Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 20(3): 6016–6029.
DOI: 10.3934/mbe.2023260
Received: 24 October 2022
Revised: 20 December 2022
Accepted: 29 December 2022
Published: 18 January 2023

Research article

Research and implementation of variable-domain fuzzy PID intelligent control method based on Q-Learning for self-driving in complex scenarios

Yongqiang Yao[1], Nan Ma[2,*], Cheng Wang[1], Zhixuan Wu[1], Cheng Xu[1] and Jin Zhang[1]

[1] Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing 100101, China

[2] Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China

* **Correspondence:** Email: manan123@bjut.edu.cn.

Abstract: In the control of the self-driving vehicles, PID controllers are widely used due to their simple structure and good stability. However, in complex self-driving scenarios such as curvature curves, car following, overtaking, etc., it is necessary to ensure the stable control accuracy of the vehicles. Some researchers used fuzzy PID to dynamically change the parameters of PID to ensure that the vehicle control remains in a stable state. It is difficult to ensure the control effect of the fuzzy controller when the size of the domain is not selected properly. This paper designs a variable-domain fuzzy PID intelligent control method based on Q-Learning to make the system robust and adaptable, which is dynamically changed the size of the domain to further ensure the control effect of the vehicle. The variable-domain fuzzy PID algorithm based on Q-Learning takes the error and the error rate of change as input and uses the Q-Learning method to learn the scaling factor online so as to achieve online PID parameters adjustment. The proposed method is verified on the Panosim simulation platform.The experiment shows that the accuracy is improved by 15% compared with the traditional fuzzy PID, which reflects the effectiveness of the algorithm.

Keywords: self-driving; intelligent control; Q-Learning; variable-domain fuzzy PID

1. Introduction

PID controller and fuzzy logic-based controller dominate the field of intelligent automation control by virtue of their scalability and structural simplicity [1]. They have been widely used in vehicle corner speed control [2, 3]. However, these conventional controllers are highly parameter-dependent. In practice, the adjustment of parameters relies mainly on expert experiences, which is not only time-consuming, but once unable to automaticallly adjust when the control parameters are determined. And It cannot be executed well in a single scene. Generally, if the system is used for other tasks, the

control effect often change, and the control parameters need to be adjusted frequently [4]. There are many diversity, time-varying and uncertainty problems in typical driving scenes, such as overtaking, meeting on narrow road, etc. [5, 6]. Therefore, tuning the parameters of conventional controllers to achieve optimal performance has become a popular research direction [7]. To handle complicated processes and combine the benefits of traditional controllers with human operator knowledge, fuzzy control [8] has recently emerged as an alternative to conventional control algorithms. Sang Hyuk Park [9] proposed a method for fuzzy self-tuning PID controller and a method for online tuning of PID controller gain values. Priyam [10] proposed PID and fuzzy PID control for DC motors and compares the results of two methods, concluding that fuzzy PID is more adaptable than classical PID. Quan et al. [11] used fuzzy PID control to study the thermal degradation behavior and kinetics of biomass microwave pyrolysis, which significantly improved the response speed of the system. Jie et al. [12] proposed a Type-2 fuzzy PID controller for improving the drive performance of autonomous mobile robots in static obstacle environments. Neerendra [13] proposed that satisfactory results can be achieved using fuzzy logic and laser sensors for navigation in unknown environments. Muhammad [14] proposed a method for designing robust Fuzzy tuned PD (FPD) controller with better tracking and minimal ball oscillations under different lighting conditions.

Reinforcement learning [15] is a machine learning algorithm that generates optimal policies by interacting with the environment. Various approaches such as Q-Learning and deep deterministic policy gradient algorithm (DDPG) have been developed for the particular self-driving decision making and control tasks, driving policy generation [16], the autonomous mobile robot obstacle avoidance [17], robot manipulator control [18], path planning [19], and path tracking [20], etc. Traditional control methods with reinforcement learning have proven to be very effective in solving problems in high-dimensional action spaces and state spaces [21]. Many researches use reinforcement learning algorithms to completely replace traditional controllers and apply them to control strategies. One such approach is to use reinforcement learning algorithms to completely replace traditional controllers. Ramanathan [22] used Q-Learning to control the level in a non-linear conical tank with satisfactory results. It is worth noting, however, that the selection of state space, action space, and rewards all affect the performance of the algorithm in a reinforcement learning algorithm, and designing a controller by reinforcement learning algorithms alone becomes challenging when the number of dimensions increases [23].

Other methods are to combine traditional controllers with reinforcement learning methods to compensate for the shortcomings of traditional controllers. Lakhani et al. [24] proposed an automatic PID tuning framework based on reinforcement learning (RL), particularly the deterministic policy gradient (DPG) method to address traditional PID tuning methods rely on trial and error for complex processes where insights about the system is limited and may not yield the optimal PID parameters. Dogru et al. [25] combined the recent developments in computer sciences and control theory to address the tuning problem. It formulates the PID tuning problem as a reinforcement learning task with constraints.Yu et al. [26] proposed a self-adaptive model-free SAC-PID control method based on reinforcement learning for automatic control of mobile robots. The upper controller based on soft actor-critic (SAC), one of the most competitive continuous control algorithms, and the lower controller based on incremental PID controller.

Q-learning algorithm has been proved as one of the most efficient model-free RL algorithms by directly parameterizing and updating value functions or policies without explicitly modeling the environ-

ment. Therefore, we design a variable-domain fuzzy PID controller based on Q-Learning. Specifically, we use a variable-domain fuzzy PID controller to change the initial domain of the fuzzy PID controller by scaling its factors, which can change with the deviation and the rate of change of the deviation to achieve the effect of intelligent adjustment of the PID parameters. However, the control function of a variable-domain fuzzy PID controller can become distorted, which decreases control accuracy. Therefore, this paper combines reinforcement learning algorithm to improve the variable-domain fuzzy PID controller, so that it has the ability of online optimization. To improve the effectiveness of adjusting the PID control settings, these two effects are combined and added.

