Hierarchical Motion Planning and Tracking for Autonomous Vehicles Using Global Heuristic Based Potential Field and Reinforcement Learning Based Predictive Control

Du, Guodong; Zou, Yuan; Zhang, Xudong; Li, Zirui; Liu, Qi

DOI
10.1109/TITS.2023.3266195

Publication date
2023

Document Version
Final published version

Published in
IEEE Transactions on Intelligent Transportation Systems

Hierarchical Motion Planning and Tracking for Autonomous Vehicles Using Global Heuristic Based Potential Field and Reinforcement Learning Based Predictive Control

Guodong Du, Yuan Zou, *Senior Member, IEEE*, Xudong Zhang, *Member, IEEE*, Zirui Li, and Qi Liu, *Student Member, IEEE*

***Abstract*— The autonomous vehicle is widely applied in various ground operations, in which motion planning and tracking control are becoming the key technologies to achieve autonomous driving. In order to further improve the performance of motion planning and tracking control, an efficient hierarchical framework containing motion planning and tracking control for the autonomous vehicles is constructed in this paper. Firstly, the problems of planning and control are modeled and formulated for the autonomous vehicle. Then, the logical structure of the hierarchical framework is described in detail, which contains several algorithmic improvements and logical associations. The global heuristic planning based artificial potential field method is developed to generate the real-time optimal motion sequence, and the prioritized Q-learning based forward predictive control method is proposed to further optimize the effectiveness of tracking control. The hierarchical framework is evaluated and validated by the numerical simulation, virtual driving environment simulation and real-world scenario. The results show that both the motion planning layer and the tracking control layer of the hierarchical framework perform better than other previous methods. Finally, the adaptability of the proposed framework is verified by applying another driving scenario. Furthermore, the hierarchical framework also has the ability for the real-time application.**

***Index Terms*— Autonomous vehicle, motion planning, tracking control, global heuristic based potential field, reinforcement learning based predictive control.**

Manuscript received 18 July 2022; revised 26 January 2023; accepted 28 March 2023. This work was supported in part by the National Key Research and Development Program of China under Grant 2021YFB2500900 and in part by the Young Elite Scientists Sponsorship Program by China Association for Science and Technology (CAST). The Associate Editor for this article was J. W. Choi. *(Corresponding author: Yuan Zou.)*

Guodong Du is with the Institute of Dynamic System and Control, ETH Zurich, 8092 Zürich, Switzerland, and also with the National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, and the Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing Institute of Technology, Beijing 100081, China (e-mail: guodongdu_robbie@163.com).

Yuan Zou, Xudong Zhang, and Qi Liu are with the National Engineering Laboratory for Electric Vehicles, School of Mechanical Engineering, and the Collaborative Innovation Center of Electric Vehicles in Beijing, Beijing Institute of Technology, Beijing 100081, China (e-mail: zouyuan@bit.edu.cn; xudong.zhang@bit.edu.cn; 3120195257@bit.edu.cn).

Zirui Li is with the Department of Transport and Planning, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China (e-mail: 3120195255@bit.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3266195

I. Introduction

THE application of the intelligent transportation system (ITS) is promoting the development of transportation industry [1]. As one of the most promising advancements in the intelligent transportation system, the autonomous vehicle is widely applied in various ground operations, which improves the driving efficiency and reduces the burden of drivers [2]. The general technical framework of autonomous vehicles mainly contains the acquisition layer, perception layer, communication layer, decision layer, control layer and actuation layer [3].

In the operation of autonomous vehicle system, decision layer and control layer determine the driving performance [4]. The decision layer is commonly constructed by the multi-stage planning decision. Firstly, the global path from the starting point to the target point is generated by using a specific planning algorithm according to the driving task requirements and the obtained external environment information. In view of dynamic driving environment, the local motion needs to be updated if necessary. Then, the autonomous vehicle uses the path and motion planned by the decision layer for its self-driving. The control layer commonly consists of the longitudinal control module and the lateral control module [5]. The autonomous vehicle can track the planned path by executing the generated control strategy and meet various driving requirements. In recent years, the motion planning and tracking for autonomous vehicle have become the extensive and long-term research area using a wide variety of methods [6].

A. Literature Review

In the research of motion planning, graph search algorithms are representative. The Dijkstra algorithm was applied to find the shortest path from start point to target point, where the configuration space was approximated as the discrete grid space [7]. In [8], another extended Dijkstra algorithm was used to generate the optimal path through the equivalent inverse transformation. In order to achieve faster node search speed, a series of variations of Dijkstra algorithm were proposed. A-Star (A*) algorithm introduces the concept of heuristic function into the graph search, and defines different weights

of path nodes [9]. An improved A* based motion planning algorithm for autonomous land vehicles was proposed, and was proved to be more efficient than the Dijkstra algorithm [10]. In addition, the D-Star (D*) is another typical graph search algorithm, where the heuristic improvement contains the dynamic cost graph search [3]. The traditional graph search methods are suitable for solving the global path in static environment. However, it can hardly deal with the real-time changes of the local environment, and the path solved is not continuous. Generally, the Dijkstra algorithm is regarded as a benchmark to verify the optimality of other algorithms [11].

The optimization-based algorithms provide new solutions for motion planning, such as genetic algorithm (GA) [12], particle swarm optimization (PSO) [13] and recurrent spline optimization (RSO) [14]. These algorithms compute the motion trajectories by minimizing or maximizing the value function constrained by different variables. Nevertheless, the solving speed is influenced by the complexity of optimization model and the real-time performance is difficult to be guaranteed all the time.

The sampling-based algorithms are also popular in the motion planning, which sample the state space randomly and find the internal connectivity. As representatives, the rapidly-exploring random tree (RRT) [15] and the probabilistic roadmap method (PRM) [16] are commonly used to realize the online motion planning. In [17], the RRT-based method was proposed to achieve the optimal motion planning, and the results showed that the cost of the returned solution was asymptotically optimal. However, the solution of the traditional RRT algorithm is limited by its high computational burden and slow convergence rate [51]. To solve the above problems, the Sparse-RRT* based motion planning algorithm was presented to reduce the computational burden [52]. The experimental results validated the effectiveness of the motion planner. Besides, another variant of RRT* algorithm called RRT*-adjustable bounds was proposed to realize faster convergence to optimal solution, and simulation results supported the improved performance [53].

To solve the motion planning problem in the continuous space, the artificial potential field (APF) algorithm is applied to generate the real-time motion information [18]. In [19], an artificial potential field guided learning method was proposed for the optimal path planning in a warehouse. The results showed that the generated path avoided multiple obstacles successfully and reached the goal position efficiently. Another modified artificial potential field method was presented for the local motion planning of intelligent vehicles, and the planning performance was validated by the comparison with A* algorithm [20]. Nevertheless, the motion of vehicles tends to fall into the local optimum and fails to reach the target position in the complex external environment. Even though some variants of the APF are proposed to overcome the above problems, the impact on path smoothness and additional computational burden still need to be considered. Above all, according to the characteristics of multiple algorithms, proposing a motion planning layer with consideration of optimality, real-time and continuity is valuable.

In the research of tracking control, a variety of classical methods are applied, such as pure pursuit (PP) algorithm, PID algorithm, linear quadratic regulator (LQR) algorithm. The PP algorithm has reliable industrial applications, which solves the control strategies based on the vehicle position deviation, the heading angle deviation and the look-ahead distance. In [21], the PP method accommodating the steady state lateral dynamics was designed to track the path with high accuracy. The results showed that the proposed PP method performed better than the traditional PP method. Besides, the modified PP method was proposed to improve the tracking performance of unmanned vehicles by using the adjustable look-ahead distance, and the effectiveness of this method was verified [22]. However, the tracking control of PP algorithm is affected by the vehicle velocity and path curvature, which performs mediocre under large curvature and high velocity conditions. PID method is another popular tracking control choice, which has the advantages of engineering application and simplicity [23]. Han et al. [24] proposed an adaptive PID network for the tracking control of the intelligent vehicle. The simulation results showed that the proposed method had real-time capability in path tracking. Nevertheless, the PID method has poor universality, and the designed controller hardly adapts to the great change in driving condition. LQR algorithm works well in a variety of operating scenarios, which uses the state linear feedback. Peng et al. [25] designed a closed-loop system with a preview steering control algorithm for the tracking control of autonomous vehicles, and its effectiveness was validated by the real vehicle experiment. However, the application of this algorithm is still limited by the linear assumption [26].

In recent years, some researches on path tracking based on model predictive control (MPC) have been carried out. The MPC algorithm generates the control policy by minimizing the tracking deviation in a prediction horizon [27]. Feher et al. [28] applied MPC algorithm to the autonomous vehicle for evasive maneuvering, and the results showed that this method outperformed human drivers. In [29], a nonlinear model predictive control (NMPC) method was presented to balance the efficiency and accuracy in vehicle control scenarios. To deal with the computational burden of the traditional NMPC method, an improved NMPC method was developed for the tracking control through serially cascaded model complexity, and the performance was proved by the real-world experiment [30]. However, the MPC algorithm has the heavy computation load and high requirement for processing unit. The stability of applications in the field of tracking control is still being verified.

As an important part of artificial intelligence (AI) methods, machine learning algorithms have been applied in many fields of optimal control [31]. A reinforcement learning (RL) based path tracking method was proposed for the autonomous driving, which considered tracking accuracy and driving smoothness [32]. The results showed that the tracking strategy generated by RL method performed better than LQR method and MPC method. Ding et al. [33] proposed a partial reinforcement learning method for the adaptive tracking control. A numerical

simulation was presented to prove the effectiveness of this method. In addition, several variants of reinforcement learning methods have been applied to the path tracking of autonomous vehicles, such as configurable RL method [34], adversarial RL method [35], and safe RL method [36]. The reinforcement learning system can effectively solve dynamic tracking control problems because of its powerful self-learning ability. It should be noted that existing reinforcement learning algorithms mainly output single step optimal control strategy based on the current state [43], [44]. They have the characteristics of Markov decision process (MDP), which are not available in a whole horizon. Therefore, the complete control strategy sequence based on reinforcement learning algorithm is usually suboptimal. Referring to the advantages of MPC method, the future prediction horizon can effectively improve the effect of the overall control strategy. Incorporating the concept of predictive control into reinforcement learning algorithm has the potential to realize better performance of the tracking control for autonomous vehicles.

Furthermore, motion planning and control need to meet the safety requirements of the autonomous vehicle [47]. For this safety-critical problem, several studies have been carried out in recent years. Arab [48] designed the safety-guaranteed learning-predictive controller for the aggressive maneuvering of autonomous vehicle. The superior performance was tested and demonstrated experimentally. In [49], a risk-aware motion planning and control method was proposed to ensure the safety, and simulation results proved the utility of this method. Another safety-guaranteed motion control for autonomous vehicles was designed based on Udwadia–Kalaba approach and Lyapunov stability theory [50]. Numerical experiments were conducted to validate the effectiveness of the designed controller.

Besides, other research groups are also working on motion planning and tracking control for specific application scenarios [54]. For the automatic parking maneuver scenario, Chai et al. [55] proposed a desensitized trajectory optimization method and deep neural network-based control scheme. The results proved the performance and the real-time applicability of the proposed scheme. Another deep learning-based trajectory planning and tracking control framework was presented for the autonomous vehicle parking maneuver problem [56]. The results showed that the proposed framework can realize the parking mission with enhanced motion planning and control performance. For the automatic overtaking maneuver scenario, a new desensitized two-layer trajectory optimization scheme was proposed and a robust model predictive control scheme was applied [57]. The numerical simulations confirmed the optimality of the derived solution. To sum up, the research on motion planning and tracking control for specific application scenarios is also of great practical significance.

Generally, the decision layer and the control layer of autonomous vehicles are directly related [37]. In recent years, some researches have been carried out on the hierarchical framework containing both layers [38], [39], [40]. For the common hierarchical framework, the decision layer is placed in the upper part and the control layer is placed in the lower part. The global or local motion path is planned and output in the upper layer, then is passed to the lower layer as input to the tracking control system. According to the proposed decision and control process, the corresponding upper and lower logic associations need to be constructed to realize the operation of the whole hierarchical framework. Logical associations include not only connections between the upper and lower layers, but also connections between submodules in each layer. As described in the literature review above, the methods of motion planning and tracking control within the hierarchical framework can still be improved and optimized.

B. Innovation and Contributions

Inspired by the analysis of the literature review above, the motion planning and tracking control are the core research aspects of autonomous vehicle applications. Based on the current research progress and methods, exploring more optimization possibilities and further improving the planning and control performance are meaningful. Specially, constructing a more intuitive and efficient hierarchical framework to realize the cooperative operation of motion planning layer and tracking control layer is of practical significance. The innovation of this research is to realize the innovative combination of multiple algorithms based on the characteristics of these algorithms, and to achieve the efficient operation between and within these two layers in the constructed framework. The new planning scheme and control scheme perform better than several previous methods.

The main contributions of this paper can be summarized as follows:

1) The efficient hierarchical framework containing motion planning and tracking control for autonomous vehicles is constructed with algorithmic improvements and logical associations. The framework has the advantage of information rolling update and supports the real-time application of motion planning and tracking control.

2) The global heuristic planning based artificial potential field method (GHP-APF) is presented to obtain the real-time optimal motion sequence. In the motion planning scheme, the global heuristic reference generated by the modified A* method and the robust locally weighted regression smoothing is incorporated with the artificial potential field conception.

3) The multi-step forward predictive control method based on the prioritized Q-learning algorithm (PQL-FPC) is proposed to further improve the tracking control performance. In the tracking control scheme, the long horizon decision process is applied to the initial tracking controller trained by PQL instead of the single step decision process.

4) The numerical simulation, virtual driving environment simulation and real-world scenario are applied to validate the superior performance of the proposed framework than the previous methods.

