SINERGI Vol. 24, No. 2, June 2020: 141-152
DOAJ:doaj.org/toc/2460-1217
DOI:doi.org/10.22441/sinergi.2020.2.008 .

ENHANCING THE PERFORMANCE OF THE WALL-FOLLOWING ROBOT BASED ON FLC-GA

Heru Suwoyo[1,3,*], Yingzhong Tian[1,2], Muhammad Hafizd Ibnu Hajar[3]
[1]School of Mechatronic Engineering and Automation of Shanghai University
Shanghai, 200444, China
[2]Shanghai Key Laboratory of Intelligent Manufacturing and Robotics
Shanghai, 200444, China
[3]Electrical Engineering Department, Faculty of Engineering, Universitas Mercu Buana
Jalan Meruya Selatan, Kembangan, Jakarta 11650, Indonesia
[*] Corresponding Author: heru.suwoyo@mercubuana.ac.id

Abstract -- *Determination of the improper speed of the wall-following robot will produce a wavy motion. This common problem can be solved by adding a Fuzzy Logic Controller (FLC) to the system. The usage of FLC is very influential on the performance of the wall-following robot. Accuracy in the determination of speed is largely based on the setting of the membership function that becomes the value of its input. So manual setting on membership function can still be enhanced by approaching the certain optimization method. This paper describes an optimization method based on Genetic Algorithm (GA). It is used to improving the ability of FLC to control the wall-following robot controlled by FLC. To provide clarity, the wall-following robot that controlled using an FLC with manual settings will be simulated and compared with the performance of wall-following robots controlled by a fuzzy logic controller optimized by a Genetic Algorithm (FLC-GA). According to comparative results, the proposed method has been showing effectiveness in terms of stability indicated by a small error.*

Keywords: Wall-Following Robot; Fuzzy Logic Controller; Genetic Algorithm; FLC-GA

Received: November 17, 2019 *Revised: March 2, 2020* *Accepted: March 6, 2020*

INTRODUCTION

The implementation of mobile robots, especially wall-following robots, has been covering many sectors such as industry and office. It is commonly used to reduce the potential risks [1], [2]. This robot behaves to follow contours of objects such as walls and obstacles in the environment and can also be combined with more intelligent behavior to complete high-level tasks [2, 3, 4, 5, 6, 7]. There are many factors related to the navigation task that makes wall following robot has been consuming the attention of many researchers. It is proven with many existences of the different types of methods approached to improve the capability of the wall-following robot. For instance, there has been an existing improvement, which is indicated by retrieving and processing sensor data and deciding when the robot is operated [1], [8].

The main objective of improving the wall following robot lies in maintaining the robot on the desired track. The robot moves away from the wall when it is closed. Oppositely, the robot moves closer to the wall when it is far. However, it is hard to find the proper value of the speed for the turn action. Consequentially, improper makes the wall-following robot moves wavy. Therefore, a certain controller is often used to overcome this issue. The most popular one is the Fuzzy Logic Controller (FLC) [2, 3, 4, 6, 9, 10, 11, 12, 13]. It can be designed and assigned as a closed-loop controller. By means, that the output of deciding is directly used as the feedback for the next step. The working principle of FLC is similar to a Proportional Integral Derivative (PID) Controller [1], [8].

In use as the controller of the wall-following robot, both are commonly working depend on the error, representing the values between the setpoint (zero-error) and actual error given by the distance sensor [3], [14]. FLC is a control design in which the decision is made by applying a fuzzy interference system based on rules or knowledge that contains the string if-then fuzzy rules [15]. The presence of these controllers is crucially benefiting as an influence major. It transforms the attention of traditional controls focusing on robot sensor accuracy to the modern controls focusing on decision making based on the high uncertainty, complexity, and nonlinearity [7]. The popularity of FLC can be faced from several types of behaviors of a mobile robot, which are successfully improved, such as path-following [9, 16, 17, 18, 19], avoiding-obstacle [6, 15, 17, 20, 21] and goal-seeking [6], [13] robots.

The effectiveness of the use of FLC is greatly influenced by the setting of the input membership function. Traditionally, it is adjusted manually by only connecting the input and output membership function with the width of all the subset [11], [22]. It is done with the assumption that the range of each representative function is known. However, the manual arrangement is no longer recommended since the difficulty of finding the value of input which proper to the certain value of the output. For this reason, there have been existing several methods adopted from the heuristic-based strategy, which commonly used. One of them is well-termed as the Genetic Algorithm (GA) [18, 23, 24, 25].

As discussed in this paper, the role of GA is used to improve the ability of FLC. In which, FLC is approached to produce the precise value of the output based on the random input. An FLC is chosen as the controller to maintain a distance between the robot to the wall. Then the FLC processes the initial distance of the robot to the wall and decides for the speed based on the combination of this initial distance and value of feedback. This value is then called an error, which is gained by referring to the diversity of the actual distance and desired distance or setpoint. Since the error is formed as the variables of the input membership function, then its arrangement is automatically handled by using GA. Initially, some numbers representing the values of the error are randomly generated. It is assumed within certain arranges to respect to the highest and lowest value of the input membership function. Next, in order to ease the process of the mutation and crossover, all the generated values are then converted as the binary values [8, 19, 24, 25, 26, 27, 28]. Continuously, they are evaluated by using the fitness function. It is the function representing the summation of the error. Besides that, the process of the mutation and crossover is then concerned as the closing step of GA producing the new population. There are two different performances in this experiment, namely the normal and optimized one. They are simulated and compared. Based on the comparison, the proposed method is able to significantly improve the performance of the wall-following robot. It can be stated based on the stability and accuracy of the proposed method compared to the conventional one.

The rest of this paper is organized as follows; Materials and Methods are described in Section 2, Experimental Results are described and discussed in Section 3, and the Conclusion is presented in section 4.

METHOD

A wheeled robot is used to apply the proposed method. This robot is designed with a rigid body made of the two-tiered acrylic. Several components, such as Arduino - Mega microcontroller, motor driver, and HCSR04, are placed on the gap between two layers. The microcontroller has a role as the programmable controller, and the motor driver is used to change the command transferred from the microcontroller aiming to control the motor rotation.

Meanwhile, HCSR04 is a type of proximity sensor that is installed on the robot to measure the distance of the surrounding environment. In this case, three proximity sensors are placed on the front, right and left side. The wheeled mobile robot consists of 3 wheels, two driven wheels connected to DC motor, and a freewheel. Each DC motor is equipped with a rotary encoder. While the freewheel is omnidirectionally movable, the appearance of this robot can be seen in Figure 1.

