IET Cyber-Systems and Robotics

WILEY

ZHEJIANG UNIVERSITY PRESS IET The Institution of Engineering and Technology

ORIGINAL RESEARCH OPEN ACCESS

Automated Parking Systems Using Reinforcement Learning Assisted Model Predictive Control

Hussein Alawsi | Zhaodong Zhou | Ali Irshayyid | Jun Chen

ECE Department, Oakland University, Rochester, Michigan, USA

Correspondence: Jun Chen (junchen@oakland.edu)

Received: 9 September 2025 | **Revised:** 21 January 2026 | **Accepted:** 15 March 2026

Keywords: adaptive control | autonomous vehicle | deep learning | nonlinear control | nonlinear model predictive control | reinforcement learning

ABSTRACT

Autonomous parking remains a challenging task due to the need for accurate trajectory tracking, smooth steering, and stable heading control under diverse manoeuvring conditions. Conventional model predictive control (MPC) can handle system constraints effectively, but its performance depends heavily on manually tuned cost weights. This paper proposes a reinforcement learning-assisted MPC (RL-assisted MPC) framework to improve autonomous vehicle parking performance. A deep Q-network (DQN) agent is trained to dynamically select the cost function weights of an MPC controller, enabling real-time adaptation based on the vehicle's current state. The hybrid approach leverages the predictive optimisation capability of MPC together with the adaptive decision-making of RL, enabling the controller to adjust trade-offs in real time without manual retuning. The framework is evaluated across five different parking scenarios and compared against static-weight MPC baselines. Experimental evaluations demonstrate that the proposed RL-assisted MPC framework achieves comparable or better lateral tracking accuracy, while consistently providing smoother steering behaviour and improved heading stability compared with baseline controllers using static MPC weights. The proposed framework is further evaluated under randomly selected and previously unseen initial vehicle positions, demonstrating its robustness and generalisation across diverse parking configurations. The results demonstrate that RL-assisted MPC improves robustness and generalisation in automated parking systems, highlighting the potential of combining model-based predictive control with RL for autonomous driving.

1 | Introduction

The continuous annual increase in vehicle populations has substantially escalated parking space requirements, rendering parking operations increasingly time-intensive and resource-demanding, particularly within densely populated metropolitan environments [1]. Furthermore, this vehicular proliferation concurrently elevates accident risk, contributing to significant economic losses and infrastructure damage [2].To mitigate these challenges, advanced driver assistance systems (ADASs) have been engineered with diverse functionalities, encompassing autonomous parking systems (APSs), specifically designed to improve vehicular safety and operational convenience [3].

APS architecture typically consists of three fundamental modules: parking spot detection, path planning, and path following [3]. Path planning and path following constitute two essential elements in the behavioural control of autonomous vehicles (AVs) [4]. Path planning synthesises feasible vehicle paths that comply with safety constraints, whereas path tracking controllers utilise real-time state feedback to compute optimal control actions for precise trajectory following. Considerable research efforts have focused on developing efficient trajectory planning methodologies [5]. Nevertheless, path tracking presents significant challenges due to the complex vehicle dynamics and constraints imposed by limited onboard computational and

communication capabilities. Path tracking controllers must synthesise precise control signals in real time while accommodating the imposed computational and communication resource limitations. Multiple control methodologies have been implemented for path tracking applications, including proportional-integral-derivative (PID) controllers, state feedback controllers, and model predictive control (MPC) [6]. MPC has received considerable attention and extensive investigation within AVs research [7–11] and mechatronics [12–14]. MPC is well-suited for real-world autonomous vehicle path-following tasks, as it can effectively manage multi-input multi-output (MIMO) systems while accommodating a wide range of system constraints [15].

Although MPC offers a systematic approach for constraint handling and predictive behaviour modelling, its efficacy is critically influenced by the appropriate tuning of cost function weighting parameters [16]. These weighting parameters establish the relative priority among competing control objectives, such as minimising lateral deviation $d_{y.\max}$, steering effort, and steering rate. Identifying a single static weighting configuration that achieves optimal performance across all scenarios presents significant difficulties, given that varying operational conditions necessitate distinct control trade-offs. This limitation necessitates the development of adaptive control strategies that enable real-time adjustment of weighting parameters based on the instantaneous vehicle state.

In recent years, reinforcement learning (RL) has achieved remarkable success in complex decision-making tasks [17–22]. Unlike the offline design process of MPC, RL optimises the control policy through data-driven adaptation based on the interaction with the environment [23, 24]. In RL, the agent learn optimal policy exclusively through observed state-action transitions and reward signal, eliminating the requirement for a prior system dynamics knowledge [25]. The reward function (RF) represents an estimate of the expected cumulative reward associated with each state. When RF converges to its optimal form, it encapsulates the global optimal information, enabling the derivation of the infinite-horizon optimal control policy. When considering the task of APS, it is challenging to mathematically define RF for an autonomous system [26, 27]. Without a properly defined RF, it becomes difficult for RL to execute parking manoeuvres both safely and efficiently [20, 28].

Additionally, RL training can be extremely time-consuming, often necessitating extensive computational cycles over multiple days to achieve policy convergence. However, in pursuit of globally optimal policies, an RL agent inherently employs exploratory strategies and trial-and-error learning mechanisms, thereby complicating the establishment of formal safety guarantees for the emergent control behaviours [29]. This issue becomes particularly critical in safety-sensitive applications, where safety is often characterised by the system's stability. Recent research efforts have focused on developing safe RL frameworks that aim to ensure stability and constraint satisfaction during both learning and deployment phases [29, 30].

Recent APS research has also explored end-to-end and continuous-action RL that directly outputs low-level control commands such as steering to execute parking manoeuvres [31, 32]. Although such approaches can learn highly nonlinear behaviours, they typically require substantial interaction data and careful reward shaping. Moreover, because safety is not enforced by the learnt policy during training, data collection must be carried out conservatively (e.g., with manual oversight), which increases the cost of generating sufficient experiences.

Given the powerful data-driven optimisation capabilities of RL, combining RL with MPC introduces the ability to incorporate system constraints explicitly, enabling safer and more reliable control while still leveraging real-time data. Simultaneously, by building upon the solid theoretical foundations of MPC, the resulting behaviour of the RL-assisted controller becomes easier to analyse and verify, with respect to stability and constraint satisfaction. These observations motivate the integration of RL and MPC, although only limited pioneering work has been reported to date [33]. Several recent studies have explored different ways to integrate RL and MPC. For example, Lowrey et al. [34] introduced a 'plan online and learn offline' approach, using MPC as a trajectory optimiser to improve exploration and accelerate RL convergence. This was extended in Ref. [35] to an actor–critic framework capable of handling sparse and binary rewards, highlighting MPC's role in propagating global information. As for real-world applications, for instance, Karnchanachari et al. [36], demonstrated the benefits of combining high-level objectives with improved value function representation. However, these efforts primarily focus on enhancing RL performance, with limited attention to system stability. The stability aspect was addressed by Gros et al. in Ref. [37] and further developed by Zanon et al. in Ref. [38] using economic MPC as a function approximator. Building on this, Zanon and Gros [39] incorporated robust linear MPC to ensure constraint satisfaction. While promising, these approaches still require further validation, and few offer both theoretical safety and practical efficiency.

This paper proposes an RL-assisted MPC framework for APS. This hybrid approach combines the adaptability of RL with the predictive optimisation capabilities of MPC. Rather than training a full control policy from scratch, the RL agent selects, at each time step, one of three predefined weight configurations based on the AV current state, enabling the controller to adapt to varying conditions and enhance path-following performance. By leveraging the strengths of both frameworks, the real-time optimisation ability of MPC and the adaptive decision-making capability of RL, the resulting approach improves path-following behaviour for APS without requiring extensive retraining or manual weight tuning, and avoids reliance on a single static set of weights. The proposed method is tested in five parking scenarios. The five parking scenarios are designed to represent a range of parking configurations with increasing geometric complexity and manoeuvring difficulty. Scenario 1 (Figure 1) represents the simplest case, serving as a baseline parking manoeuvre. Scenario 2 (Figure 2) is identical to scenario one in terms of geometry but starts with a different initial heading, which forces the vehicle to temporarily move outside the nominal path range before returning, thereby testing the controller under steering angle limitations. Scenario 3 (Figure 3) introduces the longest trajectory with two sharp corners and is intended to evaluate the controller's performance under challenging manoeuvres and potential path-planning imperfections.

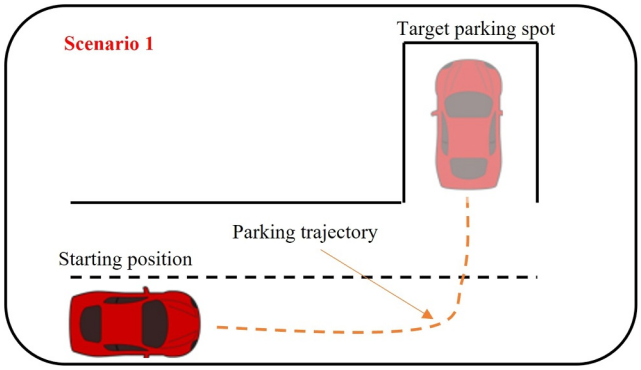

FIGURE 1 | Reference trajectory used for the parking manoeuvre, showing the AV's planned path from the starting position to the designated parking spot for scenario 1.

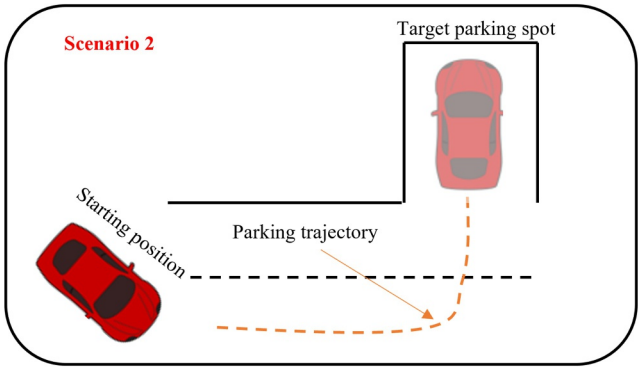

FIGURE 2 | Reference trajectory used for the parking manoeuvre, showing the AV's planned path from the starting position to the designated parking spot for scenario 2.

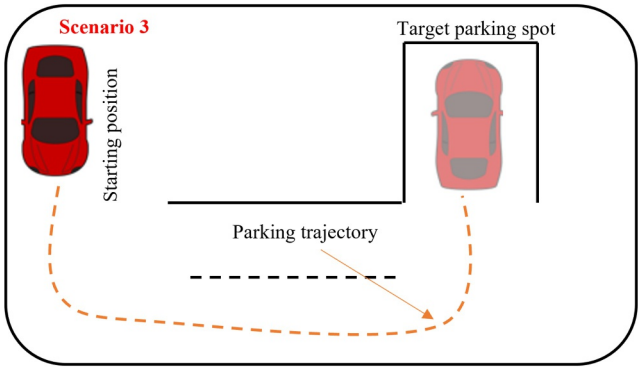

FIGURE 3 | Reference trajectory used for the parking manoeuvre, showing the AV's planned path from the starting position to the designated parking spot for scenario 3.

Scenario 4 (Figure 4) is derived from a real-world parking situation and includes both a sharp turning requirement and a tilted parking spot, making the manoeuvre significantly more demanding. Scenario 5 (Figure 5) is also obtained from a real parking configuration and represents a perpendicular parking scenario, further increasing the manoeuvring constraints. Together, these scenarios provide a comprehensive assessment of the adaptability and robustness of the proposed RL-assisted MPC framework. Each scenario starts and ends with varying parking spots.

