【서지사항】

【서류명】 특허출원서

【출원구분】 특허출원

【출원인】

【명칭】 부경대학교 산학협력단

【특허고객번호】 2-2004-016649-9

【출원인】

【명칭】 에스에이엠(주)

【특허고객번호】 1-2020-096720-1

【대리인】

【성명】 오위환

【대리인번호】 9-2001-000083-1

【대리인】

【성명】 정기택

【대리인번호】 9-2007-000771-5

【대리인】

【성명】 나성곤

【대리인번호】 9-2013-000925-8

【발명의 국문명칭】 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화 학습기반 통행 개선을 위한 장치 및 방법

【발명의 영문명칭】 System and Method for Improving of Advanced Deep Reinforcement Learning Based Traffic in Non signalalized Intersections for the Multiple Self driving Vehicles

【발명자】

【성명】 배상훈

【성명의 영문표기】 BAE, Sanghoon

【주민등록번호】 640827-1XXXXXX

【우편번호】 48513

【주소】 부산광역시 남구 용소로 45 부경대학교대연캠퍼스

【출원언어】 국어

【심사청구】 청구

【공지예외적용대상증명서류의 내용】

【공개형태】 논문발표

【공개일자】 2020.08.18

【이 발명을 지원한 국가연구개발사업】

【과제고유번호】 2020184

【과제번호】 2020-184

【부처명】 중소벤처기업부

【과제관리(전문)기관명】 중소기업기술정보진흥원

【연구사업명】 2020년 예비창업패키지[특화분야]

【연구과제명】 고속도로 주요 결렬지점 고정밀 도로지도 제작 및 자율주행 안전제고 기술

【기여율】 1/1

【과제수행기관명】 한국도로공사

【연구기간】 2020.06.01 ~ 2021.01.31

【취지】 위와 같이 특허청장에게 제출합니다.

대리인 오위환 (서명 또는 인)

대리인 정기택 (서명 또는 인)

대리인 나성곤 (서명 또는 인)

【수수료】

【출원료】 0 면 46,000 원

【가산출원료】 47 면 0 원

【우선권주장료】 0 건 0 원

【심사청구료】 15 항 803,000 원

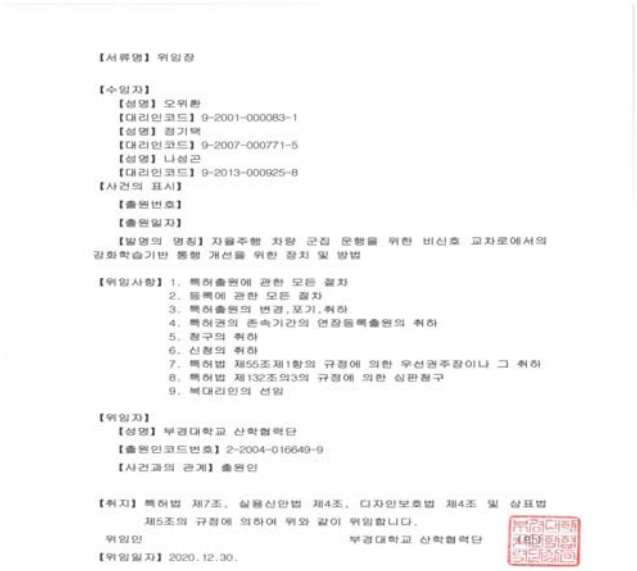
【합계】 849,000 원

【감면사유】 소기업(70%감면)[1],전담조직(50%감면)[1]

【감면후 수수료】 339,600 원

【첨부서류】 1.기타첨부서류[위임장]_1통 2.공지예외적용대상(신규성상실의예외, 출원시의특례)규정을 적용받기 위한 증명서류_1통

1 : 기타첨부서류



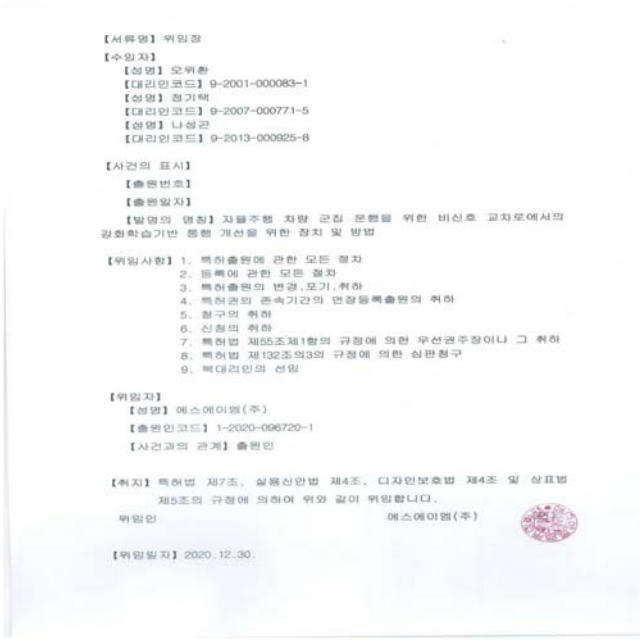
【서류명】 위임장

【수임자】
【성명】 오위환
【대리인코드】 9-2001-000083-1
【성명】 정기택
【대리인코드】 9-2007-000771-5
【성명】 나성곤
【대리인코드】 9-2013-000925-8
【사건의 표시】
【출원번호】
【출원일자】
【발명의 명칭】 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법

【위임사항】 1. 특허출원에 관한 모든 절차
2. 등록에 관한 모든 절차
3. 특허출원의 변경,보기,취하
4. 특허권의 존속기간의 연장등록출원의 취하
5. 청구의 취하
6. 신청의 취하
7. 특허법 제55조제1항의 규정에 의한 우선권주장이나 그 취하
8. 특허법 제132조의3의 규정에 의한 심판청구
9. 복대리인의 선임

【위임자】
【성명】 부경대학교 산학협력단
【출원인코드번호】 2-2004-016649-9
【사건과의 관계】 출원인

【취지】 특허법 제7조, 실용신안법 제4조, 디자인보호법 제4조 및 상표법 제5조의 규정에 의하여 위와 같이 위임합니다.
위임인 부경대학교 산학협력단
【위임일자】 2020.12.30.



【서류명】 위임장
【수임자】
【성명】 오위환
【대리인코드】 9-2001-000083-1
【성명】 정기택
【대리인코드】 9-2007-000771-5
【성명】 나성곤
【대리인코드】 9-2013-000925-8
【사건의 표시】
【출원번호】
【출원일자】
【발명의 명칭】 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법

【위임사항】 1. 특허출원에 관한 모든 절차
2. 등록에 관한 모든 절차
3. 특허출원의 변경,보기,취하
4. 특허권의 존속기간의 연장등록출원의 취하
5. 청구의 취하
6. 신청의 취하
7. 특허법 제55조제1항의 규정에 의한 우선권주장이나 그 취하
8. 특허법 제132조의3의 규정에 의한 심판청구
9. 복대리인의 선임

【위임자】
【성명】 에스에이엠(주)
【출원인코드】 1-2020-096720-1
【사건과의 관계】 출원인

【취지】 특허법 제7조, 실용신안법 제4조, 디자인보호법 제4조 및 상표법 제5조의 규정에 의하여 위와 같이 위임합니다.
위임인 에스에이엠(주)
【위임일자】 2020.12.30.

2 : 공지예외적용대상(신규성상실의예외,_출원시의특례)규정을_적용받기_위한_증명서류

applied sciences

MDPI

Article

Proximal Policy Optimization Through a Deep Reinforcement Learning Framework for Multiple Autonomous Vehicles at a Non-Signalized Intersection

Duy Quang Tran and Sang-Hoon Bae *

Smart Transportation Lab, Pukyong National University, Busan 48513, Korea; tran19868@pukyong.ac.kr
* Correspondence: sbae@pknu.ac.kr

Received: 22 July 2020; Accepted: 17 August 2020; Published: 18 August 2020

check for updates

Abstract: Advanced deep reinforcement learning shows promise as an approach to addressing continuous control tasks, especially in mixed-autonomy traffic. In this study, we present a deep reinforcement-learning-based model that considers the effectiveness of leading autonomous vehicles in mixed-autonomy traffic at a non-signalized intersection. This model integrates the Flow framework, the simulation of urban mobility simulator, and a reinforcement learning library. We also propose a set of proximal policy optimization hyperparameters to obtain reliable simulation performance. First, the leading autonomous vehicles at the non-signalized intersection are considered with varying autonomous vehicle penetration rates that range from 10% to 100% in 10% increments. Second, the proximal policy optimization hyperparameters are input into the multiple perceptron algorithm for the leading autonomous vehicle experiment. Finally, the superiority of the proposed model is evaluated using all human-driven vehicle and leading human-driven vehicle experiments. We demonstrate that full-autonomy traffic can improve the average speed and delay time by 1.38 times and 2.55 times, respectively, compared with all human-driven vehicle experiments. Our proposed method generates more positive effects when the autonomous vehicle penetration rate increases. Additionally, the leading autonomous vehicle experiment can be used to dissipate the stop-and-go waves at a non-signalized intersection.

Keywords: multiple autonomous vehicles; deep reinforcement learning; proximal policy optimization; simulation of urban mobility (SUMO); flow framework

1. Introduction

Traffic congestion leads to a lot of wasted time and slow traffic, and it is one of the main challenges that traffic management agencies and traffic participants have to overcome. According to a national motor vehicle crash survey of the United States, 47% of collisions in 2015 happened at intersections [1]. Automated vehicles (AVs) have recently shown the potential to prevent human errors and improve the quality of a traffic service, with full autonomy expected as soon as 2050 [2]. This means of transportation can save the economy of the United States approximately $450 billion each year [3]. Recently, the intelligent transport system (ITS) domain was developed to provide a smoother, smarter, and safer journey to traffic participants. The early applications of ITS, such as traffic control in Japan, route guidance systems in Berlin, or Intelligent Vehicle Highway Systems in the United States, have been in use since the 1980s. However, the ITS domain concentrates only on intelligent techniques located in vehicles and road infrastructures. To solve communication problems between vehicles and road infrastructures, cooperative intelligent transport systems (C-ITS) can be used to enable those systems to communicate and share information in real time to provide safe and convenient travel. Motivated by the uncertainty in the application of AVs in real environments, this study focuses on

mixed-autonomy traffic settings, in which complex interactions between AVs and human-driven vehicles occur in various continuous control tasks.

In car-following models, adaptive cruise control (ACC) is used to develop driver behavior. ACC systems are an important part of the driver assistance system in premium vehicles and adopt a radar sensor to set the relative distance between vehicles. Previous studies have attempted to connect automated vehicle applications in order to improve traffic safety and capacity. Rajamani and Zhu [4] applied an ACC system to a semi-automated vehicle. The cooperative ACC (CACC) model is a next-generation ACC system that considers both the lead car in the same lane and the car in front in the other lane [5]. Nonetheless, ACC and CACC both depend on constant spacing. As an improvement, the intelligent driver model (IDM) was designed to enhance ACC and CACC systems using real-world experimental data [6]. The IDM, which was introduced by Treiber et al. [7], provides more advantages and realistic values to an ACC system. In particular, the IDM improves the road capacity and reduces the real-time headway [8].

Motivated by the challenges of complex policies, reinforcement learning (RL) was developed based on a trial-and-error method in order to find the best action in uncertain and dynamic environments. RL is a kind of machine learning that differs from supervised learning and unsupervised learning. RL optimizes a reward signal instead of finding a hidden structure. Bellman [9] proposed Markovian decision processes (MDPs) as discrete stochastic methods for optimal control. Howard [10] introduced the policy iteration method that was applied in MDPs. There are basically three kinds of RL methods: policy-based, value-based, and actor-critic methods [11]. Recent studies in RL have applied RL to Atari 2600 games [12], fused reinforcement learning with the Monte Carlo tree search for AlphaGo [13], and applied RL to continuous control tasks [14]. In order to obtain reliable simulation performance, deep reinforcement learning (deep RL) can be used to learn the most appropriate actions in a dynamic environment. In deep RL, RL is fused with an artificial neural network (ANN). Deep RL has, for example, been applied for traffic signal control. Furthermore, recent breakthroughs in artificial intelligence (AI) have been used to develop deep RL methods that are suitable for a range of applications, including high-fidelity simulators, such as virtual environments including the Arcade Learning Environment for more than 55 different games [15], a testing-model-based control platform called multi-joint dynamics with a contact point for control applications [16], and deep convolutional neural networks (CNNs) for guiding the policy search method [17]. Recent studies have applied deep reinforcement learning to adaptive traffic signal control (ATSC) [18,19]. The overview of recent applications for ATSC was based on deep RL [20]. A large-scale traffic light signal for multiple agents was conducted by using a cooperative deep RL framework [21]. The multi-agent RL framework for traffic light control performed better than the previous methods [22]. However, signalized intersection rules are always broken by aggressive drivers. In addition, a non-signalized intersection is a complex traffic situation with a high collision rate. Therefore, it is necessary to study autonomous driving in a mixed-traffic condition at a non-signalized intersection by adopting deep RL.

In order to improve RL's performance during continuous tasks, various studies have applied RL using neural network function approximators, such as deep Q-learning [23], original policy gradient methods [24], and trust region policy optimization (TRPO) [25]. However, deep Q-learning remains poorly understood and fails to converge during many simple tasks. Trust region policy optimization has a high degree of complexity. Proximal policy optimization (PPO) uses multiple epoch updates along a minibatch instead of one gradient update for the sample [26]. Thus, the use of PPO through a deep RL framework has become a promising approach to the control of multiple autonomous vehicles. The PPO-based deep RL was applied to control lane-change decisions according to safety, efficiency, and comfort [27]. In addition, PPO-based deep RL was leveraged to optimize a mixed-traffic condition at a roundabout intersection [28]. Nevertheless, these studies did not consider the PPO hyperparameter within the real traffic volume. Research on PPO hyperparameter for a non-signalized intersection has been lacking.

The most difficult problem for researchers to solve regarding autonomous driving is that of training and validating driving control models in a physical environment. To solve this problem, the simulation approach has been used to represent the real world. Pomerleau [29] used an autonomous land vehicle in a neural network to simulate road images. Recently, the open racing car simulator (TORCS), which is a multi-agent car simulator, was developed based on AI through a lower-level application programming interface [30]. However, TORCS does not support urban driving simulations and lacks such factors as pedestrians, traffic rules, and intersections. More recently, researchers have adopted deep RL to analyze autonomous driving strategies. For example, the car learning to act (CARLA) open urban driving simulator is a trained and validated driving model according to perception and control [31]. However, CARLA is a three-dimensional (3D) simulator for the testing of individual autonomous vehicles. Furthermore, simulation of urban mobility (SUMO), which is an open-source traffic simulator, enables the simulation of traffic scenarios in a large area [32–34], and with traffic signal control [19]. The total possible set of SUMO simulations can be expanded by adopting a traffic control interface (TraCI), which interacts with other programming languages such as Python and Matlab [35]. In addition, Flow is a Python-based open-source tool that can be used to connect a simulator (e.g., SUMO, Aimsun) with a reinforcement learning library (e.g., RLlib, Rllab) [36]. Flow can be used to train a deep RL algorithm and evaluate a mixed-autonomy traffic controller, such as a traffic light or an urban network [37]. Recent studies have applied Flow to evaluate the effectiveness of an automated vehicle (AV) in a network [38,39] and reduce the frequency and magnitude of formed waves with AV penetration rates [40]. The experimental results showed that the multi-agents RL policy outperformed according to average velocity and rewards. In addition, the high average velocity leads to reduce the delay time, fuel consumption, and emissions. Thus, the average velocity has become an effective metric to train a deep RL policy in the real world.

In this study, we present a deep RL method for simulating mixed-autonomy traffic at a non-signalized intersection. Our proposed method combines RL and multilayer perceptron (MLP) algorithms and considers the effectiveness of the leading autonomous vehicles. In addition, we apply a set of PPO hyperparameters to enhance the simulation's performance. First, we perform a leading autonomous vehicle experiment at a non-signalized intersection with a varying AV penetration rate that ranges from 10% to 100% in 10% increments. Second, we input the PPO hyperparameters into the MLP algorithm for the leading autonomous vehicle experiment. Finally, human-driven leading vehicle and all human-driven vehicle experiments are used to evaluate the superiority of the proposed method. The major contributions of this work are as follows.

