IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 24, 2020, accepted March 10, 2020, date of publication March 23, 2020, date of current version April 7, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982702

DDPG-Based Decision-Making Strategy of Adaptive Cruising for Heavy Vehicles Considering Stability

MING SUN[1], WEIQIANG ZHAO[1], GUANGHAO SONG[1], ZHIGEN NIE[2], XIAOJIAN HAN[1], AND YANG LIU[1]

[1]State Key Laboratory of Automotive Simulation and Control, Jilin University, Changchun 130022, China
[2]Department of Transportation Engineering, Kunming University of Science and Technology, Kunming 650031, China

Corresponding author: Weiqiang Zhao (zwq@ jlu.edu.cn)

ABSTRACT The decision-making system of intelligent vehicles is the core component of an advanced driving system for both passenger vehicles and commercial vehicles. Finding ways to improve decision-making strategies to suit the complex and unfamiliar environments is a standing problem for traditional rule-based methods. This paper proposes a semi-rule-based decision-making strategy for heavy intelligent vehicles based on the Deep Deterministic Policy Gradient algorithm. Firstly, according to the car-following characteristics, the problems of high dimensions and a large amount of data in vehicle action space and state space are solved by dimension reduction and interval reduction to accelerate the training process. Subsequently, an accurate three-axle vehicle load model is established to calculate the load transfer rate value and carry out active control to increase the roll stability of heavy vehicles at high-speed corners. Furthermore, the Deep Deterministic Policy Gradient algorithm is developed based on the reward function and update function to achieve adaptive cruise control objectives for heavy vehicles on different curvature roads. Finally, the effectiveness and robustness of the algorithm are verified through simulation experiments.

INDEX TERMS Adaptive cruise control, autonomous driving control system, decision-making algorithm, deep reinforcement learning, heavy vehicle, vehicle stability.

I. INTRODUCTION

The autonomous vehicle driving control is a promising solution for increased road traffic accidents [1]. Decision-making plays the role of the brain in such intelligently controlled vehicles and is one of the key technologies involved in this field [2]. Adaptive cruise control is a crucial driving assistance technology, and the decision-making performance implemented in such a system directly affects vehicle safety and traffic efficiency [3], [4].

Traditional Adaptive Cruise Control (ACC) decision-making strategies are designed based on rules, which define the behavior mode of vehicles for each scenario and uses some characteristic variables as the basis for judgment in condition switching [5]. However, the traditional method is difficult to apply in complex scenes, which are essential for autonomous driving, and sometimes it is still necessary for human drivers to take over vehicles. For example, during the commercial operation of Waymo autonomous vehicle in Phoenix, it was pointed out that when there are other traffic participants, autonomous vehicles using ACC are often unable to complete a simple right turn independently [6]. In general, complex environments will impose even greater challenges on traditional ACC. On the one hand, it is difficult to carry out the test verification for some complex scenarios in the real world; on the other, the design rules for complex working conditions will rise exponentially.

Aiming to address this limitation, researchers have proposed a set of techniques that are mainly classified into two types, CACC (Cooperative Adaptive Cruise Control) and DRL (Deep Reinforcement Learning)-based ACC. CACC is an emerging technology in current transportation systems based on V2V (vehicle-to-vehicle) communication [7]. The general idea is to connect multiple vehicles following in line with wireless communication to form a column which can expand the perceptibility of the vehicle and aid vehicle control actively through sharing information [8]. The difficulty of the whole process lies in the accurate description of

The associate editor coordinating the review of this manuscript and approving it for publication was Chao Yang.

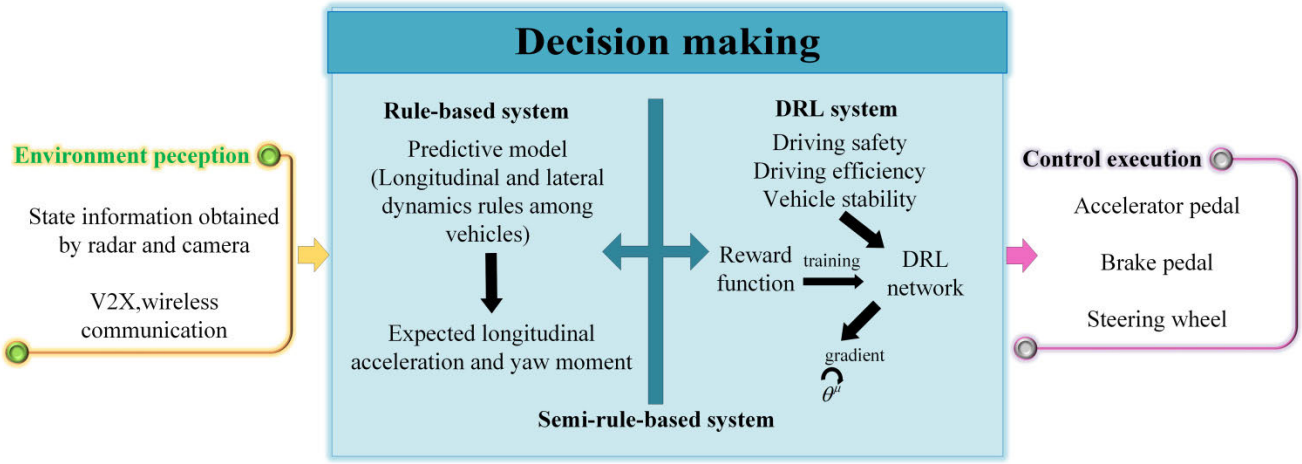

FIGURE 1. A comparison of two decision-making mechanisms.

the formation and disengagement of CACC vehicle strings when CACC vehicles are mixed with human-driven vehicles in the traffic stream [9]. In addition, V2V communications are sometimes unreliable due to failures resulting from communication-related constraints, such as interference and information congestion [10], [11]. This may lead to sudden degradation of control of such vehicles from CACC to ACC mode, which may negatively influence traffic safety [12]. Although information exchange among vehicles makes the design of the concerned rule-based algorithm less difficult, it does not fundamentally solve the complexity of the rule-based design, and it cannot interact with the environment continuously through data-driven self-learning. As a result, CACC is not applicable in some information deficient scenarios [13]–[15].

Deep reinforcement learning enables agents to make autonomous decisions in complex environments [16]. It uses deep learning and reinforcement learning to deal with high-dimensional state space, and discrete or continuous action spaces in decision-making problems [17], [13]. Agents complete reasoning, judgment, and decision-making through a Markov decision-making process and learn how to achieve decision-making control in complex scenarios after continuous interaction with the environment [18]–[22]. Accordingly, the application of DRL in intelligent vehicles has been the focus of much research. Ng *et al.* proposed a longitudinal adaptive control system that used Monte Carlo reinforcement learning and analyzed the performance of the adaptive control system for a single automobile or in a multi-vehicle platoon [23]. Zhu *et al.* proposed a framework for human-like autonomous car-following planning based on deep reinforcement learning, which had an excellent capability of generalization of various driving situations and which can adapt to different drivers by continuously learning [24]. Min *et al.* proposed a deep reinforcement learning method for high-speed driving conditions. The network outputs are advanced actions, such as desired speed, whether to change direction, and so on. Specific implementation includes a driver assistance system forming a closed-loop [25]. Gao *et al.* proposed an independent decision-making method based on reinforcement Q-learning and established the Markov decision process model by analyzing car-following [26]. Hu *et al.* proposed an SRL (Supervised Reinforcement Learning) algorithm for ACC to comply with the human driving habit and applied the SRL algorithm to the ACC problem in different scenarios [27].

The above research has achieved progress in addressing some conditions of vehicle cruise driving. However, the following four problem aspects remain and need to be addressed: *i*) The training speed of the agent is slow. *ii*) The decision-making process has little correlation with rules, and the algorithm lacks interpretability. *iii*) Most researchers still use images as the sole input for end-to-end active planning, which leads to inadequate representation of the driving state that has to be obtained by the DRL network [20], [28], [29]. *iv*) The training environment is mostly used in open-source game engines such as TORCS and Carla to verify the feasibility of the principle; this makes it difficult for the trained DRL control strategy to be directly applied from the virtual training scene to the real scene [30]–[32]. In addition, most of the decision-making algorithms are designed for passenger vehicles, lacking a systematic application for heavy commercial vehicles. Developing a feasible decision-making strategy for heavy vehicles with the capability to deal with the mentioned problems is still an open question.

This paper takes the three-axle heavy vehicle as the research objective and designs the ACC algorithm for heavy vehicles through a Deep Deterministic Policy Gradient (DDPG) network. The simulation environment for training and verification is established in the PreScan physics-based simulation platform. The surrounding environment information is obtained through various sensors. DDPG network and related algorithms are realized in MATLAB programming environment.

The main contributions of this paper are as follows: *i*) By reducing the dimension of action space and state space, the training process of the agent is accelerated, and the training efficiency and the convergence of the training process are improved. *ii*) Active control is used to increase the roll stability of heavy vehicles in curves so as to reduce the occurrence of dangerous conditions, which is an essential factor in the design of heavy vehicle control systems [30], [33]. *iii*) The reward function is designed based on certain rules to make the algorithm interpretable, and the DDPG algorithm is modified in a simple manner to speed up the training process. *iv*) The training environment selected in this paper is closer to reality than that the virtual games used for this purpose in other research on the topic. At the same time, the multi-source sensor information is fused to avoid the limitations of a single information source.

The remainder of this article is organized as follows: Section II deals with the reduction of the action space and state space, and the preliminary construction of a DDPG network structure. Section III establishes an accurate roll stability model for three-axle vehicles and calculates the load transfer rate (LTR) value. Section IV elaborates on the design of the reward function for heavy vehicles based on the control objective in ACC conditions. Section V establishes the update function for the DDPG network. Section VI analyzes and sorts out the experimental results of the control objectives. By changing the environment test, the superiority of the DDPG-based decision-making strategy of the heavy vehicle is verified. The characteristics of several ACC algorithms mentioned above are given in detail in the Appendix.

II. CONSTRUCTION OF VEHICLE-SCENE SIMULATION TRAINING ENVIRONMENT

A. ACTION SPACE DESIGN

The action design of the system should first consider the set of executable actions to be carried out by the agent. An executable action set for intelligently-controlled vehicle driving includes the underlying controls instead of the driver's action, that is, accelerator pedal degree, brake pedal degree, and steering wheel angle. Research on driver behavior shows that the driver mainly focuses on information on two issues in the actual driving process: driving safety and driving efficiency. There are contradictions between these two objectives, and drivers have different styles, as well as different trade-off criteria [34]. In order to screen out the redundant information in the action set, accelerate the convergence of the training process and simplify the model, this paper selects various drivers with different styles to carry out multiple pre-training to achieve the driver action collection. It determines the range of the final action set under the ACC condition.

For distinguishing the individual differences of drivers in the human-vehicle-road closed-loop system, Elander *et al.* took the lead in summarizing the concept of driving style and focused on expressing the driver's decision-making ability and the ability to manipulate the vehicle [35]. However, because the definition of driving style covers a lot, it is difficult to judge its type and difference through a single sentence. So in most studies, the measurement of driving style is in the form of self-report. For making the sampled action data set fully reflect the individual differences and representativeness of various drivers, this paper uses the multi-dimensional driving style inventory (MDSI) to conduct a questionnaire survey on a certain number of drivers to randomly select 5 drivers for each of the three typical driving style types, that is, aggressive drivers, conventional drivers and conservative drivers [36]. MDSI and the driving style measurement method are given in detail in the Appendix. The three-axle commercial vehicle bench used for driving information collection is shown in Fig. 2.

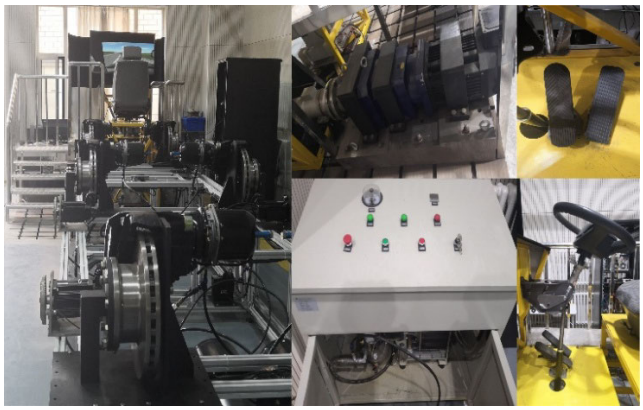

FIGURE 2. Three-axle commercial vehicle experimental bench.

For drivers with different driving styles, the desired vehicle acceleration and yaw rate are different, so their range of action space is different. Three kinds of drivers drive this type of vehicle cruising in a variety of common roads; the action signals for these kinds of driving are collected and sorted through the experiment bench of three-axle heavy vehicles, as shown in Fig. 3. According to the change range of a driver's action information, the accelerator pedal degree range is 0–80%, the brake pedal degree range is 0–60%, and the steering wheel angle limit is $\pm 120°$.

The control of vehicles needs a three-dimensional action space. Still, heavy commercial vehicles do not drive "heel-to-toe" as happens in actual driving processes, so acceleration and braking can be regarded as the same action, the output action space range is set to $(-1, 1)$, and the sign is used to distinguish acceleration and braking. In training processes, the training efficiency of multi-dimensional action space is low, and it is easy to fall into the optimal local solution. In order to further improve the training efficiency, the two-dimensional action space is divided into two one-dimensional action spaces for independent training, that is, two agents that can output a single action are established. After the first agent is trained, the trained data is imported into the same second training environment to continue training the second agent, in order to ensure a higher probability of finding the optimal solution.

