분산 A3C 를 활용한 회전식 도립 진자 시스템 설계

권도형*, 임현교*, 김주봉**, 한연희**[1]
*한국기술교육대학교 창의융합공학협동과정
**한국기술교육대학교 컴퓨터공학과
e-mail : {dohk, glenn89, rlawnqhd, yhhan}@koreatech.ac.kr

Design of Rotary Inverted Pendulum System Using Distributed A3C Algorithm

Do-Hyung Kwon*, Hyun-Kyo Lim*, Ju-Bong Kim**, Youn-Hee Han**
Korea University of Technology and Education, Republic of Korea

요 약

제어 분야의 가장 기초적인 시스템인 Rotary Inverted Pendulum 을 제어하기 위하여, 본 논문에서는 강화학습에서 Deep Q-Network 과 함께 대표적인 알고리즘으로 알려진 Asynchronous Advantage Actor-Critic 을 활용하여 다중 디바이스 제어를 설계한다. Deep Q-Network 알고리즘을 활용한 기존 연구와 동일한 방식으로 실 세계의 물리 에이전트와 가상 환경을 맵핑시키며, 스위치를 통하여 로컬 에이전트와 글로벌 네트워크 간 통신을 구성한다. 본 논문에서는 분산 Asynchronous Advantage Actor-Critic 을 이용함으로써 실 세계의 다중 에이전트 제어를 위한 강화 학습의 활용 가능성을 조명한다.

1. 서론

강화 학습 이론은 동물의 행동에 대한 이해로부터 얻은 통찰을 통해, 임의로 만든 가상 환경에 놓인 에이전트가 해당 환경 속에서 최선의 행동을 선택하도록 최적화하는 방법을 제안한다. 에이전트는 주어진 환경에 대한 정보를 수집한 후, 해당 데이터들을 통해 학습함으로써, 에이전트가 새로운 상황에 놓였을 때에도 적절히 반응할 수 있어야 한다 [1]. 최근의 강화 학습은 딥 러닝을 통해 환경으로부터 수집한 고차원 데이터의 특징을 추출할 수 있다 [2]. 강화 학습의 주요한 특징은 세 가지로 요약할 수 있다 [3].

- **행위 의존성**: 에이전트가 행하는 각각의 행동에 따라 보상이 다르다.
- **시간 의존성**: 에이전트는 자신이 한 행동에 대하여 즉각적인 보상을 받는 것이 아니라 지연된 시점에서 보상을 받는다.
- **상태 의존성**: 특정 액션에 대한 보상은 환경이 주는 상태에 의존적이다.

본 논문은 Rotary Inverted Pendulum (RIP)에 Deep Q-Network (DQN) 강화 학습 알고리즘을 적용한 연구[4]의 후속 연구로서, 기존 연구와 마찬가지로 초기 상태가 불안정하며 비선형적인 특성을 갖는 비선형 제어 도메인인 RIP 를 대상으로 하여 Asynchronous Advantage Actor-Critic (A3C)알고리즘을 적용한 시스템을 설계한다. 이를 위하여, 기존 연구와 동일한 제품인 QUANSER™사의 QUBE-Servo2 를 고려한다 [4].

2. 기존 연구

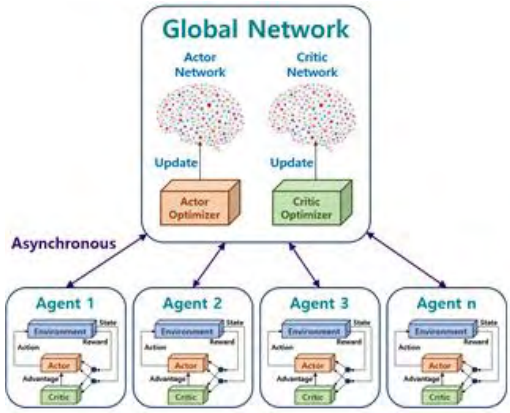
강화 학습 에이전트는 여러 행동 중 어떠한 행동이 지연된 보상을 최대화 하는지를 학습해야 하며, 이것이 에이전트의 최종목표이다.

