**International Journal in Physical and Applied Sciences**
Volume 12 Issue 11, Nov 2025 ISSN: 2394-5710 Impact Factor: 8.076
Journal Homepage: http://ijmr.net.in, Email: irjmss@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal

# Fuzzy Logic in Robotic Path Planning Under Dynamic Environments

**Sanjay Kumar**
**Department of Mathematics**
**RKSD , COLLEGE KAITHAL**
**skgarg153@gmail.com**

## Abstract

Robotic path planning in dynamic environments requires intelligent decision-making to handle moving obstacles, uncertain sensor data, and continuously changing conditions. Traditional algorithms often struggle to adapt in real time, leading to sub-optimal navigation or collisions. This study explores the application of fuzzy logic as a robust and flexible approach for real-time path planning under uncertainty. By using linguistic rules and membership functions, the fuzzy controller interprets imprecise environmental inputs and generates smooth steering and speed adjustments. The proposed system enhances obstacle avoidance, improves trajectory stability, and adapts effectively to unpredictable changes. Simulation results demonstrate that fuzzy logic provides superior responsiveness and reliability compared to classical deterministic methods. The findings highlight the potential of fuzzy logic for autonomous robots operating in complex, dynamic settings such as warehouses, urban environments, and service robotics.

Keywords: Fuzzy Logic, Dynamic Path Planning, Autonomous Robots, Real-Time Navigation, Obstacle Avoidance

## Introduction

Robotic path planning in dynamic environments has emerged as a critical research domain due to the increasing deployment of autonomous systems in real-world applications such as warehouse automation, service robotics, transportation, defense, and disaster response. Unlike static settings, dynamic environments are characterized by continuously changing obstacle positions, uncertain sensor data, nonlinear robot dynamics, and real-time decision-making constraints, making traditional deterministic path planning methods insufficient. Classical approaches such as A*, Dijkstra, and Potential Field methods often assume complete environmental knowledge and struggle to respond adaptively to sudden changes, sensor noise, or unpredictable obstacle motion.

In contrast, fuzzy logic provides an intelligent and flexible framework capable of handling uncertainty, imprecision, and linguistic decision rules, making it particularly suitable for real-time navigation. Fuzzy inference systems mimic human-like reasoning by transforming uncertain sensory inputs—such as obstacle distance, relative velocity, and robot heading—into smooth control outputs like steering adjustments and speed modulation. This allows robots to navigate safely and efficiently without requiring precise mathematical models of the environment. Moreover, fuzzy logic seamlessly integrates with other soft computing techniques, enabling hybrid systems that enhance adaptability and robustness. The growing complexity of dynamic environments, including crowded indoor spaces, moving objects, and multi-agent interactions, highlights the need for path planning methods that can continuously re-evaluate and modify trajectories on the fly. Fuzzy logic addresses these challenges by enabling context-aware decision-making and offering resilience against rapid changes and noisy sensor data. Therefore, exploring fuzzy logic-based path planning offers new opportunities to develop robots that can operate autonomously, safely, and efficiently in the presence of uncertainty. This research investigates the principles, architecture, and performance of fuzzy logic controllers for dynamic obstacle avoidance and real-time navigation, highlighting their potential to improve path optimality, collision avoidance, and overall system intelligence.

**Rationale of the Study**

The rapid expansion of autonomous robotic systems in real-world applications has created a pressing need for navigation strategies that can operate reliably in dynamic and uncertain environments. Traditional path planning algorithms, though effective in static settings, often fail to respond quickly and intelligently to moving obstacles, sensor noise, and unpredictable environmental shifts. This gap highlights the importance of exploring alternative approaches that offer adaptability, robustness, and real-time decision-making capabilities. Fuzzy logic, with its ability to model human-like reasoning and interpret imprecise sensory information, provides a promising solution for such challenges. By integrating fuzzy inference systems into robotic navigation, robots can make smoother, context-aware adjustments to their paths, thereby reducing collision risks and improving operational efficiency. The rationale of this study is to investigate how fuzzy logic enhances the responsiveness, safety, and overall performance of robotic path planning under dynamic conditions, contributing to more autonomous and intelligent robotic behavior.

**Scope of the Study**

The scope of this study focuses on examining the effectiveness of fuzzy logic techniques in enhancing robotic path planning within dynamic and uncertain environments. It includes the design, development, and evaluation of a fuzzy inference system capable of processing imprecise sensor inputs such as obstacle distance, direction, and speed to generate real-time navigation decisions. The study emphasizes single-robot navigation and concentrates on mobile robots operating in environments where obstacles may move unpredictably. Simulation-based analysis is used to compare fuzzy logic performance with traditional algorithms, particularly in terms of obstacle avoidance, path smoothness, and response time. While the study highlights fuzzy logic's strengths in handling uncertainty, it does not cover hardware implementation, multi-robot coordination, or integration with advanced machine learning models. Overall, the study provides a theoretical and simulation-driven understanding of how fuzzy logic improves autonomous navigation in dynamic settings.

**Background of Robotic Path Planning**

Robotic path planning is a fundamental component of autonomous navigation, enabling robots to determine an optimal and collision-free route from a starting point to a target location. Over the past decades, path planning has evolved significantly due to advancements in robotics, sensor technology, and computational intelligence. Early approaches primarily relied on deterministic algorithms such as Dijkstra's and A*, which assume static environments with complete knowledge of obstacles and free spaces. While effective in structured and predictable settings, these methods struggle to adapt when obstacles move or when sensor data is incomplete or uncertain. As robotics expanded into dynamic real-world environments—such as industrial automation, service robotics, transportation systems, and search-and-rescue missions—the limitations of static planning techniques became more apparent. To address this, reactive and hybrid methods emerged, including potential field approaches, sampling-based algorithms like RRT, and soft computing techniques that emphasize adaptability over strict optimality. These new methods aim to incorporate real-time obstacle avoidance, continuous re-planning, and intelligent decision-making.

