Article

A PID Parameter Tuning Method Based on the Improved QUATRE Algorithm

Zhuo-Qiang Zhao [1], Shi-Jian Liu [2] and Jeng-Shyang Pan [1,3,*]

1 Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China; zzq843460995@163.com
2 Institute of Artificial Intelligence, Fujian University of Technology, Fuzhou 350118, China; liusj2003@csu.edu.cn
3 College of Computer Science and Engineering, Shandong University of Science and Technology, Qingdao 266590, China
* Correspondence: jengshyangpan@gmail.com

Abstract: The PID (proportional–integral–derivative) controller is the most widely used control method in modern engineering control because it has the characteristics of a simple algorithm structure and easy implementation. The traditional PID controller, in the face of complex control objects, has been unable to meet the expected requirements. The emergence of the intelligent algorithm makes intelligent control widely usable. The Quasi-Affine Transformation Evolutionary (QUATRE) algorithm is a new evolutionary algorithm. Compared with other intelligent algorithms, the QUATRE algorithm has a strong global search ability. To improve the accuracy of the algorithm, the adaptive mechanism of online adjusting control parameters was introduced and the linear population reduction strategy was adopted to improve the performance of the algorithm. The standard QUATRE algorithm, particle swarm optimization algorithm and improved QUATRE algorithm were tested by the test function. The experimental results verify the advantages of the improved QUATRE algorithm. The improved QUATRE algorithm was combined with PID parameters, and the simulation results were compared with the PID parameter tuning method based on the particle swarm optimization algorithm and standard QUATRE algorithm. From the experimental results, the control effect of the improved QUATRE algorithm is more effective.

1. Introduction

The PID controller's function is to determine the steady-state error value of the system from the actual given value and the actual output value of the control system. The steady-state error is linearly calculated by proportion, integral, and differential to form the control quantity. The control effect of the PID controller on the controlled object can be changed by changing the values of three kinds of numbers. Due to this, in many cases, PID controller parameter tuning is indispensable in the control system, as it is an important control method to achieve the purpose of control. The traditional PID controller parameter tuning method is manual tuning, through calculations to obtain PID controller parameters, and then fine-tuning to achieve a more ideal control effect. The traditional PID controller parameter tuning method has a good effect on the linear time-invariant system. When facing the complex controlled object with nonlinear and time delay characteristics, the artificial calculation is large and time-consuming and the parameter error is large. Even if the parameter control system is fine-tuned, it is almost impossible to achieve the desired control effect. With the continuous development of industrial control technology and the large-scale use of power electronic devices, the control objects of actual industrial control present high-order, time-delay, and nonlinear characteristics. It is difficult to achieve the optimal tuning of P1D parameters with traditional PID parameter tuning

methods. For complex control objects, many scholars apply swarm intelligence algorithms to PID parameter tuning. The development of information and computing technology and the penetration of other related disciplines promote the research of control science and engineering, and intelligent control has become an important trend. To achieve a better control effect of the PID controller in complex controlled objects, this paper proposes a method of combining the new improved algorithm with PID parameters and proves its feasibility and superiority.

Intelligent optimization algorithm is a general term for a large class of algorithms and an important part of computational intelligence [1], including the evolutionary computing method, the swarm intelligence algorithm based on biological group behavior, and a random search algorithm based on physical process simulation [2]. For a long time, finding solutions to a problem has been one of the goals of human production and life. There are many fields that use an optimization algorithm in combination with, e.g., a control system to find the best control scheme [3]; image processing; wireless sensor network [4,5]; information hiding [6,7]; and in real life, it can be applied to vehicle routing problems [8]. The optimization algorithm was applied in image and network segmentation in [5,7]. The algorithm was also used for sensor ontology meta-matching in [9]. When the problem to be solved has a non-unique feasible solution, obtaining the optimal solution of a certain performance index through optimization is essential. The process of obtaining the optimal solution is the process of finding the best solution. In summary, an intelligent optimization algorithm is an excellent and practical tool.

To combine the swarm intelligence optimization algorithm with PID control, we must first determine the similarities between them. The algorithm must find the best solution to the equation, and the uncertainty of the control is the reason why we need to determine the best control state. Parameters also have a variety of scheme combinations. We need to match the PID control parameters with the fitness function of the algorithm to find the best control scheme. In the algorithm, all individuals are in parameter combination with the PID controller. The fitness values of all individuals were calculated by using the performance index function. The advantages and disadvantages of the PID controller parameter combination were judged by comparing the fitness values, and then the individual with the optimal fitness value was selected as the global optimal parameter combination at the current iteration number. In this paper, the improved QUATRE algorithm was combined with the parameter tuning of the PID controller:

- An improved QUATRE algorithm was proposed based on the original QUATRE algorithm. We combined the linear population size reduction, continuously reduced the population size according to the linear function, and adjusted the control parameters online to achieve better algorithm performance.
- The improved QUATRE algorithm was used to optimize the PID parameters, and the performance index function integral of time-weighted absolute error (ITAE) was used as the fitness function to find the optimal value through the improved optimization algorithm. At this time, the best individual was the global optimal parameter combination to achieve the optimal control of the control system.

