Transactions of KSAE, Vol. 32, No. 1, pp.137-148 (January, 2024)

pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2024.32.1.137

적대적 생성 모방학습 기반 종방향 운전자 모델에 관한 연구

이 승 연[1] · 이 형 철[*2]

한양대학교 전기공학과[1] · 한양대학교 전기생체공학부[2]

A Study on Longitudinal Driver Model Based on Generative Adversarial Imitation Learning

Seungyeon Lee[1] · Hyeongcheol Lee[*2]

[1]*Department of Electrical Engineering, Hanyang University, Seoul 04763, Korea*
[2]*Division of Electrical and Biomedical Engineering, Hanyang University, Seoul 04763, Korea*
(Received 23 October 2023 / Revised 27 November 2023 / Accepted 27 November 2023)

Abstract : With recent improvements in AI technology, the application of artificial intelligence is being attempted in various research area. It is being used in the development of driver model or control design of autonomous vehicle. Especially, study on reinforcement learning or imitation learning algorithm is being actively researched. Imitation Learning is algorithm for mimicking given expert's trajectory. Behavioral Cloning(BC), Dataset Aggregation(DAgger) and Inverse Reinforcement Learning(IRL) are kind of most known imitation learning method. In this paper, we propose an algorithm to develop human-like longitudinal driver model by using Generative Adversarial Imitation Learning(GAIL), which is type of Inverse Reinforcement Learning algorithm. Soft Actor Critic(SAC) RL algorithm is applied for interaction with longitudinal driving environment. Human driver's driving data is obtained from Driver In the Loop Simlation environment by using expert trajectory for GAIL agent. Train result is compared between PI controller based model and Intelligent Driver Model(IDM) result. GAIL-based longitudinal driver model can generate more human-like velocity profile better than other methods.

Key words : Vehicle simulation(차량 시뮬레이션), Inverse reinforcement learning(역강화학습), Generative adversarial imitation learning(적대적 생성 모방학습), Driver model(운전자 모델), Artificial intelligence(인공지능)

Nomenclature

a_{IDM} : idm acceleration, m/s^2
b : confort deacceleration, m/s^2
T : time headway, s
v : vehicle velocity, m/s
v_{des} : desired velocity, m/s
δ : acceleration exponent
s_0 : minimum safe distance, m
v_{ego} : ego vehicle velocity, m/s
v_{lead} : lead vehicle velocity, m/s
a_{ego} : ego vehicle acceleration, m/s^2
d_{rel} : relative distance, m
a_{actor} : actor network acceleration, m/s^2

Subscripts

N : time step
α : temperature coefficient
π_E : expert policy
π : policy
PI : proportional-integral

*Corresponding author, E-mail: hclee@hanyang.ac.kr

1. 서 론

최근 AI 기술이 발전함에 따라 인공지능을 로봇, 드론, 차량 제어 및 자율주행 자동차 등 다양한 분야에 적용이 시도되고 있다.[1,2] 그 중에서도 모방학습(Imitation learning)은 입력으로 주어진 전문가 행동을 모방할 수 있도록 에이전트가 정책을 학습하는 알고리즘을 의미하는데, 강화학습(RL: Reinforcement Learning)과 다르게 특정 대상을 모방하도록 하는 알고리즘의 특성을 이용한 연구들이 진행되고 있다.[3] 모방학습의 종류에는 가장 간단한 지도학습 AI를 적용한 방법인 행동 복제(Behavioral cloning), 에이전트가 잘못된 행동을 하는 경우 전문가의 데이터로부터 더 적절한 행동을 찾아내도록 하는 알고리즘인 DAgger(Dataset Aggregation), 그리고 강화학습 알고리즘을 적용하여 전문가로부터 주어진 행동에 암시된 전문가의 정책 π_E를 찾아내는 방법인 역강화학습(IRL: Inverse Reinforcement Learning)등이 있다.

본 연구에서는 실제 운전자의 운전 데이터를 모방하여 종방향 가감속 판단을 내리는 운전자 모델을 개발하여 실제 운전자의 주행을 모사한 속도 프로파일을 개발하였다. 운전자의 주행 특성을 모방하기 위하여 역강화학습 기반 모방학습 기법 중 하나인 적대적 생성 모방학습(GAIL: Generative Adversarial Imitation Learning)을 적용하였으며, 과거 N스텝 동안의 속도, 가속도, 전방 차량과의 상대 거리 등을 입력받아 가속도의 변화량을 출력시키도록 네트워크를 구성하였다. 모방의 기준이 되는 운전자의 운전 데이터는 dSPACE ASM 시뮬레이션 모델을 HILS와 연동하여 드라이빙 휠을 통해 운전자 조작을 통한 시뮬레이션 데이터를 취득하였으며, Python에서 동작하는 Tensorflow, Tensorflow probability를 이용하여 GAIL 학습 알고리즘을 적용하고 OpenAI Gym으로 구현된 환경과 상호작용하며 모방학습이 이루어졌다. 학습된 에이전트를 이용해 만들어진 속도 프로파일을 PI 제어기, IDM(Intelligent Driver Model)를 이용해 생성한 속도 프로파일과 비교하였다.

2. Inverse Reinforcement Leraning

강화학습이란, MDP(Markov Decision Process)로 정의된 문제를 푸는 방법으로, Fig. 1과 같이 에이전트가 환경과 상호작용을 하면서 얻는 보상과 상태를 통해 순차적 의사 결정 문제를 최적화하는 방법이다.

MDP는 일반적으로 $< S, A, R(s,s'), P(s,s'), \gamma >$와 같이 다섯 항으로 이루어진 튜플로 정의되는데, 여기서 S는 관찰 가능한 상태 공간(State space), A는 에이전트의 행동 공간(Action space), $R(s,\ s')$는 상태 s에서 다음 상태 s'로 갈 때 얻을 수 있는 보상(Reward), $P(s,\ s')$는 상태 s에서 다음 상태 s'로 전이할 확률, 마지막으로 γ는 보상의 시점에 대한 가중치를 반영한 할인율(Discount factor)이다. 에이전트는 환경과 상호작용하며 할인율이 고려된 보상의 합을 최대화하는 최적 정책 π를 학습한다.

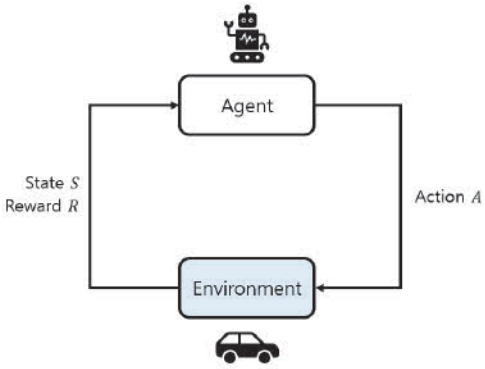

Fig. 1 Reinforcement learning

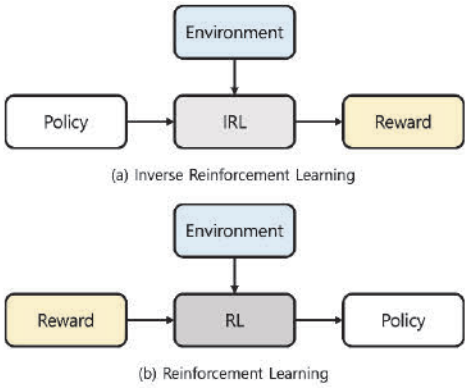

Fig. 2 RL and IRL

역강화학습(Inverse reinforcement learning)은 미리 설계된 보상 함수를 통해 정책 학습이 이루어지는 강화학습과 반대로, 사전에 보상 함수를 따로 정하지 않는 대신 미리 주어진 전문가(Expert)의 행동으로부터 적절한 보상함수를 찾아내고 정책을 학습하는 방식이며, 2000년 Ng와 Russel이 제안한 알고리즘이다.[4] 이 방법은 보상함수를 설계할 필요가 없기 때문에, RL에 비해 설계자가 시행착오를 반복하며 적절한 보상함수를 찾을 필요가 없고, 주어진 데이터가 곧 보상함수가 된다는 장점이 있다. 또한 IRL의 전문가의 행동에 암시된 보상 함수를 찾아내고 정책을 학습하는 특징을 활용하여 모방 학습(Imitation learning)에도 활용된다. 대표적인 모방 학습 방법은 지도학습을 통해 모방하고자 하는 대상의 상태와 해당 상태에서의 행동을 학습하여 대상의 행동을 따라

