

# Global Journal of Computer Sciences: Theory and Research



Volume 9, Issue 1, (2019) 001-009

[www.gjcs.eu](http://www.gjcs.eu)

## Fuzzy logic-based moving obstacle avoidance method

**Suat Karakaya\***, Mechatronics Engineering Department, Engineering Faculty, Kocaeli University, 41380 Izmit, Turkey

**Hasan Ocak**, Mechatronics Engineering Department, Engineering Faculty, Kocaeli University, 41380 Izmit, Turkey

### Suggested Citation:

Karakaya, S. & Ocak, H. (2019). Fuzzy logic-based moving obstacle avoidance method. *Global Journal of Computer Sciences: Theory and Research*. 9(1), 001–009.

Received September 15, 2018; revised January 10, 2019; accepted March 12, 2019.

Selection and peer review under responsibility of Prof. Dr. Dogan Ibrahim, Near East University, Cyprus.

©2019. All rights reserved.

---

### Abstract

In this study, a fuzzy logic-based collision avoidance method is studied. Mobile robots operating in a dynamic environment may encounter moving obstacles. The distance from the obstacles to the robot is provided as a control input. The change of distance is provided as the secondary input. The output speed value is expressed as 'stop', 'very slow', 'slow', 'medium' and 'fast', respectively. Each of these expressions has an exact numerical value, but it depends on the inputs and the rule base, and therefore the numerical values belong to the variable ratios. The input parameters do not have an exact value and they are assigned to different numerical ranges at different ratios. The input membership functions defined in this study are defined as triangular functions. The defuzzification process of the output parameter is performed by the weighted average method.

**Keywords:** fuzzy; logic-based; mobile; avoidance method

---

\* ADDRESS FOR CORRESPONDENCE: **Suat Karakaya**, Mechatronics Engineering Department, Engineering Faculty, Kocaeli University, 41380 Izmit, Turkey. *E-mail address:* [suatronik@gmail.com](mailto:suatronik@gmail.com) / Tel.: +90 (262) 303 10 00

## 1. Introduction

Detection of moving obstacles is an important issue in mobile robotics and it is more complicated than working with stationary obstacles. Since the obstacles are stationary and working on a known map presents relatively more predictable conditions, stationary environments can be considered an 'easier' problem. In some cases, the obstacles are static, but the map may not be known in advance. In such cases, there are also options to perform operations at the local map and to save the immediate sight to memory when necessary. In both situations, a predictable obstacle profile can be mentioned at certain levels. Trajectory planners used to determine instant targets classify the relevant coordinates of the map as 'occupied' or 'free' in most cases. This knowledge is benefited when passing through the same location. Such a classification is useful on a grid-based static map. However, a mobile object may intersect the position of the mobile robot and this is a predictable situation. Moving objects are also a problem for mapping and positioning methods. Simultaneous localisation and mapping and similar methods can change the status of a particular region on the map at a later stage [1]. The frequency of occurrence of an obstacle in a particular coordinate is considered as a parameter. Thus, a moving object is ignored by static map construction process. A number of studies have been carried out to detect moving obstacles and to calculate alternative trajectories against the orientation of these obstacles. Potential field and model predictive control methods are applied on moving obstacle detection in Nishio et al. [2]. Genetic algorithm and fuzzy logic-based controllers are applied on dynamic obstacle detection problem in Yeasmin and Shill [3]. Moving obstacle avoidance solution for omni-drive mobile robots is presented in Khelloufi et al. [4]. A positioning scheme is proposed in Wu et al. [5] by using moving obstacle assistance. Determination of speed and direction of moving obstacles is studied in Mansor et al. [6]. Low rank decomposition scheme is offered in Deng et al. [7]. A novel moving obstacle avoidance method for flying robots is proposed in Li and Savkin [8]. A collision avoidance scheme for high-speed ground mobile robots is presented in Febbo et al. [9] and a potential field-based hybrid dynamic moving obstacle avoidance technique is studied in Malone et al. [10]. In this paper, two dimensional laser range measurement data collected via a lidar (Light Detection and Ranging) are processed. The stationary objects are filtered out by the developed algorithm and the distance between the wheeled mobile robot (WMR) and remaining moving objects are assumed as the primary control input of the fuzzy logic controller. Gradient of the motion between the WMR and the closest moving object is assumed as the secondary input. The linear velocity of the WMR is controlled depending on the most critical moving object which is estimated to be the closest one in the next step. Theoretical background and developed method are given in Section 2. Results and discussions are given in Sections 3 and 4, respectively.

