Intelligent vehicle lateral tracking algorithm based on neural network predictive control

Yi Su[1,2]*, Lv Xu[1,2] and Jiehui Li[3]

[1]School of Intelligent Equipment and Automotive Engineering, Wuxi Vocational Institute of Commerce, Wuxi, China, [2]Jiangsu Province Engineering Research Center of Key Components for New Energy Vehicle, Wuxi, China, [3]School of Automobile and Traffic Engineering, Jiangsu University, Zhenjiang, China

Introduction: Intelligent vehicles and autonomous driving have been the focus of research in the field of transport, but current autonomous driving models have significant errors in lateral tracking that cannot be ignored.

Methods: In view of this, this study innovatively proposes a lateral trajectory algorithm for intelligent vehicles based on improved radial basis function (RBF). The algorithm first models the lateral trajectory behaviour of the car based on the pre-scanning steering theory, and then proposes an improved RBF network model to compensate for the error of the lateral trajectory model and further improve the accuracy.

Results: According to the simulation test results, after 20 iterations, the proposed algorithm always shows the highest accuracy with the same number of iterations. When the number of iterations reaches 370, the accuracy of the algorithm is stable at 88%. In addition, the bending test shows that the proposed algorithm performs best at low speeds with an overall error of 0.028 m, which is a higher accuracy compared to the algorithm without neural network compensation.

Discussion: The maximum error of the proposed algorithm does not exceed 0.04 m in complex continuous curved terrain, which is safe within the normal road width. Overall, the lateral tracking algorithm proposed in this research has better lateral tracking capability compared to other improved algorithms of the same type. The research results are of some significance to the field of lateral tracking of automatic driving, which provides new ideas and methods for the field of lateral tracking of automatic driving technology and helps to promote the overall development of automatic driving technology. By reducing the lateral tracking error, the driving stability and safety of the self-driving car can be improved, creating favourable conditions for the wide application of the self-driving technology.

KEYWORDS

neural network, intelligent vehicles, horizontal tracking, autonomous driving, radial basis function

1 Introduction

The current automotive manufacturing industry is constantly exploring the possibility of intelligence. The development of automotive intelligence currently has multiple directions, including production line automation, Internet of Things technology, and intelligent supply chain management. Among these development directions,

autonomous driving of intelligent vehicles is an important part (Zhao et al., 2023). The tracking algorithms in autonomous driving are mainly divided into longitudinal tracking and lateral tracking. Longitudinal tracking is related to the speed control and following of the vehicle, while lateral tracking is related to whether the car can correctly perform turning and lane changing operations (Abadi et al., 2023). The safety of autonomous driving in intelligent vehicles is directly related to the accuracy and error of the tracking algorithm. If the longitudinal tracking algorithm is not accurate, the vehicle may exhibit abnormal acceleration and deceleration behavior, leading to traffic congestion and increased accident risk (El-Bayoumi, 2021). Lateral tracking mainly affects the performance of vehicles driving on curves. Errors in lateral tracking algorithms may cause vehicles to deviate from the lane, and even collide with other vehicles or roadside obstacles, thereby affecting driving safety (Fu and Fu, 2023). Although the current lateral tracking algorithm is available, there are still significant errors, which may lead to safety hazards for vehicles when facing complex road surfaces and continuous bends (Halilaj et al., 2023). The most important part of the smart car system is the navigation system, which includes path planning and motion control. Path planning automatically calculates the best driving route according to the driver's destination and considers avoiding congested roads or other unfavourable factors. Motion control ensures that the vehicle is able to follow the planned route accurately, including functions such as positioning, map display and speed control. One of the core goals of route planning is to ensure that the vehicle or robot can reach its destination safely while avoiding collisions with obstacles. This needs to be achieved through algorithms and techniques to ensure that the right decisions can be made in complex environments (Yasin et al., 2020; Sunan et al., 2019). Therefore, a lateral tracking algorithm for preview vehicles on the grounds of neural network error compensation is proposed. In this algorithm, an optimized RBF neural network is used to correct the error of the lateral tracking algorithm, to improve the safety of autonomous driving. The study consists of four parts in total. The first part discusses the important research achievements in related fields in recent years. The second part constructs the proposed algorithm. The third part tests the proposed algorithm. The fourth part summarizes the entire study.

2 Related works

As an effective self-learning technique, neural networks have been studied both theoretically and practically. Bao et al. investigated the three-dimensional (3-D) dynamic trajectory tracking control of autonomous underwater vehicle (AUV), the traditional control method based on the nominal model of AUV could not guarantee the accuracy of the control system, Bao et al. (2024) proposed a prediction model based on radial basis function neural network (RBF-NN) based prediction model. The results show that the proposed controller is more efficient and robust than the standard model predictive controller (MPC) and standard model predictive controller. Bhosle and Musande (2023) proposed a model using deep learning convolutional neural networks for number recognition, addressing the difficulty of recognizing characters and numbers with different writing styles and techniques. This model to some extent imitates the thinking process of the human

brain. Compared with feedforward neural networks and random forest methods, this model exhibits higher accuracy, reaching 99.2%. It is particularly effective in organizing and unstructured data such as images, videos, and audio. In this study, Matveev (2020) modeled a small autonomous hydrofoil vessel used for interception operations. Adopting a 3-degree-of-freedom model including heave, sway, and yaw to simulate the maneuvering motion of a ship under wing loading conditions. In the simulation of horizontal ship dynamics, the forces generated by propellers, rudders, and struts were considered. The results indicate that the proposed model and results can assist engineers in designing more effective motion control methods for fast boats used for interception operations.

As a hot topic in the fields of artificial intelligence and transportation, autonomous driving has received attention in recent years. Liu led his team to propose a method of using an unscented Kalman filter to fuse radar and LiDAR data for high-precision detection and tracking of surrounding targets in automotive autonomous driving. The actual vehicle test has demonstrated the effectiveness of this fusion method in accurately detecting and tracking peripheral targets. Compared with a single sensor, this method has significant advantages and enhances the intelligence of autonomous vehicle (Liu et al., 2021). Wang et al. proposed an asynchronous supervised learning method to improve the initial performance of the auto drive system on the grounds of reinforcement learning. This method introduces prior knowledge by parallel executing supervised learning processes on multiple driving datasets during pre training. The evaluation of TORCS simulator and actual vehicle deployment has verified the reliability of this method, demonstrating a significant improvement in initial performance and convergence speed during the training process under this model (Wang et al., 2021). Hoffmann and Klein proposed a depth distribution model for autonomous driving. The model can generate calibrated steering angle density, which performs well in marginal calibration and avoiding excessive learning, increasing the interpretability of autonomous driving in automobiles (Hoffmann and Klein, 2023). Stocco and Tonella (2022) proposed a learning framework that can continuously learn predictive factors for inappropriate behavior and adapt using on-site behavioral data. The evaluation of the autonomous vehicle simulator shows that while maintaining the original ability to predict faults several seconds in advance, the false alarm rate and the ability to adapt to behavior drift are significantly reduced.

On the grounds of the research achievements in the fields of neural networks and autonomous driving in recent years, this indicates that neural networks, as algorithms with self-learning ability, have relatively mature applications in various industries. In recent years, autonomous driving has received increasing attention, but few studies have focused on the lateral tracking branch in autonomous driving. The lateral tracking algorithms used in current autonomous driving schemes often have certain errors, which are particularly evident when dealing with non-linear routes. Therefore, the study applies neural networks to the lateral tracking of intelligent vehicles to fill the research gap in this branch.

