Journal of Applied Intelligent System (e-ISSN : 2502-9401 | p-ISSN : 2503-0493)
Vol. 4 No. 1, 2019, pp. 9 – 21

Implementation of Fuzzy Logic Controller for Wall Following and Obstacle Avoiding Robot

Aryuanto Soetedjo*[1], **M. Ibrahim Ashari**[2]
[1,2]Department of Electrical Engineering, National Insitute of Technology (ITN) Malang, Indonesia
E-mail : aryuanto@gmail.com[1]*
**Corresponding author*

Cosmas Eric Septian[3]
[3]Department of Electrical Engineering, National Insitute of Technology (ITN) Malang, Indonesia

Abstract - This paper presents the development of wall following and obstacle avoiding robot using a Fuzzy Logic Controller. The ultrasonic sensors are employed to measure the distances between robot and the wall, and between the robot and the obstacle. A low cost Raspberry Pi camera is employed to measure the left/right distance between the robot and the obstacle. The Fuzzy Logic Controller is employed to steer the mobile robot to follow the wall and avoid the obstacle according to the multi sensor inputs. The outputs of Fuzzy Logic Controller are the speeds of left motor and right motor. The experimental results show that the developed mobile robot could be controlled properly to follow the different wall positions and avoid the different obstacle positions with the high successful rate of 83.33%.

Keywords - Mobile robot, Wall following, Obstacle avoiding, Sensor fusion, Fuzzy logic

1. INTRODUCTION

Wall following and obstacle avoiding are the important tasks for the autonomous mobile robot. Many algorithms and sensors have been proposed for controlling the robot to follow the wall [1-6]. The Lyapunov method was employed to steer the mobile robot which is driven by the differential wheels [1]. In the method, the error distance and the error orientation of the robot from the wall were selected as the state variables of the model. To overcome the problems of modeling error and sensing uncertainty, the Fuzzy Logic Controllers (FLCs) were employed [2-6].

In [2], the FLC was employed to steer the wheelchair along the wall using the ultrasonic sensors. Four ultrasonic sensors located at four corners of the wheelchair were used to detect the distance of objects or wall in four areas, i.e. front-left, rear-left, front-right, and rear-right. The information of distance was used by the FLC to adjust the angle of wheelchair while moving with a constant speed. In [3], five ultrasonic sensors were employed to detect the distance of mobile robot from the wall in front, left, and right direction. Then the error between the desired distance and the actual distance was calculated. The error and the derivation of error were used by the FLC for adjusting the motor's speed of mobile robot. To handle the different situations of the wall, i.e. the flat wall, the sharp corner, and the limit corner, the multi sensors system was developed in [4]. The multi sensors consisted of IR sensors and sonar sensors. The type-2 FLC was adopted to steer the mobile robot in the right wall following method.

Three laser range finders were employed as the input of fuzzy controller for multi-type-wall following mobile robot [5]. The FLC was designed for each type of wall. The behavior-based FLC was employed to switch the FLC according to the detected wall's type. In [6], the FLC was employed to adjust the orientation of hexapod robot by controlling the swing angle of middle legs. The FLC got the distance information between the robot and the wall from three infrared sensors.

The mobile robots for avoiding obstacles were addressed in [7-11]. A mobile robot was operated in the warehouse for transferring the objects [7]. Twenty four ultrasonic sensors were installed around the robot to detect the obstacles. The sensors were classified into several groups for easy processing by a rule-based expert system. The wall following approach was adopted in [8] to control the robot avoiding the obstacles. The method divided the distance between the robot and wall/obstacle into three regions, i.e. inner region, desired region, and outer region. The FLC was adopted to maintain the robot in the desired region.

In [9], the FLC with eight inputs and five outputs was employed to control the movement of a robot in the environment with the cluttered obstacle objects. They developed 256 fuzzy rules to process the information from 8 infrared sensors. While the output of FLC is the movement of a robot in five directions, i.e. forward, turn right 90^0, turn right 45^0, turn left 90^0, and turn left 45^0.

In this paper, we develop the wall following and obstacle avoiding mobile robot using the sensor fusion consisted of the ultrasonic sensors and a camera. The wall following algorithm employs the ultrasonic sensors to detect the wall. The obstacle avoiding algorithm employs the ultrasonic sensor and the camera to detect the obstacle. The FLC is adopted in each algorithm. The algorithms are implemented on a Raspberry Pi module. The main contributions of our work are in the combination of low cost sensors and the FLC implementation.

The rest of paper is organized as follows. Section 2 presents the configuration of mobile robot. Section 3 presents the FLC design. Section 4 describes the experimental results. The conclusion is covered in Section 5.

2. CONFIGURATION OF MOBILE ROBOT

The configuration of mobile robot is shown in Fig. 1. There are two controllers, i.e. a Raspberry Pi as the master controller and an ATmega162 microcontroller as the slave controller. The slave controller is employed as an interface to the ultrasonic sensors and the motors. The master controller is employed as an interface to the camera module, and to perform the FLC technique.

