Implementing Trajectory Tracking Control Algorithm for Autonomous Vehicle

1st Xuting Duan
School of Transportation Science and Engineering
Beihang University
Beijing, China
duanxuting@buaa.edu.cn

2nd Qi Wang
School of Transportation Science and Engineering
Beihang University
Beijing, China
qiwang@buaa.edu.cn

3rd Daxin Tian*
School of Transportation Science and Engineering
Beihang University
Beijing, China
dtian@buaa.edu.cn

4th Jianshan Zhou
School of Transportation Science and Engineering
Beihang University
Beijing, China
jianshanzhou@foxmail.com

5th Jian Wang
School of Transportation Science and Engineering
Beihang University
Beijing, China
07248@buaa.edu.cn

6th Zhengguo Sheng
Department of Engineering and Design
University of Sussex
Brighton, U.K.
z.sheng@sussex.ac.uk

7th Dezong Zhao
James Watt School of Engineering
University of Glasgow
Glasgow, U.K.
Dezong.Zhao@glasgow.ac.uk

8th Yafu Sun
China TransInfo Technology Co., Ltd
Beijing, China
sunyafu@ctfo.com

***Abstract*—This paper adopts the lateral and longitudinal control decoupling strategy to study the control algorithm of automated driving systems. The Frenet-frame is used for lateral control. According to the lateral error and heading error of the self-driving vehicle and the reference path, the model predictive control (MPC) is used to realize the lateral control of the self-driving vehicle to complete the tracking of the target trajectory. Longitudinal control is separated into two modes: speed control and tracking control. According to the driving environment of autonomous vehicles, a decision-making control strategy is designed to achieve smooth switching of the controller for the two control modes. The speed control mode uses the proportional-integral-derivative (PID) controller to realize the speed tracking control of the reference speed or the driver's setting. The following error model is established for the tracking control mode, and the Linear-Quadratic-Regulator (LQR) controller is used to realize the distance control of the vehicle in front. Finally, the algorithm is validated based on the ROS-CoppeliaSim simulation platform and field testing of autonomous vehicles in the actual tournament. The results show that the lateral and longitudinal decoupling algorithm has a good control effect, environmental adaptability, and stability.**

Keywords—vehicle control, Frenet-frame, MPC, LQR, distance control

I. Introduction

A. Motivation

Autonomous vehicles have made significant progress in the last few decades, thanks to the rapid development of sensors, computers, and control technology software and hardware. Their related technologies have been widely used in production, transportation, mining, military, agriculture, and other fields. The key technologies of autonomous vehicles mainly include environment perception, behavior decision-making, path planning, and trajectory tracking. As one of the core technologies of autonomous vehicles, trajectory tracking is a prerequisite for automatic driving. In this paper, we investigate the autonomous driving trajectory tracking control algorithm in a constrained environment. The purpose is to simplify the composition of the autonomous driving system so that the control module can share some of the decision-making content to work efficiently and reliably in application scenarios, such as logistics distribution in closed parks and intelligent connections. Autonomous patrol and security checks are examples of application scenarios as well. In these scenarios, the task is generally a straight line, and the surroundings with few traffic participants are relatively simple. As a result, we reduce the complexity of the decision-making system and share the task from the control aspect. Furthermore, to ensure safe driving on fixed roadways, we decouple the control laterally and longitudinally and switch the state in the longitudinal control. Figure 1 illustrates the control architecture.

The main contributions of this article are summarized as follows:

- Decoupling the control module in the automatic driving system. The longitudinal control is based on the Frenet-frame to establish a vehicle lateral tracking error model. The vehicle kinematics model is used to design the MPC to optimize the turning

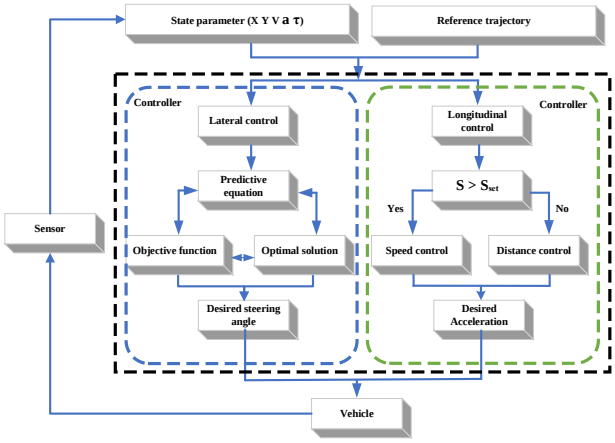


Fig. 1. Schematic diagram of control architecture.

This research was supported in part by the National Natural Science Foundation of China under Grant No. 62173012, U20A20155, and 61822101, in part by the Beijing Municipal Natural Science Foundation under Grant No. L191001, in part by the Newton Advanced Fellowship under Grant No. 62061130221, and in part by the Young Elite Scientists Sponsorship Program by Hunan Provincial Department of Education under Grant No. 18B142.

angle in different scenarios to realize the reference path for complex curvature changes.

- The speed control and distance control are switched according to the vehicle operating environment in the longitudinal control. The LQR controller is designed for the distance control state. The distance and curvature information of the reference trajectory is used to obtain accurate vehicle spacing to achieve stable follow-up to the target state.

