PorphIt: PPO Robot Manipulator Morphology Design

Steve Gillet
University of Colorado Boulder Robotics
Email: steve.gillet@colorado.edu

Jay Vakil
Email: Jay.Vakil@colorado.edu

***Abstract*—This paper aims to address the cost and overactuation of robotics used in manufacturing by leveraging reinforcement learning (RL) to optimize robot morphologies to specific use cases. RL is the right tool for this job because it can optimize over the large search space of different robot morphologies, effectively exploring a large variety of iterations from all over that search space. The core methodology involves using a PPO model to iterate on the different morphologies and MuJoCo to evaluate their effectiveness in simulation. The evaluation signals will be successes, timing, and manipulation measures and these will be used to show the effectiveness of the PPO model at creating morphologies that are faster, more effective, and more efficient for particular tasks.**

***Index Terms*—reinforcement learning, robot morphology, manipulator design**

I. Introduction and Motivation

High costs, lack of tailorability, and lack of retrofitting are some of the most cited problems for the adoption of the robotics in industrial settings [1]. The most commonly used robotic manipulator in manufacturing is FANUC series of 6-DOF manipulators which cost from $17,500 to $400,000 depending on their capacity and customizations [2] [3]. Meanwhile [4] and [5] show that most manipulators are overactuated, less cost-efficient, and less effective for most tasks they are used in. We want to address this problem by creating an automation pipeline where you feed in a use case and you get out a robot specifically optimized for that use case. By matching the robot to the task we can ensure that robot can perform that task and only the parts needed for that task are used, reducing cost and increasing efficiency. Here we describe the process by which robot manipulator morphologies are optimized to particular tasks using a PPO agent.

We chose reinforcement learning because morphology design is an optimization problem over a complex, high-dimensional search space and RL can explore and learn optimal designs through trial-and-error in simulation [6]. We chose PPO because it is robust and on-policy and excels in continuous action spaces which is ideal for morphology design where there are continuous parameters like link lengths [7]. This is more effective than imitation learning because imitation learning can only use morphologies that already exist severely limiting sample space and innovation [8]. Supervised learning suffers from a very similar issue where labelled data would have to be provided which would be infeasible and there is no mechanism for exploration [9].

The core method is to use MuJoCo to use stable_baselines3 PPO implementation to iterate on the design parameters (number of links, joint types (X, Y, Z hinge, and Z actuation), and link lengths), use MuJoCo to simulate that iteration on a simple task using RRT* and numerical optimization IK for the path planning, feed the results (success, time, manipulability measure) back into the PPO model. The evaluation signals will be successes, time, and manipulability measure. Hopefully the better morphologies will be able to do the tasks more successfully, efficiently, and quicker and then we can vary the tasks slightly to find better morphologies for particular tasks. The tasks will be kept simple but varied in ways to test different types of movements like moving near the base, moving from near to far, different elevations, more linear movements. You can imagine that a lot of pick and place tasks are mostly linear and so might be largely done by linear actuators more efficiently.

The research questions we seek to answer: Can PPO be used to generate more efficient morphologies? What are the best evaluation signals to use? How does varying evaluation signals yield different results (optimizing for time yields simpler manipulators while optimizing for manipulability measure yields more complex manipulators?)?

II. Background and Related Work

Most of the research in this area codesigns morphology and control of modular, atomic robots like [10] which uses PPO, [11] which uses TD3, [12], and [13] which uses PPO and A3C for configuration and control of a reconfigurable, modular robot. More pertinent [14] uses DDQN to co-optimize morphology and control for modular robotic manipulators and [15] which uses soft actor critic for morphology and control of quadraped robots to avoid evalutating performance in sim or real to save time. Our method will focus on manipulators with simple controllers to allow us to focus on morphology design. The method will use Behavioral Cloning (BC) as a baseline and contrast PPO and Dreamer. PPO has been used effectively in some of the relevant research and allows for searching over the continuous space of manipulator parameters instead of the discrete space of different modules. Dreamer uses a 'world model' to predict outcomes and plan ahead which improves sample efficiency which might be even more effective by reducing computation needs [16].

III. Methods

The idea is to use PPO to iteratively optimize robot manipulator morphology parameters like number of joints, joint types (hinge X/Y/Z or slide in Z), and link lengths through simulation in MuJoCo. The agent will propose designs, evaluate them on tasks using IK and RRT* path planning, and update the policy based on rewards for task success, efficiency, and manipulability.

The algorithmic structure will follow a standard RL loop: At each episode the PPO agent samples a morphology configuration from its policy (parameterized as a Gaussian distribution over continuous link lengths and discrete joint types and numbers via Gumbel-Softmax for differentiability). The configuration will be instantiated in a dynamically generated MuJoCo XML model, simulated for a pick-and-place task (ie. moving from start to goal positions with varying elevations and distances), and controlled using numerical IK (using SciPy's minimize with L-BFGS-B) for joint angles and OMPL's RRT* for path planning in joint space. Rewards are then computed and the policy is updated.

