Whole-Body End-Effector Pose Tracking

Tifanny Portela[1,2], Andrei Cramariuc[1], Mayank Mittal[1,3] and Marco Hutter[1]

***Abstract*— Combining manipulation with the mobility of legged robots is essential for a wide range of robotic applications. However, integrating an arm with a mobile base significantly increases the system's complexity, making precise end-effector control challenging. Existing model-based approaches are often constrained by their modeling assumptions, leading to limited robustness. Meanwhile, recent Reinforcement Learning (RL) implementations restrict the arm's workspace to be in front of the robot or track only the position to obtain decent tracking accuracy. In this work, we address these limitations by introducing a whole-body RL formulation for end-effector pose tracking in a large workspace on rough, unstructured terrains. Our proposed method involves a terrain-aware sampling strategy for the robot's initial configuration and end-effector pose commands, as well as a game-based curriculum to extend the robot's operating range. We validate our approach on the ANYmal quadrupedal robot with a six DoF robotic arm. Through our experiments, we show that the learned controller achieves precise command tracking over a large workspace and adapts across varying terrains such as stairs and slopes. On deployment, it achieves a pose-tracking error of 2.64 cm and $3.64°$, outperforming existing competitive baselines. The video of our work is available at: `wholebody-pose-tracking`.**

I. INTRODUCTION

Over the past decade, algorithmic advancements have substantially increased legged robots' ability to traverse complex, cluttered environments and human-designed infrastructures, such as stairs and slopes [1]–[3]. Despite these improvements, their practical applicability remains constrained by their limited manipulation capabilities. Most field operations with legged robots involve minimal environmental interactions, such as visual inspections and passive load transportation. Thus, combining a legged robot's mobility with the ability to perform manipulation tasks is critical for enhancing their applications to more real-world scenarios.

Compared to fixed base counterparts, integrating an arm onto a legged mobile platform significantly complicates the controller design because of increased degrees of freedom, redundancy and highly non-linear dynamics. To address this, the research community has mainly explored model-based and learning-based control strategies.

Model-based approaches, such as Model Predictive Control (MPC), have shown precise control on flat terrain by leveraging accurate models of the robot and its environment [4], [5]. However, solving MPC's control problem in real-time for legged manipulators often requires the use of simplified models [6], [7], which increases vulnerability to unexpected disturbances such as slipping or unplanned contacts.

Fig. 1: Our whole-body controller demonstrates precise end-effector pose tracking across a variety of challenging terrains, including soft mattresses, stairs and uneven natural ground.

In contrast, the learning-based control strategy of Reinforcement Learning (RL) has emerged as a robust alternative, directly learning control policies through interactions in simulation for locomotion [1], [3], [8] and whole-body control [9]. RL enables improved resilience to external disturbances compared to MPC because of environmental variability during training [1], [10], [11]. Research on legged robots, whether using legs [12], [13] or attached arms [9], [14], demonstrate that RL techniques can achieve effective end-effector position tracking across a large workspace through agile whole-body behavior. Although effective in outdoor and slippery terrains, these approaches remain unsuitable beyond flat terrain and lack orientation tracking.

To address these shortcomings, we propose a general-purpose whole-body controller for legged robots with an attached arm. The controller is designed to provide stable end-effector pose tracking across a large operational space. Our approach includes a terrain-aware sampling strategy for end-effector pose commands, and for the robot's initial configuration to ensure a smooth transition from a locomotion policy to the proposed whole-body controller. Experimental results show that the controller achieves precise tracking across varying terrains, such as stairs and slopes. Our key

Authors are members of [1]Robotic Systems Lab, ETH Zurich. [2]ETH AI center. [3]NVIDIA. Email: `tifanny.portela@ai.ethz.ch`

This project has received funding from the ETH AI Center and the European Union's Horizon Europe Framework Programme under grant agreement No 101121321 and NCCR automation. This work has been conducted as part of ANYmal Research, a community to advance legged robotics.

