Autonomous Surface Vehicle (ASV) Obstacle Avoidance Using Fuzzy Kohonen Network (FKN)

by Rossi Passarella

PAPER · OPEN ACCESS

Autonomous Surface Vehicle (ASV) Obstacle Avoidance Using Fuzzy Kohonen Network (FKN)

View the article online for updates and enhancements.

Autonomous Surface Vehicle (ASV) Obstacle Avoidance Using Fuzzy Kohonen Network (FKN)

R Passarella[1], **A Zarkasi**[1], **H Maghfur**[1], **Sutarno**[1], **K Exsaudi**[1], **APP Prasetyo**[1], **H Veny** [2]

[1]Department of Computer Engineering, Faculty of Computer Science, Universitas Sriwijaya, Palembang, Indonesia
[2] Faculty of Chemical Engineering, Universiti Teknologi MARA (UiTM), Shahalam, Malaysia

E-mail: passarella.rossi@gmail.com

Abstract. Autonomous Surface Vehicle (ASV) is robot boats which can navigate autonomously to avoid obstacles in their path direction. The ASV has designed to detect and measure the distance of the position of the obstacle. The Fuzzy Kohonen Network (FKN) method is applying to the ASV as its brain to divine the manoeuvre what should do. The FKN is getting the information (Crips) from two sonar sensors, where are located in front of the ASV. In this experiment the FKN has four (4) pattern scenario which is three pattern normal condition, and one pattern danger condition. The three normal conditions define as if no obstacle detecting by two sensors, or either one of the sensor has detected an obstacle. The last condition pattern (danger) will occur when each sensor has detected the obstacle. These pattern condition is coding in C# and embedded into ATMega328. The range sensor is setting for 50-210 cm with the error rate is less than 1%. The manoeuvre of this ASV is providing two DC motors by controlling the PWM value. As the results of the experiment,The manoeuvre is quicker and smooth without a crash to obstacles.

1. Introduction

The navigation system is a system that determines the direction and purpose to accomplish. This system nowadays is familiar to use in the area of the Autonomous Navigation System (ANS)[1][2][3] . The implementation of this system can see in several vehicles, such as Unmanned Aerial Vehicles (UAV)[4][5] and Unmanned Surface Vehicles (USV)[6][7]. The ANS has integrated sensors and microcontroller or processor that enable autonomous navigation, path-planning and perception. The manoeuvre of this ANS is capable of sensing the environment and move with no human interruption.

The part of ANS is an Autonomous Surface Vehicle (ASV). The ASV is robot boats which can navigate autonomously to avoid obstacles in their path direction. The ASV for this research has a design with two DC motor as propulsion and microcontroller ATMega328 (`Figure 1`). Besides that, the ASV has two sonar sensors (MB7060 XL-Max Sonar-WR1). These sensors have a function to detect and measure the distance of the obstacle in front of the ASV.

The data measured from the sensors then sent to Arduino UNO rev3 with ATMega328 as its CHIP. Inside this module, the algorithm of Fuzzy Kohonen Network (FKN) is coding and embedding into it. The FKN method chosen due to the ability to recognize patterns of the environment and also has a fast decision response movement [8]. The FKN algorithm is still a

new method for the minimum resource of the brain system. The FKN algorithm has embedded in the particle swarm robot reported by [9] to achieve real-time reactive formation control for swarm robots. Based on this, the experiment is conducting to implement FKN algorithm into the ASV.