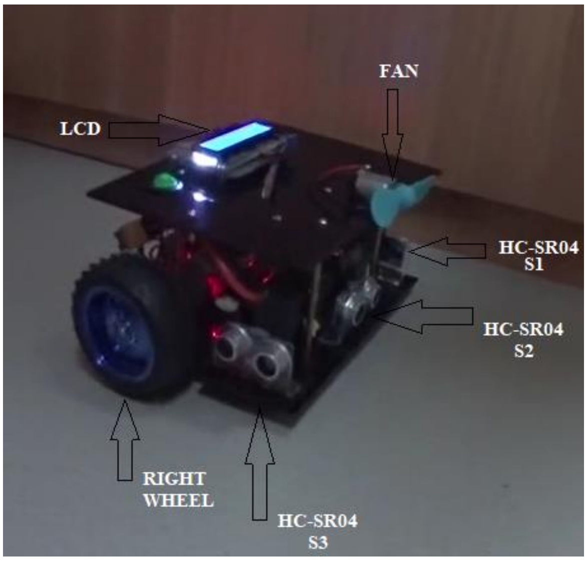

Figure 1. Wall-Following Robot

Figure 1 also depicts the displacement of the distance sensor, which is located on the right, left, and front side of the robot’s body. The sensors are then used as the main base to decide movement. The movement principle is like the wheelchair, in which all the turn and move actions are based only on the different speeds of the driven wheels. By means, it turns right when the speed of the right wheel is slower than the left one. Oppositely, when the right wheel is faster than the left one, the robot turns left.

Meanwhile, it moves forward when there is no difference between the right and left wheel's speed. It eases the user to steer the robot. However, as the effort of improving the performance of the wall-following robot, the simulation is recommended. It is due to the fast changes of dynamicity condition. In order to do simulation, the robot should be modeled with the complete assumption of the factor of the movement. In this case, it can be done by initially approaching the kinematic configuration and then adopting the differential steering system.

Kinematic Configuration and Steering System of the Wall-Following robot

The kinematic configuration is the representative model, which commonly used as the base to operate the robot movement in the case of simulation.

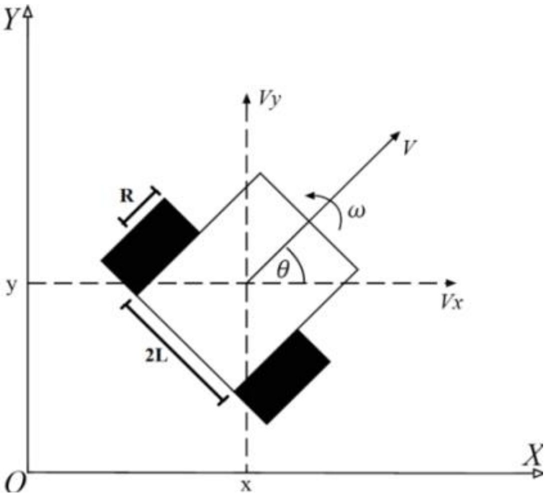

Figure 2. Kinematic Configuration of the Wheeled Mobile Robot

Fundamentally, by approaching the kinematic, it allows the user to move the robot without any consideration to the causes of the movement, such as the physic law, mass, slippage, and force. Therefore, the model of the robot can be easily modeled based only on some parameters such as the linear velocity, angular velocity, length of the robot's body, and diameters of the wheels. Moreover, referring to this assumption, the differential steering system of the robot can also be adopted. It allows the user to know the future position of the robot by knowing the previous position and values of the linear and angular velocity. Detailly, it can be discussed as follows. Assuming that the robot is initially placed on the location denoted by p representing the spatial pose (x,y) and orientation pose or heading θ, then the current pose of the robot can be analytically expressed in Equation (1).

$$p_{(t)} = [x_{(t)} \quad y_{(t)} \quad \theta_{(t)}]^T \tag{1}$$

It is noted that this analogy represents the model of the robot on the planar place of 2D Cartesian. Then by regarding the movement of the robot is influenced by angular velocity ω and linear velocity v, the movement action of the robot can be mathematically calculated in Equation (2).

$$\acute{p} = \begin{bmatrix} \acute{x} \\ \acute{y} \\ \acute{\theta} \end{bmatrix} = \begin{bmatrix} cos\theta_{(t)} & 0 \\ sin\theta_{(t)} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{2}$$

where, p' refers to the acceleration of the robot respecting to the linear velocity v and angular velocity ω. Next, after summing the current with the transition action in Equation (2), the future pose of the wheeled mobile robot can be expressed in Equation (3).

$$p_{(t+1)} = p_{(t)} + \acute{p} \tag{3}$$

where t represents the discrete-time index. Finally, the complete equation of the differential steering system, which refers to the affection of the angular velocity for each independent wheel with the angular velocity of the right ω_r and left wheel ω_l are summarized in Equation (4)-(7).

$$v = \frac{(\omega_r + \omega_l)R}{2} \tag{4}$$

$$\omega = \frac{(\omega_r - \omega_l)R}{2L} \tag{5}$$

$$\omega_r = \frac{v + \omega L}{R} \tag{6}$$

$$\omega_l = \frac{v - \omega L}{R} \tag{7}$$

Up to this point, the completeness of the differential steering system of the mobile robot used in this experiment is satisfied. However, it represents the wheel mobile robot placed on the free-obstacle environment. Therefore, in order to apply as the wall-following robot, it should be remodeled. The new model should not be much different from Figure 2 because the lack is only the displacement of the wall. For this reason, by considering that the robot is placed on the right side from the flat wall, then the model can be graphically depicted in Figure 3.

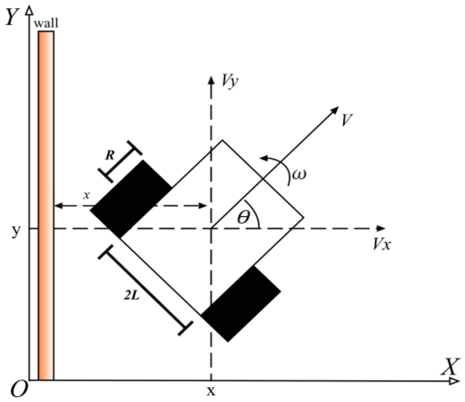

Figure 3. The Model of Wall-Following Robot in Planar Environment

Figure 3 shows the initial position of the robot is known with *R* = 3.5cm and *L* = 10cm   and . In which the parameter of *L* refers to half of the length of the robot body, and *R* refers to the radius of each wheel. Since the model is similar without any changes to the base and causes of the movement, Equation (1) - Equation (7) can also directly applied. According to Figure 3, it is clear that as the wall-following robot, the robot has the main task of tracking the wall placed on the left side. It can be done by maintaining the robot to be always on the desired path when it moves. Furthermore, by referring to the differential steering system and since the turn and transition action of the robot is based on the linear and angular velocity, making the robot to always moves on the desired track can be easily conducted by applying the angular velocity respect to the previous orientation and adjusting the proper linear velocity based on the different value between the actual distance *ad* and setpoint *sp*. This different value is commonly called as the error value. Mathematically, the error of each movement step can be calculated as follows

$$e(t+1) = sp - ad(t) \qquad (8)$$

where setpoint is set to be equal to 7cm, representing the ideal displacement of the desired path. Moreover, by summing all recorded error value during operations, the performance of the wall-following robot can be observed. Therefore, the sum of the positive error of each step can be used as the quality checker for the performance of the wall-following robot. It is then analytically expressed as follows

$$te = \sum_{it=0}^{max} e(it)^2 \qquad (9)$$

Where *it* represents the iteration of each movement of the wall-following robot reaching the final pose, and in this experiment, the final pose is represented by the maximum iteration in value of 50 times.

