

다양한 자율주행 이동체에 적용하기 위한 장애물 회피의사 결정 시스템 연구

강병준¹, 김종원^{2*}

¹한국기술교육대학교 전기전자통신공학부, ²한국기술교육대학교 기전융합공학과

Decision Support System of Obstacle Avoidance for Mobile Vehicles

Byung-Jun Kang¹, Jongwon Kim^{2*}

¹School of Electrical, Electronics & Communication Engineering, KOREATECH

²Department of Electromechanical Convergence Engineering, KOREATECH

요 약 본 논문은 자율주행 자동차 및 자율주행이 가능한 이동체를 대상으로 광범위하게 적용 가능한 지능형 의사결정 모델을 개발하기 위한 연구이다. 개발된 모듈은 다양한 주행환경에 적용하기 위하여 독립적인 모듈 구성형태를 갖추고 이를 유기적으로 운영하기 위한 플랫폼을 기반으로 각 모듈을 Deep Learning 기법을 이용한 강화학습을 통해 이동체의 차선변경 의사결정 및 안전 확보에 관하여 연구하였다. 운행 중인 자율주행 이동체는 주행상태의 변경을 위하여 기능별로 속도결정 모델과 차선변경 의사결정이 반드시 선행되어야 이동체의 다음 행동이 이루어 질 수 있는 특징을 갖고 있다. 또한, 일반 도로상에 주행 중인 이동체들이 모두 자율주행 기능을 갖춘 상태라면 특정 이동체의 돌발적인 변화 및 주변 환경의 변화가 발생하면 그에 따른 각각의 이동체들 사이에 발생할 수 있는 연쇄반응 요소는 고려되기 어려운 특징을 갖고 있다. 이런 부분을 함께 고려하여 자율주행 플랫폼에 적용하고 이를 위한 개별 이동체의 차선변경 의사결정 시스템 연구와 모듈형 학습 방법을 이용하여 시스템 복잡도를 줄이고 학습시간의 단축과 향후 모델의 교체를 할 수 있는 의사결정 모델을 연구하였다.

Abstract This paper is intended to develop a decision model that can be applied to autonomous vehicles and autonomous mobile vehicles. The developed module has an independent configuration for application in various driving environments and is based on a platform for organically operating them. Each module is studied for decision making on lane changes and for securing safety through reinforcement learning using a deep learning technique. The autonomous mobile moving body operating to change the driving state has a characteristic where the next operation of the mobile body can be determined only if the definition of the speed determination model (according to its functions) and the lane change decision are correctly preceded. Also, if all the moving bodies traveling on a general road are equipped with an autonomous driving function, it is difficult to consider the factors that may occur between each mobile unit from unexpected environmental changes. Considering these factors, we applied the decision model to the platform and studied the lane change decision system for implementation of the platform. We studied the decision model using a modular learning method to reduce system complexity, to reduce the learning time, and to consider model replacement.

Keywords : Autonomous Vehicles, Deep Learning, DQN, Modular Network, Reinforcement Learning

본 논문은 2017년도 한국기술교육대학교 교수교육연구진흥과제로 수행되었음.

*Corresponding Author : Jongwon Kim(KOREATECH)

