

Inverse kinematics of six degrees of freedom robot manipulator based on improved dung beetle optimizer algorithm

Ma Haohao^{1,2}, Azizan As'arry², Zhang Haoyang³, Mohd Idris Shah Ismail²,
Hafiz Rashidi Ramli⁴, Mohd Yusoff Moh Zuhri^{5,6}, Aidin Delgoshaei²

¹Faculty of Mechatronics and Automotive Engineering, Tianshui Normal University, Tianshui, China

²Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia

³College of Electrical and Information Engineering, Lanzhou University of Technology, Lanzhou, China

⁴Department of Electrical and Electronic Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang, Malaysia

⁵Advanced Engineering Materials and Composites Research Centre (AEMC), Department of Mechanical and Manufacturing Engineering, University Putra Malaysia, Serdang, Malaysia

⁶Laboratory of Biocomposite Technology, Institute of Tropical Forestry and Forest Product (INTROP), University Putra Malaysia, Serdang, Malaysia

Article Info

Article history:

Received May 29, 2024

Revised Jun 24, 2024

Accepted Jul 7, 2024

Keywords:

Dung beetle optimizer

Inverse kinematics

Orientation error

Quaternion

Robotics

ABSTRACT

Inverse kinematics is a basic problem in robotics, which aims to solve the robot's joint angles according to the end effector's position and orientation. This paper proposed an improved spiral search multi-strategy dung beetle optimizer (DBO) algorithm for solving the inverse kinematics problem. The improved DBO algorithm considers not only the error between the target value and the current value but also the previous position of the robot to ensure minimum displacement during the movement. To solve the end position error and orientation error of the robot end effector more accurately, the quaternion is introduced as a penalty factor in the optimization objective function, which is of great significance for reducing the orientation error. Through the improved DBO algorithm, the position error is still accurate, and the orientation error is reduced from 9.5901 to 1.8718. Experimental results show that the proposed algorithm outperforms other swarm-intelligent algorithms in terms of accuracy and convergence speed. Overall, the proposed spiral search multi-strategy DBO algorithm provides an effective and efficient solution to the inverse kinematics problem in robotics.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



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

1. INTRODUCTION

Robots are playing an increasingly important role in many areas of industry, from manufacturing and assembly to healthcare and education. They are able to perform a wide range of tasks, from simple pick-and-place operations to complex manipulations and interactions with humans [1]. Traditional industrial robots are typically large, heavy, and designed to perform repetitive tasks in isolated work cells. They are programmed to perform a specific task and do not typically have the ability to adapt to changes in their environment or interact safely with humans [2], [3]. Collaborative robots are a rapidly growing field in the robotics industry [4]. They are designed to work safely alongside humans, enhancing productivity and efficiency in manufacturing, healthcare, and other industries [5]. In order to perform their tasks, Cobots require precise control over their movements, which is achieved through inverse kinematics calculations.

One major problem that provides various challenges in robotics is inverse kinematics. Although it has been widely studied over the past few decades, a number of methods have been proposed for solving inverse kinematics problems, including numerical methods, analytical methods, and optimization-based methods. Numerical methods, such as the iterative Jacobian-based method and the Newton-Raphson method, are commonly used in real-time applications due to their efficiency. Analytical methods, such as the closed-form solution and the product-of-exponentials formula, have the advantage of providing an exact solution but are limited to certain types of robot geometries. Optimization-based methods, such as the gradient-based method and the particle swarm optimization (PSO), are flexible and can be applied to a wide range of robot structures, but may require longer computation time [6].

Xu [7] proposes an improved modal approach to solve task-oriented inverse kinematics, solves a four degrees of freedom (4-DOF) manipulator problem in groups, and determines the Cartesian coordinates of each node. Sancaktar [8] used the PSO algorithm to solve the inverse kinematics solution of a 6-DOF robot for external fixator fracture treatment and reduction, and tested the algorithm to obtain preliminary results, but the algorithm has a large error. Hu [9] established a pose coupling equation for a 5-DOF series robot to solve the inverse kinematics problem, and explained the improved inverse kinematics modeling process, proving that this solution can solve the inverse kinematics of the robot. There are several existing methods for solving inverse kinematics of redundant manipulators, such as numerical optimization algorithms, analytical methods, and artificial intelligence-based techniques. However, these methods have their own strengths and weaknesses [10]–[13].

Numerical optimization algorithms are effective in solving complex problems, but they may have convergence issues and require a careful selection of parameters. On the other hand, analytical methods are efficient and have closed-form solutions, but they are limited to certain types of manipulators and cannot handle complex geometries. Artificial intelligence-based techniques, such as neural networks and genetic algorithms, have shown promising results, but they may require large amounts of training data and lack interpretability [14]–[16].

This study proposes an improved Dung beetle optimizer (DBO) algorithm to solve the robot manipulator inverse kinematics problem. The DBO algorithm is a nature-inspired metaheuristic optimization algorithm that mimics the behavior of dung beetles in finding food sources [17], [18]. It has shown promising results in solving various optimization problems, including inverse kinematics of manipulators. The use of the DBO algorithm for inverse kinematics of redundant manipulators may overcome the limitations of existing methods and provide a more efficient and effective solution [19]. This study will discuss the inverse kinematics calculation and control of a robot manipulator using the improved DBO algorithm.