C. Organization

The remainder of this paper is organized as follows. The schematic diagram and the control model of the problem are established, and the relevant formulations are given in

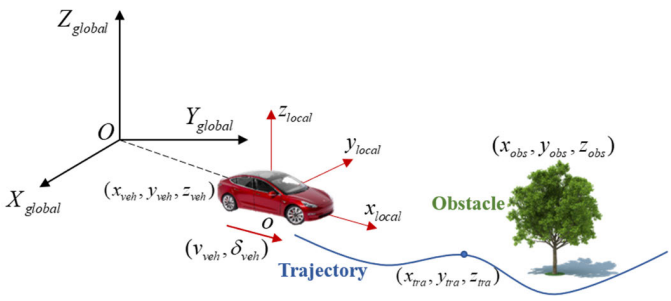


Fig. 1. The schematic diagram of the motion planning and tracking problem.

Section II. In Section III, the novel hierarchical framework of the motion planning and tracking control is proposed, which contains several algorithmic improvements and logical associations. The simulation results of the proposed hierarchical framework are shown and discussed in Section IV. Finally, Section V concludes this paper.

II. Modeling and Formulations of the Motion Planning and Tracking Control

A. Problem Description

According to the requirements of driving tasks, the autonomous vehicle needs to plan the optimal path in real time and track the path efficiently. The schematic diagram of the problem is described in Fig. 1.

$OX_{global}Y_{global}Z_{global}$ represents the inertial coordinate system, and $ox_{local}y_{local}z_{local}$ represents the local coordinate system which describes the motion of the autonomous vehicle. $(x_{tra}, y_{tra}, z_{tra})$ is the position information of a way point in the path, and $(x_{obs}, y_{obs}, z_{obs})$ is the position information of the obstacle. $(x_{veh}, y_{veh}, z_{veh})$ denotes the position information of the autonomous vehicle, and (v_{veh}, δ_{veh}) stand for the driving velocity and steering angle respectively. In this research, the vertical motion of the vehicle is not considered, and the longitudinal and lateral motions are analyzed. The kinematic model of the autonomous vehicle is shown in Fig. 2. (X_f, Y_f) and (X_r, Y_r) are the coordinates of the center points of the front and rear axles in the inertial coordinate system respectively. v_r represents the velocity of the rear axle center, and l denotes the wheelbase of the autonomous vehicle. R is the instantaneous turning radius of the rear axle center, and δ_f denotes the steering angle of the front axle. Besides, φ is the yaw angle of the autonomous vehicle.

The local coordinate system of the vehicle is established in the center of the rear axle. Several key variables have been described above. Therefore, the driving velocity of the autonomous vehicle can be described by:

$$v_r = \dot{X}_r \cos\varphi + \dot{Y}_r \sin\varphi \tag{1}$$

where $\dot{X}_r$ and $\dot{Y}_r$ denote the position derivatives in the X and Y directions of the inertial coordinate system respectively. The vector of driving velocity is the sum of the components of $\dot{X}_r$ and $\dot{Y}_r$ in the direction of driving.

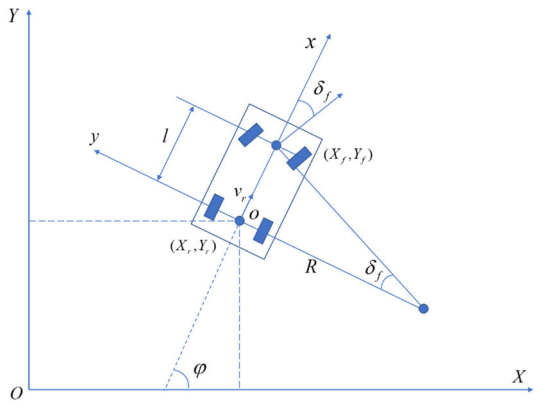


Fig. 2. The kinematic model of the autonomous vehicle.

Considering the kinematic constraints on front and rear axles, the relevant equations are as follows:

$$\begin{cases} \dot{X}_f \sin(\varphi + \delta_f) - \dot{Y}_f \cos(\varphi + \delta_f) = 0 \\ \dot{X}_r \sin\varphi - \dot{Y}_r \cos\varphi = 0 \end{cases} \tag{2}$$

where $\dot{X}_f$ and $\dot{Y}_f$ stand for the position derivatives of the front axle in the X and Y directions of the inertial coordinate system respectively. Obviously, the component of the driving velocity perpendicular to the direction of motion should be 0.

Then, the position derivative of the rear axle center can be derived by:

$$\begin{cases} \dot{X}_r = v_r \cos\varphi \\ \dot{Y}_r = v_r \sin\varphi \end{cases} \tag{3}$$

Based on the geometry of the front and rear axles:

$$\begin{cases} X_f = X_r + l\cos\varphi \\ Y_f = Y_r + l\sin\varphi \end{cases} \tag{4}$$

Then, the derivative of Equation (4) is taken, and this derivative equation is imported into Equation (2) together with Equation (3). By simplifying the combination of Equation (2 to 4), the yaw velocity and the instantaneous turning radius of the vehicle can be obtained:

$$\begin{cases} \omega = \dfrac{v_r}{l}\tan\delta_f \\ \dot{\varphi} = \omega \\ R = \dfrac{v_r}{\omega} \end{cases} \tag{5}$$

Combining the above formulas, the kinematic model can be described by:

$$\begin{pmatrix} \dot{X}_r \\ \dot{Y}_r \\ \dot{\varphi} \end{pmatrix} = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{pmatrix} v_r + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega = \begin{pmatrix} \cos\varphi \\ \sin\varphi \\ \tan\delta_f / l \end{pmatrix} v_r \tag{6}$$

Finally, the kinematic control model constructed is applied to the tracking control research of the autonomous vehicle.

B. Variables Setting

The hierarchical framework consists of a motion planning layer and a tracking control layer. In the motion planning

layer, the state variables are given by $S_{pln} = (x_{sta}, y_{sta}, x_{tar}, y_{tar}, Obs_1, Obs_2, \ldots Obs_{k-1}, Obs_k)$, where $(x_{sta}, y_{sta}) \in R^2$ and $(x_{tar}, y_{tar}) \in R^2$ are the positions of start point and target point, $k \in R$ is the number of obstacles, and $Obs_k = (x_{obs}^j, y_{obs}^j) | j = 1, 2, \ldots n$; $(x_{obs}^j, y_{obs}^j) \in R_k^2$ denotes the information of the k^{th} obstacle consisting of a large number of obstacle pixels. The planned path is given by $P_{pln} = (x_i, f(x_i)) \mid i = 0, 1, 2, \ldots m$; $(x_i, f(x_i)) \in R^2$, where m is the number of path points. $f(x)$ contains a series of specific polynomials, and each polynomial corresponds to a local segment in the whole path. In this research, the autonomous vehicle is regarded as a rigid body which occupies a certain area. The occupied area at the i^{th} path point can be represented by $A_{pln}(i) \subset R^2$, which contains the actual occupied area and safe margin area. The safe margin area is an outward extension of the actual occupied area, ensuring that the impassable area will not be crossed during the search for the next path point. Besides, the drivable surface area of the whole external environment for the autonomous vehicle is denoted by $B_{safe} \subset R^2$, and the impassable area is denoted by $B_{unsafe} = R^2 \setminus B_{safe}$. The cost function over the states and planned path is defined as $J_{plan}(S_{pln}, P_{pln})$, then the optimal solution can be obtained by the following equation:

$$\begin{aligned}
&\arg\min \ J_{plan}(S_{pln}, P_{pln}) \\
&\text{s.t. } \forall i \in \{0, \ldots, m\}: \\
&\quad S_{pln} = (x_{sta}, y_{sta}, x_{tar}, y_{tar}, Obs_1, \\
&\quad Obs_2, \ldots Obs_{k-1}, Obs_k) \\
&\quad (x_1, f(x_1)) = (x_{sta}, y_{sta}) \\
&\quad (x_m, f(x_m)) = (x_{tar}, y_{tar}) \\
&\quad A_{pln}(i) \subseteq B_{safe} \\
&\quad A_{pln}(i) = A_{pln}(x_i, f(x_i), R_{pln}) \\
&\quad R_{pln} = d_{search} \times Step_{\max}
\end{aligned} \tag{7}$$

where $A_{pln}(i)$ is determined by the information of the i^{th} path point. R_{pln} represents the radius of the safe margin area in path planning process, which is decided by single step search distance d_{search} and maximum search step $Step_{max}$.

In the tracking control layer, the state variables at time k are given by $s_k = (xr_k, yr_k, v_k, \Phi_k, cls_k) \in S$, where xr_k and yr_k represent the relative distances between the position of the autonomous vehicle and the reference path in the X and Y directions respectively. $v_k \in R$ denotes the velocity of the autonomous vehicle, and $\Phi_k \in R$ means the yaw angle of the autonomous vehicle. cls_k stands for the collision judgement based on the position of the vehicle and the information of obstacles. The path $P_{ref} = (X_{ref}, Y_{ref})$ generated by motion planning layer is regarded as the input reference for the tracking control layer, where (X_{ref}, Y_{ref}) represents the position sequence of path points:

$$(X_{ref}, Y_{ref}) = \begin{pmatrix} (x_{ref}, y_{ref})_1 \\ (x_{ref}, y_{ref})_2 \\ \vdots \\ (x_{ref}, y_{ref})_{m-1} \\ (x_{ref}, y_{ref})_m \end{pmatrix} \tag{8}$$

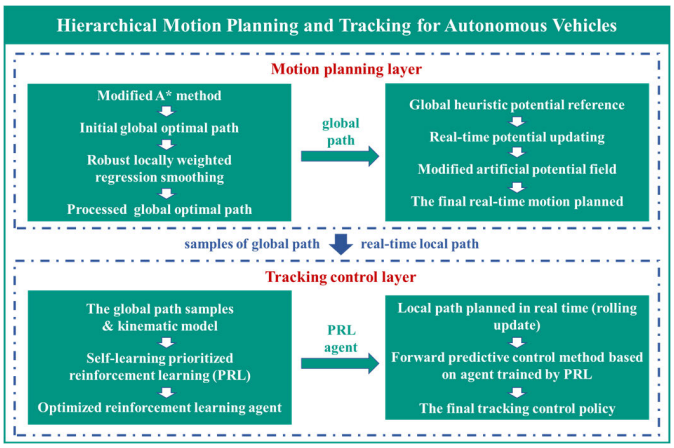


Fig. 3. The logical structure of the hierarchical framework.

The constantly updated position of the autonomous vehicle $(x_k, y_k) \in R^2$ and the information of obstacles Obs are also regarded as the inputs of the tracking control system.

The control actions at time k are given by $u_k = (a_k, \delta_k) \in U$, where $a_k \in R$ is the driving acceleration of the autonomous vehicle, and $\delta_k \in R$ is the steering angle of the autonomous vehicle. Then, the states update and inputs update are given by the following discrete dynamic formulation:

$$[s_{k+1}, (x_{k+1}, y_{k+1})] = f_{\Delta t}(s_k, u_k, (x_k, y_k), P_{ref}, Obs) \tag{9}$$

where $f_\Delta t$ represents the executive function parameterized by Δt which is 0.1 s. The occupied area of the autonomous vehicle $A_{ctr}(s_k, x_k, y_k) \subset R^2$, the drivable surface area $B_{safe} \subset R^2$, and the impassable area $B_{unsafe} = R^2 \setminus B_{safe}$ still hold in the tracking control layer. Based on the cost function $J(s_k, u_k)$, the optimal control strategies u^* can be computed by:

$$\begin{aligned}
u^* = \ &\underset{s_{1:N}, u_{0:N-1}}{\arg\min} \ J(s_{0:N}, u_{0:N-1}) \\
&\text{s.t. } \forall k \in \{0, \ldots, N-1\}: \\
&\quad [s_{k+1}, (x_{k+1}, y_{k+1})] \\
&\quad = f_{\Delta t}(s_k, u_k, (x_k, y_k), P_{ref}, Obs) \\
&\quad A_{ctr}(s_k, x_k, y_k) \subseteq B_{safe} \\
&\quad A_{ctr}(s_k, x_k, y_k) = A_{ctr}(x_k, y_k, \phi_k, R_{ctr}) \\
&\quad R_{ctr} = v_{\max} \times \Delta t
\end{aligned} \tag{10}$$

where $A_{ctr}(s_k, x_k, y_k)$ is determined by the k^{th} step state and input of the vehicle, containing the actual occupied area and safe margin area. R_{ctr} represents the radius of the safe margin area in tracking control process,which is decided by maximum velocity v_{max} and discrete time interval Δt. The safe margin area ensures the driving safety of the autonomous vehicle during the next time interval.

III. The Hierarchical Framework of Motion Planning and Tracking Control for Autonomous Vehicle

In this section, the logical structure of the hierarchical framework is described in detail, which is shown in Fig. 3.

The framework contains the motion planning layer and the tracking decision layer. Firstly, the global perception map is inputted into the motion planning layer, and the modified A-star algorithm is proposed to generate the initial global path efficiently. The motivation for the usage of modified A* algorithm mainly relies on its planning ability in the static global environment. The robust locally weighted regression smoothing (RLWR) method is applied to make the initial path smooth and continuous. The motivation of applying RLWR mainly relies on its capability of dealing with discrete points. Then, the processed global path is converted into the heuristic potential field reference, and the local potential field is updated according to the real-time perception of external environment changes, such as sudden obstacles. Based on the local potential field updating, the artificial potential field is used to plan the real-time motion sequences. The motivation for the use of APF mainly relies on its ability of real-time solution in local environment. Meanwhile, the global path is also inputted into the self-learning system as a series of the training samples. Based on the kinematic model of the autonomous vehicle and the path points to track, the prioritized reinforcement learning (PRL) algorithm is developed to optimize the tracking control agent. The motivation for the usage of PRL relies on its powerful self-learning ability in dynamic control problem. Afterwards, the trained agent based on PRL is combined with the forward predictive control method which can effectively improve the effect of the overall control strategy in the long horizon. Finally, the designed controller generates real-time tracking control policy while receiving the local path in rolling update.

