

PID Steering Control Method of Agricultural Robot Based on Fusion of Particle Swarm Optimization and Genetic Algorithm

ZHAO Longlian^{1,2} ZHANG Jiachuang^{1,2} LI Mei^{1,2} DONG Zhicheng^{1,2} LI Junhui^{1,2}

(1. College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China

2. Key Laboratory of Smart Agriculture System Integration, Ministry of Education, Beijing 100083, China)

Abstract: Aiming to solve the steering instability and hysteresis of agricultural robots in the process of movement, a fusion PID control method of particle swarm optimization (PSO) and genetic algorithm (GA) was proposed. The fusion algorithm took advantage of the fast optimization ability of PSO to optimize the population screening link of GA. The Simulink simulation results showed that the convergence of the fitness function of the fusion algorithm was accelerated, the system response adjustment time was reduced, and the overshoot was almost zero. Then the algorithm was applied to the steering test of agricultural robot in various scenes. After modeling the steering system of agricultural robot, the steering test results in the unloaded suspended state showed that the PID control based on fusion algorithm reduced the rise time, response adjustment time and overshoot of the system, and improved the response speed and stability of the system, compared with the artificial trial and error PID control and the PID control based on GA. The actual road steering test results showed that the PID control response rise time based on the fusion algorithm was the shortest, about 4.43 s. When the target pulse number was set to 100, the actual mean value in the steady-state regulation stage was about 102.9, which was the closest to the target value among the three control methods, and the overshoot was reduced at the same time. The steering test results under various scene states showed that the PID control based on the proposed fusion algorithm had good anti-interference ability, it can adapt to the changes of environment and load and improve the performance of the control system. It was effective in the steering control of agricultural robot. This method can provide a reference for the precise steering control of other robots.

Key words: agricultural robot steering; PID control; particle swarm optimization algorithm; genetic algorithm

CLC number: TP242 **Document Code:** A **Article ID:** 1000-1298(2026)01-0358-10 **OSID:** 

0 Introduction

With the development of artificial intelligence technology, the traditional production mode that relies on a large number of labors will be gradually replaced by robots. The intelligent research of agricultural robots has become a current research hotspot^[1-3]. In order to realize the autonomous navigation movement of robots in complex environment (for example the garden), steering control is one of the primary problems and technical difficulties, such as improving the

sensitivity, accuracy and stability of the system steering^[4-6].

PID algorithm is the most classic algorithm in the field of automatic control, which is widely used in the navigation control of agricultural machinery^[7-10]. The algorithm consists of three parts: proportion, integral and differential. The selection of parameters in the algorithm will greatly affect the effect of PID control. Parameter tuning is the core of PID algorithm. The tuning methods mainly include two kinds: ①the classic manual trial and error method. This method tries and

calculates parameters through experience. The process is cumbersome and complex, and the control accuracy is difficult to be guaranteed^[11]. SEN et al.^[12] established a kinematic model for the control system of agricultural plant protection vehicle, and adjusted the PID parameters by manual trial and error, so that the system can meet the operation requirements, but the reasonable selection of parameters depends on a lot of experiments and manual experience. ②The intelligent tuning method, which uses some intelligent algorithms to tune the PID parameters of the system. This method has strong robustness and can simplify the complex debugging process. ZHANG et al.^[13] proposed a fuzzy PID control method based on improved quantum genetic algorithm, designed a fuzzy self-tuning PID controller, and optimized the parameters of the fuzzy self-tuning PID controller by using quantum genetic algorithm to improve the dynamic performance of the system. However, the algorithm was only used in simulation experiments, and the effect was not verified in actual experiments. SONG et al.^[14] proposed an improved genetic algorithm (IQGA) which introduced a dynamic adjustment strategy of population and search intervals to improve the convergence speed. Experimental results indicated that the method had a good performance in PID parameters tuning. HAMEDANI et al.^[15] developed an adaptive dynamic surface impedance (ADSI) control approach for an n-link robotic manipulator by employing self-recurrent wavelet neural networks (SRWNNs) in order to overcome the saturation. The simulation results indicated the robustness of the controller against uncertainties and external disturbances.

At present, many tuning algorithms have superior effects in simulation, but in practical applications, the selection of parameters still requires a lot of experimental and artificial experience, or requires a large hysteresis in the controlled object. In order to make the tuning algorithm more concise and efficient, and meet the needs of the actual scene at the same time, on the basis of summarizing the above research results, a PID parameter tuning method that fused particle swarm optimization (PSO) and genetic algorithm (GA) was proposed, and it was compared with the system response curve obtained by artificial trial and error method PID parameter tuning and

genetic algorithm PID parameter tuning to analyze its dynamic response performance. The fused method was applied to the steering control of the actual agricultural robot. The steering system of the agricultural robot was modeled, the response time and overshoot of the system were observed, and the robustness of the system was analyzed.^[16]

1 Fusion method of PSO and GA

1.1 Basic evolutionary algorithm

Genetic algorithm is an intelligent optimization method that searches for the optimal solution by simulating the natural evolution process. It is applied in many fields and is suitable for solving complex optimization problems. The algorithm has strong robustness. However, the local search ability and convergence speed of genetic algorithm need to be further improved in practical applications.

