RESEARCH ARTICLE

Design and Kinematic Simulation Analysis of a Vision-Guided 5-DOF Robotic Arm System

Hui Zhou[1,*]

[1] School of Electromechanical Engineering, Nanjing Xiaozhuang University, Nanjing 211171, China

Abstract

Traditional teach-and-repeat robotic arms, which rely on pre-programmed trajectories and manual teaching, struggle to adapt to operational requirements in unstructured environments where the position, posture, or type of target objects may vary unpredictably. In contrast, robot control technology based on visual servoing empowers robotic arms with "visual perception" capabilities, enabling real-time environment sensing and dynamic action adjustment, which significantly enhances the flexibility and efficiency of sorting and assembly operations. This paper designs and implements a five-degree-of-freedom (5-DOF) vision-guided picking robotic arm system based on a distributed control architecture, utilizing MATLAB as the upper computer for high-level decision-making and image processing, and an STM32 microcontroller as the lower computer for real-time motor control and communication. The research specifically addresses the key technical challenges of achieving motion smoothness and positioning accuracy under low-cost hardware conditions, where traditional control methods often fall short due to mechanical limitations and sensor inaccuracies. To this end, a quintic polynomial interpolation algorithm is employed for joint-space trajectory planning, ensuring continuous velocity and acceleration profiles and effectively mitigating the mechanical shock and jitter inherent in low-cost stepper motors and servos. Furthermore, a linear regression error compensation model is proposed to correct the systematic positioning errors caused by lens distortion and mechanical flex in monocular vision-based target localization, achieving high-precision 3D coordinate calculation without the need for expensive depth sensors. Experimental results demonstrate that the proposed system achieves smooth and reliable motion control, reduces average positioning error to within ±3 mm, and attains a 90% success rate in autonomous grasping tasks, validating its practical value for automated sorting applications in cost-sensitive scenarios.

Keywords: machine vision, 5-DOF robotic arm, STM32 microcontroller, trajectory planning, kinematic modeling.

1 Introduction

1.1 Research Background and Significance

Since the beginning of the 21st century, global manufacturing has undergone a profound transformation towards intelligent manufacturing. Strategic initiatives such as Germany's "Industry 4.0," the United States' "Industrial Internet," and

Submitted: 25 January 2026
Accepted: 09 February 2026
Published: 19 February 2026

Vol. 1, **No.** 1, 2026.
10.62762/TICPS.2026.895157

***Corresponding author:**
✉ Hui Zhou
15261498478@163.com

Citation
Zhou, H. (2026). Design and Kinematic Simulation Analysis of a Vision-Guided 5-DOF Robotic Arm System. *ICCK Transactions on Intelligent Cyber-Physical Systems*, 1(1), 51–59.

China's "Made in China 2025" all point towards a core objective: smart manufacturing. In this context, industrial robots serve as the core execution units of automated production lines, and their application depth is growing exponentially [1].

However, traditional industrial robots mostly operate in structured environments, relying on trajectories recorded by teaching pendants or pre-programming to perform repetitive tasks [2]. Once the position, posture, or type of the target object changes, these robots often fail to complete the task. This limitation severely restricts their application in complex scenarios such as logistics sorting and precision assembly. To break through these limitations, introducing external sensors to endow robots with perception capabilities has become inevitable. Machine vision, acting as the "eyes" of the robot, forms a Visual Servo System when combined with manipulator control [3]. This "Eye-Hand" synergy allows the system to perceive environmental changes in real-time and dynamically adjust actions, which is of great theoretical significance and engineering value for reducing production costs and improving efficiency [4, 5].

1.2 Status of Domestic and International Research

Since the birth of the first industrial robot, Unimate, in 1959, manipulator technology has evolved from hydraulic to electric drives and from simple switch control to complex servo control. Currently, international giants like ABB, KUKA, and FANUC dominate the field of six-axis general-purpose manipulators, achieving control precisions within $\pm$0.05 mm. In China, enterprises like SIASUN and ESTUN are rapidly narrowing the gap. However, in the field of lightweight, low-cost robotic arms for light industry and education, problems such as insufficient motion smoothness and poor human-machine interaction persist, necessitating algorithmic optimization [6].

Visual servoing technology is generally categorized into "Eye-in-Hand" and "Eye-to-Hand" configurations. The "Eye-in-Hand" setup, where the camera is mounted on the end-effector, moves with the arm, making it suitable for high-precision alignment tasks, though its field of view is limited. The "Eye-to-Hand" setup fixes the camera externally, providing a global view. While deep learning-based grasping detection (e.g., CNNs) is a research hotspot, traditional image processing algorithms based on color features and geometric moments remain superior for embedded scenarios with limited computing power due to their robustness and low computational load [7].

