Electrical Engineering Technical Journal, ISSN: ISSN: 3007-9772, Vol. 3, No. 1, 2026, Pages 20-27
DOI: https://doi.org/10.51173/10.51173/eetj.v3i1.46

Electrical Engineering Technical Journal
Journal homepage*:* http://eetj.mtu.edu.iq

RESEARCH ARTICLE

Tracking the Path of a Mobile Robot Using Fuzzy Logic Algorithms

Huda Ali Muhammad [1], and Khosro Khandani [1 *]

[1] Electrical Engineering Department of Mechatronics, Arak University, Iran

[*]Corresponding Author Email: k-khandani@araku.ac.ir

Article Info.	**Abstract**
Article history: Received 02 January 2026 Accepted 26 January 2026 Published in Journal 31 January 2026	This paper presents a structured hybrid control framework for mobile robot navigation and path tracking in uncertain and dynamic environments. The proposed architecture integrates a fuzzy logic controller as the primary decision-making layer with a multilayer perception (MLP) neural network used for nonlinear compensation. The fuzzy module processes real-time sensory inputs from front, left, and right distance measurements and generates the initial steering command using a Mamdani inference system. To enhance tracking accuracy and robustness, the neural network is trained in a supervised manner using the mean squared error (MSE) loss function to learn adaptive correction signals. The final steering command is obtained by combining the fuzzy output with the neural compensation term, forming a hybrid control strategy that preserves interpretability while improving adaptability. The system is implemented in a simulation environment and evaluated under obstacle-rich scenarios. Quantitative performance metrics, including MSE and root mean square error (RMSE), are used to assess tracking accuracy. Experimental results demonstrate improved trajectory precision, reduced tracking error, and enhanced stability compared to standalone fuzzy and neural approaches. The proposed framework maintains clear functional separation between decision-making and compensation layers, improving modularity, transparency, and reproducibility. The results confirm that the integration of fuzzy reasoning with neural compensation provides an effective and computationally efficient solution for autonomous mobile robot navigation in nonlinear and uncertain environments.

This is an open-access article under the CC BY 4.0 license (http://creativecommons.org/licenses/by/4.0/) Publisher: Middle Technical University

Keywords: Mobile Robot; Fuzzy Logic; Path Tracking; Intelligent Control; Autonomous Navigation.

1. Introduction

Intelligent robots are designed to perform autonomous tasks without direct human intervention. Remote control systems enable human–robot interaction without physical presence, offering flexibility in hazardous or inaccessible environments. Efficient navigation algorithms that allow coordinated movement and collision avoidance are key challenges in autonomous robotics [1, 2].

Artificial neural networks (ANNs) replicate human brain learning and decision-making processes. Composed of interconnected neurons, neural networks can retain sensory information and learn via repetition [3, 4]. Neural networks also foster interdisciplinary collaboration across computer science, engineering, and neuroscience, accelerating advancements in practical applications [2, 5].

Robotic systems face challenges in dynamics and inverse dynamics, requiring advanced software to handle complex nonlinear equations [6]. Increased paths and obstacles heighten complexity, leading to trajectory deviations due to environmental uncertainties. Adaptive control techniques, particularly computed torque methods, help mitigate these effects but face limitations in nonlinear and rapidly changing conditions [7, 8]. Linking Python and MATLAB applications is essential to optimize simulation outcomes. Ordered neural network models support efficient, collision-free path planning while accommodating variable obstacle shapes [9, 10].

There is no meaningful correlation between the Vector Field Histogram (VFH) method and ultrasonic sensor-based path detection in mobile robots [11, 12]. This lack of relationship poses challenges for autonomous navigation, particularly in hazardous industrial or military applications. The primary objective of this research is to develop a robotic simulator leveraging neural networks to improve tracking performance of manipulators with unknown dynamics [5, 7]. The system integrates VFH and complementary algorithms to generate appropriate torque signals for precise trajectory following [2, 6].

The study is structured to progress logically from introduction to methodology, simulation, results, discussion, and conclusion, ensuring clarity and coherence in presenting contributions to autonomous robotics [9, 13].

