electronics

Article

Optimal Trajectory Planning for Manipulators with Efficiency and Smoothness Constraint

Zequan Xu [1,2], Wei Wang [1,2,3,*], Yixiang Chi [1,2], Kun Li [1,2] and Leiying He [1,2,*]

1 School of Mechanical Engineering, Zhejiang Sci-Tech University, Hangzhou 310018, China
2 Key Laboratory of Modern Textile Equipment Technology of Zhejiang Province, Zhejiang Sci-Tech University, Hangzhou 310018, China
3 State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou 310027, China
* Correspondence: wangw@zstu.edu.cn (W.W.); hlying@zstu.edu.cn (L.H.)

Abstract: Path planning to generate an appropriate time sequence of positions for a complex trajectory is an open challenge in robotics. This paper proposes an optimization method with the integration of an improved ant colony algorithm and a high-order spline interpolation technique. The optimization process can be modelled as the travelling salesman problem. The greatest features of this method include: (1) automatic generation for complex trajectory and a new idea of selecting the nearest start point instead of using the traditional way of human operation; (2) an optimized motion sequence of the manipulator with the shortest length of the free-load path improves efficiency by nearly 65% and (3) trajectories both in Cartesian space and joint space are interpolated with good smoothness to reduce shocks and vibrations. Simulations and experiments are conducted to demonstrate the good properties of this method.

Keywords: manipulator; path planning; ant colony algorithm; spline interpolation

check for updates

Citation: Xu, Z.; Wang, W.; Chi, Y.; Li, K.; He, L. Optimal Trajectory Planning for Manipulators with Efficiency and Smoothness Constraint. *Electronics* **2023**, *12*, 2928. https://doi.org/10.3390/electronics12132928

Academic Editors: Allahyar Montazeri, Zhou Zhang and Yizhe Chang

Received: 8 May 2023
Revised: 18 June 2023
Accepted: 29 June 2023
Published: 3 July 2023

1. Introduction

Manipulators are widely used in industry because of their outstanding characteristics of high efficiency and easy operation, especially for complicated tasks with demand for repetition [1]. Appropriate path planning is the foundation of efficient and stable movements of manipulators, which is also one of the significant issues in robotics. Therefore, path planning can be applied in various kinds of fields, such as the automobile [2,3], aerospace [4,5] and agriculture industries [6].

Path planning can be defined as constructing a planner to figure out an optimal trajectory with consideration of the limitations originating from the required geometric path, kinematics and dynamics [7]. Most of the open literature about path planning involves the construction of an objective function that aims to minimize the execution time, actuator torque, energy consumption, or a combination of those variables [8]. Obstacle avoidance in a complex dynamic environment is also widely necessary [9].

In order to achieve those goals, a lot of methods have been developed; heuristic algorithms are popular and commonly used [10], which are inspired by certain species of insects or animals in nature. These methods include the differential evolution algorithm (DE), genetic algorithm (GA), particle swarm optimization (PSO), ant colony algorithm (ACO), simulated annealing algorithm (SA), etc. Some others are based on probability theory, such as the A* algorithm, probabilistic roadmap method (PRM) and rapidly exploring random tree (RRT). In some special situations, particular methods exhibit much better suitability, such as Pontryagin's method applied in spacecraft slew maneuvers [11] and the hybrid S function method used in material manufacturing [12]. Each method has its own unique capability in searching for solutions with varying speed in convergence and optimization.

As for manipulator path planning and control, we are usually concerned with two types of cases because of its special open-loop structure. The motion of revolute joints can be transmitted from one link to another and finally impact the position of end-effectors of the manipulator. The path calculation for the end-effector to move along can be completed in Cartesian space, while motor angles are computed in joint space. Some research studies about path planning of manipulators are focused on either Cartesian space or joint space, in which different optimization methods have been proposed to generate an optimal trajectory with consideration of the lowest consumption of time, energy or actuator effort [3] etc.

In a study of the obtainment of a sequence of positions in Cartesian space, Wang [13] developed an improved ant colony algorithm (ACO) to find a less time-consuming path for ship component manufacturing. Aiming to finish the welding tasks of general stiffened plates at an arbitrary position on the workbench in robot workspace, Zhou [14] presented the improved lazy Probabilistic Roadmap (PRM) algorithm. Using a similar idea, Chen [15] handled the problem of the manipulator potentially being unable to work in a situation where the free space contains narrow passages. Aiming to boost the diversity of the generated solutions, Elhoseny [16] developed an improved genetics algorithm for a dynamic field. Wang [17] presented the matrix alignment Dijkstra method to reduce computational costs and to obtain an acceptable path; the novel feature lies in that dynamic restrictions were regarded as static ones.

In Cartesian space, each path point must be restricted and computed, while only the start and end points should be considered when planning in joint space. Yu [18] proposed the strategy by looking ahead synchronously to guarantee the continuous movement of joints as well as computational efficiency. Liang [19] generated the optimized path for the multi-axis system based on the constraints of maximum allowable velocity and acceleration of all the joints. Chettibi [20] proposed a radial basis functions (RBF)-based method, however, the research is limited in that the only focus is on the joint space of the manipulator. Reiter [21] derived the higher order inverse kinematics of a redundant manipulators, and then developed the time-optimal motion control method. Kim [22] defined the normalized step cost to improve the efficiency of initialization of the particles, and then applied the particle swarm optimization (PSO) method to handle the problem of path planning.

The execution time of the manipulator is the most important consideration since it is tightly linked to the demand for improving productivity in industrial fields [23,24]. Unfortunately, the time-optimal strategy alone cannot ensure the sufficient continuity of the motion trajectories. Limiting jerking motion is quite important to reduce vibration and the wearing out of the joint [25]. Furthermore, low-jerk trajectories can be executed with higher accuracy, so some efforts have been devoted into this area. Jahanpour [26] applied parametric non-uniform rational basis spline (NURBS) curves to feature serial discrete positions in the joint space of a parallel machine. Energy is another objective to be optimized since it is related to the torque and jerking of the joint. Chen [27] proposed a Hamiltonian strategy; however, the minimum energy consumption cannot ensure the optimal jerk sequence. In addition to the energy used, efficiency can be integrated to be another constraint; thus, the path planning can be equivalent to a multi-object optimization problem [28].

As can be seen from the research mentioned above, much progress in path planning in robots has been made since quite a lot of studies are focused on this issue. However, there are some other aspects that can be still further improved, which include:

(1) Lack of sufficient analysis of the complex trajectory in Cartesian space. The traditional way to generate the path for the manipulator when trying to manufacture parts with complex shapes is often to select successive discrete points randomly by the operator, and then, as adjacent points are interpolated automatically with the straight line or some other polynomials, the process can be modelled as a point-to-point path-planning problem. Insufficient analysis of the process to generate the best sequence for those discrete paths makes the motion far from efficient.

(2) No assurance of good smoothness for the whole trajectory in joint space. Optimization in reducing consumption of time or energy of the manipulator cannot ensure the continuous movement of all the joints alone. Existing research about welding, transporting and polishing has failed to consider further planning in joint space even though they managed to find an optimized sequence of the discrete position in Cartesian coordinates. As a result, the velocity, acceleration and jerk of joints are rarely smooth and are very likely to exceed the limitations, the motion is unstable and unsafe, and the accuracy of the manipulator is also unlikely to meet the demand due to the shock and vibration.

The aim of this paper is to develop an optimization method to generate an appropriate sequence of positions for a complex trajectory constituted by a large number of discontinuous paths. An improved ant colony algorithm (ACO) and B-splines interpolation are both used to guarantee the efficiency and smoothness of the movement. The greatest contributions of this work mainly include three aspects: (1) the novel ideal of selection of start points for a large number of discontinuous paths, (2) the much higher efficiency is realized based on the better motion sequence with the shortest length that is automatically generated by an optimization process and (3) the outstanding properties of continuity and smoothness for the trajectory because of the high degrees of polynomials and B-splines.

The remainder of this paper is organized as follows: Section 2 describes the process followed to generate an optimized sequence of the quantity of discontinuous paths. Section 3 mainly focuses on the kinematic model of the serial manipulator and the interpolation algorithm, both in Cartesian space and joint space. Numerical simulation and some interesting experiments are conducted in Section 4, followed by discussion in Section 5. Conclusions are summarized in the last section.

2. Generation of the Time-Optimal Sequence for the Complex Discontinuous Trajectory

2.1. Analysis of a Large Quantity of Discontinuous Scattered Paths

Complex parts often have a large number of discontinuous paths enclosed individually within themselves, and usually require the manipulator to jump frequently from one path to another while standing by. Traditional methods to construct the trajectory by use of the teach pendant in an on-line or off-line way are difficult and rather time consuming. In this work, the complex trajectory is decomposed and automatically generated by extraction from a 2D model of the figures; all the important features and the corresponding discrete path points are obtained by use of OpenCV. the geometric center of each single path is marked and treated like scattered cities. Thus, the manipulator's movement along the whole trajectory with the shortest distance can be modelled as a travelling salesman problem (TSP). The schematic for the travelling process is shown in Figure 1.

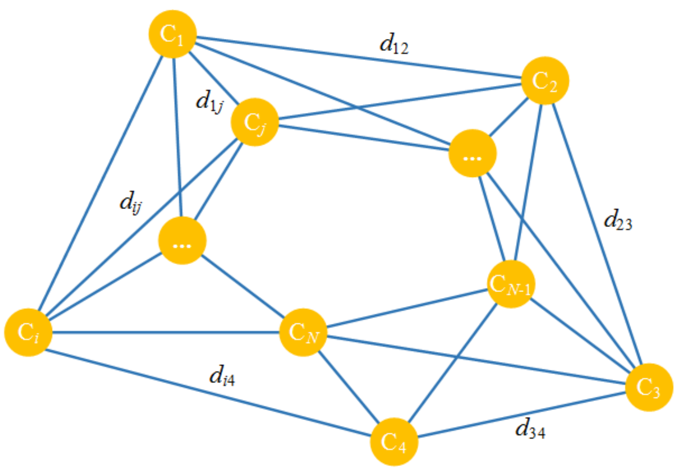


Figure 1. Travelling process of the manipulator.

