**Indonesian Journal of Electrical Engineering and Computer Science**
Vol. 31, No. 2, August 2023, pp. 725~733
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v31.i2.pp725-733

# Multi-layer perceptron neural network mobile robot navigator in unknown environment

**Mohammed Rabeea Hashim Al-Dahhan[1,2], Ruqayah Rabeea Al-Dahhan[2], Ali Tariq Radeef[3]**
[1]Department of Computer Engineering Techniques, Al-Maarif University College, Ramadi, Anbar, Iraq
[2]College of Computer Science and Information Technology, Anbar University, Ramadi, Anbar, Iraq
[3]Department of Computer Science, Al-Maarif University College, Ramadi, Anbar, Iraq

**Article Info**

***Article history:***

Received Jan 5, 2023
Revised Apr 10, 2023
Accepted Apr 16, 2023

***Keywords:***

Mobile robot
Navigation
Neural network
Obstacle avoidance
Target seeking

**ABSTRACT**

Recently, navigation in an unknown environment without hitting obstacles was considered a big challenge faced by researchers. The difficulty in finding a good mathematical model for the different systems is deciding to use artificial intelligent controllers to control the mobile robot movement. In this paper, designing two multi-layer-perceptron neural networks (MLP-NN) was done to control the movement of mobile robots in an unknown environment. The first MLP-NN is to control the linear velocity on the x-axis and angular velocity of the robot's movement while the other MLP-NN is designed to avoid the static and dynamic obstacles faced by the robot while navigating in an unknown environment. The results show each controller's advantages in performing navigation tasks and avoiding obstacles in different environments.

*This is an open access article under the CC BY-SA license.*

***Corresponding Author:***

Mohammed Rabeea Hashim Al-Dahhan
Department of Computer Engineering Techniques, Al-Maarif University College
31000, Ramadi, Anbar, Iraq
Email: alany.moahmmed5@gmail.com

## 1. INTRODUCTION

Initially, the term mobile robots were not used within the industrial and manufacturing field only. In the last decade started using it in the fields of medicine, rescuing, entertainment, education, agriculture, mining, military, and many more. In literature, different methods are used for safe navigation problems such as Logic Controller with Mamdani style [1] to navigate in static and dynamic environments without any collision. However, the shortest and safest path planning of a mobile robot in a static environment with different shaped obstacles by using A* is used in [2]. The path which results from this algorithm is a safe path with a large distance to obstacles but the robot cannot turn smoothly due to the sharp corner generated by the A* algorithm in the path. Fuzzy logic controller (FLC) is used in [3], [4] for finding the shortest and safest path and navigating in an unknown environment with different dynamic and static obstacles. Moreover, the type of robot used in that research has many constraints because it could not move in all directions without adjusting its rotation angle. Many researchers include the non-holonomic type of Robot in their work such as [5], [6]. Rao *et al.* [5] suggested neuro-fuzzy logic with integrated safe boundary algorithm to navigate safely using a non-holonomic robot. Tutuko *et al.* [6] and Patle *et al.* [7] proposed probability-based fuzzy logic for solving the path planning problems of navigation and using a non-holonomic mobile robot called "Khepera-II robot". The result of their work was limited due to the static environment only. Rubio *et al.* [8] used a combination of dynamic particle swarm optimization (DPSO) with fuzzy logic (FL) to enhance the performance of the navigation for a non-holonomic mobile robot by controlling the trajectory and reducing the time traveled by the predefined target. Within the vast developments in technology, autonomous and robotics [8] many researchers are interested in autonomous driving [9], [10]. SlamThe algorithm has been used in [11] to achieve the 2D mapping, where they used an omni-robot with an X4 liDAR sensor as an indoor scanner. The main benefit of these sensors is to determine the position of the obstacle. The results show the robot can follow the walls using the proportional-integral-derivative (PID) controller. Hani and Moin [12] slam algorithms are used to introduce the environment as static or dynamic to get prior knowledge about the environment. The results show that the algorithm is reliable for real-time applications.

Navigation in different environments [13], [14], routing [15], buildings [16], industrial [17], and unmanned aerial vehicle (UAV) [18], [19]. Mobile robot has to navigate in different environments without hitting any obstacles on their route. In static environments [20], [21]. On the other hand, binary images can be used as input to navigate the environment [22], [23]. The left-hand algorithm is used in [24] to make the robot capable to follow the line in the maze map without hitting the obstacles in a static environment. On the other hand, Dagher *et al*. [25] also used a static environment to control the navigation of a 3WD mobile robot using a neural network. Shamsuddin *et al*. [26] present many algorithms for the stability of autonomous ground and unmanned vehicles. However, this work reviewed more than 50 papers talking about unmanned surface vehicle (USV) and AVG performance. Omni-directional mobile robot in a room map environment present in [27], Raspberry Pi 3 B used to connect the light detection and ranging (LiDAR) sensor with a robot to control the movement of the robot manually or automatically, as a result, the robot is capable to follow the wall in indoor room environments. Improving the robot’s movement in avoiding obstacles was done [28] using neuro-fuzzy in this work led to enhance response and decrease time response. Discovering the path in any maze map using MATLAB was presented in [29], where the robot follows the walls to find the correct path between the start point and end point. The algorithm shows good results but it has suffered from time consumption. Generating an optimal path between two points in the static environment and then minimizing the points of the path to make it smoother were demonstrated in [30]. An artificial neural network is used to control the torque required for the mobile robot to follow the path.

