

Design of Improved PID Controller Based on PSO-GA Hybrid Optimization Algorithm in Vehicle Lateral Control

Shenqi GAO, Song GAO, Weigang PAN*, Mushu WANG

School of Information Science and Electrical Engineering, Shandong Jiaotong University,
No. 5001, Haitang Road, Jinan, 250357, Shandong, China
feize0624@163.com, gaosong8286@mail.sdu.edu.cn,
panweigang1980@163.com (*Corresponding author), wangmushu@126.com

Abstract: Considering that it is difficult for a single PID controller to adapt to different vehicle speeds in the context of vehicle path lateral tracking control, this paper proposes a segmented fuzzy PID controller based on particle swarm optimization (PSO) and on the genetic algorithm (GA), namely the hybrid optimization algorithm PCAG. Firstly, the vehicle speed is divided into several intervals, and different PID controller parameters are used for each interval. Secondly, in order to reduce the overshoot and stabilization time, the proposed PCAG algorithm is employed, which is a combination of PSO, particle swarm optimization with convergence factor (PSO-CF), adaptive particle swarm optimization (APSO) and GA. Further on, the PID controller parameters of different speed ranges are adjusted through this algorithm. Finally, in order to make up for the shortcomings of a single PID controller in the context of time-varying vehicle speed control, a fuzzy controller is employed with the purpose of compensating the parameters of the PID controller, so that the controller could adapt to a wider range of vehicle speeds. The simulation results show that the convergence speed and optimization ability of the proposed PCAG are higher than those of PSO. In addition, the segmented fuzzy PID controller optimized by PCAG can adapt to different vehicle speeds and features an excellent path tracking accuracy.

Keywords: Segmented PID, Fuzzy control, Particle swarm optimization, Genetic algorithm, PCAG.

1. Introduction

Autonomous driving as an important display platform for artificial intelligence, Internet of Things, automatic control and other technologies, is the main research focus in current vehicle engineering. The direction and difficulty of the research is related to the fact that the vehicle can replace the driver to complete the driving task, and to ensure safety and comfort during driving. With the development of 5G technology, the delay of wireless communication is further reduced, and the task scheduling management technology based on edge-cloud synergy is gradually being used in automatic driving to accomplish information sharing and precise control of vehicles in automatic driving. In the research of autonomous driving, the problem of vehicle lateral control has always been one of the difficulties in the research, owing to the impact of time-varying speed, high-speed operation, and the vehicle's own conditions, it was often impossible to achieve the expected control effect using conventional controllers. Therefore, the research on vehicle lateral control has important theoretical significance and applicative value. The main research contents include the controller design (Amer et al., 2019; Nie et al., 2020), controller parameter tuning and the application of intelligent algorithms in controller parameter tuning (He et al., 2020).

At present, the commonly used methods in vehicle lateral control and path tracking control include PID control, fuzzy control, robust control, linear quadratic regulator, model predictive control (Amer et al., 2019), and so on. Among them, PID control has been extensively applied to a variety of control scenarios due to its simplicity and reliability, but in the case of conventional fixed parameter PID control it is usually hard to accomplish the expected control effect for different vehicle speeds and in a dynamic and complex environment.

Therefore, in order to improve the condition of insufficient control ability of fixed-parameter PID controllers in the case of dynamic and complex situations, many researchers have proposed improved PID controllers. In (Zhao et al., 2012), an adaptive-PID was employed in order to reduce the instantaneous error and improves the tracking accuracy when the vehicle is under a wide range of parameter changes and disturbances. In (Han et al., 2017), the neural network PID was used for the lateral path tracking control of an intelligent vehicle. In (Mohan et al., 2019), a two-degree-of-freedom fractional-order fuzzy PID controller based on expert intelligence was proposed and used for trajectory tracking. In (Huang et al., 2019), researchers classified the working environment of electronic vehicles into seven typical types, and

established a sub-model set to describe them, and then constructed a multi-model control system based on the BP-PID controller, which is designed for the lateral control of electronic vehicles.

However, when solving the problem of vehicle lateral control, due to the fact that the vehicle dynamic model is highly affected by the environment, it is often difficult for the adaptive PID based on this model to determine the reference model. PID based on expert experience requires a lot of prior knowledge to build an accurate knowledge base. However, when a vehicle is driving under unknown conditions, it is difficult to obtain comprehensive prior knowledge. For PID controllers based on neural networks, if the network is simple, the control effect will be poor when faced with complex situations; if the network is complex, the amount of calculation will be greatly increased.

In practical applications, except for the design of the controller, tuning of the controller parameters is also important. The Ziegler-Nichols method is often used for PID parameter tuning, but this tuning method pays more attention to noise suppression, and the parameters obtained by using this method often lead to large overshoots. In order to obtain better control parameters, intelligent algorithms such as PSO and GA were widely used in parameter tuning of PID controllers (Rodríguez-Molina et al., 2020).

In (Allou & Zennir, 2018), two PID controllers were used for vehicle trajectory tracking, and through PSO to tuning PID parameters. In (Zhao et al., 2019), GA was used for tuning PID parameters. In (Prasad et al., 2021), a non-dominated Sorting GA was designed with a view to optimizing vehicles' PID controller. In (Xu, Wang & Zhang 2021), a PI controller based on proportional-integral-radius and improved PSO controller, was designed in order to reduce the overshoot of unmanned articulated vehicle and improve the path tracking accuracy. In (Qiao et al., 2020), an improved PSO was employed for optimizing PID controller parameters, and the optimized controller was used for vehicle lateral control. In (Wang et al., 2019), the adaptive fuzzy PID controller based on segmented PID was designed and an improved PSO was used for optimizing the parameters.

