Vietnam Journal of Agricultural Sciences

p-ISSN 2588-1299
e-ISSN 3030-4520

VJAS 2026; 9(3): 3110-3123
https://doi.org/10.31817/vjas.2026.9.3.06

Research on the Control and Implementation of a Machine Vision-Based Cartesian Robot for Automated Sorting Systems

Nguyen Thi Hien*, Nguyen Van Loi, Nguyen Duy Phu & Nguyen Van Tien

Faculty of Engineering, Vietnam National University of Agriculture, Hanoi 12400, Vietnam

Abstract

In the context of the rapid development of Industry 4.0, the integration of industrial robots with machine vision technology is becoming an inevitable trend to enhance the level of automation in manufacturing. This paper presents the results of the control and implementation of a three-axis Cartesian robot integrated with a machine vision system for object sorting on a conveyor belt. The system was designed with a working area of 400 × 700 × 150mm and was capable of recognizing and sorting phone chargers based on surface print quality through image processing algorithms. Object tracking was employed to enable the robot to perform picking operations under a continuously moving conveyor belt. The experimental results showed that the system achieved a defect rejection rate of 96-98% under the tested conditions, demonstrating stable operation and its potential for practical applications. The research makes the contribution of a proposed integrated solution suitable for small and medium-scale sorting applications.

Keywords

Cartesian robot, vision-based robotic system, machine vision, sorting system

Introduction

In recent years, driven by the rapid advancement of Industry 4.0, automation technologies, industrial robotics, and artificial intelligence have been increasingly adopted across various manufacturing sectors. Modern production systems not only demand high productivity but also require accuracy, reliability, and continuous operation over extended periods (Groover, 2020). Therefore, the development of intelligent automated systems has become essential to replace or assist human operators in repetitive and precision-critical tasks.

Among these tasks, product sorting plays a vital role in ensuring product quality and optimizing downstream processes such as

Received: March 23, 2026
Accepted: September 9, 2026

Correspondence to
Nguyen Thi Hien
nthien@vnua.edu.vn

ORCID
Nguyen Thi Hien
orcid.org/0000-0002-8170-4450

This work is licensed under the Creative Commons Attribution 4.0 License.

packaging, storage, and distribution. However, many existing sorting systems still rely on manual labor or simple sensor-based mechanisms, which are often limited in accuracy and lack the ability to handle complex inspection criteria. To overcome these limitations, the integration of industrial robots with machine vision systems has emerged as an effective and widely adopted solution (Corke, 2017; Craig, 2018; Khan *et al.*, 2019; Mohamed *et al*., 2019; Vo *et al*., 2022; Elassal *et al*., 2024; Nguyen Thi Hien *et al*., 2025).

Various types of industrial robots, such as SCARA, parallel, and Cartesian robots, are widely used for automation tasks depending on the application requirements. Among them, Cartesian robots are particularly suitable for pick-and-place and sorting applications due to their simple mechanical structure, ease of control, and high positioning accuracy. Their orthogonal motion structure simplifies kinematic modeling and enhances repeatability, making them well-suited for integration with vision-based systems (Craig, 2018; Mohamed *et al*., 2019; Groover, 2020; Liu & Zhu, 2024).

Additionally, machine vision has become a key enabling technology in robotic systems, providing real-time information about object features, positions, and conditions. By extracting visual features such as geometry, texture, and spatial coordinates, vision systems enable robots to perform inspection, localization, and tracking tasks with high precision (Corke, 2017; Shu *et al.*, 2022; Szeliski, 2022; Ho *et al*., 2023).

Recent studies have demonstrated the effectiveness of combining robotic manipulators with machine vision in automated sorting and handling applications. Mohamed *et al*. (2019) designed a Cartesian system integrated with machine vision. This work highlighted a modular gantry system using Arduino and vision-based geometric coding, achieving efficient pick-and-place with a time reduction of about 25% compared to manual processes. In other research, authors further advanced this approach by integrating high-performance vision (Hikvision camera, Jetson TX2), achieving over 98% color recognition accuracy and ± 2 mm positioning precision (Liu & Zhu, 2024). Earlier research (Yan & Yang, 2015) established robust image processing methods such as thresholding and centroid localization for reliable object detection. More recently, vision-based robotic sorting has also been combined with deep learning for object classification and localization, demonstrating its potential for real-time automated sorting on conveyor systems (Karadağ & Kiliç, 2023).

However, existing systems still face limitations in coordinating visual inspections, object localization, and robot motion, particularly when defective objects must be removed from a continuously moving conveyor. In addition, relatively few practical systems have addressed surface-quality verification based on printed characters while simultaneously compensating the detected position and tracking the defective object for real-time rejection. To address these limitations, this study developed a three-axis Cartesian robot integrated with a machine vision system for automated inspection and sorting of phone chargers based on the clarity of printed characters. The technical contribution of the proposed system is the integration of character verification, camera-to-robot coordinate compensation, and conveyor-synchronized tracking within a unified PLC-based control framework. This integration will enable the robot to determine the grasp position of a defective object and intercept it while the conveyor remains in motion, rather than relying only on static pick-and-place operations. Accordingly, the objectives of this study were to implement an integrated vision-robot system, develop coordinate compensation and tracking strategies, and experimentally evaluate the system's sorting performance under continuous conveyor operation.

