Special Collection: Intelligent Algorithms and Optimization with Applications

Journal of
Algorithms &
Computational
Technology

Journal of Algorithms &
Computational Technology
Volume 19: 1–18
© The Author(s) 2025
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/17483026251331506
journals.sagepub.com/home/act
Sage

Improved DBO algorithm tunes fuzzy-PD controller for robot manipulator trajectory tracking

Ma Haohao[1,2], Azizan As'arry[2], Li Chaoqun[1], Mohd Idris Shah Ismail[2], Hafiz Rashidi Ramli[3], Aidin Delgoshaei[2] and M.Y.M. Zuhri[4,5]

Abstract

This article proposes a novel approach for trajectory tracking of a six degrees-of-freedom (6-DOF) collaborative robot manipulator using an adaptive fuzzy proportional derivative (PD) controller. Based on the dynamic modeling of the robot manipulator, the PD control law is designed, and the improved dung beetle optimization (DBO) algorithm is introduced using the good point set (GPS) method for population initialization and the sine strategy for convergence factor adjustment. Furthermore, a fuzzy adaptive strategy is developed to adjust the PD controller gain based on real-time errors. This article uses discrete Lyapunov iterative stability to analyze the global asymptotic stability of the robot closed-loop system. The experimental results verify that the DBO-fuzzy-PD controller is superior to the original PD controller. The ISE value is reduced from 3.4140 to 0.0384, and the IAE value is reduced from 1.9876 to 0.1843. The DBO-fuzzy-PD controller has better tracking accuracy and response speed than traditional PD. Experimental results show that the proposed DBO-fuzzy-PD controller significantly enhances the trajectory tracking performance of the 6-DOF collaborative robot manipulator.

Keywords

Trajectory tracking, PD controller, adaptive control, fuzzy control, inverse dynamics

Received: 5 June 2024; revised: 24 October 2024; accepted: 9 January 2025

Introduction

Collaborative robots are widely used in various fields such as manufacturing, healthcare, and services due to their flexibility and safety capabilities. The six degrees-of-freedom (6-DOF) collaborative robot manipulator has received special attention among them. However, due to the nonlinearity, coupling, and uncertainty of the dynamics of 6-DOF collaborative robots, achieving accurate and stable trajectory tracking is still an important research direction.[1]

Traditional PD (proportional derivative) controllers are commonly used in robot control systems due to their simplicity and ease of implementation. Zhen et al.[2] studied the dynamic modeling of permanent magnet synchronous motors (PMSM) and harmonic reducers, and proposed a robust control method based on PD, which has good dynamic performance for single-joint control of robots. Chaudhary et al.[3] designed a PD controller to achieve hybrid force/

[1]Faculty of Mechatronics and Automotive Engineering, Tianshui Normal University, Tianshui, China
[2]Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia
[3]Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia
[4]Advanced Engineering Materials and Composites Research Centre (AEMC), Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, Serdang, Malaysia
[5]Laboratory of Biocomposite Technology, Institute of Tropical Forestry and Forest Product (INTROP), University Putra Malaysia, Serdang, Malaysia

Corresponding author:
Azizan As'arry, Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia.
Email: zizan@upm.edu.my

position control, which proved the superiority of the fuzzy-PD controller, but lacked the consideration of external disturbance factors. Ulici et al.[4] designed a sliding mode position control manipulator, which estimates joints without torque sensors in real time and achieves position control tracking. However, the performance of traditional PD controllers is often limited in complex and dynamic environments because they rely on fixed gain values that cannot adapt to changing conditions.[5] This limitation has prompted researchers to explore advanced control strategies that can provide better accuracy and robustness.

With the integration of intelligent optimization algorithms and fuzzy control technology,[6] the performance of robot controllers is gradually improved. More and more researchers are using optimization algorithms such as particle swarm optimization (PSO),[7] and genetic algorithm (GA)[8] to fine-tune controller parameters and improve control accuracy. The dung beetle optimization (DBO) algorithm[9] inspired by the foraging behavior of dung beetles, is a relatively new optimization technique that has demonstrated efficiency in solving complex optimization problems. At the same time, fuzzy controllers are recognized for their ability to handle nonlinearity and uncertainty, making them a valuable complement to traditional control methods.

The oscillation of the robot manipulator during movement requires additional energy consumption, which also affects the working stability of the mechanical body. The expected trajectory of the robot manipulator is the position and posture corresponding to a fixed time to minimize the energy consumed during the entire movement process. Recent researchers have explored related technologies for robot control. Moyrón et al.[10] explored nonlinear PID controllers to effectively regulate flexible joint robot systems. Moreno-Valenzuela et al. further improved the performance by compensating for actuator saturation and integrated advanced antisaturation techniques into the controller to improve the trajectory tracking of robotic manipulators with actuator limitations.[11,12] In this study, the end trajectory of the robot manipulator is expected to be smooth and continuous, and the expected input of each joint is third-order differentiable. The intelligent optimization algorithm can optimize fixed-point motion and design the optimal trajectory parameters of joint angle, velocity, acceleration, and jerk.[13]

Iterative learning control of robot manipulators is relatively simple. Based on the PD control method and fuzzy adaptation, the robot manipulator can track the desired trajectory of the input with high precision without relying on an accurate mathematical model. Therefore, it is possible to control uncertain nonlinear strongly coupled dynamic systems in an environment with external disturbances and deviations in dynamic parameters.

This study focuses on developing an adaptive fuzzy PD controller for trajectory tracking of a 6-DOF collaborative robot manipulator. By using an improved DBO algorithm to optimize the controller parameters, and using a fuzzy adaptive mechanism to adjust the PD gain based on real-time errors, the purpose is to achieve high-precision trajectory tracking under different conditions. This article presents several key contributions to the field of robotic control systems:

1. The DBO algorithm is improved to optimize controller parameters more effectively.
2. An adaptive iterative learning fuzzy PD control method is designed based on PD control to calculate the torque of the robot manipulator.
3. The improved DBO algorithm is combined with the fuzzy PD controller to construct a robot control system with high-precision trajectory tracking capability.

The content of this article includes: the second section introduces the method in detail, including the development of the robot dynamics model, inverse dynamics solution, PD control law, DBO algorithm and its improvement, design of fuzzy adaptive mechanism, the stability proof, and fuzzy adaptive PD optimized by DBO controller implementation. The third section introduces the experimental setup and simulation results, including a comparative analysis of robot model parameters, trajectory smoothness, and tracking performance using different control strategies. Finally, the fourth section concludes the study and proposes directions for future research.

Research method

This section details the research methodology employed in developing the adaptive fuzzy PD controller for trajectory tracking of a 6-DOF collaborative robot manipulator. The methodology encompasses several key components: the establishment of the robot's dynamic model, the solution to the inverse dynamics in joint space, the design and implementation of the PD control law, the development and enhancement of the DBO algorithm, the creation of the fuzzy adaptive mechanism, and the integration of these elements into the final DBO-optimized fuzzy adaptive PD controller. Each component is crucial for achieving precise and robust trajectory tracking under varying conditions. The subsequent sections provide a detailed explanation of each step in the methodology.

Robot dynamic model

In order to achieve precise control of a 6-DOF collaborative robot manipulator, it is crucial to develop an accurate dynamics model. The dynamics of a robot describe the relationship between joint torques and the resulting motion, taking into account the effects of inertia, Coriolis forces, centrifugal forces, and gravity. The dynamic model of the robot in this study is derived using the Lagrange formula, which provides a systematic method to derive the motion equations of complex mechanical systems, as shown in equation (1).[14]

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{d} = \boldsymbol{\tau} \tag{1}$$

where $\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, and $\ddot{\boldsymbol{q}}$ denote the joint angle vector, joint angular velocity vector, and joint angular acceleration vector, respectively. $\boldsymbol{M}(\boldsymbol{q})$ is the inertia matrix, $\boldsymbol{M}(\boldsymbol{q})\epsilon\boldsymbol{R}_{6\times6}$. $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the centrifugal force and Coriolis force matrix, $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\epsilon\boldsymbol{R}_{6\times6}$. $\boldsymbol{G}(\boldsymbol{q})$ is the gravity matrix, $\boldsymbol{G}(\boldsymbol{q})\epsilon\boldsymbol{R}_{6\times1}$. $\boldsymbol{d}$ is the time-varying bounded external disturbance of the manipulator system, $\boldsymbol{d}\epsilon\boldsymbol{R}_{6\times1}$. $\boldsymbol{\tau}$ is the joint control torque, $\boldsymbol{\tau}\epsilon\boldsymbol{R}_{6\times1}$.

It is difficult to establish an accurate dynamic model of a robot manipulator through identification of robot dynamic parameters. Therefore, this article constructs a nominal model of the robot manipulator dynamic equations based on the theoretical dynamic model of the robot. As shown in equation (2), the inertia matrix, centrifugal force and Coriolis force matrices, and gravity matrix are equal to the sum of the nominal model terms and uncertainty terms.

$$\begin{cases} \boldsymbol{M}(\boldsymbol{q}) = \boldsymbol{M}_0(\boldsymbol{q}) + \Delta\boldsymbol{M}(\boldsymbol{q}) \\ \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \boldsymbol{C}_0(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \Delta\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) \\ \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{G}_0(\boldsymbol{q}) + \Delta\boldsymbol{G}(\boldsymbol{q}) \end{cases} \tag{2}$$

Among them, $\boldsymbol{M}_0(\boldsymbol{q})$, $\boldsymbol{C}_0(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\boldsymbol{G}_0(\boldsymbol{q})$ are the nominal inertia matrix, nominal centrifugal force and Coriolis force matrix in the model, and the nominal gravity matrix. $\Delta\boldsymbol{M}(\boldsymbol{q})$, $\Delta\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$, and $\Delta\boldsymbol{G}(\boldsymbol{q})$ are the uncertainty term of the dynamic model. Equation 1 is reordered as equation (3).

$$\boldsymbol{M}_0(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}_0(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{G}_0(\boldsymbol{q}) + \boldsymbol{\rho} = \boldsymbol{\tau} \tag{3}$$

where $\boldsymbol{\rho} = \Delta\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \Delta\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \Delta\boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{d}$. $\boldsymbol{\rho}$ is the total modeling error and the total uncertainty term of the dynamic model. The control law is designed as equation (4).

$$\begin{aligned} \boldsymbol{M}_0(\boldsymbol{q_d})\big(\ddot{\boldsymbol{q}}_{\boldsymbol{d}} + K_v(\dot{\boldsymbol{q}}_{\boldsymbol{d}} - \dot{\boldsymbol{q}}) + K_p(\boldsymbol{q_d} - \boldsymbol{q})\big) \\ + \boldsymbol{C}_0\big(\boldsymbol{q_d}, \dot{\boldsymbol{q}}_{\boldsymbol{d}}\big)\dot{\boldsymbol{q}}_{\boldsymbol{d}} + \boldsymbol{G}_0\big(\boldsymbol{q_d}\big) + \boldsymbol{\rho} = \boldsymbol{\tau} \end{aligned} \tag{4}$$

$\boldsymbol{q_d}$, $\dot{\boldsymbol{q}}_{\boldsymbol{d}}$, and $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}$ denote the desired joint angle vector, joint angular velocity vector, and joint angular acceleration vector, respectively. The error of the closed-loop system under ideal conditions can be expressed as equation (5).

$$\ddot{e} + K_v\dot{e} + K_p e = 0 \tag{5}$$

where $e = q - q_d$, $\dot{e} = \dot{q} - \dot{q}_d$, and $\ddot{e} = \ddot{q} - \ddot{q}_d$.

The control law (4) designed above can be divided into two parts, the inverse dynamics of the robot and PD feedback control. Under this control, joint errors are uncoupled, so the dynamic characteristics of the robot are independent of the position of the robot manipulator. However, there are inevitable system errors in the realistic control model, which will lead to the coupling of the axes of the robot manipulator joints and the dynamic errors will not converge to zero. By choosing the appropriate K_d and K_p, the error will attenuate to zero, thereby ensuring the stability and dynamic characteristics of the system.

