Lecture Notes in Electrical Engineering 1177

Yi Qu
Mancang Gu
Yifeng Niu
Wenxing Fu *Editors*

Proceedings of 3rd 2023 International Conference on Autonomous Unmanned Systems (3rd ICAUS 2023)

Volume VII

Springer

Lecture Notes in Electrical Engineering 1177

Series Editors

The book series *Lecture Notes in Electrical Engineering* (LNEE) publishes the latest developments in Electrical Engineering—quickly, informally and in high quality. While original research reported in proceedings and monographs has traditionally formed the core of LNEE, we also encourage authors to submit books devoted to supporting student education and professional training in the various fields and applications areas of electrical engineering. The series cover classical and emerging topics concerning:

- Communication Engineering, Information Theory and Networks
- Electronics Engineering and Microelectronics
- Signal, Image and Speech Processing
- Wireless and Mobile Communication
- Circuits and Systems
- Energy Systems, Power Electronics and Electrical Machines
- Electro-optical Engineering
- Instrumentation Engineering
- Avionics Engineering
- Control Systems
- Internet-of-Things and Cybersecurity
- Biomedical Devices, MEMS and NEMS

For general information about this book series, comments or suggestions, please contact leontina.dicecco@springer.com.

To submit a proposal or request further information, please contact the Publishing Editor in your country:

China

Jasmine Dou, Editor (jasmine.dou@springer.com)

India, Japan, Rest of Asia

Swati Meherishi, Editorial Director (Swati.Meherishi@springer.com)

Southeast Asia, Australia, New Zealand

Ramesh Nath Premnath, Editor (ramesh.premnath@springernature.com)

USA, Canada

Michael Luby, Senior Editor (michael.luby@springer.com)

All other Countries

Leontina Di Cecco, Senior Editor (leontina.dicecco@springer.com)

**** This series is indexed by EI Compendex and Scopus databases. ****

Yi Qu · Mancang Gu · Yifeng Niu · Wenxing Fu
Editors

Proceedings of 3rd 2023 International Conference on Autonomous Unmanned Systems (3rd ICAUS 2023)

Volume VII

Springer

Editors
Yi Qu
Nanjing University of Science
and Technology
Nanjing, China

Mancang Gu
Beijing HIWING Scientific
and Technological Information Institute
Beijing, China

Yifeng Niu
College of Intelligence Science
and Technology
National University of Defense Technology
Changsha, Hunan, China

Wenxing Fu
Unmanned System Research Institute
Northwestern Polytechnical University
Xi'an, Shaanxi, China

ISSN 1876-1100 ISSN 1876-1119 (electronic)
Lecture Notes in Electrical Engineering
ISBN 978-981-97-1102-4 ISBN 978-981-97-1103-1 (eBook)
https://doi.org/10.1007/978-981-97-1103-1

This Springer imprint is published by the registered company Springer Nature Singapore Pte Ltd.
The registered company address is: 152 Beach Road, #21-01/04 Gateway East, Singapore 189721, Singapore

Paper in this product is recyclable.

Contents

A Personalized Ramp Merging Decision-Making Method for Autonomous Driving Based on Reverse Reinforcement Learning

Fangbing Qu, Jianyong Qi(✉), Yao Xiao, and Jianwei Gong

Beijing Institute of Technology, Haidian District, Beijing 100081, China
qijianyong2023@163.com

Abstract. In the ramp merging scenario, the merging vehicles need to make decisions during the interaction with high-speed vehicles on the main lane to achieve safe and reliable merging. The advanced driving assistance system can assist in decision-making during this process, providing reference for drivers and improving safety. The main feature of the current stage is "Human-machine Shared Control". In order to meet the personalized driving needs of drivers, while ensuring safety, the driving habits and characteristics of drivers are fully considered, so that the decision-making and control results of the intelligent driving control system meet the expectations of drivers. Inverse reinforcement learning has shown good performance in personalized human learning and can learn the driving strategies of human drivers. However, many current methods of inverse reinforcement learning do not fully consider the interaction between vehicles. Therefore, this paper proposes a personalized ramp merging decision-making method based on maximum entropy inverse reinforcement learning, taking into account the interaction between vehicles. Based on driving style classification of human ramp merging data, targeted reward function forms are learned for different types of drivers to generate corresponding merging decision methods.

Keywords: Autonomous driving · Personalized learning · Inverse reinforcement learning · Interaction between vehicles

1 Introduction

Autonomous driving technology relies on the collaboration of artificial intelligence, visual computing, radar, monitoring devices, and global positioning systems, which can effectively prevent traffic accidents, alleviate traffic congestion and other traffic pressures, and complete specific work tasks in specific traffic scenarios [1]. Its role in our daily production and life is increasingly prominent [2–4]. However, achieving complete autonomous driving still requires a long transitional period. In this context, Advanced Driver Assistance System (ADAS) has become a focus of research and innovation in the field of autonomous driving. The application of ADAS technology provides new solutions for solving social problems such as traffic congestion and frequent accidents.

Y. Qu et al. (Eds.): ICAUS 2023, LNEE 1177, pp. 1–14, 2024.
https://doi.org/10.1007/978-981-97-1103-1_1

Despite the rapid development of ADAS technology in recent years, there are still some issues that need to be addressed. For example, in 2016, there were two traffic accidents in the Uber autonomous vehicle in Pittsburgh. Google's driverless vehicle collided with a public bus in Silicon Valley. In Germany, 23% of drivers are unwilling to accept smart cars because their fixed driving mode makes them feel constrained and uneasy. Therefore, in terms of driving safety and user satisfaction, autonomous driving technology still needs further research, indicating that it is necessary to study autonomous driving strategies that consider driver driving style and human-like learning.

Ramp merging is a typical traffic scenario and a place where traffic accidents are prone to occur. The merging behavior of vehicles entering the ramp can affect the continuity of traffic flow and determine the capacity of the main lane and ramp intersection area. At the same time, this merging behavior carries certain risks and is one of the important fields of autonomous driving technology research.

2 Related Work

2.1 Personalized Driving Research

With the development of ADAS technology and the advancement of autonomous driving technology, providing safe personalized driving assistance and driving systems has become a research hotspot [5]. Chen et al. [6] investigated the personalized decision-making characteristics of different drivers by interfering with 50 drivers in driving tasks and collecting data on dangerous driving behaviors. Ramyar et al. [7] designed a personalized driving decision learning method for ADAS in highway environments, achieving personalized driving decisions while ensuring safety and not violating traffic regulations. Xiao et al. [8] extracted the conventional travel behavior of private cars based on trajectory data analysis, which is beneficial for improving people's travel experience. Based on the advantages of hidden Markov models in processing time series data, Deng et al. [9], in order to identify driving styles, used hidden Markov models to model three driving styles based on driver braking characteristics, in order to achieve efficient driving styles.