When PID parameters are optimized by a single intelligent algorithm (He et al., 2020), it is

often due to inherent defects of that intelligent algorithm, or to the fact that the population converges slowly, or the population converges prematurely, which results in the failure to converge to the global optimum, and affects the control effect (Tam et al., 2019). Therefore, in order to obtain a better control effect, many researchers consider the use of improved intelligent algorithms (Xiang et al., 2020, Li et al., 2020) for adjusting the controller parameters in order to achieve better control effects.

In order to allow the controller to adapt to different vehicle speeds and reduce the dependence on accurate models, a segmented fuzzy PID controller is designed in this paper. Firstly, the vehicle speed is divided into several intervals, different PID parameters are used in each interval, these parameters are compensated by a fuzzy controller, and the vehicle steering angle of the front wheel is calculated to control the steering of the vehicle. Secondly, PCAG algorithm is used for optimal parameter tuning of segmented PID controller in each speed range and for making each controller reduce the overshoot and stabilization time as much as possible. Finally, the simulation results show the convergence ability, global optimization of PCAG algorithm and the effect of controller path tracking at different vehicle speeds.

The remainder of this paper is structured as follows. The vehicle kinematic model is presented in Section 2. The controller structure is discussed in Section 3. Section 4 presents the PCAG algorithm. Section 5 presents the fuzzy controller. Section 6 sets forth the simulation results and the conclusion is provided in Section 7.

2. Vehicle Kinematic Model

A vehicle kinematic model established based on the geodetic coordinate system XOY describes the movement state of the vehicle in the process of traveling, as it is shown in Figure 1. The following assumptions are made in this paper to simplify the model:

1. The influence of the steering mechanism is ignored, the two front and rear wheels have the same law of motion. The front and rear wheels are described by one wheel each;
2. The movement in the vertical direction is ignored and only the lateral movement of the vehicle is considered.

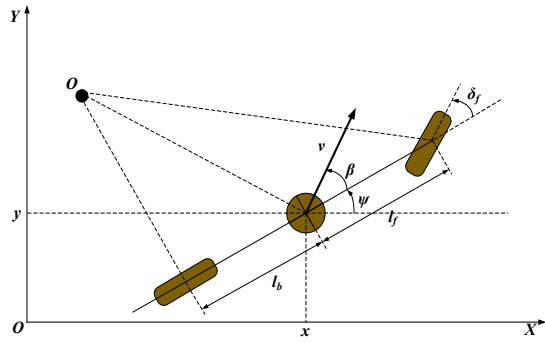


Figure 1. Schematic figure of vehicle kinematic model

According to the theorem of kinematics, the update iteration equations for each state quantity of the vehicle is as follows:

$$x_{t+1} = x_t + v_t \cos(\psi_t + \beta) \times dt \quad (1)$$

$$y_{t+1} = y_t + v_t \sin(\psi_t + \beta) \times dt \quad (2)$$

$$\psi_{t+1} = \psi_t + (v_t / l_b) \times \sin(\beta) \times dt \quad (3)$$

$$v_{t+1} = v_t + a \times dt \quad (4)$$

where β can be obtained from equation (5).

$$\beta = \tan^{-1} \{ [l_b / (l_f + l_b)] \times \tan(\delta_f) \} \quad (5)$$

x and y are the current vehicle coordinates. a is the vehicle acceleration. ψ is the angle between the vehicle body and the x-axis. v is the vehicle linear velocity. β is the tire slip angle. δ_f is the angle between the wheel and the vehicle body. l_f and l_b are the distance to vehicle body center from the front and rear wheels. O is the steering center, and dt is the control cycle.

3. Controller Structure Design

When using a single PID controller it is usually hard to achieve an excellent control effect when vehicle speed varies over a wide range. The use of a segmented PID controller can increase the speed adaptability, but when the controller parameters are not selected properly and the PID parameters of two different speed ranges are frequently switched, the discontinuous characteristics of the segmented PID controller may make the system unstable (Wang et al., 2019). Therefore, PCAG algorithm is designed for optimal parameter tuning to make the PID parameters of each speed range have the smallest possible overshoot and stabilization time, and a fuzzy controller is used for parameter compensation in order to increase the length of the speed division interval, reduce the frequency of PID parameter switching, and reduce the oscillation caused by segmented PID parameter switching.

The controller system is shown in Figure 2. In this controller, the input value is the deviation between the vehicle and the path. PCAG algorithm is used for offline tuning to obtain the segment PID parameter table for the PID controller, and according to the vehicle speed v , the switcher is used for determining which set of PID parameters is currently used. The pseudo-code of the switcher is expressed in equation (7). The input of fuzzy controller consists in vehicle speed v and acceleration a , and the output of fuzzy controller is used as an increment, which is added to the output parameters by the segmented PID parameter table to act on the PID controller in order to obtain the final steering angle of the vehicle and act on the vehicle controller.

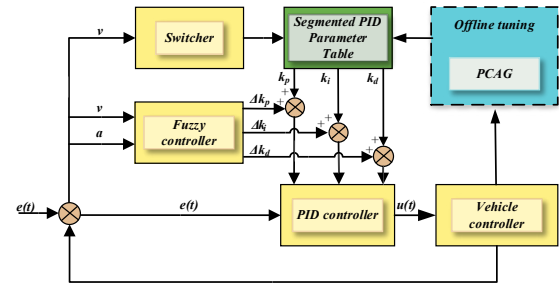


Figure 2. Controller structure

$$u(t) = (k_p + \Delta k_p) \times e(t) + (k_i + \Delta k_i) \times \sum_{i=0}^t e(t) + (k_d + \Delta k_d) \times [e(t) - e(t-1)] \quad (6)$$

where $u(t)$ is the PID output, k_p , k_i , k_d are the parameters of segmented PID determined by the switcher, and Δk_p , Δk_i , Δk_d are the output parameters of fuzzy controller. $e(t)$, $e(t-1)$ represent the vehicle error of current time and last time, respectively.

```

for i = 1 : num
    if (v >= range[i] && v < range[i+1])
        return k_p, k_i, k_d
    end
end

```

(7)

where v is the vehicle speed, num is number of segments, $range$ is vehicle speed range.

