Research on Collision Avoidance Path Planning of Double SCARA Robot

Lifeng Dang\textsuperscript{1, }* Liangan Zhang\textsuperscript{2}, Qin Shi\textsuperscript{1}

\textsuperscript{1}Zhenjiang Vocational Technical College, Zhengjiang, China
\textsuperscript{2}Anhui University of Technology, MAanshan, China

*Corresponding author e-mail: lifengdang@zjc.edu.cn

Abstract. This paper takes double SCARA robot as the research object and uses the computer graphics method to get the collision detection method of two robotic arms at a certain moment. On the basis of collision detection, the C-space method is used to plan the collision avoidance path for the two robots and the two robots considering the end-effector, and the collision avoidance algorithm for the double SCARA robot is obtained. Then, the simulation experiment is carried out to verify the correctness of the algorithm.

Keywords: Double SCARA robot, C space, End effector, Collision avoidance plan.

1. Introductions
With the proposal of the slogan "Made in China 2025", the market demand for industrial robots increases sharply. At the same time, with the rapid development of robot technology, the functional requirements of robots in various fields are getting higher and higher. Therefore, how to complete complex, diverse and flexible tasks have become a hot research direction of the industrial robots. In order to meet the above requirements and enable the robot to perform complex tasks, the concept of two-robot and multi-robot collaboration was proposed, in which two or more robots can coordinate and cooperate with each other to complete the assigned task.

The double-robot collaboration is not a simple combination of two robots, but to allow them collaborate and work together to accomplish the same specified complex task. As the two robots have crossed workspace, this requires them not to interfere or collide with each other during movement in the common workspace. Therefore, the collision avoidance of two robots becomes an important problem of path planning in movement control of two robots.

Considering the double-robot end-effector, the double-SCARA robot is taken as the research object in this paper to study the non-collision path planning of two-robot cooperation in completing the assigned task.

1.1. Establishment of geometry model for double SCARA Robot
Establish the simple geometric model of double SCARA robot as shown in Figure 1. In order to simplify the calculation and for research convenience, we simplified the big arm, small arm and end-actuators of each robot of the SCARA robot into line segments. Among them, we called the left robot in Figure 1 as the left arm and the right robot as the right arm. As shown in the figure, the distance between the bases
of the two robots is \( L \), and the set parameters of the left and right robots are exactly the same. The length of the big arm is \( l_1 \), the length of the forearm is \( l_2 \) and the length of the end-effector is \( l_3 \).

![Fig. 1 Simple geometric model of double SCARA robot](image)

2. **Collision detection algorithms for double robot manipulator**

The collision of double manipulator usually includes the following three aspects: ① The end of one manipulator collides with the end of another manipulator; ② The end of one manipulator collides with the arm connecting rod (large and small arms) of another manipulator; ③ The arm connecting rod (large and small arms) of one manipulator collides with that of another manipulator. Therefore, according to the double SCARA robot model established in the previous section, the collision of the two robots is summarized as follows: if a line segment of either robot of the two robot models intersects with any line segment of the other robot, the two robots collide. The intersection of two segments is determined by the method described below.

First of all, the intersection of two line segments can be divided into the following two situations: (1) One line segment intersects another line segment, and the two endpoints of any line segment are located on both sides of another line segment; (2) One line segment intersects another line segment, and an endpoint of one line segment is on another line segment. The schematic diagram of the above two situations is shown in Figure 2 and Figure 3.

We use the cross product to judge the intersection of two-line segments. This method is carried out in two steps, the first step is the rapid rejection experiment, and the second step is the straddled experiment. These two steps are described in detail as below.

Rapid exclusion experiment: Firstly, we judge whether the projections of two lines on \( x \) and \( y \) coordinates coincide, that is, to judge whether the endpoints with larger \( x \) coordinates of one line segment are smaller than the endpoints with smaller \( x \) coordinates of the other line segment. If so, then the two-line segments have no intersection. Similarly, judge can be made on the endpoints of \( y \) coordinate. At the same time, the rapid rejection experiment also eliminates the special case that the two-line segments are collinear and have no intersection point.

![Fig. 2 Schematic diagram of Intersection case 1 of the two-line segments](image)
Straddle experiment: If two line segments intersect and straddle each other, as shown in Figure 2, the two endpoints of any line segment are on both sides of the other line segment. To judge the intersection of line CD and line AB is to first judge whether points A and B are on both sides of line CD first. From the cross product of vectors, it can be obtained that \((A-d) \times (c-d) \leq 0\) and \((b-d) \times (c-d) \leq 0\). Then to judge whether points C and D are on both sides of line AB, that is, \((d-b) \times (a-b) \leq 0\). If both conditions are satisfied, then it can be judged that the two line segments intersect. Moreover, the cross product has an important property: the positive or negative relation between two vectors can be judged by its sign. Suppose: \(A_1= (x_1, y_1)\) and \(A_2= (x_2, y_2)\). If \(A_1 \times A_2 > 0\), then \(A_1\) is in the clockwise direction of \(A_2\). If \(A_1 \times A_2 < 0\), then \(A_1\) is in the counterclockwise direction of \(A_2\); if \(A_1 \times A_2 = 0\), then \(A_1\) and \(A_2\) are collinear, they may be in the same direction or in the opposite direction. This property is of great significance in the subsequent search algorithm, which can avoid unnecessary search paths and thus improve the search speed.