Latest Technology Techniques MPC-based algorithms and algorithms combining deep learning and reinforcement learning usually have better performance in terms of accuracy and stability, and are better able to adapt to complex and changing driving environments. The optimised radial basis function neural

FIGURE 1
Non preview model of car lateral tracking.
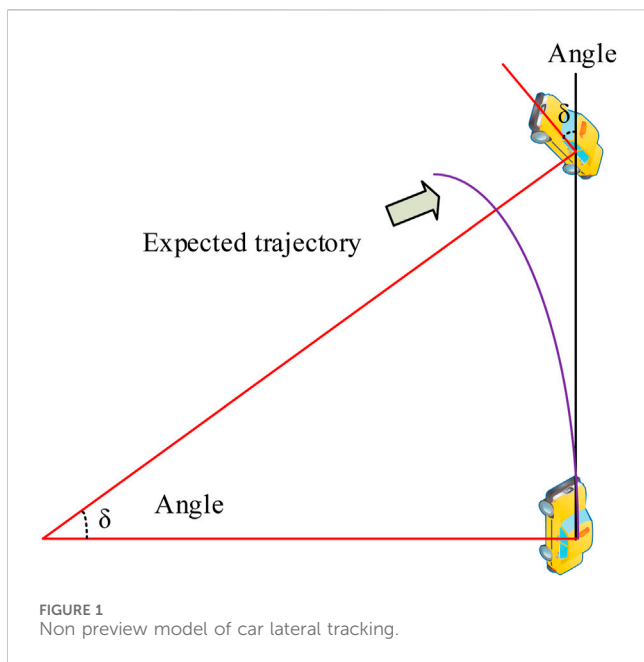
network mentioned in the research methodology shows high accuracy in cornering tests, but the stability and robustness in complex scenarios may still need to be improved. The research is specifically based on the theory of pre-aimed steering to model the lateral trajectory behaviour of a car, and then proposes an improved RBF network model to compensate the error of the lateral trajectory model to further improve the accuracy.

3 Intelligent vehicle lateral tracking algorithm on the grounds of improved RBF

In response to the current problem of lateral tracking errors in intelligent vehicles, this section proposes an intelligent vehicle lateral tracking algorithm on the grounds of improved radial basis function (RBF). This algorithm first models the lateral tracking behavior of vehicles on the grounds of the theory of preview steering, and then proposes an improved RBF network model to compensate for the error of the lateral tracking model, further improving accuracy.

3.1 Intelligent vehicle lateral tracking modeling on the grounds of preview theory

Non preview steering is a widely used lateral tracking control algorithm in the field of intelligent vehicle tracking. This algorithm is a steering control strategy that perceives the vehicle's state and environment in real time under specific assumptions, adjusts the vehicle's steering in real time, and drives it along a predetermined trajectory without the need to obtain and analyze future path information in advance. The geometric model of vehicle lateral motion under non preview steering is shown in Figure 1.

According to Figure 1, the non preview model ignores the steering system and suspension of the car, and instead uses the
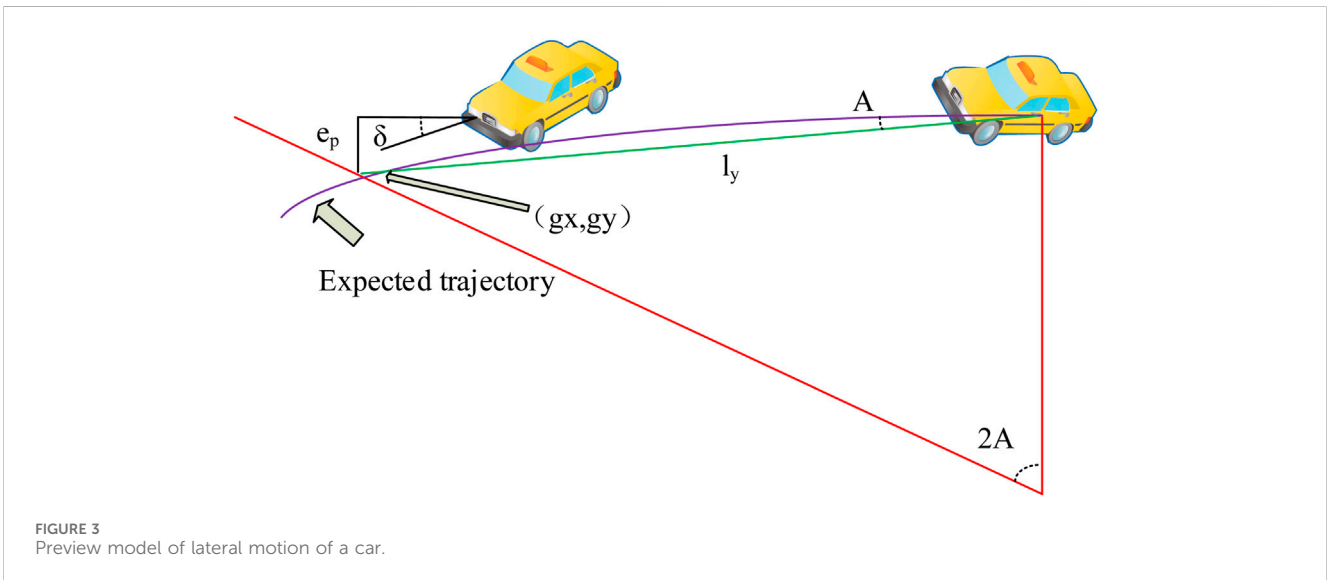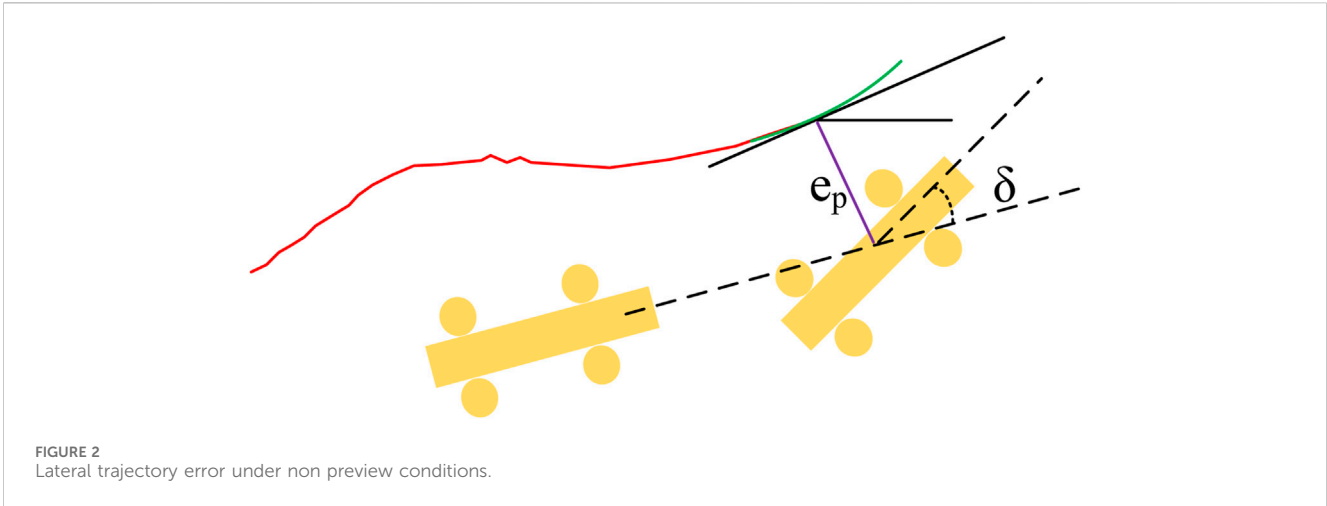
front wheel angle as the steering input, without considering the pitch and roll motion of the car. In addition, the influence of ground shear on car tires has also been ignored (Peng et al., 2022). The mathematical relationship between the curvature radius of the desired trajectory of a car's front wheel steering angle on the same road under this theory is shown in Formula 1.

$$\frac{L}{R} = \tan \delta \qquad (1)$$

In Formula 1, L represents the wheelbase of the car, R is the expected curvature radius, and $\tan \delta$ is the tan value of the front wheel angle of the car. The lateral trajectory error under the non preview model is shown in Figure 2.

