Sensors and Materials, Vol. 37, No. 6 (2025) 2351–2372
MYU Tokyo

S & M 4059

Real-time Obstacle Avoidance Control and Path Planning with Verification for Autonomous Vehicles Using Sensor Measurements

Gang Chen,[1] Ruo-Peng Kan,[1] Kun-Chieh Wang,[1*] Lei Wu,[1] and Shi-Yun Zhan[2]

[1]School of Mechanical and Electric Engineering, Sanming University, Sanming, Fujian Province 365004, China
[2]Sinotruk Fujian Haixi Automobile Co., Ltd., Yongan, Fujian Yong'an City Po Ling 99, Fujian Province, China

(Received January 27, 2025; accepted May 16, 2025)

Keywords: autonomous vehicles, lateral control, path tracking, path planning algorithm

In recent years, autonomous technology has advanced rapidly, with the real-time obstacle avoidance capability of autonomous vehicles on the road being a primary concern for ensuring safety. Therefore, we propose an innovative path planning and control methodology to enhance the real-time obstacle avoidance performance of autonomous vehicles. In terms of path planning, we propose a global path planning approach based on an optimized A* algorithm, incorporating a dynamic weighting method and cubic-spline curve smoothing. This improves path search efficiency by 20% and enhances path smoothness by 3%. Additionally, we introduce a local path planning method based on fifth-degree polynomial interpolation curves. Regarding the control approach, we design a position-velocity double-layer proportional-integral-derivative controller for longitudinal control and a feedforward linear quadratic regulator controller for lateral control. Finally, we validate the proposed methodology through control system design, driving path simulation analysis, and real-world vehicle tests using sensor measurements. The application of this innovative control method in a campus environment demonstrates a clear improvement in the real-time driving capability of autonomous vehicles, increasing the obstacle avoidance success rate by 5%. Furthermore, it enables autonomous vehicles to achieve precise control and navigation on the road, reducing the vehicle's lateral control error and shortening the longitudinal control response time by 0.1 s. These findings provide a valuable technical reference for path planning and control in autonomous vehicle systems.

1. Introduction

The widespread use of cars has considerably improved convenience in daily life but has also introduced several challenges, such as traffic accidents and urban congestion. Statistics show that human error accounts for up to 90% of traffic accidents, including issues such as drunk driving, speeding, overloading, and fatigue driving, while the proportion of accidents caused by mechanical failures or design defects is significantly lower. The advent of autonomous driving technology offers a promising solution to enhance traffic safety by mitigating human errors. It

*Corresponding author: e-mail: m18316252102@126.com
https://doi.org/10.18494/SAM5572

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

enables rapid responses in emergencies, effectively preventing accidents caused by driver mistakes, and significantly improves the overall safety of road travel.(1)

As a cutting-edge technology in the automotive field, autonomous driving technology is pivotal in addressing current traffic and mobility challenges. By integrating hardware components such as computing processors, storage management units, interface communication modules, and communication systems with software platforms and algorithms, a comprehensive hardware and software system is established. This system enables vehicles to perform a wide range of complex autonomous driving tasks without human intervention, including environmental perception, sensor fusion, decision-making and planning, control execution, and vehicle-to-road coordination. These capabilities effectively reduce traffic accidents caused by human factors such as driver fatigue and judgment errors, significantly enhancing the safety of road travel.

The Society of Automotive Engineers (SAE) classifies the automation levels of autonomous cars into six categories, ranging from L0 to L5.(2) In recent years, with advancements in big data and artificial intelligence technologies, autonomous cars have become a focal point of research in the automotive industry. As a key area of autonomous driving technology, path planning is essential for addressing current traffic challenges. Path planning involves determining a safe driving route from the starting point to the destination based on the evaluation criteria of the road environment. An optimal path not only enhances driving efficiency but also reduces energy consumption. Solving the path planning problem for autonomous vehicles requires addressing several sub-problems, including obstacle avoidance,(1) route determination,(3) the consideration of vehicle kinematic constraints,(2) and effective tracking control.(4)

The system of autonomous cars consists of four key components, namely, perception, decision-making, planning, and control, forming a closed-loop system, as illustrated in Fig. 1. This integrated system ensures the safe operation of the vehicle across various complex road environments. The perception module serves as the car’s senses, continuously gathering environmental data such as the presence of pedestrians, vehicles, and traffic signs. This information provides the foundation for subsequent decision-making. The decision-making

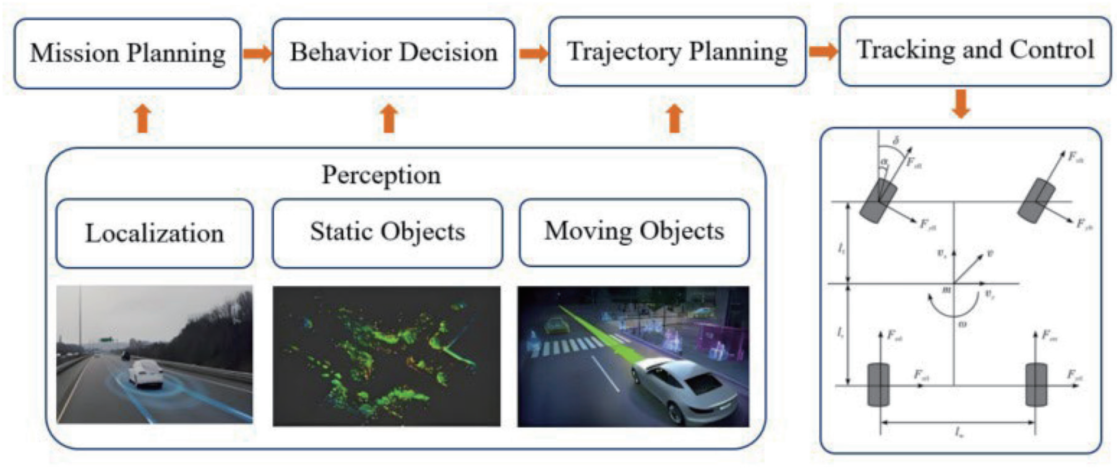


Fig. 1. (Color online) Framework of autonomous driving technology.

module processes the collected data to develop a logical driving plan based on traffic rules, real-time road conditions, and the driver's objectives. Once a driving plan is established, the planning module calculates the optimal driving path, considering factors such as obstacle avoidance, time efficiency, and energy conservation. The planned path is then relayed to the control module, which fine-tunes the vehicle's steering, acceleration, and braking to ensure precise adherence to the planned trajectory. These four components are interconnected and interdependent, working seamlessly together to guarantee the stable and reliable operation of autonomous cars.

Autonomous car technology traces its origins back to the 1970s, when the U.S. Defense Advanced Research Projects Agency (DARPA) funded the Automated Land Vehicle Research Project.(5) In the development of autonomous technology, the primary focus is on path planning and control algorithms, which enable autonomous vehicles to navigate roads, overcome various obstacles, and ensure safe driving. Path planning methods for autonomous cars can be categorized into four main types: search-based planning methods, artificial potential field (APF) methods, rapidly exploring random tree (RRT) algorithms, and discrete optimization-based planning methods.(6)

The RRT algorithm is a fundamental global pathplanning method. Its core principle involves starting at an initial point and incrementally expanding through the environment by random sampling until the goal point is reached. Regardless of the complexity of the environment, this algorithm reliably identifies a viable path. However, its convergence rate tends to be relatively low.(5) To accelerate the convergence of the RRT algorithm, techniques such as gradually adjusting the sampling region and increasing the sampling rate can be employed. These modifications not only enhance the convergence speed but also enable the algorithm to discover suitable paths more efficiently.(7,8) Nevertheless, the increased randomness introduced by these adjustments can lead to excessive redundancy in the form of unnecessary branches during the sampling process.(6) The probabilistic roadmap (PRM) stochastic algorithm is an efficient pathplanning method, particularly wellsuited for scenarios requiring obstacle avoidance.(9,10) To address the optimal pathplanning problem, researchers have proposed asymptotically optimal motion planners,(11–13) which significantly reduce the computational cost of generating the roadmap in the PRM algorithm. While traditional PRM and RRT algorithms can effectively solve optimal obstacle-avoidance trajectory planning problems, they are less suitable for scenarios with specialized optimization objectives.(14) Local path planning, on the other hand, involves continuously collecting information about the surrounding environment using the vehicle's sensors to design a collision-free, optimal path from the current position to the target destination.

Commonly used pathplanning algorithms(15–17) include the artificial potential field method, the timed elastic band algorithm, and the dynamic window approach. Both global and local pathplanning algorithms have inherent limitations. To enhance their performance, many researchers are actively exploring advancements in pathplanning techniques, achieving notable progress by improving and integrating traditional algorithms. For instance, Tang *et al.*(18) proposed an improved method for filtering redundant nodes in the A* algorithm and optimized the path's folded sections using cubic spline curves. This approach reduced the number of nodes and produced smoother paths. However, this method is not suitable for path planning in environments with dynamic obstacles. Wu *et al.*(19) combined the APF algorithm with the RRT algorithm by incorporating the virtual force field from the APF algorithm into the search tree expansion process of the RRT algorithm. This integration improved the convergence speed and search efficiency of the RRT algorithm. Nonetheless, this approach is primarily effective in relatively simple experimental environments.

