Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance

Hongqian Wei[a,c,d], Nan Zhang[b], Jun Liang[d], Qiang Ai[a,c], Wenqiang Zhao[a], Tianyi Huang[a], Youtong Zhang[a,c]*

[a] School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

[b] James Watt School of Engineering, University of Glasgow, Glasgow G12 8QQ, UK

[c] Low emission Vehicle Research Laboratory, Beijing Institute of Technology, Beijing 100081, China

d School of Engineering, Cardiff University, Wales CF24 3AA, UK

Corresponding author at: School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, PR China.
Email address: youtong@bit.edu.cn (Y. Zhang)

Highlights:

1-An intelligent torque distribution strategy for DDEVs with the reinforcement learning methodology is proposed.
2-Vehicle active safety and energy-saving performance are both considered in the Markov Decision Process.
3-Twin delayed deep deterministic policy gradient algorithm is deployed for continuous torque vector output and the learning stability.
4-Numerical test and hardware experiment validates its huge strength on handling stability and energy conservation.

Keywords: Electric vehicles; deep reinforcement learning; direct torque distribution; energy efficiency; vehicle safety

Abstract: Distributed drive electric vehicles are regarded as a broadly promising transportation tool owing to their convenience and maneuverability. However, reasonable and efficient allocation of torque demand to four wheels is a challenging task. In this paper, a deep reinforcement learning-based torque distribution strategy is proposed to guarantee the active safety and energy conservation. The torque distribution task is explicitly formulated as a Markov decision process, in which the vehicle dynamic characteristics can be approximated. The actor-critic networks are utilized to approximate the action value and policy functions for a better control effect. To guarantee continuous torque output and further stabilize the learning process, a twin delayed deep deterministic policy gradient algorithm is deployed. The motor efficiency is incorporated into the cumulative reward to reduce the energy consumption. The results of double lane change and snake lane change maneuvers demonstrate that the proposed strategy results in better handling stability performance. In addition, it can improve the vehicle transient response and eliminate the static deviation in the step steering maneuver test. For typical steering maneuvers, the proposed direct torque distribution strategy significantly improves the average motor efficiency and reduces the energy loss by 5.25%-10.51%. Finally, a hardware-in-loop experiment was implemented to validate the real-time executability of the proposed torque distribution strategy. This study provides a foundation for the practical application of intelligent safety control algorithms in future vehicles.

1. Introduction

1.1 Distributed drive electric vehicle

Excessive energy consumption and the environmental crisis are two major challenges that entangle human nerves. The promotion of electric vehicles (EVs) has effectively eased this anxiety [1]. With policy support in China, EVs have experienced a remarkable increase of 29.18% in the last year [2]. In addition, with advanced driving technology, EVs with multiple executors have emerged and attracted more attention from automotive scholars [3]. As a typical multi-actuator vehicle, a distributed drive electric vehicle (DDEV) is regarded as a promising vehicular architecture [4-6]. Explicitly, driven by four independent in-wheel motors as shown in Fig. 1, DDEVs have exhibited several huge potentials: 1) the reasonable power allocation among four motors is able to improve motor operation efficiency, which could reduce the energy consumption [7]; 2) the coordinated torque distribution among the four wheels can enhance the vehicle safe performance, such as vehicle lateral stability and maneuverability [8]. 3) the utilization of multiple driving motors can further upgrade the foundation of fault-tolerant control [9, 10]. Therefore, a reasonable torque distribution can fully utilize the potential of DDEVs. Nevertheless, nonlinear tire saturation and vehicle dynamics cause many difficulties in the active safety control. Furthermore, the energy consumption also differs from to the torque distribution strategy. Aiming at the above objectives, including active safety and energy consumption, a literature review was conducted, which is summarized as follows.

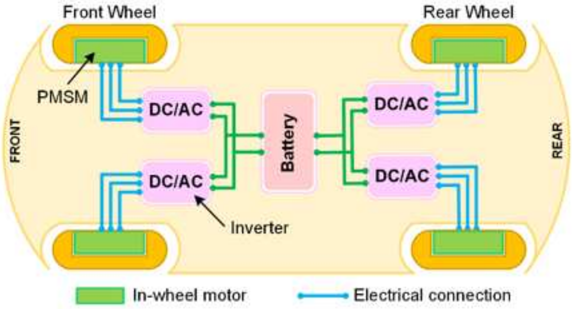


Fig. 1 The driving architecture of distributed drive electric vehicles.

1.2 Literature review

Existing studies have discussed how to improve the handling stability performance, namely direct yaw moment control (DYC) [11-13]. First, the sliding mode controllers are designed to maintain vehicle stability and the external yaw moment is distributed to the four wheels according to the vertical load transfer rules [14-16]. However, these methods only design the reference external yaw moment from the perspective of vehicle attitude regulation, but do not consider optimizing the torque vector based on the over-execution characteristics of a DDEV. Zhai *et al.* [17] designed an electronic stability controller with the fuzzy PID method to generate the external yaw moment and simultaneously guaranteed the maximum lateral stability margin of the tires. Recently, the linear quadratic regulator (LQR) has been used in yaw motion optimization for better handling stability [18, 19]; however, the linear regulation of the control efforts and effects cannot guarantee control accuracy, and it is difficult to deal with the nonlinear constraints. To improve the maneuverability and the vehicle security, Zhang *et al.* [20] investigated the merits of the second-order sliding mode DYC controller, in which the back-stepping method was used to suppress unexpected interference. By deploying the high-frequency switching logic into the original sliding mode controller, the robustness and control optimality of the DYC system are guaranteed but an uncertain boundary is not required. In addition, model predictive control (MPC) has been widely utilized to optimize the yaw motion. Milad et al. [21] realized the integration control of the lateral stability and longitudinal slip of tires using the flexible MPC method. The experiment demonstrated that the control optimality is not compromised, even though there is no driver model. To explore the tire features fully, a linear time varying (LTV) MPC design was employed. In [22], the lateral tire force was linearized using the Tayler expansion formula, and the active steering control in the trajectory tracking process was enhanced using the LTV-MPC method. Furthermore, to reduce the utilization of sensors and to precisely describe the tire lateral forces, reference [23] explored the particle filter algorithm to identify the unknown tire model under the conditions of complicated roads and vehicle uncertainty. The collaboration of the active rear wheel steering control with the DYC system realizes vehicle maneuverability and stability. To utilize the nonlinearity of tires accurately, nonlinear controllers are designed. However, online optimization of the nonlinear model is extremely complicated. Therefore, the offline optimization methods, such as look-up tables [24] and the nearest point approach [25], are widely applied to nonlinear MPC for real-time calculations. However, offline optimization requires a large storage memory for on-board controllers. To address this problem, the continuation/generalized minimal residual (C/GMRES) is incorporated into the nonlinear MPC controllers to achieve fast initialization and reduce the online computational burden [26, 27]. Accordingly, these works pay more attention to vehicle active safety, but barely focus on the energy input. Although these methods can achieve better active safety control, there are still two points to be considered: 1) the nonlinearity of the tire model will cause more difficulties in the design of the controller and online optimization process; 2) seeking the best control performance may cause excessive energy consumption, which contradicts the energy-saving purpose of EVs.

Energy consumption is another significant topic in the design of torque distribution strategies for DDEVs. The energy loss model of the motors was formulated in [28] to identify the optimal driving mode and the total energy can be efficiently allocated among different axles. Zhang *et al.* [29] optimized the traction and braking torques of in-wheel motors to minimize power loss under straight driving conditions. Similarly, the authors of [30] investigated a composite braking energy recovery strategy with the linear optimization of tire loss and motor efficiency. Efficient braking energy recovery control strategies with MPC methods were constructed based on the vehicle longitudinal dynamics in [31, 32]. However, these strategies only consider straight driving conditions and cannot be applied to steering maneuvers. Han *et al.* [33] investigated the relationship between the vehicle stability and sideslip angle, and a gain-scheduling LQR controller was designed to optimize the energy input. However, this method adopts a linear weighted approach and cannot be further extended to other approaches, such as MPC. In [33], the tire slip energy model was emphatically analyzed. On this basis, an integrated framework that includes the external yaw moment and torque distribution control was constructed, which minimizes the tire slip energy and reduces the vehicle sideslip. Hu *et al*. [34] designed a master-slave controller to balance active safety and energy consumption, wherein feedback control of the yaw rate error is employed to acquire the target yaw moment, and torque vector approach is designed to efficiently allocate the yaw moment accordingly. To explore a solution with 7 degrees of freedom (7-DoF) yaw motion, Peng *et al*. [22] utilized the linear MPC to realize yaw moment optimization and torque vector control, in which the motor efficiency characteristic is embedded into the cost function.

In summary, the aforementioned torque distribution strategies have three limitations. The first issue is the comprehensive consideration of both energy consumption and active safety. Because DDEV is a redundant control body, the safety-oriented torque distribution strategy may cause greater energy consumption. Thus, improving the handling stability and reducing the energy consumption of the DDEV control is significant. The second issue concerns the model complexity. Normally, the tire model adopts an empirical or semi-empirical formula in which the nonlinear characteristics are embedded. This would lead to more difficulties in the design of the torque allocation strategy, and the online optimization process is difficult to solve [35]. In addition, the parametric uncertainty of vehicle dynamics also weakens the control performance and even impairs robustness [36]. The third issue relates to enhancing the transient response of the control system and improving the real-time executability. The time hysteresis of the tire model and the deviation of the sampling sensors would delay the transient response of the control systems. Reinforcement learning (RL) methodology explores the optimal control trajectory through continuous iterative solutions. The vehicle dynamics and nonlinear tire features can be approximated with neural networks. Therefore, the torque distribution problem in this study can be addressed using the RL method.

Currently, the RL algorithm is widely used in energy management systems for EVs. Zou *et al.* [37] utilized the Markov chain-based action value function learning to explore the optimal energy allocation, and the Kullback-Leibler divergence rate has also been utilized to determine the optimal parametric update node. To facilitate the all-life-long optimization for the RL policy, Zhou and Xu *et al.* [38] proposed a “multi-step” model-free learning strategy in the energy management of hybrid off-highway vehicles. However, Q-learning in the discrete state space encounters the explosion of multidimensional sampling data [39]. To address this problem, Wu *et al.* [40] studied the approximation of the Q function with a deep Q-network (DQN) in the energy management of hybrid EVs. The fuel economy and the collaboration of multiple driving sources can be guaranteed with continuous state variables, but the action variables are still required to be sampled from the discrete space. Such an action sample is not conducive to continuous action output in inertial systems. Therefore, a deep deterministic policy gradient (DDPG) algorithm was developed to simultaneously approximate the Q-function and policy function in plug-in HEV energy management [41]. Although the deep reinforcement learning (DRL)-based strategy can deal with real-time control problems, several points should be emphasized. First, sampling from the high-dimension action-state spaces should be reasonably approximated to avoid the problems of “curse of dimensionality” and “over-fitting” [42]. Second, the continuous output of the torque vectors should be guaranteed in inertial DDEVs. Finally, the learning process of the algorithm should be as stable as possible, which is essential for the RL algorithm.

1.3 Proposed strategy and contribution

Although numerous EV control strategies, including the vehicle safety or energy analysis, have been studied, a torque distribution strategy that integrates active safety and energy conservation performance still requires more attention. Furthermore, the problems of vehicle nonlinearity and real-time implementation should be addressed. Recent studies have shown that the RL algorithm has a huge potential for approximating the system nonlinearity and guaranteeing real-time executability. Following this idea, this exploratory study proposes DRL-based torque distribution strategy. The direct torque distribution problem is formulated as a *Markov decision process* (MDP), in which the control effort with respect to yaw motion and energy consumption is incorporated as the *cumulative reward*. To achieve a smooth and continuous action output, critic networks are exploited to approximate the Q-function, while the actor network is deployed to fit the policy function to guide the action output. Furthermore, we introduce the *twin delayed deep deterministic policy gradient* (TD3-DDPG) algorithm to enhance the training process and improve the control optimality. The overall framework of

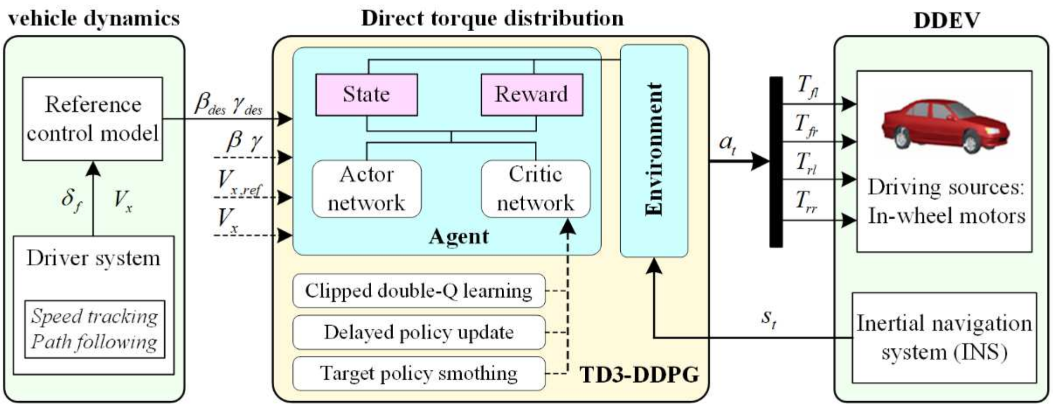


Fig. 2 Diagram of the proposed DRL based direct torque distribution strategy.

the proposed DRL based direct torque distribution strategy is depicted in Fig. 2. The innovation and contribution of this study are as follows.