Based on PD control law, dynamic calculation is added to design a double closed-loop feedback control method. Design the inner loop control law as equation (6).

$$\ddot{q}_{PD} = K_{p1}\left[e'(t) + K_{d1}\frac{de'(t)}{dt}\right] \tag{6}$$

where $e'(t) = \dot{q}_{d'} - \dot{q}$. The designed outer loop control law is as shown in equation (7).

$$\dot{q}_{d'} = K_{p2}\left[e(t) + K_{d2}\frac{de(t)}{dt}\right] \tag{7}$$

where $e(t) = \dot{q}_d - \dot{q}$. The PD control law calculated by dynamics is as shown in equation (8).

$$\ddot{q}_{d'} = \ddot{q}_d + \ddot{q}_{PD} + \dot{q}_{d'} \tag{8}$$

From the above, the value of the calculated moment can be obtained according to equation (9).

$$\tau = \boldsymbol{M}(q_d)\ddot{q}_{d'} + \boldsymbol{C}(q_d, \dot{q}_d)\dot{q}_d + \boldsymbol{G}(q_d) + \boldsymbol{\rho} \tag{9}$$

Figure 1 shows the control system's structural block diagram.

Adaptive fuzzy control strategy

Fuzzy logic control (FLC) is a method that mimics human reasoning and decision making using fuzzy sets and linguistic rules.[15] Unlike conventional control methods that rely on precise mathematical models, FLC can handle uncertainties and imprecise information, making it well-suited for complex and nonlinear robotic manipulator systems. This robot manipulator uses fuzzification, a product inference engine, and center-of-gravity average defuzzification. For fuzzy rules: ***IF*** *x_1 is F_1^j* ***and*** *...****and*** *x_n is F_n^j* ***Then*** *y is B^j (j = 1, 2, ..., N)*. The output of the fuzzy system is equation (10).

$$y(x) = \frac{\sum_{j=1}^{N}\theta_j\prod_{i=1}^{n}\mu_i^j(x_i)}{\sum_{j=1}^{N}\left[\prod_{i=1}^{n}\mu_i^j(x_i)\right]} \tag{10}$$

where $x = [x_1, x_2, \ldots, x_n]^T\epsilon R^n$, $\mu_i^j(x_i)$ is the membership function, which is the Gaussian function used in robot control applications. $\theta_j = \max_{y\epsilon R} B^j(y)$.

In this study, a fuzzy inference system (FIS) is used to enhance the traditional PD controller to dynamically adjust proportional gains (K_p) and differential gains (K_d) based on real-time error signals. This adaptive mechanism allows the controller to respond more effectively to varying conditions and uncertainties in the robot's dynamics.[16]

The rule base of FIS defines the relationship between input fuzzy sets and output control actions. In robot

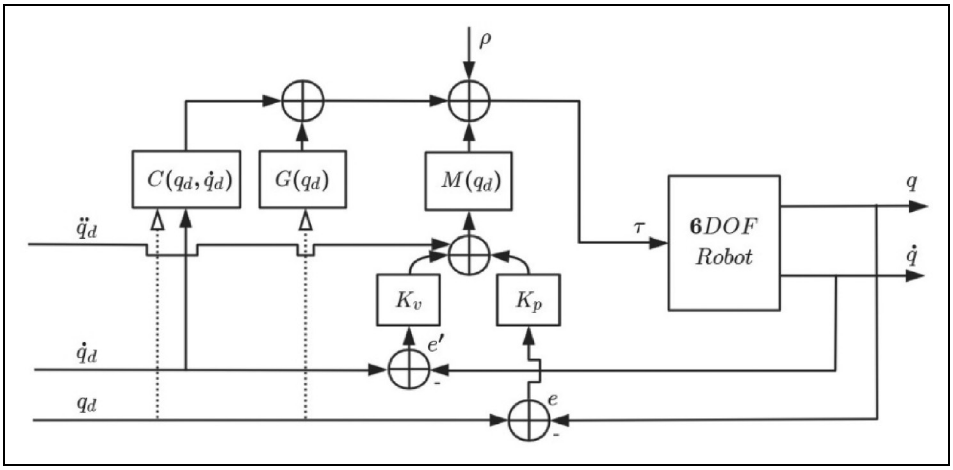


Figure 1. Diagram of the calculated torque proportional derivative control system.

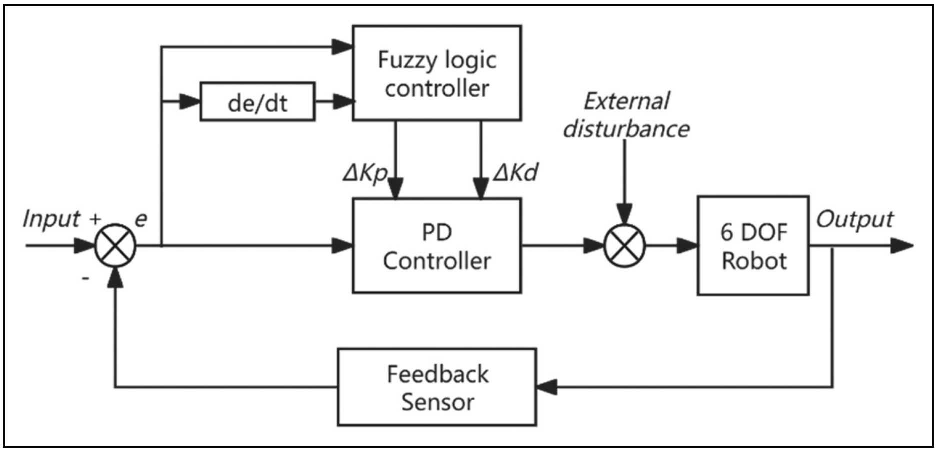


Figure 2. Fuzzy proportional derivative control system.

manipulator control, the input variables are the joint tracking angle error (*e*) and the derivative of the error (*de*), and the Gaussian function is defined as a membership function within the standardized range of the variable. The membership function of the error includes negative large (NL), negative small (NS), zero (ZO), positive small (PS), and positive large (PL).[17]

Figure 2 clearly shows the integration of fuzzy logic with the traditional PD controller, highlighting the adaptive nature of the system. By dynamically adjusting the PD gains based on real-time error information, the fuzzy PD control system offers improved performance in trajectory tracking for complex robotic systems.

In evaluating the performance of robot manipulator control systems, several integral error metrics as shown in equation (11) are commonly used to quantify the accuracy and efficiency of the system in tracking a desired trajectory. Four equations include integral square error (ISE), integral absolute error (IAE), integral time square error (ITSE), and integral time absolute error (ITAE). The IAE measures the total absolute error over time, providing a straightforward assessment of overall performance.[18]

$$\begin{cases} ISE = \int e(t)^2 dt \\ IAE = \int |e(t)| dt \\ ITSE = \int t e(t)^2 dt \\ ITAE = \int t |e(t)| dt \end{cases} \tag{11}$$

Improved DBO algorithm

The DBO algorithm is a nature-inspired optimization technique that mimics the foraging behavior of dung beetles. This algorithm effectively solves complex optimization problems due to its ability to balance exploration and exploitation in the search space.[19] The original DBO algorithm has been shown to perform well in various applications. Still, improvements can further enhance its efficiency and convergence speed, making it more suitable for a 6-DOF collaborative robot manipulator real-time control.

Compared to the random population initialization of the original DBO algorithm, the good point set (GPS) method was used in this study. The GPS method aims to distribute the initial population more evenly in the search space, improve the algorithm's ability to explore different regions and avoid premature convergence to local optima.[20] In n-dimensional problems, set the population size to m, and use equation (12) to calculate the value of r.

$$r_j = mod\left(2\cos\left(\frac{2\pi j}{7}\right)m_i,\, 1\right) \tag{12}$$

where $1 \le j \le n$, m_i denotes the i-th population individual, $1 \le i \le m$. The GPS X_i^j between the upper limit and the lower limit is constructed according to equation (13).

$$X_i^j = a_j + r_j(b_j - a_j) \tag{13}$$

where b_j denotes the upper limit and a_j denotes the lower limit. The comparison between the population initialization obtained by the GPS method and the randomly generated population initialization is shown in Figure 2. The initialization results of 100 populations in a two-dimensional space indicate that the GPS method provides superior performance and is more evenly and systematically distributed throughout the search space (Figure 3).

In the original DBO algorithm, a linear convergence factor is employed, which reduces from 1 to 0 for iterations, as equation (14). The improved DBO algorithm adopts a sinusoidal convergence strategy, as equation (15). This approach ensures a smooth and adaptive convergence factor.[21] The improved algorithm achieves a balanced convergence speed, enhancing exploration at the start and ensuring efficient exploitation toward the end.

$$R = 1 - \frac{t}{T} \tag{14}$$

$$R = \frac{1 + \sin\left(\frac{\pi}{2} + \pi\left(\frac{t}{T}\right)^k\right)}{2} \tag{15}$$

where R denotes the convergence factor, t is the current number of iterations, and T the set maximum number of iterations. Different values of parameter k can obtain different convergence factor change curves. Figure 4 shows the convergence factor for 50 iterations. In this study, $k = 1.2$ was selected.

To demonstrate the superiority of the I-DBO algorithm over the original DBO and other optimization algorithms, a comparative analysis was conducted using a practical engineering problem. The objective was to evaluate the performance of each algorithm in terms of convergence speed, and solution accuracy.

Problem 1: Three-bar truss design. $l = 100$ cm, $P = 2$ kN / cm^3, $\sigma = 2$ kN / cm^3, variable range: $0 \le x_1, x_2 \le 1$. Minimize: $f(X) = \left(2\sqrt{2}x_1 + x_2\right) \times l$. Subject to:

$$\begin{cases} g_1(X) = \dfrac{2\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0 \\ g_2(X) = \dfrac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \le 0 \\ g_3(X) = \dfrac{2\sqrt{2}x_1 + x_2}{\sqrt{2}x_2 + x_1}P - \sigma \le 0 \end{cases}$$

Problem 2: Design of I-shaped beam. Variable range: $10 \le x_1 \le 50$, $10 \le x_2 \le 80$, $0.9 \le x_3 \le 5$, $0.9 \le x_4 \le 5$. Maximize: $f(X) = \frac{5000}{x_3(x_2-2x_4)^3/12+(x_1x_4^3/6)+2bx_4(x_2-x_4/2)^2}$. Subject to:

$$\begin{cases} g_1(X) = 2x_1x_3 + x_3(x_2 - 2x_4) \le 300 \\ g_2(X) = \dfrac{18x_2 \times 10^{14}}{x_3(x_2 - 2x_4)^3 + 2x_1x_3(4x_4^2 + 3x_2(x_2 - 2x_4))} + \dfrac{15x_1 \times 10^3}{(x_2 - 2x_4)x_3^2 + 2x_3x_1^3} \le 56 \end{cases}$$

Problem 3: Speed reducer design. Variable range: $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8, x_3 \in \{17, 18, 19, \ldots, 28\}$, $7.3 \le x_4$, $x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5 \le x_7 \le 5.5$. Minimize: $f(X) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2)$. Subject to:

$$\begin{cases} g_1(X) = \dfrac{27}{x_1x_2^2x_3} - 1 \le 0 \\ g_2(X) = \dfrac{397.5}{x_1x_2^2x_3^2} - 1 \le 0 \\ g_3(X) = \dfrac{1.93x_4^3}{x_2x_6^4x_3} - 1 \le 0 \\ g_4(X) = \dfrac{1.93x_5^3}{x_2x_7^4x_3} - 1 \le 0 \\ g_5(X) = \dfrac{\sqrt{(745x_4 / x_2x_3)^2 + 16.9 \times 10^6}}{110x_6^3} - 1 \le 0 \\ g_6(X) = \dfrac{\sqrt{(745x_5 / x_2x_3)^2 + 157.5 \times 10^6}}{85x_7^3} - 1 \le 0 \\ g_7(X) = \dfrac{x_2x_3}{40} - 1 \le 0 \\ g_8(X) = \dfrac{5x_2}{x_1} - 1 \le 0 \\ g_9(X) = \dfrac{x_1}{12x_2} - 1 \le 0 \\ g_{10}(X) = \dfrac{1.5x_6 + 1.9}{x_4} - 1 \le 0 \\ g_{11}(X) = \dfrac{1.1x_7 + 1.9}{x_5} - 1 \le 0 \end{cases}$$

Problem 4: Design of pressure vessel. Variable range: $x_1, x_2 \in \{1 \times 0.0625,\ 2 \times 0.0625,\ \ldots, 1600 \times 0.0625\}$, $10 \le x_3$, $x_4 \le 200$. Minimize: $f(X) = 0.6224x_1x_3x_4$