하도록 하는 행동 복제(Behavioral cloning) 방식이 있는데, 이 방법은 단순한 작업의 경우 학습 속도가 빠르고 효율적이라는 장점이 있지만 복잡한 문제에서 성능이 떨어지며 오차가 누적될 경우 정상적으로 동작하지 못할 수 있다는 문제점이 존재한다. 반면에 IRL은 복잡한 상황이라도 그 안에 숨겨진 보상 요소를 찾아 전문가의 정책을 찾을 수 있다는 장점이 있으며, 오차의 누적에도 비교적 자유롭다. 2.1에서는 IRL 기법 중에서도 적대적 생성 신경망(GAN: Generative Adversarial Network)의 개념을 적용한 GAIL에 대해 소개한다.

2.1 Generative Adversarial Imitation Learning

적대적 생성 모방학습은 대표적인 생성 모델인 GAN의 개념을 적용한 모방학습 알고리즘으로, 2016년 Ho와 Ermon에 의해 제안되었다.[5] 여기서 GAN은 주로 생성 모델에 활용되는 기법으로, 생성자(Generator)와 판별자(Discriminator)로 정의되는 두 개의 신경망이 경쟁적으로 학습을 하는 알고리즘이다. 생성자는 학습이 진행되면서 점점 원본과 유사한 값을 생성하도록 발전하며, 판별자는 생성자가 만들어낸 가짜와 원본을 더 잘 구분하도록 학습이 이루어진다. 학습이 진행될수록 생성자가 만들어낸 값은 모방하고자 하는 대상에 가까워지고, 판별자는 생성자가 만들어낸 값을 더 잘 구분하게 된다.

IRL에 GAN의 원리를 적용한 GAIL은 RL 정책으로 만들어낸 상태-행동 튜플과 전문가의 시연에서 가져온 상태-행동 튜플을 구분하는 판별자를 적용한 모방학습 알고리즘으로, GAIL에서 생성자의 역할을 하는 부분은 환경과 상호작용을 하며 상태-행동 튜플을 만들어내는 강화학습 알고리즘(PPO, TRPO, DDPG, DQN, SAC 등)이 된다.

GAIL을 이용하면 기존의 IRL로는 데이터에 내재된 전문가의 비용함수를 구하고, 구한 비용함수를 통해 다시 강화학습을 하며 Policy π를 찾아내는 과정을 한번에 수행함으로써 시간이 매우 많이 소모된다는 단점을 극복하고 빠른 학습이 가능하다는 장점이 있다.

2.1.1 GAIL의 정의

GAIL을 정의하기에 앞서 RL과 IRL에 대한 정의는 식 (1), (2)와 같이 할 수 있다.

$$RL(c) = \arg\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)] \tag{1}$$

$$IRL_{\psi}(\pi_E) = \arg\max_{\pi\in\Pi} -\psi(c) + \left(\min_{\pi\in\Pi} -H(\pi) + \mathbb{E}_{\pi}[c(s,a)]\right) - \mathbb{E}_{\pi_E}[c(s,a)] \tag{2}$$

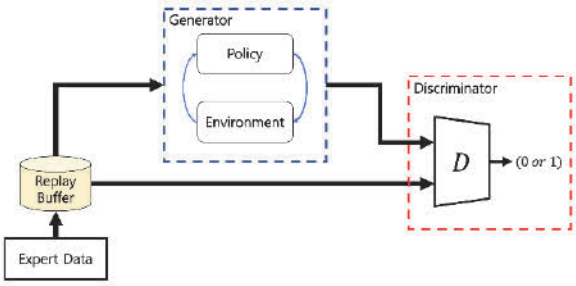

Fig. 3 Generative adversarial imitation learning

GAIL은 RL과 IRL이 동시에 이루어지기 때문에 식 (3)과 같이 표현할 수 있다.

$$RL \circ IRL_{\psi}(\pi_E) = \arg\min_{\pi\in\Pi} -H(\pi) + \psi^*(\rho_{\pi} - \rho_{\pi_E}) \tag{3}$$

모방할 대상의 시연 데이터로서 주어진 전문가의 정책 π_E에 대해 GAIL은 식 (4)의 목적 함수를 가지는 최적화 문제로 정의된다.

$$\min_{\pi} \psi^*_{GA}(\rho_{\pi} - \rho_{\pi_E}) - \lambda H(\pi) = D_{JS}(\rho_{\pi}, \rho_{\pi_E}) - \lambda H(\pi) \tag{4}$$

여기서 $\psi^*_{GA}(\rho_{\pi} - \rho_{\pi_E})$는 두 확률분포 사이의 차이를 나타내는 Jensen-Shannon Divergence $D_{JS}(\rho_{\pi}, \rho_{\pi_E})$이며, GAIL은 전문가의 정책과 Generator의 정책 사이의 JS Divergence를 최소화 시키는 문제가 된다.

2.1.2 Generative Adversarial Imitation from Observation

GAIL은 학습을 위한 데이터로 모방하고자 하는 대상의 상태와 행동에 관련된 데이터가 모두 필요하다. 하지만 실제 데이터를 취득한 환경과 시뮬레이션 환경의 차이 같은 문제로 인해 상태-행동을 그대로 적용할 수 없거나 모방하고자 하는 대상의 상태만 알고 있는 경우 또한 존재하는데, 이 때 적용할 수 있는 방법이 있는데, 이를 IfO(Imitation from Observation)라고 하며 이를 GAIL에 적용한 방법을 GAIfO(Generative Adversarial Imitation from Observation) 라고 한다. 이 방법은 GAIL과 비교했을 때 행동에 대한 정보를 제공하지 않기 때문에 더 적은 차원의 입력을 주더라도 유사한 결과를 얻을 수 있다는 장점이 있다.

일반적인 GAIL의 손실함수는 식 (5)와 같다.

$$-\left(\underbrace{\mathbb{E}_{\tau}[\log(D_{\theta}(s,a))]}_{Agent} + \underbrace{\mathbb{E}_{\tau_E}[\log(1 - D_{\theta}(s,a))]}_{Expert}\right) \tag{5}$$

GAIfO의 손실함수는 식 (6)과 같다.

$$-\left(\underbrace{\mathbb{E}_{\tau}\left[\log\left(D_{\theta}(s,s')\right)\right]}_{Agent}+\underbrace{\mathbb{E}_{\tau_E}\left[\log\left(1-D_{\theta}(s,s')\right)\right]}_{Expert}\right) \quad (6)$$

GAIL에 입력으로 들어가던 $(s,\ a)$ 튜플에서 행동에 해당하는 a를 다음 상태인 s'로 대체하여 $(s,\ s')$를 입력으로 주는 모습을 확인할 수 있다. 판별자 D에 주어지는 입력만 변하기 때문에, 비교적 간단한 수정을 통해 알고리즘을 구현할 수 있다는 장점이 있다.

