GPU-Based Simulation and Reinforcement Learning Pipeline for Tracked Ground Robots

David Korčák

Prague, January 2025

Introduction

- Original idea: train an RL policy for MARV or TRADR to control their flippers to minimize bouncing, angular speeds etc.
- What actually happened: neither of the simulators were ready to work with flippers and/or were buggy
- PyTorch simulator was also quite slow
- New objective: develop a PyTorch-based simulator able to handle flipper control, be more numerically stable, bug-free and significantly faster

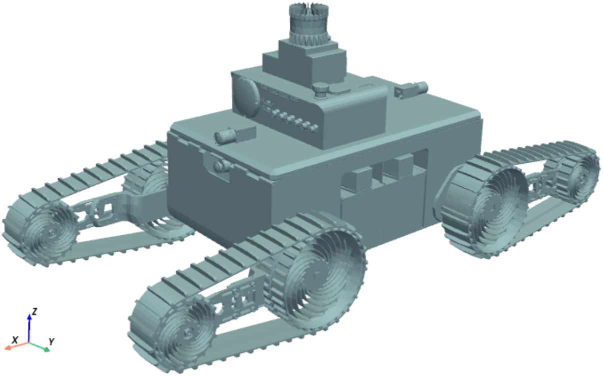
Background

- Why not use an existing simulation engine?
- Popular simulation engines like mujoco or brax are optimized for robots with joints and minimal collisions.
- No native support for tracked/wheeled robots and complex terrain collisions.

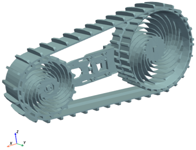
Simulation Engine

- Primarily designed for MARV and TRADR
- Simulation based on a pointcloud model of the robot.
- Implemented more elaborate and detail-preserving mesh-to-pointcloud conversion.
- Interactions modeled using spring-damper systems for terrain forces.
- Efficient computation of collision detection and dynamics.

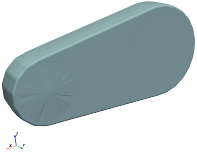
Mesh to Pointcloud Conversion



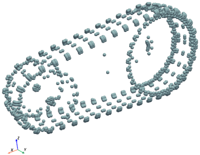
Mesh to Pointcloud Conversion



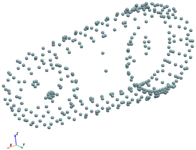
(a) Full flipper mesh.



(b) Delaunay triangulation



(c) Extracted surface points.

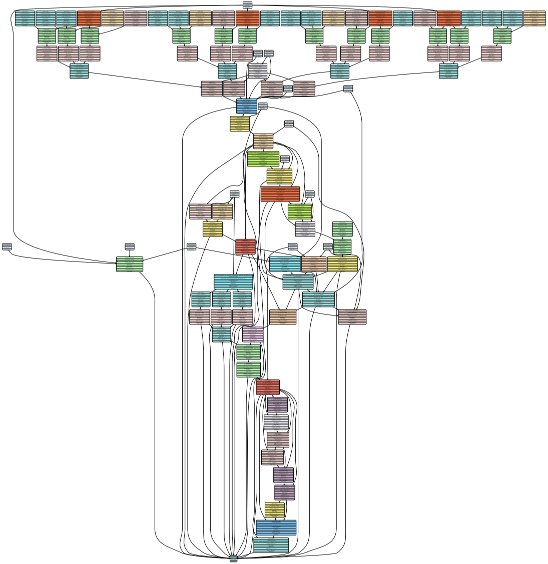


(d) Clustered surface points.

