2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Learning Humanoid Loco-manipulation with Constraints as Terminations

Pierre-Alexandre Leziart[1], Mitsuharu Morisawa[1], Fumio Kanehiro[1]

Abstract— **Deep Reinforcement Learning (RL) is now commonly used for controlling legged robots. Several recent studies have demonstrated impressive results in solving increasingly complex robotic tasks such as navigation in unstructured environments or loco-manipulation. However, this complexity often comes with intricate learning setups requiring tedious reward shaping and features to help convergence. In this work, we tackle these issues and achieve loco-manipulation with a humanoid robot using a RL algorithm that enforces constraints through stochastic terminations during policy learning. We keep the number of rewards low by reformulating them as constraints when they can be intuitively expressed that way. Moreover, we study the relevance of various learning features encountered in the literature and show that providing observations without noise or privileged information to the critic are two straightforward ways to boost locomotion performances on rough terrains. We also demonstrate that the proposed minimalist architecture is not limited to pure locomotion but extends to a loco-manipulation task involving upper limbs. Videos are available at humanoid-cat.github.io.**

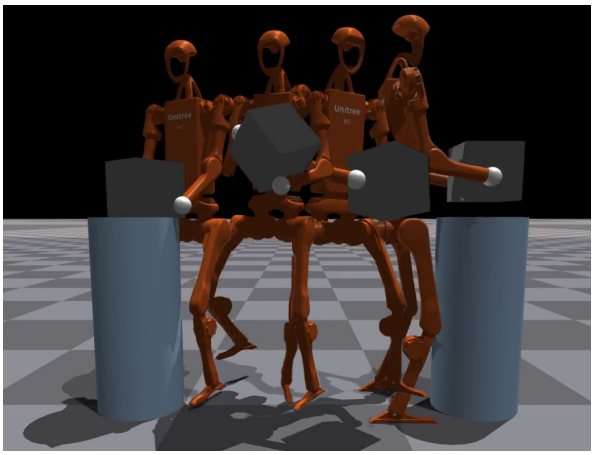

Fig. 1: Loco-manipulation with H1 to transport a box from the right pillar to the left one using a single policy trained in simulation with constraints as terminations.

I. Introduction

Deep reinforcement learning (RL) algorithms have made significant progress in recent years for controlling legged robots. RL approaches can now efficiently accomplish a wide variety of complex locomotion tasks with quadruped robots such as parkour [1], [2], search and rescue [3] or low-gravity acrobatics [4]. By comparison, the field of humanoid robotics has not been thoroughly explored, although it is currently undergoing a surge of interest given that our everyday world is designed around the human form factor [5]. Results with model-free RL are fewer and have yet to fully catch up with their quadruped counterparts [6], [7]. Price remains the main barrier that keeps research laboratories away from humanoid robots when quadrupeds have already reached large-scale commercialization with excellent cost-performance ratio [8]. Moreover, due to their greater complexity and inherent instability, additional techniques are to be developed to handle the difficulties of balancing bipeds with respect to quadrupeds [9]. For these reasons, the design of reward functions and training strategies is often more challenging when tackling humanoid learning tasks. The transfer to practical applications is also more difficult because of the additional efforts and expert knowledge required to cross the sim-to-real gap. In this paper, we sought to ease this endeavor by applying lessons learned from previous work on quadruped learning [10] to design a workflow for humanoid loco-manipulation that requires minimal design choices.

[1]CNRS-AIST JRL (Joint Robotics Laboratory), National Institute of Advanced Industrial Science and Technology (AIST), Japan. Contact: pa.leziart@aist.go.jp

In a RL scheme, properly shaping the reward function is key to guide the learning process and accomplish the desired task while abiding by the physical limitations of the system. Ensuring natural, efficient and safe motions often involve many terms that have clear physical meanings, such as joint torque and velocity limits. The burden of reward tuning can be alleviated by considering those terms as constraints rather than mixing them with purely task-related ones, a common practice in model-based control [11], [12]. Constrained RL methods have already been applied to legged locomotion [13], [14] yet their adoption remains limited. They can help reduce the scope of the hyperparameter search for the reward function, albeit often at the cost of algorithmic complexity. This complexity is a hurdle for reproducibility and broader integration in various learning frameworks as implementation details can have major impacts on performance and result consistency [15].

In this paper, we exploit the genericness of *Constraints as Terminations (CaT)* [10] to propose a streamlined approach for humanoid loco-manipulation with constrained RL. This method enforces constraints in a minimalistic fashion through stochastic terminations during policy learning. In practice, constraint violations increase a probability of terminating the future rewards the RL agent could have achieved. This naturally guides the agent toward behaviors that satisfy constraints to maximize future rewards.

Our approach is implemented on top of an off-the-shelf Proximal Policy Optimization (PPO) [16] framework for simplicity and reproducibility purpose [17]. We first design a set of safety and style constraints to achieve safe and

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

natural motion while limiting the reward shaping burden. We then draw inspiration from several successful architectures to explore the impact of various learning features to assess how relevant they are to a RL implementation [18], [19], [20], [21]. We demonstrate the effectiveness of these features and of our approach as a whole by deploying locomotion policies on a blind H1 humanoid robot crossing rough terrain in simulation. Finally, we push our scheme further with a loco-manipulation scenario involving the transport of a box to a target location (see Fig. 1).

In summary, our contributions are the following:

1) we transfer CaT to a humanoid robot for the first time and design a set of rewards and constraints to shape the behavior in a minimalist fashion in simulation,
2) we compare the relevance of learning features encountered in the literature to assess their impact on performance for a velocity tracking task on rough terrain,
3) and we highlight the potential of our approach by extending it to humanoid loco-manipulation for transporting a package in simulation.

