2025 IEEE International Conference on Unmanned Systems (ICUS) | 979-8-3315-2672-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICUS66297.2025.11294004

RL-assisted Model Predictive Control for Automated Parking Systems

Hussein Alawsi
ECE Department
Oakland University
Rochester, MI 48309, USA
husseinalawsi@oakland.edu

Zhaodong Zhou
ECE Department
Oakland University
Rochester, MI 48309, USA
zhaodongzhou@oakland.edu

Ali Irshayyid
ECE Department
Oakland University
Rochester, MI 48309, USA
aliirshayyid@oakland.edu

Jun Chen*
ECE Department
Oakland University
Rochester, MI 48309, USA
junchen@oakland.edu

***Abstract*—This paper proposes a reinforcement learning-assisted model predictive control (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 framework leverages the predictive optimization capabilities of MPC and the adaptive decision-making strength of RL. Experimental evaluations demonstrate that the proposed RL-assisted MPC framework achieves comparable lateral tracking accuracy, while consistently providing smoother steering behavior and improved heading stability compared to baseline controllers using static MPC weights. The results highlight the potential of integrating RL with model-based control for enhancing robustness and adaptability in automated parking systems.**

***Index Terms*—autonomous parking, reinforcement learning, nonlinear MPC, deep Q-Network, adaptive control**

I. Introduction

The rising number of vehicles on the roads each year has significantly increased the demand for parking spaces, making parking a time-consuming and energy-draining task, especially in densely populated urban areas [1]. Additionally, this surge in vehicles also escalates the probability of accidents, resulting in substantial losses and damages [2]. Advanced Driver Assistance Systems (ADAS) have been developed to address these challenges, incorporating various features, including autonomous parking systems (APS), to enhance safety and convenience [3].

The architecture of APS is generally comprised of three key components: parking spot detection, path planning, and path following [3]. Path planning and path following represent two fundamental components in the behavioral control of autonomous vehicles (AVs) [4]. Path planning is performed to generate a feasible trajectory while satisfying safety constraints, whereas a controller is subsequently employed to accurately track this path by accounting for the vehicle's current state and applying appropriate control inputs. Extensive research has been devoted to the development of effective path planning strategies [5]. However, path following remains a challenging task due to the highly dynamic nature of AVs and the limitations in onboard computation and communication resources. Path-following controllers are required to generate accurate control commands in real-time while operating under these computational and communication constraints. Various control strategies have been employed for path following, including proportional-integral-derivative (PID) controllers, state feedback controllers, and model predictive control (MPC) [6]. MPC has gained particular attention and has been widely studied in the field of AVs [7]–[11].

Although MPC provides a systematic framework for handling constraints and predicting future behavior, its performance is heavily dependent on the proper selection of cost function weights [12]. These weights determine the relative importance of different objectives, such as minimizing lateral error, steering effort, and steering rate. Selecting a single static set of weights that performs well in all scenarios is challenging, as different driving conditions may require different trade-offs. This motivates the need for a more adaptive approach, where the controller can adjust its weighting strategy based on the current AV state.

In recent years, RL has achieved remarkable success in complex decision-making tasks [13]–[18]. In RL, the agent relies solely on observed state transitions and corresponding rewards to update its policy, without requiring prior knowledge of the system dynamics [19]. When considering the task of APS, it is challenging to mathematically define the reward function for an autonomous system [20], [21]. Furthermore, RL training can be extremely time-consuming, often requiring days of interaction to converge. Nevertheless, to approximate the global optimal policy, an RL agent typically explores different policies and learns through trial and error, which poses challenges in providing formal guarantees on the safety of the resulting behaviors [22]. This issue becomes particularly critical in safety-sensitive applications, where safety is often characterized by the system's stability.