Particle swarm optimization algorithm^[17-18] is also an evolutionary algorithm like genetic algorithm. Its idea is to use the information sharing and team cooperation of individuals in the population to find the best. This algorithm have a strong search ability in the global range.

1.2 Particle swarm optimization genetic fusion algorithm

In view of the problems of poor local search ability and slow convergence speed of genetic algorithm, while particle swarm optimization algorithm has fast optimization ability, consider trying to introduce particle swarm optimization algorithm into the population screening process of genetic algorithm. The population initialization phase in genetic algorithms often involves undirected randomness, which can prolong the optimization process and hinder convergence speed. Keeping the good half of the population and optimizing the poor individuals through the particle swarm optimization algorithm can speed up the convergence of individuals, form a new population, speed up the convergence of the population and the optimization process, and improve the convergence speed of the algorithm, that is, particle swarm genetic fusion algorithm.

In the screening process of the algorithm population, by sorting the fitness values, half of the individuals with good performance were retained and

directly entered the next generation population. The half of the individuals with poor fitness were optimized by particle swarm optimization algorithm. Optimization process: the fitness function consistent with the genetic algorithm was used as the evaluation standard of particle performance. Firstly, calculating the fitness function value of the worse half of the particles, evaluating the performance of each particle, and getting the global optimal particle at the moment. Through information sharing and cooperation, constantly updating the position and speed of each particle to make it tend to the global optimal particle movement, and calculating the updated particle fitness value. Thus, the updated global optimal solution was obtained, and the poor individuals were continuously optimized and updated in this way.

In the optimization phase, individual i updated its position and speeded according to the following formula

$$V_{k+1}^i = w_k V_k^i + c_1 \text{rand}(P^i - X_k^i) + c_2 \text{rand}(P_G - X_k^i) \quad (1)$$

$$X_{k+1}^i = X_k^i + V_{k+1}^i dt \quad (2)$$

Where k was the number of iterations, w_k was the inertia weight coefficient with attenuation, V_k^i was the velocity of the particle, X_k^i represented the position of the particle at this time, c_1, c_2 was the learning factor, $\text{rand}()$ was the random number evenly distributed in $(0,1)$, P^i was the position of the i^{th} particle with the highest adaptability in all previous iterations, P_G was the position of the population with the highest adaptability in all previous iterations, that was, the global extreme value. The particle kept tracking the individual extreme value and the global extreme value in the solution space to search until the specified number of iterations or the specified error standard was met. After the renewal, this part of the population would enter the next generation of population, and then selected, crossed and mutated the population. The algorithm flow was shown in Fig. 1.

2 Fusion algorithm to adjust PID parameters

2.1 PID control principle

The formula of PID control can be expressed as

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (3)$$

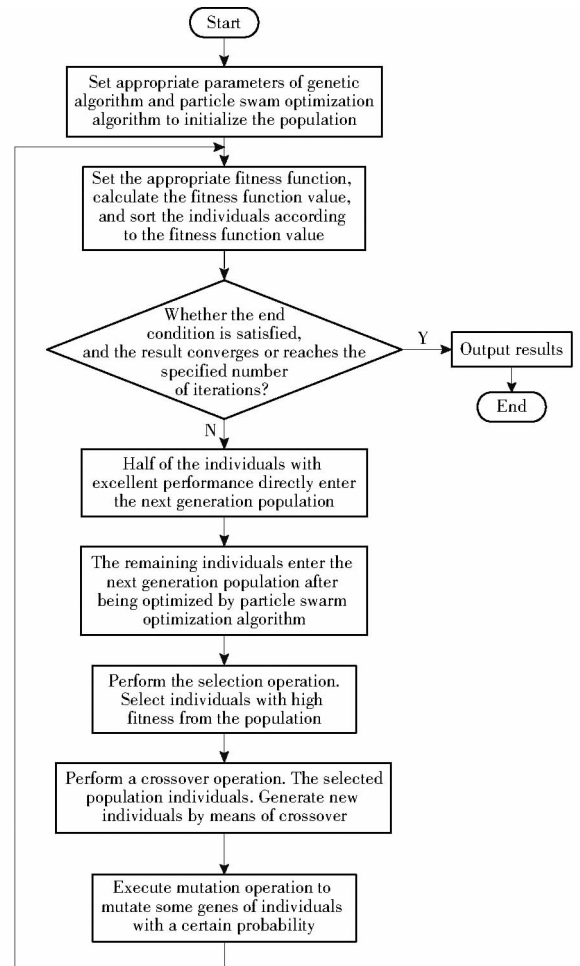


Fig. 1 Flowchart of fusion algorithm

Where $u(t)$ was the output signal of the controller, $e(t)$ was the system deviation, K_p, K_i, K_d was the proportional, integral and differential coefficients, respectively. The core problem of tuning PID control parameters was to select the most appropriate K_p, K_i, K_d parameters combination to achieve the best performance index of the system. The structure of the particle swarm genetic fusion algorithm proposed was shown in Fig. 2. The system deviation $e(t)$, the output change rate of the controlled object $e_y(t)$, and the output $u(t)$ of the controller were taken as the input variables of the particle swarm genetic fusion algorithm, where $e_y(t) = y(t) - y(t - 1)$, which was the difference between the current output $y(t)$ of the controlled object and the previous

