บทความวิจัย

การออกแบบและพัฒนาแขนหุ่นยนต์เพื่อช่วยในกระบวนการผลิตโดยใช้เทคนิคแมชชีนวิชัน

ลาภิศ พลแสน* และ วิทิต ฉัตรรัตนกุลชัย
ภาควิชาวิศวกรรมเครื่องกล คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
สาธิต วรรณวานิชชัย
ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยมหิดล

* ผู้นิพนธ์ประสานงาน โทรศัพท์ 08 7551 0851 อีเมล: lapitch.polsan@electrolux.com DOI: 10.14416/j.kmutnb.2024.10.013
รับเมื่อ 8 มกราคม 2567 แก้ไขเมื่อ 21 เมษายน 2567 ตอบรับเมื่อ 19 มิถุนายน 2567 เผยแพร่ออนไลน์ 7 ตุลาคม 2567

บทคัดย่อ

บทความนี้เป็นการนำเสนอการออกแบบพัฒนาและการสร้างต้นแบบแขนกลคู่ร่วมกับกล้องเพื่อการรับรู้ที่เป็นการมองเห็น โดยมีลักษณะการทำกิจกรรมคล้ายมนุษย์เพื่อใช้การคัดแยกวัตถุ หยิบจับและการตรวจสอบวัตถุที่พิจารณาหรือ กิจกรรมอื่น ๆ ซึ่งใช้มนุษย์ในการตัดสินใจเป็นหลัก โดยทำการศึกษาการใช้กล้องกับแขนกลคู่ในการตรวจจับวัตถุที่มีการวางตำแหน่งและทิศทางที่เป็นอิสระโดยสามารถคัดเลือกวัตถุตามรุ่นหรือลักษณะที่สนใจและการตรวจสอบคุณภาพหรือโดยส่วนประกอบสำคัญสองส่วนคือการสร้างแขนกล แบบอาร์ติคิวลาต 6 แกนอิสระ (Articulated Robot Arm) สองแขนที่สามารถทำงานโดยอัตโนมัติ ด้วยความสอดคล้องกัน โดยการสร้างส่วนประกอบของแขนกลด้วยการพิมพ์แบบ 3 มิติ (3D Print) และการสร้างโปรแกรมเพื่อการรับรู้ (Machine Vision และ Image Processing) ของแขนกลในการตอบสนองในการคัดแยกวัตถุและตำแหน่งทิศทางของวัตถุ (Objective Detection; Convolutional Neural Network หรือ CNN และ Co-ordinates of Contours; Opencv Library) ด้วยการใช้งานผ่านการแสดงผลทางสัญลักษณ์ Graphical User Interface ด้วยภาษาไพธอน (Python) ใช้งานผ่านบอร์ด ไมโครโปรเซสเซอร์ราสเบอร์รี่พาย 4 (Raspberry Pi 4 Microprocessor Board) และบอร์ดไมโครคอนโทรล เลอร์อาร์ดูโน่ (Arduino Microcontroller Board) ที่ผ่านการสื่อสารแบบ I2C เพื่อควบคุมชุดขับด้วยการควบคุมหมุนองศาและทิศทางของสเต็ปเปอร์มอเตอร์ (Stepper Motor) ของข้อต่อแต่ละข้อของแขนกลทั้งสอง จากผลการวิจัยพบว่า แนะนำให้ใช้แขนกลทั้งสองในการจับยึดชิ้นงานที่มีพิกัดความคลาดเคลื่อน สำหรับตำแหน่งแกน x, y และ z แนะนำให้ใช้ค่าเบี่ยงเบน 3±1.22 เซนติเมตร โดยมีช่วงค่าเฉลี่ยกลางอยู่ที่ –1 ถึง 1.125 เซนติเมตร สำหรับมุมการหมุนแนะนำค่าชดเชย –7±11.5 องศา โดยมีช่วงค่าเฉลี่ยกลางอยู่ที่ –3.65 ถึง 0 องศาระหว่างการจับยึด

คำสำคัญ: แขนหุ่นยนต์ วิชันซิสเต็ม การพิมพ์ 3 มิติ

การอ้างอิงบทความ: ลาภิศ พลแสน, วิทิต ฉัตรรัตนกุลชัย และ สาธิต วรรณวานิชชัย, "การออกแบบและพัฒนาแขนหุ่นยนต์เพื่อช่วยในกระบวนการผลิตโดยใช้เทคนิคแมชชีนวิชัน," *วารสารวิชาการพระจอมเกล้าพระนครเหนือ*, ปีที่ 35, ฉบับที่ 3, หน้า 1–20, เลขที่บทความ 253-077498, ก.ค.–ก.ย. 2568.

Research Article

The Design and Development of Robotic Arms to Assist in Manufacturing Processes Using Machine Vision Techniques

Lapitch Polsan* and Withit Chatlatanagulchai
Department of Mechanical Engineering Faculty of Engineering Kasetsart University, Bangkok, Thailand
Sathit Wanwanitchai
Department of Electrical Engineering Faculty of Engineering Mahidol University, Bangkok, Thailand

* Corresponding Author, Tel. 08 7551 0851, E–mail: lapitch.polsan@electrolux.com
DOI: 10.14416/j.kmutnb.2024.10.013
Received 8 January 2024; Revised 21 April 2024; Accepted 19 June 2024; Published online: 7 October 2024

Abstract

This article presents the design, development, and prototyping of a dual-arm robot equipped with a camera for visual perception. The dual-arm robot is designed to replicate human-like activities, including object sorting, picking up items, examining objects for decision-making, and performing other tasks typically reliant on human decision-making processes. The study employs the camera-equipped dual-arm robot to autonomously detect objects with varying positions and orientations. The system can select objects based on predefined models or specific characteristics and examine object quality. This research includes two main components: the creation of two independent 6-axis articulated robot arms using 3D printing and the development of a perception program in Python. This program utilizes machine vision techniques and image processing with Convolutional Neural Network (CNN) and the OpenCV library for the robot's response in sorting tasks, based on the position and direction of the object. These techniques determine the coordinates of contours, which are crucial for object detection. Control is facilitated through a Graphical User Interface (GUI) on the Raspberry Pi 4 microprocessor board and Arduino microcontroller board via I2C communication, managing the stepper motors' rotation degree and direction for each joint of both robotic arms. Based on the research findings, it is recommended to use both robotic arms for gripping workpieces with tolerance coordinates. For the x, y, and z positions, a deviation of 3±1.22 centimeters is recommended, with a mean range of –1 to 1.125 centimeters for the x, y, and z axes. For orientation angles, a compensation value of –7±11.5 degrees is suggested, with a mean range of –3.65 to 0 degrees during gripping.

Keywords: Robot Arm, Machine Vision, 3D Printing

Please cite this article as: L. Polsan, W. Chatlatanagulchai, and S. Wanwanitchai, "The design and development of robotic arms to assist in manufacturing processes using machine vision techniques.," *The Journal of KMUTNB*, vol. 35, no. 3, pp. 1–20, ID. 253-077498, Jul.–Sep. 2025 (in Thai).

1. Introduction

This research focuses on the integration of robotic arms to perform tasks traditionally carried out by humans in an electrical appliance assembly factory, with a specific emphasis on plastic part production. Injection molding machines generate plastic components, which are subsequently transported to the end of the production line for inspection and necessary corrective measures using a conveyor belt. The study centers on three types of workpieces, each weighing between 0.85 to 1 kg. Two prototype robotic arms have been designed, featuring the ability to move independently along six degrees of freedom. These robotic arms are controlled by a Raspberry Pi board, overseeing object detection and conducting image processing to determine the position and orientation of the workpiece independently during inspection. The recognition model incorporates machine vision techniques for picking up and inspecting plastic pieces through a program designed to automate and control the work. It is imperative that the two robotic arms remain cost-effective, user-friendly, and easily maintainable, with readily available replacement equipment within the country.

