scientific reports

Check for updates

OPEN

A new method for recognizing geometric parameters of industrial robots

Bin Kou[1]✉ & Yi Zhang[2]

Intelligent algorithms that are commonly used to obtain errors in the geometric parameters of industrial robots have a low accuracy, easily fall into the local optimal solution, and involve complicated coding such that they are unsuitable for use in engineering. In this study, we first apply the D-H method to establish a model of error in industrial robots, and then use the set of errors in their geometric parameters as the objective function. Following this, we improve the accuracy of global optimization of the particle swarm optimization (PSO) algorithm by drawing on the wandering behavior of the wolf pack algorithm and hybridization behavior of the genetic algorithm. We balance the convergence of the PSO algorithm by using a linearly diminishing weight. This leads to an improved PSO algorithm that can accurately determine errors in the geometric parameters of industrial robots. We compared our improve PSO algorithm with commonly used particle swarm algorithms, and the results showed that the former had a higher accuracy of convergence on average. Moreover, the errors in the geometric parameters obtained by the improved PSO algorithm can enhance the accuracy of localization of errors in industrial robots.

Keywords Computer applications, Particle swarm algorithms, Industrial robots, Localization errors

Factors such as the deformation of the rods connecting industrial robots affect positioning accuracy[1,2]. The resultant positioning error in industrial robots is usually handled through error modeling, acquisition of error-related data, and parameter identification and compensation. Commonly used error models to this end include the D-H and MD-H models[3]. The identification of error in their geometric parameters is the focus of research on compensating for localization errors in industrial robots, and a number of methods have been developed to this end. Marquardt et al. proposed the Levenberg–Marquardt (L–M) algorithm to identify error in the geometric parameters of robots, but this algorithm has inefficient iterations, and the influence of the initial systemic parameters on it reduces its accuracy of identification[4].

With the development of intelligent algorithms, a number of evolutionary algorithms have been used to control the optimization of robots. Mohamed et al. used the zebra optimization algorithm, artificial gorilla troops optimizer, Coot optimization algorithm, and a neural network to optimize the controller of a robot, and verified the performance of their method through simulations[5–7]. These results have helped widen the application of manipulators to various fields[8]. Jiang et al. used the least absolute shrinkage and selection operator (LASSO) algorithm to identify the kinematic parameters of industrial robots, and showed that it is more effective than the common least-squares method[9]. Cao et al. used an improved migration learning method for error compensation in a two-robot system, which reduced the average error of the two robots to 0.594 mm and 0.255 mm[10]. Sergio et al. proposed an online method of error compensation that can evaluate different operating conditions of the robot to reduce its online error by about 90%[11]. Fang et al. used the quantum particle swarm algorithm to identify errors in the geometric parameters of parallel and series robots, but this algorithm was too complicated to code and difficult to implement[12]. Robert et al. used the simulated annealing algorithm to identify the geometric parameters of robots, but this method suffered from slow convergence and extended periods of time to complete its iterations[13]. Zhong et al. used a neural network to inversely compensate for the localization error of the PUMA robot. However, this algorithm required a large amount of training data and had a poor overall efficiency[14]. In summary, each of the above algorithms has advantages as well as disadvantages when used to compensate for the localization error in industrial robots. We consider the particle swarm optimization (PSO) algorithm in this study because of its simple parameters, highly efficient iterations, and wide use[15,16]. However, the classical PSO algorithm is prone to premature maturity in the late iteration stage, which reduces the identification accuracy of the robot's geometric error parameters.

[1]School of Software, Taiyuan University of Technology, Taiyuan, China. [2]Xi'an BZT Electronic Technology Co., Xi'an, China. ✉email: bkou18@fudan.edu.cn

In this study, we combine the wandering behavior represented in the wolf pack optimization algorithm with the global optimal solution of the PSO algorithm to improve its ability to search for the globally optimal solution[17–19]. Moreover, we propose a hybridization-based, stochastic perturbation-related, linearly decreasing particle swarm optimization (BWPSO-RP) algorithm by drawing on the stochastic behaviors of the hybridized PSO algorithm and artificial fish swarm optimization algorithm to improve the diversity of the swarm[20–22]. We also use a strategy of linearly decrease in values of the inertial parameters of the PSO algorithm[23] to improve its ability to search for the globally optimal solution. Finally, we verify the accuracy of the proposed algorithm through simulations that involve modeling the D-H error of the robot.

