World Electric Vehicle Journal

MDPI

Article

A Hybrid Learning and Optimization-Based Path Tracking Control Strategy for Intelligent Electric Vehicles

Qiuyan Ge [1], Huajin Chen [1,2,*], Guicheng Liao [1,2,*], Hongxia Zheng [1,2], Qianqiang Lu [3] and Defeng Peng [3]

[1] School of Electronic Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China; gqy010727@163.com (Q.G.); phzheng@gxust.edu.cn (H.Z.)
[2] Guangxi Key Laboratory of Multidimensional Information Fusion for Intelligent Vehicles, Liuzhou 545006, China
[3] Dongfeng Liuzhou Motor Co., Ltd., Liuzhou 545616, China
* Correspondence: huajinchen13@fudan.edu.cn (H.C.); 2001044@gxust.edu.cn (G.L.)

Abstract

This paper proposes a hierarchical control framework designed to enhance the path tracking accuracy of intelligent electric vehicles under diverse operating conditions. For lateral control, an improved model predictive control strategy is developed, utilizing a fuzzy inference system for parameter initialization and a Deep Deterministic Policy Gradient algorithm for online adaptive tuning. For longitudinal control, a proportional–integral–derivative controller is optimized via a hybrid genetic algorithm–particle swarm optimization method. Co-simulations conducted in CarSim/Simulink under straight-line, double-lane-change, and double-sine-wave maneuvers demonstrate that the proposed framework significantly reduces lateral deviation and heading error while ensuring smoother actuator response. Compared to conventional MPC and PID controllers, the proposed method reduces maximum lateral error by over 50% and settling time by 60%, confirming its effectiveness and robustness in complex tracking scenarios.

Keywords: intelligent electric vehicle; hierarchical control; model predictive control; Deep Deterministic Policy Gradient; proportional–integral–derivative

1. Introduction

With the rapid advancement of intelligent transportation systems and new energy vehicle technologies, user expectations regarding driving safety, ride comfort, energy efficiency, and intelligent functionalities have continuously increased [1]. As a result of the deep integration of autonomous driving technologies and electric propulsion systems, intelligent electric vehicles (IEVs) have emerged as a critical platform for advanced driver assistance systems and autonomous driving applications. The overall driving safety and operational efficiency of IEVs largely depend on the perception, decision-making, and control performance of the autonomous driving system [2].

As a key intermediate layer connecting high-level trajectory planning and low-level vehicle actuation, path-following control plays a fundamental role in ensuring trajectory tracking accuracy, vehicle dynamic stability, ride comfort, and energy-efficient operation. In general, the path-following task can be decomposed into lateral and longitudinal control. Lateral control regulates the front-wheel steering angle to minimize lateral and heading deviations with respect to the reference path, while longitudinal control coordinates driving and braking forces to track the desired vehicle speed profile. Traditional control approaches, including proportional–integral–derivative (PID) control [3], preview-based control [4],

Academic Editor: Grzegorz Sierpiński

Received: 23 January 2026
Revised: 12 March 2026
Accepted: 13 March 2026
Published: 18 March 2026

sliding-mode control (SMC) [5], and linear quadratic control (LQC) [6], have been widely adopted due to their simplicity and ease of implementation. However, the practical performance of these methods is often constrained by vehicle parameter uncertainties, strong nonlinear tire dynamics, external disturbances, and the inherent trade-off between model fidelity and real-time computational efficiency. Moreover, IEVs operate under highly variable driving conditions, such as changes in vehicle speed, road curvature, and adhesion levels, which impose stringent requirements on controller adaptability and robustness.

To enhance robustness against such challenges, researchers have explored various robust control strategies. Robust control strategies have therefore attracted significant attention. Robust MPC and sliding-mode control methods have demonstrated strong disturbance rejection capability and stability performance [7–9]. Robust trajectory tracking has also been extended to intelligent connected vehicles. For example, robust output-feedback trajectory tracking for multiple connected vehicles was investigated in [10], while event-triggered robust path tracking considering network delays was proposed in [11]. Model predictive control (MPC) has become one of the most widely used path-following control methods due to its ability to explicitly handle system constraints and predict future vehicle behavior [12,13]. Various MPC-based path tracking strategies have been developed, including multi-constraint MPC [14], MPC-based trajectory tracking frameworks [15], and MPC–PID hybrid control approaches [16]. Adaptive MPC tuning based on particle swarm optimization (PSO) has also demonstrated improved tracking performance [17].

Recent studies have further incorporated artificial intelligence methods into vehicle path tracking control. Reinforcement learning has been combined with MPC to enhance adaptability under uncertain driving environments [18–20]. Fuzzy logic control has also been widely used to improve controller adaptability by introducing rule-based parameter adjustment mechanisms [21–23]. For longitudinal control, although PID controllers remain dominant in engineering applications, evolutionary optimization algorithms such as genetic algorithms (GAs) and particle swarm optimization have been widely used to improve parameter tuning performance [24–27]. Integrated lateral–longitudinal control has gradually become an important research direction for improving overall vehicle motion performance. MPC-based integrated control frameworks have demonstrated effectiveness in improving vehicle stability and coordination [28–30]. Data-driven MPC and neural-network-assisted control strategies have further improved adaptability under complex driving environments [31–33].

Despite these advances, most existing methods treat lateral and longitudinal control separately and rely on static or empirically tuned parameters. Consequently, there is a lack of an integrated framework that can simultaneously optimize both lateral and longitudinal control while adapting online to varying driving conditions. This design paradigm highlights a fundamental trade-off in control systems: robust control methods provide strong theoretical stability guarantees through techniques such as LMI-based $H\infty$ synthesis, but offer limited adaptability; conversely, learning-based methods excel in online adaptability, yet typically lack explicit theoretical stability bounds. To bridge the gap between robustness and adaptability, this paper proposes a hybrid framework that combines fuzzy inference for reasonable initial weights, Deep Deterministic Policy Gradient (DDPG) for online fine-tuning, and GA-PSO for longitudinal PID optimization. In this framework, fuzzy logic provides domain knowledge to bootstrap reinforcement learning, reinforcement learning enables continuous self-improvement, and GA-PSO ensures fast and accurate longitudinal response. This synergy enables the controller to maintain high tracking accuracy and stability without manual re-tuning across diverse scenarios. Table 1 compares representative prior studies with the proposed framework in terms of lateral adaptation, longitudinal optimization, integrated control, and online learning.

Table 1. Comparison of representative prior studies with the proposed framework.

Study	Lateral Control Strategy	Longitudinal Control Strategy	Integrated Control	Online Adaptation	Learning/Optimization Method
Falcone et al. 2007 [13]	Linear time-varying MPC	Not considered	No	No	None
Ji et al. 2017 [14]	MPC	PID	Yes	No	None
Shi et al. 2022 [16]	Improved MPC	Hybrid PID	Yes	No	None
Altan & Hacıoğlu 2022 [17]	Adaptive-weight MPC (PSO offline)	Not considered	No	No	PSO (offline)
Zhang et al. 2023 [18]	RL-MPC (DDPG online tuning)	Not considered	No	Yes	DDPG
Sun et al. 2022 [22]	Fuzzy PID	Fuzzy PID	No	Yes	Fuzzy logic + PSO (offline)
Chen et al. 2024 [19]	MPC + TD3	Not considered	No	Yes	DDPG
Proposed	FL-RL-MPC (fuzzy initialization + DDPG online adaptation)	GA-PSO-PID (offline optimization)	Yes	Yes	Fuzzy + DDPG + GA-PSO

2. Related Work

Path-following control is a core research topic in autonomous driving and intelligent vehicle control, and existing studies can generally be categorized into three main directions: lateral path-following control, longitudinal speed control, and integrated lateral–longitudinal control.

Lateral control aims to guide the vehicle along a reference path by regulating the front-wheel steering angle. Among various approaches, model predictive control has been widely applied due to its advantages in constraint handling and prediction capability [13,21]. To further improve controller adaptability, reinforcement-learning-based MPC methods have been proposed, demonstrating improved performance under uncertain driving environments [18–20]. Intelligent optimization approaches such as particle swarm optimization and genetic algorithms have also been introduced for MPC parameter tuning [17,34]. In addition, hybrid control methods combining MPC with neural networks and sliding-mode control have demonstrated improved robustness and tracking accuracy under complex conditions [35–37]. Fuzzy control methods have also been applied to improve adaptive tracking performance through rule-based parameter adjustment [22,23]. Overall, these studies indicate a clear trend toward integrating MPC with learning-based and intelligent optimization techniques to enhance robustness and adaptability in lateral path-following control.

Longitudinal control focuses on regulating vehicle speed through the coordination of driving and braking forces. Due to its simple structure and ease of implementation, the PID controllers remain one of the most widely used approaches due to their simple structure and engineering practicality [3,24]. However, their performance is highly dependent on parameter tuning. To address this limitation, MPC-based longitudinal speed control methods have been developed for constrained vehicle systems [14]. Intelligent tuning strategies including fuzzy PID control and evolutionary optimization algorithms have also been introduced to improve controller adaptability under varying driving conditions [22–27]. These approaches highlight the effectiveness of hierarchical and hybrid structures in improving longitudinal control performance under time-varying conditions.

To further improve trajectory tracking accuracy and overall vehicle performance, integrated lateral–longitudinal control strategies have attracted increasing attention. By explicitly considering the coupling between lateral and longitudinal vehicle dynamics, such methods enable coordinated control of steering and speed. MPC-based integrated control frameworks have demonstrated strong performance in improving vehicle motion coordination [28–30]. With the development of data-driven control methods, adaptive MPC and neural-network-assisted control approaches have been proposed to further improve control performance [31,32]. Reinforcement-learning-integrated MPC methods have also shown improved adaptability in complex driving scenarios such as distributed-drive electric vehicles [19,33]. In addition, robust control strategies considering disturbances and cybersecurity issues have been investigated to enhance system reliability [7,38].

Despite the substantial progress achieved, most existing control strategies rely on statically predefined or empirically tuned parameters, which limits their ability to maintain optimal performance under time-varying and uncertain operating conditions. Consequently, the development of intelligent composite control frameworks with online adaptive parameter regulation and multi-objective optimization capabilities remains a critical research direction for enhancing the comprehensive performance of path-following control in intelligent electric vehicles.

3. Path Tracking Control Framework

Path tracking is a fundamental functionality for intelligent electric vehicles in autonomous driving. Conventional single-algorithm approaches often face challenges in simultaneously achieving high tracking accuracy and smooth control under complex driving conditions. This study proposes a coordinated lateral–longitudinal path tracking control framework that integrates model predictive control, fuzzy inference, reinforcement learning, and intelligent optimization. The framework employs separate lateral and longitudinal controllers to enhance robustness and precision in dynamic environments.

3.1. Overall Architecture

The proposed lateral–longitudinal cooperative path tracking control framework is illustrated in Figure 1. Lateral and longitudinal control modules form the core of this cooperative system. By leveraging road curvature and friction information to determine suitable longitudinal speeds, the framework enables flexible coordination between lateral and longitudinal vehicle motions. This hierarchical decoupled structure is deliberately chosen based on two considerations. First, an integrated MPC that simultaneously optimizes steering and acceleration/braking commands would require a higher-dimensional prediction model and control horizon, significantly increasing the online QP problem size. The decoupled approach keeps the lateral MPC tractable and the longitudinal controller lightweight, enabling real-time execution on embedded platforms. Second, the coupling between lateral and longitudinal dynamics is not ignored but addressed at two levels: the reference speed profile provided to the longitudinal controller is adapted based on road curvature and friction (feedforward compensation), and the lateral MPC's prediction model includes longitudinal velocity as a time-varying parameter, while the DDPG-based weight adaptation can indirectly compensate for dynamic coupling effects when tracking errors arise from speed variations.

In this architecture, the upper-level planning module supplies reference trajectories, road curvature, and friction coefficient information. Lateral motion is regulated by a fuzzy logic–reinforcement learning-enhanced MPC (FL-RL-MPC). The weighting matrices of the MPC are first initialized via a fuzzy inference system and then adaptively adjusted online using the Deep Deterministic Policy Gradient algorithm. This ensures precise path tracking and control stability under variable conditions. Longitudinal motion is governed by a PID controller, with gain parameters optimized online through a hybrid genetic algorithm–particle swarm optimization approach. This configuration enables rapid convergence to target speeds while maintaining smooth acceleration and deceleration. The lateral controller outputs front-wheel steering angles, whereas the longitudinal controller generates drive and brake commands. Together, they allow synchronized tracking of both the reference path and target speed within a unified framework, achieving coordinated control of steering and velocity. The framework synergistically integrates knowledge-driven and data-driven methods: fuzzy logic injects expert knowledge to provide a safe baseline, DDPG enables continuous online self-improvement, and GA-PSO searches for near-optimal PID gains offline. Unlike conventional MPC with fixed weights, the FL-RL-MPC adjusts its weighting matrices in real time according to driving conditions, enhancing tracking accuracy and robustness without manual re-calibration. The GA-PSO-PID longitudinal controller achieves fast convergence and superior tracking performance with negligible online computational burden.

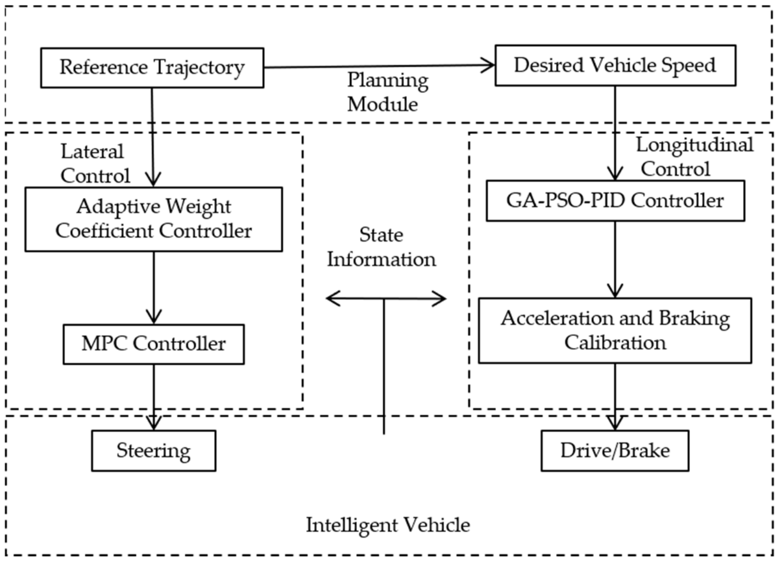


Figure 1. Overall framework of the path tracking control system.