The contribution of this work is to design and implement a controller for a mobile robot without knowing its mathematical model, then and finding the shortest path between the start and goal points on the real-environment map, also this algorithm helps the robot to avoid the static and dynamic obstacles. as a result, the robot can navigate safely in an unknown environment with less time computation by achieving the shortest path. The method will be discussed in the next part, section three will discuss the results and section four will conclude the paper's goals.

## 2. METHOD

The most important metric in the navigation of a mobile robot in any environment is finding the shortest path safely, which means that the mobile robot is capable to avoid all the static and dynamic obstacles on its route to reach the goal. This section describes the design of a neural network (NN) controller to control mobile robot navigation in unknown environments. However, we design two NN one of them for controlling the linear and angular velocity, while the other NN is used to avoid the obstacles on mobile robot’s routes as:

### 2.1. Multi-layer perceptron neural network

It can create a complicated decision area. The nonlinearity used within its nodes gives multi-layer-perceptron neural networks (MLP-NN) its capacity. A multi-layer perceptron neural network is depicted in Figure 1. Whereas,

- Input Layer: it obtains data from outside sources and sends it to the network for processing.
- Hidden Layer: the hidden layer gathers and processes information from the input layer. It is connected to the input and output layers directly.
- Output layer: it receives information that has been processed and exits the MLP structure with output signals.

Algorithm for learning the generalized delta rule, following is a summary of the generalized delta rule learning algorithm:

Step 1: To ensure learning stability, initialize the weights $W_1$, $W_2$...We and threshold to small values.
Step 2: Presents the desired output DK and the inputs $X_1$, $X_2$... Xn.
Step 3: Using (1), determine the actual output:

$$Y\mathrm{k} = Fn \sum_{i=1}^{n}(XiWi - \theta_k) \quad (1)$$

Step 4: Update the weights by (1):

$$Wi\ (new) = Wi\ (old) + \eta\ (dk - yk)\ xi,\ 0 \leq i \leq N \tag{2}$$

where,
η: positive learning rate fraction that is less than 1.
$dk$: desirable output
$\theta k$: the internal threshold of the node
Step 5: Repeat steps 2, 3, and 4 until the mission is done. Either by meeting the desired stopping criterion of a half-square error or by completing the specified number of iterations.

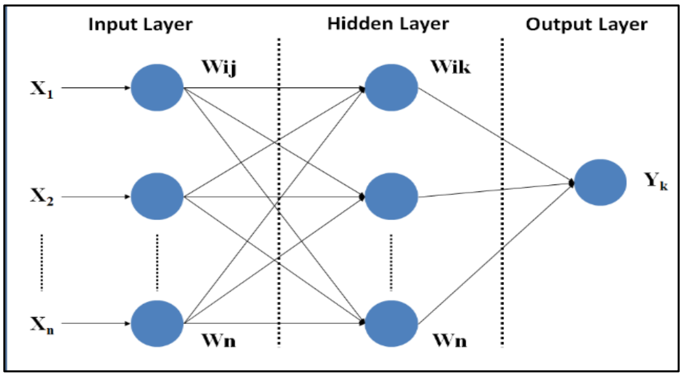


Figure 1. Multi-layer perceptron neural network

### 2.2. Path tracking control using NN

Designing multi-layer perceptron NN using the backpropagation algorithm to achieve the liner velocity on x-direction ($V_x$) and angular Velocity (*ω)* has:

- 2 nodes topology in the input layer as an input to the controller
- 5 nodes topology in the hidden layer
- 2 nodes topology in the output layer as an output velocity to the mobile robot in the *X* direction in (*ω).*

The NN has two inputs, one of them for finding the required distance from the instant position to the target. This calculation can be found by using (3). Another input for the second node is used to achieve the required rotation to make the mobile robot face the goal, after achieving this angle, the mobile robot can go directly to the goal. Calculating the required angle can be found using (4). The output data goes directly to the hidden layers.