II. Related Work

Following its rise in the recent years, reinforcement learning has become a well-established method for obtaining effective and robust controllers for legged robots. There is not a single canonical scheme but rather a wide range of algorithms that achieved impressive results in solving complex tasks [22], [7]. The work of independent research teams over the years has led to a flurry of frameworks and reimplementation of the baseline algorithms, each with their own slight differences. However, efforts are being made to provide standardized implementations with a focus on reproducibility [23], [24]. This endeavour is commendable as even minor implementation details have been shown to affect performances and result consistency [15]. Both [25] and [26] provide extensive studies on how much implementation details matter for RL. These details apply to all the parts of the learning pipeline. Decisions can be made for the gradient-based optimization to use an adaptive learning rate or to clip the gradients or the loss of the value function. The environments can also include various learning features when training in simulation. Observation noise can be applied only to the actor so that the critic has access to cleaner signals to learn from to assess the value function [19], [20]. While some prefer to encode privileged information in a latent space through a teacher-student scheme [27], [28], others choose a lighter way implementation-wise by directly providing them to the critic only. That way there is no need to reconstruct the latent space at runtime yet the critic can still use it to better assess the value function during training [19], [21]. The choice of quantities for domain randomization or the way to disturb the agents, either with a fixed schedule [29] or based on Bernoulli trials [18], are but other decisions that can impact final performances. In this work, we seek to assess the relevance of the aforementioned features for humanoid locomotion on rough terrain.

Humanoid locomotion has seen a resurgence of recent developments that benefit from techniques and tools already applied on quadruped robots. Although training policies in simulation before transferring them to the real world has been a popular approach for a while [30], [31], it has now been boosted by several GPU-based simulators capable of simulating thousands of robots in parallel [32], [17], which has streamlined this process [29]. However, even with this step up in term of sample generation, exploration remains an issue for complex tasks, especially when the robots bring their inherent complexity, like humanoid robots. Imitation learning, also referred to as learning by demonstration, offers the means to quickly transfer skills from a demonstrator to a learning agent [33]. With it, humanoid robots can learn to replicate natural whole-body locomotion patterns and execute seamless gait transitions by mimicking human motions from a motion capture dataset [34], [35]. This also applies to humanoid loco-manipulation tasks using teleoperation examples to imitate [36] or motion tracked demonstrations [37]. This reliance on prior expert data to guide the learning process has its own drawbacks as motion tracking errors or alignment errors with the robot kinematics can cause compounding issues down the line. [38] provides an alternative with a hierarchical scheme that combines a high-level waypoint planner with several specialized low-level policies that handle motion primitives (pick up, put down, walk). The closest developments from our own appears to be [39] with the sim-to-real transfer of a controller on a humanoid robot to move boxes from one table to another using 5 separate RL policies for the various stages of the motion. By comparison, we reduce the overall number of reward terms thanks to our use of constraints and focus on having a single policy for the whole pick up, walk and put down sequence, albeit for a scenario with lesser variability.

III. Method: Constraints as Termination

We consider learning a locomotion skill as an infinite, discounted Markov Decision Process (MDP) [40], defined by the tuple $(\mathcal{S}, \mathcal{A}, P, r, \gamma)$ with state space $\mathcal{S}$, action space $\mathcal{A}$, reward function r, discount factor γ and state transition probability P which represents the probability density of the next state $s_{t+1} \in \mathcal{S}$ given the current state $s_t \in \mathcal{S}$ and action $a_t \in \mathcal{A}$. Let $\rho_\pi(s_t)$ and $\rho_\pi(s_t, a_t)$ denote the state and state-action marginals of the trajectory distribution induced by a policy $\pi(s_t, a_t)$. Standard RL aims to maximize the expected sum of discounted rewards:

$$\max_\pi \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t). \tag{1}$$

Following [10], we consider instead maximizing rewards while avoiding constraint violations at each time step, which leads to the following constrained RL problem:

$$\max_\pi \mathbb{E}_{(s_t,a_t)\sim\rho_\pi} \sum_{t=0}^{\infty} \left(\prod_{t'=0}^{t} \gamma(1 - \delta(\mathbf{s}_{t'}, \mathbf{a}_{t'})) \right) r(\mathbf{s}_t, \mathbf{a}_t), \tag{2}$$

where the random variable $\delta_t : \mathcal{S} \times \mathcal{A} \to [0, 1]$ is an increasing function of the constraint violations. δ_t does not act

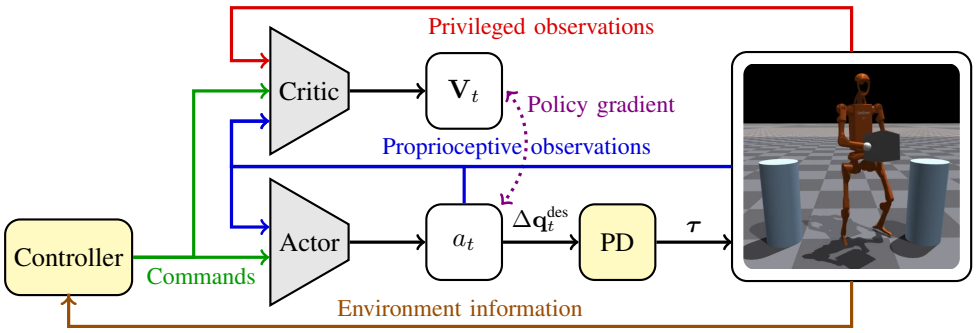

Fig. 2: Overview of the proposed constrained reinforcement learning approach. The PD controller converts desired joint position offsets into desired torques according to (4). The controller outputs either base velocity commands to track (locomotion task) or quantities related to the box and drop-off locations expressed in the frame of the robot (locomanipulation task).

on environment resets and is rather a probability of episode terminations, which from a policy learning perspective means that future rewards are terminated from time step t. If no constraints are violated then $\delta_t = 0$, whereas if one or more constraints are violated, δ_t may take positive values between 0 and 1. In that case, the sum of all future rewards will be down-scaled by $(1 - \delta_t)$. Therefore, the agent naturally gravitates towards satisfying the constraints to maximize the sum of future rewards.