2. Literature Review

Neural networks are widely used to model nonlinear robotic systems by processing inputs through weighted connections, biases, and activation functions [3, 4], with applications including inverse kinematics, dynamic control, and trajectory optimization [6, 7]. Hybrid neuro-fuzzy approaches further improve system adaptability under uncertainty and environmental variation [14, 15], while robots also employ computer vision and neural networks to construct internal representations of their environments for obstacle avoidance and path planning [9, 10]. In social and collaborative contexts, AI systems recognize human behaviors and intentions, enhancing operational safety and compliance [2, 5]. Motion planning defines sequences of movements that allow robots to reach target positions while avoiding obstacles [2, 13], utilizing common algorithmic categories such as path-based algorithms like A* and Dijkstra [1] and behavior-based models that predict motion using dynamic constraints [6]. Key challenges include large data volume, dynamic environments, and real-time execution requirements [9, 10], though these algorithms are applied extensively in industrial robotics, autonomous vehicles, and gaming systems [11, 12].

The Vector Field Histogram (VFH) method enables real-time obstacle avoidance by representing obstacle density in polar histograms [11, 12], while Radial Basis Function Networks (RBFNs) enhance prediction and classification tasks due to their fast learning and approximation capabilities [3, 5]. Embedded robotic systems rely on neural networks to process internal sensors (e.g., encoders) and external sensors (ultrasonic, infrared, laser) for localization and navigation [7, 9], often using multilayer neural networks to compensate for sensor noise and cumulative errors to improve environmental map accuracy [3, 5]. Recent studies demonstrate that hybrid approaches combining neural networks, fuzzy logic, and visual servoing (IBVS/PBVS) significantly improve real-time path tracking and collision avoidance [14, 15]. Finally, experimental evaluations confirm enhanced trajectory accuracy and robustness in dynamic environments [16].

3. Practical Application

In this section, the simulation process was applied practically by connecting the neural network to the artificial intelligence camera to recognize the movement of the robot along the specified paths, as shown in Figure 1.

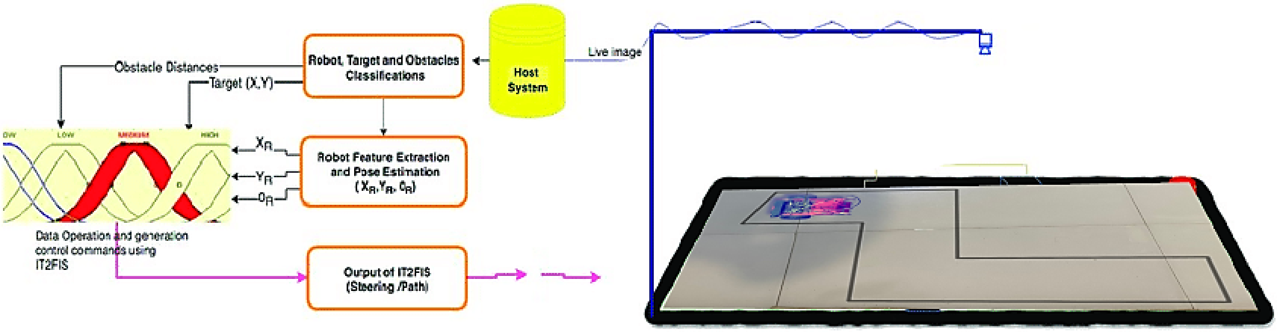


Fig. 1. Determining the path and direction of the robot using the camera.

Figure 1 shows the process of determining the direction and touch of the robot through a camera connected via a Wi-Fi network to the robot, where the data is sent through the pre-implemented simulation system, which determines the correct path to reach the goal [19]. In this study, various methods of route planning and obstacle avoidance of mobile robots are studied. Using these algorithms, a possible path is determined between the starting points and the target in such a way that no collision occurs between the robot and the obstacles, as shown in Figure 2.

