E-ISSN 2586-9027
Homepage : *https://tci-thaijo.org/index.php/SciTechAsia*
Science & Technology Asia
Vol.30 No.4 October - December 2025
Page: [147-168]
Original research article

Research on Optimizing Control Signals for Single-Agent Navigation under Multiple Scenarios in Linear Motion

Chuanji Xu, Jian Qu*

Faculty of Engineering and Technology, Panyapiwat Institute of Management, Nonthaburi 11120, Thailand

Received 5 June 2024; Received in revised form 14 October 2025
Accepted 27 October 2025; Available online 17 December 2025

ABSTRACT

This study proposes a Bezier curve optimization method for enhancing the control signals and trajectory tracking performance of the Donkey Car, a 1:16 scale autonomous vehicle platform. The method employs second-order Bezier curves for throttle optimization and third-order Bezier curves for steering angle optimization. A novel loss function, Bezier Smoothing Loss (BSL), is introduced to simultaneously optimize control signal smoothness and trajectory tracking accuracy during neural network controller training.Experiments in three scenarios (left lane driving with obstacle avoidance, straight line driving, and straight driving with continuous obstacle avoidance) show that the proposed method significantly improves trajectory tracking accuracy (RMSE reduced by up to 19.6%), control signal smoothness (throttle change rate standard deviation decreased by 28.6%, steering angular velocity standard deviation reduced by 24.5%), and vehicle posture stability (yaw rate and pitch rate RMS values decreased by 7.9%). Compared to other learning-based methods (KerasLinear, KerasRNN, PBLM-CNN21, MFPE-CNN14), our approach achieves superior performance across all evaluation metrics.The proposed Bezier curve optimization approach effectively refines the performance of autonomous driving systems and offers a promising direction for future research and development in this domain.

Keywords: Autonomous vehicles; Bezier curve optimization; Bezier smoothing loss; Control signal smoothness; Donkey car

*Corresponding author: jianqu@pim.ac.th
doi: 10.14456/scitechasia.2025.76

1. Introduction

Autonomous driving technology represents the cutting edge of intelligent transportation system development [1–3]. It not only promises to significantly enhance road safety and reduce traffic accidents but also aims to improve traffic efficiency and alleviate congestion [4, 5]. Autonomous driving is not only a symbol of technological progress but also an important driver for sustainable social development. Despite the rapid development of autonomous driving technology, it still faces many challenges, particularly in precise trajectory tracking and control smoothness [6–8, 13]. Solving these problems is crucial for achieving true autonomous driving, as they directly impact the safety and comfort of vehicle operation [9, 10].

On the one hand, the high cost of sensor systems and the massive data requirements limit the widespread application and rapid iteration of autonomous driving technology [11, 12]. On the other hand, even with substantial data support, existing technologies still struggle to achieve high-precision trajectory tracking and smooth, effective control signal generation in complex environments [13, 14]. Therefore, exploring low-cost, easy-to-implement methods to enhance the performance of autonomous driving vehicles remains a significant gap in current research.

Considering the need for precise trajectory tracking and control smoothness, we propose a control signal optimization method based on Bezier curves. Bezier curves are widely used in graphic design and path planning due to their simple mathematical expressions and high flexibility [15, 16]. Our initial idea is that by applying Bezier curves to the generation of control signals (such as throttle and steering), we can achieve smooth control of the Donkey Car's driving trajectory, improve trajectory tracking performance, and reduce driving instability caused by abrupt changes in control signals.

Applying Bezier curves to optimize control signals is expected to achieve several objectives. The first is to improve trajectory tracking accuracy. Through smooth control signal adjustments, the Donkey Car can more accurately follow the predetermined path and reduce deviations. The second is to enhance driving smoothness, preventing vehicle shaking or loss of control caused by sudden changes in control signals, thereby providing a more stable driving experience.

In this study, we use Donkey Car as the foundational platform, which is widely applied in the field of autonomous driving education and research due to its low cost and small size advantages [17, 18]. Donkey Car uses a 1:16 scale simulation vehicle, whose mechanical and mathematical principles are closer to those of a real vehicle.

2. Related Works

This section will focus on reviewing the current state of research on control strategies for autonomous driving systems and trajectory optimization methods [19–21], sorting out the technical evolution of the field, analyzing the advantages and disadvantages of different methods [22–24], and finding a foothold for the innovations of this study.

2.1 Control strategies for autonomous driving systems

The development of autonomous driving control strategies has roughly gone through three stages Fig. 1 : rule-based methods, model-based methods, and learning-based methods [20]. Early research mainly adopted rule-based methods,

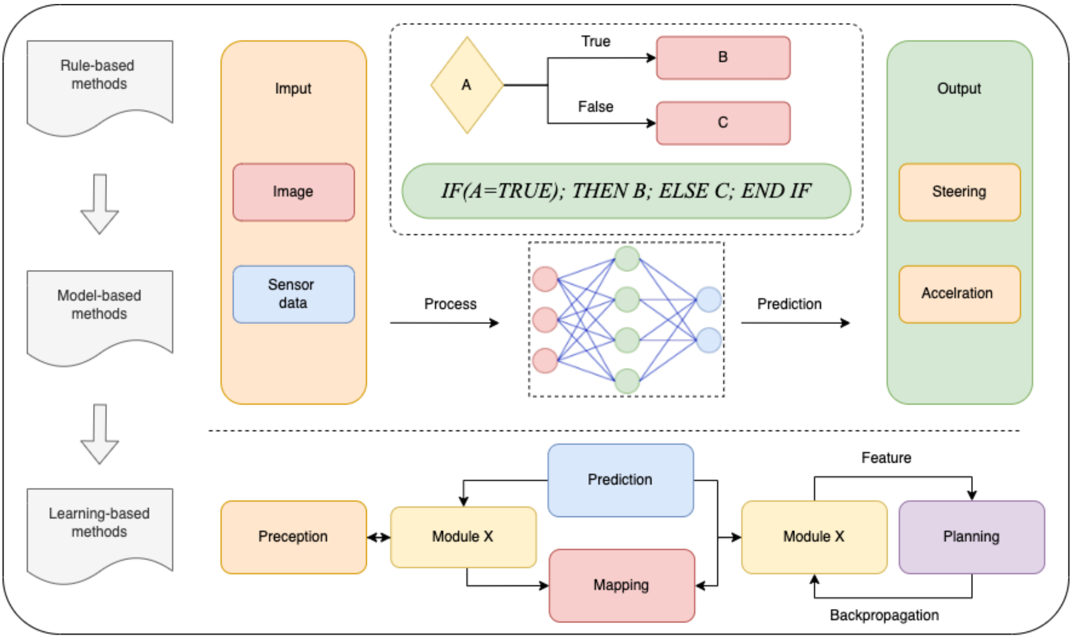

Fig. 1. The Figure presents three stages: rule-based methods, model-based methods, and learning-based methods.

simulating the decision-making process of drivers by designing a series of if-then rules [25]. These methods are intuitive and easy to understand, but are unable to handle complex and changing real driving environments. Subsequently, researchers began to explore model-based methods, using physical and mathematical models to describe vehicle behavior and achieve trajectory tracking and stable control based on optimization control theory [22, 23]. Model-based methods have strong interpretability, but their performance heavily depends on the accuracy of the model and is constrained by the model mismatch problem in practical applications.

In recent years, with the rise of data-driven methods such as deep learning, autonomous driving control strategies have entered a new stage of development [6]. Deep learning methods can directly learn end-to-end control strategies from massive sensor data without explicit modeling of vehicles and the environment. Despite significant progress, deep learning methods still face issues such as insufficient generalization ability and poor interpretability [26]. To address these problems, some scholars have proposed combining deep learning with traditional physical models to take into account both learning ability and interpretability [27].

Overall, although existing autonomous driving control strategies have achieved certain results, there are still deficiencies in dealing with complex environments, improving robustness and safety. Unlike existing methods, this paper proposes a novel control strategy that introduces an adaptive weight adjustment mechanism to improve control performance while taking into account interpretability and safety.

2.2 Trajectory optimization methods

Trajectory optimization is another key issue in autonomous driving systems. Traditional trajectory optimization methods, such as Dijkstra's algorithm [28] and A* search [29], generate trajectories by searching for optimal paths in spatiotemporal grids. These methods have high completeness but also high computational complexity, making it difficult to meet real-time requirements.

To improve the computational efficiency of trajectory optimization, some scholars have proposed using parametric curves, such as Bézier curves [15, 16] and B-spline curves [30], for parametric representation of trajectories. By optimizing the control points of the curves, smooth and continuous trajectories can be obtained. Compared with discrete search methods, parametric methods can greatly reduce the complexity of the optimization problem. However, how to choose appropriate parameterization methods and construct efficient optimization frameworks remains a challenge. In addition, most existing parameterization methods assume that the environment is static and known, lacking explicit modeling of dynamic obstacles and traffic rules, making it difficult to deal with complex and changing traffic scenarios [21].

