Intelligent Car Autonomous Driving Tracking Technology Based on Fuzzy Information and Multi-sensor Fusion

Dongxuan Gao[1*], Jing Wang[2], Rui Chai[3]
[1]School of Traffic Engineering, Sichuan Polytechnic University, Deyang 618000, China
[2]School of Automotive & Transportation Engineering, Shenzhen Polytechnic University, Shenzhen 518055, China
[3]Data Science and Technology, North University of China, Taiyuan 030051, China
E-mail: gaodongxuanxuan@163.com
[*]Corresponding author

Keywords: fuzzy information, fuzzy neural programming network, multi-sensor fusion, autonomous driving, vehicle tracking

Received: July 10, 2024

Car autonomous driving technology is a key development direction in the future, but road environment data is relatively complex, and single sensor-driven autonomous driving algorithms have the problem of insufficient trajectory prediction accuracy. In response to this, this study uses fuzzy information to construct a fuzzy neural programming network for data processing and uses a fuzzy sample dataset for training. The multi-sensor autonomous driving tracking algorithm is used to fuse and calculate the data collected by the sensors. The data are transformed into a Cartesian coordinate system to calculate the predicted lateral and longitudinal distance, velocity, and acceleration of the tracking algorithm. The experiments were conducted using an 80 Hz radar sensor, a 72-line lidar, and a 1080p vision sensor. Two sets of each of the three roadway test environments were used. The initial distance between the experimental vehicle and the target vehicle was 30m. The results demonstrated that the distance and speed predicted by the driving tracking algorithm were consistent with the true values, and the longitudinal distance was 0.34 m and 0.28 m higher than the square root Kalman filtering algorithm and the extended Kalman filtering algorithm, respectively. The longitudinal velocity denoising accuracy of the algorithm has been improved by 18.65% and 31.27% compared to other algorithms. Therefore, the autonomous driving tracking algorithm was better able to maintain a safe distance than other algorithms, and its acceleration changes were smoother, improving the ride comfort and safety of the vehicle. The tracking algorithm has a stronger ability to remove noise from sensors and can adapt to more diverse environments. The designed fuzzy information and multi-sensor fusion tracking algorithm for car autonomous driving provides a reference for subsequent research on automotive autonomous driving technology.

Povzetek: Raziskava uvaja tehnologijo sledenja za avtonomno vožnjo na osnovi nejasnih informacij in združevanja več senzorjev, kar povečuje kvaliteto sledenja in izboljšuje varnost ter udobje vožnje.

1 Introduction

In recent years, with the gradual increase in per capita car ownership, the number of traffic accidents caused by factors such as distracted driving or violating traffic rules is constantly increasing [1]. The increase in vehicles has led to traffic congestion and the efficiency of road traffic is constantly decreasing. As of January 2024, the amount of motor vehicles in China has reached 435 million, and accidents caused by driver issues account for 94% of the total accidents [2]. The deep improvement of information technology has made assisted driving and even intelligent Car Autonomous Driving (CAD) technology a hot research direction in intelligent transportation [3]. Intelligent driving technology can not only improve road traffic efficiency, but also reduce accident rates, liberate productivity, improve people's work efficiency and travel comfort, and ensure people's travel safety [4]. Both domestic and foreign technology companies have invested plenty of research and development funds in intelligent driving, promoting the continuous progress of intelligent vehicles. However, the existing intelligent CAD technology still has some problems in data acquisition and analysis of multiple targets, such as single data acquisition mode, inability to effectively remove environmental noise interference, and low data accuracy after data fusion of different sensors. Meanwhile, prolonged occlusion of sensors or the presence of multiple vehicles with similar characteristics in the surrounding area can significantly impair the accuracy of target tracking and trajectory prediction, thereby compromising the safety of autonomous driving. Regarding the issue of sensor data fusion, Nathan et al. found that excessive task volume in autonomous driving can lead to imbalanced learning outcomes, and therefore

proposed a deep multitasking learning model. The model received input data from different sensors in multiple locations of the vehicle and used intermediate sensors for data fusion and semantic segmentation. It could also have good performance with fewer parameters, occupy less memory during data processing, and have faster computing speed [5]. Senel et al. discovered the importance of sensor data in intelligent transportation and proposed a modular sensor data fusion framework. This framework adopted classic data fusion algorithms, using coordinate change modules and object association modules. This algorithm had low requirements for the number and type of sensors, and the data fusion time was much less than 10ms. It could quickly detect changes in the external environment and identify sensor faults [6]. Zhang et al. developed a new visual localization method to solve the problem of insufficient localization accuracy of autonomous vehicles. This method used the results of vehicle inertial motion and visual sensor image sequences to construct a semantic map, and then compared the map with the database, using neural networks to simplify the map-matching problem. The vertical and horizontal positioning errors of this method were 0.017 m and 0.039 m, respectively, and the positioning accuracy met the predetermined requirements [7]. Wang et al. proposed a fusion method that optimizes complementary sensors to address the offset problem in autonomous driving technology. This method established a global pose map through a multi-level optimization strategy, obtained the pose map using radar sensors and visual sensors, used GPS geographic information, and corrected cumulative offsets in the optimization layer. The average offset error of this method was 0.804%, and the average rotation error was 0.0043 m [8].

Given the problem of multi-sensor data acquisition, Zaghari et al. found that ordinary methods could not accurately detect the speed change of autonomous vehicles, so they proposed a hybrid method of fuzzy algorithm and NMS algorithm. This method was trained by establishing a set of safe driving information under different conditions, and continued obstacle detection using non-maximum threshold fuzziness. This method effectively improved detection accuracy and increased detection speed by 64.41% [9]. Awad et al. found that noise interference greatly affected the path planning of autonomous driving, and therefore proposed a comprehensive path-tracking control strategy for autonomous driving. This strategy used a fuzzy logic switching system to design and modify the linear model control parameters to obtain the optimal vehicle turning angle and angular velocity. This strategy could effectively eliminate noise interference in the data acquisition process, significantly improve path planning accuracy and calculation speed, and perform significantly better than other tracking control methods [10]. Sun et al. developed a new dual hidden layer network for path tracking in autonomous driving. This network outlined the lateral dynamics and steering angular velocity of autonomous vehicles during path tracking through their kinematic models. This network had higher robustness and faster convergence speed compared to traditional methods [11].

In summary, existing methods have studied the issues of multi-sensor data acquisition and fusion, as well as sensor noise interference removal in intelligent vehicles from multiple aspects, and have achieved certain results. However, the accuracy of autonomous driving path planning can no longer meet the growing demand. Therefore, this study improves the autonomous driving algorithm and innovatively introduces fuzzy information to construct a Fuzzy Neural Network (FNN), converting the vertical coordinate system of the vehicle into Cartesian coordinates. The Automatic Driving Tracking Algorithm (ADTA) fuses and analyzes the data from different sensors to calculate the position correlation between the intelligent car and the target car. ADTA aims to improve the accuracy of multi-sensor data fusion, and enhance the driving safety and riding comfort of intelligent vehicles. This study has three parts in total. Part 1 optimizes data collection and data fusion methods. Part 2 evaluates the performance of optimization algorithms. Part 3 summarizes the paper and provides the prospects for future directions.