Q-Learning 은 어떤 상태(s_t)와 행위(a_t)에 대한 최적의 Q 값($Q(s,a)^*$)을 구하기 위하여, 현재의 보상(r)과 다음 상태에 대한 할인된(λ) 미래 보상의 합을 통해, 정책(π)으로 부터 가장 높은 가능성을 갖는 행위(a_t)를 도출하는 벨만 방정식과 비선형적 특성을 갖는 Neural Network 가 결합한, 대표적인 강화 학습 알고리즘이다 [5]. 벨만 방정식은 특정 행위에 대한 장기적인 보상은 현재 행위에서 얻은 즉각적인 보상과 다음 상태에서 취할 행위에 대한 보상의 기대 값의 합과 같음을 나타낸다. 이를 통해 지연된 보상을 최대화하도록 정책을 업데이트 할 수 있다.

Q-Learning 방식보다 진보된 방식인 Deep Q-Learning (DQN)은 off-policy 기반의 알고리즘으로서 상태 값들 간의 상관관계를 줄이기 위하여 학습 메모리 (replay memory)를 사용한다. 반면, A3C 는 on-policy 기

[1] 교신저자: 한연희(한국기술교육대학교)
"이 논문은 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임
(No. 2018R1A6A1A03025526, No. 2016R1D1A3B03933355)"

반의 알고리즘으로서 네트워크가 정책 망과 가치 망으로 분리되어 학습이 이루어진다. 또한 여러 개의 쓰레드가 별개로 학습을 하기 때문에 별도의 학습 메모리가 필요하지 않다. A3C 의 비 동기적인 특성 때문에 복수의 로컬 에이전트의 상호작용을 통하여 글로벌 에이전트가 안정적으로 네트워크를 업데이트 할 수 있게 된다 [4].



(그림 1) Asynchronous Advantage Actor-Critic (A3C) 개념도

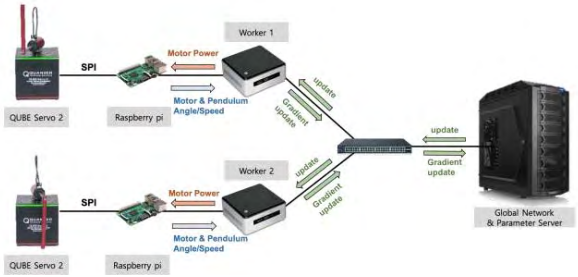
2016 년 2 월 구글의 딥마인드는 DQN 의 단점을 개선한 Asynchronous Advantage Actor-Critic (A3C)을 발표했다 [7]. DQN 에서는 샘플 사이의 강한 상관관계를 없애기 위해 리플레이 메모리를 사용하는 것이 핵심이었던 반면 [1], A3C 는 샘플 사이의 강한 상관관계를 비동기 업데이트로 해결하여 리플레이 메모리를 사용하지 않아도 각 쓰레드가 별개의 환경에서 학습하는 샘플들을 통해 학습하기 때문에 샘플 간의 상관관계가 낮아지게 된다.

A3C 는 Advantage Actor-Critic (A2C) 알고리즘을 기반으로 한다. Actor 는 에이전트가 어떠한 행동을 할지 결정하게 해주며, Critic 은 Actor 가 한 행동을 평가하고, 그 결과를 Advantage 값으로 변환하여 Actor 에게 전달한다. A2C 는 하나의 에이전트가 하나의 환경과 상호작용하는 반면, A3C 는 여러 개의 에이전트가 각각 자신의 환경과 상호작용한다 (그림 1). A3C 는 멀티스레딩을 사용하여 여러 개의 A2C 알고리즘이 동시에 진행된다고 할 수 있다. 각각의 A2C 알고리즘은 자신의 로컬 신경망을 업데이트 한 후, Global Network 를 업데이트 시키기 위해 Global Network 의 옵티마이저로 학습에 필요한 값들을 전달한다. Global Network 의 업데이트가 완료되면 해당 로컬 신경망의 가중치들을 Global Network 의 가중치들로 업데이트 시킨다. 이 과정이 비동기로 진행되기 때문에 샘플 사이의 강한 상관관계를 해결할 수 있다.