2.2 Soft Actor Critic

본 절에서는 GAIL에 사용된 정책 알고리즘인 소프트 액터-크리틱(SAC: Soft Actor-Critic)에 대해 소개한다. SAC는 강화학습 알고리즘의 하나로, 2018년 Haarnoja, Zhou등이 제안하였다.[6] SAC는 DQN(Deep Q Learning),[7] DDPG(Deep Deterministic Policy Gradient)[8]와 같은 Off-Policy 강화학습 알고리즘이며, 연속적인 액션 공간에서 좋은 성능을 보인다고 알려져 있다. SAC는 다음과 같은 특징이 있다.

1) Replay Buffer : Off-policy 알고리즘은 과거의 경험을 다시 사용하기 위해 Replay Buffer를 사용한다. 리플레이 버퍼는 에이전트가 과거에 경험한 데이터를 저장하고, 이를 이후 학습 과정에서 무작위로 샘플링해서 재활용한다. 리플레이 버퍼를 적용하면 학습에 사용되는 데이터의 분포를 다양하게 해서 학습 중 일어날 수 있는 과적합을 방지하고 안정적이며 효율적인 학습을 가능하게 한다. 전문가의 시연을 모방하고자 하는 모방학습 알고리즘인 GAIL에서는 리플레이 버퍼에 따라하고자 하는 대상의 시연 데이터를 미리 저장한 후, 이를 학습 과정에서 다시 사용하여 모방학습이 이루어진다.

2) Soft Q learning : Soft Q-Learning 혹은 Soft policy iteration에서는 상태-행동에 대한 가치를 평가하는 데 두 개의 Q 함수를 사용한다. SAC는 학습 중 그래디언트를 계산하는 과정에서 두 Q 함수의 값 중 작은 값을 활용하여 학습이 이루어진다. 각각의 Q 함수는 다른 가중치를 가지고 있으며, 학습 과정에서 낮은 값을 선택함으로써 액터가 다양한 행동을 하면서도 안정적으로 학습이 가능하다는 특징이 있다.

3) Entropy regularization : SAC는 과적합을 방지하고 에이전트가 다양한 행동을 취할 수 있도록 Entropy regularizaton을 적용하였다. 일반적인 강화학습의 목적함수에 Temperature coefficient를 추가하여, 보상의 최대화와 액션의 다양화 사이의 균형을 조절할 수 있게 된다.

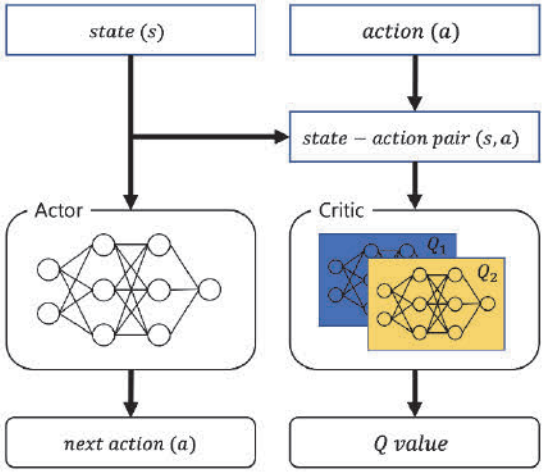

Fig. 4 Structure of soft actor-critic

3. 모방학습 기반 속도 프로파일 생성

기존의 종방향 속도 프로파일을 생성하기 위한 방법으로는 간단하게는 PI 제어기를 이용한 운전자 모델이나,[9] 교통 흐름을 시뮬레이션하기 위해 주로 사용되는 운전자 모델인 Gipps' Model,[10] Intelligent Driver Model(IDM)이 있다. 본 절에서는 기존의 방법보다 인간 운전자에 가까운 속도 프로파일을 생성하기 위해 앞서 설명한 모방학습 기법을 적용하여 종방향 운전자 모델을 학습하고 그 결과를 다른 운전자 모델로 생성한 속도 프로파일과 비교하였다.

3.1 Intelligent Driver Model

본 절에서는 차량 거동을 모델링하기 위한 방법인 Intelligent driver model에 대해 소개한다. IDM은 교통 흐

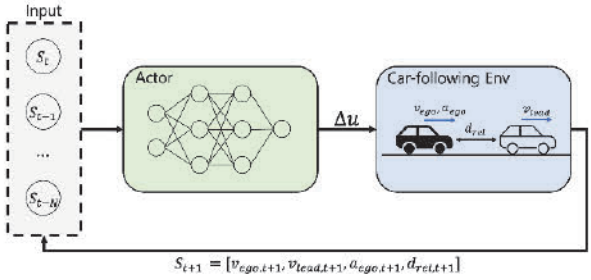

Fig. 5 Imitation learning based longitudinal car following model

름을 모델링하기 위하여 사용되는 대표적인 Car following model의 하나로, 2000년 Treiber, Hennecke 및 Helbing이 제안한 운전자 모델이다.[11] IDM은 고속도로나 도심 교통에서 주변 차량의 자연스러운 거동을 모사하기 위하여 만들어진 운전자 모델로, 차량이 적절한 가속도와 차간거리를 유지한다는 특징이 있다. IDM은 차량의 속도, 선행 차량과의 거리, 차량의 가속도 등을 고려하여 차량의 가속도를 결정한다.

IDM에서는 선행 차량과의 거리가 멀어질수록 속도를 높이고, 선행 차량과의 거리가 가까울수록 속도를 낮추어 차량 간의 거리를 유지하면서 차량의 거동을 모델링하여 보다 자연스러운 교통 흐름을 만들 수 있다. IDM에서 차량의 가속도 a_{IDM}은 식 (7)과 같이 계산할 수 있다.

$$a_{IDM}=a\left[1-\left(\frac{v}{v_{des}}\right)^{\delta}-\left(\frac{s^{*}(v,\Delta v)}{s}\right)^{2}\right]$$
$$s^{*}(v,\Delta v)=s_0+vT+\frac{v\Delta v}{2\sqrt{ab}} \tag{7}$$

여기서 a는 차량의 최대 가속도, b는 Comfortable deacceleration, v는 차량의 현재 속도, v_{des}는 차량의 목표 속도, T는 Safe time headway, s_0은 최소 안전거리, s는 선행 차량과의 상대 거리이다. δ는 Acceleration exponent로, 일반적으로 $\delta=4$가 사용된다.

IDM에서 가속도 a_{IDM}은 Free road에 관련된 항과 Interaction에 관련된 두 항으로 나눌 수 있는데, 각각 식 (9), (10)과 같다.

$$a_{IDM}=a_{free}+a_{inter} \tag{8}$$

$$a_{free}=a\left(1-\left(\frac{v}{v_{des}}\right)^{\delta}\right)$$
$$a_{inter}=-a\left(\frac{s(v^{*},\Delta v)}{s}\right)^{2} \tag{9}$$

Table 1 Parameter set for IDM

Variable	Description	Value
a	Maximum acceleration	2 m/s^2
b	Comfort deacceleration	-1.5 m/s^2
T	Time headway	1.5 s
s_0	Minimum safe distance	10 m
δ	Acceleration exponent	4

$$a_{inter}=-a\left(\frac{s(v^{*},\Delta v)}{s}\right)^{2} \tag{10}$$

전방 차량과의 거리가 매우 멀거나 전방 차량이 없는 경우, a_{inter}에 의한 영향이 사라지게 되며, 차량의 속도가 목표 속도 v_{des}에 가까워지게 된다.