The research team developed a 6-degree-of-freedom Articulated Robot, drawing inspiration from existing work. While the design aligns with the robotic arm format in [1], challenges persist, particularly within the rotation of ball bearings produced by 3D printing. [2] utilized servo actuators with low torque, prompting the research team to opt for stepper motor actuators coupled with gear sets and micro-stepper drivers to drive the rotation of each joint, offering a more cost-effective solution. The research [3] integrated the NX Program, employing Module Motion Simulation to simulate movement for numerical analysis of angular acceleration positions during mechanism operation. This approach is informed by [4], which applied a similar methodology to analyze the movement position of vehicle chassis components at various speeds. This research utilized the module for design and analysis to meet the research requirements in studying the movement of the robotic arms. It involved determining the position, reversal, and rotation in degrees for each point of rotation in each inverse kinematic joint. The research incorporates calculations proposed by [5] and [6] to determine the position of the workpiece and navigate back to each rotation point in the robotic arm, while [7] and colleagues tested the module's built-in Solver Engine in their research. In the production of robotic arm parts, the research embraces the concept of additive manufacturing for its ease and cost-effectiveness, aligning with the study on factors affecting this process by [8]. ABS material is chosen for its suitability, featuring a design that compensates for the shrinkage of plastic affecting the assembly of parts. Upon assembling the individual parts and testing the movement of each joint, positional defects persisted. To address these defects, a rotary encoder device was installed to ascertain the actual position of the robotic arms at each connection point [9]. The control section adopts the Raspberry Pi 4 microprocessor board for the GUI control system with Python [10]. Additionally, the Raspberry Pi board controls the set drive via I2C to the Arduino Nano board, managing the drive unit in response

to object characteristics detected by a camera to determine the direction of the object within the area of interest. Machine vision techniques, explored by [11], guided the end of the robotic arm or its gripper in accurately positioning to align the workpiece and orienting the gripper in accordance with the direction of the workpiece when picking up objects. In conclusion, the objective of this research is to design two robotic arms that mimic the activities of human arms, considering basic parameters such as payload, working distance, and the creation of a perception program (Machine Vision) for sorting, handling, and inspecting objects or monitoring human activities. The fundamental principles include the low-cost production of two robotic arms with optimal performance.

2. Materials and Methods

2.1 Design and Simulate the Working Conditions of the Robotic Arm Parts.

In the design phase, a 6 degree-of-freedom Articulated Robot with a radius of 40 cm was selected. The components and dimensions are illustrated in Figure 1. Utilizing the NX program, various parts were designed and assembled to form the structure of a robotic arm by shown in appendix Figure 30. To minimize errors, movements were simulated to check for points of interference, alignment, and concentricity. The robotic arm bore the load and weight of the joints during this process. To determine the suitable size of the planetary gear ratio for the stepper motor drive (stepper motor name23) available on the market, dynamic movement simulations of the robotic arm were conducted in internal and

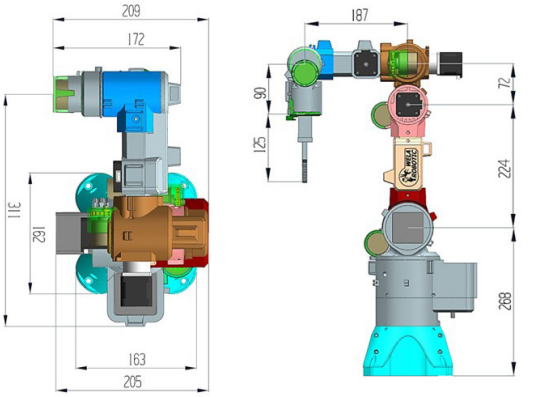

Figure 1 Size and characteristics of the robotic arm.

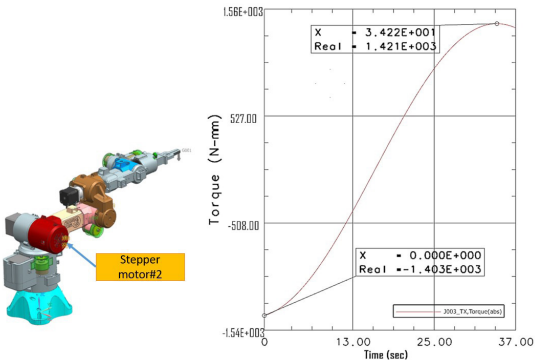

Figure 2 Torque at the 2nd rotating point of the robotic arm (operating value: 1,421 N-mm).

specified load conditions (1Kg) using the NX module of motion simulation (Multibody Dynamics) [12]. The model was represented as a rigid body with properties of mechanical material data, including mass and center of gravity of the joints. The dynamic movements of various connections were simulated, allowing the estimation of torque and gear ratio of the planetary gear under specified loads in actual operating conditions, as illustrated in Figure 2. to Figure 4. Additionally, a simulator is employed for back-calculating the inverse kinematics to validate the accuracy of the actual position. This is achieved by installing a pulse-counting rotary encoder. The

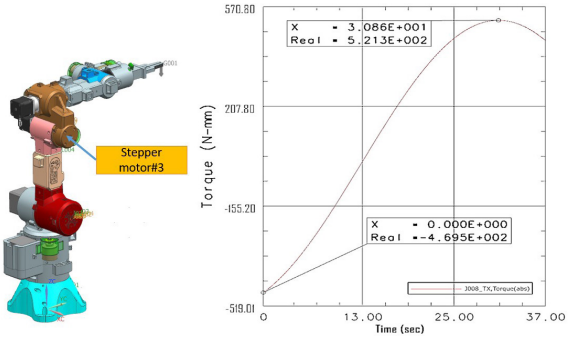

Figure 3 Torque at the 3rd rotating point of the robotic arm (operating Value: 522 N-mm).

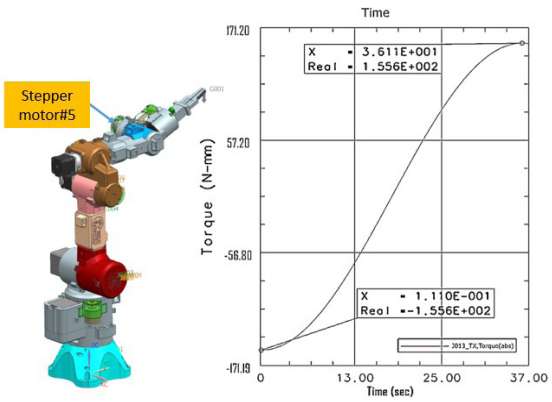

Figure 4 Torque at the 4th rotating point of the robotic arm (operating value: 156 N-mm).

steps generated by the rotation of each axis are intricately linked to the gear ratio on the spindle and the gears mounted on the rotary encoder.

2.2 Creating Structural Parts Through 3D Printing.

In the structural production process, we employed Additive Manufacturing (AM), specifically a type of Fused Deposition Modeling (FDM). by show in appendix Figure 25 shows a 3D printer this involves the use of compressed thermoplastic through a heated spray nozzle, reaching the melting temperature and layering according to the specified work style. Various materials are available for this process, and in this research, we focused on Acrylonitrile Butadiene Styrene (ABS) [13]. It has been tested and proven to exhibit relatively low temperature shrinkage effects compared to other printing materials. Tests were conducted by [14] across different frequency bands and annealing heats, showcasing its impact on elongation, particularly when producing components. With a melting temperature of 210 degrees and a heated print base ranging from 60 to 90 degrees, the results of the printed work pieces were notably satisfactory.

2.3 Robotic Arm Control Equipment and Sensors

2.3.1 Gearbox and Motor

The 1st and 2nd motors are largest, the 3rd, 4th they are second largest and the 5th, 6th is the same size. which the position joint1 and joint 2 is the bottom base, supporting the weight of the robotic arm and the load, that is used by biggest of the motor from three sizes. The selection of motors available on the market is based on the torque values shown in Figures 2–4 The torque values for each motor axis are presented in Table 1 below.

Table 1 Motor Torque and Gear Ratio.

Axis	Torque	Gear Ratio
1st: Base	1,800N-mm	1:20
2nd: Shoulder	1,800N-mm	1:20
3rd: Arm	750 N-mm	1:20
4th: Elbow	750 N-mm	1:20
5th: Forearm	520 N-mm	1:20
6th: Wrist	520 N-mm	1:1

2.3.2 Single Board Computer

The Raspberry Pi 4 serves as a microprocessor board capable of controlling GPIO pins to connect digital input and output signals to various electronic circuits. It features port pins that support 3.3V logic signals, totaling 26 pins accommodating UART, I2C, and SPI connections. Equipped with its own L inux-based operating system, the Raspberry Pi 4 includes an Application IDE. This IDE is employed to develop applications utilizing machine vision techniques, facilitating object detection through a USB camera. Additionally, it enables visual interaction with the user through a Graphical User Interface (GUI).

