machines

MDPI

Article

Deep Reinforcement Learning-Based Torque Vectoring Control Considering Economy and Safety

Huifan Deng, Youqun Zhao *, Fen Lin and Qiuwei Wang

College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China; dhf@nuaa.edu.cn (H.D.); flin@nuaa.edu.cn (F.L.); richardwang@nuaa.edu.cn (Q.W.)
* Correspondence: yqzhao@nuaa.edu.cn

Abstract: This paper presents a novel torque vectoring control (TVC) method for four in-wheel-motor independent-drive electric vehicles that considers both energy-saving and safety performance using deep reinforcement learning (RL). Firstly, the tire model is identified using the Fibonacci tree optimization algorithm, and a hierarchical torque vectoring control scheme is designed based on a nonlinear seven-degree-of-freedom vehicle model. This control structure comprises an active safety control layer and a torque allocation layer based on RL. The active safety control layer provides a torque reference for the torque allocation layer to allocate torque while considering both energy-saving and safety performance. Specifically, a new heuristic random ensembled double Q-learning RL algorithm is proposed to calculate the optimal torque allocation for all driving conditions. Finally, numerical experiments are conducted under different driving conditions to validate the effectiveness of the proposed TVC method. Through comparative studies, we emphasize that the novel TVC method outperforms many existing related control results in improving vehicle safety and energy savings, as well as reducing driver workload.

Keywords: torque vectoring control; coordinated control; electric ground vehicles; reinforcement learning (RL); vehicle motion dynamics

check for updates

Citation: Deng, H.; Zhao, Y.; Lin, F.; Wang, Q. Deep Reinforcement Learning-Based Torque Vectoring Control Considering Economy and Safety. *Machines* **2023**, *11*, 459. https://doi.org/10.3390/machines11040459

Academic Editor: Zheng Chen

Received: 28 February 2023
Revised: 28 March 2023
Accepted: 3 April 2023
Published: 6 April 2023

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

1. Introduction

The automotive industry has witnessed significant advancements in electric vehicle (EV) technology over the past decade [1,2]. EVs are gaining popularity due to their environmental friendliness and low operating costs. Conventional EVs use a centralized drive system, which transfers the power from the battery to the wheels through a transmission system. However, this system has several limitations such as limited efficiency and reduced stability and control [3]. The four-in-wheel-motor independent-drive electric vehicle (4MIDEV), which places an electric motor in each wheel of the vehicle, has emerged as a promising solution to overcome these limitations [4].

Torque vectoring control (TVC) refers to the ability to distribute the torque to each wheel of the vehicle independently. The torque allocation can be controlled by adjusting the electric motor output torque to each wheel, which is achieved by using an electronic control unit. Several factors affect TVC, including vehicle speed, acceleration, road conditions, and motor efficiency [5]. Hence, it is important to research the TVC of 4MIDEV considering both economy and safety according to the current vehicle state [6].

The TVC is usually classified into holistic and hierarchical structures according to the control framework. The holistic TVC structure typically employs model predictive control (MPC) [7]. However, this approach suffers from certain drawbacks, including a large system size, substantial computational effort, and implementation challenges [8]. Therefore, in practical applications, TVC generally adopts a hierarchical control structure. The active safety control layer generates reference control quantities based on the reference model, while the torque allocation layer generates control commands for each actuator, such as tire longitudinal force and torque.

Many scholars use sliding mode controllers (SMC) to solve the torque allocation problem. Reference [9] presents an SMC with its stability being proven using a Lyapunov function. The Lyapunov control method proposed in the article overcomes the limitations of SMC by improving control accuracy and dynamic tracking performance while effectively reducing chattering and abrupt changes in the control system. In reference [10], a new SMC law is designed to reduce chattering and make state variables converge faster. Reference [11] presents an adaptive and robust controller that combines the driving characteristics of professional drivers to improve vehicle stability and maneuverability and relieve the driver's workload. Experimental results demonstrate the effectiveness of the proposed controller in reducing the peak steering angle, reducing the driver's operating load, and addressing the chattering issue in conventional SMC. Reference [12] presents a coordination control strategy for the vehicle stability control system and differential drive-assisted steering, which considers tire sideslip and aims to improve vehicle stability in different driving conditions and presents CarSim simulations and vehicle experiments verifying its effectiveness. Moreover, some researchers have focused on developing fault-tolerant control strategies that can ensure the safety and performance of the 4MIDEV even in the presence of actuator faults. Reference [13] proposes a fault-tolerant control method for 4MIDEV that considers both vehicle safety and motor power consumption.

However, traditional TVC methods often require complex and computationally expensive control algorithms, which limit their effectiveness. To overcome these challenges, reinforcement learning (RL) offers a machine learning approach that enables an agent to learn how to make optimal decisions by interacting with an environment and receiving rewards as feedback [14]. RL has shown significant potential for improving control performance while reducing complexity in many areas, such as robot control [15] and autonomous driving [16]. Reference [17] proposed a direct torque allocation algorithm that employs a deep RL technique to enhance the safety and fuel economy of the vehicle. This paper uses the integrated control framework to train the RL directly. However, the vehicle's dynamics model is complex, making it difficult to apply RL directly. Additionally, the RL is usually applied to the problem in which objects are difficult to model, or the model is too complex. For stability control, the active safety control layer already has a detailed dynamic model; the required additional yaw moment and longitudinal force can be well solved using model-based control methods such as optimal control or sliding mode control. Therefore, it is not appropriate to use direct RL control in the active safety control layer. Motivated by the above issues, a novel TVC method based on RL for 4MIDEV is proposed in this paper. The contributions of this study are summarized as follows:

- Unlike the reference [18], this paper proposes a TVC method that takes into account both economy and safety. Specifically, the torque allocation layer based on deep RL adaptively adjusts the torque of each wheel according to the current vehicle state.
- An improved heuristic randomized ensembled double Q-learning (REDQ) algorithm is introduced for EV control, which reduces the training complexity of RL compared to existing RL algorithms for direct motor torque control.
- The Fibonacci tree-based tire model identification method is employed, which achieves higher identification accuracy than the genetic algorithm (GA) [19] and the particle swarm optimization (PSO) algorithm [20].

2. The TVC Framework and System Model

2.1. The TVC Framework

Figure 1 illustrates the block diagrams of the TVC system, which is composed of four modules: tire model identification, the vehicle reference model, the active safety control layer, and the RL-based torque allocation layer. The tire mathematical model is obtained through experimental data based on the Fibonacci tree optimization (FTO) algorithm. The active safety control layer is designed to generate longitudinal force F_x and additional yaw moment M_z. The torque allocation layer employs the heuristic REDQ algorithm with integrated consideration of both economy and safety to distribute the four-wheel torque.

Moreover, the average allocation method gives the RL algorithm heuristic training and reduces its training time.

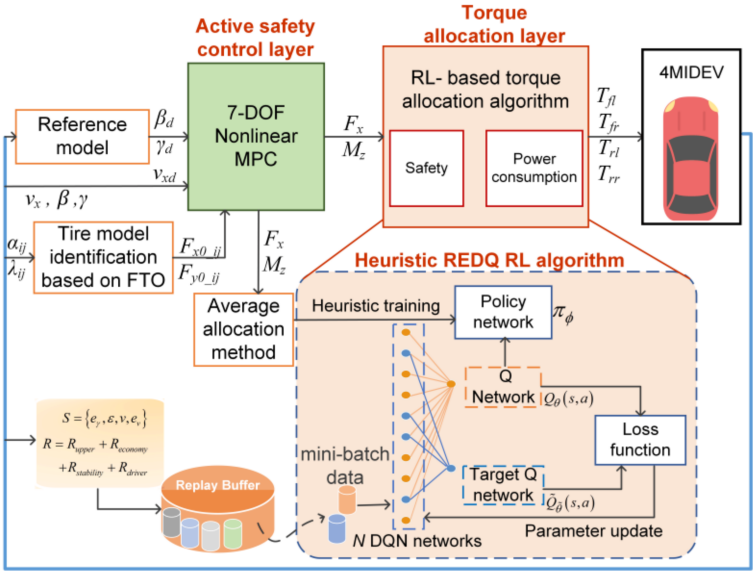

Figure 1. The block diagrams of the TVC system.

2.2. Tire Model Identification

The tire is a complex system that interacts with the road surface and the vehicle's suspension system. The tire's behavior can significantly affect the vehicle's dynamics, including acceleration, braking, cornering, and ride comfort. Therefore, accurate modeling of the tire is essential for vehicle control, design, and optimization. A good tire model can provide reliable predictions of the tire forces and moments, which are critical inputs to vehicle dynamic control algorithms. Additionally, tire models can help understand the effects of different tire designs and operating conditions on vehicle performance and handling.

The magic formula tire model is a semi-empirical tire model that has become one of the most widely used tire models in vehicle dynamics and control research. Additionally, the model includes several parameters that can be tuned to match experimental data or represent different tire designs. According to the magic formula, the longitudinal force F^0_{xij}, and lateral force F^0_{yij} of the tire model are calculated by:

$$\begin{aligned} F^0_{xij} &= D_x \sin[C_x \arctan(B_x \phi_{xij})] + S_{vx} \\ \phi_{xij} &= (1 - E_x)(\lambda_{ij}) + (E_x / B_x)\arctan(B_x(\lambda_{ij})) \\ F^0_{yij} &= D_y \sin[C_y \arctan(B_y \phi_{yij})] + S_{vy} \\ \phi_{yij} &= (1 - E_y)(\alpha_{ij}) + (E_y / B_y)\arctan(B_y(\alpha_{ij})) \end{aligned} \tag{1}$$

where F_z is the vertical load of the tire and λ_{ij} and α_{ij} tire slip angle or tire longitudinal slip ratio, respectively. $ij \in \{fl, fr, rl, rr\}$. $B_x = BCD_x/(C_x \cdot D_x)$, $C_x = b_0$, $D_x = b_1 F_z^2 + b_2 F_z$, $BCD_x = (b_3 F_z^2 + b_4 F) e^{-b_5 F_z}$, $E_x = b_6 F_z^2 + b_7 F_z + b_8$, $B_y = BCD_y/(C_y \cdot D_y)$, $C_y = a_0$, $D_y = a_1 F_z^2 + a_2 F_z$, $BCD_y = a_3 \sin(2\arctan(F_z/a_4))$, $E_y = a_6 F_z + a_7$.