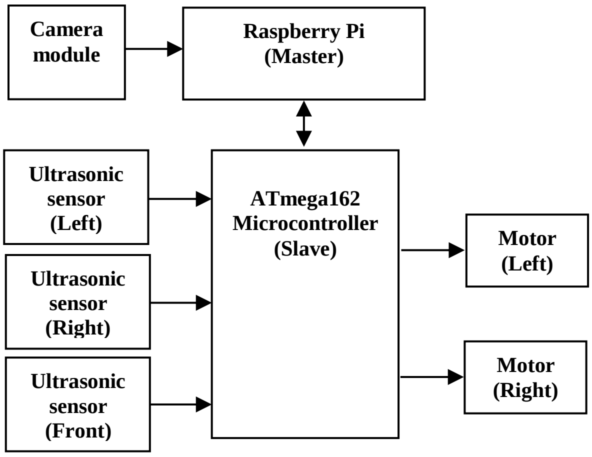

Camera module
Raspberry Pi (Master)
Ultrasonic sensor (Left)
Ultrasonic sensor (Right)
Ultrasonic sensor (Front)
ATmega162 Microcontroller (Slave)
Motor (Left)
Motor (Right)

Figure 1. Hardware configuration of mobile robot.

The mobile robot has two wheels and a free-wheel (caster wheel). The wheels are driven differentially by two DC motors. Three ultrasonic sensors are installed in the left, right, and front of the mobile robot. The left and right sensors are used to detect the wall, while the front sensor is used to detect the obstacle. The robot is equipped with a switch to select the wall following mode, i.e. right wall following or left wall following.

2.1. Ultrasonic Sensor

The Ping ultrasonic sensor [12] as shown in Fig. 2 is used to measure the distance between the robot and the objects (wall or obstacle). The module consists of a ultrasonic speaker and ultrasonic microphone. To measure the distance, at first the speaker transmits a burst signal of 40 kHz, and then measure the time when the signal returns back, i.e. received by the ultrasonic microphone. The module is able to measure the distance of object in the range of 3 cm to 3.3 m.

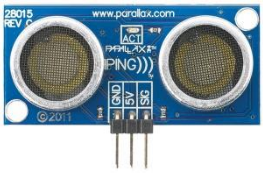


Figure 2. Ping ultrasonic sensor [12].

2.2. Camera Module

The camera module used in the experiment is a Raspberry Pi camera as shown in Fig. 3 [13]. The module uses the OmniVision OV5647 camera sensor with the resolution of 5 Megapixels. The camera module is connected to the Raspberry Pi using the CSI (Camera Serial Interface). To capture the image and process the image processing task, the OpenCV computer vision library [14] is employed. It provides an easy way to implement the image processing algorithms in the Raspberry Pi.

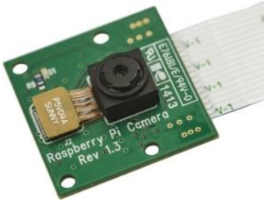


Figure 3. Raspberry Pi camera module [13].

2.3. Obstacle Detection

Since only the single camera is installed in the robot, it could not measure the distance effectively. Therefore, we employ the ultrasonic sensor to measure the distance of the obstacle in the y-direction (see Fig. 4). The camera is used to detect the position of obstacle in the x-direction.

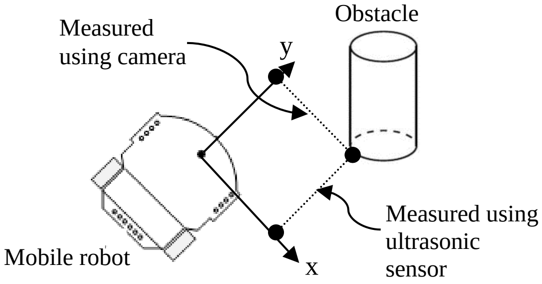


Figure 4. Obstacle distance measurement.

In this work, the obstacle is detected by its color, i.e. the red color. Thus, a simple color thresholding is employed to detect the red object using the camera. Both the distances in x-direction and y-direction are then fed to the FLC for avoiding the obstacle as described in the next section.

2.4. Motor Driver

The mobile robot is driven by two DC motors controlled separately. The motor's speed is controlled using PWM (Pulse Width Modulation) technique. To allow the motor moves in CW (Clockwise) and CCW (Counter clockwise) directions, the H-bridge driver is employed. It is implemented using IC L298, where the direction and PWM's duty cycle are controlled by the ATmega162 microcontroller.

Since the left and right motors are driven separately, the direction of robot movement is controlled by the speed of each motor. If both motors have the same speed, then robot will move forward. If the speed of left motor is lower than the right motor, then the robot will turn to the left, vice versa. The speed of each motor is controlled by the FLC controller as described in the next section.

