# Platoon Control of Connected Vehicles via Safe Reinforcement Learning Based on Lyapunov Based Soft Actor Critic Algorithm


1st Xiaoyuan Luo
*School of Electrical Engineering*
*Yanshan University*
Qinhuangdao, China
xyluo@ysu.edu.cn

2nd Tiankuo Hao
*School of Electrical Engineering*
*Yanshan University*
Qinhuangdao, China
tkhao527@qq.com

3rd Shaobao Li
*School of Electrical Engineering*
*Yanshan University*
Qinhuangdao, China
shbli84@163.com


2024 36th Chinese Control and Decision Conference (CCDC) | 979-8-3503-8778-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/CCDC62350.2024.10587774


***Abstract*—This paper addresses the platoon control problem of connected vehicular systems subject to safety constraints. A constrained Markov Decision Process model is established. A novel safe-reinforcement-learning control method based on the Lyapunov-based Soft Actor-Critic (LSAC) algorithm is developed, where the LSAC algorithm is designed to maximize the reward while minimizing the safety cost, ensuring minimal risk of collisions. Comparisons between the safe-reinforcement-learning control via LSAC algorithm and Soft Actor-Critic (SAC) algorithm are conducted to demonstrate the effectiveness of the proposed algorithm.**



***Index Terms*—platoon control, safe reinforcement learning (RL), Markov Decision Process, distributed control**


## I. Introduction

Significant progress has been achieved in cooperative platooning of connected autonomous vehicular systems [1]. Benefiting from various onboard sensors and vehicle-to-vehicle (V2V) communication technique, vehicles can exchange information such as the speed, acceleration and position between each other. Cooperative platooning of autonomous vehicular systems aims to design a distributed control law such that vehicles can travel at the same speed while maintaining a desired inter-vehicle distance. Therefore, the vehicular platooning technique can effectively reduce fuel consumption while simultaneously increasing road capacity. However, in practice, vehicles will be subject to various constraints to keep safety, such as the safe inter-vehicle distances, input saturation, and so on. It is still a challenge to cooperatively learn distributed control strategy for a connected autonomous vehicular system subject to multiple constraints.

In present, cooperative platooning studies via typical control methods like the linear feedback control, $\mathcal{H}_\infty$ controller and sliding mode controller (SMC), etc. are relatively mature [2–4]. However, the existing researches still have some limitations on platooning performance improvement under uncertain environments. For example, Zheng et al. [5] proposed a linear state feedback control law for homogeneous vehicular platoon system with inertial delay. To improve control performance under disturbances, Huang et al. [6] presented a robust dynamic output feedback controller such that the closed-loop platoon systems satisfy a given $\mathcal{H}_\infty$ performance index. Sawant et al. [7] proposed a sliding mode control method based on a disturbance observer in the absence of acceleration information of the preceding vehicles. To compensate for the effects of actuator failure and saturation, Guo et al. [8] proposed an adaptive fault-tolerant control method based on nonlinear vehicle dynamics and a new quadratic spacing strategy. However, the aforementioned algorithms depend on well modeled environments and can hardly handle system control subject to multiple constraints. Model predictive control (MPC) has been applied to cope with cooperative platooning control problem of connected vehicular systems subject to multiple constraints [9], [10]. Although the MPC methods solve the platoon control of vehicular systems subject to safety constraints, they require well established mathematical models, which are not convenient to implement in practice.

Some recent researches on vehicular platoon control have focused on machine-learning-based control [11]. In particular, reinforcement learning (RL) has been widely applied in the vehicular platooning. For example, Wang et al. [12] proposed an approximate reinforcement learning approach to learn the parameters of a classical proportional-integral-derivative (PID) controller instead of direct longitudinal control for autonomous vehicles with unknown dynamics and external disturbances. To solve the problem of model-free vehicle formation control, Luo et al. [13] proposed a critic-only Q-learning (CoQL) method , which learns the optimal tracking control from real system data. Li et al. [14] proposed an informed approximate Q-learning algorithm with efficient training, fast convergence, and good performance. However, Q-learning is suitable for discrete action spaces. For continuous action spaces, discretization is required and may lead to information loss and increased computational complexity [15]. In order to solve the vehicular platooning problem in continuous action space, some deep reinforcement learning (DRL) algorithms have been proposed. For instance, authors of [16], [17] proposed