Given the powerful data-driven optimization 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. This paper proposes an RL-assisted MPC framework for APS. This hybrid approach combines the adaptability of RL with the predictive optimization 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

*Corresponding Author

979-8-3315-2672-6/25/$31.00 ©2025 IEEE

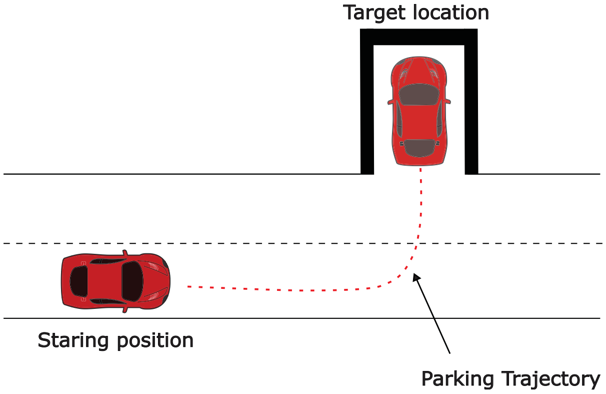

Fig. 1. Reference trajectory used for the parking maneuver, showing the AV's planned path from the starting position to the designated parking spot.

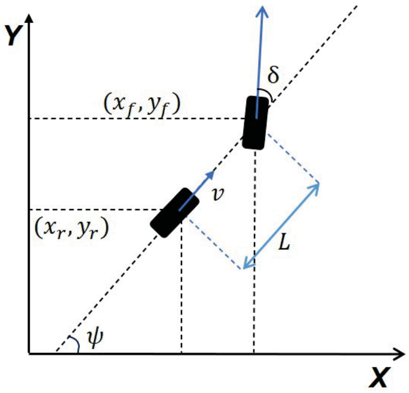

Fig. 2. Kinematic bicycle model of the vehicle referenced at the rear axle center.

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 optimization ability of MPC and the adaptive decision-making capability of RL, the resulting approach improves path-following behavior for APS without requiring extensive retraining or manual weight tuning, and avoids reliance on a single static set of weights.

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

II. Methodology

The objective is to allow an AV to follow a predefined path during parking (Fig. 1) with minimal lateral deviation, while ensuring smooth steering and stable heading transitions, all within the limits of safety constraints. The AV operates in a structured environment modeled using a kinematic bicycle model (Fig. 2). The reference trajectory is generated using a Bézier curve.

A. *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 [23]. In the context of APS, Bézier curves are particularly attractive because they allow the vehicle to generate collision-free, smooth trajectories with minimal computational cost [24]. 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 [24].

In this work, Bézier curve are employed to construct the reference parking path by specifying the initial vehicle position (p_0), the parking target position (p_3), and intermediate way-points that define the maneuvering profile (p_1 and p_2), defined as follows.

$$p(t) = (1-t)^3 P_0 + 3(1-t)^2 t P_1 + 3(1-t) t^2 P_2 + t^3 P_3. \quad (1)$$

B. *Vehicle Dynamics*

In APS, due to the low-speed nature of the maneuvers, the kinematic bicycle model provides a convenient and accurate approximation of the vehicle dynamics [25] shown in Fig. 2.

The rear axle center velocity is given by:

$$v = x_r \cos\psi + y_r \sin\psi. \quad (2)$$

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

$$\begin{aligned} \dot{x}_f \sin(\psi+\delta) - \dot{y}_f \cos(\psi+\delta) &= 0 \\ x_r \sin(\psi) - y_r \cos(\psi) &= 0. \end{aligned} \quad (3)$$

Combining (2) with (3) yields:

$$\begin{aligned} \dot{x}_r &= v\cos(\psi) \\ \dot{y}_r &= v\sin(\psi). \end{aligned} \quad (4)$$

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

$$\begin{aligned} x_f &= x_r + L\cos(\psi) \\ y_f &= y_r + L\sin(\psi). \end{aligned} \quad (5)$$

The steering angle can be represented as:

$$\delta = \arctan\frac{L\dot{\psi}}{v}. \quad (6)$$

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

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos\psi \\ \sin\psi \\ \frac{\tan\delta}{L} \end{bmatrix} v, \quad (7)$$

where v is assumed to be constant.

