


ORIGINAL ARTICLE

Open Access



Obstacle Avoidance and Multitarget Tracking of a Super Redundant Modular Manipulator Based on Bezier Curve and Particle Swarm Optimization

Li Chen^{1,2*} , Ying Ma¹, Yu Zhang³ and Jinguo Liu²

Abstract

A super redundant serpentine manipulator has slender structure and multiple degrees of freedom. It can travel through narrow spaces and move in complex spaces. This manipulator is composed of many modules that can form different lengths of robot arms for different application sites. The increase in degrees of freedom causes the inverse kinematics of redundant manipulator to be typical and immensely increases the calculation load in the joint space. This paper presents an integrated optimization method to solve the path planning for obstacle avoidance and discrete trajectory tracking of a super redundant manipulator. In this integrated optimization, path planning is established on a Bezier curve, and particle swarm optimization is adopted to adjust the control points of the Bezier curve with the kinematic constraints of manipulator. A feasible obstacle avoidance path is obtained along with a discrete trajectory tracking by using a follow-the-leader strategy. The relative distance between each two discrete path points is limited to reduce the fitting error of the connecting rigid links to the smooth curve. Simulation results show that this integrated optimization method can rapidly search for the appropriate trajectory to guide the manipulator in obtaining the target while achieving obstacle avoidance and meeting joint constraints. The proposed algorithm is suitable for 3D space obstacle avoidance and multitarget path tracking.

Keywords: Super redundant manipulator, Multi target tracking, Follow-the-leader, Obstacle avoidance, Bezier curve, PSO

1 Introduction

Super redundant manipulators are widely applied in medical services, equipment maintenance, and post earthquake assistance [1, 2]. They can be divided into two structural types, namely, soft manipulators with unlimited degrees of freedom [3, 4] and rigid manipulators with limited degrees of freedom [5, 6]. Soft manipulators have high flexibility but their movement mechanism is complex, making them difficult to accurately control under a high load capacity [3]. Rigid manipulators are

mostly composed of joint modules in series. Their entire structure is easy to maintain and is robust to external influences because of their high bearing capacity and moderate flexible bending ability; thus, they have been widely studied in practice [4, 5]. The complexity of robot motion is that the task is given in the Cartesian coordinate, and the motion is realized in the joint space. With the increase in degrees of freedom, the inverse kinematics (IK) of redundant manipulator is typical, and the calculation load in the joint space immensely increases [7, 8].

The IK and path planning of a super redundant manipulator is usually solved through optimization, where obstacle avoidance and limit constraints at the joints are

*Correspondence: cl200432@tom.com

¹ School of Air Transportation, Shanghai University of Engineering Science, Shanghai 201620, China

Full list of author information is available at the end of the article

handled implicitly as boundary conditions for the numerical algorithm, enabling complex task planning to be implemented with expensive computational cost [9, 10]. This paper aims to reduce the complexity of optimization calculation. Niu et al. [11] presented a projection strategy and region clipping method to reduce the computational complexity. Godage et al. [12] developed a mode shape function-based kinematic model of continuum manipulators and efficiently yielded accurate solutions. Ananthanarayana et al. [13] proposed analytical equations to determine the first two and last three joint angles and a numerical technique to solve the remaining joint angles. Aristidou et al. [14] presented a heuristic iterative method called forward and backward reaching IK to solve the IK problem under different scenarios. David et al. [15] proposed a tip following method using a sequential quadratic programming optimization approach to navigate the robot. Maria et al. [16] presented a follow-the-leader deployment strategy for continuum robots with extensible sections along 3D paths. Tappe et al. [17] proposed a tip following method adaptable to binary actuation with optimized switching sequences. Xiong et al. [18] developed a tip following method with constraints on the bending angles. The joint position is located on the axis of the snake arm to reduce the configuration deviation of the entire robot. Palmer et al. [19] presented an approach to control the active action of coiling a continuum arm using tip following navigation. Xu et al. [20] proposed a modified fitting method based on a mode function to solve the IK of hyper-redundant manipulator. Xie et al. [21] presented a novel follow-the-leader method for rigid-backbone serpentine manipulator with high location precision and less computation time. Collins et al. [22] proposed an integrated, scalable optimization framework that uses particle swarm optimization (PSO) to produce approximate solutions to the hyper-DOF path planning and IK problems. Bulut et al. [23] proved that taking the whole arm as discrete links is an efficient way for real-time path planning. The abovementioned studies simplify the problem from the IK perspective.

In this paper, an integrated method is proposed to control a super redundant manipulator when entering an aircraft fuel tank with internal obstacles. The integrated optimization problem of path planning and trajectory tracking is established on the basis of Bezier curve and PSO [24–26]. The Bezier curve has an extremely strong graphic description ability by changing the position of several control points [27, 28]. PSO has good convergence with small calculation time for most optimization problems [29, 30]. The optimal results are one feasible path to avoid obstacles with a series of discrete path points and a discrete path tracking of super redundant manipulator guided by using a follow-the-leader strategy.

The relative angle between two discrete path sections is limited to reduce the fitting errors of the connecting rigid links and smoothen the tracking during grasping or monitoring.

The remainder of this paper is organized as follows: Section 2 presents the manipulator model. Section 3 discusses the methodology for following the discrete Bezier path points of hyper redundant manipulators. Section 4 describes the path planning optimization problem. Section 5 extends the proposed method to different constraints and multitarget tracking. The simulation results under different situations are presented. Section 6 provides the conclusion.

2 Manipulator Parameters

The super redundant manipulator is mainly composed of a moving base, link model, joint, and end effector, and its overall structure is shown in Figure 1. The target working space of the robotic manipulator is a $120 \times 80 \times 30$ cm³ rectangular fuel tank. The bottom surface has a circular entrance with a diameter of 10 cm. The base of the manipulator is mounted on a vertical guide rail that can realize the feed movement of the robot manipulator in the vertical direction. Each joint has two degrees of freedom, namely, bending angle θ_i between the two adjacent links and rotational angle φ_i around each link axis. The base provides the feed displacement for the robot manipulator to move up and down. The robot manipulator has $2n+1$ degrees of freedom (n is the number of modular joints), and the other main parameters are shown in Table 1.

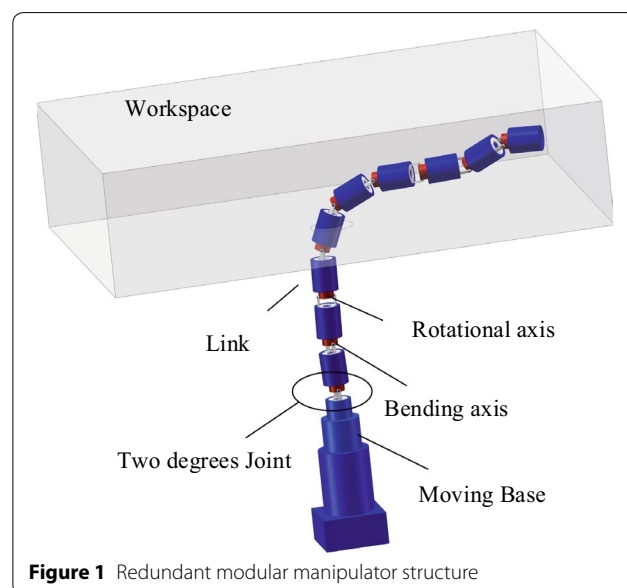
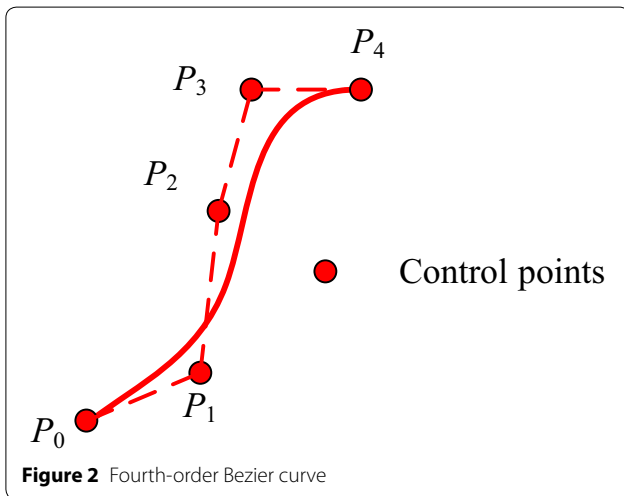


Figure 1 Redundant modular manipulator structure

Table 1 Main parameters of the modular manipulator

Parameter	Value
Diameter of manipulator d (cm)	4
Numbers of manipulator	12–16
Length of link l (cm)	5–9
Maximum bending angle θ_{\max} (°)	30–60
Feed distance (cm)	Length of manipulator



The number of robot manipulator, link length, maximum bending angle of every joint, and the distance and speed of the base movement can be designed. The maximum distance of the base movement can send the entire robotic manipulator into the fuel tank and is defined as the manipulator’s length.

3 Discrete Trajectory Tracking and Control

3.1 Bezier Curve Fitting

A Bezier curve has an extremely strong graphic description ability that can smoothly connect position points and has convexity. Changing the positions of control points can ensure that the formed path does not collide with obstacles, as shown in Figure 2.

The expression of m -order Bessel curve can be written as

$$B(t) = \sum_{i=0}^m P(i)C_{i,m}(t), t \in [0, 1], \tag{1}$$

where $B(t)$ is the position of discrete points at time t (the point on the Bezier curve moves from the start point to the end point while t changes from 0 to 1), $P(i)$ is the position of control points, and $C_{i,m}$ is the coefficient of fitting curve

$$C_{i,m}(t) = \frac{m!}{(m-i)!i!}t^i(1-t)^{m-i}, i=0, 1, \dots, m, \tag{2}$$

where m is the order of Bessel curve. When $m=1$, the Bezier curve is a straight line; when $m=2$, the Bezier curve is a parabola; and when $m \geq 3$, the curve is a high-order Bezier curve. The greater the order of m is, the more the flexible the Bezier curve is, but the more the computing resources will be. The fourth-order Bezier curve is selected to avoid obstacles.

3.2 Follow-the-Leader Strategy

All joints should be far from the obstacle avoidance trajectory to move the robotic manipulator along the optimized path because the numerical optimal path generated by this method is a series of spatially discrete points, as shown in Eq. (1). However, we cannot find the analytical solution derived by the geometric method because the trajectory is discretized into Bezier path points. A given joint position p_{i+1} consistently coincides with the discrete path point in front of it when it is an end effector. The positions of subsequent joints can be deduced from the link length constraint and are supposed to be located on the line connected by two adjacent path points, as shown in Figure 3.

