Design of a grasping robot control system using kinematics model

Wenqi Chen, Fenggang Liu*
School of Artificial Intelligence, Wuchang University of Technology, Wuhan 430223, Hubei, China

*Corresponding author e-mail: liufenggang@wut.edu.cn

Abstract. Aiming at the problem of mechanized and repeated parts grasping, and aiming to reduce the development cost, this study added an end-effector and designed a ROS-based grasping robot control system on the basis of fully analyzing the structure and workflow of the robot. The grasping robot can be controlled and monitored in real time by operating on the RVIZ interface. According to the needs and process of grasping work, the control system process design and control system programming of the robot are completed. Subsequent simulation experiments and real object control experiments show that the control system has high robustness and real-time performance. The control system can meet the task of mechanization and repeated parts grasping, and can effectively improve the production efficiency, enhance the competitiveness of enterprises, and reduce the cost of enterprises.

Keywords: grasping robot; path planning; overall control system design

1. The introduction
In the production process, the identified defective parts need to be captured and removed, because the identification and rapid capture of defective parts is difficult to use human operation and the cost is too high [1]. Professional cameras should be used to identify defective products and robots should be used to replace manual grasping [2]. After several years of development, it has been proved that robots have a very high ability to replace human work, although some current robots on the market have this function, such as Yaskawa robot, which has very high safety [3]. Xinsong robot is one of the few cooperative robots with seven degrees of freedom in the market [4]. However, the cost is too high, and the entry is difficult, so it is difficult for the purchased enterprise to achieve secondary development, and finally it is difficult to promote [5]. Therefore, we need to develop a set of simple operation, low cost, future upgrade potential, and for the vast majority of small and medium-sized enterprises can use the control system.

To sum up, in order to achieve the above stated goals, the open source robot operating system ROS was finally chosen in this study through the later research (Robot Operating System). With the grasping of target items on the production line as the application background, the control System platform [6] was built, the control System was designed and the corresponding simulation and physical control experiments were carried out.
2. The establishment of robot kinematics model

The solution of the forward and inverse kinematics of the robot is the first step to analyze the position and attitude of the grasping robot [7]. Because the forward and inverse solutions of different robots are different, the robot in this study is a 6-DOF robot with three adjacent joint axes parallel to each other [8]. The forward kinematics solution is to calculate the grasping position and attitude of the robot based on the known rotation angle and geometric parameters of the motor. The inverse solution is just opposite to the forward solution [9]. In the actual production, we know the final position of the end gripper, and calculate the corresponding three-dimensional coordinates of each joint and the absolute position of each motor through the inverse kinematics solution, so that each joint moves to a unique designated position, thus finishing the positioning of each end gripper. As shown in Table 1, the D-H method was adopted in this study to establish the connecting rod coordinate system [10], and the coordinate system of each connecting rod was established on the six connecting rods respectively. Through multiple operations of the homogeneous transformation matrix between each coordinate system, the position and pose matrix of the end grasp of the robot with respect to the base coordinate system was finally obtained [11].

The parameters are as follows: 1) i: Connecting rod; 2) di: Connecting rod bias; 3) ai: Length of connecting rod; 4) αi: Connecting rod distortion; 5) θi: Joint Angle.

Table 1. Robot D-H parameter table

| i | \( \theta_i \) | \( d_i/\text{mm} \) | \( a_i/\text{mm} \) | \( \alpha_i \) | Range |
|---|---|---|---|---|---|
| 1 | \( \theta_1 \) | 0 | 0 | 0 | \(-\frac{\pi}{2}, \frac{\pi}{2}\) |
| 2 | \( \theta_2 \) | 0 | -90 | 0 | \(-\frac{\pi}{2}, \frac{\pi}{2}\) |
| 3 | \( \theta_3 \) | \( d_2 \) | 0 | 0 | \(-\frac{\pi}{2}, \frac{\pi}{2}\) |
| 4 | \( \theta_4 \) | \( d_3 \) | \( a_2 \) | 0 | \(-\pi, \pi\) |
| 5 | \( \theta_5 \) | \( d_4 \) | \( a_3 \) | 0 | \(-\pi, \pi\) |
| 6 | \( \theta_6 \) | 0 | 0 | -90 | \(-\pi, \pi\) |

