Design of Collision Prevention based on Wall Following Mechanism on Multi-Robot System

Joao Amaral de Fatima Pereira, Agung Nugroho Jati, Randy Erfa Saputra
School of Electrical Engineering, Telkom University
Indonesia
carlosdomingas21@gmail.com

***Abstract*— Wall following mechanism is mostly used for finding a way out in the maze or room on the single robot cases, but in this paper, it will be demonstrated a multi-robot system by applying collision avoidance based on wall following mechanism to solve an unknown maze. Fuzzy logic will be implemented on right wall following algorithm to avoid the collision while the robots are in a maze, and at the same time robots will recognize which are the wall of maze, junctions, and other robot. Based on the results, shown that the robots could recognize the junctions, wall and other robots. Based on the result of all testing, robots might solve the complicated maze for 2 until 5 minutes, with 86% of solution without any collision.**

***Index Terms*— Collision Prevention, Fuzzy logic; Multi-Robot; Maze Solver; Tremaux's Algorithm.**

I. INTRODUCTION

Collision avoidance is a fundamental problem in robotics. Problems are generally defined in the context of autonomous mobile robots which navigate in environments with barriers and/or other mobile entities, where robots use continuous sensing and action cycles. Solving a maze with more than 2 robots, such a serious problem that we need to concern because collision avoidance may occur during the robots are running inside it. Robots have a goal it is to reach the endpoint, then probably collision will happen if the robots face the wall of a maze or with another robot. In this problem, we found out that ebbed an algorithm for each robot is the key to avoiding those problems. And there are a lot of algorithms have used to solve the maze easily, like flood fill[1], tremaux's algorithm, wall followers[12], random mouse[2] algorithm, dead-end filling maze-routing algorithm and etc. but most of those algorithms have already implemented in mobile robot, and it worked very well. Here, we will use one of those algorithms for solving a maze faster and no collisions happened while the robots are searching for a goal.

As we all know maze is a common unknown arena, needs to find its short way to solve the maze. And in this paper will tell you about using one of the mazes solving algorithm, take a good decision when the robots facing the arena and find the short path, during the seeking of the shortest path, there are no collisions occurred between robots and arena or with another Robots.

A single robot may use those algorithms to solve a maze, instead of it multi-robot also has an occasion to use those algorithms, but the differences in time to reach the goal or end-point[10]. A single robot solves a maze fast, but multi-robot will make it faster with the same algorithm. Regarding to this problem of maze solving, a necessary and applicable algorithm is needed, and we prefer using wall follower algorithm. Wall follower has 2 types, those are Right Wall Follower(RWF) and Left wall follower(LWF) it depends on application of both algorithm, Left Wall follower(LWF) is the most one used, but in this research, we chose Right Wall Follower (RWF).

Basically, ultrasonic and IR sensors are mostly used to apply the wall follower algorithm, right here we are used 4 ultrasonic sensors, as it demonstrated using Right Wall Follower means ultrasonic sensors are located in each side instead of back side[3][5]. An ultrasonic sensor will be located in the front side, 2 ultrasonic sensors in the right side, each ultrasonic are located in front and back of right sides of a robot, and the other one is in the left sides of the robot. The combination of that ultrasonic sensors is able to respond as quickest when the robots are in the junction, corridor or in the dead-end.

To apply the right wall follower algorithm, actually fuzzy logic is allowed to work on it, take the range on a measurement of each ultrasonic sensors are keys to implementing the fuzzy logic in right wall follower.

The algorithm which will be used in this paper is right wall follower algorithm, and inside of the RWF actually, there is the implementation of fuzzy logic applied. To use those algorithms, we need to embed it in sensors as everyone one knows, it would be possible using the camera, IR sensors, ultrasonic sensors and others. After all of this, because we're implemented these algorithms on Robots using Arduino, we chose the ultrasonic sensors as a parameter for writing and embedding the algorithms.