## 2. Theoretical background and method

In this section, mathematical basis of the proposed scheme is presented. The algorithm mentioned in this paper consists of three main steps: Local target detection, major obstacle elimination and velocity control procedure.

### 2.1. Local target detection

Local target is basically the immediate goal of the mobile robot. This goal point is determined by utilising both the stationary obstacles and the estimated poses of the moving obstacles. The moving obstacle detection procedure is explained detail in Karakaya et al. [11]. After determining the moving obstacles, a gap detection procedure given in Karakaya et al. [12] is applied on the current lidar data. But the moving obstacles' lidar profiles are assumed to be the estimated positions rather than the current status. The minimum cost point lying on the gaps is assumed to be the local target point [12]. Thus, the necessary orientation angle to steer the WMR towards the local target is determined. This angle is used to obtain left and right wheels' velocities by converting the average linear velocity of the WMR to wheel velocities. The local target determination process is illustrated in Figure 1. The hatched

regions show obstacles and the red-colored 'x' demonstrates the current local target. The triangular shape labelled with 'R' shows the WMR. The current orientation angle of the WMR is shown with  $\phi_{r0}$  and the next angle is given with  $\phi_{r1}$ . The left and right wheels' velocity is determined depending on the error between the current orientation angle and the next orientation angle (Eq. (1)). After this stage, the major moving obstacle is to be detected. The linear velocity of the center of the WMR is controlled by processing the major obstacle's particular parameters.

$$v_{left} = v_{wmr} + v_{wmr} \cos(\phi_{err}), v_{right} = v_{wmr} - v_{wmr} \cos(\phi_{err}), \phi_{err} = \phi_{current} - \phi_{next} \quad (1)$$

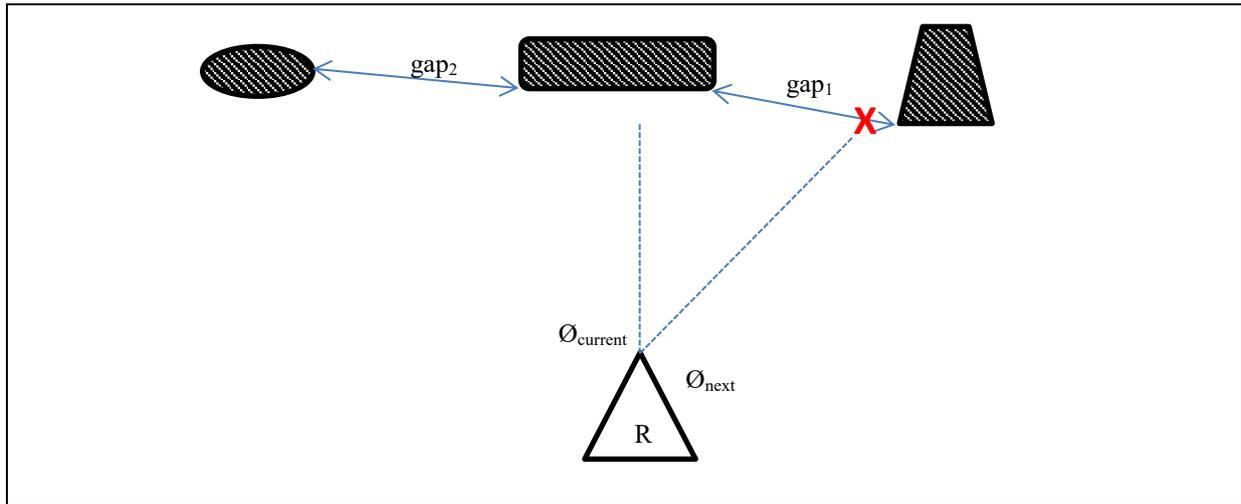


Figure 1. Local target and heading orientation determination

## 2.2. Major obstacle determination

Moving obstacle detection procedure is explained in Karakaya et al. [11]. Any different method can be adapted to this paper for motion detection. The sensor's field of view is defined as a circular section. The obstacles in the field of view are read as sequential laser measurement points. Each set of points is assumed to be an obstacle. Cluster centers of point sets represent the corresponding object. Position of an object in Cartesian space is referred with a vector  $p$  given in Eq. (2). In Eqs. (3)–(5), the sub-index  $b$  is used for obstacles,  $r$  is for robot,  $x$  for horizontal axis and  $y$  for vertical axis. The term  $n$  demonstrates the discrete time parameter.