2.2 Inverse Reinforcement Learning

Inverse reinforcement learning algorithm decomposes the process of imitative learning into two sub processes: inverse reinforcement learning and reinforcement learning. The two sub processes iterate repeatedly to obtain the optimal strategy [10, 11]. The earliest inverse reinforcement learning algorithm solved the potential reward function by maximizing the margin between the optimal and suboptimal actions, and represented the reward function in a linear combination. The potential reward function was obtained by solving the corresponding weight coefficients [12, 13]. Ratliff et al. [14, 15] proposed an inverse reinforcement learning algorithm based on maximum marginal programming to solve the reward function. Neu et al. [16, 17] introduced gradient theory to improve the efficiency of expert sample utilization in parameter updates of reward functions. In the selection of characteristic parameters for reward functions, researchers usually use different parameters. Wu et al. [18] selected speed, comfort, and safety as the characteristic

parameters of the reward function. M. Naumann et al. [19] mainly considered parameters such as comfort, speed, traffic rules, safe distance, and collision time as characteristic parameters of the reward function.

In summary, most of the reward functions for inverse reinforcement learning mainly choose the characteristics of the main vehicle itself as feature parameters, such as expected speed, comfort, safety, etc., and rarely consider the interaction with surrounding vehicles, nor do they consider the impact of environmental vehicles on the behavior of the main vehicle. Therefore, this paper takes these factors into account to make it more in line with the real driving environment.

2.3 Main Contributions of This Paper

Based on the above issues, this paper adopts a personalized ramp merging decision model based on maximum entropy inverse reinforcement learning. In addition to safety and comfort, it also considers the mutual influence between the main vehicle and surrounding vehicles, providing a certain reference for humanoid learning and personalized driving. The framework of this paper is shown in Fig. 1.

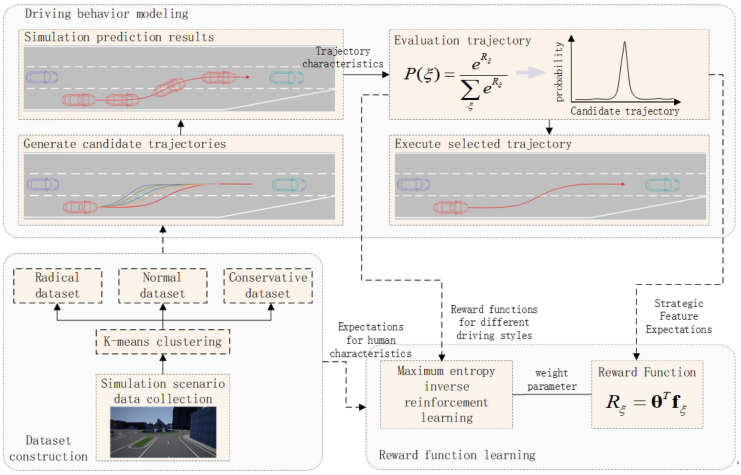


Fig. 1. Structure of This Paper.

The main research content of this paper is as follows:

(1) Construction of ramp merging driving trajectory dataset. Due to the safety and high cost of collecting ramp inflow data in real scenarios, this paper conducts data collection through a simulation environment.

(2) Driver driving style classification. This paper uses principal component analysis and K-means clustering methods to classify the driving styles of drivers, and provides personalized reward functions and driving strategies for drivers with different styles to meet the personalized needs of different drivers.

(3) Personalized driving decision-making learning. A model framework for constructing personalized input decisions using maximum entropy inverse reinforcement learning method to imitate human decision-making mechanisms. In the design of the reward function, multiple driving characteristic parameters such as speed, comfort, safe distance from surrounding vehicles, collisions, and the impact of vehicles on surrounding vehicles are considered. This paper takes into account the driving characteristics of the vehicle itself and its interaction with surrounding vehicles.

3 Math

3.1 Driving Style Classification

Before conducting driving style classification, it is necessary to extract corresponding parameters as feature parameters for driving style classification [20]. In existing studies, there is no specific unified standard for the characteristic parameters that characterize driving style, and different scholars choose different parameters for their research. Based on the specific issues studied in this paper, the selected driving style features are shown in Table 1.

Table 1. Driving style clustering feature selection.

Number	Characteristic Parameter
X_1	Standard deviation of longitudinal speed
X_2	Maximum longitudinal speed
X_3	Average longitudinal acceleration
X_4	Standard deviation of longitudinal acceleration
X_5	Vertical Jerk Average
X_6	Vertical Jerk standard deviation
X_7	Average longitudinal distance from the front vehicle

When extracting and processing features, problems involving high-dimensional feature vectors are often prone to dimensional disasters. Therefore, it is necessary to adopt a dimensionality reduction approach. This paper uses principal component analysis to reduce the dimensionality of the selected feature parameters. The steps of principal component analysis method are: first, standardize the raw data, then calculate the sample correlation coefficient matrix, then calculate the eigenvalues and corresponding eigenvectors of the correlation coefficient matrix, and then select several principal components with higher contribution rates to write the expression of the principal components.

The contribution rate measures the amount of information that the principal component contains about the original variable. Usually, the cumulative contribution rate is used to determine the number of principal components, and it is generally required that the cumulative contribution rate be greater than 85%. Using Python programming to perform principal component analysis on feature parameters, the results are shown in

Table 2. From Table 2, it can be seen that the contribution rate of the first four principal components is 96.82%, and the principal component analysis effect is very good. The first four principal components have already expressed most of the information of the original seven feature parameters well, so the first four principal components can be selected for subsequent research.

Considering the data structure, algorithm efficiency, and computational cost used, this paper selects the K-means clustering method to cluster the driving style features after PCA dimensionality reduction. K-mean clustering first requires setting the number of clusters, and then dividing the samples with smaller distances into one group by calculating the distance between them. Given a sample set $D = \{x_1, x_2, ..., x_n\}$, obtain k cluster partitions $C = \{C_1, C_2, ..., C_k\}$ and calculate the square error.

$$E = \sum_{i=1}^{k} \sum_{x \in C_i} \|x - \mu_i\|_2^2 \tag{1}$$

$$\mu_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{2}$$

Table 2. Principal component analysis results.

Principal Component	Contribution Rate	Accumulated Contribution Rate
X_1	45.53%	45.53%
X_2	22.76%	68.29%
X_3	15.88%	84.17%
X_4	12.65%	96.82%
X_5	1.76%	98.58%
X_6	0.91%	99.49%
X_7	0.51%	100%

μ_i is the mean vector of cluster C_i, and the smaller the squared error E, the higher the degree of tightness between the intra cluster samples and the cluster mean vector, and the higher the similarity between the intra cluster samples.

This paper categorizes driving styles into three categories: Aggressive, Normal, and Conservative. When drivers merge on highways, more aggressive drivers generally choose to merge at a higher speed into the main lane, while conservative drivers choose to merge at a lower speed. Aggressive drivers typically exhibit more rapid acceleration and deceleration behaviors than conservative drivers, as evidenced by the possibility of larger fluctuations in acceleration and deceleration. Moreover, aggressive drivers often choose to merge at a smaller longitudinal distance from the vehicles in front of the main lane.

