International Journal of Autonomous Robotics and Intelligent Systems
*Vol. 1, No. 1, June 2025* [93-105]

# Development of an Intelligent Sumo Robot based on Embedded Fuzzy Logic

M.S. Mohd Shukri[1], N. Abdul Rahim[1,2*] and B. Ilias[1,2]

[1]Faculty of Electrical Engineering & Technology, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia.
[2]Centre of Excellence for Intelligent Robotics & Autonomous Systems (CIRAS), Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia.

Received 5 July 2025, Revised 8 July 2025, Accepted 18 July 2025

**ABSTRACT**

*This paper presents the design and development of an intelligent sumo robot utilizing embedded fuzzy logic for real-time opponent detection and motion control. The system integrates analog infrared (IR) distance sensors and a fuzzy inference engine deployed on an Arduino Mega microcontroller to improve the robot's responsiveness and adaptability in dynamic environments. Through sensor calibration and optimized placement, the robot accurately detects opponent positions across multiple zones. A fuzzy rule base interprets the continuous sensor inputs and generates smooth, context-aware motor responses. The system was evaluated through simulation in MATLAB and embedded implementation, with test scenarios involving five fixed opponent positions and gradually varying inputs. Comparative results show that the fuzzy controller significantly outperforms traditional if-else logic, producing smoother motion, better coverage, and more reliable decision-making, with error values between MATLAB and Arduino outputs as low as 0.22%. The embedded fuzzy logic system closely replicates simulated behavior, validating its real-time feasibility on low-cost hardware. This approach demonstrates a scalable and effective solution for intelligent mobile robotics.*

**Keywords:** Embedded fuzzy logic, autonomous mobile robot, decision-making, analog infrared sensors, embedded systems

## 1. INTRODUCTION

Sumo robotics presents a distinct challenge in the field of autonomous mobile systems, as robots must make fast, strategic decisions in a highly dynamic and confined arena. Unlike structured industrial environments, sumo arenas demand that robots detect and respond to unpredictable opponent movement, while simultaneously avoiding the ring boundaries. This necessitates high responsiveness, real-time data interpretation, and rapid action execution under uncertainty. These characteristics are consistent with the demands of mobile obstacle-avoidance platforms and differential drive systems seen in other domains, such as UAVs and mobile robots [1][2].

Conventional sumo robots often use rule-based control architectures, typically implemented as simple if-else decision trees. Although computationally efficient, these systems struggle with ambiguous sensor readings and non-binary inputs, leading to sluggish responses or misinterpretations of opponent behaviour. Such deterministic control schemes have proven inadequate in environments where noise, latency, and overlapping sensor inputs reduce decision clarity [3][4].

*norasmadi@unimap.edu.my

Moreover, many legacy designs depend heavily on digital sensors, which provide binary outputs and lack the precision needed for nuanced environmental interpretation. Infrared and ultrasonic sensors, for instance, exhibit improved performance when configured to provide continuous analog feedback, allowing for richer decision inputs in variable-range scenarios [5].

To overcome these limitations, this paper introduces an intelligent sumo robot system utilizing analog infrared (IR) sensors for continuous opponent distance monitoring, integrated with a fuzzy logic controller embedded in a microcontroller unit. The analog nature of the IR sensors ensures finer granularity in opponent detection, while fuzzy logic offers the ability to manage uncertainty, partial truths, and overlapping sensor conditions [4][6]. Similar techniques have been successfully applied to differential-drive mobile robots and UAVs to enhance responsiveness, route optimization, and obstacle detection accuracy [1][2].

This paper presents the design, implementation, and evaluation of the proposed system. Section 2 details the system architecture and hardware configuration. Section 3 discusses the design and integration of the fuzzy logic controller. Section 4 presents the calibration and embedded implementation process. Section 5 evaluates the system through comparative tests against traditional control logic. Section 6 provides a discussion of challenges and observed limitations, and Section 7 concludes with key findings and suggestions for future improvements.

## 2. SYSTEM OVERVIEW AND HARDWARE ARCHITECTURE

### 2.1 Mechanical Design and Materials

The robot's chassis is constructed using a combination of PETG and stainless steel to balance weight, strength, and flexibility, as shown in Figure 1. PETG provides shock resistance and ease of fabrication, making it suitable for enclosing electronic components, while stainless steel is employed to reinforce structural rigidity and lower the robot's center of gravity. The design ensures stability during pushing engagements and rapid directional changes. A compact, low-profile form factor was chosen to minimize tipping risk, and silicone tires were used to provide high traction on the arena surface.