$$R = \sqrt{\Delta x^2 + \Delta y^2} \tag{3}$$

$$\theta_e = atan\, 2\frac{\Delta y}{\Delta x} \tag{4}$$

Where, $\Delta x = (Xd - Xa)$ and it’s explained the desired position of the mobile robot in the environment while $\Delta y = (yd - ya)$ shows the actual position. The resultant *R* calculates the required distance between the actual position and desired position. The required rotation of the robot to face its predefined target can be calculated using equation four. In the second part of NN, there are 5 nodes topology trained within 400 iterations, as a result, the accurate data will send to the output nodes to make the required action according to the instant position of the mobile robot. The final part of the NN controller is the output layers, it has two nodes’ topology, one of them to derive the mobile robot while the other node is used for controlling the angular velocity. The overall shape of NN is shown in Figure 2. Where Xe and Ye define the differences between the desired and actual axis as follows: Xe = (Xd - Xe), Ye = (Yd - Ye).

### 2.3. Obstacle avoidance using NN

This section shows the design of multi-layer perceptron NN using a backpropagation algorithm to achieve the linear velocity in the *Y* direction ($V_y$). The second NN has the specification:

- 3 nodes topology in the input layer as an input to the controller
- 5 nodes topology in the hidden layer
- 1 node topology in the output layer as an output velocity to the mobile robot in the *Y* direction.

The overall NN network is presented in Figure 3.

Three infrared sensors (IR-1, IR-2, and IR-3) are put in Infront of the robot chassis, the reading of these sensors will be the input to the NN controller, and in the hidden part of the second NN the data will be processed to give a suitable action to the mobile robot, the final part of the NN is the linear velocity on Y-axis. The learning of this NN took 500 iterations to make it capable of dealing with static and dynamic obstacles and giving suitable action in an accurate time.

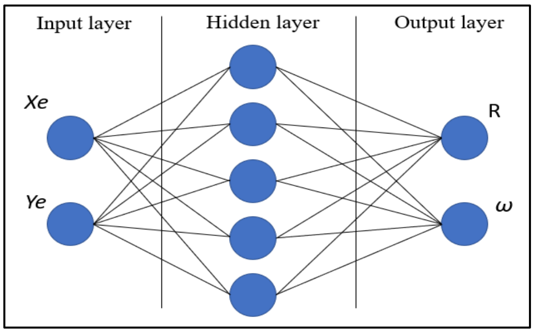


Figure 2. MLP-NN for path tracking

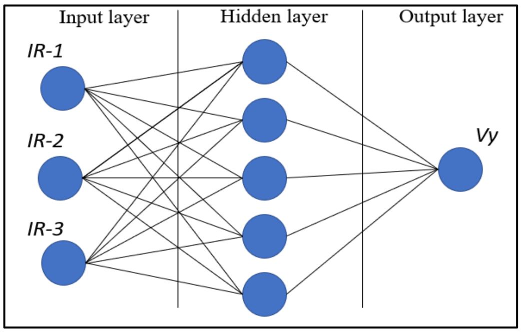


Figure 3. MLP-NN for obstacle avoidance

## 3. RESULTS AND DISCUSSION

In this section we are presenting two different experiments that were done in real-time, the first one is to test the effectiveness of the algorithm to drive the robot from the start position to the goal position in a free-obstacle environment. Here we used only the first NN for path tracking controller to drive the robot in the real-time. While the second experiment is to test both NN for path tracking and obstacle avoidance controllers in driving the robot and avoiding the dynamic obstacle on its route. Both experiments success to move the robot safely and achieve the goal.

### 3.1. First scenario

Free obstacle environment as in Figure 4 used to test the NN controller, the start location is (0, 0) and the goal location is (3,000 mm, 2,500 mm). In the beginning, the mobile robot will turn to achieve $\theta e$, when $\theta e$ is equal to zero this is mean that the robot faces its goal and can go directly to reach the pre-defined target. Figure 5 shows the real movement of the mobile robot on the XY-axis axis. As is clear, the x-axis “black line” explains the samples used during the movement, while-axis the axis “red line” explains their position respectively. The interval from (0 to 182 samples) shows how the robot turns to achieve zero $\theta e$, after that the faces face the target starts moving forward to its goal at 1,280 samples. These data are collected from the odometry of the mobile robot. The normal read in the IR sensor should be between (0 – 0 volts output it jumped up to 1 V which indicates that the sorcerer detects an obstacle witch away from it. It’s clear that the sensors did not detect any obstacle, the reading of the three IR sensors shown in Figure 6.

