Case Study on Color Sorting Robotic Arm

Amin Yazdani
12404412
Technische Hochschule Deggendorf
Cham, Germany
amin.yazdani@stud.th-deg.de

Yash Chaudhary
12403014
Technische Hochschule Deggendorf
Cham, Germany
yash.chaudhary@stud.th-deg.de

Aakash Vanmali
12402734
Technische Hochschule Deggendorf
Cham, Germany
aakash.vanmali@stud.th-deg.de

Amirhossein Ghalebi
22213143
Technische Hochschule Deggendorf
Cham, Germany
amirhossein.ghalebi@stud.th-deg.de

***Abstract*—This paper presents the design and analysis of a Robotic Arm with the capability to sort objects based on their color. The study has been performed using MATLAB Simulink and its associated libraries. The robotic arm is designed to sort red, green and blue cube shaped objects in separate piles. To achieve this, a combination of mechanical design and image processing has been used. Additionally, a vaccum gripper system is employed to achieve the gripping of the object. The project showcases the integration of image processing, robotics, and control systems, providing a comprehensive case study on the capabilities and versatility of MATLAB in robotic applications.**

***Index Terms*—RRR robotic arm, image processing, control algorith**

I. Introduction

In the rapidly advancing industrial landscape that marks today's world, automation and robotics are crucial for enhancing efficiency, precision, and productivity. Modern manufacturing processes rely on robotic systems with abilities to undertake intricate or repetitive tasks with high accuracy and speed. This is required by the industry because robots, these days, are used in automotive manufacturing industries to perform complex tasks such as welding, pick and place, painting, etc.

For instance, the automotive industry has robots like those found at Tesla's Gigafactory that perform tasks such as welding, painting, and assembly, which ensures quality production and steady output. Companies like Foxconn also use robots in their electronic sectors where they assemble smartphones and other portable electronic gadgets with unmatched precision and efficiency.

This project objective is to develop a pick-and-place robot arm, specifically, RRR (Revolute-Revolute-Revolute) type that will sort objects by their visual characteristics. A camera is mounted on the top of the frame which identifies the colors of the objects—blue, red or green—and their positions.

Using Simulink and Simscape in MATLAB, a simulation model is built that covers the whole robot control system. The Denavit- Hartenberg Convention Table to calculate forward and inverse kinematics which determine position and orientation of end effector. The arm is capable of making precise movements with trigonometric based inverse kinematics to ensure objects are aligned for sorting.

The main focus of this project is on integrating image processing with robot control. Initially, the camera classifies objects based on their color. Later on, the robot navigates itself around these objects in order to optimize its workspace using path planning techniques. This results into a system that performs efficient and accurate sorting operations thus demonstrating the robot's capability.

It is imperative to demonstrate how RRR industrial robots like this can be used in production, showing automation as an important part of both assembly line and sorting processes industrially. The target is therefore to illustrate how advanced kinematics as well as controlling mechanisms can be employed to improve efficiency, performance or functionality of industrial robots. In summary, it has been proved that adoption of robotics will revolutionize modern manufacturing process through proper object classification and good sorting systems.

II. Aim and Objectives

The aim of this case study is to create a MATLAB Simulink model of the various components of a color sorting robot and simulate it in the Simscape environment.

The objectives of the case study are as follows –

- Create a robot model in Solidworks.
- Create a mathematical model of the robot arm in Matlab.
- Demonstrate the motion of the robot arm with a path planner algorithm.
- Include basic image processing in the robot arm.
- Configure the vacuum gripper in Matlab with the required properties.
- Design the environment around the robot arm with multicolor objects.
- Integrate the blocks together and demonstrate the working of the robot.

III. Literature Survey

The case study required referencing lot of papers as there wasn't a singular paper which covered all the aspects of the project. Thus, several papers were referred which had information about different aspects of the case study.

