Using an HSV-based Approach for Detecting and Grasping an Object by the Industrial Manipulator System

Ha Quang Thinh Ngo
Faculty of Mechanical Engineering
Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet District 10, Ho Chi Minh
Vietnam
and
Vietnam National University Ho Chi Minh City (VNU-HCM), Linh Trung Ward, Thu Duc City, Ho Chi Minh
Vietnam

In the context of the industrialization era, robots are gradually replacing workers in some production stages. There is an irreversible trend toward incorporating image processing techniques in the realm of robot control. In recent years, vision-based techniques have achieved significant milestones. However, most of these techniques require complex setups, specialized cameras, and skilled operators for burden computation. This paper presents an efficient vision-based solution for object detection and grasping in indoor environments. The framework of the system, encompassing geometrical constraints, robot control theories, and the hardware platform, is described. The proposed method, covering calibration to visual estimation, is detailed for handling the detection and grasping task. Our approach's efficiency, feasibility, and applicability are evident from the results of both theoretical simulations and experiments.

Keywords: *Robotics, image processing, HSV color space, industrial manipulator, pick-and-place, motion control.*

1. INTRODUCTION

For Industry 4.0, robot arms have emerged as critical components in modern production lines. These pro–duction lines typically consist of multiple stations where a variety of tasks are executed, including assembly [1], inspection [2], and packaging [3]. Robot manipulators are programmable to carry out these tasks [4] auto–nomously, or they can collaborate with human workers [5].

Moreover, robots can be equipped with sensors and vision systems, empowering them to accomplish tasks like part recognition [6], quality control [7], and error detection [8]. This functionality enables them to identify and rectify errors in real-time, reducing waste and improving overall efficiency.

In manufacturing, robot manipulators play a pivotal role in automating assembly lines, effectively reducing labor costs and significantly enhancing production speed and efficiency [9].

These robots can be programmed to execute repetitive tasks, such as welding [10], painting [11], and material handling [12], which are often hazardous or monotonous for human workers. Their integration into the production process is crucial in boosting productivity and elevating the overall product quality. Furthermore, the vision-based approach employed by robot manipulators entails the utilization of cameras and image processing algorithms to offer visual feedback, enabling the robot to perceive and comprehend its surrounding environment.

In some research studies [13-17], cameras are strategically mounted on the robot or in the work area to capture images of both objects and the surrounding environment. These images are subsequently processed by algorithms that extract vital information, such as object location, shape, and orientation within the scene. The robot's control system then utilizes the extracted data to plan and execute its movements effectively. A common application of vision-based approaches lies in object recognition [18] and tracking [19]. The robot's camera captures images of objects in the environment, and sophisticated image processing algorithms identify and track these objects based on their appearance and movement patterns. Armed with this information, the robot can precisely pick up and manipulate objects with a high level of accuracy.

Another significant application of vision-based approaches lies in path planning [20] and obstacle avoidance [21]. Through the utilization of cameras and image processing algorithms, the robot can detect obstacles obstructing its path and subsequently devise alternative routes to circumvent them. This capability empowers the robot to navigate through intricate environments with ease, enabling it to execute tasks more efficiently and safely.

2. PREVIOUS WORKS

Recently, manipulators are widely used in manufacturing, assembly, and logistics industries for many tasks. These machines are designed to perform repetitive tasks accurately and efficiently, which can help to increase productivity and reduce the risk of human error [22]. One of the key features of a robot manipulator is its ability to move in multiple axes, which allows it to reach objects from various angles and positions [23].

When it comes to grasping, robot manipulators can be equipped with a diverse range of end-effectors, such as grippers [24], suction cups [25], or magnetic plates [26], depending on the specific task requirements. These

Received: May 2023, Accepted: August 2023
Correspondence to: Asso. Prof Dr Ha Quang Thinh Ngo
Ho Chi Minh City University of Technology (HCMUT)
268 Ly Thuong Kiet Street, Ho Chi Minh, Vietnam
E-mail: nhqthinh@hcmut.edu.vn
doi: 10.5937/fme2304512N

end-effectors are purposefully designed to securely grip objects of various shapes and sizes. Moreover, in certain cases, they possess the capability to detect the orientation and position of the object, thereby ensuring a successful and precise grasp.

Pick and place operations entail transferring an object from one location to another, and robot manipulators are excellently suited for this task [27]. These machines boast the capability to move in multiple axes and adjust their grippers or end-effectors as required, enabling swift and precise object transfers between locations. This particular feature proves especially advantageous in manufacturing and logistics applications [28, 29], where efficient and accurate handling of objects is of paramount importance.

Reaching is another vital aspect of robot manipulators, particularly when it comes to executing tasks in confined spaces or hard-to-reach areas [30]. The flexibility of extending their arms and adjusting their end-effectors enables robot manipulators to access such areas with ease, allowing them to accomplish tasks that would prove challenging or impossible for humans. Consequently, robot manipulators are highly advantageous not only in industrial applications but also in academic research. Table 1 classifies the different types of robotic actions, providing an overview of the state-of-the-art related studies from several years ago. This classification offers a comprehensive view of the advancements made in this field, aiding in the understanding of the evolution and capabilities of robot manipulators.

3. PRELIMINARIES

3.1 Theoretical analysis

First of all, the structure of the target manipulator is established and analyzed. There are a total of six DOFs (Degree of Freedom) in this robot platform, as Fig. 1. These axes are labeled from the base to the end-effector.

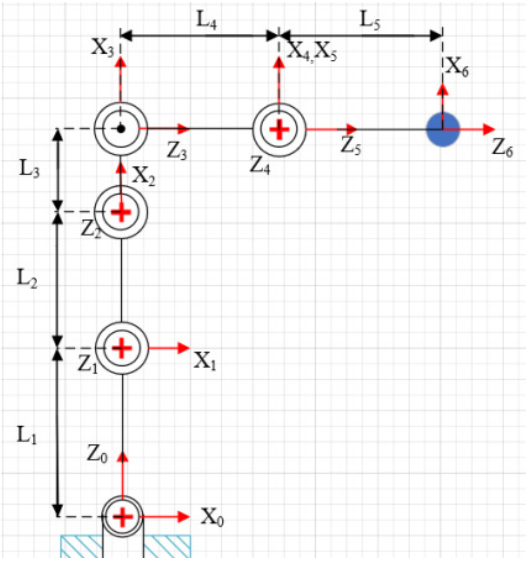

Figure 1. Dimension and theoretical structure of the proposed system

The parameters of this robot manipulator are listed below:

a: distance from Z_i axis to Z_{i+1} axis

ℓ: angular rotation from Z_i axis to Z_{i+1} axis from the view of X_{i+1} the axis

d: distance from X_i axis to X_{i+1} axis

θ: angular rotation from X_i axis to X_{i+1} axis from the view of Z_i the axis

Hence, the representation of the Denavit-Hartenberg (D-H) matrix should be launched as Table 1.