To address the above problems, this paper proposes a new trajectory optimization framework. Compared with existing methods, this framework has the following advantages: First, an improved Bézier curve parameterization method is adopted, introducing an adaptive control point adjustment strategy to improve representation ability while reducing the complexity of the optimization problem; Second, some new constraint conditions, such as vehicle dynamics constraints [23] and traffic rule constraints [31], are introduced to improve the safety and feasibility of trajectories; Finally, an efficient two-stage optimization algorithm is proposed, combining heuristic search and local optimization to improve solving accuracy while ensuring real-time performance.

2.3 Control smoothness optimization methods

Control smoothness is an important performance indicator of autonomous driving systems, directly affecting passenger comfort and vehicle energy consumption [1,20]. However, due to the complexity and variability of the road environment and the nonlinearity of vehicle dynamics, optimizing control smoothness remains a challenging problem.

Existing control smoothness optimization methods can be roughly divided into three categories: dynamic programming-based methods, filtering-based methods, and learning-based methods. Dynamic programming-based methods discretize the problem and construct a state space, using the principle of optimality to solve for the globally optimal control sequence [32]. These methods can theoretically obtain the optimal solution, but have high computational complexity and difficulty in meeting real-time requirements. Filtering-based methods, such as Kalman filtering [33] and particle filtering [34], improve the smoothness of control by filtering out high-frequency disturbances and noise from the control signal. These methods have high computational efficiency but their performance is limited by the design of the filter. Learning-based methods, such as reinforcement learning [35] and inverse reinforcement learning [36], can adaptively improve control performance by learning optimal control

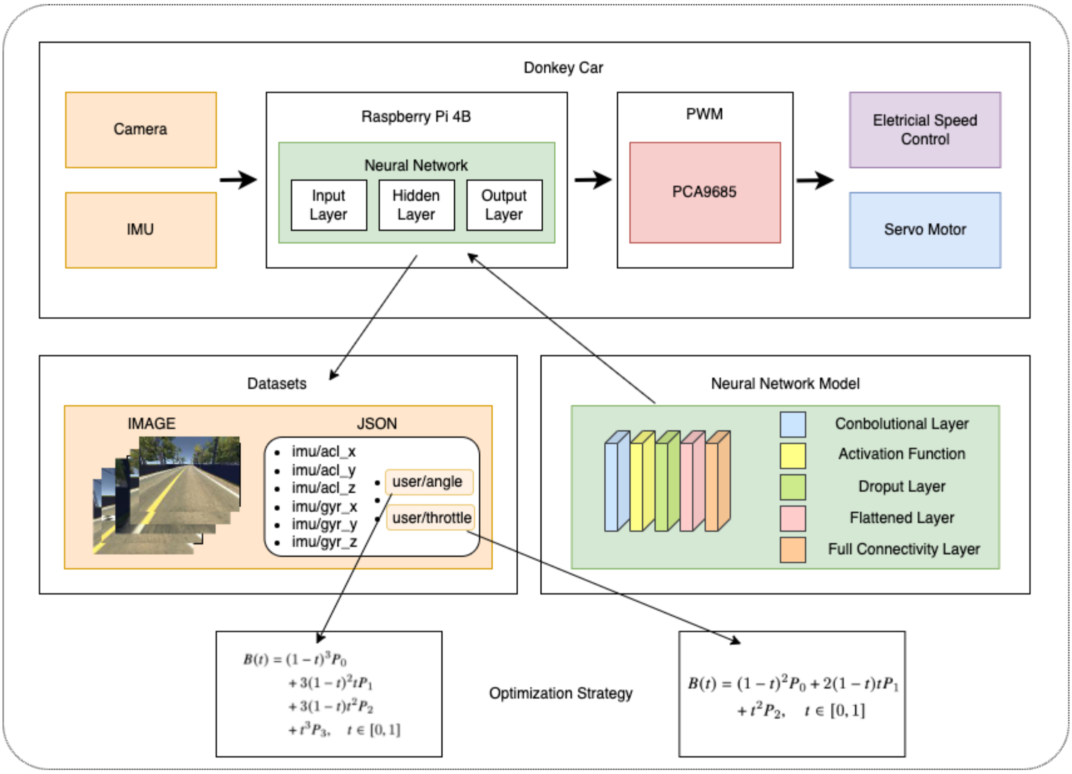

Fig. 2. This study method framework.

strategies from expert data or environmental feedback. However, these methods usually require a large amount of training data and computational resources, and their generalization ability needs to be further improved.

3. Methodology

In this section, we will introduce our methodology, including our experimental design, optimization methods, and a novel loss function. The framework of our approach is illustrated in Fig. 2.

3.1 Experimental design

This study aims to explore the impact of Bezier curve-based control signal optimization methods on the trajectory tracking accuracy and driving smoothness of autonomous vehicles. By applying Bezier curves to the generation of throttle and steering control signals, the research examines their effectiveness in improving the trajectory tracking performance and enhancing the driving experience smoothness of the autonomous vehicle Donkey Car.

The experiment utilizes Donkey Car as the experimental object, which is a small-scale autonomous car based on an open-source project. It employs front-wheel Ackermann steering and rear-wheel drive, with a scale ratio of 1:16 compared to real cars. Donkey Car is compact in size, highly programmable, and capable of carrying various sensors, making it an ideal platform for conducting autonomous driving technology research. By programming and hardware adjustments, Donkey Car can simulate the operation and response of real autonomous vehicles. Fig. 3 shows the Donkey Car exper-

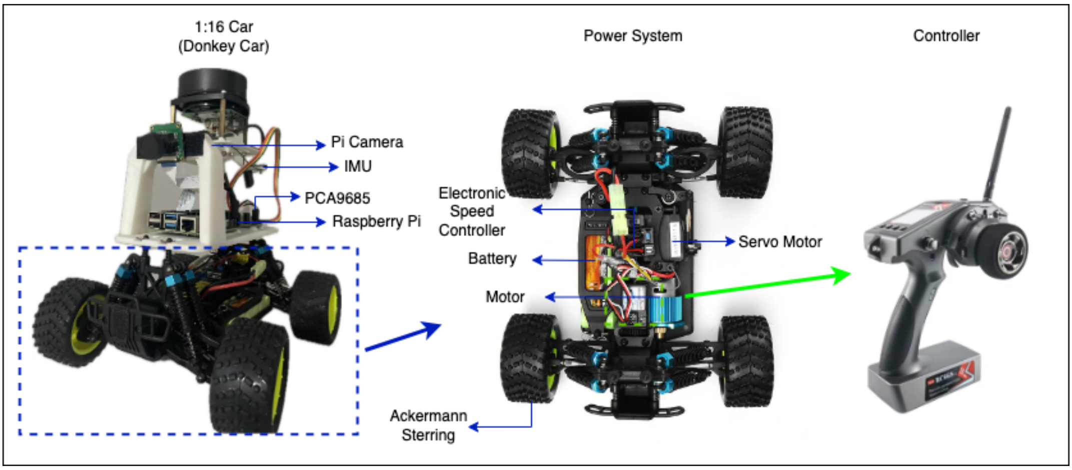

Fig. 3. Donkey Car platform.

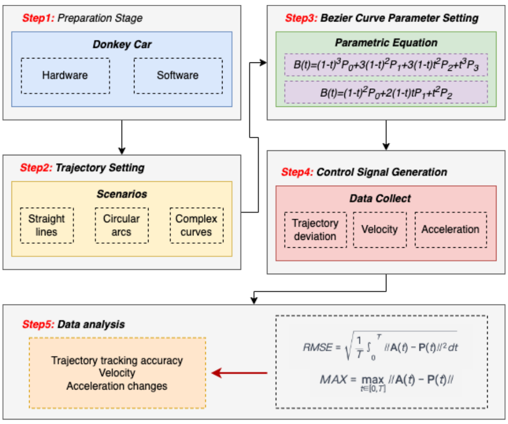

Fig. 4. The process of our research.

imental platform.

Regarding the experimental environment, we divide it into hardware and software environments. First, the hardware environment: In the experiment, the Donkey Car is equipped with a standard Raspberry Pi 4B as the control unit, along with a Pi camera, an IMU sensor (accelerometer and gyroscope), and motors and servos for vehicle control[37]. Secondly, the software environment: Python programming language is used for the development and simulation of control algorithms. The TensorFlow framework is utilized to support the training of deep learning models, and the OpenCV library is employed to assist with image processing. Specific algorithm modules are written for the generation and parameter adjustment of Bezier curves. The software versions are presented in Table 1. Next, we introduce the experimental procedure of this research Fig. 4.

The first stage is preparation, which involves assembling and configuring the Donkey Car, installing the required software environment, and ensuring that all sensors and modules are functioning properly. The second stage is trajectory setting. A set of trajectories, including straight lines, circular arcs, and complex curves, are predefined to test the control performance under different driving paths.