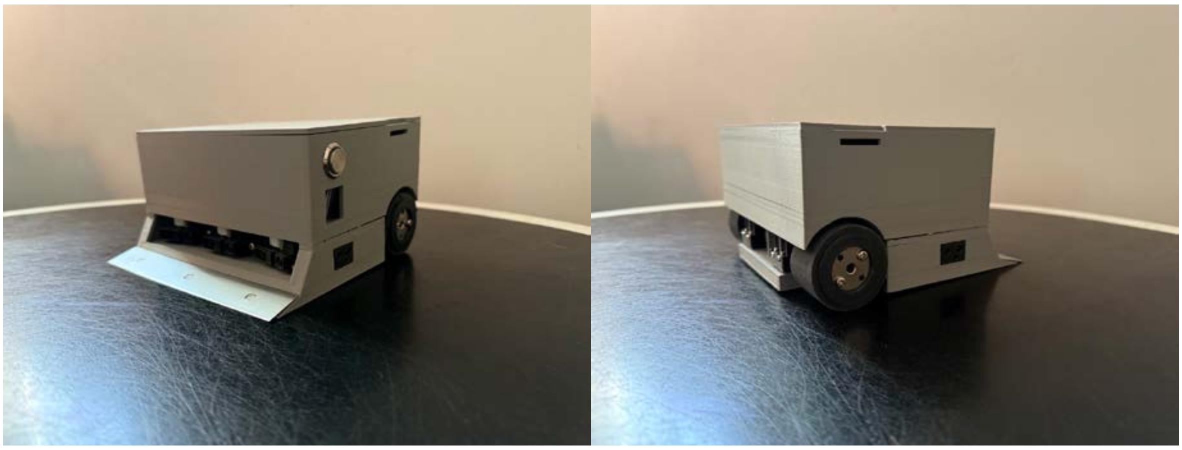

**Figure 1**. The 1 kg sumo robot

### 2.2 Sensor Layout and Placement

A total of seven IR sensors are used for opponent and boundary detection. Three analog IR distance sensors are mounted at the Front Left Sensor (FLS), Front Middle Sensor (FMS), and Front Right Sensor (FRS) to measure the distance of objects in front of the robot. These analog sensors provide continuous voltage outputs proportional to distance, making them ideal inputs for the fuzzy logic controller.

Two digital IR sensors are positioned on the left and right flanks of the robot to detect side-approaching opponents. Additionally, two analog IR reflectance sensors are mounted underneath the front corners of the chassis to detect the white perimeter line of the sumo ring (dohyo), ensuring the robot does not exit the competition area.

Sensor placement was optimized through empirical testing, comparing configurations with and without overlapping detection zones. An inward-angled configuration was chosen for the front analog sensors, enabling earlier and more accurate detection through intersecting fields of view.

### 2.3 Embedded Control and Power System

The control system is based on an Arduino Mega, selected for its large number of analog and digital I/O pins and sufficient processing power to execute the fuzzy inference engine. Sensor inputs are read continuously and processed to generate real-time motor commands. The logic is implemented in C++, with fuzzy inference rules translated from MATLAB simulations.

Motor actuation is managed by a Cytron MD13S motor driver, capable of handling up to 13 A continuous current. It drives a pair of 12 V Chihai CHR-GM25-370K DC gear motors that provide a balance of torque and speed, suitable for both quick movement and stable pushing performance.

The power architecture includes a DC-DC buck converter (5 A) to regulate input voltage to stable levels for sensors and control logic. A Li-ion battery pack powers the entire system, with onboard voltage regulation ensuring reliable operation across all components.

### 2.4 System Architecture Overview

The full system is organized in a modular control architecture in Figure 2. Sensor data is acquired via analog and digital inputs and processed by the fuzzy logic algorithm embedded in the microcontroller. The resulting motor control signals are transmitted as PWM outputs to the motor driver. The system continuously loops through sensing, inference, and actuation, enabling responsive and adaptive behaviour during sumo matches.

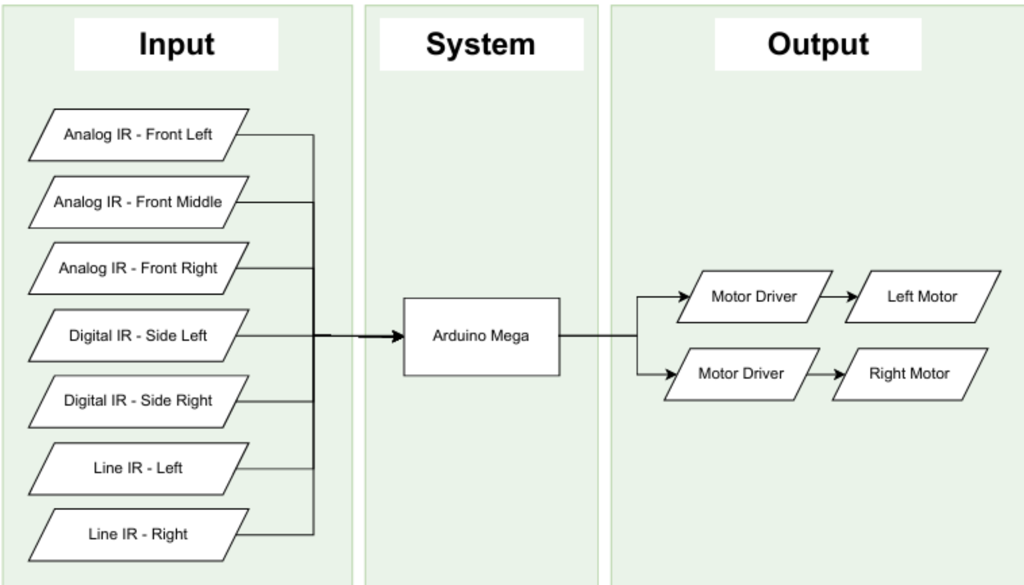


**Figure 2.** System block diagram

## 3. FUZZY LOGIC CONTROL DESIGN

The fuzzy logic control system is the core of the intelligent behaviour in the sumo robot, enabling it to interpret real-time sensor data and make adaptive decisions under uncertain and dynamic conditions. The controller processes continuous input values from analog infrared sensors and determines appropriate motor actions by evaluating fuzzy rules, allowing smoother and more context-sensitive responses compared to traditional control logic.