The current research is limited by certain shortcomings. When some of the sensors are occluded for an extended period or when there are multiple vehicles in the surrounding area that are similar to the target vehicle, the accuracy of target tracking and trajectory prediction is significantly diminished. This negatively impacts the safety of autonomous driving. This research aims to fill the gap in multi-sensor data fusion and noise interference removal. Based on the above relevant studies, Table 1 is summarized, which summarizes the topics of related studies, the main index methods, and the shortcomings of related studies.

Table 1: Summary of relevant studies

Author	Research theme	Main index	Mthod	Insufficient
Natan O et al. [5]	Balance of learning effect	Memory usage and computing speed	Multi-sensor data collectionand intermediate sensor data fusion	The requirements on parameters are high
Senel N et al. [6]	Sensor data fusion	Data fusion time	Coordinate changes, object associations	Performance may be limited when the target size is large

Zhang Z et al. [7]	Positioning accuracy	Longitudinal and lateral positioning errors	Construct semantic map and simplify map matching	High requirements on the database
Wang K et al. [8]	Autopilot offset	Rotation error and offset error	Global pose map and cumulative offset correction	The multi-level computing time is too long
Zaghari N et al. [9]	The speed of the self-driving car changes	Detection accuracy and Detection speed	Non-maximum threshold ambiguity	The requirements for safe driving information collection are higher
Awad N et al. [10]	Noise removal	Path planning accuracy and calculation speed	Fuzzy logic switching system and linear model control parameters	Face challenges High precision of control parameters is required
Sun Z et al. [11]	Autopilot path tracking	Robustness and speed of convergence	The double hidden layer neural network plans lateral dynamics and steering angular velocity	The prediction accuracy is low when the scene is complex
This study	Multi-sensor fusion and automated driving tracking	Speed prediction accuracy and denoising performance	Construct FNN and multi-sensor data fusion	/

2 Methods and materials

2.1 Intelligent vehicle driving tracking technology based on fuzzy information

The tracking technology of intelligent CAD mainly focuses on speed planning research. Based on the vehicle's surrounding environment and its state, the next step of the vehicle's motion trajectory is to calculate immediately. Fuzzy information is used to construct an FNN for car speed planning. In the offline learning stage, FNN extracts rich experience from manual driving and transforms it into a dataset of fuzzy rules. FNN constructs a triple FNN model by continuously learning fuzzy rules. The speed planning of the model is described as the mapping relationship between the collected state variables and the output data. The car captures the velocity and distance information of obstacles while driving, and the distance X and speed V are fuzzified into a language variable set as shown in equation (1) [12].

$$\begin{cases} X = (X_1, X_2, ..., X_{m-1}, X_m) \\ V = (V_1, V_2, ..., V_{n-1}, V_n) \end{cases} \quad (1)$$

In equation (1), m and n are the quantity of language variables for distance and speed. The language variable set for outputting data is equation (2).

$$C = (C_1, C_2, ..., C_{t-1}, C_t) \quad (2)$$

In equation (2), C represents the language variable dataset, and t represents the number of language variables in the output data. Based on the membership relationship between the collected state variables and the output data [13], the constructed fuzzy rule table is listed in Table 2.

Table 2: Fuzzy rules of variables

	X_1	X_2	...	X_k	...	X_{m-1}	X_m
V_1	C_{11}	C_{21}	...	C_{k1}	...	$C_{(m-1)1}$	C_{m1}
V_2	C_{12}	C_{22}	...	C_{k2}	...	$C_{(m-1)2}$	C_{m2}
...	...	...		...		...	...
V_h	C_{1h}	C_{2h}	...	C_{kh}	...	$C_{(m-1)h}$	C_{mh}
...	...	...		...		...	...
V_{n-1}	$C_{1(n-1)}$	$C_{2(n-1)}$	...	$C_{k(n-1)}$	...	$C_{(m-1)(n-1)}$	$C_{m(n-1)}$
V_n	C_{1n}	C_{2n}	...	C_{kn}	...	$C_{(m-1)n}$	C_{mn}

In Table 2, C_{kh} represents the acceleration language variable obtained by jointly outputting the k-th distance language variable and the h-th velocity language variable. The mapping relationship between the collected state variables and the output data is equation (3).

$$c_{kh} = f\left(x_k, v_h\right) \tag{3}$$

In equation (3), c_{kh}, x_k, and v_h represent the maximum membership degrees of language variables C_{kh}, X_k, and V_h. A three-layer neural network model is established based on the rules obtained from the fuzzy rule table. The number of neurons is defined as the number of collected states, the neurons in the output layer are the number of output data, and the neuron in the hidden layer is i. The hyperbolic tangent function is used to transfer data between levels, and the Back Propagation (BP) algorithm is utilized to train the weights and bias vectors [14]. The model calculates the real-time acceleration based on the real-time collected velocity and distance state variables, as shown in equation (4).

$$a_t = \tan sig\left(q_2 \cdot \tan sig\left(q_1 \cdot \left(x_t, v_t\right)^T + b_1\right) + b_2\right) \tag{4}$$

In equation (4), T is the time variable. b_1 and b_2 represent the bias vectors of the hidden and output layers after training. q_2 and q_1 are the weight vectors of the output and hidden layers after training. a_t represents the normalized real-time acceleration. After detecting surrounding vehicles, the intelligent car calculates its real-time acceleration based on the model, allowing the car following the preceding car while maintaining a secure distance and speed. The collected state variables include the relative distance Δx between the intelligent car and the front car, as well as the relative velocity Δv with the front vehicle.

The time interval $\tau = \dfrac{\Delta x}{\Delta v}$ is increased as the input state variable reduces the dimensionality of the model. Based on the dataset of manual driving experience, the membership degree set of relative speed language variables has been determined as shown in equation (5) [15].

$$\Delta v = \left\{-5, -3, -1, 0, 1, 3, 5\right\} \tag{5}$$

Table 3 shows the model constructed based on fuzzy rules that automatically follows the sample set.

Table 3: Sample set of smart car tracking driving

τ	Δv						
	-5	-3	-1	0	1	3	5
0.25	-1	-1	-2	-2	-3	-3	-5
0.50	-0.5	-1	-1.5	-2	-4	-4	-5
0.75	0	-0.5	-1	-2	-3	-4	-4
1.00	0.5	0	-1	-2	-3	-3	-4
1.25	1	0.5	0	-0.5	-1	-2	-3
1.50	2	1	0.5	0	-0.5	-1	-2
1.75	3	2	1	0.5	0	-0.5	-1
2.00	4	3	2	1	0.5	0	-0.5
3.00	5	4	3	2	1	0.5	0
4.00	4	3	2	1	0.5	0	-0.5
5.00	3	2	1	0.5	0	-0.5	-1
10.0	2	1	0.5	0	-0.5	-1	-2
15.0	1	0.5	0	-0.5	-1	-2	-3

FNN is trained using the dataset in Table 3, and the intelligent vehicle FNN model is offline modified through training and self-learning. The modified model is used to calculate the real-time Acceleration of the Vehicle (AoV). When driven by driving needs and allowed by the surrounding environment, lane changing operations can be carried out in a timely manner [16]. However, the driving environment of vehicles is complex and fuzzy, making it difficult to perform lane changing operations through accurate and standardized information, and the correlation between road environment and lane changing intentions is non-linear. The process of learning the human brain through FNN and simulating the ability to change lanes for driving is shown in Figure 1.