2. METHOD

This section describes a method for solving the inverse kinematics of a 6-DOF robot arm using the improved DBO algorithm. The Denavit-Hartenberg (DH) parameters of a robot manipulator are described, which are the basis for defining its geometry and kinematic characteristics. The improved DBO algorithm is discussed, focusing on enhancements made to optimize its performance in solving inverse kinematics problems. Finally, the fitness function of the robot's inverse kinematics is defined to evaluate the accuracy of the robot's end-effector in reaching the desired position and orientation.

2.1. Robot system and DH parameters

The collaborative robot used in this study is a 6-DOF manipulator whose model is Ufactory xArm 6. The robot is composed of a base, six links, and a tool end effector. The manipulator is controlled by a computer that sends motion commands to the robot's servo motors.

The DH method is commonly used to describe the kinematics of robotic manipulators. In this method, four parameters are defined for each joint, which are the link length, the link twist, the link offset, and the joint angle. By combining the DH parameters of all joints, the forward kinematics of the robot can be determined. The standard DH method, also known as S-DH, is a widely used method for describing the kinematics of manipulator robots. However, it has some limitations when dealing with complex structures, such as redundant manipulators, or when considering non-standard joint geometries. This led to the development of the modified DH method, also known as M-DH, which is a more flexible approach that can handle a wider range of robot configurations [20]. The main difference between S-DH and M-DH conventions lies in how they handle the assignment of reference frames and the choice of coordinate systems for defining the DH parameters. In S-DH, the reference frames are located at the joint centers, which can lead to singularities and inaccuracies in certain cases. In contrast, M-DH provides greater flexibility and accuracy. The M-DH parameters of the manipulator used in this study are listed in Figure 1 (From the robot company UFACTORY website) and Table 1.

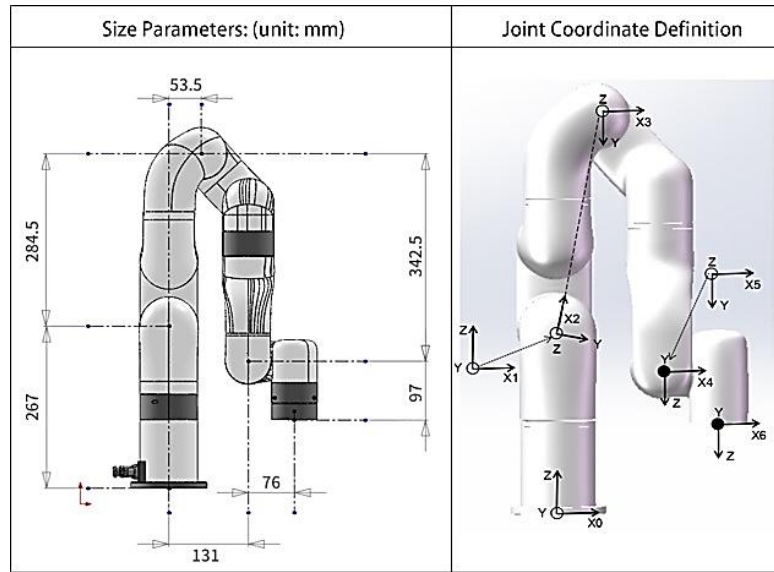


Figure 1. UFACTORY xArm 6 modified D-H parameters

Table 1. DH parameters of the robot Arm

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

2.2. Robot forward kinematics and workspace

The forward kinematics can be solved using different methods, including the DH method, homogeneous transformation matrices, and geometric algorithms [21]. The DH method is commonly used and provides a systematic way to relate the position and orientation of each link to the previous one. According to the connecting rod parameters in Table 1, the expression of the end effector in the base coordinate system can be obtained as (1). The homogeneous matrix ${}_{i-1}^i T$ transforms the frame attached to link $i-1$ into the frame attached to link i . “s” and “c” represent sine and cosine functions respectively. The final transformation is represented in (2) from the end effector frame to the base frame, where $\mathbf{R}_{3 \times 3}$ is the rotation matrix and $\mathbf{P}_{3 \times 1}$ is the position vector.

$${}_{i-1}^i T = \begin{bmatrix} c\theta_i & -c\alpha_i s\theta_i & s\alpha_i s\theta_i & \alpha_i c\theta_i \\ s\theta_i & c\alpha_i c\theta_i & -c\theta_i s\alpha_i & \alpha_i s\theta_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$${}^6 T = {}^1 T {}^2 T {}^3 T {}^4 T {}^5 T {}^6 T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3 \times 3} & \mathbf{P}_{3 \times 1} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2)$$

The workspace of a robot can be described using various methods, such as geometric modeling or numerical methods. In this study, the Monte Carlo simulation method can be used to estimate the workspace of a robot. This method involves randomly sampling joint angles within the joint limits and computing the corresponding end-effector positions. By repeating this process many times, the probability density function of the end-effector positions can be estimated, and the workspace can be determined by identifying the region where the probability density function is non-zero. The workspace is used to determine the feasible region of the robot's end effector as shown in Figure 2. The working range of the robot is flexible and can reach many positions. The working range of the X/Y axis is $[-762, +762]$, and the working range of the Z axis is

[-416, 1029], unit: *mm*. The workspace is a critical factor in determining the feasibility and effectiveness of inverse kinematics solutions. The size and shape of the workspace depend on various factors, including the robot's mechanical design, kinematic structure, and joint ranges.

