U.P.B. Sci. Bull., Series D, Vol. 84, Iss. 2, 2022 ISSN 1454-2358

KINEMATICS ANALYSIS AND TRAJECTORY PLANNING OF ABB-IRB2400 ROBOT

Qingchun ZHENG[1,2], Jingjing JIA[1,2], Peihao ZHU[1,2*], Yinhong XIAO[1,2], Fudong ZHANG[1,2], Wenpeng MA[1,2], Shuoshuo ZHAO[1,2]

According to the parameters of the ABB-IRB2400 robot model, the D-H parameter method is used to establish the robot's linkage coordinate system and determine the parameter values. The coordinate transformation matrix ${}^{0}_{6}T$ *of IRB2400 robot is obtained through forward kinematics analysis, and the inverse kinematics transformation method is used to solve the joint variables of the robot. The feasibility and correctness of the forward and inverse kinematics of the robot are verified by modeling the robot entity and importing ADAMS simulation analysis software. Combining MoveIt's robot kinematics solver and RRT algorithm motion planning library, RVIZ is used to verify the feasibility of correlation trajectory algorithm for linear and arc interpolation trajectory planning of robot arm, which provides a foundation for weld grinding robot trajectory planning under complex working conditions.*

Keywords: IRB2400robot, robot arm, kinematics, trajectory planning

1. Introduction

In today's production and manufacturing, due to the high requirements for the surface quality of grinding and polishing workpiece after welding, the domestic post-treatment technology is still in its infancy, and the vast majority of grinding work adopts manual operation. However, the dimensional consistency of manually ground parts is poor, and the grinding and polishing work is often in high temperature, dust, noise, radioactive and polluted environment. Metal dust and noise will cause serious harm to workers [1].

With the widespread application of automation and intelligent equipment,

[1] Tianjin Key Laboratory for Advanced Mechatronic System Design and Intelligent Control, Tianjin University of Technology, Tianjin 300384, China
[2] National Demonstration Center for Experimental Mechanical and Electrical Engineering Education (Tianjin University of Technology)
* Corresponding author: Peihao Zhu, E-mail: zhupeihao_de@163.com

in most grinding and polishing processes, such as the grinding of castings, the grinding and polishing of weld allowances, and the grinding and polishing of sanitary appliances, some automatic grinding and polishing equipment are often used instead of manual operations [2]. However, the universality of professional grinding machine tools is poor, which is only suitable for mass production; CNC grinding machine tool has high cost, lack of flexibility, narrow application range of equipment and small expansion range [3]. Therefore, in recent years, industrial robots have been widely used in the intelligent equipment manufacturing industry due to their high degree of freedom, strong adaptability, and simple programming [4], especially in robotic welding equipment, which has accelerated the pace of industrial automation [5].

The trajectory planning of weld grinding and polishing robot refers to the path planning of the robot's end walking after welding, which mainly involves the pose adjustment of the end effector. Chen Qiang [6] et al. carried out 3D reconstruction on the edge of the complex weld groove, and obtained the corresponding posture by querying the robot posture database to complete the trajectory planning. Zhu Denglin [7] and others combined graphics and technology to transform the three-dimensional robot pose into two-dimensional method, which effectively solved the problem of trajectory planning. For the motion planning of the welding torch in robot welding, H.C. Fang [8] et al. took a Y-shaped joint as an example and proposed an optimization algorithm that can correctly select the inclination angle of the welding torch and the path planning of the weld bead. For spatial welds with more complex trajectories, the teaching and programming efficiency of the grinding and polishing robot are low, and it is difficult to ensure the ideal robot processing posture, which brings great troubles to industrial production.

The above limitations have led to a bottleneck in the application and development of intelligent grinding and polishing automation equipment in complex applications. Therefore, this article analyzes and solves the forward and inverse kinematics for the pose of the ABB-IRB2400 grinding and polishing robot in operation, and uses the method of combining the robot kinematics solver and the RRT algorithm motion planning library to plan the trajectory of the grinding and polishing path.