1) Both active safety and energy conservation performance are considered in the proposed torque distribution strategy. Using the redundancy of DDEVs, the torque distribution strategy can efficiently allocate energy to the four wheels without sacrificing vehicle safety performance.

2) To approximate the nonlinearity feature of vehicle dynamics fully, the RL algorithm is introduced, in which the actor-critic network is utilized to train the agent for better control performance.

3) The TD3-DDPG algorithm is employed to address the problem of over-estimated Q value and stability of the learning process.

1.4 Organization of this paper

The reminder of this paper is detailed in this part. Section 2 formulates a DDEV torque distribution problem, in which the DDEV model and control-oriented reference control trajectory is formulated. Section 3 develops a DRL-based direct torque distribution method. The DDEV torque distribution problem is formulated as an MDP and TD3-DDPG algorithm is utilized to obtain the optimal torque vector. In Section 4, we validate the performance of the proposed control strategy with the help of co-simulation platform and hard-in-ware experiments. Meanwhile, the effectiveness of proposed method is compared with the traditional model-based approaches on the typical critical steering maneuvers. Section 5 concludes this paper and outlooks the future work.

2. Problem formulation

In this section, the vehicle dynamic system and control basis is mathematically formulated. First, a 2 degree of freedom vehicle is depicted with consideration of the lateral and yaw motions., the reference control model is utilized to generate the reference targets to follow the predetermined trajectory while guaranteeing the vehicle stability.

2.1 Vehicle system dynamics

Considering the vehicle lateral and yaw motions, a nonlinear dynamic DDEV model is depicted in Fig. 3. The mathematical expression with respect to the sideslip angle and yaw rate is formulated in following equations.

Lateral movement:

$$mV_x\left(\dot{\beta}+\gamma\right)=F_{yf}\cos\delta_f+F_{yr} \tag{1}$$

Yaw motion:

$$I_z\dot{\gamma}=l_fF_{yf}\cos\delta_f-l_rF_{yr} \tag{2}$$

where m and I_z are the total vehicle mass and yaw inertia coefficient, respectively. δ_f represents the front steering angle. F_{yf} and F_{yr} denote the front and rear lateral forces of tires. l_f and l_r, respectively, denote the distances from the center of gravity to front and rear axles. Since tire lateral forces change with their respective slip angles, the tire slip angles are also defined as follows.

$$\alpha_f=\beta+\frac{l_f\cdot\gamma}{V_x}-\delta_f \tag{3}$$

$$\alpha_r=\beta-\frac{l_r\cdot\gamma}{V_x} \tag{4}$$

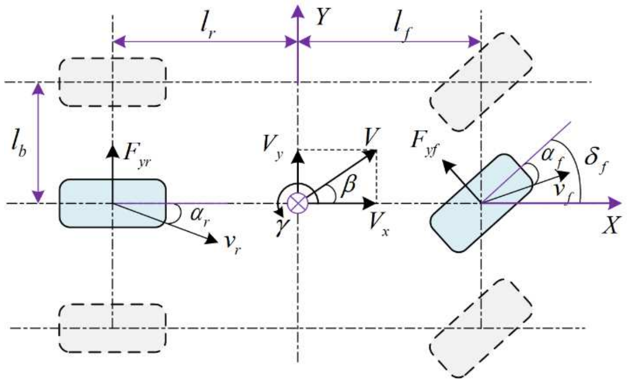


Fig. 3 Illustration of vehicle system dynamics.

2.2 Reference dynamic model

The reference dynamic model is utilized to generate the reference sideslip angle and yaw rate by setting the derivate of all state variables in Eq. (1) and Eq. (2) as zero in the equilibrium state.

$$\begin{cases} \beta_{des} = R_\beta \cdot \delta_f \\ \gamma_{des} = R_\gamma \cdot \delta_f \end{cases} \tag{5}$$

$$\begin{cases} R_\beta = \dfrac{l_r C_r - m l_f V_x^2 / L}{l_r C_r \left(1 + K V_x^2\right)} \cdot \dfrac{l_r}{L} \\ R_\gamma = \dfrac{1}{1 + K V_x^2} \cdot \dfrac{V_x}{L} \end{cases} \qquad \begin{cases} K = \dfrac{m}{L^2} \cdot \dfrac{l_f C_f - l_r C_r}{C_f C_r} \\ L = l_f + l_r \end{cases} \tag{6}$$

where the β_{des} and γ_{des} denote the reference sideslip angle and the reference yaw rate. *K* is the understeering coefficient and L denotes the tracking distance from the front axle to the rear axle.

The lateral acceleration is considerably influenced by road conditions. Thus, the reference yaw rate is bounded by the road adhesion μ .

$$\left|\gamma_{des}\right| = \min\left\{\left|R_\gamma \delta_f\right|, \left|\mu g / V_x\right|\right\} \cdot sign\left(\delta_f\right) \tag{7}$$

where $\pm \mu g / V_x$ is the boundary of yaw rate derived from the lateral acceleration limitation $a_y \le \mu g$. Furthermore, to optimize the lateral motion and reduce the tire slipping in practice, the desired sideslip angle is usually settled as zero, i.e. $\beta_{des} = 0$.

3. Deep reinforcement learning based direct torque distribution strategy

The proposed direct torque distribution strategy is shown in Fig. 2, and consists of three parts: the reference vehicle dynamic system, direct torque distribution controller, and DDEV platform. In reference vehicle dynamics, the driver in the CarSim can output the front steering angle according to the steering demand. Then, the front steering angle is used as the reference control input for the torque distribution system. The torque distribution controller adopts the twin delayed deep deterministic gradient algorithm, in which the DDEV constitutes an agent. Then, the explicit torque commands are applied to the four wheels and the vehicle states are updated simultaneously. Because the proposed torque distribution strategy employs an intelligent RL algorithm, thus, the preliminary of RL algorithm is presented first.

3.1 Description of reinforcement learning algorithm in the vehicle system

3.1.1 Markov Decision Process

The torque commands of the four wheels should be reasonably allocated to enhance the active safety and energy conservation performance of DDEVs. However, the vehicle system contains time-varying tire parameters, which would pose a significant challenge to the optimal torque vector allocation. Therefore, the torque distribution problem is a nonlinear multi-objective optimization problem. RL is an effective method for solving complex multidimensional optimization. The optimal torque distribution in this study can be abstracted as an *MDP* with DDEV states s_t, torque-vector commands a_t, and immediate reward r_t. Specifically, the agent (DDEV) takes an action and the vehicle environment is updated to a new state according to the road information and torque command in each episode. According to the vehicle state, the MDP produces an immediate reward. With repeating iterations, the neural networks in the actor and critic parts would be trained well. Therefore, the agent attains the policy function, which offers maximal cumulative rewards. In summary, MDP is formulated as a five-tuple [43], such as $MDP=\{S,A,P,R,\lambda\}$, where *A* denotes the action set and *S* denotes the state space of the DDEV. $P:S\times A\times S\rightarrow[0,1]$ depicts the probability density function which describes the state transition process, satisfying the *Markov Property*. The symbol $R:S\times A\rightarrow\square$ is the reward set, and λ denotes the discounted factor for better fitting the future reward.

In this study, the *state* is defined as real-time sampling information, such as sideslip angle, yaw rate, and longitudinal velocity. For optimal control performance, the errors and their integrator are also regarded as the state because they are directly related to the vehicle motion optimization. The vehicle longitudinal velocity is incorporated into the state tuple because the motor efficiency varied with the motor speed. The *action* is defined as a torque vector regarding four the driving motors. Explicitly, they are expressed as follows.

$$s_t=\left\{\beta,\int\beta,\Delta\gamma,\int\Delta\gamma,\gamma,V_x,\Delta V_x,\int\Delta V_x\right\} \tag{8}$$

$$a_t=\left\{T_{fl},T_{fr},T_{rl},T_{rr}\right\} \tag{9}$$

where β denotes the sideslip angle and $\int\beta$ is its integration value. $\Delta\gamma=\gamma_{des}-\gamma$ denotes the error between the reference and actual yaw rate; $\Delta V_x=V_{x,des}-V_x$ represents the reference and actual longitudinal velocities, respectively. T_{ij} denotes the torque demand of the four wheels, and symbols $ij=fl,fr,rl,rr$ designate the front left wheel, front right wheel, rear left wheel, and rear right wheel, respectively.

The reward plays a significant role in the DRL because it can guide the agent to pursue the optimal control effect. In this study, we focus on the vehicle safety performance and energy consumption. Therefore, two types of rewards were introduced. In detail, one is related to active safety which is included into R_1 and the other is concerned with the energy consumption, namely R_2.

$$r_t(s_t,a_t)=R_1+R_2 \tag{10}$$

$$R_1=\left(\underbrace{c_1\cdot H_\beta+c_2\cdot H_\gamma}_{reward}\right)-\left(\underbrace{x(t)^T\cdot Q\cdot x(t)}_{system\ performance}\right)-\left(\underbrace{c_3\cdot H_s+c_4\cdot H_f}_{system\ constraints}\right) \tag{11}$$

where $H_{(\bullet)}$ denotes the Boolean variable. Specifically, $H_\beta=1$ if the sideslip angle satisfies $|\beta|<0.0375$, otherwise, $H_\beta=0$; $H_\gamma=1$ is the yaw rate error if $|\gamma_e|<0.00873$ is satisfied, otherwise $H_\gamma=0$; $H_s=0$ if the yaw rate error does not exceed the limit $|\Delta\gamma|\le\Delta\gamma_{\max}$, otherwise $H_s=1$; $H_f=1$ if the state variables of sideslip angle β and yaw rate γ exceed their maximum values; otherwise $H_f=0$. $x(t)=[\beta(t)\ \gamma(t)\ \Delta V_x(t)]$ expresses the vehicle state and Q is the weight matrix $Q=diag(0.5,120,2)_{3\times3}$ of the control performance.

$$R_2=-\left(\underbrace{\Delta u(t)^T\cdot R\cdot\Delta u(t)}_{control\ effort}+\underbrace{c_6\cdot\sum T_{ij}\cdot\omega_{ij}\cdot\eta_{ij}^k}_{power\ input}\right) \tag{12}$$

To ensure smooth operation and eliminate noise, the command of the motor torque should have a small deviation and fluctuation. In a mathematical expression, the control cost function should reduce the magnitude of the torque vector increment. $\Delta u(t)=\left[\Delta T_{fl}\ \Delta T_{fr}\ \Delta T_{rl}\ \Delta T_{rr}\right]$ is the torque vector increment and

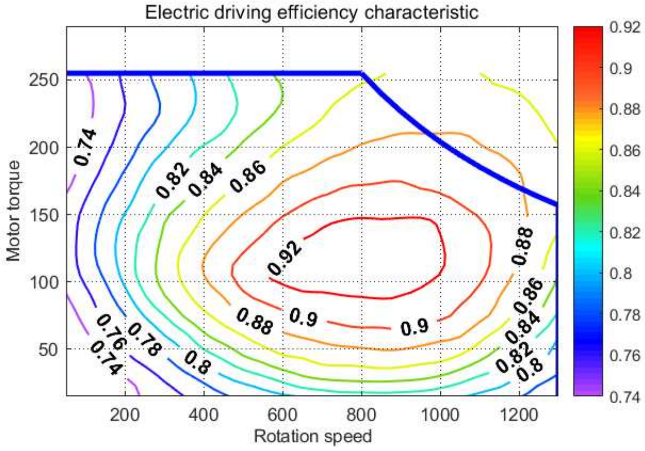


Fig. 4 Electric machine driving efficiency map.

$\Delta T_{ij} = T_{ij}(k) - T_{ij}(k-1)$. $R = (0.05, 0.05, 0.05, 0.05)_{4\times 4}$ is the weight matrix. Parameters $c_1 - c_5$ denote the weight coefficients, which balance the active safety performance and energy efficiency performance. These are obtained by repeated simulations. To achieve a better energy-saving effect, the total power input is optimized with $P_{ij} = T_{ij} \cdot \omega_{ij} \cdot \eta_{ij}{}^{k}$, where η_{ij} is the motor operational efficiency in the driving mode. If the motor works in the driving mode, $k=1$, otherwise, $k=-1$. The motor efficiency in the driving mode can be obtained by looking up the electric driving efficiency characteristic as shown in Fig. 4, in which the efficiency of the driving and braking modes is assumed to be the same.

The *discounted return* is defined as the cumulative rewards and its mathematical expression is formulated as the cumulative immediate reward from now to future timeslots.

$$U_t = r_t + \lambda \cdot r_{t+1} + \lambda^2 \cdot r_{t+2} + \cdots + \lambda^k \cdot r_{t+k} = r_t + \lambda \cdot U_{t+1} \tag{13}$$

3.1.2 Q-value learning and deep Q network

The Q-learning algorithm is the foundation of the RL algorithm. In this section, the base Q learning algorithm [44] and the conventional DQN [45] framework are described.

