machines

MDPI

Article

Trajectory Control Strategy and System Modeling of Load-Sensitive Hydraulic Excavator

Haoju Song, Guiqin Li *, Zhen Li and Xin Xiong

Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, School of Mechatronic and Automation Engineering, Shanghai University, Shanghai 200444, China
* Correspondence: leeching@shu.edu.cn

Abstract: Accurate control of excavator trajectory is the foundation for the intelligent and unmanned development of excavators. The excavator operation process requires multiple actuators to cooperate to complete the response action. However, the existing control methods to realize a single actuator of the excavator can no longer meet the practical demand. Based on this, a hybrid adaptive quantum particle swarm optimization algorithm (HAQPSO) is proposed to tune the proportional integral derivative (PID) controller parameters for enhancing the trajectory control accuracy of excavator actuators. To increase particle randomization and search speed and avoid the local convergence of QPSO, the QPSO is combined with circle chaotic mapping, Gaussian mutation operators, and adaptive adjustment factors, while the linear transformation of the contraction-expansion coefficient (*CE*) is improved to the dynamic adjustment mode. Through the interface block, a co-simulation platform for the load-sensitive system excavator is constructed, and trajectory experiments of multiple actuator compound actions are carried out. The simulation results show that—compared with ZN-PID, PSO-PID, and QPSO-PID—the trajectory error accuracy of the boom is improved by 26.59%, 32.95%, and 9.44%, respectively, which proves the high control accuracy of HAQPSO-PID in controlling the trajectory of multiple actuators.

Keywords: hydraulic excavator; load-sensitive system; hybrid adaptive quantum particle swarm optimization algorithm; trajectory control; co-simulation

check for updates

Citation: Song, H.; Li, G.; Li, Z.; Xiong, X. Trajectory Control Strategy and System Modeling of Load-Sensitive Hydraulic Excavator. *Machines* **2023**, *11*, 10. https://doi.org/10.3390/machines11010010

Academic Editor: Ning Sun

Received: 15 November 2022
Revised: 16 December 2022
Accepted: 18 December 2022
Published: 22 December 2022

Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Hydraulic excavators, as typical multifunctional construction machinery, are indispensable in energy exploitation, road building, infrastructure construction, and many other fields [1]. However, hydraulic excavator systems are complex electromechanical–hydraulic integrated systems with nonlinear and multi-constrained characteristics. Meanwhile, dead zones, valve hysteresis, fluid leakages and mechanical wear of hydraulic valves lead to certain delays and errors in the trajectory control of hydraulic actuators [2]. The actuators on a classic excavator are controlled by a professional operator using joysticks, and the digging operation is completed based on his observations. With this operation method, automatic and precise trajectory control is no longer possible.

The foundation of unmanned excavators is accurate trajectory control. The excavator, unlike indoor mechanical robotics, is operated in a harsh environment with complex circumstances and severe interference [3]. Hence, how to realize the accurate control of the operation trajectory through a simple control strategy has become a hot topic in excavator automation research. Many approaches have been proposed to handle this issue, including proportional integral derivative (PID) [4], fuzzy control [5], robust control [6], sliding mode control [7], and so on. In addition, adaptive control methods are also available and widely used in electrohydraulic servo systems [8,9]. Although many methods can achieve satisfactory control results, some methods are difficult to implement in engineering practice because of their complex structure and dependence on model parameters. Furthermore,

multiple actuators controlled by a single hydraulic pump cause nonlinear and time variability in excavator hydraulic systems, which makes it challenging to accurately establish the excavator control model.

In the field of industrial control, new control approaches have emerged. Nevertheless, PID continues to be used by more than 90% of controllers due to its simple structure, easy implementation, and high stability [10,11]. Due to the limitations of cost and working conditions, trajectory control of the excavator is mainly realized by the PID controller. To improve the tracking accuracy of a micro excavator, Zhang proposed a flow-rate-matching feedforward plus PID control approach for a zonal hydraulic system of off-road machinery. The control method has good position track accuracy, and its root mean square error (RMSE) was reduced by 20–87% in simulation experiments [12]. Tri investigated a fused fractional-order fuzzy logic and PID controller for electrohydraulic rotary actuator trajectory control, and the performance of the controller was analyzed in co-simulation experiments [13]. PID control parameters affect the stability, accuracy, and speed of the system. Additionally, the parameter adjustment process of standard PID controllers is complex and time-consuming. In recent years, particle swarm optimization (PSO) [14,15], fuzzy optimization [16], genetic optimization (GA) [17] and other optimization algorithms have been applied to tune PID parameters. Optimization algorithms are used to achieve the optimal search of PID parameters, reduce the difficulty of tuning PID parameters, and enable the PID controller to meet the control requirements.

PSO is also a swarm intelligence algorithm, which uses the power of a swarm to find the optimal solution of the problem as much as possible. The search performance is comparable to other optimization algorithms, such as ant colony optimization (ACO). PSO has the advantages of few parameters, simple software programming, and fast convergence. Some new population intelligence algorithms, such as the sparrow search algorithm (SSA) [18], the Aquila optimizer (AO) [19] and others have changed in the optimal individual updating mode, but their essence remains the population optimization method. These evolutionary algorithms have been widely used in engineering fields. However, Van den Bergh proved that PSO is not a global convergence algorithm [20]. To achieve global optimization, quantum particle swarm optimization (QPSO) was proposed to improve the global search capability of PSO. It preserves the benefits of the PSO algorithm, eliminates particle velocity and location limitations, reduces parameter settings, and improves algorithm randomness. On several benchmark test functions, the results reveal that the global search performance of QPSO outperforms that of ordinary PSO. Nevertheless, local convergence and loss of variety are still drawbacks of QPSO. Therefore, since the QPSO algorithm was put forward, numerous scholars have optimized, refined, and applied it widely.

A load-sensitive (LS) system is a closed-center hydraulic system, which is different from open-center hydraulic systems in terms of negative flow control and positive flow control. On the basis of the original valve control system, the flow-adjustable pump is added to form a pump control pressure system, which successfully overcomes the problem of median throttling loss in the open-center hydraulic system. The constant pressure difference before and after the multiway valve is achieved by a pressure compensation valve [21]. The flow to the actuator element is independent of its loads, and is only related to the spool opening of the multiway valve spool, which ensures the good handling performance of the excavator during compound action. Although the energy economy and response speed of the LS system are taken into account to some extent, the linkage of pump control and valve control in the LS system makes accurate control trajectory more challenging.

Although considerable research has been done on the trajectory control of excavator actuators, most of it has focused only on the open-center hydraulic system under the valve control system, and there have been few studies on the trajectory control of LS excavators with the combination of pump control and valve control. Furthermore, most of the research is devoted to single-actuator trajectory control such as bucket, rather than the trajectory control of all actuators in compound operation.

To enable the automatic control of multi-actuator compound operation trajectory automatic control in the hydraulic excavator. A parameter-tuning PID controller based on the hybrid adaptive QPSO algorithm (HAQPSO-PID) is proposed to improve the control accuracy of the excavator actuator trajectory. Chaotic mapping and Gaussian mutation operators are added to the classic QPSO algorithm to improve particle randomness. The linear transformation of the contraction–expansion coefficient (*CE*) is transformed into a dynamic adjustment mode to avoid local convergence, and the randomly generated factor is tuned for adaptive adjustment. HAQPSO efficiently improves the trajectory of multiple actuators under the compound operation by integrating QPSO with Gaussian variation, chaotic mapping, and adaptive adjustment. The LS excavator co-simulation model is established and the accuracy of the co-simulation model is verified by real data of an excavator. Furthermore, the trajectory control experiment of an LS excavator with multiple actuators is carried out through the co-simulation model, and the trajectory control accuracy of excavators under different PID controllers is compared.

The rest of the paper is organized as follows. The LS excavator working principle is analyzed in Section 2. Section 3 establishes the pump control and valve control model of the excavator. Section 4 provides the PID parameter-tuning method based on HQPSO. Section 5 constructs the co-simulation model of the LS excavator. Section 6 presents the comparative experimental results of trajectory comparison under different control methods. Finally, the paper is concluded in Section 7.

2. Hydraulic Excavator System Analysis

The hydraulic system of the excavator analyzed in this paper is the LS system, and the hydraulic schematic is shown in Figure 1. Displacement sensors, pressure sensors, flow meters, controllers, and other equipment are installed to collect the operating characteristics of excavator in real time. The valve control subsystem consists of actuators (boom, arm, bucket), displacement sensors, controller, and multiway valve. The diesel engine, variable pump, pressure sensors, and controller establish the pump control subsystem. Among them, the pump control subsystem controls the pump outlet pressure at a value higher than the load pressure to achieve load-sensitive control functions and reduce throttling and overflow losses [22]. The excavator trajectory is controlled by the valve control subsystem. The whole operation process of the excavator is realized by the cooperation of the valve control subsystem and the pump control subsystem. The control flowchart of the LS hydraulic excavator system is shown in Figure 2.

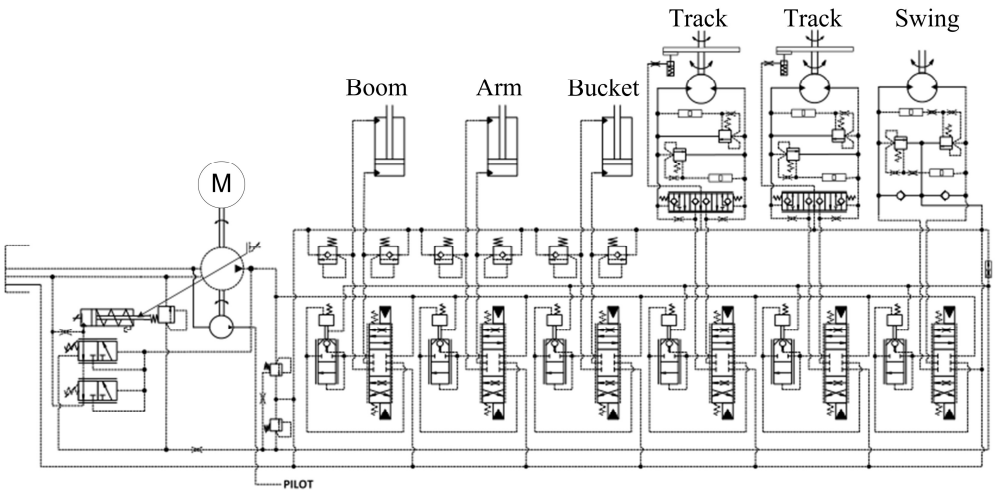


Figure 1. Hydraulic system diagram of the LS hydraulic excavator.