2.5. Communication Between Master Controller and Slave Controller

As described previously, the sensors (except the camera) and actuators are handled by the slave controller (ATmega162 microcontroller). While the main control is handled by the master controller (Raspberry Pi). Both controllers communicate via serial communication using a simple protocol as described below.

The slave controller sends the distance information from the ultrasonic sensors and the wall following mode to the master controller with the communication data format as given in Table 1. The master controller sends the motor command to the slave controller with the communication data format as given in Table 2.

Table 1. Master to slave data format

Byte number	Data	Remark
1	Left distance	Distance measured by left sensor (cm)
2	;	Separator
3	Front distance	Distance measured by front sensor (cm)
4	;	Separator
5	Right distance	Distance measured by right sensor (cm)
6	;	Separator
7	Mode	Right/left wall following
8	Enter (\n)	End of data

Table 2. Slave to master data format

Byte number	Data	Remark
1	Motor direction	Forward/backward
2	;	Separator
3	Right motor speed	PWM value of right motor
4	;	Separator
5	Left motor speed	PWM value of left motor
6	Enter (\n)	End of data

3. FLC DESIGN

Block diagram of developed FLC is shown in Fig. 5. As shown in the figure, each task (wall following or obstacle avoiding) has the own FLC. The inputs of FLC-wall following are the error of wall distance and the difference of error. The error of wall distance is calculated from the subtraction of the desired wall distance and the actual wall distance. The outputs are the speed (PWM value) of right motor and left motor, where the value of 0 represents the duty cycle of 0 %, while the value of 255 represents the duty cycle of 100%.

The inputs of FLC-obstacle avoiding are the obstacle distance in x-direction and y-direction as shown in Fig. 4. The outputs are the speed (PWM value) of right motor and left motor. It is noted here that the algorithm for wall following and obstacle avoiding does not run simultaneously, i.e. when there is no obstacle, then the robot perform the wall following task, otherwise it performs the obstacle avoiding task. Therefore, even though the outputs of two FLCs are the same, they do not cause a conflict.

Both FLCs use the same Sugeno model. The detail design of each FLC is described in the following section.

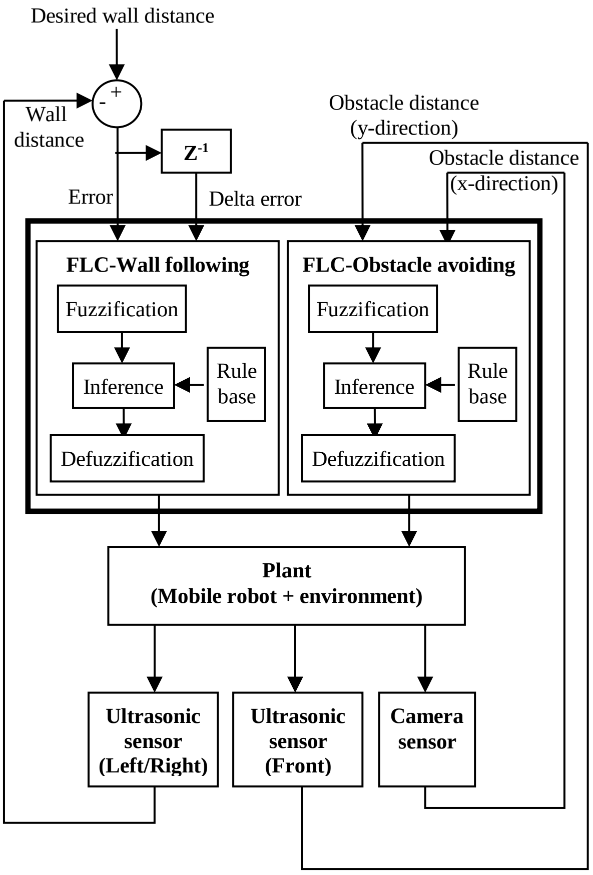


Figure 5. Block diagram of FLC.