By combining model predictive control, fuzzy reasoning, reinforcement learning, and evolutionary optimization, the proposed method can dynamically adjust lateral and longitudinal control strategies in complex scenarios, thereby enhancing both path tracking performance and vehicle dynamic stability. The detailed design of the lateral controller is presented in the following subsection.

3.2. Lateral Control

Effective lateral control requires accurate modeling of vehicle dynamic characteristics while maintaining computational efficiency [14]. In this study, a two-degree-of-freedom (2-DOF) vehicle dynamics model is adopted due to its relatively simple structure and ability to capture key lateral motion characteristics. This linearized 2-DOF model is widely adopted for vehicle lateral dynamics control and has been validated under small-slip-angle assumptions [14,39]. It forms the foundation for controller design, aiming to improve both stability and real-time performance of path tracking, as shown in Figure 2. The vehicle dynamics are described using an inertial coordinate system (XOY, fixed to the ground) and a body-fixed coordinate system (xoy). Based on Newton's second law, the force equilibrium equations are expressed as

$$\begin{aligned} m\ddot{x} &= m\dot{y}\dot{\varphi} + F_{xf}\cos(\delta_f) - F_{xr}\sin(\delta_f) \\ m\ddot{y} &= -m\dot{x}\dot{\varphi} + F_{yf}\cos(\delta_f) + F_{yr}\sin(\delta_f) \\ I_z\ddot{\varphi} &= aF_{yf}\cos(\delta_f) - bF_{yr}, \end{aligned} \tag{1}$$

where m is the vehicle mass; x and y are longitudinal and lateral displacements; $\dot{x}$ and $\dot{y}$ are longitudinal and lateral velocities; $\ddot{x}$ and $\ddot{y}$ denote the corresponding accelerations; φ and $\dot{\varphi}$ represent the yaw angle and yaw rate, respectively; δ_f is the front-wheel steering angle; I_z is the yaw moment of inertia; F_{yf} and F_{yr} are the lateral tire forces at the front and rear wheels; and a and b are the distances from the front and rear axles to the center of mass.

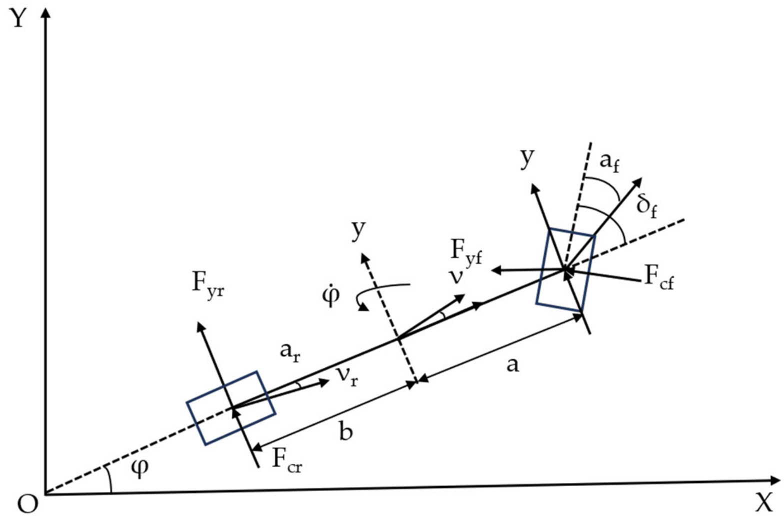


Figure 2. Vehicle dynamics model.

Under the assumptions of small front-wheel steering angle (<10°) small tire slip angles ($|\alpha| < 5°$) and constant longitudinal velocity, the 2-DOF equations simplify to

$$\begin{aligned} \dot{v}_y &= -v_x\dot{\varphi} + \tfrac{1}{m}(-F_{yf} + F_{yr}) \\ \ddot{\varphi} &= \tfrac{1}{I_z}(aF_{yf} - bF_{yr}). \end{aligned} \tag{2}$$

Tire forces are linearized for small slip angles [37]:

$$\begin{aligned} F_{yf} &= F_{cf} = C_{cf}\alpha_f = C_{cf}\left(\tfrac{\dot{y}+a\dot{\varphi}}{x} - \delta_f\right) \\ F_{yr} &= F_{cr} = C_{cf}\left(\tfrac{\dot{y}-b\dot{\varphi}}{\dot{x}}\right). \end{aligned} \tag{3}$$

In these equations, C_{cf} and C_{cr} denote the front and rear cornering stiffnesses; α_f and α_r denote the front- and rear-tire slip angles. Under the small-slip-angle assumption and constant longitudinal velocity, the linear tire model with constant cornering stiffness is a widely accepted approximation for controller design [14,39]. Consequently, the state-space form is obtained as

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}u, \tag{4}$$

where the state vector is

$$\mathbf{x} = \left[v_y, \dot{\varphi}\right]^T, \tag{5}$$

and the control input is $u = \delta_f$, $\mathbf{A} = \begin{bmatrix} \frac{C_{cf}+C_{cr}}{mv_x} & \frac{aC_{cf}-bC_{cr}}{mv_x} - v_x \\ \frac{aC_{cf}-bC_{cr}}{I_z v_x} & \frac{a^2C_{cf}+b^2C_{cr}}{I_z v_x} \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} -\frac{C_{cf}}{m} \\ -\frac{aC_{cf}}{I_z} \end{bmatrix}$. The nominal values of the vehicle parameters used in this paper are listed in Table 2. They correspond to a generic C Class hatchback in CarSim.

Table 2. Vehicle parameters.

Symbol	Parameter	Value	Unit
m	Vehicle mass	1270	kg
I_z	Yaw moment of inertia	1536.7	kg·m^2
a	Distance from CG to front axle	1.015	m
b	Distance from CG to rear axle	1.895	m
C_{cf}	Front-tire cornering stiffness	148,000	N/rad
C_{cr}	Rear-tire cornering stiffness	132,000	N/rad
μ	Road adhesion coefficient (nominal)	0.9	-
v_x	Initial longitudinal velocity	60	km/h

Parameters are derived from the default C Class model in CarSim, with cornering stiffness values representing the total stiffness for the front and rear axles.

For more accurate description of path tracking errors, the vehicle model is transformed into the Frenet coordinate system (a moving local frame attached to the reference path), as shown in Figure 3, where the tangential axis aligns with the path direction and the normal axis is perpendicular to it [40]. The lateral error and the heading error are defined as

$$\begin{aligned} e_d &= d \\ e_\varphi &= \theta - \theta_h. \end{aligned} \tag{6}$$

Additionally, the path heading rate $\dot{\theta}_h$ is introduced as an external disturbance term to enhance the model's adaptability to varying trajectories. This yields the augmented state-space equation

$$\dot{\mathbf{X}} = \mathbf{A}X + \mathbf{B}u + \mathbf{C}\dot{\theta}_h, \tag{7}$$

where the state vector is

$$\mathbf{X} = \begin{bmatrix} \dot{e}_d & \ddot{e}_d & \dot{e}_\varphi & \ddot{e}_\varphi \end{bmatrix}^T, \tag{8}$$

the input is $u = \delta_f$, and the matrices **A**, **B**, **C** are given by

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \frac{C_{cf}+C_{cr}}{mv_x} & \frac{-C_{cf}+C_{cr}}{mv_x} & \frac{aC_{cf}-bC_{cr}}{mv_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{aC_{cf}-bC_{cr}}{I_z v_x} & -\frac{aC_{cf}-bC_{cr}}{I_z} & \frac{a^2C_{cf}+b^2C_{cr}}{I_z v_x} \end{bmatrix}, \mathbf{B} = \begin{bmatrix} 0 \\ 0 \\ -\frac{C_{cf}}{m} \\ -\frac{aC_{cf}}{I_z} \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 0 \\ \frac{aC_{cf}-bC_{cr}}{mv_x} - V_x \\ 0 \\ \frac{a^2C_{cf}+b^2C_{cr}}{I_z v_x} \end{bmatrix}.$$

The kinematic relationship between the lateral error and the reference path's projection point is given by

$$\begin{aligned} \dot{d} &= |v| \sin(\theta - \theta_h) \\ \dot{s} &= \frac{|v| \cos(\theta - \theta_h)}{1 - \kappa_h d}. \end{aligned} \tag{9}$$

In these equations, d denotes the lateral error; v is the vehicle velocity; n_h and τ_h are the unit normal and tangential vectors at the projection point; x_h and y_h are the coordinates of the projection point; θ and θ_h are the vehicle and path heading angles, respectively; and κ_h is the curvature at the projection point.

Based on the derived state-space model and its Frenet frame representation, an MPC-based lateral controller is constructed. MPC leverages the rolling-horizon prediction of the vehicle dynamics model and incorporates constraint optimization algorithms to solve future control sequences in real time. This approach outputs optimal control actions while satisfying system constraints, offering advantages such as high control accuracy and the ability to handle multi-variable constrained systems. Figure 4 illustrates the designed MPC lateral control flowchart.

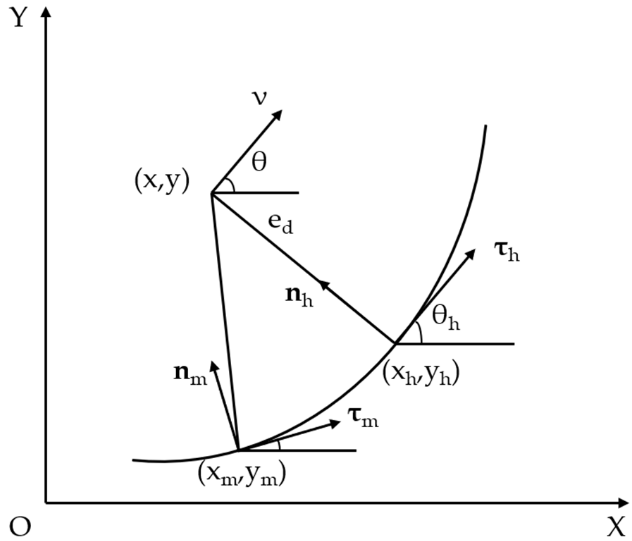


Figure 3. Vehicle error model under the Frenet coordinate system.

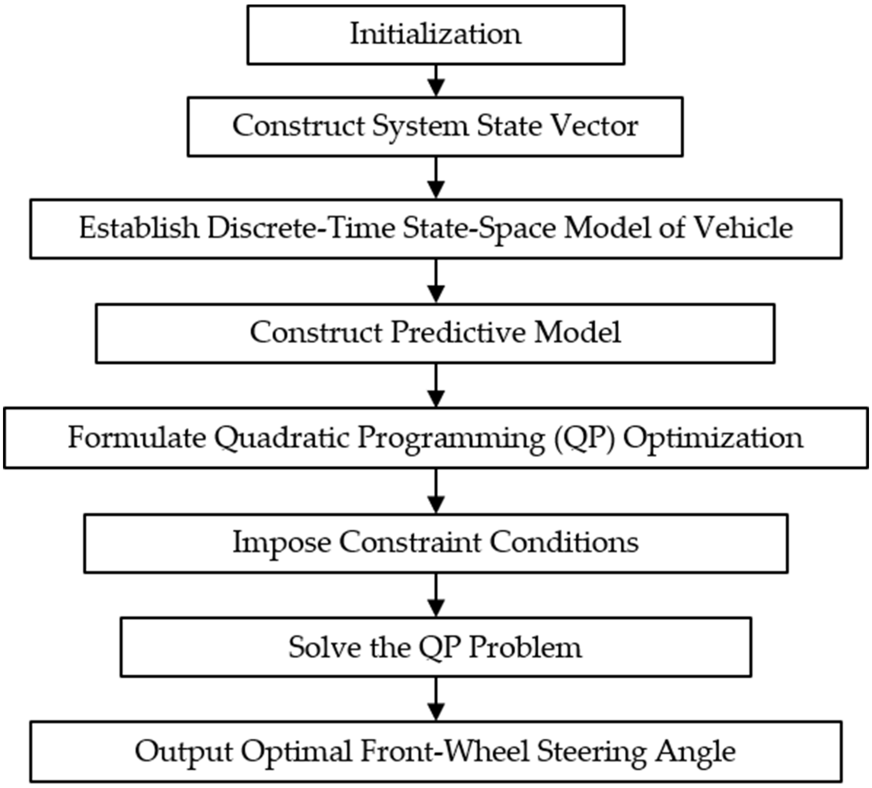


Figure 4. Flowchart of the MPC-based lateral control scheme.

