Bipedal Robot Control via Deep Reinforcement Learning

Final Project Report for CS 5824 Project

Bokun Zheng
Mechanical Engineering, Virginia Tech
Blacksburg, VA
bokunz@vt.edu

1 Abstract

In this project, we reviewed about the legged robot locomotion and deep reinforcement learning. We also conducted the case studies on various robots controlled by various algorithms from the literature. Later, we used the OpenAI gym to train a humanoid robot simulated in the Mujoco environment. The result shows the TQC has the best training performance. While the off-policy learning algorithms like SAC and TD3 can better handle the high-dimensional robotics system than the PPO.

2 Introduction

Continuous control on legged robots is always a challenging problem because of its high-dimensional nonlinear trait. Traditional model-based controllers require highly-simplified assumptions and careful modelling. While recent advances in machine learning have offered new approach for control problems [8]. The bipedal robot, which is inspired from human and other animals' locomotion ability across various types of terrain, is one of successful example that integrate ML methods. The complex interaction between the robot and the terrain is perfect for setting up the agent-environment structure of reinforcement learning. The further extension of using deep neural networking in the structure can well handle the challenges such as the high dimension, nonlinearity and uncertainty.

This project will be divided into two parts: the first part (Section 3) is a literature review about the backgrounds of bipedal robots and state-of-art applications with DRL algorithms. The part mostly reuses the milestone report. The second part is a practice of applying DRL to an bipedal agent. Section 4 will deliver the method and Section 5 will deliver the result along with discussion.

3 Background

3.1 Legged Robots Locomotion

The mobility of land robots mainly depends on two approaches: wheels and legs. While wheels are really efficient to operate in city, indoor and other flat environments, they will be in trouble in other complex terrains, such as irregular, deformable and slippery surface. Legged robots are developed in order to be capable of those complexities. Legged robots can be classified by the number of legs into bipedal, quadruped and hexapod, which are closely related to the bio-inspiration, for example, human [9], cheetah [15] and cockroach [13].

For the legged locomotion, there exist two biggest challenges: stability and controllability. Typically, with more legs, the system will be more stable, but more complex for controllers. The conventional controllers are mostly subject to a certain nonlinear dynamic model, with poor generalization into different robots or terrain. However, the increasing application of machine learning algorithms,

Preprint. Under review.

especially deep reinforcement learning, can directly map between the sensory information with low-level actuation inputs.

3.2 Deep Reinforcement Learning

DRL combines the RL and deep neural network, is a model-free method for a controller. RL is based on Markov Decision Process, where the agents make decision within the state space. The policy aims to maximize the total rewards. However, control problems often involve large inputs, that makes DRL more suitable for this kind of problem. Deep RL uses multi-layer neural networks to approximate the complex functions such as policies, and then trains the parameters with a popular class of methods to optimize the neural network, such as policy gradient method [16].

3.3 Popular Structures

The DRL has many variants of structures. Here we surveyed several applications to legged robots. Kohl et al [6] did the early attempt to optimize the robot's gait from ML approach. They applied policy gradient to automatically search for the optimized parameters of the controller. However, his work is limited by the robot platform, Sony Aibo, which is not advanced in locomotion from today's view. Xie et al [19] combined the conventional feedback control with DRL. They used the popular critic-actor structure and trained the parameter by Proximal Policy Optimization (PPO) [14]. Haarnoja et al [3] extend the maximum entropy RL to achieve an improved robustness on quadrupedal walking gaits. Lee et al [7] also studies the quadrupedal locomotion under blind terrain conditions. They applied the framework of privileged learning, where a teacher policy was trained with knowledge of ground truth, and then let the teacher policy guides a student policy. Hwangbo et al [5] chose the Trust Region Policy Optimization (TRPO) trained in simulation and validated on real robots.

3.4 Experimental Platform

The selection of experimental platform is one issue. Directly applying to DRL to real-world robots is extremely difficult since the training process demands large trials and samplings, which is time consuming and harmful to the tested robots which are sophisticated and expensive. Thus, most of those studies train their policies in simulation. A survey on robotics simulation tool has compared several popular physical engines, such as Bullet, Havok, MuJoCo, ODE and PhysX [2].

