International Journal of
Environmental Research and Public Health

MDPI

Article

Safe, Efficient, and Comfortable Autonomous Driving Based on Cooperative Vehicle Infrastructure System

Jing Chen [1], Cong Zhao [1,*], Shengchuan Jiang [1], Xinyuan Zhang [1], Zhongxin Li [2] and Yuchuan Du [1]

[1] Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China
[2] Shanghai Utopilot Technology Co., Ltd., Shanghai 201306, China
* Correspondence: zhc@tongji.edu.cn

Abstract: Traffic crashes, heavy congestion, and discomfort often occur on rough pavements due to human drivers' imperfect decision-making for vehicle control. Autonomous vehicles (AVs) will flood onto urban roads to replace human drivers and improve driving performance in the near future. With the development of the cooperative vehicle infrastructure system (CVIS), multi-source road and traffic information can be collected by onboard or roadside sensors and integrated into a cloud. The information is updated and used for decision-making in real-time. This study proposes an intelligent speed control approach for AVs in CVISs using deep reinforcement learning (DRL) to improve safety, efficiency, and ride comfort. First, the irregular and fluctuating road profiles of rough pavements are represented by maximum comfortable speeds on segments via vertical comfort evaluation. A DRL-based speed control model is then designed to learn safe, efficient, and comfortable car-following behavior based on road and traffic information. Specifically, the model is trained and tested in a stochastic environment using data sampled from 1341 car-following events collected in California and 110 rough pavements detected in Shanghai. The experimental results show that the DRL-based speed control model can improve computational efficiency, driving efficiency, longitudinal comfort, and vertical comfort in cars by 93.47%, 26.99%, 58.33%, and 6.05%, respectively, compared to a model predictive control-based adaptive cruise control. The results indicate that the proposed intelligent speed control approach for AVs is effective on rough pavements and has excellent potential for practical application.

Keywords: safety; autonomous vehicle; ride comfort; deep reinforcement learning; speed control

check for updates

Citation: Chen, J.; Zhao, C.; Jiang, S.; Zhang, X.; Li, Z.; Du, Y. Safe, Efficient, and Comfortable Autonomous Driving Based on Cooperative Vehicle Infrastructure System. *Int. J. Environ. Res. Public Health* **2023**, *20*, 893. https://doi.org/10.3390/ijerph20010893

Academic Editor: Paul B. Tchounwou

Received: 30 November 2022
Revised: 27 December 2022
Accepted: 29 December 2022
Published: 3 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Ride comfort has recently received much attention in different driving scenarios due to its influence on the public acceptance of autonomous vehicles (AVs) [1,2] and the health of passengers [3]. Ride comfort is a subjective sensation of passengers associated with the motion of vehicles in different directions. In longitudinal motion, car following is the most frequent scenario. The main task of autonomous car following is maintaining safe and comfortable following gaps via speed control [4]. Regarding vertical motion, the comfort issues caused by dramatic vehicle body vibration on rough pavements are concerned [5]. Speed control helps mitigate vertical vibration on rough pavements. However, safe, efficient, and comfortable speed control is rarely achieved in driving scenarios with car following and rough pavements. Indeed, simultaneously considering pavement conditions and vehicles in front is challenging for a human driver. Heavy congestion and traffic crashes are common on poor roads in peak periods. In this complex driving scenario, intelligent speed control of AVs promises to improve safety, efficiency, and ride comfort and mitigate driver workload.

For car-following behavior, rule-based and supervised learning-based approaches are used to establish car-following models. In rule-based approaches, conventional car-following models are usually used [6]. However, the rule-based approaches involve

strong assumptions and simplification, limiting their real-world application. Supervised learning approaches investigate the relationship between dynamic traffic and acceleration selection using extensive expert demonstrations. However, supervised learning only imitates human driving. Although neural networks can generate outputs regardless of inputs, the generalization capability is limited. Thus, for some untrained complex situations, it is difficult for supervised learning-based approaches to find optimal solutions. For speed control on rough pavements, model-based speed planning, such as dynamic programming, is commonly used [5,7]. However, model-based speed planning is also based on strong assumptions of the environment, so it struggles to address changing environments.

The application of model-free DRL algorithms in dynamic traffic scenarios has recently been researched. For example, Zhu et al. trained a DRL-based car-following model using 2000 periods of car following on urban expressways in Shanghai to outperform conventional car-following models [8]. Wu et al. trained a DRL-based differential variable speed limit controller to improve safety, efficiency, and environmental friendliness on freeways [9]. The experimental results show that the controller reduces travel times and CO_2 emissions. Mao et al. proposed a DRL-based framework to address the taxi dispatch problem with the imbalance of travel demand and taxi supply [10]. The framework outperforms the vanilla policy gradient method and shallow neural networks regarding convergence rate and quality. The above studies suggest that good performance and broad application of model-free DRL algorithms can be achieved in intelligent control.

In DRL-based speed control, a deep deterministic policy gradient (DDPG) algorithm has been widely used. Zhu et al. proposed a DDPG-based speed control for safe, efficient, and comfortable car-following behavior, which outperforms human drivers and model predictive control (MPC) [4]. However, this DDPG model only considered the dynamics of leading and following vehicles. In practice, driving environments are complex. For example, road alignment impacts vehicle dynamics and driving stability, and pavement conditions influence vehicle body vibration. Buechel and Knoll developed a DDPG-based predictive longitudinal controller that directly selects accelerations according to reference speeds and road grades [11]. Subsequently, the authors of this study have used the DDPG algorithm to control the speed with prior knowledge of the dynamic speed limit and comfortable speeds on rough pavements [12]. However, it only provides a solution to a multi-objective speed control problem for an AV without consideration of surrounding vehicles. Since the DDPG-based speed control has the characteristics of fast computation, superior driving performance, and good scalability [4,12], it promises to be a popular speed control approach in the era of autonomous driving. Thus, it is necessary to modify the existing DDPG-based speed control and extend application scenarios.

In this study, we proposed an intelligent speed control approach for safe, efficient, and comfortable car-following on rough pavements using the DDPG algorithm. As shown in Figure 1, the proposed speed control approach is applied in a cooperative vehicle infrastructure system (CVIS). In this system, AVs detect road profiles using onboard light detection and ranging (LiDAR), accelerometers, and global positioning systems (GPSs) and then send them to roadside units (RSUs) via vehicle-to-infrastructure communication. Dynamic traffic information can be detected by roadside sensors [13–17]. Furthermore, the multi-source road and traffic information is uploaded to the cloud server for integration. When an AV enters the road, it receives complete road profiles of the pavement. The AV then extracts the road profiles of the left and right wheels along the trajectory and calculates comfortable speeds on segments by vertical comfort evaluation. Meanwhile, the AV receives the location and speed information of surrounding vehicles, especially the leading vehicle, via vehicle-to-vehicle communication. Finally, the DRL-based speed control observes the information on comfortable speeds and leading vehicle and recommends accelerations. The AV adjusts driving speed to achieve safe, efficient, and comfortable driving according to recommended accelerations.

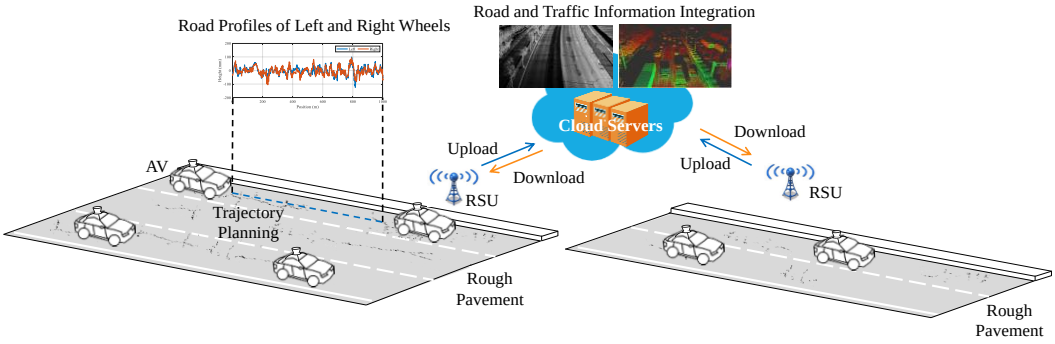

Figure 1. Intelligent speed control for AVs in the CVIS on rough pavements.

The contributions of this study are as follows:

(i) The application of DDPG-based speed control is extended to a scenario with car following and rough pavements, contributing to driving performance improvement and drivers' workload mitigation in complex driving scenarios.
(ii) A novel reward function is designed by incorporating safety, efficiency, vertical comfort, and longitudinal comfort regarding time to collision, time headway, clearance distance, annoyance rate, jerk, and acceleration.
(iii) The proposed intelligent speed control provides an approach for longitudinal acceleration selection based on dynamic traffic and road information in a CVIS.

The remainder of this paper is organized as follows. Section 2 proposes a vertical comfort evaluation approach using speeds to represent vertical comfort information on oncoming roads. Section 3 presents a DRL-based intelligent speed control for safe, efficient, and comfortable car-following on rough pavements. Section 4 details the training and testing data, DRL model training, and a performance comparison with an MPC baseline. Section 5 summarizes this study's findings and suggests directions for our future work.

