Deep Compliant Control for Legged Robots

Adrian Hartmann, Dongho Kang, Fatemeh Zargarbashi, Miguel Zamora, and Stelian Coros

***Abstract*— Control policies trained using deep reinforcement learning often generate stiff, high-frequency motions in response to unexpected disturbances. To promote more natural and compliant balance recovery strategies, we propose a simple modification to the typical reinforcement learning training process. Our key insight is that stiff responses to perturbations are due to an agent's incentive to maximize task rewards at all times, even as perturbations are being applied. As an alternative, we introduce an explicit recovery stage where tracking rewards are given irrespective of the motions generated by the control policy. This allows agents a chance to gradually recover from disturbances before attempting to carry out their main tasks. Through an in-depth analysis, we highlight both the compliant nature of the resulting control policies, as well as the benefits that compliance brings to legged locomotion. In our simulation and hardware experiments, the compliant policy achieves more robust, energy-efficient, and safe interactions with the environment.**

I. Introduction

Legged animals effectively mitigate environmental uncertainties through behaviors characterized by *compliance*. Navigating uneven terrains, they instinctively adjust their speed and direction to accommodate the variations in landscape, guaranteeing a level of stability and adaptability. Similarly, while leading a dog on a leash, we observe this compliant behavior in action. A gentle tug can subtly guide the dog, indicating a new course of movement.

Achieving this compliant behavior is crucial for legged locomotion as it enables natural movement patterns, facilitates adaptation to various environmental uncertainties, and reduces energetic costs [1]. However, in robotics, the focus on precision and functionality often overshadows this importance. While the traditional *"stiffer is better"* rule may improve precision under disturbances, it comes with costs such as increased energy or torque demand. Additionally, unexpected interactions like colliding with an obstacle in such a system can generate sharp and high-frequency motions, potentially harming the environment and the robot.

In this work, we present a deep reinforcement learning (DRL) approach to create a quadrupedal locomotion controller that exhibits compliant behaviors in the presence of external disturbances. It is inspired by the fact that the priorities of different objectives, such as precision or efficiency, change depending on the environment [2]. Given external interference, the associated rewards should consequently adapt to permit temporary deviations from the task objective. To this end, our training strategy is enhanced with a *recovery stage* activated by a randomly sampled disturbance. During this stage, the focus moves away from strict adherence to tracking accuracy, towards an emphasis on energy efficiency.

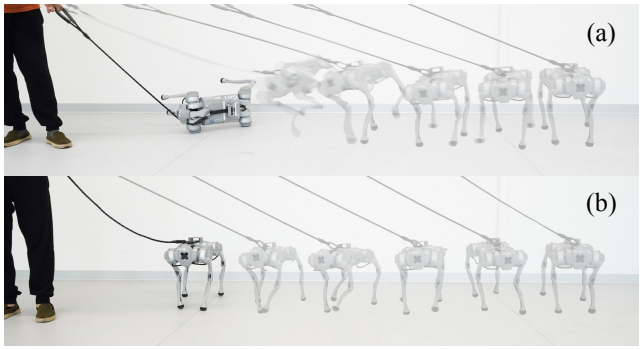


Fig. 1: The snapshots of *Unitree Go1* robot displaying distinct responses between the baseline policy (fig. 1a) and the compliant policy (fig. 1b) when subjected to a pull via a leash. The compliant version exhibits a gentler reaction to external disturbances and manages diverse perturbations more effectively. A detailed quantitative analysis can be found in Section IV.

In our extensive simulation and hardware experiments, the resulting policy could adapt to its surroundings by deviating from its given commands when necessary. This leads to smooth recoveries from pushes and minimizes the forces applied to the environment in the event of collisions. Moreover, this approach reduces energy and torque usage in response to disturbances and enhances locomotion robustness, enabling navigation across complex terrains. Deployed on the *Unitree Go1* robot, we validated this durability in natural settings.

In summary, the main contributions of this paper include:

- Introduction of a training procedure for a legged locomotion controller that exhibits compliant interactions with the environment.
- Exploring the impact of compliant behavior on robustness and energy efficiency during perturbations through simulations and hardware tests.

