Variable Stiffness for Robust Locomotion through Reinforcement Learning

Dario Spoljaric[1], **Yan Yashuai**[1] **and Dongheui Lee**[1,2]

[1] *Autonomous Systems, TU Wien, Vienna, Austria (e-mail: {dario.spoljaric, yashuai.yan, dongheui.lee}@tuwien.ac.at).*
[2] *Institute of Robotics and Mechatronics, German Aerospace Center (DLR), Wessling, Germany.*

Abstract: Reinforcement-learned locomotion enables legged robots to perform highly dynamic motions but often accompanies time-consuming manual tuning of joint stiffness. This paper introduces a novel control paradigm that integrates variable stiffness into the action space alongside joint positions, enabling grouped stiffness control such as per-joint stiffness (PJS), per-leg stiffness (PLS) and hybrid joint-leg stiffness (HJLS). We show that variable stiffness policies, with grouping in per-leg stiffness (PLS), outperform position-based control in velocity tracking and push recovery. In contrast, HJLS excels in energy efficiency. Despite the fact that our policy is trained on flat floor only, our method showcases robust walking behaviour on diverse outdoor terrains, indicating robust sim-to-real transfer. Our approach simplifies design by eliminating per-joint stiffness tuning while keeping competitive results with various metrics.

Keywords: reinforcement learning, quadruped locomotion, variable stiffness, sim-to-real.

1. INTRODUCTION

Animal and human-like locomotion has a significant advantage over wheeled mobile robots. They can traverse unstructured, challenging terrains. Therefore, various approaches are developed to solve quadrupedal and bipedal locomotion. Conventionally, this involved model-based controllers (Donghyun et al., 2019; Jared et al., 2018) with a complex pipeline that managed gait schedule, state estimation, whole body impulse control and actuator control. Recently, this discipline has made significant progress through model-free reinforcement learning (RL) methods. These approaches enable the design of controllers capable of following high-level commands (walking, running, jumping, etc.) and directly actuating the joint motors, bypassing the need for path planning and other parts within the control pipeline.

Usually, these controllers (Shuxiao et al., 2023; Gilbert et al., 2023) follow a position- or torque-based paradigm. The high-level controller (RL agent) learns a position policy in the position-based paradigm. Given the state, the RL agent predicts desired joint positions at low frequencies, which are transferred into torques by a high-frequency PD controller. This control paradigm requires manual engineering of motor stiffness and damping for different tasks. In contrast, humans and animals can adjust their stiffness and damping to handle different tasks. For example, we stiffen our foot joints when landing with a foot but relax in the swing phase. The torque-based control circumvents this by directly learning a torque policy, which shows higher compliance (Donghyeon et al., 2023). Generally, torque-based policies are more challenging to train because they require learning complex dynamics. On the other hand, position control has a good initial pose, easing the learning progress in the beginning. Torque-based RL agents are usually executed at higher speeds, necessitating more powerful hardware or smaller networks. This leaves the researcher with a choice, either additional tuning of joint stiffnesses or a hard-to-train policy with limits in scope.

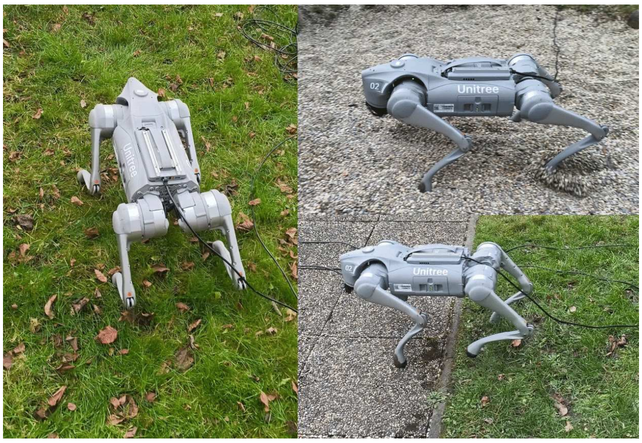


Fig. 1. A quadrupedal robot traverses across different terrains with variable-stiffness RL policy.

Another approach (Lei et al., 2024; Jiaming et al., 2024) is variable stiffness control, which is widely used in robotics, particularly in manipulator applications, to improve safety while still being able to track accurately. Preliminary studies from Lei et al. (2024) have shown that applying this technique to locomotion could enhance energy efficiency. This also seems reasonable as higher stiffness is required during contact with the ground, but less is needed during the swing phase. In quadruped locomotion, research from Xinyuan et al. (2022) within model-based frameworks has shown that variable stiffness can reduce contact forces.