A. The Modified A* Algorithm

The efficiency of the traditional A* algorithm is limited by the number of expanding points when dealing with the planning problem of large map. In the same search space, the number of expanding points is directly related to the time complexity. To improve the computation efficiency, the modified A* algorithm with variable step is designed in this research. For example, when the current search point is in an open area, the search strategy can be more aggressive and the search step can be set larger. Conversely, when the current search point is surrounded by obstacles, the search strategy can be more conservative and the search step can be set smaller. The search rule with variable step is formulated by:

$$S_{step}=\begin{cases} Step_{\max}, & if\ (d \geqslant R_{\max}) \\ Step_{\min}+\text{round}(\dfrac{d-R_{\min}}{R_{\max}-R_{\min}} \times (Step_{\max}-Step_{\min})), & if\ (R_{\max}>d>R_{\min}) \\ Step_{\min}, & if\ (d \leqslant R_{\min}) \end{cases} \tag{11}$$

where d is the nearest distance from the current position to the obstacles. R_{max} and R_{min} represent the maximum search radius and minimum search radius respectively. $Step_{max}$ and $Step_{min}$ denote the maximum search step and minimum search step respectively. The function round(x) means to take the integer of x. Besides, the heuristic function of the A* algorithm is as follows:

$$f(n)=g(n)+h(n) \tag{12}$$

where $g(n)$ represents the accumulative path cost at node n, and $h(n)$ represents the Euclidean distance from node n to target point. The pseudo-code of the modified A* algorithm is shown in the Appendix. B. Due to the characteristics of the A* algorithm, the path planning process is based on the particle model of the autonomous vehicle containing position information. Since the actual occupied area and safe margin area are introduced before, the safety of the planning can be guaranteed.

B. The Robust Locally Weighted Regression Smoothing

The path derived by the modified A* algorithm is not continuous and the smoothness is not guaranteed. Based on the requirement of heuristic potential field reference generation, the robust locally weighted regression smoothing method is applied to deal with the discrete path points. The locally weighted regression (LWR) method has been used for smoothing path or trajectory data in several studies [44], [45], [46]. In our research, the double regression process considering the robustness is introduced to avoid sharp way points and sharp turning corners.The discrete points of the whole path are piecewise fitted. For each piecewise path segment, the waypoints are ordered in time series. The fitting function is shown below:

$$f_{fitting}(x)=\sum_{j=0}^{d}\alpha_j x^j \tag{13}$$

where α_j is the coefficient of j^{th} degree term. Then, the optimization objective is to minimize the following formula:

$$\min_{\alpha}\ \sum_{i=1}^{N}K(x,x_i)\left(y_i-\sum_{j=0}^{d}\alpha_j x^j\right) \tag{14}$$

where (x_i, y_i) is the i^{th} initial point of path segment. $K(x, x_i)$ represents the distance weight function, which is described as follows:

$$K_l(x,x_0)=D(\frac{\|x-x_0\|}{l}) \tag{15}$$

where $D(t)$ denotes the cubic kernel function, which is formulated by:

$$D(t)=\begin{cases}(1-|t|^3)^3, & |t|<1, \\ 0, & otherwise.\end{cases} \tag{16}$$

Equation (14) can be converted to a matrix expression:

$$\min_{A}\ (Y-XA)^T K(Y-XA) \tag{17}$$

Then, the coefficients of the polynomial are derived by:

$$A=(X^T K X)^{-1}X^T K Y \tag{18}$$

At this stage, the initial polynomial $f_{fitting}(x)$ is formed. In previous studies, $f_{fitting}(x)$ can be directly applied as

continuous trajectory.However, the path generated by the traditional locally weighted regression may contain sharp turning corners which can not satisfy the kinematic constraints. Therefore, the second regression process avoiding sharp way points and sharp turning corners is proposed.

Based on the polynomial $f_{fitting}(x)$ obtained by the first locally weighted regression, the residual of the fitting can be calculated by:

$$e_i = y_i - f_{fitting}(x_i) \tag{19}$$

The residual weight is calculated by the double square function:

$$\delta_i = B(e_i/\kappa \cdot s) \tag{20}$$

where s is the median of the absolute value of the residuals, and κ denotes the outlier factor. $B(t)$ represents the double square kernel function, which is formulated by:

$$B(t) = \begin{cases} (1-t^2)^2, & |t| < 1, \\ 0, & otherwise. \end{cases} \tag{21}$$

The above function divides different way points according to the degrees of outliers, and the points far beyond the average degree of outliers are judged as sharp way points. By assigning the residual weight of sharp way points to 0, the formation of sharp turning corners can be avoided.

Then, the residual weight δ_i is multiplied by the distance weight function $K(x, x_i)$, and the locally weighted regression process is repeated as shown in Equations (17) and (18). Finally, the processed global path $P_{smooth} = (x, f(x)) \mid y_{sta} = f(x_{sta});\ y_{tar} = f(x_{tar});\ (x, f(x)) \in R^2$ is generated.

C. The Potential Field Method Based on Global Heuristic

The artificial potential field method utilizes virtual forces to plan the motion sequence. The APF algorithm has the advantages of high efficiency and simple calculation, and has the ability for the real-time planning. However, this method may fall into local optimum in the multi-obstacle environment. Besides, the autonomous vehicle based on the traditional APF method may have trouble passing through the narrow passage when two obstacles are close together. As analyzed in the literature review, the A* algorithm can reliably generate the global path, but hardly deal with the real-time changes of the local environment. Therefore, a motion planning strategy that satisfies both global planning and real-time response can be generated by combining the characteristics of these two methods. In this section, the real-time potential field method based on global heuristic reference is proposed. Specially, this motion planning process is still based on the particle model of the autonomous vehicle containing position information. Since the actual occupied area and safe margin area are introduced in this study, as shown in Section II-B, the safety of the planning can be guaranteed.

Based on the global path derived by the modified A* algorithm and RLWR method, the global heuristic potential field is created by the following equation:

$$U_{global}(x_{PF}, y_{PF}) = \begin{cases} 0, & (x_{PF}, y_{PF}) \in P_{smooth} \\ \frac{1}{2}\xi p_{ref}^2(x_{PF}, y_{PF}), & \\ & (x_{PF}, y_{PF}) \notin P_{smooth} \end{cases} \tag{22}$$

where P_{smooth} denotes the processed global path, and (x_{PF}, y_{PF}) is any position in the global potential field. ξ represents the heuristic gain constant, and p_{ref} is the distance function to the global path. ξ determines the gradient change rate of the global heuristic potential field. The larger value of ξ leads to the greater variation of the global heuristic potential field in unit distance. Then, the attractive potential field and the repulsive potential field are created based on the global heuristic potential field. The expression of the attractive potential field is as follows:

$$U_{att}(x_{PF}, y_{PF}) = \frac{1}{2}\tau p_{tar}^2(x_{PF}, y_{PF}, x_{tar}, y_{tar}) \tag{23}$$

where τ denotes the attraction gain constant, and p_{tar} is the distance function to the target point. τ determines the gradient change rate of the attractive potential field. The larger value of τ leads to the greater variation of the attractive potential field in unit distance. The expression of the repulsive potential field is as follows:

$$U_{rep}(x_{PF}, y_{PF}) = \begin{cases} \frac{1}{2}\eta(\frac{1}{p_{obs}(x_{PF}, y_{PF})} - \frac{1}{d_{obs}}), & \\ \quad p_{obs}(x_{PF}, y_{PF}) \leqslant d_{obs} & \\ 0, \quad p_{obs}(x_{PF}, y_{PF}) > d_{obs} & \end{cases} \tag{24}$$

where η denotes the repulsion gain constant, p_{obs} is the distance function to the obstacle, and the d_{obs} is the influence range of the obstacle. η determines the gradient change rate of the repulsive potential field. The larger value of η leads to the greater variation of the repulsive potential field in unit distance. Then, the total artificial potential field is the superposition of the global heuristic potential field, the attractive potential field, and the repulsive potential field. By setting the values of ξ, τ, and η reasonably, the expected interaction and superposition effect of the three potential fields can be achieved. In this research, ξ is set to 0.3, τ is set to 0.5 and η is set to 0.9. The expression of the total artificial potential field can be described as follows:

$$\begin{cases} U_{total}(x_{PF}, y_{PF}) = U_{global}(x_{PF}, y_{PF}) \\ \qquad\qquad + U_{dynamic}(x_{PF}, y_{PF}) \\ U_{dynamic}(x_{PF}, y_{PF}) = U_{att}(x_{PF}, y_{PF}) + U_{rep}(x_{PF}, y_{PF}) \end{cases} \tag{25}$$

The global heuristic potential field U_{global} is generated in advance based on the existing global map and is fixed, while the dynamic potential field $U_{dynamic}$ is generated in real time based on the local external environment changes, such as a sudden obstacle. The motion trajectory of the autonomous vehicle is obtained by the gradient of the total potential field as follows:

$$\nabla U_{total}(x_{PF}, y_{PF}) \tag{26}$$

Assuming that the external environment does not change, the autonomous vehicle can drive along the global path planned. Assuming that the optimal path is blocked due to the sudden changes in the external environment, the autonomous vehicle can generate the real-time local path planned by the potential field method to avoid sudden obstacles. Finally, the local optimal path is output from the motion planning layer in real time and updated continuously on a rolling basis.

D. The Prioritized Q-Learning Method

The global and local path information generated by the motion planning layer is input to the tracking control layer as training samples. Using the self-learning ability of reinforcement learning algorithm, the initial tracking controller is designed by the prioritized Q-learning (PQL) method. The PQL method mainly contains the Q-learning algorithm and the prioritized experience replay.

Since the tracking control of the autonomous vehicle is the sequential decision process, the value of performing a particular control in a particular state can be evaluated. The expectation of future cumulative rewards is used to represent the evaluation value of the particular control, when this control action is performed and the optimal control policy is followed thereafter [41]. Therefore, the optimal value function of the control u in the state s is formulated as follows:

$$\begin{aligned} V^*(s,u) = \max_{\pi}\ & E[\sum_{t=t_0}^{t=t_f}\gamma^t r(s_t,u_t)|s_{t_0}=s,u_{t_0}=u;\ \pi] \\ \text{s.t. } & \forall t \in \{t_1,t_2,\ldots,t_f\}: \\ & u_t \in \pi \\ & [s_{t+1},(x_{t+1},y_{t+1})] \\ & = f_{\Delta t}(s_t,u_t,(x_{t+1},y_{t+1}),P_{ref},Obs) \\ & A_{ctr}(s_t,x_t,y_t) \subseteq B_{safe} \\ & A_{ctr}(s_t,x_t,y_t) = A_{ctr}(x_t,y_t,\phi_t,R_{ctr}) \\ & R_{ctr} = v_{\max} \times \Delta t \end{aligned} \tag{27}$$

where π is a given follow-up optimal policy. E denotes the expectation function, and $r(s, u)$ denotes the reward function. γ represents the discount factor balancing the weights of current and follow-up rewards. In order to ensure the safety of the tracking control strategies, control constraints are also set in Equation (27), which are consistent with those of Equation (10). It should be noted that the value function in this research is the inverse of the loss function, which means that the value function is maximized while the loss function is minimized.

The value function can be converted to the Q matrix (QM) in this research, which evaluates the effectiveness of the tracking control strategies. Relying on the strong fitting ability, QM approximates the optimal value function through continuous training. The optimal value function in RL can be described as follows:

$$Q^*(s_t) = \max_{u_t}(r(s_t,u_t) + \gamma Q^*(s_{t+1})) \tag{28}$$

where s_t and u_t represent the current state and control action respectively, s_{t+1} stands for the next state. Based on Equation (29), the optimal control policy $\pi^*(s_t)$ can be derived by the following equation:

$$\pi^*(s_t) = \arg\max_{u_t}(r(s_t,u_t) + \gamma Q^*(s_{t+1})) \tag{29}$$

By solving the optimal tracking control of each step, the complete control sequence could be generated. Furthermore, the target value function and its optimal expression can be reformulated by:

$$\begin{cases} Q(s_t,u_t) = r(s_t,u_t) + \gamma \max\limits_{u_{t+1}} Q(s_{t+1},u_{t+1};\omega_t) \\ Q^*(s_t,u_t) = \max\limits_{u_t}(r(s_t,u_t) + \gamma \max\limits_{u_{t+1}} Q(s_{t+1},u_{t+1};\omega_t)) \end{cases} \tag{30}$$

where ω_t denotes the set of value factors in the Q matrix. The updating formula of Q matrix is described as follows:

$$\begin{aligned} Q(s_t,u_t;\omega_t') \leftarrow\ & Q(s_t,u_t;\omega_t) + \alpha[r(s_t,u_t) \\ & + \gamma \max_{u_{t+1}} Q(s_{t+1},u_{t+1};\omega_t) - Q(s_t,u_t;\omega_t)] \end{aligned} \tag{31}$$

where α and γ represent the learning rate and the discount factor respectively. Considering the requirements of accuracy, safety, rapidity and comfort, the reward function of the tracking control is set as the following equation:

$$\begin{aligned} r(s_t,u_t) = & k_{acu}\cdot f_{dis}(s_t,u_t) + k_{saf}\cdot f_{saf}(s_t,u_t,R_{ctr}) \\ & + k_{rap}\cdot f_{vel}(s_t,u_t,v_{\max}) + k_{com}\cdot f_{ctr}(u_t) \end{aligned} \tag{32}$$

where k_{acu}, k_{saf}, k_{rap} and k_{com} represent the accuracy coefficient, safety coefficient, rapidity coefficient and comfort coefficient respectively. f_{dis}, f_{saf}, f_{vel} and f_{ctr} denote the tracking error function, safety guaranteed function, driving velocity function and driving comfort evaluation function, respectively.

