FUZZY SYSTEM AS FOUR-JOINT ROBOT MOVEMENT CONTROL FOR MOVING GOODS BASED ON TIME AND OBJECT COLOR

Emmanuel Agung Nugroho, Nanang Roni Wibowo, Afzeri, Riqqi Rizalludin, Marwan Ruswanda
Teknologi Rekayasa Mekatronika, Politeknik Enjinering Indorama
e-mail: emmanuel.agung@pei.ac.id

Abstract

Technology Disruption has driven the acceleration of the industrial era 4.0 which is based on the speed of data-based systems and artificial intelligence. Another impact is intelligent robots as a substitute for human performance, because humans experience fatigue and decrease accuracy with increasing working time. Robots are an alternative whose presence cannot be avoided, therefore the robot's work system must be improved to optimize its speed and accuracy. One of the implementations of the robot is as a product moving machine from the initial coordinates (Pick) to the destination coordinates (Place). This study uses a four-joint robot (4 DOF) with a product transfer mechanism that can be adjusted based on the input priority that has been set using the Fuzzy Logic Control (FLC) method which is implemented using an Arduino Mega 2560 microcontroller, and the robot used is Dobot Magician. There are 2 input variables used, namely the mass of the object detected by the Weight sensor Load Cell + HX711 module and the color of the object detected using the TCS3200 color sensor. The output variable is the movement of the robot to the Place which can be analyzed by calculating the position matrix using the Matlab program.

Keyword : *Four-joint robot, DOBOT Magician, Fuzzy Logic Control, Matlab, Arduino Mega2560*

1. INDTODUCTION

1.1 Dobot Magician

The use of robots has increased exponentially over the last few decades. Extensive research has taken place in the area of robot design and control to make the robot intelligent and to improve the operating performance of the robot. This study discusses Dobot Magician as one of the robot applications in industry. Dobot Magician is a four-joint manipulator (4 DOF) with 3 DOF of which are on the Base, Shoulder and Elbow with joint motors in the form of three stepper motors and 1 DOF on the wrist which is driven by a servo motor. The end effector consists of a pneumatic vacuum suction cup mechanism, gripper, laser printing, 3D printing hot-end and a pen holder for drawing graphs [1]–[4]. Dobot Magician's specifications are delivered in Table 1.

Makalah dikirim 4 Agustus 2022; Revisi 24 Agustus 2022; Diterima 4 Oktober 2022

Table 1. *Specification of Dobot Robot[3].*

Parameter	Specification
Number of Axes	4, Base, Shoulder, Elbow and Wrist
Payload	500 grams
Max. Reach	320 mm
Position Repeatability (Control)	0.2 mm
Communication	USB/Wifi/Bluetooth
power supply	100V – 240V, 50/60 Hz
Power In	12V/7A DC
Power consumption	60 watts max
Working Temperature	-10oC – 60o C

The Dobot magician's Structure Diagram consists of the four joints were defined as J1, J2, J3, and J4. Position J2 as the starting point with coordinates (0,0,0), and the movement position is defined as (x, y, z). Dobot Magician is also equipped with a power supply and an active indicator light. The manipulator forearm length is 147 mm, the rear arm length is 135 mm, and the base arm length is 138 mm. The Dobot Magician structure diagram is shown in Figure 1.

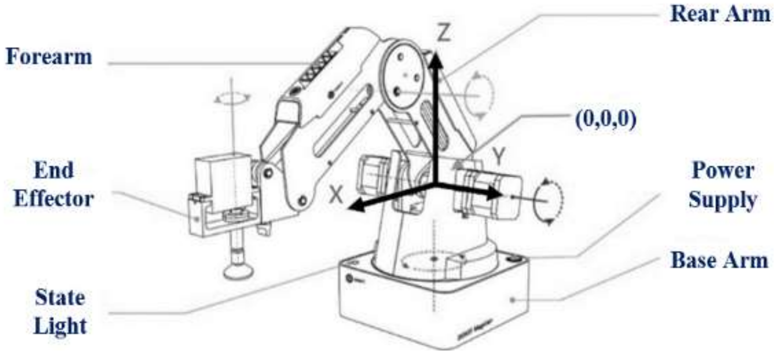

Figure 1. *Mechanical structure of Dobot Magician*[1].

