J. COMBIN. MATH. COMBIN. COMPUT. 127b (2025) 579--595

Journal of Combinatorial Mathematics and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on multilevel genetic algorithm in optimization of combined electrical drive system for industrial robots

Xiaoying Yan[1, ✉]

[1] *College of Engineering, Caofeidian College of Technology, Tangshan, Hebei, 063200, China*

ABSTRACT

In the joint electrical drive system of industrial robots, the optimization and improvement of robot motion control is one of the hotspots of current research, and this paper proposes a method of optimizing the joint electrical drive control of robots using multilevel genetic algorithm. An improved PID control method is used to fuzzify the robot motion, and the robot trajectory fuzzy PID controller is optimized according to the idea of multilevel genetic algorithm. The rise time of each joint of the robot is about 5ms, 55ms, and 75ms, respectively, and the overshooting amount is smaller, and the optimized joint electrical drive system of the industrial robot is more stable in speed control in both the acceleration and deceleration phases, and shows a good dynamic control capability of the motion. It can be seen that the work in this study effectively optimizes the control performance of the industrial robot drive system using multilevel genetic algorithm.

Keywords: multilevel genetic algorithm, fuzzy PID, industrial robot, drive system control

1. Introduction

In recent years, the robotics industry has developed rapidly and has become a popular and promising industry, which is widely used in complex and dangerous working environments, replacing human labor to do boring assembly line work [1-3]. With the rising cost of employment in China and the increasing demand for processing quality by enterprises, as well as the current lack of labor efficiency, industrial robots are in increasing demand. The application of industrial robots in factories is not only limited to traditional applications such as palletizing, painting and other occasions that do not require high trajectory tracking, but also the precision machining of workpieces, human-machine assembly collaboration, etc., which put forward higher requirements on the trajectory tracking accuracy of industrial robots [4-7]. In order for the robot to complete the specified action, it is necessary to install transmission devices at each joint, i.e., the drive system of the robot [8-10]. The transmission methods mainly include hydraulic transmission, pneumatic transmission, electric transmission, and the drive

✉ Corresponding author.
E-mail address: Yanxiaoying L L@163.com (X. Yan).
Received 05 November 2024; Revised 20 December 2024; Accepted 15 March 2025; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-033

system of the whole industrial robot can be obtained by any combination of the above three methods [11-12].

The performance of the robot servo drive system is related to the electromechanical parameters and control algorithms. The irrational selection and design of the control parameters of the servo motor and the mechanical parameters of the drive components will affect the dynamic performance of the system. Under external disturbances such as sudden load changes, the traditional PID control is difficult to meet the performance requirements, and the application of modern control algorithms, such as sliding mode control and fuzzy control, to the servo motor can enhance the system's anti-interference capability [13-17]. Enhancing the tracking accuracy of robot servo systems requires not only the joint trajectory and joint moment data of industrial robots in specific scenarios as inputs, but also the kinematic and dynamic modeling of robots as the basis for servo system control [18-20]. The control of robots is not only the control of individual servo motors, the mainstream high-precision robots in the market are based on the strategy of kinematic modeling plus dynamics feed-forward, and a variety of control rates designed based on dynamics modeling [21-23]. Research on the servo drive system composed of core components is of great significance to promote China's robotics technology and enhance the international competitiveness of robots [24-25].

This paper optimizes the control performance of a combined electrical drive system for industrial robots. The paper firstly introduces the fuzzy PID control method and the implementation process of multilevel genetic algorithm. The characteristic parameters of the fuzzy PID controller are introduced to simplify the fuzzy controller design process. A hybrid coding strategy is proposed to embed the hybrid coding of feature parameters and design parameters into the chromosome gene loci of the multilevel genetic algorithm, which improves the efficiency of optimal control of industrial robots and the result of optimization search. The response speed of this paper's method and the control optimization effect of the industrial robot drive system are examined through an arithmetic example analysis, and then the implementation effect of this paper's method is simulated by taking the pipeline robot as an example.

2. Principle of control optimization of joint electrical drive system for industrial robots

2.1. Drive system control optimization algorithm

2.1.1. Single-joint PID control. The PID control method is simple in structure, easy to compute, and robust.In the PID control method, it is a combination of three control methods, namely proportional, integral, and differential.

The PID algorithm cannot be applied to nonlinear control, but it has its unique characteristics for linear control. This control method is a feedback control and the input used in the input side part is the input of error signal which regulates the control system [26]:

$$e(t) = r(t) - y(t) \tag{1}$$

Thus input signal $u(t)$ is:

$$u(t) = k_p e(t) + k\int_0^t e(t)dt + k_d \frac{de(t)}{dt} \tag{2}$$

The conversion to a transfer function is:

$$G(S) = k_p + \frac{k_i}{s} + kds \tag{3}$$

k_p is the proportionality coefficient, k_i is the integral coefficient, and k_d is the differential coefficient.

2.1.2. Fuzzy PID control. Because the PID control algorithm can not be intelligently adjusted this feature, this topic on this basis with intelligent fuzzy control mode, so that the PID control can be allowed to carry out a parameter self-regulation process, increasing its applicability and flexibility. When fuzzy is used alone, although it has a certain degree of intelligence, its static performance is not very good, and is prone to unstable fluctuations near the error point. So when the error signal is used as an input, it is very similar to the PD algorithm alone, which is not able to neutralize the error well. So the effect of using fuzzy control method alone is also not very good. Therefore, a fuzzy PID control strategy is generated by using the combination of PID strategy and fuzzy approach, and this control strategy has some advancement.

Fuzzy PID consists of fuzzy control block diagram and PID control block diagram which is a combination of two control methods. According to the control method of industrial robots, there are two input variables, e as well as ec, and three output variables transmitted to the robot, k_p, k_i and k_d, which are different from the output variables in the traditional PID control, which are fuzzy and intelligent, and have a certain self-regulation ability [27].