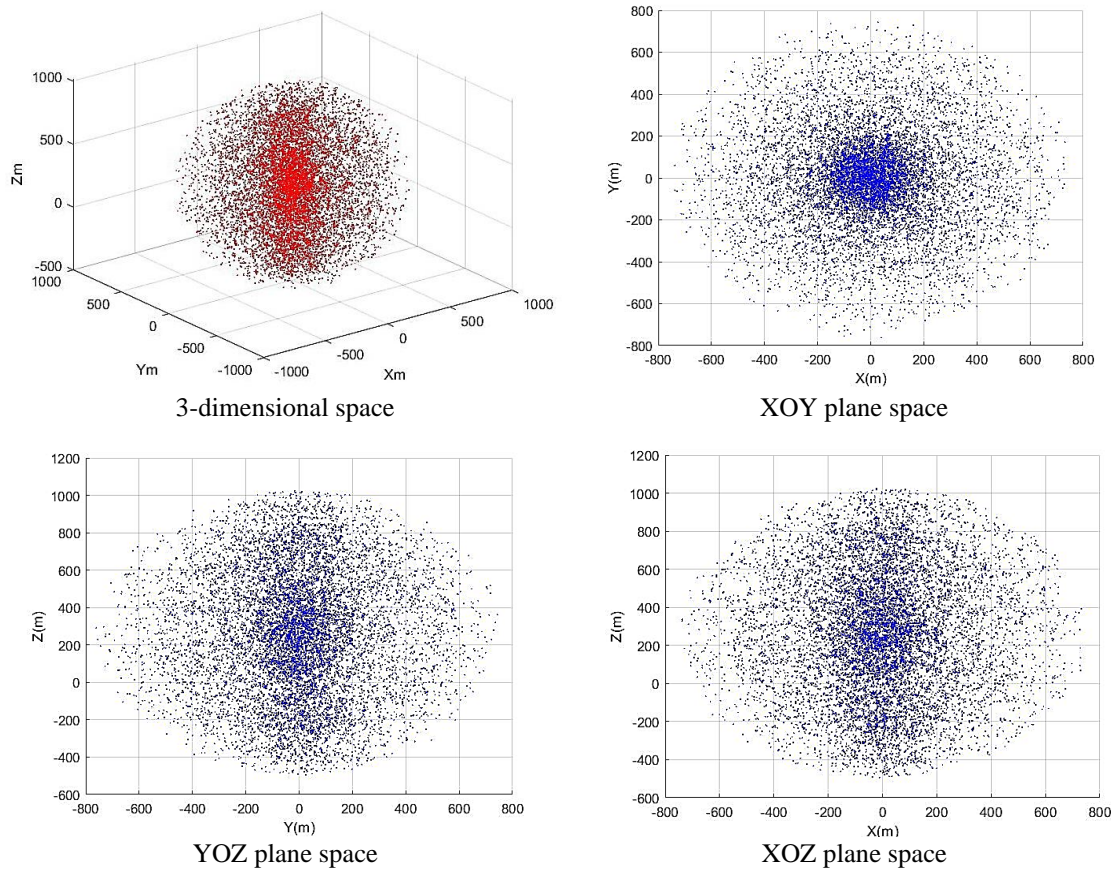


Figure 2. The workspace of robot xArm6

2.3. Improved DBO algorithm

The DBO algorithm is a recent nature-inspired optimization algorithm based on the behavior of dung beetles in finding and rolling dung balls. The algorithm is inspired by the observation that dung beetles use a heuristic method to find the optimal direction to roll their dung balls while avoiding obstacles. In the DBO algorithm, the population of dung beetles is represented by a set of candidate solutions, and each dung beetle has a position in the search space. The algorithm iteratively updates the positions of the dung beetles based on their fitness values, which are evaluated by an objective function. The default parameters for the DBO algorithm include the number of search agents, which is set to 30, and the initial position of the agents, which is randomly distributed within the search space. The radius of the dung ball, which affects the search step size, is also set to a default value of 1. The maximum number of iterations is set to 500, with an option to terminate the optimization process earlier if the convergence criterion is met.

In the original DBO algorithm, the reproduction of the population is expanded according to the current population spawning area. This strategy can easily cause the regional population to converge too quickly, reducing population diversity and easily falling into a locally optimal solution. The improved algorithm was inspired by the Whale Optimization Algorithm to round up prey. During the breeding stage, a spiral search strategy was used to update the prey position, which can not only ensure the diversity of the population, but also improve the convergence speed. The formula for the prey hunting stage is shown in (3).

$$\begin{cases} X(t+1) = D' \cdot e^{kl} \cdot \cos(2\pi l) + X^*(t) \\ D' = |X^*(t) - X(t)| \end{cases} \quad (3)$$

D' denotes the distance between the whale and the prey, k is the constant used to generate the spiral search, and l is a random number value [-1,1].

The optimization problem in this study involves a 6-dimensional problem. The population distribution is represented in 3-dimensional space as shown in Figure 3(a), which uses color, shape, and position to express high-dimensional space. To compare the algorithm improvements more intuitively, the t-SNE algorithm is used to reduce the dimensionality of the 6-dimensional problem to a 2-dimensional space expression, as shown in Figure 3(b).

