

## Article

# Research on Configuration Design Optimization and Trajectory Planning of Manipulators for Precision Machining and Inspection of Large-Curvature and Large-Area Curved Surfaces

Xiangyang Sun <sup>1,2,3</sup>, Shuai He <sup>1,3,\*</sup> , Zhenbang Xu <sup>1,3,4,\*</sup> , Enyang Zhang <sup>1,3</sup> and Yanhui Li <sup>1,3</sup>

<sup>1</sup> Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China; sunxiangyang19@mails.ucas.ac.cn (X.S.); zhangenyang@ciomp.ac.cn (E.Z.); liyanhui@ciomp.ac.cn (Y.L.)

<sup>2</sup> University of Chinese Academy of Sciences, Beijing 100049, China

<sup>3</sup> CAS Key Laboratory of On-Orbit Manufacturing and Integration for Space Optics System, Chinese Academy of Sciences, Changchun 130033, China

<sup>4</sup> Center of Materials Science and Optoelectronics Engineering, University of Chinese Academy of Sciences, Beijing 100049, China

\* Correspondence: heshuai@ciomp.ac.cn (S.H.); xuzhenbang@ciomp.ac.cn (Z.X.)

**Abstract:** In recent years, high-quality surfaces with large areas and curvatures have been increasingly used in engineering, but the precision machining and inspection of such surfaces is a particular challenge. Surface machining equipment needs to have a large working space, high flexibility, and motion accuracy to meet the demands of micron-scale precision machining. However, meeting these requirements may result in extremely large equipment sizes. To solve this problem, an eight-degree-of-freedom redundant manipulator with one linear and seven rotational joints is designed to assist in the machining described in this paper. The configuration parameters of the manipulator are optimized by an improved multi-objective particle swarm optimization algorithm to ensure that the working space of the manipulator completely covers the working surface and that the size of the manipulator is small. In order to improve the smoothness and accuracy of manipulator motion on large surface areas, an improved trajectory planning strategy for a redundant manipulator is proposed. The idea of the improved strategy is to pre-process the motion path first and then use a combination of the clamping weighted least-norm method and the gradient projection method to plan the trajectory, while adding a reverse planning step to solve the singularity problem. The resulting trajectories are smoother than those planned by the general method. The feasibility and practicality of the trajectory planning strategy are verified through simulation.

**Keywords:** redundant manipulator; large curvature; large area surface; configuration optimization; trajectory planning



**Citation:** Sun, X.; He, S.; Xu, Z.; Zhang, E.; Li, Y. Research on Configuration Design Optimization and Trajectory Planning of Manipulators for Precision Machining and Inspection of Large-Curvature and Large-Area Curved Surfaces. *Micromachines* **2023**, *14*, 886. <https://doi.org/10.3390/mi14040886>

Academic Editors: Youqiang Xing, Xiuqing Hao and Duanzhi Duan

Received: 30 March 2023

Revised: 13 April 2023

Accepted: 14 April 2023

Published: 20 April 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid development of science and technology, many mechanical products are being updated, and the requirements for large surfaces with large curvature in mechanical products are becoming more and more demanding, such as in the head shells of aircraft, rockets, and moving vehicles. In order to improve the quality and production efficiency of curved surfaces, robots have been widely used in processes such as grinding, polishing, painting, and inspection of high-quality curved surfaces [1–7].

There are two prerequisites for robots to be able to perform the task of assisting in the micron-scale precision machining of surfaces. One is that the working space of the robot covers the working surface, and the other is a suitable surface movement trajectory. The working space of the robot has a great relationship with the structure of the robot. Nowadays, robots commonly used for surface machining include SCARA robots and

five- and six-axis industrial robots, which are mainly used for flat machining and small-curvature surface machining [8–10]. Some researchers have investigated parallel and hybrid robots, which have the advantage of a large working space and good stiffness. However, their disadvantage is that the robots are large and costly and need to work in a specific workplace [11–16]. So, it is necessary to design a manipulator for surface machining with a large working space and a relatively small size.

To ensure the quality of the surface, the trajectory of the manipulator needs to be smooth when moving on the surface, and the common method for manipulator trajectory planning is to interpolate the trajectory points using straight lines, circular arcs, polynomial curves, B-spline curves, S-curves, and so on [17–19]. Zhang Peng et al. studied a large curvature surface spraying robot and its spraying path planning, combined with the idea of a particle swarm optimization algorithm, and wrote an algorithm suitable for the study of large curvature surface spraying trajectory sequencing and combination problems [20,21], but the spraying area of the robot they studied was less than 1 m<sup>2</sup>. Shao Junyi et al. studied a redundant robot for pipe inner wall spraying. A redundant robot trajectory planning method was established for general spatial curved pipe-type parts inner surface spraying operations [22]. These surface trajectory planning methods are mainly applicable to small surfaces; however, the complexity and length of the trajectory increase significantly when machining large-curvature surfaces with large areas. This requires a trajectory planning method that can quickly plan a long, continuous, and smooth trajectory, but the results obtained by general trajectory planning methods are often unsatisfactory.

This paper designs an eight-degree-of-freedom redundant surface scanning manipulator. A multi-objective particle swarm optimization algorithm is used to optimize its configuration, resulting in a large available working space and a relatively small size. A surface trajectory planning algorithm combining the pincer-weighted minimum parametric method and the projected gradient method is proposed to improve the continuity and smoothness of the manipulator trajectory, and the algorithm is verified by simulation and experiment.

## 2. Configuration Design and Optimization

### 2.1. Configuration Design

When working on large surfaces with large curvature, the working space of the manipulator must be able to cover the surface completely, and the manipulator must have as few singularities as possible when moving on the surface. It is also important to consider the utilization of the working space. In general, the working space of a robot is similar to a hollow sphere, so if a large convex surface is to be covered completely, a large part of the working space will be left unused. As an example, take a shaped spherical surface of large curvature with a radius of 600 mm and an angle of  $-10^\circ$  to  $60^\circ$ . The shape of the surface is shown in Figure 1. A typical six-degree-of-freedom industrial robot has a workspace similar to a notched hollow sphere, as shown in Figure 2, and a large part of its workspace will be unused when it completely covers the working surface.

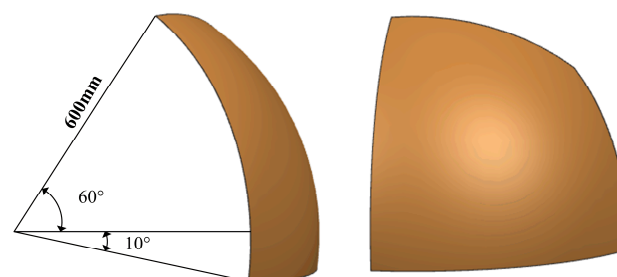
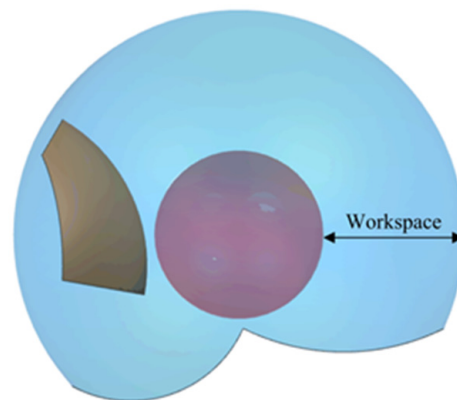
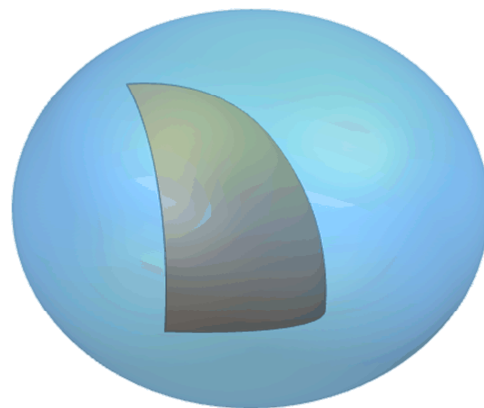


Figure 1. Working surface.



**Figure 2.** Workspace of industry robot.

In this paper, an eight-degree-of-freedom redundant manipulator is designed to meet the working requirements. The structure of the manipulator is designed with one horizontal moving joint and seven rotating joints. The seven rotating joints increase the flexibility of the manipulator, allowing it to reach all positions on the machined surface. The horizontal moving joint extends the working space of the manipulator into an ellipsoidal shape, increasing the utilization of the working space. The working space of the manipulator designed in this paper is shown in Figure 3.