2. Methods

2.1. PID

PID uses differential, integral, and proportional control to calculate the system's error, as is shown in Figure 1.

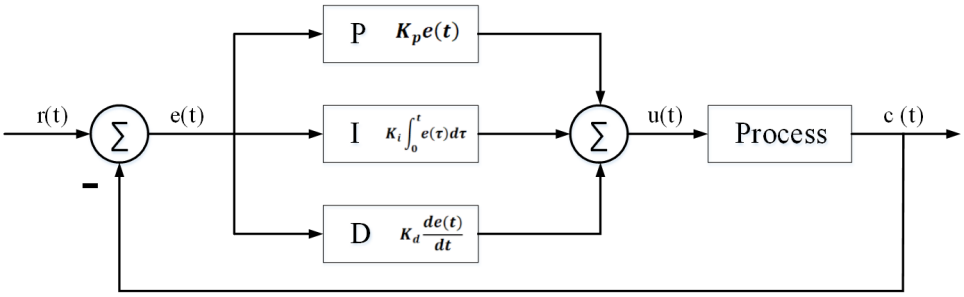


Figure 1. Principle diagram of conventional PID control.

The PID algorithm uses a linear mix of proportional, integral, and differential error information $e(t)$ to choose the desired control parameters:

$$u(t) = k_p e(t) + k_i \int_0^t e(t) + k_d \frac{de(t)}{dt} \tag{2.1}$$

where k_p, k_i , and k_d are the proportional, integral and differential coefficients respectively. In self-driving vehicle control, the deviation between $e(t) = r(t) - c(t)$, $c(t)$ is the actual trajectory and $r(t)$ is the preset trajectory. P (proportional) control is the basis of PID control and proportionally reflects the deviation signal of the control system, which is immediately controlled to reduce the deviation as soon as it occurs. When only P control is available, steady-state error or overshoot is generated. I (integral) control is used to eliminate steady-state error (steady-state error is the very small error that still exists between the vehicle's stable driving and the preset trajectory under PD control), and the error is integrated as long as it exists, so that the output continues to increase or decrease until the error is zero, the integration stops and the output no longer changes. D (differential) control reflects the trend of the deviation signal and can introduce an effective early correction signal in the system before the deviation signal becomes too large, thus speeding up the action of the system and reducing the regulation time. In terms of time, the P control adjusts for current errors, the I control adjusts for historical errors and the D control adjusts for future errors.

Although PID control has many advantages, such as simple structure, good stability and so on, it

is difficult to perform well in the face of the complex environment of self-driving vehicles and time-varying non-linear systems like speed and corner control using a set of PID parameters.

2.2. *Fuzzy PID*

Fuzzy PID control [28] is a method for optimising the parameters of a PID in real time using fuzzy logic and according to certain fuzzy rules, in order to overcome the shortcomings of traditional PID which cannot adjust the PID parameters in real time. Fuzzy PID control includes components such as fuzzification, determination of fuzzy rules and defuzzification.

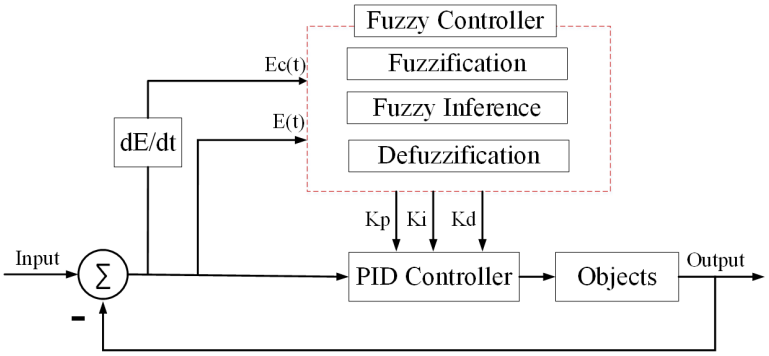


Figure 2. Fuzzy PID control schematic.

The fuzzy PID is in the form of a two-input, three-output controller that takes the error $e(t)$ and the rate of change of the error $ec(t) = \mathrm{de(t)/dt}$ as inputs $x_i(t)\,(i = 1, 2)$ and $x_1(t) = e(t), x_2(t) = ec(t)$, the three parameters of the PID are proportional, integral and differential corrections $\Delta kp, \Delta ki, \Delta kd$ as outputs $y_j(t)\,(j = 1, 2, 3)$ and $y_1(t) = \Delta kp, y_2(t) = \Delta ki, y_3(t) = \Delta kd$ and set the initial domain to be respectively $X_i = [-E_i, E_i]$, and $Y_j = [-U_J, U_J]$, where $E_i, a_n U_J$ is the boundary of the theoretical domain. The theoretical domains of both input and output variables are divided into 7 fuzzy subsets: NB (positive large), NM (positive medium), NS (positive small), ZO (zero), PS (negative small), PM (negative medium), PB (negative large), and determine the form of the affiliation function, then, the output variables are obtained by 3 processes of input fuzzification, fuzzy inference and defuzzification $\Delta kp, \Delta ki, \Delta kd$ The final control quantity is determined according to the Eq (2.2).

$$u(t) = \left(k_{p0} + \Delta kp\right) e(t) + (k_{i0} + \Delta ki) \int_0^t e(t) + (k_{d0} + \Delta kd) \frac{de(t)}{dt} \tag{2.2}$$

where k_{p0}, k_{i0}, and k_{d0} are the initial design value of the PID parameters, designed by the conventional PID controller parameters rectification method.$\Delta kp, \Delta ki, \Delta kd$ are the three outputs of the fuzzy controller, which can automatically adjust the values of the three PID control parameters according to the state of the controlled object.

