

Torque control strategy for in-wheel motor drive vehicles based on hierarchical deep reinforcement learning considering active safety and energy saving

Yalei LIU¹, Xinyu QI¹, Weiping DING¹, Honglin ZHU², Mingliang YANG¹

¹*School of Mechanical Engineering, Southwest Jiaotong University, Chengdu 610031, China*

²*College of Physics and Electronic Engineering, Chengdu Normal University, Chengdu 611130, China*

Section S1 Vehicle system modeling

Section S1.1 Reference dynamic model

This section provides the detailed derivation of the steady-state reference model used in the upper-level agent for yaw stability control. Based on the two-degree-of-freedom vehicle dynamics presented in Eqs. (1) and (2) of the main text, setting the time derivatives of the state variables to zero yields the equilibrium expressions for the desired sideslip angle and yaw rate. The derivation proceeds as follows.

$$\begin{cases} \beta_{\text{des}} = R_{\beta} \times \delta \\ \gamma_{\text{des}} = R_{\gamma} \times \delta \end{cases} \quad (\text{S1})$$

$$\begin{cases} R_{\beta} = \frac{l_f C_r - m l_f V_x^2 / L}{l_r C_r (1 + K V_x^2)} \times \frac{l_r}{L} \\ R_{\gamma} = \frac{1}{1 + K V_x^2} \times \frac{V_x}{L} \end{cases} \quad (\text{S2})$$

$$\begin{cases} K = \frac{m}{L^2} \times \frac{l_f C_f - l_r C_r}{C_f C_r} \\ L = l_f + l_r \end{cases} \quad (\text{S3})$$

Where, β_{des} and γ_{des} denote the ideal sideslip angle and yaw rate, respectively; K is the understeer coefficient; and L represents the wheelbase, i.e., the distance from the front axle to the rear axle. Lateral acceleration is significantly influenced by road conditions; consequently, the ideal yaw rate is constrained by road adhesion.

$$|\gamma_{\text{des}}| = \min \left\{ |R_{\gamma} \delta|, |\mu g / V_x| \right\} \times \text{sign}(\delta) \quad (\text{S4})$$

Where, $\pm \mu g / V_x$ is determined by the lateral acceleration limit $\alpha_y \leq \mu g$. Furthermore, to optimize lateral motion and minimize tire slip, β_{des} is typically set to zero.

Section S1.2 Magic Formula Tire Model

This section presents the detailed formulation of the tire force model used in the vehicle dynamics simulation.

Tires, as the sole components in direct contact between the vehicle and the road surface, exhibit non-negligible energy consumption during operation. Tire slip ratio estimation and tire dynamics theory form the foundation for the investigation of vehicle dynamics and tire slip energy losses. The Magic Formula, characterized by a unified model structure, high fitting accuracy, and robustness, is expressed in its generic form as follows:

$$y(x) = D \sin \left\{ C \arctan \left[\left(Bx - E \left(Bx - \arctan(Bx) \right) \right) \right] \right\} \quad (S5)$$

In the Magic Formula, $y(x)$ represents the tire output variable, which can be the lateral force, longitudinal force, or self-aligning torque, while the independent variable x corresponds to tire inputs such as slip angle or longitudinal slip ratio. B denotes the stiffness factor; $B * C * D$ represents the cornering stiffness as the slip angle approaches zero, corresponding to the slope at the origin of the Magic Formula curve; C is the shape factor that governs the functional relationship between x and $y(x)$ and influences the overall curve shape; D is the peak factor that determines the maximum value of the curve; E is the curvature factor that affects the curvature around the peak region.

The longitudinal slip ratio λ_{xi} of a wheel is defined as:

$$\lambda_{xi} = \frac{w_i R_{\text{eff},i} - v_{xi}}{\max(w_i R_{\text{eff},i}, v_{xi})} \quad (S6)$$

In the equation, w_i denotes the rotational speed of the tire; v_{xi} represents the center velocity of the wheel; and $R_{\text{eff},i}$ is the effective rolling radius of the wheel.