**Figure 3.** Workspace of eight-degree-of-freedom redundant manipulator.

The configuration of the manipulator and the linkage coordinate system are shown in Figure 4. The D-H parameters of the manipulator are shown in Table 1. DH parameters are the most common and concise way to determine the manipulator configuration, and the accuracy of DH parameters has a great impact on the motion planning of the manipulator. Structural optimization of the manipulator means optimization of DH parameters. Moreover, the DH parameters can be set and calibrated to reduce the impact of assembly errors on the accuracy of the manipulator [23]. In Table 1,  $L_1$  is the distance from the rotating joint to the moving joint, and  $L_2$ ,  $L_3$ , and  $L_4$  are the linkage lengths of the manipulator. The lengths of  $L_2$  and  $L_4$  are determined according to the requirements of stiffness, joint structure, and load-bearing capacity, while the lengths of  $L_1$  and  $L_3$  have to be optimized to obtain suitable values according to the working requirements.

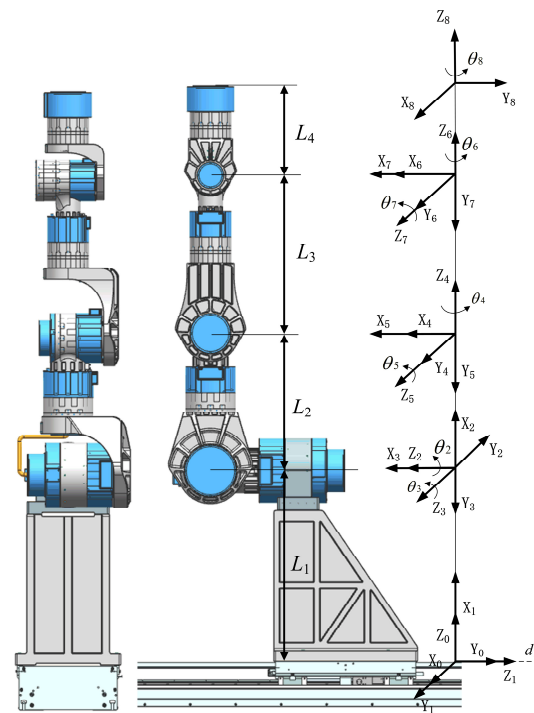


Figure 4. Configuration of the manipulator.

Table 1. D-H parameters.

Joint $i$	$\alpha_{i-1}/(^{\circ})$	$a_{i-1}/\text{mm}$	$d_i/\text{mm}$	$\theta_i/(^{\circ})$
1	-90	0	$d_1$	-90
2	180	$L_1$	0	$\theta_2$
3	90	0	0	$\theta_3 + 90$
4	90	0	$L_2$	$\theta_4$
5	-90	0	0	$\theta_5$
6	90	0	$L_3$	$\theta_6$
7	-90	0	0	$\theta_7$
8	90	0	$L_4$	$\theta_8 + 90$

### 2.2. Configuration Optimization

The working space of the manipulator is directly related to the length of the linkage. Meanwhile, the structural stiffness of the manipulator is negatively related to the length of the linkage; the longer the linkage, the less stiff the manipulator. Therefore, there are two objectives for optimizing the manipulator configuration. One is to satisfy the requirement that the working space completely covers the working surface and that the manipulator moves on the surface with fewer singularities. The other is to keep the overall length of the manipulator as short as possible to reduce its weight and increase its stiffness. To find out the optimum configuration that meets the requirements, the manipulator configuration parameters  $L_1$  and  $L_3$  need to be optimized according to the working surface.

The most common optimization algorithms are genetic algorithms, ant colony algorithms, neural network algorithms, and particle swarm algorithms [24–27]. The particle swarm optimization algorithm (PSO) is a heuristic algorithm inspired by bird flock choreography. The basic principle of particle swarm optimization algorithms is to treat the solution to an optimization problem as a particle in the search space, with each particle having a fitness value determined by the objective function. The size of the fitness value determines whether the particle is superior or inferior. Each particle also has a search speed that determines the direction and distance of the search, which is related to the historical optimal value of the particle itself and the overall optimal value of the particle

population. The optimal solution to the optimization problem can be obtained after a certain number of iterations.

Particle swarm algorithms have been found to be successful in a variety of optimization problems. The particle swarm algorithm performs well in single-objective optimization problems and converges very quickly [28–30]. Coello et al. proposed a multi-objective particle swarm optimization algorithm, confirming the feasibility and superiority of particle swarm algorithms for multi-objective optimization problems [31]. The multi-objective swarm optimization algorithm solves an optimization problem with  $k$  objective functions, which means that the fitness value of each particle is a  $k$ -dimensional vector, and it is not possible to simply compare and determine whether the particles are superior or inferior. Therefore, the multi-objective swarm optimization algorithm uses Pareto Dominance, which is defined as follows: A vector  $\vec{u} = (u_1, u_2, \dots, u_k)$  is said to dominate  $\vec{v} = (v_1, v_2, \dots, v_k)$  (denoted by  $\vec{u} \prec \vec{v}$ ) if and only if  $u$  is partially less than  $v$ , i.e.,

$$\forall i \in \{1, \dots, k\} : u_i \leq v_i \wedge \exists i \in \{1, \dots, k\} : u_i < v_i \quad (1)$$

The global optimal solution of a multi-objective particle swarm optimization algorithm is a set, and the solutions in the solution set are all non-dominated optimal solutions in the sense that none of them will be dominated by the other solutions. The global optimal solution is selected by roulette from the set of global optimal solutions when updating the particle velocity.

In this paper, the multi-objective particle swarm optimization algorithm is used to optimize the configuration of the manipulator. The elements in the particles are the lengths of the linkages  $L_1$  and  $L_3$ , and the range of values of  $L_1$  and  $L_3$  is the search space of the optimization problem. The total length of the manipulator is used as the first objective function. The number of singularities encountered during the trajectory planning process is used as the second objective function. Each particle has fitness values  $F_1$  and  $F_2$  determined by the two objective functions. The range of values for  $L_1$  and  $L_3$  in the multi-objective swarm optimization algorithm is a large range estimated according to the design criteria, so some particles may encounter no solutions for the trajectory points during the optimization process, i.e., invalid solutions. If these invalid particles are removed without processing, other particles may still become invalid during the search, which will greatly reduce the convergence speed. In order to solve the problem of slow convergence due to the wide range of particles, this paper proposes an improved optimization algorithm combining the particle swarm algorithm and the artificial potential field method. The improved algorithm is divided into two steps. The first step is to carry out a smaller number of iterations of optimization. In the optimization process, when the invalid particles are encountered, they will be considered obstacles. The exclusion potential field will be set at the obstacles, giving the particles in the vicinity of the invalid solution a speed away from the obstacles. The search speed of the particles is increased, and the entire search space is searched quickly. At the same time, all the invalid solutions are recorded to form the invalid solution set. After the optimization is complete, a more accurate range of particle values can be derived from the distribution of particles in the invalid solution set. Then, in the second optimization step, the range of particle values is changed to the range obtained in the first step so that the probability of encountering invalid solutions is greatly reduced. The invalid solution set is also substituted as an obstacle in the second step of the algorithm so that the particles can keep away from the invalid solution area, thus speeding up the convergence.

The improved multi-objective particle swarm optimization algorithm proposed in this paper is used to optimize the lengths of the linkages  $L_1$  and  $L_3$ . The initial range of values of  $L_1$  and  $L_3$  is given as  $L_1 \in (300, 600)$ ,  $L_3 \in (300, 500)$ . The set of invalid solutions is obtained by the first step of the improved algorithm, as shown in Figure 5, and according to the region where the invalid solutions are located, the new range of values for  $L_1$  and  $L_3$  is determined as  $L_1 \in (480, 600)$ ,  $L_3 \in (380, 480)$ . The invalid solution is then substituted as an obstacle in the second optimization step, and the result obtained from the optimization

is shown in Figure 6. Finally, the values of  $L_1$  and  $L_3$  are determined to be  $L_1 = 480$  and  $L_3 = 418$ .