- An enhanced hybrid deep RL method is presented that uses a PPO algorithm through MLP and RL models in order to consider the effectiveness of the leading autonomous vehicle experiment at a non-signalized intersection based on an AV penetration rate that ranges from 10% to 100% in 10% increments. The leading autonomous vehicle experiment yields a significant improvement when compared with the leading human-driven vehicle and all human-driven vehicle experiments in terms of training policy, mobility, and energy efficiency.
- A set of PPO hyperparameters is proposed in order to explore the effect of the automated extraction feature on policy prediction and to obtain reliable simulation performance at a non-signalized intersection within the real traffic volume.
- The demonstration of a significant improvement in traffic perturbations at a non-signalized intersection is based on an AV penetration rate that ranges from 10% to 100% in 10% increments.

The rest of this paper is organized as follows. Section 2 presents the deep RL framework, the longitudinal dynamic models, the policy optimization method, and the proposed model's architecture. Section 3 describes the simulation experiments and presents the results. Section 4 contains our conclusions.

2. Methods

2.1. Deep Reinforcement Learning (Deep RL)

Reinforcement learning (RL) is a subarea of machine learning and is concerned with how agents interact with an environment and learn to take actions that maximize their cumulative reward. The typical form of the RL algorithm is a Markov decision process (MDP), which is a strong framework used to determine a proper action given a full set of observations [9]. An MDP is a tuple (S, A, P, R, ρ_0, γ, T), where S and A are states and actions of a participant, respectively; $P(S', S, a)$ defines a probability for transition; $R(a, S)$ defines the reward according to the selected action; ρ_0 defines the initial state distribution; γ defines the discount factor, which ranges from 0 to 1; and T denotes the time horizon. However, automated vehicles maneuver in an uncertain environment that contains inaccuracy, intentions, and sensor noise. To solve this problem, a partially observable MDP (POMDP) was proposed that employs two more components, namely O, which defines the set of observations, and Z, which is an observation function. An objective learning agent in RL optimizes the policy π to maximize their expected cumulative discounted reward over some number of time steps.

A deep neural network (DNN) has the ability to automatically perform feature extraction due to multiple hidden layers of representations. For continuous controllers, artificial neural networks (ANNs) are commonly used methods that employ multiple hidden layers to represent complex functions. In this work, we apply an MLP to generate a set of outputs (policy) from a set of inputs (states and observations). In addition, we apply a PPO based on a gradient descent optimization method to enhance the performance of the DNN. Our proposed deep RL framework, which fuses a MLP and RL, is designed to consider the effectiveness of AVs at a non-signalized intersection. First, the SUMO simulator executes one simulation step. Second, the Flow framework sends information on the SUMO simulator's state to the RL library. Then, the RL library (RLlib) computes the appropriate action according to SUMO simulator's state through MLP. The MLP policy is applied to maximize the cumulative reward for the RL algorithm based on the traffic data. Finally, the simulation resets and iterates the RL process. Figure 1 presents the deep reinforcement learning architecture in the context of a non-signalized intersection.

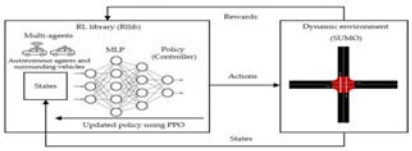


Figure 1. The deep reinforcement learning architecture in the context of a non-signalized intersection.

Importantly, a 'policy' refers to a blueprint of the communication between perceptions and actions in an environment. In other words, a policy is similar to a controller of a traffic simulation. In this work, the controller is an MLPpolicy with multiple hidden layers. The parameters of the controller are iteratively updated by using the MLPpolicy to maximize the cumulative reward based on the traffic

data sampled from the SUMO simulator. The main goal of the agent is to learn how to optimize a stochastic policy as follows.

$$\theta^* := argmax_\theta \eta(\pi_\theta) \tag{1}$$

where $\eta(\pi_\theta)$ is the expected cumulative discounted reward, which is calculated by the discount factor (γ_t) and the reward (r).

$$\eta(\pi_\theta) = \sum_{t=0}^{T} \gamma_t r_t \tag{2}$$

2.2. Longitudinal Dynamic Models

Basic vehicle dynamics can be defined by car-following models, which describe the longitudinal dynamics of a manually operated vehicle based on observations of the vehicle itself and vehicles in front. A standard car-following model is as follows:

$$a_i = f\left(h_i, \dot{h}_i, v_i\right) \tag{3}$$

where a_i is the acceleration of vehicle i, $f()$ is a nonlinear function, and v_i, $\dot{h}_i$, and h_i are the velocity, relative velocity, and headway of vehicle i, respectively.

In this work, we apply the IDM, which is a type of ACC system, for the longitudinal control of human-driven vehicles due to its capacity to depict realistic driver behavior [7]. The IDM is a commonly used car-following model. In the IDM's acceleration command, the speed of a vehicle in a non-signalized intersection environment and the identification (ID) and headway of the leading vehicle can be set to be obtained by the "get" methods. The acceleration of the vehicle is calculated as follows.

$$a_{IDM} = a\left[1 - \left(\frac{v}{v_0}\right)^\delta - \left(\frac{s^*(v, \Delta v)}{s}\right)^2\right] \tag{4}$$

where a_{IDM} is the acceleration of the vehicle, v_0 is the desired speed, δ is an acceleration exponent, s is the vehicle's headway (the distance to the vehicle ahead), and $s^*(v, \Delta v)$ indicates the desired headway, which is expressed by:

$$s^*(v, \Delta v) = s_0 + max\left(0, vT + \frac{v\Delta v}{2\sqrt{ab}}\right) \tag{5}$$

where S_0 denotes the minimum gap, T denotes a time gap, Δv denotes the velocity difference compared with the lead vehicle (current velocity–lead velocity), a denotes an acceleration term, and b denotes comfortable deceleration.

The typical parameters of an IDM controller for city traffic are represented in Table 1 based on [41].

Table 1. Typical parameters of an intelligent driver model (IDM) controller for city traffic.

Parameters	Value
Desired speed (m/s)	15
Time gap (s)	1.0
Minimum gap (m)	2.0
Acceleration exponent	4.0
Acceleration (m/s^2)	1.0
Comfortable acceleration (m/s^2)	1.5

2.3. Policy Optimization

Policy gradient methods attempt to compute an estimator of a parameterized policy function using a gradient descent algorithm rather than an action-value or a state-value function. Thus, they avoid the convergence problems that occur with estimation functions due to non-linear approximation and partial observation. We applied the MLP policy to optimize the control policy directly in the simulation

of the non-signalized intersection. The policy gradient laws, which are based on the expectation over the probabilities of the policy actions ($\log \pi_\theta$) and an estimate of the advantage function at time step t ($\hat{A}_t$), are expressed as follows.

$$\hat{g} = \hat{E}_t\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right] \tag{6}$$

where $\hat{E}_t[.]$ is the expectation operator over a finite batch of samples, π_θ indicates a stochastic policy, $\hat{A}_t$ is defined by the discounted sum of rewards and a baseline estimate, and a_t and s_t express the action and state at time step t, respectively.

PPO, which was proposed by Schulman et al. [26], is a simple TRPO that is provided by the RLlib library. In other words, PPO's objective is the same as that of TRPO, which uses a trust region constraint to force the policy update to ensure that the new policy is not too far away from the old policy. There are two types of PPO: adaptive Kullback–Leibler (KL) penalty and clipped objective. The PPO generates policy updates by adopting a surrogate loss function. This process avoids a reduction in performance during the training process. The surrogate object (J^{CPI}) is described as follows.

$$J^{CPI}(\theta) = \hat{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] \tag{7}$$

where $\pi_{\theta_{old}}$ indicates a policy parameter before update, π_θ indicates a policy parameter after update, and $r_t(\theta)$ indicates the probability ratio.

For continuous actions, the PPO's policy output is a parameter of the Gaussian distribution for each action. The policy then generates a continuous output based on these distributions. In this work, PPO with an adaptive KL penalty is used to optimize the KL-penalized objective by using minibatch stochastic gradient descent (SGD) as follows.

$$\underset{\theta}{\text{maximize}}\ \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] - \beta\hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right] \tag{8}$$

$$\text{Subject to } \hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right] \leq \delta \tag{9}$$

where β is the weight control coefficient that is updated after every policy update. If the current KL divergence is greater than the target KL divergence, we increase β. Similarly, if the current KL divergence is less than the target KL divergence, we decrease β.

In the PPO algorithm, first, the current policy interacts with the environment to generate the episode sequences. Next, the advantage function is estimated using the baseline estimate for the state value. Finally, we collect all experiences and execute the gradient descent algorithm over the policy network. The complete PPO with an adaptive KL penalty algorithm is presented in pseudocode in Algorithm 1, shown below [42].

Importantly, the PPO hyperparameters provide a robust approach to enhancing the effectiveness of RL at various tasks. In particular, Gamma (γ) is a discount factor that ranges from 0 to 1 and indicates how important future rewards are to the current state. The hidden layers affect the accuracy and performance. With more hidden layers, the accuracy increases; however, the performance decreases. Lambda (λ) is a smoothing rate that reduces the variance during the training process to ensure that training progresses in a stable manner. The Kullback–Leibler (KL) target is the desired policy change for each iteration.

Algorithm 1 PPO with an Adaptive KL Penalty Algorithm

Initial policy parameters θ_0, weight control β_0, target KL-divergence δ_{tag}
For k = 0, 1, 2 .... do
Gather set of trajectories on policy $\pi_k = \pi(\theta_k)$
Optimize the KL penalized using minibatch SGD

$$J^{KLPEN}(\theta) = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] - \beta\hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]$$

Compute KL divergence between the new and old policy

$$\delta = \hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]$$

If $\delta > 1.5\ \delta_{tag}$ then
$\beta_{k+1} = 2\beta_k$
Else if $\delta < \delta_{tag}/1.5$ then
$\beta_{k+1} = \beta_k/2$
Else
pass
End if
End for

2.4. Proposed Method's Architecture

In this study, we applied the open-source modular learning framework Flow to connect the RL library (RLlib) to the traffic simulator (SUMO). Flow allows us to simulate varied and complex traffic environments, multiple agents, and multiple algorithms [43]. The implementation is based on SUMO [35], Ray RLlib for RL [44], and the OpenAI gym for the MDP [45]. Our study focuses on an online optimization in a closed-loop setting through deep RL. Figure 2 shows the proposed method's architecture.

Figure 2. The proposed method's architecture.

SUMO, developed by the Institute of Transportation Systems at the German Aerospace Center, is an open-source microscopic traffic simulator. SUMO can simulate urban-scale traffic networks along with traffic lights, vehicles, pedestrians, and public transportation. In addition, the TraCI enables SUMO to be connected to Python in order to apply deep RL to the SUMO simulator. A typical SUMO simulator at a non-signalized intersection is shown in Figure 3.

Figure 3. A typical A simulation of urban mobility (SUMO) at a non-signalized intersection.

Flow [44], developed by UC Berkeley, provides an interface between deep RL algorithms and custom road networks. Additionally, Flow can analyze and validate a training policy. The advantages of Flow include the ability to easily implement varied road networks in order to enhance controllers for autonomous vehicles through deep RL. In Flow, a custom environment can be used to generate the main subset class, including initialized simulation, observation space, state space, action space, controller, and reward function, for various scenarios.

2.4.1. Initialized Simulation

The initialized simulation expresses the initial settings of the simulation environment for the starting episode. In particular, we set up the position, speed, acceleration, starting points, trajectories, and number of vehicles, as well as the parameters of the IDM rules and the deep RL framework. In particular, the trajectories of all vehicles is set in the initialized simulation process by SUMO simulator including specific nodes (the position of points in the network), specific edges (linked the nodes together), and specific routes (the sequence of edges vehicles traverse). Next, the acceleration of human-driven vehicles is controlled by the SUMO simulator and the acceleration of AVs is controlled by Rllib library.

2.4.2. Observation Space

The observation space expresses the number and types of observable features, namely the AV speed (ego vehicle speed), the AV position (ego position), and the speeds and bumper-to-bumper headways of the corresponding preceding and following AVs described in Figure 4. The observable output is fed into the state space to predict the proper policy.

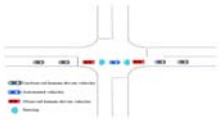


Figure 4. A typical observation space.

2.4.3. State Space

A state space represents a vector of autonomous agents and surrounding vehicles based on the observation space, including the positions and velocities of AVs, as well as preceding and following AVs. The features within the environments are extracted and fed into the policy using the get_state method. First, we obtain the ID of all vehicles at the non-signalized intersection. Then, the positions and velocities of all vehicles are obtained to generate the state space. Importantly, the current position is based on pre-specified starting point. The state space is defined as follows:

$$S = \begin{pmatrix} x_0 \\ v_0 \\ v_l \\ d_l \\ v_f \\ d_f \end{pmatrix} \tag{10}$$

where S is the state of a specific vehicle, x_0 is the corresponding coordinates of the AV, v_0, v_l, and v_f are the corresponding speeds of the AV, the preceding AV, and the following AV, respectively, and d_l and d_f denote the bumper-to-bumper headways of the preceding AV and the following AV, respectively.

2.4.4. Action Space

The action space represents the actions of the autonomous agents in the traffic environment provided by the OpenAI gym. The standard action for an automated vehicle would be an acceleration. In the action space, the bounds of the actions range from maximum deceleration to maximum acceleration. Then, the apply_RL_actions function is applied to transform a specific command into an actual action in the SUMO simulator. First, we identify all AVs at the non-signalized intersection. Then, the action commands are converted into accelerations using the base environment method.

2.4.5. Controller

The controller controls the behaviors of the actors, including human-driven vehicles and AVs. A single controller can be applied to multiple actors using shared control. In this work, the human-driven vehicles are controlled by the Flow framework, and the automated vehicles are controlled by the RLlib library.

2.4.6. Reward Function

In order to reduce the traffic congestion, we need to optimize the average speed of the network thanks to reducing delay time, queue lengths. Therefore, the average speed has become a promising metric to train deep RL policy in the real world. The reward function defines the way in which an autonomous agent will attempt to optimize a policy. In this work, the goal of an RL agent is to obtain a high average speed while punishing collisions between vehicles at a non-signalized intersection. In this study, the L2 norm was used to estimate the positive distance given the speed of a vehicle at a non-signalized intersection based on the target speed (the desired speed of all vehicles at a non-signalized intersection). In particular, we applied the get-speed method to obtain the current speed of all vehicles at the non-signalized intersection and then return the average speed as the reward. The reward function is expressed as follows [32]:

$$r_t := \max\left(\|v_{des}, 1^k\|_2 - \|v_{des} - v\|_2, 0\right) / \|v_{des}, 1^k\|_2 \tag{11}$$

where v_{des} denotes an arbitrary desired speed and $v \in R^k$ denotes the speeds of all vehicles at a non-signalized intersection.

2.4.7. Termination

The termination of a rollout is based on the training iteration and collisions as follows: (1) the training iteration is complete; (2) a collision between two vehicles occurred.

3. Experimental Results and Analysis

3.1. Hyperparameter Setting and Evaluation Metrics

The PPO with an adaptive KL penalty algorithm controls the distance between the updated policy and the old policy in order to avoid noise during a gradient update. Hyperparameter tuning was used to select the proper variables for the training process. Hence, PPO hyperparameter initializations improve the effectiveness of RL at various tasks. In this study, we propose the set of PPO hyperparameters for mixed-autonomy traffic at a non-signalized intersection shown in Table 2. The time horizon per training iteration is calculated by multiplication between the time horizon of a single rollout and the number of rollouts per training iteration. The time horizon of a single rollout is 600 and the number of rollouts per training iteration is 10. Therefore, the time horizon per training iteration is 6000. The "256 × 256 × 256" means that we set up 3 hidden layers, and each layer has 256 neurons at all. In addition, based on our experiment, the agent performs well at the number of training iteration of 200.

Table 2. Proximal policy optimization (PPO) hyperparameters for mixed-autonomy traffic at a non-signalized intersection.

