Advances in Intelligent Systems Research, volume 172
Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019)

Implementation of Fuzzy Logic Control for Navigation System in Mobile Robot Omnidirectional

RENDYANSYAH[1*], Siti NURMAINI[2], Sartika SARI[3], and Dimas Bagus RAMARTA[4]

[1,2,3,4]*Department of Computer Engineering, University of Sriwijaya, Palembang, Indonesia*
**Corresponding author: rendyansyah@ilkom.unsri.ac.id*

ABSTRACT
Robot omnidirectional is a type of wheeled robot that can move efficiently in various directions. Generally, this robot uses Omni wheels. In this study, the mobile robot omnidirectional was developed to move freely in reaching the target point. The robot uses four Omni wheels, where the direction of movement of the robot uses kinematic analysis. The method used for the movement of this robot is fuzzy logic control. The fuzzy logic controller accepts input from five proximity sensors, and the output produces the action of the movement of the robot. Fuzzy logic control in a robot is designed to work on two behaviors, namely avoiding obstacles and approaching the target. The obstacle avoidance behavior is active if the sensors detect an object or wall, while the behavior of approaching the target is based on the error value between the actual position and the reference. The experimental results show that the robot can navigate with good movement in avoiding obstacles and reaching the specified targets.
Keywords: *fuzzy logic control, mobile robot omnidirectional, navigation system*

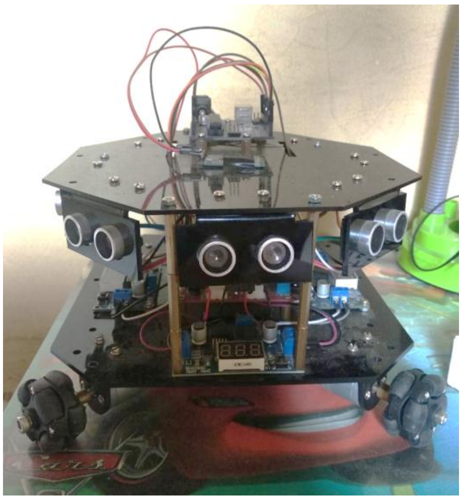
1. INTRODUCTION

A robot is a device that is designed to move both manually and automatically and can help continuous work [1]. Generally, robots use actuators to move, such as electric motors, servo motors, steppers, and others. The drive system in a mobile robot uses electric motors with wheels so they can move and change positions. Most mobile robots use two electric motors and wheels, where the movement of the robots is limited [2],[3]. This limitation of the motion of the robot makes it cannot move in various directions or namely differential steering. From this situation, the robot omnidirectional has been developed that can move freely in multiple directions and is not easy to hit obstacles or walls. The movement of the robot requires a control system embedded in the microcontroller of the robot [4].
In general, the Proportional Integral Derivative (PID) control is more often used in a linear system, because PID is not suitable in non-linear systems. This non-linear system requires additional algorithms to produce a stable system. Several studies have developed a modern control system based on artificial intelligence [5]. One of which is the fuzzy logic controller that is used to overcome the shortcomings of PID because this control system can make decisions based on logic rules that mimic an expert [5],[6]. The fuzzy logic controller makes computation more dynamic compared to the conventional PID. The fuzzy logic controller provides convenience in program design and reduces the use of mathematical functions of complex systems.
The fuzzy logic controller is used to control the movement of the robot and reduce movement errors automatically [7],[8]. Therefore this method can work well on non-linear systems. The robot omnidirectional will have difficulty if it uses conventional controllers. The problem in this work is how the robot omnidirectional can navigate freely in an unstructured environment and can reach the desired target position. The solution we offer is to use a fuzzy logic controller in managing this robot system so that the system can make the right decisions in navigation.
In this study, the fuzzy logic controller method was designed to work in the robot's behaviors, such as avoiding obstacles and approaching the target. The reason for choosing the fuzzy logic method was that it could be programmed either in a microcomputer or microcontroller that had a small memory, and it was low-cost. The movement of the robot in navigation also used kinematic analysis to get the actual position of the robot.