This paper is organized as follows: Section I will show the introduction, backgrounds, and problems about this research. Then, section II discusses about other related researches used as references in this research. In the section III, shown the conducted method and configuration of this research while section IV shown the results and analysis. The last section is a conclusion of the research.

II. RELATED WORKS

There are many algorithms can be used to solve the maze such as random mouse, wall follower, and flood fill

algorithms[2][13]. The wall follower algorithm mostly is used when the arena or place the robots face are unknown and only have one purpose to reach the end point. The end point basically is set by giving a sign. On the other hand, the flood fill algorithm is commonly used when the position of end point is recognized by the robot through some identification and from that sign robot must be find the shortest path[2].

Flood Fill Algorithm is one of the most efficient algorithms to be used[1][2]. This algorithm may solve complex and difficult maze without having any troubles/collisions by receiving the value of all cells in a maze, which those values shows the steps from any cell to reach the end point. At the same time, mostly this algorithm is used by count the every passed grid and the next grid, means that solving the maze by recording the number of grid.

Pulse Width Modulation "PWM" is a simple method by controlling analog values through the digital signal by changing or modulating the pulse width. This is a well applicable algorithm using electricity[1][14]. By tuning an analog device with a pulse wave changes between HIGH with "5V" and LOW with "0V" at a fast rate the device will behave as it getting a steady voltage somewhere between 0V and 5V. Figure-1 shows an example of a PWM with a duty cycle of 0% - 100&, which gives a voltage value of for 0V until 5V. The term of duty cycle explains the amount of time in the period that the pulse is on or HIGH and off or LOW. It will be specified as a percentage of the full period of time . 0% duty cycle means that the PWM wave is off or LOW and 100% means waves is fully on or HIGH[1].

Figure 1: PWM of 0% - 100% duty cycle [1]

Table 1
Right And Left Wall Following Routes

LEFT WALL FOLLOWER ROUTES	RIGHT WALL FOLLOWERS ROUTES
If NO wall at LEFT TURN LEFT Else if NO wall in front FORWARD Else if NO wall at RIGHT TURN RIGHT Else TURN AROUND	If NO wall at RIGH TURN RIGHT Else if NO wall in front FORWARD Else if NO wall at LEFT TURN LEFT Else TURN AROUND

The wall following algorithm is one of the simplest and most used algorithm to solve a maze. The robot will heading any direction to reach the end point by following the right side of wall or it might be left wall[12][4][14]. Robot will recognize any junction by the value of the input sensors, and robot will sense the open wall also by detecting using sensors. And the robot will follow the priority wall. By consider the wall as a main key to reach the end point, the robot doesn't need to find the shortest path to reach the end point. From fig. 2 bellow it shows how wall follower algorithm works for both right and left side, it depends of how we prefer which one is better and useable to solve and unknown maze.

From all those related works we take the Right Wall Follower algorithm as the main part, but inside of the Right Wall Follower algorithm itself, there an existence of PWM that will actually controlling the wheel to explore and shows how fuzzy logic works[3][6][9].

Briefly, Right Wall Follower Algorithm used can be seen in the following pseudocode:

```
While not Reach Goal
        If frontside is open
                If Right1 > 10
                Turn_Right Slowly
                If Right2 > 10
                Turn_Left Slowly
        Else if right1 is open then
                Turn _right
        Else if left side is open then
                Turn_left
        Else
                Turn_around
Loop
```

Pseudocode above shows the decision making of robot to follow the wall, by receiving the data from the input value of 4 ultrasonic sensors where located in every Robot.

III. SYSTEM IMPLEMENTATION

Before we describe for further about algorithms, first we need to know where those mentioned sensors placed in robot, then it would be easier to understand better the algorithm.

