Design a Low-cost Delta Robot Arm for Pick and Place Applications Based on Computer Vision

Le Hoai Phuong
Industrial Maintenance Training Center, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City, Vietnam

Vo Duy Cong
Industrial Maintenance Training Center, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City, Vietnam

Thai Thanh Hiep
Faculty of Transportation Engineering, Ho Chi Minh City University of Technology (HCMUT), VNU-HCM, Ho Chi Minh City, Vietnam

In this paper, we develop a low-cost delta robot arm for grasping objects of unspecified size thanks to a vision system. Stepper motors are used instead of ac servo motors to build a low-cost delta robot arm. Furthermore, we use available materials and machining methods such as laser cutting and 3d printing instead of CNC milling and turning to reduce fabrication costs. The controller is based on a low-cost embedded controller - Arduino Uno for controlling the robot's motion. The vision system is constructed to determine the 3D coordinate of objects in the workspace as well as the sizes of objects. The gripper is opened with a distance of two fingers equal to the size of the objects, and the robot is controlled to the objects' coordinates to grasp them. An application to pick up objects on a conveyor belt is developed to validate the design. The experimental results show that the robot system works correctly, the robot arm moves smoothly, and the information determined by the vision system has a small error, ensuring that the robot can accurately pick up products.

Keywords: *delta robot, computer vision, pick and place, stepper motor, 3D printing.*

1. INTRODUCTION

Robots have become popular in automation in recent years [1-5]. The applications of robots are increasingly expanding, gradually replacing workers in production lines. Robotic arms are designed to move like human arms to conduct various tasks in industry, agriculture, military... [6-8]. Engineers can write motion programs for robots to perform various tasks that are continuous, repetitive, dangerous, or require high accuracy. Various types of robot arms have been developed to meet different industry requirements. Each robot has its pros and cons to suit different applications. Three types of robot arms commonly used in the industry are 6-DOF robot arms, SCARA robot arms, and delta robot arms.

Delta robots are parallel robots commonly used in industrial applications requiring high speed and accu–racy [9-10]. Common applications of delta robot arms are pick-and-place, assembly, palletizing, sorting, and packaging. During the motion, the bottom mobile platform of the delta robot arm is always parallel to the top fixed platform thanks to three closed parallel chains (see Figure 1). The delta robot can move in the X, Y, and Z directions without the rotation of the end-effector. The high-speed movement of the delta robot is due to the closed loop kinematic chain mechanism whose bottom moving platform is connected to the top fixed base by separate kinematic chains. Each chain consists of an upper arm connected to the fixed base. All upper arms are connected to the end-effector in a triangle form. The end-effector can be connected to an addi–tional degree of rotation or used to mount tools. The design of the delta robot arm has been presented in many studies, from mechanical to electrical parts. Alt–hough the basic idea of the mechanical design is the same, the final results are different in the length and struct of the arms and mechanism details to meet different requirements for payload and cycle time.

Determining the robot's workspace is an important issue in the design process. Depending on the specific work environment, the applications require different workspaces. The robot must be designed to move to the required locations in the workspace [11]. The main parameters that determine the size of the robot's workspace are the length of the links and the limitation of the rotation angle of the motor [12]. To determine the workspace of the robot from the length of the links, it is necessary to develop a forward kinematic problem stating the relationship between the X, Y, and Z coordinates of the end-effector with the length of the links and the rotation angles. Additionally, to move the robot to the exact position, the inverse kinematic problem is used to convert the X, Y, and Z coordinates to the rotation angle of the motors that stimulate the upper arms [13]. The algebraic or geometric method can be utilized to find the solution for the forward and inverse kinematic problems and determine the robot's workspace [14].

Integrating the visual sensors with the robotic arms has further increased the robustness of the robot system. Vision sensors help the robot better sense its sur–roundings. Vision sensors provide more useful and insightful information than other types of sensors (proximity, position, etc.). The applications of robots are also expanded, some common applications of the robotic vision system are dynamic grasping, automatic sorting, visual servoing, vision guide robot, and vision-based inspection, ... [15-20]

Received:October 2022, Accepted: February 2023
Correspondence to: Vo Duy Cong
Industrial Maintenance Training Center,
Ho Chi Minh City University of Technology, Vietnam
E-mail: congvd@hcmut.edu.vn
doi: 10.5937/fme2301099P

Because of their wide application in industrial fields, companies have developed robotic systems with full functions, serving many different requirements. Vision systems and control programs are also developed almost completely and packaged into software for easy use by users. However, this has made the robots very expensive, only suitable for large-scale production companies in developed countries. In developing countries like Vietnam, most manufacturing enterprises are small and medium, and labor costs are also very low. Although the application of robots brings higher productivity, the initial investment cost is large, and the system maintenance cost is also expensive. So, industrial robots with complete features at a high cost are not suitable for small and medium companies. Robots usually only perform certain (sometimes simple) tasks in production lines, so fully functional robots are unnecessary. Understanding that problem, in this paper, we develop a low-cost robotic system that applies computer vision to pick objects on conveyors. The system consists of a 4-DOF delta robot with a gripper mount at the end-effector to grab moving objects on the conveyor and a camera mounted above the conveyor to capture the image of the objects. A personal computer processes images to provide information for the robot. We use materials and machining methods such as laser cutting and 3d printing instead of CNC milling and turning to build the system to reduce fabrication costs. The controller is based on a low-cost embedded controller - Arduino Uno for controlling the robot's motion.

The remainders of the paper are organized as follows: Section 2 presents the kinematic of the delta robot; Section 3 presents the mechanical design of the robot; the electrical system is presented in Section 4; Section 5 presents the computer vision to detect and calculate the position of objects on the conveyor; the experiment results are shown in Section 6, and Section 7 is the Conclusion.

2. THE ROBOT KINEMATIC EQUATION

In this section, the forward kinematic equation and the inverse kinematic equation are presented. We used the geometric method two solve the kinematic problems.