The remains of the paper are organized as follows: Part II introduces the model establishment and controller design of the lateral control algorithm. Part III introduces the longitudinal control strategy and the design of the speed controller and distance controller. Part IV is based on a simulation platform and an actual vehicle for verification. Finally, we conclude this paper.

B. *Related work*

The technology of trajectory tracking control is an essential element in the automatic driving system. Its primary function is to determine the steering, driving, and braking commands that allow the intelligent car to track the trajectory accurately based on the trajectory generated by the planning layer. When a vehicle's motion is decoupled, it could be classified into two types of control: lateral motion control and longitudinal motion control [1-3].

Current research on the lateral control of intelligent cars is divided into geometric theory and model predictive control theory. Many control techniques can be regarded as extensions of these two theories. Pure pursuit tracking algorithm and Stanley algorithm are two famous methods for path tracking control of unmanned vehicles based on geometric models[4-5]. Scholars from Carnegie Mellon University proposed the pure pursuit path tracking algorithm for robotics' path tracking control [4]. This method selects a preview point ahead of the path and draws a tangent arc between the current position and the preview point. The arc is tangent to the direction of the vehicle's body. According to the Geometric Vehicle Model, users can calculate the front wheel angle corresponding to the arc radius and use it as the control variable input. The tracking stability of the above algorithm has a strong dependence on the preview distance, and it is difficult to find a suitable value, and the algorithm is based on a simple geometric model and ignores the motion characteristics of the vehicle. Therefore, a large tracking error will occur during steering driving in a high-speed scene, so it is often used in a low-speed driving scene. The method based on model predictive control considers the motion characteristics of the vehicle, the constraints of the actuator, and the comfort constraints, etc., which can ensure stable and safe driving in the face of complex driving loops. GONG et al [6]. explained in detail how to use model predictive control methods in the tracking and control of autonomous vehicles from the two directions of vehicle kinematics and dynamics. Kuhne et al[7]. established a model for the kinematics of a mobile robot, obtained a linear discrete system state space model, and used the model predictive control algorithm for simulation tests to verify the trajectory tracking algorithm. FALCONE et al. designed predictive control based on nonlinear models and linear time for driverless vehicle lane change scenarios on low-attachment roads [8].

The PID control method is the most widely used in the longitudinal motion control of unmanned vehicles. Its main benefit is that it is simple and practical, On the theoretical level, proper adjustment of PID parameters based on the model will surely achieve good results. However, eliminate the impact of system parameter uncertainty and external disturbance on the longitudinal control performance[9]. The intelligent PID technique is used to correct for unmodeled dynamic features in the longitudinal control of the vehicle start-stop condition [10]. HANG et al. from Tongji University created an adaptive vehicle speed control rule by combining an RBF neural network with sliding mode control [11].Model predictive control is capable of dealing with multi-variable constrained optimal control problems and has a wide variety of applications. The nonlinear model predictive control was integrated to design a vehicle speed tracking controller [12]. In addition, some passenger automobiles are outfitted with advanced driving assistance systems recently. Adaptive cruise control is one of them that has received much attention from academics worldwide [13-20]. The vehicle employs intelligent decision-making algorithms to achieve standardized automatic following behavior. The driver just needs to manage the direction, which substantially lowers the driver's workload.

II. LATERAL CONTROL

A. *System Modeling and Problem Statement*

The reasonable selection of vehicle models is significant to realizing better path tracking effects for unmanned vehicles based on model predictive control. However, the reasonable selection here needs to consider the model's complexity, accuracy, and difficulty, and the complexity varies with the model. As the accuracy increases, overly complex models will not only take up a lot of computing resources but also provide the difficulty of optimization. As a result, it will be challenging to ensure the real-time performance of the control algorithm based on different application scenarios and platform computing capabilities.

This article focuses on vehicles and mobile robots with Ackerman steering structures in low-speed scenes. Because they are driving at low speeds on good roads, they do not need to consider vehicle handling stability and dynamics. Based on the kinematics model from Researching the law of motion from the perspective of geometry and designing a path tracking controller can achieve low model complexity, convenient optimization and accuracy, and a better path tracking effect.

Based on the Frenet-frame to build a tracking error model as shown in figure 2.

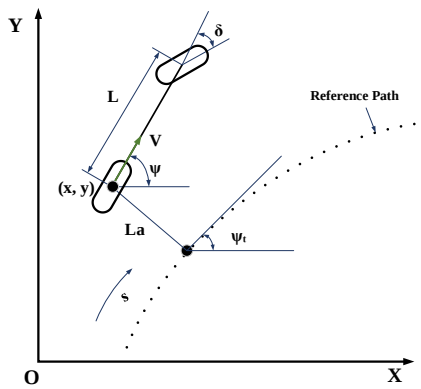


Fig. 2.Coordinitiong definition for path following task.

With reference to Fig. 2, assuming that the coordinates of the autonomous vehicle under the global path point are(x, y), the reference path is given by a series of discrete coordinate points defined by the planning module or implementation, L represents the wheelbase of the autonomous vehicle, and δ represents the vehicle front-wheel turning angle, Ψ represents the heading angle, V represents the vehicle speed, L_a represents the lateral error, Ψ_t represents the angle between the tangent of the reference path point and the X-axis, and S represents the arc length parameter along the path.

