

# Optimization of Fuzzy Social Force Model Adaptive Parameter using Genetic Algorithm for Mobile Robot Navigation Control

Alif Wicaksana Ramadhan, Bima Sena Bayu Dewantara, and Setiawardhana  
Politeknik Elektronika Negeri Surabaya  
Kampus PENS, Jalan Raya ITS, Sukolilo, Surabaya 60111  
e-mail: alifwr@outlook.com

**Abstract**—The Social Force Model (SFM) is a popular navigation technique for mobile robots that is primarily used to simulate pedestrian movement. The SFM method's drawback is that several parameter values, such as gain,  $k$ , and impact range,  $\sigma$ , must be determined manually. The reaction of the SFM is frequently inappropriate for certain environmental circumstances as a result of this manual determination. In this paper, we propose employing the Fuzzy Inference System (FIS), whose rules are optimized using a Genetic Algorithm (GA) to manage the value of the gain,  $k$ , parameter adaptive. The relative distance,  $d$ , and relative angle,  $\alpha$ , concerning the robot's obstacle are the inputs for the FIS. The test results using a 3-D realistic CoppeliaSim demonstrated that the learning outcomes of FIS rules could provide adaptive parameter values suitable for each environmental circumstance, allowing the robot to travel smoothly is represented using the robot's heading deviation which decreasing by and reaching the goal 1.6 sec faster from the starting point to the goal, compared to the SFM with the fixed parameter value. So that the proposed method is more effective and promising when deploying on the real robot implementation.

**Keywords:** social force model, gain, impact range, fuzzy inference system, genetic algorithm

## I. INTRODUCTION

Human-robot interaction has become a focus of research during the last two decades. Along with the development of social necessity, the interaction between Humans and Robots has also widened, such as in guide robot application [1]–[4], companion robot application [5], and the person following robot application [6]. The development of robot behavior previously only considered aspects of safety and efficiency. With the increase in robot development, many human activities can be helped by robots. Moreover, the user of robots is also increasing, not limited to expert people only, but also people who do not know much about robot technology. Due to those conditions, the robot behavior must also consider aspects of comfort on the side of humans interacting with them. The behavior of robots that consider social aspects is necessary to help the interaction between humans and robots that potentially pose a threat. Comfort is a psychological aspect for humans means that there is no feeling of being intimidated or threatened by something and having enough space to do activities as if nothing is interfering with their activities. Thanks to Hall [7] for doing an excellent job of approaching by dividing proxemic areas around humans, taking into account social interaction spaces and norms. This approach has also been implemented with many robot applications [6], [8]. The robot navigation method uses this divided proxemics approach, called Social Force

Model (SFM) [9].

In the previous research [10], [11], the application of the SFM to a Differential Drive Mobile Robot (DDMR) was implemented. In that research, the social force model has been improved by implementing Q-Learning, called Q-learning based Social Force Guiding Model (QL-SFGM). Q-learning works by defining the robot as an agent and some parameters such as state, actions, state-transition, and reward. The problem is that data from the environment of the DDMR is continuous data, and the data required by the Q-learning is discrete. So, the data require to be sampled discretely. This approach requires many definitions of Q-learning parameters, state-action pairs, and consumes a high manual process resource. Another research is to improve the SFM using a Fuzzy Inference System (FIS) [12]. This research uses FIS to update parameters in SFM to give an adaptive behavior called the Fuzzy Social Force Model (FSFM). The parameter updated by FIS here is the gain of obstacle repulsive force. The input of the FIS is the obstacle distance and the obstacle coming angle. This research focuses on the control of an omnidirectional soccer mobile robot. The algorithm has been tested on 3D simulation software named V-REP using an omnidirectional robot model from Robotino. The result from the test shows that FSFM can work very well and achieve the goal by minimizing collision with other obstacles.