Section S1.3 Power loss analysis of motor and modeling

This section presents the motor efficiency characteristics used for energy loss quantification. The vehicle is equipped with four identical permanent magnet synchronous motors (PMSMs). The efficiency map, derived from dynamometer test data, is illustrated in **Fig. S1**, and the key motor parameters are summarized in **Table S1**.

Table S1 Key parameters of the in-wheel motor

Parameter	Value	Unit
Motor type	PMSM	-
Rated speed	520	Rpm
Peak speed	1230	Rpm
Rated torque	325	N m
Peak torque	850	N m
Rated power	21	kW
Peak power	42	kW

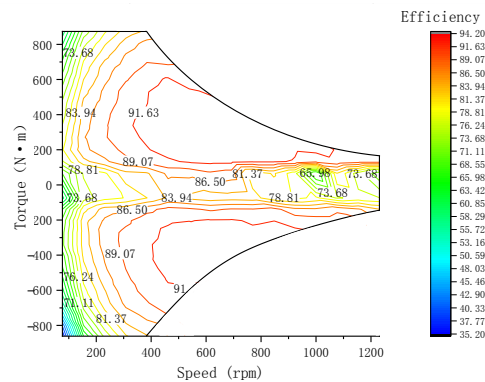


Fig. S1 Motor efficiency map

Section S2 Deep Reinforcement Learning Methodology Details

Section S2.1 Markov Decision Process Formulation

This section presents the fundamental formulation of the Markov decision process (MDP) that underlies the deep reinforcement learning framework employed in this study.

Deep reinforcement learning serves as a powerful paradigm for addressing intricate optimization challenges characterized by high-dimensional state and action spaces. Within this framework, an agent progressively refines its decision-making policy through continuous

environmental interaction. At each step, it selects actions based on current observations, receives scalar feedback signals, and updates its behavioral strategy to maximize cumulative long-term returns. The markov decision process (MDP) provides the mathematical framework describing agent–environment interactions and can be represented as a quadruple (S, A, P, R) . Specifically, S denotes the state space, i.e., the set of all possible states; A represents the action space, i.e., the set of all feasible actions the agent can select at each state; P is the state transition probability function, describing the probability of transitioning to the next state s' given the current state s and action a , denoted as $P(s'|s, a)$, which captures the dynamic characteristics and inherent uncertainty of the environment; R is the reward function, defining the scalar reward received upon executing action a in state s , denoted as $R(s, a)$. The Markov property serves as the defining characteristic of Markov decision processes, asserting that future state transitions and reward signals depend solely on the present state–action pair without dependence on preceding trajectory history. Translating real-world control challenges into this formalism demands deliberate engineering of three foundational components: the state representation, the action repertoire, and the reward specification—each critically shaping the agent’s learning trajectory and eventual policy quality.

Section S2.2 Reward Function Design for the Upper Agent

This section presents the detailed configuration of the reward function for the upper agent, including the weighting coefficients and the piecewise penalty function designed to enhance longitudinal speed tracking accuracy.

In the reward function design, priority is given to maintaining longitudinal vehicle speed and ensuring safety performance. Simultaneously, the rate of change of action commands is incorporated into the reward function to constrain action fluctuations and guarantee smooth vehicle operation. The parameters $a_1 \cdots a_6$ denotes the weighting coefficient. Additionally, a piecewise function H_v is designed to guide the vehicle toward more accurate tracking of the target speed.

$$H_v = \begin{cases} -50 & 3.6\text{km/h} < \Delta V_x \\ -10 & 0.5\text{km/h} < \Delta V_x \leq 3.6\text{km/h} \\ 5 & 0.2\text{km/h} < \Delta V_x \leq 0.5\text{km/h} \\ 25 & \Delta V_x \leq 0.2\text{km/h} \end{cases} \quad (S7)$$

The weighting coefficients through in Eq. (6) of the main text were determined via trial-and-error combined with grid search during the early stage of training. The determination process followed the principles below: (1) Longitudinal speed tracking accuracy was assigned the highest priority, with a_1 and a_6 taking relatively large weights; (2) Yaw stability was given the next priority, with a_2 and a_3 taking moderate values; (3) The penalty on action variation was set to the smallest magnitude, with a_4 and a_5 taking smaller values to allow necessary control adjustments.

