IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 3, 2019, accepted January 28, 2019, date of publication February 11, 2019, date of current version February 27, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2898218

Direct Trajectory Planning Method Based on IEPSO and Fuzzy Rewards and Punishment Theory for Multi-Degree-of Freedom Manipulators

XUEYING LV[1], ZHAOXIA YU[2], MINGYANG LIU[1,3], GUANYU ZHANG[1,3], AND LIU ZHANG[1,3]

[1]College of Instrumentation and Electrical Engineering, Jilin University, Changchun 130061, China
[2]Shanghai Institute of Satellite Engineering, Shanghai 20000, China
[3]National Engineering Research Center of Geophysics Exploration Instruments, Jilin University, Changchun 130061, China

Corresponding authors: Guanyu Zhang (zhangguanyu@jlu.edu.cn) and Liu Zhang (zhangliu@jlu.edu.cn)

This work was supported in part by the Key Project of Shanghai Science and Technology Committee under Grant 16DZ1120400, in part by the National Natural Science Foundation of China under Grant 51705187, in part by the Postdoctoral Science Foundation of China under Grant 2017M621202, in part by the National Key Research and Development Program of China under Grant 2016YFB0501003, in part by the Jilin Provincial Science and Technology Department under Grant 20170204057GX, and in part by the Shanghai Rising-Star Program under Grant 16QB1401000

ABSTRACT Manipulator is a kind of commonly used multi-degree-of-freedom nonlinear system. There is a strong coupling between the links, and the movement of each link constraints and affects each other, which increases the difficulty of motion analysis of the system and reduces its trajectory planning efficiency under specific task targets. To solve this problem, a direct trajectory planning method based on an improved particle swarm optimization (PSO) algorithm, called IEPSO, and the fuzzy rewards and punishment theory is proposed in this paper. First, on the basis of preserving the local search ability of PSO, the global search ability of the population is improved by increasing a population exchange item. At the same time, in order to avoid the population falling into the local optimal value, the last elimination principle is incorporated into the standard PSO algorithm. Second, the fuzzy rewards and punishment theory is introduced to reduce the redundant decoupling operation, which can not only ensure the accuracy of manipulator trajectory planning but also effectively reduce the calculation amount of the trajectory planning for the multi-degree-of-freedom manipulator, to improve the optimization efficiency. Finally, the direct trajectory planning method of the multi-degree-of-freedom manipulator is compared and tested. It can be seen that the efficiency scalar and accuracy of the proposed direct trajectory planning method are significantly higher than those of other optimization methods.

INDEX TERMS Improved particle swarm optimization algorithm, fuzzy rewards and punishment theory, trajectory planning, multi-degree-of-freedom manipulators.

I. INTRODUCTION

With the development of science and technology, the manipulators have been applied to many fields, such as complex pipe installation, medical manipulation, and space manipulator-assisted docking. Multi-degree-of-freedom(DOF) manipulators were developed to perform high-complexity tasks, which enhance coupling between linkages. In the process of solving the inverse kinematics of manipulator, it is impossible to find the independent and irrelevant variable equations for the pure algebraic method. Although all solutions can be found by analytical method, the real-time control requirements of the manipulator are violated. The iterative method is feasible in most cases, but it cannot find all solutions, and there is a risk that the iteration does not converge. The data volume of the planning using the differential compensation method is too large and the efficiency is low, thus increases the difficulty in trajectory planning and control of multi parameters, strong coupling, and nonlinear motion systems.

The associate editor coordinating the review of this manuscript and approving it for publication was M A Hannan.

For the problem of the trajectory planning of multi-DOF manipulators, indirect methods often use polynomial interpolation or the B-spline interpolation algorithm to interpolate existing motion trajectories. Subsequently, an intelligent trajectory algorithm is used to optimize the existing trajectory and consequently meet the desired optimization performance.

Gasparetto and Zanotto [1] used the fifth-order B-spline curve method to interpolate the path of the motion trajectory, and considered speed, acceleration, and addition speed as the constraints of the trajectory planning of a manipulator. Parys and Pipeleers [2] described the trajectory planning of a manipulator in Cartesian space to complete the specified task, used a polynomial spline to parameterize the trajectory of the manipulator end, and applied the B-spline basis function to constrain the trajectory effectively. Liu *et al.* [3] proposed a time-optimal and dynamic continuous trajectory planning method using cubic spline interpolation in Cartesian space and seventh-order B-spline interpolation in joint space, thereby potentially achieving smooth tracking performance in actual motion. Intelligent optimization algorithms have been widely used in solving multiparameter problems. Xidias [4] used a super-concave–convex surface to represent the three-dimensional workspace of a manipulator and optimized the running time of the manipulator by using a multipopulation Genetic Algorithm(GA). Wang *et al.* [5] used a constrained differential method to determine the optimal values of the design variables and consequently realized the trajectory planning of the manipulator. Wang *et al.* [6] used PSO and an interior-point method to optimize the motion planning of the rotating joint of a 3-DOF manipulator. Menasri *et al.* [7] converted the trajectory planning problem into an optimization problem to be solved by a dual G by discretizing the trajectory of the end actuator of a redundant manipulator from the initial position to the target position in Cartesian space. Marcos *et al.* [8] used a closed-loop pseudo-inverse method and GA to realize the trajectory planning of a redundant manipulator. Cao *et al.* [9] combined kinematics with PSO to overcome the dynamic singularity problem, constrained PSO with adaptive inertia weight is used to optimize the joint trajectory planning of a free-floating, dual-arm space robot to satisfy specific targets and constraints. PSO is an advantageous intelligent optimization algorithm because of its fast computational speed and few required parameters for adjustment. Thus, it is widely used in solving problems of nonlinear systems with strong coupling and multiple parameters. However, the traditional PSO algorithm is prone to premature convergence. To solve this problem, many scholars have improved the traditional PSO algorithm by adjusting relevant parameters and combining it with other intelligent algorithms. The optimization capability of the traditional PSO is improved to a certain extent by the adjustment of certain parameters, such as inertia weight [10], [11] and learning factor [12]. However, at the beginning of the iteration, the optimal value of the algorithm is similar to the local optimum, and the PSO algorithm still falls into the local extreme value. Aydilek [13] proposed a hybrid algorithm that combines the firefly and PSO algorithms, in which the local search process is determined by the optimal fitness value retained from the previous iteration. Chen *et al.* [14] proposed a cross-operation PSO algorithm, it is a bi-level PSO algorithm with a positive feedback mechanism and performs cross-variation at the best position of each particle to maintain population diversity. Chen *et al.* [15] proposed a chaotic dynamic-weight PSO algorithm. Chen *et al.* [16] proposed a dynamic differential PSO algorithm that performs differential evolution in each subgroup and combines the method of differential mutation to improve global search capability. Jiang *et al.* [17] proposed a hybrid optimization algorithm that combines wavelet mutation with PSO. Liu *et al.* [18] proposed a cooperative multigroup PSO algorithm to handle multi-objective optimization problems with uncertain, rapid environmental changes.