The termination probability is computed as follows based on the set of constraint functions $\{c_i : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, i \in I\}$, with $I \subset \mathbb{N}$ the set of constraint indices:

$$\delta = \max_{i \in I} p_i^{\max} \text{clip}(\frac{c_i^+}{c_i^{\max}}, 0, 1), \tag{3}$$

where $c_i^+ = \max(0, c_i(\mathbf{s}, \mathbf{a}))$ is the violation of constraint i and $c_i^{\max}$ is an exponential moving average of the maximum constraint violation over the last batch of samples. Hyperparameter $p_i^{\max}$ controls the maximum termination probability for the i-th constraint and can be used as part of a curriculum scheme to slowly scale the enforcement of specific constraints. Please refer to [10] for a more in-depth description of the CaT approach.

IV. Learning Humanoid Locomotion with CaT

A. Overview

The objective of the proposed controller is to achieve blind locomotion for humanoid robots, enabling them to cross rough terrains based solely on proprioceptive sensor data. To do so, the controller shall track as well as possible an horizontal linear velocity command $\mathbf{v}_{\text{xy}}^{\text{des}}$ and a turning rate $\mathbf{w}_{\text{z}}^{\text{des}}$, both for the waist and given by a user or a higher level controller. The obtained behavior shall be robust to external disturbances and terrain variations for deployment in uncontrolled environments.

Our training pipeline builds upon recent state-of-the-art work [10] that leveraged constrained reinforcement learning in a physics simulator to achieve agile locomotion over challenging terrains while satisfying safety and style constraints with the open-hardware quadruped robot Solo-12 [41]. Unlike this previous work which used exteroceptive information to walk up stairs, traverse slopes, and climb over high obstacles, we rather focus on a minimal setup with pure proprioception for the first deployment of CaT on a humanoid platform. Being generic to the agent and environment, we found that the approach transfers well to a human-sized bipedal robot.

During training the robots are regularly pushed in random direction. Moreover, they are not only placed on flat ground but also on various increasingly difficult parametrized terrains whose geometries are procedurally generated [29]. This scheme provides rich interaction data for robustness and aims to limit the encounters of out-of-distribution states during deployment.

Domain randomization is used to tackle the sim-to-real gap and ease policy transferability to the real robot by forcing the policy to cater to a wide range of possible physical properties [27], [42].

B. Observation and action spaces

For this humanoid locomotion problem, the state space $\mathcal{S} \in \mathbb{R}^{43}$ includes the measured positions $\mathbf{q}_t$ and velocities $\dot{\mathbf{q}}_t$ of the 10 lower joints of the robot (5 per leg), the previous action $\mathbf{a}_{t-1}$, the waist angular velocity $\mathbf{w}$, the gravity vector projected in waist frame ${}^{w}\mathbf{g}$, central phases ϕ for each leg defined as cosine and sine pairs $[\cos(\phi(t)), \sin(\phi(t))]$ and the linear and angular velocity commands $\mathbf{v}_{xy}^{\text{des}}$ and $\mathbf{w}_{z}^{\text{des}}$ that the robot must track. Central phases are used to guide leg motion toward a desired gait pattern.

The action space $\mathcal{A} \in \mathbb{R}^{10}$ corresponds to desired joint position offsets $\mathbf{a}_t = \Delta\mathbf{q}_t^{\text{des}}$ with respect to a default joint configuration $\mathbf{q}_0$, that are then converted to torques τ through a proportional-derivative (PD) controller operating at a higher frequency than the neural policy, with gains $(\mathbf{K}_p, \mathbf{K}_d)$:

$$\boldsymbol{\tau} = \mathbf{K}_p(\mathbf{q}_0 + \alpha\, \Delta\mathbf{q}^{\text{des}} - \mathbf{q}) - \mathbf{K}_d\, \dot{\mathbf{q}}. \tag{4}$$

Joints of the upper body that are not controlled by the policy target a constant default joint configuration. Fig. 2 highlights the flow of information in our pipeline.

C. Rewards and constraints

The complete list of rewards and constraints used in our experiments is provided in Table I. The main reward r_{vel} that defines our velocity tracking task is widely used in RL for legged locomotion [29], [14]. A single other term r_τ regularises the behaviour by penalizing joint torques.

We enforce a series of constraints to ensure that generated trajectories will be safe to transfer on the real robot once

training is complete. Like in our previous work, we distinguish two kinds of constraints to achieve a trade-off between exploration and constraint satisfaction. Hard constraints on joint limits c_{lower} and c_{upper} should be never violated and thus are assigned a $p_i^{\max}$ value of 1 in (3). On the contrary, the $p_i^{\max}$ of soft constraints progressively increases from 0.05 to 0.25 in a curriculum fashion. Having high $p_i^{\max}$ values for all constraints would lead to overly conservative exploration due to a high probability of episode termination upon constraint violations, which often happen during the initial learning phase. This could hamper the discovery of higher reward regions in the state-action space.

Beside safety, we enforce style constraints to guide learning towards natural-looking motions. Unlike [10] who enforced style constraints only on flat surfaces, we always keep them active. This distinction was unnecessary in our case due to the reduced relative size of obstacles. We drive the gait cycle with c_{gait} in a similar way than [13]. This helps us slow the gait down as it initially tended to be too fast in order to maximize stability and tracking performances. For simplicity we did not explore more refined gait schedules, such as the Von Mises distribution function used in [35]. A foot clearance constraint $c_{\text{clearance}}$ and a foot tilt one c_{tilt} are used to raise the apex of the foot swing trajectories and thus to maintain a sufficient clearance margin with respect to the considered obstacles. These two constraints follow a curriculum: they are not enabled for the first 20% of training to avoid hampering exploration, and are then progressively enabled with a full activation at 70% of the training. $h^{\text{apex min}}$ is also progressively raised from 10 cm to 25 cm.

CaT intends to reduce the burden of hyperparameters tuning and reward shaping, and as such we mostly kept the same hyperparameters than [10], reusing existing constraint settings for our new constraints and choosing the torque penalty weight so that it is an order of magnitude lower than the velocity tracking reward. Minimal tuning is required for constraints since their limits can be decided from their physical meaning based on the desired motion characteristics.