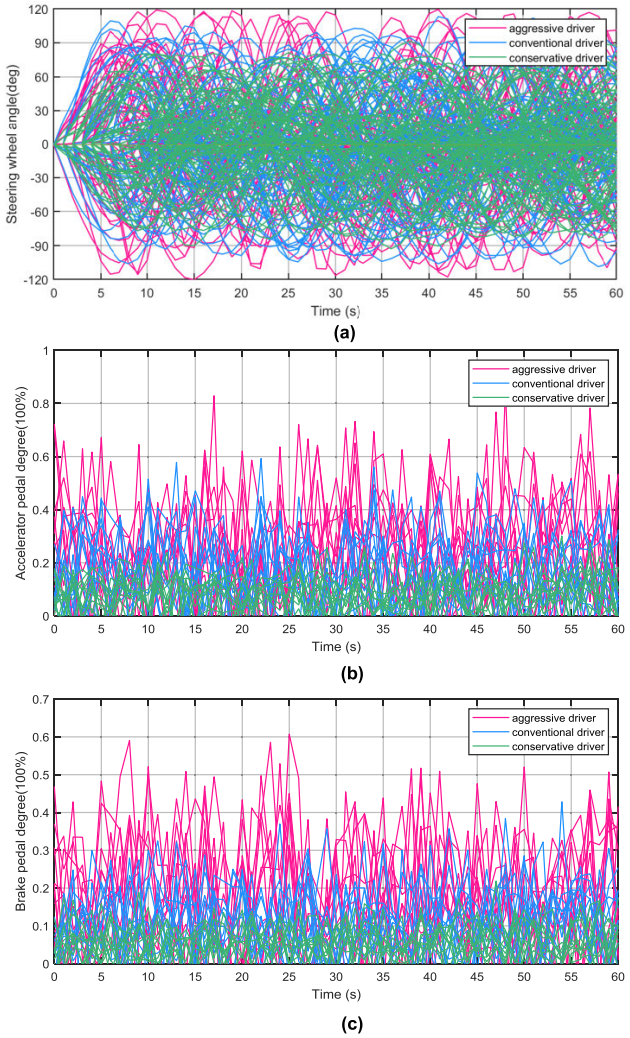

FIGURE 3. Vehicle bench experimental data: (a) Steering wheel angle. (b) Accelerator pedal degree. (c) Brake pedal degree.

The feasible formula for split training of longitudinal control and lateral control is:

$$\int_{t_1}^{t_2} \frac{C}{2\pi} v dt = \int_{t_1}^{t_2} \frac{v \tan(\delta K_{rs})}{2\pi l} dt, \tag{1}$$

where C is the curvature of the road, δ is steering wheel angle, K_{rs} is steering transmission ratio, l is steering wheelbase.

In the above formula, for the cruise vehicle with constant speed, the actual turning radius of the vehicle changes slightly at different speeds, and K_{rs} is almost unchanged. Therefore, the integral of the steering angle of the vehicle and the integral of the road direction change simultaneously with the vehicle speed. If the step size is sufficiently small, the wheel angle can be completely consistent with the change of the road curvature at every moment. In this condition, it can be seen that acceleration and braking will not affect the control effect of steering action.

B. STATE SPACE DESIGN

The state space should include the state information of the vehicle itself, and also the surrounding environmental

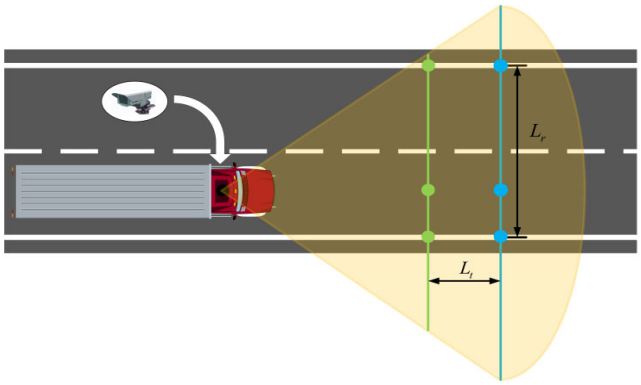

FIGURE 4. Principle of lane line recognition by camera.

information, such as the speed of the vehicle, the speed and relative distance of other vehicles. However, when complex tasks are completed, the size of the state space increases exponentially. At the same time, the relationships between the various dimensions of the state space cause a large part of the state not to appear at all, or a strong correlation between certain states to exist, that is $r_{(st,a)\sim Q} = r_{(st+1,a)\sim Q}$. So, the state characteristics are not obvious. This paper combines specific control requirements to achieve a reduction of the state space. Due to the diversity and complexity of different types of sensor information, reasonable and effective integration is required as the state input for the intelligent vehicle. Firstly, the observation data of a plurality of different types of sensors are collected, from which the sensor output data is extracted. Then the eigenvalues of the data are correlated to complete the common description of the same target. This paper analyzes this process from the two aspects of horizontal (or lateral) control and vertical control.

1) LATERAL CONTROL

The control variable of lane-keeping should bring the vehicle's driving position as close to the center of the lane as possible, and the driving direction should be consistent with the change direction of lane curvature. Therefore, the state space of steering control is divided into two parts: the lateral offset displacement and the lateral offset angle, which are expressed by e_1 and e_2, respectively. The common camera-based lane position sensor is used in the road recognition part. e_1 and e_2 are calculated as follows:

$$\begin{cases} e_1 = (\dfrac{r_{dis0}}{r_{dis0} + l_{dis0}} - 0.25) L_r, \\ e_2 = \dfrac{\arcsin(\frac{r_{dis0}}{r_{dis0}+l_{dis0}} - \frac{r_{dis1}}{r_{dis1}+l_{dis1}}) L_r}{L_t}, \end{cases} \tag{2}$$

where r_{dis} and l_{dis} are the distance between the i-th detection point and the left and right lanes, L_r is the lane width, and L_t is the distance between detection lines. Normalize e_1 to make it 1 when the vehicle reaches the left edge of the lane, and -1 when it reaches the right edge. This state reflects the deviation of the lateral offset displacement and angle of the vehicle heading relative to the lane line at the current

time. And in this algorithm, the driving convention for the model is considered to be on the right side of the road. Here, the main idea of the PID algorithm is introduced, considering the past deviation accumulation and the changing trend of the deviation at the next step. By integrating and differentiating e_1 and e_2 respectively, a six-dimensional state space is formed by $e_1, de_1/dt, \int_{t1}^{t2} e_1 dt,\ e_2, de_2/dt, \int_{t1}^{t2} e_2 dt$ to describe the lateral state of the vehicle.

2) LONGITUDINAL CONTROL

The control process of vehicle longitudinal dynamics is to enable intelligent vehicles to recognize the distance between the target vehicle and itself on the straight or curved road. When there is an obstacle in the current lane, reasonable speed and distance are maintained to avoid a collision, and constant speed is kept when the obstacle disappears. The control of longitudinal vehicle force is realized by increasing the throttle degree and applying braking force. However, in the processes of training, the vehicle is in a non-obstacle avoidance state when it is far away from the obstacle vehicle. Because the vehicle appears randomly during the driving process, the state space will contain a large number of this kind of state, which makes the agent inefficient in training. In order to reduce the space range of non-obstacle avoidance requirements and improve the training speed of the DRL network, it is necessary to remove the redundant part of the state space according to the car-following characteristics.

Considering the distance between two vehicles and the driver's subjective perception of the safety distance in different traffic environments, it is, therefore, necessary to calculate the dynamic safety distance with the help of the safety distance model [37].

$$\begin{cases} D_s = \dfrac{v_r^2}{2D_d} + d_{fl}, \\ d_{fl}(v_l) = 0.8509 v_l + 1.6109, \\ D_{error} = S_r - D_s, \end{cases} \tag{3}$$

where d_{fl} is the distance between the ego vehicle (autonomous vehicle) and the target vehicle after the relative speed is eliminated, which is obtained by the linear regression of the least square method. v_r is the relative speed of the two vehicles, D_d is the expected relative deceleration under the car-following conditions, and S_r is the straight or curved distance along the road between two vehicles, which is calculated from the sensor data.

The complete state variable in the environment should be the three-dimensional state space composed of the vehicle speed v_e, the speed of the target vehicle v_l, and the relative distance S_r between the two vehicles. The other two dimensions are reduced with v_e as a parameter, and S_r is replaced by D_{error}.

The speed of the target vehicle v_l should be reduced according to the actual driving conditions, that is, when the dynamic safe distance between the two vehicles is greater than or equal to 0; the vehicle speed should be compared with the desired speed, and when the dynamic safe distance is less than 0, the vehicle speed should be compared with the desired speed and the target vehicle speed up accordingly.

FIGURE 5. Lane shape and scene.

$$s_{rv} \begin{cases} v_e - v_{set}, & D_{error} \geq 0, \\ v_e - \min(v_{set}, v_l), & D_{error} < 0, \end{cases} \tag{4}$$

where v_e, v_l, and v_{set} are the ego vehicle speed, target vehicle speed, and desired vehicle speed, respectively. Therefore, S_{rv}, v_e, and D_{error} are selected to form a three-dimensional state space.

C. LANE SELECTION FOR TRAINING

The lane selected in this paper includes straight roads with appropriate length and some curved roads with moderate curvature, which are more conducive to the verification of the DRL algorithm. The shape of the lane and the screenshot of the lane environment are shown in Fig. 5.

D. CONSTRUCTION OF DEEP REINFORCEMENT LEARNING NEURAL NETWORK

One of the core tasks in the DDPG approach based on deep reinforcement learning is the structural design of the actor-critic network. The actor network is mainly responsible for receiving the data of the current driving state and combining them and then returning the combined characteristics to output continuous actions. The critic network is responsible for obtaining the sensors input and the actions output by the actor network in the current state, and outputting the value of the current state-action pair [38]. Previous practice has proved that if only a single neural network algorithm is used, the function approximation is not stable, because of the Markovian property of data [39], so two neural networks, the evaluate net and the target net, are constructed based on the parameterization of the commonly used parameter θ of the neural network.

$$\begin{cases} actor\ eval\ net : \mu(s|\theta^{\mu}), \\ actor\ target\ net : \mu'(s|\theta^{\mu'}). \end{cases} \tag{5}$$

The construction of actor network and critic network is based on MATLAB code. They are all regressed through four

TABLE 1. Actor network structure.

Network structure of actor		
Name	State path	None
InputLayer	3(6)×1×1	
FullyConnectedLayer	fc12	
ReluLayer	relu12	
FullyConnectedLayer	fc22	
ReluLayer	relu22	
FullyConnectedLayer	fc32	
ReluLayer	relu32	
FullyConnectedLayer	fc42	
TanhLayer	tanh2(throttle+brake/steering)	
ScalingLayer	scale=2.5, bias=-0.5	

TABLE 2. Critic network structure.

Network structure of critic		
Name	State path	Action path
InputLayer	3(6)×1×1	1×1×1
FullyConnectedLayer	fc11	fc51
ReluLayer	relu12	-
FullyConnectedLayer	fc21	-
AddtionLayer	add2	
ReluLayer	relu22	
FullyConnectedLayer	fc31	
ReluLayer	relu32	
FullyConnectedLayer	fc41 (Q value)	

full connection layers (each layer has 48 neurons). The actor network outputs the steering wheel angle and the accelerator/brake pedal degree using a nonlinear activation function (tanh function). The actor network and critic network structure are respectively arranged, as shown in Table. 1 and Table. 2.

III. STABILITY ANALYSIS OF THREE-AXLE HEAVY VEHICLE

Because of the high position of the heavy vehicle's centroid and the narrow wheelbase relative to the vehicle body, it is more prone to have stability problems, such as rollover, than other vehicles [40]. For multi-axle vehicles with large mass and long body, once an accident occurs, it is a severe traffic accident. Therefore, in order to apply the reinforcement learning algorithms to the driving decision-making of heavy vehicles, the stability of the vehicle must be considered at all times in the training processes, so that the agent can correct the vehicle state in time in case of a rollover trend.

A. VERTICAL LOAD MODELING

In this paper, the research focus of the multi-axle heavy vehicles model is firstly on the modeling of the vertical load distribution. Vertical load distribution is the biggest difference between multi-axle vehicles and two-axle vehicles. In order to avoid over-constraint and introduce dynamic load distribution ratios and other parameters, this paper analyzes the three-axle heavy vehicle in sections, introduces virtual internal force at the disconnection point, and considers the different positions of the centroid for load distribution. Ignoring the vehicle's pitch motion and the flexibility of the body, the vertical load of each axle of the vehicle is caused only by the roll angle velocity, roll angle, lateral acceleration, and longitudinal acceleration. In this paper, the selected truck model in the PreScan simulation environment is the Mercedes Benz Actros 2541. To ensure the accuracy and applicability of the load model, the parameters used in the model are real values and consistent with the data in the training environment (see Table. 3, obtained from Mercedes-Benz official website).

TABLE 3. Vehicle size parameters.

Parameter name	Parameter value
Vehicle curb quality	13 080 kg
Full load quality	26 080 kg
Driving form	6×2
Vehicle length	9750 mm
Vehicle height	3550 mm
Wheelbase	4800+1350 mm
Cargo size	5110×2036×2130 mm

Fig. 6 details the two parts after the separation of the three-axle vehicle. Point A is the breaking point, Fig. 7(a) is the first axle part after separation, and Fig. 7(b) is the second part composed of the remaining two axles and the vehicle body.