1.3 Main Contributions

This paper focuses on the development of a low-cost, high-performance vision-guided robotic grasping system. The main contributions of this research are as follows:

Distributed Architecture: A "PC + STM32" distributed control architecture is established to effectively balance the computational demands of complex algorithms with the stringent real-time requirements of motor control. The PC handles high-level tasks such as image processing and inverse kinematics, while the STM32 microcontroller manages low-level servo control and communication.

Kinematic Modeling: The forward and inverse kinematics of the 5-DOF manipulator are derived using the improved Denavit-Hartenberg (D-H) convention. To address the multi-solution nature of inverse kinematics for an under-actuated system, a numerical iterative method is employed, ensuring reliable and rapid convergence to a feasible joint configuration.

Trajectory Optimization: To mitigate mechanical shock and ensure smooth motion—issues commonly encountered with low-cost stepper motors and servos—a quintic polynomial interpolation algorithm is implemented in joint space. This approach guarantees continuous velocity and acceleration profiles, achieving "zero-speed start and zero-speed stop" and significantly reducing system jitter.

Visual Positioning: A monocular vision-based positioning method is proposed, incorporating a linear regression error compensation model. By applying a fixed-height constraint and correcting for systematic errors caused by lens distortion and mechanical flex, the system achieves high-precision 3D target localization without the need for expensive depth sensors.

2 System Overall Scheme Design

2.1 Functional Requirement Analysis

The objective of this study is to design a five-degree-of-freedom manipulator system capable of performing autonomous visual sorting tasks. To achieve this goal, the system must satisfy a set of functional requirements spanning mechanical design, visual perception, and motion control. From a mechanical perspective, the manipulator requires at least five degrees of freedom to approach objects

within the workspace with arbitrary posture. This includes the capability for base rotation, shoulder and elbow pitching, as well as wrist pitching and rotation, thereby providing sufficient dexterity for grasping operations in unstructured environments. In terms of visual perception, the system must be able to capture images of the workspace, effectively remove background noise, and reliably identify target objects based on specific color features, such as red blocks. Furthermore, it must calculate the three-dimensional coordinates of the target in the world frame to enable accurate positioning. Regarding motion control, trajectory planning is essential to ensure continuous velocity profiles during both the start and stop phases, thereby preventing inertial damage to the mechanical structure or the target object and ensuring smooth and reliable operation throughout the grasping process.

2.2 Distributed Control Architecture

Given the high computational load of vision algorithms and the strict real-time requirements of motor control, a single microcontroller proves insufficient for the task. To address this limitation, a distributed architecture is adopted, comprising a host computer for high-level decision-making and a slave microcontroller for low-level execution. The host computer, a PC running MATLAB, is selected for its powerful matrix operation capabilities and rich toolbox ecosystem, including the Robotics Toolbox and Image Acquisition Toolbox, which make it well-suited for inverse kinematics solving and image processing tasks. The slave computer, an STM32F103ZET6 microcontroller based on the ARM Cortex-M3 core, is chosen for its rich peripheral resources, handling the reception of trajectory points, generation of high-precision PWM signals for servos, and pulse sequences for stepper motors [8, 9]. The data flow follows a closed-loop logic of "Perception — Decision — Execution," where the camera captures images and transmits them to MATLAB for processing. MATLAB then extracts the target coordinates, solves the inverse kinematics, and plans a smooth trajectory composed of multiple interpolation points. These points are transmitted frame-by-frame via UART to the STM32, which converts them into electrical signals to drive the motors [10, 11], as illustrated in Figure 1.

2.3 Mechanical Structure and Drive Scheme

The manipulator adopts a serial structure comprising five rotational joints. The base joint (J1), responsible for horizontal rotation, bears the weight and inertia of the entire arm. To ensure stability and repeatability

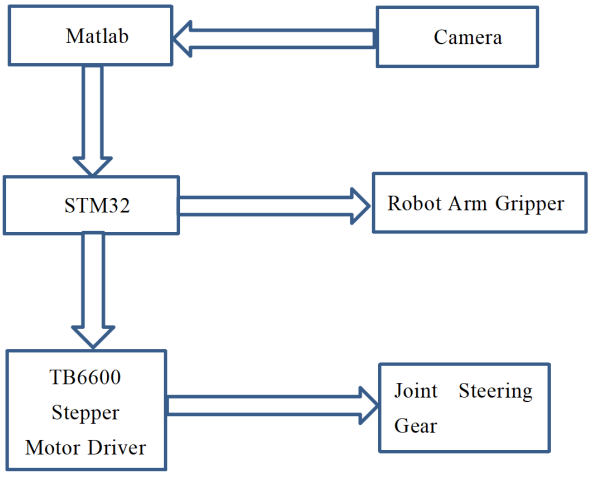


Figure 1. Block diagram of the distributed control system architecture.

under such loads, a high-torque 42 stepper motor is employed in conjunction with a thrust bearing structure. For the arm and wrist joints (J2 to J5), analog and digital servos are utilized to minimize the gravitational load on the cantilever structure. These servos integrate DC motors, reduction gears, and position feedback potentiometers, enabling direct angle control via PWM signals and thereby simplifying the closed-loop control design [12]. The physical experimental platform of the 5-DOF robotic arm is shown in Figure 2.

