



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

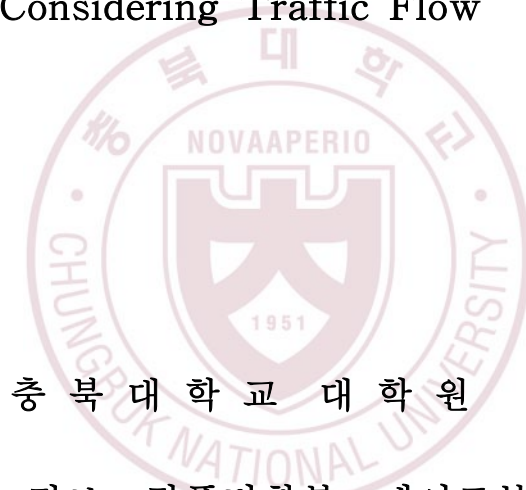
이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

공학석사학위논문

심층강화학습 기반의 자율주행 차량의  
교통흐름을 고려한 차선변경 판단 및 제어

Deep Reinforcement Learning Based Decision Making  
and Control of Lane Change for Autonomous Vehicle  
Considering Traffic Flow



충북대학교 대학원

전기·전자·정보·컴퓨터학부 제어로봇공학전공

조 기 환

2021 년 8 월

공학석사학위논문

심층강화학습 기반의 자율주행 차량의  
후방 차량을 고려한 차선변경 판단 및 제어

Deep Reinforcement Learning based Decision Making  
and Control of Lane Change for Autonomous Vehicle  
Considering Traffic Flow

지도교수 박 태 형

전기·전자·정보·컴퓨터학부 제어로봇공학전공

조 기 환

이 논문을 공학석사학위 논문으로 제출함.

2021 년 8 월

본 논문을 조기환의 공학석사학위 논문으로 인정함.

심 사 위 원 장      기 석 철 ①

심 사 위 원      박 태 형 ①

심 사 위 원      윤 희 상 ①

충 북 대 학 교 대 학 원

2021 년 8 월

# 차 례

차례 .....	i
그림 목차 .....	iii
표 목차 .....	v
Abstract .....	vi
<b>I. 서 론 .....</b>	<b>1</b>
1.1 연구 배경 .....	1
1.2 기존 연구 .....	4
1.2.1 차선변경 판단 방법 .....	4
1.2.2 차선변경 경로 생성 방법 .....	7
1.2.3 차량 제어 방법 .....	9
1.3 심층강화학습 .....	11
1.4 연구 목적 .....	13
1.5 논문 구성 .....	14
<b>II. 시스템 구성 .....</b>	<b>16</b>
2.1 시스템 구성 .....	16
<b>III. 차선변경 판단 .....</b>	<b>20</b>
3.1 차선변경 판단 구조 .....	20
3.2 신경망 입출력 설계 .....	21
3.3 신경망 구조 .....	24
3.4 학습 방법 .....	25
3.5 충돌 판단 .....	26

<b>IV. 경로계획 및 제어</b> .....	29
4.1 차선변경 경로 생성 .....	29
4.2 횡방향 제어 .....	32
4.3 종방향 제어 .....	35
<b>V. 실험</b> .....	36
5.1 실험 환경 .....	36
5.2 실험 결과 .....	37
5.2.1 차선변경 판단 결과 .....	37
5.2.2 속도 보상계수에 따른 실험 .....	40
5.2.3 차량 수 변경에 따른 실험 .....	43
5.2.4 기존 방법과 성능 비교 .....	46
<b>VI. 결론</b> .....	47
참고문헌 또는 인용문헌 .....	48
감사의 글 .....	53

## 그림 목차

그림 1. 목표 차선 차량과의 충돌 시간 계산 .....	4
그림 2. 가속도를 사용한 차선변경 판단 .....	5
그림 3. DQN 기반 차선변경 시나리오 .....	6
그림 4. 2차 베지어 곡선을 사용한 차선변경 경로 생성 .....	8
그림 5. 클로소이드 곡선을 사용한 차선변경 경로 생성 및 비교 .....	9
그림 6. Pure Pursuit 방법 .....	10
그림 7. Stanley 방법 .....	10
그림 8. DQN(Deep Q Network) 구조 .....	12
그림 9. DeepSet-Q 신경망 구조 .....	13
그림 10. CARLA 시뮬레이터 .....	16
그림 11. 실험을 위한 4차선 자동차 전용도로 .....	17
그림 12. 심층강화학습을 사용한 차선변경 판단 및 제어 시스템 구성 .....	20
그림 13. 차선변경 판단 구조 .....	20
그림 14. 자차와 주변 차량 .....	22
그림 15. 신경망의 입력 .....	23
그림 16. 신경망 구조 .....	25
그림 17. 차선변경 판단 알고리즘 .....	27
그림 18. 차선변경이 불가능한 경우 .....	27
그림 19. 안전거리 계산 .....	28
그림 20. 3차 베지어 곡선과 조절점 .....	30
그림 21. 베지어곡선을 사용한 차선변경 경로 .....	30
그림 22. 기하학적 모델 .....	32
그림 23. Pure Pursuit .....	33
그림 24. 상황에 따른 종방향 제어 방법 .....	35
그림 25. 실험 환경 .....	36

그림 26. 기존 방법(MOBIL)과 제안 방법의 평균속도 .....	38
그림 27. 기존 방법(MOBIL)과 제안 방법의 후방차량 감속률 .....	38
그림 28. 속도보상계수에 따른 평균속도 학습 그래프 .....	40
그림 29. 후방차량 감속률 보상계수에 따른 평균속도 학습 그래프 .....	41
그림 30. 후방차량 감속률 보상계수에 따른 후방차량 감속률 학습 그래프 .....	42
그림 31. 차량 수에 따른 평균속도 .....	44
그림 32. 차량 수에 따른 후방차량 감속률 .....	45



## 표 목차

표 1. 차선변경 판단 방법에 따른 장점 및 단점 .....	6
표 2. 신경망 입력 .....	21
표 3. 신경망 출력 .....	23
표 4. 강화학습 보상 .....	25
표 5. 기존 방법(MOBIL)과 제안방법 비교실험 결과 .....	39
표 6. 차량 수에 따른 평균 속도 .....	44
표 7. 차량 수에 따른 후방차량 감속률 .....	45
표 8. 기존 방법과 제안 방법 비교 .....	46

# Deep Reinforcement Learning based Decision Making and Control of Lane Change for Autonomous Vehicle Considering Traffic Flow

Jo, Gi Hwan

*Department of Control and Robot Engineering  
Graduate School, Chungbuk National University  
Cheongju, Korea  
Supervised by Professor Park, Tae Hyoung*

## Abstract

Autonomous vehicle driver negligent prevent accidents and reduce traffic congestion. Lane change determination for autonomous vehicles is an important issue. When changing lanes, it is necessary to consider not only the speed of the own vehicle, but also the flow of traffic. However, it is difficult to consider the traffic flow in the existing logic-based lane change determination method. In this paper, we apply deep reinforcement learning to determine lane change taking into account speed maintenance and traffic flow. To determine the lane change, the reward of reinforcement learning and the structure of the deep neural network are designed. It then controls the vehicle by generating a route that matches the lane change decision. To evaluate the proposed method, we tested it with simulations. The average speed is faster than the traditional logical lane change method and is less disruptive to traffic flow.

---

\* A thesis for the degree of Master in August 2021.

# I. 서 론

## 1.1 연구 배경

자율주행 자동차는 운전자 또는 승객의 조작 없이 스스로 운행이 가능한 자동차를 말한다. 자율주행 자동차는 카메라, 라이다, 레이더, 위성항법장치 등 여러 센서를 사용하여 환경을 인지하고, 판단하고 주행 경로를 생성한다. 그리고 주어진 경로를 추종할 수 있도록 차량을 제어한다. 자율주행자동차는 운전자의 과실로 발생하는 사고를 예방하고, 운전자의 편의성 향상, 교통혼잡 감소 [1] 등 여러 이점이 있다. 현재 자율주행 기술은 SAE(Society of Automation Engineers)[2] 기준으로 0~5단계 중 2~3단계이다. 2단계는 사람의 운전을 보조해주는 부분 자율주행이 가능한 수준이다. 이는 운전자가 항시 전방을 주시해야 하며 차량을 제어할 준비가 되어 있어야 한다. 3단계는 특정 상태에서 자율주행이 가능한 상태다. 운전자가 손, 발을 자유롭게 할 수 있으며 차선변경 등 자율주행차가 스스로 판단하여 운행한다. 현재 고속도로와 같은 제한된 환경에서 차선 유지와 속도 유지가 가능한 운전자 보조시스템을 탑재한 차량이 양산되고 있다[3].

차선변경의 목적은 크게 두 가지이다. 첫 번째는 고속도로 진, 출입과 같은 목적 차선으로 이동하기 위한 것이다. 두 번째는 속도 유지를 위해 전방의 느린 차량을 피하는 것이다. 본 논문에서는 속도유지를 위한 차선변경 문제를 다룬다. 차선변경 판단의 목적은 자차의 속도를 유지하는 것이다. 동시에 주변 차량과의 충돌을 회피하고 주변 차량에 영향을 최소화하는 것이다. 후방차량

과 가까울 때 차선변경을 시도하면 후방차량의 감속으로 교통체증을 유발할 수 있다.

안전거리 기반 차선변경 판단 방법[4-5]은 일반적으로 자차와 현재 차선의 앞, 뒤 차량 그리고 차선변경 목표 차선의 앞, 뒤 차량의 정보를 이용하여 판단한다. 계산이 간단하다는 장점이 있지만, 주변 차량의 속도 변화에 대해 고려를 하지 않는다. 자차의 차선변경으로 인해 후방차량의 속도가 감소할 수 있다. 이를 해결하기 위해 가속도를 고려하는 차선변경 방법[6]이 연구되었다. 자차의 차선변경으로 인해 후방차량의 가속도가 변하는 것을 고려하였다. 또한 최적화 기반의 차선변경 방법[7-9]도 연구되었다. 최적화 기반 방법은 제약 조건을 통해 안전거리를 만족하면서 주변 차량의 가속도도 고려한다. 최적화 기반 방법은 차선변경 판단뿐만 아니라 자차의 속도도 동시에 계산할 수 있는 장점이 있다. 하지만 계산량이 많아 실시간으로 사용하기 어렵다는 단점이 있다.

위에 소개된 방법들은 자차와 현재 차선의 앞, 뒤 차량 그리고 차선변경 목표 차선의 앞, 뒤 차량의 정보만 이용한다. 제한적인 주변 차량 정보만으로는 고속도로 환경에서 국소적인 차량정체를 인지하고 회피할 수 없다. 이를 해결하기 위해 강화학습 기반의 차선변경 판단 방법이 연구되고 있다. 차선변경 판단과 차량제어를 모두 강화학습으로 하는 방법[10-12] 과 차선변경 판단은 강화학습으로, 제어는 기존 고전 제어로 하는 방법[13-18]이 있다. 강화학습으로 차량제어까지 하는 방법들은 안전하게 차선변경을 하는 것이 목적이다. 차선변경 판단만 강화학습으로 하는 경우는 자차의 속도 유지 또는 목표 차선으로 이동하는 것이 목적이다. 이러한 강화학습을 사용한 방법의 단점은 성능을

내기까지 학습 시간이 오래 걸리는 것이다. 하지만 강화학습을 사용하면 많은 차량을 고려할 수 있어서 전방의 차량정체에 대응할 수 있다. 또한 차량 모델을 알지 못해도 차선변경 판단을 할 수 있기 때문에 전문가가 아니어도 문제를 해결할 수 있다.

## 1.2 기존 연구

### 1.2.1 차선변경 판단 방법

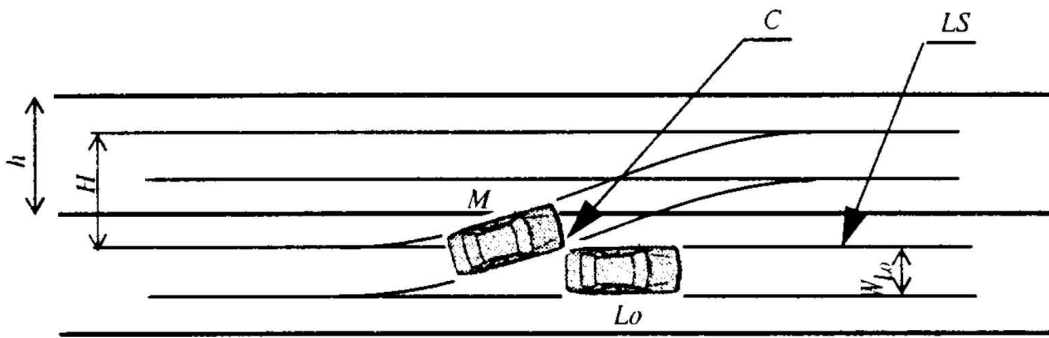


그림 1. 목표 차선 차량과의 충돌 시간 계산

자율주행 자동차의 차선변경 판단을 위해 H. Jula[4] 등이 제안한 방법은 그림 1과 같다.  $M$ 은 자차,  $L_o$ 는 현재 차선의 전방 차량이다. 자차와 전방 차량의 속도와 가속도가 일정하다고 가정하고 두 차량 간의 충돌 여부를 계산하여 차선변경 판단을 내린다. 하지만 충돌 여부만 계산할 뿐 다른 차량의 영향을 고려하지 않는다.