The calculation of the convergence factor r introduces dynamic sine search, as shown in (4), and the corresponding convergence factor curve is shown in Figure 4. It can be seen that this strategy is easily affected by the constant k . The larger the value of k , the faster the convergence speed, causing the algorithm to fall into a local optimum, while the smaller the value of k , the slower the convergence speed of the algorithm. In this study, $k=0.8$ has a better result.

$$r = \frac{1}{2} + \frac{\sin\left(\frac{\pi}{2} + \pi\left(\frac{t}{T}\right)^k\right)}{2} \tag{4}$$

The original DBO algorithm involves sequential optimization by four types of dung beetles to find the global optimal solution while also avoiding getting stuck in local optima. This paper proposes a parallel computing approach that balances the search for both local and global optimal solutions with computational efficiency.

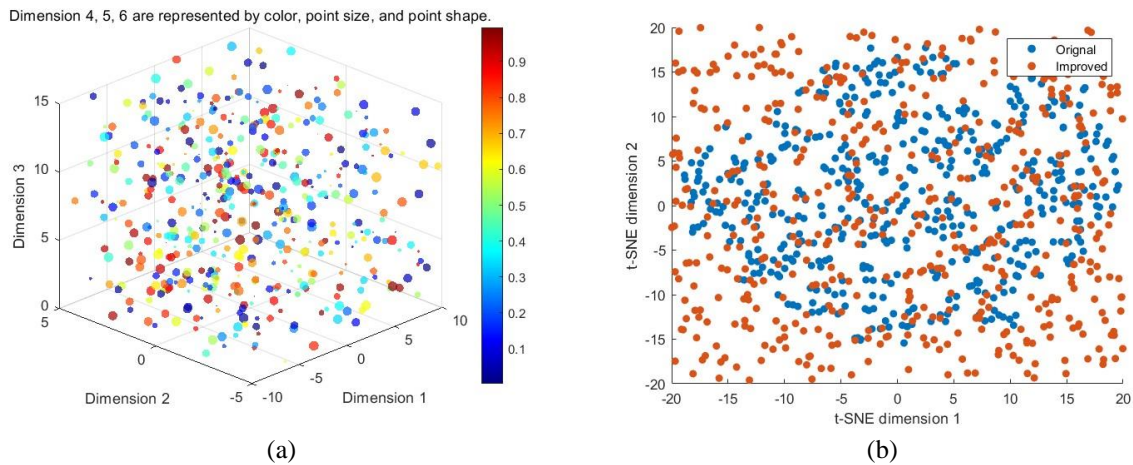


Figure 3. Population initialization comparison of (a) 6-dimensional space (color, size, shape) and (b) t-SNE dimensionality reduction distribution

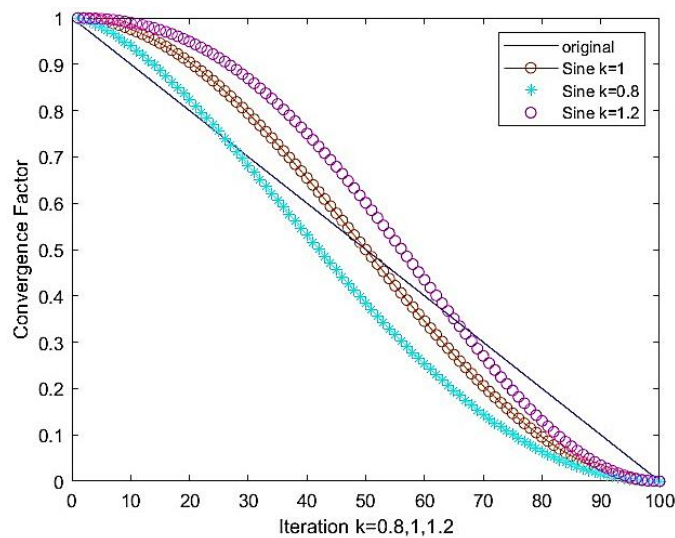


Figure 4. Convergence factor over iterations

2.4. Inverse kinematics optimization objective function

The target position error represents the difference between the desired end-effector position and the actual position obtained from the forward kinematics. To express this error in the objective function, it is defined as the Euclidean distance between the target position and the current position of the end effector. This can be represented as (5),

$$f_{pos} = \sqrt{\sum_{i=1}^m w_i \cdot (Pos_{target}^i - Pos_{end}^i)^2} \quad (5)$$

where m is the number of joints, Pos_{target} and Pos_{end} denote the desired and current positions of the end-effector respectively. The goal is to minimize this error by adjusting the joint angles of the robot. w_i is the weight for the i -th joint position error.