2.2. Basic genetic algorithm

Genetic algorithms are widely used in the field of optimization and are suitable for solving large-scale optimization problems.

The fundamental principle of the basic genetic algorithm is “survival of the fittest”, and the chromosomes used to solve the problem evolve from generation to generation through the algorithm of “survival of the fittest” (including a series of operations such as selection, crossover and mutation, etc.), so that the individual converges to the state of adapting to the environment, and thus obtains the optimal solution of the problem to be solved. Its main features are stochasticity and adaptivity, and the operations are highly parallelized. Genetic algorithms, like other heuristic algorithms, can be computed directly on the structure without worrying about the limitations of the function in terms of continuity. Compared to other algorithms, it has a better global optimization capability and can automatically optimize the optimization space and adaptively adjust the direction of optimization that is not specified [28].

2.2.1. Mathematical model of the basic genetic algorithm. The basic genetic algorithm can be expressed as:

$$SGA = (C, E, P, M, \Phi, \Gamma, \Psi, T) \tag{4}$$

In the formula:

C --individual coding methods.

E --individual fitness evaluation function.

P --the initial population.

M --population size.

Φ --selecting the operator.

Γ -- cross operator.

Ψ --variant operator.

T --Genetic operation termination conditions.

2.2.2. Steps of the basic genetic algorithm:

1) Chromosome encoding and decoding. In the basic genetic algorithm, each individual in the population is represented by a binary number of a certain length, so the value of each gene is composed of two numbers, 0 and 1. When using the genetic algorithm to deal with the problem, the first thing to do is to assign a value to each individual in the initial population in the genetic algorithm, generally using a random assignment of each individual's genes, so as to generate the initial solution of the genetic algorithm.

(1) Coding: first of all, we need to calculate the range of the required variable, so as to get the range of the parameter corresponding to the required code $[U_1, U_2]$, so that we can calculate the length of the code and k used to represent the length of this binary code, because the length of the code is k, so this code corresponds to a total of 2^k different codes, the following is the corresponding relationship:

$$\begin{gathered} 000000\cdots0000 = 0 \rightarrow U_1 \\ 000000\cdots0001 = 1 \rightarrow U_1 + \delta \\ 000000\cdots0010 = 2 \rightarrow U_1 + 2\delta \\ \vdots \\ 111111\cdots1111 = 2^k - 1 \rightarrow U_2 \end{gathered} \tag{5}$$

Among them:

$$\delta = \frac{U_2 - U_1}{2^k - 1} \tag{6}$$

(2) Decoding: the decoding formula corresponding to an individual coded as $b_k b_{k-1} b_{k-2} \cdots b_2 b_1$ is:

$$X = U_1 + \left(\sum_{i=1}^{k} b_i * 2^{i-1}\right) * \frac{U_2 - U_1}{2^k - 1} \tag{7}$$

2) Detection and evaluation of individual fitness. Individual fitness is the relative performance of the objective function value of an individual solution in the objective function. In other words, the closer the objective function value of an individual solution is to the optimal solution, the greater its individual fitness is. The detection and evaluation of individual fitness is to calculate a probability for the fitness of each individual in the population, which is proportional to the individual fitness, so as to determine the probability of each individual to be inherited to the next generation. Where the fitness of each individual is specified to be non-negative.

3) Genetic operators. The three genetic operators of the basic genetic algorithm are represented as follows:

(1) The selection operation uses the proportional selection operator as the operator of the individual exchange operation method. The proportional selection operator uses the probability of each individual's fitness to determine the likelihood of transmitting its genes to its offspring after the individual fitness is tested and evaluated. If the fitness of individual i is f_k, then the probability that individual i will pass on its gene to its offspring in a population of M is:

$$P_i = f_i \Big/ \sum_{k=1}^{M} f_k \tag{8}$$

When the probability of individual selection is found by the selection ratio operator, the algorithm will automatically generate a uniform random number between [0, 1] to determine which individual participates in the mating. The algorithm will repeatedly select the individual with a higher probability of exchange and eliminate the individual with a lower probability of selection, if an individual has a higher probability, then his genes will expand in the population.

(2) The crossover operation uses the single-point crossover operator as the operator of the individual crossover operation method. The crossover point position on an individual is selected by randomly generating a finite number, and the two individuals with a high degree of individual fitness calculated in the selection operation are used as the crossover objects, and then the genetic codes of the two individuals at this position are exchanged with each other in the position selected by Ran to form two new individuals.

(3) The variation operation uses the basic bit variation operator or uniform variation operator as the operator of the individual variation operation method. In order to avoid the problem converging prematurely and only finding a local optimal solution, the individual with high individual fitness is then randomly selected and the binary code at its random position is randomly flipped.

4) Running parameters of the basic genetic algorithm passing algorithm. The basic genetic algorithm also needs to predetermine four running parameters in advance, namely the population size, the number of planting evolutionary generations, the crossover probability and the mutation probability.

3. Multi-level genetic algorithm based control optimization method for drive system

3.1. Genetic operations of multilevel genetic algorithms

In optimization design with genetic algorithms, genetic algorithms with multi-level structure genetic algorithms are used to build their multi-level structure system representations for genetic manipulation.

3.1.1. Coding. For a genetic algorithm to perform genetic operations, it must start with coding. Proper coding will allow the algorithm to converge better to the optimal solution. In a design with a multilevel structure, the coding of individuals is based on two factors:

1) If there is a strong mutual influence relationship between the design variables, the highest bit of the binary code corresponding to each design variable can reflect this mutual relationship between the design variables.

2) It is known from the pattern theory of genetic algorithms that a pattern with a short definition length is less likely to be destroyed when the population reproduces, so the shorter the definition length, the greater the chance that the gene pattern will be retained in the genetic operation.

In the structure of this database:

Serial number: indicates the number of each scheme in the design state space.

Layer number: indicates the position of the selected design in its state space.

Parent node: indicates the location of the previous layer of design of the selected design.