2. Vertical Comfort Evaluation

On rough pavements, irregular road profiles often lead to discomfort in the vertical direction. For a vehicle, vertical ride comfort is directly related to the vertical vibration of the seats, which results from the interactions between the seats, vehicle body, suspensions, tires, wheels, and road profiles. The interactions are formulated as mathematical models [5,7]. Since the most commonly used model, the quarter-car model, is too simple to reflect the entire vibration information, a full-car model with a seat modeling is used (see Figure 2) [7]. The dynamic equation of the full-car model is summarized as

$$[M]\{\ddot{Z}\} + [C]\{\dot{Z}\} + [K]\{Z\} = \{F(t)\}, \tag{1}$$

where M, C, and K are the mass matrix, damping matrix, and spring matrix; $\ddot{Z}$, $\dot{Z}$, and Z are the acceleration vector, velocity vector, and displacement vector, respectively. For understanding, Equation (1) is further modified as a state-space formulation:

$$\begin{Bmatrix} \ddot{Z} \\ \dot{Z} \end{Bmatrix} = \begin{bmatrix} -M^{-1}C & -M^{-1}K \\ I & 0 \end{bmatrix} \begin{Bmatrix} \dot{Z} \\ Z \end{Bmatrix} + \begin{bmatrix} M^{-1} \\ 0 \end{bmatrix} F(t) \tag{2}$$

$$F(t) = \left[-k_t z_1(t),\ -k_t z_2(t),\ -k_t z_1\left(t - \frac{l}{v}\right),\ -k_t z_2\left(t - \frac{l}{v}\right),\ 0_{4\times 1} \right]^T, \tag{3}$$

where k_t, k_s, and k are the stiffness of the tire, suspension, and seat; I is the identity matrix; 0 is the null matrix; z_1 and z_2 are the road profiles of the right and left wheels; l is the distance between the front and rear axles; and v is the driving speed. Particularly, the inputs of the full-car model are road profiles in the time domain. Although the spatial road profiles

are the same, the time-domain data are changed according to the driving speeds [5]. The values of parameter coefficients are listed in the study of Cantisani and Loprencipe [18].

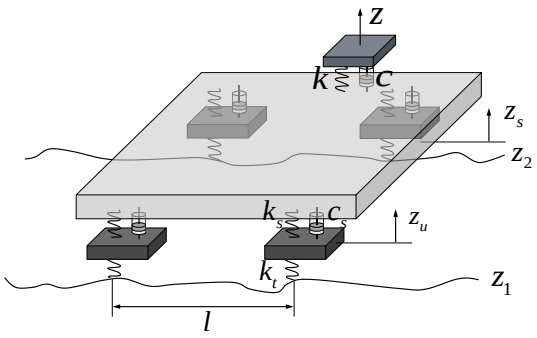

Figure 2. Full-car model.

In the state-space formulation, the output is the acceleration in the time domain with irregular fluctuations. Conversely, the patterns of frequency-domain acceleration are more stable [7]. Hence, the time-domain data are translated into the frequency domain using the power spectral density. In the frequency domain, the vibration in the frequency band 0.5–80 Hz has the largest impact on human sensation, and the effects of the separated bands within this range differ significantly. To distinguish these differences, the evaluation focuses on the vibration in the specific frequency band, and the frequency band is further separated into 23 sections by a 1/3 octave filter [19]. As recommended by ISO 2631-1-1997 [20], the weighted root mean square acceleration (WRMSA) is then used as an objective indicator to evaluate ride comfort. The WRMSA is calculated with a weighting coefficient assigned to each frequency band as

$$a_w = \sqrt{\sum_1^{23} \omega_i^2 \int_{li}^{ui} S_\alpha(f) df}, \tag{4}$$

where ω_i is the weighting coefficient for the i-th one-third octave band; ui and li are the upper and lower limiting frequencies of the i-th one-third octave band, respectively; and $S_\alpha(f)$ is the power spectral density of the vibration acceleration in the frequency domain.

Although the WRMSA can objectively evaluate ride comfort, the sensitivity differences of passengers cannot be characterized. It is noteworthy that ride comfort is a subjective sensation. Passengers may have distinct feelings even for the same vibration. To represent the proportion of passengers who cannot tolerate the vibration, the annoyance rate in experimental psychology is introduced to modify the evaluation results. The annoyance rate is formulated with random fuzzy evaluation models, membership functions, and probability distributions [19]:

$$A(a_w) = \int_{x_{\min}}^{\infty} \frac{1}{\sqrt{2\pi} x \sigma} \exp\left[\frac{-\left[\ln\left(\frac{x}{a_w}\right) + 0.5\sigma^2\right]^2}{2\sigma^2} \right] v(x) dx \tag{5}$$

$$\sigma = \sqrt{\ln(1+\delta^2)} \tag{6}$$

$$v(x) = \begin{cases} 0, & x < x_{\min} \\ a\ln(x) + b, & x_{\min} \le x \le x_{\max} \\ 1, & x > x_{\max} \end{cases}, \tag{7}$$

where $x_{\min}$ is the lower limit of vibration that passengers cannot sense; x is the vibration acceleration; σ is the scale parameter; δ is the vibration parameter ranging from 0.19 to 0.31, generally set as 0.3; a and b are the constants; and $x_{\max}$ is the upper limit of vibration that passengers cannot tolerate. Although the sensation at various magnitudes of vibration depends on passengers' expectation and activities, ISO 2631-1 proposes an approximate

indication of likely reactions to various magnitudes. Based on our previous work [19], $x_{\min}$ and $x_{\max}$ are set as 0.135 and 2.5 m/s^2, and a and b are 0.4827 and 0.5577.

In this study, the annoyance rate is calculated with a specific length according to conventional road quality evaluation [12]. For example, the road profiles along the driving trajectories are divided into several segments with equal lengths, and the annoyance rate is calculated based on speeds and spatial road profiles in each segment. The intelligent speed control aims to confine the annoyance rate to below 20% to satisfy most passengers [19]. Specifically, the control strategies should ensure that 80% of passengers would be comfortable or not annoyed. The speeds satisfying the standard are regarded as prior knowledge of vertical comfort and directly induce the speed control of AVs. As shown in Figure 3, we calculate the annoyance rates at different speeds and record them at the end of each segment. The green circles indicate annoyance rates below 20%, while the red ones indicate annoyance rates above 20%. The maximum speed on each segment, maintaining the annoyance rate at 20%, is the maximum comfortable speed (MCS). The MCS provides prior knowledge of vertical comfort and works as a reference speed for real-time speed control.

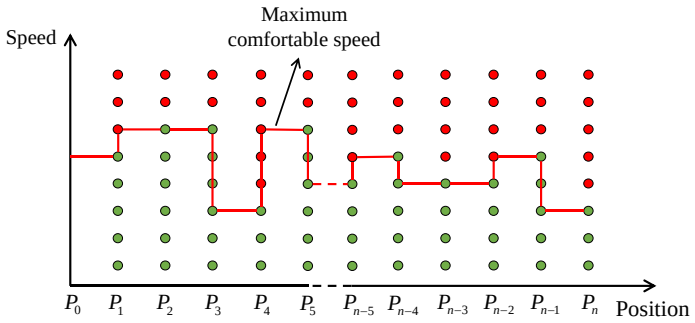

Figure 3. Schematic diagram of MCS calculation.

3. DRL-Based Intelligent Speed Control

This section proposes a DRL-based intelligent speed control for autonomous car-following on rough pavements. First, we set future road information and current traffic information in the state. We then design a reward function based on speed control objectives. Finally, we present the simulation settings and the structure of the DRL-based speed control model.

3.1. State and Action

In DRL, the agent selects an action based on the observed state. The variables in the state should provide sufficient information for the action selection to achieve the control objectives. For safety and efficiency, the relative speed and space between leading and following vehicles should be known. For longitudinal comfort, the previous acceleration limits the current action selection. For vertical comfort, prior knowledge of the MCS along planned driving trajectories provides information on acceptable speeds. Thus, the state is described by the previous acceleration $a(t-1)$, current speed $V_n(t)$, relative speed $\Delta V_{n-1,n}(t)$, clearance distance $S_{n-1,n}(t)$, and prior knowledge $V_p(t)$ for vertical comfort:

$$s(t) = \left[a(t-1), V_n(t), \Delta V_{n-1,n}(t), S_{n-1,n}(t), V_p(t)\right], \tag{8}$$

where $\Delta V_{n-1,n}(t) = V_n(t) - V_{n-1}(t)$, $V_n(t)$ is the speed of the leading vehicle, and $V_{n-1}(t)$ is the speed of the following vehicle (i.e., the AV); the prior knowledge $V_p(t) = \left\{V_p^0(t), V_p^1(t), \ldots, V_p^{N_p}(t)\right\}$ is sampled from the MCS with a certain distance interval to represent future vertical comfort information.

