PGTT: Phase-Guided Terrain Traversal for Perceptive Legged Locomotion

Alexandros Ntagkas[1,2], Chairi Kiourt[2], and Konstantinos Chatzilygeroudis[1,2,3]

***Abstract*— State-of-the-art perceptive Reinforcement Learning controllers for legged robots either (i) impose oscillator or IK-based gait priors that constrain the action space, add bias to the policy optimization and reduce adaptability across robot morphologies, or (ii) operate "blind", which struggle to anticipate hind-leg terrain, and are brittle to noise. In this paper, we propose Phase-Guided Terrain Traversal (PGTT), a perception-aware deep-RL *approach* that overcomes these limitations by enforcing gait structure purely through reward shaping, thereby *reducing inductive bias* in policy learning compared to oscillator/IK-conditioned action priors. PGTT encodes per-leg phase as a cubic Hermite spline that adapts swing height to local heightmap statistics and adds a swing-phase contact penalty, while the policy acts directly in joint space supporting morphology-agnostic deployment. Trained in MuJoCo (MJX) on procedurally generated stair-like terrains with curriculum and domain randomization, PGTT achieves the highest success under push disturbances (median +7.5% vs. the next best method) and on discrete obstacles (+9%), with comparable velocity tracking. We validate PGTT on a Unitree Go2 using a real-time LiDAR elevation-to-heightmap pipeline, and we report preliminary results on ANYmal-C obtained *with the same hyperparameters*. These findings indicate that terrain-adaptive, phase-guided reward shaping is a simple and general mechanism for robust perceptive locomotion across platforms.**

I. Introduction

Legged robots promise unmatched mobility in cluttered, uneven, and human-made environments, but robust gait control on such terrain remains challenging [1], [2]. Reinforcement learning (RL) has shown that agile locomotion behaviors can be learned from data [3], yet many studies assume *idealized sensing* (privileged terrain information) or operate "blind," which hinders anticipation of obstacles and reduces reliability on hardware [4], [5]. As a result, perception is essential, but the representation and how it interfaces with control are pivotal for generality and robustness.

We propose **Phase-Guided Terrain Traversal (PGTT)**, a perception-aware deep-RL approach that retains the benefits of rhythmic structure while avoiding IK and action-space constraints. PGTT uses a robot-centric *heightmap* (derived online from LiDAR elevation mapping) as a compact terrain representation and encodes per-leg phase with a *cubic Hermite spline* whose swing apex adapts to local height statistics. Crucially, the phase prior is enforced *only through reward shaping*, while the policy acts directly in joint space. This design keeps the action space unconstrained and *reduces inductive bias* in policy learning compared to oscillator/IK-conditioned targets, easing deployment across different morphologies.

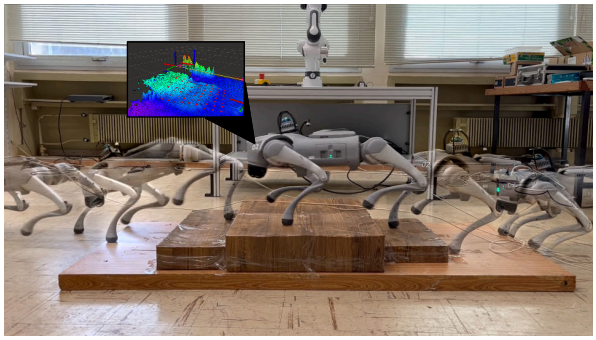


Fig. 1: Real-world example of the Unitree Go2 robot climbing stair terrain.

The main contributions of this manuscript are: 1) A terrain-adaptive, phase-guided *reward* that encodes a Hermite-spline swing trajectory driven by local heightmap statistics and penalizes swing-phase contacts, without constraining the action space or using IK, thereby reducing inductive bias and improving morphology-agnostic deployment, and 2) an accessible training stack using MuJoCo/MJX that provides accurate dynamic simulation and high throughput on a single consumer GPU, offering a lightweight alternative to Isaac Gym-based pipelines [6] (our code is available at `https://github.com/NtagkasAlex/phase_guided_terrain_traversal`).

II. Related Work

Phase-augmented controllers specify foot or joint targets as functions of a per-leg phase and track them with IK/PD controllers, improving stability but coupling the policy to morphology and introducing action-space bias [4], [7]. Central Pattern Generators (CPG)-based methods similarly embed oscillators and let RL modulate their parameters, inheriting the same limitations [8]. An alternative is to encode gait regularity in the *objective* rather than in the actions: phase-guided reward shaping encourages desired swing/stance timing and foot clearance while leaving the policy free to decide the final commands [9]. This shift reduces inductive bias and eases deployment across platforms with different kinematics.