For implementing MPC optimization in discrete time, the continuous state-space model is discretized using forward and midpoint Euler methods, resulting in the discrete augmented state-space equation

$$\mathbf{X}(k+1) = \overline{\mathbf{A}}\mathbf{X}(k) + \overline{\mathbf{B}}\Delta u(k) + \overline{\mathbf{C}}\dot{\theta}_h(k), \tag{10}$$

where the matrices are $\overline{\mathbf{A}} = (\mathbf{I} - \mathbf{A}\frac{T}{2})^{-1}(\mathbf{I} + \mathbf{A}\frac{T}{2})^{-1}$, $\overline{\mathbf{B}} = \mathbf{B}T$, and $\overline{\mathbf{C}} = \mathbf{C}T$, T is the sampling period (20 ms), and $\dot{\theta}_h(k)$ represents the yaw rate of the reference-path curvature. To impose constraints on the control increment and enhance the controllability of the prediction model, a new state vector is introduced as

$$\xi(k) = \begin{bmatrix} \mathbf{X}(k) \\ u(k-1) \end{bmatrix}, \tag{11}$$

leading to the following state-space equation:

$$\begin{aligned} &\xi(k+1) = \widetilde{A}\xi(k) + \widetilde{\mathbf{B}}\Delta\mathbf{u}(k) + \widetilde{\mathbf{C}}\dot{\theta}_h(k) \\ &\gamma(k) = \widetilde{\mathbf{D}}\xi(k), \end{aligned} \tag{12}$$

In these equations, $\widetilde{\mathbf{A}} = \begin{bmatrix} \bar{\mathbf{A}} & \bar{\mathbf{B}} \\ 0 & \mathbf{I}_{1\times1} \end{bmatrix}$, $\widetilde{\mathbf{B}} = \begin{bmatrix} \bar{\mathbf{B}} \\ \mathbf{I}_{1\times1} \end{bmatrix}$, $\widetilde{\mathbf{C}} = \begin{bmatrix} \bar{\mathbf{C}} \\ 0 \end{bmatrix}$, and $\widetilde{\mathbf{D}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$; the system output sequence over the prediction horizon is expressed as

$$\mathbf{Y} = \mathbf{\Phi}\xi(k) + \mathbf{\Psi}\Delta\mathbf{U} + \varepsilon\mathbf{\Theta}, \tag{13}$$

where $\mathbf{Y} = \begin{bmatrix} \mathbf{Y}(k) & \mathbf{Y}(k+1) & \dots & \mathbf{Y}(k+N_p) \end{bmatrix}^T$, $\mathbf{\Phi} = \begin{bmatrix} \widetilde{\mathbf{D}}\widetilde{\mathbf{A}} & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^2 & \dots & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p} \end{bmatrix}^T$, $\Delta\mathbf{U} = \begin{bmatrix} \Delta u(k) & \Delta u(k+1) & \dots & \Delta u(k+N_c-1) \end{bmatrix}^T$, $\mathbf{\Theta} = \begin{bmatrix} \dot{\theta}_h(k) & \dot{\theta}_h(k+1) & \dots & \dot{\theta}_h(k+N_c-1) \end{bmatrix}$,

$\mathbf{\Psi} = \begin{bmatrix} \widetilde{\mathbf{D}}\widetilde{B} & 0 & \dots & 0 \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}\widetilde{B} & \widetilde{\mathbf{D}}\widetilde{B} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_c-1}\widetilde{B} & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_c-2}\widetilde{B} & \dots & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{0}\widetilde{B} \\ \vdots & \vdots & \vdots & \vdots \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-1}\widetilde{B} & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-2}\widetilde{B} & \dots & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-N_c}\widetilde{B} \end{bmatrix}$, $\varepsilon = \begin{bmatrix} \widetilde{\mathbf{D}}\widetilde{\mathbf{C}} & 0 & \dots & 0 \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}\widetilde{\mathbf{C}} & \widetilde{\mathbf{D}}\widetilde{\mathbf{C}} & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_c-1}\widetilde{\mathbf{C}} & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_c-2}\widetilde{\mathbf{C}} & \dots & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{0}\widetilde{\mathbf{C}} \\ \vdots & \vdots & \vdots & \vdots \\ \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-1}\widetilde{\mathbf{C}} & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-2}\widetilde{\mathbf{C}} & \dots & \widetilde{\mathbf{D}}\widetilde{\mathbf{A}}^{N_p-N_c}\widetilde{\mathbf{C}} \end{bmatrix}$,

and N_p and N_c denote the prediction horizon and control horizon lengths, respectively. To achieve optimal tracking performance, the MPC cost function is designed as

$$J = \sum_{i=1}^{N_p} \|\gamma(k+i) - \gamma_h(k+i)\|_{\mathbf{Q}}^2 + \sum_{i=1}^{N_c} \|\Delta u(k+i)\|_{\mathbf{R}}^2 + \rho\sigma^2, \tag{14}$$

where $\mathbf{Q} = \mathrm{diag}(W_{e_d}, W_{\dot{e}_d}, W_{e_\varphi}, W_{\dot{e}_\varphi})$ is a diagonal weighting matrix. Larger values in $\mathbf{Q}$ assign higher priority for the system to converge the corresponding error (lateral error, lateral error rate, heading error, heading error rate), although excessively large weights may cause control oscillations and require calibration through experiments under different driving conditions. $\mathbf{R}$ is the weighting matrix for control increments, used to suppress abrupt changes in the control input. ρ is a weight coefficient, and σ is a slack factor. Physical limitations of the vehicle actuators are considered by imposing constraints on the control input and its increment:

$$\begin{aligned} &\mathbf{U}_{min} \le \mathbf{A_I}\Delta\mathbf{U} \le \mathbf{U}_{max} \\ &\mathbf{U}_{min} \le \Delta\mathbf{U} \le \Delta\mathbf{U}_{max}, \end{aligned} \tag{15}$$

where $\mathbf{U}_{min}$ and $\mathbf{U}_{max}$ represent the allowable range of the steering command, and the second pair defines how quickly that command is allowed to vary. The matrix $\mathbf{A_I}$ is a block-diagonal form $\mathbf{A_I} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}_{N_c\times N_c} \otimes I_{2\times2}$; the MPC optimization is handled by a quadratic programming solver. With $\boldsymbol{\eta} = \begin{bmatrix} 0 & 0 & \arctan\left(\frac{v_y}{v_x}\right) & 0 \end{bmatrix}^T$, the quadratic programming form is expressed as

$$J = \begin{bmatrix} \Delta u \\ \sigma \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Psi}^T \mathbf{Q} \Psi & 0 \\ 0 & \rho \end{bmatrix} \begin{bmatrix} \Delta u \\ \sigma \end{bmatrix} + \begin{bmatrix} 2\boldsymbol{\Psi}^T \mathbf{Q} \\ \sigma \end{bmatrix}^T \begin{bmatrix} \Delta u \\ \sigma \end{bmatrix}. \tag{16}$$

To further enhance the performance of model predictive control under complex driving conditions, fuzzy logic control is introduced for online adjustment of the MPC cost function's weight matrix **Q**. The fuzzy inference system takes the vehicle's current lateral error e_d, heading error e_φ, and their rates of change as inputs. Through membership functions and a fuzzy rule base, it outputs weight gain factors, enabling adaptive adjustment of controller response across different scenarios. The designs of the fuzzy variables and their corresponding membership functions are summarized in Table 3. Gaussian membership functions are adopted due to their smoothness and robustness in representing nonlinear relationships. The universes of discourse are defined in Table 3, which lists the settings for input and output variables, and the rule design is detailed in Table 4. Gaussian membership functions are employed for all four input variables: e_d and $\dot{e}_d$ over the range [−0.5, 0.5] m and m/s respectively; e_φ and $\dot{e}_\varphi$ over the range [−0.523, 0.523] rad and rad/s respectively. The rule base is fully populated with linguistic terms (NM, NS, Z, PS, PM) and corresponding output fuzzy sets (SS, S, M, L, LL). The online calculation process for the fuzzy-adjusted weights proceeds as follows in Table 5. The Gaussian membership function is defined as

$$\mu(x) = \exp\left(-\frac{(x-c)^2}{2\sigma 2}\right), \tag{17}$$

where c represents the center of the membership function and σ determines its width. The parameter values are selected to ensure sufficient overlap between adjacent fuzzy sets.

Table 3. Design of input/output variables and membership functions.

Variable	Linguistic Sets	Value Range	Membership Type
Lateral deviation e_d	{NM, NS, Z, PS, PM}	[−0.5, 0.5]	Gaussian
Rate of lateral deviation $\dot{e}_d$	{NM, NS, Z, PS, PM}	[−0.5, 0.5]	Gaussian
Heading deviation e_φ	{NM, NS, Z, PS, PM}	[−0.523, 0.523]	Gaussian
Rate of heading deviation $\dot{e}_\varphi$	{NM, NS, Z, PS, PM}	[−0.523, 0.523]	Gaussian
Weight output W_{e_d} and W_{e_φ}	{SS, S, M, L, LL}	[0, 100]	Trigonometric

Table 4. Rule base for fuzzy controller design.

	NM	NS	Z	PS	PM
NM	LL	L	M	L	LL
NS	LL	L	S	L	LL
Z	L	M	SS	M	L
PS	LL	L	S	L	LL
PM	LL	L	M	L	LL

The same rule base applies to both weight outputs W_{e_d} and W_{e_φ}.

Although the fuzzy system can provide a reasonable initial estimation for MPC weights, its membership function parameters and rule table rely heavily on expert experience, which may not guarantee optimality in complex, multi-scenario environments. Therefore, this paper further introduces deep reinforcement learning for online self-optimization of the fuzzy-generated weights; the entire framework is shown in Figure 5. Specifically, the Deep Deterministic Policy Gradient algorithm is employed to construct an actor–critic learning framework for dynamically correcting the fuzzy weight outputs, and three core mechanisms are integrated to ensure the efficiency and reliability of the reinforcement learning module [41–43]. The DDPG-based self-learning optimization procedure incorporates several key design elements to ensure effective online adaptation. Table 6 outlines the detailed algorithmic steps, while the following aspects merit specific emphasis:

Table 5. Online calculation process for fuzzy-adjusted MPC weights.

Online Fuzzy-Based Adjustment of MPC Weight Factors	
1: # Input: lateral and heading tracking errors $e_d, \dot{e}_d, e_\varphi, \dot{e}_\varphi$	▷ define input variables for fuzzy inference
2: # Output: fuzzy-adjusted MPC weight factors W_{e_d} and W_{e_φ}	▷ define output variables for MPC weight adjustment
3: Acquire current vehicle state	▷ get current lateral and heading errors
4: Fuzzification	▷ convert crisp inputs to fuzzy values
5: Map $e_d, \dot{e}_d, e_\varphi, \dot{e}_\varphi$ into corresponding fuzzy sets	▷ calculate membership degrees for each input
6: Fuzzy inference	▷ apply fuzzy rules to determine output fuzzy sets
7: Evaluate fuzzy rules to obtain fuzzy outputs	▷ apply rule base to determine output membership degrees
8: Obtain $\dot{W}_{e_{d\,fuzzy}}$ and $\dot{W}_{e_{\varphi\,fuzzy}}$	▷ intermediate fuzzy results before defuzzification
9: Defuzzification	▷ convert fuzzy outputs to crisp values
10: $W_{e_d} = \text{Defuzzy}(\dot{W}_{e_{d\,fuzzy}})$	▷ convert fuzzy output to crisp value using centroid method
11: $W_{e_\varphi} = \text{Defuzzy}(\dot{W}_{e_{\varphi\,fuzzy}})$	▷ obtain crisp MPC weight factors
12: Update MPC weight matrix:	▷ construct new weight matrix for MPC cost function
13: $\mathbf{Q} = \text{diag}(W_{e_d}, W_{\dot{e}_d}, W_{e_\varphi}, W_{\dot{e}_\varphi})$	▷ construct diagonal state weighting matrix for MPC cost function
14: Return W_{e_d} and W_{e_φ}	▷ output weights for MPC optimization

Table 6. DDPG-based self-learning optimization procedure for fuzzy-generated MPC weights.

DDPG-Based Self-Learning Optimization Procedure for Fuzzy-Generated MPC Weights	
1: # Input: Current fuzzy weight W_{fuzzy}, vehicle state X	▷ define input variables
2: # Output: Optimized MPC weight matrix **Q**	▷ define output variables
3: Initialize actor–critic networks (actor θ_a, critic θ_c)	▷ initialize policy and value networks
4: for each control timestep do	▷ execute online adaptation loop
5: $W_{fuzzy} = \text{Fuzzy}(X)$	▷ obtain baseline MPC weights from fuzzy inference system
6: $\Delta W = \text{Actor}(X)$	▷ generate weight correction based on current state
7: $W_{new} = W_{fuzzy} + \Delta W$	▷ combine fuzzy baseline with RL correction
8: $\mathbf{Q} = \text{diag}(W_{new})$	▷ construct MPC weight matrix from updated weights
9: $u = \text{SloveMPC}(\mathbf{Q})$	▷ solve quadratic program to obtain optimal steering angle
10: Apply u to vehicle	▷ execute control command in simulation/environment
11: $r = \text{compute reward } (e_d, \dot{e}_d, e_\varphi, \dot{e}_\varphi)$	▷ calculate reward using weighted quadratic errors
12: Store transition $(X, \Delta W, r, X_{next})$	▷ save experience tuple for later training
13: Sample mini-batch $(N = 64)$ from buffer	▷ randomly sample batch for training
14: Update critic network using TD loss $(\gamma = 0.99)$	▷ minimize mean-squared error between target **Q**-value and current **Q**-value, with discount factor
15: Update actor network using policy gradient	▷ maximize expected **Q**-value via DDPG
16: Soft update target networks $(\tau = 0.005)$	▷ gradually update target networks: $\theta' \leftarrow \tau\theta + (1-\tau)\theta'$
17: Return **Q**	▷ output optimized MPC weight matrix for current cycle

The actor and critic learning rates are set to 0.0001 and 0.001, respectively.

