Trajectory Planning of 7-Degree-of-Freedom Manipulator Based on ROS

Xiangxin Xu1*, Mihai Duguleana2*

1School of Shenyang Jianzhu University, Liaoning, China
2School of Transilvania University, Brasov, Romania

*Corresponding author e-mail: 2318373340@qq.com, *dugublack@gmail.com

Abstract. In order to solve the problems of complex modeling of multi-degree-of-freedom redundant robot, difficult motion trajectory planning, tedious operation of traditional robot simulation tools and incomplete functions, this paper obtains a trajectory planning method under ROS system through the research of open source robot operating system ROS (Robot Operating system), and realizes the trajectory planning of seven-degree-of-freedom manipulator Cyton Gamma 300. Get the package of the target robot through the GitHub website and save it to the corresponding path, display the model of the manipulator in the RViz, and use C++ related interface provided by Moveit! To simulate the straight line and circular arc trajectory planning of Cyton Gamma 300 manipulator in Cartesian space on this platform. The experimental results show that the joints of the manipulator move smoothly and the accuracy of the end position meets the requirements.

1. Introduction

With the rapid development of global economy and science and technology in the 21st century, the field of information technology presents great development momentum and prospect. With the rise of artificial intelligence, people not only enjoy the convenience of robots, but also begin to pay attention to the ethical problems involved behind them. However, with the continuous progress of science and technology, some sharp problems are gradually overcome and replaced, robots will also be better used, so as to benefit mankind.

Nowadays, the robot industry is developing rapidly, with many kinds and functions, and the degree of intelligence is getting higher and higher, which makes the control of robot more difficult, the operating system of robot more complex and the software of robot. The compilation is very tedious, a lot of redundant code, complex modeling, and the interaction is not ideal, the function level is not clear, the simulation universality is not high and so on, and all kinds of problems make the research of the robot more difficult. The ROS robot operating system used in this paper can solve these problems to a certain extent and provide a simplified and practical robot research platform for researchers. In the open source environment of ROS, various types of robots can be quickly modeled, simulated, analyzed and controlled by using the various tools and plug-ins it provides, and a large number of sample code and open source programs can be used to easily complete robot programming and Control the task. Compared with the powerful data processing, visual drawing ability and complete algorithm library of
Matlab, ROS shows the characteristics of flexible simulation operation and convenient human-computer interaction without losing its function.

In this paper, the trajectory planning and analysis of seven-degree-of-freedom redundant manipulator Cyton Gamma 300 is carried out by using ROS simulation platform, and the simulation results are more accurate, which lays a foundation for the further study of redundant manipulator.

2. Robot operating system
Ubuntu is an open source operating system based on Debian GNU/Linux, which is based on desktop applications. Ubuntu is built by (Canonical Ltd), a global professional development team. Ubuntu is designed to provide users with an up-to-date, stable operating system built mainly by free software. Ubuntu is very suitable for Intel's hyperpolar positioning, supporting x86, 64-bit and ppc architecture. Ubuntu has a huge community power and announced its development into the smartphone industry in 2013.

ROS (robot operating system) is an open source robot operating system based on Ubuntu, which is launched by Willow Garage Company in California. Because it has many functions like Windows, Linux system, such as bottom driver management, hardware abstract description, data interaction between programs, device independent selection, information message processing and so on, it can also be regarded as a secondary operating system under Ubuntu. Ros provides a series of toolkits and databases for us to use and compile. Ros is constructed the construction of robot control system is very convenient, with the characteristics of high compatibility, open source and complete algorithm. It is one of the mainstream robot development platforms at present, and is now maintained and operated by the Open Source Robotics Foundation (OSRF).

ROS's 3D visualization feature package Rviz, manipulator motion control function package Moveit! These tools are practical and powerful, which greatly reduce the complexity of robot research and improve the working efficiency. Gazebo robot simulation engine is seamlessly compatible with ROS system, and can realize the planning and control simulation of robot in complex environment. Through the joint simulation of ROS and Gazebo, the working condition of manipulator in real environment can be accurately simulated. Reliable statics and dynamics data are obtained. Universality of tool Kit and editing language in Ros, Point-to-Point Design High efficiency and good user interface interaction, so that it is easier to be accepted and used.

