IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 31, 2021, accepted February 7, 2021, date of publication February 16, 2021, date of current version March 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3059714

A General Robot Inverse Kinematics Solution Method Based on Improved PSO Algorithm

LIU YIYANG[1,2,3], XI JIALI[2,3,5], BAI HONGFEI[2,3], WANG ZHINING[2,4], AND SUN LIANGLIANG[5]

[1]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, Shenyang 110016, China
[2]Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[3]Institutes for Robotics and Intelligent Manufacturing, Chinese Academy of Sciences, Shenyang 110169, China
[4]School of Information Science and Technology, Department of Automation, University of Science and Technology of China, Hefei 230026, China
[5]Information and Control Engineering Department, Shenyang Jianzhu University, Shenyang 110168, China

Corresponding author: Liu Yiyang (sialiuyiyang@sia.cn)

This work was supported in part by the National Key R&D Program of China under Grant 2018YFB2003203, in part by the National Natural Science Foundation of China 6187317, in part by the Revitalizing Liaoning Outstanding Talents Project under Grant XLYC1907057, in part by the Liaoning Province Education Department Scientific Research Foundation of China under Grant LQGD2019014, and in part by the Liaoning Provincial Natural Science Foundation of China under Grant 2019-ZD-0218.

ABSTRACT Robots whose geometric structure does not meet the Pieper criterion are called general robots. For the inverse kinematic operation of general robots, the closed solution method cannot be solved, and the numerical solution calculation amount is too large and the singular position cannot be calculated. To solve this problem, this paper proposes an inverse kinematics calculation method based on improved particle swarm optimization (PSO) algorithm and applicable to general robots. In order to avoid the particle update rate not adapting to each stage of the optimization process, a nonlinear dynamic inertia weight adjustment method based on the concept of similarity is introduced, so that the search process is more robust; in addition, to overcome the problem of local optimal solution At the same time, multiple populations are introduced to perform optimization search at the same time, and the immigration operator is proposed to increase the diversity of the particle population in the iteration. This article uses Comau NJ-220 robot for test verification, compared with the original PSO and multi-subswarm algorithm, the results show that the proposed improved PSO has higher algorithm stability for general robot kinematics inverse solution problems, and can greatly improve the convergence accuracy and speed. This method provides a new solution to the field of robot inverse kinematics, and provides a more efficient and stable kinematics foundation for robot motion planning.

INDEX TERMS Robot kinematics analysis, PSO algorithm, robot inverse kinematics, artificial intelligence.

I. INTRODUCTION

Inverse kinematics solution is to calculate the angle value of each joint according to the pose of the rectangular coordinate space of the end effector of the robot. Robot inverse solution operation has always been one of the key problems of robot kinematics, and it has important significance for the research work of robot dynamic characteristics analysis, path and motion planning, and motion control. Conventional algorithms for inverse kinematics operations include: closed solutions, numerical solutions, and intelligent algorithms.

For industrial robots, a closed solution can only be obtained if its geometry meets the Pieper criterion [1]. The closed solution method can be divided into two types: algebraic method and geometric method. These two methods can directly derive the joint angle expression according to the formula or geometric configuration, so its calculation speed is fast, and all possible inverse solutions can be solved. However, some robot configurations do not meet the Pieper criterion. At this time, the robot is a general robot [2], and the inverse solution cannot be obtained using the closed solution method. Generally, robots have various geometric structures and are not limited to traditional configurations of industrial robots. In addition, even if the industrial robot theoretically satisfies the Pieper criterion, structural distortions caused by factors such as

The associate editor coordinating the review of this manuscript and approving it for publication was Jagdish Chand Bansal.

assembly errors and processing environment may cause the criterion to fail. At this time, the inverse kinematics solution can only use numerical solutions or intelligent algorithms.

Numerical solutions mainly include iterative algorithms, elimination methods and extension methods. At present, the commonly used numerical solution is an iterative algorithm, which calculates the Jacobian matrix of the manipulator, and then iteratively calculates the exact solution by giving the initial value. This kind of algorithm is computationally intensive, and the Jacobian matrix is singular at a singular position, so it cannot be inverted. In this case, the numerical solution has no solution.

In order to realize the efficient inverse kinematics operation of general robots at any position, in recent years, intelligent algorithms have been widely used. The main idea is to convert robot kinematics equations into optimal control problems to solve. Such algorithms mainly include particle swarm optimization (PSO) [3], genetic algorithm, annealing algorithm and neural network algorithm[4], Artificial Bee Colony (ABC) Algorithm, krill herd, monarch butterfly optimization (MBO) [5], earthworm optimization algorithm (EWA), elephant herding optimization (EHO), moth search (MS) algorithm, Slime mould algorithm (SMA), and Harris hawks optimization (HHO), etc [6], [7]. The accuracy of the neural network algorithm depends more on the structure of the network, and the selection of the structure of the BP neural network has not yet been a unified and complete theoretical guidance, generally only by experience. The structure of the network directly affects the approximation ability and generalization property of the network. Therefore, in terms of algorithm structure, neural network is not suitable for solving inverse kinematics which requires high accuracy. In recent years, researchers and scholars in related fields have studied the application of genetic algorithms to inverse kinematics problems more and more. Under the specific requirements of general robots in this paper, this algorithm is not as practical as PSO algorithm [8]. With information sharing mechanism, PSO updates its own new search direction based on its own historical flight experience; however, genetic algorithm uses crossover operation on individual chromosomes and then mutually Compared with PSO, the sharing efficiency of this method is relatively low, and the feedback information of the network is not used in time. Therefore, the search speed of the algorithm is relatively slow, and it needs more training time to get more accurate solutions. Compared with other intelligent algorithms, PSO is intuitive and efficient, simple to implement and has high execution efficiency, which can avoid tedious mathematical formula derivation and parameter selection. Therefore, PSO algorithm is more suitable for robot inverse kinematics.