Figure 1. The Autonomous Surface Vehicle

2. Design and Method

In this section, the methodology of the built ASV is dividing into several steps. The first step is designing the architecture of the system. The architecture of the system is representing by the block diagram, in general, the block diagram has input, process and output. The input block is consist of the two sonar sensor, the process block has a microcontroller with an FKN algorithm, and the output block has a motor driver and DC motor. The illustration of this system is showing in `Figure 2`.
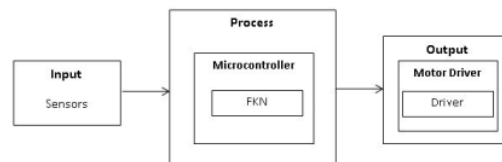

Figure 2. The block diagram of the ASV system

The second steps are calibrating the sonar sensor. The sonar sensor needs to calibrating to make the capability of measuring is precise. To do this, the calibrating is setting for measuring the distance object/obstacle in the range of 50 -210 cm. The procedure of calibrating is putting the object in the various distance such as 50 cm, 70 cm, 90 cm, 110 cm, 130 cm, 150 cm, 170 cm and 210 cm. The sensor detects the object and sent the data in the form of pulse width to the microcontroller (Arduino Uno rev3), this pulse width is getting from TX (transmitter) and receiving by RX of the sensor. The illustration of the minimum system calibrating is showing in `Figure 3`.

The output of the sensor then sent to the microcontroller for converting into digital form by using Analog Digital Converter (ADC) with a data width of 10 bit. The ratio between real distance and value ADC needs to find in the form of calibrating data by using the equation (1). From the `eq(1)`, the ratio between real distance and value ADC is 1.338.

$$q = \frac{bit}{md} \tag{1}$$

where :
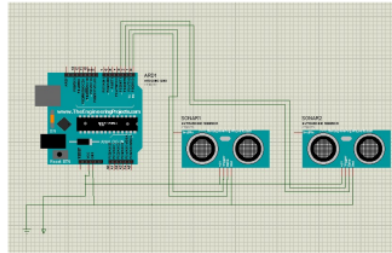
Figure 3. The ilustration of calibrating the distance object by sonar sensor

- q = the rasio between real distance and value ADC;
- md = maximum distance of the sensor(765);
- a = bit(1024);

To find the estimation distance of the object, the equation (2) is applying.

$$x = \frac{valueADC}{q} \qquad (2)$$

where x = the estimation distance between sensor and object (cm).

As a result of calibrating sonar sensor is showing in `Table` 1. from `Table` 1 is sighting that the sonar sensor is precise for the distance between 50 cm - 210 cm.

Table 1. Result of calibrating sonar sensor.

Measuring distance	Value ADC(S1)	Value ADC(S2)	Est Distance(S1)	Est Distance(S2)
50 cm	66	67	49 cm	50 cm
70 cm	93	93	70 cm	70 cm
90 cm	120	120	90 cm	90 cm
110 cm	147	147	110 cm	110 cm
130 cm	174	174	130 cm	130 cm
150 cm	200	200	150 cm	150 cm
170 cm	225	225	168 cm	168 cm
210 cm	278	277	208 cm	207 cm

The third step is developing the algorithm of FKN. The FKN is a part of Neuro-Fuzzy Method, which is the result of integration between Fuzzy Method and Neuron Network (Kohonen)[10]. The steps for learning the process in the fuzzy Kohonen network describing as:

- Step 1: Quantization of Input Value (Fuzzy Logic Section);
- Step 2: Calculate Euclidean Distance (Kohonen Section);
- Step 3: Calculate the Degree of Membership (Fuzzy Logic Section);
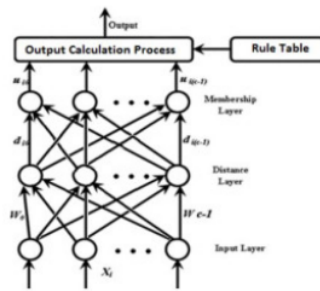- Step 4: Output Processing (Fuzzy Logic Section).
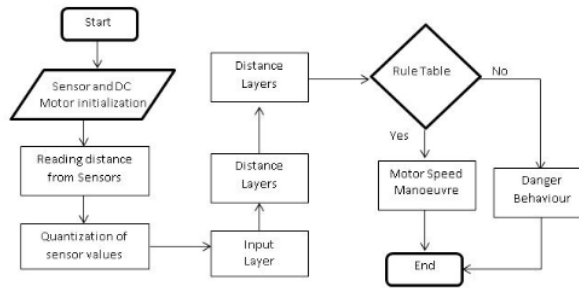
3

