Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 55-1 (2022) 120–125

Intelligent Obstacle Avoidance Controller for QBot2

Muhammad Usman Asad*. Jason Gu.* Umar Farooq*. Rajeeb Dey. Valentina E. Balas***. Ghulam Abbas******

** Department of Electrical and Computer Engineering, Dalhousie University, Halifax, NS B3H 4R2 Canada (e-mail: usmanasad01@hotmail.com, Jason.gu@dal.ca, engr.umarfarooq@yahoo.com)*

*** Electrical Engineering, National Institute of Technology Silchar,*

Assam, India, (e-mail: rajeeb@ee.nits.ac.in)

**** Department of Automatics and Applied Software, "Aurel Vlaicu" University of Arad, Arad, Romania, (e-mail: valentina.balas@uav.ro)*

**** Department of Electrical Engineering, The University of Lahore, Lahore, Pakistan, (e-mail: ghulam.abbas@ee.uol.edu.pk)}*

Abstract: This paper describes the design process and the steps involved in the implementation of a fuzzy logic-based controller for autonomous navigation of an educational robot called Qbot2. The differential-driven robot is equipped with three ultrasonic sensors (SRF-04) mounted on the front bumper and DC motors with built-in encoders and current sensors. The controller takes the inputs from three ultrasonic sensors and generates the speed commands to avoid any obstacle in its path. PCB is designed to process all the data from the sensors and direct the signal to the motors via motor driver circuits. The low-level implementation of the hurdle avoidance controller is implemented using an inexpensive Arduino UNO in real-time. The controller performance is validated through an experimental run. The designed platform can be used to implement various fuzzy inference systems in real-time and hence can be utilized as laboratory work for testing soft computing algorithms on mobile robotics.

Keywords: Fuzzy Logic, Robot sensing system, Mobile robot, Trajectory, Microcontroller

1. INTRODUCTION

Over the past decades researchers are focusing on autonomous operation of mobile robot because of their potential applications in hazardous environments (G. Kantor et al (2006)), agro-vision and harvesting (B. Astrand et al (2002)), house-hold tasks (M. Hans et al (2001)) and medical applications (J. Eriksson et al (2005)). There are various technological framework used in those application such as self-localization for autonomous navigation, remote control of mobile robot, obstacle avoidance capability using sensors and intelligent decision making to perform a certain task (R. Murphy (2000)). Soft computing techniques for instance fuzzy logic and neural networks have been used by many researchers to develop a better control design for autonomous applications. (U. Farooq et al (2010)), (Xie et al (2014)), (Bais et al (2006)) proposed an autonomous vehicle design equipped with ultrasonic sensors, GPS, wheel encoders, digital compass, and GSM module. (Y Peng et al (2015)) propose an efficient way to do obstacle detection and avoidance based on 2-D lidar algorithm and (R. Carelli et al (2003)) designed a controller for mobile to navigate in a structured/indoor environment using a wall following technique. Classical/conventional control requires complete mathematical model of robot which is somehow difficult to obtain due to the inherent nonlinearities and environmental uncertainties. Soft computing techniques becomes popular because they do not require any mathematical model to design a controller. Fuzzy logic and neural networks are two most important techniques in this paradigm. Fuzzy logic incorporates human experience in form of rules while on the other hand neural network used experimental data to design the controller (U. Farooq et al (2012)). (Yi Jincong et al(2009) proposed an intelligent fuzzy based control algorithm for obstacle avoidance using multi sensor fusion technology. Similar work by (W. Gueaieb et al (2008)), in which an autonomous mobile robot capable to navigate in unstructured and unknown environments using radio frequency technology (RFID). It gives an alternate way to help navigating robot without mapping workspace and any vision system. (Paul-Onut Negirla and Mariana Nagy (2018)) build a multi sensor platform to test fusion algorithms. In another paper (Faten Cherni et al (2017)) propose a hurdle avoidance algorithm that will work in a dynamic environment. The designed robot changes its path every time it detects the obstacle. In (Hajer Omrane et al (2016)) work, fuzzy controller is designed for mobile robot navigation and simulated in MATLAB and SIMIAM software. In (Gyula Mester (2010) work, fuzzy control is designed to navigate in an environment which contains hurdles and slopes. L.A Zadeh (L.A Zadeh (1965)) has introduced the concept of fuzzy set theory. It became readily popular in control community and getting used in application that has high degree of uncertainty, nonlinearities, and complexities due to its ability to deal with imprecise and vague information. In this paper, fuzzy logic tool is used to design an obstacle avoidance controller for QUANSER Qbot2. For distance measuring, two sonar sensors

10.1016/j.ifacol.2022.04.020

(SRF-04) is used, and output will be the variable duty cycles to left and right motor to control the speed of robot. The controller is designed in fuzzy logic toolbox available in MATLAB. The designed controller is validated in the designed arena with multiple obstacles. Controller performance is being evaluated by changing the hurdle locations within the arena and results proves its validity in real time.