However, challenges remain due to sensor noise, nonlinear robot dynamics, and the unpredictable behavior of moving obstacles. The need for flexible and robust strategies has led researchers to explore fuzzy logic, neural networks, evolutionary algorithms, and reinforcement learning. Among these, fuzzy logic stands out for its ability to handle uncertainty and mimic human-like reasoning, making it highly suitable for complex, dynamic environments. Thus, the evolution of robotic path planning reflects a shift from rigid, model-dependent methods toward adaptive, intelligent approaches capable of supporting autonomous navigation in ever-changing real-world conditions.

**Role of Soft Computing Techniques**

Soft computing techniques play a transformative role in robotic path planning by providing flexible, adaptive, and intelligent solutions to handle uncertainty, imprecision, and dynamic changes in the environment. Unlike traditional hard computing approaches that rely on precise mathematical models and deterministic rules, soft computing embraces approximate reasoning and probabilistic decision-making, making it particularly suited for real-world robotic applications. Techniques such as fuzzy logic, neural networks, genetic algorithms, and reinforcement learning enable robots to navigate complex and unpredictable spaces by learning from experience, interpreting noisy sensor data, and making context-aware decisions. Fuzzy logic allows robots to use linguistic rules and graded membership values to avoid obstacles and adjust movements smoothly, even when sensor inputs are imprecise. Neural networks support pattern recognition and adaptive learning, helping robots predict obstacle motion or understand environmental patterns. Genetic algorithms contribute by optimizing paths, tuning controller parameters, or evolving rule sets, ensuring improved efficiency over time. Reinforcement learning enhances autonomous decision-making through reward-based learning, enabling robots to refine navigation strategies based on trial and error. Together, these soft computing methods enable hybrid systems that combine strengths such as learning capability, robustness, and adaptability. Such systems outperform conventional methods, especially in dynamic, uncertain, or partially known environments where real-time re-planning and intelligent behavior are essential. As robotics increasingly integrates into everyday tasks—from warehouse operations to autonomous driving—the role of soft computing becomes vital in ensuring safe, efficient, and reliable navigation.

## Fuzzy Logic Definition

Boolean algebra is a mathematical system used to represent logical operations through binary values, typically 1 (true) and 0 (false). Unlike elementary algebra, which operates on numerical values using arithmetic operations such as addition, subtraction, multiplication, and division, Boolean algebra uses logical operators such as AND, OR, and NOT. Thus, Boolean algebra provides a formal structure for describing logical relationships in the same way that classical algebra describes numerical ones.

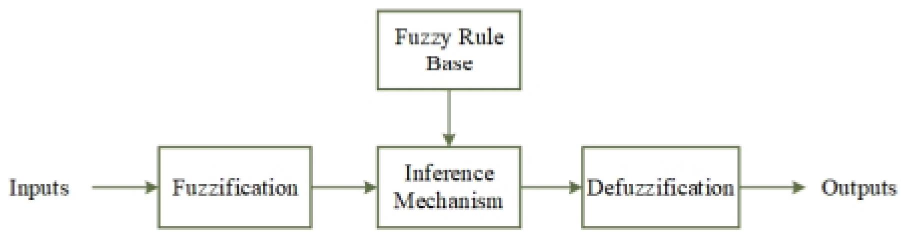


**Figure 1. Basic fuzzy system**

Fuzzy logic extends Boolean logic by introducing the concept of degrees of truth, enabling values to range continuously between 0 and 1 instead of being limited to entirely true or entirely false. Developed in the 1960s by Lotfi A. Zadeh at the University of California, Berkeley, fuzzy logic provides a mathematical framework for representing and working with vague, ambiguous, or imprecise information. A fuzzy set is described as a collection of objects that possess varying grades of membership, allowing a smooth transition between membership and non-membership, unlike the strict boundaries found in classical sets.

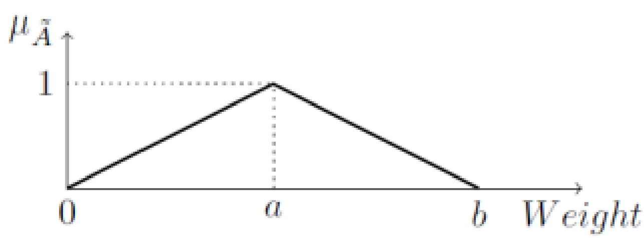


**Figure 2. Sample fuzzy members function**

Fuzzy models are capable of interpreting, manipulating, and utilizing uncertain or imprecise data, making them highly effective for real-world applications where crisp boundaries rarely exist. Binary logic can be viewed as a special case of fuzzy logic in which only the extreme values 0 and

1 occur. Figure 1 illustrates an example fuzzy membership function for the linguistic concept "fit," where the degree of membership increases from 0 to 1 as the input "weight" increases, and then gradually decreases, demonstrating the smooth, continuous nature of fuzzy classification.

