Obstacle avoidance behavior of an autonomous mobile robot in a radioactive environment based on fuzzy logic

Yassine Haddi * and *Amina* Kharchaf

Ibn Tofail University, Faculty of Science, Department of physics, Laboratory Modelisation and Science engineering, Kénitra, Morocco

Abstract. Five years after the Fukushima nuclear disaster, many robots have been developed to enter areas that are too radioactive for humans, in the event of a nuclear accident, measuring the level of radioactivity can be difficult and risky. A new detection system has been developed to come to the aid of plant operators and managers in certain emergency situations. While navigating in a radioactive environment and looking for sources.to facilitate this operation and make it safer, we have to assess the autonomy of this system, which is a robot operating in a radioactive environment based and to guarantee autonomy and intelligence for obstacle avoidance behavior, we will use a tool such as Fuzzy Logic (LF). In this article, we present an obstacle avoidance approach using this tool. The approach is validated by a virtual simulation on Matlab Simulink.

1 Introduction

Nowadays, researchers have at their disposal the hardware and software resources needed to build an intelligent dynamic system such as a mobile robot autonomous, used in various domain such as space research, industry, agriculture, military [1 - 5].

They are much more effective in developing intelligent systems than conventional methods. Among these tools, we can cite Fuzzy Logic (LF), which makes it possible to translate the human experience into a set of rules.

In addition, a mobile robot, to be autonomous and intelligent, must have avoidance behavior obstacles that allow him to acquire the ability to move in its environment. To reach this goal, classical approaches have been replaced by based on soft computing using in particular LF.

The objective of this article is therefore to study the problem of obstacle avoidance behavior in an environment radioactive.

In [6], the authors present a new navigation scheme which decomposes the 3D navigation problem into several 2D navigation problems. Every 2D navigation system uses preference-based fuzzy behavior controller. G. Antonelli et al. [7] focused their attention on solving the problem of autonomous path following in the context of fuzzy logic. In [7], the authors present an innovative mobile robot navigation technique using radio frequency identification (RFID) technology for inland navigation with a fuzzy logic controller, and in [8], the authors aim to eliminate the aforementioned drawbacks and intend to extend the surveillance control theory of FDES Fuzzy Discrete Event Systems by redefining controllable and uncontrollable fuzzy events to accommodate a robust behavior coordination mechanism for the navigation task of the mobile robot, in [9] this article proposes a Simulink model which drives the robot from start to finish avoiding obstacles using the fuzzy logic controller in addition to path tracking using the path following controller following the controller.

This article presents the principle of the fuzzy logic strategy for navigation based on the behavior of a mobile robot in unfamiliar environments using the fusion of data from sounders and position sensors. In fact, the adoption of such an approach leads to the use of fuzzy logic modules whose design is simple, fast, inexpensive and easily maintained because the rules can be interpreted linguistically by a human expert. The organization of this article is as follows: in section 2, we briefly describe the Pioneer 3-DX mobile robot. Section 3 describes behavior-based navigation of a mobile robot. Section 4 presents the fuzzy design of the decision and control algorithm. Section 5 gives the results of the simulation. Finally, this document is concluded in the section

2 Description of robot mobile

The Pioneer 3-DX mobile robot is shown in Figure 1. The platform is assembled with 500 tick motors encoders, 19 cm wheels, robust aluminum body, 8 front faces ultrasonic sensors (sonar), 8 optional real steerable sonars, 1, 2 or 3 hot-swappable batteries and a complete software development kit. The Pioneer 3-DX base platform can reach speeds of 1.6 meters per second and carry a payload of up to 23 kg [11], [12].

The Pioneer 3-DX robot is a versatile base, used for research and applications involving mapping, teleportation, location, surveillance, reconnaissance and other behaviors.

* Corresponding author: yassine.haddi2@gmail.com

Fig. 1. Pioneer 3-DX mobile robot

3 Navigation based on the behavior of a mobile robot

The general program made to orient the robot on its path and tell it to deviate if there is an obstacle, in our project the method we used for the deviation of the robot is fuzzy logic.

First, we will give some explanations about the program:

The robot moves in a rectangular path, and at each position of the path, it checks the state of the obstacle, that is, it calculates the distance between its position and that of the obstacle by the 'following mathematical application:

If we have taken the coordinates of the robot = [x y], and that of the obstacle = [u v], then the distance between the robot and the obstacle is:

Distance = sqrt ((u-x) ^ 2 + (v-y) ^ 2).