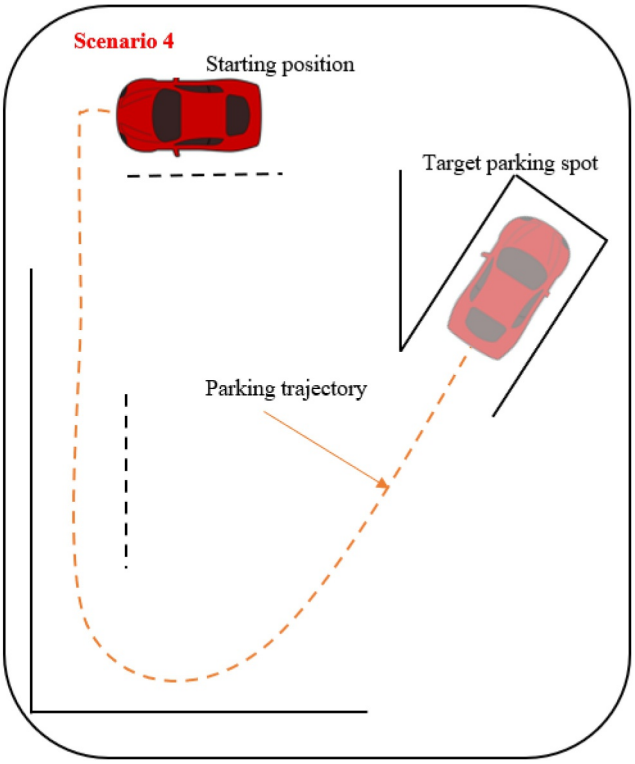

FIGURE 4 | Reference trajectory used for the parking manoeuvre, showing the AV's planned path from the starting position to the designated parking spot for scenario 4.

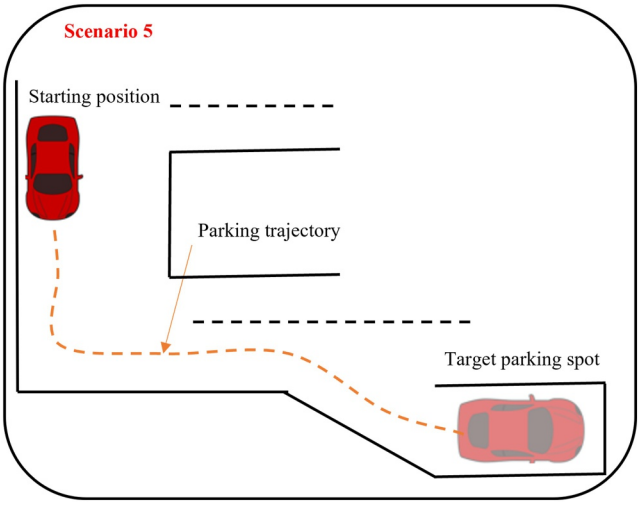

FIGURE 5 | Reference trajectory used for the parking manoeuvre, showing the AV's planned path from the starting position to the designated parking spot for scenario 5.

A preliminary version of this work was published as a conference paper [40], which presents results for a single autonomous parking scenario using RL-assisted MPC. The present paper substantially expands on the conference version by extending the framework to five distinct parking scenarios, incorporating a dynamic speed profile based on vehicle–target distance, providing a comprehensive evaluation of $d_{y,\max}$, steering effort, smoothness, heading stability, and policy adoption, and expanding discussion, results analysis, and conclusions, supported by additional figures and tables.

The remainder of the paper is organised as follows. Section 2 presents the proposed methodology, including the RL-assisted MPC framework and system design. Section 3 discusses the experimental results and performance evaluation. Finally, Section 4 concludes the paper and outlines directions for future work.

2 | Methodology

The objective is to allow an AV to follow a predefined path during parking (Figure 1) with minimal lateral deviation d_y, while ensuring smooth steering and stable heading transitions, all within the limits of safety constraints. The AV operates in a structured environment modelled using a kinematic bicycle model (Figure 6). The reference trajectory is generated using a Bézier curve.

2.1 | Béizer Curve

Bézier curves are widely used in trajectory planning tasks due to their smoothness, controllability, and ability to precisely interpolate between initial and final states. A Bézier curve is a parametric curve defined by a set of control points, where the trajectory is generated as a weighted sum of these points using Bernstein polynomials. The most common form for trajectory generation is the cubic Bézier curve, which uses four control points and ensures smooth entry and exit tangents aligned with the desired paths [41]. In the context of APS, Bézier curves are particularly attractive because they allow the vehicle to generate collision-free and smooth trajectories with minimal computational cost [42]. Given their inherent geometric properties, the generated path naturally satisfies curvature continuity, which reduces the steering effort and enhances trajectory tracking accuracy. The curve formulation is flexible and can be easily adapted to various parking scenarios by adjusting the positions of the control points [42].

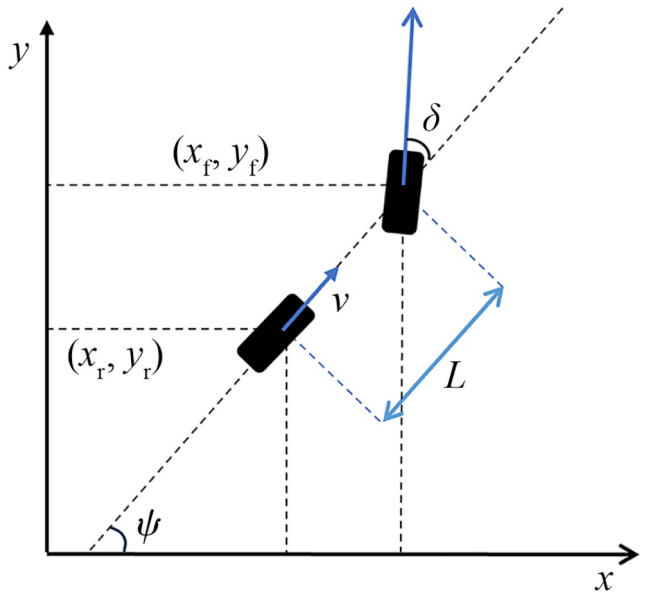

FIGURE 6 | Kinematic bicycle model of the vehicle referenced at the rear axle centre.

In this work, Bézier curves are employed to construct the reference parking path as mentioned in Equation (1).

$$P(t_c) = \sum_{i_c=0}^{N} \binom{N}{i_c} (1-t_c)^{N-i_c} t_c^{i_c} P_{i_c}, \quad t_c \in [0,1], \tag{1}$$

where $P(t_c)$ represents the points of the Bézier curve, N represents the degree of the Bézier curve, t_c represents the interpretation of the Bézier curve, P_{i_c} represents the i_c^{th} control point and i_c represent the index counter where $i_c = 0, 1, 2, ..., n$. Scenario 1 shows the simplest path with only two control points (P_1 and P_2), initial vehicle position (P_0), parking target position (P_3) represented by Equation (2), as follows:

$$P(t_c) = (1-t_c)^3 P_0 + 3(1-t_c)^2 t_c P_1 + 3(1-t_c) t_c^2 P_2 + t_c^3 P_3. \tag{2}$$

The number of control points depends on the shape and the complexity of the curve, starting from two points for scenario 1 up to 12 points for scenario 3.

2.2 | Vehicle Dynamics

In APS, due to the low-speed nature of the manoeuvres, the kinematic bicycle model provides a convenient and accurate approximation of the vehicle dynamics [43] as shown in Figure 6. The rear axle centre velocity is given by

$$v = x_r \cos\psi + y_r \sin\psi, \tag{3}$$

where ψ represents the heading of the AV.

The non-holonomic constraint equations for the front and rear wheels are given as follows:

$$\begin{aligned} \dot{x}_f \sin(\psi+\delta) - \dot{y}_f \cos(\psi+\delta) = 0, \\ x_r \sin(\psi) - y_r \cos(\psi) = 0, \end{aligned} \tag{4}$$

where $\dot{x}_f$ and $\dot{y}_f$ are the velocity of the front wheel in the x-axis and y-axis, respectively. δ represents the steering angle.

Combining Equation (3) with Equation (4) yields

$$\begin{aligned} \dot{x}_r &= v\cos(\psi), \\ \dot{y}_r &= v\sin(\psi). \end{aligned} \tag{5}$$

Based on the geometric relationship of the front and rear wheels, the following expressions can be derived:

$$\begin{aligned} x_f &= x_r + L\cos(\psi), \\ y_f &= y_r + L\sin(\psi), \end{aligned} \tag{6}$$

where L represents the distance between the front and rear wheels.

The steering angle can be represented as

$$\delta = \arctan\frac{L\dot{\psi}}{v(d)}. \tag{7}$$

where $v(d)$ (Equation (9)) represents the vehicle speed and its value depends on the vehicle's distance d (Equation (10)) to the centre of the target parking spot (TPS).

The kinematic bicycle model, referenced to the centre of the rear axle, can be described as follows:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{\tan\delta}{L} \end{bmatrix} v(d), \tag{8}$$

$v(d)$ is defined as

$$v(d) = \begin{cases} v, & d \geq 4 \text{ m}, \\ 0.9v, & 3 \leq d < 4 \text{ m}, \\ 0.8v, & 2 \leq d < 3 \text{ m}, \\ 0.4v, & 0.1 \leq d < 2 \text{ m}, \\ 0, & d < 0.1 \text{ m}. \end{cases} \tag{9}$$

where d is the Euclidean distance from the vehicle to TPS calculated as

$$d = \sqrt{\left(x_{vp} - x_{TPS}\right)^2 + \left(y_{vp} - y_{TPS}\right)^2}, \tag{10}$$

where (x_{vp}, y_{vp}) represents the current vehicle's position and (x_{TPS}, y_{TPS}) represents the x and y of TPS.

2.3 | RL-Assisted MPC Control System Design

The MPC controller is designed to enable precise and adaptive trajectory tracking for APS. It employs a predictive vehicle model and solves a constrained optimisation problem at each control step, allowing real-time adjustment of control inputs based on the predicted vehicle trajectory. This structure allows the controller to adapt its behaviour in real time based on the current AV states x_r, y_r, and ψ.

The optimisation process is subject to physical limitations of the system specifically (constraints on δ). By solving this constrained optimisation problem, the MPC computes an optimal control sequence u^*, of which only the first input is applied to the vehicle dynamics. In this paper, u represents δ. The updated vehicle state is then used in the subsequent prediction cycle, enabling closed-loop control in a receding horizon fashion until the parking task is complete or terminated.

To enhance adaptability across varying conditions, an RL agent is utilised. At each time step, the agent observes the current observation, which consists of ψ and d. Based on this state, the agent selects a set of weights $\boldsymbol{W}$ from three baselines $\boldsymbol{W} \in \{\boldsymbol{W}_1, \boldsymbol{W}_2, \boldsymbol{W}_3\}$ to be used in the MPC cost function. The controller then computes a steering command δ, which is applied to the vehicle. This interaction produces updated states for both the RL agent and MPC controller. The process repeats until the parking task is either successfully completed or terminated, as shown in Figure 7.

In this work, the action space of the RL agent is intentionally defined as discrete. The use of a discrete action space is a deliberate design choice aimed at balancing adaptability, stability, and computational efficiency within the proposed RL-assisted MPC framework. By restricting the RL agent to select from a small set of carefully designed MPC weight configurations, the controller preserves the inherent stability and constraint-handling properties of the MPC while avoiding excessive exploration that could compromise safety during training and deployment. In contrast, continuous action space formulations do not provide inherent safety guarantees and may lead to unstable or unsafe weight selections, particularly during early training stages. Furthermore, continuous-action RL typically requires substantially larger amounts of interaction data to achieve reliable convergence, which can be impractical for safety-critical and real-time autonomous parking applications. The discrete formulation therefore enables safer learning behaviour while significantly reducing training complexity and computational burden.