Organization number: indicates that the selected design is the organization number in the current layer.

Program number: indicates that the selected design is the first available program of the current organization.

From the above coding method, we find all the design variables in the design state space, code them, and complete the genetic coding space. It is assumed that an optimal individual is obtained after genetic manipulation:

With this genetic coding method, the representation of industrial robot systems and individuals can be solved in genetic algorithm programming.

3.1.2. Selection operations. In this paper, the ranked selection method is chosen to perform the selection operation. It mainly focuses on the relationship between the size of individual fitness. The main idea is: calculate the fitness of individuals in the group, rank the individuals according to the size of their values, and decide the probability size of the corresponding individual to be selected according to this ranking [29]. The specific steps are as follows:

1) Calculate the fitness value of each individual in the population and rank the individuals according to their values from largest to smallest.

2) Make a probability assignment table and assign the probability values in the table to the corresponding individuals according to the above ordering.

3) Perform a selection operation by using the assigned probabilities as the selection probabilities of its individuals.

3.1.3. Cross-operation. In multilevel structure genetic algorithms, since genes with multilevel structure are encoded, if the genes at the higher level change, the genes at their substructure level are bound to change as well. Therefore, it is necessary to define and describe the crossover operation into the genes encoded with multilevel genes.

If two individuals are randomly paired into a group to perform a crossover operation to exchange the corresponding genes between the two individuals, and if the substructure genes of one of the individuals are exchanged with the substructure genes of the other, then all the corresponding exchange operations will be performed accordingly in the next generation. The steps of the crossover operation are shown below:

Step 1 Randomly match two individuals into a crossover operation group, start the crossover operation from its highest level first, and select a group of genes to perform a crossover operation with probability P_c.

Step 2 compares group i genes $(i = 1, 2, 3\ldots, n)$ and if the gene individual 1's gene b_i is different from individual 2's gene b_i, a crossover operation is performed on b_i with probability P_c. This is a simple crossover operation in which the individual group b_i bits are exchanged while their next level of substructures are exchanged with them.

Step 3 If the crossover operation in step 2 is not performed or if b_i the genes are identical, a plus 1 operation is performed on i. If $i <= n$, step 2 is returned, and when $i > n$, the crossover operation ends.

In summary, a group of individuals with multilevel gene coding undergoes a crossover operation by first pairing groups of crossover individuals and performing the crossover operation from top to bottom, starting from the highest level of the genome.

3.1.4. Variational operators. The mutation operation is performed in a multilevel structural genetic algorithm, where the mutation starts at the highest level of structure and acts on its subgenomes in the same way as the crossover operation. The steps of the mutation operation are as follows:

Step 1 Examine the highest level substructure genome to decide whether to perform mutation or not with mutation probability P_m. If the mutation operation is performed, step 2 is executed. if the mutation operation is not performed, step 3 is executed.

Step 2 False design chromosome N_i contains a sub-gene structure, turn to step 1 while checking the sub-gene structure. If there is no subgene structure, perform step 5.

Step 3 If no mutation operation is performed, a selection operation is performed on the substructure i, and a genome is randomly selected as the gene for the substructure.

Step 4 Recombination is performed on the substructure genes.

Step 5 i increases by 1 when $i > n$. The mutation operation ends. When $i <= n$, step 1 is returned.

3.2. Drive system optimization method based on multilevel genetic algorithm

3.2.1. Characterization parameters of the fuzzy PID controller. The design parameters of the fuzzy PID controller are specifically the location of the center vertex of the affiliation function of the input and output variables and the output rulethen result obtained from each fuzzy ruleif condition. In this paper, the characteristic parameters of the fuzzy controller are introduced with the aim of simplifying the fuzzy controller design process under the premise of characterizing the design parameters, including the characteristic parameters of the affiliation function and the characteristic parameters of the fuzzy rules.

For the characteristic parameters of the affiliation function, according to the characteristics of the affiliation function design, the sparsity relationship between the position of the center vertex of the unilateral triangular affiliation function and zero can be expressed by equation (9):

$$X_{mi} = \left| \frac{i}{\left(\frac{k-1}{2}\right)} \right|^{P_m} \left(i = 1, \ldots, \frac{k-1}{2} \right) \tag{9}$$

where X_{mi} is the sparsity of the center vertex position of the triangle affiliation function with zero. k is the number of fuzzy sets of the affiliation function, where $k = 7$. P_m is the exponent of the power function, and the value of P_m will affect the sparsity distribution of each center value of the affiliation function. From equation (9), a stretch factor G can be further used to describe the degree of sparsity of the vertices of the triangular affiliation function:

$$X_i = G \cdot X_{mi} \left(i = 1, \ldots, \frac{k-1}{2} \right) \tag{10}$$

By defining P_m and G as the characteristic parameters of the subordinate function, the design parameter X_i of the subordinate function can be generated by P_m and G, i.e., the three trigonometric vertex positions required to specify the design of the subordinate function can be generated by two characteristic parameters.

For the characteristic parameters of the fuzzy rules, the output rules of ΔK_P and ΔK_D can be divided into several regions with the same rule results, and the number of regions depends on the number of fuzzy sets of the affiliation functions of the output variables k. The fuzzy rule of ΔK_P is used as an example to illustrate the method of region division using the phase plane. Firstly, the phase plane XOY is established based on the fuzzy set values {NL, NM, NS, ZE, PS, PM, PL} of the input variables e, ec corresponding to ΔK_p, respectively, and the intersection points of the different fuzzy set values of the X, Y axes are the grid points, which can be seen to represent the IF portion of the fuzzy rule, and the distances of the grid points from the X, Y axes characterize the X_{mi} value of the position of the central vertex of the triangular affiliation function corresponding to the e, ec respective values of the different fuzzy sets. The coordinates of the split points are calculated as follows:

$$\begin{cases} x_{st} = L \cdot sign(i) \cdot \left| \dfrac{i}{\left(\dfrac{k-1}{2}\right)} \right|^{P_s} \left(i = -\dfrac{k-1}{2}, \dots, 0, \dots, \dfrac{k-1}{2} \right) \\ y_{st} = x_{st} \cdot \tan(\theta_s) \end{cases} \tag{11}$$

$$L = \begin{cases} sign(\tan(\theta_s)) & (|1/\tan(\theta_s)| \geq 1) \\ 1/\tan(\theta_s) & (|1/\tan(\theta_s)| < 1) \end{cases} \tag{12}$$

$$sign(i) = \begin{cases} 1 & (i \geq 0) \\ -1 & (i < 0) \end{cases} \tag{13}$$

where (x_{ii}, y_{ii}) is the coordinates of the segmentation points, L is the range parameter used to restrict the segmentation points to fall within the phase plane, sign is the sign function, θ_s is the angle of counterclockwise rotation of the line connecting the segmentation points around the origin with X the positive semiaxis, and P_s is the power function index. Finally, a set of mutually parallel straight lines are generated by the perpendicular bisectors between two neighboring split points, and these parallel lines realize the region division of the phase plane, and at the same time specify that the results of fuzzy output rules corresponding to the grid points within the same region are the same, i.e., the fuzzy rules then have the same results.

3.2.2. Mixed coding. In this paper, a hybrid coding strategy is proposed, i.e., hybrid coding of feature parameters and design parameters to chromosome gene positions, with the aim of shortening the length of individual chromosomes in direct coding and improving the optimization efficiency and optimization search results. The hybrid coding method for optimizing the fuzzy PID controller for trajectory tracking of articulated robots by genetic algorithm is as follows: the affiliation functions of the 2 input variables e and ec, and the 3 output variables K_P, K_I, K_D are coded by the characteristic parameters P_w and G, which correspond to the gene loci such as $a_1 - a_5$ and $a_6 - a_{10}$. The fuzzy rules of the ΔK_D and ΔK_P output variables are coded by the characteristic parameters P_s and θ_s, which correspond to the gene loci $a_{11} - a_{12}$ and $a_{13} - a_{14}$. The fuzzy rule of ΔK_I is directly coded and the numbers 1-7 are used to denote the NL-PL in the fuzzy rule, respectively, then the fuzzy rule design parameter of ΔK_I can be expressed as 50 digits corresponding to the gene loci $a_{15} - a_{63}$. Such a hybrid coding method eventually forms the individual chromosome as a row vector shaped like $[a_1, a_2, \dots, a_6]$. If the fuzzy PID controller hybrid coding method is used, a total of 65 gene positions are required, which is a significant reduction in the length of the chromosome formed from direct coding.

3.2.3. Multi-level genetic algorithm optimization process based on hybrid coding. The main steps of the process of multilevel genetic algorithm to optimize the fuzzy PID controller for trajectory tracking of articulated robots are:

Step 1: Generate initial population: according to the range of values of the fuzzy controller's characteristic parameters and design parameters, adopt the hybrid coding method of characteristic parameters and design parameters to randomly generate N individual chromosomes $[a_1, a_2, \dots, a_s]$ as the current population.

Step 2: Update the population: update the current population as the parent population.

Step 3: Update the affiliation function and fuzzy rules: read the coding of each gene position of chromosome $j (j = 1, \dots, N)$ in the current population and update the affiliation function according to the coding $a_1 - a_{10}$ to generate new input variables e, ec and output variables $\Delta K_P, \Delta K_I, \Delta K_D$ of the affiliation function. Generate new fuzzy rules for output variable $\Delta K_P, \Delta K_I, \Delta K_D$ according to encoding $a_{11} - a_{63}$ and update the fuzzy rules.

Step 4: Calculate the objective function value: run the fuzzy PID controller to complete the trajectory tracking control simulation, obtain the deviation of the simulated trajectory of the robot from the desired trajectory at the moment of t, $e(t)$, and calculate the value of the objective function J. Select the ITAE criterion to design the objective function of the controller:

$$J = \int_0^T t \cdot |e(t)| \, dt \to \min \tag{14}$$

where $e(t)$ is the deviation of the simulated trajectory of the robot from the desired trajectory at the moment of t, and T is the total time to run the fuzzy PID controller to complete a trajectory tracking control simulation. Then the value of the fitness function of the multilevel genetic algorithm is $F = 1/J$.

Step 5: Judge whether to traverse N individual chromosomes: if the number of individual chromosomes j does not satisfy $j > N$, then +1, and go to step 3. If it satisfies $j > N$, then go to step 6.

Step 6: Determine whether to terminate the genetic optimization algorithm: determine whether the change in the value of the objective function is less than the set threshold or whether the number of genetic generations reaches the set value, if it is satisfied, then end the genetic optimization algorithm. If not satisfied, proceed to step 7.

Step 7: Genetic operation to generate the next-generation subpopulation: according to the value of the fitness function transformed from the value of the objective function of each individual in the current population, a new population can be obtained by the basic operation of multilevel genetic algorithm of selection, replication, crossover and mutation for all individuals in the population, and go to step 2.

4. Arithmetic experiments and simulations

4.1. Example analysis

4.1.1. Response speed analysis. The binary coding method is used to represent the proportional coefficient (k_p), integral coefficient (k) and differential coefficient (k) of the PID controller with a binary code of 10 bits in length, and the decoding process is as follows:

When decoding, the 30-bit binary code is cut off into three 10-bit binary codes, and then they are converted into the corresponding decimal integer code, which is recorded as y_1, y_2, y_3.

Assuming that the value range of the PID parameter variable x_i is $[-a_{\min}, +a_{\max}]$, the formula for converting y_i to variable x_i is:

$$x_i = a_{\min} + \frac{y_i}{1024} \times (a_{\max} - a_{\min})(i = 1, 2, 3) \tag{15}$$

where: x_1, x_2, x_3 - proportional coefficient (k), differential coefficient (k) and integral coefficient (k) of the PID controller.