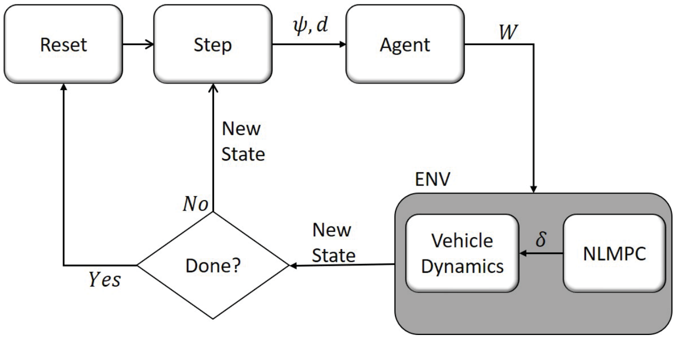

Fig. 3. Flowchart of RL-assisted MPC. d is the distance from the current AV position to the target point. W represents a single set of weights.

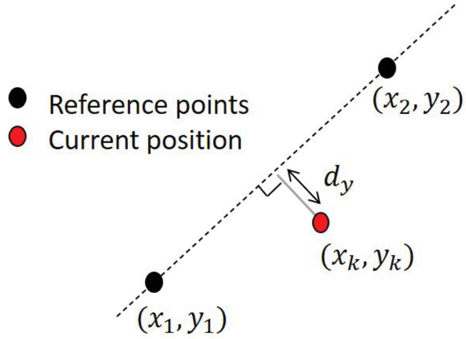

Fig. 4. Illustration of lateral deviation d_y from the reference path, computed as the perpendicular distance from the vehicle's current position to the segment between two consecutive reference points.

C. 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 optimization 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 behavior in real-time based on the current AV states x_r, y_r, and ψ.

The optimization process is subject to physical limitations of the system—specifically, constraints on the steering angle. By solving this constrained optimization 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 utilized. At each time step, the agent observes the current observation, which consists of the vehicle heading ψ and the distance d between the vehicle's current position and the center of the parking spot. Based on this state, the agent selects a set of weights W from three baselines $W \in \{w_1, w_2, w_3\}$ 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 the MPC controller. The process repeats until the parking task is either successfully completed or terminated as shown in Fig. 3.

The MPC problem is formulated as a constrained optimization problem that is solved at each control step. Specifically, the objective is to minimize a cost function J over a finite prediction horizon P, (8a) subject to vehicle dynamics and steering constraints (8c). The optimization problem consists of three weighted terms, each serving a specific purpose in shaping the vehicle's behavior. The first term penalizes the deviation from the reference path by minimizing the squared Euclidean distance between the predicted vehicle position and the corresponding reference point at each step. The second term penalizes large control inputs by minimizing the squared value of δ, encouraging smoother and more stable maneuvers. The third term penalizes abrupt changes in control input by minimizing the difference between successive steering angles, thereby promoting steering smoothness. The optimization problem computed at each control step is defined as follows [6].

$$\min_{\delta} \quad J = \sum_{k=1}^{p} \lambda_t \cdot \|X_k - X_{ref,k}\|^2 + \sum_{k=0}^{p-1} \lambda_s \cdot \delta_k^2 + \sum_{k=0}^{p-1} \lambda_d \cdot (\delta_k - \delta_{k-1})^2 \tag{8a}$$

$$\text{s.t.} \quad \text{Vehicle model (7)} \tag{8b}$$

$$\frac{-\pi}{4} \leq \delta \leq \frac{\pi}{4}, \tag{8c}$$

where $X_k = [x_k, y_k]^T$ denote the vehicle position at time step k, and let $X_{\text{ref},k} = [x_{\text{ref},k}, y_{\text{ref},k}]^T$ 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, lateral deviation 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 penalized within the cost function to ensure precise trajectory tracking. The overall structure is illustrated in Fig. 4, and the mathematical formulation is as follows [6]:

$$Y = \frac{|(x_2 - x_1)(y_1 - p_y) - (x_1 - x_k)(y_2 - y_k)|}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}. \tag{9}$$

The RL reward function R at the time step t, is defined as:

$$R_t = \begin{cases} -\left|d_y\right|, & \text{if } t \text{ is terminal} \\ -\left(\alpha_l \cdot |d_y| + \alpha_h \cdot \left|\ddot{\psi}\right|\right), & \text{otherwise,} \end{cases} \tag{10}$$