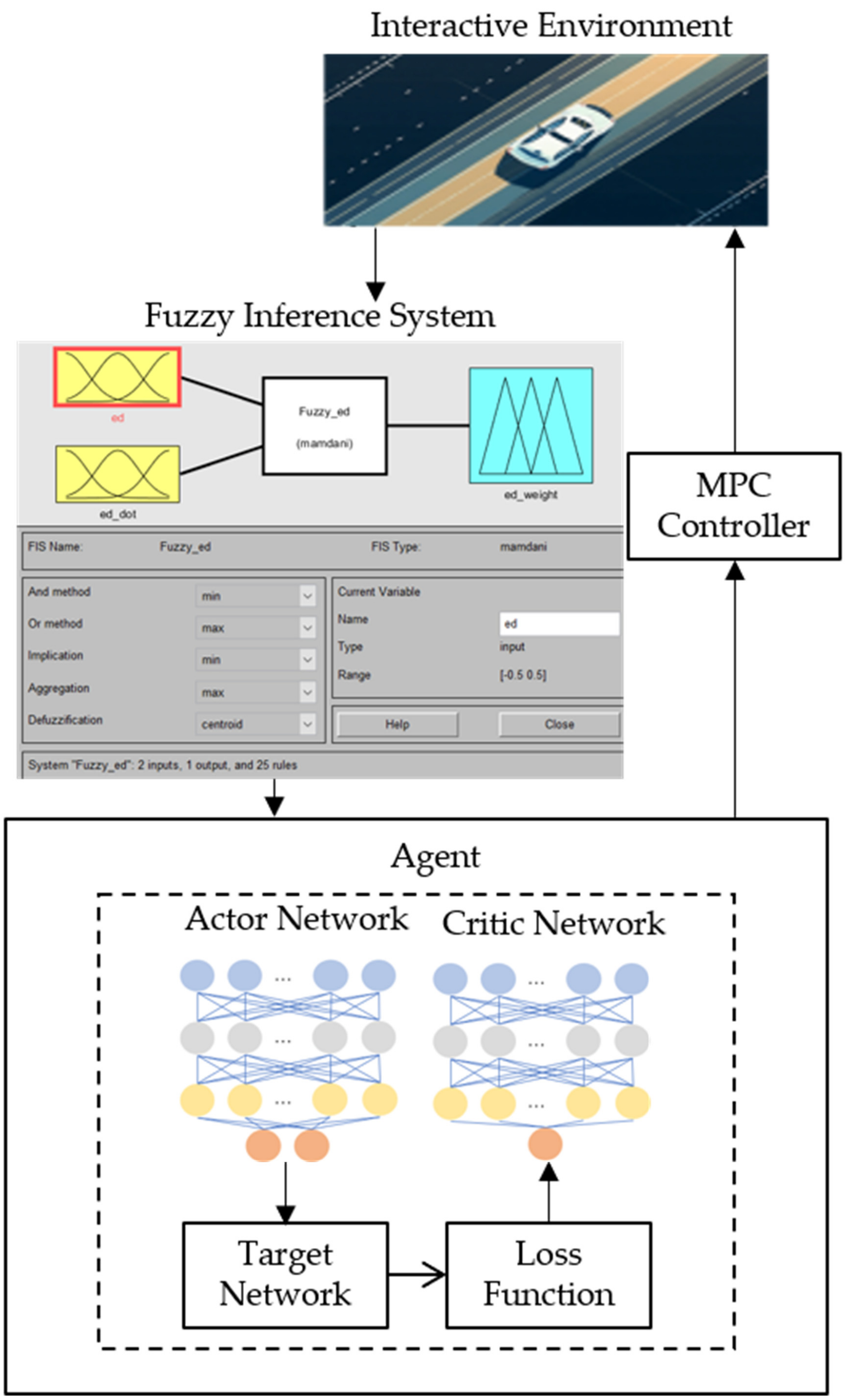


Figure 5. Structure of the FL-RL-MPC lateral control scheme.

(1) Reward function design: The reward is defined as

$$r = -(w_1 e_d^2 + w_2 \dot{e}_d^2 + w_3 e_\varphi^2 + w_4 \dot{e}_\varphi^2), \tag{18}$$

where the weights $w_1 = 1.0, w_2 = 0.5, w_3 = 1.5, w_4 = 0.8$ are chosen to balance tracking accuracy and control smoothness. This quadratic formulation penalizes both tracking errors and their rates of change, reflecting a common objective in optimal control. Sensitivity analysis confirms that moderate variations (±20%) in these weights do not significantly affect the learned policy's final performance, indicating robustness to weight selection.

(2) State and action spaces: The state space is defined as

$$S = \left[e_d, \dot{e}_d, e_\varphi, \dot{e}_\varphi, v_x, \kappa\right], \tag{19}$$

encompassing lateral and heading errors, their rates, longitudinal velocity, and path curvature. This representation captures all information necessary for adaptive weight adjustment. The action space is defined as

$$A = [\Delta W_{e_d}, \Delta W_{e_\varphi}], \tag{20}$$

which represents the corrections applied to the fuzzy-generated weights, with each component bounded in $[-0.5, 0.5]$ to prevent aggressive weight changes that could compromise stability.

(3) Training configuration: The agent is trained over 500 episodes, each lasting 15 seconds with a 20 ms sampling period. The training environment employs randomly generated continuous-curvature paths with varying curvature (± 0.02 to ± 0.08 m^{-1}) and target speeds (30–80 km/h), while the road adhesion coefficient is randomly varied between 0.5 and 0.9 to enhance robustness. The learning curve, shown in Figure 6, demonstrates stable convergence after approximately 300 episodes, with low variance in episode rewards.

(4) Generalization capability: It is worth noting that the DDPG agent is trained exclusively on randomly generated paths with continuously varying curvature, speed, and adhesion coefficient. The strong performance of the trained agent on unseen trajectories—including standard maneuvers such as double lane change and double sine wave—demonstrates its ability to generalize beyond the training distribution.

(5) Stability and feasibility guarantee: The proposed framework incorporates several mechanisms that ensure reliable online operation by design. Recursive feasibility is theoretically guaranteed through the inclusion of a slack variable σ and soft output constraints in the MPC formulation (see Equation (15)). These elements ensure that the quadratic programming solver always returns a feasible solution at each time step, provided that the original problem is feasible. Since the DDPG module only modifies the diagonal weight matrix $\mathbf{Q}$ and does not alter the prediction model, the hard constraints on the steering angle and its increment, or the slack variable mechanism, this feasibility property remains intact during online adaptation. Constraint satisfaction is explicitly enforced by the same hard constraints, while the DDPG-generated weight corrections are clipped to the bounded action space $[-0.5, 0.5]$ as specified in design element (2). This prevents abrupt weight changes that could otherwise lead to constraint violations or compromise stability. The training configuration itself is designed to promote stability: the agent converges to a stable policy with low-variance episode rewards, as evidenced by the learning curve in Figure 6. Furthermore, extensive empirical validation under diverse operating conditions—including varying road adhesion, sensor noise, and model uncertainties—confirms that the proposed controller consistently satisfies constraints and maintains stable tracking performance.

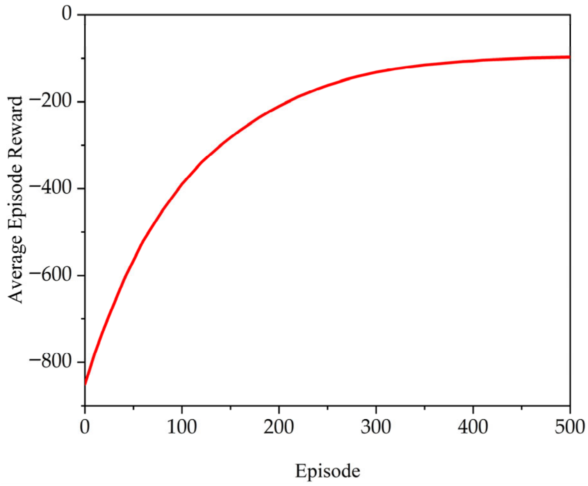


Figure 6. DDPG learning curve: average episode reward over 500 training episodes.

Within the overall control loop, the fuzzy inference system and the DDPG optimization module together constitute the MPC weight adaptation unit. In each sampling period, the system first generates a fuzzy weight estimate based on the vehicle's current error state. Subsequently, the actor network outputs a weight correction, which is used to construct the updated MPC weight matrix. This matrix is embedded into the MPC cost function, and the optimal steering control input is solved via receding horizon optimization. Following this, the system calculates the reward based on the vehicle's response and updates the policy networks, enabling continuous self-learning optimization of the weights. This integrated mechanism—combining "Fuzzy Prior Knowledge," "Reinforcement Learning Self-Evolution," and "MPC Optimization Execution"—empowers the controller to achieve high-precision and highly robust lateral path tracking control under complex driving conditions.

3.3. Longitudinal Control

To achieve vehicle speed tracking in complex dynamic scenarios, this study designs a longitudinal speed controller based on a GA-PSO-optimized PID structure. The controller adopts a hierarchical architecture, as illustrated in Figure 7. The upper-level controller generates the desired acceleration command based on the speed tracking error, while utilizing a genetic algorithm–particle swarm optimization hybrid algorithm for PID parameter tuning. The lower-level controller translates this desired acceleration into actual throttle or brake signals by incorporating an electric motor drive/braking model and an acceleration–brake calibration map, thereby executing longitudinal speed tracking.

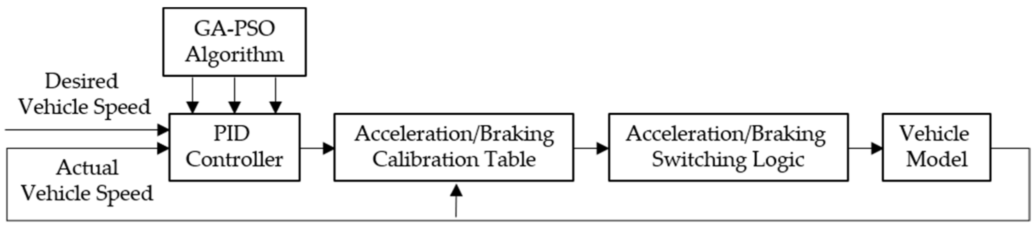


Figure 7. Overall framework of longitudinal speed tracking control.

The PID output of the upper-level controller is given by

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{de(t)}{dt}, \quad (21)$$

where $u(t)$ is the output variable of the PID controller; $e(t)$ is the input error variable; and k_p, k_i and k_d are the proportional, integral, and derivative coefficients, respectively. Conventional PID parameter tuning methods often suffer from slow convergence and a tendency to become trapped in local optima when dealing with multi-objective optimization problems [22,24]. To address these limitations, this paper employs the GA-PSO hybrid algorithm for the adaptive optimization of PID parameters. This approach combines global and local search capabilities effectively, aiming to enhance longitudinal tracking accuracy and overall system robustness. The workflow of the GA-PSO hybrid optimization process is depicted in Figure 8, with specific implementation steps detailed in Table 7.

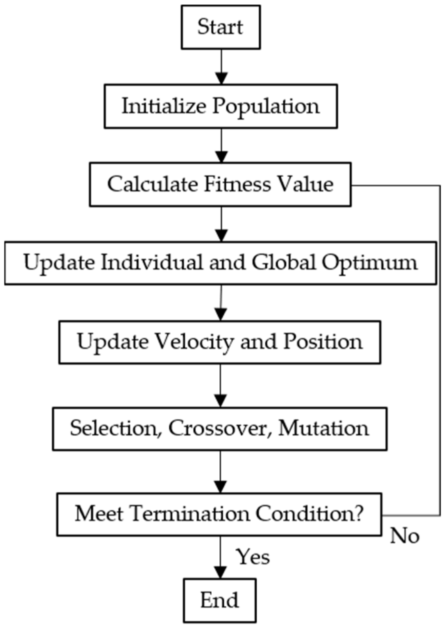


Figure 8. Flowchart of GA-PSO optimization algorithm.

Table 7. GA-PSO hybrid optimization steps for longitudinal PID controller parameters.

GA-PSO Hybrid Optimization Steps for Longitudinal PID Controller Parameters	
1: # Input: Initial PID parameters k_p, k_i, k_d, speed error $e(t)$	▷ define input variables for PID optimization
2: # Output: optimized PID parameters k_p^*, k_i^*, k_d^*	▷ define output variables
3: Initialize GA-PSO population with N candidate PID sets	▷ random initialization
4: Evaluate fitness of each candidate	▷ speed tracking, smoothness, convergence
5: while not converged and iteration $<$ max_{iter} do	▷ begin iterative optimization loop
6: # Genetic-algorithm operations	▷ GA phase
7: Select top-performing candidates	▷ selection
8: Apply crossover to generate offspring	▷ global exploration
9: Apply mutation to offspring	▷ maintain population diversity
10: # Particle swarm optimization operations	▷ PSO phase
11: Update particle velocities using personal best and global best	▷ velocity update
12: Update particle positions to obtain new PID parameters	▷ parameter update
13: Evaluate fitness of updated candidates	▷ performance reassessment
14: Update personal best and global best solutions	▷ update best solutions
15: End while	▷ exit when convergence or max iterations reached
16: Return best-performing PID parameters k_p^*, k_i^*, k_d^*	▷ output optimized gains

4. Path Tracking Controller Simulation and Verification

This chapter establishes a co-simulation platform using CarSim 2019.1 and MATLAB/Simulink R2020b to comprehensively validate the effectiveness and superiority of the proposed integrated lateral–longitudinal path tracking control strategy. A hierarchical validation approach is adopted to assess control performance. Initially, the standalone performance of the lateral FL-RL-MPC controller and the longitudinal GA-PSO-PID controller is evaluated separately. By comparing them with traditional control algorithms, the tracking accuracy and robustness of each individual controller are verified. Subsequently, the dynamic coordination and integrated control effectiveness of the combined scheme are tested under complex driving conditions. These evaluations provide data support and a theoretical foundation for the engineering application of integrated lateral–longitudinal control strategies in intelligent electric vehicles.

4.1. Simulation and Analysis of the Lateral Controller

The performance of the proposed Fuzzy Reinforcement Learning-optimized MPC (FL-RL-MPC) controller is evaluated through comparisons with traditional MPC, FL-MPC, and RL-MPC under straight-line and double-lane-change maneuvers. Before presenting the quantitative results, it is important to emphasize that the test maneuvers employed in this section (straight line, double lane change, and double sine wave) were not included in the DDPG training set. The controller's performance on these unseen trajectories thus serves as a direct assessment of its generalization capability. The straight-line condition primarily assesses steady-state trajectory tracking and controller response to minor disturbances, while the double-lane-change scenario introduces frequent curvature variations and aggressive lateral maneuvers, providing a more comprehensive test of robustness and dynamic adaptability.

Given that the double-lane-change maneuver is highly dynamic and sensitive to initial conditions, we conducted 10 repeated simulations with randomly varied initial lateral offsets (± 0.1 m) and initial heading errors (± 0.02 rad) to verify the consistency of the observed improvements. The mean and standard deviation of key metrics are reported in Table 8. The low standard deviations across all controllers—notably just 0.006 m for the FL-RL-MPC's lateral error—indicate that the performance gains are stable and reproducible, rather than artifacts of a single favorable run. For the straight-line scenario, which exhibits inherently less sensitivity to initial condition variations, a single representative run suffices to demonstrate performance, as confirmed by the smooth and consistent response in Figure 9. Figures 9 and 10 illustrate the comparative tracking performance under straight-line and double-lane-change scenarios. The corresponding quantitative indicators are summarized in Table 9. Under the straight-line condition, the FL-RL-MPC controller achieves the smallest lateral deviation (0.008 m) and heading error (0.005 rad), as well as the lowest front-wheel steering amplitude (3.5°), indicating precise trajectory tracking with minimal control effort. In the double-lane-change scenario, FL-RL-MPC maintains the lowest maximum lateral error (0.128 m) and heading error (0.052 rad), while also reducing the front-wheel steering angle compared with other controllers, demonstrating improved maneuverability and smoother actuator operation. Notably, all reported steering angles remain strictly within the prescribed bounds, confirming that constraint satisfaction is maintained throughout the experiments.

https://doi.org/10.3390/wevj17030153

Table 8. Statistical results of double-lane-change simulations over 10 runs with varying initial conditions.