The vertical load model divides the vertical load into two parts. One is the static vertical load of the wheel on one side $F_{zrsi,zlsi}$, the other is the vertical load change of the wheel on one side $\Delta F_{zri,zli}$ caused by the longitudinal acceleration, lateral acceleration and roll motion (where i is the number of the car body subsystem after the division, $i = 1, 2$). Therefore, the vertical load equation of each axle of three-axle vehicles is the sum of the static vertical load and the change value of the vertical load, as shown in "6,", where $i = 1, 2, 3$. When the vehicle rolls over, one side of the wheel bears the full mass, and the other side bears no load.

$$F_{zri,zli} = F_{zrsi,zlsi} \pm \Delta F_{zri,zli}. \tag{6}$$

The symbols used in Fig. 6–Fig. 7 are explained in the Appendix. Equation (7) is the static vertical load of each wheel of the vehicle, where K_{st} is the static axle load distribution coefficient between the second and third axles of the vehicle.

$$\begin{cases} F_{zrs1,zls1} = \dfrac{1}{2}\left[mg - \dfrac{mgl_v}{l_1 + \frac{(l_2-l_1)}{2}}\right], \\ F_{zrs2,zls2} = \dfrac{K_{st}}{2}\left[\dfrac{mgl_v}{l_1 + \frac{(l_2-l_1)}{2}}\right], \\ F_{zrs3,zls3} = \dfrac{(1-K_{st})}{2}\left[\dfrac{mgl_v}{l_1 + \frac{(l_2-l_1)}{2}}\right]. \end{cases} \tag{7}$$

Multi-axle vehicles are mostly used to carry heavy goods, and the different positions of the goods will affect the position

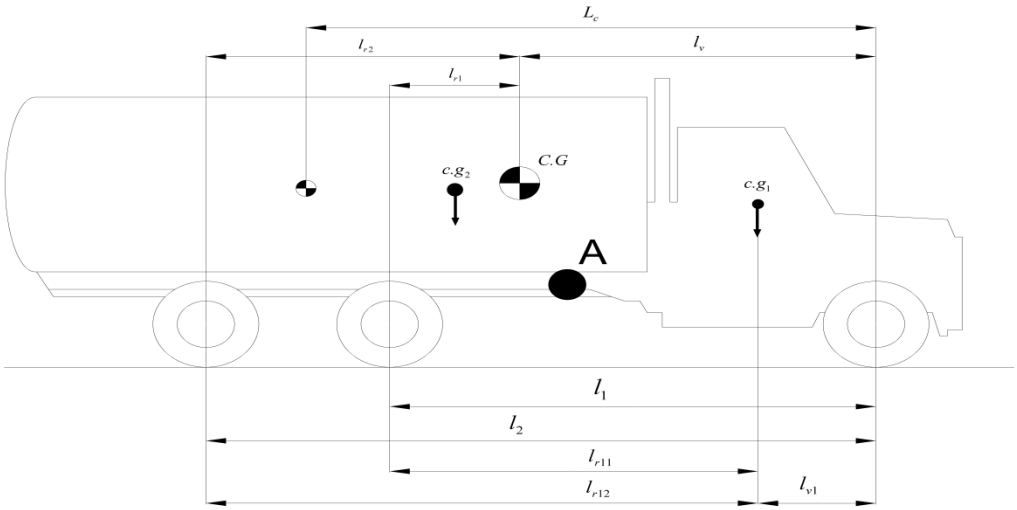

FIGURE 6. Segmentation of three-axle vehicles.

of the vehicle's centroid. Therefore, in order to discuss the extreme operating conditions, we assume that the vehicle is fully loaded, and the goods loaded by the vehicle are homogeneous goods, and the length of the goods is known, and the load distribution of each part of the vehicle body can be estimated. Equation (8) is the estimated value of the length and mass of each part of the vehicle body.

$$\begin{cases} L_1 = (l_{v1} + \dfrac{l_{r11}}{2}) - (L_c - \dfrac{L_{lc}}{2}), \\ m_1 = L_1 \dfrac{m_c}{L_{lc}}, \\ L_2 = L_{lc} - L_1, \\ m_2 = m_c - m_1, \end{cases} \tag{8}$$

where L_1 and L_2 are the length of the goods in the first and second parts of the vehicle, m_1 and m_2 are the masses of the goods in the first and second parts of the vehicle, and L_{lc} is the total length of the goods. According to the empirical equation, the distance l_{v2} between the equivalent centroid of the second part and the second axle can be estimated. This value is only used to verify the validity of the model, and correction will be made for compensation later.

$$l_{v2} = (L_2 + \frac{l_2}{2}) - (\frac{L_c}{2} + L_{lc}). \tag{9}$$

After the three-axle vehicle is divided, the vertical load of each part can be calculated through the balance of force and moment. The vertical load change of the first axle $\Delta F_{zr1,zl1}$ can be calculated from Fig. 7(a). The vertical load change of the first axis lateral moment transformation $\Delta F_{zrm1,zlm1}$, and vertical load change of the pitch moment transformation $\Delta F_{zra1,zla1}$, which is the virtual internal force applied to the split part, are shown in (10).

$$\begin{cases} \Delta F_{zr1,zl1} = \Delta F_{zrm1,zlm1} - F_{zra1,zla1} + \dfrac{K_{b1}\varphi}{H}, \\ \Delta F_{zrm1,zlm1} = \dfrac{(m_v a_y h_1 + m_1 a_y h_c)}{H} \cos\varphi - \dfrac{K_1\varphi + C_1\varphi'}{H} \\ + \dfrac{(m_v g h_{r1} + m_1 g h_c)}{H} \sin\varphi, \\ F_{zra1,zla1} = \dfrac{\Delta F_{zrm1,zlm1} l_{v1} + \frac{m_v a_x h_1}{2} + \frac{m_1 a_x h_c}{2}}{l_{s1}}, \\ l_{s1} = l_{v1} + \dfrac{l_{r11}}{2}. \end{cases} \tag{10}$$

According to Fig. 7(b), the moment balance equation of the second part is (11), where the vertical load change caused by the moment in different directions for each axle ($\Delta F_{zr2,zl2}$ and $\Delta F_{zr3,zl3}$) can be calculated.

$$\begin{cases} \Delta F_{zrm2,zlm2} = \dfrac{m_2 a_y h_2}{H} \cos\varphi - \dfrac{K_2\varphi + C_2\varphi'}{H} \\ + \dfrac{m_2 g h_{r2}}{H} \sin\varphi, \\ \Delta F_{zr2,zl2} = \dfrac{F_{zra1,zla1}(l_2 - l_{v1} - \frac{l_{r11}}{2}) - \frac{m_2 a_x h_2}{2}}{l_{r12} - l_{r11}} \\ + \dfrac{\Delta F_{zrm2,zlm2}(l_{r12} - l_{r11} - l_{c2})}{l_{r12} - l_{r11}} + \dfrac{K_{b2}\varphi}{H}, \\ \Delta F_{zr3,zl3} = -\Delta F_{zr2,zl2} + F_{zra1,zla1} \\ + \Delta F_{zrm2,zlm2} + \dfrac{K_{b3}\varphi}{H}. \end{cases} \tag{11}$$

This paper uses the lateral load transfer rate (LTR) as an evaluation index to identify whether the vehicle tends to roll over, which is defined as:

$$LTR = \left| \frac{F_{zr} - F_{zl}}{F_{zr} + F_{zl}} \right|. \tag{12}$$

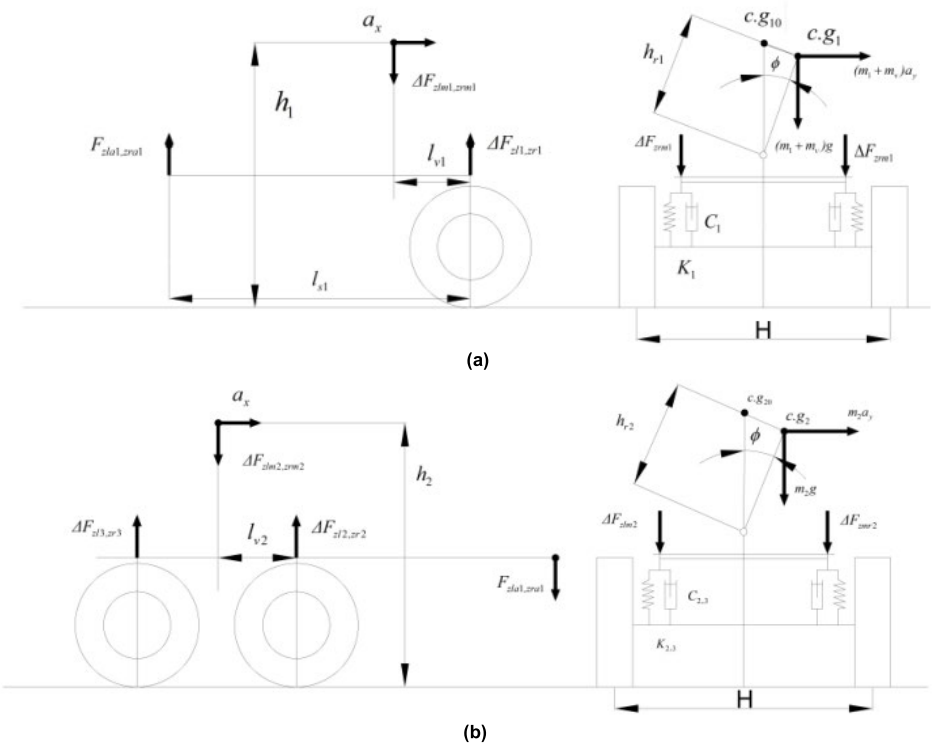

FIGURE 7. Details of each part of three axle vehicles: (a) The first part. (b) The second part.

Combined with the load model established above, the dynamic lateral load transfer can be used to calculate the vertical load of each axle. The value is related to the vehicle's roll angle, roll angle speed, lateral acceleration, and other variables. The simulation results show that the absolute value of the threshold is 0.55, i.e., when the absolute value LTR is greater than 0.55, the algorithm judges that the vehicle will tend to roll over, and the control algorithm is started to control the vehicle.

B. ESTIMATION OF THE CENTROID POSITION

Vehicle states, such as yaw rate, longitudinal acceleration, and lateral acceleration, can be obtained by sensors, but there are some parameters in the system studied in this paper that cannot be measured directly, or that require expensive sensors, and these state parameters are very important for calculating the vertical load of vehicle axle. Therefore, this paper proposes a method to identify the heavy vehicle's centroid position based on the H-infinity filter and the CKF (Cubature Kalman filter). Aiming at the problem of noise uncertainty in nonlinear filters, the H-infinity filter theory is combined with the point-based nonlinear filter, and the five-degree cubature rule is used to approximate the Gauss integral in the point-based nonlinear H-infinity filter frame and obtain the robust H-infinity Cubature Kalman filter. Finally, the effectiveness of the algorithm is verified by simulation using the TruckSim simulation software for multi-axle trucks.

Generally, the lateral disturbance of the vehicle's centroid caused by the external input is small, so the centroid parameters to be identified are the height and distance of the centroid from the front and rear axles. The vertical acceleration of the vehicle is much smaller than the gravity acceleration, so the vertical acceleration of each wheel is not considered here.

The equation for calculating the state of the centroid is as follows, with the details in Appendix [41]–[44].

$$\begin{cases} ma_x = k_s(\sum_{i=1}^{3} s_i F_{zi}) - \frac{1}{2}\rho C_d A v_x^2 - mg\sin\theta, \\ J\dot{w}_i = T_i - r_r(f + k_s s_i)F_{zi} \end{cases} \tag{13}$$

The parameters required in the above equations can be obtained from the real vehicle data or the simulation software TruckSim. Assuming that x=[l_v h], where l_v is the horizontal distance from the centroid to the first axle, h is the height of the centroid from the ground, the expression of the filtering system can be further obtained:

$$\begin{cases} x_{k+1} = x_k + w_k, \\ z_{k+1} = f(x_k, z_k) + v_k. \end{cases} \tag{14}$$

The CKF realizes the extraction of the demand signal according to the Bayesian estimation principle with its recurrence process in the Appendix. However, the robustness of CKF is not good when dealing with the filtering problem [45], so in this paper, CH∞KF filtering is introduced to improve the filtering robustness and to eliminate the degradation

problem in KF. Similarly, it is necessary to satisfy the theorem (a) when H-infinity filtering is applied to a nonlinear field [46].

Theorem (a): the state transition matrix of a linear system is F_k, and the input control matrix is G_k. If $[F_k\ G_k]$ is a full rank matrix, for a given positive number γ_k, the H-infinity suboptimal filter must exist, if and only if all k, the H-infinity suboptimal filter must satisfy:

$$P_{k|k}^{-1} = P_{k|k-1}^{-1} + H_k^T R_k^{-1} H_k - \gamma_k^{-2} L_k^T L_k > 0$$

where H_k is the measurement matrix of the system, L_k is any known matrix that can be linearly combined with the state vector x_k. Usually, $L_k = I$, where I is the unit matrix of the corresponding dimension.

When the theorem (a) holds, if $P_{0|0}$ is a given positive definite matrix, then the error covariance matrix $P_{k|k}$ at time k satisfies the Riccati equation as follows:

$$\begin{cases} P_{k|k} = P_{k|k-1} - P_{k|k-1}[H_k^T L_k^T] R'_{e,k} \begin{bmatrix} H_k \\ L_k \end{bmatrix} P_{k|k-1}^T, \\ R'_{e,k} = \begin{bmatrix} R_k & 0 \\ 0 & -\gamma_k^2 I \end{bmatrix} + R'_{e,k} \begin{bmatrix} H_k \\ L_k \end{bmatrix} P_{k|k-1} \begin{bmatrix} H_k^T & L_k^T \end{bmatrix}. \end{cases} \tag{15}$$

To reduce the amount of computation in each recursive process, transform the variables in (15) to:

$$\begin{cases} P_{k|k-1} H_k^T \approx P_{xz,k|k-1}, \\ H_k P_{k|k-1} H_k^T \approx P_{xz,k|k-1} - R_k, \\ H_k P_{k|k-1}^T = (P_{k|k-1} H_k^T)^T \approx P_{xz,k|k-1}^T. \end{cases} \tag{16}$$

Substituting "16" into "15," the error covariance matrix at time k is updated to:

$$\begin{cases} P_{CH\infty KF,k|k} = P_{k|k-1} - \begin{bmatrix} P_{xz,k|k-1} & P_{k|k-1} \end{bmatrix} \\ (R'_{e,k})^{-1} \begin{bmatrix} P_{xz,k|k-1}^T \\ P_{k|k-1}^T \end{bmatrix}, \\ (R'_{e,k})^{-1} = \begin{bmatrix} P_{zz,k|k-1} & P_{xz,k|k-1}^T \\ P_{xz,k|k-1} & P_{k|k-1} - \gamma_k^2 I \end{bmatrix}. \end{cases} \tag{17}$$

Compared with the traditional CKF, CH∞KF adopts a new method to calculate the error covariance matrix $P_{k|k}$ at time k. In order to ensure the positive definiteness of the error covariance matrix at time k, the theorem (a) is transformed into "18" according to the inverse matrix theorem and "15"-"17."

$$\gamma_k^2 > \zeta \max \left\{ eig(P_{k|k-1}^{-1} + P_{k|k-1}^{-1} P_{xz,k|k-1} \times R_k^{-1} [P_{k|k-1}^{-1} P_{xz,k|k-1}]^{-1})^{-1} \right\}. \tag{18}$$

According to the above derivation, it can be seen that CH∞KF keeps positive covariance to prevent filter divergence by adjusting the change of immunity factors γ_k. By the same method, the prediction error covariance of a dynamic system is transformed.