2.3.3 Micro Controller

The Arduino Nano is a microcontroller featuring analog pins A0–A7 for analog use. These pins are designed to receive analog values within the power range of 0–5 V. Additionally, they serve for digital use, capable of functioning as both input and output at 0V (low) and 5V (high). The microcontroller is equipped with a PWM output pin with an 8-bit resolution, providing a numerical range from 0–255. Specifically, pins A4 (SDA) and A5 (SCL) are designated for I2C communication. This functionality is employed to control stepper motors, read rotary encoder values, and display diagrams. The output of PWM and connections to other devices are illustrated in Figure 5 and appendix Figure 31.

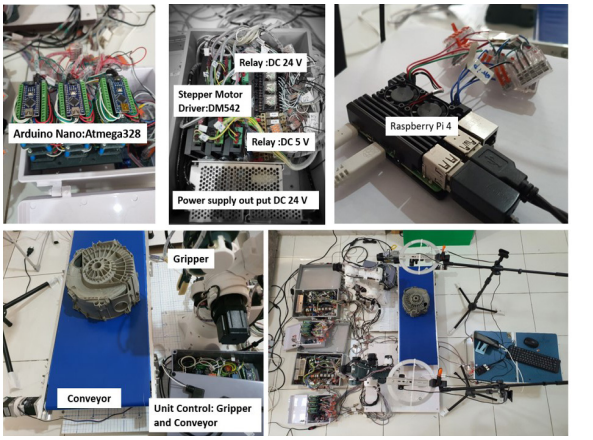

Figure 5 Equipment used to control the robotic arm.

2.4 Robotic Arm Control System

2.4.1 Control of the robotic arm movement system

The robotic arm, connected in series, establishes references to the object (Body-fixed Coordinate) at each joint. This enables the calculation of the shift from each brush reference point at the base to the subsequent reference point linked to the final tip used for holding the workpiece. These calculations provide information about the various positions of the connecting rods in a three-dimensional plane. The control system employs the principle of inverse kinematics to determine the angles of each joint necessary for placing the end of the robotic arm at the desired position. Reference [15] utilizes the Closed-form Inverse Kinematics designed for robots with a spherical wrist format. This method calculates the angles in each joint required to reach the endpoint position. The calculated values are then used to find the Inverse Transformation Matrix from the reference coordinates of the endpoint to the base axis. This process helps identify the control parameters at that pivot point, applying the Denavit-Hartenberg definition, as illustrated in Figure 6.

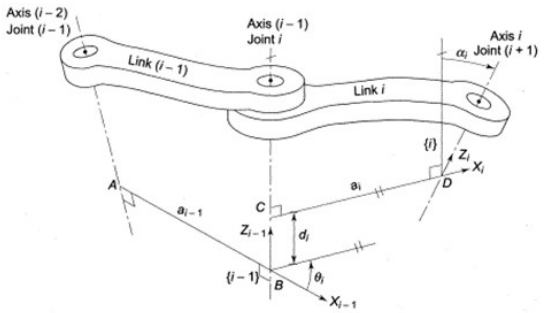

Figure 6 Configuring variables for conversion in the DH table (Springer Handbook of Robotics).

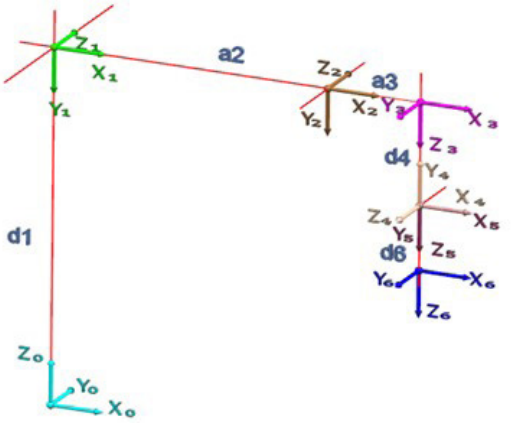

Figure 7 Direction of coordination in each rotating joint.

ai: The distance from the z_{i-1} axis to the z_i axis measured along the x_i axis.

α_i: Torsional angle between z_{i-1} and z_i around the xi axis.

d_i: The distance from x_{i-1} axis to the x_i axis measured along the z_{i-1} axis.

θ_i: Angle between the x_{i-1} axis to the x_i axis around the z_{i-1} axis.

Utilizing definitions to determine parameters through a Denavit-Hartenberg table related to links and joints in the DH Method analysis, we initiate the process with reference to Figure 7 This figure illustrates the structure of the joints in the robotic arm along with the orientation of the Cartesian coordinate axes (Frames of Reference) at each joint. Further details are presented in Table 2.

Table 2 DH Parameters from Figure 1 and Distances in Figure 7.

Joint	θ	α	d	a
1	θ_1	–90	268	0
2	θ_2	0	0	224
3	θ_3	–90	0	72
4	θ_4	90	187	0
5	θ_5	–90	0	0
6	θ_6	0	215	0

The coordinates of the point of interest obtained from the USB camera determine the position of an object in the three-dimensional system, taking into account the object's rotation in the three-dimensional space. These coordinate positions are then incorporated into one column, resulting in a 4×4 homogeneous transformation matrix with characteristics described by Equation (1).

$$A_{na} = \begin{bmatrix} Rotation_{(3\times3)} & Translation_{(3\times1)} \\ 0 \quad 0 \quad 0 & 1 \end{bmatrix} \tag{1}$$

In Equation (1), A_{na} represents the transformation operator, where "n" is the initial reference axis in the joint under consideration, and "a" is the reference axis that continues in the subsequent joints. This separation of joint and connecting rod in a 6-axis robotic arm facilitates easy calculation through algebraic and geometric methods, known as kinematic decoupling. The determination of the first three joints involves algebraic relationships presented in Equations (2)–(15):

Joint 1: $\theta_1 = tan^{-1}\left(\frac{p_{w,y}}{p_{w,x}}\right), r = \sqrt{p_{w,x}^2 + p_{w,y}^2}$ (2)

Joint 2: $S = \sqrt{p_{w,x}^2 + r^2},\ \alpha = tan^{-1}\left(\frac{p_{w,z}}{r}\right)$ (3)

Using the right triangle pattern:

$$l_3 = \sqrt{l_4^2 + l_5^2},\ \beta = cos^{-1}\left(\frac{l_2^2 + s^2 - l_3^2}{2l_2 s}\right) \quad (4)$$

$$\theta_2 = \frac{\pi}{2} - \beta - \alpha \quad (5)$$

Joint 3: $$\gamma = cos^{-1}\left(\frac{l_2^2 + l_5^2 - s^2}{2l_2 l_5}\right) \quad (6)$$

$$\theta_3 = \pi - \gamma \quad (7)$$

With a relationship:

$${}^0_1R_6(q) = {}^0_1R_3(q_{1:3}) \cdot {}^3_1R_6(q_{4:6}) \quad (8)$$

$${}^3_1R_6(q_{4:6}) = \left({}^0_1R_3(q_{1:3})\right)^T \cdot {}^0_1R_6(q) \quad (9)$$

Joint 4, 5 and 6

$${}^3_1R_6 = \left({}^0_1R_3\right)^T \cdot R_T = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \quad (10)$$

$${}^3_1R_6 = {}^3_1R_4(q_4)\,{}^4_1R_5(q_5)\,{}^5_1R_6(q_6) \quad (11)$$

$${}^3_1R_6 = \begin{bmatrix} c_4s_5c_6 - s_4s_6 & -c_4s_5c_6 - s_4c_6 & -c_4s_5 \\ s_4s_5c_6 + c_4s_6 & -s_4s_5s_6 + c_4s_6 & -s_4s_5 \\ s_5c_6 & -s_5s_6 & c_5 \end{bmatrix} \quad (12)$$

$$\theta_5 = cos^{-1}r_{33} \quad (13)$$

$$\theta_4 = tan^{-1}\left(\frac{r_{23}}{r_{13}}\right) = tan^{-1}\left(\frac{-s_4s_5}{-c_4s_5}\right) \quad (14)$$

$$\theta_6 = tan^{-1}\left(\frac{r_{32}}{r_{31}}\right) = tan^{-1}\left(\frac{-s_5s_6}{s_5c_6}\right) \quad (15)$$