Tel: +82-41-560-1170 email: kamuia@koreatech.ac.kr

Received March 30, 2018

Revised May 8, 2018

Accepted June 1, 2018

Published June 30, 2018

1. 서론

현재 자율주행 자동차 기술은 인공지능 기술의 발달로 주변 환경을 센서를 통해 인지하고 상황에 따라 대응할 수 있는 능력을 갖추었으며 이를 이용한 반-자율주행 기능을 갖춘 자동차들이 생산되고 있다. 자율주행 자동차는 장착된 센서들을 통해 주변 환경 정보를 수집하여 Cloud-Server에 저장하고, 분석하며, 저장된 데이터를 바탕으로 재학습하며 점점 발전 하고 있다. 이는 곧 자율주행 자동차가 상용화 될 것이며, 차량 제조사 마다 각각의 지능형 알고리즘을 탑재한 완성차가 나올 것이 분명하다. 각 제조사 별 시험주행과 검증을 마친 완성차가 상용화 되면 도로 위에는 자율주행 자동차로 가득 차게 될 것으로 예측된다. 자동차 제조사들은 고속도로에서의 오토파일럿 기능을 추가 안전기능을 제공하고 있으며 고급 운전자 보조 시스템(ADAS)은 자동차가 차선을 지키고 전방차량을 탐지하는 기능을 수행한다[1]. 영상과 심층 신경망을 이용하여 보행자를 탐지하는 연구[2], 차선 변경 궤도 예측[3] 자율 주행 자동차를 위한 충돌 방지 연구가 이루어지고 있다[4].

지금의 자율주행 기술은 자동차가 GPS, Radar, LiDAR 등의 센서를 통해 스스로 인지하고 대응하는 것에 초점이 맞추어져 연구가 진행되고 있다[5]. 하지만 기존의 연구는 한 대의 자율주행 자동차가 어떠한 행위 결정을 했을 때 다른 자율주행 자동차에 미치는 영향은 고려되지 못하는 문제점을 가지고 있다. 본 연구의 목적은 도로 위에 자율주행 자동차만 운행된다는 가정 하에 차량의 진행 경로 상에 예상치 못한 장애물이 발생되어 그것을 회피하기 위한 차선변경을 수행 할 때, 주행 경로에 존재하는 여러 차량들 중에 어느 한 차량의 희생 없이 모든 차량이 원활하게 통행하기 위한 자율주행 구조와 이를 위한 모듈학습 방법을 제시하였다.

장애물 회피를 위한 모듈형 학습은 DQN(Deep Q-Network)를 사용한 강화학습을 진행하지만 자율주행 자동차의 속도결정, 차선변경 결정의 두 가지 결정 모듈을 나누어 학습하는 방법으로 학습 시간의 단축과 프로 그래밍의 편의성을 도모하였다.

본 연구의 2장에서는 자율주행 자동차가 장애물을 대하여 회피하기 위한 차량의 센서 정보수집 체계, 속도 및 차선변경에 대한 의사결정 시 고려해야 할 차량 및 외부 환경 요소들을 정의하였다. 3장에서는 속도결정과 차선

변경 결정 모델 학습을 위한 Deep Reinforcement Learning 네트워크 구조와 Reward 정책에 대하여 기술하고, 4장에서 학습과 시뮬레이션에 대한 결과를 기술하였다.

2. 자율주행 자동차 이동 결정 모델

자율주행 자동차의 주행경로 전방에 장애물이 발생하면 차선변경 활동이 반드시 필요하다. 하지만 인접 차선에 다른 자동차들이 주행하고 있을 때 자율주행 자동차의 차선결정 시점이 다른 차들의 운행에 영향을 미치게 된다. 그래서 안전을 위해 모든 차들이 지나가고 난 뒤에 차선 변경 결정(ACTION)을 하게 되면, 자율 주행 자동차는 장시간 저속운행 또는 정지를 통해 차선변경 활동을 위한 대기시간이 길어지는 상황에 빠지게 된다. 또한, 무리한 차선 변경은 기존에 진행하던 후방 차들의 갑작스런 감속과 이에 따른 충돌을 야기하기 때문에 그 지점을 통과하는 모든 차량들의 평균속도가 현저히 줄어들게 된다.

자율주행 자동차는 주변 상황의 정보를 센서를 통해 수집하고, 진로 상에 있는 장애물 정보를 공공 통신망을 통해 수신 받을 때, 진로 상의 모든 자율주행 차량들이 원활하게 회피 해 나가기 위한 결정 모델을 정의하여 충돌 없이 평균 주행속도를 향상시키는 결정 모델이 필요하며, 최소정보를 이용한 자율주행 차량의 행동결정 모델은 Fig. 1과 같다.