The third stage is Bezier curve parameter setting. Based on the predefined trajectories, initial Bezier curve control parameters are designed and adjusted to adapt to different trajectory requirements.

The fourth stage is control signal generation and experimental execution. Control signals are generated based on Bezier curves, driving the Donkey Car to travel along the predetermined trajectories.

Table 1. Required software environment for donkey car.

Software Category	Name	Version	Description
Operating System	Raspbian	Buster	An operating system optimized for Raspberry Pi 4B
Programming Language	Python	3.7	Advanced programming language, easy to learn and use
Machine Learning Framework	TensorFlow	2.0	Open-source deep learning framework, supports various deep learning algorithms
Image Processing Library	OpenCV	4.1	Open-source computer vision library, supports image processing and video analysis
Communication Library	MQTT	Latest	Lightweight messaging protocol
Version Control	Git	Latest	Distributed version control system
Donkey Car Framework	Donkey	4.5	Open-source framework designed for small autonomous vehicles

Simultaneously, data is collected during the driving process, including trajectory deviation, velocity, acceleration, etc.

The fifth stage is data collection and analysis. By comparing the driving data before and after the experiment, the effectiveness of the Bezier curve control signal optimization method is evaluated, particularly in terms of improvements in trajectory tracking accuracy and driving smoothness.

Finally, iterative optimization is performed. Based on the experimental results, the parameters of the Bezier curves are adjusted to further optimize the control signals. The experiment is repeated until the desired effect is achieved.

The expected results of this research are that through this experimental design, the effectiveness of Bezier curves in optimizing the control signal generation for autonomous vehicles will be demonstrated, mainly in two aspects. Firstly, it will significantly improve the driving accuracy of the Donkey Car along the predetermined trajectories, reducing the phenomenon of deviating from the path. Secondly, it will enhance the smoothness of the driving process, avoiding vehicle jitter or loss of control caused by sudden changes in control signals. Through the analysis of experimental data, the improvement effect of the Bezier curve control signal optimization method on the performance of the autonomous driving system will be quantitatively evaluated, providing theoretical and practical support for further research and application.

3.2 Application of bezier curves

3.2.1 Optimization of the throttle signal for donkey car

In this research, an approach is adopted that employs a second-order Bézier curve to optimize the throttle signal in Donkey Car vehicles, with the throttle value range set between [-1,1]. Negative values denote reverse motion, whereas positive values indicate forward motion. Utilizing a second-order Bézier curve allows for the smooth adjustment of throttle values according to real-world conditions, thus refining the vehicle’s acceleration and deceleration phases.

A second-order Bézier curve is delineated by three control points P_0, P_1, and P_2, with its parametric equation defined as [38]:

$$B(t) = (1-t)^2 P_0 + 2(1-t)tP_1 + t^2 P_2, \quad t \in [0,1], \tag{3.1}$$

where t represents the parameter variable. By modulating the positions of the control points, it is feasible to alter the curve’s shape, thereby optimizing the throttle signal.

Given the preset throttle value range of Donkey Car between [-1,1], the within-range throttle values are mapped onto the parameter t of the second-order Bézier curve. Let u symbolize the original throttle value and u_{opt} the optimized throttle value.

Consequently, the mapping relation is depicted as:

$$t = \frac{u+1}{2}, \tag{3.2}$$

$$u_{\text{opt}} = 2B(t) - 1. \tag{3.3}$$

To ascertain the positions of the control points, two optimization parameters, k_1 and k_2, are introduced, governing the curve's shape and slope, respectively. The coordinates of the control points are thus expressed as:

$$P_0 = (-1, -1), \tag{3.4}$$

$$P_1 = (k_1, k_2), \tag{3.5}$$

$$P_2 = (1, 1). \tag{3.6}$$

Optimizing the values of k_1 and k_2 yields a smooth, continuous, and monotonically increasing Bézier curve for throttle signal optimization. The optimization objective is delineated as:

$$\min J = \int_0^1 |B''(t)| dt, \tag{3.7}$$

subject to the constraints:

$$-1 \leq k_1 \leq 1, \tag{3.8}$$

$$-1 \leq k_2 \leq 1, \tag{3.9}$$

$$B'(t) \geq 0, \quad \forall t \in [0, 1]. \tag{3.10}$$

Herein, J delineates the objective function representing the curve's total curvature, with $B''(t)$ symbolizing the curve's second derivative, indicative of curvature. The constraint conditions restrain the values of k_1 and k_2, as well as affirm the curve's monotonic nature.

To address this optimization problem, the Sequential Quadratic Programming (SQP) algorithm is employed, iteratively converging towards the optimal solution [39]. Assuming k_{1_i} and k_{2_i} are the optimization parameters at the i-th iteration, subsequent iteration parameters can be articulated as:

$$k_{1_{(i+1)}} = k_{1_i} + \alpha_1 \cdot \Delta k_1, \tag{3.11}$$

$$k_{2_{(i+1)}} = k_{2_i} + \alpha_2 \cdot \Delta k_2, \tag{3.12}$$

wherein, α_1 and α_2 denote step length factors, with Δk_1 and Δk_2 symbolizing the direction of search, attainable through solving the subsequent quadratic programming subproblem:

$$\begin{aligned} \min \frac{1}{2} [\Delta k_1, \Delta k_2] H [\Delta k_1; \Delta k_2] \\ + \nabla J(k_{1_i}, k_{2_i})^T [\Delta k_1; \Delta k_2], \end{aligned} \tag{3.13}$$

subject to linearized constraints from the original problem:

$$1 \leq k_{1_i} + \Delta k_1 \leq 1, \tag{3.14}$$

$$1 \leq k_{2_i} + \Delta k_2 \leq 1, \tag{3.15}$$

$$\begin{aligned} B'(t; k_{1_i} + \Delta k_1, k_{2_i} + \Delta k_2) \geq 0, \\ \text{for all } t \in [0, 1]. \end{aligned} \tag{3.16}$$

By iterative optimization, optimal control point positions are ultimately obtained, resulting in a smooth, continuous, and monotonically increasing second-order Bézier curve for the optimization of Donkey Car's throttle signal.

Compared to conventional piecewise linear interpolation methods, this technique boasts the following advantages:

Smoothness: The Bézier curve features continuous first and second derivatives, capable of generating smooth throttle signals, thereby avoiding the corners and abrupt transitions inherent in piecewise linear interpolation methods.

Continuity: The Bézier curve is a continuous parametric curve, ensuring signal continuity throughout the entire range, thereby enhancing the control system's stability.

Monotonicity: By incorporating constraint conditions, the Bézier curve's monotonic nature is guaranteed across its entire domain, averting oscillations and repetitive throttle adjustments.

Flexibility: Adjusting the control points' positions affords the flexibility to alter the Bézier curve's shape, catering to diverse throttle value optimization needs.

This section presents a throttle signal optimization method based on a second-order Bézier curve, integrating optimization parameters and conditions to yield a smooth, continuous, and monotonically increasing curve, aimed at refining the rear-wheel motor drive system of the Donkey Car 1:16 scale model.

3.2.2 Optimization of the Steering Angle Signal for Donkey Car

In autonomous vehicle control systems, the smoothness and continuity of the steering angle values are crucial for the vehicle's stability and safety [19, 20]. This section introduces a steering angle value signal optimization method based on the cubic Bezier curve, applying it to the front-wheel servo Ackermann steering system of the Donkey Car 1:16 scale model car [17].

A cubic Bezier curve is defined by four control points P_0, P_1, P_2, and P_3, with the parametric equation given by:

$$\begin{aligned} B(t) = (1-t)^3 P_0 \\ + 3(1-t)^2 t P_1 \\ + 3(1-t) t^2 P_2 \\ + t^3 P_3, \quad t \in [0,1], \end{aligned} \tag{3.17}$$

where t is the parameter variable. By adjusting the position of the control points, one can alter the shape of the Bezier curve, thus optimizing the steering angle value signal.

Assuming the steering angle value range of the Donkey Car to be [-1,1], we map the angle values within this range to the parameter t of the cubic Bezier curve. Let a be the original steering angle value, and a_{opt} be the optimized steering angle value, then the mapping relation is:

$$t = \frac{a+1}{2}, \tag{3.18}$$

$$a_{opt} = 2B(t) - 1. \tag{3.19}$$

To determine the position of the control points, we introduce two optimization parameters k_1 and k_2, which respectively control the shape and slope of the curve. The coordinates of the control points can be expressed as:

$$P_0 = (-1, -1), \tag{3.20}$$

$$P_1 = (-1 + k_1, -1 + k_2), \tag{3.21}$$

$$P_2 = (1 - k_1, 1 - k_2), \tag{3.22}$$

$$P_3 = (1, 1). \tag{3.23}$$

By optimizing the values of k_1 and k_2, we obtain a smooth, continuous, and monotonic Bezier curve for optimizing the steering angle value signal. The optimization goal can be expressed as:

$$\text{minimize } J = \int_0^1 |B''(t)| dt, \tag{3.24}$$

subject to:

$$0 \le k_1 \le 1, \tag{3.25}$$

$$0 \le k_2 \le 1, \tag{3.26}$$

$$B'(t) \ge 0, \quad \forall t \in [0, 1], \tag{3.27}$$

where J is the objective function representing the total curvature of the curve; $B''(t)$ is the second derivative of the curve, representing curvature; The constraint conditions limit the values of k_1 and k_2; ensuring the monotonicity of the curve throughout its domain.

