ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

Research on Grasping Robot based on MATLAB

Wen Zhang, Yongxiang Jiang

Tianjin University of Technology and Education, School of Mechanical Engineering, Tianjin 300222, China

Abstract

A vision robot, based on MATLAB simulation, optimizes the motion process of the robot arm through MATLAB simulation. It performs camera calibration to identify objects and estimates the depth of target object pixels. By utilizing coordinate transformations between manually and visually calibrated pixel coordinate systems and the spatial robot arm coordinate system, the corresponding data is transmitted to the robot through the ROS operating system, enabling autonomous grasping of the robot arm. The robot vision system acquires pose data of target information in the working environment for data processing, and then transmits the processed target pose data to the control box of the robot arm, which controls the robot arm to complete the grasping task.

Keywords

MATLAB Simulation; Robot Arm Working Space; Robot Hand Grasping; Hand-Eye Calibration.

1. Introduction

Currently, the majority of robotic arm autonomous grasping technology relies on visual control. However, as the unstructured work environment becomes increasingly complex and the uncertainty of the objects to be grasped increases, the difficulty of autonomous grasping for robotic arms continues to rise. In complex environments, robotic arms must constantly adjust their posture for grasping. During this process, frequent spikes and drops in joint acceleration and torque lead to significant energy consumption, significantly reducing the lifespan of the robotic arm. The key to addressing this issue lies in the rationality of the motion path planning for the robotic arm and the smoothness of the process. This paper employs MATLAB to simulate the workspace and motion process of a robotic arm. Based on the simulation results, the motion process of the robotic arm is continuously modified and optimized to achieve optimal performance. The hand-eye calibration method for the binocular vision robot in this paper adopts the eye-in-hand approach. The camera detects the position information of the image on the target object, utilizes the calibration coordinate transformation matrix to facilitate transformations between coordinate systems, and calculates the motion of the robotic arm through its coordinates. The controller then guides the robotic arm to grasp the target object after reaching a specific position. By integrating various advanced technologies, such as software simulation, visual calibration, motion planning, and object grasping, an experiment on autonomous grasping of target objects is ultimately achieved.

2. Robot Grasping Mechanism

For grasping robots, the grasping mechanism is a crucial component. Given the need for the robotic arm to grasp objects from multiple angles and the relatively small mass of the grasped objects, the Ryleman RM65-B six-axis robot with multiple degrees of freedom was chosen, as illustrated in Figure 1.

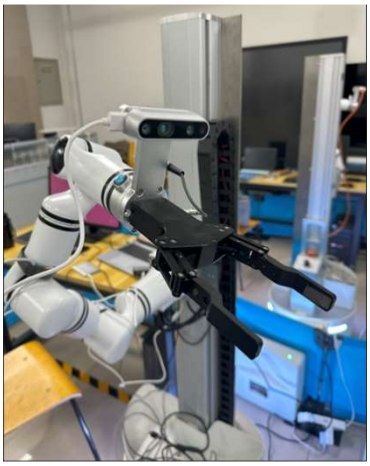


Figure 1. Ryleman RM65-B six-axis robot

3. Introduction to the Overall Experimental Platform

The binocular robot grasping system based on MATLAB studied in this paper is shown in Figure 2.

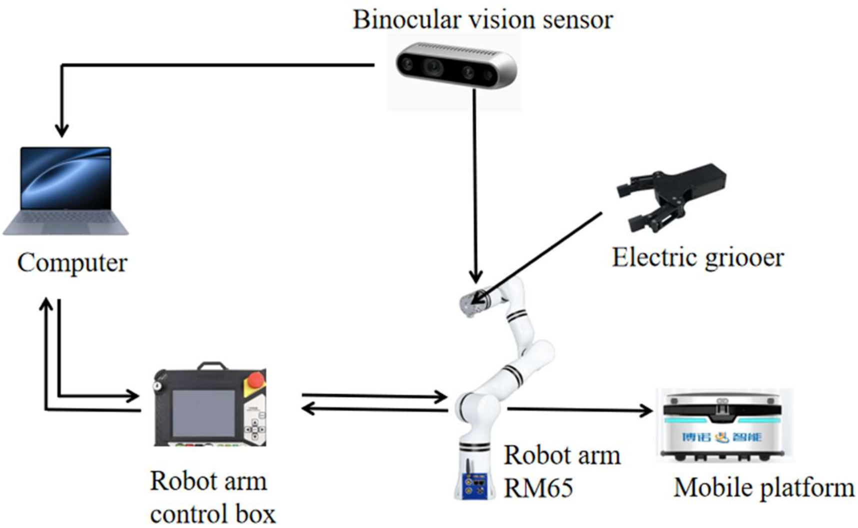


Figure 2. Binocular robot grasping system based on MATLAB

The entire system comprises six components: the robotic arm control box, RM65 robotic arm, mobile platform and gripper, computer, and vision sensor. Once the mobile platform arrives at the designated work area, the binocular vision sensor captures image information from the work environment and relays it to the computer. Following image processing, the pose data of the target information within the work environment undergoes further processing. Subsequently, the refined target pose data is transmitted to the robotic arm control box, which subsequently instructs the robotic arm to execute the grasping task.