The previous joint position p_i is located on one circle with p_{i+1} as the center and link length as the radius. The intersection of this circle and line $m_{i-1}m_i$ is point p_i , where m_{i-1} and m_i are the two adjacent path points. The distance of p_i from p_{i+1} satisfies the length constraints $le = |\overrightarrow{p_i p_{i+1}}|$. Let $\vec{w} = \overrightarrow{m_{i-1} m_i}$, then $\lambda \vec{w} = \overrightarrow{m_{i-1} p_i}$, ($0 < \lambda < 1$). The vector theorem states that

$$\lambda \vec{w} = \overrightarrow{m_{i-1} p_{i+1}} - \overrightarrow{p_i p_{i+1}}, \tag{3}$$

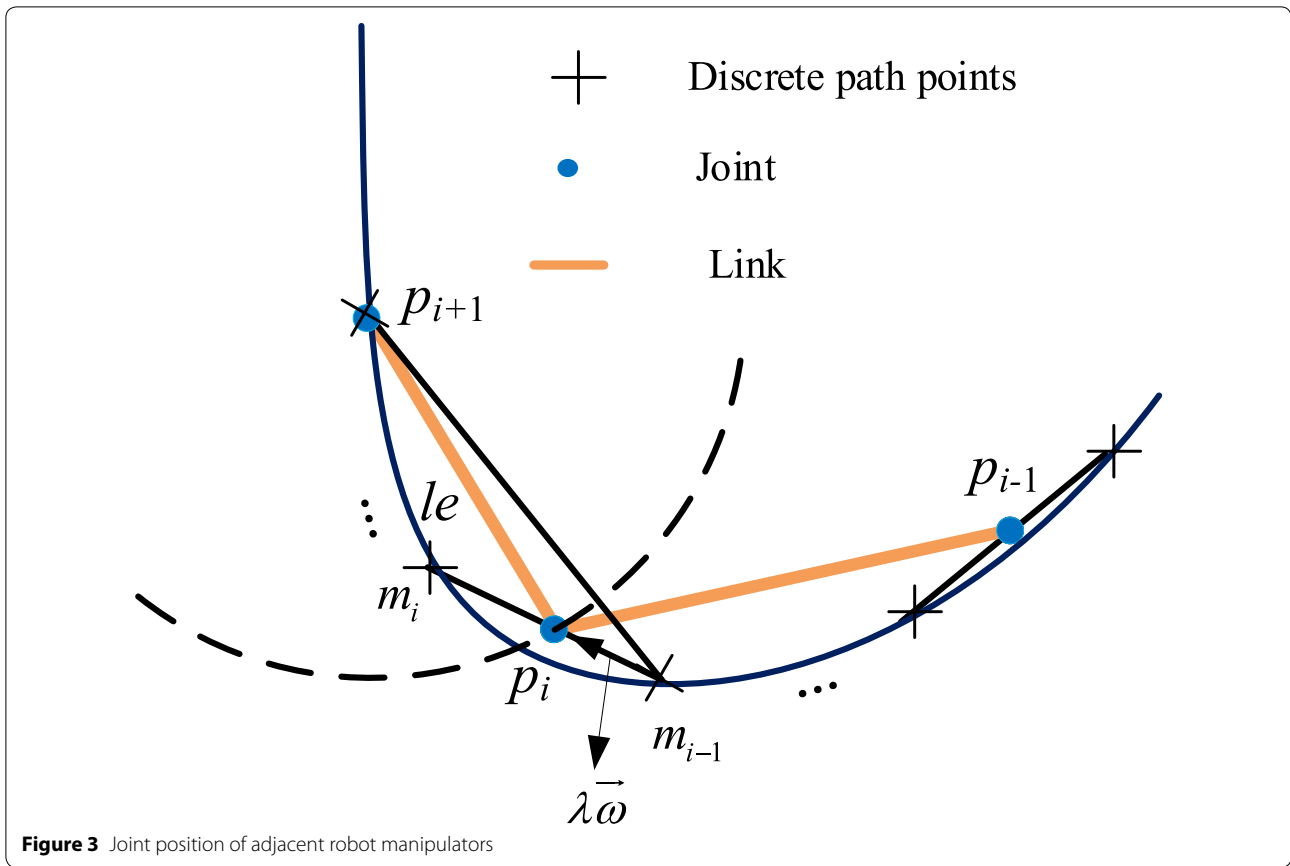


Figure 3 Joint position of adjacent robot manipulators

then

$$\lambda = \frac{\overrightarrow{m_{i-1}p_{i+1}} - \overrightarrow{p_i p_{i+1}}}{\overrightarrow{w}} \tag{4}$$

Thus, the position of p_i is

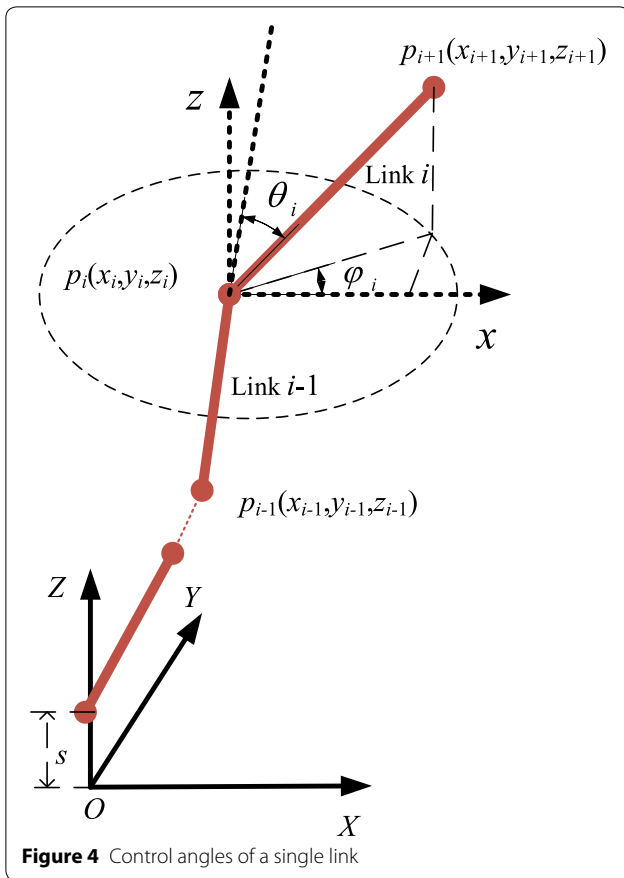
$$\overrightarrow{p_i} = \overrightarrow{m_{i-1}} + \lambda \overrightarrow{w} \tag{5}$$

The following joint position p_{i-1} is determined from p_i in the same manner. The motion of manipulator is guided by the tip following strategy, and joint position update is derived using the geometric method. However, unavoidable errors occur when fitting the continuous curve using the rigid link. The distance between two discrete path points should be as short as possible to reduce the fitting error. As shown in Figure 3, the smaller the distance between m_{i-1} and m_i is, the closer

the joint distance from the fitting curve. However, the large number of path points decrease the computation efficiency. Therefore, the interval of discrete points should be accordingly selected.

3.3 Control Variables

Assuming that the obstacle avoidance trajectory has been obtained, the trajectory is usually expressed in the inertial coordinate system $OXYZ$, and the space path is discretized. We can obtain the position of every joint by using the follow-the-leader strategy in Section 3.2. Thus, the relative position of two links is determined, as shown in Figure 4. The control joint angles are deduced by the relative position changes in the two adjacent joints at the inertial space. This condition is because the control angles of joint i change the position of joint



$i+1$, thereby avoiding the IK calculation of the entire manipulator in the joint space. The control parameters of joint angles θ_i and φ_i and base displacement can be calculated as follows:

$$\theta_i = \arccos \left(\frac{\overrightarrow{p_{i-1}p_i} \cdot \overrightarrow{p_i p_{i+1}}}{\| \overrightarrow{p_{i-1}p_i} \| \| \overrightarrow{p_i p_{i+1}} \|} \right), \tag{6}$$

$$\varphi_i = \arctan 2(y_{i+1} - y_i, x_{i+1} - x_i), \tag{7}$$

$$\begin{cases} \dot{s} = \frac{1}{\tau} (Z^t - Z^{t-\tau}), \\ s = \int_0^t \dot{s} dt, \end{cases} \tag{8}$$

where $p_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$, $p_i(x_i, y_i, z_i)$, and $p_{i-1}(x_{i-1}, y_{i-1}, z_{i-1})$ are the spatial coordinates of joints $i+1$, i , and $i-1$ on the spatial trajectory, respectively, Z^t is the base position at time t , $Z^{t-\tau}$ is the base position at previous moment $t - \tau$, τ is the sample period, \dot{s} represents a certain speed of the base movement, and s is the base displacement. Base motion is restricted along the Z axis.

The rotation angle has no limit because it is easily designed to move 180° left and right. The bending angle of adjacent links is the important design parameter for hyper redundant manipulator and depends on its flexibility. The feed distance and speed of base movement determine the manipulator position. The possible workspace is extremely small, and the height is extremely short to allow the manipulator for immensely extending inside. These physical constraints of the manipulator must be considered in path planning.

4 Obstacle Avoidance Path Planning

4.1 Obstacle Checking

In the real environment, obstacles are usually irregular in shape and difficult to describe with mathematical formulas. Therefore, a regular envelope is used to approximate the outline of obstacles. Although this method expands the volume of obstacles, it simplifies the obstacle description to improve the planning efficiency and avoid the collision of the manipulator with obstacles during the movement. The envelope radius of the obstacles includes the radius of the robot link. Let the coordinates of p_i and p_{i+1} be (x_i, y_i, z_i) and $(x_{i+1}, y_{i+1}, z_{i+1})$, respectively. The linear parameter equation of robot link i can be expressed as

$$\begin{cases} x = (1 - k)x_i + kx_{i+1}, \\ y = (1 - k)y_i + ky_{i+1}, \\ z = (1 - k)z_i + kz_{i+1}, \end{cases} \tag{9}$$

where $0 < k < 1$, (x, y, z) is the arbitrary point on link for the obstacle avoidance, and the distance of arbitrary point on link from the obstacle center should satisfy

$$(x - a)^2 + (y - b)^2 + (z - c)^2 > (r + \frac{d}{2})^2, \tag{10}$$

where (a, b, c) are the coordinates of the spherical center of the obstacle, r is the radius of the spherical obstacle,