Parameters	Value
Number of training iterations	200
Time horizon per training iteration	6000
Gamma	0.99
Hidden layers	256 × 256 × 256
Lamda	0.95
Kullback–Leibler (KL) target	0.01
Number of SGD iterations	10

The training policy's performance was verified by the maximum reward curve over 200 iterations. A flattening of the curve indicates that the training policy has completely converged. Furthermore, the simulation performance was evaluated by measures of effectiveness (MOE), which are designed to analyze traffic operations. Such an evaluation can help to predict and address traffic issues. In this study, we adopted the following MOE to evaluate the simulation's performance:

- Mean speed: the average speed of all vehicles at a non-signalized intersection.
- Delay time: the time difference between real and free-flow travel times of all vehicles.
- Fuel consumption: the average fuel consumption value of all vehicles.
- Emissions: the average emission values of all vehicles, including nitrogen oxide (NOx) and hydrocarbons (HC).

3.2. Experimental Scenarios

In this study, vehicles that crossed the non-signalized intersection followed a right-of-way rule supplied by the SUMO simulator. The objective of the right-of-way rule is to enforce traffic rules and also avoid traffic collisions. Moreover, we observed the positions of all vehicles and converted the environment from a POMDP to an MDP. Importantly, autonomous agents learned to optimize a certain reward over the rollouts using the RLlib library. Our simulation uses RL agents to represent a human-driven fleet and an entire traffic flow in mixed-autonomy traffic. The RL agents receive the updated state and bring about a new state in time steps of 0.1 s. For human-driven vehicles, acceleration behaviors are controlled by the IDM model. Furthermore, continuous routing is applied to maintain the vehicles within the network.

We executed the simulation experiments with time steps of 0.1 s, a lane width of 3.2 m, two lanes in each direction, a lane length in each direction of 420 m, a maximum acceleration of 3 m/s^2, a minimum acceleration of −3 m/s^2, a maximum speed of 12 m/s, a horizon of 600, and 200 iterations for the training process. We set an inflow of 1000 vehicles per hour in each direction. The range of the non-signalized intersection was between 200 m and 220 m. In the field, many different scenarios must be simulated. However, in this study, we limited our focus to the effectiveness of the leading autonomous vehicle at a non-signalized intersection. Platooning vehicles approach a non-signalized intersection and drive straight ahead following four different directions. In addition, we also present the results for AV penetration rates ranging from 1% to 100% in 10% increments. Importantly, we ignored lane changing and turning left for all vehicles at the non-signalized intersection. Figure 5 shows the leading autonomous vehicle experiment at the non-signalized intersection.

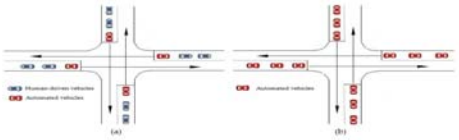


Figure 5. Leading autonomous vehicle experiments at a non-signalized intersection: **(a)** mixed-autonomy traffic with autonomous vehicle (AV) penetration rates ranging from 10% to 90% in 10% increments; **(b)** full-autonomy traffic with a 100% AV penetration rate.

To demonstrate the superiority of the leading autonomous vehicle experiment, we compared the leading autonomous vehicle experiment with other experiments, including the leading human-driven vehicle experiment and the all human-driven vehicle experiment. Figure 6 shows a comparison of the experiments at a non-signalized intersection.

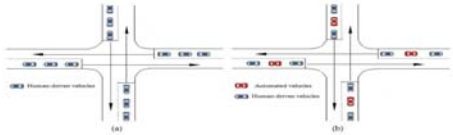


Figure 6. Comparison of experiments at a non-signalized intersection: **(a)** the all human-driven vehicle experiment with a 0% AV penetration rate; **(b)** the leading human-driven vehicle experiment with AV penetration rates ranging from 10% to 90% in 10% increments.

3.3. Experimental Results and Analysis

3.3.1. Training Policy's Performance

The RL training performance through the AV penetration rate was used to evaluate the learning performance. Figure 7 shows the average reward curve over 200 iterations based on the AV penetration rates. The flattening of the curve in all circumstances indicates that the training policy had almost converged. Moreover, the average reward increased as the AV penetration rate at the non-signalized intersection increased, except for the 50% AV penetration rate. Full-autonomy traffic outperformed the other AV penetration rates; it produced the highest average reward and significant flattening of the curve. In particular, full-autonomy traffic yielded an improvement of 6.8 times compared with the 10% AV penetration rate. Therefore, full-autonomy traffic outperformed the other AV penetration rates in all circumstances. The effectiveness of the leading autonomous vehicle experiment at the non-signalized intersection became more obvious as the AV penetration rate increased.

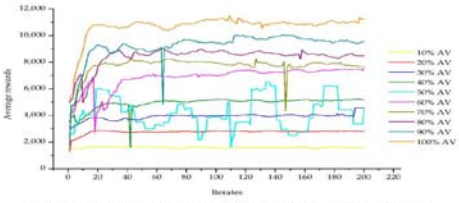


Figure 7. The average reward curve over the 200 iterations based on the AV penetration rate.

3.3.2. Effect of the Leading Automated Vehicle on the Smoothing Velocity

We considered the effect of the leading automated vehicle on the smoothing velocity. Figure 8 shows the spatio-temporal dynamics through the AV penetration rate at the non-signalized intersection. The points are color-coded based on the velocity. The points closer to the top denote smooth traffic. In contrast, the points closer to the bottom denote congested traffic. For the lower AV penetration rates, perturbations occurred due to stop-and-go waves of human-driven vehicle behavior, reducing the velocity in the non-signalized intersection area (ranging from 200 m to 220 m). As can be seen in Figure 8, almost all of the points are close to the bottom in the non-signalized intersection area with the lower AV penetration rates. This is because human-driven vehicles simultaneously approach the non-signalized intersection area and slow down according to the right-of-way rule. At the higher AV penetration rates, the points are close to the top, with the AVs slowing down in a shorter time, thereby producing fewer and shorter stop-and-go waves in the non-signalized intersection area. Full-autonomy traffic achieved the highest smoothing velocity of all AV penetration rates. Thus, the traffic congestion was partially cleared, and traffic flow became smoother as the AV penetration rate increased.

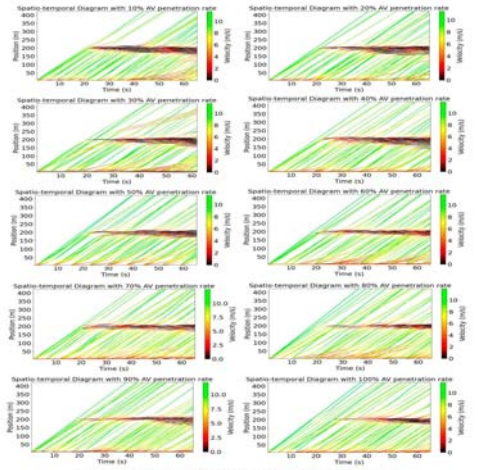


Figure 8. The spatio-temporal dynamics based on the AV penetration rate.

3.3.3. Effect of the Leading Automated Vehicle on Mobility and Energy Efficiency

Figure 9 shows the MOE evaluation in terms of average speed, delay time, fuel consumption, and emissions based on the AV penetration rates. The results of the MOE evaluation indicate that the simulation became more effective as the AV penetration rate increased. Regarding mobility, the average speed gradually increased and the delay time gradually decreased as the AV penetration rate increased. As can be seen in Figure 9a,b, full-autonomy traffic achieved an improvement in average speed of 1.19 times and an improvement in delay time of 1.76 times compared with the 10% AV penetration rate. The energy efficiency, fuel consumption, and emissions slightly decreased as the AV penetration rate increased. As shown in Figure 9c,d, full-autonomy traffic achieved an improvement in fuel consumption of 1.05 times and an improvement in emissions of 1.22 times compared with the 10% AV penetration rate. Thus, leading autonomous vehicles are more effective in terms of mobility and energy efficiency when the AV penetration rate increases.

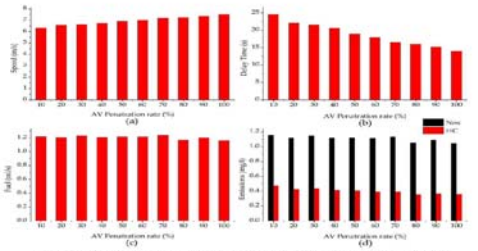


Figure 9. The results of the measures of effectiveness (MOE) evaluation based on the AV penetration rate: (**a**) average speed vs. AV penetration rate; (**b**) delay time vs. AV penetration rate; (**c**) fuel consumption vs. AV penetration rate; and (**d**) emissions vs. AV penetration rate.

3.3.4. Performance Comparison

To verify the superiority of the leading autonomous vehicle experiment (the proposed experiment), we compared it with other experiments, including an all human-driven vehicle experiment and a leading human-driven vehicle experiment. The comparison between the proposed experiment and the all human-driven vehicle experiment is shown in Figure 10. Regarding mobility, the proposed experiment achieved a higher average speed and a lower delay time compared with the all human-driven vehicle experiment. As seen in Figure 10a,b, the 10% AV penetration rate achieved an improvement in average speed of 1.16 times and an improvement in delay time of 1.44 times compared with all human-driven vehicle experiment. Furthermore, full-autonomy traffic achieved an improvement in average speed of

1.38 times and an improvement in delay time of 2.55 times compared with the all human-driven vehicle experiment. In terms of energy efficiency, the proposed experiment achieved lower fuel consumption and emissions compared with the all human-driven vehicle experiment. As shown in Figure 10c, the 10% AV penetration rate achieved an improvement in average reward of 3.63 times compared with the all human-driven vehicle experiment. In addition, full-autonomy traffic achieved an improvement in average reward of 24.77 times compared with the all human-driven vehicle experiment. Hence, the proposed experiment outperformed the all human-driven vehicle experiment in terms of both mobility and energy efficiency.

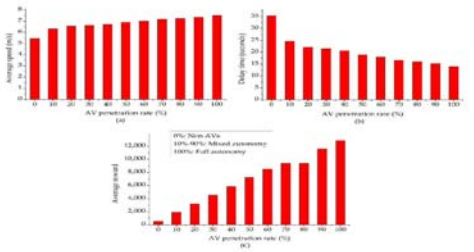


Figure 10. Comparison between the leading autonomous vehicle experiment and the all human-driven vehicle experiment: (**a**) average reward vs. AV penetration rate; (**b**) average speed vs. AV penetration rate; and (**c**) delay time vs. AV penetration rate.

The comparison between the proposed experiment and the leading human-driven vehicle experiment is shown in Figure 11. Regarding mobility, the proposed experiment achieved a higher average speed and a lower delay time compared with the leading human-driven vehicle experiment. As seen in Figure 11a,b, the proposed experiment achieved an improvement in average speed of 1.02 times and an improvement in delay time of 1.06 times compared with the leading human-driven vehicle experiment. In terms of energy efficiency, the proposed experiment achieved lower fuel consumption and emissions compared with the leading human-driven vehicle experiment. As shown in Figure 11c, the proposed experiment achieved an improvement in average reward of 1.22 times compared with the leading human-driven vehicle experiment. Hence, the proposed experiment outperformed the leading human-driven vehicle experiment in terms of both mobility and energy efficiency.

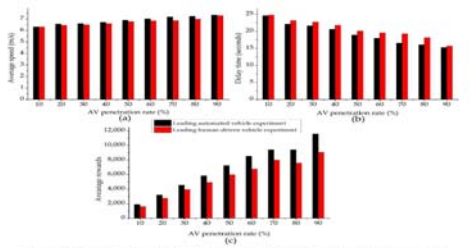


Figure 11. Comparison between the leading autonconous vehicle experiment and the leading human-driven vehicle experiment: (**a**) average reward vs. AV penetration rate; (**b**) average speed vs. AV penetration rate; and (**c**) delay time vs. AV penetration rate.

4. Discussion and Conclusions

In this study, we demonstrated that leading autonomous vehicles become more effective in terms of training policy, mobility, and energy efficiency as their AV penetration rates increase. The traffic congestion was partially cleared, and the traffic flow became smoother as the AV penetration rate increased. Full-autonomy traffic was shown to outperform all other AV penetration rates. In particular, full-autonomy traffic improved the average speed and delay time by 1.38 times and 2.55 times, respectively, compared with all human-driven vehicle experiment. The leading autonomous vehicle experiment was shown to outperform both the all human-driven vehicle experiment and the leading human-driven vehicle experiment.

In summary, the leading autonomous vehicle experiment, which uses a set of PPO hyperparameters and deep RL, performed better than the leading human-driven vehicle experiment and the all human-driven vehicle experiment. The main contributions of this work are the proposed set of PPO hyperparameters and the deep RL framework, which together resulted in a reliable simulation of mixed-autonomy traffic at a non-signalized intersection based on the AV penetration rate. The proposed method provides more positive effects when the AV penetration rate increases. Additionally, researchers could adopt the leading autonomous vehicle experiment to dissipate stop-and-go waves. In our future work, we will consider the efficiency of multiple autonomous vehicles for the network with multi-intersections by developing more advanced deep machine learning algorithms.

Author Contributions: The authors jointly proposed the idea and contributed equally to the writing of the manuscript. D.Q.T. designed the algorithms and performed the simulation. S.-H.B., the corresponding author, supervised the research and revised the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. National Highway Traffic Safety Administration. Traffic Safety Facts 2015: A Compilation of Motor Vehicle Crash Data from the Fatality Analysis Reporting System and the General Estimates System. The Fact Sheets and Annual Traffic Safety Facts Reports, USA. 2017. Available online: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812384 (accessed on 26 April 2017).
2. Wadud, Z.; MacKenzie, D.; Leiby, P.N. Help or hindrance? The travel, energy and carbon impacts of highly automated vehicles. *Transp. Res. Part A Policy Pract.* **2016**, *86*, 1–18. [CrossRef]
3. Fagnant, D.; Kockelman, K. Preparing a nation for automated vehicles: Opportunities, barriers and policy recommendations. *Transp. Res. Part A Policy Pract.* **2015**, *77*, 167–181. [CrossRef]
4. Rajamani, R.; Zhu, C. Semi-autonomous adaptive cruise control systems. *IEEE Trans. Veh. Technol.* **2002**, *51*, 1186–1192. [CrossRef]
5. Davis, L. Effect of adaptive cruise control systems on mixed traffic flow near an on-ramp. *Phys. A Stat. Mech. Appl.* **2007**, *379*, 274–290. [CrossRef]
6. Milanés, V.; Shladover, S.E. Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data. *Transp. Res. Part C Emerg. Technol.* **2014**, *48*, 285–300. [CrossRef]
7. Treiber, M.; Hennecke, A.; Helbing, D. Congested traffic states in empirical observations and microscopic simulations. *Phys. Rev. E* **2000**, *62*, 1805–1824. [CrossRef]
8. Yang, L.; Zhang, X.; Geng, J.; Liu, J. The Research of Car-Following Model Based on Real-Time Maximum Deceleration. *Math. Probl. Eng.* **2015**, *2015*, 1–9. [CrossRef]
9. Bellman, R. A Markovian Decision Process. *J. Math. Mech.* **1957**, *6*, 679–684. [CrossRef]
10. Howard, R.A. *Dynamic Programming and Markov Processes*; The M.I.T. Press: Cambridge, UK, 1960.
11. Sutton, R.; Barto, A. Reinforcement Learning: An Introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]
12. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv* **2013**, arXiv:1312.5602.
13. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; Driessche, G.V.D.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef] [PubMed]
14. Duan, Y.; Chen, X.; Houthooft, R.; Schulman, J.; Abbeel, P. Benchmarking deep reinforcement learning for continuous control. *arXiv* **2016**, arXiv:1604.06778.
15. Bellemare, M.G.; Naddaf, Y.; Veness, J.; Bowling, M. The Arcade Learning Environment: An Evaluation Platform for General Agents. *J. Artif. Intell. Res.* **2013**, *47*, 253–279. [CrossRef]
16. Todorov, E.; Erez, T.; Tassa, Y. MuJoCo: A Physics Engine for Model-Based Control. In *Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*; Institute of Electrical and Electronics Engineers (IEEE): Vilamoura, Portugal, 7–12 October 2012; pp. 5026–5033.
17. Levine, S.; Finn, C.; Darrell, T.; Abbeel, P. End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **2016**, *17*, 1–40.
18. Tan, K.L.; Poddar, S.; Sarkar, S.; Sharma, A. Deep Reinforcement Learning for Adaptive Traffic Signal Control. In *Proceedings of the Volume 3, Rapid Fire Interactive Presentations: Advances in Control Systems; Advances in Robotics and Mechatronics; Automotive and Transportation Systems; Motion Planning and Trajectory Tracking; Soft Mechatronic Actuators and Sensors; Unmanned Ground and Aerial Vehicles*; ASME International: Park City, UT, USA, 9–11 October 2019.
19. Gu, J.; Fang, Y.; Sheng, Z.; Wen, P. Double Deep Q-Network with a Dual-Agent for Traffic Signal Control. *Appl. Sci.* **2020**, *10*, 1622. [CrossRef]
20. Gregurić, M.; Vujić, M.; Alexopoulos, C.; Miletić, M. Application of Deep Reinforcement Learning in Traffic Signal Control: An Overview and Impact of Open Traffic Data. *Appl. Sci.* **2020**, *10*, 4011. [CrossRef]
21. Tan, T.; Bao, F.; Deng, Y.; Jin, A.; Dai, Q.; Wang, J. Cooperative Deep Reinforcement Learning for Large-Scale Traffic Grid Signal Control. *IEEE Trans. Cybern.* **2019**, *50*, 2687–2700. [CrossRef]

