applied sciences

MDPI

Article

Nonlinear PID Controller Parameters Optimization Using Improved Particle Swarm Optimization Algorithm for the CNC System

Xianghan Sun [1], Ning Liu [1,*], Rui Shen [2], Kexin Wang [3], Zhijie Zhao [1] and Xianjun Sheng [1]

[1] School of Electrical Engineering, Dalian University of Technology, Dalian 116024, China
[2] Industrial Control Center, China Academy of Launch Vehicle Technology, Beijing 100076, China
[3] School of Mechanical Engineering, Dalian University of Technology, Dalian 116024, China
* Correspondence: liun@dlut.edu.cn; Tel.: +86-1524-260-2608

Abstract: In this paper, a nonlinear PID (NLPID) controller is used to replace a traditional PID controller to overcome the influence of nonlinear factors in the computer numerical control (CNC) system. A particle swarm optimization based on a generalized opposition-based learning (G-PSO) algorithm is proposed to optimize the NLPID controller. The convergence speed and global optimization ability of the particle swarm optimization (PSO) algorithm are improved by introducing generalized opposition-based learning. The natural selection mutation is introduced into the G-PSO algorithm to further avoid the particles falling into local optimization. Different from the existing research, this paper designs a special fitness function according to the control objectives of improving system response speed and suppressing overshoot. By comparing the differential evolution (DE) algorithm, the ant lion optimizer (ALO) and the genetic algorithm (GA) through simulation, it is proven that the G-PSO algorithm has a faster convergence speed and better global optimization ability. Compared to Fuzzy PID and MRAC PID, G-PSO NLPID is shown to be more suitable for CNC systems. Additionally, it is proven through experiments that the rise time and settling time of the NLPID controller optimized by the G-PSO algorithm are 22.22% and 24.52% faster, respectively, than the traditional PID controller, and the system overshoot is successfully suppressed.

Keywords: opposition-based learning; particle swarm optimization; nonlinear PID controller; CNC system

Citation: Sun, X.; Liu, N.; Shen, R.; Wang, K.; Zhao, Z.; Sheng, X. Nonlinear PID Controller Parameters Optimization Using Improved Particle Swarm Optimization Algorithm for the CNC System. *Appl. Sci.* **2022**, *12*, 10269. https://doi.org/10.3390/app122010269

Academic Editors: Rodolfo Dufo-López and Giancarlo Mauri

Received: 16 August 2022
Accepted: 10 October 2022
Published: 12 October 2022

1. Introduction

Computer numerical control (CNC) systems are mainly composed of a servo control system and a feed system [1], developing high precision and high speed. Ball screws are often used as mechanical transmission in feed systems [2], and nonlinear factors such as mechanical friction and uneven screw clearance will reduce the accuracy of the CNC system [3]. Although a fully closed-loop control CNC system can be used to eliminate nonlinear factor interference, it is expensive, prone to oscillation, and requires frequent maintenance. As a result, the CNC system frequently uses semi-closed-loop control.

PID controllers are widely used in CNC systems because of their simple structure and good robustness [1,4]. Given that the parameters of the traditional PID controller are fixed, it is unable to make real-time adjustments for nonlinear factors, making it difficult to improve the control performance further. Nonlinear modification of control parameters in traditional PID controllers can reduce or even overcome the influence of nonlinear factors [5,6]. However, compared to the traditional PID controller, the nonlinear PID (NLPID) controller has more parameters to be tuned. The commonly used PID controller parameters tuning methods are the critical ratio method, response curve method, and decay method, etc. However, these methods are not globally convergent, and it is difficult to ensure that the control effect is optimal. Therefore, most researchers presently use heuristic algorithms

because the convergence of heuristic algorithms can ensure that the parameters obtained are better [7,8]. Reference [9] uses a genetic algorithm (GA) to tune the parameters of an NLPID controller in unmanned aerial vehicles system, references [10,11] use an artificial bee colony (ABC) algorithm and a differential evolution (DE) algorithm, respectively, to tune the parameters of NLPID controller in continuous stirred tank reactor. Additionally, the ant lion optimizer (ALO) [12], monkey-multiagent algorithm [13], harmony search algorithm [14], boosting algorithm [15], and particle swarm optimization (PSO) algorithm have also been used for PID controller parameter tuning. The PSO algorithm is widely used in PID controller parameter tuning because of its advantages of fast convergence speed, good robustness, and easy-to-design fitness function. References [16,17] prove that the PSO algorithm is better than the GA, ABC algorithm, and whale optimization algorithm in tuning PID controller parameters. Researchers use crossover and mutation operations, H∞ theory, and beetle antennae search, etc., to improve the PSO algorithm, and the improved algorithms are used to tune the PID controller parameters to obtain more ideal control parameters [18–22]. Reference [23] uses an adaptive PSO algorithm to tune the parameters of an NLPID controller. Compared to the PSO algorithm, this algorithm has more advantages in convergence speed and global optimization. Although different improvements have been carried out in these aforementioned studies, only the overshoot and the response speed are considered. For CNC systems that require high control accuracy and faster control speed in real-time, it is still desirable to have an improved PSO algorithm for controller optimization to accelerate the system response speed, suppress the overshoot, and obtain high steady-state accuracy.

Tizhoosh proposed opposition-based learning (OBL) in 2005 [24]. Rahnamayan proved that the OBL-based intelligent optimization algorithm converges faster [25]. Wang et al. proposed generalized opposition-based learning (GOBL) as an improvement to OBL [26]. Inspired by this characteristic of GOBL, a GOBL-based PSO (G-PSO) algorithm was proposed for controller parameter optimization. In the proposed method, GOBL and natural selection mutation are adopted to improve the convergence speed and global optimization ability. An efficient fitness function is designed to ensure that the system meets the requirements of rapid response, no overshoot, and low steady-state error. Finally, simulations and experiments are used to verify the efficiency of the NLPID controller parameters optimization method based on the G-PSO algorithm.