According to the model in Figure 2, the tracking error of the front wheel angle of the car can be obtained as the difference between the car's yaw angle and its expected yaw angle. Under the non preview mechanism, there is a problem of significant decrease in vehicle control accuracy with increasing vehicle speed. Therefore, a vehicle lateral tracking model on the grounds of the preview mechanism is introduced here. The preview mechanism incorporates the concepts of preview point and preview distance on the basis of traditional steering mechanisms. The preview point refers to a point on the expected trajectory at a certain distance in front of the car. This point is usually a part of the expected trajectory, representing the position that the vehicle is about to reach in the future. The preview distance refers to the arc length distance between the point closest to the current point of the car on the expected trajectory and the preview point. This is not the length of the line connecting the preview point and the current point of the car, but the actual path distance along the trajectory. The geometric model of the vehicle's lateral motion under the preview steering mechanism is shown in Figure 3.

In Figure 3, l_y is the preview distance, and e_p is the tracking error at the preview point. (gx, gy) represents the position of the preview point in the current coordinate system. A longer preview distance (l_y) and smaller tracking error (e_p) can improve the tracking accuracy and stability of vehicles, but may also increase the complexity and response time of the system. The selection of preview distance l_y directly affects the response speed and tracking accuracy of the vehicle to changes in the road ahead. A shorter preview distance may cause the vehicle to be unable to respond to changes in the road ahead in a timely manner, while a longer preview distance may allow for advance planning of the driving path, but it may also introduce errors due to the uncertainty of road information. On straight roads, a longer preview distance may be more advantageous for improving tracking accuracy; In curved or complex road conditions, a shorter preview distance may be more helpful for vehicles to maintain stability and safety. The magnitude and trend of tracking error e_p will directly affect the vehicle's ability to correct deviations and the stability of driving. Larger e_p may take longer to correct, while smaller e_p may return to the expected path faster. According to Figure 3, the preview model calculates the geometric relationship between the preview point and the current vehicle position, and the system can plan and adjust the vehicle's steering angle in advance to better follow the desired trajectory. Compared to non preview systems, the advantage of preview steering is that it can better predict future paths, make steering decisions in advance, and achieve smoother and more

FIGURE 2
Lateral trajectory error under non preview conditions.

FIGURE 3
Preview model of lateral motion of a car.

efficient steering control. Therefore, in this study, the preview model was used to construct the intelligent vehicle lateral tracking algorithm. Under the preview mechanism, the expected steering angle of the front wheels of the car is shown in Formula 2.

$$arc \tan \frac{L}{R} = \delta \qquad (2)$$

Defining the turning angle as A, the mathematical expression of the turning angle is shown in Formula 3.

$$\sin A = \frac{e_p}{l_y} \qquad (3)$$

According to the definition of steering angle, Formula 4 can be obtained.

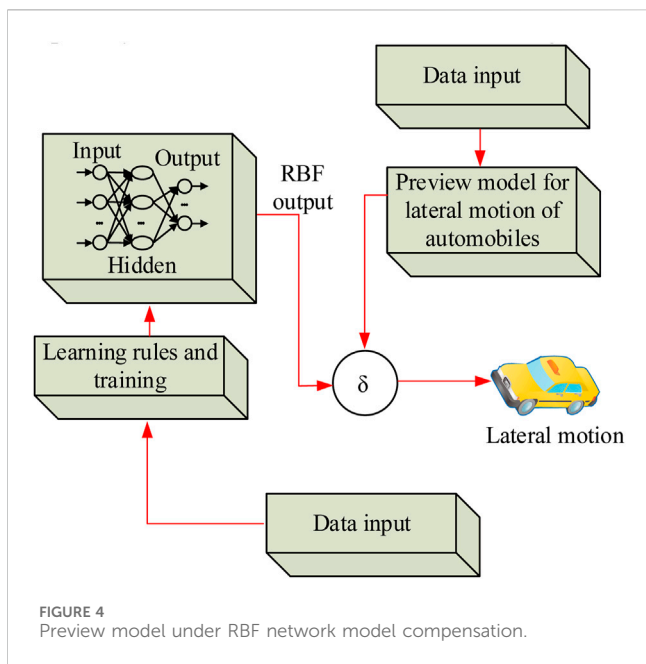$$arc \tan \frac{2L^* \sin A}{l_y} = \delta \qquad (4)$$

The lateral tracking of a car is influenced by many factors, among which the longitudinal speed has the most significant and

direct impact on the car's steering ability. When driving at high speeds, the inertia of the vehicle increases, and the lateral force required for lateral motion also increases. This results in a more sluggish response of the vehicle in the lateral direction, requiring stronger lateral control. In addition, high-speed driving of a car can also affect the response characteristics of the suspension system, which will directly affect the lateral stability and steering performance of the vehicle. Therefore, it is necessary to adjust the preview distance on the grounds of the vehicle speed, as shown in Formula 5.

$$l_y = vk \qquad (5)$$

In Formula 5, v is the longitudinal speed of the car, and v is the adjustment coefficient at the current speed. It substitutes the preview distance and turning angle into the expression of $\tan \delta$, and obtains Formula 6.

$$\delta = \arctan \left(\frac{Le_p}{\frac{1}{2}(vk)^2} \right) \qquad (6)$$

FIGURE 4
Preview model under RBF network model compensation.

Although the intelligent vehicle lateral tracking model on the grounds of the preview theory effectively explains the lateral tracking process of the vehicle, it may have significant errors when dealing with more complex nonlinear routes. On the one hand, terrain changes such as slopes and bumps can have nonlinear effects on the suspension system of vehicles, thereby affecting the accuracy of lateral motion simulation. On the other hand, when dealing with complex routes, the lateral dynamic characteristics of vehicles may become more complex, such as nonlinear tire forces and suspension feedback effects, which are difficult to incorporate into the current model.

3.2 Improved RBF network for tracking error compensation

Due to the existence of errors, a single preview model is obviously unable to accurately handle various complex situations on the road surface, which may pose potential risks to the driver's personal safety and traffic order. Therefore, it is necessary to try to eliminate such errors as much as possible. Due to the self-learning ability of neural networks, they can effectively cope with common road conditions through training with a large amount of nonlinear data. Therefore, neural network technology is used here to compensate for the error of the preview model (Sun et al., 2022). RBF neural network has excellent nonlinear approximation ability and can approximate any nonlinear function. This feature makes RBF neural network very suitable for dealing with nonlinear error problems in intelligent vehicle lateral tracking. The structure is relatively simple, the training process is concise, and the learning convergence speed is fast. This helps to achieve high-quality lateral tracking control effects within a limited time, meeting the real-time requirements of intelligent vehicles. RBF neural network achieves global optimality through local approximation, overcoming the problem of global approximation networks such as BP neural

network being prone to getting stuck in local optima. Although RBF neural networks perform well in lateral tracking control of intelligent vehicles, other types of neural networks may also achieve similar performance. For example, DNN, CNN, and other networks also have strong nonlinear approximation and generalization abilities, but these networks may require more data and computing resources during the training process, and the training time may be relatively long. The RBF network model adopted by the research institute is shown in Figure 4.

