



The kinematics analysis and improvement of the 3T1R 4-PUU parallel mechanism using the whale optimization algorithm

Kai Wen¹, Xiaoning Zhou¹, Andong Jiang², and Jie Li³

¹Nanhang Jincheng College, Nanjing 211156, Jiangsu, China

²College of Mechanical and Electrical Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, Jiangsu, China

³Nanjing Zhongke Huaxing Emergency Technology Research Institute Co., Ltd., Nanjing 211106, Jiangsu, China

Correspondence: Xiaoning Zhou (zhouxiaoning2025@163.com)

Received: 19 March 2025 – Revised: 12 June 2025 – Accepted: 7 September 2025 – Published: 11 November 2025

Abstract. This paper focuses on the 4-PUU parallel mechanism with Schönflies motion, investigating its kinematic analysis and utilizing intelligent algorithms to solve its forward kinematics equations. Using screw theory as the mathematical framework, the number and nature of the degrees of freedom of the 4-PUU parallel mechanism are analyzed, proving that the mechanism can achieve 3T1R motion. The forward and inverse position kinematics equations are established with link lengths as constraints, and the velocity and acceleration equations of the mechanism are derived using the vector method. The forward kinematics equations of the mechanism are transformed into an unconstrained optimization problem, which is solved using an improved beluga whale optimization (BWO). To enhance the uniformity of the initial population distribution, a chaotic-opposition-based learning initialization strategy is introduced. Additionally, an elite strategy and a golden-sine-based position update mechanism are incorporated to improve the optimization capability of BWO. The integration of these strategies results in an enhanced optimization algorithm with superior global search ability. Given the mechanism's dimensional parameters, theoretical and motion simulations are conducted using MATLAB and Adams. The results indicate that the motion simulation curves from Adams closely match the numerical simulation curves from MATLAB, validating the theoretical derivation. Building on this, the improved beluga whale optimization (IBWO) algorithm, along with comparison algorithms (DEb1, ABC, PSO), is applied to solve the forward kinematics equations for interpolation points along the simulated trajectory. Numerical experiments demonstrate that IBWO outperforms the other comparison algorithms in solving the problem efficiently and accurately.

1 Introduction

Compared to 6-degree-of-freedom (DOF) parallel mechanisms (PMs), those with fewer DOFs offer advantages such as simpler structure, fewer drives, and easier control and have gradually gained widespread attention from both academia and industry. Among them, the 3T1R (T representing translation and R representing revolute) PM is widely used in practical engineering applications such as packaging and sorting (Zhu et al., 2022), welding (Wen et al., 2024), assembly (Li et al., 2021), and rehabilitation (Shi et al., 2024).

The original SCARA – with 3T1R DOF (where T and R represent translational and rotational degrees of freedom, respectively) – is an extension of the 3-DOF Delta robot. An additional UPU chain was incorporated between the moving and fixed platforms to provide rotation about the z axis. The kinematic analysis, performance analysis, error compensation, and trajectory planning of the PM are all based on the integration of its configuration. Therefore, it is evident that the configuration synthesis of the SCARA of PM forms the foundation for subsequent theoretical analysis and structural design.

The configuration synthesis of SCARA PMs has shown significant research achievements. Company et al. (2003) designed a structure with both symmetric and asymmetric H4 PM based on the motion used to generate Schönflies displacement subgroups. Yang et al. (2017) proposed a configuration synthesis method for a 3T1R PM with a variable rotation axis based on screw theory. Salgado et al. (2008) used the theory of groups of displacements, based on the H4, I4 (Krut et al., 2003), and Par4 (Nabat et al., 2005) mechanisms. Li et al. (2014) replaced the original parallelogram chains with R(SS)2 chains, introducing the Cross IV series PMs, which were later industrialized. Staicu et al. (2018) proposed a fully symmetric X4 PM where each of the four chains is composed of R(SS)R, resulting in improved dynamic characteristics and simplified manufacturing. Wen et al. (2024) designed a 2PUU-2PSS PM with an axially symmetric structure based on screw theory. Xie et al. (2013), using Grassmann line geometry and the Atlas method, designed a class of PMs with 3T1R (1T3R, 2T2R, and 3T1R) motion.

Kinematic analysis of PMs forms the foundation for error compensation, trajectory planning, and workspace analysis. It primarily consists of forward kinematics analysis and inverse kinematics analysis. For coupled PMs, the inverse kinematics analysis typically involves decoupling the equations, while the forward kinematics usually leads to a coupled nonlinear system of equations. The solution to the forward kinematics equations of coupled PMs is mainly divided into analytical method and numerical method. Analytical methods involve eliminating unknowns and reducing higher-order equations to first-order equations. By solving the first-order equations and performing inverse substitution of the results, a high-precision solution to the forward kinematics can be obtained. However, this method is specific to certain configurations and lacks generality. Numerical methods, on the other hand, include numerical iteration, machine learning, and intelligent algorithms, which can be applied to a wider range of mechanisms and offer more flexibility in handling complex kinematic problems.

Wen et al. (2024) proposed a mean-based dual-strategy differential evolution algorithm for solving the forward kinematics equations of the 2PUU-2PSS PM. It was demonstrated that the performance of this algorithm surpasses that of comparison algorithms such as IABC and DEr1 through numerical examples. Liu et al. (2019) combined genetic algorithms and backpropagation (BP) neural networks to solve the forward kinematics equations of the Delta robot, with results showing better performance than traditional BP neural networks. Wu et al. (2024) introduced an improved backpropagation neural network for solving the forward kinematics equations of redundant PMs, and experimental results indicated a significant improvement in solution accuracy. Wen et al. (2023) proposed a time-varying differential evolution algorithm for solving the multiple solution problem of forward kinematics equations in PMs. Using PMs with 3 to 6

DOF as examples, the feasibility and versatility of the algorithm were verified.

This paper is based on the configuration synthesis of PMs and utilizes screw theory to design a 4-PUU PM with a fully symmetric structure, which the workspace is unaffected by the vertical direction. The structure and kinematic analysis of the 4-PUU PM are presented in the following chapters: Sect. 2 provides a structural description of the 4-PUU PM and analyzes its degrees of freedom using screw theory. Section 3 derives the kinematic equations of the mechanism using vector and differentiation methods. Section 4 introduces an improved whale optimization algorithm (IBWO) to solve the forward kinematics equations of the 4-PUU PM. Section 5 verifies the correctness of the kinematic formula derivations through motion simulations using Adams software and validates the performance of the IBWO algorithm through numerical examples. Section 6 concludes the paper and outlines future research directions.

2 4-PUU structural description and DOF analysis of PMs

2.1 Structural description

The 4-PUU PM is shown in Fig. 1. This mechanism mainly consists of a moving platform, a fixed platform, and four identical structural branches. Each branch is composed of a prismatic pair (P) and universal pair (U), which are rotational pairs with axes perpendicular to each other. The topology of the branch can be represented as $SCO\{-P_1 \parallel R_2 \perp R_3 \parallel R_4 \perp R_5\}$. That is, starting from the fixed platform, the axis of the moving pair 1 is parallel to the axis of the rotating pair 2, the axis of the rotating pair 2 is perpendicular to the axis of the rotating pair 3, the axis of the rotating pair 3 is parallel to the axis of the rotating pair 4, the axis of the rotating pair 4 is perpendicular to the axis of the rotating pair 5, and the axis of the rotating pair 5 is perpendicular to the moving platform. In the PUU chain, the intersection of the axis of the pair (P) with the fixed platform is denoted as $A_i (i = 1, 2, 3, 4)$. The center of the first universal joint intersects the axis of the pair at point B_i , and the center of the second universal joint intersects the moving platform at point C_i . To avoid the initial singularity of the mechanism, the fixed platform is set as a square with the points A_i lying on a circle with radius $\sqrt{2}R$. The moving platform is set as a rectangle with side lengths of $2a$ and $2b$.