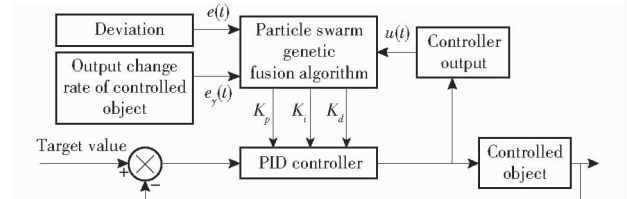


Fig. 2 PID controller structure of particle swarm optimization genetic fusion algorithm

output $y(t-1)$. The parameters of PID control were adjusted by this algorithm.

2.2 Selection of fitness function

In order to evaluate the performance of PID control system, it was necessary to design a reasonable fitness function. After comprehensive consideration, the performance index function J adopted by the optimization algorithm was

$$J = \begin{cases} \int_0^{\infty} (a_1 |e(t)| + a_2 u^2(t)) dt + a_3 t_s & (e_y(t) \geq 0) \\ \int_0^{\infty} (a_1 |e(t)| + a_2 u^2(t) + a_4 |e_y(t)|) dt + a_3 t_s & (e_y(t) < 0) \end{cases} \quad (4)$$

Of which $\int_0^{\infty} a_1 |e(t)| dt$ was the performance index of time integration of absolute value of error, which reflected the relationship between error and time. In order to prevent excessive control energy, the square term of output $\int_0^{\infty} a_2 u^2(t) dt$ was introduced into the fitness function. In order to reflect the rapidity of system response, the rise time t_s was introduced. At the same time, in order to avoid overshoot, when

Table 1 Comparison of iterative optimization results of two algorithms

Algorithm	K_p	K_i	K_d	Rise time/s	Adjustment time/s	Overshoot/%	Performance index function J
Genetic algorithm	19.35	0.24	0.05	0.02	0.15	24	24.25
Fused of PSO and GA	19.70	0.24	0.21	0.03	0.04	2	24.11

It can be seen from Table 1 that the PID control based on the fused algorithm had good control effect. Compared with the PID control based on genetic algorithm, the adjustment time was significantly reduced, and there was almost no overshoot, and its performance index was also improved to a certain extent. Fig. 3 showed the comparison of the iterative process of the two optimization algorithms, and Fig. 4 showed the step response curve results of the two optimization algorithms.

As can be seen from Fig. 3, the convergence of the fitness function, that was, the performance index function J , of the fusion algorithm was accelerated, and the final value was decreased. Observing the step response curve of the system (Fig. 4), it was found that although the response rise time did not decrease, its adjustment time was decreased from 0.12 s to 0.04 s, a decrease of about 67%, and the overshoot

overshoot occurred, $\int_0^{\infty} a_4 |e_y(t)| dt$ was introduced as the overshoot indicator. a_1, a_2, a_3 and a_4 represented different weight coefficients, and the smaller the value of performance index J was, the better the system was.

2.3 Simulation experiment of fusion algorithm

In order to verify the performance of the fusion algorithm, the PID control based on GA and the PID control based on the fusion algorithm proposed were applied to the controlled object for simulation test. Assume that the transfer function of the controlled object was

$$G(s) = \frac{400}{s^2 + 50s} \quad (5)$$

The parameters in the algorithm were set according to SONG et al. [14]. In order to reflect the objectivity of comparison, the performance of particle swarm optimization genetic fusion algorithm and traditional genetic algorithm adopted the same population number, iteration times, crossover and mutation probability and performance index. The iterative optimization results of the two algorithms were shown in Table 1.

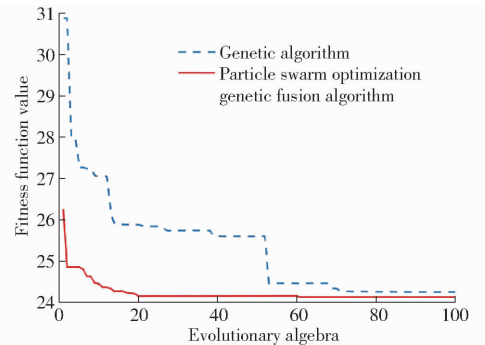


Fig. 3 Comparison of iterative process between two optimization algorithms

almost approached 0, indicating that the response speed and stability of the system were improved.