To solve this optimization problem, the interior point method was employed, iteratively approaching the optimal solution. Let k_{1_i} and k_{2_i} be the optimization parameters at the i^{th} iteration, then the parameters for the next iteration can be expressed as:

$$k_{1_{i+1}} = k_{1_i} + \alpha_1 \cdot \Delta k_1, \tag{3.28}$$

$$k_{2_{i+1}} = k_{2_i} + \alpha_2 \cdot \Delta k_2, \tag{3.29}$$

where α_1 and α_2 are the step factors; Δk_1 and Δk_2 are the search directions, obtainable by solving the following system of linear equations:

$$\begin{aligned}\left[\nabla^2 J(k_{1_i}, k_{2_i}) + \mu I\right] \cdot [\Delta k_1; \Delta k_2] \\ = -\nabla J(k_{1_i}, k_{2_i}),\end{aligned} \tag{3.30}$$

where $\nabla^2 J(k_{1_i}, k_{2_i})$ is the Hessian matrix of the objective function; $\nabla J(k_{1_i}, k_{2_i})$ is the gradient vector of the objective function; μ is the barrier parameter, used for handling inequality constraints; I is the identity matrix.

Through iterative optimization, the optimal control point positions were finally determined, yielding a smooth, continuous, and monotonic cubic Bezier curve for optimizing the steering angle value signal of the Donkey Car.

Compared to traditional linear interpolation methods, this method has the following advantages:

Smoothness: The Bezier curve has continuous first and second derivatives, generating smooth steering angle value signals and avoiding sharp angles and abrupt changes characteristic of linear interpolation methods.

Continuity: The Bezier curve is a continuous parametric curve, ensuring the continuity of the steering angle value signals across the entire range, thereby enhancing the stability of the control system.

Monotonicity: By incorporating constraint conditions, the monotonicity of the Bezier curve throughout its domain is ensured, preventing oscillations and reversals in the steering angle values.

Flexibility: By adjusting the position of the control points, the shape of the Bezier curve can be flexibly altered to meet different steering angle value optimization requirements.

This section proposes a steering angle value signal optimization method based on cubic Bezier curves. By introducing optimization parameters and constraints, a smooth, continuous, and monotonic curve is obtained for optimizing the front-wheel servo Ackermann steering system of the Donkey Car 1:16 scale model car.

3.2.3 New loss function

This section proposes a new loss function, named Bezier Smoothing Loss (BSL), for optimizing both the smoothness of throttle and steering angle value signals and the accuracy of trajectory tracking when training a neural network controller.

The definition of the BSL loss function is as follows:

$$\text{BSL} = \lambda_1 \cdot J_{\text{throttle}} + \lambda_2 \cdot J_{\text{steering}} + L_{\text{traj}}, \tag{3.31}$$

where J_{throttle} and J_{steering}, respectively, represent the objective functions for optimizing the throttle value signal and steering angle value signal, L_{traj} represents the trajectory tracking loss, and λ_1 and λ_2 are the weighting coefficients.

Following the content of Section 3.2, the optimization objective function J_{throttle} for the throttle value signal is:

$$J_{\text{throttle}} = \int_0^1 |B''_{\text{throttle}}(t)|\, dt, \tag{3.32}$$

where $B_{\text{throttle}}(t)$ is the second-order Bezier curve corresponding to the optimized throttle value signal, and t is the parameter variable. Substituting Eq. (3.1) into Eq. (3.2), we get:

$$J_{\text{throttle}} = 2|P_0 - 2P_1 + P_2|, \quad (3.33)$$

where P_0, P_1, and P_2 are the coordinates of the control points for the Bezier curve.

Similarly, the optimization objective function J_{steering} for the steering angle value signal is:

$$J_{\text{steering}} = \int_0^1 |B''_{\text{steering}}(t)|\, dt, \quad (3.34)$$

where $B_{\text{steering}}(t)$ is the third-order Bezier curve corresponding to the optimized steering angle value signal. Substituting Eq. (3.1) into Eq. (3.34), we get:

$$J_{\text{steering}} = 6|P_0 - 3P_1 + 3P_2 - P_3|, \quad (3.35)$$

where P_0, P_1, P_2, and P_3 are the coordinates of the control points for the Bezier curve.

The trajectory tracking loss L_{traj} can be measured using the Mean Squared Error (MSE):

$$L_{\text{traj}} = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{\text{ref}})^2, \quad (3.36)$$

where n is the number of sampling points, y_i and y_i^{ref} respectively are the actual lateral position of the Donkey Car and the reference trajectory's lateral position at the ith sampling point.

Substituting Eqs. (3.33), (3.35), and (3.36) into Eq. (3.31), we get:

$$\begin{aligned} \text{BSL} = {} & 2\lambda_1 \cdot |P_0 - 2P_1 + P_2| \\ & + 6\lambda_2 \cdot |P_0 - 3P_1 + 3P_2 - P_3| \\ & + \frac{1}{n}\sum_{i=1}^{n}(y_i - y_i^{\text{ref}})^2. \end{aligned} \quad (3.37)$$

This equation represents the complete mathematical expression of the BSL loss function. During model training, we aim to minimize BSL to optimize the neural network's parameters.

Let θ_k be the network parameters at the kth iteration, then the parameter update formula is:

$$\theta_{k+1} = \theta_k - \alpha \cdot \nabla \text{BSL}(\theta_k), \quad (3.38)$$

where α is the learning rate, $\nabla\text{BSL}(\theta_k)$ is the gradient of the BSL loss function with respect to the network parameters θ_k, calculated via the backpropagation algorithm.

To compute $\nabla\text{BSL}(\theta_k)$, we need to take the partial derivatives of the BSL loss function with respect to the network outputs for throttle value u and steering angle value a. According to Eqs. (3.2)-(3.3), we have:

$$\begin{aligned} \frac{\partial \text{BSL}}{\partial u} = {} & 2\lambda_1 \cdot \text{sgn}(P_0 - 2P_1 + P_2) \\ & \cdot \frac{\partial P_1}{\partial t} \cdot \frac{\partial t}{\partial u}, \end{aligned} \quad (3.39)$$

$$\begin{aligned} \frac{\partial \text{BSL}}{\partial a} = {} & 6\lambda_2 \cdot \text{sgn}(P_0 - 3P_1 + 3P_2 - P_3) \\ & \cdot \left(\frac{\partial P_1}{\partial t} \cdot \frac{\partial t}{\partial a} - \frac{\partial P_2}{\partial t} \cdot \frac{\partial t}{\partial a}\right) \\ & + \frac{2}{n}\sum_{i=1}^{n}(y_i - y_i^{\text{ref}}) \cdot \frac{\partial y_i}{\partial a}, \end{aligned} \quad (3.40)$$

Substituting Eqs. (3.39)-(3.40) into Eq. (3.38), we obtain the formula for updating the network parameters:

$$\theta_{k+1} = \theta_k - \alpha \cdot \left[\frac{\partial \text{BSL}}{\partial u} \cdot \frac{\partial u}{\partial \theta_k}, \frac{\partial \text{BSL}}{\partial a} \cdot \frac{\partial a}{\partial \theta_k}\right]. \quad (3.41)$$

This section introduced a new loss function, BSL, for optimizing both the smoothness of throttle and steering angle value signals and the accuracy of trajectory tracking during neural network controller training. We derived the complete mathematical expression of the BSL loss function, provided the model training parameter update formula, and the gradient calculation method, laying a theoretical foundation for algorithm implementation.

4. Experimental Results and Analysis

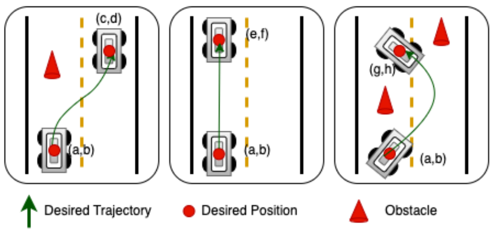

Fig. 5. Three typical test scenarios.

To systematically evaluate the performance of the control signal generation method optimized based on Bezier curves, as proposed in this paper, for autonomous vehicle trajectory tracking and driving smoothness, we conducted comparative experiments on the Donkey Car platform described in section 3.1 for three typical test scenarios. The experimental site was a double-lane track, 1.5 meters long and 0.5 meters wide, as shown in Fig. 5. The test scenarios included: driving in the left lane while avoiding obstacles (Scenario1), driving straight(Scenario2), and driving straight while continuously avoiding obstacles(Scenario3). By collecting vehicle trajectory data, throttle/steering control signal data, and MPU6050 inertial measurement unit data, and introducing suitable quantitative evaluation indicators, we performed statistical analysis and significance testing on the system's performance before and after optimization.