Materials and Methods

Based on a comprehensive analysis of existing sorting systems, the proposed architecture integrated mechanical design, control, and machine vision into a unified framework. The system consisted of a conveyor belt for object transportation, a sorting module implemented using a three-axis Cartesian robot, and a machine vision subsystem. The vision

subsystem included a camera rigidly mounted above the initial section of the conveyor belt, a uniform LED illumination setup to minimize shadows and reflections, and a PC for image acquisition and processing.

During operation, objects were conveyed to the inspection position, where a sensor triggered the camera to capture images. At this stage, the conveyor belt was temporarily stopped to ensure stable imaging conditions, thereby reducing motion blur and improving image quality. The captured images were then processed using machine vision algorithms to extract relevant features and detect defects. Based on the analysis results, the objects were classified as passing or failing. Passing objects were allowed to proceed to the collection tray at the end of the conveyor belt, whereas for failing objects, the coordinates were transmitted to a programmable logic controller (PLC). The three-axis robot employed a tracking algorithm to synchronize its motion with the conveyor belt speed, approach the target object, grasp it, and transfer it to the designated rejection tray. The entire sorting sequence was performed automatically.

The objects considered in this study were phone chargers made of rigid plastic, with dimensions 30 × 20 × 20mm (length × width × height) and an approximate weight of 80 grams. An object was considered acceptable if it exhibited a clear and sharp "USB" marking (i.e., without blurring or missing characters), otherwise, it was classified as defective. **Figure 1** illustrates an example of an acceptable object (left) and a defective one (right).

Three-axis Cartesia robot

Based on the required working space of 400 × 700 × 150mm, a three-axis Cartesian robot was designed and constructed. The dimensions of the X, Y, and Z axes are summarized in **Table 1**.

In this study, three Panasonic servo motors were employed to drive the three axes, ensuring high positioning accuracy and operational stability. The end effector attached to the bottom of the Z-axis was a vacuum suction cup for pick-and-place operations. The suction cup was connected to an air hose via a ZH07BS-06-06 vacuum ejector to create a negative airflow. To control the compressed air flow into the vacuum valve, an AIRTAC 4V110-06 (5/2) solenoid valve was used. The parameters of the servo motors and vacuum ejector are denoted in Table 2 (Misumi, 2025; Panasonic, 2025).

Due to the orthogonal prismatic configuration of the Cartesian robot, the end-effector position was directly determined by the linear displacements of the X, Y, and Z axes; therefore, complex kinematic transformations were not required, while detailed dynamic modeling was beyond the scope of this study.

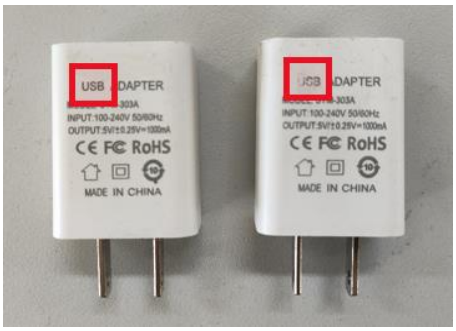


Figure 1. An example of an acceptable object (left) and a defective one (right)

Table 1. Dimensions of the XYZ robot

Axis	Dimension (mm)	Tolerance
X	410	± 2mm
Y	750	± 2mm
Z	200	± 2mm

Table 2. Parameters of the servo motors and vacuum ejector

Items	Manufacturer	Code	Image	Number	Parameters
Servo motor	Panasonic	MSMF012L1T2		3	Power: 100W Power supply: 3P/200 VAC Speed: 3000rpm (maximum speed: 5000rpm) Torque: 0.32 Nm Max. Torque: 0.955 Nm
Vacuum ejector	SMC	ZH07BS-06-06		1	Nozzle diameter: 0.7 mm Vacuum pressure: -88 kPa Air inlet - suction port: Quick-connect type Ø6 mm

The Cartesian robot served as the main actuation unit of the sorting system, while its motion was coordinated with the conveyor and vision-based inspection through the PLC control strategy described in the following sections.

Other main components

In addition to the robotic system, the other key components, including the conveyor belt, PLC, sensors, camera, etc., are summarized in **Table 3** and constituted the main components of the automated sorting system. **Figure 2** illustrates the 3D model of the three-axis robot (a) and the architecture of the sorting system with a machine-vision-based three-axis robot (b).

NI vision builder AI

NI Vision Builder for automated inspection (AI) is a development tool provided by National Instruments (NI) for building artificial intelligence and image processing applications. The software offers a menu-driven workflow that enables users to design image inspection and recognition systems efficiently. By utilizing pre-built tools in the menu, users can configure cameras, customize image analysis processes, and generate inspection results without the need for programming. Therefore, NI Vision Builder AI supports the rapid development and deployment of machine vision inspection systems (NI, 2023).

In this study, NI Vision Builder AI was employed as the primary platform for developing an image processing solution for object recognition, classification, and positional information extraction required for subsequent sorting control.

Results and Discussion

Block diagram of the control system

Figure 3 illustrates the block diagram of the control system for the proposed sorting system. Overall, the system operated through the coordinated interaction of sensing, control, actuation, and image processing modules.

The sensor and encoder unit provided input information to the controller. The sensors detected the presence of objects, while the encoder measured conveyor motion and generated pulse signals for synchronization.