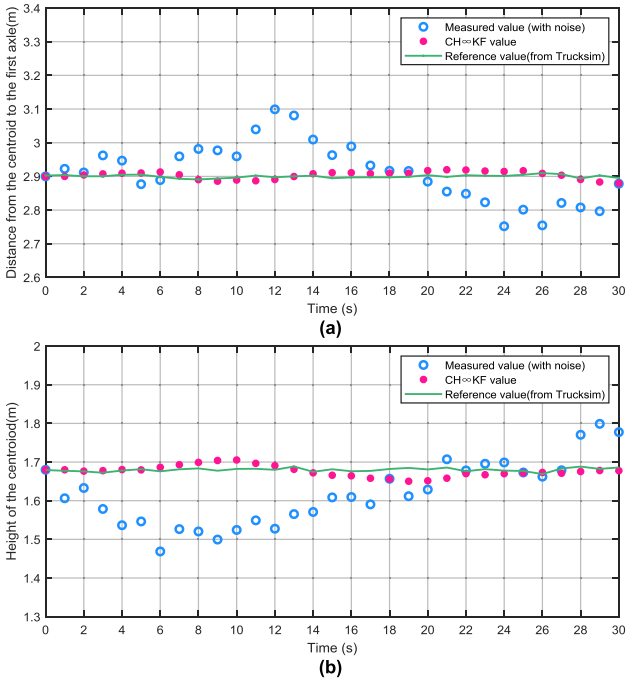

FIGURE 8. Filter graph for Centroid states identification: (a) Distance from the centroid to the first axle. (b) Height of the centroid.

$$P_{CH\infty KF,k|k-1} = \frac{1}{2n} \sum_{i=1}^{2n} X_{i,k|k-1}^* (\sum_{i=1}^{2n} X_{i,k|k-1}^*)^T - \hat{x}_{k|k-1} (\hat{x}_{k|k-1})^T + Q_{k-1}. \tag{19}$$

After obtaining the estimation expression of the centroid position of the system, the state value of the vehicle centroid can be obtained by using the CH∞KF algorithm. Since the centroid position of the vehicle does not change anymore after each start, the value of k_l and h is fixed. The sampling is performed every interval after the estimation starts, and the variance of the nearest n sampling points is calculated, and the square difference is normalized.

$$\begin{cases} \sigma_1 = \sum_{k=1}^{n} (\frac{l_{v,k} - \bar{l}_v}{\bar{l}_{v,n}})^2, \\ \sigma_2 = \sum_{k=1}^{n} (\frac{h_k - \bar{h}}{\bar{h}_n})^2, \end{cases} \tag{20}$$

where $l_{v,k}$ and h_k are the estimated values, $\bar{l}_v$ and $\bar{h}$ are the average of the estimated values of the last n sample points. If the variance σ_2 and σ_3 are less than the set thresholds σ_{20} and σ_{30}, the estimation is stopped. The filter graph for centroid states identification is shown in Fig. 8, in which the identification value of centroid states is closer to the reference value than the measured value, converges, and the error is less than 5%.

The filtered vehicle centroid position parameters are substituted into the vehicle vertical load model in III.A, and the

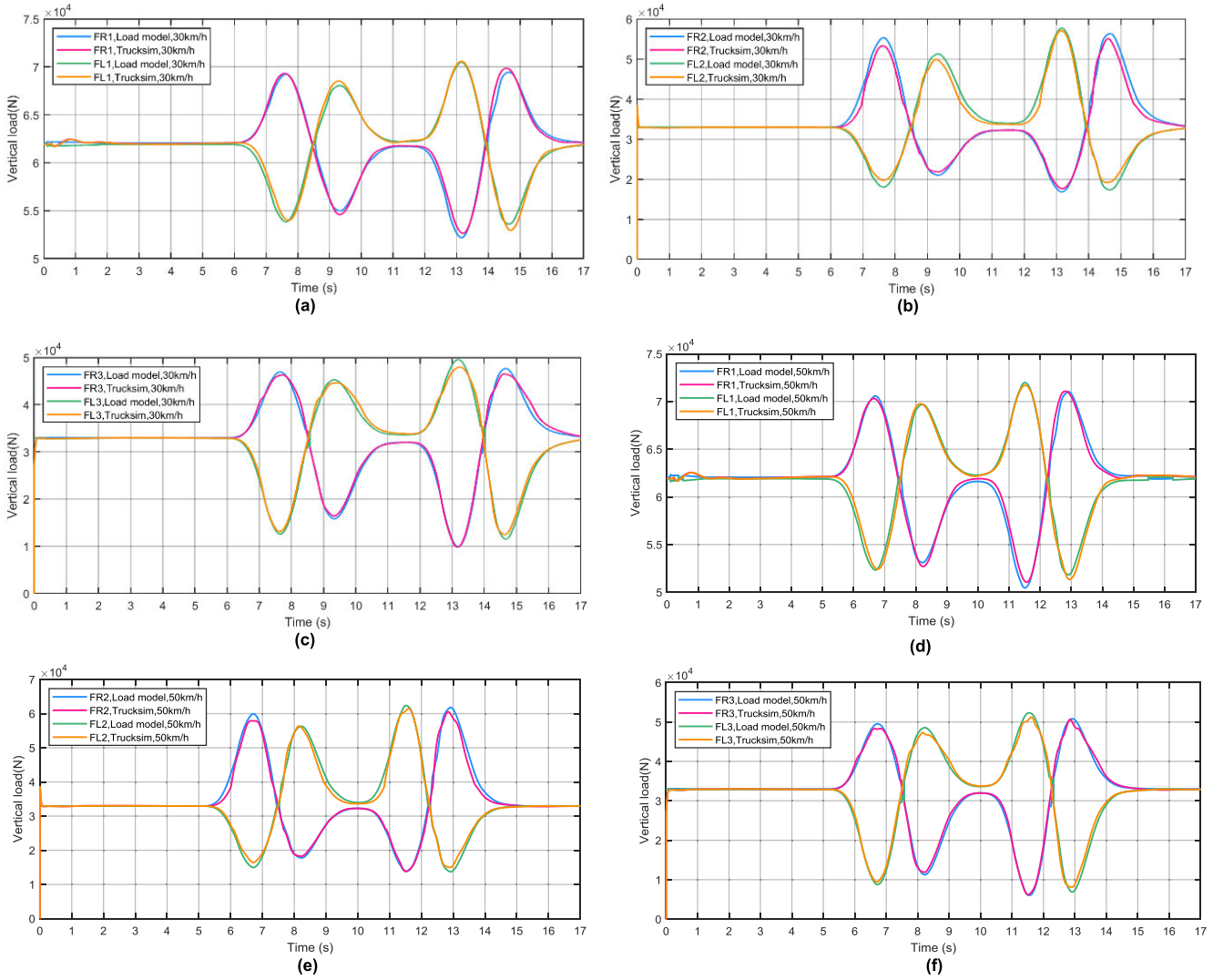

FIGURE 9. Load fitting curves: (a) Right and left wheels load of the first axle at 30 km/h. (b) Right and left wheels load of the second axle at 30 km/h. (c) Right and left wheels load of the third axle at 30 km/h. (d) Right and left wheels load of the first axle at 50 km/h. (e) Right and left wheels load of the second axle at 50 km/h. (f) Right and left wheels load of the third axle at 50 km/h.

load fitting curves of the left and right wheels of each axle are verified under the double-lane change conditions of 30 km/h and 50 km/h respectively, as shown in Fig. 9.

It can be seen from the vertical load comparison figure that the three-axle heavy vehicle vertical load calculation method has a good motion-fitting effect, but the selected working condition is extreme, and the model is not modified, so there is a state error, which is less than 5%. The TruckSim model vibrates at the beginning of the movement. The original data used here are collected from the beginning, so there are fluctuations here too. Operation and data collection will start later after the vehicle is stable. The modeling idea can effectively avoid the influence of over constraint of three-axle vehicles, and it can be seen that the vertical load of the vehicle can still be well described when a few wheels on one side are almost raised.

C. VERTICAL LOAD CORRECTION ALGORITHM

According to the vehicle load model and the state parameter calculated by the algorithm proposed above, we can obtain the vertical load of each axle with a better fitting effect. However, the vertical load varies greatly, so it is impossible to judge whether the calculation is accurate by relative deviation; also, the calculated vertical load of the vehicle is affected by the deviation between the identified parameters and the model accuracy, so it cannot be directly applied to the control system. Therefore, in order to further improve the accuracy of vertical load estimation, a correction method for the vertical load of three-axle commercial vehicles is proposed.

Firstly, the parameters of the first part are known at the no-load condition, and the vertical load of this part can be calculated more accurately, so the first axle does not need to be corrected for a vertical load.

Secondly, the second axle and the third axle are located in the second part of the vertical load model, which are relatively independent. In parameter calculation, the heavy vehicle with a long body has a rear magnification effect.

Thirdly, the vehicle has anti-roll stabilizer bars, and the excessive roll angle or roll angle speed will cause changes in the suspension equivalent stiffness and damping values.

Based on the above three points, the semi-empirical algorithm is designed to adjust these parameters to improve the accuracy of vertical load model in III.A. Because the influence of stiffness and damping coefficient is more complex than that of the anti-roll stabilizer bar, it is more reasonable to adjust the vertical load only through the anti-roll stabilizer bar stiffness.

For the vehicle's lateral load transfer rate, we assume that only the second part of the three-axle commercial vehicle is considered:

$$LTR_{2,3} = \frac{F_{zr2,zr3} - F_{zl2,zl3}}{F_{zr2,zr3} + F_{zl2,zl3}} = k_{ltr2,ltr3}\frac{2a_y h_2}{gH}, \tag{21}$$

where h_2 is the height of the second part of the vehicle's centroid simply calculated based on experience, and $k_{ltr2,ltr3}$ is the coefficient describing the rear magnification.

$$h_2 = h_1 + \frac{J_y(l_2 - l_s)m_2}{k_h J_z l_2 m_1}, \tag{22}$$

where k_h is an empirical parameter ($k_h = 7$). Combining the load model in III.A and (21), we can get:

$$2\frac{K_{b2,b3}\phi}{H} = k_{ltr2,ltr3}\frac{2a_y h_2}{gH}(F_{zr2,zr3} + F_{zl2,zl3}) + (F_{zr2,zr3} - F_{zl2,zl3}). \tag{23}$$

According to (23), two thresholds ($M_{ltr2,ltr3}$ and $N_{ltr2,ltr3}$) are set to modify. The relative sizes of $M_{ltr2,ltr3}$ and $N_{ltr2,ltr3}$ indicate that the force generated by the anti-roll stabilizer bar is too large or too small, so the force generated by the anti-roll stabilizer bar can be adjusted by the threshold. At the same time, a_y and φ are used for the fourth-order polynomial fitting to get the correction law for the vertical load coefficient.

$$\begin{cases} M_{ltr2,ltr3} = 2\dfrac{K_{b2,b3}\phi}{H}, \\ N_{ltr2,ltr3} = k_{ltr2,ltr3}\dfrac{2a_y h_2}{gH}(F_{zr2,zr3} + F_{zl2,zl3}) \\ +(F_{zr2,zr3} - F_{zl2,zl3}), \end{cases} \tag{24}$$

$$sat(M_{ltr2,ltr3} - N_{ltr2,ltr3}) = \begin{cases} 0, \\ \quad if\ \left|M_{ltr2,ltr3} - N_{ltr2,ltr3}\right| < 1.5, \\ sign(M_{ltr2,ltr3} - N_{ltr2,ltr3}), \\ \quad else. \end{cases} \tag{25}$$

The semi-empirical correction equation for vertical load $F_{zri,zli,m}$ is:

$$\begin{cases} F_{zri,zli,\mathrm{m1}} = F_{zri,zli} + sat(M_{ltr2,ltr3} - N_{ltr2,ltr3})\dfrac{K_{b2,b3}\phi}{H}LTR, \\ F_{zri,zli,\mathrm{m2}} = p_{00} + p_{10}\phi + p_{01}a_y + p_{20}\phi^2 \\ +p_{11}\phi a_y + p_{02}a_y^2 \\ +p_{30}\phi^3 + p_{21}\phi^2 a_y + p_{12}\phi a_y^2 + p_{03}a_y^3 + p_{40}\phi^4 \\ +p_{31}\phi^3 a_y + p_{22}\phi^2 a_y^2 + p_{13}\phi a_y^3 + p_{04}a_y^4, \\ F_{zri,zli,\mathrm{m}} = F_{zri,zli,\mathrm{m1}} + F_{zri,zli,\mathrm{m2}}, \end{cases} \tag{26}$$

where the polynomial fitting parameter $p_{pq}(p = 0-4, q = 0-4)$ is a fixed value obtained by semi-empirical correction, and the corrected vertical load of each axle is shown in Fig. 10.

Through comparison and verification, it can be seen that the modified load model proposed is highly consistent with the complex vertical load model in TruckSim, which can accurately estimate the vehicle state and meet the requirements of the application in the control system. The vertical load value obtained by the model will be used to calculate the LTR value of the three-axle heavy vehicle in the PreScan environment, and the stability judgment basis will be introduced into the design of the reward function to achieve stable control of the vehicle.

IV. DESIGN REWARD FUNCTION

In the DRL framework, agents can only learn how to interact with the environment according to the definition of the reward function, so the design of the reward function directly determines the control effect of agents. The reward function needs to define the rewards and punishments of corresponding actions under different driving conditions, but few people consider the stability of the vehicle from the perspective of vehicle system dynamics. Based on the load transfer model, the stability analysis has been carried out, and the reward factors based on driving efficiency, driving safety, and driving stability have been comprehensively considered.

A. DISTANCE DEVIATION PUNISHMENT TERM

$$R_e = -(k_{e1}e_1^2 + k_{\dot{e}1}\dot{e}_1^2 + k_{e2}\left|e_2\right| + k_{\dot{e}2}\left|\dot{e}_2\right|), \tag{27}$$

where $k_{e1}, k_{\dot{e}1}, k_{e2}$ and $k_{\dot{e}2}$ are the punishment coefficients corresponding to the driving deviation. This behavior of directly rewarding or punishing the state makes it easier for the agent to learn the desired action. Even if the best strategy is not found, this term can provide positive feedback. Because the rewarding process is Markovian, the design of this part of the reward function only includes the punishment for the current state and future trends, and the relative weight at the current moment should be higher than that for the next moment.

