

Linear Delta Robot Controlled with PLC Based On Image Processing

Sıtkı ÖZTÜRK¹ , Fatma KUNCAN^{2,*} 

¹ Department of Electronics and Communication Engineering, Kocaeli University, Kocaeli, 41001, Turkey, **ORCID:** 0000-0003-3804-5581

² Department of Computer Engineering, Siirt University, Siirt, 56100, Turkey, **ORCID:** 0000-0003-0712-6426

Article Info

Research paper

Received : January 16, 2022

Accepted : April 08, 2022

Keywords

Linear Delta Robot
Visual Positioning
Programmable Logic Controller
Kinematic Analysis
OPC
Image Processing

Abstract

Delta Robot is taking an important place in industry 4.0. Its ability to work fast and with precision is the reason why this robot now is used worldwide in the industrial field. Delta robots are mostly used for packaging, 3D printing, pick and place, etc. This robot can be controlled using programmable logic controller (PLC) devices. In recent years, image processing methods have been widely used in many robot applications. In this study, it is aimed to ensure that the robot successfully moves to the determined points by using image processing methods. Images for this study were obtained by attaching a camera to the robot system. In this study, the aim is to set the robot to move based on the visual data that is taken by the camera. The coordinates of each object are defined after several steps in image processing. To be able to move the robot to the desired coordinates, inverse and forward kinematic analysis is proposed. The obtained values are transferred to PLC using OPC. The necessary coding in TIA Portal is done to control the motor drivers so that the three stepper motors can ensure the delta robot moves to the desired coordinate position.

1. Introduction

The Delta Robot was invented by Reymond Clavel in 1985. He came up with the brilliant idea of using parallelograms to build a parallel robot with three translational and one rotational degree of freedom. Robots are now actively working in many areas such as military security, health and transportation. These robots are evaluated in 2 classes according to their area of utilization, which are serial and parallel robots [1]. While these robots are being classified, criterias such as speed, quality control, durability and load carrying capacity are taken into consideration [2]. There are many studies on serial and parallel robots in industrial applications and academic studies. With the help of these studies, the most efficient robot type can be easily selected according to the desired application area [3]. In this study is done based on image processing linear delta robot. It is chosen in this work since its design exceeds this part because of its three translational degrees of freedom, and its acceleration. Linear delta robot is a type of parallel robot.

Parallel robots have many benefits and drawbacks. Their most significant drawback is requiring a small and complicated workspace. Because of this, they can only be practiced when the task can be accomplished in a limited area [4]. Nevertheless, the benefits points that parallel robots are the optimal decision in many circumstances. Because their motors are attached to the robot's frame, the weight and inertia of the moving parts are weak. This results in higher top speeds and accelerations.

In this work, the aim is to get the exact position coordinates of each object on the workspace of the delta robot and set the robot to move to that position. After the linear delta robot is built, there are two big steps to make this study successful. The first step is to determine the coordinate position of the objects correctly and the second step is to get a correct trajectory result for three legs of the delta robot from the kinematic analysis.

In section 3 of this study, the image processing step is explained. Logitech C310 HD webcam is used. A camera calibration needs to be done before doing image processing. This step is very crucial to get an accurate coordinate position of the object on the workspace of the delta robot.

Coordinates of objects can be obtained by a process consisting of several steps. The image taken from the

* Corresponding Author: fatmakuncan@siirt.edu.tr



camera needs to be segmented so the only desired object will be processed further. From there this image that has been segmented using HSV color space will be analyzed using BLOB analysis to get the centroid and the bounding box feature of the object in interest.

The acquired coordinates can't be directly transferred to PLC. Each leg of the delta robot needs to be converted to trajectory. This is done by doing inverse kinematic analysis. In this work, to prove that the calculation of legs trajectory from inverse kinematic is correct, forward kinematic is done [5-8].

After the trajectory of each leg is calculated in MATLAB with the help of OPC, these data are transferred to the PLC. OPC is a really important communication interface for integration between PLC and MATLAB. In this study, Simatic Net OPC was used as the OPC platform.

Since the output of PLC is 24V and the motors used in this study work with lower voltage, a motor driver circuit is needed. A4988 driver circuit was used as stepper motor driver. The resolution of the driver can be changed with MS1, MS2, and MS3 inputs. In the study, these inputs were operated in full step by making them 0 volts. After the driver was turned into operation by jumping to 0 volts with the EN input of the driver, it was moved by the direction and position pulses of the motor with the DIR and STEP inputs.