In the control of autonomous vehicles, commonly used methods include the proportional-integral-derivative (PID) control, tracking control, model predictive (MPC) control, optimal control, sliding mode control, prescan control, and fuzzy control.(20) Ma *et al*.(21) decoupled lateral and longitudinal control through path prescanning and tracking, using signal feedback for real-time deviation correction to achieve the precise lateral and longitudinal control of unmanned vehicles. However, challenges remain regarding system stability in complex road environments. Shakouri *et al*.(22) proposed an adaptive cruise control method by comparing gain-scheduling proportional-integral and quadratic-regulator controllers to enable effective speed and distance tracking in various traffic scenarios. Similarly, Alcala *et al*.(23) introduced an autonomous vehicle control strategy that combines Lyapunov theory with the linear quadratic regulator (LQR), optimizing controller parameters to ensure vehicle stability. Mekala *et al*.(24) proposed another approach integrating Lyapunov theory and LQR with a MPC controller. This strategy leverages LiDAR data to achieve smooth and safe speed control, addressing longitudinal vehicle motion while considering obstacles, lead vehicles, and pedestrians.

To address the challenges identified in the above studies, we propose a novel methodology for obstacle avoidance control and path planning, with integrated verification, for autonomous vehicles using sensor data.

The path planning algorithm comprises two key components:

(1) Global Path Planning: We introduce a cost evaluation function that incorporates directional guidance to enhance the efficiency and accuracy of the path search process.

(2) Local Path Planning: We propose a method based on fifth-degree polynomial interpolation curves, which significantly improves the obstacle avoidance capability of autonomous vehicles on local paths.

The control methodology includes the following:

(1) a longitudinal control strategy, utilizing a position-velocity two-layer PID controller, and

(2) a lateral control strategy, employing a feedforward LQR for precise transverse motion control.

This integrated approach aims to improve both the safety and efficiency of autonomous vehicle navigation.

2. Theoretical Foundation

2.1 Vehicle motion model

The parameters of the adopted vehicle dynamics are defined in Table 1, and the coordinate system employed is illustrated in Fig. 2. The vehicle's motion model, considering two degrees of freedom, which describe the relationship between lateral acceleration, transverse angular acceleration, and tire rotation angle, is expressed as follows.

Table 1
Parameters of vehicle dynamics model.

Parameter	Symbol	Parameter	Symbol
Mass	m	Force on front, rear tire along x-axis	F_{xf}, F_{xr}
Lateral velocity	v	Force on front, rear tire along y-axis	F_{yf}, F_{yr}
Longitudinal velocity	u	Front, rear tire bias angle	α_f, α_r
Swing angle	φ	Front, rear tire bias stiffness	C_f, C_r
Swing angular velocity	$\dot{\varphi}$	Front, rear axis distance to mass center	L_f, L_r
Moment of inertia	I	Distance between front and rear tire center	L
Front wheel angle	α	Center-of-mass lateral deflection	β

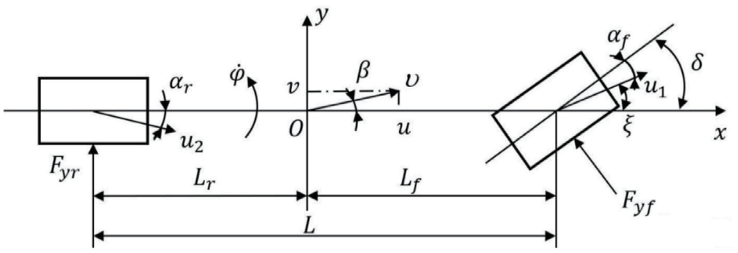


Fig. 2. Coordinate system of two-degree-of-freedom vehicle motion.

$$\frac{d}{dt}\begin{bmatrix} y \\ \dot{y} \\ \varphi \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & \dfrac{C_f + C_r}{mu} & 0 & \dfrac{L_f C_f - L_r C_r}{mu} \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{L_f C_f - L_r C_r}{I_c u} & 0 & \dfrac{L_f^2 C_f + L_r^2 C_r}{I_c u} \end{bmatrix} \begin{bmatrix} y \\ \dot{y} \\ \varphi \\ \dot{\varphi} \end{bmatrix} + \begin{bmatrix} 0 \\ -\dfrac{C_f}{m} \\ 0 \\ -\dfrac{L_f C_f}{I_z} \end{bmatrix} \delta \tag{1}$$

2.2 Autonomous vehicle path planning algorithm

2.2.1 Global path planning based on A* algorithm

2.2.1.1 Principle of A* algorithm

The A* algorithm(25) is a global path planning method based on sampling search, widely applied in the path planning of autonomous vehicles. It combines the systematic approach of Dijkstra's algorithm with the directional guidance of the Greedy Best-First Search (GBFS) algorithm. The A* algorithm is capable of finding the shortest path while incorporating directional guidance, ensuring both efficiency and accuracy in path planning.

The core of the A* algorithm lies in its cost evaluation function, as shown in Eq. (2).

$$f(n)=g(n)+h(n), \tag{2}$$

where $g(n)$ is the real cost function and $h(n)$ is the predictive cost function.

We implemented the A* algorithm in Visual Studio Code by following its core principles. First, we constructed a 70 × 70 map, added obstacles, and defined the starting and ending points. Next, we evaluated the cost function for the surrounding paths, using the Euclidean distance to estimate the distance between nodes. The node with the smallest cost function was selected as the current node, and the process continued iteratively until the target point was reached. Once the path was identified, the shortest route from the starting point to the endpoint was reconstructed by backtracking through the nodes. The results are shown in Fig. 3.

2.2.1.2 Optimization of A* algorithm

We dynamically adjust the value of the weighting function on the basis of the distance from the target position. The updated cost function is presented in Eqs. (3) and (4).

$$f(n) = g(n) + w(n)h(n), \tag{3}$$

$$w(n) = 1 + \frac{d_c}{d}, \tag{4}$$

where $w(n)$ is the weighting function, d_c is the distance from the current node to the target node, and d is the distance from the start node to the target node.

After optimization using the improved A* algorithm, we obtained an updated map, as shown in Fig. 4. It is evident that the number of search points (yellow crosses) across the entire map has been significantly reduced compared with that shown in Fig. 3. This reduction in search area

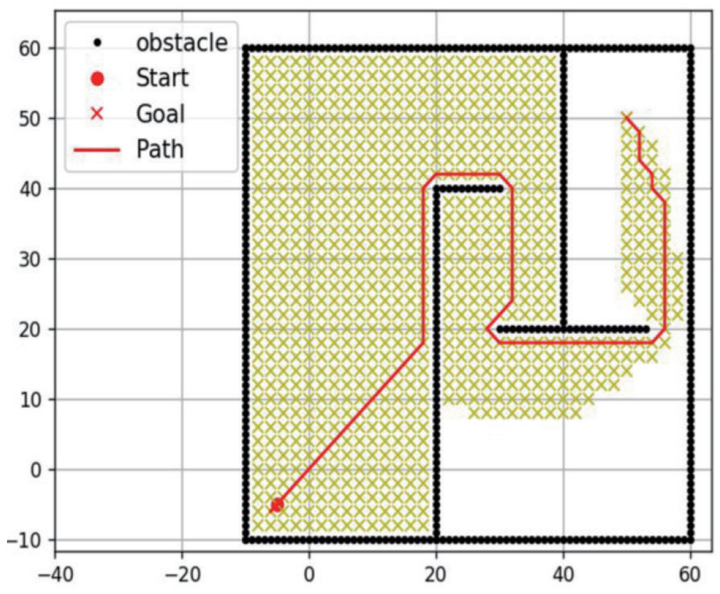


Fig. 3. (Color online) Path planning of A* algorithm.

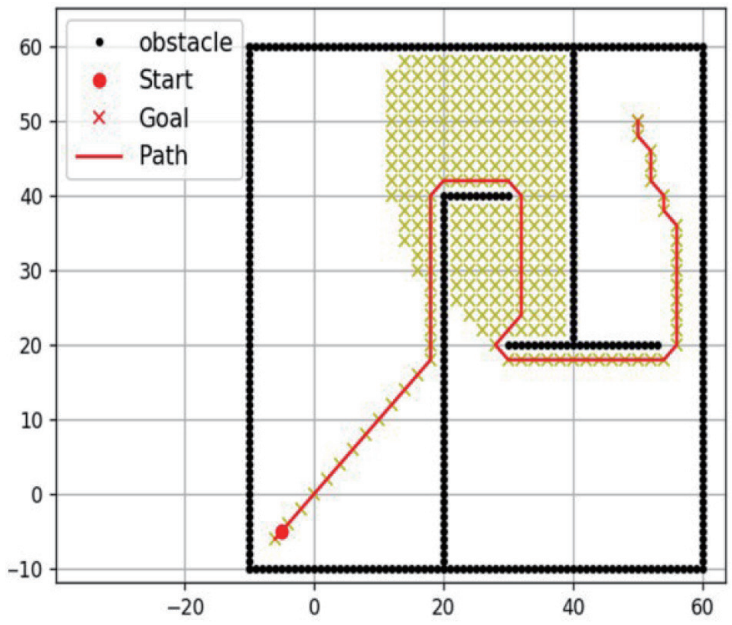


Fig. 4. (Color online) Path planning using improved A* algorithm.

indicates a decrease in unnecessary computational overhead and a substantial improvement in search efficiency.

