Capturing the Hybrid Dynamics of Planar Pushing in RL with Multimodal Categorical Exploration

Juan Del Aguila Ferrandis[1], João Moura[1,2], and Sethu Vijayakumar[1,2]

***Abstract*— Planar pushing is a hybrid dynamics system due to the different possible contact interaction modes between the robot and the object, such as sticking, sliding, and separation. Previous Reinforcement Learning (RL) literature addressing the planar pushing task achieves low accuracy, non-smooth trajectories, and only simple motions, i.e. without orientation of the manipulated object. We conjecture that previously used unimodal exploration strategies fail to capture the inherent hybrid dynamics of the task. In this paper, we incorporate the hybrid dynamics into an RL framework by proposing a multimodal exploration approach through categorical distributions, which enables us to train planar pushing RL policies for arbitrary initial and target object poses, i.e. positions and orientations, and with improved accuracy. We show that the learned policies are robust to external disturbances, scalable to tasks with multiple pushers, and exhibit smooth pushing trajectories. Furthermore, we validate the transferability of the policies, trained entirely in simulation, to a physical robot hardware using the KUKA iiwa robot arm. See our supplemental video: https://youtu.be/vTdva1mgrk4.**

I. Introduction and Related Work

Nonprehensile manipulation, defined as manipulation without grasping, endows robots with versatile behaviors, enabling them to perform a wide range of motions on objects with different properties [1], [2]. However, allowing the pose of the object relative to the end-effector to change requires the robot to constantly adapt the contact positions, leading to different possible contact modes in the form of sticking, sliding, and separation. As a result, multiple interesting challenges arise. Most notably, the underactuated nature of the system makes it infeasible to realize arbitrary motions of the object [3], in addition to the complexity of hybrid dynamics resulting from the transitions between different contact modes [3], and the hard to model frictional interactions exacerbating the uncertainty in the contact modes and the object motion [4], [5].

In this paper we consider the task of planar pushing, widely studied in the nonprehensile literature [1], [3], [5]–[7]. The task consists of using a robotic pusher to control the motion of an object sliding on a flat surface. Previous works developed robot controllers for planar pushing, generally following one of two approaches: model-based via Model Predictive Control (MPC) [3], [7], or model-free via Reinforcement Learning (RL) [8]–[11].

These approaches typically face different open problems. MPC lacks scalability to more complex scenarios, such as multiple contacts and switching contact faces [3], [12]. On the other hand, RL methods achieve low accuracy, with position errors greater than $2\,\mathrm{cm}$, non-smooth trajectories, and only simple motions, i.e. without orientation of the sliding object [8]–[11], which we aim to consider. These RL methods share a common trait: they use a multivariate Gaussian with diagonal covariance for exploration, thereby limiting the exploration to unimodal policies across each action space dimension. However, the model-based literature [3], [7] identifies the planar pushing problem as a hybrid dynamics system due to the different possible contact modes (sticking, sliding left, sliding right, and separation). This provides us with the insight that perhaps planar pushing is fundamentally a multimodal control problem, which motivates our proposed multimodal exploration approach through categorical distributions.

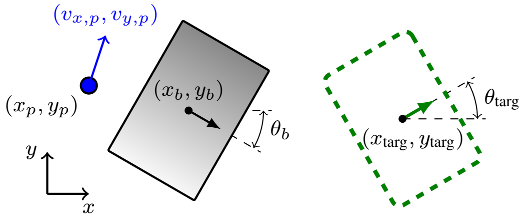

Fig. 1. Illustration of the planar pushing system.

II. Background

A. Planar Pushing

We consider the task of pushing a box to a specified target pose, composed of the box position and orientation, from a random initial system configuration, composed of the initial box pose and robot pusher position, all within a bounded planar workspace. Fig. 1 illustrates the planar pushing system, where $(v_{x,p}, v_{y,p})$ is the velocity of the pusher, located at (x_p, y_p), (x_b, y_b, θ_b) is the pose of the box, and $(x_{\text{targ}}, y_{\text{targ}}, \theta_{\text{targ}})$ is the target box pose.