With Equations (2)–(15) which show the relationship as shown appendix in the Figure 20–23

2.4.2 Movement along a Trajectory Planning

Trajectory planning involves the movement along a specified position as a function of time. This is expressed through equations of movement, incorporating angular velocity and linear velocity as functions of time. The velocity of each joint influences the angular movement of the point of interest, translating the position of the robotic arm in Cartesian coordinates (Cartesian Space Trajectories). This is followed by the movement of the robotic arm's position due to the rotation of each joint, resulting in linear motion as described in Equation (16) within the time domain.

$$\vec{r}(t) = (x_0, y_0, z_0) + t(v_x, v_y, v_z)$$
$$\vec{r}(t) = \begin{cases} x(t) = x_0 + tv_x \\ y(t) = y_0 + tv_y \\ z(t) = z_0 + tv_z \end{cases} \quad (16)$$

The position in Cartesian coordinates is determined through inverse kinematic equations to ascertain the degrees of rotation in the joints. These calculated values are then transmitted to the control unit using Jacobian Equations. These equations define the linear and angular velocities of the robotic arm frame as changes in joint parameters concerning time. These changes are derived by expanding the derivatives in scalar functions to multidimensional systems or the velocity of the point of interest. By computing the derivative of the equation at each

position with respect to all variables, let's consider a set of equations Y_i in terms of a set of variables x_i.

$$Y_i = f_i(x_1, x_1, x_1 \ldots., x_1)$$

This equation describes the change in Y_i concerning the change in x_i:

$$\begin{cases} \delta Y_1 = \dfrac{\partial f_1}{\partial x_1}\delta x_1 + \dfrac{\partial f_1}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_1}{\partial x_i}\delta x_i, \\ \delta Y_2 = \dfrac{\partial f_2}{\partial x_1}\delta x_1 + \dfrac{\partial f_2}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_2}{\partial x_i}\delta x_i, \\ \vdots \\ \delta Y_i = \dfrac{\partial f_i}{\partial x_1}\delta x_1 + \dfrac{\partial f_i}{\partial x_2}\delta x_2 + \cdots + \dfrac{\partial f_i}{\partial x_i}\delta x_i. \end{cases} \quad (17)$$

Equation (17) can be expressed in matrix form, portraying the differential relationship between each variable and the matrix function. This amalgamation of relationships is termed the Jacobian, as shown in Equation (18). Consequently, the Jacobian can be computed by deriving each equation with respect to all variables. The same principle is applied in calculating the Jacobian of the robotic arm.

$$\begin{bmatrix} \delta Y_1 \\ \delta Y_2 \\ \vdots \\ \delta Y_i \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f_1}{\partial x_1} & \dfrac{\partial f_1}{\partial x_2} & & \dfrac{\partial f_1}{\partial x_j} \\ \dfrac{\partial f_2}{\partial x_1} & \cdots & \cdots & \cdots \\ \vdots & & & \\ \dfrac{\partial f_i}{\partial x_1} & & & \dfrac{\partial f_i}{\partial x_j} \end{bmatrix} \begin{bmatrix} \delta x_1 \\ \delta x_2 \\ \vdots \\ \delta x_j \end{bmatrix} \text{ or}$$

$$[\delta Y_i] = \left[\frac{\partial f_i}{\partial x_j}\right][\delta x_j] \quad (18)$$

Utilizing conventional relationships and differentiating the position equations of the robotic arm, which are linked to both the differential movement of the robotic arm and the differential movement of the frame.

$$\begin{bmatrix} dx \\ dy \\ dz \\ \delta x \\ \delta y \\ \delta z_i \end{bmatrix} = \begin{bmatrix} \\ Robot \\ jacobian \\ \\ \\ \end{bmatrix} \begin{bmatrix} d\theta_1 \\ d\theta_2 \\ d\theta_3 \\ d\theta_4 \\ d\theta_5 \\ d\theta_6 \end{bmatrix} \text{ or}$$

$$[D] = [J][D_\theta] \quad (19)$$

In the equation dx, dy and dz in $[D]$ denote different motions along the X, Y, and Z axes, respectively. δx, δy and δz in $[D]$ represent the rotation in the differential sections of the X, Y, and Z axes. Additionally, $[D_\theta]$ signifies the differential motion of the joint as shown in Equation (19). Both matrices are divided by dt and it's important to note that these matrices represent velocity rather than derivative motion. Which the Figures 8 and 9 show the angular displacement of six joints in robotic arm 1 and arm 2 over 180 seconds as they perform tasks by references in appendix Figure 32 It shows how each joint's position changes over time during task execution, with separate plots for different sets of joints.

2.5 Software

2.5.1 Algorithm

The algorithm serves as a comprehensive mapping of image characteristics, enabling the model to learn object classes and their positions in the image which shows each of process on Figure 10 Utilizing the CNN (Convolutional Neural Network)

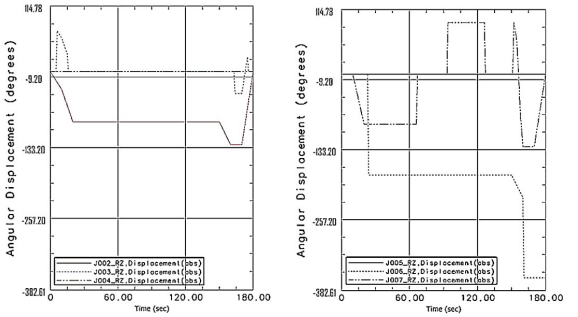

Figure 8 The six joint degrees of the first robotic arm during various tasks, including picking up the workpiece, relocating to inspect, rotating the work for inspection at four points, and placing the workpiece.

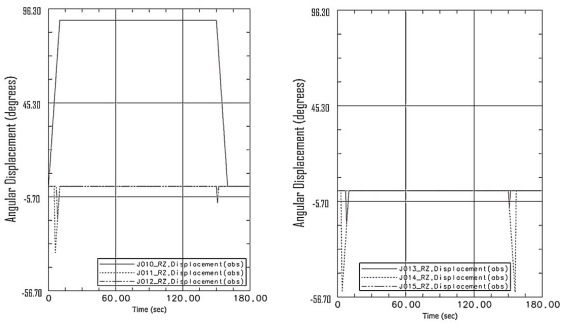

Figure 9 The six joint degrees of the second robotic arm during various tasks, including moving to inspect, rotating the work position at four points, and returning to the original position.

format for predictions [16] the algorithm comprises an input layer and an output layer. Its calculation involves layers with the following components:

2.5.1.1 Convolution Layer: This layer filters specific features of image data by analyzing convolution values between input values and the Kernel, extracted from the image dataset.

2.5.1.2. Rectified Linear Unit Activation Layer: This layer enhances relationships between input

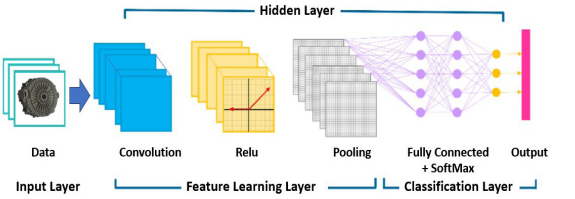

Figure 10 CNN network processed with images for learning.

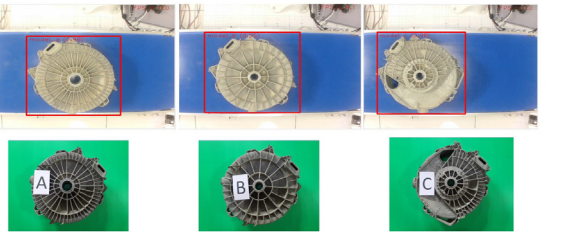

Figure 11 The outcomes of object detection obtained through the CNN algorithm processed on images during the learning phase.

data and results. The ReLU function, a non-linear function, facilitates learning and produces a feature map.

2.5.1.3. Pooling Layer: Reducing the size of the data, this layer samples the data to decrease the feature map's size, using the average function for faster processing.

2.5.1.4. Classification Layer: Also known as the Fully Connected Layer, this layer outputs a K-dimensional vector, with K representing the number of classes for image prediction in the CNN network. Each value within the vector represents the probability of belonging to a specific class for object examination, as illustrated in Figure 11.