Table 1. List of parameters in the D-H matrix

Axis number	Parameter			
	a (mm)	ℓ(°)	d (mm)	θ (°)
Axis 1	0	-90	L_1	θ_1
Axis 2	L_2	0	0	θ_2-90
Axis 3	L_3	-90	0	θ_3
Axis 4	0	90	L_4	θ_4
Axis 5	0	-90	0	θ_5
Axis 6	0	0	L_5	Θ_6

From the above information in the DH matrix, the matrix transformation among links is established as follows,

- Relation between link 0 and link 1:

$$T_{01} = \begin{bmatrix} c(\theta_1) & 0 & -s(\theta_1) & 0 \\ s(\theta_1) & 0 & c(\theta_1) & 0 \\ 0 & -1 & 0 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

- Relation between link 1 and link 2:

$$T_{12} = \begin{bmatrix} s(\theta_2) & c(\theta_2) & 0 & L_2 s(\theta_2) \\ -c(\theta_2) & s(\theta_2) & 0 & -L_2 c(\theta_2) \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2)$$

- Relation between link 2 and link 3:

$$T_{23} = \begin{bmatrix} c(\theta_3) & 0 & -s(\theta_3) & L_3 c(\theta_3) \\ s(\theta_3) & 0 & c(\theta_3) & L_3 s(\theta_3) \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (3)$$

- Relation between link 3 and link 4:

$$T_{34} = \begin{bmatrix} c(\theta_4) & 0 & s(\theta_4) & 0 \\ s(\theta_4) & 0 & -c(\theta_4) & 0 \\ 0 & 1 & 0 & L_4 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

- Relation between link 4 and link 5:

$$T_{45} = \begin{bmatrix} c(\theta_5) & 0 & -s(\theta_5) & 0 \\ s(\theta_5) & 0 & c(\theta_5) & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (5)$$

- Relation between link 5 and link 6:

$$T_{56} = \begin{bmatrix} c(\theta_6) & -s(\theta_6) & 0 & 0 \\ s(\theta_6) & c(\theta_6) & 0 & 0 \\ 0 & 0 & 1 & L_5 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (6)$$

3.2 Forward kinematics

Each movement in an axis could produce different loca–tions of the end-effector. In the view of robot control, the computation of forward kinematics must be pre–sented primarily,

$$T_{06} = T_{01} * T_{12} * T_{23} * T_{34} * T_{45} * T_{56} \quad (7)$$

$$T_{06} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & x \\ A_{21} & A_{22} & A_{23} & y \\ A_{31} & A_{32} & A_{33} & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (8)$$

where these vectors (A_{11}; A_{21}; A_{31}), (A_{12}; A_{22}; A_{32}), (A_{31}; A_{32}; A_{33}) are the unit vectors of three axes attached to the end-effector. They represent three directions of the end-effector; the other vectors such (*x*; *y*; z) indicate the location of the end-effector compared to the origin.

Then,

$$\begin{aligned} A_{11} &= s(\theta_6)*(c(\theta_4)*s(\theta_1)-s(\theta_4)*(c(\theta_1) \\ &*c(\theta_2)*s(\theta_3)-c(\theta_1)*s(\theta_2)*c(\theta_3)))-c(\theta_6) \\ &*(c(\theta_5)*(s(\theta_1)*s(\theta_4)+c(\theta_4)*(c(\theta_1)*c(\theta_2) \\ &*s(\theta_3)+s(\theta_1)*s(\theta_2)*c(\theta_3)))-s(\theta_5)*(c(\theta_1) \\ &*s(\theta_2)*s(\theta_3)-c(\theta_1)*c(\theta_3)*c(\theta_2))) \end{aligned} \quad (9)$$

$$\begin{aligned} A_{12} &= c(\theta_6)*(s(\theta_1)*c(\theta_4)+s(\theta_4)*(c(\theta_1) \\ &*c(\theta_2)*s(\theta_3)+c(\theta_1)*s(\theta_2)*c(\theta_3))+s(\theta_5) \\ &*(c(\theta_1)*c(\theta_2)*s(\theta_3)+c(\theta_1)*c(\theta_3)+s(\theta_2)))) \\ &-s(\theta_6)*(c(\theta_4)*s(\theta_1)-s(\theta_4)*(c(\theta_1)*s(\theta_2) \\ &*s(\theta_3)-c(\theta_1)*c(\theta_2)*c(\theta_3))) \end{aligned} \quad (10)$$

$$\begin{aligned} A_{13} &= -c(\theta_5)*(c(\theta_1)*s(\theta_2)*s(\theta_3)-c(\theta_1) \\ &*c(\theta_3)*c(\theta_2))-s(\theta_5)*(s(\theta_1)+s(\theta_4)+c(\theta_4) \\ &*\left(c(\theta_1)*c(\theta_2)*s(\theta_3)+c(\theta_1)*s(\theta_2)+c(\theta_3)\right) \end{aligned} \quad (11)$$

$$\begin{aligned} A_{21} &= s(\theta_6)*(c(\theta_1)*c(\theta_4)+s(\theta_4)*(s(\theta_1) \\ &*s(\theta_2)*s(\theta_3)-c(\theta_2)*c(\theta_3)*s(\theta_1)))+c(\theta_6) \\ &*(c(\theta_5)*(c(\theta_1)*s(\theta_4)-c(\theta_4)*(s(\theta_1)*s(\theta_2) \\ &*s(\theta_3)-c(\theta_2)*c(\theta_3)*s(\theta_1)))-s(\theta_5)*(c(\theta_2) \\ &*s(\theta_1)*s(\theta_3)+c(\theta_3)*s(\theta_1)*s(\theta_2))) \end{aligned} \quad (12)$$

$$\begin{aligned} A_{22} &= s(\theta_6)*(c(\theta_1)*c(\theta_4)+s(\theta_4)*(s(\theta_1) \\ &*s(\theta_2)*s(\theta_3)-c(\theta_2)*c(\theta_3)*s(\theta_1)))+c(\theta_6) \\ &*(c(\theta_5)*(c(\theta_1)*s(\theta_4)-c(\theta_4)*(s(\theta_1)*s(\theta_2) \\ &*s(\theta_3)-c(\theta_2)*c(\theta_3)*s(\theta_1)))-s(\theta_5)*(c(\theta_2) \\ &*s(\theta_1)*s(\theta_3)+c(\theta_3)*s(\theta_1)*s(\theta_2))) \end{aligned} \quad (13)$$