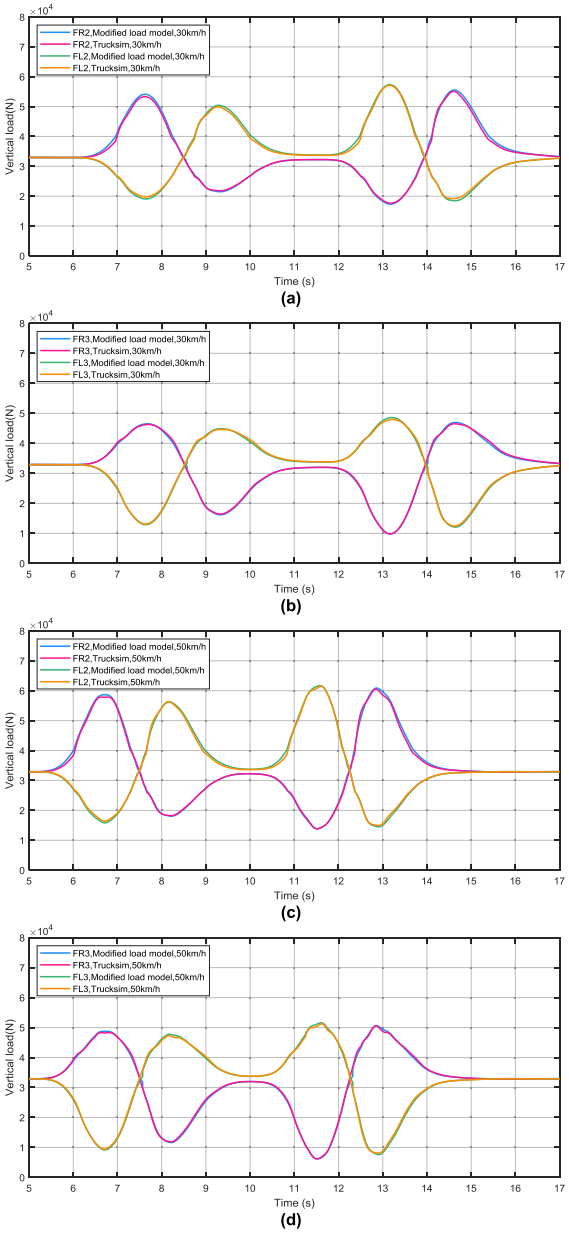

FIGURE 10. Modified load fitting curves: (a) Right and left wheels load of the second axle at 30 km/h. (b) Right and left wheels load of the third axle at 30 km/h. (c). Right and left wheels load of the second axle at 50 km/h. (d) Right and left wheels load of the third axle at 50 km/h.

B. SPEED REWARD AND OVERSPEED PUNISHMENT

$$R_v \begin{cases} \cos\varphi v_{kl} - |\sin\varphi| v_{kl}, \\ \quad v_e \le v_{set},\ Derror \ge 0, \\ \cos\varphi v_{kl} - |\sin\varphi| v_{kl} - |e_1| v_{kl}, \\ \quad v_e \le v_{set},\ Derror < 0, \\ \cos\varphi (2v_{kl} - v_{vkl}^2) - |\sin(e_2)|(2v_{kl} - v_{vkl}^2), \\ \quad v_e > v_{set},\ Derror \ge 0, \\ \cos\varphi (2v_{kl} - v_{vkl}^2) - |\sin(e_2)|(2v_{kl} - v_{vkl}^2) - |e_1| v_{kl}, \\ \quad v_e > v_{set},\ Derror \ge 0, \end{cases} \tag{28}$$

where v_e and v_{set} are the speed of the vehicle and the desired speed, and v_{kl} is the relative value of v_e and v_{set}. When the vehicle speed is less than the desired speed, as the vehicle speed increases, the reward continues to increase. If the vehicle reaches the desired speed, and the direction of the vehicle is consistent with the centerline of the lane, and the vehicle is at a safe distance from the target vehicle, the reward reaches a maximum of 1. When the vehicle speed is higher than the reference speed, the reward decreases quadratically with the increase in the vehicle speed, which can prevent the agent from repeatedly accelerating and decelerating due to greed and causing the vehicle speed to fluctuate. If the initial training state is always with a negative reward, then the learning efficiency will be very slow. If the initial training state is always with a positive reward, the system finds it difficult to learn what is the good side. Therefore, this term adopts the middle zero-point reward, which includes both positive and negative.

C. LARGE STEERING WHEEL ANGLE PUNISHMENT TERM

$$R_\delta = -k_\delta \times \max((|\delta| - 0.2), 0), \tag{29}$$

where k_δ is the punishment coefficient of steering wheel angle. In the design process of the reward function, in order to make the behavior of agents more stable, the normalized steering wheel angle is limited.

D. ROLL STABILITY PUNISHMENT TERM

$$R_r = \frac{2k_r}{\pi} \arctan\left(\frac{1}{LTR - (1+\varepsilon)}\right), \quad 0.55 < LTR \le 1, \tag{30}$$

where k_r is the punishment coefficient for roll stability. ε is an infinitesimal positive number. When LTR is greater than 0.55, it is considered that the trend of roll begins to appear. At this time, punishment is given to enable the vehicle to avoid risks in a controllable state in time.

E. DYNAMIC SAFETY DISTANCE ERROR PUNISHMENT TERM

$$R_s = \begin{cases} k_s \times \dfrac{D_{error}}{D_{safe}}, & if\ D_{error} < 0, \\ 0, & else, \end{cases} \tag{31}$$

where k_s is the punishment coefficient of dynamic safety distance error. When the difference between the dynamic safety distance and the relative distance between two vehicles is less than 0, the punishment is set for the distance error.

F. PUNISHMENT TERM

$$R_d = -k_d \times done\_signal, \tag{32}$$

where k_d is the done punishment coefficient. If the vehicle leaves the lane too much, or collides with other vehicles, or the speed is lower than 5 km/h within 100 consecutive steps, or the value of LTR is 1, or the vehicle reaches the

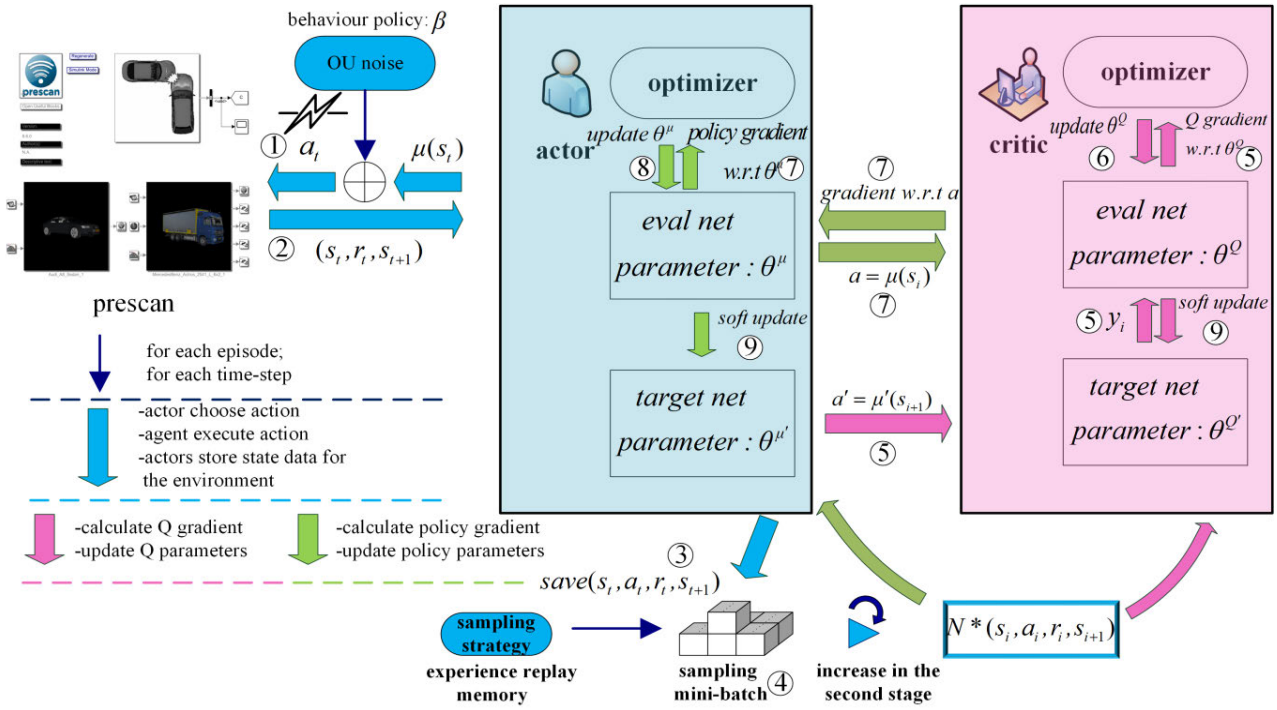

FIGURE 11. Simulation framework and algorithm implementation.

endpoint, then the training process will be terminated. The punishment will be given, and the next training will be carried out.

If the vehicle is always stuck and unable to regain the free exploration state, the experience pool will increase with too much of the data being invalid, and the single training episode will be meaninglessly extended, which greatly reduces the convergence speed of the agent network training. Therefore, in the process of simulation, appropriate intervention and termination conditions are added. However, it is very difficult to learn this signal when the vehicle training is terminated, and it is easy for the signal to disappear in a large number of data. Therefore, a larger punishment coefficient is given to quickly reduce the probability of *Q(s,a)* reappearance. By reasonably designing the structure and coefficient of other rewards and punishments, the reward function is as continuous and effective as possible to solve the sparse reward and optimize the training results and efficiency.

Summarizing the reward and punishment terms of all reward functions, the final reward function of the agent is:

$$R = R_e + R_v + R_\delta + R_r + R_s + R_d. \tag{33}$$

V. UPDATE FUNCTION

The strategy update method of the actor adopts the strategy gradient to optimize. The optimization goal is the total expected reward of the strategy $max_\theta E(R|\pi_\theta)$, R is the cumulative reward in the process, and π_θ is the behavior strategy. The objective function of reinforcement learning can be expressed as:

$$J(\theta) = E(\sum_{i=0}^{N} R(s_i, a_i); \pi_\theta) = \sum_{\tau} P(\tau; \theta) R(\tau), \tag{34}$$

where $R(\tau)$ is the return of the trajectory, and $P(\tau; \theta)$ is the probability of the trajectory appearing. For a set of state-action sequences $\tau = (s_0, a_0, s_1, a_1, \ldots, s_i, a_i)$ of the agent, in order to make the strategy produces a fixed trajectory, that is, the action output is unique under the same state, so a deterministic strategy is adopted [28]. At the same time, in order to avoid the inability to learn, due to the inability of certain strategies to access other states, the learning method of different strategies is adopted. In this paper, the calculation method of the heterogeneous deterministic strategy gradient is as follows:

$$\nabla_{\theta^\mu} J_\beta(\mu_\theta) \approx E_{s\sim p^\beta}[\nabla_{\theta^\mu} \mu(s|\theta^\mu)|_{s=s_i} \times \nabla_a Q(s, a|\theta^Q)|_{s=s_i, a=\mu(s_i)}], \tag{35}$$

where β is the sampling strategy, ρ is the state distribution, $\mu(s|\theta^\mu)$ is the deterministic strategy, and $Q(s,a|\theta^Q)$ is the action value function. In the conventional DDPG algorithm, the network randomly samples from experience replay memory buffer for offline training. The finite size state-action sequences are stored in the experience replay memory buffer and randomly sampled according to the exploration strategy. However, this leads to a general learning effect of the sample in the early stage and a slower learning rate in the later stage. Therefore, in the second stage of learning, the sample space

is increased, and the samples with better behavior are added in the later stage. The improved simulation framework and algorithm implementation are shown in the following figure.

In the simulation framework, the evaluate net in the actor guides the vehicle to make behavior decisions and controls the vehicle to drive in the unknown environment (step 1). The vehicle state information and visual image are obtained from PreScan, and the feedback data is transmitted to DDPG network for calculation (step 2). Deep neural network training often assumes that the data are independent and identically distributed. Since the RL training data is a sequential time series, memory is built to break the correlation in the data, and the loss function is defined as:

$$L(\theta)=E_{(s_i,a_i,r(s_i,a_i),s_{i+1})\sim U(D)}[y_i-Q(s_i,a_i|\theta^Q)^2], \tag{36}$$

where $U(D)$ is mini-batch, which is used for experience storage and playback. In the early stage, the sample pool has a small sample space, which increases to a fixed value as the number of iterations increases. During the training process, the update process of the deterministic critic algorithm can be expressed as:

$$\begin{cases} \delta_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \\ -Q(s_i, \mu(s_i|\theta^{\mu})|\theta^{Q}), \\ \theta^{Q'} = \theta^{Q} + \alpha_{\theta Q}\delta_i \nabla_{\theta Q} Q(s_i, \mu(s_i|\theta^{\mu})|\theta^{Q}), \\ \theta^{\mu'} = \theta^{\mu} + \alpha_{\theta^{\mu}} \nabla_{\theta^{\mu}} \mu(s_i|\theta^{\mu}) \nabla_{\mu(s_i|\theta^{\mu})} \\ Q(s_i, \mu(s_i|\theta^{\mu})|\theta^{Q}), \end{cases} \tag{37}$$

where δ_i is the time difference error, r_t is the reward at the current time, $Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'})$ is the estimated value of Q at the current time, and $Q(s_i, a_i|\theta^Q)$ is the Q value at the previous time. Equation (37) represents the method of updating the value function parameter θ^{μ} using the value function approximation method and updating the policy gradient parameter θ^Q using the deterministic strategy gradient method, where $\alpha_{\theta Q}$ and $\alpha_{\theta^{\mu}}$ are the learning rates of the value function and the strategy gradient function respectively.

The training parameters used in the above mentioned deep reinforcement learning training and the weights of the reward function are designed, as shown in the following table.