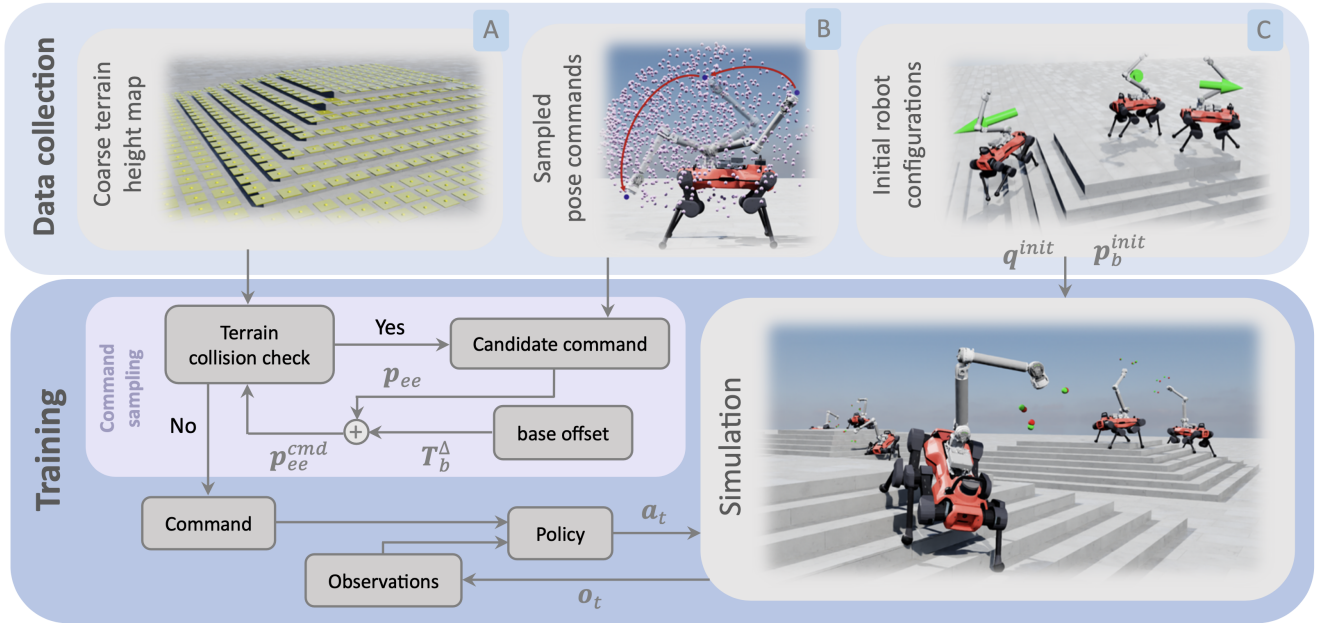

Fig. 2: The training process begins with data collection, where we gather (A) the terrain mesh and a coarse terrain height map, (B) 10000 pre-sampled end-effector pose commands with a fixed base, and (C) base poses and joint angles from a pre-trained locomotion policy to initialize robots. During training, a command from (B) is slightly transformed and checked for collisions with the terrain. If collision-free, it is concatenated with observations and input to the policy; otherwise, a new command is sampled. The policy is trained in simulation with 4000 robots in parallel, outputting joint actions.

contributions are as follows:

- We propose an RL whole-body controller for 6-DoF end-effector pose tracking for quadrupeds with an arm.
- We showcase the learned controller's tracking capabilities over challenging terrains and its robustness when faced with external disturbances.
- We compare our learned controller to model-based controllers and existing RL approaches, showing higher tracking accuracy and enlarged pose reachability, both in simulation and on hardware, reaching a pose tracking accuracy of 2.64 cm and $3.64°$.

II. METHOD

We train a policy to track end-effector target poses with minimal foot displacement, intended for use alongside a locomotion policy. Figure 2 illustrates the overall training process. We use Isaac Lab [15] as a simulation environment to train our policy and deploy our controller on ALMA [4], which integrates the Anymal D robot from ANYbotics [16] with the Dynaarm from Duatic [17].

A. Policy Architecture

We use Proximal Policy Optimization (PPO) [18], where both the actor and critic networks are implemented as multi-layer perceptrons with hidden layers of size [512, 256, 128], with ELU as the activation function. The hyperparameters of the PPO algorithm are taken from prior work [19].

