Modular Robot Snake Training Environment and Reinforcement Learning of Simple Locomotion

Tomas Musil

December 18, 2022

1 Background

Reinforcement learning has widely been used in creating control policies for robots for a wide variety of problems, such as walking of quadrupeds (TODO), walking of spider-like robots (TODO) and even as controllers for quadrotor flying robots (TODO).

In this project, I concerned myself with training a modular robot.

The robot system with very affordable parts, open-source design and simple communication and software stack was constructed by me and my 3 other colleagues at the Czech Technical University. It is also a reason for me taking this course, where I intended to learn more about RL and apply it to creating a RL-based controller for this robot system.

Reinforcement learning is particularly of interest for robots for which it is hard to create a model by humans using mathematics.

2 Problem Definition

The goal of this project is to:

- Construct a custom gym environment which would allow effortless definition and spawning of different configuration of the modular robot which could be transferred to the real robot, and have multiple terrains on which to learn movement.
- Train moving in one type of terrain for multiple robot configurations to see how the number of degrees of freedom of observation and actions affect the training.
- Train on two types of terrain using two different RL algorithms. First a simple uneven terrain simulating for example the floor of a forest. Secondly on an environment with stairs.

3 Methods

3.1 Custom AI Gym Environment Construction

I developed multiple terrains using the open-source software Blender (TODO). In this report, I trained the robot at 2 terrains - stairs and bumpy terrain, which can be seen in TODO.

3.2 Training

For training in both tasks, I decided to use the library TODO. I experimented with multiple RL algorithms and found PPO (TODO) and TODO to be the most efficient at learning motions in the created environments.

4 Results

First round - stairs, long, 2 different algos (2g) Second round - bumps, 1 algo, 2 different configurations (2g) 3rd round - bumps, 2 algo, 2 different configurations (4g) [?] VIDEO?

5 Discussion

Mention - happy that I was able to make it learn on the stairs and on the bumps. Fun would be putting it on real HW and testing its movement and trying to bridge the reality gap (different masses, powers of motors, delays, springiness, ...) which would take a lot of effort, maybe for one full masters thesis.

References

[1] J. Schulman *et al.*, “Proximal Policy Optimization Algorithms,” 07 2017.