

Obstacle Avoidance Based on Stereo Vision Navigation System for Omni-directional Robot

Faikul Umam^{1*}, Muhammad Fuad², Iswanto Suwarno³, Alfian Ma'arif⁴, Wahyu Caesarendra⁵

^{1,2}Department of Mechatronics Engineering, Universitas of Trunojoyo Madura, Bangkalan, Indonesia

³Department of Electrical Engineering, Universitas Muhammadiyah Yogyakarta, Yogyakarta, Indonesia

⁴Department of Electrical Engineering, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

⁵Faculty of Integrated Technologies, Universiti Brunei Darussalam, Brunei Darussalam

Email: ¹ faikul@trunojoyo.ac.id, ² fuad@trunojoyo.ac.id, ³ iswanto_te@umy.ac.id, ⁴ alfianmaarif@ee.uad.ac.id,

⁵ wahyu.caesarendra@ubd.edu.bn

*Corresponding Author

Abstract—This paper addresses the problem of obstacle avoidance in mobile robot navigation systems. The navigation system is considered very important because the robot must be able to be controlled from its initial position to its destination without experiencing a collision. The robot must be able to avoid obstacles and arrive at its destination. Several previous studies have focused more on predetermined stationary obstacles. This has resulted in research results being difficult to apply in real environmental conditions, whereas in real conditions, obstacles can be stationary or moving caused by changes in the walking environment. The objective of this study is to address the robot's navigation behaviors to avoid obstacles. In dealing with complex problems as previously described, a control system is designed using Neuro-Fuzzy so that the robot can avoid obstacles when the robot moves toward the destination. This paper uses ANFIS for obstacle avoidance control. The learning model used is offline learning. Mapping the input and output data is used in the initial step. Then the data is trained to produce a very small error. To support the movement of the robot so that it is more flexible and smoother in avoiding obstacles and can identify objects in real-time, a three wheels omnidirectional robot is used equipped with a stereo vision sensor. The contribution is to advance state of the art in obstacle avoidance for robot navigation systems by exploiting ANFIS with target-and-obstacles detection based on stereo vision sensors. This study tested the proposed control method by using 15 experiments with different obstacle setup positions. These scenarios were chosen to test the ability to avoid moving obstacles that may come from the front, the right, or the left of the robot. The robot moved to the left or right of the obstacles depending on the given V_y speed. After several tests with different obstacle positions, the robot managed to avoid the obstacle when the obstacle distance ranged from 173 – 150 cm with an average speed of V_y 274 mm/s. In the process of avoiding obstacles, the robot still calculates the direction in which the robot is facing the target until the target angle is 0.

Keywords—Artificial Intelligence; Stereo Vision; Navigation; Obstacle Avoidance; Omnidirectional Robot.

I. INTRODUCTION

Navigation research on robots is starting to become a separate topic. Robotics experts began to develop several algorithms to find free paths. The navigation system on the robot is considered very important because the robot must be able to be controlled from the starting point to the target (destination) safely. There are two points that become a reference, namely the first is safe, where the robot must be

able to avoid obstacles or, in other words, not experience collisions with obstacles. Second, the robot must always be sure to arrive at the target (destination).

The challenge is how to choose which of the many travel solutions. In real life, a motorist is often still filled with doubts in making choices, like in the case above. It will be very interesting if this is implemented in a mobile robot (autonomous robot). Starting from simple problems like this, new problems can be developed if these problems are applied to autonomous mobile robots where there are obstacles that the robot must avoid.

Complex problems will arise because a mobile robot must be able to detect obstacles and make decisions to avoid these obstacles, not to mention if the main target (goal) disappears from the camera's view. All of that requires a very complicated computational process. The problem from the outside that must also be taken into account is the light intensity considering the sensor to be used is the camera sensor. The above factors will make it difficult for the robot to arrive at its destination.

Research on obstacle avoidance has often been carried out by researchers, starting with presenting fuzzy algorithms for reactive navigation for mobile robots in complex environments [1]–[61]. In this study, it was stated that fuzzy logic is quite good and has a fast response to overcome obstacles. This research only addresses the problem of static obstacles in the robot's work area, while the elements of moving obstacles caused by moving objects have not been included. Model-based Predictive Controller (MBPC) using Neural Networks with ultrasonic sensors for mobile robot navigation is also carried out to overcome static obstacles that appear unexpectedly in the robot's work area [62]–[77], where this study only focused on static obstacles that unexpected presence. Dynamic Artificial Neural Network (DANN) algorithm for motion planning and mobile robot paths [78]–[80]. This research can navigate a mobile robot on a flat surface between static and dynamic obstacles.

To further improve the ability of the robot in terms of overcoming obstacle avoidance, the researchers implemented a combination of two methods by combining the Neural Network and Fuzzy methods known as Generalized Dynamic Fuzzy Neural Networks (GDFNN) to design real-time control



autonomous mobile robots [81]. The experimental results show that compared to conventional fuzzy logic control, GDFNN is superior in performance. In addition, there are also those who combine Reinforcement Learning with Neural Networks (RLNN) to overcome the problem of autonomous mobile robot obstacle avoidance [82]. The simulation results show that the robot can improve learning ability and can complete the tasks given in a complex environment.

To detect obstacles in real-time, researchers are starting to innovate by combining camera sensors with lasers. This type of sensor has reliability in detecting 2- and 3-dimensional objects [83]. Even in recent research [84], stereo-vision techniques were developed based on a combination of Omnidirectional cameras and perspective cameras. This technique combines a 360° field of view from an Omnidirectional camera with a long field of view from a perspective camera which is used to determine the approximate 3-dimensional position of obstacles. Several implementations of vision systems based on color sensors [85], camera sensor Pixy 2 CMUcam5 [86], and thermal cameras [87] were investigated in previous studies. The studies above show very good results, namely obstacles that can be detected in real-time. However, no moving obstacles have been used in the previous research. This study has a goal to develop an obstacle avoidance method to avoid moving obstacles.

Based on some of the research that has been done, this research will develop an autonomous mobile robot that can navigate independently to overcome moving obstacles caused by environmental changes in the robot's work area. In order to recognize the environment, two webcams are used as stereo vision sensors. Obstacle objects used are pedestrians who are recognized by upper body detection. The reason for choosing this object is because the robot's work area is the actual environment. The intelligence method as a control system must be able to overcome the problem of moving obstacles in the work area and be able to make the robot arrive at the target (destination). The control system used to avoid obstacles is Neuro-Fuzzy. In this study, a 3-wheeled omnidirectional robot was used, with the hope that this robot will be able to overcome obstacle problems in a smooth and flexible manner.

In order for the mobile robot to arrive at a predetermined target (goal), it is necessary to design a robot behavior that has the ability to identify the target object, the ability to detect moving obstacles and make decisions to avoid them flexibly. These behaviors will be used by the robot in navigation. In its movement from the starting point to the destination, the robot is controlled using the Neuro-Fuzzy algorithm and uses stereo vision. To improve its ability to deal with dynamic environments, omnidirectional robots and the Neuro-Fuzzy algorithm are used, which will help the robot to detect obstacles and make decisions that the robot will avoid.

This study focuses on robot navigation systems such as determining the position of the target (destination) where the target is assumed to always be in the robot's view, detecting obstacles (avoiding obstacles), and producing flexible and smooth movements. Obstacle objects used are pedestrians who are identified by upper body detection. The work area

used by the robot is a corridor and an indoor room with a length of 4 meters and a width of 4 meters. This research does not care about the shortest distance because the robot walks not based on the path.

This study aims to develop a stereo vision-based navigation system for supporting obstacle avoidance of omnidirectional mobile robots. The proposed method exploits the Neuro-Fuzzy algorithm to produce a barrier-free path in real-time and guide the movement of the robot in order to be flexible and smooth. In order to guide the mobile robot to arrive at a predetermined destination, it is necessary to design a robot behavior that has the ability to identify the target object, detect moving obstacles and make decisions to avoid them flexibly. The objective of this study is to address the robot's behaviors for navigation. This study contributes to improving state of the art in obstacle avoidance based on the visual sensor for robot navigation systems by utilizing the stereo camera for detecting a target and obstacles distance and angles as input to ANFIS, such as depicted in Fig. 1. This paper has research methodology that consists of two main parts. The first part is building a method to control the linear and angular velocity of autonomous mobile robots.

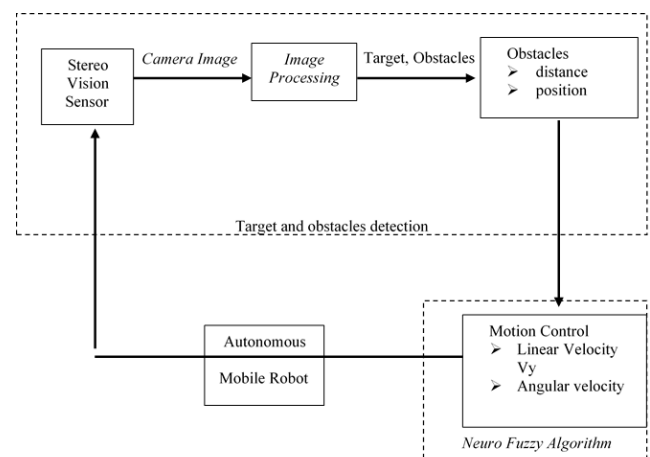


Fig. 1. Research methodology of the proposed navigation system

II. LITERATURE REVIEW

A. Modeling Navigation System

In that study, [28] implemented the Dynamic Artificial Neural Network algorithm for motion planning and mobile robot paths. This method can navigate the mobile robot on a flat surface and can overcome static and dynamic obstacles. Movement control is carried out (trained) directly (online) with the backpropagation algorithm using potential fields to avoid obstacles. Meanwhile, the movement of moving obstacles is estimated or predicted by other ANNs. In this study, the authors created a robot named "MITMÓT Robot."

The method used is Backpropagation through Time (BPTT). It is a training method of dynamic feedback artificial neural networks that involves supplying inputs forward through the network and then propagating errors backward to modify the weights. The main idea of this method is to train the ANN with the backpropagation algorithm by means of a feedback neural network that is iterated as much as possible. This training is carried out in the following way: input is given to the first ANN, which calculates its own output. This

output will be the second ANN input, and so on. The resulting output is used to calculate the error.

B. Control System

The control method functions to regulate the robot so that it can move from the initial position to the destination position without experiencing a collision. In contrast to [28], which used a Dynamic Artificial Neural Network algorithm to navigate a robot on a flat surface, Phinni et al. (2006) used a Compact Genetic Algorithm (cGA) and fuzzy logic to control the robot's movement. The Compact Genetic Algorithm (cGA) method will produce a path from the area that has been captured with the camera. Then the fuzzy method takes over to control the mobile robot so that it can follow the path that has been generated by the cGA. It produced satisfactory results in the form of a simulation in that the robot was able to follow the cGA-generated path. While both methods produced satisfactory results in simulation, however, the DANN approach was better than cGA for the ability to handle dynamic obstacles. This study is inspired by previous research in that the Neuro-Fuzzy control system is proposed to make a decision to go to the target. While the robot goes to the target, if there is an obstacle, the other Neuro-Fuzzy control systems control the robot to avoid collision while still calculating the target's position so that it does not crash out of sight.

III. PROPOSED METHOD

A. Stereo Vision-Based Navigation System

System design can be described in the form of a block diagram in Fig. 2. The system consists of three main parts. First, the stereo vision sensor functions to provide visual information on obstacle objects and targets. Secondly, the laptop functions to process visual information received from the stereo vision sensor to produce distance parameters and the position of obstacles and targets, then both of these information are processed using ANFIS. Third, the output generated by ANFIS is the rotational speed of the motor, which is sent to the robot (Robotino). Robotino's movement changes depending on the information received from the stereo vision sensor.