Assuming that Q_π denotes the action-value function with the effect of the policy function π, Q_π can be formulated with the expectation of a discounted return. In this sense, the action-value function reflects the system control performance, therefore, the higher the Q function value, the higher the profit of the agent. To facilitate the iterative calculation, the Bellman formulation in a recursive form is expressed [46].

$$Q_\pi(s_t, a_t) = E\left[U_t | s_t = s, a_t = a\right] = E\left[r_t + \lambda Q_\pi(s_{t+1}, a_{t+1}) | s_t = s, a_t = a\right] \tag{14}$$

The core of the Q-learning algorithm is to find the optimal action-value function $Q^*(s_t, a_t)$ from all policy functions π.

$$Q^*(s_t, a_t) = \max_\pi Q_\pi(s_t, a_t) = E\left[r_{t+1} + \lambda \max_a Q^*(s_{t+1}, a)\right] \tag{15}$$

With the optimal Q^* function value, the performance of all actions can be evaluated and thus the best action can be selected. However, $Q^*(s_{t+1}, a_{t+1})$ at the next timestep is related to the state transition function $P(s_{t+1}, r_t | s_t, a_t)$, which is subject to environmental disturbances and the future vehicle information. In fact, Q^* function is rarely utilized in the Q-learning algorithm because of the following two problems: 1) the multidimensional state and action sets would increase the calculation burden and incur the dimensional explosion problem. 2) a random sample from the multidimensional space set is impractical. To this end, the artificial method with neural networks is introduced to approximate the Q function, namely, the DQN [45, 47].

$$Q(s_t, a_t) \approx Q(s_t, a_t; \theta) \tag{16}$$

where θ denotes the parameter matrix. The *temporal difference* (TD) is incorporated into the gradient descent method to train the parameter matrix and improve the accuracy of the approximation. Thus, the total loss function was formulated using the TD error.

$$L(\theta) = \frac{1}{2}\left(\underbrace{y_t - Q(s_t, a_t; \theta)}_{TD\ error}\right)^2 = \frac{1}{2}\left(\underbrace{r_t + \lambda \max_a Q(s_{t+1}, a; \theta)}_{TD\ \ target} - Q(s_t, a_t; \theta)\right)^2 \tag{17}$$

The best action value function can be obtained by searching the global minimum of the loss function of Eq. (17) in a gradient-descent manner.

3.2 Direct torque distribution with deep deterministic policy gradient algorithm

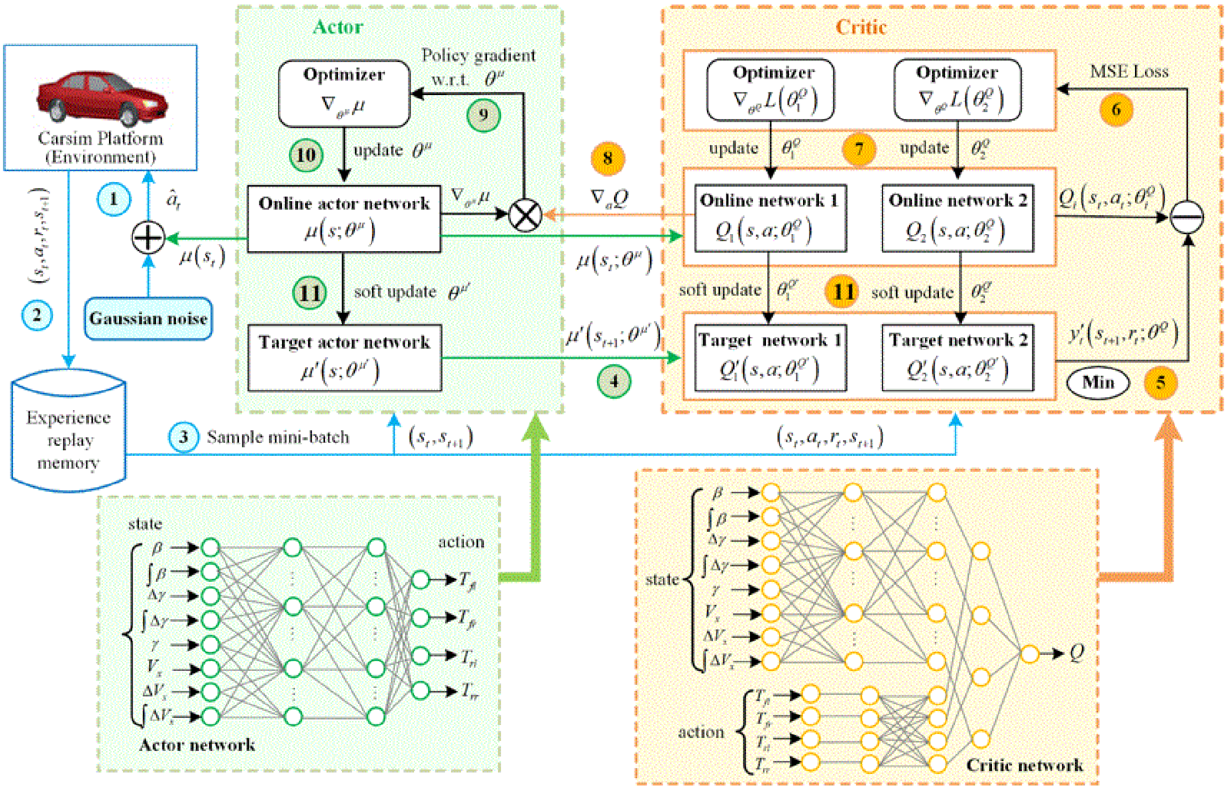
Traditional RL algorithms, such as DQN or the stochastic policy gradient method (SPG) [48], are poorly suited to the continuous action behavior. Therefore, the actor-critic control structure is deployed for the continuous torque vector solution in the inertial system. The framework of detailed implementation process is shown in Fig. 5.

3.2.1 Typical deep deterministic policy gradient algorithm description

The DDPG algorithm adopts the actor-critic framework with two deep neural network approximators. The critic network approximates the Q-value function to evaluate the performance of the given action, which is functionally similar to the DQN. Simultaneously, the actor network approximates the policy function π to directly determine the torque vector at the next time step. The complete DDPG algorithm contains three key components: network learning and parameter update, noise exploration, and experience replay [49].

1) Network learning and parameter update

Typically, an agent in the DDPG algorithm selects the deterministic policy function $a_t = \mu(s_t)$ instead of



Actor
Critic
Carsim Platform (Environment)
Optimizer $\nabla_{\theta^\mu}\mu$
Policy gradient w.r.t. θ^μ
9
10
update θ^μ
Online actor network $\mu(s;\theta^\mu)$
$\nabla_{\theta^\mu}\mu$
8
$\nabla_a Q$
1
$\hat{a}_t$
$\mu(s_t)$
(s_t, a_t, r_t, s_{t+1})
2
Gaussian noise
$\mu(s_t;\theta^\mu)$
11
soft update $\theta^{\mu'}$
Target actor network $\mu'(s;\theta^{\mu'})$
$\mu'(s_{t+1};\theta^{\mu'})$
4
Experience replay memory
3
Sample mini-batch
(s_t, s_{t+1})
(s_t, a_t, r_t, s_{t+1})
Optimizer $\nabla_{\theta^Q} L(\theta_1^Q)$
Optimizer $\nabla_{\theta^Q} L(\theta_2^Q)$
MSE Loss
6
update θ_1^Q
7
update θ_2^Q
Online network 1 $Q_1(s,a;\theta_1^Q)$
Online network 2 $Q_2(s,a;\theta_2^Q)$
$Q_i(s_t,a_t;\theta_i^Q)$
soft update $\theta_1^{Q'}$
11
soft update $\theta_2^{Q'}$
Target network 1 $Q_1'(s,a;\theta_1^{Q'})$
Target network 2 $Q_2'(s,a;\theta_2^{Q'})$
$y_i'(s_{t+1},r_t;\theta^Q)$
Min
5
state
β
$\int\beta$
$\Delta\gamma$
$\int\Delta\gamma$
γ
V_x
ΔV_x
$\int\Delta V_x$
action
T_{fl}
T_{fr}
T_{rl}
T_{rr}
Actor network
state
β
$\int\beta$
$\Delta\gamma$
$\int\Delta\gamma$
γ
V_x
ΔV_x
$\int\Delta V_x$
action
T_{fl}
T_{fr}
T_{rl}
T_{rr}
Q
Critic network

Fig. 5 Framework of detailed implementation process for the TD3-DDPG based torque distribution strategy.

performing an action randomly drawn according to the probability distribution function $a_t \sim \pi(s_t)$ as in the SPG. With the actor network, the policy function is approximated by $\mu(s_t;\theta^\mu)$ parameterized by θ^μ. To train the actor network, the policy gradient ascend algorithm is employed via the back-propagation method. As a result, the policy gradient is updated according to the chain rule (processes 8 and 9 in Fig. 5).

$$\nabla_{\theta^\mu}\mu = E_s\left[\nabla_{\theta^\mu} Q\left(s_t, a_t;\theta^Q\right)\right] = E_s\left[\nabla_a Q\left(s_t, a_t;\theta^Q\right)\cdot\nabla_{\theta^\mu}\mu\left(s;\theta^\mu\right)\right] \tag{18}$$

As presented, the policy gradient of $\nabla_{\theta^\mu}\mu$ is the expect value of $\nabla_a Q \cdot \nabla_\theta \mu$ according to the state variables' probability density function. We can use the Monte Carlo method to estimate this expect function. When the mini-batch data are randomly sampled to form the replay memory buffer, substituting the mini-batch data into the policy gradient formula, the expect value of Eq. (18) is regarded as an unbiased estimation of the policy gradient [50]. Therefore, the policy gradient can be rewritten as

$$\nabla_{\theta^\mu}\mu = \frac{1}{N}\sum_i \nabla_a Q\left(s_t, a_t;\theta^Q\right)|_{s=\hat{s},a=\mu(\hat{s})} \cdot \nabla_{\theta^\mu}\mu\left(s;\theta^\mu\right)|_{s=\hat{s}} \tag{18}$$

With the policy gradient $\nabla_{\theta^\mu}\mu$, the weight of the online actor network can be updated with the learning rate α^μ as follows (process 10 in Fig. 5).

$$\theta^\mu \leftarrow \theta^\mu + \alpha^\mu \cdot \nabla_{\theta^\mu}\mu \tag{19}$$

Similar to the principle in the DQN algorithm, the process of parameter update in the critic network also adopts the gradient descent method. The difference lies in the fact that the parameters with respect to action in the TD target are approximated with a deterministic policy function $\mu(s_{t+1};\theta^\mu)$. Then, the loss function with the mean squared error (MSE) of TD is expressed as follows.

$$L\left(s_t, a_t;\theta^Q\right) = \frac{1}{2}\left(y_t - Q\left(s_t, a_t;\theta\right)\right)^2 = \frac{1}{2}\left(\underbrace{r_t + \lambda Q\left(s_{t+1}, \mu\left(s_{t+1};\theta^\mu\right);\theta^Q\right)}_{TD\ target} - Q\left(s_t, a_t;\theta^Q\right)\right)^2 \tag{20}$$

Previous experience has proved that a single critic or actor network always causes an unstable learning process. This is because the network parameters are frequently updated and synchronously used to calculate the target function. Therefore, two sets of networks, namely online and target networks, are deployed in the actual application. The parameters of the target network are exploited to calculate the TD target (process 4 in Fig. 5). The loss function in (20) can be rewritten as (processes 5 and 6 in Fig. 5).

$$L\left(s_t, a_t;\theta^Q\right) = \frac{1}{2}\left(y_t' - Q\left(s_t, a_t;\theta\right)\right)^2 = \frac{1}{2}\left(\underbrace{r_t + \lambda Q'\left(s_{t+1}, \mu'\left(s_{t+1};\theta^{\mu'}\right);\theta^{Q'}\right)}_{TD\ target} - Q\left(s_t, a_t;\theta^Q\right)\right)^2 \tag{21}$$

The gradient of the loss function in the target critic network parameterized by θ^Q is formulated as.

$$\nabla L_{\theta^Q}\left(\theta^Q\right) = E_{s,a}\left[y_t' - Q\left(s_t, a_t;\theta^Q\right)\cdot\nabla_{\theta^Q} Q\left(s_t, a_t;\theta^Q\right)\right] \tag{22}$$

$$y_t' = r_t + \lambda Q'\left(s_{t+1}, \mu'\left(s_{t+1};\theta^{\mu'}\right);\theta^{Q'}\right) \tag{23}$$

With the gradient of the loss function, the weight of the online critic network can be updated with the learning rate α^Q as follows (process 7 in Fig. 5).

$$\theta^Q \leftarrow \theta^Q + \alpha^Q \cdot \nabla_{\theta^Q} L\left(\theta^Q\right) \tag{24}$$

After obtaining the online network parameters, the parameters in the target network can be updated complying with the *soft update* principle. It is worth noting that the soft update factor τ should be very small to stabilize the learning process. That is, $0 < \tau \ll 1$ (process 11 in Fig. 5).

$$\theta^{Q'} \leftarrow \tau\theta^Q + \left(1-\tau\right)\theta^{Q'} \tag{25}$$

$$\theta^{\mu'} \leftarrow \tau\theta^{\mu} + \left(1-\tau\right)\theta^{\mu'} \tag{26}$$

2) Noise exploration

The DDPG algorithm selects the deterministic action behavior and the action with the highest probability is chosen. However, the agent would be less effective in exploring the unknown behavior in this way. To address this problem and avoid the algorithm being trapped in the local optimal dilemma, the exploration policy is employed with the intervention of exploration (process 1 in Fig. 5).

$$\hat{a}_t\left(s_t\right) = \mu\left(s_t;\theta^{\mu}\right) + \square_t\left(0,\sigma_t^2 I\right) \tag{27}$$

$$\sigma_t = e^{-\zeta \times t} \tag{28}$$

Generally, to encourage bold exploration for better learning ability, the amplitude of the noise function should be very large during the early learning process and be gradually reduced with further continuous iteration. This is because at the initial exploration, the agent has little knowledge on the environment. However, with the continuous learning, the agent has the ability to exploit the accumulated experience and can make a better action from its previous action set. Thus, the parameter σ_t is designed to decay exponentially over time and ζ is defined as the decay rate of the *Gaussian noise*.

