Development of Efficient Obstacle Avoidance and Line Following Mobile Robot with the Integration of Fuzzy Logic System in Static and Dynamic Environments

Marwah M. Almasri, Khaled M. Elleithy, Abrar M. Alajlan
Computer Science and Engineering Department
University of Bridgeport
Bridgeport, CT
maalmasr@my.bridgeport.edu, elleithy@bridgeport.edu, aalajlan@my.bridgeport.edu

Abstract— **Autonomous mobile robots have been widely used in many researches and applications. Mobile robots can be programmed to do specific tasks such as collision avoidance and trajectory planning. Various types of sensors and actuators are mounted on these robots for sensing the surrounding environment and making decisions accordingly. In this paper, we develop collision avoidance and line following techniques for mobile robot navigation in static and dynamic environments with the integration of fuzzy logic fusion. For the collision avoidance technique, eight proximity sensors are used to detect static and dynamic obstacles. The proposed fuzzy logic for the line following mechanism consists of one input and two outputs. The input is the difference between right and left ground sensors values whereas the outputs are the left and right speeds to steer the mobile robot. Several membership functions and fuzzy rules are designed. The proposed method has been successfully tested in Webots Pro simulator and in real time experiment.**

Keywords- *Fuzzy Logic controller, collision avoidance, line following, mobile robots, data fusion, dynamic obstacles.*

I. Introduction

Autonomous mobile robots have been widely used in many researches and applications. Mobile robots can be programmed to do specific tasks such as collision avoidance and trajectory planning. Various types of sensors and actuators are mounted on these robots for sensing the surrounding environment and making decisions accordingly. However, these sensors are in different types and thus have different accuracies and features. Some of them might be robust against external environmental factors such as high temperature and pressure while others might be affected which result in reducing the efficiency and reliability of these sensors. Consequently, multiple homogeneous or heterogeneous sensors are needed in designing an efficient autonomous mobile robotic system. Data fusion is the process of combining and aggregating different types of sources and sensors to get an improved result that is more significant and more reliable [1]. Data fusion can be complementary, redundant, or cooperative. Complementary data fusion means fusing different sources to get a more complete output. Redundant data fusion is fusing the same type of inputs to increase the confidence and accuracy of the fused data. Finally, cooperative data fusion is fusing multiple data to gain a new output that is more complex compared with the original data [1].

In addition, collision avoidance and path planning are the most essential tasks needed when designing a mobile robot navigational system. Therefore, integrating the data fusion will increase the reliability and accuracy of the system. Collision avoidance concerns with preventing the mobile robot from colliding with any obstacles or objects encounter on its way while line following mostly concerns with following the robot a black line or a predefined path. Both techniques are widely used in many mobile robot applications. Obstacles in any environment can be static with fixed position or dynamic with unfixed position. Dealing with dynamic obstacles is a more complex task.

Therefore, in this paper we design collision avoidance and line following mechanisms for a mobile robot in static and dynamic environments with the integration of data fusion to improve the efficiency and the robustness of the navigation system. Fuzzy logic controller for line following is developed and designed with an input and two outputs, multiple membership functions, and a number of fuzzy rules.

The paper is organized as follows. Section II presents the related work. Section III discusses the proposed method in details. It also explains the collision avoidance mechanism and the fuzzy logic controller design for the line following mechanism. Section VI demonstrates the simulation and real time experimental setup for the mobile robot navigation using the proposed technique. Performance evaluation results and discussion are presented in section V. Finally, section VI concludes the paper based on findings.

II. Related Work

Various path planning and collision avoidance algorithms and techniques have been developed in the literature. A number of these techniques have been implemented with the integration of the data fusion system.

