Research on Path Calculation and Simulation System of Variable Parameter Manipulator

Jiacheng Li, Xiaoli Yang, Xian Wu and Liangwei Zhong*

Mechanical Engineering, School of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, Shanghai, 200093, China

*Corresponding author’s email: zlvcad@126.com

Abstract. The simulation software of manipulator is generally developed by manufacturers, and the products are only suitable for the specific company, but cannot be used in universal. Aiming at this problem, this paper establishes a mathematical model with attributes through parametric design. The standard D-H parameter method is used to model the kinematics of the manipulator. Based on the expert system, the intelligent path planning of the manipulator is carried out, and the trajectory control instructions of the manipulator are generated. According to the Monte Carlo method of point cloud, the workspace of manipulator is obtained, and the bounding box technology is used for real-time collision detection. Taking the welding work as an example, the parametric modeling and trajectory planning simulation of the manipulator are realized, and a general off-line programming platform is established.

1. Introduction

At present, mechanical arm industrial robot has been widely used in industrial production, handling, processing, welding and other fields [1]. There are many programming methods for industrial robots, among which off-line programming is more flexible and efficient, and can be combined with CAD/CAM system [2]. However, due to the current robot programming has not established a recognized international standard, each manufacturer has its own programming language, cannot be universal [3]. This is a huge gap and in order to solve this problem, this paper establishes a mathematical model with attributes through parametric modeling. The standard D-H parameter method is used to model the kinematics of the manipulator. The intelligent path planning of the manipulator is carried out based on the expert system, and the trajectory control instructions of the manipulator are generated. The workspace of the manipulator is obtained according to the Monte Carlo method of point cloud, and the bounding box technology is used for real-time collision detection. The simulation system designed in this paper realizes the parametric modeling and trajectory planning simulation of the manipulator, and establishes a general off-line programming platform. The flow of the entire system is shown in figure 1.
2. Parametric Modelling

The six-degree-of-freedom manipulator has a five-segment structure. In order to adapt to the versatility, the manipulator is parameterized and modelled, and the dimensions of each segment are input, then the manipulator automatically generates a numerical model with attributes. Taking the PUMA560 robot as an example, the parametric model is established as shown in figure 2.

\[
\begin{bmatrix}
    c\theta_i & -s\theta_i & 0 & \alpha_{i-1} \\
    s\theta_i c\alpha_{i-1} & c\theta_i c\alpha_{i-1} & -s\alpha_{i-1} & -d_i s\alpha_{i-1} \\
    s\theta_i s\alpha_{i-1} & c\theta_i s\alpha_{i-1} & c\alpha_{i-1} & d_i c\alpha_{i-1} \\
    0 & 0 & 0 & 1
\end{bmatrix}
\]

(1)
Where the $i^{-1}T_i$ represents the homogeneous transformation matrix of the link coordinate system $i-1$ related to the link coordinate system $i$, $c$ represents the cosine function, $s$ represents the sine function, $\theta_i$ represents the angle between the two links, $a_{i-1}$ represents the distance between the two axes, and $d_i$ represents the relative position of two links.

According to the D-H method, the D-H parameters of the six-degree-of-freedom manipulator are determined in Table 1. The D-H parameters are often different due to different geometry parameters and different coordinate systems.

**Table 1. DH parameters of the mechanical arm.**

| Link $i$ | $\theta_i$ | $a_{i-1}$ | $a_i$ | $d_i$ |
|---------|------------|-----------|-------|-------|
| Link1   | $\theta_1$ | $0^\circ$ | 0     | 0     |
| Link2   | $\theta_2$ | $-90^\circ$ | 0     | $d_2$ |
| Link3   | $\theta_3$ | $0^\circ$ | $a_2$ | 0     |
| Link4   | $\theta_4$ | $-90^\circ$ | $a_3$ | $d_4$ |
| Link5   | $\theta_5$ | $90^\circ$ | 0     | 0     |
| Link6   | $\theta_6$ | $-90^\circ$ | 0     | 0     |

By inputting the motion path of the manipulator and inverse solving the kinematics model established by D-H parameters, the rotation angles of each joint can be obtained. The trajectory control command of the manipulator is generated by programming, that is, the rotation of each joint makes the end effector track coincide with the path. The flow chart of instruction parser algorithm is shown in Figure 3.

**Figure 3. Algorithm flow chart of instruction parser.**

According to the two-point spatial coordinates, the algorithm searches the posture information that satisfies the error range in the knowledge base, and calculates the corresponding joint rotation through the inverse kinematics solution and sorts them. The program will select a group of data with the smallest total rotation for collision detection. If there is a collision, the next smallest data will be used until there’s no collision. The parser written by coordinate transformation and rotation instructions is used to simulate the group of data.

