JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 21, No. 2, pp. 381-386, Feb. 2020

Check for updates

강화학습 기반 자율주차 연구를 위한 시뮬레이터 개발

엄 하 영[1] · 김 정 환[2] · 지 승 윤[2] · 최 희 열[3*]
[1]한동대학교 정보통신공학과 석사과정
[2]한동대학교 전산전자공학부 학부과정
[3]한동대학교 정보통신공학과 조교수

Autonomous Parking Simulator for Reinforcement Learning

Hayoung Eom[1] · Jeonghwan Kim[2] · Seungyun Ji[2] · Heeyoul Choi[3*]
[1]Master's Course, Department of Information Communication Engineering, Handong Global University
[2]Undergraduate Student, School of Computer Science and Electrical Engineering, Handong Global University
[3]Assistant Professor, Department of Information Communication Engineering, Handong Global University

[요 약]

딥러닝 알고리즘의 발전에 따라 강화학습은 게임이나 물리 기반 모델과 같이 연속적인 동작을 필요로 하는 많은 분야에서 큰 활약을 하고 있다. 이에 따라 OpenAI의 Gym모듈과 같이 여러 강화학습 알고리즘을 실험하고 비교하기 위한 여러가지 방법들이 고안 되었고 강화학습을 시각화하고 분석하는데 많은 기여를 하였다. 현재 센서에 기반한 규칙 기반 시스템을 따르고 있는 많은 자율주차 알고리즘과 차량의 자율성에 대한 수요가 증가하는 것을 고려했을 때, 자율주차에 강화학습을 적용하는 연구의 필요성이 대두되고 있고, 이를 위한 시뮬레이터가 필요하다. 본 논문은 강화학습 연구가 가능한 자율주차 시뮬레이터를 개발한다. 실제 차량의 동작 방식을 모델링하기 위하여 애커만 조향 기하학 모델 수식에 기반한 자율주차 시뮬레이터를 개발한 뒤, Deep Deterministic Policy Gradient (DDPG) 강화학습 알고리즘을 적용하여 개발한 자율주차 시뮬레이터가 성공적으로 학습하는 모습을 보여주었다.

[Abstract]

With the advances in deep learning algorithms, reinforcement learning has shown considerable accomplishments in such tasks as game and physics-based models that require continuous actions. Many platforms and methods like OpenAI Gym were devised to evaluate and compare multiple reinforcement learning algorithms and thus made significant contributions to the deep learning community. In addition to such developments, considering the increasing demand for autonomous vehicles and rule-based parking assistance systems based on attached sensors, we need a parking simulator where reinforcement learning can be applied. In this paper, we develop a new autonomous car parking simulator which allows the learning agent to be trained with reinforcement learning algorithms. The results show the simulator being successfully trained with Deep Deterministic Policy Gradient (DDPG) algorithm.

색인어 : 자율주차, 시뮬레이터, 강화학습, 정책경사 알고리즘, 깊은 결정적 정책경사 알고리즘
Key word : Autonomous-Parking, Simulator, Reinforcement Learning, Policy Gradient, Deep Deterministic Policy Gradient

http://dx.doi.org/10.9728/dcs.2020.21.2.381

Received 20 December 2019; **Revised** 16 January 2020
Accepted 25 February 2020

***Corresponding Author; Heeyoul Choi**

Tel: +82-54-260-1303
E-mail: heeyoul@gmail.com

Ⅰ. 서 론

심층망을 기반으로하는 딥러닝 기술의 급속한 발전으로 이미지, 음성, 텍스트 등 여러 분야에서 다양한 방식으로 성공적인 결과를 얻고 있다[1]-[3]. 강화학습은 2016년도 알파고[4]와 이세돌 9단의 공개 대국 이후 학습 객체를 시행착오 방식을 통해 학습하는 강화학습 고유의 방식과 딥러닝 기반의 심층 신경망이 가지는 높은 정확도의 함수 근사 능력을 기반으로 많은 분야에 적용이 되기 시작하였다[5]. 게임이나 물리 기반 시뮬레이터 학습 같이 연속적인 동작을 요구하는 여러 분야에 강화학습 알고리즘이 적용되기 시작했고[6], 이 중 많은 분야에서 규칙 기반 시스템(Rule-based System)과 사람을 능가하는 성능을 보이고 있다[7]-[8].