In this study, Siemens S7-1215C DC/DC/DC was used as the controller of the delta robot. In this PLC, the function blocks written for motion control were programmed to control stepper motors. After setting up all three stepper motors, PLC code was written in Ladder Logic Language. Ladder logic was chosen in this work since it's faster and easy to do debugging in this language [9-16].

2. Test Setup

Before we can move the robot to the desired coordinates provided by the image processing, we need to set it to the home position. This can be done using an inductive sensor. After it is set to its home position, we can continue to take the image and get the coordinates of objects. The coordinates that we get from image processing would be still in x , y , z dimensions. These need to be converted into the trajectory of each leg of the robot. Kinematic Analysis takes a very important place in this step. After we get the trajectory of each leg, we then calculate the velocity of each step. The velocity of each leg is saved in an array named $V1$, $V2$, $V3$. In this study, an electromagnet was used to hold the object. The working logic keeps the object when the control signal is applied to the electromagnet and leaves the object when the control signal is not applied. Object coordinate values are stored in

an array (M) and the electromagnet control signal is applied according to these coordinate data. The data that we get from image processing ($L1$, $L2$, $L3$, $V1$, $V2$, $V3$, M) is sent to PLC through OPC. Then the data is saved in the OPC_Data Block in the PLC. This data will be used to control the movement of each stepper motor. The test setup is shown in Figure 1. The flow diagram of the system is shown in Figure 2.

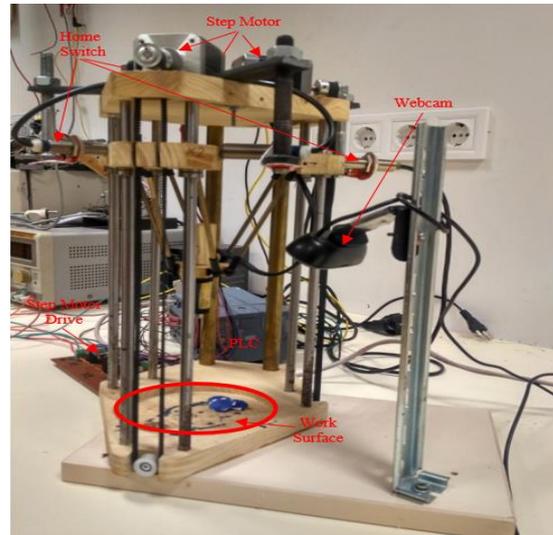


Figure 1. Test Setup of Delta Robot

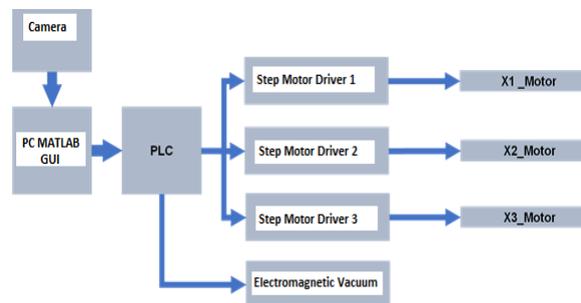


Figure 2. Flow diagram of the system

3. Image Processing Method

The camera was calibrated before starting the image processing to obtain the coordinates of the objects. Camera calibration was done to find the internal and external parameters of the camera. These parameters are then used to calculate real-world coordinates.

The algorithm for obtaining the coordinates of the location of objects is given in Figure 3. First, the photo is taken and converted to HSV. Then, the properties of the obtained two-dimensional image are extracted. By using these features, it is ensured that the desired objects and other objects around them are distinguished. This process is often called segmentation. In this study, the color thresholding method was used for segmentation. In color thresholding, the value of each layer of a given color space is used as a mask to separate objects of interest by color. In

the study, HSV color space was used to create segmentation masks. After segmentation, the HSV image was converted to a binary image. The conversion was done to start the BLOB analysis process. The purpose of the BLOB analysis is to construct pixels in close binary images as a group of pixels, which it identifies as objects. In this study, objects were found by grouping pixels with similar gray values or color values in squares as a result of the 8-pixel connection method. A filter was used to ensure that all residual noise from the segmentation process was removed. From the BLOB analysis process, the object's center of mass was obtained. These dots were still in pixel form. By following the calculations from the pinhole camera models, the coordinates of the objects in the real world were estimated.



Figure 3. Image processing algorithm

This work gives an approach to recognizing colors in a two-dimensional image using the color thresholding technique in MATLAB with the help of HSV color space to detect a selected color by a user in an image. Color detection is the fundamental step in many computer vision systems. In this work, Image Processing Toolbox in MATLAB is used for the detection of a particular color in a given image. The Image Processing toolbox provides a wide variety of referenced algorithms, methods, and applications for image processing, visualization, and segmentation. An image can be represented using many color models like Gray-scale, RGB, HSV, etc. Here, HSV color space is used to detect the colors in an image [14].

