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

Master Project ET/CE/ESE

Simulating Robots with Differentiable Simulators

Background

Differentiable physics simulators (DiffSim) [1] are reshaping how we think about learning in robotics. By equipping physics engines with automatic differentiation, they make it possible to compute pathwise gradients—an alternative to the traditional score-function gradient that is typically unbiased and enjoys much lower variance [2]. Recent work [3, 4] has already shown that integrating DiffSim into RL pipelines can outperform conventional model-free approaches in benchmark simulations. Yet, the leap from DiffSim to the real setting remains largely untouched, leaving a promising ground for new research. A key obstacle is accessibility: while well-maintained frameworks exist for deploying RL learning scripts, there is still no comparable open and regularly supported DiffSim codebase for robot simulation such as [5]. To bridge this gap, our goal is to build a robust and extensible platform that brings the power of DiffSim closer to real-robot applications.

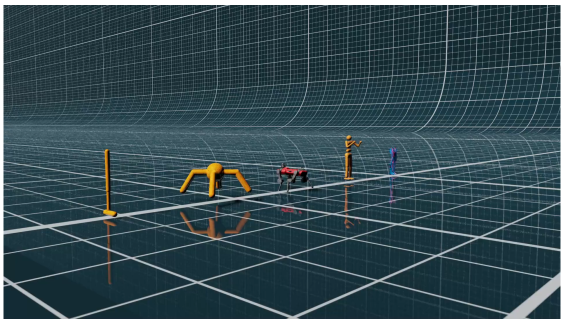
Figure 1: **Some robot models simulated by DiffSim**. Source: [3]

Project Overview

Project Goals

The main goal of the project course is to create scripts for modeling real robots using Warp [6] as DiffSim framework, which can then be utilized in future development of policy learning algorithms. The project can be readily extend to a Master Thesis, depending on the students' interests.

1. **Theoretical Knowledge.** Theoretical foundation necessary for physics simulation (e.g. necessary knowledge in robotics, physics, mathematics, etc.)
2. **Programming Skills.** Learn to use NVIDA Warp [6] and to perform interoperability with existing autograd frameworks such as PyTorch.
3. **Learning Products.** Python code base for reproducibility, scientific reports/papers.

Prerequisites

Some knowledge in the following fields are beneficial for you when joining this project couse, although they are not a must: control/system theory, robotics, deep learning, reinforcement learning, convex optimization. Programming skills are essential, since the main task of this project is simulation, which calls for knowledge in Python (PyTorch, JAX), C++ (CUDA) and/or any other simulation framework (Gazebo, IsaacSim).

What you get!

By joining this project course, you will explore a direction that is gaining attention in the robotics community. You will develop valuable knowledge and hands-on skills that provide a strong foundation for future research in robotics and related fields. Moreover, the project offers an extension into a Master's thesis depending on your interests. Finally, you will be working with enthusiastic researchers and have a chance to use our high-end computing resource to facilitate the research.

Contact

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

References

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

[2] S. Mohamed, M. Rosca, M. Figurnov, and A. Mnih, "Monte carlo gradient estimation in machine learning," 2020.

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

[4] Y. Song, S. Kim, and D. Scaramuzza, "Learning quadruped locomotion using differentiable simulation," 2024.

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

[6] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics." `https://github.com/nvidia/warp`, March 2022. NVIDIA GPU Technology Conference (GTC).

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