One such paper is titled "Kinematics Analysis and Simulation of a Robotic Arm using MATLAB" written by "Alia N, Barakat, Khaled A.Gouda, Kenz A. Bozed" in 2016. This paper focused on the use of the DH convention Method for the kinematic modelling of the robotic arm. They utilised MATLAB to simulate their equations and created a 3D visual simulation of the robot arm. A 6 DOF robot was utilised in their study. This paper provided the reference for the use of MATLAB for the task of simulating the robot arm inside the Simscape simulation environment. [1]

Another referred paper titled "Development of a 3 DOF Color Sorting based Robotic Arm using MATLAB GUI" published by "Tonusree Mohanto, Aparajita Talukde, Zinat Tasneem" in 2020. This paper utilised a 3 DOF robot, comparing both kinematic and inverse kinematic approaches to show the position and orientation of the end effector. Used MATLAB for the simulation of the mathematical models. Also focused on RGB color sorting to differentiate different colored boxes. This paper describer the use of a image processing algorithm to use for the color sorting of objects. It also used and compared both forward and inverse kinematic models for the calculations and simulations of the robot arm. [2]

Next paper referred is titled "Modeling and Simulation of Robotic Arm in MATLAB for Industrial Applications" given by "Juan Liu, Qiang Luo" in 2019. This paper investigates the modelling and simulation of a 6 DOF robotic arm using MATLAB and Robotics Toolbox. It utilised forward kinematics for the dynamic modelling of the arm. A standard DH table was created for the mathematical modelling. Jacobian Matrix was used for the path control of the robot arm. The details about the forward kinematics required in the robot arm were highly derived from this paper. It gave detailed insights regarding the different blocks used in the modelling and their programming. [3]

Last but not the least, a paper titled "Model and simulation of arm robot with 5 degrees of freedom using MATLAB" published by "M. Sabri, Rahmad Fauzi, M. S. Fajar, H. S. Geubrina, Faris A. M. Sabri" in 2020. This paper describes a 5 DOF robotic arm which uses both forward and inverse kinematics to create a robot simulation. MATLAB Simulink was used for the mathematical modelling. The movement of the gripper and the torque produced at each joint was calculated and charted using MATLAB. With this, the use of the different equations required in the forward and inverse kinematic blocks was made clearer. This was very similar to the case study, however it utilised a normal friction based gripper, however, the case study focused on a vaccum gripper. [4]

IV. Methodology flowchart

The working of the robot is described by the flowchart of the planned working of the robot arm. The given flowchart in Fig. 1, shows the desired steps of operation for the robot arm. The first step is to capture and process the image of the work table from the top view of the robot. This will clearly highlight the objects inside the space and clearly show the position of the object. The position is first calculated as number of pixels from the origin and then converted to the XY coordinate. Then, activate the servo and move the end effector to the designated zero position. Then the motor is again activated and the end effector moves to the position of the correct object. Then it moves downwards, the vaccum gripper is activated and the object is gripped and lifted up. It is then moved to the its target location and the vacuum gripper turns off as the arm lowers detaching the object. The actuator returns to zero position and the cycle continues until all objects are in their respective piles.