This work was supported by Iconic Specialized Cultivation Projectof Yanshan University (Grant No.2022BZZD005)

a DDPG-based centralized control algorithm for formation control of vehicular systems. Compared with DDPG, Haarnoja et al. [18] presented a Soft Actor-Critic (SAC) reinforcement learning algorithm using maximum entropy theory for reward function design and achieves a better exploration-exploitation trade-off. This method enhances the stability and efficiency of the learning process. Additionally, SAC can efficiently run in a distributed computing framework, making it possible to train multiple agents simultaneously and improving the efficiency and stability of the SAC algorithm. Although the SAC algorithm is a promising method, safety is rarely considered in vehicle platoon applications.

In this paper, a novel SRL algorithm will be developed to address the cooperative platooning problem of connected vehicular systems subject to multiple safety constraints. Towards this end, a novel Lyapunov-based Soft Actor Critic (LSAC) algorithm is presented to ensure safe criteria of the connected vehicular systems. The LSAC algorithm employs a Lyapunov function to guarantee stability and security during training while improving the convergence speed of the algorithm. The main contributions of this article are summarized as follows:

(1) A constrained markov decision process (CMDP) model is established to describe the platoon control system, where Lyapunov function is constructed using the distance between vehicles as the cost function to constrain the vehicular platoon system.

(2) A distributed safe reinforcement learning algorithm based on LSAC is developed for platoon control towards performance and security improvement. Compared with existing works, the proposed algorithm can reach better convergence performance.

The rest of this paper is organized as follows. Section II formulates the cooperative platooning problem of connected vehicles. Section III presents the safe reinforcement learning algorithm based on the LSAC. Simulation studies are conducted to demonstrate the effectiveness of the proposed algorithm in Section IV. Section V concludes this work.

## II. Cooperative Adaptive Cruise Control Based on Safe Reinforcement Learning

In this section, the platoon control problem of connected vehicles is described and a constrained MDP model is established for the platoon control system.

### A. Platoon control system

A platoon control system consists of multiple connected autonomous vehicles, aiming to reach a string formation with a desired inter-distance as shown in Fig. 1. Vehicles make decisions by a distributed control policy using local information, such as the position, speed of nearby vehicles , etc. Each vehicle in the platoon has a high degree of autonomy. It can sense the environment, plan its trajectory, and control its own speed and acceleration without explicit instructions from a central controller. The dynamics of the vehicles are given as follows:

$$p_{i,t} = p_{i,t-1} + v_{i,t-1} \cdot \sigma \tag{1}$$

$$v_{i,t} = v_{i,t-1} + u_{i,t-1} \cdot \sigma \tag{2}$$

where $p_{i,t}$ is the position of the $i$-th vehicle at time $t$, $v_{i,t}$ is the speed of the $i$-th vehicle at $t$ time, $\sigma$ is sampling interval, and $u_{i,t}$ is the acceleration of the $i$-th vehicle at $t$ time.

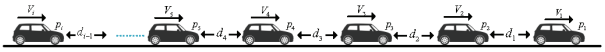


Fig. 1. Schematic diagram of a vehicular platoon system.

### B. Optimal Decision Problem

Reinforcement learning (RL) is a trial-and-error learning algorithm that optimizes the agent actions to maximize accumulated rewards (or minimize accumulated costs ) during its interaction with the environment. The vehicle platoon control is transformed into an optimal decision control. The goal is to learn an optimal policy $\pi$, which maximizes the expected $\sum_{t=1}^{T} \gamma^t r(s_t)$,$\gamma$ $\in$[0,1] is a discount, and the problem can be formulated as follows:

$$\underset{\pi}{argmax}\, \mathbb{E}_\pi[r_t + 1 + \gamma r_{t+2} + \cdots + \gamma^{T-1} r_t | s_t = s] \tag{3}$$