3.1. Feature Extraction of a Two-Dimensional Image

In this study, the color thresholding method was used to identify the object. The HSV color space is used to segment objects in stored images by applying the specific HSV value as a mask. The resulting mask was then converted to a binary image and enhanced by filtering. The feature extraction algorithm of the two-dimensional image is shown in Figure 4.

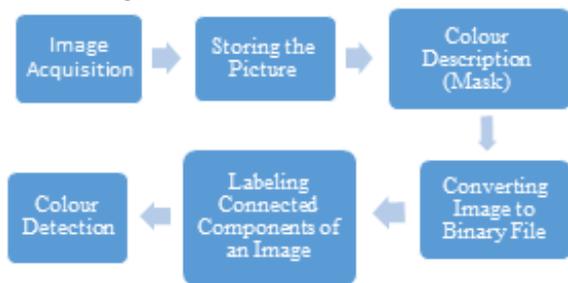


Figure 4. Feature extraction algorithm of two-dimensional image

3.2. HSV Color Space

The HSV color space shown in Figure 5 is separated by the luminance component and the chrominance component. While the brightness component was created from V (Intensity value - Value), the chrominance component was created from H (Hue - Hue) and S (Saturation - Saturation) values. In the HSV model, colors such as red, yellow, green, blue, or the combination between these colors are defined by hue. The tone value is between 0-360°. The degree of dilution of pure color with white light is given by saturation, and the intensity of the color between light and dark is shown by the intensity value [9].

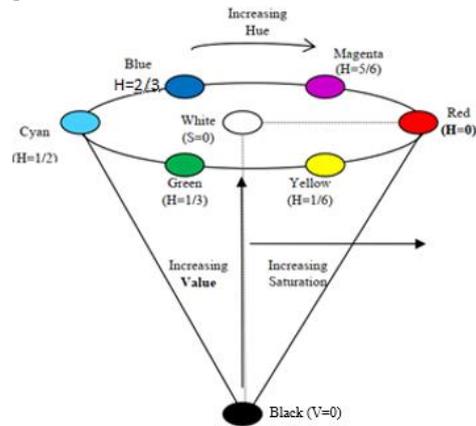


Figure 5. HSV Color Space

The HSV color space works very similarly to the color perception of the human eye and can be used as an ideal model in the image processing algorithm. In the study, more than one mask was applied to isolate the colors. Low threshold and high threshold masks are used for hue, saturation, and intensity value. Any pixels within these thresholds will be set to one and the remaining pixels will be set to zero. HSV color space masks used in the study are given in Table 1.

Table 1. HSV color space masks

Color	Hmin	Hmax	Smin	Smax	Vmin	Vmax
Yellow	0.1	0.18	0.4	1	0.4	1
Green	0.9	0.41	0.3	1	0.1	1
Red	0.97	0.047	0.4	1	0.35	1
Purple	0.7	0.96	0.1	1	0.1	1
Blue	0.58	0.68	0.25	1	0.25	1

3.3. Camera Calibration

Camera calibration is a method to see the camera parameters. These parameters can be used to make adjustments for lens distortion, scale the size of an object in world units, or determine the camera's location in the scene. These tasks are practiced in applications such as machine vision to identify and measure things. They are also applied in robotics, navigation systems, and 3-D scene reconstruction [12]. This calibration is used in applications such as machine vision to detect and measure objects. The algorithm of the camera calibration process performed in the study is given in Figure 6. In the study, the camera calibration process was started by taking a few sample checkerboard photographs with the camera. All sample photos were saved in a single folder. By detecting the corners of the checkerboard, image points and checkerboard size were obtained. The board size and the corner size of the checkerboard square in the previous step were used to obtain the world points of the checkerboard. With the two inputs (image points and world points) from the previous two steps, the camera calibration process was started. After this process, the internal parameters of the camera were obtained. To obtain the external parameters of the camera, three inputs are required, namely image points, world points, and camera parameters. The viewport used in this step was the viewport of the image found in the previous step. Also, the external parameters of the camera, namely rotation matrix and transformation matrix would be obtained. The camera calibration flowchart is shown in Figure 6. The camera calibrated image is shown in Figure 7.