II. Related Work

Over recent years, reinforcement learning (RL) has achieved significant milestones in the control of diverse legged mechanisms, ranging from hopping robots [3], to bipedal [4] and quadrupedal [5] walkers. While these methodologies exhibit admirable fidelity in tracking user-defined goals and carry out robust and sophisticated maneuvers, Xie et al. [6] observed that many controllers based on RL exhibit motion patterns that appear more rigid compared to the responses of animals or humans when subjected to disturbances. Such stiffness can often be attributed to

The authors are with the Computational Robotics Lab in the Department of Computer Science at ETH Zurich, Switzerland. {haadrian, kangd, fzargarbashi, mimora, scoros}@ethz.ch

This work has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 866480).

high PD gains, which are indispensable for facilitating dynamic movements. In response, Lee et al. [7] proposed a compliance-induced motion controller that modulates a physically simulated character's stiffness in the applications of character animation. Their approach enables the controller to mitigate impact forces when interacting with the environment, thus yields a compliant control resulting in fluid and natural motions, both in passive responses to external perturbations and active engagements, such as door operations. While our objectives align closely with their research, the implementation of their method on robot hardware remains challenging because it is dependent on interactive force measurements, which are typically unavailable in real world.

In the field of legged robotics, particularly concerning the real-world deployment of RL policies, several studies [5, 8, 9] emphasize the benefits of employing moderate joint feedback gains. This strategy is empirically recognized as advantageous for enhancing the sim-to-real transition and for minimizing potential harm to hardware, especially when positioning the robot on irregular terrains. Beyond merely adjusting joint gains, an intriguing development is presented by Gangapurwala et al. [10], where a locomotion controller is trained at a notably low frequency which encourages less aggressive movement while maintaining a commendable degree of resilience against environmental uncertainties.

Building on the strategy of employing modest joint feedback gains, we further explore the compliant responses of DRL policies to external disturbances or uncertainties in the environment, drawing inspiration from the prior studies by Lee et al. [7] and Gangapurwala et al. [10]. Our training process is enhanced by introducing random disturbances to the robot, followed by a recovery stage that induces a tempered response to perturbations. As a result, the resulting policies exhibit less aggressive response to disturbances and achieve better energy efficiency.

Our approach presents potential advantages in developing systems geared towards safe interactions between humans and robots or between the environment and robots. This is particularly relevant for applications such as assisting visually impaired people [11], fostering human-robot collaboration [12], and navigating through confined or densely-packed environments [13].

III. Deep Compliant Control

We employ DRL to train a control policy that exhibits compliant behaviors to external disturbances. To instill these behaviors, the policy must identify disturbances solely through proprioceptive sensor inputs since force measurements are inaccessible. Consequently, our observation encompasses both the current and previous states of the robot. This leads us to define an RL task that aims to map a sequence of historical robot states with respective joint commands. Furthermore, we enhance the standard RL training approach by introducing a multi-stage episodic RL training that incorporates a recovery stage. During this stage, the robot prioritizes stabilization and balance recovery over strictly following a velocity command.

TABLE I: Overview of the reduced observation vector $\hat{o}$. We use the cartesian coordinate system with the x-axis for the left, the y-axis for the upward, and z-axis for the forward direction.

Names	Symbols	Units
Base height	y	m
Local gravity	g_x, g_y, g_z	m/s^2
Joint angles	$q_{0\ldots11}$	rad
Local velocity	v_x, v_y, v_z	m/s
Local angular velocity	$\omega_x, \omega_y, \omega_z$	rad/s
Joint rates	$\dot{q}_{0\ldots11}$	rad/s

A. Observation and Action Spaces

The classical approach to estimate external force acting on a robot is to monitor the time variation of proprioceptive sensor observations [14, 15]. For example momentum-based observers, as proposed by Morlando et al. [15], are able to estimate external forces from joint angles $\mathbf{q}$, joint rates $\dot{\mathbf{q}}$, joint torques $\boldsymbol{\tau}$ and ground reaction forces $\mathbf{f}_{gr}$. Inspired by this, we include the current and previous joint angles, rates and the latest inputs of the PD controller in the observation vector. Since ground reaction forces are not easily accessible, we excluded them from our observations. This approach configures the system as a Partially Observable Markov Decision Process (POMDP), allowing our policy to learn how to identify external disturbances.

We define the reduced observation vector $\hat{\mathbf{o}}_\mathbf{t}$ (Table I) which contains the base height, gravity vector, linear and angular velocity in the local body frame as well as angles and rates of all twelve rotational joints. At time t, $\hat{\mathbf{o}}_t$ represents the most recent measurements of the robot's state.

To encode the current state of the PD controllers, we use the previous reduced observation at time $t-1$ as well as the corresponding joint action commands $\mathbf{a}_{t-1}$.

Finally, we want to pass task commands to the agent that define the desired body velocities in longitudinal direction v_z^*, sideways v_x^* as well as yaw rate ω_y^*. Together, we get an observation vector $\mathbf{o}_t = \{\hat{\mathbf{o}}_{t-1}, \mathbf{a}_{t-1}, \hat{\mathbf{o}}_t, v_x^*, v_z^*, \omega_y^*\} \in \mathbb{R}^{83}$.