$$s.t. \qquad d_{min} \le d_t \le d_{max} \tag{4}$$

$$v_{min} \le v_t \le v_{max} \tag{5}$$

$$a_{min} \le a_t \le a_{max} \tag{6}$$

$$p_{i,t} = p_{i,t-1} + v_{i,t-1} \cdot \sigma \tag{7}$$

$$v_{i,t} = v_{i,t-1} + u_{i,t-1} \cdot \sigma \tag{8}$$

where $d_t$ is the distance headway at time $t$ , $v_t$ is the vehicle speed, $a_t$ is the vehicle acceleration, and $\sigma$ is the simulation step interval. Inequality (4) is the vehicle safe distance headway constraint, (5) is the vheicle speed constraint, (6) is the vheicle acceleration constraint, and (7) and (8) are the general kinematics of the autonomous vehicles.

To implement the optimal decision problem (3)-(8), a CMDP model is established for the vehicular system as follows:

- Action Space: action $a_t \in A$ is constructed by acceleration, $a_t = [u_t]$ with $u_t \in (-u_{max}, u_{max})$, where $u_{max}$ is the maximum acceleration of the vehicles.
- State Space: the state $s_t \in S$ is constructed by speed and headway, $s_t = [v_t, d_t]$ with $v_t \in (0, v_{max}), d_t \in (d_{min}, d_{max})$, where $v_{max}$ is the maximum speed of the vehicles, and $(d_{min}, d_{max})$ is the safety range.
- Reward Function: The reward function is defined as follows:

$$r(s_t, a_t) = -|d_{target} - d_t| + \frac{1}{2}|(v_{k,t} - v_{k-1,t})| + 10 \tag{9}$$

  where $v_{k,t}$ is the $k$-th vehicle at $t$ time.
- State Transition Probability: The follower vehicle in state $s_t$ takes action $a_t$ , leading to the state transition from $s_t$ to $s_{t+1}$. The probabilistic state transition model, denoted

as $P(s_{t+1}, s_t|a_t)$, is determined by the kinematics as defined in (8).

- Cost Function: The cost function is to ensure that the following vehicle remains within the safety range. The purpose of this design is to ensure the non-negativity of the cost function and facilitate the design of subsequent constraint functions.

$$c(s_t, a_t) = \begin{cases} d_t - \dfrac{d_{max} + d_{min}}{2}, & d_t > \dfrac{d_{max} + d_{min}}{2} \\ \dfrac{d_{max} + d_{min}}{2} - d_t, & d_t < \dfrac{d_{max} + d_{min}}{2} \end{cases} \tag{10}$$

where $d_{max}$(or $d_{min}$) is the bound of the safety range.

The goal of the optimal decision problem is to find a policy $\pi^*$ that satisfies the above model.

## III. Lyapunov-Based Soft Actor-Critic

A novel off-policy RL algorithm based on the actor-critic algorithm, namely the Lyapunov-based Soft Actor-Critic (LSAC) will be designed to solve the vehicular platooning problem in this section. The architecture of the proposed algorithm is shown in Figure 2.

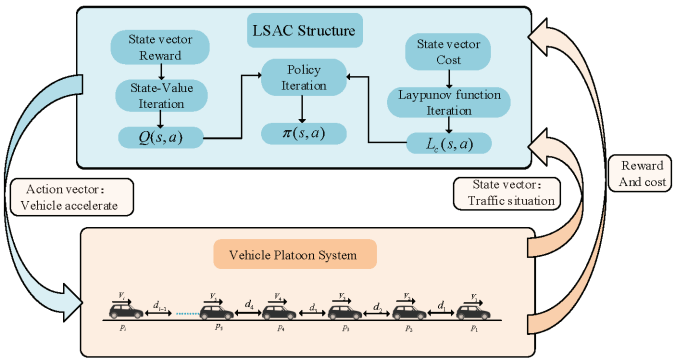



Fig. 2. LSAC-based vehicle platoon.

### A. Soft Actor-Critic

The Soft Actor-Critic(SAC) algorithm is a reinforcement learning algorithm used for training agents in environments where continuous actions and a high degree of exploration are required [18]. In the SAC, the actor aims to simultaneously maximize expected return and entropy to succeed at the task while acting as randomly as possible. For this purpose, by recalling the performance function (3), the optimal objective is given as

$$\pi^* = arg\max_{\pi} \sum_t \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}[r(s_t, a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))] \tag{11}$$

where $\mathcal{H}(\cdot)$ is the entropy and $\alpha$ is the temperature parameter that determines the relative importance of the entropy term versus the reward [18].

