IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p-ISSN: 2278-8727, Volume 27, Issue 6, Ser. 4 (Nov. – Dec. 2025), PP 01-11
www.iosrjournals.org

PID Parameter Optimization Of Intelligent Vehicle Based On Particle Swarm Optimization Algorithm

Zhang Junxiao, Dong Lin, Chen Zhongjia, Rinoshika Akira, Peng Haitao
(School Of Mechanical And Automotive Engineering, Shanghai University Of Engineering Science, Shanghai 201620, China)
(School Of Aeronautic Science And Engineering, Beihang University, Beijing 100191, China）

Abstract:
Background: *PID control is widely used due to its advantages of simple structure and strong stability. However, the limitations of the trial-and-error method for PID parameter tuning severely restrict the control accuracy and adaptability of intelligent vehicles in complex scenarios.*
Materials and Methods: *To address these drawbacks, this paper proposes a PID parameter optimization scheme for intelligent vehicles (PSO-PID) integrated with the Particle Swarm Optimization (PSO) algorithm. To verify the effectiveness of the PSO-PID scheme, three scenarios are established in the MATLAB simulation environment, and comparative experiments are conducted with trial-and-error PID, Genetic Algorithm-based PID (GA-PID), and Ant Colony Optimization-based PID (ACO-PID).*
Results: *The experimental results show that in the ideal environment, the overshoot of PSO-PID is reduced to 0%, the steady-state error approaches 0, and the adjustment time is only 1.15 seconds. In disturbed and time-varying environments, PSO-PID significantly outperforms other comparative algorithms.*
Conclusion: *Indicating that the proposed scheme greatly reduces the difficulty of PID parameter tuning for intelligent vehicle wheel operation.*
Key Word: *Intelligent vehicle; PID control; Particle swarm optimization (PSO); Parameter optimization; Fitness function; Control performance*

Date of Submission: 09-12-2025 Date of Acceptance: 19-12-2025

I. Introduction

With the increasing popularity of intelligent vehicles in the field of intelligent transportation, numerous technical issues have sparked an upsurge in research and attracted extensive attention. The intelligent vehicle system is a core research direction of intelligent transportation, featuring basic driving functions and intelligent operations such as automatic parking and path tracking. It can enhance driving safety, reduce human operation errors, alleviate traffic congestion, and plan optimal routes. Additionally, it finds applications in fields like military and agriculture, and is expanding toward underwater and aerial scenarios [1]. Among these application scenarios, the dynamic traffic environment is the most critical for intelligent vehicles. During operation, intelligent vehicles need to respond to sudden situations such as pedestrians crossing the road and other vehicles accelerating or decelerating. The dynamic adaptability of PID parameters directly affects trajectory tracking accuracy and driving safety. The trial-and-error method for PID parameter tuning, which uses fixed parameters, is prone to overshoot leading to vehicle sideslip. In low-speed automatic parking, excessive steady-state error may result in parking failure or collisions. Therefore, PID control—one of the most fundamental tasks in intelligent vehicle control—aims to dynamically adjust the control variables through algorithms, enabling the actual motion state of the vehicle to accurately follow the preset target state, and ultimately achieving stable, fast, and overshoot-free control performance.

Related research

Here is the accurate and professional English translation, optimized for academic tone and technical consistency.At present, a variety of sophisticated algorithms are available for addressing such problems. The Genetic Algorithm (GA), leveraging its global search superiority, performs optimal design for a single PID controller, thereby providing better solutions for the parameter initialization and dynamic adjustment rules of the PID controller [2]. Despite its global search capability, this algorithm has drawbacks: the parameter settings of crossover and mutation operators rely on empirical experience, and the convergence rate slows down in the later stage of iteration.To meet the parameter optimization requirements of PID controllers, the Ant Colony Optimization (ACO) algorithm is adopted. It searches for the corresponding optimal solution through two core

processes: its built-in path generation and pheromone iterative updating [3]. However, the scarcity of initial pheromone leads to low search efficiency; in time-varying environments, parameter adjustments lag behind system changes, making it difficult to satisfy the real-time control demands of intelligent vehicles.The trial-and-error PID tuning method still sees certain applications in the industrial sector, yet it presents numerous pain points when applied to multi-variable, strongly coupled nonlinear systems such as intelligent vehicles. Driven by manual experience and based on trial-and-error iteration, this method relies entirely on manual operation and cannot realize automatic parameter adjustment. The quality of parameters is directly determined by the engineer's experience, making it difficult for novices to master. Moreover, the parameters are static—once finalized, they remain fixed and cannot adapt to changes in working conditions.