Then it takes this distance and compares it with a predetermined value (the minimum distance for the robot to deviate), for example, if the distance between the robot and the obstacle is <20, it deviates. After the call to fuzzy logic, the result will be a new distance and a new angle, and besides, we get the new coordinates of the positions of the Robot (Xr, Yr). These positions are respectively the coordinates obtained after the deviation of the latter.

4 Fuzzy control algorithm

A solution to the navigation problem for mobile robots is based on adopting a fuzzy logic-based approach, as shown in Figure 3.

The use of these rules in a fuzzy system is done as follows:

- « Fuzzyfication » of inputs
- Combination of degrees of membership
- Combination with ruler weights
- Aggregation of outputs from different rules
- "Defuzzyfication" of the output thus calculated

Application: fuzzy controller for obstacle avoidance:

An example of the use of fuzzy logic in robotics is autonomous navigation in an unfamiliar environment with obstacle avoidance. This type of application can be quite complex and has robustness and speed constraints which can be satisfied by using a fuzzy controller. The end is to allow navigation avoiding collisions with obstacles.

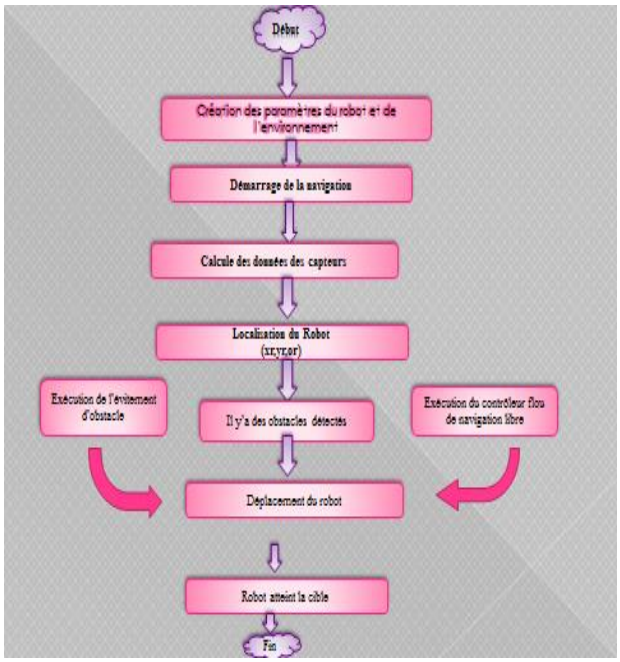

Fig. 2. Navigation behaviors

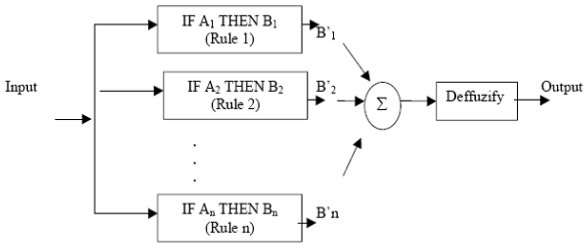

Fig. 3. Fuzzyfication

To do this, we use as inputs the distance and direction read by the ultrasonic sonar sensor from the obstacle closest to the robot and as outputs we will have the direction to follow and the distance to travel. The output variables will therefore have the same structure as the input variables.

The space analyzed by the rangefinder is divided as in figure 4.

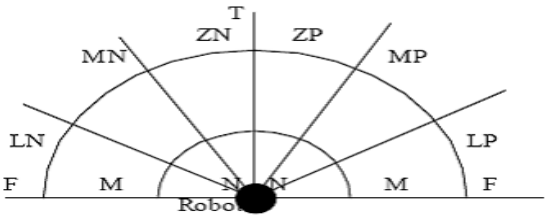

Fig. 4. space analyzed by robot

This behavior depends only on information from position sensors, such as odometry, artificial vision, GPS, etc. This is the main task for the robot, so the main

inputs to this behavior are the measurements of position X (x, y, θ) and target position Xr (xr, yr, θr).