A. Kesting등이 제안한 방법[6]은 차선변경 시 자차와 후방차량의 가속도 변화를 고려하는 차선변경 판단 방법이다. 차선변경을 하지 않았을 때와 차선변경을 했을 때 자차와 후방차량의 가속도를 계산한다. 그림 2와 같이 자차의 현재 가속도  $a_c$ , 차선변경 후 가속도  $\tilde{a}_c$ 의 예상 가속도와 현재 후방차량  $o$ , 차선변경 후 후방차량  $n$ 의 가속도 변화를 비교한다. 가속도 이득 값을 계산하여 일정 값 이상이면 차선변경 판단을 한다. 이 방법은 가속도를 사용하여 차선

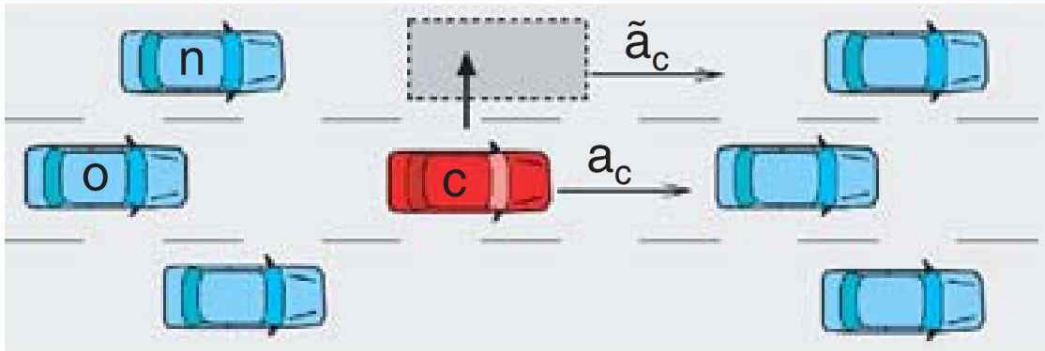


그림 2. 가속도를 사용한 차선변경 판단

변경 판단을 하므로 후방차량의 감속을 고려할 수 있다. 하지만 자차와 후방차량 그리고 바로 앞차만 고려하기 때문에 전방 정체 구간에 대해 대처를 할 수 없다.

T. Shi 등이 제안한 방법[13]은 강화학습을 사용한 차선변경 판단 방법이다. 현재 차선의 전방 차량, 목표차선의 전, 후방차량과의 거리, 그리고 자차, 현재차선의 전방 차량, 목표 차선의 전방 차량의 속도를 입력으로 하여 차선변경 판단을 한다. 보상은 차량 사이의 거리와 속도차이를 주어 안전한 차선변경을 할 수 있게 한다. 보상에 자차의 속도와 앞차와의 거리, 속도가 반영되었지만 후방 차량의 속도에 대한 보상은 없다. 후방차량의 속도를 고려하지 않으므로 대한 보상은 없으므로 목표 속도를 유지할 수 없다.

H. Carl-Hohan 등이 제안한 방법[14]은 Deep Q Network를 사용한 차선변경 방법이다. 그림 3와 같은 상황에서 트럭의 속도를 유지하는 것이 목표이다. 자차의 속도 그리고 좌, 우측의 차선 여부 그리고 자차 외의 모든 차량과의 상대 거리, 상대속도, 상대 차선 위치를 입력으로 사용한다. 그리고 출력에 차선변경 판단뿐만 아니라 속도 변화도 포함했다. 시간 단계마다 정규화된 속도

를 보상으로 주고, 충돌하거나 차선을 변경할 때 페널티를 주었다. 이 방법은 논리 기반 차선변경 판단 방법보다 평균속도가 빨라지는 것을 확인할 수 있었다. 하지만 자차의 속도 관련 보상만 있으므로 후방 교통흐름을 고려할 수 없다.

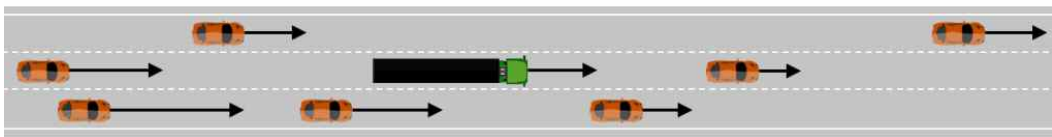


그림 3. DQN 기반 차선변경 시나리오

논리기반 차선변경 판단 방법과 강화학습 기반 차선변경 판단 방법을 소개하였다. 각각 방법의 장단점을 표 1에 나타냈다.

표 1. 차선변경 판단 방법에 따른 장점 및 단점

	관련연구	장점	단점
논리 방법	[4-9]	계산이 간단	차량 모델 필요 다수 차량 고려 어려움
강화 학습 방법	[10-18]	차량 모델 불필요, 다수 차량 고려 가능	속도유지, 교통흐름 둘 중 하나만 고려



### 1.2.2 차선변경 경로 생성 방법

J. Chan등이 제안한 방법[5]은 베지어 곡선을 사용하여 차선변경 경로를 만드는 방법이다. 자차와 전방 차량과의 정보를 사용하여 충돌 여부를 판단하여 차선변경판단을 결정한다. 차선변경 판단이 내려지면 현재 속도와 숙련된 운전자의 평균 차선변경 시간을 고려하여 경로를 생성한다. 경로를 생성할 때 하나의 베지어 곡선이 아닌 두 개의 곡선을 이어붙였다. 그리고 횡방향 가속도를 제한하여 편안함을 증가시켰다. 2차 베지어 곡선 두 개를 사용하여 계산 속도는 증가시켰다. 하지만 차량의 heading방향과 도착지점의 방향을 고려할 수 없어 곡선 경로 또는 차량이 도로와 평행한 상태가 아닐 때는 사용할 수 없다. 식 1과 같이 점  $P_0, P_1, P_2$ 와 점  $Q_0, Q_1, Q_2$ 을 사용한 베지어 곡선을 연결하여 차선변경 경로를 생성한다.

$$P(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2 \quad (1)$$

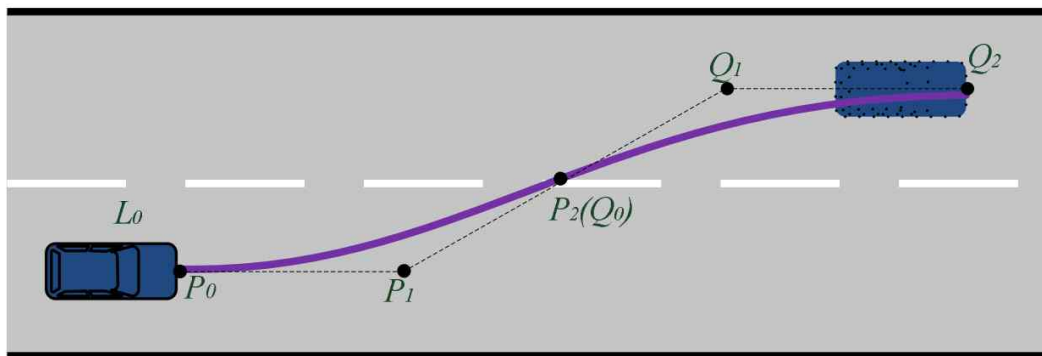


그림 4. 2차 베지어 곡선을 사용한 차선변경 경로 생성

M.H Song등이 제안한 방법[19]은 클로소이드 곡선을 사용하여 차선변경 경로를 생성하는 방법이다. 클로소이드 곡선은 실제 도로설계에 반영되는 곡선으로 고속도로 등의 직선 부와 회전구간을 잇는 완화곡선으로 사용된다. 이 방법은 혼잡한 교통상황과 원활한 교통상황의 차선변경 소요 시간을 다르게 하여 차선변경 경로를 생성한다. 비선형 경로 생성 방식인 Hybrid A\*방법과 베지어곡선 방법보다 횡 방향 가속도 변화량이 적은 것을 그림 5에서 볼 수 있다. 하지만 클로소이드 곡선은 계산이 복잡하여 다른 방법보다 실시간성이 떨어지며 차량의 heading 방향과 도로의 곡률을 반영할 수 없다.

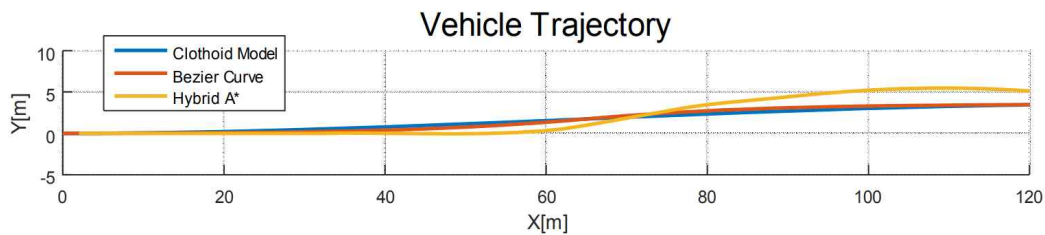


그림 5. 클로소이드 곡선을 사용한 차선변경 경로 생성 및 비교

### 1.2.3 차량제어 방법

차량을 제어하는 방법은 차량 모델에 따라 나눌 수 있다. 첫 번째 방법은 Geometry 모델을 사용한 Pure Pursuit 방법[20]이다. 차량의 뒤축을 기준으로 경로 위의 목표지점을 향하는 조향각을 구한다. Geometry 모델을 사용하기 때문에 계산이 매우 빠르고 간단하다. 외란에는 강하지만 회전 시 회전 반경 안쪽을 침범하는 것과 속도가 빠를 때 불안하다는 단점이 있다.

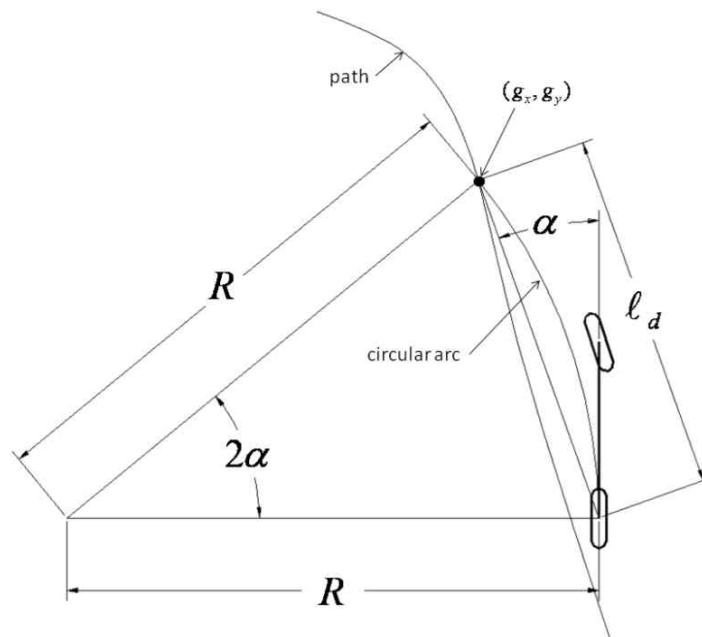


그림 6. Pure Pursuit 방법

두 번째 방법은 Kinematic 모델을 사용한 Stanley 방법[21]이다. 이 방법은 DARPA 그랜드 챌린지에서 우승한 Stanford대학이 사용한 방법이다. 차량의 앞차축을 기준으로 제어한다. 이 방법은 Pure Pursuit에서 나타난 회전 반경

안쪽을 침범하는 단점이 완화되었다. 하지만 조향각을 계산할 때 경로의 곡률이 필요하며 경로의 곡률이 연속이어야 한다. 또한 기준이 앞 차축이기 때문에 뒤 차축이 경로를 벗어나는 일이 자주 발생한다.