The Particle Swarm Optimization (PSO) algorithm is a stochastic intelligent optimization algorithm inspired by the collective behavior of biological groups. It abstracts the optimization problem into "particles," where all particles form a "swarm." In the solution space, particles dynamically adjust their movement direction and speed through "individual experience learning" and "group information sharing," ultimately converging to the "global optimal solution" [7]. The core advantages of the PSO algorithm are highly aligned with the PID control requirements of intelligent vehicles: its "group collaboration + individual learning" mechanism can quickly cover the solution space of the three PID parameters, avoiding the local optimum trap of traditional methods. Xu Yangyang et al. [7] conducted a simulation study on PID control optimization of electro-hydraulic servo systems using an improved PSO algorithm, demonstrating its simple principle, easy implementation, and low entry threshold. It also exhibits strong global search ability, being less prone to local optima; it has fewer parameters that are easy to tune and high fault tolerance; and it offers high computational efficiency with low hardware requirements. To effectively apply the PSO algorithm to PID control, HAO FENG et al. [8] proposed an improved PSO-PID controller for trajectory control in electro-hydraulic position servo systems, which solved the trajectory accuracy problem of excavators, achieving high-precision control and fast convergence, while also effectively reducing the number of iterations and improving computational efficiency. Li Hailiang et al. [10] integrated the Genetic Algorithm to optimize initial weights and thresholds, but the algorithm is relatively complex with high computational overhead. Zhao Taoyan et al. [11] designed a nonlinear controller based on immune PID, using an interval type-2 fuzzy logic system to approximate the nonlinear function in the immune feedback law, thereby enhancing the controller's ability to handle and approximate complex uncertain nonlinear systems. Chen Zhuquan [2] proposed an intelligent vehicle debugging method using GA-optimized single-neuron PID, but it lacks strong robustness and still relies on manual debugging. Xu Jian et al. [14] established a mathematical model with feed rate and spindle speed as optimization variables and minimum energy consumption as the optimization objective, using the PSO algorithm for optimization and solution. The results showed that the processing parameters obtained by this optimization method have a significant energy-saving effect.

In view of this, this paper introduces the PSO algorithm based on PID control to iteratively search for the optimal solution. The main purpose is to solve the problems caused by the trial-and-error PID control algorithm and significantly improve the control performance of intelligent vehicles. Finally, under three different environments, the results of trial-and-error PID, Genetic Algorithm, Ant Colony Optimization, and PSO-PID algorithm are compared by analyzing indicators such as overshoot, adjustment time, and settling time.

The main contributions of this paper are as follows:
(1) Clearly summarizes the pain points and operational limitations of PID parameter tuning for intelligent vehicles, points out the irrationality of traditional PID parameter tuning methods, and introduces a new algorithm for optimization and improvement.
(2) Proposes a PID parameter tuning method for intelligent vehicles using the PSO algorithm, which realizes real-time feedback of PID parameters through PSO and conducts corresponding parameter adjustments.
(3) Compares various advanced parameter tuning methods through multi-scenario experiments, demonstrating the advantages of PSO in PID parameter tuning for intelligent vehicles and verifying the effectiveness of the proposed scheme.

Particle Swarm Optimization (PSO)
Problem Description

This section clarifies the core defects of the fuzzy PID controller in the electromechanical servo system, as well as the definitions of relevant symbols and optimization objectives. Fuzzy PID adaptively modifies PID parameters through fuzzy reasoning to address the nonlinear and time-varying characteristics of the system. However, the "subjectivity of fuzzy rules and experience dependence of parameter tuning" render it difficult to simultaneously achieve the multi-objective optimization of "high steady-state precision, fast dynamic response, and small overshoot," thus necessitating the integration of Particle Swarm Optimization (PSO) to overcome these limitations. When applied to the path tracking control of intelligent vehicles, traditional fuzzy PID struggles to balance steering response and trajectory accuracy. The PSO algorithm realizes population-based iterative

optimization by quantifying steady-state error, overshoot, and adjustment time into objective function weights, dynamically adjusting the correction coefficients of Kp, Ki, and Kd to effectively avoid local optima. Meanwhile, a working condition-adaptive mapping model is established to update the fuzzy rule base online based on vehicle speed and road adhesion coefficient, transforming parameter tuning from "experience-driven" to "data-driven" and significantly enhancing the control robustness and global optimality under complex working conditions.

Standard Academic English Translation

The control performance of a PID controller is jointly determined by three parameters: proportional gain (Kp), integral time (Ti), and derivative time (Td). The core of parameter tuning lies in finding an appropriate set of Kp, Ti, and Td to achieve the optimal balance of system performance indicators such as "steady-state precision, dynamic response accuracy, and transition smoothness."

When using the trial-and-error PID method, limitations are often encountered, such as insufficient global exploration (slow convergence in the later stage when the inertia weight w is too large) or weak local exploitation capability (easy trapping in local optima when w is too small). Therefore, the Particle Swarm Optimization (PSO) algorithm is adopted with an adaptive inertia weight strategy, enabling the algorithm to "explore the global solution space with a large w in the early stage and converge to the local optimal solution with a small w in the later stage," thereby improving optimization performance.

In the Particle Swarm Optimization (PSO) algorithm, the velocity update formula of particles in the solution space is as follows:

"v" _"i" ^"t+1" "=w·" "v" _"i" ^"t" "+" "c" _"1" "r" _"1" ("p" 〖"Best" 〗_"i" "-" "x" _"i" ^"t")"+" "c" _"2" "r" _"2" ("gBest-" "x" _"i" ^"t")

Where vi,t denotes the velocity of the i-th particle at the t-th iteration; w is the inertia weight, which controls the particle's original motion trend and balances global exploration and local exploitation; c1 and c2 are acceleration coefficients (also referred to as learning factors) that regulate the influence of the individual's historical optimal experience and the swarm's global optimal experience, respectively; r1 and r2 are random numbers uniformly distributed within the interval (0,1), which enhance the randomness of the search; pBesti represents the personal best position (historical optimal solution) of the i-th particle; gBest denotes the global best position of the particle swarm (the optimal value among the historical optimal solutions of all particles); and xi,t is the position (current solution) of the i-th particle at the t-th iteration.