- A fixed coordinate system $\{f\}$: OXYZ is established on the fixed platform, where O is the geometric center of the fixed platform. The OX axis is perpendicular to A_1A_4 and passes through its midpoint, the OY axis is perpendicular to A_1A_2 and passes through its midpoint, and the OZ axis is perpendicular to the fixed platform and points vertically upwards.

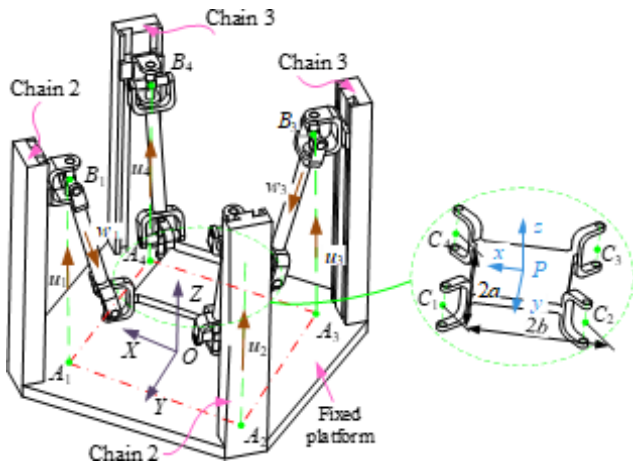


Figure 1. Three-dimensional diagram of the 4-PUU PM.

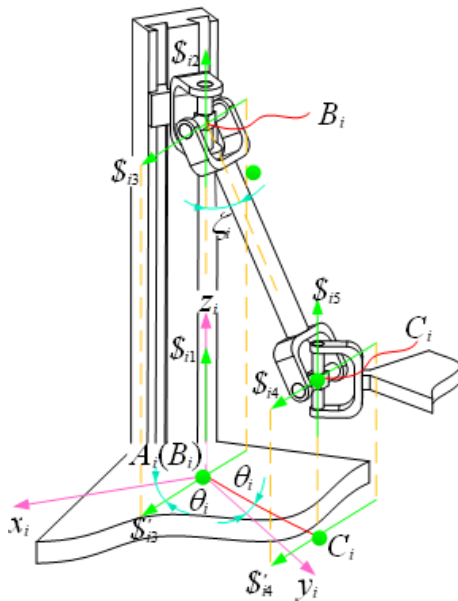


Figure 2. Schematic diagram of PUU branch chain structure.

- A moving coordinate system $\{m\}$: $Pxyz$ is established on the moving platform, where P is the geometric center of the moving platform. The Px axis is perpendicular to C_1C_4 and passes through its midpoint, the Py axis is perpendicular to C_1C_2 and passes through its midpoint, and the Pz axis is perpendicular to the moving platform and points vertically upwards.

2.2 Analysis of DOF

The branch coordinate system is established as shown in Fig. 2: $A_i-x_i y_i z_i$, where the z_i axis coincides with $A_i B_i$ and points toward B_i , the y_i axis coincides with $A_i O$ and points toward O , and the x_i axis is determined by the right-hand rule. The angle between the revolute pair R3 and the x_i axis

is θ_i , and the angle between the link $B_i C_i$ and the axis of the prismatic joint P is ζ_i . Thus, in the branch coordinate system, the motion screw of the PUU branch can be expressed as

$$\begin{cases} \$_{i1} = (0, 0, 0; 0, 0, 1) \\ \$_{i2} = (0, 0, 1; 0, 0, 0) \\ \$_{i3} = (c\theta_i, s\theta_i, 0; h_i s\theta_i, -h_i c\theta_i, 0) \\ \$_{i4} = (c\theta_i, s\theta_i, 0; (ls\zeta_i - h_i)s\theta_i, \\ \quad (h_i - ls\zeta_i)c\theta_i, -lc\zeta_i) \\ \$_{i5} = (0, 0, 1; -lc\delta_i c\theta_i, -lc\delta_i s\theta_i, 0) \end{cases}, \quad (1)$$

where $\cos\theta_i = c\theta_i$, $\sin\theta_i = s\theta_i$, $\cos\zeta_i = c\zeta_i$, and $\sin\zeta_i = s\zeta_i$.

Therefore, the constraint screw of the PUU chain can be expressed as

$${}^r\$_i = (0, 0, 0; s\theta_i, -c\theta_i, 0). \quad (2)$$

Equation (2) represents a reciprocal screw pair, which indicates that the branch constrains the moving platform's rotation around the direction parallel to $(\sin\theta_i, -\cos\theta_i, 0)$. The constraints imposed by the four symmetrically distributed PUU branches on the moving platform can be expressed using the two reciprocal screws ${}^r\$_{i1}$ and ${}^r\$_{i3}$. Clearly, these two reciprocal screws are linearly dependent. Therefore, a linear combination of the following two reciprocal screws can be used to obtain

$$\begin{cases} {}^r\$_1 = (0, 0, 0; 1, 0, 0) \\ {}^r\$_3 = (0, 0, 0; 0, 1, 0) \end{cases}. \quad (3)$$

According to Eq. (3), the constraint screws applied to the moving platform by the four branches are

$$\begin{cases} \$_{CWS,1} = (0, 0, 0; 1, 0, 0) \\ \$_{CWS,2} = (0, 0, 0; 0, 1, 0) \end{cases}. \quad (4)$$

$\$_{CWS,1}$ and $\$_{CWS,2}$ represent the constraint wrenches around the X axis and Y axis, respectively, applying rotational constraints around the X and Y axes to the moving platform, thus causing the platform to lose its rotational freedom around the X and Y axes. Therefore, the 4-PUU PM proposed in this paper has 4 DOF(3TIR_Z). In general configurations (except for singular poses), the motion pairs of each branch maintain their original geometric relationships due to structural constraints. The coordinate relations shown in Fig. 2 can still be established, and the resulting motion screws and the computed reciprocal screws remain unchanged. The constraint wrenches of the four reciprocal screw systems remain in the same plane, and their linear independence and rank remain constant. That is, the virtual constraints of the mechanism remain unchanged. Thus, the DOFs correspond to full rotational freedom.