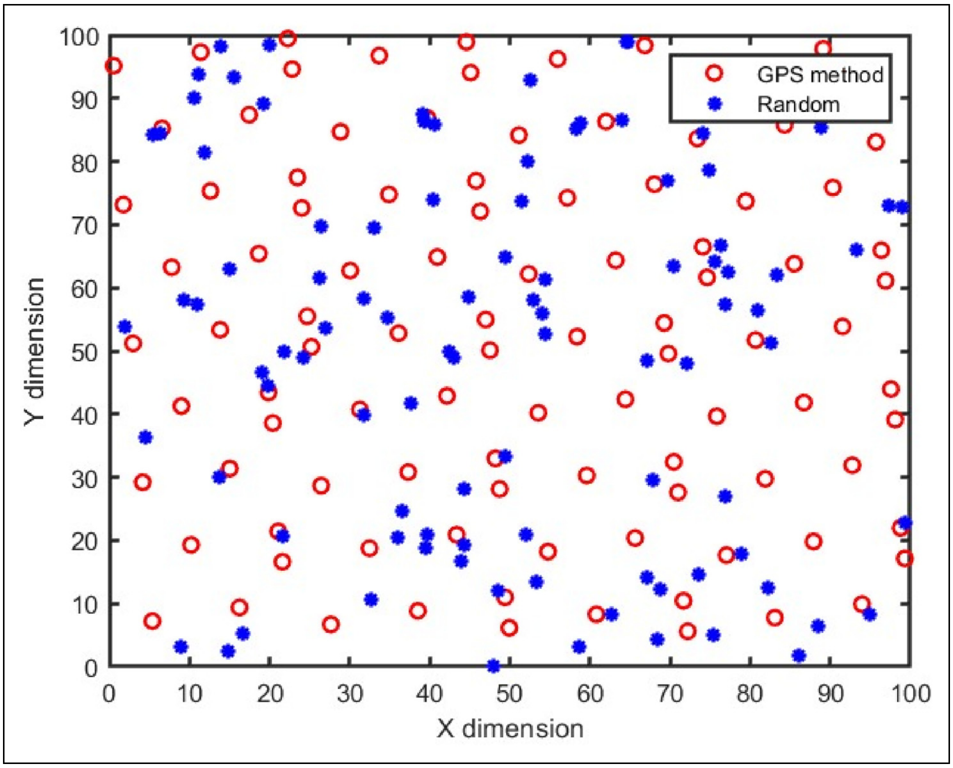


Figure 3. Population initialization of good point set method.

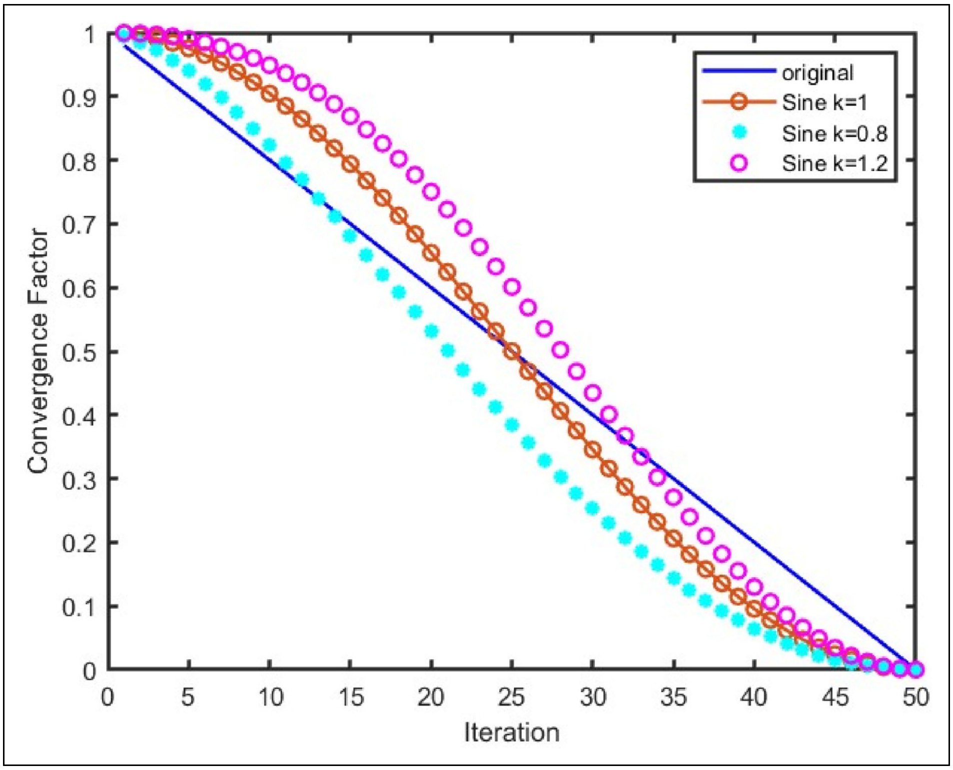


Figure 4. Curve of convergence factor.

$+ 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$. Subject to:

$$\begin{cases} g_1(X) = -x_1 + 0.0193x_3 \le 0 \\ g_2(X) = -x_2 + 0.00954x_3 \le 0 \\ g_3(X) = -\pi x_3^2 x_4 - \dfrac{4}{3}\pi x_3^3 + 1,296,000 \le 0 \\ g_4(X) = x_4 - 240 \le 0 \end{cases}$$

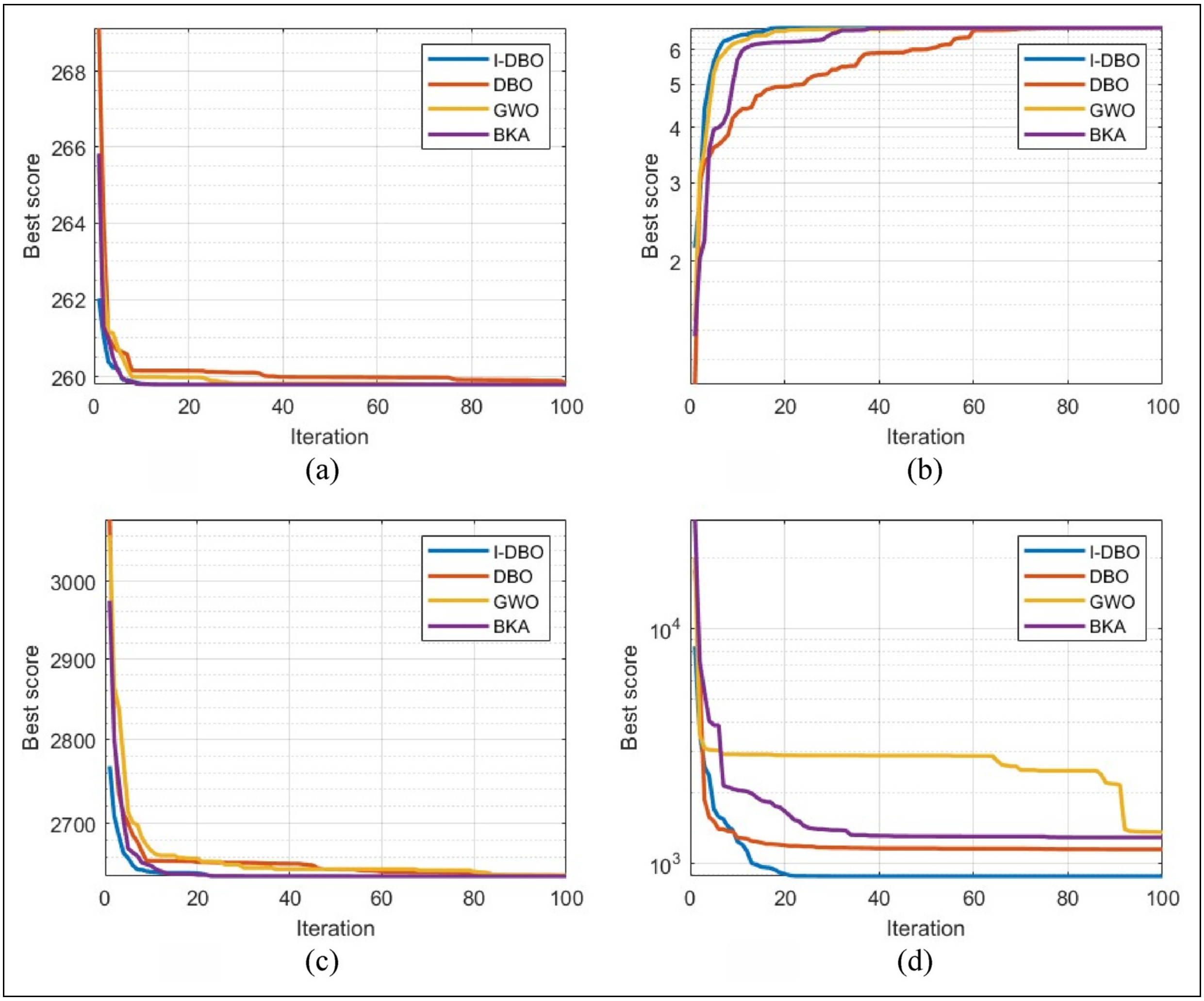
To rigorously verify the effectiveness of the I-DBO algorithm, the performance of I-DBO was compared with the original DBO algorithm, the Grey Wolf Optimization (GWO), and the Black-winged Kite Algorithm (BKA). Each algorithm solved the same engineering problem five times and the statistics were calculated. The convergence curves of the four engineering problems are shown in Figure 5.

The evaluation was based on several statistical indicators: worst value, best value, standard deviation (std), mean, median, and Friedman test value to determine the statistical significance of the observed differences. The calculation results are shown in Table 1.

The I-DBO algorithm consistently achieved the best values across all engineering problems compared to other algorithms. The lower standard deviation for the I-DBO algorithm demonstrates its stability and consistency in performance. The Friedman test was employed to validate the differences between the algorithms statistically. The Friedman test value is less than 0.05, which indicates that the improved DBO algorithm has a significant difference. These results substantiate the improved DBO algorithm's reliability and efficiency, making it a highly effective tool for solving complex engineering optimization problems.

DBO-fuzzy-PD controller

The I-DBO algorithm is implemented to optimize the parameters of the PD controller.[22] Generate an initial



(a)

(b)

(c)

(d)

Figure 5. The convergence curves of four problems: (a) three-bar truss design; (b) design of I-shaped beam; (c) speed reducer design; and (d) design of pressure vessel.

Table 1. Comparative results of improved DBO with other methods.

Problem 1	I-DBO	DBO	GWO	BKA
Worst	259.8050467	259.8071967	259.8050565	259.8050467
Best	259.8050467	259.9108546	259.8086197	259.8050467
Std	4.47586E-13	0.041774472	0.001471173	4.01944E-14
Mean	259.8050467	259.8371628	259.8060177	259.8050467
Median	259.8050467	259.8235448	259.805538	259.8050467
Friedman test	6.2985e-37 < 0.05			
Problem 2	I-DBO	DBO	GWO	BKA
Worst	6.703047608	6.700105724	6.700936899	6.703047602
Best	6.703047608	6.703047608	6.702891174	6.703047608
Std	4.44089E−16	0.00131533	0.000787226	2.42297E-09
Mean	6.703047608	6.702458657	6.702162232	6.703047607
Median	6.703047608	6.703047608	6.702216852	6.703047608
Friedman test	1.28e-36 < 0.05			
Problem 3	I-DBO	DBO	GWO	BKA
Worst	2638.819832	2638.819832	2639.0742	2638.819832
Best	2638.819832	2640.610707	2641.958472	2638.819833
Std	3.93823E-13	0.856640719	1.182644121	4.32127E-07
Mean	2638.819832	2639.51301	2639.866844	2638.819832
Median	2638.819832	2639.030648	2639.475847	2638.819832
Friedman test	1.5391e-35 < 0.05			
Problem 4	I-DBO	DBO	GWO	BKA
Worst	753.5014127	753.5189726	767.9033688	753.501422
Best	1392.749638	1429.433705	1582.270415	1513.65343
Std	285.8804973	359.2238385	335.8835073	303.9407118
Mean	881.3510578	1146.79118	1355.653077	1289.080758
Median	753.5014127	1393.171517	1475.325489	1392.74964
Friedman test	4.8618e-37 < 0.05			

BKA: Black-winged Kite Algorithm; I-DBO: improved dung beetle optimization; GWO: Grey Wolf Optimization.

population using the GPS method. Each individual in the population represents a potential set of PD controller parameters. Evaluate the minimizing of the integral error metrics based on a predefined objective function. Apply the sinusoidal convergence factor to update the positions of the individuals in the search space. Repeat the fitness evaluation and update steps for a specified number of iterations or until convergence criteria are met.[23] Finally, output the best set of PD controller parameters. Figure 6 illustrates the DBO-fuzzy-PD controller. The fuzzy logic controller can output the gain values of K_p and K_d in real time according to the trajectory tracking error of the robot manipulator, and adaptively adjust the optimal K_p and K_d values obtained by the DBO algorithm.