3.1. FLC-Wall Following

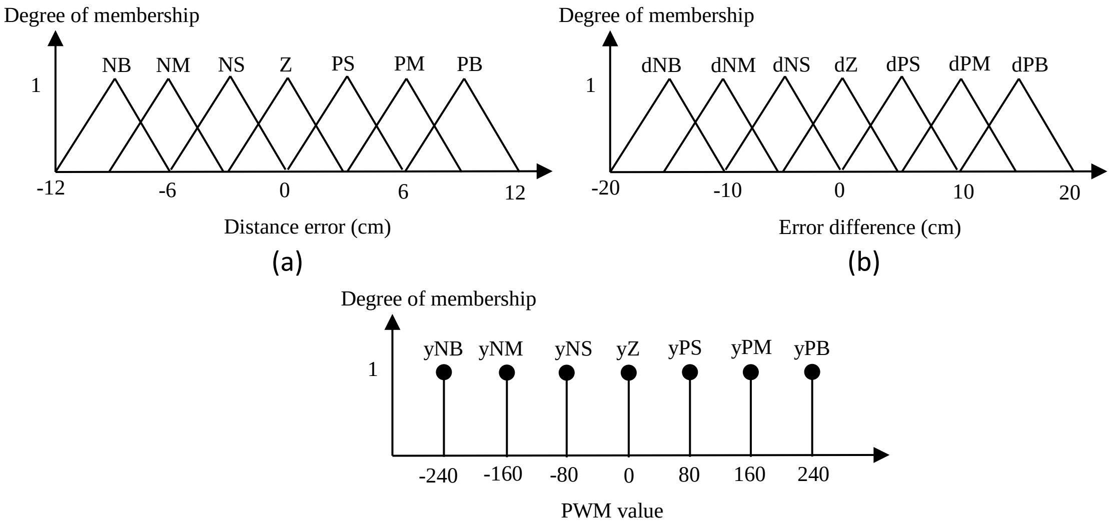
The objective of wall following task is to move the robot while maintaining the fixed distance from the wall. The developed FLC technique is based on the behavior of PD

(Proportional and Derivative) controller, where the proportional component is used to increase the rising time, while the derivative component is used to reduce the overshoot. To accomplish this behavior, the error and the error difference is employed as the input of FLC.

The error has seven linguistic values, i.e. NB (Negative Big), NM (Negative Medium), NS (Negative Small), Z (Zero), PS (Positive Small), PM (Positive Medium), and PB (Positive Big), where the membership functions are shown in Fig. 6(a).

The error difference has seven linguistic values, i.e. dNB (Negative Big), dNM (Negative Medium), dNS (Negative Small), dZ (Zero), dPS (Positive Small), dPM (Positive Medium), and dPB (Positive Big), where the membership functions are shown in Fig. 6(b).

Since the Sugeno model is adopted, the membership function of PWM value is the fuzzy singleton. The PWM value has seven linguistic values, i.e. yNB (Negative Big), yNM (Negative Medium), yNS (Negative Small), yZ (Zero), yPS (Positive Small), yPM (Positive Medium), and yPB (Positive Big) as shown in Fig. 6(c).



(a)

(b)

(c)

Figure 6. (a) Membership function of error; (b) Membership function of error difference; c) Membership function of PWM value in the FLC-wall following.

The rule base of FLC-wall following is given in Table 3. In this method, only one PWM value is generated for driving two motors to minimize the wall distance error as discussed in the following. Let the PWM value generated by FLC is *PWM_val* (a crisp value). If the right wall following mode is selected, then the motors' speed are controlled according to (1). If the left wall following mode is selected, then the motors' speed are controlled according to (2). *PWM_out_right* is the PWM output of right motor, *PWM_out_left* is the PWM output of left motor, *PWM_ofs* is the offset value of PWM.

Table 3. Rule base of FLC-wall following

error *delta error*	**NB**	**NM**	**NS**	**Z**	**PS**	**PM**	**PB**
dNB	*yNB*	*yNB*	*yNB*	*yNS*	*yNM*	*yNM*	*yZ*
dNM	*yNB*	*yNB*	*yNM*	*yNM*	*yNM*	*yZ*	*yPM*
dNS	*yNB*	*yNM*	*yNM*	*yNS*	*yZ*	*yPM*	*yPM*
dZ	*yNM*	*yNM*	*yNS*	*yZ*	*yPS*	*yPM*	*yPM*
dPS	*yNM*	*yNM*	*yZ*	*yPS*	*yPM*	*yPM*	*yPB*
dPM	*yNM*	*yZ*	*yPM*	*yPM*	*yPM*	*yPB*	*yPB*
dPB	*yZ*	*yPM*	*yPM*	*yPM*	*yPB*	*yPB*	*yPB*

$$\begin{aligned} PWM\_out\_right &= PWM\_ofs + PWM\_val \\ PWM\_out\_left &= PWM\_ofs - PWM\_val \end{aligned} \qquad (1)$$

$$\begin{aligned} PWM\_out\_right &= PWM\_ofs - PWM\_val \\ PWM\_out\_left &= PWM\_ofs + PWM\_val \end{aligned} \qquad (2)$$

3.2. FLC- Obstacle Avoiding

In the FLC-obstacle avoiding, the robot detects the obstacle by measuring the distances in x-direction and y-direction. The strategy to avoid the obstacle is by controlling the right and left motors to move away from the obstacle. To simplify the discussion, the distances of obstacle in the x-direction and y-direction are called as Xdist and Ydist respectively.

The linguistic values of Xdist are LF (Left), CT (Center), and RT (Right), where the membership functions are shown in Fig. 7(a). The linguistic values of Ydist are is VC (Very close), CL (Close), and FR (Far). where the membership functions are shown in Fig. 7(b). The linguistic values of PWM output are SL (Small), MD (Medium), and BG (Big), where the membership functions are shown in Fig. 7(c).