### 3. Intelligent Path Planning

The path of the robotic arm is the motion trajectory of the end effector. Since the path points only contain position information and no posture information, it is difficult to establish a pose matrix...
through calculation. Especially for the welding robot arm, the welding process does not have the optimal posture or the optimal posture is just a range. It is necessary to adopt multiple optimization methods to establish and optimize the end position constraints (hard constraints, strong punishments) and attitude constraints (relatively weak punishments). Moreover, CAD mainly deals with numerical calculation and drawing graphics, but is powerless in the selection and formulation of path planning, evaluation and decision-making. Therefore, this article introduces the expert system into the robotic arm path planning. The simplified block diagram of the robotic arm planning expert system is shown in Figure 4.

**Figure 4. Robot arm planning expert system’s simplified block diagram.**

First input the robot arm model or basic parameters to search or update the database, and at the same time obtain the pose matrix of the end effector of the robot arm through forward kinematics; then obtain the trajectory of the end effector, and the posture information of the database is calculated by using the principle of inverse kinematics and stored in the knowledge base. In each sampling period, the inference engine makes inferences based on the current facts (system state) and knowledge base, then generates control commands and adds them to the controlled object. That is, after the path is recognized, the space point coordinates are obtained through interpolation, then enter the knowledge base for searching, matching and backtracking, so as to control the manipulator to rotate to the corresponding position. The knowledge base stores expert experience and knowledge, which is expressed in the form of production rules as follows:

\[
\text{IF } < \text{Condition} > \text{THEN } < \text{Result} >
\]

The knowledge base is established through forward kinematics, which includes the model of the manipulator, the basic parameters, the spatial coordinates of the end effector, the corresponding position and posture matrix and the rotation angle of each joint.

In an expert control system, the minimum sampling period allowed by the system is basically determined by the inference speed. The faster the running speed of the expert system, the shorter the minimum sampling period, and the wider the scope of application [5]. When design the structure of the knowledge base, the reasoning time for searching the available knowledge in the knowledge base should be fully considered. This paper adopts a layered strategy to build a knowledge base, and each layer includes the corresponding knowledge at that level. The higher the level, the more refined the knowledge. When reasoning, search the corresponding sub-library according to the level of knowledge instead of searching the entire knowledge base, which narrows the search scope, thus shortens the search time and improves the efficiency. That is, the Cartesian coordinate system is divided into 8 quadrants, and the collected point coordinates are first judged which coordinate system belongs to, and then the corresponding knowledge base is searched.

4. **Workspace Calculation Based on Monte Carlo Method**

The working space of the robot arm is the maximum space that can be reached during normal operation, denoted as \( W(P) \) [6]. It can be regarded as the mapping of joint space variables and working space, which can be expressed as

\[
W(P) = \{ P(q) : q \in Q \} \subset R^3
\]
Where $P(q)$ is the position component of the positive solution of the kinematics mapping, $q$ is the generalized joint variable, $Q$ is the joint space, $W(P)$ is the working space, $R^3$ is the three-dimensional space.

Formula (3) represents the principle of the working space based on Monte Carlo method. Using the Monte Carlo method to calculate the working space of the multi-joint manipulator is actually assigning a fixed number of random variables that satisfy the requirements of the joint change through a uniform distribution of the joint variables, and combining the joint variables to calculates the coordinate value of the end effector of the manipulator by using the forward kinematics equation, and these coordinate values form the working space of the manipulator. The more randomly selected combinations of joint variables, the more the number of coordinate values calculated, the more accurately the actual working space can be reflected [7]. The calculation formula of the variables of each joint $\theta_{ij}$ at different moments can be expressed as

$$\theta_{ij} = \theta_{i}^{\text{min}} \cdot (\theta_{i}^{\text{max}} - \theta_{i}^{\text{min}}) \cdot \delta_{ij} (i = 1, 2, 3, 4, 5, 6; \ j = 1, 2, \ldots, n)$$

Where $\theta_{i}^{\text{min}}$ is the minimum value of each joint variable, $\theta_{i}^{\text{max}}$ is the maximum value of each joint variable, and $\delta_{ij}$ is the first random discrete variable selected arbitrarily by the joint or link $i$, with the value range $[0, 1]$. The generalized joint variables $\theta = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]$ are introduced into the forward kinematics equation to obtain the end effector position vector $p = [Px, Py, Pz]$.

Write a function to generate random numbers $\delta_{ij}$, and use the built-in point command in Auto CAD to draw the working space of PUMA560, as shown in Figure 5.

![Figure 5. The workspace of PUMA560 robot.](image)

5. Collision Detection

Collision detection is to determine whether there is a collision or interference between every part of the robot arm and obstacles. In this paper, the robot arm linkage and obstacle collision detection are carried out in the workspace. Due to the complex structure of the robot arm and the irregular shape of the obstacle, it is difficult to describe it with a mathematical model, so it is represented by a geometric model. The most commonly used method is the bounding box technique [8].

