



# Inverse Kinematics Solution of 6-DOF Manipulator Based on Multi-Objective Full-Parameter Optimization PSO Algorithm

Sha Luo<sup>†</sup>, Dianming Chu<sup>†</sup>, Qingdang Li\* and Yan He\*

College of Electromechanical Engineering, Qingdao University of Science and Technology, Qingdao, China

## OPEN ACCESS

### Edited by:

Hong Qiao,  
University of Chinese Academy of  
Sciences, China

### Reviewed by:

Jyoti Ohri,  
National Institute of Technology,  
Kurukshestra, India  
Serkan Dereli,  
Sakarya University of Applied  
Sciences, Turkey  
Serdar Kucuk,  
Kocaeli University, Turkey

### \*Correspondence:

Qingdang Li  
lqd@qust.edu.cn  
Yan He  
heyang@qust.edu.cn

<sup>†</sup>These authors share first authorship

**Received:** 09 October 2021

**Accepted:** 14 February 2022

**Published:** 17 March 2022

### Citation:

Luo S, Chu D, Li Q and He Y (2022)  
Inverse Kinematics Solution of 6-DOF  
Manipulator Based on Multi-Objective  
Full-Parameter Optimization PSO  
Algorithm.  
*Front. Neurobot.* 16:791796.  
doi: 10.3389/fnbot.2022.791796

A multi-objective full-parameter optimization particle swarm optimization (MOFOPSO) algorithm is proposed in this paper to overcome the drawbacks of poor accuracy, low efficiency, and instability of the existing algorithms in the inverse kinematics(IK) solution of the manipulator. In designing the multi-objective function, the proposed algorithm considers the factors such as position, posture, and joint. To improve PSO, the proposed algorithm comprehensively analyzes all factors affecting the global and local searching abilities. Based on this, the initial population is designed following the localized uniform distribution method. Meanwhile, the inertia weight, asynchronous learning factor, and time factor are respectively designed by introducing the iteration factor. Finally, this paper tests the performance of MOFOPSO with three typical functions to obtain a better inverse kinematics solution of the 6-DOF manipulator. Also, six other algorithms are taken for performance comparison. The experimental results indicate that the proposed method not only ensures the stability of the manipulator but also achieves high accuracy and efficiency in solving the inverse kinematics of the 6-DOF manipulator.

**Keywords:** inverse kinematics, multi-objective full-parameter optimization particle swarm optimization, inertia weight, asynchronous learning factor, time factor

## INTRODUCTION

Recently, with the increase of the application scenarios and the number of mechanical arms, achieving high precision, and stability of the manipulator motion control system has attracted much attention. As an important factor to determine the motion precision and stability of the manipulator, kinematics modeling, and solving of manipulator has become a research hotspot. Functionally, kinematics can be divided into forward and inverse kinematics. With the change of each joint angle, the position and attitude of the end-effector of the manipulator can be obtained by using forward kinematics. This process represents the transformation from joint space to Cartesian space. The inverse kinematics solves the variation of the joint angles of the manipulator based on the position and attitude of the end-effector. Since the kinematics equation of a manipulator is a set of nonlinear equations, the solution to these equations must consider the existence of single solution and multi-solution and solution method. Compared to forwarding kinematics, inverse kinematics is more difficult to model and solve.

The study of inverse kinematics is crucial in the field of robotics research. This is not only because inverse kinematics is the basis of robot trajectory planning motion control and

workspace analysis, but also it is a technical problem in robotics (Shi et al., 2020). With the continuous study of inverse kinematics, the inverse kinematics of the manipulator can be solved by four methods, including geometric method, analytical method, numerical method, and artificial intelligence method. The geometric method was first proposed to solve inverse kinematics. The paper (Xie et al., 2006) proposed a geometric method for the inverse kinematics of a 2-DOF manipulator. However, the process of solving the inverse kinematics of the multi-DOF manipulator is very complex, and the geometric method does not work. The most useful method for solving this problem is the analytical and numerical methods. Before solving inverse kinematics, the analytical method must analyze the number and structure of the joint of the manipulator. Craig (2009), Murray (2017), and Angeles and Angeles (2002) exploited the analytical method to solve the IK of the decoupled manipulator. In Funda and Paul (1990) concluded that quaternion is the most economical point transformation operator for the rigid screw displacement problem. In Raghavan and Roth (1993) proposed the elimination method as the solving method for the IK of the 6-DOF manipulator, in which the solution to each position equation of the manipulator was improved in the form of a six-member quadratic polynomial. In Husty et al. (2007) combined double quaternion with Segre flow patterns for the IK of the 6-DOF manipulator. Although the analytical method is very efficient in solving the IK equation, there are some shortcomings such as low accuracy, poor real-time performance, and difficulty in ensuring stable operations of the manipulator in a dynamic environment. In contrast, there are a variety of numerical methods that can be exploited to solve inverse kinematics, such as Newton's method, Jacobian iterative method (Buss, 2004), and mixed inverse kinematics method. Because the Jacobian iteration method has a simple principle and no special requirements on the structure and number of joints of the manipulator, it is exploited as a method for solving most IK problems. Meanwhile, the numerical method also has the advantage that it can obtain an inverse solution containing path planning from the initial pose to the target pose, so it is suitable for path planning. However, in theory, it is impossible to verify whether the inverse solution is optimal or suboptimal.

With the continuous development of artificial intelligence and intelligent algorithms, some artificial intelligence methods have been used to solve the inverse kinematics of manipulators and have achieved good effects. And compared with geometric, numerical and analytical methods to solve the inverse kinematics of manipulator, the artificial intelligence algorithm does not need to carry out derivation operation to obtain jacobian matrix, and does not have the singularity problem in the conventional iterative method (Shi et al., 2020). And artificial intelligence algorithm has no special requirements for robot mechanism and has strong universality, so it is applied to parallel (Kucuk, 2012) and hybrid robots (Tanev, 2000; Serdar and Baris Doruk, 2016). The most widely used algorithms include ANN (artificial neural network), GA (genetic algorithm), and PSO. Hasan et al. (2006) proposed an ANN-based adaptive learning algorithm to overcome the uncertainty and nonlinearity in solving IK equations. Momani et al. (2016) solved the IK of the manipulator

with GA, which obtained path planning in the iterative process. Ma et al. (2016) proposed to use ANN to solve the IK of the manipulator. Khaleel (2018) proposed a combination of neural network algorithm and GA for the IK of the manipulator. Although ANN and GA have achieved good results in solving the IK problem, both of them have some defects. ANN suffers from a long training time and weak generalization ability, while GA has a long iteration time and easily falls into local optimum. In 2018, Nizar and Alimi (2013) used PSO and improved PSO to solve the IK problem, and they compared the simulation results of the two algorithms. In 2013, Idris et al. (2018) proposed an adaptive PSO. They applied this algorithm to solve the IK of a 3-DOF manipulator and achieved good results. Dereli and Koker (2018) proposed an adaptive weighted PSO (AWPSO) algorithm to achieve high position accuracy. The 7-DOF manipulator was taken as the object for simulation experiments, and the simulation results were compared with those of the traditional PSO algorithm.

All the above studies only focused on solving the IK problem of the manipulator but rarely conducted a comprehensive and in-depth study of the IK of the manipulator. However, the development of the IK algorithm is limited by the accuracy of position and attitude and the stability of the manipulator. Aiming at the poor accuracy and low efficiency in solving the manipulator and the instability in running the manipulator, the MOFOPSO algorithm is proposed in this paper. Firstly, the algorithm minimizes the manipulator's position error, attitude error, and joint angle change. Then, it transforms the solution of the inverse kinematics of the manipulator into a multi-objective optimization problem.