Robot D-H model

KR10R1420 robot

In this paper, a KR10R1420 robot from KUKA was used for the study[24], which is a six-degree-of-freedom tandem industrial robot with six rotary joints as shown in Fig. 1:

As can be seen in Fig. 1, the KUKA industrial robot consists of a base, a large arm and a small arm, as well as a wrist mechanism with six rotary joints.

D-H model of KR10R1420 robot

The commonly used model for robots is the DH model, which is widely used in industrial robots, and the coordinate system of the robot is established by the DH method as shown in Fig. 2 below[24]:

The DH parameters of the KUKA robot can be further obtained by referring to the robot coordinate system in Fig. 2, see Table 2 below.

Adaptation function and robot position generation

Postural generation

The validity of the proposed algorithm was verified by the KR10R1420 robot manufactured by Kuka (KUKA) Robotics Ltd. the parameters of the DH model of the KR10R1420 robot are shown in Table 1. the errors of the geometric parameters are shown in Table 2[24].

We randomly generated errors in the intervals of [−0.25,0.25] (unit: mm) and [−0.05,0.05] (unit: rad), and added them to the nominal geometric parameters of the robot provided in Table 1 to obtain the angles of its joints. We then compared these values with the theoretical angle of the robotic joints to obtain the adaptation value.

Fitness function

fitness function used is shown in Eq. 1 below:

$$F = \min\left(\sum_{i=1}^{N} \sqrt{\left((\delta P_{xi})^2 + (\delta P_{yi})^2 + (\delta P_{zi})^2 \right)}\right) \tag{1}$$

N in Eq. (1) denotes the number of robot error sampling points, and N is 20 in this paper.where N denotes the number of sampling points of error and was set to 20 in this paper. F is a function of $(\Delta a_i, \Delta d_i, \Delta a_i, \Delta \theta_i,)$, and represents the error ensemble of N points. When the robot moves in space, the difference between the nominal and actual positions of each of its joints is obtained by sampling different points. The improved PSO algorithm is then used to obtain the actual parameters of error, which are in turn used to compensate for the error in the robot to improve its final accuracy of positioning.

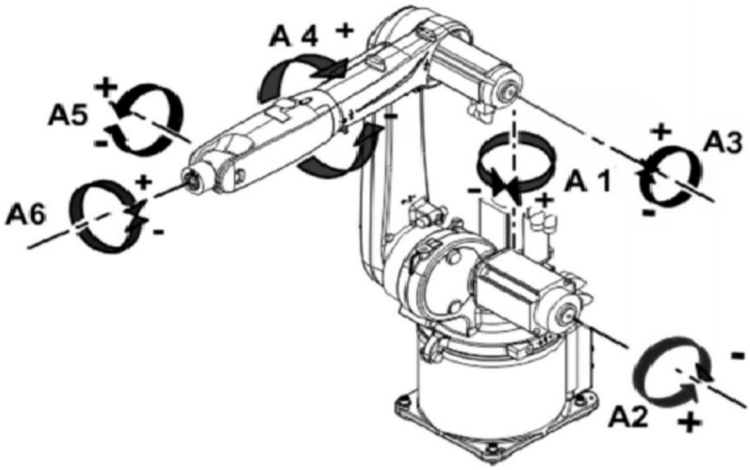


Fig. 1. KUKA robot.

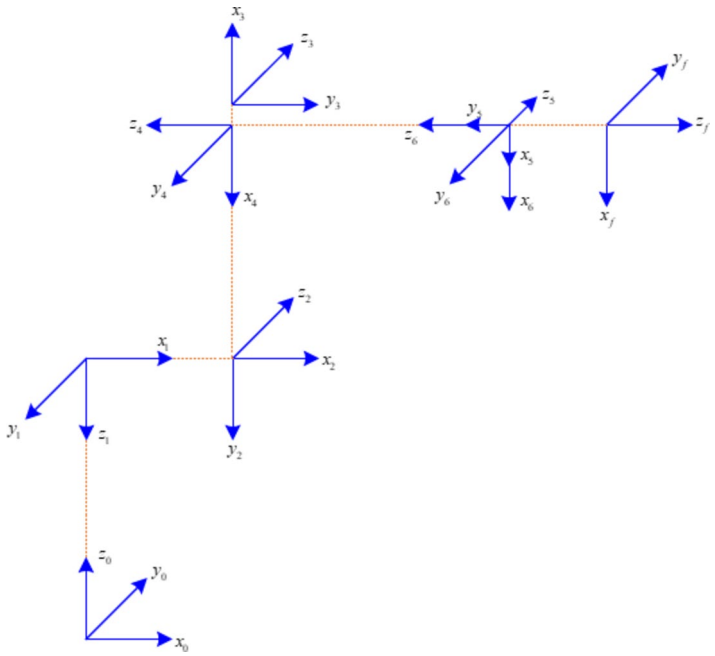


Fig. 2. KUKA robot coordinate system.