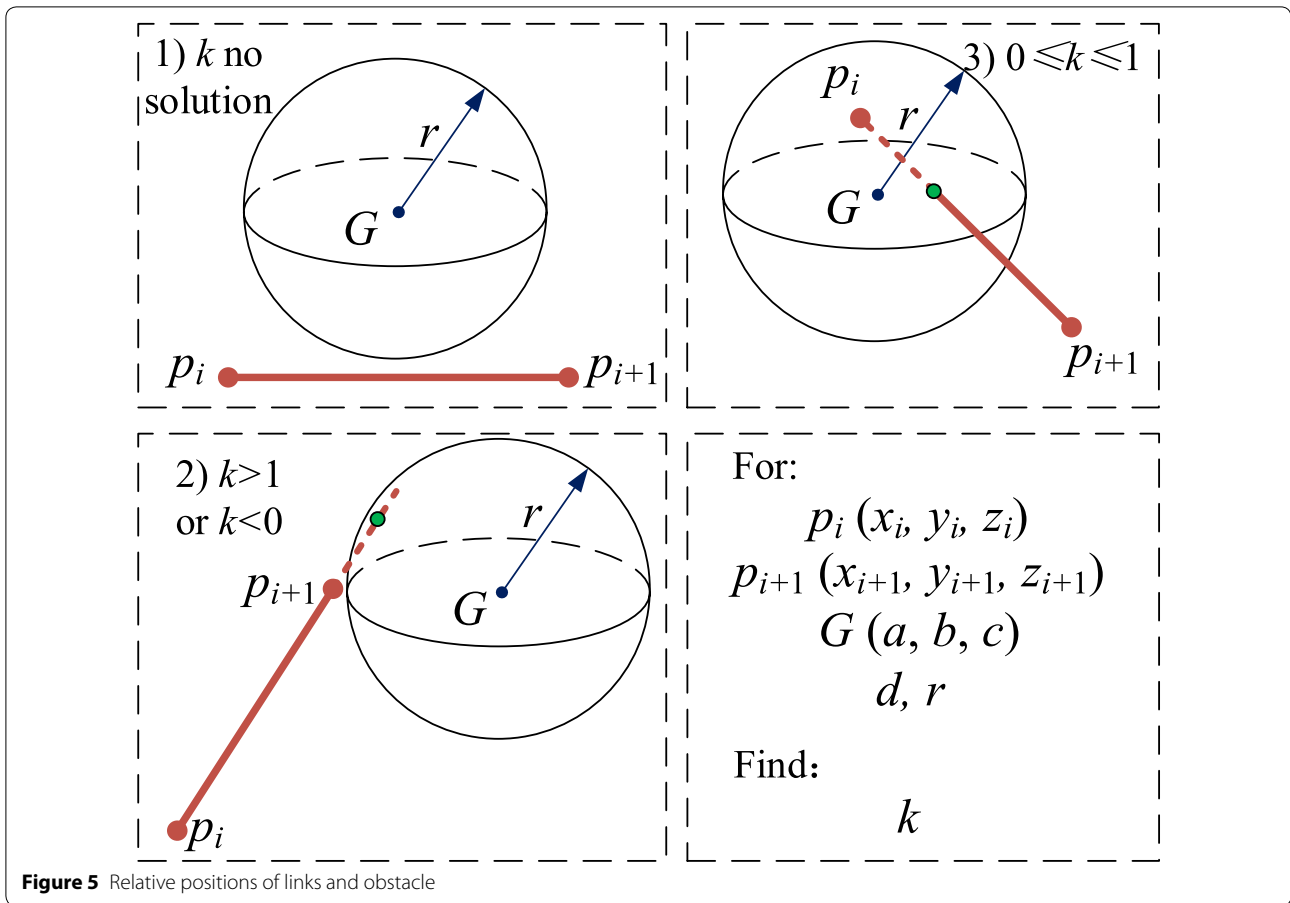


Figure 5 Relative positions of links and obstacle

and d is the diameter of robot link. Solving Eqs. (9) and (10) simultaneously yields the quadratic equation for parameter k :

$$\begin{aligned}
 & [(1 - k)x_i + kx_{i+1} - a]^2 + [(1 - k)y_i + ky_{i+1} - b]^2 + \\
 & [(1 - k)z_i + kz_{i+1} - c]^2 = (r + \frac{d}{2})^2.
 \end{aligned}
 \tag{11}$$

Figure 5 shows the three typical relative positions of $p_i p_{i+1}$ with obstacle G to represent the collision situations in path planning. The manipulator does not collide with the obstacle when k has no solution (first scenario in Figure 5). The manipulator does not collide with the obstacle when $k > 1$ or $k < 0$ (second scenario in Figure 5). The

manipulator collides with an obstacle when $0 \leq k \leq 1$ (third scenario in Figure 5).

4.2 Integrated Optimization Problem

This section provides the integrated optimization problem of obstacle avoidance path planning with kinematic constraints in path tracking control. The optimization aims to minimize the maximum joint bending angles of the manipulator at each moment. The optimization problem should satisfy the following constraints: the joint angle does not exceed the maximum bending angle, the robotic manipulator does not collide with any obstacles inside, and the manipulator is working in an effective

workspace. The mathematical expression of the integrated optimization problem is written as follows:

$$J = \min(\max\theta_j(P)),$$

$$\begin{cases} \theta_j < \theta_{\max}, j = 1, \dots, num, \\ k > 1 \text{ or } k < 0, \\ x_i, y_i, z_i \in W \in i = 1, \dots, M. \end{cases} \quad (12)$$

Here, the optimization variable is the control point positions of Bezier curve P_i , $i = 1, 2, 3$, where θ_{\max} indicates the maximum bending angle of the robot manipulator in path tracking, num indicates the number of joints, M represents the number of discrete points on the path, W indicates the working space of the robot manipulator, and (x_i, y_i, z_i) are the positions of arbitrary point on the manipulator. $k > 1$ or $k < 0$ indicates that obstacle avoidance conditions are satisfied.

PSO, which is a bionic intelligent algorithm [22], is used to solve the above optimization problem. Each particle is a potential solution evaluated by fitness value J in Eq. (12). For a group of N particles, each particle has M dimensions. For the i th particle, its position and speed are $X_i = [x_{i,1}, x_{i,2}, \dots, x_{i,M}]$ and $V_i = [v_{i,1}, v_{i,2}, \dots, v_{i,M}]$, respectively. In the optimization, the particle obtains the individual optimal value P_{besti} and group's optimal value G_{best} in accordance with the fitness value reflected in each optimization step. It judges whether to adjust in accordance with the fitness value and then gradually moves toward the optimal particle. The position and velocity of the particles are updated as follows:

$$\begin{cases} v_{i,j}^{h+1} = \omega \cdot v_{i,j}^h + c_1 r_1^h (p_{i,j}^h - x_{i,j}^h) + c_2 r_2^h (p_{g,j}^h - x_{i,j}^h), \\ x_{i,j}^{h+1} = x_{i,j}^h + v_{i,j}^{h+1}, \end{cases} \quad (13)$$

where subscripts i and j represent the particle number and particle dimension, respectively, $p_{i,j}^h$ represents the optimal solution of the j th dimension of particle i in the h th optimization step, $p_{g,j}^h$ represents the optimal solution of the j th dimension of all particles in the h th optimization step, c_1 and c_2 are the learning factors, which are taken as 2, r_1^h and r_2^h are random numbers between 0 and 1, and ω is the weight coefficient. The larger the

value is, the stronger the global search ability will be. The smaller the value is, the stronger the local search ability will be, where $\omega = 0.8$ in this study.

4.3 Obstacle Avoidance Path Generation

The steps of obstacle avoidance path generation are given as follows.

Step 1: Initialize the particle positions and velocities of the Bezier curve. Update the tracking position of the manipulator and calculate the control angles and base displacement in the initial configuration.

Step 2: Determine whether the initial curve path meets the optimization constraints. Go to Step 7 when they are met, otherwise go to Step 3.

Step 3: Update the positions and velocities of the particles by using Eq. (13) and obtain the path points by using Eq. (1).

Step 4: Compare the fitness values of the particles before and after the update by using Eq. (12). The individual optimal solution of the particles is refreshed when the fitness values of the particles after the update are better than the fitness values of the particles before the update. Otherwise, no refresh is required.

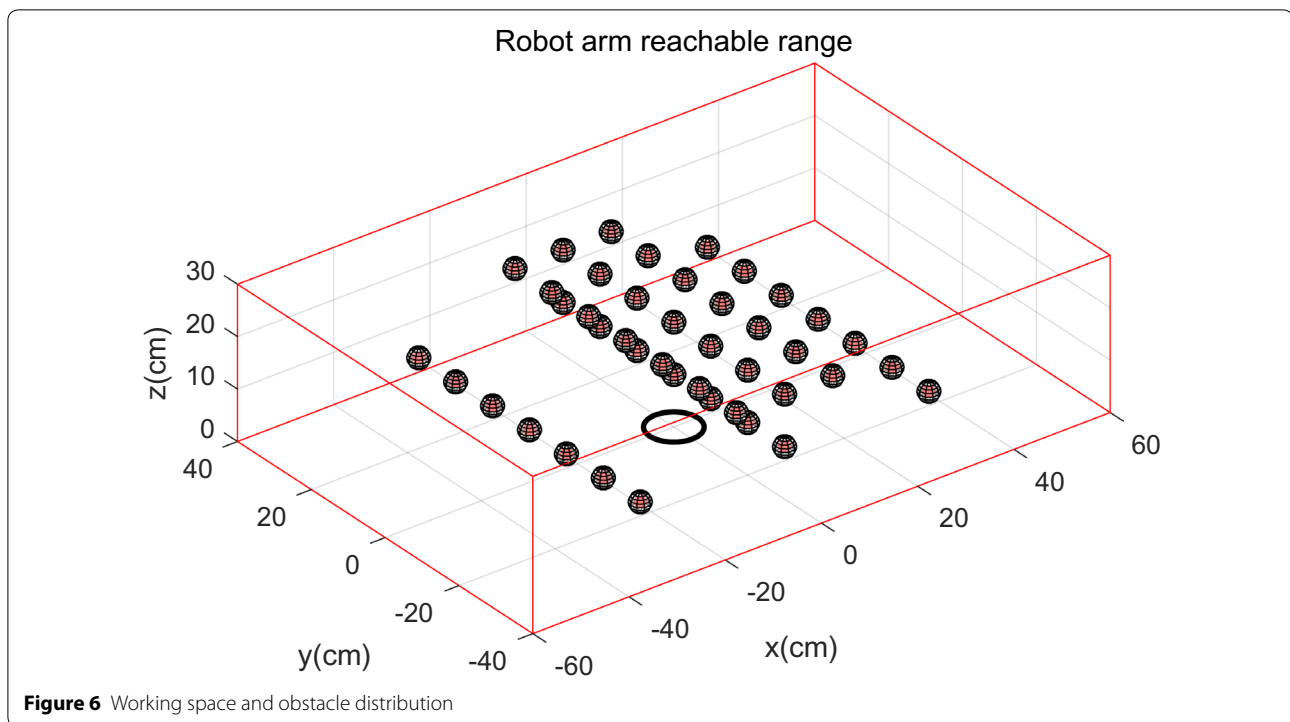
Step 5: Compare the updated particle fitness value with the group's optimal particle fitness value. Refresh the particle group's optimal solution when the former is better than the latter. Otherwise, no refresh is required.

Step 6: Update the tracking position of the manipulator and calculate the control angles and base displacement for every optimal result. Determine whether the Bezier curve corresponding to the group's optimal solution meets the optimization constraints in Eq. (12). Specifically, $\max \theta_{i,\max} < \theta_{\max}$, the manipulator does not collide with the obstacle, and the positions of every link does not exceed the workspace. Go to Step 7 when these constraints are satisfied. Otherwise, return to Step 3.