Iterative learning control. The adaptive fuzzy-PD controller, driven by the improved DBO algorithm, iteratively learns and responds efficiently to real-time external disturbances and errors. For the robot dynamics system (equation (1)), two working conditions are required: the expected trajectory $\boldsymbol{q}_d(t)$ is third-order differentiable within the working time $t \in [0, t_f]$; The iteration process satisfies the initial conditions, $\boldsymbol{q}_d(0) - \boldsymbol{q}^j(0) = 0$, $\dot{\boldsymbol{q}}_d(0) - \dot{\boldsymbol{q}}^j(0) = 0, j$ is the number of iterations, $j \in N$. The dynamic equation of the robot system with nonrepetitive uncertain disturbance at the j-th iteration is equation (16).

$$\begin{aligned}\boldsymbol{M}_0\big(\boldsymbol{q}^j(t)\big)\ddot{\boldsymbol{q}}^j(t) + \boldsymbol{C}_0\big(\boldsymbol{q}^j(t), \dot{\boldsymbol{q}}^j(t)\big)\dot{\boldsymbol{q}}^j(t) \\ + \boldsymbol{G}_0\big(\boldsymbol{q}^j(t)\big) + \boldsymbol{\rho}^j(t) = \boldsymbol{\tau}^j(t)\end{aligned} \tag{16}$$

where $\boldsymbol{\rho}^j(t) = \Delta\boldsymbol{M}(\boldsymbol{q}^j(t))\ddot{\boldsymbol{q}}^j(t) + \Delta\boldsymbol{C}(\boldsymbol{q}^j(t), \dot{\boldsymbol{q}}^j(t))\dot{\boldsymbol{q}}^j(t) + \Delta\boldsymbol{G}(\boldsymbol{q}^j(t)) + \boldsymbol{d}^j(\boldsymbol{t})$. In this system, the control law is designed as equation (17).

$$\boldsymbol{\tau}^j(t) = K_p^j\boldsymbol{e}^j(t) + K_d^j\dot{\boldsymbol{e}}^j(t) + \boldsymbol{\tau}^{j-1}(t) \tag{17}$$

Along the input trajectory $(\boldsymbol{q}_d(t), \dot{\boldsymbol{q}}_d(t), \ddot{\boldsymbol{q}}_d(t))$, the Taylor formula can be linearized into equation (18).

$$\begin{aligned}\boldsymbol{M}_0(t)\ddot{\boldsymbol{e}} + [\boldsymbol{C}_0 + \boldsymbol{C}_1]\dot{\boldsymbol{e}} + \boldsymbol{F}\boldsymbol{e} + \boldsymbol{n}(\ddot{\boldsymbol{e}}, \dot{\boldsymbol{e}}, \boldsymbol{e}, \boldsymbol{t}) \\ = \boldsymbol{H} - (\boldsymbol{M}_0\ddot{\boldsymbol{q}} + \boldsymbol{C}_0\dot{\boldsymbol{q}} + \boldsymbol{G})\end{aligned} \tag{18}$$

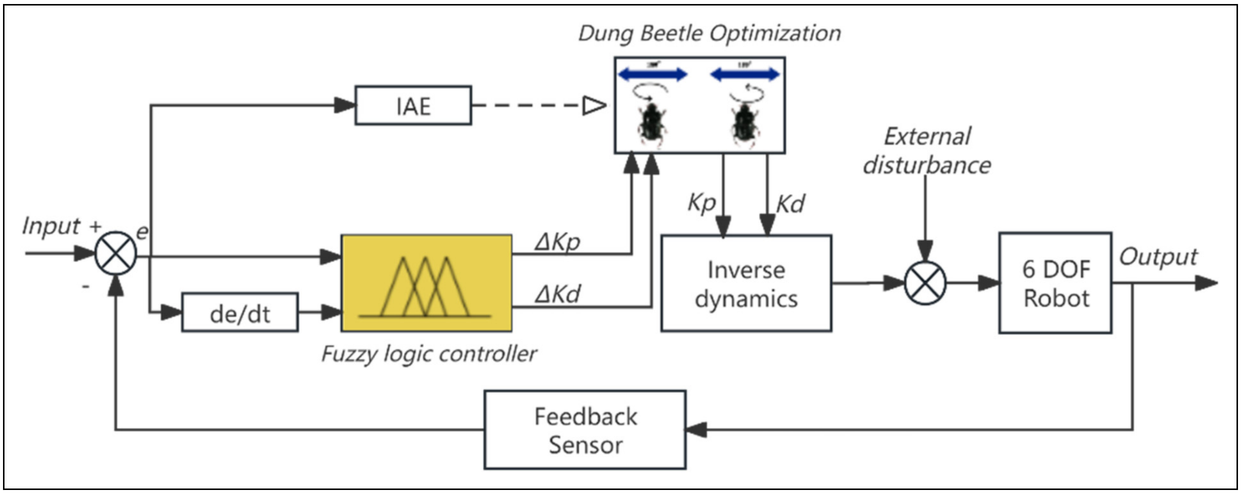


Figure 6. Dung beetle optimization-fuzzy-proportional derivative controller system.

where

$$\boldsymbol{M}_0(t) = \boldsymbol{M}_0(\boldsymbol{q}_d(t))$$

$$\boldsymbol{C}_0(t) = \boldsymbol{C}_0(\boldsymbol{q}_d(t), \dot{\boldsymbol{q}}_d(t))$$

$$\boldsymbol{C}_1(t) = \frac{\partial \boldsymbol{C}_0}{\partial \dot{\boldsymbol{q}}}|_{\boldsymbol{q}_d(t),\dot{\boldsymbol{q}}_d(t)}\dot{\boldsymbol{q}}_d(t) + \frac{\partial \boldsymbol{G}_0}{\partial \dot{\boldsymbol{q}}}|_{\boldsymbol{q}_d(t),\dot{\boldsymbol{q}}_d(t)}$$

$$\boldsymbol{F}(t) = \frac{\partial \boldsymbol{M}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d(t)}\ddot{\boldsymbol{q}}_d(t) + \frac{\partial \boldsymbol{C}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d(t),\dot{\boldsymbol{q}}_d(t)}\dot{\boldsymbol{q}}_d(t) + \frac{\partial \boldsymbol{G}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d(t)}$$

$$\boldsymbol{n}(\ddot{\boldsymbol{e}}, \dot{\boldsymbol{e}}, \boldsymbol{e}, \boldsymbol{t}) = -\frac{\partial \boldsymbol{M}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d}\ddot{\boldsymbol{e}}\boldsymbol{e} - \frac{\partial \boldsymbol{C}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d}\dot{\boldsymbol{e}}e - \frac{\partial \boldsymbol{C}_0}{\partial \boldsymbol{q}}|_{\boldsymbol{q}_d,\dot{\boldsymbol{q}}_d}\dot{\boldsymbol{e}}\dot{e} + \boldsymbol{O}_{\boldsymbol{M}}(\cdot)\ddot{\boldsymbol{q}} + \boldsymbol{O}_{\boldsymbol{C}}(\cdot)\dot{\boldsymbol{q}} - \boldsymbol{O}_{\boldsymbol{G}}(\cdot)$$

$$\boldsymbol{H}(t) = \boldsymbol{M}_0(\boldsymbol{q}_d(t))\ddot{\boldsymbol{q}}_d(t) + \boldsymbol{C}_0(\boldsymbol{q}_d(t), \dot{\boldsymbol{q}}_d(t))\dot{\boldsymbol{q}}_d(t) + \boldsymbol{G}_0(\boldsymbol{q}_d(t))$$

where $\boldsymbol{O}_{\boldsymbol{M}}(\cdot)$, $\boldsymbol{O}_{\boldsymbol{C}}(\cdot)$, and $\boldsymbol{O}_{\boldsymbol{G}}$ are the residuals of the first-order Taylor expansion of $\boldsymbol{M}_0$, $\boldsymbol{C}_0$, and $\boldsymbol{G}_0$, respectively.

Combine the residual term $\boldsymbol{n}(\ddot{\boldsymbol{e}}^j, \dot{\boldsymbol{e}}^j, \boldsymbol{e}^j, \boldsymbol{t})$ with the external disturbance $\boldsymbol{\rho}^j(t)$ into $\boldsymbol{\rho}_1^j(t)$, $\boldsymbol{\rho}_1^j(t) = -\boldsymbol{n}(\ddot{\boldsymbol{e}}^j, \dot{\boldsymbol{e}}^j, \boldsymbol{e}^j, \boldsymbol{t}) + \boldsymbol{\rho}^j(t)$. Integrating equations (17) and (18) can obtain the j-th iteration and $j + 1$-th iteration equation (19).

$$\begin{cases} \boldsymbol{M}_0(t)\ddot{\boldsymbol{e}}^j(t) + [\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^j(t) + F(t)\boldsymbol{e}^j(t) - \boldsymbol{\rho}_1^j(t) = H(t) - \boldsymbol{\tau}^j(t) \\ \boldsymbol{M}_0(t)\ddot{\boldsymbol{e}}^{j+1}(t) + [\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)]\dot{\boldsymbol{e}}^{j+1}(t) + F(t)\boldsymbol{e}^{j+1}(t) - \boldsymbol{\rho}_1^{j+1}(t) = H(t) - \boldsymbol{\tau}^{j+1}(t) \end{cases} \tag{19}$$

Stability proof. The stability proof of the Lyapunov function in the iterative control method proposed in this study cannot be performed by directly verifying that its time derivative is non-positive, because the system operates in the discrete time domain rather than the continuous time domain. For the discrete-time control law equation (17) proposed in this article, the Lyapunov function is defined as equation (20), and the corresponding analysis is performed within the framework of discrete-time stability theory.

$$V^j = \int_0^t e^{-\mu\tau}\boldsymbol{y}^{jT}\boldsymbol{K}_d^0\boldsymbol{y}^j d\tau \geq 0 \tag{20}$$

The switching rule between K_p and K_d for the PD controller gain parameters is defined as: $\boldsymbol{K}_p^j = \beta(j)\boldsymbol{K}_p^0$, $\boldsymbol{K}_d^j = \beta(j)\boldsymbol{K}_d^0$, where $\beta(j + 1) > \beta(j)$, $j = 1, 2, 3, \cdots, N$, μ is a positive real number. $\boldsymbol{K}_d^0$ is the initial gain value of K_d in the controller. In the fuzzy-PD controller proposed in this article, the initial value $\boldsymbol{K}_d^0$ is obtained by the improved DBO algorithm. Define the parameter $\boldsymbol{\Lambda}$ to obtain another value of the controller $\boldsymbol{K}_p^j = \boldsymbol{\Lambda}\boldsymbol{K}_d^j$. Based on this, continue to define $\boldsymbol{y}^j(t) = \dot{\boldsymbol{e}}^j(t) + \boldsymbol{\Lambda}\boldsymbol{e}^j(t)$, so $\delta\boldsymbol{y}^j = \boldsymbol{y}^{j+1} - \boldsymbol{y}^j = \delta\dot{\boldsymbol{e}}^j + \boldsymbol{\Lambda}\delta\boldsymbol{e}^j$. To prove that the system will gradually become stable with each iteration, the difference ΔV^j is defined for the Lyapunov function, and its calculation is as shown in equation (21).

$$\begin{aligned} \Delta V^j &= V^{j+1} - V^j \\ &= \int_0^t e^{-\mu\tau}(\delta\boldsymbol{y}^{jT} + \boldsymbol{y}^j)^T\boldsymbol{K}_d^0(\delta\boldsymbol{y}^{jT} + \boldsymbol{y}^j)d\tau - \int_0^t e^{-\mu\tau}\boldsymbol{y}^{jT}\boldsymbol{K}_d^0\boldsymbol{y}^j d\tau \\ &= \int_0^t e^{-\mu\tau}(\delta\boldsymbol{y}^{jT}\boldsymbol{K}_d^0\delta\boldsymbol{y}^j + 2\delta\boldsymbol{y}^{jT}\boldsymbol{K}_d^0\boldsymbol{y}^j)d\tau \end{aligned} \tag{21}$$

