RECOVERING A 4-DOF SCARA ROBOT ARM INTEGRATED WITH MACHINE VISION FOR REAL-TIME PRODUCT CLASSIFICATION

The-Hien Huynh[1], Minh-Vu Huynh[2], Thanh-Hung Tran[1], Hoang-Dung Nguyen[1]*

[1]College of Engineering, Can Tho University, Vietnam
[2]Tien Giang University, Vietnam

*Corresponding author: hoangdung@ctu.edu.vn

(Received: December 05, 2025; Revised: December 24, 2025; Accepted: January 16, 2026)

DOI: 10.31130/ud-jst.2026.24(1).711E

Abstract - This study proposes a real-time automated sorting solution by integrating a recovered 4-DOF robot arm with a machine-vision algorithm. The inverse kinematic computation is integrated in a PLC to control the robot arm. A QR code–based image processing method is used to decode product information, localize target positions, and transmit coordinates to the PLC for the sorting process. Experimental results confirm the successful recovery of the robot arm, achieving average positioning errors of 0.24 mm and 0.12 mm on the X- and Y-axes, respectively. The integrated system effectively classified and sorted three types of QR-labeled product boxes with an accuracy of 93.33% and a processing time of 16 s per product. The proposed potential solution is entirely suitable for small and medium-sized enterprises.

Key words - Classifying; 4-DOF SCARA robot arm; Image processing approach; PLC; Sorting

1. Introduction

The swift expansion of Vietnam's logistics sector has intensified the need for digital transformation and automation solutions that improve operational efficiency and decrease reliance on manual labor. As highlighted by Q.H.Pham et al. [1], logistics companies must choose technologies that are consistent with their financial resources in order to sustain competitiveness. However, the majority of small and medium-sized enterprises (SMEs) - which constitute the predominant segment of the national logistics sector - do not have access to the expensive automated sorting systems typically employed in large distribution centers [2]. This situation highlights the importance of developing scalable, cost-efficient robotic systems tailored to meet local operational limitations [3]. Manual sorting continues to be the primary method employed by many domestic postal facilities, resulting in limited throughput, elevated labor demands, and a higher incidence of sorting errors [4]. A. Oliveira et al. indicate that manual classification procedures represent a significant bottleneck, limiting the capacity of logistics companies to manage increasing parcel volumes [5], affirm that existing workflows continue to lag behind the speed, consistency, and precision required by contemporary supply chains [6]. These limitations underscore the pressing need for streamlined automated solutions that can enhance classification performance without incurring substantial investment costs [7].

Although previous research has investigated SCARA platforms and barcode vision-based control [8], many studies are confined to low-cost prototypes or simulation settings. For instance, R.U.Haq et al. [9] introduce an affordable SCARA model predominantly designed for educational purposes [10], lacking industrial-grade components, such as PLC control with integration features [11], [12]. Such systems are inadequate for small and medium-sized enterprises seeking reliable, continuous-operation sorting solutions [13]. Consequently, a discrepancy remains between academic research and practically deployable industrial systems designed for the Vietnamese logistics environment. This study introduces a fully integrated automated sorting station featuring a refurbished 4-DOF industrial SCARA robot equipped with Mitsubishi MR-J3 servo drives, a Siemens S7-1200 PLC, a Basler industrial camera, and an embedded IPC-based vision system. This solution illustrates the practicality of implementing SCARA platforms in applications that demand precision and rapid cycle times. The SCARA configuration is thus well-suited to meet the demands of automated sorting processes in SMEs, where dependability, affordability, and seamless integration are essential. High-resolution cameras, integrated with real-time image processing, enable precise QR code recognition, object localization, and identification across diverse lighting environments.

2. Methodology

A 4-DOF SCARA robot arm discarded from industry was restored for use in this study. Notably, all components such as power circuits, controllers, and sensors have been removed. This robot arm only has mechanical parts. The goal of this study is to restore and use it with image processing to apply to product classification, one of the important steps in the current logistics sector in Vietnam. Figure 1 illustrates the system for product classification.

- The 4-DOF-SCARA robot arm (1) is recovered from a fully damaged robot by installing all sensors, actuators (AC servo motor drivers), control plants (AC servo motors), and the controller. The controller utilized in this work is a programmable logic controller (PLC) with the S7-1200 series (Siemens, Germany). The control algorithm is developed and embedded into the PLC for controlling both the robot arm and the conveyor belt.

- Vision processing (2) was developed using an

industrial-grade camera (Basler acA3800-14UC, USB 3.0 interface). Image-processing algorithms executed on the industrial embedded computer (KMDA-2602).

- Conveyor belt model (3) is developed to transport the products. It is controlled by using the proportional integral derivative (PID) algorithm.

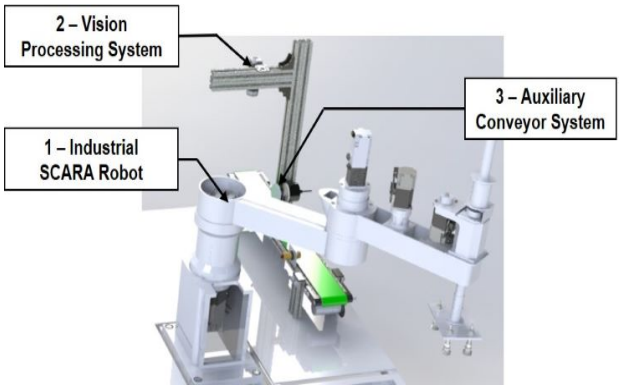

Figure 1. *Product classification system using the restored SCARA robot combined with image processing*

2.1. Recovery of the 4- DOF SCARA robot arm

Figure 2 describes the coordinate axis structure of the 4-DOF SCARA robot. The DH parameters of the SCARA manipulator are addressed from the kinematic diagram [14], [15] and represented in Table 1. Figure 3 presents a coordinate axis structure of the actual SCARA robot. The actual dimensions of the robot are carefully measured and accurately drawn.