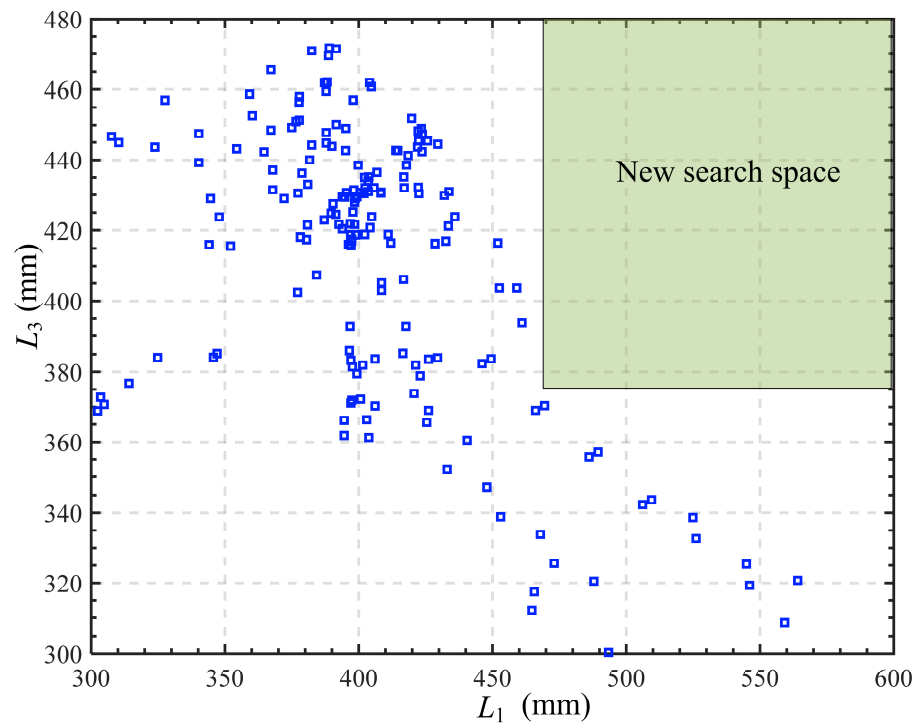


Figure 5. The result of the first step of optimization.

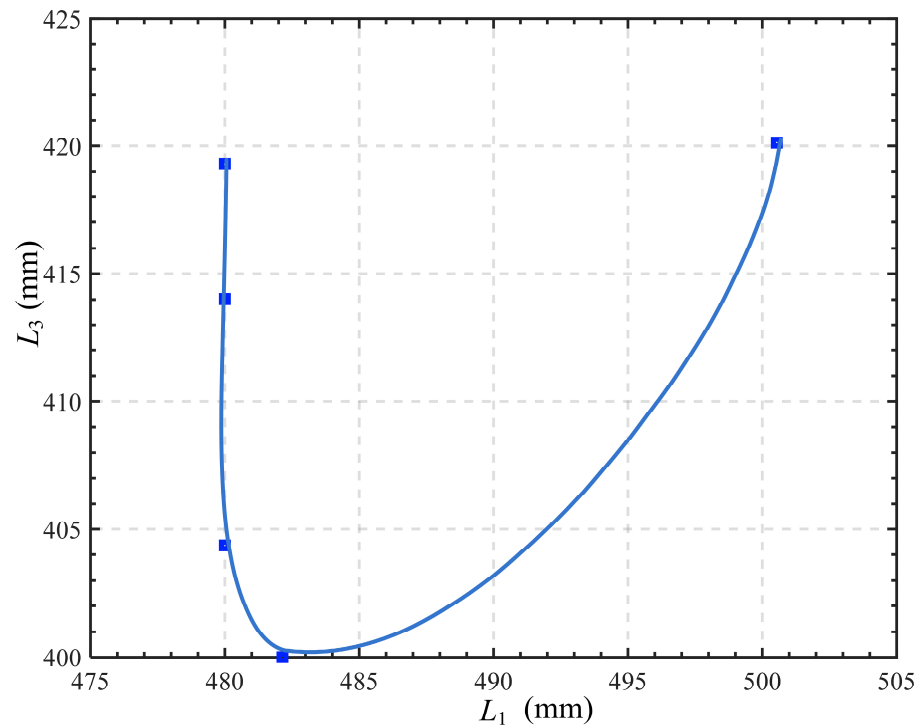


Figure 6. The result of the second step of optimization.

### 3. Trajectory Planning

#### 3.1. Trajectory Planning Method

Trajectory planning is the calculation of the velocity and acceleration of the manipulator based on the motion path, so that the motion of the manipulator is smoother and the drastic changes in velocity and acceleration are reduced [8–10]. The trajectory planning of the manipulator, whether in joint space or in Cartesian space, is to first calculate the joint angle corresponding to the pose at the path point in the path, and then interpolate the joint angle to obtain the joint angular velocity and angular acceleration.

The manipulator designed in this paper is a redundant manipulator, and one end pose of the redundant manipulator corresponds to an infinite number of inverse solutions. It means that it is difficult to derive an analytical inverse solution, so the numerical solution method is usually used. The numerical solution method is to use the pseudo-inverse of the Jacobian matrix to find the inverse solution [11]. At the same time, the range of joint angles should be taken into account. The most common numerical methods for solving inverse kinematics considering angle constraints are the gradient projection method and the weighted least-norm method [32,33]. The gradient projection method exploits the existence of a null-space for a singular Jacobi matrix to achieve joint angle restriction by minimizing the penalty function along the gradient direction in the Jacobi null-space of the master task. Null-space means that any angular velocity vector in this space multiplied by the Jacobi matrix results in a zero vector. In other words, the end pose of the manipulator does not change when the joints of the manipulator move according to the angular velocity vector in null-space. The gradient projection method is defined as

$$\dot{\theta} = J^+ \dot{x} + (I_n - J^+ J) H \quad (2)$$

where  $\dot{x}$  is the end velocity of the manipulator as a 6-dimensional vector,  $\dot{\theta}$  is the angular velocity of each joint as an  $n$ -dimensional vector,  $n$  represents the number of degrees of freedom of the manipulator,  $J$  represents the Jacobi matrix, and  $J^+$  represents the pseudo-inverse of the Jacobi matrix.  $I_n$  is a unit matrix of order  $n$ ,  $I_n - J^+ J$  is the null-space projection operator of the Jacobian matrix,  $H \in R^n$  is an arbitrary  $n$ -dimensional vector, and  $(I_n - J^+ J)H$  can be used to perform sub-tasks such as joint limiting.

The weighted least-norm method limits the joint angles by adding weight factors to the joint velocities. As the joints approach their limits, the corresponding joints are set with smaller weight factors, effectively suppressing the operating velocities of the corresponding joints. The weighted least-norm method is effective in reducing invalid self-motion and is therefore more efficient than the gradient projection method. The weighted minimum parametric method is defined as

$$\dot{\theta} = W^{-1} J^T (J W^{-1} J^T + \lambda^2 I_m)^{-1} \dot{x} \quad (3)$$

where  $W$  denotes the weight matrix and the weight matrix is an  $n$ -order diagonal array, each diagonal element is the weight of each joint angle, and  $\lambda^2$  is the damping factor to help deal with the singularity of the manipulator [33].

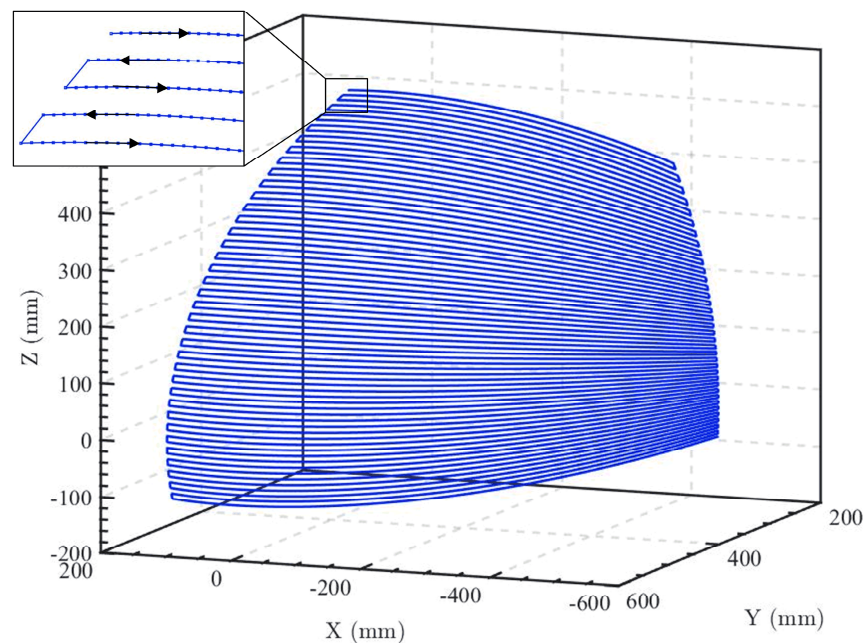
Once the inverse solution for all path points is obtained, the angles of the joints can be interpolated according to the movement time of the manipulator, typically using straight lines, circular arcs, polynomial curves, B-spline curves, S-curves, etc. After interpolation, the velocity and acceleration of the manipulator can be calculated by differentiation. The result of the trajectory planning is highly dependent on the kinematic inverse solution; if there are more drastic changes in the inverse solution, it will lead to drastic changes in the velocity and acceleration of the manipulator, making the trajectory of the manipulator discontinuous.