Ram *et al* [9] proposed a two-way PSO method to solve the inverse kinematics of multi-degree-of-freedom robots, and more quickly realized robot trajectory planning and obstacle avoidance optimization control; Xie *et al* [10] combined differential evolution algorithm and proposed A robot inverse kinematics differential adaptive chaotic particle swarm solution method, which improves the stability of the algorithm; Zhu *et al* [11] proposed an adaptive PSO to solve the inverse kinematics of the robot arm, using forward kinematics Based on the equation, the inverse kinematics solution of the redundant manipulator is transformed into an equivalent minimum problem. However, similar to other heuristic algorithms, PSO is not ideal for solving high-dimensional, non-linear and multi-peak problems. When the degree of freedom of the manipulator increases, it is easy to fall into local optimum or stagnation. The convergence accuracy and The convergence rate is difficult to guarantee, and improved algorithms are necessary.

This paper proposes A General Robot Inverse Kinematics Solution Method Based on Improved PSO Algorithm. The contribution of this paper can be summarized as follows: first, this article explores the mechanical structure of ROKAE XB6 six-degree-of-freedom general manipulator, and then selects the DH modeling strategy to establish the kinematics model of the manipulator. According to the principle of space coordinate conversion, the forward motion equation of the manipulator is derived. Then, the inverse kinematics problem is transformed into an optimal solution problem, and a solution method for inverse kinematics of a manipulator based on an improved PSO is proposed according to the characteristics of general manipulators. The specific algorithm improvements are as follows. The method is based on basic particle swarms. To solve this problem, this paper proposes an inverse kinematics calculation method based on improved PSO algorithm and applicable to general robots. In order to avoid the particle update rate not adapting to each stage of the optimization process, a nonlinear dynamic inertia weight adjustment method based on the concept of similarity is introduced, so that the search process is more robust; in addition, to overcome the problem of local optimal solution At the same time, multiple populations are introduced to perform optimization search at the same time, and the immigration operator is proposed to increase the diversity of the particle population in the iteration. Finally, experiments are performed to verify the feasibility and superiority of the algorithm.

The rest of this paper is organized as follows. Section II gives the system model. Section III describes the improved PSO algorithm for the inverse solution. Section IV gives experiments and the final section gives the conclusion.

II. DESCRIPTION OF INVERSE KINEMATICS PROBLEM

This article takes the Comau NJ-220 robot as an example to carry out research. The robot is a six-degree-of-freedom articulated robotic arm. The robot kinematics is modeled by the D-H method. Due to assembly errors, the robot has an axial offset of 5d in the 5Z direction at the joint 5, resulting in deviations in its D-H parameters, and the axes of the three axes at the end do not intersect. Therefore, the Pieper criterion is not satisfied, and there is no closed solution. At this time, the robot is a general robot, the coordinate system at each joint

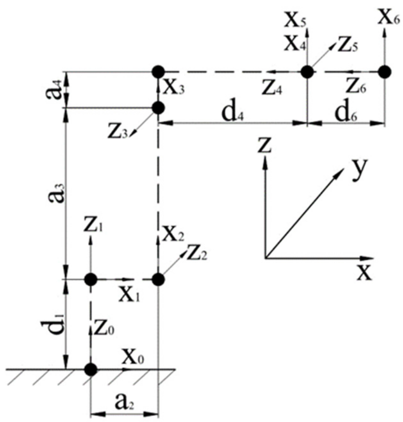


FIGURE 1. Comau NJ-220 robot link coordinate system.

TABLE 1. D-H parameters of comau NJ-220 robot.

NO.	Connecting rod length a_{i-1}(mm)	Connecting rod torque α_{i-1} (deg)	Joint offset d_i(mm)	Joint angle boundary θ_i(rad)
1	0	0	830	[-2.9，2.9]
2	400	-90	0	[-1.57，1.57]
3	1175	180	0	[-1.57，1.57]
4	250	-90	-1125	[-3.14，3.14]
5	0	-90	10	[-3.14，3.14]
6	0	90	-230	[-3.14，3.14]

is established as shown in Figure 1, and the link parameters are shown in Table 1.

The length a_i $(i = 1, 2, \ldots, 6)$ is the distance along the X_{i-1} axis from the Z_{i-1} axis to the Z_i axis, and the torsion angle α_i is along the X_{i-1} axis. The angle from the Z_{i-1} constantly being proposed. Das [12] and other evolutionary operators inherited through the concept of governance in human society and the multiple crossovers and bee colony operators inherited by genetic algorithms have improved the intensive ability of PSO. Enhance the intensive ability of PSO. Mahya [13] and others proposed a hybrid algorithm of fireflies and improved particle swarm optimization (IPSO) to achieve a better average load to develop and improve important indicators, effective resource utilization and task response time. Arfan [14] *et al.* proposed an improved self-inertia weighted adaptive PSO algorithm with gradient-based local search strategy (SIW-APSO-LS). The new algorithm strikes a balance between the improved inertial weight adaptive PSO exploration ability and the development of gradient-based local search strategy, which is updated through the evolution process to enable each particle to iteratively increase its speed and position. Although the above documents have improved the PSO algorithm, due to the structure of the algorithm itself, the local convergence problem has not been significantly improved, and the above documents have limited effects on the convergence accuracy and convergence speed improvement, and are not suitable for high-dimensional inverse motion Learn to solve.

This paper proposes an inverse kinematics solution method based on improved PSO for a six-axis articulated general manipulator. Compared with closed solution and numerical solution, the algorithm has a simple structure, does not need to perform matrix inverse operation and trigonometric function operation, and can accurately solve the general robot inverse kinematics problem of any degree of freedom; compared with the traditional PSO, the algorithm is more effective Solve the problem of local convergence, and improve the accuracy and speed of convergence axis to the Z_i axis, the offset d_i is the distance moved from the X_{i-1} axis to the X_i axis along the Z_i axis, and the joint angle θ_i is the X_{i-1} along the Z_i axis The rotation angle of the axis to the X_i axis.

It can be seen from the above conditions that the transformation matrix (the i-th coordinate system to the $(i+1)$th coordinate system) between adjacent coordinate systems is

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

By successively multiplying the transformation matrix, the transformation relationship from the end effector coordinate system to the robot base coordinate system can be obtained:

$$ {}^0_6T = \prod_{i=1}^{6} {}^{i-1}_{\ \ i}T = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} = P(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6) = \begin{pmatrix} R_{3\times3} & P \\ 0 & 1 \end{pmatrix} \quad (2)$$

Among them, a, o, n describe the three unit vectors of the robot end effector: the approach vector a is the direction of the end effector into the object; the direction vector o is the direction from one position to the next position; the normal vector n is The other two vectors constitute the right-hand rule. p_x, p_y, p_z are the three components of the end effector position vector. $P(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6)$ is the posture matrix of the robot arm, which is obtained by multiplying the transformation matrix between the adjacent joint coordinate systems of the robot. $R_{3\times3}$ is the 3×3 rotation matrix (ie attitude matrix) of the robot end effector coordinate system relative to the robot base coordinate system, and p is the position vector of the robot end effector coordinate system under the robot base coordinate system.