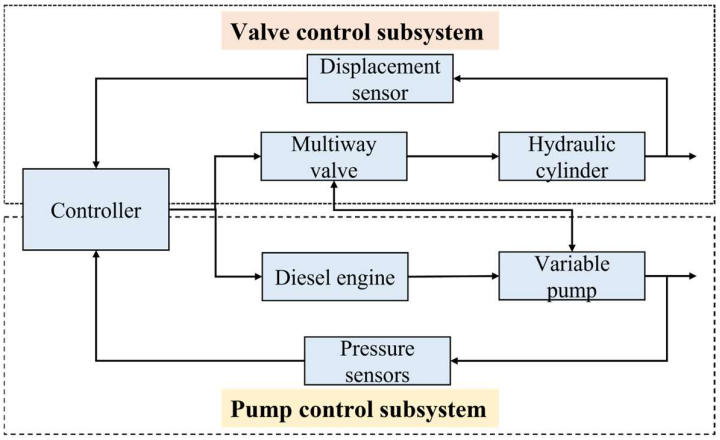


Figure 2. Control flowchart of the LS hydraulic excavator system.

During the actuator operation, the pressure signal loaded by the actuator is fed back to the spring chamber of the multiway valve pressure compensator in real time, and the pressure compensation is realized by the pressure compensator valve.

The differential pressure at both ends of the throttle port remains constant, and the throttle opening of the multiway valve determines the actuator flow rate, regardless of its load. When multiple actuators are operated, the maximum load pressure of actuators is detected by the shuttle valve and fed back to the control port of the variable pump. Concurrently, the rotation angle of the pump is adjusted to supply only the flow and pressure required by the actuators, and the throttling and overflow loss are effectively reduced. Even if the load pressure of multiple actuators is different, the multiple actuators can still be controlled quickly and precisely simultaneously, and the actuators can be guaranteed not to interfere with each other.

3. LS Excavator Systematic Modeling

3.1. Valve Control Subsystem Modeling

In the valve control subsystem, the trajectory control of the actuator entails the following processes: (1) the desired trajectory signal controls the pilot valve through joysticks, (2) the hydraulic signal of the pilot valve controls the multiway valve, (3) the multiway valve controls the hydraulic cylinder to perform the corresponding actions, and (4) the trajectory signal is fed back to the controller.

3.1.1. Joystick to Pilot Valve

During excavator operation, the driver operates the joystick to deliver directions according to the desired trajectory, and the joystick pressure-reducing valve adjusts the spool displacement to vary the output pressure depending on the joystick angle. Based on a sample excavator manual, the control equation of the joystick is:

$$P_c(\theta) = \begin{cases} 0 & 0 \le \theta \le 2.5 \\ \frac{7}{6}(\theta - 2.5) + 5 & 2.5 < \theta < 17.5 \\ 40 & 17.5 \le \theta \le 19 \end{cases} \quad (1)$$

where θ is the joystick angle and P_c is the output pressure.

The flow equation and force balance equation of the pilot valve are:

$$\begin{cases} x_{pl} K_{pl} = P_c \\ Q_{pl} = K_{pq} x_{pl} - K_{pv} P_{pv} \end{cases} \quad (2)$$

where x_{pl} is the pilot valve spool displacement, K_{pl}, K_{pq} and K_{pv} are the pilot valve spring stiffness, flow coefficient, and flow pressure coefficient, respectively, and Q_{pl} and P_{pv} are the load pressure and load flow of pilot valve, respectively.

3.1.2. Pilot Valve to Multiway Valve

The structure of the multiway valve is shown in Figure 3. The hydraulic signal generated by the pilot valve acts on both sides of the multiway valve, and the opening area of the multiway valve is adjusted through the pilot pressure and the multiway valve reset spring to achieve throttling speed control.

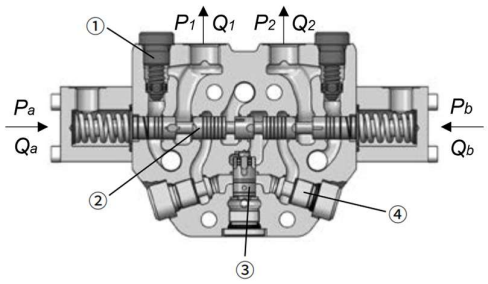


Figure 3. Schematic diagram of the multiway valve. 1. Safety valve, 2. spool, 3. pressure compensation valve, 4. load check valve.

The force balance equation and flow continuity equation of multiway valve can be given as:

$$\begin{cases} P_{pv}A_p - k_v(x_0 + x_v) = m\frac{d^2x_v}{dt^2} + B_v\frac{dx_v}{dt} + k_f x_v \\ Q_{pv} = A_p\frac{dx_v}{dt} + \frac{1}{2}C_p P_{pv} + \frac{V_p}{4\beta}\frac{dP_{pv}}{dt} \end{cases} \tag{3}$$

where x_v is the displacement of the multiway spool, m is the mass of the multiway valve spool, k_v is the spring stiffness, x_0 is spring precompression, k_f is the fluid stiffness, A_p is the control chamber area, B_v is the viscous damping factor, C_p is the leakage coefficient, and the main valve control pressure P_{pv} is:

$$P_{pv} = P_a - P_b \tag{4}$$

The spool frequently reciprocates movements during operation, which results in internal leakage. Hence, the external leakage of the multiway valve is ignored and only the internal leakage is considered, the flow continuity equation of the two control chambers in the multiway valve is:

$$\begin{cases} P_a = A_p\frac{dx_v}{dt} + C(P_a - P_t) + \frac{V_a}{\beta}\frac{dP_a}{dt} \\ P_b = A_p\frac{dx_v}{dt} + C(P_t - P_b) + \frac{V_b}{\beta}\frac{dP_b}{dt} \end{cases} \tag{5}$$

where β is the elastic bulk modulus, V_a and V_b are the volumes at both sides of the multiway valve, respectively, and P_t is the tank pressure.

3.1.3. Multiway Valve to Actuators

The bucket, arm and boom hydraulic cylinders of the excavator are all double-acting hydraulic piston actuators. The spool of the corresponding multiway valve moves as the excavator is operated, adjusting the overflow area. Its force balance equation is:

$$m_c\ddot{x}_c = A_c(P_1 - P_2) - B\dot{x}_c - K_c x_c - F_l - F_f \tag{6}$$

where m_c is the load mass, x_c is the piston displacement, P_1 and P_2 are hydraulic chamber pressures, A_c is the hydraulic cylinder effective action area, B and K_c are the viscous friction damping coefficient and spring stiffness, respectively, and F_l and F_f are the load and friction.

At this time, the flow through the multiway valve can be given as:

$$\begin{cases} Q_1 = C_d A_{d1} \sqrt{\frac{2(P_v - P_1)}{\rho}} \\ Q_2 = C_d A_{d2} \sqrt{\frac{2(P_2 - P_o)}{\rho}} \end{cases} \tag{7}$$

where Q_1 and Q_2 are the hydraulic cylinder flow rates, C_d is the multiway valve coefficient, A_{d1} and A_{d2} are the multiway valve area gradients, ρ is the oil density, and P_v and P_o are the multiway valve supply and return pressure, respectively.

The hydraulic cylinder flow continuity equation is:

$$\begin{cases} \frac{V_1}{\beta} \cdot \frac{d_{P_1}}{dt} = Q_1 - Q_{le} - \dot{x}_c(A_1 - A_2) \\ \frac{V_2}{\beta} \cdot \frac{d_{P_2}}{dt} = Q_2 + Q_{le} - \dot{x}_c(A_1 - A_2) \end{cases} \tag{8}$$

where β is the elastic bulk modulus, Q_{le} is the hydraulic cylinder internal leakage flow, and V_1 and V_2 are the volume of hydraulic cylinder rod less chamber and rod chamber.

3.2. Pump Control Subsystem Modeling

In the pump control subsystem, the LS pump (Figure 4) angle is dynamically adjusted by the pump outlet pressure and load pressure. The processes that are involved are: (1) determining the maximum pressure of multiple actuators, (2) controlling the load-sensitive valve and pressure shutoff valve by the load pressure, (3) adjusting the swash plate angle by the variable piston cylinder, and (4) calculating the LS pump displacement based on the swash plate angle.

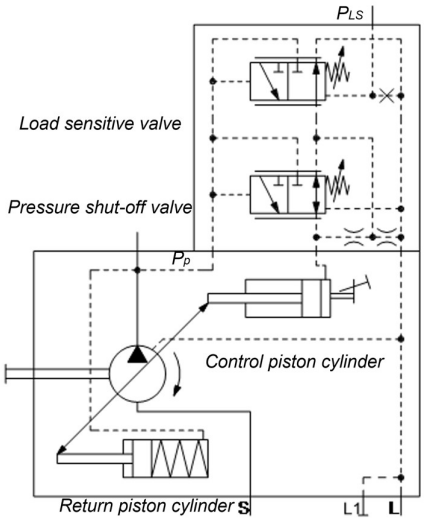


Figure 4. Schematic diagram of the LS pump.

3.2.1. Determining Load Pressure

The load pressure of each actuator and the accompanying pressure compensator valve is determined by the pressure-reducing valve when multiple actuators are operated in combination. The actuator compensates pressure ΔP_c, total pressure loss ΔP_{drop}, and pump output pressure P_p:

$$\begin{cases} \Delta P_c = P_{Lmax} - P_L \\ \Delta P_{drop} = P_p - P_L \\ P_p = P_{LS} + P_{Lmax} \end{cases} \tag{9}$$

where P_L and P_{Lmax} are the load pressure and the maximum load pressure, respectively, and P_{LS} is the pump LS pressure.

3.2.2. P_{LS} to Load Sensitive Valve/Pressure Shutoff Valve

When P_{LS} is determined, the P_{LS} is applied to the load-sensitive valve. At this time, the load-sensitive valve spool force balance equation is:

$$m_{ls}\frac{d^2x_{ls}}{dt^2} = A_{ls}(P_P - P_{ls}) - K_{ls}(x_l + x_{ls}) - B_{ls}\frac{dx_{ls}}{dt} \tag{10}$$

where m_{ls}, x_{ls}, A_{ls} are the load-sensitive valve spool mass, displacement, and effective area, respectively, K_{ls} is the load-sensitive valve spring stiffness, x_l is the spring pre-compression, B_{ls} is the piston viscous friction damping coefficient, and P_P is the outlet pressure of pump.

The flow equation of the LS valve is:

$$Q_{ls} = C_d A_{vs}\sqrt{\frac{2\Delta p_1}{\rho}} \tag{11}$$

where Δp_1 is the pressure difference of the valve, A_{vs} is the flow area of the valve, and C_d is the coefficient of flow.

With the pump outlet pressure rising, it acts on the pressure shutoff valve. The spool's force balance equation for the pressure shutoff valve is:

$$m_{ps}\frac{d^2x_{ps}}{dt^2} = A_{ps}P_P - K_{ps}(x_{l2} + x_{ps}) - B_{ps}\frac{dx_{ps}}{dt} \tag{12}$$

where m_{ps}, x_{ps}, and A_{ps} are the pressure shutoff valve spool mass, displacement, and effective area, respectively, K_{ps} is the pressure shutoff valve spring stiffness, x_{l2} is the spring pre-compression, and B_{ps} is the piston viscous friction damping coefficient.

Similarly, the flow equation for the pressure shutoff valve can be shown as:

$$Q_{ps} = C_d A_{ps}\sqrt{\frac{2\Delta p_2}{\rho}} \tag{13}$$

where A_{ps} and Δp_2 are the flow area and pressure difference of the valve, respectively.