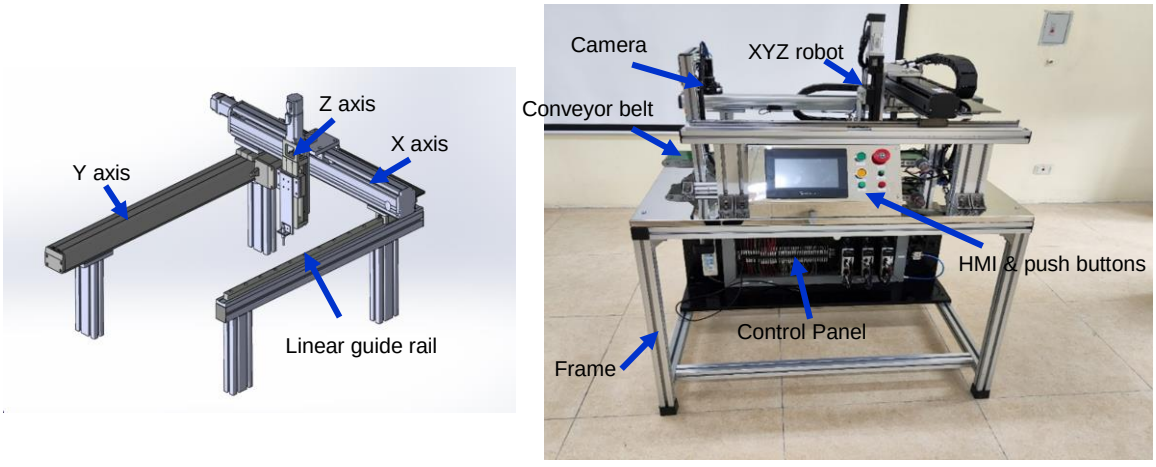
At the core of the system, the controller, implemented as a PLC, received input signals from the sensors, encoders, and control buttons, and processed them to generate appropriate control commands. It governed the operation of the actuators, relays, conveyor belt motors, and camera triggering, and facilitated data exchange with both the monitoring system and the image processing unit.

The actuators comprised the driving motors of the sorting mechanism, including three servo-driven axes (X, Y, and Z) responsible for object manipulation and sorting tasks. The camera was employed to capture images of each object at the inspection stage, which were then transmitted to the image processing computer.

The image processing computer integrated machine vision algorithms to analyze the captured images and determine whether each

Table 3. Other main components of the sorting system

No.	Equipment	Quantity	Image	Characteristics	Function
1	Conveyor Belt	1		Length × width: 800 × 100 (mm)	Transport objects
2	PLC Mitsubishi	1		Inputs/Outputs: 64 Power supply: 100/240 VAC Communication connections: Ethernet, RS485	Control the entire system
3	Camera MV-CE050-30GM	1		MT9P031 sensor Pixel size: 2.2µm × 2.2µm; 14fps Power supply: 12 VDC	Acquire image of the objects
4	CX442 Sensor	2		Sensing distance: 100mm Adjust range: 40-300mm Power supply: 12VDC-24VDC	SS1: detect an object, stop the conveyor belt, and simultaneously activate the camera to take an object image SS2: synchronize the robot velocity with the conveyor belt
5	DC motor JGB37-3530	1		Power supply: 24VDC; 60rpm; 2.2W-14.4W	Motion transmission for the conveyor belt
6	Encoder HN3806-AB-600	1		600 ppr Power supply: 8-24VDC Speed max: 6000rpm	Determine the displacement of defective objects from SS2 when multiple defective objects are detected



a) 3D model of 3-axis robot

b) Architecture of the sorting system with a machine vision based 3-axis robot

Figure 2. 3D model of the 3-axis robot (a) and architecture of the sorting system with the machine-vision-based 3-axis robot (b).

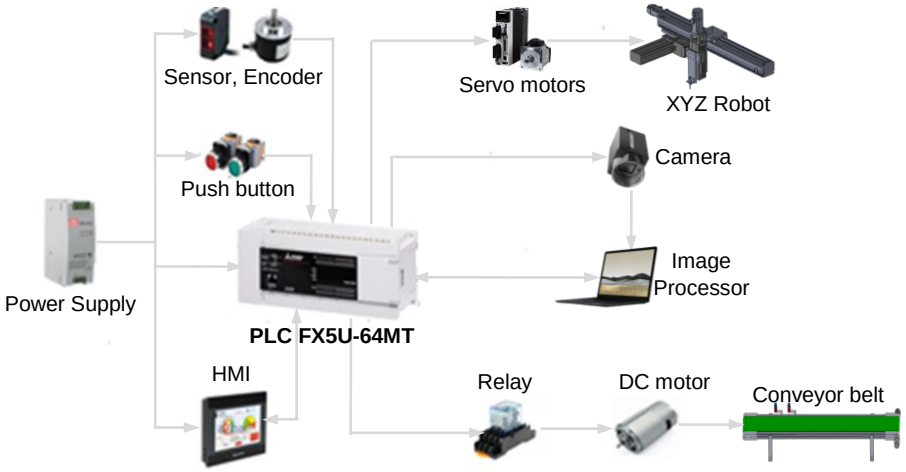


Figure 3. Functional block diagram of the overall control system.

object met the quality requirements, subsequently providing classification results (pass or fail) to the controller. Finally, the monitoring system, implemented as an HMI, was used to supervise the system operation and visualize process parameters, communicating with the controller via an RS485 communication interface.

Imade processing

The image processing procedure implemented in NI Vision Builder AI is shown in **Figure 4**. It consisted of image acquisition and calibration, object inspection, and separate processing branches for defective and acceptable objects. After image acquisition and calibration, the object was evaluated according to the predefined inspection criteria. The comparison result was then used to classify the object as either defective or acceptable, thereby determining the corresponding processing branch.