강화학습이 다른 방법론들에 비해서 가지는 강점은 일반화(Generalizability)가 높다는 점이다. 이는 규칙 기반 시스템처럼 사람이 최대한 많은 상황들을 고려하여 구현한 시스템보다 유연하며 이식성이 높다. 특히, 자율주차 같이 여러 예외 상황이 존재하는 경우에는 규칙기반 시스템이 가지는 한계점이 분명하다. 이에 강화학습은 이러한 연속적인 동작을 요구하고 예외상황이 많은 경우에 학습 객체가 실행착오(Trial-and-error) 방식으로 여러 상황에 대응할 수 있도록 학습을 가능하게 한다.

자율주차는 센서나 카메라를 통해 입력되는 외부의 값을 기반으로 자율적으로 주행하는 기능을 지니는 차량이 주차장 환경 내에서 주차를 하는 작업이다. 자율주차는 2003년 세계 최초로 상용 자동차에 선보여진 후, 여러 연구와 발전을 거듭하며 많은 상업용 차량에 보급이 되었다. 자율주차에 대한 연구에 대해서는 경로 생성, 상황별 자율주차 등 선행 연구들이 존재했지만[9]-[10] 위에 언급한 바와 같이 분명한 한계점이 존재한다. 예를 들어, 직각주차를 해야 하는 환경에서 차량이 자율적으로 평행주차를 하는 상황이 발생할 수 있고, 주차공간의 다양한 상황 (예를 들어, 주차선이 있거나 없는 경우, 주차 공간이 좁거나 넓은 경우 등) 이나 돌발적으로 일어날 수 있는 예외 상황들을 고려할 수 없다는 단점이 있다. 이러한 이유로 자율주차에 강화학습을 적용하여 학습할 수 있도록 기반 플랫폼을 만드는 것이 중요하다.

이 논문은 자율주차 작업(Task)을 강화학습으로 학습할 수 있는 자율주차 환경 및 차량 시뮬레이터를 만드는 것에 대하여 논의한다. 먼저 이 시뮬레이터는 실제 주차 환경에서 적용 되는 물리식인 애커만 조향 기하학 모델(Ackerman Steering Geometry Model)을 사용하여, 실제 차량의 주차와 거의 동일한 환경을 제공한다. 또한 환경에 보상 값(Reward)을 자유롭게 설정하여 실험할 수 있도록 구현 되어있다. 이 자율주차 시뮬레이터는 강화학습에서 연속적인 액션과 이산적인 액션에 대해서 모두 처리 가능하게 구현 되어있다. 즉, 강화학습의 대표적인 알고리즘인 Q-learning[11] 및 Deep Deterministic Policy Gradient(DDPG)[12] 에 사용이 가능하며, 실험에서 학습이 잘 되는 것을 확인하였다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 강화학습을 위한 시뮬레이터 환경에 대해서 살펴보고 3장에서 애커만 조향 기하학 모델과 강화학습 알고리즘에 관해서 간단하게 리뷰 한다. 4장에서 본 논문이 소개하는 시뮬레이터와 세부 사항에 대해서 소개하고 5장에서는 구체적인 강화학습 알고리즘을 기술한다. 6장에서는 실험 내용을 보여주고 7장에서 결론을 맺는다.