In this landscape, PGTT aligns with direct joint-space control but differs in how structure is injected: it uses a robot-centric heightmap for perception and enforces a *terrain-adaptive, phase-guided prior purely through reward shaping*, avoiding oscillators and IK. This design aims to

*This work was supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "3rd Call for H.F.R.I. Research Projects to support Post-Doctoral Researchers" (Project Acronym: NOSALRO, Project Number: 7541). This work has also been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.

[1]Laboratory of Automation and Robotics (LAR) in the Department of Electrical & Computer Engineering, University of Patras, GR-26504 Patras, Greece, a_ntagkas@ac.upatras.gr, costashatz@upatras.gr

[2]Archimedes/Athena RC, Greecechairiq@athenarc.gr

[3]Computational Intelligence Laboratory (CILab), Department of Mathematics, University of Patras, GR-26110 Patras, Greece

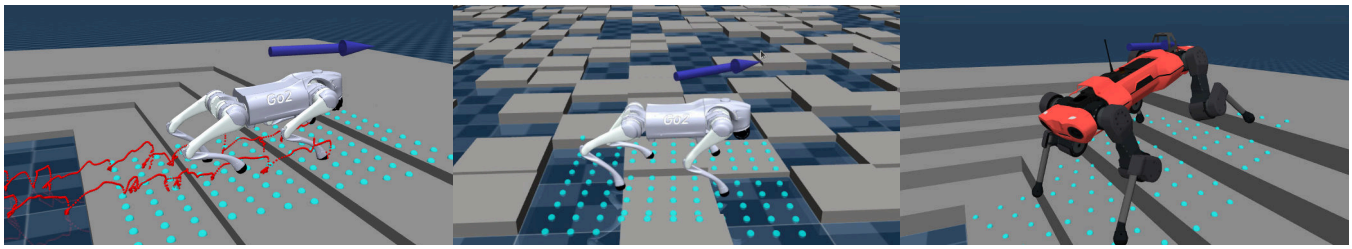


Fig. 2: Simulation snapshots of PGTT. Left: Go2 on stairs with projected front-foot trajectories (red). Middle: Go2 traversing discrete obstacles. Right: ANYmal C on stairs.

retain the benefits of rhythmic organization while minimizing action-space constraints, thereby reducing inductive bias and supporting morphology-agnostic deployment relative to oscillator/IK-conditioned policies [4], [7], [9].

III. Phase-Guided Terrain Traversal

At a high level, **Phase-Guided Terrain Traversal (PGTT)** combines three ideas (Fig. 3): (i) a compact perception module that encodes terrain as a robot-centric heightmap derived online from LiDAR measurements, (ii) phase variables and reward function that provide rhythmic structure without constraining the action space, and (iii) an asymmetric actor–critic architecture trained with PPO in GPU-accelerated MuJoCo (MJX) environments.

A. Problem Formulation

We model legged locomotion as an infinite-horizon partially observable Markov decision process (POMDP)

$$\mathcal{M}=(\mathcal{S},\mathcal{A},\mathcal{O},P,\Omega,r,\gamma,\rho_0),$$

where $s_t\in\mathcal{S}$ is the full state, $a_t\in\mathcal{A}$ the action, and $o_t\in\mathcal{O}$ a partial observation. The transition kernel is $P(s_{t+1}\,|\,s_t,a_t)$, the observation model (sensor and preprocessing pipeline) is $\Omega(o_t\,|\,s_t)$, $r:\mathcal{S}\times\mathcal{A}\to\mathbb{R}$ is the reward, $\gamma\in[0,1)$ the discount factor, and ρ_0 the initial-state distribution. In our setting, o_t comprises proprioception and a robot-centric heightmap derived online from LiDAR, while s_t additionally includes privileged quantities used only during training. A stochastic policy $\pi_\theta(a_t\,|\,o_t)$ maximizes the discounted return

$$J(\pi_\theta)=\mathbb{E}_{\substack{s_0\sim\rho_0\\ o_t\sim\Omega(\cdot|s_t),a_t\sim\pi_\theta(\cdot|o_t)\\ s_{t+1}\sim P(\cdot|s_t,a_t)}}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]. \quad (1)$$

Action Space: The action space is a 12×1 vector, a_t, corresponding to the desired joint angle of the robot. To facilitate learning, we train the policy to infer the desired joint angle around the robot's stand still pose. Hence, the robot's desired joint angles are computed as

$$q_{des}=q_{def}+ka_t, \quad (2)$$

where k is a constant *action scale* parameter.