With a mechanical structure as shown in Figure 1, all robot movements are rotational movement mechanisms which are determined by the rotation of the driving motor on J1, J2, J3 and J4. The dimensions and work area of Dobot Magician are shown in Figure 2.

Figure 2. *Dimensions and work area of Dobot Magician*[1].

Figure 2 explains that the radius of the robot's work area is 320 mm with the rear arm movement capable of up to 1800 and the forearm movement radius of 850. Meanwhile, the axis movement of each joint is shown in Table 2.

Table 2. *Working area and working speed of Dobot motor.*

Axis	Range	Maximum speed
Joint 1 (Base)	-1350 to 1350	3200/s
Joint 2 (Rear arm)	00 to 850	3200/s
Joint 3 (Forearm)	-100 to 950	3200/s
Joint 4 (Rotation servo)	900 to -900	4800/s

1.2 Robot Kinematics

Kinematics is defined as the movement mechanism of the robot in terms of position, speed and acceleration by ignoring the forces that affect the movement of the robot. Robot kinematics is generally divided into two, namely foreward kinematics and inverse kinematics. Foreward kinematics is a movement analysis to get position coordinates (x,y,z) if the position of each joint in the robot is known. While inverse kinematics is a movement analysis to get the value / magnitude of the angle from each joint if we know the position coordinate data (x,y,z) [5].

Robot kinematics consists of input or reference, controller system, electronics and robot mechanical system. The input reference is in the form of position, velocity or acceleration, which is expressed in a position vector coordinate (P) and orientation (x,y,z), while the output is an angle (θ1, θ2, 3… n) where n is the number of joints in the robot. So that the control diagram of the robot kinematics system is made as shown in Figure 3.

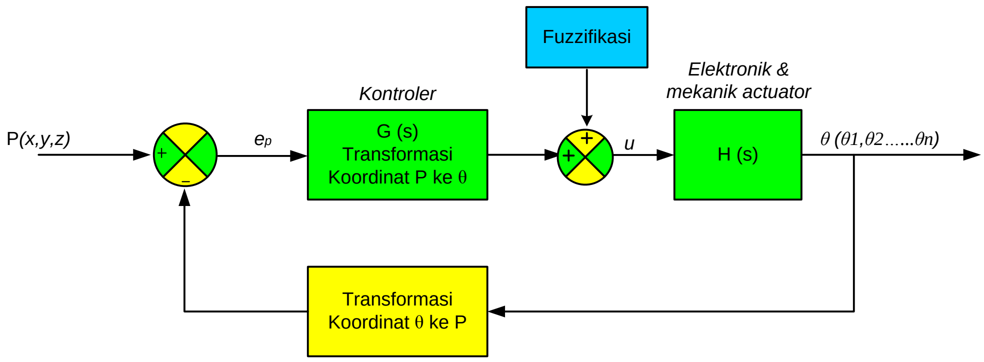

Figure 3. Robot kinematic system control diagram.

Figure 3 explains that the output measured from the movement of the robot is in the form of an angle resulting from the movement of the joint, or the angle of rotation of the motor joint. Controller G(s) is an inverse kinematic system to transform the position vector P (x, y, z) to angle. The movement of the join robot is influenced by the fuzzy output variable in the form of a variable (u) to move the actuator to get the desired angle. Changes in the angle of movement of each joint actuator is carried out with the Denavit Hatenberg (DH) Method approach and then the position of the movement angle is used as a feedback value on forward kinematics, as a comparison against the reference [6].

1.3 Fuzzy Logic Control

Fuzzy logic control is a control system that maps an input variable into a firm value in the output section. The fuzzy logic output is obtained from a rule-based decision-making process that groups the degree of membership of fuzzy sets and fuzzy rules [7],[8]. The fuzzy logic decision-making stage consists of:

1. Fuzzification
   Is the process of mapping the values of the input quantities into groups of members of the fuzzy set
2. Rule Base
   The fuzzy rule base contains fuzzy logic statements, which are in the form of IF-THEN statements. This prosess calculates the degree of truth of a set of fuzzy predicates with connectors of AND, OR, NOT.