The piecewise function in Eq. (6) of the main text is designed to impose a staged penalty on speed tracking error: when the error remains within the acceptable range (< 0.5 km/h), the negative penalty is converted into a positive reward; when the error exceeds this threshold, a staged penalty is applied based on the deviation magnitude. This piecewise design provides clearer gradient signals during early training and better highlights the priority of control objectives compared with a single quadratic function

Section S2.3 Reward Function Design for the lower Agent

This section presents the detailed configuration of the reward function for the lower agent, including the weighting coefficients, the piecewise penalty functions H_β and H_γ , and the threshold determination process.

In the reward function design, Eq. (11) (in the main text) attention is directed toward both

vehicle energy losses and active safety performance. The parameter $b_1 \cdots b_4$ denotes the weighting coefficient. The functions H_γ and H_β are designed as piecewise functions to establish a stepped reward structure that better guides the agent toward optimal actions.

$$H_\beta = \begin{cases} 8 & 0.1 \leq \Delta\beta \\ 2 & 0.0375 \leq \Delta\beta < 0.1 \\ 0 & \Delta\beta < 0.0375 \end{cases} \quad (S8)$$

$$H_\gamma = \begin{cases} 10 & 0.4 \leq \Delta\gamma \\ 3 & 0.0385 \leq \Delta\gamma < 0.4 \\ 0 & \Delta\gamma < 0.0385 \end{cases} \quad (S9)$$

The weighting coefficients through in Eq. (11) (in the main text) are also determined via trial-and-error, with the design following a safety-first principle: the weights of safety-related terms b_3, b_4 are set significantly higher than those of energy-related terms b_1, b_2 , ensuring that vehicle stability is prioritized under all operating conditions. Only when safety indicators remain within reasonable bounds, resulting in relatively low safety penalties, can the optimization of energy efficiency further enhance the reward feedback. The piecewise functions and impose staged penalties on sideslip angle and yaw rate error, respectively, sharing the same design philosophy as . No penalty is applied when the controlled variable remains within the safe region, allowing a certain margin of deviation. When the variable exceeds the safety threshold but stays within an acceptable range, a relatively small penalty is imposed. Once the variable goes beyond the acceptable range, the penalty increases sharply, compelling the agent to strictly confine key state variables within the stability boundaries and prevent vehicle instability. This design ensures the primacy of safety among the control objectives.

The specific threshold values of the piecewise functions and were determined through iterative trial-and-error: several reference studies (Wei et al., 2022a; Wei et al., 2022b) together with the stability boundaries derived from vehicle dynamics theory, served as initial references; then, through multiple comparative training sessions, the agent's convergence behavior and vehicle response were closely monitored, and the thresholds were repeatedly adjusted until satisfactory stability and convergence speed were achieved.

Section S2.4 Quadratic Programming and Constraint Formulation

This section presents the standard quadratic programming (QP) formulation of the torque allocation problem, including the objective function, the equality constraints on total driving force and yaw moment, and the inequality constraints on motor torque limits and tire-road adhesion limits.

$$\min_{\mathbf{u}} J(\mathbf{u}) = \frac{1}{2} \mathbf{u}^T \mathbf{H} \mathbf{u} + \mathbf{f}^T \mathbf{u} \quad (S10)$$

In the equation, $\mathbf{H} \in \mathbb{R}^{4 \times 4}$ denotes the positive definite Hessian matrix, whose diagonal elements are primarily composed of the quadratic coefficients of the tire workload ratio and the comprehensive energy consumption coefficients, reflecting the system's preference for balanced load distribution and enhanced energy saving. The vector $\mathbf{f} \in \mathbb{R}^{4 \times 1}$ represents the linear term vector, incorporating the first-order coefficients from the comprehensive energy consumption model. Both are dynamically modulated in real time by the weighting coefficient τ output from the lower-level agent to adjust the optimization direction. The solution process must strictly adhere to vehicle dynamic requirements and physical system boundaries. To ensure the physical feasibility and control accuracy, the optimization is subject to the following equality and inequality constraints:

The sum of the longitudinal forces generated by the four wheels must satisfy the total driving force demand, and the generated yaw moment must track the reference value from the upper-level agent. These constraints are expressed as:

$$\sum_{i=1}^4 \frac{T_i}{R_{\text{eff},i}} = F_{\text{Target}} \quad (\text{S11})$$

$$\sum_{i=1}^4 (-1)^{k_i} \frac{T_i l_b}{2R_e} = M_d \quad (\text{S12})$$

Where T_i represents the driving torque of the i -th in-wheel motor ($i=1,2,3,4$ corresponding to the front-left, front-right, rear-left, and rear-right wheels, respectively), $R_{\text{eff},i}$ is the effective rolling radius of the wheel, F_{Target} and M_d are the total target driving force and reference yaw moment, l_b is the wheel track, k_i is the index determining the direction of moment generation (0 for left wheels, 1 for right wheels).

The torque distribution is bounded by the motor's physical capabilities and the tire-road adhesion limits. First, the torque of each in-wheel motor is constrained by its maximum output capability:

$$T_{i,\min} \leq T_i \leq T_{i,\max} \quad (\text{S13})$$

Where $T_{i,\min}$ and $T_{i,\max}$ are determined by the motor's external characteristic curve and current speed.

Second, to maintain vehicle stability, the resultant force of each tire must remain within the friction circle. Considering the coupling of longitudinal force F_{xi} and lateral force F_{yi} , the constraint is given by:

$$F_{xi}^2 + F_{yi}^2 \leq (\mu \cdot F_{zi})^2 \quad (\text{S14})$$

Where μ is the road adhesion coefficient and F_{zi} is the vertical load on the i -th wheel. Substituting $F_{xi} = T_i / R_{\text{eff},i}$, the inequality constraint for the optimization variable T_i becomes:

$$-\sqrt{(\mu F_{zi})^2 - (F_{yi})^2} \leq \frac{T_i}{R_{\text{eff},i}} \leq \sqrt{(\mu F_{zi})^2 - (F_{yi})^2} \quad (\text{S15})$$

The final feasible region for the QP solver is the intersection of the motor limits (**Eq. (S13)**) and the adhesion limits (**Eq. (S15)**).

The above two agents are not trained independently or in parallel, but rather in a serial joint training manner. At each time step, the upper agent first outputs the total driving torque and reference yaw moment based on the current state; subsequently, the lower agent takes the upper agent's outputs as part of its state input and further decides the weighting coefficient τ . The QP solver then computes the four-wheel torques based on these three quantities and applies them to the vehicle environment, which subsequently returns separate reward signals to the two agents. Each agent maintains its own experience replay buffer and updates its policy and value networks independently based on its own reward signal, with no gradient backpropagation between them. This hierarchical design decomposes the original four-dimensional torque decision problem into two low-dimensional subproblems, effectively reducing the policy learning difficulty.

Section S2.5 SAC and Network Architecture Design

This section presents the detailed mathematical formulations of the Soft Actor-Critic (SAC) algorithm, including the definition of value functions, the temporal difference (TD) error, the policy and value network update rules, and the entropy-regularized objective function.

Typically, the following two types of value functions are involved:

$$\begin{cases} V(s) = E_{\pi}[G_t | S_t = s] \\ Q(s, a) = E_{\pi}[G_t | S_t = s, A_t = a] \end{cases} \quad (\text{S16})$$

In the equations, the state value function $V(s)$ represents the expected cumulative reward obtained by executing all subsequent actions following policy π from state s . The state-action value function $Q(s, a)$ denotes the expected cumulative reward obtained by selecting action a in state s and subsequently following policy π for all remaining actions.

Sutton and Barto (2018) proposed the most fundamental one-step Actor-Critic algorithm,

with the basic procedure outlined as follows:

(1 The Actor and Critic networks are randomly initialized.