The object is to maximize expected cumulative rewards, defined as $r = w_1 \cdot s + w_2 \cdot \frac{1}{t} - w_3 \cdot c + w_4 \cdot m$, where s is task success (1 if goal reached within tolerance, else 0), t is trajectory time, c is morphological complexity in number of joints, m is the manipulability measure (joint Jacobian determinant [17]), and w_i are the tunable weights. PPO minimizes the clipped surrogate loss:

$$\mathcal{L}(\theta) = \hat{\mathbb{E}}_t \left[\min\left(r_t(\theta)\hat{A}_t, \max(1-\epsilon, \min(1+\epsilon, r_t(\theta)))\hat{A}_t \right) \right] \\ + S[\pi_\theta](s_t) - \beta \cdot \mathbb{E}_t[(\hat{V}_\theta(s_t) - V_t^{\text{targ}})^2] \qquad (1)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$ is the probability ratio, $\hat{A}_t$ is the advantage estimated via Generalized Advantage Estimation (GAE) [18], $\epsilon = 0.2$ bounds the ratio for stability, S is an entropy bonus, and β balances the value loss [7]. For Dreamer the world model consists of a Recurrent State Space Model (RSSM) with encoder-decoder architecture to predict latent states and rewards, allowing imagination rollouts for policy updates [16].

Architectures use multi-layer perceptrons from Stable Baselines3. The policy network has 2 hidden layers of 64 units each with tanh activation functions for actor and critic. In Dreamer the world model adds a GRU recurrent layer (256 units) for dynamics prediction.

Update rules involve collecting trajectories over N episodes, computing advantages, and performing K epochs of minibatch SGD on the PPO loss with a clip ratio of 0.2. For Dreamer, updates alternate between model fitting and actor-critic optimization on imagined trajectories.

The baseline will involved BC trained on demonstrations from standard 6-DOF manipulators to imitate fixed morphologies to contrast with RL's exploration. As an ablation we will test removing the manipulability term from the reward function to assess the impact on design dexterity versus simplicity.

IV. Experimental Setup

V. Timeline and Milestones

VI. Risks and Mitigations

VII. Resources

VIII. Ethics and Safety

IX. Evaluation Plan

X. Expected Results

XI. Contributions and Roles

References

[1] McKinsey & Company, "Industrial robotics: Insights into the sector's future growth dynamics," McKinsey & Company, Tech. Rep., 2019. [Online]. Available: https://www.mckinsey.com/business-functions/operations/our-insights/industrial-robotics-insights-into-the-sectors-future-growth-dynamics

[2] Standard Bots, "Fanuc robot prices: Cost, models, and buying insights in 2025," 2025. [Online]. Available: https://standardbots.com/blog/fanuc-robot-price

[3] PatentPC, "Top robotics vendors by market share & installations," September 2025. [Online]. Available: https://patentpc.com/blog/top-robotics-vendors-by-market-share-installations

[4] M. Russo, L. Raimondi, X. Dong, and D. Axinte, "Task-oriented optimal dimensional synthesis of robotic manipulators with limited mobility," *Robotics and Computer-Integrated Manufacturing*, vol. 69, p. 102098, 2021.

[5] B. He, S. Wang, and Y. Liu, "Underactuated robotics: A review," *International Journal of Advanced Robotic Systems*, vol. 16, no. 4, pp. 1–14, 2019.

[6] R. S. Sutton, "Learning to predict by the methods of temporal differences," *Machine Learning*, vol. 3, no. 1, pp. 9–44, 1988.

[7] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[8] J. M. D. Delgado and L. Oyedele, "Robotics in construction: A critical review of the reinforcement learning and imitation learning paradigms," *Advanced Engineering Informatics*, vol. 54, p. 101787, 2022.

[9] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

[10] J. Bhatia *et al.*, "Reinforcement learning for freeform robot design," *arXiv preprint arXiv:2310.05670*, 2023.

[11] B. Tjanaka *et al.*, "Co-optimization of morphology and behavior of modular robots via hierarchical deep reinforcement learning," in *Robotics: Science and Systems (RSS)*, 2023.

[12] A. Spielberg *et al.*, "Accelerated co-design of robots through morphological pretraining," *arXiv preprint arXiv:2502.10862*, 2025.

[13] M. Kalimuthu, A. A. Hayat, T. Pathmakumar, M. R. Elara, and K. L. Wood, "A deep reinforcement learning approach to optimal morphologies generation in reconfigurable tiling robots," *Mathematics*, vol. 11, no. 18, p. 3893, 2023.

[14] Z. Ding, H. Tang, H. Wan, C. Zhang, and R. Sun, "A modular robotic arm configuration design method based on double dqn with prioritized experience replay," *Symmetry*, vol. 16, no. 6, p. 714, 2024. [Online]. Available: https://www.mdpi.com/2073-8994/16/6/714

[15] K. S. Luck, H. B. Amor, R. Calandra, and J. Peters, "Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2019.

[16] D. Hafner, T. Lillicrap, I. Fischer, R. Villegas, D. Ha, H. Lee, and J. Davidson, "Dream to control: Learning behaviors by latent imagination," *arXiv preprint arXiv:1912.01603*, 2019.

[17] T. Yoshikawa, "Manipulability of robotic mechanisms," *The International Journal of Robotics Research*, vol. 4, no. 2, pp. 3–9, 1985.

[18] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.