

Fuzzy Logic-Controlled 6-DOF Robotic Arm Color-based Sorter with Machine Vision Feedback

Alexander C. Abad¹, Dino Dominic Ligutan¹,
Elmer P. Dadios²

¹Electronics and Communications Engineering Department
²Manufacturing Engineering and Management Department
Gokongwei College of Engineering
De La Salle University
Manila, Philippines

Levin Jaeron S. Cruz, Michael Carlo D.P. Del Rosario,
Jho Nathan Singh Kudhal

Electronics and Communications Engineering Department
Gokongwei College of Engineering
De La Salle University – Laguna Campus
Biñan Laguna, Philippines

Abstract—A demonstration of the application of fuzzy logic-based joint controller (FLJC) to a 6-DOF robotic arm as a color-based sorter system is presented in this study. The robotic arm with FLJC is integrated with a machine vision system that can discriminate different colors. Additionally, the machine vision system composed of Kinect camera and computer were used to extract the coordinates of the gripper and the objects within the image of the workspace. A graphical user interface with an underlying sorting algorithm allows the user to control the sorting process. Once the system is configured, the computed joint angles by FLJC are transmitted serially to the microcontroller. The results show that the absolute error of the gripper coordinates is less than 2 cm and that the machine vision is capable of achieving at least 95% accuracy in proper color discrimination both for first and second level stacked color objects.

Keywords—Color-based sorter; degrees of freedom; fuzzy logic; joint controller; machine vision; robotic arm

I. INTRODUCTION

The development of machines has been a valuable tool ever since the dawn of civilization. Machines had been the humanity's innovative creations whose sole purpose was to achieve efficiency and effectiveness to different tasks that are either routine or almost impossible for humans to do by hand. Machines were meant to be driven by a human operator, until the last century [1] where automation began to be favored by industry, specially deployed in car manufacturing process. This greatly reduced the manpower needed and at the same time was efficient in terms of resources and time. From thereon, autonomous machines came into existence and diverse forms of such machines were developed for specific purposes. One such machine is the autonomous robotic arm whose design was primarily inspired by the human arm. Due to the flexibility that the human arm can do varied tasks, the development of an autonomous robotic arm has been a subject of research [2] since its development in 1960s.

Autonomous robotic arms had numerous advantages as compared to human arm. Robotic arm machines are immune to fatigue and can be made to be invulnerable in wide environment settings. Additionally, it is the most viable alternative when deployed to environments that are too harmful for humans [3] and can be programmed to perform routine

tasks efficiently. Amidst these benefits, a robotic arm is also a complex mechanical machine that exhibits time-varying inertia and friction and as such is more challenging to control by means of classical linear-based controllers. To achieve autonomous operation, the machine must have a controller that is able to sense its current state and decide its course action in much the same way humans decide. Non-classical or intelligent controllers had been developed throughout the years, such as fuzzy logic based controllers [5], [6] that mimics the way humans think, artificial neural network based controllers [7], [8] that emulates the biological human brain, genetic algorithm based controllers [9], [10] inspired by evolutionary processes or hybrid types [11]. One such controller developed in this study is the fuzzy logic-based joint controller (FLJC) [4] that is capable of dealing with system nonlinearities by moving the joints of the robotic arm at proper rate and interval according to the task at hand. Fuzzy logic controllers has been shown as an effective controller in a number of robot systems like the micro soccer robots [12]-[15], micro-golf robot [16], ball-beam balancing robot [17] and simulated and actual robotic arms [4], [6], [18]-[21].

Aside from the controller developed in [4], this study will give emphasis on the integration of the controller with a machine vision system to demonstrate the use of the fuzzy logic controlled autonomous robotic arm system into a color-based sorter system. The machine vision system will be thoroughly discussed as well as the algorithm deployed to perform the sorting process to realize a fully functional color-based sorter. Test results of accuracy of the gripper to move towards the target coordinates as well as the reliability of the machine vision system are laid out and discussed. Lastly, several points are enumerated with regards to the possible improvements that could be made for the system.

II. SYSTEM CONFIGURATION

The color-based sorter system is similar to the configuration in [4] but with the following modifications: 1) the robotic arm's end-effector are embedded with limit switches to improve tactile sensing of the object, 2) the machine vision system is now capable of discriminating at most four different object colors, and 3) the sorter is capable of sorting out stacked objects up to second level. The robotic arm

itself is composed of a 4-DOF M100RAK [22] modular arm attached with 2-DOF gripper [23]. The shoulder, elbow and wrist joints are each mounted with MPU6050 Six-Axis Gyroscope and Accelerometer [24] Inertial Measurement Units (IMU) as sensors to acquire the robotic arm's pose in real time. Attached to the robotic arm's end-effector are the force sensing resistor (FSR) and miniature limit switches as its haptic feedback sensors. The robotic arm's servo motors are controlled directly by the Arduino [25] microcontroller that communicates with the computer. Set atop on the workspace is the Kinect sensor [26] that serves as the main sensory input for machine vision system in the computer. Fig. 1. shows how the components are connected to form the fuzzy logic-controlled color-based sorter.

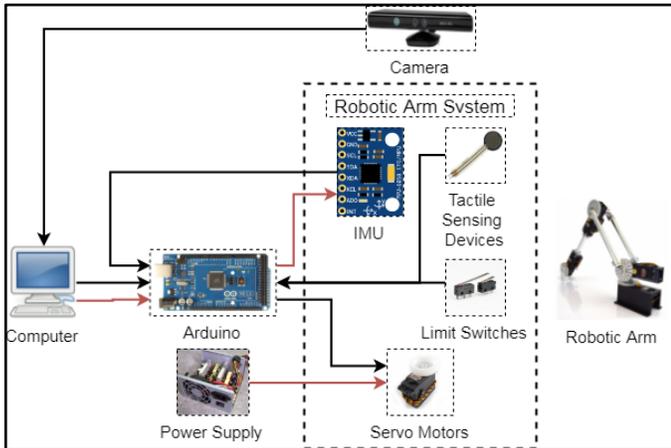


Fig. 1. Architecture of the color-based sorter.

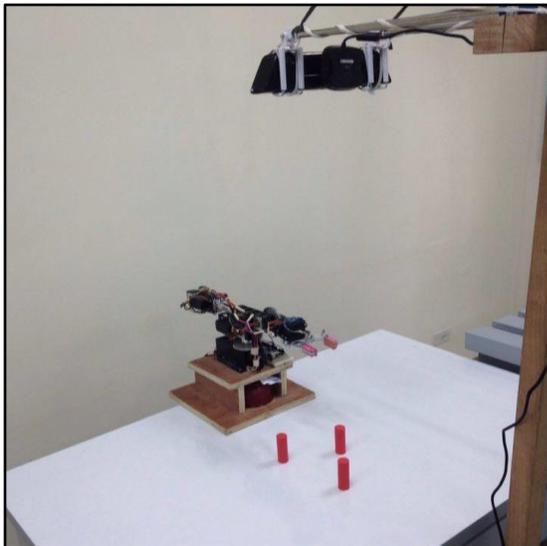


Fig. 2. Configuration of the color-based object sorter.

Shown in Fig. 2 is the hardware configuration used in this study with the robotic arm on the center in front of the cylinder objects and the camera on top. The pertinent dimensions of the workspace are shown in Fig. 3. The study focuses in the application of fuzzy logic-based controller of the robotic arm as well as the algorithm devised to properly sort the cylinder objects in place.