The total distance of travel and average speed are the two main factors that significantly influence the efficiency of the movement. In particular, the optimal motion sequence of all the discontinuous paths means the shortest free-load path, which can be arbitrary and must be figured out through the optimization algorithm. The improved ant colony algorithm (ACO) used in this work is famous and quite suitable for solving the TSP because of its good characteristics in high speed of convergence and strong capability in finding the global optimal solution.

Suppose there are N cities (discontinuous paths) and the set consisting of those points is denoted as $\boldsymbol{C} = \{c_1, c_2, \cdots, c_N\}$, the distance between the ith and jth point is $d_{ij} = f(c_i, c_j) \geq 0$, where we have $c_i, c_j \in \boldsymbol{C}(1 \leq i, j \leq N)$. The total length of all the free-load paths when the manipulator is standing by can be computed as:

$$S = \sum_{i=1,j=2}^{N} d_{ij}(i < j) \tag{1}$$

2.2. Generation of Optimal Motion Sequence Based on Improved Ant Colony Algorithm (ACO)

Assume that the scale of the ant used for searching is m, $b_i(t)$ is the number of ants passing through the city c_i at time t. $\tau_{ij}(t)$ represents the intensity measure of the pheromone deposited by each ant on the path from $c_i \rightarrow c_j$. The set $\mathbf{L}$ denotes the conjunction paths between all the discrete cities as $\mathbf{L} = \{l_{ij} | c_i, c_j \in C\}$. Obviously, we can have $m = \sum_{i=1}^{N} b_i(t)$. Make set $\mathbf{T} = \{\tau_{ij}(t) | c_i, c_j \in C\}$ represent the deposited pheromone on the path l_{ij}.

At the initial time t_0, all the ants are placed in m of N cities randomly, the intensity of pheromone on each path is equal and $\tau_{ij}(t_0) = Q$; the optimization process can be completed in the digraphs set $\mathbf{G} = (\mathbf{C},\mathbf{L},\mathbf{T})$ within the high dimensional space.

According to the principle of the algorithm, the next city that the kth $(k = 1, 2, \cdots, m)$ ant would go to can be decided based on the intensity of the pheromone; the higher the concentration, the greater the probability will be. In order to ensure each path can be travelled only once by the same kth ant, the forbidden table tab_k is established. Hence, the probability that the kth ant moves from the current city c_i to the next city c_j can be computed as:

$$P_{ij}^{k} = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{s \in A_k} [\tau_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} & , c_j \in A_k \\ 0 & , c_j \notin A_k \end{cases} \tag{2}$$

where $A_k = \{C - \text{tab}_k\}$ represents all the optional next cities for the kth ant, parameters α and β are positive constant values that denote the information heuristic factor and the heuristic factor respectively; the larger value of α, the greater the probability that an ant will choose the path passed before by other ants; the larger the value of β, the faster that the searching process can converge; $\eta_{ij}(t)$ is the heuristic function that measures the intent of the kth ant travelling from c_i to c_j.

The formulation of $\eta_{ij}(t)$ can be written as:

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \tag{3}$$

The pheromone will be updated as long as the ant finishes the whole journey, the remaining pheromone at time $(t + n)$ on the path l_{ij} can be re-computed by the rule written as follows:

$$\tau_{ij}(t + n) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij}(t) \tag{4}$$

where ρ denotes the volatility coefficient,$(1 - \rho)$ represents the remaining coefficient and ρ has the range of [0, 1]; $\Delta\tau_{ij}(t)$ is the total increased pheromone.

The changing pheromone can be computed by

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{m} \delta\tau_{ij}^{k}(t) \tag{5}$$

where $\delta\tau_{ij}^{k}(t)$ represents the pheromone produced by the kth ant when travelling on the path l_{ij}.

There are three approaches to compute $\delta\tau_{ij}^{k}(t)$ in the existing research, and the corresponding models can be named as the ant-quantity system, ant-density system and ant-cycle system. The computation in the first two types is based on the information of a local path while the last one updates the value with global information. Therefore, the ant-cycle system adopted in this work can overcome the limitation of a local optimal solution; the function can be formulated as

$$\delta\tau_{ij}^{k}(t) = \begin{cases} \frac{Q}{L_k}, & tour(i,j) \in l_{ij} \\ 0, & others \end{cases} \tag{6}$$

where Q is a positive constant value and denotes the initial intensity of pheromone; L_k is the total distance that the kth ant travels to completes the journey after travelling all the cities $tour(i,j)$ represents the current path in which the kth ant is located.

The approach used to generate the optimal sequence of the complex trajectory can be summarized as follows.

Step 1: obtain the contours (discontinuous paths) to construct the complex trajectory by extracting information from the figures. Compute the geometric centers of the N discontinuous paths, treat them as discrete cities, and calculate the distance d_{ij} between any two cities.

Step 2: initialize the scale of ant m and place it on m of the N cities randomly, along with the starting pheromone $\tau_{ij}(t_0)$, the number of iteration times N_{iter}, the information heuristic factor α, β and the volatility coefficient ρ.

Step 3: compute P_{ij}^{k} with use of Equations (2)–(6) and the roulette rule to estimate the probability of the kth ant moving from the current city c_i to the next city c_j.

Step 4: complete a single searching process when k increases from 1 to m and i varies from 1 to N step by step. The total distance after all m ants have travelled via N cities is the result. The path length of each ant is a feasible solution.

Step 5: according the set of solutions for m ants, proceed with the selection of the minimum path in the current iteration $L_{min} = \min\{L_1,\ L_2, \cdots, L_m\}$ and choose the smaller one after the comparison between L_{min} and the last optimal solution $\mathrm{L}\prime_{min}$.

Step 6: repeat steps (3)~(5) until all the iterations have been conducted.

The process to find the optimal path with the shortest length among all the possible ways is shown in Figure 2.

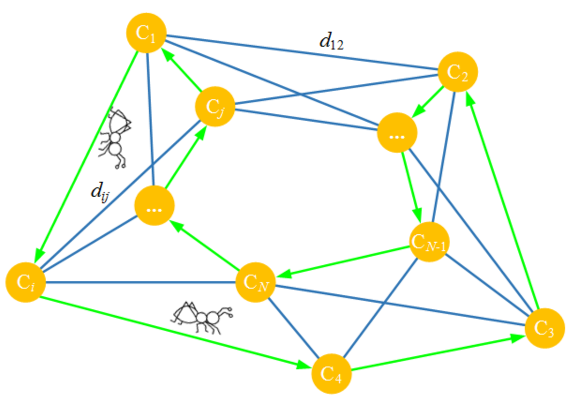


Figure 2. The searching process of ACO.

2.3. Brief Introduction of Previous Work

In order to conduct a comparative study, two popular methods, genetic algorithm [16] (GA) and particle swarm optimization algorithm [22] (PSO) are introduced. They are both heuristic methods which are the same as the improved ant colony algorithm (ACO).

In terms of the genetic algorithm, the optimization process can be summarized as follows.

(1) Initialize population and calculate fitness function: Calculate the value of the fitness function. The fitness function measures the degree of individual superiority and inferiority, which can be represented as

$$f(x_i) = \sum_{j=1}^{n-1} distance\left(c_{i,j}, c_{i,j+1}\right) \tag{7}$$

where $f(x_i)$ is the fitness function value of the ith individual, n is the scale of the individuals, $c_{i,j}$ is the jth city that the ith individual passes through, while $distance\left(c_{i,j}, c_{i,j+1}\right)$ is the distance between $c_{i,j}$ *and* $c_{i,j+1}$.

(2) Selection and cross operation: select some individuals as parents for breeding the next generation in which the roulette wheel selection method is used; the equation is

$$P_i = \frac{f(x_i)}{\sum_{j=1}^{m} f\left(x_j\right)} \tag{8}$$

where P_i is the probability that individual i is selected, $f(x_i)$ is the fitness function value, and m is the number of individuals in the population.

A sequential crossover operation is used; the process can be written as

$$\begin{cases} s_1 = x_1[1:k] + x_2[k+1:n] \\ s_2 = x_2[1:k] + x_1[k+1:n] \end{cases} \tag{9}$$

where s_1 and s_2 are the two new individuals after crossing, x_1 and x_2 are the two individuals before crossing, and n is the length of the individual.

(3) Mutation operation and update population: To mutate offspring individuals and introduce randomness to increase the breadth of search space. Merge parent and offspring individuals to form a new population.
(4) Elite retention: Select a portion of optimal individuals from a new population and keep them for the next generation, ensuring that the optimal solutions in the population are retained.

In a similar way, differences between the above-mentioned GA and particle swarm optimization algorithm (PSO) include the local optimization solution strategy, which can be formulated as

$$\begin{gathered} v(j,:) = w * v(j,:) + c1 * rand(1,d). * (pbest(j,:) - x(j,:)) + c2 \\ *rand(1,d). * (gbest - x(j,:)) \end{gathered} \tag{10}$$

where w is the inertia weight, $c1$ and $c2$ are the acceleration constants, $rand$ is the random function, $pbest$ and $gbest$ are local and global extreme values.

Repeat the update process until the termination condition is satisfied. Afterwards, the optimal solutions can be obtained by those two methods.