(2 An episode is initiated; in tasks with terminal states, the complete process from the agent's initial action output until transition to a terminal state constitutes a single episode. At each time step, a one-step MDP trajectory $\{s_t, a_t, r_t, s_{t+1}\}$ is sampled from the environment. For this trajectory, the temporal difference (TD) error δ_t is computed by introducing temporal difference learning, as formulated below:

$$\delta_t = r_t + \xi V(s_{t+1}; \phi) - V(s_t; \phi) \quad (\text{S17})$$

Based on the δ_t , the objective for updating the Critic is to minimize the mean squared error—equivalently, the loss function—formulated as follows:

$$\begin{cases} J_v(\phi) = \sum_t \delta_t^2 \\ \phi \leftarrow \phi - \lambda_c \nabla_{\phi} J_v(\phi) \end{cases} \quad (\text{S18})$$

The update objective for the Actor is to maximize the expected reward, formulated as follows:

$$\begin{cases} J_v(\phi) = \sum_t \log \pi(a_t | s_t; \theta) \delta_t \\ \theta \leftarrow \theta + \lambda_a \nabla_{\theta} J_v(\phi) \end{cases} \quad (\text{S19})$$

(3 Steps in (2 are repeated until the episode terminates, after which a new episode is initiated.

Building upon the one-step Actor-Critic algorithm, the SAC algorithm was introduced by Haarnoja et al. (2018). The principal enhancement of SAC lies in the integration of policy optimization with entropy maximization to achieve more stable and sample-efficient policy learning. By incorporating the maximum entropy principle into policy formulation, SAC not only pursues maximization of cumulative rewards but also explicitly encourages policy exploration, thereby enhancing robustness and generalization capability.

Specifically, SAC formalizes the policy optimization problem as maximization of the following objective function:

$$\begin{cases} J(\pi) = \sum_{t=0}^T \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} [r(s_t, a_t) + \alpha H(\pi(\cdot | s_t))] \\ H(\pi(\cdot | s_t)) = -\mathbb{E}_{a_t \sim \pi(\cdot | s_t)} [\log \pi(a_t | s_t; \theta)] \end{cases} \quad (\text{S20})$$

In the equation, $H(\pi(\cdot | s_t))$ denotes the entropy of the policy, and α is the entropy weight that governs the trade-off between exploration and exploitation. This objective function promotes exploration by maximizing policy entropy, enabling the agent to thoroughly explore the state-action space and thereby discover superior policies.

In the Critic component, SAC employs two Q-value functions, $Q_{\phi_1}(s, a)$ and $Q_{\phi_2}(s, a)$, to estimate the value of actions. By introducing twin Q-networks, SAC effectively reduces bias in Q-value estimation and mitigates the Q-value overestimation problem. Excessive bias can lead to overly optimistic value estimates, causing training divergence. The objective of the Critic networks is to minimize the following loss function:

$$J_Q(\phi_i) = L(\phi_i) = \mathbb{E}_{(s, a, r, s')} [(Q_{\phi_i}(s, a) - y)^2] \quad (\text{S21})$$

In the equation, the target value y is computed as follows:

$$y = r + \xi \left(\min_{i=1,2} Q_{\bar{\phi}_i}(s', a') - \alpha \log \pi(a_t | s_t; \theta) \right) \quad (\text{S22})$$

In the equation, $\bar{\phi}_i$ denotes the parameters of the target Q-network, and ξ is the discount factor.

The parameters $\bar{\phi}_i$ of the target Q-network are updated via soft update, according to the

following formula:

$$\bar{\phi}_i \leftarrow \varepsilon \phi_i + (1 - \varepsilon) \bar{\phi}_i \quad (\text{S23})$$

In this formulation, the soft update coefficient ε governs the rate of exponential moving average for target network parameters and is conventionally assigned a small magnitude on the order of 10^{-3} to ensure stable learning dynamics.

In the Actor component, SAC updates the policy network parameters θ by maximizing the following objective function:

$$J_{\pi}(\theta) = \mathbb{E}_{s \sim D} \left[\mathbb{E}_{a \sim \pi_{\theta}} (\alpha \log \pi(a_t | s_t; \theta) - Q_{\phi}(s, a)) \right] \quad (\text{S24})$$

In the equation, D denotes the experience replay buffer. The experience replay buffer in SAC stores experience samples—typically state–action–reward–next-state tuples—generated through agent–environment interactions. During policy and value network updates, the agent randomly samples experiences from the buffer. This random sampling decorrelates temporally correlated transitions, reduces gradient variance, and enhances learning stability and sample efficiency. Furthermore, experience replay enables multiple reuse of the same experiences, thereby improving data efficiency.