Observation Space: The observation space o_t, which is passed to the policy network $\pi_\theta(a_t|o_t)$, consists of mainly proprioceptive and exteroceptive measurements. To encode the leg phase, we use $\cos(\phi),\sin(\phi)$ instead of $\phi=[\phi_0,\phi_1,\phi_2,\phi_3]$, which is a smooth and unique representation for the angle [7].

$$o_t=[\omega_t\ g_t\ q_t\ \dot{q}_t\ \cos(\phi)\ \sin(\phi)\ h_t\ f\ a_{t-1}\ v_{cmd}]^T, \quad (3)$$

where $\omega_t,g_t,q_t,\dot{q}_t,\cos(\phi),\sin(\phi),h_t,f,a_{t-1}$ and v_{cmd} are the body angular velocity, the gravity vector expressed in the local frame, the joint angles, the joint velocities, the phase representation, the flattened height-scans of the terrain, the base frequency, the last action and the command.

Value Network: The value network is trained to output an estimation of the true state value, $V(s_t)$. Unlike the policy the state s_t contains privileged information

$$s_t=[o_t\ v_t\ ]^T, \quad (4)$$

where v_t is the linear velocity in the local frame. Linear velocity is critical because it correlates strongly with the main objective-track commanded velocity- and thus with the value function output.

B. Phase-Guided Reward Function

Reward design is central to legged locomotion with reinforcement learning. Most existing approaches combine a forward-velocity tracking term with a set of penalties (slip, foot clearance) to promote stable gaits. While effective, these reward structures often require extensive manual tuning and are usually combined with oscillators or IK-based controllers.

PGTT pursues a different route: we aim to generate phase-guided swing trajectories *without* inverse kinematics. The phase prior influences learning only through the reward, which reduces the number of hand-tuned terms and avoids constraining the policy. The core idea is to use *cubic Hermite splines* to define smooth foot trajectories conditioned on a per-leg phase variable and local terrain information.

We denote by $p_{f,z,i}$ the z-axis (height) position of foot i in the hip-joint frame, and by $p_{w,f,z,i}$ the corresponding position in the world frame. Let d_b be the nominal foot height in stance (default configuration) and d_s the nominal swing apex (see Fig. 4). To adapt the trajectory to terrain, we compute local statistics around each leg: $H_{\max,i}$ and $H_{\min,i}$ are the maximum and minimum terrain heights in the world frame, and $\delta H_i=H_{\max,i}-H_{\min,i}$ is added to the swing trajectory to guarantee obstacle clearance.

Formally, a cubic Hermite spline is defined by start and end positions p_0,p_1, tangents m_0,m_1, and duration T. For $t\in[0,T]$, the trajectory is

$$\begin{aligned} P(t)&=c_0+c_1t+c_2t^2+c_3t^3,\\ c_0&=p_0,\quad c_1=m_0,\\ c_2&=\frac{3}{T^2}(p_1-p_0)-\frac{2}{T}m_0-\frac{1}{T}m_1,\\ c_3&=-\frac{2}{T^3}(p_1-p_0)+\frac{1}{T^2}(m_0+m_1). \end{aligned} \quad (5)$$