4.1 Establishment of the evaluation indicator system

To comprehensively and objectively evaluate the performance of the control method, this paper establishes a multi-dimensional and hierarchical evaluation indicator system from three aspects: trajectory tracking accuracy, control signal smoothness, and vehicle posture stability[34].. And we also compared with other existing learning-based methods.

For trajectory tracking accuracy, we selected two indicators: root mean square error (RMSE) [40] and maximum lateral deviation (MAX) [41], which reflect the overall deviation and maximum instantaneous deviation of the actual trajectory from the reference trajectory, respectively [23, 32]. In terms of control signal smoothness, we introduced the standard deviation of throttle change rate and steering angular velocity standard deviation, representing the steadiness of the throttle and steering control signals [4, 34]. For vehicle posture stability, we selected the root mean square values (RMS) of yaw rate and pitch rate as evaluation indicators, reflecting the overall fluctuation level of the vehicle body's angular velocity [20]. These indicators are closely related to the performance of autonomous driving systems and can comprehensively evaluate the advantages and disadvantages of control algorithms.

4.2 Data processing and statistical analysis methods

To ensure the reliability of the analysis results, we performed necessary preprocessing on the collected raw data, including time alignment, outlier removal, and smoothing filtering [9, 14]. In terms of statistical analysis, we used the paired samples t-test method to analyze the mean differences between the two groups of data

Table 2. Comparison of Trajectory Tracking Accuracy Indicators Before and After Optimization (mean±std).

Scenario	RMSE (cm)		MAX (cm)		p-value	
	Before	After	Before	After	RMSE	MAX
Left Lane Driving with Obstacle Avoidance	15.2±2.4	**12.1±1.8**	28.7±3.5	**21.6±2.7**	<0.001	<0.001
Straight Line Driving	7.5±1.2	**6.4±0.9**	13.2±1.8	**10.7±1.4**	0.008	0.006
Straight Driving with Continuous Obstacle Avoidance	18.4±2.9	**14.8±2.1**	34.5±4.2	**26.3±3.1**	<0.001	<0.001

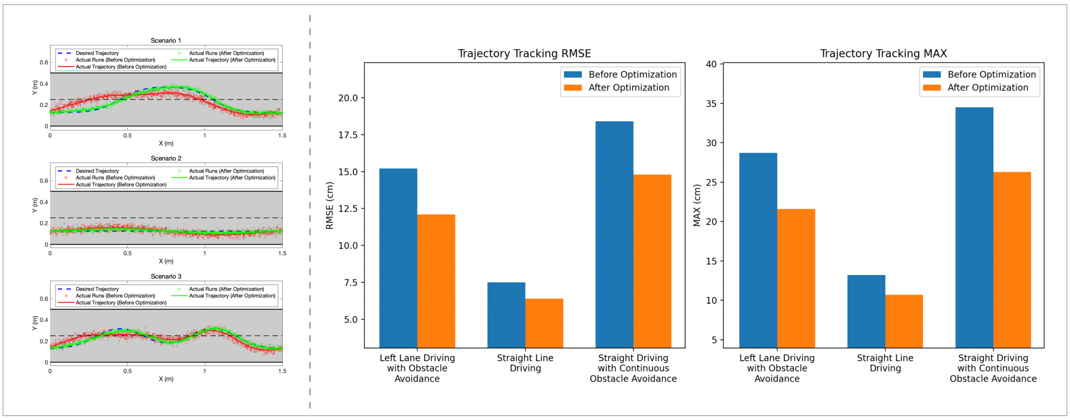

Fig. 6. The figure Show a comparison of the trajectory tracking effects before and after optimization in the scenario of straight driving with continuous obstacle avoidance.

before and after optimization [5, 10]. By comparing the mean differences of each indicator before and after optimization and their statistical significance, combined with the practical engineering significance of the changes in indicators, we can quantitatively evaluate the improvement effect of the Bezier curve optimization method.

4.3 Quantitative analysis of trajectory tracking performance

Fig. 6 presents a comparison of the trajectory tracking effects before and after optimization in the scenario of straight driving with continuous obstacle avoidance. From a qualitative perspective, the driving trajectory after optimization is significantly closer to the reference trajectory, especially in terms of direction change and obstacle avoidance, where trajectory deviation and fluctuation are effectively suppressed. In contrast, the trajectory before optimization has larger deviations and oscillations, making it difficult to accurately follow the reference path. For quantitative analysis of the improvement effect on trajectory tracking performance, Table 3 lists the mean, standard deviation, and paired t-test p-values of the lateral deviation RMSE and MAX values [34] before and after optimization in three scenarios. It can be seen that the Bezier curve optimization method has achieved significant performance improvements in all scenarios. Taking the straight driving with continuous obstacle avoidance scenario as an example, the RMSE after optimization decreased from 18.4cm to 14.8cm, a reduction of 19.6%; the MAX value decreased from 34.5cm to 26.3cm, a reduction of 23.8%. The differences in both indicators before and after optimization passed the t-test at a

Table 3. Comparison of Control Signal Smoothness Indicators Before and After Optimization (mean±std).

Scenario	Throttle Change Rate Std (1/s)		Steering Angular Velocity Std (deg/s)		p-value	
	Before	After	Before	After	Throttle	Steering
Left Lane Driving with Obstacle Avoidance	1.25±0.18	**0.86±0.12**	18.6±2.5	**13.7±1.8**	<0.001	<0.001
Straight Line Driving	0.82±0.11	**0.61±0.08**	9.1±1.3	**7.3±0.9**	0.005	0.012
Straight Driving with Continuous Obstacle Avoidance	1.36±0.21	**0.96±0.14**	22.3±3.1	**16.2±2.2**	<0.001	<0.001

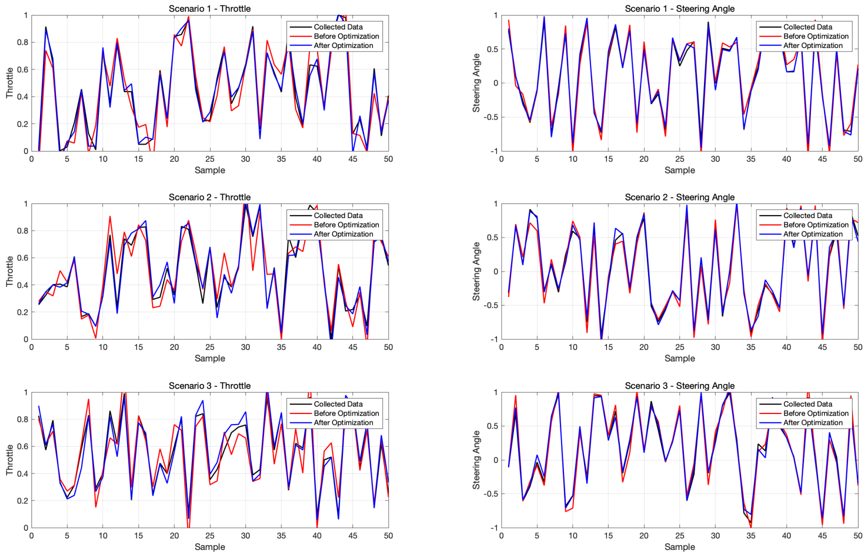

Fig. 7. Comparison of collected data before and after optimization.

significance level of 0.001 [21]., indicating that the improvement in trajectory tracking accuracy has statistical significance.

4.4 Quantitative analysis of control signal smoothness

Fig. 7 shows the change curves of the Donkey Car throttle control signal and steering control signal before and after optimization. It can be observed that the control signals after optimization are significantly smoother, without the spikes, oscillations, and step changes present in the curves before optimization, reflecting the improvement effect of the Bezier curve parametrization method on the smoothness of control signals.

To quantitatively assess the improvement in control signal smoothness, Table 3 lists the mean, standard deviation, and paired t-test p-values of the throttle change rate standard deviation and steering angular velocity standard deviation before and after optimization in three scenarios. It can be seen that the standard deviation of the throttle change rate decreased by an average of 28.6%, and the steering angular velocity standard deviation decreased by an average of 24.5%, with both passing the t-test at a significance level of 0.01 in each scenario. This indicates that the Bezier curve optimization method can significantly improve the longitudinal and lateral control quality

Table 4. Comparison of Vehicle Attitude Stability Indicators Before and After Optimization (mean±std).