3.2 A Personalized Ramp Entrance Decision-Making Model Based on Inverse Reinforcement Learning

Considering any traffic scenario, the state $s_t \in S$ observed by human drivers within a time step t includes the position, direction, and speed of themselves and surrounding vehicles, and the actions $a_t \in A$ selected by the driver include the speed and steering of the vehicle, forming the action space. Assuming a discrete time period and a finite length L, the trajectory $\xi = [s_1, a_1, s_2, a_2, ..., s_L, a_L]$ is composed of the state of each time step and the actions taken. The status can be directly obtained from the corresponding sensor.

This paper assumes a reward function with a linear structure. For highway scenarios with simple road structures, where driving modes are relatively stable and drivers' preferences or behaviors do not change significantly over time, this paper assumes that the weights of the reward function are consistent. The form of the reward function is as follows:

$$r(s_t) = \boldsymbol{\theta}^{\mathrm{T}} \mathbf{f}(s_t) \tag{3}$$

where, $\boldsymbol{\theta} = [\theta_1, \theta_2, ..., \theta_K]$ is the K-dimensional weight vector, which is the extracted feature vector in the current state. Therefore, the reward value for the entire trajectory is:

$$R(\xi) = \sum_t r(s_t) = \boldsymbol{\theta}^{\mathrm{T}} \mathbf{f}_\xi = \boldsymbol{\theta}^{\mathrm{T}} \sum_{s_t \in \xi} \mathbf{f}(s_t) \tag{4}$$

where, $\mathbf{f}_\xi$ is the cumulative eigenvalues along the trajectory ξ.

According to the assumption of this paper, human drivers follow a random strategy that induces a distribution on the generated candidate trajectory. This distribution also has the maximum entropy among all matched distributions, corresponding to the maximum entropy IRL. Formally, the probability of a trajectory is directly proportional to its reward index.

$$p(\xi|\boldsymbol{\theta}) = \frac{e^{R(\xi)}}{Z(\boldsymbol{\theta})} = \frac{e^{\boldsymbol{\theta}^{\mathrm{T}} \mathbf{f}_\xi}}{Z(\boldsymbol{\theta})} \tag{5}$$

where, $p(\xi|\boldsymbol{\theta})$ is the probability of the trajectory with given reward parameters $\boldsymbol{\theta}$, and $Z(\boldsymbol{\theta})$ is the partition function. However, the partition function is difficult to deal with for continuous space and high-dimensional space, because it needs to integrate all possible trajectories. According to the assumption, the space of possible trajectories can be simplified into some small subspaces. Therefore, on the basis of assumptions, a limited number of feasible trajectories can be generated and then used to approximate the partition function, so the probability of the trajectories is:

$$p(\xi|\boldsymbol{\theta}) \approx \frac{e^{\boldsymbol{\theta}^{\mathrm{T}} \mathbf{f}_\xi}}{\sum_{i=1}^{M} e^{\boldsymbol{\theta}^{T} \mathbf{f}_{\tilde{\xi}_i}}} \tag{6}$$

where, $\tilde{\xi}_i$ is the generated trajectory with the same initial state as the human driving trajectory ξ, $\mathbf{f}_{\tilde{\xi}_i}$ is the feature vector of the trajectory, and M is the number of generated trajectories. By approximation, $p(\xi|\boldsymbol{\theta})$will be simplified into a probability mass for easy calculation.

3.3 Setting the Reward Function

This paper will develop the form of a reward function from the following aspects:

(1) Efficiency: Designed to reflect the desire of human drivers to reach their destination as soon as possible, defined as the speed of the vehicle:

$$f_{\mathrm{v}}(s_t) = v(t) \tag{7}$$

(2) Comfort: Ride comfort is an important evaluation indicator in autonomous driving, which is mainly measured by calculating longitudinal acceleration a_{x}, lateral acceleration a_{y}, and longitudinal bumps j_{x}:

$$\begin{cases} f_{a_{\mathrm{x}}}(s_t) = |a_{\mathrm{x}}(t)| = |\ddot{x}(t)| \\ f_{a_{\mathrm{y}}}(s_t) = \left|a_{\mathrm{y}}(t)\right| = |\ddot{y}(t)| \\ f_{j_{\mathrm{x}}}(s_t) = |\dot{a}_{\mathrm{x}}(t)| = |\dddot{x}(t)| \end{cases} \tag{8}$$

where, $x(t)$ and $y(t)$ are the vertical and horizontal coordinates, respectively.

(3) Risk avoidance: Human drivers typically tend to maintain a safe distance from surrounding vehicles, which varies among different human drivers, reflecting their different levels of risk perception. In this paper, the risk level between the vehicle and the vehicle in front is defined as an exponential function, which is related to the time from the ego vehicle to the vehicle in front, and is assumed to be a constant speed movement:

$$f_{\mathrm{risk_f}}(s_t) = e^{-(\frac{x_{\mathrm{f}}(t)-x_{\mathrm{ego}}(t)}{v_{\mathrm{ego}}(t)})} \tag{9}$$

where, $x_{\mathrm{f}}(t)$ is the longitudinal position of the vehicle closest to the main vehicle, $x_{\mathrm{ego}}(t)$ is the longitudinal position of the main vehicle, and $v_{\mathrm{ego}}(t)$ is the speed of the ego vehicle.

Similarly, the risk level between the vehicle behind is defined as an exponential function related to the time from the vehicle behind to the main vehicle:

$$f_{\mathrm{risk_r}}(s_t) = e^{-(\frac{x_{\mathrm{ego}}(t)-x_{\mathrm{r}}(t)}{v_{\mathrm{r}}(t)})} \tag{10}$$

where, $x_{\mathrm{r}}(t)$ and $v_{\mathrm{r}}(t)$ represent the longitudinal position and speed of the rear vehicle closest to the main vehicle, respectively.

Meanwhile, when evaluating the trajectories generated in the environmental model, collisions may occur, including collisions with other vehicles or road restrictions. Therefore, collisions are also a risk indicator, defined as:

$$f_{\mathrm{collision}}(s_t) = \begin{cases} 1 & if \quad collision \\ 0 & otherwhise \end{cases} \tag{11}$$

(4) Interaction: Due to the impact of the behavior of the main vehicle on surrounding vehicles, which in turn affects the overall traffic efficiency of the entire environment, in order to ensure traffic efficiency in the entire environment, it is necessary to

consider the impact of the behavior of the main vehicle on surrounding vehicles as an evaluation indicator. Defined as the sum of predicted deceleration caused by the influence of the main vehicle on surrounding vehicles.

$$f_1(s_t) = \sum_i |a_i(t)|, \ if \ \ a_i(t) < 0 \tag{12}$$

where, $a_i(t)$ is the acceleration of vehicle i affected by the main vehicle.

The above features are calculated at each time step and accumulated over time to obtain trajectory features. Then divide the trajectory features by the maximum value in the dataset and normalize them between ranges to offset the effects of different units and scales.

4 Experimental Methods

4.1 Simulation Data Collection

Carla is an open-source simulator specifically designed for autonomous driving research. In addition to providing open-source code and protocols, it also provides a series of open digital assets (city layout, buildings) that can be free used. At the same time, Carla also provides a variety of sensors and traffic participant models, which can meet the needs of multiple training for machine learning and are relatively easy to control. Therefore, this project plans to use the simulation testing platform Carla to build a ramp merging road structure, deploy relevant traffic participants, and collect the driving behavior of human drivers in the traffic scene of merging through the ramp. This paper uses 3D map drawing software to design the corresponding ramp entrance scene, generate the corresponding FBX file, and import it into Carla software.