Step 7: Store the optimal solution of the group (that is, the positions of the control points satisfying obstacle avoidance and joint constraints) and generate a series of discrete Bezier path points.

5 Simulation Results and Discussion

In this simulation, the working space of fuel tank W is $120 \times 80 \times 30 \text{ cm}^3$, and the robotic manipulator enters the fuel tank with a radius of 5 cm. Obstacle radius $r = 1 \text{ cm}$,



the minimum safety distance between the robotic manipulator, and the obstacle is 1 cm. Six groups of obstacles are distributed in the form of an array in the fuel tank, and the positions of obstacle center of every column are shown in Figure 6.

Obstacle 1 = [30, -30, 10; 30, -20, 10; 30, -10, 10; 30, 0, 10; 30, 10, 10; 30, 20, 10; 30, 30, 10];

Obstacle 2 = [-30, -30, 10; -30, -20, 10; -30, -10, 10; -30, 0, 10; -30, 10, 10; -30, 20, 10; -30, 30, 10];

Obstacle 3 = [0, -30, 10; 0, -20, 10; 0, -10, 10; 0, 0, 10; 0, 10, 10; 0, 20, 10; 0, 30, 10];

Obstacle 4 = [50, -30, 20; 50, -20, 20; 50, -10, 20; 50, 0, 20; 50, 10, 20; 50, 20, 20; 50, 30, 20];

Obstacle 5 = [-50, -30, 20; -50, -20, 20; -50, -10, 20; -50, 0, 20; -50, 10, 20; -50, 20, 20; -50, 30, 20];

Obstacle 6 = [0, -30, 20; 0, -20, 20; 0, -10, 20; 0, 0, 20; 0, 10, 20; 0, 20, 20; 0, 30, 20].

As shown in Figure 6, the black balls are the obstacle avoidance envelopes, and the red balls inside are the real obstacles. The typical manipulation of this manipulator is going into the tank and is reversed at 180° to pick up the

foreign matter at the bottom of the tank or with a camera on the end effector to monitor the cracks on the tank wall.

5.1 Arbitrary Target Arrival in Space

Given robot manipulator length $l_e=9$ cm, number of sections $num=12$, maximum relative angle $\theta_{max}=30^\circ$, robot manipulator diameter $d=4$ cm, and the target position is (50, 30, 0). The simulation results are as follows: the trajectories and bending angles before and after optimization are shown in Figure 7, and a possible trajectory can be generated before optimization. However, the bending angles exceed the limit of 30° in some cases. The trajectory is divided into 60 portions, and the manipulator moves each portion at every sample time. The final optimal trajectory is shown in Figure 8 under different perspectives. As shown in Figure 8, the red line is the manipulator link, the green line is the obtained track, and they basically coincide. The discrete rigid links can fit the continuous track with small errors. These errors depend on many factors, such as link length, bending angle, and

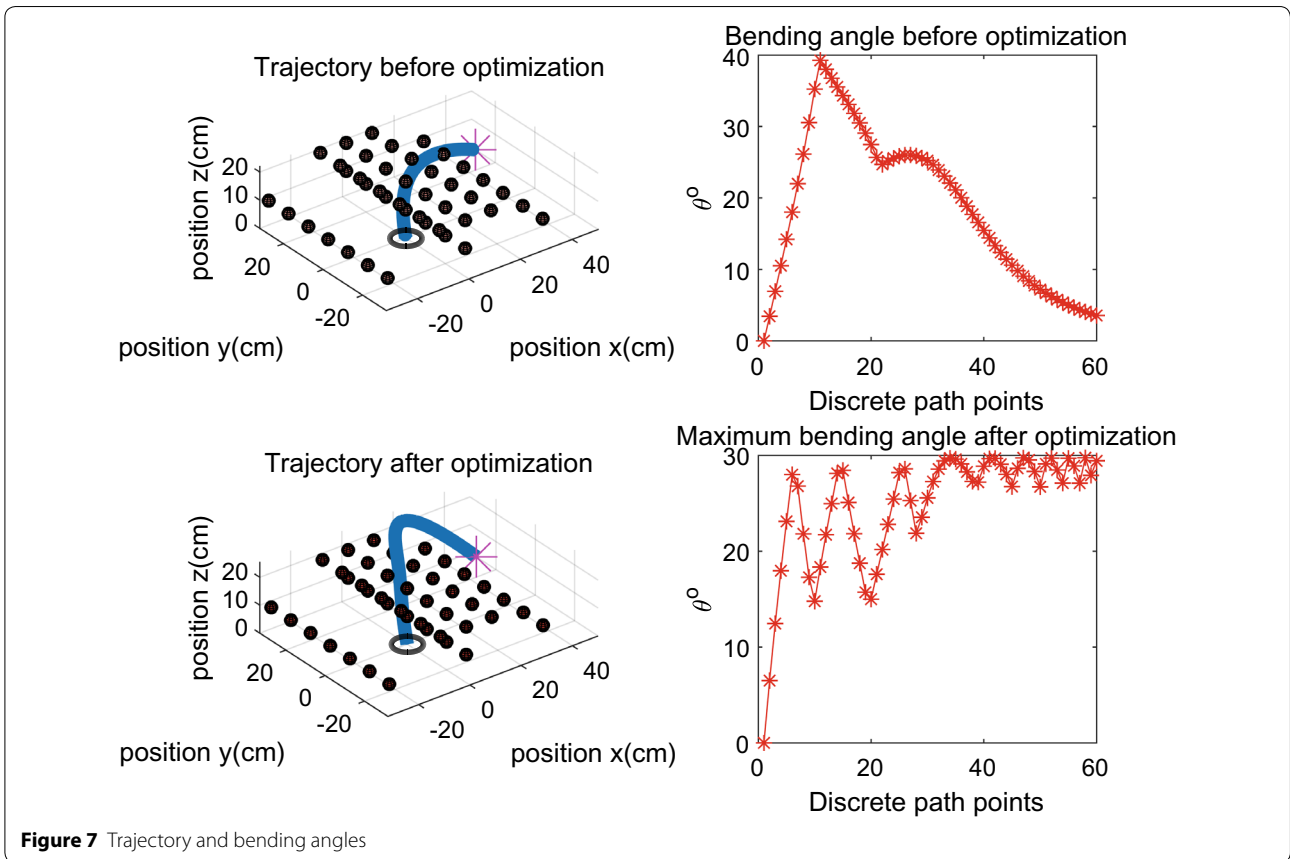


Figure 7 Trajectory and bending angles

length of each portion. The resulting control angles and base movement are shown in Figures 9 and 10.

All physical constraints are satisfied. The bending angle are within the limits of $\theta_{\max} = 30^\circ$. The base displacement change depends on the distance between two adjacent path points in every sample time.

5.2 Multitarget Path Tracking

Multipoint tracking is conducted to illustrate the generality of the proposed algorithm. The manipulator needs to move to the next position for performing subsequent tasks after completing the task at the previous position. Given that the manipulator passes through the two target points, one curve is determined by two points, namely, start and end points. Thus, multiple sections of obstacle avoidance paths for multipoint tasks should be smoothly connected to guide the multipoint tracking.

As shown in Figure 11, two fourth-order Bezier curves are found for two-target tracking. End point $P_{1,4}$ of the

former curve B_1 is the start point of the latter curve B_2 . The two curves are smoothly connected at $P_{1,4}$, that is, the two curves are continuous and differentiable at $P_{1,4}$. The derivation of Eq. (1) can be expressed as

$$\dot{B}(t) = \sum_{i=0}^m P(i) \dot{C}_{i,m}(t), \quad t \in [0, 1]. \tag{14}$$