Algorithm 1: RL-Assisted MPC

```
Initialize $\phi$, $\phi^-$, buffer $\mathcal{D}$, and AV states;
for all episodes $< M$ do
    observe AV state $S = \psi, d, x, y$;
    for $t = 1, \cdots, T$ do
        Following $\epsilon$, select weights
          $w_t = argmax_a Q_\phi(\psi, d, a)$;
        Apply $w_t$ to MPC cost function;
        Solve MPC and compute steering command $\delta_t$;
        Step in the environment using $\delta_t$ and observe
          $S_{t+1}$, and $r_t$;
        Store tuple $([\psi_t, d_t], w_t, r_t, [\psi_{t+1}, d_{t+1}])$ in $\mathcal{D}$;
        Sample a random minibatch of
          $([\psi_i, d_i], w_i, r_i, [\psi_{i+1}, d_{i+1}])$ from $\mathcal{D}$;
        set $y_i =$
          $\begin{cases} r_i, \text{if } i+1 \text{ terminates}, \\ r_i + \gamma \max_{w'} Q_{\phi^-}\left([\psi_{i+1}, d_{i+1}], w'\right), & \text{otherwise.} \end{cases}$;
        Update $\phi$ to minimize $(y_i - Q_\phi([\psi_i, d_i]))^2$;
        Every $C$ steps, $\phi^- \leftarrow \phi$;
    end
end
```

where

α_l : lateral error reward weight

α_h : heading error reward weight

$\ddot{\psi}$: yaw rate acceleration (second derivative of heading).

The full control logic of the RL-assisted MPC is outlined in Algorithm 1.

TABLE I
REWARD WEIGHTS USED IN RL-ASSISTED MPC TRAINING CASES.

RL-assisted MPC	α_l	α_h
Case 1	0.25	0.75
Case 2	0.35	0.65
Case 3	0.40	0.60
Case 4	0.45	0.55

TABLE II
WEIGHT SETS USED IN MPC COST FUNCTION FOR THE THREE FIXED-WEIGHT BASELINE CONTROLLERS.

Weights	λ_t	λ_s	λ_d
$w1$	6	2	0
$w2$	2	2	8
$w3$	1	0	5

TABLE III
PERFORMANCE METRICS ACROSS RL-ASSISTED MPC AND FIXED-WEIGHT MPC CASES.

Controller	$d_{y.max}$	TV-$\Delta\delta$	TV-δ
RL-assisted MPC Case 1	1.0802	6.2146	2.1990
RL-assisted MPC Case 2	1.1187	8.3392	2.9875
RL-assisted MPC Case 3	1.1137	12.9124	3.6070
RL-assisted MPC Case 4	0.2208	11.0745	2.5419
MPC w1	0.1923	11.3171	2.8623
MPC w2	1.0923	7.5360	2.2434
MPC w3	2.5884	5.9796	1.6727

III. RESULTS

This section presents the evaluation of the proposed RL-assisted MPC under four distinct reward weight configurations (Cases 1–4), as defined in Table I. The controller's performance is compared against three baseline MPC configurations using fixed cost function weights (w_1, w_2, and w_3), detailed in Table II. All experiments were conducted under identical initial conditions and reference path 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 $\alpha_l + \alpha_h = 1$. The RL agent was trained using the reward function defined in (10).

Key evaluation metrics include maximum lateral deviation over time ($d_{y.max}$), steering rate ($\Delta\delta$, Fig. 5), total variation in steering input (TV-δ), lateral deviation (d_y, Fig. 6), heading acceleration ($d^2\psi$, Fig. 7), and the total variation in steering rate (TV-$\Delta\delta$), providing insights into tracking accuracy, control smoothness, and overall stability.

