

ML-Agents 를 이용한 밸런스 로봇 학습 및 구현 Training and Implementation of Self-Balancing Robot with ML-Agents

○윤 혁¹, 한 지 형^{2*}

¹⁾ 서울과학기술대 컴퓨터공학과 (TEL: 010-2907-9705; E-mail: titania7777@seoultech.ac.kr)

²⁾ 서울과학기술대 컴퓨터공학과 (TEL: 02-970-6705; E-mail: jhhan@seoultech.ac.kr)

Abstract In this paper, we will show that it's possible to implement self-balancing robot on real world using the ML-Agents[1]. Currently, there are many robot simulators, but we will use Unity [2], which is famous for its game engine, as a learning environment for robot simulators. For this work we design a simple Roly-Poly model in Unity that can help to reduce complexity in environment and improve training speed. Also, to implement trained Roly-Poly model on real world we use only serial communication based on our simple protocol without ROS. We conduct the experiment to demonstrate the possibility of actual implementation of a learned robot through ML-Agents.

Keywords reinforcement learning, robot intelligence, balance robot

1. 서론

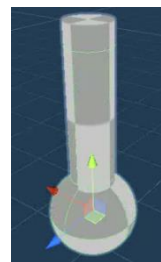
최근 로봇에 대한 연구가 많이 이루어지고 있고 이에 따라서 로봇의 구현 및 구동테스트에 시간과 비용을 절약하기 위한 Gazebo와 같은 로봇 시뮬레이터들이 많이 등장하였다. 또한 이들 시뮬레이터는 지능적인 로봇의 구현을 위해 강화학습 알고리즘과의 연동을 지원한다.

하지만 강화학습은 다른 지도 및 비지도 학습과는 다르게 따로 정답이 주어지지 않고 오직 환경과 에이전트의 상호작용을 통하여 학습이 이루어져 환경의 구성이 에이전트의 학습의 영향을 많이 준다.[3] 이러한 문제로 강화학습을 위한 많은 환경들이 등장하였고 그중 ML-Agents 는 게임엔진으로 유명한 유니티를 에이전트의 환경 구축에 사용하는 강화학습 프레임워크이다. 특히 게임엔진이라는 점에서 환경구축의 큰 유연성을 가진다는 장점이 있고 현재도 빠르게 발전하고 있다.

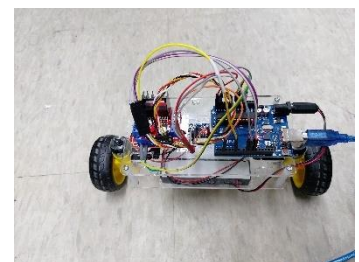
본 논문에서는 이러한 장점을 가진 ML-Agents 를 이용하여 밸런스로봇의 구현을 하기위한 핵심 컴포넌트만 분리하여 새롭게 디자인한 Roly-Poly 모델을 학습하고 학습된 에이전트를 시리얼 통신으로 그림 1 의 (b)와 같이 실제 밸런스로봇과 연결하여 다른 로봇 시뮬레이터없이 ML-Agents 로도 충분히 로봇의 학습부터 실제 구현까지 가능함을 보일 것이다.

2. 밸런스 로봇 학습 및 구현

밸런스 로봇의 학습을 위해서 그림 1의 (a)와 같이 Roly-Poly 모델을 구현하였다. 밸런스 로봇은 한 축을 기준으로 좌, 우 만을 움직임으로 이 외의 축의 정보는 필요가 없다. 이와 같이 Roly-Poly 모델도 z축으로 좌,



(a) Roly-Poly



(b) 밸런스 로봇

그림 1. 학습을 위한 Roly-Poly모델과 구현된 밸런스 로봇

우 로만 움직일 수가 있으며 다른 축은 고정되어도록 디자인하였다.

이렇게 디자인된 Roly-Poly 모델은 빠른 학습을 위해서 하나의 브레인에 여러 개의 에이전트 즉 여러 개의 Roly-Poly 모델의 프리팹을 복사하여 붙여 넣고 학습을 진행하게 된다. 이를 통해 복사된 여러 개의 에이전트는 각각 총 스텝을 분담해서 빠르게 학습하게 된다.

학습진행 과정에서 학습 상태와 보상의 설정은 먼저 다른 연구를 살펴보면 [4]에서는 상태로 한 축의 각도만을 액션으로는 속도를 이용하였고 [5]에서는 상태로 변위, 속도, 각도, 각속도를 입력으로 받고 액션으로는 가속도를 이용하였다. 또 다른 연구 [6]에서는 상태로 각도, 각속도를 액션으로 모터의 PWM 에 따른 회전을 선택하였다.

특히 강화학습의 특성상 [5]와 같이 많은 상태가 주어지면 좀더 많은 변수를 고려하여 정교하게 학습이 가능하지만 실제 밸런스 로봇은 ML-Agents 의 모델 학습 및 구현의 증명을 위해서 사용하는 간단한 모델이기 때문에 입력 상태로는 관측 각도와 각속도면 충분하다. 그러므로 본 논문에서는 학습을 진행할 때 에이전트는 현재의 각도와 각속도

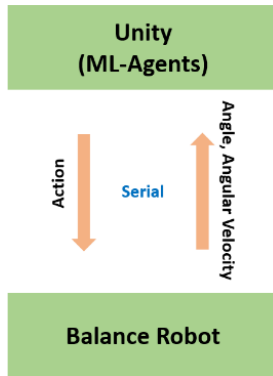


그림 2. 유니티와 밸런스 로봇과의 통신구현도

를 상태로 받고 액션은 각도를 유지하기 위해 더해야 할 각속도이다.