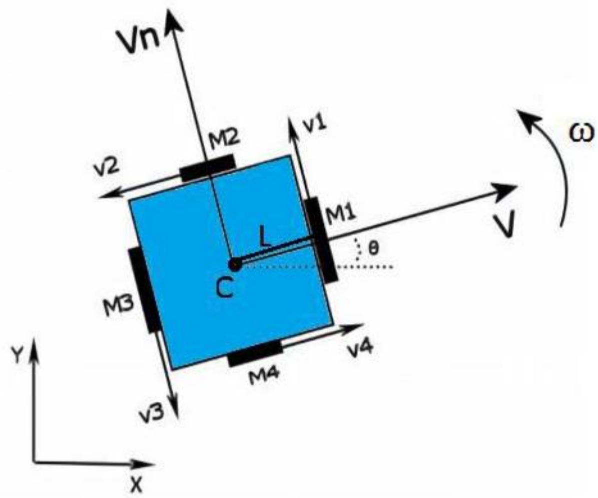
2. METHODS

A. Robot omnidirectional.

In this study, a mobile robot that was used is a type of robot omnidirectional that has four actuators, and each has an Omni wheel. The Omni wheel aimed to move in various directions without rotating the robot's body, where the robot was assumed to be moving in a flat or two-dimensional area. Fig. 1 shows the robot omnidirectional in Cartesian. Based on Fig. 1, parameters such as X and Y were the position of the robot. C was the center of the robot. θ was the angle formed by the x-axis, and ω was the angular velocity. The distance between the center and the wheels showed in L, the linear velocity for each wheel were v1, v2, v3, and v4. The linear velocity centered on C was showed in V and Vn. The speed and angle was influenced by each motor M1, M2, M3, and M4. The DC motors used were all types of metal Gearmotor.



(a)



(b)

Figure 1. (a) Robot omnidirectional, and (b) illustration of the direction of movement [9].

The basic formulation for kinematic in the robot is shown in Eq. 1. The TNH(q) parameter is a non-holonomic transformation of the mobile robot, and u(t) is input used to estimate the position and velocity of the robot in cartesian space.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = T_{NH}(q).u(t) \tag{1}$$

In Fig. 1, there is linear velocity for each of v_1, v_2, v_3, and v_4 are:

$$\begin{aligned} v_1(t) &= V_n(t) + L.\omega(t) \\ v_2(t) &= -V(t) + L.\omega(t) \\ v_3(t) &= -V_n(t) + L.\omega(t) \\ v_4(t) &= V(t) + L.\omega(t) \end{aligned} \tag{2}$$

Velocity v_1 to v_4, if added together, will produce an angle is given in Eq. 3.

$$\dot{\theta}(t) = \omega(t) = \frac{1}{4L}\left(v_1(t) + v_2(t) + v_3(t) + v_4(t)\right) \tag{3}$$

From Eq. 2, also obtained formulation V and V_n as in Eq. 4.

$$\begin{aligned} V_n &= \frac{1}{2}\left(v_1(t) - v_3(t)\right) \\ V &= \frac{1}{2}\left(v_4(t) - v_2(t)\right) \end{aligned} \tag{4}$$

Based on the formulation of non-holonomic constraints in the robot omnidirectional, the actual position obtained as in Eq. 5 and 6.

$$\begin{aligned} \dot{x}(t) &= V(t)\cos\theta(t) - V_n(t)\sin\theta(t) \\ \dot{y}(t) &= V(t)\sin\theta(t) + V_n(t)\cos\theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{aligned} \tag{5}$$

$$\begin{aligned} \dot{x}(t) &= V(t)\cos\theta(t) - V_n(t)\cos(90-\theta)(t) \\ \dot{y}(t) &= V(t)\sin(90-\theta)(t) + V_n(t)\sin\theta(t) \\ \dot{\theta}(t) &= \omega(t) \end{aligned} \tag{6}$$

In Eq. 6, combined with Eq. 4, results in the formulation of positions in the form of a matrix as in Eq. 7 and 8.

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\left(v_4(t) - v_2(t)\right)\cos\theta(t) - \frac{1}{2}\left(v_1(t) - v_3(t)\right)\cos(90-\theta)(t) \\ \frac{1}{2}\left(v_4(t) - v_2(t)\right)\sin\theta(t) + \frac{1}{2}\left(v_1(t) - v_3(t)\right)\sin(90-\theta)(t) \\ \frac{1}{4L}\left(v_1(t) + v_2(t) + v_3(t) + v_4(t)\right) \end{bmatrix} \tag{7}$$

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \end{bmatrix} = \begin{bmatrix} -\frac{1}{2}\cos(90-\theta)(t) & -\frac{1}{2}\cos\theta(t) & \frac{1}{2}\cos(90-\theta)(t) & \frac{1}{2}\cos\theta(t) \\ \frac{1}{2}\sin(90-\theta)(t) & -\frac{1}{2}\sin\theta(t) & -\frac{1}{2}\sin(90-\theta)(t) & \frac{1}{2}\sin\theta(t) \\ \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} & \frac{1}{4L} \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{bmatrix} \tag{8}$$

From these equations, the actual positions are the output, and the velocity v_1, v_2, v_3, and v_4 are the input.