2. MOBILE ROBOT ARCHITECTURE

The experimental prototype used is a QUANSER Qbot2. It is a differential driven robot with two DC motors with built-in encoder and current sensors. It also has a shock absorber feature to run on a rough or bumpy surfaces. It has a round chassis design with a diameter of 34cm and height is 10cm. Qbot2 Kobuki platform also comes with bumper sensors as well as built in gyroscope and cliff sensors. It also has caster wheel at the front and back for the support. In addition to the sensory system of the Qbot2, two sonar sensor named LS and RS is installed on the front bumper for obstacle avoidance. Some of the robot specifications and model parameter are listed in table 1.

Table 1. Qbot2 robot specifications and model parameters

Symbol	Description	Value	Unit
D	Diameter of the Qbot 2	0.35	*m*
d	Distance between left and right wheels	0.235	*m*
h	Height of the Qbot 2	0.10	*m*

The Qbot2 is powered with 12V Lithium-ion battery pack which is placed underneath the robot. It is solely used to power the motor driver circuitry. To power up the microcontroller (Arduino UNO) and sensors, another 9.6V battery is used. The robot chassis is shown in fig 1.

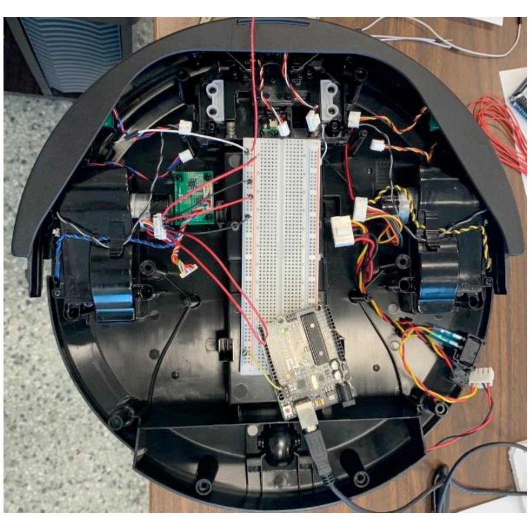


Figure 1. Qbot-2 mobile robot platform

To drive the DC motors from microcontroller, 7A/160W dual DC motor driver module is used can be seen in fig 2. It can drive two DC motors with the ability to provide single 7A current rating and high power. Optocoupler is provided to isolate the input signal from the output without interference. The control signal logic for the motor is shown in table 2.

Table 2. Control signal logic for dual DC motor driver

IN1	IN2	ENA1	OUT1, OUT2
0	0	×	Brake
1	1	×	Floating
1	0	PWM	Forward to speed
0	1	PWM	Reverse speed

A dedicated PCB is designed to process all the information from the sensors and wheel encoders to the microcontroller. The layout of the PCB is shown in fig 3.

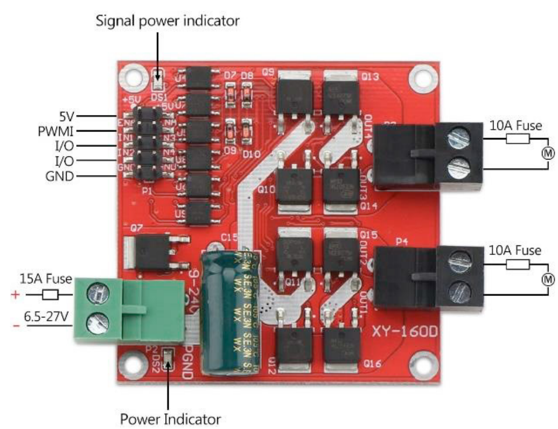


Figure 2. Dual DC motor driver Module

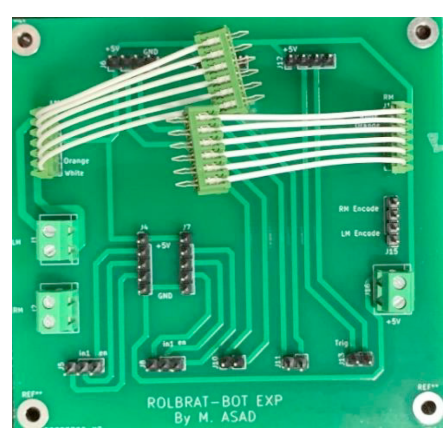


Figure 3. PCB design for Qbot2 experimentation

The final assembled chassis of Qbot2 used for real time testing is shown in fig 4. All the sensors and circuits is installed on the top of mobile robot.