### B. Safe RL With UUB

In this paper, an SRL algorithm will be proposed to learn and explore the optimal security policy for cooperative platooning of connected vehicular systems in the sense of uniformly ultimately bounded (UUB). This method will use Lyapunov functions based on data to analyze the closed-loop stability of stochastic nonlinear systems characterized by the CMDP. The Lyapunov function is set as a positive function with respect to the state $L : S \to \mathbb{R}_+$. The fundamental concept of leveraging the Lyapunov function is to guarantee that the derivative of the Lyapunov function along the state trajectory is semi-negative definite. This ensures that the state evolves in the direction of decreasing values of the Lyapunov function, ultimately converging either to the origin or a sublevel set of the Lyapunov function. The following lemma shows the property of the Lyapunov function.

Lemma [19]: For a given safety constraint $\bar{d}$, if there exists a function $L(s) : S \to \mathbb{R}_+$ and positive constants $\zeta_1, \zeta_2, \zeta_3, \eta$ and $e$, such that

$$\eta < \bar{d} \tag{12}$$

$$e - \zeta_1\eta \le \zeta_3\bar{d} \tag{13}$$

$$\mathbb{E}_{t_0\sim P(t_0|\rho,\pi),P(s|\pi,\rho,t_0)}[L(s)] < e \tag{14}$$

$$\zeta_1 c_\pi(s) \le L(s) \le \zeta_2 c_\pi(s) \quad \forall s \in S \tag{15}$$

$$\begin{aligned} &\mathbb{E}_{s\sim\mu(s)}[\mathbb{E}_{s'\sim P_\pi}[L(s')\mathbb{1}_\Delta(s') - L(s)\mathbb{1}_\Delta(s)]] \\ &\le -\zeta_3\mathbb{E}_{s\sim\mu(s)}[c_\pi(s)]\mathbb{1}_\Delta(s) \end{aligned} \tag{16}$$

where $c_\pi(s) = \mathbb{E}_{a\sim\pi}c(s, a)$ denotes the constraint function under the controller $\pi$, where $\mu_N(s)$ denotes the average distribution of $s$ over the finite $N$ time steps, and $\mu(s) \doteq (\frac{1}{N})\sum_{T=0}^{N}\mathbb{E}_{t_0\sim P(t_0|\rho,\pi)}P(s|\pi, \rho, t_0+T)$ is the average probability of being in $s$ during the time in edge set $\Delta$ under policy $\pi$ with $\Delta = \{s|c_\pi(s) \ge \eta\}$. $\mathbb{1}_\Delta(s)$ is described as follows:

$$\mathbb{1}_\Delta(s) = \begin{cases} 1 & s \in \Delta \\ 0 & s \notin \Delta \end{cases} \tag{17}$$

Then, the system is UUB stable.

The proof of Lemma is given in [19]. Inequality (15) confines the property that the Lyapunov function needs to satisfy. (16) is the data-based energy decreasing condition, which requires the Lyapunov value to be decreased in the edge set $\Delta$ and finally enter the safety range. Lyapunov function is chosen as $L(s) = \sum_t^{t+K}\mathbb{E}[c(s_t)]$, where $\sum_t^{t+K}\mathbb{E}[c(s_t)]$ is the sum of the cost over a finite time.

### C. Lyapunov critic function

The Lyapunov function $L(s)$ is crucial for stability analysis. Since $L$ is not available for the gradient of the controller $\pi$, the application of this method is not directly feasible in existing actor-critic learning frameworks. Therefore, the Lyapunov critic function $Lc(s, a) = \sum_t^{t+K}\mathbb{E}[c(s_t)]$ is introduced to guarantee the stability. The Lyapunov critic function $Lc$ depends on both the state $s$ and the action $a$. $Lc$ satisfies $L(s) = \mathbb{E}_{a\sim\pi}Lc(s, a)$, such that it can be exploited by judging

the value of (16). In this paper, $Lc$ is constructed by utilizing a fully connected deep neural network (DNN) parameterized by $\phi$. Since the Lyapunov function is positive definite, we can use the Relu activation function to ensure positive DNN output.