B. Command sampling

We pre-sample end-effector pose commands for a fixed base by iterating over the six joint angles of the arm, covering their entire range, and recording the end-effector poses that are collision-free in the base frame, as illustrated in Figure 2-B. If these poses were used directly as commands, a simple inverse kinematics solver for the arm would suffice. However, this command sampling strategy would limit the controller's ability to achieve certain end-effector poses that could otherwise be reached by utilizing the entire body of the mobile manipulator. To overcome this limitation, we introduce a random body pose transformation $T_b^{\Delta} \in \mathbb{R}^6$, applied to each pre-sampled command when a new pose is defined. This transformation is sampled within ranges of $[-0.2, 0.2]$ m for the x and y dimensions, $[-0.3, 0.1]$ m for z and, $[-\pi/6, \pi/6]$ rad for roll, pitch and yaw. This approach ensures that the end-effector pose commands are reachable with minimal base movement. Finally, commands falling below the terrain surface are resampled. To speed up training, a coarse terrain height map (Figure 2-A) stores the highest terrain points in 20×20 cm patches.

C. Command definition

The command of the policy is an end-effector pose $p_{ee} \in \mathbb{SE}(3)$, typically represented by a separate position and orientation [9], [20]. Separating these components introduces two main challenges. First, defining a rotation representation that is easily learnable is difficult [21]. Second, this separation requires a fixed trade-off between position and orientation rewards, which may not be optimal for all workspace poses.

To avoid these issues, we use a keypoint-based representation similar to that used in [22] for in-hand cube reorientation, which has been shown to improve the ability of the RL algorithm to learn the task at hand. In this formulation, the

keypoints represent the vertices of a cube centered on the end-effector's position and aligned with its orientation. While 8 corner points fully define the cube, we use the minimum required of 3 keypoints with direct correspondence between measured and target poses to uniquely and completely define the pose. The side length of the cube is set to 0.3 meters.

D. Action and Observation Space

The robot's movement is managed through an eighteen-dimensional space ($a^t \in \mathbb{R}^{18}$). This action space controls position targets for a proportional-derivative controller applied to each robot joint. The joints include the legs' thigh, calf, and hip joints and the six arm joints. The position targets are derived as $\sigma_a a^t + q_{def}$, where $\sigma_a = 0.5$ is a scaling factor, and q_{def} represents the robot's default joint configuration, which corresponds to the robot standing with its arm raised.

The observation, represented as o^t, relies solely on proprioceptive information. Its elements consist of the gravity vector projected in the base frame $g_b^t \in \mathbb{R}^3$, the base linear and angular velocities, $v_b^t \in \mathbb{R}^6$, the joint positions, $q^t \in \mathbb{R}^{18}$ and the previous actions $a^{t-1} \in \mathbb{R}^{18}$:

$$o^t = [g_b^t, v_b^t, q^t, a^{t-1}] \in \mathbb{R}^{45} \tag{1}$$

The observation, o^t, is augmented with the end-effector pose command for the policy input. This command is defined as the positional difference between the current and the target keypoints of the end-effector in the base frame ${}^b p_{ee}^{\text{cmd}} \in \mathbb{R}^9$, where each of the three keypoints provides a 3D position vector, resulting in a 9-dimensional vector in total.

E. Rewards

The final reward R is the sum of the task rewards R_T and penalties R_P: $R = R_T + R_P$. The task reward R_T can be divided into four subcategories: tracking, progress, feet contact force, and initial leg joint rewards: $R_T = \omega_1 R_t + \omega_2 R_p + \omega_3 R_f + \omega_4 R_q$, where $\omega_1 = 13$, $\omega_2 = 80$, $\omega_3 = 0.015$ and $\omega_4 = 0.4$.

Tracking Reward (R_t) is a delayed reward focused on tracking the three keypoints, representing the end-effector pose command, during the last two seconds ($T_r = 2\text{s}$) of a 4-second command cycle ($T = 4\text{s}$). Delaying the reward emphasizes the importance of being in the correct pose during the final 2 seconds without penalizing the path to getting there. This prevents the unwanted behavior that continuous rewards might encourage, such as passing through the robot's body when transitioning from a pose on one side of the robot to a target on the opposite side.

$$R_t = \begin{cases} \frac{1}{T_r} \sum_{k=0}^{3} e^{\frac{1}{\sigma_t} \| {}^b p_{ee,k}^{\text{meas}} - {}^b p_{ee,k}^{\text{cmd}} \|_2} & \text{if } t > T - T_r \\ 0 & \text{otherwise} \end{cases}$$

Here, ${}^b p_{ee,k}^{\text{meas}}$ and ${}^b p_{ee,k}^{\text{cmd}}$ are the positions of the measured and commanded keypoints in $\mathbb{R}^3$, respectively, and $\sigma_t = 0.05$.