Figure 4. FKN Structure.   **Figure 5.** Flowchart Autonomous Surface Vehicle (ASV)

Inside the FKN methods, the process of quantitation of value ADC by using the eq(3).

$$X_i = \begin{cases} 1 & \text{for } 50\text{cm} < Y_i \leq 104\text{cm} \\ 2 & \text{for } 105\text{cm} < Y_i \leq 159\text{cm} \\ 3 & \text{for } Y_i \geq 160\text{cm} \end{cases} \tag{3}$$

where: i = number of sonar sensor.

Euclidean distance is the distance between the similarity between the input value pattern and the embedded rule pattern. The calculation process carried out using the Euclidean distance formula (d_{ij}) found in equation (4)

$$d_{ij} = \sqrt{\sum \left\langle \| x_{i(n)} - w_{j(n)} \| \right\rangle^2} \tag{4}$$

where :

• d = euclidean distance;

• n = Number of Sensors used;

• w = Value of implanted weight;

The calculating membership degree (μ_{ij}) is applying the equation (5). After getting the value of μ_{ij}, the value of the highest membership degree will be multiplied by the reference value of the membership degree pattern so that it gets the output value (equation (6)).

$$\mu_{ij} = \begin{cases} 1 & d_{ij} = d_{min} \\ (d_{max} - d_{ij} \ / \ d_{max} - d_{ij}) & d_{max} \leq d_{ij} < d_{max} \\ 0 & d_{ij} \geq d_{max} \end{cases} \tag{5}$$

where :

• u = degree of membership;

• dmin = smallest Euclidean distance = 0;

• dmax = The biggest euclidean distance;

$$Output = \mu_{ij} * Out_{ref} \tag{6}$$

The rules base is developing according to the `Figure 6` . For the last condition pattern (danger) will occur when each sensor has detected the obstacle, so the motor dc reversing until the condition of the environment is back to each of the rules base.

Pattern (X_i)	Pattern Linguistic		Image pattern condition	Value pattern Prototype		Output (% PWM)	
	S₁	S₂		S₁	S₂	M₁	M₂
X_1	Near	Far		1	3	90	40
X_2	Far	Near		3	1	40	90
X_3	Far	Far		3	3	90	90

Figure 6. The Rules of Base instilled in the FKN
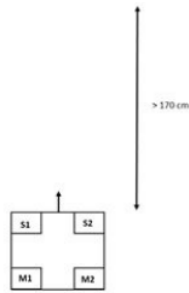
3. Results

In this test, the ASV is placing in the water (swimming pool) to monitor the movement that carried out in a particular environment to find out how the ASV moves through specified pattern recognition. The pattern tested is a pattern commonly used for testing boat robots. The manoeuvre robot in water environments is quite different from the manoeuver robot in the ground. The manoeuver robot in the ground environment is more responsive and quick compared to the water environment. This difficult must be considered by a design engineer to make a simple discussion on its brain system to reduce time-consuming before taking manoeuvre. As this is the case, the FKN algorithm suitable to implement in boat robot (ASV).

3.1. Testing of movements in the environment without obstacles

In this condition, the ASV has no obstacles blocking in front of it. The data extract from Arduino Uno Rev 3 by using cable data to monitor the FKM algorithm in 700 hundredths of seconds (hs).This value of 700 hs is equal to 7 seconds. The scenario is conducting in an environment without obstacles, as shown in `Figure` 7. The ASV is moving forward with a time that set, and the real condition is showing in `Figure` 8. The results of this experiment are showing in `Table` 2.