B. Problem Formulation

We formulate the problem as a finite horizon goal-conditioned Partially Observable Markov Decision Process (POMDP) defined by the tuple $(\mathcal{S}, \Omega, \mathcal{G}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, H, \rho_0, \rho_g)$ [10], [13]. At each time step t, the environment has state $s_t \in \mathcal{S}$, we receive an observation $o_t \in \Omega$, our goal is $g_t \in \mathcal{G}$, and we take an action $a_t \in \mathcal{A}$. Note that the goal remains fixed during an episode. Additionally, $\mathcal{O} : \mathcal{S} \times \mathcal{A} \to \Pr(\Omega)$ is the observation model, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Pr(\mathcal{S})$ is the transition dynamics, and $\mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \to \mathbb{R}$ is the reward function. We limit

[1] School of Informatics, The University of Edinburgh, Edinburgh, UK. [2] The Alan Turing Institute, London, U.K. This research is supported by the EU H2020 projects Enhancing Healthcare with Assistive Robotic Mobile Manipulation (HARMONY, 101017008) and the Kawada Robotics Corporation.

episodes to have a maximum horizon H. Finally, the initial state and goal of an episode are distributed according to ρ_0 and ρ_g respectively.

C. Proximal Policy Optimization

We wish to learn a stochastic policy $\pi_\theta : \Omega \times \mathcal{G} \to \Pr(\mathcal{A})$ parametrized by θ. To this end, we use Proximal Policy Optimization (PPO) [14], a popular on-policy RL algorithm widely applied in various control tasks, including in-hand manipulation [15] and locomotion [16]. PPO uses a truncated Generalized Advantage Estimation (GAE) [17] to estimate the advantage function for time step $t \in [0, T]$ as

$$\hat{A}_t = \sum_{i=0}^{T-t} (\gamma\lambda)^i \delta_{t+i}, \tag{1}$$

where

$$\delta_t = r_t + \gamma V_\phi(o_{t+1}, g_{t+1}) - V_\phi(o_t, g_t), \tag{2}$$

r_t is the reward, $V_\phi : \Omega \times \mathcal{G} \to \mathbb{R}$ is the value function, parametrized by ϕ, λ is the GAE parameter, and γ is the discount factor. Then, π_θ and V_ϕ can be learned together through mini-batch stochastic gradient ascent on the objective function

$$\begin{aligned} L_t(\theta, \phi) = \hat{\mathbb{E}}_t \Bigg[\min \Bigg(& \frac{\pi_\theta(a_t \mid o_t, g_t)}{\pi_{\theta_{old}}(a_t \mid o_t, g_t)} \cdot \hat{A}_t, \\ & \text{clip} \left(\frac{\pi_\theta(a_t \mid o_t, g_t)}{\pi_{\theta_{old}}(a_t \mid o_t, g_t)},\ 1-\epsilon,\ 1+\epsilon \right) \hat{A}_t \Bigg) \\ & - c_1 L_t^{V_\phi} + c_2 S_t^{\pi_\theta} \Bigg], \end{aligned} \tag{3}$$

where we compute the expectation over a mini-batch of samples. The first term within the expectation is the surrogate objective of the policy, $L_t^{V_\phi}$ is the loss of the value function, $S_t^{\pi_\theta}$ is an entropy bonus, c_1, c_2 are weights, and ϵ controls the clip range [14].

III. Method

A. Observation, Goal, Action, and Reward

Observation. The environment observation o_t consists of the current box pose (x_b, y_b, θ_b) and the current pusher position (x_p, y_p). There is important information from the environment state that this observation fails to capture, for instance, the frictional contact forces and the box velocity. We consider two architectures of the policy and value networks to attempt to capture this hidden information: an MLP with observation stacking [8], [18] and an LSTM [8], [15].