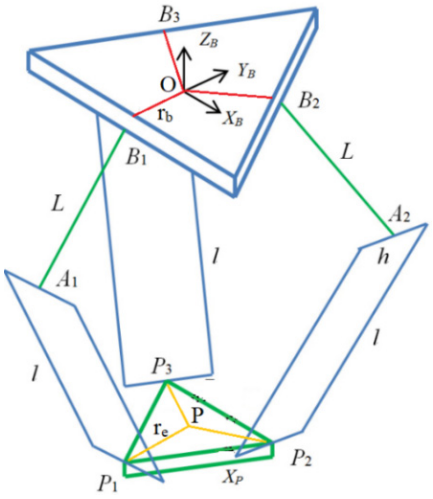


Figure 1. The Delta Robot Kinematic Diagram

Figure 1 illustrates the Delta Robot Kinematic Diagram. The robot base frame is located at the centroid O_B of the bottom fixed base. The joint variables are $\theta_i, i = 1 - 3$. The centroid of the moving platform is $BP = \{x_0, y_0, z_0\}$. The length of the upper arm is L, and the length of the lower arm is i. The distance between centroid O and the hip points B_i is $OB_i = r_b$, and the distance between centroid P and the anklepoints P_i is $PP_i = r_e$. From geometric relations in the kinematic chains, we have:

$$\overrightarrow{OB_i} + \overrightarrow{B_iA_i} + \overrightarrow{A_iP_i} = \overrightarrow{OP} + \overrightarrow{PPi} \tag{1}$$

or

$$\overrightarrow{B_iA_i} = \overrightarrow{OP} + \overrightarrow{PPi_i} - \overrightarrow{OB_i} - \overrightarrow{A_iP_i} \tag{2}$$

The coordinates of the vectors are:

$$\overrightarrow{OB_1} = \begin{bmatrix} 0 \\ -r_b \\ 0 \end{bmatrix}; \overrightarrow{OB_2} = \begin{bmatrix} \frac{\sqrt{3}}{2} r_b \\ \frac{1}{2} r_b \\ 0 \end{bmatrix}; \overrightarrow{OB_3} = \begin{bmatrix} -\frac{\sqrt{3}}{2} r_b \\ \frac{1}{2} r_b \\ 0 \end{bmatrix} \tag{3a}$$

$$\overrightarrow{B_1A_1} = \begin{bmatrix} 0 \\ -L\cos\theta_1 \\ -L\sin\theta_1 \end{bmatrix}; \overrightarrow{B_2A_2} = \begin{bmatrix} \frac{\sqrt{3}}{2} L\cos\theta_2 \\ \frac{1}{2} L\cos\theta_2 \\ -L\sin\theta_2 \end{bmatrix}; \overrightarrow{B_3A_3} = \begin{bmatrix} \frac{\sqrt{3}}{2} L\cos\theta_3 \\ \frac{1}{2} L\cos\theta_3 \\ -L\sin\theta_3 \end{bmatrix} \tag{3b}$$

$$\overrightarrow{PP_1} = \begin{bmatrix} 0 \\ -r_e \\ 0 \end{bmatrix}; \overrightarrow{PP_2} = \begin{bmatrix} \frac{\sqrt{3}}{2} r_e \\ \frac{1}{2} r_e \\ 0 \end{bmatrix}; \overrightarrow{PP_3} = \begin{bmatrix} -\frac{\sqrt{3}}{2} r_e \\ \frac{1}{2} r_e \\ 0 \end{bmatrix} \tag{3c}$$

Substituting all the above values into Equation (2) yields:

$$\overrightarrow{B_1A_1} = \begin{bmatrix} x \\ y + L\cos\theta_1 + a \\ z + L\sin\theta_1 \end{bmatrix} \tag{4a}$$

$$\overrightarrow{B_2A_2} = \begin{bmatrix} x - \frac{\sqrt{3}}{2} L\cos\theta_2 + b \\ y - \frac{1}{2} L\cos\theta_2 + c \\ z + L\sin\theta_2 \end{bmatrix} \tag{4b}$$

$$\overrightarrow{B_3A_3} = \begin{bmatrix} x + \frac{\sqrt{3}}{2} L\cos\theta_3 - b \\ y - \frac{1}{2} L\cos\theta_3 + c \\ z + L\sin\theta_3 \end{bmatrix} \tag{4c}$$

where:

$$a = -r_e + r_b; b = \frac{\sqrt{3}}{2} r_e - \frac{\sqrt{3}}{2} r_b; c = \frac{1}{2} r_e - \frac{1}{2} r_b$$

The lengths of lower arms are constant, $\left\|\overrightarrow{B_1A_1}\right\| = \left\|\overrightarrow{B_2A_2}\right\| = \left\|\overrightarrow{B_3A_3}\right\| = l$. So, we obtain the kine–matics equations for the Delta Robot:

$$x^2 + \left(y + L\cos\theta_1 + a\right)^2 + \left(z + L\sin\theta_1\right)^2 = l^2 \tag{5a}$$

$$\left(x - \frac{\sqrt{3}}{2} L\cos\theta_2 + b\right)^2 + \left(y - \frac{1}{2} L\cos\theta_2 + c\right)^2 + \tag{5b}$$
$$+ \left(z + L\sin\theta_2\right)^2 = l^2$$

$$\left(x + \frac{\sqrt{3}}{2} L\cos\theta_3 - b\right)^2 + \left(y - \frac{1}{2} L\cos\theta_3 + c\right)^2 + \tag{5c}$$
$$+ \left(z + L\sin\theta_3\right)^2 = l^2$$

Assuming that the angle of each upper arm θ_i is known in advance, the solution of the system of equations (5) is the coordinate *x, y, z* of the centroid P of the moving platform. Geometrically, the solution of Equation (5) is the intersection of three spheres with definite centroid and radius. So, it is called the forward kinematic problem.

To solve the inverse kinematic problem, the system of Equations (5) is transformed into equations in terms of θ_i.

$$E_i \cos\theta_i + F_i \sin\theta_i + G_i = 0 \tag{6}$$

where:

$$\begin{aligned} &E_1 = 2L\left(y + a\right); F_1 = 2Lz \\ &G_1 = x^2 + y^2 + z^2 + a^2 + L^2 + 2ya - l^2 \\ &E_2 = -L\left(\sqrt{3}\left(x + b\right) + y + c\right);\ F_2 = 2Lz \\ &G_2 = x^2 + y^2 + z^2 + a^2 + L^2 + 2xb + 2yc - l^2 \\ &E_3 - L\left(\sqrt{3}\left(x - b\right) - y - c\right);\ F_3 = 2Lz \\ &G_2 = x^2 + y^2 + z^2 + a^2 + L^2 - 2xb + 2yc - l^2 \end{aligned} \tag{7}$$

Solve Equation (6) to obtain rotation angles θ_i. Divide Equation (6) by $\sqrt{E_i^2 + F_i^2}$:

$$\frac{E_i}{\sqrt{E_i^2 + F_i^2}} \cos\theta_i + \frac{F_i}{\sqrt{E_i^2 + F_i^2}} \sin\theta_i = \frac{G_i}{\sqrt{E_i^2 + F_i^2}} \tag{8}$$