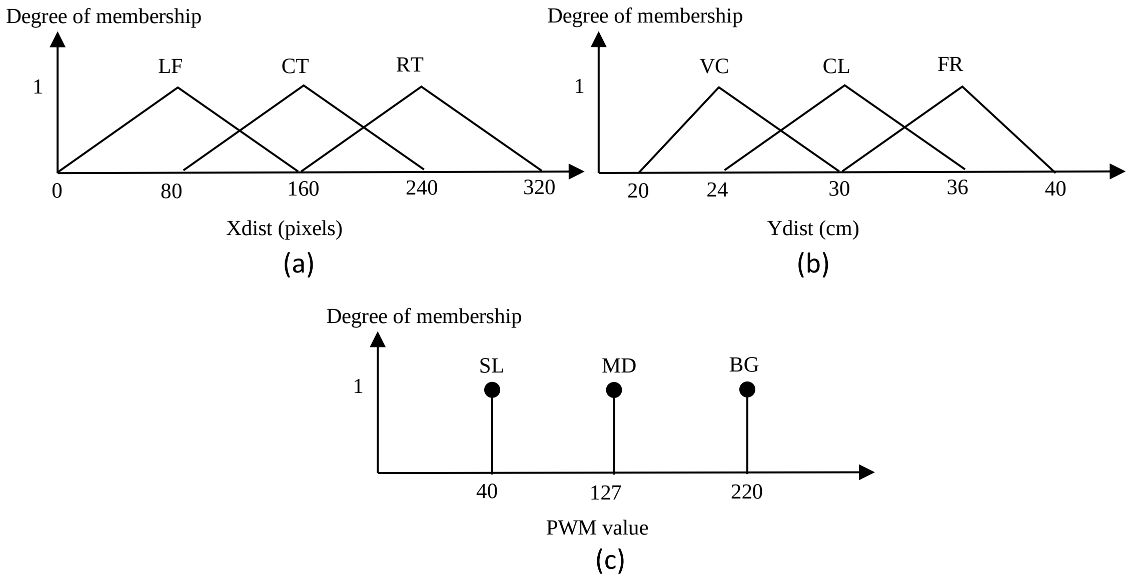


Figure 7. (a) Membership function of distance in x-direction; (b) Membership function of distance in y-direction; (c) Membership function of PWM value in the FLC-obstacle avoiding.

In the experiment, the avoiding direction follows the wall following mode, i.e. when the right wall following mode is selected, then the obstacle avoiding algorithm will steer the robot to turn right for avoiding the obstacle. The rule bases of FLC-obstacle avoiding for the left mode and the right mode are shown in Table 4 and Table 5 respectively. As shown in table, there are two outputs for each rule, i.e. PWM values of the left motor (MLeft) and the right motor (MRight). The rule is developed based-on a common practice in steering the differential motor. For example a rule in Table 4, if the Xdist is center (CT) and the Ydist is very close (VC), then the left motor will be moved fast (PWM value is big (BG) while the right motor will be moved slowly. This rule will steer the robot by turning left for avoiding the obstacle.

Table 4. Rule Base of FLC-Obstacle Avoiding (Left mode)

Xdist / Ydist	LF	CT	RT

VC	MLeft=BG
	MRight=SL
		MLeft=BG
		MRight=SL
			MLeft=MD
			MRight=SL

CL	MLeft=BG
	MRight=SL
		MLeft=MD
		MRight=SL
			MLeft=MD
			MRight=SL

FR	MLeft=BG
	MRight=MD
		MLeft=MD
		MRight=SL
			MLeft=MD
			MRight=SL

MLeft=Left motor; MRight=Right motor			

Table 5. Rule Base of FLC-Obstacle Avoiding (Right mode)

Xdist / Ydist	LF	CT	RT

VC	MLeft= SL
	MRight= BG
		MLeft= SL
		MRight= BG
			MLeft= SL
			MRight=MD

CL	MLeft= SL
	MRight= BG
		MLeft= SL
		MRight=MD
			MLeft= SL
			MRight=MD

FR	MLeft= MD
	MRight= BG
		MLeft= SL
		MRight=MD
			MLeft=SL
			MRight=MD

MLeft=Left motor; MRight=Right motor			

4. EXPERIMENTAL RESULTS

Several experiments are conducted to test the mobile robot. In the experiments, the robot is tested to navigate on the different scenarios, i.e.: a) Following a straight wall; b) Turning on 90^0 wall corner; c) Turning on 180^0 wall corner; d) Avoiding an obstacle; e) Following the walls and avoiding the obstacles. In each scenario, the robot runs ten times, and the successful rate of navigation is computed. The navigation is defined as success when the robot does not touch the wall during the wall following or does not touch the obstacle during the obstacle avoiding.