There are many functions satisfying the condition of Lyapunov function (14). These functions are called Lyapunov candidate functions. The gradient of Lyapunov candidate functions relative to the controller is not processable, and thus they cannot be directly applied to the actor critic learning process. Therefore, the Lyapunov candidate function is used as a supervisory signal to supervise the learning of the Lyapunov critical function $L_c$ network. The Lyapunov objective function ($L_{target}$) is designed to be related to the Lyapunov candidate function, minimizing the following objective function

$$J(\psi) = \mathbb{E}_{\mathcal{D}}[\frac{1}{2}(L_c(s,a) - L_{target}(s,a))^2] \tag{18}$$

where $\mathbb{D} = (s, a, s', r, c)$ is the set of transfer tuples collected under policy $\pi$.

The choice of the Lyapunov candidate plays an important role in learning a controller. For the case of using cost value function as the Lyapunov function

$$L_{target}(s,a) = c(s,a) + \gamma L_c'(s',a') \tag{19}$$

where $L_c'$ is the target Lyapunov critic function typically used in the actor critic methods, which has the same structure with $L_c$, but the parameter is updated through the exponentially moving average of weights of $L_c$ controlled by a hyperparameter $\tau$.

### D. Training the Policy

Based on SAC, an off-policy RL algorithm called LSAC is proposed in this section. LSAC is a SAC algorithm that adds UUB stability constraints (13) and (15) through Lagrangian method. Consequently, the objective function $J(\pi)$ is given as follows:

$$\begin{aligned} J(\pi) &= \mathbb{E}_{s_t\sim D}[-Q(s, f_\theta(s,\epsilon)) + \beta log\pi_\theta(f_\theta(s,\epsilon)|s)] \\ &+ \xi\mathbb{E}_{D_\Delta}[L_c(s', f_\theta(s',\epsilon))\mathbb{1}_\Delta(s') - (L_c(s,\epsilon) - \zeta_3 c(s,a))\mathbb{1}_\Delta(s)] \\ &+ \nu\mathbb{E}_{P(s|\rho,\pi,t_0),P(t_0|\rho,\pi),\pi}(L_c(s,a) - e)\mathbb{1}_\Delta(s) \end{aligned} \tag{20}$$

where $\beta$ , $\xi$ and $\nu$ are Lagrangian multipliers, and $\beta$ is similar to the temperature parameter in SAC. In (20), $\mathbb{D} = (s, a, s', r, c)$ is the replay buffer for storage of CMDP tuples. $\mathbb{D}_\Delta$ is the replay buffer for storage of edge tuples to train $\xi$ and $\nu$. The method to update $\beta$, $\xi$, $\nu$ is given as follows:

$$J(\beta) = \beta\mathbb{E}_{(s,a)\sim D}[log\pi_\theta(a|s) + \mathcal{H}] \tag{21}$$

$$\begin{aligned} J(\xi) =& \xi\mathbb{E}_{(s,a)\sim D_\Delta}[L_c(s', f_\theta(s',\epsilon))\mathbb{1}_\Delta(s') \\ &- (L_c(s,a) - \zeta_3 c(s,a))\mathbb{1}_\Delta(s)] \end{aligned} \tag{22}$$

$$J(\nu) = \nu\mathbb{E}_{P(s|\rho,\pi,t_0),P(t_0|\rho,\pi),\pi}(L_c(s,a) - e)\mathbb{1}_\Delta(s) \tag{23}$$

The pseudocode of LSAC is shown in Algorithm 1.