The MPC problem is formulated as a constrained optimisation problem that is solved at each control step. Specifically, the objective is to minimise a cost function J over a finite prediction

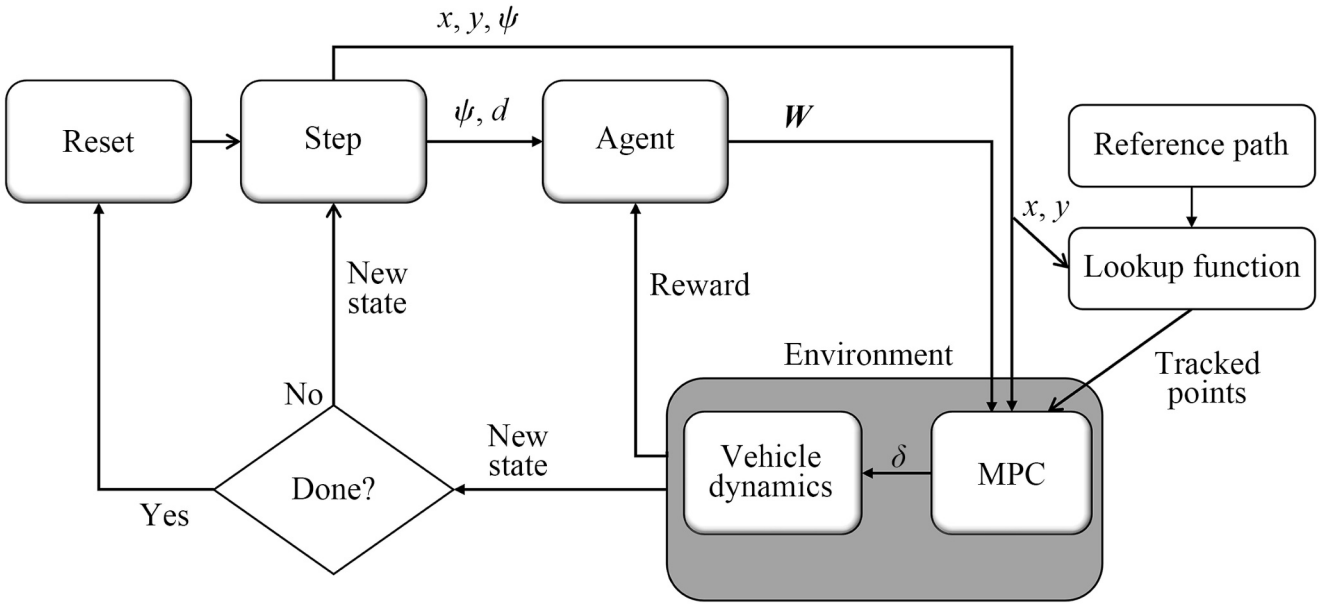

FIGURE 7 | Flowchart of RL-assisted MPC. $\boldsymbol{W}$ represents a single set of weights.

horizon p (Equation 11a), subject to vehicle dynamics and steering constraints (Equation 11c). The optimisation problem consists of three weighted terms, each serving a specific purpose in shaping the vehicle's behaviour. The first term penalises the deviation from the reference path by minimising the squared Euclidean distance between the predicted vehicle position and the corresponding reference point at each step. The second term penalises large control inputs by minimising the squared value of δ, encouraging smoother and more stable manoeuvres. The third term penalises abrupt changes in control input by minimising the difference between successive δ's, thereby promoting steering smoothness. The optimisation problem computed at each control step is defined as follows [6]:

$$\min_{\delta} \quad J = \sum_{k=1}^{p} \lambda_{\mathrm{t}} \|\boldsymbol{X}_k - \boldsymbol{X}_{\mathrm{ref},k}\|^2 + \sum_{k=0}^{p-1} \lambda_{\mathrm{s}} \delta_k^2 + \sum_{k=0}^{p-1} \lambda_{\mathrm{d}} (\delta_k - \delta_{k-1})^2 \tag{11a}$$

$$\text{subject to} \quad \text{vehicle model (Equation 8)} \tag{11b}$$

$$\frac{-\pi}{4} \leq \delta \leq \frac{\pi}{4}, \tag{11c}$$

where $\boldsymbol{X}_k = [x_k, y_k]^{\mathrm{T}}$ denote the vehicle position at time step k, and let $\boldsymbol{X}_{\mathrm{ref},k} = [x_{\mathrm{ref},k}, y_{\mathrm{ref},k}]^{\mathrm{T}}$ represent the corresponding reference point. The predicted steering input at time step k is denoted by δ_k, and the prediction horizon is given by p. The cost function includes three weighting parameters: λ_{t} for lateral tracking error, λ_{s} for steering effort, and λ_{d} for steering rate to encourage smoothness.

Since the MPC controller does not regulate the longitudinal speed, d_y from the reference path serves as the primary metric for evaluating tracking performance. In this work, the vehicle's alignment with the reference path is assessed by identifying the closest points ahead and behind the current vehicle position. This is achieved through a custom function that calculates the perpendicular distances to the path, dynamically determining the lateral offset. This deviation is then penalised within the cost function to ensure precise trajectory tracking. The overall structure is illustrated in Figure 8, and the mathematical formulation is as follows [6]:

$$d_y = \frac{\left|(x_2 - x_1)\left(y_1 - p_y\right) - (x_1 - x_k)(y_2 - y_k)\right|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}. \tag{12}$$

R_{F} at the time step t is defined as

$$R_{\mathrm{F}_t} = \begin{cases} -|d_y|, & \text{if } t \text{ is terminal} \\ -\left(\alpha_l |d_y| + \alpha_h \left|\ddot{\psi}\right|\right), & \text{otherwise,} \end{cases} \tag{13}$$

where α_{l} is $d_{y.\max}$ reward weight, α_{h} is heading error reward weight, and $\ddot{\psi}$ is yaw rate (heading acceleration).

The full control logic of the RL-assisted MPC is outlined in Algorithm 1, where t represents the time, ϕ represents the parameters in the network Q, ϕ^- represents the target parameters in Q, and a represents the action.

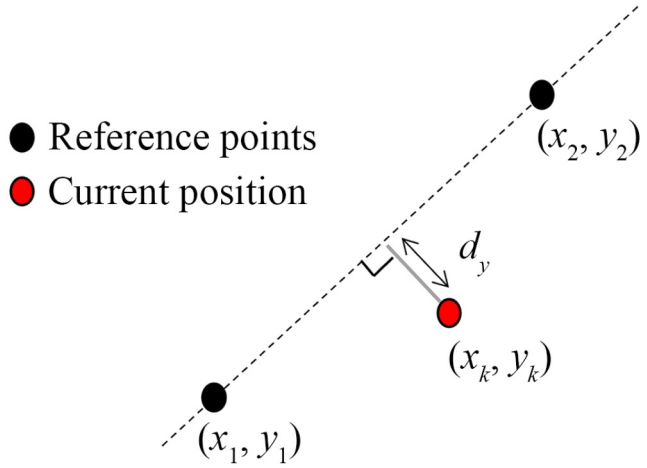

FIGURE 8 | Illustration of d_y from the reference path, computed as the perpendicular distance from the vehicle's current position to the segment between two consecutive reference points.

ALGORITHM 1 | RL-Assisted MPC

Initialise ϕ, ϕ^-, buffer $\mathcal{D}$, and AV states
for episode = 1 to M **do**
  Observe AV state $\boldsymbol{S} \leftarrow \{\psi, d, x, y\}$.
  for t = 1 to T **do**
    Following ϵ-greedy, select weights.
    $\boldsymbol{W}_t \leftarrow *argmax_a Q_\phi(\psi, d, a)$
    Apply $\boldsymbol{W}_t$ to MPC cost function.
    Solve MPC and compute steering command δ_t.
    Step in the environment using δ_t; observe $\boldsymbol{S}_{t+1}$ and r_t.
    Store tuple $([\psi_t, d_t], \boldsymbol{W}_t, r_t, [\psi_{t+1}, d_{t+1}])$ in $\mathcal{D}$.
    Sample a minibatch $([\psi_i, d_i], \boldsymbol{W}_t, r_i, [\psi_{i+1}, d_{i+1}]) \sim \mathcal{D}$.
    Compute target
$$y_i = \begin{cases} r_i, & \text{if } i+1 \text{ terminates,} \\ r_i + \gamma \max_{\boldsymbol{W}'} Q_{\phi^-}([\psi_{i+1}, d_{i+1}], \boldsymbol{W}'), & \text{otherwise.} \end{cases}$$
    Update ϕ to minimise $(y_i - Q_\phi([\psi_i, d_i]))^2$.
    Every c steps, update $\phi^- \leftarrow \phi$.
  end for
end for

3 | Results

This section presents a comprehensive evaluation of the proposed RL-assisted MPC framework across five distinct parking scenarios, as shown in Figures 1–5. Each scenario is designed to test the system under varying levels of complexity and curvature. Furthermore, the proposed RL-assisted MPC has been studied under four distinct reward weight configurations (cases 1–4) for each individual scenario, as defined in Tables 1–5. The RL-assisted MPC performance is compared against three MPC baselines ($\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$) with fixed cost function weights detailed in Tables 6–10. In addition to the fixed-weight MPC baselines, a rule-based MPC controller is also evaluated using the same performance metric in scenario 5. All experiments are conducted under identical initial conditions and reference path

TABLE 1 | Reward weights used in RL-assisted MPC training cases for scenario 1.

RL-assisted MPC scenario 1	α_l	α_h
Case 1	0.75	0.25
Case 2	0.78	0.22
Case 3	0.80	0.20
Case 4	0.82	0.18

TABLE 2 | Reward weights used in RL-assisted MPC training cases for scenario 2.

RL-assisted MPC scenario 2	α_l	α_h
Case 1	0.65	0.35
Case 2	0.68	0.32
Case 3	0.70	0.30
Case 4	0.73	0.27

TABLE 3 | Reward weights used in RL-assisted MPC training cases for scenario 3.

RL-assisted MPC scenario 3	α_l	α_h
Case 1	0.70	0.30
Case 2	0.75	0.25
Case 3	0.80	0.20
Case 4	0.85	0.15

TABLE 4 | Reward weights used in RL-assisted MPC training cases for scenario 4.

RL-assisted MPC scenario 4	α_l	α_h
Case 1	0.75	0.25
Case 2	0.78	0.22
Case 3	0.80	0.20
Case 4	0.83	0.17

TABLE 5 | Reward weights used in RL-assisted MPC training cases for scenario 5.

RL-assisted MPC scenario 5	α_l	α_h
Case 1	0.73	0.27
Case 2	0.75	0.25
Case 3	0.80	0.20
Case 4	0.82	0.18

TABLE 6 | Weight sets used in the MPC cost function for the three fixed-weight baseline controllers for scenario 1.

Weight	λ_t	λ_s	λ_d
$\boldsymbol{W}_1$	6	2	0
$\boldsymbol{W}_2$	2	2	8
$\boldsymbol{W}_3$	1	0	5

TABLE 7 | Weight sets used in the MPC cost function for the three fixed-weight baseline controllers for scenario 2.

Weight	λ_t	λ_s	λ_d
$\boldsymbol{W}_1$	1.00	0.10	0.10
$\boldsymbol{W}_2$	0.50	0.80	0.10
$\boldsymbol{W}_3$	0.20	0.15	1.00

TABLE 8 | Weight sets used in the MPC cost function for the three fixed-weight baseline controllers for scenario 3.

Weight	λ_t	λ_s	λ_d
$\boldsymbol{W}_1$	5.00	0.15	0.10
$\boldsymbol{W}_2$	2.00	3.00	0.10
$\boldsymbol{W}_3$	1.00	1.00	8.00

TABLE 9 | Weight sets used in the MPC cost function for the three fixed-weight baseline controllers for scenario 4.

Weight	λ_t	λ_s	λ_d
$\boldsymbol{W}_1$	1.2	0.1	0.1
$\boldsymbol{W}_2$	0.8	1.0	0.1
$\boldsymbol{W}_3$	0.2	0.1	1.0

TABLE 10 | Weight sets used in the MPC cost function for the three fixed-weight baseline controllers' scenario 5.

Weight	λ_t	λ_s	λ_d
$\boldsymbol{W}_1$	1.5	0.1	0.1
$\boldsymbol{W}_2$	0.9	0.7	0.2
$\boldsymbol{W}_3$	0.3	0.1	1.0

per scenario to ensure a fair comparison. Each RL training case corresponds to a specific setting of the reward weights α_l and α_h, which satisfy $\alpha_l + \alpha_h = 1$.

Key evaluation metrics include the maximum d_y over time ($d_{y.\max}$), steering rate (δ_r), total variation in steering input (δ_{tv}), d_y, heading acceleration ($d^2\psi$), and the total variation in steering rate (δ_{tvr}). These metrics collectively provide insight into the tracking accuracy, control smoothness, and adaptive stability of the controller under different operating scenarios.