In [2], authors developed a path planner mechanism which is composed of Cellular Automata (CA) and Ant Colony Optimization (ACO) methods for multiple mobile robots. The proposed method is applicable in a dynamic environment and in real time applications. It also reduces the complexity and thus improves performance. A fuzzy-logic-assisted interacting multiple model (FLAIMM) technique is introduced in [3]. Authors designed two extended Kalman filters (EKF) to solve the problem of the dynamics of mobile robots whereas an adaptive neurofuzzy inference system (ANFIS) was used for predicting the slip. The proposed approach has been tested using a robot navigating in indoor environment which proves that the proposed work enhances the location accuracy. In [4], authors used the Adaptive Neuro Fuzzy Inference System (ANFIS) Controller for mobile robot navigation in a cluttered environment with different conditions and various types of objects. The proposed ANFIS has successfully tested in simulation and in real experiment. The mobile robot has avoided obstacles and reached to the desired goal in less time.

In addition, in [5], authors implemented a stereo vision system using SLAM along with Neurofuzzy decision framework for mobile robot navigation system. The proposed method has successfully created navigation maps and tested on a realistic robotic platform. In [6], a Modified Fuzzy Associative Memory is introduced to reduce the rule base used in mobile robot navigation. The framework is tested in simulation and real time experiments and further compared against other techniques.

Authors in [7] have used a genetic algorithm and a fuzzy logic system for a mobile robot path planning. The genetic algorithm was used to compute the optimal path to destination especially in a dynamic environment due to its effective adaptability to environment changes. The simulation results validate the high efficiency of using the proposed mechanism in challenging dynamic environments. In [8], authors combined spiking neural networks technique for steering the mobile robot in a known environment with a fuzzy logic technique for navigating the mobile robot in an unknown environment. This hybrid approach reduces time and path taken to reach the target which result in an enhanced mobile robot navigation performance.

In [9], a path tracking algorithm for both straight and curved paths is developed. The proposed algorithm is based on fuzzy logic where two controllers were designed. One is for the steering angle of the mobile robot, and the other for controlling the linear speed of the mobile robot. The vision system created by the camera mounted on the robot is the input to the fuzzy system. Multiple experiments are performed to test the effectiveness of the proposed technique of path tracking in straight and curved cases. The fuzzy steering controller and the fuzzy velocity controller are successfully adjusted the mobile robot's speed in all experiments conducted.

III. Proposed Method

This section discusses the proposed method in details. The goal is to navigate the mobile robot in static and dynamic environments without colliding with any obstacles. It should also follow a black line during its movement. For efficient mobile robot navigation, we proposed a collision free and line follower robot based on fuzzy logic controller. The robot used in our work is the E-puck robot which has a large number of sensors for monitoring the surrounded environment.

A. E-puck Mobile Robot

E-puck robot was developed at the EPFL. It is simple, flexible, and user friendly mobile robot. It has 7.4 cm in diameter and 4.5 cm hight with a weight of 150 g [10]. The hardware is composed of a microcontroller and a wide set of sensors and actuators such as: infrared (IR) proximity sensors, a 3D accelerometer, RGB camera, three microphones, two stepper motors, eight LEDs, and a speaker. In addition, the E-puck robot can be connected to additional sensors such as ground sensors as physical extensions [11].

In our work, we are mainly focused on the eight proximity sensors which are infrared sensors, and the three ground sensors. Fig. 1 shows the E-puck robot mounted with the proximity and ground sensors. As shown in Fig.1, all eight proximity sensors donated as PS0, PS1, PS2, PS3, PS4, PS5, PS6, and PS7 are located around the robot at different directions. These sensors are infrared sensors which measure the amount of light reflected of any obstacle presence. The range values of these sensors are [0 2000], where an obstacle is detected if one or more of these sensors have returned a value of 1000 or more. Otherwise, there is no obstacle around. In addition, the ground sensors donated as GS0, GS1, GS2 are infrared sensors that are located underneath the E-puck to detect the line for the line following approach.