Goal and Action. The goal g_t of the policy is to reach a particular target box pose $(x_{\text{targ}}, y_{\text{targ}}, \theta_{\text{targ}})$. Given a goal g_t and an observation o_t, the policy takes an action $a_t = (v_{x,p}, v_{y,p})$, which consists of the x and y velocity of the pusher. We limit the velocity on each axis to the range $[-0.1, 0.1]\ \mathrm{m\,s^{-1}}$.

Reward. If the box reaches the target, the episode terminates successfully with a positive reward $r_t = \alpha$. If the box fails to reach the target within the maximum horizon, or the workspace boundaries are violated by the pusher or box, the episode terminates unsuccessfully with a negative reward $r_t = -\beta$. Otherwise, the reward is $r_t = k_1(1 - d_{x,y}) + k_2(1 - d_\theta) + k_3(1 - v_p)$, where $d_{x,y}$ is the normalized distance to the target position, d_θ is the normalized angular distance to the target orientation, v_p is the normalized magnitude of the pusher velocity, and k_1, k_2, k_3 are weights.

TABLE I. Dynamics Randomization and Observation Noise Parameters

Parameter	Sampling Distribution
Friction	$\mathcal{U}([0.5, 0.7])$
Restitution	$\mathcal{U}([0.4, 0.6])$
Box Length	$\mathcal{U}([0.115, 0.125])$ m
Box Width	$\mathcal{U}([0.095, 0.105])$ m
Box Mass	$\mathcal{U}([0.4, 0.6])$ kg
Pusher Radius	$\mathcal{U}([0.012, 0.013])$ m
Time Step Duration	$\mathcal{N}(1/30, (1/320)^2)$ s
Position Noise	$\mathcal{N}(0, 0.001^2)$ m
Orientation Noise	$\mathcal{N}(0, 0.02^2)$ rad

B. Exploration Strategies

Gaussian Exploration. Previous RL methods for planar pushing have used Gaussian exploration. In PPO, given an observation and goal pair, the policy function outputs the mean velocities in x and y, denoted as μ_x and μ_y. Combining them with the corresponding learned state-independent variances σ_x^2 and σ_y^2 results in a multivariate Gaussian with diagonal covariance from which we can sample the action [14]. Soft Actor Critic (SAC) [19] is a popular off-policy RL algorithm. We include SAC with Gaussian exploration and an MLP architecture as a baseline in our experiments since the current state-of-the-art RL policies for planar pushing use the same configuration [10]. In SAC, the policy function outputs μ_x, μ_y, σ_x^2, and σ_y^2 [19].

Categorical Exploration. To enable multimodal exploration, capable of capturing the hybrid dynamics of the task, we propose discretizing the action space and using categorical distributions for exploration, which can approximate any type of distribution. In particular, we discretize $v_{x,p}$ and $v_{y,p}$ using 11 bins for each velocity [15], [20]. Then, given an observation and goal pair, the policy function outputs 11 logits that define a categorical distribution over $v_{x,p}$ and another 11 logits that define a categorical distribution over $v_{y,p}$. We sample the action from these distributions.

C. Sim-to-Real Transfer

We train the policies entirely in simulation and use dynamics randomization, observation noise, and synthetic disturbances to bridge the sim-to-real gap. At the start of every episode, we sample random values for: (a) the friction and restitution of the floor, box, and pusher; (b) the dimensions of the box and the pusher; and (c) the mass of the box. Additionally, we randomize the duration of every time step [8]. We also add correlated noise, sampled at the beginning of each episode, and uncorrelated noise, sampled at every time step, to the observations of the box pose and pusher position, to simulate sensor uncertainty. Table I details the parameters and sampling distributions for the dynamics randomization and observation noise. Finally, we apply a disturbance to the box with probability 1% at each time step, in a uniformly random position, and with force in x and y independently sampled from $\mathcal{U}([-25, 25])$ N.