The paths are further refined using the cubic spline technique with Hermite interpolation functions.(26) The resultant path is shown in Fig. 5. By cubic spline interpolation, a smoother and more continuous path is achieved, effectively reducing the number of sharp turns and folds in the original path. The smoothed path is better suited for seamless and efficient vehicle navigation on a real road.

2.2.2 Local path planning using fifth-degree polynomial interpolation curves

The matrix representations of the fifth-degree polynomial interpolation curves are provided in Eqs. (5) and (6).

$$\boldsymbol{X} = \begin{bmatrix} x(0) \\ \dot{x}(0) \\ \ddot{x}(0) \\ x(T) \\ \dot{x}(T) \\ \ddot{x}(T) \end{bmatrix} = \begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1 \\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0 \\ 20t_0^3 & 12t_0^2 & 2t_0 & 2 & 0 & 0 \\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1 \\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0 \\ 20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} a_5 \\ a_4 \\ a_3 \\ a_2 \\ a_1 \\ a_0 \end{bmatrix} = \boldsymbol{TA} \tag{5}$$

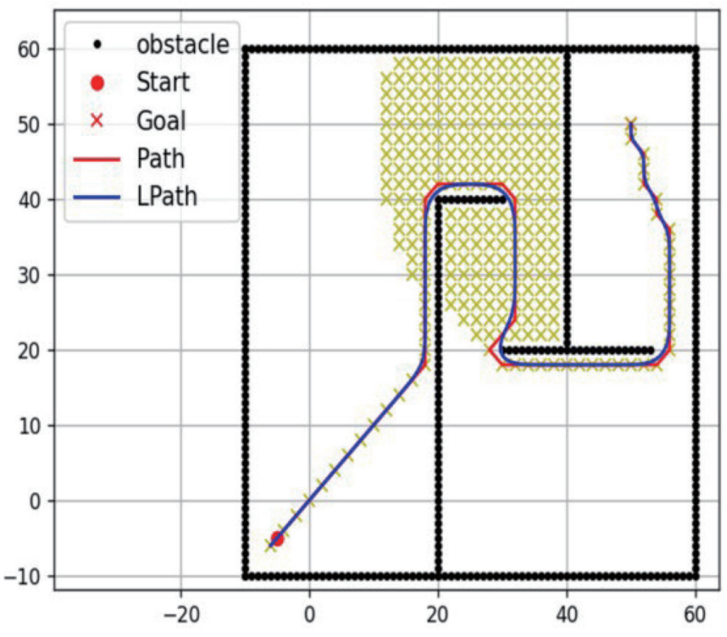
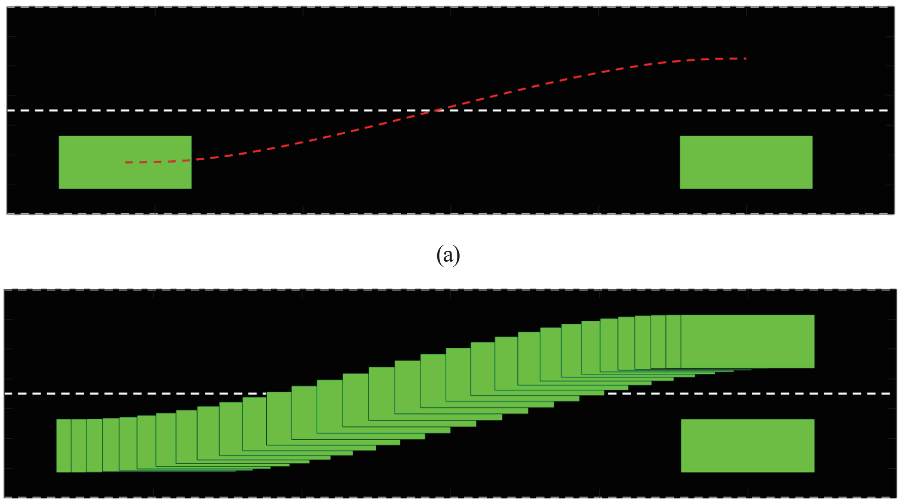


Fig. 5. (Color online) Path planning of A* algorithm with cubic spline smoothing.

$$\boldsymbol{Y}=\begin{bmatrix} y(0)\\ \dot{y}(0)\\ \ddot{y}(0)\\ y(T)\\ \dot{y}(T)\\ \ddot{y}(T)\end{bmatrix}=\begin{bmatrix} t_0^5 & t_0^4 & t_0^3 & t_0^2 & t_0 & 1\\ 5t_0^4 & 4t_0^3 & 3t_0^2 & 2t_0 & 1 & 0\\ 20t_0^3 & 12t_0^2 & 2t_0 & 2 & 0 & 0\\ t_1^5 & t_1^4 & t_1^3 & t_1^2 & t_1 & 1\\ 5t_1^4 & 4t_1^3 & 3t_1^2 & 2t_1 & 1 & 0\\ 20t_1^3 & 12t_1^2 & 6t_1 & 2 & 0 & 0\end{bmatrix}\begin{bmatrix} b_5\\ b_4\\ b_3\\ b_2\\ b_1\\ b_0\end{bmatrix}=\boldsymbol{TB} \tag{6}$$

Here, $\boldsymbol{X}$ represents the horizontal start and end point coordinate matrix, $\boldsymbol{Y}$ represents the vertical start and end point coordinate matrix, and $\boldsymbol{T}$ is the coordinate transformation matrix. Using the position coordinate data $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{T}$ obtained from various positioning sensors (16-line LiDAR, mmWave radar, and Ultrasonic radar), along with calculations based on Eqs. (5) and (6), we can determine the vehicle's trajectory. Furthermore, by analyzing the position data from these sensors, the autonomous vehicle's control system can calculate the required speed (as the change in position over time) and acceleration (as the change in speed over time) at each point along the trajectory.

On the basis of a local pathplanning scheme utilizing fifth-degree polynomial interpolation curves, we designed a vehicle bypass scenario, as illustrated in Fig. 6. In this scenario, the front vehicle remains stationary while the rear vehicle must navigate around it following the planned path. During the maneuver, the control points include the edge position of the front vehicle and points near the current position of the rear vehicle. By substituting the position coordinates of these control points into Eqs. (5) and (6), we determine the theoretical position of the vehicle at



(a)

(b)

Fig. 6. (Color online) Car obstacle-avoidance tracks: (a) local path planning track and (b) local path planning track under control.

each moment. This process results in a vehicle trajectory, as shown in Fig. 6(a). Under control, the vehicle successfully follows the planned obstacle-avoidance path, as depicted in Fig. 6(b). The trajectory planning achieved through the fifth-degree polynomial interpolation curve effectively fulfills the vehicle's bypassing task.

2.3 Autonomous vehicle travel control algorithm

2.3.1 Longitudinal control

PID control is widely used in autonomous vehicles.[20] The formula of PID control is

$$\mathrm{Y}(t) = K_p e(t) + K_i \sum_{i=1}^{n} e(t) + K_d [e(t) - e(t-1)], \tag{7}$$

where $Y(t)$ is the output at time t, K_p is the proportional coefficient, K_i is the integral coefficient, K_d is the differential coefficient, and $e(t)$ is the error between the real-time detected data by sensors and the desired theoretical data.

In this study, we employed a two-layer PID controller design consisting of a position-velocity closed-loop controller and an open-loop controller, which utilizes a throttle-brake calibration table to achieve the precise control of the vehicle's longitudinal motion. As illustrated in Fig. 7, this hybrid controller enables the autonomous vehicle to determine its steering position, velocity, and acceleration, ensuring that it follows the planned trajectory. With onboard position sensors (16-line LiDAR, mmWave radar, and Ultrasonic radar), a velocity compensation signal is first generated by the position controller on the basis of the deviation between the desired and actual

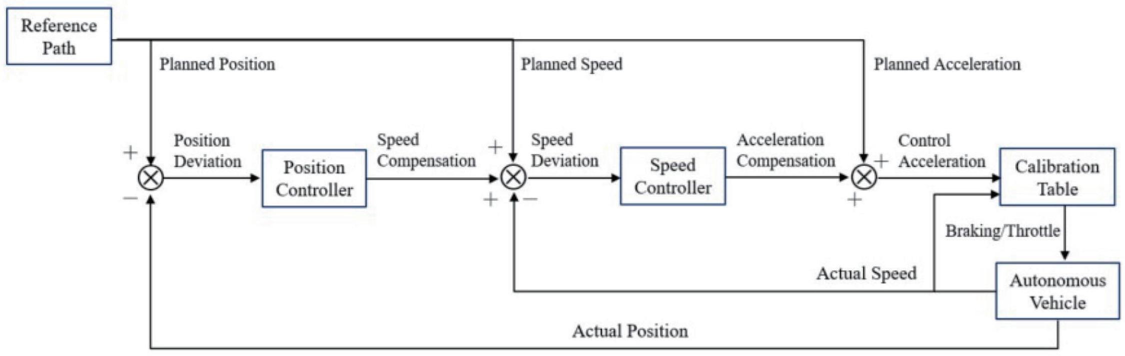


Fig. 7. Longitudinal motion control using two-layer PID controller.

positions. This signal is then transmitted to the velocity controller, which adjusts itself according to the speed difference.