### 3.1 Input and Output Variables

The fuzzy controller utilizes three input variables corresponding to the front-facing analog IR sensors: FLS, FMS, and FRS. Each sensor provides a voltage that correlates with the distance to a nearby object, typically an approaching opponent. These inputs are used to assess the relative position of the opponent.

The controller produces two output variables Left Motor (LM) and Right Motor (RM). These outputs are expressed as PWM signals to control the speed and direction of the left and right motors independently

### 3.2 Fuzzification and Membership Function Design

To implement fuzzy logic decision-making, both the input sensor readings and output motor commands were mapped to linguistic variables using membership functions. The input from each front-facing analog IR sensor was categorized into three fuzzy sets (Attack, Ready, and Standby) based on distance. These sets were designed using overlapping trapezoidal and triangular functions to allow partial activation across zones (Figure 3, Table 1).

**Table 1** Input linguistic variable

| Linguistic Variable | Linguistic Value | Range (mm) |
|---|---|---|
| FLS, FMS, and FRS | Standby | 100-300 |
| | Ready | 200-600 |
| | Attack | 500-800 |

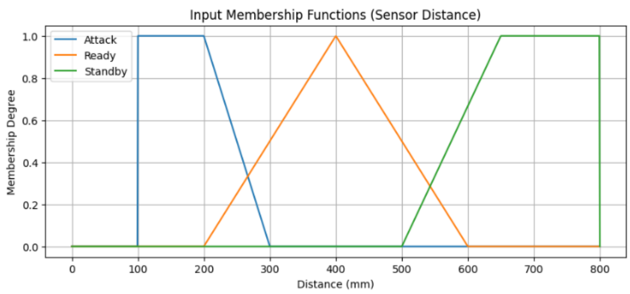


**Figure 3.** Input membership function

Similarly, the motor outputs for the left and right wheels were assigned to four fuzzy sets (Stop, Slow, Medium, and Fast) to enable smooth transitions in speed based on fuzzy inference (Figure 4, Table 2). These functions provided the flexibility required to handle ambiguous or overlapping sensor inputs in real time.

**Table 2** Output linguistic variable

| Linguistic Variable | Linguistic Value | Range (mm) |
|---|---|---|
| LM and RM | Stop | 0-30 |
| | Slow | 20-80 |
| | Medium | 70-170 |
| | Fast | 160-255 |


**Figure 4.** Output membership function

## 3.3 Rule Base Construction

A fuzzy rule base in Figure 5 was constructed to encode intuitive decision-making based on combinations of input conditions. Each rule follows an IF-THEN structure.

| Rule | FLS | FMS | FRS | LM | RM |
|---|---|---|---|---|---|
| 1 | Attack | Attack | Attack | F | F |
| 2 | Attack | Attack | Ready | M | F |
| 3 | Ready | Attack | Attack | F | M |
| 4 | Ready | Attack | Ready | F | F |
| 5 | Attack | Ready | Ready | M | F |
| 6 | Ready | Ready | Attack | F | M |
| 7 | Ready | Ready | Ready | M | M |
| 8 | Ready | Ready | Standby | S | M |
| 9 | Standby | Ready | Ready | M | S |
| 10 | Standby | Ready | Standby | M | M |
| 11 | Ready | Standby | Standby | ST | S |
| 12 | Standby | Standby | Ready | S | ST |
| 13 | Standby | Standby | Standby | ST | ST |
| 14 | Attack | Attack | Standby | S | F |
| 15 | Standby | Attack | Attack | F | S |
| 16 | Standby | Attack | Standby | F | F |
| 17 | Attack | Standby | Standby | ST | F |
| 18 | Standby | Standby | Attack | F | ST |
| 19 | Standby | Ready | Attack | F | M |
| 20 | Attack | Ready | Standby | M | F |

**Figure 5.** The fuzzy rules

These rules are designed to drive the robot toward an opponent detected in any of the three front zones while allowing for curved trajectories and smooth course corrections. All applicable rules are evaluated simultaneously, with overlapping activation levels.

### 3.4 Inference and Defuzzification

The fuzzy inference engine applies the Mamdani method to evaluate all active rules in parallel. Using the minimum operation (T-norm) for conjunction and the maximum operation (S-norm) for aggregation, the controller calculates the output fuzzy set for each motor.

Defuzzification is performed using the centroid of area (COA) method. This converts the aggregated fuzzy outputs into crisp PWM values for LM and RM, resulting in smooth and responsive motion transitions as shows in Figure 6.