In order to obtain more ideal control dynamic characteristics, the time integral of the absolute value of the error is used as the adaptive objective function for parameter selection. Throughout the optimization process, the objective function is made to reach the minimum value for the purpose. In order to prevent excessive control energy, the square term of the controller output is also added to the objective function. The objective function is shown in the following example:

$$f = \int_0^\infty (b_1 |e(t)| + b_2 u^2(t)\, dt + b_3 \cdot t) \tag{16}$$

In the formula:

$e(t)$ --System deviation value.

$u(t)$ --Output control quantity for PID control.

b_1, b_2, b_3 --Weights.

t_a --Rise time.

The initial population is selected by computer-generated random numbers uniformly distributed between 0 and 1. The number of individuals per generation is set to be 50, and after 100 generations of evolution, the optimal PID parameters for each joint are obtained, as shown in Table 1, and Figures 1 to 3 show the experimental results of the control of the three robot joints.

From the experimental results, it can be seen that the rise time of each joint is about 5ms, 55ms, and 75ms, respectively, which has a fast response speed, and compared with the traditional PID control, the overshoot is small, and the dynamic characteristics are good, which basically meets the requirements of industrial operations.

Table 1. PID parameter optimization result

	Proportional coefficient(k_p)	Integral coefficient (k_i)	Differential coefficient (k_d)
Joint 1	1.501	0.057	0.008
Joint 2	14.897	0.285	0.420
Joint 3	0.597	0.002	0.092

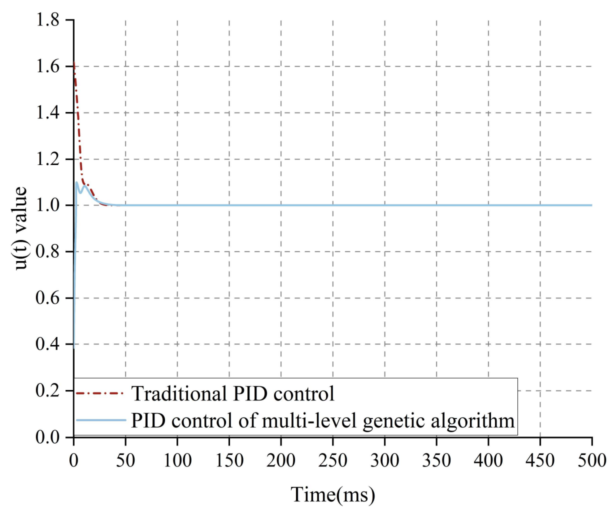


Fig. 1. Test results of joint 1

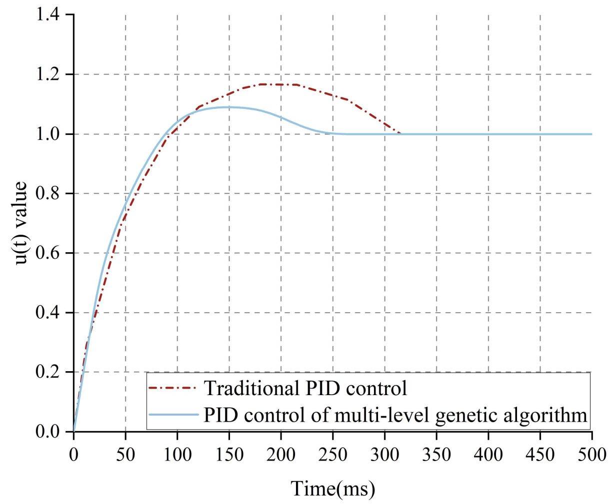


Fig. 2. Test results of joint 2

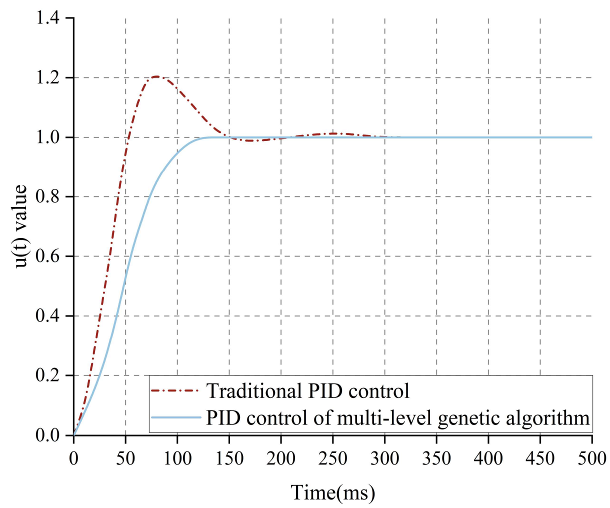
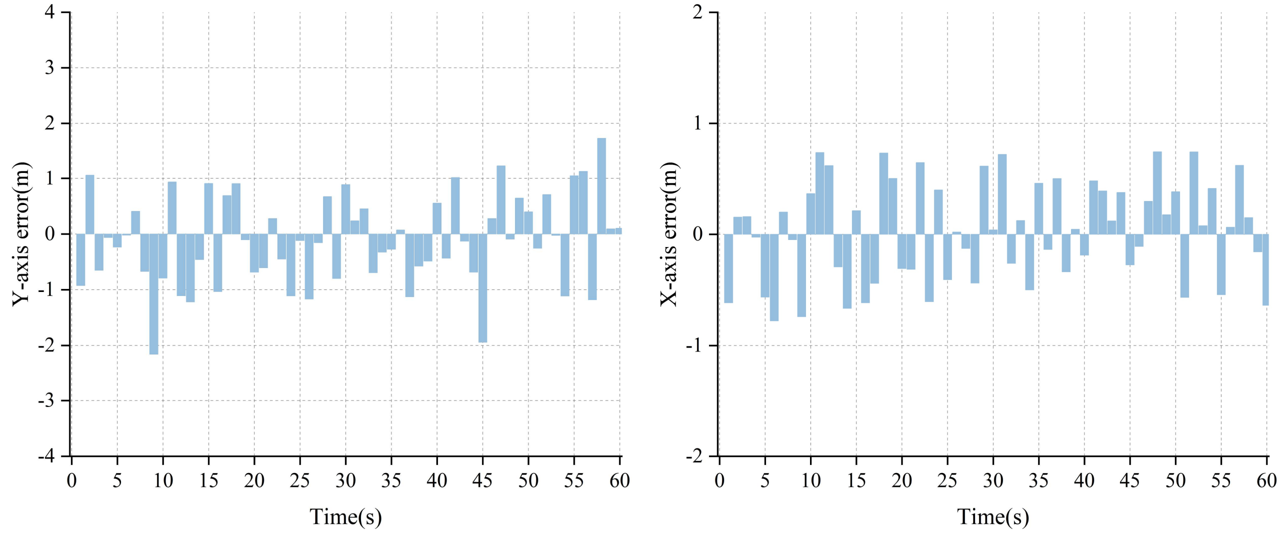