22. Bakker, B.; Whiteson, S.; Kester, L.; Groen, F. Traffic Light Control by Multiagent Reinforcement Learning Systems. *ITIL* **2010**, *281*, 475–510. [CrossRef]
23. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nat.* **2015**, *518*, 529–533. [CrossRef]
24. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.P.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. *arXiv* **2016**, arXiv:1602.01783.
25. Schulman, J.; Levine, S.; Moritz, P.; Jordan, M.I.; Abbeel, P. Trust region policy optimization. *arXiv* **2015**, arXiv:1502.05477.
26. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
27. Ye, F.; Cheng, X.; Wang, P.; Chan, C.-Y. Automated Lane Change Strategy using Proximal Policy Optimization-based Deep Reinforcement Learning. *arXiv* **2020**, arXiv:2002.02667.
28. Wei, H.; Liu, X.; Mashayekhy, L.; Decker, K. Mixed-Autonomy Traffic Control with Proximal Policy Optimization. In *Proceedings of the 2019 IEEE Vehicular Networking Conference (VNC)*; Institute of Electrical and Electronics Engineers (IEEE): Los Angeles, CA, USA, 4–6 December 2019; pp. 1–8.
29. Pomerleau, D.A. An autonomous land vehicle in a neural network. *Adv. Neural Inf. Process. Syst.* **1988**, *1*.
30. Wymann, B.; Espié, E.; Guionneau, C.; Dimitrakakis, C.; Coulom, R.; Sumner, A. TORCS, the Open Racing Car Simulator, v1.3.5. Available online: http://www.torcs.org (accessed on 1 January 2013).
31. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. *arXiv* **2017**, arXiv:1711.03938.
32. Behrisch, M.; Bieker, L.; Erdmann, J.; Krajzewicz, D. SUMO—Simulation of Urban MObility: An Overview. In Proceedings of the Third International Conference on Advances in System Simulation, Barcelona, Spain, 23–28 October 2011.
33. Krajzewicz, D.; Hertkorn, G.; Feld, C.; Wagner, P. SUMO (Simulation of Urban MObility): An open-source traffic simulation. In Proceedings of the 4th Middle East Symposium on Simulation and Modelling, Dubai, UAE, 2–4 September 2002; pp. 183–187.
34. Krajzewicz, D.; Erdmann, J.; Behrisch, M.; Bieker, L. Recent development and applications of sumo-simulation of urban mobility. *Int. J. Adv. Syst. Meas.* **2012**, *5*, 128–138.
35. Wegener, A.; Piórkowski, M.; Raya, M.; Hellbrück, H.; Fischer, S.; Hubaux, J. TraCI: An Interface for Coupling Road Traffic and Network Simulators. In Proceedings of the 11th Communications and Networking Simulation Symposium, New York, NY, USA, 14–17 April 2008.
36. Wu, C.; Parvate, K.; Kheterpal, N.; Dickstein, L.; Mehta, A.; Vinitsky, E.; Bayen, A.M. Framework for Control and Deep Reinforcement Learning in Traffic. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC); Institute of Electrical and Electronics Engineers (IEEE): Yokohama, Japan, 2017; pp. 1–8.
37. Vinitsky, E.; Kreidieh, A.; Le Flem, L.; Kheterpal, N.; Jang, K.; Wu, F.; Liaw, R.; Liang, E.; Bayen, A.M. Benchmarks for Reinforcement Learning in Mixed-Autonomy Traffic. In Proceedings of the Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018.
38. Wu, C.; Kreidieh, A.; Parvate, K.; Vinitsky, E.; Bayen, A.M. Flow: Architecture and Benchmarking for Reinforcement Learning in Traffic Control. *arXiv* **2017**, arXiv:1710.05465.
39. Wu, C.; Kreidieh, A.; Vinitsky, E.; Bayen, A.M. Emergent behaviors in mixed-autonomy traffic. In Proceedings of the 1st Annual Conference on Robot Learning, Mountain View, CA, USA, 13–15 November 2017; Volume 78, pp. 398–407.
40. Kreidieh, A.R.; Wu, C.; Bayen, A.M. Dissipating Stop-and-Go Waves in Closed and Open Networks Via Deep Reinforcement Learning. In *Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC)*; Institute of Electrical and Electronics Engineers (IEEE): Maui, Hawaii, USA, 2018; pp. 1475–1480.
41. Treiber, M.; Kesting, A. *Traffic Flow Dynamics. Traffic Flow Dynamics: Data, Models and Simulation*; Springer Berlin Heidelberg: Berlin, Heidelberg, 2013. [CrossRef]
42. Graesser, L.; Keng, W.L. *Foundations of Deep Reinforcement Learning: Theory and Practice in Python*; Addison-Wesley Professional: Boston, MA, USA, 2019; Chapter 7.

43. Wu, C.; Kreidieh, A.; Parvate, K.; Vinitsky, E.; Bayen, A.M. Flow: A Modular Learning Framework for Autonomy in Traffic. *arXiv* **2017**, arXiv:1710.05465v2.
44. Liang, E.; Liaw, R.; Nishihara, R.; Moritz, P.; Fox, R.; Gonzalez, J.; Goldberg, K.; Stoica, I. Ray RLlib: A composable and scalable reinforcement learning library. *arXiv* **2017**, arXiv:1712.09381.
45. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI Gym. *arXiv* **2016**, arXiv:1606.01540.

【발명의 설명】

【발명의 명칭】

자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법{System and Method for Improving of Advanced Deep Reinforcement Learning Based Traffic in Non signalalized Intersections for the Multiple Self driving Vehicles}

【기술분야】

【0001】본 발명은 다수의 자율주행 차량 운행 제어에 관한 것으로, 구체적으로 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선하고 안전성을 확보할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법에 관한 것이다.

【발명의 배경이 되는 기술】

【0002】자율 주행 차량(Autonomous Vehicle)은 카메라 또는 전방물체 감지 센서를 이용하여 차선을 인식하고 자동 조향을 행하는 기술이 탑재된 차량이다. 자율 주행 차량은 카메라의 이미지 프로세싱 또는 전방물체 감지센싱을 기반으로 차선 폭, 차선상의 차량의 횡방향 위치, 양측 차선까지의 거리 및 차선의 형태, 도로의 곡률 반경이 측정되며, 이와 같이 얻어진 차량의 위치와 도로의 정보를 사용하여 차량의 주행 궤적을 추정하고, 추정된 주행 궤적을 따라 차선을 변경한다.

【0003】자율 주행 차량(Autonomous Vehicle)은 차량 전방에 장착된 카메라 또는 전방물체 감지센서에서 검출되는 선행차량의 위치 및 거리를 통하여 차량의 쓰로틀밸브, 브레이크 및 변속기를 자동 제어하여 적절한 가감속을 수행함으로써, 선행차량과 적정거리를 유지하도록 할 수도 있다.

【0004】그러나 이와 같은 자율 주행 차량(Autonomous Vehicle)이 교차로를 통과하는 경우에는 신호등의 교통신호에 따라 정차 후 출발시 선행 차량의 움직임을 감지한 다음 출발하므로 차량들 간의 출발이 지체되어 교차로에서 정체가 발생될 수 있다.

【0005】특히, 자율주행 차량과 같이 센서로부터 입력되는 정보를 이용하여 주행 환경을 파악하는 경우 비신호 교차로에서의 주행은 일반적인 도로에서의 주행보다 훨씬 어려운 과제가 된다.

【0006】한편, 무선 통신 기술의 발전으로 인하여 IoT 관련 연구가 활발히 진행되고 있으며, 그와 같이 주목 받고 있는 것이 IoV(Internet of Vehicles)이다. 차량 사이의 통신을 위해 각 차량이 노드 역할을 수행하는 무선 네트워크인 Vehicular Ad-hoc Network (VANET)은 Mobile Ad-hoc Network (MANET)의 한 형태이다.

【0007】Simulation of Urban MObility(SUMO)는 도로 상에서의 교통 네트워크를 시뮬레이션 할 수 있도록 디자인되어 있는 오픈 소스이다.

【0008】SUMO를 이용하여 도로 위에서 차량 간의 움직임을 파악함으로써 교통의 흐름을 예측할 수 있다.

【0009】이와 같은 기술들을 통하여 자율주행 차량이 주행 환경을 파악하여 비신호 교차로에서의 효율적인 주행을 위한 연구들이 이루어지고 있으나, 혼합 교통류 상황(자율주행차량과 인간운전자의 혼재)에서 자율주행차량 군집주행에 따른 비신호 교차로 통행에서는 아직도 해결하여야 하는 과제가 많다.

【0010】따라서, 자율주행차량 군집주행에 따른 비신호 교차로 통행 개선 및 안전성 확보를 위한 새로운 기술의 개발이 요구되고 있다.

【선행기술문헌】

【특허문헌】

【0011】**(특허문헌 0001)** 대한민국 공개특허 제10-2020-0071406호

(특허문헌 0002) 대한민국 공개특허 제10-2020-0058613호

(특허문헌 0003) 대한민국 공개특허 제10-2018-0065196호

【발명의 내용】

【해결하고자 하는 과제】

【0012】본 발명은 종래 기술의 자율주행 차량 운행 제어 기술의 문제점을 해결하기 위한 것으로, 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선

하고 안전성을 확보할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0013】본 발명은 실제 상황과 같이 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0014】본 발명은 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 운행 행태를 학습하는 방법으로 인공신경망에 학습을 최적화하기 위한 알고리즘인 PPO 적용으로 통행 제어를 최적화할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0015】본 발명은 SUMO(Simulation of Urban MObility)를 활용하여 실험환경을 구축하고 ACC(Adaptive Cruise Control) 시스템으로 인간운전자 정의를 하여, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0016】본 발명은 강화학습 파라미터 조정 및 자율주행차량 점유율별 운행

최적화 및 검증으로 비신호 교차로에서 완전 인간운전자환경에 비해 완전 자율주행 차량 환경에서 평균 통행 속도를 향상시킬 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0017】본 발명은 부분관찰 마르코프 의사결정과정(POMDP)에 따라 시뮬레이션 환경 내의 자율주행차량의 행태를 결정하며 평균속도를 보상으로 학습하고, 멀티 에이전트 심층강화학습을 하기 위해 PPO(Proximal Policy Optimization) 알고리즘을 적용하여 행동 결정을 최적화할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0018】본 발명은 시뮬레이션 환경에서 실제 자율주행 환경을 모사하기 위해 학습과 행동 결정의 근거를 시뮬레이션의 모든 환경이 아닌 자율주행차량 센서를 통하여 얻어진 데이터(부분만 관찰)를 기반으로 하여 행동을 결정하고 행동에 대해 강화학습의 보상을 최대화할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 제공하는데 그 목적이 있다.

【0019】본 발명의 다른 목적들은 이상에서 언급한 목적으로 제한되지 않으며, 언급되지 않은 또 다른 목적들은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

【과제의 해결 수단】

【0020】상기와 같은 목적을 달성하기 위한 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치는 SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 SUMO 시뮬레이션 실행부;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 FLOW 적용부;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축부;를 포함하는 것을 특징으로 한다.

【0021】다른 목적을 달성하기 위한 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법은 SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 SUMO 시뮬레이션 실행 단계;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 FLOW 적용 단계;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를

활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축 단계;를 포함하는 것을 특징으로 한다.

【발명의 효과】

【0022】 이상에서 설명한 바와 같은 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법은 다음과 같은 효과가 있다.

【0023】 첫째, 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선하고 안전성을 확보할 수 있도록 한다.

【0024】 둘째, 실제 상황과 같이 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 한다.

【0025】 셋째, 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 운행 행태를 학습하는 방법으로 인공신경망에 학습을 최적화하기 위한 알고리즘인 PPO 적용으로 통행 제어를 최적화할 수 있다.

【0026】 넷째, SUMO(Simulation of Urban MObility)를 활용하여 실험환경을 구축하고 ACC(Adaptive Cruise Control) 시스템으로 인간운전자 정의를 하여, SUMO

와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화할 수 있도록 한다.

【0027】 다섯째, 강화학습 파라미터 조정 및 자율주행차량 점유율별 운행 최적화 및 검증으로 비신호 교차로에서 완전 인간운전자환경에 비해 완전 자율주행차량 환경에서 평균 통행 속도를 향상시킬 수 있도록 한다.

【0028】 여섯째, 부분관찰 마르코프 의사결정과정(POMDP)에 따라 시뮬레이션 환경 내의 자율주행차량의 행태를 결정하며 평균속도를 보상으로 학습하고, 멀티 에이전트 심층강화학습을 하기 위해 PPO(Proximal Policy Optimization) 알고리즘을 적용하여 행동 결정을 최적화할 수 있도록 한다.

【0029】 일곱째, 시뮬레이션 환경에서 실제 자율주행 환경을 모사하기 위해 학습과 행동 결정의 근거를 시뮬레이션의 모든 환경이 아닌 자율주행차량 센서를 통하여 얻어진 데이터(부분만 관찰)를 기반으로 하여 행동을 결정하고 행동에 대해 강화학습의 보상을 최대화할 수 있도록 한다.

【도면의 간단한 설명】

【0030】 도 1은 본 발명에 따른 비신호 교차로에서의 심층 강화 학습 아키텍처를 나타낸 구성도

도 2는 적응형 KL 페널티 알고리즘을 사용한 PPO 알고리즘

도 3은 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서

의 강화학습기반 통행 개선을 위한 장치 구성도

도 4는 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법을 나타낸 동작 흐름도

도 5는 비신호 교차로에서의 일반적인 SUMO 시뮬레이터 구성도

도 6은 일반적인 관측 영역(Observation Space)의 일 예를 나타낸 구성도

도 7은 비신호 교차로에서의 선도 자율 주행 차량 실험 특성을 나타낸 구성도

도 8은 비신호화된 교차로에서의 실험 비교 구성도

도 9는 AV 점유율을 기반으로 한 200회 이상의 평균 보상 곡선 그래프

도 10은 비신호화된 교차로에서 AV 점유율을 통한 시공간 역학 특성 그래프

도 11은 SUMO 시뮬레이션 환경에서 평균속도, 평균 지체시간, 평균연료 소모량, 평균 배기가스 값 도출 특성 그래프

【발명을 실시하기 위한 구체적인 내용】

【0031】 이하, 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법의 바람직한 실시 예에 관하여 상세히 설명하면 다음과 같다.

【0032】 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법의 특징 및 이점들은 이하에서의 각 실시 예에 대한 상세한 설명을 통해 명백해질 것이다.

【0033】도 1은 본 발명에 따른 비신호 교차로에서의 심층 강화 학습 아키텍처를 나타낸 구성도이고, 도 2는 적응형 KL 페널티 알고리즘을 사용한 PPO 알고리즘이다.

【0034】본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법은 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선하고 안전성을 확보할 수 있도록 한 것이다.

【0035】이를 위하여, 본 발명은 실제 상황과 같이 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 하는 구성을 포함할 수 있다.

【0036】본 발명은 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 운행 행태를 학습하는 방법으로 인공신경망에 학습을 최적화하기 위한 알고리즘인 PPO 적용으로 통행 제어를 최적화할 수 있도록 하는 구성을 포함할 수 있다.

【0037】본 발명은 SUMO(Simulation of Urban MObility)를 활용하여 실험환경을 구축하고 ACC(Adaptive Cruise Control) 시스템으로 인간운전자 정의를 하여, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화할 수 있

도록 하는 구성을 포함할 수 있다.