Known robot arm expected pose matrix

$$P_{des} = \begin{pmatrix} R_{3\times3,des} & P_{des} \\ 0 & 1 \end{pmatrix} \quad (3)$$

By substituting the current joint angle value into the forward kinematics equation of the robot arm, the current pose matrix can be obtained

$$P_{cur} = \begin{pmatrix} R_{3\times3,cur} & P_{cur} \\ 0 & 1 \end{pmatrix} \quad (4)$$

In order to make the robot arm reach the desired posture, it is necessary to satisfy the following constraints by formula (3) and formula (4)

$$\|P_{cur} - P_{des}\| = 0 \tag{5}$$

That is, the current pose matrix and the expected pose matrix are equal, and the joint angle value that meets the condition at this time is the inverse kinematics solution. The boundary conditions of each joint variable can be expressed by the following formula

$$\theta_k^l \leq \theta_k \leq \theta_k^u k = 1, 2, \ldots \tag{6}$$

Among them, θ_k^l and θ_k^u represent the lower and upper boundaries of the joint variable corresponding to the k-th joint axis, respectively. At this point, the inverse kinematics problem is decomposed into a nonlinear transcendental vector equation and a constraint equation. The inverse kinematics solution is transformed to solve the following optimal control problem [15]

$$min \ \|P_{cur} - P_{des}\| \tag{7}$$

$$s.t. \ \theta_k^l \leq \theta_k \leq \theta_k^u k = 1, 2, \ldots \tag{8}$$

As the objective function of the optimization control problem, it will be substituted into the PSO for solving.

III. IMPROVED PSO ALGORITHM FOR INVERSE SOLUTION

PSO is an adaptive evolutionary computing technology based on population search. It was proposed by Eberhart and Kennedy in 1995, and it was inspired by biological sociologists' research on bird predation behavior. The basic idea of PSO is to find the optimal solution through cooperation and information sharing among individuals in the group [16]. Although the traditional PSO method can be used to solve multi-peak and high-dimensional control optimization problems, the traditional PSO algorithm increases the amount of calculation and decreases the stability when the number of variables and dimensions increase, and ignores the parallel mechanism of the algorithm Potential. The flow chart of the traditional PSO is shown in Figure 2. Therefore, this algorithm is difficult to meet the requirements of robot control system for algorithm accuracy and stability. In view of the above problems, this paper makes improvements on the basis of PSO, and proposes a kinematics inverse solution operation method for general robots. This method specifically improves the PSO from the following three aspects.

A. NONLINEAR DYNAMIC INERTIA WEIGHT.

In order to avoid the particle update rate not adapting to each stage of the optimization process [17], a nonlinear dynamic inertia weight adjustment method based on the concept of similarity is introduced, which makes the search process more robust.

In the traditional PSO algorithm, the inertia weight is a fixed value. If the value is too large, it is not conducive to the small-scale local search in the later stage of the particle. If the value is too small, it is not conducive to the global search. In response to the above defects, scholars represented by Kennedy have proposed a WPSO algorithm that changes the inertia weight changes by linear decreasing laws [18]. The change formula of weight w is as follows:

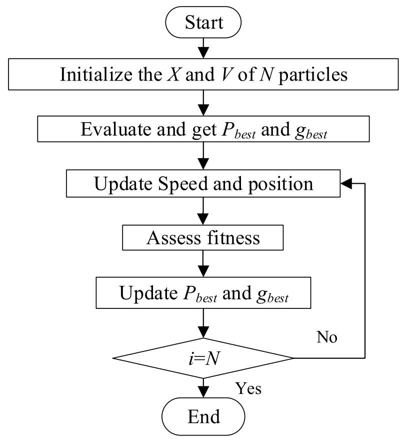


FIGURE 2. The flow of the conventional PSO.

$$w = w_{max} - \frac{(w_{max} - w_{min})}{maxgen} \cdot i \tag{9}$$

Among them: w_{max} is the maximum weight; w_{min} is the minimum weight; $maxgen$ is the maximum number of iterations; i is the current number of particle iterations. The effect of changing the weight linearly has been improved, but the particle speed update still has a strong dependence on the inertia weight. Using the concept of similarity, a method of nonlinear dynamic weight adjustment is introduced [19]. The formula for defining the similarity between particle p and particle q is as follows:

$$S(p, q) = 1 - [\frac{d(p, q)}{d_{max}}]^2 \tag{10}$$

Among them: $d(p, q)$ represents the Euclidean distance between the p-th particle and the q-th particle; dmax represents the maximum distance of the space particles, which can be approximately equal to the maximum side length of the map. When $d(p, q) \to 0$, $s(p, q) = 1$; when $d(p, q) \to d_{max}$, $s(p, q) = 0$, and $s(p, q) \subset (0, 1)$. It can be seen from Equation 10 that when the distance between the two particles is closer, the similarity is greater; when the distance between the two particles is longer, the similarity is smaller.

Suppose the similarity value between the p-th particle and the optimal particle g_{best} of the current group is $s(p, g_{best})$. According to Equation 10, when $s(p, g_{best}) = 0$, the distance between the particle and the optimal particle of the current group is the largest, so the particle needs to jump out of the current area, then the inertial weight of the particle in the next iteration should take w_{max}; when $s(p, g_{best}) = 1$, the particle is near the optimal particle of the current group, and the particle needs to be searched finely near the current small range. At this time, the inertia weight of the particle in the next iteration should be taken as w_{min}. When $s(p, g_{best}) \subset (0, 1)$, the inertia weight is updated as follows according to the change of similarity.

$$w_i = w_{min} + (w_{max} - w_{min}) \cdot (1 - s(p, g_{best})) \cdot \sqrt{\frac{maxgen - i}{maxgen}} \tag{11}$$

Among them: w_{max} is the maximum weight; w_{min} is the minimum weight; $maxgen$ is the maximum number of iterations; i is the current number of iterations; w_i is the value obtained by the current iteration.