Fig. 3. Test results of joint 3

4.1.2. Effectiveness of robot drive system control optimization. In order to verify whether the industrial robot motion control system designed in this paper can optimize the control effect of robot operation, the system performance test is carried out for motion control direction and speed control respectively. The robot operating environment is simulated inside the laboratory and all data are recorded in the SQLServer2015 database.

1) Industrial robot motion direction control test. The relevant personnel first simulate the test environment, set obstacles in the inspection robot's scheduled movement route, control the robot from (0, 0), according to 1, 2, 3, 4 counterclockwise direction forward until returning to (0, 0) coordinates. The actual robot travel route is recorded as a two-dimensional image to verify whether the method of this paper is reasonable and effective for robot motion control optimization. Fig. 4(a) and (b) show the Y and X axis error plots of the motion path, respectively.

It can be seen that the robot generates some errors when it transfers from the operation of one location to the next, and the angle also generates deviation. Overall, the industrial robot did not collide with any obstacles during the test, indicating that the robot motion control optimization system designed in this paper is suitable for industrial operations.



(a) Y-axis error (b) X-axis error

Fig. 4. Robot travel error result

2) Speed control test. In order to verify whether the method of this paper is equally effective in controlling the motion speed of industrial robots, a comparison test is conducted with the two methods of double-closed-loop method and the central processor (CPU) architecture.The results of the comparison of the speed control of the three methods are shown in Fig. 5.

Among the three methods, only the method in this paper always controls the robot's motion speed in a smooth state, no matter in the acceleration or deceleration phase. This is because the method in this paper can realize smooth operation for several phases of the robot's motion speed, reduce the route deviation caused by sudden acceleration and deceleration, and thus improve the working efficiency and control accuracy of industrial robots.

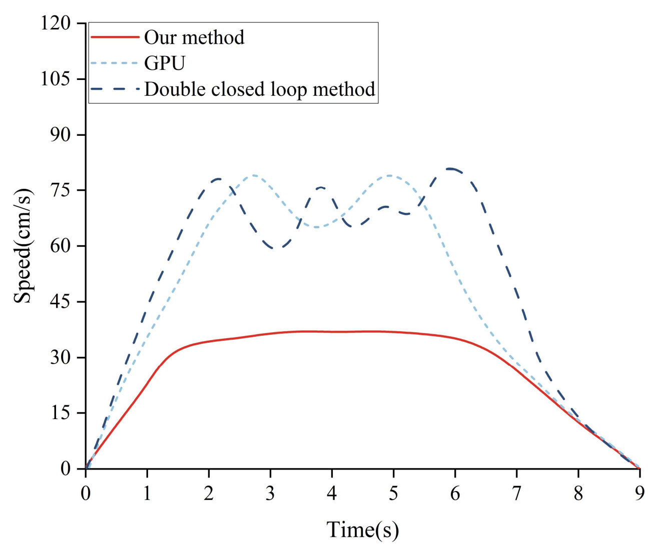


Fig. 5. The comparison of the result of speed control

4.2. Simulation

4.2.1. Control performance simulation of adaptive control. Industrial pipeline desilting robot as an example of algorithm simulation simulation. STM32 and the host computer serial debugging software through the USB serial communication, baud rate set to 115200, parity bit is none, the data bit is 8 bits, the stop bit is 1 bit, and the software in the serial communication C program to be set to exactly the same. The computer automatically assigns COM4 port for the microcontroller device, and the serial port debugging software should select COM4 at the port to complete the configuration. According to the data displayed by the serial port software to organize, simulation simulation data as Table 2.

It can be seen that when the pressure value is increasing, the speed of the pipeline robot is also increasing trend. This shows that when the robot faces the increased pressure, that is, when the sludge content in the pipe increases, it increases the output power of the pipeline robot motor to show the increase in speed, and in this way enhances the scraping ability to complete the desludging work. When the pressure value G gradually increases to be greater than G1, the robot enters the state of working condition 2 from the state of working condition 1, so the rotational speed occurs a large span, then the pressure value G continues to increase, the rotational speed stabilizes up and down within the range of 50r-min^{-1}, and the pressure value continues to increase upwards, and when the pressure value is more than G3, the core controller of the robot judges that the current When the pressure value exceeds G3, the robot core controller judges that the current pressure value exceeds the upper working limit, and sends a stop command in order to protect the motor.

And when the pressure value continues to increase, the speed of the robot walking mechanism tends to decrease. This indicates that when the pressure increases, i.e., when the sludge content in the pipe increases, the scraping capacity is increased by adjusting the speed of the pipe robot in order to clear the sludge more efficiently, but considering the actual situation, in order to make the pipe robot mix and scrape the sludge sufficiently, the walking mechanism reduces the speed to complete the coordinated operation with the pipe robot. Experiments show that the design idea is completed, the desired effect is achieved, adaptive control is realized, and the motor is protected.