【0038】 본 발명은 강화학습 파라미터 조정 및 자율주행차량 점유율별 운행 최적화 및 검증 구성, 부분관찰 마르코프 의사결정과정(POMDP)에 따라 시뮬레이션 환경 내의 자율주행차량의 행태를 결정하며 평균속도를 보상으로 학습하는 구성, 멀티 에이전트 심층강화학습을 하기 위해 PPO(Proximal Policy Optimization) 알고리즘을 적용하여 행동 결정을 최적화하는 구성을 포함할 수 있다.

【0039】 강화 학습(RL)은 기계 학습의 하위 영역이며 에이전트가 환경과 상호 작용하고 누적 보상을 극대화하는 조치를 학습하는 것이다.

【0040】 RL 알고리즘의 전형적인 형태는 마르코프 결정 과정(MDP)으로, 전체 관측치 집합이 주어진 적절한 동작을 결정하는 데 사용되는 강력한 프레임워크이다.

【0041】 MDP는 튜플(S, A, P, R, ρ_0, γ, T)이며, 여기서 S와 A는 각각 참가자의 상태와 행동이다. $P(S', S, a)$는 전이 확률을 정의하며, $R(a, S)$은 선택된 작용에 따라 보상을 정의하며, ρ_0은 초기 상태 분포를 정의하며, γ는 0에서 1까지의 할인 계수(discount factor)를 정의하며, T는 시간 범위를 나타낸다.

【0042】 그러나 자동화 차량은 부정확성, 의도 및 센서 노이즈를 포함하는 불확실한 환경에서 기동한다. 이 문제를 해결하기 위해 관측치 집합을 정의하는 O와 관측 함수인 Z라는 두 가지 요소를 더 사용하는 부분 관측 가능한 MDP(POMDP)가 제안되었다.

【0043】RL의 객관적 학습 에이전트는 정책 π를 최적화하여 몇 가지 타임 스텝에 걸쳐 예상 누적 할인 보상을 극대화한다.

【0044】심층 신경 네트워크(DNN)는 여러 개의 숨겨진 표현 계층으로 인해 형상 추출을 자동으로 수행할 수 있는 기능을 가지고 있다. 연속 제어기의 경우, 인공 신경 네트워크(ANN)는 복잡한 기능을 나타내기 위해 여러 개의 숨겨진 레이어를 사용하는 일반적으로 사용되는 방법이다.

【0045】이 작업에서는 MLP를 적용하여 입력 세트(상태 및 관찰)에서 출력 세트(정책)를 생성한다. 또한, DNN의 성능을 향상시키기 위해 경사 하강 최적화 방법에 기초한 PPO를 적용한다.

【0046】MLP와 RL을 융합하는 제안된 심층 RL 프레임워크는 비신호화된 교차점에서 AV의 효과를 고려하도록 설계되었다.

【0047】첫째, SUMO 시뮬레이터는 하나의 시뮬레이션 단계를 실행한다.

【0048】둘째, Flow 프레임워크는 SUMO 시뮬레이터의 상태에 대한 정보를 RL 라이브러리에 보낸다. 그런 다음, RL 라이브러리(RLlib)는 MLP를 통해 SUMO 시뮬레이터의 상태에 따라 적절한 조치를 계산한다. MLP 정책은 트래픽 데이터를 기반으로 RL 알고리즘에 대한 누적 보상을 최대화하기 위해 적용된다.

【0049】마지막으로 시뮬레이션은 RL 프로세스를 재설정하고 반복한다.

【0050】도 1은 비신호화 교차로에서 심층 강화 학습 아키텍처를 나타낸 것이다.

【0051】중요한 것은, '정책'은 환경에서의 인식과 행동 사이의 의사소통의 청사진을 가리킨다. 즉, 정책은 트래픽 시뮬레이션의 컨트롤러와 유사하다.

【0052】이 작업에서 컨트롤러는 여러 개의 숨겨진 계층이 있는 MLP 정책이다.

【0053】컨트롤러의 매개변수는 MLP 정책을 사용하여 반복적으로 업데이트되어 SUMO 시뮬레이터에서 샘플링된 트래픽 데이터를 기반으로 누적 보상을 최대화한다.

【0054】에이전트의 주요 목표는 다음과 같이 확률적 정책을 최적화하는 방법을 학습하는 것이다.

【0055】**【수학식 1】**

$$\theta^* := argmax_\theta \eta(\pi_0)$$

【0056】여기서, $\eta(\pi_0)$는 할인 계수(γ_i)와 보상(r)에 의해 계산되는 예상 누적 할인 보상이다.

【0057】**【수학식 2】**

$$\eta(\pi_0) = \sum_{i=0}^{T} \gamma_i r_i$$

【0058】종방향 역학 모델(Longitudinal Dynamic Models)을 설명하면 다음과 같다.

【0059】기본적인 차량 역학은 차량 자체와 전방 차량의 관찰에 기초하여 수동 작동 차량의 세로 방향 역학을 설명하는 차량 추종 모델에 의해 정의될 수 있다.

【0060】표준 차량 추종 모델은 다음과 같다.

【0061】**【수학식 3】**

$$a_i = f\left(h_i, \dot{h}_i, v_i\right)$$

【0062】여기서, a_i는 차량 i의 가속도이고, $f()$는 비선형 함수이며, v_i, $\dot{h}_i$ 및 h_i는 각각 차량 i의 속도, 상대 속도 및 방향이다.

【0063】본 발명에서는 운전자 행동을 묘사할 수 있는 능력으로 인해 인간 구동 차량의 세로 방향 제어를 위해 ACC 시스템의 일종인 IDM을 적용한다.

【0064】IDM은 일반적으로 사용되는 자동차 추종 모델이다.

【0065】IDM의 가속도 명령에서 비신호화된 교차로 환경에서의 차량 속도와 선도 차량의 식별(ID) 및 선도 차량의 진행(headway of the leading vehicle)은 "get" 방법으로 얻을 수 있도록 설정할 수 있다.

【0066】차량의 가속도는 다음과 같이 계산한다.

【0067】【수학식 4】

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

【0068】여기서, a_{IDM}은 차량의 가속이고, v_0는 원하는 속도이며, δ는 가속도지수, s는 차량의 앞길(앞차와의 거리)이며, $s^*(v,\Delta v)$는 원하는 방향을 나타내며, 다음과 같이 표현된다.

【0069】【수학식 5】

$$s^*(v,\Delta v) = s_0 + max\left(0, vT + \frac{v\Delta v}{2\sqrt{ab}}\right)$$

【0070】여기서, S_0는 최소 갭을, T는 a시간 갭을, Δv는 선두 차량과 비교한 속도 차이(현재 속도 - 선두 속도), a는 가속 구간, b는 편안한 감속을 나타낸다.

【0071】도시 교통에 대한 IDM 컨트롤러의 대표적인 매개변수는 표 1에서와 같다.

【0072】【표 1】

Parameters	Value
Desired speed (m/s)	15
Time gap (s)	1.0
Minimum gap (m)	2.0
Acceleration exponent	4.0
Acceleration (m/s^2)	1.0
Comfortable acceleration (m/s^2)	1.5

【0073】정책 최적화(Policy Optimization)를 설명하면 다음과 같다.

【0074】정책 경사 방법(Policy gradient methods)은 동작 값이나 상태 값 함수가 아닌 경사 강하 알고리즘을 사용하여 매개 변수화된 정책 함수의 추정기를 계산하려고 한다.

【0075】따라서 비선형 근사 및 부분 관측으로 인해 추정 함수에 발생하는 수렴 문제를 피한다.

【0076】본 발명은 비신호화된 교차로의 시뮬레이션에서 제어 정책을 직접 최적화하기 위해 MLP 정책을 적용한다. 정책 행동($\log \pi_\theta$)의 확률에 대한 기대치와 시간 스텝 $t^{(\hat{A}_t)}$에서의 어드밴티지 함수(advantage function)의 추정치에 기초하는 정책 경사법은 다음과 같이 표현된다.

【0077】【수학식 6】

$$\hat{g} = \hat{E}_t\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right]$$

【0078】여기서, $\hat{E}_t[.]$는 유한한 표본 배치에 대한 기대 연산자이며, π_θ는 확률적 정책을 나타내며, $\hat{A}_t$는 디스카운트된 보상 합계와 기준 추정치로 정의되며, a_t와 s_t는 시간 스텝 t의 행동과 상태를 각각 나타낸다.

【0079】슐만(Schulman) 등에 의해 제안된 PPO는 RLlib 라이브러리에서 제공하는 간단한 TRPO이다.

【0080】즉, PPO의 목표는 TRPO와 동일하며, TRPO는 신뢰 지역 제약 조건을 사용하여 새 정책이 이전 정책에서 너무 멀리 있지 않도록 정책을 업데이트하도록 강제한다.

【0081】PPO에는 적응형 쿨백-라이블러(adaptive Kullback-Leibler;KL) 페널티와 클리핑 목표(clipped objective)의 두 가지 유형이 있다.

【0082】PPO는 대리 손실 함수를 채택하여 정책 업데이트를 생성한다. 이 프로세스는 훈련 과정 중 성능 저하를 방지한다.

【0083】대리 객체(J^{CPI})는 다음과 같이 설명된다.

【0084】【수학식 7】

$$J^{CPI}(\theta) = \hat{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right]$$

【0085】여기서, $\pi_{\theta_{old}}$는 업데이트 전 정책 매개 변수, π_θ는 업데이트 후 정책 매개 변수, $r_t(\theta)$는 확률비를 나타낸다.

【0086】연속 행동의 경우 PPO의 정책 출력은 각 행동에 대한 가우스 분포의 매개 변수이다.

【0087】그런 다음 정책은 이러한 분포를 기반으로 연속 출력을 생성한다.

【0088】본 발명에서 적응형 KL 패널티를 가진 PPO는 다음과 같이 미니 배치(minibatch) 확률적 경사 하강(SGD)을 사용하여 KL 페널티 목표를 최적화하는 데 사용된다.

【0089】【수학식 8】

$$\underset{\theta}{\text{maximize}}\ \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] - \beta\hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]$$

【0090】【수학식 9】

$$\text{Subject to } \hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right] \leq \delta$$

【0091】여기서, β는 매 정책 업데이트 후 업데이트되는 가중 조절 계수(weight control coefficient)이다.

【0092】현재 KL 차이가 목표 KL 편차보다 클 경우 증가되고, 현재 KL 발산이 목표 KL 발산보다 작으면 감소한다.

【0093】PPO 알고리즘에서는 먼저 현재 정책이 환경과 상호 작용하여 에피소드 시퀀스를 생성한다. 다음으로, 어드밴티지 함수(advantage function)는 상태 값에 대한 기준 추정치를 사용하여 추정된다.

【0094】마지막으로, 모든 경험을 수집하고 정책 네트워크를 통해 경사 하강 알고리즘을 실행한다. 적응형 KL 페널티 알고리즘의 전체 PPO는 도 2의 알고리즘 1의 유사 코드로 표시된다.

【0095】도 3은 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 구성도이다.

【0096】본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치는 SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어

진 데이터를 FLOW 적용부(200)로 전달하는 SUMO 시뮬레이션 실행부(100)와, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부(300)로 전달하는 FLOW 적용부(200)와, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축부(300)를 포함한다.

【0097】여기서, SUMO 시뮬레이션 실행부(100)는 SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하여 SUMO 시뮬레이션을 수행하는 SUMO 시뮬레이션부(10)와, 배기가스, 속도 및 위치값 파일을 생성하여 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부(200)로 전달하는 결과 파일생성부(11)를 포함한다.

【0098】그리고 FLOW 적용부(200)는 인간운전자 정의, 심층 강화학습 입력값 설정 및 차량의 속도, 가속도, 출발점 등 시뮬레이션 환경 설정을 하는 시뮬레이션 초기화부(20)와, FLOW 환경 구축을 하여 상태(state)를 강화학습 라이브러리로 전달하는 FLOW 환경 구축부(21)와, 강화학습을 적용하지 않은 운전 행태 도출을 하는 운행행태 도출부(22)와, 차량 제어를 하고 제어 정보를 SUMO 시뮬레이션 실행부(100)로 전달하는 차량 제어 모듈(23)과, SUMO 시뮬레이션 실행부(100)로부터 시뮬레이션 상태를 받아 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부(300)로 전달하는 업데이트부(24)를 포함한다.

【0099】그리고 강화학습 라이브러리 환경 구축부(300)는 FLOW 적용부(200)로부터 상태(state)를 전달받는 강화학습 라이브러리(31)와, 학습할 데이터를 샘플링하는 데이터 샘플링부(32)와, 운전 행태(정책) 훈련을 하는 정책 훈련부(33)와, 훈련 결과를 평가하고 학습된 행태(주행방법)를 FLOW 적용부(200)로 전달하는 훈련 결과 평가부(34)와, 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 하는 정책 최적화부(35)와, FLOW 적용부(200)로부터 자율 주행 차량의 상태를 받아 정책 업데이트 및 저장을 하는 정책 업데이트 저장부(36)와, 업데이트된 정책이 학습 루프 조건을 만족하는지 판단하는 학습 루프조건 판단부(37)를 포함한다.

【0100】도 4는 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법을 나타낸 동작 흐름도이다.

【0101】본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법은 SUMO(Simulation of Urban MObility)을 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부(200)로 전달하는 SUMO 시뮬레이션 실행 단계와, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부(300)로 전달하는 FLOW 적용 단계와, SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이

전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축 단계를 포함한다.

【0102】여기서, SUMO 시뮬레이션 실행 단계는 SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하여 SUMO 시뮬레이션을 수행하는 SUMO 시뮬레이션 단계(S409)와, 배기가스, 속도 및 위치값 파일을 생성하여 자율주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부(200)로 전달하는 결과 파일생성 단계(S410)를 포함한다.

【0103】그리고 FLOW 적용 단계는 인간운전자 정의, 심층 강화학습 입력값 설정 및 차량의 속도, 가속도, 출발점 등 시뮬레이션 환경 설정을 하는 시뮬레이션 초기화 단계(S401)와, FLOW 환경 구축을 하여 상태(state)를 강화학습 라이브러리로 전달하는 FLOW 환경 구축 단계(S402)와, 강화학습을 적용하지 않은 운전 행태 도출을 하는 운행행태 도출 단계(S403)와, 차량 제어를 하고 제어 정보를 SUMO 시뮬레이션 실행부(100)로 전달하는 차량 제어 단계(S408)와, SUMO 시뮬레이션 실행부(100)로부터 시뮬레이션 상태를 받아 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부(300)로 전달하는 업데이트 단계(S411)를 포함한다.

【0104】그리고 강화학습 라이브러리 환경 구축 단계는 강화학습 라이브러리(31)가 FLOW 적용부(200)로부터 상태(state)를 전달받는 단계(S404)와, 학습할 데이터를 샘플링하는 데이터 샘플링 단계(S405)와, 운전 행태(정책) 훈련을 하는 정책 훈련 단계(S406)와, 훈련 결과를 평가하고 학습된 행태(주행방법)를 FLOW 적용

부(200)로 전달하는 훈련 결과 평가 단계(S407)와, 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 사용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 하는 정책 최적화 단계(S412)와, FLOW 적용부(200)로부터 자율 주행 차량의 상태를 받아 정책 업데이트 및 저장을 하는 정책 업데이트 저장 단계(S413)와, 업데이트된 정책이 학습 루프 조건을 만족하는지 판단하는 학습 루프조건 판단 단계(S414)를 포함한다.

【0105】 도 5는 비신호 교차로에서의 일반적인 SUMO 시뮬레이터 구성도이다.

【0106】 독일 항공우주센터의 교통 시스템 연구소가 개발한 SUMO는 오픈소스 마이크로스코픽 교통 시뮬레이터이다. SUMO는 신호등, 차량, 보행자 및 대중 교통과 함께 도시 규모의 교통 네트워크를 시뮬레이션할 수 있다. 또한 TraCI는 SUMO 시뮬레이터에 심층 RL을 적용하기 위해 SUMO를 Python에 연결할 수 있도록 한다.

【0107】 비신호화된 교차로에서 일반적인 SUMO 시뮬레이터는 도 5에서와 같다.

【0108】 UC Berkeley에서 개발한 Flow는 심층 RL 알고리즘과 맞춤형 도로 네트워크 간의 인터페이스를 제공한다. 또한 Flow는 훈련 정책을 분석하고 검증할 수 있다.

