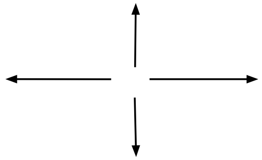
Seismic Stability of a Biped Robot

Gregory Wong, Jatheendra Kankanam Pathiranage, Hanqing Shi, Samuel Mankoff

Motivation

- We have seen cases of these kinds of robots being pushed and recovering, but we wanted to apply it to something relevant within the Bay Area.
    - In this case, earthquakes.
- With household robots becoming more of a normality, these bipedal robots need to account for regions of the world that are seismically unstable.

Problem Statement



- Simulate an earthquake environment by oscillating the ground laterally and vertically to create instability conditions for a legged robot
    - Use a controlled motion or random disturbance signal (velocity or force) to generate realistic ground motion
- Develop a control strategy that allows walker to maintain balance on an unstable surface or subjected to random forces/velocities.
- Evaluate controller robustness to different frequencies and amplitudes of ground motion and identify the limits of stability

Approach

To simulate the earthquake, we designed two ways:

1. instant velocity changes applied on the torso.

We set velocity range to be $V_x \in [-3, 3]$ m/s, $V_y \in [-0.5, 0.5]$ m/s to simulate strong forward/backward pushes and mild lateral pushes. And then wait $t \in [1,3]$ s for its recovery.

2. External forces applied on the feet.

We set force range to be $F_x \in [-75, 75]$, $F_y \in [-75, 75]$ $F_z \in [-1, 1]$ but applied more frequently.

Approach

Design of rewards: positive

Reward Name	Weight	Description
alive	**+10.0**	Main incentive to prevent termination.
feet_contact	**+2.0**	Encourages standing in place first for small perturbations.
posture	+1.0	Penalizes deviation from a nominal running joint configuration with loose arm constraints.

Approach

Design of rewards: negative

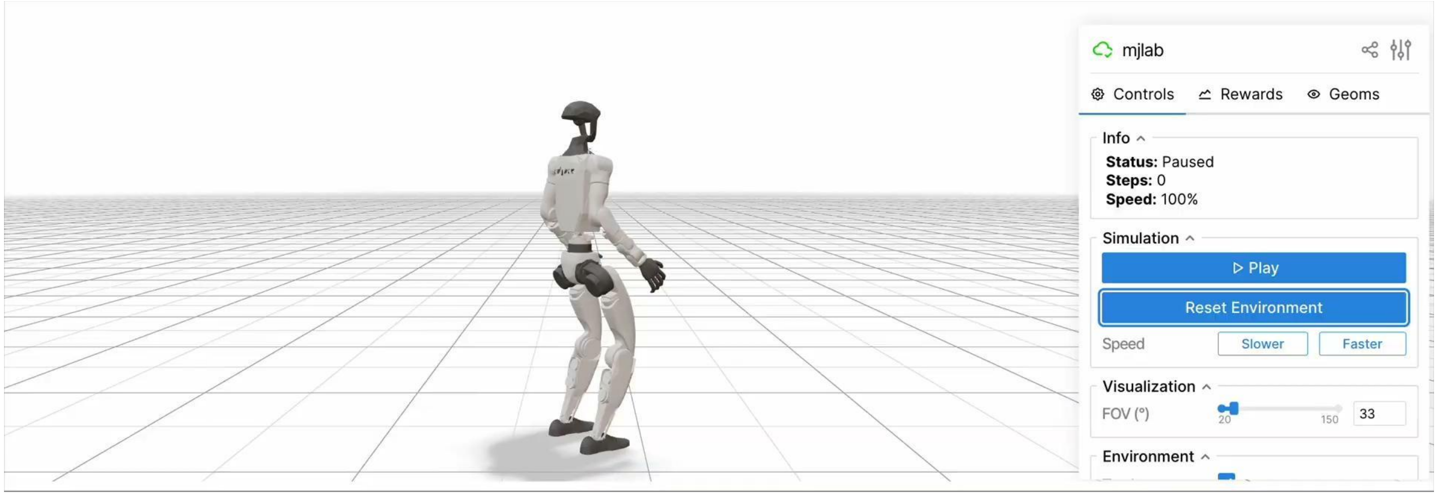
Reward Name	Weight	Description
feet_slip	**-10.0**	Penalizes foot sliding velocity to enforce stable standing.
knee_height	**-10.0**	Penalizes knee collapse.
penalty_lin_vel	-1.0	Penalizes base linear velocity (robot should remain stationary).
penalty_ang_vel	-0.2	Penalizes base angular velocity but with a small weight.
angular_momentum	-0.1	Penalizes excessive whole-body angular momentum. Encourages arm swing to cancel angular momentum.
action_rate_l2	-0.001	Regularization for smooth control actions.