Table 2. Experimental data

Pressure (N)	Rotational speed ($r \cdot min^{-1}$)	Travel speed ($m \cdot min^{-1}$)	Pressure (N)	Rotational speed ($r \cdot min^{-1}$)	Travel speed ($m \cdot min^{-1}$)
0.00	0	0.00	13.09	51	2.49
0.26	16	3.56	14.02	48	2.50
0.42	21	3.62	15.05	45	2.48
0.65	25	3.61	16.18	50	2.49
0.83	23	3.58	17.02	49	2.51
1.01	24	3.58	18.22	47	2.48
1.22	18	3.62	21.22	77	1.99
1.46	19	3.63	23.19	80	2.00
1.61	16	3.57	25.04	80	1.98
1.88	20	3.59	27.06	75	2.01
1.99	18	3.63	29.81	73	2.02
2.28	18	3.59	30.90	82	2.00
2.45	17	3.60	33.11	82	1.98
2.67	22	3.59	35.22	77	1.97
2.90	24	3.60	37.46	77	1.99
3.03	20	3.59	39.26	78	2.03
3.23	22	3.60	41.41	78	2.00
3.54	45	2.51	43.07	78	1.99
4.03	47	2.47	45.14	82	2.01
5.08	51	2.52	47.12	73	1.98
6.11	44	2.48	49.29	76	1.96
7.09	49	2.48	51.24	80	1.99
7.98	50	2.49	52.05	0	0
9.04	46	2.51	55.16	0	0
10.12	45	2.47	57.32	0	0
11.14	51	2.50	59.87	0	0
12.01	51	2.49	60.05	0	0

4.2.2. Fuzzy PID performance simulation based on multilevel genetic algorithm. When conducting experiments comparing the two control optimization algorithms, only the speed control of the robot was experimented for the convenience of the experiment. The speed target of the experiment is 80r-min^{-1} of the speed of the dredging mechanism in Case 3, and the value displayed by the digital manometer can be controlled in the G2 and G3 intervals at this time. Two groups of simulation experiments are carried out, the first group uses the traditional PID control algorithm for speed regulation, and the second group uses the fuzzy PID control algorithm optimized based on the multi-level genetic algorithm to carry out the experiments. The real data is shown in Fig. 6.

Since the rise time for the two algorithms to reach the preset speed during speed regulation is very short in the actual experiment, this experiment is prepared to compare and analyze the data from the aspect of stability. Compared with the traditional PID, the optimized fuzzy PID control algorithm has less fluctuation of the actual speed up and down in the set value during speed regulation, and the stability is relatively good. By calculating the average deviation, it can be seen that the average deviation of the traditional PID is 1.87, and the average deviation of the optimized fuzzy PID is 0.65. The larger the average deviation is, the worse the performance of keeping the speed stable at a certain value during speed control.

Through simulation, it can be seen that the optimized fuzzy PID control algorithm is more stable than the traditional PID control in terms of control effect. Therefore, the fuzzy PID optimized based on multilevel genetic algorithm can be used as an algorithm for speed adaptive control of industrial robots.

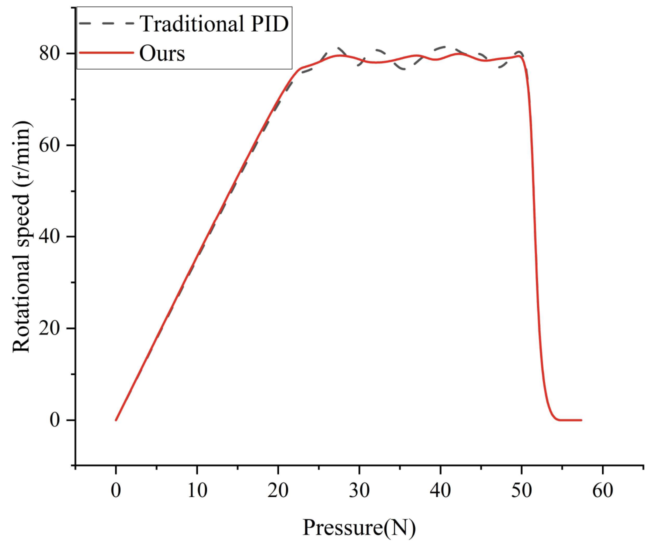


Fig. 6. Experimental data of PID

5. Conclusion

This study focuses on optimization of fuzzy PID control algorithm using multilevel genetic algorithm to optimize the control effect of joint electrical drive system of industrial robots. In the experiment, the rise time of each joint of the robot is about 5ms, 55ms, and 75ms respectively, and the overshooting amount is small compared with the traditional PID control method, which can guarantee the requirements of industrial operation. There is a certain error in the robot's motion path, but it can still avoid all the obstacles, and the robot optimized by the method of this paper has a smoother speed control in the acceleration and deceleration phases. Taking the industrial pipeline dredging robot as an example to simulate the algorithm, the optimized pipeline robot has the ability to dynamically control the travel speed, and can maintain a smooth speed. This proves that the method in this paper can better optimize the control efficiency of the robot drive system and provide convenience for industrial production.

Funding

This work was supported by Hebei Provincial Department of Education Fund Project (2205392).

References

[1] Tan, M., Liu, Z., & Wang, P. (2024, June). Application and Optimization of Robot Systems in Industrial Production Line. In 2024 Second International Conference on Inventive Computing and Informatics (ICICI) (pp. 701-706). IEEE.

[2] Ivanov, V., Andrusyshyn, V., Pavlenko, I., Pitel', J., & Bulej, V. (2024). New classification of industrial robotic gripping systems for sustainable production. Scientific Reports, 14(1), 295.

[3] Reyhani, M., Marko, L., Janisch, G., & Kugi, A. (2024). Real-time observer designs for elastic-joint industrial robots: Experimental comparison and new strategies. Mechatronics, 99, 103140.