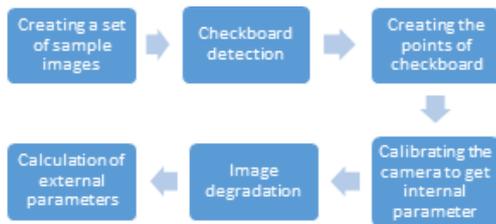


Figure 6. Camera Calibration Scheme

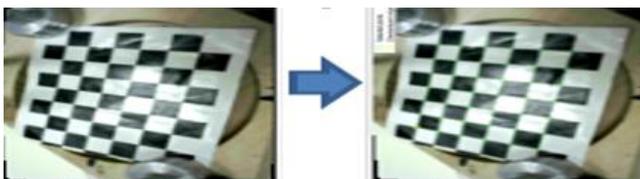


Figure 7. Camera calibration images

3.3.1. External and Internal Parameters

External Parameters: The external parameters consist of a rotation (R) and a transformation (t). The origin of the

camera's coordinate system is at its optical center, and the x and y -axis are defined as the image plane.

Internal Parameters: Internal parameters include focal length (f_x, f_y), optical center (also known as a principal point (c_x, c_y)) and skew coefficient (s). The camera's inner matrix K is shown in Equation (1) [12, 17].

$$K = \begin{bmatrix} f_x & 0 & 0 \\ s & f_y & 0 \\ c_x & c_y & 1 \end{bmatrix} \quad (1)$$

3.3.2. Calibration Process

The camera calibration process passes through 5 stages and then is saved to a file. These processes start with pattern selection. The selected pattern is mounted on a flat, stable surface. Then, pictures of the placed surface are taken from various angles and distances. The ones in focus from the captured images are selected and the rest are eliminated. Then, with the help of the remaining focused pictures, the process ends and the calibration process is concluded by saving them to the file.

The internal and external parameters obtained from the camera calibration will then be used to estimate the coordinates of the objects in the delta robot's workspace. In this study, the checkerboard was used as a sample pattern. The best calibration results have been achieved by placing the checkerboard unsymmetrically and between 45 and 90 degrees from the camera.

Logitech C310 HD Webcam is used in this project. As the more sample photos are used for camera calibration, the more accurate the camera parameters estimation results will be, so 25 photo samples were used for camera calibration, as shown in Figure 8.

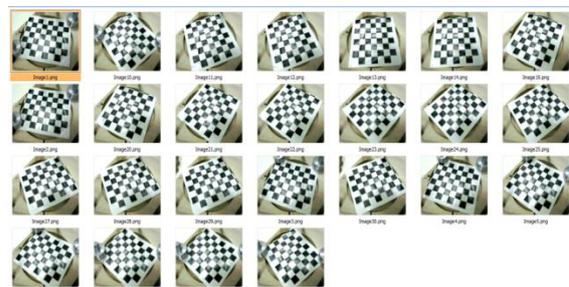


Figure 8. Sample images for camera calibration

Internal and external parameters K (Intrinsic Matrix), rotation matrix R (Rotational Matrix), and transformation matrix t (Translation Vectors) were obtained as a result of the calibration process are given in Figure 9.

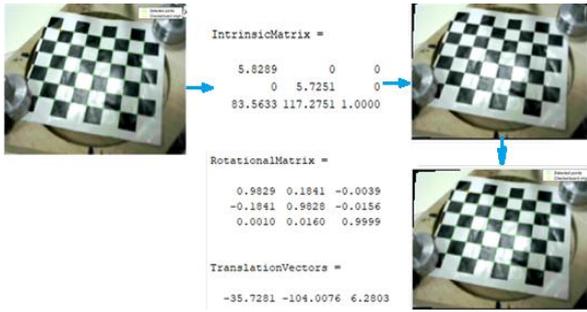


Figure 9. Internal and external parameters achieved from the checkerboard pattern

3.4. BLOB Analysis

BLOB stands for Binary Large Object and refers to the group of connected pixels in the binary image. It is indicated by the term "Large" that only objects of a certain size are processed and "small" binary objects are classified as noise.

3.4.1. BLOB extraction

The purpose of BLOB extraction is to isolate BLOBs (objects) in a binary image. The BLOB is composed of a group of connected pixels. Whether two pixels are connected or not is defined by the connection, that is, which pixels are neighbors and which are not. The two most common types of connections are given in Figure 10. The results of BLOB 4 and 8 connections applied to the same image were extracted as two and one object, respectively. 8 connections give a better result than 4 connections.