2.3. *Variable-domain fuzzy PID*

When the size of the theoretical domain in a fuzzy PID controller is not chosen properly, it is more difficult to guarantee the control effect of the fuzzy controller, so variable-domain fuzzy PID controller [29] is born.

As is shown in Figure 3, the variable theoretical domain fuzzy PID adjusts the theoretical domain range of the input and output in the fuzzy controller online by introducing a scaling factor. According

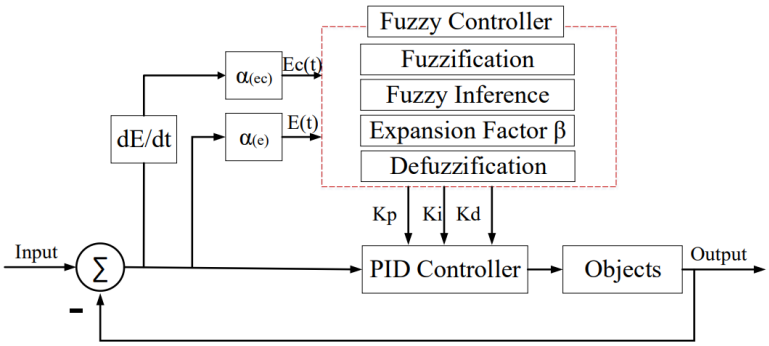


Figure 3. Variable-domain fuzzy PID.

to $e(t)$ and $ec(t)$ calculate the scaling factor $\alpha(e(t)), \alpha(ec(t)), \beta(e(t), ec(t))$, where $\alpha(e(t)), \alpha(ec(t))$ are the scaling factors of the input variables $e(t)$ and $ec(t)$ and $\beta(e(t), ec(t))$ are the scaling factors of the three output variables $\Delta kp, \Delta ki, \Delta kd$ the common scaling factor. Then, the initial theoretical domains of the input and output variables are adjusted for scaling, taking the ith input as an example, the new theoretical domain obtained after the adjustment is $[-\alpha(x_i(t))E_i, \alpha(x_i(t))E_i]$ The new domain is obtained after the adjustment. [30] designed a new scaling factor, as shown in Eq (2.3).

$$\begin{cases} \alpha(e(t)) = \left(\dfrac{|e(t)|}{E_1}\right)^{\tau_1} + \varepsilon \\ \alpha(ec(t)) = \left(\dfrac{|ec(t)|}{E_2}\right)^{\tau_2} + \varepsilon \\ \beta(e(t), ec(t)) = \dfrac{\alpha(e(t)) + \alpha(ec(t))}{2} + \varepsilon \end{cases} \tag{2.3}$$

where ε is a sufficiently small positive number, the E_1 and E_2 are the initial domain boundaries of the input variables, respectively, and $\tau_i (i = 1, 2)$ is the scaling factor design parameter, and $\tau_i \in [0, 1]$.The scaling factor should be stable. [30] verified the stability of the new scaling factor from five aspects: duality, zero avoidance, monotonicity, coordination and normality.

2.4. Reinforcement learning

Reinforcement learning is a subfield of machine learning, as is shown in Figure 4, and intelligence robots learn a strategy for maximizing expected future rewards by interacting with its environment $\pi(s)$ [31] which defines which action a should be taken in each state s. When an action is performed and the environment shifts to a new state s, the intelligence receives a reward r.

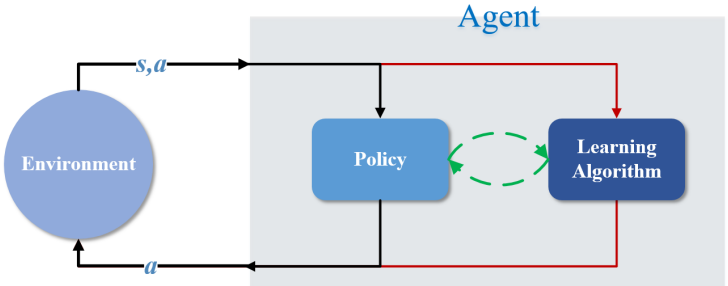


Figure 4. Agent-environment interaction in reinforcement learning.

The reinforcement learning process can be modelled as a Markov decision process (MDP) [32], the Markov decision process is defined by the tuple (S, A, T, R, γ). S represents the state space, A represents the action space, T represents the state transfer function, R represents the reward function, and γ represents the discount factor. At each time step t, the maximise reward is

$$R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k} \tag{2.4}$$

Reinforcement learning algorithms are divided into policy-based algorithms and value-based algorithms. In value-based reinforcement learning, the value function is used to estimate the value of being in each state. The state value function is derived from the states given under the strategy π.

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_\pi \left[R_t \mid S_t = s \right] \\ &= \mathbb{E}_\pi \left[\int_{k=0}^{\infty} \gamma^k r_{t+k} \mid S_t = s \right] \end{aligned} \tag{2.5}$$

where $\mathbb{E}_\pi$ denotes the expectation under strategy π. Similarly, the state action value function for taking action a in state s under strategy π. q_π can be defined as

$$\begin{aligned} q_\pi(s, a) &= \mathbb{E}_\pi \left[R_t \mid S_t = s, A_t = a \right] \\ &= \mathbb{E}_\pi \left[\int_{k=0}^{\infty} \gamma^k r_{t+k} \mid S_t = s, A_t = a \right] \end{aligned} \tag{2.6}$$

Q-Learning [33] is an offline value-based algorithm with a value function based on state actions and an iterative formula that can be described as

$$Q_{t+1}(s_t, a_t) \leftarrow Q_t(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right] \tag{2.7}$$

where $\alpha \in (0, 1]$ is the learning rate, and $\gamma \in (0, 1]$ is the discount factor describing the importance of weighting between immediate and future long-term rewards [34].