2.5.2 Describing object boundaries (Contouring)

Alignment is determined based on the principle [17] of thresholding, where boundary light

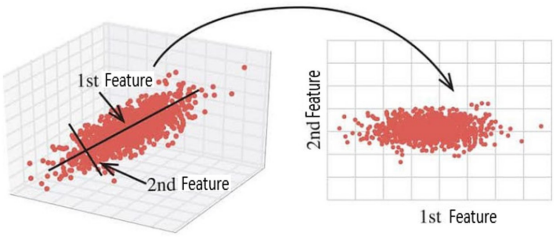

Figure 12 The characteristics of the data undergo dimension reduction. [17]

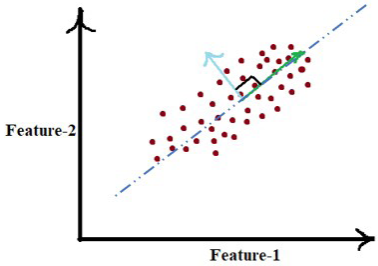

Figure 13 The distribution of data following dimensionality reduction.

intensity values are utilized to iteratively separate the desired area until a closed line is formed around it, representing a shape. The center point of this shape is calculated by determining the pixel position and converting it into 2-dimensional coordinates (X and Y) of the robotic arm. The orientation is then determined using the technique of Object Orientation Detection in Images using PCA and OpenCV. This algorithm not only reduces data size and enhances interpretability but also minimizes data loss by reducing dimensionality. It aids in identifying crucial features within a dataset, facilitating easy 2D and 3D plotting and determining the linear combination order of variables. The points are plotted on a 2D graph based on Feature-1 and Feature-2, highlighting key points of PCA (Principal Component Analysis).In Figures 12 and 13, it is evident that every point in the distribution exhibits a linear trend, allowing for the reduction of 2D data to 1D. All points follow the watercolor lines, indicating that if the location of a point is known and it lies on the blue line, more information about that point can be inferred compared to knowing its location along Features 1 or 2. These aids in determining the true direction. The main components are extracted from the data points, with the graph illustrating that the primary component is represented by the green line, describing the maximum variance. The blue component is perpendicular to the green vector, determined by the magnitude of the eigenvectors encoded in the eigenvalues. The origin of these vectors is the center of all points, crucial for object orientation detection in images, as illustrated each model in appendix Figures 26–28.

2.5.3 Robotic Arm Working Steps and Control Program

The operational process of the robotic arm is depicted in Figure 22, showcasing the collaboration among the Raspberry Pi board, USB camera, Arduino Nano board, conveyor belt, and the robotic arm. The workpiece travels along a belt that passes through the camera's field of view. Upon detecting the workpiece, the Raspberry Pi board commands the Arduino Nano board, responsible for controlling the belt, to halt its movement. The camera captures images and transmits the data to the Raspberry Pi board for processing based on the model of interest, converting it into X, Y coordinates relative to the base coordinates of the robotic arm. Subsequently, the calculated degree data is sent to the Arduino Nano board, orchestrating the motor movements at each joint, facilitating the rotation

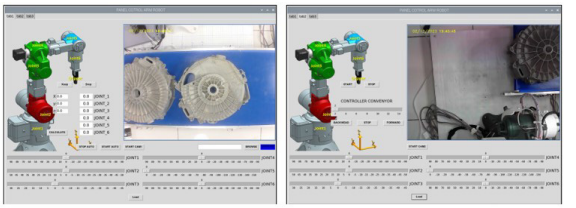

Figure 14 The interface employed for controlling the robotic arm using Python on the control board in the Raspberry Pi board.

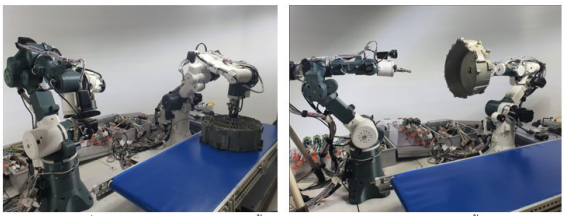

Figure 15 Operation involving the pickup of workpieces and quality inspection conducted by the two robotic arms.

of the first robotic arm to grasp the workpiece and transport it to the position of the second robotic arm. The second robotic arm inspects the workpiece for production accuracy, controlled via a screen interface. To initiate the work, an interface appears, triggered by checking both cameras upon pressing the tab2 button. Pressing the 'star cam2' button reveals the monitor of the second camera. Following this, press tab1 and 'star cam2' to display the second camera screen. The operation commences by pressing the start button, prompting the system to verify the connection between the Raspberry Pi board and other peripherals. Both robotic arms then move to the Home Position, signifying readiness to operate in an automatic system. Pressing 'start auto' in Figures 14–15 initiates the automatic working sequence.

3. Results

In this research test, the center coordinate of the work piece and the considered orientation angle are measured relative to the coordinates of both robotic arm base by shown in appendix Figure 24. This is achieved by referring to the coordinates obtained from image processing as X, Y coordinates and the relative degrees with in reference to the coordinates of the first robotic arm base with independent orientation. Measurements are carried out thirty times. At the initial stages of development, significant errors were observed in both position and directional coordinates quite highly. To align with cost-effectiveness objectives in production and They are utilized to find deviations and averages, which was referenced against ISO 9283 standards but does not fully comply. as the standard deviation is relatively high, and the dispersion deviation is relatively wide. for checking standard deviation for repeatability as cycle Figure 16 by measure discrete. Scale readings, as depicted in Figure 17.

Compare the values processed by photography with the actual measurement of the mechanical arm's move to position to the coordinates specified by the image. The images of each model A, B and C are shown in Tables 3–4. In addition, it is a measurement including the difference between the actual value and the measured value from the image. In order to find the differences in the mean and standard deviation that led to the tolerances for use as shown in Tables 5–7.

$$s = \sqrt{\frac{\sum (X - \bar{x})^2}{n-1}} \tag{20}$$

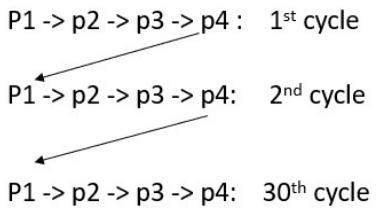

Figure 16 Sequence of movement.

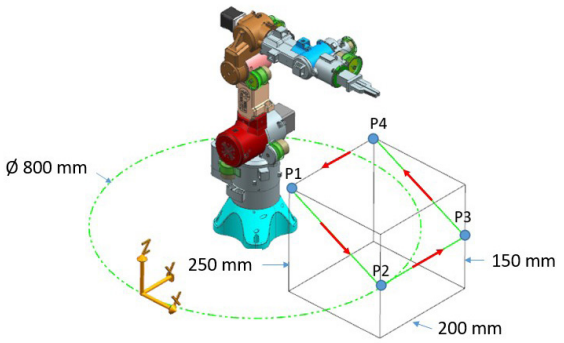

Figure 17 Position plan for test.

In the equation for determine S standard deviation X represents information about the difference between the actual position and the measured position. $\overline{x}$ is the average of the difference between the actual and measured positions and n is the amount of data.

Table 3 Robot Arm 1 Differential Error Table.

Robot1	P1			P2			P3			P4		
	x	y	z	x	y	z	x	y	z	x	y	z
mean	–0.3	–1	–0.6	0.4	–0.5	1.1	0.7	–1.1	–1.3	0.5	–0.3	1.7
s	1.2	1.4	1.1	1.6	1	1.4	1.2	1	1.9	1.3	1.3	1.1

Table 4 Robot Arm 2 Differential Error Table.

Robot2	P1			P2			P3			P4		
	x	y	z	x	y	z	x	y	z	x	y	z
mean	–0.1	–1.1	–0.9	0.3	–1.2	–1.0	0.7	–0.8	0.9	1.1	–2.0	1.7
s	1.1	1.1	1.2	1.7	1.1	1.5	1.0	1.1	1.2	0.9	1	1

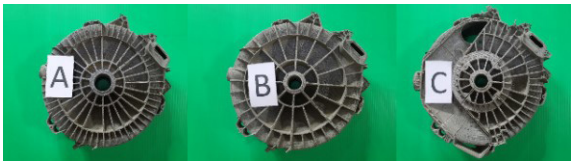

Figure 18 Picture of model A, B and C.

