Application and Future Outlook of UR5 Robotic Arm Sorting Based on ROS

Jiahao Dong[1*]

[1]Tianjin Xinhua High School, Tianjin City, 300203, China

Abstract. With the rapid development of industrial automation, robotic arms have become core equipment in intelligent production lines, and sorting tasks, as a key link in industrial processes, have an urgent need for high-precision and flexible control solutions. This paper focuses on the design, application, and future prospects of a UR5 robotic arm sorting system based on the Robot Operating System (ROS). By integrating hardware architectures such as upper and lower computers, machine vision, and communication protocols, and leveraging ROS-based software tools, the system achieves high-precision sorting of objects. The paper first elaborates on the hardware and software design of the system, then analyzes its practical applications in industrial manufacturing, agricultural product processing, and environmental protection, and finally discusses future development trends such as multi-robot collaboration, artificial intelligence integration, and adaptive control. Experimental results demonstrate that the system exhibits excellent performance in terms of sorting accuracy, real-time responsiveness, and environmental adaptability, providing a reliable technical reference for the intelligent upgrading of sorting tasks in various fields.

1 Introduction

In recent years, the global manufacturing industry has been accelerating its transformation towards intelligence, and robotic arms, as typical representatives of industrial automation equipment, have been widely used in assembly, handling, and sorting processes [1]. Among them, sorting tasks, which involve identifying, positioning, and classifying objects based on attributes such as shape, size, and color, are crucial for improving production efficiency and product quality. However, traditional sorting systems often face challenges such as poor flexibility, low precision, and difficulty in adapting to dynamic environments, which limit their application in complex scenarios.

The Universal Robot 5 (UR5), a 6-degree-of-freedom collaborative robotic arm, is favored in industrial and research fields due to its lightweight design (11kg payload capacity of 3kg, and a working radius of 500mm), high repeatability (±0.1mm), and ease of integration [2]. Meanwhile, the Robot Operating System (ROS), as an open-source robotic development framework, provides a rich set of tools and libraries, including modular software packages, communication mechanisms, and visualization interfaces, which

* Corresponding author: jdong@clarkson.edu

significantly reduce the difficulty of robotic system development and improve code reusability [1]. By combining the UR5 robotic arm with ROS, a flexible and efficient sorting system can be constructed to meet the diverse needs of sorting tasks in different fields.

Additionally, research on the integration of semantic manipulation and visual perception for the UR5 robotic arm provides new directions for the intelligent upgrading of sorting systems. For instance, constructing a semantic knowledge base using Ontology enables the association between object attributes and sorting rules, allowing the robotic arm to independently determine sorting logic in complex scenarios [3]; meanwhile, vision-based reinforcement learning algorithms enable the robotic arm to optimize sorting trajectories through autonomous exploration, reducing reliance on manual programming [4]. The integration of these cutting-edge technologies further expands the application boundaries of the UR5 robotic arm sorting system.

This paper takes the ROS-based UR5 robotic arm sorting system as the research object. Section 2 introduces the system's hardware architecture and software design, including the selection of upper and lower computers, communication protocols, motion planning algorithms, and machine vision integration. Section 3 analyzes the practical application scenarios of the system in industrial, agricultural, and environmental protection fields with reference to specific research cases. Section 4 discusses future development directions such as multi-robot collaboration, AI-driven intelligent sorting, and adaptive control. Section 5 summarizes the research conclusions and points out potential areas for improvement.

2 System design of ROS-based ur5 robotic arm sorting

2.1 Hardware architecture

The hardware of the UR5 robotic arm sorting system based on ROS mainly consists of an upper computer (ROS platform), a lower computer (UR5 robotic arm controller), a machine vision module, and a communication module.