3) Experience replay

During the training process, the state variables of the agent are updated through frequent interactions with the environment. The samples of the system state, such as the yaw rate and sideslip angle, can be updated after the execution of the new action (torque vector of the four wheels). Typically, these samples are not independently and identically distributed in the RL algorithms. It is difficult to directly solve the problem directly using numerical approaches. Therefore, an experience replay buffer $\mathcal{D}$ is deployed. This buffer includes a cache of a previous experience tuple $\left(s_t, a_t, r_t, s_{t+1}\right)$ with N_R dimensions (process 2 in Fig. 5). With these experience data, the agent uniformly selects a minibatch with N data sets to train the actor-critic networks and obtain the parameter sets θ^{Q} and θ^{μ} (process 3 in Fig. 5) at each time step. By mixing the previous experience data with the recent ones, the temporal correlation in the playback experience can be weakened. More details can be found in [47]. The total pseudocode of the proposed torque distribution using the DDPG algorithm is outlined in Algorithm I.

3.2.2 Twin-delayed deep deterministic policy gradient

The twin-delayed strategy is the extension of the typical DDPG algorithm. To stabilize the learning process and accelerate iteration, three significant modifications are deployed on the basis of the typical DDPG algorithm, consisting of the clipped double-Q learning, delayed policy updates, and target policy smoothing [51]. Details are depicted as follows.

1) Clipped double-Q learning

Owing to the existence of a susceptible error between the actual and approximated Q values in the DQN, the Q-value estimation will gradually become greater in the iteration progress. This overestimation bias is unavoidable, especially under layer-to-layer propagation using the Bellman equation. To address the problem of overestimation, two sets of critic networks, θ_1^{Q} and θ_2^{Q}, are employed to estimate the Q values separately; and only the minimum value is drawn from the training networks and is applied to the target policy update. Two networks are initialized using two different sets of parameters. In each iteration, only the minimum Q value is selected to update the TD target, which can guide the estimated Q-value to approach its actual value. This method addresses the error accumulation phenomenon of a TD target caused by an over-estimated Q value. This process is called clipped double-Q learning and its mathematical formulation is using the new TD target.

$$y_t' = r_t + \lambda \min_{i=1,2} Q_i'\left(s_{t+1}, \mu\left(s_{t+1}; \theta_i^{Q'}\right)\right) \tag{29}$$

where $(\square')$ denotes the target network parameter. For two sets of online critic networks, the training process shares the same TD target to update the target Q values, and the loss function in the individual critic network is expressed as.

$$L\left(s_t, a_t; \theta_i^Q\right) = \frac{1}{2}\left[y_t' - Q_i\left(s_t, a_t; \theta_i^Q\right)\right] \tag{30}$$

2) Delayed policy updates

The deep Q network is frequently updated in practice and the continuously varying Q parameters may result in incorrect iteration of the actor network. Therefore, the actor network may be trapped in suboptimal policies. To this end, the basic principle is to adjust the update frequency of actor networks much slower than that of critic networks. The suggested critic frequency is twice as high as the actor frequency.

3) Target policy smoothing

The overfitting of optimal value still exists owing to the effect of the function approximation error and the variance of the target. In practice, by introducing small random noise to the target policy network, the estimation of the TD target can be smoothed and the policy can be enabled to exploit the action with the higher Q-value estimations, which is called as *target policy smoothing regularization*.

$$\tilde{a} = \mu\left(s_{t+1}; \theta^{\mu'}\right) + \xi, \quad \xi \,\square\, clip\left(\square\left(0, \sigma_t^2 I\right), -c, c\right) \tag{31}$$

Where the “clip” letter is the clipper operator. In target policy smoothing, the added noise is clipped to the range of possible actions. The target of the clip operator is to keep the target close to the original actions, which avoids introducing impossible actions.

The above equation can be understood that the action can vary within a small range of action space following a certain probability guidance, with which the estimation of the target Q value can be more accurate and robust. When updating the actor network, however, the noise is neglected. This is owing to the fact that the actor can explore the action with the largest Q value, and the intervene of random noise would destroy this exploration. The description of the TD3-DDPG algorithm are depicted in Algorithm II.

Algorithm I: Typical DDPG algorithm for Direct torque distribution

Input: sideslip angle, yaw rate, torque vector. **Output**: the parameters of actor and critic networks.

```
Initialize the buffer cache size $N_R$, minibatch size $N$ and the noise parameters $\theta^N, \mu^N, \sigma$.
Initialize online networks $Q(s,a;\theta^Q)$ and $\mu(s;\theta^\mu)$ with random parameters $\theta^Q$ and $\theta^\mu$, respectively.
Update the target network parameters with $\theta^{Q'} \leftarrow \theta^Q$, and $\theta^{\mu'} \leftarrow \theta^\mu$.
for episode = 1:M do
    Execute the random torque vector in DDEV and observe the following observations with respect to sideslip angle
     and yaw rate to constitute the initial state $s_0$.
    Perform the exploration noise with Gaussian function.
    for timestep t = 1: Ts do
        Select the action with $\hat{a}_t = \mu(s_t;\theta^\mu) + \mathcal{N}_t(0, \sigma_t^2 I)$.
        Execute the external yaw moment $\hat{a}_t$ in the DDEV and observe the new state $s_{t+1}$ as well as the reward $r_t$
        produced by the environment.
        Store the experience data set $(s_t, a_t, r_t, s_{t+1})$ to the buffer cache $\mathcal{D}$ and make a random minibatch sample drawn
        from the $\mathcal{D}$.
        Calculate the TD target with $y_t' = r_t + \lambda Q'(s_{t+1}, \mu(s_{t+1};\theta^{Q'}))$ and update the weight parameters $\theta^Q$ of online critic
        network through minimizing the loss function.
        $\nabla_{\theta^Q} L(\theta^Q) = E_{s,a}\left[y_t' - Q(s,a;\theta^Q) \cdot \nabla_{\theta^Q} Q(s_t, a_t;\theta^Q)\right]$
        Update the weight parameters $\theta^\mu$ of online actor network with the policy gradient ascent method.
        $\nabla_{\theta^\mu} \mu = E_s\left[\nabla_{\theta^\mu} Q(s_t, a_t;\theta^Q)\right] = E_s\left[\nabla_a Q(s_t, a_t;\theta^Q) \cdot \nabla_{\theta^\mu} \mu(s;\theta^\mu)\right]$
        Soft update the weights of target networks:
        $\theta^{Q'} \leftarrow \tau\theta^Q + (1-\tau)\theta^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
    end for
end for
```

Algorithm II: TD3-DDPG algorithm for direct torque distribution

Input: sideslip angle, yaw rate, torque vector. **Output**: the parameters of actor and critic networks.

```
Initialize the online critic networks $Q_1(s,a;\theta_1^Q)$, $Q_2(s,a;\theta_2^Q)$ and the actor network $\mu(s;\theta^\mu)$ with the random weight
  parameters $\theta_1^Q$, $\theta_2^Q$ and $\theta^\mu$, respectively.
Update the target network parameters with $\theta_1^{Q'} \leftarrow \theta_1^Q$, $\theta_2^{Q'} \leftarrow \theta_2^Q$ and $\theta^{\mu'} \leftarrow \theta^\mu$.
Initialize the buffer cache size $N_R$, minibatch size $N$.
for episode = 1:M do
    Execute the random torque vector in DDEV and observe the new states from environment.
    for timestep t = 1: Ts do
        Select the action with $\tilde{a} = \mu(s_{t+1};\theta^{\mu'}) + \xi$, $\xi \sim clip(\mathcal{N}_t(0, \sigma_t^2 I), -c, c)$.
        Execute the torque vector with $\tilde{a}$ and observe the new states as well as the episode reward $r_t$.
        Store experience tuple $(s_t, a_t, r_t, s_{t+1})$ to the buffer cache $\mathcal{D}$ and make a random minibatch sample.
        Calculate the TD target with $y_t' = r_t + \lambda \min Q'(s_{t+1}, \mu(s_{t+1};\theta_i^{Q'}))\big|_{i=1,2}$ and update the critic parameters $\theta_i^Q$ with the
        loss function.
        $\nabla_{\theta^Q} L(\theta_i^Q) = E_{s,a}\left[y_t' - Q_i(s_t, a_t;\theta_i^Q) \cdot \nabla_{\theta_i^Q} Q_i(s_t, a_t;\theta_i^Q)\right]$
        if t mod d, then
            Update the weight parameters $\theta^\mu$ of online actor network with the policy gradient ascent method.
            $\nabla_{\theta^\mu} \mu = E_s\left[\nabla_{\theta^\mu} Q(s_t, a_t;\theta^Q)\right] = E_s\left[\nabla_a Q_1(s_t, a_t;\theta_1^Q) \cdot \nabla_{\theta^\mu} \mu(s;\theta^\mu)\right]$
            Soft update the weights of target networks:
            $\theta_i^{Q'} \leftarrow \tau\theta_i^Q + (1-\tau)\theta_i^{Q'}$, $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1-\tau)\theta^{\mu'}$
        end if
    end for
end for
```

1. Clipped double-Q learning

2. Delayed policy updates

3. Target policy smoothing

4. Results and discussions

The proposed DRL-based direct torque distribution strategy was implemented on the CarSim-Simulink co-simulation platform. The vehicle is driven by four individual permanent magnet synchronous machines (PMSMs) with the parameters listed in Table A2. Double lane change (DLC) and snake lane change (SLC) maneuvers were utilized as standard test maneuvers. The open-loop step-steering maneuver was used to test the transient response and static deviation of the control systems. Furthermore, to validate the real-time executability of the proposed torque distribution strategy, a hardware-in-loop experiment was developed. Two commonly used torque distribution methods, namely, LQR [52] and MPC [53], are presented for comparison in which the nonlinear tire saturation and system dynamics are not considered. The design details of the MPC method is presented in Appendix A.2. Notably, in all tests, the vehicle, that does not have any external safety control strategy, is marked as "baseline". Because there is no additional stability control in the "baseline" vehicle, the vehicle is inclined to be unstable and even trapped into the dangerous accidents if the vehicle steers on the slippery road. As a comparison, the vehicle controlled by the LQR, MPC and DRL methods are marked as "LQR controller", "MPC controller", and "DRL controller".

In the LQR [52] and MPC [53] based torque distribution strategy, two layered controllers are developed, in which the upper controller is utilized to generate the external yaw moment and the lower controller is responsible for allocation of the torque vector to satisfy the upper external yaw moment requirements. Explicitly, the LQR method can obtain the external yaw moment by solving the following Riccati equation.

$$PA + A^T P - PBR^{-1}BP + Q = 0 \tag{32a}$$

$$u(t) = -R^{-1}B^T Px(t) \tag{32b}$$

The MPC controller is established as the depiction in [53] and the regulation is also described in Eq. (A1) of Appendix A2.

4.1 Learning performance

The training process was performed using the help of 16-core Intel(R) Xeon(R) Platinum 8269CY CPU @ 2.5GHz workstation, where the vehicle system and the control strategy are established by the Carsim and Matlab, respectively. The DDEV, as an agent, operates under the standard DLC maneuver. The longitudinal vehicle speed is maintained between 60km/h~100km/h as shown in Fig. 6. In the training environment, the road adhesion is selected as 0.3, which was a very harshly steering condition for civilian vehicles.

The typical DDPG and TD3-DDPG algorithms were implemented in the calculation platform, and the results of their learning performance are shown in Fig. 7. Based on the learning curve of the episode reward, the DDPG algorithm appears to reach a stable episode reward (approximately 6500 after 600 iterations) much

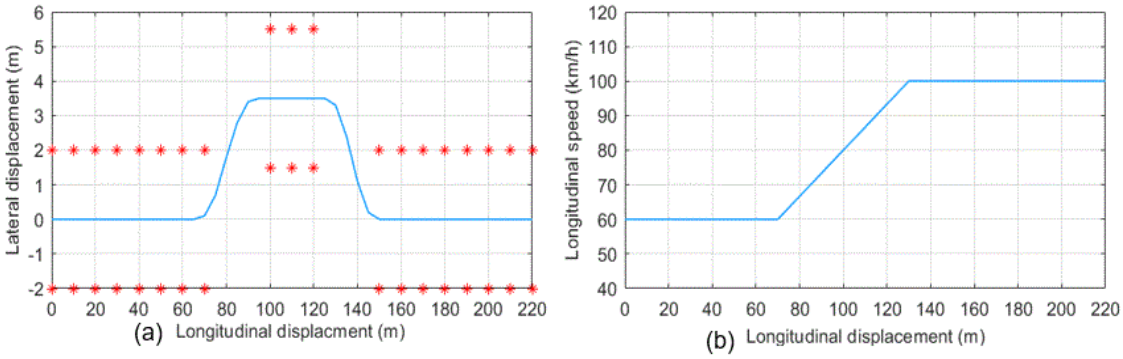


Fig. 6 Steering maneuver setting for learning process. (a) Vehicle driving path (b) Longitudinal speed.

faster while the TD3-DDPG algorithm studies slowly but can eventually reach a higher reward (around 8500 after 1000 iterations) as it avoids the overestimation of the Q value. Before 500 episodes, the reward curve of the DDPG varies drastically and the exploration process is extremely difficult. As a comparison, the reward of TD3-DDPG exhibits much smoother, which illustrates the effectiveness of delayed policy update. Specifically, although the typical DDPG agent can finally reach the stable reward, the reward at around the 1000-th episode would appear as “trap behavior”, resulting in the instability of learning process. In contrast, the TD3-DDPG agent continuously improves the learning behavior and attains a higher reward step by step, which shows its capability to stabilize the learning process.

