IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 30 July 2023, accepted 11 August 2023, date of publication 17 August 2023, date of current version 24 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3306239

RESEARCH ARTICLE

A Two-Layer Model Predictive Path-Tracking Control With Curvature Adaptive Method for High-Speed Autonomous Driving

HEBING LIU, JINHONG SUN, AND KA WAI ERIC CHENG, (Fellow, IEEE)
Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Ka Wai Eric Cheng (eric-cheng.cheng@polyu.edu.hk)

This work was supported in part by the Innovation and Technology Fund, Hong Kong, under Grant ITP/047/21AP; and in part by the Power Electronics Research Centre, Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University.

ABSTRACT A two-layer model predictive control (MPC) algorithm with curvature adaptive is introduced and adopted in path tracking, especially for high-speed autonomous driving. Whether the vehicle can stably reach a safe driving speed in advance is the main consideration of rollover and speed-overshoot avoidance, especially when the trajectory with large curvatures. Thus, the outer layer of the proposed controller, which is built based on the vehicle kinematic model, generates an optimal vehicle velocity. And, the inner layer controller that is established according to the vehicle dynamics provides an optimal front wheel angle obtained combined with the optimal tracking trajectory generated by the path planner. The cross-track error, which is considered an important judging criterion of tracking and obstacle avoidance, is chosen here to validate the control performance. With the MATLAB/Simulink-Carmaker platform for modeling, the two-layer MPC has good performance in a continuous curve, simple obstacle avoidance, and complex obstacle avoidance scenarios. Notably, when the average driving speed reaches $108km/h$, the cross-track error can be controlled within $0.21m$ when employing the proposed MPC method. In contrast, the conventional MPC method yields a cross-track error exceeding $3.4m$, while an LQR-based strategy results in a maximum error of $0.64m$. The proposed two-layer MPC algorithm with curvature adaptivity significantly enhances path-tracking performance, particularly for high-speed autonomous driving. By effectively controlling the cross-track error and considering various driving scenarios, it successfully ensures safe and stable driving speeds, thereby mitigating the risks associated with rollover and speed overshoot.

INDEX TERMS Path tracking, two-layer MPC, curvature adaptive, path planning, high-speed, energy-saving, obstacle avoidance.

I. INTRODUCTION

Autonomous driving, which technologies are classified into three categories: environment awareness, decision planning, and path tracking, plays an important role in reducing traffic congestion and improving driving safety. Among them, path tracking is the most closely integrated with the characteristics of the vehicle itself and is crucial to the final execution. At present, algorithms that are commonly used in path tracking are proportional-integral-derivative (PID) control, pure tracking control, linear quadratic regulator (LQR) control, pre-sight control, MPC, etc. Numerous scholars used PID controllers to achieve trajectory-tracking control of intelligent vehicles. Zhao et al [1] designed an adaptive-PID controller for vehicle trajectory tracking and had better performance even with $72km/h$, however, its pertinence was relatively strong as only suitable for the s trajectory tracking. As one of the pure path tracking control, the Pure-Pursuit (PP) method is also widely used, especially in the field of mobile robotics and vehicle control. To improve the tracking accuracy, Wang et al [2] used the improved pure tracking method with the selection of optimal look-ahead distance to

The associate editor coordinating the review of this manuscript and approving it for publication was Zheng Yan.

solve the lateral or steering angle control problem caused by the Salp swarm-based control algorithm in autonomous driving, while this method only applied in lower driving speed. Horváth et al. [3] proposed a method that through selecting multiple-target to handle the dynamic change of the look-ahead distance and then overcome the weakness of the PP strategy. By using the geometric relationship between the vehicle and the path, the look-ahead distance could be chosen far away [4], and this method largely decreased the cross-track error. Based on the above works about the PP-based control method, it can be concluded that the pure tracking control method guarantees the vehicle tracking on the reference trajectory when driving at low speed, but it fails at high speed, especially with high curvature. Dealing with the high-speed and high-curvature situation, a ground lateral tracking control strategy that combined the PP and the Stanley tracking methods was put forward by Cibooglu et al. [5]. It had better cutting-corner behavior when the driving speed increased, nevertheless, the control performance became less advantageous when the speed reached 80 km/h. Numerical training that is used in machines can be used to model vehicle dynamics [6]. To improve its high-speed and high-curvature adaptability, an improved PP control that involved the deep learning method of road detection was utilized by Seo et al. [7]. Nevertheless, this method only improved the effectiveness of tracking and only had high stability with the HD map provided in advance. As LQR contains the state feedback ability, it can linearize a complex nonlinear model and provide an optimal control value, and can be used to deal with the constrained issues in autonomous driving. A constrained iterative LQR algorithm was proposed by Chen [8] to efficiently deal with the issues of the nonlinear pathing tracking system and non-convex constraints. However, the effectiveness of this LQR method largely depended on the definition of the initial trajectory. Therefore, the MPC algorithm based on the prediction information and constraints becomes widely utilized due to its ability to offer the optimal control sequence. Wang et al. [9] achieved lateral control of the vehicle at high speed with good results by using an MPC strategy. Besides, with the fuzzy control algorithm added, the proposed MPC method could adaptively adjust the weight of the cost function. To improve the controllability and stability of the limitation of the vehicle dynamics, especially for the nonlinearity of the tire model, an MPC method with a prediction horizon was proposed by Li et al. [10], which had a good nonlinear control effect and could reduce the amount of calculation.

In the process of autonomous driving, the path tracking and lateral tracking error of MPC on normal roads is very small, however, there also have some issues during the control of the conventional MPC controller. The lateral tracking error of traditional MPC becomes larger when facing complex road conditions and curvature changes in the tracking path, which results in the risk of collision with surrounding obstacles. Road curvature has an important effect on the steering characteristics and formal stability of an unmanned vehicle for path tracking, which is directly related to the accuracy of the dynamic model. Normally, the vehicle can be passed directly at high speed when the curvature is small. However, considering safety and comfort driving when the curvature becomes larger, especially at a high speed, the vehicle needs to slow down in advance to reduce the out-of-control situation caused by insufficient lateral support. To better deal with the disturbances caused by the uncertainties of vehicle model parameters, modeling errors, etc, and increase the control accuracy of path tracking, some other control methods are considered and blended into the MPC strategy. In [11], a gain scheduling path-tracking controller with MPC and H_∞ was built in the LPV system, which could well suppress parameter uncertainty, modeling error, and external interference. Although it had enough robustness to resist the influence of speed-varying and tire stiffness changing, the performance of velocity over 80 km/h was not considered. An improved MPC and hybrid PID control theory was put forward to increase the vehicle's driving stability by Peicheng et al. [12], in which the constraints on the side deviation of the front wheel were added to the traditional MPC, and the relaxation factor was added. However, the test analysis only contained low-speed scenarios, and the effect of the road on the controller was not considered, and the parameters of the PID control should be adjusted depending on the real situations. To ensure path-tracking accuracy and enhance control stability at the same time, the MPC upper controller was established by Zhai et al. [13] after the kinematic preview model was built, which could track the lateral deviation, course angle deviation, sideslip angle, and yaw angle velocity. The MPC with adaptive preview characteristics was proposed in [14] based on lateral error and target curvature, and its control performance was supplemented by longitudinal vehicle speed auxiliary constraints. Whereas the lateral error was larger than 0.5m when the driving speed equaled 30m/s, and the influence of road curvature on the controller was not involved. A motion MPC method based on yaw velocity was proposed to solve the disturbance of upcoming road curvature at different speeds by Tang et al. [15]. However, this method lack of controller performance analysis under the large curvature, and the slip angle compensator should be manually adjusted according to the actual performance.

All in all, two essential considerations should be involved when designing the improved MPC controller for autonomous vehicle driving:

1). The path curvature, which also increases the risk of collision and affects the driving speed, should be considered during the MPC controller design.

2). Computing resources are very precious in the real-time computing system, and the traditional MPC has a fixed prediction time domain. When the prediction time domain is relatively small, a large control deviation may occur which is mainly due to the situation that the road information cannot be taken into account in advance.

Combined with the above considerations, the two-layer MPC with curvature adaptive strategy is given to deal with high-speed autonomous driving. The vehicle under the control of this MPC controller can slow down in advance during bending and accelerate in time after completing the bending, which means that it can maximize tracking efficiency and reduce vehicle energy consumption under the condition of ensuring safety, and the lateral tracking error is ensured within a set range during the whole control. The main novelty of this work is summarized below:

1). The upper-level path planning considers vehicle dynamics, which can avoid all obstacles and largely improve driving safety by constraining and minimizing the curvature of the generated path.

2). The outer MPC makes trajectory predictions based on vehicle kinematics. When the predicted trajectory is quite different from the actual trajectory (e.g. entering a curve), it decelerates in advance to reduce the tracking error caused by inaccurate modeling caused by high speed. After completing the corner, accelerate in advance to improve the tracking efficiency while ensuring the tracking error. Besides, an acceleration change constraint is added to limit the amount of speed change, which avoids the energy loss caused by rapid acceleration or deceleration.