On the basis of continuous derivable condition, we have

$$\dot{B}_1(1) = \dot{B}_2(0). \tag{15}$$

Expanding and simplifying Eq. (15) yields

$$4(P_{1,4} - P_{1,3}) = 4(P_{2,1} - P_{2,0}), \tag{16}$$

then

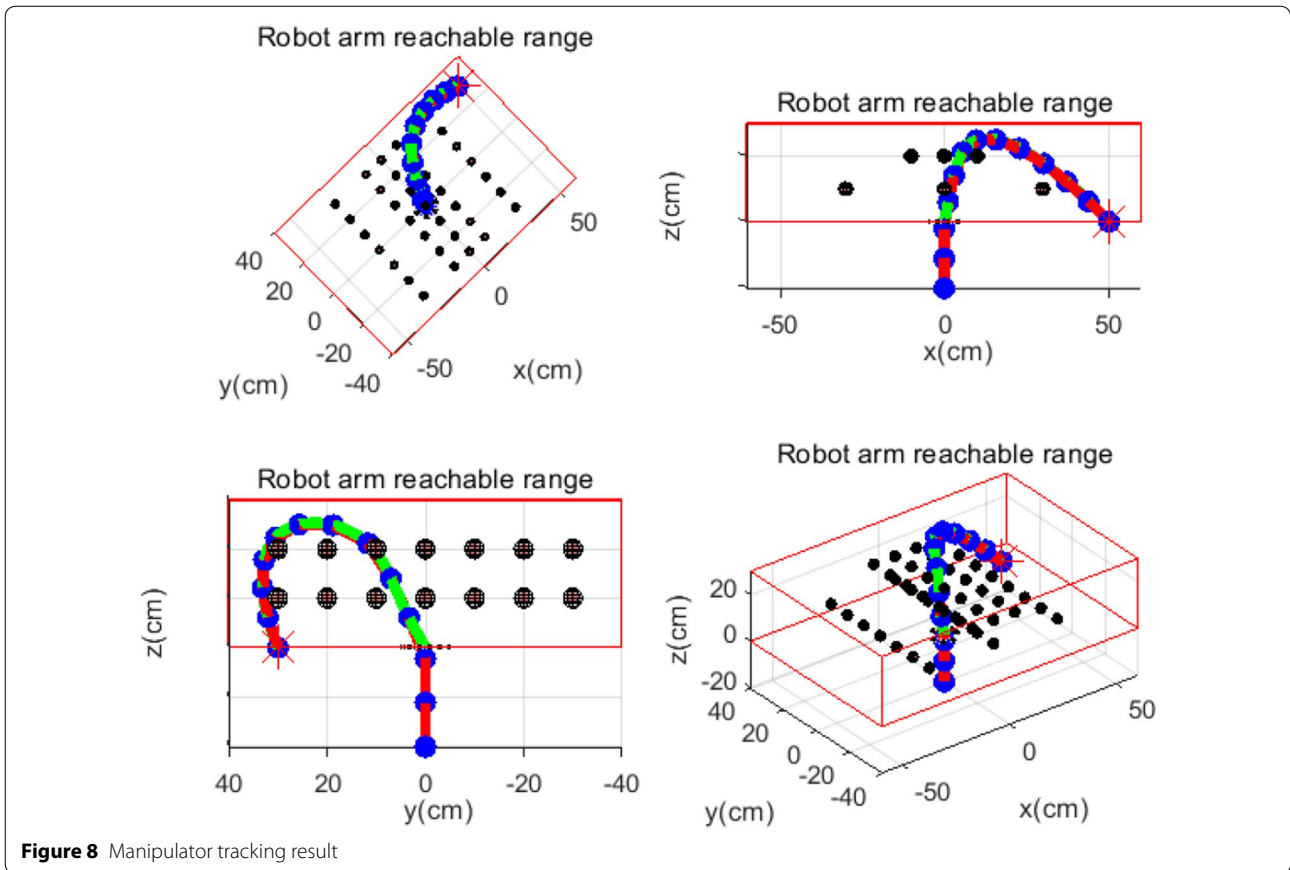


Figure 8 Manipulator tracking result

$$P_{1,4} + P_{2,0} = P_{1,3} + P_{2,1}, \tag{17}$$

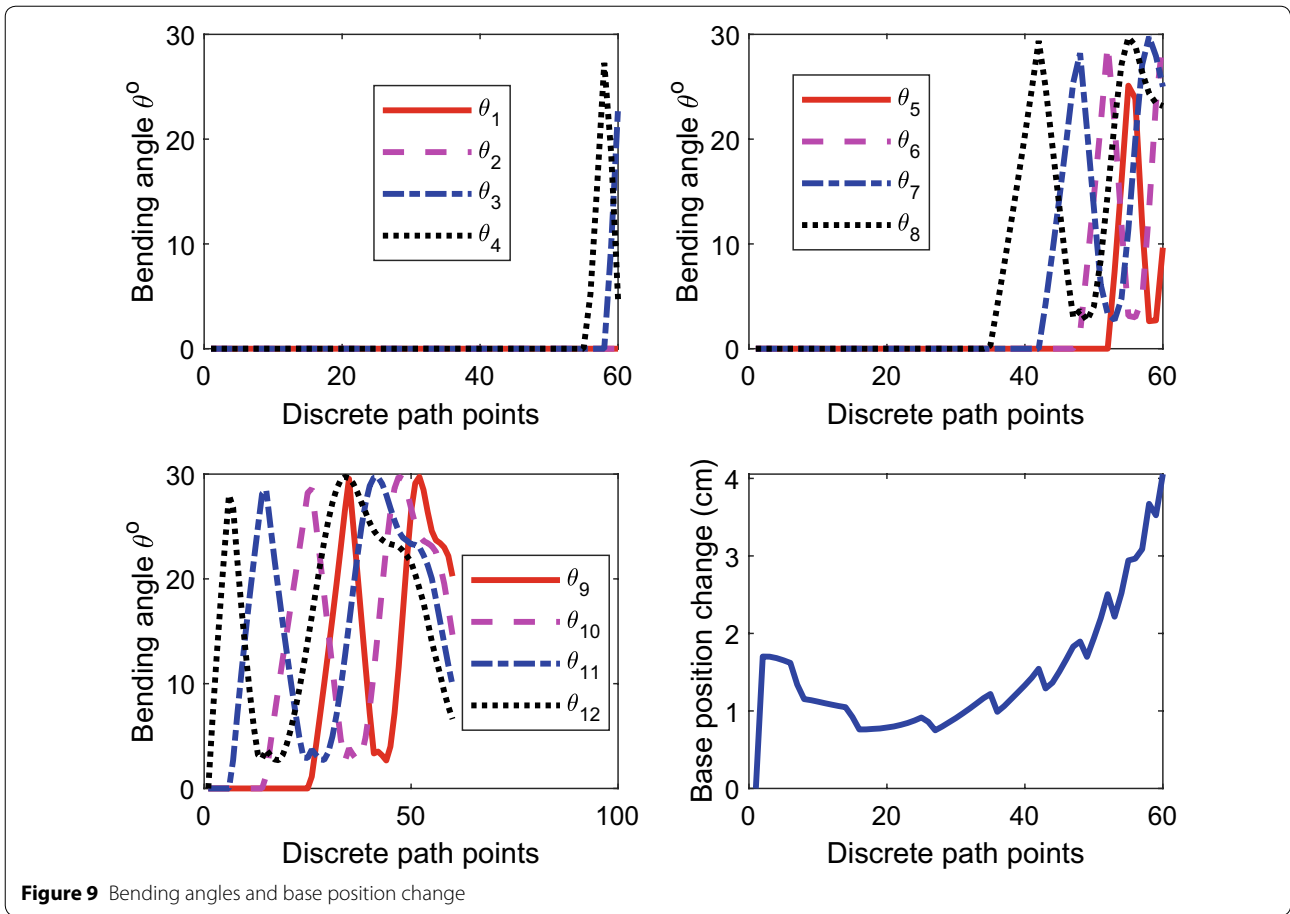
Considering that $P_{1,4} = P_{2,0}$, Eq. (17) can be written as follows:

$$P_{1,4} = \frac{1}{2}(P_{1,3} + P_{2,1}). \tag{18}$$

As shown in Eq. (18), control point $P_{1,3}$ of curve B1, connection point $P_{1,4}$ of the two curves, and second

control point $P_{2,1}$ of curve B_2 should be on the same straight line. The connection point of the curve should be the midpoint of the two control points to ensure the smooth connection of two fourth-order Bezier curves.

Considering the smooth connection condition of the fourth-order Bezier curve, only five control points need to be optimized for two-target path tracking. On the basis of the above theory, the simulation is given, the starting point is set to (0, 0, 0), the first target point is



(20, 15, 15), and the second target point is (50, 40, 0). The simulations are shown in Figures 12, 13, 14, 15. The two pink asterisks represent the two targets. The end effector is passing through every path point to ensure accurate target tracking. The optimal trajectory is suppressed with

small bending angles, the two sections of the trajectory are smoothly connected by target points, and all physical constraints are satisfied. For a long track, the modules of manipulators should be increased to lengthen them for obtaining the target points.

