UDC 004.942: 519.876.5 **doi:** https://doi.org/10.20998/2522-9052.2022.2.04

Mahabbat Khudaverdiyeva

Azerbaijan State University of Oil and Industry, Baku, Azerbaijan

# MODELING OF MOBILE ROBOT WITH OBSTACLE AVOIDANCE USING FUZZY CONTROLLER

**Abstract.** This paper presents the modeling of a robot's navigation using ultrasonic sensors under uncertainty. The robot tries to avoid obstacles by using the fuzzy logic controller to process the data coming from three ultrasonic sensors. To assess the performance of fuzzy logic optimized robot navigation controller with ultrasonic sensors, which measure the distance by calculating the time spent on the object and its return, the obstacles are placed in front of, left, and right of the robot. Mamdani fuzzy reasoning system is used for the designed controller for its intuitive properties and fewer setting parameters which reduces the amount of time spent on the programming of the controller. 25 rules are considered to cover a robot's possible interactions with obstacles. For an easy understanding of navigation architecture and rapid algorithm implementation, in this paper, a MATLAB simulation framework is developed. MATLAB/Simulink is one of the best simulation tools required to design the architecture and verify algorithms with real-time constraints. Resultant models of the fuzzy optimized controller demonstrate the superior performance of the fuzzy logic controller with high adaptability to the environment while maintaining a sufficient level of accuracy. The designed fuzzy controller can be used in microprocessor/microcontroller-based robots owing to easiness in implementation and coding.

**Keywords:** fuzzy controller; robot navigation; sensor; MATLAB environment.

## Introduction

Mobile robot autonomous navigation is a challenging task to implement. The present controllers for mobile robots fail to deliver the desired control performance because of the problems associated with the tuning methods. Fuzzy controllers, on the other hand, are capable of providing the required performance. Therefore, it is necessary to build a navigation architecture that includes modules with well-defined and self-contained operations and design the interactions between those modules.

**Analysis of publications.** Publication analysis is necessary to assess the significance of the fuzzy controlled robots with obstacle avoidance. A lot of papers have been published on robot controllers for efficient control. Star and Velocity-Obstacle Algorithms are proposed in [1].

A sensor-based control system using a Q-learning algorithm was proposed in [2]. This method utilizes a learning-based controller that learns to create the function between the sensory outputs and the system variables. Q-Learning is the reinforcement learning method selected to solve the obstacle avoidance problem. Reinforcement Learning allows a mobile robot to independently explore an optimal behavior through trial-and-error methods with its outer environment.

Another method for obstruction avoidance for mobile robots was suggested in [3]. In their technique, the robot is capable of navigating through an uncertain environment while passing by obstacles. Simulations were implemented assuming the robot had tactile sensors. Another method for obstacle avoidance Based on Computer Path Planning was suggested in [4] using a rapid and effective algorithm for obstacle avoidance. A multi-sensor integration mechanism for mobile robot navigation is proposed in [5]. A novel method for the arithmetic mean-based decision-making system is described in [6]. Recently, there is a growing interest in fuzzy-controlled mobile robots [7-9].

Dynamic-Window Method (DWM) [10] converts the environmental data in the classical Cartesian coordinate system into linear and angular velocities as references, computes the weighted sum of distances, and avoids the placed obstacles by selecting the objective function's largest value.

The Expanded Guide Circle (EGC) method [11] is an algorithm that adjusts the operator's control input to avoid collisions. Although it is a method intended for boosting the safety of remote operations, there are two issues which are 1) a control input toward obstacles existing under several specific conditions, and 2) a zigzag action along a hallway. The Dual-EGC algorithm is expanded around these two intersections [12].

Another famous method is obstacle avoidance with a vision which was proposed in [13-18]. LiDAR-based approaches are also used in mobile robots [19-20]. These methods extract spatial data from laser point-cloud with segmentation and clustering approaches.

Although the aforementioned methods are effective for better control performance, the increased complexity of these algorithms makes them unsuitable for microcontroller-based applications.

**Purpose and problem statement.** The purpose of the work is to model and simulate a fuzzy controller by developing its rule-table, linguistic variables, and input-output variables.