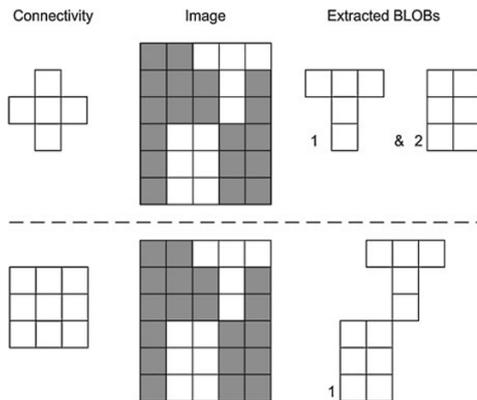


Figure 10. 4 and 8 connections. The effect of applying two different connection types

3.4.2. Obtaining a Coordinate using BLOB analysis

Several features can be extracted from the BLOB analysis. Only Bounding Box and Centroid (Center of

mass) were used in this study.

Bounding Box: The minimum rectangle that contains the BLOB. It is defined for a BLOB by going through all pixels and finding four pixels with minimum x value, maximum x value, minimum y value, and maximum y value respectively. From these values, the width of the bounding box is given as $x_{max} - x_{min}$ and the height as $y_{max} - y_{min}$.

Center of mass: The location on the object where we would place our finger to balance the thing. The focus of size for a binary image is similar. It is the standard x and y positions of the binary object. It is defined as a point whose x-value is calculated by summing the x coordinates of all pixels in the BLOB and dividing by the total number of pixels. A similar process is done for the y-value. In mathematical terms, the center of mass, (x_c, y_c) is measured as

$$x_c = \frac{1}{N} \sum_{i=1}^N x_i, \quad y_c = \frac{1}{N} \sum_{i=1}^N y_i \quad (2)$$

N is the number of pixels in the BLOB and x_i and y_i are the x and y coordinates of N pixels, respectively. In cases where the BLOB contains "additional pieces", the median can be changed. The collection box is shown in Figure 11.

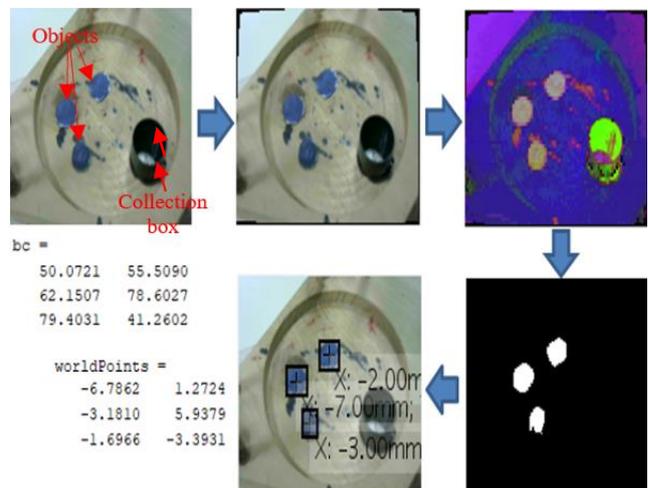


Figure 11. Coordinates of Real World Points

3.5. Real World Coordinate

In this study, the pixel position of the object obtained from the pinhole model and BLOB analysis feature is used to obtain the real-world coordinate. At this stage, rotational and translation of external parameters from the inner camera matrix, the resulting camera calibration, are important. The greater the accuracy of the results from the

calibration, the greater the accuracy of the real-world coordinates.

3.6. Pinhole Camera Model

The pinhole camera is a simple model without a lens and with a single small aperture, as shown in Figure 12. Light rays pass through the aperture and project an inverted image to the opposite side of the camera. The virtual image plane is in front of the camera and contains a vertical view of the scene.

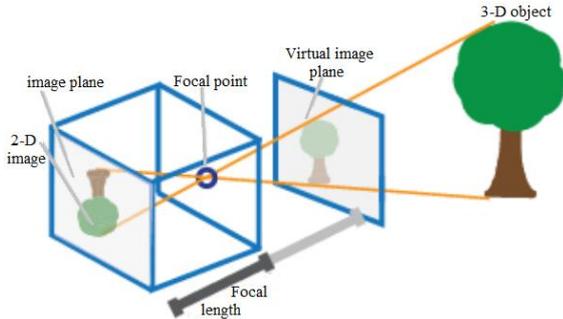


Figure 12. Pinhole Camera Model [17]

The pinhole camera parameters are represented in a 4x3 matrix called the camera matrix. This matrix maps the 3D world scene to the image plane. The camera matrix is calculated using the calibration algorithm, external and internal parameters. The external parameters are represented by the position of the camera in the 3D scene. The internal parameters represent the optical center and focal length of the camera. Using these three parameters, the real-world coordinates can be calculated in equations (3) and (4).

$$w \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} P \quad (3)$$

$$P = [R \quad t] K \quad (4)$$

- w : scale factor
- x, y : image points (pixels)
- X, Y, Z : world points (mm)
- P : camera matrix
- R : rotational matrix
- t : translation matrix
- K : inner matrix

World points were mapped to camera coordinates using external parameters, and camera coordinates were mapped to the image plane using internal parameters [17].