4.1. Following a Straight Wall

The scenario of following a straight wall is shown in Fig. 8. When robot moves from a start angle of 0^0 as shown in Fig. 8, the robot is able to follow the wall successfully in ten trials as given in Table 6.

To observe the effect of start angle, the experiments on varying start angle (20^0 to 90^0) are conducted. The navigation results are given in Table 7. From the table it is obtained that for the start angle of 60^0 and 80^0, there is one navigation failure from five trials, while for the start angle of 70^0, there are two navigation failures from five trials. The navigation failure means that the robot touches the wall, but it goes back to the desired path as shown in Fig. 9. From the observations, the failure may be caused by instability of the robot mechanic and the sensor systems that yields an improper control signal (too big PWM value) from FLC.

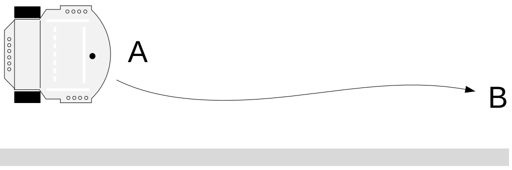


Figure 8. Scenario of following a straight wall.

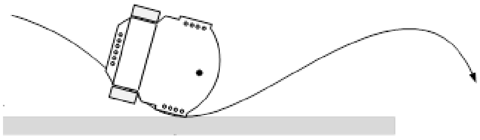


Figure 9. Navigation failure when following a straight wall.

Table 6. Results of following a straight wall

No. of Trial	1	2	3	4	5	6	7	8	9	10
Navigation result	√	√	√	√	√	√	√	√	√	√
Successful rate = 100%										
√ = Success										
X = Fail										

Table 7. Results of following a straight wall with variation of start angle

No. of Trial		1	2	3	4	5
Navigation result	Start angle = 20^0	√	√	√	√	√
	Start angle = 30^0	√	√	√	√	√
	Start angle = 40^0	√	√	√	√	√
	Start angle = 50^0	√	√	√	√	√
	Start angle = 60^0	√	√	X	√	√
	Start angle = 70^0	√	X	X	√	√
	Start angle = 80^0	√	√	√	√	X
	Start angle = 90^0	√	√	√	√	√
Successful rate = 90%						
√ = Success						
X = Fail						

4.2. Turning on 90^0 Wall Corner

The scenario of turning on 90^0 wall corner is shown in Fig 10. In this scenario, the robot should follow the wall then turn right on the wall corner. The navigation results are given in Table 8, where two navigation failures are occurred. The navigation failure as shown in Fig. 11 occurs when the robot crashes the wall after turning on the corner. Similar to the previous scenario, this failure might be caused by the instability of the robotic system.

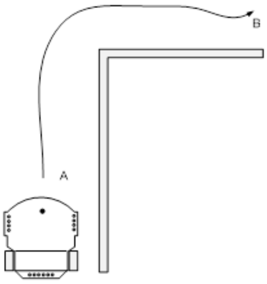


Figure 10. Scenario of turning on 90^0 wall corner.

Table 8. Results of turning on 90^0 wall corner

No. of Trial	1	2	3	4	5	6	7	8	9	10
Navigation result	X	√	√	√	√	√	X	√	√	√
Successful rate = 80%										
√ = Success										
X = Fail										

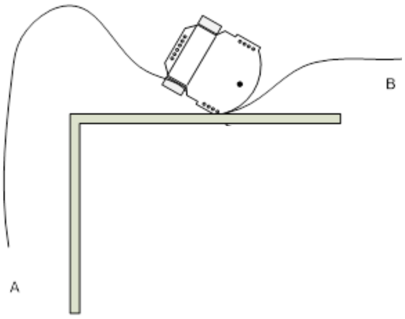


Figure 11. Navigation failure when turning on 90^0 wall corner.

4.3. Turning on 180^0 Wall Corner

The scenario of turning on 180^0 wall corner is shown in Fig. 12, where the robot should make a "U-turn" at the end of the wall. Table 9 shows the navigation results, where the robot navigates successfully on all ten trials. The results indicate that the developed robot is more robust to turn on 180^0 corner than 90^0 corner. It might be observed from the fact that on 180^0 corner, the robot has more time (i.e. make "U-turn") to follow the wall after passing the wall corner than the one on 90^0 corner.

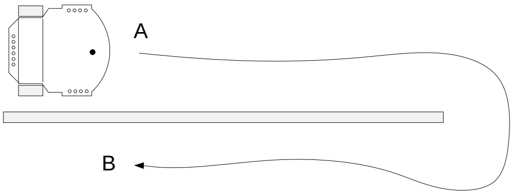


Figure 12. Scenario of turning on 180^0 wall corner.