## KINEMATICS ANALYSIS OF MANIPULATOR

For further studying the kinematics of the manipulator, this paper selects the 6-DOF manipulator as the research object. Firstly, the kinematic model of the manipulator needs to be established. In terms of kinematic model, literature (Ayiz and Kucuk, 2009) uses exponential rotation matrix to directly describe the physical mechanism of the manipulator, but the multiple transformations of matrix produce a large amount of calculation. In this paper the coordinate system of the 6-DOF manipulator is established according to the Denavit and Hartenberg model (Denavit and Hartenberg, 1964), which is shown in **Figure 1**.

In terms of the adjacent coordinate systems of the manipulator  $i$  and  $i - 1$ , the homogeneous coordinate transformation matrix  ${}^{i-1}_i T$  can be used to express the relative relation of the position and pose.

$${}^{i-1}_i T = \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

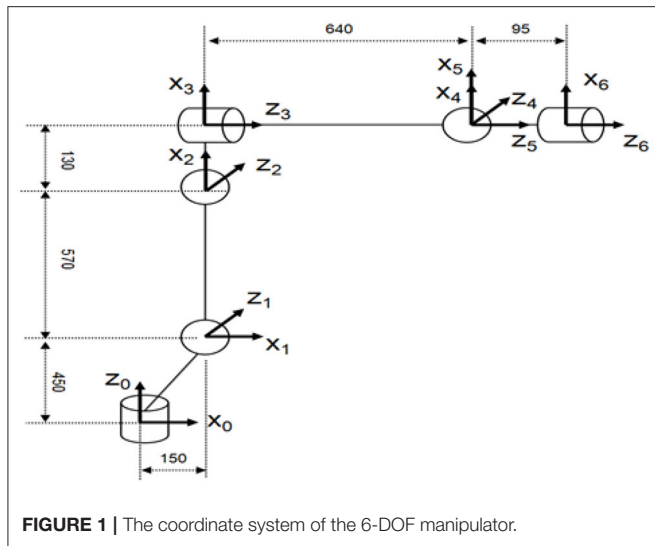


FIGURE 1 | The coordinate system of the 6-DOF manipulator.

TABLE 1 | The DH parameters of the 6-DOF manipulator.

$i$	$d_i(\text{mm})$	$\alpha_i(\text{Deg})$	$a_i(\text{mm})$	$\theta_i(\text{Deg})$	$\theta_i(\text{Degree})$
1	450	-90	150	$\theta_1$	-170~170
2	0	0	570	$\theta_2$	-110~155
3	0	-90	130	$\theta_3$	-80~210
4	640	90	0	$\theta_4$	-190~190
5	0	-90	0	$\theta_5$	-50~230
6	95	0	0	$\theta_6$	-360~360

According to the DH model, it is easy to obtain the model of the 6-DOF manipulator's forward kinematics, and the model can be expressed as follows:

$${}^0_6T = {}^0_1T_1 {}^1_2T_2 {}^2_3T_3 {}^3_4T_4 {}^4_5T_5 {}^5_6T_6 = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

Where  ${}^0_6T$  represents the position matrix of the end-effector of the mechanical arm relative to the inertial coordinate system;  ${}^{i-1}_i T$  represents the homogeneous transformation matrix of adjacent coordinate systems.

The IK of the manipulator can be solved by obtaining the pose matrix of the end-effector of the manipulator and solving the changes of the joint angles of the manipulator. The DH parameters of the 6-DOF manipulator used in this paper are listed in Table 1.

Based on the parameters in the DH table and equations (1) and (2), the expression of the pose matrix of the end-effector of the manipulator can be obtained.

## THE PROPOSED MULTI-OBJECTIVE FULL-PARAMETER OPTIMIZATION PARTICLE SWARM OPTIMIZATION METHOD

### Classical PSO Method

With the progress of the research on the IK of manipulators, artificial intelligence has been exploited to solve the inverse kinematics, and the swarm optimization algorithm has been widely used in this field. As early as 1995, Eberhart and Kennedy (1995) proposed PSO under the inspiration of birds' foraging behavior. PSO simulates the birds in the flock in the form of massless particles and gives two attributes of the particle, i.e., the speed and direction of movement. During the whole search process, there are two extreme values. One is the current individual extreme value, which is searched by the individual particle separately. Also, the individual particle exchanges information with other particles in the group. Another extreme value is the group extreme value, which is updated into the best individual extreme value. Based on the two extreme values, all particles in the group dynamically adjust their velocity and position, and the update equations are as follows:

$$v_{jd} = v_{jd} + c_1 r_1 (p_{best} - x_{jd}) + c_2 r_2 (g_{best} - x_{jd}) \quad (3)$$

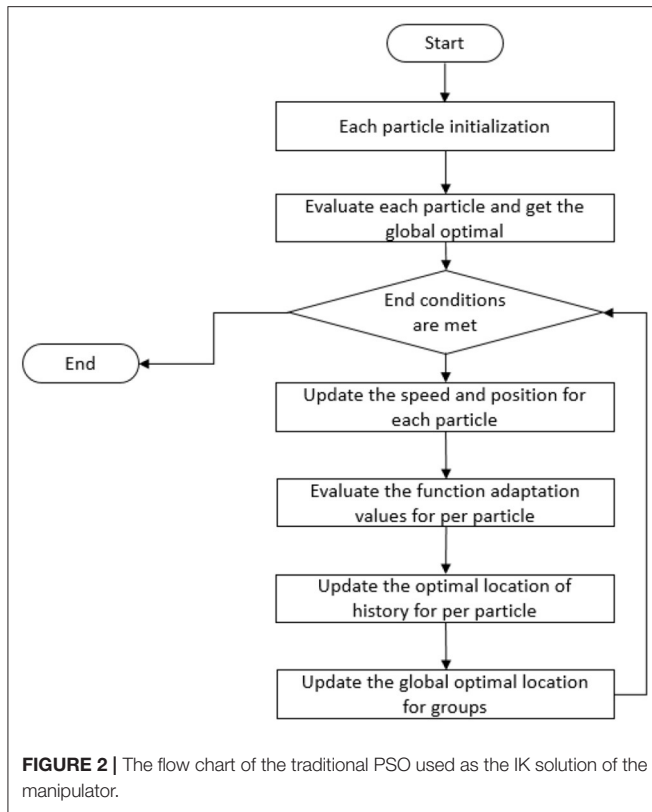
$$x_{jd} = x_{jd} + v_{jd} \quad (4)$$

Where,  $X_j = (x_{j1}, x_{j2}, \dots, x_{jD})$  is the position of the  $j$  particle;  $P_j = (p_{j1}, p_{j2}, \dots, p_{jD})$  denotes the extreme value;  $g_{best}$  is the population extreme value;  $v_{jd}$  is the velocity of the particle;  $c_1$  and  $c_2$  are constants representing the learning coefficient;  $r_1$  and  $r_2$  are random real numbers in the range of  $[0, 1]$ ;  $d = 1, 2, 3, \dots, D$  is the dimension, and  $j = 1, 2, 3, \dots, S$  is the  $j$  particle.

The flow chart of using the classical PSO to solve the IK of the manipulator is shown in Figure 2.