2.1.1 Upper and lower computer configuration

The upper computer adopts a Linux system (Ubuntu 16.04/20.04) running ROS (Kinetic/Melodic), which is responsible for high-level tasks such as object recognition, path planning, and visualization. The ROS architecture includes three core components: the Computation Graph, the File System, and the Open-Source Community. The Computation Graph uses nodes, topics, and messages to realize point-to-point data transmission and processing, ensuring efficient collaboration between multiple functional modules [1]. The lower computer is the UR5 robotic arm controller, which receives control commands from the upper computer and drives the joint motors to complete sorting actions. The UR5 robotic arm is equipped with a parallelogram mechanism to keep the end effector horizontal during movement, and its joints are driven by planetary gearbox stepper motors with a maximum holding torque of 0.25Nm, enabling stable grasping of objects weighing up to 500g [5].

In high-precision control scenarios, modeling the dynamic characteristics of the UR5 robotic arm is critical for improving sorting accuracy. A dynamic model of the UR5 can be constructed using the Subspace Identification (SID) method. This model effectively describes the motion response characteristics of the robotic arm's joints, with a Variance Accounted For (VAF) of 95%–100%, providing an accurate system model foundation for the implementation of Model Predictive Control (MPC) [6]. Additionally, there is an approximately 0.04-second time delay in the joint control of the UR5 robotic arm (corresponding to 4 sampling cycles at a 125 Hz sampling frequency), which needs to be compensated for through predictive control in the control strategy to eliminate its impact on sorting accuracy [6]. Notably, when addressing delay issues, the MPC controller can achieve compensation by predicting the system state for the next 4 steps—a strategy that has been validated in workpiece positioning tasks before curved-surface 3D printing, enabling the tilt angle error between the end effector and the workpiece surface to be controlled within 3.79° [5,6].

2.1.2Machine vision module

To achieve accurate positioning of sorted objects, the system integrates a machine vision module, which typically uses a stereo vision system (e.g., Microsoft Kinect v2) or a monocular camera. The stereo vision system captures left and right images of objects, calculates the depth value by extracting the centroid coordinates of objects in the two images instead of reconstructing the entire disparity map (which reduces computational complexity), and then obtains the 3D coordinates of objects using the pinhole camera model [5]. For example, in agricultural product sorting (e.g./ lettuce sorting), a lightweight improved YOLOv5 algorithm is used to identify the stalks of lettuce, and H-chanel threshold segmentation and morphological processing are performed to extract the optimal grasping points and tilt angles of the lecture [7]. The Point Cloud Library (PCL) is used to process depth data, and filters such as Voxel Grid (VG), Statistical Outlier Removal (SOR), and Radius Outlier Removal (ROR) are applied to remove noise and outliers, improving the accuracy of object positioning [2].

In semantic sorting scenarios, the machine vision module needs to construct a semantic knowledge graph in combination with Ontology to realize the association between object attributes and sorting rules. Object categories are identified using the TensorFlow Object Detection API (e.g., SSD MobileNet COCO model), and the detection results are associated with object attributes (such as "recyclable/non-recyclable" and "material type") in the ontology library to generate semantic sorting instructions [5]. For example, in waste sorting, the system collects object point cloud data using an Intel RealSense camera, segments objects using HSV color filtering and point cloud clustering algorithms, and then determines the recycling category of objects through semantic matching to guide the UR5 robotic arm in completing classification and placement [5]. Additionally, to address target occlusion in complex scenarios, a multi-camera collaborative perception scheme can be introduced, which improves the robustness of object recognition by fusing visual data from different perspectives— a method proven effective in semantic manipulation research at Tampere University [5].

For dynamic sorting scenarios, vision-based reinforcement learning technology can further optimize sorting performance. For instance, a UR5 robotic arm sorting scenario is constructed in the OpenAI Gym simulation environment, and a Proximal Policy Optimization (PPO) algorithm is used to train a sorting trajectory decision model. Indicators such as "grasp success reward," "distance error penalty," and "joint motion smoothness reward" are defined, enabling the robotic arm to achieve a sorting error (ERMS) of 69.3 μm in practical operations—significantly better than traditional PID control [4]. This reinforcement learning-driven control strategy is particularly suitable for sorting tasks where object positions change randomly (e.g., sorting dynamic objects on a conveyor belt), and can achieve stable grasping of moving objects by adjusting joint angles in real time [4].