$$\begin{aligned} A_{23} &= s(\theta_5)*(c(\theta_1)*s(\theta_4)-c(\theta_4)*(s(\theta_1) \\ &*s(\theta_2)*s(\theta_3)-c(\theta_2)*c(\theta_3)*s(\theta_1)))+c(\theta_5). \\ &*(c(\theta_2)*s(\theta_1)*s(\theta_3)+c(\theta_3)*s(\theta_1)*s(\theta_2)) \end{aligned} \quad (14)$$

$$\begin{aligned} A_{31} &= -s(\theta_2+\theta_3)*s(\theta_5)*c(\theta_6)-c(\theta_4) \\ &*(c(\theta_2+\theta_3)*s(\theta_5)*c(\theta_6)-c(\theta_2+\theta_3) \\ &*s(\theta_4)*s(\theta_6)) \end{aligned} \quad (15)$$

$$\begin{aligned} A_{32} &= s(\theta_6)*(s(\theta_2+\theta_3)*s(\theta_5)-c(\theta_2+\theta_3) \\ &*c(\theta_4)*c(\theta_5))-c(\theta_2+\theta_3)*c(\theta_6)*s(\theta_4) \end{aligned} \quad (16)$$

$$\begin{aligned} A_{33} &= -s(\theta_2+\theta_3)*c(\theta_5)-c(\theta_2+\theta_3) \\ &*c(\theta_4)*s(\theta_5) \end{aligned} \quad (17)$$

$$\begin{aligned} x &= L_2*c(\theta_1)*s(\theta_2)+L_3*s(\theta_2+\theta_3) \\ &*c(\theta_1)+L_4*c(\theta_2+\theta_3)*c(\theta_1)+L_5 \\ &*s(\theta_2+\theta_3)*c(\theta_1)*c(\theta_5)-L_5*s(\theta_1)*s(\theta_4) \\ &*s(\theta_5)-L_5*s(\theta_2+\theta_3)*c(\theta_1)*c(\theta_4)*s(\theta_5) \end{aligned} \quad (18)$$

$$\begin{aligned} y &= L_2*s(\theta_1)*s(\theta_2)+L_3*s(\theta_2+\theta_3) \\ &*s(\theta_1)+L_4*c(\theta_2+\theta_3)*s(\theta_1)+L_5 \\ &*s(\theta_2+\theta_3)*s(\theta_1)*c(\theta_5)-L_5*c(\theta_1) \\ &*s(\theta_4)*s(\theta_5)-L_5*s(\theta_2+\theta_3) \\ &*s(\theta_1)*c(\theta_4)*s(\theta_5) \end{aligned} \quad (19)$$

$$\begin{aligned} z &= L_1+L_2*c(\theta_2)+L_3*c(\theta_2+\theta_3) \\ &-L_4*s(\theta_2+\theta_3)-L_5*c(\theta_2+\theta_3)*c(\theta_4) \\ &*s(\theta_5)-L_5*s(\theta_2+\theta_3)*c(\theta_5) \end{aligned} \quad (20)$$

3.3 Inverse kinematics

Due to the purpose of our work, the robotics system co–uld grasp an object in the 2D plane. Thus, the problem of inverse kinematics could be solved in four DOFs. The mathematical analysis of geometrical relations among links for our approach is described in Fig. 2.

According to the geometrical relations, the traveling distance A, B, and C could be determined as below,

$$A = \sqrt{x^2+y^2} \quad (21)$$

$$B = \left|L_5+z-L_1\right| \quad (22)$$

$$C = \sqrt{A^2+B^2} \quad (23)$$

where: L_1 = 290mm, L_1 = 260mm, L_3 = 30.5mm L_4 = 270mm, L_5 = 90mm, $L_x = \sqrt{L_3^2+L_4^2}$.

$$\theta_3 = 180^o - a\cos\left(\frac{L_2^2+L_x^2-C^2}{2\times L_2\times L_x}\right) - a\cos\left(\frac{L_3}{L_x}\right) \quad (24)$$

$$\beta = a\cos\left(\frac{L_2^2-L_x^2+C^2}{2\times L_2\times C^2}\right) \quad (25)$$

$$\alpha = a\cos\left(\frac{A}{C}\right) \quad (26)$$

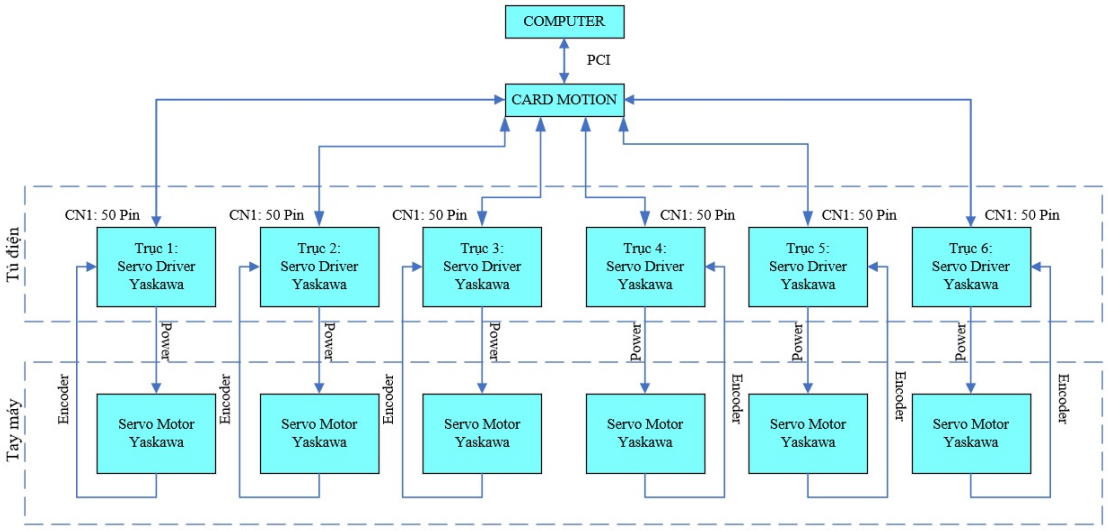

Figure 3. An overall diagram of a hardware connection.