The raw data for the FTO tire identification algorithm is obtained through a comprehensive tire mechanics test bench. The powertrain drives the simulated road surface at a translational speed of 0.3 m/s from one side of the test bench to the other. The three-dimensional force sensors of the test bench are used to measure the lateral and longitudinal forces. The friction coefficient is 0.8, the slip ratio is 0–0.7, and the lateral slip angle is 0–20 degrees. The test was conducted at three vertical loads: 10 kN, 15 kN, and 20 kN, and the experimental data are shown in Figure 2. Each point in Figure 2 represents a test result. The FTO algorithm is a computational optimization algorithm based on the golden section

method and the Fibonacci optimization principle. It optimizes the global search and local search alternately and iteratively constructs the Fibonacci tree structure to find the best solution. The FTO algorithm consists of two stages for each search: a global search and a local search.

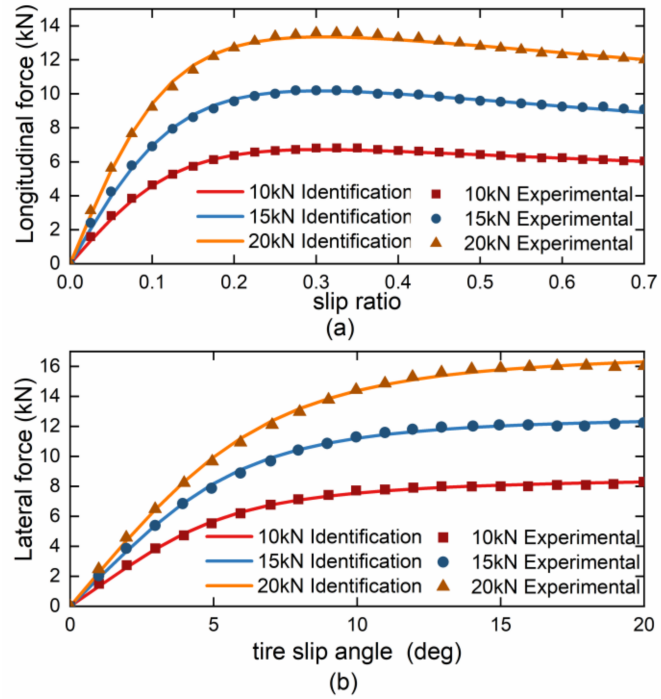

Figure 2. Tire model identification results. (**a**) Longitudinal force recognition results. (**b**) Lateral force recognition results.

In the global search stage, global nodes N_i are randomly generated within the global scope. According to the fitness of each node, global trial nodes $W_a(a = 1, 2, \cdots, F_i)$ are generated as follows:

$$W_a = \begin{cases} N_i + \frac{F_{i-1}}{F_i} \cdot (B_{ij} - N_i), \text{ if } Fit_{best}(N_i, B_{ij}) = N_i \\ B_{ij} + \frac{F_{i-1}}{F_i} \cdot (N_i - B_{ij}), \text{ otherwise} \end{cases} \tag{2}$$

F_i is the Fibonacci series, and the general formula is as follows:

$$F_i = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^i - \left(\frac{1-\sqrt{5}}{2} \right)^i \right] \tag{3}$$

where $Fit_{best}(N_i, B_{ij})$ represents the better adaptability of N_i and B_{ij}, which is calculated according to the objective functions (6) and (7).

In the local search stage, the local trial nodes V_b are generated according to the best node of the current node B_{i1} and the current node B_{ij}:

$$\begin{aligned} B_{i1} &= Fit_{best}(B_{ij}) \\ V_b &= B_{i1} + \frac{F_{i-1}}{F_i}(B_{ij} - B_{i1}) \end{aligned} \tag{4}$$

Finally, W_a, V_b, and the current node B_{ij} are sorted according to fitness, and the best $F_{(i+1)}$ nodes are retained as the next generation nodes $B_{(i+1)j}$:

$$B_{(i+1),j} = Fit_{best}(W_a, V_b, B_{i,j}) \tag{5}$$

The algorithm is expressed as Algorithm 1.

Algorithm 1. FTO algorithm

1. Set the depth of Fibonacci tree N and the number of identification parameters n;
2. Randomly generate an initial node B_{11}: randomly generate a global random node N_1;
3. **Repeat:**
4. Generate F_i global trial nodes W_1~W_{Fi} according to the global random node N_i and the node B_{ij};
5. Generate F_{i-1} local trial nodes V_1~V_{Fi-1} according to the best adaptable element B_{i1} in the current node and the remaining nodes;
6. Get the next generation node B_{i+1j};
7. Update the node set. Incorporate the newly generated trial nodes into the current node set S, calculate the fitness function and sort, and retain the first F_{i+1} nodes;
8. **Until** $F_{i+1} \geq F_N$
9. Output the optimal node;

The objective function for first-level parameter identification is as follows:

$$\begin{aligned} f1_x &= \sum_{i=1}^{N_x} \{D_x \sin(C_x \text{atan}(B_x\lambda_i - E_x(B_x - \arctan(B_x\lambda_i)))) - F_{xi}^*\}^2 \\ f1_y &= \sum_{i=1}^{N_y} \{D_y \sin(C_y \text{atan}(B_y\alpha_i - E(B_y - \arctan(B_y\alpha_i)))) - F_{yi}^*\}^2 \end{aligned} \tag{6}$$

where, $F_{xi}{}^*$, F_{yi}, ${}^*\lambda_i$, and α_i are the experimental data of longitudinal force, lateral force, slip rate, and side slip angle under different vertical loads, respectively. N_x = 29 and N_y =20 represent the number of tests.

The objective function for second-level parameter identification is as follows:

$$\begin{aligned} f2_{x1} &= \sum_{j=1}^{M} (b_0 - C_{xj}^*)^2 & f2_{y1} &= \sum_{j=1}^{M} (a_0 - C_{yj}^*)^2 \\ f2_{x2} &= \sum_{j=1}^{M} [(b_1F_{zj}^2 + b_2F_{zj}) - D_{xj}^*]^2 & f2_{y2} &= \sum_{j=1}^{M} [(a_1F_{zj}^2 + a_2F_{zj}) - D_{yj}^*]^2 \\ f2_{x3} &= \sum_{j=1}^{M} [(b_3F_{zj}^2 + b_4F_{zj})e^{-b_5F_z} - BCD_{xj}^*]^2 & f2_{y3} &= \sum_{j=1}^{M} [a_3 \sin(2\arctan(F_{zj}/a_4)) - BCD_{yj}^*]^2 \\ f2_{x4} &= \sum_{j=1}^{M} [(b_6F_{zj}^2 + b_7F_{zj} + b_8) - E_{xj}^*]^2 & f2_{y4} &= \sum_{j=1}^{M} [(a_6F_{zj} + a_7) - E_{yj}^*]^2 \\ BCD_{xj}^* &= B_{xj}^* \cdot C_{xj}^* \cdot D_{xj}^* & BCD_{yj}^* &= B_{yj}^* \cdot C_{yj}^* \cdot D_{yj}^* \end{aligned} \tag{7}$$

where B_{xj}^*, C_{xj}^*, D_{xj}^*, E_{xj}^*, B_{yj}^*, C_{yj}^*, D_{yj}^*, E_{yj}^* are the first-level parameter values identified by the vertical load applied by group j. After classifying the parameters of the tire model into first-level identification parameters and second-level identification parameters, the hierarchical identification approach is found to not only ensure the accuracy of the tire mathematical model but also improve its identification efficiency compared to the non-hierarchical approach.

The recognition result is shown in Figure 2. To further investigate the performance of the FTO algorithm, a comparative analysis with the commonly used GA and PSO optimization algorithms was conducted, and the results are presented in Table 1. The algorithm of GA is from [19], and the algorithm of PSO is from [20]. The depth of the Fibonacci tree is set to 7, while the parameters of the GA are adopted from [19]. The generation gap is set to 0.6, the population size is set to 50, and the crossover probability and mutation probability are set to 0.9 and 0.0097, respectively. Similarly, the parameters of the PSO algorithm are selected from [20], with the population size set to 40 and the inertia weight set to 0.7298, constant $c_1 = 2$, and constant $c_2 = 2$.

Table 1. First-level parameters identification results.

Item	10 kN	15 kN	20 kN
FTO relative residual of longitudinal force	1.40%	1.37%	1.40%
GA relative residual of longitudinal force	4.61%	2.21%	1.73%
PSO relative residual of longitudinal force	3.44%	1.74%	1.66%
FTO relative residual of lateral force	1.48%	1.37%	1.39%
GA relative residual of lateral force	1.78%	1.63%	1.48%
PSO relative residual of lateral force	1.65%	1.69%	1.56%

The relative residual e is expressed as:

$$e = \sqrt{\sum \left(F^{MF} - F^*\right)^2 / \sum \left(F^*\right)^2} \times 100\% \tag{8}$$

where F^{MF} is the tire force value identified by the FTO algorithm, and F^* is the experimental value of the tire force.

Based on the results presented in Table 1, it is observed that the FTO algorithm outperforms the conventional GA and PSO algorithms in terms of parameter recognition accuracy and relative residual error. The first-level parameter identification has a global relative residual of 1.48%. Moreover, for complex parameter identification problems, it is beneficial to improve the efficiency of parameter identification by classifying the parameters and reducing the number of independent variables of the single identification objective function. After obtaining the identified parameters, we can obtain the mathematical model of the tire, and the longitudinal and lateral forces of the tire can be calculated by substituting different vertical loads.

2.3. Vehicle Reference Model

In vehicle control, achieving both good tracking performance and disturbance rejection is critical for safe and efficient operation. A 2-degree-of-freedom (DOF) reference model is a popular choice for designing controllers that provide these performance objectives. The 2-DOF vehicle model is expressed as:

$$\begin{aligned} mv_x\left(\dot{\beta} + \gamma\right) &= 2C_f\left(\delta_f - \beta - a\gamma/v_x\right) + 2C_r(-\beta + b\gamma/v_x) \\ I_z\dot{\gamma} &= 2aC_f\left(\delta_f - \beta - a/v_x\right) - 2bC_r(-\beta + b/v_x) \end{aligned} \tag{9}$$

where m is the total mass of the vehicle, v_x is the longitudinal velocity, β is the side slip angle of the vehicle center of gravity (CG), γ is the yaw rate, and C_f and C_r are the front and rear tire cornering stiffness, respectively. a and b are the distance from the front and rear wheel axles to the CG, respectively. δ_f is the steering angle of the front wheels, and I_z is the yaw mass moment of inertia.