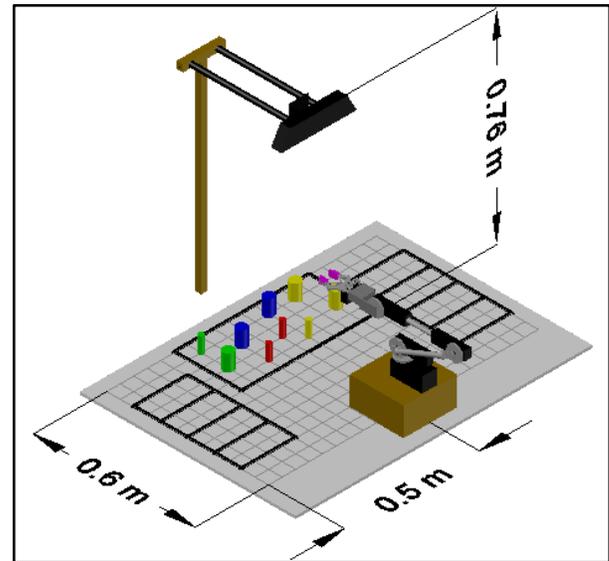


Fig. 3. Workspace dimensions of the color-based sorter.

III. FUZZY LOGIC-BASED JOINT CONTROLLER

The theory of fuzzy sets was first described by Lotfi Zadeh [27] and found its applications as a controller such as for plant processes [28]. Fuzzy sets are an extension of the bi-valued logic in that it can be used to describe half-truth statements to varying degrees. The concept of a fuzzy set can be exploited to emulate the way humans think when in control of a process by employing a human-like language describing how a complex system should be controlled. To achieve a descriptive language for control, a fuzzy logic controller consists of: 1) a fuzzifier block that converts real-world crisp values into fuzzy sets through membership functions, 2) an inference engine that interprets the input fuzzy set based on a set of human-defined language for control known as fuzzy rules to decide the output fuzzy sets, and 3) a defuzzifier block that converts the output fuzzy set back into real-world crisp values [29]. These crisp values are now used to directly control any process variables [5], [11], [17]. Shown in Fig. 4. is the conceptual block diagram of a fuzzy logic controller. The goal of the fuzzy logic controller is to move the end-effector to the desired target as close as possible. The controller is part of a closed-loop system composed of the sensors mounted on the robotic arm, the controller itself and the mechanically actuated robotic arm. The fuzzy logic controller dictates the microcontroller the amount and direction at which the servo motors are to be turned and the microcontroller in turn, through pulse width modulation signals controls the servo motors.

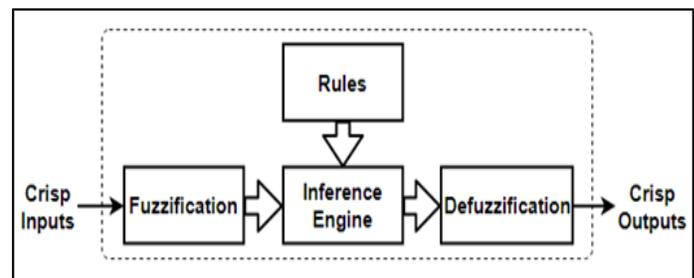


Fig. 4. Fuzzy logic system [5].

A. Input and Output Parameters

The top and side views of the robotic arm with pertinent dimensions are shown in Fig. 5. and 6. Excluding the 2-DOF gripper, there are four (4) joint angles that can be controlled to change the end-effector's position: the base angle (θ_b), shoulder angle (θ_s), elbow angle (θ_e) and wrist angle (θ_w). The fuzzy logic controller must control these joints so that the input errors in x-coordinates (e_x), y-coordinates (e_y) and z-coordinates (e_z) are close to zero as possible.

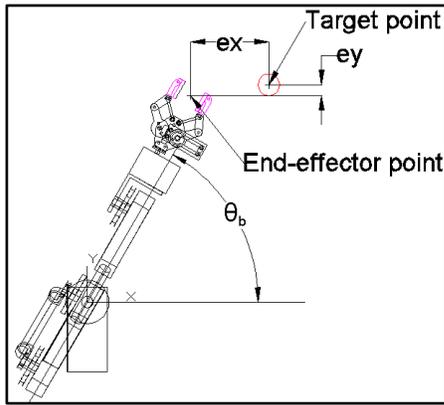


Fig. 5. Top view of the robotic arm relative to target.

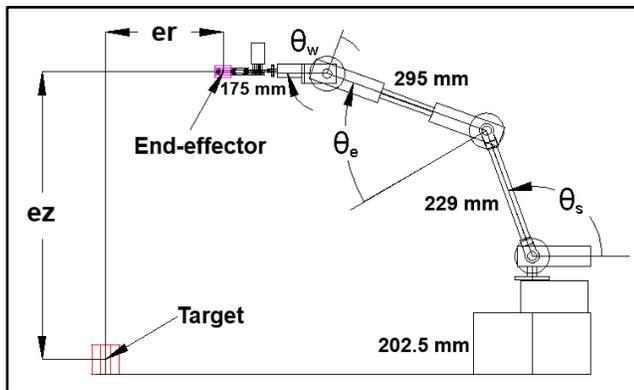


Fig. 6. Side view of the robotic arm relative to target.

The inputs to the controller were chosen according the following criteria: 1) the controller must know how close the end-effector is to the target, and 2) the controller must determine the current pose of the robotic arm to properly move the actuators in the desired direction. With these in mind, listed below are the input parameters for the fuzzy logic controller:

- 1) e_x is error in x-coordinate, defined to be difference between the x-coordinate of the end-effector and the x-coordinate of the target.
- 2) e_y is error in y-coordinate, defined to be difference between the y-coordinate of the end-effector and the y-coordinate of the target.
- 3) e_z is error in z-coordinate, defined to be difference between the z-coordinate of the end-effector and the z-coordinate of the target.
- 4) θ_b is the base angle, defined as the angle between the robotic arm and the x-axis.

- 5) θ_e is the elbow joint angle.
- 6) θ_η is defined as the gripper angle with respect to horizontal.
- 7) θ'_η is defined as the rate of change of gripper angle with respect to horizontal.

The outputs of the fuzzy logic controller are as follows:

- 1) $\Delta\theta_b$ is the change in base joint angle.
- 2) $\Delta\theta_s$ is the change in shoulder joint angle.
- 3) $\Delta\theta_e$ is the change in elbow joint angle.
- 4) $\Delta\theta_w$ is the change in wrist joint angle.

B. Membership Functions

Once the input and output parameters are defined, the appropriate membership functions for each parameter are defined according to the limitations of the robotic arm itself as well as the magnitude of the change produced by each parameter. These membership functions are then tuned and finalized through a series of tests and experimentations [5]. The input membership functions are tuned by considering the sensitivity of the controller to these inputs. In this study, the unit of measurement for the range of values sampled to discrete grades of membership functions for input errors is in millimeters while those for angular displacement are in radians. Trapezoidal membership functions were used at the extreme values of input joint angles to avoid self-collision. Shown in Fig. 7. through Fig. 13. are the membership functions of the seven input parameters. For the sake of brevity, the membership functions are labeled accordingly as follows:

Fuzzy Membership Acronyms:

- | | | |
|------------|-----------------------|---------------------|
| L – left | NL – negative large | P – positive |
| M – middle | N – negative | PL – positive large |
| R – right | Z – zero (negligible) | |



Fig. 7. Membership function for error in x-coordinate.

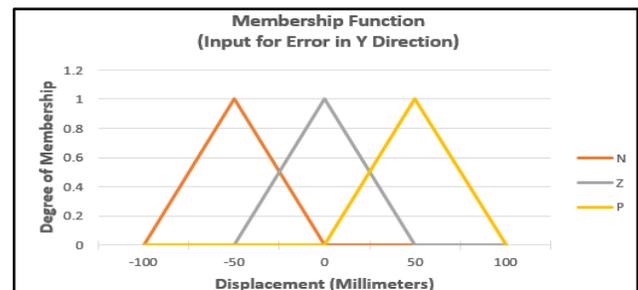


Fig. 8. Membership function for error in y-coordinate.