4. Motion Analysis of the Robot Arm

The autonomous grasping process of a robotic arm is influenced by various factors, such as path planning during the arm's movement, stability of the grasp, and energy consumption during movement, all of which determine the level of grasping performance.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

4.1 Coordinate Transformation

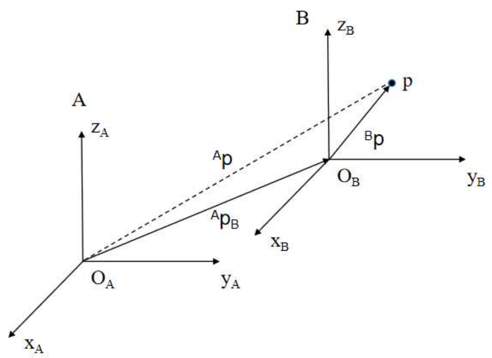
The translation motion of a robotic arm refers to the relative translational relationship between the base coordinate system A and coordinate system B with respect to the coordinate origin, where coordinate systems A and B remain consistent in all directions along the coordinate axes. In the following formula (1), ${}^{B}p$ represents the position matrix of p in the B coordinate system, and ${}^{A}p_B$ denotes the translation transformation matrix.

$$ {}^{A}p = {}^{B}p + {}^{A}p_{B} . \tag{1} $$

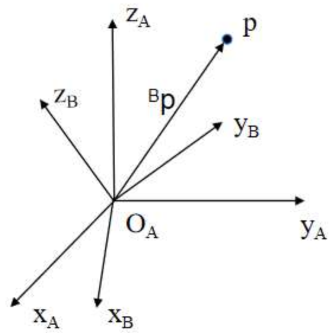
The rotational motion of the robotic arm occurs within a two-dimensional rotational plane, where the coordinate origins of coordinate system A and coordinate system B coincide. The angle between the coordinate axes corresponding to each coordinate system represents the rotation angle. In formula (2), ${}^{A}_{B}R$ is represented as the rotation matrix of coordinate system B relative to coordinate system A.

$$ {}^{A}p = {}^{A}_{B}R\, {}^{B}p . \tag{2} $$

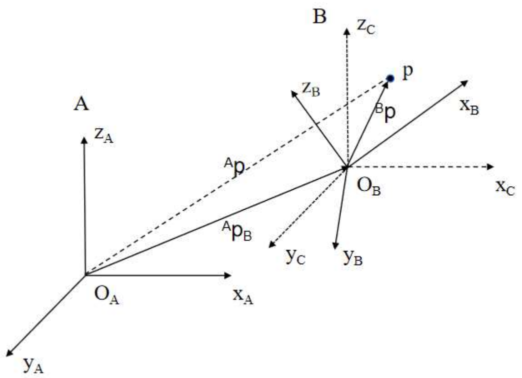
During a motion, the translation and rotation of the robotic arm may coexist, necessitating a composite coordinate transformation that encompasses both translation and rotation, as illustrated in Figure 3.



(a) Translation transformation



(b) Rotation transformation



(c) Compound transformation

Figure 3. Schematic diagram of displacement

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

If translation and rotation transformations are combined into homogeneous coordinates, as demonstrated in formulas (3) and (4), the pose matrix ${}^{A}_{B}T$ is defined as follows:

$$ {}^{A}p = {}^{A}_{B}T\,{}^{B}p \tag{3} $$

$$ {}^{A}_{B}T = \begin{bmatrix} {}^{A}_{B}R & {}^{A}p \\ 0 & 1 \end{bmatrix} \tag{4} $$

4.2 Analysis by D-H Method

The mathematical model of the six-degree-of-freedom grasping robot in this experiment was constructed using the D-H parameter method. Initially, a base coordinate system was established at the geometric center of the base. Subsequently, corresponding three-dimensional spatial coordinates were established at the geometric centers of each joint. Based on the principle of homogeneous transformation, the relationship between the coordinate systems of adjacent joints can be deduced. The end pose of the robotic arm is derived from the base coordinate system and the homogeneous transformation matrices of each joint.