Approach

Design of Terminations

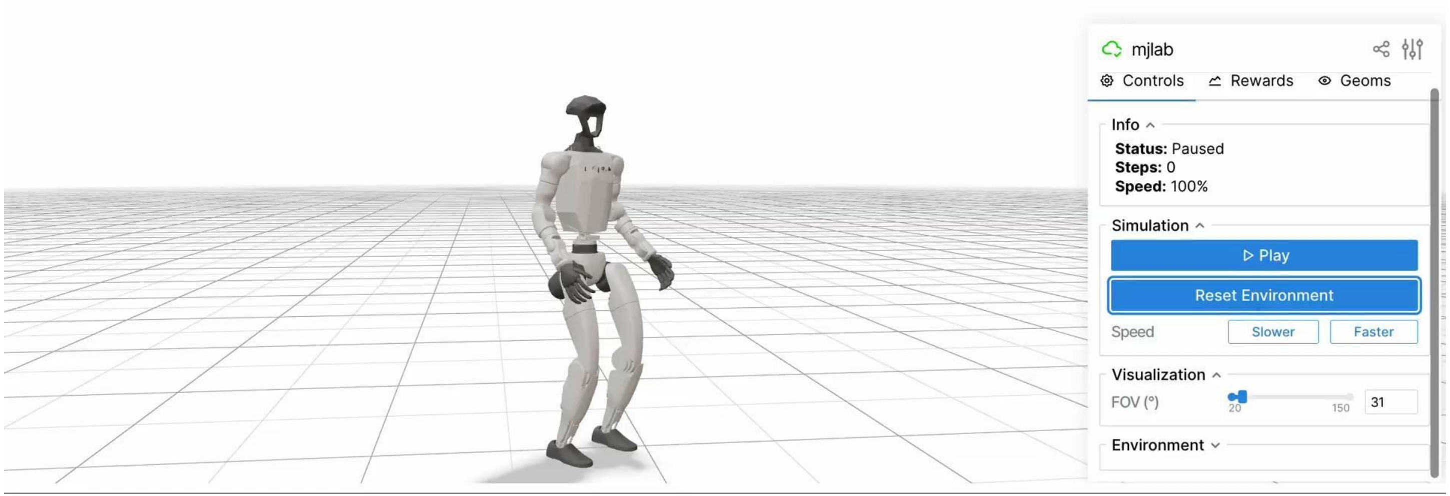
Termination	Termination Conditions	Termination Description
Fall Threshold	> 90 degree	Base orientation (Roll or Pitch)
Collapse Threshold	< 0.3 m	Base height limit
Time Limit	6.0 s	Maximum episode duration which equals 300 steps