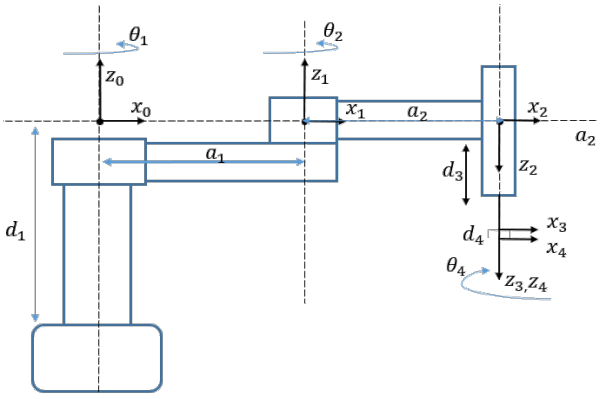

Figure 2. *Coordinate system of the 4-degree-of-freedom SCARA robot*

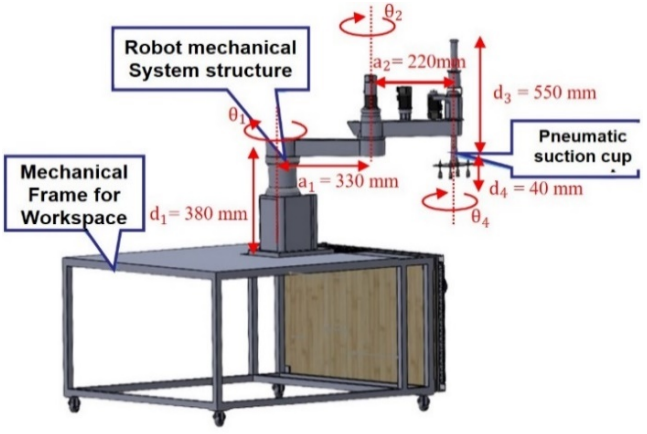

Figure 3. *Actual SCARA robot arm*

Table 1. *DH parameters of the 4 DOF SCARA robot arm*

Joint	θ_i	d_i	a_i	α_i
1	θ_1	d_1	a_1	0
2	θ_2	0	a_2	180°
3	0	d_3	0	0
4	θ_4	d_4	0	0

The end-effector coordinate frame of the robot arm is converted back to the base coordinate frame (derived from the DH parameters presented in Table 1) and computed by the following equation.

$$ {}^{i-1}A_i = \begin{bmatrix} \cos\theta_i & -\cos\alpha_i \sin\theta_i & \sin\alpha_i \sin\theta_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i \cos\theta_i & a_i \sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

Forward kinematic equation

From Equation (1) that overall platform for calculating robot kinematic, by applying the homogeneous transformation matrices corresponding to the rotational matrices that sequentially map each axis back to the base coordinate frame, result obtained as follows.

$$ {}^0 r_E = \begin{bmatrix} x_E \\ y_E \\ z_E \end{bmatrix} = \begin{bmatrix} a_2\cos(\theta_1+\theta_2)+a_1\cos\theta_1 \\ a_2\sin(\theta_1+\theta_2)+a_1\sin\theta_1 \\ d_1+d_3 \end{bmatrix} \quad (2) $$

Inverse kinematic equation

The forward kinematic equation (equation (2)) is expressed as follows, where x_E and y_E are the target coordinates.

$$ x_E^2+y_E^2 = a_1^2+a_2^2+2a_1a_2(\cos(\theta_1+\theta_2)\cos\theta_1 + \sin(\theta_1+\theta_2)\sin\theta_1) \quad (3) $$

where

$$ \begin{cases} x_E = (a_1+a_2\cos\theta_2)\cos\theta_1 - a_2\sin\theta_2\sin\theta_1 \\ y_E = a_2\sin\theta_2\cos\theta_1 + (a_1+a_2\cos\theta_2)\sin\theta_1 \end{cases} \quad (4) $$

Therefore, angle θ_1 for robot inverse kinematics is computed as follows.

$$ \sin\theta_1 = \frac{a_1y_E + a_2(y_E\cos\theta_2 - x_E\sin\theta_2)}{x_E^2+y_E^2} \quad (5) $$

$$ \cos\theta_1 = \frac{a_1x_E + a_2(x_E\cos\theta_2 + y_E\sin\theta_2)}{x_E^2+y_E^2} \quad (6) $$

And the angle θ_1 is addressed by using equations (5) and (6) as follows.

$$ \theta_1 = \arctan\left(\frac{\sin\theta_1}{\cos\theta_1}\right) \quad (7) $$

Similar to the way to get the angle θ_1, the angle θ_2 for robot inverse kinematics is computed as follows.

$$ \cos\theta_2 = \frac{a_1x_E + a_2(x_E\cos\theta_2 + y_E\sin\theta_2)}{x_E^2+y_E^2} \quad (8) $$

From equation (8), the angle θ_2 for robot inverse kinematics is computed as follows.

$$\theta_2 = \arccos\left(\frac{x_E^2 + y_E^2 - (a_1^2 + a_2^2)}{2a_1a_2}\right) \quad (9)$$

2.2. Designing control hardware components

To control the axes of the 4-DOF SCARA robot arm, the hardware component (e.g., sensors, the servo motor drives, AC servo motors, pneumatic valves, and a PLC controller) and software components (e.g., PLC firmware, image processing algorithm, and computer software) are designed. Figure 4 describes the diagram of hardware components for controlling the robot arm. The push buttons (1) and limit and optical switches (2) are sensors utilized for detecting the commands from an operator and limiting the operating space of the robot arm to ensure the safety of both the operator and the equipment around. An Industrial PC (IPC) is utilized for embedded programming and executing image-processing algorithms, as well as communicating with the PLC via the Ethernet/IP protocol. The PLC S7-1200 controller (4) controls and navigates the robot arm using input information from sensors and the operator: Controlling the AC servo motors for the 4-DOFs of the robot arm and pneumatic valves, and monitoring the operation status of the robot arm. Mitsubishi MR-J3 Driver (5) converts control signals from the PLC S7-1200 to control the joints of the robot arm and other actuators, such as pneumatic valves and AC servo motors (6), effectively ensuring the proper interaction between the various components of the classification system. Additionally, the algorithms embedded in the PLC S7-1200 could transport, assembly, inspect, classify, and pick up the products.

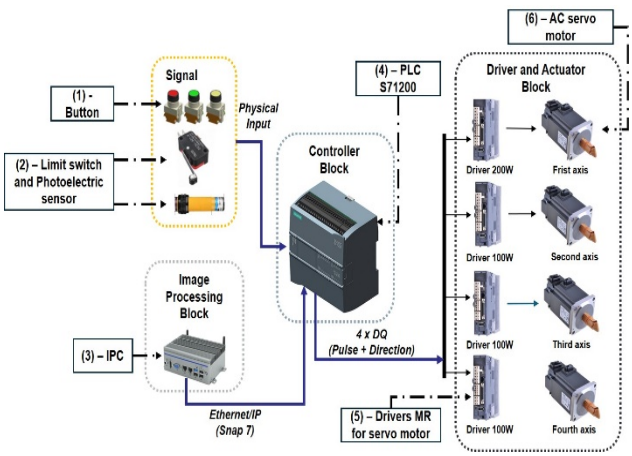

Figure 4. *Diagram of hardware components for controlling the robot arm*

Figure 5 presents the overall control panel layout of the robot arm. It consists of three main blocks: the power supply, control, and actuator units. Two 24V power supplies (AllenBradley's power supply is utilized for sensors and PLC, while Meanwell is utilized for the servo motor drivers, pneumatic valves, and conveyor belt) are utilized for the power supply unit to entirely distribute sensors and actuators as well.