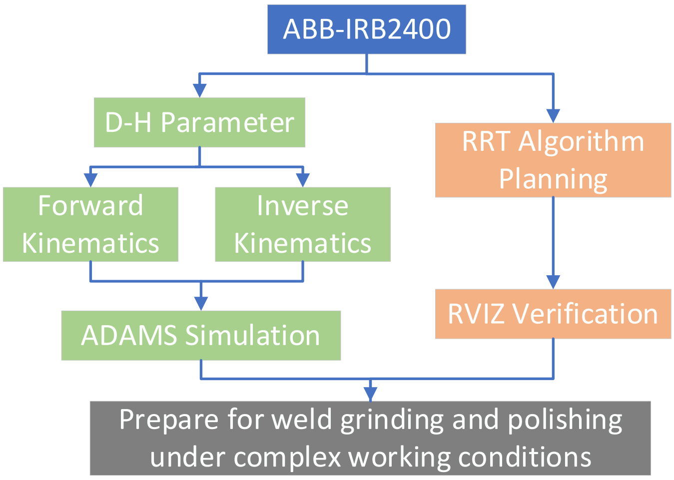


Fig. 1 Technology roadmap

The technical route of this paper is shown in Figure 1, in order to provide a feasible method for the path planning of the welding seam grinding and polishing robot under complicated working conditions.

2. Kinematic analysis

2.1 Coordinate system establishment

The IRB2400 grinding and polishing industrial robot is an articulated robot designed by ABB to imitate a human arm. It has a total of six degrees of freedom, and the six joints are all rotating joints, which execute on the base, body, lower arm, arm, wrist and end. In the case of robots, each revolute joint corresponds to a degree of freedom, as shown in Figure 2.

On the basis of using D-H parameters [9] to establish the link coordinate system rules, the base coordinate system and each link coordinate system are established for the ABB-IRB2400 robot. The coordinate system attached to the robot arm and the base is recorded as the coordinate system {0}, and the coordinate system attached to the robot arm link i is recorded as {i}, so as to determine the mutual relationship between the adjacent links of the robot arm sports relations. The base coordinates of the robot and the coordinate system {1}, {2}, {3}, {4}, {5}, {6} of the robot arm are set and the coordinate axis directions are shown in Figure 3.

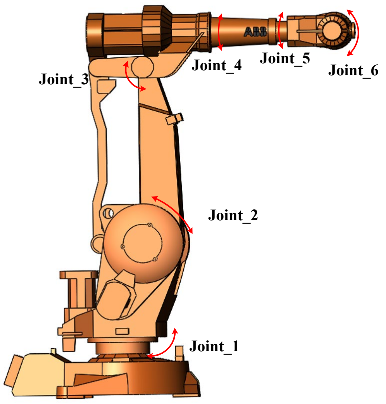


Fig. 2 ABB-IRB2400 industrial robot

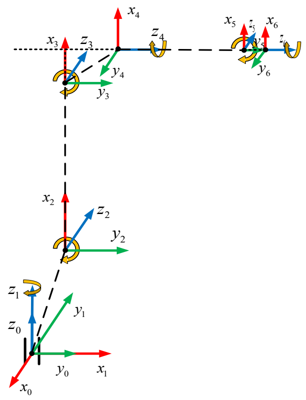


Fig. 3 The coordinate system of each link of the robot

According to the set link coordinate system, the corresponding link parameters can be defined as:

Pole length:a_{i-1}(mm), The distance measured from z_{i-1} to z_i along x_{i-1};

Twist angle: α_{i-1}(°), The angle of rotation from z_{i-1} to z_i around x_{i-1};

Offset: d_i(mm), the distance measured from x_{i-1} to x_i along z_i;

Rotation angle: θ_i, the angle of rotation from x_{i-1} to x_i around z_i.

In the case of setting the coordinate system, the corresponding connecting rod parameters are shown in the Table 1:

Table 1

Robot connecting rod parameters

link_i	a_{i-1} (mm)	α_{i-1} (°)	d_i (mm)	θ_i (°)	Joint variable range (°)
1	0	0	0	θ_1	-180~+180
2	97	-90	440	θ_2	-110~+100
3	705	0	0	θ_3	-65~+60
4	135	-90	264	θ_4	-200~+200
5	0	90	497	θ_5	-120~+120
6	0	-90	65	θ_6	-400~+400