Through the above method, we can judge the intersection of any two line segments and get an algorithm to judge the intersection of two line segments, so as to get a method to judge the collision property of the manipulator by detecting the intersection of two line segments.

3. Establishment of C space

3.1. Establishment of slave C space

In the algorithm of this paper, we specified on the left side robot as the master robot, and the right-side robot as the slave robot, the master robot has higher priority in movement compared with the slave robot, and it will move according to the planned path. In the process of movement, the slave robot needs to adjust its own movement according to the master robot to avoid the movement of the master robot, so as to avoid the collision between two robots.

C space is also known as configuration space. Its coordinate variables are a set of parameters that can determine the rigid body pose of an object in motion. Namely, a specific pose of the rigid body in motion of an object in three-dimensional space is transformed into a coordinate point in C space. As mentioned above, the master robot moves according to the pre-planned path, and the slave robot adjusts its own movement according to the movement of the master robot to avoid collision. In this case, the master robot is an obstacle to the movement of the slave robot. The slave robot’s joint angle \(\theta_{R1}, \theta_{R2}, \text{and } \theta_{R3}\) are selected as the coordinate variables of the slave robot’s C space, and the value range of the joint angle is set as \(-155^\circ \leq \theta_{R1} \leq 155^\circ, -135^\circ \leq \theta_{R2} \leq 135^\circ, -\pi \leq \theta_{R3} \leq \pi\). In other words, the slave robot's C space can be expressed as:

\[
C_R = \{ (\theta_{R1}, \theta_{R2}, \theta_{R3}) | -155^\circ \leq \theta_{R1} \leq 155^\circ, -135^\circ \leq \theta_{R2} \leq 135^\circ, -\pi \leq \theta_{R3} \leq \pi \} \tag{1}
\]

3.2. Rasterization of C space

Suppose the joint rotation of the double-SCARA robot is at the limit angular velocity, and the values are respectively \(\omega_{R1}, \omega_{R2}, \omega_{R3}, \omega_{L1}, \omega_{L2}, \omega_{L3}\), and the trajectory update time is \(\Delta t\). Then the angles turned by each joint within time \(\Delta t\) can be expressed as \(\Delta \theta_{Ri} = \omega_{Ri} \Delta t\), \(\Delta \theta_{Li} = \omega_{Li} \Delta t\), where \(i = 1, 2, 3\). Now the turned angle of each joint of the slave robot in unit time is taken as the unit to rasterize the
slave Robot’s C space into l*m*n, and the length of each part is ∆θ₁, ∆θ₂, ∆θ₃, namely, 
\[ l = \theta_R^1 / \Delta \theta_R^1, m = \theta_R^2 / \Delta \theta_R^2, \]
\[ n = \theta_R^3 / \Delta \theta_R^3. \]
As shown in Fig 4, Let θ_jk represent the spatial pose of the slave Robot at time t=k, 
then when the slave robot is moving at maximum angular velocity, there may be 8 adjacent position 
coordinates points when moves from Δt to t=k+1, which are 
\[ \theta_1 = \theta_{i-1,j+1,k+1}, \theta_2 = \theta_{i+1,j,k+1}, \theta_3 = \theta_{i+1,j-1,k+1}, \theta_4 = \theta_{i,j,k+1}, \theta_5 = \theta_{i,j,k-1}, \theta_6 = \theta_{i-1,j,k+1}, \theta_7 = \theta_{i-1,j-1,k+1}, \theta_8 = \theta_{i-1,j,k+1} \]
spectively.

![Fig. 4 Possible motion directions of the slave robot from time t=k to time t=k+1](image)

When considering the end executor, the slave robot’s C space is a three-dimensional grid space. The 
spatial pose point of slave Robot at t=k was set at the center of a small three-dimensional square, that is, 
the center of each three-dimensional small square represents the pose point, and has priority of 
movement with its six adjacent three-dimensional small squares, then the possible pose point of Slave 
robot at t=k+1 may also have 6 adjacent pose coordinate points. After the collision detection of the six 
pose coordinates is completed, the collision detection of the adjacent pose points of the six pose points 
are carried out according to the current results.