Implementation of Fuzzy Logic Controller

As mentioned earlier, the objective of keeping the robot to be always on the desired path is by adjusting the proper linear velocity, and, in this project, it is taken and handled by the Fuzzy Logic Controller. It is involved as the closed-loop controller. Theoretically, FLC is a problem mechanism system inspired by the principle of a human expert, deciding [26]. The significant characteristic of FLC commonly involves the utilization of linguistic terms as the representation of certain values that processed before it is distributed to the membership function. The effectiveness of its usage lies in the arrangement of the input membership function. And to accomplish the proper adjustment, the fuzzy set theory needs to be concerned. Generally, the architecture of FLC can be conducted by the following steps. The first is the fuzzification, which converts the chirp data into the linguistic term and arranges the membership function. The second is the fuzzy inference, which combines the membership function and rules base refers to the assumption relation IF-THEN. And the third is the defuzzification which reconverts back the linguistic data into the chirp data [26]. These processes can be expressed in Figure 4.

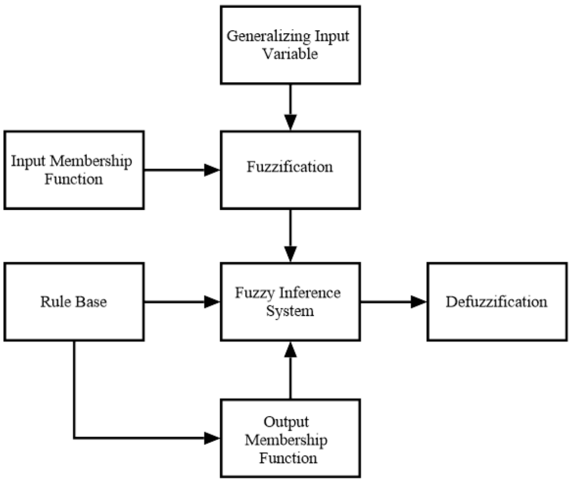
Generalizing Input Variable
Input Membership Function
Fuzzification
Rule Base
Fuzzy Inference System
Defuzzification
Output Membership Function

Figure 4. Flowchart of Fuzzy Logic Controller Design

Therefore, by referring to the flowchart shown in Figure 4 and the working principle of FLC, the error is assumed as the input, and linear velocity is assumed as the output. Meanwhile, the error calculation is obtained by using Equation (8) and the actual distance for each step from the read distance of HC-SR04 (mounted on the left side). Both the input and output are firstly modeled in the form of membership function as the first step of FLC, fuzzification. Note that the errors are divided into two types, negative and positive. Therefore, by assuming that the turn of the robot is less than 1cm, then the error is considered on the ranges of -1dm – 1 dm (see Figure 5). The arrangement of the input membership function is then assumed for having three subsets of linguistic term, close, medium, and far. They are respectively representing the distance from -1 up to 0, -0.5 up to 0.5, 0 up to 1 on the scale of decimeter.

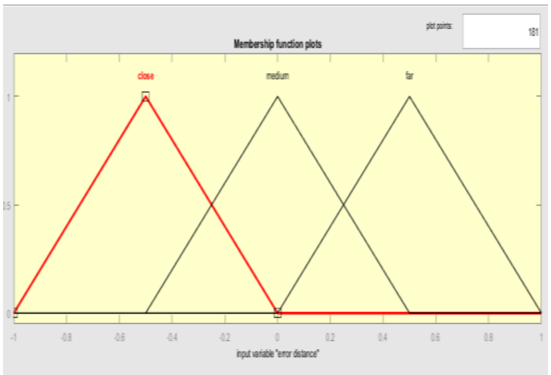

Figure 5. The Input Membership Function

Figure 5 also depicts the arrangement of the input membership function before optimization. Furthermore, by considering that the architecture of the closed-loop controller is the Single-Input-Single-Output (SISO), the rest prerequisite is then modeling the output membership function as well as classifying the representative linear velocity respecting to the input above. It is done by initially predicting the proper ranges for the output when the robot is located on the close, medium, and far distance. As the effort to make the robot moves slowly to the desired path when it is close, the output of the linear velocity is slow on the ranges 0 - 0.15 cm/s. Oppositely, in order to make a robot move vastly when the robot is on the far distance, the proportional velocity is designed to satisfy the ranges of 0.15-0.3 cm/s.

Meanwhile, the robot is assigned to moves on the ranges of 0.1 – 0.2 cm/s when the robot is on the medium distance. The reason behind this analogy is the usage of the rule-base, which connects the input and output membership function proportionally, is Mamdani's connecting law of "if-then." Accordingly, the output membership function is depicted in Figure 6.

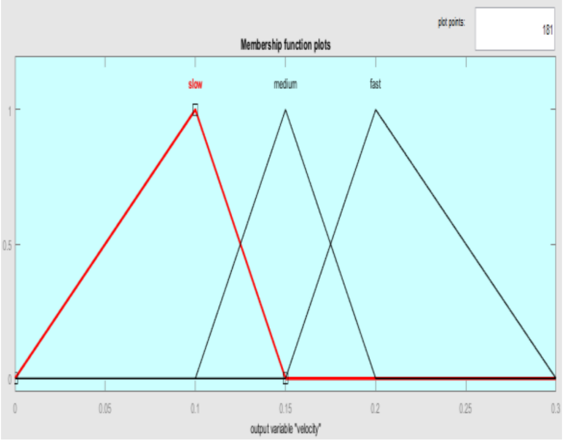

Figure 6. The Output Membership Function

Figure 6 shows the arrangement of the output membership function before optimization. It represents the corresponding output of the error might occurs along with the robot movement. Next, by concerning the Fuzzy Inference System, which takes the material consideration based on the relation between the input and output membership function and rule-base, FLC then produces the initial output. Before it is proceeded on the process, well-termed as defuzzification, it is the process of reconverting the linguistic result to values useful for the robot movement. Simply, to this assumption, the whole process of FLC controlling the wall-following robot can be streamed, as shown in Figure 7.