2.1.3 Communication protocol

The upper computer and the UR5 robotic arm controller communicate via the TCP/IP protocol. The UR5 robotic arm supports multiple communication interfaces, including Modbus TCP ports, Dashboard ports, and upper computer programming ports [1]. The Modbus TCP protocol, as an application-layer protocol based on the OSI model, enables client-server communication between the upper computer and the robotic arm. The upper computer can send script code to the robotic arm through the first client port and obtain robot status information at a frequency of 5Hz; the real-time feedback port (125Hz) allows the upper computer to receive the robotic arm's motion and pose data in real time, providing a basis for dynamic path adjustment [1]. In multi-robot sorting systems, the master controller (e.g., Raspberry Pi) communicates with multiple slave robotic arms (controlled by Arduino) via the Modbus RTU protocol through an RS485 serial interface, enabling synchronous sorting operations [5].

To realize remote monitoring and operation, the system can establish a communication link between ROS and a web interface using the WebSocket protocol. The rosbridge_suite toolkit is used to convert ROS topic data into JSON format, and a web server (e.g., based on the Flask framework) is employed to realize remote visualization of the sorting process and issuance of control commands [3]. For example, operators can access the web interface via a smartphone to view the sorting progress of the UR5 robotic arm in real time and remotely adjust sorting parameters (such as target object coordinates and grasping force), enhancing the system's flexibility and usability [3]. In reinforcement learning training scenarios, the remote monitoring function can also be used to observe the agent's learning process in real time and adjust training parameters (such as learning rate and reward function weight) in a timely manner to prevent the model from falling into a local optimum [4].

2.2 Software design

The software design of the system is based on the ROS framework, integrating functional modules such as motion planning, vision processing, collision detection, and visualization.

2.2.1 Motion planning with movelt

MoveIt is a core motion planning framework in ROS, integrating functions such as motion planning, collision detection, and kinematics solving, and providing a user-friendly visualization interface [1]. For the UR5 robotic arm, MoveIt! uses the Open Motion Planning Library (OMPL) to call path planning algorithms. The Rapidly-exploring Random Tree (RRT) algorithm is selected for trajectory planning due to its ability to efficiently handle high-dimensional space and obstacle avoidance problems [1]. The RRT algorithm constructs a search tree by randomly generating state points in the search space, finds the shortest path from the start point to the target point, and ensures that the robotic arm avoids collisions with itself and the environment during movement [1]. In addition, the system uses Model Predictive Control (MPC) to optimize the robotic arm's trajectory in real time. By predicting the future motion state of objects (e.g., the moving speed of lettuce on a conveyor belt) and dynamically adjusting the robotic arm's motion parameters, the system achieves stable grasping and sorting of moving objects [7].

In reinforcement learning-driven motion planning, the Proximal Policy Optimization (PPO) algorithm can be used to train a sorting trajectory decision model for the UR5 robotic arm. An OpenAI Gym simulation environment is constructed, and reward functions such as "grasp success/failure" and "sorting accuracy error" are defined to allow the agent to learn the optimal sorting trajectory in the simulation [4]. For example, in single-step path planning, the agent directly outputs the target angles of each joint of the UR5, and the trajectory is optimized through reward functions (such as distance error penalty and joint motion smoothness reward), ultimately achieving a sorting error (ERMS) of 69.3 μm—significantly better than the 459 μm error of traditional PID control [8]. For multi-step path planning, the target trajectory can be decomposed into multiple sub-targets, and the final position can be approached step by step through reinforcement learning to reduce the cumulative error of joint motion— a method validated in UR5 reinforcement learning research at Hamburg University of Applied Sciences [4].