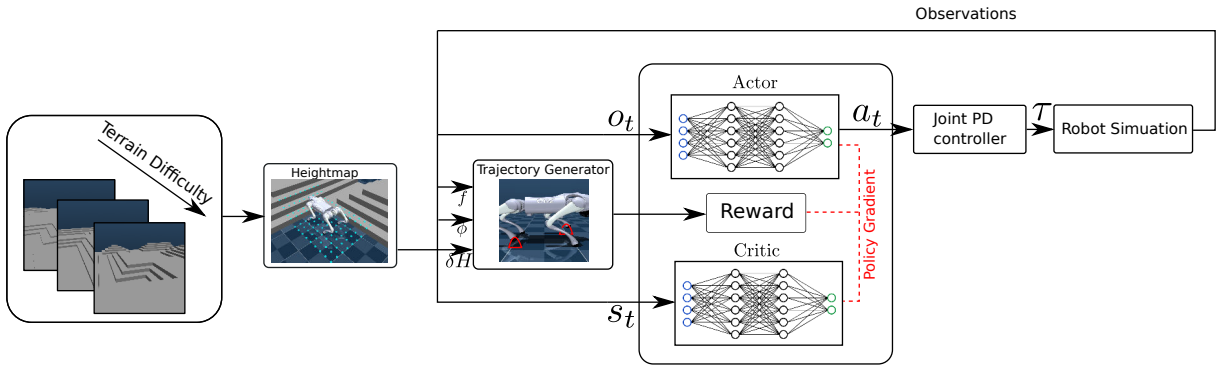


Fig. 3: PGTT combines curriculum learning, a robot-centric heightmap, reward shaping through Hermite splines, asymmetric actor-critic learning, and low-level PD controllers for effective perceptive legged locomotion.

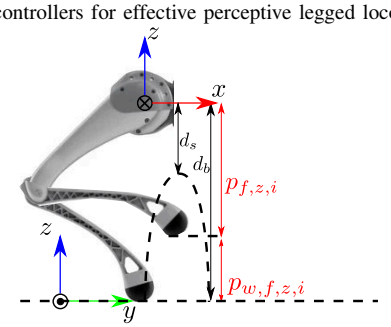


Fig. 4: Distances relative to the hip-joint frame and world frame. The *black* leg is the nominal stance, the *dashed* line a possible swing trajectory, and the *red* leg a random leg configuration.

We divide each leg trajectory into three phases (parameterized by $\phi_{i,t}$):

- **Stance:** foot remains at d_b until $\phi_{i,t} = T_{\text{stance}}$, where $T_{\text{stance}} = 2\pi p_{\text{stance}}$ and p_{stance} is the stance ratio.
- **Swing up:** spline P_{su} with parameters $(d_b,\ d_s + \delta H_i, 0, 0, T_{\text{swing}})$, duration $T_{\text{swing}} = 2\pi(1-p_{\text{stance}})/2$.
- **Swing down:** spline P_{sd} with parameters $(d_s + \delta H_i,\ d_b,\ 0,\ 0,\ T_{\text{swing}})$, starting at $\phi_{i,t} = T_{\text{peak}} = 2\pi(1+p_{\text{stance}})/2$.

The desired z-position of foot i at phase $\phi_{i,t}$ is then

$$p^{\text{des}}_{f,z,i}(\phi_{i,t}, h_t) = \begin{cases} d_b, & 0 \le \phi_{i,t} < T_{\text{stance}}, \\ P_{su,i}(\phi_{i,t} - T_{\text{stance}}, h_t), & T_{\text{stance}} \le \phi_{i,t} < T_{\text{peak}}, \\ P_{sd,i}(\phi_{i,t} - T_{\text{peak}}, h_t), & T_{\text{peak}} \le \phi_{i,t} < 2\pi. \end{cases} \quad (6)$$

Apart from the task-specific rewards (e.g. linear velocity tracking in our case), the central *positive* term encourages each foot to follow its terrain-adaptive, phase-guided trajectory:

$$r_{\text{phase}} = \sum_{i \in \text{feet}} \exp\left(- \frac{\left(p^{\text{des}}_{f,z,i}(\phi_{i,t}, h) - p_{f,z,i}\right)^2}{\sigma_f} \right). \quad (7)$$

To discourage premature contacts during swing, we include a *negative* penalty:

$$r_{\text{contact}} = \sum_{i \in \text{feet}} \mathbb{1}_{\pi \le \phi_{i,t} < 2\pi} c_i, \quad (8)$$

where $c_i = 1$ if foot i is in ground contact and 0 otherwise. This term penalizes collisions when the phase variable indicates that the leg should be swinging.

IV. Experimental Setups and Results

In this section we will present the results of the proposed policy in simulation and the real world and compare them with the baseline policies in terms of several metrics. All policies were trained on a workstation equipped with an Intel Core i9-14900K CPU and a single NVIDIA GeForce RTX 3080 GPU. Training used a physics-integration time step of $dt = 0.005$s. During deployment, in both Sim2Sim and Sim2Real transfers, control commands are issued at 50 Hz (i.e., every 0.02 s).

A. Baselines

We select baseline methods that are both relevant to our problem and representative of existing approaches to enable a fair comparison with our method. To evaluate whether locomotion without fixed gait scheduling can yield more efficient behaviors, we include *MassLoco* [10], including rewards inspired by Margolis et al. [11] to encourage more natural walking patterns. On the other hand, when considering a state-of-the-art method that leverages gait priors, we compare against *Wild* [7]. We did not include Visual CPG-RL [12], since, although its framework is similar to Wild, it is not trained or evaluated on stairs or obstacle traversal, and is therefore considered less relevant for our study.