3 Kinematic Modeling and Trajectory Planning

3.1 Improved D-H Modeling

To describe the geometric relationship of the serial manipulator, the improved Denavit-Hartenberg (D-H) parameter method is used. Based on the physical dimensions of the experimental platform (d_1=10.5 cm, a_2=10.5 cm, a_3=10.0 cm, d_5=15.5 cm), the coordinate systems were established, and the corresponding D-H parameters are presented in Table 1 [13].

Table 1. D-H Parameters of the 5-DOF Manipulator.

Joint i	Twist $\alpha_i - 1$	Link Length $a_i - 1$(cm)	Offset d_i(cm)	Angle θ_i
1	0	0	10.5	q_1
2	$\pi/2$	0	0	q_2
3	0	10.5	0	q_3
4	0	10.0	0	q_4
5	$-\pi/2$	0	15.5	q_5

The transformation matrix ${}^{i-1}T_i$ between adjacent links is derived using standard D-H conventions. The total transformation matrix ${}^{0}T_5$, describing the

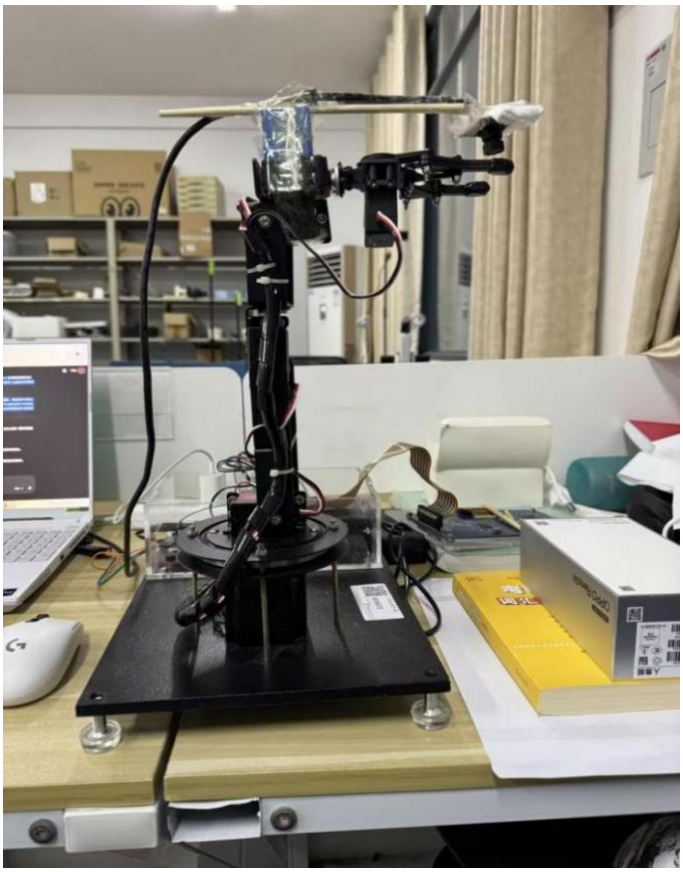


Figure 2. Physical experimental platform of the 5-DOF robotic arm.

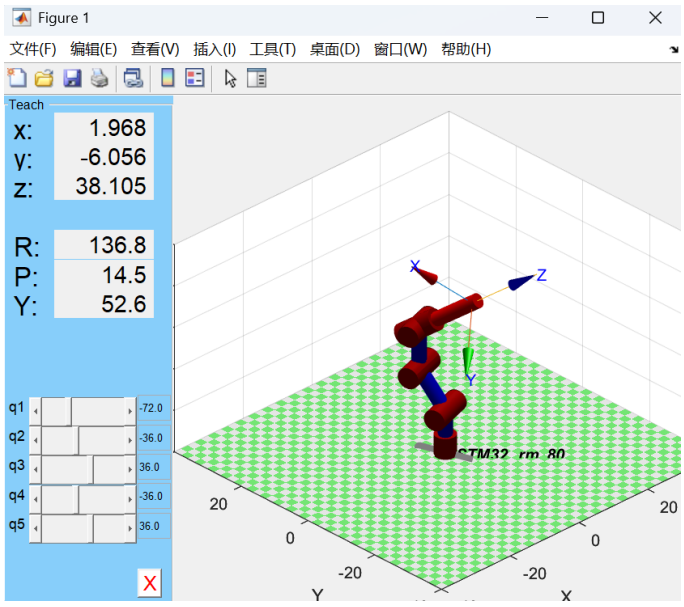


Figure 3. Kinematic simulation model of the 5-DOF robotic arm in MATLAB.