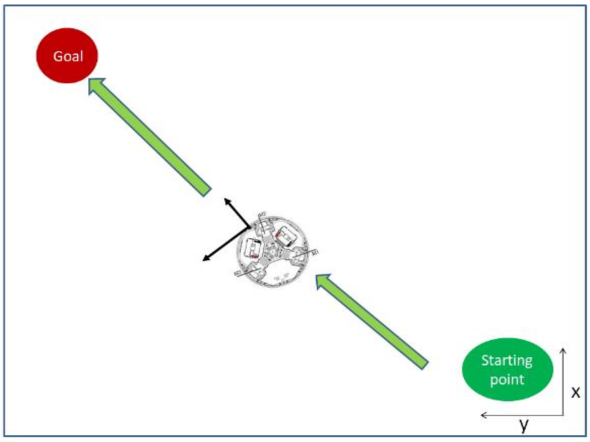


Figure 4. Free obstacle environment

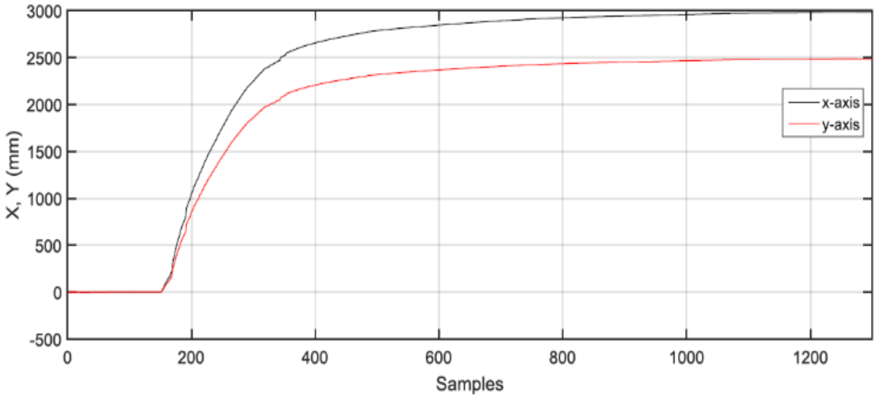


Figure 5. Time samples on the X and Y-axis

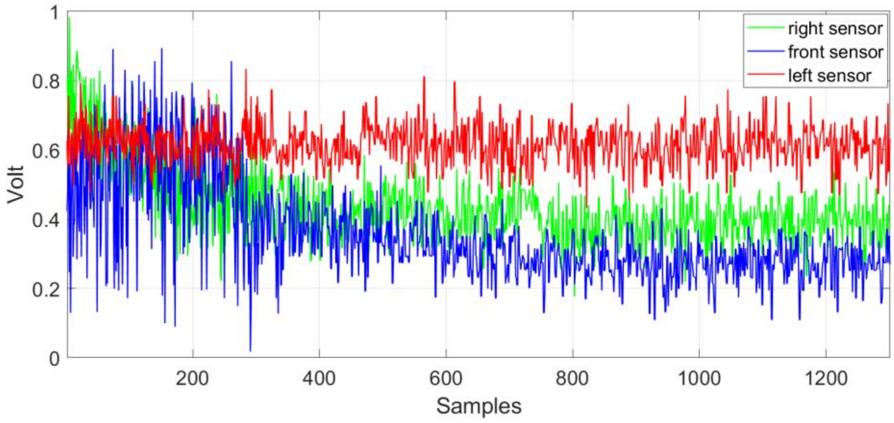


Figure 6. IR sensor readings

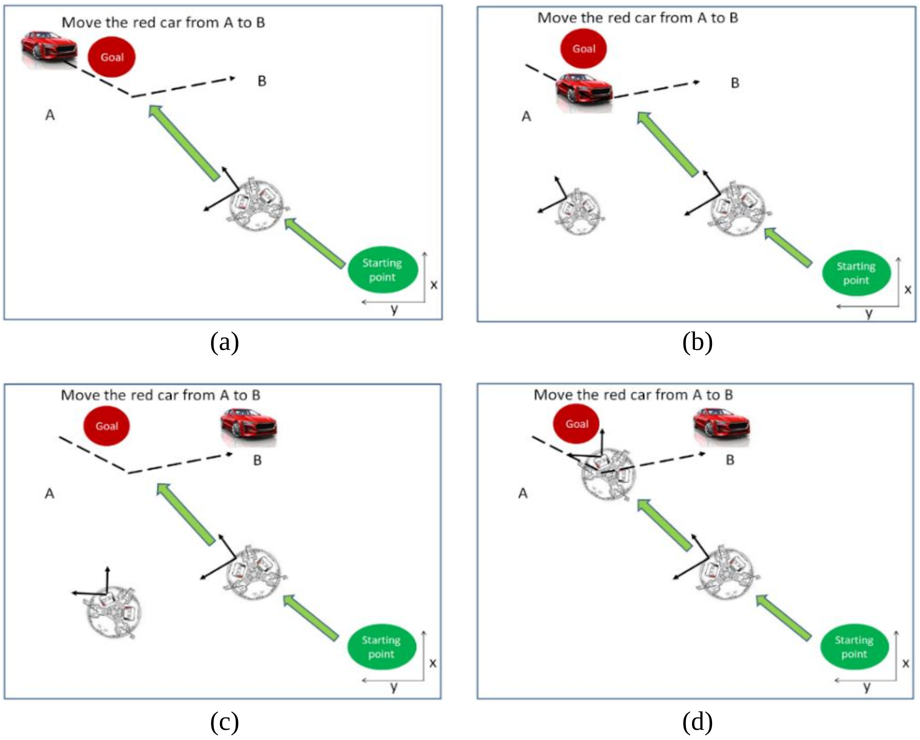
### 3.2. Second scenario

The challenge of the NN controller is to move the mobile robot in a dynamic environment safely, Figure 7 shows the overall scenario for the first obstacle, where in Figure 7(a) explains the second environment used to test the controllers. After the mobile robot faces the target and starts moving forward to reach it, a small gaming car starts moving toward the robot trying to hit it as in Figure 7(b), the sensors detect the obstacle and moved the robot to the left side to avoid the obstacle, and when the gaming car went away as in Figure 7(c), the robot turned to achieve zero $\theta e$, then the robot moves to its goal as shown in Figure 7(d).

Figure 8 shows Another gaming car comes to hit the robot, again the robot sensors detect it and kept the robot safe by moving it to the right side Figures 8(a) and 8(b), after finishing the effect of the obstacle, the mobile robot again turns to achieve zero $\theta e$, this is mean that the robot faced the goal. Then the robot moves to reach the goal safely as shown in Figure 8(c) and 8(d).