3.2. Testing of movements in the environment with obstacles on the right

For this testing, the ASV is placing in an environment where obstacle puts in front of the right sensor (`Figure` 9). Based on the human logical perception of this scenario, the ASV should manoeuvre to left sides. According to the rules base (`Figure` 3) that coding in FKN algorithm, the pattern is matched with the pattern X_2, with the output of propulsion left is 40% PWM and 90% PWM for the right. The slow capture with interval collecting data is showing in `Figure` 10. However, from `Table` 3 as the data interval collecting is showing the condition of transition percentage of PWM in time 152,456 and 532. This transition percentage PWM happens because the value of quantitation of value ADC by eq(3) shows the sensor (2) has a value of 2. Nonetheless, pattern recognition is correct (X_2).

Figure 7. The scenario that conduct for experiment without obstacles

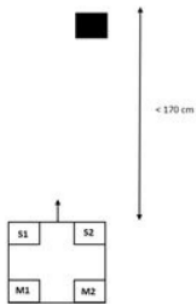**Figure 8.** The real condition of ASV experiment without obstacles

Table 2. Results of movements ASV in the environment without obstacles

| No	Time (hs)	Sensor(1)	Sensor(2)	% PWM left	% PWM Right	pattern recognition
1	0	3	3	90	90	X_3
2	76	3	3	90	90	X_3
3	152	3	3	90	90	X_3
4	228	3	3	90	90	X_3
5	304	3	3	90	90	X_3
6	380	3	3	90	90	X_3
7	456	3	3	90	90	X_3
8	532	3	3	90	90	X_3
9	608	3	3	90	90	X_3
10	684	3	3	90	90	X_3

Table 3. Results of movements ASV in the environment with obstacles in right side

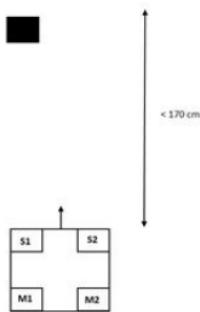| No	Time (hs)	Sensor(1)	Sensor(2)	% PWM left	% PWM Right	pattern recognition
1	0	3	3	90	90	X_3
2	76	3	3	90	90	X_3
3	152	3	2	25.86	58.18	X_2
4	228	3	1	40	90	X_2
5	304	3	1	40	90	X_2
6	380	3	1	40	90	X_2
7	456	3	2	25.86	58.18	X_2
8	532	3	2	25.86	58.18	X_2
9	608	3	3	90	90	X_3
10	684	3	3	90	90	X_3

Figure 9. The scenario that conduct for experiment with obstacles in right side
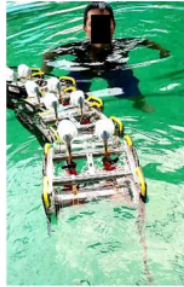
Figure 10. The real condition of ASV experiment with obstacles in right side

3.3. Testing of movements in the environment with obstacles on the left

The next test is the manoeuvre to the right side to verify the rule base of this action (X1), to do this, the obstacle is placing in front of the left sensor (170 cm) as showing in `Figure 11`, with the perception that the ASV will move to the right side. From the experiment, the captured movement between time interval collecting data shows the ASV manoeuvre to the right with a smoothly without stop or delay `Figure 12`. For the detail data collecting, `Table 4` is presenting. The transition in percentage PWM is occurring in time 152, 228 and 532. However, pattern recognition is still correct (X_1).