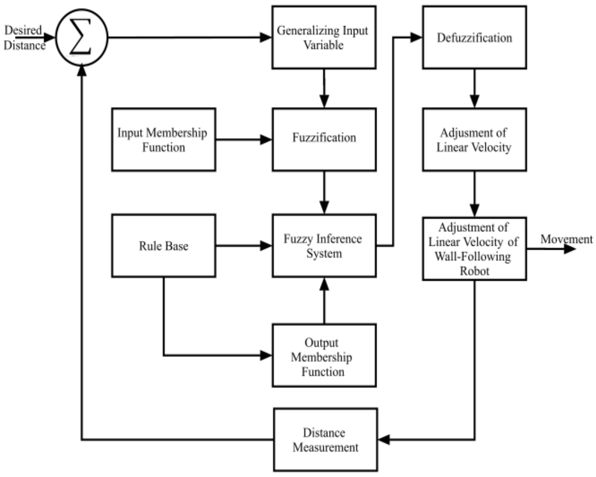
Desired Distance
∑
Generalizing Input Variable
Defuzzification
Input Membership Function
Fuzzification
Adjusment of Linear Velocity
Rule Base
Fuzzy Inference System
Adjustment of Linear Velocity of Wall-Following Robot
Movement
Output Membership Function
Distance Measurement

Figure 7. Developed Flowchart of FLC used for Controlling Wall-Following Robot

Figure 7 shows that the FLC generates the result based on the FIS and summation of the feedback value, which is representing the actual distance and the setpoint distance, which is predefined at the beginning. In which, the initial result produced by FIS is still in the linguistic term based on all the relationship shown in Table 1. The working principle of FIS can also be graphically depicted, as shown in Figure 8.

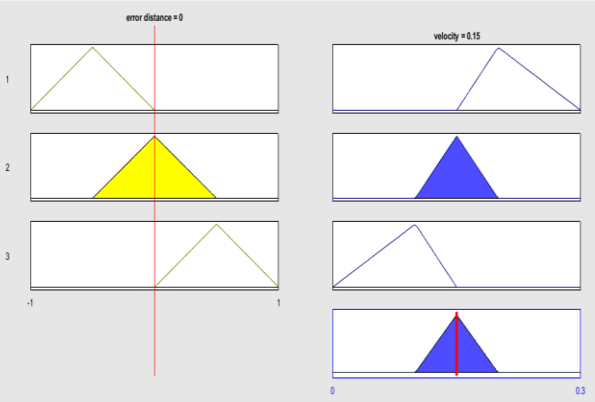

Figure 8. Result of Non-Optimized Membership Function

Since the result under the FIS process is in the linguistic term, it is then reconverted in to chirp as the initial form of the input. This step is known-well as defuzzification, which is the final step of FLC. Theoretically, the law used in this process is called centroid defuzzification [22], [29], [30]. It is mathematically expressed as follows

$$z_0 = \frac{\int \mu_i(x)\ x\ dx}{\int \mu_i\ (x)dx} \qquad (9)$$

Where z_0 is defuzzified output, μ_i is a membership function, and x is the output variable. Note that both the fuzzification and defuzzification process of this experiment directly refer to the relation of the input and output membership function, which can be presented in Table 1.

Table 1. Rule Base of Fuzzy Logic

No	Input (Error Distance)	Chirp Data (dm)	Output (Linear Velocity)	Chirp Data (dm/s)
1	Close	-1.00-0.00	Slow	0.00-0,15
2	Average	-0.50-0.50	Medium	0.10-0.20
3	Far	0.00-1.00	Fast	0.25-0.30

Adjusting Input Membership Function Using Genetic Algorithm

The success of using FLC can be achieved by arranging the input membership function. Traditionally, it can be done by manually tuning or adjusting the width of each subset. However, since the input is characteristically random and unpredictable, manual tuning is no longer recommended. For this reason, the GA is involved as the optimization method used to arrange the input membership function properly. It aims to improve the capability of FLC controlling the wall-following robot. Theoretically, GA is a heuristic-based strategy adopting the principle of the evolution and natural selection mechanism [27][28]. Generally, the process of the GA involves some evolution operators such as mutation, crossover, and selection. This process is intended to generate the new population that consists of some chromosomes based on the initial generated population. Each single chromosome stores the solutions for a particular process. And all chromosomes are considered as candidate solution which has the divergent characteristic with another chromosome. In a looping process, GA produces a new population, and the system regards it as a generation. Generally, in finding the optimal solution, the usage of GA needs to define the maximum number of the generation. Moreover, at the end of the process of GA, the final population is termed as the offspring [7, 23, 31].

Generally, the GA optimization process has the following steps:

1. Generating Initial Population
2. Fitness Evaluation
3. Selection
4. Crossover and Mutation
5. Store Generated New Population

As the proposed method, GA is used to arrange the membership function of the error value since the dynamic error is considered as the input. It is then elaborated by referring to the setpoint distance and feedback from the actual distance. This process aims to effectively and accurately produce the linear velocity when the angular is directly set based on the previous heading value. The architecture of optimizing the FLC using the GA can be graphically depicted, as shown in Figure 9.

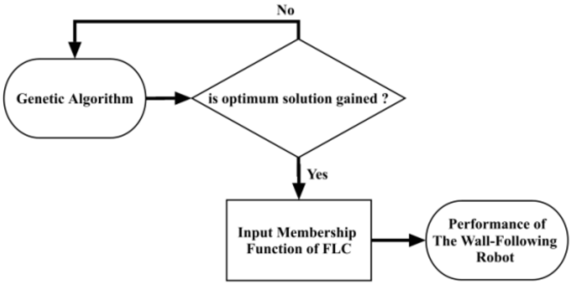
No
Genetic Algorithm
is optimum solution gained ?
Yes
Input Membership Function of FLC
Performance of The Wall-Following Robot

Figure 9. Optimization Process on FLC by GA

As can be seen from Figure 9 that the performance of the wall-following robot now depends on how well the GA adjusting the input membership function. Simply, the normal arrangement in Figure 5 is changed because of the presence of the optimization method. Since the optimization is intended to reduce the wavy movement by minimizing the error based on the previous error, the objective is the minimization-based strategy. Initially, the part of arrangement new membership function is prepared as shown in Figure 10.

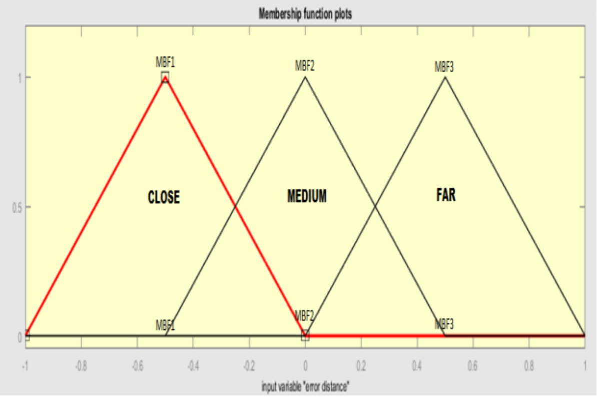

Figure 10. Optimization Target

According to Figure 10, there are three optimization targets, namely *MBF*1, *MBF*2, and *MBF*3. Their combination is considered as the chromosome storing the special influence to the FLC controller in producing the linear velocity. It is started by generating an initial population representing each chromosome in the population. The value is generated in decimal values. To ease the process of mutation and crossover, they are then converted into binary form, called a binary encoding. The arrangement of each chromosome is referred to as the ranges of each subset of the input membership function. It can be divided into three ranges, as can be seen in Figure 10. The range of *MBF*1 satisfies -1 up to 0, *MBF*2 satisfies -0.5 up to 0.5 up to, and *MBF*3 satisfies 0 up to 1. They are respectively representing the linguistic term of close, average, and far.