$$B = \{ \forall p \mid |p_r - p_b| - R \leq 0 \} = \{ p_1^n, p_2^n, p_3^n, \dots, p_{\#obstacles}^n \} \quad (2)$$

$$\nabla_{p_b}^n = \frac{\partial (p_{b_x}^n - p_{b_x}^{n-1})}{\partial x} d_x + \frac{\partial (p_{b_y}^n - p_{b_y}^{n-1})}{\partial y} d_y, \quad \forall p_b \in B \quad (3)$$

$$\hat{p}_b^{n+1} = p_b^n + \nabla_{p_b}^n \quad (4)$$

$$b^* = \{ (b) \mid \min(|p_r^n - p_b^{n+1}|) \} \quad (5)$$

$$\delta b_i = |Rb_i|^n - |Rb_i|^{n+1} \quad (6)$$

Major obstacle is the closest obstacle to the WMR in terms of the estimated positions. A demonstration is given in Figure 2, where  $b_i[m]$  is used for demonstration of the  $i$ th obstacle and  $R[m]$  is used to show the robot at the time  $m$ . The moving obstacle which has the minimum Euclidian distance to the WMR is searched at each interval.

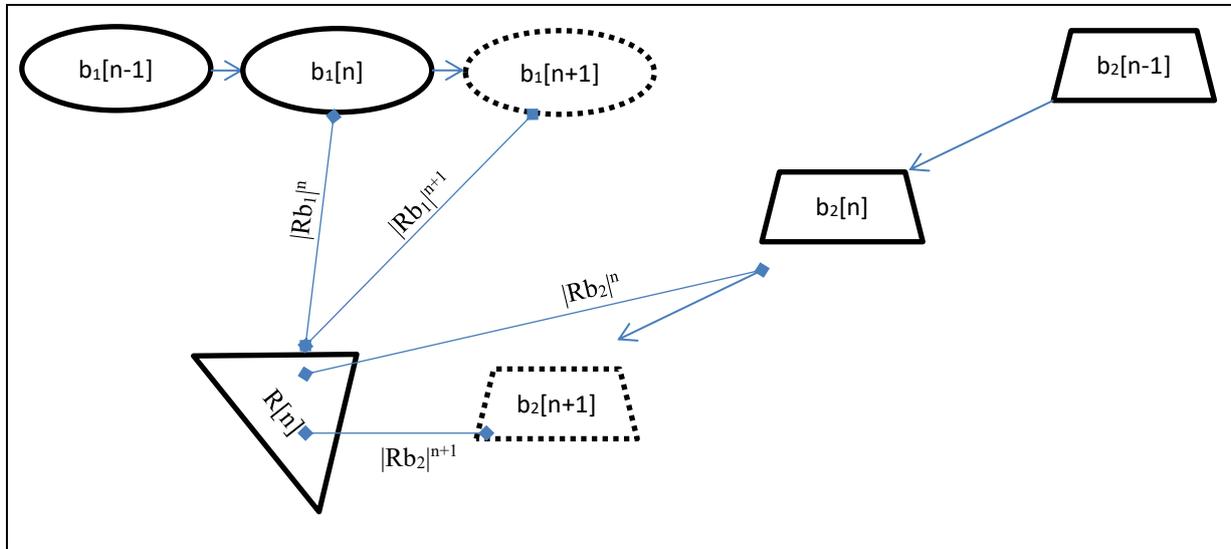


Figure 2. Major obstacle determination

The related obstacle is assumed to be the major obstacle (Eq. (5)). In addition, the first derivation of this distance is calculated at each stage. The distance and derivation values are the input parameters of the fuzzy controller. Assuming the distances are relative measurements according to the WMR's position the derivation of the  $i$ th moving obstacle is calculated as given in Eq. (6). The replacements of the moving obstacles (Eq. (6)) are searched and the major obstacle which satisfies the criterion given in Eq. (5) is determined.