### 3.2. An Improved Trajectory Planning Strategy

The manipulator studied in this paper is used on a large curvature and a large convex surface, which has a very long moving trajectory during operation. In the trajectory planning process, there is a high probability of encountering singularities that lead to drastic changes in joint angles. Therefore, this paper proposes an improved trajectory planning strategy by segmenting the motion path and setting up self-motion areas at the edges of the path. The inverse solution of the path points is then found based on a combination of the clamping weighted least-norm method and the projected gradient method, and the singular points are moved to the self-motion area by reverse planning when the singular points are difficult to solve by the projected gradient method. By increasing the motion time in the self-motion area, the joint angle changes are not as drastic, and the motion along the working path is guaranteed to be smooth and continuous.

#### 3.2.1. Pre-Processing Motion Path

Using the local spherical surface described in Section 2 as an example, the path planning method used to machine this surface is to discretize the surface into  $P$  path points and then connect all the path points in series with an s-shaped reciprocal curve. The paths are shown in Figure 7. The improved trajectory planning strategy proposed in this paper requires pre-processing of the paths. The first step is to segment the path into  $r$  rows, with  $c$  path points in each row. Next, the path points on each segmented path are given a two-dimensional coordinate shaped as  $x_{i,j}$ , where  $i$  represents the  $i$ -th segment and  $j$  represents the  $j$ -th path point of that segment, and the next point at the end of each segmented path is the starting point of the next segmented path. Finally, the transition between segments and the region at the edge of the path is set as the self-motion area. The pre-processed motion path is shown in Figure 8.



**Figure 7.** Motion path.



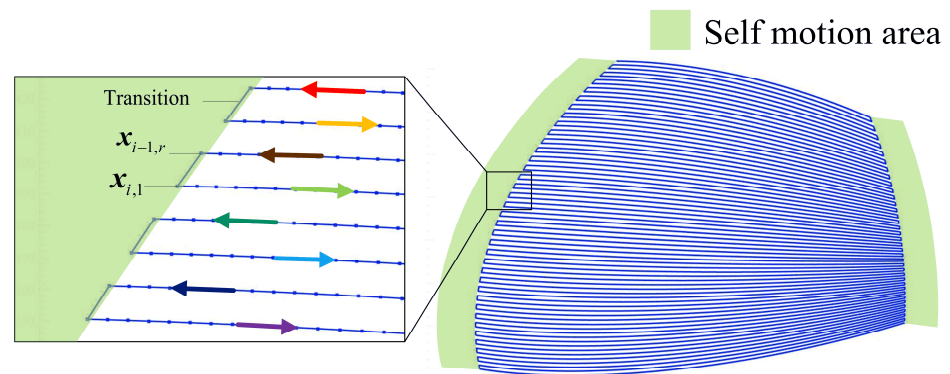


Figure 8. Processed motion path.

### 3.2.2. Trajectory Planning Strategy

This paper uses a combination of the clamping weighted least-norm method and the projected gradient method to find the inverse of the path point. The clamping weighted least-norm method is based on the weighted least-norm method with the addition of a clamp term  $(I_n - \tilde{W})\phi(\theta)$ , which has the advantage of helping the joint to move away from the joint limit as soon as possible when the joint approaches the angle limit [34]. The clamping weighted least-norm method is defined as

$$\dot{\theta} = -(I_n - \tilde{W})\phi(\theta) + \tilde{W}J^T(J\tilde{W}J^T + \lambda^2I_m)^{-1}((x_{i+1} - x_i) + J(I_n - \tilde{W})\phi(\theta)) \quad (4)$$

where  $\tilde{W}$  represents the weighting and  $\phi(\theta)$  represents the clamp task.

The specific steps of trajectory planning are as follows: (1) Solve the inverse of the poses of all path points in order according to the determined path coordinates, and use the inverse solution  $\theta_{i,j}$  of the poses of path point  $x_{i,j}$  as the iterative initial value when solving for the inverse solution  $\theta_{i,j+1}$  of the poses of path point  $x_{i,j+1}$ . The iterative solution is carried out by the clamping weighted least-norm method, and the calculation process is as follows:

$$\begin{cases} x_e = \text{forward\_kinematics}(\theta_{i,j}) \\ \dot{\theta} = -(I_n - \tilde{W})\phi(\theta_i) + \tilde{W}J^T(J\tilde{W}J^T + \lambda^2I_m)^{-1}((x_{i,j+1} - x_e) + J(I_n - \tilde{W})\phi(\theta_{i,j})) \\ \theta_{i,j+1} = \theta_{i,j} + \dot{\theta}d_t \\ x_e = \text{forward\_kinematics}(\theta_{i,j+1}) \\ \text{error} = x_{i,j+1} - x_e \\ \theta_{i,j} = \theta_{i,j+1} \end{cases} \quad (5)$$

where  $d_t$  represents the step length and error represents the error between the current pose and the desired pose. When it is less than the given accuracy requirement, the inverse solution  $\theta_{i,j+1}$  of the pose of the path point  $x_{i,j+1}$  is obtained.

(2) Determine whether the inverse solution  $\theta_{i,j+1}$  of path point  $x_{i,j+1}$  is singularity (based on the distance between the path points and the maximum angular velocity of the manipulator joint, the maximum value of the difference between the inverse solutions of the two adjacent points is greater than  $50^\circ$  as the basis for determining whether it is singularity). If it is singularity, the manipulator is first made to self-motion using the projected gradient method, so that the inverse solution  $\theta_{i,j+1}$  of path point  $x_{i,j+1}$  changes without changing the end pose, reducing the difference between it and the inverse solution  $\theta_{i,j}$  of path point  $x_{i,j}$ , and eliminating as much as possible the drastic change in joint angle. If the difference between the inverse solutions is less than  $50^\circ$  after the self-motion, proceed to step 4, otherwise, proceed to step 3.

(3) Randomly solve for path point  $x_{i,j+1}$  to obtain fifty  $\theta_{i,j+1}$  as initial values and carry out reverse planning. Reverse planning is to follow steps 1 and 2 to find the inverse solution of the path points starting at  $x_{i,j+1}$  and ending at  $x_{i,1}$ . When the inverse solution from  $x_{i,j+1}$  to  $x_{i,1}$  is completed for an initial value of  $\theta_{i,j+1}$ , reverse planning is stopped, the inverse solution from  $\theta_{i,j+1}$  to  $\theta_{i,1}$  is replaced with the new one, and trajectory planning continues from  $x_{i,j+1}$ . In this way, the singularity is transferred to  $x_{i-1,r}$ . Although the difference between  $\theta_{i,1}$  and  $\theta_{i-1,r}$  becomes larger, the large incremental change between  $\theta_{i,1}$  and  $\theta_{i-1,r}$  can be accomplished by self-motion in the self-motion area. In this way, the singularities are resolved outside the working surface, and the smooth trajectory of the manipulator on the working surface can be guaranteed. If all the initial values are unsuccessful in reverse planning, then  $x_{i,j+1}$  is marked as a singularity, and a large incremental change between  $\theta_{i,j}$  and  $\theta_{i,j+1}$  can then be achieved by sacrificing some accuracy in the motion. The method is to choose  $\theta_{i,j+1}$ , which has the smallest difference with  $\theta_{i,j}$  in the process of self-motion, as the inverse solution of  $x_{i,j+1}$ . Firstly, the manipulator is moved away from the working surface by the horizontal moving joint, then the remaining seven rotation joints are moved from  $\theta_{i,j}$  to  $\theta_{i,j+1}$ , and finally the horizontal moving joint is moved to  $\theta_{i,j+1}$ . This will make the end of the manipulator out of alignment with the working surface in the process of moving from  $x_{i,j}$  to  $x_{i,j+1}$ , but it ensures that the trajectory of the manipulator passes through all the path points accurately.