The outputs of the two controllers jointly determine the required acceleration compensation, which is then adjusted to match the desired acceleration, producing the final output acceleration. This output acceleration is calculated via a calibration table to generate a control signal for the brakes and throttle. The control signal is ultimately transmitted to the autonomous vehicle to regulate its brakes and throttle for longitudinal motion control. Simultaneously, the vehicle's actual speed and position are fed back to the position and velocity controllers, enabling the system to adapt the control strategy in real time to ensure that the vehicle remains on the planned path.

2.3.2 Lateral control

We employ the LQR controller for the lateral motion control of the vehicle with the control strategy illustrated in Fig. 8. On the basis of the previously mentioned vehicle motion model, the state-space equations governing the vehicle's lateral motion can be derived using Eq. (8).

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} = &\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\dfrac{C_f + C_r}{mu} & \dfrac{C_f + C_r}{u} & -\dfrac{L_f C_f - L_r C_r}{mu} \\ 0 & 0 & 0 & 1 \\ 0 & -\dfrac{L_f C_f - L_r C_r}{I_z u} & \dfrac{L_f C_f - L_r C_r}{I_z} & -\dfrac{L_f^2 C_f + L_r^2 C_r}{I_z u} \end{bmatrix}\begin{bmatrix} e_1 \\ \dot{e}_1 \\ e_2 \\ \dot{e}_2 \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ \dfrac{C_f}{m} \\ 0 \\ \dfrac{L_f C_f}{I_z} \end{bmatrix}\delta + \begin{bmatrix} 0 \\ -\dfrac{L_f C_f - L_r C_r}{mu} - u \\ 0 \\ -\dfrac{L_f^2 C_f + L_r^2 C_r}{I_z u} \end{bmatrix}\dot{\varphi}_{des},\end{aligned} \tag{8}$$

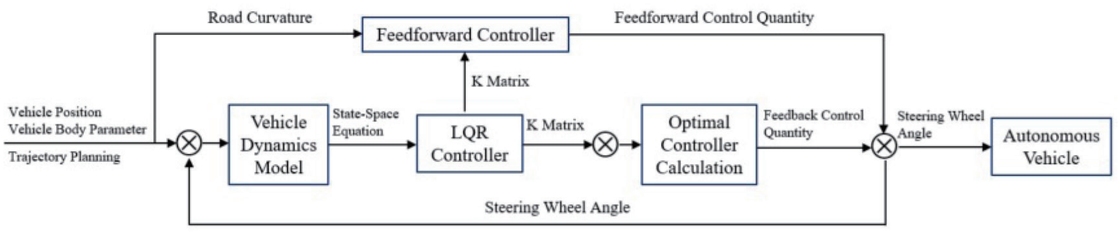


Fig. 8. Lateral motion control using LQR controller.

where e_1 is the lateral error, e_2 is the heading angle error, δ is the wheel angle, and φ_{des} is the road curvature.

The LQR controller allows us to determine the vehicle's control conditions using the state feedback matrix data, enabling the state vector to converge quickly and optimizing the vehicle's performance. The LQR controller is cost-effective, easy to tune, and capable of improving the stability of vehicle motion. Compared with the traditional PID control algorithm, the LQR controller offers superior robustness and adaptability in complex road-steering scenarios.

3. Simulation of Lateral and Longitudinal Motion Control

We utilize CarSim simulation software (developed by Mechanical Simulation Corporation, USA, 2024) to simulate the lateral and longitudinal motion control of autonomous vehicles. Its outstanding capabilities in analyzing and predicting vehicle behavior, specifically in handling stability, braking efficiency, smoothness, power performance, and fuel economy, make it a vital auxiliary tool for the research and development of modern automotive control systems.

3.1 Longitudinal motion control simulation

The primary function of the longitudinal controller is to regulate the acceleration and deceleration of the vehicle. In practical applications, the direct control of the vehicle's speed is not possible; instead, adjustments are made to the throttle opening angle and braking pressure. Using the Simulink simulation software tool (developed by The MathWorks Inc., USA), we designed a two-layer position-velocity PID controller, as illustrated in Fig. 9. With this PID controller, the vehicle's speed and acceleration are first input, after which the calibration table is used to determine the optimal throttle opening angle and braking pressure. This process ultimately achieves the desired vehicle acceleration and deceleration.

With our designed position-velocity two-layer PID controller, the resulting position, velocity, and acceleration of the unmanned vehicle's movement are shown in Figs. 10(a)–10(c), respectively. The results demonstrate that the PID controller accurately guides the vehicle to follow the preset conditions. Even during acceleration and deceleration, the vehicle's motion trajectory remains smooth and aligns well with expectations.

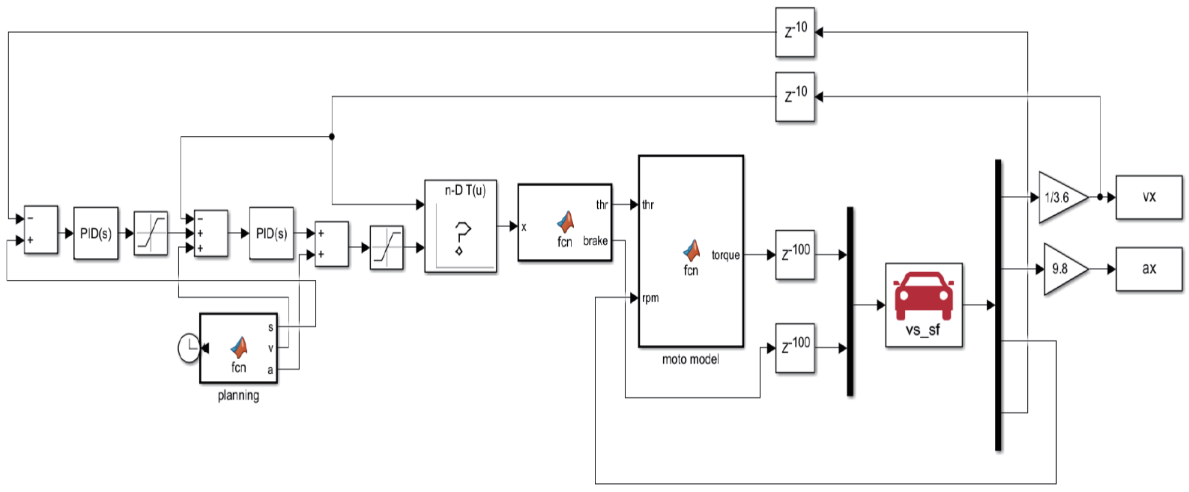


Fig. 9. (Color online) Position and velocity obtained using two-layer PID controller.

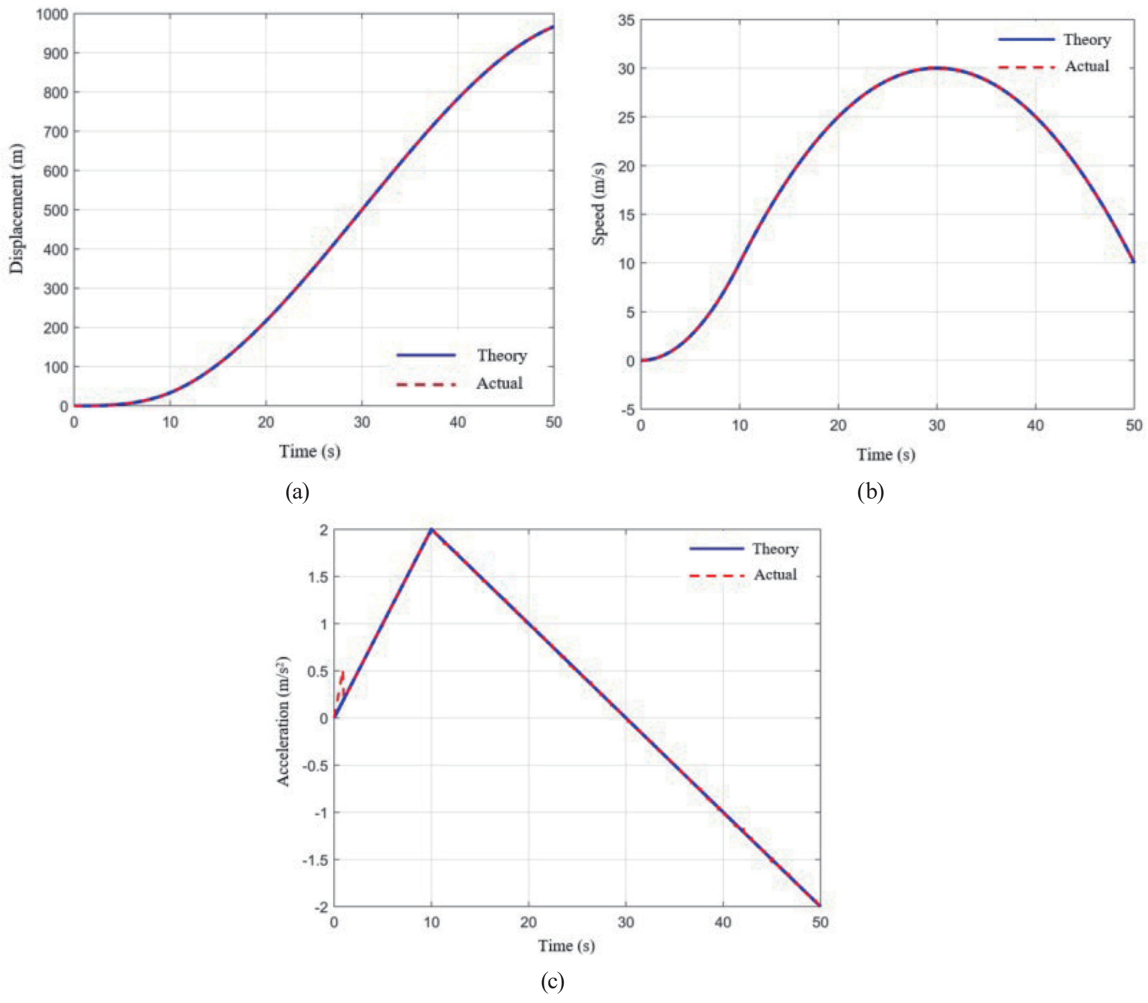


Fig. 10. (Color online) Variations in position, velocity, and acceleration of the vehicle using PID controller.