4.2 Double lane change maneuver

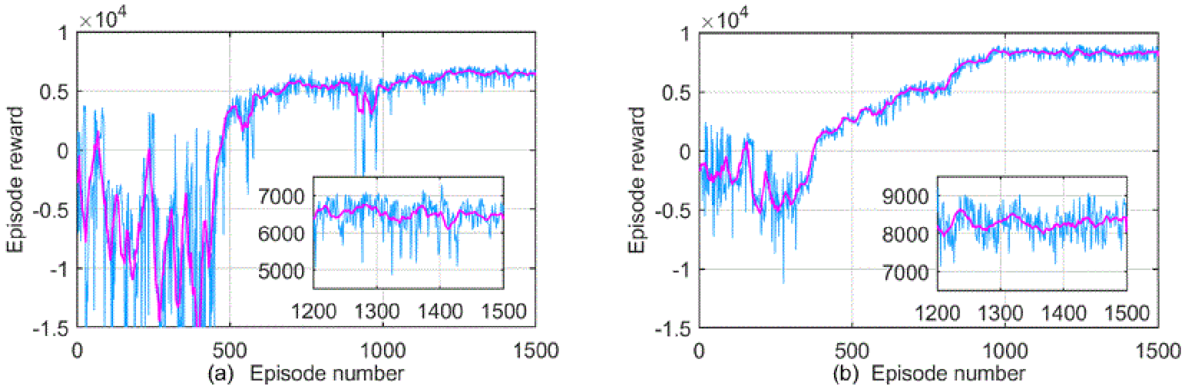


Fig. 7 Episode reward in the learning process. (a) Typical DDPG (b) TD3-DDPG.

The double lane change maneuver is an effective steering test to simulate the overtaking condition or emergency avoidance condition. By keeping the tested DDEV operating on an extremely slippery road with Mu=0.3, the vehicle handling stability performance and energy consumption were evaluated. The target longitudinal vehicle speed was set to 82 km/h. The results regarding the active safety performance are illustrated in Fig. 8-10, and the results with respect to the energy efficiency are depicted in Fig.11 and 12.

The lateral displacement and longitudinal speed-tracking curves are illustrated in Fig. 8(a) and (b), respectively. The lateral displacement with the proposed DRL strategy is only 3.8 m, while the other methods, such as LQR and MPC, incur larger deviations of 3.9-4 m, illustrating that the DRL method enables the DDEV to maintain stability and avoid the risk of lateral slipping. In contrast, owing to the lack of an auxiliary active-safety strategy, the baseline vehicle significantly deviates from the reference trajectory and will lose control severely in the later stage of the steering condition. From Fig. 8 (b), three different methods all enable DDEV to track the longitudinal speed accurately with a speed deviation of less than 1 km/h, which implies that the penalty of the velocity error in the reward can constrain the longitudinal dynamics well.

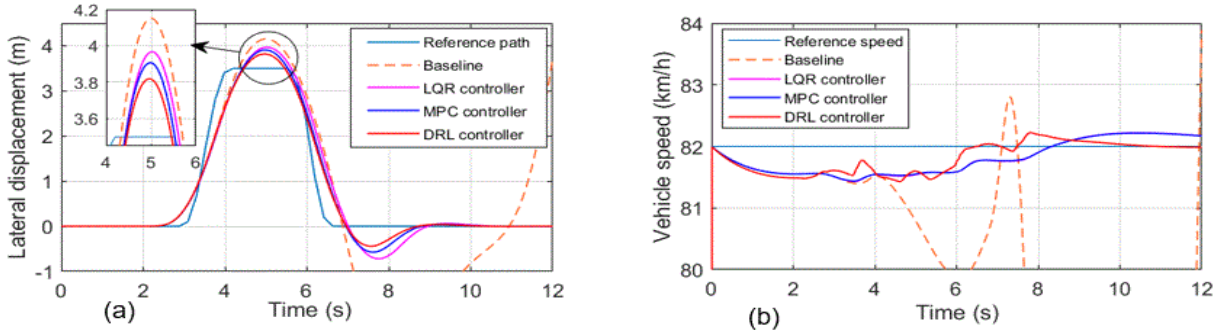


Fig. 8 Vehicle lateral displacement and vehicle tracking speed.

The vehicle sideslip angle plays a significant role in maintaining the vehicle stability. From Fig. 9 (a), it can be seen that the proposed DRL method can reduce the sideslip angle less than 1 deg to the great extent while other methods perform worse with the maximal 1.5 deg deviation. The phase plane of sideslip angle in Fig. 9 (b) indicates the tendency of vehicle to become unstable. The smaller area of phase plane is, the easier it is for the vehicle to stabilize. The proposed DRL approaches can restrict the phase plane to a smaller region, implying that the DRL method is more effective for lateral stability control.

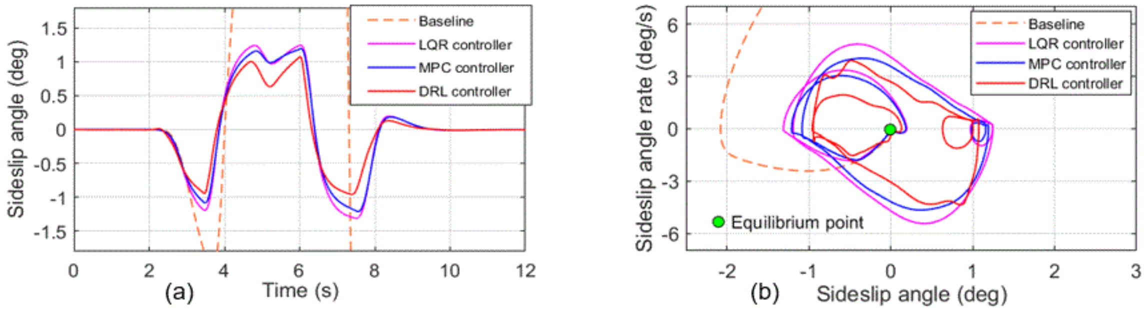


Fig. 9 Vehicle stability characteristics. (a) Vehicle sideslip angle (b) Phase plane of sideslip angle.

The results of the vehicle yaw rate are shown in Fig. 10. Without safety control, the baseline vehicle is inclined to be out of control on the slippery road, resulting in the appearance of extremely dangerous maneuvers. With active safety controllers, such as the LQR and MPC, DDEV can follow the reference yaw rate trajectory well. Comparatively, the yaw rate curve controlled by the LQR method has the largest offset and even goes beyond the constraint boundary, which is very dangerous for DDEVs. This is partially due to nonlinear tire saturation during critical steering conditions. When the yaw rate approaches the boundary, the linear tire stiffness coefficients cannot accurately describe the tire saturations or the lateral forces; therefore, a control deviation exists. Although the yaw rate error can be further reduced by applying a larger weight to the yaw rate term, the external yaw moment required is significantly enlarged, which is not beneficial to the energy efficiency objective for our present study. The comparative results are also found in Appendix A2. In contrast, the yaw rate curve controlled by the proposed DRL method can effectively constrain the yaw rate error, because the DRL method can approximate the tire force through repeated iterations. In addition, when the yaw rate curve error is excessively large and even exceeds the boundary, the punishment on the yaw rate

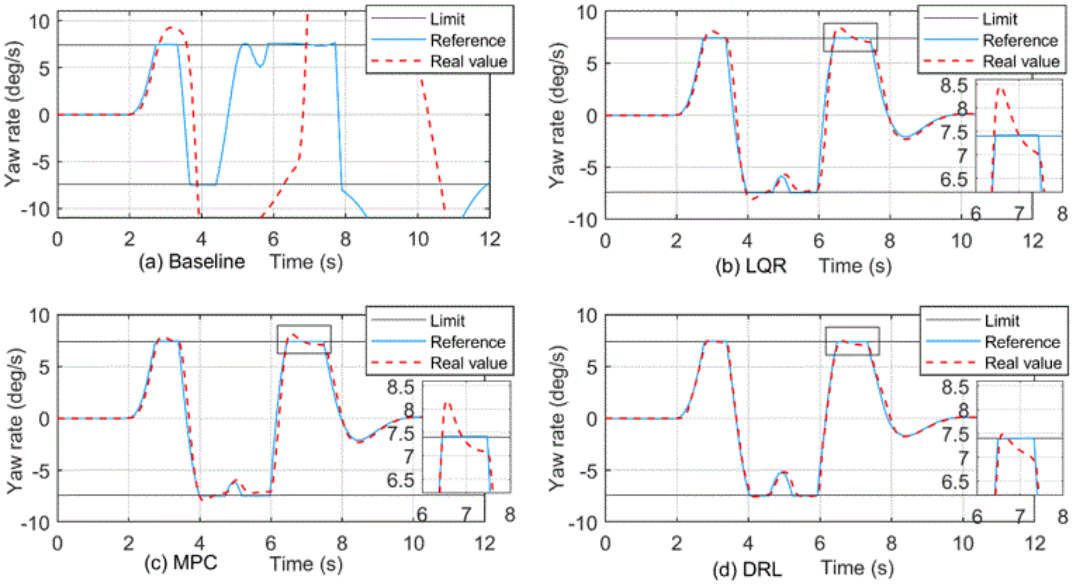
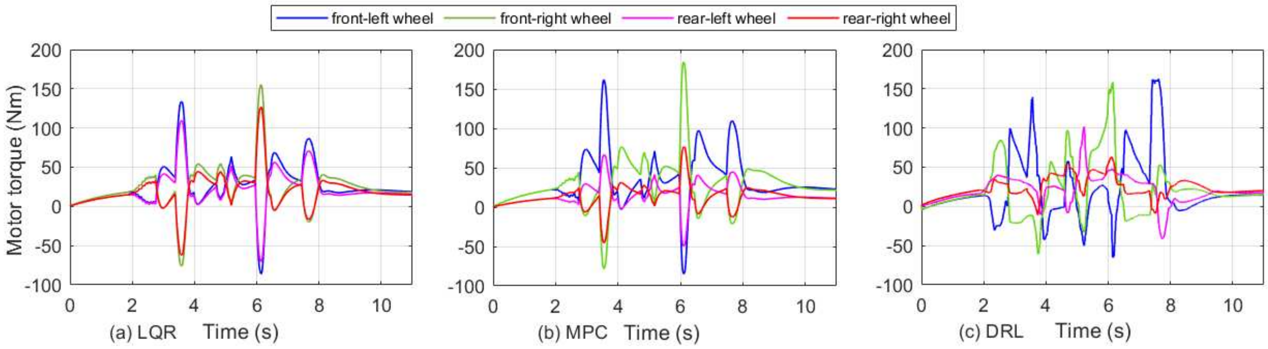


Fig. 10 Yaw rate in the DLC maneuver. (a) Baseline (b) LQR controller (c) MPC controller (d) Proposed DRL controller.

error in the RL algorithm would come into effect. Therefore, a well-trained agent constrains the yaw rate error and guides the yaw rate back within an acceptable range for a high reward. This is conducive to enhancing the handling maneuverability. With the effective control of the yaw rate, the vehicle tends to present the understeering characteristics and requires a greater steering angle to maintain the path stability, which is a tuned steering feature and helps the vehicle return to the balanced state.

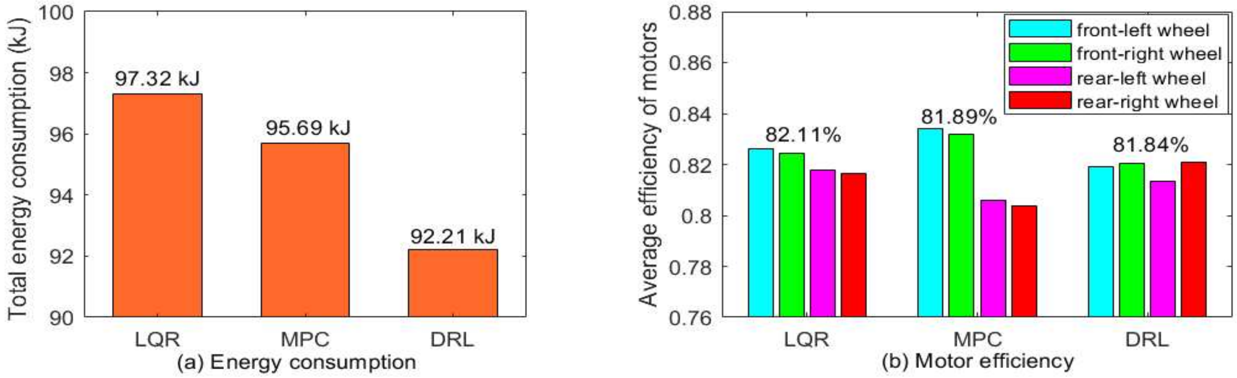
The torque distribution curves are presented in Fig. 11. The torque distribution curves controlled by the LQR and MPC exhibited a similar tendency. Because the optimization objective regarding energy conservation is applied to the RL-based torque distribution strategy, the torque distribution curve of the RL algorithm is even, and the four motors tend to operate in a higher efficiency region. Even the torque demand of the rear right wheel approaches zero torque value for the total vehicle efficiency.



(a) LQR (b) MPC (c) DRL

Fig. 11 Torque distribution of different methods.