In the reinforcement learning implementation system, Q function is generated based on the reward function. The evaluation comprehensiveness of the reward function to the control strategy directly determines the training effect of Q function. In the tracking control problem, the autonomous vehicle is required to complete the high-accuracy tracking task of the locally planned path sequence on the premise of ensuring safety. In addition, the tracking rapidity and tracking comfort should be also taken into account. In terms of the accuracy evaluation, the relative distances between the position of the autonomous vehicle and the reference path are directly regarded as the state variables, so the reward function remains valid even if the reference path changes. In terms of the safety evaluation, the collision judgement is also directly regarded as the state variable, which means that the reward function is not influenced by the change of obstacles. Besides, the rapidity evaluation and comfort evaluation are only related with driving states of the autonomous vehicle. Given the above, the reward function is feasible in different scenarios involving reference paths and obstacles. Therefore, the trained Q function has the possibility to adapt to different scenarios.

Furthermore, the training of Q matrix depends on the experience replay. In the process of experience replay, the training samples are saved in the replay buffer and selected

according to the certain rule to optimize the matrix. In previous studies, the traditional experience replay samples the experiences randomly and ignores the importance differences of different experiences. To improve the training efficiency, the prioritized experience replay (PER) is applied to support the Q-learning algorithm. The temporal difference (TD) error is used to measure the error between the target value function and the actual value function for the particular experience sample. Then, the priority of the experience replay is positively correlated with the TD error formulated by:

$$\begin{cases} TD(s_t, u_t) = |Q^{Target}(s_t, u_t) - Q^{Actual}(s_t, u_t)| \\ Q^{Target}(s_t, u_t) = r(s_t, u_t) + \gamma \max\limits_{u_{t+1}} Q(s_{t+1}, u_{t+1}; \omega_t) \\ Q^{Actual}(s_t, u_t) = Q(s_t, u_t; \omega_t) \end{cases} \tag{33}$$

Afterwards, the probability of each experience sampled can be calculated by the following equation:

$$p_t = \frac{(TD(s_t, u_t) + \sigma)^\alpha}{\sum_{k=1}^{k=n_{buffer}} (TD(s_k, u_k) + \sigma)^\alpha} \tag{34}$$

where σ is used to avoid the failure of division. α denotes the priority factor, and the experiences will be sampled randomly if the value is equal to zero.

To judge the perfection of the Q matrix training, the generalized correlation coefficient is used in this research, which evaluates the similarity of matrices at different time nodes. The training can be judged to be completed when the value of coefficient is infinitely close to 1. The expression of the generalized correlation coefficient ρ is as follows:

$$\rho(Q_{current}, Q_{last}) = \frac{tr[\mathrm{cov}(Q_{current}, Q_{last})]}{\{tr[\mathrm{cov}(Q_{current})] \times tr[\mathrm{cov}(Q_{last})]\}^{1/2}} \tag{35}$$

where $Q_{current}$ and Q_{last} denote the current Q matrix and last Q matrix respectively. tr(A) represents the mathematical trace function of matrix A, and the covariance matrix cov() is described by: (36), shown at the bottom of the next page, where E(A) is the mathematical expectation function of matrix A, and n stands for the number of the time nodes.

E. The Forward Predictive Control Based on PRL Agent

The well-trained agent based on PRL is regarded as the initial tracking controller, and outputs the control strategy based on the current state. As analyzed in the literature review, the future prediction horizon can effectively improve the performance of the overall control strategy. In this research, the forward predictive control conception is incorporated into the PRL method. The long horizon decision process is applied to tracking control instead of the single step decision process. The solution logic of the control strategy has changed fundamentally compared with the conventional PQL method. A forward prediction horizon is constructed based on the current state, and PQL agent is used to calculate the cumulative rewards of different control sequences in the prediction horizon. Then, the first control action of the optimal control sequence in the prediction horizon will be selected for the autonomous tracking on the current state. The cumulative reward function in the forward prediction horizon is formulated by:

$$\begin{aligned} J(s_k, u_k) &= Q(s_k, u_k) + \sum_{i=k+1}^{k+p} Q^*(s_i) \\ &= Q(s_k, u_k) + \sum_{i=k+1}^{k+p} \lambda_i \max_{u_i} Q(f_{\Delta t}(s_{i-1}, u_{i-1}), u_i) \end{aligned} \tag{37}$$

where p represents the length of the prediction horizon, and λ is the prediction attenuation factor which decreases as the prediction progresses. $f_{\Delta} t$ represents the executive function parameterized by sampling period Δt. Then, the control action for the autonomous vehicle on the state s_k can be derived by:

$$\begin{aligned} u^*(s_k) &= \arg\max_{u_k} J(s_k, u_k) \\ &= \arg\max_{u_k} [Q(s_k, u_k) \\ &\quad + \sum_{i=k+1}^{k+p} \lambda_i \max_{u_i} Q(f_{\Delta t}(s_{i-1}, u_{i-1}), u_i)] \end{aligned} \tag{38}$$

The whole forward predictive control process is based on the well-trained Q matrix instead of the complex model and optimization algorithm, so the solving speed is fast enough to meet the requirement of the real-time application. The detailed framework of the forward predictive control based on PRL agent for the autonomous tracking is shown in Fig. 4. Besides, the main parameters of the proposed PQL-FPC method are listed in Table I.

IV. The Simulation Validation and Analysis

In this section, the performance of the hierarchical framework contains motion planning and tracking control is validated by the numerical simulation and virtual driving environment simulation. The global heuristic planning based artificial potential field method (GHP-APF) and the prioritized Q-learning based forward predictive control method (PQL-FPC) are compared with some existing methods in this research.

A. The Simulation Results of Motion Planning

In the global path planning part, the traditional Dijkstra and A star algorithms are introduced as comparisons with the modified A star algorithm with robust locally weighted regression smoother. For the constructed global map with the unit of two meters, the global paths planned by the three algorithms are shown in Fig. 5. The red star represents the start point and the green diamond represents the target point. The black coverage areas are obstacles in the global map. In this scenario, a large obstacle exists near the start point. An empty area in the right part of the scene is available for planning an intuitive path from the start point to the target point, but this leads to a longer path length. To realize the shortest path planning, the path has to pass through a group of obstacles, especially a narrow and long alley. Therefore, facing

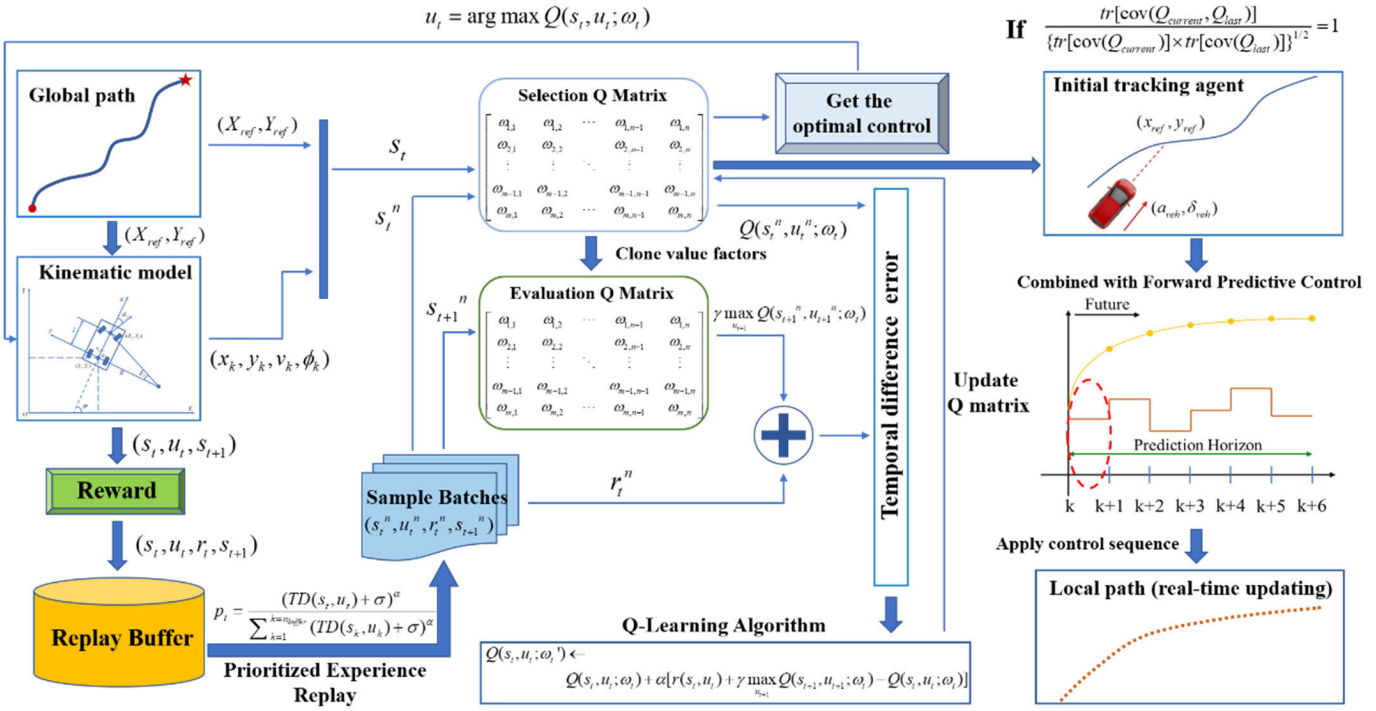


Fig. 4. The framework of the forward predictive control based on PRL agent.

TABLE I
MAIN PARAMETERS OF PQL-FPC METHOD

Names	Values
Discount factor γ	0.3
Learning rate α	0.975
Initial exploration factor ε	0.5
Exploration attenuation rate υ	0.99
Replay buffer capacity N	2000
Sample batches size S	12
Forward prediction step h_p	6
Prediction attenuation factor λ	0.35

more complex planning conditions with the goal of achieving planning optimality is typical. Obviously, the three paths have very similar trajectories, which proves the feasibility of the proposed planning algorithm.

The total length of path, the calculation time and the maximum cumulative curvature are illustrated in Table II. Since Dijkstra algorithm is regarded as the benchmark reference, the path generated by this algorithm is the shortest. As we can see, the total length of the path generated by modified A* algorithm with RLWR is very close to the shortest length, which means that the loss of optimality is small enough while smoothing the path. Besides, the Dijkstra algorithm has the longest calculation time, A* algorithm has a shorter calculation time due to the heuristic function, while the modified A* algorithm with RLWR has the shortest calculation time depending on the variable search step. The maximum cumulative curvature is used to measure the smoothness of the global path, and less curvature means smoother path. It is evident that the path curvature of the proposed algorithm is much less than that of the other two algorithms, which indicates the effect of the robust locally weighted regression smoothing method on the path.

In order to further analyze the satisfaction extent of the proposed method in optimality, rapidity and smoothness. The relative comparison of paths planned by three algorithms is shown in Table III. The comparison mainly involves three aspects which are total length, calculation time and maximum cumulative curvature. The best result of each aspect is selected as the benchmark, and the relative increases of other results

$$\begin{cases} \mathrm{cov}(Q_{current}) = \dfrac{[Q_{current} - E(Q_{current})]^T \times [Q_{current} - E(Q_{current})]}{n-1} \\ \mathrm{cov}(Q_{last}) = \dfrac{[Q_{last} - E(Q_{last})]^T \times [Q_{last} - E(Q_{last})]}{n-1} \\ \mathrm{cov}(Q_{current}, Q_{last}) = \dfrac{E\{[Q_{current} - E(Q_{current})]^T \times [Q_{last} - E(Q_{last})]\}}{1} \end{cases} \tag{36}$$

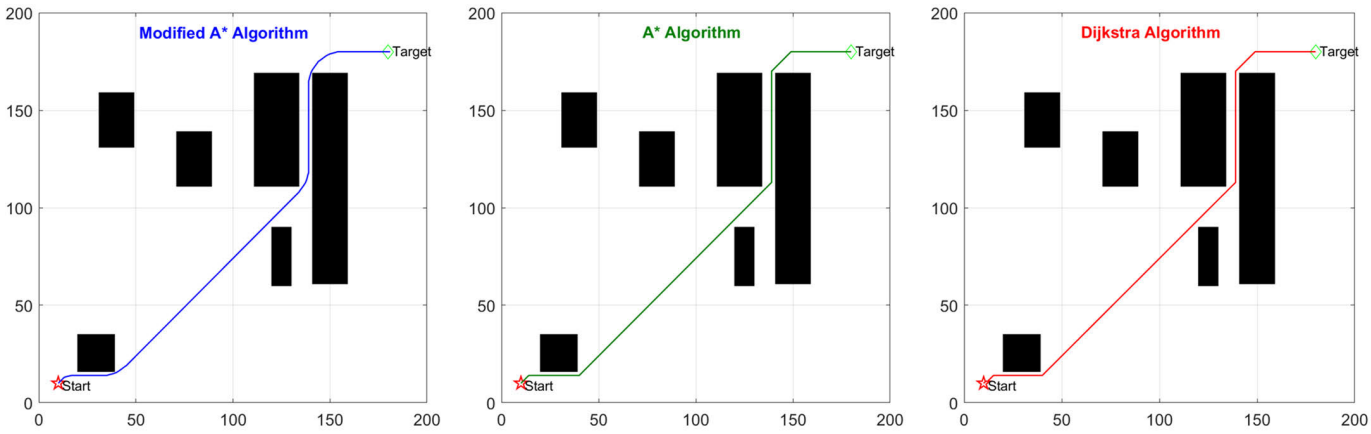


Fig. 5. The global paths planned by three algorithms.