3). Dynamically adjusting the prediction time domain in the MPC design according to the speed and road curvature has been put forward, which keeps the lateral deviation within a reasonable range and reduces the amount of calculation.

4). Finally, through using the co-simulation MATLAB /Simulink-CarMaker software, the Hardware-in-The-Loop (HIL) simulation experiment is carried out to verify the control performance of the proposed two-level MPC controller.

The rest of this paper is arranged as follows: the mathematical model of the vehicle is described in Section II, which is the base of MPC establishment. Details of the proposed two-layer MPC method are given in Section III. After modeling in the co-simulation platform, the results shown in Section IV verify that the proposed control method has good control performance.

II. MATHEMATICAL MODELS OF THE VEHICLE

A two-layer MPC controller is described in Fig.1, in which control values are obtained based on the kinematic and dynamic vehicle models, and limitations of the data-driven or experience control can be omitted due to no need for tuning or training parameters. In other words, the establishment of the vehicle model described in this section is the base of controller design, and its parameter definitions are given in Tab. 1.

A. KINEMATIC MODEL

The vehicle kinematic model [16], which contains changes in the position and velocity of objects in space over time, can be used in the path planning application and make the planned path feasible during driving. As the model constraints need to express the real physical limitations of the control object, thus how to simplify the vehicle model reasonably is one of the issues to be considered in MPC-based trajectory tracking design. A bicycle model shown in Figure.2 is well-established and widely used in the kinematics model analysis and is also used in the MPC design here [17].

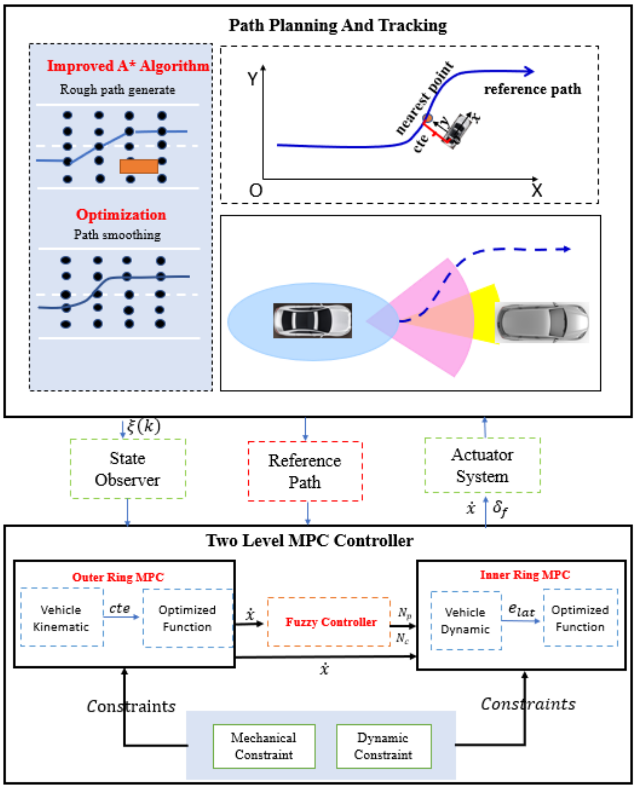


FIGURE 1. The proposed control strategy.

TABLE 1. Definitions of the parameters.

Symbol	Definition
δ_f	Front-wheel steering angle
l_a/l_b	Distance from center of mass to front /rear axle.
α_f/α_r	Front / rear wheel sideslip angle
$\dot{x}$	Vehicle longitudinal speed (in xoy)
$\dot{y}$	Vehicle lateral speed (in xoy)
v_f/v_r	Vehicle front/rear axle center speed
m	Vehicle weight
$\varphi(\dot{\varphi})$	Yaw angle (Yaw rate)
ω	Angular speed
I_z	Yaw moment of inertia
N_p/N_c	Time/Control domain for MPC controller
CAD	Curvature adaptive

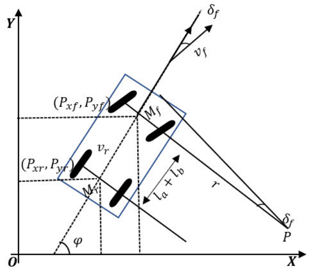


FIGURE 2. The bicycle model.

Assuming that the vehicle moves in a straight line or a circle around at any time, the steering kinematics model can be obtained. (P_{xf}, P_{yf}) and (P_{xr}, P_{yr}) are the coordinates in the inertial coordinate system XOY of the front and rear axles of the vehicle, respectively. r is the steering radius of the rear wheel, M_f/M_r is the center of the vehicle's front /rear axle, and P is the instantaneous rotation center of the vehicle. δ_f remains unchanged during the steering process, that is, the instantaneous turning radius of the vehicle is the same as the road curvature radius. The kinematics equation is given in (1), and r and δ_f can be obtained by equation (2).

$$\begin{cases} \dot{x} = v_r * cos\varphi \\ \dot{y} = v_r * sin\varphi \\ \dot{\varphi} = v_r * \dfrac{tan\delta_f}{l_a + l_b} \end{cases} \tag{1}$$

$$\begin{cases} r = v_r/\omega \\ \delta_f = tan^{-1}((l_a + l_b)/r) \end{cases} \tag{2}$$

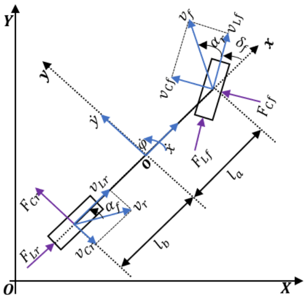


FIGURE 3. **Vehicle dynamic model.**

B. DYNAMIC MODEL

The dynamic model [18], which describes the vehicle state, is very important for the MPC controller design, especially for the path-tracking control. Some idealized hypotheses are considered during the model establishment, such as, the vehicle is driving on a flat road; the suspension system and the vehicle are rigid; the vertical and horizontal coupling relation of the tire is ignored; the monorail model is used to describe the vehicle motion without considering the transfer of load from left to right; the vehicle speed changes slowly and the load transfer of front and rear axles are ignored. Applying Newton's Second Law to longitudinal, lateral, and yaw degrees of freedom, the vehicle dynamic model in the local body-fixed coordinate system xoy shown in Fig.3 can be constructed,

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\varphi} + 2F_{xf} + 2F_{xr} \\ m\ddot{y} = -m\dot{x}\dot{\varphi} + 2F_{yf} + 2F_{yr} \\ I_z\ddot{\varphi} = 2l_aF_{yf} - 2l_bF_{yr} \end{cases} \tag{3}$$

where, F_{xf}/F_{xr} and F_{yf}/F_{yr} are the total force of the front and rear tires in the x and y directions.

$$\begin{cases} F_{xf} = F_{Lf}cos\delta_f - F_{Cf}sin\delta_f \\ F_{xr} = F_{Lr}cos\delta_r - F_{Cr}sin\delta_r \\ F_{yf} = F_{Lf}sin\delta_f + F_{Cf}cos\delta_f \\ F_{yr} = F_{Lr}sin\delta_r + F_{Cr}cos\delta_r \end{cases} \tag{4}$$

Combined equations (3) and (4), the mathematical dynamic model can be obtained in (5),

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\varphi} + 2(F_{Lf}cos\delta_f - F_{Cf}sin\delta_f) + 2F_{Lr} \\ m\ddot{y} = -m\dot{x}\dot{\varphi} + 2(F_{Lf}sin\delta_f + F_{Cf}cos\delta_f) + 2F_{Cr} \\ I_z\ddot{\varphi} = 2l_a(F_{Lf}sin\delta_f + F_{Cf}cos\delta_f) - 2l_bF_{Cr} \end{cases} \tag{5}$$

Due to the nonlinear characteristics of the tire force, the Pacejka tire model [19], [20] is adopted to help estimate the vehicle velocity. When the cornering angle and slip ratio of the tire is small, the linearized and simplified formulas of longitudinal force F_{Lf}, F_{Lr} and lateral force F_{Cf}, F_{Cr} in the tire model are given in (6),

$$\begin{cases} F_{Lf} = C_{Lf}s_f \\ F_{Cf} = C_{Cf}\alpha_f \\ F_{Lr} = C_{Lr}s_r \\ F_{Cr} = C_{Cr}\alpha_r \end{cases} \tag{6}$$

where, C_{Lf}, C_{Lr} are longitudinal stiffness, C_{Cf}, C_{Cr} are lateral stiffness, s_f, s_r are indicate tire slip ratio, α_f, α_r are side slip ratio. During the calculation process of the tire force, the small angular velocity theory is used in that $cos\theta \approx 1$, $sin\theta \approx \theta$, $tan\theta \approx \theta$.