As shown in Figure 4, the shadow area in the figure is the mapping of the master robot in slave 
robot’s C space, which is the result of collision detection. The shadow part is the obstacle space of the 
slave robot’s C space. Then the remaining 6 poses are the free space poses of the slave robot’s C space, 
and the free path of the slave robot at this moment only needs to be searched in 6 free space positions. 
Repeat the above process until the slave robot reaches the target point, then the collision detection is 
over and search the motion path of the slave robot.

In this paper, the end-effector of the master and slave robots is taken into account in consideration 
of the actual dual-SCARA robot system and its parameters such as connecting rod parameters and base 
distance, so as to facilitate the control of the system in the later stage. It is not necessary to consider the 
effect of the end-effector on the motion of the system after the collision avoidance of the two robots is 
completed.

4. Search of obstacle boundary in C space

In this algorithm, an optimized A^* algorithm, namely edge search A^* algorithm, is used to search. 
Based on the theory of A^* algorithm, we define the valuation function as: 
\[ f(n) = g(n) + h(n), \]
where 
\[ g(n) = \Delta i \Delta \theta_R^1 + \Delta j \Delta \theta_R^2 + \Delta k \Delta \theta_R^3 \]
(2)
\[ h(n) = \sqrt{(\theta_{BSC} - \theta_{BNS})^2 + (\theta_{BNC} - \theta_{BNS})^2 + (\theta_{BNC} - \theta_{BSC})^2} \]
As we know, the A^* algorithm requires Open table and Close table to record the changes of relevant points, and the maintenance of this will greatly affect the running speed of A^* algorithm. In order to improve the traversal speed of the algorithm, the edge search A^* algorithm is adopted. It uses the Now table and the Later table to record the search edge points, and adopts the idea of iterative extension to conduct depth-first search. This has avoided the shortcoming of repetitive search that every search of points in the Open table and the Close table starts from scratch when A^* algorithm is adopted, and has improved the searching efficiency.

5 Simulation experiment and verification

The simulation system is established by using the above simple model of double SCARA robot, collision detection algorithm and obstacle boundary search algorithm. The relevant parameters are set for the two robots, and the simulation results are obtained and analyzed.

The simulation experiment parameters of the double-SCARA robot system are set as follows:

(1) The distance between the bases of the two robots is L=0.8m, the big arm of the two robots is \( l_1 = 0.4m \), the forearm is \( l_2 = 0.45m \), and the length of the end-effector is \( l_5 = 0.2m \);

(2) The joint angle motion range of the master robot and the slave Robot is the same, they are \(-155^\circ \leq \theta_{R1} \leq 155^\circ, -135^\circ \leq \theta_{R2} \leq 135^\circ, -\pi \leq \theta_{R3} \leq \pi \) respectively;

(3) Updated time of the set trajectory is \( \Delta_t = 2ms \);

(4) The base position of the master robot is (-0.4, 0), and the base position of the slave robot is (0.4, 0).

Fig. 5 Search diagram of the obstacle boundary of the slave robot’s C space

Fig. 6 Path display when the master robot is stationary
Fig. 7 Path display when the master and the slave robots move simultaneously

Fig. 5 shows the search diagram of the obstacle boundary of the slave robot’s C space; Fig 6 and 7 shows respectively the collision avoidance motion path of the dual-SCARA robot when the master robot is stationary and when the master and the slave robots move simultaneously without considering the end-effector.

Fig. 8 Obstacle boundary search graph of slave robot’s C space when considering the end-effector

Fig. 9 Path display when the master robot is stationary
It can be seen from the above simulation results that the simulation results meet our expectations and verify the correctness of the algorithm. The dual-SCARA robot system can move from the starting point to the target point according to our setting and task requirements, and complete the collision avoidance movement between each other.

5. Conclusions
In this paper, we use the C space method to complete the collision avoidance path planning for the double-SCARA robot. And we adopted a simple collision detection method to quickly detect the collision between two robots, and proposed an optimized A* algorithm to solve the problem of low searching efficiency of A* algorithm, so as to improve the traversal searching speed and quickly complete the path planning of collision avoidance between the two robots.

References
[1] Chen Youdong, Yan Liang, Gu Pingping. Detection collision algorithm for two-manipulator system [J] Journal of Beijing University of Aeronautics and Astronautics 2013(12).
[2] Zou Yisheng. Research on Real-time Collision Detection Algorithm in virtual Prototype Environment [M]. Southwest Jiaotong University 2009.
[3] Ouyang Fan. Research on coordinated motion method of double robots [D]. South China University of Technology 2013.
[4] Zhang Tie, Ouyang Fan. Coordinated welding task planning and simulation of two robots [D].Journal of welding 2012(12)
[5] Hen Haoyu, Wu Hongtao, Chen Bo, Yan Cheng, Jiang Yanjie. Coordination obstacle avoidance algorithm of two-arm redundant robot [J] Journal of Agricultural Machinery 2015(09).