A Recurrent Deterministic Policy Gradient Method for Bipedal Locomotion on the Rough Terrain Challenge

Doo Re Song, Chuanyu Yang, Christopher McGreavy, Kai Yuan, Zhibin Li

***Abstract*— Most bipedal walking controls are achieved using deterministic and analytic engineering approaches. Engineering based approaches require quite a lot of human effort in designing the controllers, which is a disadvantage. On the other hand, machine learning approaches, e.g. deep RL, require less manual tuning. Recent works on DRL have demonstrated the capability of learning very complex and dynamic motor tasks. Therefore, we are motivated to explore the feasibility of producing bipedal locomotion policies using deep reinforcement learning.**

I. Introduction

The morphology of humanoid robots is similar to that of humans which provides the ability to traverse complex and dynamic terrains that are easily accessible to humans. Humanoid robots generally exhibit high manoeuvrability and flexibility, thus are capable of achieving locomotion while navigating through uneven terrain and stepping over obstacles. Considering the physical limitations of wheeled robots, there are many advantages to choosing bipedal locomotion over wheeled locomotion. Moreover, knowledge of bipedal locomotion can also help us design better exoskeletons that can benefit the lives of people with gait abnormalities. As a result, bipedal locomotion has attracted increasing attention in recent years.

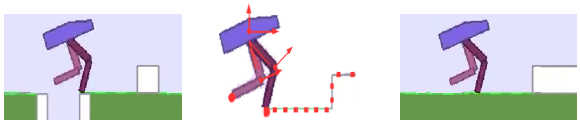
Most bipedal walking controls are achieved using deterministic and analytic engineering approaches. Yet, there are also a large amount of works that attempt to use machine learning approaches, such as reinforcement learning (RL), to achieve bipedal walking. RL can be applied to model-free learning for bipedal walking. This has been attempted in various action policy learning algorithms based on Markov Decision Process (MDP).

A variety of work by computer science and robotics researchers has used DRL to solve bipedal locomotion. However, the MDP framework of RL lacks some necessary components for walking. During sensing, a walking agent is unable to fully observe its environment (Fig.1b), meaning inferences must be made about the environment using previous observations and actions made by the agent to fill in the gaps in observability. In reality, robots never gather sufficient information to generate optimal actions, and generally the obtained information is usually noisy as well. Therefore, partial observability is a critical issue in robot locomotion when RL is applied to real world robotics.

*This work is supported by the Future AI and Robotics Hub for Space (EP/R026092/1) and UK Robotics and Artificial Intelligence Hub for Offshore Energy Asset Integrity Management (EP/R026173/1) funded by the Engineering and Physical Sciences Research Council (EPSRC)

The authors are with the School of Informatics, The University of Edinburgh, UK.

Email: sdr2002@gmail.com, {chuanyu.yang, c.mcgreavy, kai.yuan, zhibin.li}@ed.ac.uk



(a) (b) (c)

Fig. 1: Partially observable scenario: (a) Real world environment; (b) Incomplete observation of environment; (c) Incorrect perception built on the incomplete observation;.

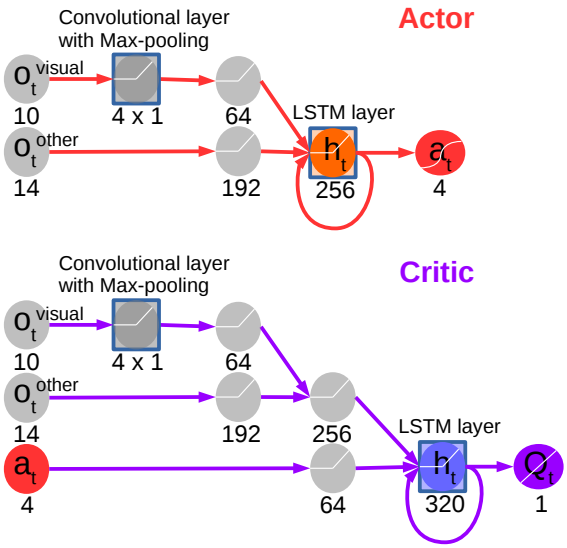


Fig. 2: Network design of Actor and Critic.

II. Learning Methods

A. Recurrent deterministic policy gradient algorithm

Partially Observable Markov Decision Process (POMDP) has to be introduced to Reinforcement learning in order to address the issue of incomplete observability in the environment.

The reinforcement learning algorithm we used is called Recurrent deterministic policy gradient (RDPG) algorithm, proposed by Heess et.al [1], which is a combination of Long Short Term Memory network (LSTM) [2] and Deterministic Policy Gradient (DPG) [3].

B. Network structure

The reinforcement learning agent consists of an Actor network and a Critic network: one to generate the action (Actor) and the other to generate the Q-value for evaluation

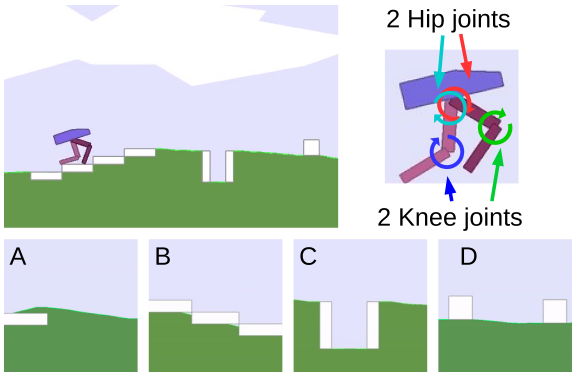


Fig. 3: Terrain feature. (A) Slope, (B) Stair, (C) Gap, (D) Hurdles.