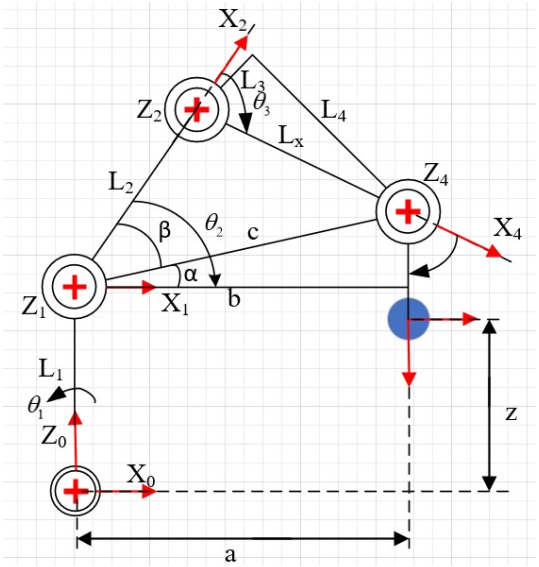

Figure 2. Geometrical analysis of the simplified model in our system.

In the next step, θ_2 must be identified. There are two cases in which $L_5 + z - L_1 \geq 0$,

$$\theta_2 = -(\beta + \alpha) + 90^0 . \tag{27}$$

Otherwise,

$$\theta_2 = -(\beta - \alpha) + 90^0 . \tag{28}$$

Then,

$$\theta_5 = -\theta_2 - \theta_3 + 90^0 . \tag{29}$$

3.4 Connection and calibration of servo system

According to the overall diagram in Fig. 3, in order to control the robot, it is necessary to have a defined communication method. In our design, the method of communication is a series of data transmissions between one master and several slaves. For each piece of information received, the system would respond due to a pre-planned action. This information is specified between our library in personal computer and the modular firmware. Since all of the six axes of our robot are pulse-based generation, the PCI-N804 card, which is able to control up to eight motors, should be utilized. This control card is designed and manufactured for the purposes of industrial applications; hence it is entirely complete in terms of library and firmware.

To configure each servo, developers usually deploy the SigmaWin+ software supported by Yaskawa com–pany as Fig. 4. In this section, several important para-meters to construct are controlled mode configuration, electric gear configuration, pulse mode configuration, and encoder output configuration.

- Control mode configuration: initially, the driver should be reset to the default mode, then follow the instructions of the manual to perform the configuration to suit the intended use. For the robotics application, three modes consist of speed control, position control, and torque control.
- Electric gear configuration: servo manufac–turers often integrate an electronic gearbox into the driver so that users can easily control and fine-tune the servo to run at will with the number of pulses suitable for each type of encoder that changes on the motor.
- Pulse mode configuration: to be able to control and read the encoder value correctly, it must be configured the pulse mode in the driver to be consistent with the motion card, then the control and the returned encoder value can be accurate.
- Encoder output configuration: there are diffe–rent types of encoder output modes used in servo motors, such as quadrature, PWM (pulse width modulation), and absolute. Because our system does not have the hardware of an absolute encoder and the output signal is differential, the quadrature configuration is chosen.

When controlling a mechanical system related to an AC servo, tuning is extremely important. It supports these motors getting used to the working load (maximum load) so that the motor can work smoothly, avoiding overload. Four steps to adjust the well-configured state of the servo motor are (1) let the software determine the force to hold the engine shaft when there is no brake, (2) use the auto-tuning feature supported on the software to calculate the necessary parameters for the engine to work properly, (3)

download the calculated parameters to the driver and (4) reset the driver.

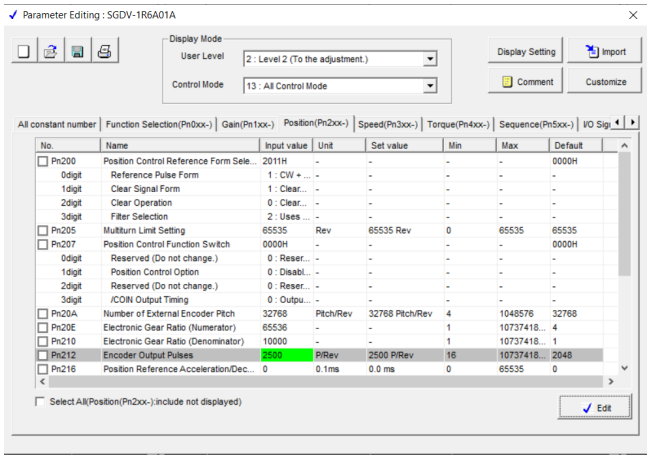

Figure 4. Configuration of each servo drives in our system.

4. THE PROPOSED APPROACH

4.1 Camera calibration

Camera calibration is important and mandatory before image processing to limit camera-induced distortion. Regularly, a 9x6 chess board, as Fig. 5, is used to conduct the calibration with the support of Matlab software. The parameters needed after the main calibration comprise the principal point and focal length. These parameters are essential for converting image coordinates to real coordinates in the following steps. Henceforth, the calibration is repeated many times to get the best results, as Fig. 6.

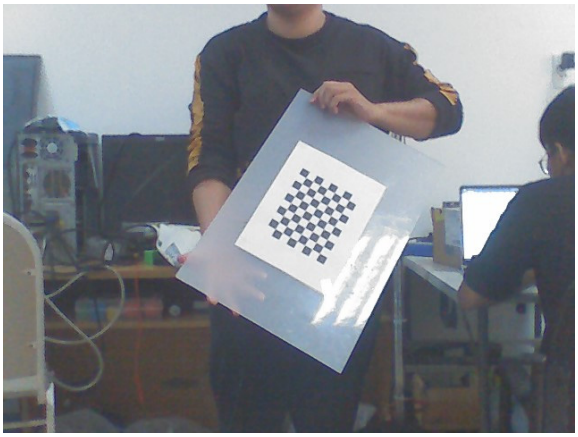

Figure 5. Configuration of each servo drives in our system.

```
Focal length (pixels):   [  966.2152 +/- 10.7587     963.7997 +/- 11.2750 ]
Principal point (pixels):[  319.5570 +/- 2.4968      234.6108 +/- 3.1996  ]
Radial distortion:       [    0.2164 +/- 0.0646        7.9240 +/- 1.7752  ]
```

Figure 6. List of the camera parameters after calibration.

4.2 Object recognition

The first step of our method is to convert the original image from the color space RGB to the color space HSV. The result of this procedure is demonstrated in Fig. 7. In that case; it should remember that the back–ground and objects must be distinguished in order to mark.