3.2.3. Control Piston Cylinder to Swash Plate

The effective area is different between the two piston cylinders of the pump, and the swash plate angle can still be adjusted under the same force. As the rotation angle of the swash plate increases, the flow continuity equation of the control piston cylinder is:

$$Q_{vh} = A_{vh}\frac{dx_{vh}}{dt} + \frac{V_{vh}}{K_{vh}}\frac{d_{P_{vh}}}{dt} \tag{14}$$

The force balance equation of the swash plate is:

$$P_k A_k r_k - P_{vh} A_{vh} r_{vh} = I\frac{d^2x_k}{x_k dt^2} \tag{15}$$

where I is the swash plate rotational inertia.

When the swash plate rotation angle decreases, the flow continuity equation of the control piston cylinder is:

$$Q_{vh} = A_{vh}\frac{dx_{vh}}{dt} - \frac{V_{vh}}{K_{vh}}\frac{d_{P_{vh}}}{dt} \tag{16}$$

The force balance equation of swash plate is:

$$P_{vh} A_{vh} r_{vh} - P_k A_k r_k = I\frac{d^2x_{vh}}{x_{vh} dt^2} \tag{17}$$

where Q_{vh}, A_{vh}, and x_{vh} are the control piston cylinder flow, effective area, and displacement, respectively, r_{vh} and V_{vh} are the control piston cylinder volume and distance from the control piston cylinder to the rotation center, respectively, K_{vh} is the bulk modulus, and A_k, x_k, and r_k are the return piston cylinder effective area, displacement, and distance from the return piston cylinder to the rotation center, respectively.

The swash plate rotation angle is satisfied:

$$\tan\alpha \cos\omega d_0 = x_{vh} - x_k \tag{18}$$

where β is the swash plate tilt angle, and d_0 is the diameter of the plunger distribution circle.

3.2.4. Swash Plate to Pump Displacement

As the rotation angle of the swash plate changes, the displacement of the pump also varies. The displacement of the pump D can be expressed as:

$$D = \frac{\pi}{4} d^2 d_o z \tan\alpha \cos\theta \tag{19}$$

where d is the plunger diameter and z is the number of plungers.

Eventually, the flow continuity equation of the LS pump is:

$$\frac{V}{\beta} P_p = D\omega\eta_{pv} - C_P\left(P_p - P_o\right) \tag{20}$$

where V is the LS pump volume, ω is the pump speed, η_{pv} is the pump volumetric efficiency, and C_P is the leakage coefficient.

4. HAQPSO-PID Controller Design

4.1. PID Controller

Hydraulic excavator system adopts PID controller to achieve trajectory control of actuators. Its standard form is:

$$\begin{cases} u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \\ e(t) = y(t) - r(t) \end{cases} \tag{21}$$

where e is the error between the desired trajectory r and the actual trajectory y, u is the control signal, and K_p, K_i and K_d are the coefficients of the controller.

The PID controller adjusts PID parameters from control signal u based on the change in error e. Although the PID control structure is simple and widely used, the parameters of PID are difficult to adjust, which leads to poor control effects [23]. Therefore, the most critical aspect of the PID controller is the tuning of K_p, K_i and K_d.

4.2. Quantum-Behaved Particle Swarm Optimization

Due to the simple computation, fast convergence, cheap implementation, and few control parameters, PSO has been widely used in optimization problems since its introduction [14,15]. However, the PSO algorithm is prone to premature convergence and cannot guarantee convergence to the optimal global solution. To better solve the above questions, Sun [24] studied the research results on particle convergence behavior of particles by Clerc et al. The QPSO was proposed by applying the idea of quantum motion to particle behavior, which is free from the limitation of the traditional velocity–position orbit model.

The fluctuation function is applied in QPSO to describe the state of particles so that the moving particles have wave properties and the search is more continuous. In addition, the Schrödinger equation is used to obtain the probability density function of each individual at any position, and the position equation of particles in the problem space is gained by introducing the Monte Carlo simulation. In the iterative search process, the divergence of particles is constrained by the attractor of the best position center in the current iteration,

which improves the probability of particles' appearance in the problem space. The global search capabilities of population can be considerably improved by tracking the individual best and global optimal to update the solution. Thus, the QPSO method outperforms the PSO algorithm in terms of global search performance.

In the standard QPSO algorithm, the position update rule of each particle is:

$$X_i^j(t+1) = p_i^j(t) \pm \alpha \left| mbest^j(t) - X_i^j(t) \right| \ln\left(\frac{1}{u}\right) \tag{22}$$

where X is the location of particle, $i(i = 1, 2, 3, \cdots M)$ is the number of particles, j is the particle dimension, t is the number of iterations, $u = \text{rand}(0, 1)$, and $p_i^j(t)$ is the random location of particle i between the local optimal position and the global optimal position after t iterations. It is defined as:

$$p_i^j(t) = \varphi \cdot Pbest_i^j(t) + (1 - \varphi) \cdot Gbest_j(t) \tag{23}$$

where $\varphi = \text{rand}(0, 1)$, *Pbest* is the local optimal position of particles, and *Gbest* is the global optimal position of particles. The corresponding update rules are:

$$\begin{aligned} Pbest_i^j(t) &= \begin{cases} X_i^j(t) & f\left[X_i^j(t)\right] < f\left[Pbest_i^j(t-1)\right] \\ Pbest_i^j(t-1) & others \end{cases} \\ Gbest_i^j(t) &= \begin{cases} Pbest_i^j & f\left[Pbest_i^j(t)\right] < f\left[Gbest^j(t-1)\right] \\ Gbest^j(t-1) & others \end{cases} \end{aligned} \tag{24}$$

where f is the fitness function.

α is the contraction–expansion coefficient (*CE*), and it is:

$$\alpha = 0.5 + 0.5 \times \frac{t_{\max} - t}{t_{\max}} \tag{25}$$

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

m_{best} is the global optimal average local optimal position of the particles, and it is defined as:

$$m_{best} = \frac{1}{M} \sum_{i=1}^{M} Pbest_i^j(t) \tag{26}$$

4.3. Hybrid Adaptive Quantum-Behaved Particle Swarm Optimization

Although the standard QPSO has many advantages, it still has some shortcomings. The valve selection method and range of parameters of *CE* have a great influence on the algorithm performance. It was proved that $\alpha < 1.781$ can make the particles converge reliably [25]. As the complexity of the problem increases, it may still enter into local convergence prematurely. Moreover, the position of the initial population is generated randomly, which has an impact on operation efficiency.

The parameters of the PID controller are tuned based on QPSO, and its tuning algorithm should have high efficiency. Therefore, the initial population, search range, and calculation methods of m_{best}, φ, and *CE* with HAQPSO are optimized in this paper, and HAQPSO is established to overcome the shortcomings of QPSO. The specific improvements are as follows.

(1) Improve search range

To improve the operation efficiency of HAQPSO, the Ziegler–Nichols (Z-N) method is used to calculate the initial PID value, and the search range of PID parameters is selected near the initial values. The initial position of particles is randomly generated within the search range.

(2) Initialized population optimization based on improved circle chaotic mapping

Since the particles are randomly generated in the initialization stage, it leads to the problem that the initial solutions are prone to aggregation, poor coverage in the solution space, and low variability among individuals. In contrast, chaotic is defined as mathematical randomness produced by simple deterministic systems, and has ergodicity and semi stochastic properties, depending on initial conditions [26]. Relevant experimental research shows that chaotic mapping is more useful than using random numbers in optimization problems [27]. Therefore, chaotic mapping is applied in the QPSO algorithm to optimize the initial population, ensure its distribution uniformity and diversity, and improve the quality and accuracy of the initial solution.

Circle mapping, an excellent mapping function, is applied to QPSO to improve its global optimization performance. However, circle mapping is still unevenly distributed [28]. The original circle chaotic mapping expression is:

$$c_{i+1} = \mathrm{mod}(c_i + b - \left(\frac{a}{2\pi}\right) \sin(2\pi \cdot c_i), 1) \tag{27}$$

where a and b are circle mapping parameters, generally $a = 0.5$ and $b = 0.2$. When $i = 2000$, the initial solution dimension distribution and histogram of initial solution dimension distribution are shown in Figure 5.

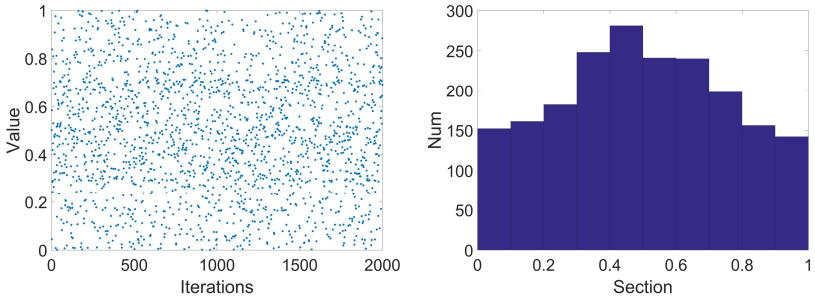


Figure 5. Distribution diagram and distribution histogram of circle mapping.

It can be seen from Figure 5 that the circle mapping is still unevenly distributed, with dense values between [0.4, 0.7]. Hence, the circle mapping formula has been improved to ensure the chaotic values are more uniformly distributed. The improved circle chaotic mapping expression is:

$$c_{i+1} = \mathrm{mod}(c_i + b - \left(\frac{a}{\pi}\right) \sin(\pi \cdot c_i) \cos(\pi \cdot c_i) + \mathrm{rand}(0,1), 1) \tag{28}$$

The improved initial distribution diagram and initial solution dimension distribution histogram are shown in Figure 6. Compared with the original circle chaotic mapping, the improved circle has a more uniform search range between [0–1] and the distribution is more uniform. Therefore, the improved circle mapping is used to initialize the population to enhance its diversity, which in turn improves the optimization ability of QPSO.

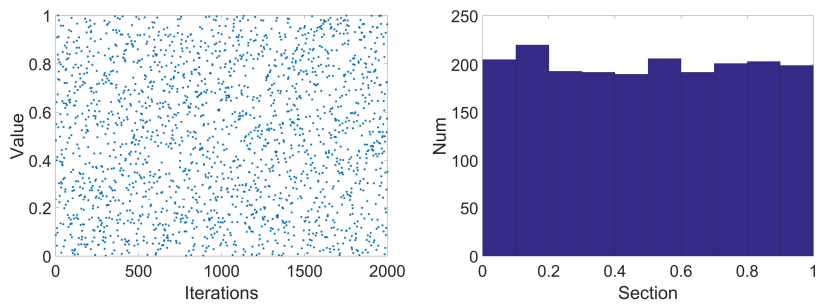


Figure 6. Distribution diagram and distribution histogram of improved circle mapping.

The improved circle chaotic mapping is introduced in QPSO, and the initial particle population is:

$$x_{ij} = c_i(ub - lb) + lb \tag{29}$$

where ub and lb are the upper and lower bounds of the search space, respectively.

