Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 3398810, 12 pages
https://doi.org/10.1155/2022/3398810

Research Article

Optimal Trajectory Planning for Minimizing Base Disturbance of a Redundant Space Robot with IQPSO

Jianxia Zhang,[1] **Fuchao Yang**,[1] **Guanghua Nie**,[1] **and Jianxin Zhang**[2]

[1]*School of Intelligent Engineering, Henan Institute of Technology, Xinxiang 453003, China*
[2]*School of Computer Science and Engineering, Dalian Minzu University, Dalian 116600, China*

Correspondence should be addressed to Jianxin Zhang; jxzhang0411@163.com

Received 26 February 2022; Revised 11 June 2022; Accepted 21 June 2022; Published 7 July 2022

Academic Editor: Yang Li

With the development of aerospace technology, the practical application of a free-floating redundant space robot has become more and more popular. The problem of minimizing base disturbance has been paid attention among academic researchers. If the space robot moves, it would have an impact on the pose of a base. The interference on a base should be reduced, which was caused by the movements of the space robot. In the paper, the simplified model of a redundant space robot has been described, which consists of a base and a 7-joint manipulator. Using the nonholonomic redundancy features, the pose of the base has been optimized planning. First, a set of kinematic equations of the redundant space robot was founded. Second, the 5-order polynomial function could be used for the parametric 7 joints. Third, on the basis of the pose requirements, a fitness function was defined. At last, the proposed improved quantum particle swarm optimization (IQPSO) algorithm was presented. The proposed IQPSO algorithm not only searched the optimal value easily but also had a good robust performance. The advantages could be shown through the numerical experiments, compared with the quantum-behaved particle swarm optimization (QPSO) algorithm, particle swarm optimization (PSO) algorithm, and simulated annealing particle swarm (SAPSO) algorithm. Then, the proposed IQPSO algorithm was used to optimize the fitness function of trajectory planning. By the simulation results, it could be confirmed that the proposed IQPSO algorithm searched the global optimal solution not only easily but also smoothly, compared with the QPSO, PSO, and SAPSO algorithms. The proposed approach was suitable for planning an optimal trajectory.

1. Introduction

With the development of aerospace technology, the practical application of the free-floating redundant space robot has become more and more popular [1, 2]. The problem of minimizing base disturbance has been paid attention among academic researchers [3]. Planning an appropriate path has been an important mission of the redundant space robot [4, 5]. It was comprised of at least one manipulator and a base. There are four strategies to control the base, including fixed, mobile, free-flying, and free-floating [6]. The base of the redundant space robot was not under control. The redundant space robot has non-holonomic constraints. Its end pose was related to the current joint and the history of movement [7].

Nowadays, more and more researchers have discussed the problem of stability of the base [8]. In 1991, a method named enhancing interference diagram was proposed. It could reduce the attitude disturbance. However, it had a low computational speed and a larger memory space [9]. According to the nonholonomic feature, Vafa and Dubowsky had proposed a self-correcting motion method in 1993. This method could only adjust the base attitude, but the final state of joints could not be changed [10]. Shi et al. had put forward a path planning method based on quantum-behaved particle swarm optimization (QPSO) to solve the problem of the global path planning for a mobile robot in 2010. It could not only search the shortest path but also avoid the obstacles effectively [11]. To overcome the problem of path planning for a soccer robot, Meng et al. had proposed a method. It overcame shortcomings of the traditional soccer robot's slow actions [12]. In 2015, Hu et al. had put forward a method that was based on quantum-behaved particle swarm optimization (QPSO) for trajectory planning to optimize base disturbance. It had not more parameters and searched quickly [13]. In order to reduce base disturbance, a method was proposed by Xiangxin Zeng. It used the Gauss pseudo general method for path planning. The movement trajectory obtained by this method was not only continuous but was also smooth [14]. A path planning method based on simulated annealing particle swarm (SAPSO) was put forward by Zhang et al. in 2016. It was proposed to reduce the interaction between the base and the space manipulator [15]. In 2018, Hu and Wang proposed a method to optimize the minimum base reaction. It was verified by the 3 DOF and 7 DOF space robot [16]. In 2017, Zhang et al. put forward a method based on multiswarm particle swarm optimization (PSO), which was proposed to solve the problem of base disturbances for a 6 DOF space manipulator [17]. In 2015, a motion planning method based on SAPSO was presented by Ji et al. to reduce the base disturbance [18]. Due to the complexity and difference in the problem of base disturbance, it has not been completely solved until now.

In the paper, an approach for optimal trajectory planning with proposed improved quantum particle swarm optimization (IQPSO) was studied. The model of the redundant space robot system was built. First, a set of kinematic equations of the redundant space robot was found. Second, the 5-order polynomial could be used to parameterize the 7 joints. Third, on the basis of the pose requirements, a fitness function was defined. At last, the proposed improved quantum particle swarm optimization (IQPSO) algorithm was presented and used to optimize the fitness function of trajectory planning. By the simulation results, it can be confirmed that the proposed IQPSO algorithm searched the global optimal solution not only easily but also smoothly, compared with the QPSO, PSO, and SAPSO algorithms.

The organizational structure of the paper is as follows: firstly, it is the introduction. Secondly, the kinematic equations are described. Thirdly, the IQPSO algorithm is stated in detail. The numerical and engineering application experiments are provided in the fourth part. Lastly, it is the conclusions.

2. Description of Kinematic Equations

As it is shown in Figure 1, the simplified model of the space robot system included a base and a 7-joint manipulator.

The D-H parameters are described in Table 1.

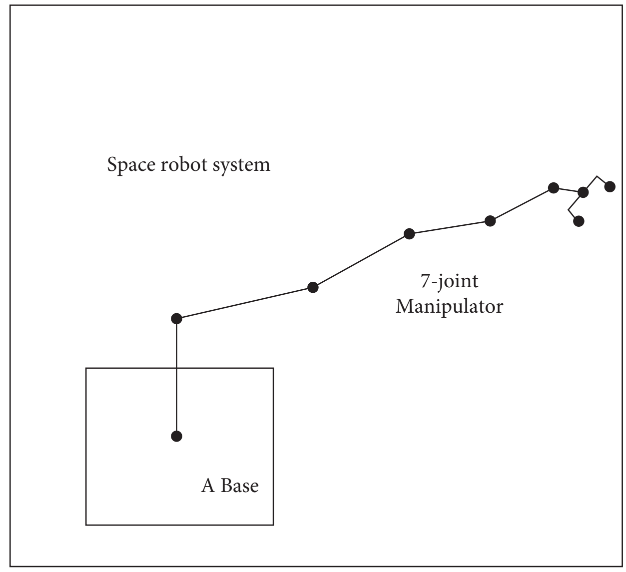

Figure 1: The simplified model of the space robot system.