3 4-PUU PM kinematic analysis

3.1 Forward and inverse kinematics analysis

3.1.1 Forward kinematics

The forward kinematics solution of the mechanism is to calculate the output position $P = (X_P, Y_P, Z_P)$ and the attitude angle γ of the moving platform's reference point P , given the scale parameters of the mechanism and the input link lengths h_i (for $i = 1, \dots, 4$) of the prismatic joints. The structural parameters of the platform are set as $\sigma_i = (2i - 1)\pi/4$, and the coordinates of point A_i in the frame $\{f\}$ are defined as

$$\{f\}\mathbf{A}_i = R(c\sigma_i, s\sigma_i, 0)^T. \quad (5)$$

The coordinate vector of point C_i in the frame $\{m\}$ is

$$\{m\}\mathbf{C}_i = (\text{sign}(c\sigma_i)a, \text{sign}(s\sigma_i)b, 0)^T. \quad (6)$$

The unit vector of the axis of the driving arm $A_i B_i$ in the frame $\{f\}$ is

$$\{f\}\mathbf{u}_i = (0, 0, 1)^T. \quad (7)$$

Assume the coordinate vector of the reference point $\mathbf{P}$ on the moving platform in the fixed frame $\{f\}$ is

$$\{f\}\mathbf{P} = (X_P, Y_P, Z_P)^T. \quad (8)$$

The attitude matrix of the moving platform is

$$\mathbf{M} = \begin{pmatrix} c\gamma & -s\gamma & 0 \\ s\gamma & c\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Then, the coordinate vector of point C_i ($i = 1, \dots, 4$) in the fixed coordinate system $\{f\}$ is

$$\{f\}\mathbf{C}_i = \{f\}\mathbf{P} + \mathbf{M}\{m\}\mathbf{C}_i = \{f\}\mathbf{A}_i + h_i \{f\}\mathbf{u}_i + l \{f\}\boldsymbol{\omega}_i. \quad (9)$$

The unit vector of the axis of the moving arm $B_i C_i$ in the fixed frame $\{f\}$ is

$$\boldsymbol{\omega}_i = \frac{\{f\}\mathbf{P} + \mathbf{M}\{m\}\mathbf{C}_i - \{f\}\mathbf{A}_i - h_i \{f\}\mathbf{u}_i}{l}. \quad (10)$$

The constraint equation of the link $B_i C_i$ is

$$\begin{aligned} & (\{f\}\mathbf{P} + \mathbf{M}\{m\}\mathbf{C}_i - \{f\}\mathbf{A}_i - h_i \{f\}\mathbf{u}_i)^T (\{f\}\mathbf{P} + \mathbf{M}\{m\}\mathbf{C}_i \\ & - \{f\}\mathbf{A}_i - h_i \{f\}\mathbf{u}_i) = l^2. \end{aligned} \quad (11)$$

According to Eq. (11), we can obtain

$$\begin{cases} (R - ac\gamma - bs\gamma - X_P)^2 + (R + ac\gamma - bs\gamma - Y_P)^2 + (h_1 - Z_P)^2 = l^2 \\ (-R + ac\gamma - bs\gamma - X_P)^2 + (-R - ac\gamma + bs\gamma - Y_P)^2 + (h_2 - Z_P)^2 = l^2 \\ (-R + ac\gamma + bs\gamma - X_P)^2 + (-R - ac\gamma + bs\gamma - Y_P)^2 + (h_3 - Z_P)^2 = l^2 \\ (R - ac\gamma + bs\gamma - X_P)^2 + (R + ac\gamma + bs\gamma - Y_P)^2 + (h_4 - Z_P)^2 = l^2 \end{cases}. \quad (12)$$

Equation (12) is a system of coupled nonlinear equations, which can be solved using both numerical methods and analytical methods. The analytical method involves substitution and elimination to obtain a single high-order equation. By solving this high-order equation and performing backward substitution, a high-precision forward kinematics solution for the mechanism can be obtained. However, this method has limited applicability and is only suitable for special configurations of PMs. The numerical method, on the other hand, solves the equations using intelligent algorithms or numerical iteration, offering advantages such as simplicity and efficiency. Therefore, in this paper, a numerical method is employed to solve the forward kinematics equations, as detailed in Sect. 4.

3.1.2 Inverse kinematics

The inverse kinematics analysis of the mechanism involves the process of solving for the input h_i given the output position and orientation of the mechanism (X_P, Y_P, Z_P, γ) . Thus, it can be deduced that, according to Eq. (12), we obtain

$$h_i = Z_P \pm \sqrt{l^2 - \lambda_{i1}^2 - \lambda_{i2}^2}. \quad (13)$$

In the equation, λ_{ij} is a coefficient ($i = 1, 2, 3, 4; j = 1, 2$), and its expression is

$$\begin{cases} \lambda_{i1} = X_P + \text{sign}(\cos \xi_i) a \cos \gamma - \text{sign}(\sin \xi_i) b \sin \gamma - R \cos \xi_i \\ \lambda_{i2} = Y_P + \text{sign}(\cos \xi_i) a \sin \gamma - \text{sign}(\sin \xi_i) b \cos \gamma - R \sin \xi_i \\ l^2 - \lambda_{i1}^2 - \lambda_{i2}^2 = 0 \end{cases}. \quad (14)$$

3.2 Kinematic velocity analysis

Let the inputs of the four driving joints of the mechanism be $\mathbf{h} = (h_1, h_2, h_3, h_4)^T$, the output quantities of the moving platform be $\mathbf{X} = (X_P, Y_P, Z_P, \gamma)^T$, the input velocities of the four driving joints be $\mathbf{v} = (\dot{h}_1, \dot{h}_2, \dot{h}_3, \dot{h}_4)^T$, the output velocities of the moving platform be $\dot{\mathbf{X}} = (\dot{X}_P, \dot{Y}_P, \dot{Z}_P)^T$, and the angular velocity output vector of the moving platform be $\dot{\gamma} = (0, 0, \dot{\gamma})^T$.

Taking the derivative of Eq. (11) with respect to time t , we get

$$[\dot{\mathbf{P}} + \dot{\gamma} \times (\mathbf{M}\mathbf{C}_i) - \dot{h}_i \mathbf{u}]^T \boldsymbol{\omega}_i = 0 \quad i = 1, 2, 3, 4. \quad (15)$$

By simplifying this, we obtain

$$\boldsymbol{\omega}_i^T \dot{\mathbf{P}} + [(\mathbf{M}\mathbf{C}_i) \times \boldsymbol{\omega}_i]^T \dot{\gamma} = \dot{h}_i \boldsymbol{\omega}_i^T \mathbf{u}. \quad (16)$$

Let $k_i = \boldsymbol{\omega}_i^T \mathbf{u}$ and take $\mathbf{J}_{in} = \text{diag}(k_1, k_2, k_3, k_4)$:

$$\mathbf{J}_{out} = \begin{pmatrix} \omega_1^T & [(\mathbf{M}\mathbf{C}_1) \times \boldsymbol{\omega}_1]^T \mathbf{u} \\ \omega_2^T & [(\mathbf{M}\mathbf{C}_2) \times \boldsymbol{\omega}_2]^T \mathbf{u} \\ \omega_3^T & [(\mathbf{M}\mathbf{C}_3) \times \boldsymbol{\omega}_3]^T \mathbf{u} \\ \omega_4^T & [(\mathbf{M}\mathbf{C}_4) \times \boldsymbol{\omega}_4]^T \mathbf{u} \end{pmatrix}.$$

It can be seen that Eq. (16) can be rewritten as

$$\mathbf{J}_{\text{in}} \dot{\mathbf{h}} = \mathbf{J}_{\text{out}} \dot{\mathbf{X}}. \quad (17)$$

If $\mathbf{J}_{\text{out}}$ is non-singular, then

$$\dot{\mathbf{X}} = \mathbf{J}_{\text{out}}^{-1} \mathbf{J}_{\text{in}} \dot{\mathbf{h}}. \quad (18)$$

If $\mathbf{J}_{\text{in}}$ is non-singular, then

$$\dot{\mathbf{h}} = \mathbf{J}_{\text{in}}^{-1} \mathbf{J}_{\text{out}} \dot{\mathbf{X}}. \quad (19)$$

Equations (22) and (23) are the forward and inverse velocity solutions of the 4-PUU PM, respectively.