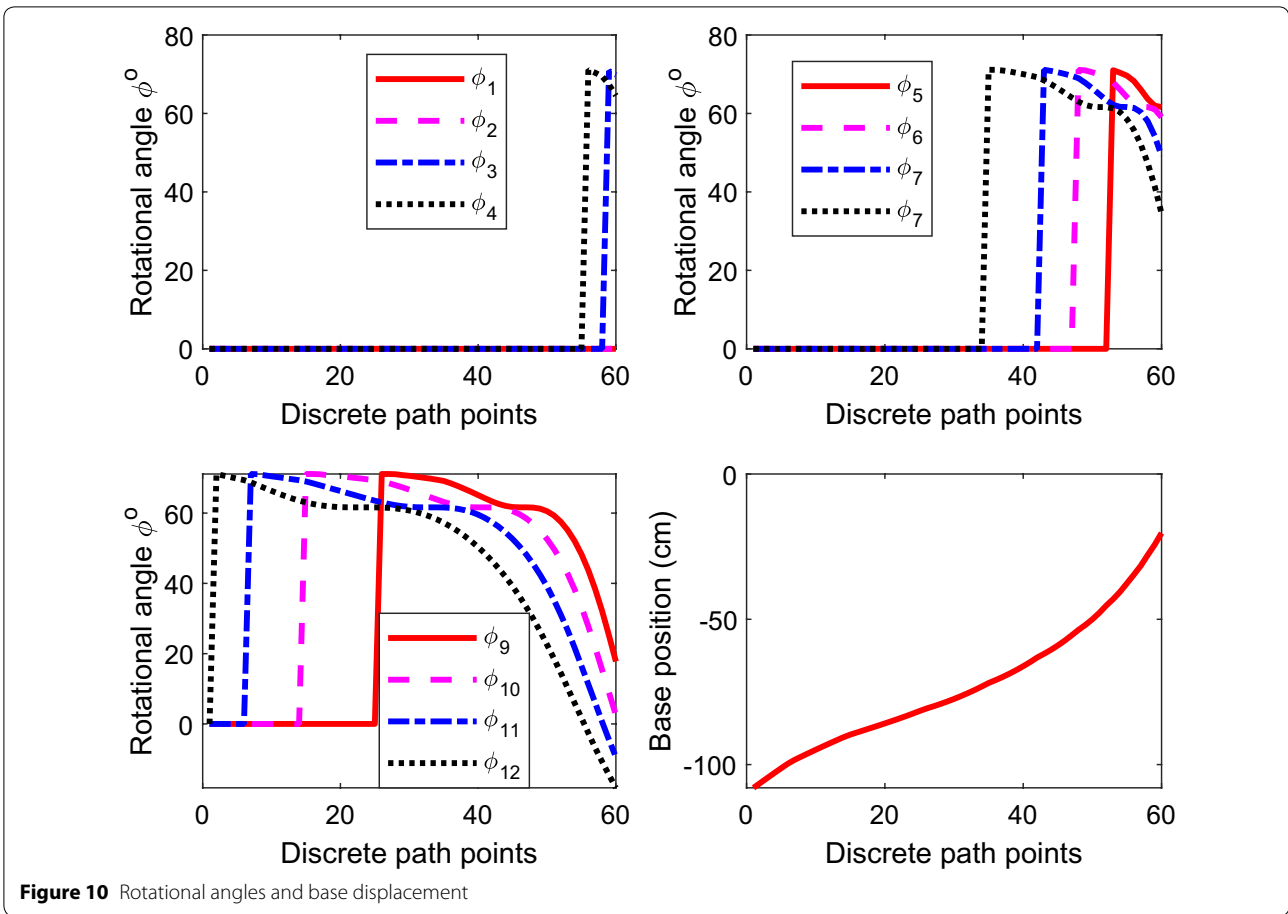


Figure 10 Rotational angles and base displacement

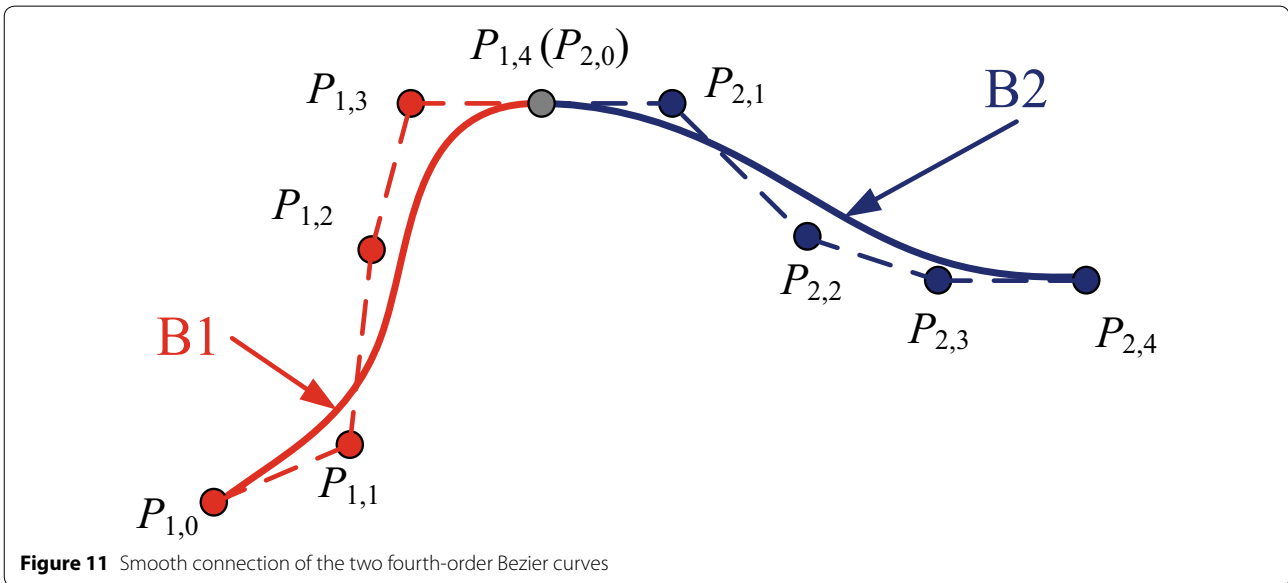


Figure 11 Smooth connection of the two fourth-order Bezier curves

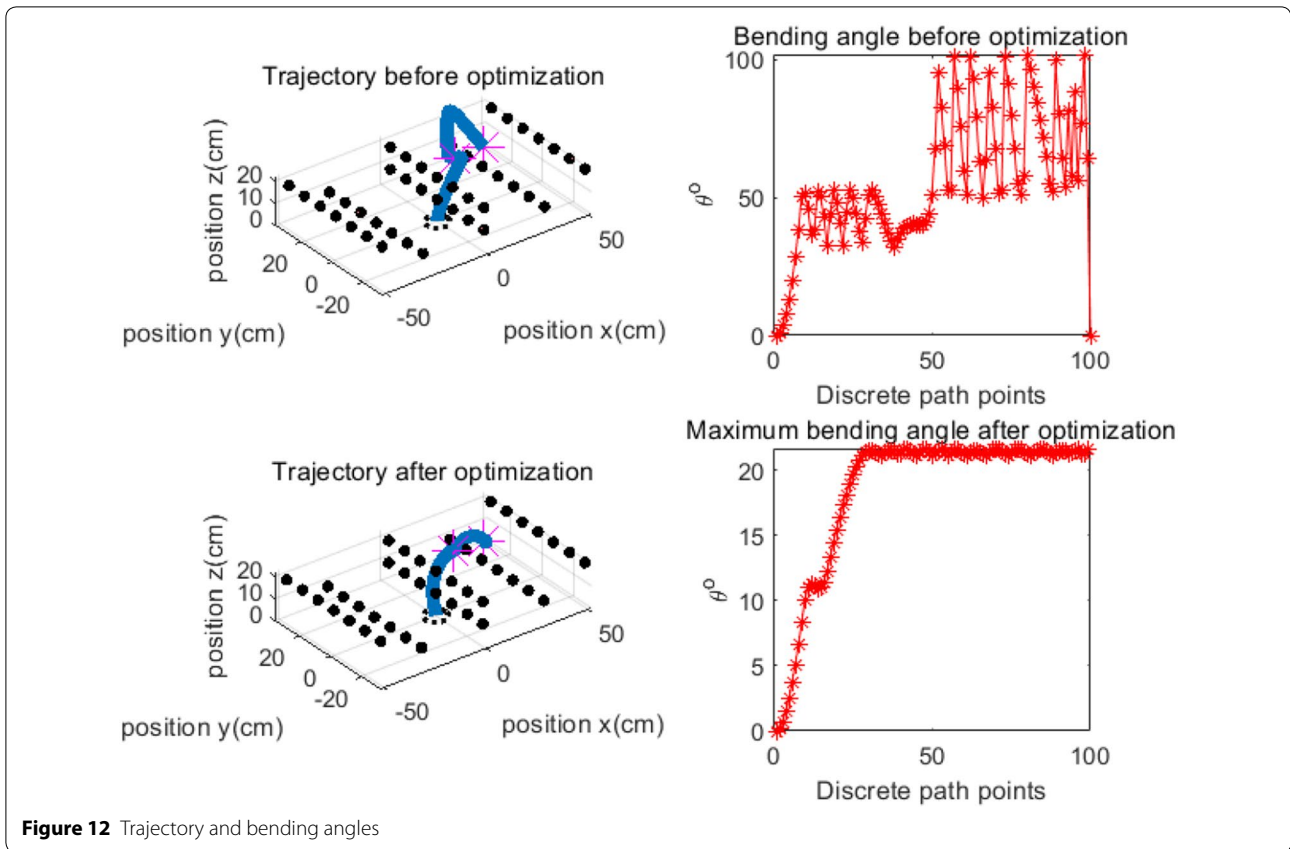


Figure 12 Trajectory and bending angles

5.3 Relaxing Design Conditions

In this section, we will relax the design conditions to obtain large curvature motion. For the super redundant manipulator, the bending angles are the critical design parameters because they determine the curvature of the designed trajectory and make the robot motion to be flexible or obtain steep motion in strictly limited environment. In this simulation, the starting point is given as (0, 0, 0), the position of the first target point is (0, 0, 15), and the second target point is (-40, 30, 0). The manipulator needs to cross between the two obstacles with a large

curvature because they are located at (0, 0, 10) and (0, 0, 20). This manipulator cannot reach this point with the former bending angle limits. Here, the maximum bending angle is enlarged to 60°. The simulation results are shown in Figures 16, 17, 18.

The simulation results show that the proposed method can be easily extended to the manipulator with different physical constraints and different target points. The links of the manipulator fit well with the trajectory to ensure the smooth transition motion and tracking accuracy. The bending angles are limited to satisfy the physical constraints.