As shown in Figure 4, the output values of neurons in the RBF network are radially distributed in space, and the input can be mapped to a high-dimensional space. This feature enables RBF networks to better handle nonlinear problems, enabling RBFs such as Gaussian functions to capture the nonlinear features of input data (Sohrabi et al., 2023). In the proposed intelligent car lateral tracking algorithm, the output of the RBF network and the preview model jointly determine the direction and size of the front wheel steering angle of the car, thereby controlling the lateral tracking trajectory of the car. Assuming that the number of neurons in the input layer, hidden layer, and output layer is n, m, and p respectively, and the corresponding vectors are represented by X, R, and Y. Formula 7 describes the expression for calculating the output of the hidden layer.

$$R\left(\|C - X\|\right) = \exp\left[-\frac{(X - C)^{T+1}}{2\sigma_i^2}\right] \qquad (7)$$

In Formula 7, C represents the center of the basis function, whose dimension is always consistent with the input vector. σ_i^2 represents the square of the width of the I-th RBF. $\|C - X\|$ is the distance between two vectors. Formula 8 is the expression for the output layer result.

$$\sum_{i=1}^{m} w_{ib} R_i\left(X\right) = y_b \qquad (8)$$

b in Formula 8 is an integer with a value range between 1 and p. And w_{ib} is the weight between the neuron output and the hidden layer. m represents the total number of floors. Although having self-learning ability, traditional RBF neural networks are prone to falling into local minima without optimization. This may make it difficult for the model to achieve global optimum during the training process, thereby limiting algorithm performance. In response to this issue, the RBF network process has been optimized, as shown in Figure 5.

As shown in Figure 5, the study first introduced the Fast Density Peak Clustering (FDPC) algorithm in RBF. FDPC is a clustering algorithm that efficiently discovers density peaks in a dataset, divides data points into different clusters, and thus better initializes the parameters of the RBF network (Yang et al., 2022). The optimized parameters can prevent the algorithm from getting stuck in local minima and missing possible optimal values, thereby improving the global optimization ability of the model (Li et al., 2021; Zheng et al., 2024). This algorithm is used to process data streams, and the distance between nodes is shown in Formula 9.

$$\left(\sum_{i,j=1}^{} \left(x_i - x_j\right)^2\right)^{\frac{1}{2}} = d\left(x_i, x_j\right) \qquad (9)$$

In Formula 9, $d\left(x_i, x_j\right)$ is the distance between node x_i and node x_i. When the value of the distance is less than the set threshold, the

FIGURE 5
Optimized RBF network.
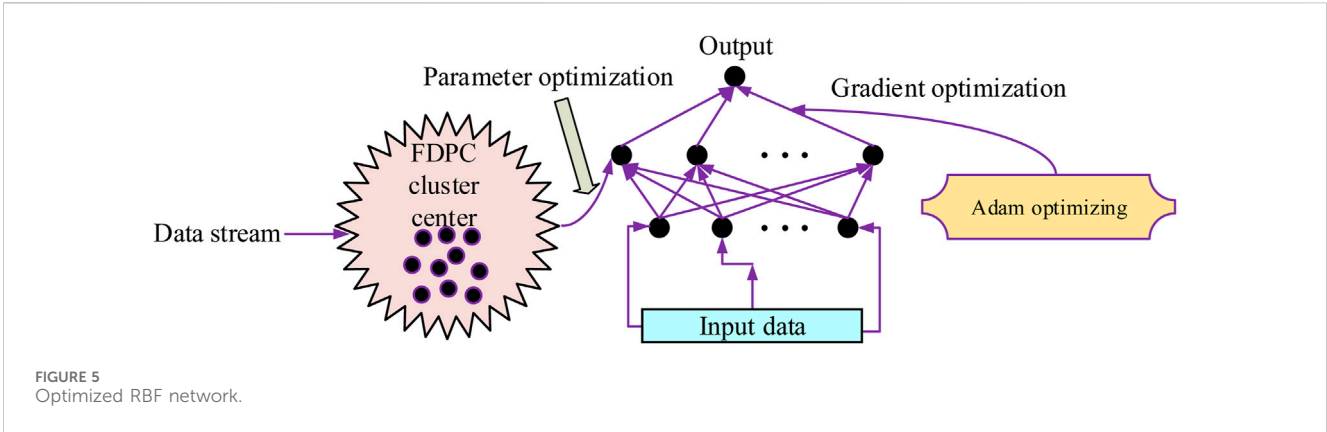
corresponding node is divided into the same cluster. The density of a cluster is determined by the number of nodes within it, as shown in Formula 10.

$$\chi(x)^\star \sum_j \chi\left(d_{ij} - d_c\right) = \rho_i \qquad (10)$$

In Formula 10, ρ_i represents the cluster density of cluster i. And $\chi(x)$ is a judgment function. Its value is fixed to 1 or 0. When the value of the judgment function is 1, node j is classified in cluster i, otherwise node j does not belong to that cluster. Due to the presence of outliers affecting the performance of FDPC, a new parameter is added here to identify outliers and process them, as shown in Formula 11.

$$\gamma = \min_{j: \ pi > pj} \left(d_{ij}\right)^\star \rho_i \qquad (11)$$

In Formula 11, γ is the parameter used to distinguish outliers, and the larger the value, the less likely the corresponding node is to be an outlier. In addition to FDPC, Adam optimization is also introduced in the output process of RBF network to modify the neural network. The Adam algorithm is a gradient based optimization algorithm that combines first-order moment estimation and second-order moment estimation of gradients. By adaptively adjusting the learning rate, it can more effectively optimize model parameters (He et al., 2023). In addition, compared to traditional gradient descent schemes, Adam avoids the problem of operators manually setting the descent step size and reduces the negative impact and bias caused by subjectivity (Al-Kasasbeh, 2022). The gradient estimation process of Adam optimization is shown in Formula 12.

$$g = \frac{\theta_{k-1} L\left(f\left(\hat{x}_i, \theta_{k-1}\right), y_i\right)}{m} \qquad (12)$$

In Formula 11, g represents gradient estimation. θ_{k-1} represents the learning parameter for the corresponding number of layers, while $\hat{x}_i$ represents the sample data. According to the results of gradient estimation, the accumulated gradient value can be obtained, and its mathematical process is shown in Formula 13.

$$\left(-g(\alpha - 1)\right) + \alpha AV_{t-1} = AV_t \qquad (13)$$

In Formula 13, AV accumulates the sign of the gradient, and α represents the momentum parameter. Under the current Adam

setting, the search speed for the minimum gradient is often slower, so the gradient is squared to speed up the search. The calculation process of this processing is shown in Formula 14.

$$Tr_k = \rho^\star Tr_{k-1} - g^{2\star}(\rho - 1) \qquad (14)$$

The Tr in Formula 14 represents the accumulated square gradient. After squared processing, further moment estimation correction is added to optimize the descent speed, as shown in Formula 15.

$$Tr_t^{'} = \frac{r_t}{1 - \rho} \qquad (15)$$

In Formula 15, $Tr_t^{'}$ is the corrected cumulative square gradient. At this point, the construction and optimization process of the RBF network has been completed. The overall process of the proposed optimized RBF network is shown in Figure 6.

As shown in Figure 6, the algorithm first uses the cluster centers obtained by the FDPC algorithm as the initial parameters of the RBF neural network. Then, the RBF neural network is trained, using sample data for forward and backward propagation, and continuously updating weight parameters through Adam algorithm to optimize network performance. The role of Adam algorithm is to optimize the calculation gradient, cumulative gradient, and cumulative squared gradient of RBF neural network, and perform parameter correction. Finally, the network parameters are continuously updated through iteration until the stopping condition is met or the specified number of iterations is reached. This combination of clustering and optimization is aimed at overcoming the problem of RBF neural networks easily falling into local minima in the initial state, and improving the global optimization ability of the model. After training, the proposed RBF model is introduced into the intelligent vehicle lateral tracking algorithm to compensate for the errors generated by the preview model.