However, successfully achieving locomotion with variable stiffness remains a significant challenge.

This paper studies reinforcement-learned controllers that learn joint stiffness alongside target positions. Consequently, the robot can automatically adjust the motor stiffness according to the task requirements. Our approach shows superior velocity tracking and push recovery performance while maintaining good energy efficiency and robust sim-to-real transfer performance. Results of the walking policy are shown in Fig. 1 and a video is available online [1].

2. RELATED WORK

RL has emerged as a promising approach for solving locomotion tasks, offering two primary paradigms: position-based and torque-based control. Gilbert et al. (2023) and Nahrendra et al. (2023) predict target joint positions, which are actuated using proportional-derivative (PD) controllers. These methods are widely favoured for their ease of training and robustness in high-level command tracking. Within this control, the torque applied to the motors is calculated using Eq. (1)

$$\tau_t = K_p(q_t^{target}(\theta) - q_t) + K_d(\dot{q}_{des} - \dot{q}_t) \qquad (1)$$

However, they suffer from limited compliance behaviour due to fixed stiffness and damping values (K_p and K_d), requiring extensive PD gain tuning that is often task- and robot-specific (Gilbert et al., 2023).

In this control, the PD controller acts as a low-level tracking module, where the proportional gain (P-gain) is representative of the stiffness and the derivative gain (d-gain) for the damping. Zhaoming et al. (2021) investigated the impact of the p-gains, suggesting that large proportional gain leads to instabilities in training. In contrast, the low proportional gain has significant tracking errors and behaves like a torque controller. Other research studied the impact on the derivative gain. Laura et al. (2023) showed small derivative gains result in learning instabilities, and large gains prevented tracking the target velocity.

To circumvent this PD controller, Shuxiao et al. (2023) and Donghyeon et al. (2023) studied torque control as an alternative and applied it to quadruped and biped locomotion. In this control, the actions are directly applied to the motors. Although this control showed higher achievable rewards in the long term, it must be executed at higher speeds to perform similarly to position control. It is more difficult to train initially. The higher control speeds limit the design freedom of torque-based controllers.

Xinyuan et al. (2022) showed for model-based control, that adapting stiffness according to the contact force led to sufficient walking for a quadruped on uneven terrains. Bogdanovic et al. (2020) studied the impact of including joint stiffness alongside joint positions. The torque was then calculated by Eq. (2).

$$\tau_t = K_t^p(\theta)(q_t^{target}(\theta) - q_t) - K_t^d(\theta)\dot{q}_t \qquad (2)$$

Overall, their work did not test their approach to locomotion but showed the superiority of this concept against torque and position control for two scenarios: a single-legged hopper and a manipulator. The single-legged hopper achieved more considerable jumping heights, and the manipulator maintained a continuous contact force during reference motions with the adjustment of the action space.

Term	Distribution	Units	Operator
Environmental properties			
Payload Mass (trunk)	$\mathcal{U}(-1.0, 3.0)$	kg	additive
Hip Masses	$\mathcal{U}(-0.5, 0.5)$	kg	additive
Ground friction	$\mathcal{U}(0.3, 1.25)$	-	multiplicative
Gravity offset	$\mathcal{U}(-1.0, 1.0)$	m/s²	additive
Noise in the observation space			
Joint positions	$\mathcal{U}(-0.01, 0.01)$	rad	additive
Joint velocities	$\mathcal{U}(-1.5, 1.5)$	rad/s	additive
Local velocity	$\mathcal{U}(-0.1, 0.1)$	m/s	additive
Local ang. Velocity	$\mathcal{U}(-0.2, 0.2)$	rad/s	additive
Projected gravity	$\mathcal{U}(-0.05, 0.05)$	rad/s²	additive
System delay	$\mathcal{U}(0.0, 15.0)$	ms	additive
Stiffness	$\mathcal{U}(0.8, 1.3)$	N/m	multiplicative
Damping	$\mathcal{U}(0.5, 1.5)$	kg/s	multiplicative
Motor strength	$\mathcal{U}(0.9, 1.1)$	-	multiplicative

Table 1. Domain randomisation in simulation

Inspired by these findings, we explore the integration of joint stiffness alongside joint positions in the action space for reinforcement learning-based quadruped locomotion.

