ELKHA : Jurnal Teknik Elektro, Vol. 14 No.2, October 2022, pp. 67 - 72
ISSN: 1858-1463 (print), 2580-6807 (online)

Multi Sensor-Based Obstacle Avoidance Algorithm in Visual Engineering Environment

Abqori Aula

Department of Electrical Engineering, Universitas Tanjungpura, Indonesia
Corresponding Email: abqoriaula@ee.untan.ac.id

Abstract – **Obstacle avoidance is an essential problem for applications involving multiple wheeled mobile robots. This research proposes a simple obstacle avoidance rule utilizing only one type of sensor, i.e., infrared sensor. In this research, multiple infrared sensors are placed on a mobile robot, arranged 45° radially equidistance. By using a low-cost and easily available infrared sensor, the cost and time consumed to build and repair a wheeled mobile robot are considerably reduced. Avoiding rules, based on simple behavior, such as "turn", "stop", "follow", and "ignore" are developed. By applying these rules, each robot can refer to the motion of other robot or stationary object to avoid collision. Simulation results show that the proposed algorithm performs well, at 66.7% chance of avoiding a moving object and at 93% chance of avoiding a stationary object.**

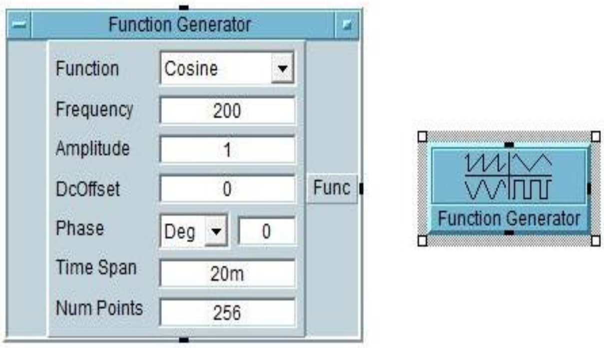
Keywords: *collision avoidance, multi sensor, visual engineering environment, wheeled mobile robots*

I. INTRODUCTION

The term of multiple robots refers to a large number of relatively small-sized wheeled robots. Each robot has a limited sensing, motion, and perception capability. This kind of robot is economically feasible to be developed compared to a large and complex robot [1]. However, controlling many multiple wheeled robots brings up a challenge, namely collision avoidance [2]. Therefore, many researches have been reported to console this challenge. Some researches dwell in the type of sensing used to avoid an obstacle [3]-[4], others focused on the control system [5]-[8]. Many researches used ultrasonic sensor with a variety of configurations, such as using only one sensor [9], three sensors [10]-[12], four sensors [13] or even eight sensors [14]. Some researches even utilized a more complex algorithm of sensor data [15]-[16], an approach based on information fusion [17]. Though its range is much shorter than sound-based ranging sensor, infrared sensor triumphs in resolution and higher distance measurement rate [18]-[19]. Using a group of infrared sensors can provide an effective full coverage of the robot's path at a cheaper price than laser-based and radar-based sensors [20]-[21].

In this paper, the author proposes a simple algorithm for collision avoidance by processing data from multiple infrared sensors. Constructed using simple and cheap infrared sensors and largely available microcontroller board, the robot in this research is considerably cost-saving and comparatively cheap to be used in robotic research.

The proposed obstacle avoidance algorithm is then simulated in a program built using Visual Engineering Environment (VEE) software [22] by Agilent (now owned by Keysight). VEE software is, first and foremost, not a simulation software, but is optimized for use with electronic instruments [23]. However, its graphical programming feature can be used to build user-specific graphical user interface [24]. Hence in this paper, using simple and intuitive VEE blocks, an arena is constructed where the robots with embedded avoidance algorithm are placed and the performance of the algorithm is simulated. The version used in this research is Agilent VEE Student 8.5. Figure 1 shows a part of visual programming within the environment of VEE software, in which a function generator block is visible in its expanded and minimized view.