Label noise refers to errors or inconsistencies in the labels in a dataset. In the field of intelligent vehicles and autonomous driving, this leads to models learning incorrect driving behaviors or patterns, thereby affecting the safety and accuracy of autonomous driving. Transfer learning can transfer knowledge learned from one task to another related task. In the presence of label noise, a relatively clean dataset (source domain) can be used to pre train the model, and then
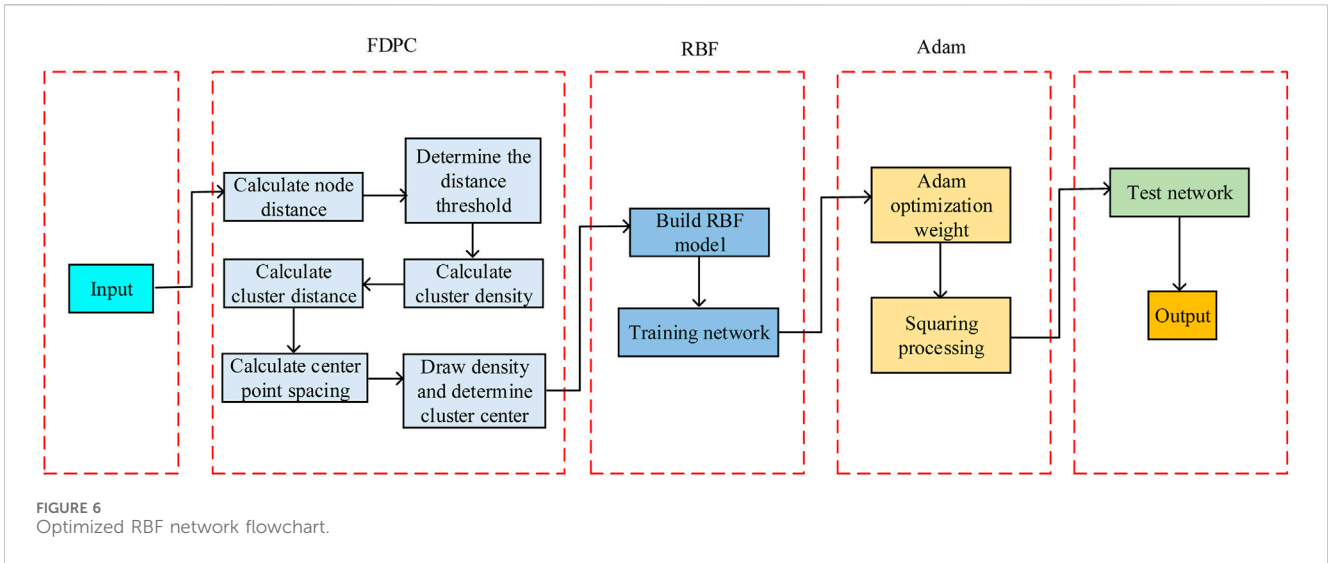
FIGURE 6
Optimized RBF network flowchart.

transferred to the target domain (i.e., dataset with label noise) for fine-tuning. In this way, the model can utilize knowledge from the source domain to improve performance in the target domain (Bailin et al., 2022). Data dispersion refers to the uneven distribution of data in the feature space, resulting in poor performance of the model in certain regions. In the field of intelligent vehicles and autonomous driving, data dispersion may arise from various factors such as road conditions, driving behavior, and vehicle types. For the problem of data dispersion, distribution calibration techniques can be used to mitigate its impact. For example, the data distribution of each category can be estimated and resampled or weighted to balance the data distribution between different categories. This method helps the model better learn the boundaries between different categories (Yang et al., 2021).

4 Simulation testing of RBF optimized lateral tracking algorithm for intelligent vehicles

This section uses simulation software to test the proposed intelligent vehicle lateral tracking algorithm. The testing is mainly divided into two parts. In the first part, the study tested the proposed optimized RBF network and evaluated its performance from different perspectives. In the second part, the RBF optimized intelligent vehicle lateral tracking algorithm was tested and simulated using simulation car dynamics software and road models.

4.1 Optimizing testing of RBF networks

The software and hardware environment and experimental parameters used in this experiment are shown in Table 1. The hardware used in the experiment includes an Intel Core i5 13, 400 F CPU, 32 GB DDR5 memory, and 4 TB hard drive, as well as an NVIDIA RTX 3070 GPU. The operating system is Windows 11 and the protocol is TCP/IP. In terms of automotive dynamics, the Virtual Dynamics Analysis of Vehicle (veDYNA) model was

TABLE 1 Experimental environment and related parameters.

Hardware		Software and environment	
Item	Detail	Item	Detail
CPU	I5 13400F	Protocol	TCP/IP
Memory	32G DDR5	Operating system	Windows 11
Hard disk	4T	Automotive dynamics model	veDYNA
GPU	RTX 3070	Data processing	Matblab

Parameters	
Item	Detail
Initial learning rate	0.70
Data set	UCI
Training set size	70
Test set size	40
Input layer unit amount	4
Output layer unit amount	1

used. Matlab is a high-performance mathematical calculation and simulation software that provides a rich library of mathematical functions and toolboxes. It can easily handle complex mathematical operations and data analysis, making it very suitable for virtual dynamic analysis of vehicles. The veDYNA model can accurately simulate the dynamic characteristics of vehicles under different conditions, including acceleration, braking, steering, and the interaction between vehicles and the ground. By combining Matlab and veDYNA models, it is possible to simulate vehicle dynamics characteristics that are close to the real world, including the dynamic response of vehicles, the impact of road conditions on vehicle performance, and so on. The data processing tool uses Matlab. The specific parameter settings for the experiment include an initial learning rate of 0.70, UCI dataset selection, training

FIGURE 7
Algorithm CPU runtime. **(A)** Training. **(B)** Testing.
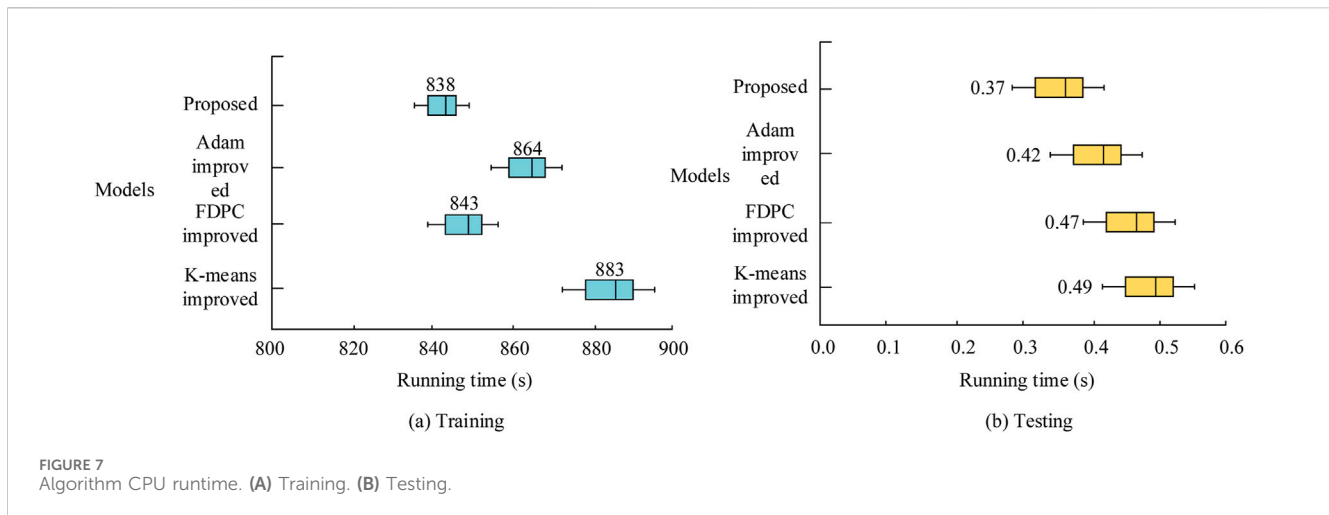
set size of 70, test set size of 40, input layer containing 4 units, and output layer containing 1 unit. These configurations will be used for conducting experiments and evaluating results.

Compared with traditional model-based control methods, neural networks focus more on data-driven approaches. As long as there is sufficient data, neural networks can learn control strategies through training without the need to establish precise mathematical models. The experiment uses a driving simulator to simulate driving. By simulating different road and traffic conditions, the driving data of vehicles in various situations is collected, and neural network training is carried out through this data to obtain the training data of the neural network.