2.2 Positive kinematics analysis

The forward kinematics of the robot is mainly aimed at solving the pose of the final position of the end effector of the robot according to the motion state of each joint of the robot [10]. In the process of analysis, it is necessary to find out the relative geometric state of each link, and then establish a coordinate system on each link, and use the frame state to represent the state of the link, so as to obtain the state description of the end of the robotic arm. The general formula of link transformation ${}^{i-1}_{i}T$ in the kinematics of robotic arm [11]:

$$ {}^{i-1}_{i}T = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1} $$

Use the general formula of the robot link parameters and link transformation ${}^{i-1}_{i}T$ in Table 1, list the transformation matrix of each link, and multiply each link transformation matrix to obtain the ABB-IRB2400 robot arm transformation matrix ${}^{0}_{6}T$:

$$ \begin{aligned} {}^{0}_{6}T &= {}^{0}_{1}T(\theta_1)\,{}^{1}_{2}T(\theta_2)\,{}^{2}_{3}T(\theta_3)\,{}^{3}_{4}T(\theta_4)\,{}^{4}_{5}T(\theta_5)\,{}^{5}_{6}T(\theta_6) \\ &= \begin{bmatrix} \vec{n} & \vec{o} & \vec{a} & \vec{p} \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{2} $$

Where:

$$ \begin{aligned}
n_x &= c_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}c_5c_6] + s_1(s_4c_5c_6 + c_4s_6) \\
n_y &= s_1[c_{23}(c_4c_5c_6 - s_4s_6) - s_{23}s_5c_6] - c_1(s_4c_5c_6 + c_4s_6) \\
n_z &= -s_{23}(c_4c_5c_6 - s_4s_6) - c_{23}s_5s_6 \\
o_x &= c_1[c_{23}(-c_4c_5c_6 - s_4s_6) + s_{23}s_5c_6] - s_1(s_4c_5c_6 - c_4s_6) \\
o_y &= s_1[c_{23}(-c_4c_5c_6 - s_4s_6) + s_{23}s_5c_6] + c_1(s_4c_5c_6 - c_4s_6) \\
o_z &= s_{23}(c_4c_5c_6 + s_4c_6) + c_{23}s_5s_6 \\
a_x &= -c_1(c_{23}c_4s_5 + s_{23}c_5) - s_1s_4s_5 \\
a_y &= -s_1(c_{23}c_4s_5 + s_{23}c_5) + c_1s_4s_5 \\
a_z &= s_{23}c_4s_5 - c_{23}c_5 \\
p_x &= c_1[c_{23}(d_5s_4 + a_3 - d_6c_4s_5) - s_{23}(d_6c_5 + d_4) + a_2c_2 + a_1] \\
&\quad - s_1(d_6s_4s_5 + d_5c_4 + d_2) \\
p_y &= s_1[c_{23}(d_5s_4 + a_3 - d_6c_4s_5) - s_{23}(d_6c_5 + d_4) + a_2c_2 + a_1] \\
&\quad + c_1(d_6s_4s_5 + d_5c_4 + d_2) \\
p_z &= s_{23}(d_6c_4s_5 - d_5s_4 - a_3) - c_{23}(d_6c_5 + d_4) - a_2s_2
\end{aligned} $$

Where c_1 means $cos\theta_1$, s_1 means $cos\theta_1$, the sine and cosine representations of other angles can be deduced by analogy, c_{23} means $\cos(\theta_2+\theta_3)$, s_{23} means $\sin(\theta_2+\theta_3)$. It can be seen that because the axes of the two rotary joints of the ABB-IRB2400 robot link 2 are parallel, a relatively simple expression can be obtained through the two angles sum formulas. The above equation represents the transformation matrix T of the ABB-IRB2400 manipulator arm, which describes the pose of the end effector's coordinate system {6} relative to the robot base coordinate system {0}, and is the basis for kinematic analysis and comprehensive application of the manipulator arm.

2.3 Inverse kinematics analysis

The inverse kinematics solution of robot is also called inverse kinematics solution, which means that the value of robot joint variables is determined according to the given position of end effector and the attitude of manipulator [12]. That is, when the position and attitude of the end effector of the robot are given, the $\vec{n}$, $\vec{o}$, $\vec{a}$ and $\vec{P}$ of the manipulator are all known conditions, and the process of reverse calculating the values of the joint variables θ_1, θ_2, θ_3, θ_4, θ_5 and θ_6 of the manipulator is the inverse solution of the robot kinematics. In this paper, the inverse solution of ABB-IRB2400 robot is obtained by using the inverse transformation method.