The iterative control formula can be further derived to obtain the equations (22) to (25).

$$\begin{aligned} \boldsymbol{M}(t)(\ddot{\boldsymbol{e}}^{j+1}(t) - \ddot{\boldsymbol{e}}^j(t)) &= -[\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)](\dot{\boldsymbol{e}}^{j+1}(t) - \dot{\boldsymbol{e}}^j(t)) \\ &\quad - \boldsymbol{F}(t)(\boldsymbol{e}^{j+1}(t) - \boldsymbol{e}^j(t)) - (\boldsymbol{\tau}^{j+1}(t) - \boldsymbol{\tau}^j(t)) \end{aligned} \tag{22}$$

$$\begin{aligned}
\boldsymbol{M}\delta\dot{\boldsymbol{y}}^j =& \boldsymbol{M}\delta\ddot{e}^{j+1} + \boldsymbol{M\Lambda}\delta\dot{\boldsymbol{e}}^j = \boldsymbol{M}(\ddot{e}^{j+1} - \ddot{e}^j) + \boldsymbol{M\Lambda}(\dot{\boldsymbol{e}}^{j+1} - \dot{\boldsymbol{e}}^j) \\
&= -[\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)](\dot{\boldsymbol{e}}^{j+1}(t) - \dot{\boldsymbol{e}}^j(t)) - \boldsymbol{F}(t)(\boldsymbol{e}^{j+1}(t) - \boldsymbol{e}^j(t)) - (\boldsymbol{\tau}^{j+1}(t) - \boldsymbol{\tau}^j(t)) + \boldsymbol{M\Lambda}(\dot{\boldsymbol{e}}^{j+1} - \dot{\boldsymbol{e}}^j) \\
&= -[\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)]\ \delta\dot{\boldsymbol{e}}^j - \boldsymbol{F}(t)\delta\boldsymbol{e}^j + \boldsymbol{M\Lambda}(\dot{\boldsymbol{e}}^{j+1} - \dot{\boldsymbol{e}}^j) - (\boldsymbol{\Lambda K}_d^{j+1}\boldsymbol{e}^{j+1} + \boldsymbol{K}_d^{j+1}\dot{\boldsymbol{e}}^{j+1}) \\
&= -[\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)]\ \delta\dot{\boldsymbol{e}}^j - \boldsymbol{F}(t)\delta\boldsymbol{e}^j + \boldsymbol{M\Lambda}[(\boldsymbol{y}^{j+1} - \boldsymbol{\Lambda e}^{j+1}) - (\boldsymbol{y}^{\boldsymbol{j}} - \boldsymbol{\Lambda e}^j)] - \boldsymbol{K}_d^{j+1}(\boldsymbol{\Lambda e}^{j+1} + \dot{\boldsymbol{e}}^{j+1}) \\
&= -[\boldsymbol{C}_0(t) + \boldsymbol{C}_1(t)]\ \delta\dot{\boldsymbol{e}}^j - \boldsymbol{F}(t)\delta\boldsymbol{e}^j + \boldsymbol{M\Lambda}\delta\boldsymbol{y}^{\boldsymbol{j}} - \boldsymbol{M\Lambda}^2\delta\boldsymbol{e}^j - \boldsymbol{K}_d^{j+1}(\delta\boldsymbol{y}^{\boldsymbol{j}} + \boldsymbol{y}^{\boldsymbol{j}}) \\
&= -(\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M} + \boldsymbol{K}_d^{j+1})\delta\boldsymbol{y}^{\boldsymbol{j}} - (\boldsymbol{F} - \boldsymbol{\Lambda}(\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M}))\delta\boldsymbol{e}^j - \boldsymbol{K}_d^{j+1}\boldsymbol{y}^{\boldsymbol{j}}
\end{aligned} \tag{23}$$

$$\begin{aligned}
&\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\dot{\boldsymbol{y}}^j d\tau = e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\boldsymbol{y}^j|_0^t - \int_0^t (e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M})'\delta\boldsymbol{y}^j d\tau \\
&= e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\boldsymbol{y}^j + \mu\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\boldsymbol{y}^j d\tau - \int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\dot{\boldsymbol{y}}^j d\tau - \int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\dot{\boldsymbol{M}}\delta\boldsymbol{y}^j d\tau
\end{aligned} \tag{24}$$

$$\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{K}_{\boldsymbol{d}}^{\boldsymbol{j}+1}\delta\boldsymbol{y}^j d\tau = \beta(j+1)\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{K}_{\boldsymbol{d}}^0\delta\boldsymbol{y}^j d\tau \geq \int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{K}_{\boldsymbol{d}}^0\delta\boldsymbol{y}^j d\tau \tag{25}$$

Based on the calculation of equations (21) to (25), the value of ΔV^j is estimated as shown in equation (26).

$$\begin{aligned}
\Delta V^j =& \frac{1}{\beta(j+1)}\left\{\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{K}_{\boldsymbol{d}}^{\boldsymbol{j}+1}\delta\boldsymbol{y}^j d\tau - 2\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\dot{\boldsymbol{y}}^j d\tau - 2\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\dot{\boldsymbol{y}}^j d\tau\right. \\
&\left. -2\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}((\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M} + \boldsymbol{K}_d^{j+1}) + (\boldsymbol{F} - \boldsymbol{\Lambda}(\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M}))\delta\boldsymbol{e}^j)d\tau\right\} \\
&\leq \frac{1}{\beta(j+1)}\left\{-e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\boldsymbol{y}^j - \mu\int_0^t e^{-\mu\tau}\delta\boldsymbol{y}^{jT}\boldsymbol{M}\delta\boldsymbol{y}^j d\tau - \boldsymbol{\Lambda}e^{-\mu\tau}\delta\boldsymbol{e}^{jT}l_p\delta\boldsymbol{e}^j\right. \\
&\left. -\mu\boldsymbol{\Lambda}\int_0^t e^{-\mu\tau}\delta\boldsymbol{e}^{jT}l_p\delta\boldsymbol{e}^j d\tau\int_0^t e^{-\mu\tau}\omega d\tau\right\}
\end{aligned} \tag{26}$$

where $l_p = \lambda_{\min}(\boldsymbol{K}_{\boldsymbol{d}}^0 + 2\boldsymbol{C}_1 - 2\boldsymbol{\Lambda M}) > 0, l_r = \lambda_{\min}(\boldsymbol{K}_{\boldsymbol{d}}^0 + 2\boldsymbol{C}_0 + 2\boldsymbol{F}/\boldsymbol{\Lambda} - 2\dot{\boldsymbol{C}}_1/\boldsymbol{\Lambda}) > 0, \lambda_{\min}(\boldsymbol{A})$ denotes the smallest eigenvalue of matrix $\boldsymbol{A}$. Take $\boldsymbol{Q} = \boldsymbol{F}/\boldsymbol{\Lambda} - (\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M})$**,** $||\boldsymbol{Q}||_{\mathbf{max}} = \mathbf{max}||\boldsymbol{Q}||$, ω in equation (26) can be calculated by equation (27).

$$\begin{aligned}
\omega =& \delta\dot{\boldsymbol{e}}^{jT}(\boldsymbol{K}_{\boldsymbol{d}}^0 + 2\boldsymbol{C}_1 - 2\boldsymbol{\Lambda M})\delta\dot{\boldsymbol{e}}^j + 2\boldsymbol{\Lambda}\delta\dot{\boldsymbol{e}}^{jT}(\boldsymbol{F}/\boldsymbol{\Lambda} - (\boldsymbol{C}_0 + \boldsymbol{C}_1 - \boldsymbol{\Lambda M}))\delta\boldsymbol{e}^j \\
&+ \boldsymbol{\Lambda}^2\delta\boldsymbol{e}^{jT}(\boldsymbol{K}_{\boldsymbol{d}}^0 + 2\boldsymbol{C}_0 + 2\boldsymbol{F}/\boldsymbol{\Lambda} - 2\dot{\boldsymbol{C}}_1/\boldsymbol{\Lambda})\delta\boldsymbol{e}^j \\
&\geq l_p||\delta\dot{\boldsymbol{e}}||^2 + 2\boldsymbol{\Lambda}\delta\dot{\boldsymbol{e}}^T\boldsymbol{Q}\delta\boldsymbol{e} + \boldsymbol{\Lambda}^2 l_r||\delta\boldsymbol{e}||^2 \\
&\geq l_p||\delta\dot{\boldsymbol{e}}||^2 - 2\boldsymbol{\Lambda}||\delta\dot{\boldsymbol{e}}||\boldsymbol{Q}||\delta\boldsymbol{e}|| + \boldsymbol{\Lambda}^2 l_r||\delta\boldsymbol{e}||^2 \\
&= l_p(||\delta\dot{\boldsymbol{e}}|| - \boldsymbol{\Lambda}/l_p(||\boldsymbol{Q}_{\mathbf{max}}||||\delta\boldsymbol{e}||))^2 + \boldsymbol{\Lambda}^2(l_r - 1/l_p(||\boldsymbol{Q}||_{\max}^2))||\delta\boldsymbol{e}||^2 \geq 0
\end{aligned} \tag{27}$$

Based on the above, it can be obtained that $\Delta V^j \leq 0$, which is shown in equation (28).

$$\Delta V^j = V^{j+1} - V^j \leq 0 \tag{28}$$

Since $\boldsymbol{K}_d^0$ is a positive definite matrix, V^j, and V^j the bounded, when $j \to \infty$, $\boldsymbol{y}^j(t) \to 0$. At the same time, since $\boldsymbol{e}^j$ and $\dot{\boldsymbol{e}}^j$ are two independent variables, $\boldsymbol{\Lambda}$ is a defined positive definite constant matrix, when $j \to \infty$, $\boldsymbol{e}^j \to 0$, $\dot{\boldsymbol{e}}^j \to 0$. The stability of the system can be expressed as $\boldsymbol{q}^j(t) \overset{j\to\infty}{\to} \boldsymbol{q}_d(t)$, $\dot{\boldsymbol{q}}^j(t) \overset{j\to\infty}{\to} \dot{\boldsymbol{q}}_d(t)$. This shows that the iterative learning control method proposed in this article will reduce the robot system equation (16) error with each iteration, ensuring that the control law equation (17) leads to stable and convergent behavior in consecutive iterations.

In the DBO-fuzzy-PD control strategy, the proportional (K_p) and derivative (K_d) gains are dynamically adjusted in real time to respond to external disturbances. The fuzzy logic controller determines the adjustments to K_p and K_d based on the tracking error and its rate of change. However, a first-order low-pass filter is applied to these gain adjustments to prevent abrupt changes and ensure smooth control actions. The algorithm principle of the first-order low-pass filter is shown in equation (29).

$$\boldsymbol{Y}^j = \alpha \boldsymbol{X}^j + (1 - \alpha)\boldsymbol{Y}^{j-1} \tag{29}$$

where j is the sampling number, α the filter coefficient, $\boldsymbol{X}^j$ the current sampling value, $\boldsymbol{Y}^{j-1}$ the previous filter output value, and $\boldsymbol{Y}^j$ the current filter output value.

The improved DBO algorithm is integrated with a fuzzy-PD adaptive control strategy to form the DBO-fuzzy-PD controller. This hybrid approach leverages the optimization capabilities of the DBO algorithm to initialize the PD control parameters, ensuring an optimal starting point for the control process. The FLC continuously adjusts the K_p and K_d gains in real time, based on the tracking error and its rate of change, to adapt to varying external disturbances. A first-order low-pass filter is applied to the adjustments of K_p and K_d. This filtering mechanism smooths the gain changes, reducing the impact of high-frequency noise and preventing abrupt variations that could destabilize the system. The iterative process of the DBO-fuzzy-PD controller, combined with the low-pass filtering, ensures precise and stable trajectory tracking for the 6-DOF collaborative robot manipulator, even under dynamic and unpredictable conditions.

Results and discussion

In this section, the performance and effectiveness of the proposed adaptive fuzzy PD controller, optimized using the improved DBO algorithm, are evaluated through a series of experiments and simulations. The content includes the smoothness of the inverse kinematics trajectory, the tracking performance of the fuzzy-PD controller, comparative analysis of DBO-fuzzy-PD with initial tracking results, motor torque responses, adaptive PD parameter adjustments, and an error analysis under different control strategies. These comprehensive evaluations provide insights into the controller's robustness, accuracy, and overall efficiency in achieving precise trajectory tracking for a 6-DOF collaborative robot manipulator.