3 Establishment of steering system model of agricultural robot

Before the experiment, the steering system of agricultural robot should be modeled first. The physical

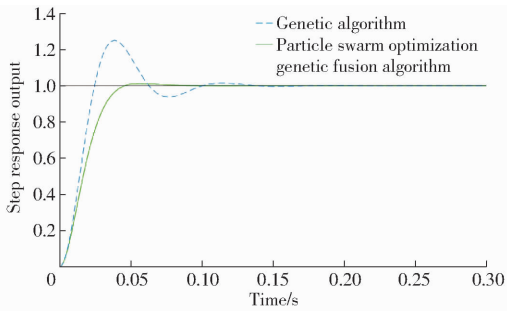


Fig. 4 Comparison of step response simulation results

diagram of the steering structure of the robot motion chassis was shown in Fig. 5a, and the geometric diagram of the steering structure was shown in Fig. 5b. The steering of the robot depended on the expansion and contraction of the electric push rod. Where point *A* was the fixed end of the electric push rod, point *B* was the telescopic end of the electric push rod, and point *C* was the steering shaft of the robot.

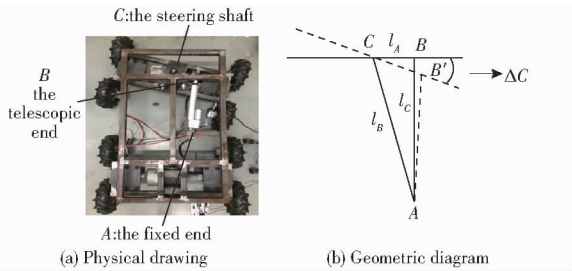


Fig. 5 Physical drawing and geometric diagrams of robot steering structure

When the front wheel of the robot was in balance, the length of l_A (cm), l_B (cm), l_C (cm) and $\angle ACB$ can be obtained. Through the cosine theorem formula, we can get

$$\cos \angle ACB = \frac{l_A^2 + l_B^2 - l_C^2}{2l_A l_B} \quad (6)$$

When the electric push rod was shortened, the wheel would turn to the right. The length of electric push rod was shortened by Δl_C (cm). When the wheel rotated by a certain angle ΔC , which was met at this time

$$\cos(\angle ACB - \Delta C) = \frac{l_A^2 + l_B^2 - (l_C^2 - 2l_C \Delta l_C + \Delta l_C^2)}{2l_A l_B} \quad (7)$$

Expanding the formula, taking into account the actual steering control of the robot, in the adjacent time, the value of ΔC and Δl_C was relatively small, and Δl_C^2 can be ignored. Since $\cos \Delta C$ approached 1, $\sin \Delta C$ was approximately equal to ΔC , we can get

$$\cos \angle ACB + \Delta C \sin \angle ACB = \frac{l_A^2 + l_B^2 - (l_C^2 - 2l_C \Delta l_C)}{2l_A l_B} \quad (8)$$

Bring formula (6) into formula (8) to further obtain

$$\Delta C = \frac{l_C}{l_A l_B \sin \angle ACB} \Delta l_C \quad (9)$$

In equation (9), ΔC and Δl_C was the amount that changes with time, that was, the numerical relationship between trolley moving angle $\Delta C(t)$ and the displacement of electric push rod $\Delta l_C(t)$.

In order to verify the performance of the algorithm and reduce the impact of environmental factors on the accuracy of modeling, the vehicle body was suspended first, ignoring the impact of the road and the sliding friction force of the road during the robot movement. It was found from the collected data that there was a certain linear relationship between the displacement speed $V(t)$ (mm/s) of the electric push rod and the motor pulse width modulation (PWM) duty cycle $P(t)$ (%) [19], as shown in Fig. 6.

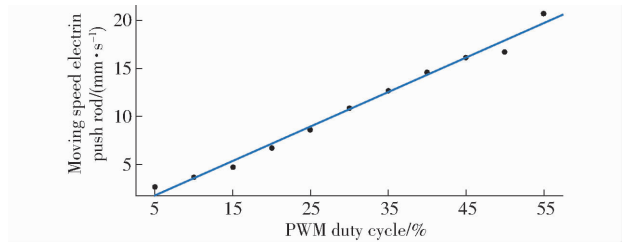


Fig. 6 Relationship between moving speed of electric push rod and PWM duty cycle

It can be found from Fig. 6 that due to the existence of dead band PWM, when the duty cycle was less than 5%, the electric push rod cannot be started. Therefore, it was necessary to superimpose the dead band PWM value on the actual PWM, and calculate the slope of the straight line in Fig. 6 to approximate the numerical relationship between the corrected speed and duty cycle

$$V(t) = 0.38P(t) \quad (10)$$

Then the displacement of the electric push rod was

$$\Delta l_C(t) = \int_0^t V(t) dt = \int_0^t 0.38P(t) dt \quad (11)$$

$$\Delta C(t) = \frac{l_C}{l_A l_B \sin \angle ACB} \int_0^t 0.38P(t) dt \quad (12)$$

The following results were obtained after Laplace transform

$$\Delta C(s) = \frac{l_c}{l_A l_B \sin \angle ACB} \frac{0.38P(s)}{s} \quad (13)$$

Where s was the Laplacian operator, thus, a simple system steering model can be obtained.