【0109】 Flow의 장점은 심층 RL을 통해 자율 주행 차량의 제어기를 개선하기 위해 다양한 도로망을 쉽게 구현할 수 있는 능력을 포함한다. Flow에서 사용자 지

정 환경은 다양한 시나리오에 대한 초기화된 시뮬레이션, 관찰 공간, 상태 공간, 작업 공간, 제어기 및 보상 기능을 포함한 주요 부분 집합 클래스를 생성하는 데 사용될 수 있다.

【0110】초기화된 시뮬레이션은 시작 에피소드에 대한 시뮬레이션 환경의 초기 설정을 나타낸다.

【0111】본 발명에서는 IDM 규칙과 심층 RL 프레임워크의 매개 변수뿐만 아니라 위치, 속도, 가속, 출발점, 궤적 및 차량 수를 설정한다.

【0112】특히, 모든 차량의 궤적은 특정 노드(네트워크의 포인트 위치), 특정 에지(노드를 함께 연결) 및 특정 경로(에지 차량이 통과하는 시퀀스)를 포함하여 SUMO 시뮬레이터에 의해 초기 시뮬레이션 프로세스에서 설정된다.

【0113】다음으로, 인간 운전 차량의 가속은 SUMO 시뮬레이터에 의해 제어되고 AV의 가속은 Rllib 라이브러리에 의해 제어된다.

【0114】도 6은 일반적인 관측 영역(Observation Space)의 일 예를 나타낸 구성도이다.

【0115】관측 공간은 AV 속도(자기 차량 속도), AV 위치(자기 차량 위치) 및 해당 선행 및 AV의 속도 및 범퍼 투 범퍼 헤드웨이와 같은 관측 가능한 형상의 수와 유형을 나타낸다.

【0116】관찰 가능한 출력이 상태 공간으로 공급되어 적절한 정책을 예측한다.

【0117】그리고 상태 공간(state space)은 AV의 위치 및 속도뿐만 아니라 이전 및 이후의 AV를 포함하여 관찰 공간을 기반으로 하는 자율 에이전트 및 주변 차량의 벡터를 나타낸다.

【0118】환경 내의 기능은 get_state 방법을 사용하여 추출되어 정책에 공급된다.

【0119】첫째, 비신호화된 교차로에서 모든 차량의 ID를 얻는다. 그런 다음 모든 차량의 위치와 속도를 파악하여 상태 공간을 생성한다.

【0120】중요한 것은 현재 위치가 미리 지정된 시작 지점을 기반으로 한다는 것이다.

【0121】상태 공간은 다음과 같이 정의된다.

【0122】**【수학식 10】**

$$S = \begin{pmatrix} x_0 \\ v_0 \\ v_l \\ d_l \\ v_f \\ d_f \end{pmatrix}$$

【0123】여기서, S는 특정 차량의 상태이고, x0은 AV의 해당 좌표이고, v_0, v_l 및 v_f는 각각 AV, 이전 AV 및 다음 AV의 해당 속도이고,

【0124】d_l와 d_f는 각각 이전 AV와 다음 AV의 범퍼-대-범퍼 헤드웨이이다.

【0125】 행동 공간(Action Space)은 OpenAI gym에서 제공하는 트래픽 환경에서 자율 에이전트의 행동을 나타낸다.

【0126】 자동화 차량의 표준 행동은 가속이고, 행동 공간에서 행동의 범위는 최대 감속부터 최대 가속까지이다.

【0127】 apply_RL_ 행동 함수는 SUMO 시뮬레이터에서 특정 명령을 실제 행동으로 변환하기 위해 적용된다.

【0128】 첫째, 비신호화된 교차로에서 모든 AV를 식별한다. 그런 다음 행동 명령은 기본 환경 방법을 사용하여 가속으로 변환된다.

【0129】 컨트롤러는 사람이 운전하는 차량과 AV를 포함하여 행위자들의 행동을 통제한다. 공유 제어를 사용하여 단일 컨트롤러를 여러 행위자에 적용할 수 있다. 본 발명에서는 인간이 운전하는 차량은 플로우 프레임워크에 의해 제어되고, 자동화 차량은 RLlib 라이브러리에 의해 제어된다.

【0130】 보상 함수(Reward Function)를 설명하면 다음과 같다.

【0131】 트래픽 정체를 줄이기 위해서는 지연 시간, 대기열 길이를 줄임으로써 네트워크의 평균 속도를 최적화해야 한다. 따라서, 평균 속도는 현실에서 심층 RL 정책을 훈련하는 유망한 측정 기준이 된다.

【0132】 보상 함수는 자율 에이전트가 정책을 최적화하는 방법을 정의한다.

【0133】 본 발명에서 RL 에이전트의 목표는 비신호화된 교차로에서 차량 간 충돌을 억제하는 동시에 높은 평균 속도를 얻는다.

【0134】본 발명에서, L2 규범은 목표 속도(비신호 교차로에서 모든 차량의 원하는 속도)에 기초하여 비신호 교차로에서 주어진 차량 속도에 주어진 양의 거리를 추정하는 데 사용된다.

【0135】특히, 비신호화된 교차로에서 모든 차량의 현재 속도를 구한 다음 평균 속도를 보상으로 돌려주는 Get-speed 방법을 적용한다.

【0136】보상 함수는 수학식 11에서와 같이 표현된다.

【0137】**【수학식 11】**

$$r_t := max\left(\|v_{des}\cdot\mathbb{I}^k\|_2 - \|v_{des} - v\|_2, 0\right)/\|v_{des}\cdot\mathbb{I}^k\|_2$$

【0138】여기서, v_{des}는 임의의 원하는 속도를 나타내고 $v \in R^k$는 비신호화된 교차로에서 모든 차량의 속도를 나타낸다.

【0139】본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법을 이용한 시뮬레이션 환경 설정 및 결과를 설명하면 다음과 같다.

【0140】도 7은 비신호 교차로에서의 선도 자율 주행 차량 실험 특성을 나타낸 구성도이고, 도 8은 비신호화된 교차로에서의 실험 비교 구성도이다.

【0141】표 2는 비신호 교차로 시뮬레이션 설정값의 일 예를 나타낸 것이다.

【0142】【표 2】

시뮬레이션 설정 내용	설정값
학습 횟수	200
학습 1회의 학습 시간	6000
보상 조정값(PPO 조정값, Gamma)	0.99
보상 조정값(PPO 조정값, Lamda)	0.95
심층 신경망의 형태	256*256*256
정책 조정값 Kullback–Leibler (KL)	0.01
최적화 방법 SGD 횟수	10
자율주행차량의 비율	0%, 10%, 20% ... 100%

【0143】시뮬레이션 시나리오는 다음과 같다.

【0144】본 발명에서 비신호 교차로를 횡단한 차량은 SUMO 시뮬레이터가 제공하는 선로설비 규칙(right-of-way rule)을 따랐다. 선로설비 규칙의 목적은 교통 규칙을 시행하고 교통 충돌을 방지하는 것이다.

【0145】또한 모든 차량의 위치를 관찰하고 POMDP에서 MDP으로 환경을 전환했다. 중요한 것은, 자율 에이전트는 RLlib 라이브러리를 사용하여 롤아웃에 대한 특정 보상을 최적화하는 방법을 학습한다. 시뮬레이션은 RL 에이전트를 사용하여 인간 운전 주행과 혼합 자율 주행에서 전체 주행 흐름을 나타낸다.

【0146】RL 에이전트는 업데이트된 상태를 수신하고 0.1초의 시간 단계에서 새 상태를 가져오고, 인간 운전 차량의 경우 가속 동작은 IDM 모델에 의해 제어된다. 또한, 연속 라우팅은 네트워크 내에서 차량을 유지하기 위해 적용된다.

【0147】0.1초의 시간 스텝, 3.2m의 차선 폭, 각 방향으로 2차선, 420m의 차선 길이, 최대 가속도 $3m/s^2$, 최소 가속도 $-3m/s^2$, 최대 속도 12m/s, 600의 시야, 훈련 과정에 대한 200회의 반복으로 시뮬레이션 실험을 수행했다.

【0148】각 방향으로 시간당 1000대의 차량이 유입되도록 설정하고, 비신호 교차로의 범위는 200m에서 220m 사이였다.

【0149】현장에서 다양한 시나리오를 시뮬레이션해야 하는데, 본 발명에서는 비신호화된 교차로에서 선도적인 자율 주행 차량의 효과로 초점을 제한했다.

【0150】군집 차량은 비신호 교차로에 접근하여 네 가지 다른 방향을 따라 직진 주행한다. 또한 1% ~ 100%의 AV 보급률에 대한 결과를 10% 단위로 제시하고, 비신호 교차로에서 모든 차량에 대해 차선 변경과 좌회전을 무시한다.

【0151】도 7의 (a)는 10% ~ 90% 범위의 자율 주행(AV) 점유율을 가진 혼합 교통 상황에서의 비신호화된 교차로에서 선도 자율 주행 차량 실험 환경이고, (b)는 100% 자율 주행(AV) 점유율의 실험 환경이다.

【0152】선도적인 자율 주행 차량 실험의 우수성을 입증하기 위해 선도적인 자율 주행 차량 실험을 선도적인 인간 주도 차량 실험과 모든 인간 주도 차량 실험을 포함한 다른 실험과 비교했다. 도 8은 비신호화된 교차로에서 실험의 비교를 보

여준다.

【0153】 도 9는 AV 점유율을 기반으로 한 200회 이상의 평균 보상 곡선 그래프이다.

【0154】 훈련 정책의 성능(Training Policy's Performance)은 다음과 같다.

【0155】 AV 점유율을 통한 RL 훈련 성과는 학습 성과를 평가하기 위해 사용되었다. 도 9는 AV 점유율을 기반으로 한 200회 이상의 평균 보상 곡선을 나타낸 것이다.

【0156】 모든 상황에서 곡선이 평평해졌다는 것은 교육 정책이 거의 융합되었음을 나타낸다. 또한, 비신호 교차로의 AV 점유율이 50% AV 점유율을 제외하고 증가함에 따라 평균 보상이 증가했다. 완전 자율 주행은 다른 AV 점유율을 능가했으며, 가장 높은 평균 보상과 상당한 곡선 평탄화를 초래했다. 특히, 전체 자율 주행은 10% AV 점유율에 비해 6.8배 향상되었다.

【0157】 따라서 전체 자율 주행은 모든 상황에서 다른 AV 점유율을 능가했고, 비신호화된 교차로에서 선도적인 자율 주행 차량 실험의 효과는 AV 점유율이 증가함에 따라 더욱 분명해졌다.

【0158】 도 10은 비신호화된 교차로에서 AV 점유율을 통한 시공간 역학 특성 그래프이다.

【0159】 선도적 자율 주행 차량이 부드러운 주행 속도에 미치는 영향은 다음과 같다.

【0160】도 10에서 점(point)은 속도에 따라 색상으로 구분되고, 맨 위에 가까운 점은 원활한 교통을 나타낸다. 이와는 대조적으로, 바닥에 가까운 지점은 혼잡한 교통량을 나타낸다.

【0161】낮은 AV 점유율의 경우, 사람이 운전하는 차량 거동의 정지 및 이동 파동으로 인해 교란이 발생하여 비신호화된 교차로 영역(200m에서 220m 범위)의 속도가 감소했다. 도 10에서와 같이, AV 점유율이 낮은 비신호화된 교차로에서 거의 모든 지점이 바닥에 근접해 있다.

【0162】이는 인간이 운전하는 차량이 비신호화된 교차로 구역에 동시에 접근하고 선로설비 규칙에 따라 속도를 늦추기 때문이다. 높은 AV 점유율에서 포인트는 상단에 가깝고, AV는 더 짧은 시간 내에 느려지며, 따라서 비신호화된 교차로에서 정지 및 이동 파동이 점점 더 적어진다.

【0163】전체 자율 주행은 모든 AV 점유율 중 가장 높은 부드러운 주행 속도를 달성했다. 따라서, 교통 체증이 부분적으로 해소되었고, AV 점유율이 증가함에 따라 교통 흐름이 원활해졌다.

【0164】도 11은 SUMO 시뮬레이션 환경에서 평균속도, 평균 지체시간, 평균 연료 소모량, 평균 배기가스 값 도출 특성 그래프이다.

【0165】도 11은 평균 속도, 지연 시간, 연료 소비량 및 AV 점유율에 따른 배출량 측면에서 MOE 평가를 나타낸 것으로, MOE 평가 결과는 AV 점유율이 증가함에 따라 시뮬레이션이 더욱 효과적이었음을 나타낸다.

【0166】이동성과 관련하여, 평균 속도는 AV 점유율이 증가함에 따라 점차적으로 증가하였고 지연 시간은 점차 감소하였다.

【0167】도11의 (a)(b)에서와 같이, 완전 자율 주행은 10% AV 점유율에 비해 평균 속도가 1.19배, 지연 시간은 1.76배 향상되었다. 에너지 효율, 연료 소비 및 배출량은 AV 보급률이 증가함에 따라 약간 감소했다.

【0168】도 11의 (c)(d)에서와 같이, 완전 자율 주행은 10% AV 점유율에 비해 연료 소비량이 1.05배, 배기 가스 배출량이 1.22배 향상되었다.

【0169】따라서, 선도적인 자율 주행 차량은 AV 점유율이 증가할 때 이동성과 에너지 효율 측면에서 더 효과적인 것을 확인할 수 있다.

【0170】이상에서 설명한 본 발명에 따른 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법은 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선하고 안전성을 확보할 수 있도록 한 것이다.

【0171】본 발명은 실제 상황과 같이 자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 한 것이다.

【0172】이상에서의 설명에서와 같이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 본 발명이 구현되어 있음을 이해할 수 있을 것이다.

【0173】그러므로 명시된 실시 예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 하고, 본 발명의 범위는 전술한 설명이 아니라 특허청구 범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

【부호의 설명】

【0174】100. SUMO 시뮬레이션 실행부

200. FLOW 적용부

300. 강화학습 라이브러리 환경 구축부

【청구범위】

【청구항 1】

SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 SUMO 시뮬레이션 실행부;

SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 FLOW 적용부;

SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축부;를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 2】

제 1 항에 있어서, SUMO 시뮬레이션 실행부는,

SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하여 SUMO 시뮬레이션을 수행하는 SUMO 시뮬레이션부와,

배기가스, 속도 및 위치값 파일을 생성하여 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 결과 파일생성부를 포함하는 것을

특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 3】

제 1 항에 있어서, SUMO 시뮬레이션 실행부는 군집 차량이 비신호 교차로에 접근하여 네 가지 다른 방향을 따라 직진 주행하는 상황에서,

1% ~ 100%의 AV 보급률에 대한 결과를 10% 단위로 제시하고, 비신호 교차로에서 모든 차량에 대해 차선 변경과 좌회전을 무시하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 4】

제 1 항에 있어서, FLOW 적용부는,

인간운전자 정의, 심층 강화학습 입력값 설정 및 차량의 속도, 가속도, 출발점을 포함하는 시뮬레이션 환경 설정을 하는 시뮬레이션 초기화부와,

FLOW 환경 구축을 하여 상태(state)를 강화학습 라이브러리로 전달하는 FLOW 환경 구축부와,

강화학습을 적용하지 않은 운전 행태 도출을 하는 운행행태 도출부와,

차량 제어를 하고 제어 정보를 SUMO 시뮬레이션 실행부로 전달하는 차량 제어 모듈과,

SUMO 시뮬레이션 실행부로부터 시뮬레이션 상태를 받아 시뮬레이션 상태 업

데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 업데이트부를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 5】

제 1 항에 있어서, 강화학습 라이브러리 환경 구축부는,

FLOW 적용부로부터 상태(state)를 전달받는 강화학습 라이브러리와,

학습할 데이터를 샘플링하는 데이터 샘플링부와,

운전 행태(정책) 훈련을 하는 정책 훈련부와,

훈련 결과를 평가하고 학습된 행태를 FLOW 적용부로 전달하는 훈련 결과 평가부와,

자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 하는 정책 최적화부를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 6】

제 5 항에 있어서, 강화학습 라이브러리 환경 구축부는,

FLOW 적용부로부터 자율 주행 차량의 상태를 받아 정책 업데이트 및 저장을 하는 정책 업데이트 저장부와,

업데이트된 정책이 학습 루프 조건을 만족하는지 판단하는 학습 루프조건 판단부를 더 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 7】