Applying the Laplace transform to Equation (9) we can obtain the following transfer function:

$$\frac{\gamma(s)}{\delta_f(s)} = G_\gamma \frac{\omega_n^2(\xi_1 s + 1)}{s^2 + 2\omega_n\xi_2 \cdot s + \omega_n^2} \tag{10}$$

where $\omega_n = \frac{2(a+b)}{v_x}\sqrt{\frac{C_fC_r\left(1+Kv_x^2\right)}{mI_z}}$, $G_\gamma = \frac{v_x}{(a+b)\left(1+Kv_x^2\right)}$, $K = -\frac{m\left(aC_f - bC_r\right)}{2C_fC_r(a+b)^2}$, $\xi_1 = \frac{mv_xa}{2C_r(a+b)}$, $\xi_2 = \frac{m\left(C_fa^2 + C_rb^2\right) + I_z\left(C_f + C_r\right)}{2(a+b)\sqrt{mI_zC_fC_r\left(1+Kv_x^2\right)}}$.

The steady-state value of the yaw rate γ^* can be obtained from Equation (10) as:

$$\gamma^* = G_\gamma\delta_f \tag{11}$$

Considering the road adhesion coefficient, γ_d is limited by the following equation:

$$a_y = \gamma_d v_x \leq \mu g \tag{12}$$

where μ is the road adhesion coefficient, and g is the acceleration of gravity.

Then, the reference yaw rate is expressed as

$$\gamma_d = \min\left\{ |\gamma^*|, \left| \frac{\mu g}{v_x} \right| \right\} \cdot \mathrm{sgn}\left(\delta_f\right) \tag{13}$$

We note that vehicle safety is guaranteed when the sideslip angle β varies within a small range for the normal operating condition of the vehicle. Following related works on vehicle dynamics control [21,22], we conservatively set the desired sideslip angle $\beta_d = 0$ to maintain vehicle safety under extreme operating conditions.

2.4. Vehicle 7-DOF Dynamic Model

Vehicle models are used to predict vehicles' behavior under different driving conditions and to design control systems that achieve desired performance objectives. Briefly, 2-DOF vehicle models are widely used due to their simplicity and ease of use, but they have limitations in representing vehicles' motion under complex driving conditions. Additionally, nonlinear 7-DOF models of vehicles provide a more accurate representation of vehicles' motion and are essential for advanced control system design, simulation, and testing [23]. Therefore, a nonlinear 7-DOF vehicle dynamics model is introduced. The vehicle dynamics model, as illustrated in Figure 3, serves as a foundation for the TVC in this research.

$$\begin{aligned}
m\dot{v}_x &= F_{xrl} + F_{xrr} + \left(F_{xfl} + F_{xfr}\right)\cos\delta_f - \left(F_{yfl} + F_{yfr}\right)\sin\delta_f + mv_y\gamma \\
m\dot{v}_y &= F_{yrl} + F_{yrr} + \left(F_{xfl} + F_{xfr}\right)\sin\delta_f + \left(F_{yfl} + F_{yfr}\right)\cos\delta_f + mv_x\gamma \\
I_z\dot{\gamma} &= a\left(F_{yfl} + F_{yfr}\right)\cos\delta_f - b\left(F_{yrl} - F_{yrr}\right) + \tfrac{d_w}{2}\left(F_{yfl} - F_{yfr}\right)\sin\delta_f + M_z
\end{aligned} \tag{14}$$

$$I_{\omega ij}\dot{\omega}_{ij} = -F_{xij}R_\omega + T_{tij} \tag{15}$$

$$\begin{aligned}
M_z &= F_{xfl}\left(a\sin\delta_f - d_w\cos\delta_f/2\right) \\
&+F_{xfr}\left(a\sin\delta_f + d_w\cos\delta_f/2\right) + d_w/2(F_{xrr} - F_{xrl})
\end{aligned} \tag{16}$$

where v_y is the vehicle longitudinal velocity. d_w is the track width. ω_{ij} is the wheel rotation rate, I_{ij} is the wheel moment of inertia, and T_{ij} is the output torque of the in-wheel motor.

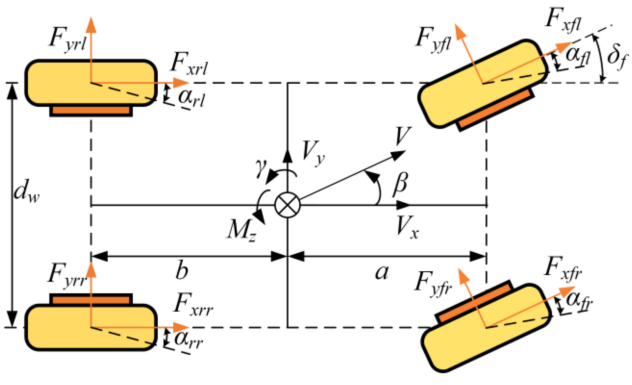

Figure 3. Vehicle 7-DOF dynamic model diagram.

The tire slip angle α_{ij} is expressed as

$$\begin{aligned} \alpha_{fl} &= \arctan\left(\frac{v_y+a\gamma}{v_x-\frac{1}{2}d_w\gamma}\right)-\delta_f,\ \alpha_{rl}=\arctan\left(\frac{v_y-b\gamma}{v_x-\frac{1}{2}d_w\gamma}\right) \\ \alpha_{fr} &= \arctan\left(\frac{v_y+a\gamma}{v_x+\frac{1}{2}d_w\gamma}\right)-\delta_f,\ \alpha_{rr}=\arctan\left(\frac{v_y-b\gamma}{v_x+\frac{1}{2}d_w\gamma}\right) \end{aligned} \tag{17}$$

The tire slip ratio λ_{ij} is calculated by

$$\lambda_{ij} = \left(v_{ij}-\omega_{\mathrm{ij}}R_\omega\right)/v_{ij} \tag{18}$$

The expressions of the four-wheel speeds are given as

$$\begin{aligned} v_{fl} &= (v_y+a\gamma)\sin\delta_f+(v_x-\gamma d_w/2)\cos\delta_f,\ v_{rl}=v_x-\gamma d_w/2 \\ v_{fr} &= (v_y+a\gamma)\sin\delta_f+(v_x+\gamma d_w/2)\cos\delta_f,\ v_{rr}=v_x+\gamma d_w/2 \end{aligned} \tag{19}$$

The tire force F_{zij} is expressed as

$$\begin{aligned} F_{zfl} &= \tfrac{m}{a+b}\left(\tfrac{gb}{2}-\tfrac{a_xh_g}{2}-\tfrac{a_yh_gb}{d_w}\right) \\ F_{zfr} &= \tfrac{m}{a+b}\left(\tfrac{gb}{2}-\tfrac{a_xh_g}{2}+\tfrac{a_yh_gb}{d_w}\right) \\ F_{zrl} &= \tfrac{m}{a+b}\left(\tfrac{ga}{2}+\tfrac{a_xh_g}{2}-\tfrac{a_yh_ga}{d_w}\right) \\ F_{zrr} &= \tfrac{m}{a+b}\left(\tfrac{ga}{2}+\tfrac{a_xh_g}{2}+\tfrac{a_yh_ga}{d_w}\right) \end{aligned} \tag{20}$$

where h_g is the height of the center of gravity.

As the tire force needs to meet the attachment ellipse, F_{xij} and F_{yij} are expressed as:

$$F_{xij} = F_{x0ij}\psi_{xij}/\psi_{ij} \quad F_{yij} = F_{y0ij}\psi_{xij}/\psi_{ij} \tag{21}$$

where $\psi_{ij}=\sqrt{\psi_{xij}^2+\psi_{yij}^2}, \psi_{xij}=-\frac{\lambda_{ij}}{1+\lambda_{ij}}, \psi_{yij}=-\frac{\tan(\alpha_{ij})}{1+\lambda_{ij}}$. F_{x0ij} and F_{y0ij} are calculated by Equation (1).

3. The TVC Algorithm

This section describes the details of the RL-based TVC algorithm, which includes an active safety control layer and an RL-based torque allocation layer.

3.1. Active Safety Control Layer

The role of the active safety control layer is to calculate the total longitudinal force F_x and additional yaw moment M_z necessary to ensure vehicle stability, which will serve as reference values for the torque allocation layer. If the calculated final torque of each wheel by the torque allocation layer can achieve the total longitudinal force and additional yaw moment reference values, the stability of the vehicle can be ensured.

According to Equations (14)–(20), we can obtain the following nonlinear continuous system.

$$\begin{aligned} \dot{x}(t) &= f(x(t),u(t)) \\ y(t) &= C\cdot x(t) \end{aligned} \tag{22}$$

where $x(t) = \left[v_x(t), v_y(t), \gamma(t)\right]^{\mathrm{T}}$, $y(t) = [v_x(t), \beta(t), \gamma(t)]^{\mathrm{T}}$, $u = [F_x(t), M_z(t)]^{\mathrm{T}}$, $C=\begin{bmatrix}1 & 0 & 0\\ 0 & 1/v_x(t) & 0\\ 0 & 0 & 1\end{bmatrix}$.

Discretizing the Equation (22), we can obtain the difference equation at time k:

$$\begin{aligned}
v_x(k+1) &= T_s\left(\tfrac{1}{m}\left(F_{xrl}+F_{xrr}+\left(F_{xfl}+F_{xfr}\right)\cos\delta_f-\left(F_{yfl}+F_{yfr}\right)\sin\delta_f\right)+v_y(k)\gamma(k)\right)+v_x(k)\\
v_y(k+1) &= T_s\left(\tfrac{1}{m}\left(F_{yrl}+F_{yrr}+\left(F_{xfl}+F_{xfr}\right)\sin\delta_f+\left(F_{yfl}+F_{yfr}\right)\cos\delta_f\right)+v_x(k)\gamma(k)\right)+v_y(k)\\
\gamma(k+1) &= T_s\left(\tfrac{1}{I_z}\left(a\left(F_{yfl}+F_{yfr}\right)\cos\delta_f-b\left(F_{yrl}-F_{yrr}\right)+\tfrac{d_w}{2}\left(F_{yfl}-F_{yfr}\right)\sin\delta_f+M_z\right)\right)+\gamma(k)\\
y(k) &= C\cdot x(k)
\end{aligned} \tag{23}$$

With the active safety control layer defined in Equation (23), we can calculate the M_z and F_x to ensure vehicle safety.