This paper randomly selects 8 drivers, four males and four females, all of whom have at least two years of driving experience and are aged between 20 and 40. During the data collection process, the driver uses a 3D driving simulator to conduct relevant data collection work in the simulation environment built in this paper. The data collection schematic is shown in Fig. 2.

4.2 Experimental Design

The environment model constructed in this paper is shown in Fig. 3.

To simplify the problem, the target sampling space $\Phi = \{v_{xe}\}$ is reduced to only the longitudinal speed as a variable. Due to the relatively fixed lateral position at the completion of the merging behavior, which is the centerline of the outermost lane of the main lane, it is set to a fixed value and other parameters are set to 0. The sampling range for longitudinal speed is $[v-5, v+5]$ m/s, with an interval of 1 m/s, where v is the initial speed of the vehicle. The time range is 5 s and the sampling interval is 0.1 s. The parameters of IDM are: expected speed $v_0 = v_{\text{current}}$ m/s, maximum acceleration $a_{\max} = 5\,\text{m/s}^2$, expected time interval $\tau = 1\,\text{s}$, comfortable braking deceleration $b = 3\,\text{m/s}^2$, and minimum distance $s_0 = 1\,\text{m}$.

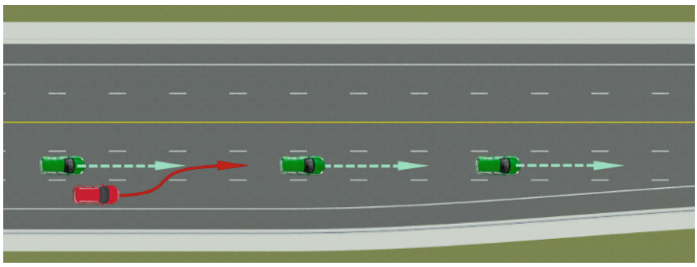


Fig. 2. Data Collection Caption.

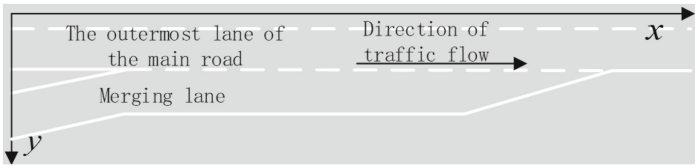


Fig. 3. Schematic Diagram of Road Environment.

This paper needs to set and adjust three hyperparameter: regularization parameters λ, learning rate α and training rounds E. After the test, the final setting of hyperparameter is selected as $\lambda = 0.01$, $\alpha = 0.05$ and $E = 200$.

In terms of selecting evaluation indicators, this paper mainly selects average feature difference and human similarity difference as evaluation indicators. The average feature difference takes into account all the feature parameters involved in the reward function, including vehicle speed $f_{\mathrm{v}}(s_t)$, longitudinal acceleration $f_{a_{\mathrm{x}}}(s_t)$, lateral acceleration $f_{a_{\mathrm{y}}}(s_t)$, longitudinal jerk, risk level with the front vehicle $f_{\mathrm{risk_f}}(s_t)$, risk level with the rear vehicle $f_{\mathrm{risk_r}}(s_t)$, collision $f_{\mathrm{collision}}(s_t)$, and the influence of the main vehicle's behavior on surrounding vehicles $f_I(s_t)$. The average difference between the learned values of these feature parameters and the human driver's feature parameter values is calculated. This can provide a comprehensive description of the corresponding parameter features mentioned above, and evaluate the quality of personalized learning for humanoid individuals from the perspective of driving characteristics. Due to the different units and values of these parameters, it is necessary to normalize the values of each feature parameter before calculating the average feature difference. The smaller the average feature parameter value, the higher the similarity between it and human drivers.

5 Experimental Results

This paper collects a total of 60 imported data in the simulation scenario, and clusters them using the selected feature values using the K-Means clustering method. The obtained results are shown in Fig. 4. Among the collected data, there are 11 cases of aggressive type, 36 cases of normal type, and 13 cases of conservative type.

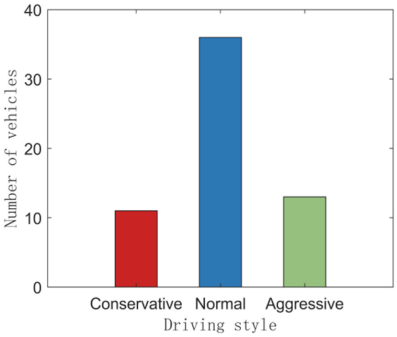
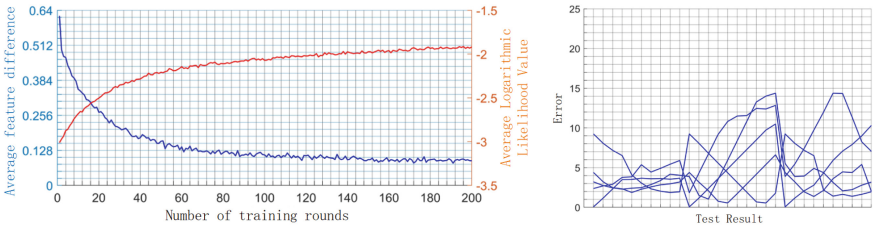


Fig. 4. Cluster Results.

5.1 Aggressive Experimental Results

Based on the clustering results of simulation data, 5 trajectories were randomly selected from the aggressive driver dataset as training data. The training results are shown in Fig. 5(a). From the figure, it can be seen that the average feature difference value continuously decreases after training, and the average logarithmic likelihood of human demonstration trajectories continues to increase. Finally, both of them tend to stabilize. Therefore, the trained model can closely approximate the driving characteristics of human drivers. After 200 rounds of training, the feature difference between the model and the human demonstration data remained stable at around 0.081, and the likelihood logarithm of the human demonstration data continued to increase and stabilize, indicating that the model can better approximate the driving style of aggressive drivers. Randomly select 5 trajectories from the remaining aggressive driving dataset as test scenarios, and use the reward function learned during the training process to evaluate and select candidate trajectories. The training result error is shown in Fig. 5(b).



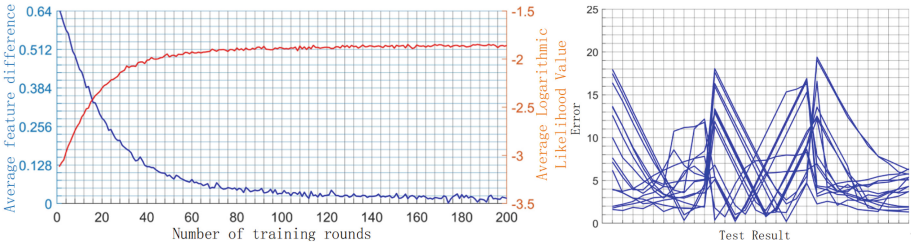
(a) Mean Feature Difference and Mean Logarithmic Likelihood (b) Test Error for Aggressive Drivers

Fig. 5. Aggressive Experimental Results

5.2 Normal Experimental Results

According to the clustering results, 15 trajectories were randomly selected from the normal driver dataset as training data. The training results are shown in Fig. 6(a). From the

figure, it can be seen that the average feature difference and average human similarity difference gradually decrease and tend to stabilize after training. The trained model can closely approximate the driving characteristics of human drivers. After 200 rounds of training, the feature difference between the model and the human demonstration data remained stable at around 0.013, and the likelihood logarithm of the human demonstration data continued to increase and stabilize. It can be seen that the model can better approximate the driving style of normal drivers. Then randomly select 15 tracks from the remaining normal driving dataset as test scenarios, and use the reward function learned during the training process to evaluate and select candidate tracks. The training result error is shown in Fig. 6(b).