The action is longitudinal acceleration $a(t)$, which is selected in a continuous action space $[a_{\min}, a_{\max}]$; $a_{\min}$ and $a_{\max}$ are the minimum and maximum longitudinal accelerations, set as -3 and 3 m/s^2, respectively. When the longitudinal acceleration $a(t)$ is given by the

agent, the AV's speed $V(t)$, relative speed $\Delta V_{n-1,n}(t+1)$, and clearance distance $S_{n-1,n}(t)$ are updated in the next timestep:

$$V_n(t+1) = V_n(t) + a(t)\Delta T \tag{9}$$

$$\Delta V_{n-1,n}(t+1) = V_{n-1}(t+1) - V_n(t+1) \tag{10}$$

$$S_{n-1,n}(t+1) = S_{n-1,n}(t) + \frac{(\Delta V_{n-1,n}(t) + \Delta V_{n-1,n}(t+1))\Delta T}{2}, \tag{11}$$

where ΔT is the simulation sample time interval, usually set as 0.1 s.

3.2. Reward Function

In DRL, the agent aims to maximize the expected reward by adjusting the action selection. The reward function plays a crucial role in learning preferred speed control strategies. The reward function should be designed based on the objectives, including safety, efficiency, and ride comfort.

3.2.1. Safety

In dynamic traffic scenarios, safety is the most important element. The time to collision (TTC) is widely used to evaluate the risk of a rear-end crash in real time [21]. The TTC of a following AV is described as

$$TTC(t) = \begin{cases} -\frac{S_{n-1,n}(t)}{V_{n-1,n}(t)}, & V_n(t) < V_{n-1}(t) \\ \infty, & V_n(t) \geq V_{n-1}(t) \end{cases} \tag{12}$$

Specifically, a small TTC value denotes a high traffic crash risk. The TTC threshold should be determined to distinguish unsafe actions. A threshold varying from 1.5 to 5 s is recommended in different studies [4,21]. Based on the experimental results of Zhu et al. [4], the TTC threshold is set as 4 s for a good overall performance. The agent should be punished if the TTC is larger than 0 s and less than 4 s. The TTC feature R_{st} is expressed as

$$R_{st} = \begin{cases} -10, & 0 \leq TTC(t) \leq 4 \\ 0, & otherwise \end{cases} \tag{13}$$

Although R_{st} can punish potentially unsafe actions, the TTC values are simultaneously related to clearance distance and relative speed. A lack of sufficient space for emergency braking is also dangerous. Meanwhile, the following AV requires a reaction time for risk assessment, decision-making, and braking. Thus, the safe distance is used as a threshold to ensure sufficient space between vehicles. The agent should be punished when the clearance distance is less than the safe distance. The safe distance feature R_{sd} is described as

$$d_s = V_{n-1}(t) \cdot t_r + \frac{V_n(t)^2}{2a_d} - \frac{V_{n-1}(t)^2}{2a_d} \tag{14}$$

$$R_{sd} = \begin{cases} -10, & S_{n-1,n}(t) < d_s \\ 0, & S_{n-1,n}(t) \geq d_s \end{cases}, \tag{15}$$

where t_r is the reaction time of the following AV, which is set as 1 s in this study; a_d is the absolute maximum deceleration.

3.2.2. Efficiency

Efficient driving refers to a short-time headway. Time headway refers to the passed time between leading and following vehicles at a specific point. Maintaining time headway within acceptable limits contributes to a large road capacity. Since the recommended time headway differs between countries, we use the vehicle trajectory data of the Next Generation Simulation (NGSIM) project. A lognormal distribution was fitted based on the

extracted car-following events [4]. The reward for driving efficiency uses the probability density function of the lognormal distribution. When the time headway is within the limits, the agent can receive a positive reward, indicating that the time headway is preferred. If the time headway is too large or small, the reward is close to zero. The time headway feature R_{eh} is expressed as

$$R_{eh} = \frac{1}{h\sigma\sqrt{2\pi}} e^{\frac{-(\ln h-\mu)}{2\sigma^2}} \Big| \mu = 0.4226, \sigma = 0.4365, \tag{16}$$

where h is the time headway.

Since the training of DRL models usually begins with the random initialization, a large clearance distance should be punished in early training episodes to avoid useless exploration. The agent is thus guided to adjust the speed control policy in time to improve driving efficiency. When the clearance distance is less than the threshold, the time headway is used to evaluate driving efficiency. Otherwise, the agent is punished. The clearance distance feature R_{ed} is described as

$$R_{ed} = \begin{cases} -\frac{S_{n-1,n}(t)}{d_e}, & S_{n-1,n}(t) > d_e \\ 0, & S_{n-1,n}(t) \le d_e \end{cases}, \tag{17}$$

where d_e the threshold of the clearance distance.

3.2.3. Vertical Comfort

As described in Section 2, driving speeds impact vertical comfort, and the MCS provides vertical comfort information on oncoming roads. To confine discomfort, an AV should maintain its speed in the region $\left[0, V_p^0(t)\right]$, which only causes discomfort to a few passengers. When the driving speed is within this region, the action is acceptable for vertical comfort, and the feature is set as zero. The agent should receive a penalty when the driving speed is outside this region. In the penalty, the speed deviation from $V_p^0(t)$ is used to guide the driving speed adjustment. The penalty is divided by the desired speed deviation ΔV_e, which helps limit the speed deviation below the expected value. The vertical comfort feature R_v is constructed as

$$R_v = \begin{cases} \frac{\left(V_p^0(t)-V(t)\right)^2}{\Delta V_e{}^2}, & V(t) > V_p^0(t) \\ 0, & V(t) \le V_p^0(t) \end{cases} \tag{18}$$

3.2.4. Longitudinal Comfort

In longitudinal motion, small absolute values of jerk and acceleration contribute to longitudinal comfort [19,22]. Thus, longitudinal comfort is evaluated by the jerk $j(t)$ and acceleration $a(t)$. However, the largest absolute value of acceleration is 3 m/s^2, while that for jerk is 60 m/s^3. Since AVs on rough pavements should achieve relatively large acceleration to adapt to changing MCS, we divide jerk and acceleration by different base values for better speed control results. Meanwhile, the jerk is recommended not to exceed 2.94 m/s^3 to retain longitudinal comfort. Thus, we punish a jerk whose absolute value exceeds 2.94 m/s^3 with a penalty coefficient φ. The jerk and acceleration features (R_{lj} and R_{la}) are described as

$$j(t) = \frac{a(t) - a(t-1)}{\Delta T} \tag{19}$$

$$R_{lj} = -\begin{cases} -\varphi\frac{j(t)^2}{3600}, & |V(t)| \ge 2.94 \\ -\frac{j(t)^2}{3600}, & |V(t)| < 2.94 \end{cases} \tag{20}$$

$$R_{la} = -\frac{a(t)^2}{90} \tag{21}$$

3.2.5. Immediate Reward

For safe, efficient, and comfortable speed control on rough pavements, the immediate reward is the summation of the above reward items with weights:

$$r = w_1 R_{st} + w_2 R_{sd} + w_3 R_{eh} + w_4 R_{ed} + w_5 R_v + w_6 R_{lj} + w_7 R_{la}, \quad (22)$$

where w_1, w_2, w_3, w_4, w_5, w_6, and w_7 are weights. The weights are used to adjust the reward values to a similar magnitude.

3.3. DDPG Algorithm

3.3.1. Simulation Settings

Since Lillicrap et al. [23] first proposed the DDPG algorithm, it has been applied in various autonomous driving environments. The driving scenarios mainly include car following [4,24] and lane changing [25]. However, the scenario of driving on real-world rough pavements is seldom considered. Du et al. first used the DDPG algorithm to solve the speed control problem on real-world rough pavements; however, the behavior of the vehicle in front was ignored [12]. Based on the work in [12], we further extend the environment of car-following tasks with rough pavements. Like most DRL algorithms, the DDPG algorithm models the speed control problem using the interactions between agents and environments. In this study, the agent is an AV. The main elements of the environment include rough pavements, leading vehicles, and following vehicles. Rather than raw road profiles detected by sensors, we conduct vehicle vibration simulation and model rough pavements using the MCS corresponding to the road profiles. In such a way, the environment is simplified. We set the leading vehicles' driving speeds and locations using empirical human data. Since the road profiles and dynamic traffic are usually detected separately, we combine the data from two irrelevant datasets to establish a stochastic environment. The AV's kinematic model is described in Equations (9)–(11).

To simulate car following on rough pavements, we elaborated on the simulation settings in the environment here. When an AV enters the road, it receives road and traffic information via vehicle-to-infrastructure and vehicle-to-vehicle communication. Since this study focuses on vehicle control strategies, we assume that the AV drives under ideal communication conditions to follow the settings in most studies [12]. Thus, the future MCS and current leading vehicle information are sent to the AV from the environment. Rough pavements and leading vehicles are randomly extracted from the datasets to ensure randomness in the environment. However, the lengths of rough pavements and empirical human data differ considerably. The length of a real-world rough pavement is generally hundreds of meters, while the length of empirical human data is only tens of seconds. Thus, we assume the AV starts at a random location, and the location and speed of the leading vehicle are set according to the sampled car-following event. When the AV reaches the end of the roads or the car-following event ends, the termination condition is satisfied. The initial speed of the AV is set as the speed of the following vehicle for a relatively good beginning to avoid unnecessary exploration [12].