### Multi-Objective Full-Parameter Optimization PSO Method

In the intelligent algorithm, the PSO algorithm has many advantages, such as easy implementation, fewer tuning parameters, and fast convergence, so it can obtain good results in the application field. However, in many cases, the classical PSO algorithm is easy to fall into the local optimum. Therefore, the studies (Shi and Eberhart, 1998a,b) optimize the traditional PSO algorithm for balancing global and local search capabilities by changing the inertia weight. Besides the inertia weight, the searching ability and efficiency of the PSO algorithm are affected by the selection of learning factors, initial population, and fitness function. In recent years, researchers have improved various particle swarm optimization algorithms in Table 2. So, considering these factors, the initial population is designed in this paper following the localized uniform distribution method. Then, the inertia weight, asynchronous learning factor, and time factor are respectively designed



**TABLE 2 |** Particle swarm optimization and its improved algorithm.

References	Algorithm	Improve
Eberhart and Kennedy, 1995	PSO	Based on the flock-based predation behavior
Lovbjerg et al., 2001	Breed	With the advantages of both two algorithms, faster speed and higher accuracy
Zhan et al., 2009	Adaption	Inertial weights change and converge faster than the Breed Algorithm
Lili et al. (2015)	Lin	Improve the ability to find excellence, reduce the oscillation phenomenon
Yaming (2019)	Nature	More optimization ability and solution speed, reduce precocious convergence

following the method of introducing the iteration factor. Finally, a multi-objective PSO algorithm with full parameters is designed in combination with the multi-objective optimization algorithm. The proposed algorithm makes the particle swarm in the initial state more representative and obtains the optimal solution quickly. Also, it integrates the inertia weight and learning factor into the design of the factors and the process of location update algorithm to balance global and local search capabilities, thus improving the diversity and generalization ability of the algorithm.

### Initialize the Population Design

The selection of the initial population directly affects the search efficiency of the PSO algorithm. Theoretically, the optimal solution of the problem is unknown at the initial stage of the problem. So, if the initial population is randomly generated, the population representation is poor, which can affect the efficiency of particle swarm search. Meanwhile, with the increase in the dimensionality of the search space, the algorithm is easy to fall into local optimum because of the randomness of the initial position of particle swarm optimization. If the particles can be uniformly distributed in the feasible region during initialization, the efficiency of the global search can be guaranteed, and the probability of searching for the optimal solution by the population can be improved. In response to this problem, a limited domain uniform distribution method is proposed to generate the initial population. Considering the robot joint angle restriction, the initial value of each dimensional particle swarm is uniformly distributed within this range.

### Optimal Inertia Weight Strategy

In PSO algorithms, inertia weight is a very important parameter to balance the global and local search capabilities of the whole algorithm. If the inertial weight is too large, the global search capability is strong but the local search capability is poor; if the inertial weight is too small, the local search capability is strong but the global search capability is poor (Xi-Hu, 2011). Although the decline of linear weight can improve the balance of the global and local search capabilities of the algorithm to some extent, the local search capability will be weakened as the number of iterations decreases linearly. It is found that a large inertial weight is conducive to the global search at the beginning of the iteration, while a small inertial weight is conducive to local precise search at the later stage of the iteration. However, a fast decline process is easy to cause the problem of search fault. Therefore, this paper designed the nonlinear function of optimal inertia weight by introducing the factor of iteration number. Based on this, the range can be quickly determined at the beginning of the iteration. Then, the local precise positioning is realized with a small change of inertia weight to obtain the optimal inertia weight at a fast speed. The function is as follows:

$$w = w_{max} - (w_{max} - w_{min}) \sin\left(\frac{r\pi}{2N}\right) \quad (5)$$

Where,  $w \in [w_{min}, w_{max}]$ ;  $r$  and  $N$  are respectively the current iteration and total iterations number of the algorithm.

### Asynchronous Learning Factor

The learning factor is usually set to a constant in the classical PSO algorithm. But it is an important factor that affects the self-cognition and social cognition of particles and the movement direction and position of particles. Also, it reflects the degree of information communication among particles in the whole population. To make the global search ability better, at the beginning of the algorithm iteration, the particles are expected to have more group communication, that is, strong social cognition ability, which facilitates the global search. In later iterations,

the self-cognition ability of the particle is expected to be strong to achieve an accurate local search. Based on the analysis, the function and iteration number factor are introduced into the learning factor design in this paper. Also, in the early iteration, the learning factor is adjusted to achieve a strong social cognition ability and a strong self-cognition ability in the late iteration, Besides, the optimal learning factor is quickly realized through nonlinear changes. The learning factors are set as follows:

$$\begin{cases} c_1 = c_{min} + (c_{max} - c_{min}) \frac{2}{1+e^{-(1+20r/N)}} \\ c_2 = c_{max} - (c_{max} - c_{min}) \frac{2}{1+e^{-(1+20r/N)}} \end{cases} \quad (6)$$

Where,  $c_1, c_2 \in [c_{min}, c_{max}]$ ;  $r$  and  $N$  are respectively the current iteration and total iterations number of the algorithm.

### Time Factor

The position update strategy of the classical PSO is based on the initial position and the current speed. From the perspective of physics, the addition of two physical quantities requires the quantities have the same dimension, which means that the direct addition of displacement and velocity does not conform to the physical theory. Therefore, in the classical PSO algorithm, the position update formula contains a time factor that is important for particles to oscillate near the optimal solution. However, this time factor is usually set to 1 (Kong et al., 2010). To improve the particle search ability, the time factor is adopted by this paper. In equation (4), a time factor is added to the local update model, and the update function is as follows:

$$x_{id} = x_{id} + Tv_{id} \quad (7)$$

Through experiments, it is found that the convergence of the algorithm is affected by the time factor. So, the iteration number is introduced to design the time factor:

$$T = 0.5 + \frac{r}{2N} \quad (8)$$

Where,  $r$  and  $N$  are respectively the current iteration and total iterations number of the algorithm.

### Multi-Objective Function

Multi-objective optimization (He and Shao-hua, 2019) is complex, and it is widely used in engineering applications. However, in this optimization problem, each objective cannot achieve the optimal at the same time due to non-linearity, multi-dimension, and other characteristics. Also, each objective must have a weight. In this case, the allocation of the weight becomes a hot research topic. There are a variety of traditional multi-objective optimization methods, such as the weighted summation method (Zadeh, 1963), objective programming method (Charnes et al., 1955), maximum and minimum value (Tseng and Lu, 1990), etc. The key of these methods is to transform the multi-objective problem into a single-objective problem and then use the single objective algorithm for optimization. However, these methods rely too much on experience and fail to achieve the optimization effect when the multi-objective problem has non-linearity and high latitude. Recently, with the continuous development of the evolutionary algorithm along with its unique update mechanism, various evolutionary algorithms have been applied to the combinatorial optimization and numerical optimization fields (Liu et al., 2020) and achieved breakthroughs. The MOPSO algorithm (Li et al., 2017) is one of the typical multi-objective algorithms. Due to the precocity of particle swarm in the traditional multi-objective PSO algorithm, the particle swarm will suffer from local optimization, poor convergence, global search, and local search ability imbalance (Feng et al., 2020). In this section, the multi-objective function will be optimized and improved for the IK of the manipulator to make the MOPSO algorithm have excellent optimization performance and pertinence.

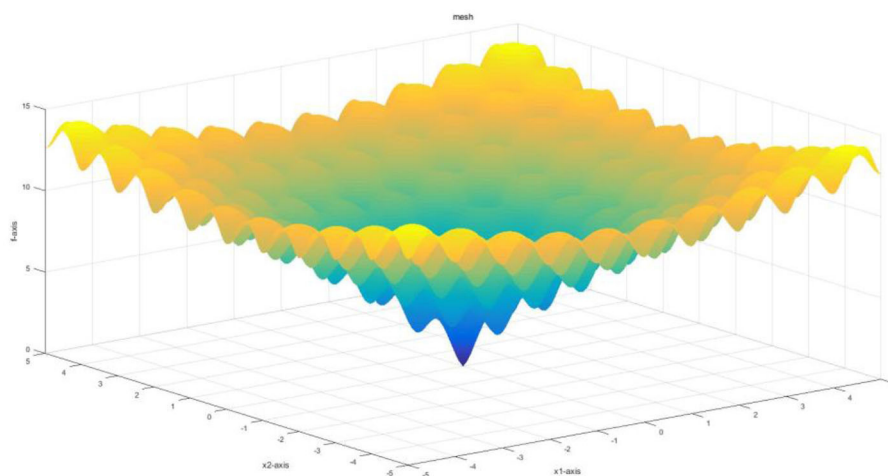


FIGURE 3 | The image of the Ackely function.