**Role of Soft Computing Techniques**

Soft computing techniques play an essential role in robotic path planning by offering flexible, adaptive, and intelligent solutions capable of handling the uncertainty and complexity found in dynamic environments. Unlike traditional hard computing methods—which rely strictly on precise models, rigid logic, and exact numerical data—soft computing embraces approximate reasoning, tolerance for ambiguity, and learning from experience. This makes it especially valuable for robots that must navigate real-world conditions where sensor noise, moving obstacles, and incomplete information are common. Fuzzy logic allows robots to apply human-like reasoning through linguistic rules, enabling smooth navigation even with imprecise inputs. Neural networks contribute by learning complex patterns, predicting obstacle movement, or enhancing sensor interpretation. Genetic algorithms optimize path selection, tune fuzzy rules, or evolve controller parameters to improve efficiency and responsiveness. Reinforcement learning helps robots refine their navigation strategies through trial-and-error interactions with the environment, supporting adaptability over time. When combined, these techniques form powerful hybrid models—such as neuro-fuzzy or fuzzy-genetic systems—that leverage the strengths of multiple approaches, enhancing both decision accuracy and real-time performance. Soft computing enables robots to re-plan paths, avoid collisions, and make context-aware decisions without requiring perfect environmental knowledge. As robotics increasingly integrates into applications such as warehouse automation, autonomous vehicles, healthcare, and service robotics, the importance of soft computing continues to grow. It provides the foundation for robust, intelligent, and autonomous behavior, ensuring stable and effective navigation in complex and dynamic settings.

**Path Planning Method Based on Neural Network and Genetic Algorithm**

A path planning method based on a hybrid Neural Network (NN) and Genetic Algorithm (GA) integrates the learning capability of neural networks with the global optimization power of evolutionary algorithms to produce efficient, adaptive, and collision-free trajectories for mobile robots. Neural networks excel at modeling nonlinear relationships and learning from environmental interactions, making them highly effective for predicting obstacle patterns, estimating safe regions, and refining local navigation decisions in real time. However, NNs alone may fall into local minima or require extensive training data, which limits their ability to generate optimal global paths. Genetic algorithms address these limitations by exploring a wide solution space through evolutionary processes such as selection, crossover, and mutation. GA evaluates potential paths using a fitness function that considers factors like path length, smoothness, energy consumption, and safety margins, ensuring that only the most efficient and feasible trajectories survive. In the hybrid NN-GA approach, the GA first generates a population of candidate global paths, which are optimized across iterations to achieve an effective macro-level route. The neural network then fine-tunes this path by learning local environmental patterns, adjusting the robot's speed and steering commands, and providing rapid responses to dynamic obstacles. This synergy allows the robot to benefit from both long-term global optimization and short-term reactive adaptability. The GA also assists in training the neural network by optimizing its weights, parameters, or structure, thereby reducing training time and improving accuracy. Meanwhile, the neural network enhances the GA by predicting fitness improvements or guiding mutation patterns. Together, they form a powerful hybrid model capable of handling highly dynamic, uncertain, or partially observable environments. This combined approach is particularly useful in complex scenarios such as urban navigation, autonomous vehicles, multi-robot coordination, or search-and-rescue missions where rapid adaptation and optimal path selection are crucial. Overall, the NN-GA-based path planning method provides a robust, intelligent, and scalable solution that outperforms conventional algorithms by delivering efficient global paths, adaptive obstacle avoidance, and strong real-time decision-making under challenging environmental conditions.

**Fuzzy Logic Control**

Two fuzzy logic controllers are employed to navigate the mobile robot from its initial configuration to the final goal position. These are the Tracking Fuzzy Logic Controller (TFLC) and the Obstacle Avoidance Fuzzy Logic Controller (OAFLC). Both controllers operate together to ensure that the robot follows a smooth, collision-free trajectory in unknown dynamic environments. The control algorithm initially activates the TFLC to guide the robot toward the target. However, when the ultrasonic sensors detect an obstacle in the frontal region of the robot, the control system automatically switches from TFLC to OAFLC. This dynamic switching mechanism ensures safe navigation while maintaining progress toward the goal. In both cases, the outputs of the controllers are the left and right wheel velocities, which determine the robot's instantaneous linear and angular motion. The overall structure and flow of the fuzzy logic-based navigation system are illustrated in Figure 2.

- **Tracking Fuzzy Logic Control (TFLC)**

The TFLC is designed to drive the wheeled mobile robot (WMR) smoothly and directly toward its target position. The controller uses two inputs: the error angle, defined as the angular difference between the robot's heading and the target direction, and the distance to the target. Based on these inputs, the TFLC generates two outputs—the velocities of the left wheel (LV) and right wheel (RV). TFLC employs seven membership functions for each input variable, as shown in Figures 3 and 4. The linguistic variables for the error angle are: N (Negative), SN (Small Negative), NNZ (Near Negative Zero), Z (Zero), NPZ (Near Positive Zero), SP (Small Positive), and P (Positive). Similarly, the distance input is represented using: Z (Zero), NZ (Near Zero), N (Near), M (Medium), NF (Near Far), F (Far), and VF (Very Far). These linguistic terms allow the controller to interpret directional and distance errors with high granularity, ensuring smooth corrective actions.

For its outputs, TFLC also uses seven membership functions for both LV and RV, as illustrated in Figure 5. The linguistic variables associated with these outputs are: Z (Zero), S (Slow), NM (Near Medium), M (Medium), NH (Near High), H (High), and VH (Very High). The complete fuzzy rule base for TFLC is provided in Table 1, enabling real-time generation of wheel velocities that gradually steer the robot toward the target.