Denote:

$$r_i = \sqrt{E_i^2 + F_i^2}$$

$$\cos\varphi_i = \frac{E_i}{\sqrt{E_i^2 + F_i^2}}; \sin\varphi_i = \frac{F_i}{\sqrt{E_i^2 + F_i^2}}$$

Substituting into Equation (8):

$$\cos\left(\theta_i - \varphi_i\right) = \frac{G_i}{r_i} \tag{9}$$

If $|G_i/r_i| \le 1$, equation (8) has two solutions:

$$\theta_i = \pm \arccos\left(\frac{G_i}{r_i}\right) + \varphi_i \tag{10}$$

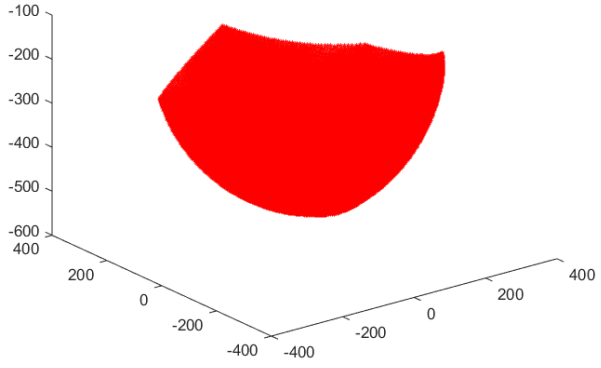
The value of θ_i must satisfy the condition:

$$-90° \le \theta_i \le 90° \tag{11}$$

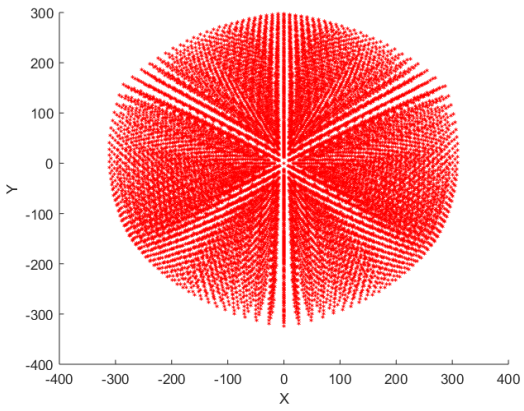
Table 1 shows the mechanism parameters value of the delta robot systems. From these values and the forward kinetic solution, the robot's workspace can be analyzed. Each joint's angle is divided into 24 equal intervals (25 values); three angles will create 25^3 = 15625 different postures of the robot. A program is written in MATLAB to calculate the 3D position of the mobile platform from these angles. Each posture corresponds with a 3D point of the center of the platform. These points are drawn in a 3D graph to illustrate the robot's workspace. The result of the robot workspace is shown in Figure 2.

Table 1. Robot parameters

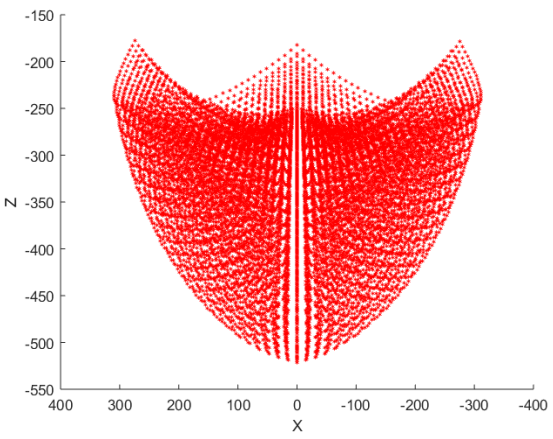
No.	Parameter	Value
1	The length of the upper arm (L)	380mm
2	The length of the lower arm (l)	150mm
3	The radius of the fixed base r_b	94mm
4	The radius of the mobile platform r_e	50mm
5	Joint angle limit	$\left[-\frac{2\pi}{9}, \frac{4\pi}{9}\right]$



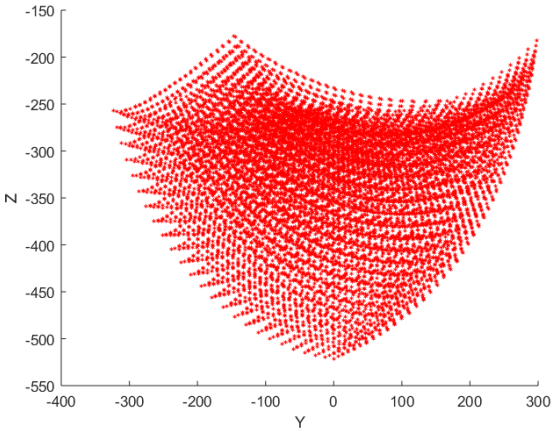
(a) 3D view



(b) XY view



(c) XZ view



(d) YZ view

Figure 2. The robot workspace

3. MECHANICAL DESIGN AND FABRICATION

This section presents the design and fabrication of the 3D mechanical model of the delta robot arm. The model is designed with the criterion of simplicity and ease of fabrication with available materials and can be fabri–cated with technologies such as laser cutting and 3d printing. The software used for the design is Autodesk Inventor 2021, with an education license. The complete model is shown in Figure 3.

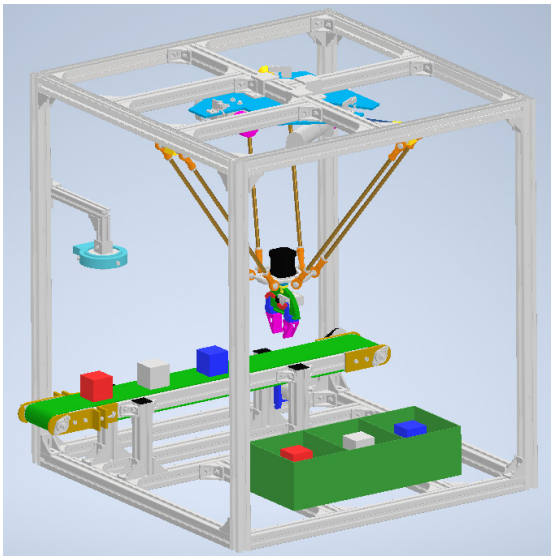


Figure 3. The design of the delta robot arm on the Autodesk Inventor software

The robot model is attached to a frame constructed from 3030 aluminum extrusion profiles. The size of the aluminum frame is 700x700x800mm. Aluminum extru–sion profiles are very common mechanical parts. We can easily connect them to build a frame using T-sliding nuts and aluminum corner brackets. These parts are very cheap and can be purchased easily.