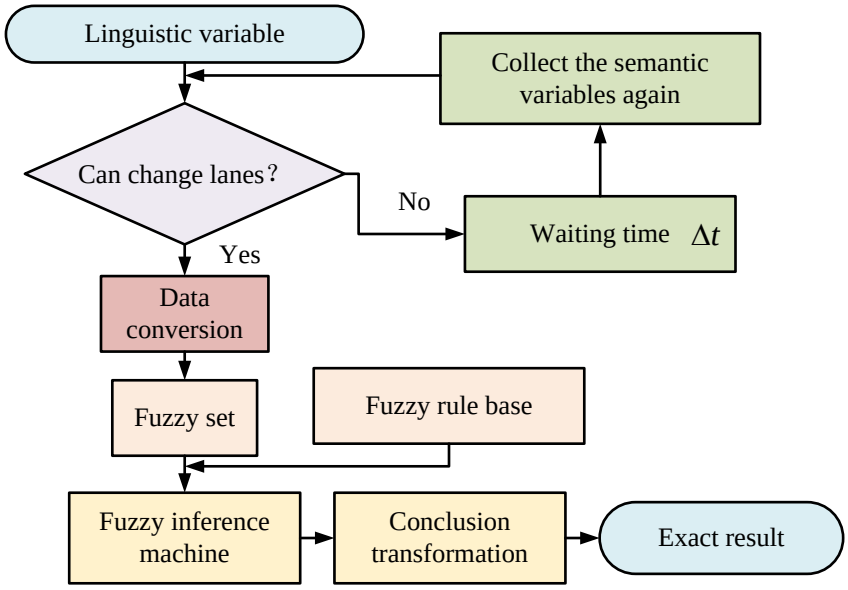

Figure 1: Flow chart of FNN lane change

In Figure 1, the language variables of relative speed and distance are inputted into the model, and the data are fuzzified to determine whether the surrounding environment meets the lane changing requirements. If it is not met, the language variables of the relevant data will be collected again after the waiting time Δt. If the lane change requirement is met, the fuzzy set is calculated to obtain a specific output fuzzy set. Based on the fuzzy rule library, the fuzzy set is deblurred to convert the data into accurate conclusions. When calculating the probability of lane change, the range of distance distribution between vehicles and surrounding vehicles is defined as [0, 1], and the membership range is equation (6) below.

$$\begin{cases} \Delta x_1 = [0, 0.25, 0.5, 1] \\ \Delta x_2 = [0.25, 0.5, 0.75, 1] \end{cases} \tag{6}$$

In equation (6), Δx_1 and Δx_2 are the membership range of the distance between the intelligent car and the front-car/vehicleon the left and right lanes. The decision to perform a lane change operation is determined based on the intensity of the lane change requirement and the feasibility of the lane change process.

2.2 Intelligent driving tracking algorithm for intelligent vehicles based on multi-sensor data collection

The visual sensor target detection method of smart cars has poor detection performance in harsh environments such as dark night, sandstorms, and haze, while the detection method using radar sensors is less affected by the environment. However, the sensor target detection method has weak ability to analyze target categories and cannot take corresponding measures in a timely manner. Therefore, visual sensors and radar sensors are combined for target detection. The Multi-sensor Driving Tracking Algorithm (MS-DTA) converts the target's motion state to a Cartesian coordinate system through radar sensors, as shown in equation (7).

$$\begin{cases} x = r \cdot \sin(\theta) \\ y = r \cdot \cos(\theta) \end{cases} \tag{7}$$

In equation (7), x, y, and r represent the lateral distance, Longitudinal Distance (LGD), and straight-line distance between two vehicles. Due to the need for radar sensors to measure the LGD difference and Longitudinal Velocity (LGV) difference between the experimental vehicle and surrounding vehicles, the observation amount at time t of the target is calculated as shown in equation (8).

$$Q\begin{cases} Q_1 = s = \sqrt{x^2(t) + y^2(t)} + r_1(t) \\ Q_2 = v = \sqrt{x^2(t) + y^2(t)} + r_2(t) \\ Q_3 = a_x = \dfrac{\left(x(t) - x(t-1)\right)}{T} + r_3(t) \\ Q_4 = a_y = \dfrac{\left(y(t) - y(t-1)\right)}{T} + r_4(t) \end{cases} \tag{8}$$

In equation (8), v is the relative LGV. T represents the time interval for data collection. r_1 - r_4 represents the observation interference at each moment. a_x and a_y represent the lateral and longitudinal AoV. A two-degree of freedom model is used to construct the vehicle's state sensor. The model ignores the influence of steering system and air resistance on the vehicle, and adopts the vehicle dynamics equations of torque balance and lateral force balance as shown in equation (9) [17].

$$\begin{cases} I_z \cdot \ddot{\vartheta} = F_1 \cdot x_1 - F_2 \cdot x_2 \\ M \cdot a_x = F_1 + F_2 \end{cases} \quad (9)$$

In equation (9), I_z represents the lateral torque of the car. $\ddot{\vartheta}$ is the vehicle's yaw angle. F_1 and F_2 are the lateral forces acting on the front and rear wheels. x_1 and x_2 are the lengths from the mass center of the vehicle to the front and rear axles. a_x represents the lateral AoV. M is the mass of the entire vehicle. The intelligent car driving tracking algorithm first preprocesses the data collected by sensors to determine the rationality of the data [18]. The algorithm analyzes sensor strategy data, distinguishes effective targets and interfering targets, and compares the consistency of reasonable target measurements within two cycles. If both the predicted and measured values of the sensor at a certain sampling time are less than or equal to the set threshold, then the measured value is valid and can update the motion state of the target car. If the predicted and measured values are greater than the threshold, the process of judging the consistency of the target car's motion state is shown in Figure 2.