## Simulation of a robot's navigation s in a MATLAB environment under uncertainty

Let's look at a simulation of a robot's navigation in a MATLAB environment. Let's determine the relationship between the input and output parameters in the microcontroller built on the fuzzy logic system in these robots. The robot tries to avoid obstacles with fuzzy logic. Obstacles are placed in front of, left, and right of the robot. It measures the distance by calculating the time spent on the object and its return.

There is an output quantity corresponding to this distance. The sum of all outputs is visible at the end, aggregation is performed. The previous or conditional block of the rule begins with the expression IF, and the result or result block begins with the expression THEN. The value assigned to the resultant block is equal to the logical product of the activation values of the previous membership functions that characterize the boundaries of the fuzzy sets. The activation value is equal to the value of the membership function where the input variable intersects at the time of evaluation.

A diffusion operation is performed to convert fuzzy values represented by logical products and sequential membership functions into a stable and accurate result. De-fusion can be done in several ways.

Most applications perform mass center or fuzzy centroid calculations in a fuzzy set. The distance is left, the distance is forward, the distance is right, and the output speed and the direction of rotation are shown.

In MATLAB we can show the general interface of the fuzzy controller as follows. controller to determine the speed and rotation angle of the robots by activating corresponding robots.

Fig. 1 illustrates the inputs and outputs of the designed controller. As is seen from the graph the data received from the front, right, and left sensors provide data on the distance from corresponding obstacles. According to these data, the controller calculates the speed and rotation angle of the robot to avoid the obstacle. Fig. 2 depicts input variable (distance_ forward) fuzzification which determines the distance between the front sensor and the obstacle in front of the robot. The same procedure is carried out for the other two input variables with the same settings.

The following Fig. 3 is a set of rules for the dependencies between the most common IF-THEN fuzzy sets in software computing technologies. These rules, according to the data coming from ultrasonic sensors, determine the activation of motors for speed and rotation angle control.