The delta robot arm consists of three upper arms driven by three motors. Three motors are fixed on a top platform. An upper arm is connected to two lower arms by two RRR joints. The lower arms are also connected to the mobile platform by RRR joints. In our design, we use stepper motors to drive the robot. Although compared with AC servo motors, stepper motors have the disad–vantage of easy lose steps when overloaded and torque drop sharply at high speed, they are still widely used in automation machines due to their advantages such as ease of control, accuracy, and low cost. In our design, stepper motors can still be used because the robot is designed to pick up light objects, and the travel speed is low. The stepper motors used in our design are the Nema 23 motors with a flange size of 57mm. The motors can supply a maximum holding torque of 2.2Nm. This torque can be maintained if the rotational speed is less than 300rpm. The motors are mounted to Alloy Steel Mounting Brackets to connect to the fixed platform. The fixed platform is designed with 4mm mounting holes for assembly with motor brackets by M4 bolts and nuts. The fixed platform is made from a 5mm aluminum alloy sheet and fabricated by laser cutting for accuracy and low cost.

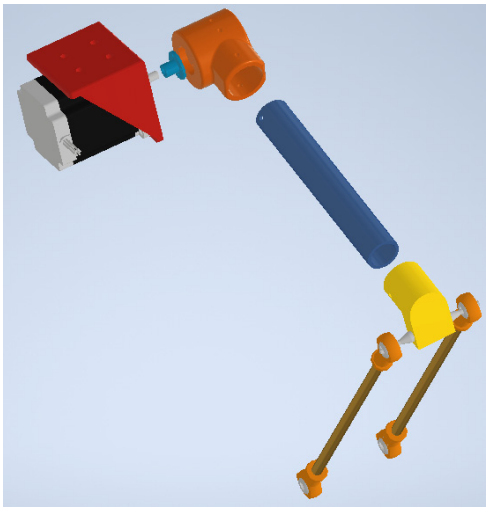


Figure 4. The struct of a robot arm

The upper arms are cut from hollow aluminum tubes with a diameter of 27mm and a thickness of 6mm. To connect the upper arms to the stepper motor, we design coupling parts and fabricate them using a 3D printing machine. The lower arms are also cut from aluminum rods with a diameter of 8mm. Each end of the rod is fitted with a ball joint to connect to the upper arm and the mobile platform. A 3D printing part is fitted at the end of the upper arm to assemble with the ball joints. The struct of a robot arm is shown in Figure 4.

Figure 5. Moving platform with gripper

The mobile platform is designed to mount the end-effector of the robot. The end-effector is an extra rotation degree of freedom driven by a stepper motor. The fourth stepper motor is mounted to the moving platform, and the motor's shaft is connected to a gripper as shown in Figure 5. The gripper is designed to can change the distance between two fingers. The distance between two fingers is adjusted by changing the rotation angle of the stepper motor. Two slider crank mechanisms are used to transform the rotation of the MG996R servo motor to the translation of two fingers.

4. ELECTRONIC SYSTEM DESIGN

Figure 6 shows the block diagram of the electronic system. The Arduino Uno R3 board is used as the robot controller to control the robot's motion and read the signal from limit switches. The personal computer is used to run Python code for the image processing algo–rithm to calculate the position and size of the objects. The camera takes the image of the objects in the work–space and transfers it to the computer through a USB port. The image is processed, and the extracted infor–mation is transferred to the Arduino board through another USB port.

The Arduino Uno R3 board is a very common embedded microcontroller board. It has 14 digital input /output pins and 6 analog input pins to connect with other devices. In this project, the Arduino board sends pulse and direction signals to the stepper motor driver to control the rotation angle of the stepper motor and read the signal from the limit switches to move the arms to the home position. The Arduino board controls the step–per motors based on the rotation angle values it receives from the computer through the USB connection.

The stepper motor driver converts the pulse and direction signal received from the Arduino to the current signal for stepper motors. The driver used in this project is the Micro step driver that uses the SI09AFTG IC. The maximum voltage that can provide for the driver is 40VDC, and the maximum current output from the driver is 3.5 A. There are many step resolutions: ½, ¼, 1/8, 1/16, and 1/32. There are 8 selectable output current settings of 0.5 – 3.5A. The resolutions and output current can be chosen via DIP switches. The Pulse input frequency is up to 60 kHz.

NEMA 23 Stepping Motor is a bipolar stepper motor with 4 control wires. It offers 24 kgf.cm of holding torque and 200 steps per revolution (full step angle of 1.8 degrees). The maximum current at each phase is 2A. Stepper motors are great motors for position control. In this work, they are used to provide rotation motion for the upper arms.

The MG996R servo motor is used to control the distance between the two fingers of the gripper.