Jointi	a_i/mm	α_i/deg	d_i/mm	$?_i$/deg
1	150	− 90	450	-185-185
2	610	0	0	-155-35
3	20	− 90	0	-130-154
4	0	90	660	-350-350
5	0	− 90	0	-130-130
6	0	0	160	-350-350

Table 1. Nominal geometric parameters of KUKA robot.

Jointi	Δa_i/mm	$\Delta \alpha_i$/rad	Δ_i/mm	Δd_i/mm
1	− 0.16	− 0.01	− 0.02	0.18
2	− 0.04	0	0.01	− 0.13
3	− 0.01	0.03	0.0	0.21
4	0.11	− 0.01	− 0.03	− 0.06
5	0.22	0.0	0.01	0.05
6	0.05	− 0.02	− 0.02	0.08

Table 2. Geometric parameters errors of KUKA robot.

Principles of improved particle swarm algorithm

When the traditional PSO algorithm is used to obtain error in the geometric parameters of the industrial robot, it can easily converge prematurely owing to its stochastic nature, and the large number of parameters that need to be solved for. The solution to these parameters constitutes a problem of optimizing a high-dimensional function with constraints. We propose improving the traditional PSO algorithm below.

Improved global optimal solution

The wandering behavior of the wolf pack optimization algorithm enables the wolves to identify the position of the prey. The relevant expression is as follows:

$$X_{id}^{p} = X_{id} + sin(2\pi \times P/h) \times step_a^d \tag{2}$$

where h is the total number of directions searched by a probing wolf, P denotes a certain direction ($p = 1, 2, \ldots, h$), and X_{id} denotes the current position of the probing wolf. Its position in d-dimensional space after optimization is, and $step_a$ is the length of its wandering step.

The global optimal solution to the PSO algorithm is a single particle that lacks the capability for local search. We propose enabling the algorithm to search for the global optimal solution Pg according to the wandering behavior of the wolf pack algorithm.

We then have:

$$P\text{gnew}=\text{Pg}+\sin(2\pi \times P/h) \times step \tag{3}$$

Population-level stochastic perturbation behavior improvement

The accuracy of late convergence of the PSO algorithm is low due to the loss of diversity in its population. Stochastic behavior is a search-related strategy that can improve the diversity of the given population, and has been widely used in a variety of intelligent algorithms, including the fish swarm optimization algorithm. We also draw on the hybridized PSO algorithm in the iterative process of the particle swarm. A specified number of particles are selected according to their probability of hybridization, and are entered into a hybridization pool to improve the diversity of the population[20–22].

Inspired by the above, we integrate stochastic behavior into the PSO algorithm. Some particles are first randomly perturbed in each iteration, and some of them are then optimally replaced according to the fitness value to enhance the diversity of the particle swarm.

$$X_{\text{new}} = X * (1 + \text{rand}) \tag{4}$$

In Eq. (4) rand represents a random number within 0 to 1, so some particles are selected from the particle swarm according to a certain probability $_{Pm}$, and their fitness values are compared with X^* (1 + rand) to optimize the next search. In this way the solution of geometric error parameters of industrial robots is optimized from the population level.

Linearly decreasing weight values *w*

Based on work by Shi et al.[25], we use a linearly decreasing weight as follows:

$$w=w\text{max}\text{-}(w_{\text{max}}\text{-}w_{\text{min}})*\frac{T}{T\text{max}} \tag{5}$$

where w_{max} denotes the maximum value of the inertial weight, w_{min} is its minimum value, T denotes the number of iterations of the particle swarm at the given time, and T_{max} denotes its total number of iterations.

Improved particle swarm algorithm for identifying geometric parameters

The flow of the proposed improved PSO algorithm is as follows:

1. The algorithm is initialized and the fitness value of each particle is calculated.
2. The particle swarm is updated according to the standard formulae for updating its velocity and position. The weight coefficients w are updated according to the process of linear reduction.
3. The globally optimal solution is searched for according to the wandering behavior of the wolf pack algorithm. If the new solution obtained from the search is better than the previous solution, it is set as the new globally optimal solution. Error in the geometric parameters of the robot is locally optimized based on this globally optimal solution.
4. Some particles from the population are selected according to a certain probability, Pm, and their fitness values are compared with Pg*(1 + rand) to optimize the next search. This optimizes the parameters related to the geometric error of the robot at the level of the overall population.
5. Determine whether the condition for terminating the iterations of the algorithm has been met. If it has not been met, return to step 2. If it has, output the globally optimal solution and obtain the relevant optimal parameters.