As shown in Fig. 5, the RL-assisted MPC produces noticeably smoother steering profiles, particularly in Case 1 and Case 2, which exhibit reduced oscillation magnitudes and fewer abrupt changes. This is quantitatively supported by TV-δ reported in Table III, where Case 1 achieves 2.1990 and Case 2 yields 2.9875, both lower than W1 (2.8623) and W2 (2.2434). Although W3 attains the lowest TV-δ (1.6727), this comes at the cost of significantly worse lateral tracking performance, with $d_{y.max}$ of 2.5884 meters more than ten times higher than Case 4 (0.2208), as seen in Fig. 6. In terms of lateral accuracy, Cases 1–3 offer comparable or slightly higher tracking errors than the best baseline (W1: 0.1923 m), but Case 4 demonstrates a clear improvement with $d_{y.max}$ of only 0.2208 m. Furthermore, heading smoothness illustrated in Fig. 7—is best in Case 2, which combines moderate α_l and α_h. This case exhibits visibly reduced heading jerk and minimized spikes, confirming that balanced reward weighting improves maneuver stability. Overall the RL-assisted MPC provides a mechanism to adaptively switch weight sets based on the vehicle's state, which can lead to improved overall maneuver stability and comparable tracking performance, yet with occasional increases in steering rate when aggressive adjustments are beneficial for trajectory correction to minimize lateral error.

Fig. 8 shows the action selection patterns of the RL agent for each of the four training cases, where each action corresponds to one of the three predefined MPC weight sets. The temporal distribution of actions reflects how the agent adapts its control strategy in response to the vehicle's state during different phases of the parking maneuver. In Case 1 and Case 2, which place greater emphasis on heading stability, the agent shows a tendency to remain with smoother-weight configu-

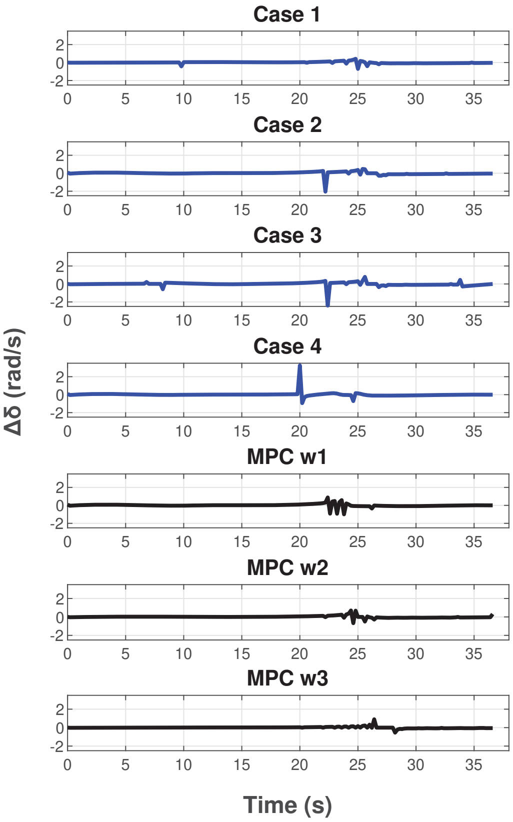

Fig. 5. Steering rate $\Delta\delta$ over time for RL-assisted MPC Cases and baseline controllers.

rations, particularly during turning and alignment segments. In contrast, Case 4, which prioritizes lateral tracking, exhibits more frequent switching between weight sets, suggesting a more reactive control strategy to minimize deviation. These variations highlight the agent's capacity to modulate control behavior in real-time and emphasize the role of reward design in shaping policy adaptation.

Finally, the proposed RL-assisted MPC framework demonstrates notable advantages over traditional static-weight MPC configurations. Across all reward settings, the RL-assisted controller achieved consistently lower steering input variability (TV-δ) and smoother heading transitions. While lateral tracking accuracy varied across cases, Case 4 achieved the lowest maximum lateral deviation ($d_{y.max}$) among all RL configurations and outperformed two of the three baseline controllers, while remaining very close to the best-performing baseline. The ability to adapt cost weights dynamically based on the vehicle's state enabled the RL agent to respond effectively across different phases of the maneuver including entry, alignment, and final positioning. These results underscore the potential of integrating model predictive control with reinforcement learning for real-time, adaptive control in APS.