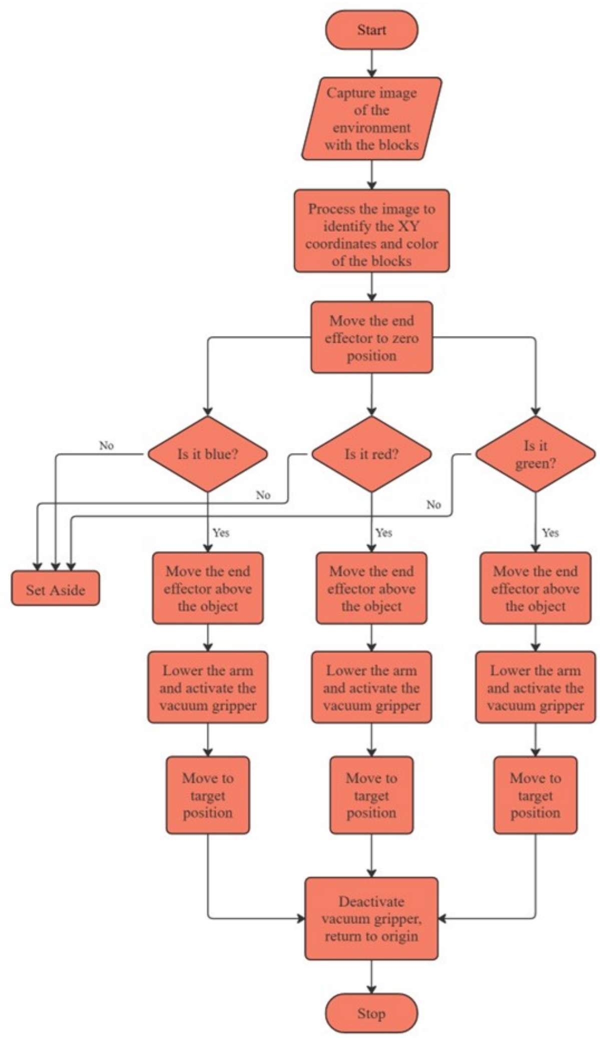

Fig. 1. Flowchart of the working of the robot arm

V. System Description

A. Solid Model

The CAD model shown in Fig. 2 is created based on the requirement of the case study and the referred references. The created robot model is of a 4R configuration i.e. it has 4 revolute joints which gives it the maximum amount of flexibility. It has 3 links and the vacuum gripper as the end effector. The vacuum gripper is modelled to allow it to grip the objects with ease. This leads to a work envelope in the shape of a sphere.

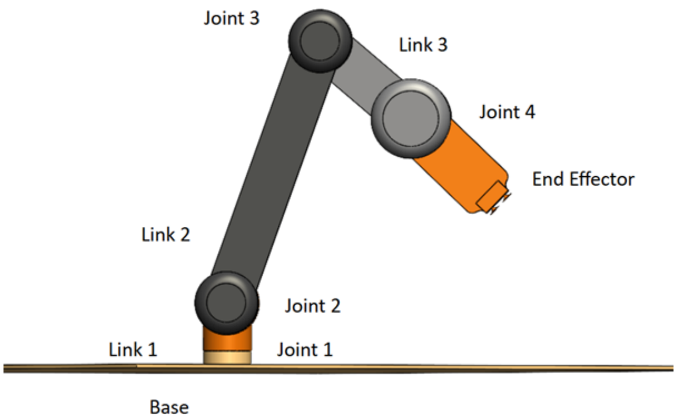

Fig. 2. CAD Model of the robot arm

B. MATLAB Simulink System

The Fig. 3 shows the system designed in MATLAB Simulink for the simulation of the different subsystems in the robot simulation model. Many sub-systems were used to define and control the functionality of the robot model. These subsystems included a path planner, inverse kinematics for control of the robot, robot links, joints and their inputs and outputs, forward kinematics to control the end effector, suction blocks to control the vacuum gripper and finally the base frame which includes the table, object positions and the world frame. Image processing is also included, however, it is done offline and thus is not part of the Simulink model. The image is processed and the coordinates are provided to the path planner in the Simulink model. These sub-systems are explained in detail in the following sections.

C. Image Processing

The pick and place robot has three blocks in the environment of red, green and blue color. Image processing was implemented to find the coordinates of these blocks and provide them as end effector goal positions for each block. These coordinates were used by the path planning algorithm to plan a path between the zero position and the goal position for the respective blocks.