Define the heading error $\Psi_e = \Psi - \Psi_r$, the instantaneous curvature of the reference path is $k(s)$, and the relationship is as follows[21]:

$$k(s) = \frac{d\Psi_r}{ds} \tag{1}$$

$$\dot{\Psi}_r = k(s)\dot{s} \tag{2}$$

which implies:

$$\dot{s} = V\cos(\Psi_e) + \dot{\Psi}_r L_a \tag{3}$$

$$\dot{L}_a = V\sin(\Psi_e) \tag{4}$$

$$\dot{\Psi}_e = \dot{\Psi} - \dot{\Psi}_r \tag{5}$$

$$\dot{\Psi}_e = \frac{V\tan(\delta)}{L} - \frac{k(s)V\cos(\Psi_e)}{1-k(s)L_a}. \tag{6}$$

Combine the above equations, and we get:

$$\begin{bmatrix} \dot{L}_a \\ \dot{\Psi}_e \end{bmatrix} = \begin{bmatrix} \sin(\Psi_e) \\ \frac{\tan(\delta)}{L} - \frac{k(s)\cos(\Psi_e)}{1 - k(s)L_a} \end{bmatrix} V. \tag{7}$$

For the heading angle deviation, use the small-angle assumption $sin(\Psi_e) = \Psi_e$, $cos(\Psi_e) = 1$, and let $k(s)L_a = 0$, we get:

$$\begin{bmatrix} \dot{L}_a \\ \dot{\Psi}_e \end{bmatrix} = \begin{bmatrix} 0 & V \\ 0 & 0 \end{bmatrix} \begin{bmatrix} L_a \\ \Psi_e \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{V}{L\cos^2\delta_r} \end{bmatrix} \delta + \begin{bmatrix} 0 \\ -Vk(s) \end{bmatrix}. \tag{8}$$

Let $x=[L_a \quad \Psi_e]^T$, $u = \delta$ The vehicle error model can be expressed as:

$$\dot{x} = Ax + Bu + z. \tag{9}$$

Further, adopt the forward Euler method for discretization processing to obtain the discrete system state equation:

$$\dot{x}(k) = Ax(k) + Bu(k) + z(k) = \frac{x(k+1) - x(k)}{T}$$

$$x(k+1) = (AT + E)x(k) + BTu(k) + Tz(k) \tag{10}$$

where $A_k = (AT + E) = \begin{bmatrix} 1 & Vt \\ 0 & 1 \end{bmatrix}$; $B_k = BT = \begin{bmatrix} 0 \\ TL\cos^2\delta_r \end{bmatrix}$; $z_k = Tz(k) = \begin{bmatrix} 0 \\ -Vk(s)T \end{bmatrix}$, and T is the sampling time.

B. Path Tracking Controller Design

The design of the objective function is an essential part of the model predictive controller, which is mainly designed according to the requirements of the control system for control indicators. For example, for the model predictive controller for path tracking, it is hoped that the control system can enable the robot chassis to quickly and smoothly catch up Target path.

$$J(k)=\sum_{i=1}^{N_p} \|y(k+i|k)\|_Q^2 + \sum_{i=1}^{N_C-1} \|u(k+i|k)-u_r(k+i|k)\|_R^2 \tag{11}$$

Among them, N_P is the prediction horizon, and N_C is the control horizon.

According to the model to construct the prediction equation to predict the future output of the vehicle:

$$Y=\zeta x(k) + \Theta U + Z \tag{12}$$

where

$$Y = \begin{bmatrix} x(k+1|k) \\ x(k+2|k) \\ \vdots \\ x(k+N_C|k) \\ \vdots \\ x(k+N_P|k) \end{bmatrix}_{N_p\times 1} \quad \zeta=\begin{bmatrix} A_k \\ A_k^2 \\ \vdots \\ A_k^{N_c} \\ \vdots \\ A_k^{N_p} \end{bmatrix}_{N_p\times 1}$$

$$\Theta=\begin{bmatrix} B_k & 0 & 0 & \cdots & 0 \\ A_kB_k & B_k & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_k^{N_c-1}B_k & A_k^{N_c-2}B_k & A_k^{N_c-3}B_k & \cdots & B_k \\ A_k^{N_c}B_k & A_k^{N_c-1}B_k & A_k^{N_c-2}B_k & \cdots & A_kB_k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_k^{N_p-1}B_k & A_k^{N_p-2}B_k & A_k^{N_p-3}B_k & \cdots & A_k^{N_p-N_c}B_k \end{bmatrix}_{N_p\times N_c}$$

$$U=\begin{bmatrix} u(k|k) \\ u(k+1|k) \\ \vdots \\ u(k+N_C-1|k) \end{bmatrix}_{N_c\times 1}$$

$$Z=\begin{bmatrix} z(k) \\ A_kz(k)+z(k) \\ A_k^2z(k)+A_kz(k)+z(k) \\ \vdots \\ A_k^{N_p-1}z(k)+A_k^{N_p-2}z(k)+\ldots+A_kz(k)+z(k) \end{bmatrix}_{N_p\times 1}.$$

According to the "prediction equation", it can be seen that the current state quantity $x(k|k)$ and the control quantity $u(k|k)$ of the system can be calculated to obtain the state quantity and output quantity in the predicted horizon.