3. Fuzzy inference engine
Fuzzy inference engine*translate* fuzzy statements in the rule base into mathematical calculations (fuzzy combinational). There are several inference engine methods, namely:
   a. Implocation.
      That is the process to get the result or value (linguistic or quantitative) of the consequent predicate of the given antecedent. The method used in this paper is the Mamdani method.
   b. Agregation.
      There are often cases where there are more than one rule. This means that the result of the Implication is worth more than one. Therefore, it is necessary to combine all these result values into a single fuzzy set. The aggregation method used here is the MAX method.

4. Defuzzification
Defuzzification returns the results of fuzzy calculations (fuzzy sets) into variables according to their ranges in the real world. Similar to the fuzzifier, the defuzzifier also uses a membership function to map the fuzzy set values into real variables. This process is to provide firm/crisp information from a fuzzification process so that it can be executed by the program.

2. RESEARCH METHODOLOGY

2.1 Weight and color based object moving robot system

Dobot movement control system as object transfer based on object weight and object color using fuzzy logic control system implemented using loadcell as weight sensor, CS-3200 as color sensor and Arduino Mega 2560 as additional control device, in addition to Dobot Integrated control. The Fuzzy system uses the Mamdani method with input parameters in the form of weight and color variables, while the output is in the form of Dobot Magician's placement movement, while the Dobot Studio programming method uses a blockly program (visual Programing Editor).

Dobot Magician is a robot manipulator that has been designed to have a communication interface that can communicate with external control addid. Arduino Mega 2560 as an external controller that functions to process data on the weight and color of objects can communicate with Dobot Magician through this interface. The mechanism for moving objects uses a suction cap Effector, namely a pneumatic vacuum that works based on supply from a mini compressor with a power of up to 35 Kpa. The schematic diagram of the Dobot working system for moving objects based on weight and color is shown in Figure 4 [9].

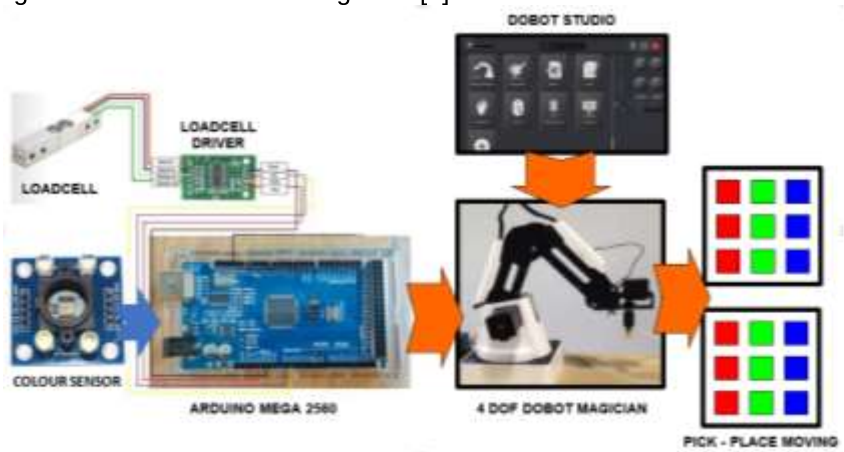

Figure 4. *Diagram of the Dobot Working System for moving objects based on weight and object color.*

The procedure for the movement mechanism of the robot arm in Figure 4 is as follows:

1. The robot arm moves from the neutral position towards the pick board to pick up the workpiece
2. Suction captake the workpiece from the pick board and move to the loadcell board then put the workpiece on the loadcell board to weigh the mass of the object,
3. Arduino stores the object's mass data as weight information in the fuzzy logic system
4. The suction cup takes the object from the loadcell board and moves to the color sensor board, then places the object on the color board.
5. Arduino stores object color data as information on the fuzzy logic system
6. The suction cup again takes the workpiece and moves to the place board and places the workpiece according to the recommended color and weight of the object.
7. For category 1 weight, it is placed on the first line with the color according to the Arduino recommendation,
8. For category 2 weight, it is placed on the second row with colors according to Arduino recommendations,
9. For category 3 weight, it is placed on the third row with the color according to the Arduino recommendation.

2.2 Dobot Magician's arm movement

In this study, the movement mechanism of the Dobot Magician was carried out using the teaching edge method combined with the blockly program. The teaching edge method is useful for determining the coordinates of the point of taking and placing the workpiece. While the blockly program is used to regulate the movement of the magician's dobot arm from each taking coordinate to the coordinates of placing objects. For every movement of the dobot magician, there are 3 stages of the process, namely pick, calculate and place. The design of the movement area and the coordinates of the dobot magician's movement angles are shown in Figure 5.