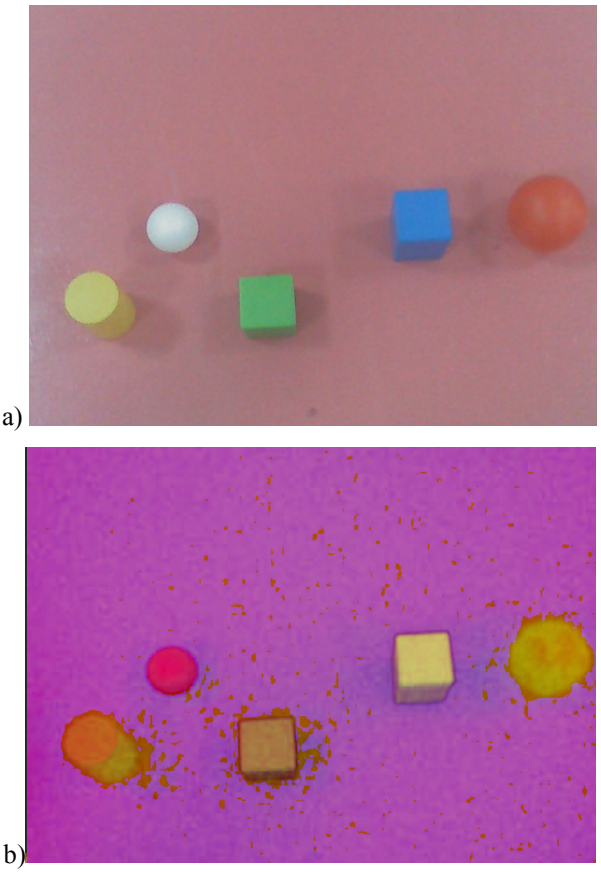

Figure 7. Object capture, (a) in the color space RGB and (b) in the color space HSV.

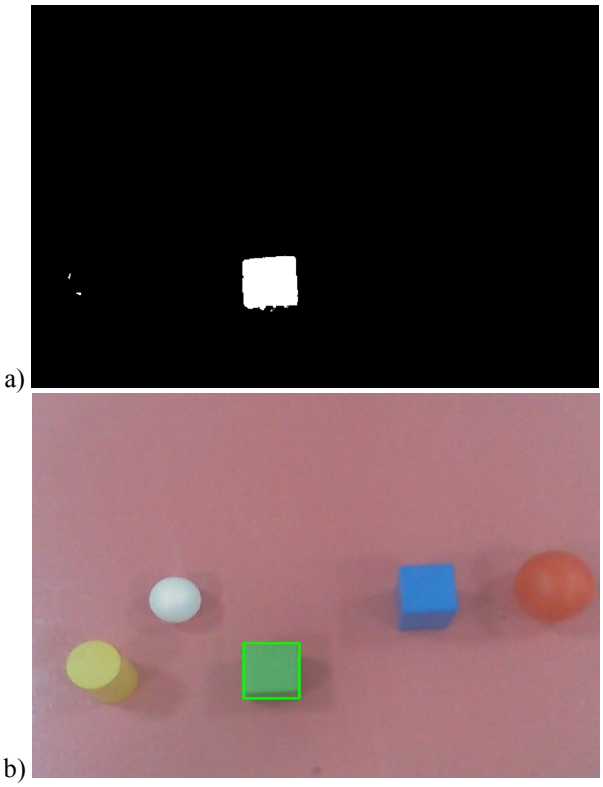

Figure 8. Result of object recognition, (a) contour detection, and (b) successful recognition.

Later, the following step is to filter the color space and find the contour to identify the object. For instance, the green block is detected. The process of filtering the Color space should be done carefully to avoid external

noise. In our design, the software GIMP that supports the HSV palette is composed of the filtering steps below,

- Alternatively, elect the maximum range and minimum range of color parameters for each object
- Compute the color value in the OpenCV library from the color space HSV
- Tune these parameters until the desired results are reached

After obtaining the desired color ranges, the contour is identified in order to obtain the shape of an object.

Then, the marking is bound to get the exact position of the color range. As Fig.8, a rectangular block is drawn. From this result, the center of the rectangle is close to the center of an object.

4.3 Coordinate transformation

After achieving the coordinates on the pixel system of each object, it converted the coordinates to the coor–dinates of the object relative to the camera.

C_x, C_y, f_x, f_y: Principal point and focal length of the camera were obtained after calibrating.

u,v: Coordinates of the object in the image coor–dinate system (pixels).

$P(X, Y, Z)$: The coordinates of the object in the real coordinate system relative to the camera.

As seen in Fig. 9, the formulas between two similar triangles to convert are illustrated. Subsequently, having the coordinates of the object relative to the camera, it determined the correlation matrix between our robot and the camera to find the coordinates of the object relative to the robot. This is the final step in the conversion process.

It is assumed that this camera, to determine the rotation matrices and translation vectors of the camera relative to the robot, is fixed on the standing mount. The conversion formula is as follows,

$$P'(X',Y',Z') = T \times P(X,Y,Z). \tag{30}$$

where,

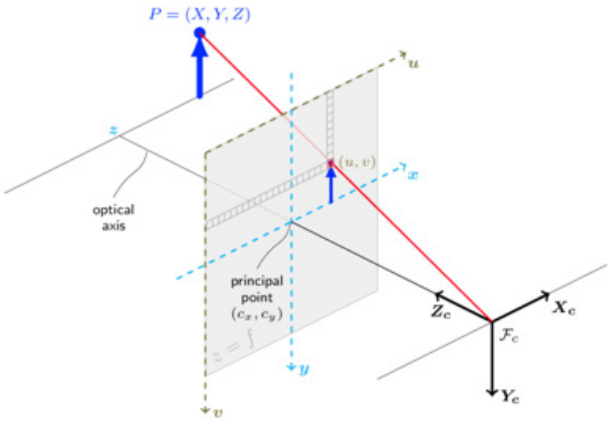

Figure 9. Coordinate transformation from pixel system to the real-world system.

$P'(X',Y',Z')$ is the coordinate of an object with respect to a robot

T is the correlation matrix from camera to robot, which is produced by rotational matrix and translational matrix between two coordinates.

Lastly, the correlation matrix is

$$T = \begin{bmatrix} 0 & 1 & 0 & 380 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 930 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{31}$$

5. RESULTS OF SIMULATION AND EXPERIMENT

4.4 Theoretical validation

Before conducting the robot experiment, several simulations to govern the accuracy of the inverse kinematics are essential to carry out as Fig. 10. By using the OpenGL-integrated MFC interface and also the Matlab robotic toolbox to simulate the trajectory of the robot, the theoretical validation is guaranteed.