기존 연구 [6]에서는 단 한 대의 물리적 환경에 DQN 을 적용함으로서 성공적으로 RIP 를 세우는 목적을 달성하였다. 본 논문에서는 기존 연구에서 더 나아가 A3C 알고리즘을 활용함으로써 분산 환경에서 복수의 에이전트를 통해 학습하는 분산형 다중 RIP 시스템을 제안한다.

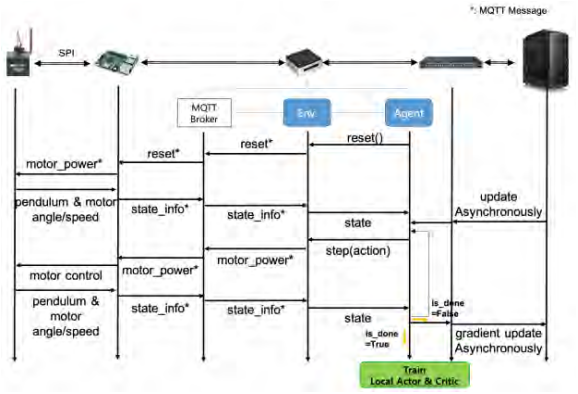
3. 시스템 설계 및 구현

기존 연구 [4]에서는 QUANSER 사의 QUBE-Servo2 를 제어하기 위하여 SPI 패널을 선택하였고, 라즈베리파이를 통해 해당 제품에서 나오는 센서 값을 수신 및 적절한 값으로 정규화하여 DQN 네트워크가 위치한 베어본 컴퓨터로 전달하였다. 본 논문에서는 이와 유사하게 SPI 통신 패널과 라즈베리파이를 이용하지만, 더 나아가 두 대의 QUBE-Servo2 를 제어하기 위하여 각각의 제품에 라즈베리파이와 베어본을 하나씩 할당하며, ipTime T24000 스위치와 함께 글로벌 신경망을 가진 별도의 베어본 한 대를 추가한 시스템을 제안한다. 전체 시스템 구성도는 (그림 2)와 같다.



(그림 2) 분산 A3C 전체 시스템 구성도

실제 시스템에 해당하는 두 대의 QUBE-servo2 가 라즈베리파이를 통해 각각의 Worker 와 통신하는데, 이 때 Worker 에는 가상의 에이전트가 존재하여 실제 시스템에서 발생한 상태정보를 받아들인 후, 학습하여 로컬 네트워크를 업데이트한다. 로컬에서 업데이트 된 내용들은 비동기적으로 스위치를 거쳐 글로벌 네트워크 컴퓨터로 전달되어 글로벌 네트워크가 업데이트 되고, 글로벌 네트워크는 각각의 Worker 로부터 받은 네트워크를 통해 업데이트를 수행한 후, 업데이트된 네트워크를 다시 스위치를 통해 비동기적으로 각각의 Worker 에게 넘긴다. Worker 는 업데이트된 네트워크를 바탕으로 QUBE-Servo2 에게 행동 (action)을 넘기며, 이 과정이 반복된다. 스위치에 연결된 세 대의 컴퓨터 각각에는 별도의 ip 가 할당되어 MQTT 를 통해 통신이 이루어진다. 이를 더 자세히 표현하면 (그림 3)과 같다.



(그림 3) 시스템 네트워킹 시퀀스 구성도

(그림 3)은 QUBE-Servo2 와 라즈베리파이 사이, 라즈베리파이와 베어본 컴퓨터 그리고 베어본 컴퓨터가 스위치와 물린 글로벌 네트워크 컴퓨터와 연결된 상황을 표현하고 있으며, 각각의 컴포넌트 간 네트워킹 시퀀스를 나타내고 있다. 그림 상에는 한 대로 표현되었지만, (그림 2)에서와 같이 두 대의 QUBE-Servo2 와 두 대의 라즈베리파이, 두 대의 베어본 컴퓨터와 한 대의 글로벌 네트워크 컴퓨터로 구성되어 있다. 베어본 컴퓨터는 환경과 에이전트가 구조적으로 분리되어 있다. 이 때, 환경과 라즈베리파이는 IoT 장비를 위해 경량화 된 프로토콜인 MQTT 를 이용하여 통신한다. MQTT-Broker 는 환경과 라즈베리파이에 특별히 지정한 Topic 을 통해 발행(publish)과 구독(subscribe)을 중계한다.