2.5. Variable-domain fuzzy PID based on Q-Learning

Traditional PID control has its own limitations in that the parameters cannot change with the external environment, adding fuzzy PID control allows for improved robustness. In order to increase the accuracy of fuzzy control, the theoretical domain can be changed by adding a scaling factor to optimise PID control, i.e., variable theoretical domain fuzzy PID control. The control function in a variable-domain fuzzy PID controller is self-adjusting through constant self-adjustment, making the PID parameters self-tuning. The control function is reduced or enlarged over time, making the updated control function less accurate and generating distortion. As is shown in Figure 5, the Q-Learning algorithm is added to make the scaling factor have the ability to find the optimum online, thus adjusting the PID parameters more accurately and seeking the optimum PID parameters for the current operating conditions.

This section analyses the key scaling factors in variable theoretical domain fuzzy PID control. Taking Eq (2.3) as an example, what determines the size of the theoretical domain are its two parameters τ_i $(i = 1, 2)$ When these two parameters are changed, the domain is changed. By invoking Q-learning control of these two parameters, a better control effect can be achieved, making the controller capable of learning and online correction, and the system is more resistant to disturbances and more robust.

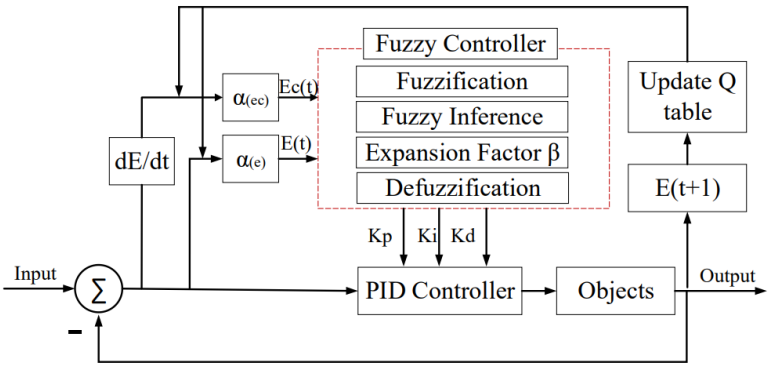


Figure 5. Variable-domain fuzzy PID based on Q-Learning.

In the process of Q-learning, the change in parameters $\Delta\tau i$ ($i = 1, 2$) as the action set, *i.e.*, $A = \{-0.075, -0.05, -0.025, 0, 0.025, 0.05, 0.075\}$, and the state set is the deviation e. The Q matrix is built. The reward function is related to the rate of change of error. A negative rate of change of error indicates that this learning direction is the reward direction and can continue to be adjusted in this direction, a positive rate of change of error indicates that this learning direction is the penalty direction and should be adjusted in the opposite direction at this time.

$$r(t) = \begin{cases} \dfrac{1}{\omega|e(t)|} + g, \text{ if ec } (t) < 0 \\ \dfrac{1}{\omega|e(t)|} - g, \text{ if ec } (t) > 0 \end{cases} \tag{2.8}$$

where $r(t)$ is the immediate reward at moment t, $e(t)$ is the deviation between the actual trajectory $c(t)$ and the preset trajectory $r(t)$, *i.e.*, $e(t) = r(t) - c(t)$, and $ec(t)$ is the rate of change of the error at moment t, *i.e.*, $ec(t) = e(t) - e(t-1)$, denotes the direction of learning. g is the penalty term, and ω is the weight, and the iteration of the Q-value table is carried out through Eq (2.7).

Algorithm 1 Variable-domain fuzzy PID based on Q-Learning (Q-fuzzyPID) algorithm

Require: ϵ, P, γ, the maximum simulation step N.
Initialize: Q, S, τ_i, t.
Loop
If $S = 0$ with probability ϵ select a random action to get out of the local minimum
Else
Select $a_t = \text{argmax}_{a'} Q(s_i, a'; \theta_{t|t-1})$ from the selection criterion of the current parameters τ_i
End
If $t < N$
State Update the Q-value function $Q(s, a)$ with updated action a.
End
$t = t + 1$
Outputs: The parameters τ_1, τ_2.

3. Experiment

To verify the effectiveness of the proposed method, we trained a Q-Learning based variable-domain fuzzy PID controller and built urban road scenarios, including large curvature scenarios, overtaking scenarios, and following scenarios, in a Panosim simulation environment, as is shown in Figure 6.

Figure 6. Simulation environment scenario.