Table 1: The D-H parameters of the model.

k	α_{k-1} (rad)	a_{k-1}	d_k	θ_k
1	$-\pi/2$	0	d_1	θ_1
2	$\pi/2$	0	0	θ_2
3	$-\pi/2$	0	d_3	θ_3
4	$\pi/2$	0	0	θ_4
5	$-\pi/2$	0	d_5	θ_5
6	$\pi/2$	0	0	θ_6
7	0	0	d_7	θ_7

2.1. Expression of the Space Manipulator Pose. The pose consisted of the position and the attitude. The position can be expressed as shown in the following formula:

$$S_p = \begin{bmatrix} p_x & p_y & p_z \end{bmatrix}^T, \tag{1}$$

where p_x, p_y, and p_z represent the three parts of point P in $\{S\}$.

In the paper, the expression of quaternion had been adopted to express the attitude. It was proposed by Jack to express the attitude in 1992 [19], as shown in the following formula:

$$\begin{cases} Q = \eta + q_1 \vec{i} + q_2 \vec{j} + q_3 \vec{k} = \eta + q, \\ \eta^2 + q_1^2 + q_2^2 + q_3^2 = 1. \end{cases} \tag{2}$$

where Q is the shorthand for quaternion, η is the scalar part, and q is the vector part.

The relations described among the parameters are as follows:

$$\begin{aligned} \begin{bmatrix} \dot{\eta} \\ \dot{q} \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} 0 & -\omega^T \\ \omega & -\tilde{\omega} \end{bmatrix} \begin{bmatrix} \eta \\ q \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} -q^T \\ \eta E - \tilde{q} \end{bmatrix} \omega. \end{aligned} \tag{3}$$

The attitude error described by the quaternion is as follows:

$$\begin{cases} \delta\eta_b = \eta_{b0}\eta_{bf} + q_{b0}^T q_{bf}, \\ \delta q_b = \eta_{b0} q_{bf} - \eta_{bf} q_{b0} - \widetilde{q_{b0} q_{bf}}. \end{cases} \tag{4}$$

2.2. Establishment of Kinematic Equations. The position of an end-effector can be described in the following formula:

$$p_e = r_0 + b_0 + \sum_{k=1}^{7} (p_{k+1} - p_k), \tag{5}$$

where p_e indicates the position of an end-effector, r_0 indicates the position of the base center, and b_0 represents the base vector.

The velocity of the end-effector was calculated through above formula (5), which is as shown in the following formula:

$$v_e = \dot{p}_e = v_0 + \omega_0 \times (p_e - r_0) + \sum_{k=1}^{7} \left[k_j \times (p_e - p_k)\right] \cdot \dot{\theta}_k, \tag{6}$$

where v_e indicates the velocity of the end-effector, v_0 and ω_0 indicate the velocity and the angular velocity of the base, respectively, andθ_k indicates the matrix of each joint.

The angular velocity of the end-effector could be expressed in the following formula:

$$\omega_e = \omega_0 + \sum_{k=1}^{7} k_k \dot{\theta}_k, \tag{7}$$

where ω_e represents the angular velocity of the end-effector.

The kinematic equation of the 7-joint space manipulator can be described in the following formula:

$$\begin{bmatrix} v_e \\ \omega_e \end{bmatrix} = J_b \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} + J_s \dot{\theta}, \tag{8}$$

where J_b represents the Jacobian matrix of the base and J_s is the Jacobian matrix of the space manipulator.

Therefore, v_0 and ω_0 are described in the following formula:

$$\begin{aligned} \begin{bmatrix} v_0 \\ \omega_0 \end{bmatrix} &= -I_b^{-1} I_{bs} \dot{\theta} \\ &= \begin{bmatrix} J_{vb} \\ J_{wb} \end{bmatrix} \dot{\theta}, \end{aligned} \tag{9}$$

where I_b indicates the inertia matrix of the base, I_{bs} indicates the coupled inertia matrix, andJ_{vb} is the component Jacobian matrix about v_0, and $J_{\omega b}$ is the component Jacobian matrix about ω_0.

At last, the calculation result is expressed in the following formula:

$$\begin{aligned} \begin{bmatrix} v_e \\ \omega_e \end{bmatrix} &= \left[J_s - J_b I_b^{-1} I_{bs}\right] \dot{\theta} \\ &= J^* (\psi_b, \theta, m_i, I_i) \dot{\theta}, \end{aligned} \tag{10}$$

where $J^* (\psi_b, \theta, m_i, I_i)$ indicates the generalized Jacobian matrix of the 7-joint space manipulator.

2.3. Equations of the Space Robot System. The equation of the space robot system can be expressed in the following formula:

$$X_b = \begin{bmatrix} Q_b \\ P_b \end{bmatrix}, \tag{11}$$

where X_b indicates the pose of the base, Q_b indicates the attitude of the base, and P_b indicates the position of the base.

This equation of the space robot system can be calculated by numerical integration as shown in the following formulas: (12)(13)

$$\mathbf{Q}_b(t) = \int_0^t \frac{1}{2} \begin{bmatrix} -q_b^T \\ \eta_b I - \tilde{q}_b \end{bmatrix} J_{bs\_\omega} \dot{\theta} \mathrm{d}t, \tag{12}$$

$$\mathbf{P}_b = \int_0^t J_{bs\_v} \dot{\theta} \mathrm{d}t. \tag{13}$$

Formula (12) was used to update the attitude by the generalized Jacobian matrix. Similarly, formula (13) was used to update the position.

The purpose of optimal trajectory planning was to reduce the base disturbance, as shown in the following formula:

$$\left\| X_{b0} - X_{bf} \right\| \longrightarrow 0, \tag{14}$$

where X_{b0} and X_{bf} are the initial and final pose of the base, respectively, during the time- $[0, t_f]$.

2.4. Description of Trajectory Planning. The description of trajectory planning should be smooth, which can be described as shown in the following formulas: (15)(17)

$$\theta_i(0) = \theta_{i0} \quad \theta_i(t_f) = \theta_{id}, \tag{15}$$

$$\dot{\theta}_i(0) = \ddot{\theta}_i(0) \quad \dot{\theta}_i(t_f) = \ddot{\theta}_i(t_f), \tag{16}$$

$$\theta_{i\min} \le \theta_i(t) \le \theta_{i\max}, \quad i = 1, 2, \ldots, 7, \tag{17}$$

where θ_{i0} indicates the initial angle of each joint, $\theta_{i\,d}$ indicates the final angle of each joint, $\theta_{i\min}$ indicates the minimum angle of each joint, and $\theta_{i\max}$ is the maximum angle of each joint.

The range angle of each joint could be directly restrained by the sinusoidal function. So it was used to parameterize the sinusoidal function. The angle of each joint can be parameterized by the five-order function, which is as shown in the following formula:

$$\theta_i(t) = \alpha_{i1} \sin\left(\beta_{i5} t^5 + \beta_{i4} t^4 + \beta_{i3} t^3 + \beta_{i2} t^2 + \beta_{i1} t + \beta_{i0}\right) + \alpha_{i2}, \tag{18}$$

$$\begin{aligned} \alpha_{i1} &= \frac{\theta_{i\max} - \theta_{i\min}}{2}, \\ \alpha_{i2} &= \frac{\theta_{i\max} + \theta_{i\min}}{2}, \end{aligned} \tag{19}$$

where $i = 1, 2, \ldots, 7$; $\beta_{i0} \sim \beta_{i7}$ are the coefficients to be determined and α_{i1} and α_{i2} are determined in accordance with the angle of each joint.