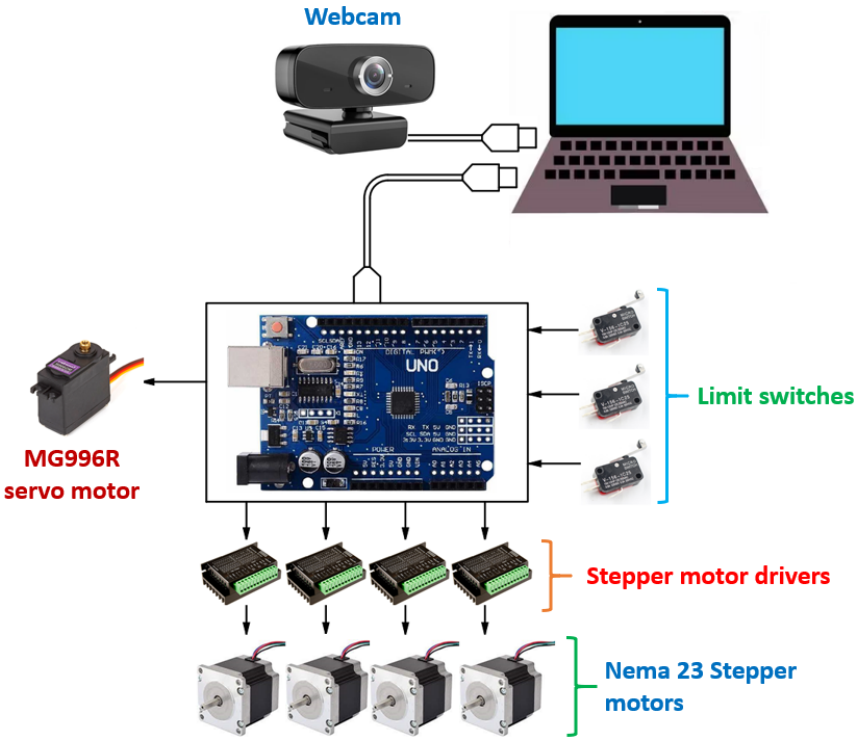


Figure 6. The electronic system

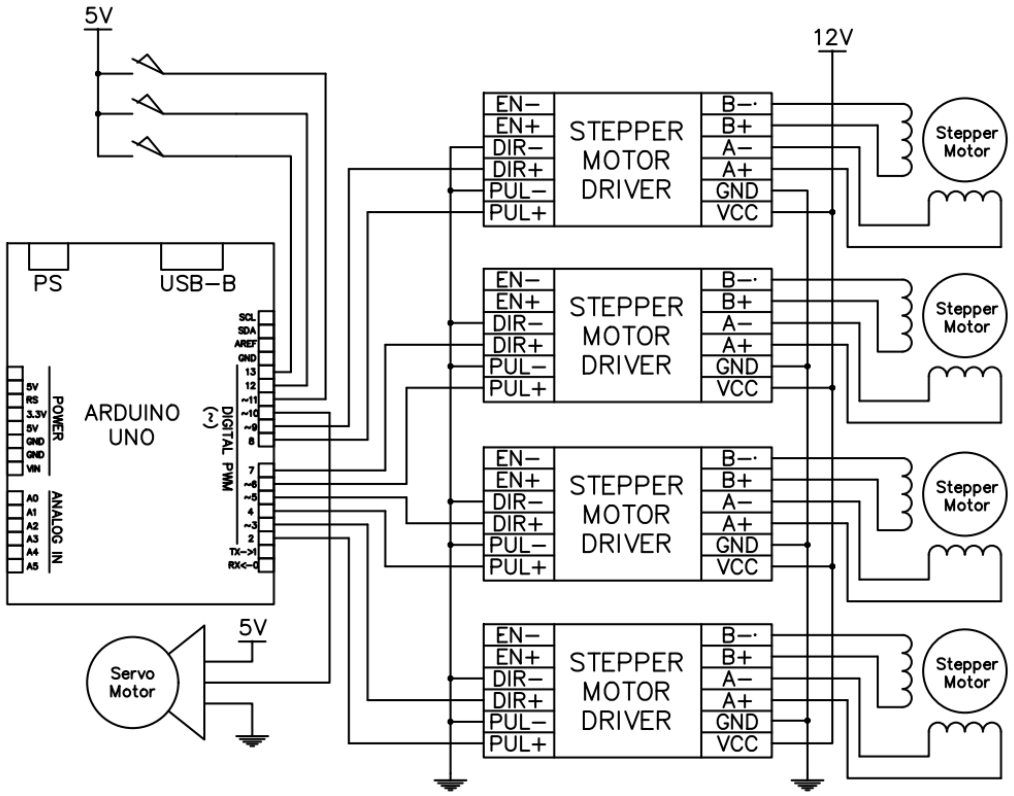


Figure 7. Wiring diagram

Limit switches: are used to determine the home position.

The wiring circuit of the robot system is shown in Figure 7. It should be noted that the motor must be properly wired with the drivers. The motor's control–wires A+, A-, B+, and B- are black, green, red, and blue, respectively.The power supply for the drivers is 12VDC. Each stepper motor consumes a maximum cur–rent of 2A, so the power supply must have a maximum output current of 8A.

5. COMPUTER VISION SYSTEM

In this section, the computer system is presented to pro–vide information for the robot to grab objects. First, an image processing algorithm processes the image to extract the objects from the background and determine the centroid as well as the size of the object. Then, the calibration method is used to transform the coordinates and distance in the image into the 3D coordinates and distance in the workspace.

5.1 Image processing algorithm

The algorithm is developed based on the contouring technique. The contour of the target object is extracted, and different features of contours, like area, perimeter, centroid, bounding box, etc., are calculated. Before performing contour finding, the target object must be separated from the background by binarization. With the stability of lighting conditions, absolute thresholding is applied to binarize the image as follows:

$$dst(u,v)=\begin{cases}1 & if\ src(u,v)>threshold\\ 0 & otherwise\end{cases} \tag{12}$$

where $src(u,v)$ is a pixel in the source image, dst(u,v) is a pixel in the destination image. The pixel in the image that is larger than a predefined threshold is set to 1, and the other pixel is set to 0. Due to the noise, the white region of the object in the binary image has tiny black holes. To overcome this issue, the closing operation is performed. The closing operation is formed by first ap–plying dilation and then applying erosion. The dilation formula is:

$$\begin{aligned}dst(u,v)&=dilate\left(src(u,v)\right)\\&=\max_{(u',v')}src(u+u',v+v')\end{aligned} \tag{13}$$

And the erosion is:

$$\begin{aligned}dst(u,v)&=erode\left(src(u,v)\right)\\&=\max_{(u',v')}src(u+u',v+v')\end{aligned} \tag{14}$$

After segmenting the object, the outer contour of the object is extracted. Contouring is a computer vision technique that joins all continuous points of the same color and intensity. Many contour features can be ext–racted to calculate the object's centroid, orientation, and size. There are many techniques to extract the contour. In this project, we use the Canny edge detection algorithm.

Because the target object has an arbitrary shape, we fit the object by a rotated rectangle for grabbing. A bounding rectangle is created with a minimum area enclosing the contour, so it considers the rotation also, as shown in Figure 8. The rectangle is structured by centroid (u, v), width, height, and angle of rotation. The OpenCV library has defined the function to calculate the minimum-area bounding rectangle for a specified point set. The function used is **cv.minAreaRect()**.

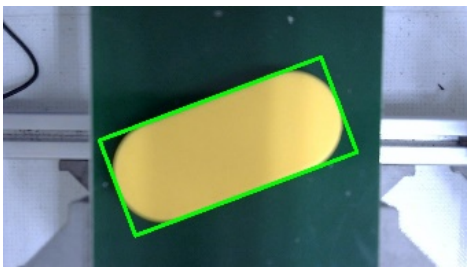


Figure 8. The rotated rectangle fits the objects

5.2 Camera calibration and 3D pose estimation

The image processing returns the bounding rectangle of the object in the image. This rectangle needs to trans–form into a 3D environment for robot grabbing. The robot will move to the centroid of the rectangle, and the gripper is opened at a distance equal to the length of the objects. The camera is calibrated to obtain the intrinsic and extrinsic parameters to perform this transformation. The 3D poses and size of the object are calculated using these parameters.