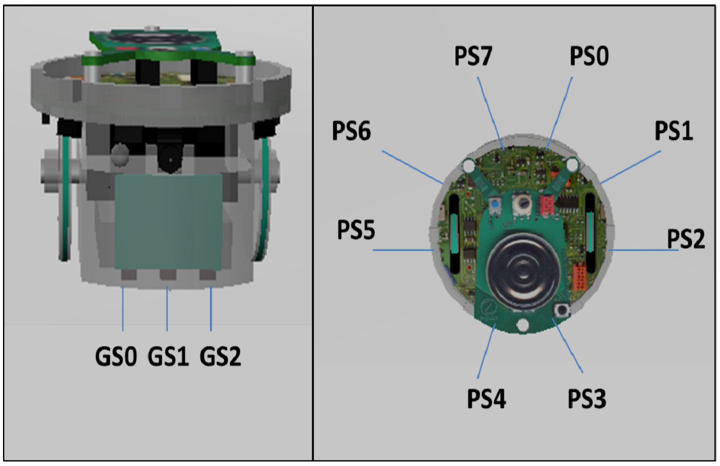


Fig.1: The E-puck robot

B. Webots Simulator

Webots simulator is a well-known mobile robot simulator that is widely used in many researches. It is used to model, design, and program environments and mobile robots. Webots is composed of four main windows. The scene tree lists all obstacles in different shapes, sizes, and colors, and also a large number of simulated sensors and actuators. The 3D window displays the actual simulation. The console prints outputs. Finally, the controller is used for editing and writing the source code. The controller can be written in different programming languages such as C, C++, Java, Python and Matlab [10].

C. Fuzzy Logic Controller Design for Line Following Mechanism

Fuzzy logic system has been used in many applications for steering the mobile robot. It is widely used due to its low computational load and its simplicity. The fuzzy logic system has three main parts: fuzzification, inference engine and defuzzification. The fuzzification process is responsible for producing crisp fuzzy values. The inference engine is responsible for computing the output variables based on defined fuzzy rules in terms of If, Then representation. Lastly, the defuzzification process is used to convert the fuzzy output to crisp values for the objective of steering the mobile robot [12].

For the line following approach in our work, we used the three ground sensors located on the E-puck robot facing to the ground. First, the difference (Δ) between the right (GS2) and left (GS0) ground sensors is calculated. The delta value (Δ) is the input to the fuzzy logic controller. Two outputs are generated which are the left motor speed (LS), and the right motor speed (RS) which adjust the robot speed to follow the line. If both speeds are equal, the robot will move forward; however, if the left speed value is lower than the right speed, then the robot will move to the left direction and vice versa. The pseudo code for the line following approach is represented in Fig.2.

1. **while (true)**
2. Enable and read 3 ground sensors values (GS0-GS2)
3. Calculate Δ= GS2- GS0.
4. Fyzzify the input Δ
5. Apply fuzzy rules
6. Defuzzify the output
7. Set output values to the robot motor speeds LS and RS
8. **end while**

Fig.2: Line following pseudo code

The input variable delta (Δ) is divided into four membership functions as follows: NB: negative big, NS: negative small, PS: positive small, and PB: positive big. All four membership functions are triangular-shaped membership functions. Fig.3 shows the membership function for the input delta (Δ).

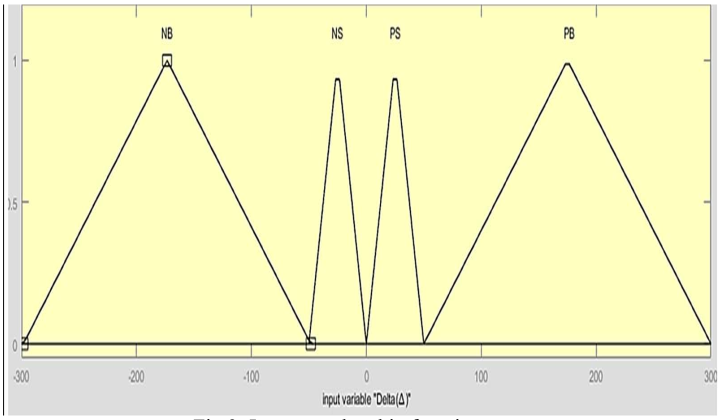


Fig.3: Input membership functions

In addition, the output variables LS and RS are divided into four membership functions as follows: Very-Low: which means very low speed, Low: corresponds to low speed, High: means high speed, and Very-High: which means a very high speed. All these membership functions are trapezoidal-shaped membership functions. The centroid method is used for the defuzzification process. Fig.4 shows the output membership functions.