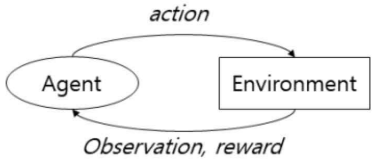

그림 1. OpenAI Gym의 일반적인 구조
Fig. 1. General framework of OpenAI Gym

Ⅱ. 기존 연구

자율주차에 강화학습을 적용하여 학습한 선행 연구 사례[10]에서는 자율주차 작업을 세 가지 단계로 나누어 학습을 진행한다. 주차 경로를 하나의 경로(Trajectory)로 학습하지 않고 세 가지의 경로로 나눈 후, 첫 번째와 마지막 단계에서 간단한 직진으로 도달점에 도착하는 단계만 강화학습을 적용하여 학습한다. 또한, “High fidelity simulator”[10] 에서 강화학습 알고리즘을 기반으로 학습을 진행하였다는 것을 언급하지만, 학습에 사용한 시뮬레이터에 대한 상세 설명이나 새로운 시뮬레이터 개발에 대한 언급이 없다. 학습에 적용한 알고리즘만큼 중요한 것은 학습에 사용한 플랫폼과 기반 시뮬레이터다. 이는 다른 연구자들이나 관련 분야에 관심을 가지고 있는 이들이 여러 학습 알고리즘을 적용할 수 있도록 기여하는 지식과 결과 공유의 기반이 된다. 이는 ImageNet[13] 데이터 셋이나 MNIST[14] 데이터 셋과 같이 여러 딥러닝 알고리즘을 학습하는 것을 가능하게 하는 중요한 자원이 된다.

OpenAI Gym[15] 라이브러리는 OpenAI에서 공개한 강화학습을 위한 환경을 손쉽게 구축할 수 있게 해주는 API(Application Programming Interface)이다. 굉장히 간단하고 쉽게 구현이 되어 있으며, 그림 1 과 같이 학습 객체(Agent)가 동작(Action)을 취하면, 환경(Environment)에 반영이 된 후, 이에 따른 관찰 값(Observation)과 보상 값(Reward)이 학습 객체에게 넘어 온다.

이에 대해서 본 논문은 Gym과 같이 자율주차에 강화학습 알고리즘을 적용하여 학습시킬 수 있는 시뮬레이터를 Python에서 Matplotlib의 Pyplot모듈로 개발한 방법에 대해서 논의한다.

Ⅲ. 배 경

3-1 애커만 조향 기하학 모델

애커만 조향 기하학 모델은 차량 조향의 기구학적 메커니즘을

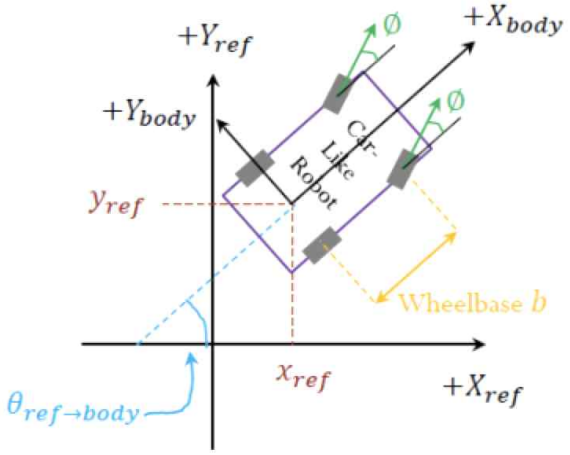

그림 2. 유클리디안 공간상의 애커만 조향 기하학 모델
Fig. 2. Ackerman Model in Euclidean Space

정의하기 위해 사용하며, 본 논문에서 시뮬레이터를 구현할 때 실제 차량의 움직임을 모델링하고자 적용 및 구현한 모델이다. 애커만 조향 기하학 모델은 2차원 유클리드 공간 상에서 x축의 값 x_{ref}과 y축의 값 y_{ref}이 차량의 후륜 정중앙의 좌표 값이고, 차의 각도를 $\theta_{ref\to body}$, 차의 각도로부터 바퀴의 각도를 $\theta_{steering}$ 이라고 하고 이를 차의 위치 p라고 하자. 속력 v와 바퀴 각도 $\theta_{steering}$을 갖는 동작 $a=[v\ \theta_{steering}]$ 에 대해서 , 시간 t의 차량 위치와 바퀴의 각도가 주어졌을 때 t+1에서 움직이는 차량의 위치$\dot{x}_{ref}$와 $\dot{y}_{ref}$는 다음과 같이 계산할 수 있다.