Nonlinear MPC involves the optimization of a nonlinear objective function, which is subject to constraints on the system dynamics and control inputs. The nonlinear MPC controller for the system described in Equation (23) can be designed by formulating an optimal control problem:

$$\min_{X,U}\sum_{k=0}^{Np-1}\left[(Y-Y_{ref})^T W_Q(Y-Y_{ref})+U^T W_R U+\Delta U^T W_S \Delta U\right] \tag{24}$$

$$\text{s.t.}\quad x_0 = x_{in} \tag{25}$$

$$\begin{aligned} x_{k+1} &= f_d(x_k,u_k)\\ y(k) &= C\cdot x(k)\end{aligned} \tag{26}$$

$$x_{\min}\le x_k\le x_{\max} \tag{27}$$

$$u_{\min}\le u_k\le u_{\max} \tag{28}$$

where $Y(k)=\left[y(k),y(k+1),\cdots,y\left(k+N_p\right)\right]^{\mathrm{T}}$ and $U(k)=[u(k),u(k+1),\cdots,u(k+N_c)]^{\mathrm{T}}$ are the N_p step state vector and the N_c step input vector, respectively. $\Delta u = u_k - u_{k-1}$. Additionally, $Y_{ref}(k) = \left[y_{ref}(k),y_{ref}(k+1),\cdots,y_{ref}\left(k+N_p\right)\right]^{\mathrm{T}}$ where $y_{ref}(k)=[v_{xd}(k),\beta_d(k),\gamma_d(k)]$ is the reference of the state at step k and f_d is the difference equation in Equation (23). W_Q, W_R, W_S are the weighting matrices.

The goal of optimal control is to minimize the objective function given by Equation (24), subject to constraints such as initial conditions Equation (25), discrete nonlinear system dynamics Equation (26), state constraints Equation (27), and control constraints Equation (28). The objective function Equation (24) comprises three main components, namely, the minimization of state-reference trajectory error, system input, and input variation.

To solve the nonlinear programming problem represented by Equations (23)–(27), the widely used sequential quadratic programming algorithm [24] can be employed. Once the optimal input vector is obtained, the first control input $u(k)$ is then applied to the system, and the process is repeated at the next time step.

3.2. Torque Allocation Layer

Based on Equations (14) and (15), the vehicle longitudinal force F_x and the additional yaw moment M_z are expressed as:

$$F_x = (T_{fl}\cos\delta_f + T_{fr}\cos\delta_f + T_{rl} + T_{rr})/R_\omega \tag{29}$$

$$\begin{aligned} M_z = {} & \tfrac{1}{R_\omega}((-\tfrac{d_w}{2}\cos\delta_f + a\sin\delta_f)T_{fl}\\ & +(\tfrac{d_w}{2}\cos\delta_f + a\sin\delta_f)T_{fr} - \tfrac{d_w}{2}T_{rl} + \tfrac{d_w}{2}T_{rr})\end{aligned} \tag{30}$$

To ensure the stability of the vehicle, the four-wheel torque needs to satisfy Equations (29) and (30). The aim of the torque allocation layer is to distribute the four-

wheel torque to achieve M_z and F_x. For 4MIDEV, it is a typical over-actuated system with more actuators than degrees of freedom of the system. Therefore, torque allocation is a problem worth researching; it is important to reduce the power consumption of the motor while ensuring safety.

In the torque allocation layer, we considered a combination of safety and power consumption. In particular, the power consumption and safety weights are dynamically adjusted according to the current vehicle state based on RL. To this end, the task of this paper is the torque allocation of the 4MIDEV, which guarantees the safety and power consumption of the EV.

3.2.1. Average Allocation Method

As presented in references [18,25], the most common method of torque allocation is the average allocation method. The average allocation algorithm here serves two purposes. The first is as a base method for comparison. The second is as a heuristic training method for RL in the early stages of training. We have stated this in the revised manuscript. The average allocation methods are shown in the following [26]:

$$\begin{aligned} T_{fl} &= \tfrac{1}{4}\tfrac{F_x R_\omega}{\cos\delta_f} + \tfrac{1}{4}\tfrac{M_z R_\omega}{-\frac{1}{2}d_w\cos\delta_f + a\sin\delta_f} \\ T_{fr} &= \tfrac{1}{4}\tfrac{F_x R_\omega}{\cos\delta_f} + \tfrac{1}{4}\tfrac{M_z R_\omega}{\frac{1}{2}d_w\cos\delta_f + a\sin\delta_f} \\ T_{rl} &= \tfrac{1}{4}F_x R_\omega - \tfrac{1}{4}\tfrac{M_z R_\omega}{\frac{1}{2}d_w} \\ T_{rr} &= \tfrac{1}{4}F_x R_\omega + \tfrac{1}{4}\tfrac{M_z R_\omega}{\frac{1}{2}d_w} \end{aligned} \tag{31}$$

3.2.2. RL-Based Torque Allocation Algorithm

This section describes the details of the RL-based torque allocation algorithm. The state plays a crucial role in the RL algorithm, and in this study, we define the state space using four states: the yaw rate of deviation $e_\gamma = \gamma - \gamma_{des}$, the stability indicator ε, velocity v, and velocity deviation e_v. Additionally, the action is defined as four-wheel torque.

$$S = \{e_\gamma, \varepsilon, v, e_v\}, A = \left\{T_{fl}, T_{fr}, T_{rl}, T_{rr}\right\} \tag{32}$$

where the ε is obtained from the phase plane. In vehicle dynamics, the driving stability and instability regions can be depicted using a phase portrait [27]. The ε is defined as follows.

$$\varepsilon = \left|\beta/B_2 + \dot{\beta}B_1/B_2\right| \tag{33}$$

where B_1 and B_2 are the parameters associated with the adhesion coefficient. Their corresponding values are specified in reference [28]. This paper aims to improve the economy of EVs while ensuring safety, so the reward function R is expressed as the following four parts:

Firstly, Equations (29) and (30) ensure that the four-wheel torque satisfies the additional yaw moment M_z and longitudinal force F_x from the active safety control layer. Therefore, to ensure vehicle stability, the torque allocation layer needs to satisfy the following equation:

$$A_s u_c = B_u \tag{34}$$

where $A_s = \frac{1}{R}\begin{bmatrix} \cos\delta_f & \cos\delta_f & 1 & 1 \\ -\frac{l_w}{2}\cos\delta_f + l_f\sin\delta_f & \frac{l_w}{2}\cos\delta_f + l_f\sin\delta_f & -\frac{l_w}{2} & \frac{l_w}{2} \end{bmatrix}$, $B_u = \begin{bmatrix} F_x & M_Z \end{bmatrix}^T$, $u_c = \begin{bmatrix} T_{fl} & T_{fr} & T_{rl} & T_{rr} \end{bmatrix}^T$.

Therefore, we constructed R_{upper} to consider vehicle stability in the torque allocation layer from the perspective of the active safety layer:

$$R_{upper} = -(A_s u_c - B_u)^T W_s (A_s u_c - B_u) \tag{35}$$

Second, the drive system power consumption is minimized by using the reward function Equation (36).

$$R_{economy} = -W_{economy} \sum_{ij=\{fl,fr,rl,rr\}} P_{ij} = -W_{economy} \sum_{ij=\{fl,fr,rl,rr\}} \frac{T_{ij}\omega_{ij}}{\eta\left(T_{ij},\omega_{ij}\right)} \tag{36}$$

where $W_{economy}$ is the penalty factors of economy, and ω_{ij} is the speed of motor ij. Additionally, the motor efficiency η is shown in Figure 4.

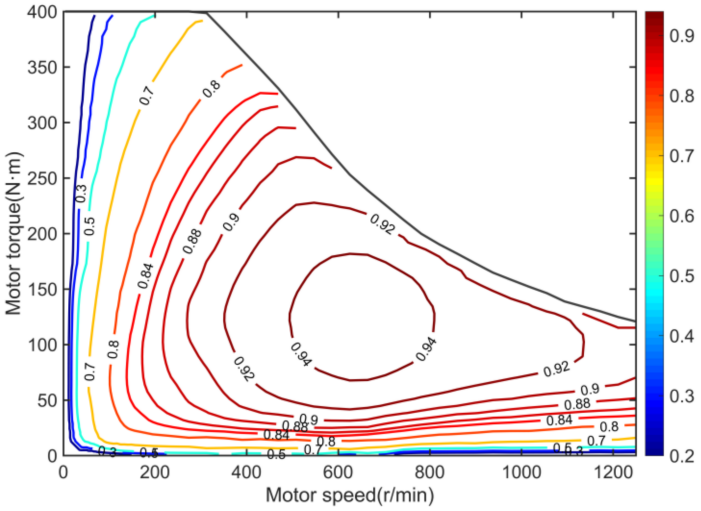

Figure 4. Motor efficiency map.

Lastly, the stability indicators and driver load are considered in the reward function:

$$\begin{aligned} R_{stability} &= -W_{stability}\left((\beta-\beta_d)^2+(\gamma-\gamma_d)^2\right) \\ R_{driver} &= -W_{driver}\left(\dot{\delta}_{sw}^2 + W_{a\_\delta}\cdot a_x^2\right) \end{aligned} \tag{37}$$

where $W_{stability}$ and W_{driver} are the penalty factors for safety and driver load, respectively.

$$R = R_{upper} + R_{economy} + R_{stability} + R_{driver} \tag{38}$$

The safety of torque allocation consideration consists of two parts: R_{upper} and $R_{stability}$. As safety is the primary concern in vehicle operation, we chose penalty factors to ensure that the reward function satisfies the following relationship:

$$\underbrace{R_{upper} + R_{stability}}_{safety} > R_{economy} > R_{driver} \tag{39}$$

The goal of the RL is to learn a policy that maximizes the expected cumulative reward over time. To encourage exploration and prevent premature convergence, the entropy term is included in the policy π:

$$\pi^* = \underset{\pi}{\operatorname{argmax}}\, \mathbb{E}_{\tau\sim\pi}\left[\gamma_f\left(r_t + \xi\mathcal{H}(\pi(\cdot|s_{t+1}))\right)\right] \tag{40}$$

where ξ is a temperature parameter that determines the level of exploration, and the additional entropy term $\mathcal{H}(\cdot)$ encourages exploration and prevents premature convergence to suboptimal policies.

Heuristic REDQ is an improved algorithm of REDQ. REDQ is a deep RL algorithm that combines ideas from both ensemble methods and double Q-learning to improve stability

and sample efficiency [29]. In the REDQ algorithm, the Q function is expressed using the Bellman equation:

$$Q^{\pi}(s_t, a_t) = \mathop{\mathbb{E}}_{a_{t+1}\sim\pi}[r_t + \gamma_f \max_{a_{t+1}} Q^{\pi}(s_{t+1}, a_{t+1}) + \xi\mathcal{H}(\pi(\cdot|s_{t+1}))] \tag{41}$$

where $Q^{\pi}(s_t, a_t)$ is the expected long-term reward of taking action a_t based on policy π in state s_t, $r_t = \mathcal{R}(s_t)$ is the immediate reward based on state s_t, $\gamma_f \in [0, 1]$ is the discount factor, s_{t+1} is the next state after taking action a_t in state s_t, and a_{t+1} is the next action.