The descriptions of the angle, velocity, and acceleration are shown in the following formulas: (20)(22)

$$\theta(t) = \alpha_{i1} \sin\left(\beta_{i5}t^5 - \frac{5}{2}\beta_{i5}t_f t^4 + \frac{5}{3}\beta_{i5}t_f^2 t^3 + \sin^{-1}\left(\frac{\theta_{i0} - \alpha_{i2}}{\alpha_{i1}} \right) \right) + \alpha_{i2}. \tag{20}$$

$$\dot{\theta}(t) = \alpha_{i1} \cos\left(\beta_{i5}\left(t^5 - \frac{5}{2}t_f t^4 + \frac{5}{3}t_f^2 t^3 \right) + \sin^{-1}\left(\frac{\theta_{i0} - \alpha_{i2}}{\alpha_{i1}} \right) \right)\left(\beta_{i5}\left(5t^4 - 10t_f t^3 + 5t_f^2 t^2 \right) \right), \tag{21}$$

$$\begin{aligned} \ddot{\theta}_i(t) = &-\alpha_{i1} \sin\left(\beta_{i5}\left(t^5 - \frac{5}{2}t_f t^4 + \frac{5}{3}t_f^2 t^3 \right) + \sin^{-1}\left(\frac{\theta_{i0} - \alpha_{i2}}{\alpha_{i1}} \right) \right)\left(\beta_{i5}\left(5t^4 - 10t_f t^3 + 5t_f^2 t^2 \right) \right)^2 \\ &+ \alpha_{i1} \cos\left(\beta_{i5}\left(t^5 - \frac{5}{2}t_f t^4 + \frac{5}{3}t_f^2 t^3 \right). + \sin^{-1}\left(\frac{\theta_{i0} - \alpha_{i2}}{\alpha_{i1}} \right) \right)\left(\beta_{i5}\left(20t^3 - 30t_f t^2 + 10t_f^2 t \right) \right). \end{aligned} \tag{22}$$

According to above formulas (15) and (16), the parameters can be obtained in the following formula: (23)

$$\begin{aligned} \beta_{i0} &= \sin^{-1}\left(\frac{\theta_{i0} - \alpha_{i2}}{\alpha_{i1}} \right), \\ \beta_{i1} &= \beta_{i2}, \\ \beta_{i3} &= \frac{5}{3}\beta_{i5}t\frac{2}{f}, \\ \beta_{i4} &= -\frac{5}{2}\beta_{i5}t_f. \end{aligned} \tag{23}$$

Eventually, it was only one parameter β undetermined. It is described in the following formula:

$$\beta = [\beta_{15}, \beta_{25}, \beta_{35}, \beta_{45}, \beta_{55}, \beta_{65}, \beta_{75}], \tag{24}$$

and the trajectory planning should be obtained if the only one parameter "β" could be fixed.

2.5. Description of Fitness Function. The only one parameter "β" was an argument. The fitness function can be described in the following formula:

$$F(\beta) = \frac{\|\delta q_b\|}{J_q} + \frac{\|\delta p_b\|}{J_p} + \frac{L_{\dot{\theta}}}{J_{\dot{\theta}}} + \frac{L_{\ddot{\theta}}}{J_{\ddot{\theta}}}, \tag{25}$$

where $\|x\| = \sqrt{x^T \cdot x}$; δq_b represents the error of the base attitude, δp_b indicates the error, $L_{\dot{\theta}}$ indicates the constraint conditions of the velocity, $L_{\ddot{\theta}}$ indicates the constraint conditions of acceleration, and J_q, J_p, $J_{\dot{\theta}}$, and $J_{\ddot{\theta}}$ indicate the weighting coefficients, which were calculated by precision demand.

The problem of trajectory planning in this paper was expressed to solve the unknown parameter a. Formula (25) should be minimized through the optimal parameter obtained by the algorithm.

3. Improved QPSO Algorithm

3.1. Overview of the QPSO Algorithm. The QPSO algorithm had originated in the quantum theory with the superposition and probability [20]. Every single particle could be expressed in many different forms through the property of the superposition state. The diversity of population was increased by this strategy. Besides, the current situation of each particle was described by a given probability through the other property. Therefore, the QPSO algorithm could find the global optimal value. Furthermore, it had not more parameters [21]. However, the QPSO algorithm also had its own shortcomings. For example, the particles might appear in cluster formation. In other words, the particles would gather one or several positions, which made it easily trapped into local optimum [22].

The QPSO algorithm was composed of M-particles, which represented the solutions of problem. At the moment, the position of ith particle is described in the following formula:

$$X_i(t) = [X_{i1}(t), X_{i2}(t), \ldots, X_{iN}(t)], \quad i = 1, 2, \ldots, A, \tag{26}$$

where A indicates the amount of particles and t indicates the amount of iterations.

It did not have the velocity vector for the particles. The individual best position can be described in the following formula:

$$P_i(t) = [P_{i1}(t), P_{i2}(t), \ldots, P_{iN}(t)], i = 1, 2, \ldots, A. \tag{27}$$

The global best position is described inthe following formula:

$$G_i(t) = [G_{i1}(t), G_{i2}(t), \ldots, G_{iN}(t)], i = 1, 2, \ldots, A. \tag{28}$$

As is known to all, the smaller the fitness function value, the better the relative fitness value. The individual best position of the ith best particle is determined in the following formula:

$$P_i(t) = \begin{cases} P_i(t-1) & \text{if } f[X_i(t)] \geq f[P_i(t-1)], \\ X_i(t) & \text{if } f[X_i(t)] < f[P_i(t-1)]. \end{cases} \tag{29}$$

The global best value can be determined in the following formula:

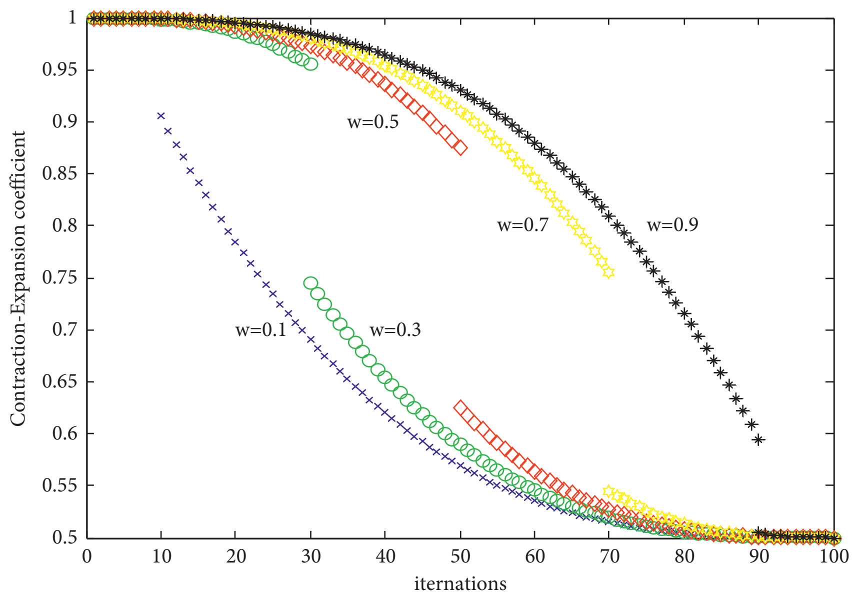

Figure 2: The changing curves of λ.

$$\begin{cases} g = \arg\min\limits_{1\le i\le A}\{f[P_i(t)]\}, \\ G(t) = P_g(t), \end{cases} \tag{30}$$

where $g \in \{1, 2, \ldots, A\}$ indicates the index of particles, which is the global best position.