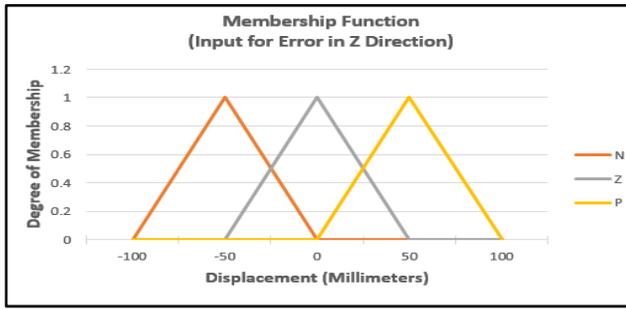


Fig. 9. Membership function for error in z-coordinate.

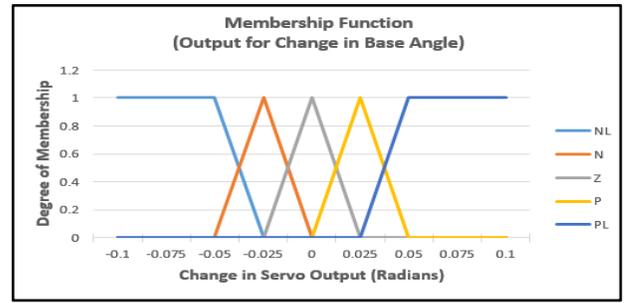


Fig. 14. Membership function for change in base joint angle.

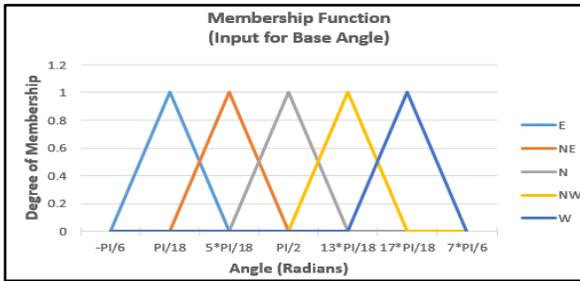


Fig. 10. Membership function for base joint angle.

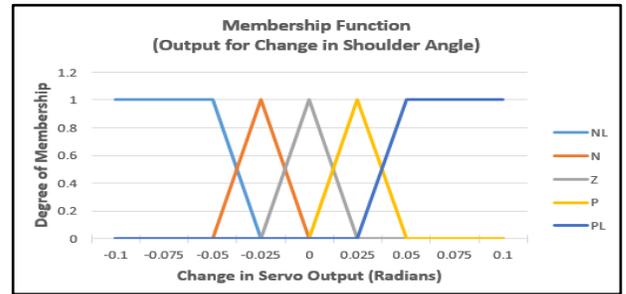


Fig. 15. Membership function for change in shoulder joint angle.

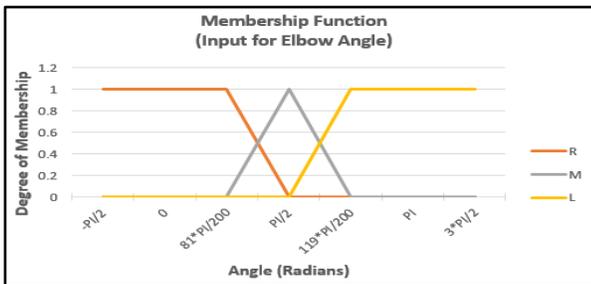


Fig. 11. Membership function for elbow joint angle.

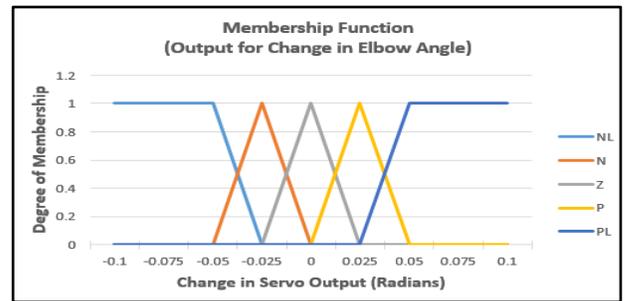


Fig. 16. Membership function for change in elbow joint angle.

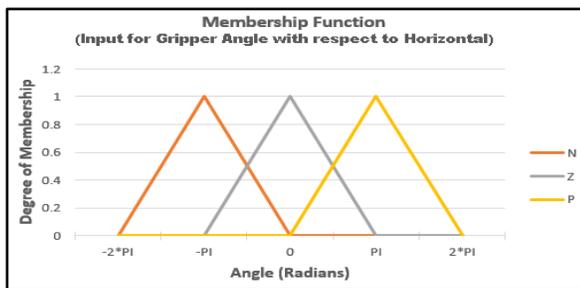


Fig. 12. Membership function for gripper angle with respect to horizontal.

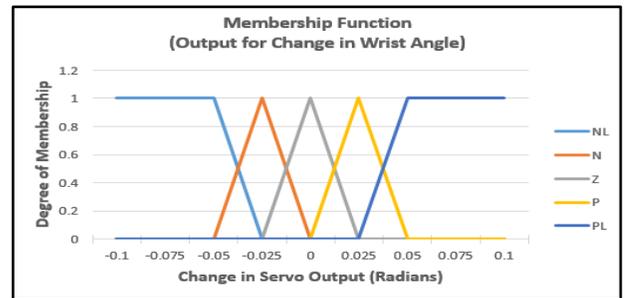


Fig. 17. Membership function for change in wrist joint angle.

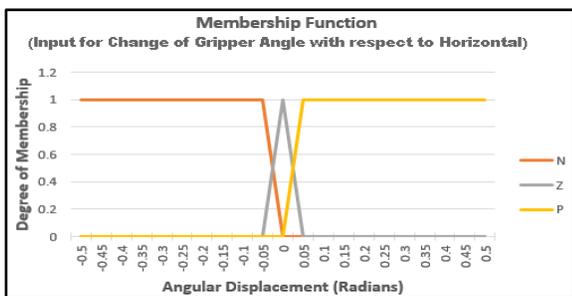


Fig. 13. Membership function for rate of change of gripper angle with respect to horizontal.

Similarly, the output membership functions are tuned by considering the sensitivity of the robotic arm as the joint angles were changed. All output joint angles are specified in units of radians. The defuzzification process used the weighted average method to reduce calculation time in calculating the crisp value. Shown in Fig. 14 through Fig. 17 are the membership functions of the four output joint angles: change in base, shoulder, elbow and wrist joint angles. The same membership labeling scheme applies as defined for the input membership functions.

C. Analysis of the Different Robotic Arm Poses

Once the input and output parameters were determined as well as their respective membership functions, the rules for inference engine are formulated. The rules can be formulated by analyzing the different robotic arm poses possible within the workspace. Of course, there are infinite arm poses that are possible within the workspace so dividing the range of possible values into subsets is necessary and it can be done by the aid of membership functions. The pose of the robotic arm is analyzed by looking at the top and side view of the robotic arm shown in Fig. 18. and 19. The different poses shall be the basis in formulating the fuzzy rules. In general, the rules are to be formulated in such a way that the input errors in x-, y- and z-coordinates are minimized in each iteration.

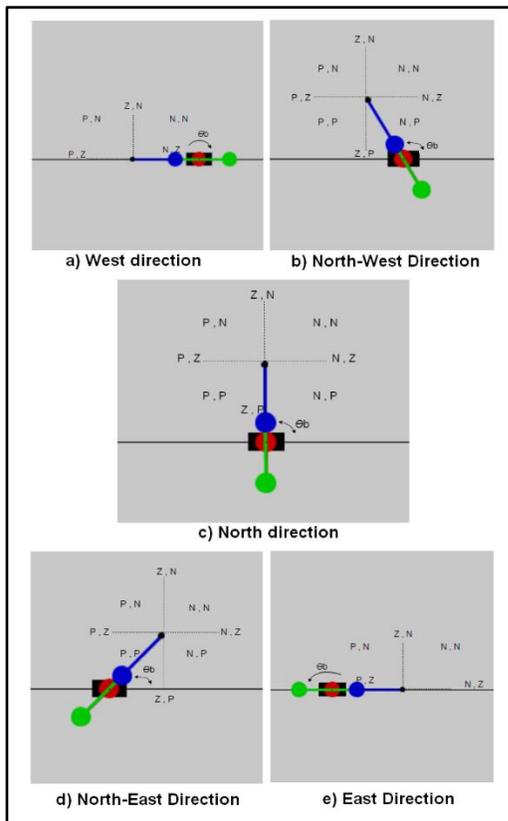


Fig. 18. Top view of possible robotic arm orientation.