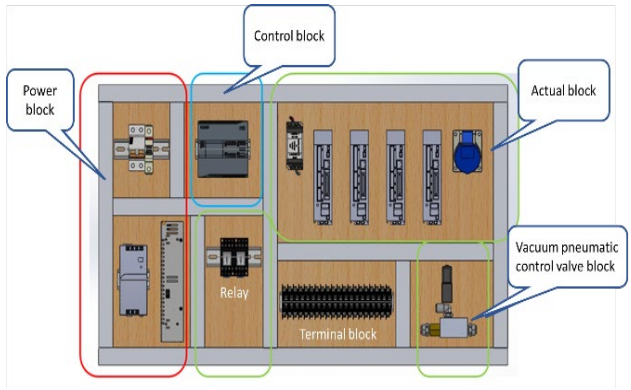

Figure 5. *The control panel layout of the robot arm*

Figure 6 illustrates the actual control panel for controlling the robot arm. It is worth noting that the S7-1200 PLC with the embedded algorithms entirely controls the operation of the system: enabling synchronous and precise control of all 4-DOF SCARA robot axes from 4 servo drives, a 3/2 solenoid valve, and a vacuum valve to perform pneumatic suction and exhaust for pick-and-place operations. Moreover, the entire system is designed following key IEC-60617 standards, ensuring ease of maintenance, technical servicing, and future system expansion.

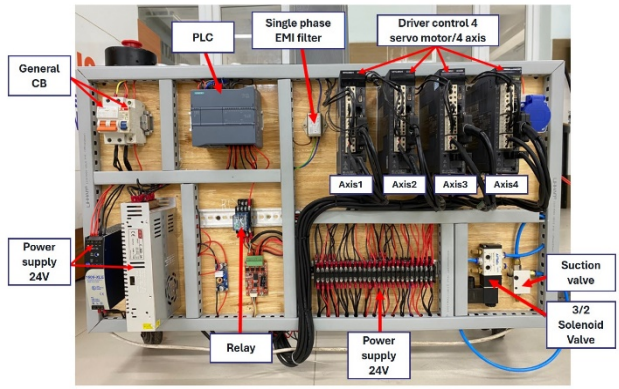

Figure 6. *Actual control panel for controlling the robot arm*

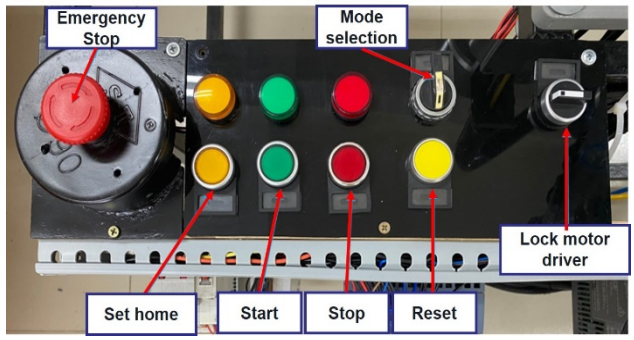

Figure 7. *Console panel for operating the robot arm*

Figure 7 illustrates the operation panel equipped with function buttons: emergency, start, stop, reset, and set home. The emergency stop button allows immediate shutdown of the entire system in case of any accidents. The set home, start, stop, and reset buttons allow the operator set up initial positions of the robot axes, starting, stopping, and restarting the system, respectively.

Figure 8 presents the diagram of the QR code image acquisition system of the product for image processing and conveyor control tasks. The Basler acA3800-14UC industrial camera (10 megapixels resolution, a frame rate of 14 FPS, and using a USB3.0 port) is utilized to acquire image data of QR codes attached to the products for classification based on the computer vision algorithms.

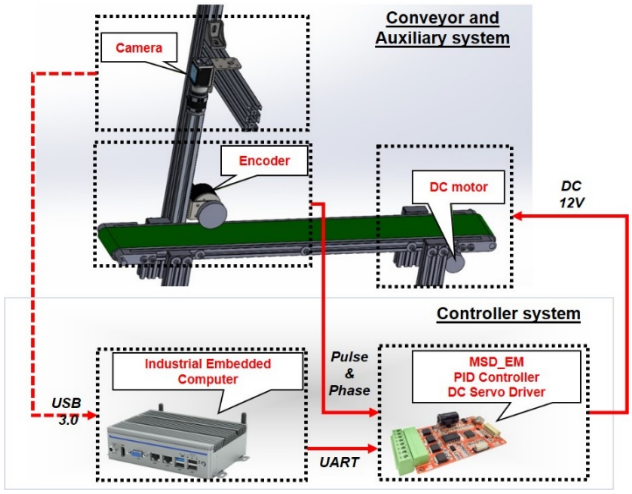

Figure 8. *Diagram of the QR code image acquisition system*

The conveyor (80 cm long, 17 cm wide, speed of 8 cm/s) includes a DC motor and an encoder (400-pulse and two-phase encoder with two phases) that provides the speed feedback signal for a PID controller (C-Smart Vietnam [16]). This solution allows for controlling the set speed (this value is transmitted and received via the UART protocol from the industrial embedded computer), ensuring the system maintains the exact speed regardless of varying cargo load volumes.

The project aims to optimize the system and coordinate seamlessly with the robot arm during pick-and-place without stopping the conveyor. DC Tuner software is installed for the industrial PC to control conveyor speed, and PyCharm software with the Python language is utilized for image processing.

2.3. Designing software components

2.3.1. Control algorithm for the robot arm

Compatibility configuration between the driver and the PLC

Table 2 presents the configuration parameters between the PLC, driver, and mechanical drive system of each robot axis to ensure compatibility during the control process. Specifically, PLC provides the maximum number of output pulses for each axis while the driver receives and processes the corresponding input pulses to control the servo motor. At the same time, the gear ratio of each gearbox reflects the relationship between the number of control pulses and the actual mechanical movement. Based on these parameters, the "target Pulse/Degree" value is calculated to represent the control resolution of each axis. The gear ratios for each robot axis were measured and calculated experimentally.

By controlling each servo motor individually (after tuning and ensuring stable and accurate system tracking for each rotation), they were then installed after the gearbox and controlled incrementally (in a safe and controlled manner) until the robot's end-effector joint rotated exactly one revolution.