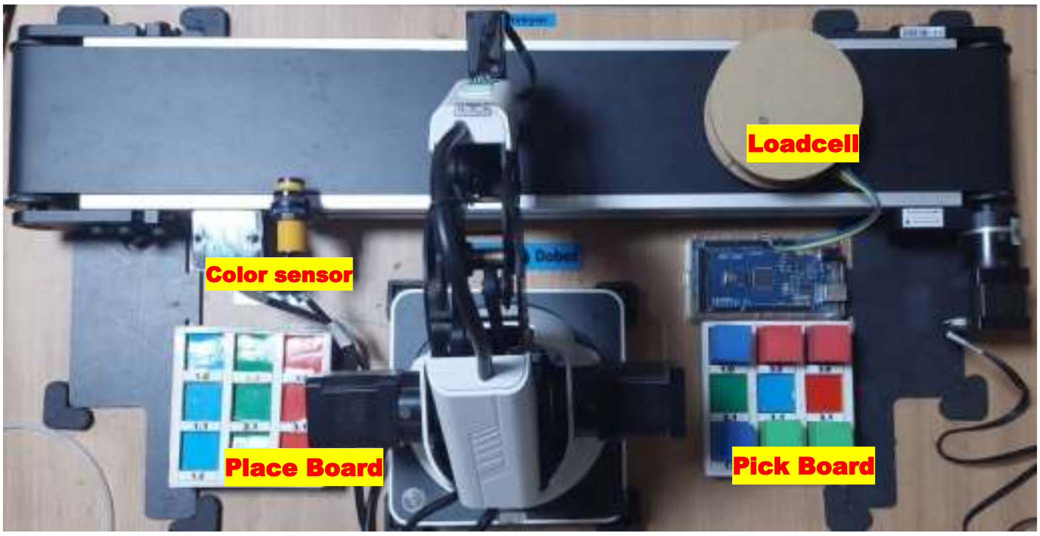

Figure 5. *Design of Product Pick and Place moving area by Dobot Magician.*

With an explanation of the scope of the dobot magician's movement area shown in Figure 5, it can be explained that the movement of the robot starts from the starting point as a zero location point moves to the pick board according to the ordinate instructions in the blockly program then goes to the Loadcell board to detect the weight of the object. The next movement is to the color sensor to determine the color dominance of objects and then move to the place board to place objects according to the color and weight criteria recommended by the fuzzy program.

3. DISCUSSION AND RESEARCH RESULTS

3.1 Kinematics of Pick and Place Coordinate Transformation

The placement of the pick board and the Place board is symmetrical to the base position of the Magician Dobot, so that each pick coordinate is in the opposite coordinate to the Place. In this implementation there are 9 Pick coordinates and 9 place coordinates shown in Figure 6.

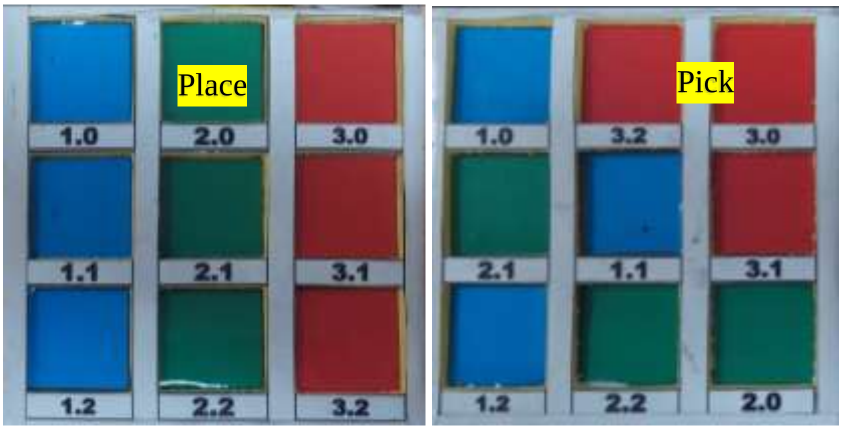

Figure 6. *Coordinates of board pick and board place.*

The coordinate points of Pick and Place are made in Table 2 below.