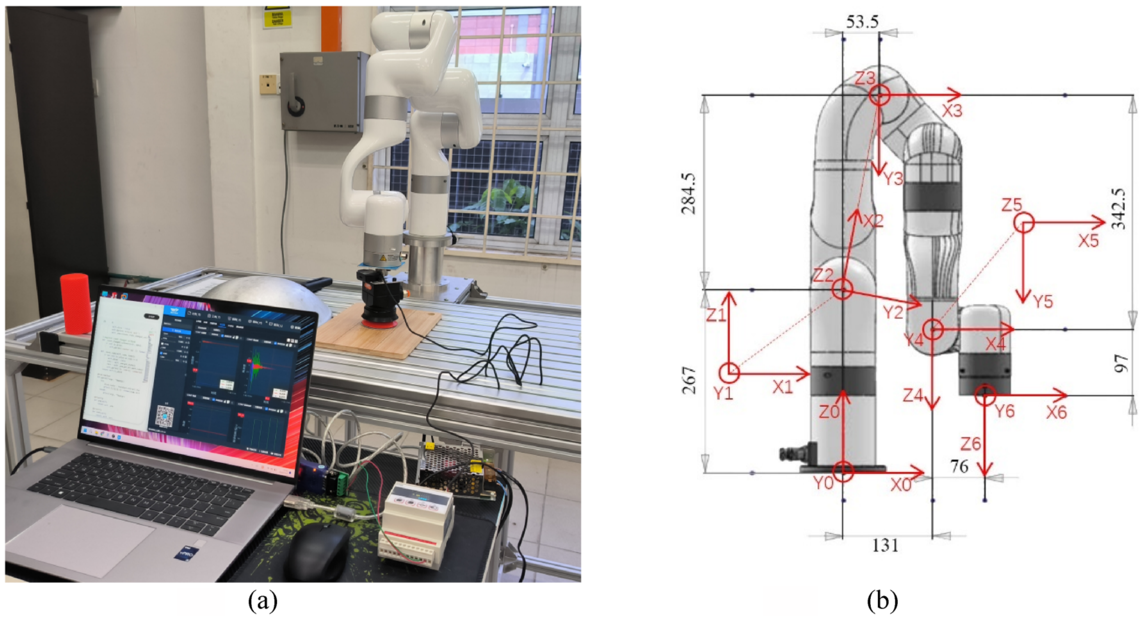


Figure 7. Robot and its modified D-H model. (a) Ufactory xArm6 and (b) modified D-H model.

Table 2. Modified D-H parameters of robot manipulator.

Joint i	a_{i-1}(mm)	α_{i-1} (deg)	d_i (mm)	Offset (deg)	θ_i limitations (deg)
1	0	0	267	0	[−360°, +360°]
2	0	−90°	0	−79.34995°	[−118°, 120°]
3	289.48866	0	0	79.34995°	[−225°, 11°]
4	77.5	−90°	342.5	0	[−360°, +360°]
5	0	90°	0	0	[−97°, 180°]
6	76	−90°	97	0	[−360°, +360°]

Table 3. Mass parameters for Ufactory xArm6.

Dynamics	Mass (kg)	Center of mass (mm)
Link1	2.177	[0.15, 27.24, −13.57]
Link2	2.011	[36.7, −220.9, 33.56]
Link3	1.725	[69.77, 113.5, 11.6]
Link4	1.211	[−0.2, 20.0, −26.0]
Link5	1.206	[63.87,29.3, 3.5]
Link6	0.17	[0, −6.77, −10.98]

Robot manipulator parameters

The robot manipulator used in this study is the Ufactory xArm6, a 6-DOF collaborative robot designed for precision tasks and flexibility in various applications.[24] The Ufactory xArm6 features a modified Denavit-Hartenberg (D-H) parameterization, which accurately models the robot's kinematic structure.[25] This parameterization is crucial for formulating the robot's kinematic and dynamic equations, enabling precise control and trajectory tracking. Figure 7(a) is the robot manipulator, and Figure 7(b) is the schematic diagram of the robot Modified D-H model.

A modified D-H parameter table was developed to accurately represent the kinematics and dynamics of the xArm6. This table includes the link lengths, twists, offsets, and joint angles that define the spatial relationships between consecutive links of the manipulator. Table 2 lists the modified D-H parameters used in this study.

Accurate modeling of the robot's dynamics requires the inclusion of mass and inertia properties for each link. These parameters influence the robot's response to control inputs and external forces, Mass parameters for Ufactory xArm6 are shown in Table 3. These parameters were derived through a combination of the manufacturer's specifications and precise measurements taken from the actual manipulator.

Trajectory preprocessing

To ensure the smooth and accurate movement of the robot's manipulator, trajectory planning is based on the principle of minimum jerk. This principle aims to minimize the third derivative of position, thus producing a smooth and continuous trajectory up to the third derivative. Figure 8 shows a certain motion of the inverse kinematics, and the trajectory is preprocessed into a third-order differentiable result.

Using minimum acceleration trajectory planning ensures that the trajectory input to Ufactory xArm6 is a third-order differentiable, the basis for the control robot manipulator.

Trajectory tracking under random disturbances

In real-world scenarios, robotic manipulators often encounter various disturbances that can affect tracking performance. These disturbances can arise from external forces, sensor noise, or dynamic changes in the environment. To evaluate the robustness and effectiveness of the proposed PD controller, an experiment was conducted to assess the manipulator's trajectory tracking under random disturbances.[26]

Random disturbances are introduced to simulate real-world perturbations. These disturbances include random external forces applied to the end effector and simulated sensor noise. Joint motion tracking is shown in Figure 9. It can be seen that under external interference, especially the motion error of the end joints gradually increases. At the same time, different impacts and disturbances have a huge impact on errors, which seriously restricts the precise work of the robot manipulator. This is why robot manipulators need controllers.

DBO-fuzzy-PD controller

The DBO-fuzzy-PD control strategy leverages the strengths of DBO and fuzzy logic to enhance the trajectory tracking performance of the 6-DOF collaborative robot manipulator. This hybrid approach begins with DBO to obtain the initial PD controller parameters and then employs a fuzzy controller to adjust these parameters in response to varying external disturbances.

Taking the IAE of the system as the optimization objective function, the improved DBO algorithm is used to optimize the initial PD controller parameters K_p and K_d, providing a robust starting point for the control system. The improved DBO algorithm efficiently explores the search space and ensures that the initial parameters are well suited to the desired trajectory under nominal

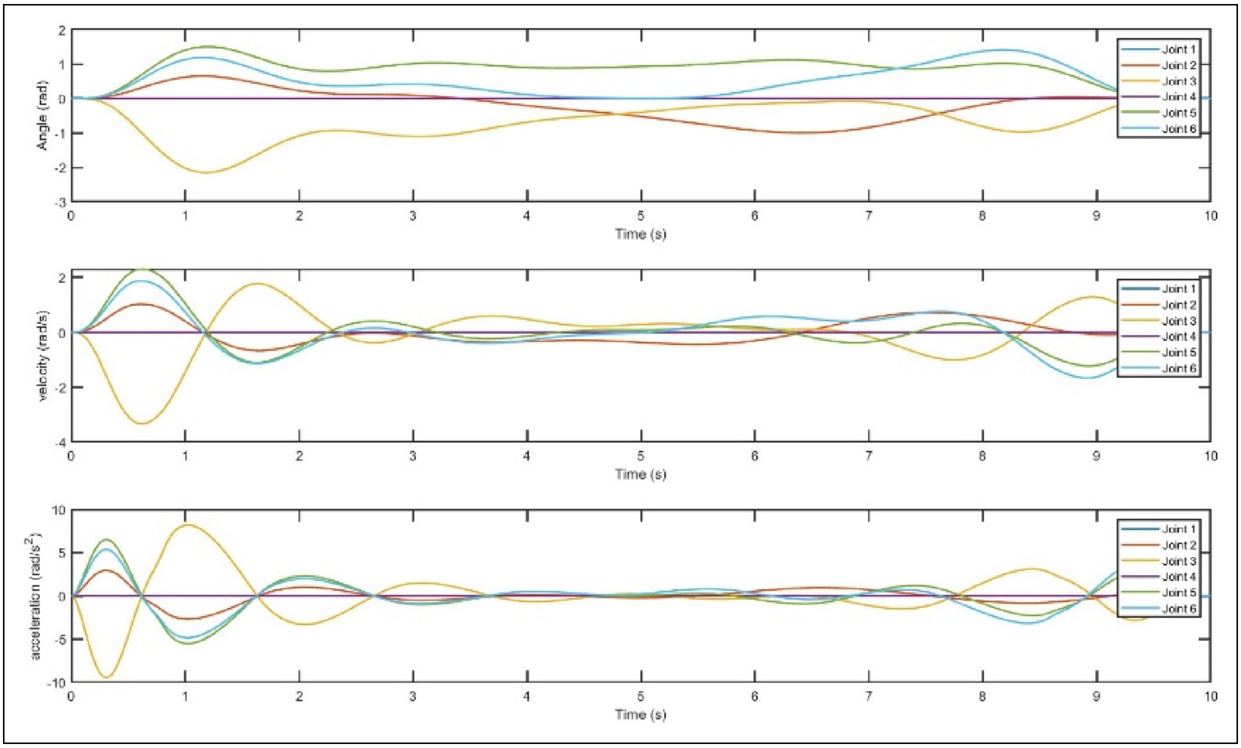


Figure 8. Three-order differentiable trajectories of six joints.

conditions. The optimal PD parameters sought by this study using the improved DBO algorithm in the initial perturbation are: $K_p = 120.2798$, and $K_d = 157.9241$.

While the DBO-optimized PD parameters provide a solid foundation, the presence of external disturbances can lead to varying performance. To address this, a fuzzy controller is integrated into the system to adaptively adjust the PD gains in real time based on the tracking error and its derivative.

The fuzzy controller continuously monitors the tracking error (e) and the change in error (Delta e). The controller adjusts the gains K_p and Kd based on predefined fuzzy rules.[27] The fuzzy rules of K_p and K_d are shown in Figure 10(a) and (b), respectively. These rules are designed to enhance the controller's responsiveness to different magnitudes and types of disturbances. The fuzzy controller outputs the adjusted gains, which modify the PD parameters dynamically, ensuring that the control system remains adaptive and robust.

The tracking accuracy of the DBO-fuzzy-PD controller is improved compared to the initial PD controller alone. The adaptive adjustments made by the fuzzy controller help to maintain the desired trajectory more closely. Figure 11 shows the results of joint trajectory tracking using the DBO-fuzzy-PD controller. The results show that the fuzzy controller effectively reduces the error by dynamically adjusting the PD gain. This adaptive mechanism enables the control system to respond quickly to disturbances and maintain stable tracking.

The effectiveness of the fuzzy adaptive mechanism can be observed through the real-time variation of the PD gains during the trajectory tracking process. When the tracking error increases due to an external disturbance, the fuzzy controller quickly adjusts the gains to counteract the deviation and bring the system back to the desired trajectory. Adjusting the controller parameters according to changes in error and error rate of change helps to suppress oscillations and prevent overshoot, which ensures that the system remains stable and avoids excessive oscillatory behavior.[28]

The performance of the DBO-fuzzy-PD controller was compared against the fuzzy-PD controller and the traditional PD controller in the trajectory tracking of a 6-DOF robotic manipulator under identical external disturbances. The experiments measured the integral absolute error (IAE) and integral square error (ISE) for overall and individual joint performance. The key performance indicators of IAE and ISE of the results of the DBO-fuzzy-PD controller, fuzzy-PD controller, and PD controller are shown in Table 4.