The similarity-based dynamic inertia weight adjustment method enables particles to dynamically adjust the evolution speed according to the difference from the global particle position, and the update rate can be adapted to each stage of the optimization process, thereby overcoming the weight of the traditional PSO. The disadvantages of inappropriate value and poor search ability.

B. MULTI-PSO WITH IMMIGRATION OPERATOR

In order to overcome the problem of local optimal solution in the search process, the standard PSO algorithm breaks through the single PSO mode [20], parallel optimization search is introduced, and the multi PSO is proposed to replace the conventional PSO.

Multi-particle swarm breakthrough The standard PSO only relies on the framework of a single population for genetic evolution, introducing multiple particle swarms for simultaneous optimization search; different populations are given different control parameters to increase the diversity of particle swarms [21]. Various particle swarms are connected through immigration operator [22] to realize the co-evolution of multi-particle swarms; the optimal individuals in each evolutionary generation of various swarms are saved, and the optimal solution is obtained by the co-evolution of multiple particle swarms Synthesizing the results also serves as the basis for judging the convergence of the algorithm. Figure3 shows the structure of the multi-PSO.

In recent years, the multi-PSO has greatly enriched the diversity of individuals and improved the global convergence ability of the algorithm. It is mostly used to solve multi-peak and high-dimensional control optimization problems [23]. However, multiple particle swarms converge when computing in parallel. The speed is unstable and the convergence accuracy is not high and the calculation time is long. Therefore, this paper innovatively proposes immigration operators to improve the speed and accuracy of multi-PSO.

This paper draws on the concept of immigration operator in genetic algorithm and introduces it into PSO for the first time, which increases the diversity of particle population in iteration. During the operation of multiple particle swarms, the various swarms are relatively independent, and are connected to each other through immigration operators. The immigration operator regularly introduces the optimal individuals appearing in the iterative process of various groups into other groups to realize the exchange of information

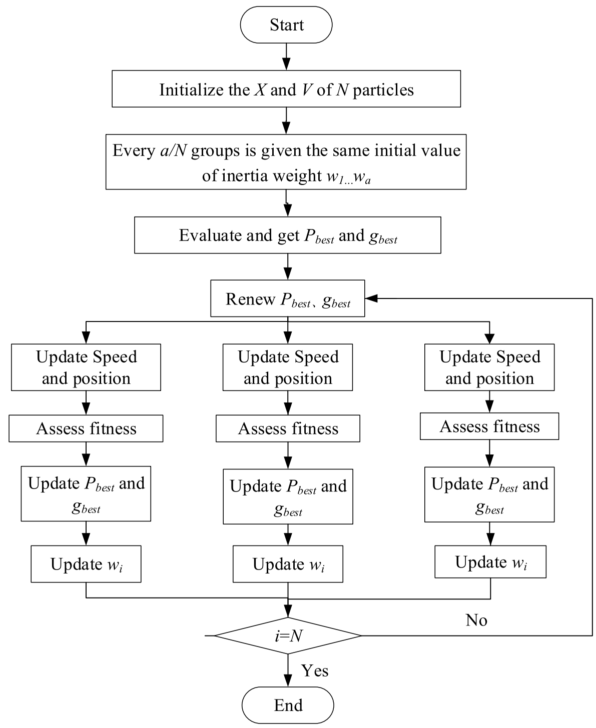


FIGURE 3. Schematic diagram of multi-PSO.

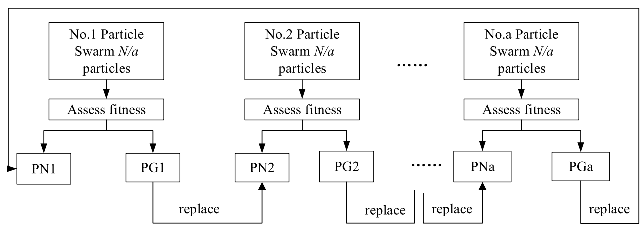


FIGURE 4. Individual replacement process.

between particle groups. In the iterative process, the best and worst individuals in each population are selected, and the immigration operator is used to replace the worst individuals in the latter particle group with the best individuals in the previous particle group according to the particle group number. In order to ensure the closedness of the algorithm, the worst individual in the first particle swarm is replaced with the best individual in the last particle swarm. Specific individual replacement [24] method flow, as shown in the figure 4.

The ability of the migration operator is strengthened here, that is, every certain algebra, all populations are sorted according to the optimal fitness value in each population, thereby selecting the optimal population and the worst population, and replacing the worst population It is the optimal population, as shown in Figure 5 This operation is placed before the individual replacement operation to optimize the evolution direction of the algorithm, which can improve the accuracy and speed of convergence.

At each generation in the iterative process, the optimal individual of each particle swarm is selected to be saved and

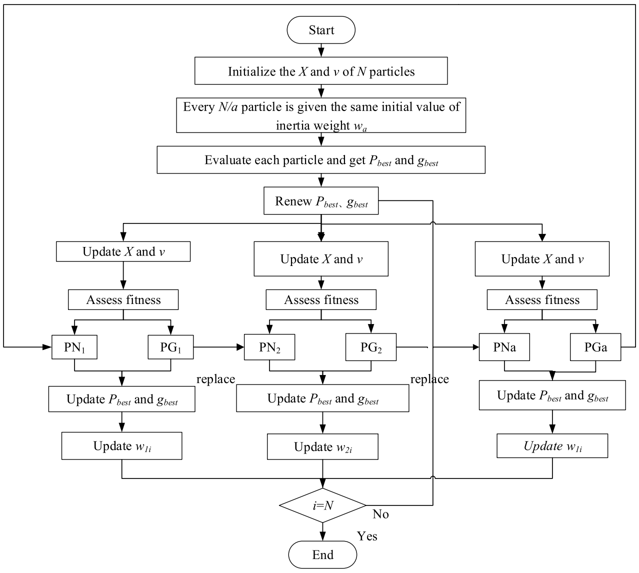


FIGURE 5. Structure diagram of improved PSO.

continuously updated to ensure that the optimal individual is not destroyed and lost, and to achieve the co-evolution of multi-particle swarm [25], the acquisition of the optimal solution It is a comprehensive result of the coordinated evolution of multiple particle swarms.