Based on the multi-objective optimization and taking the 6-DOF manipulator as the object, this paper designs a multi-objective function to minimize position error, attitude error, and joint angle change:

$$\begin{cases} F = \text{Min}(f_1(p_{tar}), f_2(e_{tar}), f_3(\theta_i)) \\ s.t. \theta_{i \min} \leq \theta_i \leq \theta_{i \max} \end{cases} \quad (9)$$

Where,  $F$  is the multi-objective function;  $f_1(p_{tar})$ ,  $f_2(e_{tar})$ , and  $f_3(\theta_i)$  are respectively the position error function, the attitude error function, and the joint angle change function;  $\theta_{i \min}$  and  $\theta_{i \max}$  are the ranges of joint angle  $\theta_i$ .

To make the algorithm more targeted, the structural principle of the manipulator is studied in-depth, and then the position

error function, attitude error function, and angle change function of each joint are designed. Especially, the joint angle change function involves the changes of multiple joint angles. If it is designed with the minimum changes of all joint angles, it does not conform to the structural principle of the manipulator. Specifically, there is the least change in the inertial coordinate system of the manipulator in the whole operation process, while the joint changes at the end of the manipulator are generally large. So, the linear weighted summation method is adopted by this paper to design the joint angle change function. The mathematical description of the position error function, attitude error function, and joint angle change function is as follows:

$$\begin{cases} f_1(p) = \|p_{tar} - H_{position}(q_{tar})\| \\ f_2(p) = \|e_{tar} - H_{orientation}(q_{tar})\| \\ f_3(\theta_i) = \sum_{i=1}^6 \lambda_i \sqrt{(\theta_i - \theta_{init})^2} \\ \sum_{i=1}^6 \lambda_i = 1 \\ i = 1, 2 \dots 6 \end{cases} \quad (10)$$

Where,  $p_{tar}$  and  $e_{tar}$  are respectively the target position and attitude of the end-effector of the manipulator;  $q_{tar}$  is the angle value of each joint angle of the manipulator, and it can be solved by the equation;  $\theta_{init}$  contains the initial values of the joint angles of the manipulator, and  $\lambda_i$  is the weight value of the change of joint angle  $\theta_i$ .

**Fitness Function**

The construction of the fitness function is the most important step in the PSO algorithm. For the inverse kinematics, this paper designs the fitness function by considering the position, attitude accuracy, and joint angle change at the same time. According to equation (9), the fitness function with the minimum variation

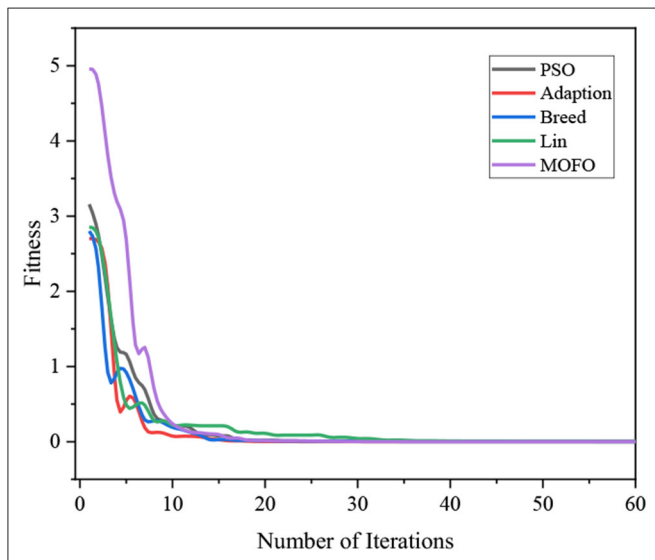


FIGURE 4 | The comparison of the convergence speed of different algorithms.

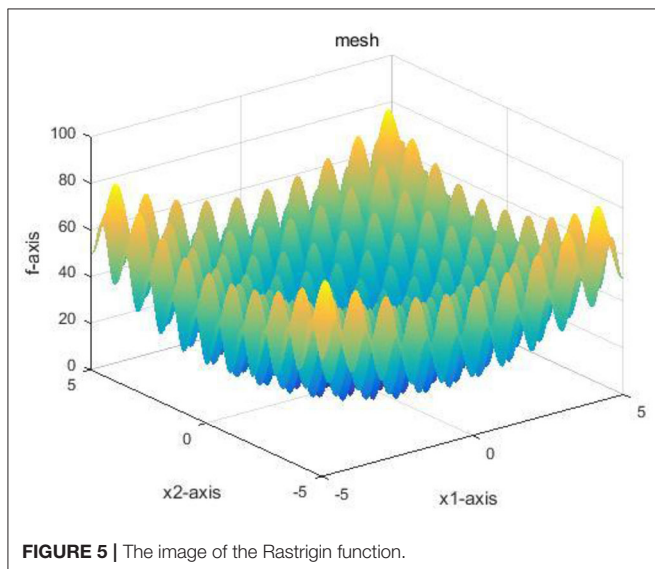


FIGURE 5 | The image of the Rastrigin function.

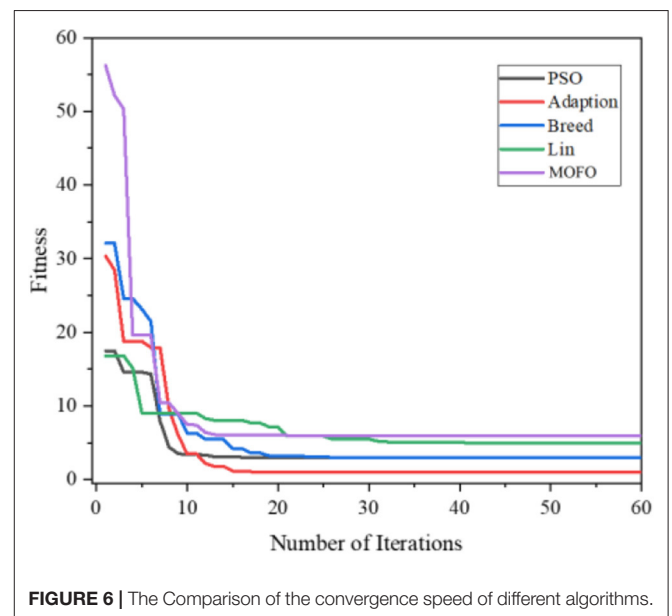


FIGURE 6 | The Comparison of the convergence speed of different algorithms.

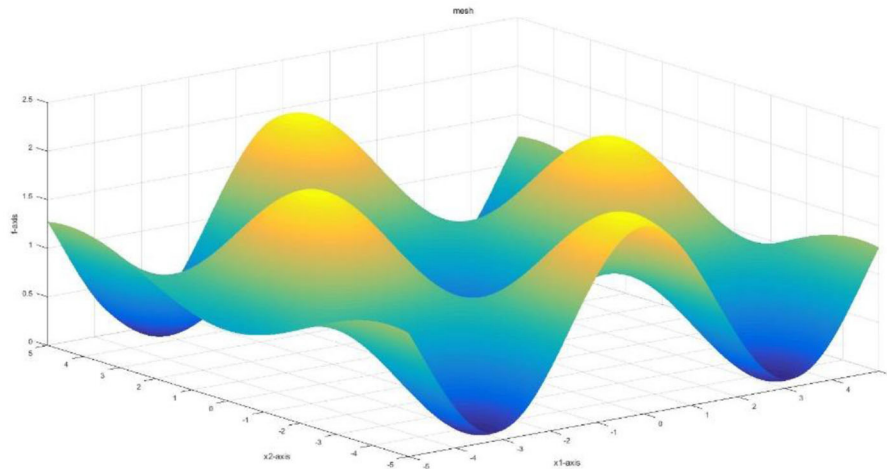


FIGURE 7 | The image of the Griewangk function.