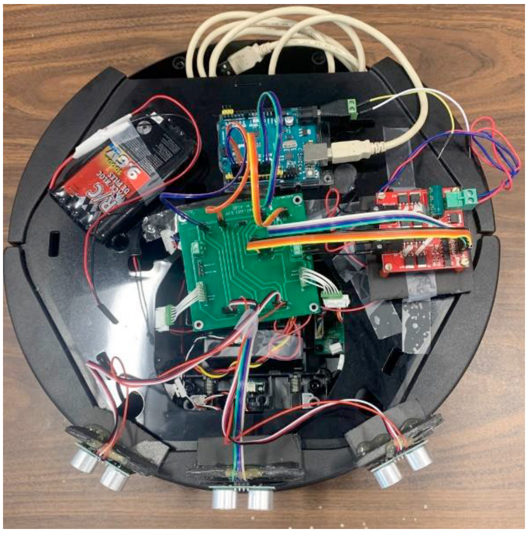


Figure 4. Qbot2 mobile robot

3. OBSTACLE AVOIDANCE CONTROLLER DESIGN

MATLAB® Fuzzy logic toolbox is used to design the fuzzy logic controller (FLC). It has user friendly features that allow user to quickly design, test and simulate. It also permit user to modify a fuzzy inference system (MATLAB® (2020)). In this paper, Sugeno inference engine is used to design two input and two output fuzzy logic controller to navigate the robot into the arena with multiple hurdles shown in fig 5.

Figure5. Hurdle avoidance arena – Robotics Lab

2.1 Fuzzification

The controller taken input from the sonar sensors named as Left Sensor (LS) and Right Sensor (RS). These sensors are +45° and -45° apart with respect to vehicle axis. The distance from any object is measured using sonar sensors and controller fuzzifies these input values known as fuzzification. Fuzzy sets is defined at this stage namely Near, Med and Far with the universe of discourse. The universe of discourse is the range of sensors which is determined experimentally. It depends on the cone of the sensor and dimension of the vehicle. Degree of membership is determined based on the sensor value such as if it is below a minimum threshold then it will automatically set to minimum and vice versa. There is also a possibility that sensor value fall between two fuzzy sets i.e., it could be in 'Near' and at the same time in 'Med' membership functions. The block diagram of the controller is also shown in fig 6. Keeping in view these situations and also low-level implementation process in microcontroller, the fuzzy sets are

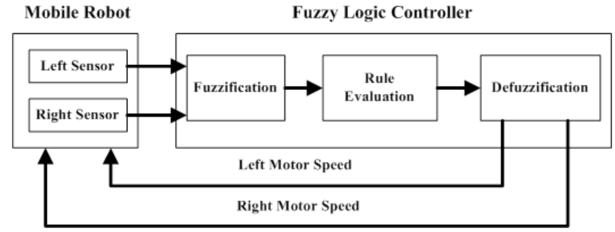
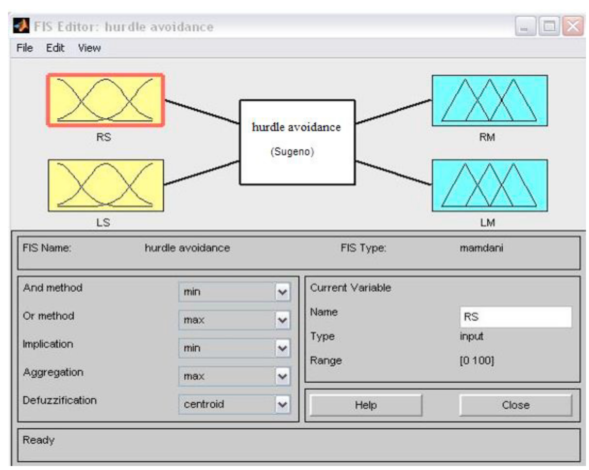


Figure 6. Fuzzy logic controller for Qbot2

described as triangular membership functions. It would then be far easier for controller to determine the degree of membership because a simple line equation will be programmed to check the degree of belongingness based on the sensor value. The typical triangular membership function can be described in (1):

$$triangle(x;a,b,c)=\begin{cases}0, x<a\\ \dfrac{x-a}{b-a}, a\le x\le b\\ \dfrac{c-x}{c-b}, b\le x\le c\\ 0, c\le x\end{cases} \tag{1}$$

Triangular membership functions are used to define all the linguistic variables as shown in fig 7.