3.3 Acceleration analysis

Let the input accelerations of the mechanism's driving joints be $\ddot{\mathbf{h}} = (\ddot{h}_1, \ddot{h}_2, \ddot{h}_3, \ddot{h}_4)^T$ and the output acceleration of the moving platform be $\ddot{\mathbf{X}} = (\ddot{X}_P, \ddot{Y}_P, \ddot{Z}_P)^T$, with the output angular velocity being $\dot{\gamma} = (0, 0, \dot{\gamma})^T$. Taking the derivative of equation with respect to time t , we get

$$\begin{aligned} & \{\ddot{\mathbf{P}} + \dot{\gamma} \times (\mathbf{M}\mathbf{c}_i) + \dot{\gamma} \times [\dot{\gamma} \times (\mathbf{M}\mathbf{c}_i) - \dot{h}_i \mathbf{u}]\}^T \omega_i \\ & + \frac{1}{l} [\dot{\mathbf{P}} + \dot{\gamma} \times (\mathbf{M}\mathbf{c}_i) - \dot{h}_i \mathbf{u}]^T [\dot{\mathbf{P}} + \dot{\gamma} \times (\mathbf{M}\mathbf{c}_i) \\ & - \dot{h}_i \mathbf{u}]^T = 0. \end{aligned} \quad (20)$$

If we denote $\kappa_i = (\omega_i \times \dot{\gamma})^T [\dot{\gamma} \times (\mathbf{M}\mathbf{c}_i)] + \frac{1}{l} \|\dot{\mathbf{P}} + \dot{\gamma} \times (\mathbf{M}\mathbf{c}_i) - \dot{h}_i \mathbf{u}\|$, then we have $\kappa = (\kappa_1, \kappa_2, \kappa_3, \kappa_4)^T$.

It can be seen that Eq. (20) can be simplified to

$$\mathbf{J}_{\text{in}} \ddot{\mathbf{h}} = \mathbf{J}_{\text{out}} \ddot{\mathbf{X}} + \kappa. \quad (21)$$

If $\mathbf{J}_{\text{out}}$ is non-singular, then

$$\ddot{\mathbf{X}} = \mathbf{J}_{\text{out}}^{-1} \mathbf{J}_{\text{in}} \ddot{\mathbf{h}} + \mathbf{J}_{\text{out}}^{-1} \kappa. \quad (22)$$

If $\mathbf{J}_{\text{in}}$ is non-singular, then

$$\ddot{\mathbf{h}} = \mathbf{J}_{\text{in}}^{-1} \mathbf{J}_{\text{out}} \ddot{\mathbf{X}} + \mathbf{J}_{\text{in}}^{-1} \kappa. \quad (23)$$

Equations (22) and (23) are the forward and inverse acceleration solutions of the 4-PUU PM, respectively.

4 Beluga whale optimization for forward kinematics analysis

4.1 Mathematical modeling and beluga whale optimization (BWO)

4.1.1 Mathematical modeling

When solving the forward kinematics (Eq. 12) of the PM using a swarm intelligence algorithm, it is necessary to first convert it into an unconstrained optimization problem and

then use the swarm intelligence algorithm to iteratively optimize the unconstrained optimization equation. Thus, we define $\mathbf{x} = (x_1, x_2, x_3, x_4)^T = (X_P, Y_P, Z_P, \gamma)^T$, and Eq. (12) can be transformed into

$$\begin{cases} f_1(\mathbf{x}) = (R - ac\gamma - bs\gamma - X_P)^2 + (R + ac\gamma - bs\gamma - Y_P)^2 \\ \quad + (h_1 - Z_P)^2 - l^2 \\ f_2(\mathbf{x}) = (-R + ac\gamma - bs\gamma - X_P)^2 + (R - ac\gamma - bs\gamma - Y_P)^2 \\ \quad + (h_2 - Z_P)^2 - l^2 \\ f_3(\mathbf{x}) = (-R + ac\gamma + bs\gamma - X_P)^2 + (-R - ac\gamma + bs\gamma - Y_P)^2 \\ \quad + (h_3 - Z_P)^2 - l^2 \\ f_4(\mathbf{x}) = (R - ac\gamma + bs\gamma - X_P)^2 + (-R + ac\gamma + bs\gamma - Y_P)^2 \\ \quad + (h_4 - Z_P)^2 - l^2 \end{cases} \quad (24)$$

The nonlinear system of equations in Eq. (24) can be transformed into an unconstrained optimization problem as follows:

$$\min F(\mathbf{x}) = \sum_{i=1}^4 f_i(\mathbf{x}).$$

4.1.2 Beluga whale optimization (BWO)

The BWO (Zhong et al., 2022) is a bio-inspired optimization algorithm that simulates the hunting behavior of the beluga whale. The BWO algorithm has characteristics such as a simple structure, easy expandability, few control parameters, and strong robustness. The main process of this algorithm includes the following steps: population initialization, encircling prey, bubble-net attacking, and searching for prey.

1. Population initialization

The initialization of the population in the BWO is similar to other optimization algorithms. It involves generating an initial population of N_P individuals (where N_P is the population size) randomly within the search space of the optimization variables. Specifically, the process is as follows:

$$\begin{aligned} \mathbf{x}_{n,d}^{(0)} &= \mathbf{x}_{\min,d} + \text{rand}()(\mathbf{x}_{\max,d} - \mathbf{x}_{\min,d}) \\ n &= 1, 2, \dots, N_P. \end{aligned} \quad (25)$$

In the equation, $x_{\min,d}$ and $x_{\max,d}$ represent the lower and upper bounds of the d th variable, respectively. Here, $d = 1, 2, \dots, D$, where D is the total number of decision variables (the dimensionality of the search space).

2. Encircling prey

In the BWO, since the optimal solution in the search space is not known in advance, it is assumed that the best solution obtained so far is the target prey or an approximation of the global optimum. Once the optimal solution's position is determined, the other whale individuals in the population move towards this optimal position. The position update for each whale is then given

by the following equation:

$$\tau = |C \cdot \mathbf{x}^{*(t)} - \mathbf{x}^{(t)}| \quad (26)$$

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{*(t)} - A \cdot \tau. \quad (27)$$

In the equation, t represents the current iteration number, $\mathbf{x}^{(t)}$ denotes the position of the whale population at the t th iteration, $\mathbf{x}^{*(t)}$ is the position of the best solution (prey) in the whale population at iteration t , $A \cdot \tau$ represents the update step size for the whale population, and A and τ are the update coefficients. Their specific expressions are as follows:

$$\begin{cases} A = 2ar_1 - a \\ C = 2r_2 \end{cases}. \quad (28)$$

In the equation, r_1 and r_2 are random numbers between $[0,1]$; this convergence factor is used to balance the global search ability and local search ability. It is given by the following formula: $a = 2 - 2t/T$, where T is the maximum number of iterations.