TABLE II
THE RESULTS OF PATHS PLANNED BY THREE ALGORITHMS

Algorithms	Total Length (m)	Calculation Time (s)	Maximum Cumulative Curvature
Dijkstra	550.44	15.34	3.58
A star	551.18	9.68	2.15
Modified A star with RLWR	551.36	6.15	1.33

TABLE III
THE RELATIVE COMPARISON OF PATHS PLANNED BY THREE ALGORITHMS

Algorithms	Relative Increase (%)		
	Total Length	Calculation Time	Maximum Cumulative Curvature
Dijkstra	-	149.4%	169.2%
A star	0.1%	57.4%	61.7%
Modified A star with RLWR	0.1%	-	-

to the benchmark are calculated. In terms of the total length, the relative increase of modified A star with RLWR to the benchmark is only 0.1%, which means that the proposed method does not sacrifice length optimality while considering other aspects. In terms of the calculation time, the relative increases of A star and Dijkstra to the benchmark are 57.4% and 149.4% respectively, which indicates that the proposed method satisfies the goal of rapidity well. In terms of the maximum cumulative curvature, the relative increases of A star and Dijkstra to the benchmark are 61.7% and 169.2% respectively, which proves that the proposed method also satisfies the goal of smoothness well. Therefore, the optimality, rapidity and smoothness of the proposed path planning method are validated.

Based on the global path derived by the modified A* algorithm and RLWR method, the initial heuristic potential field is created as shown in Fig. 6. Assuming that there is

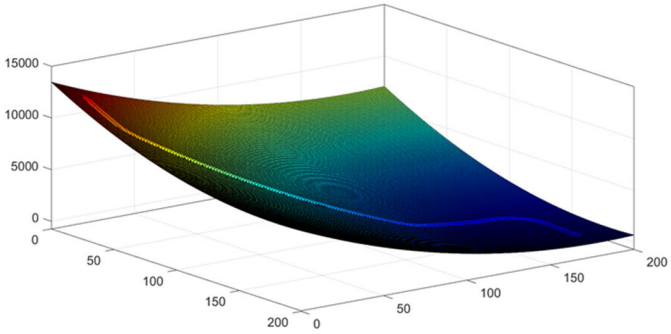


Fig. 6. The initial global heuristic potential field.

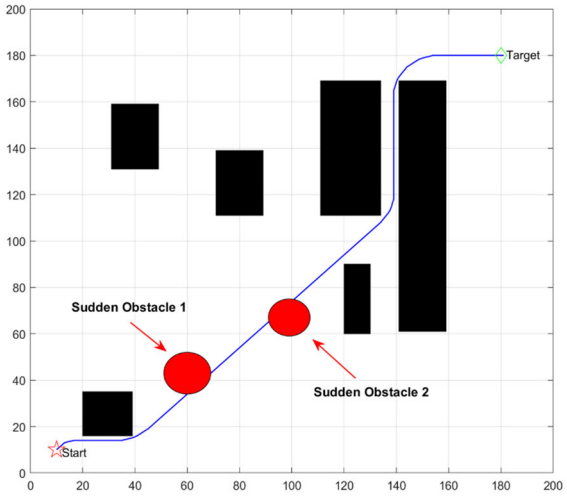


Fig. 7. The global planning map with sudden obstacles.

no sudden obstacle, the autonomous vehicle will move in the descending direction of the potential field. This paper constructs a scene with obstacles suddenly appearing which is shown in Fig. 7. When obstacle 1 and obstacle 2 appear suddenly in the process of vehicle driving, the autonomous vehicle needs to carry out real-time local path replanning to avoid the obstacles. The appearance of this uncertainty will inevitably lead to additional driving length. Since ensuring driving safety is the most important indicator, the sacrifice

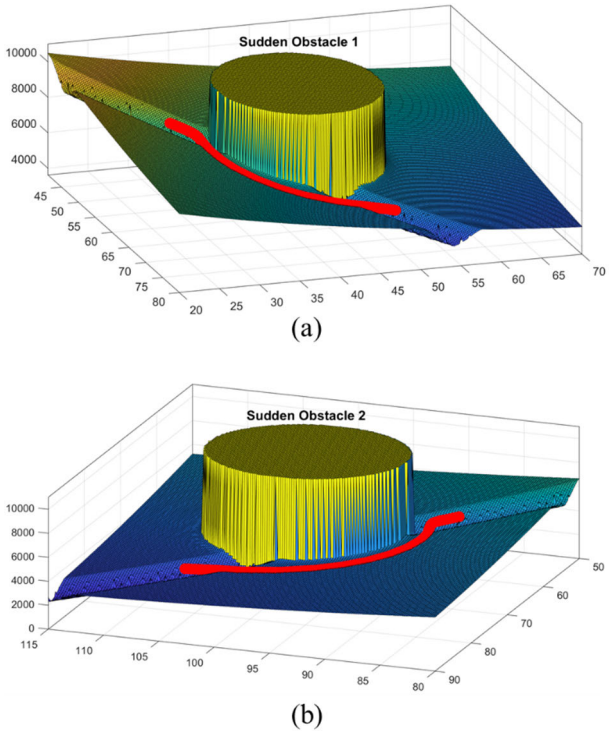


Fig. 8. The three-dimensional maps of two real-time local potential fields.

of the planned path length is necessary. The difficulty and challenge of this pattern is to plan the motion sequence through the narrow and long alley which can be used for the safe tracking control. The sudden appearance of obstacles also increases the risk of collision and causes more challenges to the planning and control.

The three-dimensional maps of two real-time local potential fields are illustrated in Fig. 8. Benefitting from the efficient calculation ability of the artificial potential field method, the new potential fields are updated immediately. Based on the newly generated local potential field, the autonomous vehicle plans new motion trajectories and returns to the global path after avoiding obstacles. The red curves in maps represent the local planned trajectories. The replanning time of traditional Dijkstra method, A* method, modified A* method and GHP-APF method is shown in Table IV. Obviously, GHP-APF method can deal with local environment planning much faster than the other three methods and is capable of being used to real-time scenarios. Due to the perception of obstacles in a wide range and the construction of safe margin area, the driving safety during the replanning period can be guaranteed.

Furthermore, the effects of the local paths replanned by four methods are shown in Table V. As can be seen, the extra avoidance distance of the GHP-APF method is very close to that of Dijkstra method, which means that the autonomous vehicle can avoid obstacles and returns to the reference path with the shortest possible distance.

B. The Simulation Results of Tracking Control

The designed control layer generates real-time tracking control policy while receiving the planned path in rolling

TABLE IV
THE LOCAL PATH REPLANNING TIME OF FOUR METHODS

Methods	Obstacle 1 Replanning Time (s)	Obstacle 2 Replanning Time (s)
Dijkstra	2.95	2.86
A star	2.37	2.19
Modified A star	2.29	2.11
GHP-APF	0.57	0.55

TABLE V
THE LOCAL PATH EXTRA DISTANCES OF FOUR METHODS

Methods	Obstacle 1 Extra Avoidance Distance (m)	Obstacle 2 Extra Avoidance Distance (m)
Dijkstra	9.76	10.02
A star	10.02	10.33
Modified A star	10.05	10.37
GHP-APF	9.81	10.06

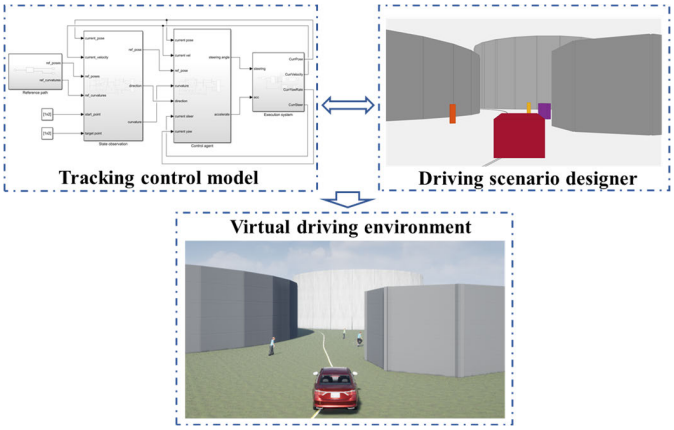


Fig. 9. The schematic diagram of the virtual driving environment simulation.

update. In this section, the pure pursuit method, the traditional Q-learning (QL) method and the prioritized Q-learning method are applied as comparisons to validate the performance of the proposed PQL-FPC method. The virtual driving environment is built to simulate the tracking control of the autonomous vehicle. The relevant schematic diagram is shown in Fig. 9.

The path tracking trajectories of the four control methods are shown in Fig. 10. As we can see, all the four control strategies complete path tracking successfully, including the global path and the local replanned path. From the partial enlargement of the red rectangle, the tracking accuracy of the proposed PQL-FPC method is the highest. Because this part of the path with a large turning radius is challenging to track, the traditional PP method leads to the largest tracking error. Compared with the PQL method, the high-accuracy tracking performance of the PQL-FPC method proves the effectiveness of the forward predictive control (FPC). Therefore, applying the long horizon decision process instead of the single step decision process in the reinforcement learning method improves the tracking accuracy.

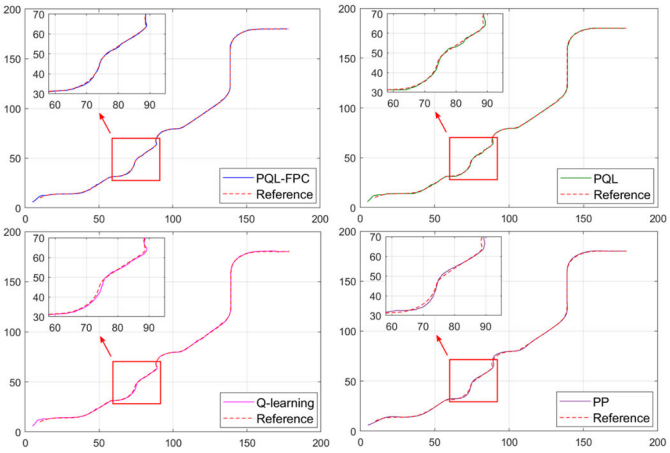


Fig. 10. The tracking trajectories of four control methods.

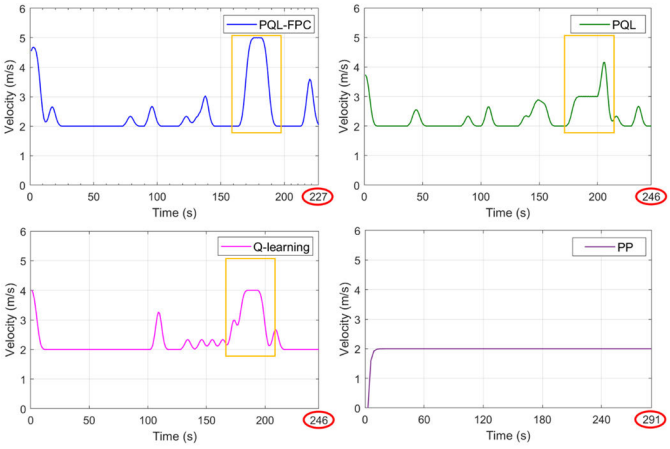


Fig. 11. The driving velocities of the autonomous vehicle with four control methods.

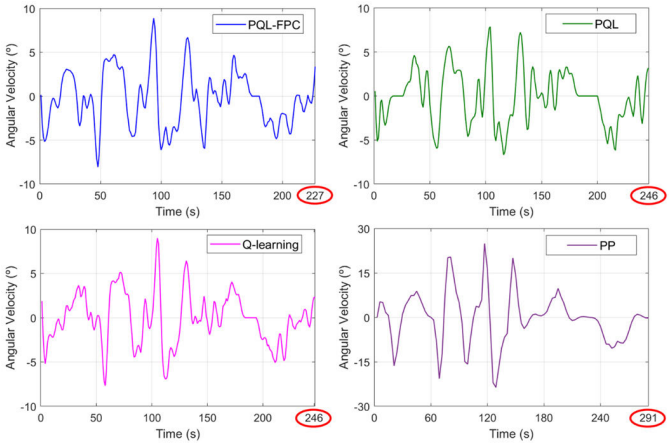


Fig. 12. The steering angular speeds of the autonomous vehicle with four control methods.

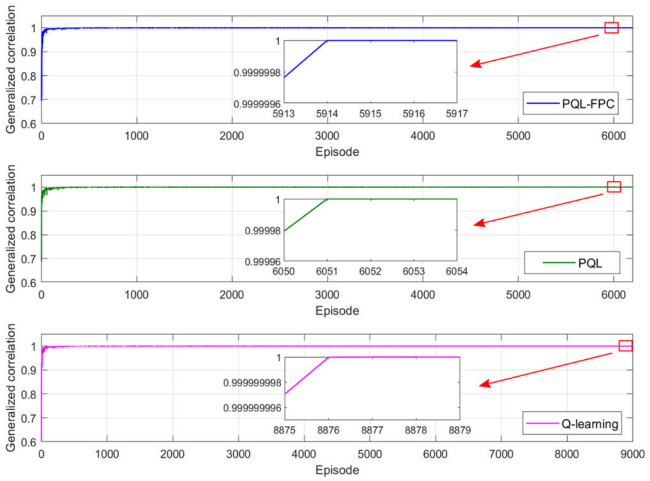


Fig. 13. The trend curves of the generalized correlation coefficients for three methods.