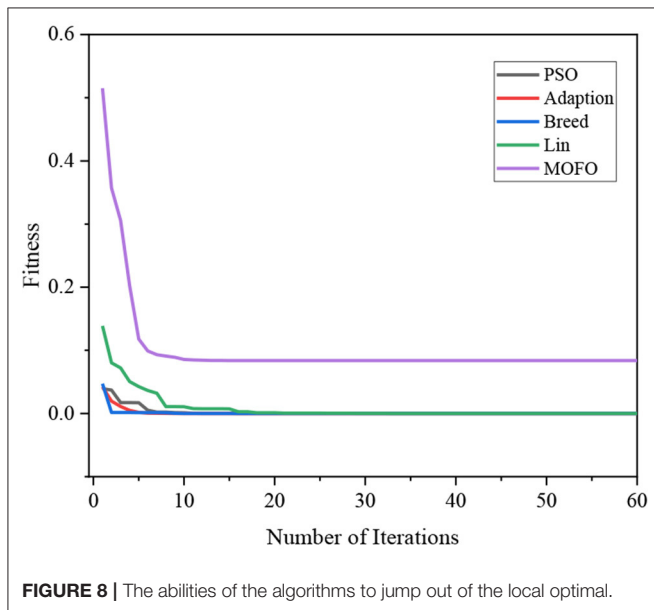


FIGURE 8 | The abilities of the algorithms to jump out of the local optimal.

of position accuracy, posture accuracy, and joint angle can be obtained:

$$\begin{cases} f_1(q_{cur}) = \sqrt{(p_{x\_tar} - p_{x\_cur})^2 + (p_{y\_tar} - p_{y\_cur})^2 + (p_{z\_tar} - p_{z\_cur})^2} \\ f_2(q_{cur}) = \sqrt{(e_{x\_tar} - e_{x\_cur})^2 + (e_{y\_tar} - e_{y\_cur})^2 + (e_{z\_tar} - e_{z\_cur})^2} \\ f_3(\theta_i) = \sum_{i=1}^6 \lambda_i \sqrt{(\theta_i - \theta_{init})^2} \end{cases} \quad (11)$$

Where,  $p_{tar} = [p_{x\_tar}, p_{y\_tar}, p_{z\_tar}]^T$  and  $e_{tar} = [e_{x\_tar}, e_{y\_tar}, e_{z\_tar}]$  are respectively the target position and attitude of the manipulator;  $p_{cur} = [p_{x\_cur}, p_{y\_cur}, p_{z\_cur}]^T$  and  $e_{cur} = [e_{x\_cur}, e_{y\_cur}, e_{z\_cur}]$  respectively represent the initial

position and attitude of the manipulator;  $\lambda_i$  is the weight value of the change of joint angle  $\theta_i$ .

### IK Model of the 6-DOF Manipulator Based on MOFOPSO

Based on the MOFOPSO algorithm, the IK model of the 6-DOF manipulator can be obtained as follows:

$$\begin{cases} F = \text{Min}(f_1(p_{tar}), f_2(e_{tar}), f_3(\theta_i)) \\ s.t. \theta_{i\min} \leq \theta_i \leq \theta_{i\max} \\ v_{jd} = w \cdot v_{jd} + c_1(p_{best} - x_{jd}) + c_2(g_{best} - x_{jd}) \\ x_{jd} = x_{jd} + T v_{jd} \\ w = w_{max} - (w_{max} - w_{min}) \sin\left(\frac{rT}{2N}\right) \\ c_1 = c_{min} + (c_{max} - c_{min}) \frac{2}{1 + e^{-(1+20r/N)}} \\ c_2 = c_{max} - (c_{max} - c_{min}) \frac{2}{1 + e^{-(1+20r/N)}} \\ T = 0.5 + \frac{r}{2N} \\ f_1(p) = \|p_{tar} - H_{position}(q_{tar})\| \\ f_2(p) = \|e_{tar} - H_{orientation}(q_{tar})\| \\ f_3(\theta_i) = \sum_{i=1}^6 \lambda_i \sqrt{(\theta_i - \theta_{init})^2} \\ \sum_{i=1}^6 \lambda_i = 1 \\ i = 1, 2, \dots, 6 \end{cases} \quad (12)$$

Where all the variables in the formula have been introduced in the above description. Among them, the value of  $\lambda_i$  ( $i = 1, 2, \dots, 6$ ) is decreasing, i.e.,  $\lambda_1 > \lambda_2 > \lambda_3 > \lambda_4 > \lambda_5 > \lambda_6$ . The linear decline method is adopted in this calculation.

The proposed MOFOPSO algorithm analyzes the structure characteristics of the mechanical arm and the instability problem in the process of mechanical arm movement. Also, it analyzes the defects of traditional inverse kinematics algorithms, such as only considering location accuracy or precision of position and attitude. So, the proposed algorithm is a combination of theory and application and has certain practicability.

**TABLE 3 |** The initial and convergent values of different algorithms.

	Algorithm	Initial value	Convergent value	Iterative step	Gradient
Ackely	PSO	3.156917	0.017917	19	0.165211
	Adaption	2.699713	0.013031	16	0.167918
	Breed	2.800934	0.017496	17	0.163732
	Lin	2.852798	0.129628	17	0.160186
	MOFO	4.957359	0.050007	16	0.30671
Rastrigin	PSO	0.03943727	0.002264	7	0.0053105
	Adaption	0.04220217	0.000696	6	0.0069178
	Breed	0.04718385	0.001574	2	0.0228052
	Lin	0.13877033	0.007731	11	0.0119127
	MOFO	0.51498297	8.55E55E-02	10	0.0429484
Griewangk	PSO	17.48753	3.39924	10	1.408829
	Adaption	30.34479	3.49541	10	2.684937
	Breed	32.09457	8.848614	7	3.320851
	Lin	16.80333	9.048955	5	1.550875
	MOFO	56.26652	10.38	7	6.555218

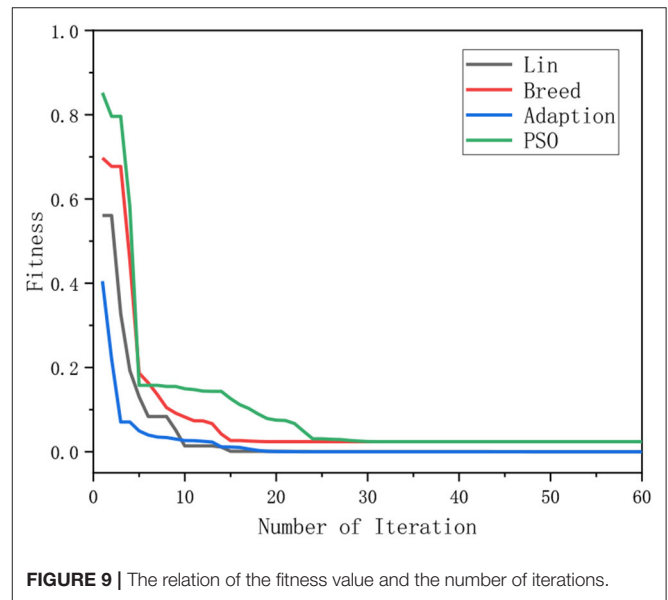
**TABLE 4 |** The relevant parameters of various algorithms.