2. Controller Design

The traditional PID controller equation is defined as follows:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \tag{1}$$

where $u(t)$ is the output of PID controller; $e(t)$ is the system error; and K_p, K_i, and K_d denote the coefficients for the proportional, integral, and derivative terms respectively.

The PID controller is mainly suitable for systems whose basic characteristics are linear. Hence, when the traditional PID controller is applied to CNC systems with non-linear factors, the control performance will be worsened. The nonlinear transformation of control parameters in the NLPID controller ensures that the control parameters can be changed in real-time according to the error of the system. Then, the disturbance of nonlinear factors in the system can be handled well.

The system error $e(t)$ in the CNC system can represent the control effect in real-time so the control parameters in NLPID controller are converted to nonlinear functions of $e(t)$, the equation is as follows:

$$u(t) = K_p(e(t)) \cdot e(t) + K_i(e(t)) \cdot \int_0^t e(t)dt + K_d(e(t)) \cdot \frac{de(t)}{dt} \tag{2}$$

The nonlinear transformations of K_p, K_i, and K_d can be determined with Equations (3)–(5) [6], as follows:

$$K_p(e(t)) = a_p + b_p\left(1 - \operatorname{sech}\left(c_p e(t)\right)\right) \tag{3}$$

$$K_i(e(t)) = a_i \cdot \operatorname{sech}(b_i e(t)) \tag{4}$$

$$K_d(e(t)) = a_d + b_d \Big/ \left(1 + c_d \cdot \mathrm{e}^{d_d e(t)}\right) \tag{5}$$

where a_p, b_p, c_p, a_i, b_i, a_d, b_d, c_d, and d_d are the nine parameters of the NLPID controller to be optimized.

Typically, K_p can reduce overshoot and increase the response speed of the system. If $e(t)$ is large, K_p should be large, and K_p should be reduced when $e(t)$ decreases to 0. K_i is primarily used to reduce steady-state error in the system. If $e(t)$ is large, K_i should be small. If $e(t)$ is small, the system is approaching or has reached a steady state, and K_i must be increased to eliminate steady-state error. K_d can reduce settling time and improve system stability. If $e(t)$ is small, K_d should be large. Otherwise, K_d should be reduced to improve the sensitivity of the system.

3. Controller Parameters Tuning

This section introduces improvements to the PSO algorithm, including GOBL, natural selection mutation, inertia weight, and learning coefficient optimization. Moreover, the design of the fitness function in the G-PSO algorithm is introduced.

3.1. Particle Swarm Optimization Algorithm Based on GOBL

The PSO algorithm was proposed by Eberhart and Kennedy in 1995 [27]. In the PSO algorithm, the particle changes its velocity v and position x in accordance with Equation (6).

$$\begin{cases} {v_{id}}^{t+1} = \omega \cdot v_{id}^{t} + c_p \cdot r_1 \cdot (pbest_{id} - x_{id}^{t}) + c_g \cdot r_2 \cdot (gbest - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \end{cases} \tag{6}$$

where i is the particle index; d is the search dimension; t is the number of iterations; ω is the inertia weight; c_p and c_g are the cognitive and social learning coefficients, respectively; r_1 and r_2 are two random numbers belonging to [0, 1]; $pbest_{id}$ is the current historical optimal position of the i-th particle in the dth search dimension; and $gbest$ is the current population optimal position.

The particle in the PSO algorithm changes its position x primarily by modifying the speed v. Hence, the particle position change is constrained by change in speed. The particles are directly transferred to new positions based on GOBL, and their fitness compared before and after movement, which broadens the search range for particles and speeds up convergence.

Let $X = (x_1, x_2, \cdots, x_N)$ be N particles in d-dimensional space where each particle satisfies $x_i \in [a_i, b_i]$ (a, b, are d-dimensional search space boundary), then the opposite particles $\breve{x}_i$ are obtained by Equation (7).

$$\breve{x}_i = a_i + b_i - x_i \tag{7}$$

As an improvement of OBL, GOBL can further enhance the search potential of particles, thereby enhancing the global optimization ability of the algorithm. The GOBL equation is shown in Equation (8).

$$\breve{x}_i = k \cdot (a_i + b_i) - x_i \tag{8}$$

where $[a_i, b_i]$ is the boundaries of the i-th particle and k is a random number belonging to [0, 1].

a_i and b_i dynamically changes in iteration to avoid unnecessary opposite search following Equation (9) [28].

$$a_i = \min(x_i), b_i = \max(x_i) \tag{9}$$

The particle is modified according to Equation (10) when it surpasses the search space.

$$\breve{x}_i = \begin{cases} rand(a_i, b_i), \breve{x}_i < a_i \text{ or } \breve{x}_i > b_i \\ \breve{x}_i, a_i < \breve{x}_i < b_i \end{cases} \tag{10}$$

The opposite population is combined with the original population during particle initialization and each iteration, and the first N particles with the best fitness are chosen to establish a new population. This method enhances the global search ability of the particles.

3.2. Natural Selection Mutation

Natural selection mutation is carried out to expand the search space surrounding the dominant particles, ensure that the particles do not fall into local optima, and further accelerate the convergence speed.

As indicated in Figure 1, this paper classifies all particles according to the natural selection rule. The first three particles with the best fitness are the dominant group P_1, the disadvantaged group P_3 is the last three particles with the worst fitness, and the rest are the common group P_2. P_1 should have more development space, while P_3 should learn from the dominant group, and no mutation is operated on P_2.

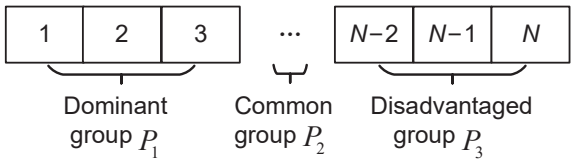

Figure 1. Natural selection of populations

The particles are prevented from falling into local optima by increasing particle disturbance through the mutation of the dominant group. Furthermore, the next generation of particles will move to the position of particles in the dominant population.