The method of solving the inverse transformation is to multiply the inverse of a certain transformation on both sides of the equation of the robot kinematics equation at the same time. The specific formula is expressed as follows.

$$ {}_{i}^{0}T^{-1}(\theta_i)\,{}_{6}^{0}T = {}_{i+1}^{i}T(\theta_{i+1})\,{}_{i+2}^{i+1}T(\theta_{i+2})\cdots{}_{6}^{5}T(\theta_6) \tag{3}$$

Since the posture of the end link has been given, that is, ${}_{6}^{0}T$ is known, the product of the inverse of each transformation on the left side of the equation and the transformation on the right side of the equation can be obtained. When solving θ_1 and θ_3, there are two solutions each, and the manipulator wrist (joints 4, 5, and 6) also has two different flip directions. Generally, there are 8 groups of inverse solutions, but in actual processing, only 1 group of inverse solutions can be selected for processing. In order to optimize the 8 groups of inverse solutions, in the joint motion space, the discarded solutions beyond the joint motion range are removed, and a group of inverse solutions that minimize the change of the robot joints is selected as the final inverse solution. First solve for θ_1, and multiply both sides of the kinematic equation with the inverse transform ${}_{1}^{0}T^{-1}(\theta_1)$:

$$ {}_{1}^{0}T^{-1}(\theta_1)\,{}_{6}^{0}T = {}_{2}^{1}T(\theta_2)\,{}_{3}^{2}T(\theta_3)\,{}_{4}^{3}T(\theta_4)\,{}_{5}^{4}T(\theta_5)\,{}_{6}^{5}T(\theta_6) = {}_{6}^{1}T \tag{4}$$

$$ \begin{bmatrix} c_1 & s_1 & 0 & 0 \\ -s_1 & c_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}_{6}^{1}T \tag{5}$$

The elements on both sides of the equation are equal, and the elements (2,1), (2,2), (2,3) and (2,4) are correspondingly equal, we can get:

$$-s_1 p_x + c_1 p_y = d_2 + d_5 c_4 - d_6 s_1 a_x + d_6 c_1 a_y \tag{6}$$

In the formula, in addition to θ_1 which is unknown, there is also θ_4 which is also unknown. Through the implicit function relationship between θ_4 and θ_1, it is assumed that the implicit function implies the θ_4 and θ_1 relationship of $c_4 = ms_1 + nc_1$. And use triangle substitution to get:

$$\theta_1 = \arctan k_1 - \arctan k_2 \tag{7}$$

$$\theta_4 = \arccos(ms_1 + mc_1) \tag{8}$$

By multiplying the kinematic equation by ${}^0_4T^{-1}$ to the left, the elements corresponding to (2,4) and (3,3) are equal, and using trigonometric transformation, we get:

$$\theta_2 = \arccos\frac{k_3}{\rho_2} - \arctan\frac{a_z s_4}{k_4} - \arcsin\frac{k_5}{a_2 s_4} \tag{9}$$

$$\theta_3 = \arcsin\frac{k_5}{a_2 s_4} \tag{10}$$

From the elements (1,3) and (3,3) correspondingly equal, we get:

$$\theta_5 = \arctan\frac{-a_x k_6 - a_y k_7 - a_z(s_{23}c_4)}{a_x(-c_1 s_{23}) + a_y(-s_1 s_{23}) + a_z(-c_{23})} \tag{11}$$

From the elements (1,2), (1,3) correspondingly equal, we get:

$$\theta_6 = \arctan\frac{n_x k_8 + n_y k_9 + n_z(s_{23}s_4)}{o_x k_8 + o_y k_9 + o_z(s_{23}s_4)} \tag{12}$$

Through the above formula, the inverse kinematics solution of each joint variable $\theta_1 \sim \theta_6$ of the manipulator will be completed under the given end effector pose.

3 Kinematics simulation analysis of robot arm

The relationship between the robot's end pose and each joint variable has been theoretically verified by the kinematic equation established above. Through the simulation and measurement functions of ADAMS, the solution of the kinematic equation can be simulated, analyzed and verified. The ABB-IRB2400 robot arm model was established in the SolidWorks software, and the model was imported into the ADAMS simulation analysis software to obtain the kinematic output displacement, velocity, and acceleration curves of the ABB-IRB2400 robot grinding and polishing workpieces [13], so as to verify the correctness and feasibility of the calculated motion equation.