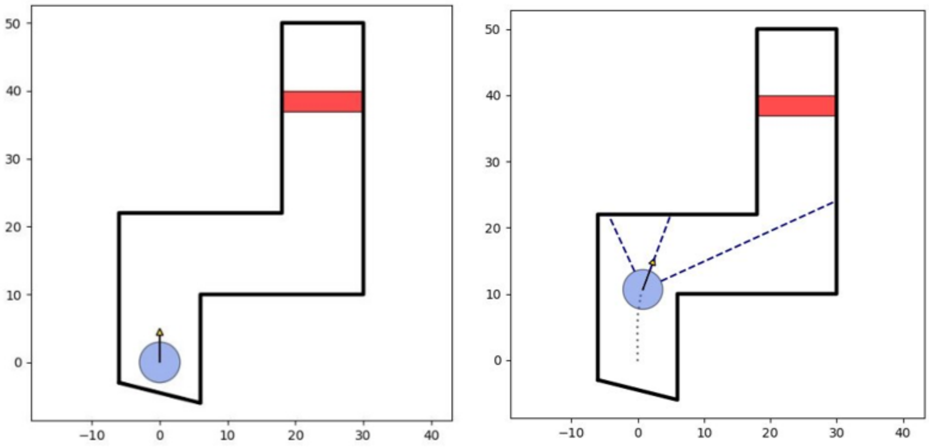


Fig. 2. The mechanism of the robot's movement and how it collides with obstacles during training.

Also, the study that was conducted in the proposed message is considered the best and most important in terms of simulation and design, through which the movement process is carried out in a simulated manner of a real robot movement in terms of design and the proposed environment, as shown in the following figure of the proposed results. Figure 3 demonstrates the mechanism followed by the distance sensor placed in the robot, which can determine whether there is an obstacle or not.

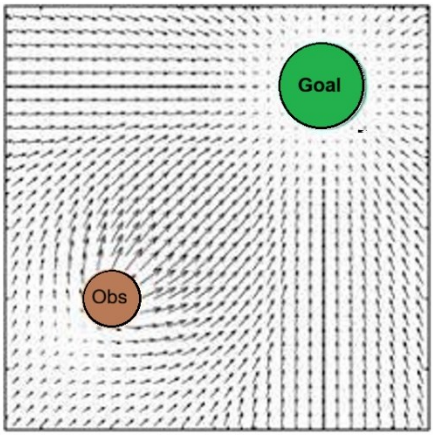


Fig. 3. Operational mechanism of the robot's distance sensor for obstacle detection and environment mapping.

The robot's movement from the starting point to the target point, comparing the time it takes to reach the target, is shown in Table 1. The first column shows the time it takes for the system to move without activating the neural network training, while the second column indicates the time the robot takes to reach the target. Here, it demonstrates the use of the neural network with the system for training the robot to reach the target in the shortest possible time.

TABLE 1. Comparative analysis of execution times required to reach the target position.

cases	Time with Training ms	Time Outside of Training ms
case01	5.12	3.70
case02	10.45	7.33
case03	11.87	12.17
case04	47.33	60.77
case05	31.44	20.34
case06	17.85	8.96
case07	16.74	12.57
case08	6.22	6.82
case09	5.80	3.69

The kinematic model of the mobile robot is described using discrete-time equations that define the evolution of the robot’s position and orientation as follows:

$$x(t+1) = x(t) + cos[\emptyset(t) + \theta(t)] + sin[\theta(t)]\, sin[\emptyset(t)] \quad (1)$$

$$y(t+1) = y(t) + sin[\emptyset(t) + \theta(t)] - sin[\theta(t)]\, cos[\emptyset(t)] \quad (2)$$

$$\emptyset(t+1) = \emptyset(t) - arc\, sin[\frac{2\, sin[\theta(t)]}{b}] \quad (3)$$

where $x(t)$ and $y(t)$ represent the Cartesian position of the robot, $\phi(t)$ denotes its orientation, $\theta(t)$ is the steering control input, and b represents the distance between the robot wheels. These equations capture the nonlinear behavior of the mobile robot system. As a result, conventional control techniques exhibit limited performance in dynamic and uncertain environments, motivating the use of intelligent control strategies such as neural networks and fuzzy logic controllers [20- 22].