Table 2. *Parameter configuration of the PLC controller and AC motor driver for each robot axis*

Axis	Configuration		*Mechanical configuration (Gearbox)*	*Target (Pulse/ Degree)*
	PLC	Servo driver		
1	20,000	3,600	1/80	*80*
2	20,000	3,600	1/50	*80*
3	1,300	1,000	1/2	*5.56*
4	1,000	100	1/21	*58.38*

An algorithm embedded in the S7-1200 PLC for controlling the robot arm

Algorithm for controlling robot to sort product based on QR Code

1. **Begin main program**
   Set up the system variables and verify safety conditions.
2. To set the safety variables, have to click the Reset button.
3. **while** no mistakes or disruptions, and an emergency stop then
4. **if** Home button is pressed **then**
5. **if** The limit switch has been turned on **then**
6. Setting the initial joint value
7. Setting the status to "installed"
8. **if** *Inverse Kinematics mode* is selected **then**
9. Interlock JOG mode
10. Initializing parameters
11. Setting parameters for forward kinematics
12. Setting inverse kinematics settings
13. Calculating for theta angles by inverse kinematics function (using SCL language)
14. **else if** *Forward Kinematics mode* is selected **then**
15. Putting each joint follow to user's target.
16. Save the position to memory (for kinematic testing or products classification).
17. **else if** *Auto mode* is selected **then**
18. Interlock in the manual mode
19. **if** Start button is pressed (Trigger) **then**
20. Moving to idle position for waitting to pick.
21. Getting data from the camera: coordinates, deviation angle, QR code for sorting.
22. **if** Sensor is affected **then**
23. The robot move to the pick production
24. **if** code value received camera <= 3 **then**
25. Placing each product to position correctly
26. **else** checking data received from camera
27. **end if**
28. **end if**
29. **else**
30. **if** Stop button is pressed (Trigger) **then**
31. Pausing the system
32. **end if**
33. **end if**
34. **else** *Manual mode* is selected **then**
35. Interlock the auto mode
36. **for** N = 1, N<=3, N++
37. **if** Button for N position is pressed **then**
38. Moving to the N position correspondingly
39. **break**
40. **end if**
41. **end for**
42. **end if**
43. **else** Find home position
44. **end if**
45. **else** Keep checking until setting home have been done
46. **end if**
47. **end while**
48. **end program**

Figure 9. *Control algorithm for the product classification based on the QR code integrated in the PLC to control the robot arm*

Figure 9 describes the control algorithm for the product classification based on the QR code. After starting, the initial parameters are set. If the "set home" button is activated directly by the operator, the initial position of the

robot axes is set. At this point, the signal it receives will compare the current motor position with the desired position, thereby determining the appropriate rotation direction and controlling the motors.

After setting the initial position for the robot axes, *JOG* or *Kinematic* mode is chosen to control and set it as the robot's working position. Once the robot positions have been set up, the operation mode is chosen, such as *Auto* or *Manual*. Manual mode allows us to check if the positions are correct by interacting with the position buttons on the WinCC-based graphic user interface (GUI). In the Auto mode, the start button is pressed to begin operating, and the Stop button is pressed to halt the operation of the robot arm and conveyor belt. The Reset button allows the parameters be set to their initial values.

2.3.2. GUI for controlling and monitoring the product classification

Figure 10 illustrates the main GUI designed using WinCC platform for configuring, controlling, and monitoring the position of a 4-DOF SCARA robot. The GUI integrates for forward and inverse kinematic computation (see Section 2.1), enabling intuitive visualization of joint angles and workspace coordinates (see areas (1) and (2) of Figure 10), thereby supporting motion-accuracy verification. The area (3) of Figure 10 provides a position selection and storage section, designed to perform tasks related to sorting, classifying, and warehousing parcels in the logistics field.

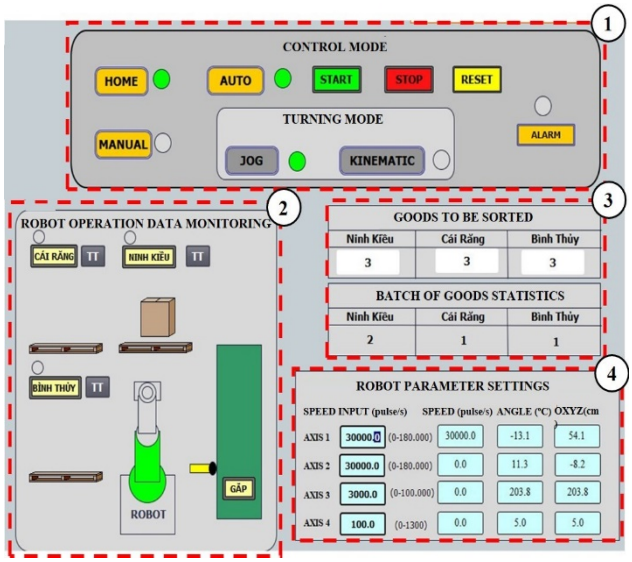

Figure 10. *Main GUI designed using WinCC*

The overall control GUI of the vision-integrated pick-and-place robot arm is shown in Figure 11. The interface comprises four functional control tabs as follows.

Section (1): the operation-mode control panel, including Manual, Auto, Start/Stop, Jog, and Kinematic modes, enabling flexible switching between manual manipulation, fully automatic operation, and kinematic-model-based control.

Section (2): the real-time operation simulation, displaying the current states of the robot, pallet, and product positions.

Section (3): statistical information on the number of products to be sorted and the quantities already processed at each designated position, supporting the performance evaluation and batch-handling planning.

Section (4): configuration of robot joint velocities and angular parameters with higher resolution, ensuring precise motion execution and maintaining synchronization with the conveyor under varying load conditions.

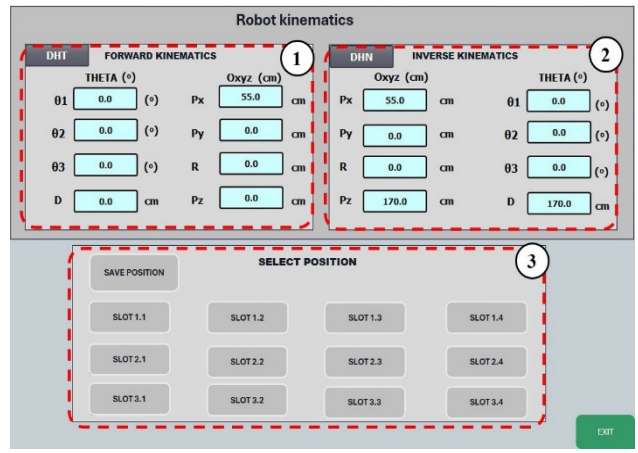

Figure 11. *Controlling and monitoring the GUI for product classification and sorting*

2.3.3. Image processing algorithms

The image processing algorithm based on Python is shown in Figure 12 using a straightforward method and described as follows.

At the starting time (1), system initialization includes setting up libraries, initializing variables for data storage, connecting the camera to the PLC, and addressing the frame size. At the loop beginning time (2), if the camera is not detected, the system waits until one is available. If a product box (size of 150mm×120mm×130mm and weight of 2.5 kg) is detected, the RGB image is converted to HSV and then to grayscale images. After performing these procedures, the next step is to check whether the product box is within the pre-calculated color range. If it is out of the color range specified by the programmer, the system returns to the detection step. In contrast, if the product box is within the correct color range, the centroid coordinates, deviation angle, and deviation direction of the product box relative to the line perpendicular to the conveyor belt is computed.