3.2 Lateral motion

We first accomplish all the relevant settings for vehicle lateral steering control in CarSim software and transfer the configured vehicle model to Simulink. Following the feedforward-LQR control principle described in the previous section, we then develop the feedforward-LQR lateral controller in Simulink, as illustrated in Fig. 11.

By fine-tuning the parameters of the feedforward-LQR lateral controller, we achieve a smoother time-dependent curve of the vehicle's steering wheel angle, as shown in Fig. 12.

Moreover, we obtain the final motion control results of the vehicle, as shown in Fig. 13. Figures 13(a)–13(d) respectively depict the vehicle's straight-line motion, left-turn motion, right-turn motion, and U-turn motion. All driving conditions meet the expected performance using the feedforward-LQR lateral controller.

4. Vehicle Driving Control Simulation and Road-driving Demonstration

4.1 Vehicle driving control simulation

We use Apollo's Dreamview software platform[27] for vehicle driving control simulation. This software is an autonomous visualization and interaction platform that enables users to monitor multiple functional modules of the self-driving vehicle in real time, such as sensing,

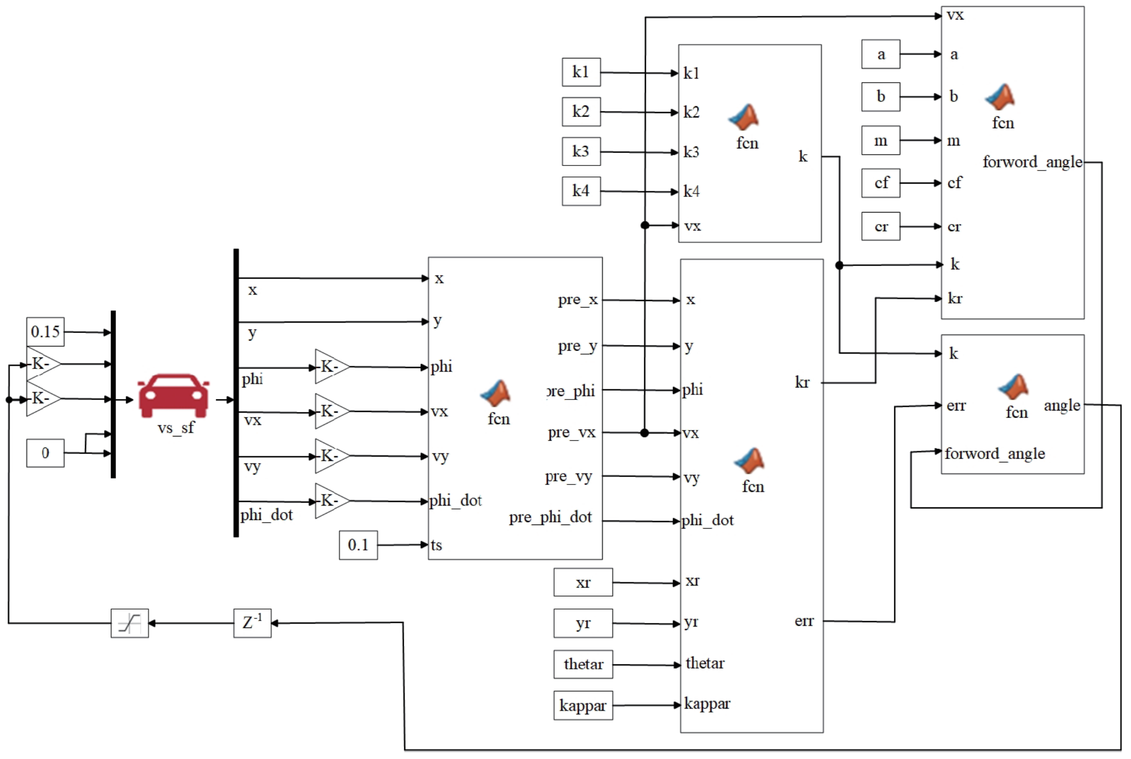


Fig. 11. (Color online) Feedforward-LQR lateral controller.

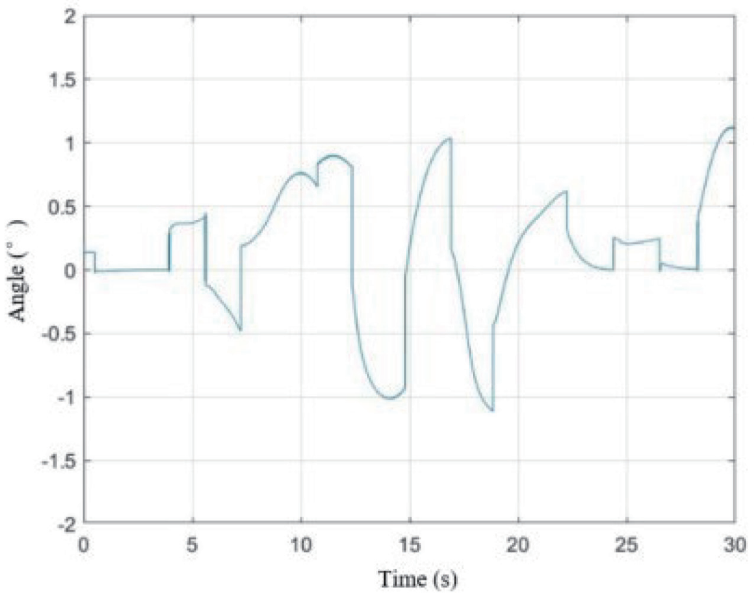


Fig. 12. (Color online) Time variation of the vehicle's steering wheel angle.

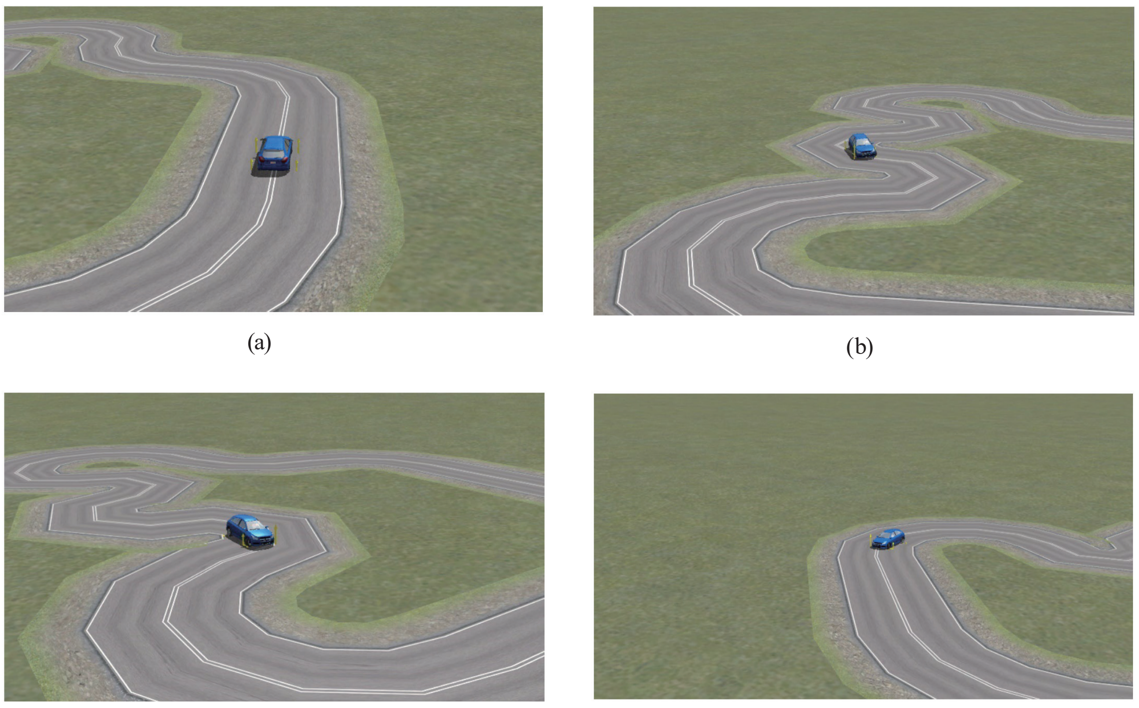


Fig. 13. (Color online) Vehicle's lateral motion: (a) straight-line motion, (b) left-turn motion, (c) right-turn motion, and (d) U-turn motion.