Results(Vel.) - Failures without knee height penalty



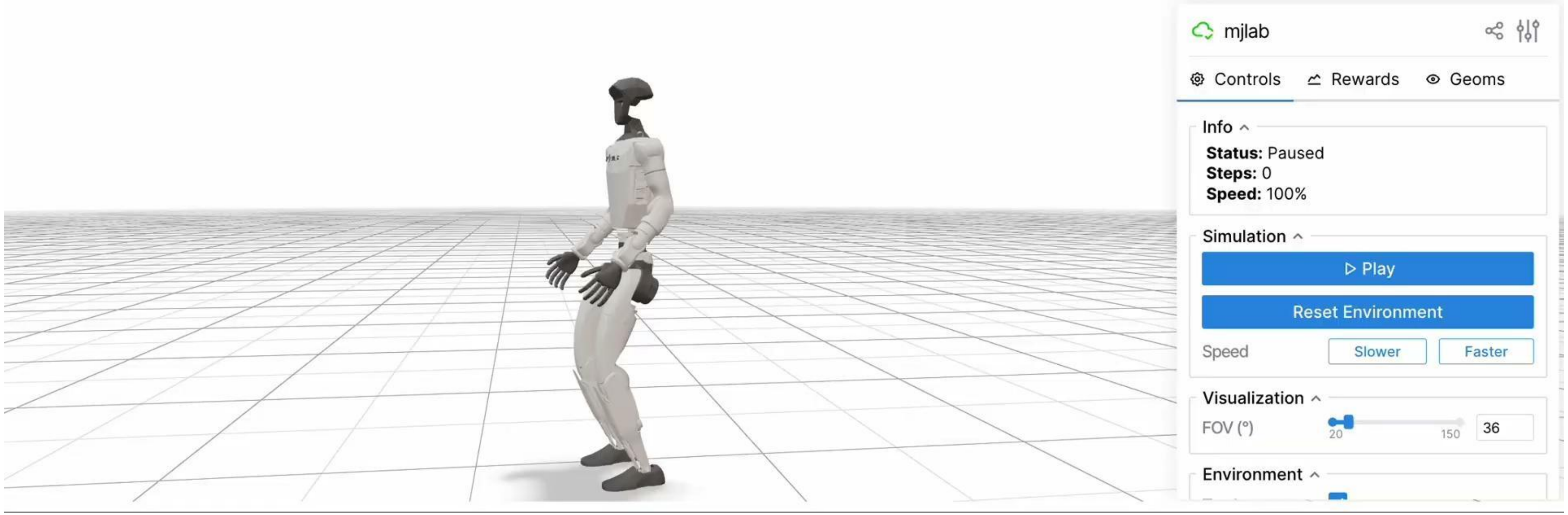
One of the first models we had the G1 robot attempt to use their knee as a way to reduce forward force. We considered this a failure due to the possible damage to the robot.

Results(Vel.) - Success



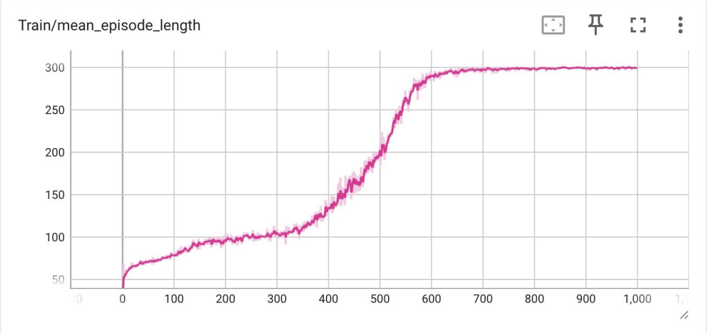
The robot maintains its posture and uses its arms as a counterbalance to stay upright. It also takes steps to help it stabilize

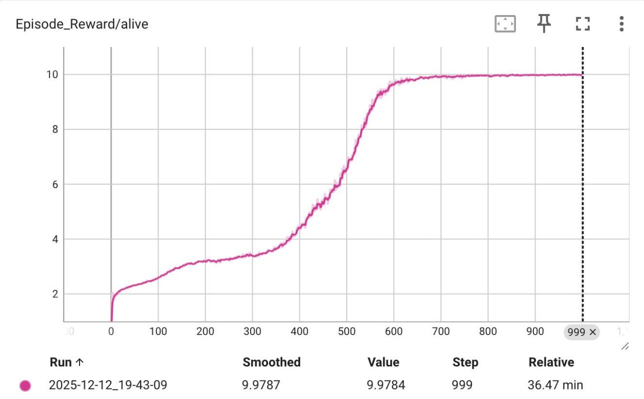
Results(Vel.) - Most Extreme Case

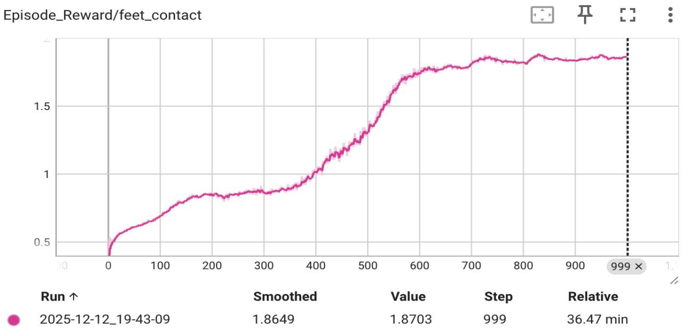


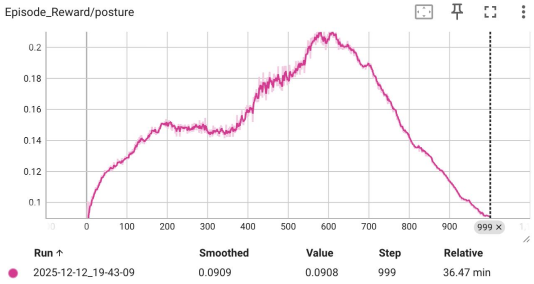
This test used a higher value (5m/s push) to simulate and extreme case. This model would work most of the time until a push became too strong to resist.