3. METHODS

We aim to train a policy π_θ with parameters θ that can follow high-level velocity command $\mathbf{v}^{cmd} = [\mathbf{v}_{xy}^{cmd}, \omega_{yaw}^{cmd}]^T$. This includes the lateral velocity v_{xy}^{cmd} as well as angular rotation speed ω_{yaw}^{cmd}. The prediction of joint targets along joint stiffnesses should accomplish this. An overview of our approach is shown in Fig. 2.

3.1 Training

We train our controllers using Proximal Policy Optimisation (PPO) (Schulman et al., 2017) with 4096 environments in parallel for 2000 epochs. To improve learning efficiency, we apply early termination if the robot's orientation exceeds 90 degrees from its horizontal position, if joint or torque limits are exceeded or if the robot falls onto its hips, trunk, or LIDAR. The episode lasts 20 seconds, and we sample commands every 5 seconds.

Simulation environment: We use Mujoco-MJX (introduced by Todorov et al. (2012)) as the simulation environment and apply Domain randomisation to achieve a successful sim-to-real transfer. Table 1 shows the randomised parameters. Additionally, we expose the policies to external pushes applied from random xy directions. The force magnitudes of the pushes are randomised between 50 - 150 N, and the impulse is 8 - 15 Ns. The push is applied every 6 seconds, so the policies have to learn to react to such disturbances.

Observation: Observations passed to the actor must also be observable during deployment. This limits the freedom of the observations that are passed. The observation vector $\mathbf{o}_t$, as in Eq. (3), is composed of the body angular $\boldsymbol{\omega}_t$ and linear velocity $\mathbf{v}_t$, projected gravitational vector $\mathbf{g}_t$, joint angle difference from the default position $\mathbf{q}_t - \mathbf{q}_{default}$, the last action $\mathbf{a}_{t-1}$ and the velocity command $\mathbf{v}^{cmd}$.

[1] `https://drive.google.com/file/d/1SmwQSM5026Ri41Ue6J_IgP0NqYIFxC0t/view?usp=sharing`

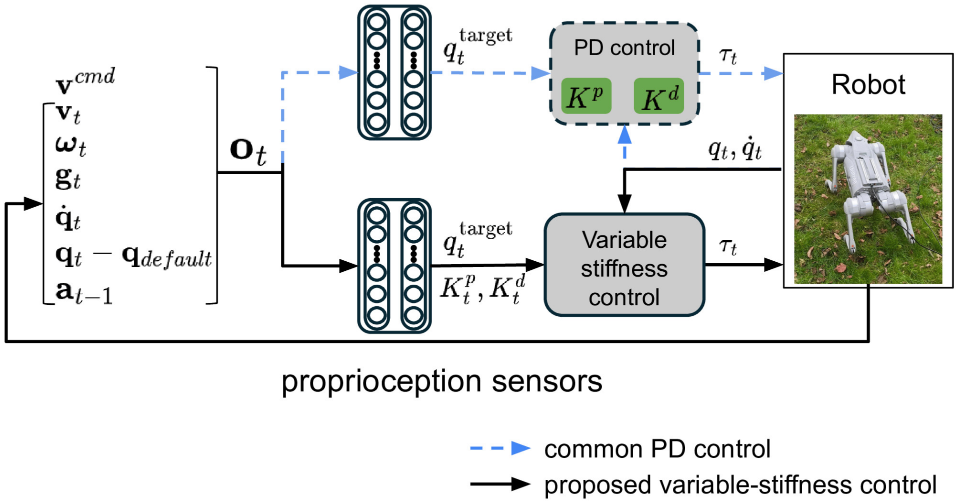


Fig. 2. Architecture of the position-based control (blue dashed line) compared to our variable stiffness control.