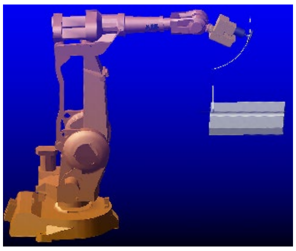
Constrain the robot arm model in ADAMS, and add driving parts, so that the robot arm can drive the end effector with the desired motion trajectory according to the established driving function, so that it can complete the required weld grinding and polishing work. Select the reference point at the center of the end grinding wheel, and set the reference point drive function as follows:

X direction: STEP(time, 4.0, 0.0, 8.0, 500.0)-STEP (time, 8.0, 0.0, 12.0, 600.0);

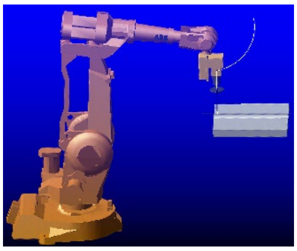
Z direction: STEP(time, 2.0, 0.0, 4.0, -314.0);

Rotate around Y axis: STEP(time, 0.0, 0.0, 2.0, 90.0)*1d.

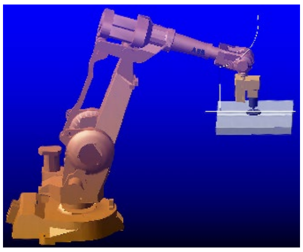
After the above steps, kinematics simulation can be performed on the 6-DOF industrial robot ABB-IRB2400 model, the simulation time is set to 12 seconds, and the number of simulation steps is 1200 steps. The simulation movement is shown in Figure 4 below.



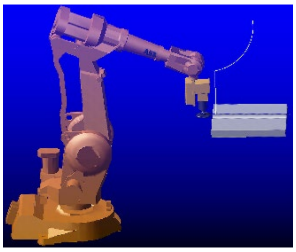
(a) The first stage of robot movement



(b) The second stage of robot movement



(c) The third stage of robot movement



(d) The fourth stage of robot movement

Figure 4 Robot motion trajectory

Use the Postprocessing module in ADAMS software to analyze and process the curve of the robot's motion trajectory, directly generate the curve function of the manipulator model when it moves in the simulation software, and display it in the form of a curve chart, according to the simulation grinding and polishing trajectory In the grinding and polishing operation of the workpiece, the robot arm mainly relies on the movement of joints 2, 3, and 5 to complete a grinding and polishing of the weld. The following Figure 5 shows the changes in the angular displacement, angular velocity and angular acceleration of the three main joints with time Simulation results.

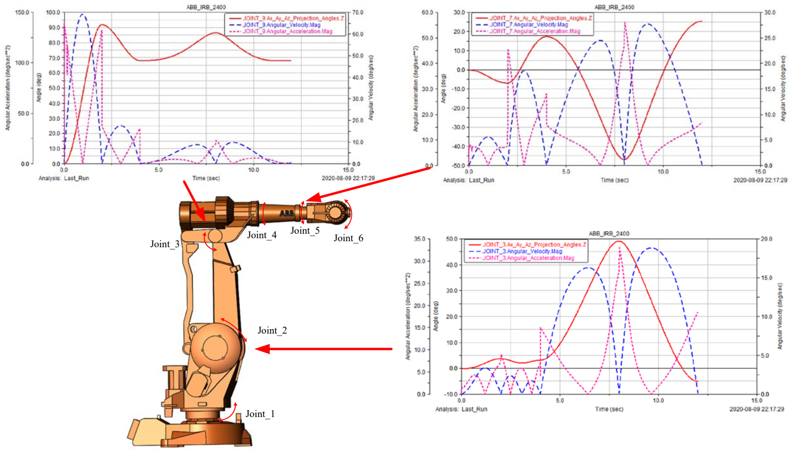


Fig. 5 Angular displacement, angular velocity and angular acceleration of the three main joints

From the above change curves of joint angular displacement, joint angular velocity, and joint angular acceleration, it can be seen that during the entire simulation movement process, the angle changes, angular velocity changes and angular acceleration changes of each joint of the robot are continuous, and there is no sudden change. It shows that the motion process of the robot is stable, and there is no severe jitter and dislocation, which shows that the kinematic analysis and solution results of the robot are reasonable and feasible, and lays a theoretical foundation for the planning and control of the motion trajectory of the robot arm.