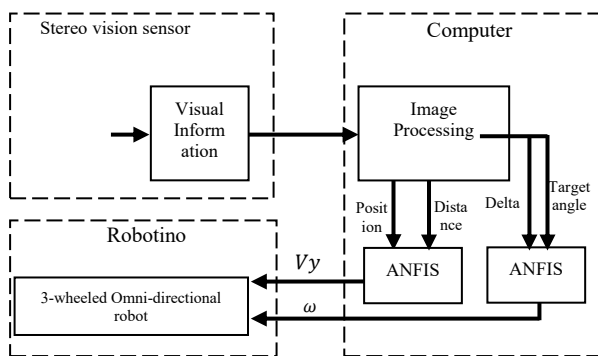


Fig. 2. Block diagram of Stereo Vision Navigation System

B. Target and Obstacle Detection

In this study, apart from detecting the upper body as a moving obstacle, it is also necessary to detect other objects as the target of the robot. The object chosen is a 2D image, namely a circle. The researcher chose 2D object detection as the target object because the computational process is not too

heavy, allowing the robot to detect it in real-time. Circle detection system using Hough transform.

As with upper body detection, in Fig. 3, the initial process for detecting circular objects is the capture process from both cameras. The result obtained is an image in RGB form, and the image is converted to an image in the grayscale form. After the image in grayscale form is obtained, then a circle is searched using the Hough transform method.

Upper body detection, as shown in Fig. 4, begins with capture by the camera, which produces an image in RGB form. The RGB image is then converted into gray form. Images in the gray form will be trained by a haar cascade classifier which contains samples of the Upper Body up to hundreds of times. If the Upper Body has not been found, the training will continue. If the Upper Body is found, a marker is made in the form of a rectangular box in the area around the Upper Body.

As explained earlier, in this study, the obstacle objects used were pedestrians who were identified by upper body detection. There are several algorithms that have been developed to detect the upper body, for example, using skin color detection and using eye blink patterns. To speed up the computation process, upper body detection in this study uses the Haar-Like Feature library, which was developed by Paul Viola and improved by Rainer Lienhart.

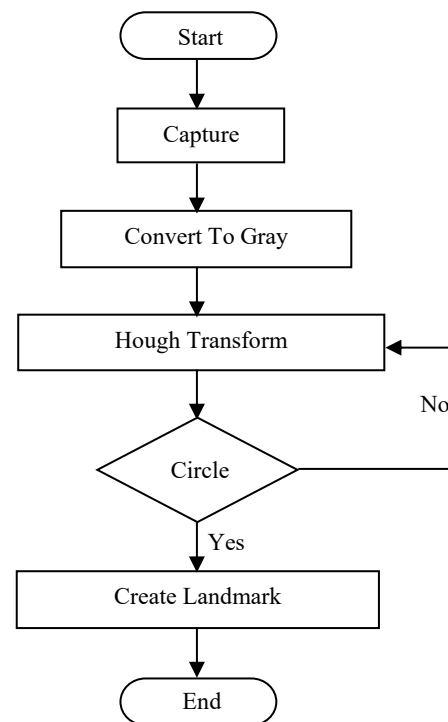


Fig. 3. Detection of the circle as the landmark of the target

C. Stereo Vision-Based Distance and Angle Estimation

In the Stereo vision system, two web cameras are placed horizontally and parallel, which are separated by a distance called BaseLine b , and the focal length of the lens (focal length) is f , as in Fig. 5. The difference between Rl_{qx} and Ll_{qx} is called the disparity, and is directly related to the distance SC_{qz} .

This study created a stereo vision system using two horizontally mounted web cameras. The distance between the center point of the right webcam and the center point of the left webcam is called the baseline. There are no binding rules for determining how wide the distance should be used to create a stereo-vision system. The wider the baseline, the greater the disparity value. In this study, the baseline used was 8 cm or 80 mm. The focal length value for each web camera may not be the same. Usually, this value has been included by the vendor in the camera specifications. The web camera used is a PCC5020 8 MegaPixel type prolink with a focal length value of 6.3 mm.

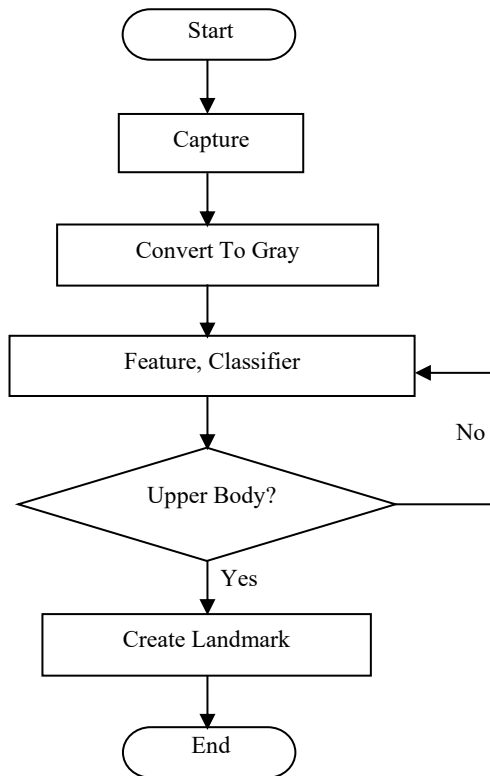


Fig. 4. Detection of the upper body as the landmark of the obstacle

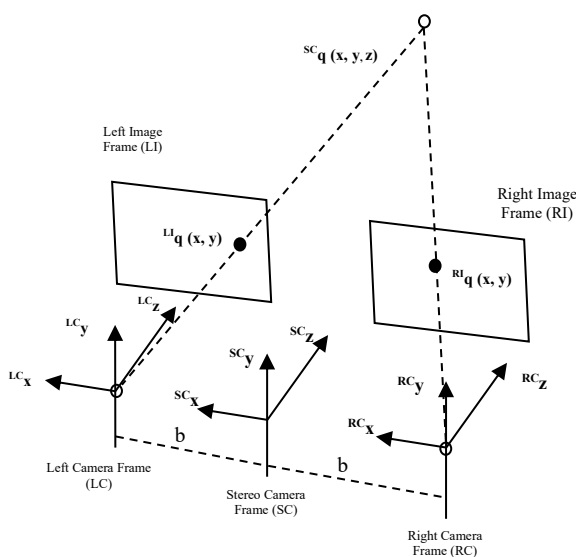


Fig. 5. Stereo imaging model

After the obstacle object and the target object are identified, then the midpoint of the two objects must be found (x_{center} , y_{center}) on each camera, so four parameters are generated, namely XL_{center} , YL_{center} , XR_{center} , and YR_{center} . As shown in Fig. 6 for an obstacle object, because the face marker uses a rectangle, the center point of the object can be found using Equations (1) and (2). Midpoint markers on the left and right camera, respectively, are described as Equations (1) to (4).

$$XL_{center} = X \left(\frac{1}{2} \times width \right) \tag{1}$$

$$YL_{center} = Y \left(\frac{1}{2} \times height \right) \tag{2}$$

$$XR_{center} = X \left(\frac{1}{2} \times width \right) \tag{3}$$

$$YR_{center} = Y \left(\frac{1}{2} \times height \right) \tag{4}$$

Based on Equations (1) to (4) and Fig. 5, the distance calculation formula is obtained by using the equation as

$$sc_{qz} = \frac{2}{RI_{qx} - LI_{qx}} (fb) \tag{5}$$

where $RI_a = CR$ is the midpoint of the object on the right frame and $LI_{qx} = CL$ is the midpoint of the object on the left frame.

Apart from the distance parameter, another parameter needed to avoid obstacles is the orientation position of the obstacle. In this case, the orientation position in question is the angle of the obstacle object. Each camera has a different angle of view. The specifications issued by the vendor state that the angle of view on the PCC5020 type Prolink camera is 70 degrees. When measuring, this type of camera only has an angle of view of 50 degrees. This value is obtained from the measurement results by drawing a straight line from the left and right viewpoints of the camera to the center point of the camera, as shown in Fig. 6.

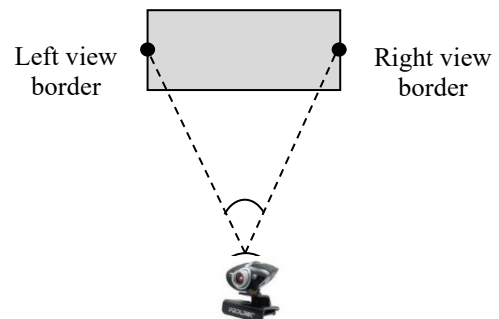


Fig. 6. Measuring angle view camera Prolink type PCC5020.

Once the camera angle is known, the object angle can be found by mapping the two camera angles with the resolution used. If the resolution used is 640×320 , then it can be mapped as shown in Table I.

TABLE I. ANGLE MAPPING WITH CAMERA PIXELS

Pixel	0	320	640
Angle	25	0	-25

If the pixel is 0, the angle is 25. If the pixel is 320, the angle is 0. And if the pixel is 640, the angle is -25. From this mapping, we can find the linear function shown as

$$y = -0.0781x + 25 \quad (6)$$

where y is the angle, and x is the pixel value. Calculation results are shown in Table II. In the case of stereo cameras, each camera's angle value is calculated first. After that, the angle values for each camera are added up and then divided by two.

TABLE II. ANGLE CALCULATION BASED ON PIXELS

No.	Pixel	Angle	No.	Pixel	Angle
1	0	25	17	320	0.008
2	20	23.438	18	340	-1.554
3	40	21.876	19	360	-3.116
4	60	20.314	20	380	-4.678
5	80	18.752	21	400	-6.24
6	100	17.19	22	420	-7.802
7	120	15.628	23	440	-9.364
8	140	14.066	24	460	-10.926
9	160	12.504	25	480	-12.488
10	180	10.942	26	500	-14.05
11	200	9.38	27	520	-15.612
12	220	7.818	28	540	-17.174
13	240	6.256	29	560	-18.736
14	260	4.694	30	580	-20.298
15	280	3.132	31	600	-21.86
16	300	1.57	32	620	-23.422
			33	640	-24.984

D. Architecture of ANFIS-Based Obstacle Avoidance

To make a robot able to avoid obstacles, it needs an appropriate learning mechanism to produce optimum control. Obstacle avoidance control in this study uses ANFIS, where the learning model used is offline learning. The initial step is mapping the input and output data. Then the data is trained using MATLAB to produce a very small error. Data mapping uses two inputs and one output, as in Appendix 1. The input used is the object distance (cm) and the object orientation position (degrees), while the output is the speed V_y (mm/s). This output value is used by the robot to avoid obstacles. The closer the obstacle is, and the smaller the obstacle angle with the robot, the greater the resulting V_y speed, and vice versa, if the obstacle distance is far and the obstacle angle is large, the V_y speed is small.

Actually, the basic gist of this training is very simple, the FIS that was previously generated based on data mapping will be trained by changing the premise and consequent parameters so that the FIS produced follows the predetermined output data. As shown in Fig. 7, the training was carried out for 5000 epochs, where the smallest error produced was 0.072 in the 1986th epoch. After the training is completed, a FIS model is generated with new premise parameters and consequence parameters. As in data mapping, the FIS training results produce input distances and angles that have been modeled in the form of a membership function.

In Fig. 8, the angle of obstruction is divided into 5 Membership functions with a value range of -25 to 25 using the Gaussian type, namely very small angles with membership values (7.621, -22.76), small angles with membership values (0.9871, -2.6), medium angles with membership values (1.239, -1.483), large angles with membership values (2.181, 6.629), and very large angles with membership values (6.941, 23.85).

Obstacle distance is also divided into 5 Membership functions, as shown in Fig. 9, with a value range of 130 to 250, namely very close distance to membership values (12.8, 130), close distance to membership values (12.72, 160), medium distance to membership values (12.84, 190), long distance with membership values (12.72, 220), and very long distance with membership values (12.8, 250). In addition, this training also produces consequent parameters, as shown in Table III. In the Sugeno FIS model, the output is not a fuzzy set but a constant or linear equation.