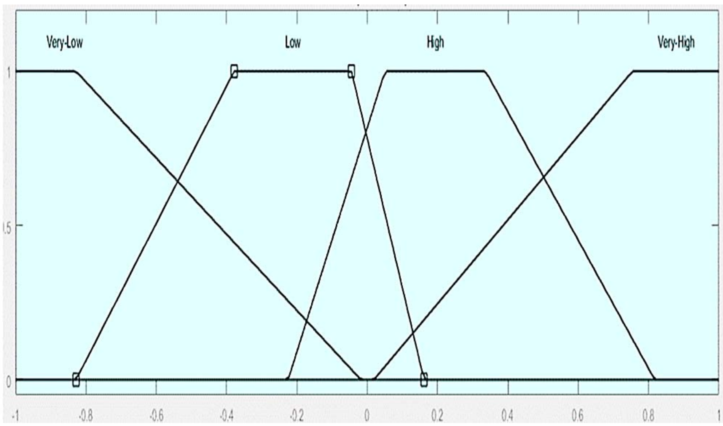


Fig. 4: Output membership functions

After defining the membership functions for the input and the outputs, four IF-THEN fuzzy rules are developed for the line following mechanism as follows:

- IF (Delta (Δ) is NB) THEN (LS is Very-High) (RS is Very-Low)
- IF (Delta (Δ) is NS) THEN (LS is High) (RS is Low)
- IF (Delta (Δ) is PS) THEN (LS is Low) (RS is High)
- IF (Delta (Δ) is PB) THEN (LS is Very-Low) (RS is Very-High)

Based on these rules, the robot will move left or right to follow the line.

D. Collision Avoidance Mechanism

Eight proximity sensors are used for obstacle detection. These sensors have a range from 0 to 2000 whereas 1000 or more means there is a close obstacle and the robot needs to adjust its speed to spin around the obstacle. After that, the robot should continue following the line if there are no obstacles. The threshold is set to 1000. Fig.5 represents the pseudo code for the collision avoidance approach.

```
1. while (true)
2. Enable and read 8 proximity sensors values
(PS0-PS7)
3. If (Sensors values > Threshold) // Obstacle is
detected
        - Adjust left and right motor speeds
        - Avoid obstacle and go around it
        - Return to the line
4. Else
        Move forward and do line following
5. end while
```

Fig.5: Collision avoidance pseudo code

IV. Simulation and real time Experimental Setup

Webots Pro simulator is used to test and simulate the proposed line following and collision avoidance mechanisms. In addition, the E-puck robot is used which is mounted with eight proximity sensors for the collision avoidance technique and three ground sensors for the line following technique. In our simulation, we created an environment that consists of multiple mobile robots and multiple obstacles in different sizes, colors, and shapes. For the purpose of the line following technique, a black line is drawn on a white surface. Fig.6 shows the simulation setup.