(i) Image Acquisition and Calibration Block

The image acquisition and calibration procedure consisted of five blocks, as described in **Figure 5**. The image acquisition stage utilized the *Acquire Image* tool with an industrial camera (HIKROBOT MV-CE050), where parameters including resolution, exposure time, and saturation were configured to ensure stable image quality for subsequent processing.

To enable accurate measurement, the *Calibrate Image* tool converted pixel coordinates into real-world units (e.g., 1 pixel = 0.26mm) using reference points with known distances, ensuring precision and repeatability.

Object detection was performed using the *Match Pattern* tool, which identified predefined templates in the image through techniques such as cross-correlation or convolution, with a full rotation search range (0-360°) and a minimum similarity threshold of 80%.

Finally, the *Set Coordinate System* tool established a consistent reference frame for object localization, while the *Set Inspection Status* tool classified products as “Pass” or “Fail”, enabling automated decision-making for sorting and control processes.

Figure 6 illustrates some of the steps in the image acquisition and calibration block.

(ii) Comparison Block

After image acquisition and coordinate assignment, preprocessing steps were applied to normalize and enhance the textual data, followed by comparison operations with a predefined reference object. The reference data were extracted from the image acquisition stage and forwarded to the comparison block (**Figure 7**) for verification.

The *Read/Verify Text* tool was configured by selecting the appropriate parameters, including the language and text recognition settings. Once properly configured, the tool automatically read and verified the text within the defined region of interest (ROI).

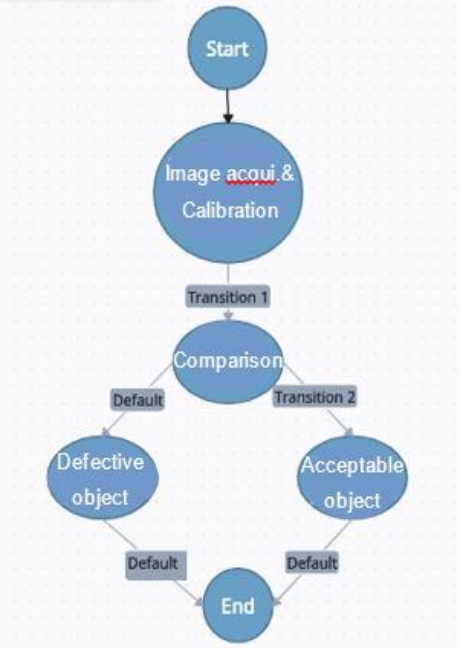
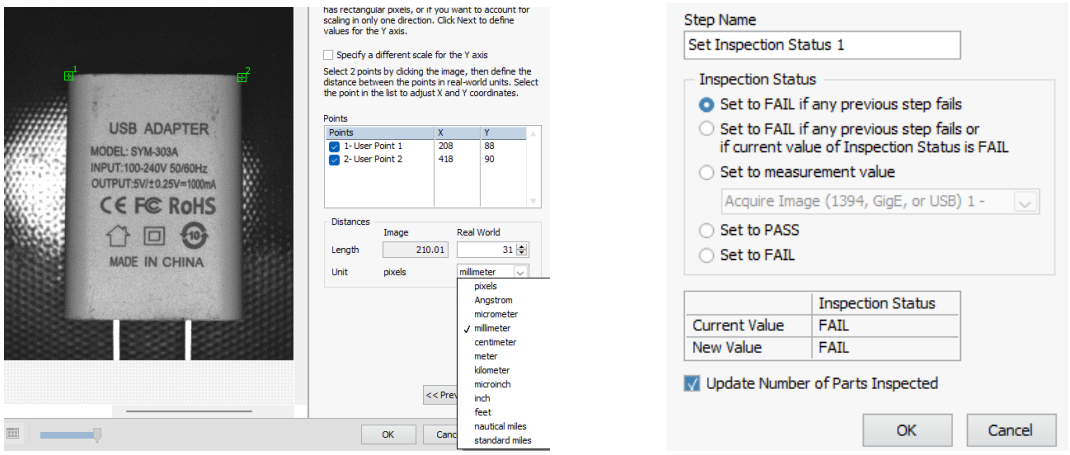


Figure 4. Image processing using NI Vision Builder AI



Figure 5. Image acquisition and calibration block



a) Calibration process b) Set inspection status

Figure 6. Some of the steps in the image acquisition and calibration process

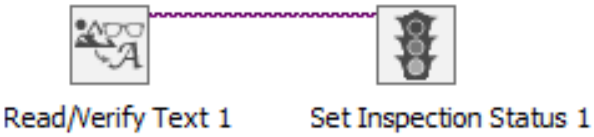


Figure 7. The comparison block

Figure 8 presents the configuration workflow in NI Vision Builder AI for "USB" character verification.

After text verification using the *Read/Verify Text* tool, the *Set Inspection Status* tool classified the product as "Pass" or "Fail" based on the verification result.

(iii) Defective Object Block

When an object failed the comparison stage, it was automatically redirected to the defective object block (**Figure 9**) for further handling. The inspection tool was configured and activated to ensure this automated classification process. To enhance visualization, the *Custom Overlays* tool was added and configured to display inspection results (e.g., "Pass" or "Fail") directly on the image.

For defective product processing, the *Calibrate Image* and *Match Pattern* tools were duplicated and integrated into the defective object block, enabling consistent localization and detection of faulty items. Subsequently, the *TCP I/O* tool was incorporated to facilitate communication with external systems. By configuring the port address and transmitting positional data obtained from calibration and pattern matching, the system was able to share defect location information with other devices, such as the controllers or robotic units, thereby supporting coordinated and automated sorting operations.