Fig. 11 shows the driving velocities of the autonomous vehicle with four control methods. The PP method uses PID to keep the driving velocity, and the value of the velocity is set to 2 m/s. The lower threshold of velocities for the other three methods is set to 2 m/s to evaluate their performance at higher driving velocities. Obviously, the proposed PQL-FPC method can achieve and keep the highest driving velocity in the yellow rectangle range. The corresponding part of the path happens to pass through the narrow roadway shown in Fig. 7, so driving at high velocity during this period is very reasonable. Besides, the PQL-FPC method completes the tracking task fastest, while PP method completes the tracking task slowest. The path tracking time of the PQL-FPC method, the PQL method, the QL method and the PP method is 227 s, 246 s, 246 s and 291 s respectively. Therefore, the tracking rapidity of the proposed method is proved. Furthermore, the steering angular speeds of the autonomous vehicle with four control methods are shown in Fig. 12. It can be seen that the angular speeds of the autonomous vehicle controlled by three learning methods fluctuate between $-10°$ and $10°$, which indicates the good driving comfort. The angular speed fluctuation of the PP method exceeds the interval $[-20°, 20°]$, which is influenced by its working mechanism. Above all, the tracking accuracy, tracking rapidity and tracking comfort of the PQL-FPC method are proved.

In terms of the training process of the reinforcement learning algorithm, the trends of the generalized correlation coefficients for these three learning-based methods are illustrated in Fig. 13. All three curves rise rapidly at the beginning of training, indicating that Q matrices are rapidly updated during this period. From the partial enlargement, the generalized correlation of the PQL-FPC method reaches 1 at the 5914^{th} training episode, which means that the control matrix has been fully trained at the time node. The generalized correlations of the PQL method and the Q-learning method reach 1 at the 6051^{th} and the 8876^{th} training episodes respectively. The PQL-FPC method and the PQL method complete the training process much faster than the traditional Q-learning method, which means that the prioritized experience replay can facilitate the training significantly.

To further validate the training effectiveness, Fig. 14 illustrates the variations of the value function errors for these three learning-based methods. It can be seen that the value function errors of the PQL-FPC method and the PQL method approach zero faster than that of the traditional Q-learning

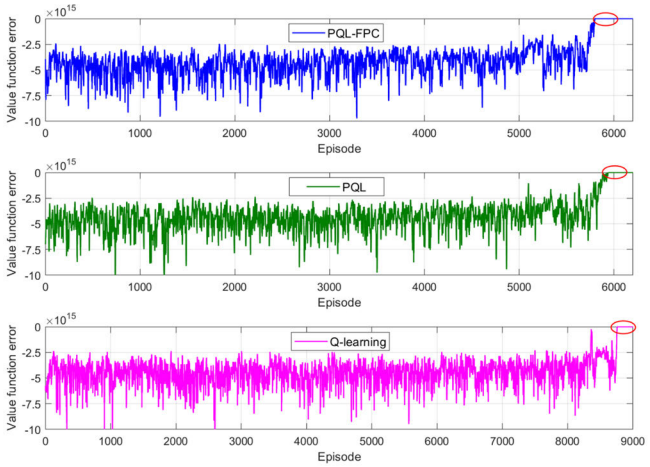


Fig. 14. The variations of the value function errors for three learning-based methods.

TABLE VI
THE RESULTS OF TRACKING CONTROL GENERATED BY FOUR METHODS

Algorithms	Total Tracking Error (m)	Maximum Tracking Error (m)	Training Time (s) / Calculation Time (s)
PQL-FPC	36.17	0.51	567.23 / 0.29
PQL	88.87	1.13	579.10 / 0.16
Q-learning	93.62	1.17	863.34 / 0.17
PP	185.05	2.30	- / 0.37

method. Besides, the results of the value function errors are consistent with those of the generalized correlation coefficients (such as the red rectangles and ellipses depicted in Fig. 13 and Fig. 14).

The comparisons of the tracking strategies generated by these four control methods are shown in Table VI. Obviously, the total tracking error of the PQL-FPC method is much smaller than those of the other methods, and the PP method has the largest total tracking error. The maximum tracking error is also the same result as the total error, which proves the effectiveness of the forward predictive control in the PQL-FPC method. Furthermore, the training time of the proposed method is shorter than that of the Q-learning method and similar to that of the PQL method, because the training mechanism of the PQL-FPC method and the PQL method is the same. The calculation time of the proposed method in the tracking control of the autonomous vehicle is 0.29 s, which provides the possibility for real-world applications.

C. The Adaptability Validation of Hierarchical Framework

To verify the adaptability of the proposed hierarchical framework, another driving scenario is constructed to evaluate the performance of the motion planning and tracking control. In this scenario, obstacles with different sizes are randomly scattered and distributed, which is a common situation encountered in the driving of autonomous vehicle. The potential optimal path selection is diversified, and determining the most reasonable path trend among them belongs to a typical problem. For the new map with the unit of two meters, the global paths planned by the three algorithms are shown in Fig. 15. Obviously, all three trajectories have similar trends, especially the trajectory of the modified A* algorithm with RLWR is smoother than other trajectories. Table VII shows the comparisons of the total length, the calculation time and the maximum cumulative curvature. As can be seen, the total length of the path planned by the modified A* algorithm with RLWR is still very close to that of the shortest path planned by the Dijkstra algorithm. Besides, the modified A* algorithm with RLWR still has the least calculation time and the smallest maximum cumulative curvature. The results are consistent with those in the first scenario.

TABLE VII
THE RESULTS OF PATHS PLANNED FOR ADAPTABILITY VALIDATION

Algorithms	Total Length (m)	Calculation Time (s)	Maximum Cumulative Curvature
Dijkstra	551.47	15.57	3.57
A star	552.12	9.52	2.86
Modified A star with RLWR	552.16	6.19	1.28

TABLE VIII
THE RELATIVE COMPARISON OF PATHS PLANNED FOR ADAPTABILITY VALIDATION

Algorithms	Relative Increase (%)		
	Total Length	Calculation Time	Maximum Cumulative Curvature
Dijkstra	-	151.5%	178.9%
A star	0.1%	53.8%	123.4%
Modified A star with RLWR	0.1%	-	-

Furthermore, Table VIII illustrates the relative comparison of paths planned by three algorithms to evaluate the satisfaction extent of the proposed method in optimality, rapidity and smoothness. In terms of the total length, the relative increase of modified A star with RLWR to the benchmark is only 0.1%, which indicates that the proposed method can still maintain length optimality while satisfying other aspects. In terms of the calculation time, the relative increases of A star and Dijkstra to the benchmark are 53.8% and 151.5% respectively, which means that the proposed method still satisfies the goal of rapidity well. In terms of the maximum cumulative curvature, the relative increases of A star and Dijkstra to the benchmark are 123.4% and 178.9% respectively, which indicates that the proposed method also satisfies the goal of smoothness well. Therefore, the optimality, rapidity and smoothness of the proposed path planning method are guaranteed in the adaptability verification scenario.

The scene with obstacles suddenly appearing and the local path replanned are shown in Fig. 16. The difficulty and challenge of this pattern is to plan the most reasonable motion

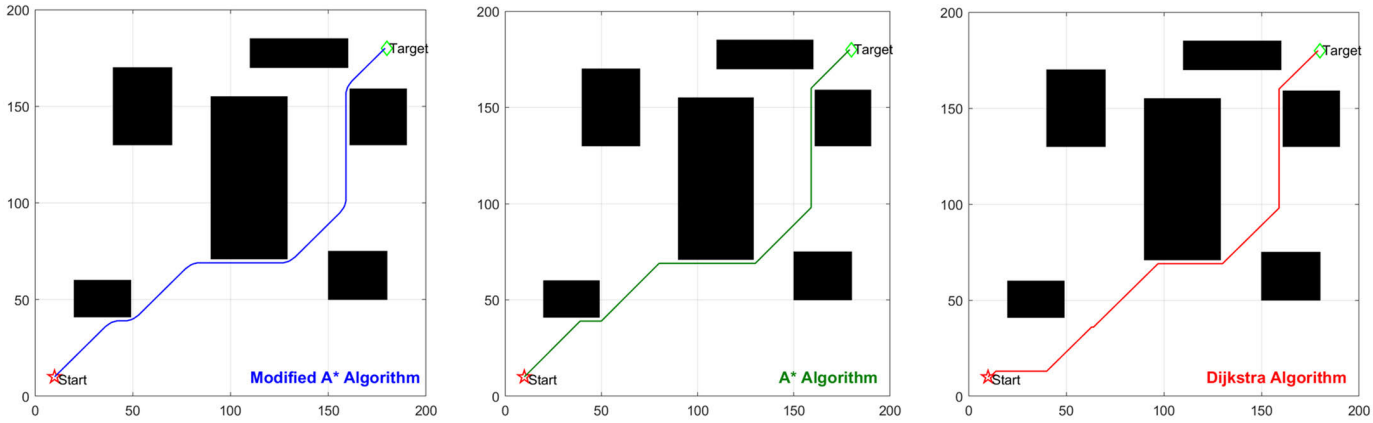


Fig. 15. The global paths planned by three algorithms for adaptability validation.

TABLE IX
THE LOCAL PATH REPLANNING TIME FOR ADAPTABILITY VALIDATION

Methods	Obstacle 1 Replanning Time (s)	Obstacle 2 Replanning Time (s)
Dijkstra	2.79	2.16
A star	2.18	1.75
Modified A star	2.06	1.69
GHP-APF	0.51	0.42

TABLE X
THE LOCAL PATH EXTRA DISTANCES FOR ADAPTABILITY VALIDATION

Methods	Obstacle 1 Extra Avoidance Distance (m)	Obstacle 2 Extra Avoidance Distance (m)
Dijkstra	10.31	5.26
A star	10.55	5.51
Modified A star	10.60	5.53
GHP-APF	10.36	5.30

sequence from a variety of potential optimal trends. The sudden appearance of obstacles also increases the risk of collision and causes more challenges to the planning and control. It is observed that the autonomous vehicle replans the real-time local path successfully to avoid the obstacles. Furthermore, the local path replanning time of these four methods is illustrated in Table IX. As we can see, the proposed GHP-APF method can still deal with local planning much faster than the other three methods. Besides, the effects of the local paths replanned by four methods are shown in Table X. Obviously, the extra avoidance distance of the GHP-APF method is still very close to that of Dijkstra method, which proves that the autonomous vehicle can avoid obstacles and returns to the reference path with the shortest possible distance. Above all, the adaptability of the hierarchical framework in motion planning has been verified.

In the adaptability verification of the tracking control part, five control methods are applied including the PQL-FPC (new) method, the PQL-FPC (original) method, the PQL method,

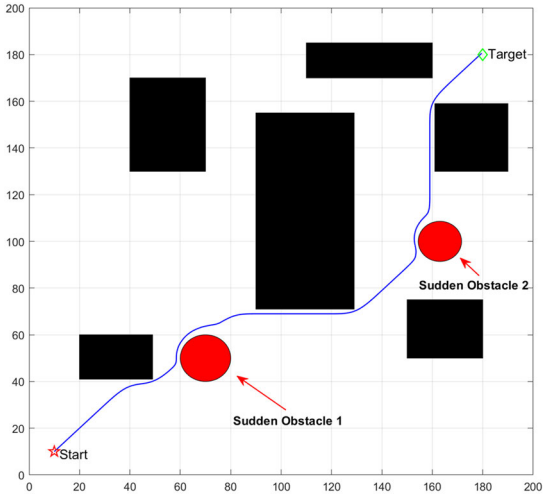


Fig. 16. The scene with obstacles suddenly appearing and the local path replanned.

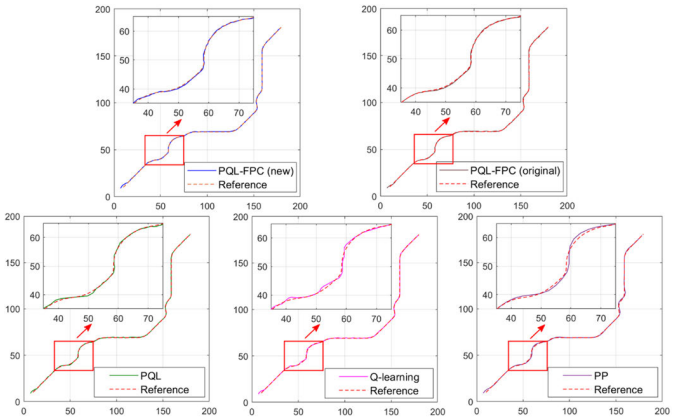


Fig. 17. The tracking trajectories of five control methods for adaptability validation.

the traditional Q-learning method and the PP method. For the PQL-FPC (new) method, the PQL agent is trained using the experience samples from the new scenario. Then, the trained agent is applied again to the tracking control of the new scenario in combination with the forward predictive control. For the PQL-FPC (original) method, the PQL agent is trained

TABLE XI
THE RESULTS OF TRACKING CONTROL FOR ADAPTABILITY VALIDATION

Algorithms	Total Tracking Error (m)	Maximum Tracking Error (m)	Training Time (s)/ Calculation Time (s)
PQL-FPC (new)	29.05	0.39	550.93 / 0.30
PQL-FPC (original)	35.62	0.52	- / 0.29
PQL	82.89	0.91	559.28 / 0.17
Q-learning	90.37	1.02	838.62 / 0.18
PP	170.94	2.81	- / 0.39

using the experience samples from the first scenario. Then, the trained agent is applied directly to the tracking control of the new scenario in combination with the forward predictive control. The performance of the PQL-FPC (original) method in the new scenario indicates whether the well-trained Q-matrix can adapt to different scenarios. The tracking trajectories of these five control methods are shown in Fig. 17. It is obvious that the PQL-FPC (new) method and the PQL-FPC (original) method can achieve the high-accuracy tracking effectiveness compared with other three methods. Furthermore, the comparisons of the tracking strategies generated by these five control methods are shown in Table XI. As can be seen, both the total tracking errors and the maximum tracking errors of these two PQL-FPC methods are much smaller than those of other three methods. Especially, the PQL-FPC (original) method can be used directly in the new scenario without further training, which has been trained in the first scenario. The calculation time of the PQL-FPC methods still meets the requirement of the real-time tracking control. The results are consistent with those in the first scenario. Above all, the adaptability of the hierarchical framework in tracking control has been proved.