The base joint angle can be oriented in five (5) different angle orientations as shown in Fig. 18: West, Northwest, North, Northeast and East. In the same figure, the black rectangle represents the base of the arm, the red link represents the shoulder-to-elbow link, the green link represents the elbow-to-wrist link and the blue link represents the gripper. The initials P, Z, and N corresponds to positive, zero and negative respectively each used to describe the position of the end-effector relative to the target. The symbol Θ_b is the base angle and a pair such as (P,N) denotes that the input errors for x and y coordinates are positive and negative respectively should the target is found at that region relative to the end-effector. Knowing the sign of the input errors will aid on formulating the fuzzy rule at which should the base angle be moved to

minimize the error. In this view, the arm can rotate clockwise or counterclockwise as well as extend or retract its links.

Shown in Fig. 19 are three possible poses when looking at the side view of the robotic arm. The three links form a coupled system that has three (3) degrees of freedom and is more than the degrees of freedom necessary to determine the radius and height of the end-effector. As such, a link can be assumed to be at fixed angle and isolate it from the other two angles. The gripper is chosen to be this link that can be fixed to maintain horizontally level with respect to the ground at all times. Effectively, we could decouple the gripper and write separate fuzzy rules for it apart from the shoulder and elbow joint angles.

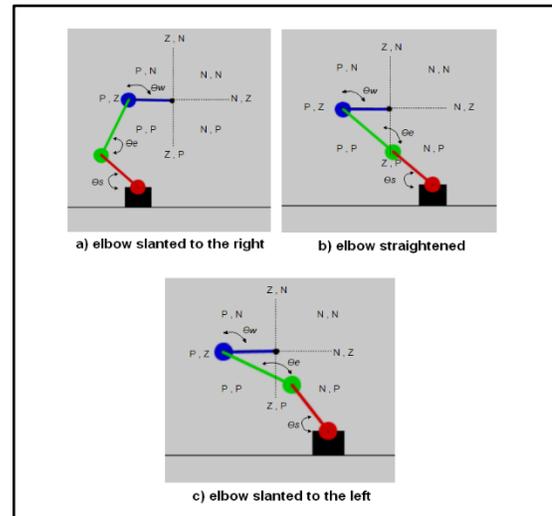


Fig. 19. Side view of possible robotic arm orientation.

D. Fuzzy Rule Formulation

By analyzing the different poses of the robotic arm, the fuzzy rules can now be facilitated by taking note of the input errors as well as their signs. In general, the rule formulation is guided by the control law that all input errors must be minimized and as close to zero as possible. From the analysis of the robotic arm, three (3) different rule blocks can be identified. For instance, if the base angle is pointing in the North direction and the target is present at the (P,N) region then the base angle must rotate counterclockwise and the robot arm must extend forward, to bring the end-effector closer to the target. The beauty of fuzzy logic controller is that you do not have to specify the magnitude explicitly but just the intuition and at which direction should the output parameters move. This analysis is applied to all enumerated poses and the rules formulated can be found on Table I through Table III(a)

Table I pertains to the fuzzy rules for the top view orientation involving the input parameters base joint angle, error in x-coordinate, error in y-coordinate and output parameter change in base joint angle. Table II contains the rules for the side view orientation involving the input parameters elbow joint angle, error in y-coordinate, error in z-coordinate and output parameters change in shoulder and elbow angles. Lastly, Table III is a list that controls how the gripper angle must maintain horizontally level at all times.

TABLE I. FUZZY RULES FOR BASE JOINT ANGLE

	Input: base joint angle (θ_b), error x (e_x), error y (e_y) Output: change in base joint angle ($\Delta\theta_b$)
1	If θ_b is E and e_x is N and e_y is N then $\Delta\theta_b$ is P
2	If θ_b is E and e_x is N and e_y is Z then $\Delta\theta_b$ is Z
3	If θ_b is E and e_x is N and e_y is P then $\Delta\theta_b$ is N
4	If θ_b is E and e_x is Z and e_y is N then $\Delta\theta_b$ is P
5	If θ_b is E and e_x is Z and e_y is Z then $\Delta\theta_b$ is Z
6	If θ_b is E and e_x is Z and e_y is P then $\Delta\theta_b$ is N
7	If θ_b is E and e_x is P and e_y is N then $\Delta\theta_b$ is P
8	If θ_b is E and e_x is P and e_y is Z then $\Delta\theta_b$ is Z
9	If θ_b is E and e_x is P and e_y is P then $\Delta\theta_b$ is N
10	If θ_b is NE and e_x is N and e_y is N then $\Delta\theta_b$ is Z
11	If θ_b is NE and e_x is N and e_y is Z then $\Delta\theta_b$ is N
12	If θ_b is NE and e_x is N and e_y is P then $\Delta\theta_b$ is N
13	If θ_b is NE and e_x is Z and e_y is N then $\Delta\theta_b$ is P
14	If θ_b is NE and e_x is Z and e_y is Z then $\Delta\theta_b$ is Z
15	If θ_b is NE and e_x is Z and e_y is P then $\Delta\theta_b$ is P
16	If θ_b is NE and e_x is P and e_y is N then $\Delta\theta_b$ is P
17	If θ_b is NE and e_x is P and e_y is Z then $\Delta\theta_b$ is P
18	If θ_b is NE and e_x is P and e_y is P then $\Delta\theta_b$ is Z
19	If θ_b is N and e_x is N and e_y is N then $\Delta\theta_b$ is N
20	If θ_b is N and e_x is N and e_y is Z then $\Delta\theta_b$ is N
21	If θ_b is N and e_x is N and e_y is P then $\Delta\theta_b$ is N
22	If θ_b is N and e_x is Z and e_y is N then $\Delta\theta_b$ is Z
23	If θ_b is N and e_x is Z and e_y is Z then $\Delta\theta_b$ is Z
24	If θ_b is N and e_x is Z and e_y is P then $\Delta\theta_b$ is Z
25	If θ_b is N and e_x is P and e_y is N then $\Delta\theta_b$ is P
26	If θ_b is N and e_x is P and e_y is Z then $\Delta\theta_b$ is P
27	If θ_b is N and e_x is P and e_y is P then $\Delta\theta_b$ is P
28	If θ_b is NW and e_x is N and e_y is N then $\Delta\theta_b$ is N
29	If θ_b is NW and e_x is N and e_y is Z then $\Delta\theta_b$ is N
30	If θ_b is NW and e_x is N and e_y is P then $\Delta\theta_b$ is Z
31	If θ_b is NW and e_x is Z and e_y is N then $\Delta\theta_b$ is N
32	If θ_b is NW and e_x is Z and e_y is Z then $\Delta\theta_b$ is Z
33	If θ_b is NW and e_x is Z and e_y is P then $\Delta\theta_b$ is P
34	If θ_b is NW and e_x is P and e_y is N then $\Delta\theta_b$ is Z
35	If θ_b is NW and e_x is P and e_y is Z then $\Delta\theta_b$ is P
36	If θ_b is NW and e_x is P and e_y is P then $\Delta\theta_b$ is P
37	If θ_b is W and e_x is N and e_y is N then $\Delta\theta_b$ is N
38	If θ_b is W and e_x is N and e_y is Z then $\Delta\theta_b$ is Z
39	If θ_b is W and e_x is N and e_y is P then $\Delta\theta_b$ is P
40	If θ_b is W and e_x is Z and e_y is N then $\Delta\theta_b$ is N
41	If θ_b is W and e_x is Z and e_y is Z then $\Delta\theta_b$ is Z
42	If θ_b is W and e_x is Z and e_y is P then $\Delta\theta_b$ is P
43	If θ_b is W and e_x is P and e_y is N then $\Delta\theta_b$ is N
44	If θ_b is W and e_x is P and e_y is Z then $\Delta\theta_b$ is Z
45	If θ_b is W and e_x is P and e_y is P then $\Delta\theta_b$ is P