(3) Gaussian distribution variational operator

Gaussian distribution is the most widely used of the continuous probability distributions. To advance the searching ability of QPSO, the mutation operation is increased in the optimization process. When the population falls into the local optimum, a disturbance is added to improve the global search performance. It is given by:

$$m_j(t) = m_{best} + \varepsilon\phi \tag{30}$$

where ε is a predetermined parameter, which is set to $\varepsilon = 0.001$ in this paper, and ϕ is a Gaussian random distribution with a mean of 0 and a variance of 1. Equation (22) becomes:

$$X_i^j(t+1) = p_i^j(t) \pm \alpha \left| m^j(t) - X_i^j(t) \right| \ln\left(\frac{1}{u}\right) \tag{31}$$

The mutation of the particle swarm decreases with the iteration of generations. It is applied to all particles at the beginning of the search to increase the search range of particles. During the middle of the search, it is applied to some particles for improving the convergence speed of the algorithm. At the end of the search, there is no more mutation, enabling the particles to perform a fine search in the vicinity of the optimal solution. Setting different mutation particles in different searching periods can effectively enhance the global search ability and searching speed of particles. Figure 7 shows the Gaussian mutation operation process when the number of iterations is 100.

(4) Adaptive weight φ

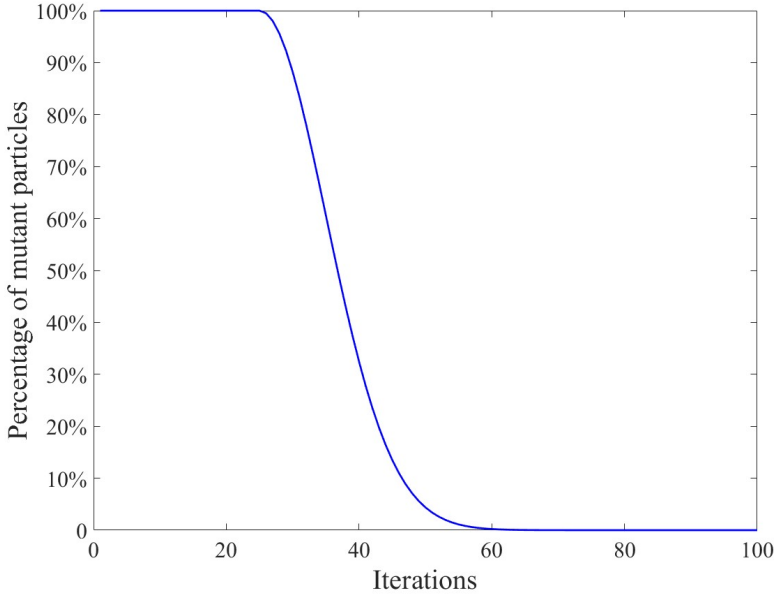


Figure 7. Gaussian mutation operation.

Analyzing the QPSO update mode, the optimal location of individual is mostly determined by φ. The global optimum and the individual optimum must be effectively balanced in order to achieve better optimization performance. However, the traditional QPSO is generated randomly and cannot guarantee the search for the global optimum. Therefore,

the random weight is converted to dynamic weight in the iterative process. The adaptive weight φ is defined as:

$$\varphi = \frac{\exp(f(j)/m)}{(1+\exp(-f(j)/m))^{iter}} \tag{32}$$

where φ is the average fitness value and $f(j)$ is the fitness value.

(5) Dynamic adjustment of *CE*

As the only control parameter that needs to be set manually, the control strategy of *CE* has a substantial impact on the performance of algorithm. *CE* decreases linearly with the increase of iterations in QPSO, which reduces the convergence speed and accuracy of the algorithm. However, the linear drop of *CE* in the early stages of particle search easily falls into the local ideal, whereas the linear decline of *CE* in the late stages of particle search may miss the optimal solution. *CE* should be dynamically changing during the search process. The adjustment function of *CE* is:

$$\alpha = \frac{1}{1+e^{20(iter/maxgen-0.7)}} \tag{33}$$

When the number of iterations is 100, the curves of *CE* in QPSO and HAQPSO are shown in Figure 8.

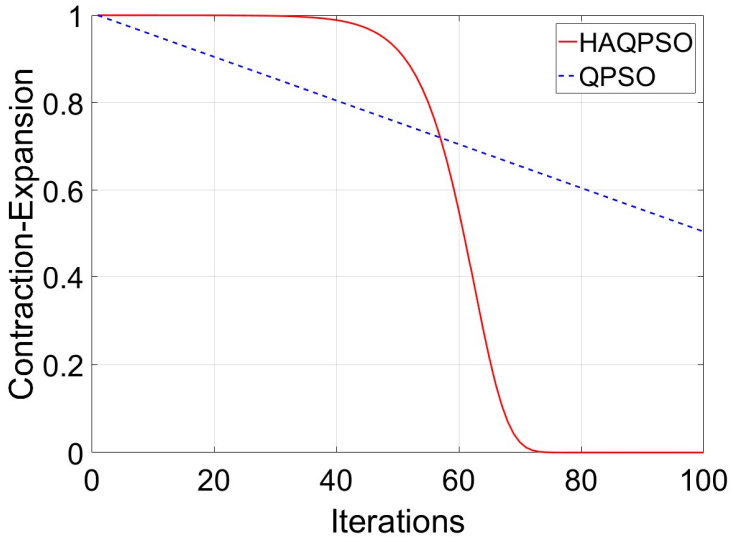


Figure 8. The comparison curves of CE in different methods.

Finally, the HAQPSO algorithm is obtained as shown below.

Algorithm HAQPSO Algorithm

1: Initialization: population size, maximum iteration number.
2: Determine the initial particle population according to Equation (29).
3: While current number of iterations is less than the maximum number of iterations,
4: Calculate the fitness value of each particle according to Equation (34);
5: Calculate *Pbest* for each particle and *Gbest* for the swarm;
6: Calculate best position m_{best} according to Equation (26);
7: Calculate φ and p_i^j according to Equations (23) and (32);
8: Calculate α and $m_j(t)$ according to Equations (30) and (33);
9: Update the position for each particle according to Equation (31);
10: Current number of iterations = current number of iterations + 1;
11: Output: Optimum PID parameters.

4.4. PID Parameter Tuning Based on HAQPSO

The purpose of the HAQPSO algorithm is to search the optimal PID gains for meeting the requirements of precise trajectory control for compound excavator operations. Every particle representing K_p, K_i and K_d files through a three-dimensional search space to seek the optimal position by the HAQPSO algorithm. In the process of optimization, the convergence of the algorithm to the global optimal solution is determined by the objective function. Commonly used objective functions include integral squared error (ISE), integral absolute error (IAE) and time integral and absolute error (ITAE) [14,15,28]. In order to obtain good transient response and minimum steady-state error of the controlled object, the minimum value of ITAE is the objective function. Meanwhile, the objective function adds the square of the input control signal to avoid excessive control. The parameters of the PID controller are optimized online by the HAQPSO algorithm to realize the precise control for the multi-actuator compound trajectory of LS hydraulic excavator. The fitness function f and the objective function J can be written as:

$$f = 1/J = \int_0^\infty \left(\omega_1|e(t)| + \omega_2 u^2(t)\right)dt \tag{34}$$

where $e(t)$ is the error value between the expected trajectory and the actual trajectory, $u(t)$ is the controller output, and ω_1 and ω_2 are the corresponding weights.

The flow of HAQPSO tuning PID parameters is shown in Figure 9.

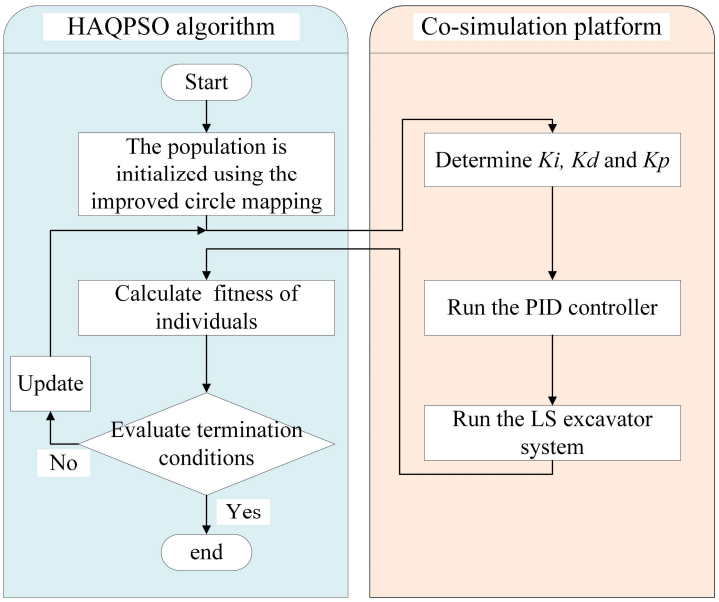


Figure 9. Framework of HAQPSO tuning PID controller parameters.

5. Hydraulic Excavator Co-simulation Platform

Controller performance tests performed directly on excavator are generally time-consuming, labor-intensive and dangerous. Therefore, a simulation model of the LS excavator is established, and the performance of the HAQPSO algorithm to optimize the PID controller in multi-actuator compound trajectory control of the LS hydraulic excavator is tested online. An electrohydraulic co-simulation platform of the LS excavator is constructed, as shown in Figures 10 and 11. The platform is built according to the mechanical structure and hydraulic principles of the excavator, which can effectively reflect some nonlinear factors in the hydraulic system. Many studies have demonstrated the effectiveness and reliability of combined simulation in studying the performance of construction machinery [29–31].