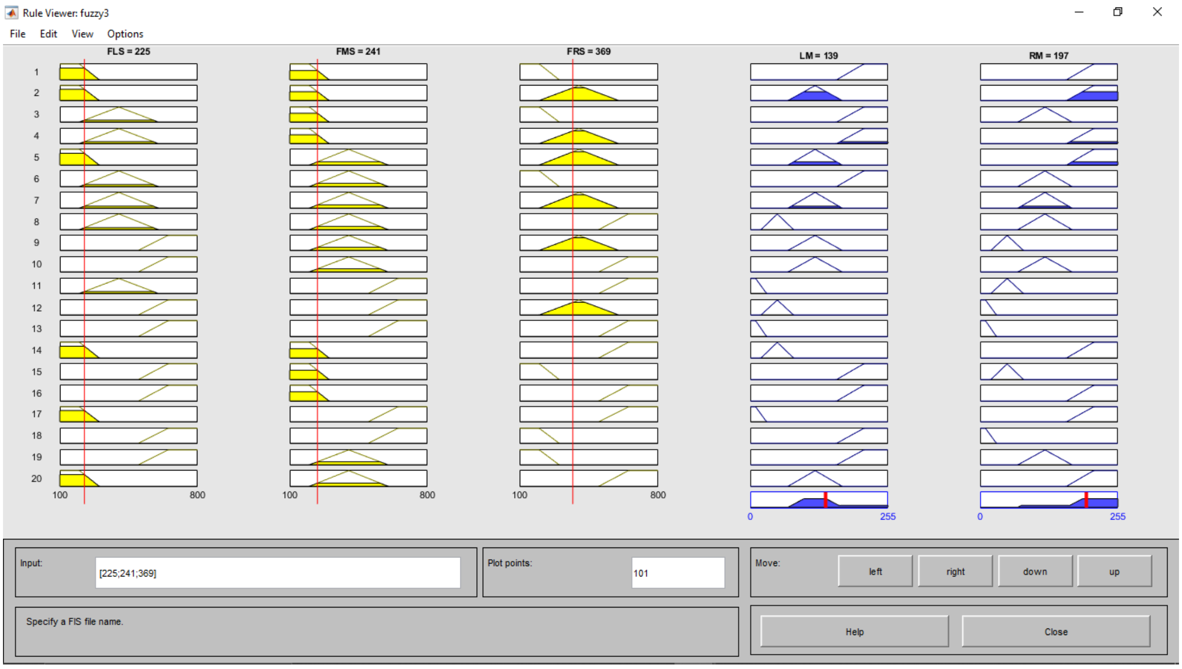


**Figure 6.** Defuzzification in MATLAB

### 3.5 Simulation and Embedded Implementation

Before implementation on the physical robot, the fuzzy inference system was designed and tested using MATLAB's Fuzzy Logic Toolbox. Membership functions, rule sets, and control surfaces were visualized and adjusted for optimal performance. Surface plots revealed smooth transitions in motor output across varying opponent positions.

Once validated, the fuzzy logic system was implemented in C++ on an Arduino Mega. The fuzzification logic, rule evaluations, and defuzzification computations were manually coded for embedded real-time execution. Testing confirmed that the Arduino-based controller closely reproduced the MATLAB simulation behaviour, with minimal deviation due to floating-point limitations and resolution differences.

## 4. SYSTEM INTEGRATION AND CALIBRATION

The performance of the intelligent sumo robot heavily depends on the accuracy of sensor readings and the reliability of the embedded control system. This section describes the calibration of analog IR distance sensors, the optimization of sensor placement, and the embedded integration required for real-time operation.

### 4.1 Analog IR Sensor Calibration

The front-facing analog IR distance sensors (Sharp GP2Y0A21YK0F) were calibrated to ensure accurate and consistent distance measurement. Each sensor was tested across a range of 10 cm to 80 cm, with corresponding output voltages recorded and compared to ground-truth distances using a ruler and multimeter. To reduce noise and suppress transient spikes during measurement, multiple analog samples were collected for each reading, and a median filter was applied.

The relationship between the analog-to-digital converter (ADC) reading and distance was modeled using an empirically derived rational function, expressed as in Equation (1):

$$D = \frac{67620}{x-9} - 40 \quad (1)$$

where $D$ is the estimated distance in millimeters, and $x$ is the median-filtered analog sensor value. This equation was implemented in the Arduino Mega firmware to convert real-time sensor readings into usable distance estimates for fuzzy logic processing. The calibrated response curve minimized measurement error and improved detection accuracy across the sensor's effective range. Output values were clamped to a maximum of 800 mm to avoid instability at longer ranges.

This equation was implemented in the Arduino code to convert real-time voltage readings into distance estimates. The calibrated response curve minimized measurement errors and improved the responsiveness of the fuzzy logic controller in detecting nearby opponents

### 4.2 Sensor Placement Strategy

Two sensor configurations in Figure 7 were evaluated to determine the optimal detection coverage: outward-facing (Placement A) and inward-facing (Placement B). Placement B was selected, as angling the sensors inward created overlapping detection zones that allowed the robot to detect opponents earlier and across a broader field of view. The calculated interception point between the sensor beams enhanced the robot's ability to detect diagonal or fast-approaching threats, improving its defensive and offensive responses.