Progress Reward (R_p) addresses the sparsity of the tracking reward and reduces end-effector oscillations by incentivizing steady progress toward the target. It compares the current distance $d^t \in \mathbb{R}^3$ between the measured and commanded keypoints to the smallest previously recorded distance $d \in \mathbb{R}^3$. If d^t is smaller, the reward is calculated as:

$$R_p = \begin{cases} \frac{1}{3} \sum_{k=0}^{3} (d_k - d_k^t) & \text{if } d^t < d \\ 0 & \text{otherwise} \end{cases}$$

Feet Contact Force Reward (R_f) encourages the robot to maintain ground contact with all four feet. The reward is non-zero only if all four feet are firmly in contact with the ground. To account for small disturbances and ensure genuine contact, 1 Newton is subtracted from the force on each foot, denoted as F_i. The reward is calculated as the sum of these adjusted forces:

$$R_f = \sum_{i=1}^{4} \max(F_i - 1, 0)$$

Initial Leg Joint Reward (R_q) encourages the robot to maintain its leg joints in a configuration close to those sampled from the locomotion policy, as these are known to result in a stable posture, with σ_q defined as 0.05.

$$R_q = \sum_{i=0}^{12} e^{\frac{1}{\sigma_q}(q_i^{\text{init}} - q_i)}$$

Penalties (R_P) penalize joint torques, joint accelerations, action rates and target joint positions above the limits: $R_P = \omega_5 \|\boldsymbol{\tau}\|^2 + \omega_6 \|\ddot{\boldsymbol{q}}\|^2 + \omega_7 \|\boldsymbol{a}_t - \boldsymbol{a}_{t-1}\|^2 + \omega_8 \|\boldsymbol{q} - \boldsymbol{q}_{\text{lim}}\|_1$, where $\omega_5 = -3e^{-5}$, $\omega_6 = -3e^{-6}$, $\omega_7 = -5e^{-2}$ and $\omega_8 = -1.3$. Finally, we terminate on knee and base contacts.

F. Terrains and Curriculum training

The robots are trained in simulation on four procedurally generated terrains: flat, randomly rough, discrete obstacles, and stairs, as defined in [19]. Gradually increasing task difficulty during training has been shown to enhance learning [1], [19], [23]. We employ a terrain curriculum similar to the one proposed in [19], adapted for end-effector pose tracking.

G. Initial poses

The proposed whole-body end-effector pose tracking policy does not include locomotion capabilities, as most tasks do not require simultaneous movement and object manipulation. Instead, it operates alongside a separate locomotion policy.

To ensure a stable leg posture, the robot's initial base pose $p_b^{init} \in \mathbb{SE}(3)$ and joint angles $q^{init} \in \mathbb{R}^{18}$ are taken from a pre-trained RL-based locomotion policy [24]. This policy, designed for the same mobile-legged manipulator, controls only the legs for rough terrain locomotion while incorporating arm joint positions and velocities in its observations and takes a 3D base velocity command as input, including linear x/y velocities and yaw rotation.

Before training, robots are placed at the terrain center with randomized heading commands for 4 seconds (Fig. 2-C). Stable configurations are recorded and used to initialize training, preventing sudden jumps when switching policies.

TABLE I: Average position $\bar{e}_p$ and orientation $\bar{e}_o$ errors for different added masses on the end-effector m_a for 10000 end-effector pose commands on flat terrain in simulation.

m_a [kg]	[0 - 2.0]	2.5	3.0	3.5	4.0	4.5
$\bar{e}_p$ [cm]	0.83	1.18	1.89	4.77	10.69	15.33
$\bar{e}_o$ [deg]	3.45	6.99	10.87	22.54	36.31	45.02

H. Sim-to-Real

When the training starts, we add a mass on the end-effector randomly sampled from the interval $[0, 2.0]$ kg, and the inertia of this rigid body links is scaled by the ratio between the new mass and the original one. Random perturbances are applied to the end-effector, such as an impulse force sampled from the interval $[-10, 10]$ N every 3 to 4 seconds, and random pushes on the robot's base simulated as base velocity impulses sampled from the interval $[-0.5, 0.5]$ $\mathrm{m\,s^{-1}}$ along the x-y dimensions. Random noise is also added to the observations.