(iv) Acceptable Object Block

After an object passed the comparison stage, it was automatically transferred to the acceptable block for subsequent processing. Once configured, the inspection tool was applied to enable this automatic classification mechanism.

The *Custom Overlays* tool was then added and configured to display the inspection result (e.g., "Pass" or "Fail") directly on the image, thereby enhancing visualization and system monitoring.

Coordinate compensation

The calibrated image coordinates were subsequently transformed into the robot reference frame to determine the grasp position of defective objects.

Objects were transported on the conveyor belt at random positions. Therefore, to determine the pickup point for defective objects on the conveyor belt, it was necessary to convert the camera coordinate system to the space coordinate frame.

Three coordinate frames O_s, O_r and O_c, were defined as the space, robot, and camera coordinate frames, respectively. The space coordinate frame was referenced to sensor SS1 (**Figure 10**). A spatial point P(X, Y, Z) can be represented in the camera coordinate system as (x, y, z) or in the image plane as (u, v, 1) in pixel units. According to Yan and Yang (2015), Vo *et al*. (2022), and Nguyen Thi Hien *et al*. (2025), the camera coordinates are mapped to the space coordinate as:

$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \textbf{(1)}$$

where R is a rotation matrix and d is the translation vector. The homogeneous transformation $T = \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix}$ represents the extrinsic parameters used to transform points between the space and the camera coordinate frames.

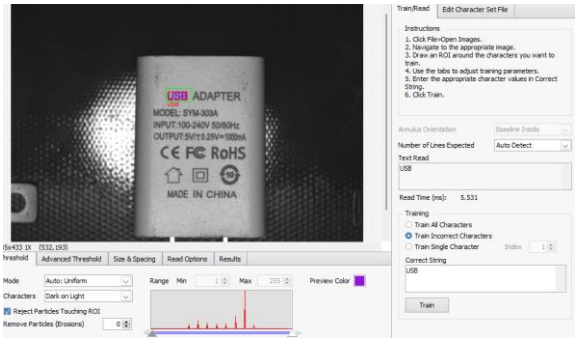


Figure 8. Setting up a character verification process on the object.

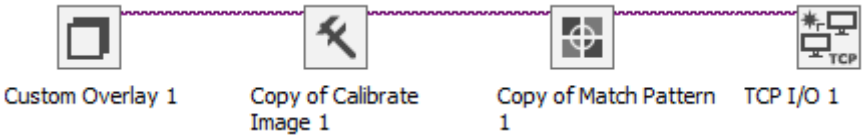


Figure 9. Defective object block.

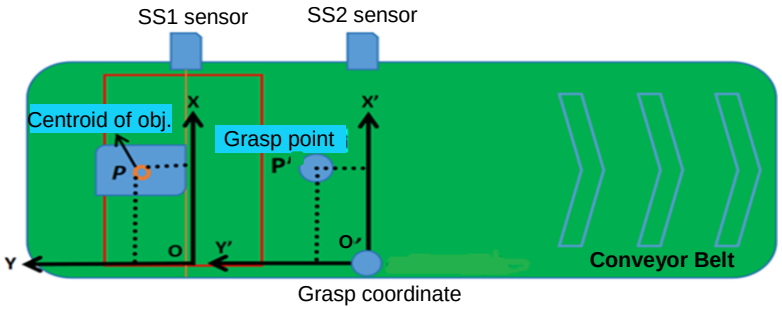


Figure 10. Illustration of coordinate compensation between the two coordinate systems.

$$R = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix}; \qquad d = \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} \quad \textbf{(2)}$$

In the case where distortion was not considered, the image plane of the camera coordinate system is described in Equation (3) by using the intrinsic matrix K and a scale factor α:

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad \textbf{(3)}$$

$$K = \begin{pmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{pmatrix} \quad \textbf{(4)}$$

where (u_0, v_0) is the centroid of the object in the image plane. We can rewrite Equations (1) – (3) as equation (5):

$$\alpha \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = K \begin{pmatrix} R & d \\ 0 & 1 \end{pmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad \textbf{(5)}$$

The Caliper tool in NI Vision Builder AI was used to determine the intrinsic and extrinsic parameters, allowing the position P(X, Y, Z) to be determined.

However, the spatial point P(X, Y, Z) shifted with the conveyor motion. Therefore, in the grasp coordinate O' (with origin (o_x, o_y, o_z)) determined by sensor SS2 (or a standby point), the grasp point P'(X', Y', Z') can be obtained as:

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix} \quad \textbf{(6)}$$

System control programming

The system control was implemented through two main processes: image processing and object rejection. The PLC coordinated the three servo-driven axes based on the compensated target coordinates from the vision system, while the sensor and encoder signals were used to synchronize the robot motion with the conveyor during object tracking and rejection.

In the image processing stage, the PLC read the signal from sensor SS1 to determine when an object reached the inspection position. The conveyor belt was temporarily stopped to ensure optimal image acquisition conditions. The captured image was then processed, and classification results were generated (**Figure 11a**). A defective object was localized and its coordinates were transmitted to the PLC for removal, whereas an acceptable object was allowed to proceed to the end of the conveyor.

In the object rejection stage, the removal process began upon receiving a signal indicating a defective object. The conveyor remained in motion, and the robot tracked the target object to perform the removal operation. The corresponding control procedure is illustrated in **Figure 11b**.