Suppose $E=\zeta x(k) + Z$ and put the prediction equation into the optimization objective function:

$$\begin{aligned} J &= Y^TQ_QY + (U - U_{ref})^TR_R(U - U_{ref}) \\ &= (E+\Theta U)^TQ_Q(E+\Theta U)+(U - U_{ref})^TR_R(U - U_{ref}) \\ &= U^T\left(\Theta^TQ_Q\Theta + R_R\right)U + 2\left(E^TQ_Q\Theta - U_{ref}^TR_R\right)U \\ &\quad + (E^TQ_QE + U_{ref}^TR_RU_{ref}). \end{aligned} \tag{13}$$

In the formula, $Q_Q = I_{N_P}\otimes Q$, $R_R = I_{N_P}\otimes R$, $\otimes$ represents the Kronecker product, and $(E^TQ_QE + U_{ref}^TR_RU_{ref})$ has nothing to do with the control quantity, so in the optimization solution, it can be ignored, and the cost function can be expressed as:

$$J=U^T\left(\Theta^TQ_Q\Theta+R_R\right)U + 2\left(E^TQ_Q\Theta-U_{ref}^TR_R\right)U. \tag{14}$$

In order to facilitate the solution, the optimization problem of model predictive control is transformed into a standard quadratic programming problem,

set $H = \Theta^TQ_Q\Theta + R_R$, $f = E^TQ_Q\Theta - U_{ref}^TR_R$,

$$J = 2\left(\frac{1}{2}U^T HU + f^T U\right). \quad (15)$$

The system constraints are:

$$u_{min} \le u(k) \le u_{max},\ k = 0,1,\cdots,N_C\text{-}1$$

After conversion

$$\min\ J = \frac{1}{2}U^T HU + f^T U \quad (16)$$

$$\text{s. t.}\ \ U_{min} \le U \le U_{max}\ .$$

After the optimization solution, a series of control variables in the control time domain is obtained

$$U=[u(k)\quad u(k+1)\quad \cdots\quad u(k+N_C\text{-}1)]^T$$

III. Longitudinal Control

A. Control Strategy

When there is no target vehicle in front of the autonomous vehicle, or the actual distance between the target vehicle and the self-driving vehicle is greater than the expected safety distance, the system enters the speed control mode and cruises at the reference speed on the waypoints. The system enters the tracking control mode when there is a target vehicle in front of the autonomous vehicle, and the actual distance between the two vehicles is less than the set safe distance S (Figure 3). The autonomous vehicle's braking is controlled when the target vehicle slows down to avoid rear-end collisions. When the target vehicle accelerates, the autonomous vehicle will soon follow and accelerate to the same speed (the target vehicle speed does not exceed the set cruising speed vmax). After the vehicle speed surpasses the indicated maximum speed, the current speed should be maintained at a constant level.

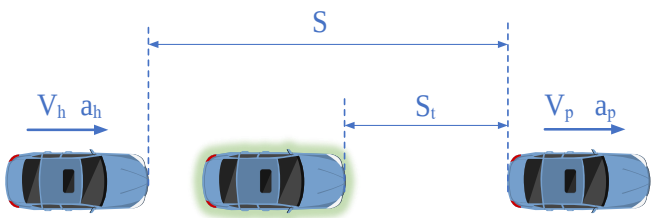


Fig. 3.Schematic diagram of longitudinal movement.

This strategy is based on a fixed-time headway model: v represents speed, a represents acceleration, subscripts p and h represent the preceding and autonomous vehicles, respectively. S represents the conditions for switching between the speed control mode and the tracking control mode. The conditions are set and affected by the system and Limit the performance of the sensor. When the distance between two cars in the same lane is less than S, the system switches to the tracking control mode. At this time, it is expected to maintain a distance of $\boldsymbol{S_t}$ from the preceding vehicle.

$$S_t = V_P\tau + d_0 \quad (17)$$

where $\boldsymbol{\tau}$ is the time headway and $\boldsymbol{d_0}$ represents the distance between the front and rear cars that should be maintained after parking.

B. Controller Design

The control objective in speed control mode is to track the speed on the reference trajectory of the vehicle or to drive at the cruise speed set by the system. This section builds a PI speed controller based on PID theory to calculate the desired acceleration so that the controlled vehicle speed tends to the set speed, V_{set}, as follows:

$$a_{des} = K_p(V_{set} - v_h) + K_i\int_0^t (V_{set} - v_h)dt\,. \quad (18)$$

In the formula: K_p is the proportional gain coefficient, K_i is the integral gain coefficient, V_{set} is the target vehicle speed, and v_h is the vehicle speed.