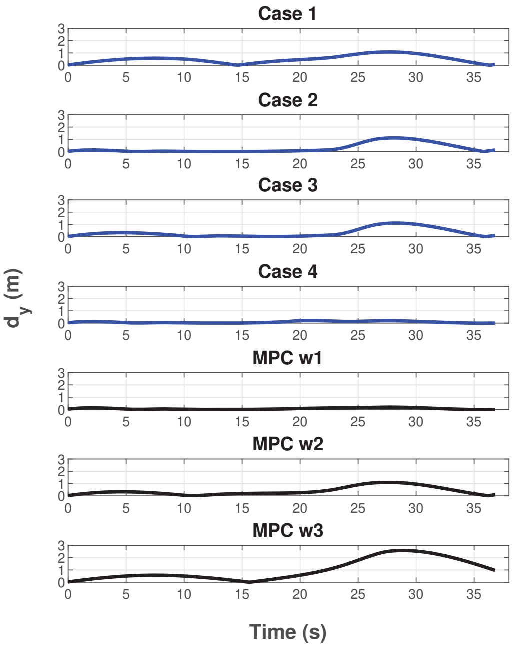

Fig. 6. Lateral deviation d_y over time for RL-assisted MPC Cases and baseline controllers.

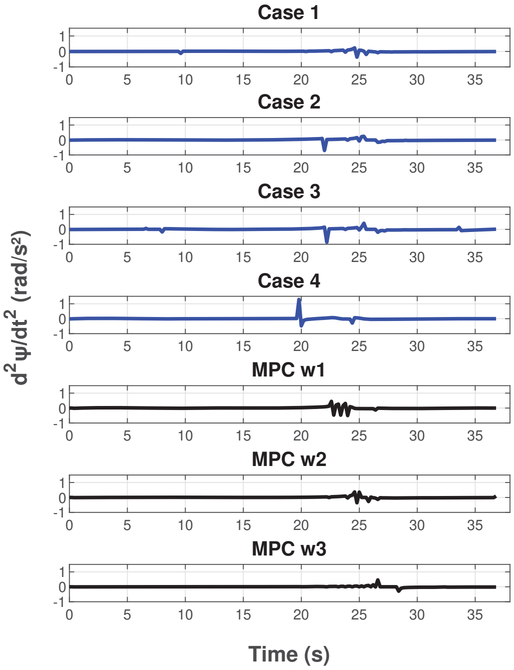

Fig. 7. Heading acceleration $\Delta^2\psi$ over time for RL-assisted MPC Cases and baseline controllers.

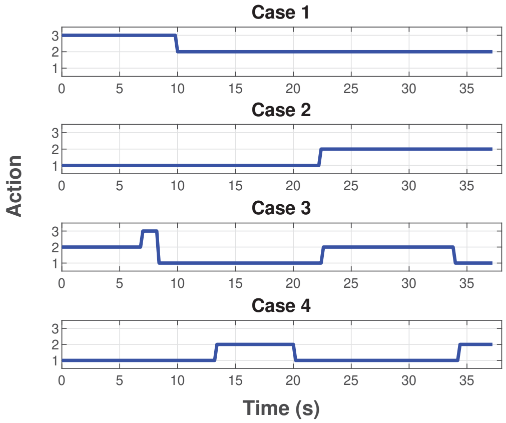

Fig. 8. Action selection over time for the RL-assisted MPC Cases.

IV. Conclusions

This paper presented an RL-assisted MPC framework that dynamically adapts cost function weights to improve trajectory tracking and control smoothness in autonomous parking systems. By leveraging the predictive capabilities of MPC and the adaptive decision-making of reinforcement learning (RL), the proposed approach enables real-time adjustment of control behavior without manual weight tuning or offline re-optimization. Simulation results demonstrate that the RL-assisted MPC controller consistently outperforms static-weight MPC configurations in terms of steering smoothness and heading stability. While lateral tracking performance varied across reward configurations, the best-performing RL-assisted MPC Case achieved lower maximum lateral deviation than two of the three baseline controllers and remained very close to the best-performing baseline. These findings highlight the potential of hybrid learning-based control for enhancing adaptability, generalization, and maneuver stability in structured parking environments. Future work will focus on expanding the experimental evaluation to include diverse parking scenarios, such as varying initial vehicle positions, target locations, and obstacle configurations. This extension will allow a more comprehensive assessment of the robustness and generalizability of the proposed RL-assisted MPC framework in real-world applications.