VI. COMPARISON OF EXPERIMENTAL RESULTS

In deep reinforcement learning, episode reward, and average reward are usually used to reflect the training convergence level and learning effect. Fig. 12(a) and Fig. 12(b) are training processes for the steering wheel angle and longitudinal force control in action space, respectively. In the early stage of the training of the first agent, the DRL model did not get obvious rewards and remained at a low level. After more than 10 episodes, the DRL model entered a relatively fast stage of exploring and learning and quickly converged. After 40 episodes, episode rewards were slightly increased and stabilized, and then maintained at a high reward value. In the early stage of training the second agent, the reward value fluctuated significantly. With the increase in the number of episodes, the amplitude of the fluctuations gradually

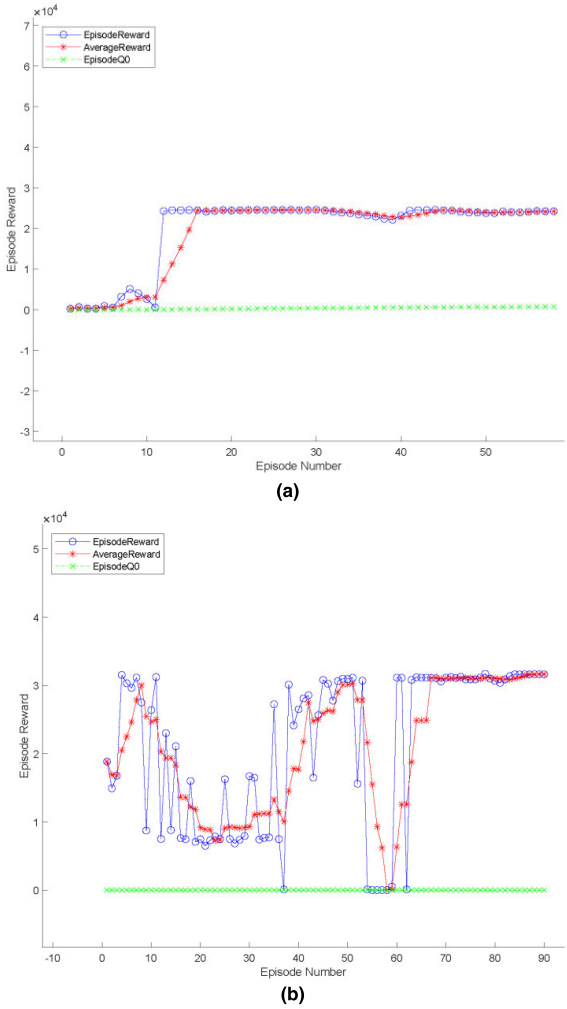

FIGURE 12. Reward value of training process: (a) First agent. (b) Second agent.

decreased, and the highest reward value was explored in about 50 episodes. Random exploration in about 60 episodes led to a decrease in reward value, while the decision-making reward was stable at a high reward value in about 70 episodes.

In the training process, due to the small step size and loose termination conditions, the agent is able to make a full exploration in each episode. The split training of the two agents reduced the dimension of the output action and reasonably reduced the state space, thereby significantly improved the learning efficiency. Both agents converged in 100 episodes and greatly reduced the number of episodes required for training.

In order to verify the training effect of the agents, the speed curves of variable speed, constant low speed, and constant high speed were selected randomly for the target vehicle. Position A in Fig. 13–Fig. 15 shows that the agent controls the vehicle and causes it to cruise at the desired speed. In Fig. 13(a), the vehicle at B position is decelerated and cannot reach the desired speed at D position due to the fact that if the desired speed is maintained to exceed the curved road, the lateral stability of the vehicle will become worse.

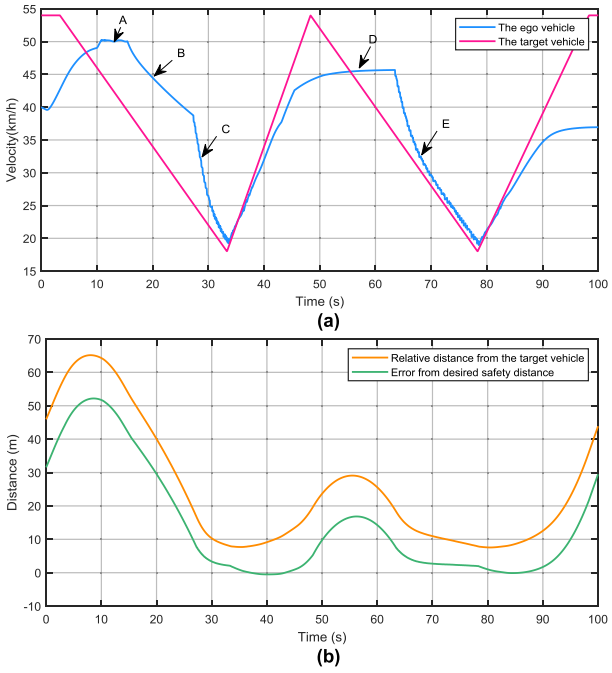

FIGURE 13. The simulation results for the target vehicle driving with variable speed: (a) Vehicle speed change. (b) Distance change.

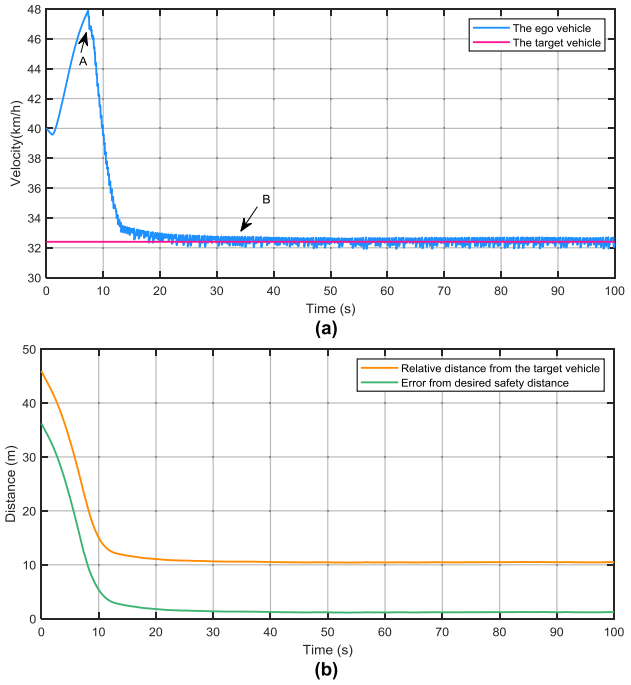

FIGURE 14. The simulation results for the target vehicle driving at a constant low speed. (a) Vehicle speed change. (b) Distance change.

TABLE 4. Training parameters and weight coefficients of reward function.

Parameter name	Parameter value	Parameter name	Parameter value
Discount factor	0.99	Target network replication weight rate	0.001
Actor network learning rate	0.0001	Critic network learning rate	0.001
Mini-batch size	64–128	Optimizer type	Adam
Maximum training period	1000	Maximum iteration step	10000
Experience pool state sequence number	8000	Desired speed	50 (km/h)
Gradient threshold	80	k_{e1}	2
$k_{\dot{e}1}$	1	k_{e2}	10
$k_{\dot{e}2}$	5	k_{δ}	0.1
k_r	1	k_d	10
k_s	1		

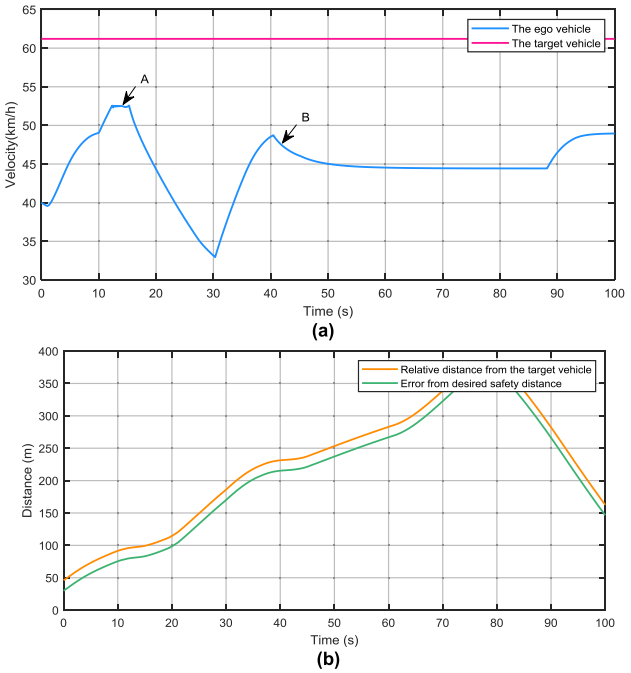

FIGURE 15. The simulation results for the target vehicle driving at a constant high speed. (a) Vehicle speed change. (b) Distance change.

The agent chooses the behavior of improving driving safety by weighing the desired speed reward and the punishment of losing stability. The C and E positions are those where the distance between the vehicle and the target vehicle is too close and less than the dynamic safety distance, leading the agent to control the vehicle by braking. In Fig. 14(a), the target vehicle speed is always less than the desired speed, so the vehicle always follows the target vehicle speed after position A and keeps the desired safe distance. In Fig. 15(a), the target vehicle speed is always greater than the desired speed, so the vehicle speed is only constrained by the ideal speed and road curvature, which is an ideal driving state. Position A is the state where the vehicle has almost reached the desired speed

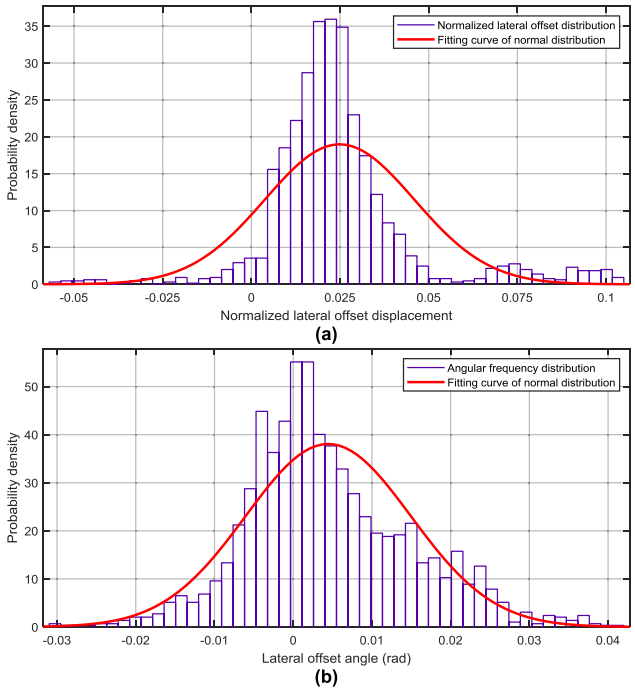

FIGURE 16. Distribution of offset: (a) Distribution of normalized lateral offset displacement. (b) Distribution of lateral offset angle.

in the initial stage. After position A, the vehicle always makes a trade-off between the desired speed and well roll stability. In each distance change figure, the error from the desired safety distance curve is always above the coordinate axis, which shows that the vehicle does not collide at every step and maintains the desired safety distance. The changing trend and amplitude of the relative distance curve and error curve are consistent, which shows that there is no obvious fluctuation of dynamic safety distance between the two vehicles, and the whole control process is relatively stable.

In order to further test and verify the trained model, the normalized lateral offset displacement of the vehicle and the lateral offset angle between the vehicle and the centerline of the lane were recorded, and their probability density distribution is shown in Fig. 16. According to the differences in the curvature of the curved road, the lateral offset between the vehicle and the lane center is slightly different, so there will be a smaller control deviation. From the numerical point of view in Fig. 16(a), the normalized lateral offset is basically kept near the lane centerline (that is, within ± 0.075), with a small number of data points distributed in the normalized lateral offset range of 0.075–0.11, and no data points are close to the left and right edges of the lane. In Fig. 16(b), most of the lateral angle offset is near the zero point, and most of the sample points are within the range of ± 0.02. To summarize, simulation results indicate that the trained model can provide the expected behavior and achieve the effect of lane-keeping.

The comparison of the LTR value of the vehicle before and after the stability control is shown in Fig. 17. If the vehicle stability control is not carried out, the LTR value of the vehicle is mostly distributed in the safe range

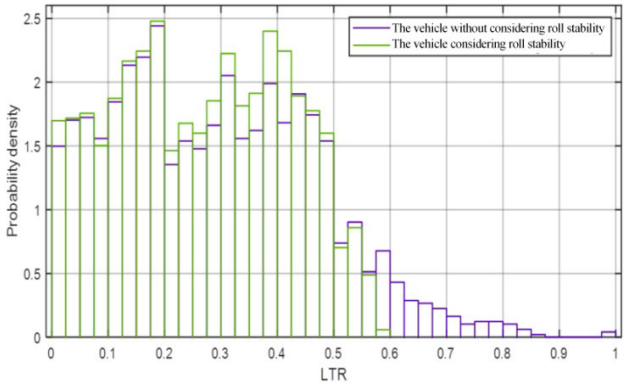

FIGURE 17. The distribution of LTR values during the experiment.

of 0–0.55, though a few LTR values are distributed in the adjacent dangerous area of 0.55–0.8. Although only a very few data points are distributed in the danger areas of 0.8–1.0, it will cause serious traffic safety problems and irreversible control processes, and even make the vehicle rollover, so it is necessary to take active control in advance. After the stability control, although the probability of the LTR value distribution in the 0.3–0.55 area increased significantly, it is still in the safe area range, and the distribution of the LTR value is basically controlled in the range of 0–0.6, and there is no dangerous rollover condition with LTR value of 1. It can be inferred that the decision-making strategy for heavy vehicles can decrease the LTR value efficiently to keep the vehicle stable and safe while operating.

VII. CONCLUSION

For a heavy-duty commercial vehicle under adaptive cruise condition, a semi-rule decision-making method was proposed, and the control accuracy guaranteed to a great extent through the accurate load model. Through the reasonable design of reinforcement learning parameters, the training time of the agent was greatly shortened, which improved the efficiency of the developing reinforcement learning-related algorithms. In addition, the control effect of the agent is very good, and it solves the problems that traditional rule-based decision-making methods cannot solve in complex and unfamiliar environments. The agent learns through continuous interaction with the environment to get rewards or punishments for self-learning, which has the potential to approach or even surpass humans and reflects the superiority of the algorithm. Future work will focus on the perception module and the execution module to build a complete system.

APPENDIX

A. ACC TECHNIQUES

See Table 5.

B. MEASUREMENT METHODS AND CONTENTS OF MDSI

The multi-dimensional driving style inventory (MDSI) presents a comprehensive, multi-dimensional picture of the various orientations people may adopt while driving, and it could delineate a person's profile across differentiated, and even antagonistic, driving orientations.

TABLE 5. Summary of conventional adaptive cruise control techniques.