4. Implementation and Results

4.1. Role of the Neural Network Controller

The neural network controller is employed to approximate system nonlinearities and uncertainties present in the kinematic and dynamic models. The neural network estimates:

- External disturbances acting on the system
- Unmodeled dynamics
- Unknown or time-varying system parameters

The output of the neural network is then used to compensate for these nonlinearities within the control law, thereby improving trajectory tracking accuracy and system robustness [23].

4.2. Role of the Fuzzy Logic Controller

The fuzzy logic controller (FLC) relies on linguistic rules to handle uncertainty in sensor measurements, particularly in environments with dynamic obstacles. Typical fuzzy rules are defined as:

IF distance is *Near* **AND** angle is *Large*
THEN steering angle is *High*

This rule-based approach enables flexible decision-making without requiring an exact mathematical model of the environment. The fuzzy controller generates the steering angle $\theta(t)$\theta(t)$\theta(t)$ required for obstacle avoidance while maintaining motion stability [22].

4.3. Hybrid Neural–Fuzzy Controller

The neural network and fuzzy logic controller are integrated within a hybrid control framework defined as:

$$\emptyset(t)f(x) = \frac{\sum_{i=1}^{n} \mu i(e,\dot{e}).yi}{\sum_{i=1}^{n} \mu i(e,\dot{e})} + \sum_{i=1}^{n} \omega j\emptyset j(X) + \varepsilon \tag{4}$$

- The fuzzy logic controller generates the primary steering decision based on environmental perception.
- The neural network compensates for nonlinearities and enhances dynamic performance.

The resulting steering command θ(t)\theta(t)θ(t) is directly applied to the robot's kinematic model to update its position and orientation, achieving:

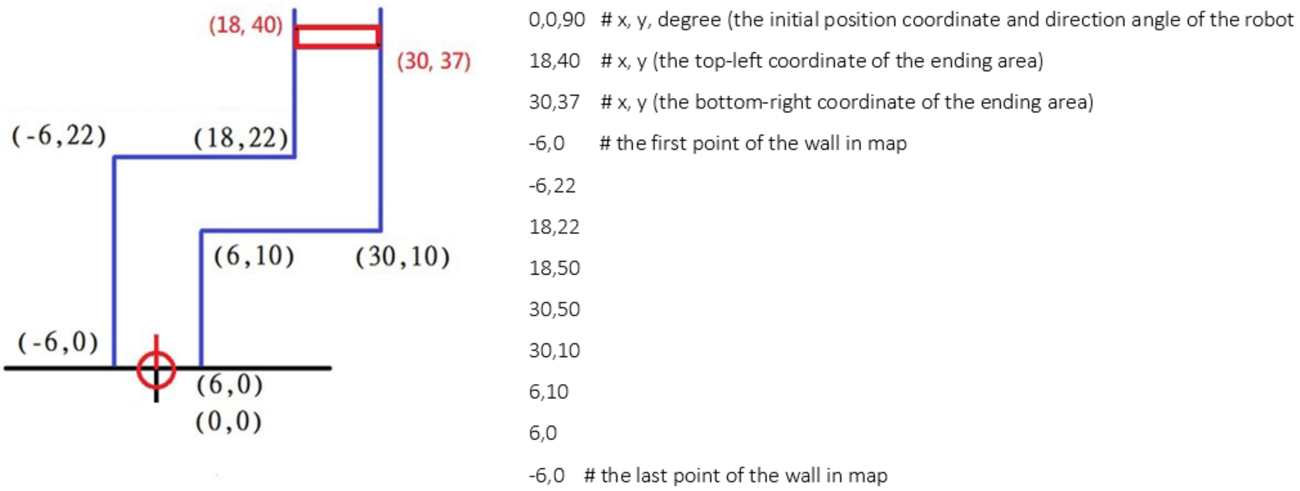
- Accurate trajectory tracking
- Effective obstacle avoidance
- Real-time adaptive response

This hybrid neural–fuzzy approach significantly improves navigation performance in complex and uncertain environments [20- 22].