Table 2. *Coordinates of pick and place locations.*

NO	PICK CODE	PICK COORDINATES X	Y	Z	PLACE CODE	PLACE COORDINATES X	Y	Z
1	3.0	315	97.5	25	1.0	-315	-97.5	25
2	3.2	250	97.5	25	2.0	-250	-97.5	25
3	1.0	185	97.5	25	3.0	-185	-97.5	25
4	3.1	315	62.5	25	1.1	-315	-62.6	25
5	1.1	250	62.5	25	2.1	-250	-62.6	25
6	2.1	185	62.5	25	3.1	-185	-62.6	25
7	2.0	315	27.5	25	1.2	-315	-27.5	25
8	2.2	250	27.5	25	2.2	-250	-27.5	25
8	1.2	185	27.5	25	3.2	-185	-27.5	25

With the coordinate data (X, Y, Z) at the pick coordinates and also at the Place coordinates as in table 2, the equations for rotational movement and translational movement of the robot arm are drawn up using a transformation matrix approach as follows:

1. The translational movement of coordinate 1 on the X axis is a mm, on the Y axis it is b mm and the Z axis is c mm. The translation matrix equation that applies to coordinate 1 is:
$$\begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix} . \begin{bmatrix} 1 & 0 & 0 & a' \\ 0 & 1 & 0 & b' \\ 0 & 0 & 1 & c' \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
The transformation is called transformation (abc) [6].
2. The rotation of coordinates 1 about the x, y and z axes, respectively, is expressed in the following rotation matrix equation:
$$x = ;\ y = ;\ z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 0 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ -\sin\theta & \cos\theta & & 0 \\ 0 & 0 & & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Then the transformation matrix of the x-coordinate displacement of the robot by considering the translational and rotational movements can be expressed by the following equation:

$$P_\theta^x = \begin{bmatrix} Px \\ Py \\ Pz \\ 1 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

4. If Pick is the initial coordinate and Place is the destination coordinate, then Place/pick is a displacement transformation from initial coordinates to final coordinates.
5. In its implementation, all statements of the value of the calculation of the transformation matrix of the magician's movement are carried out using Matlab calculations.

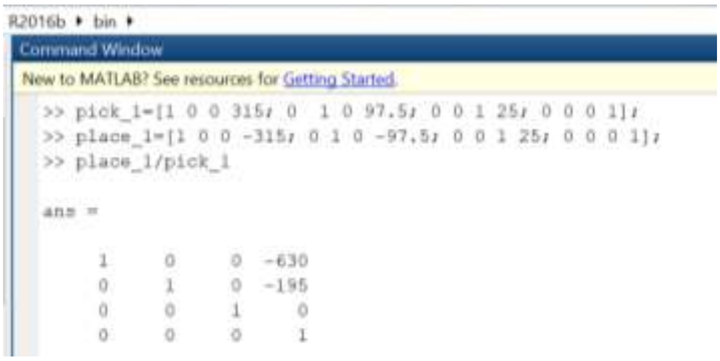

Figure 7. *Matlab as dobot movement matrix transformation analysis.*

Figure 7 implements Matlab as a tool to support the analysis of the movement of the robot arm in a matrix transformation system.

3.2 Fuzzy Programming Implementation

There are 2 input variables and 1 output variable in the fuzzy system, the input variables are mass and color, while the output variable is Place (coordinates of object placement). From the 2 variables, mass and color are derived into the following linguistic and quantitative values:

1. Time Variables:
   a. LITE : type Trapmf, Params [-50 0 50 200]
   b. LITHEAVY : type Trapmf, Params [100 200 300 400]
   c. HEAVY : type Trapmf, Params [300 450 500 550]]
2. Color Variables:
   a. RED : type Trapmf, Params [-128 0 96 224]
   b. GREEN : type Trapmf, Params [128 224 288 384]
   c. BLUE : type Trapmf, Params [288 416 512 640]

Membership Function for input variables with type and parameters is shown in Figure 8