References

[1] P. Pandita, "Evaluation of soft actor critic in diverse parking environments," Ph.D. dissertation, 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, pp. 576–579.

[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, pp. 1400–1408.

[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*, vol. 2018, no. 1, p. 8269698, 2018.

[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)*. IEEE, 2023, pp. 35–40.

[6] Z. Zhou, C. Rother, and J. Chen, "Event-triggered model predictive control for autonomous vehicle path tracking: Validation using CARLA simulator," *IEEE Tran. Intel. Veh.*, vol. 8, no. 6, pp. 3547–3555, 2023.

[7] J. Kong, M. Pfeiffer, G. Schildbach, and F. Borrelli, "Kinematic and dynamic vehicle models for autonomous driving control design," in *IEEE Intell. Veh. Symp.*, Seoul, Korea, June 28–July 1, 2015, pp. 1094–1099.

[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 Trans. Control Syst. Tech.*, vol. 21, no. 4, pp. 1236–1248, 2012.

[9] R. Yu, H. Guo, Z. Sun, and H. Chen, "MPC-based regional path tracking controller design for autonomous ground vehicles," in *IEEE International Conference on Systems, Man, and Cybernetics*, Hong Kong, China, October 09-12, 2015, pp. 2510–2515.

[10] J. Chen and Z. Yi, "Comparison of event-triggered model predictive control for autonomous vehicle path tracking," in *IEEE Conf. Control Technology and Applications*, San Diego, CA, August 8–11, 2021.

[11] J. Chen, L. Zhang, and W. Gao, "Reconfigurable model predictive control for large scale distributed systems," *IEEE Systems Journal*, vol. 18, no. 2, pp. 965–976, June 2024.

[12] A. Jain, L. Chan, D. S. Brown, and A. D. Dragan, "Optimal cost design for model predictive control," in *Learning for Dynamics and Control*. PMLR, 2021, pp. 1205–1217.

[13] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of go with deep neural networks and tree search," *nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[14] M. Ibrahim and R. Elhafiz, "Integrated clinical environment security analysis using reinforcement learning," *Bioengineering*, vol. 9, no. 6, p. 253, 2022.

[15] ——, "Security analysis of cyber-physical systems using reinforcement learning," *Sensors*, vol. 23, no. 3, p. 1634, 2023.

[16] A. Irshayyid, J. Chen, and G. Xiong, "A review on reinforcement learning-based highway autonomous vehicle control," *Green Energy and Intelligent Transportation*, vol. 3, no. 4, p. 100156, 2024.

[17] J. Chen, X. Meng, and Z. Li, "Reinforcement learning-based event-triggered model predictive control for autonomous vehicle path following," in *American Control Conf.*, Atlanta, GA, June 8–10, 2022.

[18] F. Dang, D. Chen, J. Chen, and Z. Li, "Event-triggered model predictive control with deep reinforcement learning," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 459–468, January 2024.

[19] Y. Li, "Reinforcement learning applications," *arXiv preprint arXiv:1908.06973*, 2019.

[20] X. Liu, S. Zhu, Y. Fang, Y. Wang, L. Fu, W. Lei, and Z. Zhou, "Optimization design of parking models based on complex and random parking environments," *World Electric Vehicle Journal*, vol. 14, no. 12, p. 344, 2023.

[21] 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.

[22] M. Gallieri, S. S. M. Salehian, N. E. Toklu, A. Quaglino, J. Masci, J. Koutník, and F. Gomez, "Safe interactive model-based learning," *arXiv preprint arXiv:1911.06556*, 2019.

[23] A. Irshayyid and J. Chen, "Comparative study of cooperative platoon merging control based on reinforcement learning," *Sensors*, vol. 23, no. 2, p. 990, 2023.

[24] 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*, vol. 14, no. 13, pp. 1882–1891, 2020.

[25] L. Chen, Z. Qin, M. Hu, H. Gao, F. Zhang, Y. Bian, B. Xu, X. Peng, and J. Hu, "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, pp. 49–55.