This analysis shows that two problems still remain to be solved.

(1) Using indirect methods for trajectory planning requires the kinematic inverse problem to be solved in real time, thereby increasing the calculation amount of the algorithm.

(2) Only the optimal time or the lowest energy consumption is considered to be the optimization objective, whereas the operational accuracy of the manipulator is disregarded.

To solve these problems, this research proposes a direct trajectory planning method that is based on an improved PSO algorithm and Fuzzy rewards and punishment theory. This method does not require the real-time solution of inverse kinematic problems. The last elimination principle is introduced into the improved PSO algorithm to maintain population diversity, and the increased local–global information-sharing item φ_3 is used to improve the global information-sharing capability of the algorithm. Under the existing constraint conditions, a multiparameter rewards and punishment rule is introduced to improve the efficiency of the IEPSO algorithm. By this method, the large calculation amount of indirect methods can be avoided while ensuring accuracy.

II. PROBLEM DESCRIPTION

Assuming that the base is not disturbed by the movement of the manipulator, the global coordinate system is established on the base. The coordinate systems of the joints of the manipulator are shown in Figure 1, including the global coordinate system $\boldsymbol{R}[\mathrm{X}, \mathrm{Y}, \mathrm{Z}]$, end actuator coordinate system $\boldsymbol{H}[\mathrm{X_E}, \mathrm{Y_E}, \mathrm{Z_E}]$, and local coordinate system of each joint $\boldsymbol{i}[\mathrm{X_i}, \mathrm{Y_i}, \mathrm{Z_i}]$. The Denavit-Hartenberg(D-H) method is used to describe the relationship between the translational and rotational motions of adjacent linkages [19]. Through the coordinate transformation matrix $T_{\mathrm{i-1}}^{\mathrm{i}}$ between the two linkages, the transformation matrix ${}^{\mathrm{R}}T_{\mathrm{H}}$ of the end actuator relative to the base coordinate system can be derived. The $T_{\mathrm{i-1}}^{\mathrm{i}}$ and ${}^{\mathrm{R}}T_{\mathrm{H}}$ transformation matrices are as shown in (1) and (2), as shown at the bottom of the next page.

where $T_{\mathrm{i-1}}^{\mathrm{i}}$ is the transformation matrix of the coordinate system i with respect to $i + 1$; d_{n} is the distance between

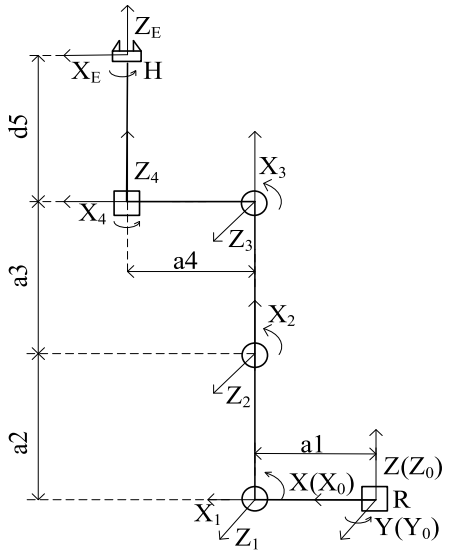

FIGURE 1. The coordinate systems of mechanical arm.

two adjacent joints in the X_i and X_{i+1} axes; a_n is the distance between two adjacent joints in the Z_i to Z_{i+1} axes; θ_i is the angle between the X_i and X_{i+1} axes, with the right side of the Z axes being the positive direction; and α_i is the angle between the Z_i and Z_{i+1} axes, with the right side of the X axes being the positive direction.

$$ {}^{R}T_H = \begin{bmatrix} n & o & a & p \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

where

$$ \begin{aligned} n_x &= c\theta_1 c\theta_5 c\theta_{2+3+4} - s\theta_1 s\theta_5; \\ n_y &= c\theta_1 c\theta_5 + c\theta_5 s\theta_1 c\theta_{2+3+4q}; \\ n_z &= -c\theta_5 s\theta_{2+3+4}; \\ o_x &= -s\theta_1 c\theta_5 - c\theta_1 s\theta_5 c\theta_{2+3+4}; \\ o_y &= c\theta_1 c\theta_5 - s\theta_1 s\theta_5 c\theta_{2+3+4}; \\ o_z &= s\theta_5 s\theta_{2+3+4}; \\ a_x &= c\theta_1 s\theta_{2+3+4}; \\ a_y &= s\theta_1 s\theta_{2+3+4}; \\ p_x &= c\theta_1(a_3 c\theta_{2+3} + a_2 c\theta_2 + a_4 c\theta_{2+3+4} + d_5 s\theta_{2+3+4} + a_1); \\ p_y &= s\theta_1(a_3 c\theta_{2+3} + a_2 c\theta_2 + a_4 c\theta_{2+3+4} + d_5 s\theta_{2+3+4} + a_1); \\ p_z &= d_5 c\theta_{2+3+4} - a_2 s\theta_2 - a_3 s\theta_{2+3} - a_4 s\theta_{2+3+4} \end{aligned} $$

where $\boldsymbol{p} = [\mathrm{p_x}, \mathrm{p_y}, \mathrm{p_z}]^{\mathrm{T}}$ is the position vector of the end actuator mechanism relative to the base coordinate system; $\boldsymbol{n} = [\mathrm{n_x}, \mathrm{n_y}, \mathrm{n_z}]^{\mathrm{T}}$, $\boldsymbol{o} = [\mathrm{o_x}, \mathrm{o_y}, \mathrm{o_z}]^{\mathrm{T}}$, $\boldsymbol{a} = [\mathrm{a_x}, \mathrm{a_y}, \mathrm{a_z}]^{\mathrm{T}}$ are the pose vectors of the end actuator mechanism relative to the base coordinate system; $s\theta_{\mathrm{n}}$ and $c\theta_{\mathrm{n}}$ represent $sin\theta_{\mathrm{n}}$ and $cos\theta_{\mathrm{n}}$, respectively; $c\theta_{2+3}$ and $c\theta_{2+3+4}$ represent $cos(\theta_2+\theta_3)$ and $cos(\theta_2+\theta_3+\theta_4)$, respectively; and $s\theta_{2+3}$ and $s\theta_{2+3+4}$ represent $\sin(\theta_2+\theta_3)$ and $\sin(\theta_2+\theta_3+\theta_4)$, respectively. Equation 2 describes the strong nonlinear coupling relationship between the joint angles of the multi-degree-of-freedom manipulator. The time-varying angle of each joint of the manipulator can be solved by the solution of the nonlinear equations of n coupling. Meanwhile, the structural and angular constraints of the manipulator are considered in the process of manipulator movement, thereby increasing the difficulty of multi-DOF manipulator trajectory planning and control. Moreover, the choice of joint running angle has multiplicity; therefore, selecting the mechanical manipulator joint running angle from the initial point to the end point is essential to improving the working efficiency and accuracy of the manipulator under certain constraints.

