A New Approach to Developing General Manipulator Control System Application Based on ROS

Xuedong Jing, Yuquan Xue and Ya'Nan Chen

June 21, 2019

A New Approach to Developing General Manipulator Control System Application Based on ROS

Xuedong Jing[1]，Yuquan Xue[2] and Ya'nan Chen[3]

Abstract. This paper proposes a new approach to developing control applications for general manipulator using the open source robot operating system (ROS) as the platform. The tool MoveIt! is used to complete the task of the motion planning and the motion data transmission between client and server is realized by setting up the relevant nodes. In combination with the experiment, this paper explains the model building of the robot with a specific manipulator and the motion data is processed by quintic polynomial interpolation algorithm to improve the stability of the manipulator. According to the above methods, it can realize the control of different types of manipulators on the market not specifically referring to certain structures and provides a new way for cross-platform control of robots.

Keywords: Manipulator，ROS，MoveIt，Quintic Polynomial Interpolation.

1 Introduction

Society is becoming more automated with robots starting to perform most tasks in factories and help out in office and home environments. In this environment, one of the most important functions of robots [1] is the ability to manipulate specific objects. However, due to the closure of the traditional manipulator's control system, it's not conducive to conduct demonstration experiments in the university

[1] X. Jing
School of Mechanical Engineering, Shanghai Institute of Technology, Shanghai, 201418, China
e-mail: jingkd2003@aliyun.com
[2] Y. Xue (✉)
School of Mechanical Engineering, Shanghai Institute of Technology, Shanghai, 201418, China
e-mail: 799393978@qq.com
[3] Y. Chen
School of Mechanical Engineering, Shanghai Institute of Technology, Shanghai, 201418, China

laboratories. When faced with different work tasks, the complexity of the closed control system undoubtedly increases the difficulty of programming. At the same time, algorithm migration takes a lot of time and efforts between different types of robots.

Considering the above deficiencies, this paper proposes a new approach to developing general manipulator control applications using the open source robot operating system (ROS) [2] as the platform.

2 Design of Manipulator Control System Application

This paper uses the tool MoveIt to build the motion planning layer of the robot and the most important part of whole system is to transmit the results to the corresponding controller specifically by the way of action. After the relevant trajectory controller interpolates the path points provided by the tool MoveIt, the motion data is sent to the corresponding motor drivers and the data that can be identified by motors is converted into them. Through the above operations, we can realize the control of the general manipulator. ROS manipulator control system framework is shown in Fig. 1

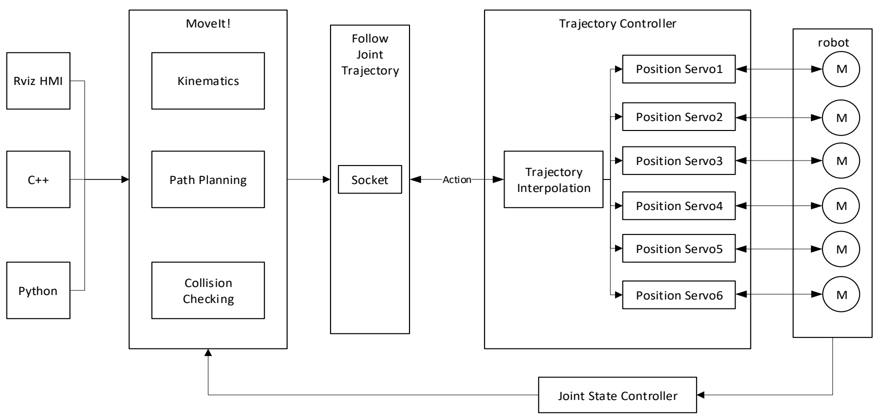


Figure 1 ROS manipulator control system framework

The above content is only for the control part. The construction of the whole manipulator control system can depend on the five-layer architecture of ROS. The five-layer architecture of ROS consists of UI layer, ROS Layer, Interface layer, Communication layer and controller layer. ROS layer is the base of the framework and it provides the basis for communication mechanism. MoveIt! layer can provide motion planning and find the kinematic solutions. Robot controllers communicate with robot client by communication protocol of simple message. ROS manipulator software architecture is shown in Fig. 2