4.4. Robot Movement Path

The robot movement path is defined using a map file stored in .txt format, which must be placed in the designated Data folder associated with the project files. This file-based structure enables the system to interpret various spatial dimensions and environmental constraints, as illustrated in the plotted robot trajectory diagram. The path is generated after completing the design and programming phases and is executed while the system is running in simulation mode. Once the input path file is selected, the simulation environment is updated immediately to reflect the new trajectory. All readable .txt files are stored in the Data directory located within the same folder as the executable program, ensuring efficient file access and consistency during runtime. This approach is widely adopted in robotic simulation frameworks due to its simplicity, flexibility, and compatibility with offline path planning methods [23, 24].

To handle uncertainty and imprecision in environmental representation, fuzzy set–based methods are applied. Specifically, techniques for computing fuzzy inclusion, fuzzy intersection, and fuzzy combination are employed to evaluate the robot's position relative to obstacles and path boundaries. These fuzzy operations enable robust decision-making under sensor noise and incomplete environmental information, improving navigation reliability in complex and dynamic environments [25, 26]. Figure 4 demonstrates the system structure when designing.



```
0,0,90  # x, y, degree (the initial position coordinate and direction angle of the robot
18,40   # x, y (the top-left coordinate of the ending area)
30,37   # x, y (the bottom-right coordinate of the ending area)
-6,0    # the first point of the wall in map
-6,22
18,22
18,50
30,50
30,10
6,10
6,0
-6,0   # the last point of the wall in map
```

Fig. 4. System structure when designing.

Figures 5, 6, 7, 8, and 9 illustrate the robot's environment, showing its initial path setup and navigation failures prior to implementing the proposed algorithm.

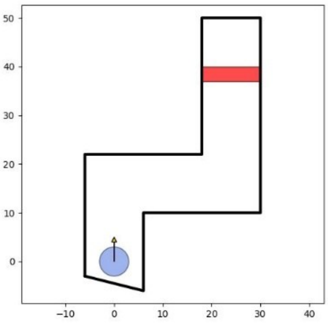


Fig. 5. The structure of the robot's path before moving and training appeared..

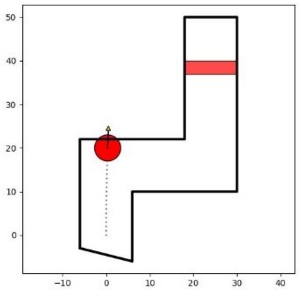


Fig. 6. The result of failure before implementing the desired algorithm result.

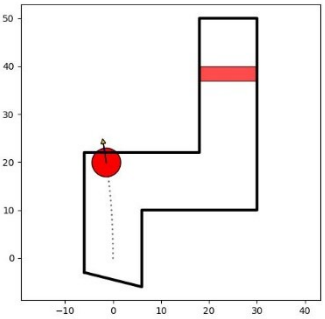


Fig. 7. The result of failure before executing the result of the desired algorithm 2.

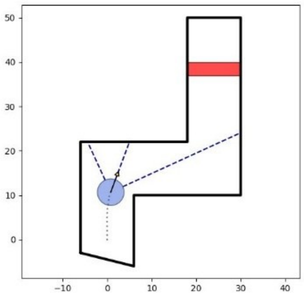


Fig. 8. The result of failure before executing the result of the desired algorithm 3.

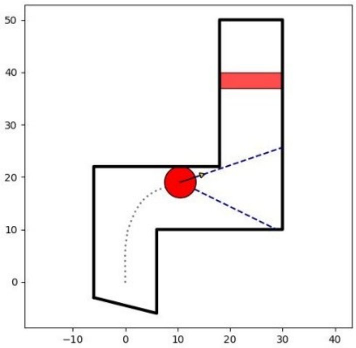


Fig. 9. The result of failure before executing the result of the desired algorithm 4.