The evolutionary equations of particles could be described in the following formulas (31)(32):

$$\begin{cases} p_{ij}(t) = \varphi_j(t)\cdot P_{ij}(t) + \left[1-\varphi_j(t)\right]\cdot G_j(t), \\ \varphi_j(t) \in U(0,1), \end{cases} \tag{31}$$

$$\begin{cases} X_{ij}(t+1) = p_{ij}(t) \pm \lambda\cdot\left|C_j(t) - X_{ij}(t)\right|\cdot\ \ln\left[1/u_{ij}(t)\right], \\ u_{ij}(t) \in U(0,1), \end{cases} \tag{32}$$

where $j = 1, 2, \ldots, 7$ represents the dimension, λ is the contraction-expansion coefficient, and $C_j(t) = (1/A)\sum_{i=1}^{A} P_{ij}(t)$, which indicated the average best position.

3.2. Improved QPSO Algorithm. To make up for the weaknesses of the QPSO algorithm, such as the low convergence speed and easily being trapped in a local optimum [23], the improved QPSO (IQPSO) algorithm was proposed. Except the population size and iterations, the parameter λ was the unique control parameter. It should be adjusted. According to the search principle, the parameter λ was relatively large in the initial stage of the algorithm. It could help the particles search for the optimal solution in a larger space. Along with the iterations, it would be better if the parameter λ turned out to be smaller. It could enable the particles to find the final optimal solution within a smaller space easily. By changing the parameter λ, it could improve the convergence speed and avoid appearing premature. The improved contraction-expansion coefficient λ is described in the following formula:

$$\lambda = \begin{cases} \text{a} - \dfrac{(a-b)\cdot t^3}{w\cdot N^3}, & t \le w\cdot N, \\ b + \dfrac{(a-b)\cdot (N-t)^3}{(1-w)\cdot N^3}, & t > w\cdot N. \end{cases} \tag{33}$$

Where $a = 1, b = 0.5$, N indicates the maximum amount of iterations, t is the current amount of iteration, and w indicates a plus quantity between (0, 1).

The changing curves of λ were along with the different w as shown in Figure 2.

A process of the IQPSO algorithm is shown in following Figure 3.

In the paper, the IQPSO algorithm was proposed to solve the unknown parameter β. Then, the optimal trajectory planning of the 7-joint space robot could be determined, which was under the constraint condition of minimizing base disturbance.

The partial coding scheme for the IQPSO is presented in Figure 4.

4. Experiments and Simulations

As is known to all, MATLAB was originally developed by the MathWorks company in the United States. It was mainly used for mathematical software. However, with the development of this software, it played an important role in various fields. In order to make a fair comparison, the numerical experiments and trajectory planning experiments have been performed on the MATLAB platform in this paper.

4.1. Numerical Experiments. To fully prove the proposed IQPSO algorithm well, the numerical experiments with four standard test functions were performed firstly. The results

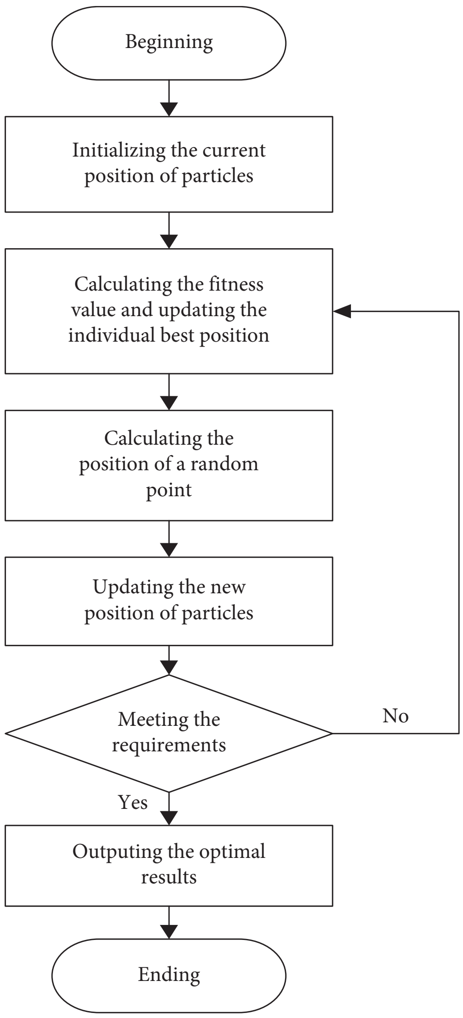

FIGURE 3: Process of the IQPSO algorithm.

```
for run=1:runno
    run
    x=(irange_r-irange_1)*rand(popsize,dimension)+irange_1;
    x=randn(popsize,dimension);
    pbest=x;
    for i=1:popsize
        f_x(i)=fitness(x(i,:));
        f_pbest(i)=f_x(i);
    end
    g=min(find(f_pbest==min(f_pbest(1:popsize))));
    gbest=pbest(g,:);
    f_gbest=f_pbest(g);
    MINIMUM=f_pbest(g);
    for t=1:MAXITER
    if t<=0.5*MAX
    alpha=1.0-((1.0-0.5)/(0.5* itmax^3))*t^3;
    else
        alpha=0.5+((1.0-0.5)/((1-0.5)*itmax^3))*((itmax-t)^3);
    end
```

FIGURE 4: Partial coding scheme for the IQPSO algorithm.

obtained from the IQPSO algorithm were compared with the QPSO algorithm, PSO algorithm, and SAPSO algorithm. The four standard test functions are shown in Table 2 [24].

The parameters of different algorithms included number of particles M, current iterations t, maximum iterations N, dimension D, a plus quantity w, acceleration coefficients c_1 and c_2, and annealing constant l. They are set as follows:

$$\begin{aligned} & M = 30, N = 100, D = 10, w_{pso} = 0.5, w_{IQPSO} = 0.1, \\ & c_1 = c_2 = 2, l = 0.5. \end{aligned} \tag{34}$$

In this paper, the experiments run independently fifty times. The numerical simulation results are shown in Table 3, including the best and worst value. The computing time of running experiments independently fifty times is shown in Table 4, which was measured in seconds.

In Tables 3 and 4, the optimal results are indicated in bold. From Table 3, it could be seen that the proposed IQPSO algorithm performed better on solving most functions, compared with the QPSO algorithm, PSO algorithm, and SAPSO algorithm. From Table 4, we could see that the computing time of the proposed IQPSO algorithm was least for most standard functions. The computing time was relatively fast. It took less than a second to run once. Through the numerical simulation results and analysis, it was obvious that the proposed IQPSO algorithm was easy to search for the best value. It has a good robust performance compared with other algorithms.