2.4. Selection of Starting Points for the Series of Discontinuous Paths

The optimal sequence for all the discontinuous paths can be figured out by use of ACO since the geometric centers are regarded as discrete cities. For each individual enclosed path, we proposed the nearest-point-based selection principle to determine the starting position for the manipulator. The process is summarized as follows:

(1) Generate the smallest-circumcircle of each discontinuous path with its geometric center, as shown in Figure 3a; (2) Determine the outermost circumcircle and select the point with the most intersections between the line along the radial direction and internal closed paths as the starting point for the entire motion, and (3) For each circumcircle in order of decreasing size, calculate the shortest distance between adjacent contours in sequence. The point with the shortest distance will serve as the starting point for the next local contour; all the starting points can be generated by the same way. For example, as shown in Figure 3b, the optimal sequence of starting points is $c_4 \to c_3 \to c_2 \to c_1$.

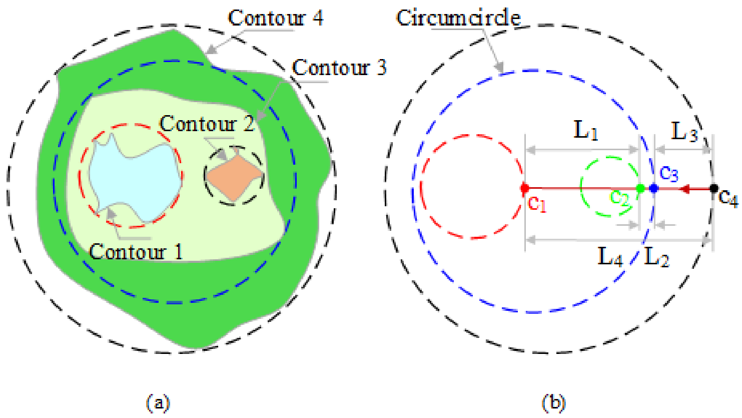


Figure 3. Schematic diagram for the starting point selection: (**a**) circumcircle of the discontinuous paths; (**b**) the best starting point of each path.

3. Jerk-Continuous and Smooth Trajectory Planning

The above part mainly focuses on the problem of optimization of the sequence of the large quantity of discontinuous trajectories. Another important issue is to guarantee the smoothness of motion of the manipulator. The typical serial manipulator usually consists of several linkages connected by rotating joints, the motion and force can be transmitted efficiently from the base frame to the tool frame.

3.1. Kinematics of the Manipulator

The local frame attached on the link is constructed to describe the relative position between two adjacent joints of the serial manipulator, which can be also called the D-H coordinate system, as shown in Figure 4.

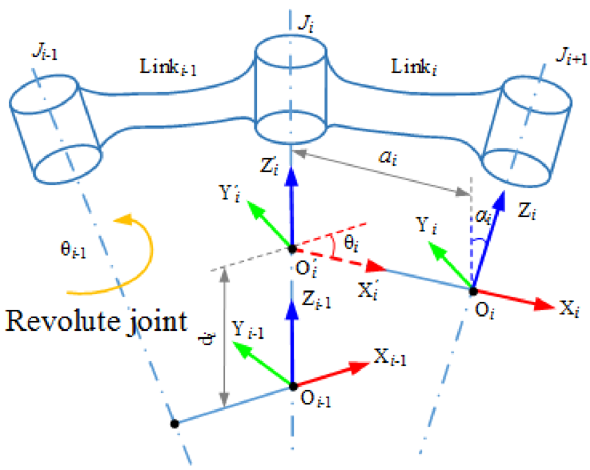


Figure 4. Transformation between adjacent joints.

The transformation can be formulated by a homogeneous matrix, derived by [29]:

$$A_i^{i-1} = \begin{bmatrix} Cos\theta_i & -Sin\theta_i Cos\alpha_i & Sin\theta_i Sin\alpha_i & a_i Cos\theta_i \\ Sin\theta_i & Cos\theta_i Cos\alpha_i & -Cos\theta_i Sin\alpha_i & a_i Sin\theta_i \\ 0 & Sin\alpha_i & Cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{11}$$

where (a_i, d_i, α_i, θ_i) are the D-H parameters of the ith link.

In this work, a manipulator with six degrees of freedom is applied to construct the kinematic model, as shown in Figure 5a.

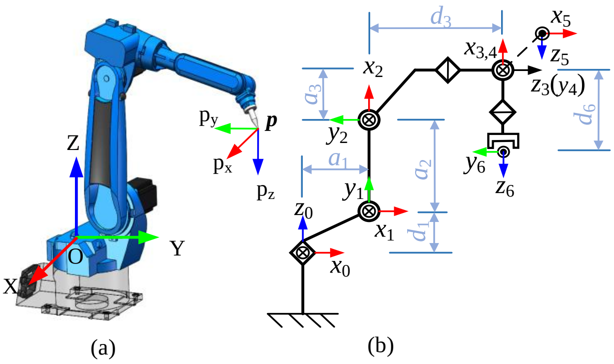


Figure 5. A schematic diagram of a serial manipulator: (**a**) 3D model of a typical manipulator; (**b**) D-H coordinates of the manipulator.

The position and orientation of the end-effector can be obtained through a product of a series of homogeneous matrices, according to the local coordinate system drawn in Figure 5b, which is formulated as:

$$T = \prod_{i=1}^{6} A_i^{i-1} = \begin{bmatrix} \boldsymbol{Rot} & \boldsymbol{Pos} \\ 0 & 1 \end{bmatrix} \tag{12}$$

where $\boldsymbol{Rot}$ denotes the orientation matrix and $\boldsymbol{Pos} = [p_x, p_y, p_z]$ represents positions.

The kinematics is studied based on a 6-DOFs manipulator, with the corresponding structure parameters listed in Table 1.

Table 1. The standard D-H parameters.

No.	a_i(mm)	d_i(mm)	α_i(°)	θ_i(°)	Initial Angle (°)
1	80	210	90	θ_1	0
2	275	0	0	θ_2	90
3	65	310	90	θ_3	0
4	0	0	−90	θ_4	0
5	0	0	90	θ_5	−90
6	0	54.5	0	θ_6	90

3.2. Trajectory Planning in Cartesian Space with Polynomial Interpolation Method

Given a set of discrete interpolation samples: $y_i = f(t_i)$, where we have $i = 0, 1, \cdots, n$ and $a = t_0 < t_1 < t_2 < \cdots < t_n = b$, if there exists $s(t)$ with n piecewise polynomials as

$$s(t) = \begin{cases} s_1(t) & t \in [t_0, t_1] \\ s_2(t) & t \in [t_1, t_2] \\ & \vdots \\ s_n(t) & t \in [t_{n-1}, t_n] \end{cases} \tag{13}$$

And each has the expression written as

$$s_j(t) = \lambda_{k,j}t^k + \lambda_{k-1,j}t^{k-1} \cdots + \lambda_{1,j}t^1 + \lambda_{0,j}\ j = 1, 2, \cdots, n \tag{14}$$

where $\lambda_{k,j}, \lambda_{k-1,j}, \cdots, \lambda_{0,j}$ represent the constant coefficients and k is the degree of polynomial $s_j(t)$ that meets the following requirements of

(i) the naturally interpolated property, $s(t_i) = y_i = f(t_i)$
(ii) the segmented path to join up, $s_j(t_j) = s_{j+1}(t_j)$
(iii) the inherent smooth property requires that the curve be $(k-1)$ times continuous differentiable, $s\prime_j(t_j) = s\prime_{j+1}(t_j), s''_j(t_j) = s''_{j+1}(t_j), \cdots, s_j^{(k-1)}(t_j) = s_{j+1}^{(k-1)}(t_j)$.

Then, $s(t)$ can be a spline interpolating function with a degree of k for the complex trajectory having a series of discrete path points. In mathematics, a spline is a special function and consists of several pieces represented by polynomials. They can exhibit impressive interpolating properties with high accuracy even though the degree is low. Because of the outstanding feature, the spline can avoid the problem of Runge's phenomenon, and they are widely used in practical engineering [14].

In order to enhance the interpolation accuracy and retain the important features of the nonlinear complex trajectory as much as possible, we adopt the cubic spline to hold the trajectory in Cartesian space as the property of a second order continuous differential, which means the acceleration of the end-effector of the manipulator along the coordinate x (or y,z) varies with time t continuously. According to the definition in Equation (9), we can have the expression represented as follows:

$$s_j(t) = \lambda_{3,j}t^3 + \lambda_{2,j}t^2 + \lambda_{1,j}t + \lambda_{0,j}\ j = 1, 2, \cdots, n \tag{15}$$

It is easy to observe that $s''_j(t)$ is a linear function in the closed interval $[t_{j-1}, t_j]$. We can derive the speed of the motion by the first-order derivative of Equation (11), which can be formulated as:

$$s'_j(t) = 3\lambda_{3,j}(t - t_{j-1})^2 + 2\lambda_{2,j}(t - t_{j-1}) + \lambda_{1,j} \tag{16}$$

With similar computation, we can derive the second-order derivative (acceleration) of Equation (11) and we have:

$$s''_j(t) = 6\lambda_{3,j}(t - t_{j-1}) + 2\lambda_{2,j} \tag{17}$$

For simplicity, we denote $h_j = t_j - t_{j-1}$ and $m_j = s''_j(t_j)$. According to the condition (i), it is obvious that $\lambda_{0,j} = s_j(t_j) = y_j$; with the utilization of conditions (ii) and (iii), we can finally derive the following equations that are written as:

$$\begin{cases} \lambda_{3,j} = \frac{m_j - m_{j-1}}{6h_{j-1}} \\ \lambda_{2,j} = \frac{m_{j-1}}{2} \\ \lambda_{1,j} = \frac{y_j - y_{j-1}}{h_{j-1}} - \frac{h_{j-1}}{2}m_{j-1} - \frac{h_{j-1}}{6}(m_j - m_{j-1}) \end{cases} \tag{18}$$

Substituting Equation (14) into Equation (11), we can construct a matrix equation only related to unknown parameters m, which is formulated as:

$$h_{j-1}m_{j-1} + 2(h_j + h_{j-1})m_j + h_j m_{j+1} = 6\varphi_j \tag{19}$$

where φ_j represents $\varphi_j = \left(\frac{y_{j+1} - y_j}{h_j} - \frac{y_j - y_{j-1}}{h_{j-1}}\right)$

There will be $(n-1)$ equations formed by Equation (15) in the entire interval of [a, b]; however, we can note that two more unknown variables of m_j exist, which means that we have $(n + 1)$ unknown variables. We still require more constraint conditions to construct another two equations.