4. PSO-GA Hybrid Optimization Algorithm

Because there are many parameters of the controller which need to be tuned, it is difficult to adjust them with conventional methods, and the optimal control effect may not be obtained.

Therefore, intelligent algorithms are employed for adjusting the controller parameters. As a commonly used intelligent algorithm, PSO has the characteristics of low time complexity and fast convergence, many researchers use it for the tuning of various controller parameters (Amiri, Ramli & Ibrahim, 2019).

PSO has been used in many cases, but it has low diverse particle, a low global search ability, and can easily to fall into a situation of premature local optimality (Lalwani et al., 2019). GA has a diverse population, strong global search ability, but individual particles in population have no memory, genetic manipulation is blind and directionless, it requires a long convergence time (Freire, Moura Oliveira & Solteiro Pires, 2017; Koessler & Almomani, 2020). Therefore, in the research related to the improved PSO, GA is often combined with PSO (Dziwiński & Bartczuk 2019). In comparison with each of them, the algorithm which results from their combination has the advantages of a better global optimization and convergence ability (Cheng et al., 2018; Farnad, Jafarian & Baleanu, 2018).

The combination of PSO and GA includes parallel hybrid, serial hybrid, embedded hybrid systems, etc (Wang et al., 2018; Roshanzamir et al., 2020). Among them, the serial hybrid performance is restricted by the sequence of the two algorithms and the switching point of the two algorithms, while embedded hybrid performance is related to the embedding mode of the two related algorithms. Parallel hybrid makes full use of the performance of PSO and GA, has a wide search range and faster convergence, which is usually used in the case of parameter optimization (Subotic, Manasijevic, & Kupusinac, 2020).

Therefore, a new PSO-GA hybrid optimization algorithm named PCAG is presented in this paper, which is based on PSO, PSO-CF, APSO and GA.

4.1 PSO and GA

PSO treats each target as a “particle”, each particle has attributes of current position, flight speed and individual optimal fitness. The current position of particles is represented by $X_i = (X_{i1}, X_{i2}, X_{i3}, \dots, X_{in})$, and flying speed of particles is described by $V_i = (V_{i1}, V_{i2}, V_{i3}, \dots, V_{in})$. Among them, X_{in} and V_{in} represent the position and speed of i-th particle in the n-th dimension. When the algorithm is initialized, a random position

X_i and speed V_i are assigned to each particle, and the position coordinates of particles are used as input values to calculate the fitness f . After that, the speed and position update are started, the speed and position update rules are shown in equation (8).

$$\begin{aligned} V_{id}(t+1) &= \omega V_{id}(t) + c_1 \times rand \times (P_{id} - X_{id}(t)) \\ &\quad + c_2 \times rand \times (P_{gd} - X_{id}(t)) \\ X_{id}(t+1) &= X_{id}(t) + V_{id}(t+1) \end{aligned} \quad (8)$$

where $rand$ is a number randomly obtained in 0~1, c_1 and c_2 are self-learning factor and social learning factor, ω is inertia weight, P_{gd} is the global best fitness of all particles, P_{id} is the best fitness of the i-th particle individual. After the speed and position are updated, the best fitness of individual particles and global fitness of the particles are updated according to the fitness function, and then one iteration is completed at this time. After the set number of iterations is completed or the convergence condition is met, the optimization ends.

PSO-CF (Houssein et al., 2021) and APSO (Li et al., 2019) are improved PSO algorithms. PSO-CF adds a convergence factor based on PSO algorithm, so that the algorithm can converge quickly. The PSO-CF velocity update rules are expressed in equations (9) and (10), position update rules are the same as for PSO.

$$\begin{aligned} V_{id}(t+1) &= \lambda(V_{id}(t) + c_1 \times rand \times (P_{id} - X_{id}(t)) \\ &\quad + c_2 \times rand \times (P_{gd} - X_{id}(t))) \end{aligned} \quad (9)$$

$$\begin{aligned} \lambda &= 2 / |2 - C - \sqrt{C^2 + 4C}| \\ C &= c_1 + c_2 \end{aligned} \quad (10)$$

where λ is convergence factor, and C is the sum of self-learning factor and social learning factor.

APSO uses the Sugeno function as the inertial weight decreasing curve, and global search and local search can be automatically adjusted. It reduces the possibility that the algorithm falls into local optimization. APSO velocity update rules are expressed in equations (11) and (12), position update rules are the same as for PSO.

$$\begin{aligned} V_{id}(t+1) &= \omega V_{id}(t) + c_1 \times rand \times (P_{id} - X_{id}(t)) \\ &\quad + c_2 \times rand \times (P_{gd} - X_{id}(t)) \end{aligned} \quad (11)$$

$$\omega = 1 - (t/T) / 1 - s(t/T) \quad (12)$$

where s is a constant greater than -1, T is the total number of iterations, and t is the current number of iterations.

PSO-CF improves the convergence ability of PSO, but it is easier for the particles to converge prematurely. APSO uses the method of dynamic weight to adaptively adjust global and local search, but it is easy to cause particles to diverge. Therefore, when choosing an algorithm, these algorithms are used to complement each other.

GA through simulated natural selection, heredity, replication and cross variation, keeps the highly adaptable individuals to form a new group. The new group inherits the information of its parents, and converges to the global optimum more easily. The steps of GA are as follows:

1. Generate individual populations randomly, and calculate the fitness of each individual;
2. Determine whether individuals in the population are inherited or eliminated according to fitness;
3. Take the individuals selected for inheritance as the parents, the partial code values of the two individuals are exchanged to obtain a new individual;
4. According to the mutation probability, randomly change the code value of the individual to obtain a new individual after mutation.