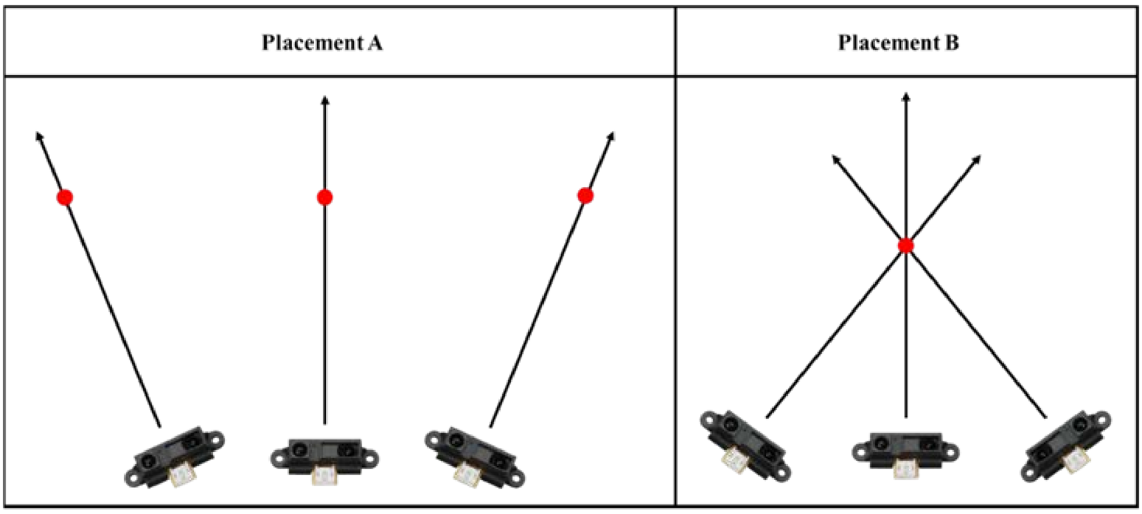


**Figure 7**. Sensor placement

### 4.3 Embedded System Integration

The control system was implemented using an Arduino Mega, which continuously reads analog and digital sensor data and executes fuzzy inference in real time. The PWM outputs generated by the fuzzy controller are fed to a Cytron MD13S motor driver, which actuates the left and right DC motors accordingly. All sensor and actuator connections were arranged on a custom Fritzing-based circuit, ensuring clean signal routing and reliable power distribution.

To compensate for the absence of encoders, motor balancing was conducted manually. Slight differences in motor response were addressed by calibrating the PWM output ratio to maintain straight-line movement during testing. With all subsystems integrated, the robot was functionally validated in controlled scenarios, confirming its ability to detect, decide, and act effectively.

## 5. EXPERIMENTAL EVALUATION AND RESULTS

To validate the effectiveness of the fuzzy logic controller, the intelligent sumo robot was evaluated through simulation and physical tests under controlled conditions. The performance was compared against a traditional if-else logic system using predefined opponent positions and gradual input transitions. The results demonstrate the advantages of fuzzy logic in terms of adaptability, response smoothness, and coverage across variable sensor inputs.

### 5.1 Test Conditions and Data Collection

The evaluation was conducted using five fixed opponent positions labeled A to E, simulating common relative locations in the sumo ring. For each position, analog readings were recorded from the front left (FLS), front middle (FMS), and front right (FRS) sensors. These sensor values were fed into both the MATLAB-based fuzzy logic simulation and the Arduino-implemented embedded controller. Corresponding motor speed outputs (LM and RM) were recorded and compared.

In addition to static position tests, a dynamic scenario was introduced where sensor inputs were gradually increased to observe the smoothness and continuity of the control response.

### 5.2 Fuzzy Logic Controller Response

The fuzzy logic controller produced smooth and context-sensitive output transitions. As shown in Table 3, the LM and RM values varied proportionally with input sensor values, resulting in curved trajectories toward the opponent. For example, when the opponent was detected predominantly on one side, the motor speed differential caused the robot to arc in that direction, improving alignment without abrupt turning.

**Table 3** Fuzzy logic results

| Position | LM (PWM) | Pertinence | | | | RM (PWM) | Pertinence | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Stop | Slow | Medium | Fast | | Stop | Slow | Medium | Fast |
| ***A*** | 11.10 | 0.91 | 0.00 | 0.00 | 0.00 | 11.10 | 0.91 | 0.00 | 0.00 | 0.00 |
| ***B*** | 139.49 | 0.00 | 0.00 | 0.59 | 0.12 | 195.53 | 0.00 | 0.00 | 0.12 | 0.59 |
| ***C*** | 218.51 | 0.00 | 0.00 | 0.00 | 1.00 | 218.51 | 0.00 | 0.00 | 0.00 | 1.00 |
| ***D*** | 216.11 | 0.00 | 0.00 | 0.00 | 0.75 | 199.18 | 0.00 | 0.00 | 0.12 | 0.75 |
| ***E*** | 114.84 | 0.00 | 0.05 | 0.73 | 0.00 | 207.41 | 0.00 | 0.00 | 0.05 | 0.73 |

In the table, the LM (PWM) and RM (PWM) values demonstrate how the controller adjusts motor outputs based on the pertinence (Stop, Slow, Medium, Fast) for each position. The values indicate gradual transitions between states, with the motors' speed increasing in response to higher pertinence values (i.e., moving from Stop to Fast). This smooth control ensures the robot's motion is continuous and closely aligned with the fuzzy control surface plots generated in MATLAB.