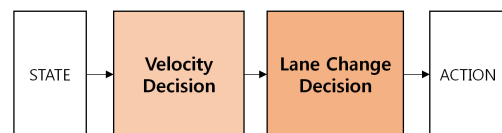


Fig. 1. Block diagram of Autonomous vehicle.

행동결정 모델을 위한 정보는 차량에 부착된 거리센서를 통해 각 영역별 주변 차량 혹은 장애물에 대한 정보, 차량의 속도, 진행차선의 상태정보, 별도의 통신망을 이용해 수신된 진로상의 장애물에 대한 차선, 거리 정보가 포함된다. 구역별 센서 정보는 Fig. 2와 같이 정의하여 각 센서로부터 수집된 정보는 미터단위로 산출하고, L과 R은 인접차선의 주행 차량 또는 장애물 유·무로 판단하여 1 또는 0의 값으로 처리하고 이를 구역별 정보처리 체계에 맞추어 사용한다.

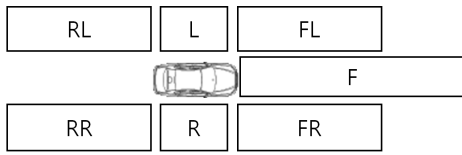


Fig. 2. Sensing informations by Areas

2.1 속도 결정

속도결정 모델은 앞 차량과의 거리 F와 FL/FR구역에 위치한 차량의 방향지시등 정보를 입력 받아 감속, 유지, 가속을 하게 된다. 이때, 앞 차량과의 안전거리에 대한 정의는 식 (1)과 같다.

$$v = \text{velocity of car (m/s)}$$

$$\text{safe_distance} = \begin{cases} v - 15, & \text{if } v > 16 \\ 1, & \text{otherwise} \end{cases} \quad (1)$$

가·감속에 대한 속도결정 모델은 Fig.3과 같이 가정하였다. 이 때, FL/FR에 위치한 차량의 방향지시등이 현재 진행 중인 차선으로 켜진다면 양보를 위한 감속은 안전거리 확보를 위한 감속의 25% 성능으로 가정하였다.

```

velocity : v(m/s) (Max 16.6667m/s)
time interval : Δt = 0.01(sec)
action : acceleration, deceleration, maintain
if action == acceleration then
    vt = 10.0 × Δt + vt-1
else if action == deceleration then
    if turn_signal == on then
        vt = -5.0 × Δt + vt-1
    else
        vt = -20.0 × Δt + vt-1
else
    vt = vt-1
    
```

Fig. 3. Definition of Velocity.

2.2 장애물 회피를 위한 차선변경 의사결정

센서로 측정할 수 없는 장애물이 자율주행 자동차의 진로 상에 존재할 때, 차간 통신(Vehicle to Vehicle, V2V), 공공망 통신(Vehicle to Internet of the public, V2I)을 통해 장애물이 위치하고 있는 차선정보와 장애물의 위치 정보를 수신 받는다[6].

이 정보를 근거로, 현재 주행차선 상에 장애물이 있는 경우, 어느 방향으로 차선을 변경해야 하는지, 어느 시점에 차선 변경을 해야 주행 진로 상에 있는 차량들이 원활히 지나갈 수 있는지에 대한 판단을 실시하고, 차선 변

경이 결정된 시점에 방향지시등을 켜고 Fig. 1의 행동(Action)이 이루어 질 수 있다.

차선 이동에 대한 방향이 결정되었다면 해당 시점에서, 해당 차선으로 이동하기 위한 공간 확보와 이동작업을 수행하게 된다. Fig. 2에서 나타난 주행차량을 중심으로, 충돌 없는 차선 변경을 위해 L의 방향으로 이동한다면 RL에 위치한 차량이 유지해야 할 안전거리의 30%, F와 FL에 위치한 차량과 안전거리의 30% 이내, L에 위치한 차량이 없는 공간을 확보한다. 공간 확보를 위한 가·감속 또는 속도 유지를 할 수 있으며 공간 확보가 되었을 때 이동하는 절차를 수행한다. 차선 변경과정에서는 2.1장에서 언급한 속도결정보다 우선하여 가·감속을 수행한다.