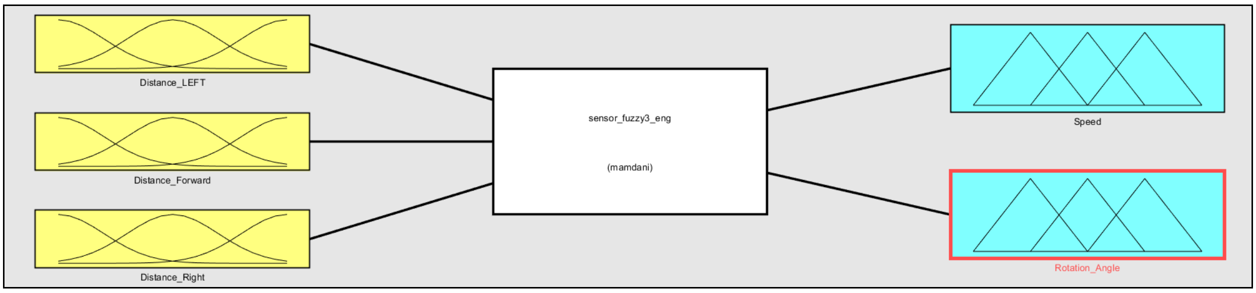


**Fig. 1.** Fuzzy Logic Toolbox editor

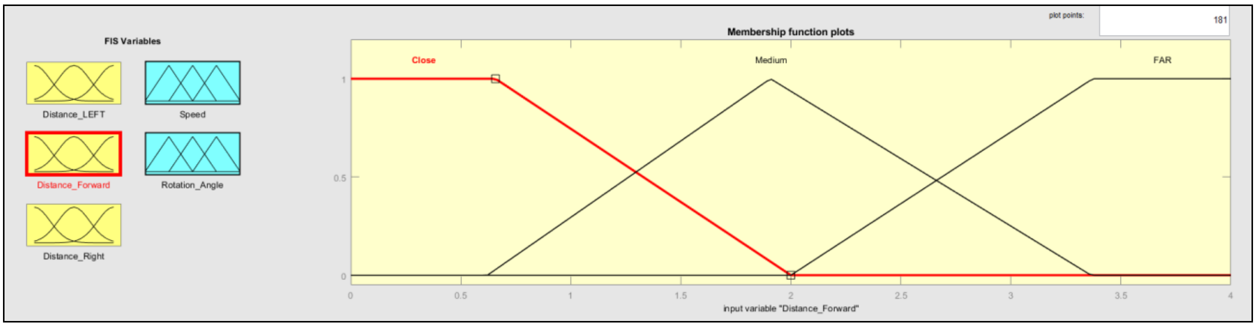
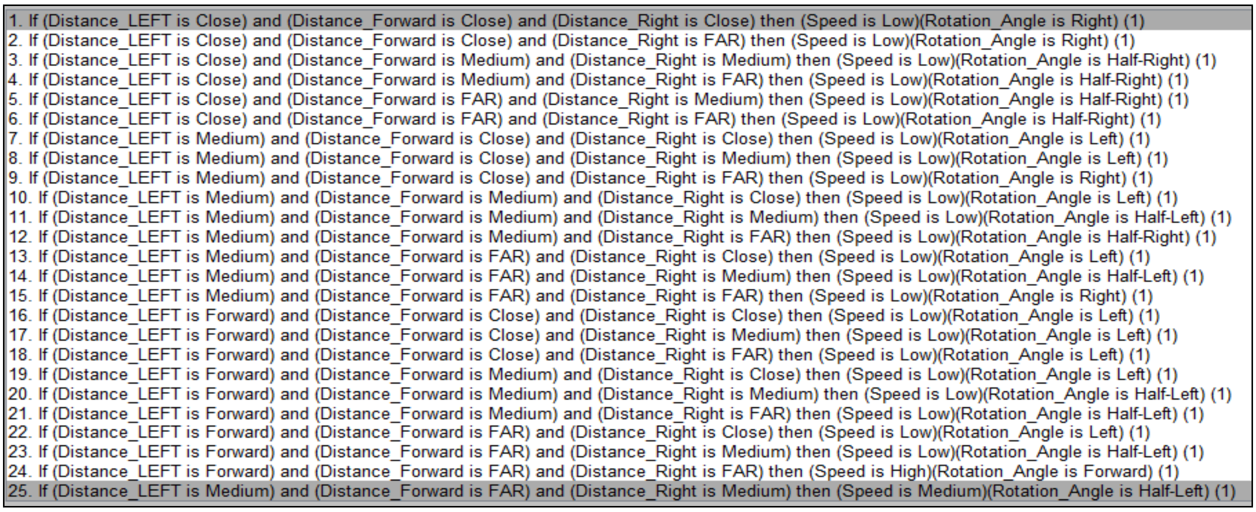


**Fig. 2.** Introduction Quantification "Distance Forward"



1. If (Distance_LEFT is Close) and (Distance_Forward is Close) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Right) (1)
2. If (Distance_LEFT is Close) and (Distance_Forward is Close) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Right) (1)
3. If (Distance_LEFT is Close) and (Distance_Forward is Medium) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Right) (1)
4. If (Distance_LEFT is Close) and (Distance_Forward is Medium) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Half-Right) (1)
5. If (Distance_LEFT is Close) and (Distance_Forward is FAR) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Right) (1)
6. If (Distance_LEFT is Close) and (Distance_Forward is FAR) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Half-Right) (1)
7. If (Distance_LEFT is Medium) and (Distance_Forward is Close) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
8. If (Distance_LEFT is Medium) and (Distance_Forward is Close) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Left) (1)
9. If (Distance_LEFT is Medium) and (Distance_Forward is Close) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Right) (1)
10. If (Distance_LEFT is Medium) and (Distance_Forward is Medium) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
11. If (Distance_LEFT is Medium) and (Distance_Forward is Medium) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Left) (1)
12. If (Distance_LEFT is Medium) and (Distance_Forward is Medium) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Half-Right) (1)
13. If (Distance_LEFT is Medium) and (Distance_Forward is FAR) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
14. If (Distance_LEFT is Medium) and (Distance_Forward is FAR) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Left) (1)
15. If (Distance_LEFT is Medium) and (Distance_Forward is FAR) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Right) (1)
16. If (Distance_LEFT is Forward) and (Distance_Forward is Close) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
17. If (Distance_LEFT is Forward) and (Distance_Forward is Close) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Left) (1)
18. If (Distance_LEFT is Forward) and (Distance_Forward is Close) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Left) (1)
19. If (Distance_LEFT is Forward) and (Distance_Forward is Medium) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
20. If (Distance_LEFT is Forward) and (Distance_Forward is Medium) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Left) (1)
21. If (Distance_LEFT is Forward) and (Distance_Forward is Medium) and (Distance_Right is FAR) then (Speed is Low)(Rotation_Angle is Half-Left) (1)
22. If (Distance_LEFT is Forward) and (Distance_Forward is FAR) and (Distance_Right is Close) then (Speed is Low)(Rotation_Angle is Left) (1)
23. If (Distance_LEFT is Forward) and (Distance_Forward is FAR) and (Distance_Right is Medium) then (Speed is Low)(Rotation_Angle is Half-Left) (1)
24. If (Distance_LEFT is Forward) and (Distance_Forward is FAR) and (Distance_Right is FAR) then (Speed is High)(Rotation_Angle is Forward) (1)
25. If (Distance_LEFT is Medium) and (Distance_Forward is FAR) and (Distance_Right is Medium) then (Speed is Medium)(Rotation_Angle is Half-Left) (1)