$$\begin{cases} v_{Cf} = (\dot{y} + l_a\dot{\varphi})cos\delta_f - \dot{x}sin\delta_f \\ v_{Lf} = (\dot{y} + l_a\dot{\varphi})sin\delta_f + \dot{x}cos\delta_f \\ v_{Cr} = l_b\dot{\varphi} - \dot{y} \\ v_{Lr} = \dot{x} \end{cases} \tag{7}$$

such,

$$\begin{cases} \alpha_f = \dfrac{\dot{y} + l_a\dot{\varphi}}{\dot{x}} - \delta_f \\ \alpha_r = \dfrac{l_b\dot{\varphi} - \dot{y}}{\dot{x}} \end{cases} \tag{8}$$

$$\begin{cases} m\ddot{x} = m\dot{y}\dot{\varphi} + 2\left[C_{Lf}s_f + C_{Cf}(\delta_f - \dfrac{\dot{y} + l_a\dot{\varphi}}{\dot{x}})\delta_f + C_{Lr}s_r\right] \\ m\ddot{y} = -m\dot{x}\dot{\varphi} + 2\left[C_{Lf}s_f\delta_f + C_{Cf}(\dfrac{\dot{y} + l_a\dot{\varphi}}{\dot{x}} - \delta_f) \right. \\ \qquad \left. + C_{Cr}\dfrac{l_b\dot{\varphi} - \dot{y}}{\dot{x}}\right] \\ I_z\ddot{\varphi} = 2l_a\left[C_{Lf}s_f\delta_f + C_{Cf}(\dfrac{\dot{y} + l_a\dot{\varphi}}{\dot{x}} - \delta_f)\right] \\ \qquad -2l_bC_{Cr}\dfrac{l_b\dot{\varphi} - \dot{y}}{\dot{x}} \end{cases} \tag{9}$$

Inserting (6)-(8) to (5), the vehicle dynamic model can be finally calculated in (9). Equation (10) describes the vehicle motion relationship between xoy and XOY. The nonlinear continuous equations (9) and (10) are considered the mathematical base to build the discrete model of the inner controller mentioned in Section III-D.

$$\begin{cases} \dot{X} = \dot{x}\ cos\ \varphi - \dot{y}\ sin\ \varphi \\ \dot{Y} = \dot{x}\ sin\ \varphi - \dot{y}\ cos\ \varphi \end{cases} \tag{10}$$

III. CONTROL ALGORITHM

A. DEFINITION OF THE PATH PLANNING

It is necessary to verify the performance of the proposed MPC controller, the driving environment should be built with some static and dynamic obstacles. The path planner, which is used to generate the optimal reference trajectory, is designed against these obstacles and with fully energy-saving considerations. A rough path generation algorithm is described in Tab 2.

TABLE 2. Rough path profile.

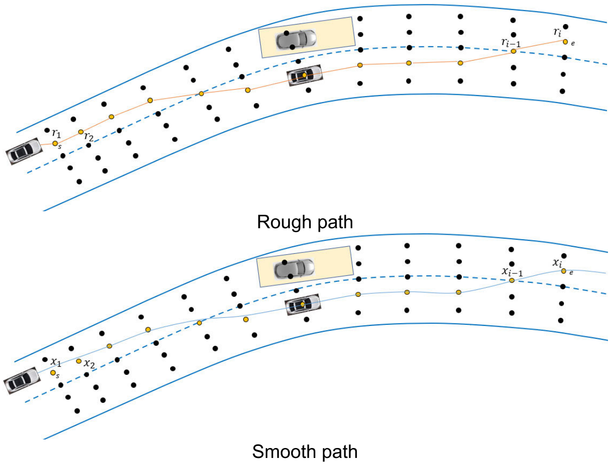
```
Algorithm 1 Rough path generation
Input:   $\dot{x}$: Current velocity of vehicles
         $l_{max}$: Max sampling length  $l_{min}$: Min sampling length
         $d_{safe}$: Safe distance between vehicle to obstacles
         $s_n$: Obstacle information
         $s$: Start vertex   $e$: End vertex $c$: Current vertex
Output:  $path$: Rough path with obstacle avoidance
         vertex←PointsSample($\dot{x}$, $l_{max,}$ $l_{min}$)
         vertex.obstacle_cost←CalculateObstacleCost(vertex,
                                    $d_{safe}, s_1, s_2, \cdots s_n$)
         $e$ ←GetEndPoint($\dot{x}, s_1, s_2, \cdots s_n$)
         $c = s$
         while $c \neq e$
             $v_{neighbor}$ ←GetNeighborVertexs($c$, vertexs)
             $g$ ←CalcurateNeighborCurrentCost($v_{neighbor}$)
             $h$ ←CalcurateNeighborHeuristicValue($v_{neighbor}$)
             $f$ ←UpdateNeighborFinalCost(
                                    $v_{neighbor}, g, h, obstacle\_cost$)
             $c$ ←UpdateCurrentVertexWithLoweastCost($v_{neighbor}$)
         endwhile
         build a path rooted at $s$
         return path
```

Based on the A^* algorithm [21], [22], the cost of each point can be calculated, then its final state is obtained by considering the obstacle situations. After connecting these points, a rough path ($r_0 \rightarrow r_1 \rightarrow r_2 \rightarrow \cdots \rightarrow r_i$) is generated in Fig.4(a). The kinematic model constraint is highly required to improve the driving conditions during the path planning process, as the vehicle cannot drive directly along the rough path. A smooth trajectory is generated by connecting these points $\{x_0, x_1, \ldots, x_i\}$ in a quintic polynomial way shown in Fig. 4(b).

And the cost function, which is used to constrain the vehicle trace profile, is put forward in (11).

$$J = \sum\nolimits_{i=0}^{n-1} (x_i - r_i)^2 + \sum\nolimits_{i=1}^{n-1} ((x_i - x_{i-1}) - (x_{i+1} - x_i))^2 + \sum\nolimits_{i=1}^{n} (x_i - x_{i-1})^2 \tag{11}$$

where, $\sum\nolimits_{i=0}^{n-1} (x_i - r_i)^2$ should be kept in a reasonably small range, the second item works for smoothing the path, and, the third one is the cost principle of energy consumption. The calculation time and load can be reduced by ignoring the consideration of the speed cost, as the speed control is done in the outer layer of the proposed MPC controller. Under the constraint of the cost function and with several iterations, the smooth trajectory ($x_0 \rightarrow x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_i$) can be finally updated and generated in Fig.4b.



Rough path

Smooth path

FIGURE 4. The generated path.

B. THE OUTER RING OF THE PROPOSED MPC

According to the kinematic model analysis mentioned in section II-A, the state update equation can be obtained,

$$\begin{cases} x_{t+1} = x_t + \dot{x}_t * cos(\varphi_t) * dt \\ y_{t+1} = y_t + \dot{x}_t * sin(\varphi_t) * dt \\ \varphi_{t+1} = \varphi_t + \frac{\dot{x}_t}{l_a} \delta_t * dt \\ \dot{x}_{t+1} = \dot{x}_t + a_t * dt \end{cases} \tag{12}$$

The error calculation is given in (13), in which the cross-track error cte shown in Fig.1 is the distance between the center of the road and the vehicle's position, and the orientation error $e\psi$ describes the distance between the desired orientation and the values that the vehicle's orientation plus the change in orientation caused by the movement. Besides, $f(x_t)$ is the cubic polynomial curve fitting equation of the reference trajectory.

$$\begin{cases} e\psi_{t+1} = \psi_t + \dfrac{\dot{x}_t}{l_a} \delta_t * dt - tan^{-1}(f'(x_t)) \\ cte_{t+1} = f(x_t) - y_t + \dot{x}_t * sin(e\psi_t) * dt \end{cases} \tag{13}$$

Equation (14) gives the cost function of the outer layer MPC controller, in which the energy consumption is also taken into account.

$$\begin{aligned} J = &\sum\nolimits_{t=1}^{N} (w_{cte}||cte_t||^2 + w_{e\varphi}||e\varphi_t||^2 + w_v||v_t - v_{target}||^2) \\ &+ \sum\nolimits_{t=1}^{N-1} (w_\delta||\varphi_t||^2 + w_a||a_t||^2) \\ &+ \sum\nolimits_{t=2}^{N-1} (w_{rate_\delta}||\delta_t - \delta_{t-1}||^2 + w_{rate_{\delta a}}||a_t - a_{t-1}||^2) \end{aligned} \tag{14}$$

During the path-tracking process, there have dynamic constraints which are considered as the limitation of acceleration $a \in [a_{min}, a_{max}]$. Moreover, the initial state of the vehicle is defined as $X = (x, y, \varphi, \dot{x}, cte, e\varphi)$. After that, the real-time vehicle speed can be calculated with the help of the first value generated by the IPOPT optimizer [23]: $\dot{x}_{opt} = \dot{x} + a_{opt} * dt$, which generating algorithm is given in Tab. 3:

TABLE 3. Rough path profile.

```
Algorithm 2 Optimal speed generation

Input:  $x, y$: Vehicle's position $dt$: update time interval
        $\varphi$: Vehicle's heading direction $v_{ref}$: Reference velocity
        $path$: Vehicle's reference path
        $C_{\delta_f}, C_a$: Constraint of $\delta_f$ and acceleration
Output: $\dot{x}_{opt}$: Optimal velocity
        $f$ ←PolynomialFitting($path$)
        for $i$ in range($1, N_p$)
            $cte$ ←CalculateCrossTrackError ($f, x$)
            $epsi$ ←CalculateOrientationError ($f', x$)
            $state_i$ ←StateUpdate ($state_{i-1}, cte, epsi$)
            $cost$ ← $cost$ + CaculateCost($state_i, cte, epsi$)
        endfor
        lower_bound, upper_bound←
                AddDynamicConstraint($C_{\delta_f}, C_a$)
        $a$ ←Optimization($state, cost$, lower_bound,
                upper_bound)
        $\dot{x}_{opt} = \dot{x} + a * dt$
        return $\dot{x}_{opt}$
```

C. ADAPTIVE FUZZY CONTROL

To strike a balance between enhancing the real-time computing performance of the main control unit and maintaining tracking accuracy, the addition of the fuzzy logic control (FLC) method [24], [25] is implemented. The FLC method is utilized to enhance the system's robustness performance while generating optimal N_p and N_c.

Four linguistic variables: *zo* (zero), *pb* (positive big), *pm* (positive middle), *ps* (positive small) are selected to form the fuzzy rules. Fig.5 describes the rule table and map of the proposed adaptive fuzzy controller, in which $\Delta\theta$ represents the error between the current heading and the reference angles.

D. THE INNER RING OF THE PROPOSED MPC

Define the state variable vector as $S = [\dot{x}, \dot{y}, \varphi, \dot{\varphi}, X, Y]^T$, the output variable as $O = [\varphi, Y]^T$, and the control variable as $u = \delta_f$, thus, the discrete equation can be yielded as (15), as shown at the bottom of the next page. where the system matrices A, B, and C are given below based on equations (9) and (10).

Then the linearization model can be yielded in (16) [26],

$$\begin{cases} S(k+1) = A_k S(k) + B_k u(k) \\ O(k+1) = CS(k) \end{cases} \tag{16}$$

Here, $A_k = I + TA$, $B_k = TB$ and T is the sampling time. Considering the mechanical limitation, $\Delta\delta_f$ should be taken into account in the control state. Therefore, a new state $\tilde{S}(k)$ is put forward combined with the current state variable $S(k)$ and the previous control variable $u(k-1)$. Thus, the updated state equation is shown in (17),

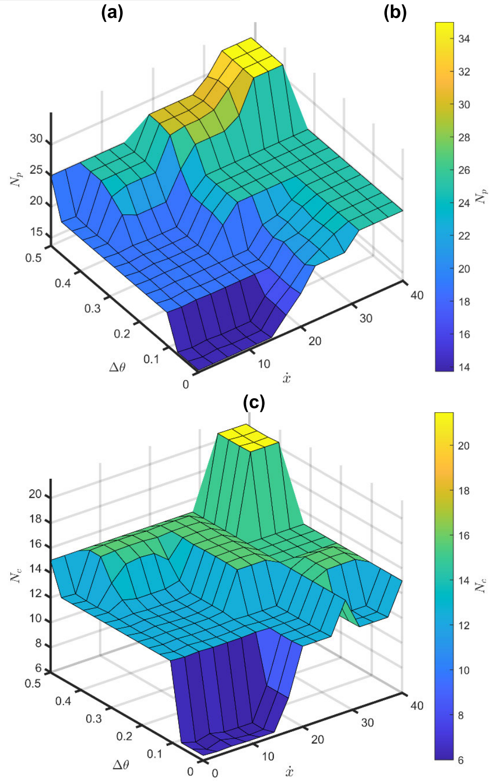
$$\begin{cases} \tilde{S}(k+1) = \tilde{A}_k \tilde{S}(k) + \tilde{B}_k \Delta u(k) \\ \tilde{O}(k) = \tilde{C}_k \tilde{S}(k) \end{cases} \tag{17}$$

(a)

N_p		$\dot{x}$			
		pb	*pm*	*ps*	*zo*
$\Delta\theta$	*pb*	*pb*	*pb*	*pm*	*pm*
	pm	*pm*	*pm*	*ps*	*ps*
	ps	*pm*	*ps*	*ps*	*ps*
	zo	*pm*	*ps*	*zo*	*zo*

(b)

N_c		$\dot{x}$			
		pb	*pm*	*ps*	*zo*
$\Delta\theta$	*pb*	*pb*	*pm*	*pm*	*ps*
	pm	*pb*	*pm*	*ps*	*ps*
	ps	*pm*	*ps*	*ps*	*ps*
	zo	*pm*	*ps*	*zo*	*zo*



(c)

(d)

FIGURE 5. Fuzzy rules: (a) table of N_p (b) table of N_c (c) map of N_p (d) map of N_c.

where,

$$\tilde{A}_k = \begin{bmatrix} A_k & B_k \\ 0_{m*n} & I_m \end{bmatrix} \tilde{B}_k = \begin{bmatrix} B_k \\ I_m \end{bmatrix} \tilde{C}_k = [C_k \; 0]$$

m and n are respectively expressed as the dimension of the state matrix and control matrix. For the prediction horizons, the state equations can be written as follow:

$$\begin{cases} \tilde{S}\left(k+1 \mid k\right) = \tilde{A}_k \tilde{S}(k) + \tilde{B}_k \Delta u(k) \\ \tilde{S}\left(k+2 \mid k\right) = \tilde{A}_{k+1} \tilde{A}_k \tilde{S}\left(k\right) + \tilde{A}_k \tilde{B}_k \Delta u\left(k\right) \\ + \tilde{B}_{k+1} \Delta u\left(k+1\right) \\ \vdots \\ \tilde{S}\left(k+N_c \mid k\right) = f\left(k, N_c-1, 0\right) \tilde{S}\left(k\right) + f\left(k, N_c-2, 0\right) \\ + \tilde{B}_k \Delta u\left(k\right) + \ldots + \tilde{B}_{k+N_c-1} \Delta u\left(k+N_c-1\right) \\ \vdots \\ \tilde{S}\left(k+N_p \mid k\right) = f\left(k, N_p-1, 0\right) \tilde{S}\left(k\right) + f\left(k, N_p-2, 0\right) \\ + \tilde{B}_k \Delta u\left(k\right) + \ldots + f\left(k, N_p-N_c, N_c\right) \\ + \tilde{B}_{k+N_c-1} \Delta u\left(k+N_c-1\right) \end{cases}$$

where, $f(k, N, j) = \prod_{i=j}^{N} \tilde{A}_{k+i}$. Then, the output variables can be expressed as,

$$\tilde{O}(k) = \begin{bmatrix} \tilde{O}(k+1\,|\,k) \\ \tilde{O}(k+2\,|\,k) \\ \vdots \\ \tilde{O}\left(k+N_p\,|\,k\right) \end{bmatrix} = \tilde{C}_k \begin{bmatrix} \tilde{S}(k+1\,|\,k) \\ \tilde{S}(k+2\,|\,k) \\ \vdots \\ \tilde{S}\left(k+N_p\,|\,k\right) \end{bmatrix}$$

As mentioned before, the new state equation is obtained while $\Delta\delta_f$ is considered the control variable,

$$\tilde{O}(k) = \mathcal{F}\tilde{S}(k) + \mathcal{M}\Delta u(k) \tag{18}$$

Here,

$$\mathcal{F} = \begin{bmatrix} \tilde{C}_k\tilde{A}_k \\ \tilde{C}_k\tilde{A}_k^2 \\ \vdots \\ \tilde{C}_k\tilde{A}_k^{N_p} \end{bmatrix},$$

$$\mathcal{M} = \begin{bmatrix} \tilde{C}_k\tilde{B}_k & \dots & 0 \\ \vdots & \ddots & \vdots \\ \tilde{C}_k\tilde{A}_k^{N_c-1}\tilde{B}_k & \dots & \tilde{C}_k\tilde{B}_k \\ \vdots & \vdots & \vdots \\ \tilde{C}_k\tilde{A}_k^{N_p-1}\tilde{B}_k & \dots & \tilde{C}_k\tilde{A}_k^{N_p-N_c-1}\tilde{B}_k \end{bmatrix},$$

$$\Delta u(k) = \begin{bmatrix} \Delta u(k|k) \\ \Delta u(k+1|k) \\ \vdots \\ \Delta u(k+N_c|k) \end{bmatrix}$$

To let the control system calculate the control quantity that can track the desired trajectory as soon as possible, the below cost function is given in (19), the first item reflects the system's ability to follow the reference trajectory, and the second item is the requirement for the steady change of control quantity.

$$J = [\tilde{O}(k) - \tilde{O}_{ref}(K)]^T \cdot \mathbb{Q} \cdot [\tilde{O}(k) - \tilde{O}_{ref}(K)] + \Delta u(k)^T \cdot \mathbb{R} \cdot \Delta u(k) + \rho\epsilon^2 \tag{19}$$

There have several constraints during the whole control, such as the control $\delta_{f_{min}}(k) \leq \delta_f(k) \leq \delta_{f_{max}}(k)$, the increment $\Delta\delta_{f_{min}}(k) \leq \Delta\delta_f(k) \leq \Delta\delta_{f_{max}}(k)$, the output $\tilde{O}_{min}(k) \leq \tilde{O}(k) \leq \tilde{O}_{max}(k)$. Due to the complexity of the dynamic model and the situation that multiple constraints exist at the same time, the relaxation factor ϵ is added to avoid the failure to obtain the optimal solution within a specified time in the actual execution process. $\mathbb{Q}$ and $\mathbb{R}$ are the weight matrix, and ρ is the weight coefficient.