4.2. Trajectory Planning Experiments. To further prove the proposed IQPSO algorithm, it was used to solve the optimal trajectory planning for minimizing base disturbance of the redundant space robot. The problem of trajectory planning has been converted into a mathematical solution, which was previously stated in this paper in detail. The parameters of IQPSO setting are in the following formula:

$$M = 30, \; N = 100, \; D = 7, \; w = 0.5. \tag{35}$$

The fitness function was solved through the proposed IQPSO algorithm, QPSO algorithm, PSO algorithm, and SAPSO algorithm separately.

The optimal parameters and the fitness value were obtained by the IQPSO algorithm as shown in the following formula:

$$\begin{cases} \beta_{\text{IQPSO}} = 1.0e^{-4} * \begin{bmatrix} -0.001729319115994; -0.000026105823252; \\ 0.002958454474523; 0.000167909227119; \\ -0.173692323936252; 0.003200377293701; \\ -0.205714762901253 \end{bmatrix}, \\ f(\beta)_{\text{IQPSO}} = 1.223010798218512e^{-10}. \end{cases} \tag{36}$$

The position of the base changes with the time as shown in Figure 5.

From Figure 5, we can see that the position of the base was $[-0.000000033621718; \; -0.081296045086891; \; -0.188036992457903]$.

Table 2: The four standard test functions.

	Standard function	Search range	$f_{\min}$
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0
Rastrigin	$f_2(x) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-10, 10]^D$	0
Griewank	$f_3(x) = (1/4000)\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(x_i/\sqrt{i}) + 1$	$[-50, 50]^D$	0
Schwefel	$f_4(x) = \sum_{i=1}^{D} \lvert x_i\rvert + \prod_{i=1}^{D} \lvert x_i\rvert$	$[-10, 10]^D$	0

Table 3: The optimal results.

Standard functions	QPSO	PSO	SAPSO	IQPSO proposed
f_1 best	$4.9868e-10$	$6.3046e-05$	0.4276	**7.9168 e − 12**
Worst	$5.0589e-07$	0.0235	3.9898	**2.7583 e − 07**
f_2 best	1.0309	2.1960	30.3178	**1.0254**
Worst	28.3054	**21.2206**	74.0135	25.4100
f_3 best	$3.9784e-10$	$6.0854e-06$	0.0950	**8.9720 e − 11**
Worst	0.0078	0.0048	0.4476	**0.0041**
f_4 best	$2.0759e-05$	0.0492	1.7189	**3.0474 e − 06**
Worst	$6.9477e-04$	0.7856	5.4503	**1.2442 e − 04**

The text in bold means the optimal results achieved by the four algorithms.

Table 4: The computing time.

Standard functions	QPSO	PSO	SAPSO	IQPSO proposed
f_1	1.447	1.266	1.265	**1.263**
f_2	1.337	1.648	1.289	**0.916**
f_3	**0.901**	1.338	1.607	1.410
f_4	1.437	1.588	1.592	**1.434**

The text in bold means the optimal results achieved by the four algorithms.

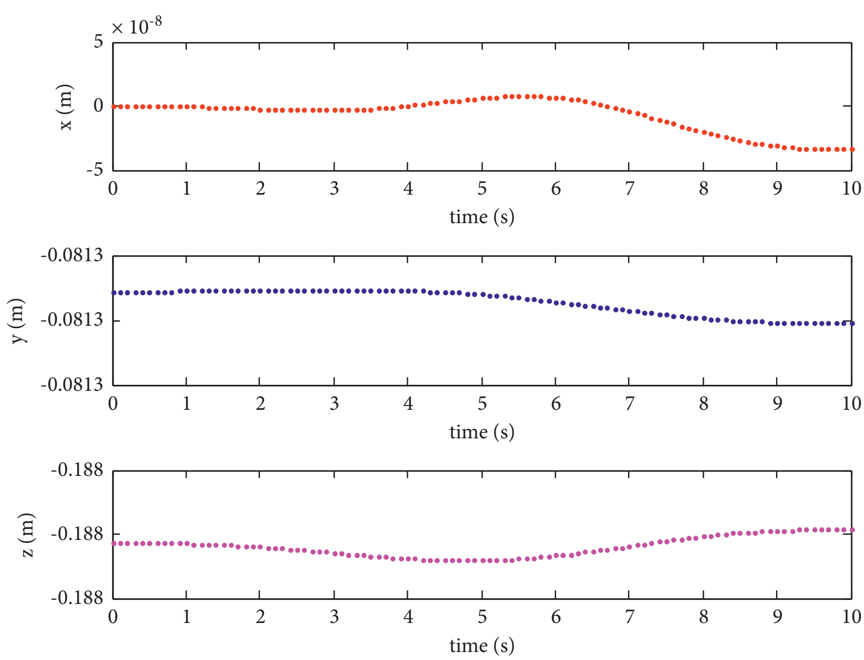

Figure 5: The position of the base obtained by the IQPSO algorithm.

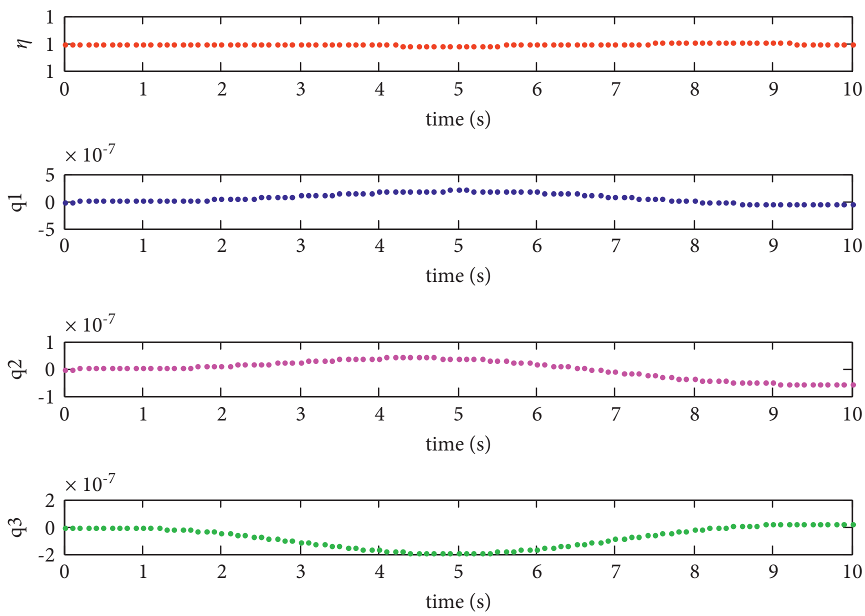

Figure 6: The attitude of the base obtained by the IQPSO algorithm.

