Simulation of Robotic Systems 2025

Course Project

Project Title: Evaluation of Classic and Reinforcement Learning Control Methods for Raibert's Hopper in MuJoCo

Course Tutor: Rakshin Egor Aleksandrovich

Student Name: Mazin Mohammed Hamdan AL-Hatmi

Group: R4137c

Date (completed report date): 15/12/2025

Abstract

This project investigates control strategies for Raibert's Hopper simulated in the MuJoCo physics engine. A classical proportional–derivative (PD) controller, a reinforcement learning (RL) controller trained using Proximal Policy Optimization (PPO), and a conceptual hybrid control approach are considered. The classical controller serves as a baseline and demonstrates the challenges of manual tuning for highly nonlinear legged locomotion systems. In contrast, the RL controller successfully learns stable hopping and forward locomotion. A hybrid control concept is proposed to combine the interpretability of classical control with the adaptability of reinforcement learning. Experimental results highlight the superior performance of the RL controller and motivate future hybrid approaches.

1. Introduction

Legged locomotion is a challenging control problem due to nonlinear dynamics, intermittent ground contact, and hybrid stance–flight phases. Raibert's Hopper is a well-known benchmark that captures these challenges in a simplified planar setting. Traditional control approaches rely on hand-crafted rules and gain tuning, while modern reinforcement learning methods can learn control policies directly from interaction with the environment.

The objective of this project is to compare classical and reinforcement learning control strategies for Raibert's Hopper in the MuJoCo simulator. The comparison focuses on stability, forward velocity, and control effort. Additionally, a hybrid control concept is discussed as a potential extension that combines the strengths of both approaches.

2. Simulation Environment

All experiments were conducted using the MuJoCo physics engine through the Gymnasium interface. The Hopper environment models a single-legged robot constrained to planar motion. Although MuJoCo performs full three-dimensional rigid-body simulation internally, the Hopper task is inherently planar, making two-dimensional performance metrics sufficient for evaluation.

The Hopper environment provides observations including body position, joint angles, and velocities, while actions correspond to joint torques. Episodes terminate when the hopper falls or violate stability constraints.

3. Control Methods

3.1 Classical PD Baseline Controller

A classical proportional–derivative (PD) controller was implemented as a baseline. The controller applies joint torques based on the error between desired and actual joint angles, along with joint velocity feedback. Simple heuristic targets were used to encourage upright posture and forward motion. Despite tuning efforts, the controller remained highly sensitive to gain selection and lacked explicit handling of stance and flight phases.

3.2 Reinforcement Learning Controller (PPO)

The reinforcement learning controller was trained using Proximal Policy Optimization (PPO), a policy-gradient method known for stable and efficient learning. The PPO agent was trained for 100,000 timesteps to maximize cumulative reward, which implicitly encourages stable hopping and forward locomotion. The learned policy outputs joint actions directly based on the observed state.

3.3 Hybrid Control Concept

A hybrid control approach is proposed that combines classical control structure with reinforcement learning. In this concept, a classical Raibert-style controller provides low-level joint control, while reinforcement learning is used to tune high-level parameters such as desired hopping height, velocity gains, or foot placement. This approach aims to retain interpretability and stability while benefiting from data-driven optimization. Due to time constraints, the hybrid controller was evaluated conceptually rather than through full simulation experiments.

4. Experimental Setup

All experiments were conducted using identical environment settings to ensure a fair comparison between controllers. Performance metrics included episode length (steps survived), average forward velocity, velocity variability, and average action magnitude as a measure of control effort. For each controller, a single representative rollout was analyzed and visualized.

5. Results

5.1 Reinforcement Learning Controller Results

The PPO-trained controller achieved stable hopping and sustained forward locomotion. During evaluation, the controller survived for 255 simulation steps and achieved an average forward velocity of 2.10 m/s with a standard deviation of 0.83 m/s. The action magnitude remained moderate, indicating efficient control. Time-series plots showed acceleration from rest, bounded velocity oscillations, and consistent vertical motion.