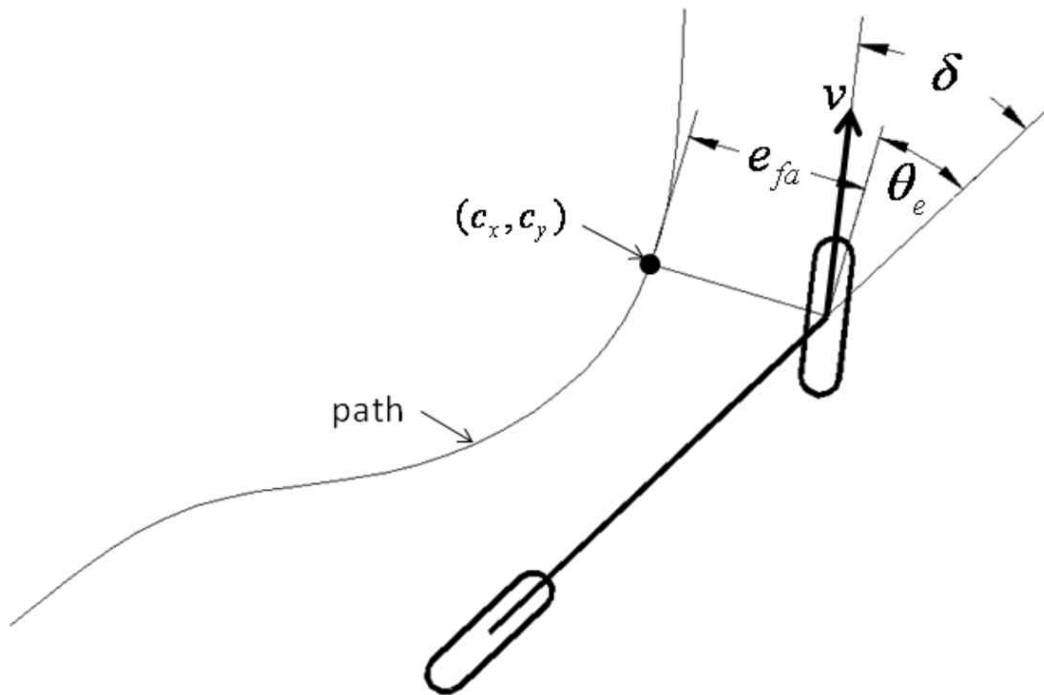


그림 7. Stanley 방법

### 1.3 심층 강화학습

DQN(Deep Q Network)[22]은 기존의 강화학습 방법인 Q-learning[23]에 심층신경망을 적용한 방법이다. Q-learning의 상태-행동 테이블을 신경망으로 근사화시켜 입력의 개수가 많고 범위가 클 때 학습이 어려운 문제를 해결하였다. 또한 학습 데이터를 한 번만 쓰지 않고 저장해 두었다 다시 사용하는 Replay Buffer를 사용하여 데이터 효율성을 증가시켰다. 하지만 연속적인 행동에는 적합하지 않다.

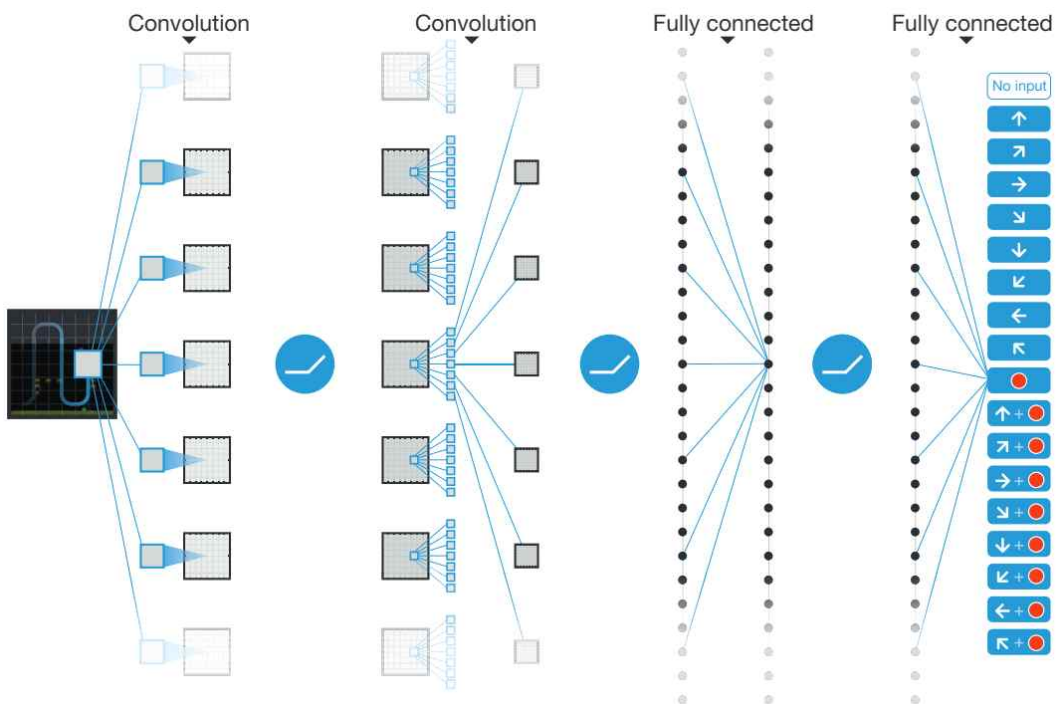


그림 8. DQN(Deep Q Network) 구조

DeepSet-Q[24]은 DQN 네트워크 앞에 차량의 특징을 하나의 벡터로 모아 그림 9와 같은 가변 신경망 구조를 만들었다. 각 차량별로  $\phi$  신경망을 통과시켜 특징 벡터를 추출한다. 추출한 특징 벡터를 Pooling을 통해 하나의 벡터로 모두 합친다. 그 후  $\rho$  신경망을 통과시켜 주변 환경 특징벡터를 추출한다. 이 특징벡터와 자차의 정보를 합쳐 DQN 신경망에 입력으로 사용한다. 이 방법은 차량의 수와 관계없이 신경망을 사용할 수 있어 다양한 환경에 사용할 수 있다.

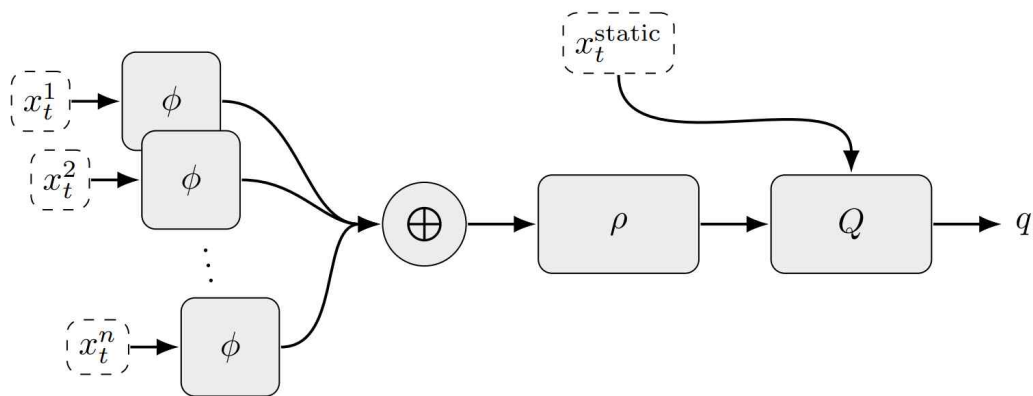


그림 9. DeepSet-Q 신경망 구조

## 1.4 연구 목적

자율주행기술은 운전자 과실로 인한 사고감소, 차량정체 완화 등 많은 이점이 있으며, 연구가 많이 진행되어 왔다. 자동차 전용도로 환경에서는 이미 운보조장치를 적용하여 속도 유지 기능을 사용하고 있다. 기존의 논리 기반 차선변경 판단 방법은 다른 차량과의 충돌 여부를 계산하거나 차선변경 시 후방차량의 속도변화를 계산한다. 하지만 주변의 가까운 차량만 고려하기 때문에 전방의 차량정체에는 대처할 수 없다. 심층강화학습 기반 차선변경 판단 방법은 차량 충돌과 안전거리를 고려하거나 자차의 속도를 고려하여 보상을 설계하고 차선변경 판단을 하였다. 하지만 후방차량에 대한 영향을 고려하지 않았다. 후방차량에 영향을 고려하지 않는다면 후방차량의 급정거로 인한 교통정체를 유발할 수 있다.

본 논문은 주변 차량의 영향을 고려하면서도 전방의 차량정체에 대처할 수 있는 심층강화학습에 의한 자율주행 차량의 차선변경 판단 및 제어 방법을 제안한다. 심층신경망의 입력으로 모든 차량을 사용하고, 보상은 차선변경 시 후방차량의 속도 변화, 자차의 속도로 설정한다. 이 보상 설계를 통해 자차의 속도를 유지하고 차선변경 시 후방차량의 영향을 최소화하는 차선변경 판단을 출력한다. 신경망을 통해 차선변경 판단이 출력되면 안전 모듈로 충돌이 예상되는 차선변경을 방지한다. 최종 차선변경 판단이 되면 베지어 곡선을 사용하여 차선변경 경로를 생성한다. 마지막으로 Pure pursuit 알고리즘으로 횡방향 제어를 하고, IDM(Intelligence Driver Model)알고리즘으로 종방향 제어를 한다.

## 1.5 논문 구성

본 논문은 다음과 같이 심층강화학습에 의한 자율주행 차량의 차선변경 판단 및 제어 방법을 설명한다.

제 I 장에서는 기존의 차량 속도 유지와 차선변경 방법에 관해 설명한다. 그리고 본 연구를 진행하기 위한 배경과 목표를 설명하였다. 기존 논리기반 차선변경 판단 방법과 강화학습 기반 차선변경 판단 방법 연구의 차이점을 설명하였다. 그리고 각 방법의 장점, 단점을 설명하였다.

제 II 장에서는 본 논문에서 제안한 강화학습에 의한 차선변경 판단 알고리즘의 시스템 구성을 상세히 설명한다. 그리고 차선변경 판단 시스템 흐름도에 관해 설명한다.

제 III 장에서는 차선변경 판단을 위해 사용하는 신경망의 구조와 입력, 출력, 보상에 관해 설명한다. 그리고 학습 방법에 대해 상세히 설명한다. 차선변경 충돌 판단을 통해 최종 차선변경 판단을 출력하는 방법에 관해 자세히 설명한다.

제 IV 장에서는 차선변경 경로 생성에 사용한 베지어 곡선을 설명한다. 알고리즘 및 차량 종방향, 횡방향 제어 알고리즘에 관해 설명한다.

제 V 장에서는 본 논문에서 주장하는 차선변경 판단을 적용한 실험에 대한 결과를 제시한다. 시뮬레이션 환경에 관해 설명하고, 평균 자차 속도, 차선 변경 시 후방차량의 속도변화 등 실험 결과를 보여준다.

제 VI 장에서는 본 논문에서 제안한 방법과 실험 결과를 바탕으로 결론을 제시한다. 본 논문의 기여, 문제점과 한계를 서술하고 추후 연구에 대한 목표



를 제시한다.

## II. 시스템 구성

### 2.1 시스템 구성



그림 10. CARLA 시뮬레이터

차선변경 판단 시스템을 실험하기 위해 그림 10과 같은 CARLA 시뮬레이터 [25]를 사용하였다. CARLA는 자율주행 시스템의 개발, 교육 및 검증을 위해 개발된 오픈소스 시뮬레이터이다. 건물, 도로, 차량 등 데이터를 제공하여 이미 만들어진 도시환경을 사용하여 시뮬레이터 환경구성을 쉽게 할 수 있다. 표지판과 같은 정적 객체 그리고 차량, 사람과 같은 동적 객체를 생성하고 제어할 수 있다. 또한 비, 구름, 안개 등 날씨환경도 변경할 수 있다.

시뮬레이터는 제어환경 균일성을 위해 0.05초의 고정 시간 단계를 가진다. 사용한 모든 차량은 BMW의 GT 모델을 사용하였다. 도로 환경은 4차선 자동차 전용 도로를 구성하였으며 자차의 출발 차선은 1차선이다. 출발지부터 목표지점까지의 거리는 1,000m이며 자차가 도달하면 시나리오가 종료된다. 차량은 총 24대를 배치하였으며, 각 차량 간의 거리는 50m 사이이며 2대씩 배치되어있다. 도로 환경은 그림 11에 나타나 있다.

1차선의 주황색 차량은 자차로서 본 논문에서 제안한 차선변경 판단 및 제어 방법을 적용한 차량이다. 자차의 목적은 목표속도를 유지면서 교통흐름을 방해하지 않게 차선변경을 하는 것이다. 그 외 차량은 파란색으로 나타냈으며, 차선변경을 하지 않고 시뮬레이터 내부의 제어기를 사용하여 속도를 제어하였다.

시뮬레이터는 환경정보를 차선변경 판단부로 전달한 뒤 시뮬레이터를 일시 정지 시킨다. 그리고 시뮬레이터에 제어입력을 인가하면 0.05초 동안 제어입력을 실행하고 다음 제어입력을 기다린다.

본 논문에서 제안하는 강화학습에 의한 차선변경 판단 및 제어 시스템의 흐름도는 그림 11와 같다. 시뮬레이터부, 차선변경 판단부, 경로생성부, 제어부로 구성된다.

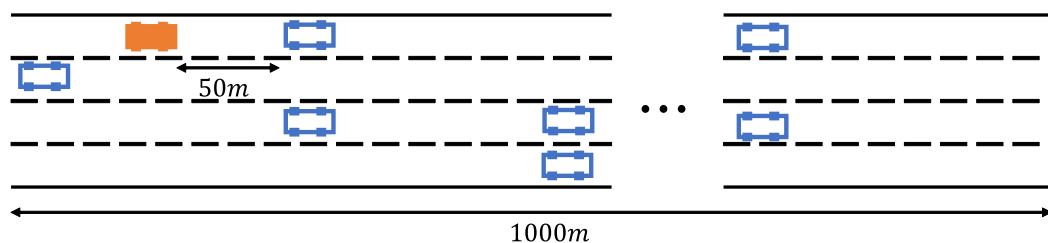


그림 11. 실험을 위한 4차선 자동차 전용도로

시뮬레이터에서는 제어입력을 받아 자차를 제어하고 환경정보와 보상을 차선변경 판단부에 전달한다. 환경정보는 자차의 속도, 가속도, 차선정보와 다른 차량의 상대속도, 상대 가속도, 상대 차선정보, 상대 거리이다. 신경망의 입력으로 사용하기 위해 모든 정보는 0-1로 정규화한다. 강화학습에 사용하는 보상은 자차의 목표속도 도달 비율, 후방 차량의 속도 변화량, 차선변경 여부이다.