D. Symmetry loss

We promote motion that abide by the bilateral symmetry of the robot by adding the symmetry term proposed by [43] to the loss function:

$$\mathcal{L}_{\text{sym}} = \mathbb{E}_{(s,a)\sim\rho_\pi} \| \mu_\theta(\mathbf{s}) - M_a(\mu_\theta(M_s(\mathbf{s}))) \|_2, \qquad (5)$$

where $M_s : \mathcal{S} \mapsto \mathcal{S}$ and $M_a : \mathcal{A} \mapsto \mathcal{A}$ are functions that respectively mirror the state and action with respect to the sagittal plane, and μ_θ is the mean of our Gaussian policy π parameterized by θ. This auxiliary loss is optimized alongside the default PPO loss $\mathcal{L}_{\text{PPO}}$:

$$\mathcal{L} = \mathcal{L}_{\text{PPO}} + \beta\, \mathcal{L}_{\text{sym}}, \qquad (6)$$

where β is a scaling hyperparameter that tunes the prominence of the symmetry objective with respect to the PPO one. This approach was shown to be the most consistent among other methods for symmetric legged locomotion [44].

TABLE I: Rewards and constraints used in our experiments.

Task formulation	
Velocity tracking	$r_{\text{vel}} = e^{-\frac{\left\|v_{xy}^{\text{des}} - v_{xy}\right\|_2^2}{0.25}} + \frac{1}{2} e^{-\frac{\left\|\omega_z^{\text{des}} - \omega_z\right\|^2}{0.25}}$
Torque regularization	$r_\tau = \|\tau\|_2$
Hard constraints for safety ($\forall k \in 1..10$)	
Upper joint limits	$c_{\text{upper}_k} = q_k - q_{\text{upper}_k}^{\text{lim}}$
Lower joint limits	$c_{\text{lower}_k} = q_{\text{lower}_k}^{\text{lim}} - q_k$
Soft constraints for safety ($\forall k \in 1..10$)	
Torque limits	$c_{\text{torque}_k} = \lvert\tau_k\rvert - \tau_k^{\text{lim}}$
Joint velocity limits	$c_{\text{joint velocity}_k} = \lvert\dot{q}_k\rvert - \dot{q}_k^{\text{lim}}$
Action rate limits	$c_{\text{action rate}_k} = \frac{\left\lvert \Delta q_{t,k}^{\text{des}} - \Delta q_{t-1,k}^{\text{des}} \right\rvert}{dt} - \dot{q}_k^{\text{des lim}}$
Soft constraints for style ($\forall j \in 1..2$)	
Waist orientation	$c_{\text{ori}} = \|{}^w\mathbf{g}_{\text{xy}}\|_2 - \text{ori}^{\text{lim}}$
Waist minimum height	$c_{\text{h}} = h_b^{\text{min}} - h_b$
Gait	$c_{\text{gait}_j} = \mathbb{1}_j^{\text{ctc}} \cdot (\sin(\phi_j) > 0.1) + \neg\mathbb{1}_j^{\text{ctc}} \cdot (\sin(\phi_j) < -0.1)$
Foot clearance	$c_{\text{clearance}_j} = h_j^{\text{apex min}} - h_j^{\text{apex}}$
Foot tilt during swing	$c_{\text{tilt}_j} = \neg\mathbb{1}_j^{\text{ctc}} \cdot (\left\|{}^j\mathbf{g}_{\text{xy}}\right\|_2 - \text{tilt}^{\text{lim}})$

	Leg / Ankle / Arm
Torque τ^{lim}	150 / 40 / 50 Nm
Joint velocity $\dot{q}^{\text{lim}}$	16 / 5 / 5 rad/s
Action rate $\dot{q}^{\text{des, lim}}$	80 rad/s
Base orientation ori^{lim}	0.07 rad
Base min height h^{lim}	0.8 m
Feet clearance $h^{\text{apex min}}$	0.25 m
Feet tilt tilt^{lim}	0.07 rad

TABLE II: Privileged observations and the reward or constraint for which they provide crucial information. $\neg$ and $\mathbb{1}_j^{\text{ctc}}$ are respectively the logical NOT operator and the contact indicator of foot j.

Privileged observation	Size	Associated Reward/Cstr.
Waist linear velocity v_{xy}	$\mathbb{R}^3$	Velocity tracking r_{vel}
Randomized ground friction	$\mathbb{R}^1$	-
Height of the waist h_b	$\mathbb{R}^1$	Waist minimum height c_{h}
Contact forces of the feet	$\mathbb{R}^6$	Gait c_{gait_j} | Foot tilt c_{tilt_j}
Swinging time of the feet	$\mathbb{R}^2$	Gait c_{gait_j} | Foot tilt c_{tilt_j}
Height of the feet	$\mathbb{R}^2$	Foot clearance $c_{\text{clearance}_j}$
Apex height of last swing	$\mathbb{R}^2$	Foot clearance $c_{\text{clearance}_j}$

E. Learning features

We focus our study on a set of 6 optional features that can be implemented as part of a RL setup. They will be compared individually with respect to a baseline policy to assess how relevant they are for learning humanoid locomotion:

- **Bernoulli push occurrences:** Regularly pushing learning agents during training promotes the discovery of recovery behaviors for robustness to external disturbances. It can also be a way to alleviate the sim-to-real gap by forcing the robots out of their limit cycle so that they do not overfit an undisturbed walk in simulation. Such pushes can be based on the result of a Bernoulli trial for each agent individually at each iteration [18].
- **Privileged observations for the critic:** Various quantities are readily available when training in simulation,

such as the friction of the ground or the contact forces. These privileged observations provide rich information which can guide the learning process. One way to use those is to rely on a teacher-student scheme that will encode this information in a latent space for both actor and critic. During deployment, the encoded privileged information is estimated through an history of observations [27]. Providing privileged information to the critic only is lighter implementation-wise: the critic can still use it to better assess the value function, yet there is no need to reconstruct it for deployment [19], [21].