2. Related Work

2.1. PID Parameter Tuning Method

The PID controller is a widely used control method [10], whose control performance depends on the parameter tuning of the PID controller, i.e., the combination and collocation of controller parameters. In 1942, Ziegler and Nichlos proposed the earliest PID parameter tuning method, namely the Z-N tuning method [11], by studying the PID parameter tuning of the first-order inertial pure lag link. Later, scholars all over the world continued to optimize and improve the Z-N tuning method and constantly proposed new tuning methods, such as the Cohn–Coon tuning method [12], relay feedback tuning method [13], internal mode PID parameter tuning method [14], RZN (refined Ziegler–Nichols) tuning method [15], etc.

In the context of the continuous development of industrial control technology and the wide applications of power electronic devices, the actual industrial control object presents the characteristics of a high order, time delay, nonlinear, etc.; whilst it is difficult to achieve optimal PID parameter tuning with the traditional PID parameter tuning method. For complex control objects, many scholars apply swarm intelligence algorithms to PID parameter tuning, which is applied in various fields. In Reference [16], the rules of the PID controller were derived by a method based on fuzzy logic; in Reference [17], a fuzzy PID controller was designed. In Reference [18], neural network theory was applied to parameter tuning and a parameter self-tuning method was proposed based on neural network PID. Simulation experiments have verified that this method can reduce the workload of parameter identification and realize online closed-loop tuning. In Reference [19], the authors introduced a genetic algorithm into PID parameter tuning and verified the feasibility of the tuning method in the simulation system. In Reference [20], a fuzzy PID controller based on a neural network was proposed, which has nonlinear control and self-learning mapping. In Reference [21], the authors used an immune information processing mechanism to optimize particle swarm optimization, solving the problem of algorithm prematurity, and applying it to the parameter self-tuning of the PID controller. In Reference [22], the neural network training algorithm was applied to PID parameter tuning and a PID neural network controller was designed. In Reference [23], the authors proposed a PID parameter tuning method based on the improved biogeography algorithm. From the simulation experiment, the classical test function was used to verify and compare with other swarm intelligence algorithms. What is satisfying is that the controller error and overshoot of the proposed method are smaller. In Reference [24], a nonlinear attenuation particle swarm optimization algorithm combined with an improved dynamic learning factor was proposed, and the algorithm was applied to the parameter tuning of the PID governor of hydro generator sets. In Reference [25], the authors proposed a multi-objective particle swarm optimization algorithm that only optimizes the overshoot and adjustment time of the system and realized the self-tuning of PID parameters.

In Figure 1, the control system block diagram is a conventional PID controller used in engineering. The system is mainly composed of a PID controller, actuator, and controlled object. As a kind of linear controller, the linear control function of deviation is composed of proportional, integral, and differential by linear combination. The output of the control function is applied to the actuator by inputting the current deviation calculation, to complete the control of the controlled object. The specific change process is given in Equation (1).
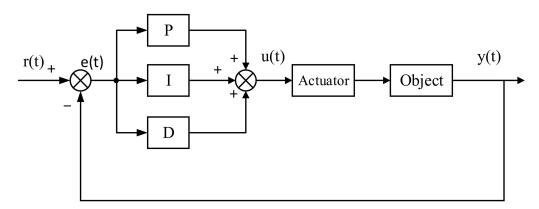

Figure 1. PID control schematic diagram.

The specific formula:

$$
\begin{aligned}
u(t) &= K_p[e(t) + \frac{1}{T_I}\int_0^t e(t)d\tau + T_D \frac{d[e(t)]}{dt}] \\
&= K_P e(t) + \frac{K_P}{T_I}\int_0^t e(t)d\tau + K_p T_D \frac{d[e(t)]}{dt} \\
&= K_P e(t) + K_I \int_0^t e(t)d\tau + K_D \frac{d[e(t)]}{dt}
\end{aligned}
\tag{1}
$$

where *r(t)* is the expected output; *y(t)* is the actual output; *e(t)* is the difference between the expected output and the actual output; *u(t)* is the output regulated by the controller; K_p is the proportional coefficient; T_I is the integrator time constant; and T_D is the differentiator time constant.