Category	objective	Description	Characteristic	research
Traditional ACC algorithm	Analyze the characteristics of the longitudinal kinematics by MPC	The characteristic of mutual longitudinal kinematics between the ego and the target vehicle is analyzed by considering the dynamic change law of vehicle spacing, relative velocity, acceleration, the variation of acceleration and preceding vehicles' acceleration.	The technology is relatively mature and the algorithm is interpretable.	[5]
CACC	Optimize IFT for CACC	Connected ACC adapts parameters based on information from other, surrounding vehicles and attenuates the negative effects when communication failures occur.	Enhance the string stability of platoon control in an unreliable V2V communication context.	[8][10] [11]
	Evaluate safety impacts on vehicles' degradation	This study quantitatively evaluated the longitudinal safety impacts of vehicles' degradation in a CACC fleet based on microscopic simulations.	Hierarchical countermeasures and the applicable length of CACC fleets can reduce the safety impacts of degradation.	[12]
	Describe the behavior of CACC vehicles in mixed traffic	Accurately describe the formation and separation of CACC train sets and the behavior of CACC vehicles under the influence of CACC strategy.	It is conducive to the implementation and management of advanced transportation technology.	[9]
	Develop a cooperative signal control algorithm	Adopt the CACC datasets and the datasets collected by the traditional fixed traffic sensors to predict future traffic conditions.	The algorithm is suitable to implement in real-world intersections under various CACC market penetration.	[13][14]
DRL Technology	Learning superhuman proficiency in challenging areas	Neural networks were trained by supervised learning from human experts moves, and by reinforcement learning from self-play.	The artificial agent that is capable of learning to excel at a diverse array of challenging tasks.	[18][19] [27]
	To address the problems of complexity in a critical situation	Propose an adaptive decision-making method that uses reinforcement learning (RL), and the decision-making system is composed of two subsystems. The first subsystem in the architecture for the proposed method criticizes the situation, and the second subsystem implements the decision-making policy.	The results of simulations and experiments demonstrate that the proposed method allows for satisfactory decision-making.	[24][29] [33]
	Improve efficiency through cloud computing	Apply a deep-Q-network model in a multi-agent reinforcement learning setting to guide the scheduling of multi-workflows over infrastructure as a service cloud.	The method is superior to the traditional one in training efficiency and effect.	[22]
DRL and intelligent vehicle	The continuous state decision-making of AV adopts DRL	It consists of two parts: the deep reinforcement learning (DRL) training program and the high-fidelity virtual simulation environment.	RL methodology can offer insight into driver behavior and can contribute to the development of human-like autonomous driving algorithms and traffic-flow models.	[13][17] [23][26] [29][33]
	The planning problem in traffic and decision-making by inverse RL	The road geometry is taken into consideration in the MDP model in order to incorporate more diverse driving styles. The desired, expert-like driving behavior of the autonomous vehicle is obtained.	The desired driving behaviors of an autonomous vehicle using both the reinforcement learning and inverse reinforcement learning techniques.	[20]
	The predictive cruise control and scene understanding	The predictive cruise control and scene understanding concern designing controllers for autonomous vehicles using the broadcasted information from the traffic lights such that the idle time around the intersection can be reduced.	Numerical simulations are presented to validate the efficacy of the proposed methodology.	[25][28]

There are 44 items in this scale, which are divided into 8 factors. The dissociative driving factor, which refers to the driver who is easily distracted during driving, makes driving errors due to this distraction and displays cognitive gaps and dissociations during driving. Anxiety driving factor, which refers to the driver who feels distressed during driving, displays signs of anxiety due to the driving situation and expresses doubts and lack of confidence about his or her

TABLE 6. Multi-dimensional driving style inventory (MDSI).

Factors and items
Factor 1: dissociative driving factor
[11] I daydream to pass the time while driving
[15] lost in thoughts or distracted, I fail to notice someone at the pedestrian crossings
[27] forget that my lights are on full beam until flashed by another motorist
[30] misjudge the speed of an oncoming vehicle when passing
[34] intend to switch on the windscreen wipers, but switch on the lights instead
[35] attempt to drive away from traffic lights in third gear (or on the neutral mode in automatic cars)
[36] plan my route badly, so that I hit traffic that I could have avoided
[39] nearly hit something due to misjudging my gap in a parking lot
Factor 2: anxious driving factor
[4] feel I have control over driving [-]
[7] on a clear freeway, I usually drive at or a little below the speed limit
[10] driving makes me feel frustrated
[25] it worries me when driving in bad weather
[31] feel nervous while driving
[33] feel distressed while driving
[40] feel comfortable while driving [-]
Factor 3: risky driving factor
[6] enjoy the sensation of driving on the limit
[20] fix my hair/ makeup while driving
[22] like to take risks while driving
[24] like the thrill of flirting with death or disaster
[44] enjoy the excitement of dangerous driving
Factor 4: angry driving factor
[3] blow my horn or "flash" the car in front as a way of expressing frustrations
[12] swear at other drivers
[19] when someone tries to skirt in front of me on the road, I drive in an assertive way in order to prevent it
[28] when someone does something on the road that annoys me, I flash them with the high beam
[43] honk my horn at others
Factor 5: high-velocity driving factor
[2] purposely tailgate other drivers
[5] drive through traffic lights that have just turned red
[9] when in a traffic jam and the lane next to me starts to move, I try to move into that lane as soon as possible
[16] in a traffic jam, I think about ways to get through the traffic faster
[17] when a traffic light turns green and the car in front of me doesn't get going immediately, I try to urge the driver to move on
[32] get impatient during rush hours
Factor 6: distress-reduction driving factor
[1] do relaxing activities while driving
[8] while driving, I try to relax myself
[26] mediate while driving
[37] use muscle relaxation techniques while driving
Factor 7: patient driving factor
[13] when a traffic light turns green and the car in front of me doesn't get going, I just wait for a while until it moves
[18] at an intersection where I have to give right-of-way to oncoming traffic, I wait patiently for cross-traffic to pass
[23] base my behavior on the motto "better safe than sorry"
[38] plan long journeys in advance
Factor 8: careful driving factor
[14] drive cautiously
[21] distracted or preoccupied, and suddenly realize the vehicle ahead has slowed down, and have to slam on the breaks to avoid a collision [-]
[29] get a thrill out of breaking the law [-]
[41] always ready to react to unexpected maneuvers by other drivers
[42] tend to drive cautiously
[ ] Numbers in brackets represent the order of the items in the scale.
[-] reversed item.

driving skills. The risky driving factor, which refers to the driver's seeking for stimulation, sensation, and risk during driving and his or her tendency to take risky driving decisions and engage in risky driving. The angry driving factor, which refers to the driver who is hostile towards other drivers, as well as behave aggressively and feel intense anger while driving. The high-velocity driving factor, which refers to the driver's fast driving tendency to display signs of time

TABLE 7. List of symbols.

Symbols	Explanation
m	mass of the whole vehicle
m_v	mass of the empty vehicle body
m_i	mass of the goods in part i
l_v	distance from the first axle to the centroid of the whole vehicle
l_i	distance from the first axle to the (i+1)-th axle
l_{ri}	distance from the centroid of the vehicle to the (i+1)-th axle
l_{r1i}	distance from the centroid of the vehicle in part 1 to the (i+1)-th axle
l_{vi}	distance from i-th axle to the centroid of the vehicle in part i
L_c	distance from the first axle to the centroid of the goods
H	wheel-base of the vehicle
h	height of the centroid of the whole vehicle
h_i	height of the centroid of the vehicle in part i
h_{ri}	distance from the centroid of the vehicle in part i to the roll axle
ϕ	roll angle
$C.G$	centroid of the whole vehicle
$c.g_i$	centroid of the vehicle in part i
K_{bi}	anti-roll stabilizer bar stiffness coefficient of the vehicle in part i
C_i	suspension damping coefficient of the vehicle in part i
a_x	longitudinal acceleration
a_y	lateral acceleration
$\Delta F_{zrmi,zlmi}$	change value of vertical load converted from the lateral moment of the i-th axle
$\Delta F_{zrai,zlai}$	change value of vertical load converted from the pitching moment of the i-th axle
$\Delta F_{zri,zli}$	total change in the vertical load of the i-th axle

pressure while driving. Decompression driving factor, which refers to the driver's tendency to engage in relaxing activities during driving, and aimed at reducing distress while driving. The Patient driving factor, which refers to the driver's tendency to be polite towards other drivers, to feel no time pressure during driving, and to display patience while driving. Careful driving style refers to the driver's tendency to be careful during driving, to plan his or her driving trajectory effectively, and to adopt a problem-solving attitude towards driving-related problems and obstacles. This scale divides eight factors combinations into four different driving styles. Accurately, the reckless and careless driving style was represented by the risky and high-velocity MDSI factors; the anxious driving style was represented by the anxious, dissociative and distress-reduction MDSI factors; the angry and hostile driving style was directly represented by the angry MDSI factor; and the patient and careful driving style was represented by two conceptually related MDSI factors—the careful and patient factors. The specific contents of MDSI are as follows:

Among the 8 factors of MDSI, dissociative driving factor has 8 items(11, 15, 27, 30, 34, 35, 36, 39), anxious driving factor has 7 items(4, 7, 10, 25, 31, 33, 40), risky driving factor has 5 items(6, 20, 22, 24, 44), angry driving factor has 5 items(3, 12, 19, 28, 43), high-velocity driving factor has 6 items(2, 5, 9, 16, 17, 32), distress-reduction driving factor has 4 items(1, 8, 26, 37), patient driving factor has 4 items (13, 18, 23, 38), careful driving factor has 5 items(14, 21, 29, 41, 42), 4 of them are reversed items.

In this paper, 20 drivers were selected for data collection. The drivers were asked to read each item and to rate the extent to which it fits their feelings, thoughts, and behavior during driving on a 6-point scale, ranging from "not at all" (1) to "very much" (6). Selected drivers were asked to complete a packet of questionnaires. The questionnaires were presented in a random order across drivers. The packet included scales tapping driving style, self-esteem, desire for control, impulsive sensation seeking, extraversion, and driving behaviors. Because this paper does not consider the driver's unskilled operation and their personal life and emotional factors when judging the driver's style, the anxiety driving style and the angry and hostile driving style are excluded from the classification results. The patient and careful driving style are refined into the patient driving style and the careful driving style according to the relative score of the two dimensions. Then, the results of the questionnaire randomly selected 5 drivers for each of the three types among a certain number of drivers, that is, conservative drivers with a patient driving style, conventional drivers with a careful driving style, aggressive drivers with a reckless and careless driving style. In the following process, driver action information will be collected according to the style category.

C. MEANING OF SYMBOLS USED IN SECTION III

See Table 7.

D. CALCULATE THE STATE OF THE VEHICLE'S CENTROID

A model for estimating the centroid position of vehicles is constructed, as shown in (A.1). When the acceleration of the vehicle exists, the vertical load of each wheel changes as the acceleration value changes:

$$\begin{cases} mgl_v - F_{z2}l_1 - F_{z3}l_2 - mah - F_a h_a = 0, \\ mg(l_2 - l_v) - F_{z1}l_2 - F_{z2}(l_2 - l_1) + mah + F_a h_a = 0, \\ F_{z1} + F_{z2} + F_{z3} = mg, \end{cases} \tag{A.1}$$

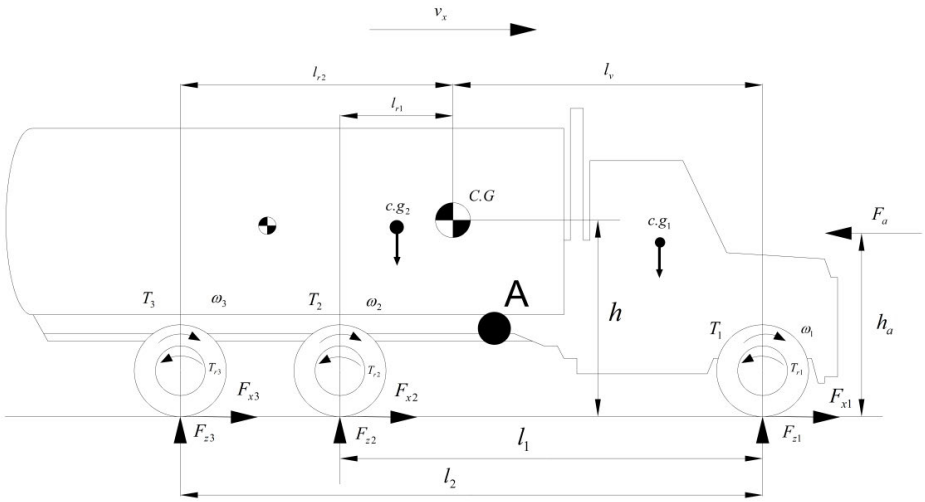

FIGURE 18. Vehicle model for centroid state estimation.

Algorithm 1 Cubature Kalman Filter

Data: tf, $X_{0|0}$, $P_{0|0}$
Result: $\hat{x}_{k|k}(k = 1 - tf)$
1. **Initialization:** $tf = 500$, $X_{0|0} = \hat{x}_{k-1|k-1}$, $P_{0|0} = P_{k-1.k-1}$
2. **For k=1:tf**
3. **Time update: Cholesky decomposition**
$p_{k-1|k-1} = S_{k-1|k-1}S^T_{k-1|k-1}$,
$X_{i,k-1|k-1} = S_{k-1|k-1}\zeta_i + \hat{x}_{k-1|k-1}$,
$X^*_{i,k-1|k-1} = f\left(X_{i,k-1|k-1}\right)$,
$\hat{x}_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} X^*_{i,k|k-1}$,
$P_{k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} X^*_{i,k|k-1}\left(X^*_{i,k|k-1}\right)^T$,
4. **Measurement update: Cholesky decomposition**
$P_{k|k-1} = S_{k|k-1}S^T_{k|k-1}$,
$X_{i,k|k-1} = S_{k|k-1}\zeta_i + \hat{x}_{k|k-1}$,
$Z_{i,k|k-1} = h\left(X_{i,k|k-1}\right)$,
$\hat{z}_{k|k-1} = \sum_{i=1}^{2n}\frac{1}{2n}Z_{i,k|k-1}$,
$P_{zz,k|k} = \frac{1}{2n}\sum_{i=1}^{2n} X_{i,k|k}Z^T_{i,k|k} - \hat{z}_{k|k-1}\hat{z}^T_{k|k-1} + R_k$,
$P_{xz,k|k-1} = \frac{1}{2n}\sum_{i=1}^{2n} X_{i,k|k}Z^T_{i,k|k} - \hat{x}_{k|k-1}\hat{z}^T_{k|k-1}$,
$K^T_k = P_{xz,k|k-1}P^{-1}_{zz,k|k-1}$,
5. **Status update:**
$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K^T_k\left(\hat{z}_{k-1|k-1} - \hat{z}_{k|k-1}\right)$,
$P_{k|k} = P_{zz,k|k-1} - K^T_k P_{xz,k|k-1}K_k$
6. **End**

where F_{zi} is the vertical load of the i-th axle, l_v is the horizontal distance from the centroid to the first axle, h is the height of the centroid from the ground, F_a is the equivalent force of air resistance, h_a is the vertical distance from the action point of the equivalent air resistance force to the ground, and a is the equivalent vehicle acceleration, where:

$$a = a_x + g(\sin\theta + f\cos\theta). \tag{A.2}$$

Since the slip rate of each axle driving force is relatively low during the start-up acceleration stage of the vehicle, this condition is selected to identify the vehicle centroid position. Dynamic analysis of each wheel rotation is given by:

$$J\dot{w}_i = T_i - T_{ri} - r_r F_{xi}, \tag{A.3}$$

where w_i is the rotational angular velocity of the i-th wheel, T_i is the driving torque of the i-th wheel; and for the rear-wheel-drive vehicle, $T_1 = 0$, T_{ri} is the rolling resistance torque generated by the i-th axis. After considering the driving force of the vehicle, the rolling resistance, the air resistance, the ramp resistance, and the acceleration of the wheel, the following vehicle travel equation can be obtained:

$$ma = \sum_{i=1}^{3} F_{xi} - F_a. \tag{A.4}$$

In the steady acceleration stage of the vehicle, there is a special relationship between the longitudinal force and the vertical force of the vehicle, which can be expressed as:

$$F_{xi} = k_s s_i F_{zi}, \tag{A.5}$$

where s_i is the slip ratio, and k_s is the slip rate slope coefficient, which is related to tire characteristics and road adhesion coefficient.