본 논문에서 적용한 IDM의 파라미터는 Table 1과 같다.

3.2 모방학습 기반 운전자 모델

본 절에서는 사람의 운전과 유사한 거동을 하는 종방향 속도 프로파일을 생성하기 위한 모방학습 기반 운전자 모델에 대해 소개한다. 앞서 소개한 IDM을 포함한 종방향 운전자 모델을 만들기 위해 다양한 방법이 제시되어 왔다. 첫 번째로 모델 예측 제어를 이용하여 운전자를 모델링하는 방법이 있다.[12,13] 두 번째는 지도학습 AI를 이용하여 운전자를 모델링하는 방법이 있다.[14,15] 하지만 이 방법은 주로 훈련 데이터셋 범위 내에서는 좋은 성능을 보일 수 있지만, 오차가 누적되거나 훈련되지 않은 데이터가 들어왔을 경우 성능이 떨어진다는 단점이 있으며, 이를 Compunding error라고 한다. 마지막으로 강화학습이나 역강화학습을 이용한 운전자 모델 또한 제안되었다.[16,17] 이 방법은 Nonlinear MPC를 이용한 방법에서 요구되는 차량의 모델링이 필요 없고(Model free), 지도학습 AI의 단점인 Compunding error가 없다는 장점이 있기 때문에, 본 절에서는 앞서 제시한 역강화학습 알고리즘인 GAIL에 더해 GAIL의 변형된 버전인 GAIfO와 SAC를 적용한 종방향 운전자 모델 설계에 대해 소개한다.

3.2.1 Actor Network

모방학습에서 RL 알고리즘인 SAC는 GAN의 생성자에 해당하는 역할을 하면서 대상을 모방하는 상태-행동을 만들어내는 부분으로, 환경과 상호작용하며 전달받은 상태를 입력으로 하여 차량의 가속도를 만들어낸다.

SAC의 액터에 입력으로 주어지는 시간 t에서의 상태 s_t는 (11)과 같다.

$$s_t=[v_{ego,t},\ v_{lead,t},\ a_{ego,t},\ d_{rel,t}] \tag{11}$$

여기서 v_{ego}는 차량의 현재 속도, v_{lead}는 선행 차량의 속도, a_{ego}는 차량의 현재 가속도, d_{rel}은 선행 차량과의 상대 거리이다. Actor network는 각각의 상태 변수에 대해 과거 N개의 시간 샘플 구간의 상태를 입력으로 받아 다음 스텝의 가속도 변화량 Δa를 출력한다.

본 논문에서는 시계열로 들어오는 입력 데이터의 특성

을 반영하기 위하여 입력 데이터를 바로 Fully connected layer에 집어넣지 않고, 1D convolutional neural network를 먼저 적용하여 입력 데이터의 시간적 특성을 추출한 후, 이를 FCN에 적용하여 상태에 따른 액션을 생성하였다.

액터 네트워크의 대략적인 구조는 Fig. 6과 같다.

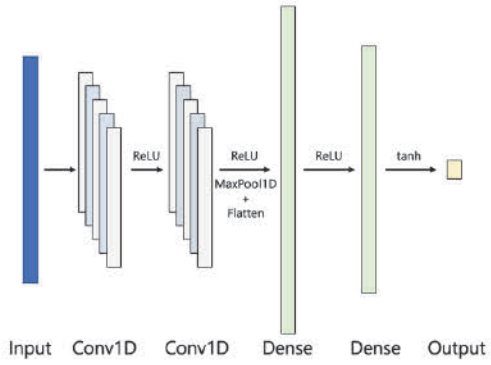

Fig. 6 Scheme of SAC actor network

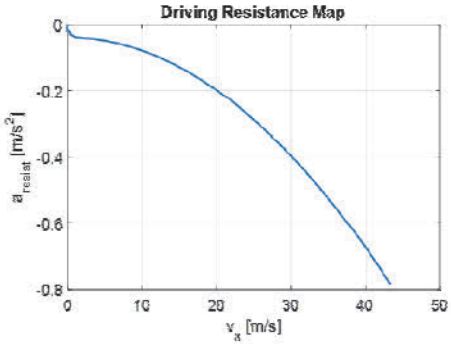

Fig. 7 Driving resistance map

액터에서 생성한 가속도의 변화량 Δa를 이전 스텝의 차량의 가속도에 더해서 차량의 가속도 a_{actor}를 계산한다. 이 때, 차량의 주행 저항에 의한 감속을 고려하기 위하여 계산한 가속도 입력에 주행 저항에 의한 감속 a_{resist}를 더해준다.

$$a_{actor,t+1} = a_{actor,t} + \Delta a_{actor,t}$$
$$a_{ego} = a_{actor} + a_{resist} \tag{12}$$

차량의 속도에 따른 주행 저항은 Fig. 7의 값을 적용하였으며, 학습에 사용된 하이퍼파라미터는 Table 2의 값을 적용하였다.

3.2.2 Critic Network

SAC의 Critic 네트워크는 액터에서 만들어낸 액션과 현재 상태를 평가하는 행동 가치 함수인 Q 함수로 구성되어 있다. 이 논문에서는 Value network 없이 Q 함수로

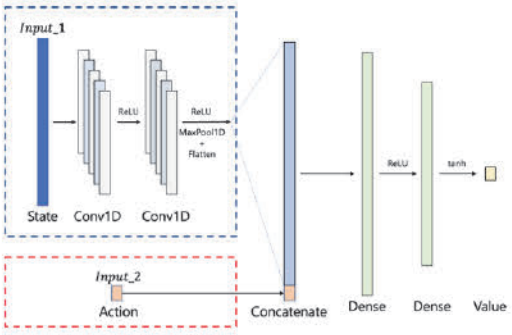

Fig. 8 Scheme of SAC critic network

Table 2 Hyperparameters of actor network

Parameter		Value
Optimizer		Adam
Learning rate		10^{-4}
1D CNN	Number of filters	32
	Dilation	1
	Kernel size	3
	Padding	Causal
	Activation	ReLU
	Pooling	MaxPooling1D
FCN	Number of hidden layer	2
	Units per hidden layer	256
	Hidden activation	ReLU
	Output activation	tanh

Table 3 Hyperparameters of critic network

Parameter		Value
Optimizer		Adam
Learning rate		10^{-4}
1D CNN	Number of filters	32
	Dilation	1
	Kernel size	3
	Padding	Causal
	Activation	ReLU
	Pooling	MaxPooling1D
FCN	Number of hidden layer	2
	Units per hidden layer	128
	Hidden activation	ReLU
	Output activation	Linear

만 구성된 형태의 SAC알고리즘을 사용하였으며,[18] Q_{θ_1}, Q_{θ_2}와 함께 Target-Q Network인 $Q_{\overline{\theta_1}}$, $Q_{\overline{\theta_2}}$가 있어야 하기 때문에, 총 4개의 동일한 구조의 Q Network를 사용하였다. 네트워크 구조는 Actor와 동일하게 과거 N개의 상태에 대한 시계열 입력을 1D CNN에 적용하였으며, 1D CNN의 출력과 액션을 Concatenate하여 FCN에 입력으로 주었다.

Critic 네트워크의 구조는 Fig. 8과 같으며, 학습에 사용된 하이퍼파라미터는 Table 3의 값을 적용하였다.