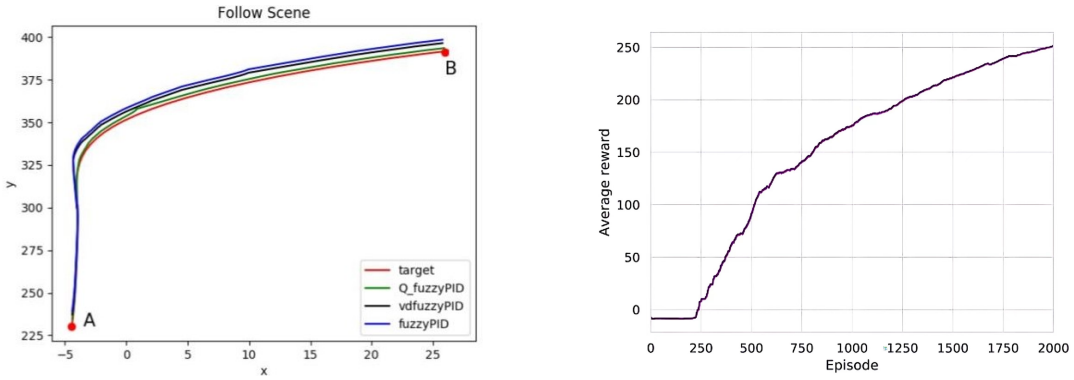
3.1. Simulation settings

Before stating the experimental results, the parameter values are presented. The stretching factor was designed according to Eq (2.3), initialized ε to 0.01, and adjusting the parameters of the scaling factor online using Q-Learning $\tau_i (i = 1, 2)$, $\tau_i \in [0, 1]$, initialized τ_i to 0.5. Initialize the theoretical domain of E as [-350, 350] , Ec as [-100, 100] , Δkp as [-20, 20] , Δki as [-5, 5] and Δkd as [-10, 10] , and divide the input and output into 8 parts with affiliation values of $NB, NM, NS, ZO, PS, PM, PB$. The input E The corresponding intervals of affiliation values are [-350, -262.5], [-262.5, -175], [-175, -87.5], [-87.5, 0], [0, 87.5], [87.5, 175], [175, 262.5], [262.5, 350] . According to Eq (2.3), the E_1 is 350 and E_2 is 100. When the error and the rate of change of the error are greater than the domain boundary, the value is taken as the boundary. The reward function is measured according to the real-time trajectory error, so initialize ω to 1 and initialized g to 20.

3.2. Simulation experiments

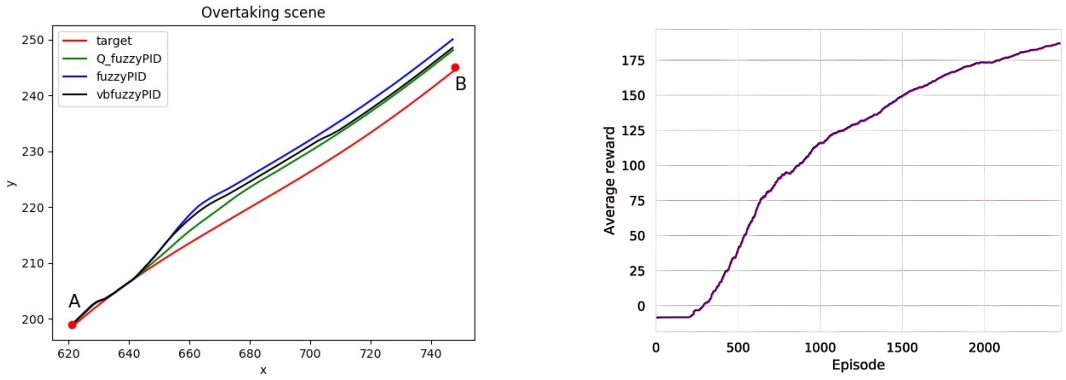
The whole experimental process is divided into training and testing, with training conducted in the following and overtaking scenarios, where the overtaking trajectories are generated from Bessel curves, and testing conducted in the cornering scenario. The proposed method is compared with the fuzzy PID and variable-domain fuzzy PID (vdfuzzyPID) using the error between the actual trajectory and the preset trajectory as an indicator.

As is shown in Figures 7–9, we comprise between fuzzy PID control,variable-domain fuzzy PID control and variable-domain fuzzy PID control based on Q-Learning. And the red line is the expected vehicle trajectory, the blue line is the vehicle trajectory under fuzzy PID control, the black line is the vehicle trajectory under variable-domain fuzzy PID control, and the green line is the vehicle trajectory under variable-domain fuzzy PID control based on Q-Learning. Figures 7(a)–9(a) shows the target trajectory and the actual trajectory of the three scenarios respectively, in which A indicates the starting point, B indicates the ending point. Figure 7(a) shows the control effect in the following scenario. Since the vehicle speed is slow and the corner change is small in the following scenario, the control



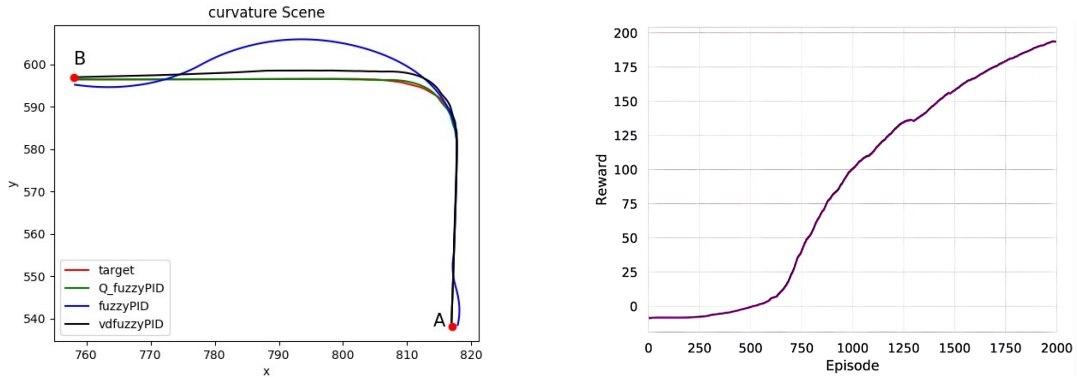
(a) **(b)**

Figure 7. Following scenario paths and performance curves. (a) is the real-time trajectory, (b) is the average reward function curve.



(a) **(b)**

Figure 8. Overtaking scenario paths and performance curves. (a) is the real-time trajectory, (b) is the average reward function curve.