III. IMPROVED PARTICLE SWARM OPTIMIZATION ALGORITHM

The improved PSO algorithm proposed in this work preserves the local development capability of the traditional PSO algorithm and increases the local–global information-sharing term φ_3 to enhance the global exploration capability of the algorithm. Moreover, on the basis of genetic variation, the principle of last elimination is adopted to maintain the diversity of the population. These improvements aim to enhance the global optimization performance of the PSO algorithm. Figure 2 shows the specific implementation process of the IEPSO algorithm.

First, the position and velocity of the particles in the population are randomly initialized, and the fitness value of each individual particle is calculated. If the position and fitness of the individual particles and the global optimum particles are maintained, then particle swarm operation is re-executed. The increased local–global information-sharing item φ_3 represents the information exchange between the local optimal particle and the global optimal particle obtained by the current iteration and is used to balance the exploration and development capability of the particles. Equations (4) and (5) are used to update speed and location in the IEPSO algorithm.

$$ \varphi_3 = C_3 R_3 \left| p_{gd}^t - p_{id}^t \right| \quad (3) $$

$$ \begin{aligned} v_{id}^{t+1} &= \omega v_{id}^t + C_1 R_1 \left(p_{id}^t - x_{id}^t\right) + C_2 R_2 \left(p_{gd}^t - x_{id}^t\right) \\ &\quad + C_3 R_3 \left| p_{gd}^t - p_{id}^t \right| \end{aligned} \quad (4) $$

$$ x_{id}^t = x_{id}^t + v_{id}^{t+1} \quad (5) $$

$$ T_{i-1}^{i} = \begin{pmatrix} \cos(\theta_i) & -\sin(\theta_i)\cos(\theta_i) & \sin(\theta_i)^2 & a_n\cos(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) & -\cos(\theta_i)\sin(\theta_i) & a_n\sin(\theta_i) \\ 0 & \sin(\theta_i) & \cos(\theta_i) & d_n \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1) $$

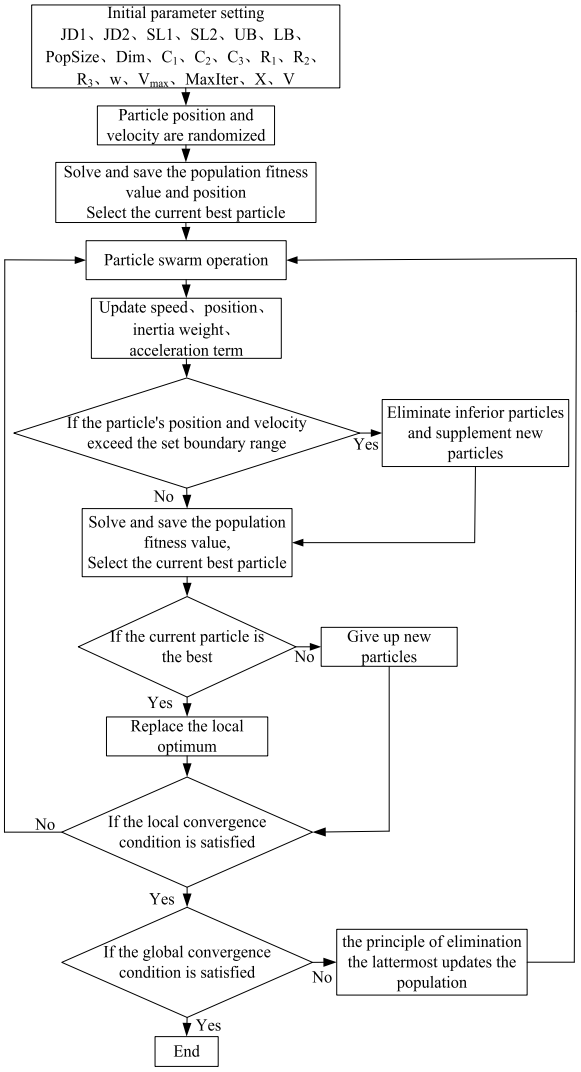

FIGURE 2. IEPSO algorithm flowchart.

where ω is the inertia weight, C_1 and C_2 are the learning factors, R_1 and R_2 are random numbers in the interval [0, 1], P_g^t is the global optimal position, P_i^t is the best historical position of the particle, x_i^d is the current position of the particle, and v_{id}^{t+1} is the next iteration speed. With a decrease in population diversity, the algorithm converges easily near the local optimal value. The fitness value is used as an evaluation criterion to eliminate particles with poor objective function, thereby maintaining population diversity. Within the predetermined range, particles to be added to the population are selected and PSO is re-executed. When the convergence conditions achieve convergence accuracy, the iteration is terminated and the global optimum is obtained.

IV. REWARDS AND PUNISHMENT THEORY

On the basis of the existing constraints in the multi-objective optimization process, a Fuzzy rewards and punishment rule for multiple parameters is introduced. In this way, the decoupling operation of the multiparameter system can be eliminated. Therefore, the application of these rules can reduce computational time while ensuring planning accuracy.

A. FUZZY REWARDS AND PUNISHMENT RULES

First, the input vector $\boldsymbol{X}$ is used as input and λ as a rewards and punishment factor. The input value is graded by rewards and punishment, and the real value becomes a rewards and punishment measure. Set a vector of X, $x_{\mathrm{i}} \in M_{\mathrm{i}} = [-e_{\mathrm{i}}, e_{\mathrm{i}}]$. The component X_{i} of set X is processed by the degree of rewards and punishment, which is $\lambda = [-1, 1]$. The rewards and punishment waken threshold value in the physics universe is set as ε, and the scope of the degree of reward and punishment is divided into two subdomains, namely, reward domain $R \in [0, 1]$ and punishment domain $P \in [-1, 0)$.

For a single-input, single-output system, the logical relationship of rewards and punishment is.

P and Q are the rewards and punishment sets of input domain A and output domain B, respectively. A rewards and punishment subset R of the direct product between P and Q is called the rewards and punishment relationship between P and Q.

$$\lambda_R(x, y) \Leftrightarrow \lambda_{P\times Q}(x, y) = \lambda_A(x) \wedge \lambda_B(y) \tag{6}$$

For a double-input, single-output system, the logical relationship of rewards and punishment is.