3.3.2. DDPG Structure

The structure of the DDPG-based speed control model is depicted in Figure 4. The DDPG model comprises two main components: an environment and an agent. The simulation settings illustrated above are used here. The agent has an actor-critic structure. The main and target networks share the same network structure. Specifically, the actor and critic networks in the main network are updated using the policy gradient and loss function in real time, while those in the target networks are updated using soft replacement with the parameters in the main networks. Regarding the structure of networks, the number of layers and neurons is usually selected based on the complexity of the reward function and state. For stable convergence, a large and deep neural network is preferred. A light model is required for a low computational burden and real-world application. Thus, we

set the neurons in layers as 50-30-20 units based on extensive trials to balance training performance and computation time. Each neuron in the hidden layer usually uses the ReLU activation function. The final layers in the actor networks use the tanh activation function and are multiplied by 3 to map the output of the actor networks to the range [−3,3].

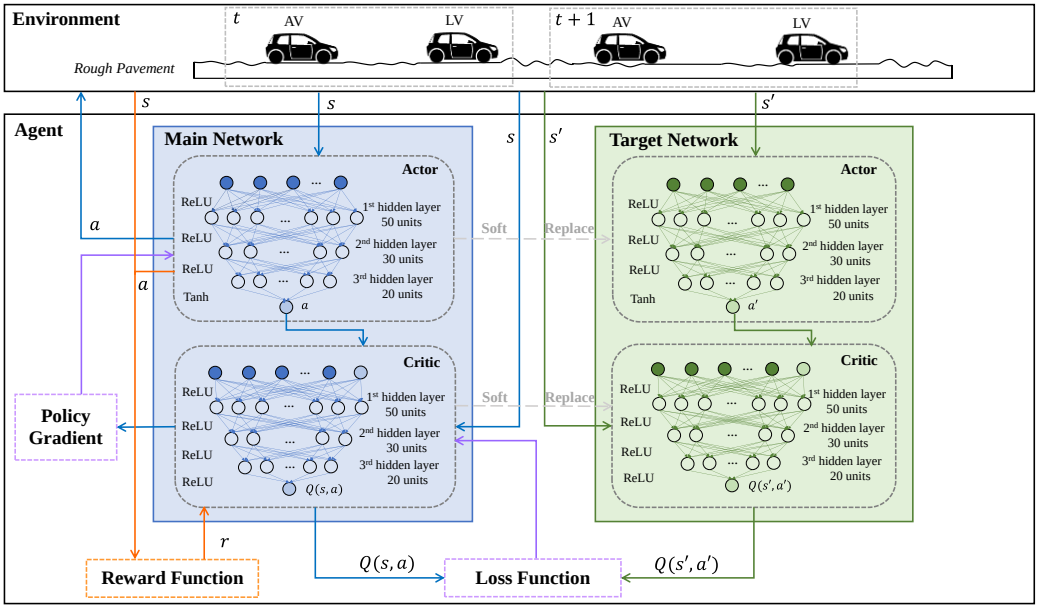

Figure 4. Structure of the DDPG model.

The actor-network outputs action (longitudinal acceleration) based on the state at each timestep. The action is conducted in the environment and changes the state in the next timestep. The reward is calculated using the reward function proposed in Section 3.2. The transition $\langle s_t, a_t, r_t, s_{t+1} \rangle$ is stored in the experience pool. When the pool is full, network training begins. The training process is described as follows. Initially, the critic and actor networks are initialized. At each timestep t, the actor networks input the state and output an action with a noise: $a_t = \mu(s_t|\theta^\mu) + N_t$. During training, the noise N_t is discounted with a factor. After convergence, the noise should be close to zero.

Although the reward function has punished situations with small TTC values and clearance distances, unsafe actions may still occur. However, unsafe actions are not acceptable in the application. Thus, following the setting in [4], we add a collision avoidance strategy for the action selection in training and testing. When the clearance distance is less than the safe distance, the AV takes a full deceleration of -3 m/s^2. Otherwise, the action is the output of the actor-network. The collision avoidance strategy is described as

$$a_t = \begin{cases} -3, & S_{n-1,n}(t) < d_s \\ \text{DDPG model output}, & otherwise \end{cases} \tag{23}$$

The critic networks input state s_t and action a_t and output $Q(s_t, a_t)$ to estimate the goodness of the action selection. The main critic network updates by minimizing the loss function L:

$$L = \frac{1}{N}\sum_i \left(r_i + \gamma Q'\left(s_{i+1}, \mu'\left(s_{i+1}\middle|\theta^{\mu'}\right)\middle|\theta^{Q'}\right) - Q\left(s_i, a_i\middle|\theta^Q\right)\right)^2 \tag{24}$$

where N is the number of samples; r is the reward; γ is the discount factor; $Q\left(s, a\middle|\theta^Q\right)$ is the main critic network; $\mu(s|\theta^\mu)$ is the main actor network; θ^Q and θ^μ are the parameters of the main critic and actor networks, respectively; $Q'\left(s, a\middle|\theta^{Q'}\right)$ and $\mu'\left(s\middle|\theta^{\mu'}\right)$ are the target critic and actor networks, respectively; $\theta^{Q'}$ and $\theta^{\mu'}$ are the parameters of the target critic and actor networks, respectively.

The main actor network then updates parameters using the policy gradient $\nabla_{\theta^\mu} J$ with the gradients $\nabla_a Q(s,a)$ calculated by the main critic network:

$$\nabla_{\theta^\mu} J = \frac{1}{N}\sum_i \nabla_a Q\left(s, a\middle|\theta^Q\right)\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^\mu}\mu(s|\theta^\mu)|_{s_i} \quad (25)$$

The target networks are updated slowly by tracking the main networks with $\tau \ll 1$:

$$\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'} \quad (26)$$

$$\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'} \quad (27)$$

4. Experiments and Results

In this section, we conduct experiments to show the performance of the proposed intelligent speed control. First, we introduce the dataset for simulating leading vehicles and rough pavements. Then, we train a DDPG model and analyze its training performance. Furthermore, we formulate an MPC-based adaptive cruise control (ACC) as the baseline speed control. The MPC is solved and implemented via CasADi in MATLAB 2020a [26,27]. Finally, we compare the driving performances of the DDPG model and the MPC baseline. All the experiments are executed on a computer with Intel Core i7-5600 at 2.60 GHz and 12 GB RAM.

4.1. Data Introduction

To simulate car-following behavior on rough pavements, we use the NGSIM trajectory data and a rough pavement dataset to establish a stochastic environment [4,12]. For an AV, the proposed DRL-based intelligent speed control outputs its acceleration based on the leading vehicle motions, following vehicle (AV) motions, and pavement conditions. During training, the DRL-based intelligent speed control can adjust control strategies adaptively according to changing conditions. In this study, NGSIM trajectory data and the rough pavement dataset are used as an example to train models and verify the feasibility of the proposed intelligent speed control approach. The trajectory and pavement data can be replaced by other datasets.

The NGSIM trajectory data were retrieved from the eastbound I-80 in Emeryville, California, in April 2005. The detection region was 500 m long and covered six lanes. The detection time of the trajectories comprises three spans of time in the afternoon: 4:00–4:15, 5:00–5:15, and 5:15–5:30, which contain the evolutionary process of congestion. The original trajectory data provide locations of vehicles with a detection frequency of 10 Hz. The dataset is reconstructed to enhance the data quality for further investigation, and car-following events are extracted. In this study, 1341 car-following events extracted from the original dataset are used and called the NGSIM data in the following sections. The training set contains 938 events, and the testing dataset contains 403 events.

For pavement data, we collected road information in March and April 2019, covering 11 districts in Shanghai, China (see Figure 5). The road information mainly includes road names, districts, pavement roughness, and road profiles. The road information was detected by advanced onboard sensors, including LiDAR, accelerometers, and GPS, under the operation of manual vehicles. The resolution of detected road profiles detected by LiDAR was 0.25 m. Based on unexpected vibration detected by accelerometers, the potential damage was located by GPS and captured using wavelet analysis [7]. We sampled 110 rough pavements in this dataset to form a rough pavement dataset for model training and testing.

Figure 5. Detection regions of road information.

4.2. Training Results

We trained a DDPG-based speed control model using the training set of the NGSIM data and rough pavement dataset. At each episode, the environment is reset using data sampled randomly from the datasets, as mentioned in Section 3.3. The preview length of the future MCS is set as 50 m, for example. The resolution of the future MCS is 1 m. According to the definition of the state in Section 3.1, the state has 54 variables. Since training a DRL-based model is time-consuming, the maximum timestep in each episode is set as 1000, and the simulation resolution is 0.1 s. For full exploration, the capacity of the reply buffer is 20,000, and the batch size is 1024. The learning rates of the actor and critic networks are set as 0.0001 and 0.001. The discount factor for calculating the cumulated reward is 0.9. All the weights in Equation (22) are set as 1 to assign equal importance to all the speed control objectives.