(a) Mean Feature Difference and Mean Logarithmic Likelihood (b) Test Error for Normal Drivers

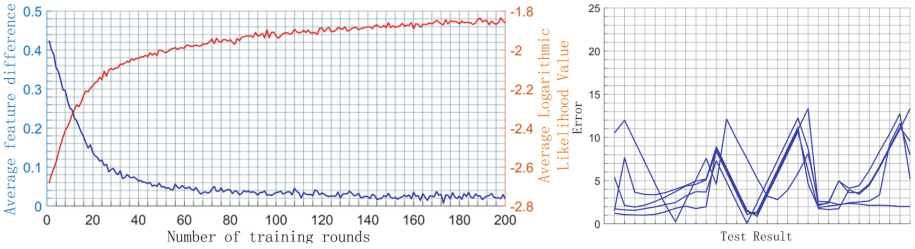
Fig. 6. Normal Experimental Results.

5.3 Conservative Experimental Results

According to the clustering results, 5 trajectories were randomly selected from the conservative driver dataset as training data. The training results are shown in Fig. 7(a). From the figure, it can be seen that the average feature difference and average human similarity difference gradually decrease and tend to stabilize after training. The trained model can closely approximate the driving characteristics of human drivers. After 200 rounds of training, the feature difference between the model and the human demonstration data remained stable at around 0.018, and the likelihood logarithm of the human demonstration data continued to increase and stabilize. It can be seen that the model can better approximate the driving style of conservative drivers. Then, randomly select 5 trajectories from the remaining conservative driving dataset as test scenarios, and use the reward function learned during the training process to evaluate and select candidate trajectories. The training error is shown in Fig. 7(b).

5.4 Comparative Experiment

In order to verify the effectiveness of the personalized decision-making method in this paper, in a random scenario, verify the human similarity difference between the personalized reward function and decision-making strategy learned through driving style classification and the reward function and decision-making strategy learned without



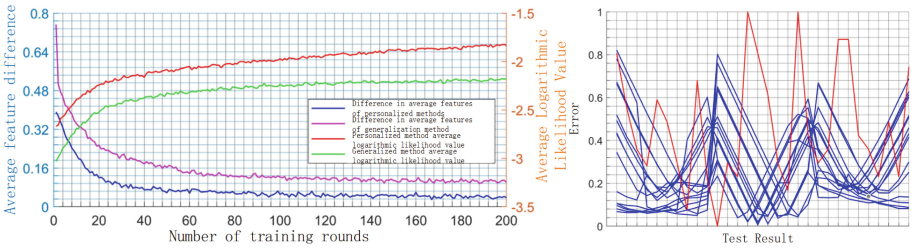
(a) Mean Feature Difference and Mean Logarithmic Likelihood (b) Test Error for Conservative Drivers

Fig. 7. Conservative Experimental Results.

classification, and reflect the effectiveness of the personalized decision-making model proposed in this paper through comparison.

Using a generalized modeling method that has not undergone driving style classification, 30 pieces of data were randomly selected from all driving data as training data. The training results are shown in Fig. 8(a).

In order to verify that this paper can better learn the personalized driving trajectory of drivers by classifying driving styles and increasing the number of reward function feature parameters, the feature parameter errors of the training results were compared with the control experiment. Due to the large number of feature parameters selected in this paper and the relatively high vehicle speed, while only the vehicle speed was selected as the evaluation indicator in the control experiment and the vehicle speed was relatively low, it is necessary to normalize the feature parameters. The error situation of the feature parameters in the multiple training results of the decision model in this paper is shown in Fig. 8(b). From Fig. 8(a), it can be seen that after training, the average feature difference of the personalized method used in this article is significantly smaller than that of the general method, and the average logarithmic likelihood value of the human demonstration trajectory of the personalized method is also significantly higher than that of the general method. From Fig. 8(b), it can be seen that the normalized error of the personalized method is significantly smaller than that of the general method. It can be seen that the personalized decision-making method obtained through driving style classification in this article can better learn the driving strategies of human drivers.



(a) Mean Feature Difference and Mean Logarithmic Likelihood (b) Comparison between different methods

Fig. 8. Comparison between personalized and general methods.

6 Conclusion

This paper conducted experiments on the proposed personalized human like learning inflow model using simulated ramp inflow data, verifying the effectiveness of the proposed model. The results of personalized training showed a stable feature difference of around 0.037. The results of training without personalized methods show a stable feature difference of around 0.098 compared to human demonstration data. The experimental results show that the proposed model can better learn human driving decisions, has high similarity with human demonstration trajectories, and has the ability to make humanoid decision planning. It can reflect the driving characteristics of human drivers and meet the personalized driving needs of drivers with different driving styles. At the same time, this paper also takes the interaction between vehicles as one of the characteristic parameters of the reward function, fully considering the interaction between vehicles, which can take into account the traffic efficiency of the entire traffic scene, and has certain significance in practical application scenarios.

References

1. Liu, S., Spiridonidis, C., Khder, M.A.: Cognitive computational model using machine learning algorithm in artificial intelligence environment. Appl. Math. Nonlinear Sci. **7**(1), 803–814 (2022)
2. Tran, Q., Bae, S.: Comprehensive automated driving maneuvers under a non-signalized intersection adopting deep reinforcement learning. Appl. Sci. **12**(19), 9653 (2022)
3. Cai, L., Yuen, K.F., Wang, X.: Explore public acceptance of autonomous buses: an integrated model of UTAUT, TTF trust. Travel Behav. Soc. **31**, 120–130 (2023)
4. Crayton, T.J., Meier, B.M.: Autonomous vehicles: developing a public health research agenda to frame the future of transportation policy. J. Transp. Health **6**, 245–252 (2017)
5. Itkonen, T.H., et al.: Trade-off between jerk and time headway as an indicator of driving style. PLoS ONE **12**(10), e185856(2017)
6. Chen, Z., Wu, C., Zhong, M., Lyu, N., Huang, Z.: Identification of common features of vehicle motion under drowsy/distracted driving: a case study in Wuhan. China Accid. Anal. Prev. **81**, 251–259 (2015)
7. Ramyar, S., Homaifar, A., Salaken, S.M., Nahavandi, S., Kurt, A.: A personalized highway driving assistance system. In: A personalized highway driving assistance system, pp. 1596–1601. IEEE (2017)
8. Xiao, Z., et al.: On extracting regular travel behavior of private cars based on trajectory data analysis. IEEE Trans. Veh. Technol.Veh Technol. **69**(12), 14537–14549 (2020)
9. Deng, C., Wu, C., Lyu, N., Huang, Z.: Driving style recognition method using braking characteristics based on hidden Markov model. PLoS ONE **12**(8), e182419 (2017)
10. Mohammedalamen, M., Khamies, W.D., Rosman, B.: Transfer learning for prosthetics using imitation learning, arXiv preprint arXiv:1901.04772 (2019)
11. Rajeswaran, A., et al.: Learning complex dexterous manipulation with deep reinforcement learning and demonstrations, arXiv preprint arXiv:1709.10087 (2017)
12. Ng, A.Y., Russell, S.: Algorithms for inverse reinforcement learning. In: Algorithms for inverse reinforcement learning, p. 2 (2000)
13. Abbeel, P., Ng, A.Y.: Apprenticeship learning via inverse reinforcement learning. In: Apprenticeship learning via inverse reinforcement learning, p. 1 (2004)