3. Deep Reinforcement Learning

2장에서 제안된 자율주행 자동차의 의사결정모델에 대하여 학습과정을 거칠 때 2가지 모델을 한 번에 학습하는 것이 아니라 결정 모델별로 따로 학습시키는 방법을 사용했다. 결정 모델별로 따로 분류하여 학습시키는 방법은 각 결정 모델별로 유효한 입력데이터만 쓰기 때문에 학습시간이 단축되며, 추후 다른 모델 적용을 위해 다른 성능을 갖는 모델로 교체할 수 있는 장점이 있다.

각 모델의 학습에 대해서는 Google DeepMind에서 Atari 게임에 대해 Deep Reinforcement Learning을 적용한 방식을 사용했다.

3.1 Deep Q-Network

DQN은 기본적으로 Q-Learning을 기반으로 환경을 State와 Action, Reward, Agent로 구분하여 Agent가 State에 대해 가장 Reward를 많이 받는 Action을 하게끔 학습시키는 강화학습을 사용한다[7].

Google DeepMind 팀은 Atari게임에 관한 실험을 통해 심층신경망을 이용한 강화학습을 통해 Correlation과 Non-stationary targets 과 같은 주요 문제의 해결방법을 찾아냈다. Correlation 문제는 과거 State에서 Agent가 취한 Action을 저장하고 학습할 때 시간에 대한 상관관계가 없도록 무작위로 Replay 하는 방식을 통해 해결하였고[8], Non-stationary targets 문제는 지속적으로 Weight가 변화되며 학습되고 있는 심층신경망을 Target 으로 지정하지 않고, 심층신경망을 이중으로 구성하여

일정시간마다 학습신경망과 Target 신경망을 복사함으로써 Non-stationary targets 문제를 해결하였다[9].

본 논문에서도 심층 신경망을 사용하는 강화학습의 문제점을 해결 하고자 Capture & Replay 방식과 Target Neural Network를 두고 심층신경망 모델을 학습하였다.

3.2 속도결정 모델 학습

속도결정 모델을 학습하기 위한 심층신경망의 구조는 Fig. 4와 같이 State에 해당하는 6개의 입력과 은닉층 (Hidden Layers) 그리고 Action에 해당하는 3가지 출력을 갖게 하였다.

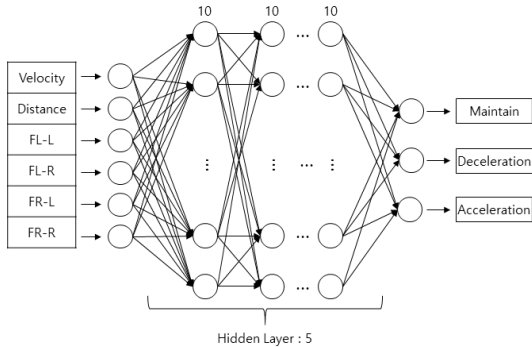


Fig. 4. The Deep Neural Networks Structure of Velocity definition model.

속도결정 모듈에서는 앞 차량과의 안전거리 확보 및 주변차량의 방향 지시등을 고려 한 효과적인 속도결정에 대한 학습이 필요하다. 따라서 State와 Action에 대한 정의 Fig. 5와 같다.

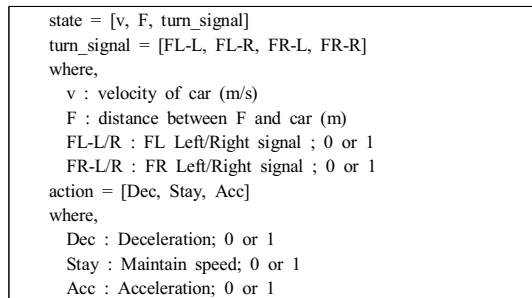


Fig. 5. Definition of State and Action for the velocity decision.