Energy conservation ability is another important factor in the design of the torque distribution controller. In this section, we evaluate the energy-saving ability of the proposed DRL strategy in terms of the total energy consumption and average motor efficiency, as shown in Fig. 12. As the total energy consumption histogram shows, the DDEV with the proposed DRL strategy consumes the least energy, only 92.21 kJ, while the MPC and LQR methods tend to use more energy, 95.69kJ and 97.32 kJ, respectively. Therefore, the DRL torque distribution can save energy at least by at least 3.5%-5.25%. The three torque distribution methods share a similar average efficiency, ranging from 81.8% to 82.1%.



(a) Energy consumption (b) Motor efficiency

Fig. 12 Energy consumption in DLC maneuver. (a) Total consumption (b) Motor efficiency.

4.3 Step steering maneuver: response performance analysis

The transient response speed of the control system determines the real-time operation capacity and practical application value of the torque distribution strategy. The open-loop step-steering angle test is a significant means of evaluating the transient response and static deviation of the driving maneuver. Typical model-based torque distribution methods are also deployed for comparison. Assuming that the DDEV operates at the constant speed 72 km/h and the road adhesion coefficient is set as 0.5, a step steering angle from 0~120 deg at time 0.5 s is applied.

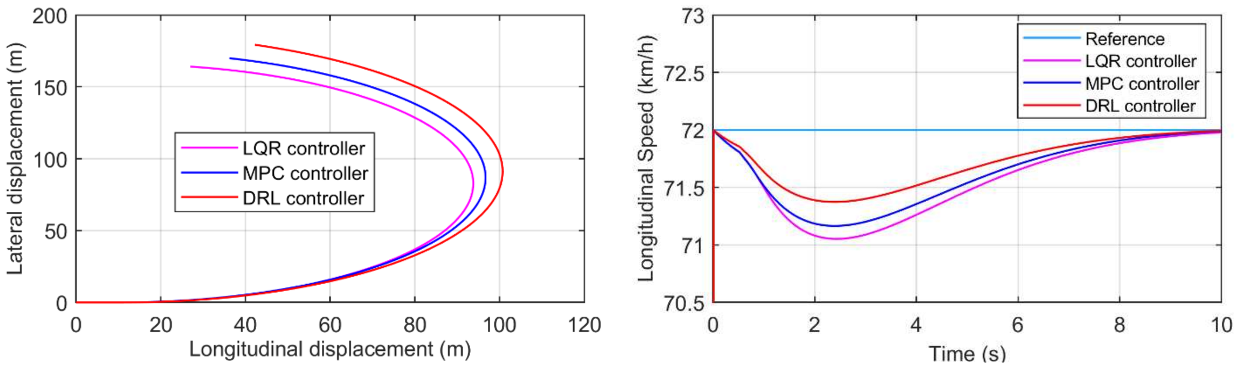


Fig. 13 Vehicle displacement and longitudinal speed in the step steering maneuver.

The curves for the lateral displacement and longitudinal speed are shown in Fig. 13. Compared with the LQR and MPC methods, the proposed DRL method enables vehicles to steer with a larger turning radius. The DDEV controlled by the LQR and MPC methods presents an oversteering characteristic owing to the yaw rate deviation. The vehicle speed is maintained at approximately 72km/h, and the DRL method can reduce the speed error comparatively.

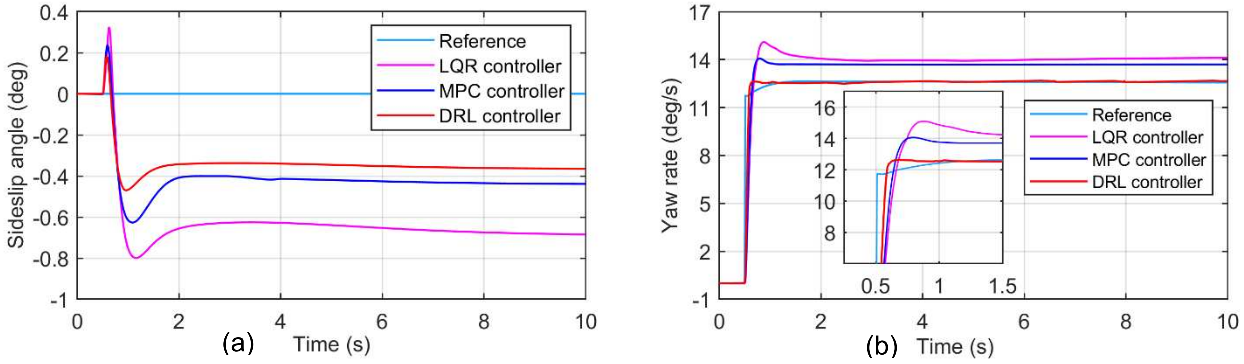


Fig. 14 Vehicle dynamic characteristics (a) Vehicle sideslip angle (b) Yaw rate.

The curves of the vehicle sideslip angle and yaw rate in Fig. 14 indicate the handling stability performance. The sideslip angle controlled by the DRL method consumes the least time (only 0.3s) to the final value of 0.37 deg, whereas the other method requires a longer time to reach a stable value. The yaw rate controlled by the DRL method follows the reference well and favors steering regulation. In contrast, DDEVs with the LQR and MPC control approach to about 14 deg/s, incurring a larger steering radius. As for the transient response, the details of the yaw rate curve show that yaw rate controlled by DRL method has almost no overshoot, and the response time of the system is reduced by approximately 0.5s. Not only that, the yaw rate controlled by DRL method also reduces the static deviation by approximately 2 deg/s compared with what the typical model-based control methods can achieve. The reason for the large static error in the LQR method is that this approach cannot adequately describe the vehicle dynamics and tire saturation. Under the large step steering maneuver, the tire would enter the nonlinear saturation region, which would impair the control optimality of the LQR and MPC approaches. The larger static deviation prevents the vehicle from taking full advantage of the understeering characteristics, and also makes the actual yaw rate exceed the boundary limit provided by the road adhesion. Although the control performance can be improved by applying a larger weight coefficient to the yaw rate or sideslip angle terms, the desired external yaw moment is thereby enlarged which would inevitably increase the total energy consumption for EVs and is in conflict with the target of the optimization of vehicle stability and energy efficiency.

The energy consumption and motor efficiency of the three methods in the step-steering test are illustrated in Fig. 15. In total, the lowest energy consumption is attributed to the DRL method (only 90.51kJ), which presents an energy saving of 18.7% compared to the LQR method. In addition, the proposed DRL method can significantly improve the average motor efficiency by 87.07%, leading the LQR and MPC methods by 3 and 4 percentage points. Finally, the proposed DRL torque distribution method can optimize the operating efficiency of the DDEV and improve the energy distribution layout.

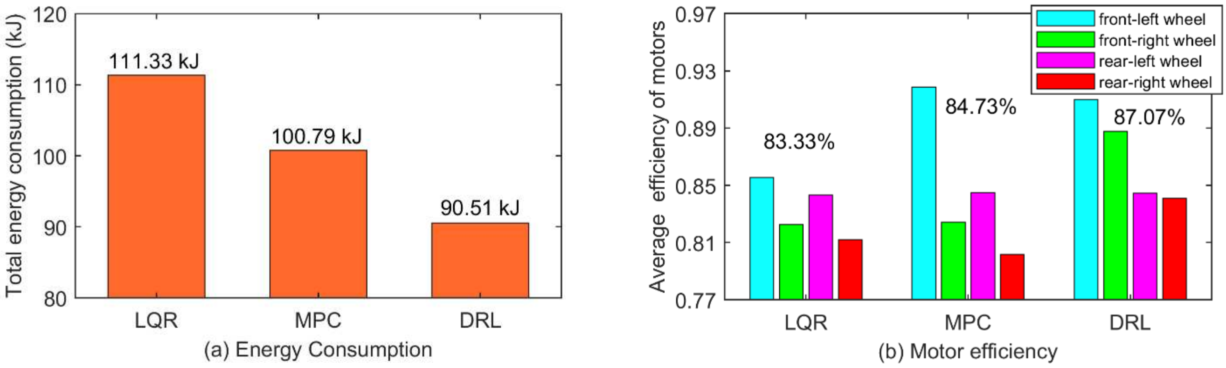


Fig. 15 Energy characteristics. (a) Total energy consumption (b) Average motor efficiency.

4.4 Hardware-in-loop executability test

To validate the real-time executability of the proposed DRL-based torque distribution strategy, a hardware-in-loop (HIL) experiment was implemented using a digital signal processor (DSP, TMS320F28335, TI corporation). The entire framework of the platform is shown in Fig. 16. The offline training processes were

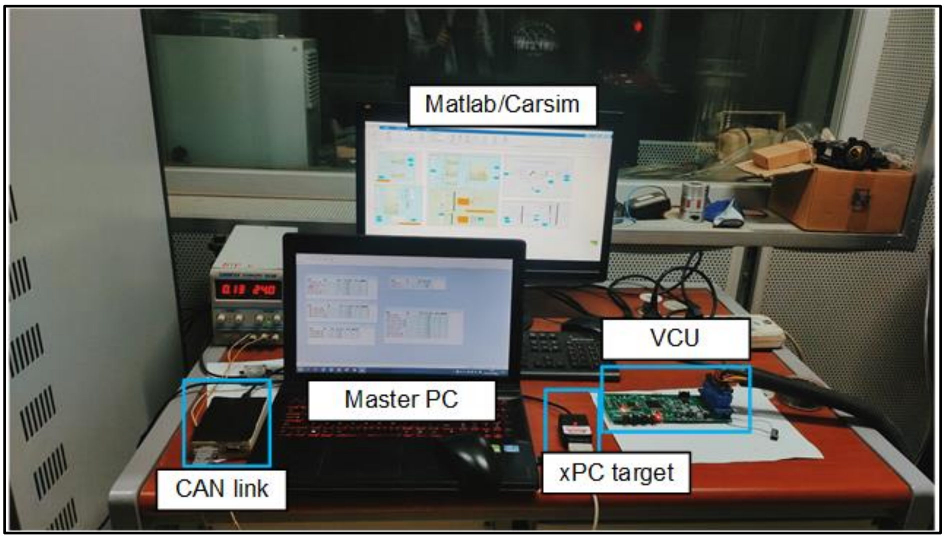


Fig. 16 The HIL test platform.

implemented in the Section 4.1. The traditional DDPG algorithm and TD3 algorithm were developed in the DSP system with well-trained network parameters.

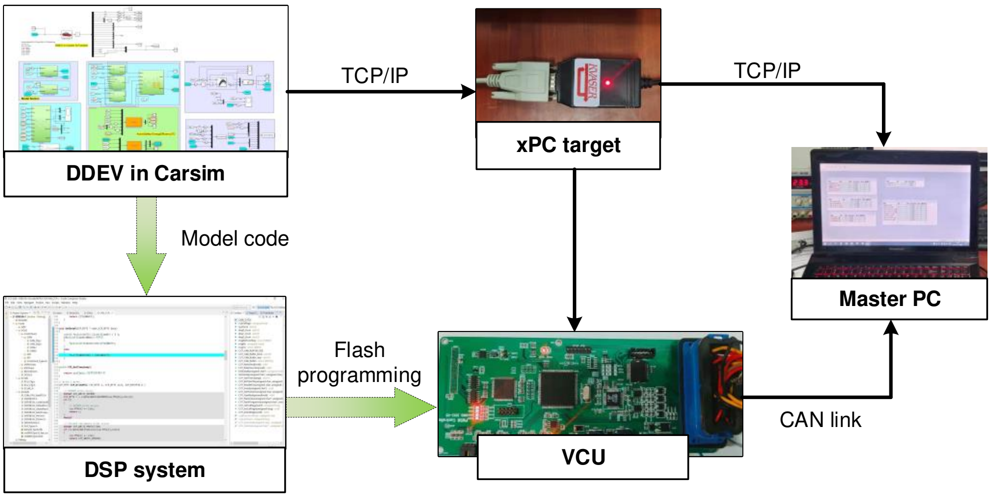


Fig. 17 Framework of HIL test system.

The HIL configuration includes a mater PC (monitoring window), xPC target (Kvaser leaf HS), CAN communication interface and vehicle control unit (VCU) for the torque distribution controller. The standard B-class hatchback is established on the CarSim platform, which is driven by four individual machines. In the experiment, real-time vehicle states, such as the sideslip angle, yaw rate, and longitudinal velocity, were sampled and transmitted to the VCU through the xPC tools. With the sampling states, the agent developed in the VCU can output the optimal torque demands of the four wheels for better handling stability and energy efficiency performance. Simultaneously, all variables and parameters can be monitored in the master PC through the CAN link tool. The DDEV follows the standard DLC steering maneuver traveling on a slippery road with a friction coefficient $\mu = 0.4$ at the reference velocity $V_x = 72\ km/h$. The results are shown in the Fig. 18.