In this paper, P_1 is mutated based on the standard Cauchy distribution according to Equations (11)–(13).

$$F(x) = \frac{1}{\pi}\arctan(x) + \frac{1}{2} \tag{11}$$

$$P_1{}^* = P_1 + u(i) \cdot F(x) \tag{12}$$

$$u(i) = \frac{\sum_{i=1}^{N} v_{id}}{N} \tag{13}$$

where N is the total number of particles.

As discussed in [29], mutation should be determined by the number of iterations and the distance between *pbest* and *gbest*, according to Equations (14)–(16).

$$x(d) = e^{-30 \cdot \frac{t}{t_{\max}}} \cdot \left(1 - \frac{r(d)}{r_{\max}}\right) \tag{14}$$

$$r(d) = |gbest(d) - avg\_pbest(d)| \tag{15}$$

$$avg\_pbest(d) = \frac{\sum_{i=1}^{N} pbest_{id}}{N} \tag{16}$$

where t is the number of iterations and $t_{\max}$ is the maximum number of iterations.

3.3. Optimization of Inertia Weight and Learning Coefficients

The inertia weight ω affects the optimization ability of the G-PSO algorithm. When the particles are aggregated, a large ω is set to ensure the search ability of the particles, and when the particles disperse, ω should be small [30]. Therefore, this paper adopts an adaptive adjustment method, as follows:

$$\omega = \begin{cases} \omega_{\min} + \frac{(\omega_{\max} - \omega_{\min})(f_i - f_{\min})}{f_{avg} - f_{\min}}, f_i \leq f_{avg} \\ \omega_{\max}, f_i > f_{avg} \end{cases} \tag{17}$$

where f_i is the fitness value of the i-th particle, $\omega_{\max}$ and $\omega_{\min}$ are the maximum and minimum inertia weight, and $f_{\min}$ and f_{avg} are the minimum and average fitness of all particles, respectively.

The cognitive and social coefficients c_p and c_g are used to determine the influence of individual and population experience on particle trajectories. The optimization results may not converge or fall into a local optimum with unsuitable values of c_p and c_g [31]. The following equation shows the dynamic learning coefficients of the proposed G-PSO algorithm:

$$\begin{aligned} c_p &= c_{p2} - (c_{p2} - c_{p1})\frac{t}{t_{\max}} + k \cdot \sin(i \cdot \pi) \\ c_g &= c_{g1} + (c_{g2} - c_{g1})\frac{t}{t_{\max}} + k \cdot \sin(i \cdot \pi) \end{aligned} \tag{18}$$

where c_{p1}, c_{p2}, c_{g1}, c_{g2} are the maximum and minimum values of c_p and c_g, respectively, and k is a positive coefficient.

3.4. Design of Fitness Function

The purpose of the NLPID controller optimization is to obtain rapid system response without overshoot. Motivated by this requirement, it is critical to design a proper fitness function in the G-PSO algorithm.

Usually, the ISE, ITSE, IAE, and ITAE are employed in control systems to evaluate the control performance. However, these criteria are designed to describe the integrated control effect, which are not suitable to describe the response speed of the system and overshoot [32]. To overcome this problem, the following fitness function is adopted:

$$fitness = \int_0^\infty [|e(t)| + ov + u(t)]dt + k \cdot t_u \tag{19}$$

where $e(t)$ is the real-time error of the system, ov is the system overshoot, $u(t)$ is the output of the NLPID controller, t_u is the rise time, and k is a constant.

In Equation (19), ov limits the overshoot of the system. When the system has overshoot, ov increases the value of fitness, while ov is zero when the system has no overshoot. $u(t)$ is added to the fitness function to ensure a fast and stable control system and avoid over-control [33]. The rise time t_u is to ensure the rapidity of the system response.

3.5. Algorithm Flow of the G-PSO Algorithm

The algorithm flow of G-PSO algorithm is shown in Figure 2, and the implementation steps are as follows.

Step 1: Initialize the position and velocity of particles.

Step 2: Update *pbest* and *gbest*.

Step 3: Calculate the inertia weight and learning coefficients according to Equations (17) and (18), respectively. According to Equation (6), update the position and velocity of the particles.

Step 4: Generate the opposite population according to Equation (8). Calculate the fitness value of the opposite population and the initial population using Equation (19).

Combine the two populations, then choose the top N particles with the best fitness to form a new population.

Step 5: According to Equations (11)–(13), apply natural selection mutation to the population.

Step 6: If the conditions are met, end the optimization algorithm and output the solution. Otherwise, return to step 2.

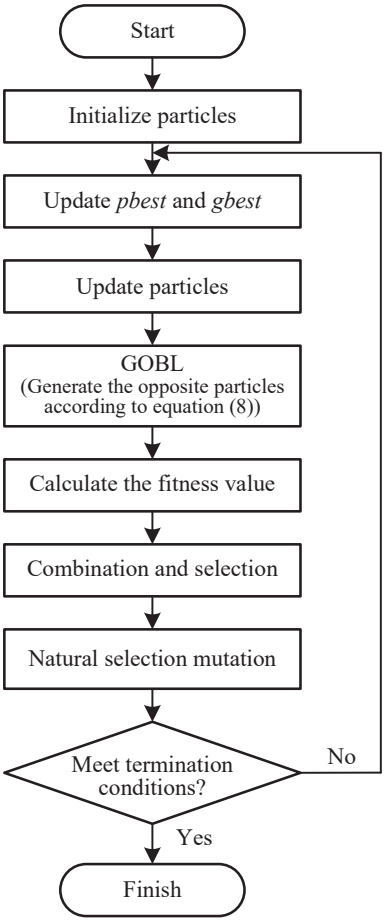

Figure 2. Flow chart of the G-PSO algorithm.