To improve stability and reduce overestimation, the REDQ algorithm maintains an ensemble of N Q-networks, where each network is represented by a set of weights θ_i, and the target value of Q-function is calculated by randomly selecting M Q-networks from N Q-networks:

$$y^{REDQ} = r_t + \gamma_f \left(\min_{m\in\mathcal{I}_M} \widetilde{Q}_m(s_{t+1}, a_{t+1}, \widetilde{\theta}_t) + \xi\mathcal{H}(\pi(\cdot|s_{t+1})) \right) \tag{42}$$

where the set $\mathcal{I}_M \subseteq \{1, 2, \ldots, N\}$ is M random elements.

The updated rule of evaluating Q-networks' weights is by minimizing the loss:

$$\mathcal{L}_{\theta}(\theta_i) = \mathop{\mathbb{E}}_{(s_t, a_t, r_t, s_{t+1})} \left[\left(y^{REDQ} - Q_{\theta_i}(s_t, a_t) \right)^2 \right], i = 1, 2, \ldots N \tag{43}$$

The target Q-networks are updated using a Polyak averaging:

$$\widetilde{\theta}_i \leftarrow \rho\widetilde{\theta}_i + (1 - \rho)\theta_i \tag{44}$$

where ρ is a hyperparameter that determines the smoothing factor.

In addition, the parameter ϕ of the policy π_{ϕ} is trained by minimizing the loss:

$$\mathcal{L}_{\phi}(\phi) = \mathop{\mathbb{E}}_{(s_t, a_t, r_t, s_{t+1})} \left[\frac{1}{N} \sum_{i=1}^{N} Q_{\theta_i}(s_t, a_t) + \xi\mathcal{H}(\pi(a_{\phi}(s)|s_t)) \right] \tag{45}$$

As the action here is the four-wheeled torque, its action space is four-dimensional, which adds difficulty to the training convergence of the agent. Therefore, a heuristic decay method is introduced here, with the final action being randomly chosen from the set of action strategies $\{a_{OP}, a_{rule}\}$ with distribution $\{1 - P_{rule}, P_{rule}\}$. The probability of selecting action a_{rule} is:

$$P_{rule} = \left(1 - \tanh\left(\frac{n_r}{\tau} \right) \right) \tag{46}$$

where $\tanh(x) = \frac{e^{2x}-1}{e^{2x}+1}$, n_r is the cumulative number of runs, and τ is a constant that denotes the decay speed. The a_{rule} is obtained from Section 4.1, and the a_{OP} is expressed as:

$$a_{OP}(s_t) = \tanh(\mu_{\phi}(s_t) + \sigma_{\phi}(s_t) \odot \varphi) T_{\max} \tag{47}$$

where $\varphi \sim N(0, I)$ is an independent noise, $\mu_{\phi}(s_t)$ is the mean of the stochastic policy that maps the state s_t to an action, and $\sigma_{\phi}(s_t)$ is a random noise sampled from a probability distribution that encourages exploration. $\odot$ is the Hadamard product, and $T_{\max}$ is the maximum motor torque, which is determined by the external characteristics of the motor.

To reduce overestimation and improve stability, REDQ uses an ensemble of Q-networks to estimate the Q-values and takes the minimum of the target Q-values from all the networks. To further improve sample efficiency, REDQ uses an update-to-data ratio denoted by G, which enables control over the number of times the data is reused. Algorithm 2 shows the detailed steps of the heuristic REDQ.

Algorithm 2. Heuristic REDQ algorithm

1. Initialize an ensemble of Q-networks $Q_\theta(s_t, a_t)$ with parameters θ_i, Set target parameters
2. Initialize the target Q-networks $\widetilde{Q}_{\widetilde{\theta}}(s_t, a_t)$ with parameters $\widetilde{\theta}_i \leftarrow \theta_i$
3. Initialize the replay buffer
4. **For** each step t **do:**
5. Randomly sample an action a_t from the set of action strategies $\{a_{OP}, a_{rule}\}$ with distribution $\{1 - P_{rule}, P_{rule}\}$
6. Execute the action a_t and observe the next state s_{t+1}, reward r_t
7. Store the experience tuple $\{s_t, a_t, r_t, s_{t+1}\}$ in the replay buffer
8. **for** G updates **do**
9. Sample a mini-batch experiences $\{(s_t, a_t, r_t, s_{t+1})\}$ from replay buffer
10. Randomly select m numbers from the set $\{1, 2, \cdots, N\}$ as a set $\mathcal{I}_M$
11. Based on (42) compute the Q-value estimates y^{REDQ}
12. **for** $i = 1, 2, \cdots, N$ **do**
13. Based on (43), update the parameters θ_i using gradient descent method
14. Based on (44), Update each target Q-network $\widetilde{Q}_{\widetilde{\theta}}(s_t, a_t)$
15. **end for**
16. **end for**
17. **end for**
18. Return the learned Q-network ensemble.

4. Evaluation Indicators and Simulation Results

4.1. Simulation Environment

To validate the effectiveness of the TVC algorithm, a joint simulation of CarSim, Simulink, and Python is used in this paper. The vehicle model in CarSim is based on the Magic Formula tire model. The vehicle model includes four in-wheel motors, which are controlled by the torque allocation algorithm implemented in Simulink. Python is used for training the RL algorithm. As the trained RL algorithm will not impose an additional computational burden on the TVC system, the main computational consumption of this paper lies in the nonlinear MPC. We have verified the real-time problem of the nonlinear MPC controller in the published paper [26], and the implemented nonlinear MPC algorithm by the ACADO toolkit [24], which has demonstrated its numerical efficiency for EV control in previous studies [5,30]. In this paper, the following five representative methods were selected for comparison and validation.

1. RLES. The torque allocation algorithm proposed in this paper. The active safety control layer is a nonlinear MPC controller, and the lower controller is based on a heuristic REDQ deep RL algorithm which integrates considering economy and safety.
2. MPC-CO. The torque allocation algorithm proposed in reference [26] which integrates considering economy and safety, where the lower controller is a quadratic planning algorithm.
3. LQR-EQ. The active safety control layer is the LQR controller in reference [31], and the torque allocation layer is a common average allocation method in Section 3.2.1. This controller considers vehicle safety only.
4. w/o control. There is no additional vehicle lateral control; steering is controlled by the driver.

4.2. Performance Indicators

To evaluate the effectiveness of the proposed torque allocation algorithm, several evaluation indicators are employed in this paper. These indicators include:

1. Handling stability

$$\varepsilon_s = -\int_0^t \left(e_\beta^2 + W_{\beta\_\gamma} e_\gamma^2\right) dt \tag{48}$$

where $e_\beta = \beta - \beta_r$, $e_\gamma = \gamma - \gamma_r$, and W_{β_γ} are the weights between e_β^2 and e_γ^2.

2. Driver workload

$$\varepsilon_{driver} = \int_0^t \left(\dot{\delta}_{sw}(\tau)^2 + W_{a\_\delta}(\tau) \cdot a_x(\tau)^2 \right) d\tau \tag{49}$$

where a_x is the longitudinal acceleration, δ_{sw} is the driver steering wheel angle, and W_{a_δ} is the weight between $\dot{\delta}_{sw}{}^2$ and ${a_x}^2$.

3. Motor load

Frequent variations in motor torque may result in thermal saturation, which will increase the wear and tear on the motor.

$$\varepsilon_{motor} = \int_0^t \left(\Delta T_{fl}(\tau)^2 + \Delta T_{fr}(\tau)^2 + \Delta T_{rl}(\tau)^2 + \Delta T_{rr}(\tau)^2 \right) d\tau \tag{50}$$

4. Additional yaw moment

The magnitude of M_z can represent the cost of stability control.

$$\varepsilon_{Mz} = \int_0^t M_z(\tau)^2 d\tau \tag{51}$$

5. Velocity tracking

Vehicle velocity is important for safety assessment, and ensuring vehicle safety while maintaining speed tracking is the goal of stability control.

$$\varepsilon_v = \int_0^t \left(v_{ref}(\tau) - v_x(\tau) \right)^2 d\tau \tag{52}$$

4.3. Training Performance

To ensure the performance of the controller under extreme conditions. We trained the RL agent in the lower controller on the simulation platform described in Section 3.2.2. The training condition is a low adhesion road with an adhesion coefficient of 0.3, and the vehicle velocity was increased from 60 km/h to 100 km/h in 10 s. The RL sampling time is 0.02 s, and the training time per episode is 10 s. The total number of training episodes is 800. The training results are shown in Figure 5.

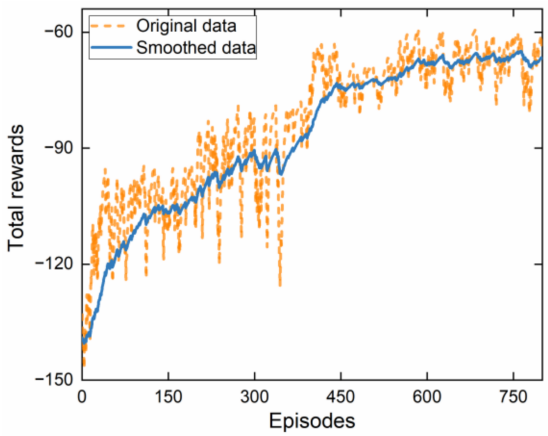

Figure 5. RL algorithm training reward curve.

The training reward curve of RL shows how the agent's cumulative reward changes over the course of the training process. We can see that after about 580 episodes of training, the total rewards of the agent begin to converge at −65. During the early stages of training, the reward curve is erratic, with the agent sometimes achieving high rewards and sometimes achieving low rewards. As the agent learns and the training progresses,

the reward curve gradually increases and stabilizes, indicating that the agent is becoming more proficient at the task.

4.4. DLC Maneuver on Slippery Road

The effectiveness of the proposed algorithm is tested under DLC conditions on slippery road surfaces. The longitudinal initial velocity is 72 km/h, and the tire–road friction coefficient is set as 0.3.

The vehicle trajectory in Figure 6a shows that the proposed RLES torque allocation algorithm provides better stability and control during the double line change maneuver compared to the MPC CO and LQR EQ controllers. The vehicle is able to maintain a smooth trajectory and quickly change its direction without large deviation or instability. Additionally, the vehicle under w/o control deviated significantly from the desired path. The yaw rate and side slip angle in Figure 6b,c also show improvements with the proposed RLES algorithm. The yaw rate and side slip angle of the RLES and MPC CO controllers can track the reference value throughout the maneuver, indicating good vehicle handling and control. The yaw rate and side slip angle in the LQR EQ and w/o controller show more oscillations and instability, which may cause the vehicle to lose control. The phase trajectory portrait in Figure 7a also proves this point.