(4) Determine whether path point  $x_{i,j+1}$  is the last path point; if it is, then proceed to step 5; otherwise, determine whether  $j + 1$  is equal to  $c$ ; if it is, then make  $i = i + 1$  and proceed to step 1, if not, then make  $j = j + 1$  and proceed to step 1.

(5) Interpolate the resulting joint angle in the joint space with a quintic polynomial to obtain the joint angular velocity and angular acceleration of the manipulator, and the trajectory planning is completed.

The overall flow of the trajectory planning strategy is shown in Figure 9.

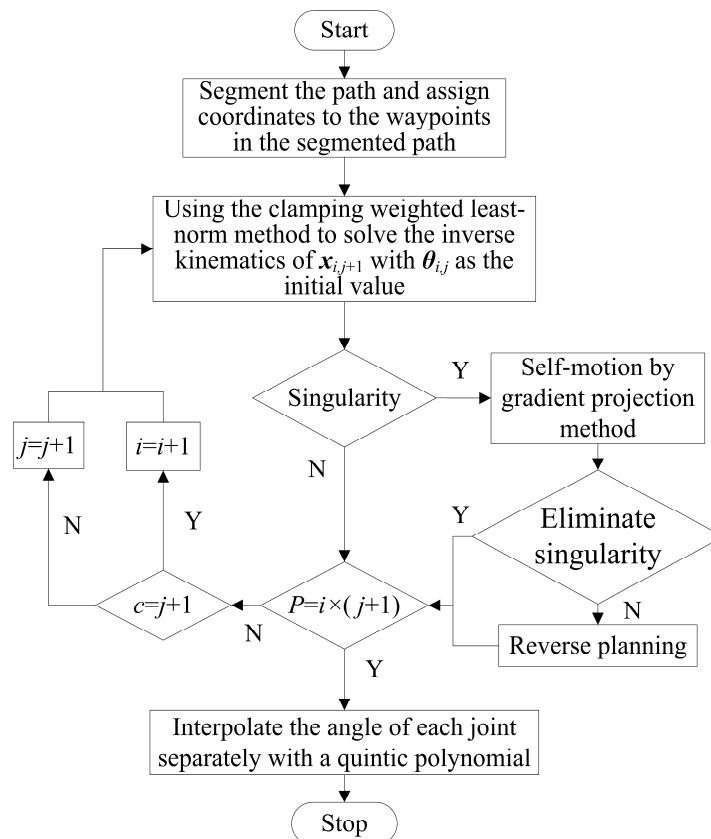


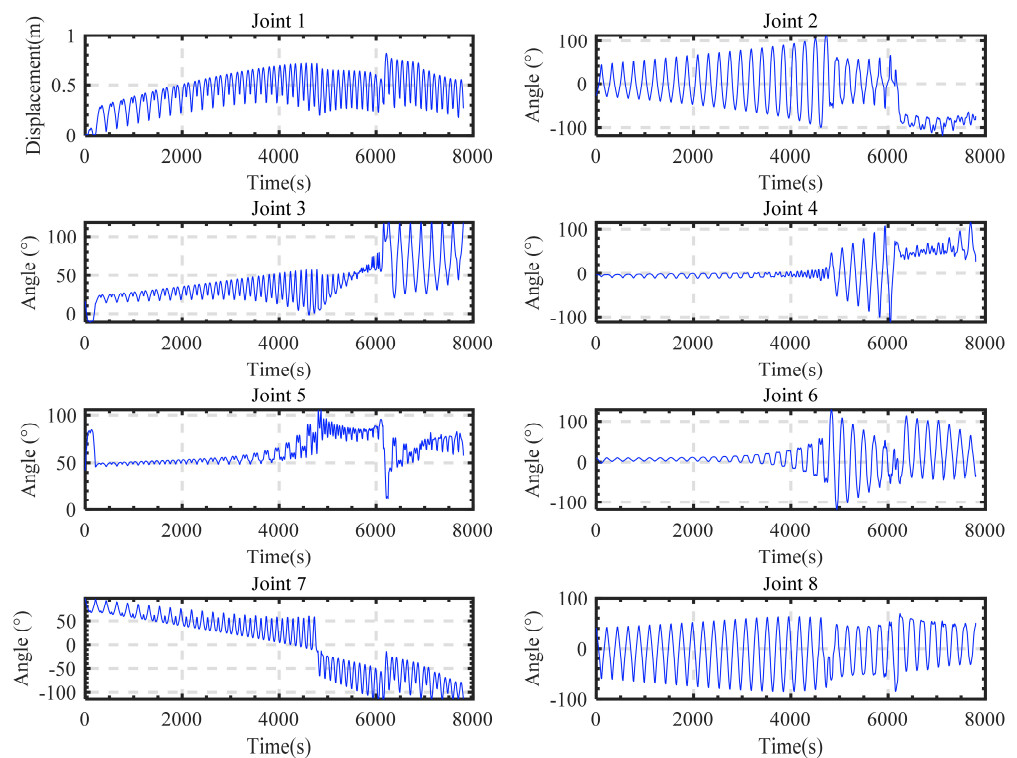
Figure 9. Trajectory planning strategy flowchart.

#### 4. Simulation and Experimentation

In order to verify the feasibility of the trajectory planning strategy proposed in this paper, the optimized manipulator in Section 2 was used as the manipulator model for the simulation experiments. The DH parameters of the optimized manipulator are calibrated to reduce the impact of assembly errors on the motion accuracy of the manipulator and to improve the smoothness of the trajectory. The improved trajectory planning strategy described in this paper was used to plan the scanning trajectory of the working sphere described in Section 3, and the motion trajectory planned with the improved strategy is shown in Figures 10–12. Figure 10 shows the angle of each joint of the trajectory planned with the improved strategy. Figure 11 shows the velocity of each joint. Figure 12 shows the acceleration of each joint. It can be seen that the velocity and acceleration planned by the improved method are smooth.

At the same time, the inverse solution was solved using only the clamping weighted least-norm method and then interpolated using a quintic polynomial as a comparison. Figures 13–15 show the results of the planning using the general method, where Figure 13 represents the angle of each joint of the trajectory, Figure 14 represents the joint velocity, and Figure 15 represents the joint acceleration. It can be seen from the graphs that there are several dramatic variations in the velocities and accelerations planned using the general method. A comparison of the two planned trajectories shows that the trajectories planned using the improved trajectory planning strategy are smoother. The feasibility of the improved trajectory planning strategy is demonstrated.

A section of the planned path is selected and run using motion simulation software and a real manipulator arm. The simulated motion is shown in Figure 16, and the motion of the manipulator is shown in Figure 17. From Figures 16 and 17, it can be seen that the simulated motion is consistent with the actual motion of the manipulator, demonstrating the practicality of the trajectory planning method.



**Figure 10.** The angle of each joint of the trajectory planned by the improved trajectory strategy.

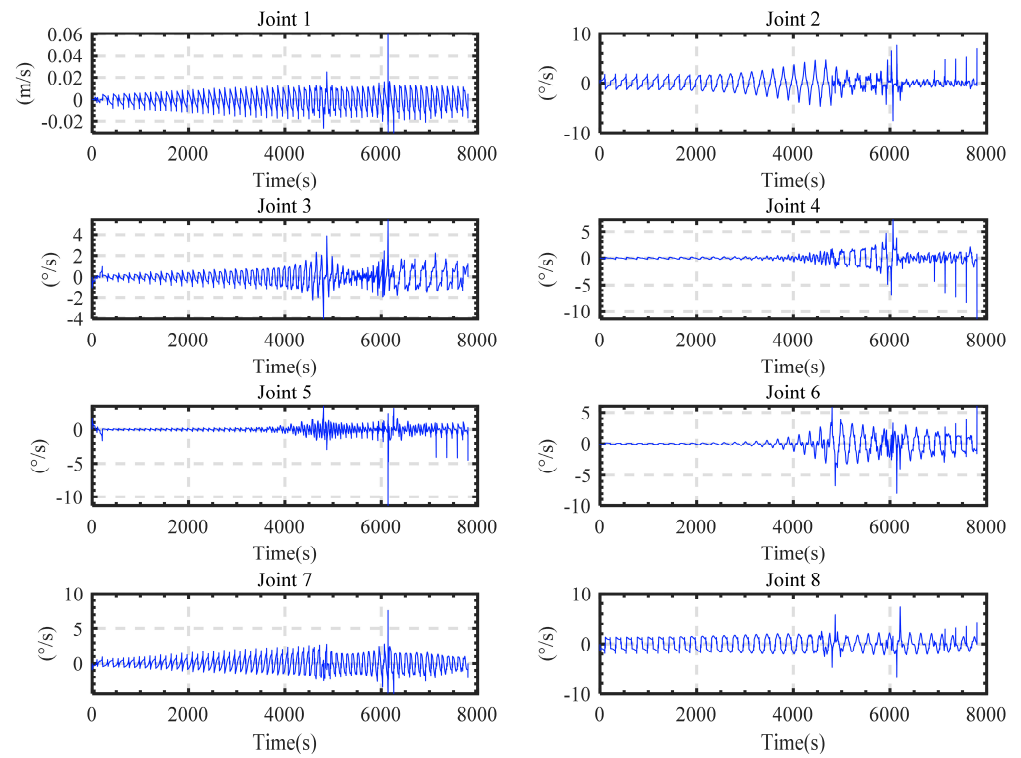


Figure 11. The velocity of each joint of the trajectory planned by the improved trajectory strategy.