4. Simulations and Experiments

In this section, the performance of the G-PSO algorithm is tested by optimization test functions. Then, the convergence speed and global convergence ability of the G-PSO algorithm are proven through simulations. Finally, the effectiveness of the NLPID controller based on the G-PSO algorithm is verified by optimizing the PID controller and NLPID controller of the X-axis servo system.

4.1. Validation of Algorithm

We selected the Sphere (A1), Ackley (A2), and Rastrigin (A3) functions to test the performance of the G-PSO algorithm. The Sphere function can test the local search ability of the algorithm and the Ackley and Rastrigin functions can test the comprehensive search ability of the algorithm [19]. The above three optimization test functions are optimized by DE, ALO, GA, and G-PSO algorithms, respectively. In the DE algorithm, the scaling factor is 0.5 and the crossover rate is 0.3. The coefficient of ants' random walks in the ALO algorithm is 2. In the GA, the crossover rate is 0.75, and the variation rate is 0.01. In the G-PSO algorithm, c_{p1} and c_{g1} are both 0.8, c_{p2} and c_{g2} are both 1.7, ω_{min} is 0.4, and ω_{max} is 0.9. The relevant parameters are shown in Table 1.

Each algorithm runs 10 times independently for each test function. The average values of the optimal solution obtained are shown in Table 2. The average convergence curves of the algorithm are shown in Figure 3.

Table 1. The parameters for optimization test function.

Function	Dimension	Search Range	Population Size	Max Iterations
Sphere	10	[−100, 100]	50	500
Ackley	10	[−32.768, 32.768]	50	500
Rastrigin	5	[−10, 10]	50	500

Table 2. The average value of the optimal solution of the test functions by different algorithms.

Function	DE	ALO	GA	G-PSO	Optimal Value in Theory
Sphere	0.0048	6.0903×10^{-9}	0.0086	3.7506×10^{-9}	0
Ackley	0.0689	3.3138×10^{-5}	0.0518	3.1704×10^{-5}	0
Rastrigin	5.2145	4.7758	1.1424	0.6969	0

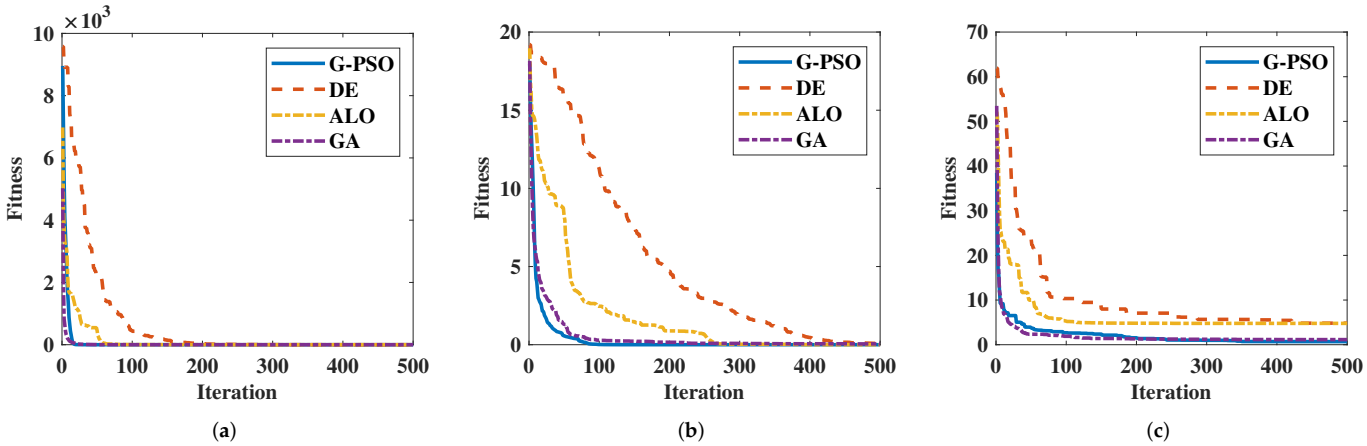

Figure 3. Average convergence curves:(**a**) Convergence curves of Sphere function, (**b**) Convergence curves of Ackley function, (**c**) Convergence curves of Rastrigin function.

As can be seen from Figure 3 and Table 2, GA converges faster in the Sphere and Rastrigin functions, while the G-PSO algorithm converges faster in the Ackley function. Moreover, the GA algorithm is easy to fall into the local optimum, so the G-PSO algorithm has more advantages in optimization.

To further verify the effectiveness of the G-PSO algorithm, the models in [7,18,21] are optimized by this algorithm and the step response curves are shown in Figure 4. Table 3 shows the system performance parameters. The rise times of the systems in [7,18,21] are 0.11 s, 1.8 s, and 1 s, respectively. After G-PSO algorithm optimization, the rise times are shortened to 0.042 s, 0.039 s, and 0.219 s and the settling times are reduced from 0.23 s, 3.7 s, and 2.1 s to 0.067 s, 0.056 s, and 0.371 s. The overshoot of the system in [7,18] is successfully suppressed. Meanwhile, with the G-PSO algorithm optimization, the number of iterations in [7,18] is reduced from 49 to 16 and 20 to 5, respectively. Although the number of iterations in this study is similar to that in [21], the response speed of the system is significantly faster.