The choice of robot models is another issue. Human are always interested in building robots that mimic animals including ourselves. There were a few examples of earlier exploration, such as Honda ASIMO and Sony Aibo. They lived up to our imaginations about robots, but lack the knowledge of principles of legged locomotion. So the successors are designed to compliant the bio-inspiration, for example, the Spring Loaded Inverted Pendulum (SLIP) model of running gaits. Recently, there are more legged robots specially for studying legged locomotion. From the paper we have reviewed, the researchers used the ANYmal [4] [3] [5] and bipedal Cassie [19] [20].

4 Methodology

4.1 Environment setup

From the previous survey about the popular robot and simulation tool used in the literature, we chose to use the integration of OpenAI gym [1] and Mujoco simulation [18] to study the Cassie robot. The workspace was setup in Ubuntu 18.04, with Gym 0.22 and Mujoco 2.10 install within a Conda environment. We also borrowed from open source packages [12] to simulate the cassie robot in Mujoco (Fig. 1).

To integrate the Mujoco with Gym, the simulation needs to be wrapped into a customized gym environment. However as we actually trying to setup the environment, we found that the difficulty was largely underestimated from the beginning, even though it is possible. The main challenges come from mapping the control and designing the rewards. The Mujoco is a model-based simulation, even though the RL is model-free, a careful mapping between the control efforts and joint angles is still needed for each step of RL. The reward design is also highly associated with the kinematics. Those

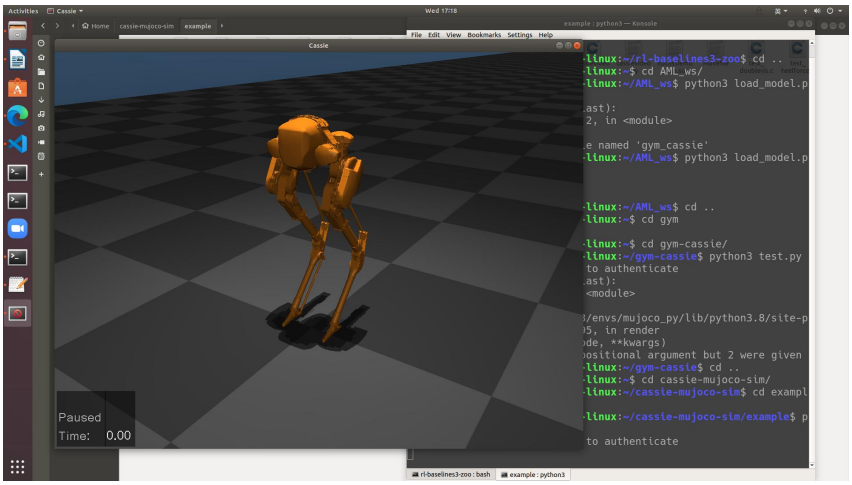

Figure 1: **Cassie robot simulated in Mujoco UI**

work needs a prior knowledge of the robot structure in details. So it seems to be not feasible with in the timeline of this project.

This prompted us to change the direction and finding other available models. With the prior experience of working with gym and Mujoco, we chose to use the "Humanoid" environment [17] that has been integrated with the gym.

The Humanoid (Fig. 2) is a 3d model with 22 DoFs. It is 1.6m tall and weighs 55Kg. It resembles the human skeleton with simplified main joints. The goal of the humanoid environment is: 1) keep the humanoid standing and balancing. 2) make the humanoid walk forward as fast as possible. By using this environment, we are able to train the agent using the DRL algorithms from the literature.