D. Curriculum Learning

We define success thresholds $T_{x,y}$ and T_θ, corresponding to the position and the orientation, such that, if $\|(x_b, y_b) - (x_{\text{targ}}, y_{\text{targ}})\| \leq T_{x,y}$ and $|\theta_b - \theta_{\text{targ}}| \leq T_\theta$, then the episode terminates successfully. Smaller $T_{x,y}$ and T_θ lead to more accurate learned policies, however, at the expense of increased task complexity and a sparser reward signal, which can lead to much slower learning, or lack of convergence entirely. To mitigate this issue, we define a curriculum such that the learning starts with larger thresholds $T_{x,y} = 1.5\,\text{cm}$ and $T_\theta = 0.34\,\text{rad} \approx 19.5°$. Then, if the policy exceeds a 90% average success rate, we halve the success thresholds to $T_{x,y} = 0.75\,\text{cm}$ and $T_\theta = 0.17\,\text{rad} \approx 9.7°$.

E. Implementation Details

We train the policies with data collected from 128 parallel actors in simulated planar pushing systems. The simulations are performed using PyBullet [21]. Additionally, the maximum episode length is $H = 300$ time steps. For the reward function, we use parameter values $\alpha = 50$, $\beta = 20$, $k_1 = 0.1$, $k_2 = 0.02$, and $k_3 = 0.004$. Our implementations of PPO and SAC are based on Stable Baselines3 [22]. We design a custom planar pushing environment for learning. At the beginning of every episode, we uniformly sample the starting configuration and the target box pose. All policies are trained in a single workstation with an Intel Core i9 3.60GHz, GeForce RTX 2080, and 64 GiB of RAM.

The MLP architecture uses a stack of 10 previous observations as well as 2 hidden linear layers for the policy and value networks of size (512, 512) and (1024, 1024) respectively. The LSTM architecture uses 3 hidden layers arranged as linear (128) $\rightarrow$ LSTM (256) $\rightarrow$ linear (128) for both the policy and value networks. We use $\tanh$ nonlinearities with PPO [14] and ReLu nonlinearities with SAC [19]. Additionally, PPO learns a state value function [14] while SAC learns two state-action value functions [19].

We use the Adam [23] optimizer with a learning rate of $3 \cdot 10^{-4}$ and a batch size of 7680 for PPO and SAC. In PPO, we use value function and entropy bonus coefficients $c_1 = 0.5, c_2 = 0$, as well as early stopping of model updates when the KL divergence of the new policy and the old policy exceeds 0.01 [22]. In SAC, we use a replay buffer of size 10^6 and apply a $\tanh$ squashing function to the sampled actions [10], [19]. The remaining hyperparameters of PPO and SAC have standard values from [14] and [19] respectively.

We evaluate the scalability of our framework on a planar pushing task with two pushers. To encourage the policy to perform motions that are feasible for a bi-manual manipulation platform, we add two constraints: (a) each pusher can exert pushing forces with a maximum magnitude of $75\,\text{N}$; and (b) the distance between pushers in the x coordinate must be at least $5\,\text{cm}$. The episode terminates unsuccessfully if the policy violates any of these constraints.

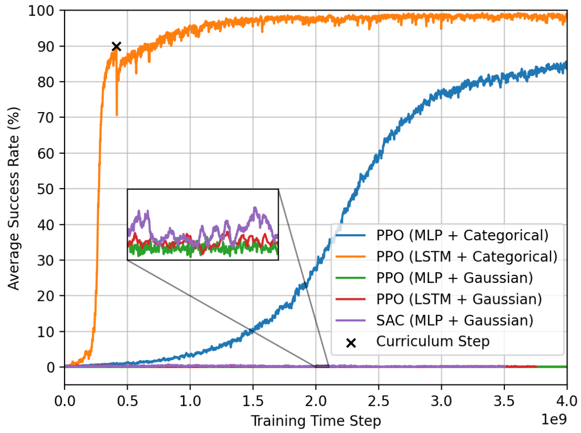

Fig. 2. Policy training performance. Success rate is averaged over the last 100 episodes completed by each of the 128 parallel actors.