"x" _"i" ^"t+1" "=" "x" _"i" ^"t" "+" "v" _"i" ^"t+1"

The particle position is determined by "current position + current velocity," which reflects the movement process within the current solution space.

Given the particle swarm's outstanding global optimization capability, its swarm intelligence mechanism enables extensive search within the parameter space, breaking through the limitation of traditional methods that are prone to trapping in local optima. Meanwhile, it features high automation—eliminating the need for manual experience-based trial-and-error methods, it can automatically perform self-iteration to find the optimal parameter combination, significantly improving the efficiency of PID tuning for intelligent vehicles. Additionally, it has low model dependence: optimization can be achieved through "input-output" performance feedback, making it adaptable to various complex electromechanical systems. It also exhibits strong multi-objective coordination, as indicators such as steady-state error, overshoot, and adjustment time can be integrated into a fitness function to realize balanced optimization of multiple control objectives. Furthermore, it remarkably enhances robustness—the optimized parameters can better cope with uncertain operating conditions such as sudden load changes and external disturbances, thereby improving system stability(as shown in Fig. 1).

Based on the above, a corresponding approach is proposed to introduce the particle swarm optimization (PSO) algorithm into the PID parameter control of intelligent vehicles.

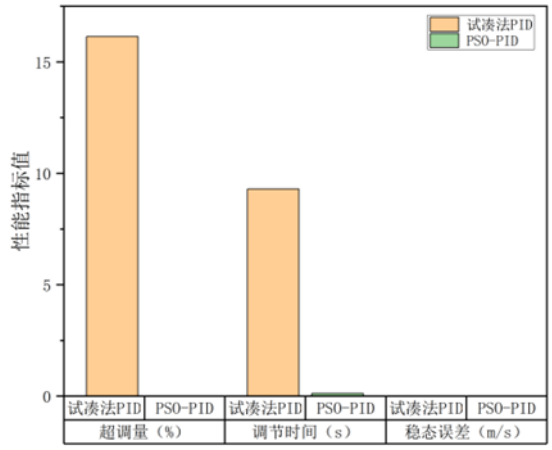

Figure 1 PSO-PID vs. Traditional PID Trial-and-Error Method: A Comparative Analysis

Table 1 The results of the performance comparison

Control Scheme	Overshoot(%)	Settling Time（s）	Steady-State Error（m/s）
Trial-and-Error PID	16.14	9.30	0.013 5
PSO-PID	0.00	0.12	0.003 0

II. Design And Implementation Of The PSO-PID Algorithm

Particle Coding and Fitness Function Definition

The PID parameters["K" _"P" "," "T" _"i" "," "T" _"d"] are treated as the position vector "X" _"i" "=[" "K" _"p,i" "," "T" _"i,i" "," "T" _"d,i" "] " of particles in a three-dimensional solution space, and the velocity vector of particles is defined as "V" _"i" "=" ["v" _("K" _"p" ",i") "," "v" _("T" _"i" ",i") "," "v" _("T" _"d" ",i")].

To comprehensively evaluate the control performance, a fitness function f(x) is defined by simultaneously considering "steady-state error, overshoot, and settling time," with the form expressed as follows:
"f(X)=" "ω" _"1" "·" "e" _"ss" "+" "ω" _"2" "·σ%+" "ω" _"3" "·" "t" _"s" /"t" _"ref" "(7)"

Where ess denotes the system's steady-state error, σ% represents the overshoot, ts is the settling time, and tref is the reference settling time (used for dimensional normalization);ω1,ω2,ω3 are the weights of the performance indicators. The smaller the fitness value, the better the control performance of the corresponding PID parameters.

In the PSO algorithm, the particle velocity update follows the mechanism of "inertia maintenance + individual cognition + social collaboration," and the formula is expressed as follows:
"V" _"i" ^"t+1" "=w·" "V" _"i" ^"t" "+" "c" _"1" "r" _"1" (〖"pBest" 〗 _"i" "-" "X" _"i" ^"t")"+" "c" _"2" "r" _"2" ("gBest-" "X" _"i" ^"t" "(8)")

Where "w" denotes the inertia weight. To balance global exploration and local exploitation, a linearly decreasing strategy is adopted, i.e.,
"w (t)=" "w" _"max" "-" ("w" _"max" "-" "w" _"min")"·" "t" /"t" _"max"

Typically, wmax=0.9 and wmin=0.4; c1 and c2 are acceleration coefficients (also referred to as learning factors) that regulate the "individual's historical optimal experience" and the "influence of the swarm's global optimal solution," respectively; r1 and r2 are random numbers uniformly distributed within the interval (0,1); pBesti denotes the personal best position of the i-th particle; gBest represents the global best position of the particle swarm; t is the current number of iterations; and tmax is the maximum number of iterations [8].