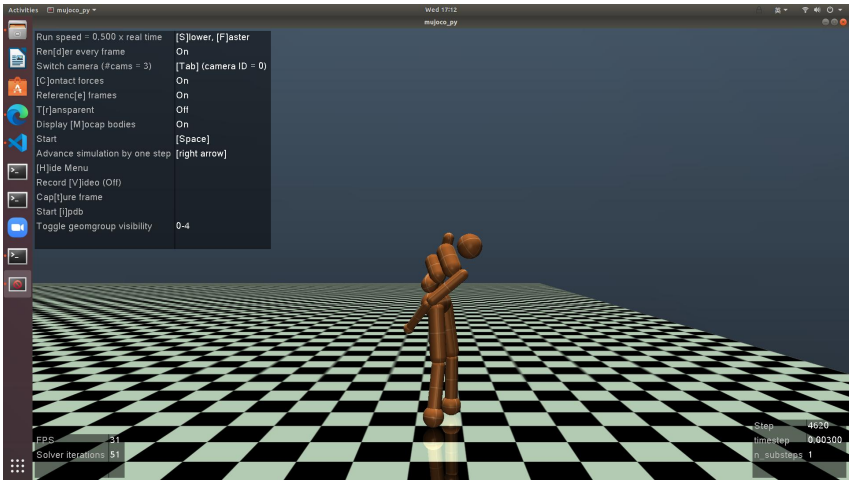

Figure 2: **A walking humanoid simulated in Mujoco**

4.2 Training

Both the observation and action space is a continuous box space, as the feature of a continuous control problem. From the literature as well as knowledge on the lecture, we chose to use different algorithms to train the humanoid:

- Proximal Policy Optimization (PPO)
- Soft Actor-Critic (SAC)

- Twin Delayed DDPG (TD3)
- Truncated Quantile Critics (TQC)

For the training, we rely on the DL library Stable Baseline3 [11] and RL Baseline3 Zoo [10] since they can directly work on the gym environment. They also provide useful tools such as hyper-parameter tunning and plotting. Here, we train the humanoid using each algorithms with 1000000 total steps.

4.3 Result and discussion

For each four algorithms we used, the trained humanoid was able to stand with balance and walk forward for a limited time and then fell down. To evaluate the training result, each trained agent will run totally 10000 steps, which covers several independent episodes. A new episodes will start when the previous one fails. Along those episodes, we recorded the number of episodes, mean episodes steps, and the mean episode rewards, summarized in Table 1.

Algorithm	episodes	Mean steps	Mean rewards	Best model rewards
SAC	11	819.36	4227	5349
PPO	80	124.74	722.37	1000
TD3	14	670.86	3455	5159
TQC	11	885	5474	6204

Table 1: Training results

Due to the wrapper of the Mujoco environment, the maximum episode length is limited to 1000. Thus, the maximum mean step is 1000 in Table 1. It clearly shows that the PPO has the worst training result since each episode will fail around 124 steps. It is likely that the hyper-parameters are not the optimal. The SAC, TD3 and TQC trained the model significantly better. Most episodes lasted for the full 1000 length. We also evaluated the best model rewards, which is the highest rewards it can achieve within 1000 steps. From the rewards, it can be calculated the TQC > SAC > TD3 > PPO. Also, according to the behaviour in Mujoco simulation of the best model, The TQC ran the fastest, and then SAC, TD3 and PPO. Which means the best rewards will reflect the walking speed.

The baselines package also provides the plotting tool that can visualize the reward changes during training. A comparison of four algorithms is plotted in figure 3. It further confirms the training performance difference of four algorithms. The TQC get the most outstanding performance since it is based on both TD3 and SAC. The SAC and TD3 is somehow similar. while both of them use the off-policy learning, SAC and TD3 are much more efficient to deal with the high-dimensional robot systems, than the PPO which is using the on-policy learning.

This project has helped we accumulate experience in working around linux, OpenAI gym, Mujoco and other python packages. We also have a more concrete understanding about the RL concepts, as well as different algorithms. Still, this project has some drawbacks. We didn't successfully setup a customized gym environment on a legged robot. Also, the training steps are limited to the computational resource.

References

[1] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

[2] Tom Erez, Yuval Tassa, and Emanuel Todorov. Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 4397–4404. IEEE, 2015.

[3] Tuomas Haarnoja, Sehoon Ha, Aurick Zhou, Jie Tan, George Tucker, and Sergey Levine. Learning to walk via deep reinforcement learning. *arXiv preprint arXiv:1812.11103*, 2018.