PSO variants	Parameters
PSO	$c_1=2, c_2=2, w=0.5$
PSO_Adaptation	$c_1=2, c_2=2, w_{max}=0.8, w_{min}=0.2$
PSO_Breed	$c_1=2, c_2=2, w=0.5, b_c=0.8, b_s=0.05$
PSO_Lamda	$c_1=2, c_2=2, lamda=0.99$
PSO_Lin	$c_1=2, c_2=2, w_{max}=0.8, w_{min}=0.2$
PSO_Nature	$c_1=2, c_2=2, w=0.5$
MOFOPSO	$C_{max}=3, C_{min}=1.5,$ $w_{max}=0.8, w_{min}=0.2$ $(c_1=2.99, c_2=1.505, w=0.466)$

### ALGORITHM PERFORMANCE TEST

In this paper, six algorithms are taken for performance test, including the classical PSO algorithm, random weight PSO (RWPSO) (Zhan et al., 2009), linear decline PSO (LDPSO) (Lili et al., 2015), PSO algorithm based on hybridization (Lovbjerg et al., 2001), PSO algorithm based on natural selection (Yaming, 2019), and PSO algorithm based on simulated annealing. However, in the performance evaluation process, the PSO algorithms based on simulated annealing and natural selection achieve poor performance in convergence quality, speed, and stability. Thus, they are excluded from the evaluation of test function, convergence speed, and stability, while the performance of the rest algorithms is compared.

Although many articles (Yiyang et al., 2021) use many test functions to test the performance of PSO algorithm, but in this paper, MOFOPSO algorithm is proposed to overcome the drawbacks of poor accuracy, low efficiency. SO three test functions are chosen for the algorithm performance test, including Ackely function, Rastrigin function, and Griewangk function to investigate the shortcomings of classical PSO



**FIGURE 9 |** The relation of the fitness value and the number of iterations.

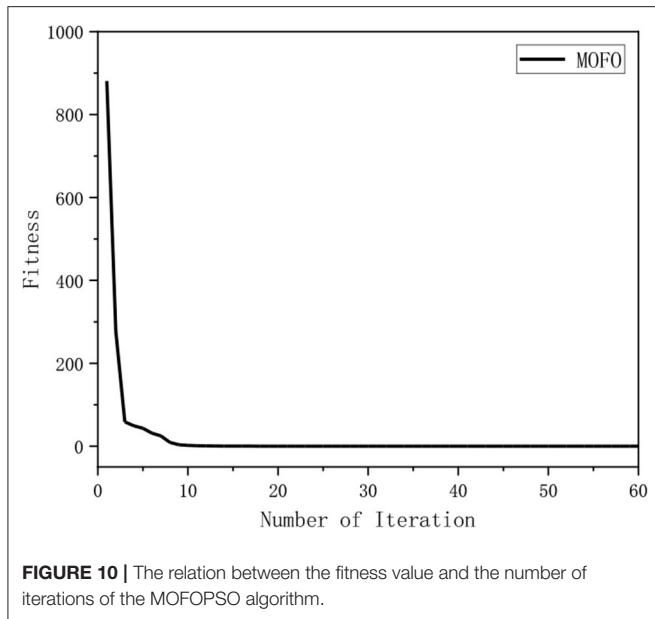
algorithms, such as slow convergence speed and falling into local optimum.

### Convergence Rate Test

Ackely function is a continuous function used to evaluate the convergence rate of an algorithm. It is obtained by superposing the exponential function with a moderately amplified cosine. As shown in Figure 3, the Ackely function presents the multi-directionality of a multi-dimensional point in the optimization process. Thus, the Ackely function is used to detect the global convergence rate of the algorithm.

Figure 4 demonstrates the results obtained by the six algorithms using the Ackely function. It indicates that the RWPSO and MOFOPSO algorithms achieve the fastest





convergence speed, while the MOFOPSO is better than the RWPSO at a close speed. From the vertical axis of the image, it can be seen that the fitness function value of the MOFOPSO algorithm rapidly decreases from the large value to the optimal value. This meets the design goal of balancing the global and local search capabilities of the algorithm.

### Convergence Test

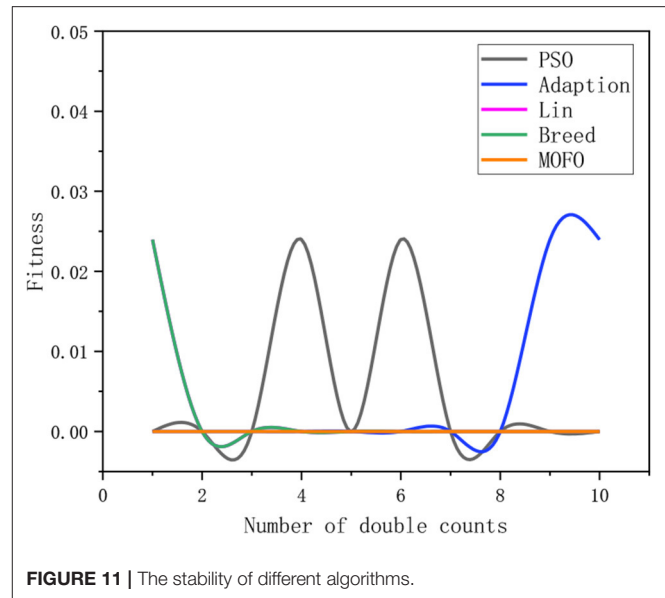
The Rastrigin function is a highly multimodal function used to test the ability of global optimization. This function is based on the De Jong function and adds a cosine modulation transfer function to generate frequent local minima. This paper makes good use of the Rastrigin function to test the practicability of the algorithm for a case with regular solutions. The figure of the Rastrigin function is illustrated in **Figure 5**.

**Figure 6** demonstrates the convergence of different algorithms. It can be seen that all the algorithms converge well, and the proposed MOFOPSO algorithm converges faster than the other algorithms with no shocks.

### Skip the Local Optimal Capability Test

The Griewangk function is used to detect the ability to jump out of the local optimum. The function changes with the quantity, and there are a large number of local extreme values in the real data distribution of the function. The image of the Griewangk function is illustrated in **Figure 7**. This function can be exploited to detect the convergence of the algorithm with a regular position of minimum value.

**Figure 8** demonstrates the abilities of the algorithms to escape from the local optimum. It can be seen that the multi-objective full-parameter optimization particle swarm



optimization algorithm can escape from the local optimum at a faster speed than other algorithms.

Above all, the three test functions, Ackely function, Rastrigin function, and Griewangk function, are all multi-modal functions, which are considered to be complex functions that are difficult for the optimization algorithm to solve. However, the proposed MOFOPSO algorithm can solve the complex multi-modal function optimization difficulty and performs better in terms of convergence speed, convergence, and the ability to escape from the local minimum value than other algorithms. And it should be noted that in the figure comparing the test functions (**Figures 6–8**), it can be seen that the initial value is the fitness function value. Due to our fitness function on the traditional PSO algorithm was improved, the mechanical arm joint Angle change values as a fitness function of a component, and the change of the mechanical arm joint Angle value compared to the position and posture error value is bigger, which makes the whole parameter optimization PSO to calculate the fitness function value is higher than other fitness function value of the algorithm.

Above all, we can obtain the initial and convergent values of different algorithms in **Table 3**. Although the initial value of the MOFOPSO algorithm is bigger than others, the gradient of the MOFOPSO algorithm shows that this algorithm has the fastest convergence speed.