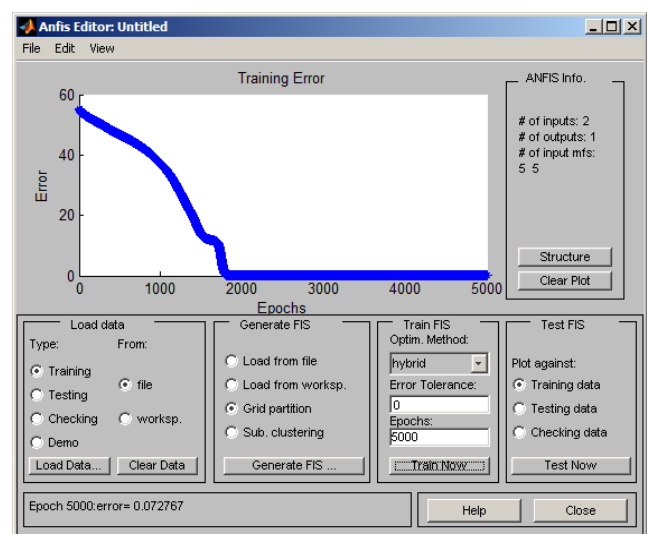


Fig. 7. Obstacle avoidance data training results using MATLAB

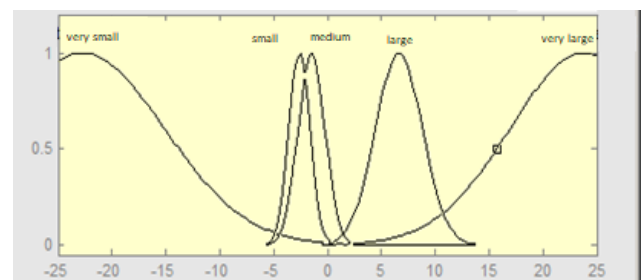


Fig. 8. Membership function angle of obstacle

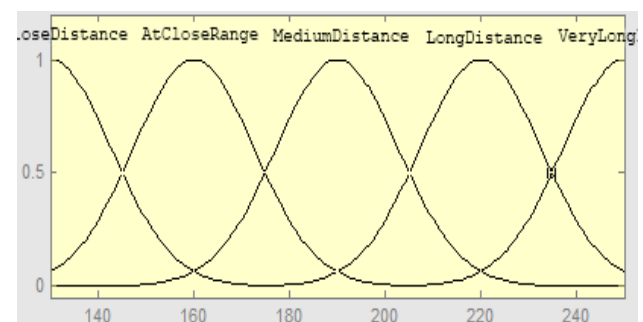


Fig. 9. Membership function distance of the obstacle

TABLE III. CONSEQUENT PARAMETERS

No	Consequent Parameter		
	<i>p</i>	<i>q</i>	<i>r</i>
1	-3.999	3.085	13.08
2	-4.001	2.288	-78.55
3	-3.998	0.5401	103
4	-4.002	0.3292	62.46
5	-3.988	0.4048	-65.27
6	-3.998	3.093	12
7	-4.004	2.293	-79.46
8	-3.989	0.5309	104.6
9	-4.032	0.2959	70.36
10	-3.884	0.4301	-71.29
11	-3.979	2.96	28.91
12	-4.132	2.212	-64.43
13	-3.802	0.6793	76.15
14	-4.178	0.8071	-47.32
15	-2.477	-0.0189	19.24
16	-3.983	3.513	-42.15
17	-4.114	2.547	-121.4
18	-3.844	0.06639	186.1
19	-4.077	-1.293	441.8
20	-2.911	1.609	-355.4
21	-4.002	3.449	-33.51
22	-4	2.508	-117.3
23	-4.022	0.1345	178.2
24	-3.885	-1.048	379.9
25	-4.35	1.548	-309.2

Rule 1 if $S = SS$ and $J = JSD$ then $f1 = p1(S) + q1(J) + r1$

Rule 2 if $S = SS$ and $J = JSJ$ then $f2 = p2(S) + q2(J) + r2$

where p, q, r are consequential parameters. The Sugeno FIS model defuzzification process is far more efficient than other FIS models. This is because the Sugeno FIS model calculates the output value by using

$$f = \frac{w1f1 + w2f2}{w1 + w2} \quad (7)$$

where $w1$ is SS AND JSD , and $w2$ is SS AND JSJ , while $f1$ and $f2$ are the i^{th} output rules.

The ANFIS architecture is determined by the number of membership functions and their membership values. As shown in Fig. 10, the designed ANFIS architecture has two inputs, namely position and obstacle distance, where each input has 5 membership functions. The output of ANFIS is the speed Vy which is the speed of the robot (mm/s) on the y -coordinate axis in the real world.

a. Layer 1: generate membership function

$$O1.i = \mu_{Ai}(x), \text{ where } i = 1, 2, 3, 4, 5 \text{ or}$$

$$O1.i = \mu_{(Bi-5)}(y), \text{ where } i = 6, 7, 8, 9, 10$$

Where x and y are inputs at node i and Ai , or $Bi - 5$, are fuzzy sets corresponding to Gaussian functions as shown in (8).

$$\mu_{Ai}(x) = e^{(-1/2((x-c)/\sigma)^2)} \quad (8)$$

where c and σ are premise parameters. If these parameters change, the Gaussian curve will also change.

b. Layer 2: generate the firing strength of a rule that is by multiplying each input signal.

$$O2i = w_i = \mu_{Ai}(x) \times \mu_{Bi}(y)$$

where $i = 1, 2, 3, 4, 5, \dots, 25$

c. Layer 3: produces normalized firing strength output

$$O3i = \bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3 + w_4 + w_5}$$

where $i = 1, 2, 3, 4, 5, \dots, 25$

d. Layer 4: calculate output based on consequent parameters

$$O4i = \bar{w}_i f = \bar{w}_i(p_i x + q_i y + r_i)$$

where i is the normalized firing strength of layer 3 and p_i, q_i, r_i are the consequent parameters.

e. Layer 5: calculates the ANFIS output signal by summing all incoming signals

$$O5i = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

E. Control System for Guiding Robot to Target

In addition to controlling obstacle avoidance, this study also made control towards the target, which functions to control the robot to move towards a predetermined target. This control also uses ANFIS with an offline learning model. Data mapping uses two inputs and one output. The input used is the target angle and angular delta, while the resulting output is the angular velocity (ω). This output value is used by the robot to go to the target.

1. Data Training

As with obstacle avoidance control, the initial step taken in controlling the target is to map the input and output data. Input and output data mapping, as in Appendix 2, will also be trained using MATLAB to produce very small errors. As shown in Fig. 11, after data training is carried out, the smallest error generated is 0.0000000932358 in the second epoch.

In addition, data training also produces a FIS model with new premise parameters and consequence parameters. As in data mapping, the FIS training results produce angle inputs and angle deltas that have been modeled in the form of a membership function.

The target angle is divided into 3 membership functions with a value range of -25 to 25, as shown in Fig. 12, namely a very small angle with a membership value (10.62, -25), a medium angle with a membership value (10.62, 0), and a large angle with a membership value (10.62, 25). The delta target angle is divided into 3 membership functions, as shown in Fig. 13.

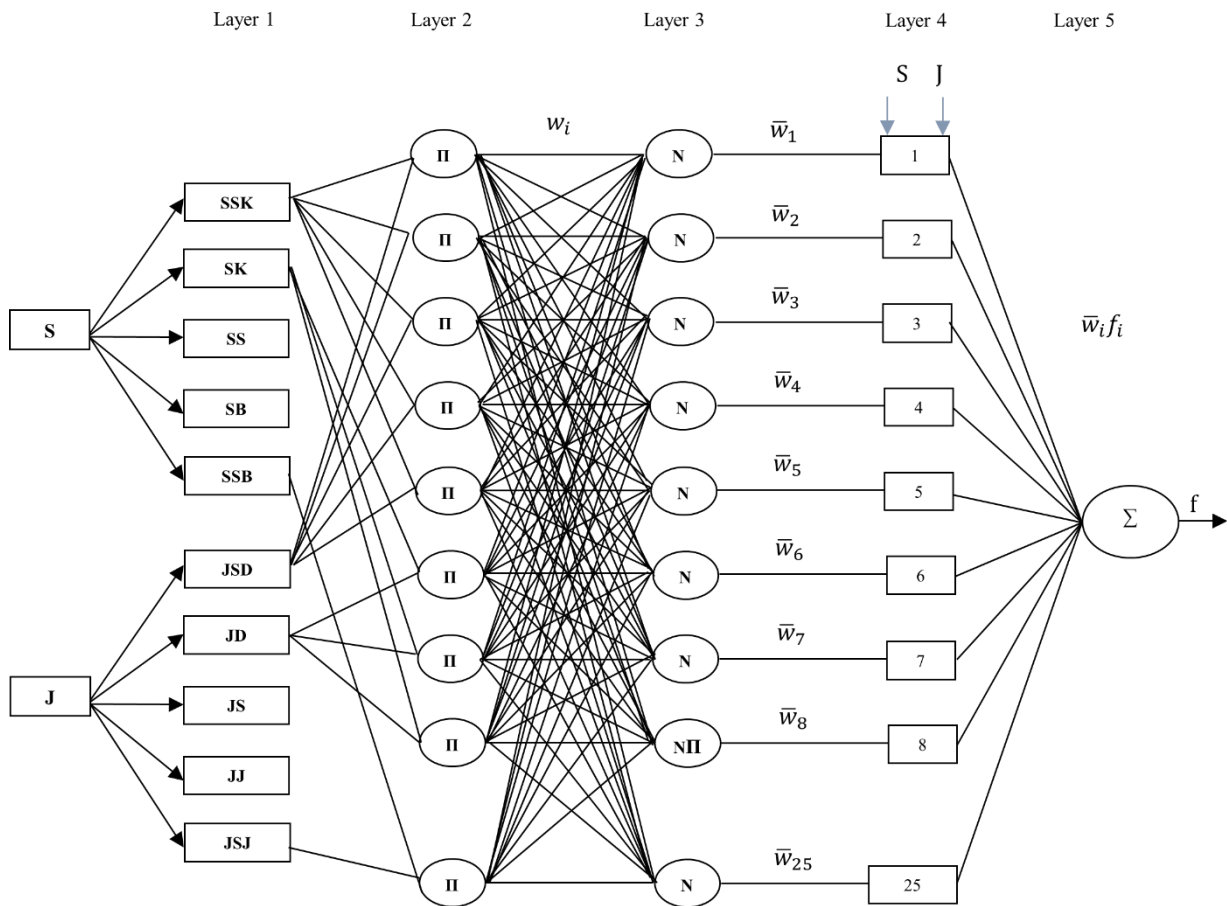


Fig. 10. ANFIS-based obstacle avoidance architectural design

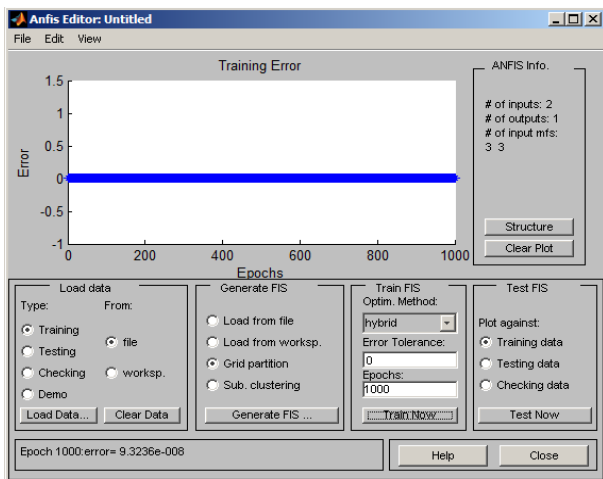


Fig. 11. Results of data training towards targets using MATLAB

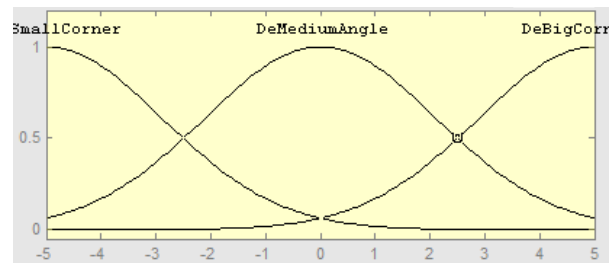


Fig. 13. Membership function delta target angle

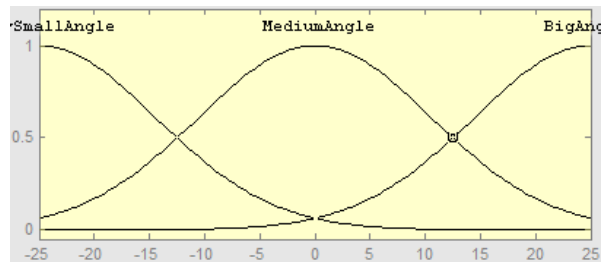


Fig. 12. Membership function target angle

2. ANFIS Architecture Design

The design of the ANFIS architecture to control towards the target is almost the same as the obstacle avoidance control system architecture, only me if in the obstacle avoidance control system, each input has 5 membership functions, in the control system towards the target, each input has 3 membership functions.