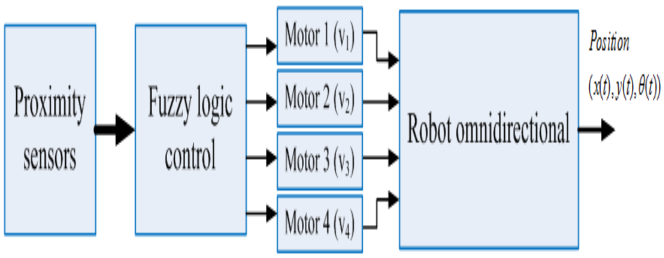


Figure 2. The fuzzy logic control for movements of the robot in avoiding obstacles.

B. Fuzzy logic controller.

The fuzzy logic controller is an artificial intelligence that mimics the workings of an expert based on experience in decision making. The fuzzy logic control consists of fuzzification, inference and logic rules, and defuzzification [7],[8],[10]. In this study, fuzzy logic control was implanted into the robot's behavior, such as avoiding obstacles and approaching the target. The block diagrams, respectively, are shown in Fig. 2 and Fig. 3. The fuzzy logic method used was the Sugeno model because the computation was simple and suitable to be embedded in a microcontroller-based system.

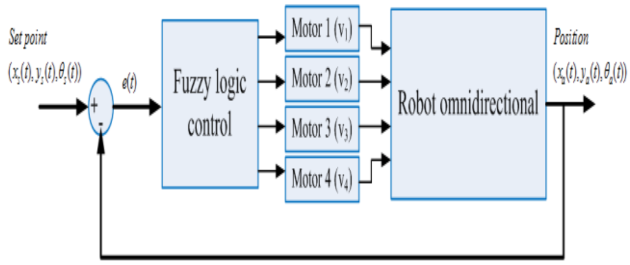


Figure 3. The fuzzy logic control for the robot moves to approach the target.

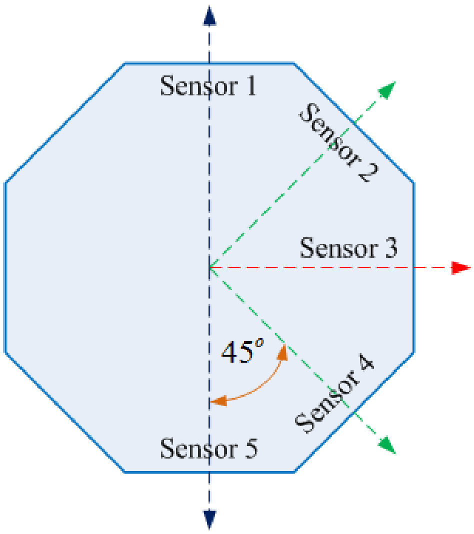
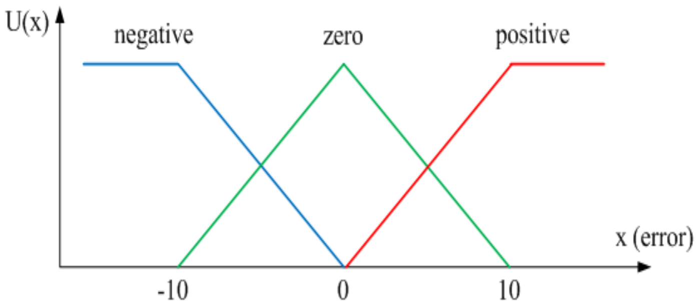
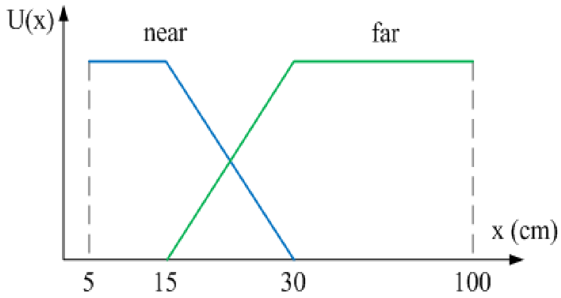


Figure 4. The positioning of proximity sensors.