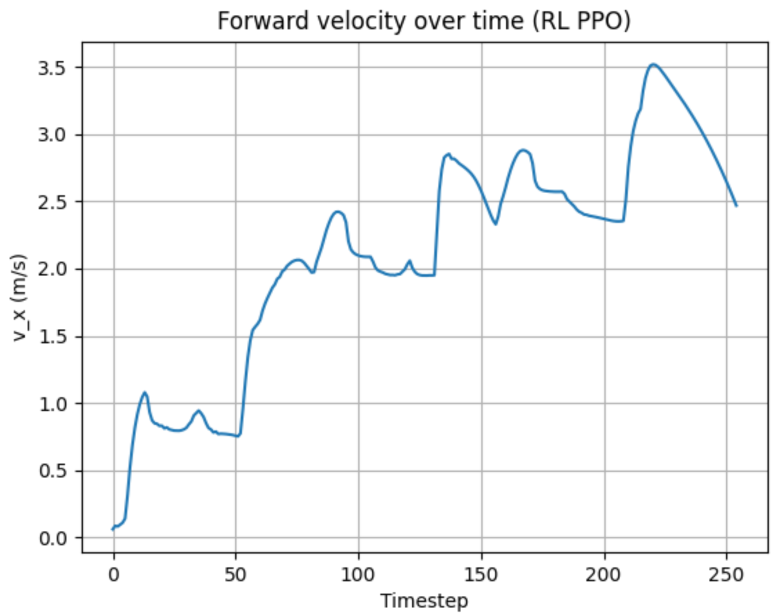

Fig.1: (Forward Velocity Over Time for the PPO-Based Reinforcement Learning Controller.)

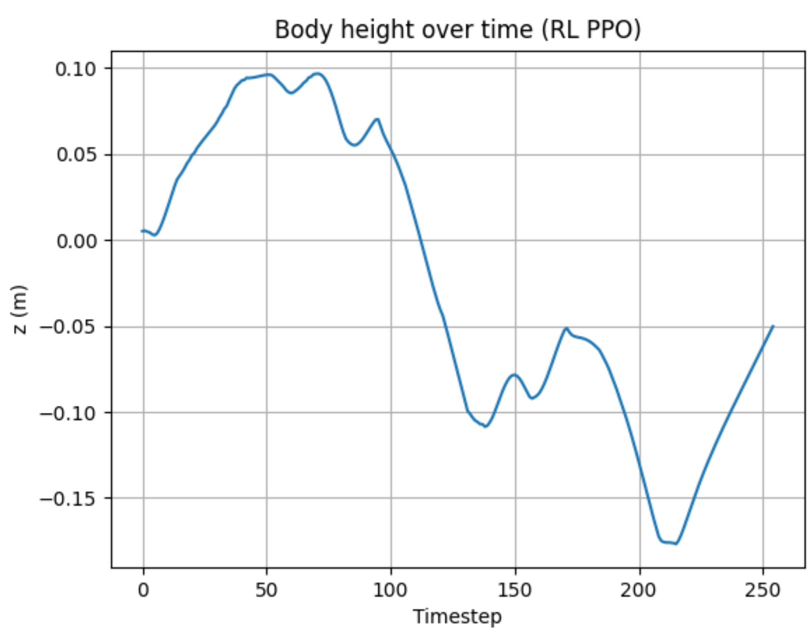

Fig.2: (Body height over time for the PPO-based reinforcement learning controller.)

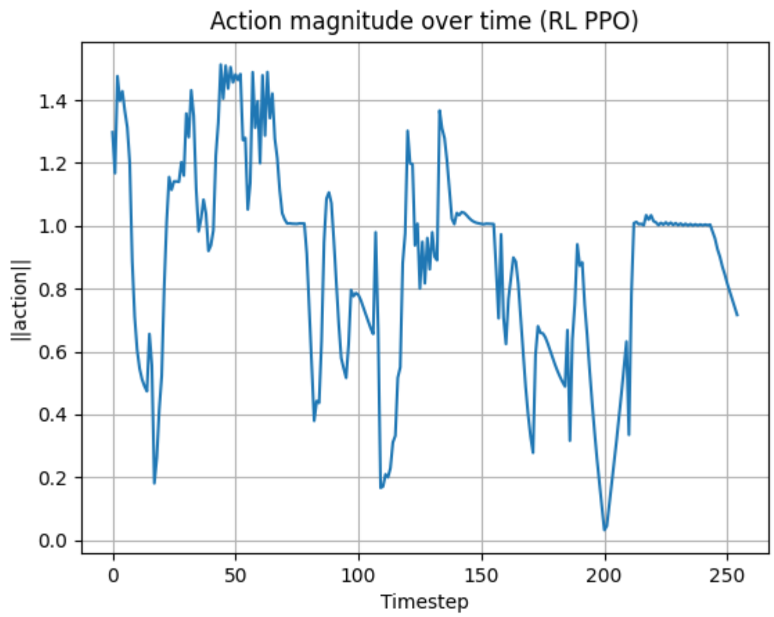

Fig.3: (Action Magnitude Over Time for the PPO-Based Reinforcement Learning Controller.)