차선변경 판단부에서는 신경망의 입력으로 변환된 환경정보를 사용하여 차선변경 판단을 도출한다. 그 후 충돌 검사를 통해 차선변경이 가능하면 차선변경 판단을 최종적으로 출력한다.

경로생성부에서는 차선변경 판단이 결정되었을 때 차선변경 경로를 생성한다. 현재 차량의 속도와 heading방향, 그리고 도로의 방향정보를 사용하여 차선변경 경로를 생성한다. 3차 베지어 곡선을 사용하며 차량의 비홀로노믹 조건을 만족하는 경로를 생성한다.

제어부에서는 경로와 차량정보를 사용하여 횡방향, 종방향 제어입력을 계산한다. 횡방향 제어는 Pure Pursuit 제어 방식을 사용한다. 주어진 경로와 차량의 속도, 자세, 조향각을 이용하여 목표 조향각을 계산한다. 종방향 제어는 전방 차량이 없을 때 PID는 제어를 사용하고, 전방 차량이 있을 때 앞 차량과의 충돌시간을 계산하여 목표 가속도를 계산한다. 계산된 목표 가속도는 PID를 사용해 엔진 Throttle과 브레이크 출력으로 나눈다. 조향각, 스로틀, 브레이크 출력을 시뮬레이터의 자차에 제어입력을 전달한다.

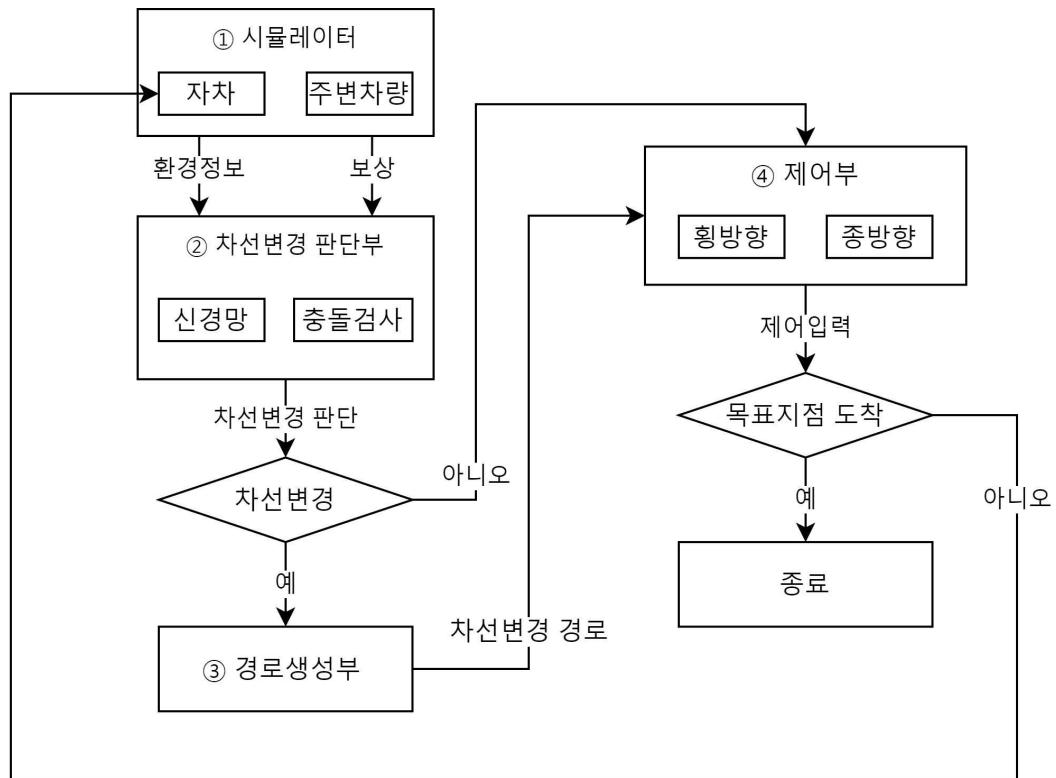


그림 12. 심층강화학습을 사용한 차선변경 판단 및 제어 시스템 구성

### III. 차선변경 판단

#### 3.1 차선변경 판단 구조

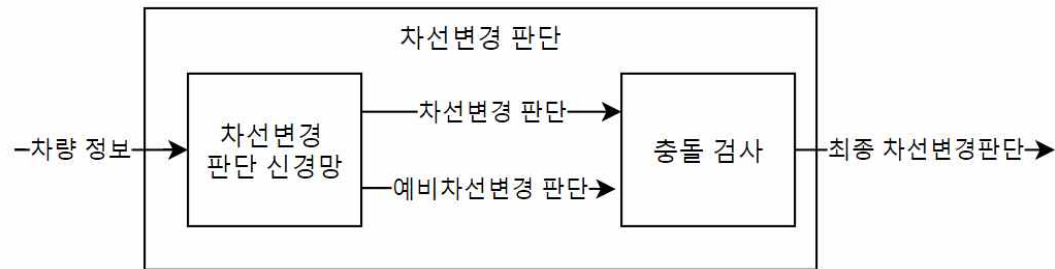


그림 13. 차선변경 판단 구조

차선변경 판단 구조는 그림 13와 같이 구성되어 있다. 시뮬레이터로부터 환경정보를 받으면 신경망의 입력으로 사용하여 차선변경 판단과 예비 차선변경 판단을 출력한다. 차선변경 판단이 도로를 벗어나 실행 불가능하거나 충돌이 예상되면 예비 차선변경 판단을 사용한다. 예비 차선변경 판단도 실행 불가능하면 최종 직진 판단을 내린다. 예비 차선변경 판단을 통해 신경망에서 가장 좋은 행동을 선택하지 못할 때 다음으로 좋은 행동을 선택할 수 있다.

### 3.2 신경망 입출력 설계

전방 차량과 후방차량을 모두 고려하기 위해 모든 차량의 정보를 상태변수로 사용한다. 자차는 속도, 가속도, 차선정보를 사용한다. 자차를 제외한 차량은 상대 속도, 상대 가속도, 상대 거리, 차선정보를 상태변수로 사용한다.

표 2는 신경망에 사용되는 상태변수를 나타내며  $s_1 \sim s_3$ 는 자차의 정보이고,  $s_{4i} \sim s_{4i+3} (i=1,2,...,n)$ 는 나머지 차량의 정보이다.  $v_e$ 와  $v_{\max}$ 는 자차의 속도와 최대 속도,  $a_e$ 와  $a_{\max}$ 는 자차의 가속도와 최대가속도,  $l_e$ 와  $l_{\max}$ 는 자차의 차선정보와 최대차선 개수이다.  $v_t$ 와  $a_t$  그리고  $l_t$ 은 자차를 제외한 각 차량의 속도, 가속도, 차선정보이다.  $s_t$ ,  $s_e$ ,  $s_{\max}$ 는 각각 목표지점과 자차와의 거리, 목표지점과 다른 차량과의 거리, 최대 거리이다. 상태변수를 바로 신경망의 입력으로 사용하면 범위가 큰 변수에 의존할 수 있으므로 모든 상태변수는 0-1으로 정규화한다.

표 2. 신경망 입력

상태변수	값	설명
$s_1$	$(v_{\max} - v_e)/v_{\max}$	자차 속도
$s_2$	$(a_{\max} - a_e)/a_{\max}$	자차 가속도
$s_3$	$l_e/l_{\max}$	현재 차선
$s_{4i}$	$(v_t - v_e)/v_{\max}$	상대 속도
$s_{4i+1}$	$(a_t - a_e)/(2*a_{\max})$	상대 가속도
$s_{4i+2}$	$(l_t - l_e)/l_{\max} + 0.5$	상대 차선
$s_{4i+3}$	$(s_t - s_e)/s_{\max} + 0.5$	상대 거리

차선변경에 직접적인 영향을 미치는 차량은 현재 차선의 전, 후방차량과 목표 차선의 전, 후방차량이다. 속도 유지를 위한 차선변경은 좌측, 우측 두 가지 모두 고려하여 더 좋은 방향을 선택해야 한다. 두 차선 모두 고려하기 위해서는 현재 차선과 양쪽 차선의 전, 후방차량의 정보가 모두 필요하다. 이 차량들을 주변 차량이라고 정의한다. 그림 14에서 진한 주황색 차량은 자차이며 파란색 차량은 주변 차량이다.

일반적으로 학습 시 입력값의 순서나 시간에 의존되지 않게 학습시킨다. 하지만 차선변경 판단은 주변 차량의 정보만으로 가능하다. 주변 차량의 정보를 왼쪽 차선의 전, 후방차량 현재 차선의 전, 후방차량 그리고 오른쪽 차선의 전, 후방차량 순서대로 정렬하여 입력으로 사용한다. 주변 차량이 항상 존재하는 것은 아니기 때문에 빈 차량이 있을 수 있다. 하지만 신경망의 입력은 고정되어 있으므로, 존재하지 않는 차량의 정보는 0으로 대체하여 신경망 입력 크기를 맞춰준다.

전방의 교통흐름을 고려하기 위해서 나머지 차량의 정보가 필요하다. 자차를 제외한 차량의 정보를 별도의 신경망을 통과시켜 교통흐름에 대한 특징을 추출한다. 자차의 정보, 주변 차량의 정보 그리고 나머지 차량의 특징들을 모아 DQN 신경망의 입력으로 사용한다. 신경망 입력은 그림 15에 나타나 있다.

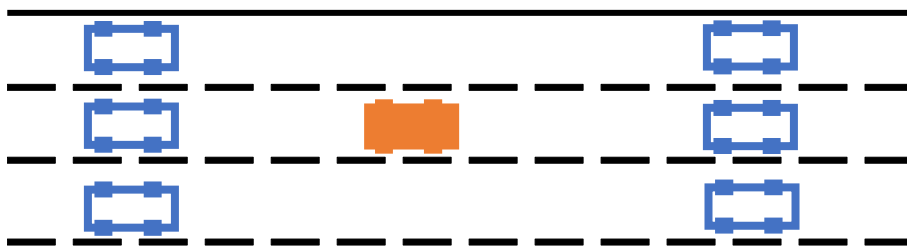


그림 14. 자차와 주변 차량





그림 15. 신경망의 입력

신경망의 출력은 차선변경 판단이며 표 3에 나타나 있다. 신경망의 출력은 3개로 차선유지, 왼쪽 차선변경, 오른쪽 차선변경을 했을 경우의 예상되는 보상이다. 이 보상 값 중 가장 큰 값을 차선변경 판단으로 선택하고, 두 번째로 큰 값을 예비 차선변경 판단으로 선택한다. 첫 번째 차선변경 판단이 불가능할 경우 예비 차선변경 판단을 선택한다. 두 가지 차선변경 판단과 충돌 안전검사를 통해 최종 차선변경 판단을 내린다.

표 3. 신경망 출력

행동	설명
$a_1$	차선 유지
$a_2$	왼쪽 차선변경
$a_3$	오른쪽 차선변경

### 3.3 신경망 구조

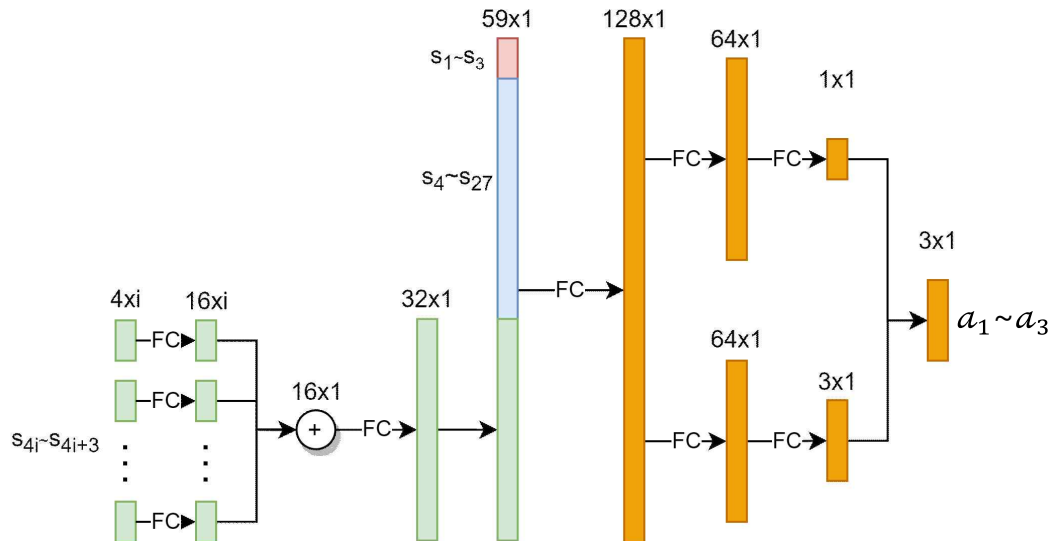


그림 16. 신경망 구조

전체 신경망 구조는 그림 16과 같다. 각 층은 모두 완전히 연결되어 있으며 층 위의 숫자는 뉴런의 개수이다. 자차와 주변 차량의 정보는 직접 DQN의 입력으로 사용한다. 자차를 제외한 차량의 정보는 별도의 신경망을 거친다. 이 신경망은 각 차량의 신경망 출력 결과를 더 해 하나의 특징 벡터를 출력한다. 이 신경망을 통해 차량의 대수와 상관없이 신경망을 사용할 수 있다. 최종 DQN의 입력은 자차, 주변 차량, 자차를 제외한 차량의 특징이다. DQN의 은닉층은 한 개이며 128개의 뉴런으로 이루어져 있다. 행동과 가치를 분리하기 위해 각 64개의 뉴런을 가진 층으로 나눈다. 그 후 완전연결층을 거쳐 차선변경 판단을 출력한다.