### 5.3 Traditional If-Else Logic Comparison

In contrast, the traditional if-else logic controller responded with abrupt and binary motor commands based solely on threshold crossings. As recorded in Table 4, motor outputs remained static within defined conditions and failed to respond in positions where sensor values did not meet hard-coded thresholds (Positions A to E).

**Table 4** Traditional if-else results

| Position | Condition | LM (PWM) | RM (PWM) |
|---|---|---|---|
| ***A*** | else | 0 | 0 |
| ***B*** | 7 | 120 | 120 |
| ***C*** | 1 | 255 | 255 |
| ***D*** | 7 | 120 | 120 |
| ***E*** | else | 0 | 0 |

The inability to interpolate between conditions resulted in jerky movement and inactivity in some valid scenarios, which would be critical in a real match.

### 5.4 Smoothness and Curve Response Validation

To evaluate the smoothness and adaptability of the fuzzy controller, two sets of tests were conducted. The first involved five predefined opponent positions (Positions A to E), each representing distinct sensor input conditions. The second involved a gradual increase in input sensor values to simulate the continuous approach of an opponent.

Table 5 presents the output of the fuzzy logic and traditional controllers for fixed positions. The fuzzy system produced varied motor outputs (LM and RM) that reflected the relative position of the opponent, enabling directional movement and curved paths. For instance, at Position B, the fuzzy controller adjusts the motors with values of 139.49 for LM and 195.53 for RM, resulting in more nuanced movement compared to the traditional approach. In contrast, the traditional if-else logic responded with either full activation or complete inactivity, depending on whether predefined thresholds were met. This is evident at Positions A and E, where the traditional controller outputs 0 for both motors, resulting in no movement, despite the presence of valid sensor inputs.

**Table 5** Response to predefined opponent positions

| Position | Fuzzy | | Traditional | |
|---|---|---|---|---|
| | LM (PWM) | RM (PWM) | LM (PWM) | RM (PWM) |
| ***A*** | 11.10 | 11.10 | 0 | 0 |
| ***B*** | 139.49 | 195.53 | 120 | 120 |
| ***C*** | 218.51 | 218.51 | 255 | 255 |
| ***D*** | 216.11 | 199.18 | 120 | 120 |
| ***E*** | 114.84 | 207.41 | 0 | 0 |

Table 6 shows motor output responses under conditions of gradually increasing sensor input values. As the FLS, FMS, and FRS values gradually increased from row 1 to row 7, the fuzzy logic controller demonstrated a smooth and continuous adaptation in LM and RM values, reflecting its ability to adjust motor output progressively as sensor input intensified. In contrast, the traditional if-else controller maintained zero output until a specific threshold was reached, after which it abruptly switched to either 120 or 255 PWM values, exhibiting a step-like behaviour.

**Table 6** Response to gradual input changes

| No | FLS | FMS | FRS | Fuzzy | | Traditional | |
|---|---|---|---|---|---|---|---|
| | | | | LM (PWM) | RM (PWM) | LM (PWM) | RM (PWM) |
| 1 | 800 | 637 | 800 | 10.44 | 10.44 | 0.00 | 0.00 |
| 2 | 685 | 544 | 688 | 92.56 | 92.56 | 0.00 | 0.00 |
| 3 | 570 | 451 | 576 | 107.28 | 107.28 | 0.00 | 0.00 |
| 4 | 455 | 358 | 464 | 120.00 | 120.00 | 120.00 | 120.00 |
| 5 | 340 | 265 | 352 | 168.93 | 168.93 | 120.00 | 120.00 |
| 6 | 223 | 173 | 238 | 191.19 | 191.19 | 120.00 | 120.00 |
| 7 | 107 | 121 | 119 | 219.17 | 219.17 | 255.00 | 255.00 |

This contrast is visually illustrated in Figure 8, which plots the change in motor output (LM and RM) for both fuzzy and traditional controllers across the seven input scenarios. The fuzzy output follows a smooth, rising curve, while the traditional logic produces a staircase pattern with flat intervals and sudden jumps. This emphasizes the fuzzy logic system's superior responsiveness to gradual input transitions, allowing for more natural and effective motion behavior in dynamic environments.