**Algorithm 1** LSAC

**Require:** hyperparameters, learning rates $l_r$, parameters regarding UUB theorem $\zeta_3$

Initialize a critic network $Q(s,a)$, Lyapunov critic network $Lc(s,a)$, actor network $\pi(a|s)$ with parameters $\theta_Q$, $\psi_{L_c}$ , $\phi$ and the Lagrangian multipliers $\xi, \beta, \nu$.
Initialize the parameters of the target networks with $\overline{\theta}_Q \leftarrow \theta_Q$, $\overline{\psi}L_c \leftarrow \psi L_c$ , $\overline{\phi} \leftarrow \phi$
**for** each iteration **do**
    Sample $s_0$ according to $\rho$
    **for** each environment step **do**
        Sample at from $\pi(s)$ and step forward
        Observe $s_{t+1}, r_t, c_t$
        Store $(s_t, a_t, r_t, c_t, s_{t+1})$ in $\mathbb{D}$
        If $s_t \in \Delta$, store $(s_t, a_t, r_t, c_t, s_{t+1})$ in $\mathbb{D}_\Delta$
    **end for**
    **for** each update step **do**
        Sample minibatches of tuples from $\mathbb{D}$ and $\mathbb{D}_\Delta$ and update $Q, L_c, \pi$ and Lagrangian multipliers $\xi, \beta, \nu$
        Update the target networks with soft replacement:
        $\overline{\theta}_Q \leftarrow \tau\theta_Q + (1-\tau)\overline{\theta}_Q$
        $\overline{\psi}_{L_c} \leftarrow \tau\psi_{L_c} + (1-\tau)\overline{\psi}_{L_c}$
        $\overline{\phi} \leftarrow \tau\phi + (1-\tau)\overline{\phi}$
    **end for**
**end for**

**Ensure:** $\theta_Q, \psi_{L_c}, \phi$

## IV. Simulations

In this section, a numerical simulation on the platooning of connected vehicular system using the proposed LSAC algorithm is conducted. Next, in the simulation environment, the success rate of the algorithm is compared with that of SAC in the training process. Finally, the performance under different response values is compared.

### A. Simulation Platform

A traffic simulation software named the simulation of urban mobility (SUMO) is adopted to simulate the platooning of connected vehicular system in this work. SUMO has been widely used traffic simulation platform in existing studies [20–22]. Compared with other microscopic simulations, SUMO provides numerous car-following and lanechanging models that can satisfy the research needs. An important advantage is its open source nature, which provides users with greater flexibility for customization and development. SUMO not only facilitates seamless communication and interaction with other software but also supports functionalities like importing Python packages and connecting to Veins.

### B. Training Environment and Parameter Settings

In this scenario, a vehicular system consisting of 1 leading vehicle and 6 follower vehicles is considered. The initial headway between the vehicles is $10m$, and their initial speed is $0m/s$. The vehicles have a maximum acceleration of $3m/s^2$ (or a maximum deceleration of -4.5$m/s^2$). Once the speed

of the leading vehicle reaches $20m/s$, it will cruise at a constant speed. The simulation has 200 steps, with each step representing 0.5 seconds, and thus the total running time is 100 seconds.

The LSAC-based vehicle platoon was trained through the interaction with SUMO. The training parameters are shown in TABLE 1.

TABLE I
LSAC HYPERPARAMETERS.

| **Hyperparmeter** | ***Value*** | ***Discirption*** |
|---|---|---|
| Learning rate | 3e-4 | learning rate in all network |
| Episode | 1200 | number of episode |
| Batch size | 256 | each stochastic gradient descent update is computed |
| Discount factor | 0.98 | discount factor gamma |
| Capacity | 20000 | replay buffer size |
| $\zeta_3$ | 1 | parameters of (16) |

### C. Evaluation on Algorithmic

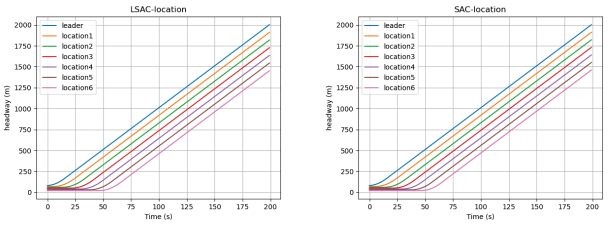
The performance of LSAC is evaluated and compared with the SAC. Evaluation LSAC , since LSAC has the limitation of safety cost, LSAC has better performance for vehicle control. During training, a leading vehicle and three follower vehicles are used. During the test, one leading vehicle and six follower vehicles are used. It is shown in Fig. 3 that vehicles reach platoon under both algorithms. Specifically, Fig. 4 and 5 illustrate that in the initial 50 steps of the following vehicle, the SAC-controlled vehicle jitters when it reaches the standard speed and distance, which is very dangerous, while the LSAC-controlled vehicle does not have this situation. As shown in Fig. 6, when the vehicle speed controlled by SAC reaches stable, the acceleration of the rear vehicle is greatly affected, and deceleration occurs. The vehicle acceleration controlled by LSAC is relatively smooth. Comparing with the rewards, one can find that the reward of LSAC reaches more than 15000 at 20 steps, while the reward of SAC is only reached at 50 steps. Thus, the LSAC is more stable than the SAC algorithm in the training as shown in Fig. 7 and 8.