Table 5 Differential Tolerances of Model A and Pictures

	A/Pic @X	A/Pic @Y	A/Pic @degree
mean	0.1	0.4	–3
s	1.73	2.06	11.46

Table 6 Differential Tolerances of Model B and Pictures.

	B/Pic @X	B/Pic @Y	B/Pic @degree
mean	1.1	1	–1
s	1.88	1.18	7.14

Table 7 Differential Tolerances of Model C and Pictures.

	C/Pic @X	C/Pic @Y	C/Pic @degree
mean	0.1	3	–6.9
s	1.22	1.22	11.46

The characteristic of the difference in each model, wherein C has a concave profile, necessitating the installation of a low-speed support device. A and B exhibit similar characteristics, but they differ in that A has more force-bearing ribs than B, which supports relatively high rotational speeds in operation. By detail as shown in Table 8 and illustrated in Figure 18

Table 8 Characteristics of A, B and C Models.

	Model A	Model B	Model C
RPM	1200	800	650
Volume	412.21	355.67	420.92

From the position test results, it was found that in both robotic arms, the average deviation in the x, y, and z axes falls within the range of –2 to 1.7 centimeters. The deviation was less than 2 centimeters. Regarding the movement of the robotic arms to positions in the image, in models A and B, the expected values compared to the actual photographs differed. In the X and Y axes, the average value was approximately 1.1 centimeters, with a standard deviation of 2.1 centimeters. The average angular error ranged from –3 to –1 degrees, with a standard deviation of 11.5 degrees. For model C, average values in the x and y axes were approximately 3 centimeters, with a standard deviation of about 1.2 centimeters. The average angular error was approximately –7 degrees, with a standard deviation of about 11.5 degrees. There was greater variability observed across all axes, with multiple factors contributing to increased deviation. This started with the components of robotic arms D-1, K-1, and G-1, particularly in Figure 29, where high forces must be sustained, resulting in wear and tear and increased deviation as depicted in Figure 19. In the set of Planetary gear set, it was found that there was a relatively high amount of backlash.

These deviations may be acceptable within a certain range of coordinates depending on the job requirements. In image processing, deviations occur due to factors such as the position of the light shining on the workpiece, the camera's placement and angle during image capture, and the reflection of the surface. These factors affect processing and the determination of the orientation of objects. Objects with clear geometric shapes, such as rectangular or similar shapes, allow for more accurate

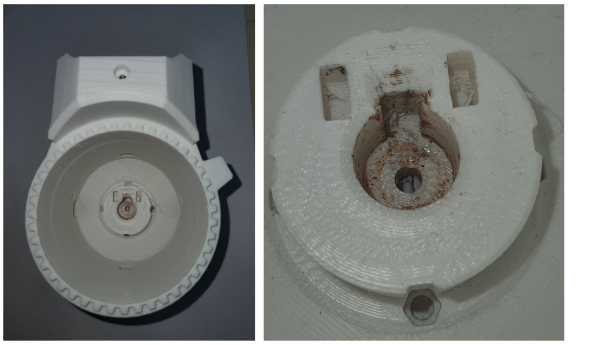

Figure 19 Wear areas causing errors in the position of each joints.

orientation determination. Additionally, workpieces with wide variations within the Profile boundary, such as those in Model C, result in significant deviations, necessitating the use of algorithms with greater accuracy.

4. Discussion and Conclusion

This research involves designing and developing a dual-arm robotic system integrated into the production line, with a focus on minimizing production costs. Raspberry Pi boards serve as controllers. During the processing phase, objects within the specified area of interest are detected and classified using a CNN algorithm. Template matching is then employed to determine shape boundaries in the image, aiding in calculating the object's X and Y centroid positions and orientation angle relative to the robotic arm's X-axis. The Raspberry Pi processes and sends data through an I2C bus to control the motors at all six joints of Arms 1 and 2, each equipped with a rotary encoder to measure movement distance. Subsequently, Arm 1 moves to the grab position while Arm 2 takes pictures for part inspection, alerting for discrepancies. Both arms

are controlled via a user interface for autonomous operation, with testing showing satisfactory performance. For processing, coordinates within a range of To compensate, it is recommended to use a deviation of 3±1.22 centimeters with a mean of –1 to 1.125 centimeters for the x, y, and z axes. For orientation angles, –7±11.5 degrees with a mean of –3.65 to 0 degrees are suggested as compensation values during gripping, along with 3 centimeters for the x, y, and z axes of the work piece position as a starting point. Values representing the correlation across different axes are displayed in the form of graphs, showing the overall mean of each value, as illustrated Figures 33–40 in the Appendix. Future development includes upgrading arm components to engineering plastics for better force resistance, flexibility, and improved force distribution. To enhance the accuracy of a plastic robot arm, stronger 3D printing materials like PA6 or PA66 with carbon fiber should be used. Components should be reinforced for stability, and precise manufacturing should be ensured. Joint design should be optimized to minimize backlash. The efficiency of real-time feedback sensors should be improved, vision systems should be integrated for additional feedback, and advanced control algorithms for trajectory optimization should be developed.

5. Acknowledgments

The research team expresses sincere gratitude to the following company and individuals for their invaluable support and contributions, which were pivotal to the successful completion of this research. We extend our appreciation for the opportunities provided during this study.

1. Electrolux Thailand Factory (Rayong)
2. Ms. Paweena Sukpaiboon
3. Mr. Theeraphong Thanasitrangsi
4. Mr. Phalatphol Wikranwong

References

[1] E. Ananias and P. D. Gaspar, “A Low-cost collaborative robot for science and education purposes to foster the Industry 4.0 implementation,” *Applied System Innovation*, vol. 5, no. 4, pp. 72–98, 2022.

[2] Zhihuijun. I made a dummy robotic arm from scratch. (Oct. 7, 2021). YouTube. [Online Video]. Available: https://www.youtube.com/watch?v=F29vrvUwqS4

[3] P. Ociepka and K. Herbus, “Strength analysis of parallel robot components in PLM Siemens NX 8.5 program,” in *IOP Conference Series: Materials Science and Engineering*, 2015, vol. 95.

[4] W. Polsan and P. Kunthong, “Fatigue lifetime estimation of rot samlor skylab structure,” presented at the 22nd Conference of Mechanical Engineering Network of Thailand, Bangkok, Thailand, Oct. 15–17, 2008 (in Thai).

[5] L. Armesto. An example on how to compute the inverse kinematics of a Robot Robotic Systems. (Dec. 28, 2020). YouTube. [Online Video].Available: https://www.youtube.com/watch?v=wmE1KQJmzVM

[6] C. Annin. 6axis robot kinematics Part 3, YouTube. (Jun. 3, 2017). [Online Video]. Available: https://www.youtube.com/watch?v=oiAiuFK7yi0

[7] J. Wang, C. Dai, K. Shi, and R. Qin, “Research on rigid body motion tracing in space based

on NX MCD," presented at the International Conference on Robotics and Mechatronics (ICRoM 2017), *Series: Materials Science and Engineering*, vol. 320, 2018.

[8] K. Chantarasamai, W. Thasana, T. Homchampa, and P. Chalisathanakritdakarn, "The study of factors affecting mechanical properties of 3D printing with a 6-axis robotic arm," *Agriculture and Technology Journal*, vol. 3, no. 2, pp. 1–11, 2022 (in Thai).

[9] Y. Tian, W. Feng, M. Ouyang, H. Bian, and Q. Chen, "A positioning error compensation method for multiple degrees of freedom robot arm based on the measured and target position error," *Advances in Mechanical Engineering*, vol. 14, no. 5, pp. 1–13, 2022.

[10] H. Smatla, R. Shaw, S. Esse, and L. Rhodes, "Implementing nonlinear control of a six degree of freedom robotic arm on a raspberry Pi 3," presented at the 35th Florida Conference on Recent Advances in Robotics, Embry-Riddle Aeronautical University, Prescott, Arizona, May 12–13, 2022.

[11] A. Sodemann. Robotics 2 2018, YouTube. (Dec. 12, 2017). [Online Video]. Available: https://www.youtube.com/playlist?list=PLT_0lwItn0sAfi3o4xwx-fNfcnbfMrXa7

[12] R. Anderl and P. Binde, *Simulations with NX: Kinematics, FEA, CFD, EM and Data Management with Numerous Examples of NX 9*. Hanser Publications, Munich, Germany, 2014.