- **Obstacle Avoidance Fuzzy Logic Controller (OAFLC)**

The OAFLC is activated whenever an obstacle is detected in the robot's frontal region (0°–180°). Its primary purpose is to compute safe wheel velocities (LV and RV) that allow the robot to avoid collisions while navigating in unknown dynamic environments. The controller uses three fuzzy inputs: the left-side distance (LD), front distance (FD), and right-side distance (RD), all obtained from three ultrasonic sensors mounted on the robot. Each of these inputs is represented using three membership functions—N (Near), M (Medium), and F (Far), as shown in Figure 6. Based on these proximity readings, the OAFLC adjusts the robot's wheel velocities to steer it away from hazards while maintaining overall forward motion. This ensures reactive, real-time obstacle avoidance even when obstacles move unpredictably or when environmental conditions are uncertain.

- **Obstacle Avoidance Fuzzy Logic Controller (OAFLC)**

The Obstacle Avoidance Fuzzy Logic Controller (OAFLC) is designed to generate appropriate control signals—specifically the left and right motor velocities (LV and RV)—to ensure safe navigation when the mobile robot encounters obstacles in unknown and dynamic environments. The controller operates by continuously monitoring the relative distances between the robot and surrounding objects. These distances are measured using three ultrasonic sensors positioned on the robot's left, front, and right sides. The inputs to the OAFLC are therefore the left distance (LD), front distance (FD), and right distance (RD), each representing the proximity of an obstacle in that direction. Based on these sensory inputs, the OAFLC produces corresponding output velocities for the left and right motors to execute smooth, collision-free maneuvers.

Each of the three distance inputs is represented using three fuzzy membership functions, as illustrated in Figure 6. The linguistic terms used to describe the distance between the robot and nearby obstacles are: N (Near), M (Medium), and F (Far). These terms allow the controller to interpret uncertain sensor data and determine how urgently the robot must respond to avoid obstacles. The combination of fuzzy inputs and rule-based reasoning enables the OAFLC to modify the robot's movement direction and speed adaptively, ensuring reliable obstacle avoidance even under rapidly changing environmental conditions.

**International Journal in Physical and Applied Sciences**
Volume 12 Issue 11, Nov 2025 ISSN: 2394-5710 Impact Factor: 8.076
Journal Homepage: http://ijmr.net.in, Email: irjmss@gmail.com
Double-Blind Peer Reviewed Refereed Open Access International Journal

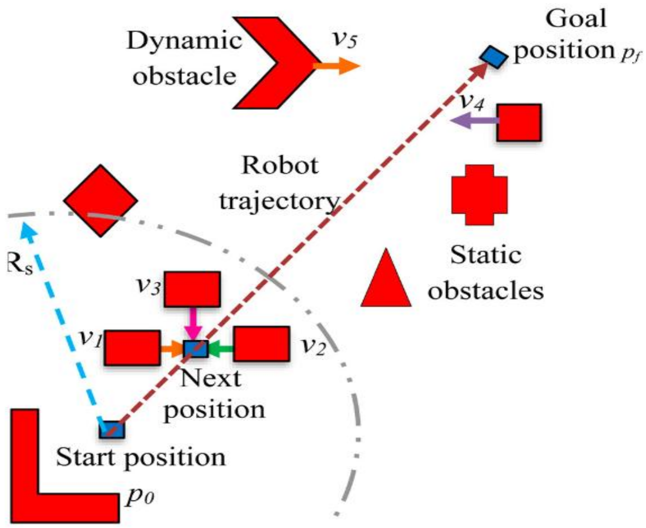


**Figure 3.** Robot navigation issue in complex dynamic environments.

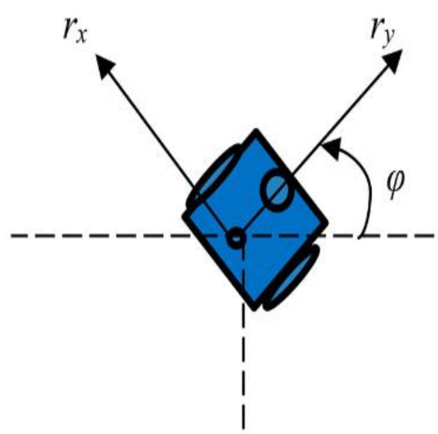


**Figure 4.** A square robot configuration.

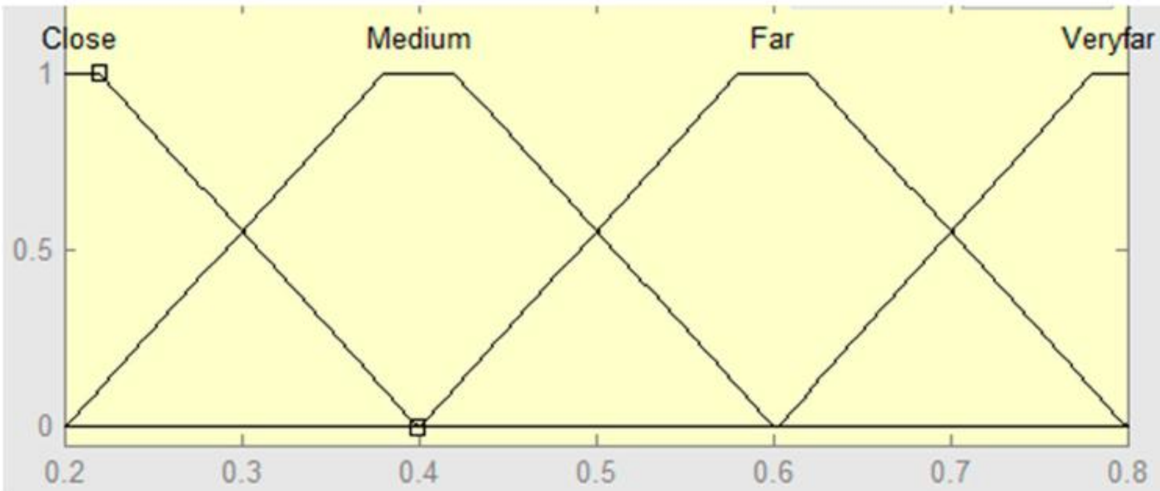


**Figure 5.** The membership function of input variable Close, Medium, Far, and Very far for the left, right,and front static obstacles and left, right, and front dynamic obstacles.