Furthermore, the SAC algorithm incorporates an automatic entropy adjustment mechanism. This mechanism dynamically tunes the entropy weight α to regulate policy entropy magnitude, thereby governing the trade-off between exploration and exploitation. While maximizing expected return, the policy entropy is driven toward a target entropy H_{target} . The target entropy is typically set according to the dimensionality of the action space; for instance, for an action space of dimension d , the target entropy can be configured as $-d$. The automatic adjustment mechanism updates the entropy weight by minimizing the following objective function:

$$J(\alpha) = \mathbb{E}_{a_t \sim \pi_{\theta}} \left[-\alpha \log \pi(a_t | s_t; \theta) - \alpha H_{\text{target}} \right] \quad (\text{S25})$$

The entropy weight α is automatically adjusted via gradient descent to drive the actual policy entropy toward the target entropy:

$$\alpha \leftarrow \alpha - \lambda_{\alpha} \nabla_{\alpha} J(\alpha) \quad (\text{S26})$$

Section S3 Results and discussion

Table S2 Vehicle model parameters

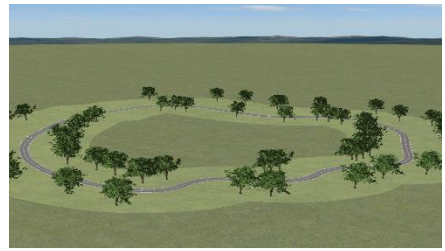
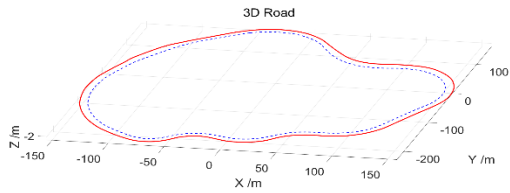
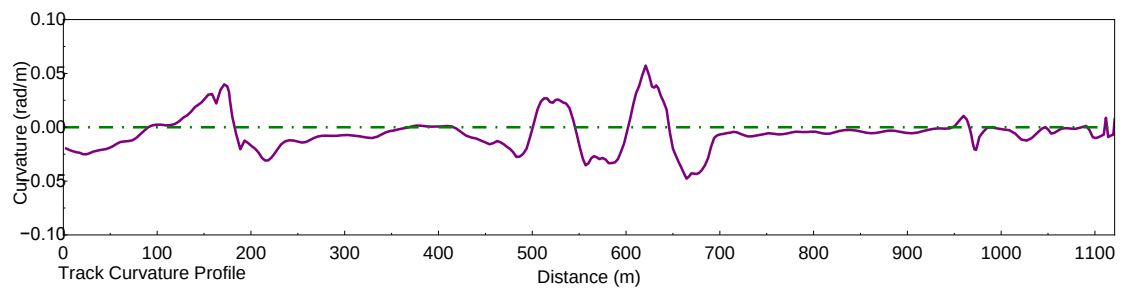
Vehicle Parameters	Description	Value
m	Total vehicle mass	1235.9(kg)
m_s	Sprung mass	1110(kg)
l_f	Distance from CG to front axle	1.040(m)
l_r	Distance from CG to rear axle	1.560(m)
d_f	Front track width	1.480(m)
d_r	Rear track width	1.480(m)
R_e	Tire rolling radius	0.298(m)
h_c	Vehicle CG height	0.540(m)
I_Z	Yaw moment of inertia	1343.1 (kg m ²)
J_w	Wheel moment of inertia	0.60(kg m ²)
C_{yf}	Front tire cornering stiffness	93976.6 (N/rad)
C_{yr}	Rear tire cornering stiffness	138820 (N/rad)
A	Frontal area	1.6(m ²)
C_d	Aerodynamic drag coefficient	0.3
δ_s	Steering ratio	15