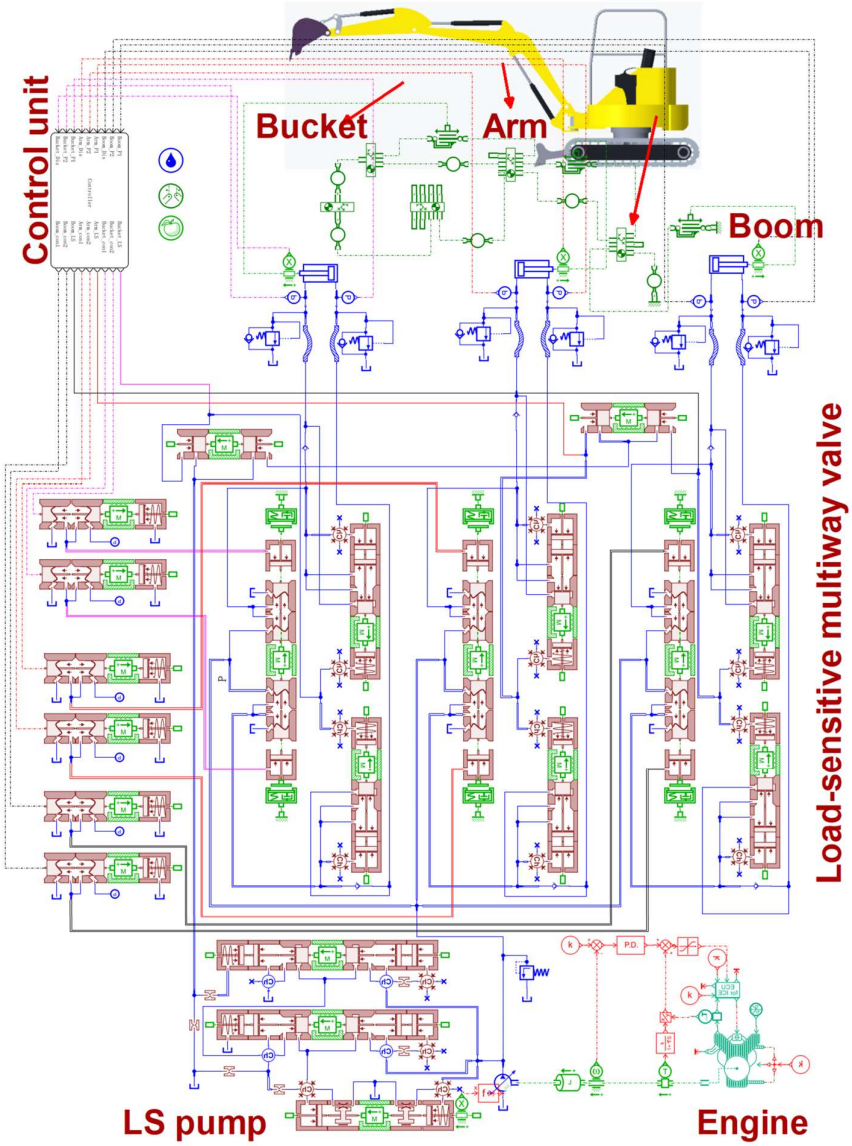


Figure 10. Co-simulation platform of LS excavator.

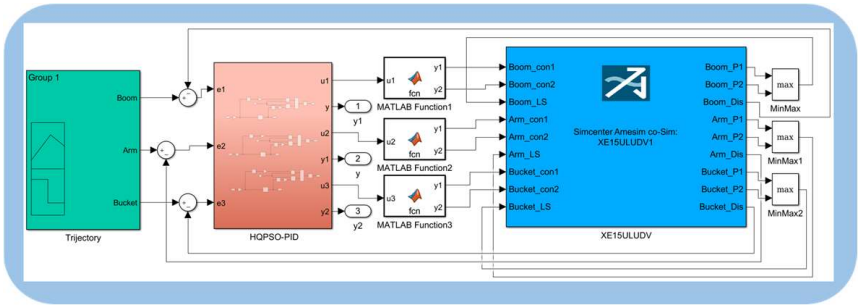


Figure 11. Co-simulation platform of controller.

The LS excavator model is built in AMESIM. The hydraulic system consists of a hydraulic component design (HCD) library and hydraulic components established according to parameters, including the LS pump, multiway valve, shuttle valve, hydraulic cylinder, oil tank, oil pipe, etc. The mechanical system is built by the mechanical library, which is based on mapping the actual mass, size, and moment of inertia of each actuator. The control strategy is constructed in Matlab/Simulink through co-simulation control. The

box section is the interface between AMEsim and Matlab/Simulink. Table 1 lists the main parameters in the platform.

Table 1. Basic parameters of the co-simulation platform.

Parameter	Value	Unit
Oil density	850	kg/m^3
Oil bulk modulus	17,000	bar
Engine speed	2300	r/min
Pump maximum displacement	34	cc/rev
Pump volumetric efficiency	0.92	
Pump hydraulic–mechanical efficiency	0.96	
Relief valve cracking pressure	270	bar
Pressure compensation valve piston diameter	10	mm
Pressure compensation valve rod diameter	5	mm
Spring force at zero displacement	37.8	N
Spring rate	20	N/mm
Coefficient of pressure compensating valve viscous friction	50	N/m/s
Cylinder leakage coefficient	0.01	L/min/bar
Piston diameter of the boom hydraulic cylinder	74	mm
Rod diameter of boom hydraulic cylinder	34	mm
Travel of boom hydraulic cylinder	650	mm
Boom mass	60.94	kg
Moment of boom inertia	1245.46	kg/m^2
Piston diameter of the arm hydraulic cylinder	70	mm
Rod diameter of arm hydraulic cylinder	34	mm
Travel of arm hydraulic cylinder	503	mm
Arm mass	28.38	kg
Moment of arm inertia	127.57	kg/m^2
Piston diameter of the bucket hydraulic cylinder	75	mm
Rod diameter of bucket hydraulic cylinder	34	mm
Travel of bucket hydraulic cylinder	410	mm
Bucket mass	16.55	kg
Moment of Bucket inertia	29.32	kg/m^2

6. Experiment Results and Discussion

6.1. Performance Analysis of HAQPSO Performance

To verify the performance of the HAQPSO algorithm, the experiments are performed using four classical benchmark functions. These benchmark functions are widely used in numerical optimization algorithms [32–34], where Sphere and Rosenbrock are the unimodal functions and Rastrigin and Ackley are multimodal functions. The settings of the functional form, search range, and initialization range of the benchmark functions are shown in Table 2. n denotes the dimension of the solution space. The compared optimization algorithms are PSO, QPSO, and HAQPSO.

For all compared algorithms, experiments are conducted with particle populations N of 20, 50, and 80, particle dimensions D of 10, 30, and 50, and maximum iterations T of 1000, 1500, and 2000, respectively. Each case is run 100 times separately and the average value of the optimal solution is calculated to test the difference between the algorithms' solution accuracy and stability. The convergence criterion is set to the maximum number of iterations. All experiments are performed using MATLAB 2016b on a personal computer with an Intel Core i7-9700F 3.0 GHz CPU with 16 GB memory. The final results of each benchmark function are shown in Tables 3–6.

Table 2. Classical high-dimension benchmark functions.

Name	Function Expression	Search Domain	Initial Range
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$	[50, 100]
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$	$[-30, 30]^n$	[15, 30]
Rastrigin	$f_4(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^n$	[2.56, 5.12]
Ackley	$f_5(x) = 20 + e - 20e(-\frac{1}{5}\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - e(-\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)$	$[-32, 32]^n$	[16, 32]

Table 3. Test results of Sphere.

N	*D*	*T*	Algorithm PSO	Algorithm QPSO	Algorithm HAQPSO
20	10	1000	2.4332×10^{-6}	3.2146×10^{-28}	1.1313×10^{-43}
	30	1500	9.8434×10^{-4}	6.5730×10^{-17}	7.047×10^{-17}
	50	2000	5.9672×10^{-3}	5.6147×10^{-5}	1.6556×10^{-7}
50	10	1000	5.4372×10^{-10}	4.4054×10^{-35}	1.2969×10^{-60}
	30	1500	2.7452×10^{-7}	1.0317×10^{-23}	2.0601×10^{-43}
	50	2000	4.6132×10^{-4}	1.9244×10^{-8}	1.8286×10^{-19}
80	10	1000	1.2378×10^{-12}	4.4054×10^{-46}	1.5692×10^{-97}
	30	1500	2.0232×10^{-4}	2.1565×10^{-24}	2.1925×10^{-52}
	50	2000	8.4451×10^{-2}	1.6424×10^{-11}	1.8021×10^{-25}

Table 4. Test results of Rosenbrock.

N	*D*	*T*	Algorithm PSO	Algorithm QPSO	Algorithm HAQPSO
20	10	1000	16.0366	15.4821	8.5116
	30	1500	46.2074	47.8458	37.3123
	50	2000	156.7582	104.8860	82.8742
50	10	1000	3.9565	5.1495	2.9243
	30	1500	36.6328	45.4307	23.2881
	50	2000	54.7541	49.9782	34.9334
80	10	1000	3.9058	3.8884	2.5978
	30	1500	32.7823	45.9152	16.8176
	50	2000	47.9091	45.9244	28.0241

Table 5. Test results of Rastrigin.

N	*D*	*T*	Algorithm PSO	Algorithm QPSO	Algorithm HAQPSO
20	10	1000	8.6263	6.3694	5.1833
	30	1500	26.8967	22.3258	12.3823
	50	2000	41.4143	28.7685	22.2701
50	10	1000	4.0992	3.7285	2.4873
	30	1500	31.3259	23.4073	22.3915
	50	2000	35.8055	24.8728	18.7974
80	10	1000	3.2351	2.3979	1.9899
	30	1500	12.4536	8.6923	9.5942
	50	2000	25.6276	17.7753	16.3971

Table 6. Test results of Ackley.

N	*D*	*T*	Algorithm PSO	QPSO	HAQPSO
20	10	1000	5.6949×10^{-4}	1.4460×10^{-14}	4.4409×10^{-15}
	30	1500	0.0369	0.0061	9.4508×10^{-8}
	50	2000	0.0985	0.6591	2.8150×10^{-4}
50	10	1000	2.1999×10^{-6}	1.5099×10^{-14}	5.0804×10^{-15}
	30	1500	0.0059	3.1556×10^{-7}	1.9984×10^{-13}
	50	2000	0.0254	0.0057	1.5727×10^{-9}
80	10	1000	8.1035×10^{-8}	4.5409×10^{-15}	4.4119×10^{-15}
	30	1500	0.0016	9.3718×10^{-12}	1.5099×10^{-14}
	50	2000	0.0097	3.2142×10^{-5}	1.4255×10^{-12}

As seen from Tables 3–6, for unimodal functions, the HAQPSO achieves better results than PSO and QPSO. The QPSO results are generally better than PSO, though in the Rosenbrock function, QPSO results are inferior to PSO at 50 particles. When the number of particles is fixed, the results of PSO, QPSO, and HAQPSO all increase with increases in dimension and iteration number. When the dimension and iteration number are determined, the algorithm results decrease with the increase in particle numbers. The two unimodal functions show the same pattern. For multimodal functions, the results of HAQPSO still outperformed those of PSO and QPSO for different particle numbers, different dimensions and different iterations, and both multimodal functions show the same variation pattern as the unimodal functions. Through the above analysis, it is demonstrated that the proposed HAQPSO algorithm outperforms PSO and QSPO in terms of optimization ability.

To vividly illustrate the performance advantages of HAQPSO, the convergence processes of the average fitness values for the four functions are shown in Figures 12–15. It can be seen in Figure 12 that the optimal result of HAQPSO algorithm is better than QPSO and PSO. However, the convergence speed of HAQPSO and QPSO is lower than PSO in the early stage. With the increase in iteration numbers, the QPSO and HAQPSO algorithms converge faster than PSO. Both algorithms can produce better solutions in the final stage of searching, and the results of HAQPSO algorithm are better. Therefore, HAQPSO is more advantageous than QPSO. In Figure 13, the same pattern is observed in the Rosenbrock function test: HAQPSO is superior to QPSO and PSO in the optimal results. In addition, the initial solution of HAQPSO is significantly better than PSO and QPSO. This shows that the added circle mapping proposed can optimize the range of initial solutions.