3.2.3 Discriminator Network

GAIL의 판별자(Discriminator)는 SAC 정책으로 만들어진 값과 모방하고자 하는 실제 운전자의 궤적을 구별하는 역할을 하며, 0(거짓) 또는 1(참)의 값을 출력한다.

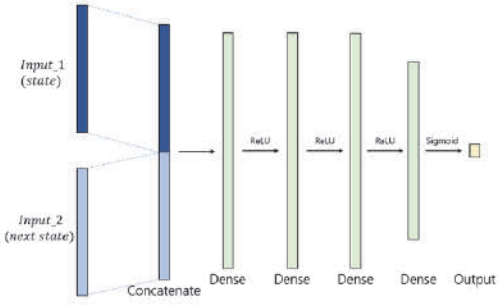

Fig. 9 Scheme of IfO discriminator network

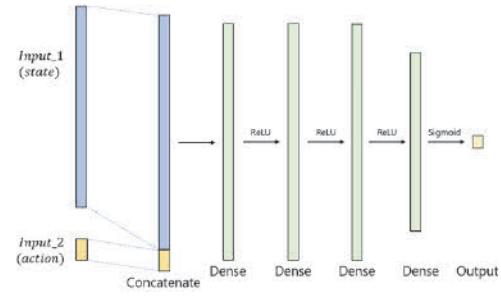

Fig. 10 Scheme of IL discriminator network

Table 4 Hyperparameters of discriminator network

Parameter	Value
Optimizer	Adam
Learning rate	10^{-4}
Number of hidden layers	5
Units per hidden layer	256
Hidden activation	ReLU
Output activation	Sigmoid

Table 5 Hyperparameters of proposed algorithm

Parameter	Value
Max steps per episode	$2.5 \cdot 10^4$
Maximum episodes	100
Maximum steps	$2.5 \cdot 10^6$
Size of replay buffer	10^6
SAC Batch size	32
SAC temperature coefficient	0.2
GAIL/GAIfO Batch size	128

이 때 두 값을 구분하기 위해 줄 수 있는 입력은 두 가지가 있는데, 첫 번째는 GAIL과 같이 상태-행동 튜플을 입력으로 하는 방법이 있으며, 두 번째는 액션을 제외한 상태만을 입력으로 하여 비교하는 방법이 있다. 이를 IfO(Imitation from Observation)라고 하며, GAIL에 IfO를 적용한 알고리즘을 GAIfO(Generative Adversarial Imitation from Observation)라고 한다.[19]

IfO와 IL 네트워크의 구조는 각각 Fig. 9, Fig. 10과 같으며 하이퍼파라미터는 Table 4의 값을 적용하였다.

그 밖의 알고리즘에 적용된 하이퍼파라미터는 Table 5와 같다.

3.3 대상 경로 및 시나리오

본 절에서는 앞서 소개한 방법들을 적용하기 위한 실도로 모사 전방 차량 주행 시나리오에 대해 소개한다. 본 논문에서는 실제 도심 주행 환경을 모사하기 위해 실제 지도상에 존재하는 경로를 선정하고, 이를 도로 모델로 만들어 시뮬레이션에 활용하였다. 선정한 경로에 대해 실제 교통량을 반영하기 위하여 표준 링크번호를 이용하여 주행경로에 대한 교통정보를 가져오고, 이를 적용하여 실제 교통량을 적용한 주행 시나리오를 생성하였다. 또한, 주행 경로 상의 교통 신호를 적용하여 교통 신호에 의한 차량의 정차 및 출발 또한 구현하였다.

시뮬레이션은 MATLAB/Simulink 기반 차량 시뮬레이션 소프트웨어인 dSPACE사의 ASM(Automotive Simulation

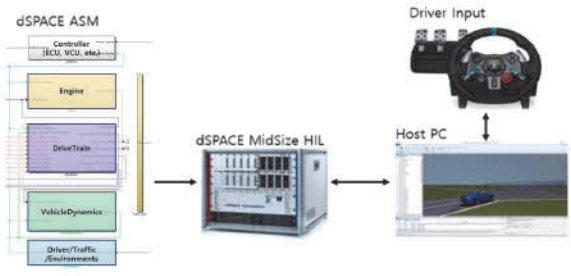

Fig. 11 Driver-HILS environment

Model)을 활용하였으며, 학습에 활용된 실제 운전자의 데이터 취득은 HIL 시뮬레이터에 드라이빙 휠을 추가하여 실제 운전자의 입력에 의한 Driver-HILS 환경을 구축하여 취득하였다.

3.3.1 대상 경로

본 논문에서 사용할 시나리오를 만들기 위해, 실제 도심 경로를 선정하여 시뮬레이션 경로를 적용하였다. 또한 경로상에 존재하는 교통 신호에 의한 영향을 반영하기 위해, 실제 신호등의 실측 주기를 적용한 신호등을 도로에 배치하여 교통 신호에 의한 정차를 구현하였다.

Fig. 12 Target road

3.3.2 실시간 소통정보 적용

실시간으로 변하는 교통상황을 반영하기 위하여 서울교통정보센터 교통정보 시스템(TOPIS)에서 제공하는 실시간 도로 소통 정보를 이용하여 대상 경로에 대한 교통정보를 취득하였다.[20] 소통정보는 5분 단위로 갱신되어 제공되며, 국토교통부 제공 국내 도로의 표준노드링크 ID를 API에 제공하면 해당 구간의 평균 속도와 여행시간에 대한 정보를 반환한다.

대상 경로에 대해 교통 신호 및 실시간 소통정보가 반영된 종방향 속도 프로파일은 Fig. 13과 같다.

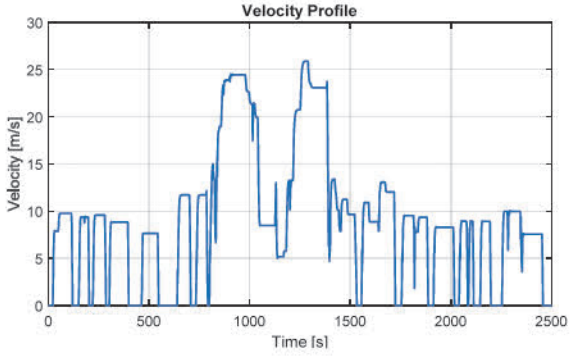

Fig. 13 Velocity profile

4. 학습 결과 및 시뮬레이션

4.1 모방학습 알고리즘 학습 환경

본 논문에서는 Python 기반 TensorFlow 2.6과 TensorFlow Probability를 활용하여 모방학습 알고리즘을 구현하였다. 학습 과정에서 상호작용이 일어나는 환경은 OpenAI Gym을 이용해 만들어졌으며, 매 스텝마다 학습을 진행하였다. GAIL/GAIfO 모방학습 결과와 비교하기 위해 PI 제어기 및 IDM으로 생성된 가상의 주행 시나리오를 비교군으로 선정하였다. PI 제어기, IDM, 모방학습 모델 모두 0.1초의 샘플 시간을 갖도록 알고리즘을 구성하였다.

4.2 학습 결과

본 절에서는 제안한 모방학습 알고리즘의 학습 결과를 확인한다. 실제 학습이 이루어지는 네트워크인 Discriminator, Actor, Critic 네트워크의 Loss, GAIL/GAIfO의 JS Divergence를 확인하였다. GAIL/GAIfO 모두 같은 하이퍼파라미터를 적용하였으며, 100회의 에피소드 동안 학습을 진행시켰다.

학습 결과는 Fig. 14, Fig. 15에서 확인할 수 있다.