a. Layer 1: generate membership function

$$O1. i = \mu_{Ai}(x), \text{ for } i = 1, 2, 3 \text{ or}$$

$$O1. i = \mu_{(Bi - 3)}(y), \text{ for } i = 4, 5, 6$$

where x and y are the inputs at the i th node and Ai , or $Bi - 3$, are the fuzzy sets corresponding to the Gaussian function as

$$\mu Ai(x) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \quad (9)$$

where c and σ are premise parameters. If these parameters change, the Gaussian curve will also change.

b. Layer 2: generates the firing strength of a rule by multiplying each input signal

$$O2i = w_i = \mu Ai(x) \times \mu Bi(y)$$

where $i = 1, 2, 3, 4, 5, \dots, 9$

c. Layer 3: produces normalized firing strength output

$$O3i = \bar{w}_i = \frac{w_i}{w_1 + w_2 + w_3}$$

where $i = 1, 2, 3, 4, 5, \dots, 9$

d. Layer 4: calculate output based on consequent parameters

$$O4i = \bar{w}_i f = \bar{w}_i (p_i x + q_i y + r_i)$$

where i is the normalized firing strength of layer 3 and p_i, q_i, r_i are the consequent parameters.

e. Layer 5: calculates the ANFIS output signal by summing all incoming signals

$$O5i = \sum \bar{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

IV. RESULTS AND DISCUSSIONS

This section discusses the results of several experiments and explains how successfully the system can function.

A. Obstacle Detection

Obstacle objects are pedestrians who are recognized by detecting the upper body, as shown in Fig. 14. The detection process uses haar-feature. An obstacle object (pedestrian) is caught by the camera. Then the object is marked with a box. Obstacle objects can be detected by the stereo vision system with the shortest distance of 130 cm and the longest distance of 300 cm.

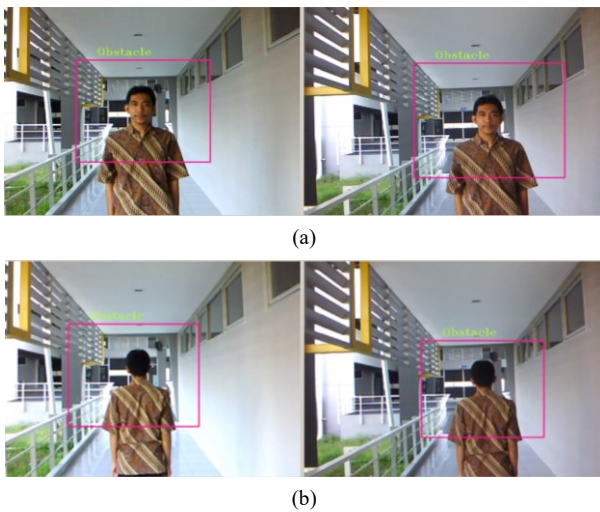


Fig. 14. (a) Front obstruction detection, (b) Rear obstruction detection.

After the obstacle object has been identified, the marker box (detector) is not fixed in one place, the marker box moves

or oscillates continuously even though the object is stationary. This means that the center point of the object also changes. If this is allowed, it will be very influential in the distance measurement process. To overcome this, filtering is carried out using a moving average against the midpoint of the object. Every ten midpoint values generated are stored in an array for filtering. As shown in Fig. 15, the moving average filter is good enough to overcome this.

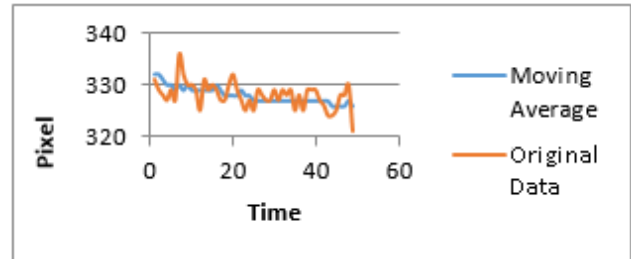


Fig. 15. Moving average filter against barrier object data center

B. Target Detection

The target object is a circle like in Fig. 16, which is detected using the Hough Transform. Just like detecting an obstacle object, the circle marker also continues to move or oscillate. This also greatly affects the measurement of the target distance. Therefore, the midpoint of the target detection is also filtered using a moving average. As shown in Fig. 17, moving averages are quite good at dealing with this.



Fig. 16. Target detection

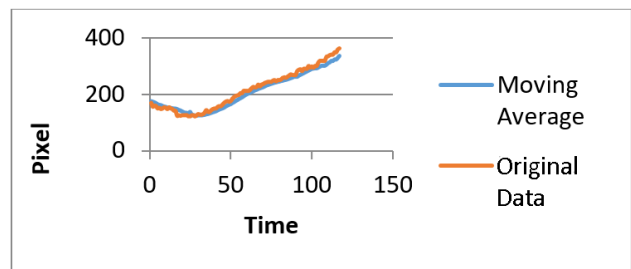


Fig. 17. Moving average filter against target object data center

C. Obstacle Distance Measurement

Obstacle distance measurements were tested in the range of 0 – 300 cm. Based on test results, the distance to obstacle objects can be measured with the closest distance of 130 and the farthest distance of 270 cm. It can recognize objects. This is due to setting the size of the detector. If the detector is made very small, then when at a close distance, the object will not be recognized and vice versa. For this reason, the detector size is 200×200, with a minimum distance range of 130 cm and a maximum distance of 270 cm. As shown in Table IV, the minimum error for distance measurement occurs when the obstacle distance reaches 210 cm with an error value of 0%,

and the maximum error is at a distance of 170 cm with an error of 5.2%.

TABLE IV. OBSTACLE DISTANCE MEASUREMENT

Actual Distance (cm)	Measurement Distance (cm)	Error (%)
130	130.39	0.30
150	150.09	0.06
170	173.09	1.81
190	190.00	0.00
210	210.10	0.04
230	232.77	1.20
250	258.62	3.40
270	271.29	0.40

D. Target Distance Measurement

Target distance measurements were tested in the range of 80 – 300 cm. Based on the test results as presented in Table V, the distance to the target object was measured between 80 to 300 cm, and the minimum error occurred when the target distance was 80 cm and 240 cm with an error value of 0% and an error maximum at a distance of 280 cm with an error of 3.64%.

TABLE V. TARGET DISTANCE MEASUREMENT

Actual Distance (cms)	Measurement Distance (cms)	Error (%)
80	80	0.00
100	99.96	0.04
120	119.96	0.03
140	141.76	1.24
160	160.04	0.02
180	182.54	1.39
200	199.98	0.01
220	220.04	0.01
240	240	0.00
260	259.97	0.01
280	270.08	3.67
300	290	3.44

E. Control of Obstacle Avoidance Motion

Obstacle avoidance control tests were carried out 15 times with different obstacle positions. The test was carried out in an open corridor with the obstacle directly in front of the robot 5 times during the test, the obstacle was on the left front of the robot 5 times during the test, and the obstacle was on the right front of the robot 5 times the test.

The test was carried out in an open corridor area with a length of 400 cm and a width of 320 cm, as presented in Fig. 18. The robot starts moving from the Start line straight ahead. When there is an obstacle, the robot will shift left or right depending on the given V_y Speed.

In testing with the position of the obstacle straight in front of the robot, the robot manages to avoid it by shifting to the right with varying speeds V_y , even though the position of the obstacle object is the same, that is, it is directly in front of the robot, but the response speed V_y is different, this difference is caused by Calculation of distance and angle of obstruction. Based on the tests carried out 5 times, the robot only failed to avoid obstacles caused by wheel skidding once, while in the other 4 tests, the robot managed to avoid them.

As shown in Fig. 19, in the first test, the robot detects an obstacle when the obstacle distance is 210.10 cm and the obstacle angle is 1.41 degrees. The robot begins to shift to the right by increasing the speed V_y which was originally 0 to -200.62 mm/s when the obstacle distance is getting closer, namely 190 cm with an angle of 0.94 degrees.

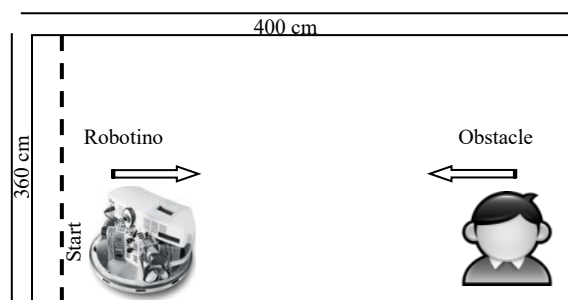
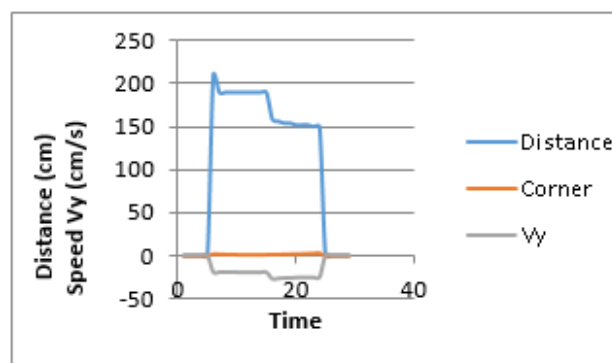


Fig. 18. Obstacle avoidance control system test scenario

Fig. 19. Response to the 1st test with a straight obstacle position in front of the robot

The robot increases the speed of V_y which was originally -200.62 mm/s, to -202.19 mm/s when the obstacle distance is 158.79 cm with an angle of 1.21 degrees, the robot increases the speed of V_y becomes -269.65 mm/s, the closer the obstacle is, the robot will continue to increase its speed V_y , so that when the obstacle distance is 150.08 with an obstacle angle of 2.07 the robot still maintains a speed V_y of -260.91, the robot's speed decreases because the obstacle angle becomes wider. In the end, if the obstacle distance is 0 and the angle is 0, then the robot will reduce the speed V_y to 0. This means that the robot will walk straight ahead again. In this test, the robot successfully avoids obstacles.

In the second test, as shown in Fig. 20, the robot detects an obstacle when the obstacle distance is 210.10 cm with an angle of 0.78 degrees and produces a V_y speed of -210.33 mm/s. The detection process in this test was better than the first one because the robot managed to detect obstacles when the obstacle distance was still far away. When the obstacle distance is 190 cm and the angle is 2.78 degrees, the robot gives a speed V_y of -200 mm/s. When the obstacle distance is 173 cm and the angle is 3.60 degrees, the speed of the robot V_y is -268.63. When the obstacle distance is 158 cm, and the angle is 4.8 degrees, the speed of the robot is V_y -267.23 until the obstacle distance is 152.17 cm and the angle is 5.63 degrees, the robot's V_y speed becomes -247.10, after that because the obstacle distance is 0 and the angle is 0, the robot reduces its V_y speed to 0, and the robot returns to walking

straight ahead. In the second experiment, the robot also managed to avoid obstacles.

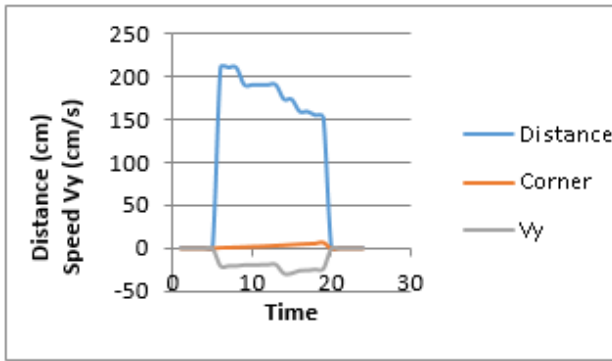


Fig. 20. Response to the 2nd test with a straight obstacle in front of the robot

The response characteristics in the third test, as shown in Fig. 21, are almost the same as the first experiment. The robot detects obstacles when the obstacle distance is 158 cm, and the angle is 0.20 degrees with speed V_y of -289.94 mm/s, when the obstacle distance is 156.52, and the angle is 1.10 the speed V_y is -276.10, and when the obstacle distance is 154.31, and the angle is 2.62 the speed V_y is -274.26, until when the obstacle distance is 150.08, and the angle is 3.24 the speed of V_y is -271.76. On the third attempt, the robot managed to avoid obstacles.