각 State와 Action에 대한 Reward 정책은 Fig. 6과 같다.

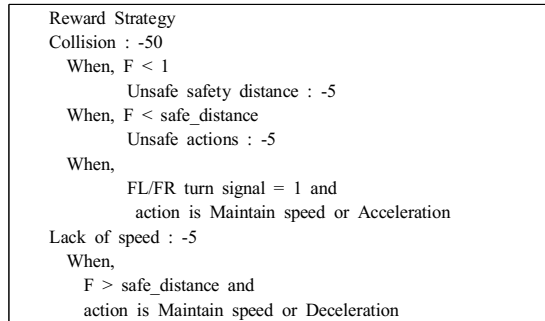


Fig. 6. Definition of reward strategy for the velocity decision.

초기 State 중 F의 차량의 속도와 위치는 무작위로 지정되며, FL/R차량의 방향지시등은 episode 진행 중에 무작위로 지정 된다.

3.3 차선변경 결정 모델의 학습

차선변경 결정 모델을 학습하기 위한 심층신경망의 구조는 Fig. 7과 같이 State에 해당하는 18개의 입력을 갖는데, 이것은 t-2부터 t까지의 State를 결합하여 입력시킴으로써 심층신경망에서 주변 차량들의 속도변화에 따른 최적의 Action을 결정하기 위함이다.

차선 변경 결정 모델은 주변 차량의 속도와 거리에 따라 변경할 방향의 차선에 대한 안전한 공간을 확보한 뒤 차선변경을 통해 차량이 이동하는 순서를 기준으로 학습하고 차선변경 모델의 State와 Action의 정의는 Fig. 8과 같다.

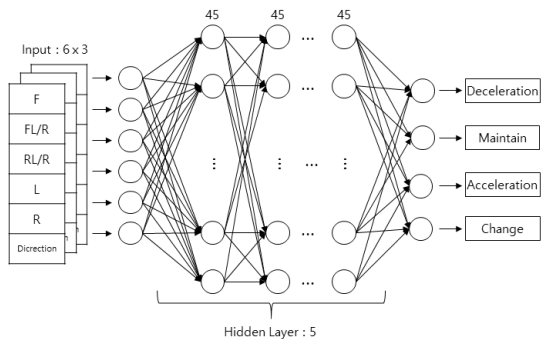


Fig. 7. The Deep Neural Networks Structure for lane changing.

```

state = [[Ft, FL/Rt, RL/Rt, Lt, Rt, direction],
         [Ft-1, FL/Rt-1, RL/Rt-1, Lt-1, Rt-1, direction],
         [Ft-2, FL/Rt-2, RL/Rt-2, Lt-2, Rt-2, direction]]
direction = [Left, Right]
action = [Dec, Stay, Acc, Move]
where,
F, FL/R, RL/R : distance from car (m)
L, R : presence or absence ; 0 or 1
Left, Right : directions; 0 or 1
Dec. : Deceleration; 0 or 1
Stay : Maintain speed; 0 or 1
Acc. : Acceleration; 0 or 1
Move : Change Lane; 0 or 1
    
```

Fig. 8. Definition of State and Action for the lane changing.

과거의 데이터를 기반으로 각 구역별 센서 데이터로부터 속도를 계산하고, State는 과거의 상태를 고려하여 t-2부터 t까지의 2차원 상태 데이터를 이용하여 학습한다. 이를 위한 각 State와 Action에 대한 Reward 정책은 Fig. 9와 같다.

```

Reward Strategy
Collision : -1,000
When,
    Safe distance of located in RL/R < 30%,
    Safe distance of located in FL/R < 30%,
    R/L = 1
Successful lane change : 0
Otherwise : -1
    
```

Fig. 9. Definition of reward strategy for the lane changing.