P and Q are the degree of rewards and punishment in the domains $\boldsymbol{X} \times \boldsymbol{Y}$ and $\boldsymbol{Y} \times \boldsymbol{Z}$, respectively. $\boldsymbol{R}$ is the degree of rewards and punishment in the domain, $\boldsymbol{X} \times \boldsymbol{Z}$. $\boldsymbol{R}$ is $\boldsymbol{P} \circ \boldsymbol{Q}$. The scope of the degree of rewards and punishment after synthesis is:

$$\lambda_R = P \times Q \to [-1, 1] \tag{7}$$

The degree of rewards and punishment can be defined as.

$$\lambda_R(x, z) \Leftrightarrow \lambda_{P\circ Q}(x, z) = \vee\left(\lambda_P(x, y) \wedge \lambda_Q(y, z)\right) \tag{8}$$

where, the operator $\wedge$ means to take the maximum value, $\vee$ means to take the minimum value, and $\circ$ is a composite operator.

In this study, we use the rule sentence "if A and B is C" to describe the rewards and punishment rule base. For two input variables, both variables are divided into two sets of rewards and punishments, and 24 rewards and punishment rule statements are established. The degrees of rewards and punishments for the input variables are set to [very far (VF), so far (SF), little far (LF), little close (LC), equal (BE), close(ZC), so close (SC),very close (VC)] and [high (H), middle (M), low (L)]. The degrees of rewards and punishment for the output variables are set to [very strong punishment (VSP), strong punishment (VVP), a little punishment (RSP), light punishment (RLP), no reward and punishment (NRP), light reward (RLR), strong reward (RSR), very strong reward (VSR)]. The established rewards and punishment rules table is shown in Table 1.

TABLE 1. Rewards and punishment reasoning table.

λ_R	VF	SF	LF	LC	BE	ZC	SC	VC
H	VSP	VSP	VSP	RSP	NPR	RLR	RLR	RSR
M	VSP	VVP	VVP	RLP	NPR	RLR	RLR	RSR
L	VSP	VVP	RSP	RLP	NPR	RLR	RSR	VSR

TABLE 2. Rewards and punishment rule functions.

Type	Function form	parameters
linear	$y = x - a/b - a \qquad a \le x \le b$	a and b are the maximum and minimum values of the $\boldsymbol{X}$ axis, respectively.
Gauss	$y = \exp\left(-(x-c)^2/\sigma^2\right)$	The parameter x determines the domain range of the variable; C determines the central point of the rewards and punishment rule function, and the width of the rewards and punishment rule function is determined by σ.
Bell	$y = 1/\left(1+\left\|(x-c)/a\right\|^{2b}\right)$	The parameter x determines the domain range of the variable; [a, b and c] determine the shape of the bell function.
Sigmoid	$y = 1/\left(1+\exp\left(-a*(x-c)\right)\right)$	The parameter x determines the domain range of the variable; [a,c] determine the shape of the Sigmoid function.
π Function	$y = 1/\left(1+\exp\left(-a_1*(x-c_1)\right)\right)+1/\left(1+\exp\left(-a_2*(x-c_2)\right)\right)$	The parameter x determines the domain range of the variable; [a1,c1,a2,c2] determine the shape of the Sigmoid function.
Hyperbolic tangent symmetric S	$y = \tanh\left((1/2)*x\right) = \left(1-\exp(-x)\right)/\left(1+\exp(-x)\right)$	The parameter x determines the domain range of the variable

B. REWARDS AND PUNISHMENT RULE FUNCTIONS

The rewards and punishment rules function generally assumes the following forms, as shown in Table 2.

The linear rewards and punishment rule function is applicable when the input parameters are densely distributed and the rewards and punishment are segmented. The sigmoid-type rewards and punishment rule function has a semi-open shape and is suitable for the ''maximum'' and ''minimum'' degrees of rewards and punishment. The π-type, bell-type, hyperbolic tangent symmetric S-type, and Gaussian-type rewards and punishment functions show a nonlinear increasing and decreasing trend when approaching zero.

V. EFFICIENCY OF OPTIMAL TRAJECTORY PLANNING

A. OPTIMIZATION OBJECTIVE

In this study, the trajectory planning problem with the best operating efficiency and minimum motion error is transformed into a multi-objective optimization problem. Equation (9) is an optimization objective function, consisting of two evaluation criteria.

$$\min F(X) = R(X) + S(X) \tag{9}$$

$$R(X) = \max\left(\frac{\theta_1}{\omega}, \frac{\theta_2}{\omega}, \frac{\theta_3}{\omega}, \frac{\theta_4}{\omega}\right) \tag{10}$$

$$err_i = \sqrt{(x-x_f)^2 + (y-y_f)^2 + (z-z_f)^2} \tag{11}$$

where F(X) is an efficiency scalar; R(X) is the longest time consumed for each joint of a manipulator; S(X) is the rewards and punishment function; θ_1, θ_2, θ_3, and θ_4 are the transformation angles; θ_5 is the rotation joint angle; and θ_6 is the execution joint angle of the end actuator mechanism. This study considers only the position of the end actuator mechanism reaching the target point, so the effects of θ_5 and θ_6 are not considered. In Equation (11), err_i is the position error between the target point and the end actuator mechanism of the manipulator. The position of the end actuator mechanism is P_c = [x, y, z], and the position of the target point is P_f = [x_f, y_f, z_f].

B. CONSTRAINT CONDITION

The purpose of manipulator trajectory planning is to generate the optimal joint trajectories without violating the constraints to accomplish the desired tasks. The optimization problems of the specific target F(X) in the case of the equality constraint H_i and the inequality constraint L_i are expressed as

Joint constraints,

$$H_i = \theta_2 + \theta_3 + \theta_4 = \varepsilon \tag{12}$$

$$\varepsilon = \arccos\left(\frac{\vec{z}\cdot\vec{p}}{|\vec{z}|\,|\vec{p}|}\right) \tag{13}$$

Linkage constraints,

$$L_{\max} < L_{cur} < L_{obj} \tag{14}$$

where joint 1 is a rotating joint without joint restraint, and only joints 2, 3, and 4 are considered. ε is the angle swept by

TABLE 3. The D-H parameters.

Number	θ	d	a	α
1	θ_1	0	10	-90°
2	θ_2	0	104	0
3	θ_3	0	96	0
4	θ_4	0	27	90°
5	θ_5	137	0	0

the end actuator mechanism of the manipulator in the global coordinate system from the initial position to the end position; $-135 \leq \varepsilon \leq 135$. The sum of the angles of joints 2, 3, and 4 is equal to the angle swept by the end actuator mechanism of the manipulator. The initial posture of the manipulator is upright. At this point, the distance from the end of the manipulator to the origin of the global coordinate system is the maximum rod length $l_{\max}$. When the joints are coordinated with each other, many kinds of posture are realized with the movement of the manipulator. However, the real distance l_{cur} from the end of the manipulator to the origin of the coordinate system of the base must not be longer than the length of the rod; otherwise, it will violate the physical rules. Furthermore, for the end of the manipulator to reach the target point, l_{cur} cannot be less than the distance l_{obj} of the target point to the base of the manipulator base coordinate system; otherwise, the end of the manipulator cannot reach the target point. In this study, the angular velocity of each steering gear is fixed, so angular velocity and acceleration are disregarded.