For the convenience of test, the target angle value in the PID steering test of agricultural robot was replaced by the target pulse value. Setting the electric push rod to complete a whole stroke movement of s_1 , the encoder actually detects and feeds pulse back to d . When the target pulse value was set to e , the corresponding electric push rod moved $s_1 \frac{e}{d}$. In the

steering test, when it was detected that the actual value of the pulse number was inconsistent with the set target value and the pulse width modulation duty cycle was less than the upper limit threshold, the PID control program automatically accumulated the duty cycle of PWM, so as to overcome the environmental resistance, making the actual value close to the target value and reducing the steering error^[20-21]. From the actual measured data of the robot, it can be calculated that the relationship between the displacement of the robot's electric push rod and the pulse number $k(t)$ is

$$\Delta l_c(t) = 0.55k(t) \quad (14)$$

Therefore, the relationship between pulse number $k(t)$ and PWM was

$$k(t) = \int_0^t 0.69P(t) dt \quad (15)$$

Therefore, the transfer function $H(s)$ of the steering system can be simplified as

$$H(s) = \frac{k(s)}{p(s)} = \frac{0.69}{s} \quad (16)$$

This was the mathematical model of the steering system of agricultural robot in the state of no-load suspension.

4 Steering test and analysis of agricultural robot

4.1 Test equipment

The movement chassis of the bionic agricultural robot shown in Fig. 7 was selected as the experimental object in the test, which is powered by 24V70AH lithium-ion battery. As the steering unit of the trolley, the electric push rod changes the steering speed and angle through the telescopic speed and displacement of the push rod. STM32F103ZET6 single chip microcomputer was selected as the microcontroller, and

programmable output PWM signal acting on the electric push rod. The timer interrupt mode was adopted to read the data of the electric push rod encoder every 10 ms and transmit the data to the serial port assistant at the same time. The electric push rod adopted a 60 A direct drive motor drive module, which can be used to drive general high-power DC motors. The maximum current can reach 60 A, which can meet the normal operation of the electric push rod. The wheel motor adopted 100 A direct drive motor drive module, which can meet the normal operation of the two motors.

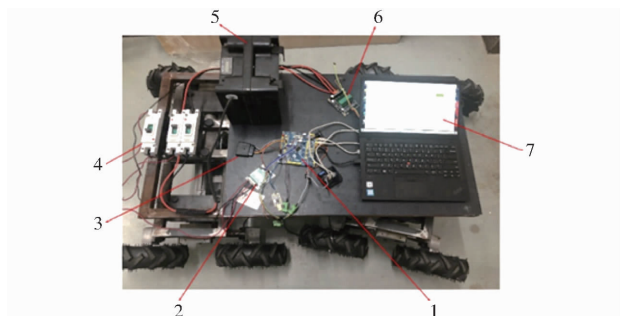


Fig. 7 Physical drawing of agricultural robot chassis

1. Microcontroller unit
2. Electric push rod driver
3. Remote control module
4. Air switch
5. Lithium battery
6. Motor driver
7. Upper computer

The parameters of the electric push rod directly determined the effect of the proposed algorithm in the actual scene. According to the actual situation, the electric push rod with the stroke of 112 mm, the maximum speed of 24 mm/s and thrust of 500 N was selected. According to the installation position of the electric push rod in Fig. 5, the actual turning radius was 1.7 m, which met the turning needs of large orchards. Because the robot was designed to steer while traveling, the wheels were subjected to a rolling friction force opposite to the direction of travel, and the rolling friction coefficient was less than a value of 1, and the thrust of the actual electric actuator was greater than the rated maximum thrust, which met the actual operation needs of the orchard robot in terms of response speed and thrust.

4.2 Steering test of no load suspended state

According to Section 2 and Section 3, the PID control of the fusion algorithm achieved good control effect in Simulink simulation. Due to the complexity of the actual experiment, the actual experimental results may be different from the ideal state. In order to verify the actual control effect, firstly, the PID steering

control test in the unloaded suspended state was carried out on the agricultural robot platform. In this state, the influence of all the road environment can be ignored. At the same time, because it was in the suspended state, the robot steering process, that was, the expansion and contraction process of the electric push rod, was not affected by the load change of the robot. The whole test process was in a relatively ideal environment.