Figure 6 illustrates the training process with the episode mean rewards in translucent colors and the rolling mean reward in solid colors. The episode mean reward is the mean value of rewards received in an episode, while the rolling mean reward is the mean value of mean episode rewards within a rolling window. The rolling window is ten episodes. As shown in Figure 6a, the training trajectory of the mean episode reward has a convergence tendency after 400 episodes. In Figure 6b, the headway reward is large in early episodes but decreases later. This is because the agent should balance multiple speed control objectives. Thus, in Figure 6b–d, the longitudinal comfort feature converges after 400 episodes, while there are fluctuations in efficiency and vertical comfort features, indicating that the agent learns longitudinal comfort first and then tries its best to balance comfort and efficiency for higher rewards.

4.3. Comparison with MPC

4.3.1. MPC-Based ACC Baseline

MPC is the most common speed control method to achieve multi-objective car-following behavior, including safety, efficiency, and comfort [4,28,29]. At each timestep, MPC solves an optimal control problem in a prediction horizon and generates an acceleration sequence. The first in the sequence is then conducted. This optimization process is repeated until the termination conditions are satisfied. Since MPC-based speed control can handle constraints and perform predictive control, it functions as a baseline for performance comparison with the DDPG model [12]. The kinematic point-mass model mentioned in Section 3.1 is described in a vector form:

$$x(t+1) = Ax(t) + Bu(t), \tag{28}$$

where t is the timestep, $x(t) = [S_{n-1,n}(t), \Delta V_{n-1,n}(t), V_{n-1,n}(t)]^T$, $u(t) = a(t)$, $A = \begin{bmatrix} 1 & \Delta T & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$, and $B = \begin{bmatrix} -0.5\Delta T^2 \\ -\Delta T \\ \Delta T \end{bmatrix}$.

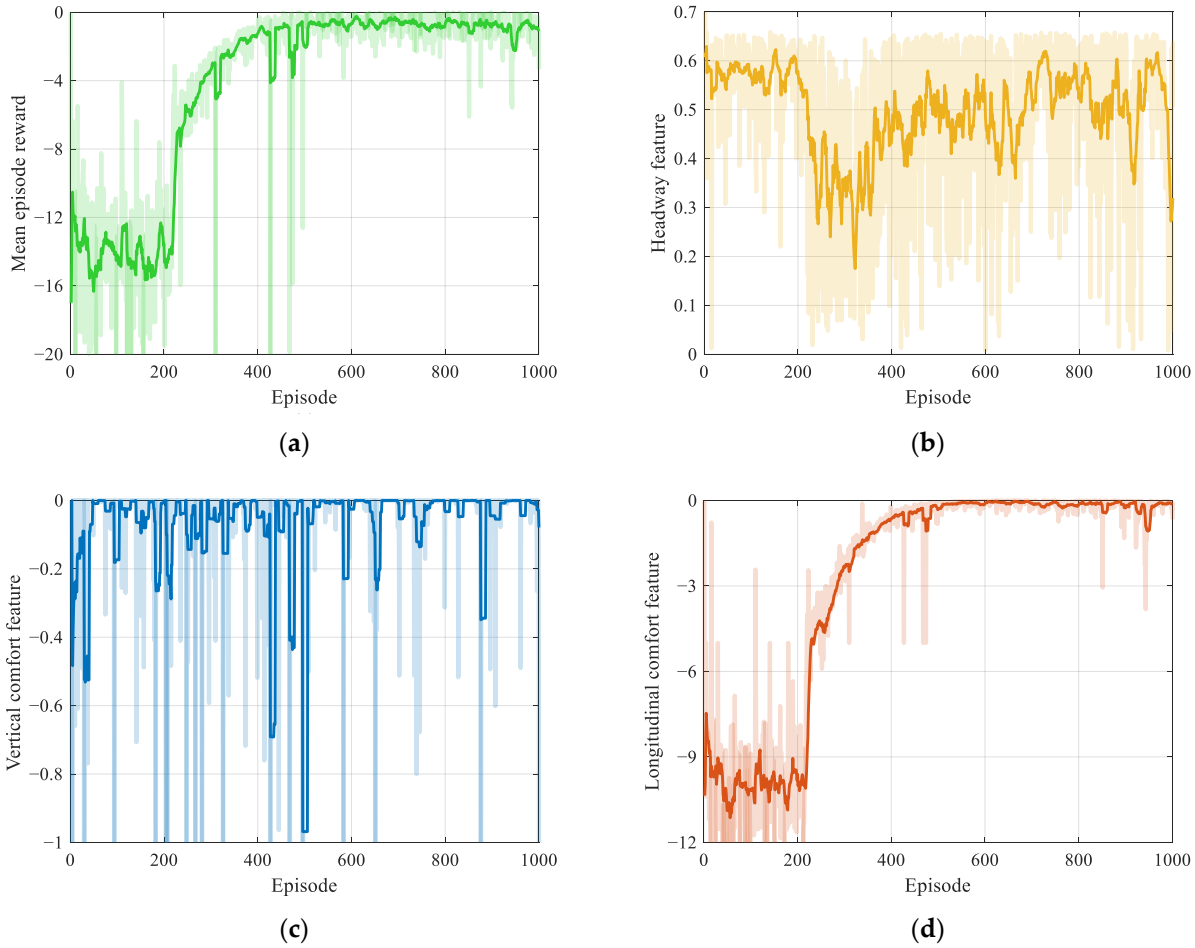

Figure 6. Training trajectories of the DDPG model: (**a**) mean episode reward, (**b**) headway feature, (**c**) vertical comfort feature, and (**d**) longitudinal comfort feature.

The MPC-based ACC baseline is implemented by optimizing the problem of safe, efficient, and comfortable speed control under constraint conditions. For comparison, the objective function and constraint conditions should refer to the DDPG model. In this study, we follow the modeling of the MPC-based ACC in [4]. For safety and efficiency, AVs follow the leading vehicles with the desired distance $\widetilde{S}_{n-1,n}$ and a small relative speed $\Delta V_{n-1,n}$. For comfort, the deviations between speed and the current MCS and the absolute jerk and acceleration values should be minimized. Therefore, a constrained MPC formulation is defined as

$$\sum_{t=0}^{N-1}\left[W_1\left(\frac{S_{n-1,n}(t)-\widetilde{S}_{n-1,n}(t)}{S_{\max}}\right)^2+W_2\left(\frac{\Delta V_{n-1,n}(t)}{\Delta V_{\max}}\right)^2+W_3\left(\frac{V(t)-V_p^0(t)}{\Delta V_e}\right)^2+W_4\left(\frac{j(t)}{j_{\max}}\right)^2+W_5\left(\frac{a(t)}{\alpha}\right)^2\right] \quad (29)$$

$$s.t.\ x(t+1)=Ax(t)+Bu(t) \quad (30)$$

$$0<V(t)<V_{\max} \quad (31)$$

$$-3<a(t)<3 \quad (32)$$

where N is the prediction horizon (N= 30 in this study); the desired distance $\widetilde{S}_{n-1,n}(t)=V_n(t)h(t)$; and $S_{\max}$, $\Delta V_{\max}$, ΔV_e, $j_{\max}$, and α are the constants for normalization. Specifically, $S_{\max}$ and $\Delta V_{\max}$ are the maximum acceptable clearance space and relative speed, set as 15 m and 8 m/s^2, respectively; ΔV_e is the expected relative speed, set as 3 m/s; $j_{\max}$ is the maximum absolute value of longitudinal jerk, set as 60 m/s^3; α^2 is the base value

and is set as 90. The weights are set as $W_1 = 1$, $W_2 = 1$, $W_3 = 1$, $W_4 = 1$, and $W_5 = 1$; $u = [a(0), a(1), \ldots, a(N-1)]$ is the solved action sequence in each timestep, and only the first action $a(0)$ is implemented. This process is repeated until the termination conditions are reached.

4.3.2. Comparison Results

To compare driving performances, we conducted experiments using a sampled rough pavement and the testing set of the NGSIM data. Our rationale for this was that 44,330 combinations of rough pavements and leading vehicles exist when AVs start at the same location on each pavement. Since the driving speeds of leading vehicles in the testing set range from 0.0722 to 61.0570 m/s, the deviation between AVs' speeds and the MCS varies, although the same pavement is used. Since the number of combinations of rough pavements and leading vehicles is large, we sampled an extremely rough pavement from the dataset for testing. The sampled road profiles of left and right wheels, annoyance rate analysis, and MCS of the Yangshupu Road is shown in Figure 7. Specifically, the MCS is fitted using B-spline interpolation to provide precise information for speed tracking, called the fitted MCS [12].