**Fig. 3.** Dependencies between fuzzy sets in the most common IF-THEN fuzzy relationships in software computing technologies

25 rules have been constructed for the fuzzy rule table.

Fig. 4 shows the rotation degrees ranging from -90 to +90 degrees, with -90 representing leftmost, -45 for half-left, 0 forward, +45 for half-right, and +90 for rightmost direction in a non-fuzzy/ crisp system.

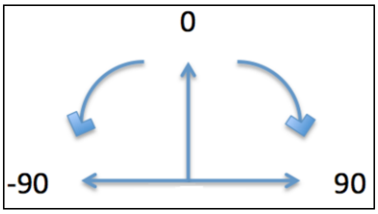


**Fig. 4.** Rotation angles

Two output variables (speed and rotation angle) are assumed. Their corresponding fuzzification process is described in Fig. 5 and Fig. 6, respectively. Triangle type membership function for each linguistic variable is considered because of its simplicity and easiness in implementation. In Fig. 5 and Table 1, five linguistic variables (Left, Half-Left, Forward, Half-Right, Right) are assigned for output rotation angles. In Fig. 6 and Table 2, three linguistic variables (Low, Medium, and High) are defined for the output speed of the robot.

*Table 1* – **Fuzzy Linguistic variables for rotation angles**

| Linguistic Variable | Rotation degrees |
|---|---|
| Left | -90 to -87 |
| Half-Left | -90 to 0 |
| Forward | -45 to 45 |
| Half-Right | 0 to 90 |
| Right | 75 to 90 |

*Table 2* – **Fuzzy Linguistic variables for rotation angles**

| Linguistic Variable | Speed |
|---|---|
| Low | 0 to 0.2 m/s |
| Medium | 0.1 to 0.7 m/s |
| High | 0.6 to 1 m/s |

The purpose of the inclusion of output quantity “speed’’ is to decrease the speed when the distance decreases between the robot and the obstacle to avoid the obstacle.