(a)

2.2 Fuzzy Rule Base

Nine rules are designed to tackle the hurdles in the designed arena. Two rules at maximum are fired at any instant out of the nine rules. These rules also describe a relation between the sensor value and motor speed. In addition to that, control surfaces are being plotted to give a better visualization of how motor speed is varying with respect to sensor values is shown in fig 8. The rule base for right and left motor is tabulated in Table 3 and 4.

2.3 Fuzzy Implications

Mamdani method is selected to analyze the fuzzy implication on the designed rule base. After fuzzifying the sensors values, controller execute the rules as per the defined threshold values and the degree of fulfillment (DOF) of each rule is computed using AND operator. Accordingly, the output membership functions are truncated at that DOF level.

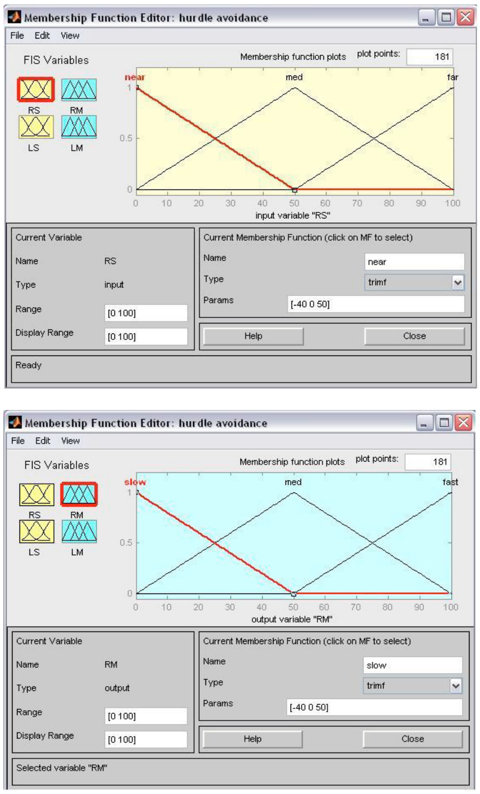
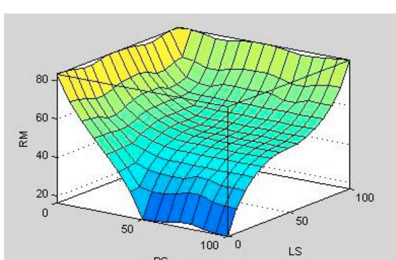
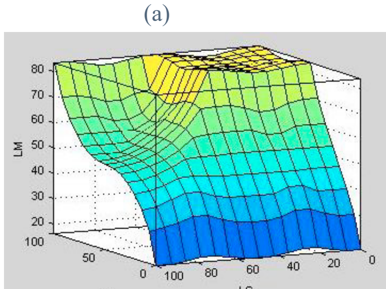


Figure 7. (a) FLC GUI (b) Input MF (c) Output MF



(a)



(b)

Figure 8. (a) Control Surface for Right Motor (b) Control Surface for Left Motor

To yield the final fuzzy output, all the rules are evaluated and by aggregating them all in a cumulative way using OR operator. The fuzzy implication process in shown in fig 9. It helps to determine the consequent part of each rule. In this case Sugeno implication method is selected. An example of implication process is shown in fig 9, where right sensor measures ‘far’ while left sensor measures ‘near’.