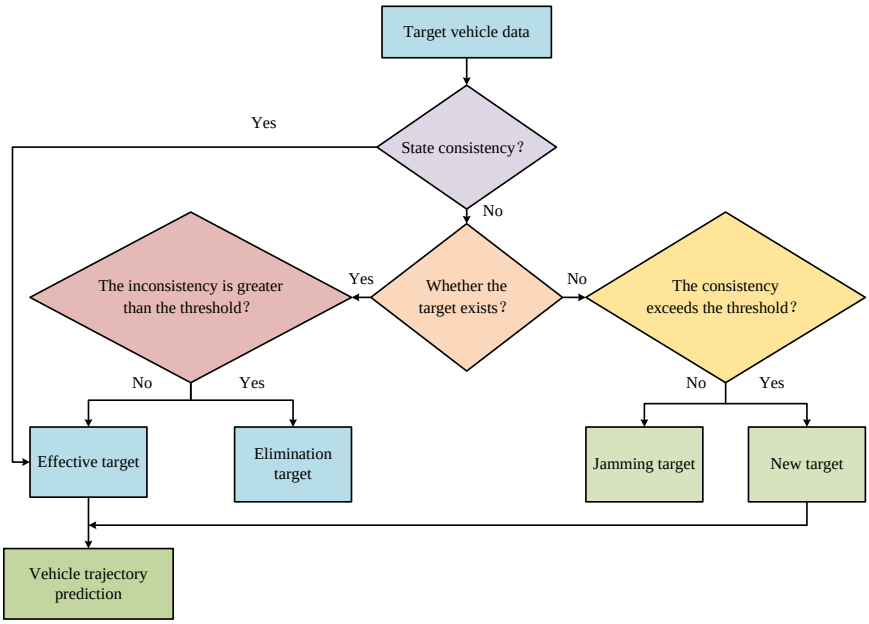

Figure 2: Process of judging vehicle motion consistency

In Figure 2, if the motion condition of the target vehicle in a certain sampling is inconsistent, it is determined whether the target exists. If it exists, to determine whether the number of times the measured value at a certain sampling time is inconsistent with the target state at multiple sampling times is greater than the set inconsistency threshold. If it is greater than the threshold, the target is eliminated; If it is less than the threshold, the target is considered a valid target. If the target does not exist, it is determined whether the number of times the measured value at a certain sampling time is consistent with the target state at multiple sampling times is greater than the set consistency threshold. If it is less than the threshold, the target is considered an interference target, and if it is greater than the threshold, a new target is generated. After determining as an effective target, continuing with target information prediction. The location of the target car in ADTA is divided into three categories: in front of the original lane, right lane, or left lane. The lane relationship between the intelligent vehicle and the target vehicle is shown in Figure 3.

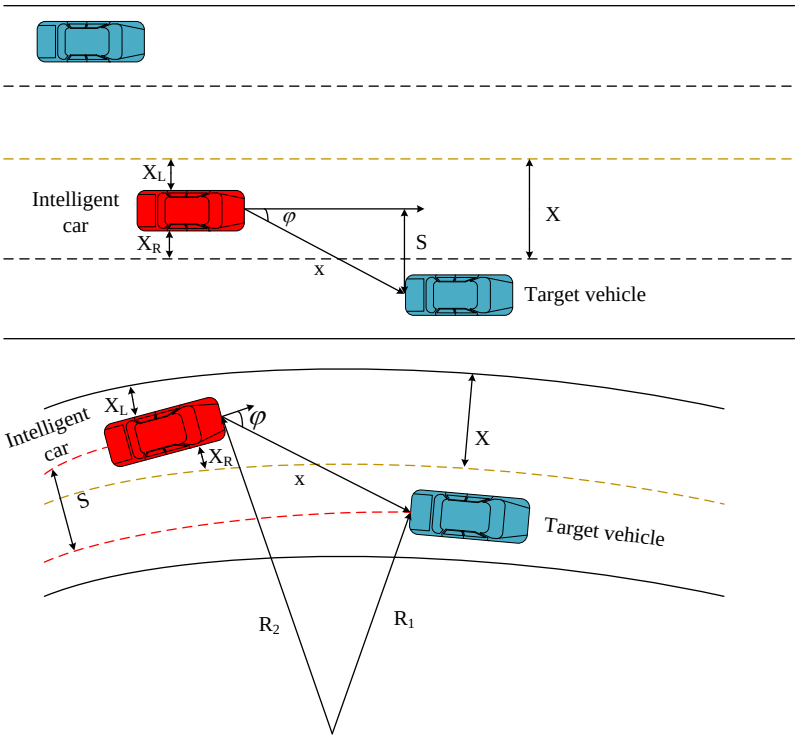

Figure 3: Lane distance recognition of target vehicle by ADTA

In Figure 3, the sensor is located in front of the vehicle. Based on the data returned by the sensor and the Cartesian coordinate system, the vertical distance between the target vehicle and the intelligent car's longitudinal axis is calculated to distinguish which lane the target vehicle is in. In a straight lane, the lateral distance between two vehicles is calculated as shown in equation (10) [19].

$$S = x \cdot \sin(\varphi) \tag{10}$$

In equation (10), x represents the straight-line distance between two cars, and φ represents the angle between the vertical axis of the intelligent car and the connecting line between the two cars. When the target vehicle is on the right side of the smart car, $\varphi > 0$, and when it is on the left side, $\varphi < 0$. When the vehicle is driving in a bend, the calculation of the distance S between the central axes of the two vehicles is equation (11).

$$\begin{cases} S = R_1 - R_2 \\ \dfrac{1}{R_2} = \dfrac{\omega}{v_2} \\ R_1^2 = R_2^2 + x^2 - 2R_0 \cdot x \cdot \cos(90^\circ - \varphi) \end{cases} \tag{11}$$

In equation (11), R_1 and R_2 are the driving radii of the target and intelligent vehicles. ω and v_2 are the lateral velocity and LGV of smart cars. The resolution method for the target vehicle in the right lane is equation (12).

$$|X_L| + \frac{1}{2}x_1 + \frac{1}{2}x_2 + x_3 \rangle S \rangle |X_L| + \frac{1}{2}x_1 + \frac{1}{2}x_2 \tag{12}$$

In equation (12), X_L represents the distance between the smart car and the left lane. x_1, x_2, and x_3 represent the width of smart cars, lane markings, and single lanes. The resolution method for the target vehicle in the left lane is equation (13).

$$|X_R| + \frac{1}{2}x_1 + \frac{1}{2}x_2 + x_3 \rangle S \rangle |X_R| + \frac{1}{2}x_1 + \frac{1}{2}x_2 \tag{13}$$

In equation (13), X_R represents the distance between the smart car and the right lane line. Accurate operation of ADTA requires the use of a distributed information fusion structure, which integrates the predicted trajectory values of target vehicles calculated by visual sensors and radar sensors [20]. The structure uses a fast and simple convex fusion algorithm for target trajectory fusion, and the calculation of the convex fusion algorithm is equation (14).

$$s = \left[(P_1)^{-1} + (P_2)^{-1}\right]^{-1} \cdot (P_1)^{-1} \cdot s_1 + \left[(P_1)^{-1} + (P_2)^{-1}\right]^{-1} \cdot (P_2)^{-1} \cdot s_2 \tag{14}$$

In equation (14), s_1 and s_2 are trajectory data obtained from visual sensors and radar sensors. P_1 and P_2 are covariance matrices, as shown in equation (15).

$$P^{-1} = (P_1)^{-1} + (P_2)^{-1} \tag{15}$$

The MS-DTA runtime framework is shown in Figure 4.