For image processing, an image of the environment from the top view was captured. Then the red, blue and green

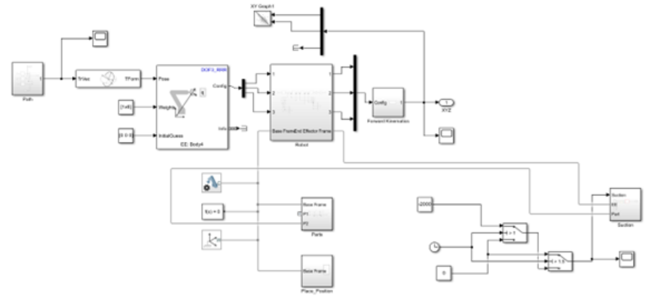

Fig. 3. MATLAB Simulink System

channels of the image were extracted and stored in variables in the form of matrices. A logical mask was created for each matrix where the pixels above 180 were set to “True” and all others to “False”. This helps in detecting the blocks in the environment, as only the pixels above the threshold value are detected and all other pixels that contain the other elements in the environment were eliminated. This reduces the noise and allows for accurate detection of the blocks. The indices of these “True” values were found and their minimum and maximum values were calculated, to find the center of the block. These coordinates were then transformed to obtain the actual coordinates for the red, green and blue blocks in the environment.

Fig. 3 shows the image captured for the image processing algorithm. The robot arm and the 3 blocks are clearly visible. The algorithm is then able to distinguish the 3 blocks based on their colors and form 3 different matrices with their XY coordinates. These are processed and visualised based on the images shows in Fig. 4, Fig. 5 and Fig. 6 for red blue and green blocks respectively.

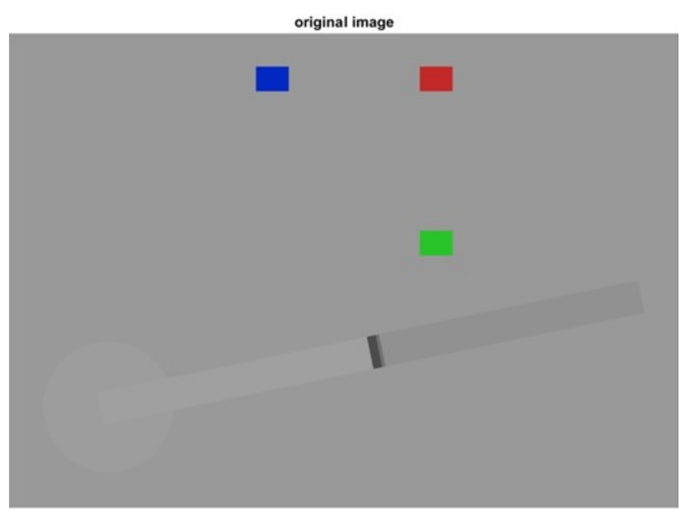

Fig. 4. Top view image of the robot workspace

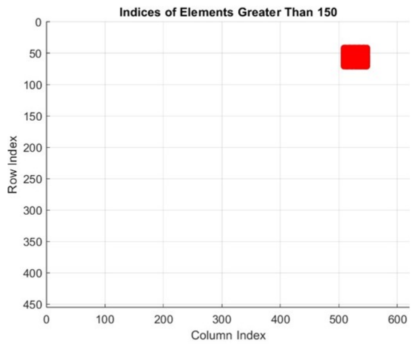

Fig. 5. Red object coordinates

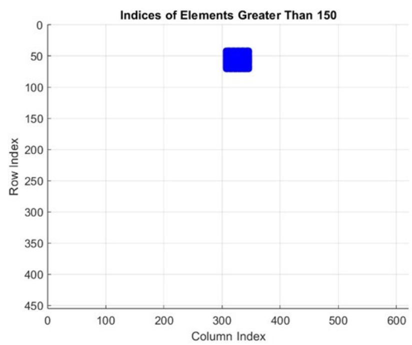

Fig. 6. Blue object coordinates

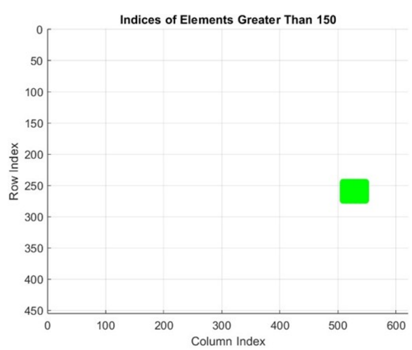

Fig. 7. Green object coordinates

of the weights, and the third input was the zero position of the end effector. The final output of the inverse kinematics was the joint angles of the actuators which were provided to the respective joints. These joint angles were then used by the forward kinematics to determine the position and orientation of the end effector.