5.2 Classical PD Baseline Results

The classical PD controller failed to achieve stable locomotion. It survived for only 16 simulation steps and exhibited a negative average forward velocity of −0.46 m/s, indicating backward motion. Despite the short duration, the controller applied relatively high control effort, highlighting inefficiency and instability.

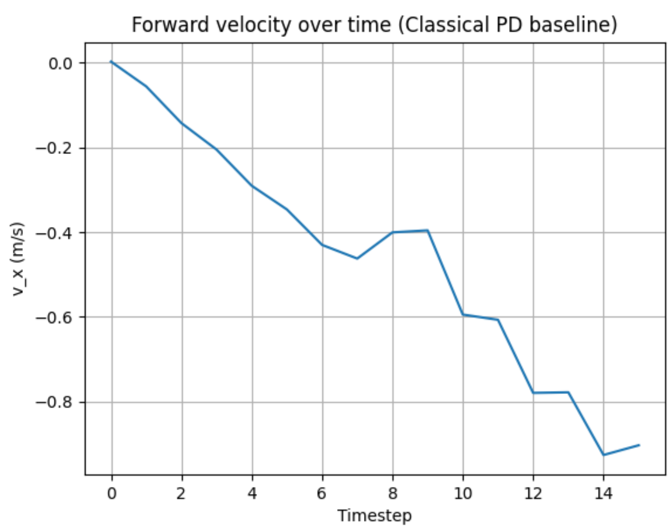

Fig.4:(Forward Velocity Over Time for The Classical PD Baseline Controller.)

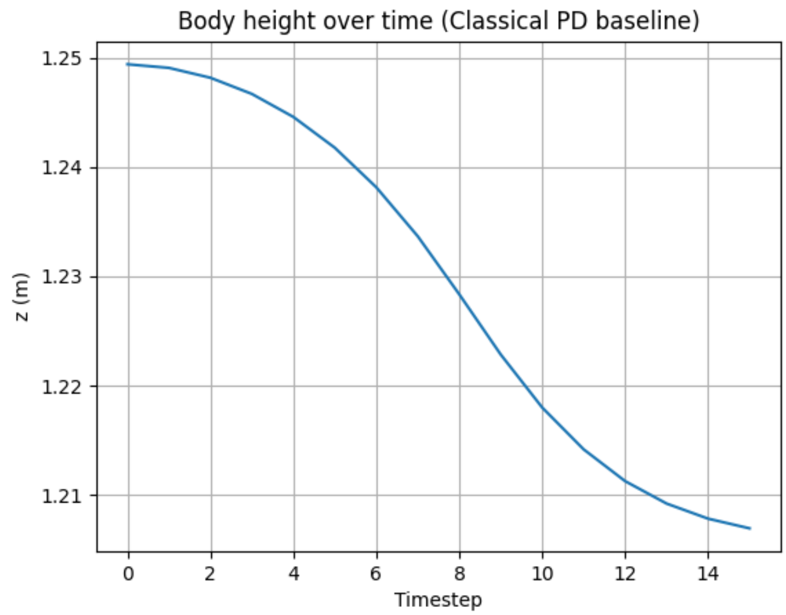

Fig.5: (Body Height Over Time for The Classical PD Baseline Controller.)