The action of the control policy is joint target angles which are passed to the low-level PD controller.

B. Reward

We adapt the reward structure proposed by Ji et al. [16] that splits reward terms in a positive r_{pos} and a negative r_{neg} term and define the total reward r as

$$r_{pos} = \max(0, w_{lin}\, r_{lin} + w_{ang}\, r_{ang} + w_h\, r_h) \quad (1)$$

$$r_{neg} = w_e\, r_e + w_\tau\, r_\tau + w_{pose}\, r_{pose} + w_{cl}\, r_{cl} \quad (2)$$

$$r = r_{pos} \cdot \exp(0.45 \cdot r_{neg}) \quad (3)$$

This structure guarantees a positive reward, preventing the policy from opting for early termination.

The mathematical expressions of each subreward term and their corresponding weights are detailed in Table II. The terms r_{lin} and r_{ang} align the robot's velocity with the desired velocity by penalizing deviations. The terms r_e and r_τ impose penalties based on the instantaneous mechanical power of each joint, considering both its absolute value

TABLE II: Reward terms and corresponding weights.

Name	Function	Weight
Linear velocity r_{lin}	$\exp(-8\,((v_x - v_x^*)^2 + (v_z - v_z^*)^2))$	0.8
Angular velocity r_{ang}	$\exp(-8\,(\omega_z - \omega_z^*)^2)$	0.5
Energy r_e	$\lvert\boldsymbol{\tau}\cdot\dot{\mathbf{q}}\rvert$	-0.015
Torque r_τ	$\lVert\boldsymbol{\tau}\rVert_2^2$	-0.0015
Pose r_{pose}	$\phi^2 + \psi^2 + 10\cdot(y - y^{\text{des}})^2$	-2.0
Foot clearance r_{cl}	$\sum_{i=0\ldots3} \lVert\mathbf{v}_{i,xz}\rVert_2^2 \cdot (p_{f_i,y} - p_{f_i,y}^{\text{des}})^2$	-0.1
Foot height r_h	$\sum_{i=0\ldots3} (p_{f_i,y}^{\text{peak}}/p_{f_i,y}^{\text{des}} - 1)^2$	-0.7

and the actuator losses proportionate to the applied torque, expressed as $P \sim \sum_i |\tau_i \cdot \dot{q}_i| + \alpha \cdot \tau_i^2$, where α is a certain coefficient. This approach is influenced by the work of Fu et al. [17], highlighting the important role of energy efficiency in generating natural locomotion patterns. The term r_{pose} aligns the robot base relative to the ground plane at zero pitch ϕ, zero roll ψ and a base height of $y_{\text{des}} = 0.31\,\text{m}$. Finally, the terms r_{cl} and r_h encourage the policy to take larger strides. The dense reward r_{cl} promotes larger swing height. Meanwhile, the sparse reward r_h penalizes deviations between the actual peak foot height during the swing and the target value. This penalty is applied once the foot makes contact with the ground. The desired step height is set at $p_{f_i,y}^{\text{des}} = 0.08\,\text{m}$.

C. Multi-stage episodic RL training

Starting from the same initial state, the robot has a designated $4.0\,\text{s}$ time slot to learn walking with a desired velocity randomly sampled in the ranges of $[-1.0, 1.0]\,\text{m/s}$, $[-0.5, 0.5]\,\text{m/s}$ and $[-0.5, 0.5]\,\text{rad/s}$ for forward, sideways, and turning speed, respectively.

Inspired by a recent study proposing the division of a training episode into multiple stages to tackle the explore-vs-exploit dilemma [18], we divide the training episode into three distinct stages: *walking*, *recovery*, and *post-recovery*. This segmentation recognizes that objectives shift in priority based on the environmental context. Emphasizing compliance, we reduce the focus on tracking accuracy after a push, instead prioritizing energy efficiency and smooth recovery. Consequently, we adjust the *recovery stage* rewards to support this focus. More details on each stage are as follows.

1) Walking stage: First, we train the robot to walk undisturbed for a duration of $2.0\,\text{s}$. Throughout this stage, the agent receives the full reward, as described in Section III-B.

2) Recovery stage: In this stage, we introduce a disturbance impulse by applying a velocity offset to the robot's base, simulating a push in the horizontal direction. The velocity offset for both forward and sideward directions is uniformly sampled up to a maximum magnitude of $1.0\,\text{m/s}$. Additionally, a rotational push with a maximum magnitude of $1.0\,\text{rad/s}$ is applied in all three axes.