Reward	Equation (r_i)	Weight (w_i)
Lin. velocity tracking	$\exp\left(-4\left(\mathbf{v}_{xy}^{\text{cmd}} - \mathbf{v}_{xy}\right)^2\right)$	1.5
Ang. velocity tracking	$\exp\left(-4\left(\omega_{\text{yaw}}^{\text{cmd}} - \omega_{\text{yaw}}\right)^2\right)$	0.8
Linear velocity (z)	v_z^2	-2.0
Angular velocity (xy)	$\boldsymbol{\omega}_{xy}^2$	-0.05
Orientation	$\mathbf{g}_{xy}^2$	-5.0
Feet air time	$\sum_{f=0}^4 (t_{air,f} - 0.1), \lvert\mathbf{v}_{cmd}\rvert > 0.1$	0.2
Joint accelerations	$\ddot{\boldsymbol{q}}^2$	-2.5×10^{-7}
Joint power	$\lvert\boldsymbol{\tau}\rvert \cdot \lvert\dot{\boldsymbol{q}}\rvert$	-2×10^{-5}
Power distribution	$\text{var}(\lvert\boldsymbol{\tau}\rvert \cdot \lvert\dot{\boldsymbol{q}}\rvert)$	-10^{-5}
Foot slip	$\sum_{f=0}^4 \lvert\lvert\mathbf{v}_{f,xy}\rvert\rvert^2, \text{if } z < 0.01$	-0.1
Action rate	$(\mathbf{a}_t - \mathbf{a}_{t-1})^2$	-0.01
Foot clearance	$\sum_{f=1}^4 \left(\mathbf{p}_{f,z}^{des} - \mathbf{p}_{f,z}\right)^2 \lvert v_{f,xy}\rvert$	-0.1
Center of mass	$\left(\mathbf{p}_{com,xy} - \mathbf{p}_{xy}^{des}\right)^2, \; \mathbf{p}_{xy}^{des} = \frac{\sum_{f=1}^4 \mathbf{p}_{f,xy}}{4}$	-1.0
Joint tracking	$(\mathbf{q}_t^{target} - \mathbf{q}_{t+1})^2$	-0.1
Base height	$(h^{des} - h)^2$	-0.6
Hip	$\exp\left(-4 * \sum_{k=1}^4 (q_{hip,k} - q_{hip,k}^{default})^2\right)$	0.05
Collisions	$n_{collisions}$	-10.0
Termination	$n_{termination}$	-10.0

Table 2. Reward functions.

$$\mathbf{o}_t = \left[\mathbf{v}^{cmd}, \mathbf{v}_t, \boldsymbol{\omega}_t, \mathbf{g}_t, \dot{\mathbf{q}}_t, \mathbf{q}_t - \mathbf{q}_{\text{default}}, \mathbf{a}_{t-1}\right]^T \qquad (3)$$

We utilize privileged information to learn the critic network, which is dropped in the inference phase. The privileged vector $\mathbf{s}_t$ consists of the scaling factor for proportional and derivative gains and motor strength σ_t. In addition, $\mathbf{s}_t$ contains ground friction, adapted masses, disturbance force, and regular observation, as stated in Eq. (4).

$$\mathbf{s}_t = [k_{p,t}, k_{d,t}, \sigma_t, \mu, m_t, \mathbf{F}_{\text{kick}}, \mathbf{o}_t]^T \qquad (4)$$

Reward: The reward functions, similar to Nahrendra et al. (2023); Rudin et al. (2022); Bogdanovic et al. (2020), consist of task rewards to follow the command given and auxiliary rewards (penalties) to stabilize the learning process. An overview is provided in Table 2.

Given the reward components, the total reward is calculated using Eq. (5).

$$r_t = \sum r_i \cdot w_i \cdot dt \qquad (5)$$

In addition to existing rewards from the literature, we introduced a "Center of Mass" reward to encourage a stable walking gait. This reward calculates the desired centre-of-mass position p_{xy}^{des}, as the mean xy-components of the feet positions and penalises the squared error from this target, ensuring the centre of mass remains within the support polygon.

Action Space: The position-based controller uses 12 actions to specify target positions for each joint. Our controllers extend actions to adjust the stiffness of the PD controller, ranging from 20 to 60. To study the stiffness adjustments on quadrupedal robots with 12 DoFs, we propose different grouping strategies:

- **Individual Joint Stiffness (IJS)**: Predicts stiffness for each joint individually, extending the action space to 24 dimensions.
- **Per Joint Stiffness (PJS)**: Groups joints into hip, thigh, and knee categories, predicting one stiffness per group for a 15-dimensional action space.
- **Per Leg Stiffness (PLS)**: Predicts one stiffness value per leg, adding four actions for a 16-dimensional space.
- **Hybrid Joint-Leg Stiffness (HJLS)**: Combines PJS and PLS by representing stiffness as the outer product of a leg stiffness vector ($\mathbf{k^l} \in \mathbb{R}^4$) and a joint group stiffness vector ($\mathbf{k^j} \in \mathbb{R}^3$), resulting in 19 dimensions.

The damping of the PD controller is set to a fixed relationship to the p-gain, according to Eq. 6.

$$K_t^d = 0.2\sqrt{K_t^p} \qquad (6)$$

This relationship is inspired by the ratio between PD gains as in the works of Gilbert et al. (2023) and Rudin et al. (2022).