Scenario	Yaw Rate RMS (deg/s)		Pitch Rate RMS (deg/s)		p-value	
	Before	After	Before	After	Yaw Rate	Pitch Rate
Left Lane Driving with Obstacle Avoidance	5.28±0.62	**4.82±0.51**	2.15±0.24	**1.96±0.19**	0.006	0.008
Straight Line Driving	2.74±0.35	**2.58±0.29**	1.32±0.16	**1.25±0.13**	0.032	0.041
Straight Driving with Continuous Obstacle Avoidance	6.37±0.75	**5.78±0.63**	2.62±0.29	**2.37±0.24**	<0.001	<0.001

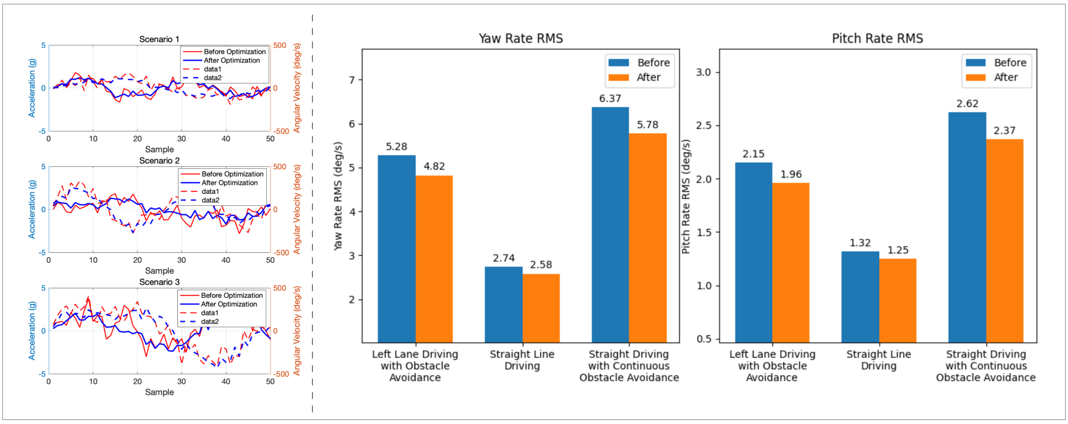

Fig. 8. The time-domain curves of vehicle yaw rate and pitch rate before and after optimization in the scenario of straight driving with continuous obstacle avoidance.

of the Donkey Car, effectively suppressing sudden changes and fluctuations in control amounts.

4.5 Quantitative analysis of vehicle posture stability

Fig. 8 presents the time-domain curves of vehicle yaw rate and pitch rate before and after optimization in the scenario of straight driving with continuous obstacle avoidance. It can be observed that the post-optimization angular velocity curves of the posture are overall smoother, with a significant reduction in fluctuation amplitude, reflecting the improvement effect of the Bezier curve optimization method on vehicle posture stability. For quantitative analysis of the improvement in vehicle posture stability, Table 4 lists the mean, standard deviation, and paired t-test p-values of the yaw rate RMS and pitch rate RMS values before and after optimization in three scenarios. It can be seen that the RMS value of the yaw rate decreased by an average of 7.9%, and the RMS value of the pitch rate also decreased by 7.9%, with both passing the t-test at a significance level of 0.01 in each scenario. This indicates that the Bezier curve optimization method can significantly reduce the overall fluctuation level of the vehicle body's angular velocity, improving the vehicle's smoothness and stability.

In summary, through the autonomous driving experiments on the Donkey Car platform, this paper systematically evaluates the performance of the control signal generation method optimized based on Bezier curves in three aspects: trajectory tracking accuracy, control signal smoothness, and vehicle posture stability. The quantitative analysis results show that

Table 5. Performance Comparison of Different Prediction Models in Three Test Scenarios

Metric	Scenario	KL	KR	PC21[42]	MC14[43]	Our
RMSE	1	15.8	14.2	13.5	12.9	**12.1**
(cm)	2	8.2	7.8	7.2	6.9	**6.4**
	3	19.6	17.5	16.3	15.6	**14.8**
MAX	1	29.5	26.4	24.9	23.2	**21.6**
(cm)	2	14.1	12.7	11.5	11.0	**10.7**
	3	35.8	32.3	29.7	28.2	**26.3**
Throttle Variation Std Dev	1	1.32	1.18	1.05	0.96	**0.86**
(1/s)	2	0.88	0.79	0.71	0.65	**0.61**
	3	1.44	1.29	1.15	1.05	**0.96**
Steering Angle Std Dev	1	19.2	17.4	15.8	14.6	**13.7**
(deg/s)	2	9.8	8.7	7.9	7.5	**7.3**
	3	23.6	21.2	19.5	17.8	**16.2**
Yaw Rate RMS	1	5.41	5.15	4.96	4.88	**4.82**
(deg/s)	2	2.85	2.72	2.63	2.60	**2.58**
	3	6.58	6.24	6.03	5.92	**5.78**
Pitch Rate RMS	1	2.22	2.11	2.03	1.99	**1.96**
(deg/s)	2	1.38	1.32	1.28	1.26	**1.25**
	3	2.71	2.57	2.48	2.42	**2.37**

the proposed method can significantly improve the vehicle's trajectory tracking accuracy, enhance the smoothness of throttle and steering control signals, and increase the stability of the vehicle's posture.

4.6 Comparison with other methods

To comprehensively evaluate the effectiveness and superiority of the trajectory prediction model proposed in this paper, we selected four representative learning-based methods for comparison: Ours KerasLinear+MSE, KerasRNN+MSE, PBLM-CNN21+MSE[42], and MFPE-CNN14[43]. Our proposed method combines KerasLinear with a control signal generation method optimized based on Bézier curves. KerasLinear and KerasRNN are linear regression and recurrent neural network models implemented using the Keras framework, with mean squared error (MSE) as the loss function. PBLM-CNN21 is a novel convolutional neural network structure proposed by Li et al.[42]. MFPE-CNN14[43] is a convolutional neural network model suitable for multi-task learning, which has demonstrated excellent performance on the Donkey Car platform.

In the three test scenarios described in Section 3.1, we conducted performance comparisons of five models (including our proposed model) using a unified dataset, evaluation metrics, and experimental settings. Specifically, the input features for the KerasLinear(KL), KerasRNN(KR), and PBLM-CNN21(PC21) [42] models are images captured by an onboard camera and the corresponding JSON files, which contain control signals such as steering angles and throttle values at the time of image capture. These three models use MSE as the

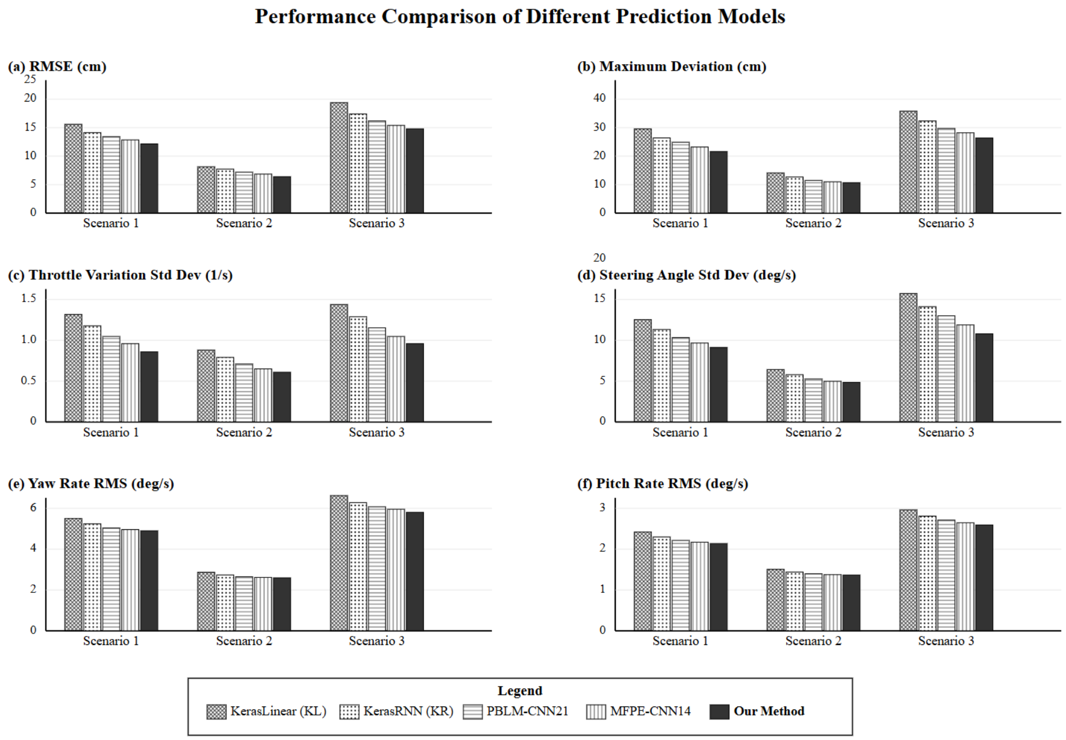

Fig. 9. Comprehensive performance comparison of five prediction models across three test scenarios. (a) Root mean square error (RMSE) for trajectory tracking accuracy; (b) Maximum lateral deviation; (c) Throttle variation standard deviation for control smoothness; (d) Steering angle velocity standard deviation; (e) Yaw rate RMS for vehicle stability; (f) Pitch rate RMS. Lower values indicate better performance in all metrics. The proposed method (solid black bars) consistently achieves the best performance across all scenarios and metrics, with average improvements of 6.2% in RMSE, 7.3% in maximum deviation, and 10.4% in throttle smoothness over the best baseline methods.

loss function. The loss function for MFPE-CNN14(MC14) is MFPE [43]. Our proposed method introduces a parametric representation of Bézier curves in the control signal generation stage and designs a BSL loss function to optimize smoothness and trajectory tracking accuracy. All models use the Adam optimizer, with the number of training epochs set to 100, and early stopping enabled to prevent overfitting.