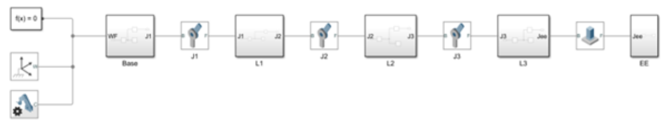

Fig. 8. Robot Joints and Actuators

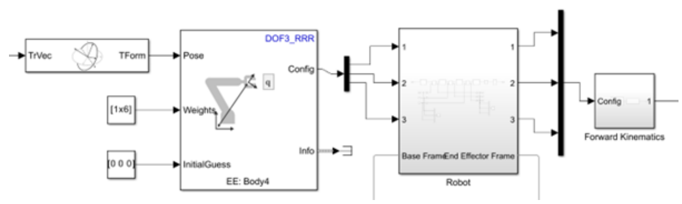

Fig. 9. Robot Subsystem with IK and FK

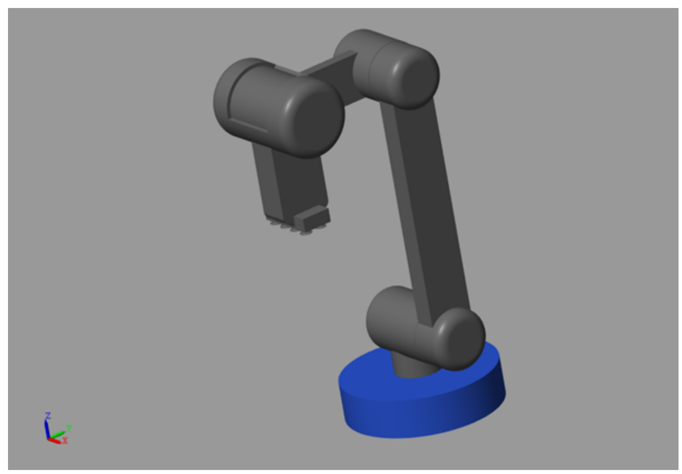

Fig. 10. Robot Model in MATLAB environment

D. Robot Model Subsystem

The robotic arm that was used for the simulation is a R-R-R robot with three revolute joints with an end effector that is a vacuum gripper. This configuration allowed for a spherical workspace and maximum flexibility. To control the movement of the robotic arm, forward and inverse kinematics were used. It performed a series of transformations based on the length of the arms and the joint angles. The inverse kinematics were used to calculate the joint angles with the input of the desired position and orientation of the end effectors. The inverse kinematics received three inputs: one from the path planner, which provided the path that the robotic arm must follow to complete the pick and drop task, the second input consisted

E. Path Planner

The path planner took the coordinates of the centers of the colored blocks as input from the image processing algorithm. The code then created a lookup table that defined the path of the end effector inside the Simscape simulation. The end effector first moved to the coordinates of the center of the green block and picked up the block by switching the suction on. Once it reached the place position for the green block the suction was turned off. Thus, it placed the green block to the predefined place position. The end effector then returned to the zero position. This process was repeated for the blue and red blocks, placing them in their respective place position before returning back to the zero position each time. The

Simulink model of the path planner is shown in Fig. 10
The predefined coordinates for the location of the place position and the zero position of the end effector is given as follows -
Predefined Positions:

- Zero Position: [0.1, 0, 0.2]
- Place Green: [0.2, -0.15, 0.02]
- Place Red: [0.15, -0.15, 0.02]
- Place Blue: [0.1, -0.15, 0.02]