4.2 PCAG Algorithm

PCAG algorithm combines the characteristics of PSO, PSO-CF, APSO and the GA. Based on these four algorithms, it optimizes and expands the scope of particle search, so that each population can share the global optimal information with other populations. After every iteration, each population uses the global optimal particles of other populations to replace the worst particles it includes, so as to ensure that the population continues to move in the optimal direction. The PCAG algorithm uses the fast convergence feature of PSO to allow the population to converge quickly, while utilizing the strong search ability of GA to make the population converge to the global optimum. So it can quickly converge and effectively avoid the situation of local optimality.

PCAG algorithm uses equation (13) as the definition of fitness function to evaluate the pros and cons of each particle and of the parameters. The parameters found by PCAG algorithm with the smallest fitness are the optimal parameters.

$$J = \int_0^{\infty} t |e(t)| dt \quad (13)$$

where t is the current moment, and $e(t)$ is the vehicle error at the current moment.

The steps of the PCAG algorithm are as follows:

1. Determine the fitness function, determine the population size, number of iterations of each intelligent algorithm, initialize the location of the population, fitness, etc.;
2. According to the update rules of the four selected algorithms, update the corresponding parameters of each particle, calculate the fitness and update individual best fitness and global best fitness;
3. Replace the global worst particles of a population with the global optimal particles of other populations;
4. Determine whether the condition of convergence or the number of iterations is reached, if this condition is satisfied, go to step 5, else go to step 2;
5. The iteration is completed and the global optimum is obtained.

The PCAG algorithm flow chart is illustrated in Figure 3:

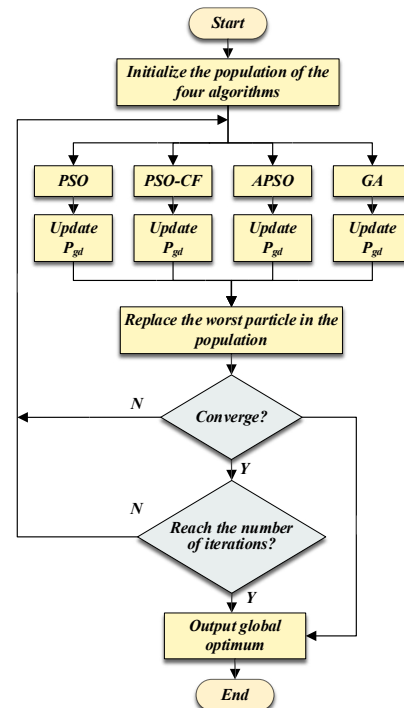


Figure 3. PCAG algorithm flow chart

The pseudo-code of PCAG algorithm is as follows:

```

Step1:
initialize populations of PSO, PSO-CF, APSO and GA
Step2:
for i = 1 : 4
    for i = 1 : n // n is the number of particles
        update particles  $V_{id}$  and position  $X_{id}$  and fitness
    end
end
Step3:
//find the worst particle for each algorithm
PWorst[:, :] = findWorst(particles)
//find the best particle for each algorithm
PBest[:, :] = findBest(particles)
//Replace the individuals with the worst of the other
//algorithms with the individual with the best of
//each algorithm
for i = 1 : 4
    ReplaceParticles(particles[i,:], PBest[i,:], PWorst[i,:])
end
Step4: Until a terminate condition is met,
or else, returns to step2
Step5: output global best

```

5. Fuzzy Controller Design

In order to reduce the oscillation of segmented PID during parameter switching and expand the applicable range of parameters, a fuzzy controller is employed with the purpose of performing parameter compensation on the basis of the segmented PID.

5.1 Fuzzification

Take the vehicle speed v and acceleration a as the input of fuzzy control, where the domain of v is

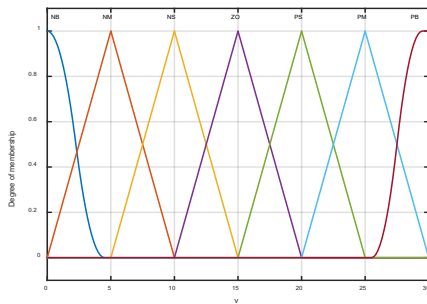


Figure 4. The membership function of v

[0,30], and the domain of a is [0,3], the output of fuzzy control is represented by PID parameters Δk_p , Δk_i , Δk_d . The quantization level is set to 7, namely Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (ZO), Positive Small (PS), Positive Medium (PM), Positive Big (PB). As the membership function is concerned, triangle and S-shaped membership functions were adopted, in which context, NB and PB use S-shaped membership functions, in order to respond quickly when there is a big error. NM, NS, ZO, PS and PM adopt the triangular membership function with high resolution and sensitivity. The membership functions diagrams for v and a are shown in Figures 4 and 5, respectively.

5.2 Fuzzy Rules

Fuzzy rules have a very obvious influence on control effect. This paper uses “if-then” rules, 49 rules are set for each parameter, that is k_p , k_i , k_d . The specific rules are shown in Table 1, from left to right, they are fuzzy rules of k_p , k_i , k_d .