Controller	Max Lateral Error (m)	Max Heading Error (rad)
MPC	0.263 ± 0.015	0.104 ± 0.008
FL-MPC	0.192 ± 0.011	0.081 ± 0.006
RL-MPC	0.170 ± 0.009	0.074 ± 0.005
FL-RL-MPC	0.127 ± 0.006	0.051 ± 0.004

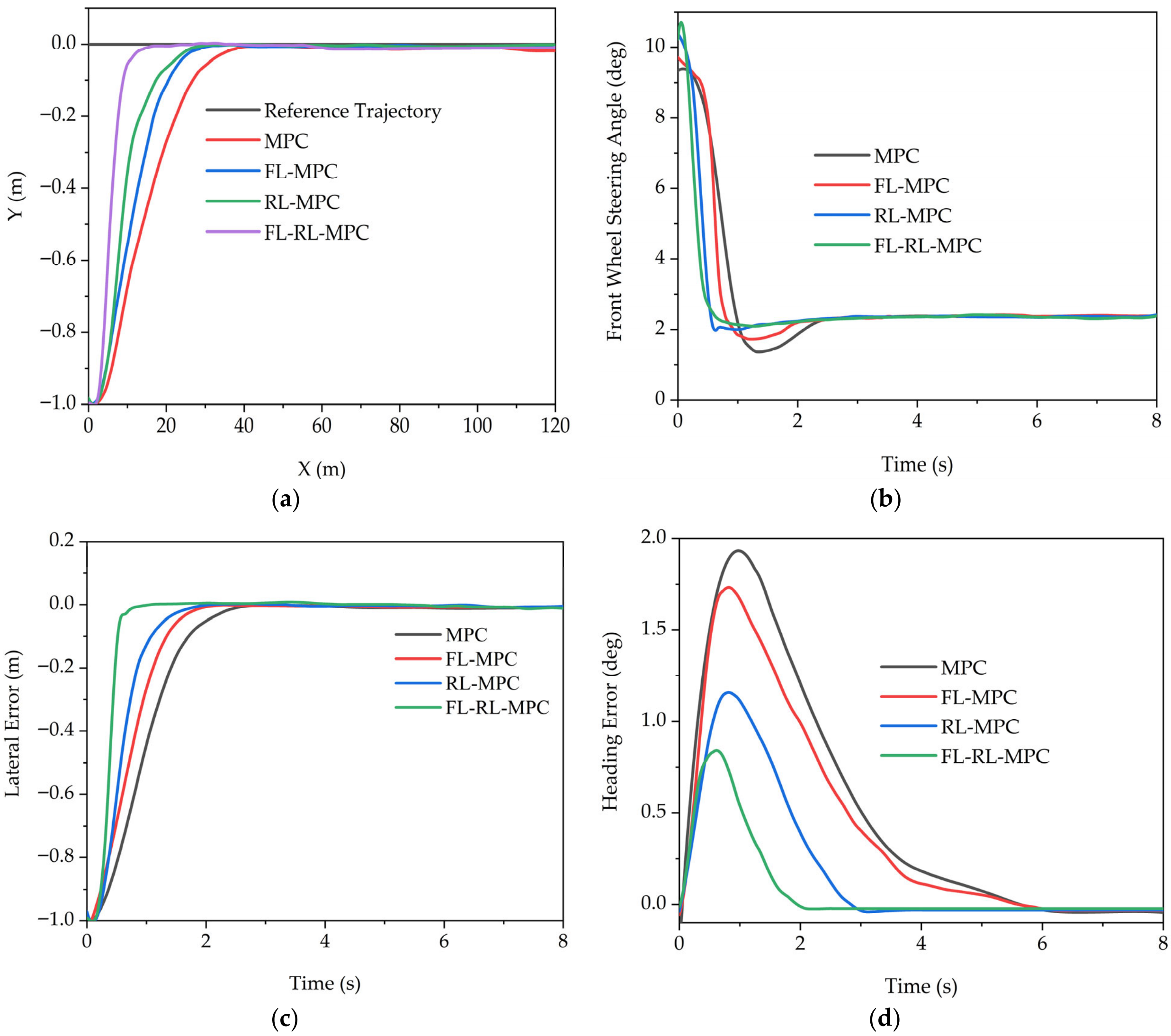


Figure 9. Comparative analysis of different path tracking methods under the straight-line driving scenario. (**a**) Trajectory tracking comparison, (**b**) front-wheel steering angle comparison, (**c**) lateral error comparison, (**d**) heading error comparison.

These results indicate that the FL-RL-MPC controller provides higher trajectory tracking accuracy, enhanced heading control, and smoother steering performance than conventional MPC, FL-MPC, and RL-MPC across both steady and highly dynamic driving conditions. The improvements are particularly notable under complex maneuvers, highlighting the benefits of integrating fuzzy logic with reinforcement learning for adaptive weight tuning in MPC.

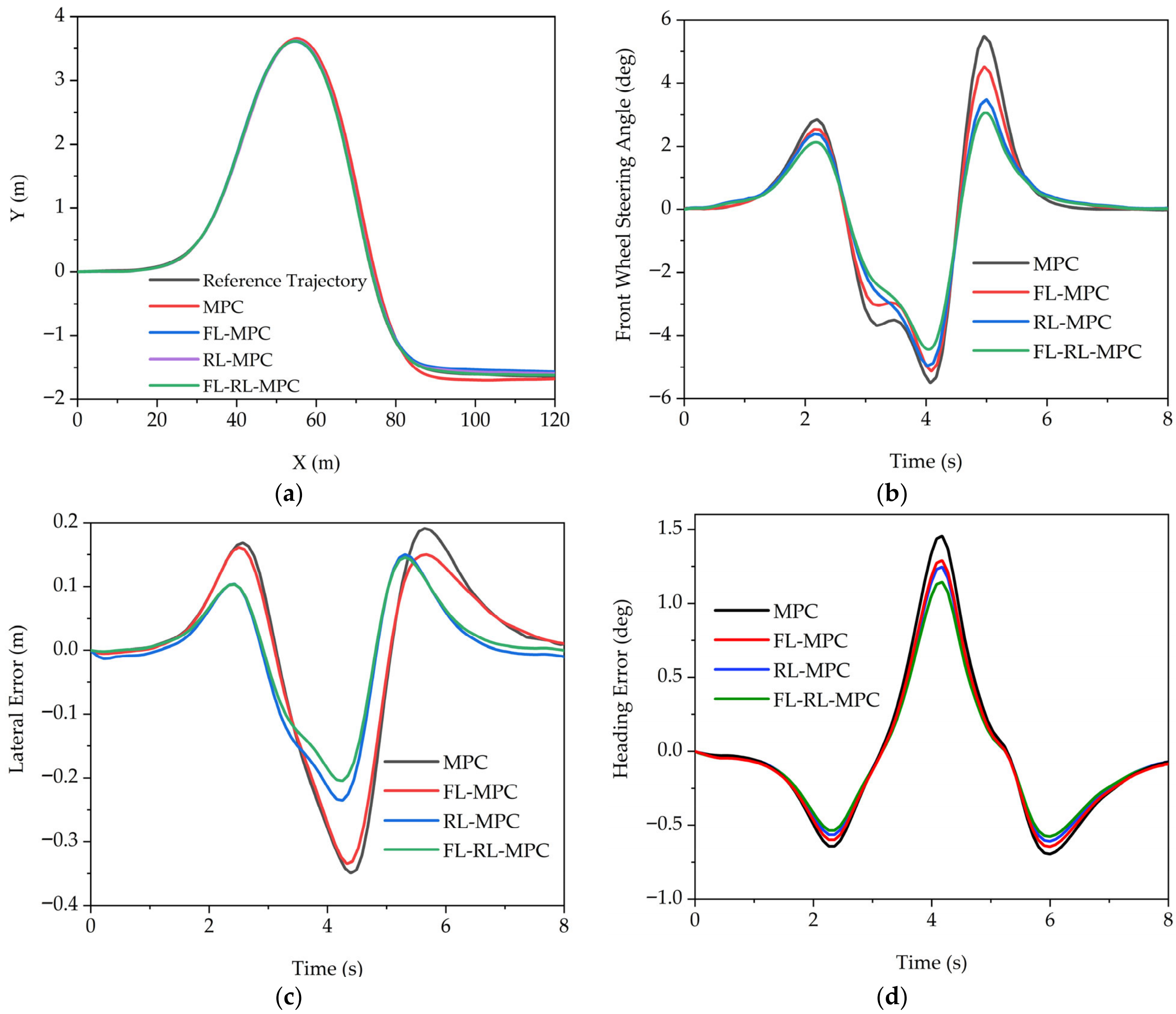


Figure 10. Comparative analysis of different path tracking methods under the double-lane-change scenario. (**a**) Trajectory tracking comparison, (**b**) front-wheel steering angle comparison, (**c**) lateral error comparison, (**d**) heading error comparison.

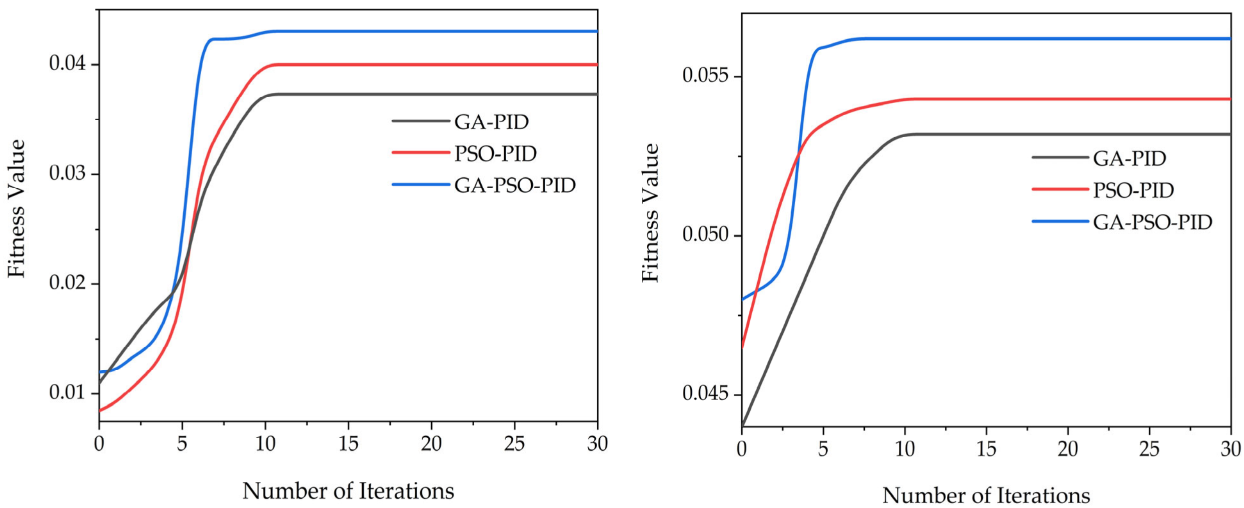
Table 9. Performance comparison of lateral controllers under different conditions.

Controller	Condition	Max Lateral Error (m)	Steady-State Lateral Error (m)	Max Heading Error (rad)	Max Front-Wheel Angle (°)
MPC	Straight	0.025	0.005	0.020	6.0
FL-MPC		0.020	0.004	0.016	4.8
RL-MPC		0.015	0.003	0.013	4.2
FL-RL-MPC		0.008	0.002	0.005	3.5
MPC	Double Lane Change	0.265	0.015	0.105	8.2
FL-MPC		0.195	0.010	0.082	6.9
RL-MPC		0.172	0.008	0.075	6.4
FL-RL-MPC		0.128	0.005	0.052	5.8

4.2. Simulation and Analysis of the Longitudinal Controller

The longitudinal speed tracking performance of the proposed GA-PSO hybrid optimized PID controller is evaluated on a straight-line reference path with a road adhesion coefficient of 0.9. Two representative scenarios are considered, including constant-speed tracking and variable-speed tracking, to comprehensively assess transient response characteristics and steady-state accuracy. For comparison, the GA-PSO-PID controller is benchmarked against a manually tuned PID controller, as well as PID controllers optimized using standalone genetic algorithm (GA-PID) and particle swarm optimization (PSO-PID) methods.

To demonstrate the optimization performance of the hybrid GA-PSO algorithm, Figure 11 presents the convergence curves of GA-PID, PSO-PID, and GA-PSO-PID under constant-speed (a) and variable-speed (b) conditions. The fitness function evaluates the integral of absolute error and overshoot to ensure both tracking accuracy and transient performance. As illustrated in Figure 11a, for the constant-speed scenario, the GA-PSO-PID algorithm (blue curve) starts with a fitness value of approximately 0.01 and rapidly rises to a stable fitness level of about 0.04 within 5 iterations, quickly approaching the optimal region and stabilizing. This indicates efficient global search and fast convergence. In the variable-speed scenario (Figure 11b), the GA-PSO-PID algorithm (blue curve) starts at a fitness value of approximately 0.047 and converges to a stable fitness level of about 0.056 within only 4 iterations, benefiting from a closer initial state to the target. In both cases, GA-PSO-PID outperforms standalone GA-PID and PSO-PID. In the constant-speed case, GA-PSO-PID reaches its stable fitness value in about 5 iterations, whereas PSO-PID (red curve) requires approximately 8 iterations and GA-PID (black curve) needs about 10 iterations to stabilize. In the variable-speed case, GA-PSO-PID converges in about 4 iterations, while PSO-PID requires roughly 6 iterations and GA-PID needs about 10 iterations to reach stable fitness.