2.2. QUATRE Algorithm

QUATRE is an evolutionary algorithm put forward by Meng Zhenyu and others [26], and the QUATRE algorithm is a novel evolutionary algorithm which uses a similar geometry mapping in the principle of transition function as well as random initialization population particles, where each particle through the same mapping transformation function realizes their evolution; and then it searches and evaluates the fitness function of the problem again, writes down the optimal value, and repeats the iteration until the termination condition is satisfied at the time of the current—solution which is the approximate optimal solution. Particle evolution in the new algorithm can be written in a form similar to affine transformation, so it is named after the quantum affine transformation formation evolutionary algorithm. The QUATRE algorithm has only one parameter to be set in terms of particle change and spatial expansion. Moreover, the QUATRE algorithm has significant advantages in enhancing cooperation among particles, having strong global search ability, and solving many difficult points in optimization. The specific change process is shown in Equation (2):

$$\begin{cases} B = X_{gbest} + F \times (X_{r1} - X_{r2}) \\ X \mid \rightarrow M \otimes X + \overline{M} \otimes B \end{cases} \tag{2}$$

Equation (2) shows the core idea of the QUATRE algorithm population evolution. X_{r1} and X_{r2} are two matrices obtained by the random permutation of row vectors. The first line of the formula indicates that the particles of each population move a certain distance in the random direction represented by $X_{r1} - X_{r2}$ with the position X_{best} of the global optimal particle as the center, where *F* is the coefficient of the moving step, and the result is stored in the matrix *B*. The second row indicates that the random perturbation represented by *B* is randomly applied to the current particle coordinate matrix *X* in a form of affine transformation, where the joint search matrix *M* is obtained in the manner shown in Equation (3). In other words, the rows of the lower triangular matrix with the element value of 1 are randomly exchanged, and then the 0 and 1 elements in each row are randomly exchanged. This is obtained by taking the "logical negation" operation on all the binary elements in transformed M_tmp.

M_tmp is a lower triangular matrix, in which each row is randomly arranged which ensures that the elements of each row are unchanged, then the row vectors are randomly arranged to obtain the scrambled collaborative search matrix *M*. The search matrix *M* was used to update the position of the algorithm:

$$M_{tmp} = \begin{bmatrix} 1 & & & \\ 1 & 1 & & \\ & & \ddots & \\ 1 & 1 & \cdots & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & & \\ & \cdots & & \\ 1 & 1 & \cdots & 1 \\ & & 1 & 1 \end{bmatrix} = M \tag{3}$$

2.3. Particle Swarm Optimization

Particle swarm optimization (PSO) was first proposed by two American scientists based on the process of birds foraging to find the best foraging area [27]. As an intelligent algorithm, PSO simulates the best decision-making process. Bird foraging is similar to the human decision-making process. Think about whether you will be affected by your own experience (local optimum) and the experience of people around you before you make a choice (global optimum). In the same way, in the process of foraging, the initial position of each bird is in a random state. Of course, we do not know where the best foraging point is, and the flight direction of each bird is also random. It can be considered that during the early stage of foraging, the trajectory of birds is disordered. Over time, birds in

a random position learn from each other and share foraging information within the group. In each foraging process, each bird estimates the value of finding food by combining its own experience and information transmitted by peers. Based on this way of research, particle swarm optimization (PSO) emerges as the times require.

In particle swarm optimization, the search for the objective function is solved by using the motion of each particle. The motion direction of particles includes two parts: the random part and definite part. That is to say, the possible motion direction of each particle is not only towards the current global optimum and the optimum of the particle itself but also random motion. Compared with genetic algorithm, it does not use an evolutionary operator, but treats each individual as a particle in the optimization space and has a certain speed. Particle motion is dynamically adjusted by individual trajectory and group historical trajectory. The following is the algorithm process:

1. Determine the specific function equation to be optimized f(x);
2. Define initial value: velocity v and position x of N particles;
3. Start to search the global optimum and the fitness value of each individual was calculated;
4. The fitness value of each particle's current position is compared with the global best and its own historical best to update the particle's historical best and global best;
5. The updating formula of velocity and position is used to change the v and x;
6. Repeat steps 2–5 until the stop rule is met.

Update formula of v and x:

$$v_i^d = wv_i^d + c_1 r_1 (p_i^d - x_i^d) + c_2 r_2 (p_g^d - x_i^d) \tag{4}$$

$$x_i^d = x_i^d + \alpha v_i^d \tag{5}$$

where $i = 1, 2, \ldots, m$ denotes the ith particle (bird); $d = 1, 2, \ldots, D$ denotes the d dimension; w is a non-negative number, which is called the inertia factor; r_1 and r_2 are random numbers from 0 to 1; α is called constraint factor, which is to control the weight of speed.

3. The Proposed Algorithm

In this section, an improved QUATRE algorithm named L-EQUATRE is proposed, which is based on the variant of QUATRE and adds the adaptive strategy of parameters and a strategy of linear decreasing population. The specific process of the improved algorithm is given below.

3.1. Mutation Method of B

To achieve a better convergence effect, Matrix *B* adopts the following strategy changes:

$$\begin{cases} B_{i,G} = X_{i,G} + F \times (X_{best,G}^p - X_{i,G}) + F \times (X_{r1,G} - \hat{X}_{r2,G}) \\ B = [B_{1G}, B_{2G}, \ldots, B_{N,G},]^T, i \in \{1, 2, \ldots, N\} \end{cases} \tag{6}$$

where $X_{best,G}^p$ denotes a chosen individual from the top 100*P*% individuals randomly in the current population. Moreover, an optional external archive A was employed to store the inferior individuals during the evolution. Additionally, the individual $\hat{X}_{r2,G}$ was chosen from a combination of external archives and the current population. In addition, during the early stage of the algorithm, the point is to perceive the whole search space, let the population explore on a large scale, rather than be limited to a good solution. During the late stage of the algorithm, our key is to find the best, so the local search becomes important. In order to enable space perception during the initial stage of the algorithm, balance with

the development search ability during the later stage, a new linear decreasing parameter P was used in the improved algorithm to define the percentage of optimal individuals:

$$P = \max \left\{ \frac{P_{\min} - P_{\max}}{nfe_{\max}} \times nfe + P_{\max}, 2 \right\} \tag{7}$$

where $P_{\max} = 0.2$ and $P_{\min} = 0.05$ are the recommended values.