In this work, we apply the natural constraint with consideration that the manipulator can remain in the static state at the start as well as end of the path. We set the acceleration equal to zero, $m_0 = s''(t_0) = 0$, $m_n = s''(t_n) = 0$ and hence Equation (15) can be rewritten in the form of a matrix, formulated as

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ h_1 & 2(h_1+h_2) & h_2 & 0 \\ 0 & h_2 & \ddots & \vdots \\ \vdots & \vdots & \ddots & h_n \\ 0 & 0 & h_{n-1} & 2(h_{n-1}+h_n) \end{bmatrix} \begin{bmatrix} m_0 \\ m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} = 6 \begin{bmatrix} 0 \\ \varphi_1 \\ \vdots \\ \varphi_{n-1} \\ 0 \end{bmatrix} \tag{20}$$

The above matrix equation can be solved by the LU decomposition strategy and the chase after method.

3.3. Trajectory Planning in Joint Space with Multi-Degree B-Splines Interpolation Method

Path planning in Cartesian space is mainly focused on the motion of the manipulator along the specified contour of parts. The stability and efficiency of the manipulator are significantly influenced by the smoothness of the trajectory in joint space. The B-spline with multi-degree condition exhibits good smoothness and local support [14]; thus, it has been widely used in the interpolation of joint movement recently.

Through inverse kinematics transformation, the joint angles with reference to a specified path point in Cartesian space can be computed; the formulation can be written as

$$\boldsymbol{\theta} = f^{-1}(\boldsymbol{T}) \tag{21}$$

where $\boldsymbol{\theta}$ is the vector of sequential joint angles varying with time $t_i(i = 0, 1, \cdots, n)$ and $\boldsymbol{T}$ is the vector of position and posture of the end-effector.

The objective of interpolation in joint space is to generate a sequence of angles with the property of smoothness. Given $n + 1$ control points $C_0, C_1, \cdots, C_n$ and a non-decreasing knot vector, $\boldsymbol{U} = \{u_0, u_1, \cdots, u_m\}$. The corresponding data points $P_0, P_1, \cdots, P_n$ can be solved through the B-spline function. In order to facilitate presentations and to unify the formula expression, the definition domain of u is firstly normalized from the range of [u_0, u_m] into [0, 1] based on the Accumulative Chord Length method; the result can be written as

$$u_i = \begin{cases} 0, & i = 0, 1, \cdots, k \\ \mathrm{u}_{\mathrm{i}-1} + \frac{t_{i-k} - t_{i-k-1}}{t_n - t_0}, & i = k+1, \cdots, n+k-1 \\ 1, & i = n+k, \cdots, n+2k \end{cases} \tag{22}$$

The B−spline curve of degree k defined by those control points and the knot vector is

$$P_i = \sum_{i=0}^{n} N_{i,k}(u) C_i \quad u \in [u_0, u_m] \tag{23}$$

where $N_{i,k}(u)$ are the basic functions of degree k. they can be derived by the Cox–de Boor recursion method [14], and the formulation is

$$N_{i,k}(u) = \frac{u - u_i}{u_{i+k} - u_i} N_{i,k-1}(u) + \frac{u_{i+k+1} - u}{u_{i+k+1} - u_{i+1}} N_{i+1,k-1}(u) \tag{24}$$

where $N_{i,0}(u) = 1$ as long as u is in the range of $[u_i, u_{i+1}]$, otherwise $N_{i,0}(u) = 0$.

To guarantee the start as well as the end point of the B-spline is coincident with the that of control points (also called clamped), the number of control points, knots and the degree must be inherently satisfied $m = n + k + 1$. As can be deduced from Equation (19), there are ($n + k$) control points to be computed but only ($n + 1$) equations are available, which means we still need to construct another ($k - 1$) equations. In this work, we apply quintic B-spline (k = 5) to fit the joint rotation curve. In a similar way to computation in Cartesian space, we can create constraint conditions in kinematics at the start as well as the end points.

The sequence of data point P_i is the joint angle θ_i at time t_i; the joint velocity can be derived by the first-order derivation of the joint angle, which can be written as

$$\omega_i = \frac{dP_i}{du} = P_i' = \sum_{i=0}^{n-1} N_{i,k-1}(u_k)\{n(C_{i+1} - C_i)\} \tag{25}$$

Hence, we set the velocities of the joint at the start ($\omega_{t_0} = 0$) and end ($\omega_{t_n} = 0$) of the motion to zero to obtain the additional two equations.

Based on the velocity function, the joint acceleration γ_i can be further obtained by the first-order derivation, which can be formulated as

$$\gamma_i = \frac{d^2P_i}{du^2} = \sum_{i=0}^{n-2} N_{i,k-2}(u_k)\{n(n-1)(C_{i+2} - 2C_{i+1} + C_i)\} \tag{26}$$

We can also set the acceleration of the joint at the start ($\gamma_{t_0} = 0$) and end ($\gamma_{t_n} = 0$) of the motion to zero to obtain the last two equations.

With the utilization of all the equations, the control points can be solved and the corresponding trajectory in joint space can be naturally interpolated, satisfying the kinematic limitations and requirement for smoothness.

4. Simulation and Experiment

In order to verify the effectiveness of the proposed method, two cases are designed and both simulations and experiments are conducted. The platform is constructed based on a 6 DoFs serial manipulator produced by JAKA Co., Ltd. The algorithm is programmed and operated on the computer and the control system of the manipulator and computer is connected by the internet, in which TCP/IP is used to send and accept information of the motion data. The experiment platform is shown in Figure 6.

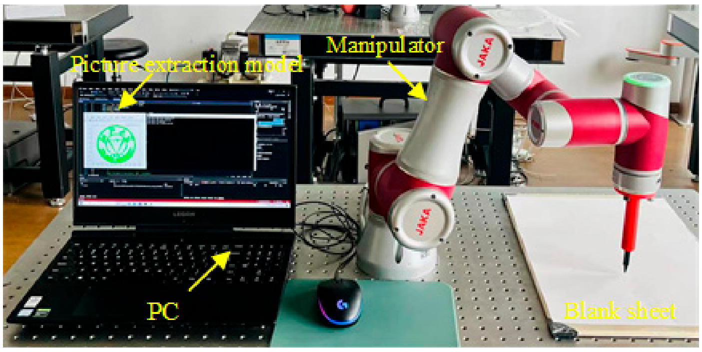


Figure 6. Experiment platform.

4.1. Case 1 of Zhejiang University Logo

In this case, the logo of Zhejiang University, made up of a complex trajectory with many discontinuous closed paths in XY plane, is predefined as for the manipulator to track in an effective and smooth way. According to the proposed method described above, the following steps should be performed.

Step 1: extract the contours of the whole trajectory from the original picture (as shown in Figure 7a) with the use of OpenCV and compute the geometric center, which is treated as discrete cities (as shown in Figure 7b) for each of the closed local paths.

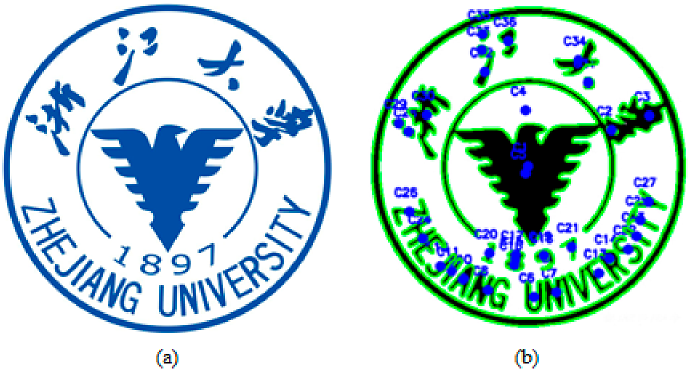


Figure 7. Analysis of the complex trajectory; (**a**) path generation; (**b**) geometric center computation.

Step 2: use the improved ACO to generate the optimal sequence that has the shortest free-load path for the large quantity of discontinuous paths (cities). In order to demonstrate the feasibility and efficiency of the proposed method, another two typical heuristic methods including the genetic algorithm (GA) proposed in article [16] and particle swarm optimization (PSO) presented in article [22] are both used to conduct a comparison analysis. The initial parameters used in each method are given and listed in Table 2.

Table 2. Key initial parameters set in the three methods.

Method	Initial Values of Parameters
improved ACO	Total group size 50, number of iterations 100, heuristic factor $\alpha = 1$ and $\beta = 5$, volatility coefficient $\rho = 0.1$.
GA [16]	Total group size 50, number of iterations 100.
PSO [22]	Total group size 50, number of iterations 100, accelaration constants $c_1 = 2$ and $c_2 = 2$.

The iteration process of searching for the global optimal solution using the three methods is drawn in Figure 8.