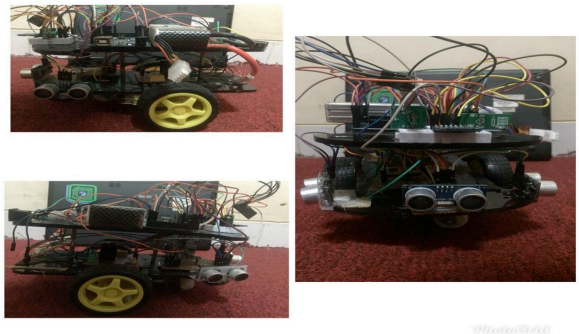


Figure 2: Implementation of Robots

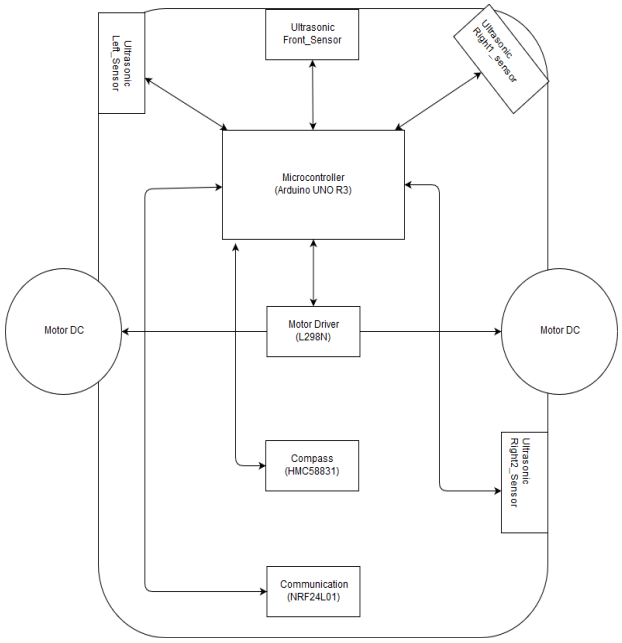


Figure 3: Design Scheme of Robot

Right wall Follower as an algorithm is used to avoid collision between robots and walls[11], but inside of it, we need to add some system to let this RWF algorithm working better. Fuzzy logic will be established, to make the RWF algorithm works and keep making a good decision and knows where the robot is located. The algorithm for robot will be different depends on the location of the robot, it means that when the robots are in corridor or junction will certainly check all the time. In figure 4 below shows us how robots taking the decision when they are in the corridor.

For the RWF Algorithm will combine with PWM controlling. But those algorithms are combined using the Fuzzy logic. Fig 4 demonstrates that the RWF is working with the PWM and controlled by the inputs of data decided by fuzzy logic[6].

Flowchart bellow will show, how we implement the Right Wall Follower algorithm on Multi-Robot. And this flowchart also shows the certain value of ultrasonic to make any decision for robot such follow the wall, turn left, turn right or turn around. The purposes of this flowchart is to find the end point by implementing the Right wall follower algorithm, and at the same time there are no collisions for robots and maze, also with other robots.

Before this algorithm works, it took some decision making to run it, like what sensors will do if the data they have from the inputs, and how it works. Fuzzy logic implementation led the robot to make its own decision to run the designing algorithm[7][12]. As the flow-chart demonstrated that, there are only 3 ultrasonic used to show how RWF work. The table below has shown that the inputs of every three sensors will tell the motors by controlling using PWM. Actually, using PWM to controlling the motor such a good implementation, because PWM would be worked well if a microcontroller received the data from more than 2 sensors.