Secondly, once the initial population is generated, all possible solutions are evaluated using the fitness function. Since the performance of the wall-following robot can be evaluated by knowing the total positive-error for the whole process, Equation (9) can be directly used in the step of evaluation. Furthermore, the smallest fitness given by a certain chromosome is expected to be selected. There are some types of the selection process, but the roulettes wheel is used in this experiment. It is done by providing a level for each potential solution. Thus, the chromosome that has better solutions will have greater potential to be selected. Next, the qualified and selected chromosomes or the so-called new parent of a potential solution then proceeds on the crossover and mutation process. The crossover process can be done easily by exchanging bits of genes from two parental chromosomes by referring to the selection of specific genes randomly. It involves the probability of crossover, which allows the system to choose the bits or genes based on the value of the ratio. A smaller probability set a smaller number of bits are processed under the crossover process. The probability of crossover used in this experiment is set to be 0.85, which means that 85 of the 100 genes of a chromosome is exchanged for a paired chromosome. This process yields one potential solution that might better than the parental one. The crossover regenerates the new population by increasing the quality of all of the new chromosomes. Since this process is conducted, the new population is then proceeded into the crucial process, namely the mutation step.

The mutation process allows each particular gene to mutate. By definition, it is only shifting the bits to increase the quality of each chromosome, respecting to the fitness function. The bit is chosen randomly based on the probability of mutation. Since the mutation probability greatly affects the results, so in this experiment, it is set to be equals to 1/8. It means that 1 of 8 genes on 1 chromosome is mutated or shifted, from 1 to 0 or vice versa. The mutation process closes all the steps of performing GA. Simply, after processing the chromosome on the mutation step, the corresponding generation produces a new result, it is further called an optimum solution.

Finally, by evaluating the potential solution using the fitness function, two optimum solutions from a different generation can be compared. If the result obtained has a smaller error to the previous solution, then it is stored automatically as the new base for the next optimization. Meanwhile, if the result is still bigger, the process in a certain generation is repeated. But, if the result is the same as the previous generation, it is then ignored. By following this analogy, the solution is must be improved generation-by-generation. Simply, the solution stored by the last generation can be stated and considered as the desirably optimum solution. The representative result for each generation can be depicted in Figure 11. It shows the value of the fitness function for the whole performance of following the wall.

Figure 11 also shows that the maximum number of the generation is 140, in which each generation stores a better solution compared to the previous one. According to Figure 11, the optimum solution produced by GA is about 0.6386. The last solution (140th Generation) in Figure 11 is then transferred to the FLC system to arrange the input membership function. This process is done offline, in which the performance of the wall-following robot is heuristically observed. Therefore, in order to apply the GA as the online optimization method used to improve the FLC ability, the fast GA is applied in the real-implementation.

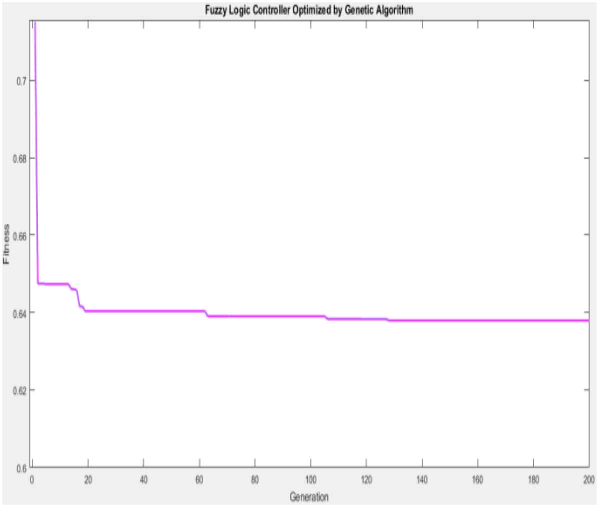

Figure 11. Fitness Value in Gaining the Best Solution for Input Membership Function

It is done by preparing the ranges of each chromosome with respect to the obtained solution from the simulation. Besides overcoming the lack of kinematic approach, which ignores some factors mentioned earlier, the existence of the fast GA has no effect on the sensing process of HC-SR04.

RESULT AND DISCUSSION

This experiment aims to improve the performance of the wall-following robot by solving its general problem, namely the wobble when it is operated. The proposed method is a controller applied for a wall-following robot based on FLC optimized by using GA. As mentioned above, the usage of FLC is to control a linear velocity of the wall-following robot based on the sensed distance and predefined setpoint in terms of the error value. Accordingly, the arrangement of the input membership function of the normal case in Figure 5 is automatically improved since the GA is approached. The new arrangement can be seen in Figure 12.

Figure 12 shows that referring to the wall and generated error for each movement, the subset for each membership function is rearranged. Simply, the *MBF*1, *MBF*2 and *MBF*3 are changed automatically. Of course, it influences the performance of the wall-following robot.

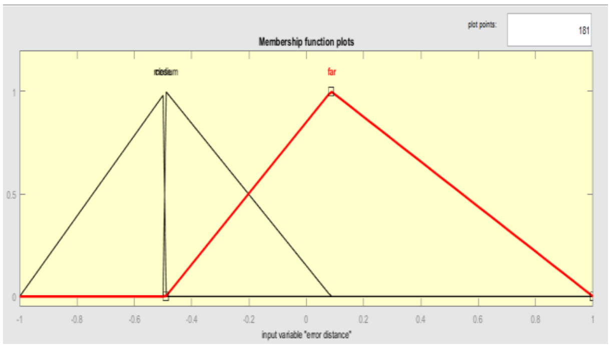

Figure 12. Input Membership Function Arranged by GA

Therefore, to get a significant comparison, the simulated performance of a wall-following robot based on FLC with manual tuning and FLC-GA are presented. The initial simulation begins by assuming that the wall is located on the left side of the robot. The wall is flat with the initial length of 2.5m, a bend, and back straight along the next 3.5m and a curve back straight along with the last form, which is 2.5m. It can be seen in Figure 13 and Figure 14.