end-effector's pose relative to the base, is obtained by chain multiplication:

$$^{0}T_5 = ^{0}T_1 \cdot ^{1}T_2 \cdot ^{2}T_3 \cdot ^{3}T_4 \cdot ^{4}T_5 \tag{1}$$

Based on the derived kinematic model, a simulation model of the 5-DOF robotic arm was constructed in MATLAB using the Robotics Toolbox, as shown in Figure 3. This simulation environment allows for visualization of the manipulator's configuration under different joint angles and verification of the kinematic calculations prior to physical implementation.

3.2 Inverse Kinematics with Numerical Iteration

Inverse kinematics (finding joint angles q from end-effector pose) is complex for 5-DOF arms as they are under-actuated regarding the full 6-DOF spatial description. Analytical solutions are often difficult to derive. Therefore, this system employs a numerical iterative approach using the **ikine** function in the MATLAB Robotics Toolbox.

Specifically, a mask vector $M = [1, 1, 1, 0, 1, 0]$ is introduced. The mask vector corresponds to the Cartesian DoF sequence $[x, y, z, \theta_x, \theta_y, \theta_z]$. $M = [1, 1, 1, 0, 1, 0]$ indicates that the solver prioritizes the translation (x, y, z) and the pitch angle (θ_y), ignoring roll (θ_x) and yaw (θ_z) constraints. This mask instructs the solver to prioritize the 3D position (x, y, z) and the pitch attitude (rotation about Y) while ignoring constraints on X and Z rotation. This strategy reduces the solution dimension, ensuring rapid convergence to a valid configuration for grasping tasks [14].

3.3 Joint Space Quintic Polynomial Interpolation

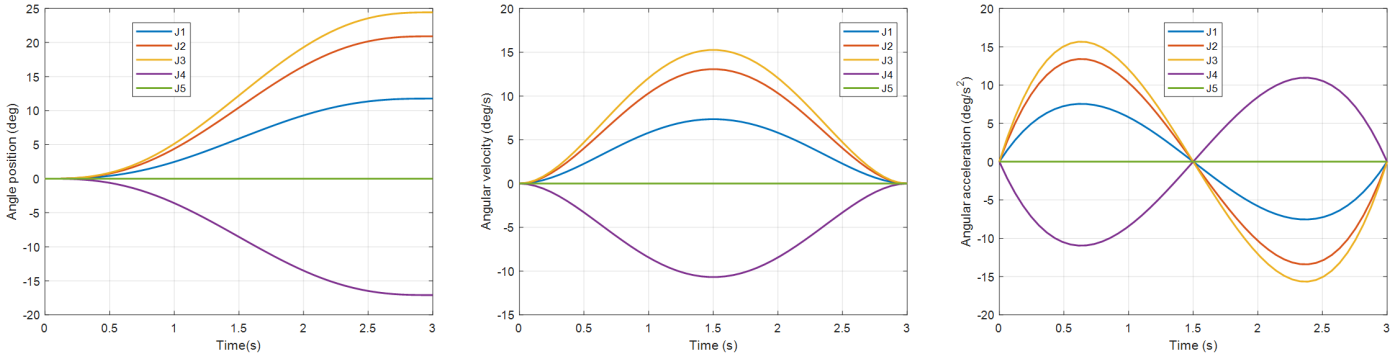
Directly sending the target angles calculated by inverse kinematics to the lower-level controller results in a "step" signal. The servos would attempt to move at maximum speed, causing infinite acceleration spikes, mechanical shock, and potential gear damage.

To solve this, a quintic polynomial interpolation algorithm is applied in the joint space. The angle function for each joint is defined as:

$$q(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 + c_4 t^4 + c_5 t^5 \tag{2}$$

The six coefficients $c_0 \sim c_5$ are determined by six boundary conditions:

- Initial position q_0 at t_0
- Initial velocity $v_0 = 0$
- Initial acceleration $a_0 = 0$
- Final position q_f at t_f
- Final velocity $v_f = 0$



(a) (b) (c)

Figure 4. Kinematic simulation results in joint space: (a) Angle position; (b) Angular velocity; (c) Angular acceleration.