III. RESULTS AND DISCUSSION

A. Simulation experiments

Table I presents the average position and orientation errors on flat terrain for different added masses on the end-effector in simulation. Within the training range $[0, 2.0]$ kg, the tracking errors remain stable, with an average position error of 0.83 cm and orientation error of $3.45°$. When the added mass exceeds the training distribution, the tracking performance degrades with errors reaching 15.33 cm and $45.02°$ for a 4.5 kg load. This highlights the controller's robustness beyond the training range while also demonstrating its limitations when encountering significantly higher payloads.

1) Comparison to model-based control: We compare our whole-body RL policy to the model-based MPC controller from [25], optimized for the same robot. For fairness, we fine-tuned its parameters for whole-body behavior. We evaluate both controllers on flat terrain using the same 35 end-effector pose commands in the expanded workspace. Both controllers perform similarly in terms of median errors, with the RL controller achieving 1.81 cm / $1.73°$, and the MPC controller 2.17 cm / $1.53°$. However, the mean errors are notably higher for the MPC controller, reaching 6.43 cm / $6.88°$, while the RL controller maintains significantly lower values at 2.21 cm / $2.01°$. This discrepancy results from the MPC controller's inability to manage the trade-off between pose tracking and self-collision avoidance, causing the arm to get stuck near the base during transitions between distant poses – an issue that our RL controller effectively avoids. Since MPC was tuned for flat terrain, we did not compare it to RL on stairs. Unlike our RL policy, MPC would require a terrain model to handle rough terrain, which was beyond our evaluation scope.

B. Hardware experiments

1) Pose tracking accuracy: We assess our controller's tracking performance using a motion capture system across 20 randomly sampled poses in the expanded workspace. Poses

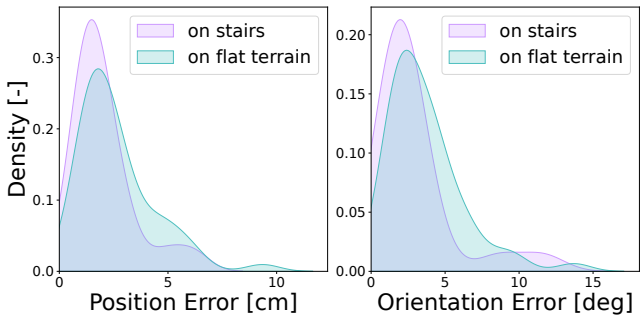

Fig. 3: Distribution of the position and orientation errors for 20 end-effector pose commands, measured on hardware, on both flat terrain and stairs.

are sent sequentially, with substantial changes in position and orientation, resulting in effective whole-body behavior as shown in Figure 1. The average error reaches 2.03 cm and $2.86°$. These results, which are illustrated in Figure 3, closely match the performance observed in simulation, demonstrating a minimal sim-to-real gap.

2) Robustness to external disturbances: We evaluate the tracking performance of our whole-body RL policy on stairs using a motion capture system across 20 sampled poses in the expanded workspace in the half-space in front of the robot under three base orientations: sideways on the stairs, facing up and facing down, as shown in Figure 1. The average position error reaches 2.64 cm, and the average orientation error $3.64°$. Figure 3 shows that the tracking performance remains consistent with that on flat terrain.

When switching orientations on the stairs a locomotion policy is used [24] and the transition between policies is smooth thanks to the robot initialization process described in Section II-G. Without this initialization step, the robot experiences jumps when transitioning between policies.

Additionally, the system can handle up to 3.75 kg of weight on the end-effector when stationary, and up to 1.3 kg in movement. This flexibility is advantageous as it avoids the need to model weight changes, typically required in model-based approaches, allowing for the attachment of various end-effectors and dynamic carrying of unknown payloads during operation.

IV. CONCLUSION

We have presented a whole-body RL-based controller for a quadruped with an arm that can reach even the most difficult poses. Our controller achieves high accuracy also on rough terrain (*e.g.*, on stairs), which we show in real-world experiments with ANYmal with an arm. Additionally, our contributions include providing a formulation for learning pose tracking that is superior to existing methods with poor accuracy or only considering position tracking. Our hardware experiments show an average tracking accuracy of 2.64 cm for position and $3.64°$ for orientation on challenging terrain.