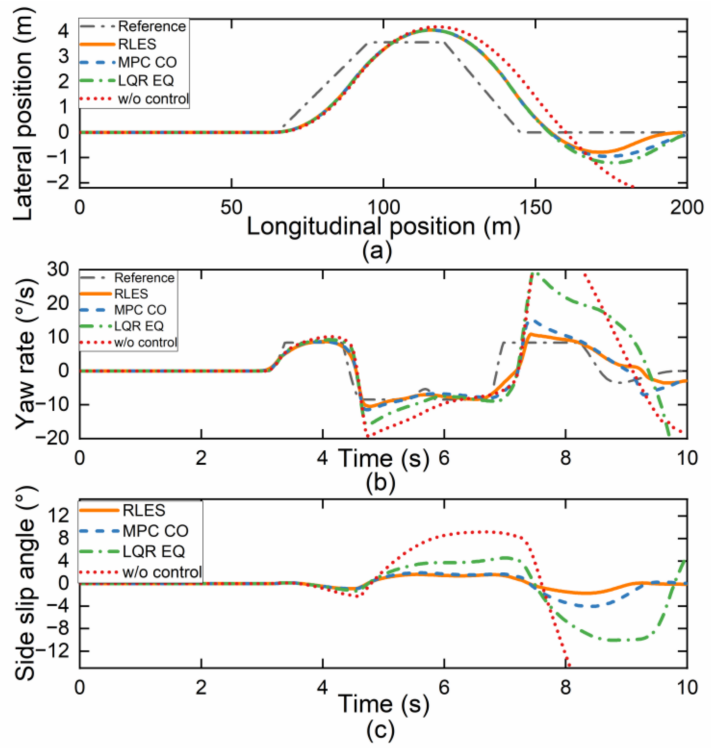

Figure 6. Simulation results on slippery road. (**a**) Vehicle displacement; (**b**) yaw rate; (**c**) side slip angle.

The motor power consumption in Figure 7b also shows improvement with the proposed RLES controller. The motor power consumption is reduced, indicating better fuel economy. The other control strategies show higher motor power consumption, which may result in higher energy consumption and lower fuel economy. Compared to w/o control, RLES, MPC CO, and LQR EQ reduce motor power consumption by 80.2%, 78.4%, and 31.4%, respectively. Table 2 shows the results of the evaluation indicators for the four controllers. The RLES controller shows advantages for all indicators, especially for ε_{driver}, ε_{motor}, and ε_v.

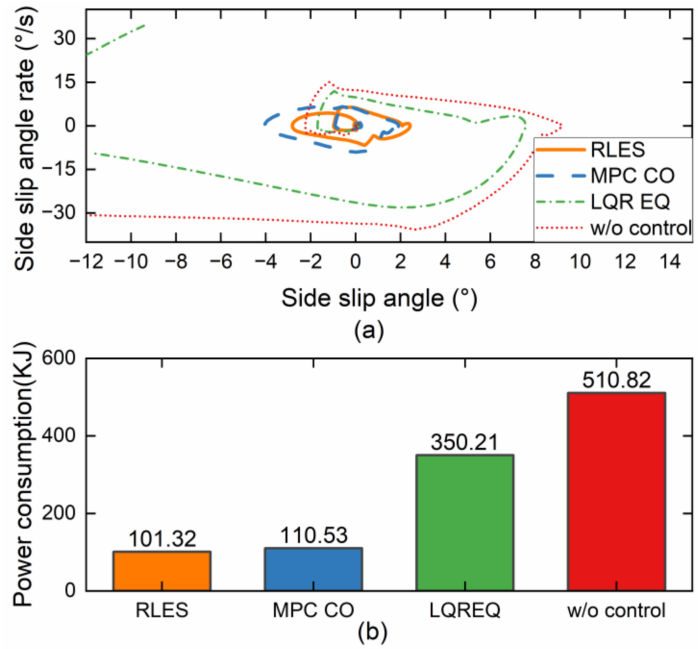

Figure 7. Simulation results on slippery road. (**a**) Phase trajectory portrait; (**b**) motor power consumption.

Table 2. Performance indicators in DLC maneuver on slippery road.

Controller	ε_s		ε_{driver}		ε_{motor}		ε_{Mz}	ε_v
RLES	0.4085	−96%	16.28	−81%	20,820	−93%	735,200	1.1×10^{-5}
MPC CO	0.5640	−94%	19.58	−77%	23,430	−92%	887,200	1.5×10^{-5}
LQR EQ	3.6532	−61%	32.54	−62%	29,680	−90%	892,000	3.5×10^{-4}
w/o control	9.260	-	85.46	-	306,300	-	0	2.9×10^{2}

Overall, the simulation results demonstrate the effectiveness of the torque allocation algorithm in improving vehicle stability, control, and fuel economy during a double-line change maneuver. This is because the RL-based RLES controller can learn the optimal strategy in continuous interaction with the environment and thus output the optimal four-wheel torque according to the current vehicle state.

4.5. DLC Maneuver on Joint Road

To simulate the condition of suddenly encountering an icy road surface at the start of a lane change and analyze the controller performance under extreme conditions, we set the road surface adhesion coefficient as Figure 8. Additionally, the target vehicle speed was 72 km/h, the EV was run in a DLC condition, and the reference trajectory is shown in the reference term in Figure 9a.

The trajectory of a vehicle controlled by RLES has a smaller displacement offset than other controllers as shown in Figure 9a. The vehicle can still complete the DLC maneuver under the w/o controller because of the decrease in vehicle speed, as shown in Figure 10b where the minimum speed of the w/o controller is 51 km/h. The side slip angle of the vehicle controlled by RLES was also well-controlled and remained close to zero, while the LQR EQ and w/o control controllers showed larger deviations from zero, indicating poorer vehicle stability, as shown in Figure 9c. The phase trajectory portrait in Figure 10a shows the relationship between the side slip angle and side slip angle rate, which demonstrates that the proposed RLES torque allocation algorithm provides better stability and control. The phase trajectory of the RLES is within the stability range, while the phase trajectory of

the LQR EQ and w/o control controller exceeded the stability boundary and cannot return to the stability point, which may cause the vehicle to lose control.

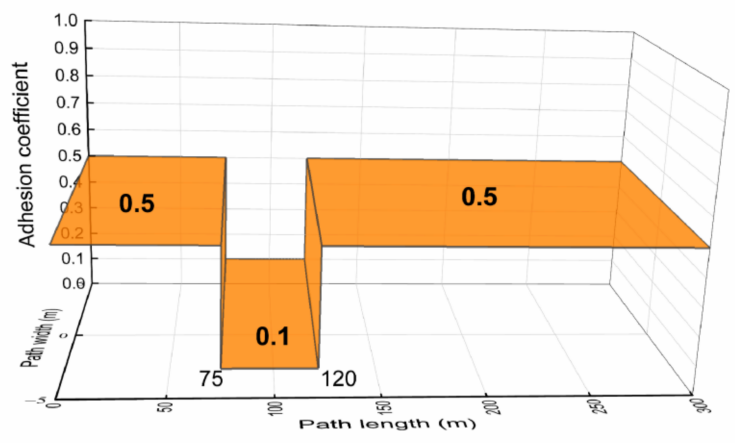

Figure 8. Adhesion coefficient of the joint road.

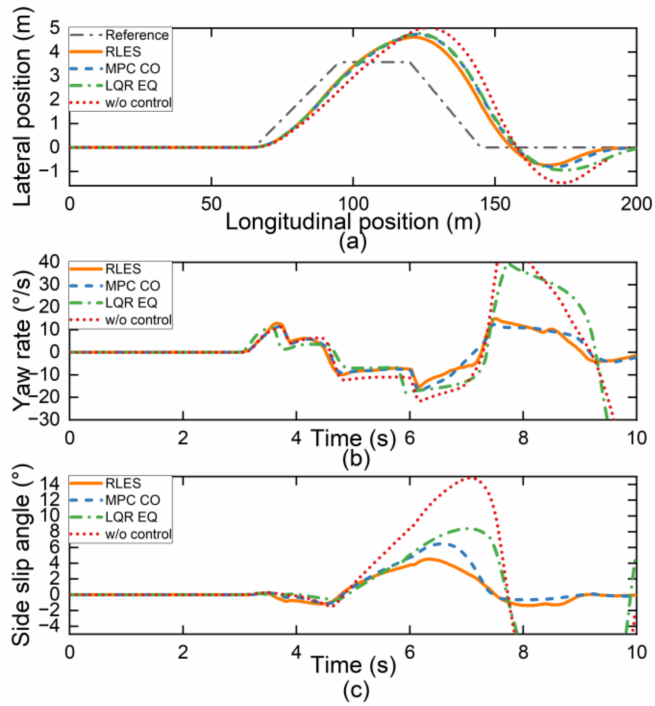

Figure 9. Simulation results under joint road. (**a**) Vehicle displacement; (**b**) yaw rate; (**c**) side slip angle.

Figure 11 shows the stability index and power consumption of the four controllers. It can be seen that both the RLES and MPC CO controllers perform significantly better than the LQR EQ and w/o controller. RLES and MPC CO improve the vehicle economy while ensuring vehicle safety. The performance of RLES is better than the MPC CO controller in terms of stability and power consumption, which indicates that the RLES based on RL training is suitable for different working conditions and can adaptively adopt the optimal control strategy according to the current vehicle state.

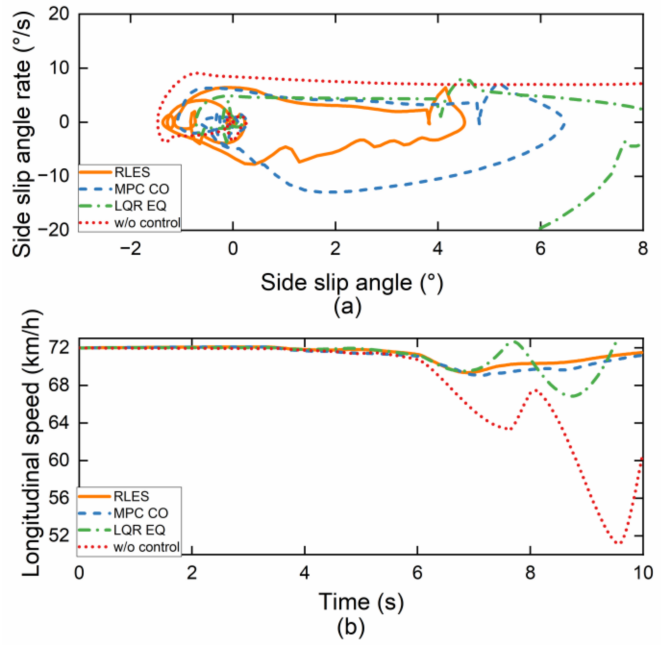

Figure 10. Simulation results under joint road. (**a**) Phase trajectory portrait; (**b**) longitudinal velocity.