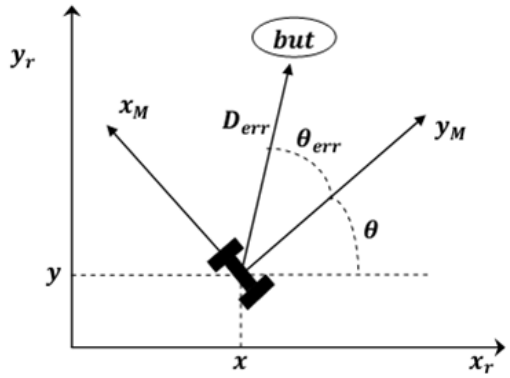

Fig. 5. Position of robot

In this case, we have two fuzzy logic inputs (Derr, θerr)

or:

θerr: the angle error which is the difference between the angle of the target to be reached and the robot's current angle error.

Drr: the distance error which is the difference between the target position and the current position of the robot

The shape of the fuzzy sets for distance are shown in the following figure:

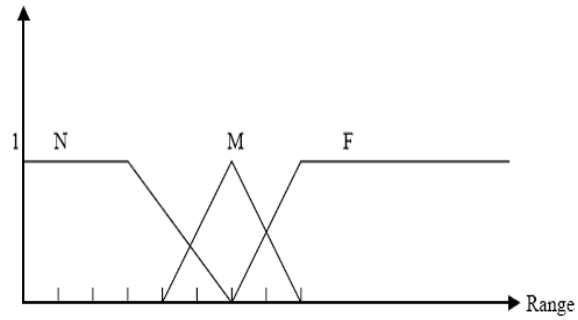

Fig. 6. Membership functions of distance error

The shape of the fuzzy sets for direction is shown in the following figure.

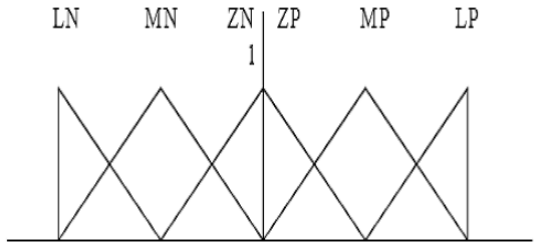

Fig. 7. Membership functions of the angle error

The values of the parameters of the membership functions depend on the sensitivity of the range finder used, and for this reason most often the data is normalized.

Once we have the data relating to the closest obstacle, we can determine the degrees of belonging to the sets we have just shown.

We then apply the following rules for obstacle avoidance:

- IF obstacle is NEAR AND ZERO POSITIVE THEN move MEDIUM AND MEDIUM NEGATIVE
- IF obstacle is MEDIUM AND ZERO POSITIVETHEN move NEAR AND MEDIUM NEGATIVE
- IF obstacle is NEAR AND MEDIUM POSITIVE THEN move MEDIUMAND NEAR NEGATIVE
- IF obstacle is MEDIUM AND MEDIUM POSITIVE THEN move NEAR AND MEDIUM NEGATIVE
- IF obstacle is NEAR AND ZERONEGATIVE THEN move MEDIUM AND MEDIUM POSITIVE
- IF obstacle is MEDIUM AND ZERO NEGATIVE THEN move NEAR AND MEDIUM POSITIVE
- IF obstacle is NEAR AND MEDIUM NEGATIVE THEN move MEDIUM AND NEAR POSITIVE
- IF obstacle is MEDIUM AND MEDIUM NEGATIVE THEN move NEAR AND MEDIUM POSITIVE

No more rules activated if the path is free:

- IF obstacle is LARGE POSITIVE OR LARGE NEGATIVE OR FAR THEN move FAR AND NEAR POSITIVE AND NEAR NEGATIVE

We can also add other types of rules, such as reaching a certain goal.

In the implementation we use the min for the AND, the max for the OR and for the aggregation of the rules, and the method of the centroid for the defuzzyfication of the results.

With such a type of system, it is possible to obtain very varied behaviors, by simply changing the membership functions, the normalization constants or by adding additional rules.

We worked with the scientific calculation software Matlab which facilitated the task of developing the simulation of our application. The implementation of this application includes two main lines which are as follows:

- Give the robot the positions to go, in this case the ground truth.
- Use of fuzzy logic for obstacle detection after having calculated the distance and the orientation of the robot to the sensor.

We used Matlab software for the simulation of the movement of the autonomous robot. In our case, the robot must follow a predetermined path avoiding the obstacles that exist in its path.

First, we will start with the simulation on Matlab, showing the different programs intended to complete this task.