The flow of the proposed improved PSO algorithm is shown in Fig. 3.

Simulation experiment and result analysis

The geometric parameters of the robot were identified by using the proposed improved PSO algorithm in comparison with the basic PSO algorithm, linearly decreasing weights-based PSO (WPSO) algorithm, and hybridization-based PSO (BPSO) algorithm in MATLAB R2016b.

The population for the proposed PSO method was set to 200, the learning factors were $c_1 = c_2 = 1.49$, the inertial weights were $w_{max} = 0.8$ and $w_{min} = 0.4$, the coefficient of variation in the population during the iterative process of the algorithm was $p_m = 0.2$, and the coefficient of the iterative process of the globally optimal solution was $P = i$, where i is the number of iterations of the particle swarm at the given time. The direction T was set to 200, and signified the total number of iterations of the swarm. The step size was set to two, the maximum number of iterations was set to 200, and we used the average of six iterations and the average fitness value of each algorithm, as shown in Table 3.

Table 3 shows that the classical PSO algorithm had an average fitness value of 5.39 over six iterations, and delivered the worst performance. The average fitness values of the BPSO and WPSO algorithms were 4.58 and

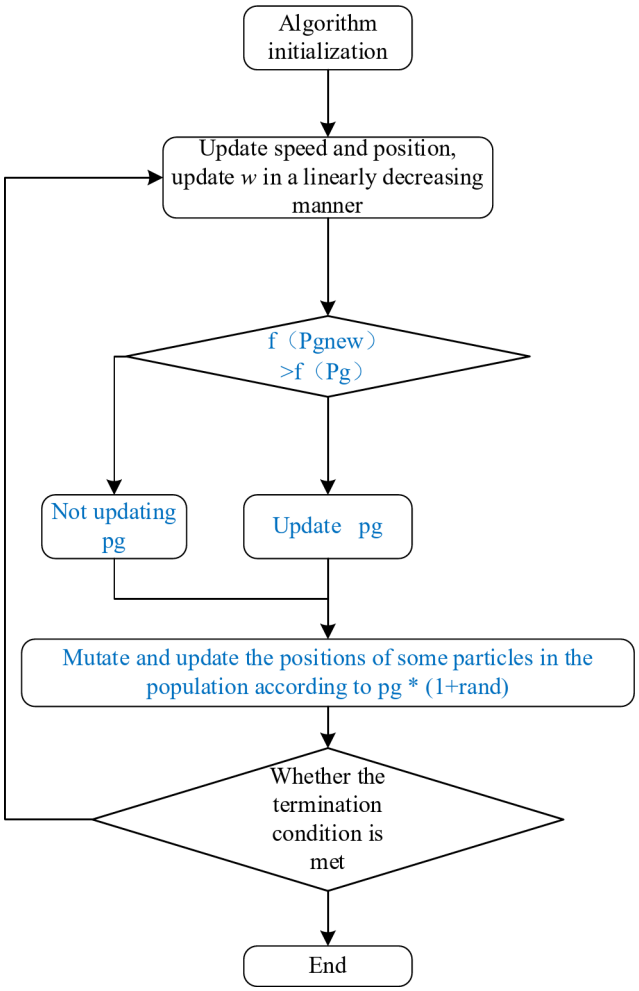


Fig. 3. Improve PSO process diagram.

Arithmetic	PSO	WPSO	BPSO	BWPSO-RP
Average adaptation	5.39	4.02	4.58	3.12

Table 3. Comparison of experimental results.

4.02, respectively, while the proposed BWPSO-RP algorithm yielded the best average fitness value of 3.12. Its average adaptive accuracy was higher than those of the PSO, WPSO, and BPSO by 42.1%, 22.4%, and 31.9%, respectively.

The curve of optimal iterations of the proposed method in Fig. 4.

The trend of iterations of the improved PSO algorithm is shown in Fig. 4. Combined with the information listed in Table 3, it is evident that it had a higher accuracy of convergence than the ordinary PSO algorithm when updating the speed and positional information of the swarm because its particles had been optimized during the iterative process.

Figure 5 shows the absolute positional errors of the robot along the x-, y-, and z-axes before and after its optimization by the proposed method.

It is evident from Fig. 5 that before error compensation, the absolute positional errors of the industrial robot on the x-, y-, and z-axes oscillated in the interval [− 40, 40] mm, with a maximum error of 40 mm. After error compensation, these errors decreased, and oscillated in the interval [− 0.25, 0.25] mm with a maximum error of 0.25 mm.