To construct the objective function for the end-effector orientation error, the difference between the desired and actual orientation of the end-effector can be calculated using the angle-axis representation. The objective function can be defined as the Euclidean distance between the two angles, as shown (6),

$$f_{angle} = \sqrt{(\theta_{dx} - \theta_x)^2 + (\theta_{dy} - \theta_y)^2 + (\theta_{dz} - \theta_z)^2} \quad (6)$$

where $\theta_x, \theta_y, \theta_z$ and $\theta_{dx}, \theta_{dy}, \theta_{dz}$ are the actual and desired orientation angles of the end-effector, respectively. Forward kinematics is used to obtain the current end-effector quaternion, and inverse kinematics is used to calculate the target end-effector quaternion. The dot product of these two quaternions represents their similarity and serves as the penalty factor in the orientation error objective function. To convert a rotation matrix as shown in (2) to a quaternion, use (7).

$$q: \begin{cases} w = \frac{1}{2} \sqrt{1 + r_{11} + r_{22} + r_{33}} \\ x = \frac{1}{2\sqrt{1+r_{11}+r_{22}+r_{33}}} \cdot (r_{32} - r_{23}) \\ y = \frac{1}{2\sqrt{1+r_{11}+r_{22}+r_{33}}} \cdot (r_{13} - r_{31}) \\ z = \frac{1}{2\sqrt{1+r_{11}+r_{22}+r_{33}}} \cdot (r_{21} - r_{12}) \end{cases} \quad (7)$$

Among (7), r_{ij} represents the element of row i and column j of the rotation matrix in (2). Specifically, the objective function is defined as the difference between the magnitude of the desired orientation quaternion and the actual orientation quaternion. The dot product of two quaternions can be calculated using (8). The closer the value is to 1, the more similar their rotation poses are.

$$f_{ori} = dot(q_1, q_2) = w_1 w_2 + x_1 x_2 + y_1 y_2 + z_1 z_2 \quad (8)$$

When constructing the objective function for position optimization, not only the error between the target value and the current calculated value, but also the previous position should be considered to ensure the smallest possible displacement. By considering the displacement of previous positions, the optimization process can be more stable and efficient, ultimately leading to better results. In the optimization calculation, the objective functions (9) to (11) are mainly tested.

$$f_{obj} = w_1 \cdot f_{pos} + w_2 \cdot f_{angle} \quad (9)$$

$$f_{obj} = w_1 \cdot f_{pos} + w_2 \cdot f_{angle} + w_3 \cdot (1 - f_{ori}) \quad (10)$$

$$f_{obj} = w_1 \cdot f_{pos} + w_2 \cdot f_{ori} \cdot f_{angle} \quad (11)$$

3. RESULTS AND DISCUSSION

This section compares the results obtained using different algorithms for solving robot inverse kinematics. Analyzes the performance of the improved DBO algorithm with various inverse kinematics fitness functions. The position and orientation error values implemented by the improved DBO algorithm

were evaluated, demonstrating its effectiveness in accurately determining the desired end-effector position and orientation.

3.1. Comparison of multiple algorithms

Current popular optimization algorithms will be applied to solve robot inverse kinematics, including the gray wolf optimization algorithm (GWO) [22], whale optimization algorithm (WOA) [23], antlion optimization algorithm (ALO) [24], and sparrow search algorithm (SSA) [25]. The relevant parameters of these several algorithms are given in Table 2. However, the focus will be on the application of the DBO algorithm and its improvements to solving inverse kinematics problems.

Table 2. Parameter values of algorithms

| Algorithm | Parameter | Value |
|-----------|------------------------------------|-----------------------|
| GWO | a_{min} | 0 |
| | a_{max} | 2 |
| WOA | a | Decreased from 2 to 0 |
| ALO | w | 2, 3, 4, 5, 6 |
| SSA | Leader position update probability | 0.5 |
| DBO | K and λ | 0.1 |
| | b | 0.3 |
| | s | 1 |

To better test the superiority of different algorithms, the number of search groups is set to 30 and the number of iterations is set to 500. The target position of the robot is [300, 100, 200], and the posture is [-180, 0, 0], which are the rotation angles around the X-axis, Y-axis, and Z-axis respectively. The target optimization function is (18), w_1 is 0.55, and w_2 is 0.45.

Table 3 shows the results of solving inverse kinematics using different swarm intelligence optimization algorithms such as DBO, GWO, WOA, ALO, and SSA. It includes the solved theta, the value of the objective optimization function at 100 iterations, the time required for 100 iterations, the final positional bias, and angular bias. It should be noted that these data are the average of the results calculated by running each algorithm at least 10 times, corresponding to the same objective optimization function for solving the inverse kinematics of the collaborative robot in the paper, and it is the same if they are all run on the computer. DBO, ALO, and SSA algorithms can accurately obtain the target position, and the DBO algorithm is the best.

Table 3. Results of DBO, GWO, WOA, ALO and SSA for inverse kinematics

| Algorithm | Theta(deg) | Function ₁₀₀ | Time ₁₀₀ (s) | Position _{error} (mm) | angle _{error} (deg) |
|-----------|--|-------------------------|-------------------------|--------------------------------|------------------------------|
| DBO | [19.40846, -9.899925, -19.76793, 17.06617, 29.53511, 0] | 4.3155 | 67.5977 | 2.1E-04 | 9.5901 |
| GWO | [-336.4719, -7.188253, -13.93191, -338.4281, -0.6719646, 359.3463] | 22.3812 | 71.2862 | 0.7048 | 21.8371 |
| WOA | [25.39506, -5.568372, -13.58481, -333.8359, -6.741634, -10.44632] | 38.2246 | 69.9685 | 3.4285 | 26.6872 |
| ALO | [26.36544, -7.022345, -15.48362, 36.62569, 1.674225, 350.8061] | 12.4436 | 92.3064 | 3.8E-04 | 21.4435 |
| SSA | [-331.0763, -5.265419, -15.71843, 43.69305, -4.473759, -17.25292] | 11.0840 | 143.756 | 3.5E-04 | 24.4887 |

3.2. Statistical analysis of improved DBO algorithm

Through the analysis of several popular algorithms in the previous section, the DBO algorithm has great advantages in calculating the inverse time of robot kinematics. Its calculations are more accurate and calculation times are faster. Therefore, this section improves the DBO algorithm, especially considering the problem of large angle error of the end effector. Next, calculations will be performed using (9) to (11) with different improved DBO algorithms. The parameter values of the improved DBO algorithm are shown in Table 4. The initial value of b is 0.2, and the initial value of S is 1. When the target value function is reduced to 20, reduce the small values of b and S .