학습 결과 두 알고리즘 모두 학습이 진행될수록 안정적으로 수렴하는 모습을 확인할 수 있었으며, 1,500,000 Step 이후에는 두 알고리즘 모두 유의미한 학습의 개선이 이루어지지 않는 것을 확인할 수 있었다. 학습 과정에서는 GAIL에 비해 GAIfO의 경우가 비교적 더 안정적으로 수렴하는 경향이 나타났고, 판별자의 Loss와 JS Divergence의 값도 학습하는 동안 그 변동폭이 적게 변하는 모습을 확인할 수 있었다. 그리고 Actor 네트워크의 Loss 또한 더 작은 값으로 수렴하였다. 그리고 수렴하는 속도 또한 GAIL에 비해 GAIfO가 약간 더 빠르게 수렴하는 것을 확

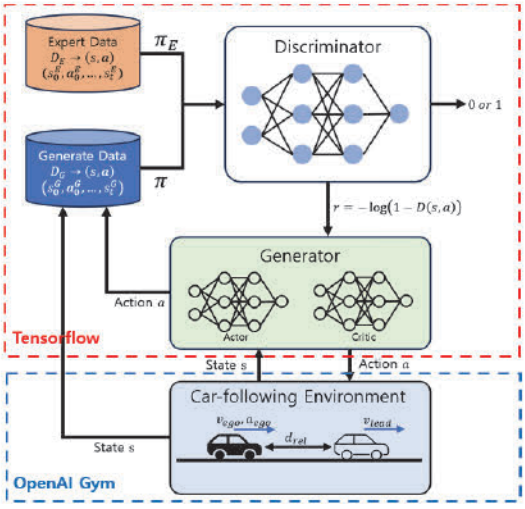

Fig. 14 Training environment

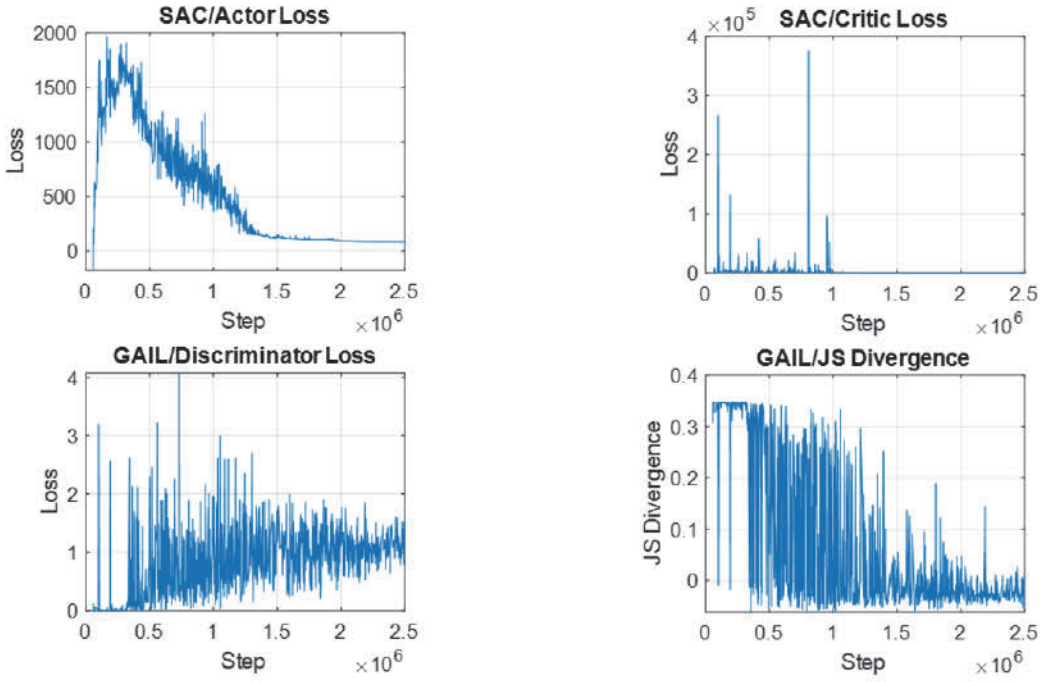

Fig. 15 GAIL train result

인할 수 있다.

하지만, 1,500,000 Step 이후의 결과값은 크게 차이가 나지 않는 것을 확인할 수 있는데, 4.3절에서 최종적으로 학습을 수행한 후 생성한 속도 프로파일의 차이를 비교하였다.

4.3 속도 프로파일 생성 결과

PI 제어기와 IDM, 모방학습을 이용해 3.3에서 만든 종방향 속도 프로파일을 가지는 전방 차량을 추종하는 운전자 모델에 의한 속도 프로파일 생성 결과는 Fig. 16과 같다. PI 제어기를 적용한 경우, 전방 차량의 속도를 잘 추종하는 모습을 확인할 수 있지만, 속도에 따른 차량의 상대거리 등을 고려하지 않으며 실제 운전자의 운전과는 상이한 속도 프로파일을 나타내는 모습을 확인할 수 있었다.

IDM을 적용한 경우에는 차량의 속도에 따라 전방 차량과의 안전거리를 조절하면서 주행하는 모습을 보여주나, PI 제어기와 마찬가지로 실제 운전자보다는 PI 제어기의 출력에 가까운 속도 프로파일이 만들어지는 것을 확인할 수 있었다. 실제 운전자는 일정한 속도를 유지하는 전방 차량을 추종하더라도, 타력 주행으로 적절한 평균 속도를 유지하면서 전방 차량을 따라가는 모습을 확인할 수 있는데, PI 제어기나 IDM의 경우에는 실제 운전자와 달리 목표 속도를 그대로 추종하려 하는 경향을 나타내는 것을 알 수 있다.

조금 더 자세한 결과를 확인하기 위해 생성한 속도 프로파일에 대해 부분적으로 확대한 그래프를 Figs. 20 ~ 22에 표시하였다.

먼저, PI 제어기와 IDM의 경우 상대속도의 차이 없이 전방 차량의 속도와 동일하게 주행하는 모습을 확인할 수 있다. PI 제어기의 경우는 전방 차량의 속도를 거의 동일하게 추종하고 있으며, IDM은 상대 거리를 고려하고 있기 때문에 목표 속도가 전방 차량의 속도가 적절하게 유지되는 선에서 속도를 조절하고 있지만, 사람이 주행했을 때와는 차이가 큰 모습을 확인할 수 있다.

두 번째로, GAIL을 이용해 생성한 속도 프로파일의 경우 PI/IDM보다 실제 운전자가 주행했을 때와 유사한 궤적을 그리며 주행하는 모습을 확인할 수 있다. 하지만, 실제 운전자의 주행에 비해서는 속도의 변화가 조금 더 부드럽게 나타나는 것을 확인할 수 있다.

세 번째로, GAIfO를 이용해 생성한 속도 프로파일을 실제 운전자의 데이터와 비교하였다. GAIfO의 경우에도 실제 운전자와 비교적 유사한 궤적을 그리며 주행하는 것을 확인할 수 있었으며, GAIL을 이용했을 때와 유사한 경향을 보이는 것을 알 수 있었다.

마지막으로 실제 운전자의 운전 데이터 및 각 방법에 의해 생성한 속도 프로파일의 제곱평균제곱근 오차(RMSE: Root Mean Square Error)를 비교하였다.