VI. SIMULATION RESULT ANALYSIS

A. SIMULATION PARAMETERS SETTING

The D-H parameters and joint variables are shown in Table 3.

Where θ is the joint angle that rotates around the Z axis, d is the distance between two adjacent connecting rods along the Z axis, a is the distance between two adjacent connecting rods along the X-axis, and α is the joint angle that rotates around the X axis. The parameters of the IEPSO algorithm, six-DOF manipulator model, and rewards and punishment rule are shown in Table 4.

The error value $E(\mathrm{e}_1, \mathrm{e}_2, \mathrm{e}_3, \ldots, \mathrm{e}_n)$ between the end of the manipulator and the target position is processed by rewards and punishment according to Equation (15). The domain of the rewards and punishment is $\lambda = [-1, 1]$, which indicates the degree to which the error is close to zero. The closer λ is to 1, the greater the degree of the error approaching zero. Conversely, the closer λ is to -1, the less the degree of the error approaching zero. $\lambda = 0$ indicates that the error satisfies the presupposition requirement, and ε is the evaluation index. The elements of $\lambda = [-1, 0]$ belong to the punishment domain, and the elements in $\lambda = (0, 1]$ belong to the reward domain.

$$\lambda = \begin{cases} -1/\varepsilon \times err_i + 1 & err_i \leq \varepsilon \\ -1/(err_{\max} - \varepsilon) \times err_i + \varepsilon/(err_{\max} - \varepsilon) & \varepsilon < err_i \leq e_{\max} \end{cases} \quad (15)$$

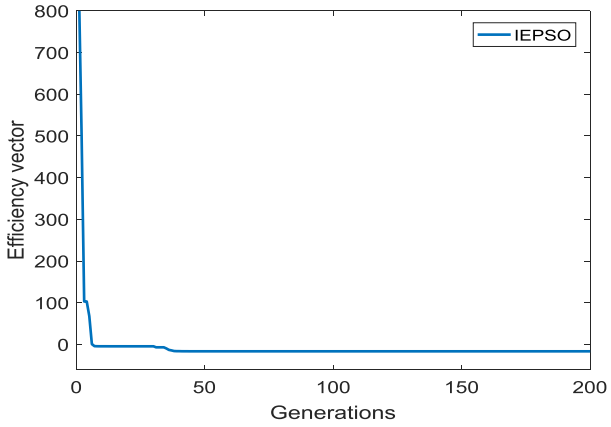

FIGURE 3. Efficiency scalar.

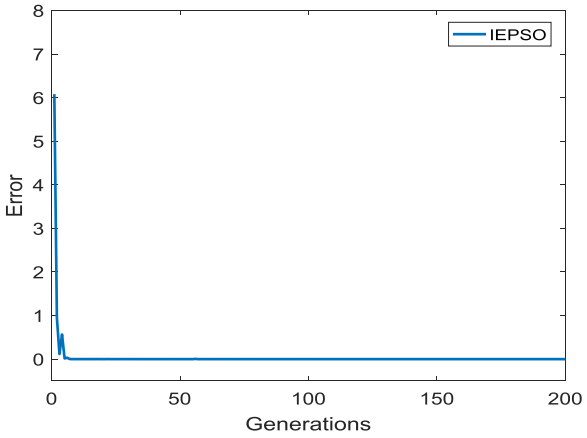

FIGURE 4. The error value between the target point and actuator mechanism.

where ε is the rewards and punishment waken threshold set to 0.005, err_{i} is the error between the target point and the end actuator of the manipulator, and $err_{\max}$ is the maximum error. Rewards and punishment with different reaching degrees are introduced into this article to prevent particles with small errors from being eliminated because of the longer time consumed to find the path than that consumed by optimal particles. With the optimal motion efficiency of the actuator regarded as the optimization target, the error err_{i} between the current position x_{i} of the particle and the target position x_{obj} is calculated and compared with the preset error value ε when the particle finds the target point. If $err_{\text{i}} > \varepsilon$, which indicates that the particle search path is poor, the particle will

TABLE 4. IEPSO algorithm, rewards and punishment rules and the manipulator model parameters.

Type	Value	Type	Value
Convergence precision	0.1e-06	Joint angle 1(θ_1)	-135°~135°
Population size	40	Joint angle 2(θ_2)	-90°~90°
Dimension	4	Joint angle 3(θ_3)	-135°~135°
Learning factor(C_1、C_2)	2	Joint angle 4(θ_4)	-135°~90°
Learning factor(C_3)	2~0	Joint angle 5(θ_5)	-180°~180°
Inertia weight(ω)	0.9~0.4	Minimum of rod l_{obj}	301mm
Random number(R_1、R_2、R_3)	[0,1]	Maximum of rod l_{max}	337mm
Iteration number(MaxIter)	1000	Reward and punishment waken threshold(ε)	0.005
Angular velocity(ω_0)	0.2	Reward waken threshold(s)	0.5e-06
Reward function 1(o)	-12~0	Reward function 2(u)	-24~-12