(**a**) (**b**)

Figure 11. Fitness evolution trend. (**a**) Fitness evolution under constant velocity, (**b**) fitness evolution under variable velocity.

The superior convergence behavior of GA-PSO stems from its effective combination of GA's global exploration capability and PSO's local exploitation, avoiding premature convergence and local optima, thus providing a reliable foundation for high-quality PID parameter optimization. The optimized PID parameters obtained by each method are summarized in Table 10.

Beyond convergence speed, the stability characteristics of the optimized PID controllers are further evaluated through phase margin and gain margin analysis. Table 11 presents the phase margin and gain margin of the closed-loop system with each set of gains. The GA-PSO-optimized PID exhibits a phase margin of 62° and a gain margin of 14 dB under constant-speed conditions, which are superior to those of GA-PID (55°, 10 dB) and PSO-PID (58°, 12 dB). Similar improvements are observed under variable-speed conditions.

https://doi.org/10.3390/wevj17030153

These results indicate that the GA-PSO-optimized PID controller not only achieves faster convergence during optimization but also provides enhanced stability margins, contributing to more robust speed tracking performance. The corresponding speed response curves are presented in Figures 12 and 13, and the quantitative performance indices are reported in Table 12. As shown in Table 12, the GA-PSO-PID controller consistently achieves the shortest settling time and the smallest maximum speed error in both scenarios. Under constant-speed conditions, compared with the manually tuned PID controller, the GA-PSO-PID reduces the settling time from 5.2 s to 2.0 s (a reduction of approximately 61.5%) and decreases the maximum speed error from 0.62 m/s to 0.19 m/s (an improvement of about 69.4%). In the more demanding variable-speed scenario, the GA-PSO-PID controller similarly outperforms all benchmarks. It achieves a settling time of 3.0 s, which is 50.8% faster than the PID's 6.1 s, and records a maximum speed error of 0.18 m/s, representing a 71.4% reduction compared to the PID's 0.63 m/s. These quantitative results confirm the superior dynamic response and steady-state accuracy of the proposed hybrid optimization approach under varying operational conditions.

Table 10. PID parameters for different controllers under constant/variable-speed conditions.

Controller	Constant Speed (k_p,k_i,k_d)	Variable Speed (k_p,k_i,k_d)
PID	2, 0.5, 1	10, 0.2, 1
GA-PID	3, 0.5, 3	11, 0.5, 2.5
PSO-PID	3, 0.3, 2.8	11.5, 0.3, 2.2
GA-PSO-PID	3.5, 0.2, 2.5	12.5, 0.1, 2

Table 11. Stability analysis of different PID controllers.

Controller	Condition	Phase Margin (°)	Gain Margin (dB)
GA-PID	Constant	55	10
PSO-PID		58	12
GA-PSO-PID		62	14
GA-PID	Variable	53	9
PSO-PID		56	11
GA-PSO-PID		60	13

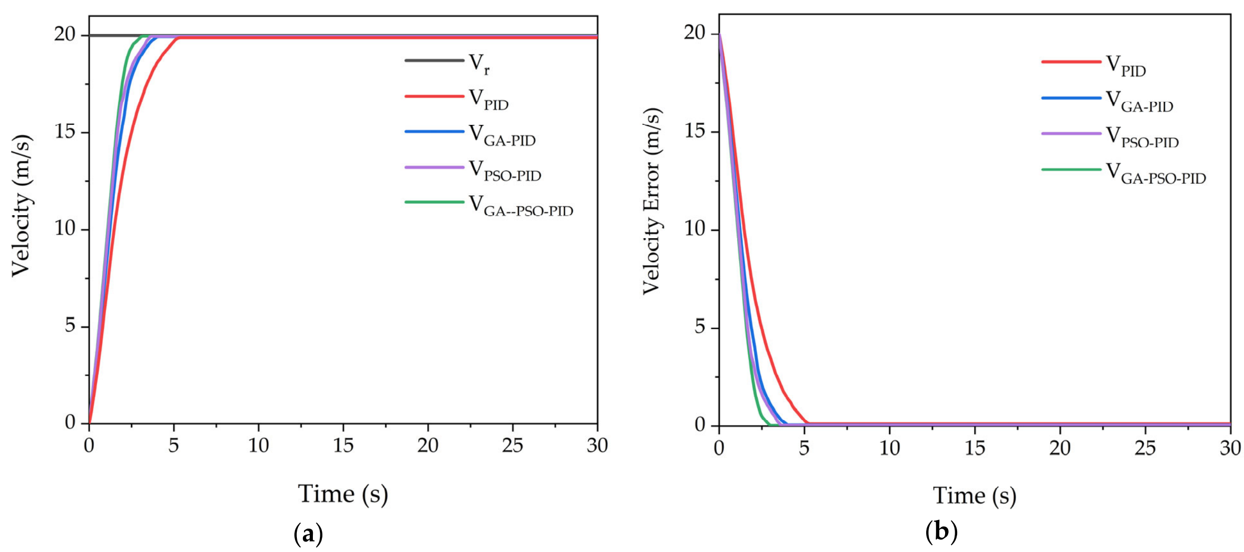


Figure 12. Simulation results under constant-speed conditions. (**a**) Velocity tracking curve, (**b**) velocity error curve.

https://doi.org/10.3390/wevj17030153

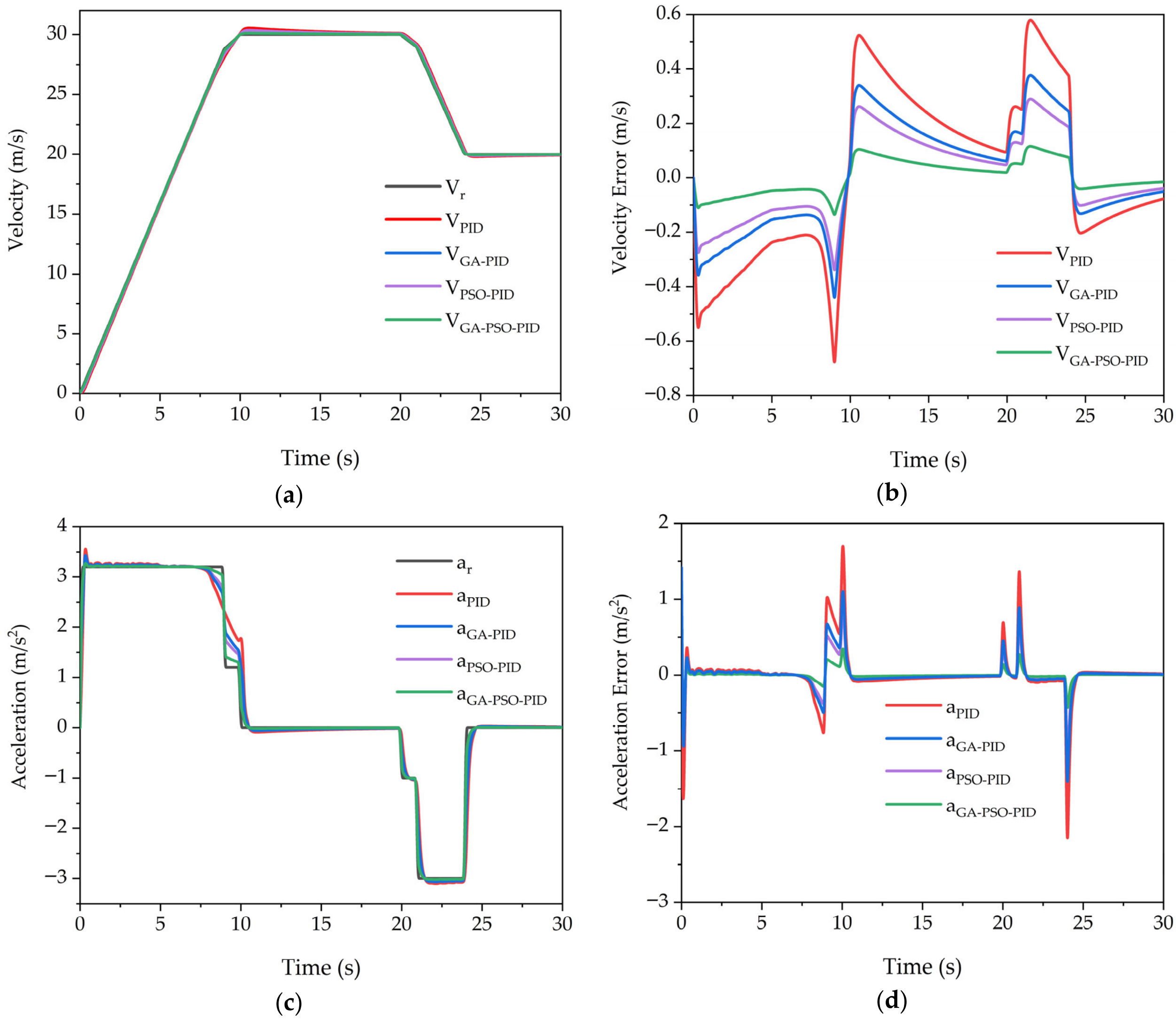


Figure 13. Simulation results under variable-speed conditions. (**a**) Velocity tracking curve, (**b**) velocity error curve, (**c**) acceleration tracking curve, (**d**) acceleration error curve.

Table 12. Performance comparison of longitudinal controllers.

Controller	Condition	Settling Time (s)	Max Speed Error (m/s)
PID	Constant	5.2	0.61
GA-PID		3.1	0.48
PSO-PID		2.9	0.41
GA-PSO-PID		2.0	0.19
PID	Variable	6.1	0.63
GA-PID		4.2	0.49
PSO-PID		3.7	0.40
GA-PSO-PID		3.0	0.18

The simulation results confirm that the GA-PSO-PID controller provides faster dynamic response, improved steady-state accuracy, and smoother speed tracking curves compared with conventional and single-algorithm optimized PID controllers. These findings validate the effectiveness of the hybrid GA-PSO optimization approach for longitudinal control of intelligent electric vehicles.

4.3. Integrated Lateral–Longitudinal Coordinated Control Validation

To evaluate the overall effectiveness of the proposed integrated lateral–longitudinal control framework, different lateral controllers (MPC, FL-MPC, RL-MPC, and FL-RL-MPC) are combined with the GA–PSO–PID longitudinal controller and tested under typical dynamic driving conditions. Double-lane-change and double-sine steer maneuvers are selected as representative scenarios to assess tracking accuracy, control smoothness, and coordination performance.

The simulations are conducted on a CarSim-MATLAB/Simulink co-simulation platform, forming a closed-loop vehicle dynamics and control system. The initial vehicle speed is set to 60 km/h, the road adhesion coefficient is 0.9, and the control sampling period is 0.01 s. The comparative performance results of different lateral–longitudinal controller combinations are illustrated in Figures 14 and 15, while the corresponding quantitative indicators are summarized in Table 13. As shown in Table 13, the proposed FL-RL-MPC combined with the GA-PSO-PID longitudinal controller consistently achieves the smallest maximum lateral error, heading error, and front-wheel steering angle under both test scenarios. Under the double-lane-change maneuver, compared with the conventional MPC-based scheme, the FL-RL-MPC combination reduces the maximum lateral error from 0.265 m to 0.128 m (a reduction of approximately 51.7%), decreases the maximum heading error from 0.105 rad to 0.052 rad (an improvement of 50.5%), and lowers the maximum front-wheel steering angle from 8.2° to 5.8° (a reduction of 29.3%). For the double-sine steer maneuver, the proposed controller achieves even more pronounced improvements in heading error: the maximum lateral error is reduced from 0.280 m to 0.128 m (54.3% improvement), the maximum heading error decreases from 0.110 rad to 0.012 rad (89.1% improvement), and the front-wheel steering angle is reduced from 8.5° to 6.2° (27.1% reduction). Furthermore, the FL-RL-MPC-based strategy maintains smoother steering angle variations across both maneuvers, indicating improved actuator smoothness and reduced control effort. The maximum longitudinal speed error remains consistently low across all controller combinations, with the FL-RL-MPC + GA-PSO-PID achieving the smallest values of 0.21 m/s and 0.23 m/s in the two scenarios, demonstrating the effectiveness of the GA-PSO-PID longitudinal controller in maintaining speed tracking accuracy during complex lateral maneuvers.

Table 13. Performance of integrated lateral–longitudinal controller combinations.