IV. RESULTS

To verify the effectiveness of the proposed control method, simulation verification with two different scenarios and among various vehicle speeds is carried out in MATLAB/Simulink-Car maker platform shown in Fig.6, in which BYD-Qin is chosen as the vehicle model, and the dry road condition is utilized during the test. Considering that there may be some deviation in the data collection, and also exist the time delay during the data transmission, some Gaussian noise is added in the simulation environment. The parameters used in this simulation model are given in Tab. 4.

$$\begin{cases} \dot{S} = AS + Bu \\ O = CS \end{cases}$$

$$A = \begin{bmatrix} \frac{2C_{Cf}\delta_f E_a}{m(\dot{x})^2} & \dot{\varphi} - \frac{2C_{Cf}\delta_f}{m\dot{x}} & 0 & \dot{y} - \frac{2C_{Cf}l_a\delta_f}{m\dot{x}} & 0\ 0 \\ E_a & \frac{2C_{Cf}-2C_{Cr}}{m\dot{x}} & 0 & -\dot{x} + \frac{E_l}{m\dot{x}} & 0\ 0 \\ 0 & 0 & 0 & 1 & 0\ 0 \\ E_b & \frac{E_l}{I_z\dot{x}} & 0 & \frac{2(C_{Cf}l_a^2 - C_{Cr}l_b^2)}{I_z\dot{x}} & 0\ 0 \\ cos\varphi & -sin\varphi - \dot{x}sin\varphi & -\dot{y}cos\psi & 0 & 0\ 0 \\ sin\varphi & cos\varphi & \dot{x}cos\varphi - \dot{y}sin\varphi & 0 & 0\ 0 \end{bmatrix}$$

$$E_a = -\dot{\varphi} - \frac{2C_{Cf}(\dot{y} + l_a\dot{\varphi}) + 2C_{Cr}(l_b\dot{\varphi} - \dot{y})}{m(\dot{x})^2}$$

$$E_b = \frac{2l_bC_{Cr}(l_b\dot{\varphi} - \dot{y}) - 2l_fC_{Cf}(\dot{y} + l_a\dot{\varphi})}{I_z(\dot{x})^2}$$

$$E_l = 2(C_{Cf}l_a + C_{Cr}l_b)$$

$$B = \begin{bmatrix} \frac{2C_{Cf}}{m}(2\delta_f - \frac{\dot{y}+l_a\dot{\varphi}}{\dot{x}}) \\ \frac{2(C_{Lf}s_f - C_{Cf})}{m} \\ 0 \\ \frac{2l_a(C_{Lf}s_f - C_{Cf})}{I_z} \\ 0 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{15}$$

TABLE 4. Parameters of the simulation model.

Symbol	Unit	Value
m	kg	2000
I_z	$kg.m^2$	4000
l_a	m	1.4
l_b	m	1.6
C_{yf}	N/rad	66900
C_{yr}	N/rad	62600

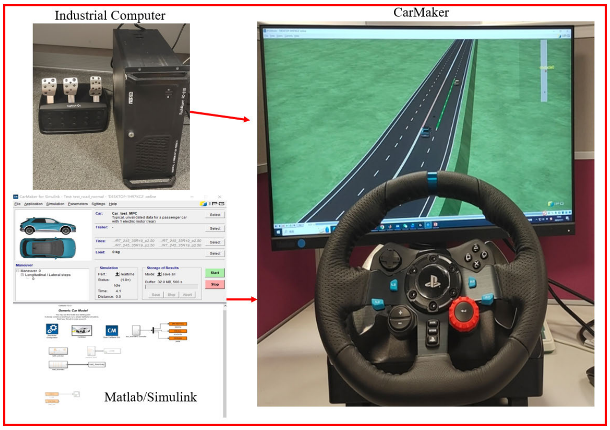
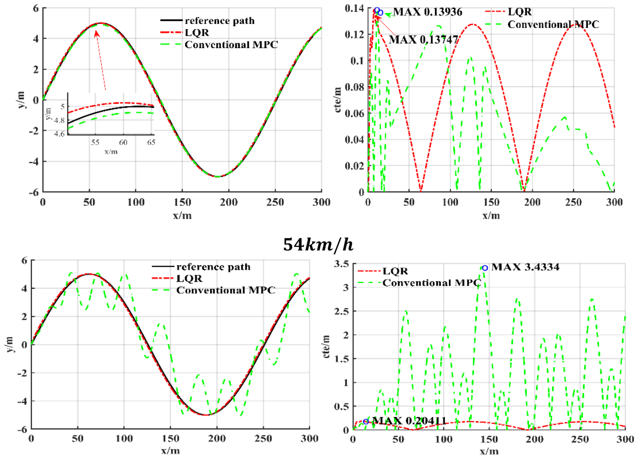


FIGURE 6. Experimental platform.

Due to the energy consumption of the cost functions mentioned in the path planner and outer-layer MPC design, the reference path can be treated as the optimal energy-saving path against these scenarios. Moreover, *cte* is chosen to test the control performance in this Section, as it is proportional to both the energy consumption and obstacle collision risk. During the whole HIL test, four driving speeds are used, $30km/h$ is the maximum speed allowed by the campus, $54km/h$ is the general driving speed in the city, and $72km/h(20m/s)$, $108km/h(30m/s)$ are selected to represent the highway speeds.



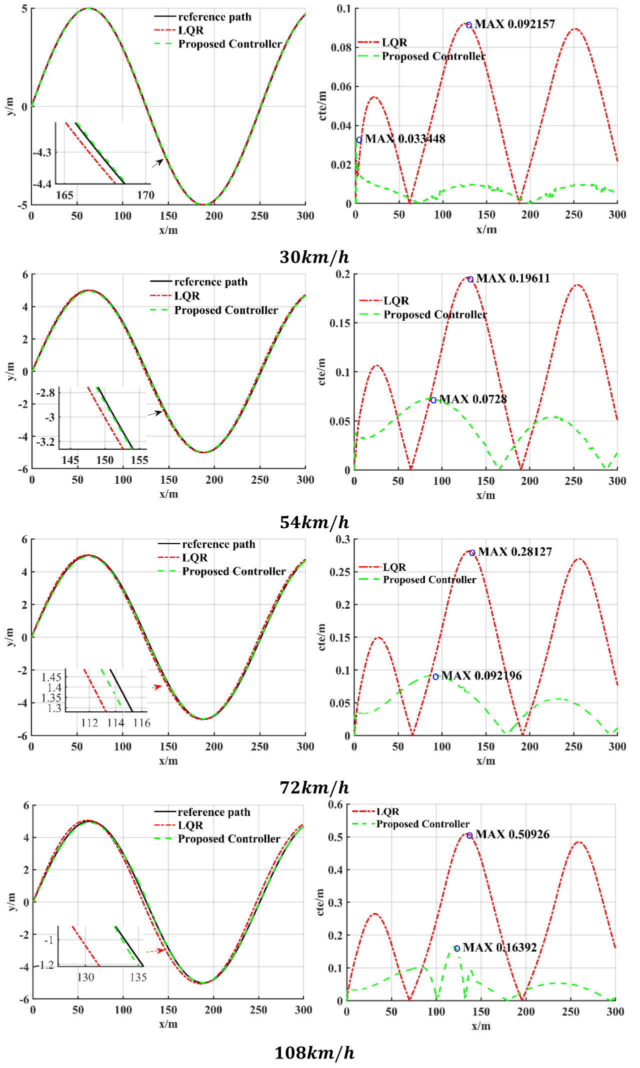
$54km/h$

$72km/h$

FIGURE 7. Path tracking at a fixed speed.