The D-H i-1 parameter method is shown in the figure. Taking three joints as an example, the first joint i-1 is set as, and the connecting rod i connects the joint with the joint i-1. Similarly, a robot parameter model of six axes and five connecting rods can be constructed. The steps are as follows:

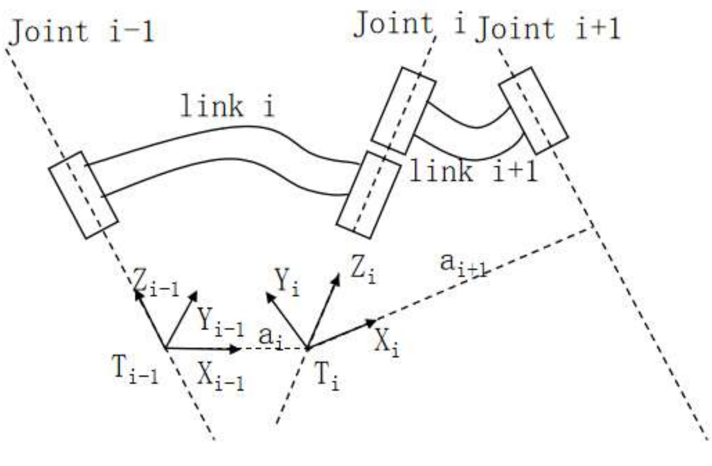


Figure 4. Schematic diagram of D-H method

(1) Set the Z axis as the vertical line of the joint, and calibrate the Z axis as.The right hand rule selects the direction and the sliding direction.

(2) Taking the shortest distance as the principle, select the common vertical line of the vertical line in the adjacent joint. In Figure 4, the distance between Z_{i-1} and Z is represented by the length of the common vertical line a_i, and the direction of the X-axis can be referred to by the direction a_i.

(3) If the adjacent Z-axes are parallel, then the architecture of handling the entire model can be simplified, and the common vertical line of the previous joint can be selected for the simplified collinear relationship.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

(4) If the Z axes of adjacent joints are intersecting, there is no orthogonal common perpendicular between them, because.This selects the cross product direction of the Z axis, defined as the direction of the X axis.

In this paper, the six-free Ryleman RM65-B robot is taken as the research object, and the robot arm model created by D-H method is shown in Figure 5.

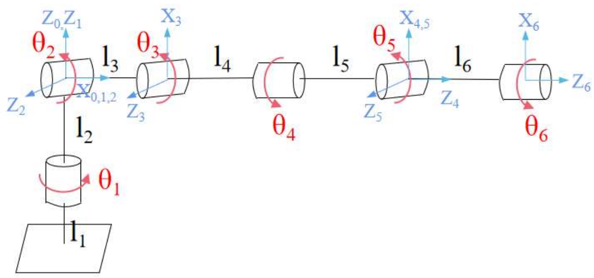


Figure 5. Robot arm model

The six-axis Ryleman robot requires different angles for grasping various target objects. To simplify model construction, this experiment only utilizes the translation component from the end-effector pose matrix of the robotic arm to construct the mathematical model using the D-H method. The D-H parameters for the robotic arm are listed in Table 1, and the six-axis robot model is depicted in Figure 6.

Table 1. D-H parameters of the robot arm

Connecting Rod i	Joint Angle θ_i	Rotation Angle a_{i-1}	Rod length l_i	Offset d_i
1	θ_1	0	0	0
2	θ_2	$\pi/2$	0	0
3	θ_3	0	l_3	0
4	θ_4	$\pi/2$	0	l_4
5	θ_5	$-\pi/2$	0	0
6	θ_6	$\pi/2$	0	0

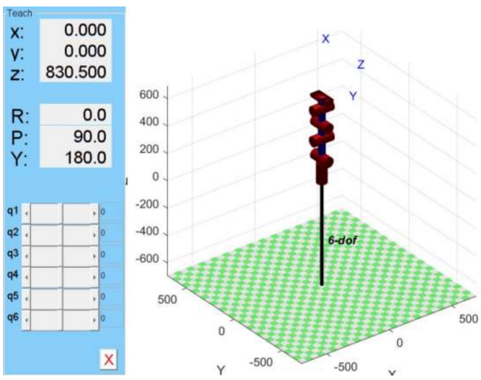


Figure 6. Six-axis robot model

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

The Denavit-Hartenbery method is used to solve the kinematics of the established coordinate system. The transformation matrix can be obtained by using the symbol variable syms in MATLAB, and the ${}^0_1T, {}^1_2T, {}^2_3T, {}^3_4T, {}^4_5T, {}^5_6T$ corresponding parameters can be obtained after solving all the elements in the matrix.

5. Work Space Analysis

In the above process of creating the robot arm model, we have obtained the forward kinematics equation of the six-axis Ryleman robot arm by using D-H method, and the pose of the end of the robot arm can be obtained by substituting the joint variables into the equation. In the case that the rotatable range of each joint is known, we can also get all the positions that the end of the robot arm can reach. We call this possible set of reachable workspace, which is also an important parameter index for the motion analysis of the robot arm.