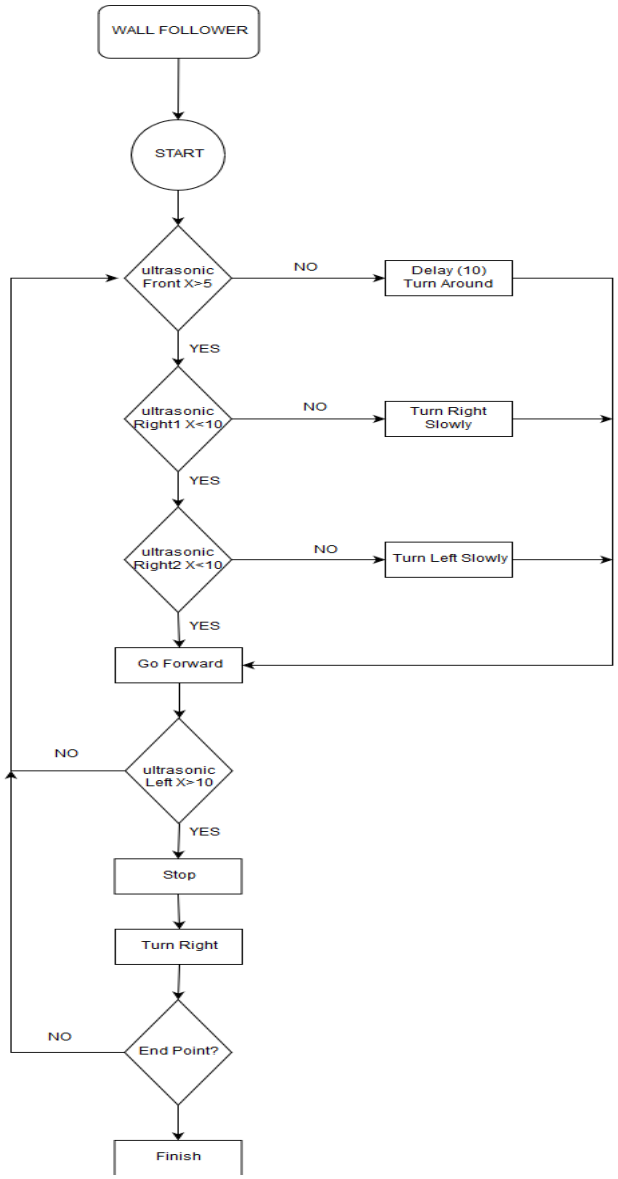


Figure 4: Flow-chart for Right Wall Follower Algorithm for Multi-Robot

In the next table, shown how the turn left slowly and turn right slowly work and the combination of PWM for two motors making a decision, while the Right Wall Follower algorithm is running, but no collision happened.

Table 2
Motor Rotation Speed based on Distance from Wall

Front sensor	Right1 sensor	Right2 sensor	Left motor	Right motor
X > 10	X > 10	X > 10	*PWM=110	*PWM=90
X > 10	X < 10	X > 10	*PWM=60	*PWM=90
X > 10	X > 10	X < 10	*PWM=90	*PWM=60
X > 10	X < 10	X < 10	*PWM=90	*PWM=110
X < 10	X > 10	X > 10	*PWM=0	*PWM=0
X < 10	X < 10	X > 10	*PWM=0	*PWM=0
X < 10	X > 10	X < 10	*PWM=0	*PWM=0
X < 10	X < 10	X < 10	*PWM=0	*PWM=0

Beside of Right Wall Follower algorithm as you see in TABEL II, at the same time there is an existence of another algorithm to recognize the junctions[13]. In this paper describes that the robot will consider another robot as a junction, but the decision making of each robot for maze solving actually different. When the robot is facing the others are different, those differences are the embedded of the delay and PWM of turning around. For the delay, every robot has 10 seconds bigger from another when they are in the junctions[8].

In the table below, also will be demonstrated, when the robot will make a decision to choose the paths. Paths here are mentioned to the heading points of a robot, turn left, turn right, turn around or moving forward.