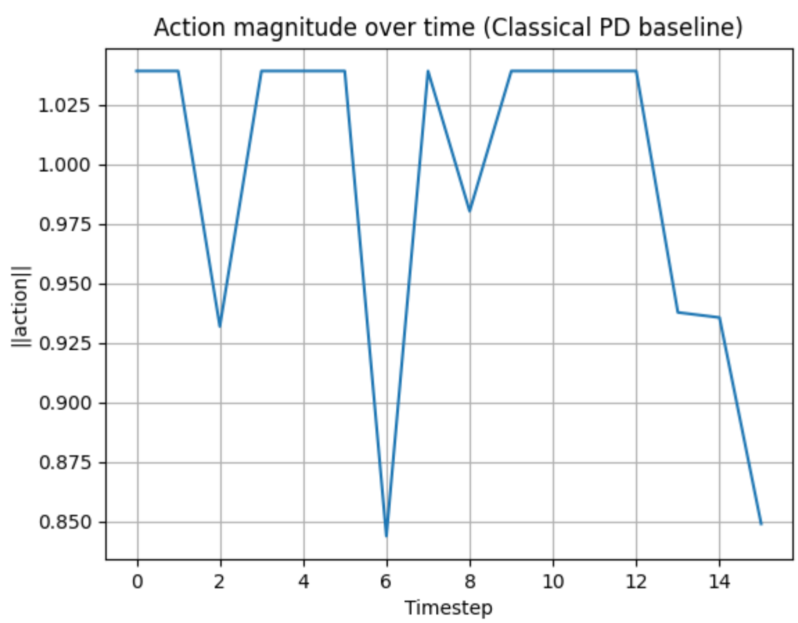

Fig.6: (Action magnitude over time for the classical PD baseline controller.)

5.3 Quantitative Comparison

Controller	Steps Survived	Avg Velocity (m/s)	Velocity Std (m/s)	Avg Action Magnitude
Classical PD baseline	16	−0.46	0.28	0.99
RL (PPO)	255	2.10	0.83	0.88

6. Discussion

The results demonstrate a clear performance gap between classical and reinforcement learning control strategies. The classical PD controller, while simple and interpretable, was highly sensitive to parameter tuning and lacked explicit phase-based logic. As a result, it failed to stabilize the hopper and produced inefficient control actions.

In contrast, the reinforcement learning controller learned an effective policy through interaction with the environment. The PPO policy exploited the system dynamics to maintain balance and generate forward propulsion, achieving significantly longer survival times and higher forward velocity with lower average control effort. Although moderate oscillations remained, the overall performance was markedly superior to the classical baseline.

The hybrid control concept offers a promising middle ground by combining structured classical control with learning-based parameter optimization. Such an approach could reduce learning complexity while preserving interpretability and robustness.

7. Conclusion

This project compared classical, reinforcement learning, and hybrid control approaches for Raibert's Hopper in the MuJoCo simulation environment. A classical PD-based controller served as a baseline but demonstrated limited stability and poor locomotion performance. In contrast, a PPO-trained reinforcement learning controller achieved stable hopping and effective forward velocity tracking. The comparison highlights the advantages of learning-based control for complex legged locomotion tasks and motivates future exploration of hybrid control strategies.

8. References

- Raibert, M. H., *Legged Robots That Balance*, MIT Press, Cambridge, MA, 1986.

- Raibert, M. H., Brown, H. B., and Chepponis, M., "Experiments in balance with a 3D one-legged hopping machine," *The International Journal of Robotics Research*, vol. 3, no. 2, pp. 75–92, 1984.

- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O., "Proximal Policy Optimization Algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

- Sutton, R. S., and Barto, A. G., *Reinforcement Learning: An Introduction*, 2nd ed., MIT Press, Cambridge, MA, 2018.

- Todorov, E., Erez, T., and Tassa, Y., "MuJoCo: A physics engine for model-based control," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 5026–5033, 2012.

- Tassa, Y., Erez, T., and Todorov, E., "Synthesis and stabilization of complex behaviors through online trajectory optimization," *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4906–4913, 2012.

- Kober, J., Bagnell, J. A., and Peters, J., "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1238–1274, 2013.

- Lillicrap, T. P., et al., "Continuous control with deep reinforcement learning," *International Conference on Learning Representations (ICLR)*, 2016.

- Hwangbo, J., Lee, J., and Hutter, M., "Per-contact iteration method for solving contact dynamics," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 895–902, 2018.

- Peng, X. B., Abbeel, P., Levine, S., and van de Panne, M., "DeepLoco: Dynamic locomotion skills using hierarchical deep reinforcement learning," *ACM Transactions on Graphics*, vol. 36, no. 4, pp. 1–13, 2017.