The control program in the robot microcontroller adopted the absolute PID control mode, and the PID parameters of the system were adjusted by manual trial and error method, genetic algorithm and the fusion algorithm. By continuously adjusting three parameters, selecting a group of parameters that can minimize the adjustment time and overshoot of the system and have the best effect as PID parameters adjusted by manual trial and error method, which was $K_p = 50$, $K_i = 0.2$, $K_d = 20$. According to this value in genetic algorithm and fusion algorithm, the value range of K_p , K_i , K_d was set to $[0, 100]$, $[0, 1]$, $[0, 30]$ respectively, and then other parameters in the algorithm were selected according to SONG et al.^[14]. Finally, the PID parameters adjusted by genetic algorithm were: $K_p = 83$, $K_i = 0.04$, $K_d = 0.4$, and the fusion algorithm were: $K_p = 92$, $K_i = 0.1$, $K_d = 6.4$.

In order to observe the effect of PID control with manual trial and error method, PID control based on genetic algorithm and PID control based on the fusion algorithm. At the initial moment, setting the initial state of the robot as the electric push rod shrinks to the shortest, the electric push rod extends from the shortest to the longest, and the turning angle of the robot changed from -20° to 30° . The encoder detected about 300 pulses in the whole process. Therefore, in the PID control program, setting the target pulse value as 50 - 55 - 60 - 65 - 60 - 55 - 50 in turn, corresponding to the robot to achieve different steering angles, and change the target value every 10 s. At the same time, the timer collected the cumulative value of encoder feedback pulses every 10 ms and uploaded it to the serial port assistant. Finally, the response curve of PID was drawn, as shown in Fig. 8.

It can be seen from Fig. 8 that due to the possible reasons of insufficient accuracy of hardware equipment, friction of mechanical structure, environmental noise

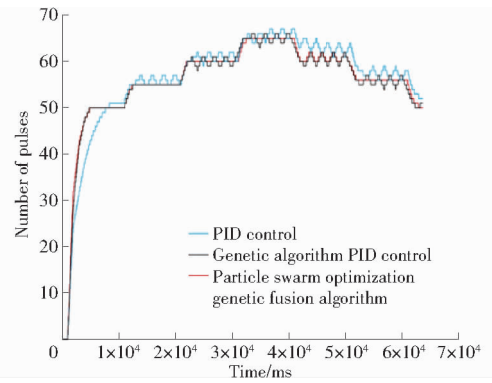


Fig. 8 Steering follow deviation curves

and insufficient accuracy of modeling, the response curve of the actual test of the three methods can not be as idealized as the simulation experiment, and the response exhibited a lag of approximately 1 s. At the same time, even if the target value reached, the response curve was still prone to oscillation. When the artificial trial and error PID control reached the target value, the amplitude of oscillation was significantly larger than that of the other two methods. The rise time of artificial trial and error PID control was about 7 s, while the rise time of genetic algorithm based PID control and fusion algorithm based PID control were about 4 s. The rise time, adjustment time and overshoot were reduced compared with artificial trial and error PID control. At the same time, it was found that the fusion algorithm PID control was easier to stabilize at the target value than the genetic algorithm PID control. After reaching the target value, the response curve can remain stable, and there was a slight oscillation only in the change interval of the target pulse number of 65 - 60, while the genetic algorithm PID control produced oscillation in many change intervals such as 60 - 65, 65 - 60, 60 - 55, and 55 - 50. The overshoot of the fusion algorithm was reduced and the robustness was enhanced. Therefore, the performance of PID control based on fusion algorithm was better than that of artificial trial and error PID control and PID control based on genetic algorithm. Its rise time, adjustment time and overshoot were reduced, and the response speed and robustness of the system were enhanced, which can better realize the motion steering control of the robot.

4.3 Actual road steering test

In order to verify the actual performance of the algorithm and the control effect of the controller, the steering test in the motion state of the robot was carried

out on the playground lawn of China Agricultural University. Compared with the suspension steering test, the robot needed to overcome the sliding friction in the steering process due to its own load weight in the actual road movement process, and it would also be disturbed by the actual road. According to the method in Section 3, the relevant data of the road was collected, and the mathematical model of the agricultural robot steering system was established under the actual road. Through a large number of tests, the PID parameters obtained by manual trial were $K_p = 90$, $K_i = 0.05$, and $K_d = 15$. Setting the value range of K_p , K_i , K_d in the algorithm accordingly as $[0, 150]$, $[0, 1]$ and $[0, 20]$, respectively, so the PID parameter set by genetic algorithm were $K_p = 123.3$, $K_i = 0.07$, $K_d = 0.05$, the PID parameter set by fusion algorithm were $K_p = 137.4$, $K_i = 0.05$, $K_d = 4.02$. Setting the initial state of the robot as the electric push rod shrinks to the shortest position, and setting the target pulse number to 100 in the PID control program, so that the robot can achieve a steering angle of about 17° . Fig. 9 showed the comparison of the steering control effects of the three methods. It was found that the response curves of the three methods of steering test had a response lag of about 1s. Due to the unevenness of the actual road surface, the mutual friction of mechanical structures in the movement process and the complex environmental noise, the response value oscillated near the target value, and there was a certain deviation.