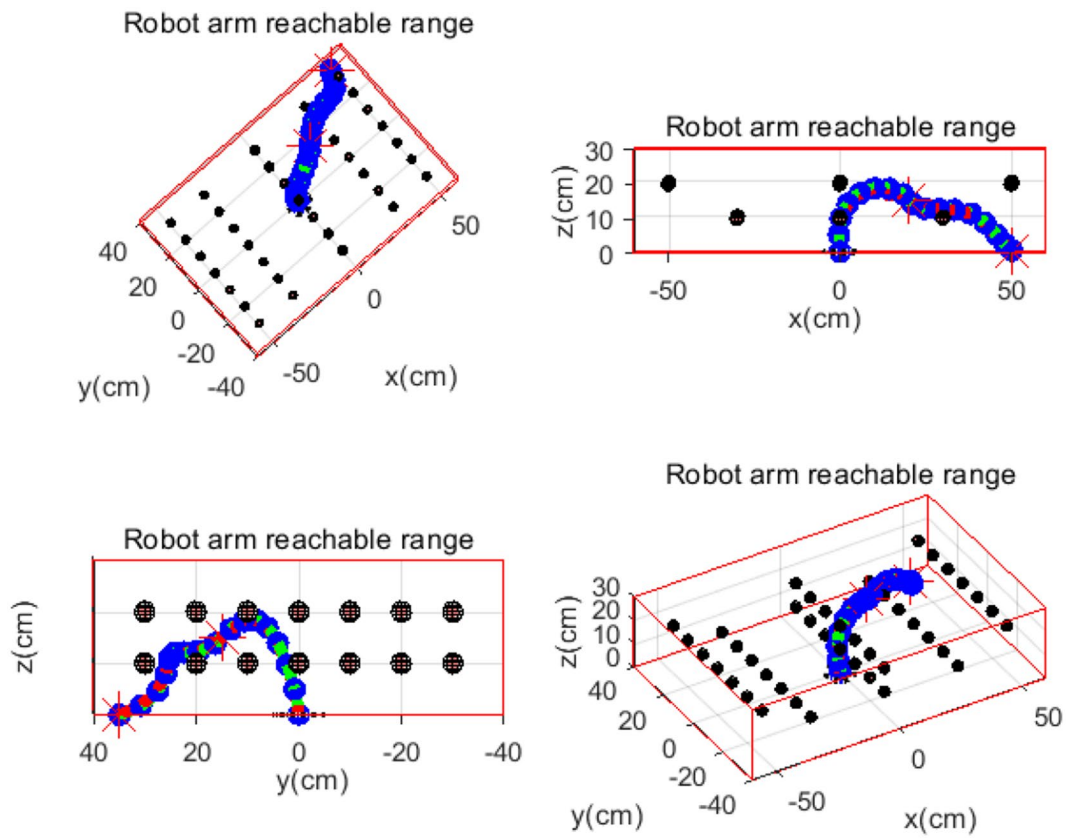


Figure 13 Robot manipulator tracking result

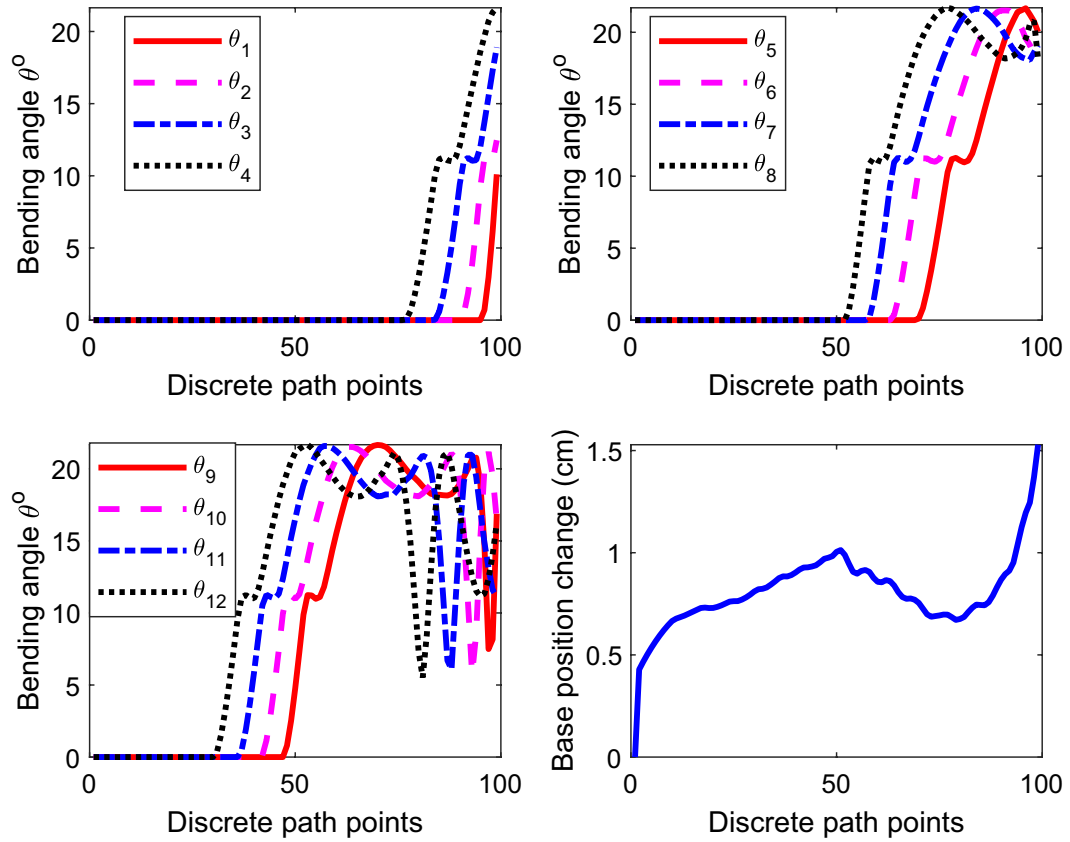
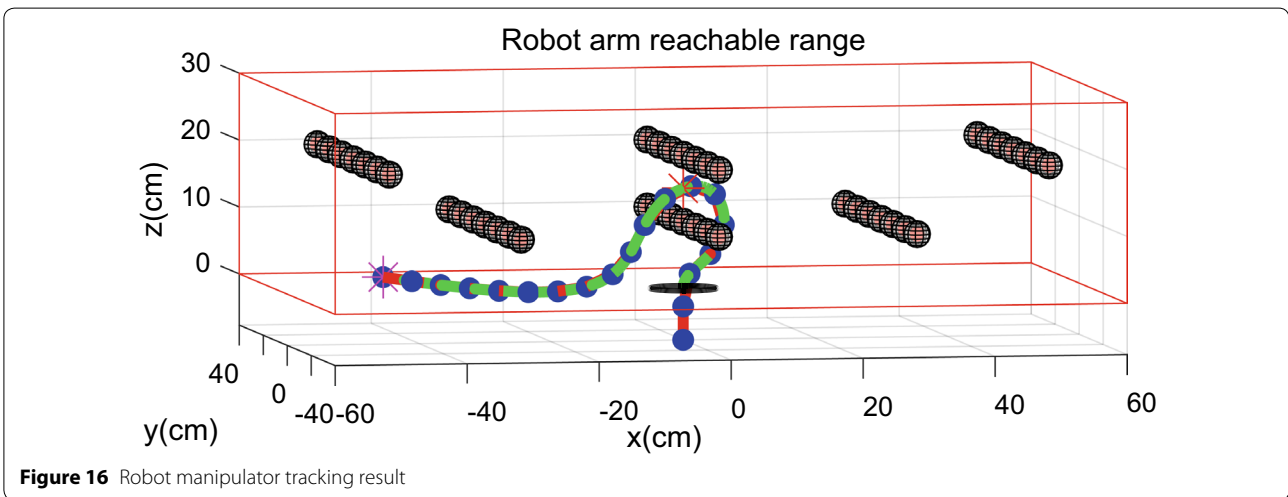
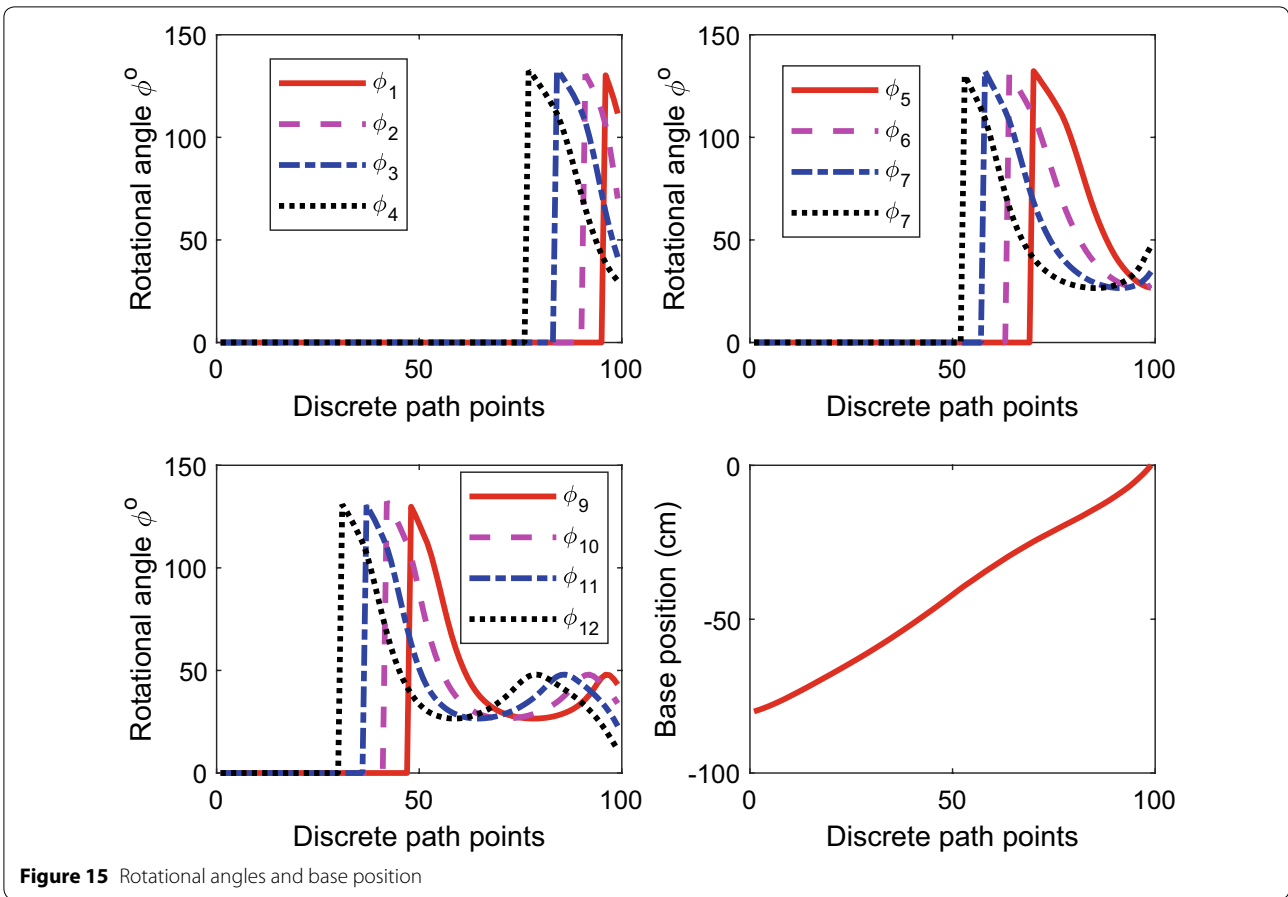


Figure 14 Bending angles and base position change



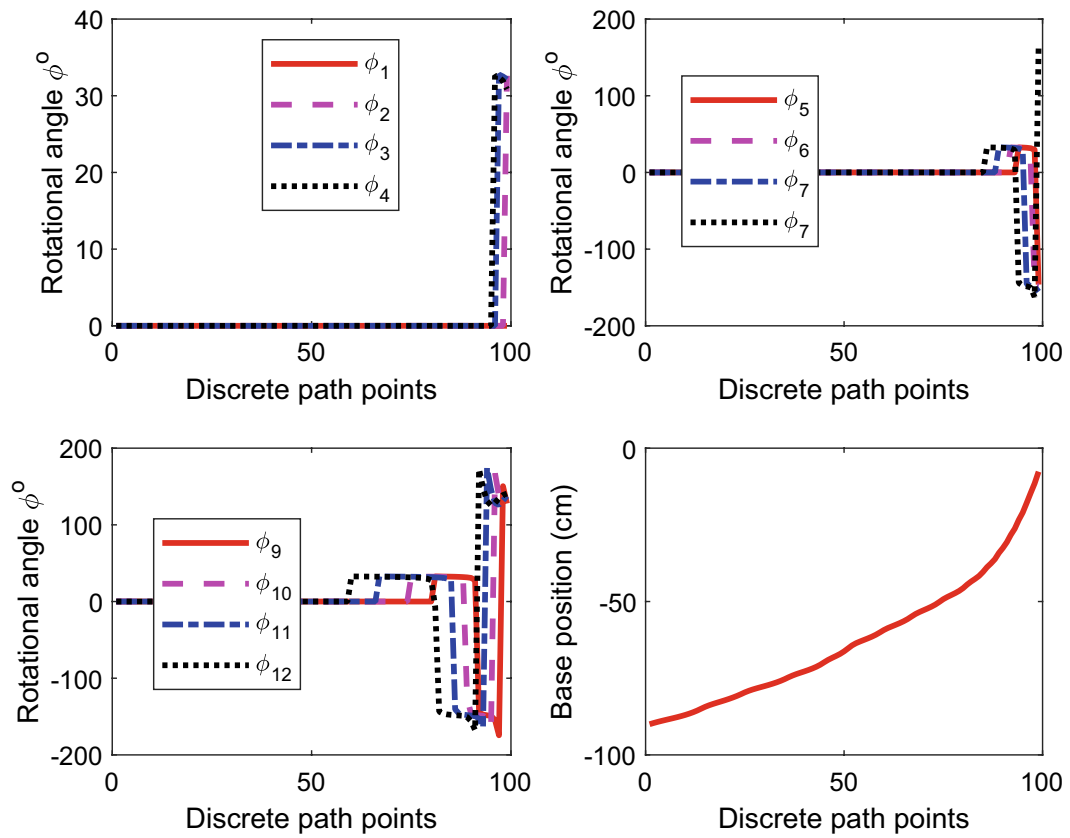
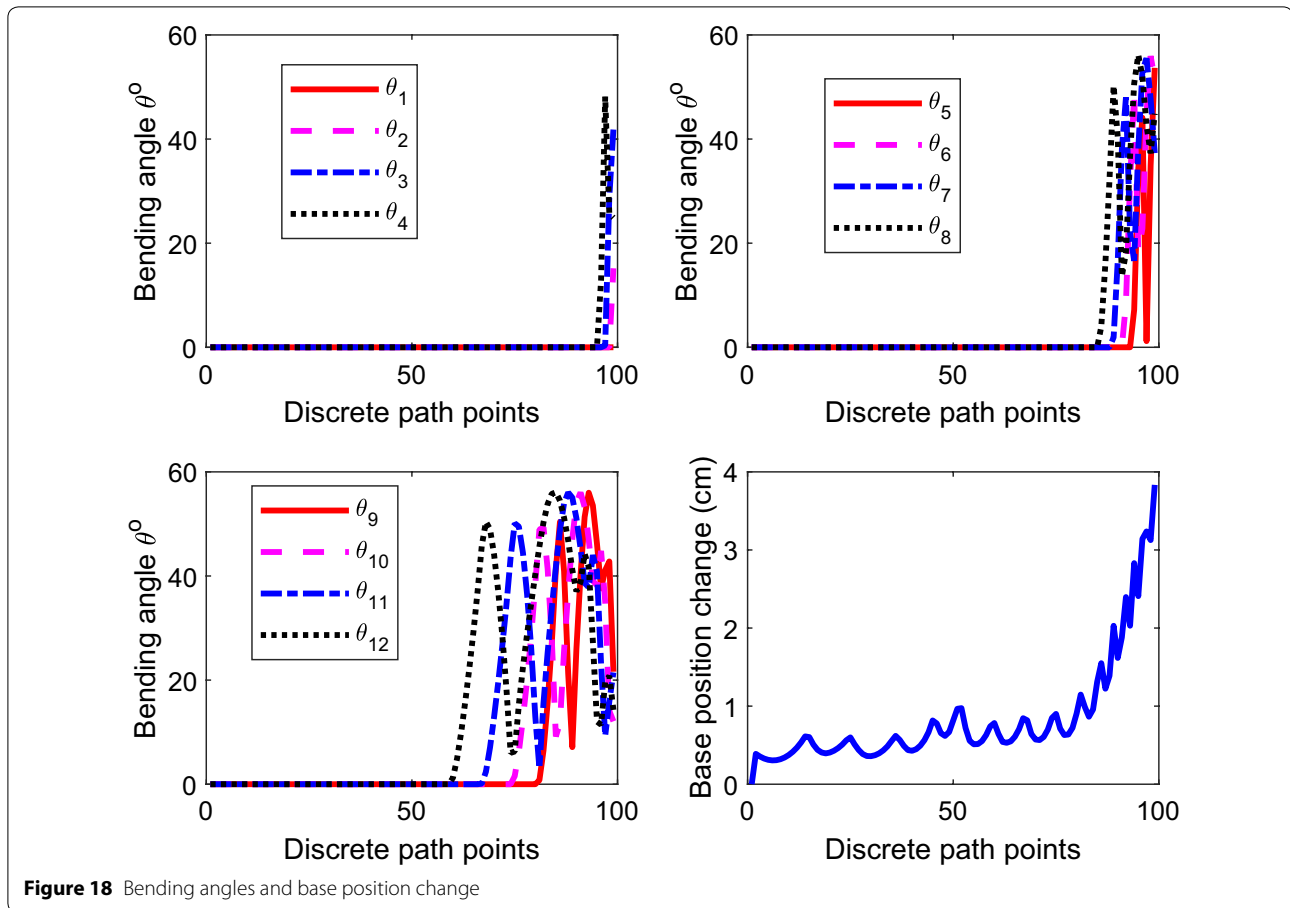