To improve the reliability and comparability of the experiment, three similar algorithms were used as controls to evaluate the performance of the proposed algorithm. The three comparison algorithms are K-means Improved Radial Basis Function (K-means Improved RBF), Adam Improved Radial Basis Function (Adam Improved RBF), and Fast Density Peak Clustering Improved Radial Basis Function (FDPC Improved RBF). K-means improved RBF is a cutting-edge and widely used optimized RBF network in this field. In addition, RBF optimized using Adam and FDPC separately was also utilized to compare the proposed combination optimization scheme. Firstly, the proposed optimized RBF network was tested for its Central Processing Unit (CPU) runtime on the UCI dataset, as shown in Figure 7. Figure 7A shows the data of the algorithm during the training phase, and Figure 7B shows the data of the algorithm during the testing phase. In the two stages of training and testing, there was no significant difference in the standard deviation of the running time of each algorithm, so the analysis was mainly conducted from the perspective of average time. During the training phase, the average running time of the proposed algorithm is 838 s, which is the shortest among the four algorithms. The longest duration is K-means improved RBF, which takes an average of 883 s during the training phase. During the testing phase, the average time taken by the proposed algorithm was 0.37 s, while the average time taken by Adam improved RBF was 0.42 s. The other two algorithms took longer than this value.

After comparing the CPU running time, the next step is to compare the accuracy of each algorithm in testing with the number of iterations. The comparison results are shown in Figure 8.

According to the test results, after 20 iterations, the proposed algorithm consistently shows the highest accuracy at the same number of iterations. When the number of iterations reaches 370, the accuracy of the algorithm stabilizes at 88%. In contrast, the FDPC improved RBF algorithm achieved an accuracy of 79% at this iteration, while the accuracy of the other two algorithms was lower than this value. The trend of image changes shows that the proposed algorithm has the most significant increase in accuracy at different iteration stages, and its performance is more outstanding compared to the other three algorithms. Among the four algorithms, the Adam improved RBF algorithm showed the earliest convergence trend, but the final accuracy was only 72%, which is more than 10% lower than the proposed algorithm. This emphasizes the advantages of the proposed algorithm in terms of convergence speed and final performance. This experimental result provides strong support for algorithm performance.

4.2 Simulation testing of intelligent vehicle lateral tracking algorithm

After testing the performance of the proposed RBF network, it is necessary to further test the performance of the intelligent vehicle lateral tracking algorithm that integrates the RBF network. The first project to be tested at this stage is the stability of the RBF network combined with the preview model. Real road surface data was used as standard values here to fit the proposed algorithm for testing. For the convenience of comparative research, the current K-means improved RBF network is still used as a comparison, and the results are shown in Figure 9. Figure 9A shows the running results of the proposed algorithm, while Figure 9B shows the comparison results of the K-means improved RBF network. From the trend of curve changes, the proposed algorithm's curve trend is closer to the standard value. It shows slightly lower than the standard value at some extreme points, but can fit the trend more completely. The comparison results of K-means improved RBF networks show a greater difference. The K-means improved RBF network has a low degree of fit with the standard value data under these conditions, and the error at the extreme points is also greater than the fitting results of the proposed algorithm.
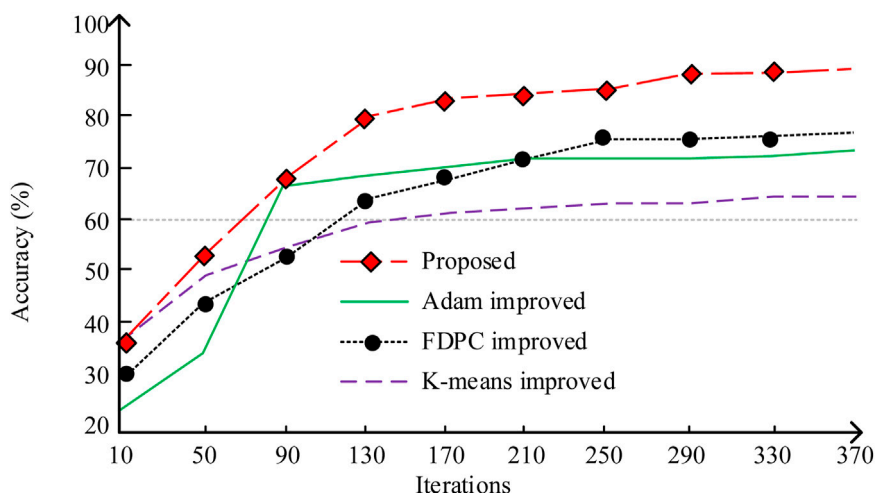
FIGURE 8
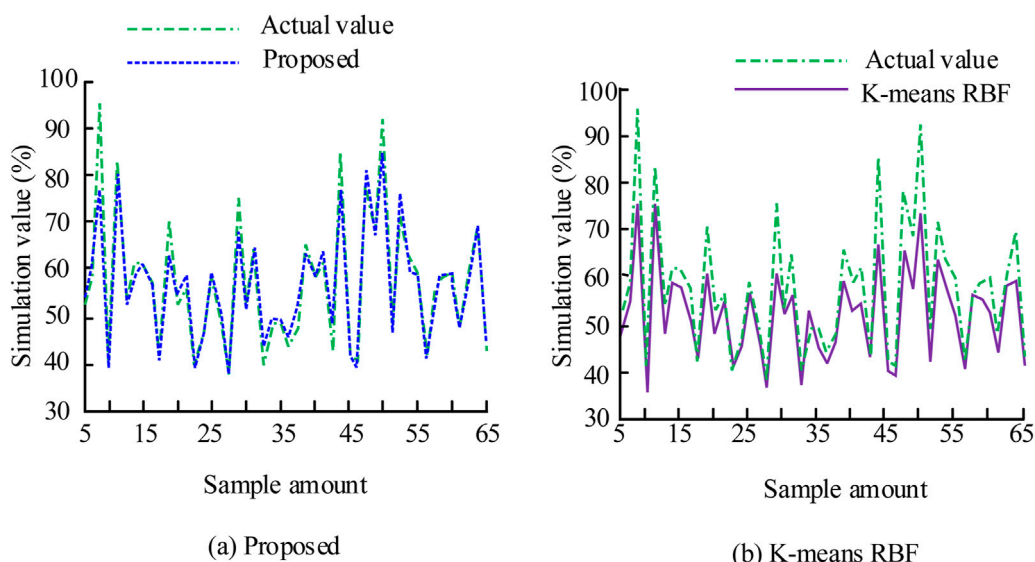Changes in algorithm accuracy under test set.

FIGURE 9
Comparison of algorithm stability. **(A)** Proposed. **(B)** K-means RBF.

Due to significant differences in the steering ability and handling of vehicles at different longitudinal speeds, it is necessary to test the output error of the proposed algorithm at slow and fast speeds to evaluate its usability and safety. Here, vehicle speeds of 5 m/s and 20 m/s were selected as experimental standards for different vehicle speeds, and the test results are shown in Figure 10. Figure 10A shows the lateral tracking error of the algorithm at slow speeds, and Figure 10B shows the lateral tracking error of the algorithm at fast speeds. This indicates that at slow speeds, the pre aiming trajectory without neural network compensation exhibits the maximum error, with a maximum absolute value of 0.1 m. The proposed algorithm has the smallest overall error, with a maximum absolute value of only 0.028. The algorithm error under K-means RBF compensation is between the two. At fast speeds, although the

curve shape and trend are more complex, the error ranking of the three algorithms is similar to that at slow speeds. The overall error of the proposed algorithm is the smallest, with a maximum absolute error of 0.08 m, while the maximum absolute error of the preview tracking without neural network compensation is 0.14 m.