3.1 | Lateral Deviation Comparison

The RL-assisted MPC consistently achieves $d_{y.\max}$ that are comparable to, and in several cases lower than, fixed-weight baselines, as shown in Figures 9–13. For instance, in scenario 1 (Table 11), case 3 achieves a $d_{y.\max}$ of 0.2145 m, nearly identical to the best baseline $\boldsymbol{W}_1$, 0.2043 m. Although other RL cases show slightly higher $d_{y.\max}$, this shows that the adaptive framework can reach baseline-level accuracy in this manoeuvre. The reason for not outperforming it is to take in consideration a better softer δ which indeed achieves better results. In scenario 2 (Table 12), case 1 obtained 0.6714 m, which is lower than all

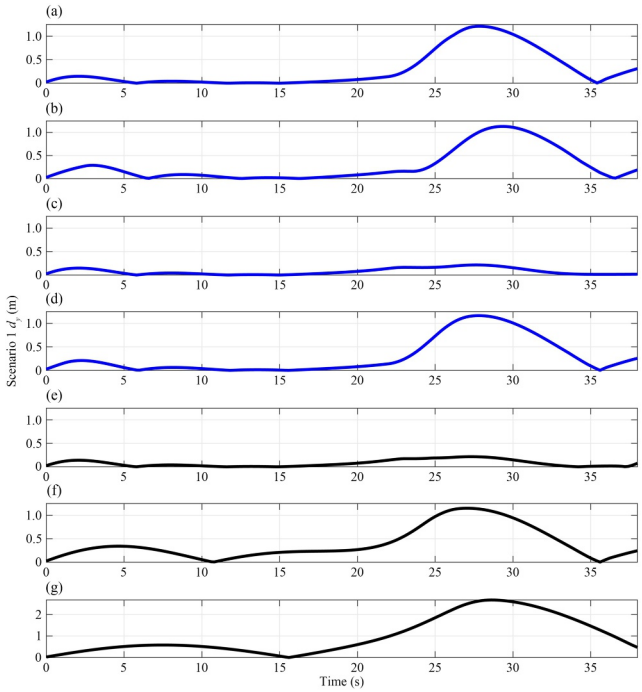

FIGURE 9 | d_y over time for RL-assisted MPC cases and baseline controllers for scenario 1: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

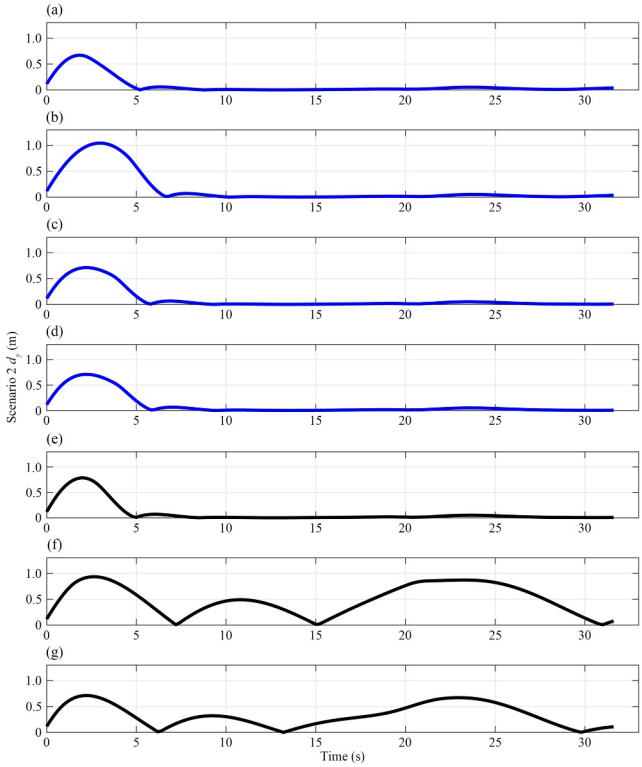

FIGURE 10 | d_y over time for RL-assisted MPC cases and baseline controllers for scenario 2: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

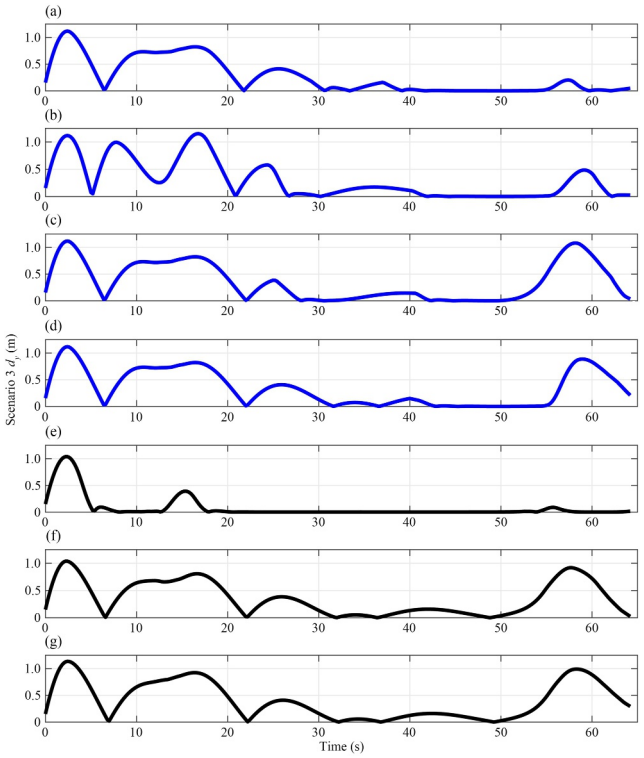

FIGURE 11 | d_y over time for RL-assisted MPC cases and baseline controllers for scenario 3: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

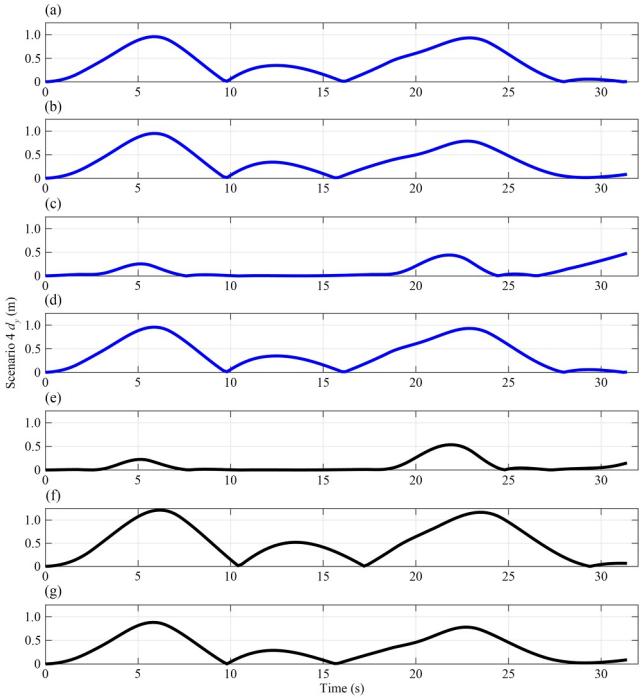

FIGURE 12 | d_y over time for RL-assisted MPC cases and baseline controllers for scenario 4: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

baseline $d_{y.\max}$ (0.7874–0.9358 m). Furthermore, RL achieves the lowest total steering variation. In scenario 3 (Table 13), none of the RL cases fully surpasses the best baseline $d_{y.\max}$ (1.0376 m), with results ranging from 1.1175 to 1.1910 m. However, $d_{y.\max}$ remains within a narrow margin, indicating that the adaptive controller maintains stable tracking even under complex geometry, and further trade-offs in steering smoothness (further discussed in Section 3.2) favour the RL-assisted MPC controller. In scenario 4 (Table 14), case 3 clearly outperforms all baselines, with a $d_{y.\max}$ of 0.4816 m compared with 0.5326–1.2164 m. This shows the RL agent's ability to reduce error by dynamically selecting more suitable weight sets. Finally, in scenario 5 (Table 15), case 4 achieves the lowest $d_{y.\max}$ across all baselines at 0.2011 m, outperforming the best baseline $\boldsymbol{W}_3$ (0.2160 m). This result highlights the benefit of adaptive weighting for precise final positioning in tighter manoeuvres. The rule-based controller exhibits a higher $d_{y.\max}$ of 0.3187 m, indicating inferior tracking performance compared with the proposed RL-assisted MPC.

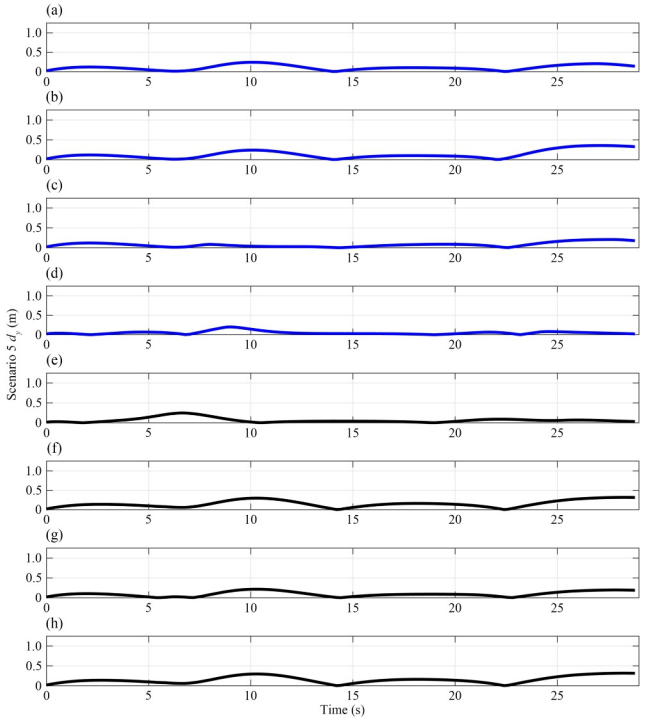

FIGURE 13 | d_y over time for RL-assisted MPC cases and baseline controllers for scenario 5: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$; (h) rule-based MPC.

TABLE 11 | Performance metrics across RL-assisted MPC and fixed-weight MPC cases for scenario 1.

Controller scenario 1	$d_{y.\max}$ (m)	δ_{tvr}	δ_{tv}
RL-assisted MPC case 1	1.2131	22.4619	25.0362
RL-assisted MPC case 2	1.1317	23.3324	21.2433
RL-assisted MPC case 3	0.2145	12.0842	9.8560
RL-assisted MPC case 4	1.1662	17.9587	13.2100
MPC $\boldsymbol{W}_1$	0.2043	13.5871	10.9559
MPC $\boldsymbol{W}_2$	1.1145	9.9164	4.5771
MPC $\boldsymbol{W}_3$	2.6365	8.6211	5.8627

TABLE 12 | Performance metrics across RL-assisted MPC and fixed-weight MPC cases for scenario 2.

Controller scenario 2	$d_{y.\max}$ (m)	δ_{tvr}	δ_{tv}
RL-assisted MPC case 1	0.6714	22.0034	4.0571
RL-assisted MPC case 2	1.0462	24.9393	5.0454
RL-assisted MPC case 3	0.7130	26.7528	5.3948
RL-assisted MPC case 4	0.7130	24.4939	5.3948
MPC $\boldsymbol{W}_1$	0.7874	27.6094	5.1092
MPC $\boldsymbol{W}_2$	0.9358	26.7528	3.3444
MPC $\boldsymbol{W}_3$	0.7930	24.4939	3.0571

TABLE 13 | Performance metrics across RL-assisted MPC and fixed-weight MPC cases for scenario 3.

Controller scenario 3	$d_{y.\max}$ (m)	δ_{tvr}	δ_{tv}
RL-assisted MPC case 1	1.1524	58.2192	6.1582
RL-assisted MPC case 2	1.1175	29.1807	11.4425
RL-assisted MPC case 3	1.1910	62.5398	9.6677
RL-assisted MPC case 4	1.1175	62.5398	8.3953
MPC $\boldsymbol{W}_1$	1.0376	32.3245	6.9936
MPC $\boldsymbol{W}_2$	1.0376	20.1160	4.9584
MPC $\boldsymbol{W}_3$	1.1354	11.0020	4.7309

TABLE 14 | Performance metrics across RL-assisted MPC and fixed-weight MPC cases for scenario 4.