Table 2 Comparison of steering control effects of three methods

Control method	Rise time/s	Steady state regulation	Steady state regulation
		stage average	stage standard deviation
Particle swarm optimization genetic algorithm PID control	4.43	102.9	1.53
Genetic algorithm PID control	4.44	105.6	1.90
Artificial trial and error PID control	5.13	103.8	1.64

steering performance of agricultural robots, and provide a technical reference for the actual motion control of agricultural robot.

The experiment of the maximum speed steering of the electric push rod and the steering with load, by simulating the load weight of the actual spraying or fertilization operation was also carried out, the 50 kg load was placed on the robot, then passed the actual test. The response time of the steering was 6.89 s, meeting the actual production needs. The experiment of no-load maximum steering speed of the electric

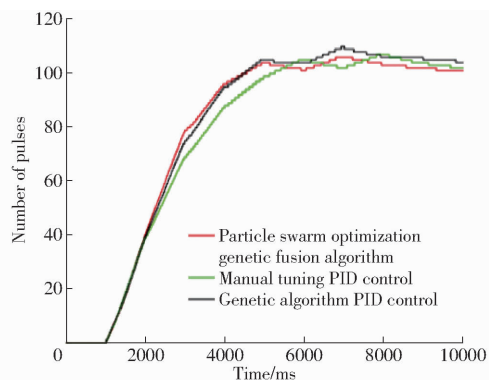


Fig. 9 Comparison of steering control effect

Table 2 showed the comparison of PID control effects by using three algorithms respectively. The test results showed that the response rise time of the artificial trial and error PID control was the longest, about 5.1 s, and the average value of pulse counts for the steady-state adjustment stage (calculated after reaching the target value for the first time) was about 103.8, and the standard deviation was 1.64. The genetic algorithm PID control had the largest deviation from the preset target in the steady-state regulation stage. The response rise time of fusion algorithm PID control was the shortest, about 4.43 s, and the mean value of the steady-state regulation stage was closest to the preset target, and the standard deviation was the smallest. The actual road steering test results showed that the fusion algorithm PID control response rise time, the average value and standard deviation of steady-state adjustment stage were better than that of the other two methods, which can better improve the

actuator was also carried out, at the maximum speed, the response time of the steering was 1.56 s, and the time was reduced by 2.87 s compared with that of the previous no-load experiment. It can be seen that the actual steering response time was changed by changing the speed of the electric push rod, making the steering more flexible in practical use.

5 Conclusions

(1) A novel PID control method was proposed based on the fusion of particle swarm optimization

(PSO) and genetic algorithm (GA), which accelerated the convergence of the performance index J and required fewer iterations compared with the standard GA-based approach.

(2) Both simulation and actual road tests demonstrated that the proposed fusion algorithm achieved superior control performance, notably

improving the system's adjustment time.

(3) The practical effectiveness of the method was validated on an agricultural robot, where it achieved the shortest rise time of 4.43 s and the most accurate and stable steering control with a steady-state average of 102.9 pulses and the smallest standard deviation of 1.53, outperforming both GA - PID and manual PID control.