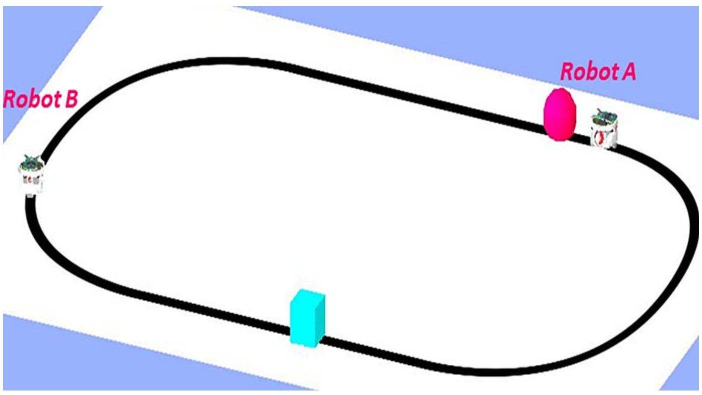


Fig.6: Simulation setup

Furthermore, Fig.7 demonstrates the simulation runs at different times. The robot senses the surrounding environment with its sensors. It starts by following the line until it finds an obstacle on its way. As shown in Fig.7 (i, ii, and iii), Robot A detects a static obstacle and thus spins around it to avoid collision. Then, it returns to the line. In Fig.7 (iv, v, and vi), Robot A and Robot B detect each other as dynamic obstacles. Both avoid collisions by going to opposite directions and then continue following the line. In addition, Robot B detects another static obstacle and tries to avoid it by turning around the obstacle as shown in Fig. 7(vii and viii). Finally, Robot A detects a second static obstacle while Robot B follows the line due to a non-obstacle presence as in Fig.7 (ix). The real time experiment is displayed in Fig. 8. Both robots A and B detect each other as dynamic obstacles and avoid each other by adjusting their motor speeds and finally follow the line. The mobile robots follow the line by calculating the delta value (Δ) which is further fed to the fuzzy logic controller to adjust the left and right motor speeds accordingly.

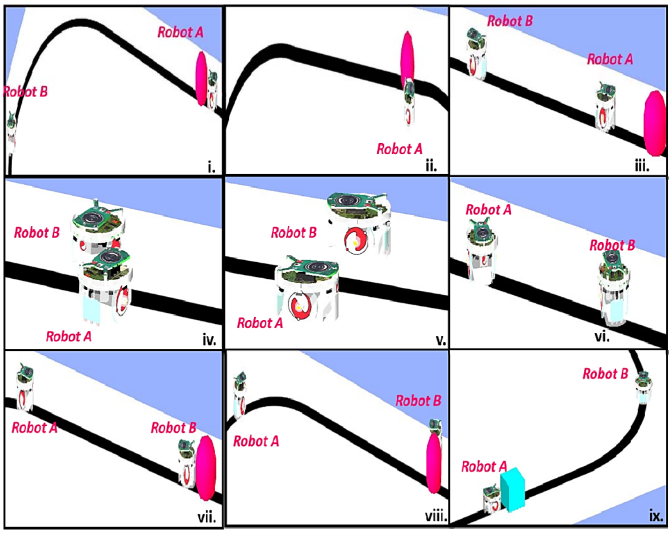


Fig.7: The simulation runs at different times.

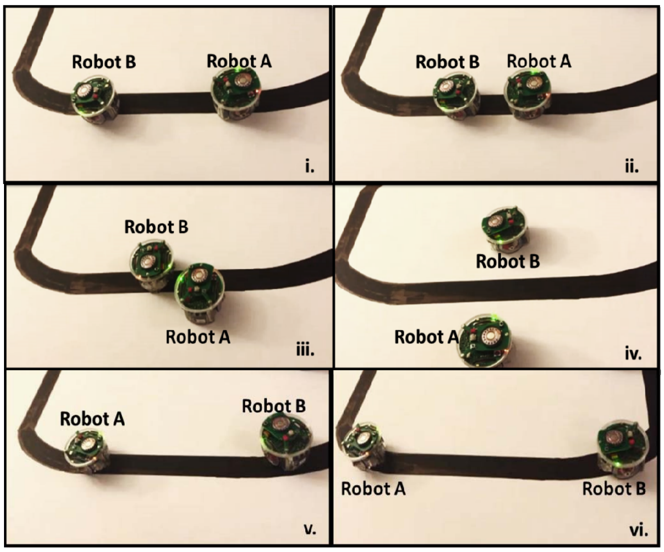
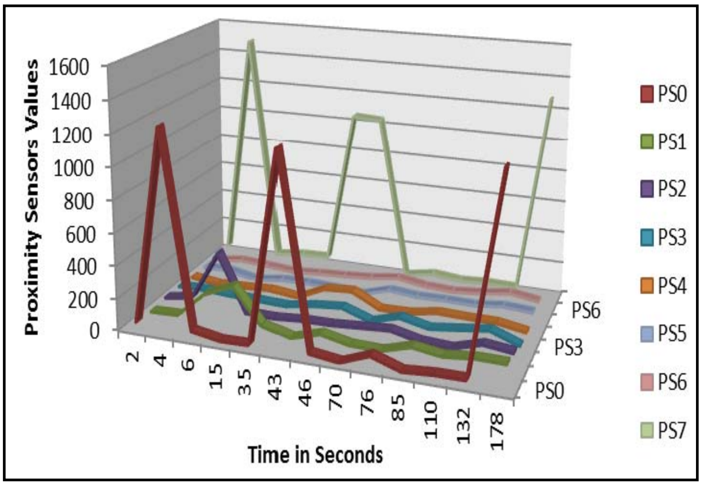