Controller scenario 4	$d_{y.\max}$ (m)	δ_{tvr}	δ_{tv}
RL-assisted MPC case 1	0.9577	14.9410	3.8390
RL-assisted MPC case 2	0.9507	12.3650	3.7927
RL-assisted MPC case 3	0.4816	19.1405	3.9413
RL-assisted MPC case 4	0.9577	14.9410	3.8390
MPC $\boldsymbol{W}_1$	0.5326	25.3487	4.7574
MPC $\boldsymbol{W}_2$	1.2164	13.5047	3.8005
MPC $\boldsymbol{W}_3$	0.8819	10.2370	3.6680

TABLE 15 | Performance metrics across RL-assisted MPC and fixed-weight MPC cases for scenario 5.

Controller scenario 5	$d_{y.\max}$ (m)	δ_{tvr}	δ_{tv}
RL-assisted MPC case 1	0.2400	22.6566	3.9686
RL-assisted MPC case 2	0.3571	18.5000	3.3090
RL-assisted MPC case 3	0.2083	22.3993	3.9515
RL-assisted MPC case 4	0.2011	42.1801	6.7398
MPC $\boldsymbol{W}_1$	0.2473	40.5182	6.3254
MPC $\boldsymbol{W}_2$	0.3180	23.0532	3.7998
MPC $\boldsymbol{W}_3$	0.2160	17.1130	3.2572
Rule-based MPC	0.3187	29.0364	3.8703

3.2 | Steering Effort and Smoothness

Steering effort and smoothness are evaluated using the δ_{tv} and δ_{tvr}. These two metrics reflect the stability of the vehicle steering. In each scenario, at least one RL case produces smoother steering than the static baselines, even when not always yielding the lowest $d_{y.max}$. In scenario 1 (Table 11), case 3 yields the lowest δ_{tv} (9.8560) among the RL cases, outperforming the best baseline $\boldsymbol{W}_1$ (10.9559) but with reduced oscillations visible in Figure 14. Although baseline $\boldsymbol{W}_3$ shows lower steering variability, it does so at the cost of a much larger $d_{y.max}$ (2.6365 m), demonstrating an unfavourable trade-off. The RL-assisted MPC framework, especially case 3, achieves a more balanced compromise between steering smoothness and tracking accuracy. In scenario 2 (Table 12), case 1 achieves the lowest δ_{tv} (4.0571), outperforming baselines $\boldsymbol{W}_1$ (5.1092), except baseline $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$, while simultaneously offering lower $d_{y.max}$ than baseline $\boldsymbol{W}_3$. This confirms that the RL agent can minimise steering variability without sacrificing lateral performance. In scenario 3 (Table 13), despite slightly higher $d_{y.max}$, RL case 2 produces a moderate steering effort δ_{tv} of 11.4425 compared with baselines that either increase variation (baseline $\boldsymbol{W}_1$ (6.9936)) or reduce it excessively at the cost of responsiveness (baseline $\boldsymbol{W}_3$ 4.7309). Figures 14–18 illustrate how the RL-assisted MPC controller smooths steering transitions, reducing abrupt spikes that are present in the baseline controller. Furthermore, case 2 in scenario 4 achieves 3.7927 in δ_{tv}, nearly identical to the best-performing (baseline $\boldsymbol{W}_3$ (3.6680)), while keeping δ_{tvr} lower than most baselines. Showing that the RL agent adapts well to smooth trajectories while preserving easy controllability, as shown in Table 14. In scenario 5 (Table 15), case 2 offers the most favourable trade-off, with δ_{tv} (3.3090) compared with baselines ranging from 3.2572 to 6.3254. Although baseline $\boldsymbol{W}_3$ has a marginally lower value of 3.2572, the RL-assisted MPC controller achieves smoother steering without the high variability in steering rate seen in static configurations. The rule-based controller achieves 29.0364 of δ_{tvr} and 3.8703 for δ_{tv}, which are higher than those obtained using the RL-assisted MPC.

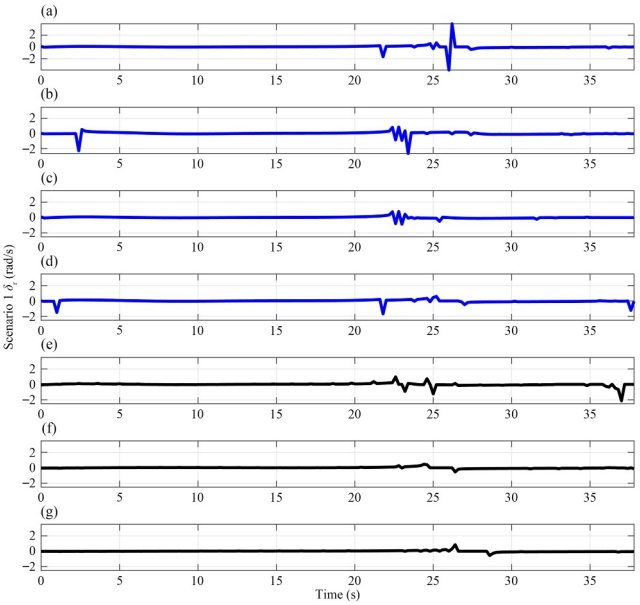

FIGURE 14 | δ_r over time for RL-assisted MPC cases and baseline controllers for scenario 1: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

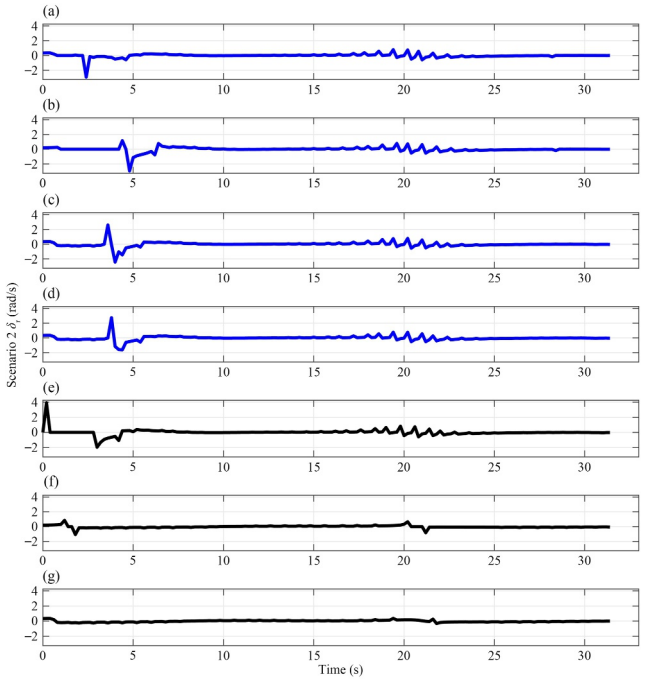

FIGURE 15 | δ_r over time for RL-assisted MPC cases and baseline controllers for scenario 2: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

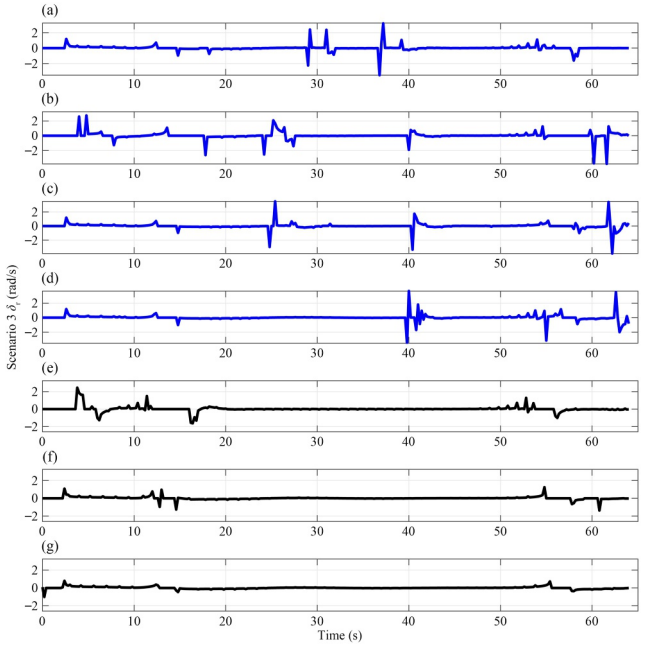

FIGURE 16 | δ_r over time for RL-assisted MPC cases and baseline controllers for scenario 3: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

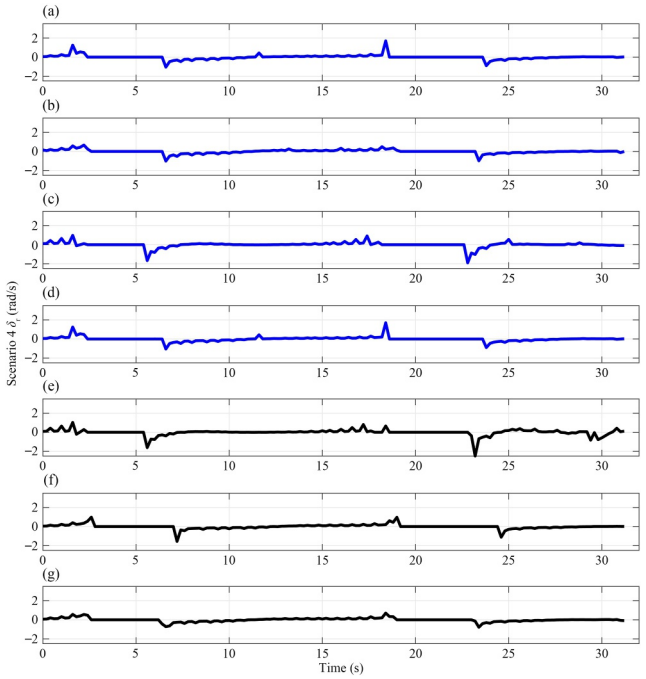

FIGURE 17 | δ_r over time for RL-assisted MPC cases and baseline controllers for scenario 4: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

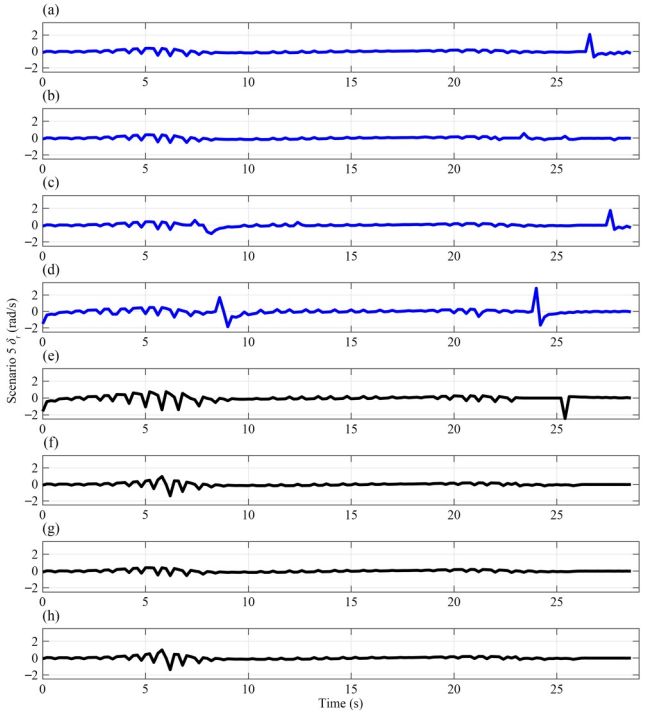

FIGURE 18 | δ_r over time for RL-assisted MPC cases and baseline controllers for scenario 5: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$; (h) represents rule-based MPC.

3.3 | Heading Stability

Heading stability is evaluated using heading acceleration $\mathrm{d}^2\psi$ illustrated in Figures 19–23, which captures the smoothness of directional changes. Large oscillations in this metric indicate abrupt or unstable manoeuvres, whereas lower and more consistent values correspond to smoother more comfortable trajectories. In most scenarios, at least one RL case exhibited reduced heading jerk compared with static baselines. In scenario 1 (Figure 19), case 3 achieves both low $d_{y.\max}$ and favourable steering smoothness, and also demonstrates stable heading behaviour. Oscillations are limited and more damped compared with baseline $\boldsymbol{W}_1$ (the best in $d_{y.\max}$), confirming the RL agent's balanced adaptation in this manoeuvre. It is worth noting that even though baseline $\boldsymbol{W}_2$ and $\boldsymbol{W}_3$ have lower margins, the oscillations in both last longer and suffer from high $d_{y.\max}$.