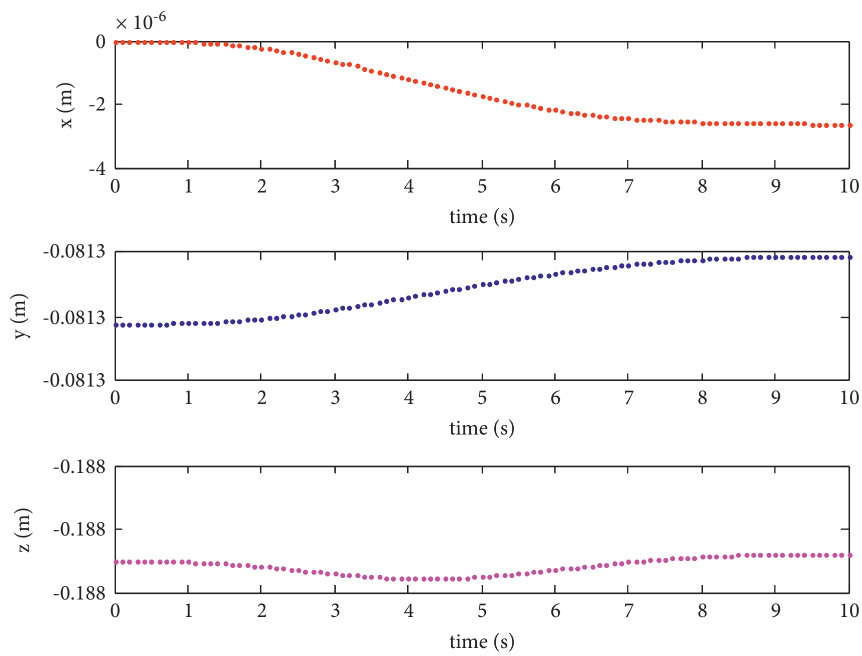

Figure 7: The position of the base obtained by the QPSO algorithm.

The attitude of the base changes along with the time as shown in Figure 6.

From Figure 6, it can be seen that the attitude of the base was [1.000000000000000; −0.000000060486705; −0.000000055551370; 0.000000025741104].

The optimal parameters and the fitness value were obtained by the QPSO algorithm as shown in the following formula:

$$\begin{cases} \beta_{QPSO} = 1.0e^{-4} * \begin{bmatrix} -0.053727145056947; 0.000014734907040; \\ 0.069479886180796; -0.000187541265537; \\ 0.370120344884837; 0.024346823909793; \\ 0.072405199767136 \end{bmatrix}, \\ f(\beta)_{QPSO} = 1.787582295257875e^{-8}. \end{cases} \quad (37)$$

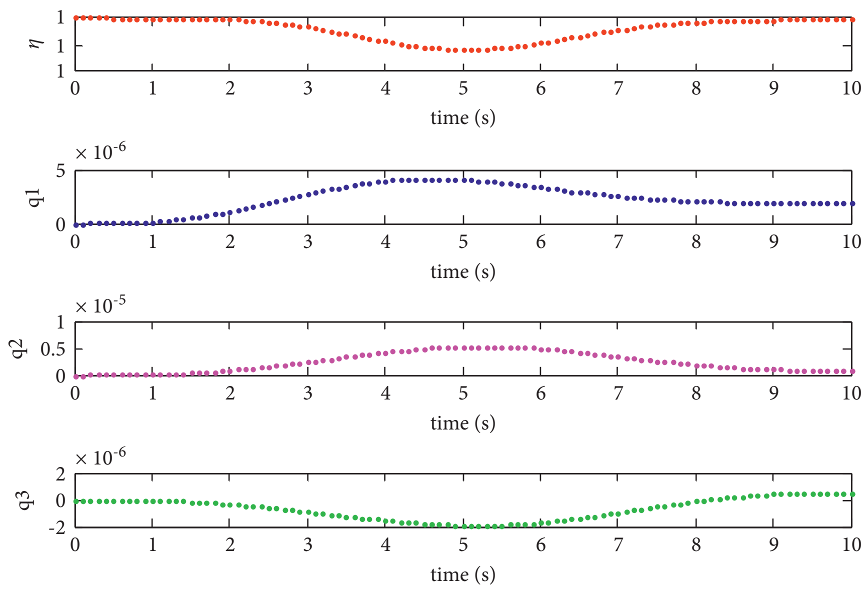

FIGURE 8: The attitude of the base obtained by the QPSO algorithm.

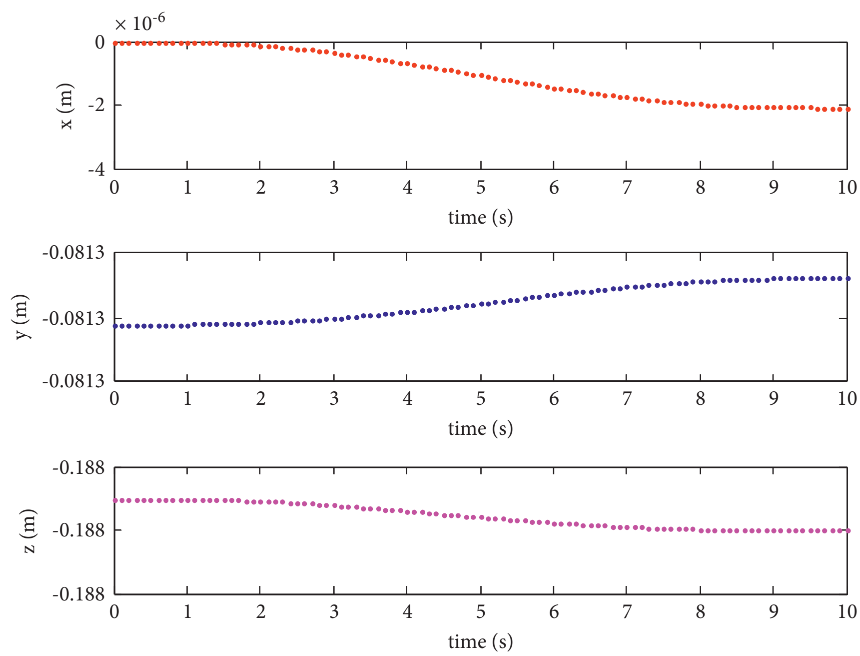

FIGURE 9: The position of the base obtained by the PSO algorithm.

The position of the base changes with the time as shown in Figure 7.

From Figure 7, we can see that the position of the base was [−0.000002614389786; −0.081290281543783; −0.188036781531653].

The attitude of base changes along with the time as shown in Figure 8.

From Figure 8, it can be seen that the attitude of the base was [0.999999999998799; 0.000001894650367; 0.000000896566092; 0.000000506153035].