(a) **(b)**

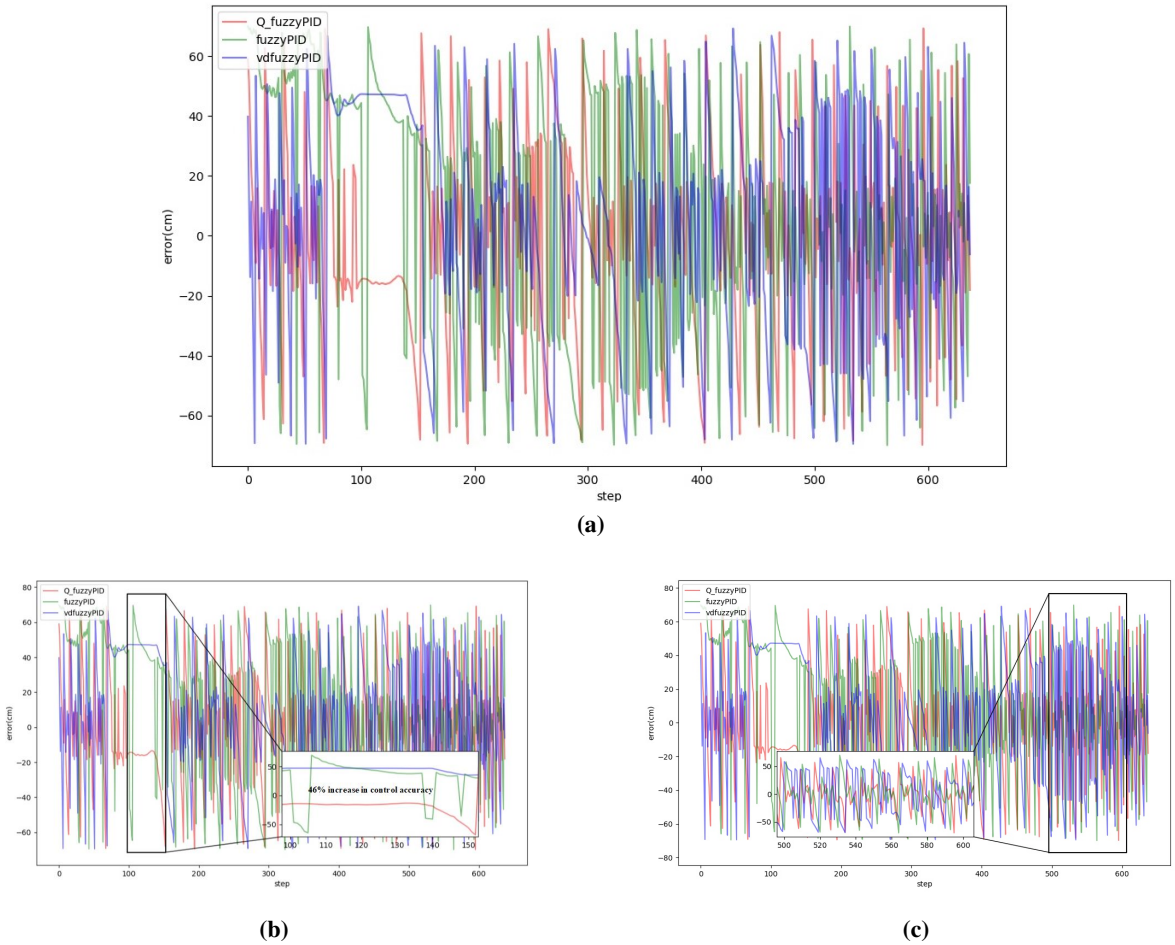
Figure 9. Curvature scenario paths and performance curves. (a) is the real-time trajectory, (b) is the average reward function curve.

effect is no different between the three methods. Figure 8(a) shows the control effect in the overtaking scenario, where there is a slow moving vehicle in front of the original trajectory obstructing the car

from moving forward, so it make a left lane change decision, and it can be seen from the steering path that the steering is smoother than fuzzy PID under variable-domain fuzzy PID control based on Q-Learning. Figure 9(a) shows the control effect in the curvature scenario, where variable-domain fuzzy PID control based on Q-Learning results in a trajectory that is more closely aligned with the preset trajectory, and fuzzy PID results in a larger actual trajectory arc during cornering due to the slow response time and small change in steering angle.

As shown in Figures 7(b)–9(b), the average reward curve for each path showed a steady upward trend with no significant drop or fluctuation, which means that the training process was very stable.

The control error is used as a measure of the stability of the proposed method, and the error varies with the movement of the vehicle. As shown in Figure 10, it is a real time error and local zoom of the car following scene. As shown in Figure 10(a), the control errors of both methods fluctuate around 0. The enlarged area in Figure 10(b) corresponds to the curve in Figure 7(a). It can be seen from both the trajectory and the error diagram that the fuzzy PID control effect and variable-domain fuzzy PID effect are poor. Figure 10(c) the error change in the enlarged area is relatively dense. It can be seen from the enlarged view that Q-fuzzyPID control error is small.



(a) **(b)** **(c)**

Figure 10. Real-time control error and local zoom of following scenario. The red line is the real-time control error of Q-fuzzyPID, the green line is the real-time control error of fuzzy PID, the blue line is the real-time control error of vdfuzzyPID.

As shown in Figure 11, it is real-time control trajectory error of overtaking and curvature. As shown in Figure 11(a), after lane changing and overtaking, the error between the actual control track and the preset track of the three methods is large, and the control effect of the proposed method will be better by comparison. Figure 11(b) is the control error of the three methods at the curvature curve. Due to the slow response time of fuzzy PID at the turning angle, the small change of steering angle leads to large error. Table 1 shows the average error of the three methods in each of the three scenarios. From the data in the table, it can be seen that the average error of the proposed method in all three scenarios is smaller than the fuzzy PID control error and variable-domain fuzzy PID control error.