- Final acceleration $a_f = 0$

This formulation ensures "zero-speed start and zero-speed stop." The resulting velocity curve is a smooth bell shape, and acceleration is continuous.

In the software implementation, the motion from start to finish is set to 3.0 seconds. The trajectory is discretized into 150 sampling points with a period of 40ms. MATLAB generates these 150 sets of angles and sends them sequentially, ensuring the servos track a smooth path without jitter [15].

The simulation results of the planned trajectory are shown in Figure 4. As observed in Figure 4(b) and (c), the velocity and acceleration curves are continuous and smooth without mutation. Specifically, the velocity follows a bell-shaped profile with zero velocity at both the start and end points, effectively eliminating mechanical shock.

4 Vision Recognition and Positioning Algorithms

4.1 Camera Imaging Model and Calibration

The system uses the Pinhole Camera Model to describe the projection of 3D points onto the 2D image plane. The relationship involves the camera's intrinsic matrix K, which contains the focal lengths (f_x, f_y) and the principal point (u_0, v_0) [16].

To obtain accurate intrinsics, the MATLAB Camera Calibrator toolbox was used. 15 images of a checkerboard pattern were captured at different angles. The calibration process minimized the reprojection error, yielding precise values for the intrinsic matrix stored in `myCamParams.mat` for runtime use. The calibration process, including the detected checkerboard corners and reprojection errors, is illustrated in Figure 5.

4.2 Image Preprocessing and Segmentation

Denoising: Industrial environments introduce noise. Median filtering is selected over mean filtering because it effectively removes salt-and-pepper noise while preserving the sharp edges of the target object, which is crucial for contour extraction [17].

Color Segmentation: The target is a red block. An analysis of RGB components showed that in the target area, the R component is significantly higher than G and B. A combined threshold logic was designed:

$$\text{Binary}(u, v) = 1, \text{ if } (R > 100) \cap (R > 1.5G) \cap (R > 1.5B) \quad (3)$$

This strict logic prevents false positives from white backgrounds or other interferences.

Feature Extraction: Morphological opening (erosion followed by dilation) is applied to remove small noise artifacts. The `bwareaopen` function removes areas smaller than 300 pixels.

Finally, the centroid $(u_{\text{target}}, v_{\text{target}})$ of the largest connected domain is calculated.

4.3 Coordinate Mapping and Linear Error Compensation

Since a monocular camera loses depth information, 3D reconstruction requires a constraint. The system uses a "Fixed Height Constraint." Since the grasping plane is flat and the object height is known, the vertical distance Z_c from the camera to the object top surface is fixed and known:

$$Z_c = (\text{Look } Z + \Delta\text{offset}) - \text{Block\_Height} \quad (4)$$

Based on similar triangles, the raw physical coordinates $(X_{\text{raw}}, Y_{\text{raw}})$ can be calculated.

However, initial tests showed significant errors. When the object was moved 50 mm physically, the vision