The response isseis presented in Figure 9 from the beginning to the f the movement. The robot turns and reaches zero $\theta e$ at point 253 o the samples, head, and the robot starts moving to the goal. The robot changes its position at 600 samples to avoid the first obstacle which is the " red gaming car", the effect of the first obstacle end at point 800 sample then gets zero $\theta e$ back, this is pointing the robot start moving to the goal, when another dynamic obstacle come to hit the robot " yellow gaming car" at point 1,800 of samples, agin the robot changes its position to the right side to avoid the second obstacle. After that, the robot went back to its pre-defined goal at that 2,100 samples and moved to its goal, then stopped at 3,500 samples. All the robot movement is recorded in Figure 9.

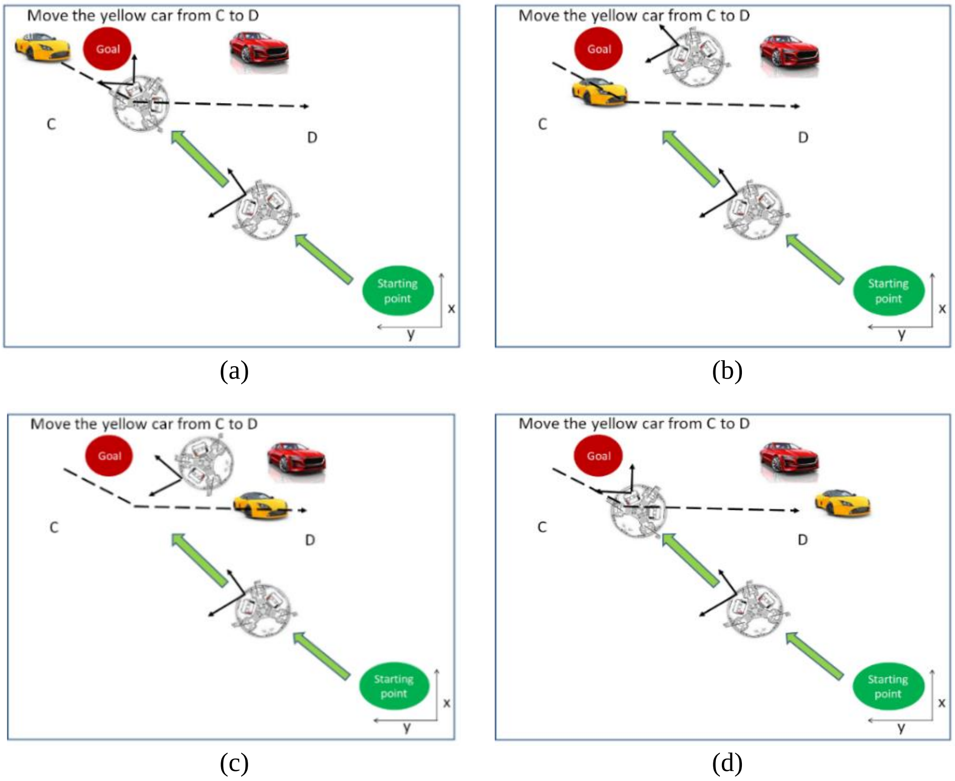
The reading of IR sensors to avoid the obstacles is shown in Figure 10. First, the reading of the right sensor at time response 600 goes to more than 1 V means there is an obstacle, after the movement of the robot away from it, the reading comes back to less than 1 V. the same thing happened with the robot when the second obstacle occurs, it is detected from the left sensor, as a result, the robot changed its location again.