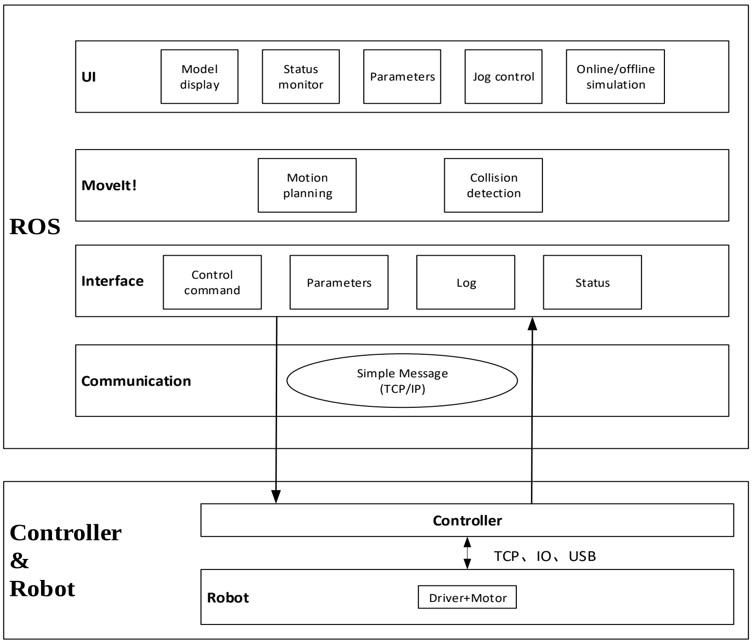


Figure 2 ROS manipulator software architecture

At the software level, we design the framework of the control system based on ROS. And the above four layers can be completed in ROS environment, mainly including motion planning, application processing of manipulator and sending message to the underlying controller. The controller is basically equipped with the function of trajectory interpolation, kinematic solution, system management, interface expansion, algorithm acceleration and motor control, etc. Next, we choose the Jibot-Zxs provided by the ZL-robot for example to build the model of this robot, solve the kinematics, interpolate the motion trajectory provided by the tool MoveIt! and carry out the experiments to realize the stable control of it.

3 Robot Model Building and Motion Planning

3.1 Robot Model Building

The manipulator can be seen as a kinematic chain composed of links through the joint. Each link has 4 parameters, which can be divided into two groups: the rod parameters (a_i, α_i) that determine the structure of the rod and the joint parameters that determine the position of the adjacent rod (d_i, θ_i). In this paper, the D-H parameter method is used to build model and analyze the manipulator. The coordinate system is established in each joint and pose between the coordinate systems is described by homogeneous transformation. Table 1 gives D-H parameter table of the manipulator linkage coordinate system.

Table 1. D-H parameter table.

i	a_i	α_i	d_i	θ_i
1	0	0	339	θ_1
2	π/2	0	0	θ_2
3	0	250	0	θ_3
4	π/2	70	250	θ_4
5	-π/2	0	0	θ_5
6	π/2	0	95	θ_6

The unified robot description model URDF is used in ROS. The URDF is used to describe the kinematic chain of the robotic arm, the inertial characteristics and the parent-child relationship diagram of each link and joint. For complicated mechanical structures such as six-DOF manipulator, the model of the robot should be created using the 3D modeling software Solidworks. The position and orientation of each joint coordinate system are determined by the link coordinate system. The manipulator linkage coordinate system is shown in Fig. 3

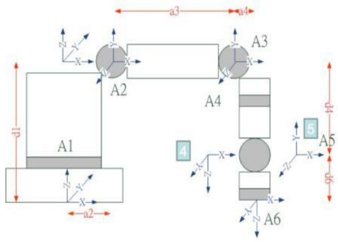
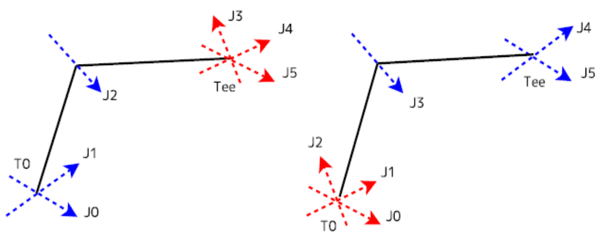


Figure 3 Manipulator linkage coordinate system

3.2 Kinematics

The kinematics [4] of the manipulator is divided into forward kinematics and inverse kinematics. Forward kinematics is known to solve the pose of the end effector at each joint angle; the solution process of inverse kinematics is the opposite of forward kinematics. For forward kinematics, the position of the mechanical arm link relative to the position can be represented by a homogeneous transformation matrix.

For the inverse kinematics, if the end pose is given, it is more complicated to get the angle value of each joint limited by the constraint of the workspace. This article uses the ROS operating tool MoveIt! which provides a way for motion planning, kinematics analysis, collision detection. It uses KDL (Kinematics and Dynamics Library) to solve the inverse kinematics problem by default. This way uses numerical methods which can be applied to different types of robots. However, it requires a lot of iterative operations and the speed of its solution is a little low. At the same time, the numerical solution is sensitive to the initial value and an inappropriate initial value may result in no solution.