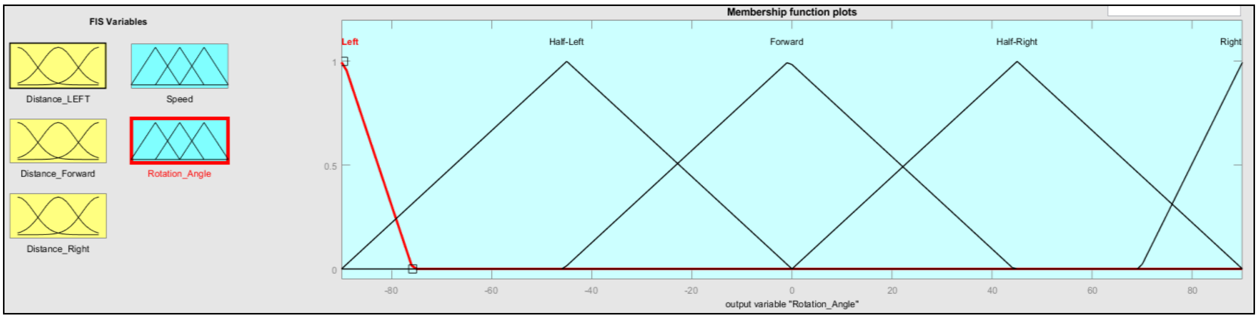


**Fig. 5.** Output quantity "Rotation angle" classification

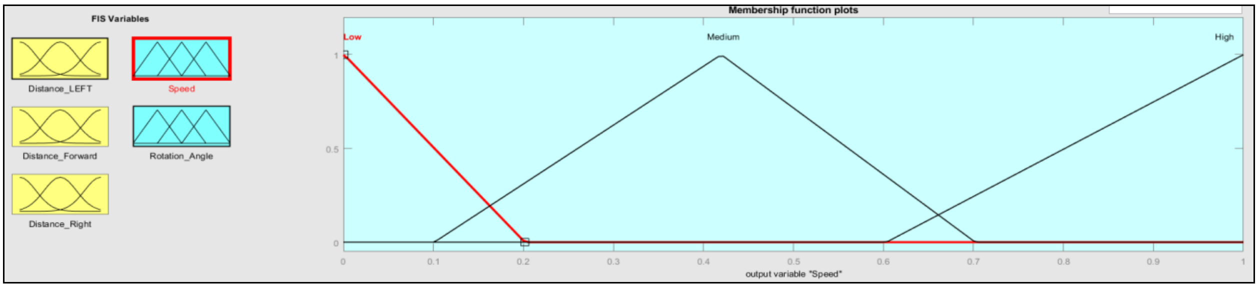


**Fig. 6.** Output quantity "Speed" classification

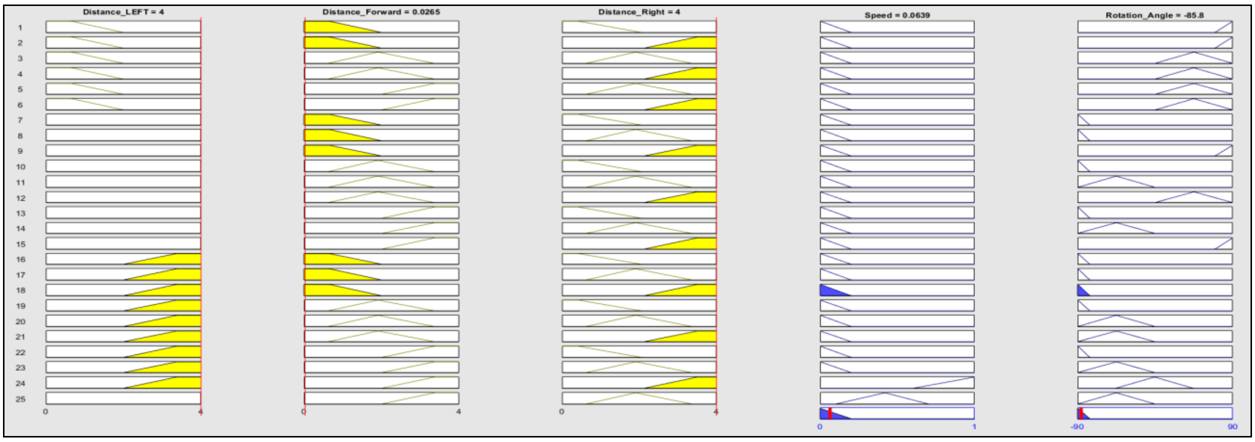


**Fig. 7.** Activation window for fuzzy rules

## Simulation Results

Activation of rules is depicted in Fig. 7, whereas simulation results are shown in Fig. 8.