The general program made to orient the robot on its path and tell it to deviate if there is an obstacle, in our article the method we used for the deviation of the robot is fuzzy logic. First, we will give some explanations on:

The program

The robot moves in a rectangular path, and at each position of the path, it checks the state of the obstacle,

that is, it calculates the distance between its position and that of the obstacle by the 'following mathematical application:

If we have taken the coordinates of the robot = [x y], and that of the obstacle = [u v], then the distance between the robot and the obstacle is:

Distance = sqrt ((u-x) ^ 2 + (v-y) ^ 2).

Then it takes this distance and compares it with a predetermined value (the minimum distance for the robot. to deviate), for example, if the distance between the robot and the obstacle is <20, it deviates. After the call to fuzzy logic, the result will be a new distance and a new angle, and besides, we get the new coordinates of the positions of the Robot (Xr, Yr). These positions are respectively the coordinates obtained after the deviation of the latter.

Part of the captured Matlab program is as follows:

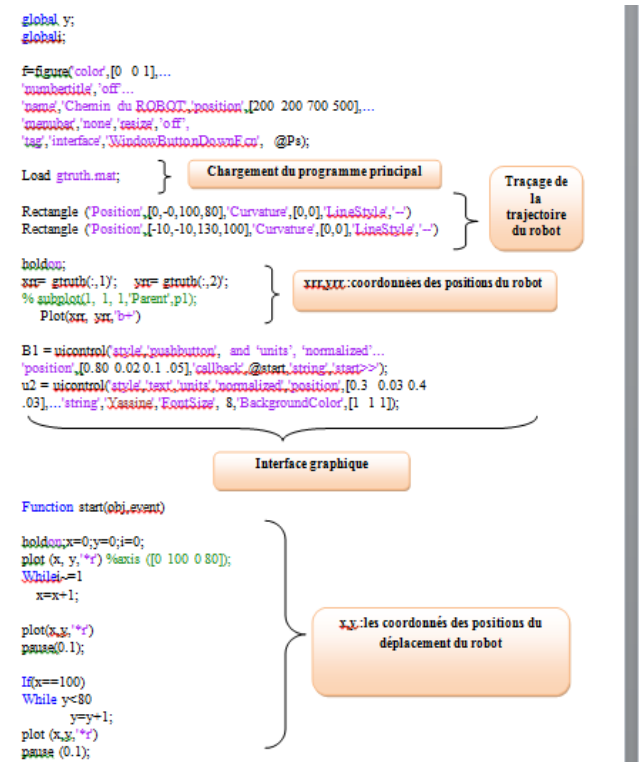

Fig. 8. Program of navigation

5 Implementation & results

The following figures represent the movement of the robot in a field by indicating the positions traveled estimated positions (Xr, Yr)

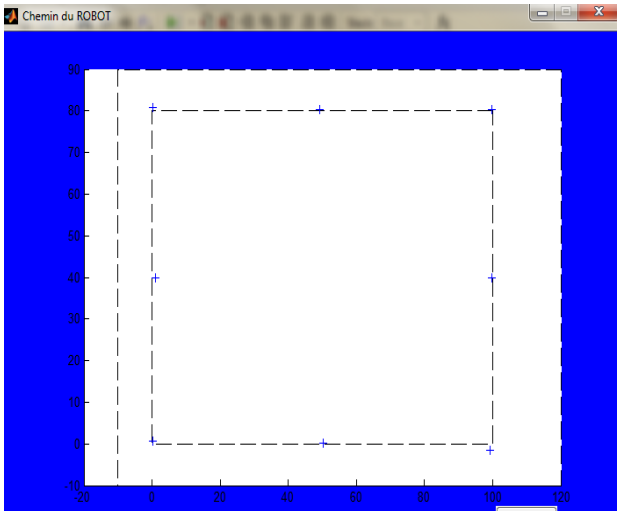

Fig. 9. The points to be covered by the robot

After START, the result:

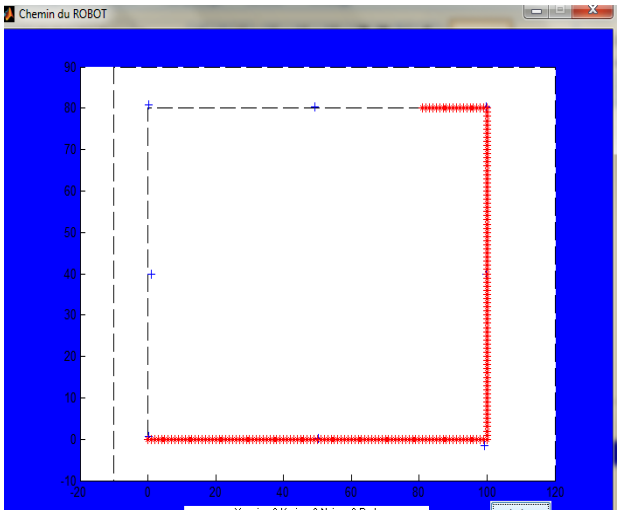

Fig. 10. Trip made by the Robuter without using the LF

« * »: The positions traveled by the robot
: positions of obstacle
: New position of robot after deviation
« + »: The estimated position of robot