In this research, we implemented FIS to replace

Q-learning and adopt adaptive behavior on DDMR. The drawback of FIS is a subjective element when defining the rule base. We cannot objectively know whether or not the defined rule base has been defined very well. To objectively define the FIS rule base, we use a Genetic Algorithm (GA) as the optimizer to generate the rule base for FIS. In a research done by Guo [13], he proved that GA could improve the performance of his fuzzy controller by updating its rule base. This research is tested on the simulation software CoppeliaSim (formerly V-REP). The experiment shows that GA has improved with the result of the movement of the DDMR.

The rest of this paper is structured as follows. Section II contains a brief description of SFM and Section III explains the design of the system and the approach method. The experiment setup, result, and discussion are included in Section IV. In Section V, we wrap up this study with conclusions.

## II. LITERATURE REVIEW

In 1995, Helbing and Molnar [9] first introduced the SFM for modeling pedestrian dynamics. This model works by representing how humans interact and navigate with each other in a shared workspace as shown in Figure 1. The robot must be modeled as an object receiving force in Newton's law to implement the social force model. The first is goal force ( $F_g$ ), calculated by multiplying its mass and acceleration. Assume the acceleration is the difference between current speed ( $v_0$ ) and target speed ( $v$ ), so the goal force equation can be written as the follows [9]:

$$F_g = m \left( \frac{v - v_0}{\Delta t} \right). \quad (1)$$

The second force is the repulsive force ( $F_r$ ) of an object. Repulsive force ( $F_r$ ) can be modeled on two characteristics of force gain: social force ( $f_{soc}$ ) and physical force ( $f_{phy}$ ) than can be expressed as follows [9]:

$$F_r = f_{soc} + f_{phy}, \quad (2)$$

$$f_{soc} = k \cdot \exp\left(\frac{r_i - d_i}{\sigma}\right) \cdot \vec{e}_i, \quad (3)$$

$$f_{phy} = k \cdot \exp(r_i - d_i) \cdot \vec{e}_i, \quad (4)$$

where  $k$  is linear gain of the force,  $r_i$  is the radius of the robot to starting response,  $d_i$  is the distance between the robot with the obstacle,  $\sigma$  is the impact range of the force, and ( $\vec{e}_i$ ) is the vector direction of the working force. And then, the navigation force ( $F_{nav}$ ) is obtained by summing the goal force  $F_g$  and the repulsive force  $F_r$ . The navigation force ( $F_{nav}$ ) can be written as follows [9]:

$$F_{nav} = F_g + F_r. \quad (5)$$

The determination of the SFM parameters is crucial. Each parameter has its impact on the result response of the robot behavior. For example, for the gain of the repulsive force ( $k$ ), it is better to set it by a high value

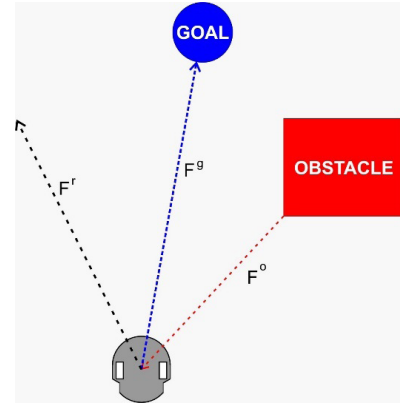


Figure 1. The forces that work within the SFM framework [10]

if the environment where the robot run is wide enough. However, if the environment is narrow and the gain of repulsive force ( $k$ ) is too high, the robot response will bounce toward the obstacle. It becomes a problem when the robot works in various conditions of the environment. So, an ability to update its parameters is required to make the SFM can work in various types of environments. Then, the FSFM will be employed in this research.