Figure 6 shows that the absolute positional error of the robot oscillated in the interval [5,40] mm before error compensation, with a maximum value close to 40 mm and a minimum close to 5 mm. After error compensation, its absolute positional error decreased to oscillate in the interval [0.0.2] mm, with minimum and maximum errors of 0 mm and 0.2 mm, respectively. Therefore, the improved PSO algorithm improved the accuracy of positioning of the industrial robot. This, along with Table 3, shows that the proposed method also had the optimal stability.

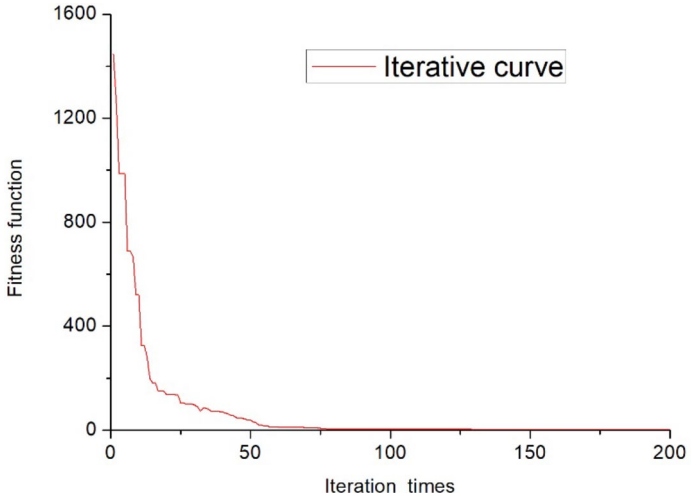


Fig. 4. iterative curve of algorithm.

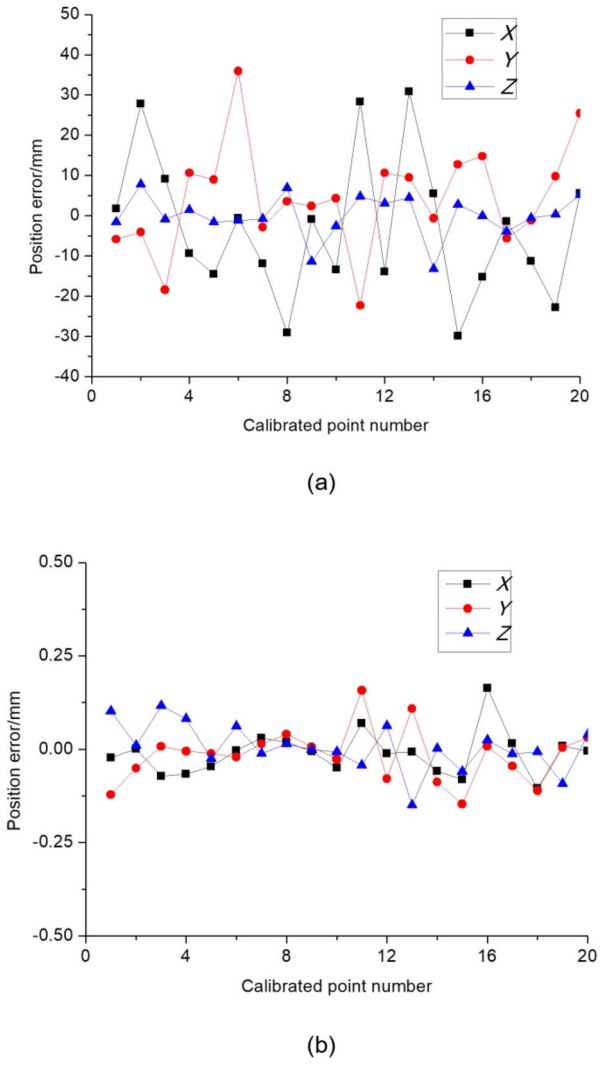


Fig. 5. Error in x, y, and z axes before and after robot calibration.