- **Unoisy observations for the critic:** In simulation, applying noise to the observations is a way to ease sim-to-real gap by accounting for imperfections in the sensors. If done only for the actor, then the critic has access to the ground truth, thus to cleaner signals from which to assess the value function [19], [20].
- **Adaptive / linear learning rate:** Rather than being constant the learning rate l_r is changed based on the current Kullback–Leibler divergence $D_{KL,t}$ to keep under control the evolution of the probability distribution:

$$l_{r,t} = \begin{cases} 2/3\, l_{r,t-1} & \text{if } D_{KL,t} > \quad 2\, D_{KL,threshold} \\ 3/2\, l_{r,t-1} & \text{if } D_{KL,t} < 1/2\, D_{KL,threshold} \end{cases} \quad (7)$$

  This approach was evoked in [16] without being tested against a fixed learning rate. It also differs from learning rate annealing which progressively reduces l_r over time and with which [26], [25] obtained better performances.
- **Gradient clipping:** For each update iteration in an epoch, PPO rescales the gradients of the networks so that the L^2 norm of the concatenated gradients of all parameters does not exceed a given value. Gradients are modified in-place. [26] found that such a clipping offered a small performance boost for their use cases.
- **Value function loss clipping:** PPO clips the value function in a similar way than what it does for the surrogate objective. With $\mathbf{V}$ the values, the value loss function $\mathcal{L}_\mathbf{V}$ changes from $(\mathbf{V}_{\theta_t} - \mathbf{V}_{\text{target}})^2$ to:

$$\mathcal{L}_\mathbf{V} = \max\left[(\mathbf{V}_{\theta_t} - \mathbf{V}_{\text{target}})^2, (\mathbf{V}_{\text{clip}} - \mathbf{V}_{\text{target}})^2\right] \quad (8)$$

$$\mathbf{V}_{\text{clip}} = \text{clip}\left(\mathbf{V}_{\theta_t}, \mathbf{V}_{\theta_{t-1}} - \epsilon, \mathbf{V}_{\theta_{t-1}} + \epsilon\right) \quad (9)$$

  Recent results indicate that value function loss clipping either has no effect on training performances [25] or may even negatively impact them [26].

When providing privileged observations to the critic its observation vector goes from $\mathbb{R}^{43}$ up to $\mathbb{R}^{60}$. These observations are described in Table II. Most of them have strong ties either with a reward or a constraint, and as such provide clear learning signals for the critic to better understand the state of the system and react accordingly to track the reference velocity while avoiding constraint violations.

V. Ablation Study

A. Experimental setup

To train our policies, we leverage the PPO algorithm [16] using the implementation from rl-games [45], slightly modified in our previous work [10] to use constraints as terminations, alongside massively parallel simulation of Isaac Gym [17]. Hyperparameters are mostly left unchanged to highlight the portability of the approach to a new robotic platform. Policies are trained for 3000 epochs both on flat terrain and for the challenging terrain curriculum. These two cases respectively amounts to 1 hour and 1.5 hours of training on a single RTX 4090 GPU.

Except for CaT specific implementations, the resulting training procedure is similar to [29]. We use a model of the H1 humanoid robot by Unitree. The policy runs at 50 Hz and send target joint positions to the low-level PD controller running at 500 Hz alongside the simulation. The agents undergo a terrain curriculum and move to increasingly difficult terrains as they progress. They are spread over 4 kinds of terrains: flat, a terrain deformed with Perlin noise, blocks spread evenly in a checker pattern and blocks randomly spreads. Domain randomization affects the mass of the torso, the position of its center of mass, the friction of the ground, the strength of the motors and the $(\mathbf{K}_p, \mathbf{K}_d)$ gains. We evaluate the velocity tracking performance of a baseline policy with none of the features described in Section IV-E, as well as policies with each feature enabled individually. The velocity command is set to 0.6 m/s forwards.

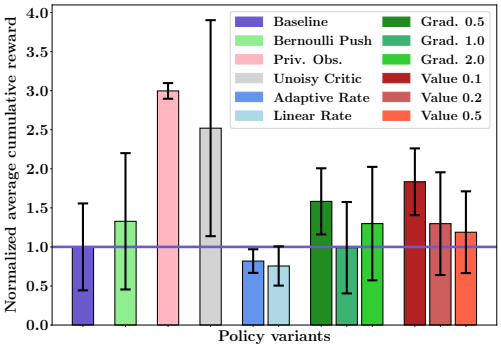

Fig. 3: Evaluation and comparison of the total cumulative reward for the different learning setups.

B. Results and Analysis

Fig. 3 reports the normalized average of the cumulative reward over 250 trials randomly done on the 4 kinds of terrains at medium difficulty. For each trial, a robot is spawned and its cumulative reward is gathered till the end of the episode (either a timeout or a fall). 5 seeds are ran for each case and we present the mean and standard deviation. Moreover, Fig. 4 reports the linear velocity tracking reward, that is the left part of r_{vel} in Table I. The mean and standard deviation are plotted for each case over the 5 seeds, using the same color code as Fig. 3. The performance ranking of the features is roughly the same for both graphs. Perfect consistency is not to be expected as they do not display the same quantities: while Fig. 4 reports the raw linear velocity tracking reward, Fig. 3 rather presents the total reward, that is the sum of linear and angular tracking rewards combined with the torque regularization, and down-scaled by the constraint violations. Besides, the former one highlights evaluation results whereas the later one discloses data gathered during training. All

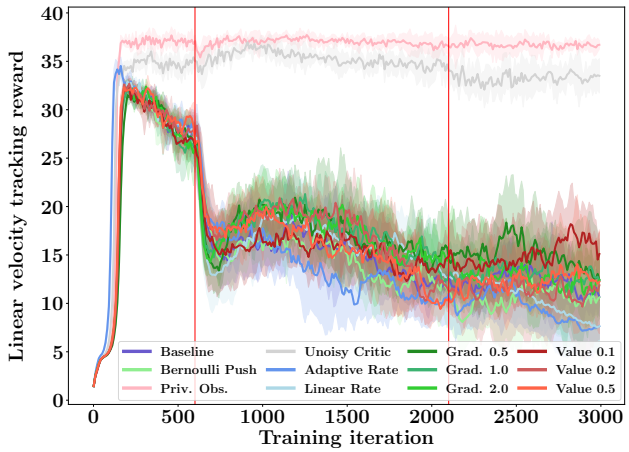

Fig. 4: Comparison of the tracking reward during training between the different learning setups. The vertical red lines indicate the start and end of the curriculum on the foot clearance and tilt constraints.

policies achieve balance and manage to walk around, but with varying degrees of success.