(a) (b)

Figure 1. A function generator block in VEE application, in (a) expanded view and (b) minimized view.

The rest of this paper is structured as follows. In Methodology section, how the program was designed, how the avoidance algorithm was built and how the simulation was programmed are elaborated. Next, the simulation setup, the result of the simulation, some screenshots of simulations and the analysis of the result are shown in section III. This paper is concluded in section IV.

Manuscript received 2021-10-19; revised 2022-07-11; accepted 2022-07-20

II. METHODOLOGY

A. Program Design

This project proposes a scenario for obstacle avoidance for multiple wheeled mobile robots. Each mobile robot is fitted with eight flying fish type infrared sensor modules, mounted radially on it, to provide a visional coverage of 360 degrees at a time. Figure 2 show the configuration of the proposed sensory system for this work. Each robot is applied with an obstacle avoidance algorithm.

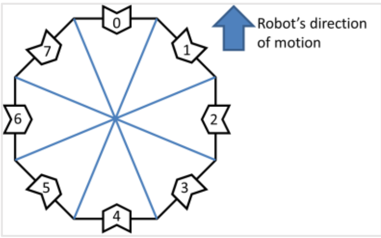


Figure 2. Configuration of sensors

The sensors are given a simple identification (ID) number, which is 0,1,2 through 7, starting from the front-facing robot.

This program works as follows. Firstly, it initializing the simulation. This includes calling global settings (width and height of the simulation window, number of sensor, distance between sensors, sensor IDs, and size of the robot), constructing the arena, reading user inputs (number of robot, robot's speed, and sensor range), building the simulated robots (or simBot), reading the given goal, and placing them randomly onto the created arena. These steps are grouped within a block called "InitRobotEnv" and "InitRobot". Then, each simBot is ordered to move forward from their initial random position in a straight motion. If it detects an object, it checks the sensor ID, then apply the proposed behavior. These steps are then illustrated by a flowchart as shown in Figure 3.

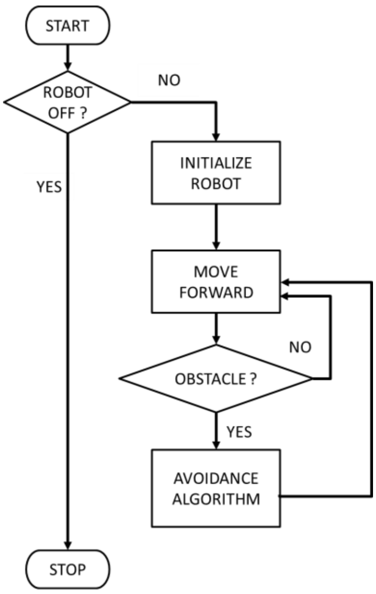


Figure 3. Program's flowchart

Figure 4 shows the corresponding block diagram visualized in Agilent VEE software.

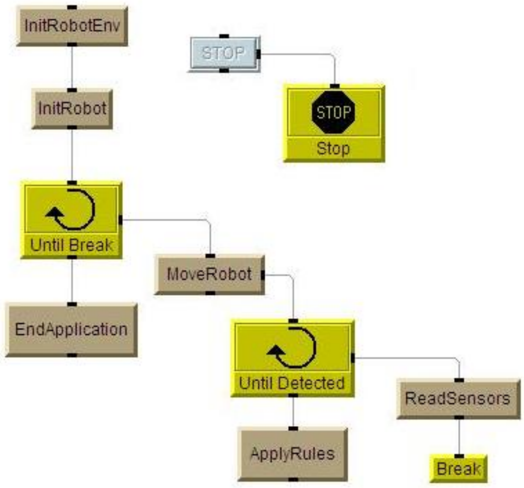


Figure 4. The steps visualized in blocks using Agilent VEE software

B. Obstacle Detection

The position of an obstacle (*q*, whether a robot or an object) is determined by calculating the robot's current position coordinate (*x*), distance sensed by infrared sensor (*d*), and the ID of the sensor (*n*), such that:

$$q(x,d,n) \tag{1}$$

The proposed avoidance algorithm is made up of four simple behaviors, which are "turn", "stop", "follow", and "ignore". The behavior is summarized as follow:

1. If an ID is sensed, then
   a. If the sensed ID is the same as sensing ID or near to the sensing ID, then it means a frontal collision is imminent, then both simBot must turn left;
   b. If the sensed ID is perpendicular to the sensing ID, then the simBot with smaller ID must stop;
   c. If the sensed ID is a rear-located ID, then both simBot should ignore and keep moving forward;
2. If no ID is sensed, then
   a. If distance is sensed, then a stationary object is detected, then simBot must turn away;
   b. If no distance is sensed, then simBot should ignore and keep moving forward.

The behaviors in point 1 are mapped in Table 1.

Table 1. Map of behavior for each sensing pair

		Sensed ID							
		0	1	2	3	4	5	6	7
Sensing ID	0	I	I	S	S	T	S	S	I
	1	I	I	S	S	S	T	I	I
	2	I	I	I	S	S	I	I	I
	3	I	I	I	I	I	I	I	I
	4	I	I	I	I	I	I	I	I
	5	I	I	I	I	I	I	I	I
	6	I	I	I	I	S	S	I	I
	7	I	I	I	T	S	S	S	I

Note that T means Turn Left, S means Stop, and I means Ignore. The behaviors for sensing stationary object are mapped in Table 2.

Table 2. Map of behavior for sensing stationary object

&	0	1	2	3	4	5	6	7
0	L/R	L	-	-	-	-	-	R
1	L	L	L	-	-	-	-	-
2	-	L	I	I	-	-	-	-
3	-	-	I	I	I	-	-	-
4	-	-	-	I	I	I	-	-
5	-	-	-	-	I	I	I	-
6	-	-	-	-	-	I	I	R
7	R	-	-	-	-	-	R	R

The symbol '&' indicates that both sensor ID are detecting the object. Note that L means turn left and R means turn right. These behaviors are then converted into algorithm, then converted into appropriate programming blocks.

C. *Visual Programming*

The main program made using Agilent VEE software's blocks is shown in Figure 5, while Figure 6 and 7 show the program blocks for sensing module and position tracking module, respectively.