Table 9. Results of turning on 180^0 wall corner

No. of Trial	1	2	3	4	5	6	7	8	9	10
Navigation result	√	√	√	√	√	√	√	√	√	√
Successful rate = 100%										
√ = Success										
X = Fail										

4.4. Avoiding an Obstacle

The scenario of avoiding an obstacle is shown in Fig. 13. The obstacle is a red colored tube with the diameter of 5 cm and the height of 30 cm. The robot should avoid the obstacle by moving it as shown in the figure. The navigation results are given in Table 10. The table shows that there is one navigation failure (i.e. the robot crashes the obstacle) from ten trials. From the observation, the failure might be caused by the performance of camera sensor due to the lighting variation and the low frame rate, i.e. 10 fps

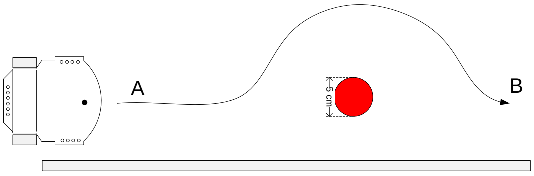


Figure 13. Scenario of avoiding an obstacle.

Table 10. Results of avoiding an obstacle

No. of Trial	1	2	3	4	5	6	7	8	9	10
Navigation result	√	√	√	X	√	√	√	√	√	√
Successful rate = 90%										
√ = Success										
X = Fail										

4.5. Following the Walls and Avoiding the Obstacles

The scenario of following the walls and avoiding the obstacles is shown in Fig. 14. In this scenario, the robot should follow four walls called as W1, W2, W3, W4, and avoid two obstacles called as OBS1 which is located in the area of W1 and OBS2 which is located in the area of W2. The real photo of the robot, the walls and the obstacle are shown in Fig. 15.

In the experiments, the navigation failure in each task is computed as given in Table 11. From the table, it is obtained that the successful rates of wall following, obstacle avoiding, and overall tasks are 85%, 80% and 83.33% respectively. From the results, it could be said that the developed mobile robot is able to perform the wall following and the obstacle avoiding tasks with the high successful rate using the low cost sensor systems.

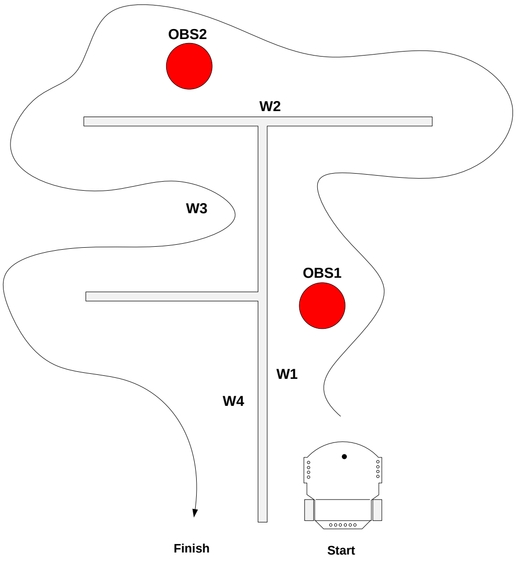


Figure 14. Scenario of following the walls and avoiding the obstacles.

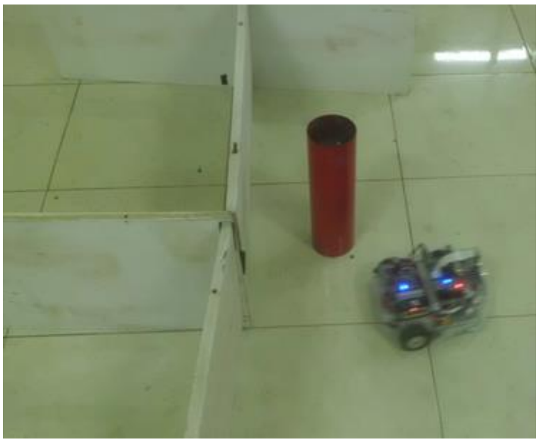


Figure 15. The real photo of the robot, the walls and the obstacle.

Table 11. Results of following the walls and avoiding the obstacles

No. of Trial	1	2	3	4	5	6	7	8	9	10
Navigation result on W1	√	√	√	√	√	√	√	√	√	√
Navigation result on W2	√	√	X	X	X	√	√	√	√	X
Navigation result on W3	√	X	√	X	√	√	√	√	√	√
Navigation result on W4	√	√	√	√	√	√	√	√	√	√
Navigation result on OBS1	X	√	√	√	√	√	√	X	√	√
Navigation result on OBS2	√	√	√	√	X	X	√	√	√	√
Successful rate of wall following = 85%										
Successful rate of obstacle avoiding = 80%										
Overall successful rate = 83.33%										
√ = Success										
X = Fail										

4.6. Comparison to the Existing Systems

To evaluate the effectiveness of our proposed system to the existing ones, we compare the sensor types and the features as given in Table 12. As shown in the table, our proposed system provides the complete solution for both tasks, i.e. the wall following and the obstacle avoiding tasks which are applied simultaneously on a single robot. Further our method exploits the benefits of FLC for handling both tasks. In the context of sensor technology, our method employs the common ultrasonic sensor for wall following task and the advanced camera sensor for detecting obstacle. Compared to [4], [7-9], our camera sensor offers the flexible detection of the obstacles.