To reduce the randomness during the iteration process, four simulations in total were conducted for each method. It can be easily observed from Figure 8a–c that the speed of convergence is different. For a better understanding and more persuasive analysis, we computed the mean values μ_L and standard deviations σ_L of solutions and counted the average iteration times μ_{ite} that each method required for convergence.

As we can see from Table 3, the average of the optimal solution for improved ACO is 1273 mm, which is the smallest among the three methods (the results of GA and PSO are 2397 mm and 3280 mm, respectively); it can be implied that GA and PSO are weaker in their ability to find the global solution. In terms of efficiency, improved ACO needs only about 40 iterations. GA yields the worst result (iteration times are 73), which is even poorer than the PSO dose. In addition, the smaller the standard deviation σ_L, the better the stability. Given all, the improved ACO outperforms two other approaches (GA and PSO) in terms of efficiency and capability in searching for a global optimal solution.

Table 3. Characteristics of each optimization method.

Method	$L_{\max}$/mm	$L_{\min}$/mm	μ_L (mm)	σ_L	μ_{ite}
improved-ACO	1276.8	1269.9	1273.2	2.6	42
GA	2766.7	2144.0	2397.1	243.0	73
PSO	3350.8	3190.4	3280.9	59.1	70

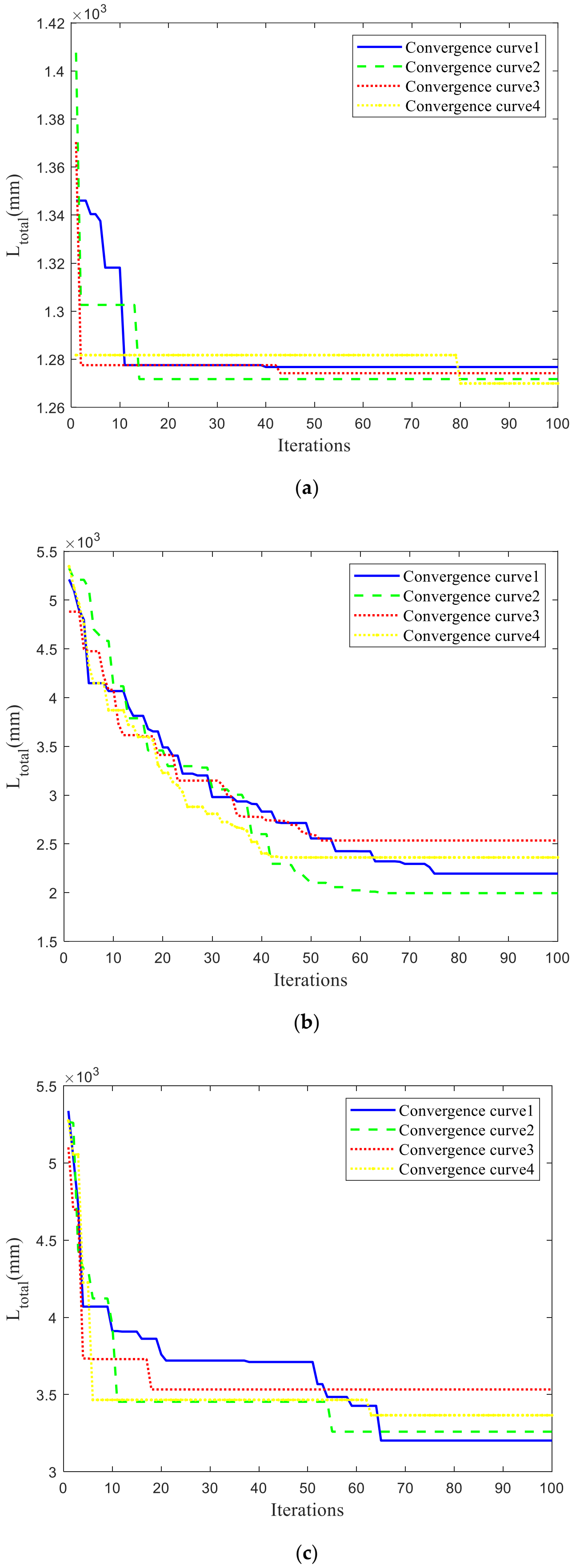
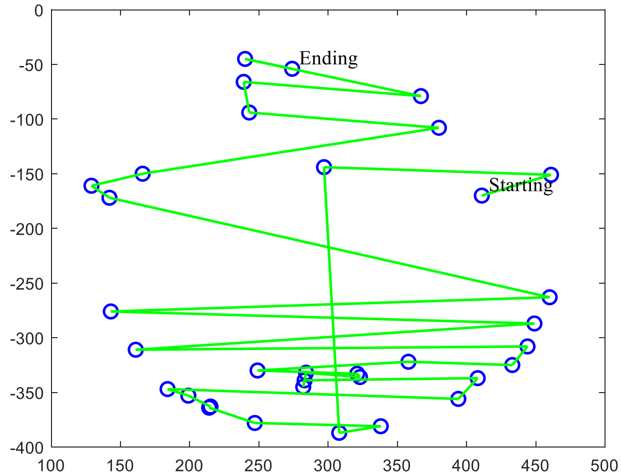
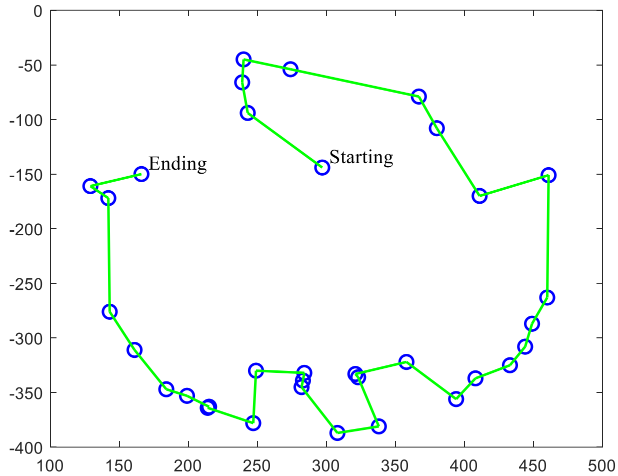


Figure 8. Iteration process of three methods: (**a**) the improved ACO; (**b**) the GA; (**c**) the PSO.

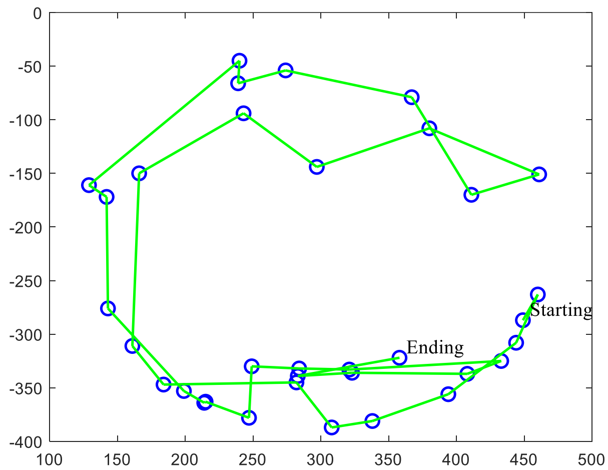
Based on the optimization process, a better sequence of the discontinuous paths can be determined. In order to demonstrate their necessity and significance, a random sequence is generated as the benchmark result, which is similar to the traditional manual teaching method, all the results of each method can be shown in Figure 9a–d.



(**a**)

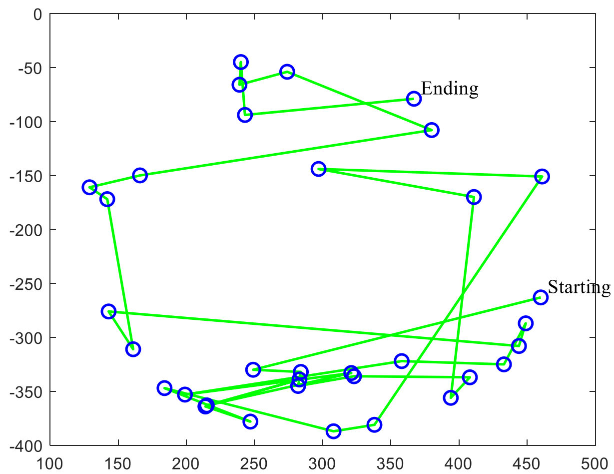


(**b**)



(**c**)

Figure 9. *Cont.*



(**d**)

Figure 9. Sequences generated in different ways: (**a**) a random sequence without any optimization; (**b**) the optimized sequence by improved ACO; (**c**) the optimized sequence by GA; (**d**) the optimized sequence by PSO.

Step 3: determine the starting points of each local path. Generate the curves to fit the series of discrete points obtained in Step 1 and construct a smooth trajectory for the manipulator in Cartesian space with use of the cubic polynomial interpolation functions, as shown in Figure 10. The initial parameters used in each optimization method are given and listed in Table 2.