## III. METHOD

### A. Fuzzy Social Force Model

The FSFM uses FIS as the adaptive controller of the SFM parameter controller. There are three steps to employing the FIS, first is fuzzification which converts a crisp value into fuzzy data. In this work, the fuzzification converts two inputs: the obstacle distance ( $d$ ) and the obstacle coming angle ( $\alpha$ ). This process utilizes a triangular membership function since the system requires a proportional ratio of the input-output relation. The membership degree can be calculated as follows:

$$\mu = \begin{cases} \frac{x-a}{b-a} & \text{if } a < x \leq b \\ \frac{c-x}{c-b} & \text{if } b < x \leq c, \\ 0 & \text{if others} \end{cases}, \quad (6)$$

where  $\mu$  is the degree of the membership function,  $x$  is the crisp input, and  $a$ ,  $b$ , and  $c$  represent the membership function's left, center, and right boundaries, respectively.

Figures 2 and Figure 3 illustrate how the membership function of each input is divided. The relative distance, as shown in Figure 4, is separated into four sections to correspond to the proxemic distance suggested by Hall in [7], namely intimate, personal, social, and public. However, since the robot does not respond to the existence of an obstacle in public distance, then we ignored the public distance in our system. Our decision to ignore the public distance part is to simplify the computation. Additionally, the obstacle coming angle, as shown in Figure 5, is separated into three parts, front, side-front,

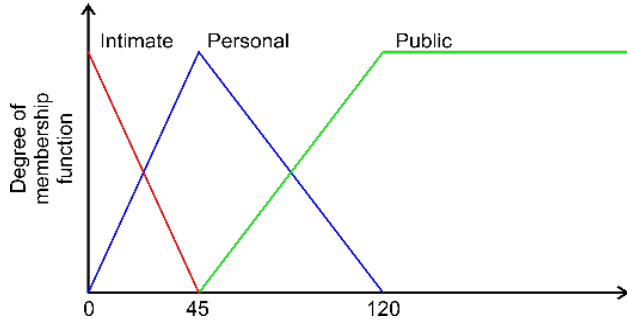


Figure 2. Membership function of obstacle distance input ( $d$ )

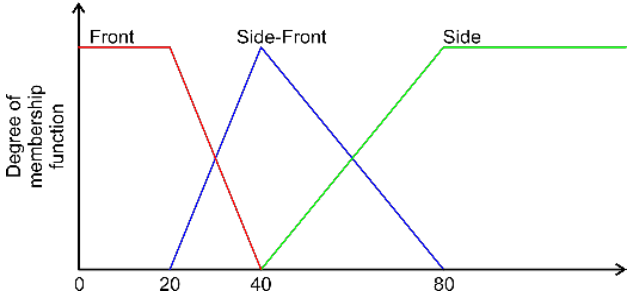


Figure 3. Membership function of obstacle coming angle input ( $\alpha$ )

and side, to simplify the computation further. Since we have two inputs, obstacle distance ( $d$ ) and obstacle angle of approach ( $\alpha$ ), the inference value can be determined using the following formula.

$$B = [\mu_m^d \mu_m^\alpha]_{m=1,2,\dots,M}, \quad (7)$$

where  $B$  denotes the T-Norms between  $\mu_m^d$  and  $\mu_m^\alpha$ . Thus,  $M$ -memberships exist for obstacle distance ( $d$ ) and obstacle coming angle ( $\alpha$ ), for a whole of  $F=M^2$  dimensions for which a fuzzy rule  $R$  must be determined. We defined our fuzzy rule,  $R = \{r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9\}$ . The design of the fuzzy rule is shown in Table 1.

Furthermore, the last step is defuzzification, which converts the inference system's fuzzy output into a crisp value using the Center of Area (CoA) equation as follows [14]:

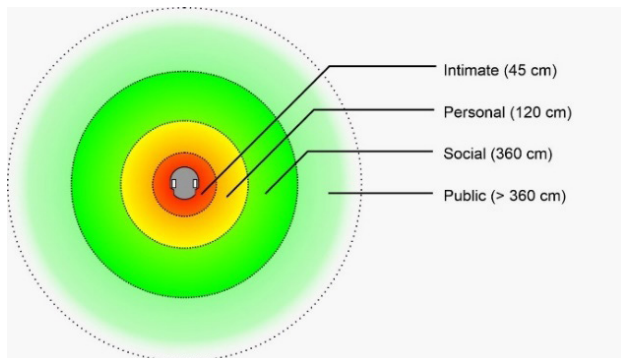


Figure 4. The proxemic distances proposed by Hall [7]