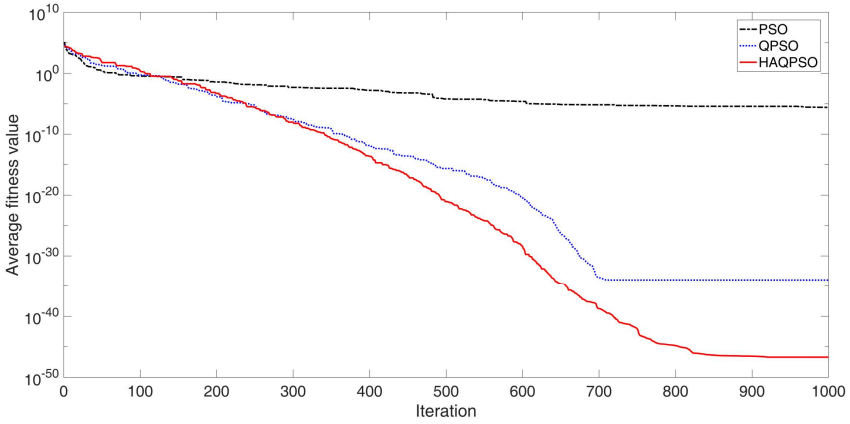


Figure 12. Convergence process of Sphere function (M = 20, D = 10, T = 1000).

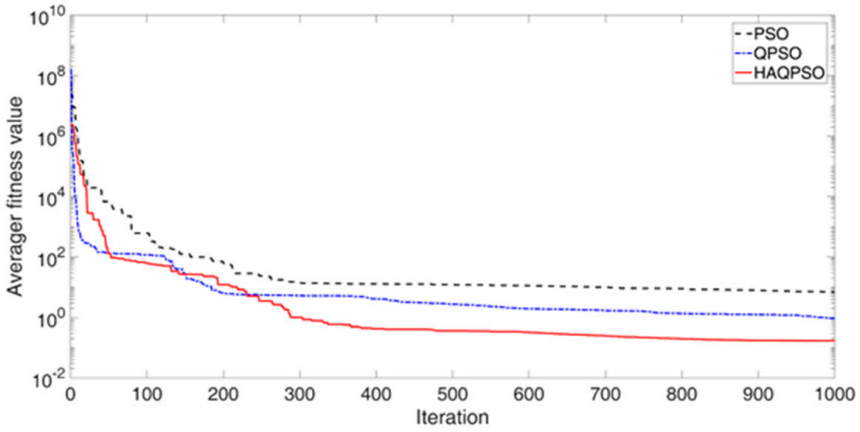


Figure 13. Convergence process of Rosenbrock function (M = 80, D = 10, T = 1000).

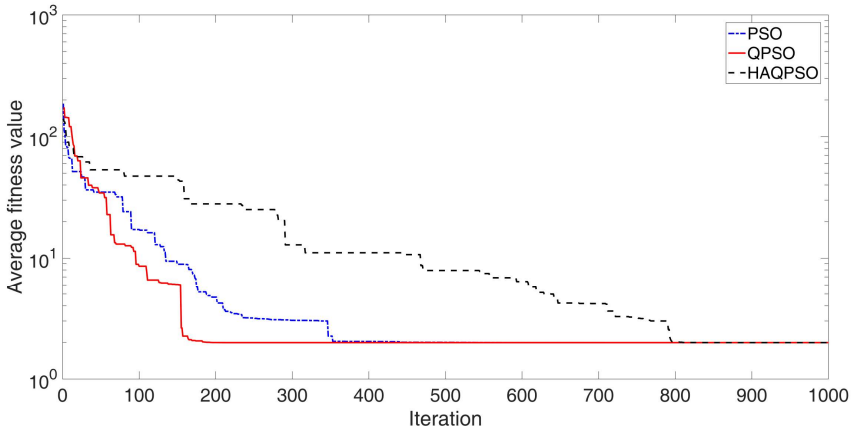


Figure 14. Convergence process of Rastrigin function (M = 20, D = 10, T = 1000).

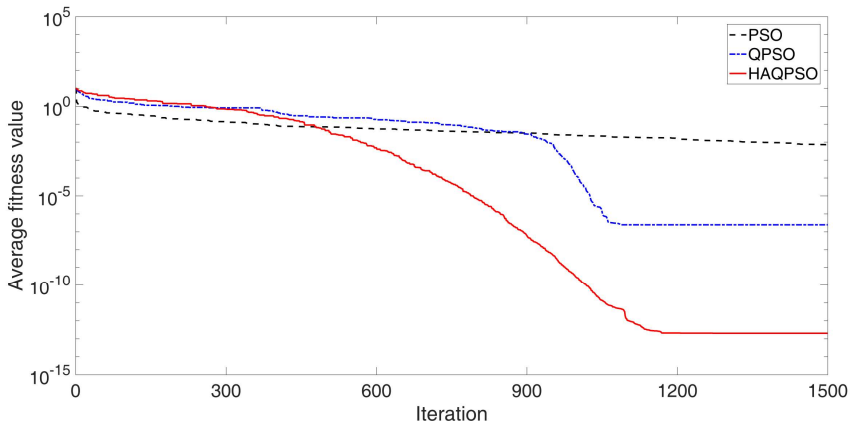


Figure 15. Convergence process of Ackley function (M = 50, D = 30, T = 1500).

In Figure 14, the optimal result of HAQPSO is also best in Rastrigin function, but the effect is not obvious. However, HAQPSO has the smallest number of iterations to search for the optimal solution. The optimal results of HAQPSO are also obviously better than QPSO and PSO in Figure 15. As with the Sphere function, the initial iteration speed of QPSO and HAPQSO is not as fast as that of PSO, but in the middle and late stages, HAQPSO is able to produce more optimal solutions. This indicates that HAQPSO does not fall into local convergence, and the particles enter a small area of fine search with better convergence accuracy than PSO and QPSO under the effect of adaptive learning. It is further verified that HAQPSO has better search accuracy and global convergence than PSO and QPSO.

6.2. Simulation Model Validation

The accuracy of the above co-simulation platform is verified by an LS hydraulic excavator, as shown in Figure 16. The joystick angle (Figure 17) is utilized as the input signal, and the same data as in the experiment are used. The experiment is a complete extension–retraction action of the bucket. Figure 18 shows the simulation results and the actual results. It can be seen that the co-simulation platform can effectively simulate the excavator actuator action. Therefore, in order to improve the efficiency, the co-simulation platform can effectively evaluate the performance of different controllers in controlling the excavator multiple actuators' compound trajectory.



Figure 16. Experimental platform of LS excavator.

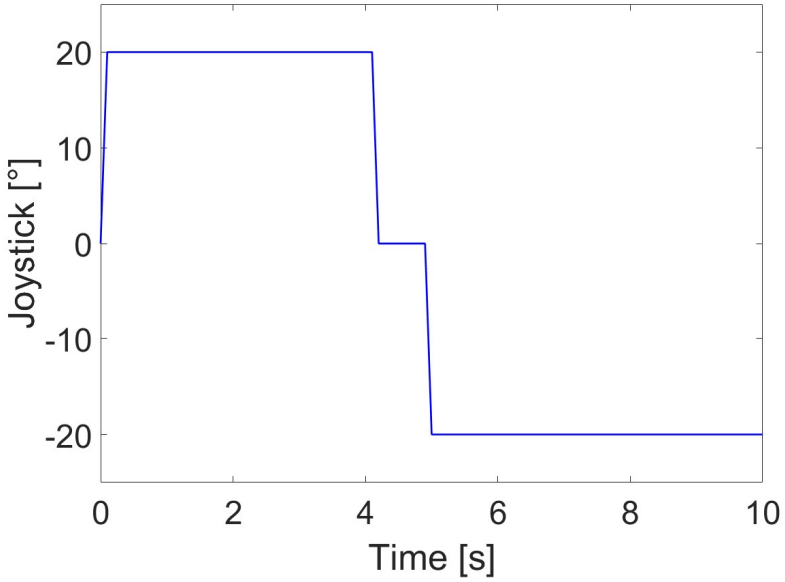


Figure 17. Input signal of simulation model.

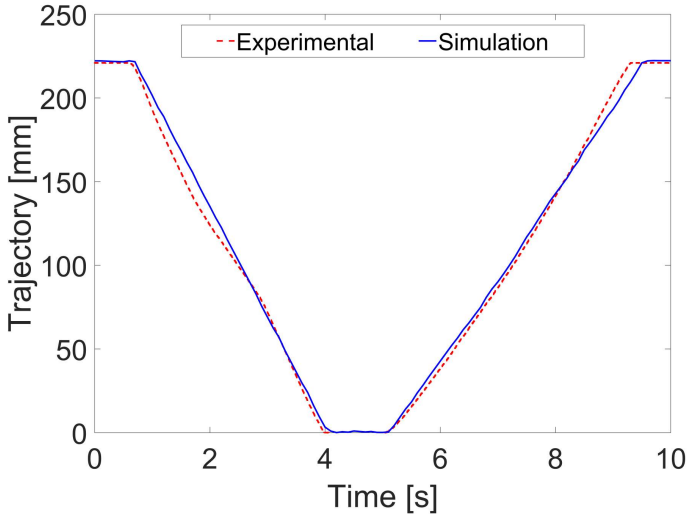


Figure 18. Comparison of trajectory between experiment and simulation.

6.3. Trajectory Control Experiment

During excavator operation, the most frequent action is the actuator compound operation, which completes the excavator descending, internal retraction, ascending, and external swing operation cycle. To verify the performance of the proposed HAQPSO-PID controller in excavator dynamic trajectory, comparative experiments are carried out with different approaches: HAQPSO-PID, ZN-PID, PSO-PID, and QPSO-PID. The desired trajectory signals of boom, arm, and bucket are also provided, and their desired trajectories are used as position inputs to evaluate the efficacy of different excavator trajectory control methods. Table 7 shows the parameter-tuning results of the ZN-PID controller.

Table 7. Parameter optimization results by Z-N tuning method.

	Kp	***Ki***	***Kd***
Boom	23.28	6.48	3.63
Arm	21.96	5.37	2.17
Bucket	19.64	5.89	2.48

The main operating parameters that need to be determined in QPSO include population size, the number of iterations, and the range of search. To the best of our knowledge, there is no mature theory to choose reasonable values for these parameters [35], but ranges have been studied in some research [34,36–38]. According to the results of the literature discussion, the parameters of PSO-PID, QPSO-PID, and HAQPSO-PID are established as shown in Table 8.

Table 8. Parameters of PSO-PID, QPSO-PID and HAQPSO-PID.