After applying the impulse, the agent learns to act compliantly while recovering from the push. We give it a time frame of $1.0\,\text{s}$ to get back to the steady state of walking in which it does not get a penalty for deviating from the given motion command. More specifically, we exchange the reward r_{lin} and r_{ang} by constant values that represents the average rewards during the walking stage. The duration of this stage is picked empirically based on the assumption that is must be long enough to enable a smooth recovery but short enough to not unnecessarily extend training duration.

3) Post-recovery stage: Finally, in the third stage, we revert to the full reward as in the walking stage, to ensure the robot resumes regular walking post-push recovery. Spanning a duration of $1.0\,\text{s}$, this stage aims to prevent any decline in walking performance subsequent to a disturbance.

TABLE III: PPO hyperparameters.

Batch Size	8192	GAE discount factor	0.95
Mini-batch size	512	Desired KL-divergence	0.02
Number of epochs	30	Learning rate	adaptive
Clip range	0.2	Initial std. deviation	$\exp(-1)$
Entropy coeff.	0.01	Policy & VF layers	[256, 128, 64]
Discount factor	0.99	Activation Function	ELU

D. Other Training Details

In our training environment, we simulate a robot on a flat surface but introduce variable slopes by adjusting the gravity vector that peaks at a 10% incline. It is resampled uniformly at the start of an episode. This measure enables the RL policies to navigate across sloping or irregular terrains.

To ensure efficient training, we take additional measures. First, we early terminate an episode if the robot collapses, which is determined when the base height $y < 0.2\,\text{m}$, or if any part of the robot, apart from its feet, comes into contact with the ground. Second, an adaptive push curriculum is designed in consideration of the agent's limitations in walking and recovery abilities at the early phase of training. During the early phase, we assess the agent's walking performance, skipping the recovery and post-recovery stages if it doesn't meet the mark. Specifically, we take the average of the r_{lin} reward and check if it has exceeded 85% of its maximal value. This strategy is designed to guarantee the agent effectively masters the stepping motion before it is exposed to external perturbations. It's essential to emphasize that initiating pushes too early can prompt an excessive caution in the agent, possibly steering it into an unfavorable local optimum because of the steep penalties associated with collapses. Third, we progressively increase the maximum desired velocity command, reaching up to $1.0\,\text{m/s}$. This approach ensures a gradual challenge escalation for the agent, allowing it to adapt more effectively.

Additionally, to prevent large KL-divergence during the training process, we utilize an algorithm that adaptively adjusts the learning rate based on the current training progress, as proposed by Rudin et al. [19].

IV. Results

We demonstrate the efficacy of our approach through simulation and hardware experiments. For the experiments, we trained policies for the quadruped robot *Unitree Go1*, utilizing data generated by a physical simulation tool based

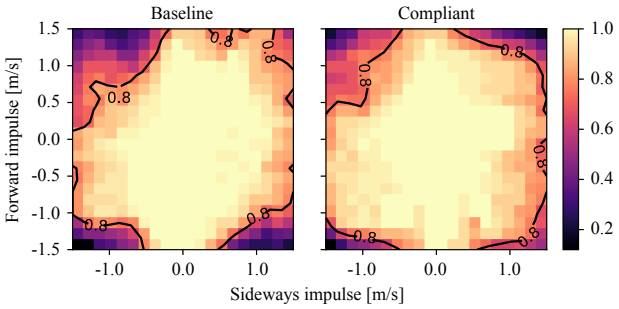


Fig. 2: Heatmaps depicting the push recovery success rates for both the baseline and the compliant controller. A contour line indicates the 80% success threshold

on Open Dynamics Engine [20], operating at a simulation rate of 240 Hz. Every joint on the robot was equipped with a PD controller with gains set at $K_p = 40$ and $K_d = 1$. To enhance the realism in simulation, we incorporated an actuator latency of 30 ms as identified by Margolis and Agrawal [8]. Our control policy is queried at 60 Hz, updating the PD references for the robot's joints.

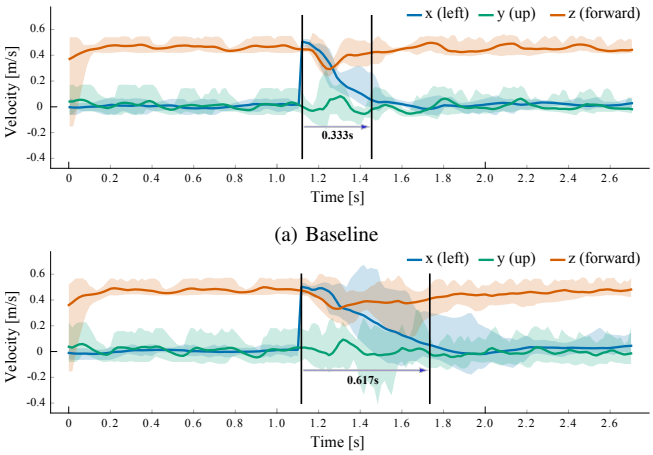
For RL training, we used the implementation of proximal policy optimization (PPO) algorithm [21] from stable-baseline3 [22] with the hyperparameters specified in Table III. The agent was trained for up to 200 million samples, and when executed on a system equipped with a Ryzen 7 3700X CPU and GTX 1070 GPU, full training takes roughly 15 hours. To ensure consistency and reproducibility of the results, we ran every simulation experiment multiple times using five policies trained with distinct random seeds. The resulting data from these runs was averaged and is presented in the figures along with standard deviations in the subsequent subchapters.