TABLE II. FUZZY RULES FOR SHOULDER AND ELBOW JOINT ANGLES

	Input: elbow joint angle (θ_e), error y (e_y), error z (e_z) Output: change in shoulder joint angle ($\Delta\theta_s$), change in elbow joint angle ($\Delta\theta_e$)
1	If θ_e is R and e_y is N and e_z is N then $\Delta\theta_s$ is P and $\Delta\theta_e$ is P
2	If θ_e is R and e_y is N and e_z is Z then $\Delta\theta_s$ is P and $\Delta\theta_e$ is Z
3	If θ_e is R and e_y is N and e_z is P then $\Delta\theta_s$ is P and $\Delta\theta_e$ is N
4	If θ_e is R and e_y is Z and e_z is N then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is P
5	If θ_e is R and e_y is Z and e_z is Z then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is Z
6	If θ_e is R and e_y is Z and e_z is P then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is N

7	If θ_e is R and e_y is P and e_z is N then $\Delta\theta_s$ is N and $\Delta\theta_e$ is P
8	If θ_e is R and e_y is P and e_z is Z then $\Delta\theta_s$ is N and $\Delta\theta_e$ is Z
9	If θ_e is R and e_y is P and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
10	If θ_e is M and e_y is N and e_z is N then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is Z
11	If θ_e is M and e_y is N and e_z is Z then $\Delta\theta_s$ is N and $\Delta\theta_e$ is Z
12	If θ_e is M and e_y is N and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
13	If θ_e is M and e_y is Z and e_z is N then $\Delta\theta_s$ is P and $\Delta\theta_e$ is Z
14	If θ_e is M and e_y is Z and e_z is Z then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is Z
15	If θ_e is M and e_y is Z and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
16	If θ_e is M and e_y is P and e_z is N then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
17	If θ_e is M and e_y is P and e_z is Z then $\Delta\theta_s$ is N and $\Delta\theta_e$ is Z
18	If θ_e is M and e_y is P and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
19	If θ_e is L and e_y is N and e_z is N then $\Delta\theta_s$ is P and $\Delta\theta_e$ is N
20	If θ_e is L and e_y is N and e_z is Z then $\Delta\theta_s$ is P and $\Delta\theta_e$ is N
21	If θ_e is L and e_y is N and e_z is P then $\Delta\theta_s$ is P and $\Delta\theta_e$ is N
22	If θ_e is L and e_y is Z and e_z is N then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
23	If θ_e is L and e_y is Z and e_z is Z then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
24	If θ_e is L and e_y is Z and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
25	If θ_e is L and e_y is P and e_z is N then $\Delta\theta_s$ is Z and $\Delta\theta_e$ is N
26	If θ_e is L and e_y is P and e_z is Z then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N
27	If θ_e is L and e_y is P and e_z is P then $\Delta\theta_s$ is N and $\Delta\theta_e$ is N

TABLE III. FUZZY RULES FOR WRIST ANGLE

	Input: gripper angle w.r.t. horizontal (θ_η), rate of change of eta ($\Delta\theta_\eta$) Output: change in wrist joint angle ($\Delta\theta_w$)
1	If θ_η is N and $\Delta\theta_\eta$ is N then $\Delta\theta_w$ is N
2	If θ_η is N and $\Delta\theta_\eta$ is Z then $\Delta\theta_w$ is N
3	If θ_η is N and $\Delta\theta_\eta$ is P then $\Delta\theta_w$ is Z
4	If θ_η is Z and $\Delta\theta_\eta$ is N then $\Delta\theta_w$ is P
5	If θ_η is Z and $\Delta\theta_\eta$ is Z then $\Delta\theta_w$ is Z
6	If θ_η is Z and $\Delta\theta_\eta$ is P then $\Delta\theta_w$ is N
7	If θ_η is P and $\Delta\theta_\eta$ is N then $\Delta\theta_w$ is Z
8	If θ_η is P and $\Delta\theta_\eta$ is Z then $\Delta\theta_w$ is P
9	If θ_η is P and $\Delta\theta_\eta$ is P then $\Delta\theta_w$ is P

IV. MACHINE VISION SYSTEM

The machine vision system is composed of the camera as its sensory vision input and the computer as an image processing unit. The camera is the Kinect sensor [26] capable of providing not only colored images as well as image depth data. The image depth data was used to properly determine the height of the detected objects and consequently the stacking level of the cylinder objects. Furthermore, the depth data was used to filter out the white platform background by exploiting the fact that its distance is farther away from the camera itself. This method is referred to as depth masking.

The computer uses the Java-based Processing [30] software environment that provides the interfacing between the devices attached to it such as the Kinect sensor and the Arduino. Processing-based OpenCV [31] and SimpleOpenNI [32] libraries were used for the software development. The OpenCV library provided the tools to filter the image based on Hue-Saturation-Value (HSV) as well as detect the presence of blobs and their respective coordinates. On the other hand, the SimpleOpenNI library allows the system to communicate with the Kinect sensor to get the RGB image and depth data. Fig. 20. shows how the raw RGB image is eventually filtered out to keep the blue cylinder objects. The binarized image on the

right was processed further to detect the blobs present and store their coordinates for sorting purposes.

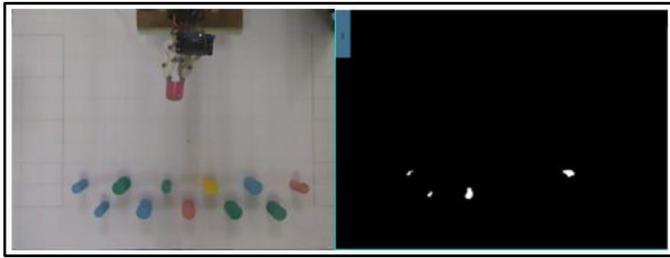


Fig. 20. On the left: actual gripper and cylinder objects as seen by the camera atop; on the right: filtered image showing the blue cylinder objects.

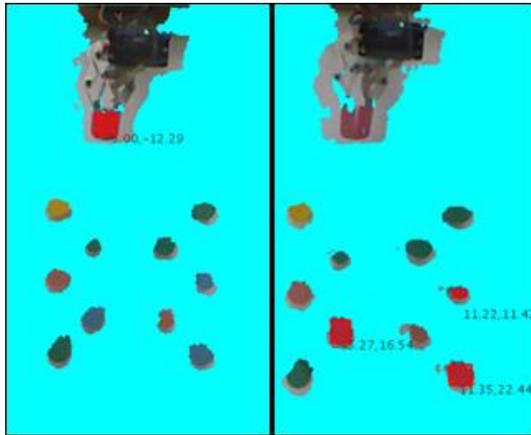


Fig. 21. Detection of gripper (left) and blue cylinder objects (right).

This filtering, detection and coordinate acquisition processes were done for other colors as well as for the gripper. The gripper is colored differently from the possible colors of the cylinder objects to properly recognize and locate the coordinates of the gripper itself. Shown in Fig. 21. is the results of detection and extraction of the coordinates of the gripper and the blue cylinder objects. It is worth mentioning that the gripper has two distinguishable shapes if it is wide open. In such a case, the reported coordinates of the gripper are found by calculating the centroid of the two separately detected gripper objects.