Table S3 SAC hyperparameter settings

Hyperparameter	Upper agent value (specific numerical values used for training)	Lower agent value (specific numerical values used for training)
Minibatch size	256	128
Discount factor	0.99	0.95
Actor learning rate	$1e^{-4}$	$1e^{-4}$
Critic learning rate	$3e^{-4}$	$3e^{-4}$
Experience Buffer Length	$4e^5$	$4e^5$
Entropy coefficient learning rate	$1e^{-4}$	$1e^{-4}$
Target Entropy	-1	-1
Initial temperature coefficient	1	1
Gradient clipping threshold	1	0.8
Target smoothing coefficient	$1e^{-3}$	$1e^{-3}$
Optimizer	Adam	Adam

Table S4 Weighting coefficients in reward functions

Symbol	Value
a_1	10
a_2	3.2
a_3	3.2
a_4	0.5
a_5	0.5
a_6	20
b_1	2.3
b_2	2.3
b_3	8
b_4	10

Section S3.1 Case 2: 3D road validation**Fig. S2 Schematic diagram of the 3D validation road geometry**

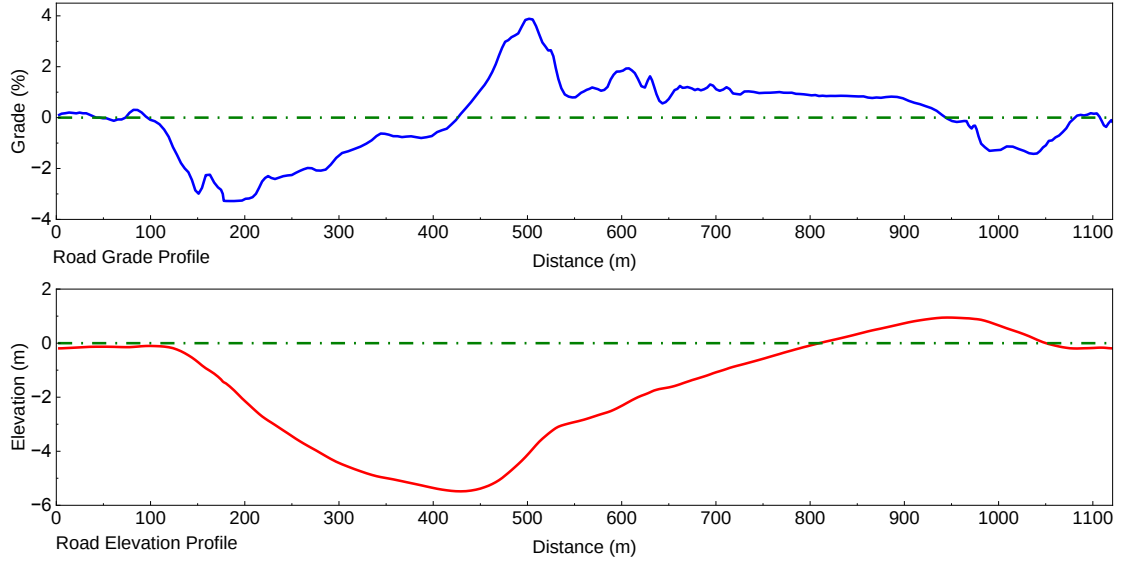


Fig. S3 Road geometric and elevation profiles of the selected validation route

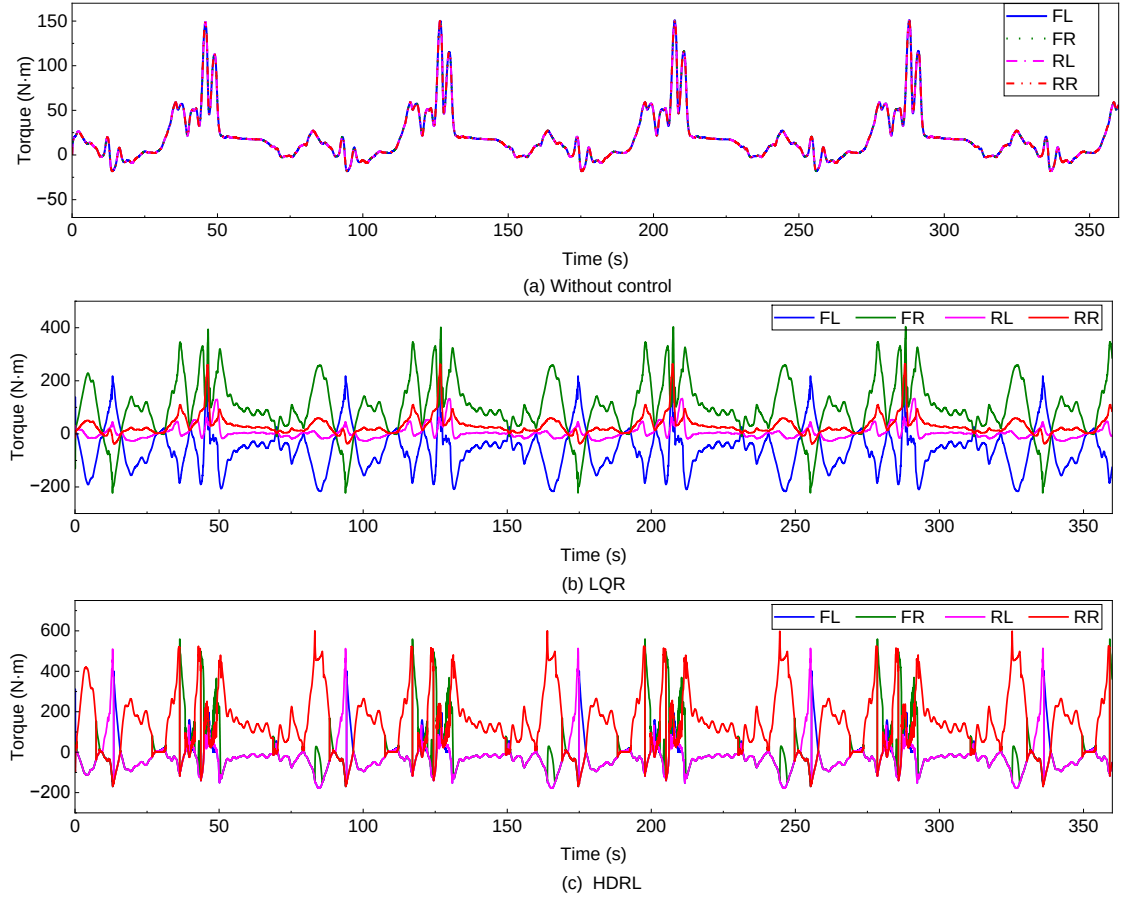


Fig. S4 Wheel torque response in 3D road validation: (a) Without control; (b) LQR; (c)HDRL

Torque distribution profiles are presented in **Fig. S4**. Under LQR control, distinct positive and negative torque adjustments are observed, indicating active yaw moment intervention with torque differentials distributed among wheels according to wheel load proportions. The HDRL method exhibits a similar pattern of yaw moment engagement; however, torque adjustments become notably more frequent during critical maneuvers—such as steering transitions—where rapid regulation of sideslip angle and yaw rate is required to ensure stability. Crucially, unlike the