3. Bubble net attack

In the BWO, the bubble net attack is a crucial phase that combines two different movement strategies: shrink encircling and spiral updating. These two strategies allow the whales to explore the solution space globally and locally, respectively. The choice of which strategy to use is determined by a probability p , which controls the switching between the two methods. The specific operations are as follows:

$$\mathbf{x}^{(t+1)} = \begin{cases} \mathbf{x}^{*(t)} - A \cdot \tau & p < 0.5 \\ \tau' \cdot e^{bl} \cos(2\pi l) + \mathbf{x}^{*(t)} & p \geq 0.5 \end{cases}. \quad (29)$$

In the equation, b is the spiral shape parameter; τ' is the Euclidean distance between the current global best solution and the current whale's position; l is a random number uniformly distributed between $[-1,1]$, used to introduce randomness into the movement; and p is a random number uniformly distributed between $[0,1]$, which controls the probability of selecting a specific update method (shrink encircling or spiral updating).

4. Search for prey

The BWO updates the positions of the whales based on the distance between the individual whales, thereby performing a random search. When $|A| \geq 1$, the BWO algorithm randomly selects a whale position as the target whale $\mathbf{x}_{\text{rand}}^{(t)}$, and the other whales update their positions relative to this target in search of a better prey. The updated formula is as follows:

$$\mathbf{x}^{(t+1)} = \mathbf{x}_{\text{rand}}^{(t)} - A \cdot \tau''. \quad (30)$$

In the equation, τ'' represents the step size coefficient, $\tau'' = |C \cdot \mathbf{x}_{\text{rand}}^{(t)} - \mathbf{x}^{(t)}|$.

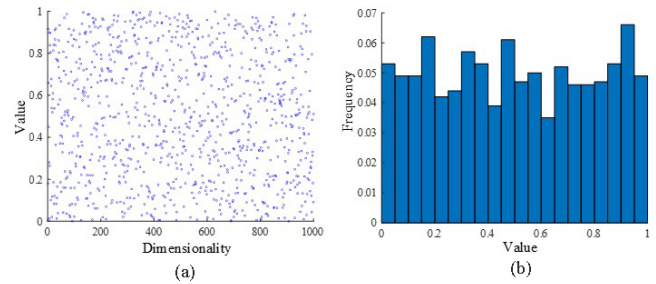


Figure 3. Distribution diagram of rand population initialization methods showing (a) value and (b) frequency.

4.2 Improved beluga whale optimization (IBWO)

4.2.1 Chaos-inspired backward learning population initialization

The initial population of the basic differential evolution algorithm is randomly generated using Eq. (25), which can lead to an uneven distribution and clustering of the generated initial population. The use of chaotic mapping can effectively address the issue of population initialization. Commonly used chaotic maps include the Kent map (Xiang et al., 2007), Circle map (Arora et al., 2019), and Logistic map (Herbadji et al., 2019). According to the reference (Sankalap et al., 2019), the Circle map generates a more evenly distributed initial population compared to other chaotic maps. However, the initial population generated by the Circle map still tends to be denser in the range $[0.2, 0.6]$ compared to other values. Therefore, in this paper, the Circle map is improved to achieve a relatively uniform distribution across the entire solution space. The improved Circle map formula is as follows:

$$\mathbf{k}_{n,D} = \text{mod} \left(2\mathbf{k}_{n-1,D} + 0.35 - \frac{0.7}{2\pi} \sin(2\pi \mathbf{k}_{n-1,D}), 1 \right). \quad (31)$$

In the equation, $n = 1, 2, \dots, N_P$; when $n = 2$, $\mathbf{k}_{n-1,D} = \text{rand}(1,D)$. Combining with Eq. (25), the initial population is generated using the following method:

$$\mathbf{x}_n = \mathbf{x}_{\min} + (\mathbf{x}_{\max} - \mathbf{x}_{\min})\mathbf{k}_n. \quad (32)$$

As shown in Figs. 3–5 to clearly and intuitively demonstrate the advantages of the improved Circle map, the initial population distributions for random initialization, Circle map, and improved Circle map are plotted for $n = 1000$, along with a frequency statistical analysis.

As shown in Fig. 3, the improved Circle chaotic map results in a more evenly distributed chaotic value. Therefore, the improved Circle chaotic map is used to generate the initial population, which enhances the population diversity and, in turn, improves the optimization ability of the algorithm.

Backward learning can improve the quality and diversity of the population solutions. However, the distance between the solutions generated by backward learning and the current

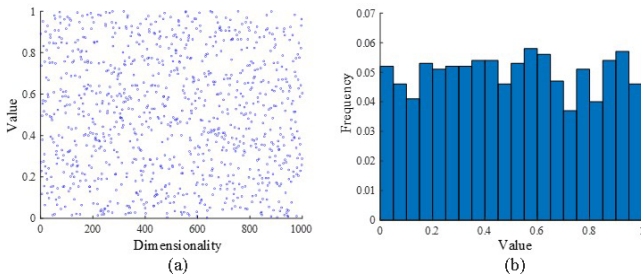


Figure 4. Distribution diagram of circle map population initialization methods showing (a) value and (b) frequency.

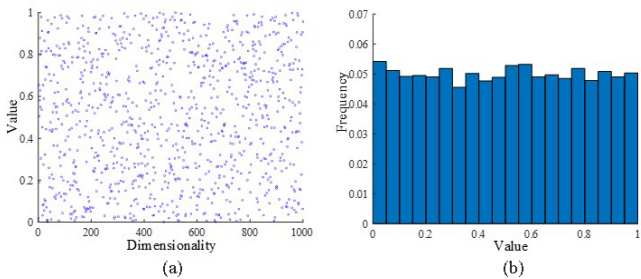


Figure 5. Distribution diagram of improved circle map population initialization methods showing (a) value and (b) frequency.

solution is fixed, which is detrimental to population diversity and lacks a certain degree of randomness. Therefore, based on the improved Circle map, a dynamic backward learning strategy is employed to further increase population diversity and the number of elite individuals, thereby expanding the search range of the algorithm. The dynamic backward learning strategy is as follows:

$$x_n = x_n + \text{rand}(\text{rand}(x_{\max} + x_{\min} - x_n) - x_n). \quad (33)$$

Combining Eqs. (32) and (34), for the minimization optimization problem, the chaos-backward learning population initialization can be expressed as

$$x_n^{(0)} = \begin{cases} x_n & f(x_n) < f(x_n) \\ x_n & \text{else} \end{cases}. \quad (34)$$

4.2.2 Elite strategy

According to Eq. (29), the BWO algorithm updates positions based on the current optimal position of the whale. If the current optimal whale position becomes trapped in a local optimum, all the whales will converge around the local optimal individual, causing the algorithm to stagnate. In the gray wolf optimization algorithm (GWO), to increase population diversity, a hierarchical system is proposed. This involves applying weighted processing to the three individuals with the best fitness values in the current population, using them as the current optimal position. This approach helps to avoid the risk of the population getting trapped in a local optimum.

Table 1. Inverse kinematics of the 4-PUU PM.