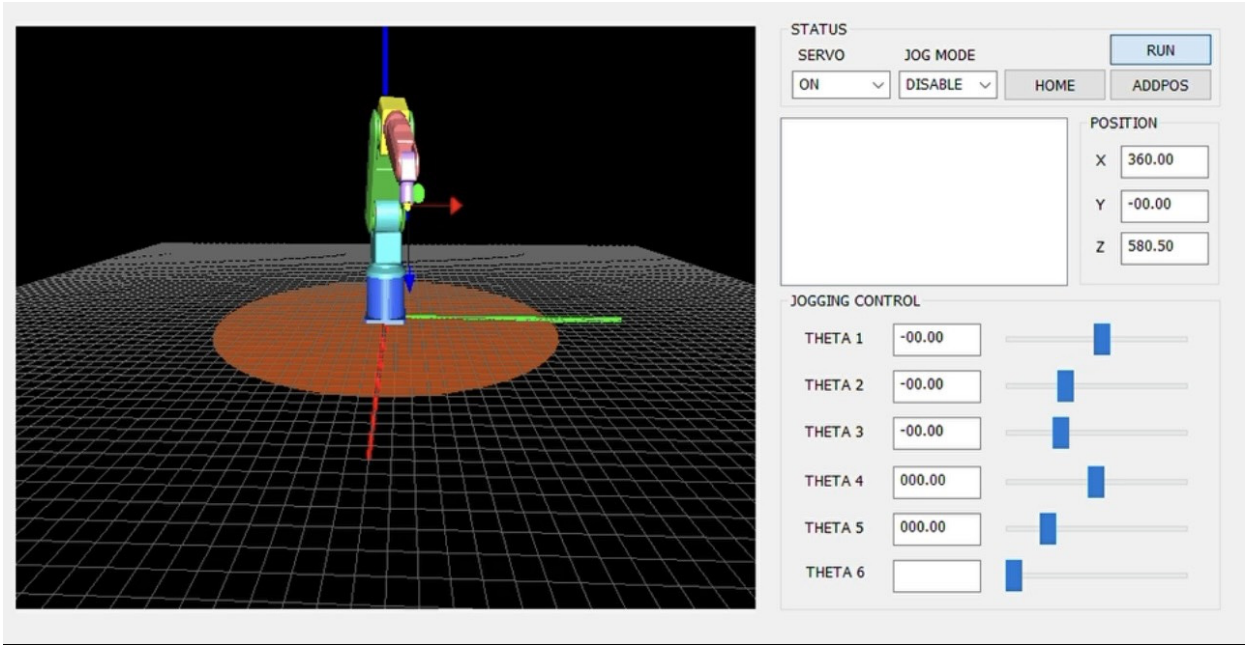

Figure 10. Software design to validate the robot control theory.

Additionally, the repetition error in this robotics system needs to be evaluated. The test is set up as follows: primarily, when the robot is in the HOME position, place the counter in the robot's workspace. Far ahead, the tooltip of the robot is moved to the position of the X point and lightly touched counter-clockwise so that the needle points to the X position other than 0 is accepted as Fig.11. At that time, return the robot to HOME and then back to the old position, do five times with the same speed. The difference in the clock between the first time and the next iteration is the Z-direction repetition error of the robot, as shown in Table 2.

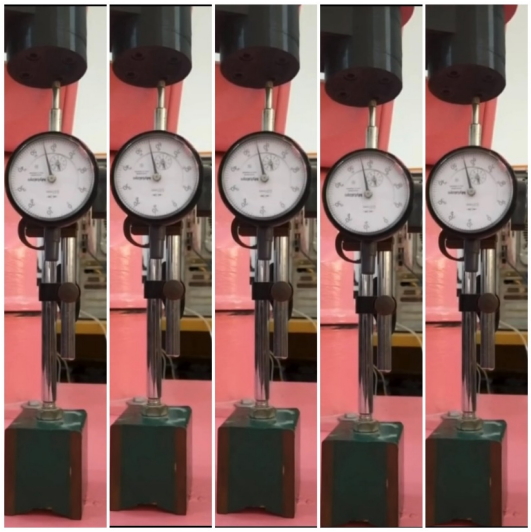

Figure 11. Result of the repetition error measurement.

Table 2. List of the repetition error measurements.

No.	The first level of speed	The second level of speed
1	Error: 0mm	Error: 0.01mm
2	Error: 0.01mm	Error: 0.02mm
3	Error: 0mm	Error: 0.02mm
4	Error: 0mm	Error: 0.01mm
5	Error: 0.01mm	Error: 0.01mm

5.1 Experimental validation

To prove the feasibility and effectiveness of the pro–posed method, some practical tests have been con–ducted. The overall system, including one NHG-D6 manipulator, one Real-sense digital camera, one per–sonal laptop, and one industrial computer (IPC), is launched as Fig 12. The NHG-D6 robot arm is an industrial manipulator with six DOFs (degree-of-freedom) which is manipulated by an industrial computer. The Real-sense camera is deployed to obtain the image data continuously. This signal is transmitted to a personal laptop programmed for image processing analysis. The result of vision-based computation could be referred for releasing the control decision. Later, IPC receives the control command and drives the robot according to the reference trajectory. The whole system is located in our laboratory, where the indoor environment ensures stable brightness. These tests are repeated several times in order to guarantee the stability of our method.

In the initial stage, the robot arm rotates to a specific position so that it does not obscure the view of the camera. After the digital camera recognizes an object, it transmits the coordinate values of that position to the laptop. At that moment, it theoretically calculates from the image coordinates to the real coordinates and uses the inverse kinematics after the calculation is completed. In this situation, the laptop would send the parameters of the robot kinematics via TCP/IP protocol communication to the drive. It is done at the same Z value without touching the objects. Then, the robot proceeded to move the objects and repeat the steps.

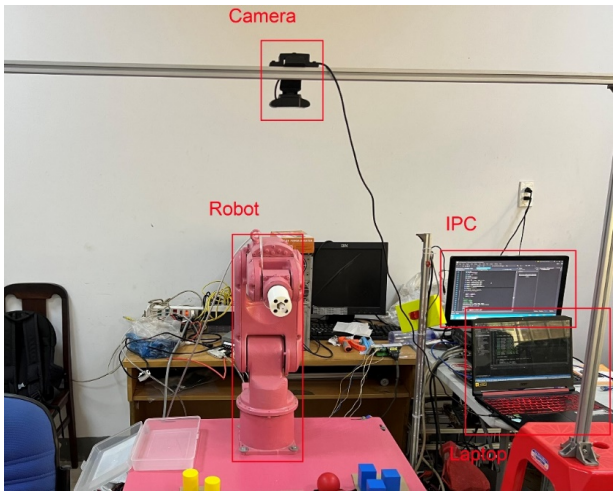

Figure 12. Hardware setup for the experimental validation.