[4] Marco Hutter, Christian Gehring, Dominic Jud, Andreas Lauber, C Dario Bellicoso, Vassilios Tsounis, Jemin Hwangbo, Karen Bodie, Peter Fankhauser, Michael Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 38–44. IEEE, 2016.

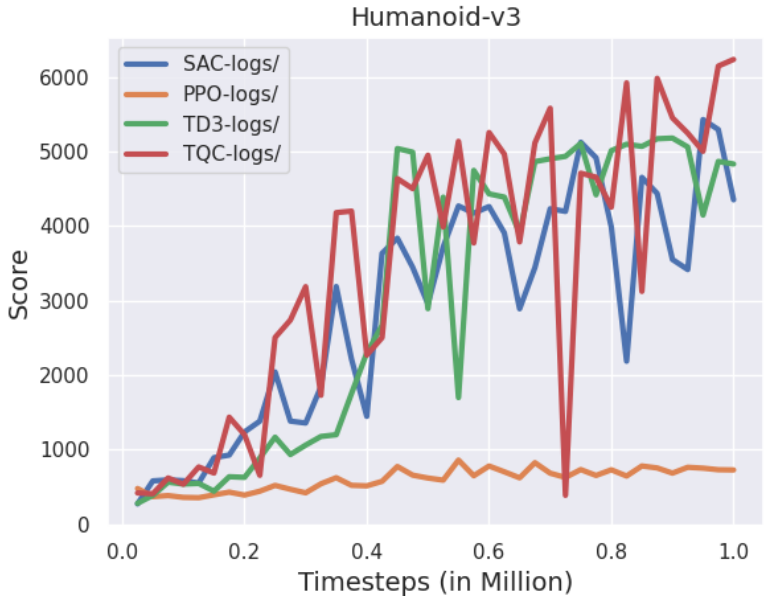

Figure 3: **Rewards changing over timesteps during training**

[5] Jemin Hwangbo, Joonho Lee, Alexey Dosovitskiy, Dario Bellicoso, Vassilios Tsounis, Vladlen Koltun, and Marco Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872, 2019.

[6] Nate Kohl and Peter Stone. Policy gradient reinforcement learning for fast quadrupedal locomotion. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA'04. 2004*, volume 3, pages 2619–2624. IEEE, 2004.

[7] Joonho Lee, Jemin Hwangbo, Lorenz Wellhausen, Vladlen Koltun, and Marco Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[8] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[9] Gabe Nelson, Aaron Saunders, Neil Neville, Ben Swilling, Joe Bondaryk, Devin Billings, Chris Lee, Robert Playter, and Marc Raibert. Petman: A humanoid robot for testing chemical protective clothing. *Journal of the Robotics Society of Japan*, 30(4):372–377, 2012.

[10] Antonin Raffin. Rl baselines3 zoo. `https://github.com/DLR-RM/rl-baselines3-zoo`, 2020.

[11] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[12] Agility Robotics. cassie-mujoco-sim, 2018.

[13] Uluc Saranli, Martin Buehler, and Daniel E Koditschek. Rhex: A simple and highly mobile hexapod robot. *The International Journal of Robotics Research*, 20(7):616–631, 2001.

[14] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[15] Sangok Seok, Albert Wang, Meng Yee Chuah, David Otten, Jeffrey Lang, and Sangbae Kim. Design principles for highly efficient quadrupeds and implementation on the mit cheetah robot. In *2013 IEEE International Conference on Robotics and Automation*, pages 3307–3312. IEEE, 2013.

[16] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[17] Yuval Tassa, Tom Erez, and Emanuel Todorov. Synthesis and stabilization of complex behaviors through online trajectory optimization. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4906–4913. IEEE, 2012.

[18] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[19] Zhaoming Xie, Glen Berseth, Patrick Clary, Jonathan Hurst, and Michiel van de Panne. Feedback control for cassie with deep reinforcement learning. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1241–1246, 2018.

[20] Zhaoming Xie, Patrick Clary, Jeremy Dao, Pedro Morais, Jonanthan Hurst, and Michiel Panne. Learning locomotion skills for cassie: Iterative design and sim-to-real. In *Conference on Robot Learning*, pages 317–329. PMLR, 2020.