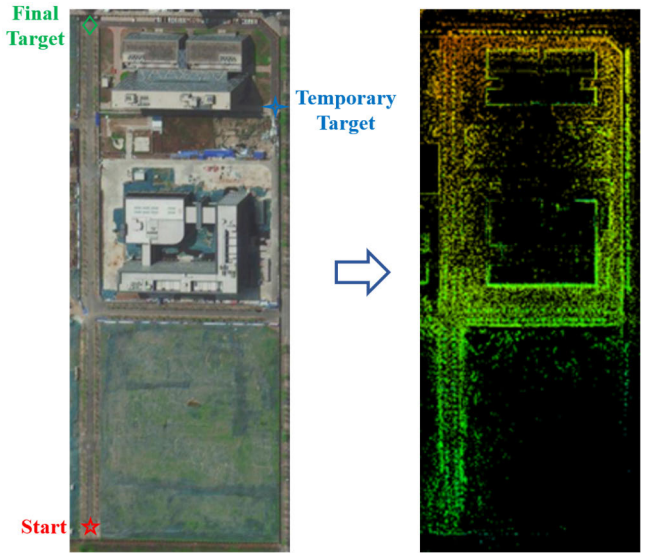


Fig. 18. The satellite map and points cloud map of the real-world scenario.

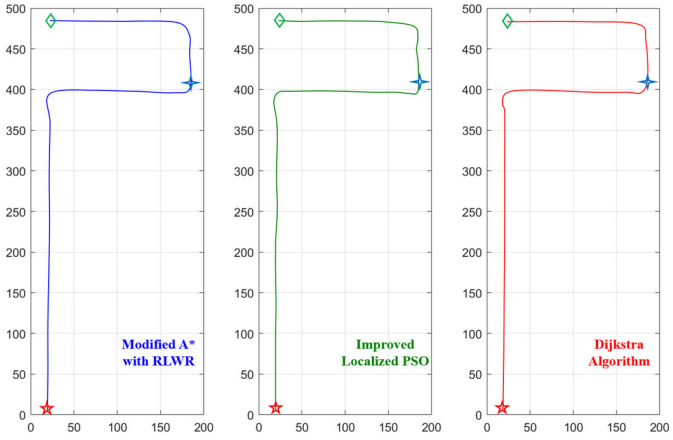


Fig. 19. The global paths of different algorithms in real-world scenario.

D. The Further Validation in The Real-World Scenario

Based on the above results and discussion, the performance of the proposed framework has been preliminarily verified. To further validate the feasibility and effectiveness of the proposed framework, a real-world scenario is introduced, which is the part of the Liangxiang Innovation and Technology Center. The satellite map based on global positioning system (GPS) and the points cloud map based on simultaneous localization and mapping (SLAM) are shown in Fig. 18.

In this scenario, a start point, a temporary target point, and a final target point are set up. The autonomous vehicle is required to firstly drive from the start point to the temporary target point, then drives from the temporary target point to the final target point. In order to improve the complexity of the verification, the driving speed requirement of the autonomous vehicle is further increased, and two external moving vehicles are treated as uncertainties. Besides, the motion planning performance of the proposed framework is compared with the improved localized particle swarm optimization algorithm (ILPSO) proposed in [13], and the tracking control performance of the proposed framework is compared with the asynchronous multithreading proximal policy optimization-based trajectory tracking method (AMPPO-TT) proposed in [40].

In terms of the motion planning validation, the global path and local real-time motion trajectory planned by the proposed method are evaluated. The global paths of different algorithms are shown in Fig. 19. It is evident that all three paths successfully complete two stages of planning and have similar trends. Especially, the global path of the modified A* algorithm with RLWR is smoother than other global paths.

Table XII shows the comparison results and relative increases of the paths planned by different algorithms. As can be seen, the total length of the proposed algorithm is only 0.2% longer than that of the global benchmark (Dijkstra algorithm), and is shorter than that of the ILPSO algorithm. The comparison result indicates that the modified A* with RLWR maintains the length optimality while considering other factors. Besides, the calculation time and maximum cumulative curvature of the proposed algorithm are also the least compared with those of other algorithms. Therefore, the proposed algorithm performs better than the ILPSO in length optimality, planning rapidity and path smoothness.

TABLE XII
THE COMPARISON RESULTS AND RELATIVE INCREASES OF DIFFERENT PATHS

Algorithms	Total Length (m)	Calculation Time (s)	Maximum Cumulative Curvature
	(Relative Increase (%))		
Dijkstra Algorithm	760.35	21.94	5.29
	(-)	(153.6%)	(86.9%)
Improved Localized PSO	763.24	11.03	3.16
	(0.4%)	(27.5%)	(11.7%)
Modified A star with RLWR	761.62	8.65	2.83
	(0.2%)	(-)	(-)

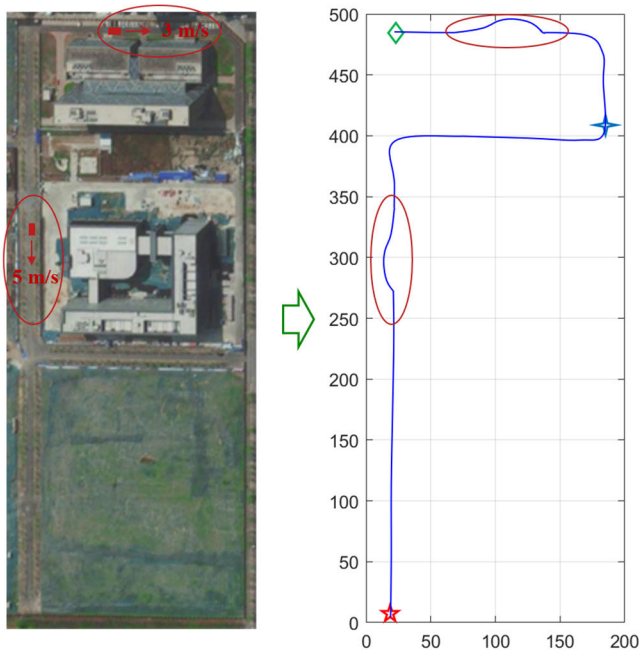


Fig. 20. The scenario with external moving vehicles and the local paths replanned.

As mentioned above, two external moving vehicles are regarded as the uncertainties and sudden obstacles for the local motion planning problem. As shown in the left part of Fig. 20, when the autonomous vehicle reaches a certain area, these two external vehicles start moving at speeds of 5 m/s and 3 m/s, respectively. The local paths replanned by the GHP-APF method are shown in the right part of Fig. 20. Obviously, the autonomous vehicle replans the real-time local path successfully to avoid the moving vehicles. Benefitting from the advantage of rolling update of the hierarchical framework, the proposed method can solve the replanning problem in the dynamic changing environment, while the Dijkstra and ILPSO algorithms are unable to deal with such situation.

In terms of the tracking control validation, the adaptability of the tracking controller to the changing scenario is also evaluated. The trajectory from the start point to the temporary target point is named trajectory #1, and the trajectory from the temporary target point to the final target point is named trajectory #2. The initial control agent of PQL-FPC is trained using only samples from trajectory #1. Then, the trained agent is applied directly to the tracking control of trajectory #1 and trajectory #2 in combination with the forward predictive control. Therefore, the performance of the generated tracking control strategy in trajectory #2 is the focus of the validation.

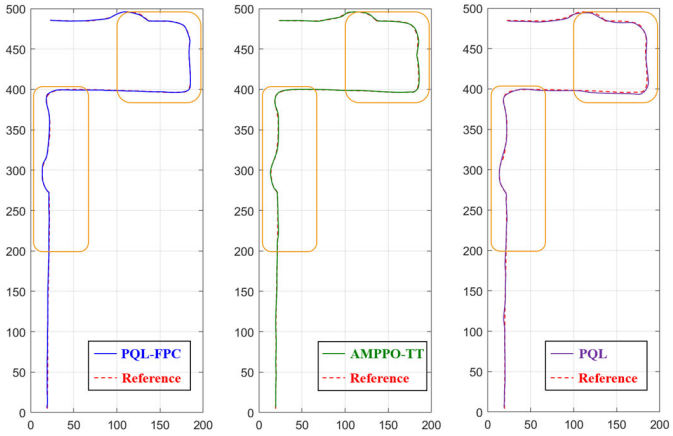


Fig. 21. The tracking results of different control methods in real-world scenario.

The tracking results of different control methods are illustrated in Fig. 21. It can be observed that the tracking accuracy of the PQL-FPC and AMPPO-TT is much higher than that of the PQL method. Specially, the PQL-FPC based controller still achieves high-accuracy tracking in trajectory #2, even though it is not trained by the samples from trajectory #2. Therefore, the proposed tracking control method can adapt to the changing scenario.

Besides, the total tracking error, average driving velocity and mean steering angular speed of different control methods are shown in Table XIII. It is obvious that the total tracking error of the proposed PQL-FPC method is less than that of AMPPO-TT method, which proves its higher tracking accuracy. The average driving velocity of the proposed method is the fastest among the three methods, which indicates that it performs well in tracking rapidity. The mean steering angular speed of the proposed method is also the least, which means less steering fluctuation and better driving comfort. Therefore, the proposed control method achieves better performance than the AMPPO-TT in tracking accuracy, tracking rapidity and tracking comfort. Above all, the effectiveness of the proposed framework has been validated in the real-world scenario.

V. CONCLUSION AND FUTURE RESEARCH

In this paper, an efficient hierarchical framework containing motion planning and tracking control for the autonomous vehicles is proposed. To generate the real-time optimal motion sequence, the global heuristic planning based artificial potential field method is presented in the upper layer of the framework. Subsequently, the prioritized Q-learning based forward predictive control method is proposed in the lower

TABLE XIII
THE RESULTS OF TRACKING CONTROL FOR DIFFERENT CONTROL METHODS

Algorithms	Total Tracking Error (m)	Average Driving Velocity (m/s)	Mean Angular Speed (°/s)
PQL-FPC	37.52	8.92	3.60
AMPPO-TT	39.64	7.65	4.26
PQL	91.15	7.31	4.30

layer of the framework to further improve the tracking control performance. Based on the results of the numerical simulation, virtual driving environment and real-world scenario, it can be concluded that:

1) The hierarchical framework can realize the cooperative operation of motion planning layer and tracking control layer, and support the real-time application of these two layers.

2) The global heuristic planning based artificial potential field method performs better than the previous methods in length optimality, planning rapidity and path smoothness.

3) The prioritized Q-learning based forward predictive control method achieves superior effects in tracking accuracy, tracking rapidity and driving comfort.

4) The adaptability of the proposed framework is also verified by applying different driving scenarios.

In the future research, we plan to carry out the study of the proposed hierarchical framework based on a more detailed dynamic model, and apply it to a series of real vehicle tests.

APPENDIX A
ABBREVIATIONS

Abbreviations	Full Names
GHP-APF	global heuristic planning based artificial potential field method
PQL-FPC	prioritized Q-learning based forward predictive control method
ITS	intelligent transportation system
A*	A-Star
D*	D-Star
GA	genetic algorithm
PSO	particle swarm optimization
RSO	recurrent spline optimizatio
RRT	rapidly-exploring random tree
PRM	probabilistic roadmap method
APF	artificial potential field
PP	pure pursuit
LQR	linear quadratic regulator
MPC	model predictive control
NMPC	nonlinear model predictive control
AI	artificial intelligencev
RL	reinforcement learning
MDP	Markov decision process
RLWR	robust locally weighted regression
PRL	prioritized reinforcement learning
LWR	locally weighted regression
PQL	prioritized Q-learning
QM	Q matrix
PER	prioritized experience replay
TD	temporal difference
QL	Q-learning
FPC	forward predictive control
GPS	global positioning system
SLAM	simultaneous localization and mapping
ILPSO	improved localized particle swarm optimization
AMPPO-TT	asynchronous multithreading proximal policy optimization-based trajectory tracking

APPENDIX B
THE PSEUDO-CODE OF MODIFIED A* ALGORITHM

Algorithm 1 : Modified A-Star With Variable Step

1. Initialize openlist (empty set), closelist (empty set), search radius R_{max} and R_{min}
2. Push start point into openlist and set the highest priority
3. **if** openlist is not empty
4. select the current node with highest priority from openlist
5. **if** current node = target point
6. progressively trace the parent nodes from the target point to the start point
7. get the global path from the start point to target point
8. **return** path
9. **else**
10. pop the current node from openlist and push this node into closelist
11. get the nearest distance from current node to obstacles d
12. **if** $d > R_{max}$
13. set search step S_{step}= $Step_{max}$
14. **elseif** $d < R_{min}$
15. set search step S_{step}= $Step_{min}$
16. **else**
17. S_{step}= $Step_{min}$+ round$((d-R_{min})/(R_{max}-R_{min})\times(Step_{max}-Step_{min}))$
18. **end**
19. **for** neighbor nodes of current node with search step S_{step}:
20. cost of neighbor node: $cost(neighbor) = g(current) + cost(current\text{to } neighbor)$
21. **if** neighbor node is in closelist
22. skip and select the next neighbor node
23. **elseif** neighbor node is in openlist & $cost(neighbor) \geq g(neighbor)$
24. skip and select the next neighbor node
25. **elseif** neighbor node is in openlist & $cost(neighbor) < g(neighbor)$
26. $g(neighbor) = cost(neighbor)$, $f(neighbor) = g(neighbor) + h(neighbor)$
27. $neighbor$. parent = $current$
28. **else**
29. $neighbor$. parent = $current$
30. $g(neighbor) = cost(neighbor)$, $f(neighbor) = g(neighbor) + h(neighbor)$
31. push the neighbor node into openlist
32. **end**
33. **end**
34. **end**
35. **end**

ACKNOWLEDGMENT

The authors would like to thank them for their support and help.