4 Robotic arm motion trajectory planning

MoveIt is a motion planning library that integrates path planning, kinematics solving, collision detection and other component packages related to mobile operations in ROS (Robot Operating System). It provides a variety of dynamics solvers and motion planning algorithms [14]. When the manipulator is grinding, we hope that the manipulator can generate specific trajectories for different situations. When using MoveIt in ROS for motion planning, we only need to specify the target position for the manipulator, and the manipulator will automatically plan according to the current scene. Create a path to avoid obstacles, and make further smooth modifications to the planned path.

In MoveIt, various algorithms for robot motion planning are implemented by motion planners. There are various motion planning algorithms with different focuses. Different planning algorithms can be selected according to requirements. In this paper, kinematics solution and rapid expansion of random tree-RRT motion planning algorithm are used to realize linear interpolation trajectory planning and arc interpolation trajectory planning for the joint space and Cartesian space of the manipulator [15]. And through the ROS three-dimensional visualization platform RVIZ simulation to verify the results of the algorithm to achieve trajectory planning.

4.1 RRT trajectory planning algorithm

The trajectory planning of the RRT algorithm [16] in the two-dimensional space is based on a tree-shaped exploration method for interpolation, which can be visually regarded as a “tree-shaped algorithm”. Starting from the initial position, the tree data is continuously extended, random sampling is performed in the complex constraint space, and the path is expanded until the “branch” covers the entire area to complete the planning. Among the multiple interpolation paths planned by the algorithm, the shortest path is selected as the output result. The idea of interpolation planning is as follows:

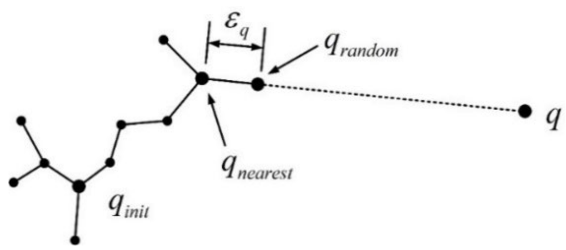


Fig. 6 RRT algorithm principle diagram

(1) $T_n(1,2,3,...,n)$ is expressed as a random tree of n nodes in the trajectory planning space, the initial root node of the T_n random tree is expressed as q_{init}, the random sampling node of the tree is expressed as q_{random}, and $dis(q_i,q_{i+1})$ is expressed between any two adjacent nodes in the tree, ε_q is the search step length of the tree;

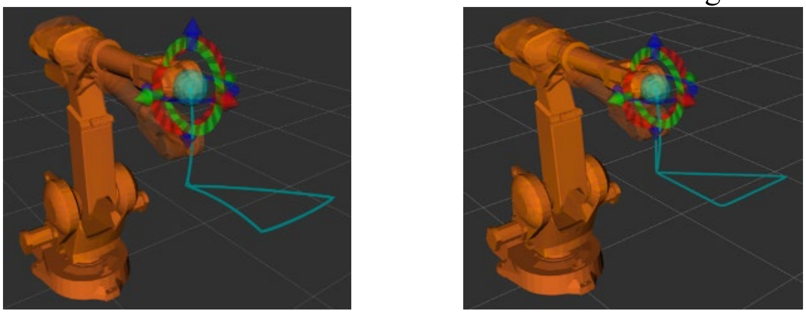
(2) Find the node closest to $q_{nearest}$ on the T_n random tree, expressed as q_{random}, connect $q_{nearest}$ and q_{random}, and judge whether the two points will collide with obstacles;

(3) If a collision occurs, the connection is abandoned, and no collision occurs, the relationship between the distance between the two points and the ε_q search step is judged. The distance is less than the step ε_q, then q_{random} is a new node, and the distance is greater than ε_q, then restart on the connecting line of node $q_{nearest}$ and q_{random}, select the point whose distance from $q_{nearest}$ is the step

length ε_q as the new node to complete a spatial search expansion, and repeat the above process until the exploration reaches the target position. Figure 6 shows the process of RRT exploring the path.

4.2 ROS-based trajectory planning results

Through MoveIt's dynamics solving algorithm and RRT motion planning algorithm, the joint space and Cartesian space trajectory planning nodes are programmed for the entire path of the robotic arm grinding and polishing, and the different nodes of ROS are activated. The results are shown in Figures 7 and 8.