3.2. Adaptive Scale Factor F and M-Matrix Evolution Scheme

The scale factor F obeys Cauchy distribution C (X_F, γ_F), and during the initial stage, the position parameter $X_F = 0.5$, while $\gamma = 0.2$. nfe means the number of function evaluations. The dynamic rule is in Equation (6):

$$\begin{cases} \omega_s = \frac{std(\Delta loc_i)}{\sum_{s-1}^{|S_F|} std(\Delta loc_i)} \\ \Delta loc_i = loc(U_{i,G} - X_{i,G}) \\ mean_{WL}(S_F) = \frac{\sum_{s-1}^{|S_F|} \omega_s \times S_F^2(s)}{\sum_{s-1}^{|S_F|} \omega_s \times S_F(s)} \\ X_{F,G+1} = \begin{cases} mean_{WL}(S_F), S_F \neq \varnothing \\ X_{F,G}, otherwise \end{cases} \end{cases} \tag{8}$$

where S_F represents a collection of F values associated with success individuals and $loc(U_{i,G} - X_{i,G})$ indicates the parameter of the positioning position difference $U_{i,G} - X_{i,G}$ and std() represents the operation of calculating standard deviation.

The key point of M-matrix generation lies in the number of kth vectors incorporated into the initialization matrix in each generation of the lower triangular matrix. The number of the kth vector can be calculated by probability $P(A = K)$. The update formula of P is given in Equation (9):

$$\begin{cases} ns = \sum_{d=1}^{D} ns_d \\ P(A = k) = \begin{cases} \frac{ns_k^2}{ns \times (ns_k + nf_k)}, ns_k > 0 \\ \varepsilon \end{cases} \end{cases} \tag{9}$$

where ns is the size of the successful individuals of the recorded population, ns_k is the number of successful individuals inherited from the donor vector by k parameter and $\varepsilon = 0.01$ is to avoid zero probability [28].

In addition, in the process of evolution, our algorithm has two groups. The selection probability $P(j)$ indicates the possibility of an individual being classified into the jth group. Then, all the current individuals can be divided into different groups through random universal selection. In the groups, $P(j)$, $j = 1, 2$ dynamically change according to the success rate of the individual in the evolution process, as in Equation (10):

$$\begin{cases} r_j = \begin{cases} \frac{ns_j^2}{ns \times (ns_j + nf_j)}, ns_j > 0 \\ \varepsilon, otherwise \end{cases} \\ ns = \sum_{j=1}^{2} ns_j \\ P(j) = \frac{r_j}{\sum_{j=1}^{k} (r_j)} \end{cases} \tag{10}$$

where ns_j represents the number of individuals who found a better solution in the jth group and nf_j refers to individuals who cannot find a better solution in the jth group. The upper and lower selection probability of each group is fixed, so $P(j)$ should satisfy $P(j) \in [0.3, 0.7]$, $j = 1, 2$, and only satisfy that the least selected group needs to update the probability for each generation.

3.3. Population Size Reduction Scheme

For the overall performance of the algorithm, the population reduction scheme is effective and practical, and it is a simple scheme. Commonly, a population size reduction scheme can achieve a balance between the diversity of trial vectors and the number of generations. Accordingly, in this paper, we employed a simply linear population size reduction strategy to improve the performance of the proposed algorithm. After each generation G, the population size in the next generation N_{G+1} is computed according to Equation (11):

$$N_{G+1} = round\left[\left(\frac{N^{\min} - N^{init}}{MAX\_NFE}\right) \times NFE + N^{init}\right] \tag{11}$$

The population size of the first generation is N^{init}, the population at the end of the run is $N^{\min}$, $N^{\min}$ set as the minimum possible value, NFE is the current number of fitness evaluations. MAX_NFE is the maximum number of fitness evaluations. Whenever $N_{G+1} < N_G$, the ($N_G - N_{G+1}$) are the worst ranked individuals which will be removed from the population.

To show the improved QUATRE algorithm more intuitively, the following Algorithm 1 is the pseudo code of the improved QUATRE algorithm.

Algorithm 1 Pseudo Code of the L-EQUATRE Algorithm.