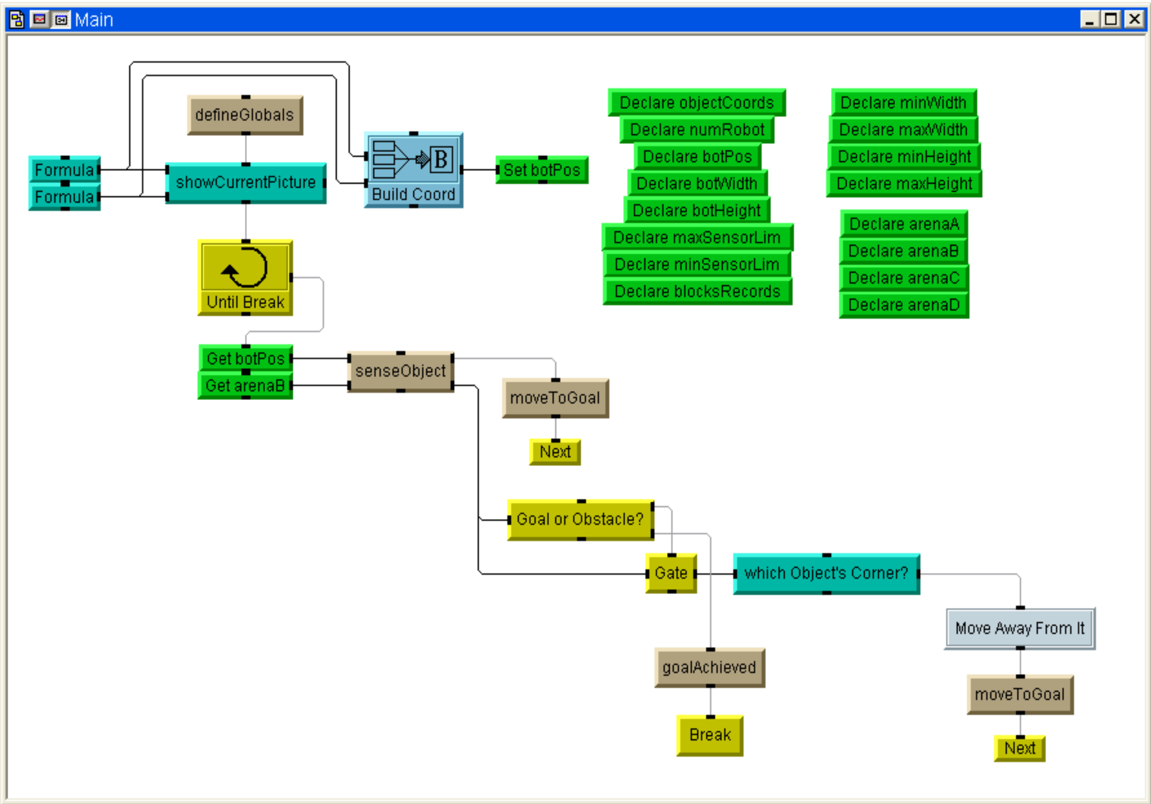


Figure 5. Agilent VEE's program block for simulating the proposed avoidance algorithm