제 5 항에 있어서, 강화학습 라이브러리 환경 구축부는,

정책 최적화(Policy Optimization)를 위하여 동작 값이나 상태 값 함수가 아닌 경사 강하 알고리즘을 사용하여 매개 변수화된 정책 함수의 추정기를 계산하는 정책 경사 방법(Policy gradient methods)을 적용하여,

비선형 근사 및 부분 관측으로 인해 추정 함수에 발생하는 수렴 문제를 피하도록 하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 8】

제 7 항에 있어서, 강화학습 라이브러리 환경 구축부는,

비신호화된 교차로의 시뮬레이션에서 제어 정책을 직접 최적화하기 위해 MLP(multilayer perceptron)정책을 적용하고,

정책 행동($\log \pi_\theta$)의 확률에 대한 기대치와 시간 스텝 t $(\hat{A}_t)$에서의 어드밴티지 함수(advantage function)의 추정치에 기초하는 정책 경사법은,

$$\hat{g} = \hat{E}_t\left[\nabla_\theta \log \pi_\theta(a_t|s_t)\hat{A}_t\right]$$ 으로 정의하고,

여기서, $\hat{E}_t[.]$는 유한한 표본 배치에 대한 기대 연산자, π_θ는 확률적 정책, $\hat{A}_t$는 디스카운트된 보상 합계와 기준 추정치로 정의되며, a_t와 s_t는 시간 스텝 t의 행동과 상태인 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 9】

제 5 항에 있어서, 강화학습 라이브러리 환경 구축부는,

훈련 과정 중 성능 저하를 방지하기 위하여 대리 손실 함수를 채택하여 정책 업데이트를 생성하는 PPO(Proximal policy optimization)를 적용하고,

대리 객체(J^{CPI})는 $$J^{CPI}(\theta) = \hat{E}_t\left[\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\hat{A}_t\right] = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right]$$ 으로 정의되고,

$\pi_{\theta_{old}}$는 업데이트 전 정책 매개 변수, π_θ는 업데이트 후 정책 매개 변수, $r_t(\theta)$는 확률비인 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 10】

제 9 항에 있어서, 연속 행동의 경우 PPO의 정책 출력은 각 행동에 대한 가우스 분포의 매개 변수이고,

적응형 KL 패널티를 가진 PPO는 미니 배치(minibatch) 확률적 경사 하강

(SGD)을 사용하여 KL 페널티 목표를 최적화하는 데 사용되고,

$$\underset{\theta}{\text{maximize}}\ \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] - \beta\hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right],$$

$$\text{Subject to } \hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right] \leq \delta$$

여기서, β는 매 정책 업데이트 후 업데이트되는 가중 조절 계수(weight control coefficient)인 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 11】

제 10 항에 있어서, 현재 KL 차이가 목표 KL 편차보다 클 경우 증가되고, 현재 KL 발산이 목표 KL 발산보다 작으면 감소되고,

PPO 알고리즘에서 먼저 현재 정책이 환경과 상호 작용하여 에피소드 시퀀스를 생성하고, 어드밴티지 함수(advantage function)는 상태 값에 대한 기준 추정치를 사용하여 추정되어 모든 경험을 수집하고 정책 네트워크를 통해 경사 하강 알고리즘을 실행하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치.

【청구항 12】

SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 SUMO 시뮬레이션 실행 단계;

SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 FLOW 적용 단계;

SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축 단계;를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법.

【청구항 13】

제 12 항에 있어서, SUMO 시뮬레이션 실행 단계는,

SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하여 SUMO 시뮬레이션을 수행하는 SUMO 시뮬레이션 단계와,

배기가스, 속도 및 위치값 파일을 생성하여 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 결과 파일생성 단계를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법.

【청구항 14】

제 12 항에 있어서, FLOW 적용 단계는,

인간운전자 정의, 심층 강화학습 입력값 설정 및 차량의 속도, 가속도, 출발

점 등 시뮬레이션 환경 설정을 하는 시뮬레이션 초기화 단계와,

FLOW 환경 구축을 하여 상태(state)를 강화학습 라이브러리로 전달하는 FLOW 환경 구축 단계와,

강화학습을 적용하지 않은 운전 행태 도출을 하는 운행행태 도출 단계와,

차량 제어를 하고 제어 정보를 SUMO 시뮬레이션 실행부로 전달하는 차량 제어 단계와,

SUMO 시뮬레이션 실행부로부터 시뮬레이션 상태를 받아 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 업데이트 단계를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법.

【청구항 15】

제 12 항에 있어서, 강화학습 라이브러리 환경 구축 단계는,

강화학습 라이브러리가 FLOW 적용부로부터 상태(state)를 전달받는 단계와,

학습할 데이터를 샘플링하는 데이터 샘플링 단계와,

운전 행태(정책) 훈련을 하는 정책 훈련 단계와,

훈련 결과를 평가하고 학습된 행태를 FLOW 적용부로 전달하는 훈련 결과 평가 단계와,

자율주행차량이 관찰할 수 있는 범위 내의 정보를 통하여 학습하는 방법으로 강화학습과 마르코프 의사결정 모델 사용(Partial Observability MDP, POMDP)을 적

용하여 행동에 대한 강화학습의 보상을 최대화할 수 있도록 하는 정책 최적화 단계와,

FLOW 적용부로부터 자율 주행 차량의 상태를 받아 정책 업데이트 및 저장을 하는 정책 업데이트 저장 단계와,

업데이트된 정책이 학습 루프 조건을 만족하는지 판단하는 학습 루프조건 판단 단계를 포함하는 것을 특징으로 하는 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 방법.

【요약서】

【요약】

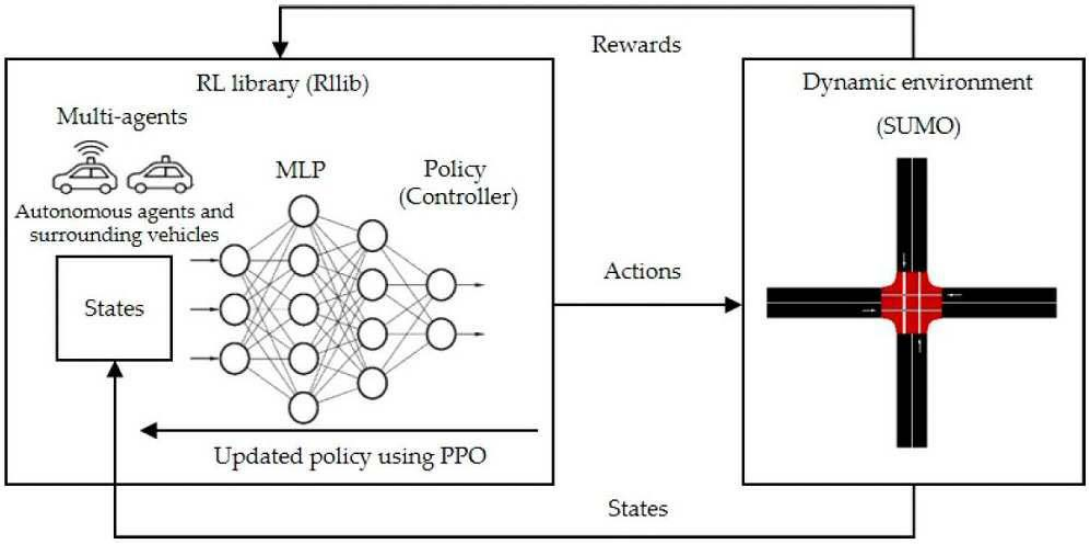
본 발명은 군집된 자율주행차량과 인간운전자 차량이 혼재되어 있는 혼합 교통류 상황에서 자율주행차량 군집주행 학습으로 비신호 교차로 통행을 개선하고 안전성을 확보할 수 있도록 한 자율주행 차량 군집 운행을 위한 비신호 교차로에서의 강화학습기반 통행 개선을 위한 장치 및 방법에 관한 것으로, SUMO(Simulation of Urban MObility)를 활용하여 시뮬레이션 환경을 구축하고, 자율 주행 차량의 속도,위치,센서에서 얻어진 데이터를 FLOW 적용부로 전달하는 SUMO 시뮬레이션 실행부;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW 환경에서 시뮬레이션 환경을 구축하고, 강화학습을 적용하지 않은 운전 행태 도출, 차량 제어 및 시뮬레이션 상태 업데이트를 하여 상태 및 보상 정보를 강화학습 라이브러리 환경 구축부로 전달하는 FLOW 적용부;SUMO와 연동할 수 있는 강화학습 플랫폼 FLOW를 활용하여 멀티 에이전트 심층강화학습(Multi agent Deep Reinforcement Learning)으로 통행 제어를 최적화하는 강화학습 라이브러리 환경 구축부;를 포함하고, SUMO 시뮬레이션 실행부는 군집 차량이 비신호 교차로에 접근하여 네 가지 다른 방향을 따라 직진 주행하는 상황에서, 1% ~ 100%의 AV 보급률에 대한 결과를 10% 단위로 제시하고, 비신호 교차로에서 모든 차량에 대해 차선 변경과 좌회전을 무시한다.

【대표도】

도 4

【도면】

【도 1】



【도 2】

Algorithm 1 PPO with an Adaptive KL Penalty Algorithm

1: Initial policy parameters θ_0, weight control β_0, target KL-divergence δ_{tag}
2: For k = 0, 1, 2 … do
3: Gather set of trajectories on policy $\pi_k = \pi(\theta_k)$
4: Optimize the KL penalized using minibatch SGD

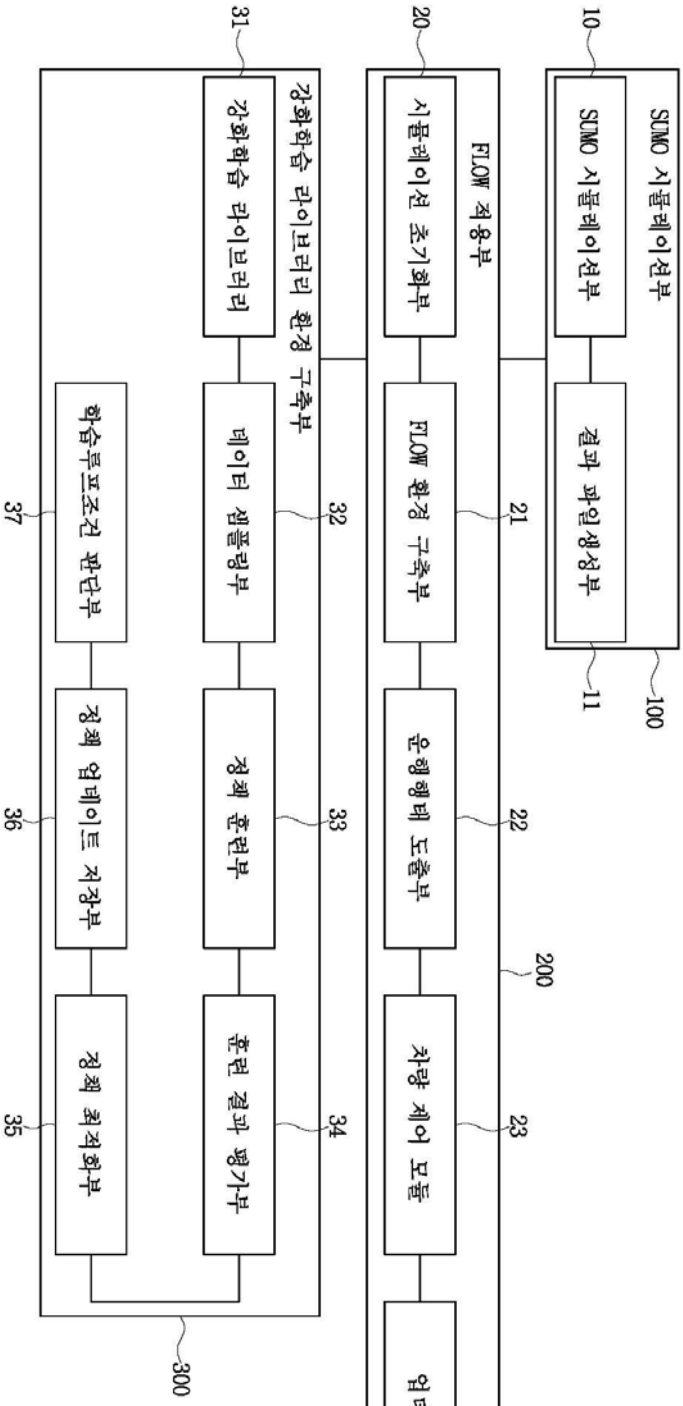
$$J^{KLPEN}(\theta) = \hat{E}_t\left[r_t(\theta)\hat{A}_t\right] - \beta\hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]$$

5: Compute KL divergence between the new and old policy

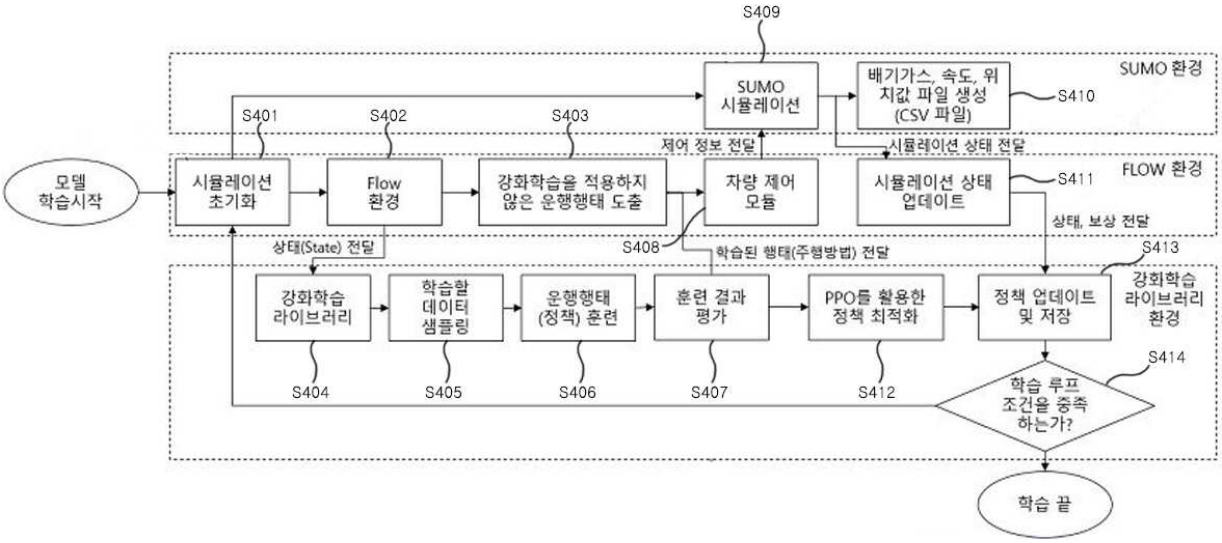
$$\delta = \hat{E}_t\left[KL\left[\pi_{\theta_{old}}(a_t|s_t), \pi_\theta(a_t|s_t)\right]\right]$$

6: If $\delta > 1.5\,\delta_{tag}$ then
$\beta_{k+1} = 2\beta_k$
7: Else if $\delta < \delta_{tag}/1.5$ then
$\beta_{k+1} = \beta_k/2$
8: Else
pass
9: End if
10: End for