### 2.3. Velocity control

#### 2.3.1. Fuzzification of the input parameters

In this section, the linear velocity control of the WMR's centroid ( $v_{wmr}$ ) (Eq. (1)) is explained. The first controller input is the Euclidian distance between major obstacle and the WMR ( $|Rb^*|$ ). The second input is the replacement of the major moving obstacle ( $\delta b^*$ ). The exact values of the fuzzy controller are given in Table 1.

Table 1. Exact values of the output parameters (meters per second)

Stop	Very slow	Slow	Average	Fast
0	2	4	5	7

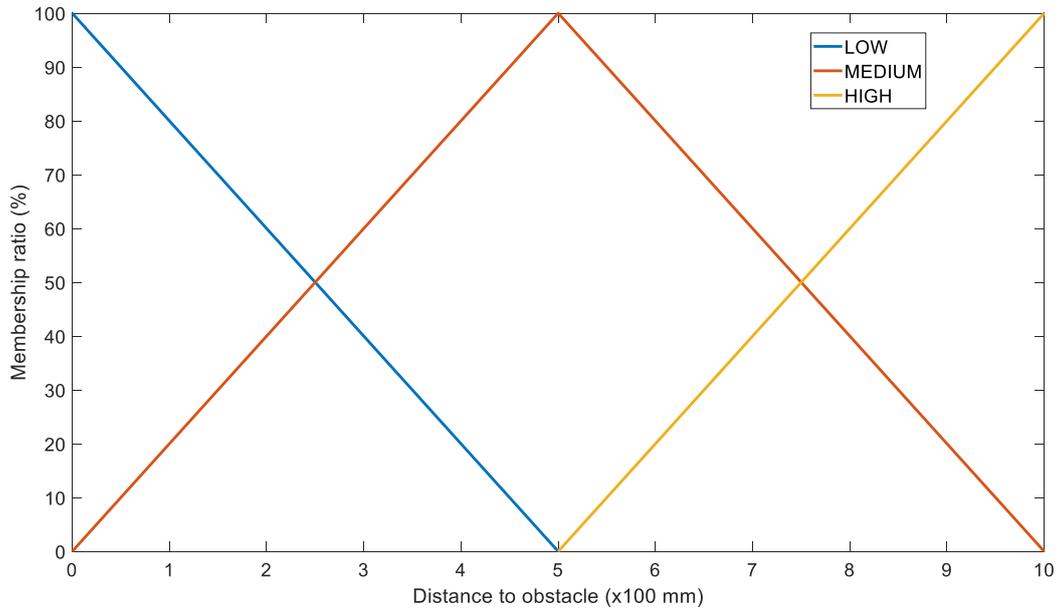
The fuzzification technique is min-max method [13]. The rule base of proposed controller is given in Table 2. The input 'distance to obstacle' is described with triangular functions: 'low', 'medium' and 'high'. The input 'derivation of the distance to obstacle' is defined with triangular functions: 'closing', 'stationary' and 'away'. The scanning range for moving obstacles is 1 m. Displacement of each moving obstacle in the scanning range is measured and tagged as 'closing', 'stationary and 'away'. These words correspond to the secondary input membership clusters. Membership limitations of the clusters for both first and the second input are given in Figures 3 and 4, respectively. The down limit of

the first input is 0 and the up limit is 1,000 mm. The down limit of the second input is -500 mm and the up limit is 500 mm.