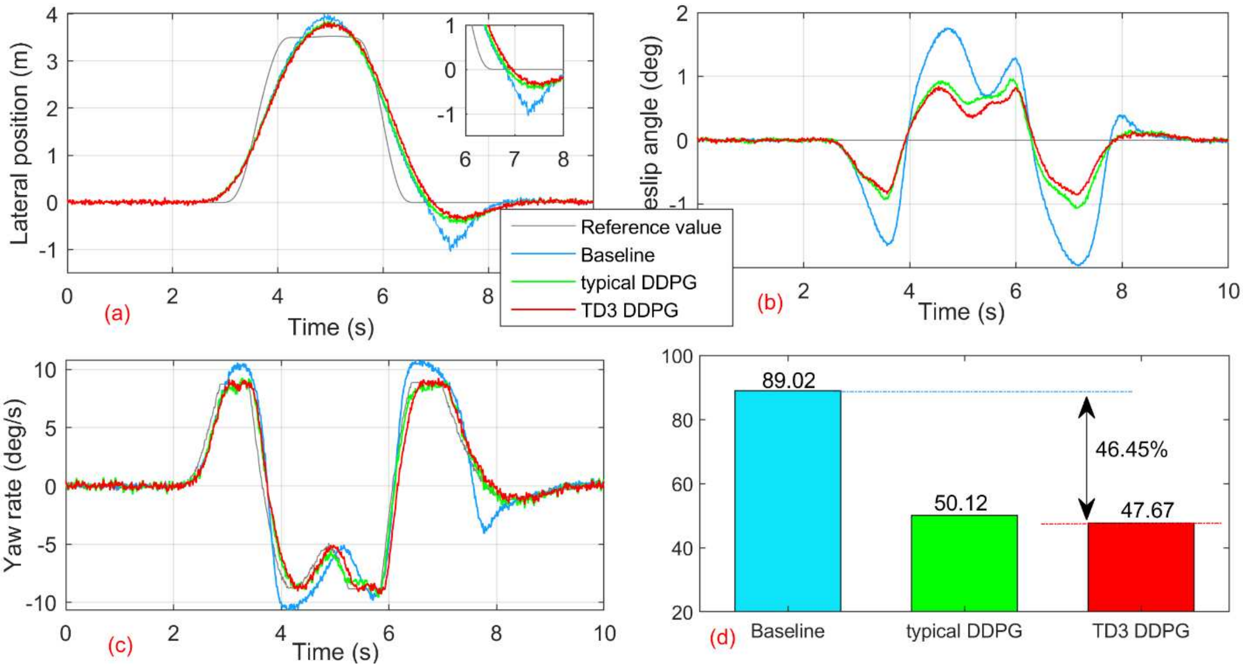


Fig. 18 HIL test results. (a) Lateral displacement, (b) Sideslip angle, (c) Yaw rate curve, (d) Energy consumption of powertrains.

The performance curves of the vehicle without any active-safety control is marked as “Baseline”, whereas the results achieved by the typical DDPG and TD3-DDPG algorithms are marked as “Typical DDPG” and “TD3 DDPG”, respectively.

Fig. 18(a) shows the lateral displacement of the DDEV. Obviously, without an active safety controller, the vehicle is inclined to deviate from the reference path and the maximal offset is approximately 1m at time t = 7s. This indicates that the vehicle tends to become unstable under critical steering conditions. The curve of the sideslip angle marked by “Baseline” also validates this phenomenon with the largest sideslip angle of more than 1.5 deg. In contrast, the lateral displacement controlled by the typical DDPG and TD3-DDPG algorithms presents less deviation, with less than 0.5 m. Specifically, the TD3-DDPG algorithm can further reduce the sideslip angle of the DDPG, as shown in Fig. 18(b). Fig. 18(c) shows the yaw rate curve. It can be observed that the DDPG and TD3-DDPG based torque distribution strategies can well guide the yaw rates to track their references and constrain the yaw rate error as much as possible, which is effective in enhancing the maneuverability of DDEVs. The energy efficiency of the powertrains is shown in Fig. 18(d). Compared with the “Baseline” vehicle, the energy consumption of the motors under the DDPG and TD3-DDPG based torque distribution strategies has been reduced by 38.9 kJ and 42.35 kJ, thus saving energy by 43.7% and 46.45%, respectively. Overall, the results of the HIL test indicate that the proposed TD3-DDPG based torque distribution strategy can conserve the energy consumption of powertrains while maintaining the handling stability performance. In addition, the HIL experiment further validates that the proposed DRL-based torque distribution strategy is capable of real-time execution in the real world.

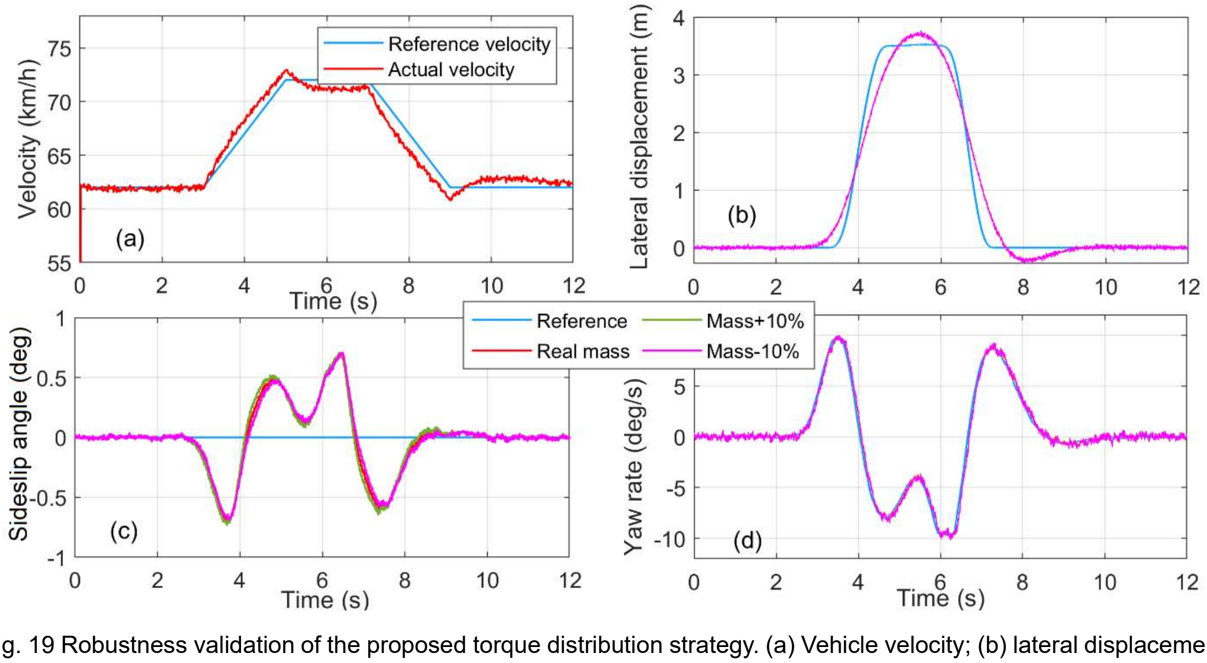


Fig. 19 Robustness validation of the proposed torque distribution strategy. (a) Vehicle velocity; (b) lateral displacement; (c) Sideslip angle; (d) Yaw rate.

On the HIL basis, we implemented another test to validate the robustness of the proposed torque distribution strategy. Explicitly, the vehicle mass in the CarSim is changed ranging from +10% to -10% while the control networks of the DRL strategy in VCU are maintained unchanged in all the tests. The vehicle travels on the road with low adhesion coefficients ($\mu = 0.4$) and the variant velocity is supposed to vary from 62km/h to 72km/h as illustrated in Fig. 19(a). The results of the active safety performance are illustrated in the Fig. 19(b)-(d). Notably, the results with the actual vehicle mass are marked as “Real mass” whereas the settings of vehicle mass in CarSim increased and reduced by 10% are marked as “Mass+10%” and “Mass-10%”, respectively.

From Fig. 19(a), It can be seen that the proposed DRL-based torque distribution strategy can enable the vehicle to track the reference velocity with an acceptable deviation. As depicted in Fig. 19(b), we can find that the lateral displacement has a similar tendency, which implies that the DRL method can achieve good path tracking despite the uncertain vehicle mass. In addition, it can be found that although there are different vehicle masses, the vehicle state parameters almost remain the same, indicating that the DRL algorithm will not fail owing to the variant vehicle dynamics.

5. The limit and the way forward

The DRL-based torque distribution can improve the handling stability of DDEV well while reducing the total energy consumption of powertrains. Meanwhile, it exhibits a strength in dealing with the vehicle nonlinearity such as tire saturation in the low adhesion road. In addition, the HIL experiment further validates its executability in digital signal processors. Nevertheless, there are still some problems that are worth to solve in the future works.

1) The road adhesion plays a significant role for the active-safety control and further research should be capable of identifying the road adhesion. An interesting research direction would be to apply the state constraints (including the road friction coefficients) into the boundary limit for better safety of vehicles.

2) The current study is adaptive to the road condition variation and different maneuvers. However, in future research, the control robustness regarding vehicle dynamics, such as the vehicle mass and vehicle parameter uncertainty, should be paid more attention. It is worth to develop the robust DRL algorithm to address the mentioned problems.

3) The HIL tests validate the theoretical values and the real-time executability of the DRL algorithm in the torque distribution problems. Nevertheless, its practical application in practical vehicles still requires more study in depth. To explore the strength of the RL algorithm, we are conceiving a framework on the torque distribution strategy with the cloud-technology.

6. Conclusion

In this paper, a DRL-based direct torque distribution strategy for DDEVs is proposed to improve the active safety and save energy simultaneously. Unlike the traditional hierarchy control framework, the direct torque vector control is realized without solving the external yaw moment, which is conducive to reduce the design complexity of the controller and guaranteeing control performance. The torque distribution is formulated as an MDP in which the energy loss and active safety index are incorporated into the cumulative reward. The critic and actor networks are utilized to approximate the action and policy value function, respectively. The TD3-DDPG smoothens the learning process and enhances the algorithm performance. The numerical simulation results are as follows:

1) The actor-critic networks effectively deal with the problem of a continuous torque vector solution in inertial distributed drive electric vehicles. The learning curve of the episode reward demonstrates that the twin-delayed algorithm presents better learning stability than the typical deep deterministic policy gradient algorithm. The twin delayed algorithm enables agent to reach a higher episode reward and avoids the overestimation of the Q value in the critic network.

2) The DRL controller can achieve better handling and stability performance. With the proposed torque distribution strategy, the DDEV can reduce the lateral displacement and fully exploit the understeering characteristics to enter back to the equilibrium region quickly, even in critical steering maneuvers.

3) The proposed torque distribution strategy can effectively reduce the power loss and improve the motor efficiency. Under typical steering conditions, such as double lane change and snake lane change maneuvers,

the energy consumption is reduced by 5.2%-10.51% and the average working efficiency of the motors is improved by approximately 2%. The results of the energy evaluation test illustrate that the proposed energy-efficient strategy realizes the energy conservation without compromising the active safety performance of the DDEV.

4) The results of the step steering test show that the proposed direct torque distribution can improve the system response speed and reduce the static deviation. The sideslip angle controlled by DRL is limited to within less than 0.4 deg. The step curve of yaw rate also shows that the proposed strategy can improve the transient response by 0.5 s as fast as possible. The static deviation of the yaw rate error is reduced, which is conducive to improving the maneuverability of DDEVs.

5) The HIL experiment has validated the real-time executability of the proposed DRL-based torque distribution strategy. Furthermore, the TD3-DDPG algorithm shows better control effect than the typical DDPG algorithm. Summarily, the DRL-based control strategy can be a good supplement of the existing control methods especially in the future autonomous ground EVs.

Appendix

A.1 Key parameters of vehicle dynamics and the controller description

Table A1 Nomenclature used in this paper.

Nomenclature		β	Vehicle sideslip angle
DDEV	Distributed drive electric vehicle	a_t	Action output at timestep t
DYC	Direct yaw moment control	s_t	Observed state set at timestep t
RL	Reinforcement learning	r_t	Immediate reward at timestep t
DRL	Deep reinforcement learning	λ	Discounted factor
MDP	Markov decision process	U_t	Discounted return at timestep t
DQN	Deep Q network	$Q(s,a)$	Action function with the given state and action
DDPG	Deep deterministic policy gradient	$\mu(s)$	Policy function with the given state
TD3	Twin delayed DDPG	$L(\theta)$	Loss function for critic network θ
SPG	Stochastic policy gradient	y_t	Temporal difference target
MSE	Mean squared error	θ^{μ}	Parameters in actor network
γ	Yaw rate	θ^{Q}	Parameters in critic network
V_x	Longitudinal vehicle speed	N_R	buffer cache size
T_{ij}	The torque command of in-wheel motors	N	Minibatch size
δ_f	Front steering angle	□$_t$	Gaussian Noise at timestep t

Table A2 Parameters of distributed drive electric vehicle and the permanent magnet synchronous machines.

Symbol	Description	Values
M	Total vehicle mass	1410 kg
l_f	Tracking between front wheel-axle and CG	1.305 m
l_r	Tracking between rear wheel-axle and CG	2.595 m
l_b	Half wheelbase	0.74 m
R_{eq}	Equivalent wheel radius	0.298 m
I_z	Vehicle yaw inertia	2031.4 kg·m^2
C_{f0}	Static cornering stiffness of the front tires	46550 N/rad
C_{r0}	Static cornering stiffness of the rear tires	46350 N/rad
N	Minibatch size	64

P	Maximum power of motors	21 kW
T_{max}	Maximum of motor torque	255 Nm

Table A3 Description of the critic network.

Layer array	Symbol	Function description
1	State	State observation matrix with dimension 8
2	FCs1	Fully connected layer 1 with 10 neurons.
3	RUs1	ReLU activation layer 1 (Rectified Linear Unit, ReLU).
4	FCs2	Fully connected layer 2 with 10 neurons.
5	adds	Addition layer with 2 neurons
6	RUs2	ReLU activation layer 2
7	FCs3	Fully connected layer 3 with 10 neurons.
8	RUs3	ReLU activation layer 3
9	FCs4	Fully connected layer 4 with 1 neuron.
10	Action	Action matrix with dimension 4
11	FCs5	Fully connected layer 5 with 20 neurons.