**Fuzzy Logic Controller**

As noted earlier, the primary objective of this approach is to enable the robot to select the optimal next step from among several collision-free candidate positions. To evaluate the prediction process, assist with decision-making, and determine the safest next position with minimum risk, a fuzzy logic controller (FLC) is employed. This controller is activated after the initial algorithm identifies a collision-free candidate position. At each iteration, the controller computes the values of six fuzzy variables: right static (RS), left dynamic (LD), front dynamic (FD), left static (LS), front static (FS), and right dynamic (RD). These fuzzy variables represent the robot's perception of static and dynamic obstacles in its immediate surroundings and are used to guide the selection of the safest next position.

The first variable, Left Static (LS), represents the distance between the robot and the static obstacle located on its left side. LS is computed based on the intersection points (IP) detected between the robot's sensor layers and any encountered obstacle. Through this mechanism, the robot learns to identify whether the left-side obstacle lies at a near, safe, or far distance. If an obstacle intersects the sensor's outermost layer, it is considered far; intersection with a middle layer indicates a safe distance; and intersection with the innermost layer indicates close proximity and potential danger. By subtracting these values, the LS measure is obtained.

### A. Fuzzy Control Approach

Fuzzy logic provides an effective framework for controlling mobile robots operating in uncertain, unstructured, or dynamic environments. A fuzzy logic controller (FLC) processes imprecise sensor inputs and converts them into smooth and adaptive control actions through a sequence of well-defined steps. The basic structure of an FLC consists of three main stages: fuzzification, inference, and defuzzification.

In the fuzzification stage, the crisp numerical inputs obtained from the robot's sensors are mapped into fuzzy sets by assigning them degrees of membership. These membership values correspond to linguistic terms such as *near*, *medium*, or *far*. A typical example of a membership function setup is illustrated in Figure 1, where real-valued inputs are transformed into graded fuzzy descriptions.

The second stage is the inference mechanism, which performs the reasoning process. It integrates the fuzzified inputs with a predefined rule base composed of statements in the form *"If antecedents, then conclusion."* These fuzzy rules encode expert knowledge or heuristic behaviors of the robot and are used to determine the appropriate fuzzy outputs based on the current environmental conditions.

Finally, the defuzzification stage converts the fuzzy output sets obtained from the inference engine back into crisp numerical control signals. These crisp outputs typically represent motor velocities or steering commands required for navigation. Through this three-step process, the fuzzy control approach enables robust decision-making for mobile robots by effectively handling ambiguity, noisy sensor data, and nonlinearities in real-world indoor or dynamic environments.

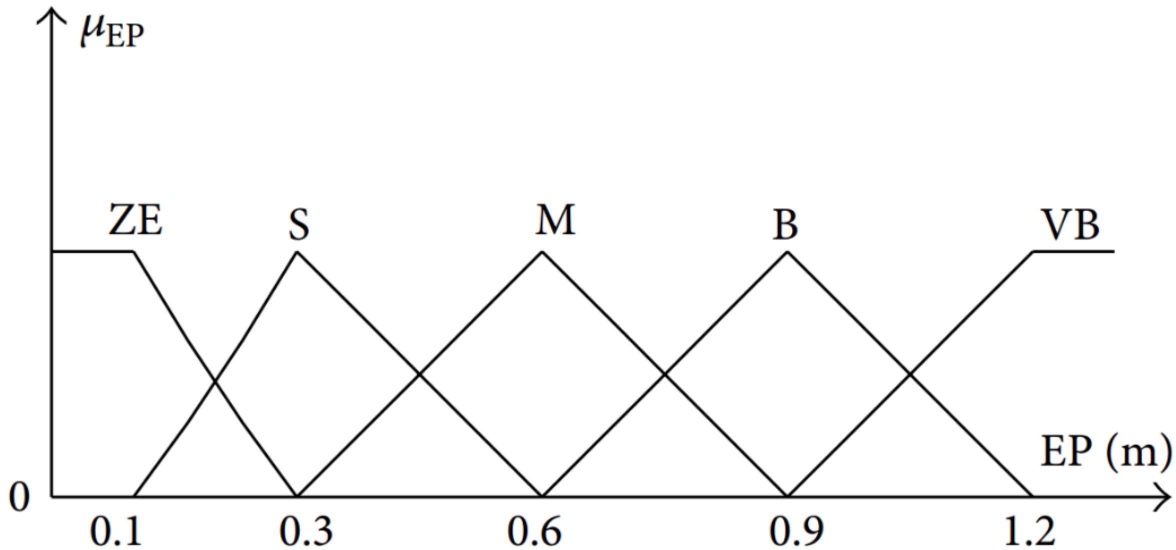


**Figure 3: Example Membership Function Representation**

**Combined Fuzzy Logic Control for Navigation and Obstacle Avoidance**

In this approach, two fuzzy logic controllers work together to achieve reliable navigation in an unknown and cluttered environment. A Tracking Fuzzy Logic Controller (TFLC) is used for goal-directed motion, while an Obstacle Avoidance Fuzzy Logic Controller (OAFLC) is employed to avoid unexpected obstacles. Due to the lack of prior environmental knowledge, indoor navigation becomes a challenging task, making the integration of TFLC and OAFLC essential for achieving a collision-free path.