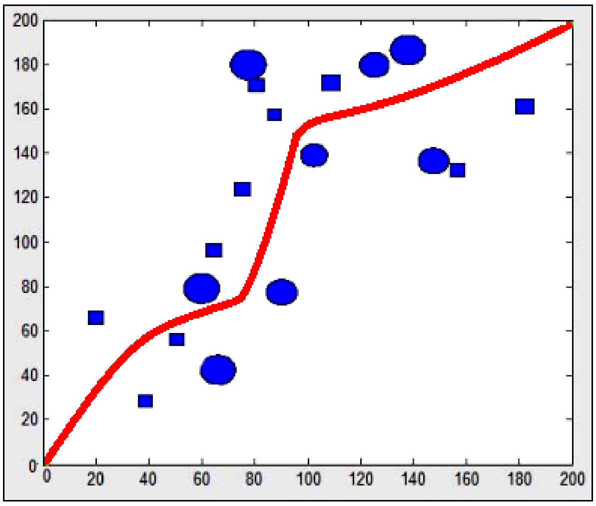


**Fig. 8.** Obstacle avoidance by a fuzzy robot

As is seen from the result, a robot using a fuzzy controller optimized controller has successfully managed to avoid obstacles placed in various directions.

For example, if the distance between the front and left sensor of the robot and the obstacle is small, while that is far from the right sensor, then the sensor rotates to the right. Furthermore, speed also changes depending on the distance between the obstacle and the sensor. If the distance decreases, then speed must also decrease proportionately for a smooth stop action.

## Conclusions

In this paper, the fuzzy control approach has been considered to model the robot with an obstacle avoidance control system based on three ultrasonic wave sensors under unknown situations. In the model, 25 fuzzy rules of three inputs and two outputs are designed for the fuzzy controller with all inputs described by three linguistic variables. Rotation angles are defined with seven states for the movement.

Simulations present that this method can efficiently realize autonomous obstacle avoidance in an unknown environment. A robot can change its speed and rotation angle depending on the position of obstacles.

Furthermore, resultant models of the fuzzy optimized controller demonstrate the superior performance of the fuzzy controller with high adaptability to the environment while maintaining a sufficient level of accuracy.

## REFERENCES

1. Yan, H., Zhu, Q., Zhang, Y., Li, Z. and Du, X. (2022), "An Obstacle Avoidance Algorithm for Unmanned Surface Vehicle Based on A Star and Velocity-Obstacle Algorithms", *2022 IEEE 6th Information Technology and Mechatronics Engineering Conference* (ITOEC), pp. 77-82, DOI: https://doi.org/10.1109/ITOEC53115.2022.9734642.
2. Ribeiro, T., Gonçalves, F., Garcia, I., Lopes, G. and Ribeiro, A. F. (2019), "Q-Learning for Autonomous Mobile Robot Obstacle Avoidance", *2019 IEEE International Conference on Autonomous Robot Systems and Competitions* (ICARSC), pp. 1-7, DOI: https://doi.org/10.1109/ICARSC.2019.8733621.
3. Liu, Y., Chen, D. and Zhang, S. (2018), "Obstacle avoidance method based on the movement trend of dynamic obstacles", *2018 3rd International Conference on Control and Robotics Engineering* (ICCRE), pp. 45-50, DOI: https://doi.org/10.1109/ICCRE.2018.8376431.
4. Nan, J. (2021), "Research on Robot Obstacle Avoidance System Based on Computer Path Planning", *2021 IEEE International Conference on Power Electronics, Computer Applications* (ICPECA), pp. 909-913, DOI: https://doi.org/10.1109/ICPECA51329.2021.9362699.
5. Cui, W. (2019), "Multi-sensor Information Fusion Obstacle Avoidance based on Fuzzy Control", *2019 IEEE 2nd International Conference on Automation, Electronics and Electrical Engineering* (AUTEEE), pp. 198-202, DOI: https://doi.org/10.1109/AUTEEE48671.2019.9033349.
6. Arif, M. U. and Marzoughi, A. (2020), "Arithmetic mean-based decision-making algorithm for obstacle avoidance and multiple intruder detection", *2020 Australian and New Zealand Control Conference* (ANZCC), pp. 103-107, DOI: https://doi.org/10.1109/ANZCC50923.2020.9318343.
7. Babinski, D., Berisha, J., Zaev, E. and Bajrami, X. (2020), "Application of Fuzzy Logic and PID Controller for Mobile Robot Navigation", *2020 9th Mediterranean Conference on Embedded Computing* (MECO), pp. 1-4, DOI: https://doi.org/10.1109/MECO49872.2020.9134317.
8. Allagui, N. Y., Abid, D. B. and Derbel, N. (2019), "Autonomous navigation of mobile robot with combined fractional-order PI and fuzzy logic controllers", *2019 16th International Multi-Conference on Systems, Signals & Devices* (SSD), pp. 78-83, DOI: https://doi.org/10.1109/SSD.2019.8893176.
9. Waga, A., Laminin, C., Benhlima, S. and Bekri, A. (2021), "Fuzzy logic obstacle avoidance by an NAO robot in an unknown environment", *2021 Fifth International Conference On Intelligent Computing in Data Sciences* (ICDS), pp. 1-7, DOI: https://doi.org/10.1109/ICDS53782.2021.9626718.
10. Zhu, Z., Xie, J. and Wang, Z. (2019), "Global Dynamic Path Planning Based on Fusion of A Algorithm and Dynamic Window Approach", *2019 Chinese Automation Congress* (CAC), pp. 5572-5576, DOI: https://doi.org/10.1109/CAC48633.2019.8996741.
11. Park, S. H. and Kim, G. W. (2014), "Expanded Guide Circle-based obstacle avoidance for the remotely operated mobile robot", *Journal of Electrical Engineering and Technology*, vol. 9, no. 3, pp. 1034-1042.
12. Kim, D. G. and Kim, G. W. (20160, "Multi-Expanded Guide Circle-based Obstacle Avoidance for the Remotely Operated Mobile Robot", *2016 Conference on Information and Control Systems*, pp. 84-85.
13. Lee, D. Y., Lu, Y. F., Kang, T. K., Choi, I. H. and Lim, M. T. (2012), "3D vision-based local obstacle avoidance method for a humanoid robot", *2012 12th International Conference on Control, Automation and Systems*, pp. 473-475.