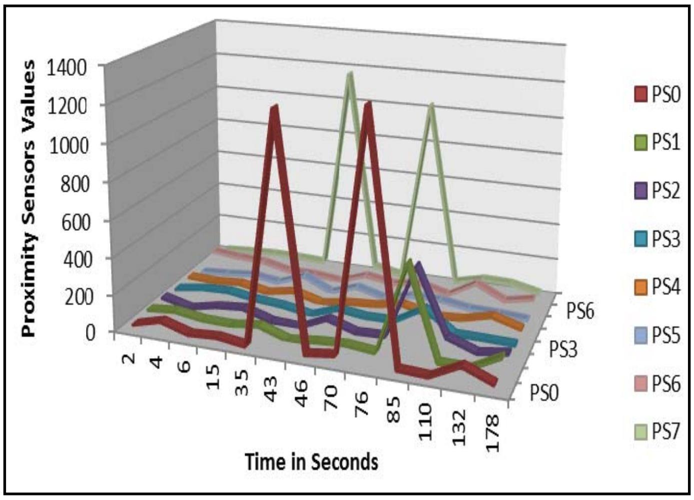
Fig.8: The real time experiment.

V. Performance Evaluation results and Discussion

The main contribution of this work is to steer the mobile robot while following the line. The mobile robot also avoids colliding with any obstacles on its path. The fuzzy logic controller was designed for the line following mechanism where it takes the delta value (Δ) as an input and provides the left and right robot motor speeds based on the defined fuzzy rules. The proximity sensors are responsible for the collision avoidance technique. If one of these sensors reaches the threshold value, obstacle is close to the robot and thus it needs to adjust its speed and direction. Fig. 9 shows the proximity sensors readings for robots A and B. As shown in Fig.9 (a), the front proximity sensors (PS0 & PS7) for Robot A have values higher than the threshold (1000) at times 4s, 43s, and 178s. This means that the robot has detected an obstacle. At time 4s, Robot A detects the first obstacle and at time 178s detects the second obstacle. At time 43s, Robot A has detected the other mobile robot as a dynamic obstacle.



a. Proximity sensors readings for Robot A



b. Proximity sensors readings for Robot B

Fig.9. Proximity sensors readings

Similarly, the front proximity sensors (PS0 & PS7) for Robot B have values higher than the threshold at times 43s and 76s, respectively. At time 43s, Robot B has detected the other mobile robot and considered it as a dynamic obstacle. At time 76s, it detects another obstacle on its way.

In addition, Fig.10 demonstrates the left and right speeds during the simulation. As shown in Fig.10, Robot A has negative values for the left motor speed and positive values for the right motor speeds at times 6s and 46s which indicates that the robot has turned left to avoid collisions with static and dynamic obstacles right after the detection of these obstacles. At time 15s, Robot A has a positive value for the left speed and negative value for the right speed which means that the robot has turned right to return to the line. Also, Robot B has turned left at time 46s after the detection of the dynamic obstacle. At time 85s, Robot B has turned right to return to the line.

Fig.11 and Fig.12 show the ground sensors reading and the delta values (Δ) which corresponds to the difference between the right (GS2) and left (GS0) ground sensors at different simulation times, respectively.