The inverse kinematics solution based on the improved PSO is divided into 8 steps, which are described as follows:

1) As with the basic PSO, first initialize the parameters of the N particle swarm, the speed and position of the particles, and their maximum and minimum values.
2) Find and update the individual and global extreme values of the group.
3) Divide the N particle population into n particle populations, and perform the optimization search simultaneously.
4) Screen out the best and worst individuals in each population, and use the immigration algorithm to replace the worst individuals in the latter group with the best individuals in the former group. In order to ensure the closedness of the algorithm, the worst individual in the first population is replaced with the best individual in the last population.
5) Recalculate the global optimal value and individual extremum of the new generation group.
6) First determine the similarity between each particle and the optimal particle of the group, then find the corresponding inertia weight value w, and finally update the position of the particle.
7) When the number of iterations of the algorithm is greater than or equal to the maximum number of iterations, end the algorithm, otherwise jump to step (2) to continue the loop operation.
8) After the iteration is over, the global best advantage is output.
9) The evolution is over and the optimal solution is output.

IV. EXPERIMENTAL VERIFICATION

The experimental verification is carried out for Comau NJ-220 general robots. The computer CPU used is Intel (R) Core (TM) i5-10210U, the memory is 8GB, and the simulation program is written in Matlab2013a. Two comparative experiments were designed in Matlab environment. The first group used 4 test functions to simulate with traditional PSO algorithm and improved PSO algorithm to verify the performance of improved PSO algorithm in high-dimensional space optimization problem. Significant results have been achieved, featuring fast convergence and high search accuracy. The second group uses a traditional PSO and a modified PSO to calculate a known singular pose (failure of numerical solution), and verifies the convergence speed and stability of the improved PSO when solving inverse kinematics. Advantages, at the same time, comparing the two algorithm errors further illustrates the accuracy advantages of the improved PSO algorithm. The position coordinates of the end effector of the robotic arm have nothing to do with the joint 6. Therefore, the particles in the algorithm population are 5-dimensional individuals. The specific parameters of the test are set as follows.

1) Traditional PSO: the maximum number of iterations is set to 200, the total number of particles involved in the calculation is 500, C_1 takes 1.4962, C_2 takes 1.4962, and w takes 0.7298.
2) Improved PSO: the maximum number of iterations is set to 200, and the total number of particles involved in the calculation is 500 (divided into 10 small particle swarms, each small particle swarm is assigned 50 particles). The inertia parameters w_i of 10 small particle swarms are set as: 0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4, 1.6, 1.8, 2.0, C_1 takes 1.4962, C_2 takes 1.4962, and changes dynamically with the increase of the number of iterations, And combined with immigration algorithm.

A. FUNCTION TEST

In order to evaluate the performance of the algorithm, we adopted 10 test functions. See Table 2.

These four functions include single-mode functions and multi-mode functions, correlated and uncorrelated functions. Among them, 1-6 is a single-mode function, 7-10 is a multi-mode function, and the number of local extremums increases exponentially as the dimension increases. Table 3 and the results of the traditional PSO algorithm and the improved PSO algorithm repeated 30 times for the test function 1-10, the simulation results include the average optimal value of the test function in the case of successful convergence, and the number of successful convergence of the algorithm within 30 times.

It can be seen from Table 3 that the improved PSO algorithm in this paper can converge on the single-mode function 1-6 repeated 30 simulations, and they have achieved an optimal value that is significantly better than the former two. For the global single peak function f_1, although all three algorithms converge successfully, the accuracy of the optimal

TABLE 2. **Test function expressions, dimensions, search spaces and global optimal solutions.**

NO.	expression	dimension	Search space	Global optimal
1	$f_1(\vec{x}) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	$f_1(\vec{0}) = 0$
2	$f_2(\vec{x}) = \sum_{i=1}^{n} \lvert x_i\rvert + \prod_{i=1}^{n} \lvert x_i\rvert$	30	$[-10,10]^n$	$f_2(\vec{0}) = 0$
3	$f_3(\vec{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100,100]^n$	$f_3(\vec{0}) = 0$
4	$f_4(\vec{x}) = \max\{\lvert x_i\rvert, 1 \le i \le n\}$	30	$[-100,100]^n$	$f_4(\vec{0}) = 0$
5	$f_5(\vec{x}) = \sum_{i=1}^{n} i x_i^2$	30	$[-1.28,1.28]^n$	$f_5(\vec{0}) = 0$
6	$f_6(\vec{x}) = \sum_{i=1}^{n} (\lvert x_i + 0.5\rvert)^2$	30	$[-100,100]^n$	$f_6(\vec{p}) = 0$ $-\frac{1}{2} \le p \le \frac{1}{2}$
7	$f_7(\vec{x}) = \sum_{i=1}^{n} (100(x_{i+1} - x_i{}^2)^2 + (x_i - 1)^2$	30	$[-30,30]^n$	$f_7(1) = 0$
8	$f_8(\vec{x}) = 418.9829 \times n - \sum_{i=1}^{n} x_i \sin(\sqrt{\lvert x_i\rvert})$	30	$[-500,500]^n$	$f_8(\overrightarrow{420.96}) = 0$
9	$f_9(\vec{x}) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	30	$[-5.12,5.12]^n$	$f_9(0) = 0$
10	$f_{10}(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$	30	$[-600,600]^n$	$f_{10}(\vec{0}) = 0$

TABLE 3. **Simulation results of test functions 1-10.**

NO.	Tradition PSO		Multi-PSO		This article improves PSO	
	average value	Successful convergence times	average value	Successful convergence times	average value	Successful convergence times
f_1	5.08e-6	30	2.23e-7	30	1.56e-12	30
f_2	4.51e-5	24	8.31e-12	25	1.68e-19	30
f_3	3.71e-5	15	2.25e-8	24	3.29e-16	30
f_4	1.69e-9	26	6.25e-15	22	2.56e-22	30
f_5	2.53e-5	8	4.77e-8	17	4.81e-17	30
f_6	7.52e-7	23	5.47e-11	27	5.27e-13	30
f_7	7.58e-3	11	5.66e-5	14	6.92e-12	26
f_8	2.85e-2	14	5.24e-11	24	5.52e-18	30
f_9	6.27e-2	25	2.73e-2	19	8.38e-12	30
f_{10}	1.29e-2	10	6.91e-9	19	5.27e-15	27