In actual autonomous driving scenarios, continuous turns pose significant challenges to autonomous driving algorithms. It has continuous changes in curvature and radius, which requires automatic tracking algorithms to adapt and adjust the vehicle's driving trajectory in a timely manner. In order to further verify the practicality of the algorithm, a continuous road surface was simulated and tested using the proposed algorithm and existing algorithms. The existing algorithm is a tracking method proposed in reference (Liu et al., 2021) that uses an odorless Kalman filter to fuse
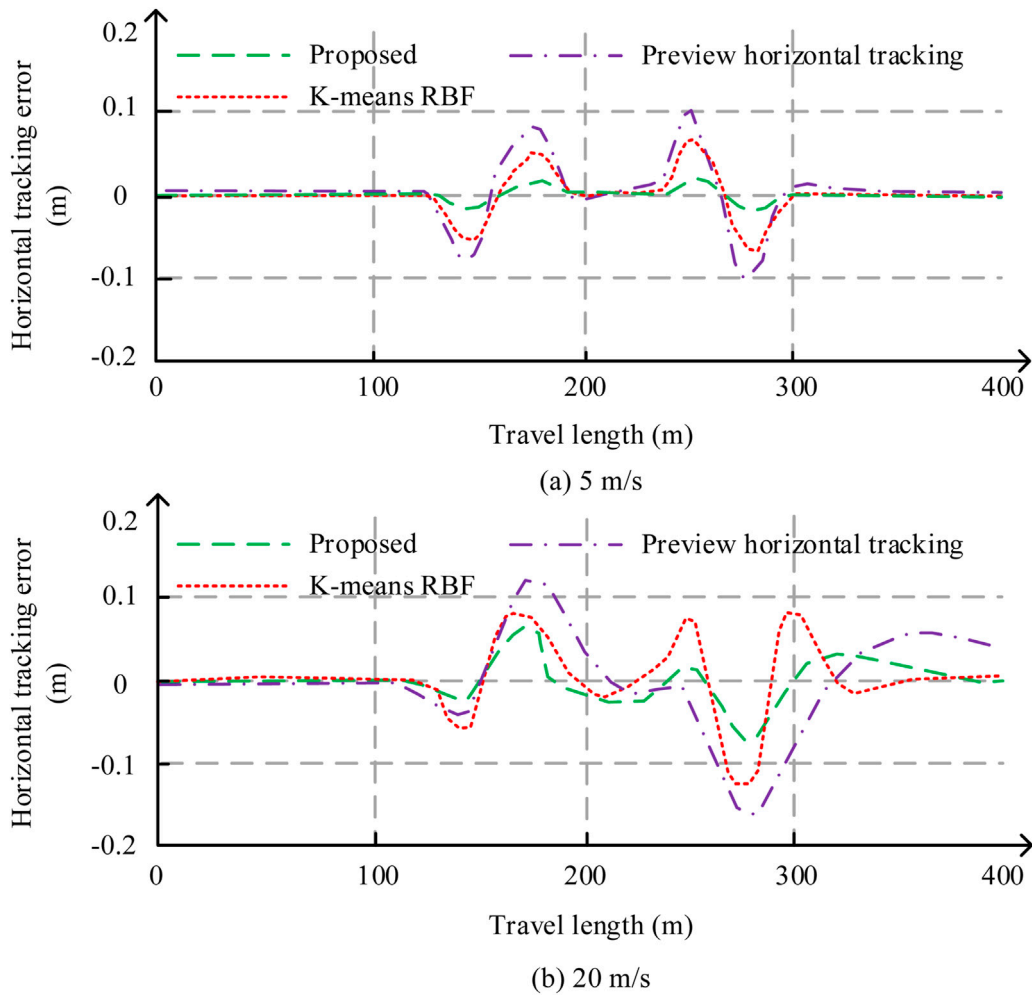
FIGURE 10
Error comparison of the algorithm at different vehicle speeds. **(A)** 5 m/s. **(B)** 20 m/s.
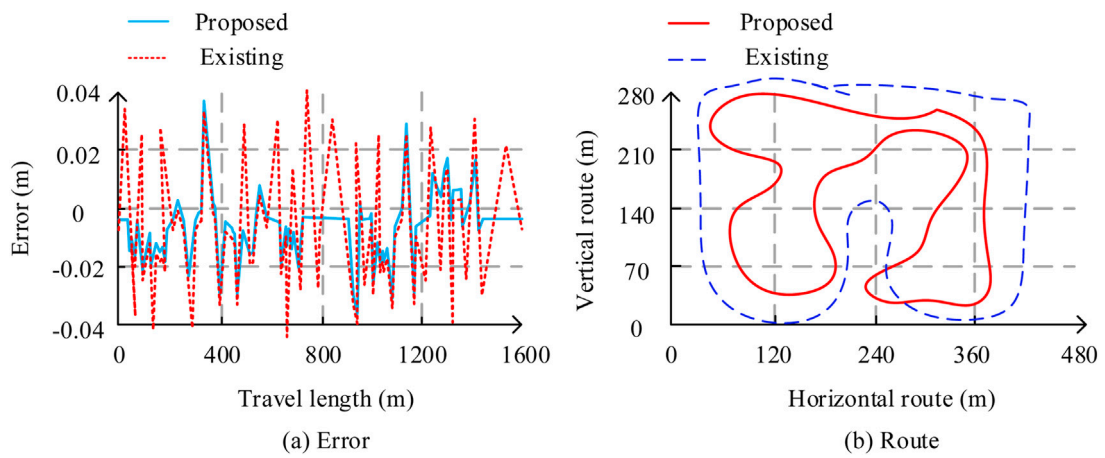

FIGURE 11
Performance of the tracking algorithm under continuous bends. **(A)** Error. **(B)** Route.

radar and LiDAR data. The results are shown in Figure 11. Among them, Figure 11A is a topographic map of continuous curves. It can be seen that the error of the proposed algorithm still fluctuates to some extent during the turning process, but the overall error has been stably controlled. However, the existing methods have frequent and unstable fluctuations, resulting in overall instability. Figure 11B shows the error situation of the proposed algorithm during the driving process. The maximum error does not exceed 0.04 m, and considering the width of a normal road surface, this error is a relatively safe range in actual driving. However, the tracking of existing methods during continuous turns is a rough process, and the tracking is not precise enough.

Complex terrain such as mountains, hills, canyons, etc., with significant terrain undulations and elevation differences, may cause multipath effects, diffraction, scattering, and other phenomena in wireless signal propagation, thereby affecting positioning accuracy. The initialization parameters of RBF neural network have a significant impact on algorithm performance. If the initialization parameters are not selected properly, it may cause the algorithm to fall into local minima in the early stages of training, thereby affecting the final positioning accuracy. Increasing the density of sensor nodes in complex terrain or high-altitude areas can improve network connectivity and data redundancy, thereby enhancing the robustness and positioning accuracy of the algorithm. Combining the RBF neural network positioning algorithm with other positioning technologies such as GPS and inertial navigation to achieve multi-source fusion positioning. By integrating the advantages of various positioning technologies, the overall positioning accuracy and reliability can be improved.

The research results can be disseminated through news media, social media, and other channels, including press releases, videos, graphics, etc., to increase public interest and attention to autonomous driving technology. Carry out science popularization and education activities on autonomous driving technology, such as holding lectures, seminars, workshops, etc., to popularize the principles, applications, and development prospects of autonomous driving technology to the public.