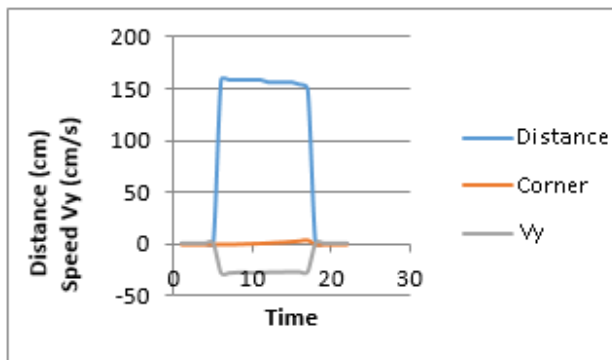


Fig. 21. Response to the 3rd test with a straight obstacle in front of the robot

In the 4th test, the robot failed to avoid obstacles. This was because the robot failed to detect an obstacle. When the robot begins to move forward, the robot's wheels slip, which causes the robot to always move obliquely to the left. That's why the robot cannot detect any obstacles because the robot's point of view has changed.

In the 5th Test, as shown in Fig. 22, the robot detects an obstacle when the obstacle distance is 232.77 cm, and the obstacle angle is 1.6 degrees with a speed of V_y -132.64 mm/s. When the obstacle distance is closer, namely 173.09 cm with an obstacle angle of 1.14 degrees, the robot increases the speed of V_y to -312.45 mm/s. It will further increase to -404.67 mm/s. In the 5th test, the robot managed to avoid obstacles. In testing with the obstacle positioned straight in front of the robot, the robot always dodges to the right with different velocities V_y . Fig. 23 shows the trajectory of the test results, where the robot shifts to the right with an average shift width of 45.5 cm from the previous robot position.

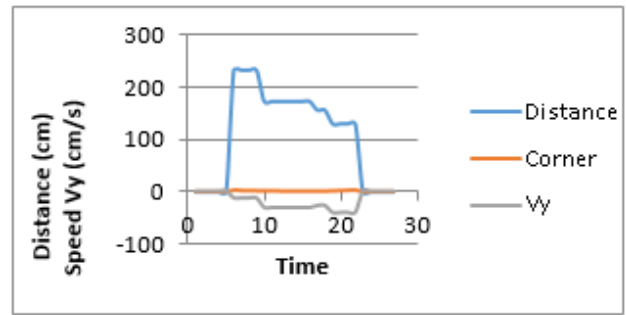


Fig. 22. Response to the 5th test with a straight obstacle in front of the robot

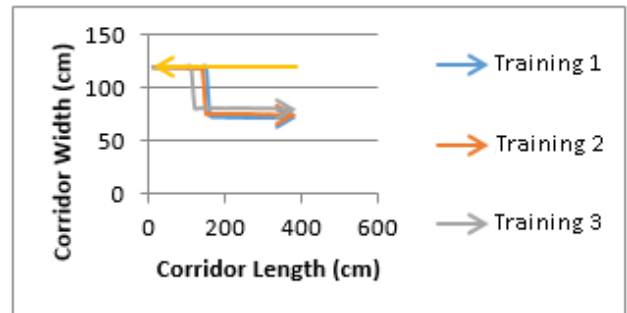


Fig. 23. The trajectory generated when obstacles are straight in front of the robot

After testing with the obstacle position straight in front of the robot is complete, the next test is to change the position of the obstacle to the right of the robot. This test was carried out 5 times, where in this test, the robot managed to avoid it without even experiencing a collision.

As shown in Fig. 24, the 6th test with the obstacle position in front of the right of the robot, the robot detects an obstacle with a distance of 190 cm and an obstacle angle of -4.09 degrees, the robot starts to shift to the left by increasing the speed V_y from 0 mm/s to 221.86 mm/s, when the obstacle distance is 158.79 cm, and the obstacle angle is -4.01 degrees, the robot increases the speed of V_y to 300.51 mm/s, until when the distance is 156.52 cm with the obstacle angle -4.44 degrees the robot maintains the speed V_y of 297.35 mm/s, after that the robot returns walk straight ahead. In this test, the robot managed to avoid obstacles.

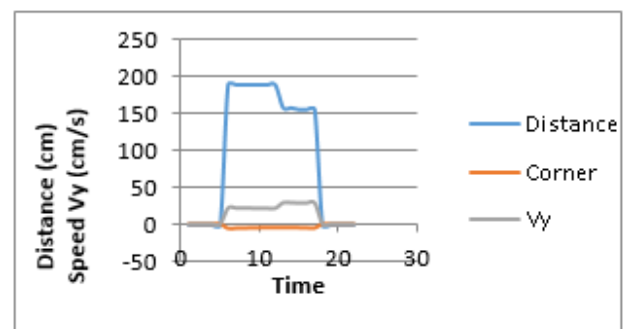


Fig. 24. Response to the 6th test with the position of the obstacle in the front right of the robot

The response of the 7th test is shown in Fig. 25. When the robot detects an obstacle with a distance of 210.10 cm and an obstacle angle of -5.88 degrees, the robot begins to increase its speed V_y from 0 mm/s to 239.63 mm/s when the obstacle

distance is 190 cm, and the obstacle angle is - 5.65 degrees, the robot maintains a speed of V_y of 228.06 mm/s until when the distance is 173.09 cm with an obstacle angle of -5.61 degrees, the robot increases the speed of V_y to 339.93 mm/s, after which the robot continues to walk straight ahead. In this test, the robot managed to avoid obstacles.

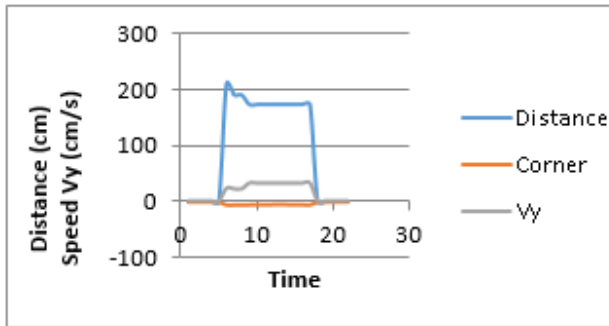


Fig. 25. Response to the 7th test with the obstacle positioning in front of the right robot

The 8th test is shown in Fig. 26, the robot begins to detect obstacles at a distance of 258.62 cm with an obstacle angle of -7.56 degrees, so the robot begins to increase the speed of V_y which was originally 0 mm/s to 69.33 mm/s. When the obstacle distance is 232.77 cm and the obstacle angle is -6.8 degrees, the robot increases the speed of V_y to 166.91 mm/s. When the obstacle distance is 210.10 cm, and the obstacle angle is -7.41 degrees, the robot increases the speed V_y to 245.70 mm/s. After that, the robot returns to moving straight in front. In this test, the robot managed to avoid obstacles.

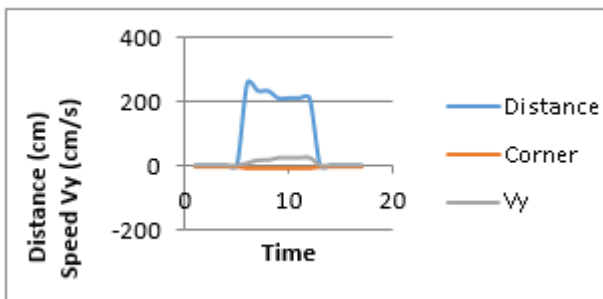


Fig. 26. Response to the 8th test with the position of the obstacle in the front right of the robot

In the 9th and 10th tests, the response characteristics were almost the same as in the 6th test. In the 9th test, the robot began to detect obstacles when the obstacle distance was 173.09 cm and the obstacle angle was -7.09 degrees with a speed V_y of 345.86 mm/s, whereas, in the in-10th test, the robot began to detect obstacles when the obstacle distance is 158.79 cm, and the obstacle angle is -4.48 with a V_y speed of 302.69 mm/s. In the 9th and 10th tests, the robot also managed to avoid obstacles.

In testing with the obstacle position in front of the robot's right, the robot always dodges to the left at different speeds V_y . Fig. 27 shows the trajectory of the test results, where the robot shifts to the right with an average shift width of 25.23 cm from the previous robot position.

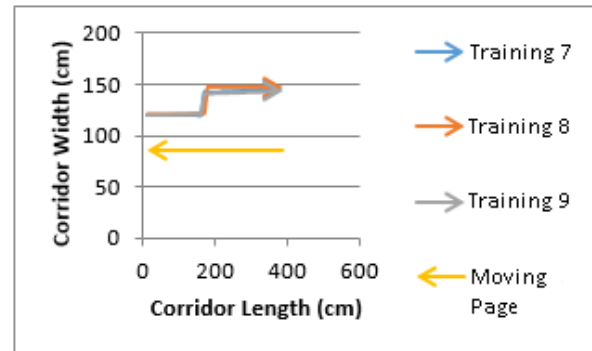


Fig. 27. The trajectory generated when the position of the obstacle in front and right of the robot

The next test is to change the position of the obstacle to the left of the robot. Testing with an obstacle position like this was also carried out 5 times, and in each test, the robot managed to avoid the obstacle. The characteristics of the response in testing with the obstacle position to the left of the robot are almost the same as the obstacle is straight in front of the robot, namely, the robot shifts to the right.

In the 11th test, as shown in Fig. 28, the robot begins to detect obstacles at a distance of 190 cm with an obstacle angle of 8.52 degrees, then the resulting speed V_y is -172.45 mm/s when the distance is 156.52 cm with an obstacle angle of 11.72 degrees, then the robot increases the speed of V_y to -233.35 mm/s. After that, the robot moves forward again. In the 11th test, the robot managed to avoid obstacles.

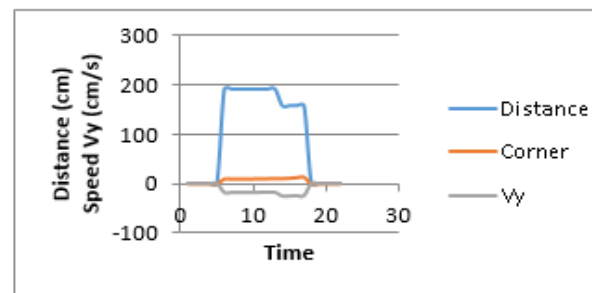


Fig. 28. Response to the 11th test with the position of the obstacle in front and left of the robot

As shown in Fig. 29, in the 12th test, the robot begins to detect obstacles at a distance of 210.10 cm with an obstacle angle of 6.88 degrees, the given speed V_y is -188.69 mm/s when the distance is 173.09 cm with an obstacle angle of 9.06 degrees, then the robot increases the speed of V_y to -280.98 mm/s. After that, the robot moves forward again. In the 12th test, the robot managed to avoid obstacles.

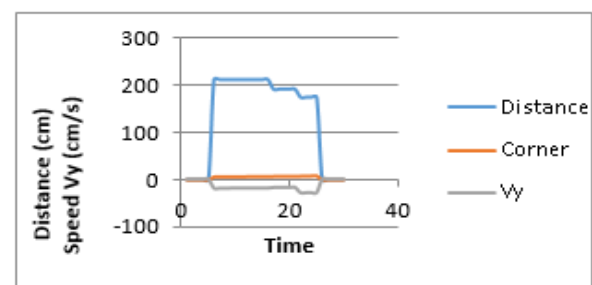


Fig. 29. Response to the 12th test with the position of the obstacle in front and left of the robot

In the 13th test, as shown in Fig. 30, obstacles began to be detected at a distance of 232.77 cm with an angle of 7.81 degrees, and the robot began to increase the speed of V_y by -107.71 mm/s when the obstacle distance was 210.10 cm and the angle of 7.62 degrees the robot increased the speed of V_y to -109.13 mm/s, then when the obstacle distance is 173.09 cm with an obstacle angle of 8.48 as shown in Fig. 30, the robot continues to increase the speed of V_y to -283.44 mm/s, after that the robot moves forward again. In this test, the robot managed to avoid obstacles.