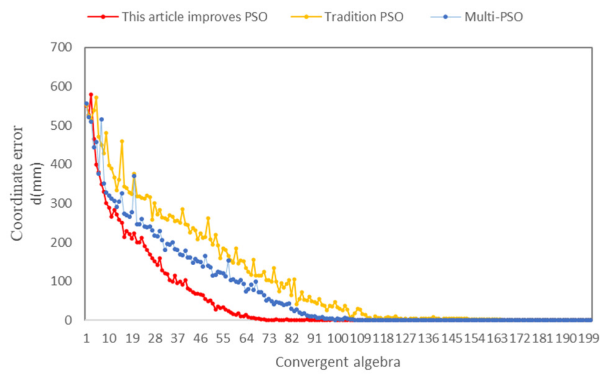


FIGURE 6. **Comparison of the position error values of the three algorithms.**

value found by the improved PSO algorithm in this paper is 6 orders of magnitude higher than that of the traditional PSO algorithm, and it is a multi-subgroup Five orders of magnitude of the algorithm.

There are a lot of local optimal values in the solution space of the multimode function, and it is difficult to find its global best. In the 30 simulations of the two multimode functions by the improved PSO algorithm, the number of convergences is greater than that of the traditional PSO algorithm and multi-subgroups, and The multi-mode function f_{7-10} can all successfully converge, which shows that the algorithm has greatly improved from the traditional PSO algorithm in getting rid of the local extremum of the function.

B. GENERAL ROBOT INVERSE KINEMTICS SOLUTION EXPERIMENT

Through calculation and actual operation, an experimental point of Comau NJ-220 general robot is obtained:

$$R = [-0.4967, 0.0445, 0.8668]$$
$$P = [1639.26, -146.94, 900.79]$$

The experimental environment and parameter settings are described above.

Substituting the above test points into a Jacobian matrix, the matrix is singular, so the test point is a singular position point, and the specific parameters are consistent with the function test. For details, see above, using three algorithms to repeatedly calculate the point 5 times, select the fastest convergence, The three curves corresponding to the three algorithms are plotted in the figure 6.

It can be seen from Fig. 6that for the singular position, the traditional PSO algorithm and the improved PSO algorithm can be solved normally. However, the improved PSO can quickly converge to the target coordinate error of 1 when computing less algebra, while the traditional PSO requires more algebra, and its target solution mutation will appear during the convergence process Case. In order to further compare the difference between the convergence rates of the two algorithms, the convergence algebra of the above five calculations is listed in Table 4. The coordinate error in Table 4 is the joint angle obtained by the algorithm. The corresponding coordinates p_x, p_y, p_z and the target coordinates (p'_x, p'_y, p'_z). The spatial distance error

TABLE 4. Comparison of three algorithm simulation results.

NO.	algorithm	Convergent algebra	Coordinate error d(mm)	Angular error e(deg)
1	Tradition PSO	128	3×10^{-1}	3.31
	Multi-PSO	117	9×10^{-2}	1.96
	improves PSO	92	1×10^{-3}	0.13
2	Tradition PSO	141	5×10^{-2}	3.25
	Multi-PSO	129	5×10^{-3}	2.53
	improves PSO	96	8×10^{-4}	0.28
3	Tradition PSO	136	4×10^{-2}	2.31
	Multi-PSO	125	9×10^{-4}	1.77
	improves PSO	86	4×10^{-4}	0.11
4	Tradition PSO	145	5×10^{-2}	2.47
	Multi-PSO	139	1×10^{-4}	1.91
	improves PSO	78	9×10^{-4}	0.10
5	Tradition PSO	156	7×10^{-2}	1.99
	Multi-PSO	131	3×10^{-3}	1.25
	improves PSO	92	1×10^{-3}	0.07

d, $d = \sqrt{(p_x - \bar{p_x})^2 + (p_y - \bar{p_y})^2 + (p_z - \bar{p_z})^2}$, the joint angle error e is the optimal solution $(\theta_1, \theta_2, \theta_3, \theta_4, \theta_5)$ obtained by the algorithm and the exact solution $(\theta_1', \theta_2', \theta_3', \theta_4', \theta_5')$ The sum of the absolute values of the components between the two, $e = \sum_{i=1}^{5} \left|\theta_i - \theta_i'\right|$, used to measure the excellent degree of the search solution. From the data in Figure 6 and table 4, it can be seen that the traditional PSO, multi subgroup and the improved PSO in this paper can search the convergence solution for the given target point.

For the Comau NJ-220 robot, although the optimal solutions obtained by the three algorithms are close to the exact solutions obtained by the analytical method, comparing the target coordinate errors D of the three algorithms, it can be found that the traditional PSO algorithm and the multi subgroup algorithm are 2-3 orders of magnitude higher than the improved PSO algorithm, among which the traditional PSO algorithm has the largest coordinate error. From this we can see that the introduction of multiple subgroups can avoid the algorithm falling into local optimum, thus reducing the error of the results. Observing the joint angle error e of the three algorithms, the traditional PSO algorithm and multi subgroup algorithm are about 102 times of the improved algorithm, and the angle error of the traditional PSO algorithm is slightly higher than that of the multi subgroup algorithm. The results show that the accuracy of the improved algorithm is greatly improved due to the introduction of migration operator.

At the same time, in terms of convergence speed, the traditional PSO algorithm needs 130 generations on average from the beginning of iteration to the convergence state, 120 generations for many subgroups, and 90 generations for the improved PSO algorithm. It can be seen that the improved PSO algorithm reduces the error of local optimization and improves the convergence speed.

TABLE 5. Comparison of three algorithm simulation results.

	Pass times	Excellent times
improves PSO	100	74

C. GENERAL ROBOT INVERSE KINEMTICS SOLUTION EXPERIMENT

In order to more macroscopic prove the stability and accuracy of the improved algorithm, we use the same singular position,

$$R = [-0.4967, 0.0445, 0.8668]$$
$$P = [1639.26, -146.94, 900.79]$$

use the improved PSO algorithm to solve its inverse solution 100 times, the test environment remains unchanged, the parameter settings remain unchanged (see the above parameter settings), the pass rate and excellent rate of all experiments are counted, we set the pass level as: coordinate within 200 generations The error value converges to 10-2, set the excellent level to 10-4 within 100 generations, and the times of convergence to "pass and excellent" level in 100 experiments are shown in Table 5, and the statistical results are used to prove the significance of the experiment.