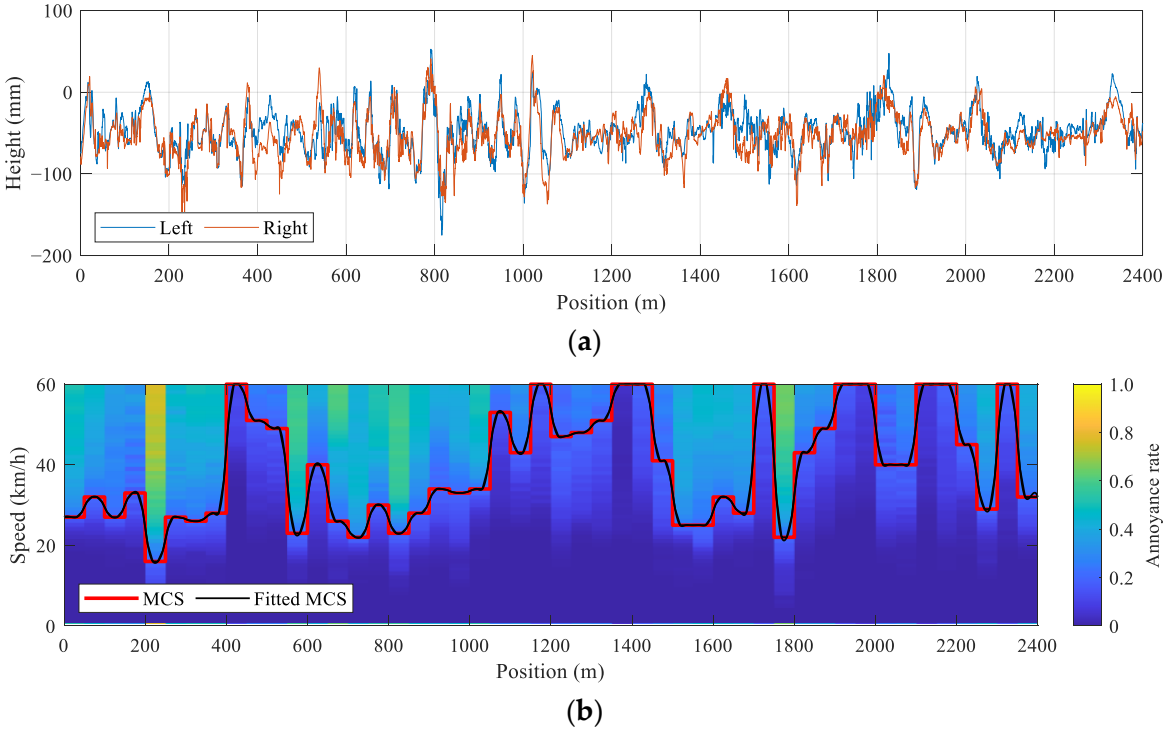

Figure 7. Sampled rough pavement for testing: (**a**) road profiles, and (**b**) annoyance rate and MCS.

In the testing, we assume that all the AVs start at a location 0 m on Yangshupu Road, and the leading vehicle is set using the speeds and locations in the testing set. The number of trials is 403. The computation times of the DDPG model and MPC baseline are 125.56 s and 1922.77 s, respectively. Compared to the rolling optimization used in MPC, the DDPG-based speed control exploits linear computations in the networks. The computational efficiency is improved by 93.47%. As shown in Figure 8, we further compare the driving performance using the TTC, time headway, annoyance rate, and jerk. Since the TTC values can be infinity, we pay attention to the TTC values in the region of [0, 50] for analysis and comparison. Similarly, we only show the time headway below 8 s in Figure 8. Figure 8a demonstrates that the MPC baseline has more large TTC values while the DDPG model has a small proportion of small TTC values, indicating that the DDPG model can effectively reduce the risk of rear-end crash and retain safety. Figure 8b shows that the DDPG model

has better driving efficiency than the MPC baseline, where almost 80% of the time headway values are less than 2 s. Figure 8c shows that both the DDPG model and MPC baseline can adjust speed according to pavement conditions. Interestingly, the highest annoyance rate of the DDPG model is less than the MPC baseline, but the annoyance rates of the DDPG model on some pavements are slightly larger due to the higher driving efficiency. Figure 8d demonstrates that the DDPG model can limit the absolute value of longitudinal jerk below 2.94 m/s^3 more effectively, indicating that the DDPG model has better longitudinal comfort. The DDPG model can improve driving efficiency, longitudinal comfort, and vertical comfort by 26.99%, 58.33%, and 6.05%, respectively.

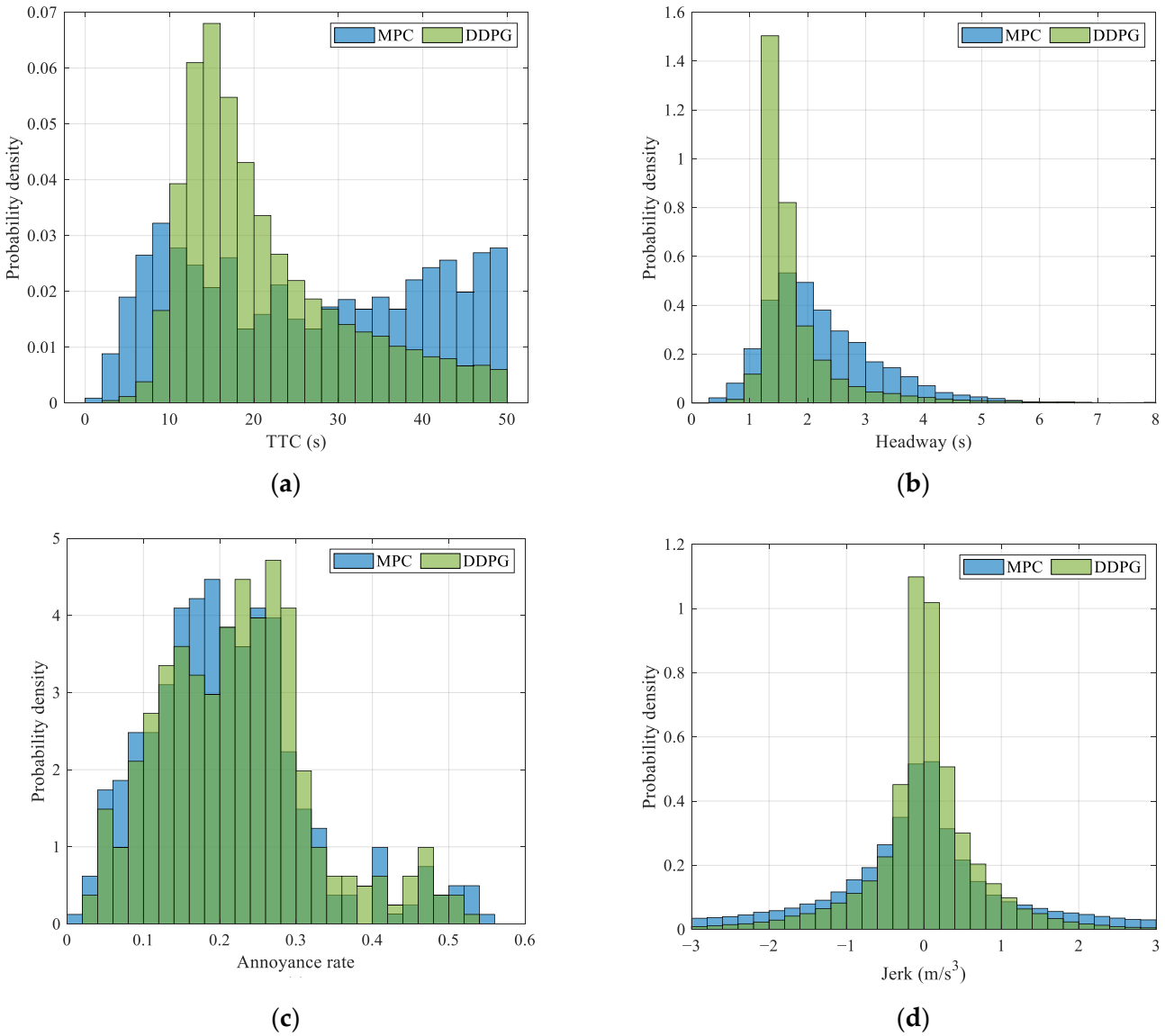

Figure 8. Probability density distribution of (**a**) TTC, (**b**) time headway, (**c**) annoyance rate, and (**d**) jerk.

We further tested the model with different starting points to show the details of the speed control results. In Figure 9a, the speeds of the leading vehicle are below the fitted MCS, indicating that the main task of the AV is to follow the leading vehicle. As shown in Figure 9b,c, the DDPG model can generate lower absolute values of jerk and acceleration. Thus, Figure 9a indicates that the speed profile generated by the DDPG model is smoother. Consequently, the space of the DDPG model is much larger than the MPC baseline. Unlike the example in Figure 9, some of the fitted MCS values in Figure 10a are below the leading

vehicle's speeds. The AV should balance driving efficiency and ride comfort. Figure 10d shows that the MPC baseline first follows the leading vehicle at a certain clearance distance and then adjusts its speed to improve vertical comfort. Compared to the MPC baseline, the DDPG model can maintain a relatively large clearance distance for safety. Meanwhile, the DDPG model has lower absolute values of jerk and acceleration when following the leading vehicle. With sufficient space between two vehicles, the AV can adjust its speed in advance for better vertical comfort in future situations (see Figure 10b,c).