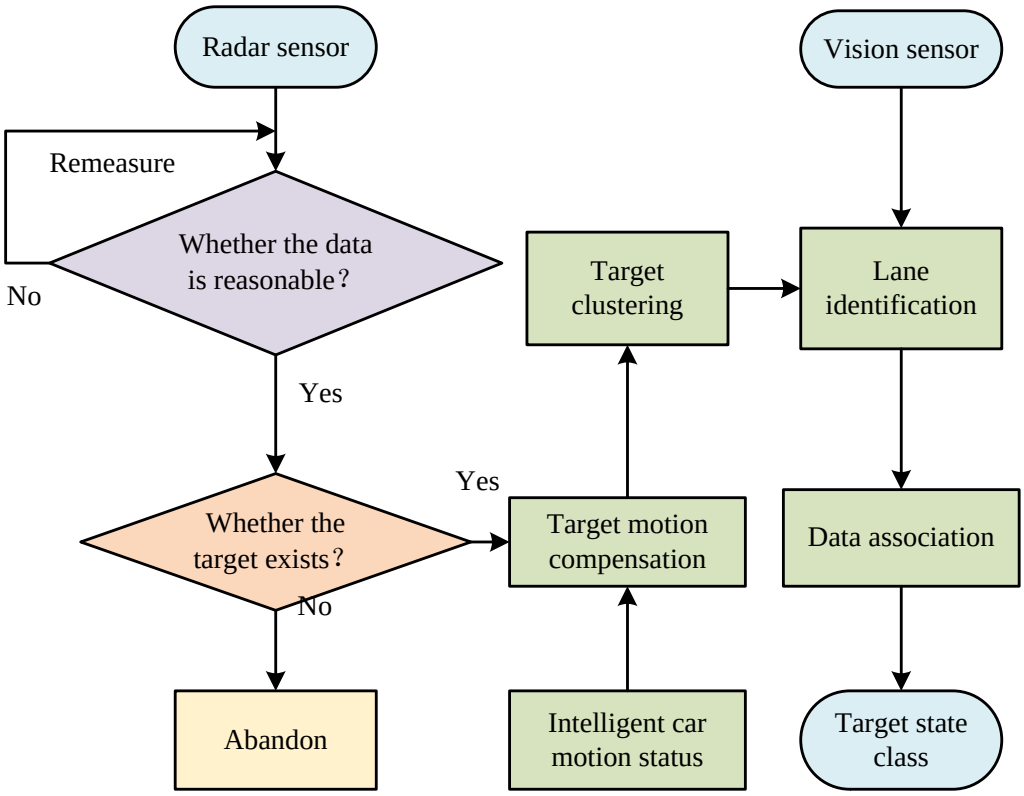

Figure 4 Multi-sensor driving tracking algorithm

In Figure 4, the numerical values returned by the radar sensor are first preprocessed to determine whether the target data is within the preset accuracy range. If it is not, the set of data is remeasured and validity is checked within the range. Then, based on the motion status of the intelligent vehicle, the effective data is used to perform certain motion compensation on the target car, and the lane where the target one is located after motion compensation is recognized. Finally, the tracking speed of the smart car is calculated built on the target lane and real-time motion status. Apollo Scape dataset released by Baidu is used for trainingin this research (https://apolloscape.auto/). The dataset includes scene analysis, 3D object instance, lane segmentation, self-positioning, trajectory estimation, detection and tracking, stereo vision, and other scene related data. The Track Estimation section contains more than 140K high-resolution GPS tracks and camera images. These images have a resolution of 3384×2710 and cover a variety of road conditions.

3 Results

3.1 Simulation Experiment of CAD tracking technology based on fuzzy information

The simulation experiment scenario is set as the target vehicle traveling at a speed of 18m/s, with an initial distance of 30 m between the experimental and target vehicles. The experimental vehicle follows the target vehicle at the same speed. The front vehicle gradually slows down with an acceleration of -1 m/s, and after 13 seconds, the target vehicle's speed drops to 5m/s and continues to drive at that speed. The initial data of the vehicle in another scenario are the same as the previous scenario, and the speed of the experimental vehicle continuously changes with the target one. The data for the interference experiment are set to add a random velocity measurement error of about 1m/s, and the distance measurement error of the sensor fluctuates randomly within 2m. The results obtained from the simulation experiment are shown in Figure 5.

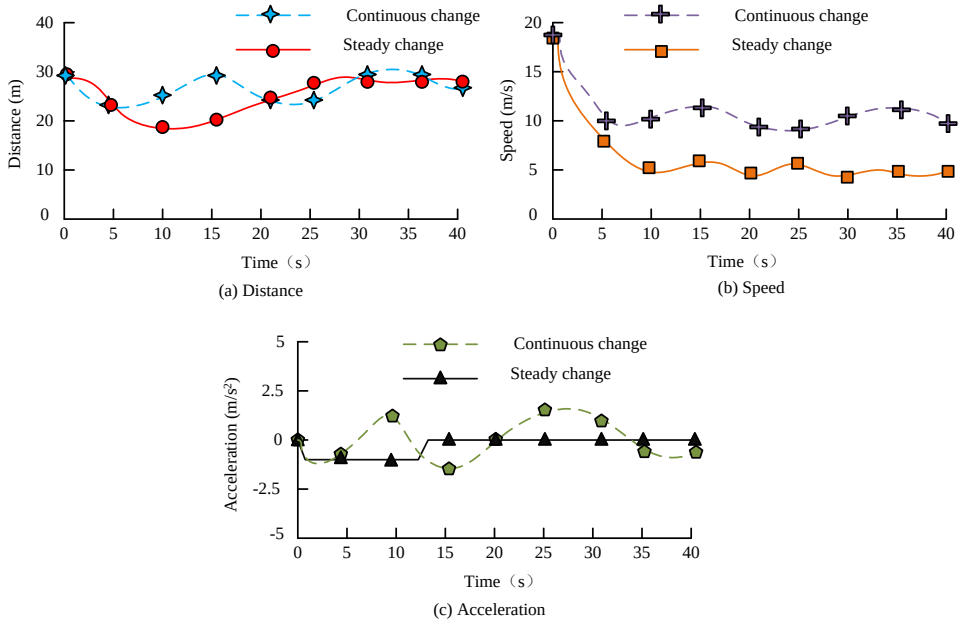

Figure 5: Test of following speed change under different environments

In Figure 5 (a), within 0-10 seconds, the distance between the experimental car and the target car gradually approaches as the preceding vehicle decelerates, but the experimental vehicle quickly adjusts and returns to a safe distance within 10-20 seconds. When the current vehicle speed continuously changes, the experimental vehicle can quickly complete adjustments in a short period of time and maintain a safe distance at all times. In 5 (b), the experimental vehicle can accurately recognize the velocity changes of the preceding car and adjust its own speed accordingly. In 5 (c), the acceleration of the experimental vehicle remainswithin a safe range. The acceleration and distance variation curves of FNN model, Fuzzy Programming Model (FPM), and Traditional Programming (TPM) are displayed in Figure 6.

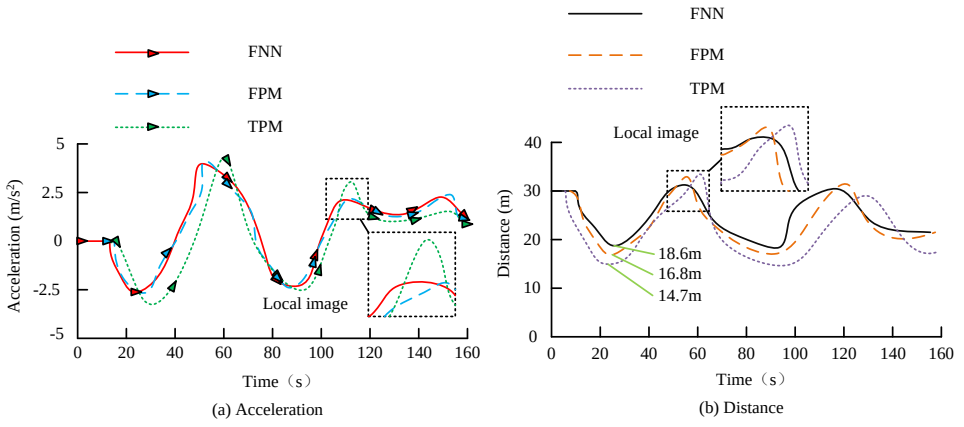

Figure 6: Acceleration and distance curves of different models

In Figure 6 (a), as the velocity of the preceding car changes, the acceleration change time of FNN is 0.42s and 1.03s faster than FPM and TPM, respectively, indicating that FNN has a faster reaction speed. Moreover, the acceleration continuous variation curve of the FNN model is smoother than other models, and the highest acceleration is also lower than the highest acceleration of other models, indicating its superior performance and safety. In 6 (b), FNN has a shorter deceleration distance and maintains a relatively safe distance of 1.8m and 3.9m longer compared to FPM and TPM, respectively. The comparison of FNN's anti-interference ability with other models is shown in Figure 7.