A. *Push recovery*

In the push recovery test, we assessed policies trained using our method and compared them against the performance of *baseline* policies. The baseline policies were trained with the same RL training setup but excluded the recovery stage. The test involved a brief forward walking segment, after which a push was introduced. We evaluate the policies' ability to withstand the push with maximal disturbance of $1.5\,\mathrm{m/s}$ in both horizontal axes. With varying push timings, 10 experiments per seed for every grid point get conducted and marked as either pass or fail in case the robot falls over. The averaged success rate is visualized in Figure 2.

From the result, it is evident that the compliant policies hold a slight advantage. For compliant policies, the 80% success rate contour encloses 78% of all grid points, whereas the baseline achieves 73%. However, this difference is subtle and we conclude that both the baseline and compliant policies perform commendably when evaluated against the straightforward pass-or-fail criterion.

Upon closer observation of the policies' behaviors, the advantages of compliant behavior become clear. We examined the robot's base velocity data after receiving an impulse of $0.5\,\mathrm{m/s}$ in the sideways direction. As shown in Figure 3, it's evident that compliant policies facilitate a more gradual



(a) Baseline

(b) Compliant

Fig. 3: The plots show the decay time of a velocity disturbance for the baseline (fig. 3a) and the compliant controller (fig. 3b.) The average is represented by the lines, with shaded regions indicating the minimum and maximum values of 20 individual runs of the same policy.

TABLE IV: Statistics of power and torque (averaged over specific time intervals) during a sideways pushing scenario. Values represent the mean and standard deviation from 5 experiments with different seeds.

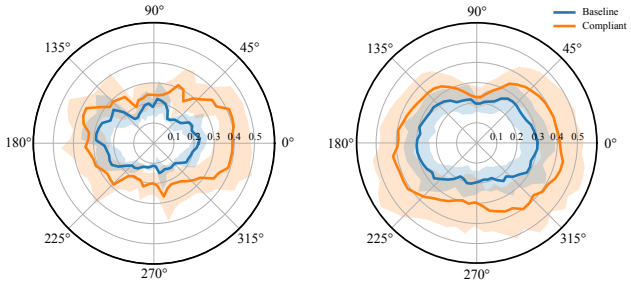
Quantity	Average	
	Baseline	Compliant
Power before push [W]	80.1 ± 1.38	78.5 ± 1.96
Power after push [W]	166.9 ± 25.8	$\mathbf{140.9 \pm 17.0}$
Torque before push [N · m]	26.9 ± 0.66	25.5 ± 0.76
Torque after push [N · m]	27.4 ± 1.28	$\mathbf{25.8 \pm 0.98}$

recovery. This indicates the system does not resort to aggressive forces to counter disturbances. In comparison, our compliant policy takes 85% more time to stabilize after a disturbance, resulting in smoother movements.

Additionally, we evaluated the energy and torque demands of the policies trained with both the baseline and compliant setups across 5 different seeds. table IV displays the mean values and standard deviations of the averaged energy and torque over a 1 s time window before and after the push. Following the application of a push, the compliant policies show a considerable improvement in energy efficiency, consuming 15% less mechanical power and requiring 6% fewer motor torques compared to the baseline (bold numbers in table IV). Interestingly, even in situations without disturbances, compliant policies perform marginally better than their counterparts.

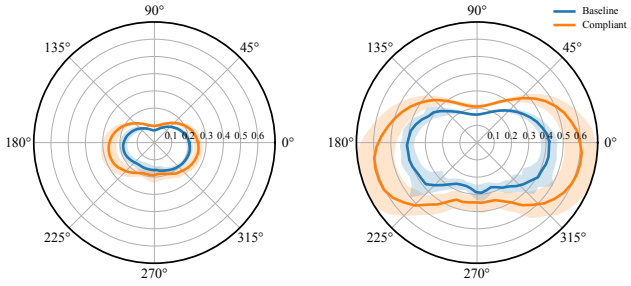
We assessed the response of the policies to an impulse of $0.5\,\mathrm{m/s}$ applied from various directions. Figure 4 indicates the duration, in seconds, required for the policy to recover from the impulse. In both scenarios—when the robot is in motion (Figure 4b) and when the robot remains stationary (Figure 4a)—the compliant policies exhibit a gentler reaction and, as a result, need more time to return to a stable state.