Parameter	**PSO-PID**	**QPSO-PID**	**HAQPSO-PID**
Population size	30	30	30
Maximum iteration number	100	100	100
Search range of *Kp*	[0, 40]	[0, 40]	/
Search range of *Ki*	[0, 10]	[0, 10]	/
Search range of *Kd*	[0, 10]	[0, 10]	/
Learning coefficient 1	2	/	/
Learning coefficient 2	2	/	/
[Vmin Vmax]	[0, 0.9]	/	/
Inertia weight	0.6	/	/
ω_1	0.999	0.999	0.999
ω_2	0.001	0.001	0.001

The trajectory and trajectory error curves of boom, arm and bucket are shown in Figure 19, and it can be seen that HAQPSO-PID and ZN-PID, PSO-PID and QPSO-PID have certain control effects on the trajectories of multiple actuators. However, the maximum error of all trajectories occurs in the actuator reversing stage due to the sudden change in pressure and flow. Meanwhile, the trajectory control precision interferes with others due to the mutual coupling of the three actuators. The trajectory error of HAQPSO-PID is better than the other three approaches for boom, arm and bucket, as shown in Figure 20. Taking boom as an example, the trajectory error range of HAQPSO-PID is only 0.0403 m, which is smaller than the 0.0601 m of ZN-PID, 0.0549 m of PSO-PID, and 0.0445 m of QPSO-PID, and the error accuracy is improved by 32.95%, 26.59% and 9.44%, respectively. Therefore, HAPSO-PID has the highest control accuracy in terms of the trajectory error curve.

Figure 20 shows the iteration curves of HAQPSO-PID, QPSO-PID and PSO-PID controllers for the boom, arm and bucket trajectories after 100 iterations. Table 9 shows the corresponding optimal *J*, number of iterations, and rectified PID parameters. Figure 20 and Table 9 shows that the best *J* and iteration number of HAQPSO-PID are smaller than those of PSO-PID and QPSO-PID. Taking the boom as an example, HAQPSO-PID can be converged after 27 generations with an optimal fitness value of 22.1850, while PSO-PID needs 62 generations to complete convergence, and the optimal fitness value is 21.9931. QPSO-PID needs 43 generations, with a best-fit value of 21.9931. In summary, the comparative analysis verifies that the HAQPSO-PID controller proposed in this paper has the best performance.

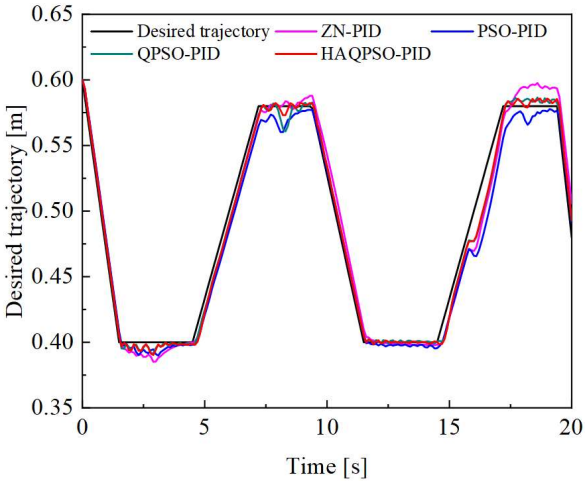
Table 9. Parameter optimization results by different tuning methods.

Tuning Method	Indicator	Boom	Arm	Bucket
PSO-PID	Best J	22.19	19.84	26.13
	Number of iterations	62	59	64
	Kp	21.08	20.63	23.09
	Ki	5.53	4.17	3.24
	Kd	3.73	3.96	4.35
QPSO-PID	Best J	21.99	19.64	25.62
	Number of iterations	43	51	65
	Kp	22.61	21.57	21.83
	Ki	4.62	6.75	5.79
	Kd	4.18	3.37	2.35
HQPSO-PID	Best J	21.51	19.37	24.79
	Number of iterations	27	38	57
	Kp	22.34	19.98	22.73
	Ki	6.97	8.34	4.52
	Kd	4.47	4.69	3.65

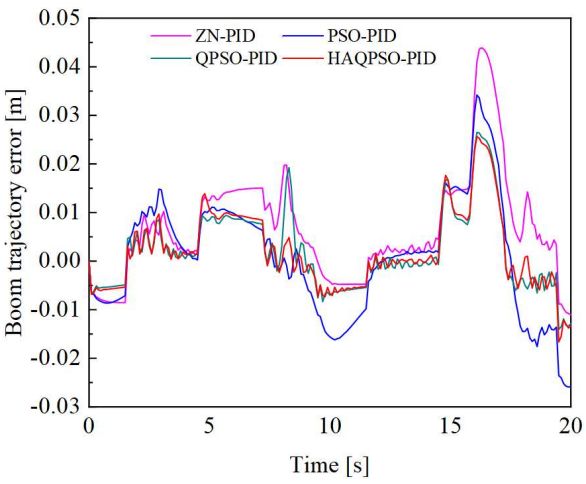
To further prove the excavator trajectory control accuracy of the HAQPSO-PID controller, the root mean square error (RMSE) of the bucket, arm, and boom trajectory is calculated, as shown in Table 10. The results show that among the four different PID controllers, the RMSE of HAQPSO-PID is the smallest, which indicates that HAQPSO-PID has good trajectory control performance and further verifies the effectiveness of the HAQPSO-PID method proposed in this paper.

Table 10. Root mean square error of different methods.

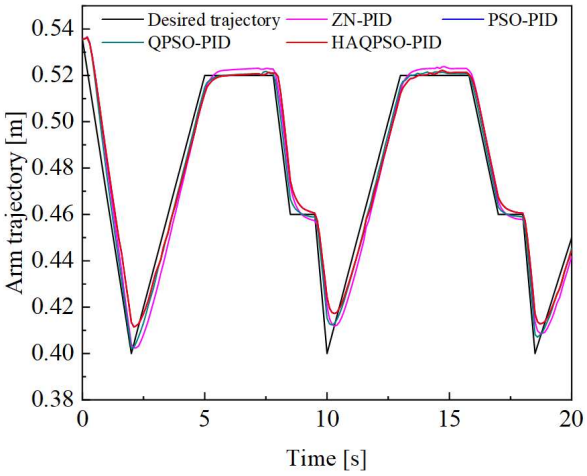
Methods	RMSE		
	Boom	Arm	Bucket
ZN-PID	12.9459	10.2999	44.9140
PSO-PID	10.0977	9.6542	34.3970
QPSO-PID	8.3153	7.7642	29.6588
HAQPSO-PID	8.1798	6.6442	24.8488



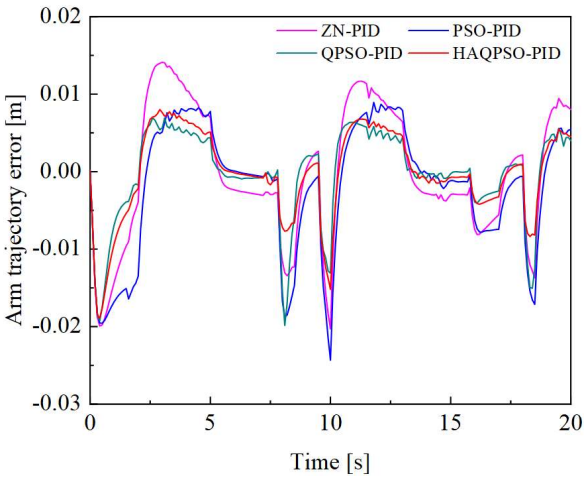
(**a**) Boom trajectory curve



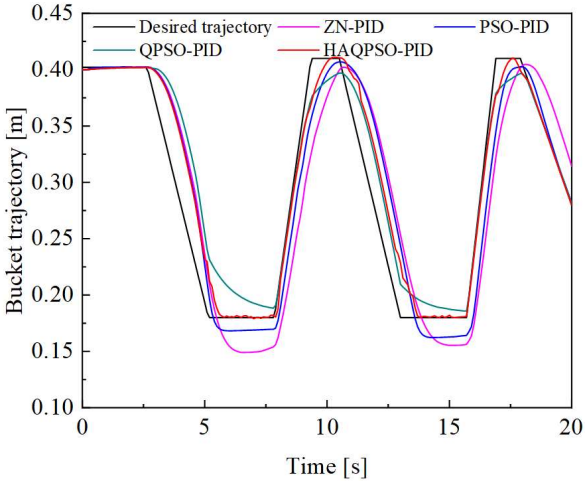
(**b**) Boom trajectory error curve



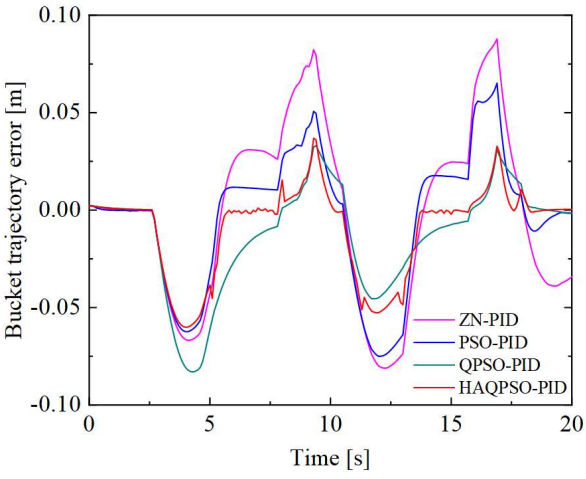
(**c**) Arm trajectory curve



(**d**) Arm trajectory error curve



(**e**) Bucket trajectory curve



(**f**) Bucket trajectory error curve

Figure 19. Experimental results of actuators under compound operation.

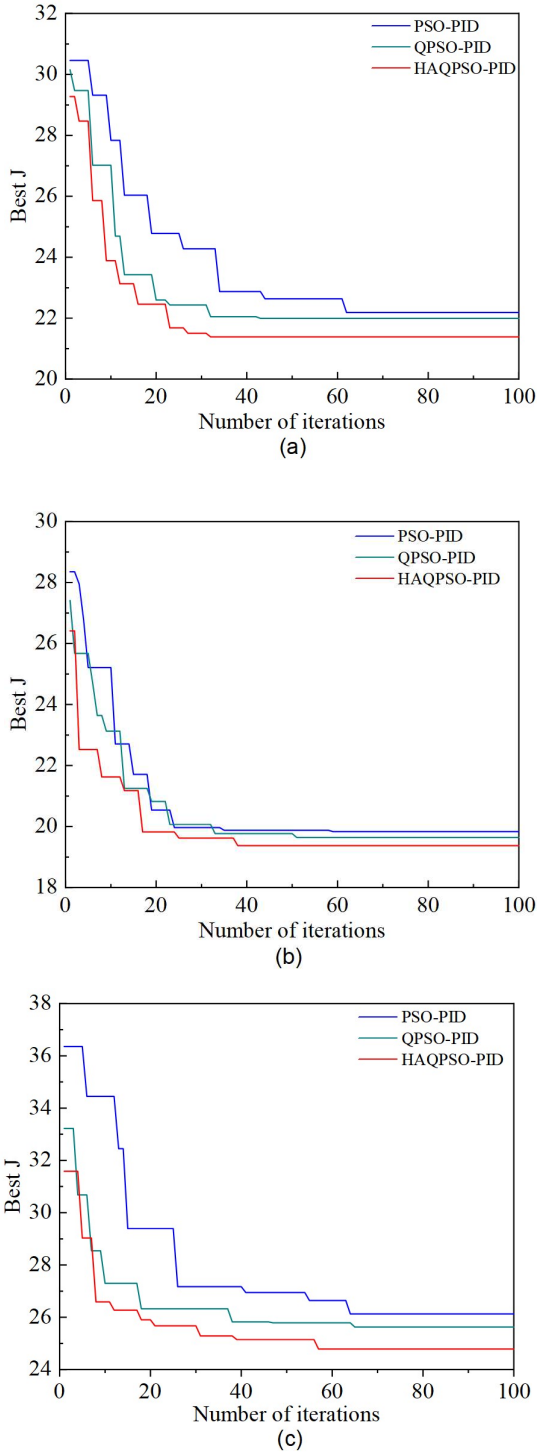


Figure 20. Iteration curves of different tuning methods with different actuators. (**a**) Iteration curves of different tuning methods with boom, (**b**) Iteration curves of different tuning methods with arm, (**c**) Iteration curves of different tuning methods with bucket.