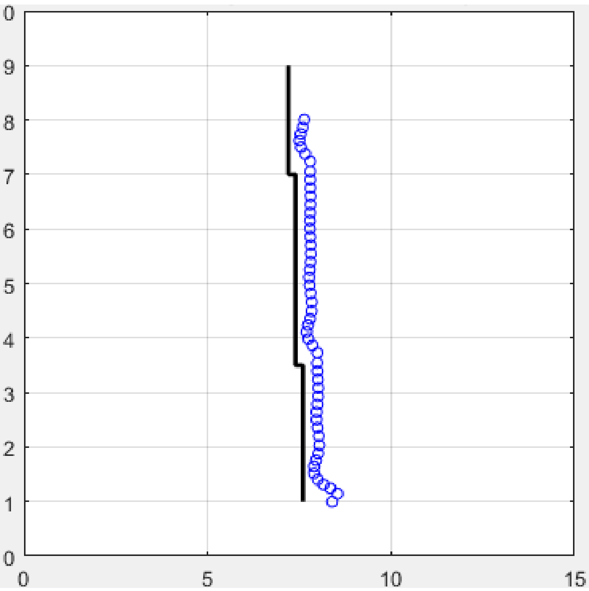

Figure 13. Performance of The Wall-Following Robot Controlled by FLC with manual tuning

Figure 13 illustrates the movement of the wall-following robot controlled by FLC tuned manually within 50 times. It started from an initial position located on x = 8.4, y = 10 and θ = π/4. Although the speed of the robot is relatively fast, the main problem of the wall-following robot is not solved, which is the wavy movement. It is not safe when the wall is not flat with a wide dynamic condition.

For this reason, the main threshold used as the validation parameter is only the smoothness of the movement. It also proves that the performance of the wall-following robot controlled by FLC with manual tuning is not recommended. For this reason, the optimization method is strongly concerned with this experiment. Similarly, the graphical result representing the wall-following robot controlled by FLC-GA is presented in this paper. It can be seen in Figure 14.

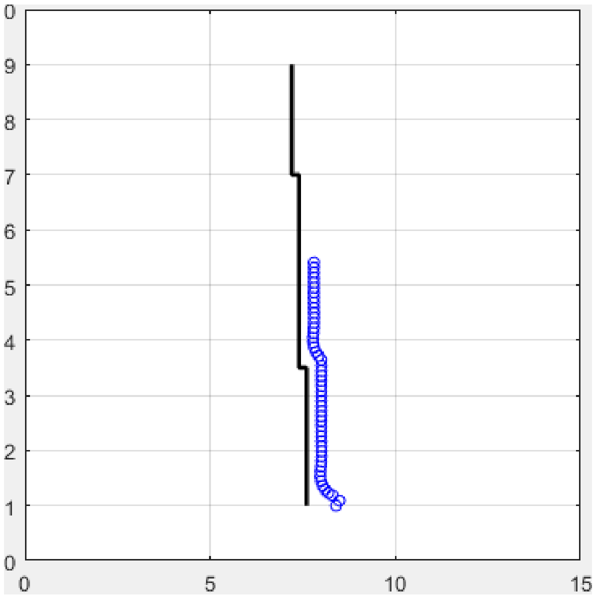

Figure 14. Performance of the Wall-Following Robot Controlled by FLC-GA

Figure 14 illustrates the movement of the wall-following robot controlled by FLC-GA within 50 times of move stepping. It started from the same initial position, which is located on $x = 8.4, y = 1, \theta = \pi/4$. Besides that, it also shows that by changing the arrangement of the input membership function, the wavy movement presented by FLC with manual tuning is gone. Although the speed is slower compared to the FLC with manual tuning, the proposed method is significantly overcoming the main issue of the wall-following robot. Moreover, since the stability and accuracy are used as the threshold in this experiment, it can now be stated by the proposed method gives better performance compared to the conventional one, FLC, with manual tuning. The effectiveness offered by the proposed method can also be observed and validated according to the error values in each step movement. It can be seen in Table 2 clearly.

Table 2 shows that the wall-following robot performs by executing 50 commands from the microcontroller. Each command assigns the robot to move forward or turn based on the structure of the wall. It aims to follow the wall located around its body.

It can be seen from Table 2, that the performance of the wall-following robot controlled by FLC-GA always gives better act shown by a small error. It can be analyzed from the error value, which always closed to zero error or ideal distance. Furthermore, the achievement distance from Table 2 illustrates that by applying the proposed method, the wall-following robot is not that slow compared to the conventional performance. Again, it proves that the effectiveness and stability of the proposed method are satisfied. The comparison shown by Table 2 might be hard to be validated. For this reason, the graphical representation of the error values for a different performance of the wall-following robot is presented. It can be seen in Figure 15.

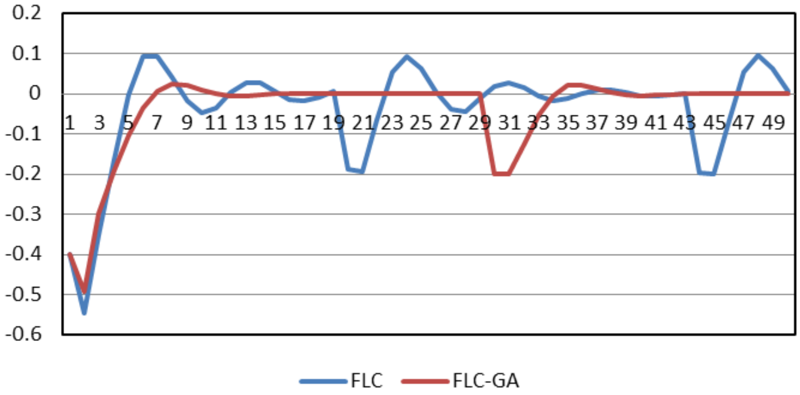

Figure 15. Error Comparison

Table 2. Error and Distance Achievement of the Wall-Following Robot Controlled by FLC with manual tuning and FLC optimized with GA