5.3 Fuzzy Reasoning and Defuzzification

The fuzzy reasoning method used in this paper is Mamdani method, it is essentially a synthetic

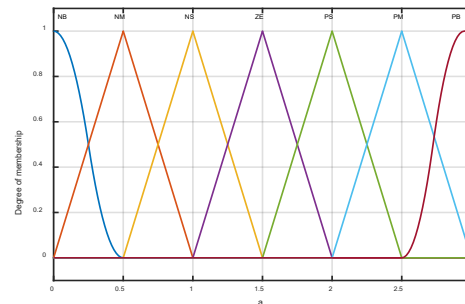


Figure 5. The membership function of a

Table 1. Fuzzy rules for k_p , k_i , k_d

v	a						
	NB	NM	NS	ZO	PS	PM	PB
NB	PB/NB/PS	PM/NM/NS	PM/NS/NB	PM/ZO/NB	PM/PS/NB	PS/PM/NM	ZO/PB/PS
NM	PM/PB/PS	PM/PM/NS	PS/PS/NB	PS/NB/NM	PS/NM/NM	ZO/NS/NS	NS/ZO/ZO
NS	PM/PM/ZO	PS/PS/NS	PS/PS/NM	PS/NM/NM	ZO/PS/NS	NS/ZO/NS	NS/NS/ZO
ZO	PM/PS/ZO	PM/PS/NS	PS/PS/NS	ZO/NS/NS	NS/ZO/NS	NS/NS/NS	NS/NM/ZO
PS	PS/NM/ZO	PS/NS/ZO	ZO/NS/ZO	PS/ZO/ZO	NS/PS/ZO	NS/PS/ZO	NM/PS/ZO
PM	PS/NM/PB	ZO/NS/NS	PS/ZO/PS	PM/NS/PS	NS/PS/PS	NM/PS/PS	NM/PM/PB
PB	ZO/NS/PB	PS/ZO/PM	PM/PS/PM	PB/NM/PM	NM/PS/PS	NM/PS/PS	NB/PM/PB

reasoning method, namely R_i : if v is E_j and a is E_k , then u is U_{jk} , where R_i represents the i -th control rule, and E_j, E_k represent the j row and the k column respectively. After fuzzy reasoning, a fuzzy set can be obtained, but the fuzzy set cannot be directly used in the controller and needs to be converted into an accurate value that the controller can use. This process is called defuzzification. In this paper, the defuzzification adopts the center of gravity method, as it is shown in equation (14).

$$u = \frac{\sum_{i=1}^m \mu(u_i)u_i}{\sum_{i=1}^m \mu(u_i)} \quad (14)$$

where m represents the number of quantization level, which is 7 in this paper. u_i represents the value in the range of fuzzy controller, $\mu(u_i)$ represents the membership value of u_i , and u is the final output value after defuzzification.

Through fuzzy reasoning and defuzzification, the fuzzy control output parameters can be finally obtained.

6. Simulation Experiments

In this section, several simulation experiments are conducted to display the superior capability of the proposed controller. In all simulation experiments, the sampling time is set at 0.1s.

Case 1. To verify the optimization ability of the PCAG algorithm, a comparative experiment of optimization and vehicle path tracking was performed.

Based on the vehicle model proposed in this paper, PSO, PSO-CF, APSO, GA, and PCAG algorithms are used for optimizing. The population size for all algorithms is 50, and the number of iterations is 200. The initial deviation of the vehicle is set at 0.5m, the vehicle speed at 5m/s and 20m/s, and there is no acceleration.

Equation (13) is used to calculate the fitness of each algorithm, so as to compare the convergence ability and optimization ability of each algorithm. The fitness convergence results related to optimization effects are shown in Figures 6 and 7.

Figure 6 shows the optimization effects of each algorithm when the vehicle speed is set at 5m/s. It can be noticed that at the 13th iteration, the

fitness of PCAG algorithm is already better than that of the other algorithms and it finally converged at 59th iteration, and its optimal fitness is 1.2. By contrast, GA has the minimum fitness among the selected algorithms, and it converges at the 14th.

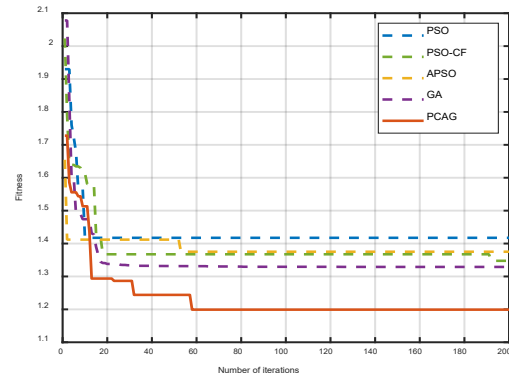


Figure 6. Comparison of optimization effects at 5m/s

As it is illustrated in Figure 7, when the vehicle speed is 20m/s, the PCAG algorithm converges at the 22th iteration, and its optimal fitness is 2.58. However, GA has the minimum fitness among the selected algorithms, which is 2.61, and it converges at the 111th iteration.

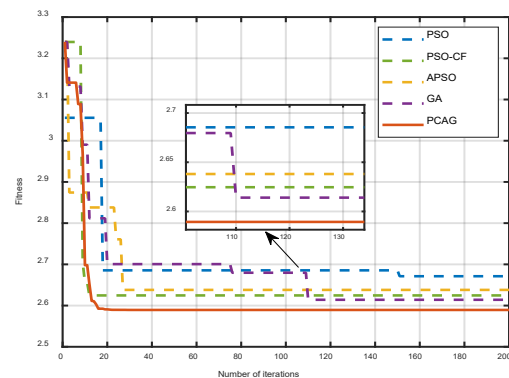


Figure 7. Comparison of optimization effects at 20m/s

It can be seen that because the PCAG algorithm shares the optimal particles of the four populations related to the other algorithms, in comparison with any single algorithm, it improves the global search ability, and it is easier to avoid the local optimality and find better results.

Figures 8-9 respectively show the comparison of vehicle path tracking deviation effects when using PSO, PSO-CF, APSO, GA, and PCAG to adjust PID controller parameters under different

set speeds. As demonstrated in Figure 8, when the set speed is 5m/s, the maximum overshoot of the vehicle is 0.1m and adjustment time is 7s in the case of using the parameters adjusted by PCAG algorithm, whereas when using the parameters tuned by GA the minimum overshoot is 0.23m and the adjustment time is 13s.

