MATLAB에서 회전형 도립 진자 제어를 위한 DDPG 기반 멀티에이전트 강화 학습

지창훈[1], 김주봉[1], 최호빈[1], 임현교[2], 한연희[1+]
한국기술교육대학교 미래융합공학전공[1], 한국기술교육대학교 창의융합공학협동과정[2]
{koir5660, rlawnqhd, chb3350, glenn89, yhhan}@koreatech.ac.kr

Multi-Agent Reinforcement Learning for Rotary Inverted Pendulum in MATLAB

Chang-Hun Ji[1], Ju-Bong Kim[1], Ho-Bin Choi[1], Hyun-Kyo Lim[2], Youn-Hee Han[1]
Future Convergence Engineering, Korea University of Technology and Education[1]
Department of Interdisciplinary Program in Creative Engineering, Korea University of Technology and Education[2]

요 약

강화 학습은 최적의 행동을 찾을 때까지 반복 학습을 수행한다. 이런 강화 학습의 특징은 많은 장점에도 불구하고 현실 세계에서 강화 학습 적용을 어렵게 만든다. 본 논문은 제어분야에서 제어 시스템을 설명하기 위해 많이 사용되는 Rotary Inverted Pendulum을 3D 모델링 하여 가상 환경을 구축하고 구축된 가상 환경의 시뮬레이션을 통해 강화 학습의 반복적인 학습으로 인한 실제 환경의 제약을 해결할 수 있다. 따라서, 본 논문에서는 구축된 가상 Rotary Inverted Pendulum 시뮬레이션을 이용하여 Multi-Agent 강화 학습을 수행함으로써 이를 검증한다.

Ⅰ. 서론

강화 학습은 주어진 환경에서 에이전트가 임의의 행동을 선택하여 문제를 해결하는 기계학습의 한 종류이다. 강화 학습은 게임, 로봇 제어, 네트워크 통신과 같은 복잡한 환경에 적용할 수 있고 관련 분야의 전문 지식이 없어도 최적의 행동을 찾아낼 수 있다는 장점이 있어 주목받고 있다 [1]. 그러나 강화 학습은 최적의 행동을 찾기 위한 반복 학습으로 인해 환경의 제약이 있다. 환경의 제약은 현실 세계에서의 강화 학습 적용을 어렵게 한다. 반복 학습으로 인한 강화 학습 환경의 제약을 해결하기 위해 본 논문에서는 Rotary Inverted Pendulum(RIP) 가상 환경에서 Multi-Agent 강화 학습을 수행하여 환경의 제약을 극복한다.

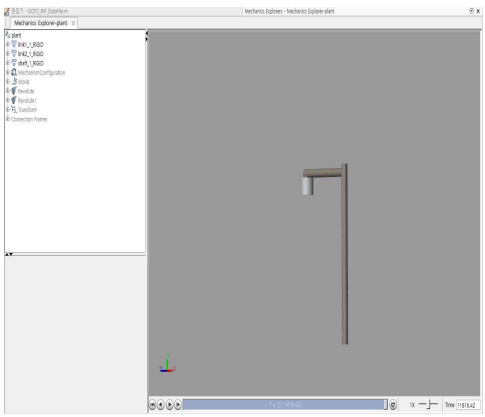
Ⅱ. 본론

1. Rotary Inverted Pendulum 환경

RIP는 비선형적이고 불안정한 동적 시스템으로 실험이 간단하여 제어 시스템 분야에서 검증을 위한 환경으로 사용되어왔다. RIP는 Pendulum, Motor와 Pendulum을 연결하는 Arm으로 구성되어 있으며, Pendulum을 도립 시키고 유지하는 환경이다. 제어 시스템 분야에서는 RIP를 제어하기 위해 복합적으로 여러 controller를 동시에 사용한다 [2].

RIP 실제 환경에 강화 학습을 적용하여 훈련 시키기에는 부품의 소모, 에너지 소비와 오랜 시간의 학습이 필요로 하는 제약 사항들이 존재한다. 따라서, 실제 환경의 제약 사항들을 해결하기 위해 시뮬레이션 환경에 강화 학습을 적용하여 훈련을 진행한다.