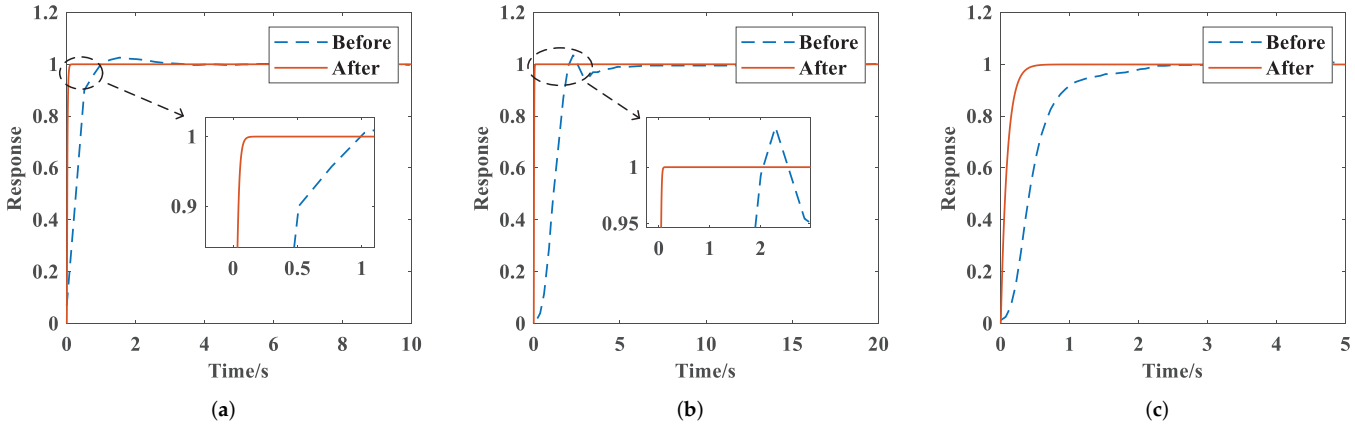

Figure 4. Response curves of different references after G-PSO algorithm optimization: (**a**) Reference [7], (**b**) Reference [18], (**c**) Reference [21].

Table 3. Comparison table of system performance parameters before and after G-PSO algorithm optimization.

	Reference [7]		Reference [18]		Reference [21]	
	Before	**After**	**Before**	**After**	**Before**	**After**
Rise time/s	0.11	0.042	1.8	0.039	1	0.219
Settling time/s	0.23	0.067	3.7	0.056	2.1	0.371
Overshoot/%	0.45	0	4.183	0	0	0
Number of iterations	49	16	20	5	3	4

The comparison results show that the G-PSO algorithm-optimized PID controller in the system has a faster response speed and can suppress the overshoot. The main reason for this is that the G-PSO algorithm has a good global optimization ability, and a fitness function is designed to improve the system response speed and suppress the system overshoot. During optimization, the system response speed and overshoot can be taken into account, so the optimization effect is obviously improved.

4.2. CNC System Simulation

The experimental object in this paper is the X-axis servo system in the X–Y motion platform in Figure 5. The server driver and X-axis are controlled by the upper machine PC and lower machine GALIL multi-axis motion control card, as shown in Figure 6, and their parameters are listed in Table 4.

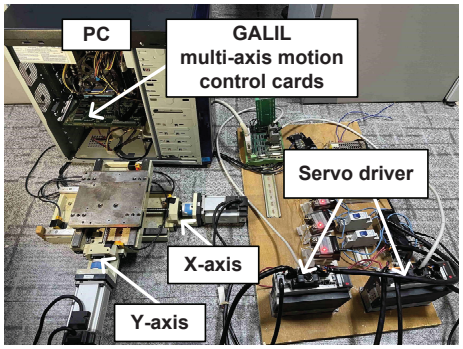

Figure 5. Experimental platform.

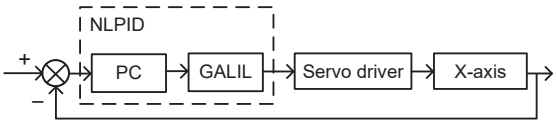

Figure 6. Block diagram of the control system.

Table 4. Parameters of experimental platform.

Indicators	Parameters
Servomotor power rating	400 W
Servo motor rated torque	1.3 N · m
Servo motor rated speed	3000 r/min
Lead of ball screw	5 mm
X-axis stroke	200 mm
Motion control card	GALIL DMC-1886

Using the ARX model identification system to obtain the number of zeros and poles, we then use the system identification toolbox in MATLAB to identify the servo system [34], and the X-axis servo system model is obtained, as follows:

$$F(s) = \frac{1.17 \times 10^3 s + 1.25 \times 10^4}{s^3 + 37.98s^2 + 99.25s + 1.88} \tag{20}$$

The model is simulated in MATLAB/Simulink. The G-PSO algorithm-optimized NLPID controller (G-PSO NLPID) is compared to the initial PID system (INIT PID), G-PSO algorithm-optimized traditional PID controller (G-PSO PID) and PSO algorithm-optimized NLPID controller (S-PSO NLPID) to evaluate its control performance. The simulations are repeated ten times to avoid random interference, and the final convergence curve is obtained by averaging the values, as shown in Figure 7. The fitness value of G-PSO NLPID reaches 1.1564 in the 13th generation, while the fitness of S-PSO NLPID reaches 1.4751 in the 45th generation. It can be seen that the convergence speed of G-PSO NLPID is significantly faster than that of S-PSO NLPID.

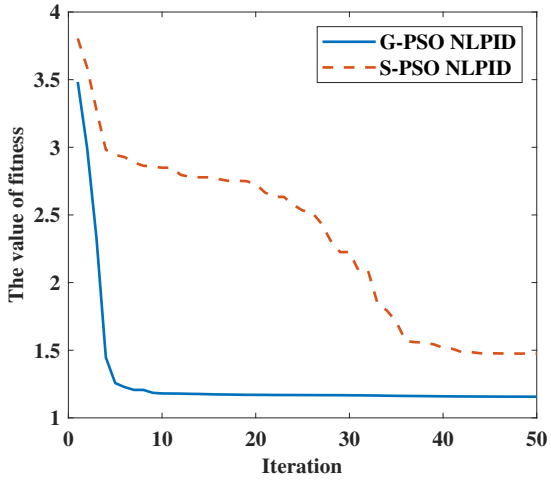

Figure 7. Convergence comparison of S-PSO NLPID and G-PSO NLPID.