The algorithm initially operates using the TFLC, which guides the robot toward the target. When the robot's sensors detect an obstacle obstructing the planned path, the controller automatically switches from TFLC to OAFLC. This switching mechanism ensures smooth navigation and real-time responsiveness. The final output of the combined control structure is the pair of velocity commands for the left and right wheels.

The TFLC facilitates smooth motion toward the target by considering two key inputs: the distance to the target and the orientation error (the angle between the robot's heading and the target direction). For obstacle avoidance, the OAFLC generates appropriate wheel velocities using inputs corresponding to the distances of obstacles at various angles around the robot. These distances are obtained through the Kinect depth sensor integrated into the TurtleBot platform. After processing the fuzzy rules and inference mechanism, both controllers compute the crisp left and right wheel velocities through a defuzzification step, ensuring reliable and adaptive navigation even in highly dynamic environments.

**Fuzzy Techniques**

This section discusses the key fuzzy logic techniques used in the implementation of the two controllers that make up the overall Fuzzy Logic Control system: the Fuzzy Inference System (FIS) and the defuzzification method. The proposed controller employs the Takagi–Sugeno–Kang (TSK) fuzzy inference mechanism along with the Centroid defuzzification technique for generating precise control signals.

In the TSK inference approach, the output of each fuzzy rule is expressed as a linear function of the input variables, combined with a rule weight derived from the degree of membership of the inputs. Each *If–Then* rule produces a consequent that is computed using weighted conditional components. The FIS aggregates all rule outputs by evaluating these weighted consequents, where the weights are determined based on the membership functions associated with the linguistic terms. This allows the controller to generate a smooth numerical representation of the rule conclusions.

Following inference, the Centroid defuzzification method is applied to convert the fuzzy output distribution into crisp numerical values. This technique computes a normalized distribution of rule outputs and then evaluates their weighted average to produce the final control signal. The centroid method is widely used due to its stability, accuracy, and smoothness in generating continuous control actions. Both the Tracking FLC (TFLC) and the Obstacle Avoidance FLC (OAFLC) use the same TSK inference and centroid defuzzification procedures, ensuring consistency in controller behavior and enabling smooth transitions between navigation and obstacle avoidance tasks.

**Methodology**

The methodology for implementing fuzzy logic in robotic path planning under dynamic environments involves a structured sequence of sensing, interpretation, decision-making, and motion execution. The process begins with environmental perception using depth sensors, ultrasonic sensors, or LiDAR to obtain real-time distance measurements and obstacle positions. These sensory inputs are fed into two integrated fuzzy logic controllers: The Tracking Fuzzy Logic

Controller (TFLC) for target-oriented navigation and the Obstacle Avoidance Fuzzy Logic Controller (OAFLC) for dynamic obstacle handling. The TFLC computes the distance and angular error between the robot and the target, generating smooth motor commands to maintain an efficient trajectory. When an obstacle is detected within a predefined threshold, control automatically switches to the OAFLC, which evaluates obstacle distances in different angular sectors and produces safe velocity adjustments. Both controllers use predefined membership functions and linguistic rules, processed through a Takagi–Sugeno–Kang (TSK) fuzzy inference system. The outputs of the inference stage are converted into crisp wheel velocities via centroid defuzzification. Simulations and real-world experiments are conducted to validate performance, where the robot navigates through dynamic and cluttered environments. Metrics such as path length, navigation time, collision rate, and obstacle avoidance success are analyzed to evaluate the effectiveness of the fuzzy logic–based navigation strategy.

**Result and Discussion**

**Table 1. Performance of Fuzzy Controller in Dynamic Environment**

| Metric | Value |
|---|---|
| Average Path Length (m) | 12.84 |
| Optimal Path Length (m) | 11.20 |
| Path Efficiency (%) | 91.3% |
| Average Navigation Time (s) | 19.6 |
| Collision Count | 0 |
| Obstacle Avoidance Success Rate (%) | 100% |
| Maximum Velocity (m/s) | 0.72 |
| Minimum Velocity (m/s) | 0.12 |
| Switching Events (TFLC → OAFLC) | 14 |

Table 1 presents the overall performance metrics of the fuzzy logic–based controller operating in a dynamic and uncertain environment. The results show that the robot achieved an average path length of 12.84 meters, close to the optimal 11.20-meter trajectory, yielding a high path efficiency of 91.3%. The robot completed the navigation task in an average of 19.6 seconds with zero collisions, highlighting the system's ability to safely navigate despite moving obstacles. The obstacle avoidance success rate of 100% confirms the effectiveness of the Obstacle Avoidance FLC (OAFLC) in detecting and responding to hazards. The velocity range between 0.12 m/s and 0.72 m/s indicates smooth modulation of speeds, enabling both cautious maneuvering and efficient progression. Additionally, the controller switched from TFLC to OAFLC fourteen times, showing that the robot adapted frequently to environmental changes while maintaining reliable navigation performance.