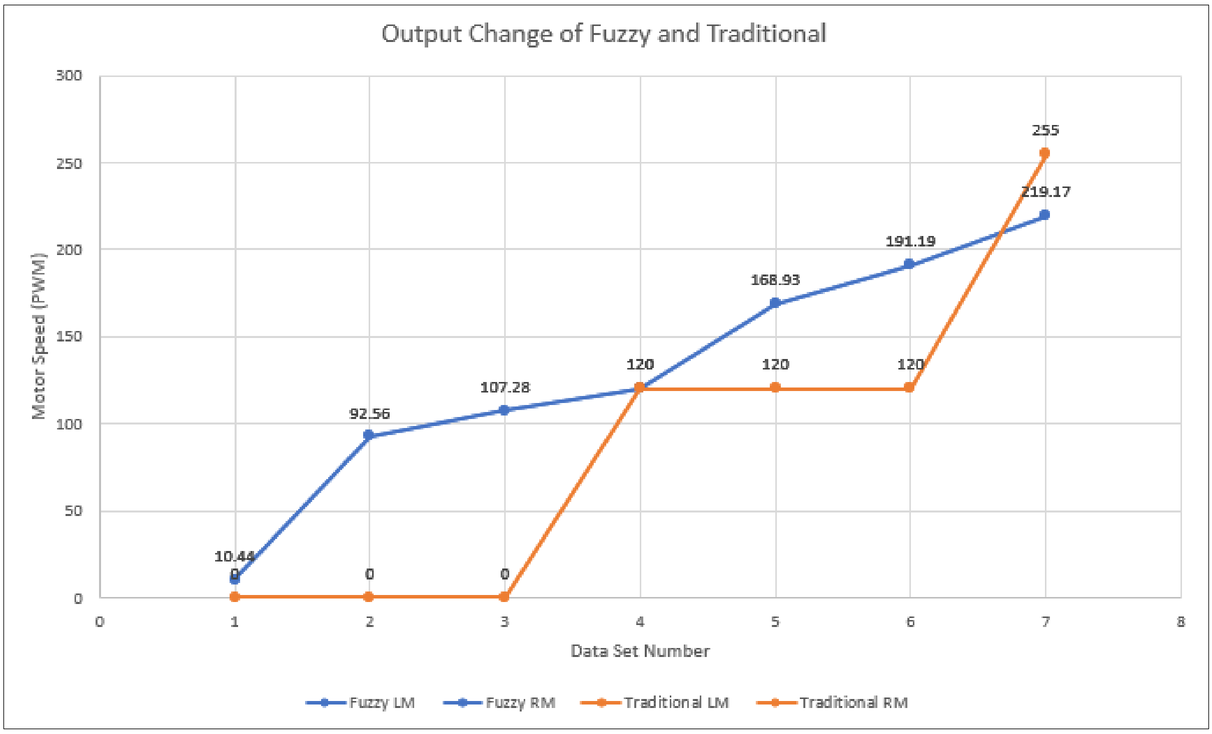


**Figure 8.** Output responses comparison plot

### 5.5 Simulation Accuracy: MATLAB vs. Arduino

Table 7 presents the comparison between the MATLAB simulations and the Arduino IDE results, showing the accuracy of the fuzzy logic system implementation. MATLAB was used to simulate the fuzzy logic, and this comparison helps to verify that the simulation results are consistent with the actual performance of the system when run on the Arduino IDE (on the microcontroller). The error values in percentage (LM and RM) are minimal, indicating that the fuzzy logic system behaves similarly in both MATLAB and Arduino environments.

**Table 7** Simulation validation

| Position | Arduino IDE | | MATLAB | | Error | |
|---|---|---|---|---|---|---|
| | LM (PWM) | RM (PWM) | LM (PWM) | RM (PWM) | LM (%) | RM (%) |
| ***A*** | 11.10 | 11.10 | 10.40 | 10.40 | 6.31 | 6.31 |
| ***B*** | 139.49 | 195.53 | 139.00 | 197.00 | 0.35 | 0.75 |
| ***C*** | 218.51 | 218.51 | 219.00 | 219.00 | 0.22 | 0.22 |
| ***D*** | 216.11 | 199.18 | 217.00 | 201.00 | 0.41 | 0.91 |
| ***E*** | 114.84 | 207.41 | 116.00 | 209.00 | 1.01 | 0.77 |

The slight discrepancies in error percentages are attributed to differences in numerical resolution and floating-point computation between MATLAB and the Arduino platform, but they are negligible and confirm the reliability of the translation from simulation to embedded code.

## 6. DISCUSSION

The experimental evaluation confirms that the fuzzy logic-based control system offers several advantages over traditional if-else logic, particularly in dynamic, uncertain environments such as sumo robot competitions. Key observations from both simulation and physical testing highlight the benefits and limitations of the proposed approach.

### 6.1 Improved Responsiveness and Adaptability

The fuzzy logic controller demonstrated the ability to respond smoothly and continuously to varying opponent positions. Unlike the rigid decision structure of if-else logic, fuzzy logic evaluates all applicable rules in parallel and produces proportionally blended motor commands. This allowed the robot to perform curved trajectories toward opponents and maintain engagement even when sensor inputs were partially activated. In contrast, the traditional system failed to respond in two out of five tested positions due to its dependency on predefined threshold conditions.