Standard Academic English Translation,
"X" _"i" ^"t+1" "=" "X" _"i" ^"t" "+" "V" _"i" ^"t+1" ("9")

The application of the Particle Swarm Optimization (PSO) algorithm can appropriately reduce the difficulty of PID parameter tuning, and the pseudocode is shown in Algorithm 1:

Algorithm 1 PSO-PID Parameter Optimization Control Algorithm for Intelligent Vehicles

Input: Target velocity vtarget, current velocity vcurrent, system error e, overshoot σ%, settling time ts, reference settling time tref, particle swarm size N, maximum number of iterations max_iter, initial inertia weight wmax, final inertia weight wmin, acceleration coefficients (learning factors) c1 and c2, performance indicator weights ω1, ω2, and ω3.
Output: Optimized PID parameters [Kp,Ti,Td], intelligent vehicle control velocity vcontrol.

1.Initialize_PSO 0;/初始化粒子群，随机生成N 组 PID 参数作为粒子位置"X" _"i" "=[" "K" _"p,i" "," "T" _"i,i" "," "T" _"d,i" "]"， 初始化粒子速度Vi/

2.while选代次数 t< max_iter do

3.for 每个粒子为i in 粒子群 do

4.Compute_PID_Output(Xi);/根据当前PID参数计算控制输出，更新智能车当前速度v_current/

5.Calculate_Performance_Index(e,σ%,t_s);/计算系统稳态误差e、超调量σ%、调节时间t_s/

6.f(X_i) = "ω" _"1" ·e + "ω" _2·0% + "ω" _"3" ·(t s/t_ref;/计尊适应度函数值/

7.Update_pBest(f(X_i),X_i);/ 更新粒子个体最优位置为 pBest i/

8. end for

9.Update_gBest(pBest_i);/更新粒子群全局最优位置 gBest/

10.w(t)= w_max-(w_max-w_min)·(t/max_iter);/更新慢性权重(线性递减策略)/

11.for 每个粒子i in 粒子群 do

12.V_i^(t+1)= w:V i^t+ c1.r1.(pBesti-X_i^t)+ c2.r2.(gBest-X_i^t; /更新粒子速度(r1、r2 为(0,1)随机数)/

13.X_i^(t+1) = X_i^t+ V_i^(t+1);/更新粒子位置 (PID 参数)/

14.end for

15.t=t+ 1;

16.end while

17.[Kp_opt, Ti_opt, Td_opt]= gBest; /获取最优PID 参数/

18.v_control=Kp_opt.e+ (Kp_opt/Ti_opt),fe dt + Kp_opt·Td_opt·(de/dt); /计算最终控制速

19.Send_Control_Signal(v_control); /向智能车控制系统发送速度控制指令/

20.End_Control0;/控制流程结束/

To verify its rationality, the simulation environment is configured as follows: particle swarm size (N): 15 particles; maximum number of iterations (max_iter): 30 generations; acceleration coefficients (learning factors) (c1,c2): both set to 2; inertia weight (w): linearly decreasing from 0.9 to 0.4. Finally, through iterative updates, the particles gradually converge to the region of the "optimal PID parameter combination" (as shown in Fig. 2).

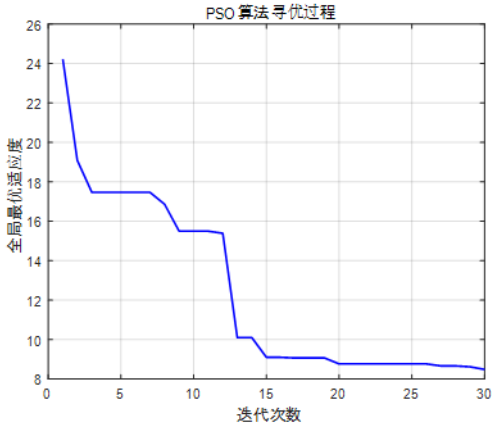

Figure 2 Fitness Function Value Change Curve

III. Specific Process

The PSO-PID algorithm introduces the Particle Swarm Optimization (PSO) algorithm into the conventional PID parameter tuning process, providing a concise and efficient implementation pathway for PID parameter tuning of intelligent vehicles (the algorithm flow is shown in Fig. 3).

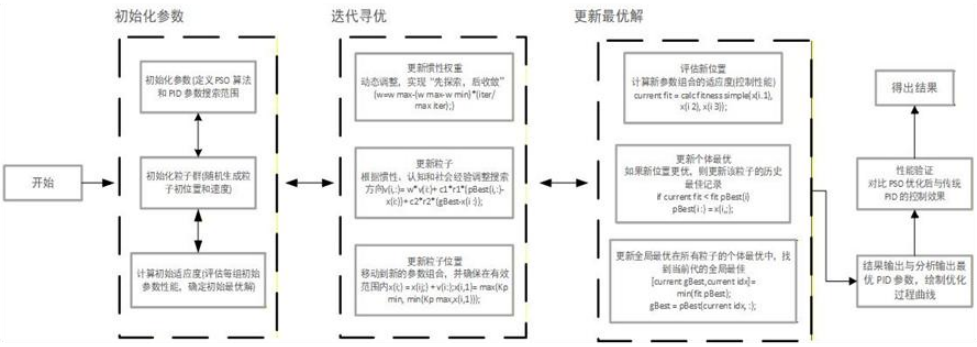

Figure 3 PSO-PID Algorithm Flow

IV. Experiments And Analysis

To verify the rationality of the aforementioned content, we conduct a comparative analysis between the proposed algorithm and the various algorithms introduced in the introduction, aiming to validate the advancement of its performance. The evaluation metrics include algorithm stability, response speed, and adjustability.