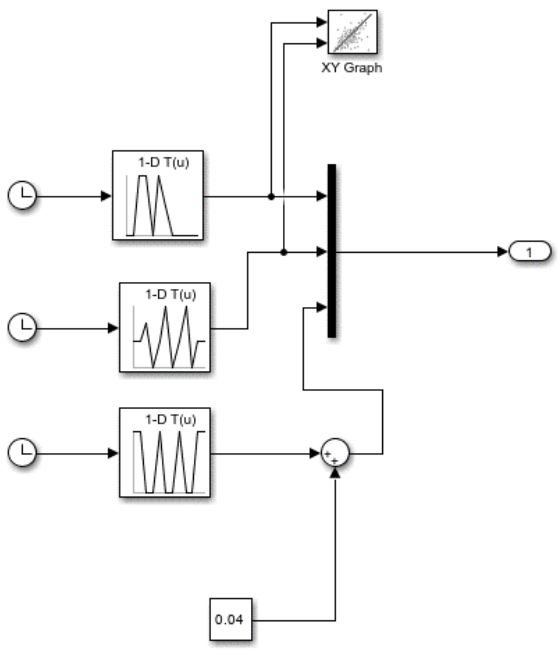

Fig. 11. Simulink Model of the Path Planner

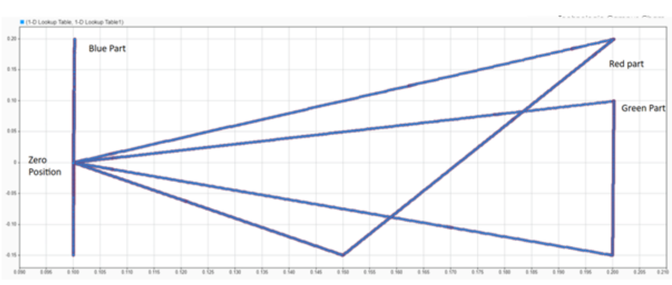

Fig. 12. Path Planner path lookup table

F. Suction Block

The suction block is used for generating an internal force between the vacuum based end effector and the parts. It sends the signal that provides the action logic of this block. This signal passes through a gain block, which applies a negative value according to the suction force. The output from the gain block is converted to a physical signal using a PC converter. This physical signal is then fed into the internal force block, which generates the required suction. Additionally, two physical signals from the end effector and parts are also input to the internal force block, enhancing the suction mechanism to grip objects effectively. This integrated process makes control and application of suction force for lifting objects controllable.

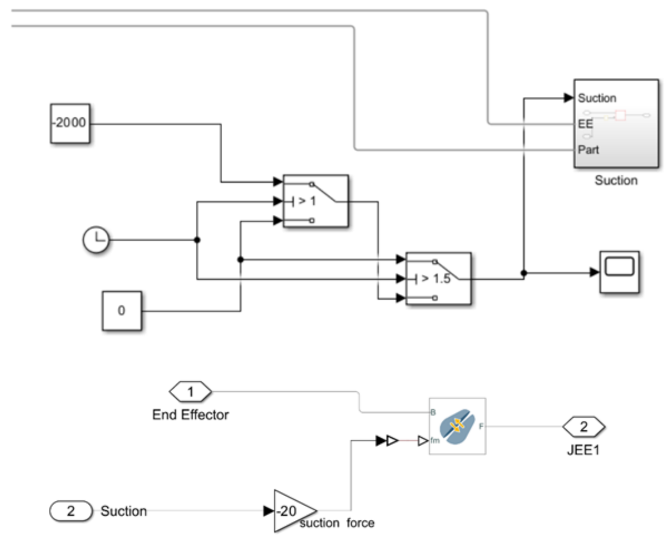

Fig. 13. Suction System Model

VI. Results

The path planner is able to create the path based on the inputs from the image processing algorithm. The end effector is able to traverse this path, however with minor inaccuracies. However, these inaccuracies were within design and thus did not have much effect on the path of the end effector. In the Figs. 14 15 and 16, the blue lines indicate the planned path for the end effector and the yellow lines indicate the actual path taken by the end effector. As it is visible, apart from the X axis graph, the Y and Z were able to follow it essentially perfectly. This was, however, based on the robot model without the CAD model attached.
When the solid model was added instead of simple links, the amount of variation and unevenness increased tremendously. This presumably took place due to the scale of the design robot being much larger than the MATLAB environment and thus accurate control of the joints was lost. It was not feasible to scale up the environment due to processing power constraints and thus the simulation was performed on the MATLAB based model with the simple links.