(a) Joint space trajectory planning (b) Cartesian space trajectory planning

Fig. 7 Linear interpolation trajectory planning in different spaces based on RRT algorithm

Figure 7 (a) and (b) are the results of linear interpolation trajectory planning of the robot working path in different spaces based on the RRT algorithm. The trajectory planning path of the robot arm in the joint space shows that the end effector of the robot arm moves 0.4m down the Z axis of the end coordinate system from the initial pose and 0.4m along the positive direction of the Y axis. During the whole planning process, the two spatial points are constrained, and the kinematics solver can be used to solve the joint angle change of each joint of the manipulator at the two spatial constraint points. The fifth-order polynomial method is used to plan the motion trajectory of the manipulator's working path. The path planning result is shown in Figure 7(a) above. In order to ensure the continuity and smoothness of the angular acceleration of the joints of the manipulator, the planned path is generally curve. The result of trajectory planning in the Cartesian coordinate system is the same as in the joint space. The end effector first moves down 0.4m along the Z axis and then 0.4m along the positive direction of the Y axis from the initial state. When trajectory planning is performed in Cartesian space, in order to ensure that the path between the two spatial constraint points is a straight line, the RRT algorithm is used to interpolate the constraint conditions of the straight line.

In the process of tracking and controlling surface welds, arc trajectory planning in Cartesian space is required. Figure 8 below is a perfect arc path obtained by arc planning in Cartesian space using RRT algorithm. In the whole control process of the welding seam grinding of the robotic arm, from the initial test position approaching the welding seam and returning to the initial position after finishing the grinding and polishing work, the joint space planning using the kinematics solver is adopted, and several planning methods are used. The combination can complete all the motion path planning required for the entire grinding and polishing operation of the robot.

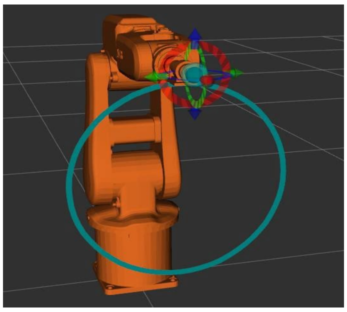


Fig. 8 Arc interpolation trajectory planning based on RRT algorithm

When the robot arm is planning the trajectory in the joint space, during the movement between the three points of the planning path, the position of each joint of the robot arm and the speed and acceleration of the motion are stipulated. The robot arm is carried out in Cartesian space. When planning, the path trajectory of the end effector is curved. In Cartesian space, the straight-line travel path between points in the path is specified. 300 interpolation points are planned for the straight-line trajectory, and they are solved. In the arc trajectory planning, 628 interpolation points are planned and arced. The trajectory is solved by interpolation. The result is shown in the figure above. The kinematics solver and the RRT algorithm have completed the calculation of all trajectory planning points, verifying that the algorithm can realize the spatial path planning of the robotic arm.

5. Conclusions

This paper mainly analyzes the kinematics of the ABB-IRB2400 robot, studies the forward and inverse solutions of the robot's working pose, and uses related algorithms to complete the trajectory planning work. The main conclusions are as follows:

(1) Through the model parameters of the ABB-IRB2400 grinding and polishing robot, the D-H parameter method was used to discuss the displacement relationship between the manipulator and each link. The manipulator model was established, and the forward and inverse solutions of the manipulator's motion process was studied using kinematic equations.

(2) The kinematics analysis of the ABB-IRB2400 robot was carried out based on ADAMS software, and the relationship between the joint angle and the end pose at any time was obtained, and the verification of the kinematic equation solution was completed, which laid a theoretical foundation for trajectory planning.

(3) Using the kinematics solver and RRT motion planning algorithm, the grinding and polishing robot ABB-IRB2400 was planned in the joint space and the Cartesian space for linear and arc path planning, and the combination can complete the entire grinding and polishing work route plan.

(4) Using ROS visualization tool RVIZ for simulation, the applicability and feasibility of the related trajectory algorithm in the motion planning of the manipulator was verified by the success rate of the solution of the planning points in the motion trajectory, which provides a basis for the grinding and polishing work under complex working conditions.

In future work, in-depth research on robot kinematics and dynamics can be carried out. Research on the robot kinematics solver algorithm to improve the solution rate and solution time in the robot workspace. Therefore, optimizing the inverse kinematics solution algorithm will be the focus of future research and improvement.