In this paper, Robotic Toolbox in MATLAB was used for kinematics simulation of the robotic arm. The control system should satisfy the rapidity when analyzing the working space of the robotic arm, so Monte Carlo method which was easy to implement and fast to analyze was used to obtain the reachable working space of the robotic arm. The flow of algorithm analysis is shown in Figure 7.

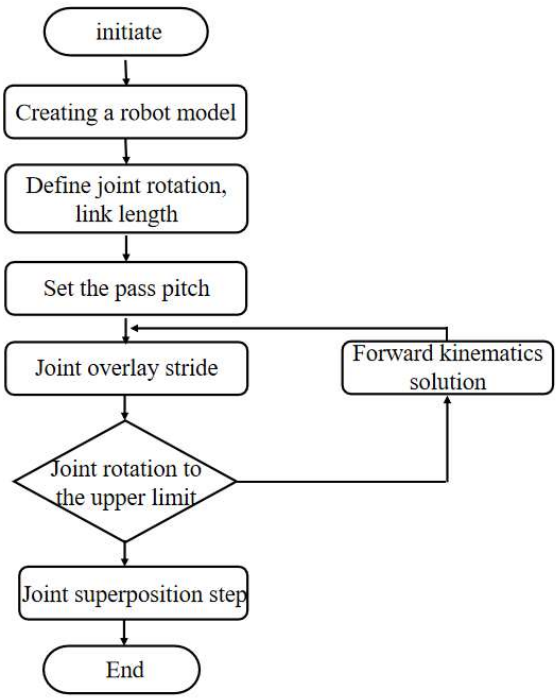
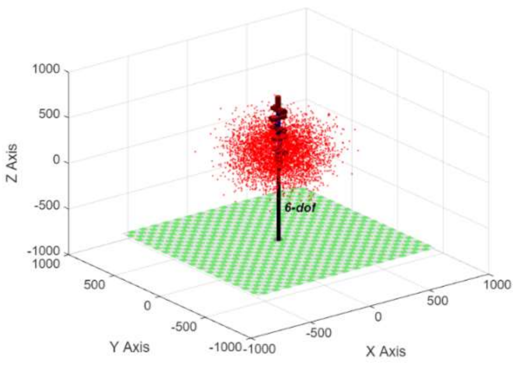


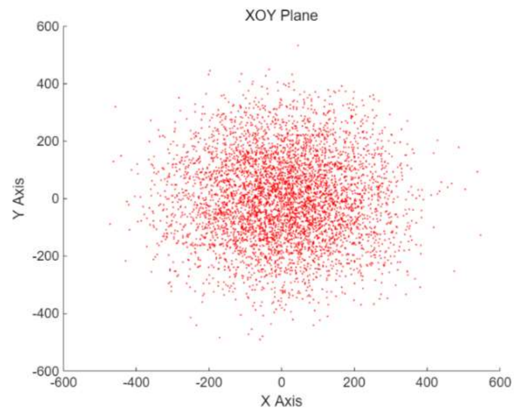
Figure 7. Flowchart of workspace analysis

This paper employs the Monte Carlo method to compute the workspace of a six-degree-of-freedom robot. It sets the number of discrete points to collect and utilizes the lengths and rotation angles between the robotic arm's joints to constrain the generation of various probability distribution random variables. Ultimately, the workspace of the robotic arm is determined using forward kinematics solving functions to obtain the final motion data. By importing the previously established robotic arm model and configuring the forward kinematics simulation function, we set the joint restriction angles and set the number of calculation points (n) to 10000, resulting in a point cloud map depicting the robotic arm's workspace in the XYZ coordinate space. The workspace simulation results for the six-axis Ryleman robot are exhibited in Figure 8.

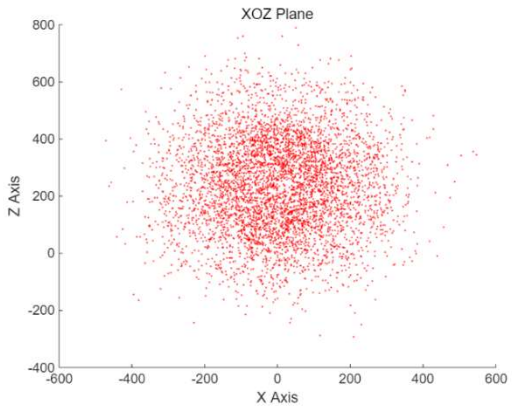
ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006