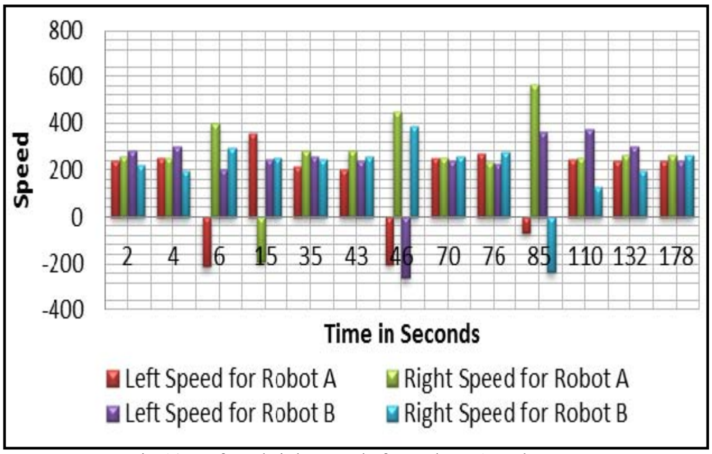


Fig.10: Left and right speeds for Robots A and B.

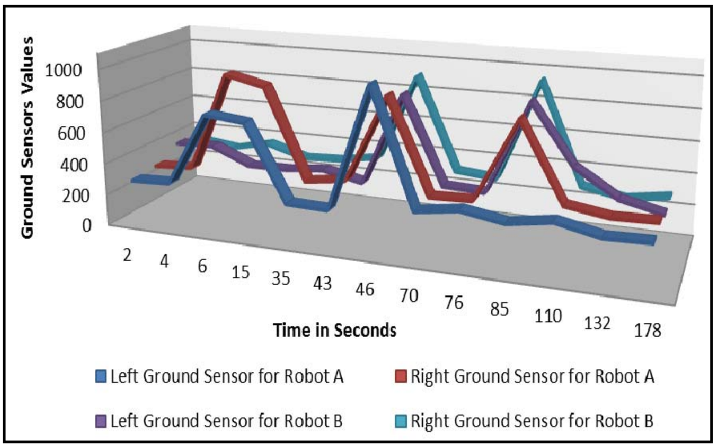


Fig.11: Left and right ground sensors readings for Robots A and B

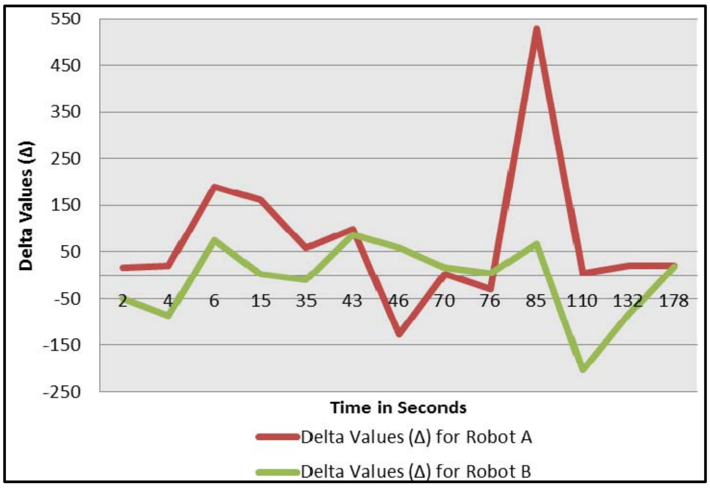


Fig.12: Delta values (Δ) for Robots A and B

If delta is a negative value, this means that the robot has shifted to the left outside the line. Therefore, the robot needs to adjust its speed to move right to follow the line. According to the designed fuzzy logic controller and the fuzzy rules, when delta is NB, the left motor will be set to a very high speed and the right motor will be set to a very low speed so the robot will move hardly to the right. As shown in Fig.12, Robot B has a big negative delta at time 110s. As a result, Robot B moves to the right and set LS to a very high value and RS to a very low value to keep following the line as shown in Fig.10. Similarly, if delta is NS, LS will be high and RS will be low so the robot will move softly to the right. To illustrate, at time 76s, Robot A has small negative value so the robot adjusts its speed to move softly to the right as shown in Fig.12 and Fig.10.