REFERENCES

[1] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[2] 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, 2022.

[3] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning (CoRL)*, 2023, pp. 22–31.

[4] C. D. Bellicoso, K. Krämer, M. Stäuble, D. Sako, F. Jenelten, M. Bjelonic, and M. Hutter, "ALMA - Articulated Locomotion and Manipulation for a Torque-Controllable Robot," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8477–8483.

[5] J.-P. Sleiman, F. Farshidian, M. V. Minniti, and M. Hutter, "A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 6, no. 3, pp. 4688–4695, 2021.

[6] H. Dai, A. Valenzuela, and R. Tedrake, "Whole-body motion planning with centroidal dynamics and full kinematics," in *IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 295–302.

[7] Y. Abe, B. Stephens, M. Murphy, and A. Rizzi, "Dynamic whole-body robotic manipulation," in *Unmanned Systems Technology XV*, vol. 8741, 2013, pp. 280–290.

[8] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[9] Z. Fu, X. Cheng, and D. Pathak, "Deep Whole-Body Control: Learning a Unified Policy for Manipulation and Locomotion," in *Conference on Robot Learning (CoRL)*, 2023, pp. 138–149.

[10] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, "DeepMimic: Example-guided Deep Reinforcement Learning of Physics-based Character Skills," *ACM Transactions On Graphics*, vol. 37, no. 4, pp. 1–14, 2018.

[11] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac Gym: High Performance GPU-Based Physics Simulation For Robot Learning," *arXiv preprint arXiv:2108.10470*, 2021.

[12] P. Arm, M. Mittal, H. Kolvenbach, and M. Hutter, "Pedipulate: Enabling Manipulation Skills using a Quadruped Robot's Leg," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5717–5723.

[13] X. Cheng, A. Kumar, and D. Pathak, "Legs as Manipulator: Pushing Quadrupedal Agility Beyond Locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5106–5112.

[14] T. Portela, G. B. Margolis, Y. Ji, and P. Agrawal, "Learning Force Control for Legged Manipulation," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 15 366–15 372.

[15] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A Unified Simulation Framework for Interactive Robot Learning Environments," *IEEE Robotics and Automation Letters (RA-L)*, vol. 8, no. 6, p. 3740–3747, 2023.

[16] ANYbotics, "Anymal specifications," 2023, Accessed 1-September-2024. [Online]. Available: https://www.anybotics.com/anymal-autonomous-legged-robot/

[17] Duatic, "Dynaarm specifications," 2023, Accessed 1-September-2024. [Online]. Available: https://duatic.com/

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[19] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to Walk in Minutes Using Massively Parallel Deep Reinforcement Learning," pp. 91–100, 2022.

[20] H. Ha, Y. Gao, Z. Fu, J. Tan, and S. Song, "UMI on Legs: Making Manipulation Policies Mobile with Manipulation-Centric Whole-body Controllers," in *Conference on Robot Learning (CoRL)*, 2024.

[21] A. R. Geist, J. Frey, M. Zhobro, A. Levina, and G. Martius, "Learning with 3D rotations, a hitchhiker's guide to SO(3)," in *Proceedings of the 41st International Conference on Machine Learning*, vol. 235, 2024, pp. 15 331–15 350.

[22] A. Allshire, M. MittaI, V. Lodaya, V. Makoviychuk, D. Makoviichuk, F. Widmaier, M. Wüthrich, S. Bauer, A. Handa, and A. Garg, "Transferring Dexterous Manipulation from GPU Simulation to a Remote Real-World TriFinger," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[23] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne, "ALLSTEPS: Curriculum-driven Learning of Stepping Stone Skills," in *Computer Graphics Forum*, vol. 39, no. 8, 2020, pp. 213–224.

[24] P. Arm, G. Waibel, J. Preisig, T. Tuna, R. Zhou, V. Bickel, G. Ligeza, T. Miki, F. Kehl, H. Kolvenbach, and M. Hutter, "Scientific exploration of challenging planetary analog environments with a team of legged robots," *Science Robotics*, vol. 8, no. 80, p. eade9548, 2023.

[25] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A Collision-Free MPC for Whole-Body Dynamic Locomotion and Manipulation," pp. 4686–4693, 2022.