Table 4. Parameter values of Improved DBO algorithm

| Algorithm | Parameter | Value | Function | Behavior |
|-----------|-----------|-------|----------|----------|
| a-DBO | N | 30 | (9) | I-DBO-1 |
| | K | 0.2 | (10) | I-DBO-2 |
| | | | (11) | I-DBO-3 |
| b-DBO | N | 30 | (9) | I-DBO-4 |
| | K | 0.1 | (10) | I-DBO-5 |
| | | | (11) | I-DBO-6 |
| c-DBO | N | 50 | (9) | I-DBO-7 |
| | K | 0.2 | (10) | I-DBO-8 |
| | | | (11) | I-DBO-9 |

Based on the improved DBO algorithm, 3 different parameters corresponding to *a-DBO*, *b-DBO*, and *c-DBO* are set, and different objective optimization functions (9) to (11) are set for each parameter, so there are 9 calculation results. Figure 5 shows the iterative curve of the improved DBO algorithm. It can be seen that the improved DBO algorithm has faster convergence in the calculation of robot inverse kinematics. According to the iterative calculation of the parameters in Table 4, the final fitness quickly converges to less than 10, and the error comparison with Table 3 has a better advantage.

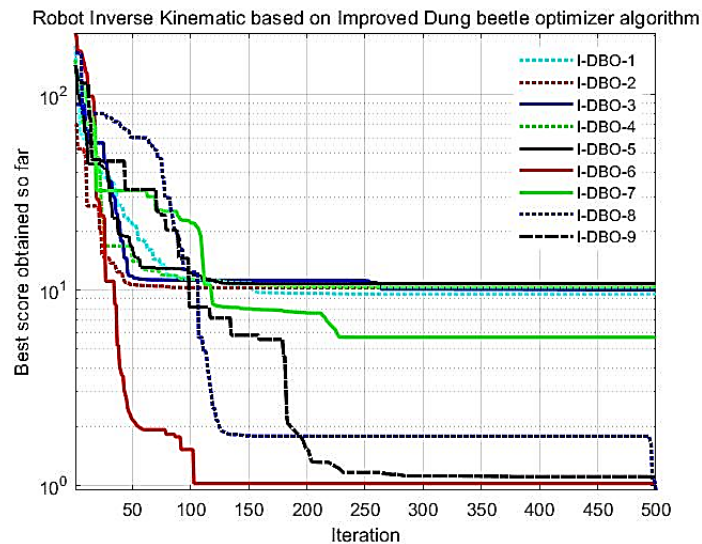


Figure 5. Iterative curves of improved DBO algorithm

For the *c-DBO*, increasing the number of populations can calculate good results just like the data of *I-DBO-9*, but the calculation time and the convergence speed are not dominant. In contrast, *I-DBO-6* achieves a good result (1.1023) with perfect convergence speed and computation time. The relevant parameters are $N=30$, $K=0.1$, and the selected objective optimization function is (11).

4. CONCLUSION

This study introduces the DBO algorithm for solving robot inverse kinematics problems. The performance of DBO is compared with other swarm intelligent algorithms (GWO, WOA, ALO, and SSA), demonstrating that the DBO algorithm achieves excellent results in terms of convergence speed and solution accuracy. Furthermore, a spiral search multi-strategy improved DBO algorithm is proposed that takes into account the robot's previous position, thereby improving pose estimation.

The position error obtained by the original DBO algorithm is $2.1E-04$, and the angle error is 9.5901, while the improved DBO algorithm can obtain more accurate results. For the improved DBO algorithm, various factors such as calculation time and pose accuracy are considered, and do not blindly pursue calculation accuracy. Using this improved DBO algorithm to solve the inverse kinematics of the robot, the solution position error is 0.1325, and the orientation error is 1.8718. Compared with the original DBO algorithm, it has obvious advantages and is worth adopting.

Results show that the DBO algorithm is a promising approach for solving robot inverse kinematics problems. The improved spiral search multi-strategy DBO algorithm provides a more accurate and efficient method for robot pose estimation. Overall, this study contributes to the development of efficient and practical solutions for robotic applications and opens the way for future research in this field.

ACKNOWLEDGEMENTS

This work was supported/funded by the Ministry of Higher Education under 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). We also want to thank the Tianshui Normal University scientific research project (No. CXJ2022-05), China for their continuous support in this research work.