Num.	Driving rod length			
	l_{11}/mm	l_{21}/mm	l_{31}/mm	l_{41}/mm
1	22.0013	49.0150	59.4375	36.7018
2	14.9987	-9.0150	-19.4375	3.2982

Based on the hierarchical system of the GWO algorithm (Ghalambaz et al., 2021), this paper introduces the concept of an elite pool. Specifically, the three best individuals in the population, along with the weighted individuals of high quality, are stored in the elite pool. During the position update process, an individual is randomly selected from the elite pool to serve as a guide, enhancing the algorithm's ability to escape local optima. The elite strategy is as follows:

$$\text{Elite}^{(t)} = [x_{\text{best},1}^{(t)}, x_{\text{best},2}^{(t)}, x_{\text{best},3}^{(t)}, x_{\text{mean}}^{(t)}]. \quad (35)$$

$x_{\text{best},1}^{(t)}, x_{\text{best},2}^{(t)}$, and $x_{\text{best},3}^{(t)}$ are the top three individuals ranked in descending order based on their fitness values. $x_{\text{mean}}^{(t)}$ is considered to denote the weighted individuals, and the calculation method is as follows:

$$\begin{cases} x_{\text{mean}}^{(t)} = \sum_{i=1}^{0.2N_p} \bar{f}_i x_i^{(t)} \\ \bar{f}_i = \frac{f_i}{\sum_{i=1}^{0.2N_p} f_i} \end{cases}. \quad (36)$$

4.2.3 Incorporating the golden sine position update strategy

The BWO algorithm uses the spiral position update strategy from Eq. (30). During the later stages of the algorithm, all individuals in the population gather near the optimal individual, which accelerates the convergence speed of the algorithm but can lead to getting stuck in local optima, resulting in poor population diversity. The golden sine algorithm (Golden-SA) is a new metaheuristic algorithm (Lu and Zhang, 2023) that has strong global search capabilities. Golden-SA introduces a golden ratio coefficient in its position update process, allowing the algorithm to thoroughly search regions that can produce excellent solutions during each iteration, thereby accelerating the convergence speed and escaping local optima. Therefore, when updating positions in the BWO algorithm, the position update strategy from Golden-SA can be adopted to enhance the global search ability and accelerate its convergence speed. Thus, the new position update strategy is as follows:

$$x_i^{(t+1)} = x_i^{(t)} |\sin R_1| + r_1 \sin(2\pi r_2) |\varepsilon_1 \text{Elite}^{(t)} - \varepsilon_2 x_i^{(t)}|. \quad (37)$$

In the equation, r_1 and r_2 are random numbers between $[0, 1]$; R_1 is a random number between $[0, 2]$; and ε_1 and ε_2 are

golden ratio coefficients, and their specific expressions are

$$\begin{cases} \varepsilon_1 = -\pi\Gamma + \pi(1 - \Gamma) \\ \varepsilon_2 = -\pi(1 - \Gamma) + \pi\Gamma \end{cases} \quad (38)$$

In the equation, Γ is the golden ratio coefficient, and its value is $\Gamma = \frac{\sqrt{5}-1}{2}$.

5 Numerical experiment verification

5.1 Kinematic simulation verification

In this section, MATLAB numerical examples and Adams motion simulation are used together to verify the correctness of the kinematic equations derived for the 4-PUU PM. First, a numerical example of the mechanism's position analysis is provided. The mechanism's scale parameters are set as $R = 38$, $a = 18.75$, $b = 12.5$, $l = 43$ mm. The initial position of the fixed platform is set as $(X_P, Y_P, Z_P, \gamma)^T = (-20, 20, 20 \text{ mm}, -25^\circ)^T$; the final position of the moving platform is set as $(X_P, Y_P, Z_P, \gamma)^T = (20, -10, 50, -45^\circ)^T$. The initial and final positions of the moving platform are substituted into Eq. (13) to determine the corresponding sub-configurations of the mechanism ($h_i \geq \{f\} Z_{Ci}$) and the input rod lengths. The change in the driving pair of each limb is calculated as $\Delta h_i = (65.3012, 31.5674, 13.5590, 48.1959)$. To avoid motion shocks caused by sudden changes in velocity and acceleration, which could affect the kinematic simulation results, the four moving driving pairs follow a 3–4–5 polynomial motion law, expressed as

$$H_i = h_i + \Delta h_i \left(10 \left(\frac{\tau}{T} \right)^3 - 15 \left(\frac{\tau}{T} \right)^4 + 6 \left(\frac{\tau}{T} \right)^5 \right). \quad (39)$$

In the equation, H_i is the function representing the variation of the driving pair rod length of the i th limb with respect to time, and h_i is the initial rod length of the i th limb. $h_i = [25.0013, 49.0150, 59.4375, 36.7018]$; τ is the time variable, and its value range is $[0, 5]$; T is the maximum simulation time.

The total motion time of the mechanism is set to 5 s, and the continuous motion time T is discretized with a time interval of 0.1 s. Based on this, the position forward kinematics of the mechanism is solved using the method from Sect. 3.1.2. Then, the velocity and acceleration forward kinematics are obtained using Eqs. (18) and (22). Based on this, the angular displacement, angular velocity, and angular acceleration of the moving platform under the given motion law are obtained as MATLAB theoretical numerical curves, as shown in Fig. 6. At the same time, a three-dimensional model of the mechanism is established based on the mechanism's scale parameters and assembled. The assembly is then imported into Adams software, where the driving parameters from Eq. (39) are set, resulting in the output motion curve of the moving platform, as shown in Fig. 6.

As shown in Fig. 4a and f, the mechanism has translational DOF in the X , Y , and Z directions, as well as a rotational

DOF around the Z axis, thereby validating the correctness of the mechanism's DOF analysis in Sect. 2.2. At the same time, the Adams simulation results are in good agreement with the MATLAB theoretical results, confirming the correctness of the position, velocity, and acceleration equations derived for the 4-PUU PM in this paper.

5.2 Numerical experiments of the BWO algorithm

5.2.1 Single-point trajectory experiment

The mechanism scale parameters are consistent with those in Sect. 5.1. The coordinates of the end-effector reference point are $(X_P, Y_P, Z_P, \gamma)^T = (-20, 20, 20 \text{ mm}, -25^\circ)^T$. By substituting these values into Eq. (13), the inverse kinematics of the mechanism can be solved, as shown in Table 1.

For the first set of parameters in Table 1, the mechanism scale parameters are consistent with those in section 5.1. The range of the variables to be determined is $\mathbf{x}_{\min} = (-50, -50 \text{ mm}, 0, -\pi)^T$, $\mathbf{x}_{\max} = (50, 50, 100 \text{ mm}, \pi)^T$. In order to evaluate the computational performance of the IBWO algorithm in the forward kinematics of the PM, the algorithm is compared with the DEb1, PSO, and GWO algorithms. The acceptable error for all five algorithms is $f_{ac} = 10^{-8}$, the population size is $N_P = 40$, and the maximum number of iterations is $T = 500$. The algorithm is independently run 50 times, and the average evolution curve of the objective function is shown in Fig. 7. Table 2 presents the results of the four algorithms for the forward kinematics problem with independent random runs. f_w , f_m , f_b , f_{std} , and n_s represent the worst value, average value, best value, standard deviation, and the number of times the algorithm successfully found the forward kinematics solution, respectively, after 50 independent runs of the algorithm.

From Table 2, it can be observed that the values of f_b and n_s in the IBWO algorithm are the same as those in the PSO algorithm. However, the values of f_w , f_m , and f_{std} obtained by the IBWO algorithm are better than those of the PSO algorithm. At the same time, as shown in Fig. 7, the IBWO algorithm exhibits faster convergence speed and greater robustness. In summary, the performance of the IBWO algorithm is superior to the other three comparison algorithms.