Fig. 2 is the block diagram for obstacle avoidance behavior, where the input came from proximity sensors, and the output produced navigation of robotic to avoid obstacles. The positioning of the proximity sensors with an angle of 45° is shown in Fig. 4. Fig. 3 is the block diagram for the behavior of approaching the target, where the input was the difference between the reference position and the actual position, and the output was the actual position. The actual position was obtained from the kinematic formulation of the robot because in Eq. 8, the input was the speed of each wheel and the output was the actual position. These behaviors were active alternately depending on the stimuli received by the robot. The behavior of avoiding obstacles had a higher priority than the behavior of approaching the target. The design of fuzzy logic to these behaviors is as follows:



(a)



(b)

Figure 5. The membership function, (a) for error values, and (b) proximity sensors.

1. Fuzzification: In this study, there are two fuzzifications, namely, to approach the target and avoid obstacles. The fuzzification for approaching the target behavior is the error value between the actual position and the reference position. Fuzzification for avoiding obstacles is the value of proximity sensors. Fig. 5 shows the membership function for error values and the membership function for the proximity sensors.

Table 1. The logic rules for the robot approaching the target.

		error x		
		negative	zero	positive
error y	negative	v1 = positive v2 = positive	v1 = positive v2 = zero	v1 = positive v2 = negative

		v3 = negative v4 = negative	v3 = negative v4 = zero	v3 = negative v4 = positive
	zero	v1 = zero v2 = positive v3 = zero v4 = negative	v1 = zero v2 = zero v3 = zero v4 = zero	v1 = zero v2 = negative v3 = zero v4 = positive
	positive	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero	v1 = negative v2 = negative v3 = positive v4 = positive

Table 2. The logic rules for the robot avoiding obstacles.

		Sensors 4 and 5			
		near, near	near, far	far, near	far, far
Sensors 1, 2, and 3	near, near, near	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero
	near, near, far	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero

	near, far, near	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero
	near, far, far	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = zero v2 = negative v3 = zero v4 = positive	v1 = zero v2 = negative v3 = zero v4 = positive
	far, near, near	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero
	far, near, far	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = positive v3 = positive v4 = negative	v1 = negative v2 = zero v3 = positive v4 = zero
	far, far, near	v1 = positive v2 = zero v3 = negative v4 = zero	v1 = positive v2 = zero v3 = negative v4 = zero	v1 = positive v2 = zero v3 = negative v4 = zero	v1 = negative v2 = zero v3 = positive v4 = zero
	far, far, far	v1 = positive	v1 = positive	v1 = zero	v1 = zero

		v2 = zero v3 = negative v4 = zero	v2 = zero v3 = negative v4 = zero	v2 = negative v3 = zero v4 = positive	v2 = negative v3 = zero v4 = positive

2. Inference and logic rules: In this study, the logic rules are determined based on our experience while learning the stages of fuzzy logic. The logic rules that we have designed for the robot approaching the target and avoiding obstacles can be seen in Table 1 and Table 2, respectively. The fuzzy output of the logic rules must be synchronous with the membership function on the crisp output of the system, as shown in Fig. 6. As for the minus sign, it is the direction of movement that is opposite to the illustration of normal movement (see Fig. 1(b)). The decision-making to produce fuzzy output using the max-min mechanism [7],[10]. The output of this membership function applies to these two behaviors.
3. Defuzzification: The defuzzification is used to change the value of fuzzy output to crisp output. The Sugeno model represents the membership function of fuzzy output as a singleton is shown in Fig. 6. In this study, the defuzzification process uses the Centre of Area method [6],[7].