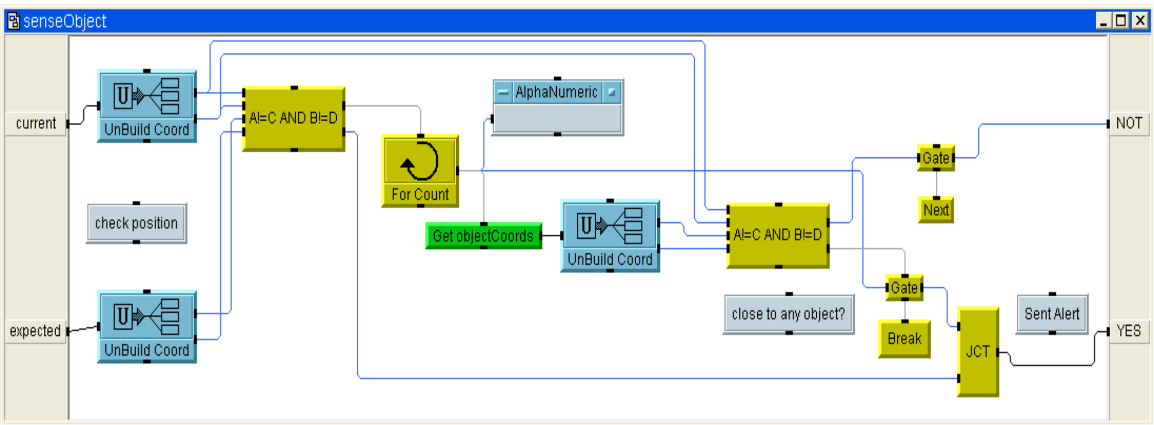


Figure 6. Agilent VEE's program block for sensing module

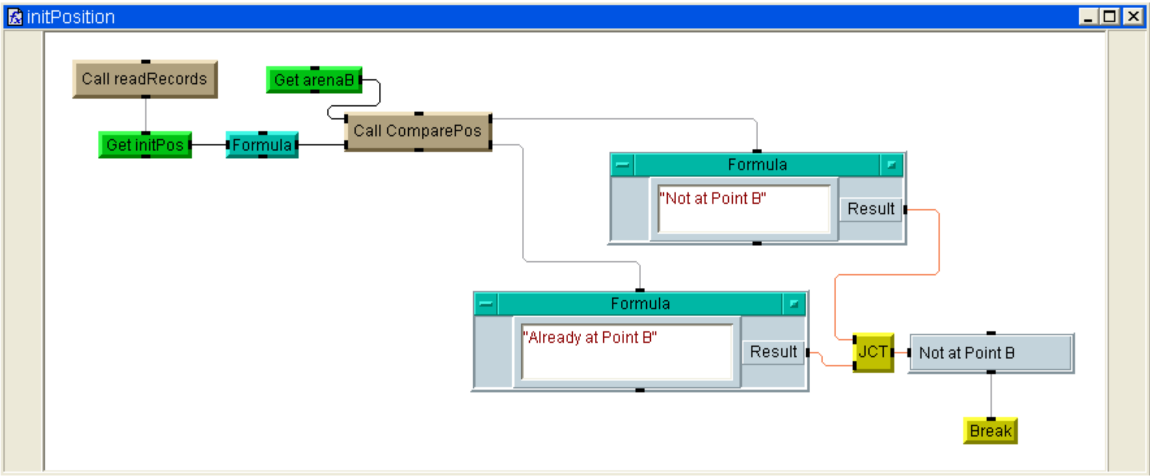


Figure 7. Agilent VEE’s program block for tracking simBot's position

III. Results and Discussion

A. Simulation Setup

The designed program and avoidance algorithm are then validated through a series of simulations using the Agilent VEE software. The computer used for running the simulation has the following specification:

(1) Windows XP

(2) Processor: 64 bit Dual Core 2.1 GHz;

(3) Graphics: ATI Radeon HD3450, memory 256 MB;

(4) RAM: 2 GB.

The following settings are used during all of the simulations:

(1) the speed of motion for the simBot and the sensor range are set at 1du/tu and 10 du, respectively (du = distant unit, tu = time unit);
(2) turning radius is set a 45°; and
(3) four simBots for each simulation.

The following conditions are considered as a collision ONLY if the simBot:

1. Does not turn away from obstacle;
2. Pass through the obstacle or other simBot.

The models for a simBot and an obstacle are shown in Figure 8:

(a) (b)

Figure 8. Model for (a) simBot and (b) stationary object

B. Analysis

The simulation was conducted 30 times. Each takes approximately 2 minutes of runtime. Figure 9 shows the simulation window. Figure 10 and 11 show the “late escape” phenomenon, in which it is not considered as a collision. Figure 12 shows a pass through motion, which is considered as a collision.