The architecture of ROS system can be divided into three parts: file system level, computing graph level and open source community level, each of which represents a level. The file system level is used to show the internal structure of ROS, folder structure, as well as the core files needed for work. The calculation graph level reflects the communication between processes and systems, where each sub-unit will provide data to superiors, and the calculation graph level also introduces the concepts and functions of ROS, including the establishment of systems, processing of various processes, communication with multiple computers, and so on. The open source community level mainly involves ROS resources, including the communication between users, the improvement and updating of resource packages and the system. Maintenance, which makes ROS in the continuous development.

3. Robot model establishment
3.1. Cyton Gamma 300 manipulator
In this paper, the Cyton Gamma 300 manipulator is a seven-degree-of-freedom redundant manipulator produced by ROBAI Company of the United States, which consists of a 7-degree-of-freedom arm and a terminal control device, such as claw, as shown in figure 1. The seven independent rotating joints on the arm ensure that they can avoid obstacles, reduce vibration, and optimize the path and so on under the premise of constant position and posture at the end of the arm. Redundancy of 1 greatly improves its flexibility, so that it is called humanoid manipulator and can perform various movements similar to that of human arm. Hand claw can complete the simple grasp, release, rotation and so on in its reachable space. Behind the manipulator's claws and joints is a steering gear, eight of which are used to control
the movement of each unit, and operators program the steering gear to move the manipulator. In 2016, the Cyton Gamma 300 manipulator, printed with thermoplastic material 3D by Robai, a robotics expert at Stratasys, Massachusetts, was installed on the International Space Station.

Figure 1. Cyton Gamma 300 manipulator.

3.2. ROS Modeling
To construct a robot model in ROS, first, we need to write the description file (urdf) of the target robot studied. The urdf file is the unified description line of the robot physical model in the ROS system. Urdf can be obtained by first drawing the robot's structural model in solidworks, then using the sw2urdf plug-in to export urdf file.

When a description file is obtained, it is opened for the related configuration in the configuration assistant of Moveit, as shown in Figure 2. Moveit! Is the most advanced software for mobile operation in ROS platform? It combines the latest achievements of motion planning, manipulation, three-dimensional perception, kinematics, control and navigation, and provides an easy-to-use platform for us to develop advanced robot applications and evaluate new robot design and building integrated robot products, its structural framework is shown in figure 3. Because of its friendly interactive interface and convenient user experience, Moveit! Has been widely used in industry, commerce, research and other fields, and has been used in most known models of robot research.

Figure 2. Configuration process.
After the configuration assistant is configured, the file will generate the configuration file (config), import it into the 3D visualization tool RViz, the latter will generate the scene and visually output the robot. RViz is a graphical tool provided with ROS. It can easily operate the program of ROS. Here, the robot model built with code is converted into visual 3D model. In RViz, the robot is set up to interact with the planner, so as to achieve the simulation purpose of path planning and trajectory optimization. The overall flow of ROS modeling is shown in Fig. 4.

3.3. Cyton Gamma 300 manipulator model acquisition
In this paper, the Cyton Gamma 300 manipulator is taken as the research object, because it is a type of known mainstream manipulator, so in order to simplify the work, we can download the relevant ROS code package of the Cyton Gamma 300 manipulator directly from the GitHub website, as shown in figure 5.
GitHub is a code hosting website based on Git, which is similar to a supporting file package library of ROS, which contains the file packages of various types of robots, which can be used for reference by scholars. Developers can open source the edited code on GitHub, browse the code of other projects, quote to your own name to make changes, copy it back to use locally (no access except), and initiate a request to submit their own changes upstream.

Download the files related to the Cyton Gamma300 robot arm to the local location and move to the corresponding path. Open the terminal, launch the launch file, and open the visualizer RViz. In the environment of RViz generation, we can see the model of Cyton Gamma 300, as shown in Fig. 6. After setting the corresponding parameters, you can drag the arm pawl with the left mouse button to make the arm move as planned and make some simple trajectory planning.