After the coordinate location code of the product box is computed, its QR code is processed. At this time, if the QR code of the product is not detected, it moves to the next product box. On the other hand, if a QR code is detected, a bounding frame is drawn around it, and the QR codes are addressed. In the current work, products with three different addresses are assigned QR codes. This QR code is compared with three stored QR codes. If it is matched with the stored QR code, a code is sent to the PLC for controlling the robot arm to put the correct position of the product box (the coordinate values, tilt angle, and direction of the product box). The Snap7 protocol with an open-source library for Python is utilized to communicate with the S7-1200 PLC [17].

Algorithm for image processing program: detection QR code from the package and interface with the PLC.

```
1.  Begin main program
        Import libraries and initialize variables
        Establish a connection to the PLC (SNAP7 TCP/IP)
        Activate and exhibit camera
        Modify frame for size fitting
2.  while Not user interrupt or failure system do
3.      Grayscale conversion, thresholding, and background
4.      Subtraction facilitate object detection inside image frame
5.      while An object is present within photo frame do
6.          if Identifying the barrel then
7.              Transform RGB color to HSV and grayscale
8.              if Maximum contour > a (high limited thresshold) then
9.                  if Maximum contour < b (low limited thresshold) then
10.                     Determine coordinates of barrel,
11.                     Determine deviation and orientation of barrel
12.                     if QR code recognition then
13.                         Interpret QR code data. Encapsulate it with a
                            frame and enumerate the QR codes
14.                         if "Ninh Kiều" code then
15.                             Enumerate the Ninh Kieu codes
16.                             Transmit value code #0001 to the PLC
17.                         else if "Bình Thủy" code then
18.                             Enumerate the Bình Thủy codes and
19.                             Transmit value code #0002 to the PLC.
20.                         else if "Cái Răng" code then
21.                             Enumerate the Cái Răng codes
22.                             Transmit value code #0003 to the PLC.
23.                         else Assign the value code to #0000
24.                         end if
25.                         Transmit the barrel center coordinate of box
26.                         Deviation angle and direction to the PLC.
27.                         Assign the value code to #0000
28.                     else Error: The barrel lacks a label.
29.                     end if
30.                 else Error report: box under size or small objects
31.                 end if
32.             else Error report: box over size or light noise
33.             end if
34.         else end while
35.         end if
36.     end while
37. end while
38. End main program
39. Return errors
```

Figure 12. *Python based algorithm for detecting the boxes, the QR code integrated in the IPC, and interfacing to the PLC*

3. Results and discussion

3.1. Recovered robot arm

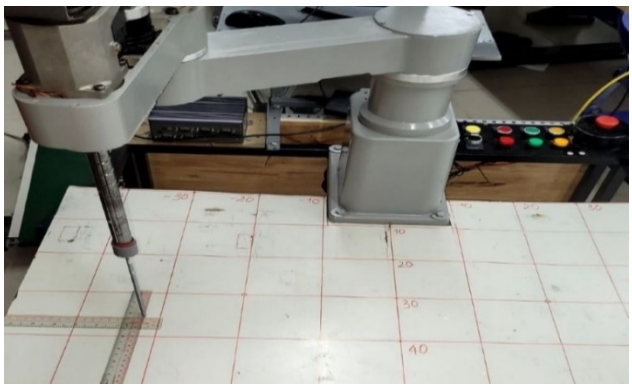

Figure 13. *The recovered 4-DOF SCARA robot arm*

The 4-DOF SCARA robot arm has been successfully recovered (see Figure 13). Figure 14 presents the working plane of the robot arm. In this scenario, the inverse kinematics algorithm is utilized to control the SCARA robot arm in the real world (see Section 2.1). To evaluate the accuracy of the recovered robot arm, the actual X and Y axes movement values of the robot arm without any load were compared with those of the expected distance values (see Figure 15). Table 3 describes the experimental results of 10 coordinates and their errors, respectively. It can be seen that the average errors of X- and Y coordinates are 0.24 mm and 0.12 mm, respectively. Additionally, the standard deviation of X- and Y axes is 0.17 mm and 0.08 mm, respectively.

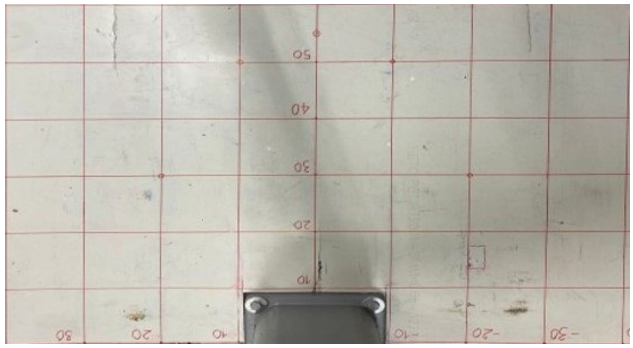

Figure 14. *The configuration of the SCARA robot arm with the working plane*

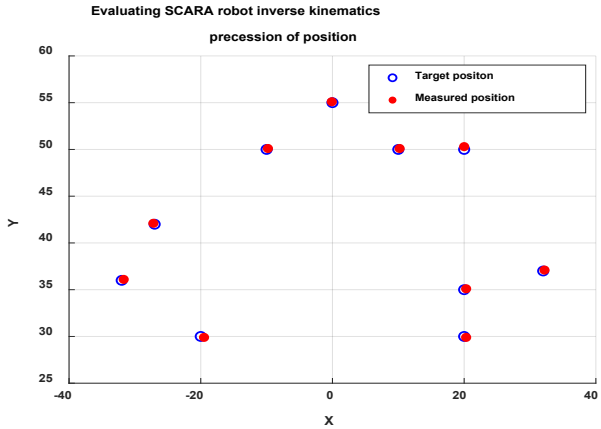

Figure 15. *Comparison of the expected and actual coordinates of the robot arm using the inverse kinematics algorithm*

Table 3. *Accuracy evaluation of X and Y axes*

No.	Expected value (mm)		Actual value (mm)		Error (mm)	
	X	Y	X	Y	X	Y
1	20	30	20.3	29.9	0.3	0.1
2	-20	30	-19.5	30.1	0.5	0.1
3	20	35	20.3	35.1	0.3	0.1
4	-32	36	-31.7	35.9	0.3	0.1
5	32	37	32.2	37.1	0.2	0.1
6	-27	42	-27.2	41.9	0.2	0.1
7	20	50	20.1	50.3	0.1	0.3
8	10	50	10.2	50.1	0.2	0.1
9	-10	50	-9.8	50.1	0.2	0.1
10	0	55	0.1	54.9	0.1	0.1
Average			**-**	**-**	**0.24**	**0.12**
Standard deviation			**0.17**	**0.08**	**0.11**	**0.06**

The recovered 4-DOF SCARA robot arm, combined with the image processing approach and a conveyor belt, was applied to classify and sort three kinds of product

boxes with their QR codes. The destinations/addresses (e.g., NK for Ninh Kieu ward, CR for Cai Rang, and BT for Binh Thuy) are encoded with QR codes.