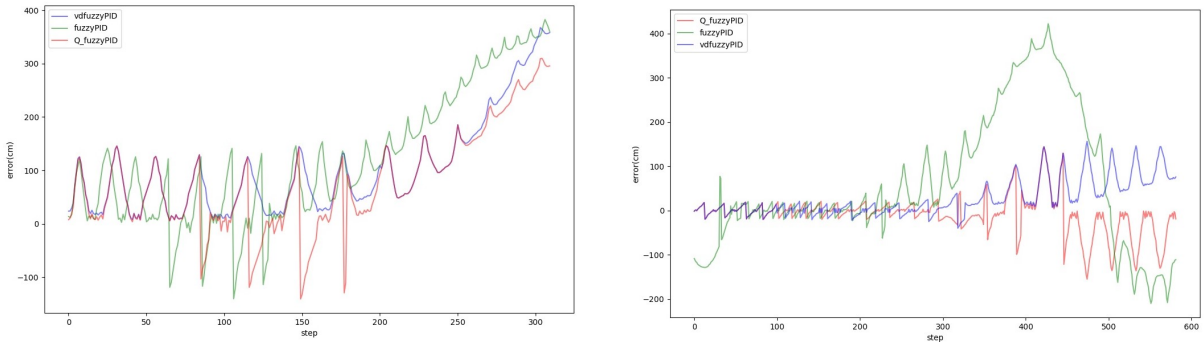


Figure 11. (a) is real-time control trajectory error of overtaking. (b) is real-time control trajectory error of curvature. The red line is the real-time control error of Q-fuzzyPID, the green line is the real-time control error of fuzzy PID, the blue line is the real-time control error of vdfuzzyPID.

Table 1. Comparison of the mean errors of the three methods for the three scenarios.

Scene	Method	Average Error
Follow	Q-Fuzzy PID	**9.41**
	Vdfuzzy PID	13.25
	Fuzzy PID	15.33
Overtaking	Q-Fuzzy PID	**244.16**
	Vdfuzzy PID	250.64
	Fuzzy PID	262.04
Curvature	Q-Fuzzy PID	**15.31**
	Vdfuzzy PID	35.76
	Fuzzy PID	117.73

4. Conclusions

Self-driving vehicles require different control accuracies when faced with different complex scenarios. Traditional PID control uses a set of parameters that are difficult to adapt to changes in the

scenario. The control function in a variational domain fuzzy PID controller will make the PID parameters self-tuning by constantly adjusting itself. However, the control function is reduced or enlarged with time, which can make the updated control function no longer accurate and generating distortion. In this paper, the Q-Learning algorithm is added to make the scaling factor have the ability to find the best online, so that the PID parameters can be adjusted more accurately and the optimal PID parameters can be found under the current working conditions. The effectiveness of the method is also verified on the Panosim simulation platform.

Acknowledgments

This work was supported by Beijing Natural Science Foundation (No.4222025), the National Natural Science Foundation of China (Nos. 61871038 and 61931012), Beijing Municipal Science & Technology Commission , Administrative Commission of Zhongguancun Science Park.

Conflict of interest

The authors declare there are no conflict of interest.

References

1. R. K. Khadanga, A. Kumar, S. Panda, Frequency control in hybrid distributed power systems via type-2 fuzzy pid controller, *IET Renewable Power Gener.*, **15** (2021), 1706–1723. https://doi.org/10.1049/rpg2.12140
2. M. K. Diab, H. H. Ammar, R. E. Shalaby, Self-driving car lane-keeping assist using pid and pure pursuit control, in *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, IEEE, (2020), 1–6. https://doi.org/10.1109/3ICT51146.2020.9311987
3. H. Maghfiroh, M. Ahmad, A. Ramelan, F. Adriyanto, Fuzzy-pid in bldc motor speed control using matlab/simulink, *J. Rob. Control (JRC)*, **3** (2022), 8–13. https://doi.org/10.18196/jrc.v3i1.10964
4. J. R. Nayak, B. Shaw, B. K. Sahu, K. A. Naidu, Application of optimized adaptive crow search algorithm based two degree of freedom optimal fuzzy pid controller for agc system, *Eng. Sci. Technol. Int. J.*, **32** (2022), 101061. https://doi.org/10.1016/j.jestch.2021.09.007
5. N. Ma, D. Li, W. He, Y. Deng, J. Li, Y. Gao, et al., Future vehicles: interactive wheeled robots, *Sci. China Inf. Sci.*, **64** (2021), 1–3. https://doi.org/10.1007/s11432-020-3171-4
6. N. Ma, Y. Gao, J. Li, D. Li, Interactive cognition in self-driving, *Chin. Sci.: Inf. Sci.*, **48** (2018), 1083–1096.
7. D. Li, N. Ma, Y. Gao, Future vehicles: learnable wheeled robots, *Sci. China Inf. Sci.*, **63** (2020), 1–8. https://doi.org/10.1007/s11432-019-2787-2
8. T. Yang, N. Sun, Y. Fang, Adaptive fuzzy control for a class of mimo underactuated systems with plant uncertainties and actuator deadzones: Design and experiments, *IEEE Trans. Cybern.*, **52** (2022), 8213–8226. https://doi.org/10.1109/TCYB.2021.3050475