5.2.2 Continuous trajectory error experiment

To calculate the error of the IBWO algorithm in solving the forward kinematics equation of the 4-PUU PM, the 51 end-effector output poses from Fig. 6a are substituted into Eq. (13) to obtain the corresponding input driving rod lengths h_i . These are used as the mechanism scale parameters. The IBWO algorithm is then independently run five times to solve for the output poses. The difference between the output poses obtained by the IBWO algorithm and the theoretical parameters is considered the error of the IBWO algorithm. The spe-

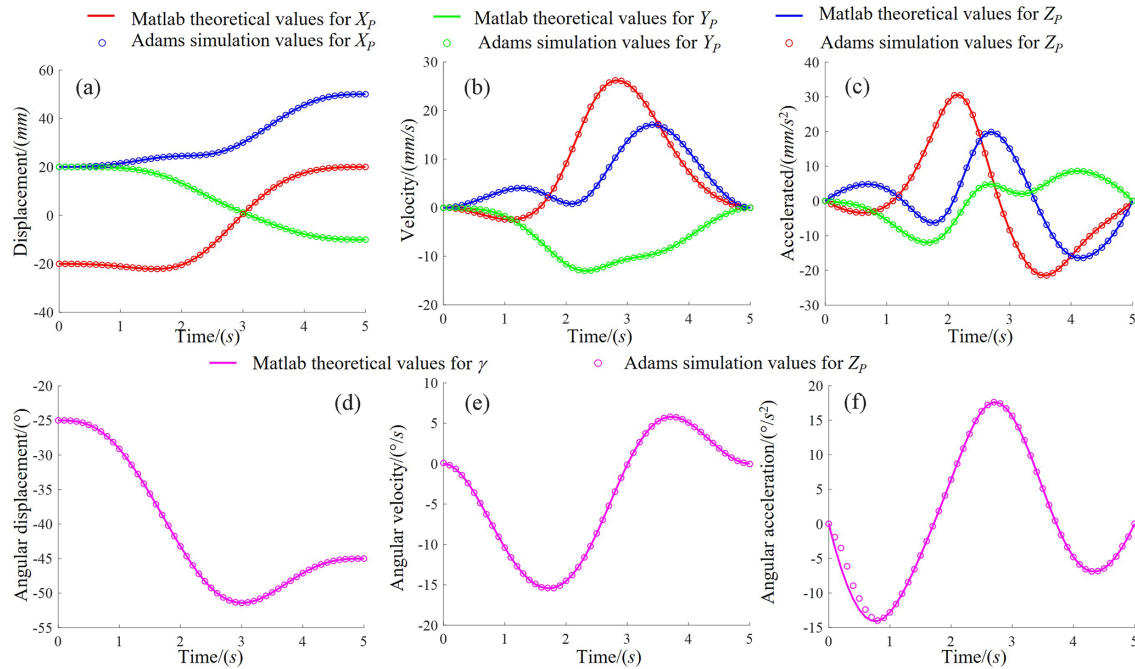


Figure 6. Comparison between MATLAB theoretical calculations and adams simulation analysis results showing (a) displacement, (b) velocity, (c) acceleration, (d) angular displacement, (e) angular velocity, and (f) angular acceleration.

Table 2. Performance statistical results of the four algorithms.

Algorithm	f_w	f_m	f_b	f_{std}	n_s
IBWO	1.0340×10^{-25}	2.4815×10^{-26}	0	2.8104×10^{-26}	50
DEb1	1.1175×10^5	1.8631×10^4	0	2.4603×10^4	21
PSO	1.0339×10^{-24}	8.4786×10^{-25}	0	1.9421×10^{-25}	50
ABC	8.5163×10^2	84.1835	0.1112	1.5004×10^2	2

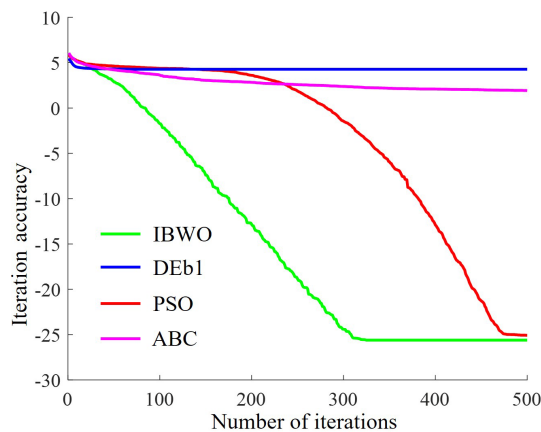


Figure 7. Evolution curve.

cific calculation expression is as follows:

$$\Delta Er = \frac{\sum_{i=1}^n (F_{IBWO} - F_T)}{n} \quad (40)$$

In the equation, ΔEr is the error generated by the IBWO algorithm in solving the forward kinematics equation of the PM, n is the number of independent runs of the IBWO algorithm, F_{IBWO} is the output position of the PM obtained by the IBWO algorithm, and F_T is the theoretical output pose of the PM.

Based on this, the average error curve of the IBWO algorithm in solving the forward kinematics equation of the 4-PUU PM is shown in Fig. 6.

As shown in Fig. 8a, when the IBWO algorithm is used to solve the forward kinematics equation for the continuous trajectory of the 4-PUU PM, the maximum error generated is approximately 2×10^{-7} mm. However, in practical engineering applications, an error of 0.01 mm is sufficient to meet the requirements. In conclusion, the roots of the forward kine-

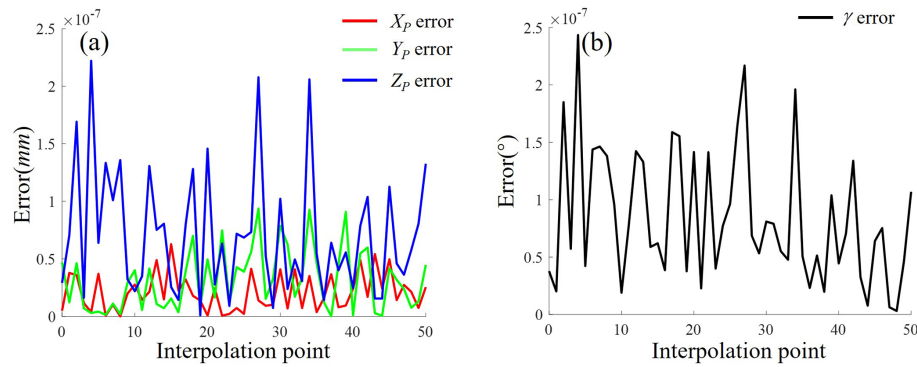


Figure 8. IBWO algorithm error showing (a) position error and (b) pose error.

matics equation of the 4-PUU PM solved by the IBWO algorithm meet the practical requirements. To improve the accuracy of the IBWO algorithm, it is only necessary to increase the acceptable root accuracy f_{acc} in the algorithm to achieve higher-precision roots.