본 논문은 대표적인 3D 모델링 프로그램 중 하나인 Solid Works로 RIP를 3D 모델링하고 공학용 프로그램인 MATLAB의 시뮬레이션을 이용해 강화 학습을 위한 환경을 구성한다. MATLAB 시뮬레이션으로 구성한 RIP를 통해 Pendulum 각도 θ_P, Pendulum 속도 w_P, Arm 각도 θ_A와 Arm 속도 w_A를 얻는다. MATLAB 시뮬레이션은 초기 상태일 때 θ_P, θ_A = 0이다 (<표 2> 참조). (그림 1)은 MATLAB 시뮬레이션의 초기 상태를 보여준다.



(그림 1) Rotary Inverted Pendulum MATLAB 시뮬레이션 초기 상태

2. 강화 학습 에이전트

강화 학습 에이전트는 환경으로부터 상태(State)와 보상(Reward)을 받아 행동(Action)을 수행하고 다시 환경으로부터 상태와 보상을 받는 상호작용을 한다. 이러한 상호작용을 바탕으로 보상을 최대화하는 것이 에이전트의 목표이다.

RIP를 비롯한 로봇 제어는 연속적인 제어 값을 추출해야 한다. 따라서 본 논문은 심층 네트워크를 이용해 지정한 행동 범위 내에서 연속적인 행동을 뽑아내는 Deep Deterministic Policy Gradient (DDPG) 알고리즘을 적용한 에이전트를 사용한다 [3].

본 논문에서는 RIP를 역할에 따라 swing-up, balancing 2단계로 나눈다. swing-up은 Pendulum을 도립 시키는 역할이고, balancing은 도립 시킨 Pendulum을 유지 시키는 역할이다. swing-up과 balancing의 행동 범위는 상이하기 때문에 행동 범위가 다른 2개의 에이전트를 이용한다. swing-up과 balancing을 나누는 기준과 행동 범위는 <표 1>과 같다.

+ : 교신저자 한연희(한국기술교육대학교)

<표 1> 강화학습의 에이전트 기준, 행동 범위, 해당 status

에이전트 종류	swing-up	balance
기준	$\theta_P < 177^\circ$, $\theta_P > 183^\circ$	$177^\circ < \theta_P < 183^\circ$
행동 범위	-0.035 ~ 0.035(torque)	-0.01 ~ 0.01(torque)
해당 status	swing-up, swing-up to balance	balance, balancing to swing-up

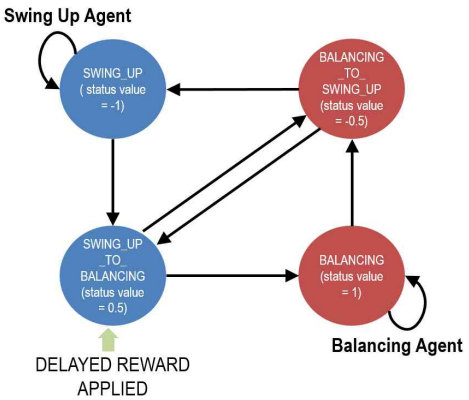
3. 학습 환경 및 구조

본 논문에서는 MATLAB의 python API를 이용해 MATLAB 시뮬레이션을 수행할 때 python으로 구성된 강화 학습 환경에 정보를 실시간으로 전달한다. 우리는 MATLAB 시뮬레이션이 주는 정보를 에이전트가 이해할 수 있도록 재구성하여 상태와 보상을 정의한다.

<표 2> 전달받는 정보, 에이전트의 상태, 보상

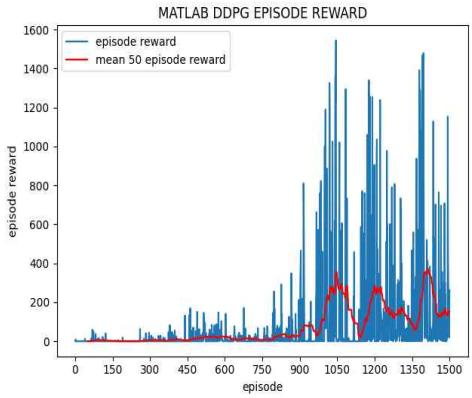
환경이 시뮬레이션에서 전달받는 정보					
Pendulum 각도 (radian)		Pendulum 속도 (radian/s)	Arm 각도 (radian)		Arm 속도 (radian/s)
$0 < \theta_P < 2\pi$		w_P	$0 < \theta_A < 2\pi$		w_A
상태					
$\cos(\theta_P)$	$\sin(\theta_P)$	w_P	$\cos(\theta_A)$	$\sin(\theta_A)$	current status value = (e.g. -1, -0.5, 0.5, 1)
보상					
$\{(2*\pi-\theta')-0.001 * w_P^2 - 50 * \lvert action \rvert * \theta_P\}/1000$ $\theta' = \begin{cases} 2\pi - \theta_P & \text{if } \theta_P > \pi \\ \theta_P & else \end{cases}$					