When designing distance control, first consider longitudinal following performance and safety to ensure that the vehicle can safely follow the vehicle in front when the vehicle is following the control. Car-following performance shows that the distance between two cars is reasonable during the follow-up process. There will be no congestion and reduced traffic efficiency due to excessive distance. Safety requires that the vehicle's following distance cannot be too small to prevent the vehicle from being difficult to brake in time under dangerous conditions. Based on the above analysis, in the distance control process, the method based on the fixed period is used to determine the safe distance to the vehicle in front. It is worth noting that, unlike the traditional adaptive cruise, it uses the information on the reference path to match the sensor-based vehicle position to the reference path to obtain the relative distance further S_t of the vehicle during the driving process, as shown in Figure 4.

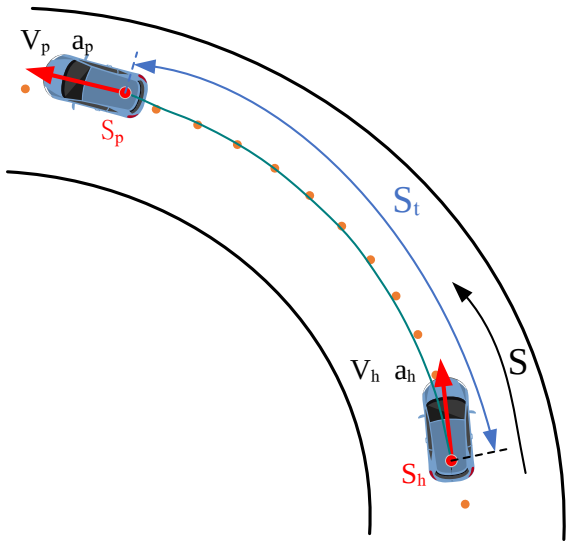


Fig. 4.Schematic diagram of the relative position of the vehicle.

The longitudinal relationship between vehicles is shown in the following formula:

$$S = S_p - S_h\,. \quad (19)$$

In the above formula, S_p and S_h are the positions of the vehicle ahead and the autonomous vehicle in the same lane in the path coordinate system.

State space equation:

$$\dot{x} = Ax + Bu + Gw \quad (20)$$

where:

$$x=\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} S_t\text{-}S \\ V_p\text{-}V_h \end{bmatrix}$$

$$u = a_{des} \quad w = a_p$$

$$A = \begin{bmatrix} 0 & -1 \\ 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, G = \begin{bmatrix} \tau \\ 1 \end{bmatrix}.$$

Further get the discrete state equation:

$$x(k+1) = A_t x(k) + B_t(k) + G_t w(k). \tag{21}$$

where $u = -Kx$, the cost function:

$$J = \frac{1}{2}\sum_{0}^{\infty}\left[(x^T(k)Qx(k) + u^T(k)Ru(k)\right] \tag{22}$$

where Q and R are the weighting matrices.

IV. Algorithm Verification

A. Simulation

In order to test the proposed path tracking controller, the high-fidelity software package CoppeliaSim is used to realize the high-precision model of the unmanned vehicle, and simulations are carried out in different scenarios. The overall simulation environment is a collaborative system that includes the CoppeliaSim simulation platform and is combined with ROS. Add motion constraints to the mechanical joint points of the high-precision vehicle model on the CoppeliaSim simulation platform to form a front-wheel steering rear-wheel drive smart car model with Ackerman steering capability; add GPS and IMU sensors and write feedback in its script file. The vehicle model's real-time position, speed, and attitude information interface in the simulation environment communicate with ROS.

In order to test the control performance of the lateral path tracking controller and the rationality of the system design, the design of simulation scenario 1 is shown in Figure 5: Blue and red static obstacles are used to form a 3-meter-wide 8-shaped road. The static obstacles on the left and right sides and the vehicle model are all set with collision attributes. When the vehicle touches the obstacles on both sides, it cannot move forward. The center path is the desired path, composed of two circles with a diameter of 18.25m tangent to the coordinate (15, 0). In the experiment, the vehicle accelerates from a stationary state to 5m/s from the starting position (-5, 0) and then longitudinally enters the figure-8 at a constant speed. First rounds the right circle and then cuts into the left circle at the tangent point and travels one round. Then drive out longitudinally at the tangent point. The wheelbase of the vehicle model is 1.54m, and the left and right wheelbase is 1.10 m. The simulation result is shown in Figure 6.

In simulation, the prediction step length and control step length in the model predictive control algorithm are particularly important. It should be ensured that the prediction step length is not less than the control step length $N_c \leq N_p$. The larger the selected prediction step length, the more considerable information about the future vehicle dynamics. However, better tracking performance can be obtained. On the other hand, a more extensive prediction range means that the calculation time of MPC is longer. Theories and experiments show that in the model predictive control algorithm, if the control step is larger, it is easier to obtain a smoother control effect. Therefore, the selection of prediction step length and control step length is essentially a trade-off between tracking performance and real-time calculation results. In summary, select the prediction step length and control step length as $N_p = 70$ and $N_c = 50$. The minimum and maximum steering wheel angle is limited to ± 0.61(rad).