## RESULTS DISCUSS

Based on the algorithm performance test, a 6-DOF manipulator from Japan Yaskawa Company is taken to simulate the solution to the inverse kinematics problem. Meanwhile, six algorithms are selected for performance comparison, and the simulation results of the MOFOPSO algorithm proposed in the article are

**TABLE 5** | The simulation results of different algorithms.

PSO variants	Parameters	Average fitness	Average time
PSO	$c_1=2, c_2=2, w=0.5$	0.004791622	3.3710435
PSO_Adaptation	$c_1=2, c_2=2, w_{max}=0.8, w_{min}=0.2$	0.004791622	5.0460895
PSO_Breed	$c_1=2, c_2=2, w=0.5, b_c=0.8, b_s=0.05$	0.002395811	3.8617253
PSO_Lamda	$c_1=2, c_2=2, lamda=0.99$	8.605051908	3.0658353
PSO_Lin	$c_1=2, c_2=2, w_{max}=0.8, w_{min}=0.2$	0.002395811	4.9974822
PSO_Nature	$c_1=2, c_2=2, w=0.5$	10.31912828	1.6046351
MOFOPSO	$C_{max}=3, C_{min}=1.5, w_{max}=0.8, w_{min}=0.2 (c_1=2.99, c_2=1.505, w=0.466)$	4.39E-09	2.6588259

**TABLE 6** | The angle values of the six joints of the manipulator calculated by different algorithms.

Algorithm	1-DOF	2-DOF	3-DOF	4-DOF	5-DOF	6-DOF
Standard value	-1.047198	-1.047198	1.047198	-0.785398	0.785398	0.523599
PSO	-0.49164228	-0.9547942	0.003137195	-1.42360713	1.275797884	1.306460089
PSO_Adaptation	-1.1199608	-1.58311273	0.63145572	-0.16697007	1.031499907	-0.33419752
PSO_Breed	-1.08357914	-1.31515517	0.839326647	-0.47618412	1.132207126	0.185101802
PSO_Lamda	0.557517444	-0.12569131	-0.71358997	1.027890172	-2.00106961	2.265754194
PSO_Lin	-1.71189772	-1.31515511	0.839326614	-0.47618413	0.673626419	0.643682534
PSO_Nature	-2.54977472	0	0	0	0	0
MOFOPSO	-1.04719681	-1.04719668	1.047197423	-0.78539912	0.785401078	0.523591577

analyzed to further verify the superiority of the performance of the algorithm.

## Parameter Setting Analysis

When the MOFOPSO algorithm is simulated, it is not necessary to set the relevant parameters of the algorithm but to input the parameters of the 6-DOF manipulator. However, for other algorithms, it is necessary to set the relevant parameters (inertia weight, learning factor, or one of both) to the empirical values during simulation. Thus, in terms of parameter setting, the MOFOPSO is more practical, effective, and intelligent than other algorithms. The relevant parameters of various algorithms are listed in **Table 4**.

In this process, it is found that the optimal value of the inertia weight  $w$  is 0.466 rather than the empirical value of 0.5. Also, the best value of the learning factor is  $c_1 = 2.99$  and  $c_2 = 1.505$  rather than the empirical value and  $c_2 = 2$ . Thus, it shows that the MOFOPSO algorithm designed in this paper achieves good convergence and efficiency when it is applied to solve the IK of the 6-DOF manipulator.

## Convergence Rate Analysis

In the simulation, the relationship between the fitness and the number of iterations for solving the IK of the 6-DOF manipulator using different algorithms is illustrated in **Figure 9**. It indicates that the MOFOPSO algorithm converges very fast and finds the optimal solution successfully. Due to the large difference in the order of the fitness function value of different algorithms, the convergence speed of the MOFOPSO algorithm is not obvious. Thus, the simulation results of the MOFOPSO algorithm are

extracted separately, which are shown in **Figure 10**. It shows that the MOFOPSO reaches convergence after about nine iterations, indicating a very fast convergence speed of the MOFOPSO algorithm.

It should be noted that the initial value (in **Figures 9, 10**) is the fitness function value, the relation between the fitness value and the number of iterations of these algorithms can indicate the convergence speed of the algorithm (Serkan, 2021).

## Stability Analysis

It is acknowledged that stability is a prerequisite for the operation of an algorithm. If there is no guarantee of stability, it is meaningless to discuss the convergence of the algorithm. Therefore, this paper conducts experiments on the stability of the algorithms. **Figure 11** shows the average value of the fitness function obtained by applying the algorithms to solve the IK of the 6-DOF (each algorithm is run 10 times consecutively). It indicates that the average fitness value of the classical PSO algorithm and RWPSO algorithm fluctuates greatly, indicating the poor stability of these two algorithms. In contrast, the average fitness value of the MOFOPSO algorithm is basically unchanged, indicating that the algorithm has good stability.

## Synthetic Simulation Results Analysis

**Table 3** lists the comprehensive results of solving the IK of the manipulator using seven different algorithms including MOFOPSO. By analyzing the data in **Table 5**, the PSO algorithms based on natural selection and simulated annealing obtain the worst accuracy and large minimum values of the fitness function. The precision of the classical PSO algorithm, RWPSO

algorithm, LDPSO algorithm, and hybridization-based PSO is all better. The MOFOPSO algorithm proposed in this paper achieves the highest precision, and it improves the fitness value to six orders of magnitude compared with other algorithms. In terms of solving speed, the proposed MOFOPSO algorithm is the fastest, about 0.7 s faster than the fastest one among other algorithms with the same accuracy, and the speed is increased by about 20%.

In summary, **Table 6** shows the angle values of the six joints of the manipulator calculated by different algorithms. The standard value is the joint Angle value of the manipulator obtained by mathematical derivation. It can be seen from the calculation results that the full-parameter optimization multi-objective particle swarm optimization algorithm proposed in this paper has the highest accuracy.

## CONCLUSION

In this paper, the problem of solving the inverse kinematics of the manipulator is analyzed firstly, and the advantages and disadvantages of the existing solution algorithms are summarized. Then, the PSO algorithm based on the principle of multi-objective optimization is selected to solve the IK of the manipulator. Next, the MOFOPSO algorithm is proposed and realized. Finally, to obtain a better inverse kinematics solution of the 6-DOF manipulator, this paper tests the

performance of MOFOPSO with three typical functions. Also, six algorithms are taken for performance comparison. The experimental results indicate that the proposed method achieves good generalization ability, convergence speed, and accuracy.

## DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding authors.

## AUTHOR CONTRIBUTIONS

DL planned the study design. YH collected the data. SL and DC conducted experiments, analyzed the data, and wrote the paper. All authors contributed to the article and approved the submitted version.

## FUNDING

This work was supported by the Taishan Scholar Project of Shandong Province (ts20190937) and Universities Serving Qingdao Industrial Development Science and Technology Project.