Results(Vel.) - Positive Rewards



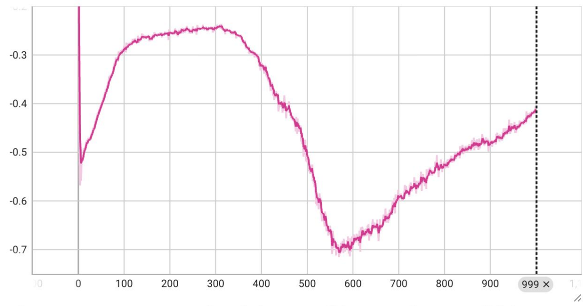






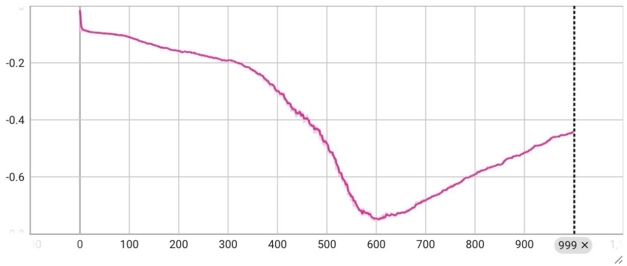
Results(Vel.) - Main Negative Rewards

Episode_Reward/angular_momentum



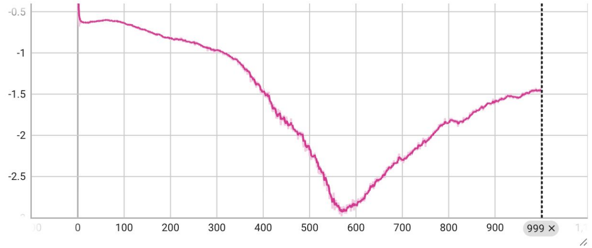
Run ↑	Smoothed	Value	Step	Relative
2025-12-12_19-43-09	-0.4115	-0.4056	999	36.47 min

Episode_Reward/penalty_ang_vel



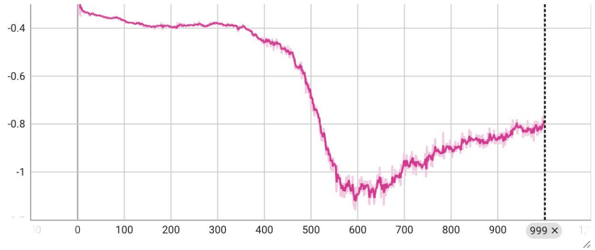
Run ↑	Smoothed	Value	Step	Relative
2025-12-12_19-43-09	-0.4429	-0.4418	999	36.47 min