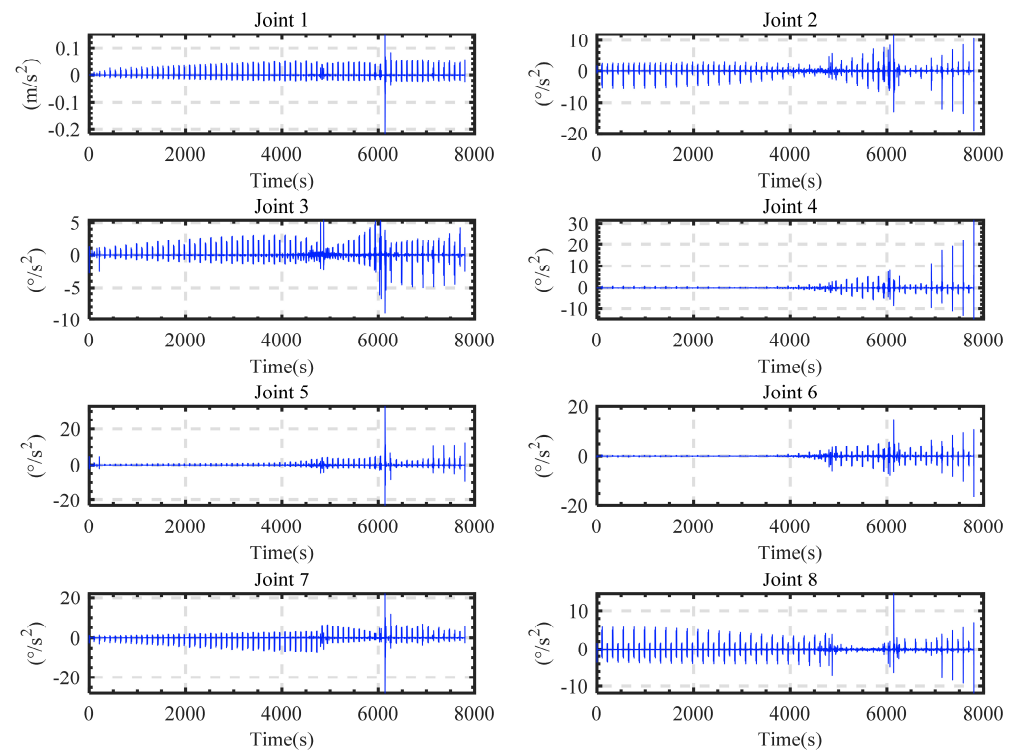


Figure 12. The acceleration of each joint of the trajectory planned by the improved trajectory strategy.

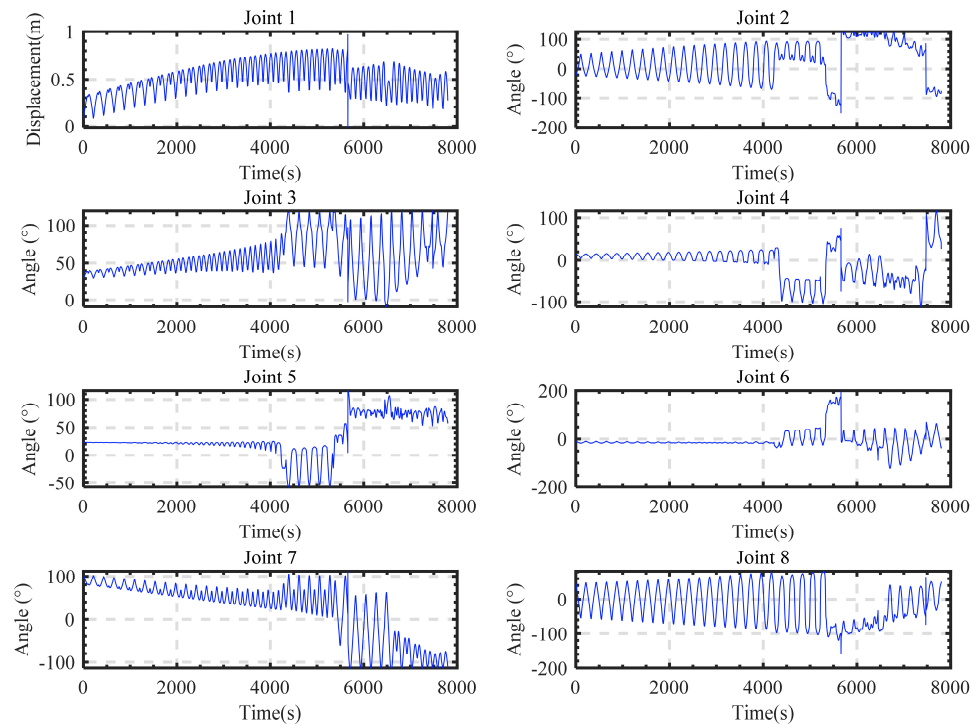


Figure 13. The angle of each joint of the trajectory planned by the general trajectory planning method.