## REFERENCES

- Angeles, J., and Angeles, J. (2002). *Fundamentals of Robotic Mechanical Systems*. New York, NY: Springer-Verlag.
- Ayız, C., and Kucuk, S. (2009). "The kinematics of industrial robot manipulators based on the exponential rotational matrices," in *IEEE International Symposium on Industrial Electronics*, 977–982 doi: 10.1109/ISIE.2009.5222601
- Buss, S. R. (2004). Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE J. Robot Automat.* 17:16.
- Charnes, A., Cooper, W. W., and Ferguson, R. O. (1955). Optimal estimation of executive compensation by linear programming. *Manage. Sci.* 1:138. doi: 10.1287/mnsc.1.2.138
- Craig, J. J. (2009). *Introduction to Robotics: Mechanics and Control*. Pearson Education.
- Danavit, J., and Hartenberg, R. (1964). *Kinematic Synthesis of Linkages*. New York, NY: McGraw-Hill.
- Dereli, S., and Koker, R. (2018). IW-PSO approach to the inverse kinematics problem solution of a 7-DOF serial robot manipulator. *Sigma J. Eng. Nat. Sci.* 36, 77–85.
- Eberhart, R. C., and Kennedy, J. (1995). "Particle swarm Optimization," in *Proceedings of IEEE International Conference on Neural Networks*, 1942–1948 doi: 10.1109/ICNN.1995.488968
- Feng, Q., Li, Q., Quan, W., and Pei, X. M. (2020). Overview of multi objective particle swarm optimization algorithm. *J. Eng. Sci.* 43.
- Funda, J., and Paul, R. P. (1990). A computational analysis of screw transforms in robotics. *IEEE Transact. Robotics Automat.* 6, 348–356 doi: 10.1109/70.56653
- Hasan, A. T., Hamouda, A. M. S., Ismail, N., and Al-Assadi, H. M. A. A. (2006). An adaptive-learning algorithm to solve the inverse kinematics problem of a 6 D.O.F serial robot manipulator. *Adv. Eng. Softw.* 37, 432–438 doi: 10.1016/j.advengsoft.2005.09.010
- He, X. M., and Shao-hua, D. (2019). Research on rush order insertion rescheduling problem under hybrid flow shop with multi-objective and multi-constraint. *Chin. J. Eng.* 41:1450. doi: 10.13374/j.issn2095-9389.2018.11.27.002
- Husty, M. L., Pfurner, M., and Schrcker, H. P. (2007). A New and efficient algorithm for the inverse kinematics of a general serial 6R66R manipulator. *Machine Theory* 42, 66–81. doi: 10.1016/j.mechmachtheory.2006.02.001
- Idris, S., Burak, T., and Ulutas, M. (2018). Inverse kinematics application on medical robot using adapted PSO method. *Eng. Sci. Tech. Int. J.* 21, 1006–1010. doi: 10.1016/j.jestch.2018.06.011
- Khaleel, H. Z. (2018). Inverse kinematics solution for redundant robot manipulator using combination of GA and NN. *Al-Khwarizmi Eng. J.* 14, 136–144. doi: 10.22153/kej.2018.10.008
- Kong, X. R., Qin, Y. L., and Luo, W. B. (2010). Model updating of aluminum honeycomb sandwich plate based on particle swarm optimization algorithm with flying factor. *J. Computer Appl.* 30, 682–683. doi: 10.3724/SP.J.1087.2010.00786
- Kucuk, S. (2012). Simulation and design tool for performance analysis of planar parallel manipulators. *Simulation* 88, 542–556. doi: 10.1177/0037549711414157
- Li, F., Liu, J. C., and Shi, H. T. (2017). Multi-objective particle swarm optimization algorithm based on decomposition and differential evolution. *Control Decis.* 32:403. doi: 10.13195/j.kzyjc.2016.0186
- Lili, D., Miao, Y., and Shuqin, X. (2015). Optimization of projection tracking model and application of linear decreasing weight particle group. *Eng. J. Drainage Irrigat. Machinery* 33, 1044–1048. doi: 10.969/j.issn.1674-8530.15.0112
- Liu, Q., Liu, Q., Yang, J. P., Zhang, J. S., Gao, S., and Li, Q. D. (2020). Progress of research on steel making-continuous casting production scheduling. *Chin. J. Eng.* 42:144. doi: 10.13374/j.issn2095-9389.2019.04.30.002
- Lovbjerg, M., Rasmussen, TK., and Krink, T. (2001). "Hybrid particle swarm optimizer with breeding and subpopulations," in *Proceedings of the 3rd Genetic and Evolutionary Computation Conference* (Sanfrancisco, CA), 469–476.

- Ma, C., Zhang, Y., Cheng, J., Wang, B., and Zhao, Q. J. (2016). "Inverse kinematics solution for 6R66R serial manipulator based on RBF neural network," in *International Conference on Advanced Mechatronic Systems* (Melbourne, VIC). doi: 10.1109/ICAMechS.2016.7813473
- Momani, S., Abo-Hammour, Z. S., and Alsmadi, O. M. K. (2016). Solution of inverse kinematics problem using genetic algorithms. *Appl. Math. Informat. Sci.* 10:225. doi: 10.18576/amis/100122
- Murray, R. M. (2017). *A Mathematical Introduction to Robotic Manipulation*. Boca Raton: CRC press (2017).
- Nizar, R., and Alimi, M. A. (2013). Inverse kinematics using particle swarm optimization, a statistical Analysis. *Sci. Direct.* 64, 11602–1611. doi: 10.1016/j.proeng.2013.09.242
- Raghavan, M., and Roth, B. (1993). Inverse kinematics of the general 6R66R manipulator and related linkages. *J. Mech. Design.* 115, 502–508. doi: 10.1115/1.2919218
- Serdar, K., and Baris Doruk, G. (2016). "Inverse kinematics solution of a new hybrid robot manipulator proposed for medical purposes," in *Medical Technologies National Congress* (TIPTEKNO). doi: 10.1109/TIPTEKNO.2016.7863076
- Serkan, D., et al. (2021). Strengthening the PSO algorithm with a new technique inspired by the golf game and solving the complex engineering problem. *Complex Intelligent Syst.* 7, 1515–1526. doi: 10.1007/s4s0747-021-002922-2
- Shi, J. P., Liu, P., and Chen, D. Y. (2020). Inverse kinematics solution of redundant manipulator based on improved particle swarm optimization algorithm. *Mech. Transm.* 2, 69–75.
- Shi, Y., and Eberhart, R. C. (1998a). "A modified particle swarm optimizer," in *Proceedings of IEEE International Conference on Evolutionary Computation Piscataway* (Berlin: IEEE Press), 69–73.
- Shi, Y., and Eberhart, R. C. (1998b). "Parameter selection in particle swarm optimization," in *Proceedings of 7th International Conference on Evolutionary Programming VII* (Berlin: Springer-Verlag Press), 591–600. doi: 10.1007/BFb0b040810
- Tanev, T. K. (2000). Kinematics of a Hybrid (Parallel-Serial) robot manipulator. *Mech. Mach. Theory.* 35, 1183–1196. doi: 10.1016/S05094-114X44X(99)000733-7
- Tseng, C. H., and Lu, T. W. (1990). Minimax multi objective optimization in structural design. *Int. J. Numer. Methods Eng.* 30:1213. doi: 10.1002/nme.1620300609
- Xie, C. F., Wang, Q., and Cheng, Y. (2006). *Engineering Mechanics-Statics*. Higher Education Press.
- Xi-Hu, Z., et al. (2011). Particle swarm optimization with time factor. *Basic Sci. J. Textile Univers.* 24, 303–308. doi: 10.13338/j.issn.1006-8341.2011.02.003
- Yaming, R. (2019). Pid parameter optimization based on the natural selection particle group algorithm. *Volkswagen Tech.* 21, 1–3.
- Yiyang, L., Xi, J., Hongfei, B., Zhining, W., and Liangliang, S. (2021). A general robot inverse kinematics solution method based on improved PSO algorithm. *IEEE Access.* 9, 323411–32350. doi: 10.1109/ACCESS.2021.3059714
- Zadeh, L. A. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Trans. Autom. Control* 8:59. doi: 10.1109/TAC.1963.1105511
- Zhan, Z. H., Jun, Z., Yun, L., and Henry, S. (2009). *Adaptive Particle Swarm Optimization. Transactions on Systems Man and Cybernetics Society*. Piscataway: IEEE Press 362–381.

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

**Publisher's Note:** All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Copyright © 2022 Luo, Chu, Li and He. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.