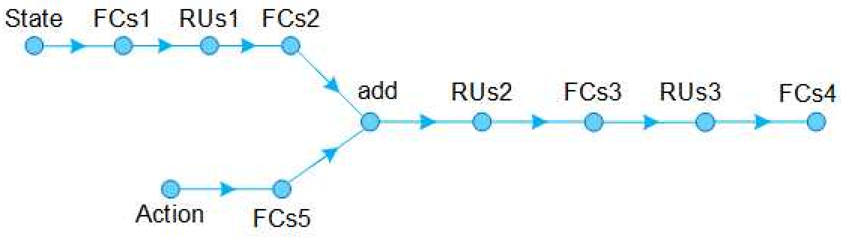


Fig. A1 Framework of critic network.

Table A4 Description of the actor network.

Layer array	Symbol	Function description
1	State	State observation matrix with dimension 8.
2	FCa1	Fully connected layer 1 with 10 neurons.
3	RUa1	ReLU activation layer 1.
4	FCa2	Fully connected layer 2 with 10 neurons.
6	RUa2	ReLU activation layer 2
7	FCa3	Fully connected layer 3 with 4 neurons.
10	Tanh	Tanh activation layer.
11	SC	Scaling layer, ‘Scale’=[255, 255, 255, 255], ‘Bias’=[0, 0, 0, 0]

Table A5 Training parameters of TD3-DDPG agent.

Sample time	0.02
Learn rate of critic network	1e-3
Learn rate of actor network	1e-4
Soft update factor	1e-5
Target smooth factor	1e-3
Discount factor	0.99

Mini batch size	64
Experience buffer	1e6
Maximum episode	2000
Training time consumed	Almost 120 min

Acknowledge

This work is supported by the China Scholarship Council (Grant No. 201906030036) and in part by the National Academician Workstation Project (Grant No. 20180525YX002). The authors appreciate their support and help. Besides, the authors would like to thank the reviewers for their helpful comments and suggestions.

References

[1] Z. Li, A. Khajepour, and J. Song, "A comprehensive review of the key technologies for pure electric vehicles," *Energy,* vol. 182, pp. 824-839, 2019.

[2] D. Ouyang, S. Zhou, and X. Ou, "The total cost of electric vehicle ownership: A consumer-oriented study of China's post-subsidy era," *Energy Policy,* vol. 149, p. 112023, 2021.

[3] J. Hu, J. Li, Z. Hu, L. Xu, and M. Ouyang, "Power Distribution Strategy of a Dual-engine System for Heavy-duty Hybrid Electric Vehicles Using Dynamic Programming," *Energy,* p. 118851, 2020.

[4] L. Zhang, L. Yang, X. Guo, and X. Yuan, "Stage-by-phase multivariable combination control for centralized and distributed drive modes switching of electric vehicles," *Mechanism and Machine Theory,* vol. 147, p. 103752, 2020.

[5] L. Zhang, W. Liu, and B. Qi, "Energy optimization of multi-mode coupling drive plug-in hybrid electric vehicles based on speed prediction," *Energy,* p. 118126, 2020.

[6] Z. Long, J. Axsen, I. Miller, and C. Kormos, "What does Tesla mean to car buyers? Exploring the role of automotive brand in perceptions of battery electric vehicles," *Transportation Research Part A: Policy and Practice,* vol. 129, pp. 185-204, 2019.

[7] R. Wang, Y. Chen, D. Feng, X. Huang, and J. Wang, "Development and performance characterization of an electric ground vehicle with independently actuated in-wheel motors," *Journal of Power Sources,* vol. 196, no. 8, pp. 3962-3971, 2011.

[8] H. Zhang and W. Zhao, "Decoupling control of steering and driving system for in-wheel-motor-drive electric vehicle," *Mechanical Systems and Signal Processing,* vol. 101, pp. 389-404, 2018.

[9] H. Zhang, W. Zhao, and J. Wang, "Fault-Tolerant Control for Electric Vehicles With Independently Driven in-Wheel Motors Considering Individual Driver Steering Characteristics," *IEEE Transactions on Vehicular Technology,* vol. 68, no. 5, pp. 4527-4536, 2019.

[10] B. Li, H. Du, and W. Li, "Fault-tolerant control of electric vehicles with in-wheel motors using actuator-grouping sliding mode controllers," *Mechanical Systems and Signal Processing,* vol. 72, pp. 462-485, 2016.

[11] H. Fang, L. Dou, J. Chen, R. Lenain, B. Thuilot, and P. Martinet, "Robust anti-sliding control of autonomous vehicles in presence of lateral disturbances," *Control Engineering Practice,* vol. 19, no. 5, pp. 468-478, 2011.

[12] C. Hu, R. Wang, and F. Yan, "Integral sliding mode-based composite nonlinear feedback control for path following of four-wheel independently actuated autonomous vehicles," *IEEE Transactions on Transportation Electrification,* vol. 2, no. 2, pp. 221-230, 2016.

[13] S. Fallah, A. Khajepour, B. Fidan, S.-K. Chen, and B. Litkouhi, "Vehicle optimal torque vectoring using state-derivative feedback and linear matrix inequality," *IEEE Transactions on Vehicular Technology,* vol. 62, no. 4, pp. 1540-1552, 2012.

[14] M. Canale, L. Fagiano, A. Ferrara, and C. Vecchio, "Comparing internal model control and sliding-mode approaches for vehicle yaw control," *IEEE Transactions on Intelligent Transportation Systems,* vol. 10, no. 1, pp.

31-41, 2008.

[15] Z. Wang, Y. Wang, L. Zhang, and M. Liu, "Vehicle stability enhancement through hierarchical control for a four-wheel-independently-actuated electric vehicle," *Energies,* vol. 10, no. 7, p. 947, 2017.

[16] D. Zhang, G. Liu, H. Zhou, and W. Zhao, "Adaptive Sliding Mode Fault-Tolerant Coordination Control for Four-Wheel Independently Driven Electric Vehicles," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 11, pp. 9090-9100, 2018.

[17] L. Zhai, T. Sun, and J. Wang, "Electronic stability control based on motor driving and braking torque distribution for a four in-wheel motor drive electric vehicle," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 6, pp. 4726-4739, 2016.

[18] C. Hu, R. Wang, F. Yan, and N. Chen, "Output constraint control on path following of four-wheel independently actuated autonomous ground vehicles," *IEEE Transactions on Vehicular Technology,* vol. 65, no. 6, pp. 4033-4043, 2015.

[19] B. Zhao, N. Xu, H. Chen, K. Guo, and Y. Huang, "Stability control of electric vehicles with in-wheel motors by considering tire slip energy," *Mechanical Systems and Signal Processing,* vol. 118, pp. 340-359, 2019.

[20] L. Zhang *et al.*, "An adaptive backstepping sliding mode controller to improve vehicle maneuverability and stability via torque vectoring control," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 3, pp. 2598-2612, 2020.

[21] M. Jalali, A. Khajepour, S.-k. Chen, and B. Litkouhi, "Integrated stability and traction control for electric vehicles using model predictive control," *Control Engineering Practice,* vol. 54, pp. 256-266, 2016.

[22] H. Peng, W. Wang, C. Xiang, L. Li, and X. Wang, "Torque coordinated control of four in-wheel motor independent-drive vehicles with consideration of the safety and economy," *IEEE Transactions on Vehicular Technology,* vol. 68, no. 10, pp. 9604-9618, 2019.

[23] Q. Wang, Y. Zhao, Y. Deng, H. Xu, H. Deng, and F. Lin, "Optimal Coordinated Control of ARS and DYC for Four-Wheel Steer and In-Wheel Motor Driven Electric Vehicle With Unknown Tire Model," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 10, pp. 10809-10819, 2020.

[24] M. Metzler, D. Tavernini, A. Sorniotti, and P. Gruber, "An explicit nonlinear MPC approach to vehicle stability control," in *Proceedings of The 14th International Symposium on Advanced Vehicle Control*, 2018: Tsinghua University.

[25] M. Canale and L. Fagiano, "Vehicle yaw control using a fast NMPC approach," in *2008 47th IEEE Conference on Decision and Control*, 2008: IEEE, pp. 5360-5365.

[26] N. Guo, B. Lenzo, X. Zhang, Y. Zou, R. Zhai, and T. Zhang, "A Real-time Nonlinear Model Predictive Controller for Yaw Motion Optimization of Distributed Drive Electric Vehicles," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 5, pp. 4935-4946, 2020.

[27] N. Guo, X. Zhang, Y. Zou, B. Lenzo, and T. Zhang, "A Computationally Efficient Path Following Control Strategy of Autonomous Electric Vehicles with Yaw Motion Stabilization," *IEEE Transactions on Transportation Electrification,* 2020.

[28] X. Yuan, J. Wang, and K. Colombage, "Torque distribution strategy for a front and rear wheel driven electric vehicle," 2012.

[29] X. Zhang, D. Göhlich, and J. Li, "Energy-efficient toque allocation design of traction and regenerative braking for distributed drive electric vehicles," *IEEE Transactions on Vehicular Technology,* vol. 67, no. 1, pp. 285-295, 2017.

[30] H. Sun, H. Wang, and X. Zhao, "Line braking torque allocation scheme for minimal braking loss of four-wheel-drive electric vehicles," *IEEE Transactions on Vehicular Technology,* vol. 68, no. 1, pp. 180-192, 2018.

[31] L. Li, Y. Zhang, C. Yang, B. Yan, and C. M. Martinez, "Model predictive control-based efficient energy recovery control strategy for regenerative braking system of hybrid electric bus," *Energy conversion and management,* vol. 111, pp. 299-314, 2016.

[32] W. Xu, H. Chen, H. Zhao, and B. Ren, "Torque optimization control for electric vehicles with four in-wheel motors

equipped with regenerative braking system," *Mechatronics,* vol. 57, pp. 95-108, 2019.

[33] Z. Han, N. Xu, H. Chen, Y. Huang, and B. Zhao, "Energy-efficient control of electric vehicles based on linear quadratic regulator and phase plane analysis," *Applied Energy,* vol. 213, pp. 639-657, 2018.

[34] X. Hu, P. Wang, Y. Hu, and H. Chen, "A stability-guaranteed and energy-conserving torque distribution strategy for electric vehicles under extreme conditions," *Applied Energy,* vol. 259, p. 114162, 2020.

[35] Y. Ma, J. Chen, X. Zhu, and Y. Xu, "Lateral stability integrated with energy efficiency control for electric vehicles," *Mechanical Systems and Signal Processing,* vol. 127, pp. 1-15, 2019/07/15/ 2019, doi: https://doi.org/10.1016/j.ymssp.2019.02.057.

[36] H. Peng, W. Wang, Q. An, C. Xiang, and L. Li, "Path Tracking and Direct Yaw Moment Coordinated Control Based on Robust MPC with the Finite Time Horizon for Autonomous Independent-Drive Vehicles," *IEEE Transactions on Vehicular Technology,* 2020.

[37] Y. Zou, T. Liu, D. Liu, and F. Sun, "Reinforcement learning-based real-time energy management for a hybrid tracked vehicle," *Applied Energy,* vol. 171, pp. 372-382, 2016/06/01/ 2016, doi: https://doi.org/10.1016/j.apenergy.2016.03.082.

[38] Q. Zhou *et al.*, "Multi-step reinforcement learning for model-free predictive energy management of an electrified off-highway vehicle," *Applied Energy,* vol. 255, p. 113755, 2019/12/01/ 2019, doi: https://doi.org/10.1016/j.apenergy.2019.113755.

[39] M. Srouji, J. Zhang, and R. Salakhutdinov, "Structured control nets for deep reinforcement learning," *arXiv preprint arXiv:1802.08311,* 2018.

[40] J. Wu, H. He, J. Peng, Y. Li, and Z. Li, "Continuous reinforcement learning of energy management with deep Q network for a power split hybrid electric bus," *Applied Energy,* vol. 222, pp. 799-811, 2018/07/15/ 2018, doi: https://doi.org/10.1016/j.apenergy.2018.03.104.

[41] Y. Wu, H. Tan, J. Peng, H. Zhang, and H. He, "Deep reinforcement learning of energy management with continuous control strategy and traffic information for a series-parallel plug-in hybrid electric bus," *Applied Energy,* vol. 247, pp. 454-466, 2019/08/01/ 2019, doi: https://doi.org/10.1016/j.apenergy.2019.04.021.

[42] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[43] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[44] C. J. Watkins and P. Dayan, "Q-learning," *Machine learning,* vol. 8, no. 3-4, pp. 279-292, 1992.

[45] V. Mnih *et al.*, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602,* 2013.

[46] R. Bellman, "On the theory of dynamic programming," *Proceedings of the National Academy of Sciences of the United States of America,* vol. 38, no. 8, p. 716, 1952.

[47] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *nature,* vol. 518, no. 7540, pp. 529-533, 2015.

[48] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971,* 2015.

[49] R. S. Sutton, D. A. McAllester, S. P. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Advances in neural information processing systems*, 2000, pp. 1057-1063.

[50] R. M. Karp, M. Luby, and N. Madras, "Monte-Carlo approximation algorithms for enumeration problems," *Journal of algorithms,* vol. 10, no. 3, pp. 429-448, 1989.

[51] S. Fujimoto, H. v. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," *ArXiv,* vol. abs/1802.09477, 2018.

[52] M. Mirzaei, "A new strategy for minimum usage of external yaw moment in vehicle dynamic control system," *Transportation Research Part C: Emerging Technologies,* vol. 18, no. 2, pp. 213-224, 2010.

[53] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Transactions on Vehicular Technology,* vol. 66, no. 2, pp.

952-964, 2016.