2.2.2 Rviz visualization and gazebo simulation

Rviz is a 3D visualization tool in ROS that can display the robotic arm's model, joint states, sensor data, and planned trajectories in real time [1]. The UR5 robotic arm's model is constructed using the Unified Robot Description Format (URDF). SolidWorks is used to draw the 3D model of the robotic arm, and the model is converted to the URDF format using ROS conversion plugins. After compiling the URDF file and checking the correctness of the joints, the model is imported into the ROS workspace. By launching the Rviz node, the 3D model of the UR5 robotic arm can be displayed, and the consistency between the simulated model and the physical robotic arm is verified [1]. Gazebo, as a physical simulation tool, is used to build a virtual sorting environment (including conveyors, objects, and obstacles) to test the system's performance before actual operation, reducing the risk of damage to the physical robotic arm [2].

To simulate curved-surface sorting scenarios, a workpiece model with curved-surface features (e.g., curved conveyor belts, irregular curved parts) can be constructed in Gazebo. The force sensor feedback at the end of the UR5 robotic arm is used to adjust the grasping posture, ensuring stable contact between the end effector and the curved workpiece during the sorting process [6,4]. For example, in pre-sorting positioning for curved-surface 3D printing, Gazebo simulation can simulate the posture adjustment process of the UR5 robotic arm for curved workpieces. MPC control is used to compensate for the dynamic errors of the robotic arm, controlling the tilt angle error between the end effector and the workpiece surface to within 3.79° [6,4]. Additionally, in reinforcement learning training, Gazebo can construct diverse sorting scenarios (e.g., different object placement densities, lighting conditions), and improve the agent's generalization ability through a large number of simulation samples, reducing training costs in the actual environment [4].

2.2.3 Collision detection

Collision detection is crucial for ensuring the safe operation of the robotic arm. MoveIt! integrates the Flexible Collision Library (FCL) for collision detection [1]. The system sets up obstacles in the UR5 robotic arm's working environment (e.g., conveyor belts, partitions), and during the motion planning process, the FCL algorithm checks whether the robotic arm's joints, links, or end effector collide with obstacles. If a collision is detected, the corresponding joint or link is marked in red in Rviz, and the path planning algorithm re-plans the trajectory [1]. In practical applications, the collision detection module can also be combined with machine vision to dynamically update the obstacle map, adapting to changes in the working environment [2].

In semantic collision detection, the size and position information of objects in the ontology library can be used to predict in advance whether there is a collision risk in the motion trajectory of the UR5 robotic arm. For example, the Separating Axis Theorem (SAT) is used to model the robotic arm's links and obstacles with Oriented Bounding Boxes

(OBBs), quickly determining the possibility of collision during movement. Semantic rules (such as "avoid collision with flammable objects") are combined to adjust the sorting path [6,3]. In reinforcement learning scenarios, collision detection results can be integrated into the reward function as a penalty term, enabling the agent to independently avoid collision behaviors during the learning process and improving the safety of the sorting process [4].

3 Application scenarios of UR5 robotic arm sorting based on ROS

3.1 Industrial manufacturing

In industrial manufacturing, the ROS-based UR5 robotic arm sorting system is widely used in tasks such as part sorting, product assembly, and quality inspection. For example, in the automotive parts manufacturing industry, the system can sort different types of bolts, nuts, and washers based on their shape and size. The machine vision module (equipped with a high-resolution camera) captures images of parts on the conveyor belt, uses the improved YOLOv5 algorithm to identify part types with an accuracy of 98% and a recall rate of 95%, and calculates the 3D coordinates of parts using the pinhole camera model [7]. The upper computer plans the robotic arm's trajectory using MoveIt! and sends control commands to the UR5 robotic arm via the TCP/IP protocol, enabling the robotic arm to grasp parts and place them in the corresponding storage bins. The system's average single-frame detection time is 0.014s, which is 50% faster than traditional algorithms, meeting the real-time requirements of industrial production lines [7].