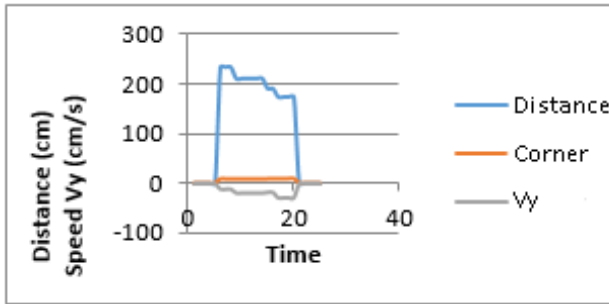


Fig. 30. Response to the 13th test with the position of the obstacle in front and left of the robot

In the 14th test, as shown in Fig. 31, the robot begins to detect obstacles when the obstacle distance is 190 cm with an obstacle angle of 8.55 degrees. Then the robot increases the speed of V_y to -172.50 mm/s until when the obstacle distance is 158.79 and the obstacle angle is 8.83, the speed of V_y increases become -250.31. After that, the robot moves forward again. In the 14th test, the robot managed to avoid obstacles.

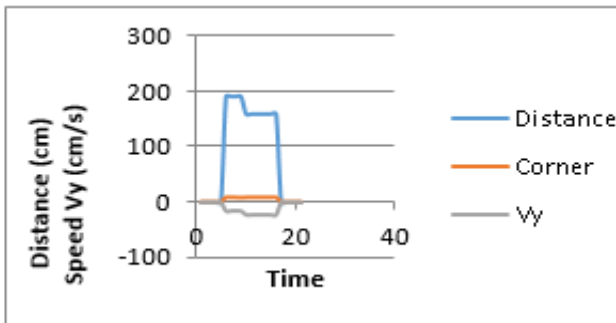


Fig. 31. Response to the 14th test with the position of the obstacle in front and left of the robot

The response characteristic of the 15th test is almost the same as the 12th test. Namely, the robot begins to detect obstacles when the obstacle distance is 210 and the obstacle angle is 5.00 degrees. At the same time, the robot increases the speed of V_y to -196.32 mm/s. When the obstacle distance is 173.09, and the obstacle angle is 6.84, the robot increases the speed of V_y to -296.92 mm/s, after which the robot moves forward again.

In testing with the position of the obstacle in front of the robot's left, the robot always dodges to the left at different speeds V_y . Fig. 32 shows the trajectory of the test results, where the robot shifts to the right with an average shift width of 35.78 cm from the previous robot position.

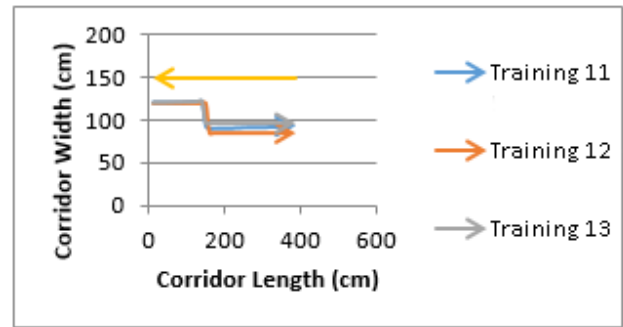


Fig. 32. Trajectory generated when obstacle position in front and left of the robot

Based on the results of the tests that have been carried out, as shown in Table VI, in the 15 tests, only once did the robot fails to avoid obstacles, namely in the 4th test, which was caused by wheel slippage. As shown in Fig. 33, the robot process avoids obstacles by shifting to the right or left depending on the given V_y value. The average robot starts to dodge when the obstacle distance is 173 – 150 cm with an average V_y speed of 274 mm/s.



Fig. 33. The robot process avoids obstacles when the object is straight in front of the robot

TABLE VI. RESULTS OF OBSTACLE AVOIDANCE EXPERIMENTS

No. Experiment	Obstacle Position	Result
1	Straight Front	succeed
2	Straight Front	succeed
3	Straight Front	succeed
4	Straight Front	Failed
5	Straight Front	succeed
6	Front right	succeed
7	Front right	succeed
8	Front right	succeed
9	Front right	succeed
10	Front right	succeed
11	Front left	Succeed
12	Front left	Succeed
13	Front left	Succeed
14	Front left	Succeed
15	Front left	Succeed

F. Obstacle Avoidance and Target Toward Motion

The next test is to combine obstacle avoidance control with control towards the target, as shown in Fig. 34. If the obstacle avoidance control that is controlled is the speed of the robot V_y , then the control towards the target that is controlled is the angular velocity.

These two controls are integrated and tested 3 times, with the target position being right straight in front of the robot, the target in front of the right of the robot, and the target in front of the robot's left, while the position of the obstacle that comes also varies. Of the three tests, the robot managed to avoid obstacles and arrived at the target without experiencing a collision.

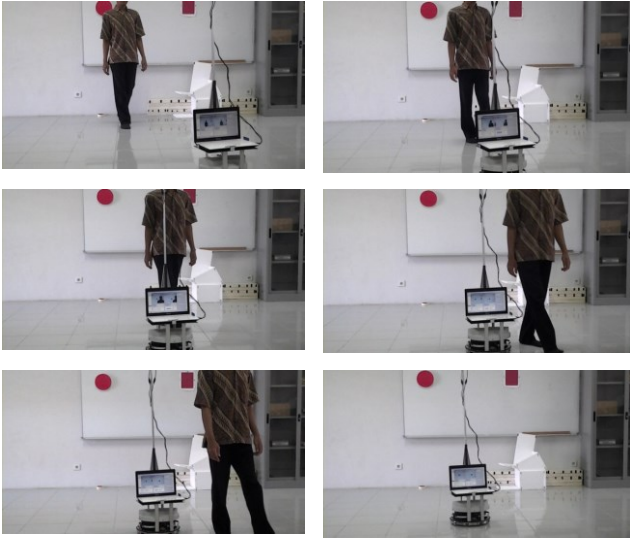
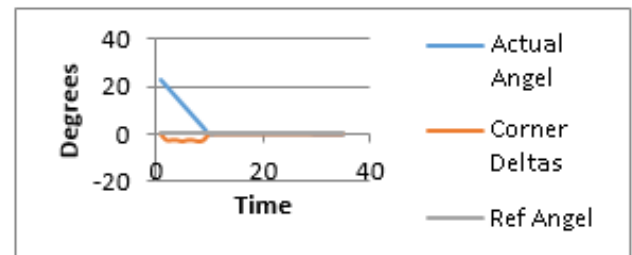


Fig. 34. The robot process avoids obstacles and heads to the target

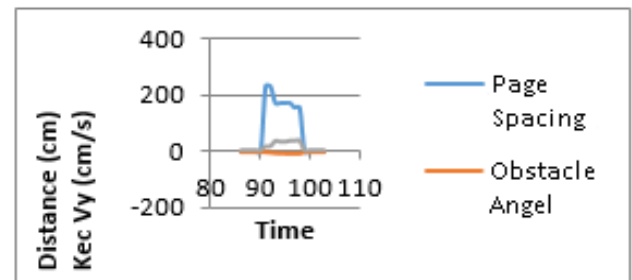
As shown in Fig. 34, when the robot starts moving, what is calculated first is the target angle and the angle delta. If the target angle is not equal to 0, this means that the direction facing the robot the target is not straight. Therefore, the robot will adjust the direction toward the target until the target angle is equal to 0.

During testing, as shown in Fig. 35 (a), the first time the robot moves, the target angle is calculated. When the target angle reaches 22.53 degrees to the left of the robot, the robot will adjust the angular speed to rotate to the left by 22.53 degrees until the target angle is 0. After the angle of the object is equal to 0, this means that the direction the robot is facing is straight with the object. Thus, the robot will reduce the angular velocity to 0 and move straight again. The same thing is also done when the target is positioned at -22.53 degrees to the right of the robot. The robot will set the angular velocity to rotate to the right by -22.53 degrees until the target angle is 0. In essence, whatever the angle of the target, the robot will adjust the direction towards the target by how to adjust the angular velocity until the target angle is 0. The direction of the robot towards the desired target is 0 degrees, but in reality, on several tests, the robot can never reach that value. The value of the robot's direction towards the smallest target produced during the test is 0.03 degrees. If the target angle is 0, you can be sure that the target is being covered by a passing obstacle.

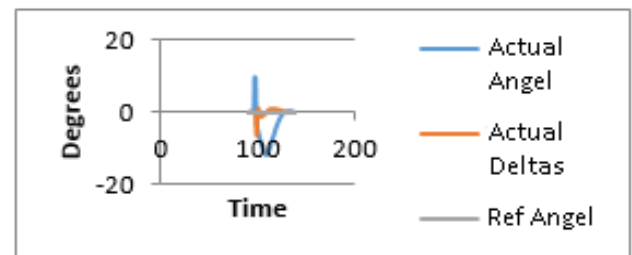
When the robot travels towards the target, as shown in Fig. 34 (b), the robot detects an obstacle with an obstacle distance of 232.77 and an obstacle angle of -0.38, so that the robot produces a speed V_y of 140.70 mm/s. When the obstacle distance is 156.52, and the obstacle angle is -6.20, the robot increases the speed of V_y to 340.22 mm/s. When avoiding obstacles, the robot still calculates the target's position so that the target does not disappear from the robot's view, meaning that both controls work simultaneously. As shown in Fig. 34 (c), after successfully avoiding obstacles, the robot returns to correct the direction towards the target so that the target angle returns to 0. Compared to previous research, this study has successfully been able to produce an obstacle avoidance method for obstacles moving like walking humans.



(a)



(b)



(c)

Fig. 35. Response integration of obstacle avoidance control and target control

V. CONCLUSION

Based on the research that has been done, the distance measurement process is obtained by determining the focal length and baseline divided by the difference in the midpoints of the two frames. To support the ability to avoid obstacles, it is important to improve the accuracy of the distance measurement to obstacles and targets. The largest error value for measuring the distance to an obstacle is 3.40% when the obstacle distance is 250 cm, while the biggest error for measuring the target distance is 1.39% when the target distance is 180 cm. The smallest error for measuring the

obstacle distance is 0.00% when the obstacle distance is 190 cm, while the smallest error for measuring the target distance is 0.00% when the target distance is 80 cm and 240 cm.

The trajectory of the robot will change when the robot identifies a target object based on the target angle and angle delta values, where these values are processed by ANFIS to generate angular velocity. The smallest value of the robot's direction towards the target during several tests is 0.03 degrees. When the robot detects an obstacle, the robot avoids it by shifting to the right or left depending on the speed of V_y , the speed of V_y is obtained from processing the distance and angle of the obstacle. The average robot avoids obstacles when the object is 173 – 150 cm away with an average speed of V_y 274 mm/s. While moving toward the target and moving to avoid obstacles, the robot's movements are very smooth. This is because the robot used in this study is a robot with omnidirectional wheels.

From the research results, this study contributes to improving the robot navigation system in obstacle avoidance of moving obstacles based on visual sensors by utilizing ANFIS. The research carried out still has a limitation. The drawback is that the forward direction speed or x -axis speed is still done manually, so the robot does not care about the distance of objects that are far or near. The speed used remains the same. Supposedly if the object starts to get closer, then the speed of the forward direction starts to decrease little by little so that when evading the robot's movement, it is even smoother. For future research, control of the robot's speed is required to be controlled automatically.