Table 3. Rule base for Right Motor (RM)

RS/LS	Near	Med	Far
Near	Fast	Fast	Fast
Med	Slow	Med	Fast
Far	Slow	Med	Fast

Table 4. Rule base for Left Motor (LM)

RS/LS	Near	Med	Far
Near	Slow	Slow	Slow
Med	Fast	Fast	Med
Far	Fast	Fast	Fast

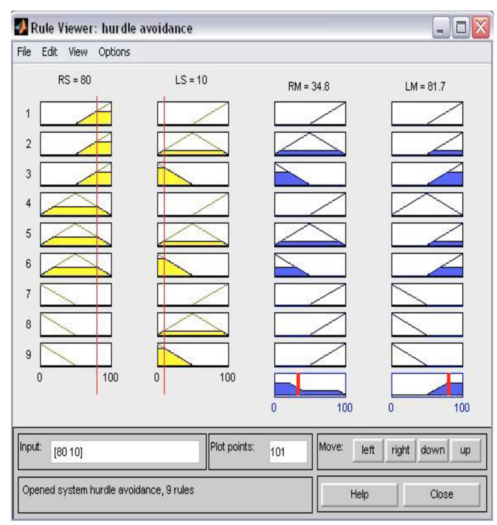


Figure 9. Fuzzy Implication process

2.4 Defuzzification

The process of converting the fuzzy output to crisp values is known as defuzzification. Centre of aggregation (COA) is used in this case and can be described as:

$$Z_o = \frac{\sum_{i=1}^{n} Z_i \mu_{out}(Z_i)}{\sum_{i=1}^{n} \mu_{out}(Z_i)} \qquad (2)$$

where $\mu_{out}(Z_i)$ are the i=1,2,…,n sampled values of the aggregated output membership function and Z_o is the crisp values for the PWM signals to control the speed of motor.

4. FUZZY CONTROLLER IMPELEMENTATION

To implement the fuzzy controller, a single-chip Arduino UNO microcontroller is used. All the algorithm computations and data from the sensors and wheel encoders altogether are processed in a single Arduino. The block diagram for the implementation is shown in fig 10.

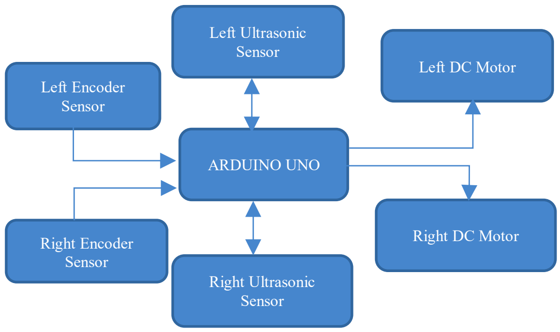


Figure 10. Single Chip microcontroller implementación

In Arduino UNO, algorithm read the sonar sensors (range is defined [0, 100]) and do fuzzification process and then PWM signals are generated (range is defined [0, 50]) after the rule base is being executed. All the algorithm is written in C language in Arduino IDE platform. The main steps involved in real-time implementation is described as follow:

4.1 PWM initialization

An array of 50 characters is initialized to store the output of fuzzy controller. Frequency of PWM is set to 100Hz and duty cycle to 100%.

4.2 Sonar initialization

The sonar sensor is triggered from the microcontroller to read the distance information. This distance measurement is then scaled to between 0 and 100.

4.3 Calculating the linguistic variables

Degree of membership for each linguistic variable (near, med, far) is calculated. The range is defined in a whole number instead of floating point to reduce the computational complexity and reduce burden on microcontroller. For this reason, data type of "unsigned char" is selected during programming.

4.4 Rule base evaluation and execution at low level

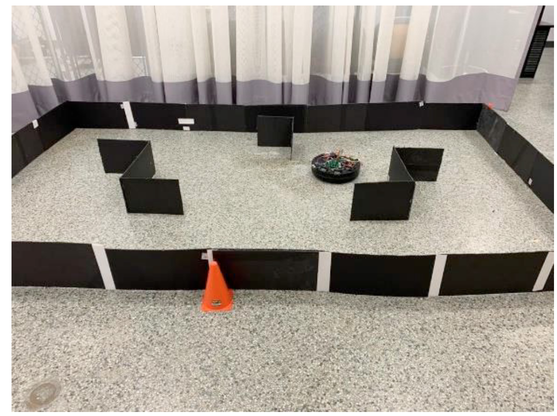
All the rules are executed in a sequential manner. Membership functions are described as equation of line for simplicity. The loop is created which will store the result of each rule when it is fired. The loop will check DOF for each and every rule and finally give the fuzzy output which is the right motor speed.