### 3.4 학습 방법

기계학습에서는 손실함수를 정의하고 최소화하듯이 강화학습은 보상을 정의하고 최종 누적 보상을 최대화한다. 본 논문에서 사용한 보상은 표 4와 같다.

표 4. 강화학습 보상

보상	설명	값
$r_{speed}$	속도 보상	$\alpha * -(v_{max} - v_e / v_{max})$
$r_{lc}$	차선변경 보상	-1
$r_{fv}$	후방차량 감속도 보상	$\beta * -(v_t - v_{t-l} / v_{t-l})$

$v_e$ 는 자차의 속도,  $v_{max}$ 는 자차의 목표속도,  $v_t$ 는 후방차량의 속도,  $v_{t-l}$ 는 차선변경 후 후방차량의 속도,  $\alpha$ 는 속도 보상계수,  $\beta$ 는 후방차량 감속도 보상계수이다.  $r_{speed}$ 는 자차의 속도를 유지하기 위해 존재하는 보상이며 시간 단계마다 목표속도에 도달한 비율의 값을 준다. 속도가 느릴수록, 시뮬레이션이 길어질수록 속도 보상의 총합이 낮아진다.  $r_{lc}$ 는 차선변경 시 받게 되는 보상으로서 차선변경이 끝났을 때 속도 보상과 함께 받는다. 불필요한 차선변경을 막기 위해 차선변경 보상을 사용하였다. 마지막으로  $r_{fv}$ 는 후방차량의 감속도이다. 이 보상으로 후방 차량을 방해하지 않는 역할을 한다.

### 3.5 충돌 판단

차선변경 충돌 판단 알고리즘은 그림 17과 같으며 다음 순서를 따른다.

S1. 도로를 벗어나는 차선변경을 확인한다.

S2. 차선변경 충돌을 검사한다.

S3. 도로를 벗어나거나 충돌 예상 시 예비 차선변경 판단을 선택한다.

S4. 예비 차선변경 판단에 대해 충돌을 검사한다.

S5. 차선변경 불가능 시 직진을 선택한다.

S6. 직진 불가능 시 예비 차선변경 판단을 선택한다.

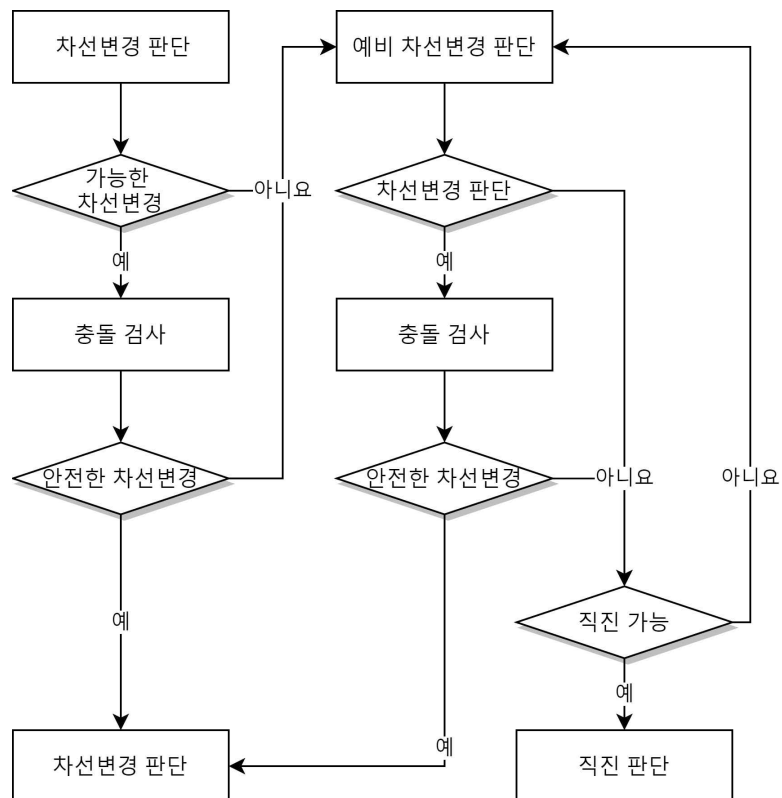


그림 17. 차선변경 판단 알고리즘

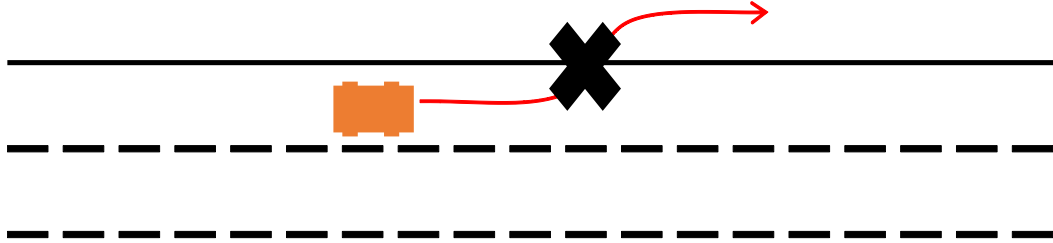


그림 18. 차선변경이 불가능한 경우

신경망의 출력으로 차선변경 판단을 내렸지만 그림 18와 같이 차선변경이 불가능할 수 있다. 도로를 벗어나지 않는지 확인하기 위해 차선정보와 행동을 비교한다. 자차가 왼쪽 끝 차선일 경우 왼쪽 차선변경을 막고, 오른쪽 끝 차선일 경우에는 오른쪽 차선변경을 막는다.

도로를 벗어나지 않는 차선변경이라면 충돌검사로 차선변경 시 차량과 충돌하는 것을 방지한다. 충돌검사는 그림 19와 같이 차선변경이 결정된 시점에서 목표 차선의 앞, 뒤 차량을 고려하여 안전거리를 계산한다. 안전거리는 식 2에 나타나 있다.

$$\begin{aligned} s_f &= v_e + 0.8 * (v_e - v_f) \\ s_l &= v_e + 0.8 * (v_l - v_e) \end{aligned} \quad (2)$$

$s_f$ 는 목표 차선 후방차량과의 안전거리,  $s_l$ 은 목표 차선 전방 차량과의 안전거리,  $v_f, v_e, v_l$ 은 각각 목표 차선 후방차량의 속도, 자차의 속도, 목표 차선 전방 차량의 속도이다. 차량 간의 거리가 안전거리보다 큰 경우 안전한 차선변

경으로 판단한다.

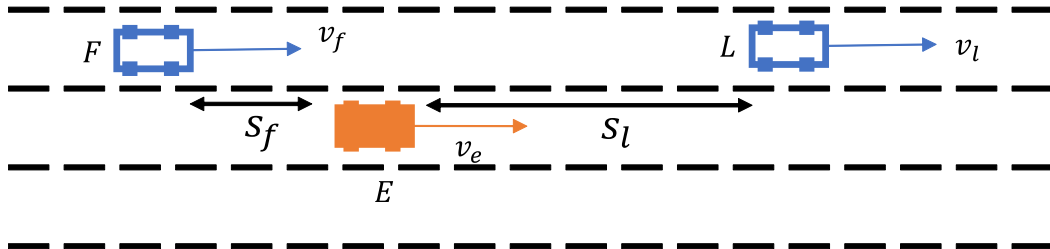


그림 19. 안전거리 계산

도로를 벗어나거나 충돌검사에 실패하는 차선변경이면 예비 차선변경 판단에 대해 다시 검사한다. 안전한 차선변경이 판단되면 최종 차선변경 판단을 출력하고 그렇지 않으면 직진 판단을 출력한다.

직진 판단 때 만약 차선이 사라지거나 전방의 사고 차량이 정지해 있는 등 직진할 수 없을 수가 있다. 이때에는 예비차선변경 판단을 사용하여 차선변경 판단을 내린다.

## IV. 경로계획 및 제어

### 4.1 차선변경 경로 생성

신경망에서 차선변경 판단을 내리면 자차와 주변 차량의 정보를 사용하여 차선변경 경로를 생성한다. 조절점  $P$ 가 4개인 3차 베지어 곡선을 이용하여 차선변경 경로를 생성한다. 베지어 곡선은 차량의 실제 움직임처럼 곡률이 연속이므로 차량제어에 적합한 곡선이다. 3차 베지어 곡선을 좌표계로 나타내면 그림 20과 같다. 하지만 실제 차량은 항상 도로와 평행하게 움직이지 않고, 도로 또한 항상 직선이 아니다. 그러므로 차선변경 경로를 생성할 때 각도를 고려해야 한다. 그림 21은 차량과 도로의 각도를 고려한 차선변경 경로이다.

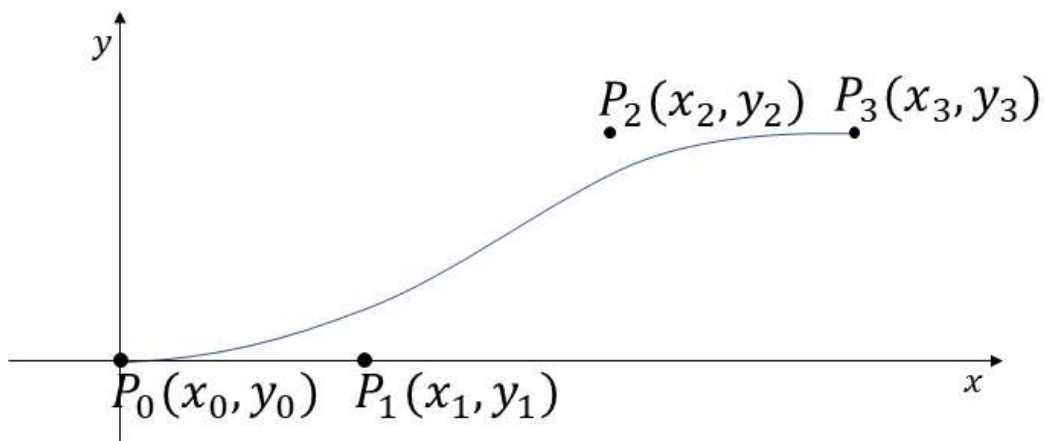


그림 20. 3차 베지어 곡선과 조절점

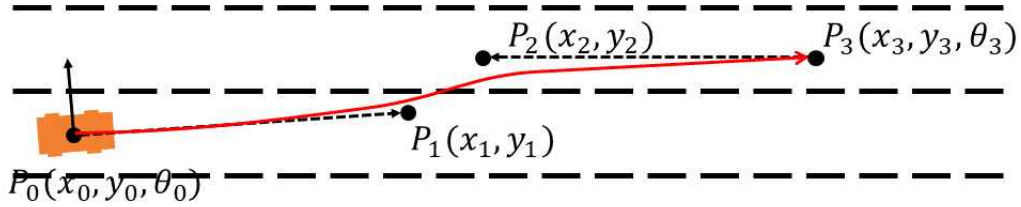


그림 21. 베지어곡선을 사용한 차선변경 경로

점  $P_0$ 는 자차의 위치, 점  $P_3$ 은 차선변경 완료 지점이며 식 4에 나타나 있다.

$$\begin{aligned} s_{lc} &= v_e * t_{lc} \\ s_c &= 0.6 * s_{lc} \end{aligned} \quad (3)$$

$$\begin{aligned} x_3 &= x_0 + s_{lc} \\ y_3 &= y_0 + s_{lw} \end{aligned} \quad (4)$$

$s_{lc}$ 는 차선변경 길이,  $s_c$ 는 조절점 생성을 위한 변수,  $v_e$ 는 자차의 속도,  $t_{lc}$ 는 차선변경 시간,  $d_l$ 은 차폭의 길이이다.