6 Conclusions

This paper presents a 4-PUU PM with 3T1R DOF, whose workspace is unaffected by the actuated joints. The mechanism's DOF characteristics are analyzed using screw theory. Based on the constraint of link lengths, the forward and inverse kinematics equations of the mechanism are derived. On this basis, the first and second time derivatives of the forward kinematics equations are obtained, yielding the velocity and acceleration equations of the mechanism. Given the scale parameters and actuator parameters of the mechanism, simulation experiments are conducted using both MATLAB and Adams software. The experimental results validate the correctness of the kinematic analysis of the 4-PUU PM. An IBWO is proposed to solve the forward kinematics equations of the 4-PUU PM, and comparative algorithm experiments are performed. The experimental results show that the IBWO algorithm outperforms other comparison algorithms. Finally, based on the forward kinematics solutions for the interpolation points of continuous trajectories, it is found that the IBWO algorithm meets the accuracy requirements for the mechanism's practical applications.

In the future, the IBWO algorithm will be used for the optimization of the scale parameters of the 4-PUU PM. Based on the optimization results, a prototype will be designed, and the IBWO algorithm will be applied to the prototype for kinematic trajectory planning of the mechanism.

Code and data availability. No data sets were used in this paper. Readers interested in the MATLAB implementation of the IBWO algorithm may contact the corresponding author via email to request the source code.

Author contributions. KW: writing (original draft), idea, methodology, validation, and investigation. XZ: modeling, simulation, and data processing. AJ: experimental code debugging. JL: supervision and manuscript revision.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Review statement. This paper was edited by Daniel Condurache and reviewed by Baoyu Wang and three anonymous referees.

References

- Arora, S. and Anand, P.: Chaotic grasshopper optimization algorithm for global optimization, *Neural Comput. Appl.*, 31, 4385–4405, <https://doi.org/10.1007/s00521-018-3343-2>, 2019.
- Company, O., Marquet, F., and Pierrot, F.: A new high-speed 4-DOF parallel robot synthesis and modeling issues. *IEEE Trans. Robot. Autom.*, 19, 411–420, <https://doi.org/10.1109/TRA.2003.810232>, 2003.
- Ghalambaz, M., Yengejeh, R. J., and Davami, A. H.: Building energy optimization using Grey Wolf Optimizer (GWO), *Case Studies in Thermal Engineering*, 27, 101250, <https://doi.org/10.1016/j.csite.2021.101250>, 2021.
- Herbadji, D., Derouiche, N., Belmeguenai, A., Herbadji, A., and Boumerdassi, S.: A Tweakable Image Encryption Algorithm Using an Improved Logistic Chaotic Map, *Trait Signal*, 36, 407–417, <https://doi.org/10.18280/ts.360505>, 2019.
- Krut, S., Benoit, M., Ota, H., and Pierrot, F.: I4: A new parallel mechanism for SCARA motions/Proceedings of the IEEE International Conference on Robotics and Automation, 14–19

- September 2003, Taiwan, China, New York: IEEE, 2003, 1875–1880, <https://doi.org/10.1109/ROBOT.2003.1241868>, 2003.
- Li, C., Wu, H., and Eskelinen, H.: Design and Multi-Objective Optimization of a Dexterous Mobile Parallel Mechanism for Fusion Reactor Vacuum Vessel Assembly, *IEEE Access*, 2021, 153796–153810, <https://doi.org/10.1109/ACCESS.2021.3127947>, 2021.
- Li, Y., Mei, J., Liu, S., and Huang, T.: Dynamic dimensional synthesis of a 4-DOF high-speed parallel manipulator, *J. Mech. Eng.*, 50, 32–40, 2014.
- Liu, C., Cao, G., and Qu, Y.: Safety analysis via forward kinematics of delta parallel robot using machine learning, *Safety Sci.*, 117, 243–249, <https://doi.org/10.1016/j.ssci.2019.04.013>, 2019.
- Lu, Z. H. and Zhang, J.: An Enhanced Dung Beetle Optimization Algorithm for Global Optimization, *Curr. J. Appl. Sci. Technol.*, 42, 9–22, <https://doi.org/10.9734/cjast/2023/v42i174133>, 2023.
- Nabat, V., de la O Rodriguez, M., Company, O., Krut, S., and Pierrot, F.: Par4: Very high speed parallel robot for pick-and-place. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2–6 August 2005, Alberta, New York, IEEE, 2005, 1202–1207, <https://doi.org/10.1109/IROS.2005.1545143>, 2005.
- Salgado, O., Altuzarra, O., Petuya, V., and Hernández, A.: Synthesis and Design of a Novel 3T1R Fully-Parallel Manipulator, *J. Mech. Des.*, 130, 042305, <https://doi.org/10.1115/1.2839005>, 2008.
- Shi, K., Wang, Z., Yan, C., and Wang, Z.: Design and performance analysis of the 4UPS-RRR parallel ankle rehabilitation mechanism, *Mech. Sci.*, 15, 417–430, <https://doi.org/10.5194/ms-15-417-2024>, 2024.
- Staicu, S., Shao, Z., Zhang, Z., Tang, X., and Wang, L.: Kinematic analysis of the X4 translational–rotational parallel robot, *Int. J. Adv. Robot. Syst.*, 15, 1–12, <https://doi.org/10.1177/1729881418803849>, 2018.
- Wu, X., Wu, S., Wang, C., Luo, J., and Wang, Y.: Kinematics analysis of a novel hybrid manipulator for optomechanical modules assembly, *Fusion Eng. Des.*, 202, 1–6, <https://doi.org/10.1016/j.fusengdes.2024.114316>, 2024.
- Wen, S., Ji, A., Che, L., and Yang, Z.: Time-varying external archive differential evolution algorithm with applications to parallel mechanisms. *Appl. Mathmat. Model.*, 114, 745–769, <https://doi.org/10.1016/j.apm.2022.10.026>, 2023.
- Wen, S., Wu, S., Qi, Q., Wu, H., Yang, Y., Cheng, Y., and Ji, A.: Design and position analysis of 2PUU-2PSS parallel mechanism for auxiliary assembly in DEMO vacuum vessel, *Fusion Eng. Des.*, 2007, 114645, <https://doi.org/10.1016/j.fusengdes.2024.114645>, 2024.
- Xiang, T., Liao, X., and Wong, K. W.: An improved particle swarm optimization algorithm combined with piecewise linear chaotic map, *Appl. Math. Comput.*, 190, 1637–1645, <https://doi.org/10.1016/j.amc.2007.02.103>, 2007.
- Xie, F., Li, T., and Liu, X.: Type synthesis of 4-DOF parallel kinematic mechanisms based on Grassmann line geometry and atlas method, *Chin. J. Mech. Eng.*, 26, 10731081, <https://doi.org/10.3901/cjme.2013.06.1073>, 2013.
- Yang, S., Sun, T., and Huang, T.: Type synthesis of parallel mechanisms having 3T1R motion with variable rotational axis, *Mech. Mach. Theory.*, 109, 220–230, <https://doi.org/10.1016/j.mechmachtheory.2016.11.005>, 2017.
- Zhong, C., Li, G., and Meng, Z.: Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowl. Based Syst.*, 251, 109215, <https://doi.org/10.1016/j.knosys.2022.109215>, 2022.
- Zhu, Z., Liu, Y., He, Y., Wu, W., Wang, H., Huang, C., and Ye, B.: Fuzzy PID Control of the Three-Degree-of-Freedom Parallel Mechanism Based on Genetic Algorithm, *Appl. Sci.*, 12, 11128, <https://doi.org/10.3390/app122111128>, 2022.