PyChrono and gym-chrono: a Deep Reinforcement Learning framework leveraging Multibody Dynamics to control Autonomous Vehicles and Robots.


*Dipartimento di Ingegneria ed Architettura, University of Parma, Parma, Italy
**University of Wisconsin – Madison, Madison, WI, USA

Abstract. gym-chrono is a set of simulated environments extending OpenAI Gym [1] with robotics and autonomous driving tasks. The physics of these environments is simulated thanks to Project Chrono [2], an open-source physics simulation engine capable of simulating Multibody Dynamics with contacts. The majority of most used Deep Learning frameworks (such as PyTorch and Tensorflow) have Python API. For this reason a condition for the creation of these environments has been the development of PyChrono, a Python module consisting of the Python bindings to Project Chrono C++ API.

Introduction

Reinforcement Learning (RL) is a Machine Learning technique based on agent-environment interactions: at each interaction the agent performs an action and collects the state of the system and a reward measuring its performance in solving some task; the goal of RL is, given the state, picking the action that maximizes the expected sum of reward, thus solving the task. In the last few years [3] Deep Learning used in conjunction with RL (called Deep Reinforcement Learning, DRL) has demonstrated to be a viable approach to solve complex real-world robotics tasks [4]. DRL methods, like any other Deep Learning approach, require large dataset to optimize the Neural Networks, thus arising interest in physics engine providing a Python API. The DRL community heavily relies MuJoCo and PyBullet for robotics environments and on CARLA and AirSim for autonomous driving.

We provide in a single Python framework a set of reinforcement learning environments that feature: (1) multi-body dynamics simulation (2) deformable bodies simulation, (3) the capability of importing 3D CAD models (4) vehicle dynamics (5) sensors simulation.

Results and discussion

Together with a concerted effort to improve the Python wrappers of Chrono, that lead to an Anaconda-distributed package with a good user base [5], we built a set of increasingly challenging DRL environments, and used state-of-the-art continuous actions DRL algorithms to solve them. The first step has been building and solving environments such as the inverted pendulum and the 4-legged walker [6]. Then, we included models of real 6-DOF robots by leveraging the tools for 3D CAD parsing of Chrono [7]. This feature proved to be useful by making changes in the model extremely easy to be passed to the training environment.

The latest development are about autonomous driving in off-road conditions, simulating vehicle dynamics and terrain deformation.

References