IV. Experiments and Results

A. Simulation

We first consider the standard set-up with one pusher. We train PPO policies with the MLP and LSTM architectures, and compare the categorical and Gaussian exploration strategies for each configuration. We also train a SAC policy with the MLP architecture and Gaussian exploration. The resulting learning curves are shown in Fig 2. We find that only the policies using the proposed categorical exploration approach manage to learn the task. Additionally, the LSTM architecture provides substantially faster convergence. The PPO (LSTM + Categorical) policy achieves over 98% average success rate with the reduced success thresholds.

We further investigate whether exploration through categorical distributions indeed leads to multimodal strategies. We examine the evolution during training of the categorical distribution in PPO (LSTM + Categorical) for the action $v_{y,p}$ in various environment states. As expected, we often find that the action distribution is multimodal. Fig. 4 shows the results for one of these cases. In particular, it broadly has two modes that correspond to upward and downward motions. Therefore, it seems that the categorical exploration strategy enables the policy to explore different possible contact modes concurrently during training.

To evaluate the scalability of our framework, we train the PPO (LSTM + Categorical) policy on a planar pushing task with two pushers. The resulting learning curve is shown in Fig. 5. The policy scales well to this more complex task and achieves an average success rate greater than 97% with the reduced success thresholds. Nevertheless, convergence is slower, which is expected due to the increased dimensionality of the problem. Some additional simulation experiments and results can be found in [24].

B. Hardware

We investigate the performance of the PPO (LSTM + Categorical) policy on a physical planar pushing set-up with the KUKA iiwa robot. We use the Vicon motion capture system to track the current and target box pose, and use OpTaS [25] to map policy actions in the end-effector task-space to robot

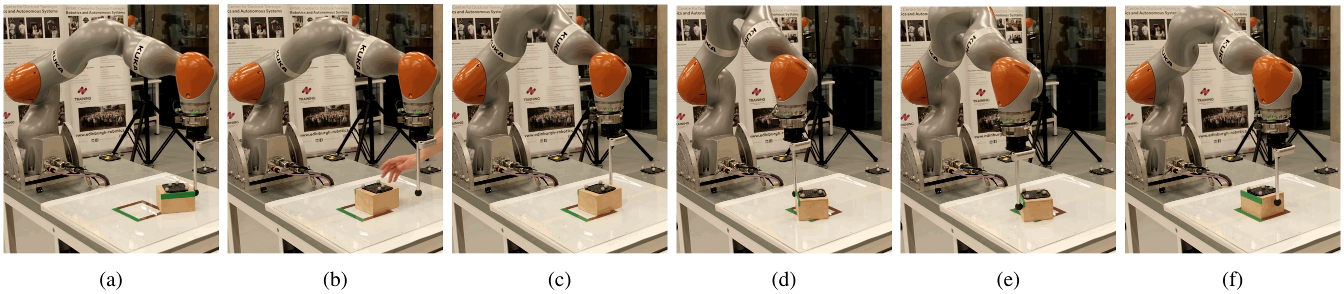
(a) (b) (c) (d) (e) (f)

Fig. 3. Key frames of the KUKA iiwa robot pushing the box to a target pose. (a) Shows the starting configuration, a large disturbance is applied in (b), and (c)-(f) exhibit the RL policy recovering from the disturbance and reaching the goal.

joint configurations. We use the ROS-Pybullet interface [26] to develop and test the robot software implementation.

Fig. 3 shows a sequence of key frames of the robot pushing the box to a target pose and recovering from an external disturbance. The supplemental video (https://youtu.be/vTdva1mgrk4) clearly demonstrates the behavior of the policy in simulation and on the physical robot. We find that the policy translates well to the real world and is able to effectively cope with the dynamics of the new environment. It is robust to large external disturbances as well as changes in the initial and target pose, and achieves both accurate and smooth pushing trajectories.