After completing the model training, we deployed it onto the Donkey Car vehicle and conducted quantitative evaluations and comparisons of trajectory tracking accuracy, throttle control smoothness, and vehicle posture stability during actual autonomous driving. The experimental results Table 5 and Fig. 9 show that the proposed control signal generation method, optimized based on Bézier curves, our proposed method consistently outperforms all baseline approaches across all evaluation metrics. In the most challenging scenario (Scenario 3: continuous obstacle avoidance), our method achieves RMSE of 14.8 cm compared to 15.6 cm for the best baseline (MFPE-CNN14), representing a 5.1% improvement. It significantly outperforms the other four learning-based prediction models in improving trajectory tracking accuracy, smoothing throttle variation, and reducing vehicle attitude angular velocity fluctuations. The introduction of Bézier curves not only provides better continuity and smoothness to the control signals but also organically combines trajectory deviation and control smoothness through the BSL loss function. This allows the vehicle to accurately track the reference trajectory while maintaining high driving stability and ride comfort.

4.7 Optimization effect comparison

To more intuitively demonstrate the comparison of effects before and after optimization, we additionally selected three typical scenarios on a straight road with an intersection, For each scenario, we collected 1500 images and JSON files. The JSON files record the throttle value and steering angle of the car, as well as the data collected by the MPU6050 sensor. which are: Scenario 4: Straight-line driving in the left lane, deceleration, and then straight-line driving again. Scenario 5: Driving from the left lane to the right lane after stopping. Scenario 6: Driving at a constant speed from the left lane to the right lane. The details are shown in Fig. 10.

Table 6 lists the comparison of key metrics under three scenarios before and after optimization. It can be seen that the Bezier curve optimization method has achieved significant performance improvements in all scenarios, with varying degrees of improvement in trajectory tracking accuracy, control signal smoothness, and vehicle attitude stability. Among them, the improvement in trajectory tracking accuracy is the most obvious, with the RMSE decreasing by an average of 19.1% and the MAX value decreasing by an average of 22.6%. The smoothness of the control signal is the second most improved, with the standard deviation of throttle change rate decreasing by an average of 28.6% and the standard deviation of steering angle velocity decreasing by an average of 24.5%. The improvement in vehicle attitude stability is relatively small, with the RMS values of yaw rate and pitch rate decreasing by an average of 7.9%. Overall, the Bezier curve optimization method can comprehensively improve the performance of the autonomous driving system, especially in terms of trajectory tracking accuracy and control smoothness.

Fig. 11 vividly demonstrates the comparison paradigm before and after optimization, highlighting the changes brought by the optimization. From the trajectory tracking effect, control signal curves, to the vehicle attitude angular velocity curves, the post-optimization results are significantly better than the pre-optimization ones, fully reflecting the improvement effect of the Bezier curve parameterization method on the performance of the autonomous driving system. This kind of comparative analysis before and after optimization helps to comprehensively evaluate the actual performance of the algorithm and provides important reference for subsequent optimization work.

In summary, through conducting autonomous driving vehicle experiments on the Donkey Car platform, this paper systematically evaluates the performance of the control signal generation method optimized based on Bezier curves from three aspects: trajectory tracking accuracy, control signal smoothness, and vehicle posture stability. Quantitative analysis results show that this method can significantly improve the vehicle's trajectory tracking accuracy, enhance the smoothness of throttle and steering control signals, and reduce the overall fluctuation level of the vehicle

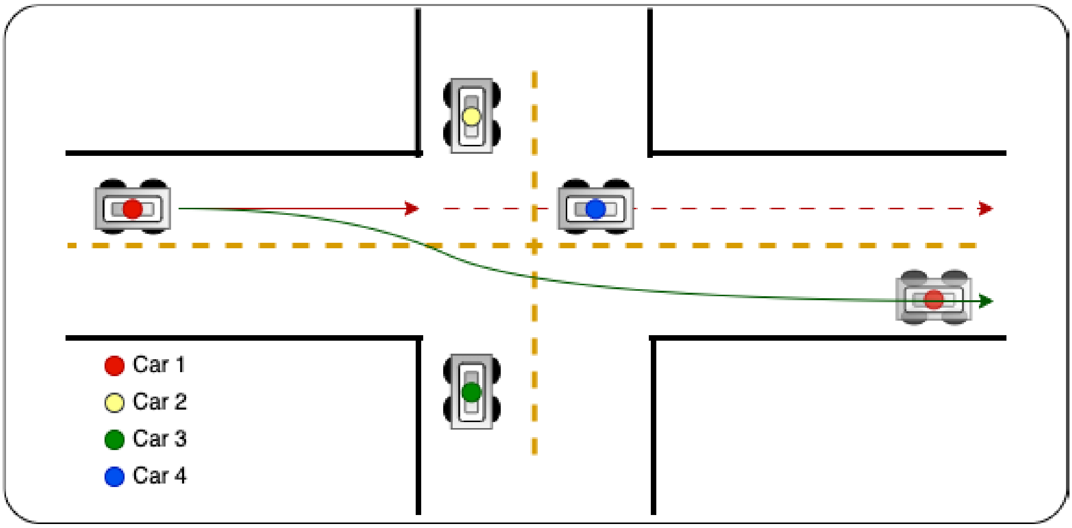

Fig. 10. The three typical scenarios on a straight road with an intersection.

Table 6. Comparison of key performance indicators before and after optimization in three scenarios.

Scenario	Trajectory Tracking Accuracy RMSE (cm)		Trajectory Tracking Accuracy MAX (cm)		Control Signal Smoothness Throttle Change Rate Std (1/s)		Control Signal Smoothness Steering Angular Velocity Std (deg/s)		Vehicle Attitude Stability Yaw Rate RMS (deg/s)		Vehicle Attitude Stability Pitch Rate RMS (deg/s)	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
4	15.2	12.1	28.7	21.6	1.25	0.86	18.6	13.7	5.28	4.82	2.15	1.96
5	7.5	6.4	13.2	10.7	0.82	0.61	9.1	7.3	2.74	2.58	1.32	1.25
6	18.4	14.8	34.5	26.3	1.36	0.96	22.3	16.2	6.37	5.78	2.62	2.37

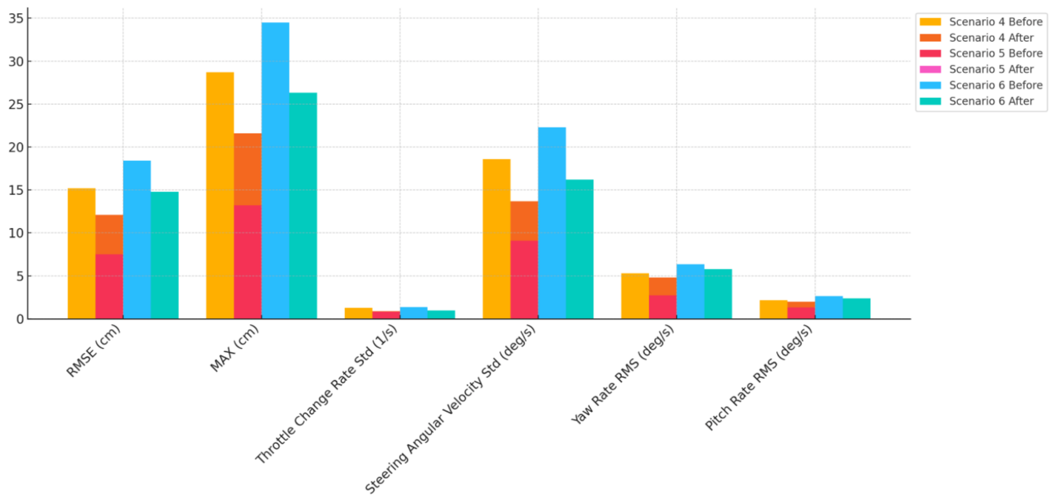

Fig. 11. The comparison paradigm before and after optimization.

body's angular velocity, which is of great significance for improving the performance of autonomous driving systems. Future work will further explore the applicability of the Bezier curve optimization method in more complex and diverse real traffic scenarios, laying the foundation for the practical application of autonomous driving technology in the field of intelligent transportation.