Experimental Conditions

This paper conducts offline field tests based on the intelligent vehicle track to ensure the program's usability (as shown in Fig. 4). Taking the 19th National University Student Intelligent Car Competition as the foundation, the Y-car model of the intelligent vision group is adopted for vehicle model construction (as shown in Fig. 5), verifying that the PSO algorithm can be applied to PID parameter tuning with high efficiency and stability

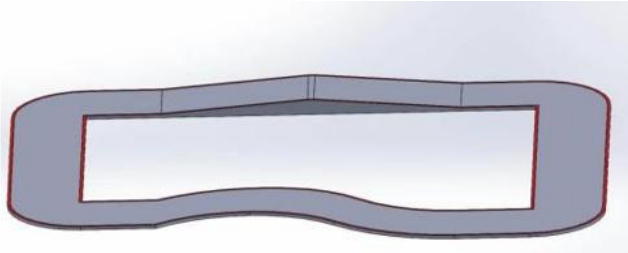

Figure 4 Example of Intelligent Vehicle Track Model

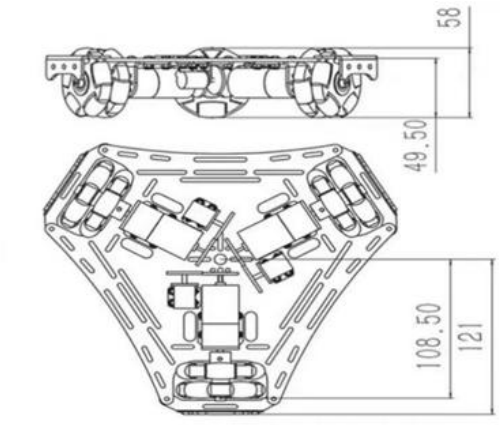

Figure 5 Reference Diagram of Y - shaped Car Model

To verify the robustness and effectiveness of the proposed method in this paper, the computer hardware configuration is as follows: 16 GB of RAM, an RTX 3060 graphics card, and an Intel Core i9-11900 CPU. Comparative experiments under multiple different environments are conducted using MATLAB simulation

software. Focusing on the parameter optimization problem of PID controllers, four algorithms are compared: trial-and-error PID, genetic algorithm-based PID (GA-PID), ant colony optimization-based PID (ACO-PID), and PSO-PID. Three different environments are designed for data comparison (as shown in Table 2).

Table 2 Environmental Conditions

Environment Type	Environmental Characteristics	Mathematical Model
Ideal Environment	No external disturbances and constant load	$v(t) = -kd \cdot v(t) + ku \cdot u(t)$
Disturbed Environment	Superimposed random pulse disturbances and ±10% load fluctuation	$v(t) = -kd \cdot v(t) + ku \cdot (1 + \Lambda kL) \cdot u(t) + drand(t)$
Time-Varying Environment	Time-varying model parameters and slow external disturbances	$v(t) = -[kd_0 + kd_1 \sin(\omega_t \cdot t)] \cdot v(t) + ku \cdot u(t) + dslow(t)$

Comparison and Analysis of Experimental Results

Data comparison is conducted among the various PID algorithms mentioned in the introduction. Comparative analysis is performed under specified conditions, with data simulation carried out across three environments: "Ideal Environment," "Disturbed Environment," and "Time-Varying Environment." The results are presented in Table 3.

Table 3 Three Types of Environments

Comparison dimension	Ideal environment	Disturbed environment	Time-varying environment
Core interference types	No interference	Random transient pulse interference	Time-varying parameters
Load state	Constant	±10%Random fluctuation	Indirectly fluctuate with parameter changes
Model characteristics	Parameters remain unchanged	Linear time-invariant	Dynamic parameter changes
Experimental objectives	Verify the basic control performance	Verify the anti-random interference ability	Verify parameter adaptability and long-term stability
Key indicators	Overshoot、Settling Time、Steady-State Error	Disturbance suppression speed	Parameter tracking accuracy

Verify the algorithm's stability and adjustability

We employed the Keil uVision5 compilation environment to compare the performance of four aforementioned homogeneous PID parameter tuning algorithms, namely trial-and-error PID, genetic algorithm-based PID (GA-PID), ant colony optimization-based PID (ACO-PID), and PSO-PID. Three distinct environments were assumed: Ideal Environment, Disturbed Environment, and Time-Varying Environment. Experiments were repeated multiple times under each of these three environments.

For the comparative analysis of settling time: The genetic algorithm involves mandatory operations such as encoding, decoding, crossover, and mutation. These processes are performed on the entire population in each iteration, resulting in a reduced computational speed. Specifically, the settling time reaches 2.1 s in the Ideal

Environment, 2.0 s in the Disturbed Environment, and increases to 3.75 s in the Time-Varying Environment due to the dynamic changes of parameters.