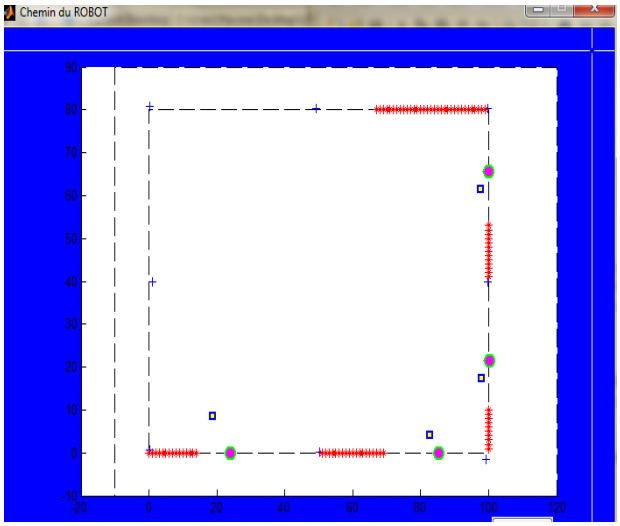

Fig. 11. Trajectory performed by the Robuter using the LF.

For the calculation of the positions after the deviation, already knowing the displacement between the robot and the obstacle:

```
figure;
u=2;
v=3;
d=1;
angle=pi/4;
```

The calculation equation for the new coordinates is as follows:

```
x=u-d/sqrt(1+(tan(angle))^2)
y=v+d/sqrt(1+1/(tan(angle))^2)
```

le calcul de l'angle de l'orientation du robot :

```
function a = AngleWrap(a)
 if(a>2*pi)
   a=a-2*pi;
elseif(a<-2*pi)
   a = a+2*pi;
end;
```

The use of fuzzy logic is done on the fuzzy library:

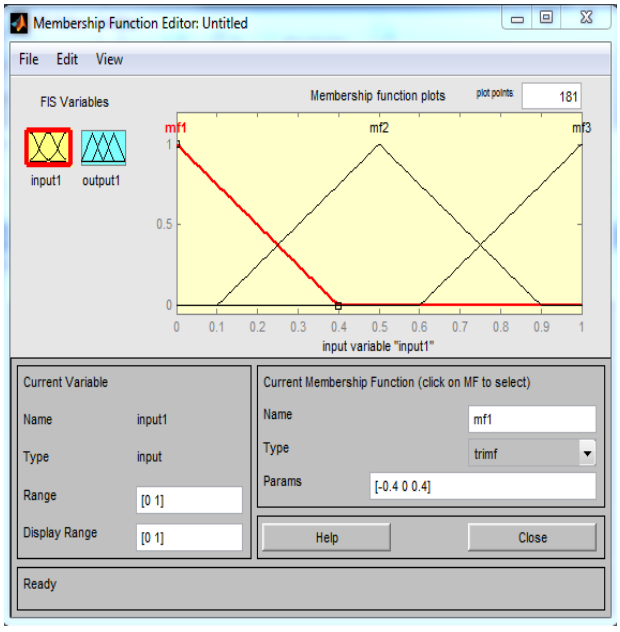

Fig. 12. Bibliotheque Fuzzy

We determine the interval of inputs and output.