Table 3
Decision Making For Motors

Direction	Left Sensor	Right Sensor	Front Sensor
TURN LEFT	X > 15	X < 15	X < 10
TURN RIGHT	X < 15	X > 15	X < 10
TURN AROUND	X < 15	X < 15	X < 10

Table 4
Heading Directions Of Motors

Direction	Left Motor	Right Motor
TURN LEFT	Backward	Forward
TURN RIGHT	Forward	Backward
TURN AROUND	Forward	Backward

The table 3 shows that every robot's direction depends on its input of ultrasonic sensors. On the other side, fig 6 shows how both motors work after robot make a decision of the heading direction. And the next table will show you the implementation of PWM for the Direction of the robot.

Table 5
Speed Motors For Heading Directions

Direction	Left Motor		Right Motor	
	Forward	Backward	Forward	Backward
TURN LEFT	PWM=0	PWM=55	PWM=110	PWM=0
TURN RIGHT	PWM=110	PWM=0	PWM=0	PWM=55
TURN AROUND	PWM=110	PWM=0	PWM=0	PWM=55

Robot considers the junction by the inputs data of three ultrasonic sensors located on front side of the robot (Front, left and Right1). When those three ultrasonic detect some several data, they found out the junction. Then, it will automatically choose its own decision. Flow-chart bellow demonstrates how robot knows that he's facing the junction.

Before we end this part of discussing the algorithm, here a picture that shows how the decision making of the robot in maze arena to avoid the collision between the maze and another robot.

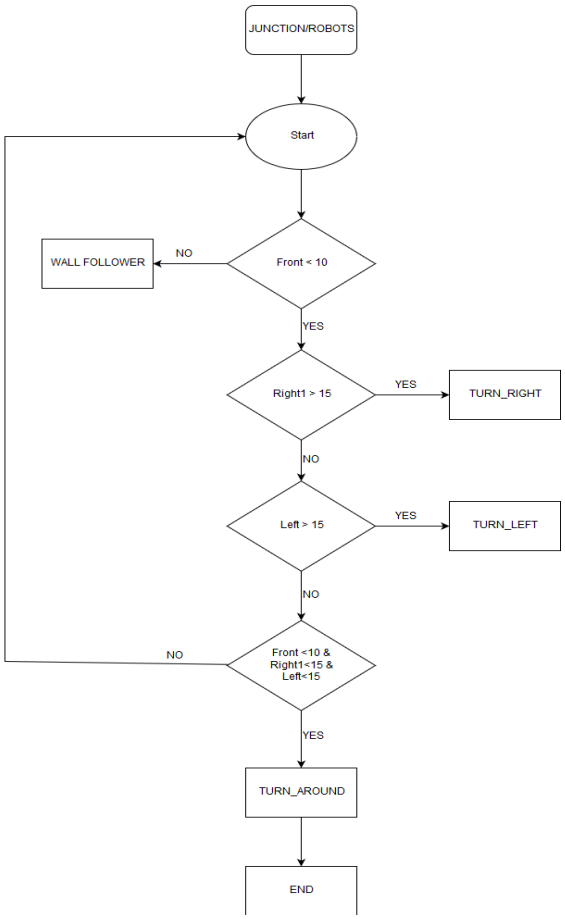


Figure 5: Flow-Chart For Facing The Junction/other Robot

In fig 7, shown that when the robot in the arena, and if the inputs of ultrasonic sensors in the front side detect that it is a junction, and it will automatically turn around, by the right side. The robot will find the endpoint if three ultrasonic sensors in front sides detects in a certain distance with the certain timing.