The ant colony algorithm requires pheromone updating in each iteration, and its initial iteration speed is relatively slow, necessitating a buffer period. Consequently, the settling time is 1.25 s in the Ideal Environment, 1.5 s in the Disturbed Environment, and 2.2 s in the Time-Varying Environment—where pheromone guidance fails due to dynamic parameter variations.

Experimental simulation results indicate that the PSO-PID algorithm exhibits exceptional rapidity and steady-state accuracy, with a settling time of only 1.15 s in the Ideal Environment. In the Disturbed Environment, the settling time of PSO-PID is 1.10 s, and in the Time-Varying Environment, it is 2.17 s (as shown in Fig. 6).

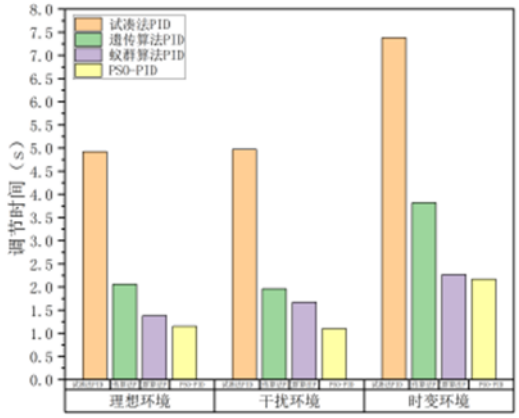

Figure 6 Comparison of Settling Time

Comparative experiments were also conducted to test the overshoot under different algorithms. As illustrated in Fig. 7, the genetic algorithm-based PID (GA-PID) is susceptible to parameter fluctuations during the optimization process due to the randomness of crossover and mutation operators. Specifically, its overshoot is 1% in the Ideal Environment, 0.35% in the Disturbed Environment, and increases to 4.5% in the Time-Varying Environment—where dynamic parameter changes exacerbate fluctuations.

In the Disturbed Environment, the overshoot of PSO-PID is 0.46%, demonstrating outstanding capability in rapidly suppressing disturbances and restoring stability. In the Time-Varying Environment, the overshoot of PSO-PID is 2.37%, indicating that it still maintains a small overshoot and exhibits excellent dynamic adaptability under such conditions.

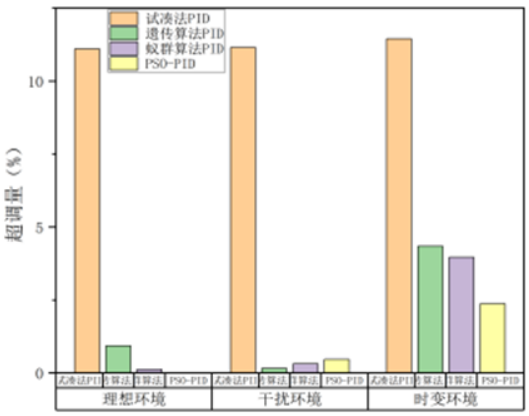

Figure 7 Comparison of Overshoot

A series of comparative analyses were conducted on the steady-state error of various algorithms. It was found that the ant colony optimization-based PID (ACO-PID) is prone to errors due to excessive pheromones and tends to fall into local optima. For the genetic algorithm-based PID (GA-PID), the limited precision of discrete encoding leads to certain information loss, resulting in the existence of steady-state error. Simulation results show

that in the disturbed environment of the experiment, the steady-state error of PSO-PID is 0.0067. In the time-varying environment, the steady-state error of PSO-PID is 0.0286, indicating that it still maintains excellent adaptability and control accuracy even under the time-varying environment with dynamic parameter changes (as shown in Fig. 8).

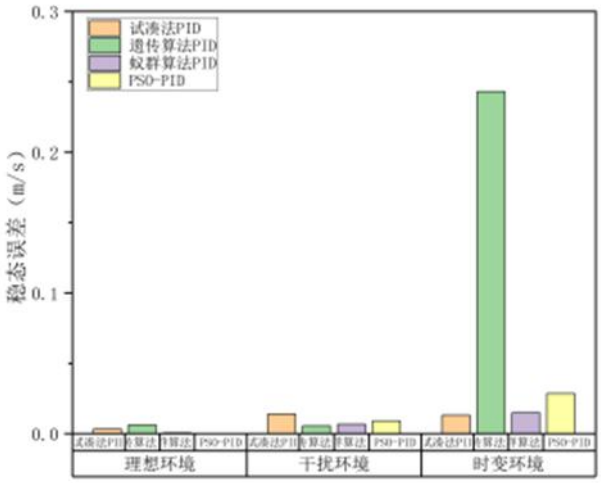

Figure 8 Comparison of Steady-State Error

Based on the comparisons across the three distinct environments, the stability of the PSO-PID algorithm exceeds that of the other algorithms to a certain extent, indicating that its stability is superior to the latter.

Verify the algorithm's response speed

Speed parameter tuning was further performed under the three pre-established mathematical model environments, with homogeneous comparative analysis conducted to verify the effectiveness of its response speed.

The target speed was set to 5 m/s, the total simulation time was assumed to be 10 s, and the sampling period was 0.01 s.