V. GRAPHICAL USER INTERFACE AND SORTING ALGORITHM

The coordinates of the gripper and the objects are measured relative to the origin point at the center of the region of interest as well as their respective heights relative to the white background platform. Together with the measured joint angles of the robotic arm, these values constitute the input variables for the sorting algorithm. The process begins by the user configuring the system through a designed graphical user interface (GUI) shown in Fig. 22. The user selects which object colors are to be sorted first according to the priority the user wishes. The user can also choose how the objects are to be sorted and decided where each color should land on pre-determined locations. After configuration, the “Sort” button can be pressed to begin the sorting process.



Fig. 22. The graphical user interface for the color-based sorter.

Pressing the “Sort” button invokes the sorting algorithm. The sorting algorithm first applies depth masking to differentiate depth levels among the platform, the objects and the gripper. The next process applies HSV-based color filtering to detect the presence of the objects in a specific color. The sequence at which colors are detected is determined by the priority configuration set by the user. As objects are detected for each color, the coordinates of the objects are acquired and stored to their respective buffers. After all object colors are found, the coordinates of the gripper are acquired. A hysteresis function is applied to the obtained coordinates to eliminate the sudden changes in coordinates due to noise.

Now that the coordinates of all objects are found, the gripper coordinates and the first object to be sorted or targeted is fed to the FLJC. There are several rules that determine which object should be fed to the FLJC: 1) based from the priority set by the user, and 2) the distance of the target object from the gripper. Since it is possible that there are multiple objects of the same color, the sorting algorithm would pick the target object with the minimal distance from the gripper. Should objects have same color and same distance from the gripper, the object with least change in base angle needed to reach will be picked up first. In this manner, the priority of which object should be picked up is resolved. Now that the target object is determined, the FLJC applies the appropriate changes to the joint angle needed to further minimize the distance between the gripper and the target. The machine vision will then locate the gripper coordinates and feed it back to the FLJC. This process repeats until the gripper coordinates is sufficiently coincident with that of the target object. It is also worth mentioning that a parallax error will be imposed upon the gripper coordinates relative to the platform depending on its height and location. To mitigate the parallax shift, a proper coordinate transformation is applied to the gripper coordinates before fed to the FLJC. For the coordinates (x, y) and height z as seen by the camera, the actual coordinates (x', y') are found to be:

$$x' = (x - x_c)(1 - p(z + B)/h) + x_c \quad (1)$$

$$y' = (y - y_c)(1 - p(z + B)/h) + y_c \quad (2)$$

where (x_c, y_c) are the coordinates of the center of ROI relative to the base of the robotic arm, p is the parallax factor, h is the camera height and B is the base height of the robotic arm.

As the gripper closes in to the object, the gripper is closed to grip the target object. The attached tactile sensors will determine if the object was indeed grasped. If the object is found to be grasped, a predetermined sequence of robotic arm

movements will place the object on its specific location according the configuration set by the user. Once the object is placed, the robotic arm goes into its home position. The sorting algorithm will feed the gripper coordinates and the next target object. This process will repeat until all such objects are placed to their appropriate destinations. During the sorting process, the user cannot reconfigure the sorter until it is finished but may press the emergency button to terminate.

VI. DATA AND RESULTS

A. Robotic Arm Simulator

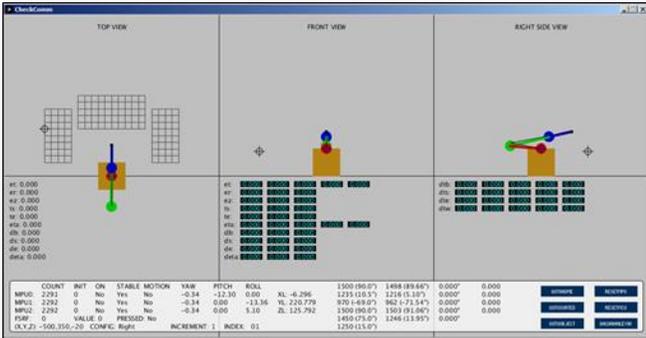


Fig. 23. Robotic arm simulator.

The robotic arm simulator shown in Fig. 23. is an improvement over that shown in [4]. The simulator can function as a monitoring tool to show the actual robotic arm pose in real time as well as provide a visualization of the controller in action. Furthermore, important parameters are indicated below to guide in the tuning of the fuzzy membership functions and several buttons that are programmed to move the robotic arm in a pre-determined sequence such as going to its home position and placing an object to designated areas. Once the actual robotic arm's response is satisfactory, the tuned fuzzy membership functions are transferred to a final program to be integrated with the sorting algorithm and a designed GUI.

B. Robotic Arm Movement

To test the accuracy of the fuzzy logic controller, the robotic arm was stretched forward along the Y direction. The plots of actual robotic arm end-effector coordinates plotted against the desired y-coordinate are shown in Fig. 24. through Fig. 26. A comparison was made against the inverse kinematic implementation. From Fig. 24. the fuzzy logic controller had lesser sideway excursions as compared to inverse kinematics implementation. The fuzzy logic controller was able to follow closely the ideal y-coordinate value as compared to inverse kinematic implementation shown in Fig. 25. The inverse kinematic implementation is found below the required y-value because of the weights of the robotic arm links. This effect is more pronounced as the height has significantly drooped shown in Fig. 26. Again, the moment due to the weight of extending arm is increasing as the y-coordinate increases. The fuzzy logic controller on the other hand managed to maintain a satisfactory level that is within 5 mm from the ideal height of 100 mm.

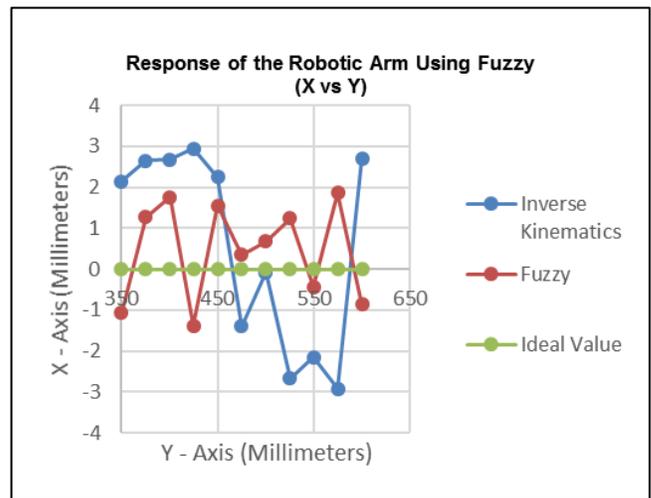


Fig. 24. X-Y movement response.

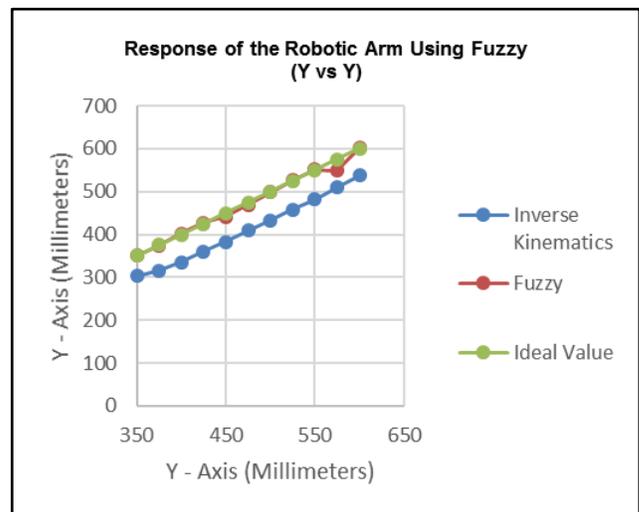


Fig. 25. Y-Y movement response.