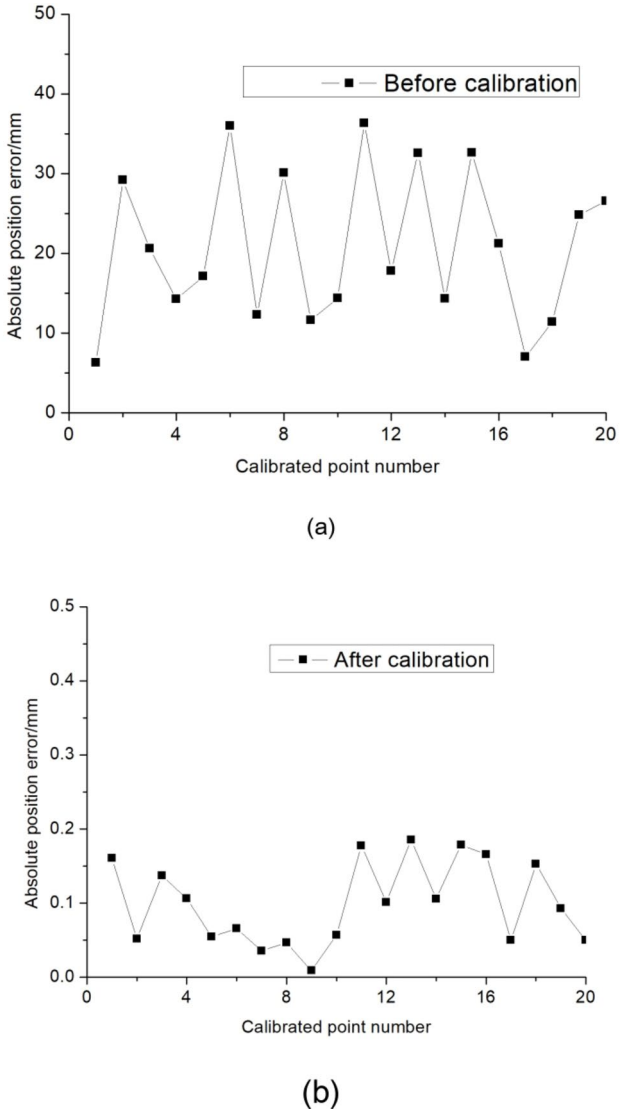


Fig. 6. Absolute position error before and after robot calibration.

To further validate the effectiveness of the proposed algorithm, we randomly generated 15 points in the space of movement of the industrial robot. The errors in the absolute position of the robot along the x-, y-, and z-axes on the validation set before and after its calibration are shown in Fig. 7.

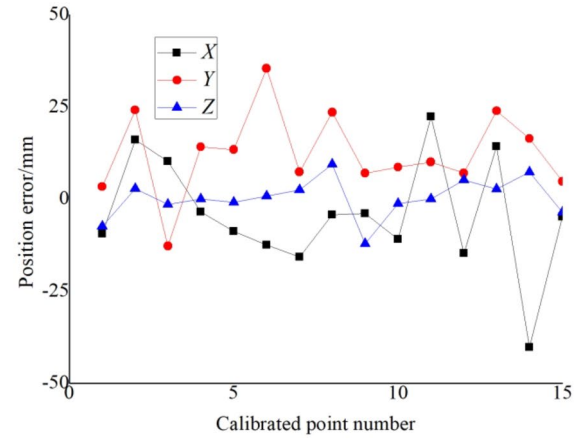
Figure 7 shows that error in the position of the robot on the x-, y-, and z-axes oscillated in the interval [−30,30] mm before calibration. The error incurred in the calculation of individual points was close to 50 mm. After calibration based on the proposed BWPSO-RP, however, these errors decreased to oscillate in the interval [−0.2,0.2] mm. The error at certain points was close to 0 mm. This further verifies the effectiveness of the BWPSO-RP algorithm.

Figure 8 shows errors in the absolute position of the robot on the validation set before and after its calibration.

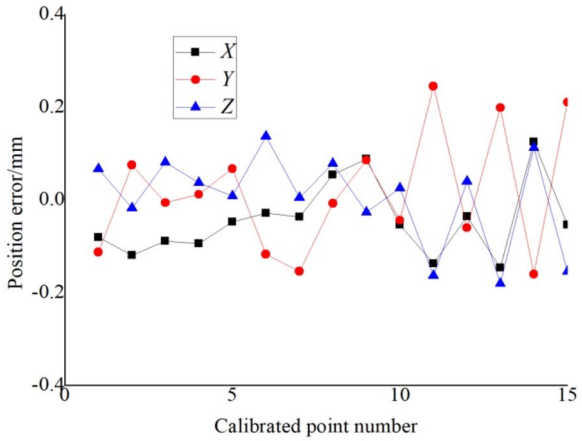
From Fig. 8, it can be seen that the absolute position error of the validation set before calibration is in the interval of [-10,50] mm, and the absolute position error of most points is near 25 mm, and the absolute position error of the validation set falls back to [0.1,0.4] mm after calibration with the BWPSO-RP algorithm, and the absolute position error of most of the points is in the interval of [0.1,0.2] mm, and from the validation results of the validation set It can be seen that the BWPSO-RP algorithm can effectively improve the absolute localization accuracy of industrial robots.