4. Kinematic Analysis

One of the most essential tools in controlling a robot is the Inverse and Forward Kinematic solutions. The Inverse Kinematics is practiced to define the angle of each motor shaft for an end effector location in (x, y, z) coordinates [12]. The Inverse Kinematics is particularly significant in the control of the robot because they play a role in every move made by the robot [11]. The Inverse Pose Kinematics is required for the control of any robot. The Forward Kinematics is utilized to arrange the end effector location in (x, y, z) coordinates if the angle of each motor shaft is identified. The Coordinates of motion surface is shown in Figure 13. The L1, L2, L3 Orbit is shown in Figure 14. The Inverse and Forward Kinematics Result is shown in Figure 15.

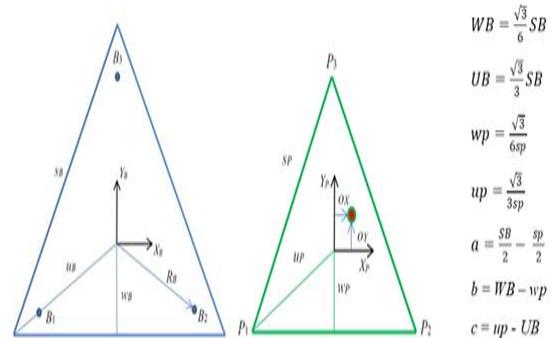


Figure 13. Coordinates of motion surface

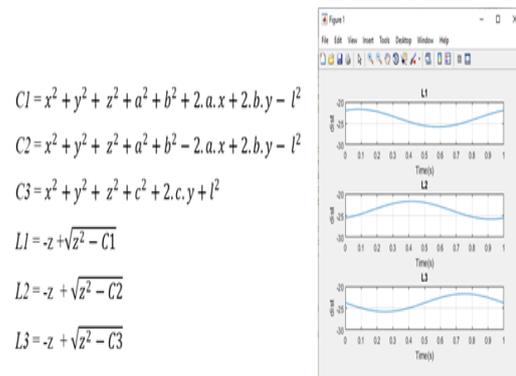


Figure 14. L1, L2, L3 Orbit

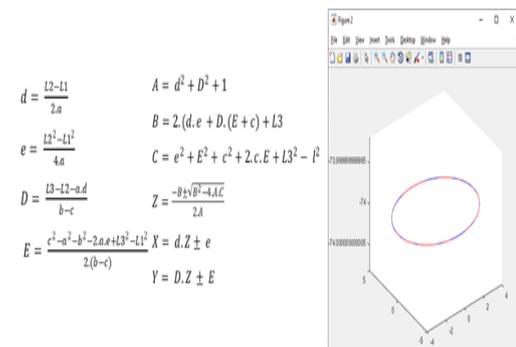


Figure 15. Inverse and Forward Kinematics Results

5. Visual Positioning Based on Proposed Techniques

In this work, before we can move the robot to the desired coordinate from image processing, we need to set it to the home position. This can be done using an inductive sensor. After it is set to its home position, we can continue to take the image and get the coordinates of objects.

The coordinates that we get from image processing are still in x, y, z. This has to convert into the trajectory of each robot leg. Kinematic Analysis plays a significant role in this step. [16].

After we get the trajectory of each leg, we then calculate the velocity of each step. The velocity of each leg is saved in an array named V1, V2, V3. In this study, an electromagnet is used. The electromagnet will turn on if we give logic “1” and will turn off if we give logic “0”. The electromagnet will be ON during picking and lifting the object from the found coordinates and carrying it to the box and will be OFF during placing and lifting from the box to the next object. The required Logic Data for electromagnet is saved into an array named M. Then the data that we get from image processing (L1, L2, L3, V1, V2, V3, M) is sent to PLC through OPC.

The data is saved into OPC_Data Block in PLC. This data will be used to control the movement of each stepper motor [18]. The PLC Programming Map is shown in

Figure 16. The Image taken from camera is shown in Figure 17. The Schematic drawing of the experimental air conditioning system is shown in Figure 18.