For a single defective object, the robot, which was initially positioned at the standby location (sensor SS2), moved at a velocity equal to the conveyor belt speed to grasp the object. In the case of multiple defective objects (e.g., two defective objects), after handling the first object, the robot may not have time to return to the standby position. In this situation, the system calculated the object's displacement from sensor SS2 and determined an appropriate standby point (interception point) such that the robot's travel time to this point was shorter than that of the object ($T_R < T_O$). Then, from this interception point, the robot synchronized its motion with the conveyor belt, moving at the same speed to perform the sorting operation.

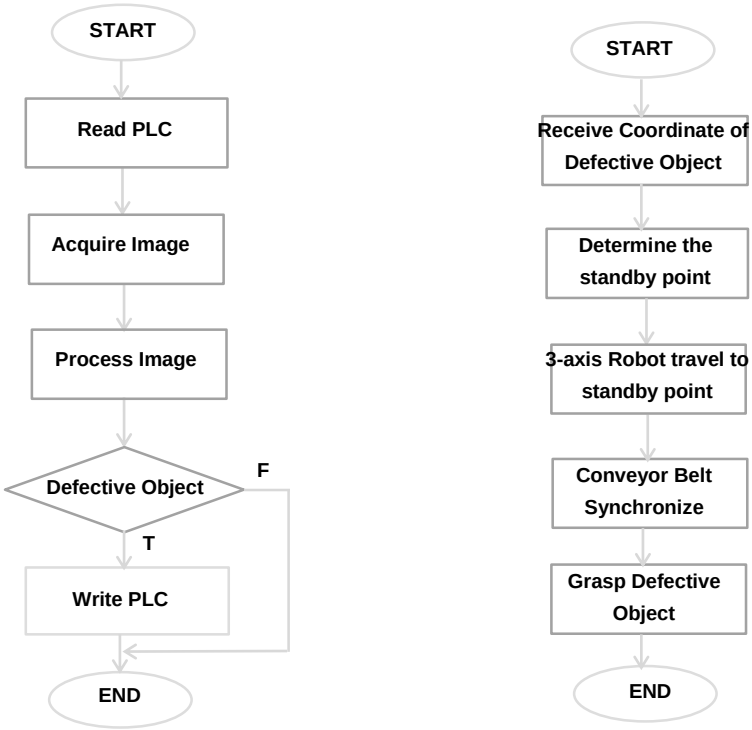
Experimental Results

(i) Validation Test Based on Simulation

GX Works3 software in GX Simulator 3 mode was used to evaluate the PLC control program against the established operational requirements and verify that the control sequence complied with the designed algorithm. The test simulated different operating conditions while monitoring the control-bit states to identify potential deviations (**Figure 12**).

(ii) Validation Test PLC – Servo Communication Using Hercules Software

Communication between the PLC and servo system was tested using Hercules software to verify connection establishment, data transmission/reception, and PLC data processing. The test also verified the compensated coordinate data and the execution of the corresponding control sequences for single and multiple defective-object cases (**Figure 13**).



a) Flowchart of image processing control b) Flowchart of object rejection control

Figure 11. Flowchart of the sorting system control with a machine-vision-based Cartesian robot.

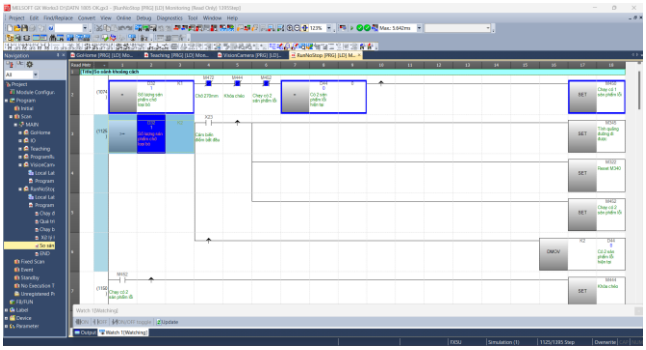


Figure 12. Simulating program testing using GX Works3.

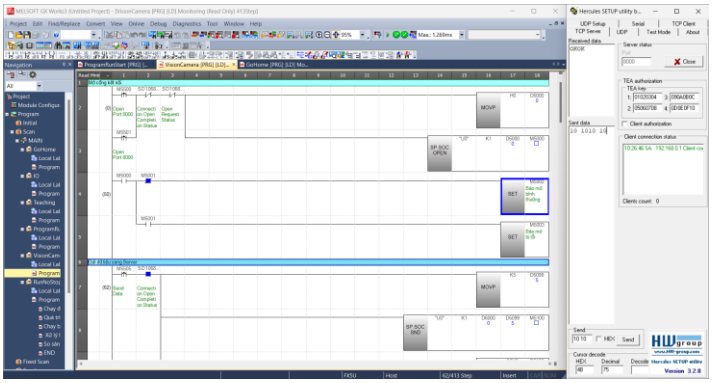


Figure 13. PLC-servo communication test using Hercules software

(iii) Validation Test PLC – HMI Communication

Communication between the HMI and PLC was tested by verifying the exchange of register values and signal-bit states. The system response to control commands from the HMI was also evaluated to verify synchronized operation between the HMI and PLC. **Figure 14** shows the I/O monitoring interface on the HMI screen during the PLC – HMI communication test.

(iv) Operation with the Real Sorting System

After system startup, the operational readiness conditions, including interlocking conditions and equipment status, were verified, and all mechanisms were confirmed to be in their correct initial positions. The control program was then tested, including point-to-point motion, trajectory tracking, and distance verification according to the designed process.