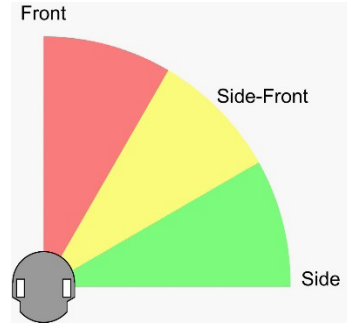


Figure 5. The division of relative direction of the obstacle

$$k = \frac{\sum_{n=1}^N B_n R_n}{\sum_{n=1}^N B_n}. \quad (8)$$

where  $N$  is the number of the inference  $B$  as well as the number of  $R$ , where  $k$  is the gain value of the obstacle's force.

### B. Genetic Algorithm Fuzzy Social Force Model

Determining a fuzzy rule base leads to the designer's subjectivity. So, it is hard to know whether it is already the most suitable rule base solution or not. The GA is an optimization algorithm developed by John Holland and his colleagues in response to Charles Darwin's evolutionary theory [15]. Natural selection and genetics provide the basis of GA. Previous results reinforce this creative use of random search to lead the search to an area of the solution space with superior performance. GA replicates the process of natural selection, demonstrating that a species able to adapt to environmental changes will live and reproduce to produce the next generation. GA, in plain terms, exemplifies "Survival of the Fittest" among problem-solving humans from consecutive generations. Each generation consists of population of individuals, each having a point in the search space, and the solution is executed. Each individual can be represented as a string of characters, integers, floats, or bits values. Chromosomes are the names given to these sequences.

In this research, we defined the fuzzy rule as the chromosome of an individual. First, some random individuals are initialized with a value of chromosomes chosen randomly. Then each will be judged by the fitness function. The system will run the simulated FSFM enforced robot in a defined simulated environment and accumulate the total time taken and heading error rate. The fitness function can be obtained by using the equation as follows:

Table 1. The design of the fuzzy rule

R	Obstacle coming angle ( $\gamma$ )	Obstacle distance (d)		
		Intimate	Personal	Social
	Front	$r_1$	$r_2$	$r_3$
	Side-front	$r_4$	$r_5$	$r_6$
	Side	$r_7$	$r_8$	$r_9$

$$f_{fit} = \sum_{i=1}^I \Delta t_i + \sum_{i=1}^I \Delta \gamma_i. \quad (8)$$

where  $f_{fit}$  is the fitness function,  $I$  is the total process iteration of the current fitness function calculation, is the time taken for one process iteration, and  $\gamma_i$  is the heading error of the root toward the goal direction.

The next step is crossover and mutation. Crossover is a process for determining a new solution represented by an individual. In this work, we use a uniform crossover, which randomly chooses some chromosomes of two individuals to be mixed to get a new individual. Then the uniform mutation is the process of getting a new value of chromosomes by randomizing some of its chromosomes. The algorithm is shown in the Pseudocode 1.

### C. Simulation

The GA is an iterative training algorithm with certain number of individuals. To deal with GA, our robot must be run under certain conditions many times to find the best individual as the solution. This is not possible if done using a real robot due to power supply problems. In addition, we also argue that in carrying out iterative learning, we must also ensure that the possibility of robot damage or robots injuring humans can be avoided. Therefore, to deal with those requirements, we used simulation software. There are many simulations software available nowadays, from advanced featured software to simple featured software, which provides features like physical engine, body 3D object modeling, scripting, and animation rendering. A thorough breakdown of robot simulation software is provided in [16].