14. Ratliff, N.D., Bagnell, J.A., Zinkevich, M.A.: Maximum margin planning. In: Maximum margin planning, pp. 729–736 (2006)
15. Ratliff, N.D., Silver, D., Bagnell, J.A.: Learning to search: Functional gradient techniques for imitation learning. Auton. Robot.. Robot. **27**, 25–53 (2009). https://doi.org/10.1007/s10514-009-9121-3
16. Neu, G., Szepesvári, C.: Apprenticeship learning using inverse reinforcement learning and gradient methods, arXiv preprint arXiv:1206.5264 (2012)
17. Neu, G., Szepesvári, C.: Training parsers by inverse reinforcement learning. Mach. Learn. **77**, 303–337 (2009). https://doi.org/10.1007/s10994-009-5110-1
18. Wu, Z., Qu, F., Yang, L., Gong, J.: Human-like decision making for autonomous vehicles at the intersection using inverse reinforcement learning. Sensors-Basel **22**(12), 4500 (2022)
19. Naumann, M., Sun, L., Zhan, W., Tomizuka, M.: Analyzing the suitability of cost functions for explaining and imitating human driving behavior based on inverse reinforcement learning. In: Analyzing the suitability of cost functions for explaining and imitating human driving behavior based on inverse reinforcement learning, pp. 5481–5487. IEEE (2020)
20. Martinez, C.M., Heucke, M., Wang, F., Gao, B., Cao, D.: Driving style recognition for intelligent vehicle control and advanced driver assistance: a survey. IEEE Trans. Intell. Transp. **19**(3), 666–676 (2017)

An Adaptive Cruise Hybrid Control Method for Unmanned Driving

Yuqi Sun[1,2(✉)], Zhexu Wu[1], Xu Wang[1], and Xuyang An[1]

[1] China North Artificial Intelligence and Innovation Research Institute, Collective Intelligence and Collaboration Laboratory, No. 4 Yard, Huaishuling, Fengtai District, Beijing 100072, China
201832021003@alu.cqu.edu.cn

[2] China North Vehicle Research Institute, NORINCO Unmanned Vehicle Research and Development Center, No. 4 Yard, Huaishuling, Fengtai District, Beijing 100072, China

Abstract. With the improvement of computer technology and the development of artificial intelligence technology, the intelligent automobile is constantly advancing. As an important part of Advanced Driving Assistance System (ADAS), Adaptive Cruise Control (ACC) technology has been a research hotspot. In order to make adaptive cruise system adapt to the auto intelligent technology development, considering the mutual interference between the trajectory tracking control and speed control in the unmanned vehicle mode, this paper adopts the direct structure to carry on the design for ACC, realizing the hybrid control with function of trajectory and speed control. Firstly, a trajectory tracking model is designed, and MPC control method is used to track the desired trajectory of the vehicle. Simulation results show that the lateral trajectory error of the vehicle is 0.025 m, which well meets the driving requirements. After using fuzzy PID algorithm for stability control, the simulation results show that the algorithm used in the direct type control system can effectively improve the safe distance for the stability of the vehicle ahead under the condition of constant speed of 60 km/h, the traditional PID actual distance fluctuated among 50 m, while algorithm in this paper controls the actual distance in 15 m. Based on Carsim/MATLAB software, simulation was carried out for constant speed condition, variable speed condition and multi-vehicle condition. The simulation results show that the integrated ACC control system proposed in this paper can meet the requirements of the above conditions.

Keywords: Unmanned Driving · Adaptive Cruise Control System · Hybrid Control

1 Background

As a key part of unmanned driving technology, ACC is an integrated intelligent control system that contains perception, decision making and control. In traditional ACC, radar is used as the lateral distance detector, and lateral position information of vehicles in front is less considered [1]. In the face of multiple moving vehicles on the road ahead, redundant acceleration and deceleration operations will not only affect driving experience but also threaten the driving safety. Besides, when the ACC system cut into the unmanned model,

Y. Qu et al. (Eds.): ICAUS 2023, LNEE 1177, pp. 15–26, 2024.
https://doi.org/10.1007/978-981-97-1103-1_2

the pilots remove their hands from the steering wheel, and no longer operate movement to vehicle accelerator and brake pedal, the need increases to keep the specified steering wheel angle and vehicle maintain a certain speed, which makes the vehicle follow a prescribed trajectory so as to ensure the safety of driving, while the local vehicle also needs to control the forward speed to ensure that the vehicle will not collide with the front vehicle under the condition that the road space is utilized in the most efficient way. At this time, the motion interference between path following and speed control, as well as the collaborative work among the three systems, increase the difficulty of ACC system in controlling the vehicle speed [2]. In order to realize the vehicle trace adapt to complex scenes and effectively control the vehicle speed, which ensures the safety of unmanned vehicles in complex road environment, it is very necessary to study the hybrid control of ACC system.

The mainstream algorithms for speed control mainly include PID control, fuzzy control and deep learning. PID control algorithm got the characteristics of simple structure and stable control, and was widely applied in the field of engineering [3]. As for the longitude speed control, because the speed fluctuation of traditional PID control was large and the parameter adjustment process was too complicated, scholars improved the traditional PID algorithm in different degrees. [4, 5] designed fuzzy adaptive PID controller to adapt to the different speed, but this method did not consider the traffic scene for car unmanned model. Xu and Lu Chihua [6] used genetic algorithm to encode PID three control parameters and used fitness function to select the appropriate control parameter from different combinations generated by crossover and mutation which showed stable following performance for variable vehicle distance with fixed headway. As people's demand for driving comfort and economy increased, ACC control based on single objective was far from meeting people's needs. Therefore, MPC based on multi-objective optimization was widely used in ACC speed control [7–9]. [10] and [11] not only considered the distance, relative speed and acceleration of the main vehicle, but also focused on the variation characteristics of vehicle acceleration (JERK) based on comfort performance. However, in order to ensure the stability of control, MPC algorithm need to solve the time-varying state quantity within the prediction step, which often required a lot of operation time. To solve this problem, [12] used neural network to optimize and simulate the MPC algorithm. The simulation results showed that, the neural network optimization model possessed the same control ability and strong robustness as the original MPC model, and the optimized control system showed lower computational complexity. In the lateral speed control field, MPC controller was employed under different velocities and road friction coefficient [13], however, the velocities carried out in their simulation were constant, and could not fit complex driving condition. Literature [14] designed a method based on EPS and DBC, the joint control strategy adopts DBC control when the controller is located in the classical domain, and EPS control when it is located in the non-domain, thus expanding and improving the working condition adaptability of LDAS, while those methods did not consider bending disturbance caused by vehicle in front of adjacent track under track condition.