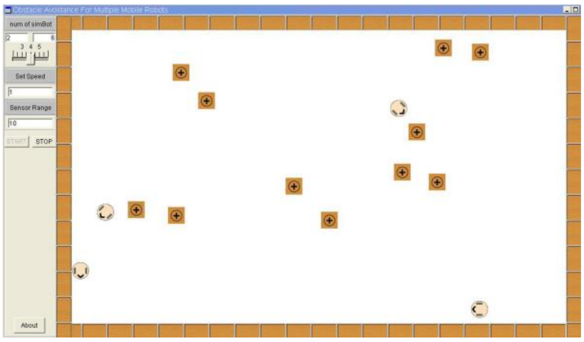


Figure 9. Simulation process

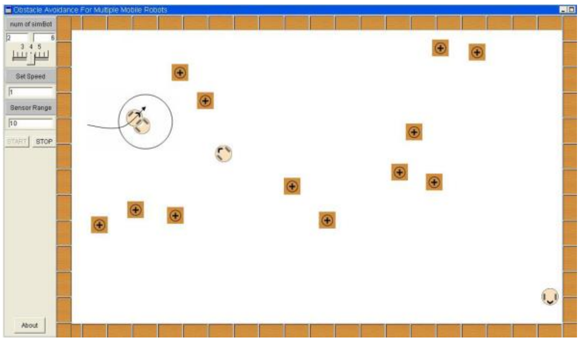


Figure 10. A situation considered as a late escape

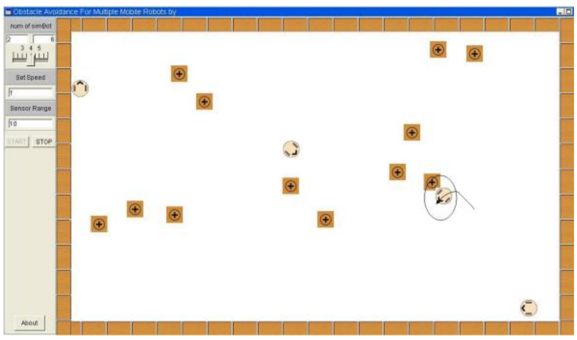


Figure 11. Another late escape

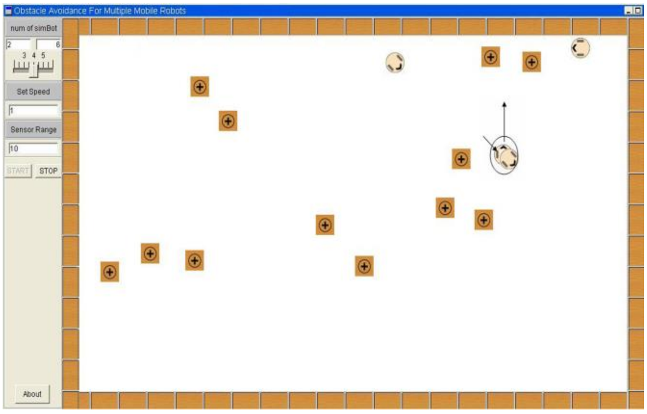


Figure 12. simBot failed to avoid another simBot, but instead passed straight through it

Figure 13 and Figure 14 exhibit simBot's avoidance behaviors. Figure 13 shows a simBot is passing the wall to avoid an object, while Figure 14 shows a well-avoiding movement of the simBot in one of the simulations. Table 3 summarizes the result of each simulation.

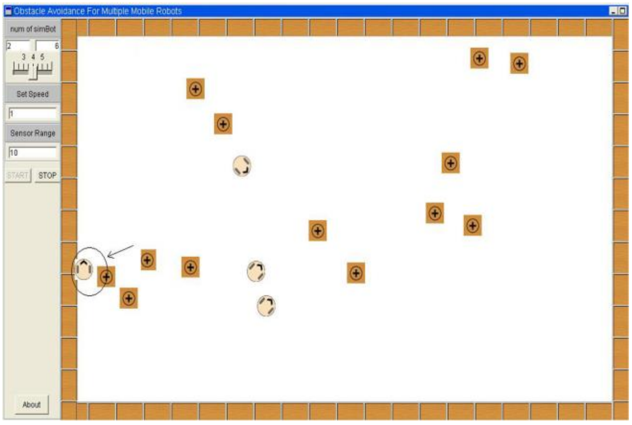


Figure 13. SimBot decided to crash onto the wall to avoid an obstacle

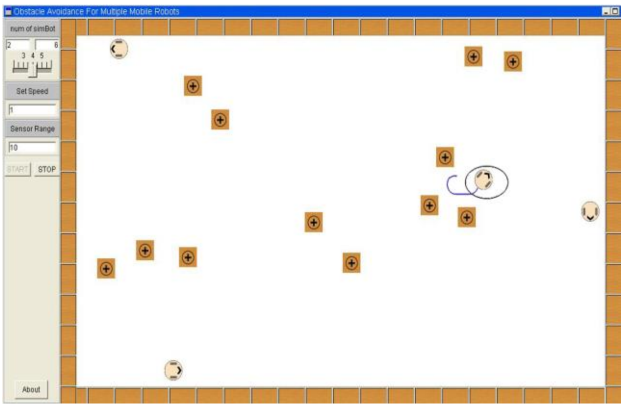


Figure 14. A simBot perfectly avoided a collision in a difficult position

Table 3. Simulation result

#	Collision	
	Obstacle	Other simBot
1	0	0
2	1	1
3	0	0
4	0	1
5	0	0
6	0	1
7	0	1
8	0	0
9	0	1
10	0	1
11	0	0
12	1	0
13	0	0
14	0	1
15	0	0
16	0	1
17	0	0
18	0	0
19	0	0
20	0	0
21	0	1
22	0	0
23	0	0
24	0	1
25	0	0
26	0	0
27	0	0
28	0	0
29	0	0
30	0	0
Sum	2	10

C. *Analysis*

The speed of the simBot was chosen as described in simulation setup, because if the speed value was set too fast, the simBot might turn away from an obstacle after 'touching' it or the simBot has to move pass through it because the sensor was too late to detect it.