Then, we apply an input signal with an amplitude of 10mm to the system, and at 1 s, a disturbance with an amplitude of 5 mm lasting 0.05 s is applied. Figure 8 shows the response curves, Table 5 shows the system performance indicators. G-PSO NLPID has a rise time and a settling time of 0.138 s and 0.182 s, respectively, which are 43.74% and 52.60% faster than those of G-PSO PID. In addition, overshoot is successfully suppressed after optimization. G-PSO NLPID responds faster to disturbance than G-PSO PID, has stronger

ability to suppress disturbances, and adjusts quickly after disturbance is removed. At the same time, Fuzzy PID and model reference adaptive control (MRAC) PID are selected for simulation comparisons. The simulation results are shown in Figure 8, and the detailed performance parameters are shown in Table 5. The Fuzzy PID has a faster response speed, but it has overshoot and is extremely sensitive to disturbances. The MRAC PID has the greatest inhibitory effect on disturbances; however, the system response speed is slower. To avoid the influence of backlash in CNC systems with high precision requirements, the system should not have overshoot, so G-PSO NLPID has advantages.

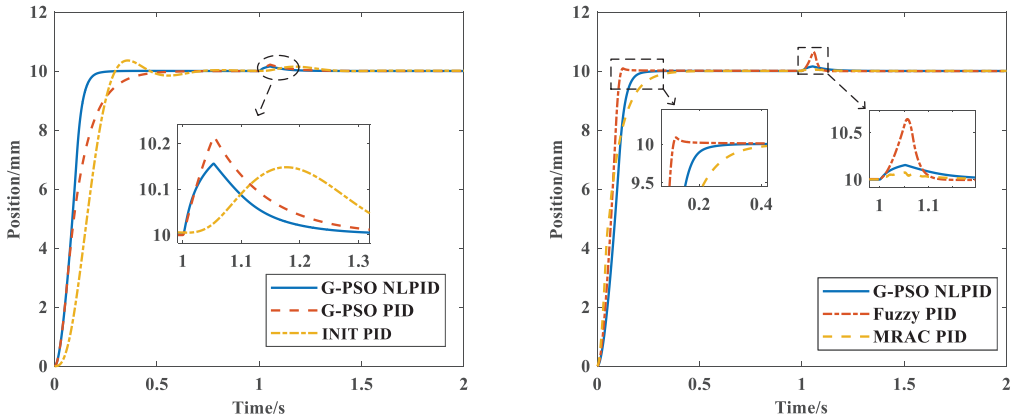

Figure 8. Simulation response curves of different controllers: (**a**) Comparison of PID and NLPID controllers based on G-PSO algorithm, (**b**) Comparison of G-PSO NLPID, Fuzzy PID, and MRAC PID.

Table 5. Detailed performance indicators of different controllers.

Indicators	INIT PID	G-PSO PID	MRAC PID	Fuzzy PID	G-PSO NLPID
Rise time/s	0.249	0.241	0.169	0.099	0.138
Settling time/s	0.414	0.384	0.278	0.111	0.182
Overshoot/%	3.518	0	0	0.903	0

4.3. Experimental Verifications

Simulations demonstrate the advantages of G-PSO algorithm in terms of convergence speed and global optimization, and that G-PSO NLPID can speed up response time and suppress overshoot. In order to verify the effectiveness of G-PSO NLPID in practical applications, the X-axis servo system shown in Figure 5 is used in this paper for relevant experiments.

The response curves of different controllers are shown in Figure 9 and the detailed performance comparison parameters are shown in Table 6. When the system is not optimized, there is overshoot, and the overshoot will introduce the backlash error in the ball screw into the system, which will reduce the precision of the CNC system. As can be seen from Table 6, the rise time and settling time of G-PSO NLPID are 0.21 s and 0.237 s, respectively, which are 22.22% and 24.52% faster than those of G-PSO PID. Furthermore, the overshoot of the system is successfully suppressed after optimization.

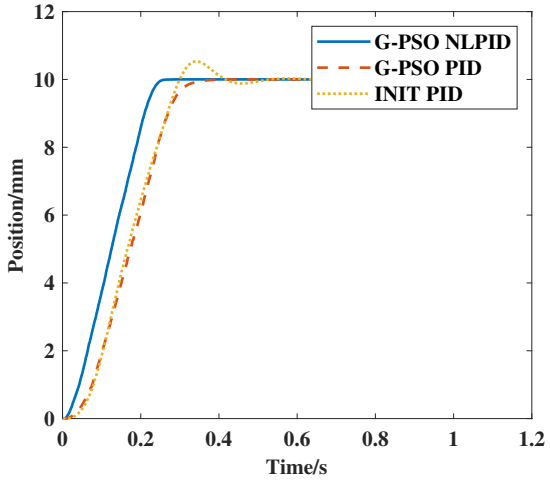

Figure 9. Experimental response curves of different controllers.

Table 6. Detailed indicators of response curves for experiments.

Indicators	INIT PID	G-PSO PID	G-PSO NLPID
Rise time/s	0.269	0.27	0.21
Settling time/s	0.389	0.314	0.237
Overshoot/%	5.3	0	0

As shown in Figures 8 and 9, the trends in the system response curves in simulations and experiments are the same, but the system response in the experiment is not as fast as in a simulation, mainly due to the input signal of the actual system (which is not the ideal step signal), the delay between the motion control card and servo drive, and the limitation of servo motor acceleration and deceleration. Although the simulation results and the experimental results are not the same, the dynamic performance is consistent, and the simulation is still crucial.

The simulation and experimental results show that the NLPID controller optimized by G-PSO algorithm has better control performance.

5. Conclusions

In this paper, an NLPID controller is used instead of a PID controller to improve the control performance of a CNC system, and an improved PSO algorithm based on GOBL is proposed for tuning the NLPID controller parameters, resulting in a satisfying control effect. With the simulation and experimental results, the following conclusions can be draw:

(1) The PSO algorithm has been optimized by GOBL, effectively expanding the particle search range and enhancing the search potential.