baseline strategies whose torque outputs predominantly reside in the low-efficiency range of 0–100 N·m, the HDRL controller concentrates torque distribution within the high-efficiency operating region of 200–450 N·m, particularly during periods with modest stability demands. This behavior reflects a dynamic trade-off mechanism inherent to the HDRL framework: stability takes precedence during critical moments requiring immediate intervention, while energy efficiency is prioritized when stability constraints are less stringent. Such distribution characteristics originate from the reward function of the lower-level agent, which explicitly incorporates motor losses and tire slip losses. Through learning, the agent autonomously constrains torque allocation within high-efficiency regions while satisfying stability requirements, thereby manifesting an intrinsic capability for multi-objective coordinated optimization in the torque response morphology.

References

- Sutton RS, Barto AG, 2018. Reinforcement learning: An introduction. translators, MIT press, USA.
- Wei H, Zhang N, Liang J, et al., 2022a. Deep reinforcement learning based direct torque control strategy for distributed drive electric vehicles considering active safety and energy saving performance. *Energy*, 238:121725. <https://doi.org/10.1016/j.energy.2021.121725>
- Wei H, Zhao W, Ai Q, et al., 2022b. Deep reinforcement learning based active safety control for distributed drive electric vehicles. *IET Intelligent Transport Systems*, 16(6):813-824. <https://doi.org/10.1049/itr2.12176>