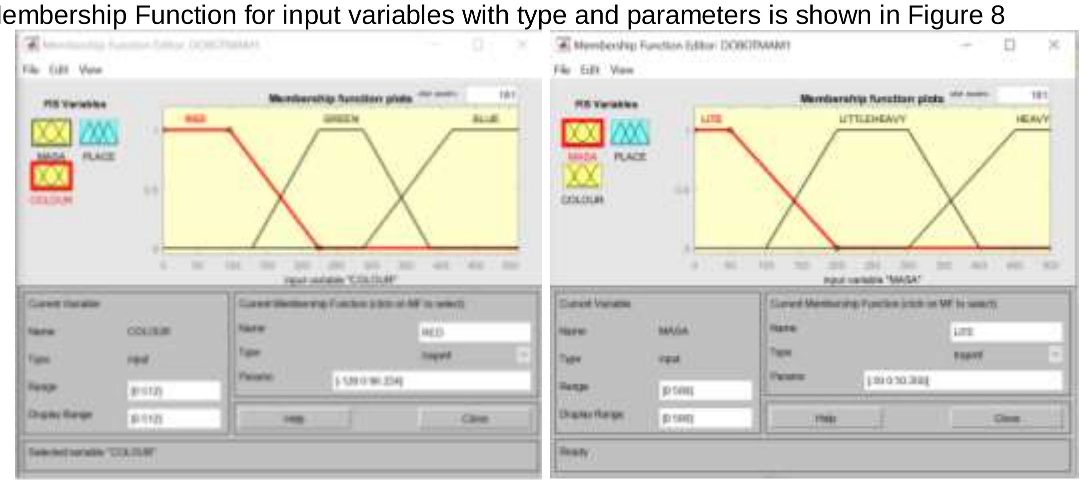

Figure 8. *Membership function variable input color and object mass.*

From 1 output variable Place is derived into the following linguistic and quantitative values:

1. Place RED
   a. R3.0 : type Trimf, Param [0 1 2]
   b. R3.1 : type Trimf, Param [1.5 2.5 3.5]
   c. R3.2 : type Trimf, Param [2 3 4]
2. Place GREEN
   a. G2.0 : type Trimf, Param [3 4 5]
   b. G2.1 : type Trimf, Param [3.5 4.5 5.5]
   c. G2.2 : type Trimf, Param [4 5 6]
3. Place BLUE
   a. B1.0 : type Trimf, Param [6 7 8]
   b. B1.1 : type Trimf, Param [6.5 7.5 8.5]
   c. B1.2 : type Trimf, Param [7 8 9]

With the output variable consisting of 9 membership functions, fuzzy rules can be made that result in defuzzification as shown in Figure 9 below.

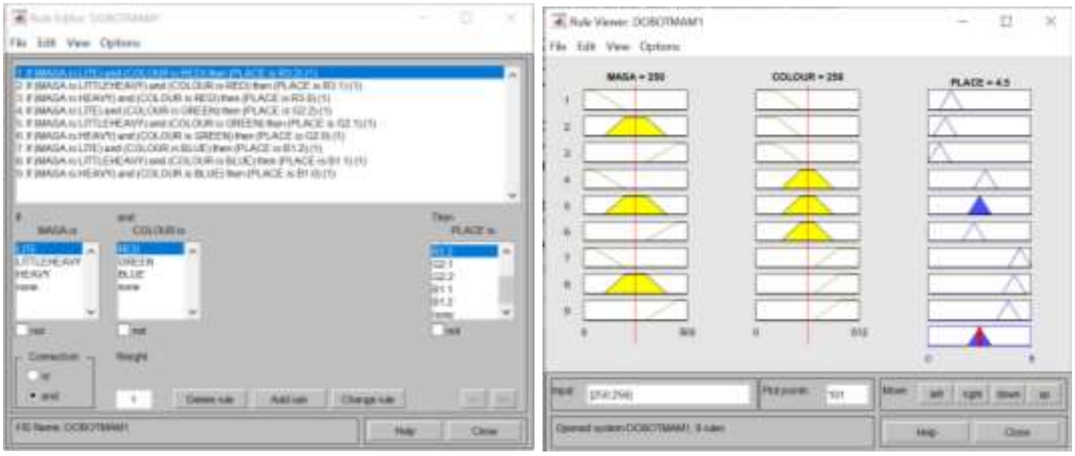

Figure 9. *Rule editor and rule Viewer Fuzzy.*

From the analysis of testing with fuzzy logic control, the data for the rule Viewer test is shown in Table 3 below.