The success rate p is calculated as ratio of number of collision r and total number of simulation R, such that:

$$p = \frac{r}{R} \times 100\% \qquad (2)$$

The simulation result in Table 3 shows that 20 out of 30 simulations (at 66.7% probability) were successful of avoiding a moving object. It also shows that 28 out of 30 simulations (93% success rate) were able to avoid a stationary object.

After further debugging and analysis, it is deduced that if a collision happens, it is not because of the code nor the rules nor the software, but merely and mostly because of the way the proposed algorithms are coded into the software. Around 70% of the code's weight was used on keeping the simBots within the given simulation arena while at the same time avoiding any obstacle or other simBot. This approach causes the simulation runs slower if any of the simBot is on the wall.

IV. Conclusion

This paper has elucidated the design and modelling of multi-sensor based obstacle avoidance algorithm. Eight infrared sensors are placed on a mobile robot, arranged 45° radially equidistance. Using the simulated features of an infrared sensor, avoidance rules, based on simple behavior, such as “turn”, “stop”, “follow”, and “ignore” were developed. The simulation result shows that 20 out of 30 simulations (at 66.7% probability) were successful of avoiding a moving object. The result also shows that 28 out of 30 simulations (at 93% probability) were able to avoid a stationary object. These results conclude that the robots which use the proposed algorithm are able to avoid any collision with moving or stationary obstacles.

References

[1] E. Sahin, "Swarm robotics: From sources of inspiration to domains of application”, *Swarm Robot.*, pp. 10-20, 2004.

[2] Y. F. Zhu and X. M. Tang, "Overview of swarm intelligence”, in *International Conference on Computer Application and System Modeling*, vol. 9, 2010.

[3] E. D. Marindani, "Robot Mobile Penghindar Halangan (Avoider Mobile Robot) Berbasis Mikrokontroler AT89S51", *Jurnal ELKHA*, vol. 3, no. 2, pp. 13-19, Juli 2011.

[4] S. K. Indrawan, F. Hadary, and A. Hartoyo, "Rancang Bangun Robot Terbang Penghindar Halangan Dalam Ruangan Bersekat", *Jurnal ELKHA*, vol. 10, no. 2, pp. 82-87, Oktober 2018.

[5] Rendyansyah et al., "Sistem Navigasi Robot Hexapod Menggunakan Behavior Dan Learning Vector Quantization ", *Jurnal ELKHA*, vol. 11, no. 1, pp. 33-38, April 2019.

[6] A. H. Shallal, O. N. Ucan, and O. Bayat, "Formation Control and Obstacle Avoidance in Swarm Robots", *International Journal of Computer Science and Network Security*, vol. 18, no. 3, pp. 126-133, March 2018.

[7] Y. Yu, X. Chen, Z. Lu, F. Li, and B. Zhang, "Obstacle avoidance behavior of swarm robots based on aggregation and disaggregation method", *Simulation: Transactions of the Society for Modeling and Simulation International*, vol. 93, no. II, pp. 885-898, 2017.

[8] S. M. H. Rostami, A. K. Sangalah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm”, EURASIP *Journal on Wireless Communications and Networking*, vol. 70, 2019.

[9] E. D. Widianto, U. Alfianto, and R. R. Isnanto, "Robot Beroda Perambat Dinding Berbasis Mikrokontroler ATmega 2560 Dilengkapi Kendali Nirkabel dan Penghindar Rintangan", *Jurnal Teknologi dan Sistem Komputer*, vol. 5, no. 2, pp. 49-56, 2017.