14. Luo, R. C. and Kuo, C. (2016), “Intelligent Seven-DoF Robot With Dynamic Obstacle Avoidance and 3-D Object Recognition for Industrial Cyber-Physical Systems in Manufacturing Automation”, *Proceedings of the IEEE*, vol. 104, no. 5, pp. 1102-1113, May 2016, DOI: https://doi.org/10.1109/JPROC.2015.2508598.
15. Song, K., Chang, Y. and Chen, J. (2019), "3D Vision for Object Grasp and Obstacle Avoidance of a Collaborative Robot", *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics* (AIM), pp. 254-258, DOI: https://doi.org/10.1109/AIM.2019.8868694.
16. Wehrmann, D., Hildebrandt, A. -C., Wittmann, R., Sygulla, F., Rixen, D. and Buschmann, T. (2016), "Fast object approximation for real-time 3D obstacle avoidance with biped robots", *2016 IEEE International Conference on Advanced Intelligent Mechatronics* (AIM), pp. 38-45, DOI: https://doi.org/10.1109/AIM.2016.7576740.
17. Li, Y. and Liu, Y. (2020), "Vision-based Obstacle Avoidance Algorithm for Mobile Robot", *2020 Chinese Automation Congress* (CAC), pp. 1273-1278, DOI: https://doi.org/10.1109/CAC51589.2020.9326906.
18. Nagarajan, V. R. and Singh, P. (2021), "Obstacle Detection and Avoidance For Mobile Robots Using Monocular Vision", *2021 8th International Conference on Smart Computing and Communications* (ICSCC), pp. 275-279, DOI: https://doi.org/10.1109/ICSCC51209.2021.9528162.
19. Ghorpade, D., Thakare, A. D. and Doiphode, S. (2017), "Obstacle Detection and Avoidance Algorithm for Autonomous Mobile Robot using 2D LiDAR," *2017 International Conference on Computing, Communication, Control and Automation* (ICCUBEA), pp. 1-6, DOI: https://doi.org/10.1109/ICCUBEA.2017.8463846.
20. Padgett, S. T. and Browne, A. F. (2017), "Vector-based robot obstacle avoidance using LIDAR and mecanum drive", *SoutheastCon 2017*, pp. 1-5, DOI: https://doi.org/10.1109/SECON.2017.7925312.