As illustrated in Fig. 9, in the Ideal Environment, the speed of the PSO-PID algorithm reaches 5.05 m/s at t = 1.0 s, which is close to the target speed. In contrast, the speed of the trial-and-error PID is only 3.87 m/s, and that of the genetic algorithm-based PID (GA-PID) is 4.21 m/s. It can thus be concluded that the PSO-PID algorithm exhibits a significantly higher speed increase rate during the start-up phase. Regarding the target arrival time, the PSO-PID algorithm only requires 1.58 s—representing a 44.6% reduction compared to the trial-and-error PID (2.85 s) and a 29.1% reduction compared to the GA-PID (2.23 s). This implies that when applied to intelligent vehicles, the PSO-PID algorithm enables faster entry into a stable driving state and reduces trajectory deviation during the start-up phase. In terms of response delay, the PSO-PID algorithm achieves 0.12 s, which is a 33.3% reduction compared to the trial-and-error PID (0.18 s), demonstrating superior real-time performance and avoiding speed control lag caused by algorithmic delay.

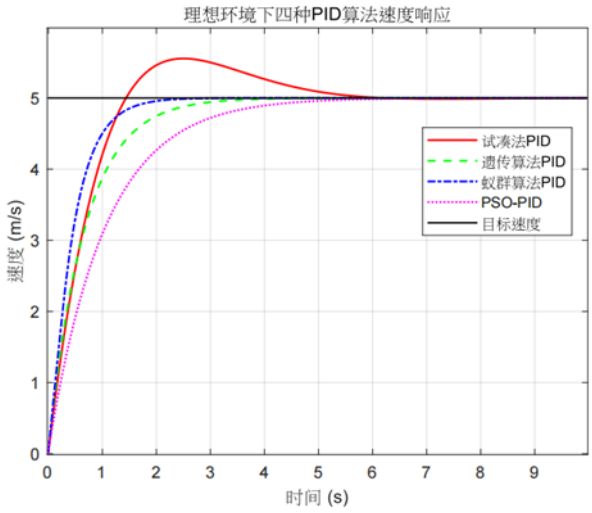

Figure 9 Comparison of Response Speed in Ideal Environment

In the Disturbed Environment, at t = 3 s, the speed of the PSO-PID algorithm reaches 4.78 m/s, which, as illustrated in the figure, is merely 0.22 m/s lower than the target speed. In contrast, the speed of the trial-and-error PID drops to 4.21 m/s, while that of the genetic algorithm-based PID (GA-PID) is 4.53 m/s—both slower than the PSO-PID algorithm. This indicates that the PSO-PID algorithm is less susceptible to disturbances, with a speed fluctuation amplitude only 27.8% of that of the trial-and-error PID. These results demonstrate that the PSO-PID algorithm possesses exceptional anti-disturbance capability in the Disturbed Environment, enabling it to rapidly suppress random pulse disturbances, maintain speed stability, and ensure that intelligent vehicles can still operate stably when encountering sudden disturbances in complex traffic scenarios (as shown in Fig. 10).

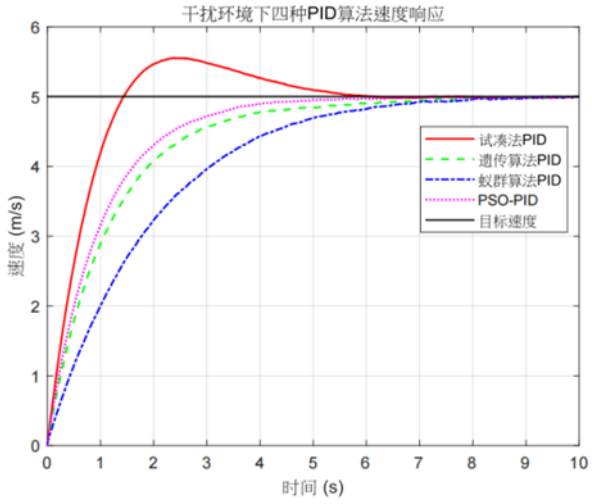

Figure 10 Comparison of Response Speed in Disturbed Environment

In the Time-Varying Environment, at t = 3 s, the speed of the PSO-PID algorithm reaches 4.89 m/s with a deviation of 0.11 m/s. In contrast, the deviation of the trial-and-error PID is as high as 0.65 m/s, and that of the ant colony optimization-based PID (ACO-PID) is 0.49 m/s. Similarly, it can be observed that the PSO-PID algorithm exhibits stronger adaptive capability to time-varying parameters, with a significantly reduced deviation compared to other algorithms. This demonstrates its exceptional parameter adaptability and long-term stability, ensuring that intelligent vehicles can still accurately track the target trajectory when encountering time-varying parameters and slow disturbances in dynamic traffic scenarios (as shown in Fig. 11).