On the other hand, if delta is a positive value, this means that the robot has shifted to the right outside the line. Therefore, the robot needs to adjust its speed to move left to follow the line. According to the designed fuzzy logic controller and the fuzzy rules, when delta is PS, the left motor will be set to a low speed and the right motor will be set to a high speed so the robot will move to the left which is the case for Robot A at times 2s, 35s, 132s, and 178s where it moves softly to the left to follow the line as shown in Fig.12 and Fig.10. Finally, when delta is PB, LS will be set to a very low speed and RS will be set to a very high speed so the robot can move hardly to the left to follow the line. To illustrate, Robot A at time 85s where delta is a big positive value, the robot moves hardly to the left as shown in Fig.12 and Fig.10. There are other cases where fuzzy rules are not applicable as the robot is in the collision avoidance control and its speed is adjusted accordingly. As a result, the collision avoidance has a higher priority than the line following to avoid colliding with any obstacles along the path.

VI. CONCLUSIONS

This paper developed the collision avoidance and line following mechanisms for mobile robot navigation. The E-puck robot is used in this work which is equipped with eight proximity sensors for the collision avoidance approach, and three ground sensors for the line following approach. A fuzzy logic controller was designed with one input and two outputs. The input is the difference between the right and left ground sensors values and the outputs are the left and right motor speeds. Membership functions of the input and the outputs are developed. The fuzzy logic controller is based on four fuzzy rules. The simulation and real time experiments validate the effectiveness and the robustness of the proposed method in static and dynamic environments.

REFERENCES

[1] F. Castanedo, "A Review of Data Fusion Techniques", *The Scientific World Journal*, vol. 2013, pp. 1-19, 2013.

[2] K. Ioannidis, G. Sirakoulis and I. Andreadis, "Cellular ants: A method to create collision free trajectories for a cooperative robot team", *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 113-127, 2011.

[3] H. Lee, J. Jung, K. Choi, J. Park and H. Myung, "Fuzzy-logic-assisted interacting multiple model (FLAIMM) for mobile robot localization", *Robotics and Autonomous Systems*, vol. 60, no. 12, pp. 1592-1606, 2012.

[4] J. Pothal and D. Parhi, "Navigation of multiple mobile robots in a highly clutter terrains using adaptive neuro-fuzzy inference system", *Robotics and Autonomous Systems*, vol. 72, pp. 48-58, 2015.

[5] K. Al-Mutib, E. Mattar and M. Alsulaiman, "Implementation of Fuzzy Decision Based Mobile Robot Navigation Using Stereo Vision", *Procedia Computer Science*, vol. 62, pp. 143-150, 2015.

[6] S. Parasuraman, "Sensor Fusion for Mobile Robot Navigation: Fuzzy Associative Memory", *Procedia Engineering*, vol. 41, pp. 251-256, 2012.

[7] S. Farshchi, S. NezhadHoseini and F. Mohammadi, "A Novel Implementation of G-Fuzzy Logic Controller Algorithm on Mobile Robot Motion Planning Problem", *CIS*, vol. 4, no. 2, 2011.

[8] Z. LAOUICI, M. MAMI and M. KHELFI, "Hybrid Method for the Navigation of Mobile Robot Using Fuzzy Logic and Spiking Neural Networks", *IJISA*, vol. 6, no. 12, pp. 1-9, 2014.

[9] E. Vans, G. Vachkov and A. Sharma, "Vision based autonomous path tracking of a mobile robot using fuzzy logic," in *Computer Science and Engineering (APWC on CSE),* Asia-Pacific World Congress on , pp.1-8, 2014.

[10] Cyberbotics.com, "Webots: robot simulator", 2016. [Online]. Available: https://www.cyberbotics.com/. [Accessed: 15- Feb-2016].

[11] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano and A. Martinoli, "The e-puck, a Robot Designed for Education in Engineering", *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, vol.1, no. 1, pp. 59-65, 2009.

[12] B. Shah, F. Iqbal, A. Abbas and K. Kim, "Fuzzy Logic-Based Guaranteed Lifetime Protocol for Real-Time Wireless Sensor Networks", *Sensors*, vol. 15, no. 8, pp. 20373-20391, 2015.