4. EVALUATION

In this section, we present the experimental results in both Mujoco-MJX and real-world hardware experiments to answer the following questions:

- Can variable stiffness policies improve velocity command tracking performance?
- Do variable stiffness policies show greater robustness against disturbances?
- Are variable stiffness policies more energy efficient?

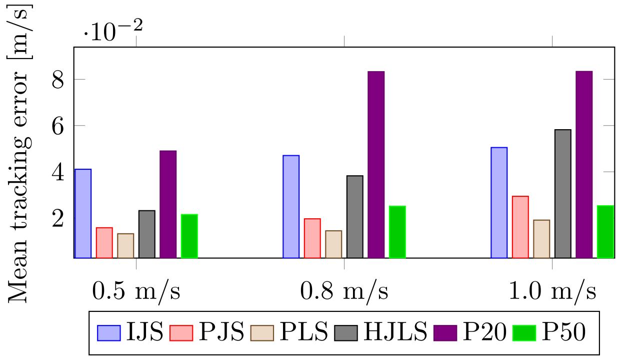


Fig. 3. Absolute tracking error for trained policies.

4.1 Baseline

The baseline is formed by position control policies with low stiffness 20, referred to as P20, and high stiffness 50, referred to as P50. These baselines, trained under identical conditions, follow prior choices from Nahrendra et al. (2023); B et al. (2024) for lower stiffness and Gilbert et al. (2023) for higher stiffness. While Gilbert et al. (2023) used even higher gains, we observed instabilities in training and selected 50 as the upper stiffness. These baselines highlight trade-offs, with each stiffness excelling in specific tasks.

4.2 Performance on walking and running

We evaluate walking and running performance by measuring tracking errors between the commanded and achieved velocities, a common method used in the works of Joonho et al. (2020). Controllers are tested on eight discrete headings (0°, 45°, ..., 315°) with target speeds of 0.5 m/s, 0.8 m/s, and 1.0 m/s to prevent bias for speed class or direction. Each heading direction is held for eight seconds, and domain randomisation is turned off to focus on tracking accuracy. Fig. 3 shows the absolute velocity tracking error averaged over the heading directions at different speeds. The comparison indicates that P20 has a significant tracking error, whereas P50 maintains lower error. Predicting individual joints in a policy (IJS) demonstrates the highest tracking error. However, our grouped stiffness policies, PJS and PLS, show lower or comparable tracking errors than P50. PLS manages to outperform all other controllers in every speed class.

4.3 Push recovery

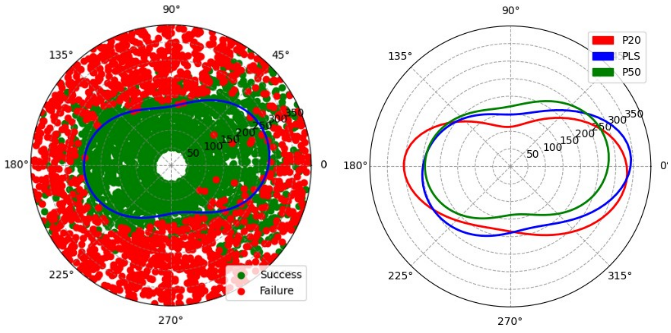
Robustness is evaluated by exposing the locomotion policy to external disturbances. Specifically, force pushes to the robot's trunk with domain randomisation. The evaluation is performed in the same simulation environment as the training. Push recovery is measured under the following conditions:

- Walking speed: 0.3 m/s
- Force push magnitude: 50 - 300N
- Push duration: 0.1 sec

Pushes are applied randomly in the xy-plane. The robot must walk straight for 5 seconds, with a random push applied between 2.5 and 3.5 seconds. A fall results in failure, while recovery and walking for 5 seconds are considered a success. The randomisation of the push event is done to prevent bias due to specific postures.

Control paradigm	Success Rate (%)				
	$< 100N$	$< 150N$	$< 200N$	$< 250N$	$< 300N$
IJS	**100.00**	98.73	96.06	**93.53**	**90.44**
PJS	**100.00**	**100.00**	96.63	89.34	81.64
PLS	**100.00**	99.66	**97.66**	92.34	85.65
HJLS	99.83	99.32	97.03	91.58	83.93
P20	99.48	99.07	94.98	89.93	83.08
P50	**100.00**	99.75	97.20	89.93	81.77

Table 3. Push recovery success rates within the specified push force magnitudes.