In addition, the system can be applied to the sorting of electronic components. Due to the small size and large variety of electronic components, traditional manual sorting is inefficient and error-prone. The ROS-based UR5 system uses a high-precision vision system (e.g., Intel RealSense D435i) to capture 3D point clouds of components, and PCL filters are used to extract component features. The RRT algorithm plans a collision-free trajectory for the robotic arm, and the end effector (equipped with a vacuum suction cup) grasps components with a positioning accuracy of ±0.1mm, significantly improving sorting efficiency and accuracy [5].

3.2 Agricultural product processing

In agricultural product processing, the system is mainly used for sorting fruits, vegetables, and grains based on attributes such as size, shape, and ripeness. Taking lettuce sorting as an example, the system addresses the problem of random lettuce postures on the packaging line, which affects the packaging process [7]. The machine vision module uses a USB HD wide-angle camera (60 frames/s, 1920×1080 resolution) to capture lettuce images. The lightweight improved YOLOv5 algorithm (with MobileNetV3 as the backbone network and CIOU loss function) is used to identify lettuce stalks, and H-channel threshold segmentation ($105<H<125$) and morphological processing (expansion followed by corrosion) are performed to extract the optimal grasping points and tilt angles of the lettuce [7]. The system dynamically measures the moving speed of lettuce on the conveyor belt (0-50mm/s) and uses MPC to control the robotic arm's motion, ensuring stable grasping of lettuce at different speeds. Experimental results show that the system's average axial error is 0.71cm, average radial error is 1.02cm, and average tilt angle error after adjustment is 3.79°, meeting the requirements of agricultural product sorting [7].

Another application is fruit sorting (e.g., apples, oranges). The stereo vision system (two parallel cameras with a baseline of 60mm) captures fruit images, calculates the 3D coordinates of fruits by extracting centroid coordinates, and uses compactness ($c = p^2/A$, where p is the perimeter and A is the area) and fullness ($F = A_p/A$, where A_p is the area of the minimum enclosing rectangle) to classify fruits by shape and size [5]. The master robotic arm places fruits of different shapes on corresponding conveyors, and the slave robotic arms sort fruits by size, improving the efficiency of fruit packaging and grading [5]. In semantic sorting scenarios, the system can combine an ontology library to establish a "ripeness-color-texture" mapping model for fruits. For instance, in apple sorting, the Intel RealSense camera collects color and depth information of apples, and the HSV color space is used to segment the red, yellow, and green regions of apples. The semantic reasoning module (based on Apache Jena Fuseki [3]) judges the ripeness level of apples, and the UR5 robotic arm sorts them into "fully ripe," "semi-ripe," and "unripe" categories. This semantic-driven sorting method achieves a ripeness judgment accuracy of over 92%, which is 15% higher than traditional color-based sorting [3].

4 Future outlook

4.1 Multi-robot collaborative sorting

With the expansion of production scale and the complexity of sorting tasks, single-robot systems can no longer meet the requirements of high efficiency and large throughput. Future systems will adopt multi-robot collaboration, where multiple UR5 robotic arms work together under the coordination of a master controller. The master controller assigns sorting tasks to each robotic arm based on factors such as task priority, robotic arm status, and working area, and optimizes the task allocation strategy using algorithms such as genetic algorithms and particle swarm optimization to improve overall sorting efficiency [5]. In addition, multi-robot systems will use distributed communication architectures (e.g., 5G) to achieve low-latency and high-reliability data transmission, ensuring synchronous operation of multiple robotic arms [5].

4.2 AI driven intelligent sorting

The integration of artificial intelligence (AI) technologies will further enhance the intelligence level of the sorting system. Deep learning algorithms (e.g., convolutional neural networks, transformers) will be used to improve the accuracy and speed of object recognition. For example, the improved YOLOv8 algorithm can be used to identify objects with complex shapes and occlusions, and the Vision Transformer (ViT) algorithm can extract global features of images, improving the generalization ability of the system [7]. Reinforcement learning will be applied to the robotic arm's motion planning. By rewarding the robotic arm for successful sorting actions and punishing failed actions, the robotic arm can continuously optimize its motion strategy in dynamic environments, adapting to changes in object attributes and working conditions [8]. In addition, the system will use digital twin technology to build a virtual model of the sorting process, simulate and optimize the sorting strategy, and realize the integration of virtual and real operations [1].