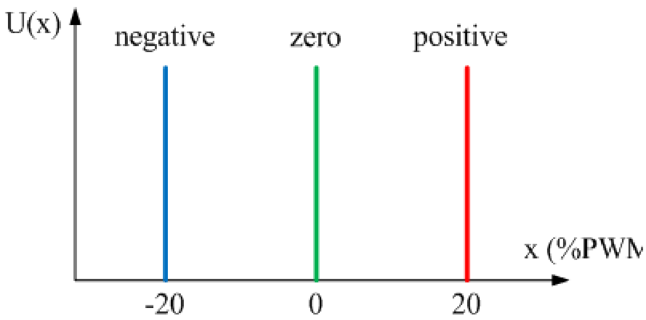
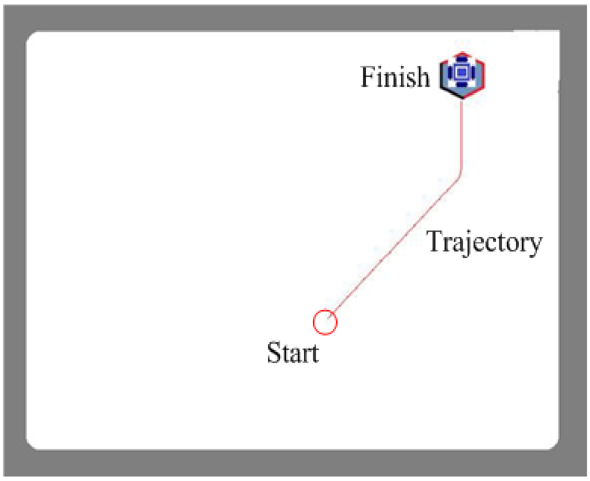


Figure 6. The membership function for crisp output of the system.

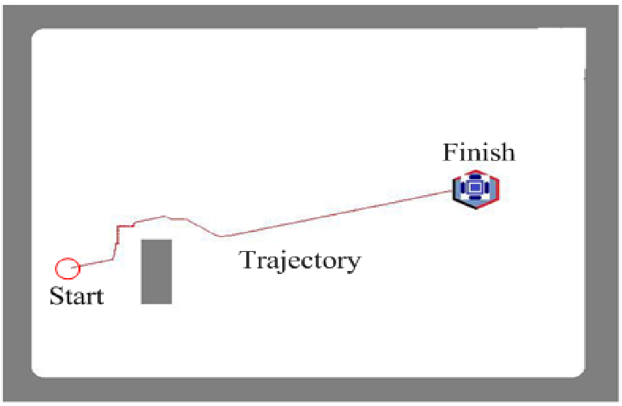
EXPERIMENTAL RESULTS

The experiment was conducted to determine the response of the movement of the robot in the environment. The robot was tested in a flat area with a length of 300 cm and a width of 250 cm, and there were various obstacles. In this study, the robot omnidirectional was tested in six environmental conditions. In the first experiment, the robot was placed in an environment that had no obstacle. In this condition, the robot moved directly to the target as shown in Fig. 7(a). The initial position was $x_s(t)$ = 160 cm and $y_s(t)$ = 110 cm, and the target was final position $x_r(t)$ = 230 cm and $y_r(t)$ = 230 cm. The initial and final positions were freely determined. In Fig. 7(a) it is seen that the robot navigated a linear trajectory because the kinematics of the robot system accepted input motor speed, and the output produced the actual position. In the second experiment, there was one obstacle condition that was placed in front of the robot. In the second experiment, this condition changed at the initial position of the robot (35, 75), and the final position (230, 135). The experimental result on the robot for one obstacle is shown in Fig. 7(b). Based on Fig. 7(b), it can be seen the navigation of the movement of the robot when detecting an obstacle, and the robot could maneuver, and after that, the robot went directly to the target. The robot activated the obstacle avoidance behavior as the priority.

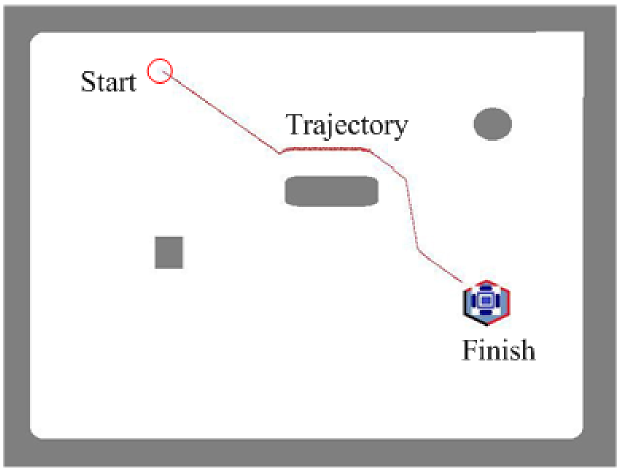
In the third experiment, there were three obstacles placed in front of the robot. In this experiment, the initial position (80, 230) and the final position (230, 110), were determined. The experimental result in this environmental condition is shown in Fig. 7(c). When the robot was activated, the robot moved to approach the target and detected an obstacle. The robot activated obstacle avoidance behavior, and after that, the robot moved to approach the target. In the fourth experiment, there were four obstacles, where the robot can navigate to avoid obstacles and reach the target as shown in Fig. 7(d). This fourth experiment was a determined initial position (110, 230), and the final position (115, 55). Based on Fig. 7(d), the robot avoided two obstacles, namely the form of a rectangle and triangle, and the robot could also navigate on this condition.