Episode_Reward/feet_slip



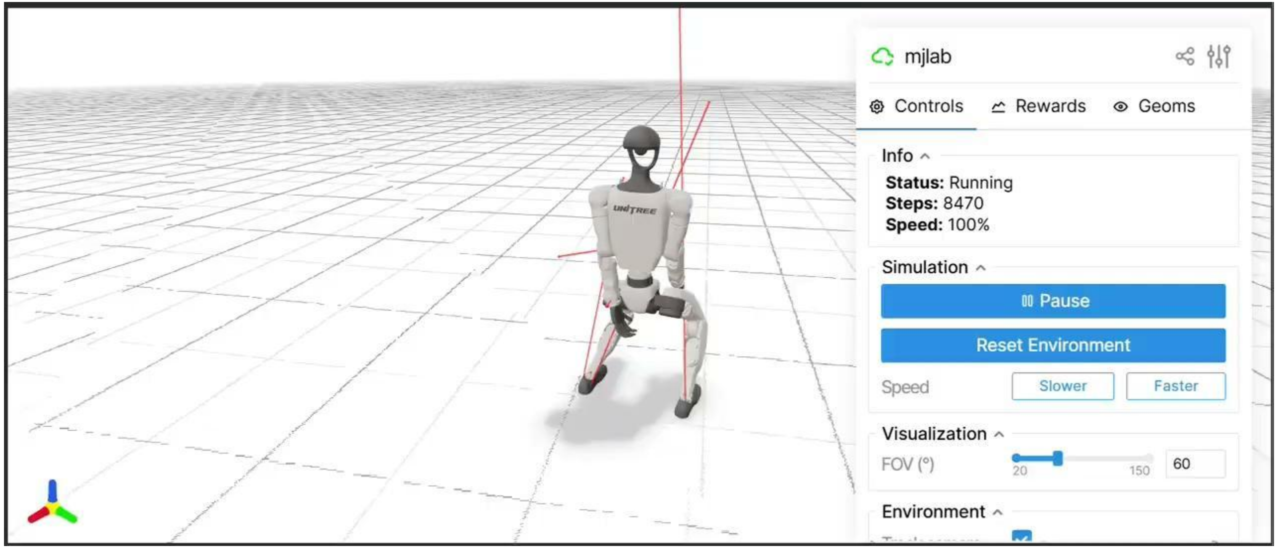
Run ↑	Smoothed	Value	Step	Relative
2025-12-12_19-43-09	-1.4466	-1.4411	999	36.47 min

Episode_Reward/penalty_lin_vel



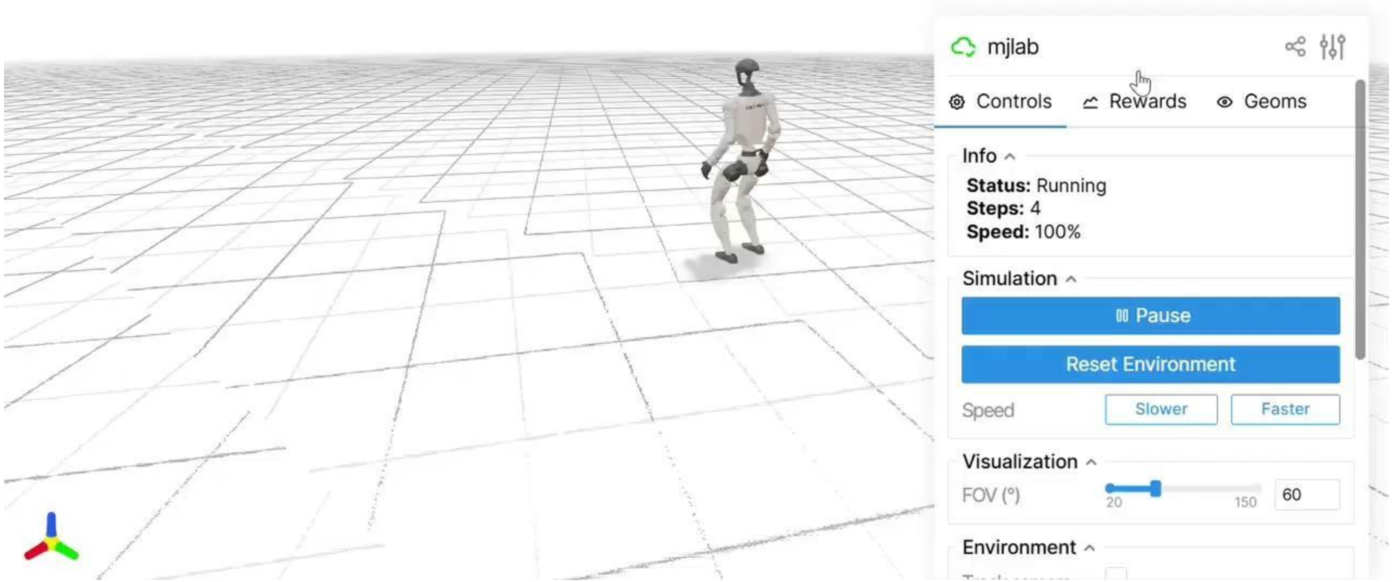
Run ↑	Smoothed	Value	Step	Relative
2025-12-12_19-43-09	-0.7853	-0.7586	999	36.47 min

Results(Force) - Success



The robot maintains its balance when randomized forces are applied to the feet. Forces are shown in the video.