After completing the initial checks, objects were placed on the conveyor belt to evaluate the system under the tested operating conditions. During this process, the performance of the vision system (image acquisition), and the data transmission/reception and processing capability for defective object detection were assessed. At the same time, the robot's tracking, central suction, and removal of defective objects were observed both physically and through the HMI monitoring screen.

In the experiment, 100 products (including 50 defective samples) were fed sequentially onto the conveyor belt with a spacing of 15cm between consecutive items, and the conveyor belt speed was set at 50 mm s^{-1}. The experiment was repeated five times to evaluate the stability and reliability of the system. Three performance indicators were considered: successful image acquisition, central suction, and successful rejection of defective objects. Central suction was defined as suction occurring within the predefined central region of the object; off-center suction was not counted as central suction, although it could still provide sufficient holding force for successful object rejection. The image acquisition rate was calculated based on all 100 tested objects in each run, whereas the central suction and defect rejection rates were

calculated based on the 50 defective objects requiring robotic handling. Each rate was determined as the number of successful operations divided by the corresponding number of tested objects. The system operation results are presented in **Table 4**. **Figure 15** shows some images of the system's operation.

Remarks

The experimental results in **Table 4** indicate relatively stable system performance across the five experimental runs. The image acquisition rate ranged from 98% to 100%, indicating reliable visual input under the tested conditions. The relatively high performance can be attributed to the controlled image acquisition conditions, coordinate calibration, and conveyor-synchronized tracking strategy, which together reduced image uncertainty and improved the positioning of defective objects during robotic handling. The central suction rate ranged from 94% to 98%, while the defect rejection rate

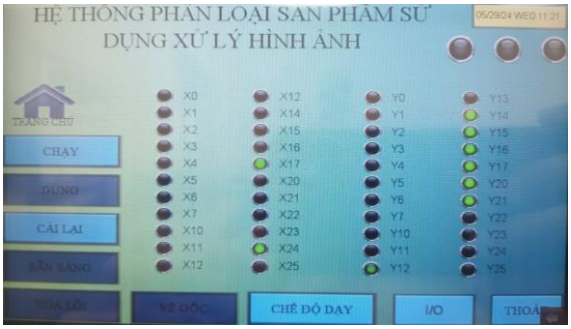
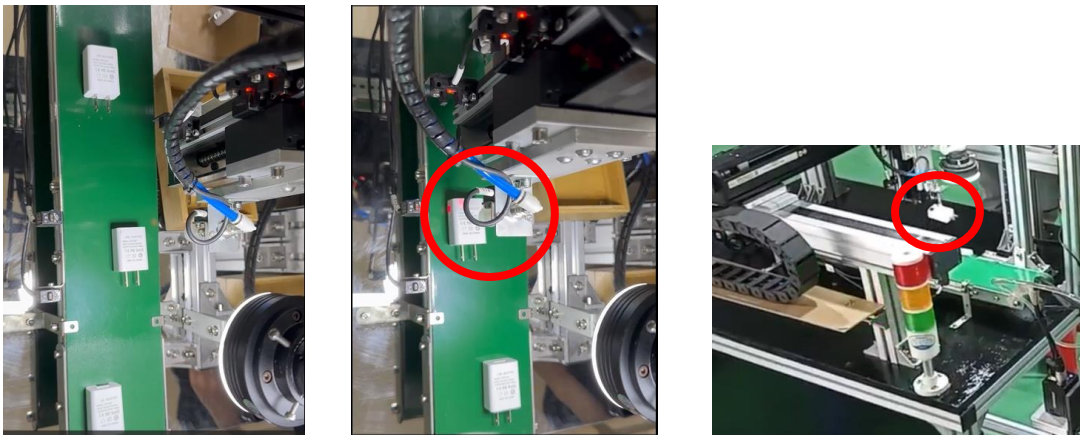


Figure 14. I/O monitoring interface on HMI screen

Table 4. Experimental results for the sorting system

No.	Acquired image	Central suction	Rejected defective objects	Display on HMI screen
1	98 (98%)	47 (94%)	48 (96%)	All
2	100 (100%)	49 (98%)	49 (98%)	All
3	100 (100%)	47 (94%)	49 (98%)	All
4	99 (99%)	48 (96%)	48 (96%)	All
5	100 (100%)	48 (96%)	48 (96%)	All

Note: Central suction indicates suction within the predefined central region of the object. Off-center suction may still result in successful grasping and rejection.



a) Objects transported on the conveyor belt

b) Robot tracking the defective object

c) Robot grasping and removing the defective object

Figure 15. Representative images of the system in operation.

remained between 96% and 98%. The slightly higher rejection rate in some runs indicates that off-center suction could still provide sufficient holding force for successful object removal. The small variations among the experimental runs can mainly be attributed to differences in object positioning and to mechanical tracking and suction deviations during robotic handling. The results therefore suggest that the remaining errors were mainly associated with the robotic handling and tracking stages rather than with image acquisition. All operational data were displayed on the HMI, enabling real-time monitoring of the sorting process.