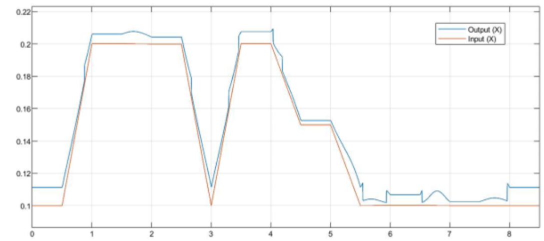

Fig. 14. End Effector X path

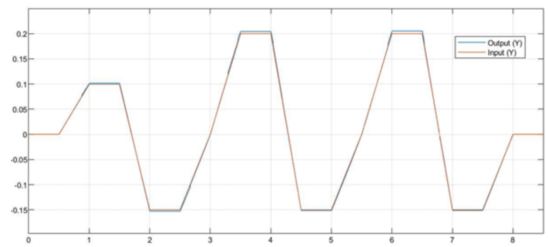

Fig. 15. End Effector Y path

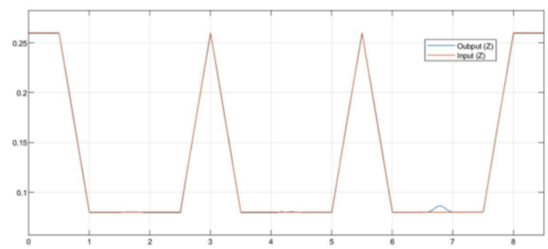

Fig. 16. End Effector Z path

VII. Conclusion

The robotic arm is able to perform the pick and place operation, however, the suction gripper does not work consistently. The arm is able to accurately go to the position of the colored blocks and place them in the predefined place position. The image processing is able to accurately distinguish between the red, green and blue blocks in the environment and find the coordinates of their centers. The path planning algorithm accurately determines the path of the end effector using the lookup table which is generated using the goal, the center coordinates of the colored blocks and the zero position. The forward and inverse kinematics can accurately calculate the transformation of joint angles and the final end effector position and orientation. In conclusion, all the objectives of the case study were successfully met.

References

[1] A. N. Barakat, K. A. Gouda and K. A. Bozed, "Kinematics analysis and simulation of a robotic arm using MATLAB," 2016 4th International Conference on Control Engineering and Information Technology (CEIT), Hammamet, Tunisia, 2016, pp. 1-5, doi: 10.1109/CEIT.2016.7929032. keywords: Robot kinematics;Zinc;MATLAB;Jacobian matrices;Kinematics;Trajectory;Modeling;Simulation;MATLAB;6 DOF Robot arm;Forward Kinematics;Inverse Kinematics;Trajectory Planning;Jacobian.

[2] T. Mohanto, A. Talukde and Z. Tasneem, "Development of a 3DOF Color Sorting based Robotic Arm using MATLAB GUI," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ICCIT51783.2020.9392750. keywords: Image color analysis;Computational modeling;Kinematics;Robot sensing systems;Mathematical model;Load modeling;Sorting;3 DOF;kinematics;inverse kinematics;end effecter;MATLAB;GUI;RGB color;color sorting.

[3] J. Liu and Q. Luo, "Modeling and Simulation of Robotic Arm in MATLAB for Industrial Applications," 2019 11th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, China, 2019, pp. 346-349, doi: 10.1109/IHMSC.2019.00086. keywords: robotic arm;simulation;kinamatics model;MATLAB.

[4] M Sabri et al 2021 IOP Conf. Ser.: Mater. Sci. Eng. 1122 012032 DOI 10.1088/1757-899X/1122/1/012032.