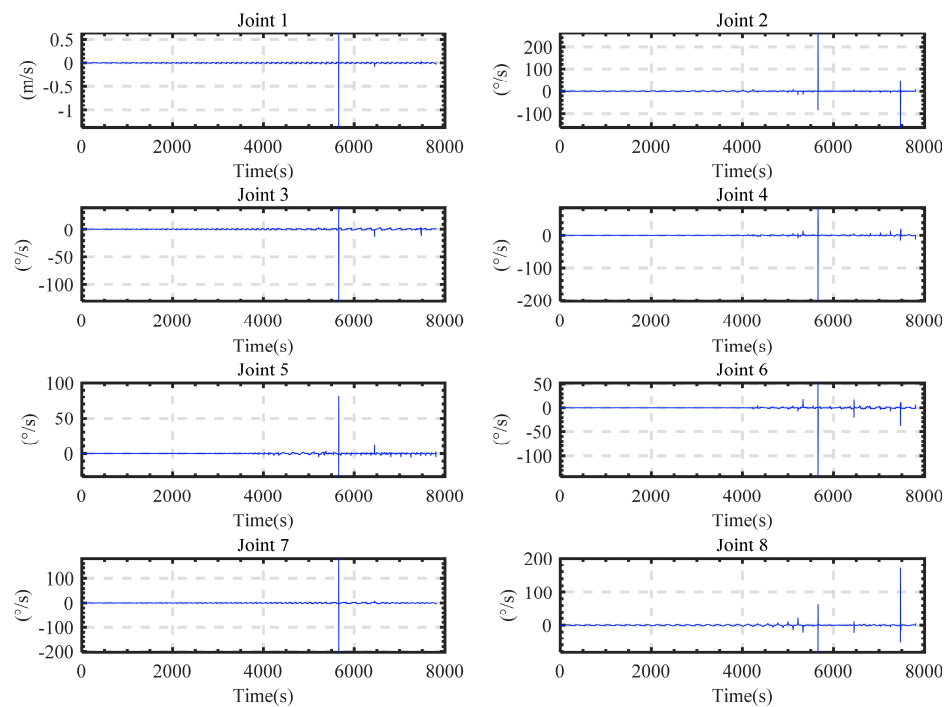
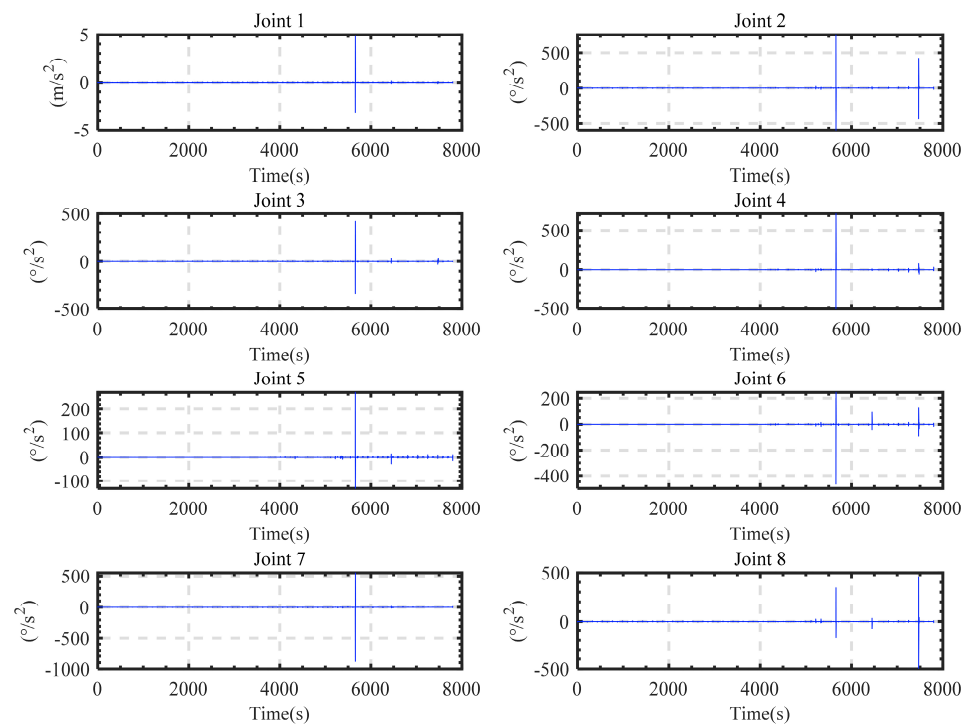
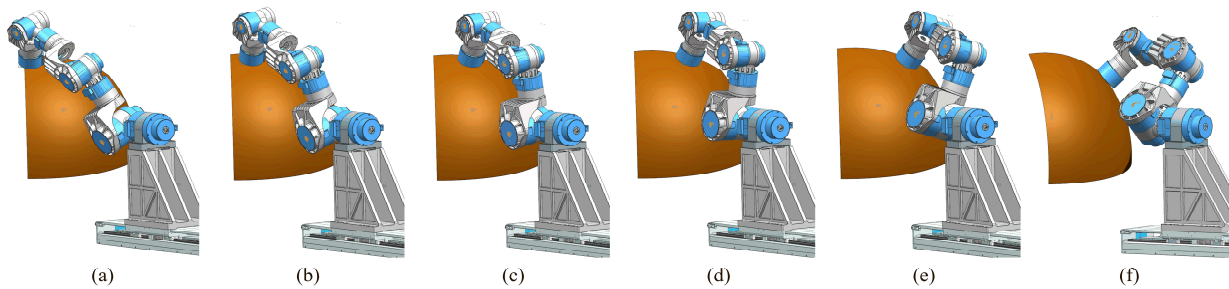


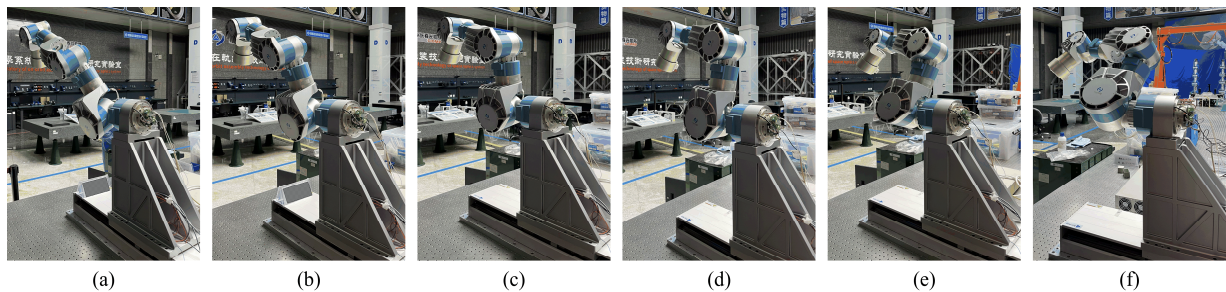
Figure 14. The velocity of each joint of the trajectory planned by the general trajectory planning method.



**Figure 15.** The acceleration of each joint of the trajectory planned by the general trajectory planning method.



**Figure 16.** Simulation of manipulator motion along a section of trajectory. (a) is the starting point of the trajectory, (b–e) is the equally spaced midpoint of the trajectory, and (f) is the end point of the trajectory.



**Figure 17.** The actual process of manipulator motion along a section of trajectory. (a) is the starting point of the trajectory, (b–e) is the equally spaced midpoint of the trajectory, and (f) is the end point of the trajectory.

## 5. Discussion

In this paper, a redundant manipulator suitable for machining large curvature and large convex surfaces has been investigated, and the following conclusions are made:

(1) An eight-degree-of-freedom redundant manipulator is designed, and an improved multi-objective particle swarm optimization algorithm is used to optimize the configuration to obtain a suitable configuration.

(2) A trajectory planning strategy that combines the pincer-weighted minimum parametric method and the projected gradient method is proposed. The trajectory planning strategy is able to plan smooth machining trajectories for large-area curved surfaces.

(3) The simulation shows that the configuration of the eight-degree-of-freedom redundant manipulator can satisfy the requirements of large-area curved surface machining. Additionally, the trajectory planned by the improved strategy is smoother, and the experimental results prove the feasibility of the trajectory planning strategy proposed in this paper.

The optimization methods and trajectory strategies proposed in this paper for the large-surface machining manipulator still have some limitations, and the following aspects can be investigated in depth in future work.

The configuration optimization process requires human judgment to determine a more accurate optimization range. Therefore, future research can be directed toward simplifying optimization methods and reducing human involvement.

The trajectory planning takes longer because of the reverse planning step. Therefore, future research can be directed toward investigating more efficient singularity avoidance methods to improve the efficiency of trajectory planning.

Although there is no mention of the machining of complex surfaces, the machining manipulator proposed in this paper is capable of machining complex surfaces. To machine complex surfaces, more detailed planning of the machining path is required. If we want to deal with different cases of workpieces, we can introduce visual inspection to determine the condition of the surface and then perform targeted trajectory planning, which is also an important future research direction [35].

Only the workspace and manipulator size were considered when optimizing the manipulator configuration. In the future, more performance parameters can be considered to make the manipulator perform better, while Boosting ensemble can be used as a reference to improve the speed and quality of optimization [36].

In addition, the proposed manipulator can be considered for the preparation and processing of novel materials, such as 3D-graphene and quantum dots [37,38]. The working space of the manipulator is large enough, and the planned trajectory is smooth, so the motion accuracy of the manipulator can be further improved to make it more versatile in the future.

**Author Contributions:** Conceptualization, X.S., S.H. and Z.X.; Methodology, X.S., S.H. and E.Z.; Software, X.S. and Y.L.; Validation, X.S., E.Z. and Y.L.; Writing—original draft, X.S.; Writing—review & editing, S.H. and Z.X. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China, grant number 11972343 and 62235018.