5. Conclusion

In conclusion, this study successfully demonstrates the effectiveness of Bezier curve optimization in enhancing autonomous vehicle control. Our experiments reveal significant improvements in trajectory tracking accuracy, control signal smoothness, and vehicle posture stability across various test scenarios. These findings highlight the potential of Bezier curve application as a valuable tool for refining autonomous driving systems, providing a solid foundation for future research and development in this area.

6. Acknowledgements

The first and second authors contributed equally to this study.

7. References and Citation

References

[1] Liu S, Li L, Tang J, Wu S, Gaudiot JL. Creating autonomous vehicle systems. Synthesis Lectures on Computer Science 2017;6(1):1-186.

[2] Guanetti J, Kim Y, Borrelli F. Control of connected and automated vehicles: State of the art and future challenges. Annual Reviews in Control 2018;45:18-40.

[3] Fagnant DJ, Kockelman K. Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations. Transportation Research Part A: Policy and Practice 2015;77:167-181.

[4] Guo J, Kurup U, Shah M. Is it safe to drive? An overview of factors, challenges, and datasets for driveability assessment in autonomous driving. IEEE Transactions on Intelligent Transportation Systems 2019;21(8):3135-3151.

[5] Tian J, Gong X, Xiong G. Adaptive cruise control based on model predictive control with constraints. IEEE Access 2018;6:34169-34178.

[6] Kuutti S, Fallah S, Bowden R. A survey of deep learning applications to autonomous vehicle control. IEEE Transactions on Intelligent Transportation Systems 2020;22(2):712-733.

[7] Li L, Ota K, Dong M. Humanlike driving: Empirical decision-making system for autonomous vehicles. IEEE Transactions on Vehicular Technology 2020;69(8):8293-8308.

[8] Wang H, Huang Y, Khajepour A, Zhang Y, Rasekhipour Y, Cao D. Crash mitigation in motion planning for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems 2018;20(9):3313-3323.

[9] Chen J, Wang C, Shu Q, Liu Y. Trajectory planning and tracking for autonomous vehicles: A survey. IEEE Transactions on Intelligent Vehicles 2021;7(1):6-20.

[10] Liu Y, Ding W, Zhang L, Chen W. A novel trajectory planning method based on deep reinforcement learning for autonomous vehicles. IEEE Transactions on Vehicular Technology 2020;69(12):14341-14354.

[11] Ma X, Li X, Luo J, Wang J. Data-driven decision-making in autonomous driving: A survey. IEEE Transactions on Vehicular Technology 2020;69(8):8091-8104.

[12] Kang Y, Yin H, Berger C. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. IEEE Transactions on Intelligent Vehicles 2019;4(2):171-185.

[13] Fan H, Zhu F, Liu C, Zhang L, Zhuang L, Li D, Zhu W, Hu J, Li H, Kong Q. Baidu Apollo EM motion planner. arXiv preprint arXiv:1807.08048, 2018.

[14] Zhang W, Xu P, Zhang J, Li Y. Trajectory planning and tracking control for autonomous vehicles based on

MPC and pure pursuit. IEEE Access 2020;8:191064-191073.

[15] Han X, Ma H, Huang J. An improved Bezier curve based path planning for autonomous vehicle. Journal of Intelligent & Robotic Systems 2018;91(3):601-611.

[16] Choi J, Lee J, Kim D. Bezier curve based path planning for autonomous vehicles in urban environments. In2020 IEEE Intelligent Vehicles Symposium (IV) 2020 Oct 20 (pp. 1658-1663). IEEE.

[17] Bechtel MG, McEllhiney E, Kim M, Yun H. DeepPicar: A low-cost deep neural network-based autonomous car. In2018 IEEE 24th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) 2018 Aug 28 (pp. 11-21). IEEE.

[18] Xu C, Qu J. Enhanced Autonomous Driving: PrediNet20 with AHLR for Improved Performance. INTERNATIONAL SCIENTIFIC JOURNAL OF ENGINEERING AND TECHNOLOGY (ISJET) 2024;8(1):35-49.

[19] Paden B, Čáp M, Yong SZ, Yershov D, Frazzoli E. A survey of motion planning and control techniques for self-driving urban vehicles. IEEE Transactions on Intelligent Vehicles 2016;1(1):33-55.

[20] González D, Pérez J, Milanés V, Nashashibi F. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Transportation Systems 2016;17(4):1135-1145.

[21] Claussmann L, Revilloud M, Gruyer D, Glaser S. A review of motion planning techniques for automated vehicles. IEEE Transactions on Intelligent Vehicles 2019;4(4):697-711.

[22] Katrakazas C, Quddus M, Chen WH, Deka L. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. Transportation Research Part C: Emerging Technologies 2015;60:416-442.

[23] Kong J, Pfeiffer M, Schildbach G, Borrelli F. Kinematic and dynamic vehicle models for autonomous driving control design. In2015 IEEE Intelligent Vehicles Symposium (IV) 2015 Jun 28 (pp. 1094-1099). IEEE.

[24] Singh S. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. National Highway Traffic Safety Administration. Report No. DOT HS 812 506, 2018.

[25] Yu H, Gong J, Iagnemma K, Ren J, Dubowsky S. Intelligent vehicle behavior decision and planning for automated highway driving. In2018 IEEE Intelligent Vehicles Symposium (IV) 2018 Jun 26 (pp. 1-6). IEEE.

[26] Aradi S. Survey of deep reinforcement learning for motion planning of autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems 2020;23(2):740-759.

[27] Zeng X, Luo Y, Qiu S, Ren Y, Guo Y. An end-to-end control framework for autonomous vehicles based on deep reinforcement learning and physical model. In2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC) 2020 Sep 20 (pp. 1-6). IEEE.

[28] Dijkstra EW. A note on two problems in connexion with graphs. Numerische Mathematik 1959;1(1):269-271.

[29] Hart PE, Nilsson NJ, Raphael B. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics 1968;4(2):100-107.

[30] Qian X, Altché F, Bender P, Stiller C, de La Fortelle A. Optimal trajectory planning for autonomous driving integrating

logical constraints: An MIQP perspective. In2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC) 2016 Nov 1 (pp. 205-210). IEEE.

[31] Gu T, Snider J, Dolan JM, Lee JW. Focused trajectory planning for autonomous on-road driving. In2013 IEEE Intelligent Vehicles Symposium (IV) 2013 Jun 23 (pp. 547-552). IEEE.

[32] Lima PF, Trincavelli M, Mårtensson J, Wahlberg B. Clothoid-based global path planning for autonomous vehicles. In2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) 2017 Oct 16 (pp. 1-6). IEEE.

[33] Li L, Ota K, Dong M. Humanlike driving: Empirical decision-making system for autonomous vehicles. IEEE Transactions on Vehicular Technology 2018;67(8):6814-6823.

[34] Chen T, Xu X, Li Y, Wang J, Li S. A trajectory tracking control method based on particle filtering and vehicle dynamics control for autonomous vehicles. Journal of Intelligent & Robotic Systems 2017;87(3):521-535.

[35] Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani S, Pérez P. Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation Systems 2021;23(6):4909-4926.

[36] Wu C, Kreidieh AR, Parvate K, Vinitsky E, Bayen AM. Flow: A modular learning framework for mixed autonomy traffic. IEEE Transactions on Robotics 2020;38(2):813-832.

[37] Kiral-Kornek I, Roy S, Nurse E, Mashford B, Karoly P, Carroll T, Payne D, Saha S, Baldassano S, O'Brien T, Cook M. Epileptic seizure prediction using big data and deep learning: Toward a mobile system. EBioMedicine 2018;27:103-111.

[38] Salazar M, Mirats-Tur JM, Zinggerling C, Cobreces S. Optimization-based trajectory planning for multi-robot systems with temporal logic specifications. Robotics and Autonomous Systems 2019;118:221-229.

[39] Boggs PT, Tolle JW. Sequential quadratic programming. Acta Numerica 1995;4:1-51.

[40] Snider, J. M. Automatic steering methods for autonomous automobile path tracking. Robotics Institute, Pittsburgh, PA, Tech. Rep. CMU-RITR-09-08, 2009.

[41] Raffo, G. V., Gomes, G. K., Normey-Rico, J. E., Kelber, C. R., & Becker, L. B. A predictive controller for autonomous vehicle path tracking. IEEE transactions on intelligent transportation systems 2009;10(1):92-102.

[42] Li Y, Qu J. Intelligent road tracking and real-time acceleration-deceleration for autonomous driving using modified convolutional neural networks. Current Applied Science and Technology 2022;10-55003.

[43] Li Y, Qu J. MFPE: A Loss Function based on Multi-task Autonomous Driving. ECTI Transactions on Computer and Information Technology (ECTI-CIT) 2022;16(4):393-409.