5 Conclusion

Intelligent vehicles and their autonomous driving are the current development direction of the automotive industry, but the current autonomous driving algorithms have significant errors in lateral tracking. A smart car lateral tracking algorithm combining RBF neural network and preview tracking mechanism is proposed to address this issue. According to the simulation test results, the proposed algorithm has the smallest overall error at slow driving speed, with a maximum absolute value of only 0.028 m. At this point, the maximum error of the algorithm without neural network compensation reached 0.1 m. At fast speeds, the proposed algorithm has the smallest overall error, with a maximum absolute error value of 0.08 m, while the maximum absolute error value of preview tracking without neural network compensation is 0.14 m. In addition, in complex terrain with continuous bends, the maximum error of the proposed algorithm does not exceed 0.04 m. Considering the width of normal road surfaces, this error is a relatively safe range of error in actual driving. In theory, this study introduces attention mechanisms, feature fusion, and other strategies to further reveal the internal working

mechanism of neural networks in processing complex information, providing a new theoretical perspective for designing and optimizing neural networks. In practice, the application of attention mechanism, feature fusion, regularization and optimization strategies has enhanced the adaptability of the auto drive system to complex and changing environments, improved the robustness and generalization ability of the system, accelerated the research and development and optimization process of automatic driving technology, promoted the commercial application of automatic driving technology, and provided strong support for the development of intelligent transportation and smart cities. The car dynamics model used in this modeling is on the grounds of the preview mechanism, and there is a lot of idealization and simplification in the modeling and restoration of vehicle and road conditions, which to some extent limits the accuracy and leads to errors. Therefore, in future research, more parameters and models will be added to more accurately simulate vehicle and road conditions, further reducing errors.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

YS: Conceptualization, Methodology, Software, Writing–original draft, Writing–review and editing. LX: Data curation, Investigation, Writing–review and editing. JL: Data curation, Formal Analysis, Software, Validation, Visualization, Writing–review and editing.

Funding

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abadi, A. D., Gu, Y., Goncharenko, I., and Kamijo, S. (2023). Detection of cyclist's crossing intention based on posture estimation for autonomous driving. *IEEE sensors J.* 23 (11), 11274–11284. doi:10.1109/jsen.2023.3234153

Al-Kasasbeh, B. (2022). A modified on-demand vector distance routing protocol incorporating alternate vehicle-RSU-vehicle and vehicle-RSU- RSU paths. *Inform. Slov.* 46, 1–11. doi:10.31449/inf.v46i6.4226

Bailin, F., Yi, Z., Ye, C., and Lingbei, M. (2022). Intelligent vehicle lateral control based on radial basis function neural network sliding mode controller. *CAAI Trans. Intell. Technol.* 7 (3), 455–468. doi:10.1049/cit2.12075

Bao, H., Zhu, H., and Liu, D. (2024). Improved SSA-RBF neural network-based dynamic 3-D trajectory tracking model predictive control of autonomous underwater vehicles with external disturbances. *Optim. Control Appl. Methods* 45 (1), 138–162. doi:10.1002/oca.3050

Bhosle, K., and Musande, V. (2023). Evaluation of deep learning CNN model for recognition of devanagari digit. *Artifcial Intell. Appl.* 1 (2), 114–118. doi:10.47852/bonviewaia3202441

El-Bayoumi, A. (2021). ISO-26262 compliant Safety-Critical autonomous driving applications: real-Time Interference-Aware multicore architectures. *Int. J. Saf. Secur. Eng. An Interdiscip. J. Res. Appl.* 11 (1), 21–34. doi:10.18280/ijsse.110103

Fu, S., and Fu, H. (2023). Modeling and TOPSIS-GRA algorithm for autonomous driving Decision-Making under 5G-V2X infrastructure. *Comput. Mater. and Continua* 21 (4), 1051–1071. doi:10.32604/cmc.2023.034495

Halilaj, L., Luettin, J., Monka, S., Henson, C., and Schmid, S. (2023). Knowledge Graph-Based integration of autonomous driving datasets. *Int. J. Semantic Comput.* 17 (2), 249–271. doi:10.1142/s1793351x23600048

He, F., Guan, X., Jin, H., and Duan, C. (2023). Intelligent vehicle longitudinal acceleration control method based on feedforward and feedback combination. *Proc. Institution Mech. Eng.* 237 (5), 1065–1081. doi:10.1177/09544070221085037

Hoffmann, C., and Klein, N. (2023). Marginally calibrated response distributions for end-to-end learning in autonomous driving. *Ann. Appl. Statistics* 17 (2), 1740–1763. doi:10.1214/22-aoas1693

Li, R., Li, F., Wu, C., and Song, J. (2021). Research on vehicle network security situation prediction based on improved CLPSO-RBF. *J. Phys. Conf. Ser.* 1757 (1), 012148–8. doi:10.1088/1742-6596/1757/1/012148

Liu, Z., Cai, Y., Wang, H., and Chen, L. (2021). Surrounding objects detection and tracking for autonomous driving using LiDAR and radar fusion. *Chin. J. Mech. Eng.* 34 (5), 117–180. doi:10.1186/s10033-021-00630-y

Matveev, K. I. (2020). Modeling of autonomous hydrofoil craft tracking a moving target. *Unmanned Syst.* 8 (2), 171–178. doi:10.1142/s2301385020500107

Peng, Y., Tan, G., Si, H., and Li, J. (2022). DRL-GAT-SA: deep reinforcement learning for autonomous driving planning based on graph attention networks and simplex architecture. *J. Syst. Archit.* 126, 102505–102511. doi:10.1016/j.sysarc.2022.102505

Sohrabi, P., Shokri, B. J., and Dehghani, H. (2023). Predicting coal price using time series methods and combination of radial basis function (RBF) neural network with time series. *Mineral. Econ.* 36 (2), 207–216. doi:10.1007/s13563-021-00286-z

Stocco, A., and Tonella, P. (2022). Confidence-driven weighted retraining for predicting safety-critical failures in autonomous driving systems. *J. Softw. Evol. Process* 34 (10), 2386–2411. doi:10.1002/smr.2386

Sun, Y., Xu, J., Lin, G., Ji, W., and Wang, L. (2022). RBF neural network-based supervisor control for maglev vehicles on an elastic track with network time delay. *IEEE Trans. Industrial Inf.* 18 (1), 509–519. doi:10.1109/tii.2020.3032235

Sunan, H., Huat, S. R. T., and Kiong, K. T. (2019). Collision avoidance of multi unmanned aerial vehicles: a review. *Annu. Rev. Control* 48, 147–164. doi:10.1016/j.arcontrol.2019.10.001

Wang, Y., Zheng, K., Tian, D., Duan, X., and Zhou, J. (2021). Pre-training with asynchronous supervised learning for reinforcement learning based autonomous driving. *Front. Inf. Technol. and Electron. Eng.* 22 (5), 673–686. doi:10.1631/fitee.1900637

Yang, T., Bai, Z., Li, Z., Feng, N., and Chen, L. (2021). Intelligent vehicle lateral control method based on feedforward + predictive LQR algorithm. *Actuators* 10 (9), 228. doi:10.3390/act10090228

Yang, Y., Jian, W., Xin, L., Ying, Z., and Shuai, W. (2022). A lateral trajectory tracking control method for intelligent commercial vehicles considering active anti-roll decision based on Stackelberg equilibrium. *IET Intell. Transp. Syst.* 16 (9), 1193–1208. doi:10.1049/itr2.12202

Yasin, J. N., Mohamed, S. A. S., Haghbayan, M. H., Heikkonen, J., Tenhunen, H., and Plosila, J. (2020). Unmanned aerial vehicles (uavs): collision avoidance systems and approaches. *IEEE access* 8, 105139–105155. doi:10.1109/access.2020.3000064

Zhao, Z. W., Yuan, K., Du, J. T., Wang, Y. L., Huang, Y. J., and Chen, H. (2023). Planning and control of autonomous driving in lane-change manoeuvre based on MPC: a framework and design principles. *Int. J. Veh. Des.* 9 (2), 357–381. doi:10.1504/ijvd.2023.134754

Zheng, A. Z., Ye, Z., and Zheng, X. (2024). Intelligent vehicle lateral control strategy research based on feedforward + predictive LQR algorithm with GA optimisation and PID compensation. *Sci. Rep.* 14 (1), 22317. doi:10.1038/s41598-024-72960-5