[13] R. B. Kristiawan, F. Imaduddin, D. Ariawan, Ubaidillah, and Z. Arifin, "A review on the fused deposition modeling (FDM) 3D printing: Filament processing, materials, and printing parameters," *Open Engineering*, vol. 11, no. 1, pp. 639–649, 2021.

[14] P. Ferretti, C. Leon-Cardenas, G. M. Santi, M. Sali, E. Ciotti, L. Frizziero, G. Donnici, and A. Liverani, "Relationship between FDM 3D printing parameters study: Parameter optimization for lower defects," *Polymers*, vol. 13, no. 13, pp. 921–929, 2021.

[15] L. Armesto. (2020, Dec. 22) Closed-form Inverse Kinematics for Robots with a Spherical Wrist Robotic Systems, YouTube. [Online]. Available: https://www.youtube.com/watch?v=hEQ1p2Wffm

[16] L. V. G. Khánh. Introduction to object detection on Raspberry Pi. (Dec. 9, 2021). TensorFlow. [Online Video]. Available: https://www.youtube.com/watch?v=mNJXEybFn98&t=1s

[17] V. Lendave (2021). "Detecting Orientation of Objects in Image using PCA and OpenCV, Analytics India Magazine." [Online]. Available: https://analyticsindiamag.com/detecting-orientation-of-objects-in-image-using-pca-and-opencv/

Appendix

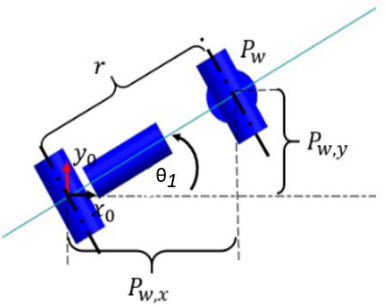

Figure 20 Relationship between θ_1 and other parameters in the calculation in top view.

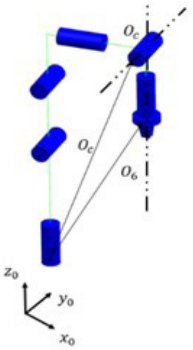

Figure 21 Relationship between θ_1 and other parameters in the calculation in isometric view

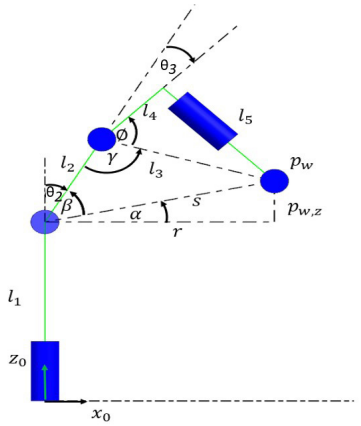

Figure 22 Joint separation for Kinematic Decoupling calculation.

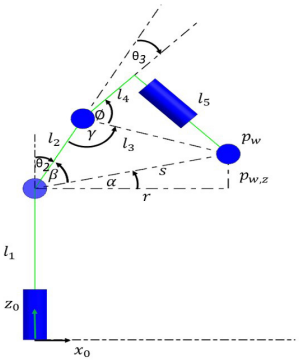

Figure 23 Relationship between θ_2 and θ_2 and other parameters used in inverse kinematics calculation.

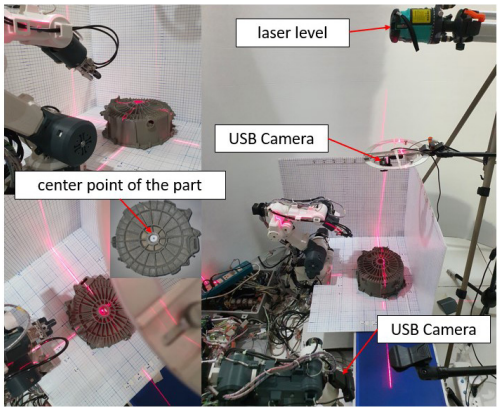

Figure 24 Components and equipment used for measurement position.

Figure 25 3D Printer Flashforge Creator Pro2.

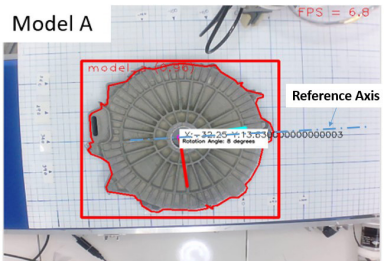

Figure 26 Results of detection, coordination, and workpiece orientation angle for model A.

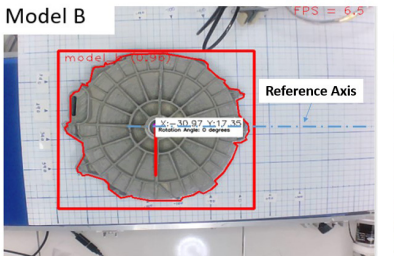

Figure 27 Results of detection, coordination, and workpiece orientation angle for model B.

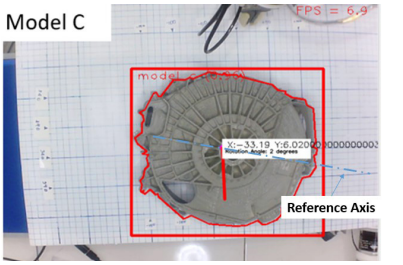

Figure 28 Results of detection, coordination, and workpiece orientation angle for model C.

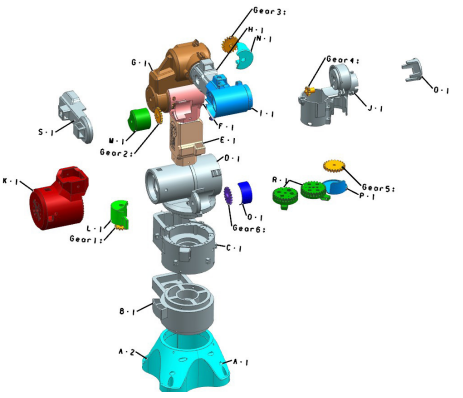

Figure 29 Parts produced by 3D printers.

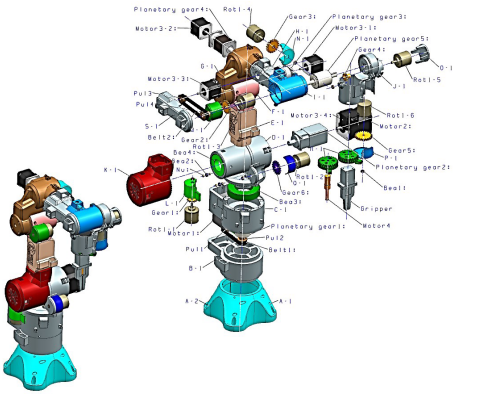

Figure 30 Components of a mechanical arm.

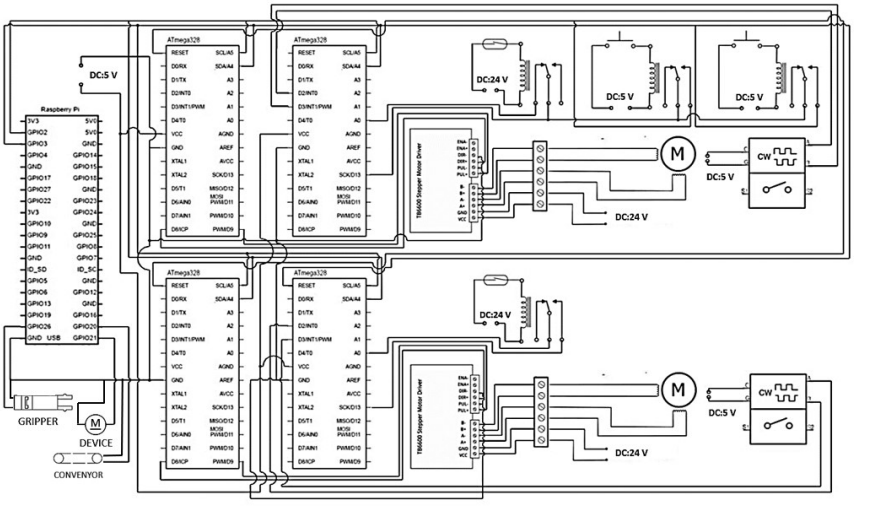

Figure 31 Diagram showcasing the connection of various devices to the Raspberry Pi board for controlling operations.