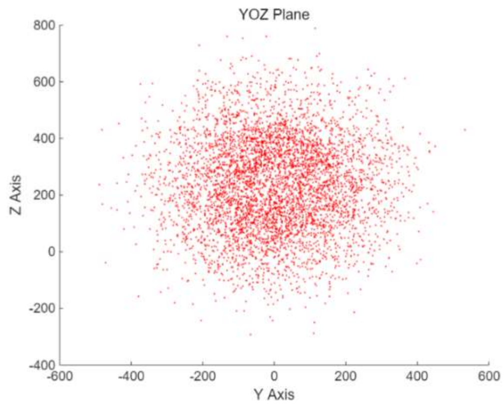
(a) The working space that the robot arm can reach



(b) Projection of the XOY plane of the working space



(c) Projection of the YOZ plane of the workspace



(d) Projection of the XOZ plane of the workspace

Figure 8. Simulation results of workspace

To enhance the precision and stability of robotic arm operations, kinematic analysis and trajectory optimization constitute a core aspect of robotics technology. In the preceding section, we utilized the D-H method to perform kinematic and workspace analyses on the robotic arm, demonstrating the feasibility of establishing kinematic equations. The introduction of trajectory optimization stems from the fact that robotic arms predominantly adopt simple path movements between two points during autonomous grasping tasks. However, this movement pattern frequently involves abrupt directional changes, leading to frequent acceleration and deceleration of the robotic arm, which in turn requires more energy during its motion process. The rational optimization of the robotic arm's motion path primarily focuses on exploring changes in joint acceleration, angular acceleration, joint torque, and other related factors during its motion. This not only enhances the smoothness of the robotic arm's motion but also reduces overall energy consumption and lowers its operational costs. To validate the reasonableness and effectiveness of the kinematic parameters, this experiment employs the fifth-degree polynomial interpolation method to analyze the robotic arm's motion process. When the robotic arm transitions from its initial pentagram position to the pose depicted in Figure 9, its end-effector motion trajectory is illustrated in Figure 9.

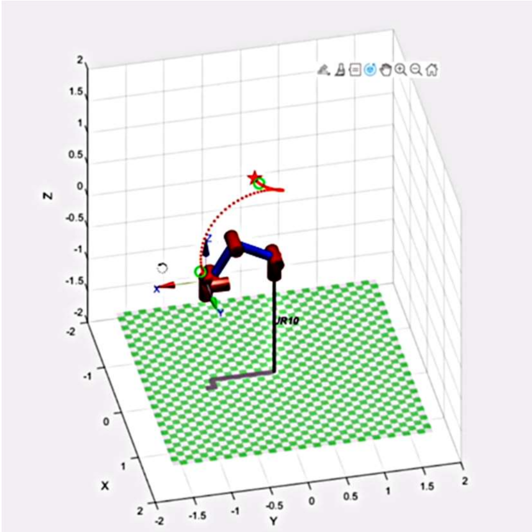
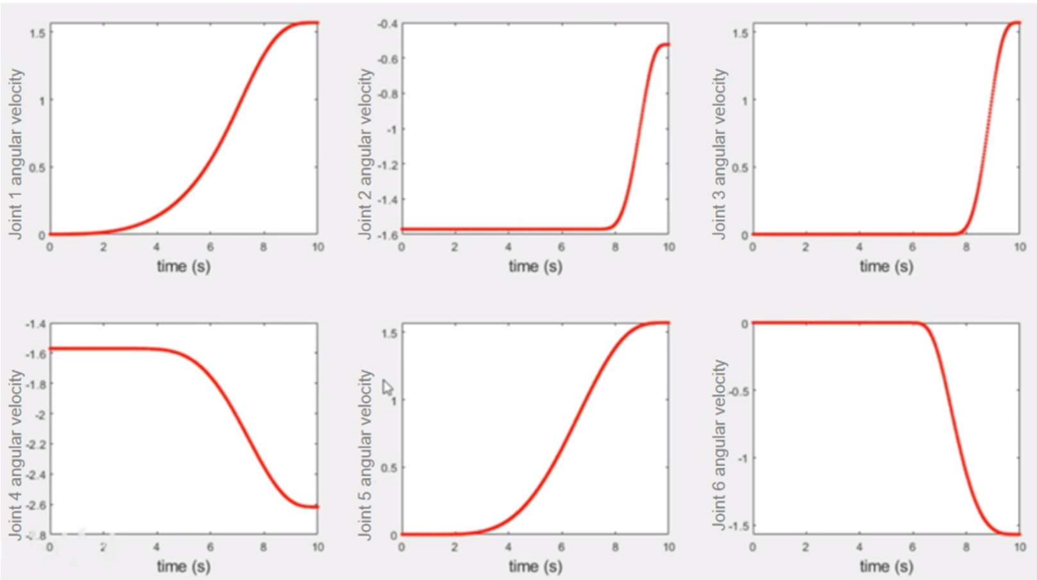