1) 2)

Figure 4 Two 6-DOF solvable structure types of the IKFast analytical algorithm

In response to the above problems, the inverse kinematics problem is solved by the IKFast analytical algorithm provided by OpenRAVE [5] motion planning software. The IKFast analytical algorithm can solve the inverse kinematics problem of the two types of 6-DOF manipulators. The first type is the structure that the last three joints J3, J4, J5 axes intersect at one point; The second type is the one whose first three joints J0, J1, and J2 axes intersect at one point, as shown in Fig. 4 The manipulator used in this paper belongs to the first type and its last three joint axes intersect at one common point. The inverse kinematics of the manipulator is solved by the IKFast [6] analytical algorithm to realize the transformation between Cartesian space and joint space.

3.3 Trajectory Interpolation

There are not enough path points to meet the actual task requirements just depending on the tool MoveIt. The robot may shake from time to time if it is lack of motion trajectories, which would seriously affect the quality of the specific experiment. In order to improve the quality and precision of the manipulator during the demonstration experiment, this paper interpolates the planned motion trajectories to improve the stability of the robot. First of all, we analyze the differences between cubic and quintic polynomial interpolation [7] algorithm and provide a new interpolation algorithm to reduce the amplitude of the jitter during the experiment and meet the requirements of different tasks.

3.3.1 Cubic Polynomial Interpolation Trajectory Planning

There are 4 unknown coefficients in the cubic polynomial [8] and the joint angles at the beginning or end of the motion are known in the circumstances. At the same time, the joint speeds are all 0 and the four unknown coefficients can be solved by four known conditions. The joint angle, angular velocity, angular acceleration function equations and constraints are as follows:

$$\theta(t) = h_3 t^3 + h_2 t^2 + h_1 t + h_0 \quad (1)$$

$$\dot{\theta}(t) = 3h_3 t^2 + 2h_2 t + h_1 \quad (2)$$

$$\ddot{\theta}(t) = 6h_3 t + 2h_2 \quad (3)$$

$$\begin{cases} \theta(t_0) = \theta_0, \ \ \dot{\theta}(t_0) = 0 \\ \theta(t_f) = \theta_f, \ \ \dot{\theta}(t_f) = 0 \end{cases} \quad (4)$$

By simultaneous equations (1)(2)(4), we can get the coefficient expressions:

$$\begin{cases} h_3 = \dfrac{-2(\theta_f - \theta_0)}{t_f^3}, h_2 = \dfrac{-3(\theta_f - \theta_0)}{t_f^2} \\ h_1 = 0, h_0 = 0 \end{cases} \quad (5)$$

We can conclude that the cubic polynomial interpolation algorithm can ensure that the joint angle and the angular velocity curve are smooth. There are still some problems with this algorithm. The acceleration mutation would happen if we do not set the acceleration constraint.

3.3.2 Quintic Polynomial Interpolation Trajectory Planning

Based on the cubic polynomial, the quintic polynomial algorithm [9] adds the angular acceleration constraints at the beginning and the end of the motion. It solves the six unknowns by combining the following six equations. The joint angle, angular velocity, angular acceleration function equations and constraints are as follows:

$$\theta(t) = h_5 t^5 + h_4 t^4 + h_3 t^3 + h_2 t^2 + h_1 t + h_0 \quad (6)$$

$$\dot{\theta}(t) = 5h_5 t^4 + 4h_4 t^3 + 3h_3 t^2 + 2h_2 t + h_1 \quad (7)$$

$$\ddot{\theta}(t) = 20h_5 t^3 + 12h_4 t^2 + 6h_3 t + 2h_2 \quad (8)$$

$$\begin{cases} \theta(t_0) = \theta_0, \dot{\theta}(t_0) = 0, \ddot{\theta}(t_0) = 0 \\ \theta(t_f) = \theta_f, \dot{\theta}(t_f) = 0, \ddot{\theta}(t_f) = 0 \end{cases} \tag{9}$$

By simultaneous equations (6)(7)(9), we can get the coefficient expressions:

$$\begin{cases} h_5 = \dfrac{6(\theta_f - \theta_0)}{t_f^5}, h_4 = \dfrac{-15(\theta_f - \theta_0)}{t_f^4}, \\ h_3 = \dfrac{10(\theta_f - \theta_0)}{t_f^3}, h_2 = 0, h_1 = 0, h_0 = \theta_0 \end{cases} \tag{10}$$