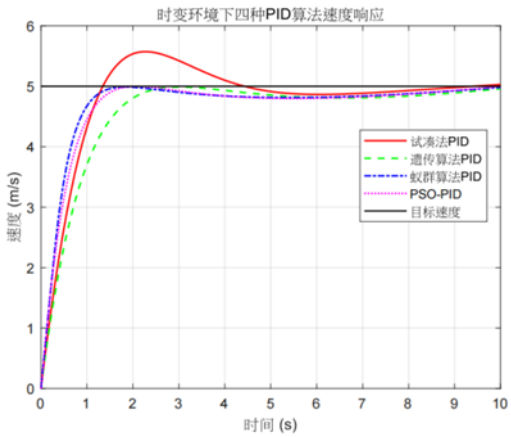

Figure 11 Comparison of Response Speed in Time-Varying Environment

Through repeated comparative experiments conducted across the three distinct environments, it is ultimately concluded that the speed performance of the PSO-PID algorithm is at a normal level compared to most other algorithms. With excellent response speed and the ability to rapidly reach the required target speed, it can serve as a novel PID parameter tuning method.

V. Conclusion

A PID-PSO algorithm is proposed, which leverages the iterative update mechanism of the particle swarm optimization (PSO) algorithm to reduce the difficulty of PID parameter tuning for intelligent vehicle wheel motion control. By adopting a novel theoretical mathematical model, the algorithm achieves superior computational response performance. It is applicable to PID control parameter tuning for various types of intelligent vehicles, featuring broad versatility.

Compared with various existing parameter tuning methods—each with its own inherent advantages—a series of comparative experiments demonstrate the rationality of introducing the PSO algorithm for intelligent vehicle parameter tuning. The proposed algorithm also exhibits inherent logical advantages in parameter tuning, which can reduce the workload associated with traditional tuning methods. The optimization effect of the algorithm has been verified through experimental tests.

References

[1]. Ding P. Research On PID Control Of Self-Tracking Intelligent Vehicle[D]. University Of South China, 2018.
[2]. Chen Z Q. Research On Intelligent Vehicle Path Tracking Algorithm Based On Genetic Algorithm Optimized Single-Neuron PID Control[J]. Information Recording Materials, 2025, 26(07): 47-49+75. DOI:10.16009/J.Cnki.Cn13-1295/Tq.2025.07.061.
[3]. Gong X. Research On Power PID Control Of Ship Inverter Controller Based On Ant Colony Algorithm[J]. Ship Science And Technology, 2023, 45(20): 142-145. DOI:CNKI:SUN:JCKX.0.2023-20-020.
[4]. Huang J F, Zhou Y L, Wang H T, Et Al. Underwater Acoustic Transponder Sound Field Modeling Method Based On Simulated Annealing Algorithm[J/OL]. Chinese Journal Of Ship Research, 1-10[2025-10-06].
[5]. Wang Y Q, Xu X Y, Jiang Q S. Control Design Of Crane-Double Pendulum System Based On Immune PID Algorithm[J]. Control Engineering Of China, 2016, 23(06): 895-900. DOI:10.14107/J.Cnki.Kzgc.150664.
[6]. Huang K Q, Xu J. Mobile Robot Path Planning Based On Multi-Strategy Improved Whale Optimization Algorithm[J]. Transducer And Microsystem Technologies, 2025, 44(07): 121-125. DOI:10.13873/J.1000-9787(2025)07-0121-05.
[7]. Xu Y Y, Feng C, Xue D J. Simulation Research On PID Control Of Electro-Hydraulic Servo System Optimized By Improved Particle Swarm Optimization Algorithm[J]. Chinese Journal Of Construction Machinery, 2025, 23(04): 586-590. DOI:10.15999/J.Cnki.311926.2025.04.033.
[8]. Hao Feng, Wei Ma, Chenbo Yin, Donghui Cao,Trajectory Control Of Electro-Hydraulic Position Servo System Using Improved PSO-Pidcontroller,Automation In Construction,Volume 127,2021,103722,ISSN 0926-5805
[9]. Zhang Y P, Tang W X, Xiao Y N, Et Al. Application Of BP-PID Optimized By Hybrid Algorithm In SLS Preheating Temperature Control System[J]. Manufacturing Automation, 2022, 44(10): 132-136.
[10]. Li H L, Li Z G, Ning X G, Et Al. Mobile Robot Path Planning Algorithm Based On Ant Colony Algorithm Guided Deep Q-Network[J/OL]. Acta Armamentarii, 1-14[2025-11-12].
[11]. Zhao T Y, Cao J T, Li P, Et Al. Application Of Interval Type-2 Fuzzy Immune PID In Cyclohexane Non-Catalytic Oxidation Temperature Control System[J]. CIESC Journal, 2022, 73(07): 3166-3173.
[12]. Ni M, Geng S J, Zhang X Y, Et Al. Influence Of Air Humidity On Performance Of Two-Stage Opposed Centrifugal Compressor For Marine Gas Turbine[J]. Ship Science And Technology, 2020, 42(19): 110-117.
[13]. Yan J G. Fault Prediction And Maintenance Technology Of Central Air Conditioning System Based On Swarm Intelligence[J]. Development & Innovation Of Machinery & Electrical Products, 2025, 38(01): 106-109.
[14]. Xu J, Lin X K, Han S Z. Optimization Of Cutting Parameters For Dual-Station Based On Particle Swarm Optimization Algorithm[J]. Manufacturing Automation, 2011, 33(17): 42-44.
[15]. Yang J W, Gong T, Guo L, Et Al. Research On Dynamic Performance Testing And Control Technology Of High-Precision Mechanical Equipment[J]. China Plant Engineering, 2025, (07): 190-192.