2.1. Positive kinematic solution

Grasping robot can be regarded as a combination of end grippers and connecting rods in series. For this reason, if the setting system is established according to the connecting rod of the manipulator, a small coordinate system shall be first established for each connecting rod, and then the relative relation between the coordinate systems of these different connecting rods can be expressed by the homogeneous coordinate transformation matrix [12]. In this study, these matrices are collectively referred to as matrix A, and the order of definition is the homogeneous transformation matrix of the coordinate system from the base coordinates to the end grip, which are respectively \( A_1 \), \( A_2 \),...... Then the position and pose of connecting rod No. 2 relative to the base coordinates can be represented by the product of \( A_1 \) and \( A_2 \), namely:

\[
T_2 = A_1A_2 \tag{1}
\]

Similarly, if \( A_3 \) represents the position and posture of connecting rod 3 relative to connecting rod 2, that is:

\[
T_3 = A_2A_3 = A_1A_2A_3 \tag{2}
\]

The coordinate systems of the connecting rod are successively connected, and the coordinate transformation relationship between them can be described by the transformation matrix \( A_1A_2A_3A_4A_5A_6 \). Thus, the pose of the end of the manipulator can be expressed as:
In the formula, \( n_x, n_y \) and \( n_z \) respectively represent the cosine of the coordinate axis \( X_6 \) of the robot's grasp relative to the initial coordinate system in the direction of \( x, y, \) and \( z \) axes; \( o_x, o_y \) and \( o_z \) respectively represent the cosine of the coordinate axis \( Y_6 \) of the robot's grasp relative to the initial coordinate system in the direction of \( x, y, \) and \( z \) axes; \( a_x, a_y \) and \( a_z \) respectively represent the cosine of the coordinate axis \( Z_6 \) of the robot's grasp relative to the initial coordinate system in the direction of \( x, y, \) and \( z \) axes; \( p_x, p_y \) and \( p_z \) represent the position coordinates of the coordinate axis of the robot's grip relative to the initial coordinate system.

Wherein, the transformation matrix of link \( i \) relative to link \( i-1 \) is:

\[
T_i = A_{i-1} A_i A_{i+1} A_i = \begin{bmatrix}
n_x & o_x & a_x & p_x \\
n_y & o_y & a_y & p_y \\
n_z & o_z & a_z & p_z \\
0 & 0 & 0 & 1
\end{bmatrix}
\] (3)

In Formula (4), \( \theta_n \) is the joint Angle. Since a joint contains two common perpendicular lines, a projection is generated on a plane, and the Angle between the projections is \( \theta_n \). \( \alpha_n \) is the torsion Angle of connecting rod, that is, the included Angle between the NTH +1 joint axis and the plane formed by the NTH joint axis and common perpendicular. \( d_n \) is the offset of connecting rod, that is, the offset distance between two common perpendicular lines of a joint; \( C \) and \( S \) are abbreviated cos and sin; \( N \) is the number of joints. This subject is a six-degree-of-freedom robot, so \( n \) is 6.

By substituting the parameters in Table 1 into Equations (1) and (2), the forward kinematics solution can be obtained.

2.2. Inverse kinematic solution

The postures of each joint can be solved by the postures of the end of the robot gripper in reverse. However, there may be multiple solutions, that is, there may be a variety of positions and postures that meet the requirements in part of the joint, so the calculation is heavy. In order to facilitate the calculation, we introduced the inverse transformation method [13], which deduced the parameters of each joint through the known matrix. By using this method, we needed to pay attention to the order of solving the parameters of each joint, instead of following the sequence from back to front, that is, in the process of solving the inverse solution, we first solved \( \theta_1, \theta_2, \) and then solved \( \theta_3, \theta_4, \theta_5, \) and \( \theta_6 \).

One angle of the three parallel joints \( \theta_2 \). Then ask \( \theta_23 \). Pass \( (\theta_23-\theta_2) \). Obtain \( \theta_3 \). Re-request \( \theta_234 \), with \( (\theta_234-\theta_2) \) Solution \( \theta_4 \). Among \( \theta_23=\theta_2 + \theta_3, \theta_234=\theta_2 + \theta_3 + \theta_4 \). The joint parameters of the inverse kinematics solution can be obtained by this method.