(2) By introducing the natural selection mutation, adaptive inertia weight, and dynamic learning coefficients into G-PSO algorithm, the global search capacity and convergence speed can be improved greatly.

(3) The dynamic performance of the system can be evaluated by introducing real-time error, overshoot, rise time and the output of NLPID controller into the fitness function.

(4) Simulation and experimental results prove the advantages of the proposed G-PSO algorithm based NLPID controller in accelerating system response and suppressing overshoot.

The main goal of this paper is to propose a better PSO algorithm for tuning the NLPID controller, but the complexity of the algorithm will be increased when the PSO algorithm is improved. In future work, we should focus on giving consideration to the simplicity and effectiveness of the algorithm.

Author Contributions: Conceptualization, X.S. (Xianghan Sun) and X.S. (Xiangjun Sheng); methodology, X.S. (Xianghan Sun); validation, K.W. and Z.Z.; formal analysis, X.S. (Xianghan Sun); investigation, X.S. (Xianghan Sun) and N.L.; resources, X.S. (Xianjun Sheng); data curation, X.S. (Xianghan Sun) and R.S.; writing—original draft preparation, X.S. (Xianghan Sun); writing—review and editing, N.L. and X.S. (Xianjun Sheng); visualization, Z.Z.; project administration, X.S. (Xianjun Sheng). All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ABC	Artificial bee colony
ALO	Ant lion optimizer
CNC	Computer numerical control
DE	Differential evolution
GA	Genetic algorithm
GOBL	Generalized opposition-based learning
G-PSO	Particle swarm optimization based on generalized opposition-based learning
IAE	Integral absolute error
ISE	Integral square error
ITAE	Integral time absolute error
ITSE	Integral time square error
MRAC	Model reference adaptive control
NLPID	Nonlinear PID
OBL	Opposition-based learning
PID	Proportion integral differential
PSO	Particle swarm optimization

Appendix A. Optimization Test Functions

The Sphere Function:

$$f(x) = \sum_{i=1}^{d} x_i^2 \tag{A1}$$

where d is the search dimension, $i = 1, ..., d$. $x_i \in [-100, 100]$.

The Ackley Function:

$$f(x) = -a \cdot e^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\sqrt{\frac{1}{d}\sum_{i=1}^{d} \cos(cx_i)}} + a + e \tag{A2}$$

where d is the search dimension, $i = 1, ..., d$. $x_i \in [-32.768, 32.768]$, $a = 20$, $b = 0.2$, $c = 2\pi$ and e is natural constant.

The Rastrigin Function:

$$f(x) = 10d + \sum_{i=1}^{d} \left[x_i^2 - 10\cos(2\pi x_i) \right] \tag{A3}$$

where d is the search dimension, $i = 1, ..., d$. $x_i \in [-10, 10]$.