점  $P_2$ 와  $P_3$ 는 차량의 헤딩 방향과 도로의 각도를 이용하여 생성하며 식 5에 나타나 있다.

$$\begin{aligned} x_1 &= x_0 + s_c * \cos\theta_0 \\ y_1 &= y_0 + s_c * \sin\theta_0 \\ x_2 &= x_3 - s_c * \cos\theta_3 \\ y_2 &= y_3 - s_c * \sin\theta_3 \end{aligned} \quad (5)$$



위 식으로 구해진 점  $P_0, P_1, P_2, P_3$ 을 이용하여 식 6과 같이 차선변경 경로를 생성한다.

$$P(\lambda) = (1-\lambda)^3 P_0 + 3\lambda(1-\lambda)^2 P_1 + 3\lambda^2(1-\lambda) P_2 + \lambda^3 P_3 \quad (6)$$

## 4.2 횡방향 제어

횡방향 제어 알고리즘인 Pure Pursuit은 그림 22와 같이 기하학적 모델을 사용한다.

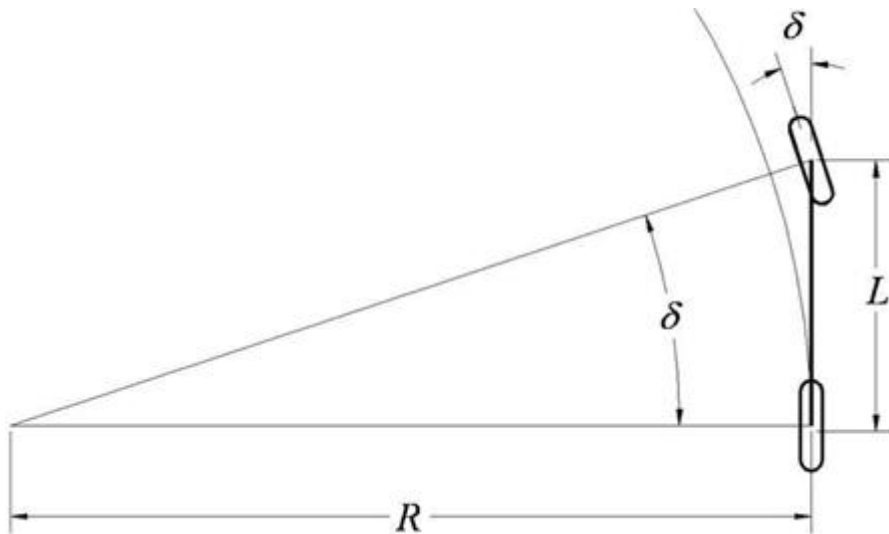


그림 22. 기하학적 모델

$\delta$ 는 앞바퀴의 조향각,  $L$ 은 차량의 앞바퀴와 뒷바퀴 사이의 거리(차간거리)이고,  $R$ 은 앞바퀴의 조향각이 주어졌을 때의 뒷바퀴의 회전 반경이다. 이 기하학적 모델로부터  $\delta, L, R$ 의 관계를 식 7과 같이 나타낼 수 있다.

$$\tan(\delta) = \frac{L}{R} \quad (7)$$

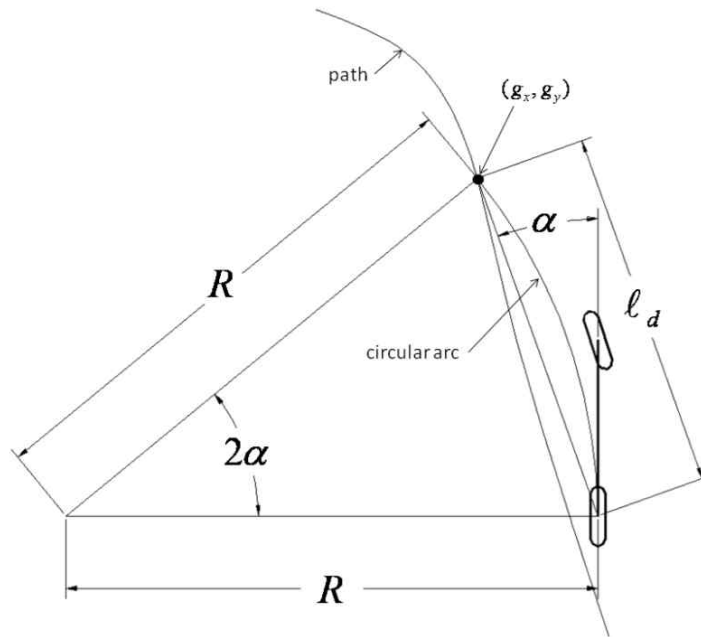


그림 23. Pure Pursuit

Pure pursuit는 그림 23와 같이 경로상의 한 점을 목표지점으로 설정하고 회전 반경을 계산하여 경로를 추종하는 방법이다.  $l_d$ 는 차량 뒷바퀴로부터 목표 지점( $g_x, g_y$ )까지의 거리이다. 일반적으로 차량의 속도에 비례하여 증가시킨다. 조향각을 계산하기 위해 그림 23에서 삼각함수를 적용하면 다음과 같은 식을 얻을 수 있다.

$$\frac{l_d}{\sin(2\alpha)} = \frac{R}{\sin(\frac{\pi}{2} - \alpha)} \quad (8)$$

$$\frac{l_d}{2\sin(\alpha)\cos(\alpha)} = \frac{R}{\cos(\alpha)} \quad (9)$$

$$\frac{l_d}{\sin(\alpha)} = 2R \quad (10)$$

식 7과 식 10을 이용하여 조향각  $\delta$ 를 계산할 수 있다.

$$\delta = \tan^{-1} \left( \frac{2L \sin(\alpha)}{l_d} \right) \quad (11)$$

### 4.3 종방향 제어

종방향 제어는 목표속도 유지와 전방 차량과의 거리 유지를 담당한다. 종방향 제어는 그림 24와 같이 전방 차량이 있는 경우와 없는 경우 두 가지 상황에 따라 제어 방법이 달라진다.

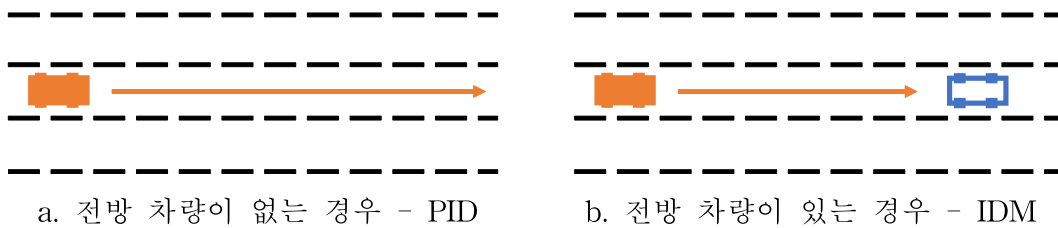


그림 24. 상황에 따른 종방향 제어 방법

본 논문에서는 전방 차량이 없으면 PID제어기를 사용하고 전방 차량이 있을 경우에는 Intelligent Driver Model(IDM) 제어기를 사용한다.

IDM의 가속도 계산 방식은 식 12와 같다.

$$\dot{v} = a \left[ 1 - \left( \frac{v}{v_0} \right)^\delta - \left( \frac{s^*(v, \Delta v)}{s} \right)^2 \right] \quad (12)$$

$$s^*(v, \Delta v) = s_0 + Tv + \frac{v\Delta v}{2\sqrt{ab}} \quad (13)$$

$v_0$ 는 목표속도,  $T$ 는 앞차량의 차두에 뒤 차량의 차두가 도달할 때까지의 시간,  $a$ 는 최대가속도  $b$ 는 설정 감속도,  $\delta$ 는 가속도 계수이며  $s_0$ 는 안전거리다.

## V. 실험

### 5.1 실험 환경

본 실험에서는 정해진 실험환경 대해서 자차의 평균속도, 후방차량의 감속률을 비교하였다. 기존 논리 기반 차선변경 판단 방법인 MOBIL과 본 논문에서 제안한 강화학습기반 차선변경 판단 방법을 비교실험 하였다. 그리고 다른 강화학습 기반 차선변경 판단 방법의 평균속도 성능을 비교하였다.

실험에 사용된 PC의 환경은 Intel i7-6700 CPU를 사용하고 GPU는 Geforce RTX 3090을 사용하였다. 시뮬레이터는 CARLA 0.9.11 버전을 이용하였다. 심층 신경망 학습은 Python을 사용하였다.

실험 환경은 그림 25과 같다. 도로의 길이는 총 1,000m이며 차량 자차 1대, 자차 뒤 5대, 전방 차량 18대로 구성하였다. 각 차량의 종방향 거리는 30m이며 전방 차량은 두 대씩 배치되어있다. 자차의 목표속도는 65km/h이며 나머지 차량의 목표속도는 40km로 설정하였다. 주변 차량의 정보는 센서를 통해 받는 것이 아닌 V2X(Vehicle to Everthing)통신으로 직접 받는다고 가정한다.

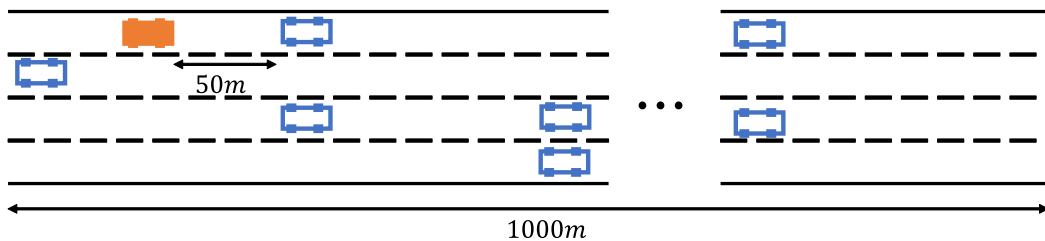


그림 25. 실험 환경

## 5.2 실험 결과

### 5.2.1 차선변경 판단 결과

실험은 각 10번씩 2번 실험하였으며 차량 배치는 실험마다 무작위로 변경하였다. MOBIL의 변수는 논문에 제시된 기본값을 사용하였다. 평균속도는 기존 방법보다 18%증가하였다. 하지만 후방차량 속도변화량의 경우 약 90% 개선되었다. MOBIL 알고리즘은 자차의 속도와 차량의 감속도의 중요도가 같으므로 자차의 가속도가 많이 증가할 수 있으면 뒤 차량의 가속도 감소량도 많이 증가할 수 있다. 기존 방법은 차선변경이 가능한 시점이 되면 바로 차선변경을 하였다. 하지만 제안한 방법은 차선변경이 가능한 구간에서 조금 더 전진하여 후방차량의 영향을 줄여주었다. 기존 방법의 경우 현재 차선과 옆 차선의 차량이 속도가 비슷하면 차선변경을 하지 않는다. 그 문제로 인해 차선변경을 하지 않고 느린 차량을 계속 추종하는 문제가 있다. 본 논문에서 제안한 방법은 차선변경을 더 하여 그 상황을 탈출할 수 있다. 차선변경을 한 번도 하지 않은 경우와 후방차량에 영향을 주지 않은 경우에는 후방차량 감속률을 0으로 설정하였다.