A flexible simulation framework with an integrated development environment, CoppeliaSim or previously

---

#### Pseudocode 1. Genetic Algorithm

---

**Input:** Number of Fuzzy Rule,  $C$   
 Number of Population,  $N$   
 Number of Iteration, Iter

**Output:** Optimized Solution for Fuzzy Rule

**Initialization:** Generate  $N$  sets of Fuzzy Rule  
 $R_n = \{R_1, R_2, \dots, R_c\}$ ,  $n = \{1, 2, \dots, N\}$

**Begin**

1. **for**  $i = 1$  to Iter **step 1 do**
  2. **Evaluation:** sorting the population by Fitness value using eq. (8), from lowest to highest
  3. **Reproduction:**  $R_1 \leftarrow R$  with lowest fitness value,  $R_2 \leftarrow R$  which randomly chosen from the population
  4. Crossover between  $R_1$  and  $R_2$
  5. New composition population  $R_n$  is formed and replaced two of the 2 worst solutions in the population
  6. Mutation one of randomly chosen from the population
  7. **end for**
  8. The most optimized fuzzy rule solution  $R$  is obtained after Iter step
- 

known as V-REP comes with a large library of models and sample scripts for working mobile and fixed robots. It is employed in the creation of algorithms, the simulation of factory automation, remote monitoring, and instruction. The usage of embedded scripts, which may be attached to specific objects in the scene like bodies, joints, or sensors [17], is the key distinguishing characteristic. This feature makes it easier to simulate a robot's behavior, as will be shown in the next sections. Another benefit is the ability to communicate with other clients (such as Python, Matlab, ROS, or IoT apps), which enables a seamless transition to actual hardware.

Because the object of the optimization is DDMR, there is also another advantage to using CoppeliaSim, which is there is an already defined Pioneer3DX robot that can be used as shown in Figure 6. Besides the defined robot, we also defined a simulated environment which can be called a scene in CoppeliaSim. The simulation environment can be seen in Figure 7. Our basis for determining the simulation environment model is that the shape of the environment where the indoor environment with winding shapes, walls, and many other objects is more difficult for the robot to navigate. So, we hope the results can be generalized to any condition. The process of GA optimization and testing has been done in a computer.

## IV. RESULTS AND DISCUSSION

In our experiments, we used a desktop PC equipped with specification as follows. CPU Intel Core i7 with 3.6 GHz x 8, 4 GB of RAM, Linux Ubuntu 20.04, python, C++, and ROS Noetic. For the simulation hardware, we use a Pioneer3DX robot equipped with Hokuyo Lidar. We also defined a simulation environment in CoppeliaSim as shown in Figure 7. The GA training is set to 500 iterations with a population consists of 20 individuals. The enhancement of the fitness value during the training



Figure 6. Defined Pioneer3DX in the CoppeliaSim



Figure 7. Defined simulated environment in the CoppeliaSim



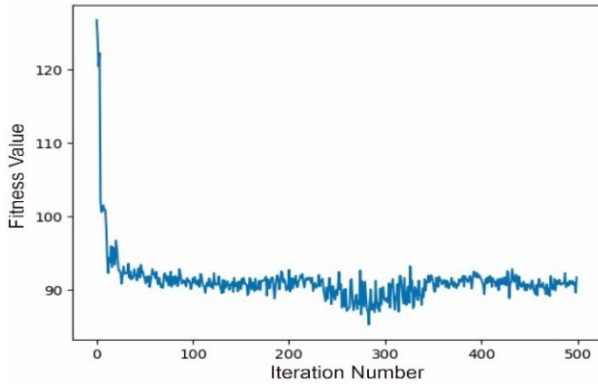


Figure 8. The enhancement of the fitness value during 500 iterations

process is shown in Figures 8. Furthermore, Figure 9 shows the movement trajectories of the robot during training.

From the graph shown in Figure 8, the GA has proved that it can optimize the value of its movement efficiency by calculating the time taken and the error heading rate as the fitness function. The fitness value has already been steady on around 100 iterations. The result of the optimized rule base can be seen in Table 2. As the comparison, we also test the FSFM algorithm without the GA optimization, so the rule base of the fuzzy is tuned manually using human intuitive as shown in Table 3.