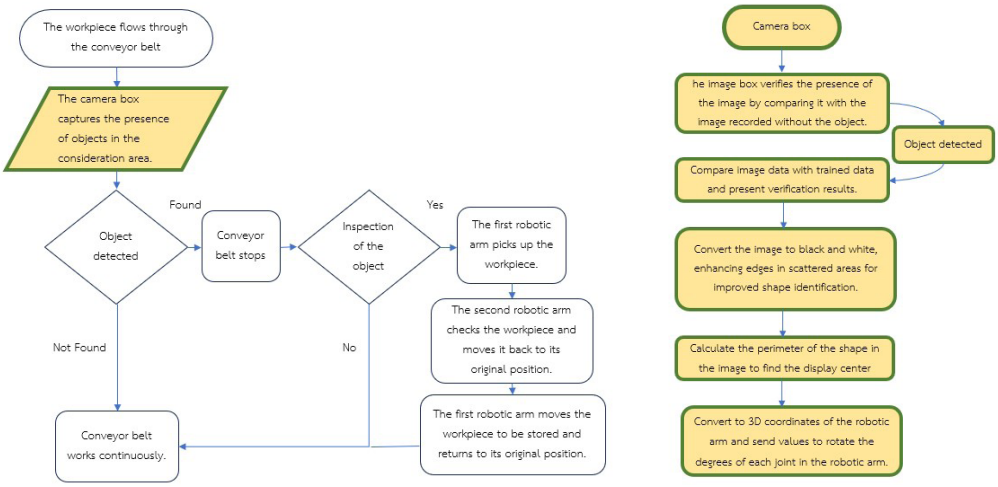

Figure 32 Workflow of the two robotic arms, conveyor belt, and camera controlled by the Raspberry Pi board.

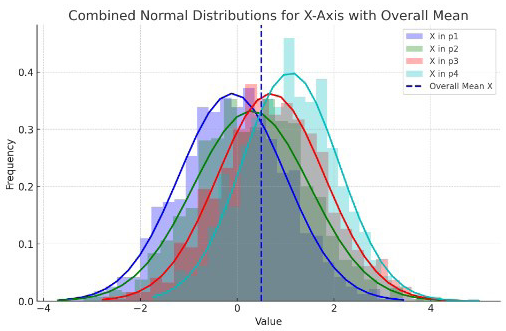

Figure 33 The data for p1, p2, p3, and p4 indicate that the overall mean for the x-axis of Robot Arm 1 is 0.5.

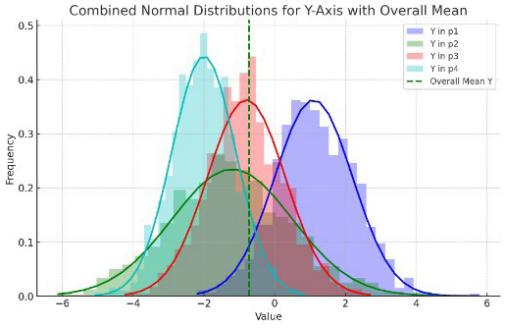

Figure 34 The data for p1, p2, p3, and p4 indicate that the overall mean for the y-axis of Robot Arm 1 is –0.725

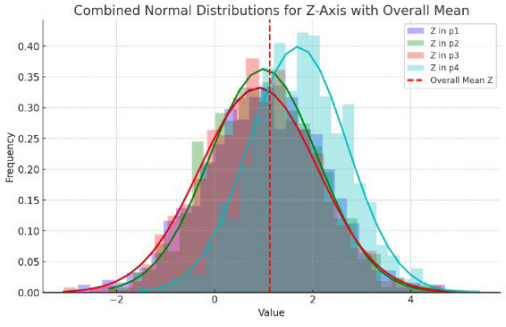

Figure 35 The data for p1, p2, p3, and p4 indicate that the overall mean for the z-axis of Robot Arm 1 is 1.125

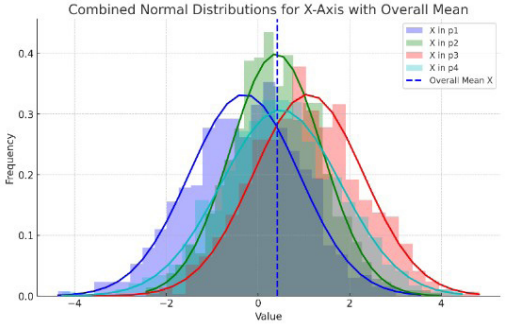

Figure 36 The data for p1, p2, p3, and p4 indicate that the overall mean for the x-axis of Robot Arm 2 is 0.425

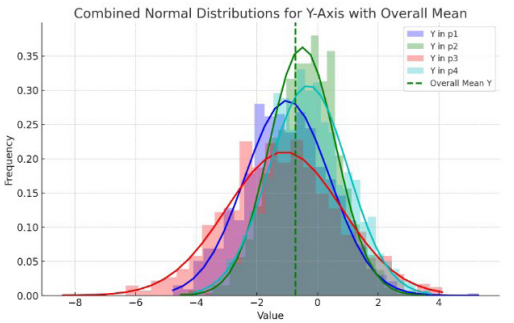

Figure 37 The data for p1, p2, p3, and p4 indicate that the overall mean for the y-axis of Robot Arm 2 is -0.725

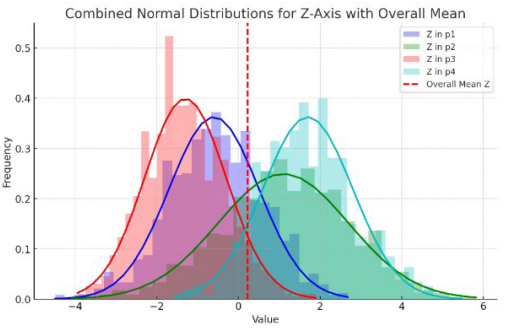

Figure 38 The data for p1, p2, p3, and p4 indicate that the overall mean for the z-axis of Robot Arm 2 is 0.225

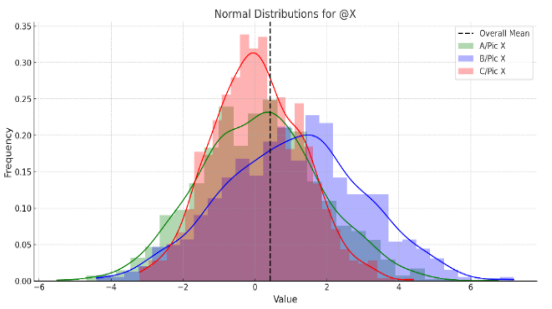

Figure 39 Comparing the actual positions in each model with the positions in the image along the x-axis, with an overall mean of 0.42 for the x-axis.

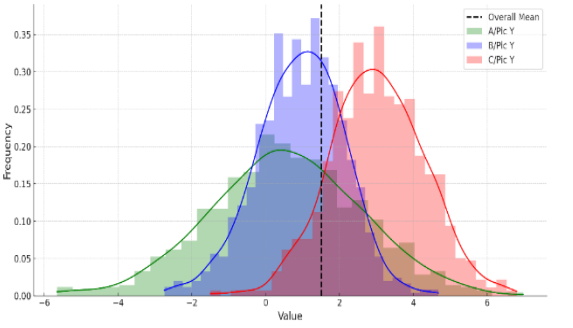

Figure 40 Comparing the actual positions in each model with the positions in the image along the y-axis, with an overall mean of 1.50 for the x-axis.

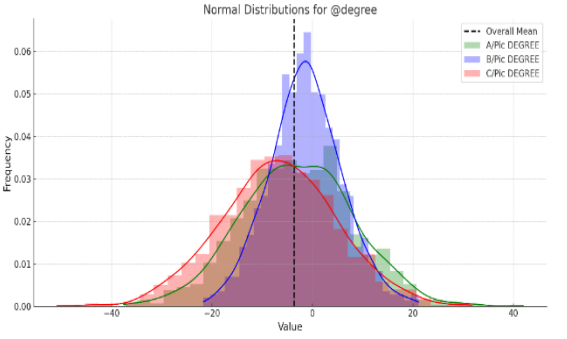

Figure 41 Comparing the actual angles in each model with the angles in the image, with an overall mean of –3.65 degrees.