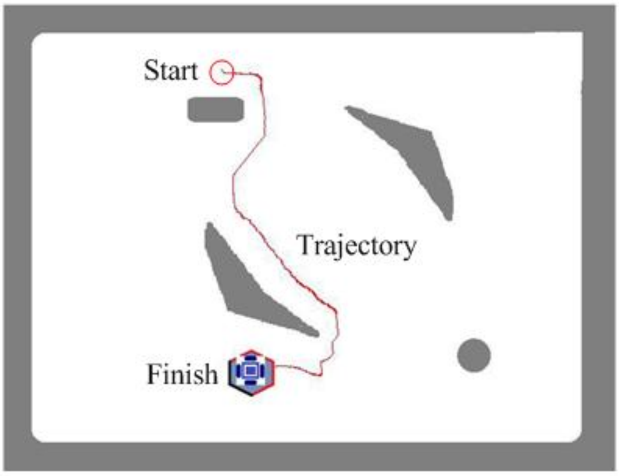
(a)



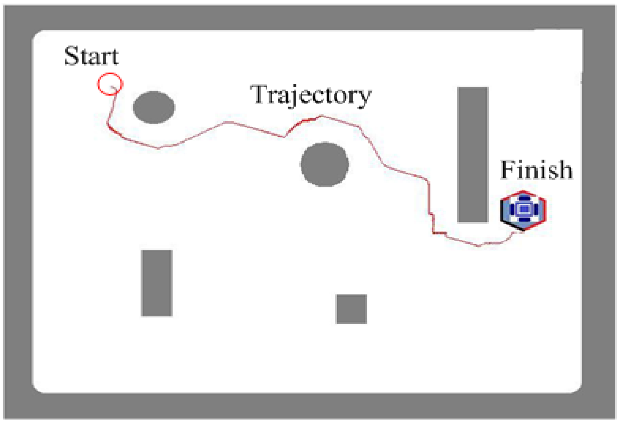
(b)



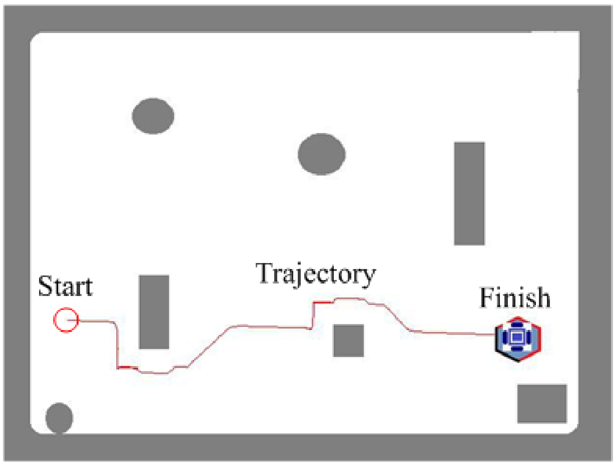
(c)



(d)



(e)



(f)

Figure 7. The trajectory on the robot omnidirectional in various environmental conditions. (a) without obstacle, (b) one obstacle, (c) three obstacles, (d) four obstacles, (e) five obstacles and (f) seven obstacles.

Next, in the fifth experiment, there were five obstacles. The robot and the target determined at the initial and final positions, were respectively (70, 225) and (250, 130). The experimental result on the robot in this environment is shown in Fig. 7(e). The robot could navigate in avoiding two obstacles in the form of a circle and rectangular obstacle, after which the robot could reach the target. In the last experiment, there were many obstacles. This experiment was similar to the previous experiment, and there were additional obstacles in the corner of the environment. The initial and final positions were (35, 70) and (250, 65), respectively. The trajectory of the robot on the seventh experiment is shown in Fig. 7(f). In this condition, the robot was also able to navigate, avoiding two rectangle obstacles, and then the robot moved to the target. In experiments that had been done, the robot omnidirectional could work well when avoiding obstacles and moving approaching the targets with various environmental conditions. This experiment had also performed a comparison with the movement of the robot without using the fuzzy logic controller. The results of the movement of the robot using fuzzy logic and produced smaller errors in reaching the target position compared to without fuzzy logic can be seen in Table 3. This work could be achieved because fuzzy logic control could make decisions when controlling the system with various inputs and conditions.