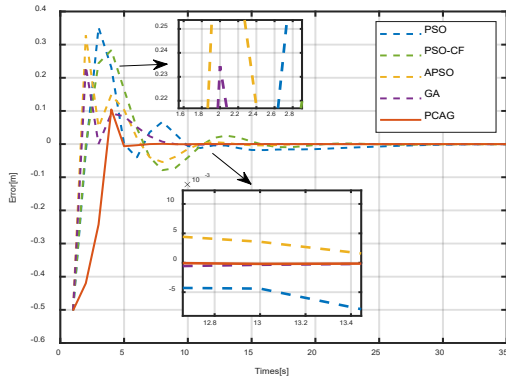


Figure 8. Comparison of control effects at 5m/s

As it is shown in Figure 9, when the set speed is 20m/s and the PCAG algorithm is used to adjust the parameters, the maximum overshoot of the vehicle is 0.4m and adjustment time is 28s, while when using the parameters tuned by PSO the minimum overshoot is 0.52m and when using the parameters tuned by APSO the minimum adjustment time is 31.2s.

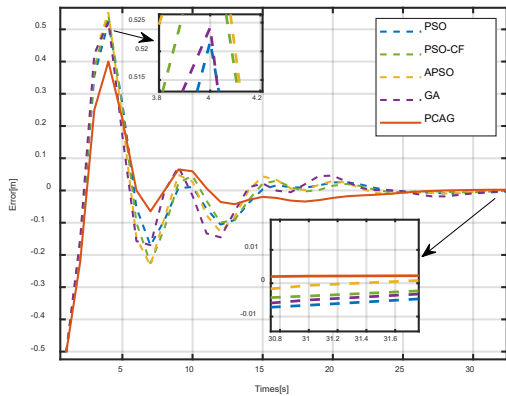


Figure 9. Comparison of control effects at 20m/s

It is clear that because the optimization effect of PCAG is better than that of the other algorithms, when adopting the parameters adjusted by PCAG algorithm, the vehicle path tracking control system has a smaller overshoot and adjustment time.

Case 2. To verify the control effect at constant speed for the segmented PID and fuzzy controller, the comparative experiment of vehicle path tracking at constant speed was performed.

Based on the vehicle model proposed in this paper, the PCAG algorithm is used for tuning the optimal parameters of segmented PID at constant vehicle speeds of 10m/s, 15m/s, and 20m/s. Segmented PID and segmented fuzzy PID controller are used to conduct the vehicle lateral path tracking and the vehicle path tracking effects under the two controllers are compared.

Figures 10 to 12 respectively show the comparison of the effect of segmented PID and segmented PID + fuzzy control at different constant speeds. After adding fuzzy control for parameter compensation, when the set speed is 10m/s, the path tracking error is reduced from 0.31m~0.11m to 0.3m~-0.059m, and the adjustment time is reduced from 16s to 10s. When the set speed is 15m/s, the path tracking error is reduced from 0.5m~-0.18m to 0.42m~-0.1m, and the adjustment time is reduced from 20s to 17s. When the set speed is 20m/s, the path tracking error is reduced from 0.49m~-0.18m to 0.42m~-0.12m, and the adjustment time is reduced from 27s to 20s.

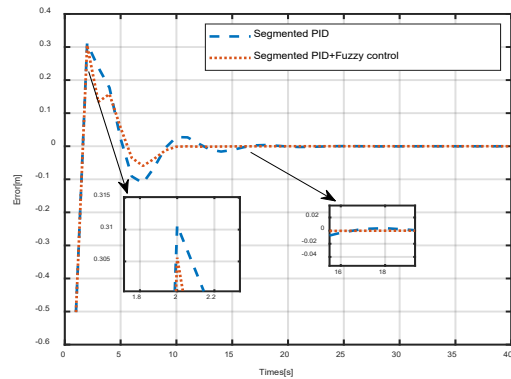


Figure 10. Comparison of control effects at 10m/s

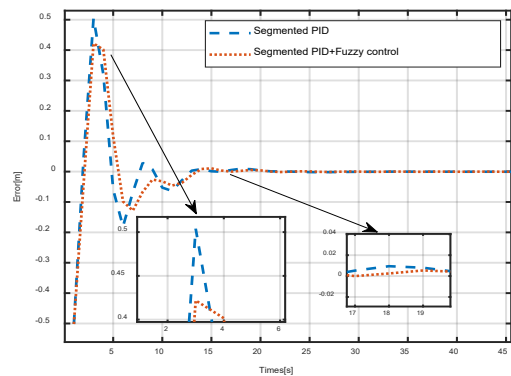


Figure 11. Comparison of control effects at 15m/s

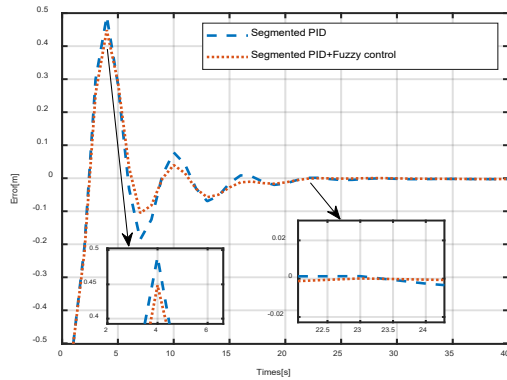


Figure 12. Comparison of control effects at 20m/s

It can be noticed that due to the use of fuzzy control for parameter compensation, the overshoot of vehicle is reduced, and the vehicle can be adjusted to near predetermined position faster at a constant speed.