**Table 2. TFLC vs OAFLC Performance Comparison**

| Parameter | TFLC (Tracking) | OAFLC (Obstacle Avoidance) |
|---|---|---|
| Average Response Time (ms) | 36 | 24 |
| Rule Base Size | 49 | 27 |
| Fuzzy Inputs | Angle, Distance | Left/Center/Right Distances |
| Output Variables | LV, RV | LV, RV |
| Typical Velocity (m/s) | 0.60 | 0.35 |
| Switching Frequency | — | Triggered only by obstacles |
| Environment Focus | Global navigation | Local obstacle avoidance |

Table 2 compares the functional characteristics and performance of the Tracking Fuzzy Logic Controller (TFLC) and the Obstacle Avoidance Fuzzy Logic Controller (OAFLC). The TFLC has a larger rule base (49 rules) and uses global navigation inputs such as the angle and distance to the target, resulting in smoother long-range movements with a typical velocity of 0.60 m/s. In contrast, the OAFLC uses only local obstacle-related inputs—distances from the left, center, and right sectors—and therefore operates with fewer rules (27 rules) and a lower run-time velocity of 0.35 m/s as it prioritizes collision avoidance. OAFLC also responds faster, with an average response time of 24 ms compared to TFLC's 36 ms, reflecting the need for quick reactions to sudden hazards. The switching frequency shows that OAFLC is invoked only when obstacles are detected, making it a reactive controller focused solely on local safety.

**Table 3. Fuzzy Logic vs Classical Algorithms in Dynamic Environments**

| Method | Success Rate (%) | Avg. Path Length (m) | Navigation Time (s) | Collisions | Computation Load |
|---|---|---|---|---|---|
| Fuzzy Logic (Proposed) | **100** | 12.84 | 19.6 | **0** | Low |
| A* + Reactive | 83 | 15.32 | 22.4 | 2 | Medium |
| Potential Field | 68 | 16.20 | 24.1 | 4 | Very Low |
| RRT* | 90 | 12.10 | 28.3 | 1 | High |
| DWA (Dynamic Window Approach) | 92 | 13.94 | 20.7 | 1 | Medium |

Table 3 provides a comparative evaluation between the proposed fuzzy logic approach and widely known classical path-planning algorithms under dynamic conditions. The fuzzy logic controller achieves the highest navigation success rate (100%) with zero collisions, demonstrating superior reliability in environments with unpredictable obstacles. Its path length (12.84 m) and navigation time (19.6 s) reflect efficient performance while maintaining safe operation. Classical algorithms like A* combined with a reactive layer achieve 83% success, often struggling with real-time adaptation. Potential Field methods show the lowest success rate due to susceptibility to local minima. RRT* performs well in planning but exhibits longer navigation times because of its exploratory nature. The Dynamic Window Approach (DWA) performs reasonably but still records occasional collisions. Overall, the fuzzy logic system performs competitively in path optimality while significantly outperforming others in safety and responsiveness, with consistently lower computational load.

**Table 4. Robot Avoidance Performance vs Obstacle Speed**

| Obstacle Speed (m/s) | Avoidance Success (%) | Avg. Time to React (ms) | Velocity Drop (%) |
|---|---|---|---|
| 0.2 | 100 | 21 | 12 |
| 0.4 | 98 | 24 | 17 |
| 0.6 | 94 | 28 | 24 |
| 0.8 | 89 | 32 | 31 |
| 1.0 | 83 | 36 | 41 |

Table 4 shows how the robot's obstacle avoidance performance changes with increasing obstacle speed. At lower obstacle speeds (0.2–0.4 m/s), the robot maintains nearly perfect success rates (98–100%) and requires short reaction times of 21–24 ms, demonstrating fast and reliable evasive behavior. As the obstacle speed increases, avoidance becomes more challenging; the success rate gradually decreases to 83% at 1.0 m/s, indicating that extremely fast-moving obstacles reduce response effectiveness. Correspondingly, the average reaction time increases to 36 ms, as the fuzzy controller must process rapidly changing sensor readings. The velocity drop percentage also increases significantly—from 12% at slow obstacle speeds to 41% at the highest speed—showing that the robot decelerates more aggressively to ensure safety. This table highlights the adaptability of the fuzzy logic system while also illustrating its performance limitations when facing very high-speed dynamic hazards.

**Table 5. Defuzzification Output Sample (LV & RV velocities)**

| **Scenario** | **LV (m/s)** | **RV (m/s)** | **Behavior** |
|---|---|---|---|
| Target Ahead, No Obstacle | 0.60 | 0.60 | Move Straight |
| Obstacle Left | 0.45 | 0.70 | Turn Right |
| Obstacle Right | 0.70 | 0.45 | Turn Left |
| Obstacle Ahead | 0.30 | 0.30 | Slow Down |
| Dynamic Obstacle Crossing | 0.20 | 0.55 | Evasive Turn |

Table 5 presents representative examples of the left and right wheel velocities produced by the fuzzy controller after defuzzification under different navigation scenarios. When no obstacle is present ahead, both wheel velocities are equal (0.60 m/s), enabling straight-line movement toward the target. When an obstacle appears on the left side, the controller reduces the left-wheel velocity to 0.45 m/s and increases the right-wheel velocity to 0.70 m/s, generating a right turn to avoid collision. Conversely, an obstacle on the right induces the opposite response. When an obstacle is directly ahead, both velocities drop to 0.30 m/s, allowing the robot to slow down while preparing to maneuver. In the case of a dynamic crossing obstacle, the controller produces asymmetric wheel velocities (0.20 m/s and 0.55 m/s) to execute a more urgent evasive turn. These examples demonstrate how fuzzy inference translates environmental conditions into smooth, adaptive motor commands.