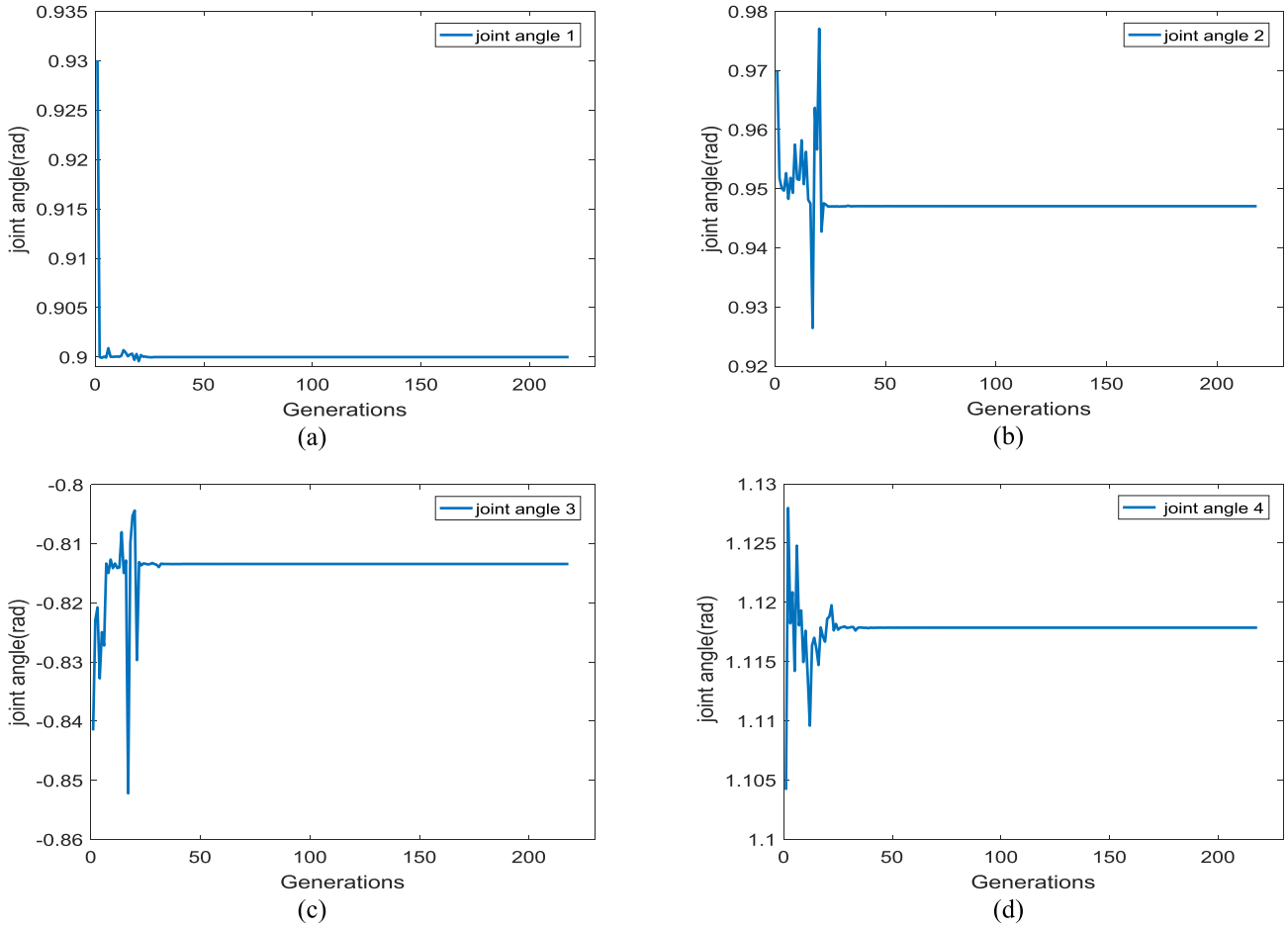

FIGURE 5. The error value between the target point and actuator mechanism. (a) Joint 1. (b) Joint 2. (c) Joint 3. (d) Joint 4.

be punished to increase the time consumed by the particle in this path. Consequently, the path that the particle finds is better and closer to the global optimal solution than the previous path. The particle is rewarded to reduce the time it consumes in this path. If the rewards function is used to reduce the fitness value of the target function and it is still larger than the optimal fitness value in the population, then the particle is abandoned and the optimal solution obtained by the current iteration is still selected. The degrees of rewards and punishment have an important impact on finding the global optimal solution. Too light or heavy rewards and punishment will affect the accuracy of the global search. If the punishments are too light, then the effect of the error will be ignored, and the actuator accuracy will be reduced. Conversely, if the punishment is too heavy when the particle is close to the edge of punishment, the particles will be forced to escape from the region. Therefore, piecewise rewards and punishment are adopted on the basis of the degree of error approaching zero. The rewards and punishment functions are expressed as (16) and (17).

$$\alpha = k \times (\exp(-5 \times (\lambda + 1)^2)) \tag{16}$$

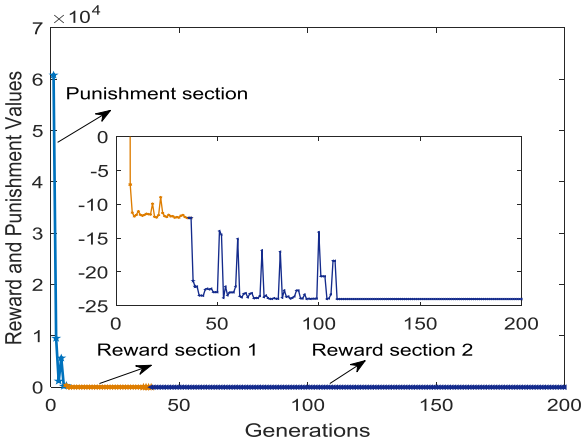

FIGURE 6. Reward and punishment function value.

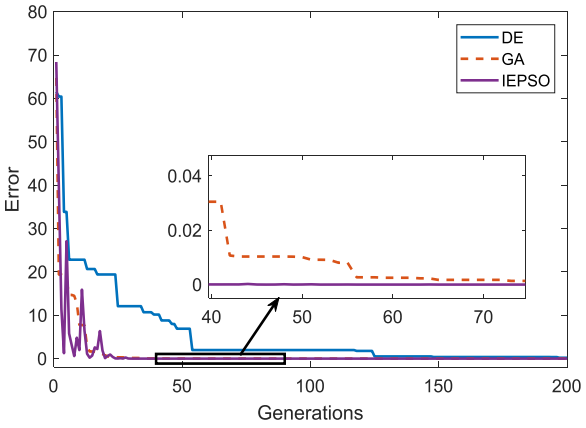

FIGURE 8. Error values between end actuator mechanism and target point of GA, DE and IEPSO.

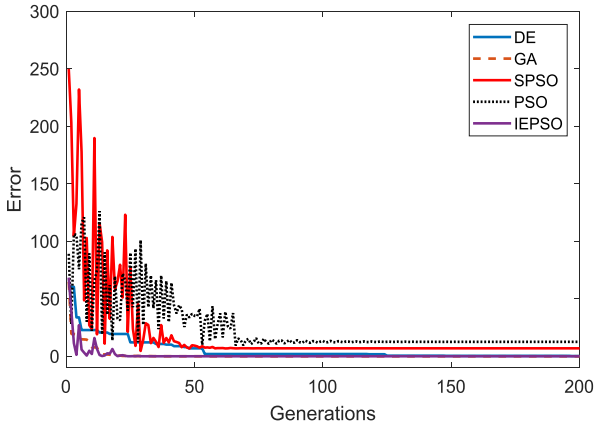

FIGURE 7. The error values of the five algorithms.