The basic idea of the bounding box technology is to use a slightly larger and simple geometric body (called a bounding box) approximately replace complex geometric objects. Commonly used methods are: AABB (Aligned Axis Bounding Box) bounding box, cylindrical bounding box, bounding ball, OBB (Oriented Bounding Box) bounding box model, etc. [9].

Since each link of the manipulator arm is similar to the shape of a cylinder, in order to simplify the model without losing its structural characteristics as much as possible, this article uses a cylinder to envelop each link of the manipulator. Due to the advantages of the simple calculation of the enclosing sphere model and the minimum storage space required, this paper uses enclosing sphere technology to represent obstacles. Through the combination of the cylindrical bounding box and the bounding sphere model, the detection of collision between the robot arm and the obstacle is transformed into the
calculation of the distance between the center line of the connecting rod and the center of the obstacle, as shown in Figure 6.

![Figure 6. The Relationship Between Center Lines of Links and the Center Point of Obstacles.](image)

Where the red dotted line $L_i$ is the center line of the connecting rod of the robot arm, $J_i$ is the joint, $O$ is the center of the obstacle, and $R_o$ is the radius of the ball envelope. Set the distance from the center of the obstacle to the center line of the connecting rod as $d_i$. The radial radius is $R_i$, the safety distance is $S$, at that time $S \leq d_i$, the robot arm will not collide with the obstacle.

Using the Line command in Auto CAD to draw the center line of the connecting rod, and then obtain the primitive information in the graphics database to obtain the vector of the center line of the two connecting rods $\vec{l}_i$. The distance $\vec{d}_i$ from the center of the obstacle bounding box to the center line of the connecting rod as the collision detection condition used in formula (4).

$$S \leq d_i = \frac{\|\vec{d}_i \times \vec{l}_i\|}{\|\vec{l}_i\|}$$  \hspace{1cm} (4)

Taking welding work as an example, this article selects a random path in the work space, and randomly creates the position and size of the part. The collision detection in the robot arm simulation system is shown in Figure 7.

![Figure 7. Collision detection in simulation process.](image)

The robot arm can perform collision detection during movement. In Figure 7(a), the robotic arm does not collide with obstacles during operation. As shown in figure 7(b), if a collision occurs, the system will issue a warning. If there are multiple consecutive collisions, the system will issue a warning of damage to the robot arm and force the simulation to stop, see figure 7(c). If there is no collision in the process, the robot arm will travel along the path to the end, and the system will prompt that the simulation is successful.

6. Conclusion
This paper uses D-H parameters to establish a kinematics model of the robot arm, realizes the command analysis of the robotic arm through programming and generates the path of the robotic arm based on the expert system. The point cloud based on Monte Carlo method is used to obtain the workspace of the six-degree-of-freedom manipulator, the bounding box technology is used to realize
the collision detection, and the welding work is taken as an example for demonstration. The parametric modeling and trajectory planning simulation of the robotic arm are realized, and a general offline programming platform is established by using secondary development technology in the Auto CAD system. The simulation process of this platform is intuitive, the collision detection during the movement can be fed back in real time, and it provides a good interface of users and computer. The working principle, workspace, kinematics and inverse kinematics equations of the robot arm can be verified by this platform. The level of various control algorithms can also be compared by debugging the parameters of the manipulator, which can be used as a tool for the design and theoretical study of the manipulator.

References
[1] Gao Sheng, Li Zheng, Zhang Wei. (2020) Automatic Modelling Simulation and 3D Navigation Planning of Motion Trajectory for Manipulator. J. Computer Simulation, 36:1–7.
[2] Li Yuhui. (2016) Design and Implement of General Graphical Programming Software for Industrial Robot. D. Zhejiang University.
[3] Yang S, Mao X, Ge B, et al. The Roadmap and Challenges of Robot Programming Languages. C. Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE, 2015.
[4] Cai Zixin, (2000) Robotics. Tsinghua University Press, Beijing.
[5] Shen Huanhuan, Deng Xiaohong, Yu Yong, et al. (2008) Application of Intelligent Control Strategy Arm Wrestling Robot. J. Automation & Instrumentation, 36:5–8.
[6] Lin Zhiqiu, Luo Hongbo, Zhao Kang, et al. (2018) Workspace Analysis and Simulation of Hydraulic Drill Jumbo Boom. J. Machinery Design & Manufacture, 41:165–167.
[7] Ma Yuhao. (2019) The Research on Obstacle Avoidance Trajectory Planning and Control Algorithms of A 6-DOF Manipulator. D. Xian. University of Chinese Academy of Sciences.
[8] He Zhifeng, Zhu Jianmin. Kinematical Modelling and Trajectory Tracking control Simulation of Minimally Inverse Surgery Robot. D. Shanghai. University of Shanghai for Science and Technology.
[9] Zou Yisheng. Survey on Real-time Collision Detection Algorithms Application Research of Computers, 37:8–12.