Move the red car from A to B
Goal
A
B
Starting point
x
y
Move the red car from A to B
Goal
A
B
Starting point
x
y
Move the red car from A to B
Goal
A
B
Starting point
x
y
Move the red car from A to B
Goal
A
B
Starting point
x
y

(a) (b)

(c) (d)

Figure 7. Dynamic obstacle environment contain dynamic obstacle; (a) overall environment, (b) first obstacle faced by the robot, (c) the obstacle moved away and to robot turns to Its goal, and (d) the robot moves again to the goal try to reach it


Move the yellow car from C to D
Goal
C
D
Starting point
x
y
Move the yellow car from C to D
Goal
C
D
Starting point
x
y
Move the yellow car from C to D
Goal
C
D
Starting point
x
y
Move the yellow car from C to D
Goal
C
D
Starting point
x
y

(a) (b)

(c) (d)

Figure 8. Dynamic obstacle environment contain two obstacle; (a) the second obstacle occurs to word the robot, (b) the algorithm drives the robot away, (c) the robot rotates to face the goal, and (d) the robot reaches the goal after skipping the obstacle

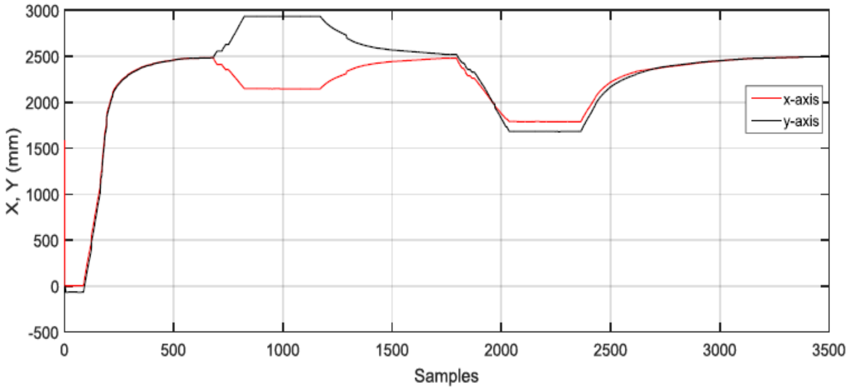


Figure 9. Time response on the X and Y axis

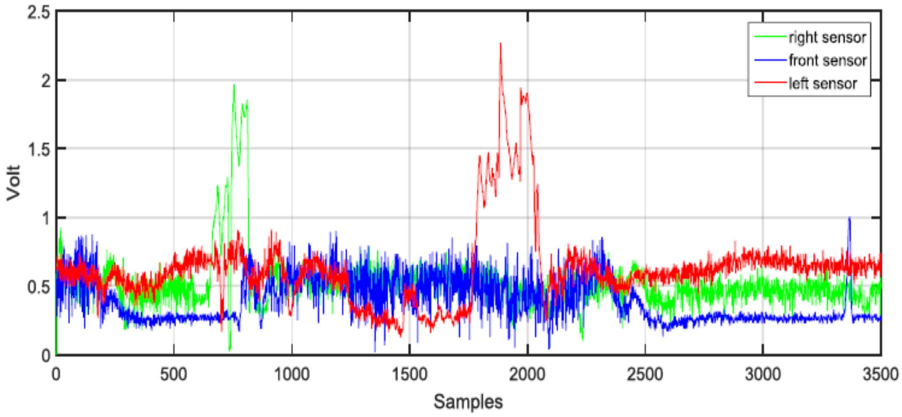


Figure 10. IR sensors reading

## 4. CONCLUSION

The subject of this paper is to find the shortest path for a mobile robot between points in an unknown environment without hitting obstacles on its route. An omnidirectional mobile robot was used to test the effectiveness of the proposed algorithm, MLP-NN with 5 hidden layers was trained in offline mode and used to drive the robot on the x-axis and get the correct theta. Another MLP-NN is used to keep the robot safe during navigation. Each NN was trained to perform the required task. The online experiments show how the robot finds the goal and avoids the obstacles on the route. The proposed controllers are currently formulated for dynamic environments. Three IR sensors are mounted on the chassis to detect the obstacles 30 cm away from the robot. Our future work will focus on the type of sensors to detect the obstacles and choose a more suitable decision regarding the position of the mobile robot, one of the proposed solutions is to replace the IR with ultrasonic sensors or a camera.

## ACKNOWLEDGEMENTS

This work was supported and funded by Al-Maarif University College under grant (DCS - 412).