$$\dot{x}_{ref} = v^{*}\cos\theta_{ref\to body} \tag{1}$$
$$\dot{y}_{ref} = v^{*}\sin\theta_{ref\to body} \tag{2}$$

이에 대한 2차원 유클리드 공간상에서 차량 객체가 가지는 움직임의 모양을 도식화하면 그림 2 와 같다.

본 시뮬레이터는 언급한 물리학적 모델을 시뮬레이터에 구현하였다. 따라서 타이어 바퀴가 미세하게 헛돌거나 강한 바람이 부는 등과 같이 자연계의 노이즈에 해당하는 요소들을 고려하지 않는다면, 시뮬레이터의 환경은 실제 물리 세계의 주차 환경과 같다.

3-2 강화학습

강화학습은 기계학습(Machine Learning)의 한 분야로써, 행동심리학에서 영감을 받아 발전하였다. 강화학습은 특정 환경(Environment)안에서 동작(Action)을 취할 수 있는 객체가 있다는 전제로 시작한다. 어떤 환경 내에서 특정 학습 객체(Agent)가 현재 환경의 상태(State)를 인식하고, 선택 가능한 행동(Action)들 중 보상(Reward)을 최대화 하는 방향으로 학습 한다. 이는 사람이 실행착오(Trial-and-error) 방식으로 지식과 경험을 습득하는 방식과 유사하다[5].

그리고 강화학습은 크게 두 가지 방법론을 사용한다 : 1) 가치 함수 기반 학습(Value Function Approach), 2) 정책 함수 기반 학습(Policy Gradient Approach). 가치 함수로는 이 두 가지 방법들이 가장 많이 사용되는 방식이고, 마르코프 결정 과정에 대한 수식은 다음과 같다[5]. $P(S_{t+1}|S_t)$는 t 시점의 상태 S_t에서 S_{t+1}로 가는 확률이다.

$$P(S_{t+1}=s'|S_0,S_1,...,S_{t-1},S_t)=P(S_{t+1}=s'|S_t) \tag{3}$$

본 논문은 가치 함수 기반 학습 방법인 Tabular Q-learning[11]을 사용한 방식과 정책 함수 기반 학습 방식 중 하나인 DDPG[12] 방식을 적용하여 시뮬레이터에 강화학습 알고리즘을 성공적으로 학습 시킨다.

Ⅳ. 주차 시뮬레이터

이러한 강화학습 알고리즘들을 자율주차에 적용 할 수 있도록 그림 3 과 같은 자율주차 환경을 구현하였다. 본 자율주차 시뮬레이터는 Python언어로 구현되었고, Matplotlib 라이브러리의 Pyplot 모듈을 사용하였다.

시뮬레이터 상의 자율주차 환경은 가로 15m, 세로 10.5m의 비율로 설계되었으며, 직각주차(T-자 주차)를 실험하기 위해 설계되었다.

차량은 실제 차량의 비율을 고려하여 현대의 NF소나타의 비율을 반영하여 길이 4.9m, 폭 1.83m의 비율로 고정하였으며, 축간거리(Wheelbase)는 2.84m의 길이로 설정하였다.

그리고 시뮬레이터 상에서 직각주차 환경을 고려하기 위해 너비 5.7m, 높이 5m인 두 개의 장애물을 각각 주차공간의 좌측 하단과 우측 하단에 위치 시켰다. 실제에서는 다른 자동차들일 수 있다.