Figure 11. The scenario that conduct for experiment with obstacles in left side

Figure 12. The real condition of ASV experiment with obstacles in left side

Table 4. Results of movements ASV in the environment with obstacles in left side

No	Time (hs)	Sensor(1)	Sensor(2)	% PWM left	% PWM Right	pattern recognition
1	0	3	3	90	90	X_3
2	76	3	3	90	90	X_3
3	152	2	3	58.18	25.86	X_1
4	228	2	3	58.18	25.86	X_1
5	304	1	3	90	40	X_1
6	380	1	3	90	40	X_1
7	456	1	3	90	40	X_1
8	532	2	3	58.18	25.86	X_1
9	608	3	3	90	90	X_3
10	684	3	3	90	90	X_3

4. Conclusion

Based on the result from the experiments, the ASV with FKM Algorithm is showing successful implement as autonomous navigation thought three patterns recognition. The manoeuvre is quicker and smooth without a crash to obstacles. This manoeuvre is similar to the ground robot experiment with FKN that has been conducted by [8][9]. The manoeuvre robot in the water environment is a different treatment compared to the ground environment. This result shows that the ASV with FKN algorithm has the same behaviour with a ground robot.

References

[1] Courbon, J., Mezouar, Y. and Martinet, P., 2009. Autonomous navigation of vehicles from a visual memory using a generic camera model. IEEE Transactions on Intelligent Transportation Systems, 10(3), pp.392-402.

[2] Huntsberger, T., Aghazarian, H., Howard, A. and Trotz, D.C., 2011. Stereo vision–based navigation for autonomous surface vessels. Journal of Field Robotics, 28(1), pp.3-18.

[3] Galyean, T.A., 1995, April. Guided navigation of virtual environments. In Proceedings of the 1995 symposium on Interactive 3D graphics (pp. 103-ff). ACM.

[4] Duan, G.J. and Zhang, P.F., 2014. Research on Application of UAV for Maritime Supervision. Journal of Shipping and Ocean Engineering, 4, pp.322-326.

[5] Ahilan, T., Adityan, V.A. and Kailash, S., 2015. Efficient Utilization of Unmanned Aerial Vehicle (UAV) for Fishing through Surveillance for Fishermen. World Academy of Science, Engineering and Technology, International Journal of Mechanical, Aerospace, Industrial, Mechatronic and Manufacturing Engineering, 9(8), pp.1468-1471.

[6] Yang Fang, Muye Pang, Biao Wang, 2017. A course control system of unmanned surface vehicle (USV) using back-propagation neural network (BPNN) and artificial bee colony (ABC) algorithm.Procedia Computer Science. Vol 111, pp 361-366.

[7] Blaich, M., Wirtensohn, S., Oswald, M., Hamburger, O. and Reuter, J., 2013. Design of a twin hull based USV with enhanced maneuverability. IFAC Proceedings Volumes, 46(33), pp.1-6.

[8] Prasetyo, A.P., Astrowulan, K. and Fatoni, A., 2016. Implementation of Navigation Target Seeker Mobile Robot Based on Pattern Recognition with Fuzzy Kohonen Network (FKN) Methods. IPTEK Journal of Proceedings Series, 2(1).

[9] Nurmaini, S., Tutuko, B. and Aditya, A., 2015. Pattern Recognition Approach for Formation Control for Swarm Robotics Using Fuzzy-Kohonen Networks. Proceeding of the Electrical Engineering Computer Science and Informatics, 2(1), pp.143-144.

[10] Song, K. T., Huang, S. Y. (2004, September). Mobile robot navigation using sonar direction weights. In Proceedings of the 2004 IEEE International Conference on Control Applications, 2004. (Vol. 2, pp. 1073-1078). IEEE.

Autonomous Surface Vehicle (ASV) Obstacle Avoidance Using Fuzzy Kohonen Network (FKN)

1 R Passarella, A Zarkasi, H Maghfur, Sutarno, K Exsaudi, APP Prasetyo, H Veny. "Autonomous Surface Vehicle (ASV) Obstacle Avoidance Using Fuzzy Kohonen Network (FKN)", IOP Conference Series: Materials Science and Engineering, 2019
Publication
82%

2 Submitted to Aston University
Student Paper
4%

3 www.ggmfg.com
Internet Source
1%

4 Submitted to Universitas Jenderal Soedirman
Student Paper
1%

5 www.biano.cz
Internet Source
1%