References

- [1] ZHAO Ying, XIAO Hongru, MEI Song, et al. Current situation and development strategy of orchard mechanized production in China[J]. Journal of China Agricultural University, 2017,22(6): 116 - 127.
- [2] CHENG Chao, FU Jun, SU Hang, et al. Recent advancements in agriculture robots: benefits and challenges[J]. Machines, 2023,11(48): 1 - 24.
- [3] SHAMSHIRI R R, WELTZIEN C, HAMEED I A, et al. Research and development in agricultural robotics: a perspective of digital farming[J]. Int. J. Agric. & Biol. Eng., 2018,11(4): 1 - 14.
- [4] LI Tonghui, XIE Bin, WANG Rong. Design and experiments of electronic steer-by-wire system in electric tractor[C] // IOP Conference Series Earth and Environmental Science, 2019.
- [5] JIAO Jun, CHEN Jing, QIAO Yan, et al. Single neuron PID control of agricultural robot steering system based on online identification[C] // 2018 IEEE Fourth International Conference on Big Data Computing Service and Applications, 2018: 193 - 199.
- [6] XU Shaobing, PENG Huei. Design, analysis, and experiments of preview path tracking control for autonomous vehicles[J]. IEEE Transactions on Intelligent Transportation Systems, 2020, 21(1): 48 - 58.
- [7] QUN Ren. Intelligent control technology of agricultural greenhouse operation robot based on fuzzy PID path tracking algorithm [J]. Inmateh-Agricultural Engineering, 2020, 62(3): 181 - 190.
- [8] JIA Weikuan, TIAN Yuyu, DUAN Huichuan, et al. Autonomous navigation control based on improved adaptive filtering for agricultural robot [J/OL]. International Journal of Advanced Robotic Systems, 2020. <https://doi.org/10.1177/1729881420925357>.
- [9] DING Youchun, XIA Zhongzhou, PENG Jingye, et al. Design and experiment of the single-neuron PID navigation controller for a combine harvester[J]. Transactions of the CSAE, 2020, 36(7): 34 - 42.
- [10] DING Youchun, HE Zhibo, XIA Zhongzhou, et al. Design of navigation immune controller of small crawler-type rape seeder [J]. Transactions of the CSAE, 2019, 35(7): 12 - 20.
- [11] LIU Jinkun. Advanced PID control and its Matlab simulation[M]. Beijing: Electronic Industry Press, 2004.
- [12] SEN Yue, HE Siwei, LIU Hui, et al. Modeling and control of self-steering electric chassis structure of high clearance sprayer [J]. Transactions of the Chinese Society for Agricultural Machinery, 2020, 51(11): 385 - 392.
- [13] ZHANG Jian, LIU Li, LI Huanzhou, et al. PID tuning based on improved quantum genetic algorithm[C] // 2013 Sixth International Symposium on Computational Intelligence and Design, 2013.
- [14] SONG Hongfa, JIN Qibing, ZHAO Mei. PID parameter tuning strategy based on improved genetic algorithm[J]. Journal of Beijing University of Chemical Technology (Natural Science Edition), 2005, 32(2): 101 - 103.
- [15] HAMEDANI Mohammad Hossein, ZEKRI Maryam, SHEIKHOLESLAM Farid. Adaptive impedance control of uncertain robot manipulators with saturation effect based on dynamic surface technique and self-recurrent wavelet neural networks [J]. Robotica, 2019, 37(1): 161 - 168.
- [16] LI Yining, ZHOU Wei, SONG Shujie, et al. Design of experimental prototype of flexible chassis used in greenhouse[J]. Transactions of the CSAE, 2017, 33(19): 41 - 50.
- [17] BONYADIB Mohammad Reza, MICHALEWICZ Zbigniew. Particle swarm optimization for single objective continuous space problems: a review[J]. Evolutionary Computation, 2017, 25(1): 1 - 54.
- [18] FAROKHI A, MAHMOODABADI M J, JAMES L. Optimal fuzzy inverse dynamics control of a parallelogram mechanism based on a new multi-objective PSO[J]. Cogent Engineering, 2018, 5(1): 1 - 20.
- [19] WEN Sheng, ZHANG Quanyong, DENG Jizhong, et al. Design and experiment of a variable spray system for unmanned aerial vehicles based on PID and PWM control[J]. Applied Sciences, 2018, 8(12): 2482.
- [20] ZHANG Tiemin, HUANG Han, HUANG Penghuan, et al. Design and test of drive and control system for electric wheeled mobile car[J]. Transactions of the CSAE, 2014, 30(19): 11 - 18.
- [21] DU Xiaoping, ZHAO Kaiqi. Design and implementation of mobile robot motion control system based on PID[J]. Journal of Communications, 2016, 37(Supp.1): 43 - 49.

基于粒子群优化与遗传算法融合的农业机器人 PID 转向控制方法

赵龙莲^{1,2} 张佳创^{1,2} 李 梅^{1,2} 董志城^{1,2} 李军会^{1,2}

(1. 中国农业大学信息与电气工程学院, 北京 100083; 2. 现代精细农业系统集成研究教育部重点实验室, 北京 100083)

摘要: 为解决农业机器人在运动过程中出现的转向不稳定和滞后问题,提出了一种粒子群优化(PSO)与遗传算法(GA)融合的 PID 控制方法。该融合算法利用 PSO 的快速优化能力,对 GA 的种群筛选环节进行优化。Simulink 仿真结果表明,融合算法的适应度函数收敛速度加快,系统响应调整时间缩短,超调量几乎为零。随后,该算法被应用于农业机器人在不同场景下的转向测试。在对农业机器人的转向系统进行建模后,无负载悬浮状态下的转向测试结果表明,与人工试错 PID 控制和基于 GA 的 PID 控制相比,基于融合算法的 PID 控制减少了系统的上升时间、响应调整时间和超调量,提高了系统的响应速度和稳定性。实际道路转向测试结果表明,基于融合算法的 PID 控制响应上升时间最短,约为 4.43 s。当目标脉冲数设为 100 时,稳态调节阶段实际均值约为 102.9,在 3 种控制方法中最为接近目标值,同时超调量也有所减小。不同场景状态下的转向测试结果表明,本研究提出的基于融合算法的 PID 控制具有良好的抗干扰能力,能够适应环境和负载的变化,提高控制系统的性能。该方法在农业机器人的转向控制中效果显著,为其他机器人的精确转向控制提供了参考。

关键词: 农业机器人转向; PID 控制; 粒子群优化算法; 遗传算法

基金项目: 国家自然科学基金项目(31772064)