[10] R. Y. Endra et al., "Otomatisasi Navigasi penghindar Obstacle pada Mobile Robot dengan Metode Fuzzy Sugeno dan Mikrokontroler Arduino", EXPLORE: *Jurnal Sistem Informasi dan Telematika*, vol. 11, no. 2, pp. 110-117, Desember 2020.

[11] S. Sembiring and K. Exaudi, "PERANCANGAN ROBOT KAPAL DENGAN PERILAKU MENGHINDARI RINTANGAN", in *Konferensi Nasional Teknologi Informasi dan Aplikasinya* / KNTIA, 2017, pp. 117 -124.

[12] A. Suherman, "Robot pencari api dan penghindar rintangan", *Telekontran*, vol. 3, no. 2, pp. 37-46, October 2015.

[13] B. Rahmat, I G. S. M. Diyasa, and M. M. Munir. "Craft Enterprise Robot (CER) Berbasis Arduino", in *Seminar SANTIKA*, Jawa Timur, 2019, pp. 144-149.

[14] S. Gustanto, S. Sugeng, A. H. Paronda, and M. Nurjaya. "Kinerja Algoritma Vector Field Histogram Dalam Perilaku Menghindari Rintangan Pada Robot Sepak Bola Beroda", in *Seminar Nasional Teknik Elektro UIN Sunan Gunung Djati* / SENTER, Bandung, 2019, pp. 220-233.

[15] L. Duan and F. Tao. "Application of Multi-sensor Data Fusion Technology in Wheelbarrow Obstacle Avoidance System”, *Journal of Engineering Mechanics and Machinery*, vol. 2, no. I, pp. 27-34, 2017.

[16] H. Nascimento, M. Benoussaad, and M. Mujica. "Collision Avoidance Interaction Between Human and a Hidden Robot Based on Kinect and Robot Data Fusion", IEEE *Robotics and Automation Letters*, vol. 6, no. 1, pp. 88-94, January 2021.

[17] H. F. Durrant-Whyte. "Sensor Models and Multisensor Integration", *The International Journal of Robotics Research*, vol. 7, no. 6, pp. 97-113, December 1988.

[18] R. Rambabu, M. R. Bahiki and S. Azrad, “Multi-sensor Fusion Based UAV Collision Avoidance System,” *Jurnal Teknologi*, vol. 76, no. 8, pp. 89-93, 2015.

[19] Y. Yu, “Collision Avoidance Technology of Intelligent Vehicle System Based on Multi-Sensor Fusion,” *International Journal of Mechatronics and Applied Mechanics*, vol. 1, no. 6, pp. 225-235, 2019.

[20] I. Arrouch, J. M. Saleh and P. A. N. S. Goh, “A Comparative Study of Artificial Neural Network Approach for Autonomous Robot's TTC Prediction,” *Int. J. Mech. Eng. Rob. Res.*, vol. 11, no. 5, pp. 345-350, 2022.

[21] C. Perra, A. Kumar, M. Losito, P. Pirino, M. Moradpour and G. Gatto, “Monitoring Indoor People Presence in Buildings Using Low-Cost Infrared Sensor Array in Doorways,” *Sensors*, vol. 21, no. 4602, 2021.

[22] R.B. Angus and T. E. Hulbert. *VEE Pro: Practical Graphical Programming*. United States of America: Springer-Verlag, 2005.

[23] V. Srivastava, K. Yadav and G. Singh, “Measurement Process of MOSFET Device Parameters with VEE Pro Software for DP4T RF Switch,” *International Journal of Communications, Network and System Sciences*, vol. 4, no. 9, pp. 590-600, 2011.

[24] A. P. Mulmule, R. A. Vatti and P. M. Porwal, “MPPT Technique To Improve Efficiency In Wind-Solar,” *International Journal Of Electrical Engineering & Technology*, vol. 4, no. 6, pp. 74-82, 2013.