The Performance of WFR Controlled by FLC with normal setting			The Performance of WFR Controlled by FLC-GA		
Step	Error (cm)	Achievement (m)	Step	Error (cm)	Achievement (m)
1	-4.0000000	0.14606994	1	-4.0000000	0.09038344
2	-5.4606994	0.24302586	2	-4.9038344	0.18728276
3	-3.5215811	0.31270675	3	-2.965848	0.23035583
4	-1.6190492	0.40417552	4	-1.9097319	0.28903056
5	-0.0084757	0.52101942	5	-1.0396302	0.36047047
6	0.9374044	0.64588188	6	-0.3574749	0.44382135
7	0.9426959	0.75885652	7	0.0757956	0.53298874
8	0.4131813	0.88227195	8	0.2351711	0.62219793
9	-0.1685348	1.03521386	9	0.2013628	0.71113083
10	-0.4844985	1.20005622	10	0.0967905	0.80108268
11	-0.3455901	1.35718728	11	0.0062258	0.89212952
12	0.0350587	1.50366153	12	-0.0378366	0.98364267
13	0.288159	1.64364406	13	-0.0406853	1.0751631
14	0.2636209	1.7829708	14	-0.0233712	1.16653977
15	0.0628797	1.92929998	15	-0.0047828	1.25777712
16	-0.1299975	2.08407392	16	0.0058789	1.3489331
17	-0.1786582	2.24024872	17	0.0080588	1.44007108
18	-0.0771465	2.39257731	18	0.0053798	1.53123067
19	0.0589272	2.54021022	19	0.0017066	1.62242078
20	-1.884126	2.72837616	20	-0.0007463	1.71363128
21	-1.9395664	2.86583752	21	-0.0015246	1.80484817
22	-0.6445939	2.98733481	22	-0.0011842	1.89606226
23	0.5336664	3.11425983	23	-0.0004889	1.98727067
24	0.9427419	3.23480951	24	5.115E-05	2.07847466
25	0.6210753	3.35355836	25	0.0002741	2.16967683
26	0.0613277	3.49456861	26	0.0002508	2.2608792
27	-0.3765622	3.65704211	27	0.0001258	2.35208258
28	-0.4263829	3.8181332	28	1.142E-05	2.44328691
29	-0.1230788	3.96933909	29	-4.594E-05	2.5344917
30	0.1992792	4.11189435	30	-2.0000511	2.64012085
31	0.2870069	4.25124382	31	-2.0000269	2.71481085
32	0.1481596	4.39428685	32	-1.2531078	2.78569086
33	-0.0571121	4.54606022	33	-0.5442982	2.86642769
34	-0.1695455	4.70220147	34	-0.0384384	2.95473437
35	-0.1249577	4.85631966	35	0.2018874	3.04425922
36	0.0056925	5.00580567	36	0.2190934	3.13321009
37	0.0990896	5.15206565	37	0.1293031	3.22281076
38	0.0949267	5.29829843	38	0.0311485	3.31357004
39	0.022476	5.44725782	39	-0.0275288	3.40498847
40	-0.0482252	5.59910302	40	-0.0417665	3.49652542
41	-0.0652895	5.75155377	41	-0.029167	3.58794836
42	-0.0285297	5.9025788	42	-0.0100749	3.67922555
43	0.0207721	6.05175636	43	0.0032365	3.77040229
44	-1.957948	6.24079283	44	0.0078295	3.86154242
45	-1.9775814	6.37602507	45	0.0063546	3.95269407
46	-0.6536929	6.49651726	46	0.0027863	4.0438752
47	0.5377226	6.62317836	47	-0.0001108	4.13508043
48	0.9517099	6.74351147	48	-0.0013815	4.22629617
49	0.6281808	6.8619101	49	-0.0013309	4.31751146
50	0.064775	7.00265505	50	-0.0007009	4.4087216

Figure 15 illustrates that there is no significant wobble of the wall-following robot controlled by FLC-GA. Meanwhile, the main issue of the wall-following robot still occurs when it is controlled by FLC with only manual tuning.

CONCLUSION

This paper presents a method properly used to improve the performance of the wall-following robot. It utilizes a closed-controller, namely a Fuzzy Logic Controller (FLC). The main task handled by FLC is to produce the linear velocity of the robot based on the error value, which is calculated by referring to the actual distance and setpoint distance. Besides that, the FLC is optimized using the GA algorithm because of the lack of manual tuning. By optimizing the FLC, the arrangement of the input membership function is automatically conducted. Henceforth, the proposed method in this paper is often-termed as a Fuzzy Logic Controller optimized by the Genetic Algorithm (FLC-GA). According to the kinematic configuration and differential steering system, the performance of the proposed method is simulated as the controller for the wall-following robot. The result is then compared with the conventional one, which is the performance of the wall-following robot controlled by FLC with manual tuning. The comparison is validated, focusing on the stability and accuracy represented by the total error for the whole performance. Based on this comparison, the proposed method has been showing effectiveness.

ACKNOWLEDGMENT

The research was supported by Special Plan of Major Scientific Instruments and Equipment of the State (Grant No. 2018YFF01013101), National Natural Science Foundation of China (61704102), the IIOT Innovation and Development Special Foundation of Shanghai (2017-GYHLW01037), Project named "Key Technology Research and Demonstration Line Construction of Advanced Laser Intelligent Manufacturing Equipment" from Shanghai Lingang Area Development Administration, Universitas Mercu Buana, Jakarta, Indonesia.

REFERENCES

[1] A. Adriansyah, H. Suwoyo, and Y. Tian, "Improving wall-following robot performance using PID-PSO Controller," *Journal Teknologi*, vol. 3, pp. 119–126, May 2019. DOI: 10.11113/jt.v81.13098

[2] U. Farooq, A. Khalid, M. Amar, A. Habiba, S. Shafique, and R. Noor, "Design and low cost implementation of a fuzzy logic controller for wall following behavior of a mobile robot," *ICSPS 2010 - Proc. 2010 2nd Int. Conf. Signal Process. Syst.*, Dalian, China, 2010, pp. V2-740-V2-746. DOI: 10.1109/ICSPS.2010.5555781

[3] G. Der Wu, Z. W. Zhu, and C. W. Chien, "Sparse-sensing-based wall-following control design for a mobile-robot," *Proc. 2016 IEEE Int. Conf. Control Robot. Eng. ICCRE 2016*, Singapore, 2016, pp. 1–5. DOI: 10.1109/ICCRE.2016.7476144

[4] Y. T. Lee, C. S. Chiu, and I. T. Kuo, "Fuzzy wall-following control of a wheelchair," *IFSA-SCIS 2017 - Jt. 17th World Congr. Int. Fuzzy Syst. Assoc. 9th Int. Conf. Soft Comput. Intell. Syst.*, Otsu, Japan, 2017, pp. 1-6. DOI: 10.1109.IFSA-SCIS.2017.8023223

[5] L. H. Prayudhi, A. Widyotriatmo, and K. S. Hong, "Wall following control algorithm for a car-like wheeled-mobile robot with differential-wheels drive," *ICCAS 2015 - 2015 15th Int. Conf. Control. Autom. Syst. Proc (ICCAS).* Busan, South Korea, 2015, pp. 779–783. DOI: 10.1109/ICCAS.2015. 7364726

[6] K. Al-Mutib, M. Faisal, M. Alsulaiman, F. Abdessemed, H. Ramdane, and M. Bencherif, "Obstacle avoidance using wall-following strategy for indoor mobile robots," *2016 2nd IEEE Int. Symp. Robot. Manuf. Autom. ROMA,* 2016, pp. 1–6. DOI: 10.1106/ROMA.2016.7847817

[7] C. Chen, H. Du, and S. Lin, "Mobile robot wall-following control by improved artificial bee colony algorithm to design a compensatory fuzzy logic controller," *ECTI-CON 2017 - 2017 14th Int. Conf. Electr. Eng. Comput. Telecommun. Inf. Technol.*, Phuket, Thailand, 2017, pp. 856–859. DOI: 10.1106/ECTICon.2017.8096373

[8] H. Suwoyo, Y. Tian, C. Deng, and A. Adriansyah, "Improving a wall-following robot performance with a PID-genetic algorithm controller," *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, Malang, Indonesia, 2018, pp. 314–318. DOI: 10.1109/EECSI.2018.8752907

[9] E. A. Elsheikh, M. A. El-Bardini, and M. A. Fkirin, "Dynamic path planning and decentralized FLC path following implementation for WMR based on visual servoing," *2016 3rd MEC Int. Conf. Big Data Smart City, ICBDSC,* Muscat, Oman, 2016, pp. 148–154. DOI: 10.1109/ICBDSC.2016. 7460359