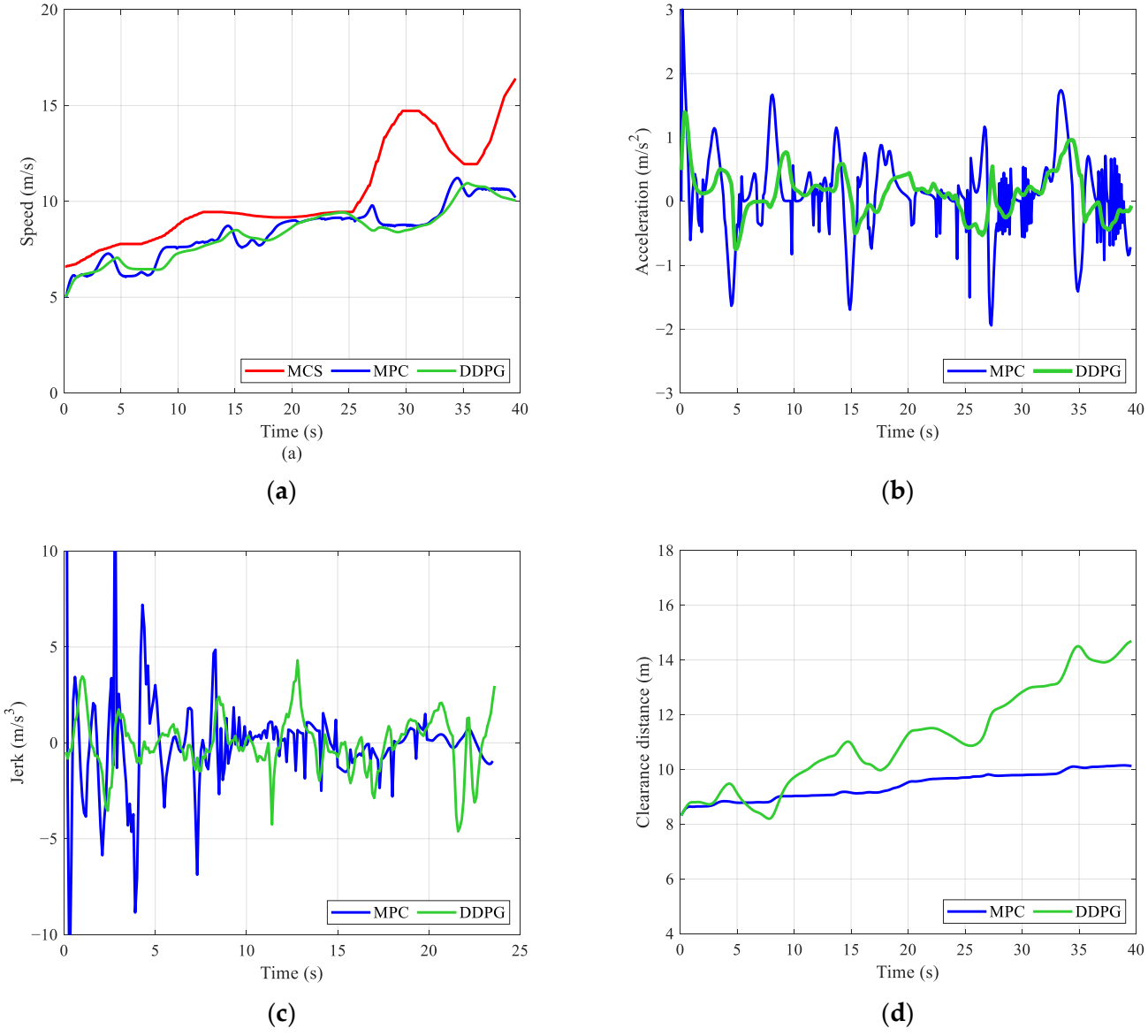

Figure 9. Speed control results for the AV with the leading vehicle No. 107 and a starting point at location 839 m on the sampled rough pavement: (**a**) speed, (**b**) acceleration, (**c**) jerk, and (**d**) clearance distance.

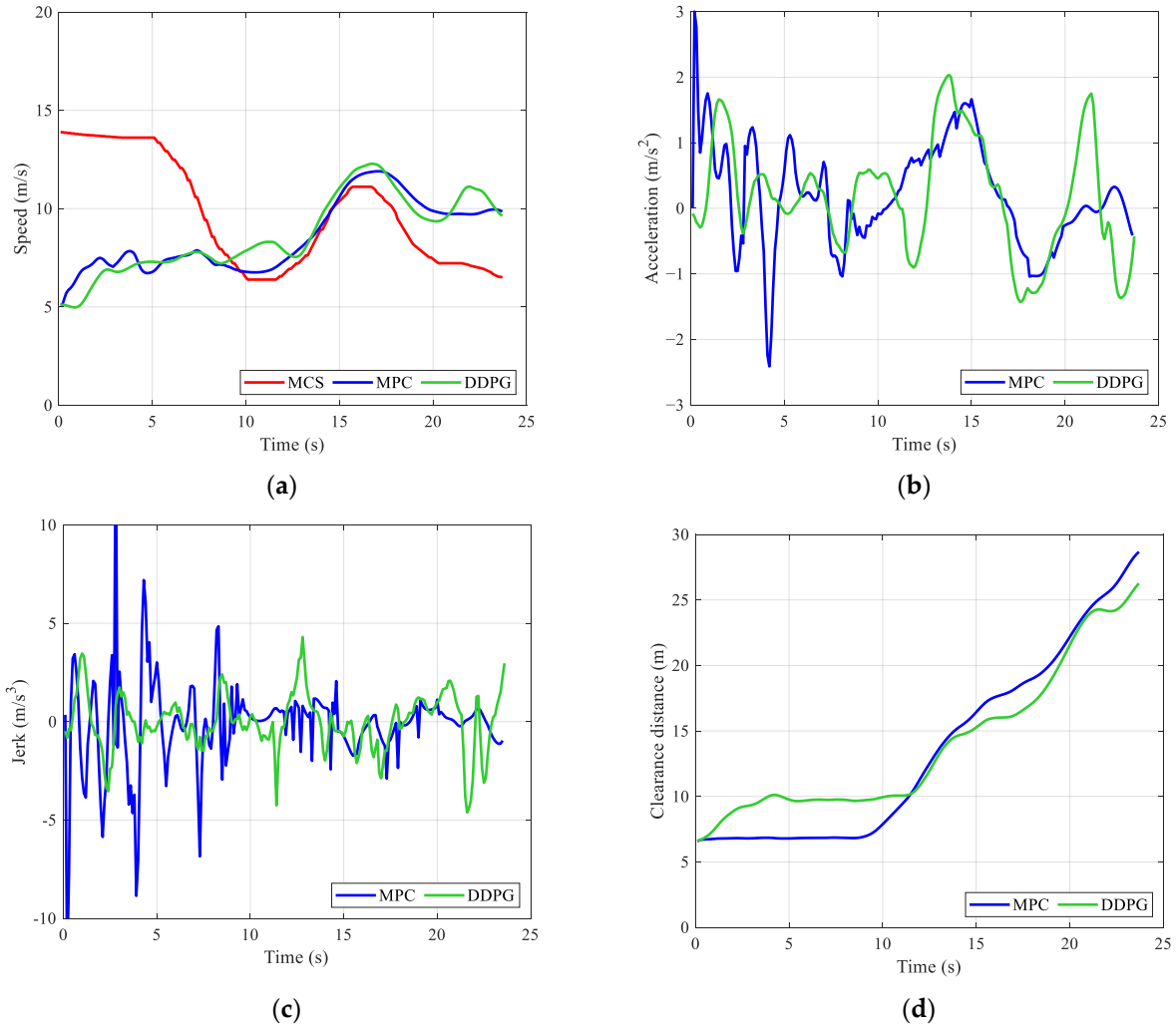

Figure 10. Speed control results for the AV with the leading vehicle No. 326 and a starting point at location 500 m on the sampled rough pavement: (**a**) speed, (**b**) acceleration, (**c**) jerk, and (**d**) clearance distance.

5. Conclusions

To summarize, this study proposes an intelligent speed control approach for autonomous car following on rough pavements in a cooperative vehicle infrastructure system using deep reinforcement learning (DRL). In experiments, the car-following events in the NGSIM data and road profiles in the rough pavement dataset are used for model training and testing. The experimental results show that the proposed DRL-based speed control has a better driving performance than a model predictive control baseline. Specifically, the DRL-based speed control can improve computational efficiency, driving efficiency, longitudinal comfort, and vertical comfort in car following by 93.47%, 26.99%, 58.33%, and 6.05%, respectively. The results indicate that the proposed intelligent speed control can contribute to autonomous driving on rough pavements and has excellent potential for practical application.

In our future research, we plan to extend driving scenarios with lane-changing behavior. Although lane changing does not have the highest priority in conservative driving strategies, it remains a challenging task with the requirements of safe and comfortable trajectory planning [25,30]. Meanwhile, the proposed intelligent speed control approach can be applied to several AVs with multi-agent RL and used to improve the driving performance in an environment of fully or partially AVs [31]. Moreover, transfer learning and

ensemble learning can be used to improve the training efficiency, robustness, and reliability of DRL models [7,32].

Author Contributions: Conceptualization, J.C. and C.Z.; methodology, J.C. and C.Z.; software, J.C.; validation, J.C., X.Z. and C.Z.; data curation, S.J. and Y.D.; writing—original draft preparation, J.C.; writing—review and editing, C.Z., S.J. and Z.L.; visualization, J.C.; supervision, C.Z.; project administration, C.Z.; funding acquisition, Y.D. and C.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China under Grant 2021YFB1600403, and in part by the Innovation Program of Shanghai Municipal Education Commission under Grant 2021-01-07-00-07-E00092, and in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: All authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interests.