Figure 17 Rotational angles and base position



6 Conclusions

This paper presents an integrated optimization method to solve path planning and trajectory tracking simultaneously. Metaheuristic optimization is used to choose the optimal solution within the physical constraints in the following trajectory of a given hyper-redundant manipulator. The innovations of this paper are as follows: (1) The motion of manipulator is guided by using a tip following strategy, and the joint position update is derived by using a geometric method. (2) The control joint angles are calculated in the inertial space, thereby avoiding the complex IK calculation. (3) The proposed method is suitable for multitarget tracking and effective for obstacle avoidance in 3D spaces.

The limitations of the proposed numerical optimization problem are as follows: (1) The feasible solution of particle filter numerical optimization problem is related to the selection of initial values. Thus, giving reasonable initial solutions are necessary. (2) The obstacle avoidance algorithm is based on the obstacles with spherical profile. The obstacle checking algorithm should be changed accordingly if the obstacle is a hole or other shapes. Future

studies should include the adaptive change of parameters in the PSO algorithm to accelerate the convergence speed under strict constraints.

Acknowledgements

The authors would like to thank Associate Professor Jinjun Rao of Shanghai University for his crucial discussion and valuable comments during manuscript preparation.

Authors' Contributions

LC was in charge of the whole trial; YM wrote the manuscript; YZ assisted with the simulations; JL was in charge of technical expression and language. All authors read and approved the final manuscript.

Authors' Information

Li Chen, born in 1975, is currently a professor at *School of Air Transportation, Shanghai University of Engineering Science, China*. She received her Ph.D. degree from *State Key Laboratory of Robotics, Shenyang Institute of Automation (SIA), China*, in 2004. Her research interests are controller design, dynamic modelling.

Ying Ma, born in 1995, is currently a master candidate at *School of Air Transportation, Shanghai University of Engineering Science, China*. Her research interests are robot control.

Yu Zhang, born in 1996, is currently a master candidate at *School of Aeronautics and Astronautics, Shanghai Jiao Tong University, China*. His research interests are fluid dynamics.

Jinguo Liu, born in 1978, is currently a professor at *Shenyang Institute of Automation, Chinese Academy of Sciences (CAS), China*. He received his Ph.D. degree in Robotics from *Shenyang Institute of Automation (SIA), Chinese Academy of Sciences (CAS), China*, in 2007. His research interests include bioinspired robot, space robot, human–robot interaction, and their practical applications in extreme environment.

Funding

Supported by National Natural Science Foundation of China (Grant No. 61733017), Foundation of State Key Laboratory of Robotics of China (Grant No. 2018O13), and Shanghai Pujiang Program of China (Grant No. 18PJD018).

Competing Interests

The authors declare no competing financial interests.

Author Details

¹ School of Air Transportation, Shanghai University of Engineering Science, Shanghai 201620, China. ² State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China. ³ School of Aeronautics and Astronautics, Shanghai Jiao Tong University, Shanghai 200240, China.

Received: 23 March 2020 Revised: 17 September 2020 Accepted: 7 October 2020

Published online: 28 October 2020

References

- B Robert. Robots in the nuclear industry: A review of technologies and application. *Industrial Robot: an International Journal*, 2011, 38(2): 113–118.
- R Buckingham, C Chitrakaran, R Conkie, et al. Snake-manipulator robots: A new approach to aircraft assembly. *SAE*, 2007-01-3870.
- D Trivedi, C D Rahn, W M Kier, et al. Soft robotics: Biological inspiration, state of the art, and future research. *Applied Bionics and Biomechanics*, 2008, 5(3): 99–117.
- J Z Yang, P E Pitarch, J Potratz, et al. Synthesis and analysis of a flexible elephant trunk robot. *Advanced Robotics*, 2006, 20(6): 631–659.
- M W Hannan, I D Walker. Kinematics and the implementation of an elephant's trunk manipulator and other continuum style robots. *Journal of Field Robotics*, 2003, 20(2): 45–63.
- D Nahar, P M Yanik, I D Walker. Robot tendrils: Long, thin continuum robots for inspection in space operations. 2017 IEEE Aerospace Conference, Big Sky, MT, USA: IEEE, 2017: 1–8
- X Dong, D Axinte, D Palmer, et al. Development of a slender continuum robotic system for on-wing inspection repair of gas turbine engines. *Robotics and Computer-Integrated Manufacturing*, 2017, 44: 218–229.
- H Y Hu, M T Li, P F Wang. Development of a continuum robot for colonoscopy. *High Technology Letters*, 2009, 15(2): 115–119.
- E S Conkur. Path following algorithm for highly redundant manipulators. *Robotics and Autonomous Systems*, 2003, 45(1): 1–22.
- E K Xidias. Time-optimal trajectory planning for hyper-redundant manipulators in 3D workspaces. *Robotics and Computer-Integrated Manufacturing*, 2018, 50: 286–298.
- G C Niu, Z C Zheng, Q J Gao. Collision free path planning based on region clipping for aircraft fuel tank inspection robot. *International Conference on Robotics & Automation*, Hong Kong, China: IEEE, 2014: 3227–3233.
- I S Godage, D T Branson, Guglielmino E, et al. Path planning for multi-section continuum manipulators. *IEEE International Conference on Mechatronics and Automation*, Piscataway, USA: IEEE, 2012: 1208–1213.
- H Ananthanarayanan, R Ordonez. Real-time inverse kinematics of $(2n+1)$ DOF hyper-redundant manipulator via a combined numerical and analytical approach. *Mechanism and Machine Theory*, 2015, 91: 209–226.
- A Aristidou, J F Lasenby. A fast, iterative solver for the inverse kinematics problem. *Graphical Models*, 2011, 73(5): 243–260.
- P David, C G Salvador. Real-time method for tip following navigation of continuum snake manipulator robots. *Robotics and Autonomous Systems*, 2014(62): 1478–1485.
- N Maria, B K Jessica. Considerations for follow-the-leader motion of extensible tendon-driven continuum robots. *2016 IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 16–21, 2016: 917–923.
- S Tappe, J Pohlmann, J Kotlarski, et al. Optimization strategies for task specific path-following capabilities of a binary actuated snake-Like robot using follow-the-leader. *Control 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, Munich, Germany, July 3–7, 2017: 1574–1582.
- Z L Xiong, J F Tao, C L Liu. Kinematics of hyper-redundant snake-arm robots with improved tip following movement. *Robot*, 2018, 40(1): 37–46. (in Chinese)
- D Palmer, D Axinte. Active uncoiling and feeding of a continuum arm robot. *Robotics and Computer Integrated Manufacturing*, 2019, 56: 107–116.
- W Xu, Z G Mu, T Liu, et al. A modified modal method for solving the mission-oriented inverse kinematics of hyper-redundant space manipulators for on-orbit servicing. *Acta Astronautica*, 2017, 139: 54–66.
- H B Xie, C Wang, S S Li, et al. A geometric approach for follow the-leader motion of serpentine manipulator. *International Journal of Advanced Robotic Systems*, 2019, 1–18.
- T Collins, W M Shen. PASO: An integrated, scalable PSO-based optimization framework for hyper-redundant manipulator path planning and inverse kinematics. In: *ISI Tech Report*, January 2016.
- Y Bulut, E S Conkur. A real-time path-planning algorithm with extremely tight maneuvering capabilities for hyper-redundant manipulators. *Engineering Science and Technology an International Journal*, 2020. <https://doi.org/10.1016/j.jestech.2020.07.002>.
- J Y Li, R Z Sun, C M Cheng. Roaming path generation algorithm and optimization based on Bezier curve. *IFAC-Papers Online*, 2018, 51(17): 339–345.
- A Machmudah, S Parman, M B Baharom. Continuous path planning of kinematically redundant manipulator using Particle Swarm Optimization. *International Journal of Advanced Computer Science and Applications*, 2018, 9(3): 207–217.
- K Kawabata, L Ma, J R Xue. A path generation for automated vehicle based on Bezier curve and via-points. *Robotics and Autonomous Systems*, 2015, 74: 243–252.
- L C Yu, K Q Wang, Q H Zhan. Trajectory planning of a redundant planar manipulator based on joint classification and particle swarm optimization algorithm. *Multibody Syst. Dyn.*, 2019. <https://doi.org/10.1007/s11044-019-09720-1>.
- N Homsup, W Silabut, V Kesorpatumanum, et al. A new technique to design planar dipole antennas by using Bezier curve and Particle Swarm Optimization. *Archives of Electrical Engineering*, 2016, 65(3): 513–525.
- K Tang, Z Li, L Luo, et al. Multi-strategy adaptive particle swarm optimization for numerical optimization. *Eng. Appl. Artif. Intell.*, 2015, 37: 9–19.
- R V Rama, P M Pathaka, S J Juncob. Inverse kinematics of mobile manipulator using bidirectional particle swarm optimization by manipulator decoupling. *Mechanism and Machine Theory*, 2019, 13: 385–405.

Submit your manuscript to a SpringerOpen® journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)