B. Metrics

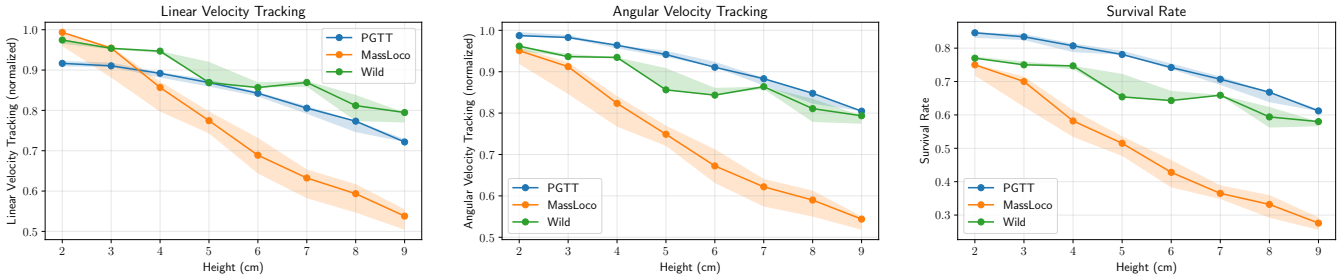
We compare our method PGTT with the two aforementioned baseline MassLoco and Wild in terms of linear and angular velocity tracking and success rate. Success rate (SR) is defined as follows *Gangapurwala et al.* [13]:

$$\text{SR} = 1 - \frac{N_e}{N_T}, \quad (9)$$

with N_e referring to the number of rollouts that terminated early due to a prohibited behavior and N_T being the total number of rollouts. Using $N_T = 1000$, we randomize the base linear and angular velocity command with $0.7 \cdot v^{\max}$ from the one used during training.

C. Simulation Results

We apply perturbations of uniformly sampled magnitude between 7.5 to 30 N and we also sample durations and wait times between consecutive perturbations. We replicate the whole training pipeline over 5 different seeds. All metrics excluding the success rate are normalized with respect the



(a) Linear velocity (b) Angular velocity (c) Success rate

Fig. 5: Comparison of PGTT with baseline methods MassLoco and Wild across three metrics: linear velocity tracking, angular velocity tracking and success rate. *Solid lines* show the median over 5 different training seeds and the shaded regions are the regions between the 25-th and 75-th percentiles.

TABLE I: Evaluation metrics when the robot traverses discrete obstacles of varying height from 2cm to 9cm. Success rate, normalized body linear velocity error $\bar{v}$, and normalized body angular velocity error $\bar{\omega}$ for 1000 quadrupeds. Results are (**median**, 25th percentile, 75th percentile) over 5 training seeds.

Method	Success Rate	$\bar{v}$	$\bar{\omega}$
PGTT	(**0.848**,0.842,0.855)	(**0.965**,0.958,0.972)	(**0.991**,0.986,0.994)
MassLoco	(**0.702**,0.659,0.711)	(**0.983**,0.939,0.986)	(**0.903**,0.863,0.904)
Wild	(**0.756**,0.756,0.769)	(**0.998**,0.998,1.000)	(**0.935**,0.935,0.941)

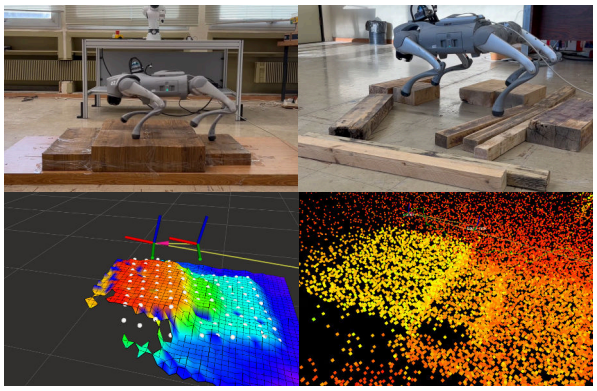


Fig. 6: Real world experiments. The bottom left image shows the gridmap and the bottom right shows the odometry.

the maximum value. The results reveal several clear trends (Fig. 5): PGTT and Wild exhibit very similar commanded velocity tracking, whereas MassLoco lags behind. PGTT achieves the highest success rate, **outperforming the second-best method, Wild, by 7.5% on average**.