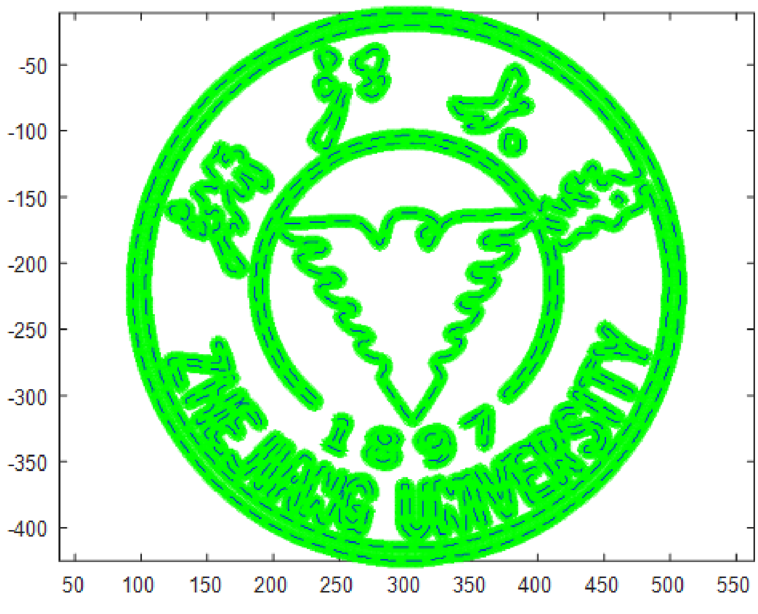
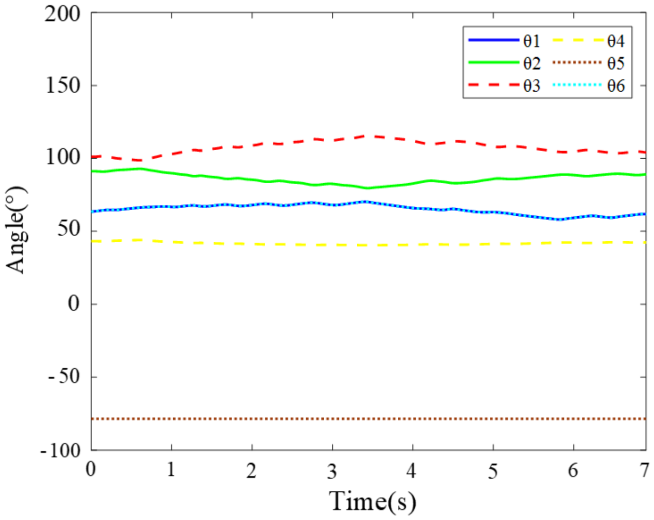


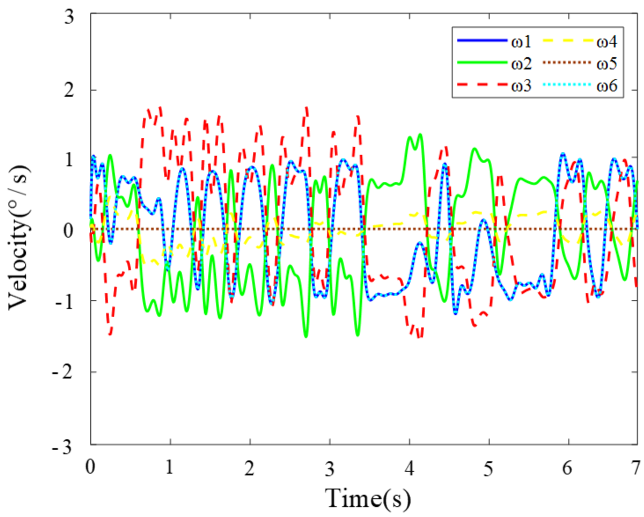
Figure 10. Interpolation in Cartesian space.

Step 4: calculate the joint angles with reference to the coordinate values of the position in Cartesian space based on the kinematic model of the serial manipulator.

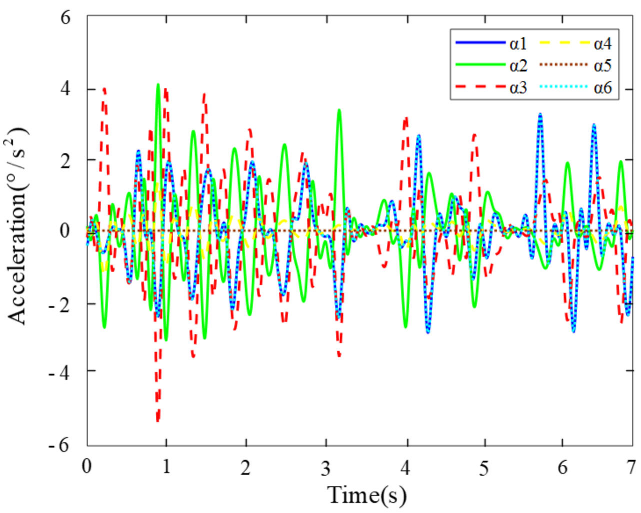
Step 5: conduct path planning in joint space by use of the quintic B-spline interpolation technique, satisfying all the limitations in position, velocity, acceleration and jerk of each joint of the manipulator. The movement is safe and stable since the curve of both acceleration and jerk is continuous as well as quite smooth, as shown in Figure 11.



(**a**)

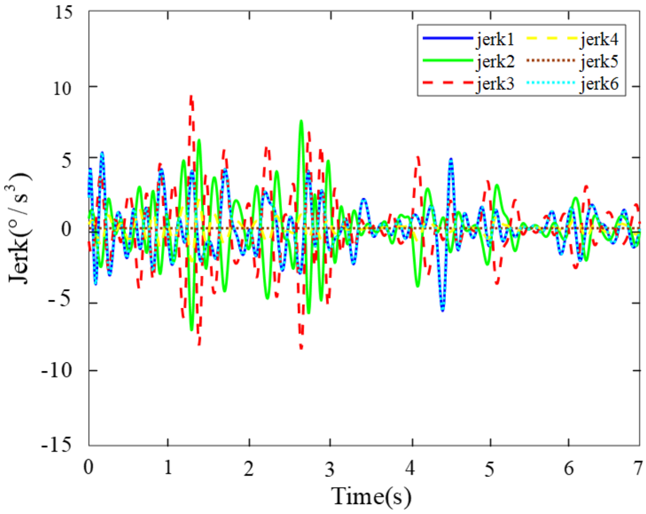


(**b**)



(**c**)

Figure 11. *Cont.*



(**d**)

Figure 11. Interpolation in joint space: (**a**) Position of the joints; (**b**) Velocity of the joints; (**c**) Acceleration of the joints; (**d**) Jerk of the joints.

As we can see from Figure 11a–d, the curves of the velocity, acceleration and jerk of each joint are all continuous and very smooth; there are rarely mutations, which means that the motion of the manipulator can be quite stable without noticeable vibration or periodic shocks. To showcase the practical effects of 5th-order B-spline planning in joint space, we calculated the standard deviations of acceleration and jerk. For a comparative study, values without any interpolation (directly obtained through the inverse kinematic model based on positions in Cartesian space) are treated as benchmark results. The magnitude of variances effectively indicates a decrease in fluctuation in joint state changes after interpolation, as listed in Table 4. The improvement can be calculated by equation, written as:

$$opt_x = \frac{x - \hat{x}}{x} \times 100\% \tag{27}$$

where x represents original value of variables including standard deviation of acceleration σ_α and jerk σ_{jerk}, and $\hat{x}$ is new value after interpolation in joint space by B-spline.

Table 4. Improvement in acceleration and jerk of each joint.

Joint	σ_α	$\hat{\sigma}_\alpha$	opt_α(%)	σ_{jerk}	$\hat{\sigma}_{jerk}$	opt_{jerk}(%)
θ_1	0.0529	0.0015	97.1645	0.5433	0.0370	93.1898
θ_2	0.0223	0.0013	94.1704	0.1384	0.0304	78.0347
θ_3	0.0827	0.0015	98.1862	0.6679	0.0367	94.5052
θ_4	0.0917	0.0015	98.3642	0.6871	0.0364	94.7024
θ_5	0	0	0	0	0	0
θ_6	0.0529	0.0015	97.1645	0.5434	0.0371	93.1726

It can be easily observed that the standard deviation of all joints (except the fifth joint since its angles are zero during the whole process) are smaller after the interpolation process. From a certain point of view, the vibrations and shocks originating from revolute joints can be reduced by more than 90%. The experimental process is shown in the following Figure 12.

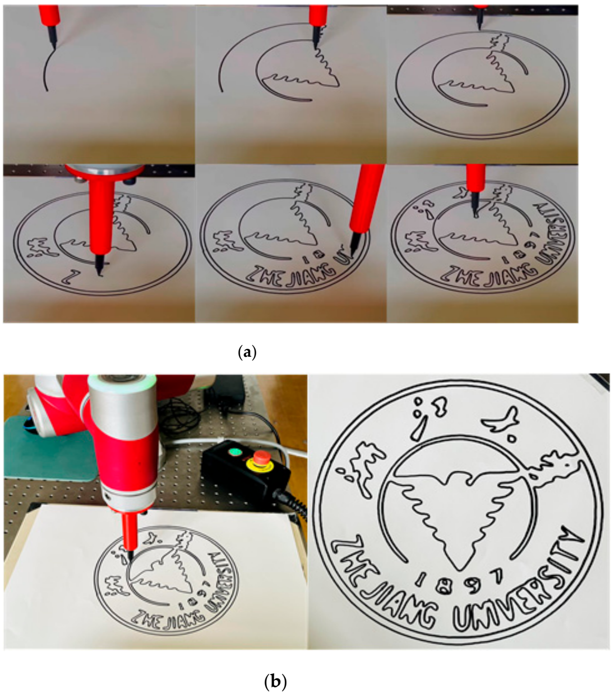


Figure 12. Experiment of the Zhejiang University logo: (**a**) Main steps for the complex trajectory; (**b**) Result of the test.

The whole process for the path planning of a complex trajectory can be summarized in the following flow chart, as shown in Figure 13.