**Data Availability Statement:** The datasets generated and analysed during the current study are available from the corresponding author on reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Solanes, J.E.; Gracia, L.; Muñoz-Benavent, P.; Esparza, A.; Miro, J.V.; Tornero, J. Adaptive robust control and admittance control for contact-driven robotic surface conditioning. *Robot. Comput.-Integr. Manuf.* **2018**, *54*, 115–132. [[CrossRef](#)]
2. Gracia, L.; Solanes, J.E.; Muñoz-Benavent, P.; Miro, J.V.; Perez-Vidal, C.; Tornero, J. Adaptive Sliding Mode Control for Robotic Surface Treatment Using Force Feedback. *Mechatronics* **2018**, *52*, 102–118. [[CrossRef](#)]

3. Walker, D.; Dunn, C.; Yu, G.Y.; Bibby, M.; Zheng, X.; Wu, H.Y.; Li, H.; Lu, C. The role of robotics in computer controlled polishing of large and small optics. *Optical Manufacturing and Testing XI. Int. Soc. Opt. Photonics* **2015**, *9575*, 50–58.
4. Liu, H.T.; Wan, Y.J.; Zeng, Z.G.; Xu, L.C.; Fang, K. Freeform surface grinding and polishing by CCOS based on industrial robot. 8th International Symposium on Advanced Optical Manufacturing and Testing Technologies: Advanced Optical Manufacturing Technologies. *SPIE* **2016**, *9683*, 587–593.
5. Klein, A. CAD-based off-line programming of painting robots. *Robotica* **1987**, *5*, 267–271. [[CrossRef](#)]
6. Antonio, J.K. Optimal trajectory planning for spray coating. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation, San Diego, CA, USA, 8–13 May 1994; pp. 2570–2577.
7. Hertling, P.; Hog, L.; Larsen, R. Task curve planning for painting robots. I. Process modeling and calibration. *IEEE Trans. Robot. Autom.* **1996**, *12*, 324–330. [[CrossRef](#)]
8. Omodei, A.; Legnani, G.; Adamini, R. Three methodologies for the calibration of industrial manipulators: Experimental results on a SCARA robot. *J. Robot. Syst.* **2000**, *17*, 291–307. [[CrossRef](#)]
9. Nubiola, A.; Bonev, I.A. Absolute calibration of an ABB IRB 1600 robot using a laser tracker. *Robot. Comput.-Integr. Manuf.* **2013**, *29*, 236–245. [[CrossRef](#)]
10. Hasirden; Zeng, Z.; Liu, H.; Zhao, H. Measurement and analysis on positioning accuracy for optical processing robots. *Opto-Electron. Eng.* **2017**, *44*, 558.
11. Fang, H.R.; Zhu, T.; Zhang, H.Q.; Yang, H.; Jiang, B.S. Design and analysis of a novel hybrid processing robot mechanism. *Int. J. Autom. Comput.* **2020**, *17*, 403–416. [[CrossRef](#)]
12. Gao, Z.; Zhang, D. Performance analysis, mapping, and multiobjective optimization of a hybrid robotic machine tool. *IEEE Trans. Ind. Electron.* **2014**, *62*, 423–433. [[CrossRef](#)]
13. Gao, F.; Peng, B.; Zhao, H.; Li, W. A novel 5-DOF fully parallel kinematic machine tool. *Int. J. Adv. Manuf. Technol.* **2006**, *31*, 201–207. [[CrossRef](#)]
14. Kucuk, S. Dexterous workspace optimization for a new hybrid parallel robot manipulator. *J. Mech. Robot.* **2018**, *10*, 064503. [[CrossRef](#)]
15. Tanev, T.K. Kinematics of a hybrid (parallel–serial) robot manipulator. *Mech. Mach. Theory* **2000**, *35*, 1183–1196. [[CrossRef](#)]
16. Coppola, G.; Zhang, D.; Liu, K. A 6-DOF reconfigurable hybrid parallel manipulator. *Robot. Comput.-Integr. Manuf.* **2014**, *30*, 99–106. [[CrossRef](#)]
17. Fang, Y.; Hu, J.; Liu, W.; Shao, Q.; Qi, J.; Peng, Y. Smooth and time-optimal S-curve trajectory planning for automated robots and machines. *Mech. Mach. Theory* **2019**, *137*, 127–153. [[CrossRef](#)]
18. Božek, P.; Trnka, K. Path planning with motion optimization for car body-in-white industrial robot applications. *Adv. Mater. Res.* **2013**, *605*, 1595–1599. [[CrossRef](#)]
19. Chembuly, V.S.; Voruganti, H.K. Trajectory planning of redundant manipulators moving along constrained path and avoiding obstacles. *Procedia Comput. Sci.* **2018**, *133*, 627–634. [[CrossRef](#)]
20. Zhang, P.; Gong, J.; Zeng, Y.; Li, C. Optimizing Trajectory of Painting Robot’s Spray Gun for Large Curvature Surface. *Mechanical Sci. Technol. Aerospace Eng.* **2015**, *34*, 1670–1674.
21. Zhang, P.; Gong, J.; Ning, H.; Zeng, Y.; Liu, Y.; Wei, L. Study on Trajectory Combination and Connection Problems of Spray-painting Robot for Large Curvature Combination Surfaces. *J. Sichuan Univ. (Eng. Sci. Ed.)* **2016**, *48*, 217–222. [[CrossRef](#)]
22. Shao, J.Y.; Zhang, C.Q.; Chen, Y.; Chen, K. Trajectory planning for redundant robots for internal surface spraying. *J. Tsinghua Univ.* **2014**, *54*, 799–804.
23. Lamikiz, A.; López de Lacalle, L.N.; Ocerin, O.; Díez, D.; Maidagan, E. The Denavit and Hartenberg approach applied to evaluate the consequences in the tool tip position of geometrical errors in five-axis milling centres. *Int. J. Adv. Manuf. Technol.* **2008**, *37*, 122–139. [[CrossRef](#)]
24. Katoch, S.; Chauhan, S.S.; Kumar, V. A review on genetic algorithm: Past, present, and future. *Multimed. Tools Appl.* **2021**, *80*, 8091–8126. [[CrossRef](#)] [[PubMed](#)]
25. Dorigo, M.; Stützle, T. Ant Colony Optimization: Overview and Recent Advances. In *Handbook of Metaheuristics*; Springer Nature: Basel, Switzerland, 2019; pp. 311–351.
26. Ding, S.; Su, C.; Yu, J. An optimizing BP neural network algorithm based on genetic algorithm. *Artif. Intell. Rev.* **2011**, *36*, 153–162. [[CrossRef](#)]
27. Wang, D.; Tan, D.; Liu, L. Particle swarm optimization algorithm: An overview. *Soft Comput.* **2018**, *22*, 387–408. [[CrossRef](#)]
28. Bai, Q. Analysis of particle swarm optimization algorithm. *Computer Inform. Sci.* **2010**, *3*, 180. [[CrossRef](#)]
29. Jiang, Y.; Hu, T.S.; Huang, C.; Wu, X.N. An improved particle swarm optimization algorithm. *Appl. Math. Comput.* **2007**, *193*, 231–239. [[CrossRef](#)]
30. Chen, C.Y.; Ye, F. Particle swarm optimization algorithm and its application to clustering analysis. In Proceedings of the 2012 Proceedings of 17th Conference on Electrical Power Distribution, Tehran, Iran, 2–3 May 2012; pp. 789–794.
31. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **2004**, *8*, 256–279. [[CrossRef](#)]
32. Chan, T.F.; Dubey, R.V. A weighted least-norm solution based scheme for avoiding joint limits for redundant joint manipulators. *IEEE Trans. Robot. Autom.* **1995**, *11*, 286–292. [[CrossRef](#)]



33. Phuoc, L.M.; Martinet, P.; Lee, S.; Kim, H. Damped least square based genetic algorithm with Gaussian distribution of damping factor for singularity-robust inverse kinematics. *J. Mech. Sci. Technol.* **2008**, *22*, 1330–1338. [[CrossRef](#)]
34. Huang, S.H.; Peng, Y.G.; Wei, W.; Xiang, J. Clamping weighted least-norm method for the manipulator kinematic control with constraints. *Int. J. Control* **2016**, *89*, 2240–2249. [[CrossRef](#)]
35. Rodríguez, A.; González, M.; Pereira, O.; López de Lacalle, L.N.; Esparta, M. Edge finishing of large turbine casings using defined multi-edge and abrasive tools in automated cells. *Int. J. Adv. Manuf. Technol.* **2021**, *124*, 3149–3159. [[CrossRef](#)]
36. Bustillo, A.; Urbikain, G.; Perez, J.M.; Pereira, O.M.; López de Lacalle, L.N. Smart optimization of a friction-drilling process based on boosting ensembles. *J. Manuf. Syst.* **2018**, *48*, 108–121. [[CrossRef](#)]
37. He, Z.; Zhang, S.; Zheng, L.; Liu, Z.; Zhang, G.; Wu, H.; Wang, G. Si-Based NIR Tunneling Heterojunction Photodetector With Interfacial Engineering and 3D-Graphene Integration. *IEEE Electron Device Lett.* **2022**, *43*, 1818–1821. [[CrossRef](#)]
38. Zhou, W.; Zheng, L.; Ning, Z.; Cheng, X.; Wang, F.; Xu, K.; Yu, Y. Silicon: Quantum dot photovoltage triodes. *Nat. Commun.* **2021**, *12*, 6696. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.