3. Control system design

In the working environment as shown in Fig.1, a camera is installed above the identification area. When the storage box containing the goods is placed to the specified position, the camera will identify the defective products. If the defective products are identified, the location of the defective products will be sent to the industrial computer. IPC controls the movement of the grasping machine to the specified position, grasping the defective products and placing the recycling bin. If no defective goods are identified, a message is sent to tell the upper computer to prepare the identification of the next item.
The control process is divided into three parts: image recognition, robot movement, grip control. The control of grasping robot is essentially the control of motor rotation angle, speed and switching time at each joint, as well as the control of the position and posture of the end grip. Therefore, the specific control steps of this control system are as follows.

1) Call the camera for image recognition, identify the defective products, send the data to the control master station, and the master station sends the information to the computer control system after analysis, ready to grab the target items.

2) After receiving the data, the robot will locate the target object through the feedback information of the camera, plan the motion path, and drive the robot to the target position.

3) When the robot moves to the specified position, adjust the position and posture of the gripper, start the gripper, grab the defective parts and place them in the recycling box.

4) The robot returns to its position and is ready for the next grab.

The overall control process is shown in Fig.2.

![Control process flow chart](image)

**Figure 2.** Control process flow chart
4. Setup of ROS configuration environment

4.1. An overview of the ROS system body

ROS is a robot control development system. It includes a variety of functions such as three-dimensional reconstruction, dynamic simulation, real-time scene simulation, and physical control, which can provide all kinds of URDF files that need to be written for robot development to the maximum [16], including parameters such as robot model and its internal structure, joints, degrees of freedom, color, etc. This is an XML file that ROS recognizes and prepares it for subsequent work. It is a built-in operating system under the ROS system, but has nothing to do with the process management of the system. It provides various function packages, and then establishes the communication between the function packages through the point-to-point mechanism, builds the control platform, and realizes the simulation and physical control of the robot.

1) Construction of URDF files

URDF file [16] was written, including the robot model and its internal structure, joints, degrees of freedom, color and other parameters. This is an XML file that ROS recognizes and prepares it for subsequent work.

2) MoveIt! Configure feature packs

ROS MoveIt! ROS software package is specially developed for mobile operating platform [17]. It integrates integrated software for path planning, stereoscopic perception, kinematics, motion control and navigation, and provides an easy-to-use software platform for developing advanced robot applications. Fig. 3 shows Ros MoveIt! The high layer frame frame. Among them, the Move_Group node is at the core. It acts as a hub connecting individual components to provide completed services. The overall framework is as follows.

![Figure 3. ROS MoveIt! High-level framework](image)

3) Robot simulation

The configured robot is imported into the 3D visualization tool (RVIZ for short) built into the ROS to plan different targets, add and remove objects in the scene and other operations. For this study, the grasping robot model was imported into RVIZ, and the grasping object was added to control the robot to grasp the target object, and the simulation experiment was carried out.

4.2. ROS feature pack configuration

This study calls ROS module, takes its built-in algorithm as the core, combined with the research objective, calculates the necessary parameters of each joint, and finally sends them to each motor. The steps are as follows.
1) Draw a 3D diagram of the robot in SolidWorks, and then call SW2URDF plug-in to convert it into a file that can be recognized by ROS. Finally, perform Setup under ROS and set the initial position of end-effector.

2) Control system programming design. RVIZ interface opens the configured simulation robot, receives the terminal coordinates sent by the upper computer, then calls the path planning algorithm to calculate the rotation Angle, speed, acceleration, position and other parameters required by the movement of each joint to the target position, and sends out the calculation results of each part in turn.

3) The communication between upper and lower machines is established by means of Controller Area Net work (CAN communication), and the rotation Angle of each joint calculated and other necessary parameters are sent to the lower machine through the communication interface.

4) The lower computer chip receives the data, converts it into parameters that can be recognized by the motor and sends them to the motor to control the robot movement.