Input: Bound constraints $[R_{D,\min}, R_{D,\max}]$, the fixed maximum number of function evolution nfe_{max};
Output: Best individual X_{gbest}, Best fitness value $f(X_{gbest})$, number of function evaluation nfe;
1. Initialize the population size ps, scale factor F, inheritance probability P, all individuals $X = \{X_1, X_2, \ldots, X_{ps}\}$, $A = \varnothing$, $r^{arc} = 1.6$, $p_{max} = 0.2$, $p_{min} = 0.05$, $G = 1$;
2. **while** $nfe \leq nfe_{max}$ **do**
3.   **for** $i = 1; i \leq ps; i ++$ **do**
4.   Generate $X^p_{best,G}$, $X_{r_1,G}$ and $\widehat{X}_{r_2,G}$;
5.   **if** G > 2 **then**
6.   Adjust the population size;
7.   Adjust the size of storage A;
8.   **end if**
9.   Generate F_i for ith individual;
10.   Readjust F_i into the bound constraints if necessary;
11.   **end for**
12.   Generate evolution matrix M according to the adopt scheme;
13.   Generate B according to Equation (6);
14.   Generate trial candidates by employing evolution matrix M;
15.   Calculate fitness values of all $U_{i,G}$;
16. $nfe = nfe + ps$;
17.   **for** $i = 1; i \leq ps; i ++$ **do**
18.   **if** $f(U_{i,G}) \leq f(X_{i,G})$ **then**
19. $X_{i,G+1} = U_{i,G}$;
20.   **else**
21. $X_{i,G+1} = X_{i,G}$;
22.   **end if**
23.   **end for**
24.   if $S_F \neq \varnothing$ **then**
25.   Update X_F;
26.   Update P ($A = k$) ($k = 1, 2, \ldots, D$);
27.   **end if**
28. $G = G + 1$;
29.   Update storage A;
30. **end while**
31. $f(X_{gbest}) = f(X_{gbest,G})$, $X_{gbest} = X_{gbest,G}$;
32. Return X_{gbest}, and $f(X_{gbest})$;

---

4. PID Controller Parameter Tuning Based on the Improved QUATRE Algorithm

The performance index of PID control is a set of standards for judging the quality of the controlled object. The evaluation index has the function of guiding the field personnel to adjust the PID parameters to establish a reasonable control system [29]. The selection of control indicators has an important influence on the results of the control. Commonly used performance indicators include dynamic performance indicators and error integral performance indicators [30]. This article uses the absolute error multiplied by time (ITAE) performance index:

$$ITAE = \int_0^\infty t\,|e(t)|dt \tag{12}$$

In the improved QUATRE algorithm, all the particles can be regarded as a combination of PID controller parameters. In a given search space, the particles are constantly iteratively optimized, and the globally optimal individuals are constantly updated. The algorithm will converge to the optimal solution, that is, the parameter combination with the best control performance for the controlled object. Additionally, QUATRE algorithm is improved to optimize PID parameters, especially the first set value range of PID parameters, namely the particles under the different dimensions of search, then all particles and random distribution patterns are initialized using the Sim function assigned to the PID controller parameters, through the system performance index function to calculate all the particle's fitness, and the optimal adaptive value and position of particles are kept in the search space with the constant iterative update, which obtains the global optimal particle of the optimal PID controller parameters. The main steps are as follows:

1. Set the search space of PID parameters, initialize the position of the three parameters of the particle, particle swarm size, number of iterations, and constant term C of the difference matrix.
2. The position of each particle is transferred to the established PID control model PID MODEL, and the SIMULINK program is called to simulate and calculate the ITAE performance index of the controller as a fitness function which is then transmitted back to the QUATRE algorithm.
3. Determine according to the ITAE performance index of the current particle swarm, search all particles to determine the optimal position of all, and perform position updates for individuals that meet the performance requirements.
4. Judge whether it is required according to the termination condition. If the current situation meets the stop condition of the algorithm, exit the algorithm and take the current optimal position as the optimal proportional, integral, and differential coefficients of the PID controller. If not, perform a speed and position update operation, judge and limit particle update speed and optimization range, then execute Step 2.

5. Experiment Analysis

5.1. Analysis of Algorithm Experimental Results

In this paper, three algorithms were selected for testing: first, the improved QUATRE algorithm where $N\_min = 4$; $N\_ini = \text{round}[25\log(D)\sqrt{D}]$; the dimension is 30; and $F = 0.7$. The second and third were PSO and QUATRE, where: the population number is 100; the number of iterations is 3000; and the dimension is 30. $c_1 = c_2 = 2$, $w = 0.6$ in PSO. $F = 0.7$. in QUATRE. The abscissa is $nfe/100$, the ordinate is the fitness value, and the absolute error by time (ITAE) performance index is taken as the fitness function. The implicit test of the test function of CEC2013 was adopted, and the experimental results are shown in Figures 2–5.
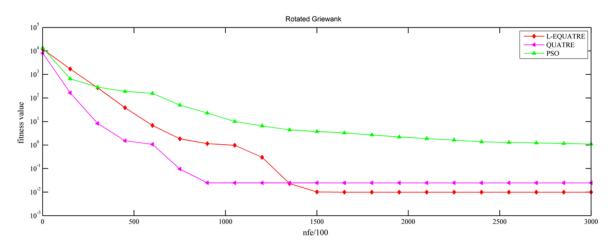
Figure 2. Rotated Schaffer's F7 function.
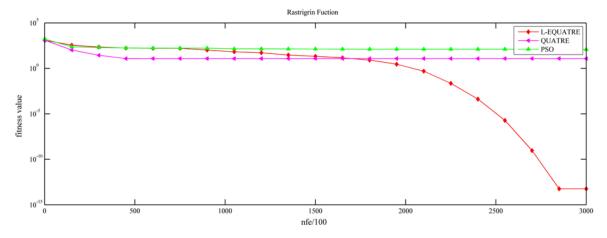

Figure 3. Rotated Griewank function.