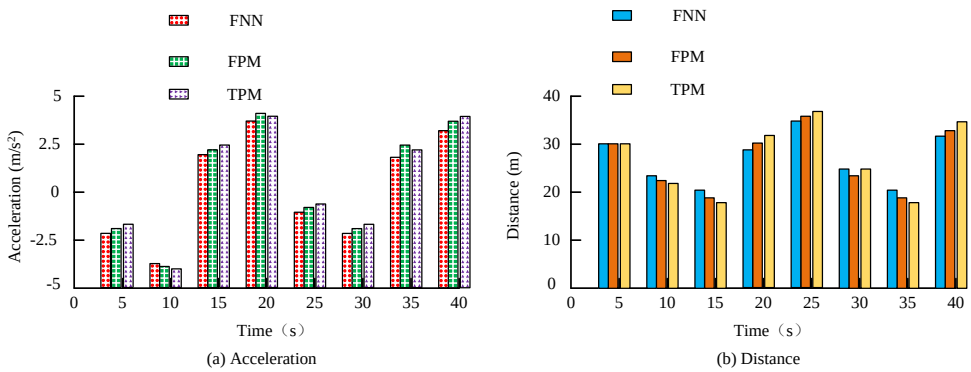

Figure 7: Acceleration and distance changes of different models under interference

In Figure 7 (a), the acceleration change of FNN is significantly better than that of FPM and TPM, and the average speed of acceleration change with the preceding vehicle is 0.58 s and 1.67 s faster than the other two algorithms, respectively. The acceleration variation of FNN only fluctuates slightly and can maintain a stable following speed. In 7 (b), the average following distance of FNN is 2.5 m and 4.1m longer than other algorithms when the preceding vehicle accelerates, and 2.1 m and 3.8 m shorter when the preceding vehicle decelerates. The following distance of the model is always within a safe range and will not leave the preceding vehicle too far.

3.2 Simulation experiment analysis based on MS-DTA

The simulation experiment uses an 80 Hz radar sensor and a 72 line LiDAR, with a visual sensor resolution of 1080p. The vehicle's status is measured in real-time using a navigator. The experiments are compared with the test data of SH-EKF, SR-CKF, and RM algorithms under noise interference. The Root Mean Square Error (RMSE) of the algorithm can be tested in three environments: groups 1 and 2 on highways, groups 3 and 4 on urban loops, and groups 5 and 6 on old neighborhoods. The MS-DTA algorithm preprocesses and fuses the data returned by the radar sensor to determine whether the target data is within the preset accuracy range. If not, the dataset is remeasured. The valid data are used to perform certain motion compensation on the target vehicle according to the motion state of the intelligent vehicle, and the lane in which the target vehicle is located after motion compensation is recognized. The real-time tracking of the intelligent vehicle's traveling speed is calculated based on the lane in which the target is located and the real-time motion state. The comparison between the estimated vehicle yaw velocity and lateral velocity from MS-DTA and the measured values from the navigation instrument is shown in Figure 8.

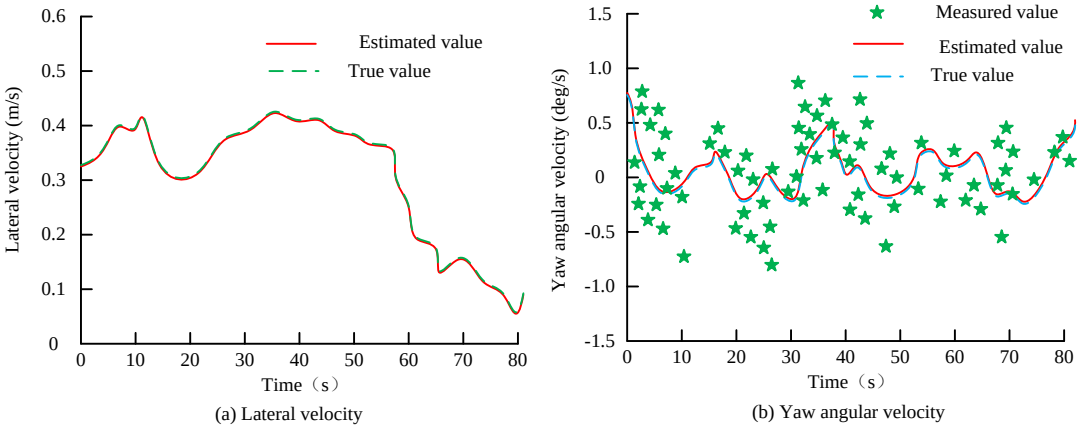

Figure 8: Yaw velocity and lateral velocity estimation of MS-DTA

In 8 (a), the estimated lateral velocity by MS-DTA is basically consistent with the actual lateral velocity, except that at the turning position, the fluctuation of the estimated value is slightly larger than the actual value. In 8 (b), the estimated yaw velocity of the algorithm has good consistency with the actual yaw velocity. The measured values of yaw rate are well centered around the true value, and the dense part of the measured values is concentrated at the turning position of the vehicle. The LGD and LGV of MS-DTA vehicles are compared with

other algorithms, as shown in the following Figure (9).

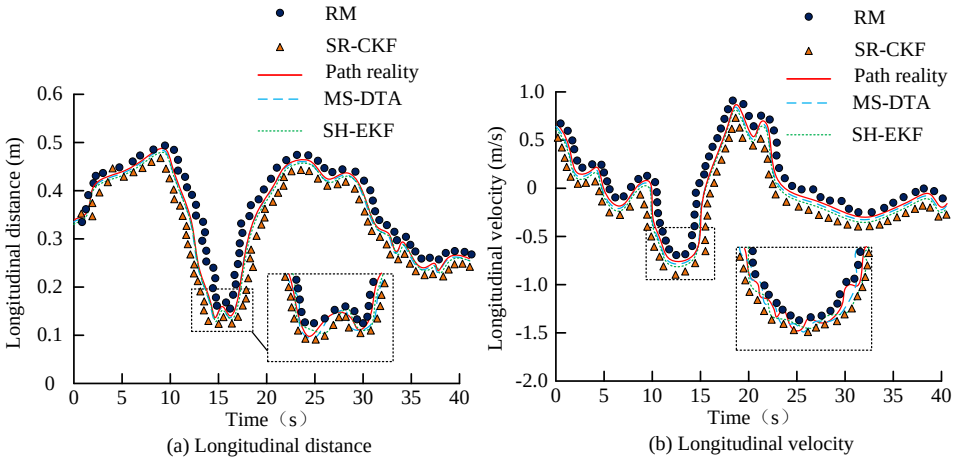

Figure 9: LGD and LGV curves of various algorithms with time

In Figure 9 (a), the data estimated by MS-DTA are closer to the true value, and the average LGD estimated by the RM algorithm is 0.37m higher than the true value on average under noise interference. The average LGD estimated by SR-CKF and SH-EKF is 0.34 m and 0.28 m lower than the true value, respectively, and the fluctuation at the turning is more severe, with a greater difference from the true value. In 9 (b), the LGV variation estimated by MS-DTA is smoother and closer to the true value. When the preceding vehicle decelerates, the MS-DTA estimated value is higher, the algorithm responds faster, and the deceleration is smoother. This indicates that using MS-DTA for autonomous vehicles is more comfortable and safer to ride.