## V. CONCLUSION

In this paper, the vehicle platooning problem of connected vehicular systems with safety constraints is studied. A safe reinforcement learning (SRL) algorithm is proposed, where Lyapunov functions are employed to represent constraints, and lagrangian methods are used to integrate constraints into the SAC algorithm to solve this safe reinforcement learning problem. A Lyapunov-based Soft Actor-Critic (LSAC) algorithm is developed and the control performance is compared with the Soft Actor-Critic (SAC) vehicle model as a benchmark.

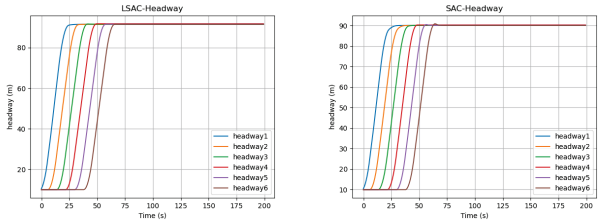
## REFERENCES


[1] V. Lesch, M. Breitbach, M. Segata, C. Becker, S. Kounev, and C. Krupitzer, "Anoverview on approaches for coordination of platoons," *IEEETransactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp.10 049–10 065, 2022.

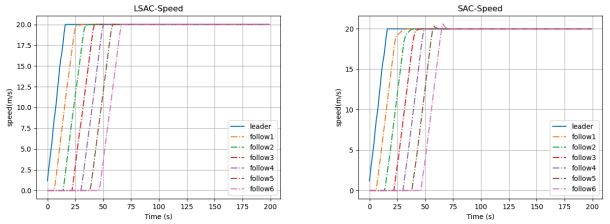

(a) LSAC location (b) SAC location

Fig. 3. vehicle location



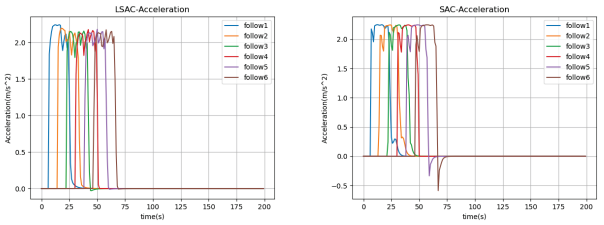

(a) LSAC headway (b) SAC headway

Fig. 4. vehicle headway



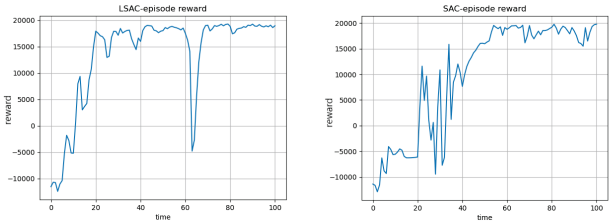

(a) LSAC speed (b) SAC speed

Fig. 5. vehicle speed



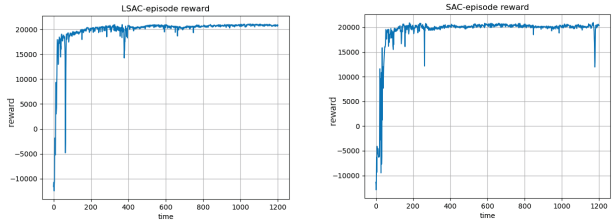

(a) LSAC acceleration (b) SAC acceleration

Fig. 6. vehicle acceleration




(a) LSAC reward 100-episode (b) SAC reward 100-episode

Fig. 7. rewards for the first 100-episode simulation




(a) LSAC reward all episode (b) SAC reward all episode

Fig. 8. rewards for the all episode simulation

[2] S. E. Li, Y. Zheng, K. Li, Y. Wu, J. K. Hedrick, F. Gao, and H. Zhang, "Dynamical modeling and distributed control of connected and automated vehicles: Challenges and opportunities," *IEEE Intelligent Transportation Systems Magazine*, vol. 9, no. 3, pp. 46–58, 2017.