9. S. H. Park, K. W. Kim, W. H. Choi, M. S. Jie, Y. Kim, The autonomous performance improvement of mobile robot using type-2 fuzzy self-tuning PID controller, *Adv. Sci. Technol. Lett.*, **138** (2016), 182–187. https://doi.org/10.14257/astl.2016.138.37
10. P. Parikh, S. Sheth, R. Vasani, J. K. Gohil, Implementing fuzzy logic controller and pid controller to a dc encoder motor—"a case of an automated guided vehicle", *Procedia Manuf.*, **20** (2018), 219–226. https://doi.org/10.1016/j.promfg.2018.02.032
11. Q. Bu, J. Cai, Y. Liu, M. Cao, L. Dong, R. Ruan, et al., The effect of fuzzy pid temperature control on thermal behavior analysis and kinetics study of biomass microwave pyrolysis, *J. Anal. Appl. Pyrolysis*, **158** (2021), 105176. https://doi.org/10.1016/j.jaap.2021.105176
12. M. S. Jie, W. H. Choi, Type-2 fuzzy pid controller design for mobile robot, *Int. J. Control Autom.*, **9** (2016), 203–214.
13. N. Kumar, M. Takács, Z. Vámossy, Robot navigation in unknown environment using fuzzy logic, in *2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, (2017), 279–284. https://doi.org/10.1109/SAMI.2017.7880317
14. T. Muhammad, Y. Guo, Y. Wu, W. Yao, A. Zeeshan, Ccd camera-based ball balancer system with fuzzy pd control in varying light conditions, in *2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC)*, IEEE, (2019), 305–310. https://doi.org/10.1109/ICNSC.2019.8743305
15. A. Wong, T. Back, A. V. Kononova, A. Plaat, Deep multiagent reinforcement learning: Challenges and directions, *Artif. Intell. Rev.*, **2022** (2022). https://doi.org/10.1007/s10462-022-10299-x
16. Z. Cao, S. Xu, H. Peng, D. Yang, R. Zidek, Confidence-aware reinforcement learning for self-driving cars, *IEEE Trans. Intell. Transp. Syst.*, **23** (2022), 7419–7430. https://doi.org/10.1109/TITS.2021.3069497
17. T. Ribeiro, F. Gonçalves, I. Garcia, G. Lopes, A. F. Ribeiro, Q-learning for autonomous mobile robot obstacle avoidance, in *2019 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, IEEE, (2019), 1–7. https://doi.org/10.1109/ICARSC.2019.8733621
18. S. Danthala, S. Rao, K. Mannepalli, D. Shilpa, Robotic manipulator control by using machine learning algorithms: A review, *Int. J. Mech. Prod. Eng. Res. Dev.*, **8** (2018), 305–310.
19. X. Lei, Z. Zhang, P. Dong, Dynamic path planning of unknown environment based on deep reinforcement learning, *J. Rob.*, **2018** (2018). https://doi.org/10.1155/2018/5781591
20. Y. Shan, B. Zheng, L. Chen, L. Chen, D. Chen, A reinforcement learning-based adaptive path tracking approach for autonomous driving, *IEEE Trans. Veh. Technol.*, **69** (2020), 10581–10595. https://doi.org/10.1109/TVT.2020.3014628
21. T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, et al., Continuous control with deep reinforcement learning, preprint, arXiv:1509.02971. https://doi.org/10.48550/arXiv.1509.02971
22. P. Ramanathan, K. K. Mangla, S. Satpathy, Smart controller for conical tank system using reinforcement learning algorithm, *Measurement*, **116** (2018), 422–428. https://doi.org/10.1016/j.measurement.2017.11.007

23. L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, et al., Safe learning in robotics: From learning-based control to safe reinforcement learning, *Annu. Rev. Control Rob. Auton. Syst.*, **5** (2022), 411–444. https://doi.org/10.1146/annurev-control-042920-020211
24. A. I. Lakhani, M. A. Chowdhury, Q. Lu, Stability-preserving automatic tuning of PID control with reinforcement learning, preprint, arXiv:2112.15187. https://doi.org/10.20517/ces.2021.15
25. O. Dogru, K. Velswamy, F. Ibrahim, Y. Wu, A. S. Sundaramoorthy, B. Huang, et al., Reinforcement learning approach to autonomous pid tuning, *Comput. Chem. Eng.*, **161** (2022), 107760. https://doi.org/10.1016/j.compchemeng.2022.107760
26. X. Yu, Y. Fan, S. Xu, L. Ou, A self-adaptive sac-pid control approach based on reinforcement learning for mobile robots, *Int. J. Robust Nonlinear Control*, **32** (2022), 9625–9643. https://doi.org/10.1002/rnc.5662
27. B. Guo, Z. Zhuang, J. S. Pan, S. C. Chu, Optimal design and simulation for pid controller using fractional-order fish migration optimization algorithm, *IEEE Access*, **9** (2021), 8808–8819. https://doi.org/10.1109/ACCESS.2021.3049421
28. M. Praharaj, D. Sain, B. Mohan, Development, experimental validation, and comparison of interval type-2 mamdani fuzzy pid controllers with different footprints of uncertainty, *Inf. Sci.*, **601** (2022), 374–402.
29. Y. Jia, R. Zhang, X. Lv, T. Zhang, Z. Fan, Research on temperature control of fuel-cell cooling system based on variable domain fuzzy pid, *Processes*, **10** (2022), 534. https://doi.org/10.3390/pr10030534
30. J. Wei, L. Gang, W. Tao, G. Kai, Variable universe fuzzy pid control based on adaptive contracting-expanding factors, *Eng. Mech.*, **38** (2021), 23–32. https://doi.org/10.6052/j.issn.1000-4750.2020.11.0786
31. R. S. Sutton, A. G. Barto, *Reinforcement Learning: An Introduction*, MIT press, 2018.
32. P. R. Montague, Reinforcement learning: an introduction, by Sutton, RS and Barto, AG, *Trends Cognit. Sci.*, **3** (1999), 360. https://doi.org/10.1016/S1364-6613(99)01331-5
33. D. Wang, R. Walters, X. Zhu, R. Platt, Equivariant q learning in spatial action spaces, in *Conference on Robot Learning*, PMLR, (2022), 1713–1723.
34. E. Anderlini, D. I. Forehand, P. Stansell, Q. Xiao, M. Abusara, Control of a point absorber using reinforcement learning, *IEEE Trans. Sustainable Energy*, **7** (2016), 1681–1690. https://doi.org/10.1109/TSTE.2016.2568754

AIMS Press