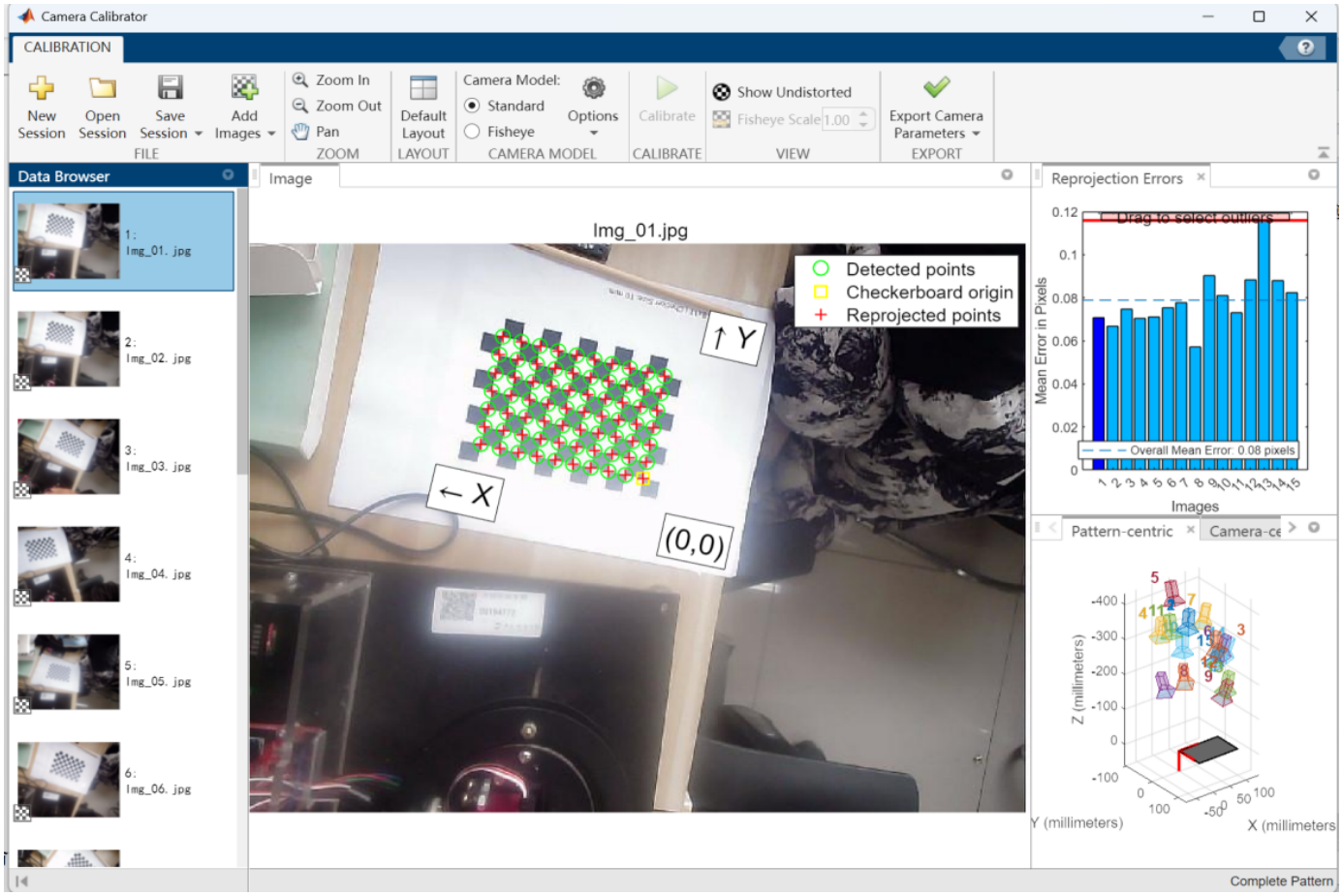


Figure 5. Camera Calibration.

system only measured a change of 35 mm in X and 40 mm in Y. This error is linear and stems from radial distortion and mechanical flex. To correct this, a linear regression compensation model was established based on calibration data:

$$K_{\text{scale\_x}} = D_{\text{real\_x}}/D_{\text{measure\_x}} \approx 50/35 \approx 1.43 \quad (5)$$

$$K_{\text{scale\_y}} = D_{\text{real\_y}}/D_{\text{measure\_y}} \approx 50/40 \approx 1.25 \quad (6)$$

The final corrected coordinates are derived as:

$$\text{Rel\_X} = X_{\text{raw}} \cdot 1.43 \cdot \text{Dir}_X \quad (7)$$

$$\text{Rel\_Y} = Y_{\text{raw}} \cdot 1.25 \cdot \text{Dir}_Y \quad (8)$$

Application of this model reduced the average positioning error from $\pm$15 mm to $\pm$3 mm.

5 System Debugging and Experimental Verification

System debugging is a critical link in translating theoretical models into engineering practice. Following the construction of the hardware and software platforms, this study followed a "local to global, open-loop to closed-loop" principle. The debugging process sequentially verified the lower-level motion control, the upper-lower computer communication interface, the visual perception algorithms, and finally, the integrated autonomous grasping process.

5.1 Lower Computer and Communication Debugging

The primary goal of the lower computer debugging was to ensure that the STM32 microcontroller could generate correct drive signals. Using an oscilloscope, the PWM signals for the servos were verified. Initially, irregular jitter was observed in the servos due to voltage ripple from the USB power supply. This was resolved by integrating a strictly regulated 5V/3A independent power supply and establishing a common ground, stabilizing the PWM period at 20 ms with a linear controllable pulse width between 0.5 ms and 2.5 ms. For the distributed control architecture, communication stability is paramount. The system uses a baud rate of 115200 bps. During the transmission of trajectory points generated by the quintic polynomial interpolation (150 sets of

data), it was found that continuous burst transmission caused buffer overflows in the STM32, resulting in packet loss and motion lag. To address this, a "Time Slice" synchronization mechanism was introduced in the MATLAB transmission module. A forced delay of 40 ms was inserted after sending each frame, matching the 50Hz control cycle of the lower computer. This optimization ensured smooth trajectory tracking without data congestion [18].