시뮬레이터가 주는 보상(Reward)은 벽이나 장애물에 충돌할 경우$R_{collsion}$, 주차에 성공할 경우R_{goal}, 서브 골에 도달 할 경우 $R_{subgoal}$, 주차 시간이 초과하는 경우에 대한 리워드 $R_{timeover}$ 와 골과의 거리에 따른 리워드$R_{distance}$, time-step에 따른 리워드 $R_{timestep}$ 등으로 설정 할 수 있도록 구현하였다. 실험에서 원하는 형태로 설정하여 사용할 수 있다.

또한, 주차 지점은 유클리디안 공간상에 있는 시뮬레이터라는 점과 상태(State) 값과 동작(Action) 값이 정규화 되어 평균이 0이라는 점을 고려하여, 원점 (0, 0) 으로 설정하였다.

상태(State)의 경우 시뮬레이터의 이미지(Raw image)를 프레임 단위로 사용할 경우에 비싼 계산 비용과 느린 학습 속도를 보완하기 위해서 입력 값을 길이 17(17-dimension)인 벡터 값을 사용하도록 하였다. $g1x, g1y, ..., g6x, g6y$는 그림 3에서 찍힌 각 점의 좌표 값이며, 이에 대한 상태 s는 다음과 같다.

$$s = [x, y, \theta_{heading}, \theta_{steer}, g1x, g1y, ..., g6x, g6y] \tag{4}$$

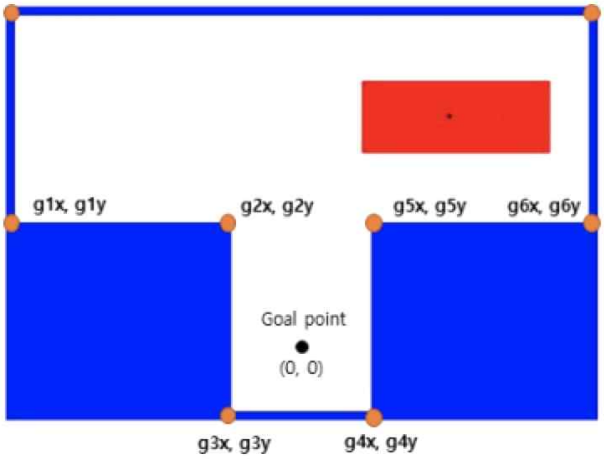

그림 3. 자율주차 시뮬레이터 화면 (직각주차)
Fig. 3. Autonomous Parking Simulator (Perpendicular Parking)

행동(Action)의 경우 이동할 거리와 바퀴를 돌릴 각도 값을 각각 실수로 갖는 벡터를 입력으로 주도록 구성하였다. 이 때 바퀴의 각도는 45도 이상으로 커지거나 –45도 이하로 내려가지 않도록 하였으며, 이동 거리는 시뮬레이터 가로 길이를 넘지 못하도록 하였다.

Ⅴ. 어플리케이션

위에서 소개된 주차 시뮬레이터에는 앞서 강화학습 분야에서 가장 많이 사용되는 두 가지 방법인 Tabular Q-learning[11]과 DDPG[12]를 적용 할 수 있다.

Q-learning은 가치 함수 기반 학습 방법임과 동시에 모델 프리 강화학습(Model-free Learning) 기법 중 하나인데[11], 모델 프리 강화학습은 환경에 대한 정보, 즉 환경이 어떤 방식으로 동작을 하는지 모른다는 특징을 가진다. 이러한 환경에서 학습 객체는 탐사(Exploration)라는 시행착오를 통해 직접 동작을 취한 후에 환경에서 받는 보상 값 r과 상태 s를 통해서 가치 함수와 정책 함수를 학습 시켜야한다. 반복적인 학습을 통해 각 상태에서 최적의 동작에 대한 가치 평가 측정치인 Q를 구한다. Q-learning과 같은 가치 함수 기반 학습 방식은 벨만 방정식(Bellman Equation)을 통해서 가치 반복(Value Iteration) 문제를 해결하고, 이러한 가치를 최대로 만드는 정책(Policy)를 찾는다. Tabular Q-learning은 Q라는 행동 가치를 테이블 형태로 저장하여 업데이트 하는, 기본적인 강화학습 방식 중 하나이다. 테이블 내에 저장된 각각의 Q 값을 업데이트 하는 것은 다음과 같이 벨만 업데이트 방정식 (Bellman Update Equation)을 사용한다. α는 Learning Rate, γ는 Discount Factor에 해당한다.