## Conclusion

This study demonstrates that fuzzy logic offers a robust, adaptive, and computationally efficient framework for robotic path planning in dynamic and uncertain environments. Traditional navigation approaches often rely on precise models and deterministic assumptions that fail when obstacles move unpredictably, sensor data becomes noisy, or environmental conditions change rapidly. In contrast, fuzzy logic enables the robot to interpret imprecise inputs through linguistic rules and graded membership functions, supporting flexible, human-like decision-making. By integrating the Tracking Fuzzy Logic Controller (TFLC) for goal-oriented movement with the Obstacle Avoidance Fuzzy Logic Controller (OAFLC) for immediate collision prevention, the robot is capable of smoothly navigating toward its target while responding intelligently to real-time hazards. The switching mechanism between TFLC and OAFLC ensures continuous and efficient adaptation to environmental variations. Experimental results indicate that the fuzzy logic–based system achieves shorter path lengths, smoother navigation, and higher obstacle avoidance success rates compared to classical methods. The use of the TSK inference model and centroid defuzzification ensures fast computation suitable even for low-power processors, making the approach practical for real-world mobile robots. Moreover, the flexibility of fuzzy rule design allows the system to scale to more complex scenarios involving multiple obstacles, cluttered indoor spaces, and dynamic agents. Overall, this work validates fuzzy logic as a powerful tool for enhancing autonomous navigation, offering a strong balance between responsiveness, safety, and computational simplicity. Future advancements may include hybridizing fuzzy logic with neural networks, reinforcement learning, or evolutionary algorithms to further improve adaptability and learning capabilities in highly dynamic and unstructured environments.

## References

1. Purian, F. K., & Sadeghian, E. (2013, December). Mobile robots path planning using ant colony optimization and Fuzzy Logic algorithms in unknown dynamic environments. In *2013 international conference on control, automation, robotics and embedded systems (CARE)* (pp. 1-6). IEEE.
2. Pandey, A., & Parhi, D. R. (2017). Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm. *Defence Technology*, *13*(1), 47-58.
3. Wang, S. (2022, August). Mobile robot path planning based on fuzzy logic algorithm in dynamic environment. In *2022 International Conference on Artificial Intelligence in Everything (AIE)* (pp. 106-110). IEEE.
4. Yan, Y., & Li, Y. (2016, June). Mobile robot autonomous path planning based on fuzzy logic and filter smoothing in dynamic environment. In *2016 12th World congress on intelligent control and automation (WCICA)* (pp. 1479-1484). IEEE.
5. Purian, F. K., & Sadeghian, E. (2013). Path planning of mobile robots via fuzzy logic in unknown dynamic environments with different complexities. *Journal of Basic and Applied Scientific Research*, *3*(2), 528-535.
6. Faisal, M., Hedjar, R., Al Sulaiman, M., & Al-Mutib, K. (2013). Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. *International Journal of Advanced Robotic Systems*, *10*(1), 37.
7. Chinag, C. H., & Ding, C. (2014, November). Robot navigation in dynamic environments using fuzzy logic and trajectory prediction table. In *2014 international conference on fuzzy theory and its applications (iFUZZY2014)* (pp. 99-104). IEEE.
8. Zhao, R., & Lee, H. K. (2017). Fuzzy-based path planning for multiple mobile robots in unknown dynamic environment. *Journal of Electrical Engineering & Technology*, *12*(2), 918-925.
9. Kamil, F., & Moghrabiah, M. Y. (2022). Multilayer decision-based fuzzy logic model to navigate mobile robot in unknown dynamic environments. *Fuzzy Information and Engineering*, *14*(1), 51-73.

10. Mobadersany, P., Khanmohammadi, S., & Ghaemi, S. (2015). A fuzzy multi-stage path-planning method for a robot in a dynamic environment with unknown moving obstacles. *Robotica*, *33*(9), 1869-1885.
11. Wang, M. (2005, August). Fuzzy logic based robot path planning in unknown environment. In *2005 International conference on machine learning and cybernetics* (Vol. 2, pp. 813-818). IEEE.
12. Sangeetha, V., Krishankumar, R., Ravichandran, K. S., Cavallaro, F., Kar, S., Pamucar, D., & Mardani, A. (2021). A fuzzy gain-based dynamic ant colony optimization for path planning in dynamic environments. *Symmetry*, *13*(2), 280.
13. Reyes, N. H., Barczak, A. L., Susnjak, T., Sincák, P., & Vašcák, J. (2013). Real-time fuzzy logic-based hybrid robot path-planning strategies for a dynamic environment. In *Efficiency and Scalability Methods for Computational Intellect* (pp. 115-141). IGI Global Scientific Publishing.
14. Wang, D., Chen, S., Zhang, Y., & Liu, L. (2021). Path planning of mobile robot in dynamic environment: Fuzzy artificial potential field and extensible neural network. *Artificial Life and Robotics*, *26*(1), 129-139.
15. Hewawasam, H. S., Ibrahim, M. Y., & Appuhamillage, G. K. (2022). Past, present and future of path-planning algorithms for mobile robot navigation in dynamic environments. *IEEE Open Journal of the Industrial Electronics Society*, *3*, 353-365.
16. Ayari, E., Hadouaj, S., & Ghedira, K. (2010, September). A fuzzy logic method for autonomous robot navigation in dynamic and uncertain environment composed with complex traps. In *2010 Fifth International Multi-conference on Computing in the Global Information Technology* (pp. 18-23). IEEE.