Figure 9. Position change diagram of the end of the mechanical arm

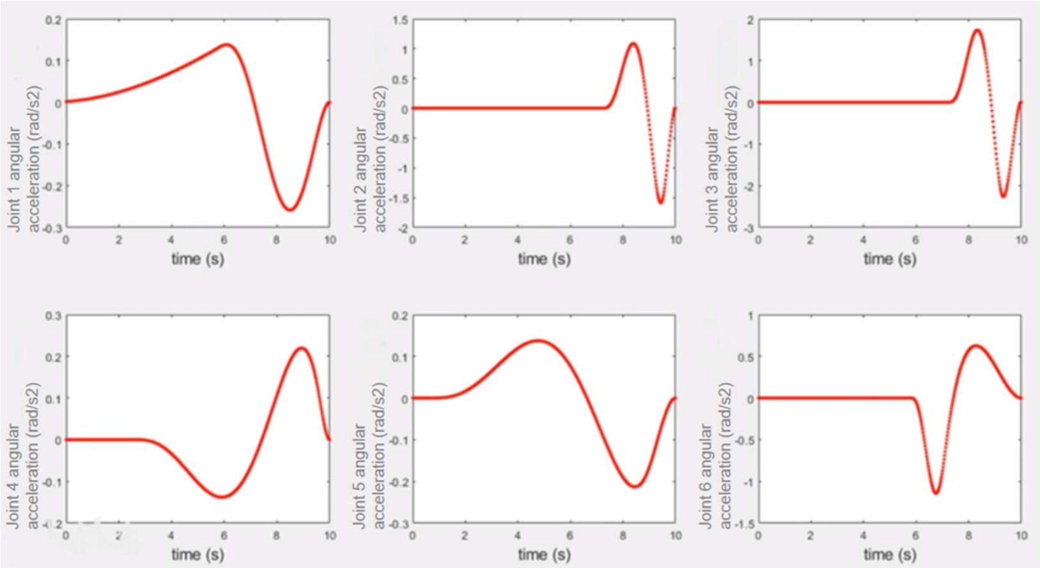
After the trajectory planning of the mechanical arm is completed, the changes of angular velocity, angular acceleration and joint torque of the six joints of the mechanical arm are shown in Figure 10.

The MATLAB simulation experiment generates curve graphs depicting the angular velocity, angular acceleration, and torque variations of six joints during the motion of the robotic arm. This is primarily aimed at analyzing the stability of the robotic arm's motion process. From the trajectory curve derived from fifth-degree polynomial programming and simulation, it is evident that the angular velocity and angular acceleration curves of the six joints of the robotic arm exhibit relative smoothness during operation. There is no abrupt increase evident at any point, indicating that the simulation algorithm effectively ensures stable operation of the robotic arm. In Figure 10 (b), the angular acceleration curves of the six joints demonstrate a gradual increase and decrease during startup and shutdown. This variation in angular acceleration prevents vibrations caused by sudden acceleration changes during startup and shutdown, ensuring smooth operation of the robotic arm.

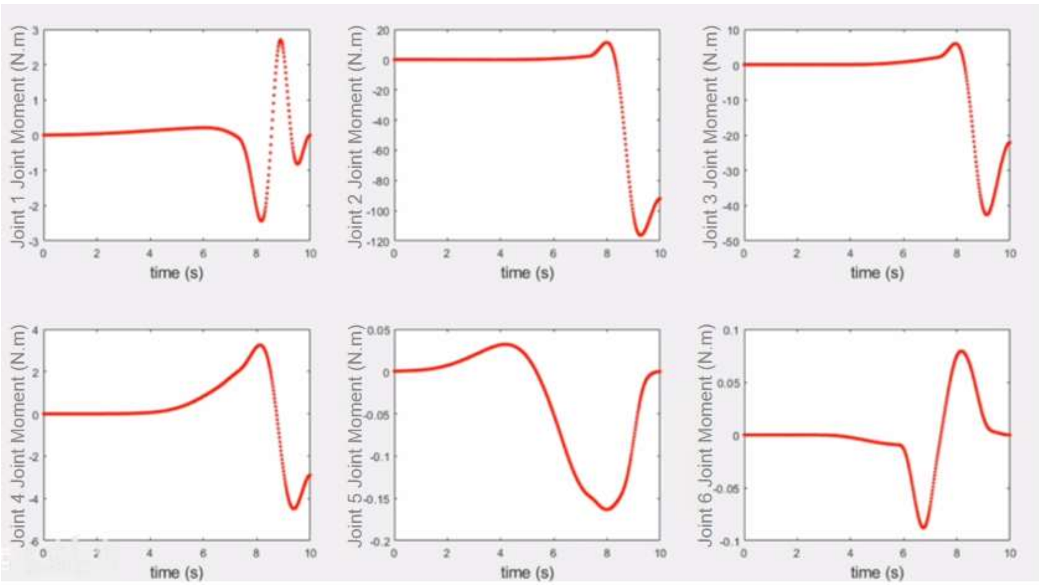


(a) Diagram of angular velocity variation of the joint of the mechanical arm

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006



(b) Variation diagram of angular acceleration of the joint of the robot arm



(c) Variation diagram of joint torque degree of mechanical arm

Figure 10. Variation diagram of angular velocity, angular acceleration and joint torque