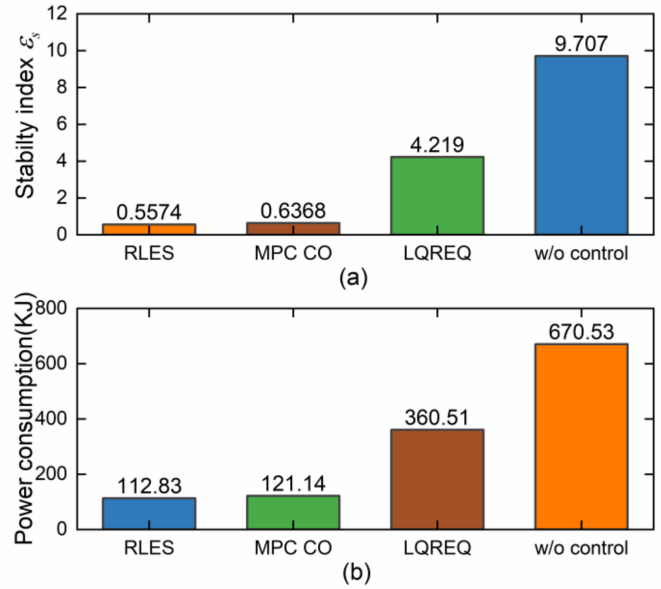

Figure 11. Simulation results under joint road. (**a**) Stability index ε_s; (**b**) motor power consumption.

Table 3 summarizes the performance comparison of the four controllers. The simulation results demonstrate that the proposed RLES controller achieves superior performance in terms of power consumption, driver burden, and vehicle safety, making it a promising solution for the torque allocation of 4MIDEV.

Table 3. Performance indicators in DLC maneuver on joint road.

Controller	ε_s		ε_{driver}		ε_{motor}		ε_{Mz}	ε_v
RLES	0.5574	−94%	16.28	−81%	20,820	−93%	735,200	0.88
MPC CO	0.6368	−93%	19.58	−77%	23,430	−92%	887,200	1.42
LQR EQ	4.2190	−57%	32.54	−62%	29,680	−90%	892,000	2.47
w/o control	9.707	-	85.46	-	306,300	-	0	36.78

4.6. Step Steering Maneuver

To verify the dynamic characteristics of the vehicle, a step steering maneuver was conducted for simulation validation. The vehicle was driving at a speed of 72 km/h on a good road surface with a road adhesion coefficient of 0.75. Within 0.5 s, the steering wheel angle increased from 0 deg to 120 deg and remained unchanged. The simulation results are shown in Figures 12 and 13.

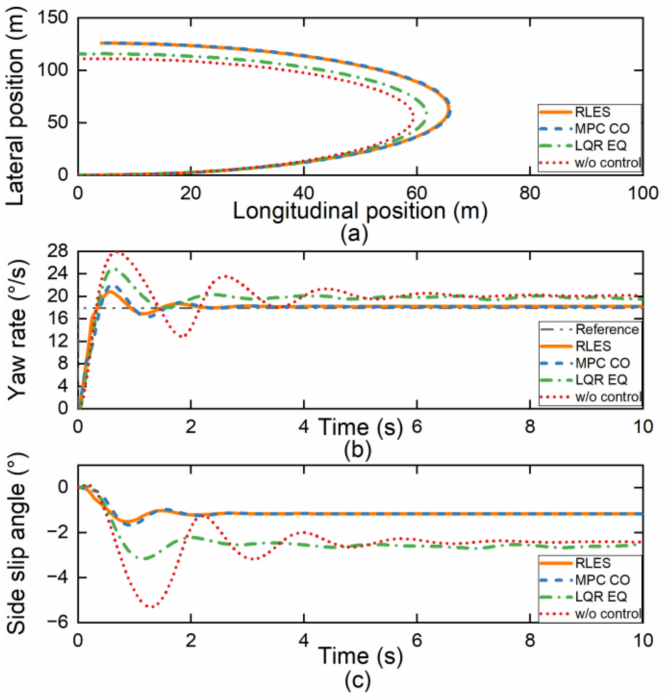

Figure 12. Simulation results under step steering maneuver. (**a**) Vehicle displacement; (**b**) yaw rate; (**c**) side slip angle.

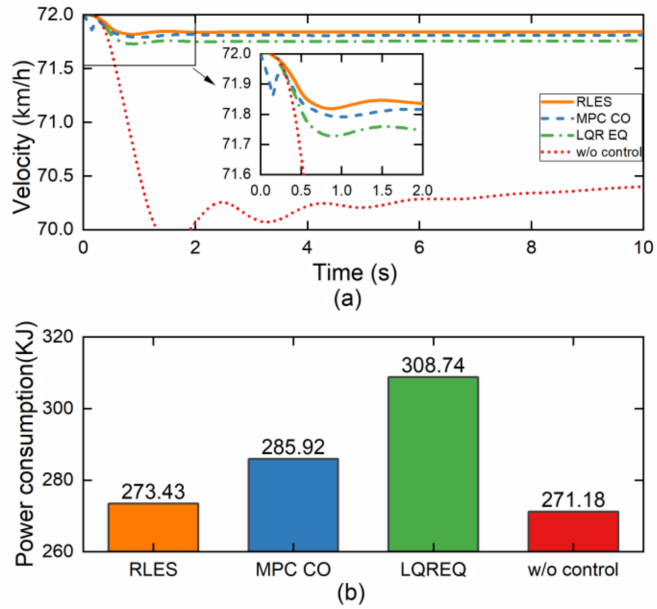

Figure 13. Simulation results under step steering maneuver. (**a**) Longitudinal velocity; (**b**) motor power consumption.

As shown in Figure 12a, the RLES control method shows relative understeering characteristics, which is beneficial for improving driver maneuverability. The yaw rate is

shown in Figure 12b, indicating that both the RLES and MPC CO control methods could track the reference value well. Similarly, the side slip angle for the RLES and MPC CO control methods in Figure 12c is better than that of the LQR EQ and w/o control methods, and the RLES control method has the best maneuvering stability. Figure 13a shows the vehicle velocity variation curve in the step steering simulation, and the RLES method has better velocity tracking performance compared to other methods. The vehicle stability is improved under the RLES method while the longitudinal speed tracking performance is guaranteed. As seen in Figure 13b, except for the w/o control method, RLES consumed the least energy during the simulation. The w/o control method consumed the least energy because it does not require torque distribution to generate additional yaw moment by providing differential drive-assisted steering. However, this control method which sacrifices the vehicle stability is actually unsafe.

Table 4 shows the objective evaluation indicators under different controllers. It can be seen from the table that, except for the w/o control method, the RLES controller outperformed the other controllers in terms of vehicle stability, driver burden, and longitudinal speed tracking performance. In conclusion, the controller based on RLES not only ensures vehicle safety but also reduces the energy consumption of the drive system and improves economy under open-loop simulation conditions.

Table 4. Performance indicators in step steering maneuver.

Controller	ε_s		ε_{driver}		ε_{motor}	ε_{Mz}	ε_v
RLES	0.0183	−62%	42.46	−1.7%	9930	1×10^7	0.01895
MPC CO	0.0193	−60%	42.46	−1.7%	9873	1×10^7	0.0273
LQR EQ	0.0274	−43%	43.21	0.1%	4166	170,100	0.04486
w/o control	0.0483	-	43.18	-	132.7	0	2.202

4.7. Driving Cycles

To evaluate the effectiveness of the proposed TVC algorithm in terms of energy savings, a series of tests were conducted under two different driving cycles: New European Driving Cycle (NEDC) and US06. The NEDC case was designed to simulate urban and high-speed driving scenarios, while the US06 driving cycle was designed to replicate a more aggressive driving behavior on the highway. Figure 14a depicts the vehicle velocity allocation for each driving cycle. Note that the driving cycle conditions do not include lateral control, so the performance of the LQR EQ controller is the same as without the control.

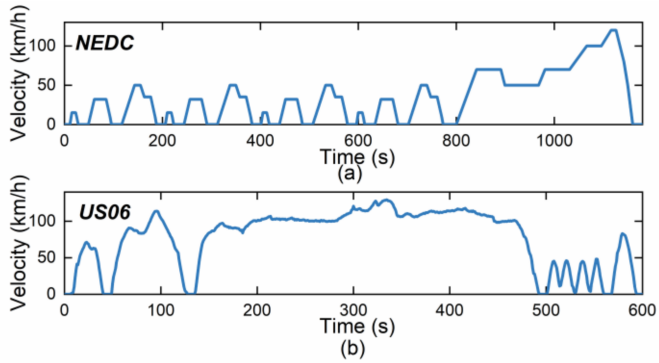

Figure 14. Velocity distribution of driving cycles. (**a**) NEDC; (**b**) US06.

As depicted in Figure 15, the RLES algorithm increases the motor efficiency by 10.9% and reduces the motor power consumption by 11.0% compared to the LQR EQ algorithm in the NEDC cycle. This reduction in power consumption is achieved by optimizing the torque allocation strategy, thus allowing the motor to operate in the high-efficiency range as much as possible. In addition, the RLES also achieves optimal performance in the US60

driving cycle in Figure 15. These results demonstrate the RLES algorithm can adapt to different driving conditions and optimize the torque allocation strategy to achieve better energy efficiency.