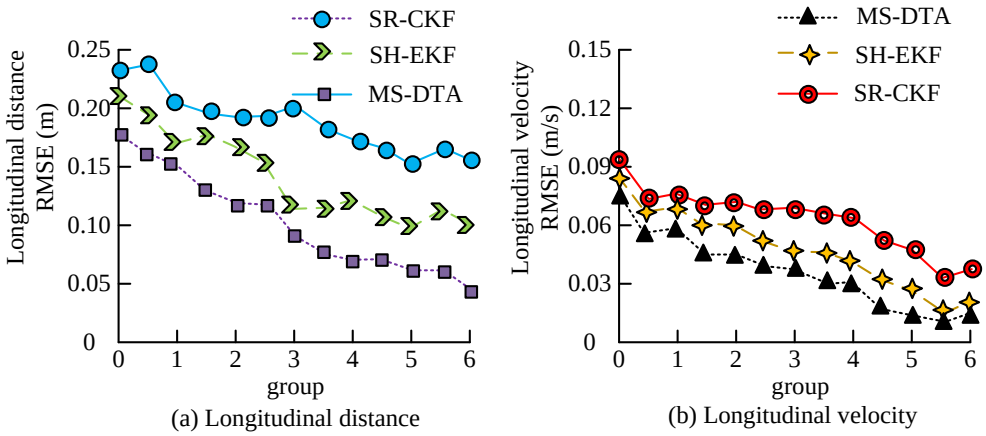

Figure 10: RMSE of LGD and LGV of different algorithms

In Figure 10 (a), during the 5 and 6 experiments, the system noise is diverse and constantly changing. The LGD RMSE of MS-DTA is 0.04m and 0.08m lower than that of SH-EKF and SR-CKF, respectively. However, in the highway environment, noise interference is stronger, so its LGD RMSE is 0.03m and 0.06m lower than other algorithms, respectively. The vertical distance denoising accuracy of the algorithm is 34.29% and 47.34% higher than that of SH-EKF and SP-CKF, respectively. In 10 (b), the average LGV RMSE of MS-DTA is 0.01m/s and 0.02m/s lower than other algorithms, and the LGV denoising accuracy is 18.65% and 31.27% higher than SH-EKF and SP-CKF, respectively. This indicates that MS-DTA can effectively remove external interference, continuously monitor the trajectory of the preceding vehicle, and reduce algorithm tracking errors.

To verify the trajectory prediction ability of the MS-DTA algorithm in real driving environments, real-vehicle tests are conducted through a variety of traffic scenarios, including intersections, multi-lane straight driving, and curved driving. The experiments show that the MS-DTA algorithm is able to accurately predict the trajectory of vehicles located in the turning lane at intersections, and the SH-EKF algorithm has the case of incorrectly predicting the turning vehicle as traveling straight. At the same time, the MS-DTA algorithm is able to effectively predict the future trajectory of the car based on the historical trajectory of

the previous car, and the trajectory prediction error of the vehicle is significantly smaller than other algorithms. To further verify the performance of the proposed algorithm, it is compared with baseline algorithms for automatic driving tracking, including Rapidly-Exploring Random Tree (RRT) and the Probabilistic Roadmap Method (PPM) based on the heuristic node enhancement strategy. The comparison results are shown in Table 4.

Table 4: Performance comparison between the MS-DTA algorithm and the baseline algorithm

Algorithm	RMSE of the longitudinal distance	RMSE of the longitudinal velocity	Yaw velocity RMSE	Lateral velocity RMSE
MS-DTA	0.04	0.02	0.04	0.05
RRT	0.07	0.04	0.06	0.09
PPM	0.09	0.08	0.12	0.11

In Table 4, the RMSE of each value of the MS-DTA algorithm is the minimum. The RMSE of the longitudinal distance is 0.03 and 0.05 lower than that of the RRT and PPM algorithms, respectively. The RMSE of the longitudinal velocityis 0.02 and 0.06 lower than that of the RRT and PPM algorithms. The RMSE of yaw velocity and lateral velocity of MS-DTA algorithm is much different from that of the two benchmark algorithms, indicating that the proposed MS-DTA algorithm is more accurate in predicting vehicle trajectory during turning.

4 Discussion

In recent years, with the continuous development of information technology, assisted driving or even intelligent vehicle automatic driving technology has become a popular research direction in intelligent transportation. However, traditional intelligent driving tracking algorithms suffer from low trajectory prediction accuracy. This study proposes an MS-DTA algorithm based on fuzzy information and multi-sensor data fusion, and verifies its good performance in automatic driving trajectory prediction and noise interference removal through experimental analysis. The lateral velocity predicted by the MS-DTA algorithm is basically consistent with the true lateral velocity of the vehicle, except that the fluctuation of the predicted value is slightly greater than the true value at the turning position. Since the algorithm lags behind when turning and enters the curve later, it is necessary to increase the lateral speed of the vehicle to ensure safe driving in the curve. The intensive part of the transverse angular velocity measurements are concentrated at the vehicle turning position to adjust the steering angle and speed of the vehicle in time. Compared with the deep multi-task learning model proposed by Natan O et al. [5], the MS-DTA algorithm's transverse swing angular velocity and lateral velocity are smaller and closer to the real values. Itindicates that the MS-DTA algorithm using multi-sensor fusion is more responsive and safer. After adding noise interference, compared with the hybrid method of fuzzy algorithm and NMS algorithm proposed by Zaghari N et al. [9], the longitudinal distance RMSE of MS-DTA algorithm gradually decreases with the diversity and complexity of noise, and the decrease speed is faster. This indicates that the MS-DTA algorithm for multi-sensor data fusion has stronger anti-interference ability, and the accuracy of longitudinal distance denoising has improved by 34.29% compared to Zaghari N et al.'s method. Compared with the existing automobile autopilot trajectory prediction techniques, MS-DTA algorithm can effectively remove the external interference, continuously focus on the trajectory of the front vehicle, and reduce the algorithm tracking error.