## REFERENCES

[1] M. R. H. Al-Dahhan and M. M. Ali, “Path tracking control of a mobile robot using fuzzy logic,” in *13th International Multi-Conference on Systems, Signals and Devices, SSD 2016*, Mar. 2016, pp. 82–88, doi: 10.1109/SSD.2016.7473656.

[2] H. Mujtaba, “Safe Navigation of Mobile Robot Using A* Algorithm,” *International Journal of Applied Engineering Research*, vol. 12, pp. 5532–5539, 2017, [Online]. Available: http://www.ripublication.com.

[3] M. Boujelben, C. Rekik, and N. Derbel, “A reactive approach for mobile robot navigation in static and dynamic environment using fuzzy logic control,” *International Journal of Modelling, Identification and Control*, vol. 27, no. 4, pp. 293–302, 2017, doi: 10.1504/IJMIC.2017.084722.

[4] M. Boujelben, D. Ayedi, C. Rekik, and N. Derbel, “Fuzzy logic controller for mobile robot navigation to avoid dynamic and static obstacles,” in *2017 14th International Multi-Conference on Systems, Signals & Devices (SSD)*, Mar. 2017, vol. 2017-Janua, pp. 293–298, doi: 10.1109/SSD.2017.8166963.

[5] A. M. Rao, K. Ramji, B. S. K. S. S. Rao, V. Vasu, and C. Puneeth, “Navigation of non-holonomic mobile robot using neuro-fuzzy logic with integrated safe boundary algorithm,” *International Journal of Automation and Computing*, vol. 14, no. 3, pp. 285–294, Jun. 2017, doi: 10.1007/s11633-016-1042-y.

[6] B. Tutuko, S. Nurmaini, Saparudin, and P. Sahayu, “Route optimization of non-holonomic leader-follower control using dynamic Particle Swarm Optimization,” *IAENG International Journal of Computer Science*, vol. 46, no. 1, 2019.

[7] B. K. Patle, D. R. K. Parhi, A. Jagadeesh, and S. K. Kashyap, “Application of probability to enhance the performance of fuzzy based mobile robot navigation,” *Applied Soft Computing Journal*, vol. 75, pp. 265–283, Feb. 2019, doi: 10.1016/j.asoc.2018.11.026.

[8] F. Rubio, F. Valero, and C. Llopis-Albert, “A review of mobile robots: Concepts, methods, theoretical framework, and applications,” *International Journal of Advanced Robotic Systems*, vol. 16, no. 2, 2019, doi: 10.1177/1729881419839596.

[9] S. Spanogianopoulos and K. Sirlantzis, “Car-like mobile robot navigation: A survey,” in *Studies in Computational Intelligence*, vol. 627, 2016, pp. 299–327.

[10] S. G. Tzafestas, “Mobile robot control and navigation: A global overview,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 91, no. 1, pp. 35–58, 2018, doi: 10.1007/s10846-018-0805-9.

[11] M. Rivai, D. Hutabarat, and Z. M. J. Nafis, “2D mapping using omni-directional mobile robot equipped with LiDAR,” *Telkomnika (Telecommunication Computing Electronics and Control)*, vol. 18, no. 3, pp. 1467–1474, Jun. 2020, doi: 10.12928/TELKOMNIKA.v18i3.14872.

[12] U. Hani and L. Moin, “Realtime autonomous navigation in V-Rep based static and dynamic environment using EKF-SLAM,” *IAES International Journal of Robotics and Automation (IJRA)*, vol. 10, no. 4, p. 296, Dec. 2021, doi: 10.11591/ijra.v10i4.pp296-307.

[13] M. Hoy, A. S. Matveev, and A. V. Savkin, “Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey,” *Robotica*, vol. 33, no. 3, pp. 463–497, Mar. 2015, doi: 10.1017/S0263574714000289.

[14] I. Noreen, A. Khan, H. Ryu, N. L. Doh, and Z. Habib, “Optimal path planning in cluttered environment using RRT*-AB,” *Intelligent Service Robotics*, vol. 11, no. 1, pp. 41–52, Jan. 2018, doi: 10.1007/s11370-017-0236-7.

[15] L. G. D. O. Veras, F. L. L. Medeiros, and L. N. F. Guimaraes, “Systematic literature review of sampling process in rapidly-exploring random trees,” *IEEE Access*, vol. 7, pp. 50933–50953, 2019, doi: 10.1109/ACCESS.2019.2908100.

[16] D. G. MacHaret and M. F. M. Campos, “A survey on routing problems and robotic systems,” *Robotica*, vol. 36, no. 12, pp. 1781–1803, Dec. 2018, doi: 10.1017/S0263574718000735.

[17] Y. Dong, E. Camci, and E. Kayacan, “Faster RRT-based Nonholonomic Path Planning in 2D Building Environments Using Skeleton-constrained Path Biasing,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 89, no. 3–4, pp. 387–401, 2018, doi: 10.1007/s10846-017-0567-9.