A. ADAPTIVE CURVATURE PERFORMANCE

A continuous sinusoidal curve without obstacles is used to verify the adaptive ability of the proposed algorithm. As shown in Fig.7, when the vehicle speed is $54km/h$, the maximum *cte* amplitudes generated by the traditional MPC controller and the LQR-based controller are almost equal, and the convergence state is superior. However, the performance of the conventional MPC control degrades rapidly when the vehicle speed is increased, and the maximum *cte* value in Fig.7 is already over $3m$. In contrast, the maximum *cte* value remains around $0.2m$ under the LQR control. The main reason for this phenomenon is that the modeling error that existed in the conventional MPC controller will inevitably be amplified in the high-speed scenario. When the initial path curvature is small, the path can be accurately tracked, whereas, the *cte* rapidly increases as the curvature becomes large, which causes system oscillation and out of control.



$30km/h$

$54km/h$

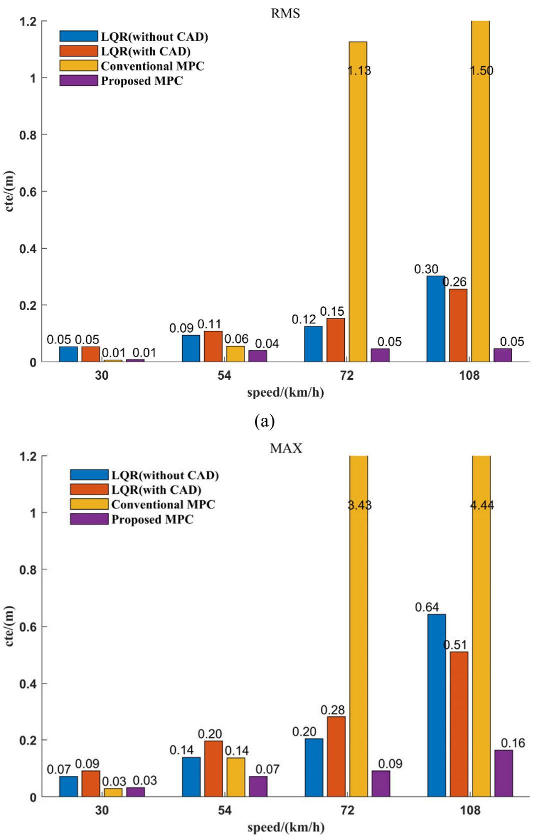
$72km/h$

$108km/h$

FIGURE 8. Path tracking with a curvature adaptive control.

To solve this problem, the speed constraint is added in the proposed MPC method to control the speed within a reasonable range to ensure that each curvature segment and the controller based on the dynamic model are in the controllable range, to ensure that the system is always in the controllable range. As shown in Fig.8, when the curvature adaptive is added, both the LQR-based and the proposed MPC-based controllers are improved in terms of *cte*. For the 30km/h scenario, the maximum *cte* of traditional LQR is about 0.1m, while this value can be controlled within 0.035m under the method proposed here, which is 1/3 of the traditional method. If the system deviation is set within 0.05m, it can be considered as zero deviation. Also, for the 54km/h scenario, the maximum *cte* under LQR and our method are 0.1961m and 0.0728m, respectively. In general, there are complex driving situations on campus or urban roads, such as multiple vehicles, roadblocks, bicycles, and pedestrians. Since the tracking deviation of the system is very small, the safety of the entire automatic driving process can be guaranteed only by ensuring the safety of the planned path.

When the average speed increases from 72km/h to 108km/h, the maximum *cte* of the vehicle under LQR control increases from 0.2813m to 0.5093m, while the controller proposed in this paper only increases from 0.0922m to 0.1630m.



(a)

(b)

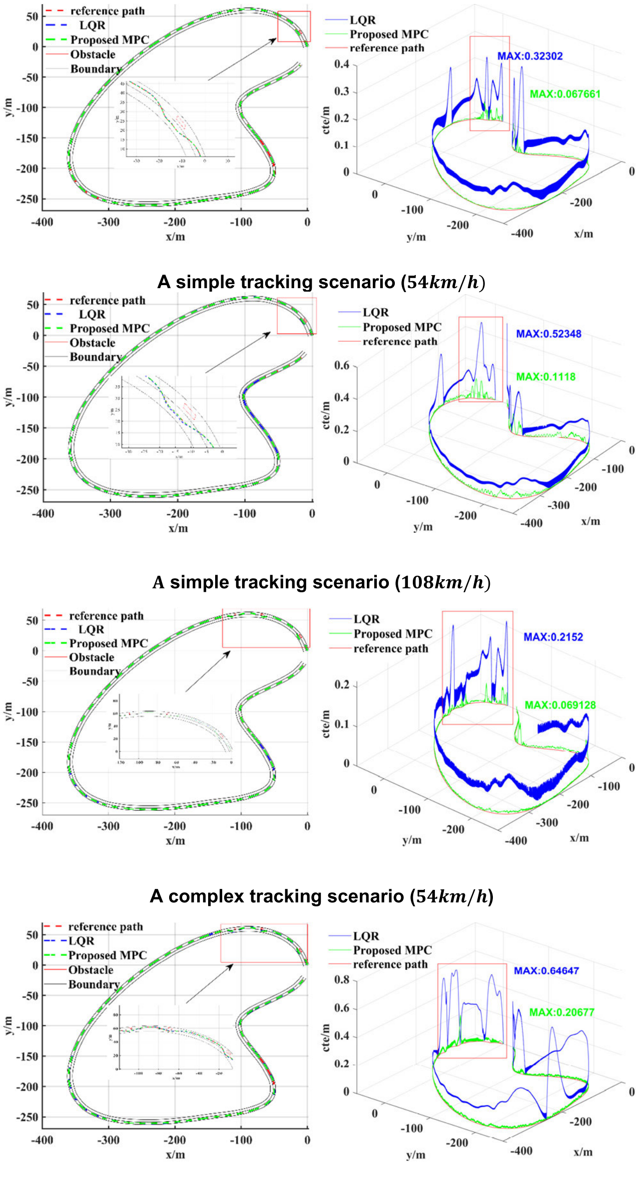
FIGURE 9. The *cte* against the vehicle speed for scenario one (a) RMS (b) Maximum.

Quantitation results of the *cte* among various vehicle speeds are summarized in Fig.9. For the lower driving speed, such as 30km/h and 54km/h, both the conventional MPC-based and the proposed MPC-based controllers have smaller *cte* compared with the LQR-based controller. However, the controllers with curvature adaptive appear better performance at high speeds, and the proposed MPC controller has the smallest value of *cte* with all vehicle speeds, that is to say, the vehicle under the control of the proposed method with the safest driving and the least energy consumption. Especially, with the proposed two-level MPC method, the RMS value of the *cte* for all these speeds can be controlled within 0.05m. Even if the maximum *cte* in 108km/h reaches 0.16m, the RMS and maximum values of *cte* are less than 0.1m when the driving speed is not exceeding 72km/h. In general, the smaller the driving speed, the smaller the *cte* will be, that is when the average driving speed does not exceed 72km/h, *cte* can be controlled within a reasonable range to meet the needs of automatic driving.

B. CONTINUOUS OBSTACLE AVOIDANCE

Using the continuous obstacle condition to test the path planning and obstacle avoidance ability of the control methods during the path tracking. Define the simple scenario with a single static/dynamic obstacle, and several static/dynamic obstacles in the complex scenario. As shown in Fig.8 and Fig. 9, the conventional MPC method has poor robustness at high speed, such only the LQR-based and proposed MPC-based controllers are tested with the obstacle situation. And, 54km/h and 108km/h are used as low and high speeds respectively in the test. In Fig.10, when faced with a simple tracking scenario, the maximum value of *cte* for the proposed MPC method is much smaller than the LQR-based method: 54km/h(0.067$m \ll$ 0.323m), 108km/h(0.112$m \ll$ 0.523m). Also, for a complex tracking scenario, the controller put forward in this paper has better performance than the LQR-based controller: 54km/h(0.069$m \ll$ 0.215m), 108km/h(0.206$m \ll$ 0.646m).

In the control process, the obstacle will be detected in real-time whether there is a danger of collision with the vehicle in the future, the detection period is 10ms. When there is a collision risk, the path will be planned in real-time according to the surrounding environment, and the planning time is controlled within 100ms, to ensure that the tracking process can deal with the situation of multiple obstacles in the surrounding environment. Quantitative results details of *cte*(RMS) are given in Fig.11, which show the superiority of a two-layer MPC method in tracking accuracy and energy consumption aspects. Whether the vehicle speed is 54km/h or 108km/h, the maximum value of *cte* under the control of the proposed MPC method can be ensured within 0.21m, however, it rises from 0.21m to 0.7m with control of LQR, especially in the vicinity of large path curvature for tracking. As mentioned in Section III-A, a constraint that the lateral distance of the vehicle with distance barriers is larger than 1/2 of the body is added in the path planning algorithm due to the consideration of vehicle traffic and safety. When the driving speed is 108km/h, the maximum *cte* of the method proposed here can be controlled within 0.21m, while the maximum *cte* that based on the LQR method reaches 0.65m> 3 × 0.21m. In other words, the maximum deviation of the superimposed



A simple tracking scenario ($54km/h$)

A simple tracking scenario ($108km/h$)

A complex tracking scenario ($54km/h$)

A complex tracking scenario ($108km/h$)

FIGURE 10. Path tracking with different kinds of obstacles.