Shifting our focus from impulses to applied forces, we simulated a robot on a leash (as depicted in Figure 1) by consistently applying force while directing the robot to walk straight forward. Figure 5 illustrates the discrepancy in velocity tracking along different directions when the robot is



(a) Standing (b) Walking forward

Fig. 4: Measurements of the decay time in seconds of a velocity disturbance in all horizontal directions while standing still (fig. 4a) or walking forward (fig. 4b) at $0.5\,\mathrm{m/s}$. The solid line represents the average values from five policies trained with different seeds, with 10 runs each. The surrounding shaded area indicates the standard deviation.



(a) 10N force (b) 20N force

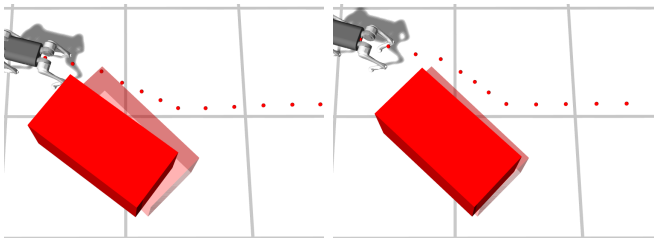
Fig. 5: Measurements of the average walking velocity offset in m/s when being dragged by a constant force and walking forward at 0.5m/s. In an undisturbed case both policies perform with an error $< 0.05\,\mathrm{m/s}$.

dragged with forces of magnitudes $10\,\mathrm{N}$ and $20\,\mathrm{N}$. As with the impulse test, the baseline policy has a stiffer response and permits less movement in the direction of the force compared to the compliant policy. As a result, the velocity deviation of the compliant policy tends to surpass that of the baseline. It's worth noting that the robot's compliance varies with the direction of the force. This variation can be attributed to the robot's geometry, which offers enhanced stability when subjected to forward or backward forces.

B. Safe Interaction with Environment

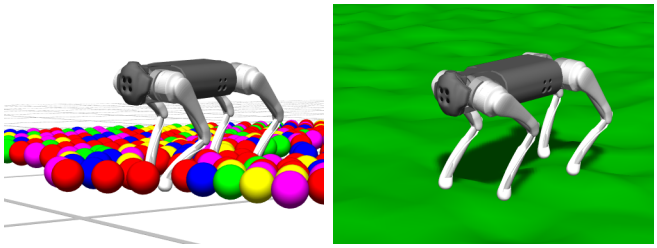
Compliance is closely related to safety, especially during interactions with the environment. When robots come into contact with humans or inadvertently bump into objects, it's crucial they don't apply excessive force. Overexertion risks damaging both the environment and the robot hardware.

To evaluate the safety aspect of controllers, we spawned a large box within the simulation for the robot to encounter via collision. As showcased in Figure 6, we directed the robot to proceed linearly towards the obstacle. Upon impact, the compliant policy enforces a milder force on the obstacle, effectively minimizing its progression in that direction. This results in the robot being gently deflected to the side upon contact, resuming its original trajectory once the obstruction no longer counteracts the preset command. On the other hand, the baseline policy remains undeterred in its onward motion, overlooking the countering force from the object, causing a considerable relocation of the obstacle.



(a) Baseline (b) Compliant

Fig. 6: The robot collides with a box weighing 2kg while following the command $v_x^* = 0.4$, $v_z^* = 0$, and $\omega_y^* = 0$. The box's initial position is indicated by the transparent area, and the robot's path is traced by the dotted lines.



(a) Cluttered terrain (b) Uneven terrain

Fig. 7: The robot traverses a ball pit of spheres weighing $0.5\,\mathrm{kg}$ (fig. 7a) and an uneven terrain with height variations up to $5\,\mathrm{cm}$ (fig. 7b.)