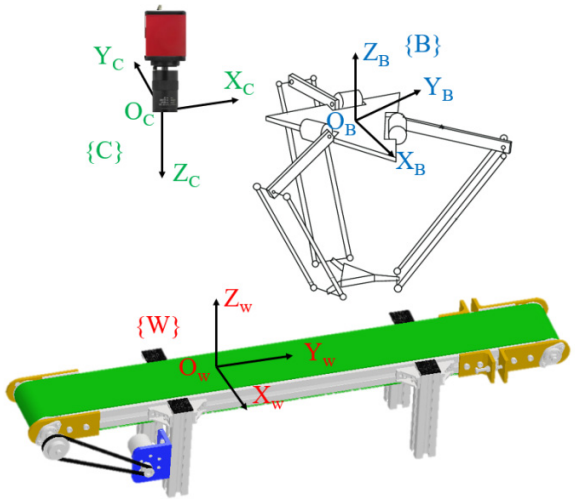


Figure 9. The coordinate frames' relationship

The coordinate frames illustrated in Figure 9 consist of the robot base coordinate system {B}, the camera coordinate system {C}, and the world coordinate system {W}. The robot base frame and the world frame have the same axes direction. So, we can locate it in a fixed position. After that, move the robot's end-effector to the origin of the world coordinate frame. The pose of the end-effector related to the robot base frame can be calculated from the rotation angles of the motors by using the forward kinematic equation. So, the position of the world frame related to the robot base frame is determined. The next task is to determine the relation of the camera frame to the world frame.

The calibration is conducted by using Calibration Toolbox in Matlab. We need to capture about 10-20 images of a checkerboard in which the first checkerbo–ard image is located at the origin of the world coordi–nate frame. Import the images to the calibration session, enter the size of the checkered square in the Image and Pattern Properties dialog box, then click OK, as shown in Figure 10. Matlab will auto-detect the checkerboards in the image. Click the calibration button to start the calibration process. After completing, Matlab will return the intrinsic matrix of the camera and the rotation matrix and translation vector of the coordinates attached to the checkerboards with the camera frame. Because the first checkboard is located at the origin of the world coordinate frame, we have the rotation matrix and translation vector representing the relationship between the camera frame and the world coordinate frame.

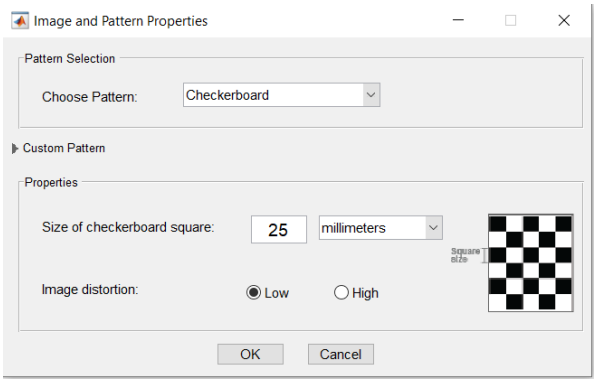


Figure 10. Image and Pattern Properties dialog box

CP denotes the centroid of the bounding rectangle in the camera frame, and in the world, the frame is denoted by ${}^{W}P$; we have the relationship:

$$ {}^{W}P = {}^{W}_{C}T\,{}^{C}P \tag{15}$$

where ${}^{B}_{W}T$ is the transformation matrix:

$$ {}^{W}_{C}T = \begin{bmatrix} R_C & T_C \\ 0 & 1 \end{bmatrix} \tag{16}$$

R_C is the rotation matrix, and T_C is the translation vector from the camera frame to the world frame. In this project, the camera's optical axis is vertical, or the camera is parallel to the plane of the workspace. So, the R_C represents the rotation around the Z-axis, and the third element of T_C is the distance from the camera to the plane of the workspace.

Also, using the translation matrix ${}^{B}_{W}T$ from the world frame to the robot base frame, the centroid of the bounding rectangle in the robot base frame (denoted by ${}^{B}P$) is determined by:

$$ {}^{B}P = {}^{B}_{W}T\,{}^{W}P = \begin{bmatrix} R_W & T_W \\ 0 & 1 \end{bmatrix} {}^{W}P \tag{17}$$

where R_W is the rotation matrix, and T_W is the translation vector from the world frame to the camera frame. From Equations (15), (16) and Equation (17) we have:

$$ {}^{B}P = \begin{bmatrix} R_W & T_W \\ 0 & 1 \end{bmatrix} \begin{bmatrix} R_C & T_C \\ 0 & 1 \end{bmatrix} {}^{C}P \tag{18}$$

The coordinate ${}^{C}P = \{X_C, Y_C, Z_C\}$ is calculated from its pixel in the image. From the pinhole camera model, we have:

$$ \frac{X_C}{Z_C} = \frac{\overline{u} - u_0}{f} \qquad \frac{Y_C}{Z_C} = \frac{\overline{v} - v_0}{f} \tag{19}$$

where $\{\overline{u},\overline{v}\}$ is the pixel coordinate of the centroid of the bounding rectangle, $\{u_0,v_0\}$ is the centroid of the image, and f is the camera's focal length. The Z_C is equal to the distance from the camera to the object; it is determined by subtracting the distance from the camera to the plane of the workspace from the height of the target object. All parameters in Equation (19) have been determined; the coordinate ${}^{C}P$ is calculated and trans–formed into the robot base frame by Equation (18).

Using Equation (19), we can also covert the distance in the image to the 3D environment. Consider two points $P_1 = \{X_1,Y_1,Z_1\}$ and $P_2 = \{X_2,Y_2,Z_2\}$ in the camera frame, the pixel coordinates are $p_1 = \{u_1,v_1\}$ and $p_2 = \{u_2,v_2\}$, respectively. The distance between the two points is:

$$P_1P_2 = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} = \\ = \frac{Z_C}{f}\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2} = \frac{Z_C}{f}p_1p_2 \tag{20}$$

So, a distance in the image is converted to the distance in the 3D environment by multiplying it with a scale factor Z_C/f.