The optimal parameters and the fitness value were obtained by the PSO algorithm as shown in the following formula:

$$\begin{cases} \beta_{PSO} = 1.0e^{-5} * \begin{bmatrix} -0.209594516686051; -0.000210216283730; \\ 0.273603505071900; 0.001617425203244; \\ 0.259691788163232; -0.098117637819463; \\ -0.963511035955538 \end{bmatrix}, \\ f(\beta)_{PSO} = 1.614561098924839e^{-8}. \end{cases} \tag{38}$$

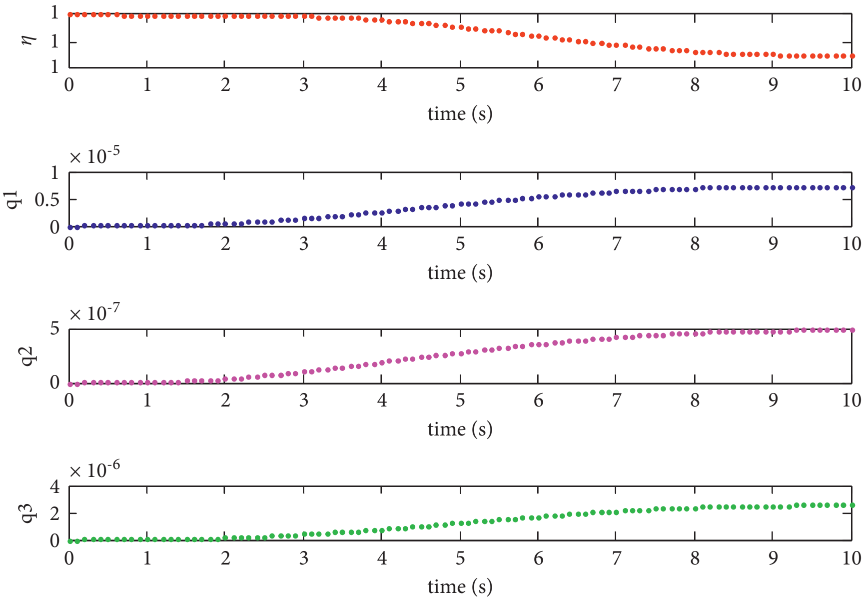

Figure 10: The attitude of the base obtained by the PSO algorithm.

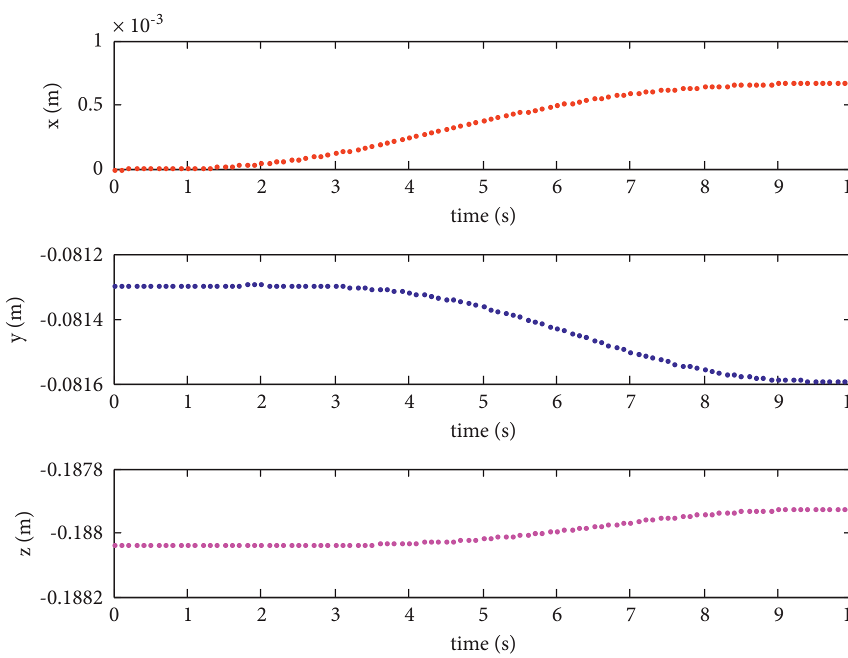

Figure 11: The position of the base obtained by the SAPSO algorithm.

The position of the base changes with the time as shown in Figure 9.

From Figure 9, it can be seen that the position of base was [−0.000002082241332; −0.081291931890445; −0.188038015952370].

The attitude of the base changes along with the time as shown in Figure 10.

From Figure 10, we can see that the attitude of the base was [0.999999999970260; 0.000007323764193; 0.000000485141535; 0.000002545858789].

The optimal parameters and the fitness value were obtained by the SAPSO algorithm as shown in the following formula:

$$\begin{cases} \beta_{SAPSO} = 1.0e^{-4} * \begin{bmatrix} -0.103388224055090; 0.006965470186139; \\ 0.030780489043212; -0.045285273740815; \\ -0.114872139743147; -0.043720810837529; \\ -0.165513882004819 \end{bmatrix}, \\ f(\beta)_{SAPSO} = 0.001045522740319. \end{cases} \tag{39}$$

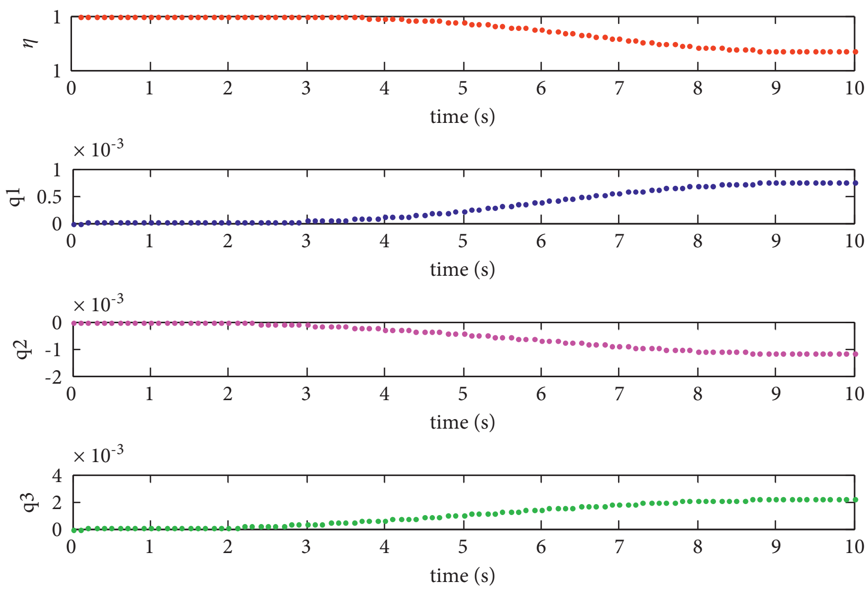

Figure 12: The attitude of the base obtained by the SAPSO algorithm.

Table 5: The comparison of the computing time.

Algorithms	QPSO	PSO	SAPSO	IQPSO proposed
Computing time (s)	16.931	21.750	16.463	16.174

The position of the base changes with the time as shown in Figure 11.

From Figure 11, it could be concluded that the position of the base was [0.000670871333624; −0.081591602625149; −0.187924872794062].

The attitude of the base changes along with the time as shown in Figure 12.

From Figure 12, it can be seen that the attitude of the base was [0.999996687432307; 0.000754961397235; −0.001174026294713; 0.002185046791830].

The comparison of the computing time is recorded in Table 5.

Comparing formulas (36)–(39), the optimal result calculated through the proposed IQPSO algorithm is the best. Through Figures 5–12, it is evident that the results calculated through the IQPSO algorithm were closest to the desired value. By analyzing the simulation results, it was concluded that the proposed IQPSO algorithm could search for the optimal value. The computational cost of the proposed approach was least, compared with other methods. Furthermore, it was suitable for solving the problem of trajectory planning to minimize base disturbance of the redundant space robot.