As we see in Table 2, the rule shows that the value around Front area has the largest value, followed by Side-Front area, and then the one which has smallest value is around Side area. In the Front area, the closer our robot to the obstacle, the greater the rule value obtained. But, in around Side area, the rule value has the opposite characteristic. The closer to the obstacle, the smaller the rule value obtained. This anomaly happens because, during the training process in the defined simulation environment, not all rule base conditions occur with the same frequency. So, some rule base values affect the result of the robot's movement. However, some are not giving effect too much.

We also tested both the optimized rule base and the manually tuned rule base in the same environment. The result shows that GA optimization has improved the performance of the FSFM by minimizing the robot

Table 2. The optimized value of the fuzzy rule

R		Obstacle distance (d)		
		Intimate	Personal	Social
Obstacle coming angle ( $\gamma$ )	Front	393	301	164
	Side-front	52	199	64
	Side	7	13	60

Table 3. The manual defined value of the fuzzy rule

R		Obstacle distance (d)		
		Intimate	Personal	Social
Obstacle coming angle ( $\gamma$ )	Front	300	250	150
	Side-front	200	150	125
	Side	40	20	10

Table 4. Comparison table of optimized and manually tuned rule base

Parameters	Optimized rule	Manually tuned
Time taken (s)	39.3411	40.9209
Mean square error of the heading (o)	52.9577	58.2952

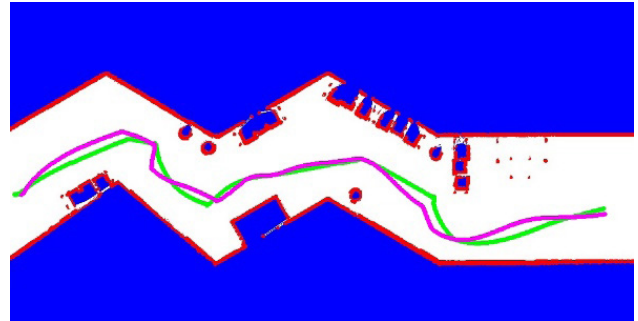


Figure 10. The comparison between manual defined value of the rule base and our proposed algorithm

error heading toward the goal direction. The comparison data between our optimized FSFM and the manually tuned FSFM is shown in Table 4. It can be seen that the optimized version achieves better results by providing faster result in arriving at the goal position and a smaller number of heading bounces during completing the task. Furthermore, the comparison of the trajectory is shown in Figure 10. The purple line indicates the robot trajectory with manual defined value. The green line indicates the robot trajectory when implementing our algorithm. From Figure 10, the trajectory of optimized version is smoother than the manual version. However, both versions can reach the goal safely.

## V. CONCLUSION

We have developed an adaptive control strategy for the SFM gain parameter value,  $k$ , since it greatly affects the repulsive forces of the SFM. Controlling this parameter will significantly affect to the robot's behaviour. We used FIS to dynamically change the gain parameter value, where its rule was optimized using GA. We proved the performance of our proposed algorithm using a simulation that was conducted using the CoppeliaSim for DDMR. The controller successfully drove the DDMR pass through the obstacles in a simulated environment toward its goal. We compared the manually tuned fuzzy rule values and the optimized fuzzy rule values by the GA. Two factors were evaluated to see the algorithm's performances, namely time needed for the robot to reach the goal and the robot's heading deviation to evaluate the robot's behaviour. As the results, the proposed FSFM with the GA optimization outperforms the manually tuned FSFM by achieving 1.6 seconds faster to run the robot from start to goal, and the robot heading has reduced by about  $5.3^\circ$ . The implementation of our algorithm to the real robot will be our focus for the next plan.