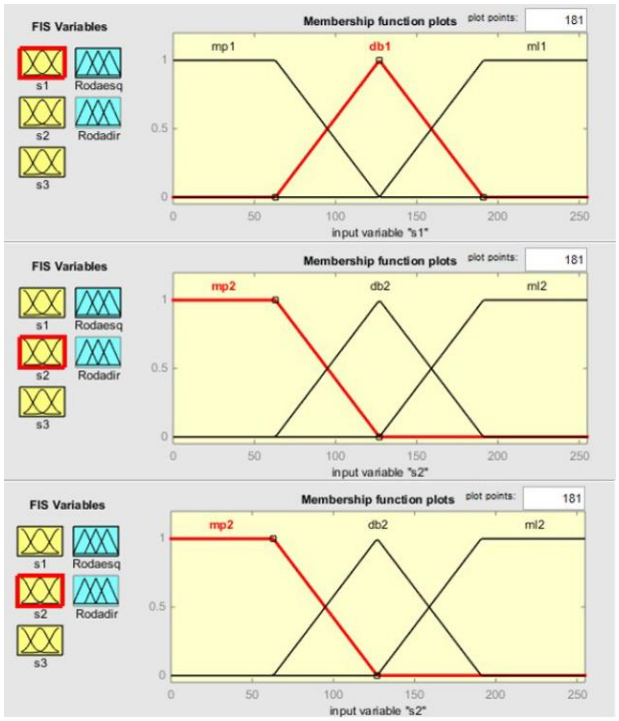

Fig. 13. Input and output LF of biblio fuzzy

6 Conclusion

This article presents an approach of Fuzzy Logic for the navigation control of the mobile robot. This method consists of managing obstacle avoidance behavior.

Matlab simulink simulations on the Robot gave results satisfactory which allows endowing the robot with a certain degree of intelligence.

With its autonomy, the robot should help and protect people from clashing with radioactive rays

References

1. G. Bonow and A. Kroll, "Gas leak localization in industrial environments using a TDLAS - based remote gas sensor and autonomous mobile robot with the Tri - Max method *," 2013 IEEE Int. Conf. Robot. Autom. Karlsruhe, pp. 987–992, 2013.
2. M. E. and T. S. J. Rossmann, N. Hempe, "A RealTime Optical Sensor Simulation Framework for Develop- ment and Testing of Industrial and Mobile Robot Applications Simulation of Various Optical Sen- sors Using Graphics Hardware," Robot. 2012; 7th Ger. Conf. Robot. Munich, Ger., pp. 1–6, 2012.
3. N. Shalal, T. Low, C. Mccarthy, and N. Hancock, "Orchard mapping and mobile robot localisation using on-board camera and laser scanner data fusion – Part B**:** Mapping and Localisation," Comput. Electron. Agric., vol. 119, pp. 267–278, 2015.
4. E. Wang, H. Kang, C. Hyun, and M. Park, "Robust Backstepping Control Based on a Lyapunov Redesign for Skid-Steered Wheeled Mobile Robots

Regular Paper," Int. J. Adv. Robot. Syst., vol. 10, 2013.
5. K. Zhou, G. Ebenhofer, C. Eitzinger, U. Zimmermann, C. Walter, P. Luis, and M. A. Fern, "Mobile Manipulator Is Coming To Aerospace Manufacturing Industry," 2014 IEEE Int. Symp. Robot. Sensors Environ. Proceedings, Timisoara, pp. 94–99, 2014.
6. M. Wang and J. N. K. Liu, "FUZZY LOGIC BASED ROBOT PATH PLANNING IN UNKNOWN ENVIRONMENT," vol. 1, no. August, pp. 18–21, 2005.
7. W. Gueaieb, S. Member, S. Miah, and S. Member, "An Intelligent Mobile Robot Navigation Technique Using RFID Technology," vol. 57, no. 9, pp. 1908–1917, 2008.
8. A. Jayasiri, G. K. I. Mann, and R. G. Gosine, "Behavior Coordination of Mobile Robotics Using Supervisory Control of Fuzzy Discrete Event Systems," vol. 41, no. 5, pp. 1224–1238, 2011.
9. H. Belaidi, H. Bentarzi, and M. Belaidi, "Implementation of a Mobile Robot Platform Navigating in Dynamic Environment," CMME 2016, MATEC Web of Conferences 95, 08004 2017.
10. L. Li, C. Lin, M. Huang, S. Kuo, and Y. Chen, "Mobile robot navigation control using recurrent fuzzy Robot by Applying Fuzzy Approach on Sonar Sensors," vol. 6, no. 3, pp. 36–44, 2010.
11. N. K. A. A.- Sahib and A. R. Jasim, "Guiding Mobile Sensors," vol. 6, no. 3, pp. 36–44, 2010.
12. L. Li, C. Lin, M. Huang, S. Kuo, and Y. Chen, "Mobile robot navigation control using recurrent fuzzy cerebellar model articulation controller based on improved dynamic artificial bee colony," vol. 8, no. 11, pp. 1–10, 2016 cerebellar model articulation controller based on improved dynamic artificial.