Lateral + Longitudinal Controller	Condition	Max Lateral Error (m)	Max Heading Error (rad)	Max Front-Wheel Angle (°)	Max Longitudinal Speed Error (m/s)
MPC + GA-PSO-PID	Double Lane Change	0.265	0.105	8.2	0.48
FL-MPC + GA-PSO-PID		0.195	0.082	6.9	0.38
RL-MPC + GA-PSO-PID		0.172	0.075	6.4	0.35
FL-RL-MPC + GA-PSO-PID		0.128	0.052	5.8	0.21
MPC + GA-PSO-PID	Double Sine Wave	0.280	0.110	8.5	0.59
FL-MPC + GA-PSO-PID		0.180	0.015	7.1	0.42
RL-MPC + GA-PSO-PID		0.155	0.014	6.8	0.31
FL-RL-MPC + GA-PSO-PID		0.128	0.012	6.2	0.23

https://doi.org/10.3390/wevj17030153

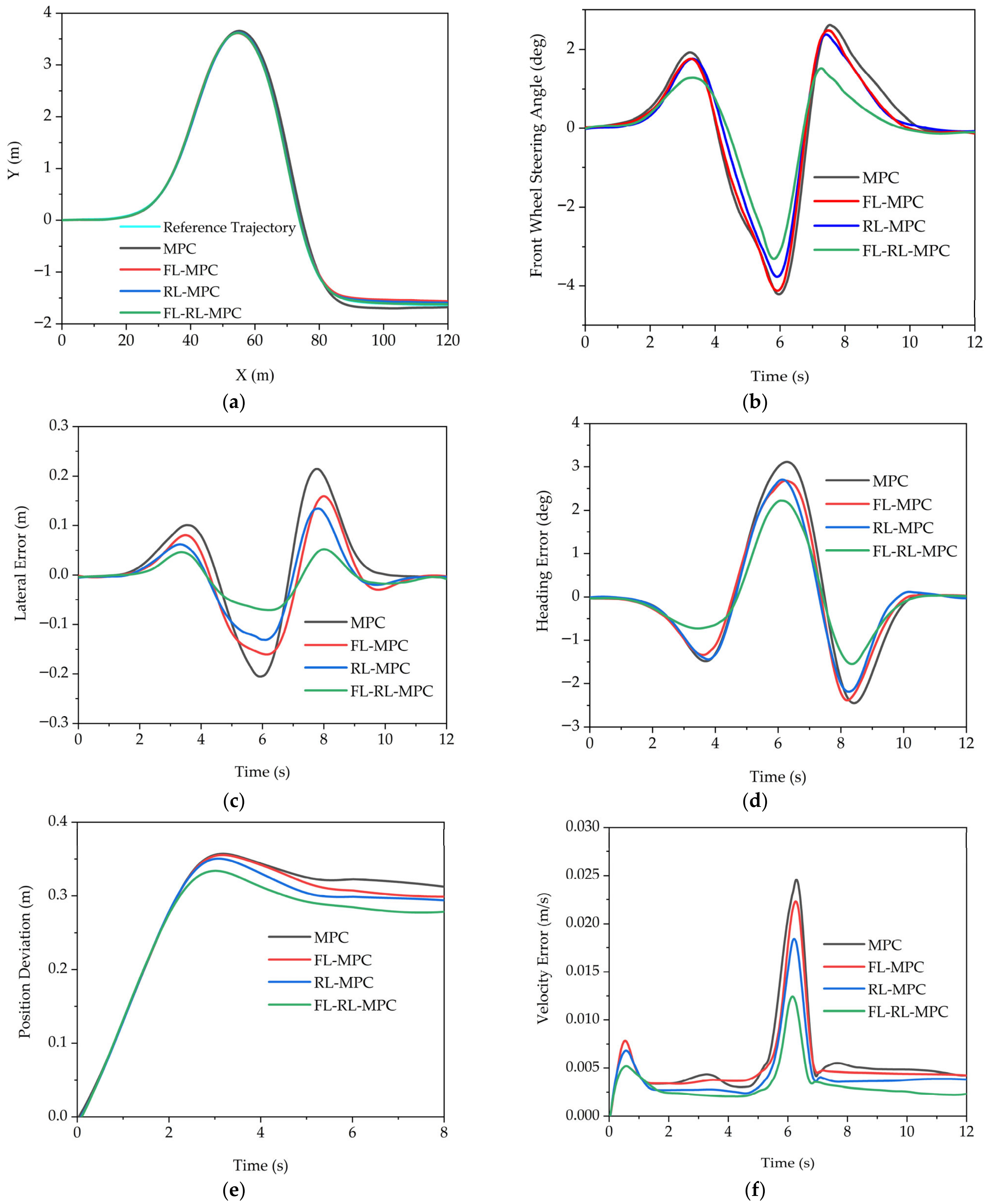


Figure 14. Comparative analysis of methods under double-lane-change condition. (**a**) Trajectory tracking comparison, (**b**) front-wheel steering angle comparison, (**c**) lateral error comparison, (**d**) heading error comparison, (**e**) position error comparison, (**f**) velocity error comparison.

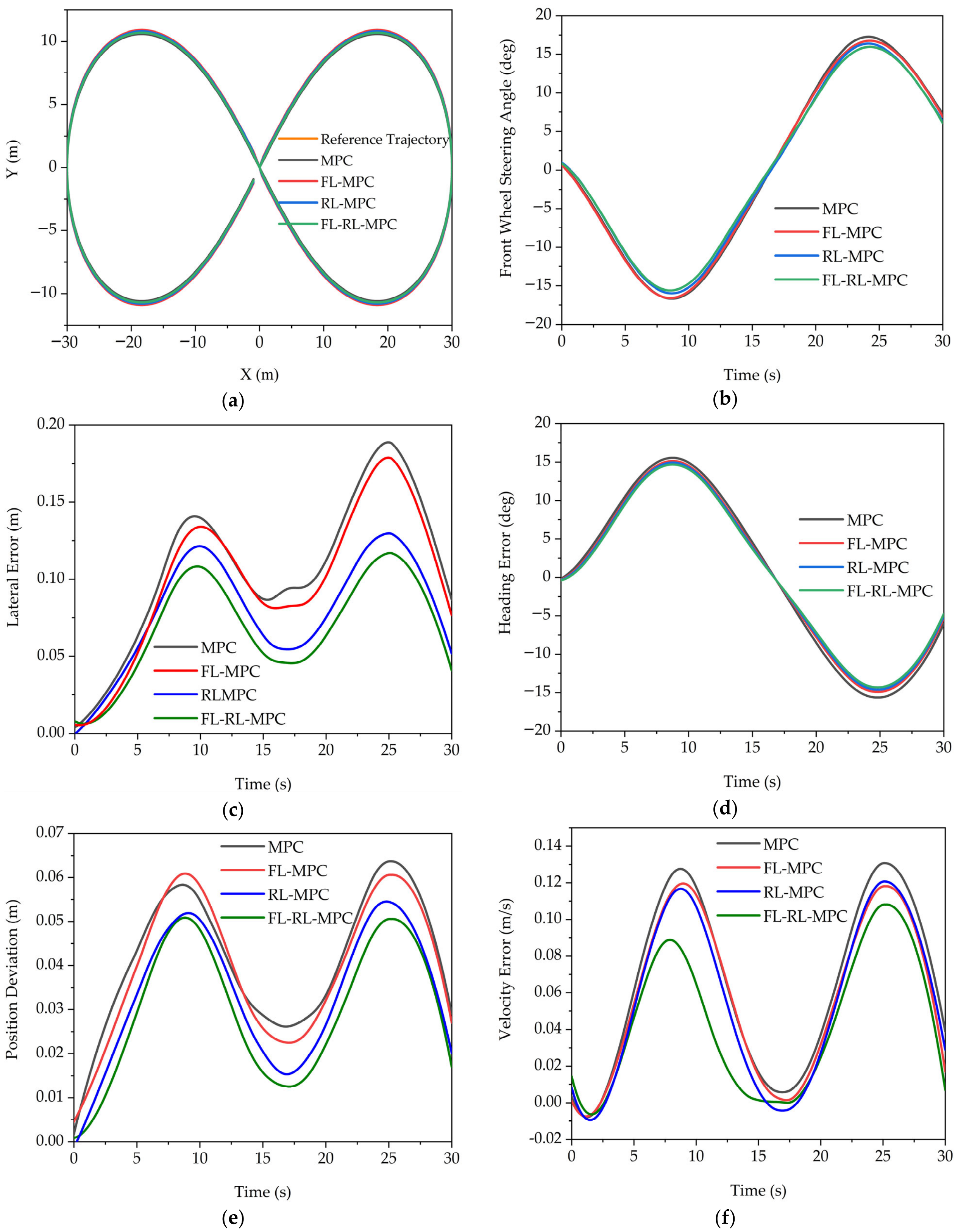


Figure 15. Comparative analysis of methods under double-sine-wave steering condition. (**a**) Trajectory tracking comparison, (**b**) front-wheel steering angle comparison, (**c**) lateral error comparison, (**d**) heading error comparison, (**e**) position error comparison, (**f**) velocity error comparison.

To further assess the controller's robustness under non-ideal conditions, additional tests were performed considering sensor noise, model uncertainty, and varying road adhesion. These tests included the following:

(1) adding zero-mean Gaussian noise ($\sigma = 0.1$ m) to the measured lateral position;
(2) simultaneously increasing vehicle mass by 20% and reducing cornering stiffness by 20%;
(3) conducting both double-lane-change and double-sine-wave maneuvers at a reduced adhesion coefficient of $\mu = 0.5$;
(4) applying a step reduction in road adhesion from 0.9 to 0.5 mid-maneuver (at $t = 5$ s).

The results, summarized in Table 14, demonstrate that the FL-RL-MPC controller maintains superior performance across all perturbed conditions for both test maneuvers. For the double-lane-change scenario, under sensor noise, FL-RL-MPC achieves a maximum lateral error of 0.15 m compared to 0.29 m for conventional MPC. Under model uncertainty, the peak error increase is limited to 14% for FL-RL-MPC versus 38% for conventional MPC. In low-adhesion and sudden-adhesion-change scenarios, FL-RL-MPC consistently achieves maximum lateral errors below 0.22 m, while conventional MPC exceeds 0.41 m. Similar trends are observed for the double-sine-wave maneuver, where FL-RL-MPC maintains maximum lateral errors below 0.22 m across all perturbed conditions, substantially outperforming conventional MPC. It is worth noting that the robustness test with simultaneous 20% mass increase and 20% cornering stiffness reduction represents a far more severe perturbation than the actual stiffness variations expected within the linear tire operation region, where slip angles remain below 5° as verified earlier. The fact that the proposed FL-RL-MPC maintains stable tracking performance under such a large stiffness variation provides strong evidence that the controller is robust to model mismatch. Therefore, even if the actual cornering stiffness deviates from the nominal constant values used in the prediction model, the DDPG-based weight adaptation mechanism can compensate online, justifying the constant stiffness assumption for the operating conditions considered in this study. These results collectively demonstrate the enhanced robustness of the proposed framework under realistic disturbances and varying operating conditions.

Table 14. Robustness analysis under disturbances.

Lateral Controller + Longitudinal Controller	Condition	Nominal (μ = 0.9)	Sensor Noise (σ = 0.1 m)	Mass+20% & Cornering Stiffness-20%	Low Adhesion (μ = 0.5)	Sudden Adhesion Step (0.9→0.5 @ t = 5 s)
MPC + GA-PSO-PID	Double Lane Change	0.265	0.29	0.366 (+38%)	0.42	0.41
FL-MPC + GA-PSO-PID		0.195	0.22	0.238 (+22%)	0.31	0.30
RL-MPC + GA-PSO-PID		0.172	0.19	0.207 (+20%)	0.28	0.27
FL-RL-MPC + GA-PSO-PID		0.128	0.15	0.146 (+14%)	0.21	0.22
MPC + GA-PSO-PID	Double Sine Wave	0.280	0.31	0.392 (+40%)	0.45	0.43
FL-MPC + GA-PSO-PID		0.180	0.21	0.225 (+25%)	0.29	0.28
RL-MPC + GA-PSO-PID		0.155	0.18	0.194 (+25%)	0.26	0.25
FL-RL-MPC + GA-PSO-PID		0.128	0.15	0.154 (+20%)	0.22	0.21

Overall, the results demonstrate that the proposed FL-RL-MPC + GA-PSO-PID coordinated control strategy provides superior trajectory tracking accuracy, enhanced control smoothness, and effective lateral–longitudinal dynamic coordination, especially under aggressive and highly coupled driving maneuvers.

4.4. Further Analysis and Discussion

To further validate the proposed framework, additional analyses were conducted regarding computational efficiency and component-wise contributions. To assess the real-time feasibility of the proposed framework, the average computation time per control cycle was measured on an Intel Core i5-8300H CPU @ 2.30 GHz with 16 GB RAM over 10,000 simulation steps. Table 15 summarizes the computation time for each algorithmic module. As shown in Table 15, the total computation time per control cycle is approximately 13.0 ms, which is below the sampling period of 20 ms used in all simulations, leaving a 7 ms margin for real-time execution. Among the modules, MPC optimization contributes the largest portion (12.0 ms), while fuzzy inference and DDPG inference impose relatively low overhead. Based on these measurements on a consumer-grade laptop CPU (Intel Core i5-8300H), it is reasonable to infer that with code optimization and fixed-point implementation, the proposed framework could meet typical 10–20 ms sampling constraints on modern automotive processors such as Infineon AURIX and NVIDIA Orin, which are capable of running similar QP solvers and lightweight neural network inferences within 10 ms. Hardware-in-the-loop validation is planned as future work to confirm real-time performance under more realistic conditions.

Table 15. Average computation time per control cycle (over 10,000 steps).

Module	Average Time (ms)
Fuzzy inference	<0.02
DDPG actor forward pass	0.50
MPC optimization (OSQP solver)	12.00
FL-RL-MPC lateral control total	12.50
Longitudinal PID online calculation	<0.02
Total per cycle	~12.54

Note: All tests run on an Intel Core i5-8300H (2.30 GHz, 16 GB RAM). Fuzzy inference includes membership and rule evaluation. DDPG uses a two-layer (64-64) actor network. MPC employs OSQP with $N_p = 20$ and $N_c = 10$.

Beyond the real-time feasibility assessment, a quantitative analysis was conducted to isolate the contribution of each algorithmic component through an ablation study under the double-lane-change scenario. Five configurations were compared: baseline MPC with fixed weights; MPC with fuzzy-initialized weights but no reinforcement learning adaptation; MPC with DDPG-based adaptation but no fuzzy initialization (MPC + RL); the proposed FL-RL-MPC combining both fuzzy initialization and RL adaptation; and the full framework integrating FL-RL-MPC with GA-PSO-PID longitudinal control. The comparative results, summarized in Table 16, reveal several insights. Compared to the baseline MPC, fuzzy-initialized weights alone reduce the maximum lateral error by approximately 26%, demonstrating the effectiveness of expert knowledge injection. Reinforcement learning adaptation alone achieves a 35% error reduction, indicating its capability for online improvement. However, during early training stages, the RL-only configuration exhibited less stable behavior, highlighting the importance of proper initialization. The combination of fuzzy initialization and RL adaptation (FL-RL-MPC) yields a 52% error reduction, which is significantly greater than the sum of individual improvements, confirming the synergistic effect of the two techniques. Finally, integrating GA-PSO-PID longitudinal control further enhances speed tracking performance without affecting lateral control metrics.

These additional analyses collectively confirm that the proposed framework is both computationally feasible for real-time implementation and that each component contributes synergistically to the overall performance improvement.

Table 16. Ablation study results under double-lane-change scenario.