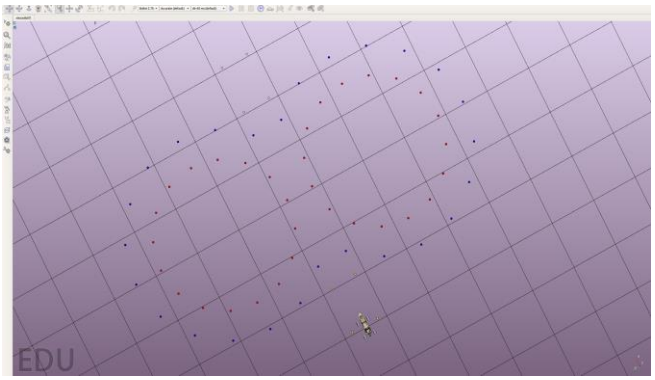
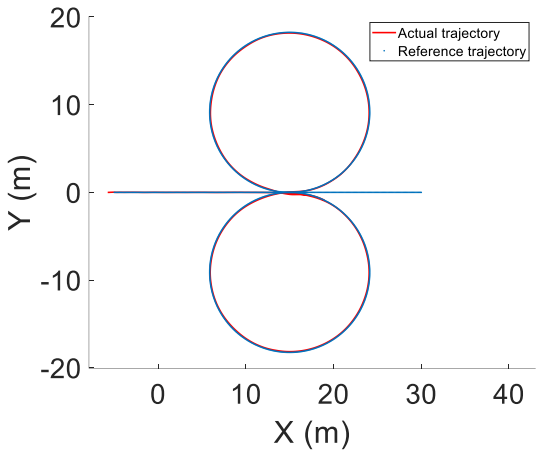
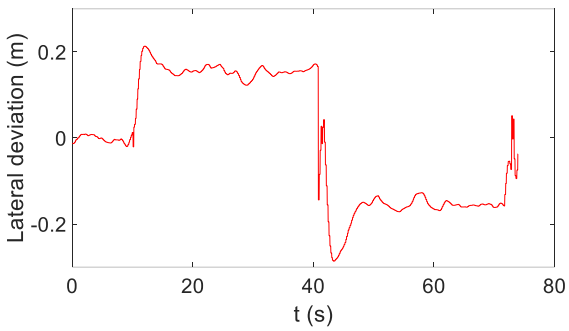


Fig. 5. Fixed curvature road scene.

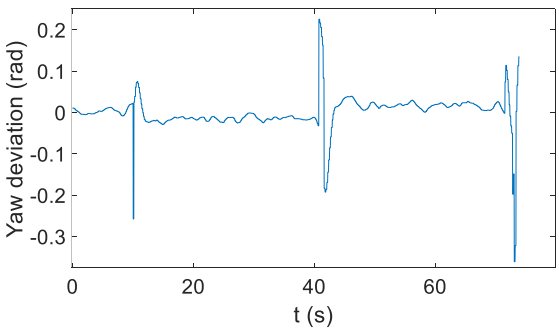
Experiment 1 mainly verifies the controller to test the tracking effect of the controller when the straight road changes to a curve and the curve changes to a straight road. Compared with the reference path, the tracking effect of the vehicle is shown in figure 6(a):



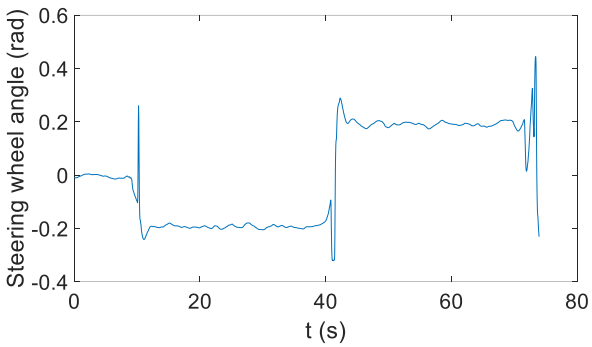
(a) Position



(b) Lateral deviation



(c) Yaw deviation



(d) Steering wheel angle

Fig. 6. Simulation results for experiment 1. (a) Position. (b) Lateral deviation. (c) Yaw deviation. (d) Steering wheel angle.

Note: The sudden changes at 10s, 40s, and 73s in the above result graph are caused by errors when the vehicle passes through the tangent point of the two circles and selects the left and right circular paths, and has nothing to do with our control algorithm (the work will be improved later). Figure 6(a) shows that the vehicle can accurately track the reference trajectory. The lateral error can be observed in Fig. 6(b). The maximum lateral error is no more than 0.2m. The heading error can be observed in Fig. 6(c). Again, the maximum error is not more than 0.1rad.

Based on experiment 1, in order to verify the tracking effect of the designed path tracking controller when facing a path with variable curvature, experiment 2 is designed. As shown in Figure 7, a reference path with continuous curvature is set on an analog map 120 meters long and 100 meters wide.

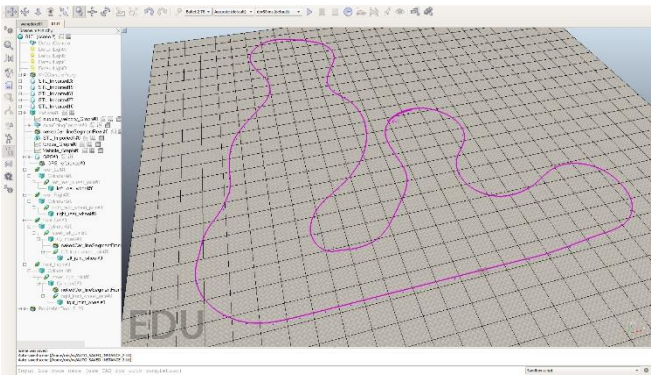


Fig. 7. Continuously variable curvature road.