PI제어기나 IDM을 이용하여 속도 프로파일을 생성했을 때 보다 GAIL과 GAIfO를 이용했을 때 보다 실제 운

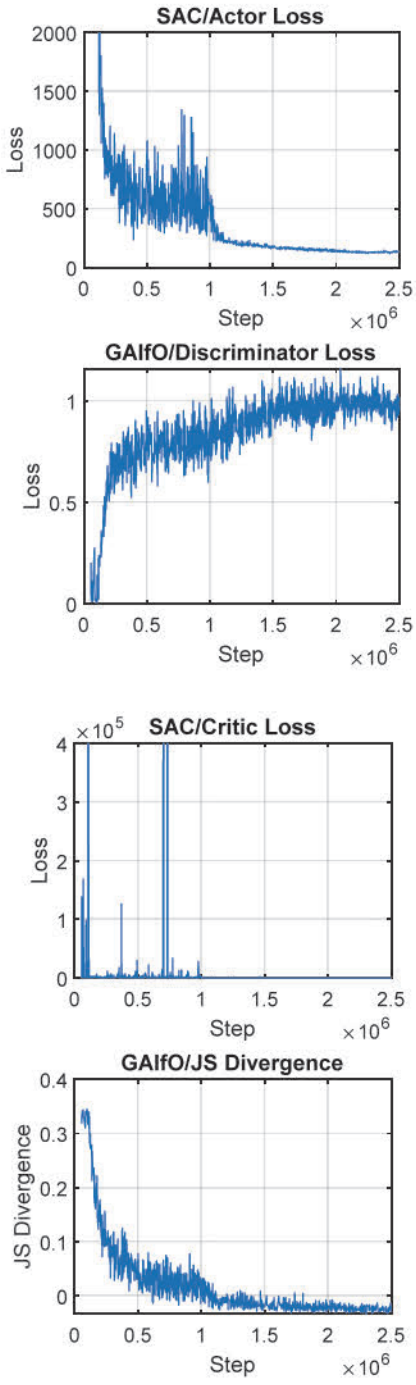

Fig. 16 GAIfO train result

전자와 비슷한 RMSE 값이 나오는 것을 확인할 수 있었다. 특히, 두 방법 중 GAIfO를 사용했을 때 GAIL에 비해 근소하지만 실제 운전자와 더 유사한 RMSE가 나타났는데, 이는 데이터를 취득한 환경과 학습에 이용한 시뮬레이션 환경과의 차이에 의해 행동만을 사용해서 학습했을 때가 조금 더 나은 결과를 보여준다고 생각된다.

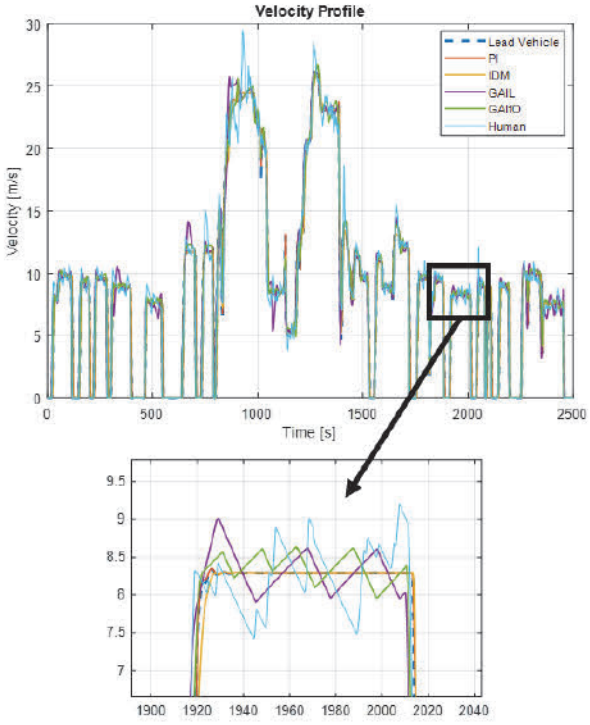

Fig. 17 Longitudinal velocity compare

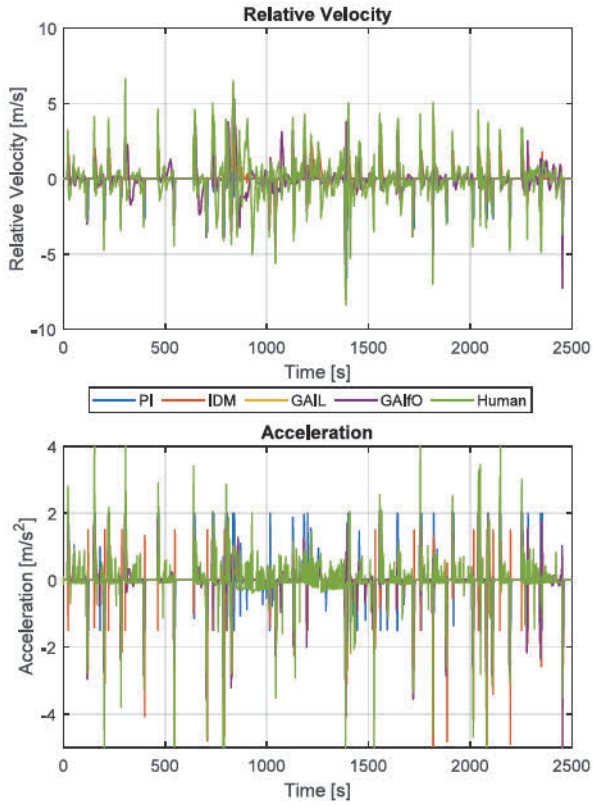

Fig. 18 Relative velocity and acceleration

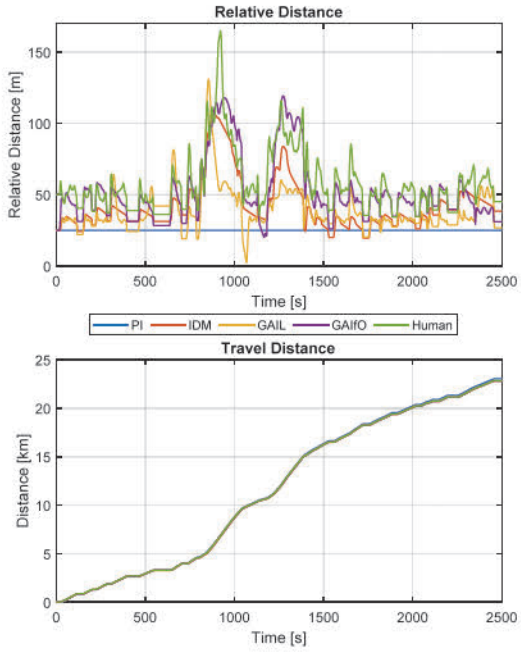

Fig. 19 Relative distance and cumulative distance

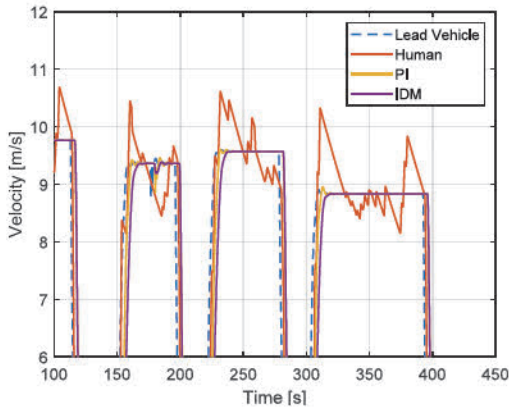

Fig. 20 Longitudinal velocity(Human vs PI/IDM)