4.5. Fuzzy Value

Fuzzy values originate from fuzzy set theory introduced by L. A. Zadeh and are used to represent uncertainty and ambiguity in data [27, 28]. Unlike classical binary logic, fuzzy logic allows elements to belong to a set with varying degrees of membership between 0 and 1. For example, a person with a height of 170 cm may belong to the set *tall* with a membership value of 0.7 and to the set *short* with a value of 0.3.

Fuzzy values are widely applied in artificial intelligence, control systems, and decision-making, where they enable flexible reasoning and human-like interpretation of imprecise information. This approach enhances model accuracy and decision robustness in environments characterized by uncertainty [26, 27]. Figure 10 illustrates a robot navigation system that uses fuzzy logic controller to calculate a steering angle α by processing the angle θ toward a target and the repulsive force generated by a nearby obstacle.

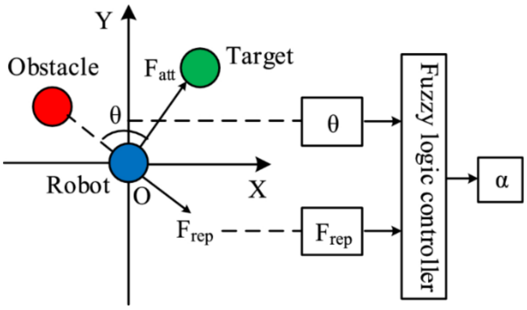


Fig. 10. Schematic of a fuzzy logic controller calculating steering angle α from target-tracking and obstacle-avoidance forces.

The following output is generated by a Python-based machine learning algorithm that is designed to optimize the robot navigation controller shown in your diagram.

```
Training start
Training complete
Best parameters:
theta: [-0.01315845]
means: [30.610657   21.09245913 17.21238213 10.75470524 33.73703398 32.44443506
 10.58453168 23.18600461 10.7037302  36.14803276 25.84163583 23.14960372]
weight: [-0.50514279 -0.15295307  1.06938944 -1.4688426 ]
SD: [4.71516805 4.95161578 4.66159846 4.37114028]
Error rate: 11.544226545111352
Training start
Training complete
Best parameters:
theta: [-0.03600505]
means: [18.49620182 37.55144581 31.71226107 10.97749001 21.78374002 28.20498068
 17.32776633 31.40822308  8.86415586 17.17385645  5.85711146 33.42838887]
weight: [-0.59521957 -0.32420868  0.8486033   0.41544254]
SD: [1.53505651 1.5643313  2.79424867 2.86379059]
Error rate: 12.398679297418267
```

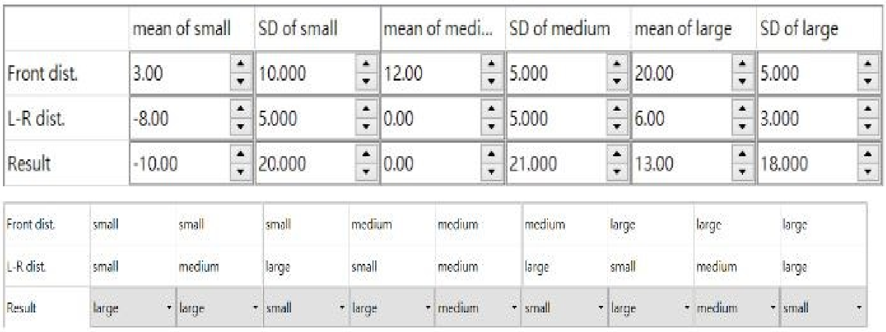
4.6. Calculation Methods and Goal Variables

The calculation methods related to the fuzzy system are only convenient and easy to implement, so the minimum value for the fuzzy intersection, the maximum value for the fuzzy union, and the field implication for the ambiguous inclusion are determined, so that the process can be reduced as in the textbook, and the method for removing the puzzle is not difficult and easy in terms of difference, The effect is good or bad before you know it, so I chose to use the separate center of gravity method to implement it. The goal variables are:

- If the distance in front of the car is small, and the difference between the distance to the left and the right is small, this means that the car is about to collide with the wall. In this case, the steering wheel should be turned to the right to avoid the collision, so it is preferable to set it to a large angle first.
- If the front distance is small, and the difference between the left and the right side is close to the average, the car may approach the wall. Here, the steering wheel can be turned either to the right or to the left, so it is preferable to start adjusting it to a large angle.
- If the distance in front of you is small, and the difference between the left and the right is large, this indicates that the car is about to collide with the wall, with a larger space on the left side compared to the right side. Therefore, the steering wheel should be turned to the left to avoid the collision as much as possible, so it is preferable to set the angle to small first.
- If the distance in front of you is medium, and the distance to the left is less than the right, this means that the front distance is medium and the left side is closer than the right side. Therefore, the steering wheel should be turned to the right to avoid colliding with the wall, so it is preferable to set the angle to large first.

4.7. Parameter part

Figure 11 shows the final results of the proposed system, through which the optimal solution for the input data was reached through several theories and hypotheses that were solved when improving the optimal solution for the input.

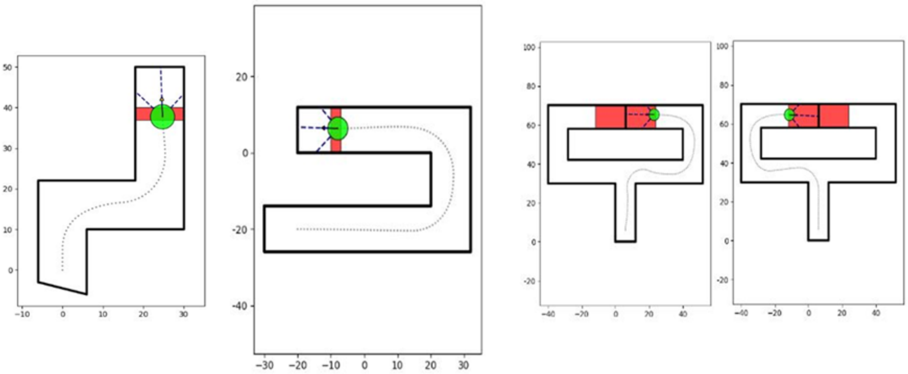


	mean of small	SD of small	mean of medi...	SD of medium	mean of large	SD of large
Front dist.	3.00	10.000	12.00	5.000	20.00	5.000
L-R dist.	-8.00	5.000	0.00	5.000	6.00	3.000
Result	-10.00	20.000	0.00	21.000	13.00	18.000

Front dist.	small	small	small	medium	medium	medium	large	large	large
L-R dist.	small	medium	large	small	medium	large	small	medium	large
Result	large	large	small	large	medium	small	large	medium	small

Fig. 11. Parameter configuration for fuzzy membership functions and decision rules.

The robot uses a fuzzy logic controller to navigate different paths based on sensor data. In case01, the robot follows a simple L-shaped corridor by staying centered between walls. Case02 shows it making a tight U-turn in a narrow space without hitting obstacles. Case 6 demonstrates decision-making where the robot chooses to turn either left or right at a junction depending on the specific fuzzy rules and parameters set in the system, as shown in Figure 12.



case01 case02 case06

Fig. 12. Robot navigation performance across different scenarios.

The distances to the left and right are equal as you approach the wall. At this point, regardless of whether you choose to turn left or right, the decision correlates with adjusting the opposite direction when the left-right (LR) blur variable is averaged, indicating that the left and right distances are equidistant. Figure 13 shows two examples illustrating the turning behavior.

Fig. 13. The robot moves according to the specified path as it is in the simulation-1.