Known from the analysis of the above, most existing research of ACC tend to study on perception, decision-making, control system independently, only a few scholars consider ACC technology in the intelligent vehicle as the research object of possible problems

in the application, so considering the path tracking and kinematic interference between the speed control and other issues for unmanned adaptive cruise technology, a solution of ACC mechanism model is proposed in this article.

2 ACC Control System Design

Our control system architecture consists of two modules, Trajectory tracker guarantees the stability of the vehicle trajectory in the process of unmanned model and keeps car from deviation. And the speed controller is designed to keep vehicle in safe driving distance. As is shown in Fig. 1.

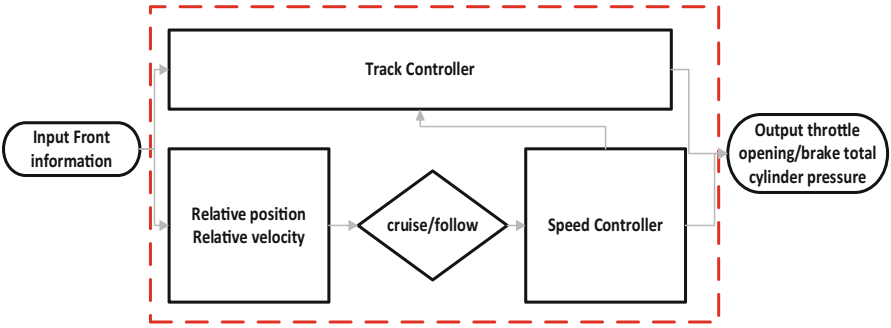


Fig. 1. Integrated speed control model

2.1 The Track Controller Design

When the vehicle enters the driving mode, the uneven road tremor and the vehicle acceleration transformation will be transmitted to the steering wheel through the vehicle suspension and transmission system, resulting in the vehicle track deviation. Therefore, it is necessary to design a stable trajectory tracking model for vehicle lateral stability control. In this paper, the two-wheel model is used as the vehicle dynamics research object, the front wheel angle and the vehicle speed are the input, and take the vehicle longitudinal displacement, lateral displacement, yaw angle as the output to design MPC controller. The vehicle dynamics model is:

$$\begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\rho} \end{bmatrix} = \begin{bmatrix} cos\rho \\ sin\rho \\ \frac{tan\delta}{l} \end{bmatrix} v_p \tag{1}$$

According to Eq. 1, transform the whole dynamic system into the general form of the control system, set $\dot{\zeta} = f(\zeta, \mu)$, where the state variable $\zeta = [\mathrm{XY}\rho]^T$ and control variable $\mu = [v, \delta]^T$.

In the process of modeling, the track of road model is taken as the reference track, the reference point of state quantity is $\zeta_r = [X_r \mathrm{Y}_r \rho_r]^T$ and the corresponding control

quantity is $\mu_r = [v_r \delta_r]^T$. Equation 2 is expanded along the reference point ζ_r, and the results are as follows:

$$\dot{\zeta} = f(\zeta_r, \mu_r) + \frac{\partial f(\zeta, \mu)}{\partial \zeta}\Big|_{\substack{\zeta = \zeta_r \\ \mu = \mu_r}} (\zeta - \zeta_r) + \frac{\partial f(\zeta, \mu)}{\partial \mu}\Big|_{\substack{\zeta = \zeta_r \\ \mu = \mu_r}} (\mu - \mu_r) \quad (2)$$

order to make the vehicle position as close as possible to the reference track and minimize the control quantity input of the system, the objective function is set as follows:

$$\mathrm{j(k)} = \min\Big(\big(\mathrm{z} - \mathrm{z}_{ref}\big)^T \mathrm{Q}\big(\mathrm{z} - \mathrm{z}_{ref}\big) + \mathrm{U}^T \mathrm{RU}\Big) \quad (3)$$

We set the constraint of control quantity and control increment. The front wheel angle of the vehicle is $-45° \leq \delta \leq 45°$ to realize the tracking of the vehicle for the specified trajectory.

Figure 2 displays the tracking result of vehicle under the condition of straight line. The linear tracking error of the vehicle is within 0.001 m, which meets the vehicle's requirements for driving track under given working conditions.

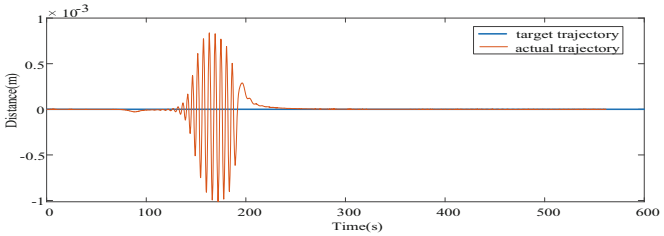


Fig. 2. Linear trajectory tracking

2.2 Design of Speed Controller

After the design of tracking controller, the speed controller is designed so that the vehicle can keep a safe driving distance from the vehicle ahead without deviating from the established driving track. Considering the realistic scenario, the vehicle front view usually contains multiple vehicles. Therefore, taking the vehicles ahead whether interference with the virtual feasible domain as the judgment standard, and taking the minimum actual longitudinal distance as a basis to the car, we design the multi-vehicle safe distance on the basis of variable the headway safety distance model. As shown in Fig. 3.

The multi-vehicle safe distance allocation model consists of two parts:

1. Vehicle identification model in current lane: Given the vehicle width B, lane width L, yaw Angle of the front target I is θ_i, and the actual distance between the front target and the vehicle s_i, the interference criterion between the front target I and the virtual feasible region is $s_i * \sin\theta_i - 0.5\mathrm{l} < 0.5\mathrm{b} + \varepsilon$, and θ is obtained from $\theta = \arctan(\mathrm{x/y})$, x and y respectively are the longitudinal and lateral distances based on the vehicle coordinate system, where ε is the safety factor and is set to 0.5 m.