Learning rate schedulers: Of all compared features, only the learning rate schedulers appear to be detrimental to performances. Although it seems to perform the same as the others before the foot clearance and tilt are enabled, the tracking reward falls off more than the others afterwards. The reward drop can be explained by the switch from a situation where the robot can freely move its feet to a situation where feet must reach a minimum apex height. Since this constraint is suddenly enabled and violated, it down-scales the overall reward through δ as in (2), except for the two best performing cases which adapt on the fly. We hypothesize that these schedulers struggle the most afterwards because of the decrease of the learning rate which limits the rate at which the policy can adapt. In fact, while the baseline has a constant learning rate of 3e-4, the linear scheduler keeps decreasing from that value, and the adaptive one can go down to 1e-4 depending on the Kullback–Leibler divergence in (7).

Gradient and value clipping: Clipping the norm of the global gradient and clipping the loss of the value function seems to have a slight beneficial effect, especially for the stronger clippings with a maximum gradient norm of 0.5 and a loss clipping at 0.1. In practice, gradient clipping amounts to clipping the error derivatives when computing the backward propagation through the neural network. Clipping the loss of the value function limits the maximum impact the value prediction error can have in the overall loss function. We think these two procedures might have a stabilizing effect on the learning process by keeping under control the magnitude of the involved quantities.

Bernoulli push occurrences: It appears applying push disturbances to the agents based on the result of individual Bernoulli trials is slightly beneficial compared to pushing all of them at once with a fixed schedule. Again, we assume the increase in performances is linked to a stabilizing effect of these trials that better spread disturbances in the observation buffer. With the baseline method, for most iterations this buffer will be filled with samples of undisturbed locomotion, then will suddenly be filled only with samples of robots

TABLE III: Rewards and additional constraint with respect to Table I used for the loco-manipulation.

Task formulation	
Left hand proximity	$r_{L\to T} = \mathbb{1}_{L\to T} \cdot (0.3 - \lVert \mathbf{v}_{L\to B} \rVert_2)$
Right hand proximity	$r_{R\to T} = \mathbb{1}_{R\to T} \cdot (0.3 - \lVert \mathbf{v}_{R\to B} \rVert_2)$
Carry box to target	$r_T = 2.0 - \lVert \mathbf{v}_{B\to T} \rVert_2$
Drop box at target	$r_{B\to T} = \mathbb{1}_{B\to T} \cdot (e^{-\frac{\sum F_{L/R}}{100}} + e^{-\frac{\lVert \mathbf{q}_{B\to T} - \mathbf{q} \rVert_2}{0.01}})$
Waist heading to target	$r_\psi = e^{-\frac{\psi_T^2}{0.5}}$
Exploration	$\frac{v_{\text{xy}}}{\lVert v_{\text{xy}} \rVert_2} \cdot \frac{\mathbf{v}_{B\to T,xy}}{\lVert \mathbf{v}_{B\to T,xy} \rVert_2}$
Soft constraints	
Box dropping	$c_{B\to T} = \mathbb{1}_{B\to T} \cdot F_{L/R}$

trying to recover. This creates a cycle where the policy focuses fully on walking, then fully on recovering, which might not be optimal from a learning perspective.

Privileged and unoisy observations for the critic: Providing observations with no noise to the critic has a notable beneficial effect, albeit with an important variance. Besides, providing to the critic privileged information from the simulation achieve best performances, with both the highest average and the lowest result variance. We hypothesize that these two features enable the critic to access learning signals with improved quality, which makes it easier to assess the value function and thus the critic can better guide the actor. Privileged information go even one step further by bringing data that have direct ties to the constraints, such as the status of the feet for the gait, tilt and clearance constraints.

These results should not be considered as a strict benchmark since that would require a more in-depth ablation study in various scenarios. This comparison ought to serve as an indicator for similar legged locomotion tasks. Overall, considering the number of seeds and the standard deviations observed in Fig. 3 and Fig. 4, providing privileged and perfect observations to the critic seems to have a clear beneficial effects. The impacts of other learning features are more muddy and up to variations depending on the scenario.

VI. Loco-manipulation

We extend our RL scheme to humanoid loco-manipulation for transporting a package in simulation. We seek to move a cubic box with side length of 25 cm from its starting location atop a 90 cm pillar. For simplicity, the robot starts in a position for which the package is placed between its hands. This eases the pick-up procedure compared to works like [38], [39] where the robot has to get into position first. The goal is to release the package atop another 90 cm pillar placed randomly in a half circle behind the robot with a $1.1m$ radius. The velocity tracking reward r_{vel} is removed but the rest of Table I is retained. The velocity command is replaced by the box to target location $\mathbf{v}_{B\to T}$ vector, the left hand to box $\mathbf{v}_{L\to B}$ and right hand to box $\mathbf{v}_{R\to B}$ vectors, the waist heading with respect to the box ψ_B, and the waist heading with respect to the target location ψ_T, all expressed in waist frame. We provide privileged information to the

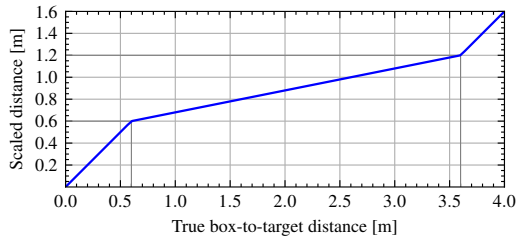

Fig. 5: The box-to-target distance is scaled down to fall back to the $[0.0, 1.6]m$ range seen during training. We keep a 1:1 scaling for the pick-up and drop-off parts of the motion since they might more sensitive to the distance. Here, for a target initially at a 4m distance, we flatten the distance between $[0.6, 3.6]m$ to $[0.6, 1.2]m$ such that the whole scaled range is $1.6m$.