Case 3. To verify the control effect at variable speed of the segmented fuzzy PID controller, a comparative experiment of vehicle path tracking at variable speed was performed.

The PCAG algorithm is used for tuning the optimal parameters of segmented PID at variable vehicle speeds of 0-10m/s, 0-20m/s acceleration and deceleration.

Figure 13 shows path tracking error in the case of variable speed. Let the speed be accelerated from 0 m/s to 10m/s, and then decelerated from 10m/s to 0m/s. Using segmented PID control, after the vehicle is stabilized, the error is within the range of ± 0.1 m. After adding fuzzy control, the error of the vehicle after stabilization is within the range of ± 0.05 m.

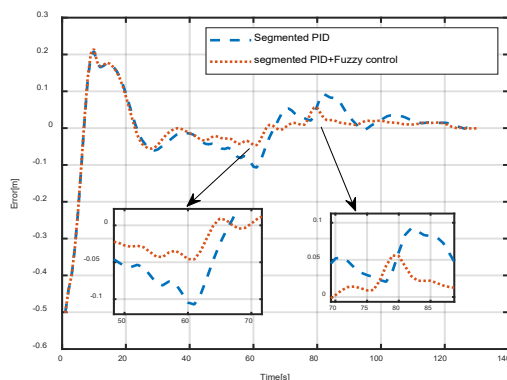


Figure 13. Comparison of control effects when the vehicle is accelerating

In the case of acceleration and deceleration, only segmented PID control can be used. Due to continuous switching of controller parameters, the control is unstable when the vehicle speed changes and the vehicle control accuracy is reduced. However, the use of fuzzy control for parameter compensation improves the continuity of parameters and improves the stability and accuracy of vehicle control.

As it is illustrated in Figure 14, when the vehicle accelerates to 20m/s, the use of segmented PID and fuzzy control can effectively suppress the oscillation of the vehicle. When the speed is high, the vehicle vibration decreases from 0.1~0.05m to ± 0.01 m.

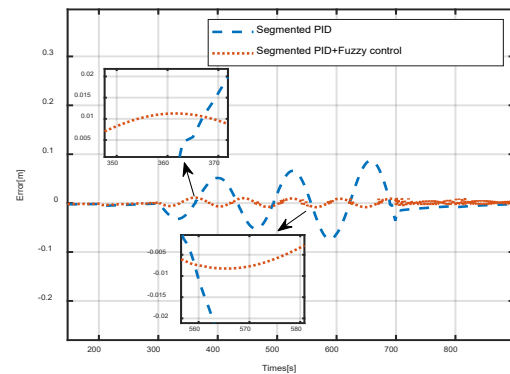


Figure 14. Comparison of vibration suppression

When the vehicle is running at a high speed, the control discontinuity caused by frequent switching of control parameters is more obvious. If the parameters are set improperly, it is easy to cause the system to diverge. After adding fuzzy control for parameter compensation, it can effectively reduce the vibration of the vehicle under high-speed path tracking.

Based on the experiments above, it can be noticed that, compared with PSO, PSO-CF, APSO and GA, PCAG has the advantages of faster convergence speed, stronger global optimization ability, and smaller parameter overshoot, along with a higher control accuracy after the controller uses it. Regardless of the different vehicle speeds or acceleration/deceleration conditions, in comparison with segmented PID controller, the segmented fuzzy PID controller can effectively reduce the vehicle lateral tracking deviation, improve the accuracy of vehicle lateral tracking, and effectively suppress the oscillation caused by segmented PID parameter switching.

7. Conclusion

This paper takes the vehicle kinematic model as the research object, focusing on the problem that the vehicle path lateral tracking control is difficult to adapt to different vehicle speeds. A segmented fuzzy PID controller is proposed, and PCAG algorithm is used for optimizing the segmented PID parameters. On the simulation platform built by means of MATLAB, the lateral control simulation experiments for the above-mentioned control algorithm were carried out for different vehicle speeds. The experimental results show that the proposed controller can effectively control the vehicle to track the vehicle path under the conditions of constant speed, acceleration and deceleration, and improve the control accuracy.

However, the method proposed in this paper also has some shortcomings. PCAG algorithm

uses multi-intelligent algorithms for parallel optimization, which increases the time complexity. The proposed controller uses a fixed global fuzzy controller for the compensation of segmented PID parameters, but once the speed range is too large, the control effect would decrease. Future works could focus on how to reduce the time complexity of PCAG and consider using the variable universe fuzzy controller for parameter compensation.

Acknowledgements

This research was funded through the Major Science and Technology Innovation Project of Shandong Province, China, Grant No. 2020CXGC010110 and Shandong Provincial Transportation Industry Key Laboratory Support Program.