From what has been discussed above, the control accuracy, convergence speed, and other indicators are improved in the trajectory control of the excavator using the proposed HAQPSO-PID controller. The effectiveness of this method is verified.

7. Conclusions

In this research, an HAQPSO-PID controller is proposed to improve the accuracy of multi-actuator compound trajectory control of LS hydraulic excavator. The HAQPSO combines QPSO with circle chaotic mapping, Gaussian mutation operators, and adaptive adjustment to avoid premature convergence in the basic QPSO algorithm. The results show that the proposed HAQPSO algorithm can better tune the PID controller. Considering the nonlinear relationship in the excavator hydraulic system, the LS hydraulic excavator model with pump control subsystem and valve control subsystem is established. Combined with the parameters measured by the real excavator, a co-simulation platform is constructed. The accuracy of the co-simulation platform is verified through comparing experiments and simulations.

The trajectory control experiments of boom, arm, and bucket are carried out based on the co-simulation platform. Comparative analysis results for the trajectory error curve, trajectory error range, number of iterations, and RMSE show that HAQPSO-PID has higher control accuracy and faster convergence than other control methods. The accuracy of the excavator trajectory control can be effectively improved. The effectiveness of the HAQPSO algorithm in setting PID controller parameters in excavator nonlinear system position control is revealed. On the premise of ensuring the accurate control of excavator trajectory, how to reduce its energy consumption will be a future research direction.

Author Contributions: Conceptualization, H.S. and G.L.; methodology, H.S. and X.X.; data curation, Z.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China, (Grant No.2020YFB1709902).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank Qiang Qin, Yubing Xu, and others of XCMG Excavator Machinery Business Department for their help in this research, and express sincere appreciation to the reviewers of this paper for their helpful comments.

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

References

1. Do, T.C.; Dang, T.D.; Dinh, T.Q.; Ahn, K.K. Developments in energy regeneration technologies for hydraulic excavators: A review. *Renew. Sustain. Energy Rev.* **2021**, *145*, 111076. [CrossRef]
2. Reginald, N.; Seo, J.; Cha, M. Integrative Tracking Control Strategy for Robotic Excavation. *Int. J. Control Autom. Syst.* **2021**, *19*, 3435–3450. [CrossRef]
3. Chen, C.; Zhu, Z.; Hammad, A. Automated excavators activity recognition and productivity analysis from construction site surveillance videos. *Autom. Constr.* **2020**, *110*, 103045. [CrossRef]
4. Lee, C.S.; Bae, J.; Hong, D. Contour control for leveling work with robotic excavator. *Int. J. Precis. Eng. Manuf.* **2013**, *14*, 2055–2060. [CrossRef]
5. Shen, W.; Wang, J. Adaptive Fuzzy Sliding Mode Control Based on Pi-sigma Fuzzy Neutral Network for Hydraulic Hybrid Control System Using New Hydraulic Transformer. *Int. J. Control Autom. Syst.* **2019**, *17*, 1708–1716. [CrossRef]
6. Zabihifar, S.H.; Yushchenko, A.S.; Navvabi, H. Robust control based on adaptive neural network for Rotary inverted pendulum with oscillation compensation. *Neural Comput. Appl.* **2020**, *32*, 14667–14679. [CrossRef]
7. Kim, J.; Jin, M.; Choi, W.; Lee, J. Discrete time delay control for hydraulic excavator motion control with terminal sliding mode control. *Mechatronics* **2019**, *60*, 15–25. [CrossRef]
8. Dao, H.V.; Na, S.; Nguyen, D.G.; Ahn, K.K. High accuracy contouring control of an excavator for surface flattening tasks based on extended state observer and task coordinate frame approach. *Autom. Constr.* **2021**, *130*, 103845. [CrossRef]
9. Park, J.; Cho, D.; Kim, S.; Kim, Y.B.; Kim, P.Y.; Kim, H.J. Utilizing online learning based on echo-state networks for the control of a hydraulic excavator. *Mechatronics* **2014**, *24*, 986–1000. [CrossRef]
10. Hua, H.; Fang, Y.; Zhang, X.; Qian, C. Auto-tuning nonlinear PID-type controller for rotorcraft-based aggressive transportation. *Mech. Syst. Signal Process.* **2020**, *145*, 106858. [CrossRef]

11. Lui, D.G.; Petrillo, A.; Santini, S. An optimal distributed PID-like control for the output containment and leader-following of heterogeneous high-order multi-agent systems. *Inf. Sci.* **2020**, *541*, 166–184. [CrossRef]
12. Zhang, S.; Minav, T.; Pietola, M.; Kauranne, H.; Kajaste, J. The effects of control methods on energy efficiency and position tracking of an electro-hydraulic excavator equipped with zonal hydraulics. *Autom. Constr.* **2019**, *100*, 129–144. [CrossRef]
13. Do, T.C.; Tran, D.T.; Dinh, T.Q.; Ahn, K.K. Tracking Control for an Electro-Hydraulic Rotary Actuator Using Fractional Order Fuzzy PID Controller. *Electronics* **2020**, *9*, 926. [CrossRef]
14. Ye, Y.; Yin, C.-B.; Gong, Y.; Zhou, J.-J. Position control of nonlinear hydraulic system using an improved PSO based PID controller. *Mech. Syst. Signal Process.* **2017**, *83*, 241–259. [CrossRef]
15. Feng, H.; Ma, W.; Yin, C.; Cao, D. Trajectory control of electro-hydraulic position servo system using improved PSO-PID controller. *Autom. Constr.* **2021**, *127*, 103722. [CrossRef]
16. Zhang, H.; Li, L.; Zhao, J.; Zhao, J. The hybrid force/position anti-disturbance control strategy for robot abrasive belt grinding of aviation blade base on fuzzy PID control. *Int. J. Adv. Manuf. Technol.* **2021**, *114*, 3645–3656. [CrossRef]
17. Cao, F. PID controller optimized by genetic algorithm for direct-drive servo system. *Neural Comput. Appl.* **2018**, *32*, 23–30. [CrossRef]
18. Zhang, C.; Ding, S. A stochastic configuration network based on chaotic sparrow search algorithm. *Knowl. Based Syst.* **2021**, *220*, 106924. [CrossRef]
19. Abualigah, L.; Yousri, D.; Abd Elaziz, M.; Ewees, A.A.; Al-qaness, M.A.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
20. Van den Bergh, F. An Analysis of Particle Swarm Optimizers. Ph.D. Thesis, University of Pretoria, Pretoria, South Africa, 2001.
21. Shi, Y.; Li, W.; Lu, P.; Chen, F.; Qi, X.; Xiong, C. Research on hydraulic motor control system based on fuzzy neural network combing sliding mode control and time delay estimation. *J. Intell. Fuzzy Syst.* **2022**, *43*, 3815–3826. [CrossRef]
22. Fu, S.; Wang, L.; Lin, T. Control of electric drive powertrain based on variable speed control in construction machinery. *Autom. Constr.* **2020**, *119*, 103281. [CrossRef]
23. Nie, Z.Y.; Zhu, C.; Wang, Q.G.; Gao, Z.; Shao, H.; Luo, J.L. Design, analysis and application of a new disturbance rejection PID for uncertain systems. *ISA Trans.* **2020**, *101*, 281–294. [CrossRef] [PubMed]
24. Sun, J.; Xu, W.B.; Feng, B. A global search strategy of quantum-behaved particle swarm optimization. In Proceedings of the 2004 IEEE Conference of Cybernetics and Intelligent Systems, Singapore, 1–3 December 2004.
25. Sun, J.; Fang, W.; Wu, X.J.; Palade, V.; Xu, W.B. Quantum-Behaved Particle Swarm Optimization: Analysis of Individual Particle Behavior and Parameter Selection. *Evol. Comput.* **2012**, *20*, 349–393. [CrossRef] [PubMed]
26. Varol Altay, E.; Alatas, B. Bird swarm algorithms with chaotic mapping. *Artif. Intell. Rev.* **2019**, *53*, 1373–1414. [CrossRef]
27. Caponetto, R.; Fortuna, L.; Fazzino, S.; Xibilia, M.G. Chaotic sequences to improve the performance of evolutionary algorithms. *IEEE Trans. Evol. Comput.* **2003**, *7*, 289–304. [CrossRef]
28. Arora, S.; Anand, P. Chaotic grasshopper optimization algorithm for global optimization. *Neural Comput. Appl.* **2018**, *31*, 4385–4405. [CrossRef]
29. Xing, H.-Y.; Zhao, J.-Y.; Yin, X.; Ding, H.-G.; Meng, X.-Z.; Li, J.-S.; Wang, G.-Y. Simulated research on large-excavator boom based on hydraulic energy recovery. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2021**, *236*, 10690–10700. [CrossRef]
30. He, C.; Wang, J.; Wang, R.; Zhang, X. Research on the characteristics of hydraulic wind turbine with multi-accumulator. *Renew. Energy* **2021**, *168*, 1177–1188. [CrossRef]
31. Wang, H.P.; Mustafa, G.I.Y.; Tian, Y. Model-free fractional-order sliding mode control for an active vehicle suspension system. *Adv. Eng. Softw.* **2018**, *115*, 452–461. [CrossRef]
32. Chen, K.; Zhou, F.; Liu, A. Chaotic dynamic weight particle swarm optimization for numerical function optimization. *Knowl. Based Syst.* **2018**, *139*, 23–40. [CrossRef]
33. Yao, X.; Liu, Y.; Lin, G. Evolutionary Programming Made Faster. *IEEE Trans. Evol. Comput.* **1999**, *3*, 82–88.
34. Chen, S. Quantum-Behaved Particle Swarm Optimization with Weighted Mean Personal Best Position and Adaptive Local Attractor. *Information* **2019**, *10*, 22. [CrossRef]
35. 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]
36. Coelho, L.S. Novel Gaussian quantum-behaved particle swarm optimiser applied to electromagnetic design. *IET Sci. Meas. Technol.* **2007**, *1*, 290–294. [CrossRef]
37. Sun, J.; Wu, X.; Palade, V.; Fang, W.; Lai, C.-H.; Xu, W. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Inf. Sci.* **2012**, *193*, 81–103. [CrossRef]
38. Tian, N.; Lai, C.-H. Parallel quantum-behaved particle swarm optimization. *Int. J. Mach. Learn. Cybern.* **2013**, *5*, 309–318. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.