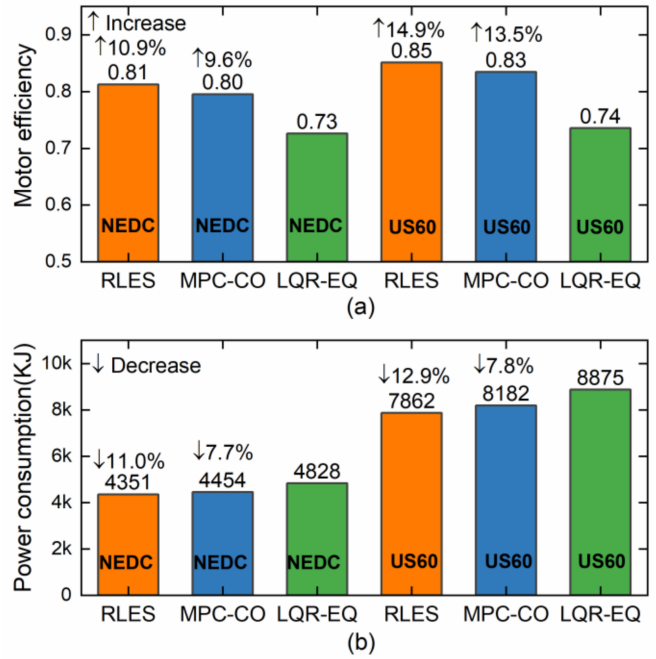

Figure 15. Simulation results under driving cycles. (**a**) Motor efficiency; (**b**) motor power consumption.

5. Conclusions

The 4MIDEV has the feature of independently controllable four-wheel motors. To fully utilize this feature, the paper introduces a novel RL-based TVC algorithm for 4MIDEV that takes into account both economy and safety. The four-wheel tire model is identified using FTO with experimental data, and the active safety control layer utilizes a nonlinear MPC to calculate the required additional yaw moment and longitudinal force. The torque allocation layer employs a heuristic REDQ deep RL algorithm to compute the optimal four-wheel torque. The proposed RLES controller is validated and compared with typical MPC CO, LQR EQ, and w/o controllers under different driving scenarios. The results demonstrate that the proposed RLES enhances vehicle economy and reduces driver workload while ensuring vehicle safety. Future work will focus on designing a TVC algorithm for different driver steering characteristics and conducting real vehicle experiments.

Author Contributions: Conceptualization, H.D.; methodology, H.D.; software, Q.W.; validation, F.L.; formal analysis, Y.Z.; investigation, Y.Z.; writing—original draft preparation, H.D.; writing—review and editing, Y.Z.; supervision, Y.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Postgraduate Research and Practice Innovation Program of Jiangsu Province, grant number KYCX21_0188, the National Natural Science Foundation of China grant number 52272397, 11672127, the Fundamental Research Funds for the Central Universities, grant number NP2022408, and the Army Research and the National Engineering Laboratory of High Mobility anti-riot vehicle technology, grant number HTF B20210017.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No external data were used in this study.

Acknowledgments: The authors gratefully acknowledge the financial support from the Postgraduate Research and Practice Innovation Program of Jiangsu Province (No. KYCX21_0188).

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

References

1. Wu, J.; Zhang, J.; Nie, B.; Liu, Y.; He, X. Adaptive Control of PMSM Servo System for Steering-by-Wire System With Disturbances Observation. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 2015–2028. [CrossRef]
2. Wu, J.; Kong, Q.; Yang, K.; Liu, Y.; Cao, D.; Li, Z. Research on the Steering Torque Control for Intelligent Vehicles Co-Driving With the Penalty Factor of Human–Machine Intervention. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 59–70. [CrossRef]
3. Lei, F.; Bai, Y.; Zhu, W.; Liu, J. A novel approach for electric powertrain optimization considering vehicle power performance, energy consumption and ride comfort. *Energy* **2019**, *167*, 1040–1050. [CrossRef]
4. Karki, A.; Phuyal, S.; Tuladhar, D.; Basnet, S.; Shrestha, B.P. Status of Pure Electric Vehicle Power Train Technology and Future Prospects. *Appl. Syst. Innov.* **2020**, *3*, 35. [CrossRef]
5. Dalboni, M.; Tavernini, D.; Montanaro, U.; Soldati, A.; Concari, C.; Dhaens, M.; Sorniotti, A. Nonlinear Model Predictive Control for Integrated Energy-Efficient Torque-Vectoring and Anti-Roll Moment Distribution. *IEEE/ASME Trans. Mechatron.* **2021**, *26*, 1212–1224. [CrossRef]
6. Chatzikomis, C.; Zanchetta, M.; Gruber, P.; Sorniotti, A.; Modic, B.; Motaln, T.; Blagotinsek, L.; Gotovac, G. An energy-efficient torque-vectoring algorithm for electric vehicles with multiple motors. *Mech. Syst. Sig. Process.* **2019**, *128*, 655–673. [CrossRef]
7. Xu, W.; Chen, H.; Zhao, H.; Ren, B. Torque optimization control for electric vehicles with four in-wheel motors equipped with regenerative braking system. *Mechatronics* **2019**, *57*, 95–108. [CrossRef]
8. Hu, X.; Wang, P.; Hu, Y.; Chen, H. A stability-guaranteed and energy-conserving torque distribution strategy for electric vehicles under extreme conditions. *Appl. Energy* **2020**, *259*, 114162. [CrossRef]
9. Ding, S.H.; Liu, L.; Zheng, W.X. Sliding Mode Direct Yaw-Moment Control Design for In-Wheel Electric Vehicles. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6752–6762. [CrossRef]
10. Zhao, B.; Xu, N.; Chen, H.; Guo, K.; Huang, Y. Stability control of electric vehicles with in-wheel motors by considering tire slip energy. *Mech. Syst. Sig. Process.* **2019**, *118*, 340–359. [CrossRef]
11. Zhang, L.; Chen, H.; Huang, Y.; Wang, P.; Guo, K. Human-Centered Torque Vectoring Control for Distributed Drive Electric Vehicle Considering Driving Characteristics. *IEEE Trans. Veh. Technol.* **2021**, *70*, 7386–7399. [CrossRef]
12. Li, Q.; Zhang, J.; Li, L.; Wang, X.; Zhang, B.; Ping, X. Coordination Control of Maneuverability and Stability for Four-Wheel-Independent-Drive EV Considering Tire Sideslip. *IEEE Trans. Transp. Electrif.* **2022**, *8*, 3111–3126. [CrossRef]
13. Deng, H.; Zhao, Y.; Nguyen, A.T.; Huang, C. Fault-Tolerant Predictive Control With Deep-Reinforcement-Learning-Based Torque Distribution for Four In-Wheel Motor Drive Electric Vehicles. *IEEE/ASME Trans. Mechatron.* **2023**, early access. [CrossRef]
14. Aradi, S. Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 740–759. [CrossRef]
15. Zhu, Y.; Wang, Z.; Chen, C.; Dong, D. Rule-Based Reinforcement Learning for Efficient Robot Navigation With Space Reduction. *IEEE/ASME Trans. Mechatron.* **2022**, *27*, 846–857. [CrossRef]
16. Kiran, B.R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A.A.A.; Yogamani, S.; Pérez, P. Deep Reinforcement Learning for Autonomous Driving: A Survey. *IEEE Trans. Intell. Transp. Syst.* **2022**, *23*, 4909–4926. [CrossRef]
17. Wei, H.; Zhang, N.; Liang, J.; Ai, Q.; Zhao, W.; Huang, T.; Zhang, Y. Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance. *Energy* **2022**, *238*, 121725. [CrossRef]
18. Peng, H.; Wang, W.; Xiang, C.; Li, L.; Wang, X. Torque Coordinated Control of Four In-Wheel Motor Independent-Drive Vehicles With Consideration of the Safety and Economy. *IEEE Trans. Veh. Technol.* **2019**, *68*, 9604–9618. [CrossRef]
19. Cabrera, J.A.; Ortiz, A.; Carabias, E.; Simon, A. An Alternative Method to Determine the Magic Tyre Model Parameters Using Genetic Algorithms. *Veh. Syst. Dyn.* **2004**, *41*, 109–127. [CrossRef]
20. Alagappan, A.; Rao, K.V.N.; Kumar, R.K. A comparison of various algorithms to extract Magic Formula tyre model coefficients for vehicle dynamics simulations. *Veh. Syst. Dyn.* **2015**, *53*, 154–178. [CrossRef]
21. Hu, C.; Wang, R.R.; Yan, F.J.; Chen, N. Should the Desired Heading in Path Following of Autonomous Vehicles be the Tangent Direction of the Desired Path? *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 3084–3094. [CrossRef]
22. Ji, X.; He, X.; Lv, C.; Liu, Y.; Wu, J. A vehicle stability control strategy with adaptive neural network sliding mode theory based on system uncertainty approximation. *Veh. Syst. Dyn.* **2018**, *56*, 923–946. [CrossRef]
23. Zhang, H.; Liang, J.; Jiang, H.; Cai, Y.; Xu, X. Stability Research of Distributed Drive Electric Vehicle by Adaptive Direct Yaw Moment Control. *IEEE Access* **2019**, *7*, 106225–106237. [CrossRef]
24. Houska, B.; Ferreau, H.J.; Diehl, M. An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica* **2011**, *47*, 2279–2285. [CrossRef]
25. Wang, J.; Luo, Z.; Wang, Y.; Yang, B.; Assadian, F. Coordination Control of Differential Drive Assist Steering and Vehicle Stability Control for Four-Wheel-Independent-Drive EV. *IEEE Trans. Veh. Technol.* **2018**, *67*, 11453–11467. [CrossRef]
26. Deng, H.; Zhao, Y.; Feng, S.; Wang, Q.; Zhang, C.; Lin, F. Torque vectoring algorithm based on mechanical elastic electric wheels with consideration of the stability and economy. *Energy* **2021**, *219*, 119643. [CrossRef]
27. Wu, X.; Zhou, B.; Wen, G.; Long, L.; Cui, Q. Intervention criterion and control research for active front steering with consideration of road adhesion. *Veh. Syst. Dyn.* **2018**, *56*, 553–578. [CrossRef]

28. Zhai, L.; Sun, T.M.; Wang, J. Electronic Stability Control Based on Motor Driving and Braking Torque Distribution for a Four In-Wheel Motor Drive Electric Vehicle. *IEEE Trans. Veh. Technol.* **2016**, *65*, 4726–4739. [CrossRef]
29. Chen, X.; Wang, C.; Zhou, Z.; Ross, K. Randomized Ensembled Double Q-Learning: Learning Fast Without a Model. *arXiv* **2021**, arXiv:2101.05982.
30. Parra, A.; Tavernini, D.; Gruber, P.; Sorniotti, A.; Zubizarreta, A.; Perez, J. On Nonlinear Model Predictive Control for Energy-Efficient Torque-Vectoring. *IEEE Trans. Veh. Technol.* **2021**, *70*, 173–188. [CrossRef]
31. Mirzaei, M. A new strategy for minimum usage of external yaw moment in vehicle dynamic control system. *Transp. Res. Part C Emerg. Technol.* **2010**, *18*, 213–224. [CrossRef]

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