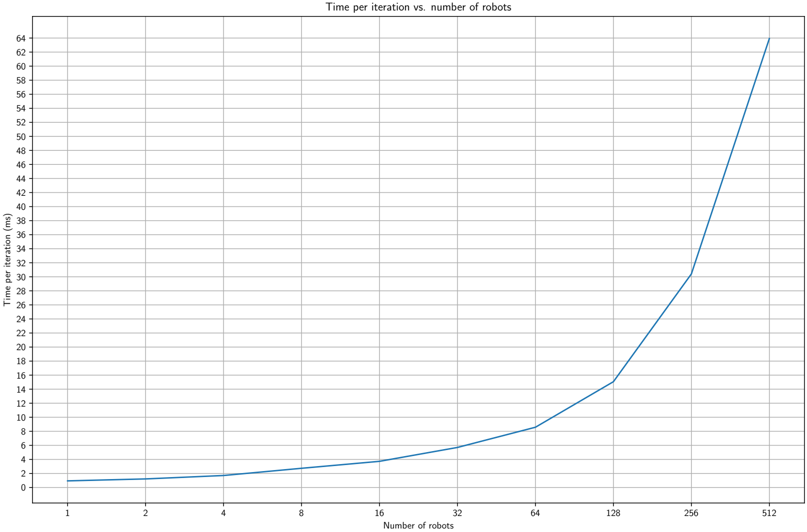
GPU Optimization

- Leveraging PyTorch for gpu-accelerated matrix operations.
- Conditional branching removed to avoid GPU thread divergence and halting.
- Torch's new compile functionality used for kernel optimization.

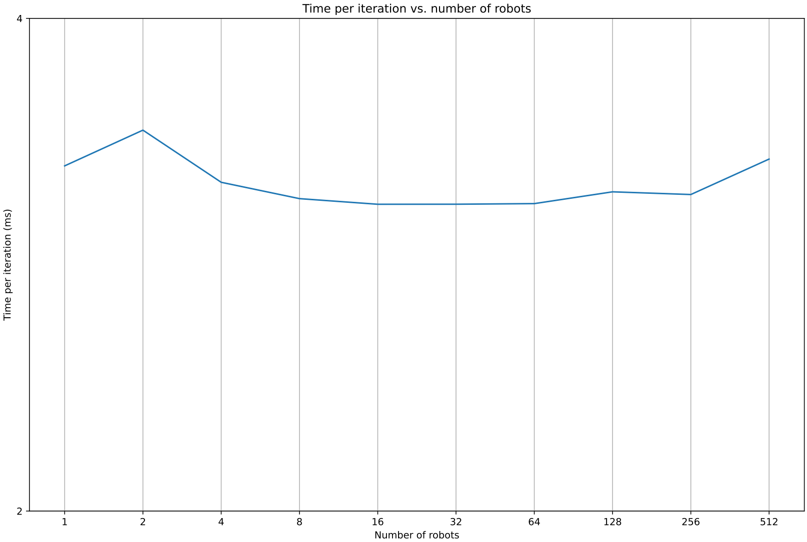
Forward Computational Graph



Performance Evaluation - CPU (M1 Pro)



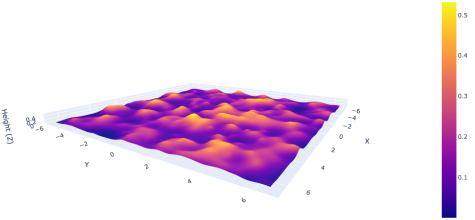
Performance Evaluation - GPU (A100)



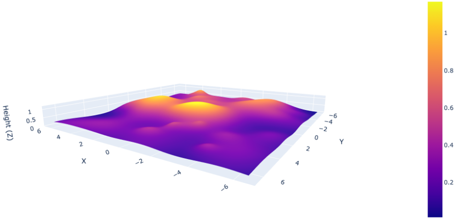
Reinforcement Learning Environment

- Implemented a basic RL framework and environment using TorchRL (native interop with Tensors, devices, datatypes, shapes).
- Observation types: bird-view heightmap camera and 3d lidar pointcloud.
- Random procedural terrain generation.

Procedural Terrain Generation



(a) Generated terrain with rough preset.



(b) Generated terrain with smooth preset.

Conclusion

- GPU-friendly simulation engine and basic rl environment developed.
- Future work: refine physics models and complete rl experiments.
- Merge with Aleš and his more advanced joint modeling