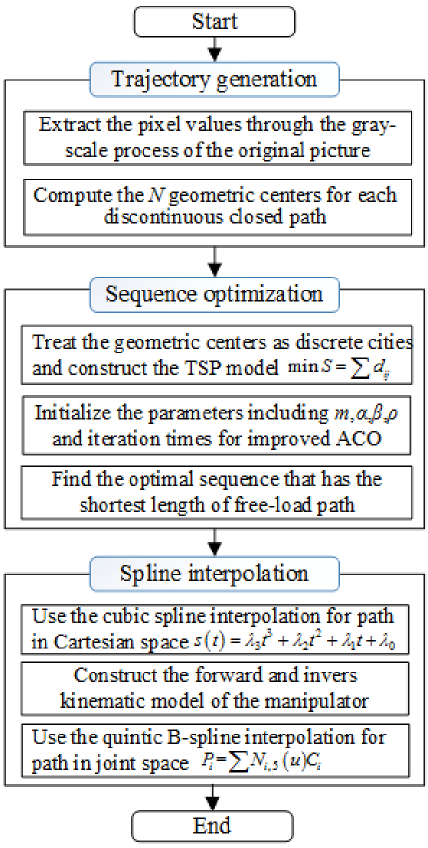


Figure 13. Flow chart of the path planning.

4.2. Case 2 Logo of the Zhejiang SCI-Tech University

In a similar way, we use another pattern of the Zhejiang SCI-Tech University to verify the proposed method. As shown in Figure 14, all the important steps completed in case 1 are carried out once again, in which the significant contours are first extracted and a suitable sequence having the shortest moving distance is obtained by use of the ACO.

Figure 14. Trajectory generation for Zhejiang SCI-Tech University.

After that, the path planning is completed both in Cartesian and joint spaces based on the spline interpolation, as shown in Figure 15.

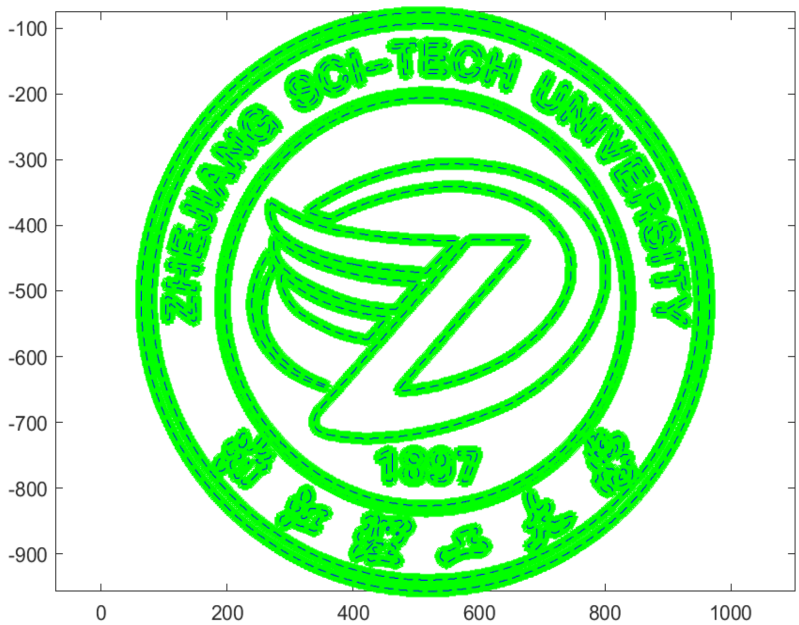


Figure 15. Interpolation in Zhejiang SCI-Tech University logo.

The corresponding experiment was carried out and the result is shown in Figure 16.

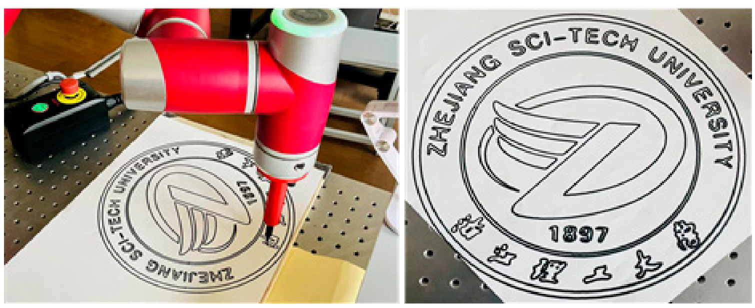


Figure 16. Experiment in trajectory planning of Zhejiang SCI-Tech University.

5. Discussion

5.1. Optimization Process Analysis of Different Methods

Complex trajectory planning with a large quantity of discontinuous paths can be equivalent to a multi-objective nonlinear optimization problem, for which heuristic algorithms are feasible and quite suitable. A random sequence for all the discrete points (cities) is generated to represent the result obtained by the traditional manual teaching method; since no analysis of the complex trajectory is conducted, the total distance of the free-load path is quite long. In addition, with the increase in the scale of discrete points, it is difficult to obtain the same sequence result in a repetitive task, and there are no assurances in the reliability or stability of the manufacturing. With the optimization process, as shown in Figure 8a–c, we can also find that each method exhibits different characteristics and capability in convergence to the optimal solution under the same initial parameters setting. The searching process of improved ACO can be finished after about 20 iterations, while GA requires at least 40 times and PSO takes an average of about 60 times. Therefore, the improved ACO can outperform the other two methods in terms of convergence efficiency.

5.2. Enhancement in Efficiency of Path Planning

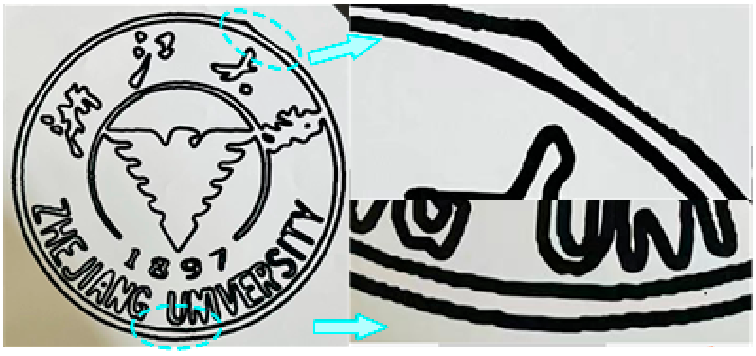
The total length of the free-load path during the transition between each discontinuous path can cause a significant impact on the efficiency of the whole trajectory planning. As shown in Figure 9a–d, the total lengths are calculated based on the sequences generated in four different ways and the data are listed in Table 5. It can be seen that using the random generation method to obtain the sequence, as shown in Figure 9a, the total length of the free-load path L_0 is 3613.6 mm and is treated as the benchmark result. After the optimization by use of the improved ACO, the optimal solution (length) can be limited to a range of which the maximum value is 1276.8 mm and the minimum value is 1269.9 mm. The length can be dramatically shortened and is only about 0.35 of the result of a random sequence. That means that the efficiency can be improved by nearly 65%. The average lengths with the optimization of GA and PSO are about 2144 mm and 3199 mm respectively, which can be decreased to about 0.59 and 0.88 compared with the benchmark result. Therefore, all three heuristic algorithms can improve the path planning efficiency through optimizing the motion sequence to shorten the length of the free-load path. The improved ACO can maximize the amelioration in planning efficiency and obtain the best result among the three methods. In addition, the improved ACO exhibits outstanding capabilities in searching for the global optimal solution while GA and PSO are highly likely to converge into a local optimal result.

Table 5. Efficiency improvement of the three methods.

Method	m	L_0/mm	$L_{\max}$/mm	$L_{\min}$/mm	L_{min}/L_0	Improvement (%)
improved-ACO	50	3613.6	1276.8	1269.9	0.35	65%
GA	50	3613.6	2766.7	2144.0	0.59	41%
PSO	50	3613.6	3350.8	3190.4	0.88	12%

5.3. Improvement in Smoothness of the Trajectory

Path planning mainly includes the generation of an optimal movement sequence and trajectory interpolation. The interpolations in Cartesian space and joint space are critical to ensure the motion continuity and smoothness of the manipulator. The initial contours extracted from the original picture with use of OpenCV are made up of a series of discrete points (e.g., the number of the Zhejiang University logo is 3625). In order to achieve an accurate fitting, cubic splines interpolation is conducted (this results in an increase in scale of discrete points to 7929) and the manipulator is able to move along the predesigned path. However, periodic shocks, vibrations and abrasion may occur due to discontinuous mutation of the speed of joint rotation. It is essential to carry out further planning in joint space; the angles with reference to each discrete path point are calculated based on a kinematic model of the manipulator. Quintic B-spline is used to complete a denser interpolation (the number of discrete points is increased to 23,823). As we can see Figure 11a–d, the high order of interpolation functions can guarantee excellent continuity and smoothness in the curves of joint position, velocity, acceleration and jerk, which can significantly strengthen the stability of motion and improve accuracy (as shown in Figure 17).



(**a**)

(**b**)

Figure 17. Comparison of trajectory smoothness: (**a**) Planning only in Cartesian space; (**b**) Planning both in Cartesian space and joint space.

6. Conclusions

A comprehensive optimal method for path planning with integration of the improved ant colony algorithm (ACO) and the high-order spline interpolation technique is proposed. The efficiency of the planner and the excellent smoothness in the trajectory make it highly suitable for application in industrial including sorting, handing, polishing and welding, and even in fruit picking in agriculture. Some conclusions are summarized as follows, for a better readability, all symbols and acronyms are listed in Appendix A.

(1) The efficiency of path planning for a complex trajectory can be improved by nearly 65% after an optimization process. The trajectory consists of a large number of discontinuous closed paths, which makes it rather difficult to find an appropriate motion sequence. The improved ACO can ameliorate the efficiency by nearly three times compared with a traditional random method, since the total length of the free-load path can be shortened to 0.35, while the values are approximately 0.59 and 0.88 for GA and PSO, respectively.