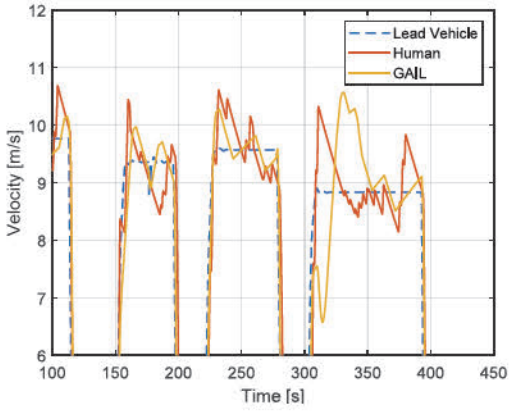

Fig. 21 Longitudinal velocity(Human vs GAIL)

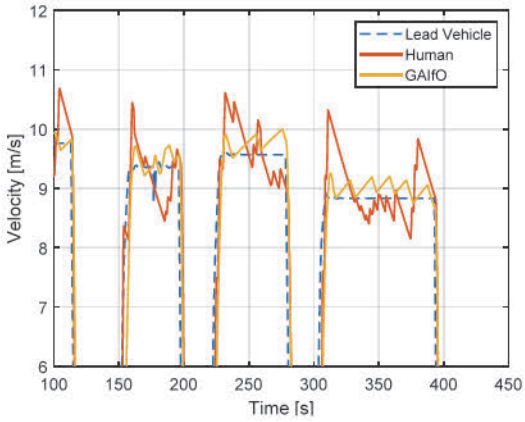

Fig. 22 Longitudinal Velocity(Human vs GAIfO)

Table 6 RMSE between lead vehicle vs ego vehicle

	RMSE
Human	1.359
PI	0.436
IDM	0.593
GAIL	0.942
GAIfO	1.040

5. 결 론

본 논문에서는 모방학습 기반 운전자 모델을 학습하여 이를 활용한 종방향 속도 프로파일 생성 방법을 제시하였다. 실시간 소통정보 API에서 제공하는 구간별 평균 속도만을 이용하여 실제 운전자의 거동과 유사한 가상의 속도 프로파일을 생성하고, 이를 기존의 방법과 비교하였다.

우선, 기존의 IDM이나 PI 제어기를 사용했을 때 보다 실제 운전자와 유사한 속도 프로파일을 생성할 수 있었다. 학습된 에이전트는 전방 차량과의 거리나 차량의 속도, 가속도를 기반으로 적절하게 전방 차량을 추종하였으며, 실제 운전자와 유사한 타력주행을 하는 모습 또한 보여주었다. 모방학습 알고리즘으로 적용한 GAIL과 GAIfO 두 가지 방법을 적용하였을 때 모두 괜찮은 결과를 보여주었기 때문에, 어떤 방법을 적용하더라도 유의미한 결과를 얻을 수 있다는 점을 알 수 있었다. 본 연구에서는 데이터의 시간적인 특성을 고려하는 동시에 빠른 학습 속도를 확보하기 위해 일반적으로 시계열 데이터 학습에 사용되는 RNN(Recurrent Neural Network)나 LSTM(Long Short-Term Memory)가 아닌 1D CNN을 적용하였지만, 차후 연구에서는 시계열 특성을 더 충분히 반영할 수 있는 네트워크를 적용한 연구를 진행할 필요가 있다. 또한 본 논문에서 개발한 종방향운전자 모델에 더

해, 조향이 고려된 종, 횡방향이 통합된 운전자 모델 개발로의 확장을 통해 보다 다양한 상황에 적용할 수 있는 운전자 모델의 개발을 연구할 계획이다.

후 기

이 연구는 2023년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임('20010132').

References

1) S. Kim, Y. Kim, H. Jeon, D. Kum and K. Lee, "Autonomous Driving Technology Trend and Future Outlook: Powered by Artificial Intelligence," Transactions of KSAE, Vol.30, No.10 pp.819-830, 2022.
2) T. Kim, H. Lee, K. Kim and S. H. Hwang, "Path-Following Strategies for 4-Wheel Independent Steering EVs Using PPO Reinforcement Learning and Turning Radius Gain," Transactions of KSAE, Vol.31, No.8, pp.575-584, 2023.
3) A. Hussein, M. M. Gaber, E. Elyan and C. Jayne, "Imitation Learning: A Survey of Learning Methods," ACM Computing Surveys(CSUR), Vol.50, No.2, pp.1-35, 2017.
4) Y. Ng, Andrew and S. Russell, "Algorithms for Inverse Reinforcement Learning," Icml, Vol.1, 2000.
5) J. Ho and S. Ermon, "Generative Adversarial Imitation Learning," Advances in Neural Information Processing Systems, Vol.29, pp.4572-4580, 2016.
6) T. Haarnoja, A. Zhou, P. Abbeel and S. Levine, "Soft Actor-Critic: Off-policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," International Conference on Machine Learning, pp.1861-1870, 2018.
7) V. Mnih, K. Kavukcuoglu, D. Silver and A. Graves, "Playing Atari with Deep Reinforcement Learning," arXiv preprint arXiv:1312.5602, 2013.
8) T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous Control with Deep Reinforcement Learning," arXiv preprint arXiv:1509.02971, 2015.
9) R. Rajamani, Encylopedia of Systems and Control, Springer, Berlin, 20-26, 2021.
10) P. G. Gipps, "A Behavioural Car-Following Model for Computer Simulation," Transportation Research Parb B: Methodological, Vol.15, No.2, pp.105-111, 1981.
11) M. Treiber, A. Hennecke and D. Helbing, "Congested Traffic States in Empirical Observations and Microscopic Simulations," Physical Review E, Vol.62, No.2, pp.1805-1824, 2000.
12) C. Wei, E. Paschalidis, N. Merat, A. Solemou, F. Hajiseyedjavadi and R. Romano, "Human-Like Decision Making and Motion Control for Smooth and Natural Car Following," IEEE Transactions on Intelligent Vehicles, 2021.
13) V. L. Bageshwar, W. L. Garrard and R. Rajamani, "Model Predictive Control of Transitional Manuevers for Adaptive Cruise Control Vehicles," IEEE Transactions on Vehicular Technology, Vol.53, No.5, pp.1573-1585, 2004.
14) J. Morton, T. A. Wheeler and M. J. Kochenderfer, "Analysis of Recurrent Neural Networks for Probabilistic Modeling of Driver Behavior," IEEE Transactions on Intelligent Transportation Systems, Vol.18, No.5, pp.1289-1298, 2016.
15) X. Huang, J. Sun and J. Sun, "A Car-Following Model Considering Asymmetric Driving Behavior Based on Long Short-Term Memory Neural Networks," Transportation Research Part C: Emerging Technologies, Vol.95, pp.346-362, 2018.
16) Q. J. Zou, H. Li and R. Zhang, "Inverse Reinforcement Learning Via Neural Network in Driver Behavior Modeling," IEEE Intelligent Vehicles Symposium(IV), pp.1245-1250, 2018.
17) H. Gao, G. Shi, G. Xie and B. Cheng, "Car-Following Method Based on Inverse Reinforcement Learning for Autonomous Vehicle Decision-Making," International Journal of Advanced Robotic Systems, Vol.15, No.6, 2018.
18) T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel and S. Levine, "Soft Actor-Critic Algorithms and Applications," arXiv preprint arXiv:1812.05905, 2018.
19) F. Torabi, G. Warnell and P. Stone, "Generative Adversarial Imitation from Observation," arXiv preprint arXiv:1807.06158, 2018.
20) Seoul Metropolitan Government, http://data.seoul.go.kr/dataList/OA-13291/A/1/datasetView.do, 2023.