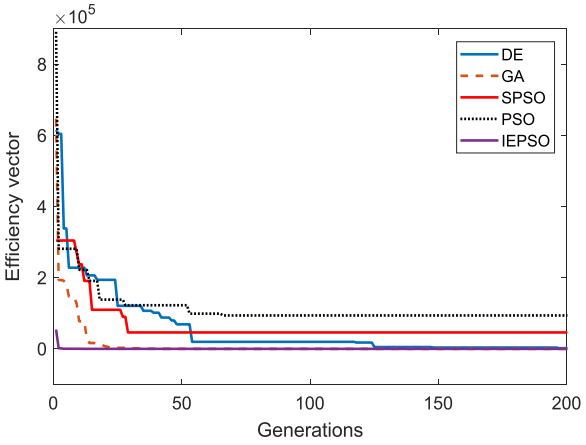

FIGURE 9. Efficiency vectors of five algorithms.

A reward is expressed in piecewise form as,

$$\beta_1 = -j \times ((-4/(\varepsilon - s)) \times err_i + 0.02/(\varepsilon - s)) \quad \lambda \in (0 \sim 0.99)$$
$$\beta_2 = -j \times ((-4/s) \times err_i + 8) \quad \lambda \in [0.99 \sim 1) \qquad (17)$$

where α is the punishment factor, β is the reward factor, λ is the degree of error approaching zero between the target point and the end actuator mechanism, and k and j are the proportional coefficients; In this study, $k = 10000$ and $j = 3$.

B. SIMULATION ANALYSIS

The IEPSO algorithm is used to search the global optimal solution from a number of feasible solutions. The trajectory planning for the 6-DOF robotic manipulator is completed quickly and efficiently. The initial position is [37, 0, 337], and the terminal position is [151.171, 191.62, 172.124].

Figure 3 shows the efficiency scalar using the direct trajectory planning algorithm, which converges around 45 generations. In order to improve the global optimization speed of IEPSO algorithm, Fuzzy rewards and punishment rules are applied to the error between the manipulator and the target point in the optimization objective function. Figure 4 shows the descent of the error between the target point and the end actuator mechanism of the manipulator. The position error is evidently close to zero at around the 50th generation, and the position error can reach zero around 180 generations. By adding Fuzzy rewards and punishment rules, the global optimization speed of the PSO can be improved, and the accuracy of the trajectory planning of the manipulator can be guaranteed. Figure 5 shows the motion angles of each joint. In order to verify the effectiveness of the algorithm, five optimization algorithms including PSO, SPSO, IEPSO, GA and DE are used for comparative analysis of position error and efficiency scalar.

As shown in Figure 6, when the error value is large, a large punishment value is generated. When the error value approaches the target value, a large reward value is generated. Improved the calculation efficiency of IEPSO by means of rewards and punishment. Figures 7 and 9 show that the position errors of the PSO and SPSO algorithms are larger than those of the proposed algorithm. Owing to their poor local search capability, the PSO and SPSO algorithms fall into the local extremum. Efficiency scalar and error values are relatively large, unable to meet the accuracy of multi-parameter and strongly coupled trajectory planning. As shown

TABLE 5. Variable values of five algorithms.

Algorithm	Variable1 (rad)	Variable2 (rad)	Variable3 (rad)	Variable4 (rad)	Efficiency scalar	Error (mm)
PSO	-2.24	-1.31	1.06	-1.11	8.71e+04	10.41
SPSO	0.89	0.12	1.47	-0.67	5.52 e+02	5.46
IEPSO	0.9	1.49	-0.23	1.17	-16.56	6.96-14
GA	0.9	0.92	0.84	-1.54	-3.59	3.06e-04
DE	0.9	0.09	1.13	-0.09	-2.57	0.12

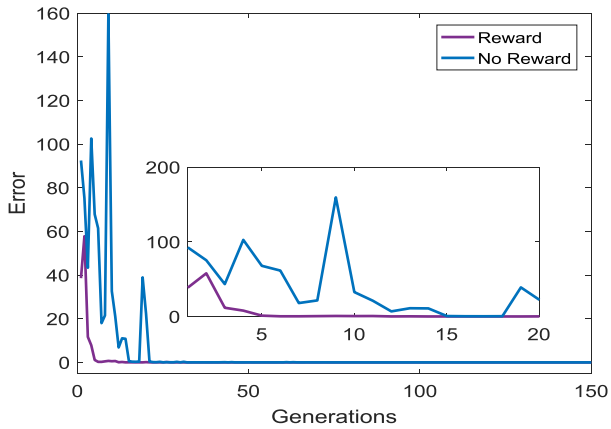

FIGURE 10. Comparison of reward function and no reward function.

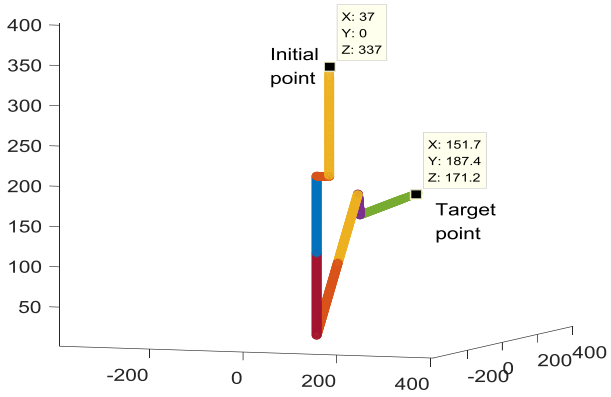

FIGURE 11. GUI image of the manipulator moving to target point.

in Figures 7–9, the IEPSO algorithm increases the information exchange between the global and the local and introduces the last elimination principle to balance the global search and local search capabilities of the PSO algorithm. Consequently, the proposed algorithm has strong global optimization capability. Therefore, the IEPSO algorithm converges faster than the GA and DE algorithm does. The position error of the GA optimization algorithm is 3.59e−04. The position error of DE optimization algorithm is 5.47e-03. The position error of IEPSO algorithm can reach zero in about 180 generations. With the introduction of the Fuzzy rewards and punishment rules, the proposed algorithm reduces computational time and improves the efficiency of optimization. The variables, error values, and efficiency scalar values of the four algorithms are shown in Table 5.

Figure 10 presents the convergence curve of the rules with and without reward. The error value of the reward function at the 19th iterations meets the presupposition precision requirement, and the error converges to zero around 130 generations. The optimization objective function with rewards and punishment function is able to converge asymptotically to zero because of the error value is softened by introducing the Fuzzy reward and punishment theory. When the asymptotic convergence of the objective function is satisfied, the convergence curve of the non-reward function error is particularly prominent and the oscillation is severe. As shown in Figure 10 and Table 6, compared with the non-reward rules, the speed of convergence with the reward rule is relatively fast, and the accuracy of the manipulator reaching the target point with the reward rule is relatively high. Figure 11 shows that the angle is included in the forward kinematics solution and is displayed in the GUI image. The position of the end-effector of the manipulator is the same as that of the target point. The initial position and the terminal position of the end of the manipulator are shown in Figure 11.