위의 Reward 정책 중 기타의 경우 차선 변경 전까지 매 Frame 당 변경할 차선의 빠른 공간 확보를 위하여 (-1)의 Reward를 고려한다. 초기 State 중 변경할 차선의 방향은 episode 시작 시점에 무작위로 설정되며 episode가 끝날 때 까지 변경되지 않는다. F의 차량은 안전거리가 확보된 시점을 기준으로 초기위치를 설정하고 속도는 5~60km/h 내에서 무작위로 설정된다. 또한 주위의 차량은 각각 2%의 확률로 없을 수 있으며 Target 차량 주위에 무작위로 배치되며 속도결정 학습을 통해 이미 학습된 속도결정 모듈을 적용하여 동작된다.

4. 시뮬레이션 및 결과

3.2절에서 언급된 속도결정 모듈의 학습은 앞 차량과 안전거리 확보 및 주변 차량의 방향지시등이 켜졌을 때 속도를 줄여 안전한 주행을 학습하였고 Fig. 10과 같이

시뮬레이션 하였다.

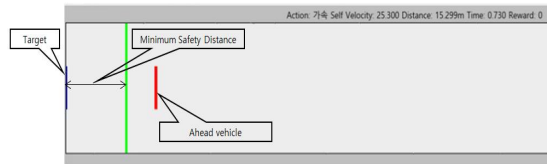


Fig. 10. The simulation of velocity decision model.

빨간색 라인은 일정 속도로 주행하고 있는 전방차량을 나타내며 이 차량의 속도는 0~15km/h의 느린 무작위 속도를 가진다. 초록색 라인은 전방에 주행하는 차량의 속도 변화에 대응한 Target 차량까지의 안전거리를 나타낸 것이다. 3.2장을 통해 언급한 속도결정 모델에 의한 학습 결과는 1,000회의 평가 결과가 모두 안전거리와 동일한 거리를 확보하며 주행하는 결과를 볼 수 있었다.

Fig. 11과 같이 3.3절에서 언급한 차선변경모듈 학습의 경우, 적용되는 모든 차량의 주행에 속도 결정모듈이 적용되었고, 그 중 한 차량이 차선변경을 하고자 할 때 충돌 없이 차선변경을 할 수 있는 조건에 대하여 학습하였다.

Target 차량(빨간색 블록)이 차선변경을 하려는 대상 차량이고 이를 중심으로 각각의 영역에 차량이 속도변화를 갖으며 주행하고 있다. 이때 주변차량의 속도가 전방에 예견된 장애물에 의해 속도저하가 발생하고, 전방의 차량과 변경 할 차선의 차량과 충돌 없는 차선변경을 위하여 안전한 이동시점이 결정되어야 한다.

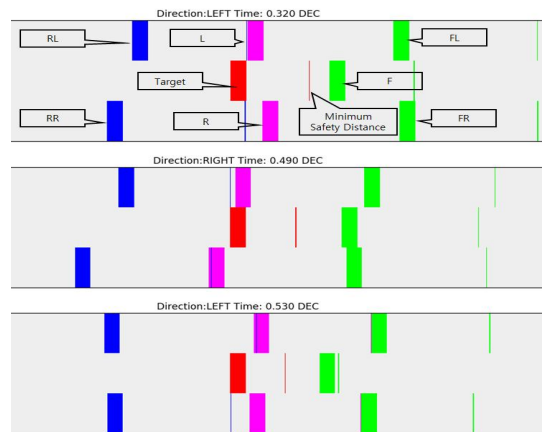


Fig. 11. The simulation of lane changing.

차선변경 결정 모델의 평가는 1,000번을 수행하였으며, 변경하려는 차선은 무작위, 전방의 차량의 속도는 0~60km/h의 무작위 속도, 이동하려는 차선에 위치한 바로 옆 차량(보라색)은 2%의 확률로 존재하거나 존재하지 않는 시나리오로 평가하였다. Fig. 11과 같이 주행 시간이 경과됨에 따라 각각의 차량의 속도에 따른 안전 거리 변화는 시뮬레이션 상에서 차량의 상태정보로 나타내어지고, 평가 결과 100% 충돌 없이 차선변경을 수행하는 좋은 결과를 얻을 수 있었다.