REFERENCES

[1] M. Veres and M. Moussa, "Deep learning for intelligent transportation systems: A survey of emerging trends," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3152–3168, Jul. 2020.

[2] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 315–329, Feb. 2020.

[3] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Nov. 2016.

[4] Y. Liang, Y. Li, Y. Yu, Z. Zhang, L. Zheng, and Y. Ren, "Path-following control of autonomous vehicles considering coupling effects and multi-source system uncertainties," *Automot. Innov.*, vol. 4, pp. 284–300, Aug. 2021.

[5] E. Siampis, E. Velenis, S. Gariuolo, and S. Longo, "A real-time nonlinear model predictive control strategy for stabilization of an electric vehicle at the limits of handling," *IEEE Trans. Control Syst. Technol.*, vol. 26, no. 6, pp. 1982–1994, Oct. 2018.

[6] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826–1848, May 2020.

[7] B. Patz, Y. Papelis, R. Pillat, G. Stein, and D. Harper, "A practical approach to robotic design for the DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 8, pp. 528–566, Aug. 2008.

[8] M. Luo, X. Hou, and J. Yang, "Surface optimal path planning using an extended Dijkstra algorithm," *IEEE Access*, vol. 8, pp. 147827–147838, 2020.

[9] E. Shang, B. Dai, Y. Nie, Q. Zhu, L. Xiao, and D. Zhao, "A guide-line and key-point based A-Star path planning algorithm for autonomous land vehicles," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, Art. no. 20303282.

[10] E. Shang, B. Dai, Y. Nie, Q. Zhu, L. Xiao, and D. Zhao, "An improved A-Star based path planning algorithm for autonomous land vehicles," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 5, pp. 1–13, 2020.

[11] S. Kim, H. Jin, M. Seo, and D. Har, "Optimal path planning of automated guided vehicle using Dijkstra algorithm under dynamic conditions," in *Proc. 7th Int. Conf. Robot Intell. Technol. Appl. (RiTA)*, 2019, pp. 231–236.

[12] C. Lamini, S. Benhlima, and A. Elbekri, "Genetic algorithm based approach for autonomous mobile robot path planning," *Proc. Comput. Sci.*, vol. 127, pp. 180–189, Jan. 2018.

[13] L. Zhang, Y. Zhang, and Y. Li, "Mobile robot path planning based on improved localized particle swarm optimization," *IEEE Sensors J.*, vol. 21, no. 5, pp. 6962–6972, Mar. 2021.

[14] W. Xu, Q. Wang, and J. M. Dolan, "Autonomous vehicle motion planning via recurrent spline optimization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 7730–7736.

[15] C. Yuan, G. Liu, W. Zhang, and X. Pan, "An efficient RRT cache method in dynamic environments for path planning," *Robot. Auto. Syst.*, vol. 131, Jan. 2020, Art. no. 103595.

[16] K. Cao, Q. Cheng, S. Gao, Y. Chen, and C. Chen, "Improved PRM for path planning in narrow passages," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 45–50.

[17] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Robot. Res.*, vol. 30, no. 7, pp. 846–894, Jun. 2011.

[18] C. Sun, Q. Li, B. Li, and L. Li, "A successive linearization in feasible set algorithm for vehicle motion planning in unstructured and low-speed scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 4, pp. 3724–3736, Jan. 2022.

[19] J. Ren and X. Huang, "Potential fields guided deep reinforcement learning for optimal path planning in a warehouse," in *Proc. IEEE 7th Int. Conf. Control Sci. Syst. Eng. (ICCSSE)*, Jul. 2021, pp. 257–261.

[20] F. Bounini, D. Gingras, H. Pollart, and D. Gruyer, "Modified artificial potential field method for online path planning applications," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2017, pp. 180–185.

[21] M. Elbanhawi, M. Simic, and R. Jazar, "Receding horizon lateral vehicle control for pure pursuit path tracking," *J. Vibrat. Control*, vol. 24, no. 3, pp. 619–642, Feb. 2018.

[22] Y. Huang, Z. Tian, Q. Jiang, and J. Xu, "Path tracking based on improved pure pursuit model and PID," in *Proc. IEEE 2nd Int. Conf. Civil Aviation Saf. Inf. Technol. (ICCASIT*, Oct. 2020, pp. 359–364.

[23] Y. Pan, X. Li, and H. Yu, "Efficient PID tracking control of robotic manipulators driven by compliant actuators," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 2, 915-922, 2019.

[24] G. Han, W. Fu, W. Wang, and Z. Wu, "The lateral tracking control for the intelligent vehicle based on adaptive PID neural network," *Sensors*, vol. 17, no. 6, p. 1244, 2017.

[25] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 1, pp. 48–58, Feb. 2020.

[26] Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020.

[27] A. Nguyen, J. Rath, T. Guerra, R. Palhares, and H. Zhang, "Robust set-invariance based fuzzy output tracking control for vehicle autonomous driving under uncertain lateral forces and steering constraints," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5849–5860, Jul. 2021.

[28] A. Feher, S. Aradi, and T. Becsi, "Hierarchical evasive path planning using reinforcement learning and model predictive control," *IEEE Access*, vol. 8, pp. 187470–187482, 2020.

[29] F. Gao, Y. Han, and D. Dang, "Balancing accuracy and efficiency: Fast motion planning based on nonlinear model predictive control," in *Proc. 5th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2021, pp. 3074–3089.

[30] V. Laurense and J. Gerdes, "Long-horizon vehicle motion planning and control through serially cascaded model complexity," *IEEE Trans. Control Syst. Technol.*, vol. 30, no. 1, pp. 166–179, Mar. 2022.

[31] S. Aradi, "Survey of deep reinforcement learning for motion planning of autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 740–759, Sep. 2022.

[32] Y. Shan, B. Zheng, L. Chen, L. Chen, and D. Chen, "A reinforcement learning-based adaptive path tracking approach for autonomous driving," *IEEE Trans. Veh. Technol.*, vol. 69, no. 10, pp. 10581–10595, Aug. 2020.

[33] L. Ding, S. Li, H. Gao, C. Chen, and Z. Deng, "Adaptive partial reinforcement learning neural network-based tracking control for wheeled mobile robotic systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 7, pp. 2512–2523, May 2020.

[34] X. Wang, H. Krasowski, and M. Althoff, "CommonRoad-RL: A configurable reinforcement learning environment for motion planning of autonomous vehicles," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 466–472.

[35] V. Behzadan and A. Munir, "Adversarial reinforcement learning framework for benchmarking collision avoidance mechanisms in autonomous vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 2, pp. 236–241, Summer 2021.

[36] L. Zhang, R. Zhang, T. Wu, R. Weng, M. Han, and Y. Zhao, "Safe reinforcement learning with stability guarantee for motion planning of autonomous vehicles," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 12, pp. 5435–5444, Jul. 2021.

[37] F. Ye, S. Zhang, P. Wang, and C.-Y. Chan, "A survey of deep reinforcement learning algorithms for motion planning and control of autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jul. 2021, pp. 1073–1080.

[38] Y. Lu, X. Xu, X. Zhang, L. Qian, and X. Zhou, "Hierarchical reinforcement learning for autonomous decision making and motion planning of intelligent vehicles," *IEEE Access*, vol. 8, pp. 209776–209789, 2020.

[39] M. Chen et al., "FaSTrack: A modular framework for real-time motion planning and guaranteed safe tracking," *IEEE Trans. Autom. Control*, vol. 66, no. 12, pp. 5861–5876, Feb. 2021.

[40] Z. He, L. Dong, C. Sun, and J. Wang, "Asynchronous multithreading reinforcement-learning-based path planning and tracking for unmanned underwater vehicle," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 52, no. 5, pp. 2757–2769, May 2022, doi: 10.1109/TSMC.2021.3050960.

[41] G. Du, Y. Zou, X. Zhang, L. Guo, and N. Guo, "Energy management for a hybrid electric vehicle based on prioritized deep reinforcement learning framework," *Energy*, vol. 241, Feb. 2022, Art. no. 122523.

[42] G. Li et al., "Deep reinforcement learning enabled decision-making for autonomous driving at intersections," *Automot. Innov.*, vol. 3, pp. 374–385, Dec. 2020.

[43] B. Peng et al., "End-to-end autonomous driving through dueling double deep Q-network," *Automot. Innov.*, vol. 4, pp. 328–337, Aug. 2021.

[44] T. Toledo, H. N. Koutsopoulos, and K. I. Ahmed, "Estimation of vehicle trajectories with locally weighted regression," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 1999, no. 1, pp. 161–169, Jan. 2007.

[45] Y. Zhou, D. Tang, H. Zhou, X. Xiang, and T. Hu, "Vision-based online localization and trajectory smoothing for fixed-wing UAV tracking a moving target," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–8.

[46] N. Raju, P. Kumar, C. Reddy, S. Arkatkar, and G. Joshi, "Examining smoothening techniques for developing vehicular trajectory data under heterogeneous conditions," *J. Eastern Asia Soc. Transp. Stud.*, vol. 12, pp. 1549–1568, Dec. 2017.

[47] A. Arab, *Safe Motion Control and Planning for Autonomous Racing Vehicles*. New Jersey, NJ, USA: Rutgers The State University of New Jersey, 2021.

[48] A. Arab and J. Yi, "Safety-guaranteed learning-predictive control for aggressive autonomous vehicle maneuvers," in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Jul. 2020, pp. 1036–1041.

[49] A. Hakobyan, G. Kim, and I. Yang, "Risk-aware motion planning and control using CVaR-constrained optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3924–3931, Jul. 2019.

[50] Z. Hu, Z. Yang, J. Huang, and Z. Zhong, "Safety guaranteed longitudinal motion control for connected and autonomous vehicles in a lane-changing scenario," *IET Intell. Transp. Syst.*, vol. 15, no. 2, pp. 344–358, 2021.

[51] I. Noreen, A. Khan, and Z. Habib, "Optimal path planning using RRT* based approaches: A survey and future directions," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 11, pp. 97–107, 2016.

[52] A. Arab, K. Yu, J. Yi, and D. Song, "Motion planning for aggressive autonomous vehicle maneuvers," in *Proc. IEEE Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2016, pp. 221–226.

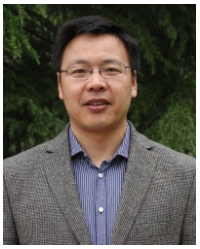
[53] I. Noreen, A. Khan, H. Ryu, N. Doh, and Z. Habib, "Optimal path planning in cluttered environment using RRT*-AB," *Intell. Service Robot.*, vol. 11, pp. 41–52, Jan. 2017.

[54] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. Chen, "Multiobjective optimal parking maneuver planning of autonomous wheeled vehicles," *IEEE Trans. Ind. Electron.*, vol. 67, no. 12, pp. 10809–10821, Jan. 2020.

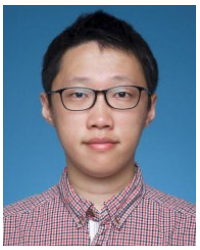
[55] R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. Chen, "Design and implementation of deep neural network-based control for automatic parking maneuver process," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 4, pp. 1400–1413, Dec. 2022.

[56] R. Chai, D. Liu, T. Liu, A. Tsourdos, Y. Xia, and S. Chai, "Deep learning-based trajectory planning and control for autonomous ground vehicle parking maneuver," *IEEE Trans. Automat. Sci. Eng.*, early access, Jun. 23, 2022, doi: 10.1109/TASE.2022.3183610.

[57] R. Chai, A. Tsourdos, S. Chai, Y. Xia, A. Savvaris, and C. Chen, "Multiphase overtaking maneuver planning for autonomous ground vehicles via a desensitized trajectory optimization approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 74–87, Apr. 2023.



Yuan Zou (Senior Member, IEEE) received the Ph.D. degree from the Beijing Institute of Technology in 2005. He is currently a Professor with the Beijing Collaborative and Innovative Center for Electric Vehicles and the School of Mechanical Engineering, Beijing Institute of Technology. He is also working as the Co-Director of the ETHZ-BIT Joint Research Center for New Energy Vehicle Dynamic System and Control. He conducted research about ground vehicle propulsion modeling and optimal control with the University of Michigan, Ann Arbor, and ETH Zurich. His research interests include modeling and control for electrified vehicle and transportation systems.



Xudong Zhang (Member, IEEE) received the M.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2011, and the Ph.D. degree in mechanical engineering from the Technical University of Berlin, Berlin, Germany, in 2017. Since 2017, he has been an Associate Professor with the Beijing Institute of Technology. His main research interests include distributed drive electric vehicles, vehicle dynamics control, vehicle state estimation, torque allocation, and the power management of hybrid electric vehicles.

Zirui Li received the B.S. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering under the supervision of Prof. Jianwei Gong. From June 2021 to July 2022, he was a Visiting Researcher with the Delft University of Technology (TU Delft) with CSC funding from China. Since August 2022, he has been a Visiting Researcher with the Chair of Traffic Process Automation, Faculty of Transportation and Traffic Sciences "Friedrich List," TU Dresden. His research focuses on interactive behavior modeling, risk assessment, and motion planning of automated vehicles.

Guodong Du received the B.S. degree in mechanical engineering from the Beijing Institute of Technology, Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in automobile engineering. He is also working as an Academic Guest with ETH Zurich. His research interests include motion planning and control, reinforcement learning algorithm, vehicle dynamics control, and energy management of hybrid electric vehicles.

Qi Liu (Student Member, IEEE) received the B.S. degree from the Beijing Institute of Technology (BIT), Beijing, China, in 2019, where he is currently pursuing the Ph.D. degree in mechanical engineering. His research interests include intelligent vehicles, environmental perception, and decision making.