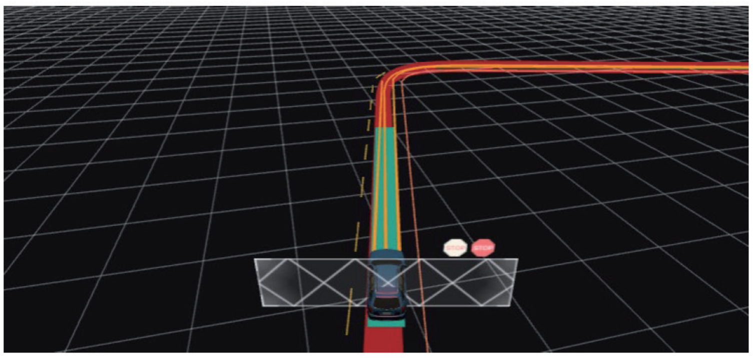
localization, planning, and control. It also displays the vehicle's state, sensor data, and dynamic trajectory. Dreamview supports the simulation of multiple driving scenarios, allowing developers to validate algorithms without the need for real-vehicle testing conditions.

4.1.1 Vehicle cruise driving simulation

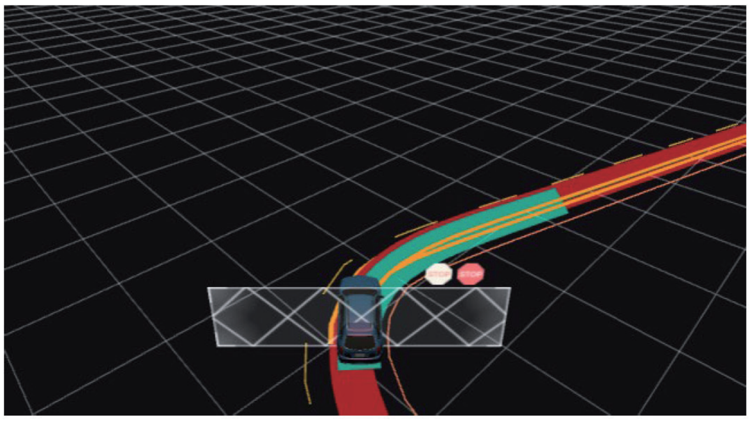
In the vehicle cruise driving simulation, we need to consider how to ensure that the autonomous car can follow the predetermined path at the preset stable driving speed while simultaneously monitoring changes in the surrounding environment to adjust the car's speed in a timely manner or make an emergency stop if necessary. By using the Dreamview platform, we have performed both straight-line and curve path cruise simulations, as shown in Figs. 14(a) and 14(b). It is observed that the autonomous vehicle exhibits excellent pathtracking ability while maintaining speed along the preset route, and it can drive stably even during transitions from straight to curved paths.

4.1.2 Simulation of vehicle driving for pedestrian avoidance

The simulation results of the vehicle traveling to avoid pedestrians, including detecting pedestrians, slowing down to yield, and replanning to accelerate forward, using Dreamview are shown in Figs. 15(a)–15(c), respectively.



(a)



(b)

Fig. 14. (Color online) Vehicle cruise driving simulation: (a) straight-line path cruise and (b) curve path cruise.

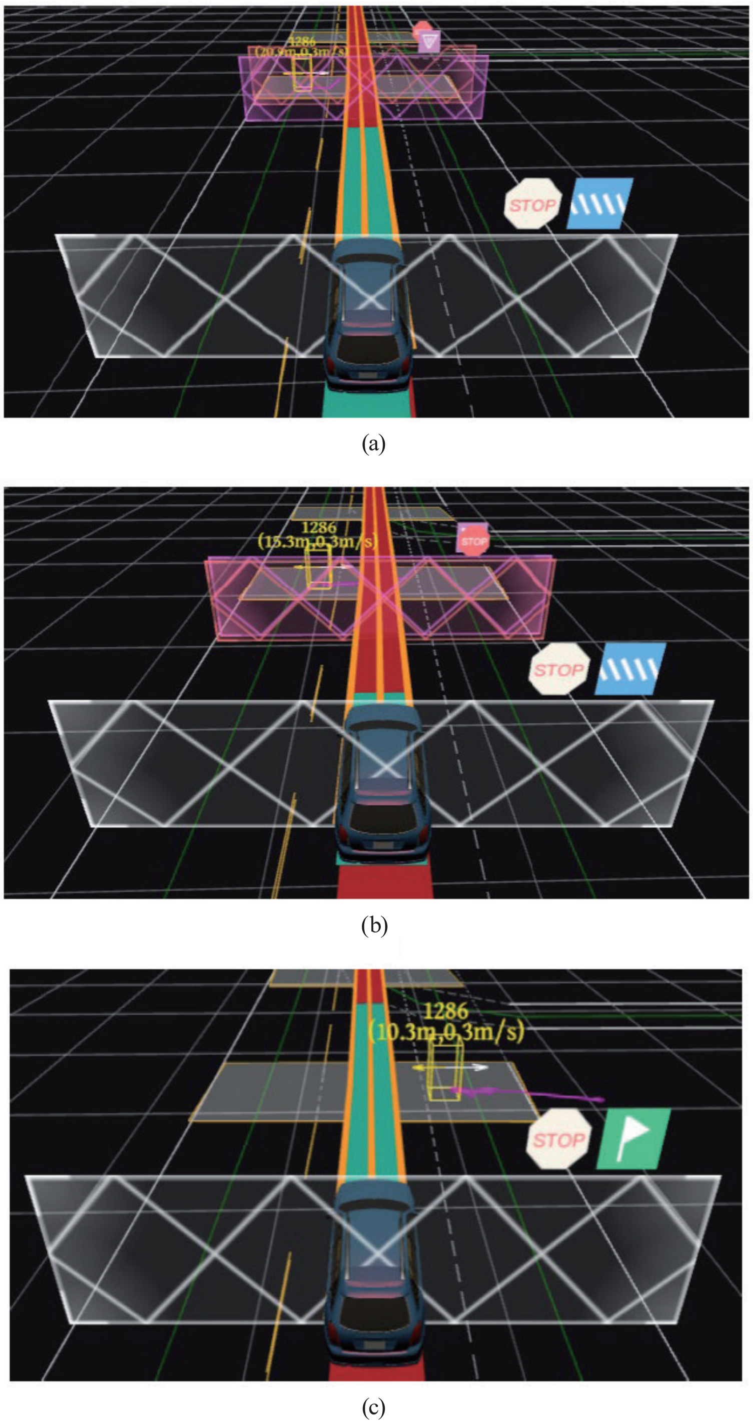
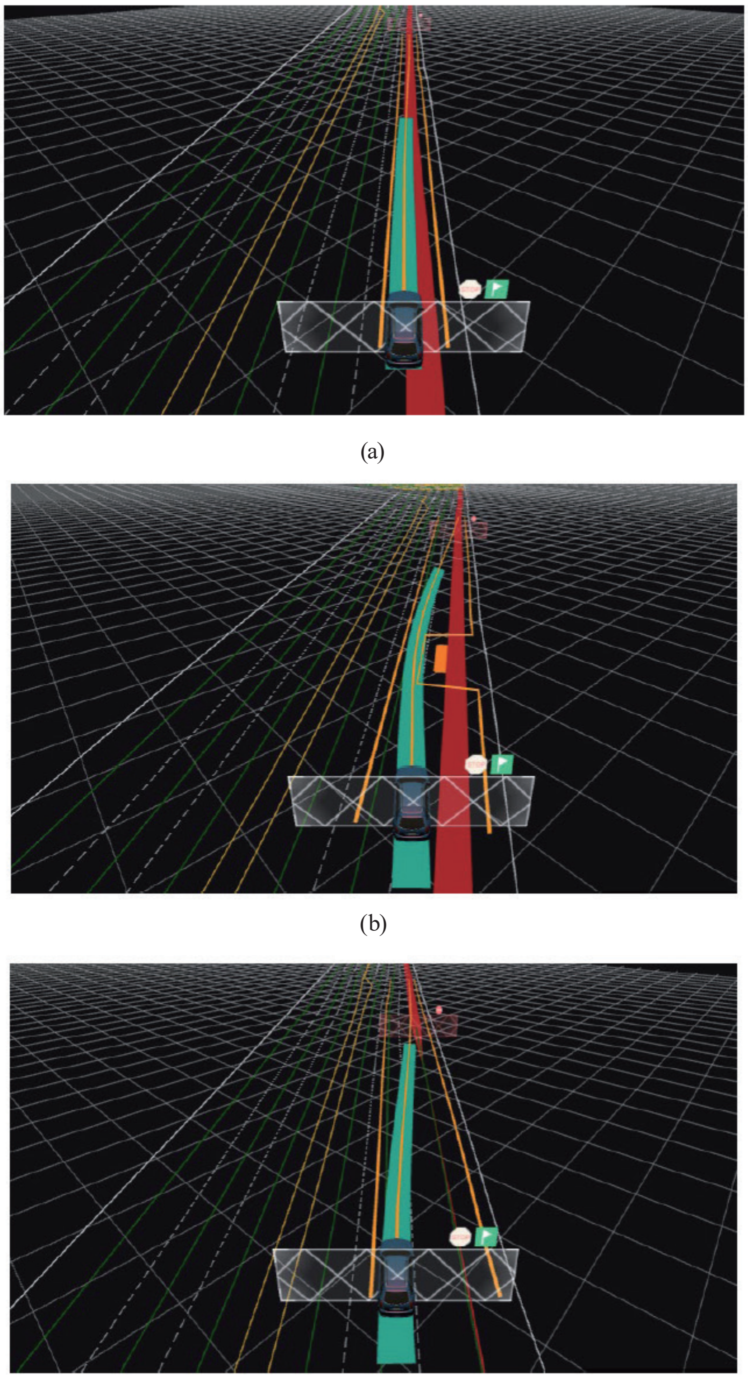


Fig. 15. (Color online) Simulation of vehicle traveling for pedestrian avoidance: (a) detecting pedestrians, (b) slowing down to yield, and (c) replanning to accelerate forward.