TABLE 5. Quantities of the computational performance.

Scenario	Speed (km/h)	Adaptive prediction horizon (s) (Maximum, RMS)	Constant prediction horizon (s) (Maximum, RMS)
Simple	30	(0.0197, 0.0043)	(0.0216, 0.0061)
	54	(0.0221, 0.0051)	(0.0232, 0.0069)
	108	(0.0232,0.0065)	(0.0238, 0.0076)
Complex	30	(0.0205, 0.0049)	(0.0219, 0.0062)
	54	(0.0223, 0.0054)	(0.023, 0.0068)
	108	(0.0238, 0.0066)	(0.024, 0.0079)

object obstacle detection error is close to half the width of the vehicle, however, under the controller proposed in this paper, the value can be controlled within the range of 1/4 of the vehicle body.

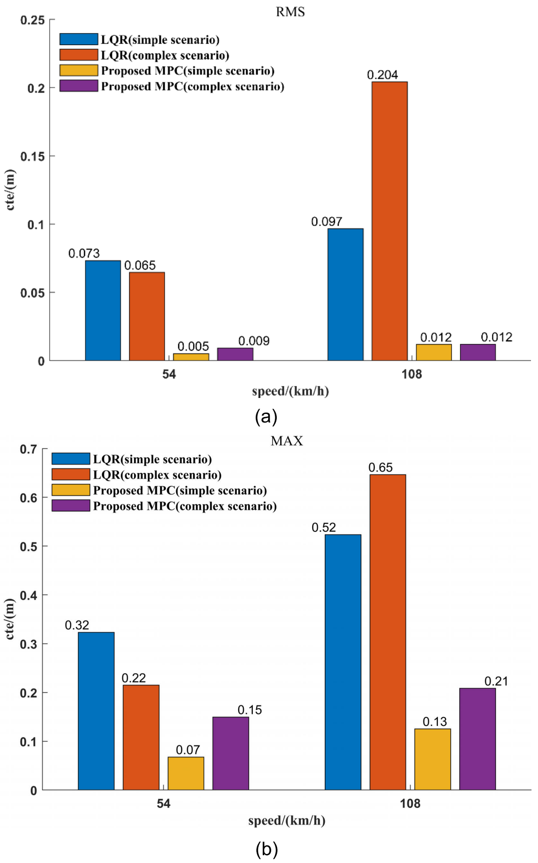


FIGURE 11. Quantitative results of cte (a) RMS (b) Maximum.

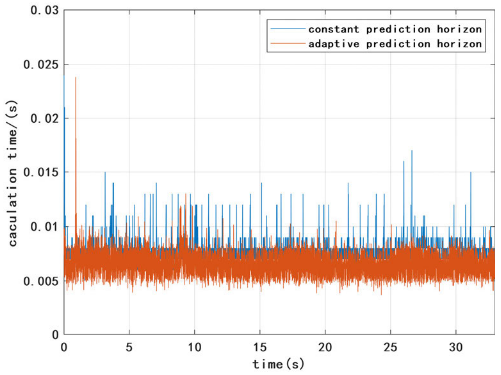


FIGURE 12. Computational performance in $108km/h$.

The comparison of computational performance between the normal MPC that had a constant prediction horizon and the control method proposed in this paper with the adaptive prediction horizon for various speeds and scenarios is given. The processor used in this control is the 11th Gen Intel(R) Core(TM) i7-11700 @ 2.50GHz 2.50 GHz. Quantity details of the computational performance are given in Tab.5, and details of the performance are shown in Fig. 12 for the driving speed at $\mathbf{108}\boldsymbol{km/h}$. Its obvious that the calculation time of the whole control system can be stabilized within $\mathbf{25}\boldsymbol{ms}$, and the control frequency is greater than $\mathbf{40}\boldsymbol{Hz}$, which fully meets the requirements of automatic driving for real-time vehicle control. As the speed increases and the complexity of the scene increases, the computation time increases, regardless of the maximum value or the RMS, and the time of the constant prediction horizon is longer than that of the adaptive prediction horizon. For the RMS time, the adaptive prediction horizon decreases relative to the constant prediction horizon, in which the maximum reduction is $\mathbf{41.8\%}$ (simple scenario, $\boldsymbol{30km/h}$) and the minimum reduction is $\mathbf{16.9\%}$ (simple scenario, $108km/h$). Based on the above analysis, it can be concluded that the adaptive prediction horizon has a relatively large improvement ($\mathbf{> 16.9\%}$) in the calculation time compared with the constant prediction horizon, which greatly improves the safety of automatic driving.

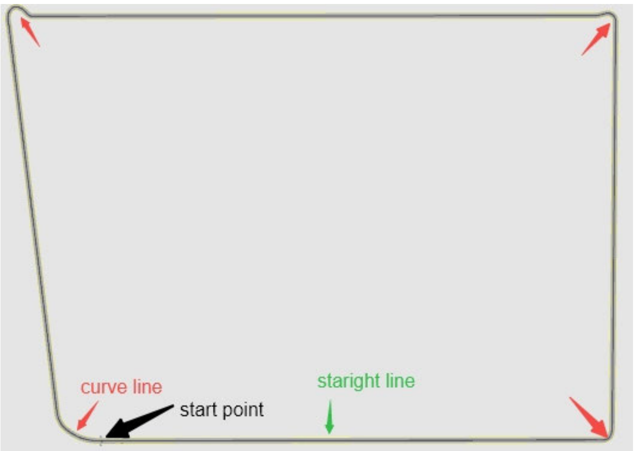


FIGURE 13. **A testing path built in Car maker.**

C. ANALYSIS OF THE ENERGY CONSUMPTION

The close-road testing environment with four straights and four corners is built in CarMaker and shown in Fig. 13, in which the black arrow marks the starting point of the driving test and the total path is $\boldsymbol{5km}$. Through using the energy system that exists in Car maker, the quantities of energy consumption in vehicles can be obtained, and the initial energy is set as $\mathbf{70\%}$. If the curvature is not considered in the control process, the tracking error will become larger with the increase of the driving speed, which will easily lead to the vehicle running out of the corner. Therefore, only $\boldsymbol{54km/h}$ is selected to view the energy consumption (case 1) of the control process without curvature consideration. After the vehicle has completed a circle of the path, the energy displayed by the system is $\mathbf{68.2\%}$, such the actual energy consumption is $\mathbf{1.8\%}$.

Normally, the electric vehicles with high speed, the power consumption capacity will be greatly improved. To better show the energy-saving performance, $\boldsymbol{108km/h}$ (case 2) is chosen for the vehicle after the curvature control is added. Also for one circle path, the battery capacity change rate is reduced from $\mathbf{70\%}$ to $\mathbf{68.3\%}$, and the actual energy consumption is $\mathbf{1.7\%}$, which is $\mathbf{0.1\%}$ less energy consumption than the normal speed. Meanwhile, with the same path tracking period, the driving time of case 1 takes twice as long as in the case.

V. CONCLUSION

An improved two-layer MPC method is proposed here to deal with path tracking in the presence of obstacles, especially facing multiple curvatures and high-speed driving. On the premise of energy saving and obstacle avoidance, the path planner based on the improved A* algorithm is built to generate an optimal reference path. According to this path, a series of verification tests are done among different kinds of control strategies. After comparing with the conventional MPC-based and LQR-based controllers, the proposed one has superior tracking accuracy and lower energy consumption mainly shown through quantitative values of the cross-track error. Moreover, the proposed controller that adopts a two-layer design with an optimal prediction domain largely reduced the computing load.

Key contributions of this design are summarized below:

1. The conventional process of autonomous driving is path planning, speed planning, and path tracking. The method proposed here can simplify this process and eliminate the intermediate speed planning, which reduces the steps required for actual deployment and improves the operability of the system.

2. The dynamic prediction time domain function is added to the path-tracking design. Compared with the fixed time domain method, the calculation cycle is shortened by about 20%. For autonomous driving, the shorter processing cycle means that more tasks can be processed simultaneously, and the impact of changes in the surrounding environment can be processed faster, improving the system's safety.

3. Of course, the MPC method proposed in this paper is not only applicable to the high speed $108km/h$, but it also plays a better role in the campus driving speed like $30km/h$. In the lower speed scenarios, combined with the path planning algorithm proposed in this paper, zero-deviation path tracking can be realized to improve the safety of automatic driving. In the high-speed scenarios (e.g. $108km/h$) the maximum cte is kept within $0.21m$, while the target point is reached faster with the same energy consumption or less (CarMaker/Simulink simulation test), which largely improves the traffic efficiency of the vehicles.