The manipulator moves in the RViz scene according to the RRT algorithm in the OMPL library. OMPL is a sampling-based motion planning library, which is mostly used to solve the path planning problem under constraints, and the fast extended random tree algorithm (RRT) is one of the most effective algorithms. The fast extended random tree algorithm (RRT) is aimed at the motion planning of manipulator, which has the advantages of simple principle and high applicability, the disadvantage is that the path is of poor quality. The principle of RRT algorithm is that the initial point is first selected as the root node, and then random sampling is carried out from the root node to expand the child nodes, and each node is connected to generate a random node. When the target point is also included in the child node of the extension tree, a path can be found in the random tree, which connects the initial point with the target point. The demonstration of RRT algorithm in three-dimensional space is shown in figure 7. (The left side is the initial point and the right side is the target point)

Figure 5. Cyton Gamma 300 code package in the GitHub Web site.
4. Cartesian spatial trajectory planning
In the cartesian space, the obtained Cyton Gamma 300 type seven-degree-of-freedom mechanical arm is respectively subjected to linear and circular arc track planning, so that the fast and smooth movement of the robot can be realized under the condition that the accuracy requirement is met. Moveit is used here to do something as Inverse kinematic analysis, kinematic algorithm, planning and controller for motion planning. As also shown in Figure 3, the core node of Moveit, move_group can interact with the user using the action and service of the ROS. The robot information can be received from the URDF file and the configuration file, and the status and environment information of the robot can be fed back through the ROS action and topic.

4.1. Linear trajectory planning
Make the end of the manipulator move from the initial point $P_1(X_1,Y_1,Z_1)$ to the end point $P_2(X_2,Y_2,Z_2)$ along the straight line. The distance between the two points is $d$, the terminal velocity is $v$, and the
time required is $T$. At any time $t$, the interpolation coordinate $(X_t, Y_t, Z_t)$ of the position vector $P_t$ at the end of the robot can be expressed as follows:

$$
\begin{align*}
X_t &= X_i + \frac{X_f - X_i}{d} v_{i,t} \\
Y_t &= Y_i + \frac{Y_f - Y_i}{d} v_{i,t} \\
Z_t &= Z_i + \frac{Z_f - Z_i}{d} v_{i,t}
\end{align*}
$$

**Figure 8.** Linear trajectory.  
**Figure 9.** Joint coordinate change at linear motion.
In the ROS platform, the motion of the manipulator is simulated by RViz visualization plug-in. Firstly, given two points \( P_1 \) and \( P_2 \), and the trajectory of the end of the manipulator from \( P_1 \) to \( P_2 \) is shown in figure 8. The position coordinates of each joint captured in RViz are drawn into a broken line diagram, and the position change of each joint of the robot is shown in figure 9. It can be seen from the figure that joint 1 and joint 2 close to the pedestal remain basically motionless, so neglecting the relative motion of joint 1 and joint 2 has the least effect on the experimental results. It can be understood that the angle of joint angle 2 is 0, or it can be understood that the two joints close to the pedestal are fixed and only the remaining 6 degrees of freedom are analyzed. We know that the number of degrees of freedom required to get the end of the manipulator to any position is 6, so adding this constraint does not affect the inverse kinematic analysis of the manipulator. Then the straight line interpolation simulation of the simplified manipulator is carried out by RViz. The trajectory constraint of Cartesian space is used here, and the change process of the simplified six joints is obtained by inverse solution operation. The change curves of each joint are drawn by rqt_plot plug-in as shown in figure 10.
nearest joint to the pedestal is defined here as joint [1], and so on to define joint [2] ~ [6]. The transverse coordinates in the figure are running time in units, the unit is s, the longitudinal coordinates are joint angles, adopt arc system. Then the end position of the manipulator generated in ROS is compared with the difference position of the theory, and the error curve is drawn as shown in figure 11.