[10] M. Boujelben, D. Ayedi, C. Rekik, and N. Derbel, "Fuzzy logic controller for mobile robot navigation to avoid dynamic and static obstacles," *2017 14th Int. Multi-Conference Syst. Signals Devices, SSD 2017*, Marrakech, Maroco, 2017, pp. 293–298. DOI: 10.1109/SSD.2017.8166963

[11] M. Collotta, G. Pau, and V. Maniscalco, "A Fuzzy Logic Approach by Using Particle Swarm Optimization for Effective Energy Management in IWSNs," *IEEE Trans. Ind. Electron.*, vol. 64, no. 12, pp. 9496–9506, December 2017. DOI: 10.1109/TIE.2017. 2711548

[12] M. R. H. Al-Dahhan and M. M. Ali, "Path tracking control of a mobile robot using fuzzy logic," *13th Int. Multi-Conference Syst. Signals Devices, SSD 2016*, Leipzif, Germany, 2016, pp. 82–88. DOI: 10.1109/ SSD.2016.7473656

[13] F. Lachekhab and M. Tadjine, "Goal seeking of mobile robot using fuzzy actor critic learning algorithm," *Proc. 2015 7th Int. Conf. Model. Identif. Control. ICMIC 2015*, Sousse, Tunisia, 2015, pp. 1–6. DOI: 10.1109/ICMIS.2015.7409370

[14] C. F. Juang, Y. H. Jhan, Y. M. Chen, and C. M. Hsu, “Evolutionary Wall-Following Hexapod Robot Using Advanced Multiobjective Continuous Ant Colony Optimized Fuzzy Controller,” *IEEE Trans. Cogn. Dev. Syst.*, vol. 10, no. 3, pp. 585–594, September 2018. DOI: 10.1109/TCDS.2017.2681181

[15] A. S. Handayani, T. Dewi, N. L. Husni, S. Nurmaini, and I. Yani, “Target tracking in mobile robot under uncertain environment using fuzzy logic controller,” *Int. Conf. Electr. Eng. Comput. Sci. Informatics*, Yogyakarta, Indonesia, 2017, pp. 19–21. DOI: 10.1109/EECSI.2017.8239079

[16] R. Kuemmerle, “State Estimation and Optimization for Mobile Robot Navigation,” *Dissertation,* Univerisitaatsbibliothek Freiburg, pp. 1–191, 2013.

[17] C. H. Chinag and C. Ding, “Robot navigation in dynamic environments using fuzzy logic and trajectory prediction table,” *iFUZZY 2014 - 2014 Int. Conf. Fuzzy Theory Its Appl. Conf. Dig.*, Kaouhsiung, Taiwan, 2014, pp. 99–104. DOI: 10.1109/iFUZZY.2014.7091240

[18] X. Xinying, X. Jun, and X. Keming, “Path planning and obstacle-avoidance for soccer robot based on artificial potential field and genetic algorithm,” *Proc. World Congr. Intell. Control Autom.*, Dailan, China, 2006, pp. 3494–3498. DOI: 10.1109/WCICA.2006.1713018

[19] M. Wu et al., “Path planning of mobile robot based on Improved Genetic Algorithm,” *2017 Chinese Automation Congress (CAC),* Jinan, China, 2017, pp. 6696-6700. DOI. 10.1109/CAC.2017.8243983

[20] J. T. Marasigan, I. M. B. Saberon, D. P. B. San Jose, P. A. T. Sevilla, and A. A. Bandala, “Autonomous parallel parking of four wheeled vehicles utilizing adoptive Fuzzy-Neuro control system,” *IEEE TENSYMP 2014 - 2014 IEEE Reg. 10 Symp.*, Kuala Lumpur, Malaysiam 2014, pp. 640–644. DOI: 10.1109/TENCONSpring.2014.6863112

[21] A. Budianto *et al.*, “Analysis of artificial intelligence application using back propagation neural network and fuzzy logic controller on wall-following autonomous mobile robot,” *2017 Int. Symp. Electron. Smart Devices, ISESD 2017*, Yogyakarta, Indonesia, 2017, pp. 62–66. DOI: 10.1109/ISESD.2017.8253306

[22] Y. Bai and D. Wang, “Fundamentals of Fuzzy Logic Control - Fuzzy Sets, Fuzzy Rules and Defuzzifications,” *Advanced Fuzzy Logic Technologies in Industrial Applications. Advances in Indutstrial Control.* Springer, London, 2006, pp. 17–36.

[23] N. Lu, Y. Gong, and J. Pan, “Path planning of mobile robot with path rule mining based on GA,” *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, Yinchuan, China, 2016, pp. 1600-1604. DOI: 10.1109/CCDC.2016.7531239

[24] C. Caceres, J. M. Rosario, and D. Amaya, “Approach of Kinematic Control for a Nonholonomic Wheeled Robot using Artificial Neural Networks and Genetic Algorithms,” *2017 Int. Work Conf. Bio-Inspired Intell. Intell. Syst. Biodivers. Conserv. IWOBI 2017 - Proc.* Funchal, Protuga, 2017, pp. 1-6. DOI: 10.1109/IWOBI.2017.7985533.

[25] T. Duckett, “A genetic algorithm for simultaneous localization and mapping,” *Proc. - IEEE Int. Conf. Robot. Autom.* Taipei, Taiwan, 2003, pp. 434-439. DOI: 10.1109/ROBOT. 2003.1241633

[26] R. A. Dain, “Developing Mobile Robot Wall-Following Algorithms Using Genetic Programming,” *Appl. Intell.*, vol. 8, no. 1, pp. 33-41, Jan. 1998. DOI: 10.1023/A:1008216530547

[27] J. Ni, K. Wang, H. Huang, L. Wu, and C. Luo, “Robot path planning based on an improved genetic algorithm with variable length chromosome,” *2016 12th Int. Conf. Nat. Comput. Fuzzy Syst. Knowl. Discov. ICNC-FSKD 2016*, pp. 145–149, 2016.

[28] Y. Zhou, L. Zheng, and Y. Li, “An improved genetic algorithm for mobile robotic path planning,” *Proc. 2012 24th Chinese Control Decis. Conf.,* Taiyuan, China, 2012, pp. 3255-3260. DOI: 10.1109/CCDC.2012.6244515

[29] S. C. Satapathy, B. N. Biswal, S. K. Udgata, and J. K. Mandal, “Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014,” *Adv. Intell. Syst. Comput.*, vol. 327, pp. 175–183, 2014.

[30] X. Li and D. Wang, “Behavior-based mamdani fuzzy controller for mobile robot wall-following,” *Proc.-2015 Int. Conf. Control. Autom. Robot,* Singapore, 2015, pp. 78-81. DOI: 10.1109/ICCAR.2015.7166006

[31] F. Fei, H. Hongjie, and G. Zhongtong, “Application of genetic algorithm PSO in parameter identification of SCARA robot,” *Proc. - 2017 Chinese Autom. Congr. CAC 2017*, Jinan, China, 2017, pp. 923–927. DOI: 10.1109/CAC.2017.8242898