Nevertheless, the external disturbance caused by the different friction coefficients of the road surface, and the internal disturbance caused by the load change of the vehicle are not taken into account during the design of the two-level MPC method. Future work will focus on the introduction of the extended state observer to estimate the total disturbance of the system and compensate for it at the control end to improve the stability of the control system. In addition, the actual autonomous vehicle will be ready during the project, and thus, the control method will also be applied to the actual vehicle, and there will be more scenarios to be tested.

REFERENCES

[1] P. Zhao, J. Chen, Y. Song, X. Tao, T. Xu, and T. Mei, "Design of a control system for an autonomous vehicle based on adaptive-PID," *Int. J. Adv. Robot. Syst.*, vol. 9, no. 2, p. 44, Aug. 2012.

[2] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, "A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller," *IEEE Access*, vol. 8, pp. 166525–166540, 2020.

[3] E. Horváth, C. Hajdu, and P. Korös, "Enhancement of pure-pursuit path-tracking algorithm with multi-goal selection," in *Proc. 1st IEEE Int. Conf. Gridding Polytope Based Modeling Control (GPMC)*, Budapest, Hungary, Nov. 2019, pp. 13–18.

[4] J. Ahn, S. Shin, M. Kim, and J. Park, "Accurate path tracking by adjusting look-ahead point in pure pursuit method," *Int. J. Automot. Technol.*, vol. 22, no. 1, pp. 119–129, Feb. 2021.

[5] M. Cibooglu, U. Karapinar, and M. T. Söylemez, "Hybrid controller approach for an autonomous ground vehicle path tracking problem," in *Proc. 25th Medit. Conf. Control Autom. (MED)*, Valletta, Malta, Jul. 2017, pp. 583–588.

[6] X. D. Xue, K. W. E. Cheng, and S. L. Ho, "A self-training numerical method to calculate the magnetic characteristics for switched reluctance motor drives," *IEEE Trans. Magn.*, vol. 40, no. 2, pp. 734–737, Mar. 2004.

[7] E. Seo, S. Lee, G. Shin, H. Yeo, Y. Lim, and G. Choi, "Hybrid tracker based optimal path tracking system of autonomous driving for complex road environments," *IEEE Access*, vol. 9, pp. 71763–71777, 2021.

[8] J. Chen, W. Zhan, and M. Tomizuka, "Autonomous driving motion planning with constrained iterative LQR," *IEEE Trans. Intell. Vehicles*, vol. 4, no. 2, pp. 244–254, Jun. 2019.

[9] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.

[10] S. Li, G. Wang, B. Zhang, Z. Yu, and G. Cui, "Vehicle stability control based on model predictive control considering the changing trend of tire force over the prediction horizon," *IEEE Access*, vol. 7, pp. 6877–6888, 2019.

[11] Y. Tian, Q. Yao, P. Hang, and S. Wang, "A gain-scheduled robust controller for autonomous vehicles path tracking based on LPV system with MPC and H_∞," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9350–9362, Sep. 2022.

[12] S. Peicheng, L. Li, X. Ni, and A. Yang, "Intelligent vehicle path tracking control based on improved MPC and hybrid PID," *IEEE Access*, vol. 10, pp. 94133–94144, 2022.

[13] L. Zhai, C. Wang, Y. Hou, and C. Liu, "MPC-based integrated control of trajectory tracking and handling stability for intelligent driving vehicle driven by four hub motor," *IEEE Trans. Veh. Technol.*, vol. 71, no. 3, pp. 2668–2680, Mar. 2022.

[14] C. Dai, C. Zong, and G. Chen, "Path tracking control based on model predictive control with adaptive preview characteristics and speed-assisted constraint," *IEEE Access*, vol. 8, pp. 184697–184709, 2020.

[15] L. Tang, F. Yan, B. Zou, K. Wang, and C. Lv, "An improved kinematic model predictive control for high-speed path tracking of autonomous vehicles," *IEEE Access*, vol. 8, pp. 51400–51413, 2020.

[16] C. M. Kang, S.-H. Lee, and C. C. Chung, "Multirate lane-keeping system with kinematic vehicle model," *IEEE Trans. Veh. Technol.*, vol. 67, no. 10, pp. 9211–9222, Oct. 2018.

[17] X. Lin, Y. Tang, and B. Zhou, "Improved model predictive control path tracking strategy based an online updating algorithm with cosine similarity and a horizon factor," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12429–12438, Aug. 2022.

[18] K. Liu, J. Gong, A. Kurt, H. Chen, and U. Ozguner, "Dynamic modeling and control of high-speed automated vehicles for lane change maneuver," *IEEE Trans. Intell. Vehicles*, vol. 3, no. 3, pp. 329–339, Sep. 2018.

[19] S. Boyle, "Pacejka magic formula tire model parser," in *Proc. Int. Conf. Comput. Sci. Comput. Intell. (CSCI)*, Las Vegas, NV, USA, Dec. 2019, pp. 517–518.

[20] R.-H. Zhang, Z.-C. He, H.-W. Wang, F. You, and K.-N. Li, "Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system," *IEEE Access*, vol. 5, pp. 6649–6660, 2017.

[21] G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou, "Geometric A-star algorithm: An improved A-star algorithm for AGV path planning in a port environment," *IEEE Access*, vol. 9, pp. 59196–59210, 2021.

[22] J. Ji, A. Khajepour, W. W. Melek, and Y. Huang, "Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints," *IEEE Trans. Veh. Technol.*, vol. 66, no. 2, pp. 952–964, Feb. 2017.

[23] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Math. Program.*, vol. 106, no. 1, pp. 25–57, Mar. 2006.

[24] L. Qin, J. Hu, H. Li, and W. Chen, "Fuzzy logic controllers for specialty vehicles using a combination of phase plane analysis and variable universe approach," *IEEE Access*, vol. 5, pp. 1579–1588, 2017.

[25] J. Sun, X. Xue, and K. W. E. Cheng, "Four-wheel anti-lock braking system with robust adaptation under complex road conditions," *IEEE Trans. Veh. Technol.*, vol. 70, no. 1, pp. 292–302, Jan. 2021.

[26] D. Chu, H. Li, C. Zhao, and T. Zhou, "Trajectory tracking of autonomous vehicle based on model predictive control with PID feedback," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 2, pp. 2239–2250, Feb. 2023.

HEBING LIU received the M.Sc. degree in electrical and information engineering from the City University of Hong Kong, Hong Kong, China, in 2015. He is currently pursuing the Ph.D. degree in electrical and electronic engineering with The Hong Kong Polytechnic University, Hong Kong. His research interests include intelligent vehicle decision, planning, and control.

JINHONG SUN received the B.Sc. degree from the Electrical Engineering and Automation Department, Shandong University of Science and Technology, Shandong, China, in 2014, the M.Sc. degree in electronic and information engineering from the City University of Hong Kong, Hong Kong, China, in 2015, and the Ph.D. degree in electrical engineering from The Hong Kong Polytechnic University, Hong Kong, in 2021. She is currently a Postdoctoral Fellow with the Department of Electrical and Electronic Engineering, The Hong Kong Polytechnic University. Her research interests include power electronics, renewable energy, and technical control of electric vehicles.

KA WAI ERIC CHENG (Fellow, IEEE) received the B.Sc. and Ph.D. degrees from the University of Bath, Bath, U.K., in 1987 and 1990, respectively. Before joining The Hong Kong Polytechnic University, Hong Kong, in 1997, he was with Lucas Aerospace, London, U.K., as a Principal Engineer. He is currently a Professor and the Director of the Power Electronics Research Centre, Department of Electrical and electronic Engineering, Faculty of Engineering, The Hong Kong Polytechnic University. He has authored or coauthored more than 400 articles and seven books. His research interests include all aspects of power electronics, motor drives, electromagnetic interference, electric vehicles, battery management, and energy saving.

He was a recipient of the Institution of Electrical Engineers Sebastian Z. de Ferranti Premium Award, in 1995; the Outstanding Consultancy Award, in 2000; the Faculty Merit Award for Best Teaching from The Hong Kong Polytechnic University, in 2003; the Faculty Engineering Industrial and Engineering Services Grant Achievement Award, in 2006; the Brussels Innova Energy Gold Medal with Mention, in 2007; the Consumer Product Design Award, in 2008; the Electric Vehicle Team Merit Award of the Faculty, in 2009; the Geneva Invention Expo Silver Medal, in 2011; the Eco Star Award, in 2012; the Gold Prize at Seoul International Invention Fair, in 2015; the iCAN Gold Medal at Canada, in 2016; the Gold Award of HK Innovation and Technology, in 2017; the Geneva Invention Expo Silver Medal for Contribution in e-Antilock Braking Systems, in 2021; and TechConnect Innovation Award for his Contribution in Ammonia Electric Vehicle, in 2023.