REFERENCES

- [1] R. Ojstersek, B. Buchmeister, and A. Javernik, "The importance of cobot speed and acceleration on the manufacturing system efficiency," *Procedia Computer Science*, vol. 217, pp. 147–154, 2023, doi: 10.1016/j.procs.2022.12.210.
- [2] L. Roveda, P. Veerappan, M. Maccarini, G. Bucca, A. Ajoudani, and D. Piga, "A human-centric framework for robotic task learning and optimization," *Journal of Manufacturing Systems*, vol. 67, pp. 68–79, Apr. 2023, doi: 10.1016/j.jmsy.2023.01.003.
- [3] Z. Wang *et al.*, "A magnetic soft robot with multimodal sensing capability by multimaterial direct ink writing," *Additive Manufacturing*, vol. 61, p. 103320, Jan. 2023, doi: 10.1016/j.addma.2022.103320.
- [4] A. Taheri, "Impacts of socially assistive robots on improving the quality of life in children with autism," in *Encyclopedia of Child and Adolescent Health*, Elsevier, 2023, pp. 99–125. doi: 10.1016/b978-0-12-818872-9.00125-4.
- [5] J. Zhang, N. Zhang, L. Tian, Z. Zhou, and P. Wang, "Robots' picking efficiency and pickers' energy expenditure: the item storage assignment policy in robotic mobile fulfillment system," *Computers & Industrial Engineering*, vol. 176, p. 108918, Feb. 2023, doi: 10.1016/j.cie.2022.108918.
- [6] S. Poorghasem and Y. Bao, "Review of robot-based automated measurement of vibration for civil engineering structures," *Measurement*, vol. 207, p. 112382, Feb. 2023, doi: 10.1016/j.measurement.2022.112382.
- [7] W. Xu, Z. Mu, T. Liu, and B. Liang, "A modified modal method for solving the mission-oriented inverse kinematics of hyper-redundant space manipulators for on-orbit servicing," *Acta Astronautica*, vol. 139, pp. 54–66, Oct. 2017, doi: 10.1016/j.actaastro.2017.06.015.
- [8] I. Sancaktar, B. Tuna, and M. Ulutas, "Inverse kinematics application on medical robot using adapted PSO method," *Engineering Science and Technology, an International Journal*, vol. 21, no. 5, pp. 1006–1010, Oct. 2018, doi: 10.1016/j.jestech.2018.06.011.
- [9] B. Hu, T. Gao, J. Zhao, and Z. Liu, "One key issue in inverse kinematic modeling of lower mobility serial mechanisms," *Mechanism and Machine Theory*, vol. 177, p. 105066, Nov. 2022, doi: 10.1016/j.mechmachtheory.2022.105066.
- [10] G. Wang, S. Chen, Y. Guan, Z. Shi, X. Li, and J. Zhang, "Formalization of the inverse kinematics of three-fingered dexterous hand," *Journal of Logical and Algebraic Methods in Programming*, vol. 133, p. 100861, Jun. 2023, doi: 10.1016/j.jlamp.2023.100861.
- [11] F. Chen, H. Ju, and X. Liu, "Inverse kinematic formula for a new class of 6R robotic arms with simple constraints," *Mechanism and Machine Theory*, vol. 179, p. 105118, Jan. 2023, doi: 10.1016/j.mechmachtheory.2022.105118.
- [12] I. Chawla, P. M. Pathak, L. Notash, A. K. Samantaray, Q. Li, and U. K. Sharma, "Inverse and forward kineto-static solution of a large-scale cable-driven parallel robot using neural networks," *Mechanism and Machine Theory*, vol. 179, p. 105107, Jan. 2023, doi: 10.1016/j.mechmachtheory.2022.105107.
- [13] S. Qiu and M. R. Kermani, "Precision fingertip grasp: A human-inspired grasp planning and inverse kinematics approach for integrated arm-hand systems," *Robotics and Autonomous Systems*, vol. 162, p. 104348, Apr. 2023, doi: 10.1016/j.robot.2022.104348.
- [14] I. Pikalov, E. Spirin, M. Saramud, and M. Kubrikov, "Vector model for solving the inverse kinematics problem in the system of external adaptive control of robotic manipulators," *Mechanism and Machine Theory*, vol. 174, p. 104912, Aug. 2022, doi: 10.1016/j.mechmachtheory.2022.104912.
- [15] F. Chen and H. Ju, "Applications of an improved Dixon elimination method for the inverse kinematics of 6R manipulators," *Applied Mathematical Modelling*, vol. 107, pp. 764–781, Jul. 2022, doi: 10.1016/j.apm.2022.03.006.
- [16] G. DONG, P. HUANG, Y. WANG, and R. LI, "A modified forward and backward reaching inverse kinematics based incremental control for space manipulators," *Chinese Journal of Aeronautics*, vol. 35, no. 12, pp. 287–295, Dec. 2022, doi: 10.1016/j.cja.2021.08.014.
- [17] J. Xue and B. Shen, "Dung beetle optimizer: a new meta-heuristic algorithm for global optimization," *The Journal of Supercomputing*, vol. 79, no. 7, pp. 7305–7336, Nov. 2022, doi: 10.1007/s11227-022-04959-6.
- [18] T. Hu, H. Zhang, and J. Zhou, "Prediction of the debonding failure of beams strengthened with FRP through machine learning models," *Buildings*, vol. 13, no. 3, p. 608, Feb. 2023, doi: 10.3390/buildings13030608.
- [19] Z. Yin and L. Zinn-Björkman, "Simulating rolling paths and reorientation behavior of ball-rolling dung beetles," *Journal of Theoretical Biology*, vol. 486, p. 110106, Feb. 2020, doi: 10.1016/j.jtbi.2019.110106.
- [20] X. Wang *et al.*, "Model-based kinematic and non-kinematic calibration of a 7R 6-[robot with non-spherical wrist," *Mechanism and Machine Theory*, vol. 178, p. 105086, Dec. 2022, doi: 10.1016/j.mechmachtheory.2022.105086.
- [21] M. Caruso, L. Gastaldi, S. Pastorelli, A. Cereatti, and E. Digo, "An ISB-consistent upper limb model based on the Denavit-Hartenberg convention for joint angle estimation during prolonged rehabilitation exercises," *Gait & Posture*, vol. 97, p. 11, Oct. 2022, doi: 10.1016/j.gaitpost.2022.09.024.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, pp. 46–61, 2014, doi: 10.1016/j.advengsoft.2013.12.007.
- [23] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, May 2016, doi: 10.1016/j.advengsoft.2016.01.008.