(2) Improved ACO exhibits an impressive capability in searching for the global optimal solution with better efficiency by 40% in convergence. With the same size of population and other parameters, the optimization process can be converged only after about 40 iterations, much faster than that of GA and PSO, which require at least

73 and 70 iterations, respectively. Furthermore, the optimized solution for improved ACO is about 1267 mm (close to the global optimal point), while the solutions for the other two methods are 2144 mm and 3190 mm, respectively (both are local optimal points).

(3) Planning in both Cartesian and joint spaces can significantly enhance the smoothness of trajectories and improve accuracy. The use of cubic spline interpolation in Cartesian space ensures the generation of precise complex trajectory after the preliminary extraction from the original picture. While path planning in joint space using quintic B-spline can strengthen the continuity and smoothness of acceleration and jerk of the joint, the standard deviation which can be used to measure stability of joint rotation is decreased by more than 90%, and can naturally reduce motion shocks and vibrations.

(4) The proposed method has a wide range of applications and can be used in many kinds of tracking tasks, such as spraying, welding, painting, and so on.

Author Contributions: Conceptualization, Z.X.; Data curation, L.H.; Funding acquisition, W.W. and L.H.; Investigation, K.L.; Methodology, Z.X., W.W., Y.C. and K.L.; Resources, Y.C. and K.L.; Software, Z.X. and Y.C.; Supervision, W.W.; Visualization, L.H.; Writing—original draft, Z.X.; Writing—review & editing, W.W. and L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Zhejiang Province (grant number LQ22E057053), the National Natural Science Foundation of China (grant number 52175032) and the Key R&D Program of Zhejiang Province (grant number 2023C01180, 2022C01101 and 2021C04017).

Data Availability Statement: All data presented in this study are included in the paper.

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

Appendix A

All the important acronyms and variables are provided and listed in the following tables to make it easier to understand the research.

Table A1. Acronym list.

Acronym	Meaning	Acronym	Meaning
ACO	improved ant colony algorithm	SA	simulated annealing algorithm
PSO	particle swarm optimization algorithm	NURBS	non-uniform rational basis spline
GA	genetic algorithm	TSP	travelling salesman problem
PRM	probabilistic roadmap method	RRT	rapidly exploring random tree

Table A2. Variable list.

Variable	Meaning	Variable	Meaning
c_i	geometric center of each local path	$\delta\tau_{ij}^k$	pheromone produced by the kth ant
L	total length of free-load path	$s(t)$	piecewise polynomials
L_0	length of free-load path by randomly generated	N	total iteration times
L_{max}	maximum total length of free-load path	m	initial population size
L_{min}	minimum total length of free-load path	$\lambda_{k,j}$	coefficients of polynomials
d_{ij}	distance between ith city and jth city	k	degree of polynomial
τ_{ij}	intensity measure of the pheromone	u_i	knot vector in B-spline
η_{ij}	heuristic value	ω_i	joint velocity
ρ	volatility coefficient	γ_i	joint acceleration
$\Delta\tau_{ij}$	total increase in pheromone	σ	standard deviation of variable

References

1. Wang, X.; Zhou, X.; Xia, Z.; Gu, X. A survey of welding robot intelligent path optimization. *J. Manuf. Process.* **2021**, *63*, 14–23. [CrossRef]

2. Ma, G.; Duan, Y.; Li, M.; Xie, Z.; Zhu, J. A probability smoothing Bi−RRT path planning algorithm for indoor robot. *Future Gener. Comput. Syst.* **2023**, *143*, 349–360. [CrossRef]
3. Gao, W.; Tang, Q.; Yao, J.; Yang, Y. Automatic motion planning for complex welding problems by considering angular redundancy. *Robot. Comput. Integr. Manuf.* **2020**, *62*, 862–878. [CrossRef]
4. Xue, Z.; Zhang, X.; Liu, J. Trajectory planning of a dual−arm space robot for target capturing with minimizing base disturbance. *Adv. Space Res.* **2023**. [CrossRef]
5. Sanchez-Lopez, J.L.; Castillo-Lopez, M.; Olivares-Mendez, M.A.; Voos, H. Trajectory Tracking for Aerial Robots: An Optimization−Based Planning and Control Approach. *J. Intell. Robot. Syst.* **2020**, *100*, 531–574. [CrossRef]
6. Fan, P.; Yan, B.; Wang, M.; Lei, X.; Liu, Z.; Yang, F. Three−finger grasp planning and experimental analysis of picking patterns for robotic apple harvesting. *Comput. Electron. Agric.* **2021**, *188*, 106353. [CrossRef]
7. Gasparetto, A.; Zanotto, V. Optimal trajectory planning for industrial robots. *Adv. Eng. Softw.* **2010**, *41*, 548–556. [CrossRef]
8. Tamizi, M.G.; Yaghoubi, M.; Najjaran, H. A review of recent trend in motion planning of industrial robots. *Int. J. Intell. Robot. Appl.* **2023**, *7*, 253–274. [CrossRef]
9. Givehchi, M.; Ng, A.; Wang, L. Evolutionary optimization of robotic assembly operation sequencing with collision−free paths. *J. Manuf. Syst.* **2011**, *30*, 196–203. [CrossRef]
10. Abualigah, L. Group search optimizer: A nature−inspired meta−heuristic optimization algorithm with its results, variants, and applications. *Neural Comput. Appl.* **2021**, *33*, 2949–2972. [CrossRef]
11. Sandberg, A.; Sands, T. Autonomous Trajectory Generation Algorithms for Spacecraft Slew Maneuvers. *Aerospace* **2022**, *9*, 135. [CrossRef]
12. 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]
13. Wang, T.; Xue, Z.; Dong, X.; Xie, S. Autonomous Intelligent Planning Method for Welding Path of Complex Ship Components. *Robotica* **2020**, *39*, 428–437. [CrossRef]
14. Zhou, X.; Wang, X.; Xie, Z.; Li, F.; Gu, X. Online obstacle avoidance path planning and application for arc welding robot. *Robot. Comput. Integr. Manuf.* **2022**, *78*, 413–425. [CrossRef]
15. Chen, G.; Luo, N.; Liu, D.; Zhao, Z.; Liang, C. Path planning for manipulators based on an improved probabilistic roadmap method. *Robot. Comput. Integr. Manuf.* **2021**, *72*, 196–214. [CrossRef]
16. Elhoseny, M.; Tharwat, A.; Hassanien, A.E. Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm. *J. Comput. Sci.* **2018**, *25*, 339–350. [CrossRef]
17. Wang, J.; Li, Y.; Li, R.; Chen, H.; Chu, K. Trajectory planning for UAV navigation in dynamic environments with matrix alignment Dijkstra. *Soft Comput.* **2022**, *26*, 12599–12610. [CrossRef]
18. Yu, X.; Dong, M.; Yin, W. Time−optimal trajectory planning of manipulator with simultaneously searching the optimal path. *Comput. Commun.* **2022**, *181*, 446–453. [CrossRef]
19. Liang, Y.; Yao, C.; Wu, W.; Wang, L.; Wang, Q. Design and implementation of multi−axis real−time synchronous look−ahead trajectory planning algorithm. *Int. J. Adv. Manuf. Technol.* **2022**, *119*, 4991–5009. [CrossRef]
20. Chettibi, T. Smooth point−to−point trajectory planning for robot manipulators by using radial basis functions. *Robotica* **2018**, *37*, 539–559. [CrossRef]
21. Reiter, A.; Muller, A.; Gattringer, H. On Higher−Order Inverse Kinematics Methods in Time−Optimal Trajectory Planning for Kinematically Redundant Manipulators. *IEEE Trans. Ind. Inform.* **2018**, *14*, 1681–1690. [CrossRef]
22. Kim, J.J.; Lee, J.J. Trajectory Optimization With Particle Swarm Optimization for Manipulator Motion Planning. *IEEE Trans. Ind. Inform.* **2017**, *11*, 620–631. [CrossRef]
23. Taitler, A.; Ioslovich, I.; Karpas, E.; Gutman, P.-O. Minimum Mixed Time–Energy Trajectory Planning of a Nonlinear Vehicle Subject to 2D Disturbances. *J. Optim. Theory Appl.* **2022**, *192*, 725–757. [CrossRef]
24. Consolini, L.; Laurini, M.; Locatelli, M.; Cabassi, F. Convergence Analysis of Spatial−Sampling−Based Algorithms for Time−Optimal Smooth Velocity Planning. *J. Optim. Theory Appl.* **2020**, *184*, 1083–1108. [CrossRef]
25. Yang, Y.; Wei, Y.; Lou, J.; Cabassi, F. Nonlinear dynamic analysis and optimal trajectory planning of a high−speed macro−micro manipulator. *J. Sound Vib.* **2017**, *405*, 112–132. [CrossRef]
26. Jahanpour, J.; Motallebi, M.; Porghoveh, M. A Novel Trajectory Planning Scheme for Parallel Machining Robots Enhanced with NURBS Curves. *J. Intell. Robot. Syst. Theory Appl.* **2016**, *82*, 257–275. [CrossRef]
27. Chen, K.Y.; Wu, T.L.; Fung, R.F. Hamiltonian−based minimum−energy trajectory planning and tracking control for a motor−table system: Part I minimum−energy methods in trajectory planning. *Int. J. Dyn. Control* **2019**, *7*, 866–874. [CrossRef]
28. Wang, Z.; Li, Y.; Shuai, K.; Zhu, W.; Chen, B.; Chen, K. Multi—objective Trajectory Planning Method based on the Improved Elitist Non—dominated Sorting Genetic Algorithm. *Chin. J. Mech. Eng.* **2022**, *35*, 7–15. [CrossRef]
29. Craig, J.J. *Introduction to Robotics*; Addison—Wesley: Boston, MA, USA, 2010; pp. 2187–2195.