The experimental validation using our approach is depicted in Fig. 13. After conducting a few steps, such as object recognition and location analysis, owing to the hei–ght of the samples being different, the robot arm adjusted the Z value for each object so that it could pick and place. It is first applied with separate orders of a single object for complicated circumstances. And then, the robot could work for each area with multiple objects in the same color.

6. CONCLUSION

In this study, a highly efficient solution for the object detection and grasping task was developed from scratch. By employing an HSV-based approach, the robot mani–pulator demonstrated remarkable proficiency in recogni–zing, classifying, and picking target objects without nee–ding expert intervention. The methodology begins with a comprehensive explanation of background concepts, followed by establishing the hardware platform for the robotics system designed to meet the specific requi–rements of our approach. The proposed method was ri–gorously tested and validated through simulations and real-world experiments, where theoretical computations and practical trials were conducted and demonstrated successful results. This approach proves to be highly effi–cient, feasible, and applicable in real-world manufac–turing systems, such as line production, sorting systems, and even educational settings. The outcomes of this investigation hold significant promise for advancing automation technologies and enhancing object manipulation capabilities in various industrial and educational domains.

ACKNOWLEDGMENT

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

REFERENCES

[1] Polishchuk, M., & Tkach, M. (2021). Experimental studies of robotic assembly of precision parts. *FME Transactions*, *49*(1), 44-55.

[2] Caruana, L., & Francalanza, E. (2021). Safety 4.0 for Collaborative Robotics in the Factories of the Future.

[3] Bueno, A., Godinho Filho, M., & Frank, A. G. (2020). Smart production planning and control in the Industry 4.0 context: A systematic literature review. *Computers & Industrial Engineering*, *149*, 106774.

[4] Wang, J., & Li, D. (2019). Task scheduling based on a hybrid heuristic algorithm for smart production line with fog computing. *Sensors*, *19*(5), 1023.

[5] Mujica, M., Crespo, M., Benoussaad, M., Junco, S., & Fourquet, J. Y. (2023). Robust variable admittance control for human–robot co-manipulation of objects with unknown load. *Robotics and Computer-Integrated Manufacturing*, *79*, 102408.

[6] Xiang, F., Qin, Y., Mo, K., Xia, Y., Zhu, H., Liu, F., ... & Su, H. (2020). Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11097-11107).

[7] Kazemian, A., Yuan, X., Davtalab, O., & Khoshnevis, B. (2019). Computer vision for real-time extrusion quality monitoring and control in robotic construction. *Automation in Construction, 101, 92-98*.

[8] Gai, J., Tang, L., & Steward, B. L. (2020). Automated crop plant detection based on the fusion of color and depth images for robotic weed control. *Journal of Field Robotics*, Vol *37*(1), 35-52.

[9] Ji, W., & Wang, L. (2019). Industrial robotic machining: a review. *The International Journal of Advanced Manufacturing Technology*, *103*, 1239-1255.

[10] Yang, L., Liu, Y., & Peng, J. (2020). Advances techniques of the structured light sensing in intelligent welding robots: a review. *The International Journal of Advanced Manufacturing Technology*, *110*, 1027-1046.

[11] Zhang, B., Wu, J., Wang, L., & Yu, Z. (2020). Accurate dynamic modeling and control parameters design of an industrial hybrid spray-painting robot. *Robotics and Computer-Integrated Manufacturing*, *63*, 101923.

[12] Wu, X., Chi, J., Jin, X. Z., & Deng, C. (2022). Reinforcement learning approach to the control of heavy material handling manipulators for agricultural robots. *Computers and Electrical Engineering*, *104*, 108433.

[13] Horng, J. R., Yang, S. Y., & Wang, M. S. (2020). Object localization and depth estimation for eye-in-hand manipulator using mono camera. *IEEE Access*, *8*, 121765-121779.

[14] Li, J., Ito, A., & Maeda, Y. (2019, October). A SLAM-integrated kinematic calibration method for industrial manipulators with RGB-D cameras. In *2019 19th International Conference on Control, Automation and Systems (ICCAS)* (pp. 686-689). IEEE.

[15] Liang, X., Wang, H., Liu, Y. H., You, B., Liu, Z., Jing, Z., & Chen, W. (2021). Fully uncalibrated image-based visual servoing of 2dofs planar manipulators with a fixed camera. *IEEE Transactions on Cybernetics*, *52*(10), 10895-10908.

[16] Yang, L., Yuan, C., & Lai, G. (2023). Adaptive fault-tolerant visual control of robot manipulators using an uncalibrated camera. *Nonlinear Dynamics*, *111*(4), 3379-3392.

[17] Belkhiri, A., Amouri, A., & Cherfia, A. (2023). Design of Fractional-Order PID controller for trajectory tracking control of continuum robots. *FME Transactions*, *51*(2), 243-252.

[18] Sajjan, S., Moore, M., Pan, M., Nagaraja, G., Lee, J., Zeng, A., & Song, S. (2020, May). Clear grasp: 3d shape estimation of transparent objects for manipulation. In *2020 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3634-3642). IEEE.

[19] Khan, A. H., Li, S., Chen, D., & Liao, L. (2020). Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach. *Neurocomputing*, *400*, 272-284.

[20] Baressi Šegota, S., Anđelić, N., Lorencin, I., Saga, M., & Car, Z. (2020). Path planning optimization of six-degree-of-freedom robotic manipulators using evolutionary algorithms. *International journal of advanced robotic systems*, *17*(2), 1729881420908076.

[21] Zhang, W., Cheng, H., Hao, L., Li, X., Liu, M., & Gao, X. (2021). An obstacle avoidance algorithm for robot manipulators based on decision-making force. *Robotics and Computer-Integrated Manufacturing*, *71*, 102114.

[22] Sahu, V. S. D. M., Samal, P., & Panigrahi, C. K. (2022). Modelling, and control techniques of robotic manipulators: A review. *Materials Today: Proceedings*, *56*, 2758-2766.

[23] Wang, C., Zhang, X., Zang, X., Liu, Y., Ding, G., Yin, W., & Zhao, J. (2020). Feature sensing and robotic grasping of objects with uncertain information: A review. *Sensors*, *20*(13), 3707.