[18] S. Agnisarman, S. Lopes, K. C. Madathil, K. Piratla, and A. Gramopadhye, “A survey of automation-enabled human-in-the-loop systems for infrastructure visual inspection,” *Automation in Construction*, vol. 97, pp. 52–76, Jan. 2019, doi: 10.1016/j.autcon.2018.10.019.

[19] R. Cui, Y. Li, and W. Yan, “Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT∗,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 7, pp. 993–1004, Jul. 2016, doi: 10.1109/TSMC.2015.2500027.

[20] G. Jain, G. Yadav, D. Prakash, A. Shukla, and R. Tiwari, “MVO-based path planning scheme with coordination of UAVs in 3-D environment,” *Journal of Computational Science*, vol. 37, p. 101016, Oct. 2019, doi: 10.1016/j.jocs.2019.07.003.

[21] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, Jun. 2011, doi: 10.1177/0278364911406761.

[22] M. G. Mohanan and A. Salgoankar, “A survey of robotic motion planning in dynamic environments,” *Robotics and Autonomous Systems*, vol. 100, pp. 171–185, Feb. 2018, doi: 10.1016/j.robot.2017.10.011.

[23] R. Kala, “Homotopic roadmap generation for robot motion planning,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 82, no. 3–4, pp. 555–575, 2016, doi: 10.1007/s10846-015-0278-z.

[24] B. Tovar, R. Murrieta-Cid and S. M. LaValle, “Distance-Optimal Navigation in an Unknown Environment Without Sensing Distances,” *IEEE Transactions on Robotics*, vol. 23, no. 3, pp. 506-518, Jun. 2007, doi: 10.1109/TRO.2007.898962..

[25] K. E. Dagher, R. A. Hameed, I. A. Ibrahim, and M. Razak, “An adaptive neural control methodology design for dynamics mobile robot,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 2, pp. 392–404, 2022, doi: 10.12928/TELKOMNIKA.v20i2.20923.

[26] P. N. F. M. Shamsuddin, R. M. Ramli, and M. A. Mansor, “Navigation and motion control techniques for surface unmanned vehicle and autonomous ground vehicle: a review,” *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 1893–1904, Aug. 2021, doi: 10.11591/eei.v10i4.3086.

[27] M. S. Aslam, M. I. Aziz, K. Naveed, and U. K. Zaman, “An RPLiDAR based SLAM equipped with IMU for Autonomous Navigation of Wheeled Mobile Robot,” in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, pp. 1-5, 2020, doi: 10.1109/INMIC50486.2020.9318133.

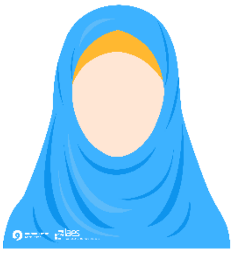
[28] A. A. A. Baker and Y. Ghadi, “Mobile robot controller using novel hybrid system,” *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, p. 1027, Feb. 2020, doi: 10.11591/ijece.v10i1.pp1027-1034.

[29] I. M. Mohammed, M. Z. N. Al-Dabagh, S. J. Rashid, and N. A. M. Isa, “Path discovering in maze area using mobile robot,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 20, no. 2, p. 416, Apr. 2022, doi: 10.12928/telkomnika.v20i2.19408.

[30] Y. Purnama and F. E. Gunawan, “Vibration-based damaged road classification using artificial neural network*,”. TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 16, no. 5, pp. 2179-2190, Oct. 2018, doi: 10.12928/telkomnika.v16i5.7574.

## BIOGRAPHIES OF AUTHORS

**Mohammed Rabeea Hashim Al-Dahhan** received a B.Eng. degree in computer technology engineering from the Al-Maarif College University, Iraq, in 2013, and M.Sc. degree in mechatronics from Philadelphia University, Jordan, in 2015, and the Ph.D. degree in electronic and communication engineering from Çankaya University, Turkey, in 2020. His research interests include robotics and intelligent control. He can be contacted at email: alany.moahmmed5@gmail.com.



**Ruqayah Rabeea Al-Dahhan** received a B.Sc. University of Anbar- College of Computer Science and Information Technology-Iraq-2006, and M.Sc. University of Anbar- College of Computer Science and Information Technology-Iraq-2010, and PhD Liverpool John Moores University-Faculty of Engineering and Technology-United Kingdom-2019. Her research interests include cloud computing, the internet of things, network security, robotics, and information security. She can be contacted at email: ruqayah1985@gmail.com.

**Ali Tariq Radeef** is currently with the Computer Science Department, Al-Maarif University College. His research interests include artificial intelligence, machine learning, and data science. He can be contacted at email: ali.taraq@uoa.edu.iq.