5.2 Visual System Debugging and Error Compensation

The visual debugging focused on validating the coordinate mapping model and the effectiveness of the linear error compensation coefficient. First, for color thresholding, analysis of the RGB distribution under laboratory lighting conditions determined that a single component threshold was insufficient. A combined strategy of $(\mathrm{R} > 1.5\mathrm{G}) \bigcap (\mathrm{R} > 1.5\mathrm{B})$ was adopted. As shown in Figure 6, this strategy successfully extracted the red block contours while suppressing white reflections from the background, resulting in a clean binary mask.

Figure 6. Camera recognition of the red target object.

Second, the positioning accuracy was verified. In a "blind grasp" test without compensation at a specific observation height ($Z = 20\,\mathrm{cm}$), the average centripetal deviation was approximately 15 mm due to lens distortion and mechanical flex. After applying the linear regression compensation model derived in Chapter 4 ($K_{\mathrm{scale\_x}} \approx 1.43, K_{\mathrm{scale\_y}} \approx 1.25$), 20 repeated grasping tests were conducted. The results showed that the average error between the end-effector center and the target centroid converged to within ± 3 mm, meeting the grasping tolerance requirements.

5.3 Integrated Autonomous Grasping Experiment

5.3.1 *Experimental Setup*

A comprehensive joint experiment was conducted to verify the system's overall performance. The target object was a red cubic block with a side length of 20 mm. The workspace was defined as a sector area in front of the robotic arm, with $X \in [15, 30]\,\mathrm{cm}$ and $Y \in [-15, 15]\,\mathrm{cm}$. The experiment involved placing the target randomly within this area 30 times. A typical successful grasping sequence, showing the robotic arm approaching, grasping, and lifting the target object, is presented in Figure 7.

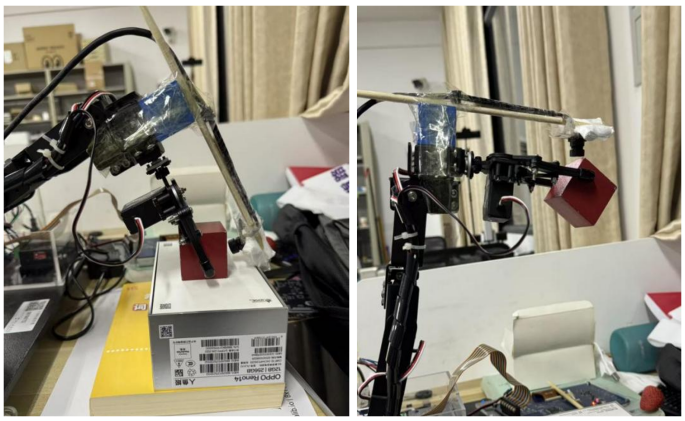


Figure 7. The robotic arm successfully grasping and lifting the target object.

5.3.2 *Results and Analysis*

Out of 30 grasping trials, the system successfully grasped and transported the object 27 times, achieving a success rate of 90%. To validate the statistical significance of this result, the 95% confidence interval was calculated using the Wilson Score method, yielding a success rate range of approximately [74.4%, 96.5%]. This confirms the reliability of the error compensation model.The analysis of the three failed attempts reveals two main causes:Lens Distortion at Edges: When the target was located at the extreme edge of the camera's field of view, the coordinate calculation error exceeded 5 mm due to non-linear distortion, causing the gripper to close without securing the object.Signal Interference: In rare instances, transient electromagnetic interference affected the serial communication, causing the loss of critical trajectory frames, which resulted in minor end-effector jitter that knocked the target over.In conclusion, the experimental results demonstrate that the trajectory planning algorithm ensures smooth motion, and the vision algorithm with linear compensation effectively overcomes monocular positioning errors, proving the system's capability for

autonomous sorting in unstructured environments.

6 Conclusion

To address the limitations of traditional teaching robotic arms regarding poor adaptability and low positioning accuracy in unstructured environments, this paper designed and implemented a 5-DOF visual grasping system based on a distributed control architecture. Specifically, targeting the issues of rigid impact and mechanical jitter inherent in low-cost drive units, a joint-space quintic polynomial interpolation algorithm was proposed. This method successfully ensured continuous velocity and acceleration profiles, effectively realizing smooth motion control and reducing mechanical wear. Furthermore, to overcome the non-linear distortion and lack of depth information in monocular vision, a linear regression error compensation model was developed based on a fixed-height constraint. This approach significantly corrected coordinate deviations, converging the average grasping error from ± 15 mm to within ± 3 mm and achieving a 90% success rate in autonomous sorting tasks. Ultimately, integrated with a time-slice synchronization mechanism, the system achieved a stable and precise "perception-decision-execution" closed loop, validating its practical value for automated sorting applications.

Data Availability Statement

Data will be made available on request.

Funding

This work was supported without any funding.