Received (Надійшла) 14.02.2022
Accepted for publication (Прийнята до друку) 27.04.2022

## ABOUT THE AUTHORS / ВІДОМОСТІ ПРО АВТОРІВ

**Худавердієва Махаббат** – завідувачка навчальної лабораторії, здобувач PhD, кафедра приладобудування, Азербайджанський державний університет нафти і промисловості, Баку, Азербайджан;
**Mahabbat Khudaverdiyeva** – Head of the teaching laboratory, candidate for PhD, Instrumentation Engineering Department, Azerbaijan State Oil and Industry University, Baku, Azerbaijan;
e-mail: Khudaverdieva62@mail.ru, ORCID ID: http://orcid.org/0000-0002-5090-4628.

### Моделювання мобільного робота з вибігом перешкод з використанням нечіткого контролера

М. А. Худавердієва

**Анотація.** У цій статті наведено моделювання навігації робота з використанням ультразвукових датчиків в умовах невизначеності. Робот намагається уникнути перешкод, використовуючи контролер нечіткої логіки для обробки даних, що надходять від трьох ультразвукових датчиків. Для оцінки продуктивності оптимізованого на основі нечіткої логіки навігаційного контролера робота з ультразвуковими датчиками, які вимірюють відстань, обчислюючи час знаходження об'єкта і його повернення, перешкоди розміщують попереду, ліворуч і праворуч від робота. Система нечітких міркувань МАМДАНІ використовується для розробленого контролера через його інтуїтивно зрозумілі властивості і меншу кількість параметрів налаштування, що скорочує час, що витрачається на програмування контролера. Вважається, що 25 правил охоплюють можливу взаємодію робота з перешкодами. Для легкого розуміння архітектури навігації та швидкої реалізації алгоритму у цій статті розроблено середовище моделювання MATLAB. MATLAB/Simulink - один з кращих інструментів моделювання, необхідних для проектування архітектури та перевірки алгоритмів з обмеженнями в реальному часі. Результуючі моделі нечіткого оптимізованого контролера демонструють чудову продуктивність нечіткого логічного контролера з високою адаптованістю до навколишнього середовища при збереженні достатнього рівня точності. Розроблений нечіткий контролер може бути використаний у роботах на базі мікропроцесора/мікроконтролера завдяки простоті реалізації та кодування.

**Ключові слова:** нечіткий контролер; навігація робота; датчик; середовище MATLAB.

### Моделирование мобильного робота с избеганием препятствий с использованием нечеткого контроллера

М. А. Худавердиева

**Аннотация.** В статье представлено моделирование навигации робота с использованием ультразвуковых датчиков в условиях неопределенности. Робот пытается избежать препятствий, используя контроллер нечеткой логики для обработки данных, поступающих от трех ультразвуковых датчиков. Для оценки производительности оптимизированного на основе нечеткой логики навигационного контроллера робота с ультразвуковыми датчиками, которые измеряют расстояние, вычисляя время нахождения объекта и его возвращения, препятствия размещают впереди, слева и справа от робота. Система нечетких рассуждений МАМДАНИ используется для разработанного контроллера из-за его интуитивно понятных свойств и меньшего количества параметров настройки, что сокращает время, затрачиваемое на программирование контроллера. Считается, что 25 правил охватывают возможные взаимодействия робота с препятствиями. Для легкого понимания архитектуры навигации и быстрой реализации алгоритма в этой статье разработана среда моделирования MATLAB. MATLAB/Simulink — один из лучших инструментов моделирования, необходимых для проектирования архитектуры и проверки алгоритмов с ограничениями в реальном времени. Результирующие модели нечеткого оптимизированного контроллера демонстрируют превосходную производительность нечеткого логического контроллера с высокой адаптируемостью к окружающей среде при сохранении достаточного уровня точности. Разработанный нечеткий контроллер может быть использован в роботах на базе микропроцессора/микроконтроллера благодаря простоте реализации и кодирования.

**Ключевые слова:** нечеткий контроллер; навигация робота; датчик; среда MATLAB.