigation with Collision-Resilient Aerial Vehicles”, IEEE/ASME Transactions on Mechatronics (T-MECH), under review.

Thank you!

UNIVERSITY OF MICHIGAN

Arizona State University

THE OHIO STATE UNIVERSITY