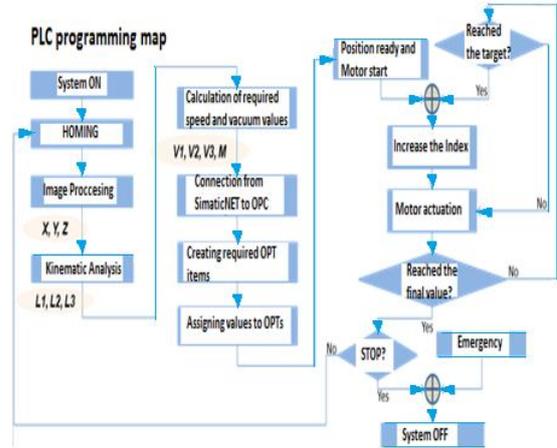


Figure 16. PLC Programming Map

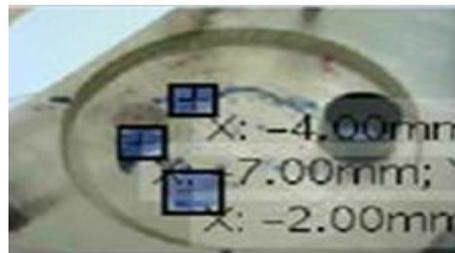


Figure 17. Image taken from camera

	X	Y	Z
0	0	0	0
1	-6.79	2.97	-90
2	-6.79	2.97	-76.80
3	-6.79	2.97	-90
4	8	0	-90
5	8	0	-76.80
6	8	0	-90
7	-2.12	7.21	-90
8	-2.12	7.21	-76.80
9	-2.12	7.21	-90
10	8	0	-90
11	8	0	-76.80
12	8	0	-90
13	-4.45	-2.54	-90
14	-4.45	-2.54	-76.80
15	-4.45	-2.54	-90
16	8	0	-90
17	8	0	-76.8
18	8	0	-90
19	0	0	0

Indeks	L1	L2	L3	V1	V2	V3	M
0	0	0	0	0	0	0	0
1	-9.8182	-3.717	-9.1024	0.8182	0.3097	0.7585	0
2	-23.018	-16.917	-22.302	3.3	3.3	3.3	0
3	-9.8182	-3.717	-9.1024	3.3	3.3	3.3	1
4	-3.9002	-11.041	-7.5395	1.4795	1.831	0.3907	1
5	-17.1	-24.241	-20.74	3.3	3.3	3.3	1
6	-3.9002	-11.041	-7.5395	3.3	3.3	3.3	0
7	-6.66	-4.732	-11.221	0.69	1.5772	0.9204	0
8	-19.86	-17.932	-24.421	3.3	3.3	3.3	0
9	-6.66	-4.732	-11.221	3.3	3.3	3.3	1
10	-3.9002	-11.041	-7.5395	0.69	1.5772	0.9204	1
11	-17.1	-24.241	-20.74	3.3	3.3	3.3	1
12	-3.9002	-11.041	-7.5395	3.3	3.3	3.3	0
13	-10.33	-6.3913	-6.416	1.6076	1.1624	0.2809	0
14	-23.53	-19.591	-19.616	3.3	3.3	3.3	0
15	-10.33	-6.3913	-6.416	3.3	3.3	3.3	1
16	-3.9002	-11.041	-7.5395	1.6076	1.1624	0.2809	1
17	-17.1	-24.241	-20.74	3.3	3.3	3.3	1
18	-3.9002	-11.041	-7.5395	3.3	3.3	3.3	0
19	0	0	0	0.325	0.9201	0.6283	0

Figure 18. Location data and values that written to OPC

6. Conclusions and Recommendations

Delta Robot takes a very important place in industry 4.0. Its ability to work fast and in precision is the reason why this robot is used worldwide in the industry.

Delta robots are mostly used for packaging, 3D printing, pick and place, etc. This robot can be controlled

using programmable logic controller (PLC) devices. In recent years, image processing methods have been widely used in many robot applications. This study, it is aimed to ensure that the robot successfully moves to the determined points by using image processing methods. For the study, images were obtained by attaching a camera to the robot system. All objects on the workstation of the robot can be

detected according to their colors. HSV color space is preferred since its mask is more accurate than RGB. In this study, color thresholding was preferred because of its easy use and high success rate. The number of samples used in the calibration is one of the factors that increase the accuracy in determining the real-world coordinates.

In this study, the coordinates are determining methods are chosen according to the needs of the robot. Coordinates from outside the study area are not suitable. It needs such manipulation that it suits the need. The coordinates taken from image processing are 100% successful in being reconstructed by kinematic analysis and the data of each trajectory of the leg is successfully transferred to OPC Data in PLC Data Block through SimaticNet OPC Server.