4.2. Circular arc trajectory planning
For the space arc trajectory problem, the 3D space problem should be converted to the two-dimensional plane problem. In the spatial rectangular coordinate system \(O_1 x_1 y_1 z_1\), a plane or an arc can be determined by any three points \(P_1(X_1, Y_1, Z_1), P_2(X_2, Y_2, Z_2), P_3(X_3, Y_3, Z_3)\) which are not collinear the line. In the Cartesian space, it is known that three non-collinear points \(P_1, P_2,\) and \(P_3\), make vertical bisection lines of \(P_1P_2\) and \(P_2P_3\) respectively, and the two bisector intersect at the point \(O_B\), and the point \(O_B\) is the center of the arc. Then the circular arc radius \(R\), the center angle \(\phi_1\) and \(\phi_2\) can be obtained, and then the circular coordinate system \(O_B x_B y_B z_B\) can be established with the center \(O_B\) as the origin. It can be seen that the \(x_B\) axis is the direction of the outer normal, as shown in figure 12.

![Figure 12. Arc trajectories of Cartesian space.](image)

Where the angle between the \(x_B\) axis and the \(x_A\) axis is \(\alpha\), and the angle between the \(z_B\) axis and \(z_A\) axis is \(\beta\), the circular arc trajectory of the robot arm needs to be referenced by the fixed basic coordinate system, so the circular arc coordinate system can be obtained through translation and rotation. The specific method is to translate the coordinate origin into each other, and then rotate to make the coordinate axis coincide. The transformation matrix from the circular arc coordinate system to the basic coordinate system can be expressed as follows:

\[
\begin{bmatrix}
1 & 0 & 0 & X_{OB} \\
0 & 1 & 0 & Y_{OB} \\
0 & 0 & 1 & Z_{OB} \\
0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\cos\beta & -\sin\beta & 0 & 0 \\
\sin\alpha \cos\beta & -\sin\alpha \sin\beta & 0 & 0 \\
\sin\alpha \sin\beta & \sin\alpha \cos\beta & \cos\alpha & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
= \begin{bmatrix}
\cos\beta & -\sin\beta & 0 & X_{OB} \\
\cos\alpha \sin\beta & -\sin\alpha \cos\beta & -\sin\alpha & Y_{OB} \\
\sin\alpha \sin\beta & \sin\alpha \cos\beta & \cos\alpha & Z_{OB} \\
0 & 0 & 0 & 1
\end{bmatrix}
\]
Where \((X_{OB}, Y_{OB}, Z_{OB})\) represents the coordinates of the origin \(O_B\) of the circular coordinate system in the base coordinate system. At any time \(t\), the coordinate value in the base coordinate system is \(P_{At}(X_{At}, Y_{At}, Z_{At})\), and the coordinate value in the circular arc coordinate system is \(P(X_{At}, Y_{At}, Z_{At})\).

In \(t\) time, the center angle corresponding to the adjacent interpolation point is:

\[
\Delta \phi = \frac{vt}{R}
\]

The total number of interpolation is:

\[
N = \frac{(\phi_1 + \phi_2)}{(\Delta \phi + 1)}
\]

Then the coordinates of the interpolation point \(P_i\) number \(i\) can be expressed as follows:

\[
\begin{align*}
X_i &= R \cos \phi_i = R \cos \left(i(\phi_1 + \phi_2)/N\right) \\
Y_i &= R \sin \phi_i = R \sin \left(i(\phi_1 + \phi_2)/N\right) \\
Z_i &= 0
\end{align*}
\]

The interpolation coordinate \((X_i, Y_i, Z_i)\) here is considered under the arc coordinate system, which can be represented in the base coordinate system by the transformation matrix:

\[
P_{Ai} = \frac{1}{\theta} T \times P_i + \begin{bmatrix} X_0 & Y_0 & Z_0 \end{bmatrix}^T
\]

On the ROS platform, the arc trajectory of the end of the manipulator is shown in figure 13. At this time, the coordinate changes of each joint of the robot are shown in figure 14. In the same way, the joint 1 joint 2 is fixed, and the joint angle is changed as shown in figure 15. The end position of the manipulator generated in ROS is compared with the theoretical difference position, and the error curve is drawn as shown in figure 16.

**Figure 13.** Arc trajectory.
Figure 14. Coordinate change of circular arc motion joint.
5. Conclusion
In this paper, aiming at the problems of complex modeling and difficult simulation of redundant manipulator, the trajectory planning and simulation of Cyton Gamma 300 seven-degree-of-freedom manipulator are carried out through ROS robot operating platform, and the straight line and circular arc trajectory model of manipulator in Cartesian space is obtained, and the simulation results are more reliable, and the motion of each joint of the manipulator is stable and without severe jitter. The end trajectory error is controlled within 2 mm, which verifies the effectiveness of the method and plays an important role in the trajectory planning of multi-degree-of-freedom redundant manipulator to a certain extent. This method also has some limitations, such as poor accuracy and not suitable for large motion of manipulator, which needs to be further improved and studied.