The results demonstrate that the DBO-fuzzy-PD controller significantly outperformed the other control strategies. The IAE and ISE values for the DBO-fuzzy-PD controller were consistently lower, indicating superior accuracy and robustness in tracking the desired joint trajectories despite external disturbances. These findings confirm that the

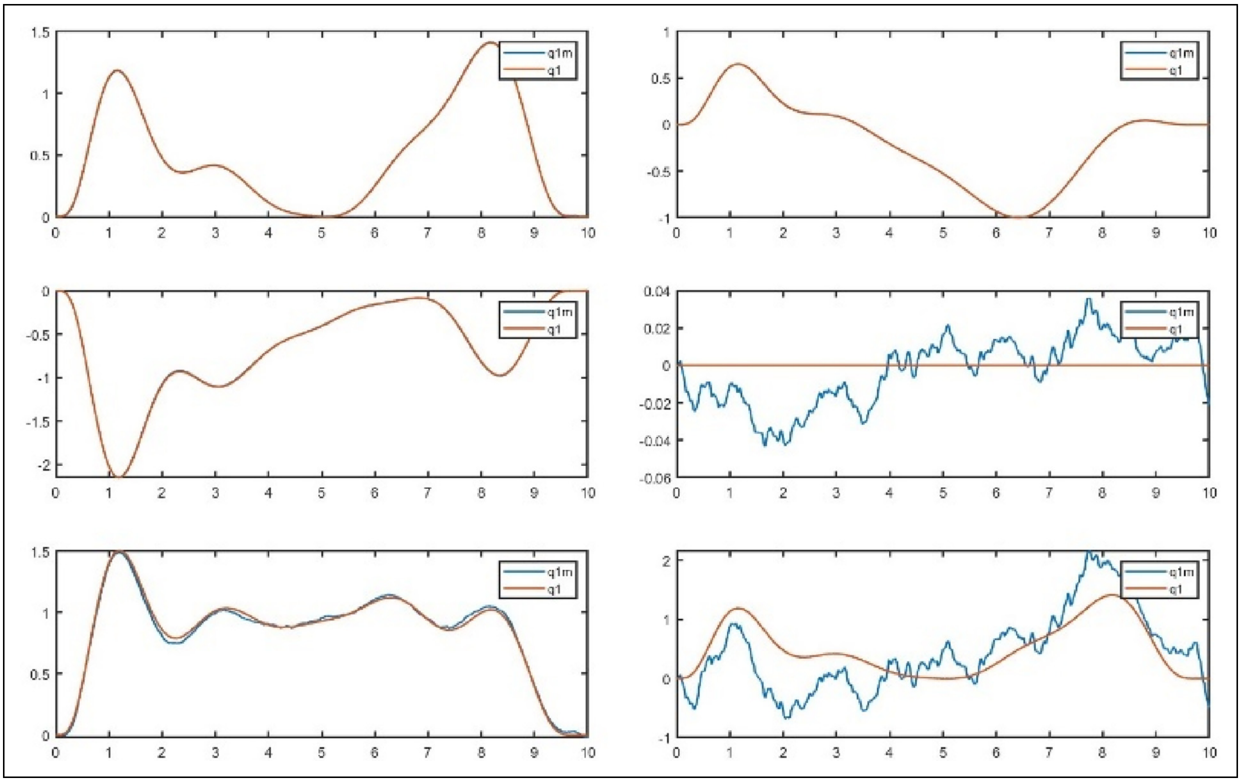


Figure 9. Joint tracking under external disturbances.

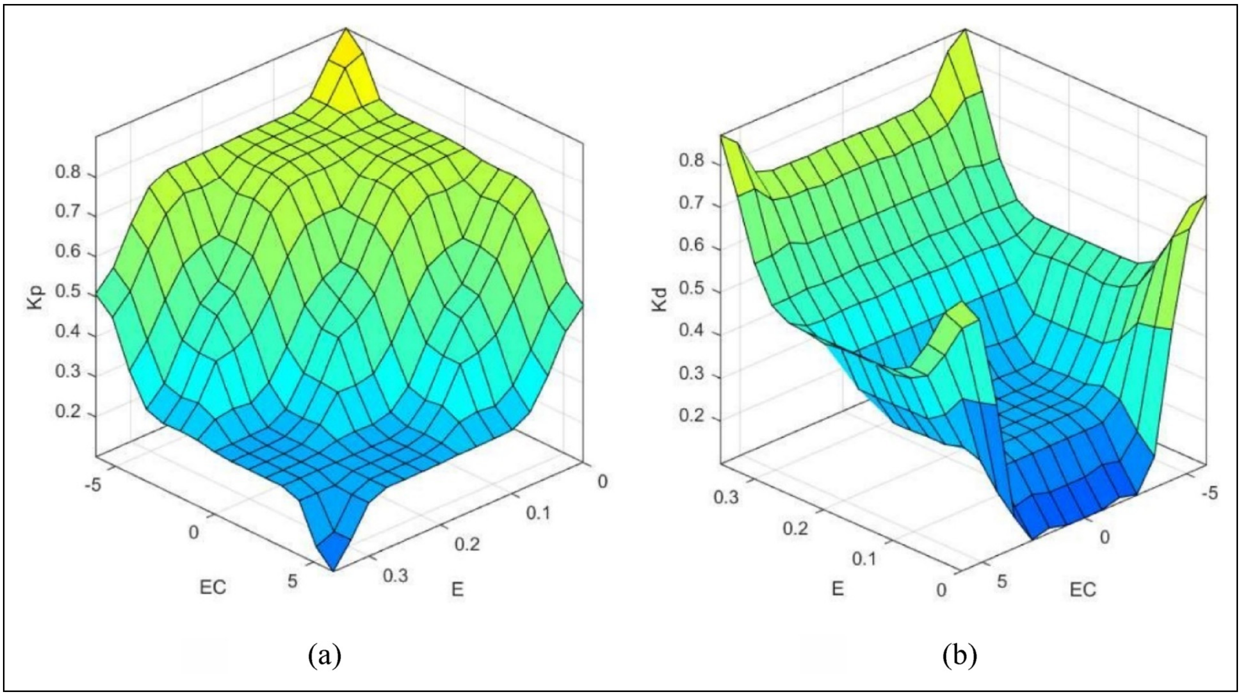


Figure 10. Fuzzy rules for (a) Kp and (b) Kd.

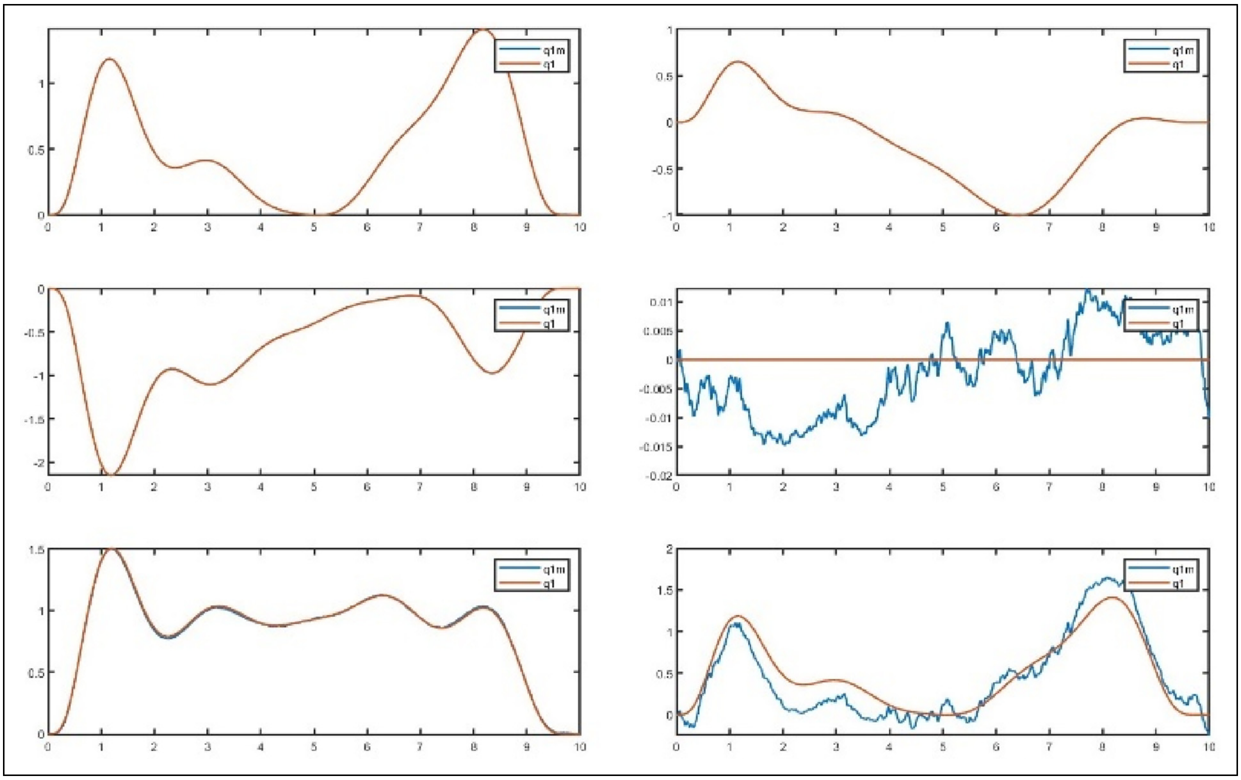


Figure 11. Tracking using the dung beetle optimization-fuzzy-proportional derivative controller.

Table 4. Comparison of different errors, ISE, and IAE as performance indices.

Controller	Indices	Total	Joint 1	Joint 2	Joint 3	Joint 4	Joint 5	Joint 6
DBO Fuzzy PD	ISE	0.0384	e-07	e-07	e-06	e-05	0.0001	0.0383
	IAE	0.1843	0.0007	0.0003	0.0014	0.0064	0.0085	0.1670
Fuzzy PD	ISE	0.0595	e-06	e-07	e-06	e-05	e-04	0.0592
	IAE	0.2332	0.0010	e-04	0.0017	0.0080	0.0108	0.2114
PD	ISE	3.4140	0.0003	0.0001	0.0027	0.0963	0.0621	3.2525
	IAE	1.9876	0.0132	0.0080	0.0412	0.2587	0.2120	1.4545

DBO: dung beetle optimization; IAE: integral absolute error; ISE: integral square error; PD: proportional derivative.

DBO-fuzzy-PD controller can more efficiently and effectively maintain precise control over the robotic manipulator's movements, ensuring optimal performance under challenging conditions.

The fuzzy adaptive mechanism continuously monitors the tracking error (e) and the change in tracking error (de/dt) to adjust the proportional (K_p) and derivative (K_d) gains of the PD controller. The tracking error is the difference between the desired and actual positions of the robot joints. Based on the output of the FIS, the fuzzy controller adjusts the PD gains in real time, and the results after first-order low-pass filtering are shown in Figure 12. These adjustments ensure that the controller adapts to changing conditions, maintaining optimal performance.

The fuzzy controller adjusts the PD gains in real time by multiplying the optimal K_p and K_d values derived under stable conditions to obtain dynamically adjusted values, ensuring that the controller adapts to continuous external disturbances while maintaining optimal performance. This adaptive adjustment allows the controller to effectively respond to changing disturbances and ensure accurate trajectory tracking. The torque graphs in Figure 13 calculate and illustrate the adaptive joint torques under external disturbances.

The comparison between the adaptive joint torque and the ideal calculated torque in Figure 13 illustrates that the adaptive DBO-fuzzy-PD controller is highly sensitive to rapid changes in the error signal and external disturbances, and it can dynamically adjust the K_p and K_d gains in real time. The control method proposed in this article has applied a low-pass filter equation (29) to the adaptive gains to reduce the rapid fluctuations of K_p and K_d. Introducing a boundary layer or saturation in the fuzzy gain adjustment

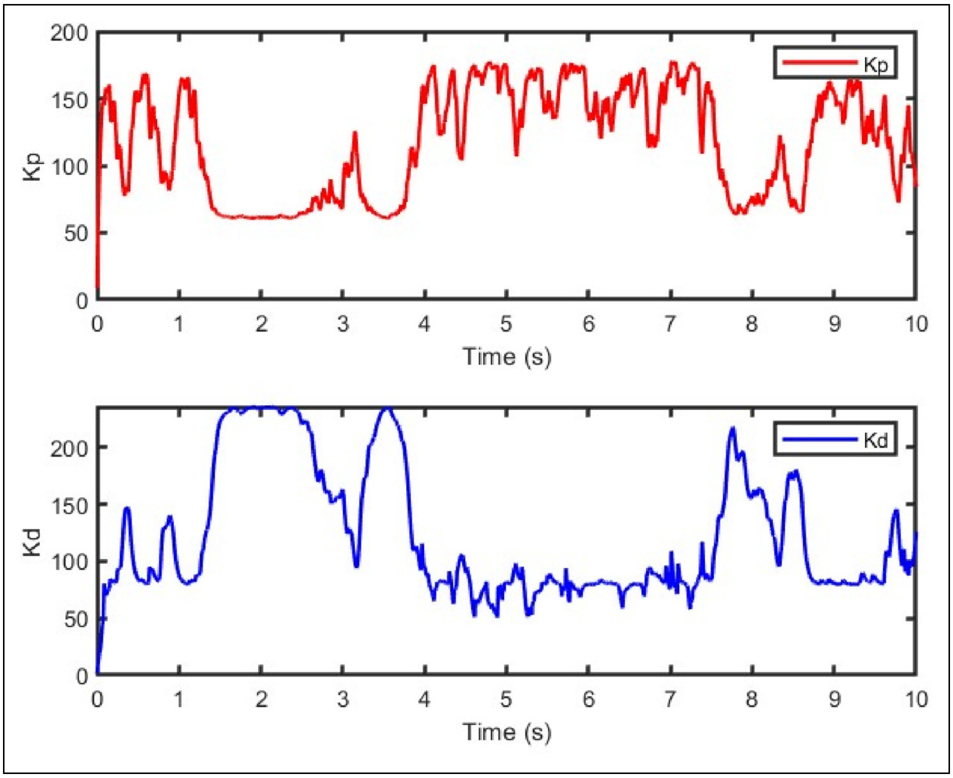


Figure 12. Adaptive change of proportional derivative controller parameters.