[4] Kouritem, S. A., Abouheaf, M. I., Nahas, N., & Hassan, M. (2022). A multi-objective optimization design of industrial robot arms. Alexandria Engineering Journal, 61(12), 12847-12867.

[5] Ge, L., Chen, J., Li, R., & Liang, P. (2017). Optimization design of drive system for industrial robots based on dynamic performance. Industrial Robot: An International Journal, 44(6), 765-775.

[6] Jiang, P., Wang, Z., Li, X., Wang, X. V., Yang, B., & Zheng, J. (2023). Energy consumption prediction and optimization of industrial robots based on LSTM. Journal of Manufacturing Systems, 70, 137-148.

[7] Liu, X. J., Han, G., Xie, F., Meng, Q., & Zhang, S. (2018). A novel parameter optimization method for the driving system of high-speed parallel robots. Journal of Mechanisms and Robotics, 10(4), 041010.

[8] Yan, X., Yan, X., & Zhang, K. (2018). An Optimization Algorithm for Direct Torque Control of Industrial Robot Servo Motors. International Journal of Simulation--Systems, Science & Technology, 19(5).

[9] Padilla-Garcia, E. A., Rodriguez-Angeles, A., Resendiz, J. R., & Cruz-Villar, C. A. (2018). Concurrent optimization for selection and control of AC servomotors on the powertrain of industrial robots. IEEE Access, 6, 27923-27938.

[10] He, Y., Mei, J., Fang, Z., Zhang, F., & Zhao, Y. (2018). Minimum energy trajectory optimization for driving systems of palletizing robot joints. Mathematical Problems in Engineering, 2018(1), 7247093.

[11] Xie, Z., Xie, F., Liu, X. J., Wang, J., & Shen, X. (2019). Parameter optimization for the driving system of a 5 degrees-of-freedom parallel machining robot with planar kinematic chains. Journal of Mechanisms and Robotics, 11(4), 041003.

[12] Xu, B., Li, S., Yuan, L., Yu, H., & Li, Y. (2024). Model Prediction and Optimization of Belt Drive System Vibration Suppression Based on Industrial Robot Joints. Journal of Electrical Systems, 20(3), 64-74.

[13] Yoo, S., Jang, J., Lee, J. K., & Park, J. W. (2020). Optimal robot motion for accelerated life testing of a 6-DoF industrial robot. Applied Sciences, 10(21), 7459.

[14] Ji, T., Wei, H., Wang, J., Tian, S., Yao, Y., & Hu, S. (2023). Research into the Beetle Antennae Optimization-Based PID Servo System Control of an Industrial Robot. Mathematics, 11(19), 4066.

[15] Ai-Juan, J. I. N. (2019). Direct torque control system of induction motor based on space vector pulse width modulation. Automation in Petro-Chemical Industry, 55(2), 0.

[16] Ou, G. (2024, August). Design of servo driver for industrial robots based on TMS320F28335. In Ninth International Conference on Electromechanical Control Technology and Transportation (ICECTT 2024) (Vol. 13251, pp. 168-173). SPIE.

[17] Bilancia, P., Monari, L., Raffaeli, R., Peruzzini, M., & Pellicciari, M. (2022). Accurate transmission performance evaluation of servo-mechanisms for robots. Robotics and Computer-Integrated Manufacturing, 78, 102400.

[18] Bien, D. X. (2023). Investigation of Driving Torques at the Joints of Industrial Robot Arms Based on the Topology Optimization Technique. Journal of Applied and Computational Mechanics, 9(3), 820-833.

[19] Zhang, L. X., Meng, X. J., Ding, Z. J., & Han, H. X. (2023). Simulation-based reliability design optimization method for industrial robot structural design. Applied Sciences, 13(6), 3776.

[20] Singh, G., & Banga, V. K. (2022). Kinematics and trajectory planning analysis based on hybrid optimization algorithms for an industrial robotic manipulators. Soft Computing, 26(21), 11339-11372.

[21] Tao, J., Ye, B., Xie, Y., Tang, X., & Song, B. (2018, July). Dynamic modeling and load identification of industrial robot using improved particle swarm optimization. In 2018 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM) (pp. 75-80). IEEE.

[22] Tang, S., Liu, G., Lin, Z., & Li, X. (2022). Multi-station test scheduling optimization method for industrial robot servo system. Journal of Ambient Intelligence and Humanized Computing, 1-17.

[23] Yang, Z., Cui, W., Zhang, W., Wang, Z., Zhang, B., Chen, Y., ... & Hu, W. (2023, June). A new performance optimization method for linear motor feeding system. In Actuators (Vol. 12, No. 6, p. 233). MDPI.

[24] Nicolescu, A., Avram, C., & Ivan, M. (2014). Optimal servomotor selection algorithm for industrial robots and machine tools NC axis. Proceedings in Manufacturing Systems, 9(2), 105-114.

[25] Olszewski, M. (2020). Modern industrial robotics. Pomiary Automatyka Robotyka, 24(1), 5-20.

[26] Maoliang Jian,Shuai Zhu,Wangyu Du,Lianqiao Yang & Jianhua Zhang. (2025). Design of a temperature control system for transient thermal measurement utilizing the backpropagation PID algorithm. International Journal of Thermal Sciences109654-109654.

[27] Yongjun Zhang & Xinqing Xiao. (2024). Fuzzy PID control system optimization and verification for oxygen-supplying management in live fish waterless transportation. Information Processing in Agriculture(4),421-437.

[28] Liu Tian-Wei,Bai Jiang-Bo,Fantuzzi Nicholas,Bu Guang-Yu & Li Dong. (2022). Multi-objective optimisation designs for thin-walled deployable composite hinges using surrogate models and Genetic Algorithms. Composite Structures.

[29] A J Sobey & P A Grudniewski. (2018). Re-inspiring the genetic algorithm with multi-level selection theory: multi-level selection genetic algorithm. Bioinspiration & Biomimetics(5),056007.