REFERENCES

- Allou, S. & Zennir, Y. (2018). A Comparative Study of PID-PSO and Fuzzy Controller for Path Tracking Control of Autonomous Ground Vehicles. In *15th International Conference on Informatics in Control, Automation and Robotics* (pp. 306-314).
- Amer, N. H., Zamzuri, H., Hudha, K. & Kadir, Z. A. (2017). Modelling and control strategies in path tracking control for autonomous ground vehicles: a review of state of the art and challenges, *Journal of Intelligent & Robotic Systems*, *86*(2), 225-254.
- Amiri, M. S., Ramli, R. & Ibrahim, M. F. (2019). Hybrid design of PID controller for four DoF lower limb exoskeleton, *Applied Mathematical Modelling*, *72*, 17-27.
- Cheng, S., Lu, H., Lei, X. & Shi, Y. (2018) A quarter century of particle swarm optimization, *Complex & Intelligent Systems*, *4*(3), 227-239.
- Dziwiński, P. & Bartczuk, Ł. (2019). A new hybrid particle swarm optimization and genetic algorithm method controlled by fuzzy logic, *IEEE Transactions on Fuzzy Systems*, *28*(6), 1140-1154.
- Farnad, B., Jafarian, A. & Baleanu, D. (2018). A new hybrid algorithm for continuous optimization problem, *Applied Mathematical Modelling*, *55*, 652-673.
- Freire, H., Moura Oliveira, P. B. & Solteiro Pires, E. J. (2017). From single to many-objective PID controller design using particle swarm optimization, *International Journal of Control, Automation, and Systems*, *15*(2), 918-932.
- Han, G., Fu, W., Wang, W. & Wu, Z. (2017). The lateral tracking control for the intelligent vehicle based on adaptive PID neural network, *Sensors*, *17*(6), 1244.
- He, Z., Nie, L., Yin, Z. & Huang, S. (2020). A two-layer controller for lateral path tracking control of autonomous vehicles, *Sensors*, *20*(13), 3689.
- Houssein, E. H., Gad, A. G., Hussain, K. & Suganthan, P. N. (2021). Major Advances in Particle Swarm Optimization: Theory, Analysis, and Application, *Swarm and Evolutionary Computation*, *63*, 100868.
- Huang, G., Yuan, X., Shi, K. & Wu, X. (2019). A BP-PID controller-based multi-model control system for lateral stability of distributed drive electric vehicle, *Journal of the Franklin Institute - Engineering and Applied Mathematics*, *356*(13), 7290-7311.
- Koessler, E. & Almomani, A. (2020). Hybrid particle swarm optimization and pattern search algorithm, *Optimization and Engineering*, *22*, 1539-1555. DOI: 10.1007/s11081-020-09534-7
- Lalwani, S., Sharma, H., Satapathy, S. C., Deep, K. & Bansal, J. C. (2019). A survey on parallel particle swarm optimization algorithms, *Arabian Journal for Science and Engineering*, *44*(4), 2899-2923.
- Li, M., Chen, H., Wang, X., Zhong, N. & Lu, S. (2019). An improved particle swarm optimization algorithm with adaptive inertia weights, *International Journal of Information Technology & Decision Making*, *18*(3), 833-866.

- Li, W., Meng, X., Huang, Y. & Fu, Z. H. (2020). Multipopulation cooperative particle swarm optimization with a mixed mutation strategy, *Information Sciences*, 529, 179-196.
- Mohan, V., Chhabra, H., Rani, A. & Singh, V. (2019). An expert 2DOF fractional order fuzzy PID controller for nonlinear systems, *Neural Computing & Applications*, 31(8), 4253-4270.
- Nie, Z.-Y., Zhu, C., Wang, Q.-G., Gao, Z., Shao, H. & Luo, J. L. (2020). Design, analysis and application of a new disturbance rejection PID for uncertain systems, *ISA Transactions*, 101, 281-294.
- Prasad, V., Pawaskar, D. N. & Seshu, P. (2021). Controller design and multi-objective optimization of heavy goods vehicle suspension system by geometry-inspired GA, *Structural and Multidisciplinary Optimization*, 64(1), 89-111.
- Qiao, N., Wang, L., Zhu, W., Tang, X., Jin, C., Yaoming, L. & Xu, L. (2020). An improved path-tracking controller with mid-angle adaptive calibration for combine harvester, *Journal of Instrumentation*, 15(01), 01025.
- Rodríguez-Molina, A., Mezura-Montes, E., Villarreal-Cervantes, M. G. & Aldape-Pérez, M. (2020). Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem, *Applied Soft Computing*, 93, 106342.
- Roshanzamir, M., Balafar, M. A. & Razavi, S. N. (2020). A new hierarchical multi group particle swarm optimization with different task allocations inspired by holonic multi agent systems, *Expert Systems with Applications*, 149, 113292.
- Subotic, M., Manasijevic, A. & Kupusinac, A. (2020). Parallelized Multiple Swarm Artificial Bee Colony (PMS-ABC) Algorithm for Constrained Optimization Problems, *Studies in Informatics and Control*, 29(1), 77-86. DOI: 10.24846/v29i1y202008
- Tam, J. H., Ong, Z. C., Ismail, Z., Ang, B. C. & Khoo, S. Y. (2019). A new hybrid GA-ACO-PSO algorithm for solving various engineering design problems, *International Journal of Computer Mathematics*, 96(5), 883-919.
- Wang, F., Zhang, H., Li, K., Lin, Z., Yang, J. & Shen, X. (2018). A hybrid particle swarm optimization algorithm using adaptive learning strategy, *Information Sciences*, 436-437, 162-177.
- Wang, L., Wang, W., Du, Y. & Huang, Y. (2019). A novel adaptive fuzzy PID controller based on piecewise PID controller for dynamic positioning of sandglass-type FDPSO, *Journal of Marine Science and Technology*, 24(5), 720-737.
- Xiang, Z., Shao, X., Wu, H., Ji, D., Yu, F. & Li, Y. (2020). An adaptive integral separated proportional-integral controller based strategy for particle swarm optimization, *Knowledge-Based Systems*, 195, 105696.
- Xu, T., Wang, D. & Zhang, W. (2021). A non-overshooting controller for vehicle path following, *Transactions of the Institute of Measurement and Control*, 43(10), 2282-2291.
- Zhao, B., Wang, H., Li, Q., Li, J. & Zhao, Y. (2019). PID trajectory tracking control of autonomous ground vehicle based on genetic algorithm. In *2019 Chinese Control and Decision Conference (CCDC)*, (pp. 3677-3682).
- Zhao, P., Chen, J., Song, Y., Tao, X., Xu, T. & Mei, T. (2012). Design of a control system for an autonomous vehicle based on adaptive-PID, *International Journal of Advanced Robotic Systems*, 9(2), 44. DOI: 10.5772/51314