(a)

(b)

Fig. 4. **Comparison of maximal recoverable force push.** The polar scatter plot (a) shows the outcome of the experiment for specific forces (red=failure, green=success). The SVM, with classification confidence 90%, is used to draw a success boundary in the polar plot (a), which is then used to compare our methodology against the baselines in plot (b).

The success rate for pushes within the magnitude constraints is reported in Table 3. Within the push force of $150N$ (training range), our controller PJS performs best. Above $150N$ IJS shows the highest success rate, closely followed by PLS and HJLS. Since PLS also outperformed the baselines in velocity tracking and shows the second-highest success rate ($< 300N$), we use this policy for further comparison.

For comparison, we draw a polar scatter plot illustrating the magnitude and angle of the force applied to the robot trunk. We utilise a Support vector machine(SVM) with a radial bias function kernel to classify a maximum recovery boundary. This maximum recovery boundary is used to compare the methods. Figure 4 shows the results of this experiment.

All policies show higher resilience in and against the walking direction (0, 180°). Pushes applied from the side are not well compensated. P50 shows a smaller region than P20. Our policy PLS demonstrates a convex and similar shape to the P20.

Figure 5 further demonstrates how the baseline policies react to a force of 190N applied from the frontal left direction. The graph shows the stiffness values plotted over time for the respective legs. The baseline policies fall, whereas the PLS manages to recover by adjusting the stiffnesses of the legs. Notably, the legs opposite to the push stiffen way above the maximum stiffness of the high stiffness policy, and the legs towards the push relax. These

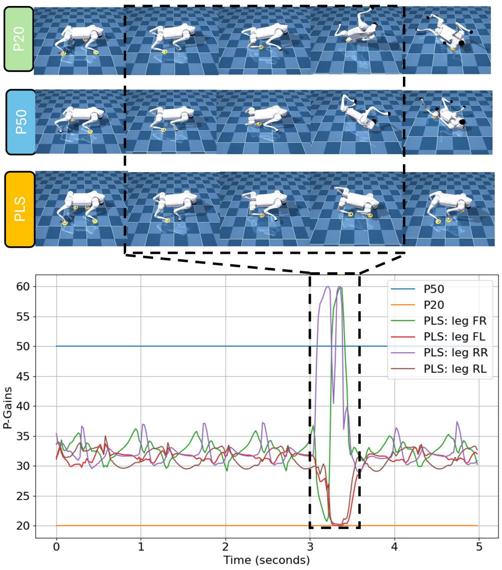


Fig. 5. **Push experiment**: Stiffness plotted over time. Push conditions: $t_{push} = 3s$, $t_{duration} = 0.1s$, $F = 190N$, $\theta = 225°$ accounting for the point in time, duration, magnitude and direction at which the push is applied. The force is applied from the frontal left direction. (FR/FL: front left/right, RR/RL: Rear right/left)

results demonstrate the superior performance of variable stiffness in this setting.

4.4 Energy efficiency

Similar to the work of Joonho et al. (2020), we evaluate energy efficiency with the cost of transport (CoT) for different speed classes. They define the CoT as Eq. (7), where M_{total} accounts for the total mass of the robot, τ denotes the measured torques, $\dot{q}$ the joint velocities, g the gravitational acceleration and v for the measured velocity.

$$CoT = \frac{E}{M_{total}gd} = \frac{P}{M_{total}gv} = \frac{\tau\dot{q}}{M_{total}gv} \tag{7}$$

We measure this metric applied to the same experiment as described in section 4.2. The results are shown in Fig. 6 and reveal mixed results. The lower the CoT, the more energy efficient. Our policies PJS, PLS and HJLS demonstrate lower CoT than P50 but higher than P20. This is also expected as higher stiffness leads to higher torques and, therefore, higher energy consumption. Our policies can adjust their stiffness between 20 and 60; thus, we expect the CoT to be somewhere in between. Lowering the lower border of stiffness compromised gait quality in our experiments. HJLS demonstrates the lowest CoT among our policies and, therefore, remains competitive with P20. These results validate the effectiveness of our proposed methodology in grouping the stiffness.

4.5 Sim-to-real transfer

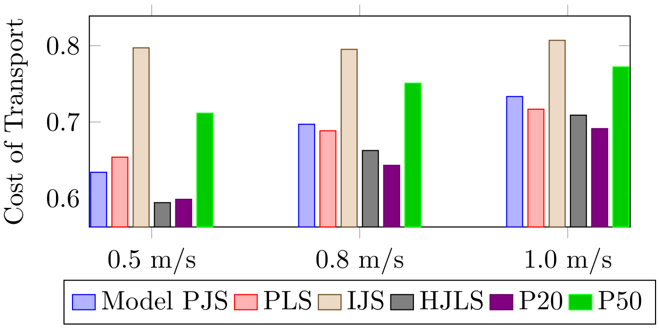


Fig. 6. Cost of Transport for trained policies.