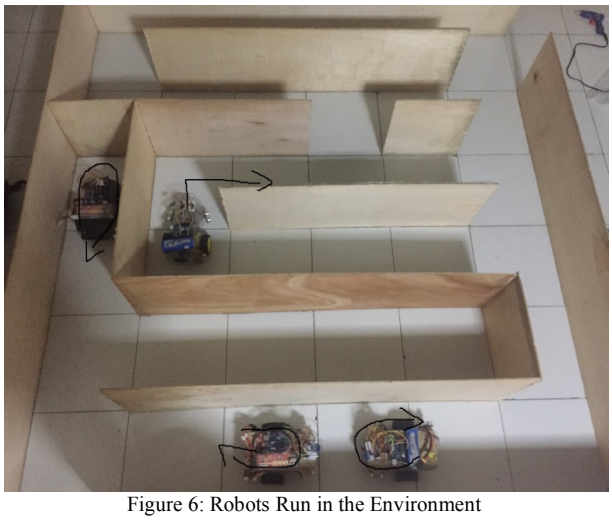
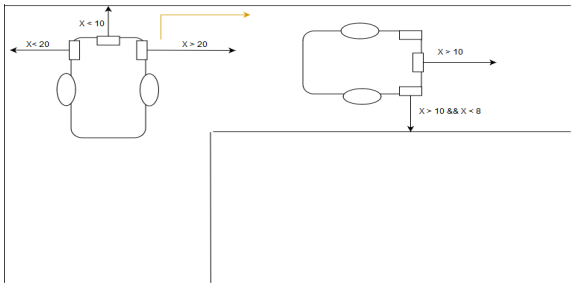
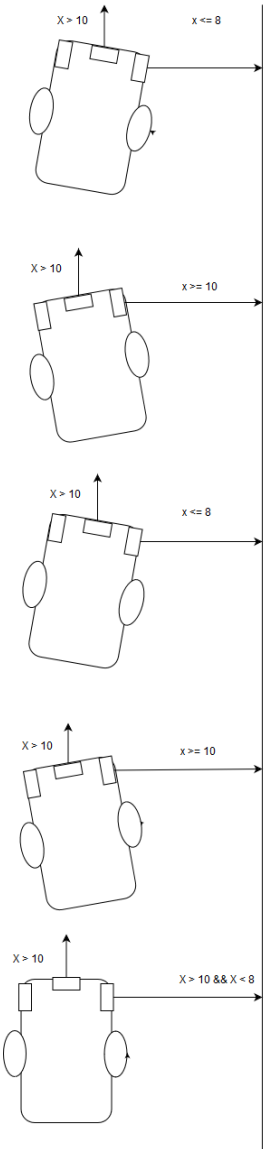


Figure 6: Robots Run in the Environment



(a)

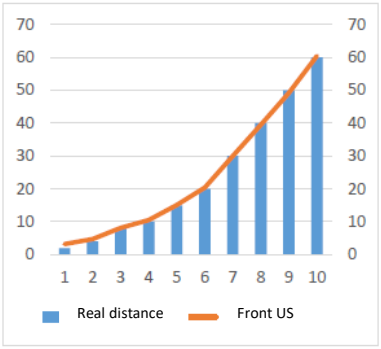


(b)

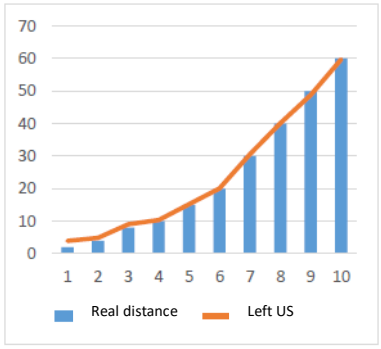
Figure 6: (a) Example of How Robot will Make a Decision, (b) RWF based Robot's Movements

IV. DISCUSSION

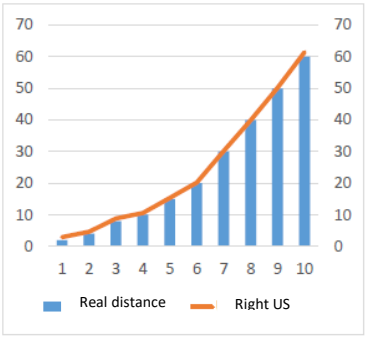
The movements of each robots depend on the location of each sensors. Based on the result of the test, found that the main point in collision prevention on robot which uses ultrasonic sensors, are value of every ultrasonic sensors. It can be seen as follows:



(a)



(b)



(c)

Figure 7: (a) Frontside US Sensor (b) Leftside US Sensor (c) Rightside US Sensor

Based on test shown on the graph, there are error values arround 5% between distance captured by sensor and real distance. It was caused by different material used as wall. Furthermore, it happened because of signal reflection. Different materials influence the spped of signal reflection received by robot, so that they defined different value of distance eventough it's similar in real condition.