Concluding remarks

1. This paper proposes a new particle swarm algorithm, combined with the iterative law of the standard particle swarm algorithm, firstly adopts the linearly decreasing weight value *w*, and then introduces the wandering behavior of the wolf pack algorithm into the iterative process of the particle swarm algorithm, which improves the local optimization ability of the particle swarm algorithm, and meanwhile adopts the hybrid



(a)



(b)

Fig. 7. Error in *X*, *Y*, and *Z* axes before and after robot calibration (validation set).

operation of the genetic algorithm in the late stage of particle swarm algorithm iteration, which further improves the global optimization ability of the particle swarm algorithm. Simulation experiments show that compared with the common particle swarm algorithms such as PSO, WPSO, BPSO, the BWPSO-RP algorithm proposed in this paper, the average adaptive accuracy is improved in order, 42.1%, 22.4%, 31.9%. The results of the BWPSO-RP algorithm identification can effectively improve the end localization error of industrial robots.

2. Future work in the field should seek to apply the idea used to improve the PSO algorithm in this study to other intelligent algorithms, such as the cuckoo and ant colony algorithms. This is because the globally optimal solution of many intelligent algorithms is a key factor influencing their overall convergence.
3. The algorithm proposed in this paper can also be used in robot path planning and friction force parameter identification, but the corresponding experiments are needed to verify the effectiveness of the algorithm proposed in this paper.

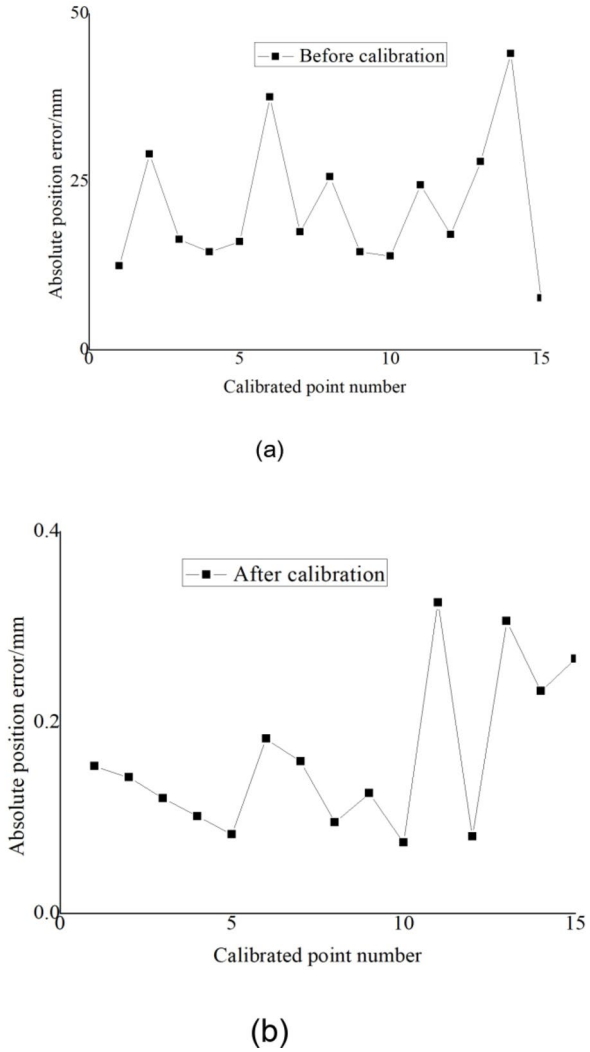


Fig. 8. Absolute position error before and after robot calibration (validation set).

Data availability

Data Availability: The datasets generated and/or analysed during the current study are not publicly available duedue to Privacy and school related regulations.but are available from the corresponding author on reasonable request.

Received: 27 November 2024; Accepted: 15 January 2025
Published online: 22 January 2025