We evaluate the success rate and time to target on the physical robot, using success thresholds $T_{x,y} = 0.75\,\mathrm{cm}$, $T_\theta = 0.17\,\mathrm{rad}$, and a time limit of $30\,\mathrm{s}$. We uniformly sample 5 target poses and, for each target pose, we run the policy from 15 uniformly sampled initial configurations. The average success rate and time to target are 97.3% and $6.5\,\mathrm{s}$. In simulation with the same success thresholds and time limit, the average success rate and time to target are 99.2% and $5.0\,\mathrm{s}$. The policy exhibits similar performance in simulation and in the physical robot, indicating good sim-to-real transfer.

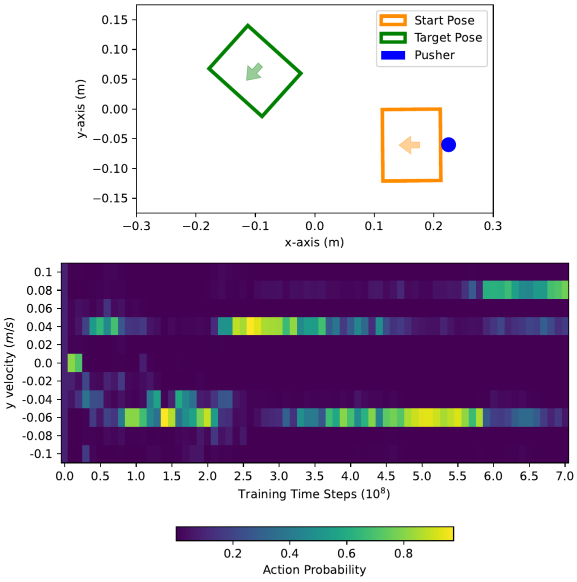

Fig. 4. Evolution during training of the categorical action distribution for the pusher velocity in the y axis ($v_{y,p}$) for the configuration shown above.

V. Summary and Discussion

In this paper, we incorporate hybrid dynamics into an RL framework by proposing a multimodal exploration approach, through categorical distributions on a discrete action space, to enable the learning of planar pushing RL policies for arbitrary initial and target object poses, i.e. different positions and orientations, with improved accuracy. Our experiments demonstrate that the policies, trained only in simulation, successfully recover from external disturbances and achieve both smooth trajectories and small target error when executed on the physical robotic hardware. Furthermore, we show that our framework can be easily scaled to a planar pushing task with two pushers.

One of the key realizations in this work was that, when attempting to learn planar pushing RL policies for the case of arbitrary object poses, the use of a multivariate Gaussian with diagonal covariance for exploration as per previous literature [8]–[11], would lead to the RL failing to converge. Borrowing the insight from the model-based literature [3], [7], that planar pushing has hybrid dynamics reflected in a set of different contact modes that constraint the control actions, we hypothesized that we can reason about planar pushing as a multimodal control problem. Therefore, we proposed describing the action space through categorical distributions to capture the multimodal nature of the problem, potentially leading to more effective exploration of different contact modes during training. We have also shown that indeed, during training, the categorical action distributions exhibit multimodal exploration strategies.

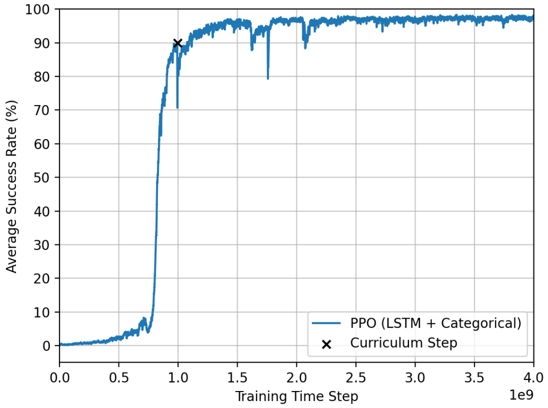

Fig. 5. Training performance on a planar pushing task with two pushers.

References

[1] M. T. Mason, "Mechanics and planning of manipulator pushing operations," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 53–71, 1986. DOI: 10.1177/027836498600500303.