4.3 Adaptive control and human-robot collaboration

Future sorting systems will have stronger adaptive control capabilities. The system will use sensors such as force sensors and tactile sensors to perceive the physical properties of objects (e.g., weight, hardness, surface texture) and adjust the grasping force and speed of the end effector in real time, avoiding damage to fragile objects (e.g., fruits, glass products) [5]. In human-robot collaboration scenarios, the system will use human-computer interaction technologies such as voice recognition and gesture control to enable operators to issue sorting commands to the robotic arm in a natural way. The robotic arm will also monitor the operator's position and motion in real time using vision sensors, and reduce its speed or stop moving when the operator is in the danger zone, ensuring the safety of human-robot collaboration [5].

4.4 Energy saving and environmental protection

With the increasing emphasis on environmental protection, future sorting systems will pay more attention to energy saving and environmental protection. The UR5 robotic arm's motor will adopt energy-saving technologies such as variable frequency drives to reduce power consumption. The system will use renewable energy sources (e.g., solar energy) to power the upper computer and sensors, reducing reliance on traditional energy sources [5]. In addition, the system's materials and packaging will be environmentally friendly, and the waste generated during the sorting process will be recycled and treated in a centralized manner, minimizing environmental pollution [8].

5 Conclusion

This paper systematically introduces the design, application, and future outlook of a ROS-based UR5 robotic arm sorting system. The system integrates hardware components such as the UR5 robotic arm, machine vision module, and communication module, and uses ROS-based software tools to realize high-precision, real-time, and flexible sorting of objects. Practical applications in industrial manufacturing, agricultural product processing, and environmental protection show that the system has excellent performance in sorting accuracy, efficiency, and environmental adaptability.

Future research will focus on multi-robot collaboration, AI integration, adaptive control, and energy saving to further improve the system's intelligence level and practical value. The ROS-based UR5 robotic arm sorting system will play an increasingly important role in the intelligent upgrading of various industries, promoting the development of industrial automation and intelligent manufacturing.

References

1. Y. Wang, H. Wang, Y. Dai. Design of UR5 robotic arm control system based on ROS. J.Harbin Vocat. Tech. Coll., (3):99-101(2024)
2. B. Teke. Real-time target tracking and following with UR5 collaborative robot arm. Tampere: Tampere Univ. Technol., (2018)
3. T. Le. Visual perception with UR5 arm towards semantic manipulation. Tampere: Tampere Univ., (2020)
4. M. Stehr. Vision-based reinforcement learning on an UR5 robot arm. Hamburg: Hamburg Univ. of Appl. Sci., (2021)
5. V. D. Cong, D. A. Duy, L. H. Phuong. Development of multi-robotic arm system for sorting system using computer vision. J. Robot. Control (JRC), 3, (5): 690-698 (2022)
6. M. R. de Gier. Control of a robotic arm: Application to on-surface 3D-printing. Delft: Delft Univ. Technol., (2015).
7. R. N. Zhang, W. Lu, X. L. Jian, H. Luo. Intelligent sorting method for assembly line based on visual positioning and robotic arm model predictive control. Int. J. Agric. Biol. Eng., 16, (4): 206-214 (2023).

8. N. I. Giannoccaro, L. Spedicato, A. Lay-Ekuakille. A smart robotic arm for automatic sorting of objects with different tags, In Proc. 4th Imeko TC19 Symp. Environ. Instrum. Measur. Protect. Environ. Clim. Chang. Pollut. Control, Lecce, Italy, 95-98 (2013)