The condition above, can lead to collision of robots or robot because robot wll make a late and/or wrong decision caused by invalid distance value from sensor. To avoid them, all of sensor should be callibrated before use. Moreover, perfect input value of ultrasonic sensors will be automatically validated, and rate of collision will be decrease significanly.

According to the experiment, robots can avoid the collision between them in arround 86% success rate. It is taken after 10 times experiment in the environment maze. Besides, robots can prevent to face each other by defining their range and orientation in the maze.

In addition, communication between robot to share its location based on their sensor values can prevent more

significant to collide. However, there were delay communication arround 0.75 s can be tolerated. Robot can communicate in range approximately 2.95 m.

Table 6
Result of Test in Decision Making to Avoid Collision

No.	Real Distance	Robot 1	Robot 2	Decision		Result
1	F:9 L:80 R:90	F:9.43 L:88.23 R:95.22	F:9.15 L:81.95 R:95.22	Turn Right	Turn Right	Success
2	F:45 L:78 R:19	F:44.24 L:75.33 R:17.02	F:43.13 L:77.06 R:18.55	Move Forward	Move Forward	Success
3	F:9 L:50 R:87	F:9.57 L:51.63 R:127.98	F:10.01 L:53.81 R:85.05	Turn Right	Turn Right	Success
4	F:10 L:19 R:12	F:10.06 L:19.29 R:12.45	F:14.28 L:20.33 R:14.26	Turn Arround	Move Forward	Success
5	F:40 L:25 R:15	F:40.01 L:25.04 R:15.89	F:39.32 L:24.29 R:15.19	Move Forward	Move Forward	Success
6	F:8 L:12 R:70	F:9.15 L:4.22 R:63.56	F:9.66 L:9.08 R:0	Turn Right	Error	Failed
7	F:30 L:52 R:12	F:31.11 L:49.39 R:12.44	F:32.02 L:51.45 R:15.12	Turn Right	Move Forward	Success
8	F:30 L:10 R:40	F:30.35 L:9.21 R:41.60	F:30.44 L:10.22 R:39.82	Error	Turn Right	Failed
9	F:60 L:70 R:70	F:58.12 L:68.09 R:98.22	F:57.91 L:69.12 R:71.49	Turn Right	Move Forward	Success
10	F:8 L:15 R:15	F:8.76 L:15.77 R:15.98	F:9.07 L:21.53 R:15.62	Turn Arround	Turn Left	Success

V. CONCLUSION

Collision avoidance on multi-robot system is such a complicated problem which can be solved by mixing more than one algorithm. In advance, it can be improved to collision prevention to reduce the possibility for robots to collide. However, it needs sensor fusion system, such as combining ultrasonic sensor as range finder and compass to define robot's orientation.

Each area have to be marked by the robots by using values of various sensors in order to determine the well path of every single robot. To overcome them, fuzzy logic control can be added to the system, especially in right wall following mechanism. Right wall following is used to simplify the movement model of robot system. Furthermore, communication between robots is established to share the location of each robot.

On the other hand, some problems are still happened in the system. It is mostly caused by inaccuray of sensor values. Somehow, it can make robot moves unproperly. Since multi-robot system involves more than a robot in the environment, timing is very considered as an important parameters. For example, robot can collide when they face each others if a robot make a late decision to turn arround even another can make a well decision.

In the next research, it will be added a mapping system in order to know exact location of the explored environment. It also can be used to defined the path of each robot. In advance, it can be developed to coverage the area.