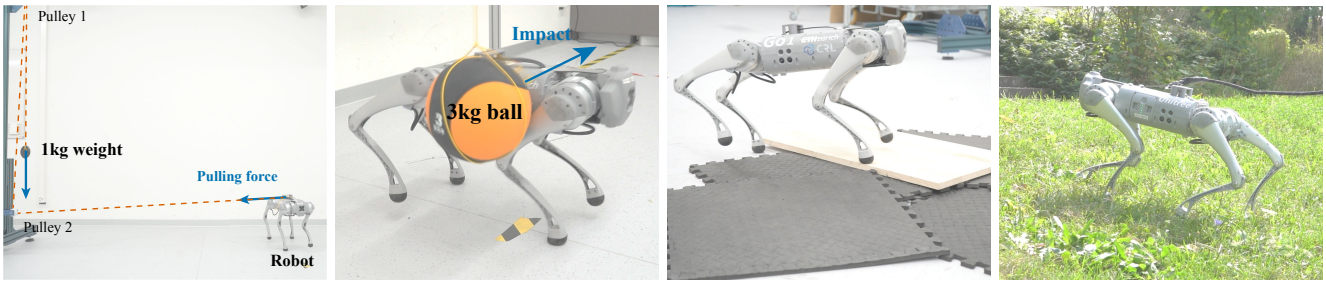
C. Uneven and Cluttered Terrain Navigation

We conducted additional simulation experiments to evaluate the proficiency of the policies in managing environmental uncertainties, particularly with respect to diverse terrains. In the first scenario, the compliant policy encounters an area scattered with balls, as illustrated in Figure 7a. The balls are modeled as rigid spheres, each with a weight of $0.5\,\mathrm{kg}$ and a radius of $5\,\mathrm{cm}$. They are positioned in a 20 by 20 grid with a spacing of $3\,\mathrm{cm}$ between each sphere.

In the second scenario, we deployed the robot to uneven terrains, which were randomly generated using Perlin noise [23] with a magnitude of $5\,\mathrm{cm}$ and a frequency of 0.5 as illustrated in Figure 7b. We emphasize that both scenarios were not shown to the policy during the RL training phase.

In both scenarios, the compliant policy adeptly navigates through the challenges presented by the obstacles and the roughness of the terrain. It is worth highlighting that the policy can handle disturbances that are exclusively acting on the feet while during training only pushes on the base were seen. Meantime, the baseline policy exhibits stiffer and more aggressive reactions in challenging conditions. The forceful reactions produce high impacts on the feet, resulting in less fluid and graceful movement. Due to this, there's an elevated chance of the robot becoming trapped or toppling over.

These results are consistent with the observations made by Lee et al. [23], suggesting that on challenging terrains for legged robots, it is often more effective to flexibly follow a desired velocity, especially when the feasible range of target speeds is unclear. For a more comprehensive grasp of the policies' behaviors, we encourage readers to watch the supplementary video showcasing our experiments.



(a) Pull test. (b) Impact test. (c) Cluttered terrain test. (d) Outdoor test.

Fig. 8: The snapshots from the hardware experiments featuring the *Unitree Go1* robot.

D. Hardware Verification

We verified our findings from the simulation experiments on the *Unitree Go1* robot hardware. To transfer our policy to the hardware effectively, we employed a Kalman filter-based state estimator by Bledt et al. [24] paired with a contact estimator trained through supervised learning. The contact estimator runs at a frequency of $240\,\mathrm{Hz}$ and processes joint states and actions from the previous four timesteps as inputs and generates boolean contact states for each foot as outputs. In contrast to the approach by Ji et al. [16], this training was not done concurrently but with data gathered from a fully trained policy. Furthermore, to bridge the simulation-to-reality gap more efficiently, we integrated dynamics randomization [25] into the RL training. This entailed random sampling of the friction coefficient within $[0.5, 1.2]$ and adding a calibration offset to each joint, which was drawn from a zero-mean normal distribution with a standard deviation of $0.01\,\mathrm{rad}$.

After the RL training of both the compliant and baseline policies, we executed pull and push tests in a controlled environment. Initially, we utilized a $1\,\mathrm{kg}$ weight linked to a pulley system to exert pulling forces on the robot, as shown in Figure 8a. During this test, the robot under the compliant policy generally moved in the force's direction rather than opposing it. Conversely, the baseline policy led the robot to counteract the force, which made it more prone to toppling. In the following test, depicted in Figure 8b, we applied an impulse to the robot by swinging a suspended $3\,\mathrm{kg}$ ball from a set height. Consistent with our simulation outcomes, the compliant policy reacted to the impact in a more gradual manner, while the baseline policy responded more abruptly.

In another test illustrated in Figure 8c, we placed the robot in an environment littered with wooden blocks and soft mats. Here again, the compliant policy outperformed, showing superior resilience and maneuvering over obstacles better than the baseline. We also tested the robot with the compliant policy in an outdoor setting with varied and sloped terrains, as highlighted in Figure 8d. Even though the policy was not specifically trained for cluttered or rugged terrains, the robot skillfully managed these challenges using our compliant policy.

For a visual representation of these hardware tests, we encourage readers to check out the footage included in our supplementary video.