$$Q(s,a) \leftarrow Q(s,a) + \alpha(r + \gamma \max_{a'} Q(s',a') - Q(s,a)) \quad (5)$$

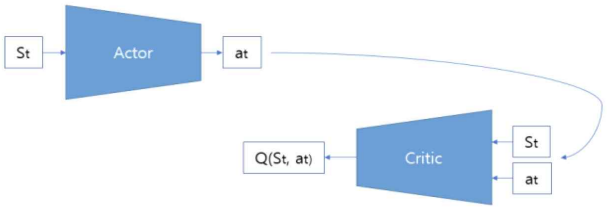

그림 4. DDPG 모델
Fig. 4. DDPG model

하지만, Tabular Q-learning을 적용할 경우 이산적인 state 와 action 의 수가 너무 많아지는 문제가 있고, 실제 성능도 학습중에 61% 정도의 성공률을 넘지 못했다. DDPG는 모델 프리 강화학습 중 한 가지 방법이지만, 가치 함수 기반인 Tabular Q-learning 모델과는 차이점이 존재한다.

DDPG는 policy gradient 방식으로 학습을 진행하는데, 가치 함수만 근사하는 첫 번째 방식과는 다르게 학습 객체가 취할 동작을 바로 예측하는 것을 학습한다. 이를 위해 DDPG에서는 기존의 행동-평가(Action-Critic)네트워크 구조를 채택하고, 리플레이 버퍼(Replay buffer)와 소프트 타겟 갱신(Soft target update) 방식을 사용하여서 취한 액션과 상태 쌍의 상관관계를 감소시키고 네트워크가 학습 중에 발산하는 것을 방지한다[16]. 행동 네트워크(Action network)는 다음 시간대에 환경 내에서 동작을 예측하고, 평가 네트워크(Critic network)는 취한 동작에 따른 가치를 계산하여 자기 자신과 행동 네트워크를 업데이트 한다. 그림 4는 DDPG 구조의 기본적인 도식이다. St 는 현재 환경의 상태를 나타내고, at는 행동 네트워크에서 St를 입력값으로 예측한 학습 객체의 동작을 나타낸다.

Ⅵ. 실 험

애커만 조향 기하학 모델을 기반으로 구현한 시뮬레이터 상에서 DDPG 알고리즘을 적용하기 전에 앞서 환경 상의 상태와 동작 값에 대한 정규화, 보상 함수(Reward Function)설계, 주차 지점 설정, 그리고 입력값의 차원 설정을 우선적으로 실험하였다.

보상함수 R 은 4가지 경우(벽에 부딪힌 경우, 부딪히지 않은 경우, 객체에 부딪힌 경우, 시간 초과된 경우)에 대해 각각 다음과 같이 설정했다. 참고로 다양한 실험을 위해 보상함수는 수정할 수 있도록 구현되어 있다.

$$\begin{aligned} R_{collision} &= -100 \\ R_{goal} &= 1000 \\ R_{hitobstacle} &= -100 \\ R_{timeover} &= -10 \end{aligned} \quad (6)$$

학습 객체(Agent)가 취하는 동작은 행동 네트워크에서 나오는 출력값으로 행해진다. 행동 네트워크에서 예측하는 출력값의 구조는 다음과 같다.