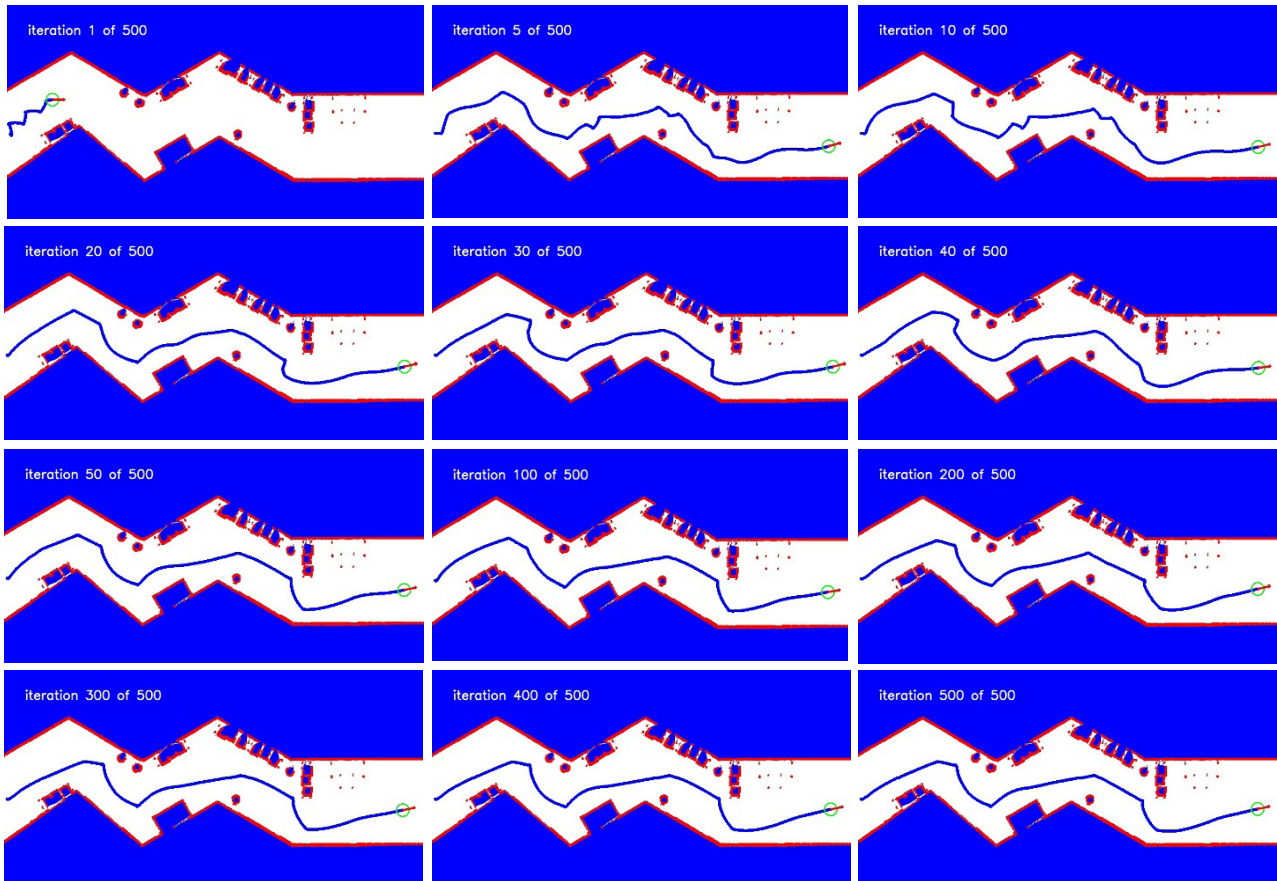


Figure 9. Robot trajectory during 500 iterations

#### ACKNOWLEDGMENT

We would like to thank to the Directorate of Vocational Higher Education for supporting our research fund with a contract No. 093/SPK/D4/PPK/01/APTV/VI/2022. We also thank to all members of the Signal, Vision and Graphics, and all members of the Social Robotics and Smart Systems Applications Research Group for their valuable support and help us in finishing our work.