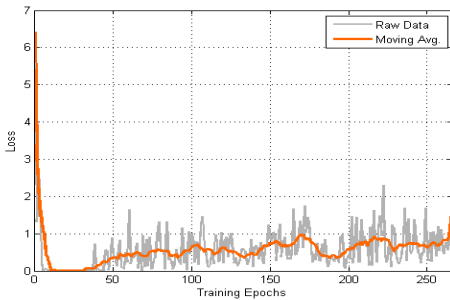


Fig. 12. Loss function value for each iteration (Velocity decision model).

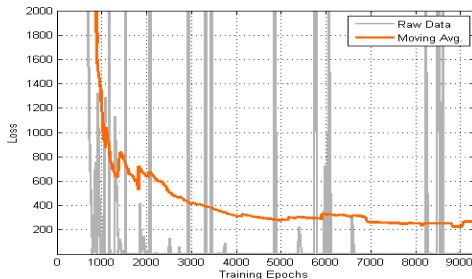


Fig. 13. Loss function value for each iteration (Lane changing decision model).

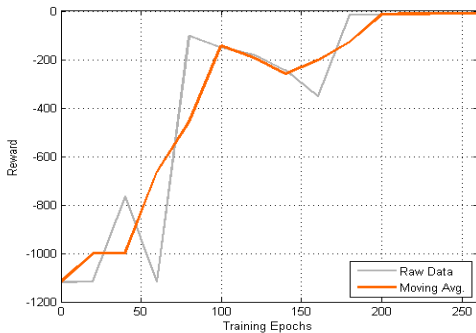


Fig. 14. Reward for each iteration (Velocity decision model).

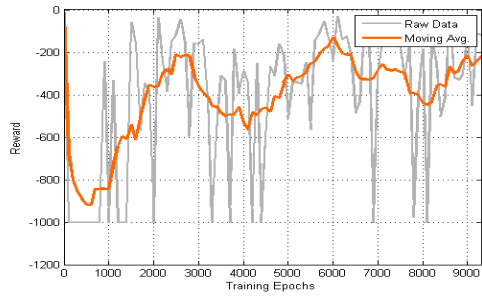


Fig. 15. Reward for each iteration (Lane changing decision model).

속도와 차선변경 학습 모델의 Loss 값은 Fig. 12,13과 같이 학습의 반복 횟수가 증가할수록 감소하는 형태를 볼 수 있다. 또한, Fig. 14, 15에서 보는 것과 같이 학습이 진행됨에 따라 받는 Reward도 증가함을 볼 수 있다.

5. 결론

자율주행 이동체의 진로상에 있는 장애물 회피를 위해 반드시 필요한 기능을 모듈별로 구성하고 주어진 플랫폼 내에서 활용 가능한 정보를 이용하여 차선변경 활동을 위한 의사결정 시스템을 연구하였다. 이 연구는 3차선 내에 모든 자율주행 자동차의 돌발적인 상황을 가정하여 다른 차량에 미치는 영향을 고려한 Target 차량의 차선변경 의사결정 과정과 그에 따른 결론의 적합성을 시뮬레이션을 통해 검증하였다.

자율주행 이동체의 장애물 회피를 위한 의사결정 모델을 한 개의 모델로 통합하여 학습할 경우, 속도결정에서 고려한 6개의 환경 요소와 차선변경 결정에서 고려한 18개의 환경 요소를 합쳐 동시에 24개의 환경요소를 입력해야 한다. 이것은 차선변경 모델부분에서 최소 6배 이상의 복잡한 환경을 고려해야 한다. 또한 기존에 학습한 속도결정 부분을 교체하기 위해서는 전체 네트워크를 다시 학습시켜야 하는 단점을 갖게 된다. 따라서 각각의 모듈은 의사결정 모델을 기반으로 모듈별 학습을 통해 학습시간의 단축과 기능적 모델의 변경을 고려한 재학습 등이 고려되어 이상적인 의사결정 모델의 도출을 위한 구조적 효과성을 검증하였다.