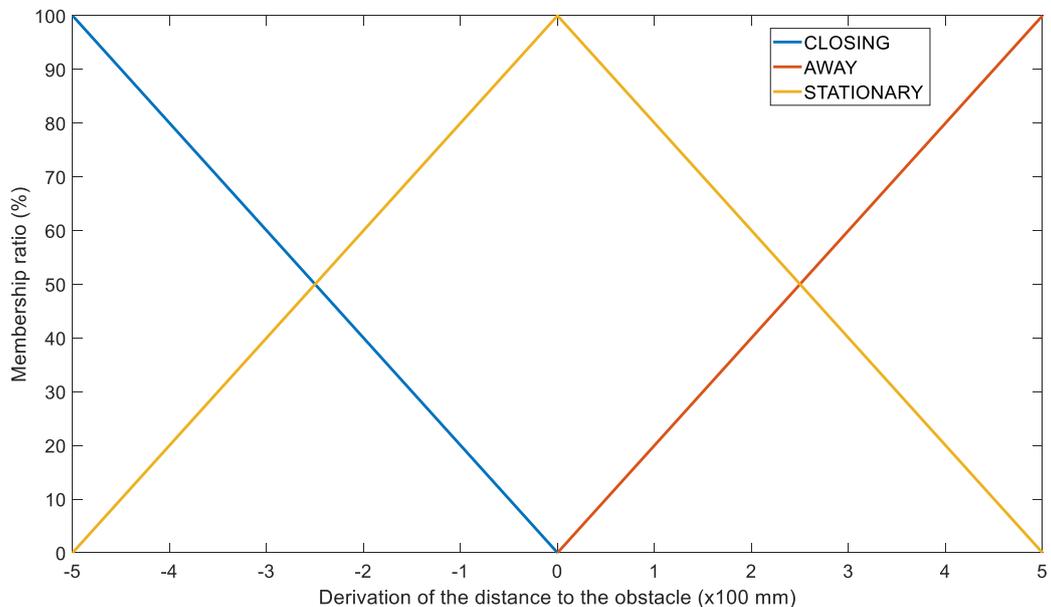
**Table 2. Rule table for fuzzy-logic controller**

Input <sub>2</sub> ↓ Input <sub>1</sub> →	Low <sup>a</sup>	Medium <sup>a</sup>	High <sup>a</sup>
Closing <sup>b</sup>	Stop	Very slow	Slow
Stationary <sup>b</sup>	Very slow	Slow	Average
Away <sup>b</sup>	Slow	Average	Fast

<sup>a</sup>Distance to obstacle. <sup>b</sup>Derivation of the distance to obstacle.



**Figure 3. Membership function of the input 'Distance to obstacle'**



**Figure 4. Membership function of the input 'Derivation of the distance to obstacle'**

### 2.3.2. Defuzzification of the input parameters

The exact values of each output are given in Figure 5. Defuzzification scheme of the fuzzy-logic controller is weighted-averages method. The final membership coefficient of each output weighted as given in Eq. (7). This method is a simple and effective defuzzification method with low computational cost. The term  $o$  is used for output,  $\mu$  for membership ratio and  $r$  for the exact output.