Table 3. *Defuzzification test data.*

NO	TIME	COLOUR	PLACE	DESCRIPTION
1	65	399	7.9	Lite and blue at Row 3.0
2	250	412	7.5	LittleHeavy and blue at Row 3.1
3	450	430	7	Heavy and Blue at Row 3.2
4	86	289	4.5	Lite and Green at Row 2.0
5	268	272	4.5	LittleHeavy and Green at Row 2.1
6	445	168	4	Heavy and Green at Row 2.2
7	74.2	147	2.91	Lite and Red at Row 1.0
8	189	130	2.51	LittleHeavy and Red at Row 1.1
9	432	119	1.02	Heavy and Red at Row 1.2

Fuzzy logic control provides a fairly high flexibility to the firm values of each variable and is calculated as the degree of membership of the variable, a method of calculating the degree of membership is shown in the following calculation.

1. The 65gr period is on membership down from Variable LITE with a membership limit of 50gr – 200gr. The degree of membership is expressed by:

$\frac{b-x}{b-a} = \frac{200-65}{200-50} = \frac{135}{150}\ 0{,}9\ LITE$

2. Color 399 is found in the rising membership of the BLUE variable with the membership limit being 288 – 416, the degree of membership is stated by:
$\frac{x-a}{b-a} = \frac{399-288}{416-288} = \frac{111}{128}\ 0{,}86\ BLUE$

3. Place 7.9 is the highest degree of membership in Row 1.2, namely in the Blue row.

4. CONCLUSION

1. In this article, it has been proven that Dobot Magician which is controlled with dobot integrated control can synergize with external control in the form of an Arduino device in which the Fuzzy Logic control system is programmed for the implementation of controlling the movement of the robot arm to move objects based on the difference in color and weight of the object.
2. Matlab has provided an analysis of every movement of the dobot magician's arm from the initial coordinates to the destination coordinates using the Denavit Hatenberg matrix transformation method. Matlab calculations are used for robot movement instructions using a blockly program showing a shift in the coordinates of the destination with an average error of 0.005% or 0.5 mm for every 100 mm shift.
3. Process the fuzzy control in this study proves that the weight reading of the object through the loadcell and the color difference read by the color sensor can be processed as an input variable to provide instructions for moving the robot arm to move objects from the pick to the place according to the fuzzification results.

BIBLIOGRAPHY

[1] MR Sheikh, "Trajectory Tracking of a Four Degree of Freedom Robotic Manipulator,"*ProQuest Diss. Theses*, no. June, p. 76, 2019, [Online]. Available: http://proxy.libraries.smu.edu/login?url=https://search.proquest.com/docview/2289582610?accountid=6667%0Ahttps://smu.primo.exlibrisgroup.com/discovery/openurl?institution=01SMU_INST&vid=01SMU_INST:01SMU&?url_ver=Z39.88-2004&rft_val_fmt=info:ofi/fmt:kev:mt.

[2] PS Tsai, TF Wu, JY Chen, and FH Lee, "Drawing system with dobot magician manipulator based on image processing,"*Machines*, vol. 9, no. 12, 2021, doi:10.3390/machines9120302.

[3] L. Shenzhen Yuejiang Technology Co., "Dobot Magician User Manual," p. 138, 2017.

[4] R. Islam, A. Rahaman, and M. Habibur, "Cartesian Trajectory Based Control of Dobot Robot,"*IEOM Soc. int.*, no. Craig 2017, pp. 1507–1517, 2019.

[5] AK Critchlow, "Introduction to robotics.," 1985, doi:10.1016/0005-1098(86)90074-9.

[6] W. Budiarto,*Modern Robotics Theory & Implementation*, Sen. Yogyakarta: Andi Offset, 2014.

[7] EA Nugroho, "Traffic Light Control System Based on Fuzzy Logic,"*J. Symmetrical*, vol. 8, no. 1, pp. 75–84, 2017, doi: https://doi.org/10.24176/simet.v8i1.827.

[8] SN Sivanandam, S. Sumathi, and SN Deepa,*Introduction to fuzzy logic using MATLAB*. 2007.

[9] S. Riyadi, M. Suhanda, and EA Nugroho, "Kinematics of Heavy-Based Material Sorting Using Blockly Programs," vol. 1, no. 2, pp. 9–16, 2021.