향후 장애물 정보에 대한 구체적인 내용의 전달이 플랫폼 내에 활용 가능하거나, 주변 차량의 자율주행 상태 정보의 구체성이 Target 차량에 공유되는 조건이 가능하

다면 각 모듈별 기능을 세분화 한 기능별 모듈들의 구성 또는 전체적인 플랫폼의 변경 등이 현실세계에서의 적합성 향상과 각각의 자율주행 차량의 통합 운영에 크게 도움 될 수 있을 것으로 생각된다.

References

- [1] B. Huval, T. Wang, S. Tandon, J. Kiske, W.Song, J. Pazhayampallil, M. Andrilukam, P. Rajpurkar, T. Migimatsu, R. Cheng-Yue, F.Mujica, A. Coates, and A. Y. Ng, "An empirical evaluation of deep learning on highway driving," arXiv preprint, arXiv:1504.01716, pp. 1-7, Apr. 2015.
- [2] D. Tome, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi and S. Tubaro, "Deep convolutional neural networks for pedestrian detection," *Signal Processing: Image Communication*, vol. 47, pp. 482-489, May. 2016. DOI: <https://doi.org/10.1016/j.image.2016.05.007>
- [3] R. S. Tomar, S. Verma, "Neural network based lane change trajectory prediction in autonomous vehicles," *Transactions on computational science XIII, Springer*, pp. 125-146, Berlin Heidelberg, 2011. DOI: https://doi.org/10.1007/978-3-642-22619-9_7
- [4] D. Geronimo, A. M. Lopez, A. D. Sappa, T. Graf, "Survey of pedestrian detection for advanced driver assistance systems," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 7, pp. 1239-1258, July. 2010. DOI: <https://doi.org/10.1109/TPAMI.2009.122>
- [5] Hyunmin Chae, Chang Mook Kang, Chung Choo Chung, Jun Won Choi, "Deep Reinforcement Learning based Autonomous Braking System for safe Urban Driving," *KSAE Conference*, pp. 625-629, May. 2017
- [6] Ke Cao, et al. "Efficient Urban Broadcast Protocol for V2V Communications with Relay Control", *IEEE Vehicular Networking Conference*, pp. 24-30, 2013. DOI: <https://doi.org/10.1109/VNC.2013.6737586>
- [7] C. Watkins and P. Dayan. "Q-learning. Machine learning," 8(3-4), pp. 279 -292, 1992. DOI: <https://doi.org/10.1007/BF00992698>
- [8] MNIH, Volodymyr, et al., "Playing atari with deep reinforcement learning." arXiv preprint, arXiv:1312.5602, 2013.
- [9] MNIH, Volodymyr, et al., "Human-level control through deep reinforcement learning," *Nature*, 518.7540: 529-533, 2015. DOI: <https://doi.org/10.1038/nature14236>

강 병 준(Byung-Jun Kang)

[정회원]



- 2015년 8월 : 한국기술교육대학교 전기전자통신공학부(학사)
- 2015년 8월 : 한국기술교육대학교 전기전자통신공학부(공학석사)
- 2015년 9월 ~ 현재 : 한국기술교육대학교 전기전자통신공학부 박사과정

<관심분야>

Deep Learning, Neural Networks, System Control, Computer Science

김 중 원(Jongwon Kim)

[정회원]



- 2007년 8월 : KOREATECH 전기전자공학과 (공학박사)
- 2016년 9월 ~ 현재 : KOREATECH 기전융합공학과 조교수

<관심분야>

지능제어, 기술융합, 산업응용 시스템 제어.