Conflicts of Interest

The author declares no conflicts of interest.

AI Use Statement

The author declares that no generative AI was used in the preparation of this manuscript.

Ethical Approval and Consent to Participate

Not applicable.

References

[1] Qian, R., Wu, R., & Wu, H. (2025). Research on adaptive control of six-degree-of-freedom manipulator based on fuzzy PID. *Systems Science & Control Engineering, 13*(1), 2498912. [CrossRef]

[2] Xu, T., Han, X., Liu, H., & Li, Y. (2025). EPSO-based rigid robotic arm for obstacle avoidance object grasping. *Scientific Reports, 15*(1), 39462. [CrossRef]

[3] Xin, Z., & Qu, S. (2025). An adaptive fast sliding mode control for trajectory tracking of robotic manipulators. *Automatic Control and Computer Sciences, 59*(1), 102–115. [CrossRef]

[4] Lee, E. A., & Seshia, S. A. (2016). *Introduction to embedded systems: A cyber-physical systems approach*. MIT Press.

[5] Zhang, Q., & Gao, G. Q. (2020). Hand–eye calibration and grasping pose calculation with motion error compensation and vertical-component correction for 4-R (2-SS) parallel robot. *International Journal of Advanced Robotic Systems, 17*(2), 1729881420909012. [CrossRef]

[6] Shen, J., Zhang, W., Zhou, S., & Ye, X. (2023). Fuzzy adaptive compensation control for space manipulator with joint flexibility and dead zone based on neural network. *International Journal of Aeronautical and Space Sciences, 24*(3), 876-889. [CrossRef]

[7] Xu, J., Ren, C., & Chang, X. (2023). Robot time-optimal trajectory planning based on quintic polynomial interpolation and improved Harris Hawks algorithm. *Axioms, 12*(3), 245. [CrossRef]

[8] Tong, Z. (2025). Space motion control optimization of robot arm based on modeling analysis and RRT algorithm. *Advanced Control for Applications: Engineering and Industrial Systems, 7*(4), e70033. [CrossRef]

[9] Bhat, S. P., & Bernstein, D. S. (2000). Finite-time stability of continuous autonomous systems. *SIAM Journal on Control and Optimization, 38*(3), 751-766. [CrossRef]

[10] Yan, L., Liu, Z., Kao, Y., & Jiang, B. (2025). Global sliding mode finite-time control for uncertain robotic manipulators with asymmetric output constraints. *Journal of the Franklin Institute*, 108282. [CrossRef]

[11] Ali, M., Giri, S., Yang, Q., & Liu, S. (2025). Digital twin-enabled real-time control for robot arm-based manufacturing via reinforcement learning. *Journal of Intelligent Manufacturing*, 1-17. [CrossRef]

[12] Zhang, W., Ye, X., & Ji, X. (2013). RBF neural network adaptive control for space robots without speed feedback signal. *Transactions of the Japan Society for Aeronautical and Space Sciences, 56*(6), 317-322. [CrossRef]

[13] Sánchez-Torres, J. D., Gómez-Gutiérrez, D., López, E., & Loukianov, A. G. (2018). A class of predefined-time stable dynamical systems. *IMA Journal of Mathematical Control and Information, 35*(Supplement_1), i1-i29. [CrossRef]

[14] Kim, Y., & Kim, S. (2025). Teaching-Based Robotic

Arm System with BiLSTM Pattern Recognition for Food Processing Automation. *Applied Sciences, 15*(24), 12936. [CrossRef]

[15] Hoang, N. B., & Kang, H. J. (2016). Neural network-based adaptive tracking control of mobile robots in the presence of wheel slip and external disturbance force. *Neurocomputing, 188*, 12-22. [CrossRef]

[16] Chen, Y., Li, X., Chen, J., Yi, C., & Li, Y. (2025). Taylor-Type Direct-Discrete-Time Integral Recurrent Neural Network with Noise Tolerance for Discrete-Time-Varying Linear Matrix Problems with Symmetric Boundary Constraints. *Symmetry, 17*(11), 1975. [CrossRef]

[17] Chen, Y., & Wang, S. (2025). Optimization Control Method of Robot Arm Based on Multi-Agent Reinforcement Learning. *Journal of Circuits, Systems and Computers*, 2650021. [CrossRef]

[18] Moghaddam, M. B., & Chhabra, R. (2026). Lagrange–Poincaré–Kepler equations of disturbed space-manipulator systems in orbit. *Acta Astronautica, 240*, 816–829. [CrossRef]

Hui Zhou is pursuing a Master's degree in Control Engineering at Nanjing Xiaozhuang University. Research Direction: Machine Control and Machine Vision. (Email: 15261498478@163.com)