$$r = \frac{\sum \mu_i o_i}{\sum \mu_i} \quad (7)$$

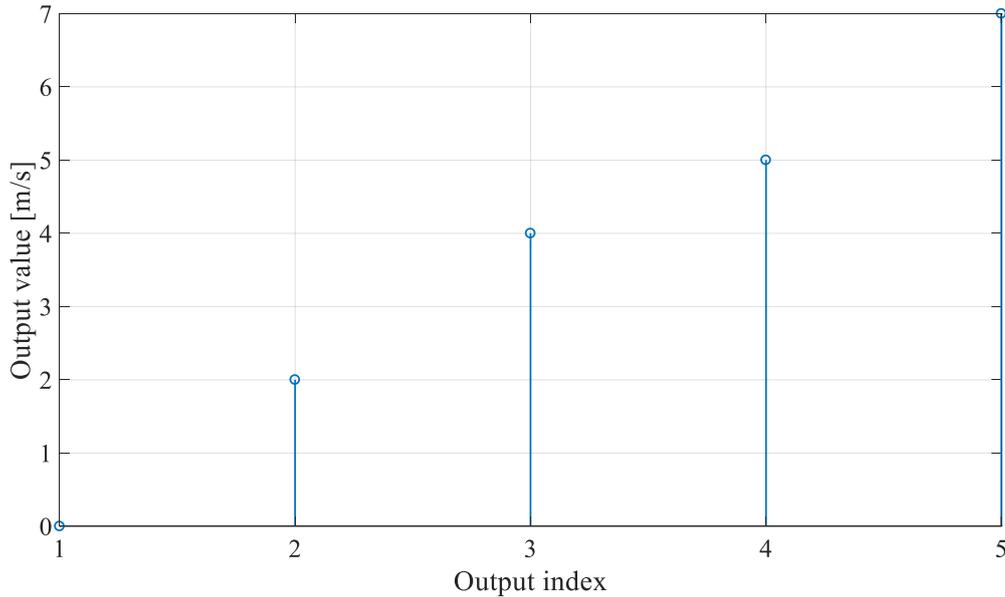


Figure 5. Exact output values

## 3. Results

In this section, an example simulation result is given. The test environment is the Mobile Robot Toolbox [14]. Motion of the simulated differential drive WMR is given in Figure 6 under dynamic conditions. The WMR directly steers towards the goal until its path is crossed by the obstacle. The controller reduces the velocity while the obstacle is closing to the WMR. The controller stops the WMR when the obstacle to robot distance is not safe. Thus, the WMR waits until it clears the obstacle (scenes 9–13 in Figure 6). The WMR continues its motion soon after the obstacle’s estimated pose is a safe distance (scenes 17–24 in Figure 7). The locations of the WMR during the navigation process are given in Figure 6. The instant poses are illustrated with dot-points. Starting point, goal and obstacle clearing region are labelled on the figure. The motion of the moving obstacle is given in Figure 8. The obstacle is assumed to be a round object and it moves on a linear path. The obstacle reverses when it reaches the reverse manoeuvring point. The green path illustrates the planned trajectory and the red path is tracked path by the WMR. The red rectangular figure is WMR and the beam on the WMR is current directory of the robot.