REFERENCES

[1] Angelo Martinez, Eddie Tunstel1, and Mo Jamshidi " Fuzzy Logic Based Collision Avoidance For a Mobile Robot". CAD Laboratory for Intelligent and Robotic Systems, 1994.

[2] Reinhard Braunstingl, Pedro Sanz, Jose Manuel Ezkerra "Fuzzy Logic Wall Following of a Mobile Robot Based on the Concept of General Perception". ICAR '95, 7th International Conference on Advanced Robotics, Sant Feliu de Guixols, Spain, PP.367-376, SEPT., 1995.

[3] I. Gavrilut, V. Tiponut, A. Gacsadi, L. Tepelea" Wall-following Method for an Autonomous Mobile Robot using Two IR Sensors "University of Timisoara, B-dul Vasile Parvan No. 2, 300223 Timisoara, Romania.

[4] Reinhard Braunstingl, Jose Manuel Ezkerra, Pedro Sanz "Fuzzy Logic Wall Following of a Mobile Robot Based on the Concept of General Perception" ICAR '95, 7th international conference on advanced robotics, sant Feliu De Guixols, Spain, PP.367-376, SEPT., 1995.

[5] Michael G. Murphy, Ph.D. "Fuzzy Logic Path Planning system for collision Avoidance by an Autonomous Rover vehichle", NASA/ASEE Summer Faculty Fellowship Program, 1991.

[6] C.G. Rusu, I.T. Birou "Obstacle Avoidance Fuzzy System for Mobile Robot with IR Sensors ",10th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 27-29, 2010.

[7] David Llorca, Vicente Milanés, Ignacio Parra, Miguel Gavilán, Iván García Daza, Joshué Pérez Rastelli, M.A. Sotelo, " Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller", IEEE Transactions on Intelligetn Transportation System, VOL. 12, No. 2, June 2011.

[8] Akib Islam, Farogh Ahmad, P.Sathya, " Shortest Distance Maze Solving Robot", International Journal of Research in Engineering and Technology, eISSN: 2319-1163 | pISSN: 2321-7308.

[9] Marco A. C. Simões, Helder Guimarães Aragão, Victor Souza, Simon Viegas, " Using Fuzzy Logic to Build a Heterogeneous Multiagent System for the Robotics Soccer Problem " Computer Architecture and Operating Systems Group(ACSO) State University of Bahia (UNEB).

[10] Yee Mon Nyein, Nu Nu Win, "Path Finding and Turning with Maze Solving Robot", International Journal of Science, Engineering and Technology Research (IJSETR) Volume 5, Issue 9, September 2016, ISSN: 2278 – 7798.

[11] Yuan Zhou, Hesuan Hu, Yang Liu, and Zuohua Ding, "Collision and Deadlock Avoidance in Multirobot Systems: A Distributed Approach "IEEE Transaction on Systems, man and Cybernetics Systems. 2017.

[12] Dali Sun, Alexander Kleiner, Bernhard Nebel, "Behavior-based Multi-Robot Collision Avoidance", 2014 IEEE International Conference on Robotics & Automation (ICRA), Hong Kong Convention and Exhibition Center May 31 - June 7, 2014. Hong Kong,China.

[13] Aulian Miftahul Fathan, Agung Nugroho Jati, Randy Erfa Saputra, " Mapping Algorithm Using Ultrasonic And Compass Sensor On Autonomous Mobile Robot," ICCEREC, September 2016.

[14] NT AlGhifari, AN Jati, RE Saputra, "Coordination Control for Simple Autonomous Mobile Robot ", International Conference on Instrumentation Control and Automation, pp 93-98, August 2017.

[15] Agung Nugroho Jati, Randy Erfa Saputra, Muhammad Ghozy Nurcahyadi, Nasyan Taufiq Al Ghifary,"A Multi-Robot System Coordination Design and Analysis On Wall Follower Robot Group", IJECE, Vol 8 (6), Dec 2018.