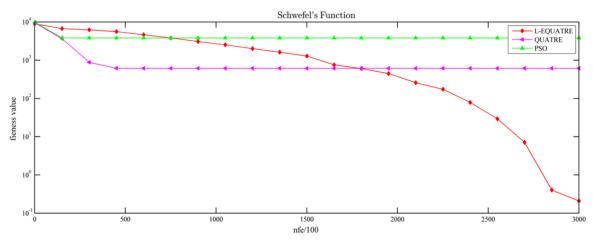**Figure 4.** Rastrigrin function.

Figure 5. Schaffer's function.

In the rotated Schaffer's function, the algorithm began to flatten before 500 generations, and the convergence speed of the three algorithms was not much different. However, the improved QUATRE algorithm determines that the optimal solution is smaller, i.e., that the value of convergence is smaller; so under the same conditions, the optimization accuracy is higher, and the performance in this function is superior. In the rotated Griewank function, the optimization results of the improved algorithm and the standard QUATRE algorithm are evidently better than that of the PSO algorithm, and the optimization accuracy is improved. In the Rastrigin function and the Schwefel function, the optimization accuracy of the improved algorithm is improved by several orders of magnitude, and a very good fitness value is obtained.

From the test function results—compared with the standard QUATRE algorithm and particle swarm optimization algorithm—in terms of optimization accuracy, it can be concluded from the curves of these test functions that the improved algorithm has been greatly improved, with convergence and smaller values. This was also due to the improvement of our strategies. When the particles start searching, they pay attention to the perception of the whole search space, and take into account the search development ability of the particles in the future. The result is that the particles can be closer to each other when searching. It is close to finding the solution we need. The performance of the algorithm has been enhanced. In the convergence rate of the algorithm, the improved algorithm is not good enough in some functions. The experimental results verify the feasibility and effectiveness of the improved QUATRE algorithm.

5.2. PID Parameter Tuning Experiment Results Analysis

MATLAB software was used to realize the simulation experiment. Two intelligent algorithms, the PSO algorithm, and the improved QUATRE algorithm are combined with PID parameters. The purpose of the intelligent algorithm is to find the combination of Ki, Kp, and Kd with the best control effect for the control system through its optimization ability. In the entire parameter tuning process, the value of the fitness is expressed by the error, and generally, the ITAE of the error integration criterion of the control system is adopted. The control objects are shown in Table 1.

Table 1. Control object model.

Mathematical Model	Properties of the Model
$\frac{e^{-12s}}{20s+1}$	First-order delay mode
$\frac{e^{-s}}{(s+1)^2}$	Second-order delay model
$\frac{e^{-s}}{(s+1)^3}$	Third-order delay model

After the original QUATRE algorithm is improved, it is combined with PID parameters, and the improved QUATRE algorithm is feasible in PID parameter tuning. The abscissa is the time (s), and the ordinate is the unit step response (amplitude). The experimental results are shown in Figures 6–8.
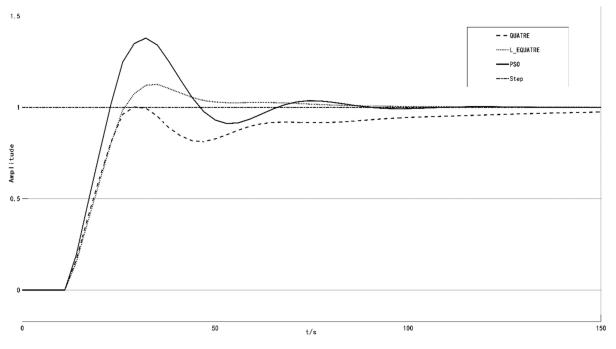