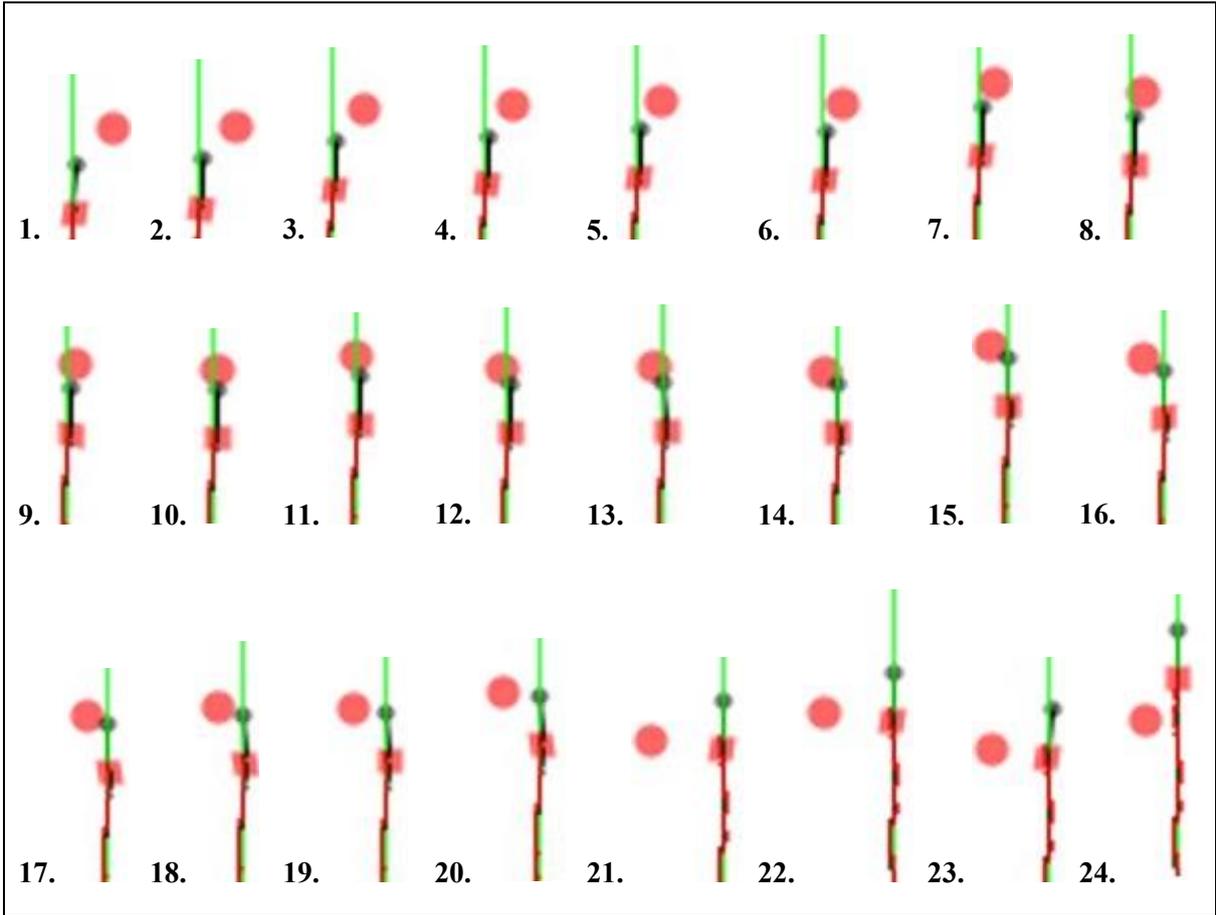


Figure 6. Sample screenshots of obstacle avoidance process

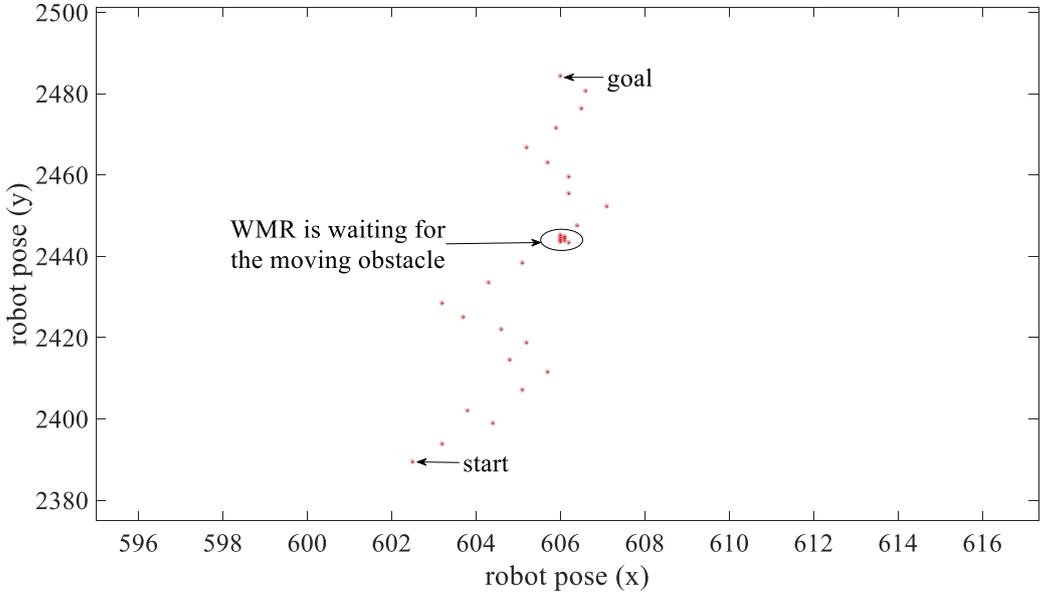
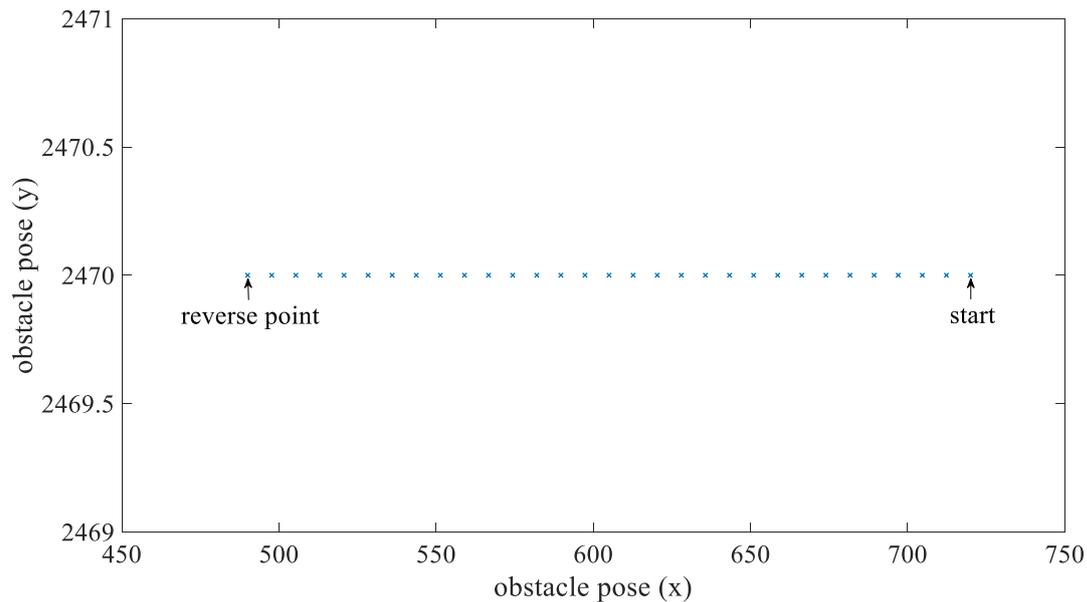