5. Conclusions

An approach based on the IQPSO algorithm was presented in the paper, which was applied for optimizing the problem of trajectory planning. In order to convert it to a mathematical problem, the model of the space robot was first built. Then, the kinematic equations of the redundant space robot were established. To smoothen the joint trajectory, the 5-order sine polynomial function was adopted. Thirdly, the fitness function was described through the only one parameter "β," and the trajectory planning problem of the free-floating redundant space robot was expressed as a nonlinear optimization problem. Finally, the proposed improved quantum-behaved particle swarm optimization (IQPSO) algorithm was used for optimizing the fitness function. The IQPSO algorithm searched the optimal value easily. Compared with the QPSO algorithm, PSO algorithm, and SAPSO algorithm, it not only has a good robust performance but also has a fast convergence speed. These advantages can be shown through the experiments of standard functions and trajectory planning. Through the simulation results, it was concluded that the proposed IQPSO algorithm could search for the global optimal value easily. Furthermore, it was suitable for solving the problem of optimal trajectory planning to minimize base disturbance of the redundant space robot.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research study was supported financially by the Henan Provincial Department of Science and Technology Research Project, grant nos. 222102210291 and 212102210325, the Key Scientific Research Projects of Colleges and Universities in Henan Province, grant no. 20B590001, and the Henan Institute of Technology Doctoral Research Fund Project, grant no. KQ1812.

References

[1] L. Liu, W. Yao, and Y. Guo, "Prescribed performance Tracking control of a free-flying Flexible-joint space robot with disturbances under input Saturation," *Journal of the Franklin Institute*, vol. 358, pp. 4571–4601, 2021.

[2] T. Rybus, M. Wojtunik, and F. L. Basmadji, "Optimal Collision-free path planning of a free-floating space robot using Spline-based Trajectories," *Acta Astronautica*, vol. 190, pp. 395–408, 2022.

[3] A. Hilli, M. Ibadi, S. Abdulshaheed, and S. H. Hadi, "Optimal path finding in Stochastic Quasi-Dynamic Environments using particle swarm optimization," *Expert Systems with Applications*, vol. 186, Article ID 115706, 2021.

[4] G. Rekleitis and E. Papadopoulos, "On-orbit Cooperating space robotic Servicers Handling a Passive Object," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 2, pp. 802–814, 2015.

[5] A. Flores-Abad, O. Ma, S. Pham, and S. Ulrich, "A Review of space Robotics Technologies for on-orbit Servicing," *Progress in Aerospace Sciences*, vol. 68, no. 68, pp. 1–26, 2014.

[6] C. Xiaoqian, Y. Jianping, Y. Wen, and Z. Yong, "Spacecraft on-orbit Servicing technology," *Beijing: China Astronautic Publishing House*, vol. 1, no. 1, pp. 8–10, 2009.

[7] Y. Li and C. P. Dang, "The development of Orbital Servicing technology in space," *Ordnance Industry Automation*, vol. 31, no. 5, pp. 79–82, 2012.

[8] W. Yan, Y. C. Liu, Q. J. Lan, T. Zhang, and H. Tu, "Trajectory planning and low-chattering fixed-time nonsingular terminal sliding mode control for a dual-arm free-floating space robot," *Robotica*, vol. 40, no. 3, pp. 625–645, 2021.

[9] D. Steven and A. T. Miguel, "Path planning for space manipulators to minimize Spacecraft attitude disturbances," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2522–2528, IEEE, Sacramento, CA, USA, April 1991.

[10] Z. Vafa and S. Dubowsky, "On the Dynamics of space manipulators using the Virtual manipulator, with applications to path planning," *Kluwer International Series in Engineering and Computer Science*, vol. 188, pp. 45–76, 1993.

[11] Z. B. Shi, Y. Li, X. Wang et al., "Path planning for robot based on quantum-behaved particle swarm optimization," *Personal and Ubiquitous Computing*, vol. 42, no. 2, pp. 33–37, 2010.

[12] X. P. Meng, X. Wang, Y. Li et al., "Choose the best path of the multi-agent soccer robot system," *Intelligent Automation and Soft Computing*, vol. 42, no. 2, pp. 25–27, 2010.

[13] T. T. Hu, J. X. Zhang, and Q. Zhang, "Trajectory planning to optimize base disturbance of 7-DOF free-floating space manipulator based on QPSO," in *Proceedings of the International Workshop on Multi-disciplinary Trends in Artificial Intelligence*, pp. 281–293, Springer, Fuzhou China, November 2015.

[14] X. X. Zeng, Y. Z. Guan, Z. Yan, and W. Li, "Path planning for minimizing base disturbance of free-floating space robot," in *Proceedings of the 2017 Symposium on Space Electromechanical and Space Optics*, pp. 67–73, IEEE, Naha, Okinawa, Japan, November 2017.

[15] Q. Zhang, L. Wang, and D. S. Zhou, "Trajectory planning of 7-DOF space manipulator for minimizing base disturbance," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 44, 2016.

[16] J. C. Hu and T. S. Wang, "Minimum base attitude disturbance planning for a space robot during Target Capture," *Journal of Mechanisms and Robotics*, vol. 10, no. 5, pp. 1–13, 2018.

[17] Q. Zhang, L. Ji, D. S. Zhou, and X. Wei, "Nonholonomic motion planning for minimizing base disturbances of space manipulators based on multi-swarm PSO," *Robotica*, vol. 35, no. 4, pp. 861–875, 2017.

[18] L. Ji, D. S. Zhou, and Q. Zhang, "Motion planning for minimizing base disturbance of space manipulator," *Journal of Donghua University*, vol. 32, no. 2, pp. 252–256, 2015.

[19] J. Chou, "Quaternion kinematic and Dynamic Differential equations," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 53–64, 1992.

[20] C. X. Yang, J. Zhang, and M. S. Tong, "A Hybrid quantum-behaved particle swarm optimization algorithm for solving Inverse Scattering problems," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 9, pp. 5861–5869, 2021.

[21] S. N. Omkar, R. Khandelwal, T. V. S. Ananth, G. N. Narayana, and S. Gopalakrishnan, "Quantum behaved particle swarm optimization (QPSO) for multi-Objective Design optimization of composite structures," *Expert Systems with Applications*, vol. 36, no. 8, Article ID 11322, 2009.

[22] W. Chen, B. Feng, and J. Sun, "Simulation study on the parameters optimization of Radial basis function Neural Network based on QPSO algorithm," *Computer Applications*, vol. 26, no. 8, pp. 1928–1931, 2006.

[23] X. Z. Xu, D. Shan, G. Y. Wang, and X. Jiang, "Multimodal medical image fusion using PCNN optimized by the QPSO algorithm," *Applied Soft Computing*, vol. 46, pp. 588–595, 2016.

[24] H. M. L. Haiming and G. Yen, "Rank-density-based Multi-objective Genetic algorithm and Benchmark test function study," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 4, pp. 325–343, 2003.