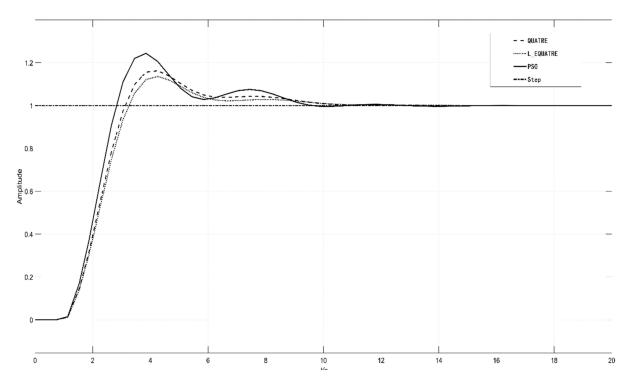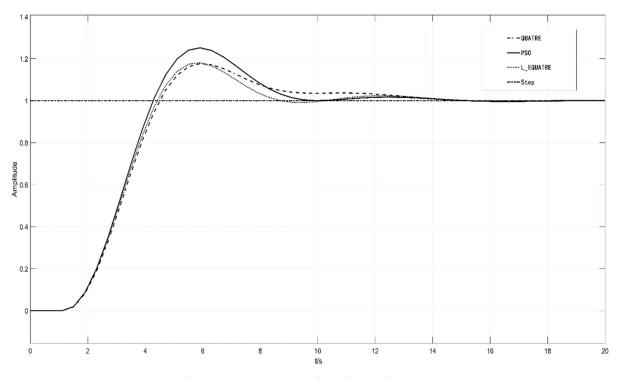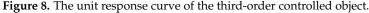**Figure 6.** The unit response curve of the first-order controlled object.

Figure 7. The unit response curve of the second-order controlled object.

Figure 8. The unit response curve of the third-order controlled object.

The algorithm parameter settings included: the number of the initial population is 100; the maximum number of iterations is 500; the value range of K_p and K_i is [0,30]; and K_d is [−20,20].

From the step response curve of the control object, it can be concluded that the PID based on the improved QUATRE algorithm has less overshoot than that of the standard QUATRE algorithm and particle swarm algorithm. The system has a small shock. On the other hand, the adjustment time is relatively small. Therefore, on the one hand, the improved PID parameter adjustment improves the stability of the control system; on the other hand, it reduces the risk of large vibration. Therefore, the PID parameters of the improved algorithm should be better for the control system. The experiment shows that the improved QUATRE algorithm is feasible in the PID parameter setting.

6. Conclusions

In this paper, there is a lot of room to improve the accuracy of the original algorithm. Therefore, an improved QUATRE algorithm was proposed. The idea is the adaptive change of matrix *B*; scale factor *F*; matrix *M*; and the population size reduction scheme. In general, programs to reduce population size can strike a balance between the diversity of test vectors and the number of generations. Therefore, this paper used the linear population reduction strategy to improve the overall ability of the algorithm. After tests of the cec213 test function, through the comparison with PSO and the standard QUATRE algorithm, the superiority of the improved algorithm was proven.

After the algorithm was improved, this paper proposed a PID parameter tuning method based on an improved QUATRE algorithm, which used the excellent search capability of the improved algorithm to determine the best parameter combination of the PID controller. Through our experimental simulation, from the unit step response diagram, it can be concluded that our new approach is better than the PID parameter tuning of POS and the original QUATRE algorithm smaller overshoot and rise time make the whole control more stable, which also shows that the PID parameter tuning method based on improved QUATRE algorithm is more effective.