The lateral tracking algorithm still has a good tracking effect on the road with continuous curvature, as shown in Figure 8.

B. *Real vehicle*

The longitudinal control algorithm is directly deployed on the actual vehicle for verification, as shown in Figure 9. The control algorithm in this article was used to participate in the "2021 World Intelligent Driving Challenge" held in Tianjin, and our team won the bronze medal. The participating vehicle includes light sensors such as lidar, camera, radar, GPS, etc. Based on the high-definition map provided by the race group, the company can drive safely under the restriction of 17 scene rules such as starting, changing lanes, following, parking, and V2X.

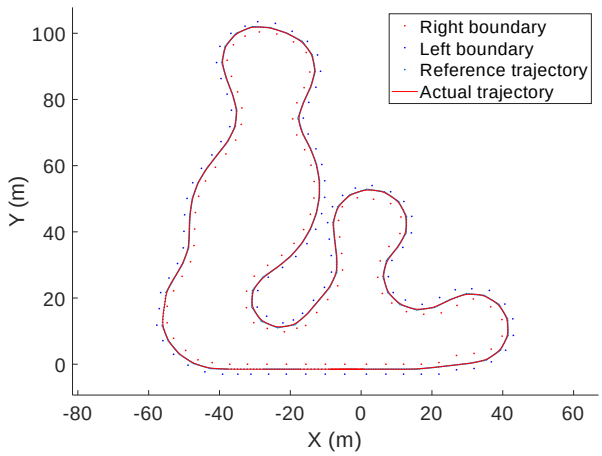


Fig. 8. Simulation results for experiment 2.

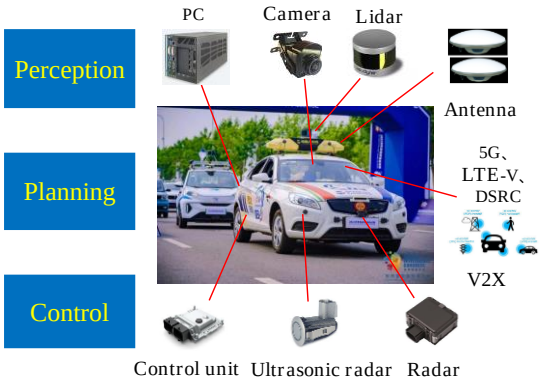
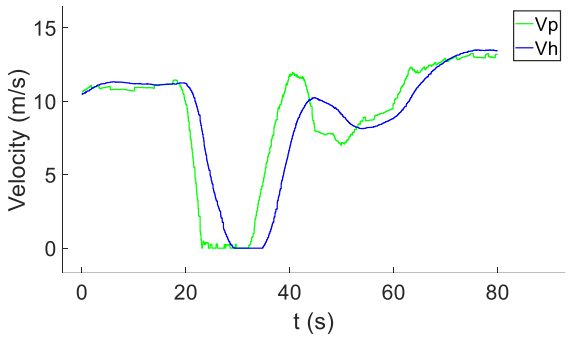
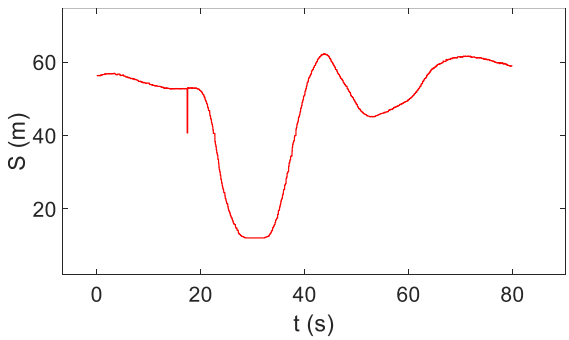


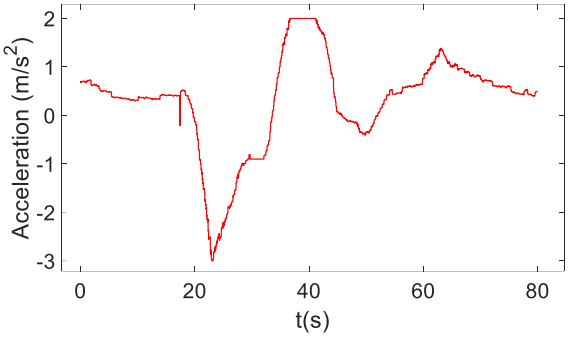
Fig. 9. Algorithm verification platform.



(a) Velocity



(b) Distance between front and rear vehicles



(c) Acceleration

Fig. 10. Results for longitudinal control. (a) Velocity. (b) Distance between front and rear vehicles. (c) Acceleration.