Substituting (A.4) and (A.5) into (A.1) and (A.3), the derivation expression of the centroid position can be obtained:

$$\begin{cases} ma_x = k_s(\sum_{i=1}^{3} s_i F_{zi}) - \frac{1}{2}\rho C_d A v_x^2 - mg\sin\theta, \\ J\dot{w}_i = T_i - r_r(f + k_s s_i)F_{zi}. \end{cases} \tag{A.6}$$

E. RECURRENCE PROCESS OF CUBATURE KALMAN FILTER

After the measurement update, CKF gets the state error covariance at this time and computes the Cholesky decomposition for it. Through the third-order Spherical-Radial cubature rule, the corresponding cubature points are obtained to approximate the nonlinear dynamic system. According to the corresponding weights, the multi-dimensional integration is transformed by the method of weighted integration into the Gaussian domain, and the problem of cubature point summation is replaced by the problem of multi-dimensional integration of nonlinear system and Gaussian probability density product.

REFERENCES

[1] Y. Xiao and D. Ren, "A hierarchical decision architecture for network-assisted automatic driving," in *Proc. IEEE Int. Conf. Energy Internet (ICEI)*, Beijing, China, May 2018, pp. 35–37.

[2] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 1–6.

[3] V. S. Dolk, J. Ploeg, and W. P. M. H. Heemels, "Event-triggered control for string-stable vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3486–3500, Dec. 2017.

[4] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 1, no. 1, pp. 187–210, May 2018.

[5] L. Guo, P. Ge, Y. Qiao, and L. Xu, "Multi-objective adaptive cruise control strategy based on variable time headway," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 203–208.

[6] D. Riley, "Report: Waymo self-driving cars are having problems turning around corners," SiliconANGLE, Palo Alto, CA, USA, Tech. Rep., Aug. 2018. [Online]. Available: https://siliconangle.com/2018/08/28/report-waymo-self-driving-cars-problems-turning-around-corners/

[7] Z. Wang, G. Wu, and M. J. Barth, "A review on cooperative adaptive cruise control (CACC) systems: Architectures, controls, and applications," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 2884–2891.

[8] B. Goñi-Ros, W. J. Schakel, A. E. Papacharalampous, M. Wang, V. L. Knoop, I. Sakata, B. van Arem, and S. P. Hoogendoorn, "Using advanced adaptive cruise control systems to reduce congestion at sags: An evaluation based on microscopic traffic simulation," *Transp. Res. C, Emerg. Technol.*, vol. 102, pp. 411–426, May 2019.

[9] H. Liu, X. Kan, S. E. Shladover, X.-Y. Lu, and R. E. Ferlis, "Modeling impacts of cooperative adaptive cruise control on mixed traffic flow in multi-lane freeway facilities," *Transp. Res. C, Emerg. Technol.*, vol. 95, pp. 261–279, Oct. 2018.

[10] C. Wang, S. Gong, A. Zhou, T. Li, and S. Peeta, "Cooperative adaptive cruise control for connected autonomous vehicles by factoring communication-related constraints," *Transp. Res. C, Emerg. Technol.*, early access, Apr. 29, 2019, doi: 10.1016/j.trc.2019.04.010.

[11] S. Gong, A. Zhou, and S. Peeta, "Cooperative adaptive cruise control for a platoon of connected and autonomous vehicles considering dynamic information flow topology," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2673, no. 10, pp. 185–198, Oct. 2019, doi: 10.1177/0361198119847473.

[12] Y. Tu, W. Wang, Y. Li, C. Xu, T. Xu, and X. Li, "Longitudinal safety impacts of cooperative adaptive cruise control vehicle's degradation," *J. Saf. Res.*, vol. 69, pp. 177–192, Jun. 2019.

[13] C. Desjardins and B. Chaib-Draa, "Cooperative adaptive cruise control: A reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1248–1260, Dec. 2011.

[14] J. Zhou, H. Zheng, J. Wang, Y. Wang, B. Zhang, and Q. Shao, "Multi-objective optimization of lane-changing strategy for intelligent vehicles in complex driving environments," *IEEE Trans. Veh. Technol.*, early access, Nov. 28, 2019, doi: 10.1109/TVT.2019.2956504.

[15] H. Zheng, J. Zhou, Q. Shao, and Y. Wang, "Investigation of a longitudinal and lateral lane-changing motion planning model for intelligent vehicles in dynamical driving environments," *IEEE Access*, vol. 7, pp. 44783–44802, 2019.

[16] N. D. Nguyen, T. Nguyen, and S. Nahavandi, "System design perspective for human-level agents using deep reinforcement learning: A survey," *IEEE Access*, vol. 5, pp. 27091–27102, 2017.

[17] Y. Ye, X. Zhang, and J. Sun, "Automated vehicle's behavior decision making using deep reinforcement learning and high-fidelity simulation environment," *Transp. Res. C, Emerg. Technol.*, vol. 107, pp. 155–170, Oct. 2019.

[18] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016.

[19] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017.

[20] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robot. Auto. Syst.*, vol. 114, pp. 1–18, Apr. 2019.

[21] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[22] Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning," *IEEE Access*, vol. 7, pp. 39974–39982, 2019.

[23] L. Ng, C. M. Clark, and J. P. Huissoon, "Reinforcement learning of dynamic collaborative driving part I: Longitudinal adaptive control," *Int. J. Vehicle Inf. Commun. Syst.*, vol. 1, nos. 3–4, pp. 208–228, Jan. 2009.

[24] M. Zhu, X. Wang, and Y. Wang, "Human-like autonomous car-following model with deep reinforcement learning," *Transp. Res. C, Emerg. Technol.*, vol. 97, pp. 348–368, Dec. 2018.

[25] K. Min, H. Kim, and K. Huh, "Deep q learning based high level driving policy determination," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Changshu, China, Jun. 2018, pp. 226–231.

[26] Z. Gao, T. Sun, and H. Xiao, "Decision-making method for vehicle longitudinal automatic driving based on reinforcement Q-learning," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 3, May 2019, Art. no. 172988141985318.

[27] Z. Hu and D. Zhao, "Adaptive cruise control based on reinforcement leaning with shaping rewards," *J. Adv. Comput. Intell. Intell. Informat.*, vol. 15, no. 3, pp. 351–356, May 2011.

[28] S. Yang, W. Wang, C. Liu, and W. Deng, "Scene understanding in deep learning-based end-to-end controllers for autonomous vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 53–63, Oct. 2019.

[29] V. Mnih, A. P. Badia, and M. Mirza, "Asynchronous methods for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, New York, NY, USA, 2016, pp. 1928–1937.

[30] X.-Y. Lu and S. Shladover, "Integrated ACC and CACC development for heavy-duty truck partial automation," in *Proc. Amer. Control Conf. (ACC)*, Seattle, WA, USA, May 2017, pp. 4938–4945.

[31] W. Luo, P. Sun, F. Zhong, W. Liu, T. Zhang, and Y. Wang, "End-to-end active object tracking and its real-world deployment via reinforcement learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 14, 2019, doi: 10.1109/TPAMI.2019.2899570.

[32] H. Li, Q. Zhang, and D. Zhao, "Deep reinforcement learning-based automatic exploration for navigation in unknown environment," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, doi: 10.1109/TPAMI.2019.2899570.

[33] Z. Huang, X. Xu, H. He, J. Tan, and Z. Sun, "Parameterized batch reinforcement learning for longitudinal control of autonomous land vehicles," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 4, pp. 730–741, Apr. 2019.

[34] D. D. Salvucci, E. R. Boer, and A. Liu, "Toward an integrated model of driver behavior in cognitive architecture," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1779, no. 1, pp. 9–16, Jan. 2001.

[35] J. Elander, R. West, and D. French, "Behavioral correlates of individual differences in road-traffic crash risk: An examination of methods and findings," *Psychol. Bull.*, vol. 113, no. 2, pp. 279–294, Mar. 1993.

[36] O. Taubman-Ben-Ari, M. Mikulincer, O. Gillath, "The multidimensional driving style inventory—Scale construct and validation," *Accident Anal. Prevention*, vol. 36, no. 3, pp. 323–332, May 2004.
[37] W. D. Jones, "Keeping cars from crashing," *IEEE Spectr.*, vol. 38, no. 9, pp. 40–45, Sep. 2001.
[38] Z. Yang, K. Merrick, L. Jin, and H. A. Abbass, "Hierarchical deep reinforcement learning for continuous action control," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5174–5184, Mar. 2018.
[39] T. De Bruin, J. Kober, and K. Tuyls, "Experience selection in deep reinforcement learning for control," *J. Mach. Learn. Research.*, vol. 19, no. 1, pp. 347–402, Aug. 2018.
[40] S. Sulaiman, P. M. Samin, H. Jamaluddin, R. A. Rahman, and S. A. A. Bakar, "Dynamic tire force control for light-heavy duty truck using semi active suspension system," in *Proc. Int. Conf. Comput., Commun., Control Technol. (I4CT)*, Kuching, Malaysia, Apr. 2015, pp. 98–102.
[41] I. Arasaratnam and S. Haykin, "Cubature Kalman filters," *IEEE Trans. Autom. Control*, vol. 54, no. 6, pp. 1254–1269, Jun. 2009.
[42] Y. Shi, "Adaptive high-degree cubature Kalman filter with unknown noise statistics," *J. Inf. Comput. Sci.*, vol. 11, no. 18, pp. 6703–6712, Dec. 2014.
[43] S. Wang, G. Qi, and L. Wang, "High degree cubature H-infinity filter for a class of nonlinear discrete-time systems," *Int. J. Innov. Comput. Inf. Control.*, vol. 11, no. 2, pp. 627–640, Nov. 2015.
[44] D. Simon, *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006, pp. 399–421.
[45] B. Hassibi, A. H. Sayed, and T. Kailath, "Linear estimation in Krein spaces. I. Theory," *IEEE Trans. Autom. Control*, vol. 41, no. 1, pp. 18–33, Jan. 2016.
[46] W. Li and Y. Jia, "H-infinity filtering for a class of nonlinear discrete-time systems based on unscented transform," *Signal Process.*, vol. 90, no. 12, pp. 3301–3307, Dec. 2010.

MING SUN was born in Yichun, Heilongjiang, China, in 1996. He received the B.S. degree from the Department of Automotive Engineering, Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree with the State Key Laboratory of Automotive Simulation and Control, College of Automotive Engineering.

His research interests include deep reinforcement learning and active control of intelligent vehicle collision avoidance.

WEIQIANG ZHAO received the M.S. and Ph.D. degrees in automotive engineering from Jilin University, China, in 2008 and 2013, respectively.

He is currently an Associate Professor with the State Key Laboratory of Automotive Simulation and Control, China. He is devoting to the study of multibody system dynamics, stability, and advanced control technology. His research interest focuses primarily on the use of these technologies to enhance the performance of commercial vehicles.

GUANGHAO SONG was born in Changchun, Jilin, China, in 1995. He received the B.S. degree in automotive engineering from Jilin University, Changchun, China, in 2018, where he is currently pursuing the M.S. degree in automotive engineering.

His research interests include articulated vehicle dynamics and control and collision avoidance of autonomous vehicles.

ZHIGEN NIE received the B.S. degree in vehicle engineering and the M.S. degree in power machinery and engineering from the Kunming University of Science and Technology, Kunming, China, in 2006 and 2010, respectively, and the Ph.D. degree in vehicle engineering with Jilin University, Changchun, China, in 2014.

He was with the China Automotive Engineering Research Institute and Chongqing Changan Automobile Company Ltd., in 2011 and 2015, respectively. Since January 2019, he has been an Associate Professor with the Faculty of Transportation Engineering, Kunming University of Science and Technology and a Postdoctoral Researcher with LIS Lab (UMR CNRS 7020), Aix-Marseille University, Marseille, France. His research interests include the vehicle system modeling and control, intelligent vehicles control, parameter identification, state estimation, new energy vehicles control, overall optimization of pedestrian, vehicle and road, and fault tolerant control of vehicles. He is a Reviewer of multiple journals, including the *International Journal of Heavy Vehicle System*.

XIAOJIAN HAN received the B.S. degree in automotive engineering from the Henan University of Technology, China, in 2014. He has been taking successive postgraduate and doctoral programs of study for doctoral degree with Jilin University, China, since September 2014, where he is currently pursuing the Ph.D. degree in automotive engineering.

His research interests are in path planning and tracking control, dynamic analysis, and stability control of commercial vehicle.

YANG LIU was born in Weifang, Shandong, China. He received the B.E. and M.S. degrees in vehicle engineering from the Shandong University of Science and Technology, Qingdao, China, in 2014 and 2017, respectively. He is currently pursuing the Ph.D. degree in vehicle engineering with Jilin University.

His research interests include the multivehicles control and advanced control methods of the 4WIS/4WID vehicles.