In scenario 2 (Figure 20), case 1 outperforms baselines in $d_{y.\max}$ and steering variation, exhibiting the smoothest heading profile. $\mathrm{d}^2\psi$ spikes are smaller and less frequent than in baselines, supporting the conclusion that case 1 provides the most stable and low $d_{y.\max}$ in this scenario.

In scenario 3 (Figure 21), even though RL case 2 is selected for smoother steering effort despite slightly higher $d_{y.\max}$, its $\mathrm{d}^2\psi$ remains more stable than baselines $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. Although baseline $\boldsymbol{W}_3$ displays lower oscillations, it comes at the cost of poor lateral performance, reinforcing that the RL case offers a better overall trade-off.

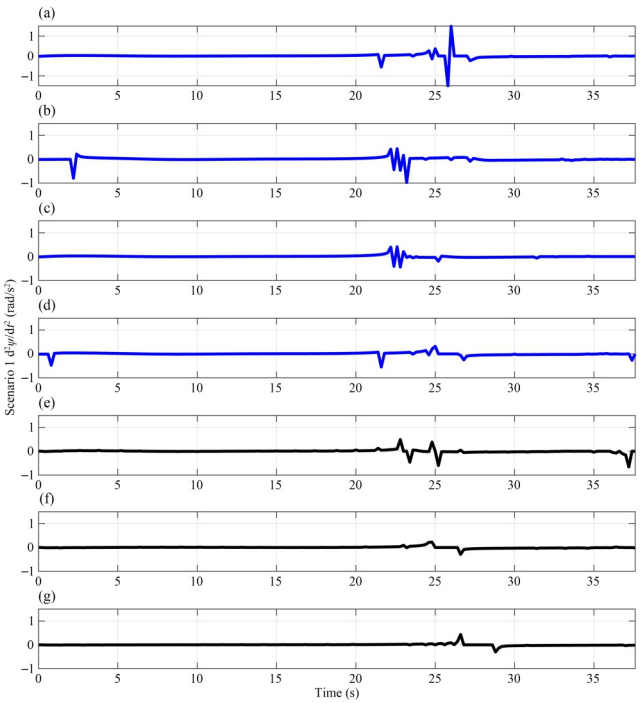

FIGURE 19 | $\mathrm{d}^2\psi/\mathrm{d}t^2$ over time for RL-assisted MPC cases and baseline controllers for scenario 1: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

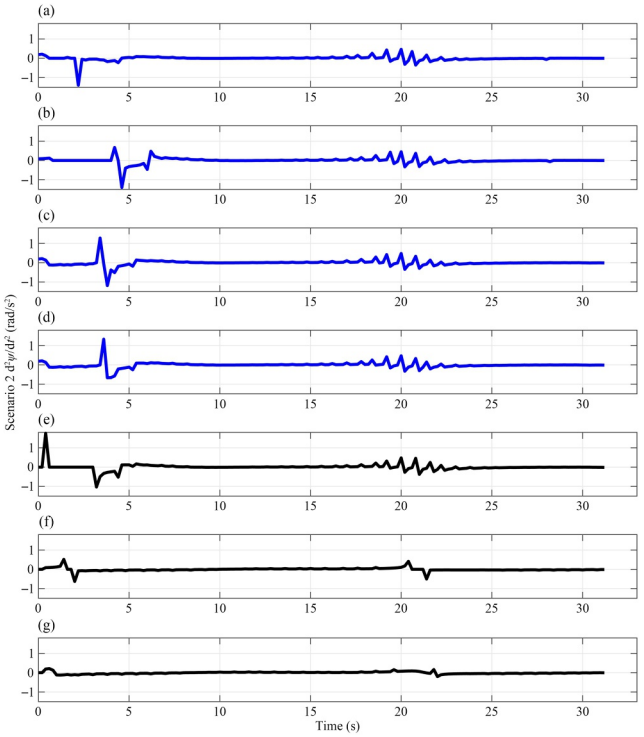

FIGURE 20 | $d^2\psi/dt^2$ over time for RL-assisted MPC cases and baseline controllers for scenario 2: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

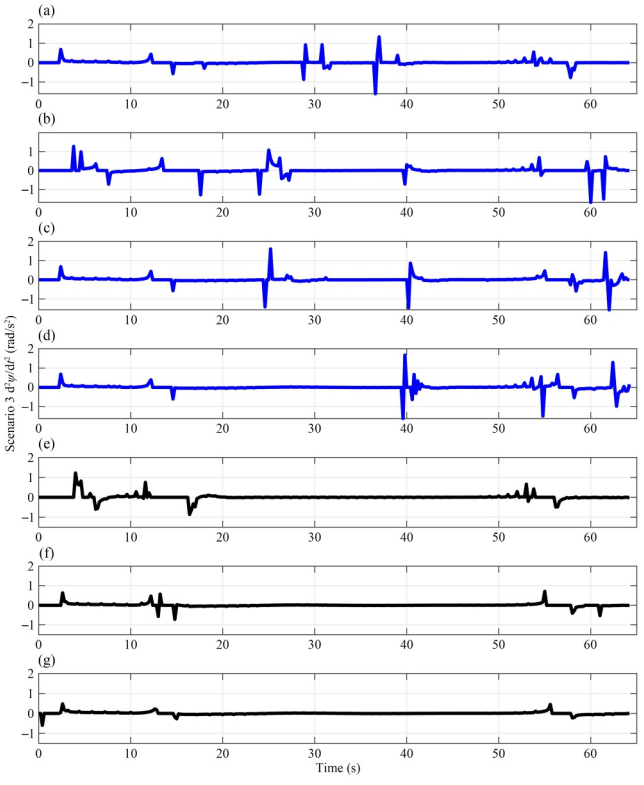

FIGURE 21 | $d^2\psi/dt^2$ over time for RL-assisted MPC cases and baseline controllers for scenario 3: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

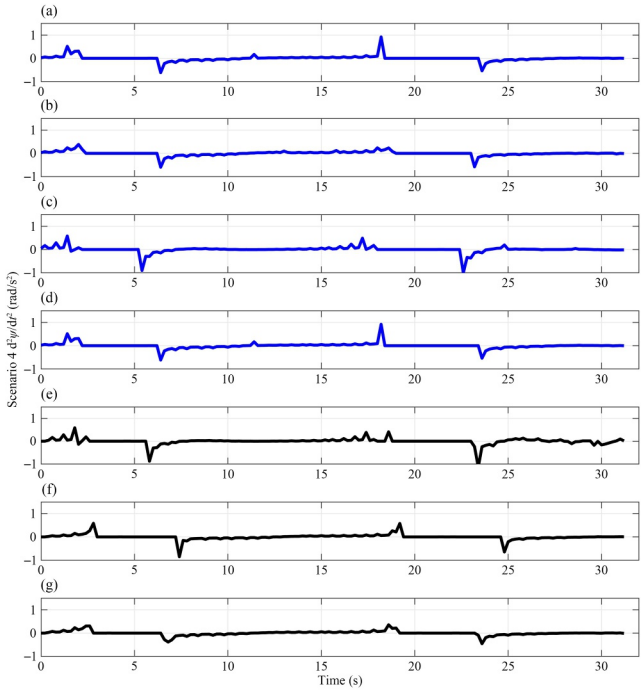

FIGURE 22 | $d^2\psi/dt^2$ over time for RL-assisted MPC cases and baseline controllers for scenario 4: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) represents MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$.

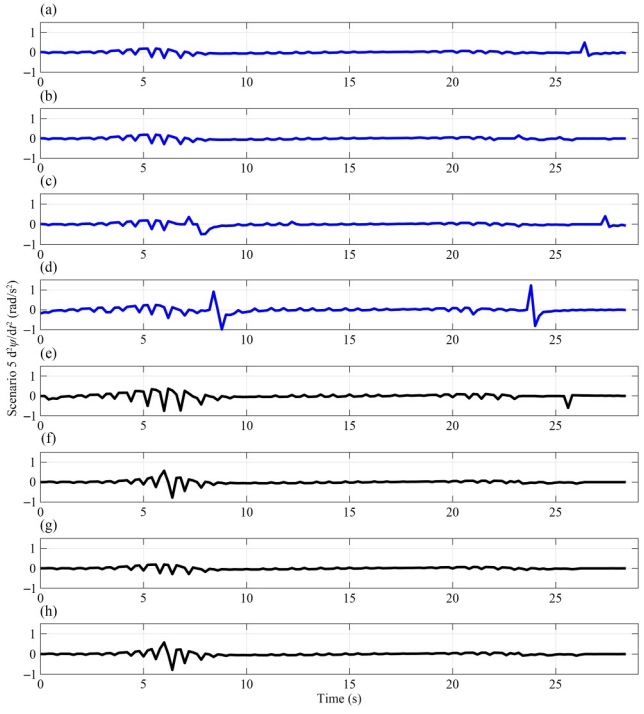

FIGURE 23 | $d^2\psi/dt^2$ over time for RL-assisted MPC cases and baseline controllers for scenario 5: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3, (d) RL-assisted MPC case 4; (e) MPC $\boldsymbol{W}_1$; (f) MPC $\boldsymbol{W}_2$; (g) MPC $\boldsymbol{W}_3$; (h) rule-based MPC.

In scenario 4 (Figure 22), case 3, with the best performer in $d_{y.\max}$, shows stronger oscillations in heading compared with case 2. However, case 2, which already offers competitive steering smoothness, produces a notably smoother heading profile with fewer abrupt spikes. This shows how different RL-assisted MPC cases can complement each other, balancing accuracy and stability.

Finally, in scenario 5 (Figure 23), case 4, which achieves the lowest $d_{y.\max}$ overall, also maintains heading stability with relatively smooth transitions. Although case 2 is competitive in steering effort, case 4's profile combines accurate positioning with stable heading, making it the most favourable trade-off in this manoeuvre. The RL-assisted MPC outperforms the rule-based method in terms of $d_{y.\max}$ and achieves lower total variation in both δ_{tv} and δ_{tvr}, indicating smoother and more stable control behaviour.

3.4 | Policy Adoption and Action Selection

The action–selection patterns of the RL agent are illustrated in Figures 24–28, where each action corresponds to one of the three baseline MPC weight sets ($\boldsymbol{W}_1$, $\boldsymbol{W}_2$, and $\boldsymbol{W}_3$). These temporal distributions provide insight into how the agent adapts its control strategy throughout the different phases of the parking manoeuvre.

In scenario 1 (Figure 24), case 3, which achieves both small $d_{y.\max}$ and stable δ, alternates primarily between baselines $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$ during alignment, while reducing reliance on baseline $\boldsymbol{W}_3$ until the last phase close to parking. This reflects a preference for weight sets that balance $d_{y.\max}$ control with smooth steering.

In scenario 2 (Figure 25), case 1 (the best performer in $d_{y.\max}$ and steering effort) shows a dominant preference for smoother weight sets during turning and alignment. The reduced switching frequency indicates that the agent converges to a stable control strategy, contributing to its improved heading stability.

In scenario 3 (Figure 26), case 2 that balances steering effort against slightly higher $d_{y.\max}$, exhibits more frequent switching between weight sets. This behaviour suggests that in more complex trajectories, the agent leverages dynamic adaptation to correct path errors, even if it introduces some variability in heading.

In scenario 4 (Figure 27), case 2 demonstrates consistent selection of $\boldsymbol{W}_3$ throughout most of the manoeuvre, contributing to smoother steering and heading stability. Case 3, while

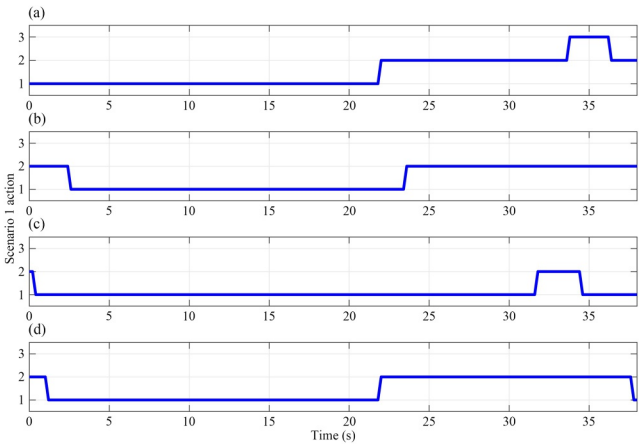

FIGURE 24 | Action selection over time for the RL-assisted MPC cases for scenario 1: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4.