5. The experiment

Based on UR cooperative robot, a gripper is installed on the end-effector. The upper computer receives the position information of defective products and sends it to the robot, which is sent to the robot core chip through communication. The chip converts the received data into parameters that can be recognized by the motor through analysis, and then controls the speed and rotation time of the motor. Secondly, the gripper moves to the specified position, and then the position and posture of the gripper are carefully adjusted to finally realize the grasping and placing of defective products into the recycling box. The robot experiment process is shown in Figure 4. Fig. 4 (a) shows the robot moves to the initial test position, then receives instructions, moves to Fig. 4 (b), grabs the target, and then proceeds through path planning, as shown in Fig. 4 (c), moves to the target position, as shown in Fig. 4 (d).

![Robot Experiment Images](a) (b) (c) (d)

**Figure 4.** Physical robot control experiment

6. Conclusion

In order to realize the design of the control system for grasping robots, this research realizes the design of the control system for grasping robots based on the Ubuntu system combined with ROS. A control system based on ROS was developed, and the communication mode of upper and lower computers in ROS was improved, so as to achieve a communication mode with wide applicability and
simple operation under the premise of guaranteeing the original real-time performance and robustness. Simulation experiments and physical control experiments were built. It can be seen from the experiment that the control system can realize the capture of defective products, which has practical application background and economic value. In the future, the storage box can be placed on the assembly line to form a fully automated production line, greatly improving production efficiency.

References
[1] YING Weijun, FAN Xingduo, LI Yingping. Visual detection and positioning of chips' residual products for screening of the residual products on the chips production line: identification: positioning: opencv[J]. Equipment Manufacturing Technology, 2018(1): 94-97.
[2] SUN Jiyuan. On the technology of camera color recognition[J]. Science and Technology Economic Guide, 2016(36): 87-88.
[3] ZHANG Ying, ZENG Konggeng. Secondary development function of intelligent application of Sichuan robot[J]. Metal Processing (Hot Working), 2016(12): 30-33.
[4] XU Fang, ZOU pengshan, ZHENG chunhui. Development and application of Xinsong robot industry[J]. Robot Technology and Application, 2011(5): 14-18.
[5] CHEN Yixuan. The road of robot secondary development[J]. Shanghai state owned capital, 2014(7): 64-66.
[6] Araujo A, Portugal D, Couceiro M S, et al. Integrating Arduino-based educational mobile robots in ROS[C]// International Conference on Autonomous Robot Systems. 2013.
[7] Manufacturing Automation, 2014 (15) : 109-112. Huang Chenhua. Geometric Solution of Inverse Kinematics of Industrial Robot[J]. Manufacturing Automation, 2014 (15) : 109-112.
[8] FU Guodong, ZHENG Xiangzhou, HUANG Si. Kinematic analysis of a 5-DOF manipulator[J]. Mechanical transmission, 2015(9): 151-154.
[9] XIE Zhijiang, FENG Chao, WANG Chengfei. Forward kinematics solution of 6-PSS parallel robot based on BP neural network[J]. Journal of agricultural machinery, 2014, 45(10): 313-318.
[10] JIAO Guotai, YU Yueqing, LIANG Hao. Structural matrix analysis method of robot pose error[J]. Journal of applied foundation and Engineering Sciences, 2001, 9(2): 259-265.
[11] Zhang H, Wang Y X, Wang S Z. Mechanism Design and Kinematics Positive Solution of the Exoskeleton Upper Limb Rehabilitation Robot with 6-DOF[J]. Advanced Materials Research, 2013:74474-77.
[12] D'Souza A, Vijayakumar S, Schaal S. Learning inverse kinematics[C]// Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Ex-panding the Societal Role of Robotics in the the Next Mil-lenium(Cat. No.01CH37180). IEEE, 2002.
[13] LIU Qifan, XIE Ming, JIANG Lei. Motion planning of industrial robot based on ROS[J]. Modular machine tool and automatic processing technology, 2017(5): 36-39.
[14] HE Daxun, CHEN Gang. In depth analysis of the Ubuntu operating system[M]. Beijing: Tsinghua University, 2008.
[15] MENG Shaonan, LIANG Yanbing, SHI Heng. Motion planning of 6-DOF manipulator based on ROS platform[J]. Journal of Shanghai Jiaotong University, 2016(S1): 94-97.
[16] LI Yuncheng. Research on Motion Planning of Manipulator in High-dimensional Conformation Space Based on ROS Platform [D]. Beijing: Beijing University of Technology, 2017.