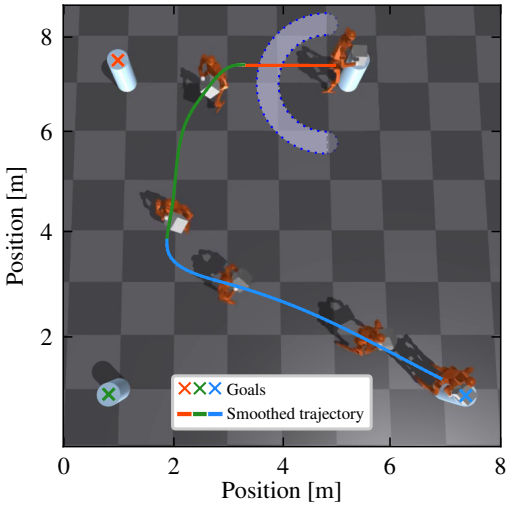

Fig. 6: Long distance box transport can be achieved by changing the position of the target during walking. The color-coded smoothed trajectory highlights the robot motion while aiming for the goal of the same color. The color discontinuities indicate the moments when the target was changed. As a reference, the light blue half-circle in the upper part represents the $1.1m$ range in which the target is randomly spawned during training.

critic and augment the information contained in Table II with an indicator of proximity between the box and the target $\mathbb{1}_{B\to T}$, as well as the norm of the contact forces applied by each hand $F_{L/R}$. The complete list of additional rewards and constraints used for the loco-manipulation is provided in Table III. Some of theses rewards take inspiration from [46]. The indicator $\mathbb{1}_{L/R\to T}$ is equal to 0 when hands are more than 30 cm from the box, +1 otherwise during the pick-up and transport, or -1 once the box has been dropped at the target location. $\mathbf{q}_{B\to T}$ is similar to the default configuration but with the arms spread further apart to encourage the box dropping. Ultimately, $\mathcal{S} \in \mathbb{R}^{66}$ ($\mathbb{R}^{83}$ with privileged information) and $\mathcal{A} \in \mathbb{R}^{14}$ as we include the shoulder yaw joints and elbow joints to the action space.

Due to the complexity of the manipulation sequence, this task requires more reward engineering compared to the velocity tracking one. The expected sequence of motion needs to be tightly specified to guide the exploration process. This part might be simplified by leveraging reference trajectories or motion from trajectory optimization or reference datasets.

As shown in Fig. 1, the robot achieves the desired sequence of motion by grabbing the box placed between its hands, turning around and walking to the target location before dropping the box and moving backwards.

This framework can be extended to long distance transport by scaling down the box-to-target location $\mathbf{v}_{B\to T}$ vector to fall back to the range for which the robot was trained. This way we avoid providing out-of-distribution observations and the robot behaves as if it was in the nominal scenario. Such a scaling is illustrated in Fig. 5. The target location can also be changed online to control the robot trajectory by setting intermediate goals, as in Fig 6. This could be leveraged by a high-level planner to achieve a specific pathing around obstacles or to navigate to targets that cannot be reached with a straight line. Finally, this policy trained for a square box of 25 cm achieves mild generalization for the package size, going down to a side length of 15 cm and up to 32 cm, above which to robot struggles to release the package at the target position due to not opening the arms wide enough.

Several demonstrations are in the supplementary video.

VII. Conclusion

In this study, we transfer *Constraints as Termination* to a humanoid platform for the first time. We shape its behaviour in a minimalist fashion as many of the terms usually implemented as rewards can be more effectively and intuitively formulated as constraints. We propose a reinforcement learning architecture to achieve a velocity tracking task on rough terrain with the H1 humanoid robot in simulation. Then, we compare in this framework the performance impact of several learning features encountered in the literature. Our approach successfully extends to a loco-manipulation scenario in simulation involving the transport of a small package, which highlights its potential for controlling both upper and lower limbs with a single policy. Future work could focus on transferring these policies to a real H1 robot, but also on pushing the loco-manipulation further with a full locate-pick-drop cycle and more variety for the environment, the package and the task settings.

Acknowledgements

This work was partially supported by the Japan Society for the Promotion of Science (JSPS) Postdoctoral Fellowships for Research in Japan and by the JSPS KAKENHI grants number JP22H05002 and JP24KF0125.

References

[1] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," in *Conference on Robot Learning (CoRL)*, 2023.

[2] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[3] M. Tranzatto, T. Miki, M. Dharmadhikari, L. Bernreiter, M. Kulkarni, F. Mascarich, O. Andersson, S. Khattak, M. Hutter, R. Siegwart, *et al.*, "Cerberus in the darpa subterranean challenge," *Science Robotics*, vol. 7, no. 66, p. eabp9742, 2022.

[4] N. Rudin, H. Kolvenbach, V. Tsounis, and M. Hutter, "Cat-like jumping and landing of legged robots in low gravity using deep reinforcement learning," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 317–328, 2021.

[5] S. Ha, J. Lee, M. van de Panne, Z. Xie, W. Yu, and M. Khadiv, "Learning-based legged locomotion: State of the art and future perspectives," *The International Journal of Robotics Research*, p. 02783649241312698, 2024.

[6] Z. Zhuang, S. Yao, and H. Zhao, "Humanoid parkour learning," *arXiv preprint arXiv:2406.10759*, 2024.

[7] L. Bao, J. Humphreys, T. Peng, and C. Zhou, "Deep reinforcement learning for bipedal locomotion: A brief survey," *arXiv preprint arXiv:2404.17070*, 2024.

[8] Unitree. (2024) Quadruped products. [Online]. Available: https://shop.unitree.com/collections/frontpage

[9] Y. Xie, B. Lou, A. Xie, and D. Zhang, "A review: Robust locomotion for biped humanoid robots," in *Journal of Physics: Conference Series*, vol. 1487, no. 1. IOP Publishing, 2020, p. 012048.

[10] E. Chane-Sane, P.-A. Leziart, T. Flayols, O. Stasse, P. Souères, and N. Mansard, "Cat: Constraints as terminations for legged locomotion reinforcement learning," 2024.

[11] F. Risbourg, T. Corbères, P.-A. Léziart, T. Flayols, N. Mansard, and S. Tonneau, "Real-time footstep planning and control of the solo quadruped robot in 3d environments," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 950–12 956.