References

1. Bellem, H.; Thiel, B.; Schrauf, M.; Krems, J.F. Comfort in Automated Driving: An Analysis of Preferences for Different Automated Driving Styles and Their Dependence on Personality Traits. *Transp. Res. Pt. F-Traffic Psychol. Behav.* **2018**, *55*, 90–100. [CrossRef]
2. Paddeu, D.; Parkhurst, G.; Shergold, I. Passenger Comfort and Trust on First-Time Use of a Shared Autonomous Shuttle Vehicle. *Transp. Res. Pt. C-Emerg. Technol.* **2020**, *115*, 102604. [CrossRef]
3. Sharma, R.C.; Sharma, S.; Sharma, S.K.; Sharma, N.; Singh, G. Analysis of Bio-Dynamic Model of Seated Human Subject and Optimization of the Passenger Ride Comfort for Three-Wheel Vehicle Using Random Search Technique. *Proc. Inst. Mech. Eng. Pt. K-J. Multi-Body Dyn.* **2021**, *235*, 106–121. [CrossRef]
4. Zhu, M.; Wang, Y.; Pu, Z.; Hu, J.; Wang, X.; Ke, R. Safe, Efficient, and Comfortable Velocity Control Based on Reinforcement Learning for Autonomous Driving. *Transp. Res. Pt. C-Emerg. Technol.* **2020**, *117*, 102662. [CrossRef]
5. Wu, J.; Zhou, H.; Liu, Z.; Gu, M. Ride Comfort Optimization via Speed Planning and Preview Semi-Active Suspension Control for Autonomous Vehicles on Uneven Roads. *IEEE Trans. Veh. Technol.* **2020**, *69*, 8343–8355. [CrossRef]
6. Treiber, M.; Hennecke, A.; Helbing, D. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Phys. Rev. E* **2000**, *62*, 1805. [CrossRef]
7. Du, Y.; Chen, J.; Zhao, C.; Liao, F.; Zhu, M. A Hierarchical Framework for Improving Ride Comfort of Autonomous Vehicles via Deep Reinforcement Learning with External Knowledge. *Comput.-Aided Civ. Infrastruct. Eng.* **2022**. [CrossRef]
8. Zhu, M.; Wang, X.; Tarko, A.; Fang, S. Modeling Car-Following Behavior on Urban Expressways in Shanghai: A Naturalistic Driving Study. *Transp. Res. Pt. C-Emerg. Technol.* **2018**, *93*, 425–445. [CrossRef]
9. Wu, Y.; Tan, H.; Qin, L.; Ran, B. Differential Variable Speed Limits Control for Freeway Recurrent Bottlenecks via Deep Actor-Critic Algorithm. *Transp. Res. Pt. C-Emerg. Technol.* **2020**, *117*, 102649. [CrossRef]
10. Mao, C.; Liu, Y.; Shen, Z.-J.M. Dispatch of Autonomous Vehicles for Taxi Services: A Deep Reinforcement Learning Approach. *Transp. Res. Pt. C-Emerg. Technol.* **2020**, *115*, 102626. [CrossRef]
11. Buechel, M.; Knoll, A. Deep Reinforcement Learning for Predictive Longitudinal Control of Automated Vehicles. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; pp. 2391–2397.
12. Du, Y.; Chen, J.; Zhao, C.; Liu, C.; Liao, F.; Chan, C.-Y. Comfortable and Energy-Efficient Speed Control of Autonomous Vehicles on Rough Pavements Using Deep Reinforcement Learning. *Transp. Res. Pt. C-Emerg. Technol.* **2022**, *134*, 103489. [CrossRef]
13. Du, Y.; Qin, B.; Zhao, C.; Zhu, Y.; Cao, J.; Ji, Y. A Novel Spatio-Temporal Synchronization Method of Roadside Asynchronous MMW Radar-Camera for Sensor Fusion. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 22278–22289. [CrossRef]
14. Zhao, C.; Song, A.; Du, Y.; Yang, B. TrajGAT: A Map-Embedded Graph Attention Network for Real-Time Vehicle Trajectory Imputation of Roadside Perception. *Transp. Res. Pt. C-Emerg. Technol.* **2022**, *142*, 103787. [CrossRef]
15. Zhao, C.; Song, A.; Zhu, Y.; Jiang, S.; Liao, F.; Du, Y. Data-Driven Indoor Positioning Correction for Infrastructure-Enabled Autonomous Driving Systems: A Lifelong Framework. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–14. [CrossRef]
16. Ji, Y.; Ni, L.; Zhao, C.; Lei, C.; Du, Y.; Wang, W. TriPField: A 3D Potential Field Model and Its Applications to Local Path Planning of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, 1–14. [CrossRef]
17. Zhao, C.; Ding, D.; Du, Z.; Shi, Y.; Su, G.; Yu, S. Analysis of Perception Accuracy of Roadside Millimeter-Wave Radar for Traffic Risk Assessment and Early Warning Systems. *Int. J. Environ. Res. Public Health* **2023**, 1–27.

18. Cantisani, G.; Loprencipe, G. Road Roughness and Whole Body Vibration: Evaluation Tools and Comfort Limits. *J. Transp. Eng.* **2010**, *136*, 818–826. [CrossRef]
19. Du, Y.; Liu, C.; Li, Y. Velocity Control Strategies to Improve Automated Vehicle Driving Comfort. *IEEE Intell. Transp. Syst. Mag.* **2018**, *10*, 8–18. [CrossRef]
20. International Standards Organization. *Mechanical Vibration and Shock-Evaluation of Human Exposure to Whole-Body Vibration-Part 1: General Requirements*; ISO: Geneva, Switzerland, 1997.
21. Li, Y.; Xu, C.; Xing, L.; Wang, W. Integrated Cooperative Adaptive Cruise and Variable Speed Limit Controls for Reducing Rear-End Collision Risks near Freeway Bottlenecks Based on Micro-Simulations. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 3157–3167. [CrossRef]
22. Hu, J.; Shao, Y.; Sun, Z.; Wang, M.; Bared, J.; Huang, P. Integrated Optimal Eco-Driving on Rolling Terrain for Hybrid Electric Vehicle with Vehicle-Infrastructure Communication. *Transp. Res. Pt. C-Emerg. Technol.* **2016**, *68*, 228–244. [CrossRef]
23. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1509.02971.
24. Yan, R.; Jiang, R.; Jia, B.; Yang, D.; Huang, J. Hybrid Car-Following Strategy Based on Deep Deterministic Policy Gradient and Cooperative Adaptive Cruise Control. *arXiv* **2021**, arXiv:2103.03796. [CrossRef]
25. Wang, P.; Li, H.; Chan, C.-Y. Continuous Control for Automated Lane Change Behavior Based on Deep Deterministic Policy Gradient Algorithm. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; IEEE Press: Piscataway, NJ, USA, 2019; pp. 1454–1460.
26. Andersson, J.A.; Gillis, J.; Horn, G.; Rawlings, J.B.; Diehl, M. CasADi: A Software Framework for Nonlinear Optimization and Optimal Control. *Math. Program. Comput.* **2019**, *11*, 1–36. [CrossRef]
27. Zhao, C.; Liao, F.; Li, X.; Du, Y. Macroscopic Modeling and Dynamic Control of On-Street Cruising-for-Parking of Autonomous Vehicles in a Multi-Region Urban Road Network. *Transp. Res. Pt. C-Emerg. Technol.* **2021**, *128*, 103176. [CrossRef]
28. Li, S.E.; Jia, Z.; Li, K.; Cheng, B. Fast Online Computation of a Model Predictive Controller and Its Application to Fuel Economy–Oriented Adaptive Cruise Control. *IEEE Trans. Intell. Transp. Syst.* **2014**, *16*, 1199–1209. [CrossRef]
29. Takahama, T.; Akasaka, D. Model Predictive Control Approach to Design Practical Adaptive Cruise Control for Traffic Jam. *Int. J. Automot. Eng.* **2018**, *9*, 99–104. [CrossRef]
30. Zhao, C.; Zhu, Y.; Du, Y.; Liao, F.; Chan, C.-Y. A Novel Direct Trajectory Planning Approach Based on Generative Adversarial Networks and Rapidly-Exploring Random Tree. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 17910–17921. [CrossRef]
31. Zhang, X.; Zhao, C.; Liao, F.; Li, X.; Du, Y. Online Parking Assignment in an Environment of Partially Connected Vehicles: A Multi-Agent Deep Reinforcement Learning Approach. *Transp. Res. Pt. C-Emerg. Technol.* **2022**, *138*, 103624. [CrossRef]
32. Lee, K.; Laskin, M.; Srinivas, A.; Abbeel, P. Sunrise: A Simple Unified Framework for Ensemble Learning in Deep Reinforcement Learning. In Proceedings of the 38th International Conference on Machine Learning (PMLR), Online, 18–24 July 2021; pp. 6131–6141.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.