3.2. Mapping from camera coordinates to robot coordinates

After recovery, the robot arm's reverse kinematics are calculated to determine the position of X and Y coordinates using Equations (7) and (9). A PLC is utilized to control the robot arm to move to the desired coordinates. They include 2 coordinates: (1) the coordinate of the product box on the conveyor belt identified for NK, CR, BT destinations and moved to the picking area; (2) the coordinate in which known desired areas are utilized for storing the product boxes. A camera connected to an IPC is utilized to decode address/destination information (e.g., NK, CR, and BT) from a QR code using an image processing algorithm. By using the mapping algorithm to transfer product-box coordinates from the camera to the robot's coordinates, combined with a synchronization mechanism between the conveyor belt and the robot arm, to determine the position of the product box. Therefore, the robot arm can pick up and drop the production box in the correctly expected position.

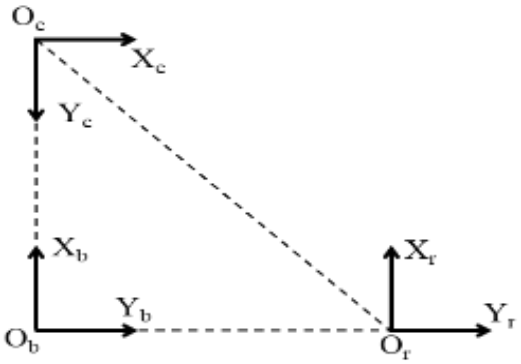

Figure 16. *Mapping the camera's coordinates to the robot's coordinates*

To pick up and drop the product boxes correctly, the camera's coordinates need to be mapped to the robot's coordinates. Figure 16 illustrates the mapping coordinates between the camera and the robot arm. (X_b, Y_b) represents the coordinates of the product box on the conveyor belt, (X_c, Y_c) refers to the read coordinates from the camera, and (X_r, Y_r) stands for the robot's coordinates. Initially, the camera is attached to the Z-axis of the robot arm for observing the entire working area where the product boxes are picked. Multiple coordinates are taken to find the relationship between the camera coordinates (pixels) and the robot coordinates (mm). Then, the available curve fitting tool in Matlab is utilized to find the X_r and Y_r coordinates of the robot arm. Finally, the camera is moved from the picking area to the beginning position of the conveyor belt to detect the product box, the center of the product box, and the QR code. This equation is programmed on the embedded IPC. These computed coordinates are sent to the PLC to control the robot arm for accurately picking the product box. The coordinate equations of the robot arm are addressed as follows.

$$X_r = \left(X_c + \frac{400}{2} \right) \times 0.0525 + 33 \quad (10)$$

$$Y_r = \left(Y_c + \frac{600}{2} \right) \times 0.0503 - 33 \quad (11)$$

3.3. Classifying and sorting the product based on the image processing algorithm

For performance evaluation, the experiment was repeated 10 times (3 products per group and 3 groups with QR code locations of NK, CR, and BT). Therefore, the total number of actual experimental samples in this study is 90. The criterion for evaluating the correct classification of the proposed approach is based on simultaneously meeting the following two conditions: (1) correctly recognizing the QR code (classification) and (2) the robot moving to pick/drop in the correct target location (sorting). If either of these conditions is not met, the classification result is considered incorrect. The QR code recognition library (pyzbar) available in Python is utilized to decode QR code content (character string). Then it is mapped to a specific address/destination using a lookup table, such as NK for Ninh Kieu ward, CR for Cai Rang, and BT for Binh Thuy.

To accurately pick up the product box, the robot arm is synchronized with the conveyor belt speed and the position of the product box. The synchronization mechanism is deployed by using the time-based box position prediction method, $x(t) = x_0 + v \times \Delta t$, where $x(t)$ refers the center position of the product box, v represents the conveyor belt speed controlled by the PID controller, x_0 stands for the position of the product box determined in the image processing frame, and Δt is the delay time from the moment of the product box leaving the image processing area (detected by the optical sensor) to the robot's picking area.

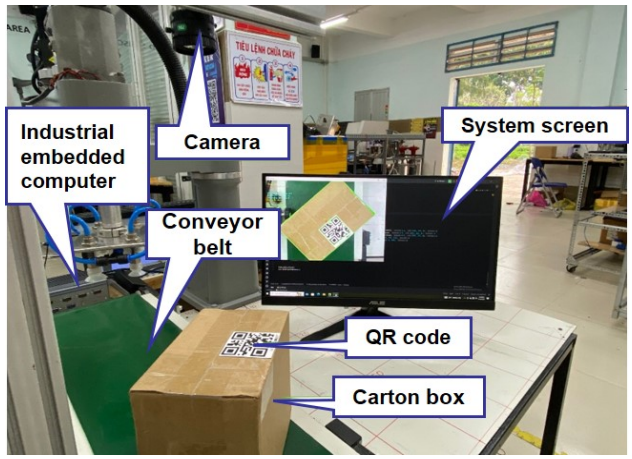

Figure 17. *An experimental model for classifying and sorting the product boxes using the image processing algorithm*

Figure 17 illustrates an experimental model for classifying and sorting the product boxes using the image processing algorithm. The QR codes on the product boxes are captured by the camera, and then edge detection and QR code recognition algorithms are applied to accurately determine the position and orientation of the product box

on the conveyor belt. Its position and orientation are sent to the PLC in real time for picking up and sorting the product box.

Figure 18 illustrates the Identification and classification results of the products based on the QR code image processing algorithm. The green outlines represent the segmented object regions, while the central markers indicate the calculated reference coordinates.

The conveyor speed is adjusted with various values (e.g., 18, 12, and 8 cm/s) to evaluate the effectiveness of the image processing stage in the case of the light intensity and the color of the product boxes is not changed. The experimental results demonstrated that if the conveyor speed increases, the identification accuracy of the products based on the QR code image processing algorithm decreases. The suitable conveyor speed for the image processing algorithm is 8 cm/s. And the accuracy in this scenario is 95% (see Figure 19).