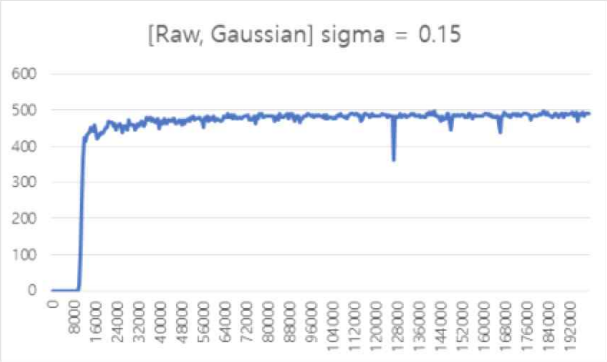

그림 5. 베이스 라인으로 잡은 DDPG에 대한 자율주차 시뮬레이터 성능, 가로축은 학습한 episode, 세로축은 testing 500번에 대한 성공 횟수이다. sigma는 exploration으로 사용한 Gaussian Noise의 sigma 값을 의미한다.

Fig. 5. Performance of the baseline model (DDPG). Horizontal axis is the number of episodes, vertical axis is the number of success in 500 trials. (sigma = Standard deviation for Gaussian noise used in the experiment)

$$y = [distance, \theta_{steer}] \tag{7}$$

출력값에서 distance는 차량이 이동하는 거리를 나타내고, steer 각도 값은 핸들을 트는 각도의 양을 의미한다. 자율주차 시뮬레이터는 이러한 출력값을 이용하여 차량의 행동과 위치를 수정한다.

본 논문에서는 시뮬레이터 실험에서 위의 입력값과 출력값을 기반으로 강화학습 알고리즘을 적용하여 학습이 가능함을 확인했다. 그림 5 에 의하면 DDPG 알고리즘의 경우 500회의 주차 시도마다, 평균 97%의 정확도로 자율주차 시뮬레이터 상에서 주차를 성공시킨다는 결과를 관찰할 수 있다. 강화학습 문제에 대한 베이스라인의 성능은 시뮬레이터 마다 달라질 수 있고, 97% 정확도는 다른 자율주차 문제를 다루는 논문[10]에서 제시하는 성능과 비교했을 때, 베이스라인으로 적당한 수치이다.

Ⅶ. 결 론

결론적으로 본 논문은 효과적인 강화학습을 위한 자율주차 시뮬레이터를 구현한 것에 대하여 논의한다. 애커만 조향 기하학 모델이 제안하는 수식을 기반으로 구현된 본 시뮬레이터는 실제 직각주차 상황과 차량의 물리적 특성을 고려하여 설계 되었다. OpenAI Gym과 같은 강화학습을 훈련시키고 학습 모델을 비교하며 공유하는 플랫폼과 같이, 강화학습 알고리즘이 더욱 더 발전하기 위해서는 여러 물리적 작업들을 학습할 수 있는 효과적인 플랫폼이 필요하다. 본 논문에서 구현한 시뮬레이터는 DDPG 알고리즘을 구현된 시뮬레이터에 적용하여 효과적인 학습이 가능함을 입증하였다. 향후 연구에서는 시뮬레이터 상에서 직각주차 뿐만이 아닌 평행주차, 후진주차와 같은 다른 종류의 주차를 학습 할 수 있도록 동적으로 설정 가능한 시뮬레이터 구현으로 앞으로 강화학습을 이용한 자율주차에서 다른 상황들에 대해서 학습할 수 있는 플랫폼을 기반으로 본 연구 분야의 발전에 추가적인 기여가 있을 것으로 예상된다.

감사의 글

이 논문은 과학기술정보통신부와 정보통신기술 진흥센터의 소프트웨어중심대학 지원사업(2017-0-00130)의 지원을 받아 수행하였음. 또한 2017년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임 (2017R1D1A1B03033341). 그리고 VADAS 의 프로젝트 지원에 대해 감사드립니다.

참고문헌

[1] H. Choi, Y. Min, “Intelligent Information System; Introduction to deep learning and major issues”, *Korea Information Processing Society* Review, vol. 22, no. 1, pp. 7-21, 2015.

[2] C. Kang, Y. Ro, J. Kim, and H. Choi, “Symbolizing Numbers to Improve Neural Machine Translation,” *Journal of Digital Contents Society*, vol. 19, no. 6, pp. 1161-1167, 2018.