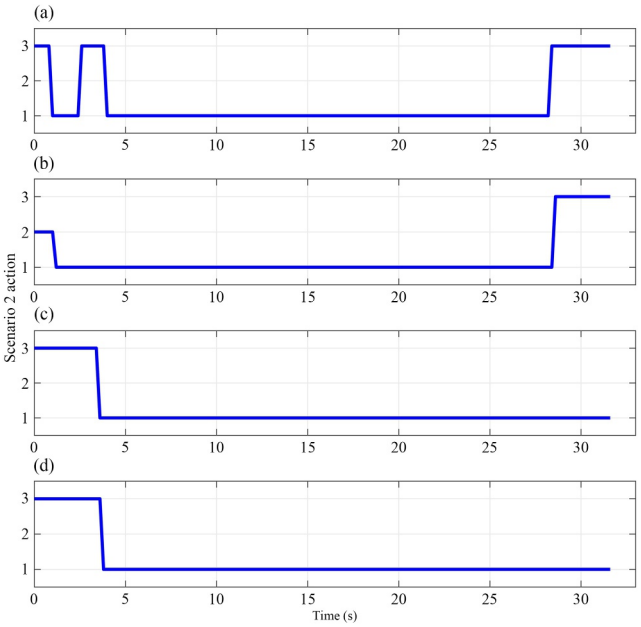

FIGURE 25 | Action selection over time for the RL-assisted MPC cases for scenario 2: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4.

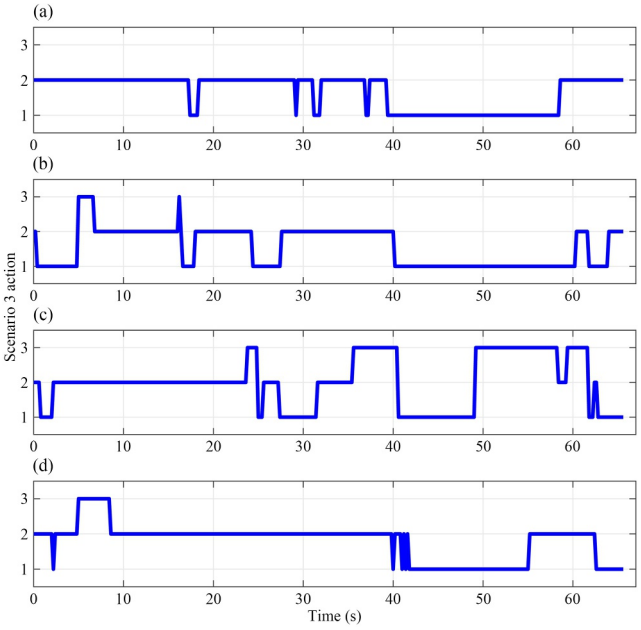

FIGURE 26 | Action selection over time for the RL-assisted MPC cases for scenario 3: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4.

achieving the lowest $d_{y.\max}$, involves more frequent shifts towards the end, highlighting the trade-off between tracking precision and control stability.

Finally, in scenario 5 (Figure 28), case 4, which produces the best lateral accuracy, displays a more dynamic switching pattern between weight sets, particularly in the final positioning phase. This reflects the agent's emphasis on minimising $d_{y.\max}$ at the expense of increased steering variability.

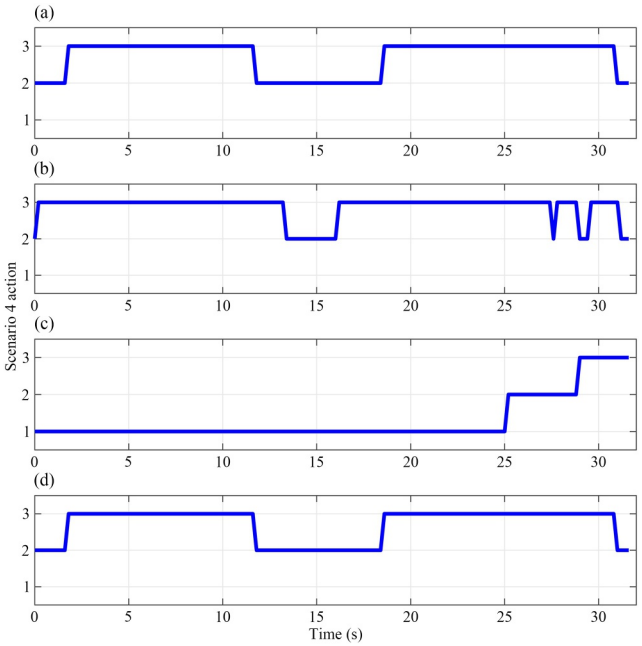

FIGURE 27 | Action selection over time for the RL-assisted MPC cases for scenario 4: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4.

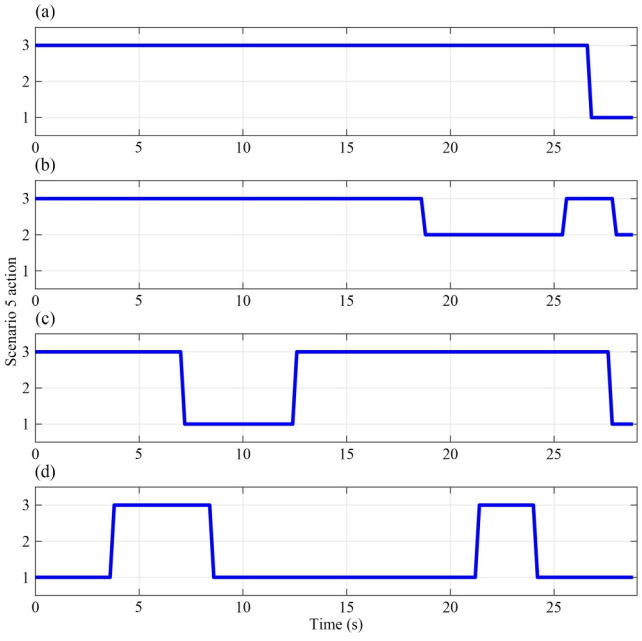

FIGURE 28 | Action selection over time for the RL-assisted MPC cases for scenario 5: (a) RL-assisted MPC case 1; (b) RL-assisted MPC case 2; (c) RL-assisted MPC case 3; (d) RL-assisted MPC case 4.

3.5 | Performance Evaluation Under Random Starting Position

In this section, we evaluate the generalisation capability of the proposed framework by deploying the vehicle from new starting positions that are not encountered during training. The objective is to assess whether the trained model can effectively control the vehicle and guide it to the designated parking spot under unseen initial conditions. Scenarios 4 and 5 are selected for this validation task, with each scenario tested using six distinct starting points to examine the robustness and adaptability of the RL-assisted MPC controller. The agent has never been exposed to any of these points during training. The results show that the RL-assisted MPC successfully controls the vehicle from all deployment points, converges back to the nearest reference path, and reaches the target parking spot, as shown in Figure 29. Similarly, in scenario 5, the validation results are also promising, showing that the agent effectively navigates the vehicle from six unseen random starting locations to the target position, as illustrated in Figure 30.

3.6 | Overall Assessment

Overall, in terms of $d_{y.\max}$, the RL-assisted MPC achieves better or close performance in all five scenarios. Even in scenario 3, where fixed-weight baselines are slightly more accurate in lateral tracking, the RL framework remains competitive and later demonstrated advantages in steering smoothness and heading stability. Across scenarios, the RL-assisted MPC produces smoother and more stable steering profiles, which emphasise heading stability in the RF. Importantly, when static baselines minimise steering variation, they often do so at the expense of d_y. By contrast, the RL-assisted MPC achieves smoother steering while preserving competitive accuracy, offering a superior balance between comfort and tracking. Across all five scenarios, the RL-assisted MPC consistently improves heading stability relative to fixed weight baselines. These results confirm that adaptive weight selection not only enhances lateral tracking and steering smoothness but also contributes significantly to stable and

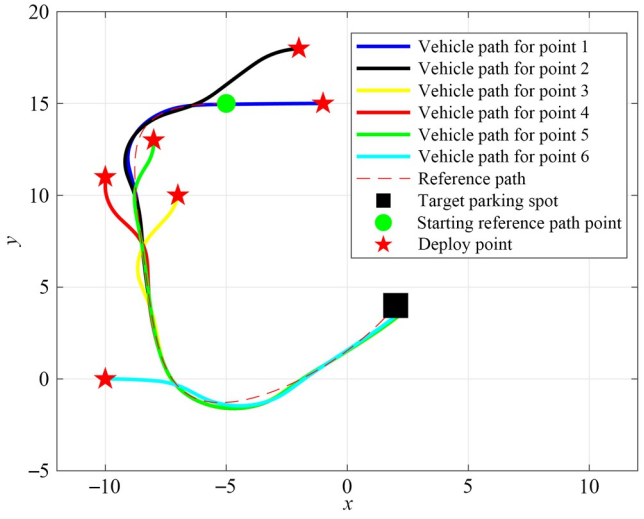

FIGURE 29 | Validation vehicle trajectories for scenario 4 from six unseen starting points. The red stars represent the random deployment points, and the black square represents the TPS point.

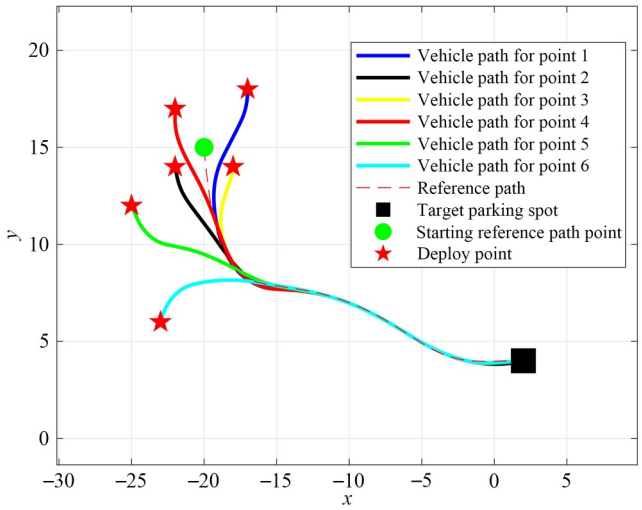

FIGURE 30 | Validation vehicle trajectories for scenario 5 from six unseen starting points. The red stars represent the random deployment points, and the black square represents the TPS point.

predictable heading dynamics. The action–selection results confirm that the RL agent adapts its weight choice according to the demands of each manoeuvre phase. Scenarios where smoother steering and heading stability are prioritised show more stable weight selection with limited switching, whereas cases prioritising lateral accuracy involve more frequent adaptation across weight sets. These findings demonstrate that reward design directly shapes policy behaviour, and that the agent can flexibly modulate its strategy to balance accuracy, comfort, and stability in real time. In addition to the fixed-weight MPC baselines, the proposed RL-assisted MPC is compared with a rule-based MPC controller in scenario 5. The results show that the RL-assisted MPC achieves improved lateral tracking accuracy and smoother steering behaviour, highlighting the benefit of adaptive weight selection over manually designed heuristic rules.

In summary, the proposed RL-assisted MPC framework improves the robustness and generalisation of automated parking controllers by balancing competing objectives without manual tuning.

4 | Conclusions

This paper introduces an RL-assisted MPC framework for APS, where an RL agent dynamically selects MPC weight sets in real time. By combining the predictive optimisation of MPC with the adaptive decision-making of RL, the proposed method addresses the limitations of static-weight controllers and improves adaptability across diverse parking manoeuvres. The experimental study across five scenarios demonstrates that the RL-assisted MPC can achieve reliable trajectory tracking while simultaneously reducing steering variability and enhancing heading stability. Rather than relying on a single fixed trade-off, the adaptive controller flexibly prioritises accuracy, comfort, or stability depending on the phase of the manoeuvre. Furthermore, the policy adaptation analysis provides interpretable insights into how reward design shapes control strategies, underscoring the transparency of the hybrid approach. In addition to the five predefined parking scenarios, the proposed RL-assisted MPC framework is evaluated under randomly selected and previously unseen initial vehicle positions, demonstrating robust performance and good generalisation across diverse deployment conditions. The proposed approach is also compared with a rule-based MPC controller in selected scenarios, demonstrating improved tracking and control smoothness. Future work will extend this approach to validate the framework in real-world experiments to demonstrate its practical feasibility.