V. Conclusion

In this work, we present a DRL approach to learn a legged locomotion control policy that exhibits compliant behaviors to environmental uncertainties. By inducing the compliant behavior that recovers from disturbance with a smooth response, we experimentally demonstrate that our compliant policy achieves more robust, energy-efficient, and safe interactions within the environment. Notably, we demonstrated that the policy reliably navigates uncluttered or uneven terrains, even those not presented during the RL training phase. These findings align with insights shared by Lee et al. [23], emphasizing that the ability to flexibly adhere to a desired velocity enhances traversal over difficult terrains.

In this paper, we leverage a basic RL framework with a simple reward function. Yet, our multi-stage episodic RL training strategy can be combined with a variety of RL techniques, such as motion imitation [9, 26], which integrates demonstrations from animals or model-based controllers into the learning process. Determining how this training strategy influences and promotes more compliant behaviors using these methods remains a subject for our future research.

Furthermore, we are keen to explore the generation of complex behaviors that require active interactions with the environments [7] in our future work. This might encompass tasks like door-opening or transporting items along a designated route, especially when there are uncertainties regarding the object's dimensions or weight. Although our current research primarily centers on passive responses, broadening its ambit to include active engagements offers promising prospects for a multitude of potential applications.

References

[1] D. Torricelli, J. Gonzalez, M. Weckx, R. Jiménez-Fabián, B. Vanderborght, M. Sartori, S. Dosen, D. Farina, D. Lefeber, and J. L. Pons, "Human-like compliant locomotion: state of the art of robotic implementations," *Bioinspiration Biomimetics*, vol. 11, no. 5, p. 051002, aug 2016.

[2] A. Kulkarni, C. Cui, S. Rietdyk, and S. Ambike, "Humans prioritize walking efficiency or walking stability based on environmental risk," *PLOS ONE*, vol. 18, no. 4, pp. 1–19, 04 2023.

[3] P. Fankhauser, M. Hutter, C. Gehring, M. Bloesch, M. A. Hoepflinger, and R. Siegwart, "Reinforcement learning of single legged locomotion," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 188–193.

[4] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth, and K. Sreenath, "Reinforcement learning for robust parameterized locomotion control of bipedal robots," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 2811–2817.

[5] 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.

[6] K. Xie, P. Xu, S. Andrews, V. B. Zordan, and P. G. Kry, "Too stiff, too strong, too smart: Evaluating fundamental problems with motion control policies," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 6, no. 3, aug 2023.

[7] S. Lee, P. S. Chang, and J. Lee, "Deep compliant control," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

[8] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," *Conference on Robot Learning*, 2022.

[9] D. Kang, J. Cheng, M. Zamora, F. Zargarbashi, and S. Coros, "Rl + model-based control: Using on-demand optimal control to learn versatile legged locomotion," *IEEE Robotics and Automation Letters*, vol. 8, no. 10, pp. 6619–6626, 2023.

[10] S. Gangapurwala, L. Campanaro, and I. Havoutis, "Learning low-frequency motion control for robust and dynamic robot locomotion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 5085–5091.

[11] A. Xiao, W. Tong, L. Yang, J. Zeng, Z. Li, and K. Sreenath, "Robotic guide dog: Leading a human with leash-guided hybrid physical interaction," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 11 470–11 476.

[12] Y. Sun, I. Jeelani, and M. Gheisari, "Safe human-robot collaboration in construction: A conceptual perspective," *Journal of Safety Research*, 2023.

[13] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[14] B. J. Stephens, "State estimation for force-controlled humanoid balance using simple models in the presence of modeling error," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3994–3999.

[15] V. Morlando, A. Teimoorzadeh, and F. Ruggiero, "Whole-body control with disturbance rejection through a momentum-based observer for quadruped robots," *Mechanism and Machine Theory*, vol. 164, p. 104412, 2021.

[16] G. Ji, J. Mun, H. Kim, and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4630–4637, apr 2022.

[17] Z. Fu, A. Kumar, J. Malik, and D. Pathak, "Minimizing energy consumption leads to the emergence of gaits in legged robots," in *5th Annual Conference on Robot Learning*, 2021.

[18] J. Tuyls, S. Yao, S. M. Kakade, and K. R. Narasimhan, "Multi-stage episodic control for strategic exploration in text games," in *International Conference on Learning Representations*, 2022.

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

[20] R. Smith, "Open dynamics engine," 2008, http://www.ode.org/.

[21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017.

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

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

[24] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.

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

[26] X. B. Peng, Y. Guo, L. Halper, S. Levine, and S. Fidler, "Ase: Large-scale reusable adversarial skill embeddings for physically simulated characters," *ACM Transactions On Graphics (TOG)*, vol. 41, no. 4, pp. 1–17, 2022.