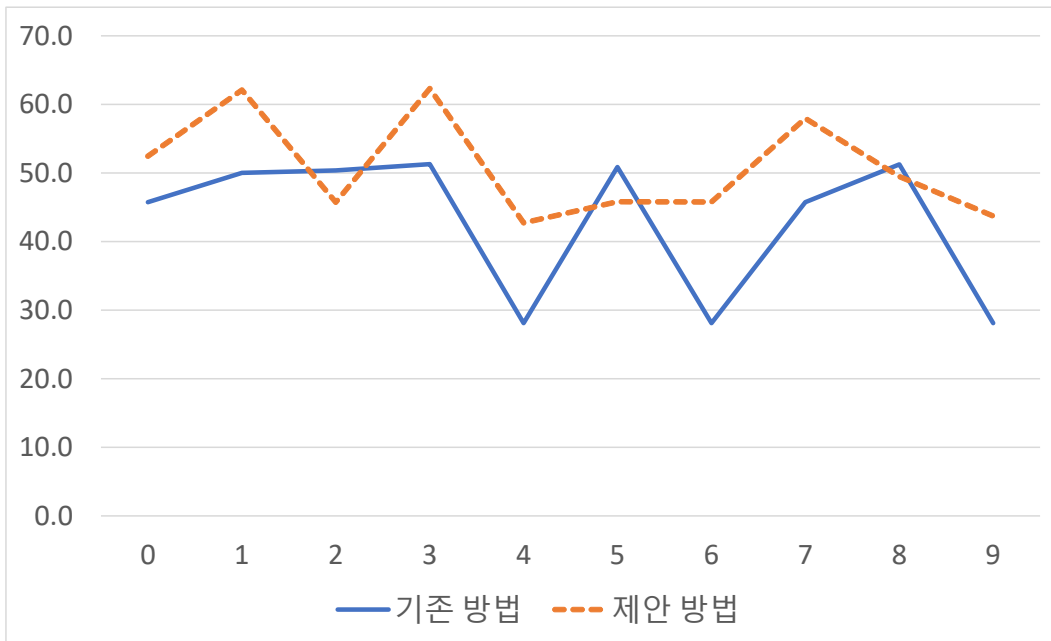


그림 26. 기존 방법(MOBIL)과 제안 방법의 평균속도

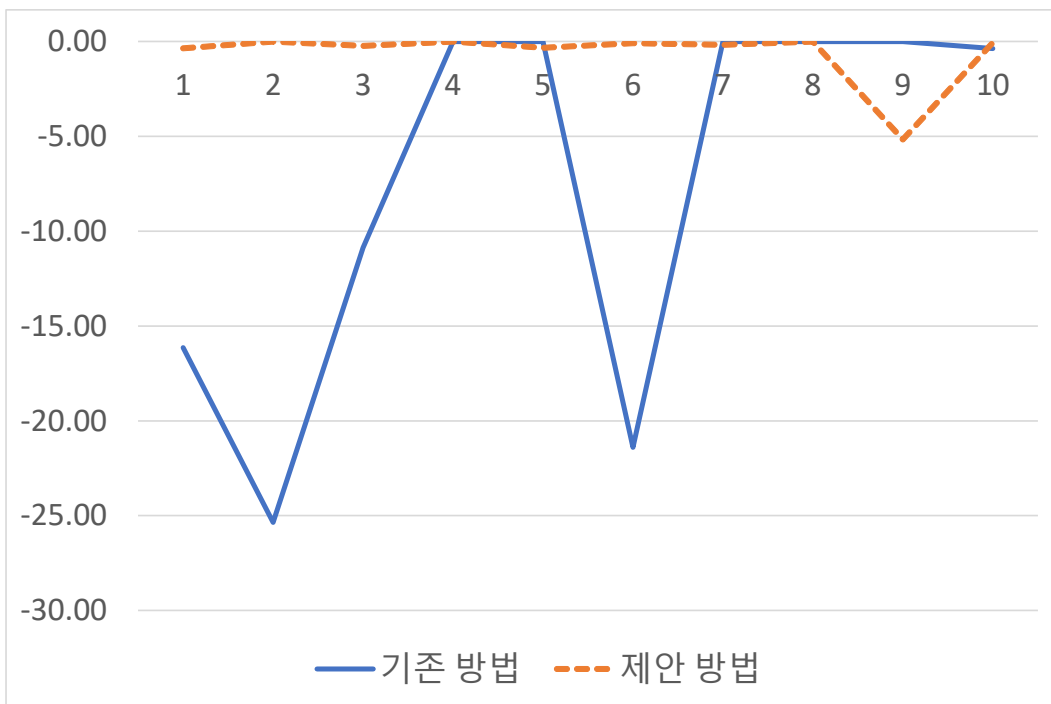


그림 27 기존 방법(MOBIL)과 제안 방법의 후방차량 감속률



표 5. 기존 방법(MOBIL)과 제안방법 비교실험 결과

실험 횟수	기존 방법		제안 방법	
	평균속도 (km/h)	후방차량 감속률 (%)	평균속도 (km/h)	후방차량 감속률 (%)
1	45.7	-16.15	52.5	-0.35
2	50.0	-25.35	62.1	0.00
3	50.4	-10.88	45.8	-0.23
4	51.3	0.00	52.3	0.00
5	28.1	0.00	42.7	-0.32
6	50.8	-21.40	45.8	-0.09
7	28.1	0.00	45.8	-0.17
8	45.7	0.00	58.0	0.00
9	51.3	0.00	49.5	-5.16
10	28.1	-0.36	43.7	-0.06
평균	43.0	-7.41	<b>50.8</b>	<b>-0.64</b>

### 5.2.2 속도 보상계수에 따른 실험

속도유지를 위해 속도 보상을 추가하였다. 강하학습에서 보상의 계수는 중요하다. 보상 자체의 범위뿐만 아니라 다른 보상과의 비율도 고려해야 한다. 차선변경의 가장 큰 목적인 속도 유지를 위한 계수 조정을 위해 속도계수  $\alpha=0.01, 0.05, 0.1$ 일 때 실험하였다.  $\alpha=0.01$ 일 때는 속도에 대한 보상이 낮아서 속도 유지를 할 수 없었다.  $\alpha=0.1$ 일 때는 학습 초기에 평균속도가 가장 빠르다. 하지만 이후에 평균속도가 떨어져 학습이 잘되지 않는 것을 볼 수 있다.  $\alpha=0.05$ 일 때는 평균속도가 완만하게 증가하면서 학습이 잘 되는 것을 볼 수 있다.

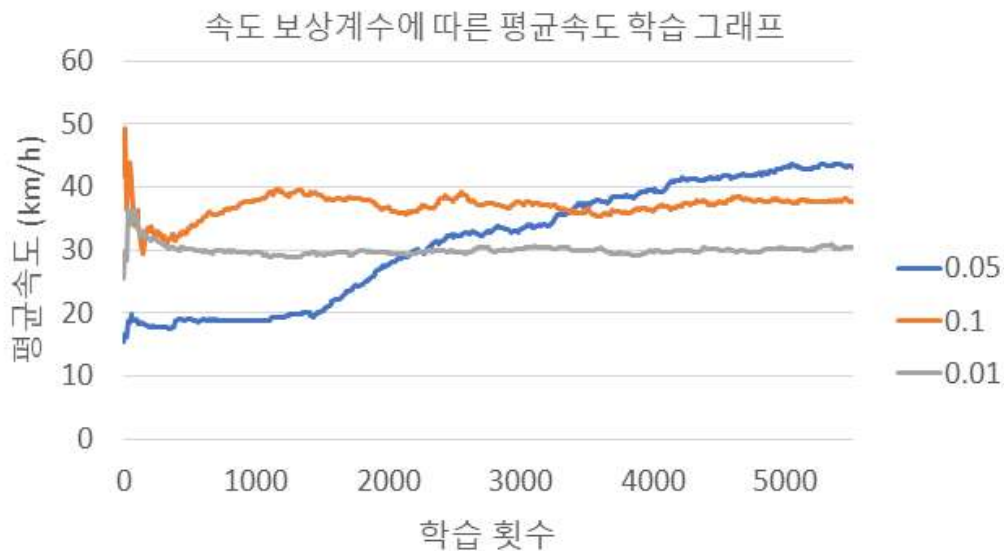


그림 28. 속도보상계수에 따른 평균속도 학습 그래프

후방차량의 교통흐름을 위해 보상으로 후방차량 감속률을 추가하였다. 이 보상의 역할을 검증하기 위해 후방차량 감속률 보상계수  $\beta$ 가 0, 1일 때를 비교하여 실험하였다. 그림 29와 같이 평균속도는  $\beta=0$ 일 때 평균속도 증가량이 가파르다. 하지만 동시에 후방차량 감속률도 증가하는 것을 볼 수 있다. 반면에  $\beta=1$ 일 때 평균속도 학습 속도는 더디지만, 후방차량 감속률도 함께 감소하는 것을 볼 수 있다.

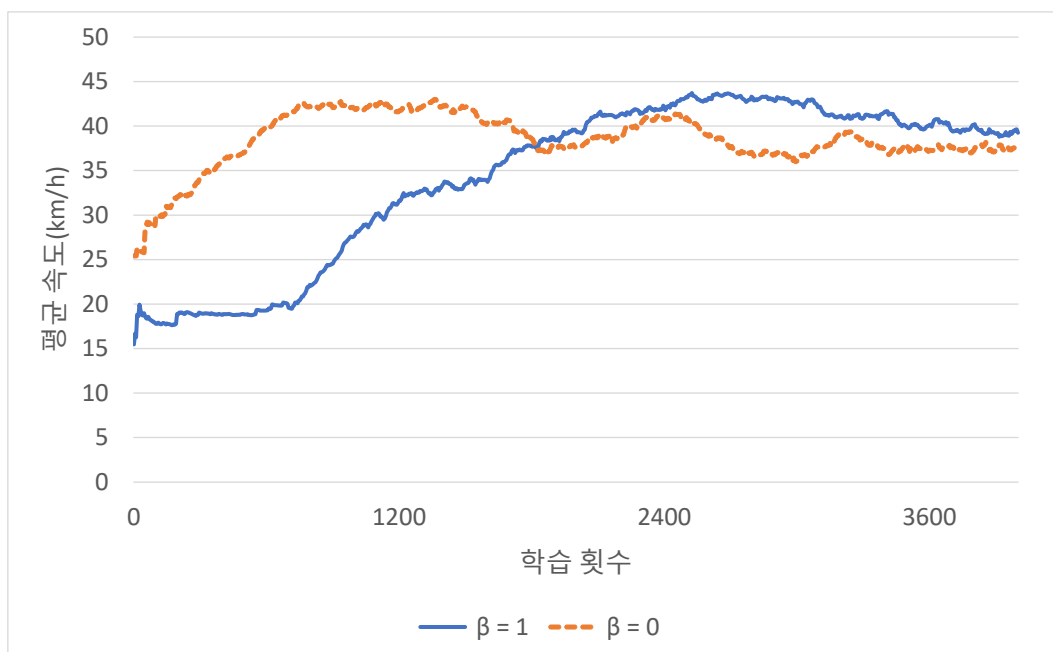


그림 29. 후방차량 감속률 보상계수에 따른 평균속도 학습 그래프

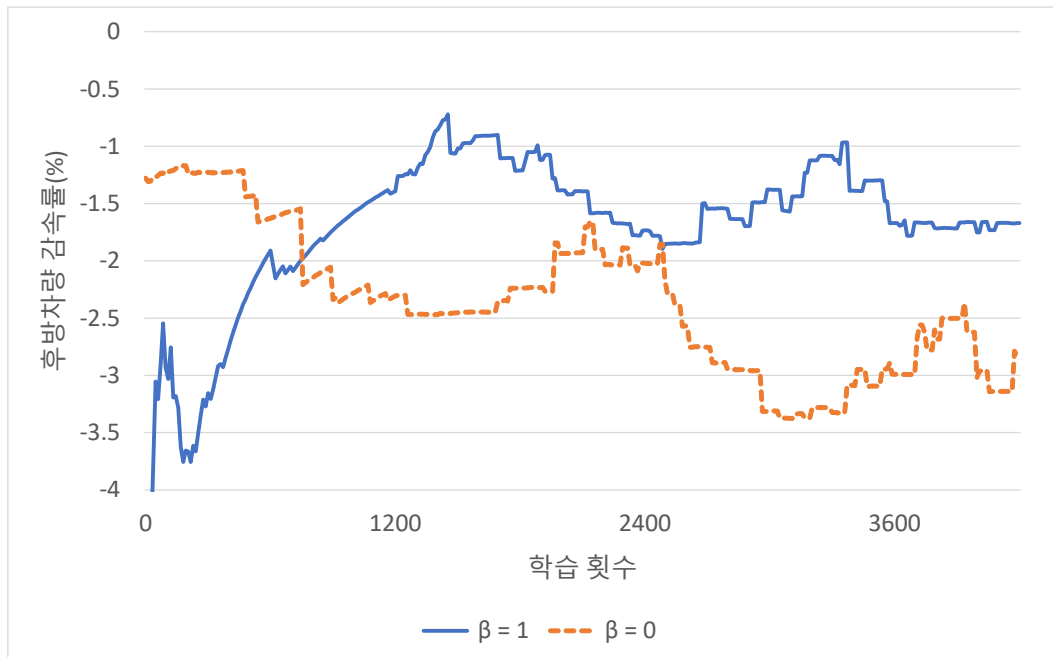


그림 30. 후방차량 감속률 보상계수에 따른 후방차량 감속률 학습 그래프

### 5.2.3 차량 수 변경에 따른 실험

본 논문에서 총 24대의 차량으로 신경망을 학습시켰다. 하지만 실제 도로 환경에서는 차량의 수가 변할 수 있다. 가변 신경망을 사용하여 차량수가 변하더라도 대응할 수 있다. 그림 31와 그림 32는 차량 수의 변화에 따른 평균속도, 후방차량 감속률을 나타낸다. 차량 수가 16대로 줄었을 경우 차량의 수가 줄어 평균속도가 증가하였다. 후방차량 감속률은 증가하였지만 후방차량을 거의 방해하지 않는다. 차량수가 8대로 줄었을 때는 전체 차량수가 많이 줄어 후방차량 감속률과 평균속도 모두 향상되었다. 이 실험에서는 차선변경을 하지 않았을 때 후방차량의 감속률을 N으로 표기하며 평균에 포함하지 않았다.