에이전트의 학습이 완료되면 에이전트는 관측할 수 있는 현재의 각도와 각속도를 기준으로 어떤 방향으로 각속도를 더해야 넘어지지 않는지를 알려 실제구현에는 이런 에이전트의 정보를 그림 2와 같이 실시간으로 교환만 하면 된다. 다만 구현에 있어서 유니티에 구축한 모델의 질량과 중심축 위치 등 실제 물리량과 다소 차이가 있기 때문에 데이터를 전송할 시 약간의 파라미터를 추가해서 전송을 하였다. 또한 각도를 측정하는데 있어서 실제 자이로스코프의 데이터를 그대로 사용하면 노이즈가 심해 바로 사용할 수가 없으므로 칼만필터를 거쳐 나온 각도를 전송하도록 하였다. 그리고 빠르고 정확한 데이터 전송을 하기위해서 무선통신 대신 유선 시리얼 통신을 사용하였고 통신속도는 115200 보레이트를 사용하였다.

3. 실험 및 결과

실험에 사용된 학습 파라미터는 ML-Agents 에서 기본으로 제공되는 PPO 알고리즘[7]을 사용하였고 Roly-Poly 모델은 복잡한 모델이 아니기에 배치 사이즈는 16, 버퍼 사이즈는 128 로 설정하였다. 그리고 총 스텝은 500K 스텝이고 에이전트의 에피소드 당 최대 스텝은 1000 으로 설정을 하였으며 리워드는 매 스텝마다 각도 $\pm 1.5^\circ$ 범위를 유지하면 1을 아니면 에피소드를 끝내도록 하였다. 또 추가적인 내용은 <https://github.com/titania7777/Self-Balancing-Robot> 에서 코드로 확인할 수가 있으며 <https://www.youtube.com/watch?v=zNuGCi0jJcc> 에서 실제 동작하는 영상을 확인할 수가 있다.

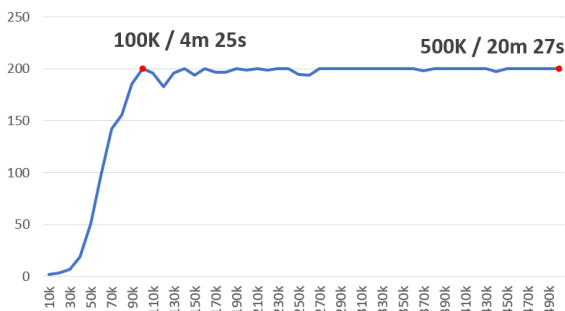


그림 3. 스텝에 따른 누적보상

그림 3 은 스텝에 따른 누적보상 그래프이며 100K 스텝에서 총 200 리워드를 받았고 그 뒤로 쭉 수렴하는 모습을 보인다. 또한 100K 스텝까지 학습하는데 4분 25초가 소요되었다. 그리고 뒤로 총 스텝인 500K 까지 학습하는데 20분 27초가 소요됨을 확인할 수가 있다.

4. 결론

현재까지 많은 로봇 시뮬레이터가 존재하지만 대부분은 강화학습을 위한 정교한 환경 구축이 불가능하다. 예를 들어 자율주행을 위한 환경은 단순 장애물을 피하는 것으로 끝나면 안되고 실제 도로를 최대한 비슷하게 디자인하고 시간대에 따른 차량의 통행량 날씨 등을 고려해야하는 변수와 또 CNN 을 사용하여 상황과악 시뮬레이션을 가동해야 할 경우 실제와 비슷한 그래픽을 연출하는 강력한 그래픽 기술이 필요하다. 이러한 기능은 게임엔진이 가장 잘 구현이 되어있으며 유니티도 이 모두를 지원한다. 또 이러한 강력한 환경과 강화학습을 위한 ML-Agents 는 유니티를 강화학습 환경으로 사용하는 프레임워크로 유니티와 연동이 매우 잘된다.

본 논문에서는 이러한 ML-Agents 를 사용하여 유니티에서 모델을 디자인하고 학습한 후 실제 로봇 구동이 가능함을 보였다. 더 나아가서 ROS 와 연동하면 좀더 큰 규모의 로봇의 학습 및 구현이 가능할 것으로 기대 된다.

참고문헌

- [1] Juliani, A., Berges, V., Vckay, E., Gao, Y., Henry, H., Mattar, M., Lange, D., "Unity: A General Platform for Intelligent Agents," *arXiv:1809.02627*, 2018.
- [2] <https://unity.com/>
- [3] Sutton, Richard S. amd Barto, Andrew G., "Reinforcement Learning: An Introduction," *The MIT Press*, 2018.
- [4] M. D. Muhaimin Rahman and S, M, Hasanur Rashid and M. M. Hossain, "Implementation of Q learning and deep Q network for controlling a self-balancing robot model," *Robotics and biomimetics*, 2018.
- [5] Sun Liang and Feimei Gan, "Balance control of two-wheeled robot based on reinforcement learning," *International Conference on Electronic & Mechanical Engineering and Information Technology*, 2011.
- [6] Ching Lung Chang and Shin Yu Chang, "Using Reinforcement Learning to Achieve Two Wheeled Self Balancing Control," *International Computer Symposium*, 2016.
- [7] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, "Proximal Policy Optimization Algorithms," *arXiv:1707.06347*, 2017.