The orientation angle of the target object in the pixel coordinate frame also needs to convert to the 3D coordinate frame. We do not directly use the orientation in the image to send to the robot because the camera frame is rotated at an angle of α in the Z direction with the robot base frame. So, we must offset the angle in the image with the value α. Assuming that the orientation angle in the image is β, the angle of the target object in the robot base frame is:

$$\theta = \beta - 180° + \alpha \tag{21}$$

6. RESULTS AND DISCUSSION

From the design on Inventor software, the mechanical parts of the robot are fabricated and assembled with other parts to form a complete system. The total cost to build our robot system is only 498.9 USD, much cheaper than an industrial robot arm with a price of several thousand USD. The practice robot is shown in Figure 11.

The computer vision system is used to detect three objects with different shapes and sizes, as shown in Figure 12. The object images are converted from the RGB space to the binary image using the proposed algorithm, and the results are shown in Figure 13. In the binary image, the objects are distinguished from the background; they are white blobs on a black background. Some noisy objects with small sizes are in the resulting images, so an area threshold is applied to remove the noisy white blobs.

The contouring technique approximates objects by the rotation rectangles, as shown in Figure 14. The rectangles provide the coordinates of the centroid and the width of the objects in the 2D image. To convert these parameters to the 3D space, using the proposed method in Section 5.2.

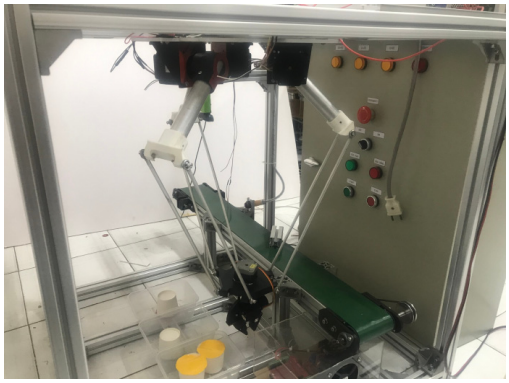


Figure 11. The practice robot

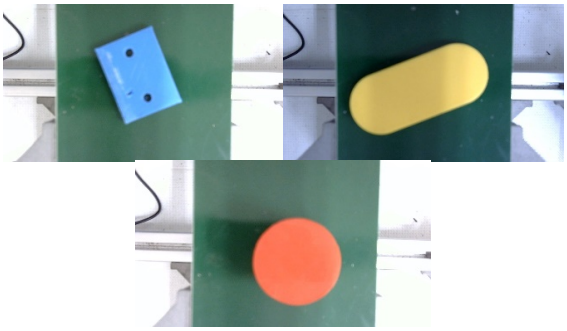


Figure 12. The objects for the experiment test

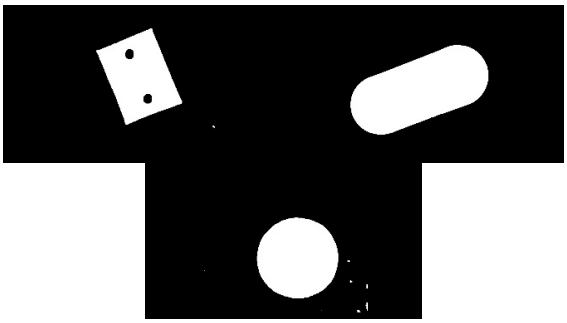


Figure 13. The objects for the experiment test

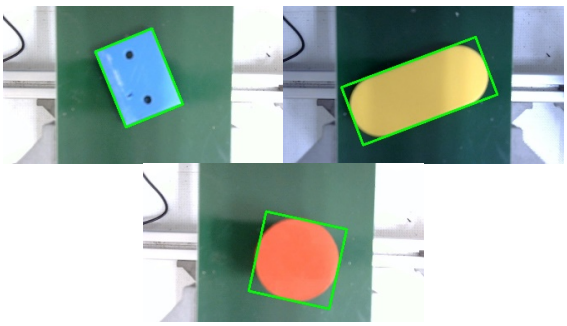


Figure 14. The objects for the experiment test

Table 2. 3D coordinate calculation

	Object 1	Object 2	Object 3
Image coordinate (u, v,β)	(619.59, 327.7, 67.79^0)	(621.42, 387.28, 69.44^0)	(704.94, 439.98, 12.52^0)
camera pose (X, Y,θ)	(-15.79mm, -24.86mm, 67.79^0)	(-14.66mm, 11.76mm, 69.44^0)	(37.00mm, 44.20mm, 12.52^0)
Robot pose (X, Y, θ)	(109.86mm, -135.79mm,-22.21^0)	(73.24mm, -134.66mm, -20.56^0)	(40.8mm, -83.0mm,-77.48^0)
Actual pose (X, Y, θ)	(110mm, -135mm, -20^0)	(72mm, -136mm, -20^0)	(42mm, -85.0mm, -75^0)
X-axis error	0.14mm	1.24mm	1.2mm
Y-axis error	0.79mm	1.34mm	2mm
Angle error	2.21^0	0.56^0	2.48^0

After performing the calibration process, the relati–onship between the coordinate systems is obtained as follows:

$$ {}^{W}_{C}T = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 305.7 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ {}^{B}_{W}T = \begin{bmatrix} 1 & 0 & 0 & 85 \\ 0 & 1 & 0 & -120 \\ 0 & 0 & 1 & -500 \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

The intrinsic matrix of the camera is:

$$ K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 494.21 & 0 & 645.12 \\ 0 & 496.73 & 368.16 \\ 0 & 0 & 1 \end{bmatrix} $$

The results for calculating the 3D coordinates of three objects are shown in Table 2. The maximum 3D coordinate error for detecting three objects is smaller than 2mm; the orientation angle error is smaller than 2.5 degrees. Due to using a low-cost camera, the result is quite large. However, it is enough for typical pick-and-place applications. Table 3 shows the results for size estimation. The size of the objects is calculated with an error of less than 1mm. The 3D coordinates are trans–formed into the rotation angle of the motors to control the robot, and the gripper is open with a distance equal to the size of objects to grasp objects. The robot needs about 1.4s to move from the home position to the object.

7. CONCLUSION

This paper has developed a low-cost delta robot arm for a pick-and-place application with the support of the vision system. The robot is designed and built for less than $500. Stepper motors and the Arduino board are used for developing the electronic system. The 3D prin–ter and laser cutting machine are the mechanical parts. Some other parts can be easily bought at a low cost.

The vision system is built to help the robot detect objects and calculate the 3D coordinates and size of the objects. Thanks to that, the robot can pick up objects of different sizes and be placed in any workspace. Expe–rimental results show that the visual system can rela–tively accurately estimate the position and size of the objects.