QUBE-Servo2 가 라즈베리파이를 거쳐 에이전트에게 상태를 넘길 때, 에이전트가 이해할 수 있는 형태로 넘겨야 한다. Pendulum 과 Motor 의 angle 과 speed 는 <표 1>에 나타난 바와 같이 각각 θ_{k1}, θ'_{k1}, α_{k1} 그리고 a'_{k1}로 표현된다. 또한 에이전트는 QUBE-Servo2 가 이해할 수 있는 값을 넘겨 행동을 하도록 해야 한다. 이를 위해 에이전트 측에서는 세 가지 행위를 리스트로 유지하며, -60V, 0, 60V 로 정의하고, 이 중 하나의 voltage 에 해당하는 인덱스를 QUBE-Servo2 에게 넘긴다. 인덱스를 넘겨받은 QUBE-Servo2 는 동일한 리스트를 유지하여, 넘겨받은 인덱스에 맵핑된 voltage 로 전력을 가하여 모터를 구동시킨다. 행동과 상태에 대한 자세한 내용은 <표 1>로 보였다.

<표 1> Action 과 State 정보

Action		
0	1	2
-60	0	60
State		
Pendulum Angle(radian)	Pendulum Speed(radian/s)	Motor Angle(radian)	Motor Speed(radian/s)
θ_{k1}	θ'_{k1}	α_{k1}	a'_{k1}

4. 결론

기존 연구 [4][6]에 이어 본 논문에서는 강화 학습의 A3C 를 이용하여 다중 RIP 를 위한 시스템 설계를 제안하였다. 기존 연구의 한 대의 RIP 를 세우는 것과는 다르게, 두 대 이상의 RIP 를 세우는 것은 기기마다 서로 다른 상태정보를 낼 수 있다는 점에서 생각보다 어려운 문제이다. 즉, 강화 학습이 현실세계의 단일한 물리적인 시스템을 제어할 수 있는 것 뿐만 아니라 복수의 물리적인 시스템을 제어할 수 있다면, 강화 학습이 더욱 넓은 활용도를 가질 수 있다는 것을 입증할 수 있다. 추후 시스템 구성을 끝낸 후, 실험을 통해 이를 입증할 예정이다.

참고문헌

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg and D. Hassabi, “Human-level control through dee p reinforcementlearning,” in Nature, 518(7540):529–533, 02, 2015.

[2] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra and M. Riedmiller, “Playing atari with deep reinforce mentlearning,” In NIPS Deep Learning Workshop, 2013.

[3] A. Juliana, "Simple Reinforcement Learning in Tensorflow: Part 1 - Two-armed Bandit," [Online], Available: https://medium.com/@awjuliani/super-simple-reinforcement-learning-tutorial-part-1-fd544fab149.

[4] J. B. Kim, D. H. Kwon, Y. G. Hong and M. S. Kim, "Design of Rotary Inverted Pendulum System Using Reinforcement Learning", Korea Information Processing Society(KICS), Vol. 25, No. 2, 11, 2018.

[5] C. Watkins and P. Dayan, "Q-learning," pp. 279-292, 1992.

[6] J. B. Kim, H, K, Lim, C. M. Kim, M. S. Kim, Y. G. Hong and Y. H. Han, "Imitation Reinforcement Learning-Based Remote Rotary Inverted Pendulum Control in OpenFlow Network," IEEE Access, Vol. 7, pp. 36682-36690, Mar, 2019.

[7] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver and K. Kavukcuoglu, “Asynchronous Methods for Deep Reinforcement Learning,” arXiv:1602.01783v2, 2016.