Figure 7. Position of the WMR during the obstacle avoidance process



**Figure 8. Position of the moving obstacle during the obstacle avoidance process**

#### 4. Discussions

This paper proposes a fuzzy-logic controlled obstacle avoidance scheme for WMRs. The main assumption is that the WMR is equipped with a range sensor or able to get visual feedback. The path planning is out of scope. The obstacle avoidance scheme is executed while the WMR is tracking a pre-planned trajectory. Increasing number of the mobbing obstacle may be challenging for this scheme. At one local search, the maximum number of avoided obstacle is approximately 5; therefore, it will be improved for future studies.

#### Acknowledgements

This study was supported by the Scientific and Technical Research Council of Turkey (TUBITAK) under the grant TUBITAK-BIDEB 2211A.

#### References

- [1] Norgren, P. & Skjetne, R. (2018). A multibeam-based SLAM algorithm for iceberg mapping using AUVs. *IEEE Access*, 6, 26318.
- [2] Nishio, Y., Nonaka, K. & Sekiguchi, K. (2018). *Moving obstacle avoidance control by fuzzy potential method and model predictive control*. Asian Control Conference, Australia, p. 1298.
- [3] Yeasmin, S. & Shill, P. C. (2017) GA-based adaptive fuzzy logic controller for a robotic arm in the presence of moving obstacle. International Conference on Electrical Information and Communication Technology, Khulna, p. 1.
- [4] Khelloufi, A., Achour, N., Passama, R. & Cherubini, A. (2017). *Tentacle-based moving obstacle avoidance for omnidirectional robots with visibility constraints*. International Conference on Intelligent Robots and Systems, Vancouver, p. 1331.
- [5] Wu, D., Zhu, D., Liu, Y. & Zhao, D. (2018) Location verification assisted by a moving obstacle for wireless sensor networks. *IEEE Internet of Things Journal*, 5, 322.

- [6] Mansor, A. H., Rashid, Z. A. & Abdullah, E. R. (2014). *Determination of direction and speed of a uniformly moving obstacle from intersections with virtual window*. IEEE Symposium on Industrial Electronics & Applications, Kota Kinabalu, p. 148.
- [7] Deng, Y. Y., Xu, Y. R., Wang, D., Gao, Y. F. & Wu, X. Q. (2016) *Moving obstacle removal via low rank decomposition*. International Conference on Digital Home, Guangzhou, p. 48.
- [8] Li, H. & Savkin, A. (2017) *A method for collision free sensor network based navigation of flying robots among moving and steady obstacles*. Chinese Control Conference, Dalian, p. 6079.
- [9] Febbo, H., Liu, J., Jayakumar, P., Gao, Y. F., Stein, J. L. & Ersal, T. (2017). *Moving obstacle avoidance for large, high-speed autonomous ground vehicles*. American Control Conference, Seattle, p. 5568.
- [10] Malone, N., Chiang, H. T., Lesser, K., Oishi, M. & Tapia, L. (2017). Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field. *IEEE Transactions on Robotics*, 33, 1124.
- [11] Karakaya, S., Kucukyildiz, G. & Ocak, H. (2018). One dimensional moving obstacle detection method for mobile robotics applications. International Signal Processing and Communications Applications Conference, Izmir, p. 1.
- [12] Karakaya, S., Kucukyildiz, G. & Ocak, H. (2016). Detection of obstacle-free gaps for mobile robot applications using 2-D LIDAR data. *International Journal of Natural and Engineering Sciences*, 10, 23.
- [13] Kundu, S. (1998). The min-max composition rule and its superiority over the usual max-min composition rule. *Fuzzy Sets and Systems*, 93, 319.
- [14] Karakaya, S, Kucukyildiz, G. & Ocak, H. (2016). A new mobile robot toolbox for Matlab. *International Journal of Intelligent & Robotic Systems*, 87, 125.