Table 3. The comparison of error on the movement of the robot to reach the target by using fuzzy logic and without fuzzy logic.

No	Experimental with several environment conditions	Using the fuzzy logic controller		Without the fuzzy logic controller	
		error x	error y	error x	error y
1	without obstacle	0.9	-1	4.2	-4.6
2	one obstacle	1.9	1	4.8	4.5
3	tree obstacles	2	0.9	4.4	4.5
4	four obstacles	1.9	2	-4.5	-4.5
5	five obstacles	-1.6	-1	-5	-4.5
6	seven obstacle	2	0.9	4.3	4.5

CONCLUSION

Robot omnidirectional has succeeded in carrying out tasks such as avoiding obstacles and reaching targets. This robot has five proximity sensors and four electric motors. The positioning of the proximity sensors with angle is 45°, and the motors of 90°. In this study, robot's experiments have been carried out in various environmental conditions, ranging from without obstacle to many obstacles. The control system that is implanted in this robot omnidirectional is fuzzy logic control, which is designed to work on two behaviors, namely avoiding obstacles and approaching the target. These experiments show that robot omnidirectional can navigate in avoiding various obstacles and reaching the specified targets. It can be concluded that fuzzy logic control can make decisions in controlling systems based on reasoning.

ACKNOWLEDGMENT

This research was carried out with support from the Institution of Research and Community Services (LPPM), University of Sriwijaya.

REFERENCES

[1] N. Arifin, “Robot Arm Pemindah Dan Penyeleksi Ban Berdasarkan Kode Pemilah Warna Berbasis Arduino Uno Dengan Fuzzy Logic Control Pada Pt.Multi Strada Arah Sarana.Tbk,” *J. Inform. SIMANTIK*, vol. 2, pp. 25–31, 2017.

[2] D. Ahad, M. Mynuddin, and D. Rashid, “Analyzing Behavior of Differential Drive Mobile Robot by Using Statistical Tools,” *J. Electr. Eng. Electron. Technol.*, vol. 05, no. 03, pp. 1–4, 2016.

[3] B. Deepak and D. R. Parhi, “Path Generation of a Differential Mobile Robot Using Fuzzy Inference system,” *5th Int. Conf. Inf. Syst. Des. Intell. Appl.*, no. 2, pp. 1–6, 2018.

[4] M. Rahman and H. Aprilianto, “Penerapan Metode Fuzzy Pada Robot Beroda Menggunakan Omni-Directional Wheels,” *Jutisi*, vol. 5, no. 2, pp. 1075–1082, 2017.

[5] D. K. Tiep, K. Lee, D. Y. Im, B. Kwak, and Y. J. Ryoo, “Design of Fuzzy-PID Controller for Path Tracking of Mobile Robot with Differential Drive,” *Int. J. Fuzzy Log. Intell. Syst.*, vol. 18, no. 3, pp. 220–228, 2018.

[6] Fahmizal, B. B. Murti, D. B. Pratama, and A. Mayub, “Kendali Logika Fuzzy pada Car Like Mobile Robot (CLMR) Penjejak Garis,” *ELKOMIKA J. Tek. Energi Elektr. Tek. Telekomun. Tek. Elektron.*, vol. 6, no. 3, pp. 451–464, 2018.

[7] M. Rivai, Rendyansyah, and D. Purwanto, “Implementation of Fuzzy Logic Control in Robot Arm for Searching Location of Gas Leak,” in *2015 International Seminar on Intelligent Technology and Its Applications, ISITIA 2015 - Proceeding*, 2015, pp. 69–74.

[8] H. Omrane, M. S. Masmoudi, and M. Masmoudi, “Fuzzy Logic Based Control for Autonomous Mobile Robot Navigation,” *Comput. Intell. Neurosci.*, pp. 1–10, 2016.

[9] Conceição, A. Scolari, and A. P. Moreira, “Model Identification of a Four Wheeled Omni-Directional Mobile Robot,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[10] Fahmizal and C. H. Kuo, “Development of a Fuzzy Logic Wall Following Controller for Steering Mobile Robots,” *Int. Conf. Fuzzy Theory Its Appl.*, pp. 7–12, 2013.