6. Grasp Experiment of Robot Arm

The camera was fixed to the end of the robot arm and connected to the host through the network cable. The relevant node program in the Moveit frame of the robot arm control module was started in the ROS environment. After starting the grasp program, the camera recognized the position information of the target object and carried out the independent grasp task.

Easy_hand eye is used in the hand-eye calibration process, which is one of the feature packs for hand-eye calibration. In this experiment, we will use easy_hand eye to complete the task of eye-hand visual calibration. At the same time, in the visualization environment of Rviz, TF transformation in ROS is used to obtain the pose information of ArUco identification code in the coordinates of depth sensor Realsense D435, as shown in Figure 11. The red square represents the pose information of the ArUco identifier. Its name is aruco_marker_frame. The name of the camera coordinate system is camera_frame.

Volume 11 Issue 2, 2025
DOI: 10.6919/ICJE.202502_11(2).0006

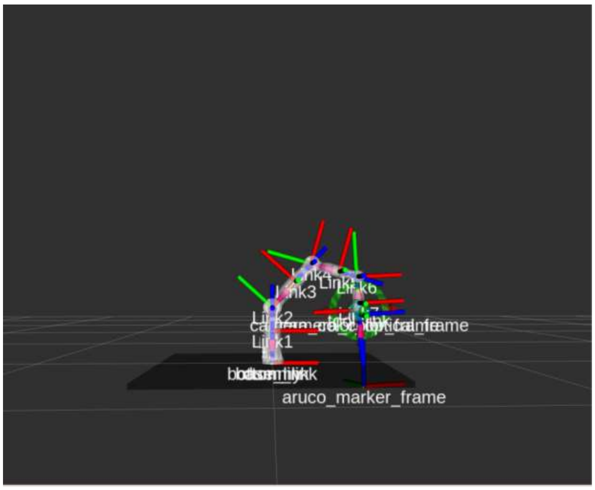
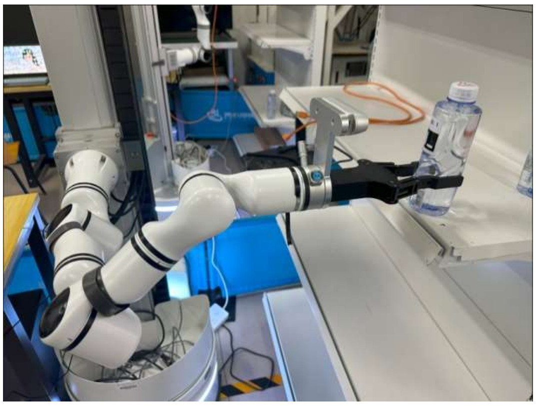


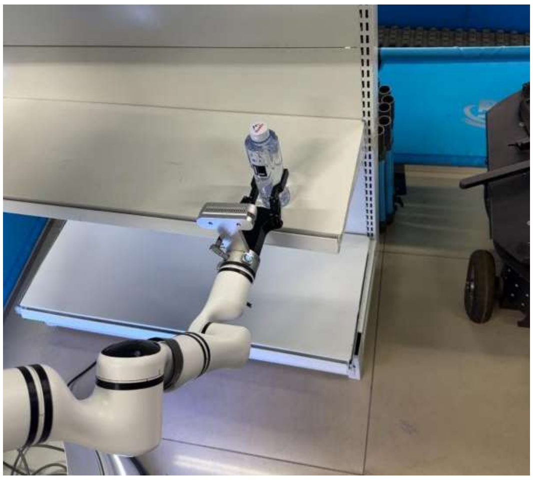
Figure 11. Pose in the Rviz environment

(a) Visual recognition



(b) Grasping

(c) Carrying



(d) Place

Figure 12. Process of grabbing target object

The RealSense D435 camera, mounted at the end of the robotic arm in our experimental platform, is primarily utilized in visual inspection systems. The camera's Type-C data cable is connected to the host's USB 3.0 interface, transmitting collected data on the working environment and target object depth to the host. The host controls the Ryleman RM65-B robotic arm to engage in the grasping of target objects. The robot grasping system primarily comprises three components: target object

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

recognition, positioning, and grasping. Intelligent target recognition is achieved by recognizing target object information through the ArUco node in ROS and displaying image information via the image_view node. Based on this, the target object's position information is clarified through data conversion between the relevant coordinates of the robot, as elaborated earlier, and then grasped by the robotic arm. The grasping process of the robotic arm is illustrated in Figure 11.