References

1. Pan, J.S.; Zhuang, J.W.; Luo, H.; Chu, S.C. Multi-group Flower Pollination Algorithm Based on Novel Communication Strategies. *J. Internet Technol.* **2021**, *22*, 257–269.
2. Xue, X.; Chen, J. Optimizing ontology alignment through hybrid population-based incremental learning algorithm. *Memet. Comput.* **2018**, *11*, 154–196. [CrossRef]
3. Hlaing, W.M.; Liu, S.J.; Pan, J.S. A Novel Solution for Simultaneously Finding the Shortest and Possible Paths in Complex Networks. *J. Internet Technol.* **2019**, *20*, 1693–1707.
4. Dong, X.; Zhang, Y.; Yu, S. An Uneven Clustering Routing Protocol based on Improved K-means Algorithm for Wireless Sensor Network in Coal-mine. *J. Inf. Hiding Multimed. Signal Process.* **2019**, *10*, 53–62.
5. Nguyen, T.T.; Pan, J.S.; Lin, J.C.; Dao, T.K.; Nguyen, T.X. An Optimal Node Coverage in Wireless Sensor Network Based on Whale Optimization Algorithm. *Data Sci. Pattern Recogn.* **2018**, *2*, 11–21.
6. Wu, C.M.; Shirui, F.; Li, T.T. Research of The WSN Routing based on Artificial Bee Colony Algorithm. *J. Inf. Hiding Multimed. Signal Process.* **2017**, *8*, 120–126.
7. Pan, J.S.; Nguyen, T.-L.P.; Ngo, T.-G.; Dao, T.-K.; Nguyen, T.-T.-T.; Nguyen, T.-T. An Optimizing Cross-Entropy Thresholding for Image Segmentation based on Improved Cockroach Colony Optimization. *J. Inf. Hiding Multimed. Signal Process.* **2020**, *11*, 162–171.
8. Pan, J.S.; Wang, X.P.; Chu, S.C.; Nguyen, T.-T. A Multi-group Grasshopper Optimisation Algorithm for Application in Capacitated Vehicle Routing Problem. *Data Sci. Pattern Recogn.* **2020**, *4*, 41–56.
9. Xue, X.; Pan, J.S. A Compact Co-Evolutionary Algorithm for sensor ontology meta-matching. *Knowl. Inf. Syst.* **2017**, *56*, 335–353. [CrossRef]
10. Stuart, B. The past of PID controller. *Annu. Rev. Control* **2001**, *25*, 43–53.
11. Ziegler, J.G.; Nichols, N.B. Optimum Setting for Automatic Controllers. *Trans. ASME* **1942**, *64*, 759–768. [CrossRef]
12. Cohen, G.H.; Coon, G.A. Theoretieal Consideration of Retarded Control. *Trans. ASME* **1953**, *75*, 827–833.
13. Astrom, K.J.; Hagglund, T. Automatic Tuning of Simple Regulators with Specfications on Phase and Amplitude Margrins. *Automatiea* **1984**, *20*, 645–651. [CrossRef]
14. Rivera, D.E.; Morari, M.; Skogestad, S. Interal model control-4.PID controller design. *Ind. Eng. Chem. Process. Des. Dev.* **1986**, *25*, 252–265. [CrossRef]
15. Hang, C.C.; Astrom, K.J. How Rfinements of the Ziegler-Nichols Tuning Formula. *IEEE Proc.* **1991**, *137*, 111–118. [CrossRef]
16. Baras, J.S.; Patel, N.S. Derivation of Fuzzy Rules for Parameter Free PID Gain Tuning. *Adv. Control Chem. Process.* **1994**, *27*, 303–308.
17. Zhu, M. Research on the Design of Fuzzy PID Controller. 1994. Available online: http://cdmd.cnki.com.cn/Article/CDMD-10056-2006053125.htm (accessed on 1 April 2021).
18. Wang, K.Q.; Cao, J.; Jiao, J.; Zhang, X.D. Self Tuning of PID Controller Parameters Based on Neural Network. 1996. Available online: https://www.cnki.com.cn/Article/CJFDTotal-DBLY199603015.htm (accessed on 1 April 2021).
19. Wu, C.J.; Huang, C.H. A hybrid method for parameter tuning of PID controllers. *J. Frankl. Inst.* **1997**, *334*, 547–562. [CrossRef]
20. Li, W.; Zhao, Q. A Multivariable Fuzzy Self Tuning PID Controller Based on Neural Network. 1998. Available online: https://en.cnki.com.cn/Article_en/CJFDTotal-DLTD801.010.htm (accessed on 1 April 2021).
21. Shao, Y.Y. PID Parameters Tuning Based on a Self-Adaptive Immunity Ant Colony Algorithm. In Proceedings of the 2013 3rd Interational Conference on Electric and Electronics (EEIC 2013), Hong Kong, China, 24 December 2013.
22. Maraba, V.A.; Kuzucuoglu, A.E. PID Neural Network Based Speed Control of Asynchronous Motor Using Programmable Logic Controller. *Adv. Electr. Comput. Eng.* **2001**, *11*, 23–28. [CrossRef]
23. Zou, H.B.; Chai, T.; Bao, G. Application of Improved BBO Algorithm in PID Parameter Tuning. 2019. Available online: https://www.cnki.com.cn/Article/CJFDTotal-ZHJC201912019.htm (accessed on 1 April 2021).
24. Wang, G.R.; Zhang, J.L. PID parameter tuning of hydro generator unit based on improved particle swarm optimization algorithm. *Sinotrans* **2019**, *19*, 92–94.
25. Zhou, L.X.; Chen, Y.Z.; Ji, P.P. Application of multi-objective particle swarm optimization algorithm in PID optimization design. *Wuhan Voc. Tech. Coll. Acta Sin. Sin.* **2019**, *18*, 90–94.

26. Meng, Z.Y.; Pan, J.S.; Xu, H. QUasi-Affine TRansformation Evolutionary (QUATRE) algorithm: A cooperative swarm based algorithm for global optimization. *Knowl. Based Syst.* **2016**, *109*, 104–121. [CrossRef]
27. Kennedy, J.; Eberhart, R. Particle Swarm Optimization/Icnn95-international Conference on Neural Networks. In Proceedings of the ICNN'95—International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995.
28. Meng, Z.; Chen, Y.; Li, X.; Yang, C.; Zhong, Y. Enhancing QUasi-Affine TRansformation Evolution (QUATRE) with adaptation scheme on numerical optimization. *Knowl. Based Syst.* **2020**, *197*, 105908. [CrossRef]
29. Li, Y.; Ang, K.H.; Chong, G.C. PID control system analysis and design. *IEEE Control Syst. Mag.* **2006**, *26*, 32–41.
30. Crowe, J.; Chen, G.R.; Ferdous, R.; Greenwood, D.R.; Grimble, M.J.; Huang, H.P.; Jeng, J.C.; Johnson, M.A.; Katebi, M.R.; Kwong, S.; et al. *PID Control: New Identification and Design Methods*; Springer: Berlin, Germany, 2005; pp. 473–524.