[3] C. Jeong, and H. Choi, “Neural Machine Translation with Word Embedding Transferred from Language Model”, *Journal of Digital Contents Society*, vol. 20, no. 11, pp. 2211-2216, 2019.

[4] D. Silver, A. Huang, C. Maddison, A. Guez, L. Sifre, “Mastering the game of Go with deep neural networks and tree search”, in *Nature*, Vol. 529, pp. 484-503, 2016.

[5] Y. Li, “Deep Reinforcement Learning : An Overview”, https://arxiv.org/abs/1701.07274, 2017.

[6] V. Mnih Silver, 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, J. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, “Human-level control through deep reinforcement learning”, in *Nature*, Vol. 518, pp. 529-533, 2015.

[7] V. Minh, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, M. Riedmiller, “Playing Atari with Deep Reinforcement Learning”, https://arxiv.org/abs/1312.5602, 2013.

[8] Z. Xie, P. Clary, J. Dao, P. Morals, J. Hurst, M. V. Panne, “Iterative Reinforcement Learning Based Design of Dynamic Locomotion Skills for Cassie”, in *Robotics*, https://arxiv.org/abs/1903.09537, Mar. 2019

[9] B. Esiyok, A. C. Turkmen, O. Kaplan, C. Clik, “Autonomous Car Parking System with Various Trajectories”, in *Periodicals*

of Engineering and Natural Sciences, Vol. 5, No. 3, pp.364-370, Nov. 2017.

[10] Y. Zhuang, Q. Gu, B. Wang, J. Luo, H. Zhang, W. Liu, "Robust Auto-parking: Reinforcement Learning based Real-time Planning Approach with Domain Template", in *Neural Information Processing (NIPS)*, Nov. 2018.

[11] Watkins, C. JCH, Dayan, Peter, "Q-learning", in Machine learning, 8(3-4), 279-292, 1992.

[12] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous Control With Deep Reinforcement Learning", in *International Conference on Learning Representations (ICLR)*, 2016.

[13] A. Berg, J. Deng, L. Fei-Fei, "Large Scale Visual Recognition Challenge",www.imagenet.org/challenges, 2010.

[14] Y. Lecun, C. Cortes, C. J. C. Burges, "The MNIST Dataset of Handwritten Digits (Images)", http://yann.lecun.com/exdb/mnist/, 1999.

[15] G. Brockman, V. Cheung, L. Petterson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, "OpenAI Gym" https://arxiv.org/abs/1606.01540, Jun. 5, 2016.

[16] T. P. Lillicrap, J.J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, "Continuous Control With Deep Reinforcement Learning", *International Conference on Learning Representations (ICLR)*, 2016

엄하영(Hayoung Eom)

2018년 : 한동대학교 전산전자공학부 (학사)
2018년~현 재 : 한동대학교 대학원 정보통신공학과 석사과정
※관심분야 : 딥러닝(Deep Learning), 강화학습(Reinforcement Learning)

김정환(Jeonghwan Kim)

2020년 : 한동대학교 전산전자공학부 (학사)

※관심분야 : 강화학습(Reinforcement Learning), 자연어처리(Natural Language Processing)

지승윤(SeungYun Ji)

2020년 : 한동대학교 전산전자공학부 (학사)

※관심분야 : 딥러닝(Deep Learning), 강화학습(Reinforcement Learning)

최희열(Heeyoul Choi)

2005년: 포항공과대학교, 컴퓨터공학과 (이학석사)
2010년: Dept. of Computer Science and Engineering, Texas A&M University (Ph.D)
2010년 ~ 2011년: Indiana University (PostDoc)
2015년 ~ 2016년: University of Montreal (Visiting Researcher)
1998년 ~ 2001년: OromInfo (Programmer)
2011년~2016년: 삼성전자 종합기술원 (Research Staff Member)
2016년~현 재 : 한동대학교 전산전자공학부 조교수
※ 관심분야 : 머신러닝, 딥러닝, 인공지능