4.1.3 Vehicle travel with bypass simulation

The simulation of vehicle travel using a lane-borrowing bypass, built by using Dreamview, is shown in Fig. 16. The autonomous vehicle begins traveling on the road according to the preset



(a)

(b)

(c)

Fig. 16. (Color online) Simulation of vehicle traveling on a detour: (a) going straight, (b) avoiding an obstacle, and (c) returning to the original lane.

path and speed. When the vehicle's sensors detect an obstacle ahead, the car performs a lane-borrowing detour [Fig. 16(a)], traveling to another lane to bypass the obstacle [Fig. 16(b)], and then looks for an appropriate time to return to the original lane and continues along the initial route [Fig. 16(c)].

4.2 Verification with real-vehicle controlled driving test

We conducted two scenario-based validation tests for path planning and vehicle control using the Baidu Apollo Standard-S unmanned driving development kit.(28) These tests involved an autonomous vehicle equipped with the Apollo Standard-S software platform, as shown in Fig. 17. The platform integrates cameras, LiDAR, mmWave radar, GPS/IMU, and other sensors to enable the comprehensive perception of the surrounding environment. Building on this platform, we carried out the secondary development of autonomous driving technologies. Our contributions include developing a dynamic vehicle model, designing path planning algorithms, and creating control strategies. These developments were validated through both simulation testing and real-world driving experiments in scenarios such as constant-speed cruising, pedestrian avoidance, and lane-changing maneuvers.

4.2.1 Real-vehicle cruise driving test

The experimental vehicle is first positioned at the planned location on the campus road. The recorder in the vehicle is activated to record the speed and trajectory data during driving. Next, the remote controller is switched to the automatic driving mode, and the vehicle begins to travel on the road. The results are shown in Fig. 18. It can be observed that the autonomous vehicle moves smoothly along the established trajectory, maintaining stable steering throughout the entire journey and consistently avoiding obstacles in a timely manner.

4.2.2 Test of real-vehicle driving for pedestrian avoidance

The result of the real-vehicle cruise driving is shown in Fig. 19. First, the autonomous vehicle moves forward at a normal speed. Suddenly, a pedestrian steps into the lane from the curb

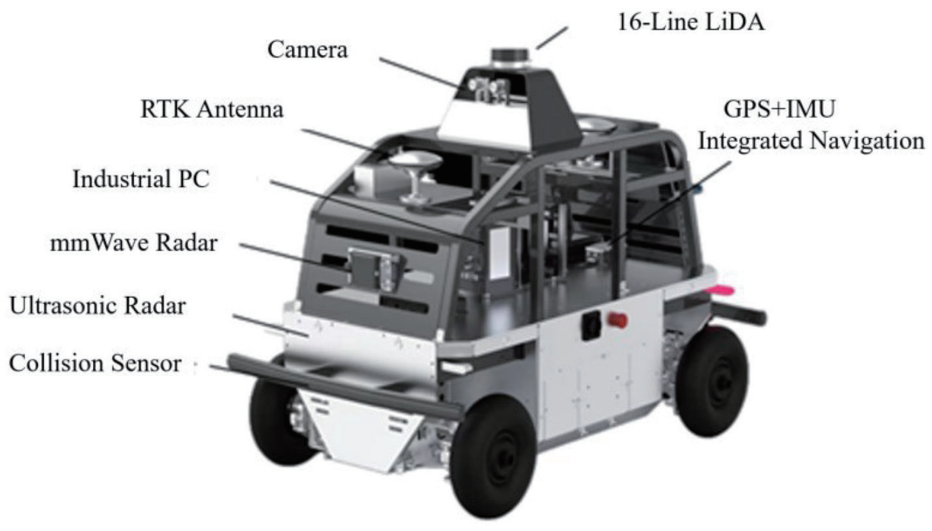


Fig. 17. (Color online) Autonomous vehicle with Apollo Standard S platform.

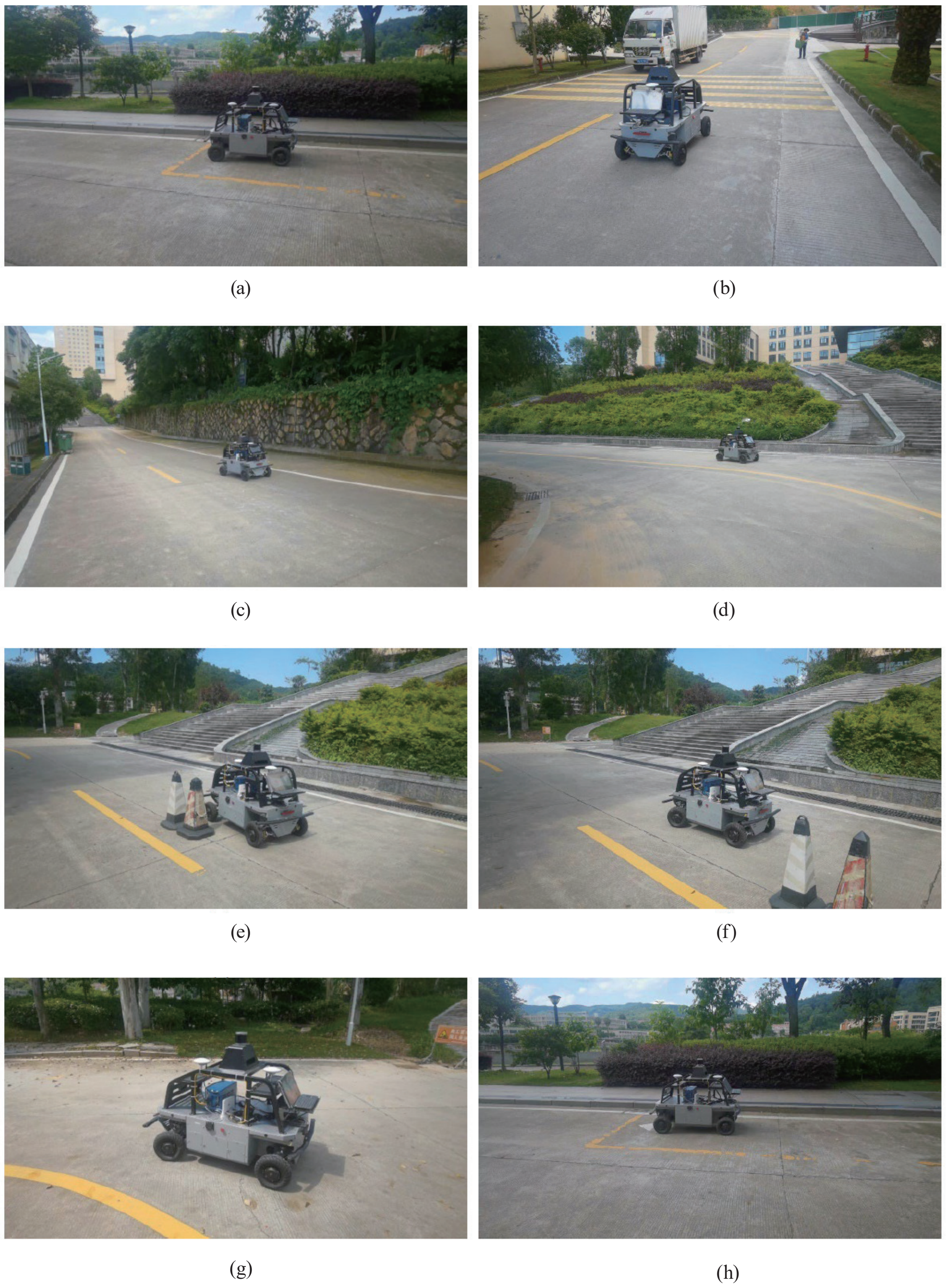


Fig. 18. (Color online) Test of real-vehicle cruise driving: (a) getting started, (b) straight-line deceleration, (c) straight-line acceleration, (d) turning left, (e) obstacle avoidance, (f) returning, (g) turning left, and (h) stopping.