4.5 Defuzzification

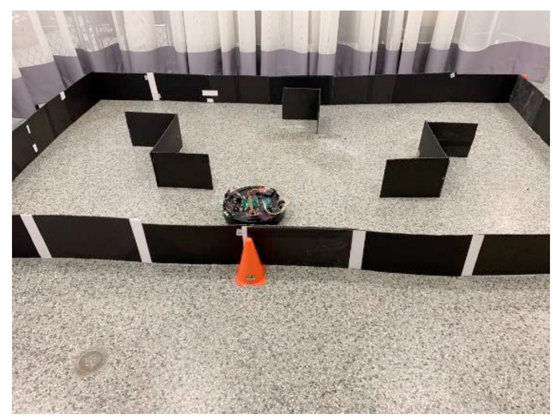
Defuzzification is performed using the method of COA. A loop is performer the sum of product of all the values at every index of array. Then, it will be divided by summing the values at all the indexes of array to calculate crisp output. The scaling is done at the output to bring the value in the defined range [0,50].

5. EXPERIMENTAL RESULTS

The controller is tested in real time on QUANSER Qbot2 differential driven platform. The resulting motion of the robot is shown in the demonstration at the following link https://youtu.be/pQ1dTXWS2aw and in figure 11.



(a)



(b)

(c)

(d)

Figure 11. (a), (b), (c), (d) Hurdle Avoidance steps

6. CONCLUSIONS

In this paper, a hurdle avoidance controller is designed using fuzz logic toolbox. The controller is tested in a real time specially designed arena with obstacles. The controller is implemented using Arduino UNO microcontroller on a QUANSER Qbot2 platform. The real time performance of the controller has showed the validity of the proposed system.

REFERENCES

Abdul Bais., Roberet Sablatnig., and Jason Gu. (2006). Single Landmark Based Self-localization of Mobile Robots. *The Third IEEE Canadian Conference on Computer and Robot Vision*, Quebec, Canada.

Faten Cherni., Maissa Boujelben., Lotfi Jaiem., Yassine Boutereaa., Chokri Rekik andNabil Derbel. (2016). Autonomous mobile robot navigation based on an integrated environment representation designed in dynamic environments. *International Journal of Automation and Control (IJAAC),* 11(1), pp-35-53.

Hajer Omrane, Mohamed Slim Masmoudi, and Mohamed Masmoudi. (2016). Fuzzy logic-based control of autonomous mobile robot navigation. *Computational Intelligence and Neuroscience,* Hindawi.

Gyula Mester. 2010. Intelligent Mobile Robot Motion Control in Unstructured Environments. *Acta Polytechnica Hungarica,* 7(4), pp. 153-165

Ricardo Carelli, and Eduardo Oliveira Freire. (2003). Corridor navigation and wall following stable control for sonar based mobile robots. *Robotics and Autonomous Systems*, 45, pp. 235-247.

S Xie., P Wu., Y Peng., J Luo., D Qu., Q Li and J Gu. (2014). The obstacle avoidance planning of USV based on improved artificial potential field. *IEEE International Conference on Information*, pp746-751, Hailar, China

L. A. Zadeh. *Fuzzy sets*.(1965). Information and Control, pp. 338-353

MATLAB® (2020). Fuzzy Logic Toolbox, Mathworks

Paul-Onut Negirla and Mariana Nagy. (2018). Mobile robot platform for study sensors fusion localization algorithms. *8th International workshop Soft Computing applications (SOFA),* Arad Romania.

Umar Faooq., Muhammad Amar., Eitzaz ul Haq., Muhammad Usman Asad and Hafiz Muhammad Atiq. (2010). Microcontroller based neural network controlled low-cost autonomous vehicle. *International Conference on Machine Learning and Computing*, pp. 96-100, Bangalore, India

Umar Farooq., K.M. Hasan and Syed Omer Saleh. (2012). Fuzzy logic-based wall tracking controller for mobile robot navigation. *IEEE Conference on Industrial Electronics and Applications (ICIEA),* Singapore.

Yi Jincong., Zhang Xiuping., Ning Zhengyuan and Huang Quanzhen. (2009). Intelligent Robot Obstacle Avoidance System Based on Fuzzy Control. *1st International Conference on Information Science and Engineering (ICISE)*, pp. 3812-3815, Nanjing, China.

Y Peng., D Qu., Y Zhong., S Xie., J Luo and J Gu. (2015). The obstacle detection and obstacle avoidance algorithm based on 2-d lidar. *IEEE International conference on information and automation*, pp-1648-1653, Lijiang, China

W. Gueaieb and Md. S. Miah. (2008). An intelligent mobile robot navigation technique using RFID Technology. *IEEE Transactions on Instrumentation and Measurement*, 57(9), pp. 1908-1917.