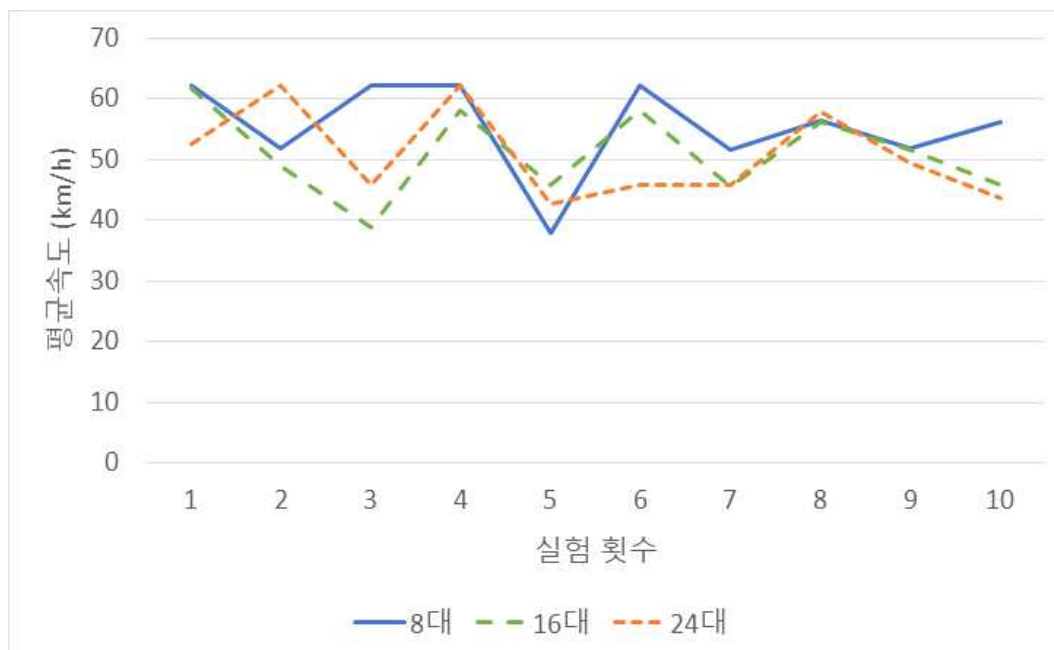


그림 31. 차량 수에 따른 평균속도

표 6. 차량 수에 따른 평균 속도

차량 수	8대	16대	24대
실험 횟수	평균속도 (km/h)	평균속도 (km/h)	평균속도 (km/h)
1	62.3	61.7	52.5
2	51.9	48.9	62.1
3	62.3	38.8	45.8
4	62.3	58.1	62.3
5	37.8	45.8	42.7
6	62.3	58.1	45.8
7	51.7	45.6	45.8
8	56.3	56.2	58.0
9	51.9	51.7	49.5
10	56.2	45.8	43.7
평균	<b>55.5</b>	51.0	50.8

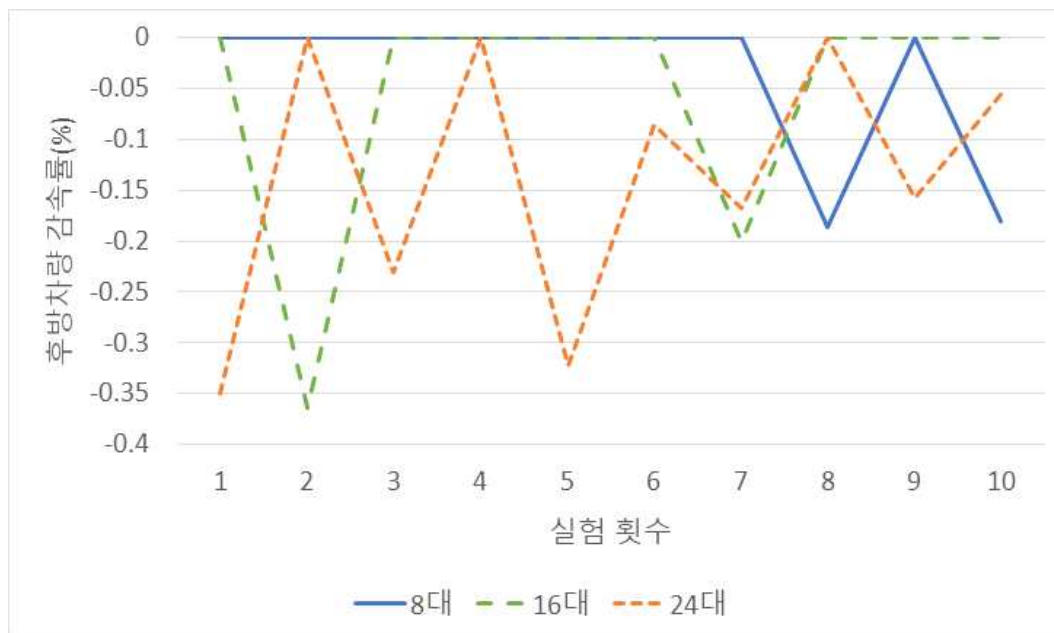


그림 32. 차량 수에 따른 후방차량 감속률

표 7. 차량 수에 따른 후방차량 감속률

차량 수	8대	16대	24대
실험 횟수	후방차량 감속률(%)	후방차량 감속률(%)	후방차량 감속률(%)
1	N	N	-0.35
2	N	-0.364	N
3	N	N	-0.231
4	N	N	N
5	N	N	-0.323
6	N	N	-0.086
7	N	-0.201	-0.167
8	-0.186	N	N
9	N	N	-0.158
10	-0.181	N	-0.056
평균	<b>-0.18</b>	-0.28	-0.20

#### 5.2.4 기존 방법과 성능 비교

본 논문에서 제안한 방법과 기존 논리방법인 MOBIL과 강화학습 기반 방법의 성능을 비교하였다. MOBIL을 기준으로 평균속도와 후방차량 감속률을 표 8에 나타냈다. DQN 기반 강화학습 차선변경은 후방차량 감속률은 측정되지 않았다. 제안한 방법이 기존 방법보다 평균속도가 높은 것을 확인하였다.

표 8. 기존 방법과 제안 방법 비교

	평균속도 (MOBIL 대비 비율)	후방차량 감속률 (MOBIL 대비 비율)
MOBIL	1.0	1.0
DQN [14]	1.1	-
제안 방법	<b>1.18</b>	<b>0.09</b>



## VI. 결론

본 논문에서는 고속도로에서 속도 유지와 차선변경 시 교통흐름 개선을 위해 심층강화학습을 이용하여 차선변경 판단을 내리고 차량을 제어하는 방법을 제시하였다.

기존의 차선변경 방법은 후방차량의 속도를 고려하지만, 주변 차량만 계산하기 때문에 연속된 차량을 고려하기 어려운 단점이 있다. 심층강화학습을 사용하여 후방차량의 교통흐름을 고려하면서도 전방 차량을 고려하는 차선변경 판단 방법을 제안하였다. 본 논문에서 제안한 방법은 기존의 차선변경 판단 방법보다 평균속도가 증가하였다. 또한 시뮬레이션을 통해 차선변경 시 후방 차량의 감속률이 감소하는 것을 확인하였다.

그러나 기존 신경망으로 학습한 차량의 수와 입력 차량의 수가 달라진다면 신경망 성능이 저하될 수 있다. 그리고 제한된 시나리오에서 실험하였기에 실제 환경에서 적용하기 힘들다. 추후 충돌까지 모두 강화학습으로 판단할 수 있는 신경망을 설계하고, 단순한 차량 배치가 아닌 복잡한 환경에서도 적용 가능한 차선변경 판단이 가능한 신경망을 연구할 예정이다.

## VII. 참고 문헌

- [1] S. W Jung, Y. J. Moon, "Impacts of automated vehicles on traffic flow changes", The Journal of The Korea Institute of Intelligent Transport Systems vol. 16, no. 6, pp. 244-257, 2017.
- [2] SAE, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles", Apr 2021.
- [3] M. Endsley, "Autonomous driving systems: A preliminary naturalistic study of the Tesla Model S," pp. 225-238, 2017.
- [4] H. Julia, E. B. Kosmatopoulos and P. A. Ioannou, "Collision avoidance analysis for lane changing and merging," IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2295-2308, Nov. 2000.
- [5] J. Chen, P. Zhao, T. Mei and H. Liang, "Lane change path planning based on piecewise bezier curve for autonomous vehicle," IEEE International Conference on Vehicular Electronics and Safety, pp. 17-22, 2013.
- [6] A. Kesting, T. Martin, and H. Dirk. "General lane-changing model

- MOBIL for car-following models.” Transportation Research Record 1999, no. 1, pp. 86 - 94. Jan 2007.
- [7] M. S. Kim, and T. H Park, “Vehicle overtaking algorithm considering speed information of front and side vehicles”, Institute of Control, Robotics and Systems pp. 9-10, 2020.
- [8] J. Nilsson, M. Brännström, E. Coelingh and J. Fredriksson, "Lane change maneuvers for automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 18, no. 5, pp. 1087-1096, May 2017.
- [9] T. Y. Oh, W. I. Son, T. W. Ahn, Y. K. Lee, K. H. Park, “Development of automated lane change algorithm considering safety of surrounding vehicles” Transaction of the Korean Society of Automotive Engineers vol.. 29, no. 5, pp. 391-405, May. 2021.
- [10] P. Wang, C. Chan and A. de La Fortelle, "A reinforcement learning based approach for automated lane change maneuvers," IEEE Intelligent Vehicles Symposium (IV), pp. 1379-1384, Jun 2018
- [11] H. G. An, J. I. Jung “Decision-making system for lane change using

- deep reinforcement learning in connected and automated driving," Electronics, vol 8, no. 5, pp. 543. May 2019.
- [12] D. Li, D. Zhao and Q. Zhang, "Reinforcement Learning based Lane Change Decision-Making with Imaginary Sampling," IEEE Symposium Series on Computational Intelligence (SSCI), pp. 16–21, Dec 2019.
- [13] T. Shi, P. Wang, X. Cheng, C. Chan and D. Huang, "Driving Decision and Control for Automated Lane Change Behavior based on Deep Reinforcement Learning," IEEE Intelligent Transportation Systems Conference (ITSC), pp. 2895–2900, 2019.
- [15] F. Ye, X. Cheng, P. Wang, C. -Y. Chan and J. Zhang, "Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning," IEEE Intelligent Vehicles Symposium (IV), pp. 1746–1752, 2020.
- [16] C. Hoel, K. Driggs-Campbell, K. Wolff, L. Laine and M. J. Kochenderfer, "Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving," IEEE Transactions on Intelligent Vehicles, vol. 5, no. 2, pp. 294–305, Jun 2020.

- [17] J. Wang, Q. Zhang, D. Zhao and Y. Chen, "Lane Change Decision-Making through Deep Reinforcement Learning with Rule-based Constraints," International Joint Conference on Neural Networks (IJCNN), pp. 1-6, Jul 2019.
- [18] S. Jiang, J. Chen and M. Shen, "An Interactive Lane Change Decision Making Model with Deep Reinforcement Learning," International Conference on Control, Mechatronics and Automation (ICCMA), pp. 370-376, Nov 2019.
- [19] M. H. Song, C.I. Kim, "Study on the Path Planning Method for Ride Comfort Improvement of Autonomous Vehicle during Automatic Lane Change", The Institute of Electronics and Information Engineers, Nov 2016.
- [20] W. Wang, T. Hsu and T. Wu, "The Improved Pure Pursuit Algorithm for Autonomous Driving Advanced System," IEEE 10th International Workshop on Computational Intelligence and Applications (IWCIA), pp. 33-38, 2017.
- [21] G. M. Hoffmann, C. J. Tomlin, M. Montemerlo and S. Thrun, "Autonomous Automobile Trajectory Tracking for Off-road Driving:

Controller Design, Experimental Validation and Racing," American Control Conference, pp. 2296–2301, 2007.

[22] V. Mnih, K. Kavukcuoglu, D. Silver. et al. "Human-level Control through Deep Reinforcement Learning." *Nature* 518, pp. 529 - 533, Feb 2015.

[23] C. J. Watkins and P. Dayan, "Q-learning", *Machine learning*, vol. 8, no. 3-4, pp. 279-292, 1992.

[24] M. Huegle, G. Kalweit, B. Mirchevska, M. Werling and J. Boedecker, "Dynamic Input for Deep Reinforcement Learning in Autonomous Driving," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 7566–7573, 2019.

[25] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, V. "CARLA: An Open Urban Driving Simulator," *Proceedings of the 1st Annual Conference on Robot Learning*, Nov 2017.

## 감사의 글

먼저, 대학원 석사과정 2년 동안 저를 지도해주신 박태형 교수님께 감사드립니다. 교수님 지도 아래에서 자율주행차에 대한 전반적인 지식과 제어 그리고 강화학습에 대해 알게 되었습니다. 그리고 자율주행차대회를 나갈 기회를 주셔서 감사합니다. 또한 물심양면으로 지원해주셔서 대회에서 좋은 결과를 얻을 수 있었습니다. 처음에는 몰랐던 자율주행 자동차 제어에 익숙해지고 강화학습을 공부할 수 있어 좋았습니다.

함께 연구실에서 생활했던 이재설 형, 김영규 형, 권순섭 형, 김택림 형, 하민호 형에게도 감사를 표합니다. 그리고 자율주행자동차 대회에 같이 참가했던 형택이와 전혁이에게 정말 고생했다고 전하고 싶습니다. 그리고 다시 공부를 하고 싶어 석사를 오신 준경이형, 전기차로 자율주행 제어를 같이하고 강화학습에 도움을 준 민성이, 아직 첫 학기이지만 바쁜 규석이형, 마지막으로 저희 연구실의 서류를 담당하시는 문지혜 선생님에게도 고마움을 표합니다. 또한 저에게 가르침을 주신 전자정보대학 교수님과 졸업을 위해 다양한 서류처리를 맡아주신 김다현 조교님, 이현령 조교님 및 다른조교님께 감사의 말씀을 드립니다.

마지막으로 항상 저를 아껴주시고 대학원 진학을 지지하고 지원해주신 부모님에게 진심으로 감사의 인사를 드립니다. 부모님의 지원 덕분에 석사를 무사히 마칠 수 있었습니다.