We selected the real-time data of one of the scenes. From Figure 10, we can see that the autonomous vehicle can follow the speed change of the preceding vehicle in time, and the calculated acceleration is within an appropriate range while maintaining a stable and safe distance from the preceding vehicle. In the period of 23s-44s, the effect of follow-up and stop-and-go was achieved, and the experimental results met the design requirements.

V. CONCLUSION

This article studies the longitudinal and lateral trajectory tracking control of autonomous vehicles. The specific research conclusions are summarized as follows: for lateral control, the model based on Frenet-frame tracking error paired with the MPC controller offers a superior control effect. The speed control and distance control state switches in the longitudinal control can be used to track the trajectory in the longitudinal direction. In addition, by mapping the vehicle on the path coordinates, the relative position of the front and rear vehicles can be accurately determined, substantially improving the longitudinal tracking ability.

VI. ACKNOWLEDGEMENT

We would like to thank the joint team of Yi Xian & Beihang University for their support in the research of this paper.

REFERENCES

[1] PADEN B, CAP M, YONG S Z, “A survey of motion planning and control techniques for self-driving urban vehicles[J],” IEEE Transactions on Intelligent Vehicles, 2016, 1(1) : 33-55.

[2] BUEHLER M, IAGNEMMA K, SINGH S. The DARPA urban challenge : Autonomous vehicles in city traffic[M]. Berlin : Springer, 2009.

[3] AMER N H, ZAMZURI H, HUDHA K, et al. Modelling and control strategies in path tracking control for autonomous ground vehicles : A review of state of the art and challenges[J]. Journal of Intelligent & Robotic Systems, 2017, 86(2) : 225-254.

[4] COULTER R C. Implementation of the pure pursuit path tracking algorithm[R]. Pittsburgh : Carnegie Mellon University Pittsburgh PA Robotics Institute, CMU-RI-TR-92-01, 1992.

[5] SNIDER J M. Automatic steering methods for autonomous automobile path tracking[R]. Pittsburgh : Carnegie Mellon University Pittsburgh PA Robotics Institute, CMURI TR-09-08, 2009.

[6] GONG Jianwei, JIANG Yan, XU Wei. Model predictive control for self-driving vehicles[M]. Beijing : Beijing Institute of Technilogy Press, 2014

[7] KUHNE F, LAGES WF, et al. Model Predictive Control of a Mobile Robot Using Linearization[C]. IEEE Prodeedings of Mechtronics and Robototics, 2004, 4 : 525 - 530.

[8] FALCONE P, BORRELLI F, ASGARI J, et al. Predictive active steering control for autonomous vehicle systems[J]. IEEE Transactions on Control Systems Technology, 2007, 15(3) : 566-580.

[9] MARINO R, SCALZI S, NETTO M. Nested PID steering control for lane keeping in autonomous vehicles[J]. Control Engineering Practice, 2011, 19(12) : 1459-1467.

[10] CHOI S, D'ANDRÉANOVEL B, FLIESS M, et al. Model-free control of automotive engine and brake for stop-and-go scenarios[C] Proceedings of the European Control Conference. Budapest : IEEE, 2009 : 3622-3627.

[11] HANG P, CHEN X, ZHANG B, et al. Longitudinal velocity tracking control of a 4WID electric vehicle[J]. IFAC PapersOnLine, 2018, 51(31) : 790-795.

[12] ZHU Daji. Longitudinal speed control for electric vehicles based on state observation[D]. Changchun : Jilin University, 2017.

[13] Zeeshan A, Charles G. Model predictive control with constraints for a nonlinear adaptive cruise control vehicle model in transition manoeuvres[J]. Vehicle System Dynamics, 2013, 51(6):943-963.

[14] Moon S, Yi W C K. Intelligent vehicle safety control strategy in various driving situations[J]. Vehicle System Dynamics, 2010, 48(sup1):537-554.

[15] Moon S, Yi K. Human driving data-based design of a vehicle adaptive cruise control algorithm[J]. Vehicle System Dynamics, 2008, 46(8):661-690.

[16] Moon S, Moon I, Yi K. Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance[J]. Control Engineering Practice, 2009, 17(4):442-455.

[17] Han D, Yi K. A driver-adaptive range policy for adaptive cruise control[J]. Proceedings of the Institution of Mechanical Engineers Part D Journal of Automobile Engineering, 2006, 220(3):321-334.

[18] Venhovens, P, Naab, K.and Adiprasito, B. Stop and go cruise control.In Seoul 2000 FISITA World Automotive Congress, Seoul, Korea, 12-15 June 2000.

[19] Yi K, Lee. An investigation of intelli- gent cruise controllaws for passenger vehicles. Proc. Instn Mech. Engrs,Part D, Journal of Automobile Engineering, 2001,215(D2),159-169.

[20] Sathiyan S P, Kumar S S, Selvakumar A I. Optimised fuzzy controller for improved comfort level during transitions in cruise and adaptive cruise control vehicles[C]. International Conference on Signal Processing and Communication Engineering Systems. IEEE, 2015:86-91.

[21] Moritz Werling, Julius Ziegler, S¨oren Kammel, and Sebastian Thrun. Optimal trajectory generation for dynamic street scenarios in a frenet frame. In Robotics and Automation (ICRA), 2010 IEEE International Conference on, pages 987–993. IEEE, 2010.