Figure 18. *Identification and classification results of the products based on the QR code image processing algorithm*

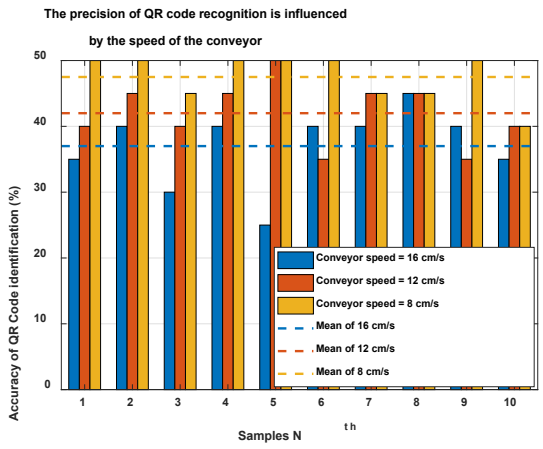

Figure 19. *QR code recognition with different conveyor speeds*

For further evaluation, the following experiment is deployed: The conveyor belt runs continuously while the robot arm tracks, picks up, and sorts products. In this experiment, three addresses (Ninh Kieu, Cai Rang, and Binh Thuy wards) are utilized to create 3 different QR codes. The products are put on the conveyor randomly. The experiment is repeated 10 times. Figure 20 presents the classifying and sorting results of the 3 aforementioned products ((1) for picking up location, (2) for Cai Rang, (3) for Ninh Kieu, and (4) for Binh Thuy). The experimental result revealed that the robot arm could classify and sort the products with an average accuracy of 93.33% (see Table 4), and the average classifying and sorting time is 16 s/product (see Figure 21 and Table 5). It is noted that QR code recognition and product grasping-holding stages made errors, while all stages left were not. Additonally, the average classifying and sorting time is the duration of the entire cycle including (1) placing the product on the conveyor and moving it to the image processing position; (2) processing the image for detecting the box, extracting the QR code information, calculating the center coordinates and tilt angle of the box, and transmitting data to the PLC; (3) controlling the robot to pick and place the product boxes according to the delivery area (see Table 5 describes the estimated time of each stage).

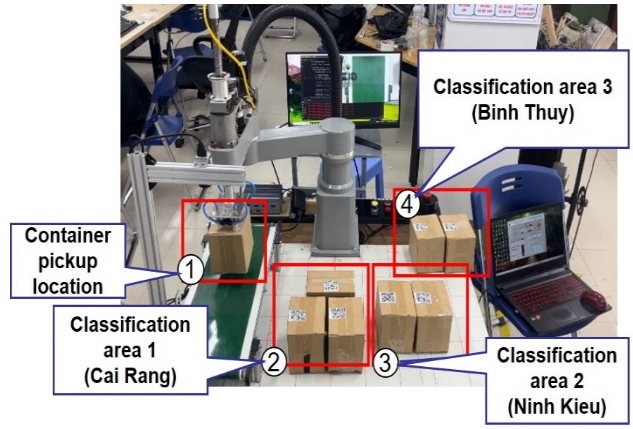

Figure 20. *Classifying and sorting the results of three products with different QR codes*

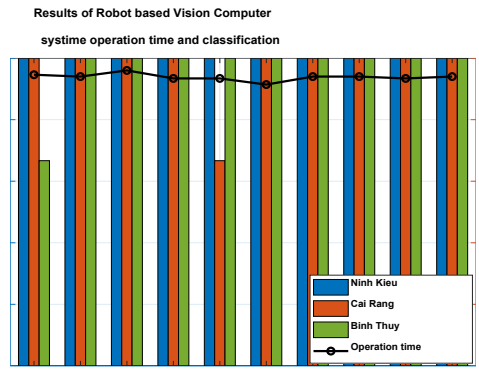

Figure 21. *Classifying and sorting results of 3 products with 3 different QR codes*

Table 4. *The average accuracy of the group of three product boxes and each product box*

Classification	QR code identification	Pick / hold product	Average
Group of the product box	28/30 (93.33%)	28/30 (93.33%)	93.33%
Product box	88/90 (97.78%)	88/90 (97.78%)	97.78%

Table 5. *The estimated time of each stage for classifying and sorting the product box*

Stage	Activity	Estimated time (s)
Image processing	Latency time for image capture	0.2
	Product box detection	1.0
	QR Code identification	0.5
	Sending the data to PLC	0.1
Latency time	Moving the product box from the camera to the picking-up area	3.5
Robot movement and picking up the product box	Inverse kinematic computation (Mathematical instructions)	0.05
	The pulse generation instructions related to absolute movement and the latency time of the driver response	0.15
	Robot controlled for picking up the production box	1.5
	Downward movement (The Z axis of the robot arm)	0.5
	Activating the vacuum suction	0.15
	Upward movement (The Z axis of the robot arm)	0.5
Robot movement and product-box placement	Robot controlled to move to the target areas (Ninh Kieu, Cai Rang, and Binh Thuy)	6.5
	Downward movement (The Z axis of the robot arm)	0.75
	Deactivate the vacuum suction	0.1
	Upward movement (The Z axis of the robot arm)	0.5
Total		16.0

It is noted that the communication protocol between the IPC and the PLC is TCP/IP under a Server/Client architecture. In the proposed approach, the open-source Snap7 (Python) and the PUT/GET mechanism with S7Comm (Siemens S7-1200 PLC, Germany) allow the data to be transmitted and received between the PLC and the IPC through PLC memory areas such as DB, I, Q, and M. In addition, the one-way data transmission frequency from the IPC to the PLC is 100ms. The data transmission process time is requested to be greater than 10 ms to ensure the data packet is not lost. In the current work, the processing time total is approximately 100 ms. Therefore, the data sent from IPC to PLC is ensured.

In general, the experiments demonstrated that (1) the 4-DOF SCARA robot arm was successfully recovered (stable operation, accurately picking up and placing the products), (2) the image processing algorithm for classifying the destinations based on the QR codes was effective, (3) the conveyor speed should be suitably chosen (8 cm/s).

Recently, the robot-based sorter systems have strongly developed. They have been applied in varieties of fields such as warehouses [2], [5]; logistics [1], [2], [5], [6]; palletizing [10], [11]; and agriculture [18], [19]. To increase the accuracy of the classifying and sorting stages, the robot arm could combine with image processing/ computer vision [18], machine learning/deep learning [7]. Our current work, the recovered robot arm, was utilized for classifying and sorting the product boxes using the QR code-based image algorithm with an accuracy of 93.33%. Our research result is entirely appropriate with the previous works [9], [10], [18]. Most studies used readily available equipment such as controllers (e.g., Arduino and PLC), industrial robot arms, and conveyors to identify and classify products. After the product identification stage, the conveyor is controlled to stop so the robot arm picks up the product and places it in the desired location [8]. For our current study, the steps of product box identification/ classification based on image processing, the robot arm picks up the classified product, and places it in the known destination area while the continuously running conveyor. In particular, the robot arm has been successfully recovered, including the components of hardware, PLC firmware, and user graphic interface software.