[24] Samadikhoshkho, Z., Zareinia, K., & Janabi-Sharifi, F. (2019, May). A brief review on robotic grippers classifications. In *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)* (pp. 1-4). IEEE.

[25] Wan, W., Harada, K., & Kanehiro, F. (2020). Planning grasps with suction cups and parallel grippers using superimposed segmentation of object meshes. *IEEE Transactions on Robotics*, *37*(1), 166-184.

[26] Li, A., Li, H., Li, Z., Zhao, Z., Li, K., Li, M., Song, Y. (2020). Programmable droplet manipulation by a

magnetic-actuated robot. *Science advances*, *6*(7), eaay5808.

[27] Dewi, T., Nurmaini, S., Risma, P., Oktarina, Y., & Roriz, M. (2020). Inverse kinematic analysis of 4 DOF pick and place arm robot manipulator using fuzzy logic controller. *International Journal of Electrical & Computer Engineering (2088-8708)*, *10*(2).

[28] Wahrmann, D., Hildebrandt, A. C., Schuetz, C., Wittmann, R., & Rixen, D. (2019). An autonomous and flexible robotic framework for logistics applications. *Journal of Intelligent & Robotic Systems*, *93*, 419-431.

[29] Slavković, N., Živanović, S., Vorkapić, N., & Dimić, Z. (2022). Development of the programming and simulation system of 4-axis robot with hybrid kinematic. *FME Transactions*, *50*(3), 403-411.

[30] Marwan, Q. M., Chua, S. C., & Kwek, L. C. (2021). Comprehensive review on reaching and grasping of objects in robotics. *Robotica*, *39*(10), 1849-1882.

[31] Károly, A. I., & Galambos, P. (2023). Task-Specific Grasp Planning for Robotic Assembly by Fine-Tuning GQCNNs on Automatically Generated Synthetic Data. *Applied Sciences*, *13*(1), 525.

[32] Lobbezoo, A., & Kwon, H. J. (2023). Simulated and Real Robotic Reach, Grasp, and Pick-and-Place Using Combined Reinforcement Learning and Traditional Controls. *Robotics*, *12*(1), 12.

[33] Pardi, T., & Stolkin, R. (2018, November). Choosing grasps to enable collision-free post-grasp manipulations. In *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (pp. 299-305). IEEE.

[34] Tian, H., Song, K., Li, S., Ma, S., & Yan, Y. (2023). Rotation adaptive grasping estimation network oriented to unknown objects based on novel RGB-D fusion strategy. *Engineering Applications of Artificial Intelligence*, *120*, 105842.

[35] Mohammadi, S. S., Duarte, N. F., Dimou, D., Wang, Y., Taiana, M., Morerio, P., ... & Santos-Victor, J. (2023). 3DSGrasp: 3D Shape-Completion for Robotic Grasp. *arXiv preprint arXiv:2301.00866*.

[36] Kang, H. C., Han, H. N., Bae, H. C., Kim, M. G., Son, J. Y., & Kim, Y. K. (2021). HSV color-space-based automated object localization for robot grasping without prior knowledge. *Applied Sciences*, *11*(16), 7593.

[37] Kang, H., Han, H., Bae, H., Lee, E., Kim, M., Son, J., Kim, Y. K. (2019, October). HSV Color Space Based Robot Grasping for Personalized Manu–facturing Services. In *2019 International Confe–rence on Information and Communication Techno–logy Convergence (ICTC)* (pp. 1010-1012). IEEE.

[38] Wong, C. C., Tsai, C. Y., Chen, R. J., Chien, S. Y., Yang, Y. H., Wong, S. W., & Yeh, C. A. (2022). Generic Development of Bin Pick-and-Place System Based on Robot Operating System. *IEEE Access*, *10*, 65257-65270.

[39] Guerra-Zubiaga, D., Franklin, A., Escobar-Escobar, D., Lemley, T., Hariri, N., Plattel, J., & Ham, C. Computer Vision and Machine Learning to Create an Advanced Pick-and-Place Robotic Operation Using Industry 4.0 Trends. In *ASME 2022 International Mechanical Engineering Congress and Exposition*. American Society of Mechanical Engineers Digital Collection.

NOMENCLATURE

a	distance from Z_i axis to Z_{i+1} axis
ℓ	angular rotation from Z_i axis to Z_{i+1} axis from the view of X_{i+1} the axis
d	distance from X_i axis to X_{i+1} axis
θ	angular rotation from X_i axis to X_{i+1} axis from the view of Z_i the axis
T_{01}	transformation matrix from link 0 to link 1
T_{12}	transformation matrix from link 1 to link 2
T_{23}	transformation matrix from link 2 to link 3
T_{34}	transformation matrix from link 3 to link 4
T_{45}	transformation matrix from link 4 to link 5
T_{56}	transformation matrix from link 5 to link 6
T_{06}	transformation matrix from link 0 to link 6
A, B, C	traveling distance of the robot
β	angle between Z_1Z_2 and Z_1Z_4
α	angle between Z_1X_2 and Z_1X_4

ABBREVIATION

HSV	Hue Saturation Value
RL	Reinforcement Learning
RGB-D	Red Green Blue Depth
ROS	Robot Operating System
YOLO	You Only Look Once
CAD	Computer-Aided Design

КОРИШЋЕЊЕ ПРИСТУПА ЗАСНОВАНОГ НА ХСВ-У ЗА ОТКРИВАЊЕ И ХВАТАЊЕ ПРЕДМЕТА ОД СТРАНЕ СИСТЕМА ИНДУСТРИЈСКОГ МАНИПУЛАТОРА

Х.К.Т. Нго

У контексту ере индустријализације, роботи пос–тепено замењују раднике у неким фазама произ–водње. Постоји неповратан тренд ка укључивању техника обраде слике у домен контроле робота. Последњих година, технике засноване на визији постигле су значајне прекретнице. Међутим, већина ових техника захтева сложена подешавања, специ–јализоване камере и веште оператере за изра–чунавање оптерећења. Овај рад представља ефи–касно решење засновано на визији за детекцију и хватање објеката у затвореним окружењима. Описан је оквир система који обухвата геометријска огра–ничења, теорије управљања роботима и хардверску платформу. Предложени метод, који покрива кали–брацију до визуелне процене, детаљно је описан за руковање задатком детекције и хватања. Ефикас–ност, изводљивост и применљивост нашег приступа су евидентни из резултата и теоријских симулација и експеримената.