The experimental testing involved the robotic arm's visual recognition, grasping, and placement of water bottles. The robotic arm completed 100 grasping tests within a designated area. Each successful cycle, from the camera's initial recognition of the water bottle, through its grasping, and onto its placement on the next shelf, was counted as one grasping process. Throughout the entire grasping process, the operating parameters of the robotic arm were not manually adjusted. Once the target object's position was determined, the grasping operation commenced. On average, it took the robotic arm 8 seconds to reach the water bottle from its starting position, 4 seconds to grasp it, and another 4 seconds to return to its initial pose. The grasping results are detailed in Table 2.

Table 2. Crawling results

Grabbing Time /s	Placement Time /s	Next time Time /s	Total attempts	Hits	Success rate /%
8	4	4	100	95	95

In this experiment, a specific object a water bottle was grasped 100 times, achieving a success rate of 95%. The reasons for the grasping failures included: the mechanical arm grasped the target object with a smooth surface, lacking significant friction and adaptability; the water bottle was either placed too far or outside the operable range of the robotic arm, preventing the arm from planning its motion trajectory. Factors such as hardware and environmental conditions influenced the calibration process, potentially leading to errors in the calibration results. After calibration, the pose of the object in camera coordinates can be obtained and then converted to the pose under the robotic arm for debugging. Any errors can be compensated and calculated in the program later.

7. Epilogue

Based on the analysis of experimental results, the visual grasping robot designed in this paper, which relies on MATLAB analysis, has essentially achieved the anticipated outcomes. It has accomplished closed-loop operations, including visual calibration, target object recognition and positioning, as well as autonomous grasping. This verifies the accuracy and stability of the coordinate conversion algorithm between the camera and the robotic arm, the camera recognition and positioning algorithm, and the path planning algorithm optimized through MATLAB analysis. Ultimately, it has successfully achieved autonomous grasping of the target object. Looking ahead, we will persist in unlocking the immense potential of visual recognition matching and flexible grasping of robotic arms. We aim to enhance the accuracy of autonomous grasping in more intricate environments and leverage the potential advantages of multi-directional fusion analysis.

References

[1] H.M.Liu, G.F. Zhang, H.J.Bao: A review of simultaneous localization and map construction methods based on Monocular vision,Journal of Computer Aided Design and Graphics, Vol. 28(2016)No.6,p.855-868.

[2] H.L. Wei, Y.T. Shang, S. Sun, et al. Design of Sorting Experimental Platform for Industrial Robot based on Machine Vision. Navigation and Control, Vol. 18(2019)No.1, p.105-111.

[3] C.S. Hu, X.P. Yan, H.X. Wei, et al. A Review of object Detection and trajectory Prediction based on stereovision [J]. Computer Engineering and Applications, Vol. 58(2022)No.3, p.50-65.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202502_11(2).0006

[4] Z.Y. Chen: Visual Guidance and Motion Control of Mobile Palletizing Robot. Jiangxi University of Science and Technology, Vol. 2015.

[5] Fauger O D, Toscani G. Camera calibration for 3D computer vision[J].Proceedings of the International Workshop on Machine Vsion and Machine Intelligence,1987:240-247.

[6] BOURBONNAISF,BIGRASP,BONEVIA. Mini-mum-Time Trajectory Planning and Control of a Pick-And-Place Five-Bar Parallel IEEE/ASME Trans Mech, Vol. 20(2015). p740-749.

[7] J.H. Han. Industrial Robot . 4th Ed. Wuhan. Huazhong University of Science and Technology Press, Vol. 2019. p1-5.

[8] C.S. Xiong, X.H. Huang, M. Wang. Hand-eye Stereo Vision Algorithm and Implementation Robot (Robot , China). Vol. 23(2001)No.2. p113-117.

[9] W.M. Zhang, B. Liu, H.B. Li. Research on feature point extraction and matching algorithm in 3D reconstruction based on binocular vision. Optical Technique, Vol.34(2008)No.2. p181-185.

[10] Y. Yun, H. Song ,B.J. Xu. Research on Kinematics Simulation of 6-DOF Industrial Manipulator. Mechanical Science and Technology for Aerospace Engineering, Vol. 2018(8). p1167-1176.

[11] G.S.Young,T.H.Hong,M.Herman,et al.Kinematic calibration of an active camera system,in Proc.IEEE Conf.Computer Vsion Pattern Recognition,1992. p748-751.

[12] W.T. Wang. Research on trajectory Planning and Dynamics of Robotic arm(Shandong University, China 2017). p31-34.