#### REFERENCES

- [1] J. Lai, H. Lei, S. Bao, L. Du, J. Yuan, and S. Ma, "Design of a portable indoor guide robot for blind people," in *Proc. 6th IEEE International Conference on Advanced Robotics and Mechatronics*, July 2021, pp. 592–597.
- [2] A. T. Rifqi, B. S. B. Dewantara, D. Pramadihanto, and B. S. Marta, "Fuzzy social force model for healthcare robot navigation and obstacle avoidance," in *Proc. Inter. Electronics Symp.*, Sep. 2021, pp. 445–450.
- [3] X. T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 4, pp. 1743–1760, Oct. 2017.
- [4] C. Wang, Y. Li, S. S. Ge, and T. H. Lee, "Adaptive control for robot navigation in human environments based on social force model," in *Proc. IEEE Inter. Conf. on Robotics and Automation*, May 2016, pp. 5690–5695.
- [5] G. Ferrer, A. Garrell, and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," in *Proc. IEEE Inter. Conf. on Intelligent Robots and Systems*, Nov. 2013, pp. 1688–1694.
- [6] J. Satake, M. Chiba, and J. Miura, "Visual person identification using a distance-dependent appearance model for a person following robot," *Int. J. Autom. Comput.*, vol. 10, no. 5, pp. 438–446, Oct. 2013.
- [7] E. T. Hall, *Handbook for Proxemic Research*, Washington D.C., USA: Society for the Anthropology of Visual Communication, 1974.
- [8] P. Patompok, S. Jeong, N. Y. Chong, and I. Nilkhamhang, "Mobile robot navigation for human-robot social interaction," in *Proc. 16th Inter. Conf. on Control, Automation and Systems*, Oct. 2016, pp. 1298–1303.
- [9] D. Helbing and P. Molnár, "Social force model for pedestrian dynamics," *Phys. Rev. E*, vol. 51, no. 5, pp. 4282–4286, May 1995.
- [10] B. S. B. Dewantara, "Building a socially acceptable navigation and behavior of a mobile robot using Q-learning," in *Proc. Inter. Conf. on Knowledge Creation and Intelligent Computing*, Nov. 2016, pp. 88–93.
- [11] B. S. B. Dewantara and J. Miura, "Generation of a socially aware behavior of a guide robot using reinforcement learning," in *Proc. Inter. Electronics Symp.*, Sep. 2016, pp. 105–110.
- [12] B. S. B. Dewantara and B. N. D. Ariyadi, "Adaptive behavior control for robot soccer navigation using fuzzy-based social force model," *Smart Sci.*, vol. 9, no. 1, pp. 14–29, Jan. 2021.
- [13] L. Guo and C. Zheng, "Optimization of fuzzy sliding mode controller with improved genetic algorithm," in *Proc. 22nd Inter. Conf. on Electrical Machines and Systems*, Aug. 2019, pp. 1–5.
- [14] A. V. Patel and B. M. Mohan, "Some numerical aspects of center

- of area defuzzification method,” *Fuzzy Sets and Syst.*, vol. 132, no. 3, pp. 401-409, Dec. 2002.
- [15] X. S. Yang, *Nature-Inspired Optimization Algorithms*, 1st edition. Boston, USA: Elsevier, 2014.
- [16] Y. Rizal, *Computer Simulation of Human-Robot Collaboration in The Context of Industry Revolution 4.0*, in: *Becoming Human with Humanoid-From Physical Interaction to Social Intelligence*, London, United Kingdom: IntechOpen, 2019.
- [17] E. Rohmer, S. P. N. Singh, and M. Freese, “V-REP: A versatile and scalable robot simulation framework,” in *Proc. IEEE Inter. Conf. on Intelligent Robots and Systems*, Nov. 2013, pp. 1321–1326.