REFERENCES

- [1] T. E. Mora and E. N. Sanchez, "Fuzzy logic-based real-time navigation controller for a mobile robot," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, vol. 1, pp. 612–617, 1998, doi: 10.1109/IROS.1998.724686.
- [2] L. M. Waghmare, P. Tallapragada, and N. Bidwai, "Reactive Navigation of Autonomous Vehicle using Neuro-Fuzzy Controller," in *2006 IEEE International Conference on Industrial Technology*, pp. 2681–2685, 2006, doi: 10.1109/ICIT.2006.372675.
- [3] S. Seghour and M. Tadjine, "Consensus-based approach and reactive fuzzy navigation for multiple no-holonomic mobile robots," in *2017 6th International Conference on Systems and Control (ICSC)*, pp. 492–497, 2017, doi: 10.1109/ICoSC.2017.7958658.
- [4] T. Shen and J. Zhai, "Reactive Obstacle Avoidance Strategy Based on Fuzzy Neural Network and Arc Trajectory," in *2019 Chinese Automation Congress (CAC)*, pp. 4792–4796, 2019, doi: 10.1109/CAC48633.2019.8996374.
- [5] Y. Wang and Y. Yuan, "A dynamic reactive power compensation method for high-power and high-voltage electronic motors based on self-adaptive fuzzy PID control," in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*, pp. 10–15, 2016, doi: 10.1109/CGNCC.2016.7828749.
- [6] E. Ruiz, R. Acuña, P. Vélez, and G. Fernández-López, "Hybrid Deliberative Reactive Navigation System for Mobile Robots Using ROS and Fuzzy Logic Control," in *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, pp. 67–72, 2015, doi: 10.1109/LARS-SBR.2015.24.
- [7] P. Chand, "Fuzzy reactive control for wheeled mobile robots," in *2015 6th International Conference on Automation, Robotics and Applications (ICARA)*, pp. 167–172, 2015, doi: 10.1109/ICARA.2015.7081142.
- [8] H. Zerfa and W. Nouibat, "Fuzzy reactive navigation for autonomous mobile robot with an offline adaptive neuro fuzzy system," in *3rd International Conference on Systems and Control*, pp. 950–955, 2013, doi: 10.1109/ICoSC.2013.6750971.
- [9] A. Melendez and O. Castillo, "Optimization of type-2 fuzzy reactive controllers for an autonomous mobile robot," in *2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)*, pp. 207–211, 2012, doi: 10.1109/NaBIC.2012.6402263.
- [10] E. Baklouti, M. Jallouli, N. Ben Amor, S. Titi, and A. Nafti, "Autonomous mobile robot navigation coupling fuzzy logic and reactive DVZ 3D obstacle avoidance control," in *2015 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–6, 2015, doi: 10.1109/INISTA.2015.7276748.
- [11] Y. Lv and P. Jiang, "The Design of Indoor Mobile Robot Navigation System Based on UWB Location," in *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, pp. 334–338, 2018, doi: 10.1109/IMCCC.2018.00077.
- [12] A. Itta, G. Attolico, and A. Distanto, "Combining reactive behaviors using a hierarchy of fuzzy controllers," in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ- IEEE 2000 (Cat. No.00CH37063)*, vol. 2, pp. 1041–1044, 2000, doi: 10.1109/FUZZY.2000.839194.
- [13] M. A. O. Mendez and J. A. F. Madrigal, "Fuzzy Logic User Adaptive Navigation Control System For Mobile Robots In Unknown Environments," in *2007 IEEE International Symposium on Intelligent Signal Processing*, pp. 1–6, 2007, doi: 10.1109/WISP.2007.4447633.
- [14] I. I. Ismail and M. F. Nordin, "Reactive navigation of autonomous guided vehicle using fuzzy logic," in *Student Conference on Research and Development*, pp. 153–156, 2002, doi: 10.1109/SCORED.2002.1033080.
- [15] N. Zhang, D. Beetner, D. C. Wunsch, B. Hemmelman, and A. Hasan, "An Embedded Real-Time Neuro-Fuzzy Controller for Mobile Robot Navigation," in *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ '05.*, pp. 319–324, 2005, doi: 10.1109/FUZZY.2005.1452413.
- [16] G. Castellano, G. Attolico, E. Stella, and A. Distanto, "Reactive navigation by fuzzy control," in *Proceedings of IEEE 5th International Fuzzy Systems*, vol. 3, pp. 2143–2149, 1996, doi: 10.1109/FUZZY.1996.552796.
- [17] W. L. Xu and S. K. Tso, "Sensor-based fuzzy reactive navigation of a mobile robot through local target switching," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 29, no. 3, pp. 451–459, 1999, doi: 10.1109/5326.777079.
- [18] M. M. Joshi and M. A. Zaveri, "Fuzzy Based Autonomous Robot Navigation System," in *2009 Annual IEEE India Conference*, pp. 1–4, 2009, doi: 10.1109/INDCON.2009.5409419.
- [19] M. Alwan, P. Y. K. Cheung, A. Saleh, and N. E. C. Obeid, "Combining goal-directed, reactive and reflexive navigation in autonomous mobile robots," in *1996 Australian New Zealand Conference on Intelligent Information Systems. Proceedings. ANZIS 96*, pp. 346–349, 1996, doi: 10.1109/ANZIS.1996.573982.
- [20] A. A. S. Al-Jumaily and S. H. M. Amin, "Fuzzy logic based behaviors blending for intelligent reactive navigation of walking robot," in *ISSPA '99. Proceedings of the Fifth International Symposium on Signal Processing and its Applications (IEEE Cat. No.99EX359)*, vol. 1, pp. 155–158, 1999, doi: 10.1109/ISSPA.1999.818136.
- [21] M. Skubic, S. Graves, and J. Mollenhauer, "Design of a two-level fuzzy controller for a reactive miniature mobile robot," in *Third International Conference on Industrial Fuzzy Control and Intelligent Systems*, pp. 224–227, 1993, doi: 10.1109/IFIS.1993.324183.
- [22] J. L. Overholt, G. R. Hudas, and K. C. Cheok, "A modular neural-fuzzy controller for autonomous reactive navigation," in *NAFIPS 2005 - 2005 Annual Meeting of the North American Fuzzy Information Processing Society*, pp. 121–126, 2005, doi: 10.1109/NAFIPS.2005.1548519.
- [23] W. Li, "Fuzzy logic based robot navigation in uncertain environments by multisensor integration," in *Proceedings of 1994 IEEE International Conference on MFI '94. Multisensor Fusion and Integration for Intelligent Systems*, pp. 259–264, 1994, doi: 10.1109/MFI.1994.398444.
- [24] N. Melik and N. Slimane, "Autonomous navigation with obstacle avoidance of tricycle mobile robot based on fuzzy controller," in *2015*

- 4th International Conference on Electrical Engineering (ICEE), pp. 1–4, 2015, doi: 10.1109/INTEE.2015.7416799.
- [25] O. Aycard, F. Charpillet, and J.-P. Haton, “A new approach to design fuzzy controllers for mobile robots navigation,” in *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA '97. 'Towards New Computational Principles for Robotics and Automation,'* pp. 68–73, 1997, doi: 10.1109/CIRA.1997.613840.
- [26] B. C. Arrue, F. Cuesta, R. Braunstingl, and A. Ollero, “Fuzzy behaviors combination to control a nonholonomic mobile robot using virtual perception memory,” in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 3, pp. 1239–1244, 1997, doi: 10.1109/FUZZY.1997.619465.
- [27] W. Li and X. Feng, “Behavior fusion for robot navigation in uncertain environments using fuzzy logic,” in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, vol. 2, pp. 1790–1796, 1994, doi: 10.1109/ICSMC.1994.400110.
- [28] A. A. S. Al-Jumaily and S. H. M. Amin, “Behaviors blending for intelligent reactive navigation of climbing robot,” in *2000 26th Annual Conference of the IEEE Industrial Electronics Society. IECON 2000. 2000 IEEE International Conference on Industrial Electronics, Control and Instrumentation. 21st Century Technologies*, vol. 2, pp. 795–799, 2000, doi: 10.1109/IECON.2000.972224.
- [29] X. Yang, M. Moallem, and R. V Patel, “A fuzzy logic-based reactive navigation algorithm for mobile robots,” in *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, pp. 197–202, 2005, doi: 10.1109/CCA.2005.1507124.
- [30] Y. Duan and Xin-Hexu, “Fuzzy reinforcement learning and its application in robot navigation,” in *2005 International Conference on Machine Learning and Cybernetics*, vol. 2, pp. 899–904, 2005, doi: 10.1109/ICMLC.2005.1527071.
- [31] W. L. Xu, S. K. Tso, and Y. H. Fung, “Sensor-based reactive navigation of a mobile robot through local target switching,” in *1997 8th International Conference on Advanced Robotics. Proceedings. ICAR '97*, pp. 361–366, 1997, doi: 10.1109/ICAR.1997.620207.
- [32] A. M. Anvar, “Intelligent navigation process for autonomous underwater vehicles (AUVs) using time-based fuzzy temporal reasoning,” in *10th International Symposium on Temporal Representation and Reasoning, 2003 and Fourth International Conference on Temporal Logic. Proceedings.*, pp. 56–61, 2003, doi: 10.1109/TIME.2003.1214880.
- [33] H. Maaref and C. Barret, “Fuzzy help in mobile robot navigation,” in *Proceedings IEEE Conference on Industrial Automation and Control Emerging Technology Applications*, pp. 387–390, 1995, doi: 10.1109/IACET.1995.527593.
- [34] C. Barret, M. Benreguieg, and H. Maaref, “Fuzzy agents for reactive navigation of a mobile robot,” in *Proceedings of 1st International Conference on Conventional and Knowledge Based Intelligent Electronic Systems. KES '97*, vol. 2, pp. 649–658, 1997, doi: 10.1109/KES.1997.619449.
- [35] A. A. S. Al-Jumaily, S. H. M. Amin, and M. Khalil, “A fuzzy multi-behaviour reactive obstacle avoidance navigation for a climbing mobile robot,” in *Proceedings of IEEE International Conference on Intelligent Engineering Systems*, pp. 147–152, 1997, doi: 10.1109/INES.1997.632408.
- [36] W. Li, “A hybrid neuro-fuzzy system for sensor based robot navigation in unknown environments,” in *Proceedings of 1995 American Control Conference - ACC '95*, vol. 4, pp. 2749–2753, 1995, doi: 10.1109/ACC.1995.532349.
- [37] G. Mester, “Obstacle Avoidance of Mobile Robots in Unknown Environments,” in *2007 5th International Symposium on Intelligent Systems and Informatics*, pp. 123–127, 2007, doi: 10.1109/SISY.2007.4342637.
- [38] M. Benreguieg, H. Maaref, and C. Barret, “Fusion of fuzzy agents for the reactive navigation of a mobile robot,” in *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robot and Systems. Innovative Robotics for Real-World Applications. IROS '97*, vol. 1, pp. 388–394, 1997, doi: 10.1109/IROS.1997.649091.
- [39] M. Dupre and S. X. Yang, “Two-Stage Fuzzy Logic-Based Controller for Mobile Robot Navigation,” in *2006 International Conference on Mechatronics and Automation*, pp. 745–750, 2006, doi: 10.1109/ICMA.2006.257683.
- [40] A. Jayasiri, G. K. I. Mann, and R. G. Gosine, “Supervisory control of Fuzzy Discrete Event Systems and its application to mobile robot navigation,” in *2009 Canadian Conference on Electrical and Computer Engineering*, pp. 1147–1151, 2009, doi: 10.1109/CCECE.2009.5090305.
- [41] D. Shi, M. F. Selekw, E. G. Collins, and C. A. Moore, “Fuzzy behavior navigation for an unmanned helicopter in unknown environments,” in *2005 IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3897–3902, 2005, doi: 10.1109/ICSMC.2005.1571754.
- [42] G. Mondelli, G. Castellano, G. Attolico, E. Stella, and A. Distante, “Self-tuning fuzzy logic controller for reactive navigation,” in *Proceedings of Conference on Intelligent Vehicles*, pp. 87–92, 1996, doi: 10.1109/IVS.1996.566358.
- [43] M. F. Selekw, D. D. Dunlap, and E. G. Collins, “Implementation of Multi-valued Fuzzy Behavior Control for Robot Navigation in Cluttered Environments,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 3688–3695, 2005, doi: 10.1109/ROBOT.2005.1570682.
- [44] B. B. K. Reddy, B. Kimiaghala, and A. Homaifar, “Reactive real time behavior for mobile robots in unknown environments,” in *2004 IEEE International Symposium on Industrial Electronics*, vol. 1, pp. 693–697, 2004, doi: 10.1109/ISIE.2004.1571890.
- [45] S. Kundu and D. R. Parhi, “Behavior-based navigation of multiple robotic agents using hybrid-fuzzy controller,” in *2010 International Conference on Computer and Communication Technology (ICCCCT)*, pp. 706–711, 2010, doi: 10.1109/ICCCCT.2010.5640437.
- [46] A. Safiotti, “Fuzzy logic in autonomous robotics: behavior coordination,” in *Proceedings of 6th International Fuzzy Systems Conference*, vol. 1, pp. 573–578, 1997, doi: 10.1109/FUZZY.1997.616430.
- [47] H. Maaref and C. Barret, “Progressive optimization of a fuzzy inference system,” in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 1, pp. 47–52, 2001, doi: 10.1109/NAFIPS.2001.944225.
- [48] J. C. Vega Oliver and P. F. Huamani Navarrete, “Fuzzy control to simulate 4 autonomous navigation behaviors in a differential-drive mobile robot,” in *2017 IEEE International Conference on Aerospace and Signals (INCAS)*, pp. 1–4, 2017, doi: 10.1109/INCAS.2017.8123498.
- [49] J. Yung-Jen Hsu, D.-C. Lo, and S.-C. Hsu, “Fuzzy control for behavior-based mobile robots,” in *NAFIPS/IFIS/NASA '94. Proceedings of the First International Joint Conference of The North American Fuzzy Information Processing Society Biannual Conference. The Industrial Fuzzy Control and Intelligent*, pp. 209–213, 1994, doi: 10.1109/IJCF.1994.375097.
- [50] A. Zhu and S. X. Yang, “A goal-oriented fuzzy reactive control for mobile robots with automatic rule optimization,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3688–3693, 2010, doi: 10.1109/IROS.2010.5651799.
- [51] A. Zhu and S. X. Yang, “A fuzzy logic approach to reactive navigation of behavior-based mobile robots,” in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, pp. 5045–5050, 2004, doi: 10.1109/ROBOT.2004.1302517.
- [52] S. Kundu and D. R. Parhi, “Fuzzy based reactive navigational strategy for mobile agent,” in *2010 International Conference on Industrial Electronics, Control and Robotics*, pp. 12–17, 2010, doi: 10.1109/IECR.2010.5720149.
- [53] S. G. Goodridge and R. C. Luo, “Fuzzy behavior fusion for reactive control of an autonomous mobile robot: MARGE,” in *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, vol. 2, pp. 1622–1627, 1994, doi: 10.1109/ROBOT.1994.351358.
- [54] K. C. Tan, K. K. Tan, T. H. Lee, S. Zhao, and Y. J. Chen, “Autonomous robot navigation based on fuzzy sensor fusion and reinforcement learning,” in *Proceedings of the IEEE International Symposium on Intelligent Control*, pp. 182–187, 2002, doi: 10.1109/ISIC.2002.1157759.
- [55] M. A. Sotelo *et al.*, “Vehicle fuzzy driving based on DGPS and vision,” in *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference (Cat. No. 01TH8569)*, vol. 3, pp. 1472–1477, 2001, doi: 10.1109/NAFIPS.2001.943766.