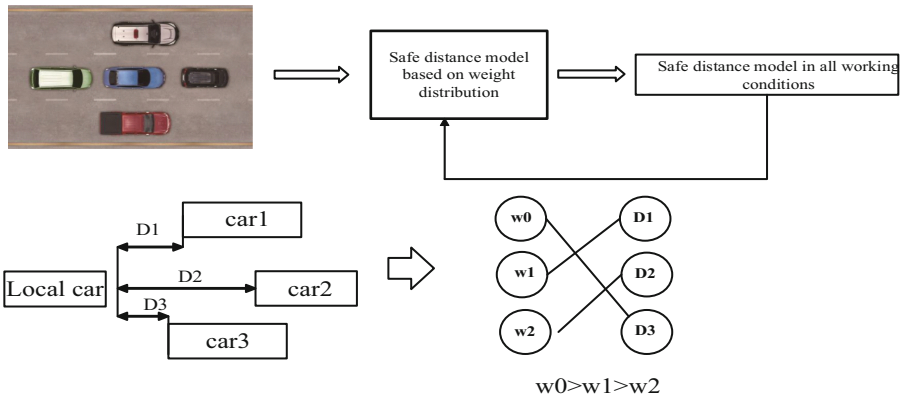


Fig. 3. Multi-vehicle safe distance

2. Weight distribution model of multi-vehicle safe distance:

Assuming that the local car processes the color of green in Fig. 3, first of all, the vehicle obtains the relative distances D_1, D_2 and D_3 of the three front vehicles and assigns different weights w_0, w_1 and w_2 according to the relative distances to generate the actual distance D of multiple vehicles:

$$D = w_0 * D_3 + w_1 * D_1 + w_2 * D_2 \tag{4}$$

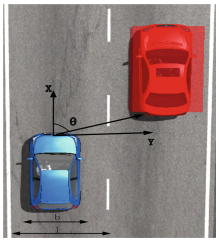


Fig. 4. Schematic diagram of weight update

Then the weight of the vehicle is updated according to whether the vehicle in front interferes with the virtual feasible region, as shown in Fig. 4.

Taking two front vehicles I and J for examples, first we judge whether I and J interfere with the virtual feasible region. When both I and J are not in the interference state, the relative distance weight of I and J is set to 0, and the vehicle executes the autonomous cruise mode. When only one vehicle (such as I) is in the virtual feasible region, the relative distance weight distribution of J update to 0, and the local vehicle selects $d_i = s_i * \cos\theta_i$ as the actual distance. When I and J both interfere with the virtual feasible region, the actual distance d_i and d_j are compared, and the smaller one is assigned to 1, while the other is assigned weight 0. In the case of multiple vehicles, In order to improve

the utilization rate of roads, the unmanned driving should shorten the workshop distance as much as possible, the minimum actual distance is finally selected as the basis for ACC to follow.

Due to the fact that constant vehicle distance is unable to adapt to complex road conditions, the safety distance model of variable headway stability is poorer, the safety distance model with variable headway is commonly used, but in the actual traffic scene, vehicle acceleration and relative velocity in the scene are the factor that cannot be ignored, Therefore, this paper builds a safe distance model in all working conditions based on variable headway.

The safe distance of variable headway Δx_{des} and variable headway t_h are calculated as follows:

$$\Delta x_{des} = t_h v + \Delta x_0 \tag{5}$$

$$t_h = \begin{cases} t_{h\_max} t_0 - c_v v_{rel} - c_a a_p > t_{h\_max} \\ t_0 - c_v v_{rel} - c_a a_p t_{h\_min} < t_0 - c_v v_{rel} - c_a a_p < t_{h\_max} \\ t_{h\_min} t_0 - c_v v_{rel} - c_a a_p < t_{h\_min} \end{cases} \tag{6}$$

where v_{rel} is the relative velocity of local vehicle and front vehicle, $t_{h\_max} = 2.2\,s$, $t_{h\_min} = 0.2\,s$, $t_0 = 1.5\,s$, $c_v = 0.05 . c_a = 0.3$ which is a constant greater than 0, Δx_{des} represent desire distance, a_p is the acceleration of front vehicle, Δx_0 is the minimum distance of the vehicles, which is 3 m.

According to traffic laws and regulations, we choose 70 km/h as cruise speed, and the control strategy is traditional PID. In order to optimize the instability of variable headway, this paper adopts fuzzy PID algorithm for tracking control. Set the output s as control object:

$$s = \Delta x - \Delta x_{des} + \alpha v_{rel} \tag{7}$$

where $v_{rel} = v_1 - v_2$, v_1 is the speed of local vehicle, v_2 is the speed of the front vehicle. In practical operation, when a driver is dealing with various uncertain conditions, he will use his own experience in navigation, but it is impossible to use this experience as an algorithm directly. Therefore, this paper utilized a fuzzy PID controller to control the output s. By adjusting the input and output parameters of the controller, the vehicle speed control with adaptive adjustment parameters can be realized. Finally, fuzzy resolution is carried out according to the fuzzy parameters and PID control parameters are calculated.

$$\begin{aligned} K_{P(k+1)} &= \Delta K_P + K_{P(k)} \\ K_{I(k+1)} &= \Delta K_I + K_{I(k)} \\ K_{D(k+1)} &= \Delta K_D + K_{D(k)} \end{aligned} \tag{8}$$

As shown in Fig. 5, R is the set value of the real-time acquisition system; Y is the real-time output acquisition system; E and EC get the fuzzy output set through the fuzzy rules, and get the adjustment amount of each parameter. After tuning with the initial PID, the new PID parameters are finally obtained. The initial PID values used in this paper are 0.3 and 8, 0.0001 respectively.

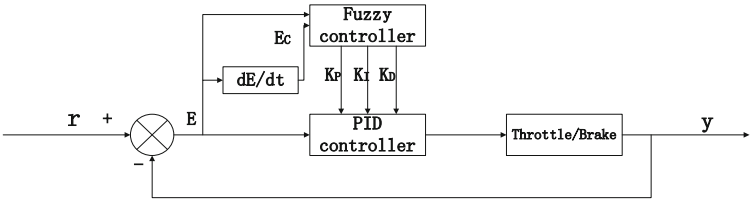


Fig. 5. Schematic diagram of fuzzy controller

Set the relative distance between the initial position of the front vehicle and the vehicle as 80 m, the initial speed of the local vehicle is 0 km/h, and the initial speed of the front vehicle is 60 km/h. The result of vehicle distance control with variable headway using the traditional PID control method and Fuzzy PID are shown in Fig. 6 and Fig. 7.

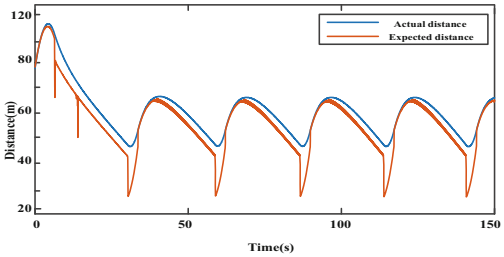


Fig. 6. Traditional PID control

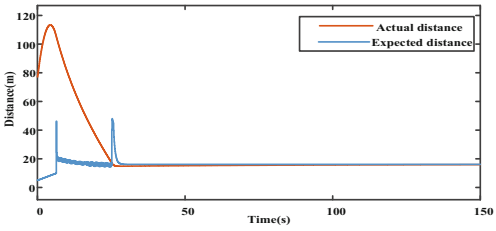


Fig. 7. Fuzzy PID control

As can be seen, the actual distance of the strategy without fuzzy PID algorithm fluctuates around 50 m, while the control strategy with fuzzy PID algorithm is 15 m. Obviously, in high-speed driving environment, the variable headway strategy using fuzzy PID algorithm can improve the utilization rate of road under the precondition of ensuring driver safety.