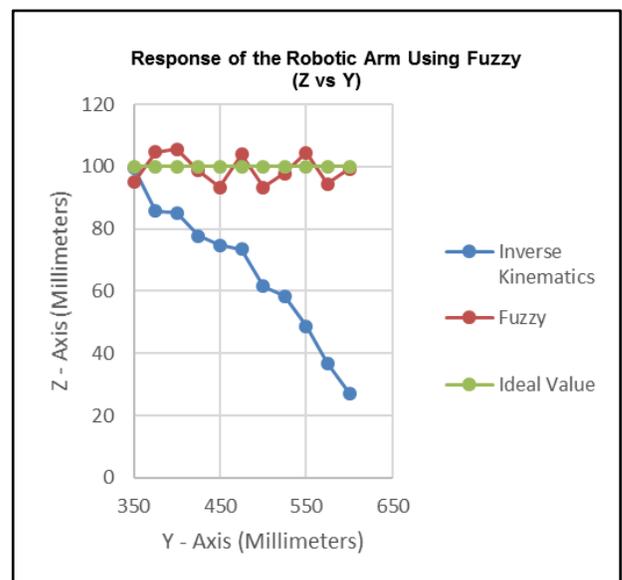


Fig. 26. Z-Y movement response.

C. Accuracy of the End Effector

The end-effector’s accuracy is tested by feeding the FLJC with an ideal coordinate coincident to the intersection of gridlines on the platform. By marking of the platform beneath the gripper, the distance between x-, y- and z-coordinates are obtained for at least 30 trials.

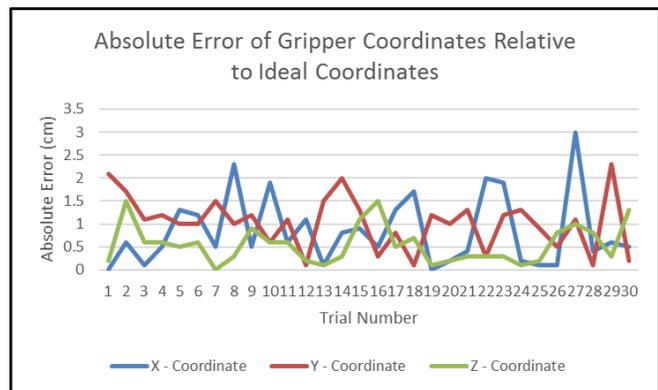


Fig. 27. Measure of end-effector’s absolute error before pickup.

These coordinates are randomly picked from the workspace area where the objects to be sorted are placed. The differences are measured and plotted as shown in Fig. 27. The average values of 0.8, 1 and 0.6 cm for absolute errors in x-, y- and z-coordinates were calculated for the end-effector relative to origin, respectively. On average, the end-effector coordinates were accurate enough to allow tolerance of 2 cm radius.

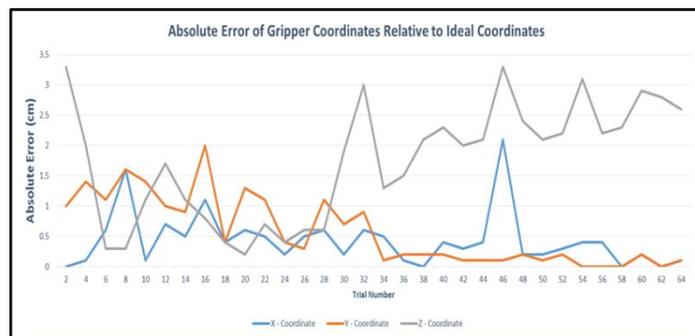


Fig. 28. Measure of end-effector’s absolute error after placement.

Fig. 28 shows the plot of absolute error of gripper coordinates in x-, y- and z-coordinates as it moved towards the pre-determined coordinates as destination for sorted objects. The average values of 0.45, 0.92 and 1.77 cm for absolute errors in x-, y- and z-coordinates were calculated for the end-effector relative to origin respectively. On average, the end-effector coordinates were accurate enough to allow tolerance of 2 cm radius.

D. Reliability of Machine Vision System

Two tests are performed to determine the reliability of the machine vision system: 1) accuracy in acquisition of coordinates, and 2) accuracy in color discrimination of objects both on the platform level or the second stack level. Twelve (12) trials each containing at least 10 colored objects are to be detected and the coordinates acquired. The average error for each trial is shown in Table IV. On average, the absolute error

for the overall test of the vision system was about 0.19 cm for x-coordinates and about 1.41 cm for the y-coordinates, well within 2 cm tolerance value of accuracy.

TABLE IV. AVERAGE ERROR FOR OBJECT COORDINATES

Trial No.	Average Error		Trial No.	Average Error	
	x	y		x	y
1	0.16	0.76	7	0.59	1.83
2	0	1.16	8	0.25	1.38
3	0.22	1.06	9	0.22	1.33
4	0.13	1.73	10	0.08	1.3
5	0.14	1.67	11	0.07	1.45
6	0.15	2.03	12	0.21	1.22

For the second test, the ability of the machine vision system to properly discriminate colors are tested for the same number of trial with same number of objects. A summary of confusion matrix was constructed as shown in Table V for all tested objects on the first level. The data shows that the colors blue, green and yellow were detected 100% accurately. Notice also that there is a 100% precision for colors blue, green and yellow, and 96% for red. The red color was found to have the least among them all because of the proximity of the red color to the gripper’s color, making it hard to delineate in HSV space. Nevertheless, the gripper is never mistakenly detected as an object.

TABLE V. SUMMARY OF CONFUSION MATRIX FOR 1ST LEVEL OBJECTS

Color	Blue	Green	Red	Yellow
Accuracy	100.00%	100.00%	81.54%	100.00%
True Positive Rate	100.00%	100.00%	100.00%	100.00%
False Positive Rate	0.00%	0.00%	0.94%	0.00%
True Negative Rate	100.00%	100.00%	99.06%	100.00%
False Negative Rate	0.00%	0.00%	0.00%	0.00%
Precision	100.00%	100.00%	96.00%	100.00%

Additional objects were stacked on top of the first level making it a second level stacked object. Similar test for the first level were conducted to test the ability of the system to properly discriminate stacked colored objects. Table VI shows the summary of confusion matrix for detection of objects on the second stack level. The data shows that the green and red were detected 100% accurately. Notice also that there is a 100% precision for colors green and red and about 97% for red and yellow.

TABLE VI. SUMMARY OF CONFUSION MATRIX FOR 2ND LEVEL OBJECTS

Color	Blue	Green	Red	Yellow
Accuracy	96.97%	100.00%	100.00%	96.88%
True Positive Rate	100.00%	100.00%	100.00%	100.00%
False Positive Rate	4.17%	0.00%	0.00%	4.55%
True Negative Rate	95.83%	100.00%	100.00%	95.45%
False Negative Rate	0.00%	0.00%	0.00%	0.00%
Precision	90.00%	100.00%	100.00%	90.91%

VII. CONCLUSION AND RECOMMENDATION

The study was successful in integrating the autonomous robotic arm with fuzzy logic-based joint controller (FLJC) with a machine vision system capable of accurate color discrimination into a color-based sorter system. An improved robotic arm simulator made it possible to tune the membership functions and see the actual effect on the robotic arm's response. Additionally, the end-effector is well accurate enough to have less than 2 cm absolute error. The coordinates of the different target objects with different colors and stacking levels of up to second level as well as the coordinates of the gripper were successfully acquired by means of Processing with SimpleOpenNI and OpenCV library. The overall accuracy of the machine vision system shows that it has the same precision as the end-effector and is at least 95% accurate in properly discriminating colored objects. This extended study has demonstrated that it is capable of sorting even second level stacked color objects. The utilization of the depth data made it possible to determine the height of the colored object in question.

As for improvement, the researchers aim to introduce different controllers such as the hybrid neuro-fuzzy system and genetic algorithm to aid in fine tuning the membership functions and fuzzy rule formulation. Furthermore, the machine vision system can be further improved by applying more advanced color clustering techniques which will eventually allow more colors to be discriminated without ambiguity.

ACKNOWLEDGMENT

The authors would like to thank their family and friends, De La Salle University – Manila (DLSU), De La Salle University – Laguna Campus – Biñan Laguna (DLSU-Laguna) and Department of Science and Technology – Engineering Research and Development for Technology (DOST-ERDT) for funding and helping us to finish this study.