Funding

This research was supported by the Faculty Startup Fund from Oakland University.

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. P. Pandita, Evaluation of Soft Actor Critic in Diverse Parking Environments. Master's thesis, (2022).

2. B. Thunyapoo, C. Ratchadakorntham, P. Siricharoen, and W. Susutti, "Self-Parking Car Simulation Using Reinforcement Learning Approach for Moderate Complexity Parking Scenario," in *2020 17th International Conference on Electrical engineering/electronics, Computer, Telecommunications and Information Technology (ECTI-CON)* (IEEE, 2020), 576–579, https://doi.org/10.1109/ECTI-CON49241.2020.9158298.

3. M. Albilani and A. Bouzeghoub, "Dynamic Adjustment of Reward Function for Proximal Policy Optimization With Imitation Learning: Application to Automated Parking Systems," in *2022 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2022), 1400–1408, https://doi.org/10.1109/IV51971.2022.9827194.

4. B. B. K. Ayawli, R. Chellali, A. Y. Appiah, and F. Kyeremeh, "An Overview of Nature-Inspired, Conventional, and Hybrid Methods of Autonomous Vehicle Path Planning," *Journal of Advanced Transportation* 2018, no. 1 (2018): 8269698, https://doi.org/10.1155/2018/8269698.

5. Z. Zhou, J. Chen, M. Tao, P. Zhang, and M. Xu, "Experimental Validation of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking," in *2023 IEEE International Conference on Electro Information Technology (eIT)* (2023), 35–40, https://doi.org/10.1109/eIT57321.2023.10187304.

6. Z. Zhou, C. Rother, and J. Chen, "Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking: Validation Using CARLA Simulator," *IEEE Transactions on Intelligent Vehicles* 8, no. 6 (2023): 3547–3555, https://doi.org/10.1109/tiv.2023.3266941.

7. J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design," in *2015 IEEE Intelligent Vehicles Symposium (IV)* (IEEE, 2015), 1094–1099, https://doi.org/10.1109/IVS.2015.7225830.

8. S. Di Cairano, H. E. Tseng, D. Bernardini, and A. Bemporad, "Vehicle Yaw Stability Control by Coordinated Active Front Steering and Differential Braking in the Tire Sideslip Angles Domain," *IEEE Transactions on Control Systems Technology* 21, no. 4 (2012): 1236–1248, https://doi.org/10.1109/TCST.2012.2198886.

9. R. Yu, H. Guo, Z. Sun, and H. Chen, "MPC-Based Regional Path Tracking Controller Design for Autonomous Ground Vehicles," in *2015 IEEE International Conference on Systems, Man, and Cybernetics* (IEEE, 2015), 2510–2515, https://doi.org/10.1109/SMC.2015.439.

10. J. Chen and Z. Yi, "Comparison of Event-Triggered Model Predictive Control for Autonomous Vehicle Path Tracking," in *2021 IEEE Conference on Control Technology and Applications (CCTA)* (IEEE, 2021), 808–813, https://doi.org/10.1109/CCTA48906.2021.9659192.

11. J. Chen, L. Zhang, and W. Gao, "Reconfigurable Model Predictive Control for Large Scale Distributed Systems," *IEEE Systems Journal* 18, no. 2 (2024): 965–976, https://doi.org/10.1109/jsyst.2024.3366911.

12. X. Sun, Y. Zhang, Y. Cai, and X. Tian, "Compensated Deadbeat Predictive Current Control Considering Disturbance and VSI Nonlinearity for in-Wheel PMSMs," *IEEE/ASME Transactions on Mechatronics* 27, no. 5 (2022): 3536–3547, https://doi.org/10.1109/TMECH.2021.3135936.

13. I. Jammeli, A. Chemori, H. Moon, S. Elloumi, and S. Mohammed, "An Assistive Explicit Model Predictive Control Framework for a Knee Rehabilitation Exoskeleton," *IEEE/ASME Transactions on Mechatronics* 27, no. 5 (2021): 3636–3647, https://doi.org/10.1109/TMECH.2021.3126674.

14. Q. Gong, J. Xu, J. Ye, H. Feng, and A. Shen, "Nonlinear Model Predictive Control for Premixed Turbocharged Natural Gas Engine," *IEEE/ASME Transactions on Mechatronics* 27, no. 5 (2021): 3694–3705, https://doi.org/10.1109/TMECH.2021.3130910.

15. C. Liu, C. Li, and W. Li, "Computationally Efficient MPC for Path Following of Underactuated Marine Vessels Using Projection Neural Network," *Neural Computing & Applications* 32, no. 11 (2020): 7455–7464, https://doi.org/10.1007/s00521-019-04273-y.

16. A. Jain, L. Chan, D. S. Brown, and A. D. Dragan, "Optimal Cost Design for Model Predictive Control," in *Learning for Dynamics and Control*, (2021), 1205–1217.

17. D. Silver, A. Huang, C. J. Maddison, et al., "Mastering the Game of Go With Deep Neural Networks and Tree Search," *nature* 529, no. 7587 (2016): 484–489, https://doi.org/10.1038/nature16961.

18. M. Ibrahim and R. Elhafiz, "Integrated Clinical Environment Security Analysis Using Reinforcement Learning," *Bioengineering* 9, no. 6 (2022): 253, https://doi.org/10.3390/bioengineering9060253.

19. M. Ibrahim and R. Elhafiz, "Security Analysis of Cyber-Physical Systems Using Reinforcement Learning," *Sensors* 23, no. 3 (2023): 1634, https://doi.org/10.3390/s23031634.

20. A. Irshayyid, J. Chen, and G. Xiong, "A Review on Reinforcement Learning-Based Highway Autonomous Vehicle Control," *Green Energy and Intelligent Transportation* 3, no. 4 (2024): 100156, https://doi.org/10.1016/j.geits.2024.100156.

21. J. Chen, X. Meng, and Z. Li, "Reinforcement Learning-Based Event-Triggered Model Predictive Control for Autonomous Vehicle Path Following," in *2022 American Control Conference (ACC)* (IEEE, 2022), 3342–3347, https://doi.org/10.23919/ACC53348.2022.9867347.

22. F. Dang, D. Chen, J. Chen, and Z. Li, "Event-Triggered Model Predictive Control With Deep Reinforcement Learning," *IEEE Transactions on Intelligent Vehicles* 9, no. 1 (2024): 459–468, https://doi.org/10.1109/tiv.2023.3329785.

23. H. Li, Q. Zhang, and D. Zhao, "Deep Reinforcement Learning-Based Automatic Exploration for Navigation in Unknown Environment," *IEEE Transactions on Neural Networks and Learning Systems* 31, no. 6 (2019): 2064–2076, https://doi.org/10.1109/tnnls.2019.2927869.

24. R. Kamalapurkar, L. Andrews, P. Walters, and W. E. Dixon, "Model-Based Reinforcement Learning for Infinite-Horizon Approximate Optimal Tracking," *IEEE Transactions on Neural Networks and Learning Systems* 28, no. 3 (2016): 753–758, https://doi.org/10.1109/tnnls.2015.2511658.

25. Y. Li, "Reinforcement Learning Applications," *arXiv preprint arXiv:1908.06973* (2019), https://arxiv.org/abs/1908.06973.

26. X. Liu, S. Zhu, Y. Fang, et al., "Optimization Design of Parking Models Based on Complex and Random Parking Environments," *World Electric Vehicle Journal* 14, no. 12 (2023): 344, https://doi.org/10.3390/wevj14120344.

27. R. Dong, M. Hu, T. Cui, D. Wan, and X. Meng, A Mathematical Modeling Approach for Optimal Parking Space Selection and Path Planning in Autonomous Parking Systems with UAV-Assisted Topsis Entropy Weight Method (2023), https://doi.org/10.21203/rs.3.rs-3639234/v1.

28. F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the Limitations of Behavior Cloning for Autonomous Driving," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (IEEE, 2019), 9328–9337, https://doi.org/10.1109/ICCV.2019.00942.

29. M. Gallieri, S. S. M. Salehian, N. E. Toklu, et al., "Safe Interactive Model-Based Learning," *arXiv preprint arXiv:1911.06556* (2019), https://arxiv.org/abs/1911.06556.

30. F. Berkenkamp, M. Turchetta, A. Schoellig, and A. Krause, "Safe Model-Based Reinforcement Learning With Stability Guarantees," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems* (ACM, 2017), 908–919, https://doi.org/10.48550/arXiv.1705.08551.

31. X. Tang, Y. Yang, T. Liu, X. Lin, K. Yang, and S. Li, "Path Planning and Tracking Control for Parking via Soft Actor-Critic Under Non-Ideal Scenarios," *IEEE/CAA Journal of Automatica Sinica* 11, no. 1 (2023): 181–195, https://doi.org/10.1109/jas.2023.123975.

32. P. Zhang, L. Xiong, Z. Yu, et al., "Reinforcement Learning-Based End-to-End Parking for Automatic Parking System," *Sensors* 19, no. 18 (2019): 3996, https://doi.org/10.3390/s19183996.

33. M. Lin, Z. Sun, Y. Xia, and J. Zhang, "Reinforcement Learning-Based Model Predictive Control for Discrete-Time Systems," *IEEE Transactions on Neural Networks and Learning Systems* 35, no. 3 (2023): 3312–3324, https://doi.org/10.1109/tnnls.2023.3273590.

34. K. Lowrey, A. Rajeswaran, S. Kakade, E. Todorov, and I. Mordatch, "Plan Online, Learn Offline: Efficient Learning and Exploration via Model-Based Control," *arXiv preprint arXiv:1811.01848* (2018), https://arxiv.org/abs/1811.01848.

35. F. Farshidian, D. Hoeller, and M. Hutter, "Deep Value Model Predictive Control," *arXiv preprint arXiv:1910.03358* (2019), https://arxiv.org/abs/1910.03358.

36. N. Karnchanachari, M. I. Valls, D. Hoeller, and M. Hutter, "Practical Reinforcement Learning for MPC: Learning From Sparse Objectives in Under an Hour on a Real Robot," in *Learning for Dynamics and Control* (2020), 211–224.

37. S. Gros and M. Zanon, "Data-Driven Economic NMPC Using Reinforcement Learning," *IEEE Transactions on Automatic Control* 65, no. 2 (2019): 636–648, https://doi.org/10.1109/tac.2019.2913768.

38. M. Zanon, S. Gros, and A. Bemporad, "Practical Reinforcement Learning of Stabilizing Economic MPC," in *2019 18th European Control Conference (ECC)* (IEEE, 2019), 2258–2263, https://doi.org/10.23919/ECC.2019.8795816.

39. M. Zanon and S. Gros, "Safe Reinforcement Learning Using Robust MPC," *IEEE Transactions on Automatic Control* 66, no. 8 (2020): 3638–3652, https://doi.org/10.1109/tac.2020.3024161.

40. H. Alawsi, Z. Zhou, A. Irshayyid, and J. Chen, "RL-Assisted Model Predictive Control for Automated Parking Systems," in *2025 IEEE*

International Conference on Unmanned Systems (ICUS) (IEEE, 2025), https://doi.org/10.1109/ICUS66297.2025.11294004.

41. A. Irshayyid and J. Chen, "Comparative Study of Cooperative Platoon Merging Control Based on Reinforcement Learning," *Sensors* 23, no. 2 (2023): 990, https://doi.org/10.3390/s23020990.

42. L. Zheng, P. Zeng, W. Yang, Y. Li, and Z. Zhan, "Bézier Curve-Based Trajectory Planning for Autonomous Vehicles With Collision Avoidance," *IET Intelligent Transport Systems* 14, no. 13 (2020): 1882–1891, https://doi.org/10.1049/iet-its.2020.0355.

43. L. Chen, Z. Qin, M. Hu, et al., "Nonlinear Model Predictive Lateral Control for Automated Parking System With Position and Attitude Coordination Control," in *2022 37th Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (IEEE, 2022), 49–55, https://doi.org/10.1109/YAC57282.2022.10023808.