Department of Electrical Engineering and
Information Technology
Automatic Control Group
https://en.ei.uni-paderborn.de/de/rat/

Master Thesis

Differentiable Physics Simulators for Legged Locomotion

Background

Physics simulators, such as Isaac Sim [1], have been widely used in learning-based robotic control to address the data-intensive nature of training deep reinforcement learning (DRL) policies. A new class of simulators, known as differentiable physics simulators (DiffSim) [2], models the forward dynamics as differentiable functions, providing analytical gradient information for the learning objective w.r.t states and actions. Recent research [3, 4, 5] has explored the theoretical properties of this approach to calculate the objective function's gradients and investigated methods to efficiently integrate them into reinforcement learning (RL) algorithms. These approaches have demonstrated superior performance compared to model-free RL methods in traditional simulation environments. However, their application in real-world settings remains unexplored, presenting an exciting avenue for future research.

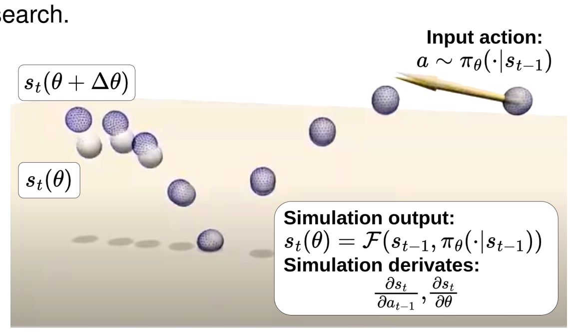

Figure 1: **A visualization of Diff Sim outputs**. Conceptually, one system dynamics (the simulator) to be a function $\mathscr{F}(s_{t-1}, a_{t-1})$ with $a_{t-1} \sim \pi_\theta(\cdot|s_{t-1})$. In this example, a differentiable simulator can provide gradient information w.r.t. the input sequence of action and the parameters θ of the policy network.

Project Overview

Project Goals

1. **Theoretical Knowledge.** Equip students with theoretical foundation of policy learning algorithms (PODS, SHAC) in DiffSim environments.
2. **Practical Simulation Skill.** Offer hands on training with in Warp (NVIDIA) to simulate quadruped robots and deploy the policy learning algorithms.
3. **Real-World Application.** Validate the learning framework by deploying on the quadruped robot Unitree Go2 as a case study.
4. **Learning Products.** Create a reproducible code base for learning quadruped robot locomotion using Warp/PyTorch.

Project Schedule

- Week 1-6: Literature review and background reinforcement.
- Week 7-13: Programming-focused training.
- Week 14-22: Experiments and analysis the obtained results.
- Week 23-24: Project wrap-up and preparation for thesis defense.

Requirements

- Background in reinforcement learning (especially actor-critic and policy optimization methods). Knowledge in control/system theory is not required, but a plus.
- Background in deep learning algorithms.
- Good Python programming skill (preferably experience with PyTorch).

References

[1] J. Liang, V. Makoviychuk, A. Handa, N. Chentanez, M. Macklin, and D. Fox, "Gpu-accelerated robotic simulation for distributed reinforcement learning," 2018.

[2] R. Newbury, J. Collins, K. He, J. Pan, I. Posner, D. Howard, and A. Cosgun, "A review of differentiable simulators," 2024.

[3] M. A. Z. Mora, M. Peychev, S. Ha, M. Vechev, and S. Coros, "Pods: Policy optimization via differentiable simulation," in *Proceedings of the 38th International Conference on Machine Learning*, Proceedings of Machine Learning Research, PMLR, 2021.

[4] I. Georgiev, K. Srinivasan, J. Xu, E. Heiden, and A. Garg, "Adaptive horizon actor-critic for policy learning in contact-rich differentiable simulation," 2024.

[5] J. Xu, V. Makoviychuk, Y. Narang, F. Ramos, W. Matusik, A. Garg, and M. Macklin, "Accelerated policy learning with parallel differentiable simulation," 2022.

Contact: Dinh Quang Dung, qdinh@mail.uni-paderborn.de, Room: P1.7.15.1