Acknowledgments
This work was financially supported by International Joint Laboratory of Shenyang Jianzhu University and Transilvania University.
References
[1] Li Hao, Zheng Zhizhen, Huang Shunzhou, Yang Fuwei, Liu Zeyu. Trajectory planning and simulation of industrial robot based on ROS [J]. Modular Machine tools and Automation processing Technology, 2018 (12): 59/62.
[2] Liu Haojin, Zhao Yankang, Zhou Zhenxing. Optimal trajectory planning of six-degree-of-freedom series manipulator [J]. Shandong Chemical Industry, 2019, 48 (08): 157-160.
[3] Liu Lei, Ning Yi. Trajectory planning of six-degree-of-freedom manipulator based on ROS [J]. Automation and instrumentation, 2018, 33 (03): 22-25+51.
[4] Liu Yiming, Xu Hui, Geng Changxing, Chen Guodong. Research on manipulator Motion Planning in Ros / Gazebo Environment [J]. Coal Mine Machinery, 2018, 39 (03): 42-44.
[5] Su Ying, Lin du, Xianhao, Liu Jinhao. Cyton Gamma 300 manipulator kinematic analysis [J]. Manufacturing Automation, 2015, 37 (10): 22-25.
[6] Bo Ying. Modeling and simulation of GAMMA300 seven-degree-of-freedom manipulator [D]. Central North University, 2015.
[7] Zhang Songwei. Research on the motion control and visual localization of four-axis mechanical arm based on ROS [D]. Chinese Academy of Sciences (Institute of Optoelectronics, Chinese Academy of Sciences), 18.
[8] Wang Chunrong, Xia Erdong. Motion simulation analysis of six-degree-of-freedom manipulator based on MATLAB [J]. Journal of Yulin University, 2017, 27 (02): 11-14;25.
[9] Wang Deguang, Wang Boheng. Simulation and motion planning of seven-degree-of-freedom manipulator based on ROS [J]. Science, Technology and Innovation, 2018 (02): 116-117.
[10] Zhang Songwei, Feng Chang. Research on motion simulation and control of four-axis manipulator based on ROS [J]. Electronic Design Engineering, 2018, 26 (10): 124-128.
[11] Chen Fei, Wang Zhenhua, Chen Guodong, Xie Shuxin. Research on Motion Planning of Mobile manipulator based on ROS [J]. Coal Mine Machinery, 2018, 39 (06): 36-38.
[12] Yu Qulei. Research on trajectory Planning and tracking Technology of 7-degree-of-Freedom redundant manipulator [D]. Huazhong University of Science and Technology, 2016.
[13] Diego R. Faria, Ricardo Martins, Jorge Lobo, Jorge Dias. Extracting data from human manipulation of objects towards improving autonomous robotic grasping [J]. Robotics and Autonomous Systems, 2011, 60 (3).
[14] Luo Jiman, Wei Zeming, Liu Siyuan, Du Chuang. Research on initial point coordinates of hybrid robot based on binocular vision principle [J]. Journal of Shenyang University of Architecture (from Science Edition), 2018, (1): 150-157. [doi: 10.11717/j.issn:2095-1922.2018.01.17]
[15] Luo Jiman, Liu Siyuan, Guo Songtao, etc. Optimization of silting disk of pipeline desilting robot based on static and dynamic characteristics [J]. Journal of Shenyang University of Architecture (from Science Edition), 2018, (6): 1099-1109. [doi: 10.11717/j.issn:2095-1922.2018.06.16]
[16] Luo Jiman, Zhang Dongyue, Wei Zeming, Liu Siyuan. Analysis of kinematic and mechanical properties of a new type of pipeline robot [J]. Journal of Shenyang University of Architecture (from Science Edition), 2017, (2): 337-346. [doi: 10.11717/j.issn:2095-1922.2017.02.18]