References

1. Chen, Y. & Dong, F. Robot machining recent development and future research issues. *Int. J. Adv. Manuf. Techology*. **66** (9–12), 1489. https://doi.org/10.1007/s00170-012-4433-4 (2013).
2. Chen, G. et al. Review on kinematics cali-bration technology of serial robots. *Int. J. Precision Eng. Manuf.* **15** (8), 1759. https://doi.org/10.1007/s12541-014-0528-1 (2014).
3. Xu, B. *Research on Kinematic Calibration Method of Sixdegree of Freedom Industrial Robot* (Jiangxi University of Science and Technology, 2020).
4. Marquardt, D. W. An algorithm for least-squares estimation of nonlinear parameters. *J. Soc. Ind. Appl. Math.* **11** (2), 431–441 (1963).
5. Jasim Mohamed, M., Oleiwi, B. K., Azar, A. T. & Mahlous, A. R. Hybrid controller with neural network PID/FOPID operations for two-link rigid robot manipulator based on the zebra optimization algorithm. *Front. Robot AI* **11** (2024).
6. Mohamed, M. J., Oleiwi, B. K., Azar, A. T. & Hameed, I. A. Coot optimization algorithm-tuned neural network-enhanced PID controllers for robust trajectory tracking of three-link rigid robot manipulator. *Heliyon*. **10**(13), e32661 (2024).
7. Mohamed, M. J., Oleiwi, B. K., Abood, L. H., Azar, A. T. & Hameed, I. A. Neural fractional order PID controllers design for 2-link rigid robot manipulator. *Fractal Fract.* **7**, 693 (2023).
8. Abdelmaksoud, S. I., Al-Mola, M. H., Abro, G. E. M. & Asirvadam, V. S. In-depth review of advanced control strategies and cutting-edge trends in robot manipulators: analyzing the latest developments and techniques. *IEEE Access.* (2024).
9. Jiang, G. et al. High-precision robotic kinematic parameter identification and positioning error compensation method for industrial robot. *Meas. Sci. Technol.* **35**(5), 055016 (2024).

10. Cao, S., Wang, H., Guo, Y., Zhu, W. & Ke, Y. A method for predicting relative position errors in dual-robot systems via knowledge transfer from geometric and nongeometric calibration. *Industrial Robot: Int. J. Rob. Res. Application*. **51**(2), 314–325 (2024).
11. Ferrarini, S., Bilancia, P., Raffaeli, R., Peruzzini, M. & Pellicciari, M. A method for the assessment and compensation of positioning errors in industrial robots. *Robot. Comput. Integr. Manuf.* **85**, 102622 (2024).
12. Fang, L. & Dang, P. Kinematic calibration method of robots based on quantum-be-haved particle swarm optimization. *J. Mech. Eng.* **52** (7), 23. https://doi.org/10.3901/JME.2016.07.023 (2016).
13. Horning, R. J. A comparison of identification techniques for robot calibration. Master Dissertation (Case Western Reserve University, 1998).
14. Zhong, X., Lewis, J. & N-Nagy, F. L. Inverse robot calibration using artificial neural networks. *Eng. Appl. Artif. Intell.* **9**(1), 83–93 (1996).
15. Ozcan, E. & Mohan, C. K. Analysis of a simple particle swarm optimization system. In *Intelligent Engineering Systems Through Artificial Neural Networks*. vol. 8, 253–258 (1998).
16. Ozcan, E. & Mohan, C. K. Particle swarm optimization: surfing the waves. In *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99, 1999*. 6–9. (1999).
17. Wu, H. S., Zhang, F. & Wu, L. A new swarm intelligence algorithm - wolf pack algorithm. *Syst. Eng. Electron. Technol.* **35**(11), 2430–2438 (2013).
18. Zhang, S. & Zhou, Y. Q. Grey wolf optimizer with ranking-based mutation operator for IIR model identification. *Chin. J. Electron.* **27**(5), 071–1079 (2018).
19. Kou, B., Ren, D. & Guo, S. Geometric parameter identification of medical robot based on improved Beetle Antennae search algorithm. *Bioengineering* (2022).
20. Yu, C. Application of hybrid particle swarm optimization algorithm in tourism route optimization with fuzzy value evaluation. *Digit. Technol. Appl.* (2022). (040–005).
21. Li, X. A. *New Intelligent Optimization Method - Artificial Fish Swarm Algorithm* (Zhejiang University, 2003).
22. Kou, B. et al. A new method for identifying kinetic parameters of industrial robots. *Actuators* (2022).
23. Yang, F. *Revision of Finite Element Model for Bridge Structures Based on Response Surface Linear Decreasing Weight Particle Swarm Algorithm* (Chang'an University, 2021).
24. Tan, D. C. *Research on Compensation for End Positioning Error of 6-DOF Industrial Robot* (Jilin Institute of Chemical Technology, 2024).
25. Shi, Y. & Eberhart, R. C. Empirical study of particle swarm optimization. In *Proc of Congress on Computational Intelligence, Washington DC, USA*, 1945–1950 (1999).

Author contributions
Bin KOU , Manuscript, original manuscript, experimental design, paper editing; Yi ZHANG , paper editing;

Declarations

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to B.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025