[12] P.-A. Léziart, T. Corbères, T. Flayols, S. Tonneau, N. Mansard, and P. Souères, "Improved control scheme for the solo quadruped and experimental comparison of model predictive controllers," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9945–9952, 2022.

[13] Y. Kim, H. Oh, J. Lee, J. Choi, G. Ji, M. Jung, D. Youm, and J. Hwangbo, "Not only rewards but also constraints: Applications on legged robot locomotion," *IEEE Transactions on Robotics*, vol. 40, pp. 2984–3003, 2024.

[14] J. Lee, L. Schroth, V. Klemm, M. Bjelonic, A. Reske, and M. Hutter, "Evaluation of constrained reinforcement learning algorithms for legged locomotion," *arXiv preprint arXiv:2309.15430*, 2023.

[15] S. Huang, R. F. J. Dossa, A. Raffin, A. Kanervisto, and W. Wang, "The 37 implementation details of proximal policy optimization," in *ICLR Blog Track*, 2022, https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/. [Online]. Available: https://iclr-blog-track.github.io/2022/03/25/ppo-implementation-details/

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

[17] 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.

[18] H. Hu, K. Zhang, A. H. Tan, M. Ruan, C. Agia, and G. Nejat, "A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6569–6576, 2021.

[19] D. Youm, H. Jung, H. Kim, J. Hwangbo, H.-W. Park, and S. Ha, "Imitating and finetuning model predictive control for robust and symmetric quadrupedal locomotion," *IEEE Robotics and Automation Letters*, 2023.

[20] J. Eschmann, D. Albani, and G. Loianno, "Learning to fly in seconds," *IEEE Robotics and Automation Letters*, 2024.

[21] S. Chamorro, V. Klemm, M. d. L. I. Valls, C. Pal, and R. Siegwart, "Reinforcement learning for blind stair climbing with legged and wheeled-legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 8081–8087.

[22] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: a comprehensive survey," *Artificial Intelligence Review*, vol. 55, no. 2, pp. 945–990, 2022.

[23] 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. [Online]. Available: http://jmlr.org/papers/v22/20-1364.html

[24] S. Huang, R. F. J. Dossa, C. Ye, J. Braga, D. Chakraborty, K. Mehta, and J. G. Araújo, "Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms," *Journal of Machine Learning Research*, vol. 23, no. 274, pp. 1–18, 2022. [Online]. Available: http://jmlr.org/papers/v23/21-1342.html

[25] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep policy gradients: A case study on ppo and trpo," in *International Conference on Learning Representations*, 2020.

[26] M. Andrychowicz, A. Raichuk, P. Stańczyk, M. Orsini, S. Girgin, R. Marinier, L. Hussenot, M. Geist, O. Pietquin, M. Michalski, *et al.*, "What matters in on-policy reinforcement learning? a large-scale empirical study," in *ICLR 2021-Ninth International Conference on Learning Representations*, 2021.

[27] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[28] M. Aractingi, P.-A. Léziart, T. Flayols, J. Perez, T. Silander, and P. Souères, "A hierarchical scheme for adapting learned quadruped locomotion," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

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

[30] G. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," in *Robotics: Science and Systems*, 2022.

[31] M. Aractingi, P.-A. Léziart, T. Flayols, J. Perez, T. Silander, and P. Souères, "Controlling the solo12 quadruped robot with deep reinforcement learning," *scientific Reports*, vol. 13, no. 1, p. 11945, 2023.

[32] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 5026–5033.

[33] A. Gams, T. Petrič, B. Nemec, and A. Ude, "Manipulation learning on humanoid robots," *Current Robotics Reports*, vol. 3, no. 3, pp. 97–109, 2022.

[34] A. Tang, T. Hiraoka, N. Hiraoka, F. Shi, K. Kawaharazuka, K. Kojima, K. Okada, and M. Inaba, "Humanmimic: Learning natural locomotion and transitions for humanoid robot via wasserstein adversarial imitation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 13 107–13 114.

[35] Q. Zhang, P. Cui, D. Yan, J. Sun, Y. Duan, G. Han, W. Zhao, W. Zhang, Y. Guo, A. Zhang, *et al.*, "Whole-body humanoid robot locomotion with human reference," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2024, pp. 11 225–11 231.

[36] M. Seo, S. Han, K. Sim, S. H. Bang, C. Gonzalez, L. Sentis, and Y. Zhu, "Deep imitation learning for humanoid loco-manipulation through human teleoperation," in *2023 IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids)*. IEEE, 2023, pp. 1–8.

[37] J. Liu, H. Sim, C. Li, K. C. Tan, and F. Chen, "Birp: Learning robot generalized bimanual coordination using relative parameterization method on human demonstration," in *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE, 2023, pp. 8300–8305.

[38] Z. Xie, J. Tseng, S. Starke, M. van de Panne, and C. K. Liu, "Hierarchical planning and control for box loco-manipulation," *Proceedings of the ACM on Computer Graphics and Interactive Techniques*, vol. 6, no. 3, pp. 1–18, 2023.

[39] J. Dao, H. Duan, and A. Fern, "Sim-to-real learning for humanoid box loco-manipulation," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 16 930–16 936.

[40] R. Bellman, "A markovian decision process," *Journal of mathematics and mechanics*, pp. 679–684, 1957.

[41] P.-A. Léziart, T. Flayols, F. Grimminger, N. Mansard, and P. Souères, "Implementation of a reactive walking controller for the new open-hardware quadruped solo-12," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 5007–5013.

[42] Z. Xie, X. Da, M. Van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4955–4961.

[43] W. Yu, G. Turk, and C. K. Liu, "Learning symmetric and low-energy locomotion," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–12, 2018.

[44] F. Abdolhosseini, H. Y. Ling, Z. Xie, X. B. Peng, and M. Van de Panne, "On learning symmetric locomotion," in *Proceedings of the 12th ACM SIGGRAPH Conference on Motion, Interaction and Games*, 2019, pp. 1–10.

[45] D. Makoviichuk and V. Makoviychuk, "rl-games: A high-performance framework for reinforcement learning," https://github.com/Denys88/rl_games, May 2021.

[46] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2497–2503.