4. Conclusions

The paper presented a solution for recovering the 4-DOF SCARA robot arm and applying it to classify and sort the product boxes using the QR code-based image processing algorithm. The proposed approach was applied to classify and sort the product boxes assigned with 3 different QR codes. The experimental results revealed that the 4-DOF SCARA robot arm was successfully recovered. The average accuracy of the X and Y coordinates of the robot arm was 0.24 mm and 0.12 mm, respectively. Additionally, the robot arm was combined with the QR code-based image processing algorithm to classify and sort the product boxes. The experimental results demonstrated that the average accuracy of classifying and sorting product boxes was 93.33%. And the average productivity for both classifying and sorting was 16 s/product**.** The results are promising for classifying and sorting applications in the logistics field.

Acknowledgment: We would like to express our sincere appreciation to Mr. To Truong An Le, Mr. Vinh-An Lu, and Mr. Thanh-Dieu Son for their efforts in deploying all experiments in this study. We are especially grateful to Mr. Phu-Chau Huynh for his valuable donation of the mechanical components of the SCARA robot arm. Especially, this study is part of a scientific research project (T2024-68) approved by Can Tho University.

REFERENCES

[1] P. Q. Hai, P. Q. Phat, and D. H. Quan, "Digital transformation in Vietnamese logistics enterprises," *VNU Journal of Economics and Business,* vol. 3, no. 1, Feb. 2023, doi:10.57110/jebvn.v3i1.160.

[2] B. Gutelius and N. Theodore, "The Future of Warehouse Work: Technological Change in the U.S. Logistics Industry," *UC Berkeley Labor Center & Working Partnerships USA,* Oct. 2019. doi: 10.13140/RG.2.2.16952.29444.

[3] M. Obosu and S. Frimpong, "Advances in automation and robotics: The state of the emerging future mining industry," *Journal of Safety and Sustainability*, vol. 2, no. 3, pp. 181–194, Sep. 2025, doi:

10.1016/j.jsasus.2025.05.003.

[4] A. (Nick) Dedeke and K. T. Hung, "A simulation study of automating parcel sorting in postal industry," *International Journal of* Industrial *and Systems Engineering*, vol. 3, no. 5, p. 515, 2008, doi: 10.1504/IJISE.2008.018230.

[5] A. V. de Oliveira, C. M. O. Pimentel, R. Godina, J. C. de O. Matias, and S. M. P. Garrido, "Improvement of the Logistics Flows in the Receiving Process of a Warehouse," *Logistics*, vol. 6, no. 1, p. 22, Mar. 2022, doi: 10.3390/logistics6010022.

[6] A. Al Zadajali and A. Ullah, "The Effectiveness of Logistics Services on Firms' Performances – A Literature Review," *American Journal of Economics and Business Innovation*, vol. 3, no. 1, pp. 125–132, Apr. 2024, doi: 10.54536/ajebi.v3i1.1744.

[7] S. Rath, M. Pandey, and S. S. Rautaray, "Synergistic review of automation impact of big data, AI, and ML in current data transformative era," *F1000Res*, vol. 14, p. 253, May 2025, doi: 10.12688/f1000research.161477.2.

[8] Ei Ei Wai and Phyu Zin Oo, "Implementation Of Barcode-Based Package Sortation System With Picked & Placed Robotic Arm," *International journal for innovative research in multidisciplinary field*, vol. 5, no. 7, pp. 82–91, Jul. 2019.

[9] R. U. Haq, U. Kumar, N. Kumar, Y. Barnanwal, M. Mursaleen, and N. A. Sheikh, "Design and Manufacturing of Low Cost SCARA Robot," *International Journal of Engineering Research &* Technology *(IJERT),* vol. 10, no. 6, Oct. 2021.

[10] S. F. Noshahi, A. Farooq, M. Irfan, T. Ansar, and N. Chumuang, "Design and Fabrication of an Affordable SCARA 4-DOF Robotic Manipulator for Pick and Place Objects," in *2019* 14th *International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP)*, IEEE, Oct. 2019, pp. 1–5. doi: 10.1109/iSAI-NLP48611.2019.9045203.

[11] P. P. V and J. Suresh, "Integration of SCARA Robot for Pick and Place Application using PLC," in *Proceedings of the First International* Conference *on Combinatorial and Optimization, ICCAP 2021, December 7-8 2021, Chennai, India*, EAI, 2021. doi: 10.4108/eai.7-12-2021.2314569.

[12] E. John S. Tiongson, "PLC-Controlled Coordination of Multi-Arm Robotics Without Cooperative Devices for Task Execution," *Journal of Advanced Mechanical Engineering Applications*, vol. 6, no. 1, Jun. 2025, doi: 10.30880/jamea.2025.06.01.004.

[13] M. Ahamed and H. Gu, "Package sorting control system based on barcode detection," in *2022 7th International Conference on Automation, Control and* Robotics *Engineering (CACRE)*, IEEE, Jul. 2022, pp. 148–152. doi: 10.1109/CACRE54574.2022.9834212.

[14] S. Kucuk and Z. Bingul, "The inverse kinematics solutions of industrial robot manipulators," in *Proceedings of the IEEE International Conference on Mechatronics,* 2004, pp. 274–279. doi: 10.1109/ICMECH.2004.1364451.

[15] H. Huang, A. Arogbonlo, S. Yu, L. C. Kwek, and C. P. Lim, "Trajectory tracking of a SCARA robot using intelligent active force control," *Neural Comput Appl*, vol. 37, no. 19, pp. 13345–13370, Jul. 2025, doi: 10.1007/s00521-025-11200-x.

[16] CC-smart Lab, "MSD_E10 DC Servo Driver Medium Current," *Cc-smart.com,* 2022. [Online]. Available: https://cc-smart.net/vi/san-pham/msde10.html [Accessed Jan 26, 2026].

[17] G. Molenaar, "Introduction Snap7 library," *Snap7 Documentation,* 2023. [Online]. Available: https://python-snap7.readthedocs.io/en/1.3/API/client.html [Accessed Jan 26, 2026].

[18] L. Wang, R. Wang, H. Liu, L. Guo, W. Xu, H. Wang, H. Zhao, J. Yu, Q. Gao, and Z. Su, "Design and research of intelligent strawberry sorting robot based on machine vision technology," *Smart Agricultural Technology*, vol. 13, p. 101695, 2026, doi: 10.1016/j.atech.2025.101695

[19] Y. Lyu, A. Ping, Y. Zhang, F. Wang, C. Liu, H. Ruan, H. Jiang, and M. Zhou, "Design and evaluation of robotic seed potato sorting system based on monocular two-stage ROI-guided detection framework," *Smart Agricultural Technology*, vol. 12, p. 101602, 2025, doi: 10.1016/j.atech.2025.101602.