- [56] A. Howard, B. Werger, and H. Seraji, "Integrating terrain maps into a reactive navigation strategy," in *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, vol. 2, pp. 2012–2017, 2003, doi: 10.1109/ROBOT.2003.1241889.
- [57] H. Liu, P. Hu, Y. Luo, and C. Li, "A goal-oriented fuzzy reactive control method for mobile robot navigation in unknown environment," in *2009 IEEE International Symposium on Industrial Electronics*, pp. 1950–1955, 2009, doi: 10.1109/ISIE.2009.5219773.
- [58] H. A. Hagaras, "A hierarchical type-2 fuzzy logic control architecture for autonomous mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 12, no. 4, pp. 524–539, 2004, doi: 10.1109/TFUZZ.2004.832538.
- [59] S. Kundu and R. P. Dayal, "A fuzzy approach towards behavioral strategy for navigation of mobile agent," in *INTERACT-2010*, pp. 292–297, 2010, doi: 10.1109/INTERACT.2010.5706164.
- [60] B.-K. Shim, J.-H. Kim, I.-M. Park, and S.-H. Han, "An intelligent control of non-holonomic mobile robot based on fuzzy perception," in *ICCAS 2010*, pp. 2111–2114, 2010, doi: 10.1109/ICCAS.2010.5670182.
- [61] A. A. S. Al-Jumaily and S. H. M. Amin, "Blending multi-behaviors of intelligent reactive navigation for legged walking robot in unstructured environment," in *2000 TENCON Proceedings. Intelligent Systems and Technologies for the New Millennium (Cat. No.00CH37119)*, vol. 2, pp. 297–302, 2000, doi: 10.1109/TENCON.2000.888751.
- [62] J. G. Ortega and E. F. Camacho, "Mobile robot navigation in a partially structured static environment, using neural predictive control," *Control Eng Pract*, vol. 4, no. 12, pp. 1669–1679, 1996, doi: [https://doi.org/10.1016/S0967-0661\(96\)00184-0](https://doi.org/10.1016/S0967-0661(96)00184-0).
- [63] L. E. Zarate, M. Becker, B. D. M. Garrido, and H. S. C. Rocha, "An artificial neural network structure able to obstacle avoidance behavior used in mobile robots," in *IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02*, vol. 3, pp. 2457–2461, 2002, doi: 10.1109/IECON.2002.1185358.
- [64] O. Gamal, X. Cai, and H. Roth, "Learning from Fuzzy System Demonstration: Autonomous Navigation of Mobile Robot in Static Indoor Environment using Multimodal Deep Learning," in *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*, pp. 218–225, 2020, doi: 10.1109/ICSTCC50638.2020.9259786.
- [65] A. S. Lafmejani, S. Berman, and G. Fainekos, "NMPC-LBF: Nonlinear MPC with Learned Barrier Function for Decentralized Safe Navigation of Multiple Robots in Unknown Environments," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 10297–10303, 2022, doi: 10.1109/IROS47612.2022.9981177.
- [66] G. Chen *et al.*, "Robot Navigation with Map-Based Deep Reinforcement Learning," *2020 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, pp. 1–6, 2020, doi: 10.1109/ICNSC48988.2020.9238090.
- [67] Y. Pan and J. Wang, "A neurodynamic optimization approach to nonlinear model predictive control," in *2010 IEEE International Conference on Systems, Man and Cybernetics*, pp. 1597–1602, 2010, doi: 10.1109/ICSMC.2010.5642367.
- [68] N. Hirose, F. Xia, R. Martín-Martín, A. Sadeghian, and S. Savarese, "Deep Visual MPC-Policy Learning for Navigation," *IEEE Robot Autom Lett*, vol. 4, no. 4, pp. 3184–3191, 2019, doi: 10.1109/LRA.2019.2925731.
- [69] T. Ono and T. Kanamaru, "Prediction of pedestrian trajectory based on long short-term memory of data," in *2021 21st International Conference on Control, Automation and Systems (ICCAS)*, pp. 1676–1679, 2021, doi: 10.23919/ICCAS52745.2021.9649937.
- [70] T. Kim, H. Lee, S. Hong, and W. Lee, "TOAST: Trajectory Optimization and Simultaneous Tracking Using Shared Neural Network Dynamics," *IEEE Robot Autom Lett*, vol. 7, no. 4, pp. 9747–9754, 2022, doi: 10.1109/LRA.2022.3184769.
- [71] Y. Guo, R. Jena, D. Hughes, M. Lewis, and K. Sycara, "Transfer Learning for Human Navigation and Triage Strategies Prediction in a Simulated Urban Search and Rescue Task," in *2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN)*, pp. 784–791, 2021, doi: 10.1109/RO-MAN50785.2021.9515526.
- [72] A. Polevoy, C. Knuth, K. M. Popek, and K. D. Katyal, "Complex Terrain Navigation via Model Error Prediction," *2022 International Conference on Robotics and Automation (ICRA)*, pp. 9411–9417, 2022, doi: 10.1109/ICRA46639.2022.9811644.
- [73] S. Lai, M. Lan, and B. M. Chen, "Model Predictive Local Motion Planning With Boundary State Constrained Primitives," *IEEE Robot Autom Lett*, vol. 4, no. 4, pp. 3577–3584, 2019, doi: 10.1109/LRA.2019.2928255.
- [74] F. Gauthier-Clerc, A. Hill, J. Laneurit, R. Lenain, and É. Lucet, "Online velocity fluctuation of off-road wheeled mobile robots: A reinforcement learning approach," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2421–2427, 2021, doi: 10.1109/ICRA48506.2021.9560816.
- [75] E. P. Ferreira and V. M. Miranda, "Development of static neural networks as full predictors or controllers for multi-articulated mobile robots in backward movements - new models and tools," in *2011 9th IEEE International Conference on Control and Automation (ICCA)*, pp. 985–990, 2011, doi: 10.1109/ICCA.2011.6138028.
- [76] N. D. Weerakkodi Mudalige *et al.*, "DogTouch: CNN-based Recognition of Surface Textures by Quadruped Robot with High Density Tactile Sensors," in *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, pp. 1–5, 2022, doi: 10.1109/VTC2022-Spring54318.2022.9860815.
- [77] X. Zou, B. Sun, D. Zhao, Z. Zhu, J. Zhao, and Y. He, "Multi-Modal Pedestrian Trajectory Prediction for Edge Agents Based on Spatial-Temporal Graph," *IEEE Access*, vol. 8, pp. 83321–83332, 2020, doi: 10.1109/ACCESS.2020.2991435.
- [78] I. Engedy and G. Horvath, "Artificial neural network based mobile robot navigation," in *2009 IEEE International Symposium on Intelligent Signal Processing*, pp. 241–246, 2009, doi: 10.1109/WISP.2009.5286557.
- [79] M. K. Bugeja, S. G. Fabri, and L. Camilleri, "Dual Adaptive Dynamic Control of Mobile Robots Using Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 39, no. 1, pp. 129–141, 2009, doi: 10.1109/TSMCB.2008.2002851.
- [80] J. Yuan, H. Wang, C. Lin, D. Liu, and D. Yu, "A Novel GRU-RNN Network Model for Dynamic Path Planning of Mobile Robot," *IEEE Access*, vol. 7, pp. 15140–15151, 2019, doi: 10.1109/ACCESS.2019.2894626.
- [81] M. J. Er, T. P. Tan, and S. Y. Loh, "Control of a mobile robot using generalized dynamic fuzzy neural networks," *Microprocess Microsyst*, vol. 28, no. 9, pp. 491–498, 2004, doi: <https://doi.org/10.1016/j.micpro.2004.04.002>.
- [82] B.-Q. Huang, G.-Y. Cao, and M. Guo, "Reinforcement Learning Neural Network to the Problem of Autonomous Mobile Robot Obstacle Avoidance," in *2005 International Conference on Machine Learning and Cybernetics*, pp. 85–89, 2005, doi: 10.1109/ICMLC.2005.1526924.
- [83] S. Soumare, A. Ohya, and S. Yuta, "Real-time obstacle avoidance by an autonomous mobile robot using an active vision sensor and a vertically emitted laser slit," in *Intelligent Autonomous Systems*, vol. 7, pp. 301–308, 2002.
- [84] M. Lauer, M. Schönbein, S. Lange, and S. Welker, "3D-objecttracking with a mixed omnidirectional stereo camera system," *Mechatronics*, vol. 21, pp. 390–398, 2011.
- [85] M. Tahmasebi, M. Gohari, and A. Emami, "An Autonomous Pesticide Sprayer Robot with a Color-based Vision System," *International Journal of Robotics and Control Systems*, vol. 2, no. 1, pp. 115–123, Feb. 2022, doi: 10.31763/ijrcs.v2i1.480.
- [86] S. D. Perkasa, P. Megantoro, and H. A. Winarno, "Implementation of a camera sensor pixy 2 cmucam5 to a two wheeled robot to follow colored object," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, pp. 496–501, Nov. 2021, doi: 10.18196/jrc.26128.
- [87] M. I. Rusydi *et al.*, "Autonomous Movement Control of Coaxial Mobile Robot based on Aspect Ratio of Human Face for Public Relation Activity Using Stereo Thermal Camera," *Journal of Robotics and Control (JRC)*, vol. 3, no. 3, pp. 361–373, May 2022, doi: 10.18196/jrc.v3i3.14750.