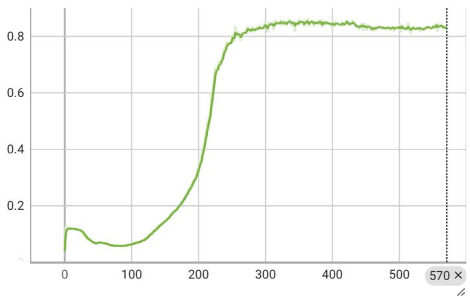
Results(Force) - Most Extreme Case



This test used a higher value [-150N, 150N] to simulate an extreme case.

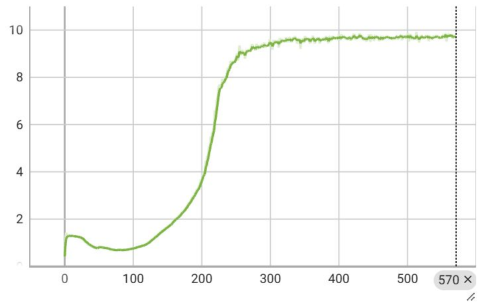
Results(Force) - Positive Rewards

Episode_Reward/posture



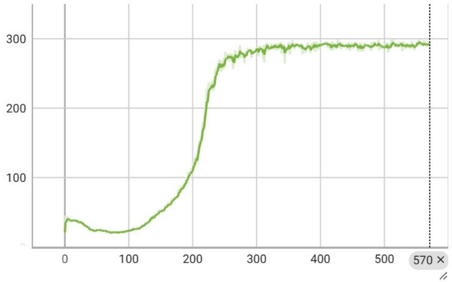
Run ↑	Smoothed	Value	Step	Relative
2025-12-13_01-36-38	0.8303	0.8302	570	15.19 min

Episode_Reward/alive



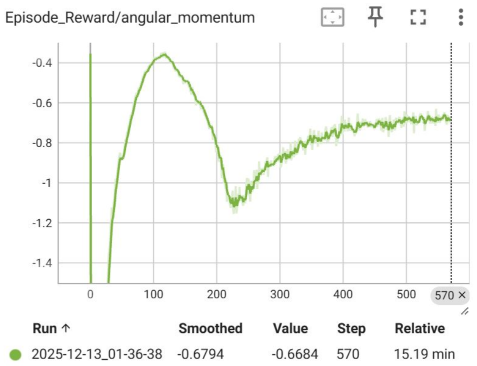
Run ↑	Smoothed	Value	Step	Relative
2025-12-13_01-36-38	9.7128	9.7192	570	15.19 min

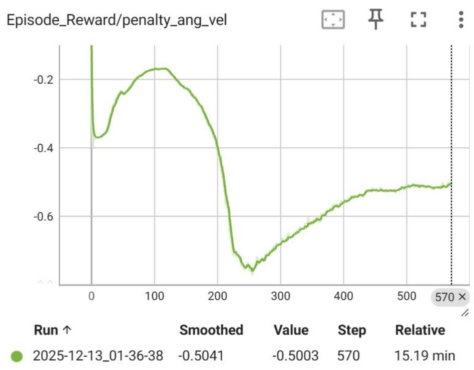
Train/mean_episode_length

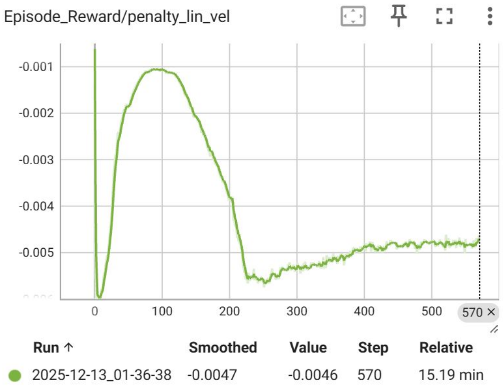


Run ↑	Smoothed	Value	Step	Relative
2025-12-13_01-36-38	293.1664	296.59	570	15.19 min

Results(Force) - Main Negative Rewards







References

Lecture slides 21-22

[HLD+19] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. Science Robotics, 4(26):eaau5872, 2019.

[TFR+17] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 23–30. IEEE, 2017.