Some mechanical errors are found in this work; some coordinates need a different z position than the others. These errors are caused by the lack of ability of the robot design itself. The accuracy of movement is limited to a certain point. This is why for pick and place jobs; parallel delta robot is more preferred than linear delta robot.

Declaration of Ethical Standards

The author of this article declares that the materials and methods used in this study do not require ethical committee permission and/or legal-special permission.

Conflict of Interest

The author declares that she has no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] Köse İ., Öztürk S., Kuncan M., 2019. Pantography application with real-time PLC based on image processing in gantry robot system. *European Journal of Technique (EJT)*, **9**, pp. 219-229.
- [2] Çubukçu A., Kuncan M., İmren M., Erol F., Ertunç HM., Öztürk S., & Kaplan K., 2015. Görüntü işleme ile 3 eksenli robot mekanizması üzerinde nesne ayırtılması ve sıralanması. *Otomatik Kontrol Ulusal Toplantısı*, pp. 637-641.
- [3] Bakır A., Güney ÖF., Kuncan M., Ertunç HM., 2012. 3 Eksenli Robot Mekanizmasına Monte Edilmiş Bir Kamera Vasıtasıyla Farklı Rotasyon ve Boyutlardaki Geometrik Cisimlerin Tanımlanarak Vakum Tutucu ile Ayrılması. *Otomatik Kontrol Ulusal Toplantısı*, Niğde, Turkey, pp. 606-609.
- [4] Horoz E., Öten HF., Kuncan M., Ertunç HM., 2013. Kamera Yardımı ile Ayırt Edilen ve Tanımlanan Cisimlerin 3 Eksenli Robot Mekanizması ile Taşınması. *Otomatik Kontrol Ulusal Toplantısı*, Malatya, Turkey, pp. 1236-1240.
- [5] Warin P., Kamol C., and Ratchatin C., 2013. VISUAL POSITIONING OF A DELTA ROBOT, Mechanical Engineering Department, Chulalongkorn University, Bangkok, 10330, Thailand, April 20.
- [6] Vishesh G., Tarun J., Sahil S., Silica K., 2017. Specific Color Detection in Images using RGB Modelling in MATLAB, *International Journal of Computer Applications*, **161**, March 2017, pp. 0975 – 8887.
- [7] Robert L., Williams II., 2015. Ph.D., The Delta Parallel Robot: Kinematics Solutions, Mechanical Engineering, Ohio University, April.
- [8] Joseph Q., 2016. Oberhauser, Design, Construction, Control, and Analysis of Linear Delta Robot, the faculty of the Russ College of Engineering and Technology of Ohio University, April.
- [9] Priyanto H., Miftahuddin Z., 2015. Color-Texture Based Object Tracking Using HSV Color Space and Local Binary Pattern, *International Journal on Electrical Engineering and Informatics*, **7**, June.
- [10] Senthilkumaran N., Vaithegi S., 2016. Image Segmentation by Using Thresholding Techniques for Medical Images, *Computer Science & Engineering: An International Journal (CSEIJ)*, **6**, February.
- [11] Seth H., Gregory DH., Peter IC., 1996. A Tutorial on Visual Servo Control, *IEEE Transactions on Robotics and Automation*, **12**, October.
- [12] Burger W., 2016. Zhang's camera calibration algorithm: in-depth tutorial and implementation. **16**, pp. 1-6.
- [13] Vocetka M., Huňady R., Hagara M., Bobovský Z., Kot T., & Krys V., 2020. Influence of the Approach Direction on the Repeatability of an Industrial Robot. *Applied Sciences*, **10**, 8714.
- [14] Bolton W., 2015. Programmable logic controllers. Newnes.
- [15] Krawczyk M., 2022. Project and implementation of the control of the loading and unloading station of the technological line for surface treatment of metal bottles (Doctoral dissertation, Instytut Automatyki Robotyki).

- [16] Guzmán HSG., 2017. OOP with S7-SCL, implemented to plastic welding machines.
- [17] Fetić A., Jurić D., & Osmanković D., 2012. The procedure of a camera calibration using Camera Calibration Toolbox for MATLAB. In 2012 Proceedings of the 35th International Convention MIPRO, pp. 1752-1757.
- [18] Tekinalp Z., Öztürk S., Kuncan M., 2013. OPC Kullanılarak gerçek zamanlı haberleşen Matlab ve PLC kontrollü sistem. Otomatik Kontrol Ulusal Toplantısı, TOK2013, pp. 26-28.