Similar behavior is observed when evaluating the three methods in environments with discrete obstacles, with PGTT achieving the highest success rate; **9% higher than the second-best method**. Additionally, both angular and linear velocity tracking are very similar across all methods (Table I).

ANYmal C Experiments: We also applied our method on the same task with the ANYmal C robot. Our preliminary experiments showcase that PGTT is able to generate walking behaviors without even changing the hyper-parameters (see Fig. 2 and the supplementary video).

D. Real-World Deployment

We evaluate the Sim2Real capabilities of our method on a Unitree Go2 quadruped. For perception, a L1 LiDAR is fused with IMU data using Point-LIO [14], a tightly coupled LiDAR–Inertial Odometry framework. The resulting odometry and transformed point cloud are used to construct a robot-centric elevation grid map [15], where each cell (i,j) stores a mean height $\hat{h}ij$ and variance σij^2 to represent terrain uncertainty. Since raw LiDAR maps often contain holes (`NaN` values) that can destabilize the policy, we apply a median-fill filter that in-paints only small gaps (below radius r_{hole}) surrounded by reliable data, while leaving larger unknown regions untouched.

To provide real-time input to the policy, we extract a robot-centric heightmap by sampling an 11×9 grid within a $1.1m\times0.9m$ area centered on the robot. Although accuracy is bounded by grid resolution, the domain randomization used during training makes the policy robust to such imperfections. The locomotion policy executes at $50Hz$, producing joint targets that are translated into torques through a lightweight PD controller ($k_p=60$, $k_d=3$) before being applied by the Go2's onboard low-level controller.

Our experiments showcase that policies trained with the PGTT effectively transfer to the real-world (Fig. 6), and the robot is able to walk both on static stair and discrete obstacles environments, and withstand real-life perturbations. The supplementary video showcases such examples.

V. Conclusion

In this work, we introduced **Phase-Guided Terrain Traversal (PGTT)**, a perception-aware locomotion framework that integrates local heightmap perception, reinforcement learning, and terrain-adaptive gait priors to achieve robust and efficient terrain traversal. Our results demonstrate that PGTT increases traversal success rates by **7.5%**. Moreover, it generalizes across robot platforms without relying on inverse kinematics and transfers successfully to real hardware, as demonstrated by reliable deployment on a Unitree Go2. A key strength of PGTT is its reliance on the lightweight MuJoCo simulation stack, which allows perception-aware locomotion policies to be developed and trained on affordable hardware such as a single consumer-grade GPU, thus lowering the barrier to entry for this line of research.

Overall, PGTT provides an accessible and effective foundation for advancing agile, robust, and affordable legged locomotion in real-world environments, empowering researchers and laboratories without extensive computational resources to contribute to this field.

REFERENCES

[1] M. Hutter, C. Gehring, A. Lauber, *et al.*, "Anymal - a highly mobile and dynamic quadrupedal robot," *IEEE/RSJ IROS*, 2017.

[2] S. Kuindersma, F. Permenter, and R. Tedrake, "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot," *Autonomous Robots*, vol. 40, pp. 429–455, 2016.

[3] A. Kumar, K. Jatavallabhula, *et al.*, "Rma: Rapid motor adaptation for legged robots," *Robotics: Science and Systems (RSS)*, 2021.

[4] J. Lee, J. Hwangbo, *et al.*, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, 2020.

[5] Y. Duan, M. Zhang, *et al.*, "Learning a bipedal walking policy via reinforcement learning and gait library," in *Conference on Robot Learning (CoRL)*, 2021.

[6] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[7] 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, p. eabk2822, 2022.

[8] G. Bellegarda and A. J. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12547–12554, 2022.

[9] Y. Shao, Y. Jin, X. Liu, W. He, H. Wang, and W. Yang, "Learning free gait transition for quadruped robots via phase-guided controller," *IEEE Robotics and Automation Letters*, vol. 7, p. 1230–1237, Apr. 2022.

[10] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," 2022.

[11] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," 2022.

[12] G. Bellegarda, M. Shafiee, and A. Ijspeert, "Visual cpg-rl: Learning central pattern generators for visually-guided quadruped locomotion," 2024.

[13] S. Gangapurwala, L. Campanaro, and I. Havoutis, "Learning low-frequency motion control for robust and dynamic robot locomotion," 2023.

[14] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point–LIO: Robust high-bandwidth lidar–inertial odometry," *Advanced Intelligent Systems*, vol. 5, no. 7, p. 2200459, 2023.

[15] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3019–3026, 2018.