Table 12. The comparison of existing systems and proposed system

Ref.	**Wall following mode**			**Obstacle avoiding mode**		**Remark**
	Sensor type	Wall corner	Method	Sensor	Method	
[3]	Ultrasonic	90^0, 180^0	FLC	NA	NA	
[4]	Ultrasonic, Sharp Infra red	Polygon	FLC	Heat sensor	Presence detection	Robot detects the presence of the human and send an alarm
[5]	NA	Convex, concave	FLC	NA	NA	MATLAB Simulation
[7]	NA	NA	NA	Ultrasonic, wheel encoder	Rule based system	
[8]	Ultrasonic	90^0, 180^0 (*)	FLC	Ultrasonic	FLC	* The wall is considered as the obstacle
[9]	NA	NA	NA	Infrared*	FLC	* MATLAB and WEBOT Simulation
[11]	NA	NA	NA	Camera	Image processing	
Proposed system	Ultrasonic	90^0, 180^0	FLC	Camera	Image processing, FLC	

5. CONCLUSION

The sensor fusion consists of ultrasonic sensors and camera sensor are employed as the inputs of the Fuzzy Logic Controllers to steer the mobile robot for following the wall and avoiding the obstacle. The method offers the effective approach to move a robot by following the wall, while avoiding the obstacles. The experimental results show that the successful rate of wall following technique is higher than obstacle avoiding technique. In the experiments, the obstacle is still limited to the red colored object.

Compared to the existing systems, our approach provides the effective solution, in which both wall following and obstacle avoiding tasks are implemented on a single robot. Further the camera sensor could be exploited to detect various obstacles.

In future, the techniques to handle more complicated obstacles will be developed. Further, the control techniques will be extended to cope with the real condition.

REFERENCES

[1] Prayudhi LH, Widyotriatmo A, Hong KS. *Wall following control algorithm for a car-like wheeled-mobile robot with differential-wheels drive*. Proceedings of the 15th International Conference on Control, Automation and Systems (ICCAS). Busan. 2015: 779-783.

[2] Lee YT, Chiu CS, Kuo IT. *Fuzzy wall-following control of a wheelchair*. Proceedings of Joint 17th World Congress of International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS). Otsu. 2017: 1-6.

[3] Fahmizal, Kuo CH. *Development of a fuzzy logic wall following controller for steering mobile robots*. Proceedings of International Conference on Fuzzy Theory and Its Applications (iFUZZY). Taipei. 2013: 7-12.

[4] Lo CW, Wu KL, Liu JS. *Wall following and human detection for mobile robot surveillance in indoor environment*. Proceedings of IEEE International Conference on Mechatronics and Automation. Tianjin. 2014: 1696-1702.

[5] Li X, Wang D. *Behavior-based mamdani fuzzy controller for mobile robot wall-following*. Proceedings of International Conference on Control, Automation and Robotics. Singapore. 2015: 78-81.

[6] Yang ZY, Juang CF, Jhan YH. *Hexapod robot wall-following control using a fuzzy controller*. Proceedings of the 11th IEEE International Conference on Control & Automation (ICCA). Taichung. 2014: 574-578.

[7] Zaki AM, Arafa O, Amer SI. Microcontroller-based mobile robot positioning and obstacle avoidance. *Journal of Electrical Systems and Information Technology*. 2014; 1: 58-71.

[8] Al-Mutib K, Abdessemed F, Faisal M, Ramdane H, Alsulaiman M, Bencherif M. *Obstacle avoidance using wall-following strategy for indoor mobile robots*. Proceedings of the 2nd IEEE International Symposium on Robotics and Manufacturing Automation (ROMA). Ipoh 2016: 1-6.

[9] Ibrahim MI, Sariff N, Johari J, Buniyamin N. *Mobile robot obstacle avoidance in various type of static environments using fuzzy logic approach*. Proceedings of the 2nd International Conference on Electrical, Electronics and System Engineering (ICEESE). Kuala Lumpur. 2014: 83-88.

[10] Kang MC, Kim KS, Noh DK, Han JW, Ko SJ. A robust obstacle detection method for robotic vacuum cleaners. *IEEE Transactions on Consumer Electronics*. 2014; 60(4): 587-595.

[11] Kim J, Do Y. Moving obstacle avoidance of a mobile robot using a single camera. *Procedia Engineering*. 2012; 41: 911-916.

[12] https://www.parallax.com/product/28015

[13] https://www.raspberrypi.org/documentation/hardware/camera/README.md

[14] https://opencv.org/