For future work, to increase the robot's flexibility, we will design an interface app to control the robot. The app is designed to the user can easily control the robot by changing the joint angles or position in the 3D coordinates. The user can write a program on the app and send the command to the robot.

Table 3. Estimate the size of objects

	Object 1	Object 2	Object 3
Image distance	372.11	280.07	373.38
3D distance	229.33mm	210.53mm	280.68mm
Actual distance	230.21mm	209.81mm	280.24mm
error	0.88mm	0.72mm	0.44mm

ACKNOWLEDGMENT

We acknowledge Ho Chi Minh, the City University of Technology (HCMUT), VNU-HCM for supporting this study.

REFERENCES

[1] Cong, V.D.: Industrial Robot Arm Controller Based on Programmable System-on-Chip Device, FME Transactions, Vol. 49, No. 4, pp. 1025-1034, 2021.

[2] Ali, H.M., Hashim, Y. and Al-Sakkal, G.A.: Design and implementation of Arduino based robotic arm, International Journal of Electrical and Computer Engineering (IJECE), Vol. 12, No. 2, pp. 1411-1418, 2021.

[3] Chen, C.S., S.K., Lai, C.C. and Lin, C.T.: Sequential motion primitives recognition of robotic arm task via human demonstration using hierarchical BiLSTM classifier, IEEE Robotics and Automation Letters, Vol. 6, No. 2, pp. 502–509, Apr. 2021.

[4] Song, R., Li, F., Fu, T. and Zhao, J.: A Robotic Automatic Assembly System Based on Vision, Applied Sciences, Vol. 10, No. 3, 2020.

[5] Caruana, L. and Francalanza, E.: Safety 4.0 for Collaborative Robotics in the Factories of the Future, FME Transactions, Vol. 49, No. 4, pp. 842-850, 2021.

[6] Weckenborg, C., Kieckhäfer, K., Müller, C. et al.: Balancing of assembly lines with collaborative robots, Business Research, Vol. 13, pp. 93–132, 2020.

[7] Wei, Q., Yang, C., Fan, W. and Zhao, Y.: Design of demonstration-driven assembling manipulator, Applied Sciences, Vol. 8, No. 5, pp. 797, May 2018.

[8] Wu, X., Ling, X. and Liu, J.: Location recognition algorithm for vision-based industrial sorting robot via deep learning, International Journal of Pattern Recognition and Artificial Intelligence, Vol. 33, No. 07, Jun. 2019.

[9] Wu, M., Mei, J., Zhao, Y. and Niu, W.: Vibration reduction of delta robot based on trajectory planning, Mechanism and Machine Theory, vol. 153, 2020.

[10] Deabs, A., Gomaa, F.R. and Khader, K.: Parallel Robot - Review Article, Journal of Engineering Science and Technology Review, No. 14, Vol. 6, pp.10-27, 2021.

[11] Behera, L., Rybak, L., Malyshev, D., Gaponenko, E.: Determination of Workspaces and Intersections of Robot Links in a Multi-Robotic System for Trajectory Planning. Applied Sciences, Vol. 11, No. 11, 2021.

[12] Malyshev, D. et al.: Optimal Design of a Parallel Manipulator for Aliquoting of Biomaterials Considering Workspace and Singularity Zones. Applied Sciences, Vol. 12, No. 4, 2022.

[13] Cong, V.D., Hanh, L.D., Phuong, L.H. and Duy, D.A.: Design and development of robot arm system

for classification and sorting using machine vision, FME Transactions, Vol. 50, No.1, 2022.

[14] Slavkovic, N., Zivanovic, S., Dimic, Z.: Development of the programming and simulation system of 4-axis robot with hybrid kinematic, FME Transactions, Vol. 50, No. 3, pp. 403-411, 2022.

[15] Tahmasebi, M.,Gohari, M. and Emami, A.: An Autonomous Pesticide Sprayer Robot with a Color-based Vision System, International Journal of Robotics and Control Systems, Vol.2, No. 1, pp. 115-123, 2022.

[16] Li, C.H.G. and Chang,Y.M.: Automated visual positioning and precision placement of a workpiece using deep learning, International Journal of Advanced Manufacturing, Vol. 104, pp. 4527–4538, 2019.

[17] Cong, V.D. and Hanh, L.D.: A new decoupled control law for image-based visual servoing control of robot manipulators, International Journal of Intelligent Robotics and Applications, vol. 6, pp. 576–585, 2022.

[18] Dewi, T., Mulya, Z., Risma, P. and Oktarina, Y.: BLOB analysis of an automatic vision guided system for a fruit picking and placing robot, International Journal of Computational Vision and Robotics, Vol. 11, No. 3, pp.315–327, 2021.

[19] Wang, Z., Xu, Y., Xu, G., Fu, J., Yu, J. and Gu, T.: Simulation and deep learning on point clouds for robot grasping, Assembly Automation, Vol. 41, No. 2, pp. 237-250, 2021.

[20] Hegedus, M., Gupta, K. andMehrandezh, M.: Efficiently finding poses for multiple grasp types with partial point clouds by uncoupling grasp shape and scale, Autonomous Robots, Vol. 46, No. 4, pp. 749–767, 2022.

ДИЗАЈНИРАЈТЕ ЈЕФТИНУ ДЕЛТА РОБОТСКУ РУКУ ЗА АПЛИКАЦИЈЕ ЗА ОДАБИР И ПОСТАВЉАЊЕ НА ОСНОВУ КОМПЈУТЕРСКОГ ВИДА

Л.Х. Фуонг, В.Д. Конг, Т.Т. Хиеп

У овом раду развијамо јефтину делта роботску руку за хватање предмета неспецифициране величине захваљујући систему визије. Корачни мотори се користе уместо ац серво мотора за изградњу јефтине делта роботске руке. Штавише, користимо доступне материјале и методе машинске обраде као што су ласерско сечење и 3D штампа уместо CNC глодања и стругања како бисмо смањили трошкове производње. Контролер је заснован на јефтином уграђеном контролеру - Ардуино Уно за контролу кретања робота. Систем визије је конструисан да одреди 3Д координате објеката у радном простору као и величине објеката. Хватало се отвара на удаљености од два прста која је једнака величини предмета, а робот се контролише на координате објеката да их ухвати. Апликација за подизање предмета на покретној траци је развијена за валидацију дизајна. Експериментални резултати показују да роботски систем ради исправно, роботска рука се креће глатко, а информације које је одредио систем визије имају малу грешку, осигуравајући да робот може прецизно покупити производе.