5. Conclusion

This study introduced a structured hybrid fuzzy–neural control framework for mobile robot navigation in uncertain environments. The proposed architecture separates decision-making and nonlinear compensation into distinct layers, enhancing interpretability, modularity, and system transparency. The fuzzy controller provides the primary steering decision, while the neural network compensates for nonlinear dynamics and modeling uncertainties. The system was implemented and evaluated in a simulation environment using quantitative performance metrics. Experimental results demonstrate improved trajectory tracking accuracy, reduced error, and enhanced stability compared to standalone fuzzy and neural approaches. The proposed framework maintains computational efficiency and is suitable for real-time robotic applications. Overall, the results confirm that integrating fuzzy reasoning with neural compensation significantly enhances navigation performance without increasing system complexity. The approach is scalable and applicable to autonomous mobile robots operating in dynamic and uncertain environments.

References

[1] J. Borenstein and Y. Koren, "The vector field histogram—Fast obstacle avoidance for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278–288, 1991. doi: 10.1109/70.88137.

[2] J. Borenstein and Y. Koren, "Real-time obstacle avoidance for fast mobile robots," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 19, no. 5, pp. 1179–1187, Sep. 1989, doi: 10.1109/21.44018.

[3] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 2nd ed. London, U.K.: Springer, 2010, doi: 10.1007/978-1-84628-642-1.

[4] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006, doi: 10.1017/CBO9780511546877.

[5] J. J. E. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ, USA: Prentice Hall, 1991.

[6] M. W. Spong, S. Hutchinson, and M. Vidyasagar, *Robot Modeling and Control*. Hoboken, NJ, USA: Wiley, 2006.

[7] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson, 2009.

[8] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006, doi: 10.1007/978-0-387-45528-0.

[9] C. N. Kim and M. M. Trivedi, "A neuro-fuzzy controller for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 28, no. 6, pp. 829–840, Dec. 1998, doi: 10.1109/3477.735392.

[10] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.

[11] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, Mar. 1990, doi: 10.1109/72.80202.

[12] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[13] X. Yao, "Evolving artificial neural networks," *Proceedings of the IEEE*, vol. 87, no. 9, pp. 1423–1447, Sep. 1999, doi: 10.1109/5.784219.

[14] P. Corke, *Robotics, Vision and Control*, 2nd ed. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-54413-7.

[15] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 2nd ed. London, U.K.: Springer, 2010, doi: 10.1007/978-1-84628-642-1.

[16] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, San Francisco, CA, USA, 2000, pp. 1572–1577, doi: 10.1109/ROBOT.2000.844081.

[17] H. Zhang and S. Wang, "Fuzzy logic-based visual navigation for mobile robots," *IEEE Transactions on Fuzzy Systems*, vol. 18, no. 6, pp. 1155–1167, Dec. 2010, doi: 10.1109/TFUZZ.2010.2042282.

[18] G. Kratzig, "Simulated pistol training: The future of law enforcement training?" *International Police Training Journal*, Mar. 2022, doi: N/A.

[19] A. H. Zahraee, J. Szewczyk, J. K. Paik, and M. Guillaume, "Robotic hand-held surgical device: Evaluation of end-effector kinematics," in *Proc. Int. Conf. Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, Beijing, China, 2022, doi: 10.1007/978-3-642-15711-0_54.

[20] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*, 2nd ed. London, U.K.: Springer, 2010, doi: 10.1007/978-1-84628-642-1.

[21] F. L. Lewis, S. Jagannathan, and A. Yesildirak, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. London, U.K.: Taylor & Francis, 1999.

[22] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Transactions on Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990, doi: 10.1109/72.80202.

[23] C. N. Kim and M. M. Trivedi, "A neuro-fuzzy controller for mobile robot navigation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 28, no. 6, pp. 829–840, 1998, doi: 10.1109/3477.735392.

[24] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2006, doi: 10.1017/CBO9780511546877.

[25] P. Corke, *Robotics, Vision and Control*, 2nd ed. Cham, Switzerland: Springer, 2017, doi: 10.1007/978-3-319-54413-7.

[26] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965, doi: 10.1016/S0019-9958(65)90241-X.

[27] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*. Upper Saddle River, NJ, USA: Prentice Hall, 1999, doi: N/A.

[28] L. A. Zadeh, "Fuzzy sets," *Information and Control*, vol. 8, no. 3, pp. 338–353, 1965, doi: 10.1016/S0019-9958(65)90241-X.