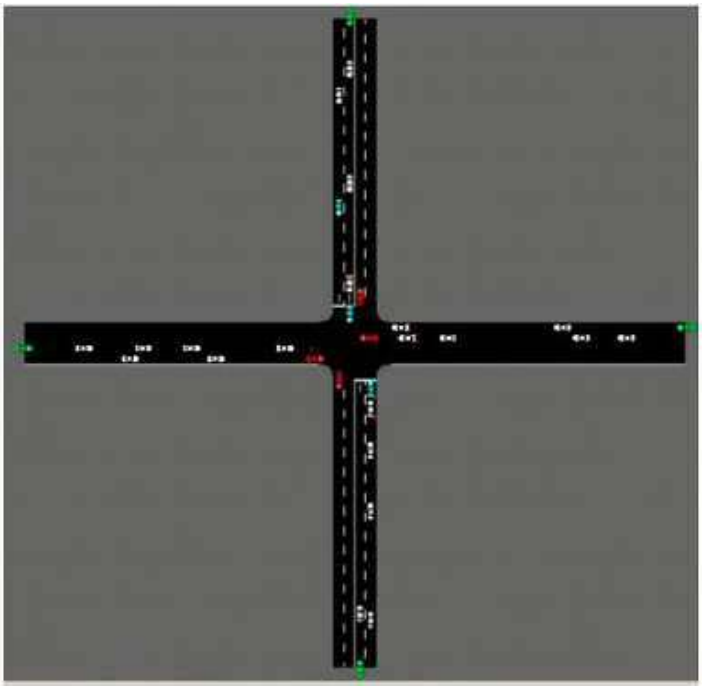
【도 3】


SUMO 시뮬레이션부
SUMO 시뮬레이션부
결과 파일생성부
10
11
100
FLOW 적용부
시뮬레이션 초기화부
FLOW 환경 구축부
운행행태 도출부
차량 제어 모듈
업데이트부
20
21
22
23
24
200
강화학습 라이브러리 환경 구축부
강화학습 라이브러리
데이터 샘플링부
정책 훈련부
훈련 결과 평가부
31
32
33
34
학습루프조건 판단부
정책 업데이트 저장부
정책 최적화부
37
36
35
300

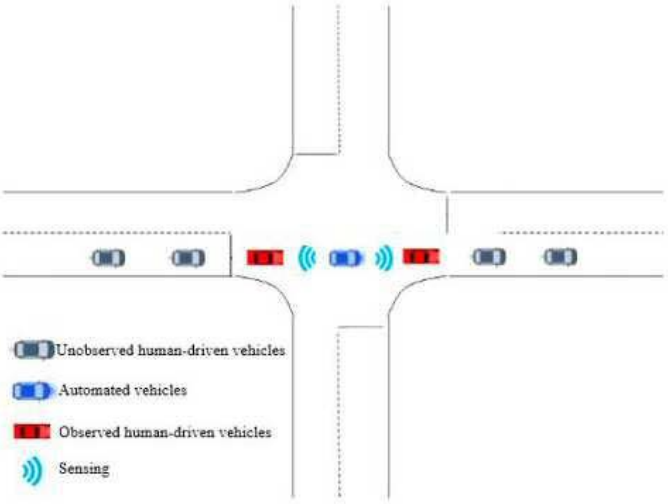
【도 4】


S409
SUMO 환경
SUMO 시뮬레이션
배기가스, 속도, 위치값 파일 생성 (CSV 파일)
S410
S401
S402
S403
제어 정보 전달
시뮬레이션 상태 전달
모델 학습시작
시뮬레이션 초기화
Flow 환경
강화학습을 적용하지 않은 운행행태 도출
차량 제어 모듈
시뮬레이션 상태 업데이트
S411
FLOW 환경
상태, 보상 전달
상태(State) 전달
S408
학습된 행태(주행방법) 전달
S413
강화학습 라이브러리 환경
강화학습 라이브러리
학습할 데이터 샘플링
운행행태 (정책) 훈련
훈련 결과 평가
PPO를 활용한 정책 최적화
정책 업데이트 및 저장
S414
S404
S405
S406
S407
S412
학습 루프 조건을 충족하는가?
학습 끝

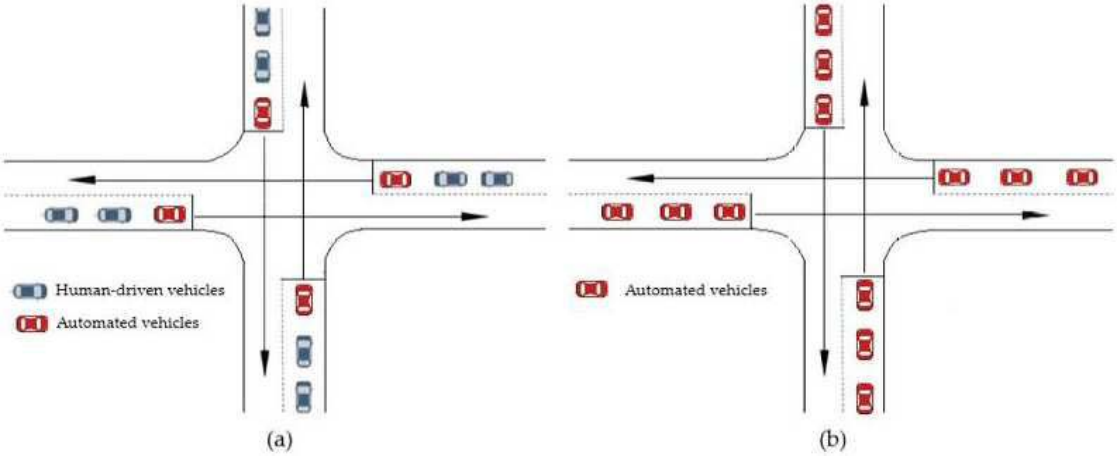
【도 5】



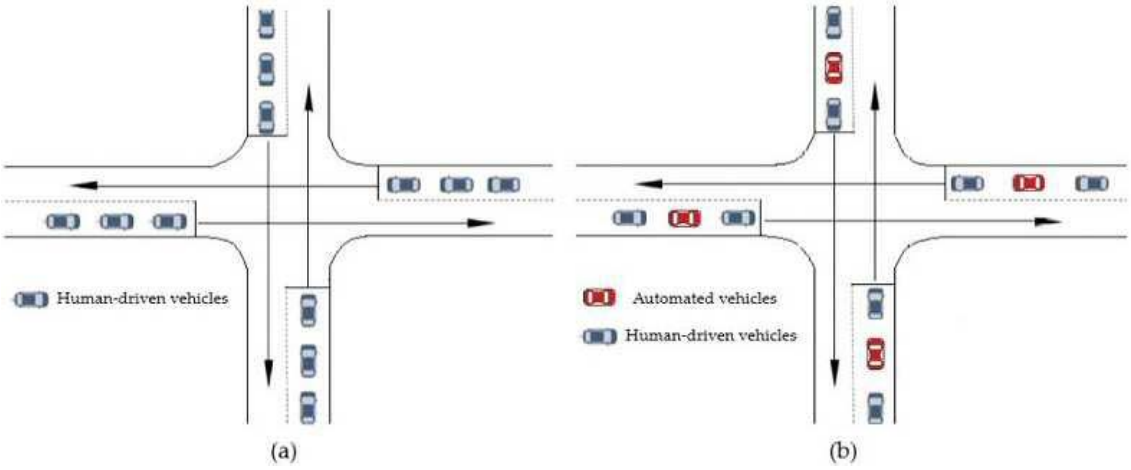
【도 6】


Unobserved human-driven vehicles
Automated vehicles
Observed human-driven vehicles
Sensing

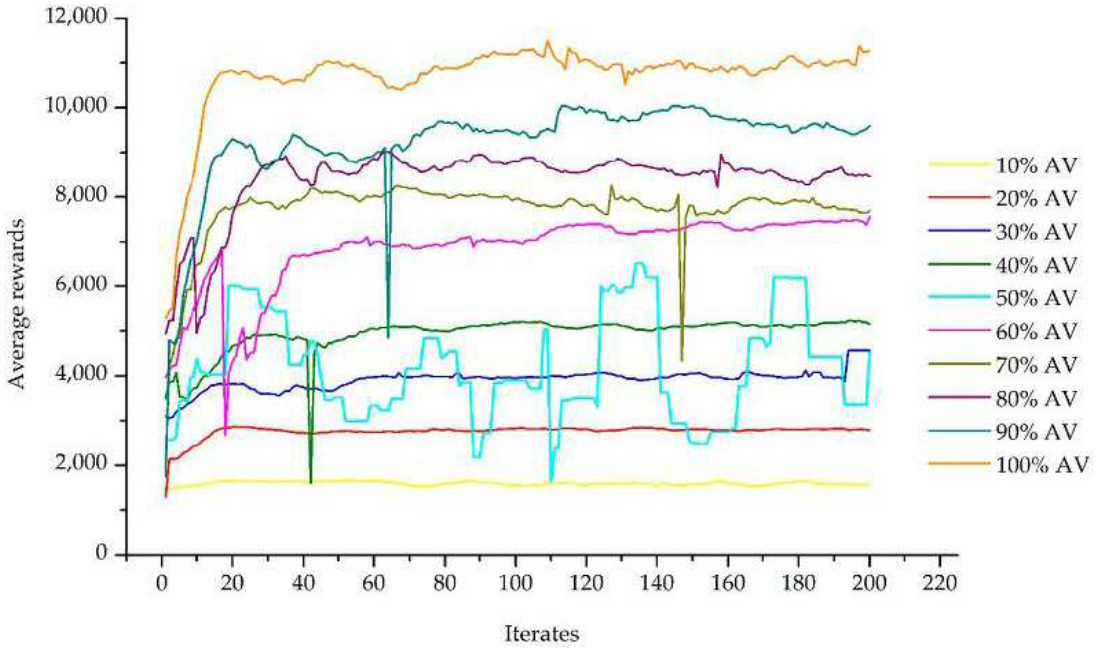
【도 7】


Human-driven vehicles
Automated vehicles
(a)
Automated vehicles
(b)

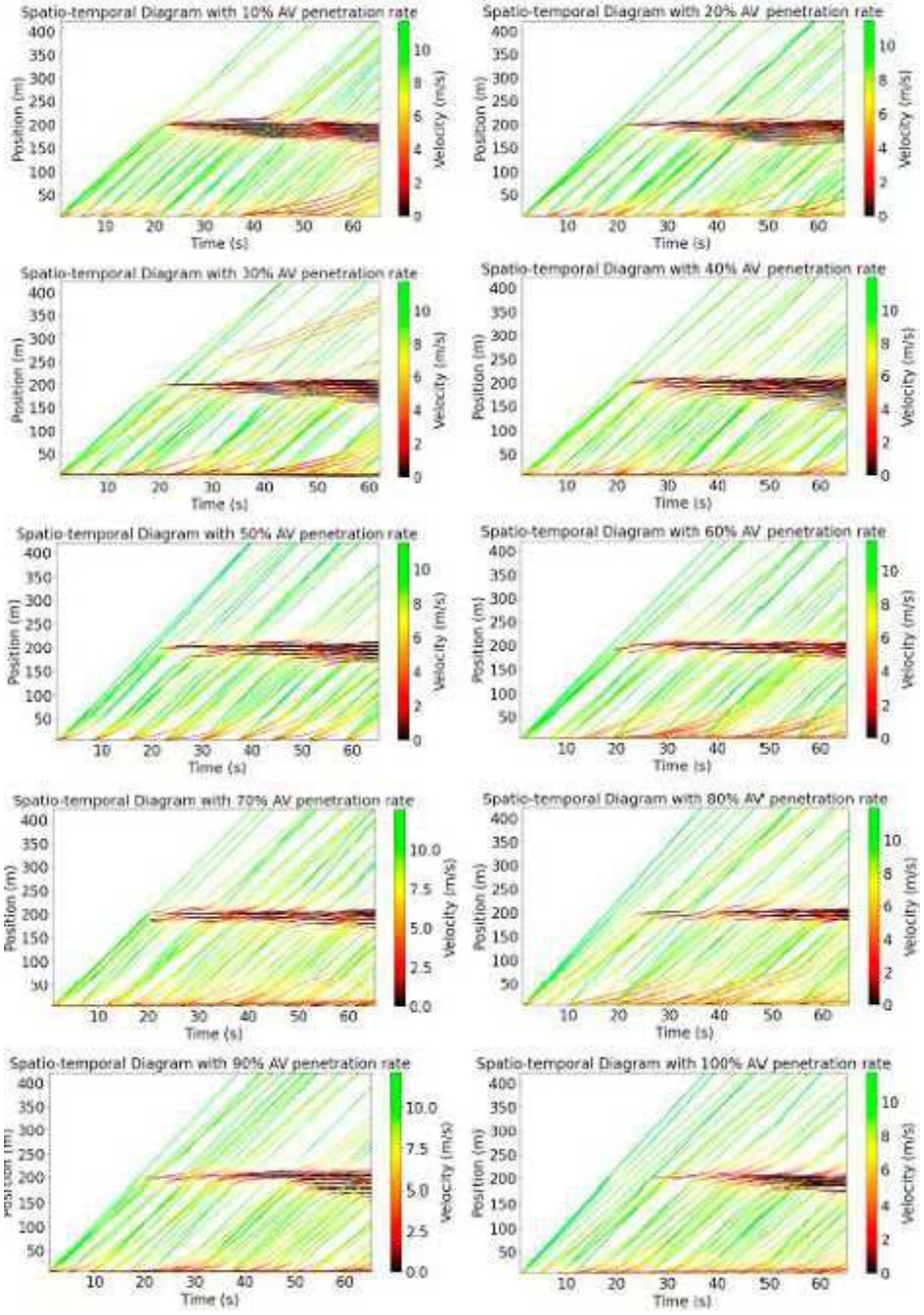
【도 8】


Human-driven vehicles
(a)
Automated vehicles
Human-driven vehicles
(b)

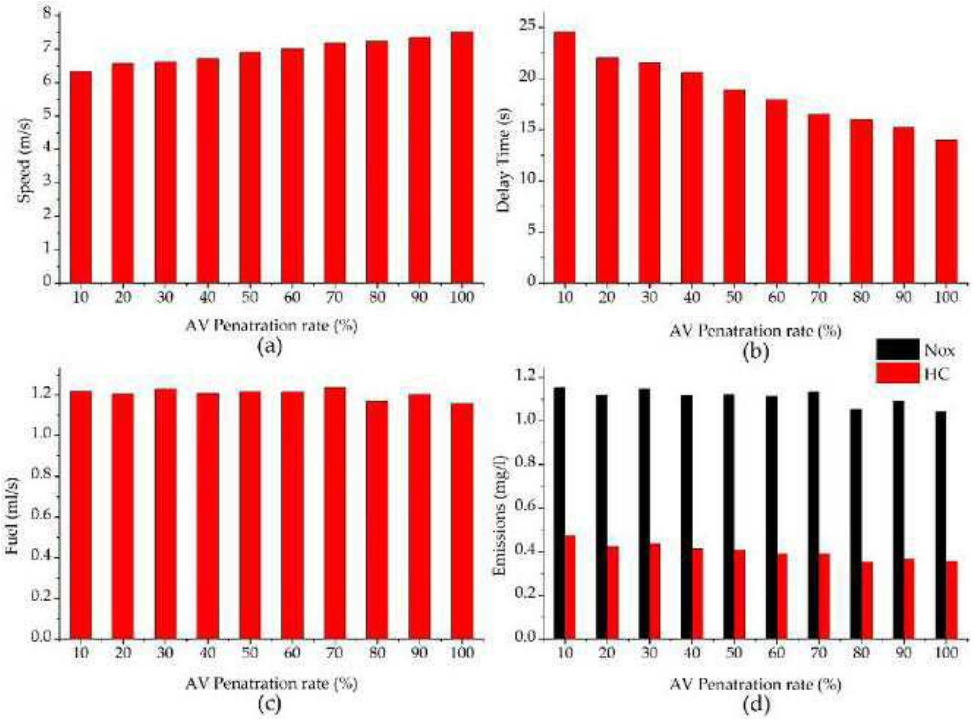
【도 9】


12,000
10,000
8,000
6,000
4,000
2,000
0
Average rewards
0
20
40
60
80
100
120
140
160
180
200
220
Iterates
10% AV
20% AV
30% AV
40% AV
50% AV
60% AV
70% AV
80% AV
90% AV
100% AV

【도 10】


Spatio-temporal Diagram with 10% AV penetration rate
Spatio-temporal Diagram with 20% AV penetration rate
Spatio-temporal Diagram with 30% AV penetration rate
Spatio-temporal Diagram with 40% AV penetration rate
Spatio-temporal Diagram with 50% AV penetration rate
Spatio-temporal Diagram with 60% AV penetration rate
Spatio-temporal Diagram with 70% AV penetration rate
Spatio-temporal Diagram with 80% AV penetration rate
Spatio-temporal Diagram with 90% AV penetration rate
Spatio-temporal Diagram with 100% AV penetration rate
Position (m)
Time (s)
Velocity (m/s)

【도 11】


Speed (m/s)
AV Penatration rate (%)
(a)
Delay Time (s)
AV Penatration rate (%)
(b)
Fuel (ml/s)
AV Penatration rate (%)
(c)
Emissions (mg/l)
Nox
HC
AV Penatration rate (%)
(d)