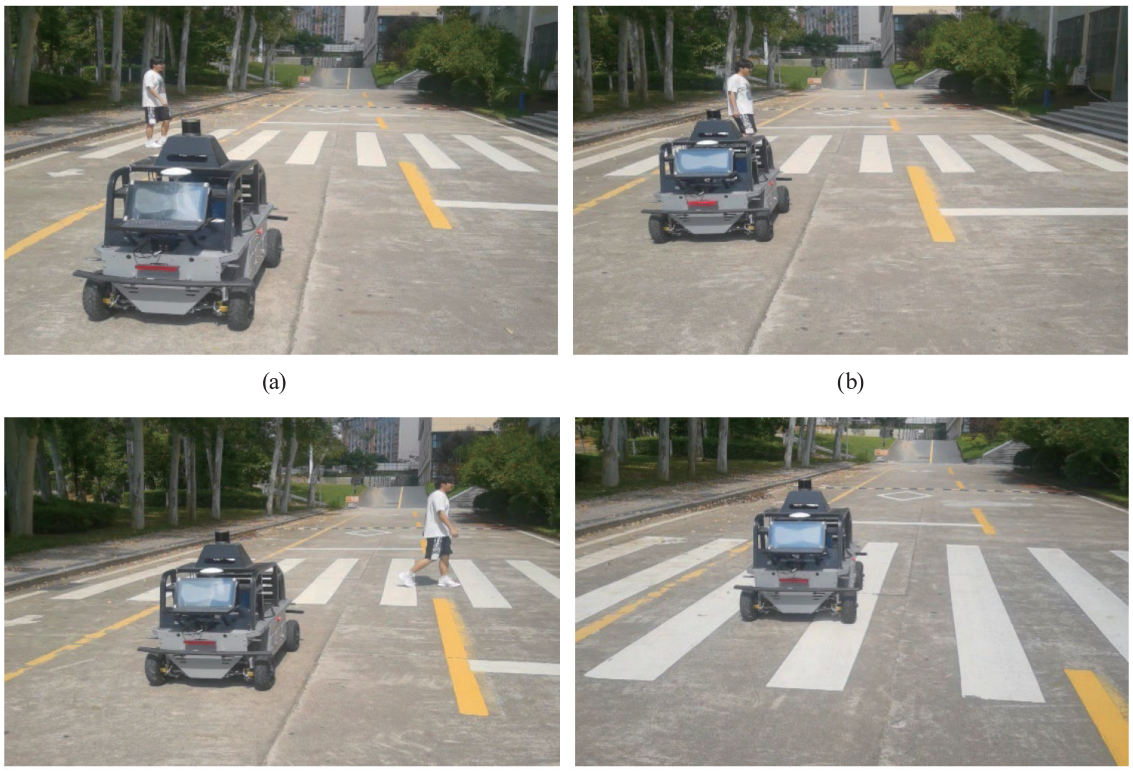


Fig. 19. (Color online) Test of real-vehicle driving for pedestrian avoidance: (a) detection of pedestrian, (b) braking, (c) straight-line acceleration, and (d) passing through.

without warning. The vehicle promptly decelerates and stops in time to avoid the pedestrian's movement, based on the pedestrian's behavior [Figs. 19(a) and 19(b)]. After the pedestrian passes, the vehicle gradually accelerates [Figs. 19(c) and 19(d)].

5. Conclusions

We proposed a novel real-time path planning and vehicle control strategy for autonomous vehicles navigating obstacles. In path planning, we developed an optimized A* algorithm with dynamic weighting and cubic spline curve smoothing, while a fifth-degree polynomial interpolation curve was used for local path planning. This approach not only enhanced the efficiency of path searching and smoothing but also allowed for flexible responses to obstacles and real-time obstacle avoidance along the local path. For autonomous vehicle control, we designed a position-velocity two-layer PID controller for longitudinal control and a feedforward-LQR controller for lateral control. The proposed strategy improved the path search efficiency by 20%, increased the path smoothness by 3%, and boosted the obstacle avoidance success rate by 5%.

The feasibility and stability of these control strategies were verified through joint simulations using CarSim and Simulink software. In the simulation analysis, scenarios were modeled via the

Apollo software platform, including constant-speed cruising, pedestrian avoidance, and lane borrowing for obstacle bypassing. The simulation results demonstrated that the proposed methodology effectively recognizes environmental changes and performs timely path adjustments and speed control.

Moreover, a real vehicle with the Apollo Standard S platform was used on the campus road. Under the detection of various sensors, the test autonomous car conducted experiments such as constant-speed cruising and pedestrian avoidance on the basis of the proposed path planning and control strategies. The test results indicated that the vehicle safely followed the predefined path and avoided obstacles in a timely manner, meeting the traveling requirements in a campus road environment.

Acknowledgments

This work was supported by the Special Project of the Central Government to Guide Local Scientific and Technological Development under Grant no. 2021L3046, the Project of the Department of Science and Technology of Fujian Province, China (Grant nos. 2021G02013, 2020H0049, and 2021H0060), and in part by Sanming University of Fujian Province, China (Grant nos. 19YG05 and 19YG04).

References

1 J. Fu, G. Sun, W. Yao, and L. Wu: Intell. Transp. Syst. **23** (2022) 24008. https://doi.org/10.1109/TITS.2022.3195521
2 R. Chai, A. Tsourdos, A. Savvaris, S. Chai, Y. Xia, and C. L. Philip Chen: IEEE Trans. Cybern. **51** (2020) 4035. https://doi.org/10.1109/TCYB.2020.2973748
3 Y. Hu, Y. Yao, Q. Ren, and X. Zhou: Comput. Syst. **102** (2020) 762. https://doi.org/10.1016/j.future.2019.09.030
4 Z. Zuo: Automatica **54** (2015) 305. https://doi.org/10.1016/j.automatica.2015.01.021
5 H. Ma, W. Pei, and Q. Zhang: Mathematics **10** (2022) 2555. https://doi.org/10.3390/math10152555
6 X. Wang, X. Luo, B. Han, Y. Chen, G. Liang, and K. Zheng: Appl. Sci. **10** (2020) 1381. https://doi.org/10.3390/app10041381
7 Z. Wu, Z. Meng, W. Zhao, and Z. Wu: Appl. Sci. **11** (2021) 11777. https://doi.org/10.3390/app112411777
8 S. M. Yang and Y. A. Lin: Sensors **21** (2021) 2244. https://doi.org/10.3390/s21062244
9 J. D. Marble and K. E. Bekris: IEEE Trans. Rob. **29** (2013) 432. https://doi.org/10.1109/TRO.2012.2234312
10 Z. Yao and K. Gupta: IEEE Trans. Rob. **27** (2011) 997. https://doi.org/10.1109/TRO.2011.2142470
11 R. Sandström, D. Uwacu, J. Denny, and N. M. Amato: IEEE Rob. Autom. Lett. **5** (2020) 6161. https://doi.org/10.1109/LRA.2020.3010487
12 J. Wang, W. Chi, C. Li, and M. Q. H. Meng: IEEE Trans. Autom. Sci. Eng. **19** (2022) 1859. https://doi.org/10.1109/TASE.2021.3130372
13 R. Wang, X. Zhang, Y. Fang, and B. Li: IEEE Trans. Syst., Man, Cybern. Syst. **52** (2022) 2073. https://doi.org/10.1109/TSMC.2020.3044347
14 J. Fu, W. Yao, G. Sun, H. Tian, and L. Wu: IEEE Trans. Intell. Veh. **8** (2023) 2415. https://doi.org/10.1109/TIV.2023.3237518
15 L. Keyu, L. Yonggen, and Z. Yanchi: 2020 5th Int. Conf. Commun. Image Signal Process. (CCISP), Chengdu, China (2020) 159. https://doi.org/10.1109/CCISP51026.2020.9273463
16 J. Wu, X. Ma, T. Peng, and H. Wang: Sensors **21** (2021) 8312. https://doi.org/10.3390/s21248312
17 X. Lai, D. Wu, J. H. Li, and H. Yu: Ind. Rob. **50** (2023) 186. https://doi.org/10.1108/IR-05-2022-0130
18 G. Tang, C. Tang, C. Claramunt, X. Hu, and P. Zhou: IEEE Access **9** (2021) 59196. https://doi.org/10.1109/ACCESS.2021.3070054
19 D. H. Wu, L. S. Wei, G. L. Wang, L. Tian, and G. Z. Dai: Appl. Sci. **12** (2022) 10905. https://doi.org/10.3390/app122110905

20 Y. Huang, W. Luo, and H. Lan: World Electr. Veh. J. **13** (2022) 55. https://doi.org/10.3390/wevj13040055
21 D. Ma, I. Boussaada, J. Chen, C. Bonnet, S. I. Niculescu, and J. Chen: Automatica **137** (2022) 110102. https://doi.org/10.1016/j.automatica.2021.110102
22 P. Shakouri, A. Ordys, D. S. Laila, and M. Askari: IFAC Proc. **44** (2011) 12964. https://doi.org/10.3182/20110828-6-IT-1002.02250
23 E. Alcala, V. Puig, J. Quevedo, T. Escobet, and R. Comasolivas: Control Eng. Pract. **73** (2018) 1. https://doi.org/10.1016/j.conengprac.2017.12.004
24 G. K. Mekala, N. R. Sarugari, and A. Chavan: 2020 IEEE Int. Conf. Innov. Techn. (INOCON), Bangluru, India (2020) 1. http://doi.org/10.1109/INOCON50539.2020.9298213
25 Y. Liu, Y. Wang, Z. Wen, and X. Yuan: 2022 7th Int. Conf. Intell. Inf. Biomed. Sci. (ICIIBMS), Nara, Japan (2022) 19. https://doi.org/10.1109/ICIIBMS55689.2022.9971587.
26 S. Amat, Z. Li, J. Ruiz-Álvarez, C. Solano, and J. C. Trillo: J. Sci. Comput. **95** (2023) 84. https://doi.org/10.1007/s10915-023-02191-9
27 S. Hess and D. Palma: J. Choice Modell. **32** (2019) 100170. https://doi.org/10.1016/j.jocm.2019.100170
28 Apollo Studio: https://apollo.baidu.com/community/apollo_d_kit (in Chinese).