- [24] S. Mirjalili, "The ant lion optimizer," *Advances in Engineering Software*, vol. 83, pp. 80–98, May 2015, doi: 10.1016/j.advengsoft.2015.01.010.
- [25] J. Xue and B. Shen, "A novel swarm intelligence optimization approach: sparrow search algorithm," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 22–34, Jan. 2020, doi: 10.1080/21642583.2019.1708830.

BIOGRAPHIES OF AUTHORS



Ma Haohao    was born in Tianshui, China. He received a bachelor's degree in mechanical design, manufacturing, and automation from North University of China, in 2011, and a master's degree in mechanical and electronic engineering from Xidian University, China, in 2014. He is pursuing a Ph.D. degree at Universiti Putra Malaysia. Currently, he is a Lecturer at Tianshui Normal University, China. His research interests include computer-aided design and manufacturing, robot control, and simulation. He can be contacted at email mahao1001@outlook.com.






Azizan As'arry    was born in Kuala Lumpur, Malaysia. He received bachelor's, master's, and Ph.D. degrees in mechanical engineering from Universiti Teknologi Malaysia, Malaysia, in 2007, 2009, and 2013, respectively. His major field of study is control and mechatronics. He is currently a senior lecturer at Universiti Putra Malaysia, Malaysia. His research interests include active force control, system identification, system modeling, and simulation. He can be contacted at email zizan@upm.edu.my.






Zhang Haoyang    was born in Tianshui, China. He received the Ph.D. degree in control theory and control engineering in 2022 from Xidian University, Xi'an, China, he is currently a lecturer at Lanzhou University of Technology. His research interests include machine learning and computer vision with applications in detection and tracking. He can be contacted at email zhanghy_xd@163.com.







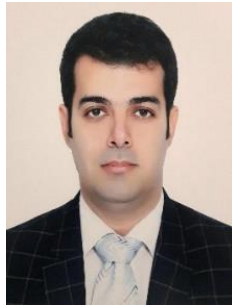
Mohd Idris Shah Ismail    was born in Johor Bahru, Malaysia. He is currently an Associate Professor at the Department of Mechanical and Manufacturing Engineering, Universiti Putra Malaysia. He received his Ph.D. degree in mechanical engineering from Okayama University, Japan, in 2012. His research interests include non-conventional machining, process modeling and simulation, and intelligent manufacturing. He can be contacted at email ms_idris@upm.edu.my.







Hafiz Rashidi Ramli    received a B.Eng. degree in electrical and electronic engineering and an M.Sc. degree in control and automation engineering from the Universiti Putra Malaysia (UPM), in 2007 and 2010, respectively, and a Ph.D. degree in biomedical engineering from Imperial College London, in 2017. His current research interests include artificial intelligence, image processing, haptics, and surgical simulation. He can be contacted at email hrhr@upm.edu.my.



Mohd Yusoff Moh Zuhri     is a senior lecturer at Universiti Putra Malaysia (UPM) and is currently the Head of the Laboratory of Biocomposite at the Institute of Tropical Forestry and Forest Products (INTROP), UPM. He received his B.Eng. degree in Mechanical Engineering (Design and Innovation) from Universiti Teknikal Malaysia Melaka, Malaysia in 2007. In 2009, he was awarded the Master of Science from Universiti Putra Malaysia in the field of composite materials. He was awarded his PhD from the University of Liverpool, UK in 2015. His main research interests are lightweight sandwich structures, composite materials (especially natural fiber composites), and building information modeling (BIM). He can be contacted at email zuhri@upm.edu.my.



Aidin Delgoshaei     is a senior lecturer at *Universiti Putra Malaysia* (UPM) and earned his PhD in Manufacturing Systems Engineering from UPM in 2016. His research interests include Modeling and Optimization, Heuristics and Meta-heuristics, Simulation, and Modeling of Complex Systems. Dr. Delgoshaei has successfully advised several Ph.D. and M.Sc. students in the industrial and manufacturing engineering fields, and has been involved with many international conference as a technical committee panel in the U.S, Canada, Spain, Italy, Turkey, Iran, Malaysia and Greece. Additionally, he serves as an editorial board member for international scientific journals. He can be contacted at email delgoshaei.aidin@upm.edu.my.