Acknowledgement

The work was funded by the National Natural Science Foundation of China(62073239).

R E F E R E N C E S

[1] *Daxian Hao*, *Wei Wang*, *Qilong Wang*, et al. “Applications and Development Trend of Robotics in Composite Materials Process”. Journal of Mechanical Engineering, **vol. 5**, no. 3, Feb. 2019, pp. 1-17.

[2] *Zhihui Gao*, *Xiaodong Lan*, *Yushu Bian*. “Structural Dimension Optimization of Robotic Belt Grinding System for Grinding Workpieces with Complex Shaped Surfaces Based on Dexterity Grinding Space”. Chinese Journal of Aeronautics, **vol. 24**, no. 3, Jun. 2011, pp. 346-354.

[3] *C. Forbrigger*, *R. Bauer*, *A. Warkentin*. "Improving the accuracy of a profile grinding wheel grooving robot attachment for a CNC grinding machine". International Journal of Advanced Manufacturing Technology, **vol. 98**, no. 9-12, Oct. 2018, pp. 3205-3216.
[4] *Shiming Ji*, *Xihuan Huang*. "Review of development and application of industrial robot technology". Journal of Mechanical & Electrical Engineering, **vol. 32**, no. 1, Jan. 2015, pp. 1-13.
[5] *M.C. Lee*, *S.J. Go*, *M.H. Lee*, et al. "A robust trajectory tracking control of a polishing robot system based on CAM data". Robotics and Computer-Integrated Manufacturing, **vol. 17**, no. 1-2, Jun. 2000, pp. 177-183.
[6] *Qiang Chen*, *Jingrong Lu*, *Zhenguo Sun*, et al. "Research on torch attitude planning system of arc welding robot". China Mechanical Engineering, **vol. 13**, no. 11, Jun. 2002, pp. 956-958.
[7] *Denglin Zhu*, *Sheng Xiao*, *Hudi Hou*, et al. "Automatic planning of welding torch's gesture of arc welding robot". Journal of Shanghai Jiaotong University, **vol. 38**, no. 9, Sep. 2004, pp. 1472-1475.
[8] *H.C. Fang*, *S.K Ong*, *A.Y.C. Nee*. "Robot path planning optimization for welding complex joints". International Journal of Advanced Manufacturing Technology, **vol. 90**, no. 9-12, Jun. 2017, pp. 3829-3839.
[9] *Yi Gan*, *Junlei Wang*, *Fujia Sun*. "Optimal Design of D-H Parameters of a 6R Robot for a Prescribed Workspace". China Mechanical Engineering, **vol. 25**, no. 22, Nov. 2014, pp. 3003-3007.
[10] *Yanfei Zhang*, *Jinliang Gong*, *Weimin Li*, et al. "Kinematics Analysis and Simulation of a 6-DOF Redundant Driven Parallel Robot". Chinese Journal of Mechanical Engineering, **vol. 41**, no. 8, Aug. 2005, pp. 144-148.
[11] *J.J. Craig*. Introduction to robotics. China Machine Press, Beijing, 2006.
[12] *J.D. Rubio*, *V. Aquino*, *M. Figueroa*. "Inverse kinematics of a mobile robot". Neural Computer & Applications, **vol. 23**, no. 1, Jul. 2013, pp. 187-194.
[13] *Ani Luo*, *Jiatai Zhang*, *Heping Liu*. "Simulation of five degree of freedom manipulator using ADAMS". Computer Simulation, **vol. 22**, no. 7, Jul. 2005, pp. 201-203.
[14] *G. Diaz-Arango, H. Vazquez-Leal, L. Hernandez-Martinez, et al.* "Homotopy Path Planning for Terrestrial Robots Using Spherical Algorithm". IEEE Transactions on Automation Science and Engineering, **vol. 15**, no. 2, Apr. 2018, pp. 567-585.
[15] *Bo Pan*, *Yili Fu*, *Zongpeng Yang*, et al. "Efficient inverse kinematics solution for redundant robot to real-time control". Control and Decision, **vol. 24**, no. 2, Feb. 2009, pp. 176-180.
[16] *Mingbo Du*, *Mei Tao*, *Jiajia Chen*, et al. "RRT-based motion planning algorithm for intelligent vehicle in complex environments". Robot, **vol. 37**, no. 4, Jul. 2015, pp. 443-450.