(Critic). Both the actor network and critic network consists of an Long short term memory (LSTM) layer (Fig.2).

III. Simulation Environment Setup

A. *OpenAI Gym bipedal walker environment*

The bipedal walking task that we aim to solve is the OpenAI's Bipedal-Walker challenge[1]. The 2D simulation environment is partially observable to the bipedal walking agent.

The bipedal character has 4 degrees of freedom, which includes 2 hip joints and 2 knee joints. The simulation environment provides 24-dimensional sensory feedback. This information consists of 10-D LIDAR (visual) with limited range, 4-D translational/rotational displacement and velocity of hull, 8-D rotational displacement and velocity of the joints, and 2-D binary haptic feedback of the feet. The control loop runs the same frequency as the physics simulation at 50Hz.

The goal of the challenge for the bipedal agent is to traverse a variety of rugged terrains (Fig.3) without falling. Our algorithm is tested on hardcore/difficult mode. The environment runs episodically, ie an episode terminates if: the body of robot touches the environment, or the agent reaches the goal, or the maximum runtime (40s) is out.

IV. Simulation Results

Our DRL framework has successfully trained policies for a stable and dynamic locomotion and is capable of solving the partially-observed bipedal walking tasks. It is capable of negotiating a diversity of terrain features including slopes, stairs, gaps and hurdles in a very agile manner.

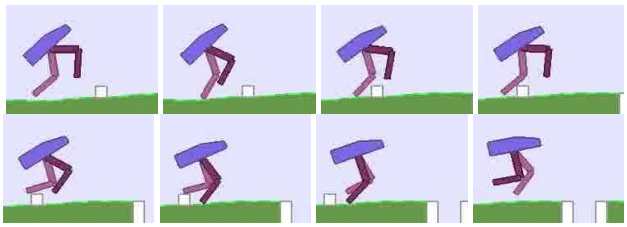
References

[1] N. Heess, J. J. Hunt, T. P. Lillicrap, and D. Silver, "Memory-based control with recurrent neural networks," *arXiv preprint arXiv:1512.04455*, 2015.

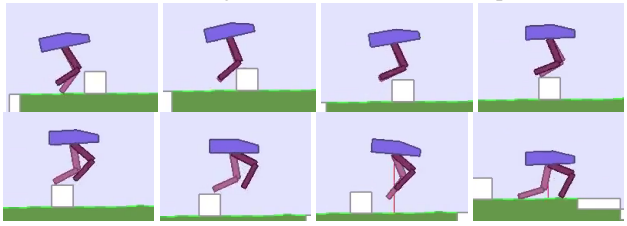
[2] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[3] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 387–395.

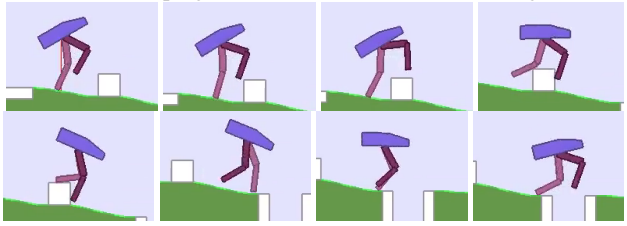
[1]https://gym.openai.com/envs/BipedalWalkerHardcore-v2



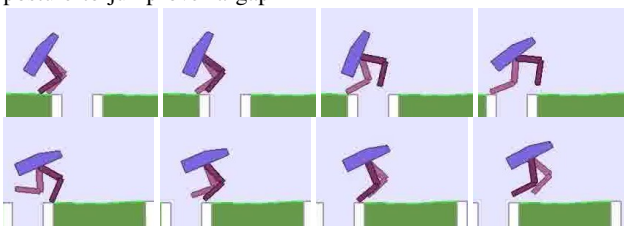
(a) Walking on flat terrain with a bump



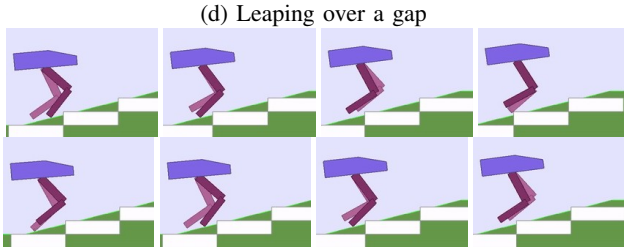
(b) Jumping over a hurdle with stable landing



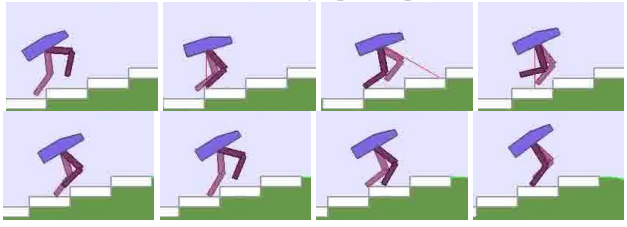
(c) Crossing over a hurdle followed by agile adjustment of posture to jump over a gap



(d) Leaping over a gap



(e) Running up a slope



(f) Climbing up stairs

Fig. 4: Terrain specific agile behaviors.