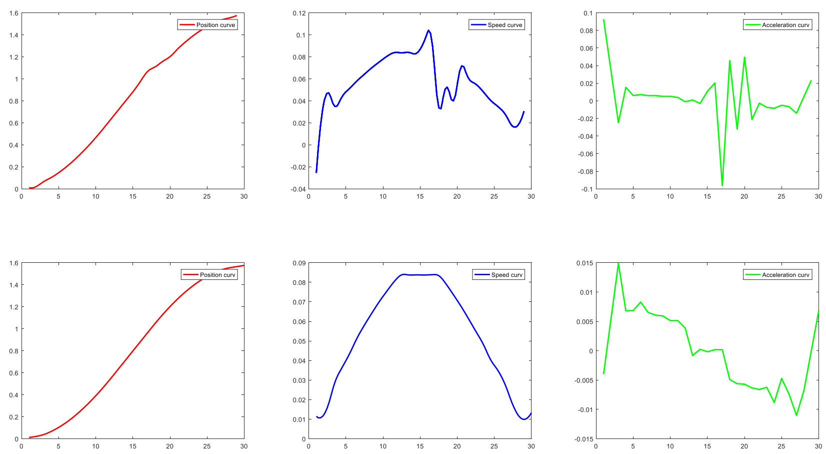
Compared with the cubic polynomial, the quintic polynomial adds the constraint conditions such as the angular acceleration of the beginning time and the end time. From the theoretical and the angular acceleration curve of the second derivative, it can be concluded that there is no jump in the acceleration curve of the quintic polynomial. When the joint angle, the angular velocity and the angular acceleration curves are smooth, the manipulator may run more reliably and steady theoretically.

4 Analysis of Experimental Results

Combined with the demonstration experiment, we verify the above interpolation algorithm by setting the motion path of the manipulator and make sure that the end-effector runs along the edge of a rectangular object. In practice, if the planned trajectory provided by the tool MoveIt! is not interpolated, the manipulator will shake during operation. To solve the problem of jitter, we track and analyze the trajectory data of one joint before and after interpolation in this experiment. The structure of Jibot-Zxs is shown in Fig. 5

Figure 5 the structure of Jibot-Zxs

Comparing and analyzing the motion data before and after the interpolation, it can be drawn that the lack of the path planning point will cause the sudden change of the acceleration during the operation and then the jitter will be generated. After the motion data is processed by polynomial interpolation operations, the acceleration curve is more stable and stable and the manipulator runs more steadily and reliably. The robot can perform more complex tasks by adding intermediate points and path constraints. The experiment results show that for a reasonable demonstration task, the control system can quickly provide the appropriate trajectory and control the manipulator to complete the corresponding operations to meet the actual work requirements. Comparison of motion data before and after is shown in Fig. 6



1)Position curve 2) Speed curve 3) Acceleration curve

Figure 6 Comparison of motion data of single joint before and after interpolation

5 Conclusion

This paper introduces a new approach to developing general manipulator control system based on ROS. This approach can realize the control of different types of manipulators on the market not specifically referring to certain structures and provides a new way for cross-platform control of robots. In combination with the experiment of Jibot-Zxs, we provide a quintic polynomial algorithm to process the motion trajectory provided by the tool MoveIt! Through the above process, the stability of the manipulator can be improved greatly during the operation and we lay the foundation for more complex tasks of manipulator in the future.

Acknowledgments

This work was supported by 'Study on Key technologies of Parallel Robot for Minimally Invasive Spine Surgery', Scientific Research Project of Shanghai Municipal Science and Technology Commission, (Projection No. 16090503700)

References:

1. CRAIG J J. Introduction to Robotics: Mechanics and Control (3rd Edition) [M]. Upper Saddle River: Prentice Hall, Inc. 2004.
2. KANEHIRO Fumio, NAKAOKA Shin'ichiro. ROS integration of the disaster response robot simulator "Choreonoid" [J]. The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec), 2016,2016(0)
3. Andreas GRANAT, Hans HÖFKEN, Marko SCHUBA. Intrusion Detection of the ICS Protocol EtherCAT[P]. 2nd International Conference on Computer, Network Security and Communication Engineering (CNSCE 2017),2017.
4. Frederik Petri Svenningsen, Mark de Zee, Anderson Souza Oliveira. The effect of shoe and floor characteristics on walking kinematics[J]. Human Movement Science,2019,66.
5. Tsuchiya, K., Kagami, S., Yoshizaki, W., Mizoguchi, H..Grasp planning pre-computation by considering center of gravity of objects and its evaluation using OpenRAVE[P]. Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on,2012.
6. Rosen Diankov P R. Automated Construction of Robotic Manipulation Programs[J]. 2010.

7. Xi Yang Ou, Xing Zhe Hou, Ke Zheng, Xiang Du, Jing Ji. Space Trajectory On-Line Interpolation Technique of Six-DOF Robot[J]. Applied Mechanics and Materials,2014,3138(536).
8. Yu Qian, Li Ying. Weighted Combination Interpolation by Piecewise Cubic Polynomial on Triangulations[P]. Artificial Intelligence and Computational Intelligence (AICI), 2010 International Conference on,2010.
9. Rai, J.K., Tewari, R..Quintic polynomial trajectory of biped robot for human-like walking[P]. Communications, Control and Signal Processing (ISCCSP), 2014 6th International Symposium on,2014.