[2] M. T. Mason, "Progress in nonprehensile manipulation," *The International Journal of Robotics Research*, vol. 18, no. 11, pp. 1129–1141, 1999. DOI: 10.1177/02783649922067762.

[3] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *The International Journal of Robotics Research*, vol. 39, no. 7, pp. 755–773, 2020. DOI: 10.1177/0278364920913938.

[4] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason, "A probabilistic planning framework for planar grasping under uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2111–2118, 2017. DOI: 10.1109/LRA.2017.2720845.

[5] M. Bauza and A. Rodriguez, "A probabilistic data-driven model for planar pushing," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 3008–3015. DOI: 10.1109/ICRA.2017.7989345.

[6] S. Goyal, A. Ruina, and J. Papadopoulos, "Limit surface and moment function descriptions of planar sliding," in *1989 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 2, 1989, pp. 794–799. DOI: 10.1109/ROBOT.1989.100081.

[7] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 970–976. DOI: 10.1109/ICRA46639.2022.9811942.

[8] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel, "Sim-to-real transfer of robotic control with dynamics randomization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3803–3810. DOI: 10.1109/ICRA.2018.8460528.

[9] K. Lowrey, S. Kolev, J. Dao, A. Rajeswaran, and E. Todorov, "Reinforcement learning for non-prehensile manipulation: Transfer from simulation to physical system," in *2018 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR)*, 2018, pp. 35–42. DOI: 10.1109/SIMPAR.2018.8376268.

[10] L. Cong, H. Liang, P. Ruppel, *et al.*, "Reinforcement learning with vision-proprioception model for robot planar pushing," *Frontiers in Neurorobotics*, vol. 16, 2022, ISSN: 1662-5218. DOI: 10.3389/fnbot.2022.829437.

[11] R. Jeong, J. Kay, F. Romano, *et al.*, "Modelling generalized forces with reinforcement learning for sim-to-real transfer," *arXiv preprint arXiv:1910.09471*, 2019.

[12] T. Xue, H. Girgin, T. S. Lembono, and S. Calinon, "Demonstration-guided optimal control for long-term non-prehensile planar manipulation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 4999–5005. DOI: 10.1109/ICRA48891.2023.10161496.

[13] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, vol. 7, no. 62, eabk2822, 2022.

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

[15] OpenAI, M. Andrychowicz, B. Baker, *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[16] N. Heess, D. TB, S. Sriram, *et al.*, "Emergence of locomotion behaviours in rich environments," *arXiv preprint arXiv:1707.02286*, 2017.

[17] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[18] V. Mnih, K. Kavukcuoglu, D. Silver, *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[19] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, PMLR, 2018, pp. 1861–1870.

[20] Y. Tang and S. Agrawal, "Discretizing continuous action space for on-policy optimization," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 4, pp. 5981–5988, 2020. DOI: 10.1609/aaai.v34i04.6059.

[21] E. Coumans and Y. Bai, *Pybullet, a python module for physics simulation for games, robotics and machine learning*, http://pybullet.org, 2016–2021.

[22] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research*, vol. 22, no. 268, pp. 1–8, 2021.

[23] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[24] J. Del Aguila Ferrandis, J. Moura, and S. Vijayakumar, "Nonprehensile planar manipulation through reinforcement learning with multimodal categorical exploration," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023.

[25] C. E. Mower, J. Moura, N. Z. Behabadi, S. Vijayakumar, T. Vercauteren, and C. Bergeles, "Optas: An optimization-based task specification library for trajectory optimization and model predictive control," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9118–9124. DOI: 10.1109/ICRA48891.2023.10161272.

[26] C. Mower, T. Stouraitis, J. Moura, *et al.*, "Ros-pybullet interface: A framework for reliable contact simulation and human-robot interaction," in *Proceedings of The 6th Conference on Robot Learning*, PMLR, 2023, pp. 1411–1423.