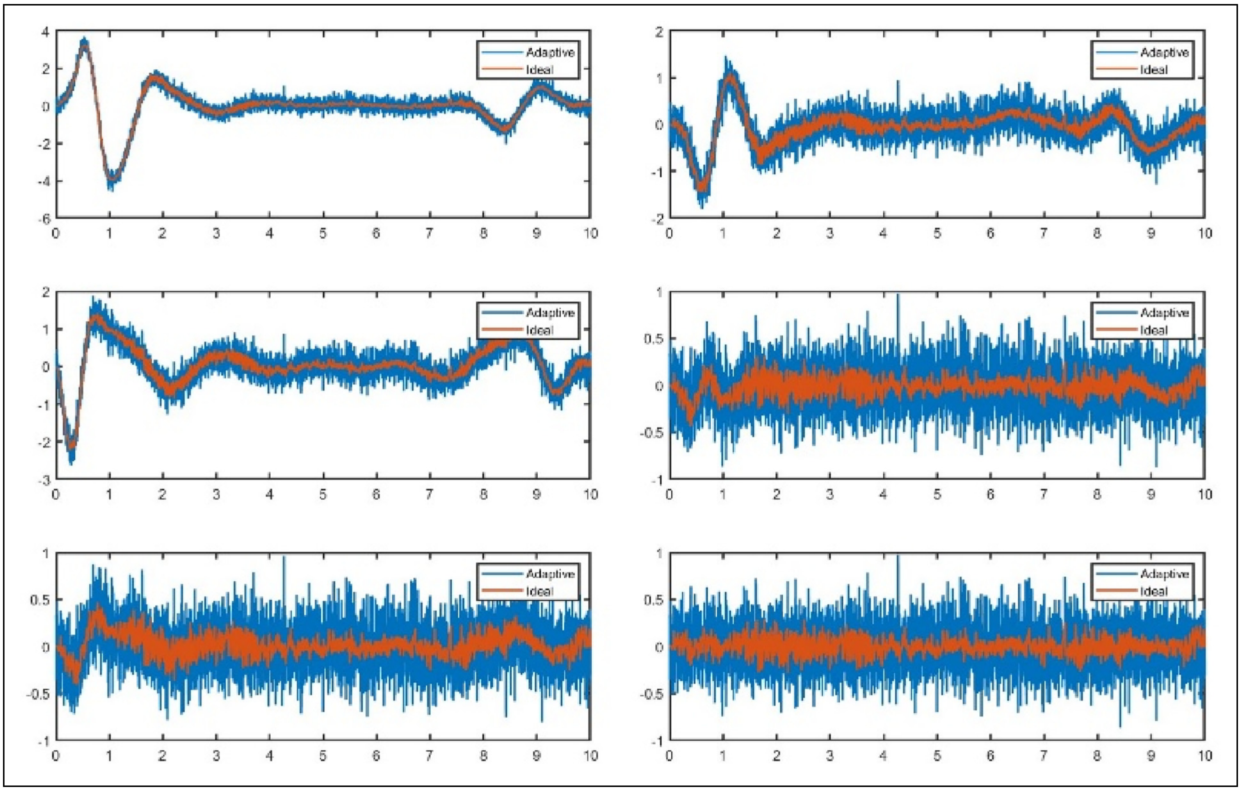


Figure 13. Adaptive joint torque and ideal calculated torque.

to deal with disturbances and noise can help avoid overreaction to small disturbances.

DBO-fuzzy-PD controller combined the initial optimization of PD parameters using DBO with the real-time adaptive capabilities of a fuzzy logic controller. The IAE and ISE values were significantly lower than those of the other controllers. This indicates superior tracking accuracy and robust performance, as the fuzzy controller continuously adjusted the gains to minimize errors in the presence of disturbances.

The DBO-fuzzy-PD controller improved the overall tracking accuracy and demonstrated enhanced robustness and stability in the face of external disturbances. These findings validate the effectiveness of integrating the improved DBO algorithm with fuzzy adaptive control for high-precision robotic applications, making it a promising approach for future advancements in robotic control systems.

Conclusion

This study proposes the DBO-fuzzy-PD controller to enhance the trajectory tracking performance of a 6-DOF collaborative robot manipulator. The DBO algorithm optimizes the initial PD controller parameters and provides a robust starting point for trajectory tracking. The fuzzy logic controller continuously adjusts the PD gain based on the tracking error and its rate of change. This real-time adaptability enables control systems to maintain high performance despite external disturbances and changing conditions. The DBO-fuzzy-PD controller maintains stable and accurate trajectory tracking. This article uses discrete Lyapunov iterative stability analysis to prove the global asymptotic stability of the robot manipulator system. The adaptive nature of the fuzzy controller ensures that the system can respond quickly to disturbances and maintain optimal performance.

The effectiveness of the DBO-fuzzy-PD controller is verified through extensive simulations and experiments. The performance of the DBO-fuzzy-PD controller is significantly better than that of the traditional PD controller, the ISE value is reduced from 3.4140 to 0.0384, and the IAE value is reduced from 1.9876 to 0.1843. It can be seen that the adaptive fuzzy strategy provides dynamic gain adjustment and enhances the system's ability to maintain precise and stable control under different conditions.

In summary, the PD control method combining DBO-fuzzy provides a powerful solution to achieve high accuracy and robustness in trajectory tracking tasks. This research lays a solid foundation for the future development of advanced robot control systems and contributes to the continuous progress of collaborative robots.

ORCID iDs

Ma Haohao https://orcid.org/0009-0001-6964-2357
Azizan As'arry https://orcid.org/0000-0001-6804-3697

Statements and declarations

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the Gansu Province University Teachers Innovation Fund Project (No.: 2025A-143), Tianshui Normal University Scientific Research Innovation Platform (No.: PTJ2023-01), Ministry of Higher Education under the Fundamental Research Grant Scheme (FRGS/1/2015/TK03/UPM/02/7) and the Universiti Putra Malaysia (UPM) under the Geran Insentif Putra Siswazah (GP-IPS) fund (800-2/1/2021/GP-IPS/9697000).

Conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Supplemental material

Supplemental material for this article is available online.

References

1. Olubodun PA, Woodside MR and Bristow DA. Procedures and performance of a robotic machining system with metrology-in-the-loop feedback control. *Manuf Lett* 2023; 35: 361–370.
2. Zhen S, Peng X, Liu X, et al. A new PD based robust control method for the robot joint module. *Mech Syst Signal Process* 2021; 161: 107958.
3. Chaudhary H, Panwar V, Sukavanum N, et al. Fuzzy PD+I based hybrid force/ position control of an industrial robot manipulator. *IFAC Proc Volumes* 2014; 47: 429–436.
4. Ulici IA, Codrean A and Natsakis T. Human-robot interaction with sliding mode control for rehabilitation. *IFAC-PapersOnLine* 2023; 56: 1133–1138.
5. Spong MW, Hutchinson S and Vidyasagar M. *Robot modeling and control*. New York: John Wiley & Sons, 2006.
6. Muñoz-Vázquez AJ, Gaxiola F, Martínez-Reyes F, et al. A fuzzy fractional-order control of robotic manipulators with PID error manifolds. *Appl Soft Comput* 2019; 83: 105646.
7. Bonyadi MR and Michalewicz Z. Particle swarm optimization for single objective continuous space problems: a review. *Evol Comput* 2017; 25: 1–54.
8. Gerges F, Zouein G and Azar D. Genetic Algorithms with Local Optima Handling to Solve Sudoku Puzzles. Presented at the Proceedings of the 2018 International Conference on Computing and Artificial Intelligence, Chengdu, China, 2018.
9. Xue J-K and Shen B. Dung beetle optimizer: a new metaheuristic algorithm for global optimization. *J Supercomput* 2023; 79: 7305–7336.
10. Moyrón J, Moreno-Valenzuela J and Sandoval J. Global regulation of flexible joint robots with input saturation by nonlinear I-PID-type control. *IEEE Trans Control Syst Technol* 2024; 32: 2385–2393.

11. Moreno-Valenzuela J, Martinez-Lopez M and Sandoval J. Static anti-windup for global regulation of constrained Euler-Lagrange systems. *IEEE Trans Circuits Syst Express Briefs* 2024; 71: 3870–3874.
12. Moreno-Valenzuela J, Moyrón J and Montoya-Chairez J. Limited integrator antiwindup-based control of input-constrained manipulators. *IEEE Trans Ind Electron* 2024; 71: 1738–1748.
13. Khorashadizadeh S and Sadeghijaleh M. Adaptive fuzzy tracking control of robot manipulators actuated by permanent magnet synchronous motors. *Comput Electr Eng* 2018; 72: 100–111.
14. Birjandi SAB, Fortunić EP and Haddadin S. Evaluation of robot manipulator link velocity and acceleration observer. *IFAC-PapersOnLine* 2023; 56: 292–299.
15. Saraswat R and Suhag S. Type-2 fuzzy logic PID control for efficient power balance in an AC microgrid. *Sustainable Energy Technol Assess* 2023; 56: 103048.
16. Sain D and Mohan BM. Mathematical modelling of some simplest fuzzy two-term (PI/PD) controllers. *IFAC-Papers OnLine* 2022; 55: 276–281.
17. Oh S-K, Jang H-J and Pedrycz W. Optimized fuzzy PD cascade controller: a comparative analysis and design. *Simul Model Pract Theory* 2011; 19: 181–195.
18. Siva Krishna P and Gopi Krishna Rao PV. Fractional-order PID controller for blood pressure regulation using genetic algorithm. *Biomed Signal Process Control* 2024; 88: 105564.
19. Zhu F, Li G, Tang H, et al. Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems. *Expert Syst Appl* 2024; 236: 121219.
20. Xiao Y, Cui H, Hussien AG, et al. MSAO: a multi-strategy boosted snow ablation optimizer for global optimization and real-world engineering applications. *Adv Eng Inf* 2024; 61: 102464.
21. Zhang X, Wen S and Wang D. Multi-population biogeography-based optimization algorithm and its application to image segmentation. *Appl Soft Comput* 2022; 124: 109005.
22. Zhang D, Zhang C, Han X, et al. Improved DBO-VMD and optimized DBN-ELM based fault diagnosis for control valve. *Meas Sci Technol* 2024; 35: 075103.
23. Righettini P, Strada R, Cortinovis F, et al. An experimental investigation of the dynamic performances of a high speed 4-DOF 5R parallel robot using inverse dynamics control. *Robotics* 2024; 13: 54.
24. Lu J, Zou T and Jiang X. A neural network based approach to inverse kinematics problem for general six-axis robots. *Sensors* 2022; 22: 8909.
25. Liang X, Yao Z, Deng W, et al. Adaptive control of n-link hydraulic manipulators with gravity and friction identification. *Nonlinear Dyn* 2023; 111: 19093–19109.
26. Yang J, Sun T and Yang H. Spatial hybrid adaptive impedance learning control for robots in repetitive interactive tasks. *ISA Trans* 2023; 138: 151–159.
27. Deng X and Chen J. The lower partial moments risk measure in a novel fuzzy framework based on possibility density function. *Comput Ind Eng* 2022; 169: 108309.
28. Arya PP and Chakrabarty S. Controller considerations for attenuating noise in process variables in presence of uncertainties in plant parameters. *IFAC-PapersOnLine* 2022; 55: 832–837.