[3] H. Guo, J. Liu, Q. Dai, H. Chen, Y. Wang, and W. Zhao, "A distributed adaptive triple-step nonlinear control for a connected automated vehicle platoon with dynamic uncertainty," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3861–3871, 2020.

[4] X. Guo, J. Wang, F. Liao, and R. S. H. Teo, "Distributed adaptive integrated-sliding-mode controller synthesis for string stability of vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 9, pp. 2419–2429, 2016.

[5] Y. Zheng, S. Eben Li, J. Wang, D. Cao, and K. Li, "Stability and scalability of homogeneous vehicular platoon: Study on the influence of information flow topologies," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 1, pp. 14–26, 2016.

[6] C. Huang, S. Coskun, J. Wang, P. Mei, and Q. Shi, "Robust $h_\infty$ dynamic output-feedback control for cacc with rosss subject to rodas," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 1, pp. 137–147, 2022.

[7] J. Sawant, U. Chaskar, and D. Ginoya, "Robust control of cooperative adaptive cruise control in the absence of information about preceding vehicle acceleration," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5589–5598, 2021.

[8] G. Guo, P. Li, and L.-Y. Hao, "A new quadratic spacing policy and adaptive fault-tolerant platooning with actuator saturation," *IEEE transactions on intelligent transportation systems*, vol. 23, no. 2, pp. 1200–1212, 2020.

[9] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Rolling horizon control framework for driver assistance systems. part ii: Cooperative sensing and cooperative control," *Transportation research part C: emerging technologies*, vol. 40, pp. 290–311, 2014.

[10] M. Wang, W. Daamen, S. P. Hoogendoorn, and B. van Arem, "Cooperative car-following control: Distributed algorithm and impact on moving jam features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1459–1471, 2015.

[11] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takac, "Reinforcement learning for solving the vehicle routing problem," in *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., vol. 31. Curran Associates, Inc., 2018.

[12] J. Wang, X. Xu, D. Liu, Z. Sun, and Q. Chen, "Self-learning cruise control using kernel-based least squares policy iteration," *IEEE Transactions on Control Systems Technology*, vol. 22, no. 3, pp. 1078–1087, 2014.

[13] B. Luo, D. Liu, T. Huang, and D. Wang, "Model-free optimal tracking control via critic-only q-learning," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 10, pp. 2134–2144, 2016.

[14] Z. Li, T. Chu, I. V. Kolmanovsky, and X. Yin, "Training drift counteraction optimal control policies using reinforcement learning: An adaptive cruise control example," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 9, pp. 2903–2912, 2018.

[15] J. Guo, "Human-level control through deep reinforcement learning," 2016.

[16] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[17] G. Wang, J. Hu, Y. Huo, and Z. Zhang, "A novel vehicle platoon following controller based on deep deterministic policy gradient algorithms," in *18th COTA International Conference of Transportation Professionals*. American Society of Civil Engineers Reston, VA, 2018, pp. 76–86.

[18] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018.

[19] M. Han, Y. Tian, L. Zhang, J. Wang, and W. Pan, "Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee," *Automatica*, vol. 129, p. 109689, 2021.

[20] C. Sommer, R. German, and F. Dressler, "Bidirectionally coupled network and road traffic simulation for improved ivc analysis," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 3–15, 2011.

[21] K. Pandit, D. Ghosal, H. M. Zhang, and C.-N. Chuah, "Adaptive traffic signal control with vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 4, pp. 1459–1471, 2013.

[22] A. Al-Fuqaha, A. Gharaibeh, I. Mohammed, S. J. Hussini, A. Khreishah, and I. Khalil, "Online algorithm for opportunistic handling of received packets in vehicular networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 285–296, 2019.