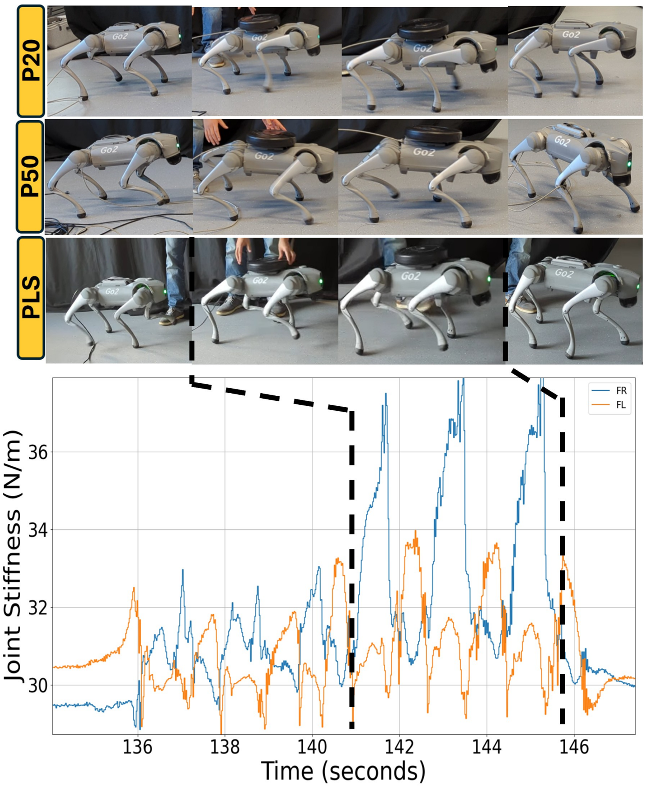


Fig. 7. **Payload experiment:** Adding a 5kg mass during walking to the baselines and PLS policy.

During hardware deployment, we evaluate the robustness of the learned locomotion by adding a payload or walking over diverse terrains. In training, the policy encounters a randomised payload of up to 3kg. For evaluation, we add a 5 kg payload on the robot during walking and observe increased stiffness while maintaining a proper gait, as seen in Fig. 7. When the payload is removed, the stiffness decreases accordingly.

We also evaluate our variable stiffness policy in the outdoors to traverse various terrains. Although it is trained solely on a flat floor, the learned policy demonstrates robust walking across diverse surfaces, including mud, grass, sidewalks, and sand (see Fig. 8).

5. FUTURE WORK

Our work demonstrates the benefits of variable stiffness. In future works this method could be applied to learn even more different tasks like crouching, hopping stair traversal and imitating motions. As this approach is able to adjust

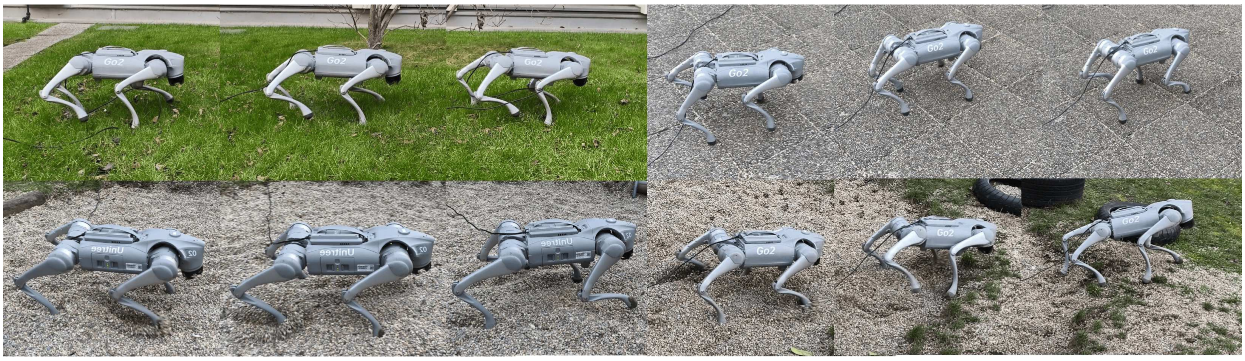


Fig. 8. Despite being trained solely on a flat floor in simulation, we showcase the robot's ability with our variable-stiffness RL policy to walk robustly on diverse outdoor terrains, such as grass, stones, and sand.

the stiffness this method might also learn these tasks for multiple robot types and combine it into one policy.