Configuration	Max Lateral Error (m)	Reduction vs. Baseline
Baseline MPC (fixed weights)	0.265	-
MPC + Fuzzy (no RL)	0.195	26%
MPC + RL (no fuzzy initialization)	0.172	35%
MPC + Fuzzy + RL (FL-RL-MPC)	0.128	52%
Full framework (FL-RL-MPC + GA-PSO-PID)	0.128	52% (lateral) + longitudinal improvement

5. Conclusions

This paper proposed a hierarchical control framework for intelligent-electric-vehicle path tracking, integrating a fuzzy logic–reinforcement learning-enhanced model predictive control (FL-RL-MPC) lateral controller with a genetic algorithm–particle swarm optimization-optimized PID (GA-PSO-PID) longitudinal controller. The lateral controller employs fuzzy inference for reasonable weight initialization and deep reinforcement learning for online adaptive tuning, enabling the MPC to maintain optimal performance under varying driving conditions without manual re-calibration. The longitudinal controller utilizes a hybrid GA-PSO algorithm to optimize PID gains offline, achieving fast convergence and accurate speed tracking with negligible online computational burden.

The effectiveness of the proposed framework was validated through CarSim/Simulink co-simulations under multiple scenarios. Quantitative results demonstrate substantial improvements: under double-lane-change maneuvers, the FL-RL-MPC reduces maximum lateral error by 51.7% and maximum heading error by 50.5% compared to conventional MPC. For longitudinal control, the GA-PSO-PID reduces settling time by 61.5% and maximum speed error by 69.4% under constant-speed conditions. Integrated, the full framework achieves over 50% reduction in lateral error and approximately 70% reduction in speed tracking error relative to baseline MPC + PID controllers. These findings indicate that the developed control framework provides a practical and scalable solution for integrated lateral–longitudinal control of intelligent electric vehicles.

It is worth noting that while the extensive simulation results provide strong empirical evidence of closed-loop stability, a formal theoretical guarantee remains an open research challenge for reinforcement-learning-adaptive MPC. The integration of formal stability guarantees into the proposed adaptive framework is therefore identified as a key future research direction. Meanwhile, the current study employs a simplified vehicle dynamics model, which may not fully capture nonlinear tire behavior or real-world actuator dynamics. Future work will prioritize the integration of state observers (e.g., EKF) for practical deployment, followed by hardware-in-the-loop validation and proving-ground tests. Efforts will also be directed toward incorporating formal stability guarantees into the adaptive control framework and optimizing computational efficiency for embedded implementation.

Author Contributions: Conceptualization, Q.G., H.C. and G.L.; Methodology, Q.G., H.C. and G.L.; Software, Q.G.; Validation, Q.G., H.Z., Q.L. and D.P.; Formal Analysis, Q.G. and G.L.; Investigation, Q.G. and H.Z.; Resources, H.C., G.L., Q.L. and D.P.; Data Curation, Q.G. and H.Z.; Writing—Original Draft Preparation, Q.G.; Writing—Review and Editing, H.C. and G.L.; Visualization, Q.G.; Supervision, H.C. and G.L.; Project Administration, H.C. and G.L.; Funding Acquisition, H.C. and G.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Guangxi Science and Technology Project under Grants 2023GXNSFFA026002, AD23026117, and 2024GXNSFBA010261; and in part by the Guangdong Province Talent Recruitment Program.

Data Availability Statement: The original simulation data supporting the conclusions of this article are openly available in Mendeley Data at https://doi.org/10.17632/3gzhy9d6sw.1. The dataset includes path tracking test data of intelligent electric vehicles under three typical operating conditions: straight-line path, double-lane-change maneuver, and double-sinusoidal trajectory, which are consistent with the simulation scenarios reported in the manuscript.

Acknowledgments: The authors would like to express their sincere gratitude to the School of Electronic Engineering and Guangxi Key Laboratory of Multidimensional Information Fusion for Intelligent Vehicles, Guangxi University of Science and Technology, for providing the necessary research platforms and technical support during the study. Special thanks are extended to Dongfeng Liuzhou Motor Co., Ltd., for their valuable collaboration and practical resources that facilitated the smooth progress of this research.

Conflicts of Interest: Authors Qianqiang Lu and Defeng Peng were employed by the company Dongfeng Liuzhou Motor Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Faghihian, H.; Sargolzaei, A. Energy efficiency of connected autonomous vehicles: A review. *Electronics* **2023**, *12*, 4086. [CrossRef]
2. Hu, L.; Zhou, X.; Zhang, X.; Wang, F.; Li, Q.; Wu, W. A review on key challenges in intelligent vehicles: Safety and driver-oriented features. *IET Intell. Transp. Syst.* **2021**, *15*, 1093–1105. [CrossRef]
3. Chaurasia, V.; Tiwari, A.N.; Tripathi, S.M. Coordinated control of autonomous electric vehicles with lateral and longitudinal control using a hybrid approach. *J. Field Robot.* **2025**, *42*, 3380–3398. [CrossRef]
4. Rokonuzzaman, M.; Mohajer, N.; Nahavandi, S.; Mohamed, S. Review and performance evaluation of path tracking controllers of autonomous vehicles. *IET Intell. Transp. Syst.* **2021**, *15*, 646–670. [CrossRef]
5. Zhang, Y.; Liu, K.; Gao, F.; Zhao, F. Research on path planning and path tracking control of autonomous vehicles based on improved APF and SMC. *Sensors* **2023**, *23*, 7918. [CrossRef]
6. Zheng, Z.A.; Ye, Z.; Zheng, X. Intelligent vehicle lateral control strategy research based on feedforward + predictive LQR algorithm with GA optimisation and PID compensation. *Sci. Rep.* **2024**, *14*, 22317. [CrossRef]
7. Zhang, Z.; Cao, X.; Huang, C.; Hu, L.; Wu, W. Robust MPC-based centralized cooperative control of trajectory tracking and handling stability for distributed drive electric vehicles. *J. Frankl. Inst.* **2025**, *362*, 108217.
8. Yao, Q.; Shi, Y.; Hang, p.; Tian, Y. Path Tracking Robust Control Strategy for Intelligent Vehicle Based on Force-Driven with MPC and H∞. *Chin. J. Mech. Eng.* **2025**, *38*, 184.
9. Lelkó, A.; Németh, B.; Gáspár, P. Reinforcement learning-based robust vehicle control for autonomous vehicle trajectory tracking. *Eng. Proc.* **2024**, *79*, 30.
10. Viadero-Monasterio, F.; Meléndez-Useros, M.; Zhang, N.; Zhang, H.; Boada, B.L.; Boada, M.J.L. Motion planning and robust output-feedback trajectory tracking control for multiple intelligent and connected vehicles in unsignalized intersections. *IEEE Trans. Veh. Technol.* **2025**, *74*, 18543–18555. [CrossRef]
11. Viadero-Monasterio, F.; Nguyen, A.-T.; Lauber, J.; Boada, M.J.L.; Boada, B.L. Event-triggered robust path tracking control considering roll stability under network-induced delays for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 14743–14756. [CrossRef]
12. Mayne, D.Q.; Rawlings, J.B.; Rao, C.V.; Scokaert, P.O.M. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814. [CrossRef]
13. Falcone, P.; Borrelli, F.; Asgari, J.; Tseng, H.E.; Hrovat, D. Predictive active steering control for autonomous vehicle systems. *IEEE Trans. Control Syst. Technol.* **2007**, *15*, 566–580. [CrossRef]
14. Ji, J.; Khajepour, A.; Melek, W.W.; Huang, Y. Path planning and tracking for vehicle collision avoidance based on model predictive control with multi-constraints. *IEEE Trans. Veh. Technol.* **2017**, *66*, 952–964. [CrossRef]
15. Du, Q.; Zhu, C.; Li, Q.; Tian, B.; Li, L. Optimal path tracking control for intelligent four-wheel steering vehicles based on MPC and state estimation. *J. Automob. Eng.* **2021**, *236*, 1964–1976. [CrossRef]
16. Shi, P.; Li, L.; Ni, X.; Yang, A. Intelligent vehicle path tracking control based on improved MPC and hybrid PID. *IEEE Access* **2022**, *10*, 94133–94144. [CrossRef]
17. Altan, A.; Hacıoğlu, R. Model predictive control of autonomous vehicles for path tracking with an adaptive weight tuning strategy based on particle swarm optimization. *Sensors* **2022**, *22*, 3789.

18. Zhang, H.; Liu, M.; Yang, R.; Guo, H. A reinforcement learning-based adaptive model predictive controller for autonomous vehicle path tracking under slippery conditions. *Sensors* **2023**, *23*, 6785.
19. Chen, Y.; Gai, J.; He, S.; Li, H.; Cheng, C.; Zou, W. MPC TD3 trajectory tracking control for electrically driven unmanned tracked vehicles. *Electronics* **2024**, *13*, 3747. [CrossRef]
20. Zhang, X.-L.; Lu, Y.; Li, W.-Z.; Xu, X. Receding horizon reinforcement learning algorithm for lateral control of intelligent vehicles. *Acta Autom. Sin.* **2023**, *49*, 2481–2492.
21. Li, C.; Jiang, H.; Yang, X.; Wei, Q. Path tracking control strategy based on adaptive MPC for intelligent vehicles. *Appl. Sci.* **2025**, *15*, 5464. [CrossRef]
22. Sun, Q.; Li, T.; Chu, D.; Wu, J. A hierarchical control strategy for path tracking of autonomous vehicles with fuzzy logic and particle swarm optimization-tuned PID. *Sensors* **2022**, *22*, 7043.
23. Baz, R.; Majdoub, K.; Giri, F.; Taouni, A. Self-tuning fuzzy PID speed controller for quarter electric vehicle driven by in-wheel BLDC motor and Pacejka's tire model. *IFAC-PapersOnLine* **2022**, *55*, 598–603. [CrossRef]
24. Elias, G.H.S.; Al-Moadhen, A.; Kamil, H. Optimizing the PID controller to control the longitudinal motion of autonomous vehicles. *AIP Conf. Proc.* **2023**, *2591*, 040045.
25. Nada, A.; Ahmed, L.; Elnaggar, M.; Kamel, A. Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles. *ISA Trans.* **2022**, *129*, 193–205. [CrossRef]
26. Jeong, Y.; Yim, S. Model predictive control-based integrated path tracking and velocity control for autonomous vehicle with four-wheel independent steering and driving. *Electronics* **2021**, *10*, 2812. [CrossRef]
27. Li, Y.; Wang, Z.; Li, K.; Chen, B. An integrated path tracking and lateral stability controller based on model predictive control with adaptive preview for autonomous vehicles. *Sensors* **2024**, *24*, 987.
28. Zhou, T.; Xu, C.; He, Y.; Feng, Y. Cooperative path tracking control of connected and autonomous electric vehicles using a data-driven adaptive MPC framework. *Sensors* **2022**, *22*, 8760.
29. Garcia, C.E.; Villagra, J.; Milanés, V. On the combined use of model predictive control and radial basis function networks for vehicle path tracking. *Sensors* **2021**, *21*, 1450.
30. Wu, D.; Guan, Y.; Xia, X.; Du, C.; Yan, F.; Li, Y.; Hua, M.; Liu, W. Coordinated control of path tracking and yaw stability for distributed drive electric vehicle based on AMPC and DYC. *Proc. Inst. Mech. Eng. Part D J. Automob. Eng.* **2024**, *239*, 1745–1762. [CrossRef]
31. Peng, H.; Wang, W.; An, Q.; Xiang, C.; Li, L. Path tracking and direct yaw moment coordinated control based on robust MPC with the finite time horizon for autonomous independent-drive vehicles. *IEEE Trans. Veh. Technol.* **2020**, *69*, 6053–6066. [CrossRef]
32. Wang, F.; Zhang, Y.; Zhao, Y.; Liu, C. Robust lateral control of intelligent vehicles using an adaptive sliding mode controller optimized by a genetic algorithm. *Sensors* **2020**, *20*, 7298.
33. Park, S.; Kim, D.; Lee, S. Real-time trajectory tracking for electric vehicles using a constrained neural network-based nonlinear model predictive controller. *Sensors* **2021**, *21*, 5412.
34. Zhang, Z.; Lu, H.; Feng, W.; Wu, W. Real-time improvement measures of nonlinear model predictive control for intelligent vehicle trajectory tracking. *Automot. Innov.* **2025**, *8*, 137546. [CrossRef]
35. Wang, L.; Jiang, H.; Yin, G.; Li, S. Path following control of autonomous ground vehicles subject to cyber-attacks via a sliding mode and disturbance observer-based approach. *Sensors* **2021**, *21*, 6589.
36. Chen, J.; Wang, P.; Li, X.; Zhang, L. A hybrid MPC and DDPG deep reinforcement learning approach for autonomous driving path tracking. *Sensors* **2023**, *23*, 4321.
37. Kim, H.; Park, J.; Lee, J.; Choi, S. Adaptive lane-keeping control via fuzzy model predictive control with online learning for intelligent vehicles. *Sensors* **2024**, *24*, 1876.
38. Liu, Y.; Zhang, W.; Wang, J.; Chen, Z. Multi-objective optimization of autonomous vehicle path tracking controller using an improved NSGA-III algorithm. *Sensors* **2023**, *23*, 5567.
39. Abe, M.; Manning, W. *Vehicle Dynamics and Control*; Elsevier Ltd.: Amsterdam, The Netherlands, 2009; pp. 1–368.
40. Yoon, S.; Kwon, Y.; Ryu, J.; Kim, S.; Choi, S.; Lee, K. Reinforcement-learning-based trajectory learning in Frenet frame for autonomous driving. *Appl. Sci.* **2024**, *14*, 6977. [CrossRef]
41. Cao, Y.; Ni, K.; Jiang, X.; Kuroiwa, T.; Zhang, H.; Kawaguchi, T.; Hashimoto, S.; Jiang, W. Path following for autonomous ground vehicle using DDPG algorithm: A reinforcement learning approach. *Appl. Sci.* **2023**, *13*, 6847. [CrossRef]
42. Carrasco, A.V.; Sequeira, J.S. Tuning path tracking controllers for autonomous cars using reinforcement learning. *PeerJ Comput. Sci.* **2023**, *9*, e1550. [CrossRef]
43. Wang, Y.; Hu, F.Q.; Tian, C.; Li, P.S.; Huang, H.L.; Yin, G.D.; Huang, C. FTEKFNet: Hybridizing physical and data-driven estimation algorithms for vehicle state estimation. *IEEE Trans. Intell. Veh.* **2024**, *9*, 7301–7311. [CrossRef]