VII. CONCLUSION

A method for direct trajectory planning is proposed in this paper for multiparameter, strong-coupling and nonlinear multi-DOF manipulators. The direct trajectory planning method is based on the IEPSO algorithm and Fuzzy rewards and punishment theory. The last elimination principle and the local–global information-sharing term are introduced into the IEPSO algorithm to improve the global search capability of the algorithm. On the basis of the constraint conditions of multi-DOF manipulators, a multiparameter ''rewards and punishment'' constraint model is established. Through the direct trajectory planning method, computational time can be reduced while ensuring motion accuracy. The following conclusions can be drawn through the comparison and analysis of simulation results.

(1) The local–global information-sharing term φ 3 and the last elimination principle, which are introduced into the IEPSO algorithm, can maintain the diversity of the population and enhance the global search capability of the IEPSO algorithm.

(2) Fuzzy rewards and punishment rules based on the equality constraints and inequality constraints of multi-DOF manipulators are introduced. A comparison of the optimization results of the GA and the improved PSO algorithm suggests that the method can reduce the computational time and ensure that the robot manipulator can reach the target position accurately.

In summary, the results of the simulation analysis indicate that the accuracy of the manipulator trajectory obtained by IEPSO algorithm can reach 8.20e-12, and the accuracy can reach 0 after adding the fuzzy rewards and punishment rules into the IEPSO algorithm. As can be seen from Table 5, the IEPSO algorithm proposed in this paper is significantly more planning efficiency than the other three algorithms. The direct path planning method proposed in this paper is verified, which can effectively reduce a large number of decoupled operations, and can guarantee motion accuracy, which is of great application value for solving the trajectory planning problem of multi-parameter, strongly coupled nonlinear systems.

ACKNOWLEDGMENT

The authors would like to thank all the editor and the anonymous reviewers for their valuable suggestions that helped them improve this paper.

REFERENCES

[1] A. Gasparetto and V. Zanotto, "A new method for smooth trajectory planning of robot manipulators," *Mechanism Mach. Theory*, vol. 42, no. 4, pp. 455–471, 2007.
[2] R. Van Parys and G. Pipeleers, "Spline-based motion planning in an obstructed 3D environment," *IFAC World Congr.*, vol. 50, no. 1, pp. 8668–8673, 2017.
[3] H. Liu, X. Lai, and W. Wu, "Time-optimal and jerk-continuous trajectory planning for robot manipulators with kinematic constraints," *Robot. Comput.-Integr. Manuf.*, vol. 29, no. 2, pp. 309–317, 2013.
[4] E. K. Xidias, "Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces," *Robot. Comput.-Integr. Manuf.*, vol. 50, pp. 286–298, Apr. 2018.
[5] M. Wang, J. Luo, F. Fang, and J. Yuan, "Optimal trajectory planning of free-floating space manipulator using differential evolution algorithm," *Adv. Space Res.*, vol. 61, no. 6, pp. 1525–1536, 2018.
[6] M. Wang, J. Luo, J. Yuan, and U. Walter, "Coordinated trajectory planning of dual-arm space robot using constrained particle swarm optimization," *Acta Astronautica*, vol. 146, pp. 259–272, May 2018.
[7] R. Menasri, A. Nakib, B. Daachi, H. Oulhadj, and P. Siarry, "A trajectory planning of redundant manipulators based on bilevel optimization," *Appl. Math. Comput.*, vol. 250, pp. 934–947, Jan. 2015.
[8] M. D. G. Marcos, J. A. T. Machado, and T.-P. Azevedo-Perdicoúlis, "Trajectory planning of redundant manipulators using genetic algorithms," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 14, no. 7, pp. 2858–2869, 2009.
[9] H. Cao, S. Sun, K. Zhang, and Z. Tang, "Visualized trajectory planning of flexible redundant robotic arm using a novel hybrid algorithm," *Optik*, vol. 127, no. 20, pp. 9974–9983, 2016.
[10] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. Congr. Evol. Comput. (CEC)*, vol. 3, Jul. 1999, pp. 1945–1950.
[11] Z. Wang and J. Cai, "The path-planning in radioactive environment of nuclear facilities using an improved particle swarm optimization algorithm," *Nucl. Eng. Des.*, vol. 326, pp. 79–86, Jan. 2018.
[12] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, Jun. 2004.
[13] I. B. Aydilek, "A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems," *Appl. Soft Comput.*, vol. 66, pp. 232–249, May 2018.
[14] Y. Chen, L. Li, J. Xiao, Y. Yang, J. Liang, and T. Li, "Particle swarm optimizer with crossover operation," *Eng. Appl. Artif. Intell.*, vol. 70, pp. 159–169, Apr. 2018.
[15] K. Chen, F. Zhou, and A. Liu, "Chaotic dynamic weight particle swarm optimization for numerical function optimization," *Knowl.-Based Syst.*, vol. 139, pp. 23–40, Jan. 2017.
[16] Y. Chen, L. Li, H. Peng, J. Xiao, and Q. Wu, "Dynamic multi-swarm differential learning particle swarm optimizer," *Swarm Evol. Comput.*, vol. 39, pp. 209–221, Apr. 2018.
[17] F. Jiang, H. Xia, Q. A. Tran, Q. M. Ha, N. Q. Tran, and J. Hu, "A new binary hybrid particle swarm optimization with wavelet mutation," *Knowl.-Based Syst.*, vol. 130, pp. 90–101, Aug. 2017.
[18] R. Liu, J. Li, J. Fan, C. Mu, and L. Jiao, "A coevolutionary technique based on multi-swarm particle swarm optimization for dynamic multi-objective optimization," *Eur. J. Oper. Res.*, vol. 261, no. 3, pp. 1028–1051, 2017.
[19] R. S. Hartenberg and J. Denavit, *Kinematic Synthesis of Linkages*.

XUEYING LV was born in Changchun, China, in 1989. She is currently pursuing the Ph.D. degree with the College of Instrumentation and Electrical Engineering, Jilin University. Her research interests include satellite attitude and orbit integration control.

ZHAOXIA YU, photograph and biography not available at the time of publication.

MINGYANG LIU, photograph and biography not available at the time of publication.

GUANYU ZHANG was born in Changchun, China, in 1986. He received the Ph.D. degree in mechanical engineering from Jilin University, in 2015.

His research interests include satellite attitude and orbit integration control and space image processing technology research.

LIU ZHANG was born in Bengbu, Anhui, China, in 1978. He received the Ph.D. degree from the Harbin Institute of Technology, in 2007. He is currently a Professor with the College of Instrumentation and Electrical Engineering, Jilin University.

His research interests include aerospace optical remote sensing system design, simulation and application technology, and star sensor technology.