Direct quantitative comparison with previous studies is difficult because the reported systems differ in object types, inspection criteria, system configurations, and experimental conditions. Compared with systems mainly focusing on object recognition and conventional pick-and-place operations, the present system combined character-based surface-quality inspection with coordinate compensation and conveyor-synchronized tracking, allowing defective objects to be rejected while the conveyor was moving. Previous vision-based robotic sorting systems have demonstrated the effectiveness of integrating machine vision with robotic manipulation (Mohamed *et al*., 2019; Vo *et al.*, 2022; Liu & Zhu, 2024). In the present study, the integration of character-based quality inspection with conveyor tracking and automated rejection achieved a defect rejection rate of 96-98% under the tested conditions. These results demonstrate the feasibility of the proposed system for automated sorting applications, particularly for small- and medium-scale production environments. A standardized quantitative comparison with alternative vision and control methods should be considered in future work.

However, limitations were observed when multiple defective objects appeared in close proximity. Under such conditions, the reduced time available for robot repositioning and interception may affect tracking and grasping performance, indicating the need for further optimization of control algorithms and scheduling strategies. Furthermore, the current image processing system was primarily based on traditional methods such as pattern matching and character checking, which limits its adaptability to more complex object variations. The integration of deep learning algorithms in future work could significantly enhance recognition capabilities and broaden the system's application scope.

Overall, the system achieved a throughput of approximately 15 objects per minute under the tested operating conditions, further demonstrating the practical potential of the proposed vision-based Cartesian sorting system.

Conclusions

This research successfully developed and validated a three-axis Cartesian robot system integrated with machine vision for automated sorting applications. The main contribution of this study is the integration of character-based quality inspection, coordinate compensation, and conveyor-synchronized tracking within a PLC-based control framework, enabling defective objects to be detected and removed while the conveyor remains in motion. The system operated stably, with a defect rejection rate of 96-98% under the tested conditions, demonstrating its potential for real-world applications and providing a foundation for further research to optimize performance and expand functionality. Despite the promising results, the current system relied on conventional image processing techniques, which may limit its adaptability to more complex scenarios. In addition, tracking and grasping performance may be affected when multiple defective objects occur in close proximity. Future research should focus on incorporating advanced vision algorithms, including deep learning-based methods, and optimizing tracking and interception strategies for closely spaced defective objects to improve system robustness and scalability.

References

Elassal A., Mahmoud A., Mohmad O. & Hager E. (2024). Low-cost parallel delta robot for a pick-and-place application with the support of the vision system. Advances in Electrical Engineering, Electronics and Energy. 8(2024): 1-16. DOI: 10.1016/j.prime.2024.100518.

Corke P. (2017). Robotics, Vision and Control (2nd ed.). Springer, Germany.

Craig J. J. (2018). Introduction to Robotics: Mechanics and Control (4th ed.). Pearson, New Jersey.

Groover M. P. (2020). Automation, Production Systems, and Computer-Integrated Manufacturing (5th ed.). Pearson, New Jersey.

Ho V. N., Vo D. C. & Phan X. T. (2023). Development of a SCARA robot arm for palletizing applications based on computer vision. FME Transactions. 51(4): 541-549.

Karadağ A. E. & Kiliç A. (2023). Non-destructive robotic sorting of cracked pistachio using deep learning. Postharvest Biology and Technology. 198: 112229. DOI: 10.1016/j.postharvbio.2022.112229.

Khan A., Li J. P., Malik A. & Yusuf Khan M. (2019). Vision-based inceptive integration for robotic control. Soft Computing and Signal Processing. Advances in Intelligent Systems and Computing. 898. Proceedings of ICSCSP 2018, Vol. 2. Springer, Singapore. DOI: 10.1007/978-981-13-3393-4_11.

Liu G. & Zhu B. (2024). Design and implementation of intelligent robot control system integrating computer vision and mechanical engineering. International Journal of Computer Science and Information Technology. 3(1): 219-226.

Misumi (2025). Box-type vacuum ejector SH series (ZH07BS-06-06). Retrieved from https://vn.misumi-ec.com/vona2/detail/221300205791/?HissuCode=ZH07BS-06-06&srsltid=AfmBOor8qiKt-Lxu3mnuf5YSXs3PyvgqKe-_NHZoeucArHgdYk2J0m0O on May 16, 2025.

Mohamed F. U., Baalaji V., Gowtham K. & Balaji R. (2019). Design and fabrication of Cartesian system integrated with machine vision. Journal of Emerging Technologies and Innovation Research (JETIR). 6(3): 686-690.

Nguyen Thi Hien, Nguyen Dac Toan & Pham Trong Hung (2025). Study on control of Delta robot in sorting systems using machine vision. Vietnam Journal of Agricultural Sciences. 23(7): 934-946 (in Vietnamese).

NI (2023). NI Vision Builder for Automated Inspection: Configuration Help. Retrieved from https://documentation.help/NI-Vision-Builder/ on May 20, 2024.

Panasonic (2025). MINAS A6 Family Servo Motor. Retrieved from https://industry.panasonic.com/global/en/products/motor/fa-motor/ac-servo/number/msmf012l1t2 on May 16, 2025.

Shu M., Li C., Xiao Y., Deng D. & Liu J. (2022). Research on items sorting robot based on SSD target detection. The proceedings of ITM Web of Conferences 47 (CCCAR2022). DOI: 10.1051/itmconf/20224701031.

Szeliski R. (2022). Computer Vision: Algorithms and Applications (2nd ed.). Springer, Germany.

Vo D. C., Le D. H., Le H. P. & Dang A. D. (2022). Design and development of robot arm system for classification and sorting using machine vision. FME Transactions. 50: 181-192. DOI: 10.5937/fme2201181C.

Yan J. & Yang H. (2015). Research on workpiece sorting system based on machine vision mechanism. Intelligent Control and Automation. 6: 1-9.