In Table 5, we can see that in the 100 times of solving experiments, 100 times have reached the pass level, which shows that the improved algorithm has a certain stability, which can meet the daily use of general industrial robots; at the same time, 74 experiments have reached the excellent level, which shows that the improved algorithm has high accuracy and can meet the needs of some precise process production.

V. CONCLUSION

In this paper, an improved PSO algorithm for general robot inverse kinematics is proposed. The objective function and fitness function of the improved PSO are established by using the forward kinematics equation of the robot. In order to solve the problem that traditional PSO algorithm has fallen into the local optimum, the algorithm structure of multiple particle swarm is improved, and the migration algorithm is proposed to enhance the diversity of particle swarm.

The improved PSO algorithm is applied to solve the inverse problem of Comau nj-220 robot. The results show that, compared with the traditional PSO algorithm, the improved PSO algorithm solves the local convergence problem, improves the stability, not only improves the convergence accuracy, but also effectively improves the operation speed. It is proved that the method proposed in this paper is suitable for solving inverse kinematics problems of general robots, and shows certain advantages; in addition, in the promotion and practicability of the algorithm, we will make further exploration in the future research and practical operation. To solve the problem of general robot inverse kinematics, we will explore more research methods to apply in the future, and provide more solutions for this field.

REFERENCES

[1] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*. Berlin, Germany: Springer, 2008.

[2] L. Yang, Z. Huan, and D. Han, "Inverse kinematics of general robots based on multi population genetic algorithm," *J. Mech. Eng.*, vol. 53, no. 3, pp. 1–8, Sep. 2017.

[3] T. Sun, B. Lian, J. Zhang, and Y. Song, "Kinematic calibration of a 2-DoF over-constrained parallel mechanism using real inverse kinematics," *IEEE Access*, vol. 6, pp. 67752–67761, 2018, doi: 10.1109/ACCESS.2018.2878976.

[4] I. Zaplana and L. Basanez, "A novel closed-form solution for the inverse kinematics of redundant manipulators through workspace analysis," *Mechanism Mach. Theory*, vol. 121, pp. 829–843, Mar. 2018.

[5] Y. Feng, S. Deb, G.-G. Wang, and A. H. Alavi, "Monarch butterfly optimization: A comprehensive review," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114418, doi: 10.1016/j.eswa.2020.114418.

[6] J. Li, H. Lei, A. H. Alavi, and G.-G. Wang, "Elephant herding optimization: Variants, hybrids, and applications," *Mathematics*, vol. 8, no. 9, p. 1415, Aug. 2020, doi: 10.3390/math8091415.

[7] G.-G. Wang, A. H. Gandomi, A. H. Alavi, and D. Gong, "A comprehensive review of krill herd algorithm: Variants, hybrids and applications," *Artif. Intell. Rev.*, vol. 51, no. 1, pp. 119–148, Jan. 2019, doi: 10.1007/s10462-017-9559-1.

[8] S. N. Makhadmeh, A. T. Khader, M. A. Al-Betar, S. Naim, Z. A. A. Alyasseri, and A. K. Abasi, "Particle swarm optimization algorithm for power scheduling problem using smart battery," in *Proc. IEEE Jordan Int. Joint Conf. Electr. Eng. Inf. Technol. (JEEIT)*, Apr. 2019, pp. 672–677.

[9] R. V. Ram, P. M. Pathak, and S. J. Junco, "Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling," *Mechanism Mach. Theory*, vol. 131, pp. 385–405, Jan. 2019.

[10] X. Hong, X. Qijun, and C. Haibin, "Differential adaptive chaos particle swarm optimization of robot inverse kinematics," *Comput. Eng. Appl.*, vol. 53, no. 8, pp. 126–131, Dec. 2017.

[11] Z. Jingwei, F. Husheng, and S. invent, "Adaptive particle swarm optimization for inverse kinematics of redundant manipulator," *Comput. Eng. Appl.*, vol. 55, no. 14, pp. 215–220, Dec. 2019.

[12] P. K. Das and P. K. Jena, "Multi-robot path planning using improved particle swarm optimization algorithm through novel evolutionary operators," *Appl. Soft Comput. J.*, vol. 92, no. 1, Jun. 2020, Art. no. 106312.

[13] M. M. Golchi, S. Saraeian, and M. Heydari, "A hybrid of firefly and improved particle swarm optimization algorithms for load balancing in cloud environments: Performance evaluation," *Comput. Netw.*, vol. 10, no. 162, pp. 1286–1389. Sep. 2019.

[14] A. A. Nagra, F. Han, and Q. H. Ling, "An improved hybrid self-inertia weight adaptive particle swarm optimization algorithm with local search," *Eng. Optim.*, vol. 51, no. 7, pp. 305–315, Jul. 2019.

[15] M. Kelemen, I. Virgala, T. Lipták, L. Miková, F. Filakovský, and V. Bulej, "A novel approach for a inverse kinematics solution of a redundant manipulator," *Appl. Sci.*, vol. 8, no. 11, p. 2229, Nov. 2018, doi: 10.3390/app8112229.

[16] Y. Shen and G. Ge, "Multi-objective particle swarm optimization based on fuzzy optimality," *IEEE Access*, vol. 7, pp. 101513–101526, 2019, doi: 10.1109/ACCESS.2019.2926584.

[17] J. P. Tripathi and S. Ghoshal, "Combining inertia and constriction technique in the PSO applied to fault identification in a hydraulic system," *Proc. Inst. Mech. Eng., C, J. Mech. Eng. Sci.*, vol. 231, no. 14, pp. 2730–2740, Jul. 2017.

[18] G. Shikai and S. Xin, "Path planning of mobile robot based on improved particle swarm optimization algorithm," *Electron. Meas. Technol.*, vol. 42, no. 3, pp. 60–64, Aug. 2019.

[19] H. Alkhraisat and H. Rashaideh, "Dynamic inertia weight particle swarm optimization for solving nonogram puzzles," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 10, pp. 277–280, 2016, doi: 10.14569/IJACSA.2016.071037.