6. CONCLUSION

In this paper, we studied an alternative approach to learning locomotion on a quadruped robot, which uses joint positions alongside stiffnesses as the action space in a reinforcement learning paradigm. Simulation and real-world experiments are conducted to investigate performance on walking and running, as well as push recovery, energy efficiency and sim-to-real transfer. Our policy, which predicts stiffness per leg, outperformed baselines in the robustness test and velocity tracking. On the other hand, individual joint stiffness prediction struggled, underscoring the efficiency of our found groupings. Hardware tests show stiffness adaptation when encountered with payload and robust walking over diverse terrains. Our research highlights the potential of reinforcement learned variable stiffness locomotion as it combines the advantages of low and high stiffness.

REFERENCES

B, M.G., Ge, Y., Kartik, P., Tao, C., and Pulkit, A. (2024). Rapid locomotion via reinforcement learning. *The International Journal of Robotics Research.*

Bogdanovic, Miroslav, Majid, K., and Ludovic, R. (2020). Learning variable impedance control for contact sensitive tasks. *IEEE Robotics and Automation Letters.*

Donghyeon, K., Glen, B., Mathew, S., and Jaeheung, P. (2023). Torque-based deep reinforcement learning for task-and-robot agnostic learning on bipedal robots using sim-to-real transfer. *IEEE Robotics and Automation Letters.*

Donghyun, K., Jared, D.C., Benjamin, K., Gerardo, B., and Sangbae, K. (2019). Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control. *arXiv.*

Gilbert, F., Hongbo, Z., Zhongyu, L., Bin, P.X., Bhuvan, B., Linzhu, Y., Zhitao, S., Lizhi, Y., Yunhui, L., Koushil, S., et al. (2023). Genloco: Generalized locomotion controllers for quadrupedal robots. In *Conference on Robot Learning.*

Jared, D.C., M, W.P., Benjamin, K., Gerardo, B., and Sangbae, K. (2018). Dynamic locomotion in the mit cheetah 3 through convex model-predictive control. In *2018 IEEE RSJ international conference on intelligent robots and systems (IROS).*

Jiaming, F., Ziqing, Y., Han, L., Lianxi, Z., and Dongming, G. (2024). A novel variable stiffness compliant robotic link based on discrete variable stiffness units for safe human robot interaction. *Journal of Mechanisms and Robotics.*

Joonho, L., Jemin, H., Lorenz, W., Vladlen, K., and Marco, H. (2020). Learning quadrupedal locomotion over challenging terrain. *Science Robotics.*

Laura, S., Ilya, K., and Sergey, L. (2023). Demonstrating a walk in the park: Learning to walk in 20 minutes with model-free reinforcement learning. *Robotics: Science and Systems (RSS) Demo.*

Lei, Y., Haizhou, Z., Siying, Q., Gumin, J., and Yuqing, C. (2024). A compact variable stiffness actuator for agile legged locomotion. *IEEE ASME Transactions on Mechatronics.*

Nahrendra, I.M.A., Yu, B., and Myung, H. (2023). Dreamwaq: Learning robust quadrupedal locomotion with implicit terrain imagination via deep reinforcement learning. arXiv.

Rudin, N., Hoeller, D., Reist, P., and Hutter, M. (2022). Learning to walk in minutes using massively parallel deep reinforcement learning.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal policy optimization algorithms.

Shuxiao, C., Bike, Z., W, M.M., Akshara, R., and Koushil, S. (2023). Learning torque control for quadrupedal locomotion. In *IEEE-RAS 22nd International Conference on Humanoid Robots (Humanoids).*

Todorov, E., Erez, T., and Tassa, Y. (2012). Mujoco: A physics engine for model-based control. In *2012 IEEE RSJ International Conference on Intelligent Robots and Systems.*

Xinyuan, Z., Yuqiang, W., Yangwei, Y., Arturo, L., and Nikos, T. (2022). Variable stiffness locomotion with guaranteed stability for quadruped robots traversing uneven terrains. *Frontiers in Robotics and AI.*

Zhaoming, X., Xingye, D., van de Panne Michiel, Buck, B., and Animesh, G. (2021). Dynamics randomization revisited: A case study for quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation.*