5 Conclusion

In response to the problems of complex path planning and insufficient trajectory prediction accuracy in traditional ADTA, this study proposed MS-DTA using fuzzy information and multi-sensor fusion. The research results indicated that the FNN constructed with fuzzy information identified the target vehicle's speed change time, which was 0.42 s and 1.03 s faster than other models. The relative distance between the predicted path of the model and the target vehicle was 1.8 m and 3.9 m longer than other models. The safe distance between the two vehicles was longer, and they could also closely follow the target vehicle within the safe distance when accelerating. The acceleration curve of the model changed more smoothly, with the highest acceleration being 0.13 m/s2 and 0.45 m/s2 lower. Under noise interference, the velocity changes of the model were 0.58 s and 1.67 s faster than other models, and the following distance was 2.5 meters and 4.1 meters longer than other models. The vehicle lateral velocity and yaw velocity estimated using MS-DTA were highly consistent with the actual measured values. The LGD estimated by MS-DTA was basically consistent with the true value, and the average LGD of SR-CKF and SH-EKF were 0.34 m and 0.28 m lower than the true value. The MS-DTA estimated longitudinal acceleration change of the vehicle was smoother, which could enhance the comfort of autonomous vehicles. In the noise interference experiment, the LGD RMSE of MS-DTA was 0.04m and 0.08m lower than other models, and the LGV RMSE was 0.01m/s and 0.02m/s lower than them. There are still

some shortcomings in this study, such as the fact that traffic lights are also an important factor affecting vehicle driving. In the future, research on algorithm detection of traffic lights will be added to improve the full scenario application ability of autonomous driving.

Funding

The research is supported by Sichuan Polytechnic University research project XJ2024KJ-24 (Research on Optimal design of target detection of millimeter wave radar for automotive intelligent driving).

References

[1] Z. Wang, and J. Fei, "Fractional-order terminal sliding-mode control using self-evolving recurrent chebyshev fuzzy neural network for MEMS gyroscope," IEEE Transactions on Fuzzy Systems, vol. 30, no. 7, pp. 2747-2758, 2021. https://doi.org/10.1109/TFUZZ.2021.3094717

[2] C. Ben Jabeur, and H. Seddik, "Design of a PID optimized neural networks and PD fuzzy logic controllers for a two-wheeled mobile robot," Asian Journal of Control, vol. 23, no. 1, pp. 23-41, 2021. https://doi.org/10.1002/asjc.2356

[3] M. Elsisi, "Improved grey wolf optimizer based on opposition and quasi learning approaches for optimization: case study autonomous vehicle including vision system," Artificial Intelligence Review, vol. 55, no. 7, pp. 5597-5620, 2022. https://doi.org/10.1007/s10462-022-10137-0

[4] J. Fei, L. Liu, "Real-time nonlinear model predictive control of active power filter using self-feedback recurrent fuzzy neural network estimator," IEEE Transactions on Industrial Electronics, vol. 69, no. 8, pp. 8366-8376, 2021. https://doi.org/10.1109/TIE.2021.3106007

[5] O. Natan, and J. Miura, "Towards compact autonomous driving perception with balanced learning and multi-sensor fusion," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 9, pp. 16249-16266, 2022. https://doi.org/10.1109/TITS.2022.3149370

[6] N. Senel, K. Kefferpütz, K. Doycheva, and G. Elger, "Multi-sensor data fusion for real-time multi-object tracking," Processes, vol. 11, no. 2, pp. 501-527, 2023. https://doi.org/10.3390/pr11020501

[7] Z. Zhang, J. Zhao, C. Huang, and L. Li, "Learning visual semantic map-matching for loosely multi-sensor fusion localization of autonomous vehicles," IEEE Transactions on Intelligent Vehicles, vol. 8, no. 1, pp. 358-367, 2022. https://doi.org/10.1109/TIV.2022.3173662

[8] K. Wang, C. Cao, S. Ma, and F. Ren, "An optimization-based multi-sensor fusion approach towards global drift-free motion estimation," IEEE Sensors Journal, vol. 21, no. 10, pp. 12228-12235, 2021. https://doi.org/10.1109/JSEN.2021.3064446

[9] N. Zaghari, M. Fathy, S. M. Jameii, and M. Shahverdy, "The improvement in obstacle detection in autonomous vehicles using YOLO non-maximum suppression fuzzy algorithm," The Journal of Supercomputing, vol. 77, no. 11, pp. 13421-13446, 2021. https://doi.org/10.1007/s11227-021-03813-5

[10] N. Awad, A. Lasheen, M. Elnaggar, and A. Kamel, "Model predictive control with fuzzy logic switching for path tracking of autonomous vehicles," ISA Transactions, vol. 129, no. 6, pp. 193-205, 2022. https://doi.org/10.1016/j.isatra.2021.12.022

[11] Z. Sun, J. Zou, D. He, and W. Zhu, "Path-tracking control for autonomous vehicles using double-hidden-layer output feedback neural network fast nonsingular terminal sliding mode," Neural Computing and Applications, vol. 34, no. 7, pp. 5135-5150, 2022. https://doi.org/10.1007/s00521-021-06101-8

[12] S. Choudhuri, S. Adeniye, and A. Sen, "Distribution alignment using complement entropy objective and adaptive consensus-based label refinement for partial domain adaptation," Artificial Intelligence and Applications, vol. 1, no. 1, pp. 43-51, 2023. https://doi.org/10.47852/bonviewAIA2202524

[13] T. Liu, S. Du, C. Liang, B. Zhang, and R. Feng, "A novel multi-sensor fusion-based object detection and recognition algorithm for intelligent assisted driving," IEEE Access, vol. 9, no. 12, pp. 81564-81574, 2021. https://doi.org/10.1109/ACCESS.2021.3083503

[14] V. Shepelev, S. Zhankaziev, S. Aliukov, V. Varkentin, A. Marusin, A. Marusin, and A. Gritsenko, "Forecasting the passage time of the queue of highly automated vehicles based on neural networks in the services of cooperative intelligent transport systems," Mathematics, vol. 10, no. 2, pp. 282-294, 2022. https://doi.org/10.3390/math10020282

[15] S. K. Swain, J. J. Rath, and K. C. Veluvolu, "Neural network based robust lateral control for an autonomous vehicle," Electronics, vol. 10, no. 4, pp. 510-523, 2021. https://doi.org/10.3390/electronics10040510

[16] P. J. Navarro, L. Miller, F. Rosique, C. Fernández-Isla, and A. Gila-Navarro, "End-to-end deep neural network architectures for speed and steering wheel angle prediction in autonomous driving," Electronics, vol. 10, no. 11, pp. 1266-1273. https://doi.org/10.3390/electronics10111266

[17] N. Tork, A. Amirkhani, and S. B. Shokouhi, "An adaptive modified neural lateral-longitudinal control system for path following of autonomous vehicles," Engineering Science and Technology, an International Journal, vol. 24, no. 1, pp. 126-137, 2021. https://doi.org/10.1016/j.jestch.2020.12.004

[18] W. Wang, T. Qie, C. Yang, W. Liu, C. Lang, and K. Huang, "An intelligent lane-changing behavior

prediction and decision-making strategy for an autonomous vehicle," IEEE Transactions on Industrial Electronics, vol. 69, no. 3, pp. 2927-2937, 2021. https://doi.org/10.1109/TIE.2021.3066943

[19] C. J. Lin, and J. Y. Jhang, "Intelligent traffic-monitoring system based on YOLO and convolutional fuzzy neural networks," IEEE Access, vol. 10, no. 12, pp. 14120-14133, 2022. https://doi.org/10.1109/ACCESS.2022.3147866

[20] R. Dastres, and M. Soori, "Artificial neural network systems," International Journal of Imaging and Robotics (IJIR), vol. 21, no. 2, pp. 13-25, 2021. https://hal.science/hal-03349542