[20] X. Hong, Y. Wenke, and Y. Peng, "Inverse motion solution method of multi sub group hierarchical differential particle swarm optimization," *J. Electron. Meas. Instrum.*, vol. 29, no. 10, pp. 1456–1463, Oct. 2015.

[21] J. Chrouta, A. Zaafouri, and M. Jemli, "An improved heterogeneous multi-swarm PSO algorithm to generate an optimal T-S fuzzy model of a hydraulic process," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 2039–2053, Apr. 2018.

[22] Y. Peng, *Application of Multi Sub Group Hierarchical Particle Swarm Optimization Difference Algorithm in Inverse Kinematics of Robot*, vol. 3. Changsha, China: Hunan Univ., Apr. 2015. [Online]. Available: http://kns.cnki.net/kns/detail/detail.aspx?FileName=1015734504.nh&DbName=CMFD2017

[23] A. Ebrahimi-Zade, H. Hosseini-Nasab, Y. Zare-mehrjerdi, and A. Zahmatkesh, "Multi-period hub set covering problems with flexible radius: A modified genetic solution," *Appl. Math. Model.*, vol. 40, no. 4, pp. 2968–2982, Feb. 2016.

[24] Y. Zhang, D.-W. Gong, and J. Cheng, "Multi-objective particle swarm optimization approach for cost-based feature selection in classification," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 1, pp. 64–75, Jan. 2017, doi: 10.1109/TCBB.2015.2476796.

[25] S. Wheatley, V. Filimonov, and D. Sornette, "The Hawkes process with renewal immigration & its estimation with an EM algorithm," *Comput. Statist. Data Anal.*, vol. 94, no. 7, pp. 120–135, Feb. 2016.

[26] A. El Afia, O. Aoun, and S. Garcia, "Adaptive cooperation of multi-swarm particle swarm optimizer-based hidden Markov model," *Prog. Artif. Intell.*, vol. 8, no. 4, pp. 441–452, Dec. 2019.

[27] A. K. Dubey, A. Kumar, and R. Aggarwal, "An efficient ACO-PSO-based framework for data classification and preprocessing in big data," *Evol. Intell.*, pp. 1–14, Sep. 2020, doi: 10.1007/s12065-020-00477-7.

[28] S. Fuenzalida, K. Toapanta, J. Paillacho, and D. Paillacho, "Forward and inverse kinematics of a humanoid robot head for social human robot-interaction," in *Proc. IEEE 4th Ecuador Tech. Chapters Meeting (ETCM)*, Nov. 2019, pp. 1–4, doi: 10.1109/ETCM48019.2019.9014887.

[29] A. A. Nagra, F. Han, Q.-H. Ling, and S. Mehta, "An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization," *IEEE Access*, vol. 7, pp. 50388–50399, 2019, doi: 10.1109/ACCESS.2019.2903137.

[30] Q. Feng, Q. Li, P. Chen, H. Wang, Z. Xue, L. Yin, and C. Ge, "Multi-objective particle swarm optimization algorithm based on adaptive angle division," *IEEE Access*, vol. 7, pp. 87916–87930, 2019, doi: 10.1109/ACCESS.2019.2925540.

LIU YIYANG was born in Shenyang, Liaoning Province, China in 1980. He received the B.S. degree and M.A. degree in automation, control theory and control engineering from Northeast University in 2003 and 2006, and the Ph.D. in pattern recognition and intelligent system from Shenyang Institute of automation, Chinese Academy of Sciences in 2010.

He is now a professor and principle investigator of Shenyang Institute of automation, Chinese Academy of Sciences. His main research interests include: cyber physical system, complex system modeling and simulation, intelligent optimization and intelligent control. As the project leader, he has presided over a number of National Natural Science Foundation projects, Ministry of industry and information technology's intelligent manufacturing special projects and industrial Internet special projects.

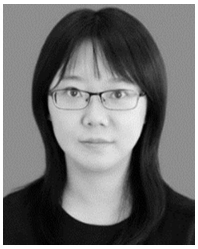
At present, he is a member of the Edge Computing Committee in the Chinese Association of Automation, an expert in the field of advanced manufacturing technology of "National Technology Prediction" of the Ministry of Science and Technology, and an peer-reviewed expert of the National Natural Science Foundation of China.

XI JIALI was born in Huludao, Liaoning, China, in 1996. She received the B.S. degree in electrical engineering from Shenyang Jianzhu University, Shenyang, in 2018. She is currently pursuing the Master's degree in control science and engineering jointly trained by Shenyang Jianzhu University and China Shenyang Institute of Automation.

Since enrolling in school, she has been engaged in professional research in artificial intelligence and started research on the background of industrial robots, mainly studying the kinematic analysis and trajectory planning methods of industrial robots, and obstacle avoidance in dynamic environments.

BAI HONGFEI was born in Yingkou, Liaoning, China, in 1986. He received the B.S. degree in mechanical design, manufacturing and automation from the Dalian University of Technology, in 2008.

He is currently an Associate Professor with the Shenyang Institute of Automation, Chinese Academy of Sciences. His main research interests include automation, intelligent optimization, and intelligent control. As a Technical Leader, he has presided over a number of Ministry of industry and information technology's intelligent manufacturing special projects and industrial Internet special projects.



WANG ZHINING was born in Shenyang, Liaoning, China, in 1990. She received the B.S. degree in electronic and information engineering from China Agricultural University, Beijing, in 2013, and the M.S. degree in computer science and technology, in 2015.

Since 2016, she has been with the Shenyang Institute of Automation, Chinese Academic of Sciences. Her research interests include intelligent manufacturing, OPC UA, and AGV path planning.

SUN LIANGLIANG was born in Shenyang, Liaoning, China, in 1981. He received the B.S. degree in electrical engineering from Beijing Jiaotong University, Beijing, in 1999, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, Liaoning, in 2015.

Since 2015, he has been an Assistant Professor with the Department of Information and Control Engineering. His research interests include Intelligent manufacturing systems – planning, scheduling, and coordination of design, manufacturing, and service activities; smart and green buildings and eco communities – optimized energy management, HVAC fault detection and diagnosis, and emergency crowd guidance.

Dr. Liangliang is the member of IEEE Robotics and Automation Society, the member of Technical Committer 8.3 ''Modeling and Control of Environmental System'' IFAC, and the National Natural Science Foundation Committee Peer-reviewed expert.