<표 2>는 MATLAB 시뮬레이션에서 받은 정보로 상태와 보상을 재구성하는 것을 보여준다. 상태는 ($\cos(\theta_P)$, $\sin(\theta_P)$, w_P, $\cos(\theta_A)$, $\sin(\theta_A)$, current status value)으로 정의된다. current status value는 4개로 구성된 status 중 현재 status를 나타내는 value 값을 말한다 ((그림 2) 참조). current status value를 상태에 넣음으로 에이전트는 현재 어떤 status인지 알 수 있다. 보상 식의 θ'는 완전히 도립한 Pendulum의 각도인 π를 가장 높은 값으로 갖도록 θ_P를 변환시킨 것이다. 에이전트는 θ_P가 π에 가깝고 w_P^2와 |action|이 작을수록 큰 보상을 받는다.



(그림 2) status 구성 및 status value

(그림 2)와 같이 RIP 강화 학습 환경은 4개의 status로 나뉜다. swing-up to balancing status와 balancing to swing-up status는 에이전트가 바뀌는 step에만 할당된다. swing-up status와 balancing status는 에이전트가 유지되면 할당된다. 각 status의 에이전트는 <표 1>에서 알 수 있다. 강화 학습 환경이 swing-up to balancing status를 할당 받게 되면 에이전트에게 보상을 넘겨주지 않고 보류시킨다. 그 후 강화 학습 환경 에이전트가 바뀌어 balancing to swing_up status를 할당받게 되면 그동안 balancing status에서 누적된 보상들로 보류 시켰던 swing_up to balancing status의 보상을 대체한다. 이로 인해 swing-up 에이전트는 balancing이 오래 유지되면 더 높은 보상을 얻기 때문에 도립 시킨 후 유지되도록 쉽게 행동을 뽑는다.



(그림 3) 학습 결과 그래프

(그림 3)의 파랑 선은 매 episode의 보상이고 빨간 선은 50 episode의 평균 보상이다. (그림 3)을 보면 DDPG를 이용한 학습의 episode가 증가함에 따라 보상이 증가하는 것을 볼 수 있다. 따라서, 가상 환경에서 RIP의 제어 문제를 성공적으로 해결하고 있음을 알 수 있다.

Ⅲ. 결론

본 논문에서는 강화 학습 환경의 제약을 극복하기 위해 RIP 가상 환경에서 학습을 수행하였다. RIP 시스템은 서로 상이한 행동 범위를 가진 2단계로 구분되었기 때문에 2개의 강화 학습 에이전트를 이용해 학습을 진행하였고 학습이 성공한 것을 확인하였다.

가상 환경의 학습 성공은 실제 환경의 학습에도 좋은 영향을 미칠 수 있다. 성공한 가상 환경을 가진 강화 학습의 좋은 경험들을 실제 환경에서 강화 학습을 수행할 때 에이전트에게 미리 줄 수 있다면, 에이전트의 학습이 빠르게 수렴하는 것을 기대할 수 있다 [4]. 현재 가상 환경의 강화 학습을 끝내고 실제 환경과 연동하기 위한 환경을 구축 중이고 계속적으로 피드백하고 있다.

ACKNOWLEDGMENT

이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행 기초연구사업임(No. 2018R1A6A1A03025526 및 NRF-2020R1I1A3065610)

참 고 문 헌

[1] Mnih et al. "Human-level control through deep reinforcement learning." Nature 518 , no. 7540 (2015): 529--533.

[2] Potsaid el al. "Optimal mechanical design of a rotary inverted pendulum.." Paper presented at the meeting of the IROS, 2002.

[3] Lillicrap el al. "Continuous control with deep reinforcement learning.." Paper presented at the meeting of the ICLR, 2016.

[4] Parisotto el al. "Actor-Mimic: Deep Multitask and Transfer Reinforcement Learning.." Paper presented at the meeting of the ICLR (Poster), 2016.