REFERENCES

- [1] J. Rifkin, *The end of work: the decline of the global labor force and the dawn of the post-market era*. New York: Jeremy P. Tacher, 2004.
- [2] Adelhard Beni Rehiara (2011). "Kinematics of AdeptThree Robot Arm," Robot Arms, Prof. Satoru Goto (Ed.), InTech, DOI: 10.5772/17732. Available from: <https://www.intechopen.com/books/robot-arms/kinematics-of-adeptthree-robot-arm>
- [3] OSHA, "OSHA Technical Manual (OTM) | Section IV: Chapter 4 - Industrial Robots and Robot System Safety", 2016. [Online]. [Accessed: 13- Feb- 2016].
- [4] D. D. Ligutan, L. J. S. Cruz, M. C. D. P. Del Rosario, J. N. S. Kudhal, A. C. Abad and E. P. Dadios, "Design and implementation of a fuzzy logic-based joint controller on a 6-DOF robot arm with machine vision feedback," 2017 Computing Conference, London, 2017, pp. 249-257.
- [5] E. P. Dadios, D. J. Williams, "Multiple fuzzy logic systems: A controller for the flexible pole-cart balancing problem," Proc. of the IEEE Robotics and Automation International Conference, Minneapolis, Minnesota USA, April 24-26, 1996. ICRA 1996: 2276-2281.
- [6] A. A. Khalate, G. Leena, G. Ray, "An Adaptive Fuzzy Controller for Trajectory Tracking of Robot Manipulator," Intelligent Control and Automation, 2011, 364-370
- [7] A. R. F. Quiros, A. C. Abad, and E. P. Dadios, "Object locator and collector robotic arm using artificial neural networks," International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, December 2015.
- [8] X. Li, H. Wang, X. Lu, Y. Liu, Z. Chen, and M. Li, "Neural network method for robot arm of service robot based on D-H model," 2017 Chinese Automation Congress (CAC), 2017.
- [9] Elmer P. Dadios, Patrick S. Fernandez, and David J. Williams, "Genetic Algorithm On Line Controller for the Flexible Inverted Pendulum Problem," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.10 No.2, 2006
- [10] S. Števo, I. Sekaj, and M. Dekan, "Optimization of Robotic Arm Trajectory Using Genetic Algorithm," IFAC Proceedings Volumes, vol. 47, no. 3, pp. 1748–1753, 2014.
- [11] A. R. F. Quiros, A. Abad, R. A. Bedruz, A. C. Uy, and E. P. Dadios, "A genetic algorithm and artificial neural network-based approach for the machine vision of plate segmentation and character recognition," 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, December 2015.
- [12] R. R. P. Vicerra, K. K. A. David, A. R. D. Cruz, E. A. Roxas, K. B. C. Simbulan, A. A. Bandala, and E. P. Dadios, "A multiple level MIMO fuzzy logic based intelligence for multiple agent cooperative robot system," TENCON 2015 - 2015 IEEE Region 10 Conference, November 2015.
- [13] E. Maravillas, E.P. Dadios, "Hybrid Fuzzy Logic Strategy for Soccer Robot Game," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol 8 No. 1, pp 65-71, FUJI Technology Press, January 2004.
- [14] O. F. D. Astilla, J. S. Guerrero, R. S. S. Mendoza, M. T. P. Roxas, A. C. T. Sy, R. R. P. Vicerra, E. P. Dadios, A. R. D. Cruz, E. A. Roxas, and A. A. Bandala, "Obstacle avoidance of hybrid mobile-quadrotor vehicle with range sensors using fuzzy logic control," 2015 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), Cebu City, December 2015.
- [15] A. Abad, G. Abulencia, W. Pacer, E. Dadios, N. Gunay, "Soccer Robot Shooter Strategy," National Electrical, Electronics, and Computing Conference 2009, Science Discovery Center SM Mall of Asia – December 9-11, 2009
- [16] K.G. B. Leong, S. W. Licarte, G. M. S. Oblepias, E. M. J. Palomado, and E.P. Dadios N. G. Jabson, "The Autonomous Golf Playing Micro Robot: With Global Vision And Fuzzy Logic Controller," International Journal on Smart Sensing and Intelligent Systems, vol. 1, no. 4, pp. 824-841, December 2008.
- [17] E. P. Dadios, R. Baylon, R. De Guzman, A. Floren Lee, and Z. Zulueta, "Vision guided ball-beam balancing system using fuzzy logic," in Industrial Electronics Society, 2000. IECON 2000. 26th Annual Conference of the IEEE 2000, pp. 1973-1978 vol.3.
- [18] Zheng Li, Ruxu Du, Haoyong Yu and Hongliang Ren, "The Inverse Kinematics Solution of a 7 DOF Robotic Arm Using Fuzzy Logic". 7th IEEE Conference on Industrial Electronics and Application. 2012.
- [19] Vishank Bhatia., V. Kalaichelvi, Karthikeyan R., "Application of a Novel Fuzzy Logic Controller for a 5-DOF Articulated Anthropomorphic Robot," IEEE international conference on Research in Computational Intelligence and Communication Networks, Kolkotta, Nov 20-22, 2015.
- [20] M. Mirzadeh, M. Khezri, J. Mahmoodi, H. Karbasi, "Design Adaptive Fuzzy Inference Controller for Robot Arm", IJITCS, vol.6, no.9, pp.66-73, 2014.
- [21] J. R. Sanchez-Lopez, A. Marin-Hernandez and E. R. Palacios-Hernandez, "Visual Detection, Tracking and Pose Estimation of a Robotic Arm End Effector", in Proc. of ROSSUM 2011, Xalapa, Ver., Mexico, June 27-28, pp 41-48, 2011.
- [22] "RobotShop | Robot Store | Robots | Robot Parts | Robot Kits | Robot Toys," RobotShop Blog. [Online]. Available: <http://www.robotshop.com/>. [Accessed: 13-May-2016].
- [23] "Store | Robots | 3D Printers | CNC | Telepresence Robots | R&D," RoboSavvy. [Online]. Available: <https://robosavvy.com/store/>. [Accessed: 23-June-2016].
- [24] "MPU-6050 | TDK," InvenSense. [Online]. Available: <https://www.invensense.com/products/motion-tracking/6-axis/mpu-6050/>. [Accessed: 13-October-2017].

- [25] "Arduino - Home," *Arduino Reference*. [Online]. Available: <https://www.arduino.cc/>. [Accessed: 13-May-2016].
- [26] "Kinect Sensor," *About Processes and Threads (Windows)*. [Online]. Available: <https://msdn.microsoft.com/en-us/library/hh438998.aspx>. [Accessed: 07-March-2018].
- [27] L. A Zadeh, "Fuzzy Sets," *Information and Control*, Vol. 8, pp. 338-353, 1965.
- [28] E. H. Mamdani. "Application of fuzzy algorithms for control of simple dynamic plant". *Proceedings of the Institution of Electrical Engineers*. 121 (12): 1585-1588, 1974.
- [29] S. Sarkar and A. Basu, "Reasoning with uncertainty-Fuzzy Reasoning," Department of Computer Science & Engineering, Indian Institute of Technology, Module 11, Version 1 CSE IIT, Kharagpur, 2009.
- [30] C. Reas and B. Fry, "Processing: programming for the media arts", *Journal AI & Society*, volume 20(4), pp. 526-538, 2006.
- [31] Atduskreg, "atduskreg/opencv-processing," *GitHub*, 22-May-2017. [Online]. Available: <https://github.com/atduskreg/opencv-processing>. [Accessed: 07-March-2018].
- [32] Wexstorm, "wexstorm/simple-openni," *GitHub*. [Online]. Available: <https://github.com/wexstorm/simple-openni>. [Accessed: 07-March-2018].