References

1. Gai, H.; Li, X.; Jiao, F.; Cheng, X.; Yang, X.; Zheng, G. Application of a New Model Reference Adaptive Control Based on PID Control in CNC Machine Tools. *Machines* **2021**, *9*, 274. [CrossRef]
2. Liu, J.; Ma, C.; Wang, S. Precision loss modeling method of ball screw pair. *Mech. Syst. Signal Process.* **2020**, *135*, 106397. [CrossRef]
3. Zhang, F.R. The Influence of Nonlinear Factors of Mechanical Transmission on the NC Machining Accuracy. In *Proceedings of the Advanced Materials Research*; Trans Tech Publications, Ltd.: Zurich, Switzerland, 2012; Volume 586, pp. 254–258. [CrossRef]
4. Ye, H.; Zhang, X. Design and Simulation of Closed Loop Control System for Large Precision Machining. In Proceedings of the 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 20–22 December 2019; Volume 1, pp. 1639–1643. [CrossRef]
5. Lei, Z.; Zhou, Y. A kind of nonlinear PID controller for Refrigeration Systems based on Vapour Compression. *IFAC-PapersOnLine* **2018**, *51*, 716–721. [CrossRef]
6. Junoh, S.; Salim, S.; Abdullah, L.; Anang, N.; Chiew, T.; Retas, Z. Nonlinear PID triple hyperbolic controller design for XY table ball-screw drive system. *Int. J. Mech. Mechatronics Eng.* **2017**, *17*, 1–10.
7. Mathias, A.; Anila, M.; Sivanandan, K.S.; Jayaraj, S. Comparison of Z-N and PSO based tuning methods in the control strategy of prosthetic limbs application. *J. Theor. Appl. Mech.* **2020**, *58*, 841–851. [CrossRef]
8. Lian, Z.; Zhu, F.; Guan, Z.; Shao, X. The analysis of particle swarm optimization algorithm's convergence. In Proceedings of the 2008 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 623–626. [CrossRef]
9. Najm, A.A.; Ibraheem, I.K. Nonlinear PID controller design for a 6-DOF UAV quadrotor system. *Eng. Sci. Technol. Int. J.* **2019**, *22*, 1087–1097. [CrossRef]
10. Pugazhenthi, P.N.; Selvaperumal, S.; Vijayakumar, K. Nonlinear PID controller parameter optimization using modified hybrid artificial bee colony algorithm for continuous stirred tank reactor. *Bull. Pol. Acad. Sci. Tech. Sci.* **2021**, *69*, e137348. [CrossRef]
11. ALYOUSSEF, F.; Ibrahim, K. TRMS experimental results of new nonlinear PID tuned by DE algorithm. In Proceedings of the 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID), Elazig, Turkey, 25–27 September 2019; Volume 1, pp. 1–6. [CrossRef]
12. Marhoon, H.M.; Ibrahim, A.R.; Basil, N. Enhancement of Electro Hydraulic Position Servo Control System Utilising Ant Lion Optimiser. *Int. J. Nonlinear Anal. Appl.* **2021**, *12*, 2453–2461. [CrossRef]
13. Zhang, H.; Assawinchaichote, W.; Shi, Y. New PID parameter autotuning for nonlinear systems based on a modified monkey–multiagent DRL algorithm. *IEEE Access* **2021**, *9*, 78799–78811. [CrossRef]
14. Shamseldin, M.A.; Sallam, M.; Bassiuny, A.; Ghany, A.A. Real-time implementation of an enhanced nonlinear PID controller based on harmony search for one-stage servomechanism system. *J. Mech. Eng. Sci.* **2018**, *12*, 4161–4179. [CrossRef]
15. Kawanari, S.; Ohnishi, Y. A Design of Nonlinear PID Control Systems Using Boosting Algorithm. *IEEJ Trans. Electron. Inf. Syst.* **2008**, *128*, 1767–1772. [CrossRef]
16. Jamil, I.A.A.; Moghavvemi, M. Optimization of PID Controller Tuning Method Using Evolutionary Algorithms. In Proceedings of the 2021 Innovations in Power and Advanced Computing Technologies (i-PACT), Kuala Lumpur, Malaysia, 27–29 November 2021; pp. 1–7. [CrossRef]
17. Puchta, E.D.P.; Bassetto, P.; Biuk, L.H.; Itaborahy Filho, M.A.; Converti, A.; Kaster, M.D.S.; Siqueira, H.V. Swarm-inspired algorithms to optimize a nonlinear gaussian adaptive PID controller. *Energies* **2021**, *14*, 3385. [CrossRef]
18. Yu, Y.; Xu, Y.; Wang, F.; Li, W.; Mai, X.; Wu, H. Adsorption control of a pipeline robot based on improved PSO algorithm. *Complex Intell. Syst.* **2021**, *7*, 1797–1803. [CrossRef]
19. Sayed, M.; Gharghory, S.M.; Kamal, H.A. Gain tuning PI controllers for boiler turbine unit using a new hybrid jump PSO. *J. Electr. Syst. Inf. Technol.* **2015**, *2*, 99–110. [CrossRef]
20. Jia, L.; Zhao, X. An improved particle swarm optimization (PSO) optimized integral separation PID and its application on central position control system. *IEEE Sens. J.* **2019**, *19*, 7064–7071. [CrossRef]
21. Bian, L.; Che, X.; Chengyang, L.; Jiageng, D.; Hui, H. Parameter optimization of unmanned surface vessel propulsion motor based on BAS-PSO. *Int. J. Adv. Robot. Syst.* **2022**, *19*, 17298814211040688. [CrossRef]
22. Ma, Y.; Gu, L.C.; Xu, Y.G.; Shi, L.C.; Wang, H.T. Research on control strategy of asymmetric electro-hydraulic servo system based on improved PSO algorithm. *Adv. Mech. Eng.* **2022**, *14*, 16878132221096226. [CrossRef]
23. Valluru, S.K.; Singh, M. Performance investigations of APSO tuned linear and nonlinear PID controllers for a nonlinear dynamical system. *J. Electr. Syst. Inf. Technol.* **2018**, *5*, 442–452. [CrossRef]
24. Tizhoosh, H.R. Opposition-based learning: A new scheme for machine intelligence. In Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06), Vienna, Austria, 28–30 November 2005; Volume 1, pp. 695–701. [CrossRef]
25. Rahnamayan, S.; Tizhoosh, H.R.; Salama, M.M. Opposition-based differential evolution. *IEEE Trans. Evol. Comput.* **2008**, *12*, 64–79. [CrossRef]
26. Wang, H.; Wu, Z.; Liu, Y.; Wang, J.; Jiang, D.; Chen, L. Space transformation search: A new evolutionary technique. In Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation, Shanghai, China, 12–14 June 2009; pp. 537–544. [CrossRef]

27. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the International Conference on Neural Networks, Perth, WA, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948. [CrossRef]
28. Wang, H.; Wu, Z.; Rahnamayan, S.; Liu, Y.; Ventresca, M. Enhancing particle swarm optimization using generalized opposition-based learning. *Inf. Sci.* **2011**, *181*, 4699–4714. [CrossRef]
29. Wang, H.; Li, H.; Liu, Y.; Li, C.; Zeng, S. Opposition-based particle swarm algorithm with Cauchy mutation. In Proceedings of the 2007 IEEE Congress on Evolutionary Computation, Singapore, 25–28 September 2007; pp. 4750–4756. [CrossRef]
30. Zhou, X.; Wang, P.; Long, Z. Parameters Optimization for suspension system of maglev train via improved PSO. In Proceedings of the 2020 Chinese Automation Congress (CAC), Shanghai, China, 6–8 November 2020; pp. 2197–2202. [CrossRef]
31. Zhai, H.; Wang, W.; Li, Q.; Zhang, W. Weapon-Target Assignment Based on Improved PSO Algorithm. In Proceedings of the 2021 33rd Chinese Control and Decision Conference (CCDC), Kunming, China, 22–24 May 2021; pp. 6320–6325. [CrossRef]
32. Ramesh, H.; Xavier, S. Optimal Tuning of Servo Motor Based Linear Motion System Using Optimization Algorithm. *J. Electr. Eng. Technol.* **2022**, 1–16. [CrossRef]
33. Feng, H.; Yin, C.B.; Weng, W.W.; Ma, W.; Zhou, J.J.; Jia, W.H.; Zhang, Z.L. Robotic excavator trajectory control using an improved GA based PID controller. *Mech. Syst. Signal Process.* **2018**, *105*, 153–168. [CrossRef]
34. Sheng, X.J.; Wang, L.F. A comparison strategy for improving the precision of contour error estimation. *Int. J. Precis. Eng. And Manufacturing* **2019**, *20*, 1395–1403. [CrossRef]