### 6.2 Smooth Output and Motion Behaviour

The use of overlapping membership functions and centroid-based defuzzification enabled the robot to produce fluid motor responses across a wide range of input values. This resulted in smoother transitions, more controlled movement, and reduced oscillation or jitter—traits that are critical for maintaining stability during pushing actions or evasive manoeuvres. Gradual input changes were met with gradual output responses, eliminating the abrupt behaviour seen in traditional control methods.

### 6.3 Simulation and Embedded Fidelity

The MATLAB-based fuzzy controller served as a reliable prototype for system behaviour. When translated to the Arduino platform, the embedded implementation closely matched the simulation results with minimal numerical deviation. This consistency validates the transferability of fuzzy logic systems from software simulation to hardware execution, even on resource-limited microcontrollers.

### 6.4 Design Challenges and System Constraints

Although the fuzzy logic controller improved motion smoothness and decision adaptability, several practical challenges emerged during implementation. The most critical limitation is the system's open-loop nature. Without an Inertial Measurement Unit (IMU) or encoder-based feedback, the robot lacks awareness of its own velocity, orientation, or displacement over time. All decisions are based on instantaneous sensor readings, which makes the system susceptible to cumulative errors, especially in situations involving wheel slip, surface variation, or asymmetric traction. These deviations cannot be corrected without a closed-loop feedback mechanism.

Another major constraint lies in the behaviour of the analog infrared (IR) distance sensors, which are inherently sensitive to electrical noise and ambient environmental factors. During testing, the sensors frequently exhibited spiking and unstable values, even under controlled conditions. This introduced uncertainty in the input to the fuzzy inference system, potentially causing erratic or suboptimal motor outputs. Software-based smoothing techniques, such as rolling average filters, were implemented to mitigate this, but they introduced a slight delay in response and could not fully eliminate transient errors.

Furthermore, environmental factors such as lighting conditions and surface reflectivity also influenced sensor consistency. This added another layer of complexity to reliable opponent detection and boundary sensing, especially under rapid movements or angled surfaces. Overall, the limitations highlight the importance of robust sensor calibration, filtering, and the future integration of closed-loop feedback for enhanced stability and accuracy.

## 7. CONCLUSION AND FUTURE WORK

This paper presented the design and implementation of an intelligent sumo robot utilizing embedded fuzzy logic for adaptive and responsive motion control. The system integrated analog infrared (IR) sensors with a rule-based fuzzy inference engine to interpret proximity data and control motor actions in real time. The fuzzy logic controller enabled smooth, continuous movement by evaluating overlapping sensor inputs and generating context-aware motor outputs.

Through simulation and physical testing, the fuzzy controller consistently outperformed traditional if-else logic, particularly in terms of adaptability, decision smoothness, and consistent engagement with opponents. The embedded implementation closely matched the MATLAB simulations, confirming the fidelity and reliability of the fuzzy control algorithm under real-time constraints.

While the system successfully demonstrated the benefits of fuzzy logic in autonomous robotics, certain limitations remain. The open-loop architecture lacks motion feedback, and analog sensor variability required filtering and calibration. These findings highlight opportunities for further enhancement.

Future improvements will focus on the following areas:

- Closed-loop feedback integration using encoders or an inertial measurement unit (IMU) to enhance motion accuracy and stability.
- Wireless data logging and communication for real-time monitoring and performance analysis.
- Adaptive or hybrid fuzzy logic combining machine learning techniques, such as reinforcement learning or neuro-fuzzy systems, to refine decision-making.

By addressing these areas, the robot can evolve into a more autonomous, intelligent, and competitive platform, reinforcing the value of fuzzy logic in embedded robotic control systems.

## ACKNOWLEDGEMENTS

Authors are grateful to the Faculty of Electrical Engineering & Technology of UniMAP for providing the resources and support to this work.

## REFERENCES

[1] X. Yang, S. Bai, Y. Cao, J. Cai, S. Liu, H. Chen, 2024. Research on Fuzzy Control System for Two-Wheeled Differential Drive Robot, In *IMCEC 2024 - IEEE 6th Adv. Inf. Manag. Commun. Electron. Autom. Control Conf.*, vol. 6, pp. 1288–1292, doi: 10.1109/IMCEC59810.2024.10575470.

[2] E. A. Otsapa, S. M. Sani, O. A. Ayofe, 2022. Design and implementation of obstacle detection system for an unmanned aerial vehicle, In *Proc. 2022 IEEE Niger. 4th Int. Conf. Disruptive Technol. Sustain. Dev. NIGERCON 2022*, pp. 1–5, doi: 10.1109/NIGERCON54645.2022.9803168.

[3] P. D. Pour, K. M. J. Alsayegh, M. A. Jaradat, 2022. Type-2 Fuzzy Adaptive PID Controller for Differential Drive Mobile Robot: A Mechatronics Approach. In *2022 Adv. Sci. Eng. Technol. Int. Conf. ASET 2022*, no. 1, pp. 1–6, doi: 10.1109/ASET53988.2022.9734882.

[4] Y. Jeong, W. S. Jeong, J. Y. Shin, S. E. Lee, 2023. The Design of Embedded Fuzzy Logic Controller for Autonomous Mobile Robots. In *Proc. - Int. SoC Des. Conf. 2023, ISOCC 2023*, pp. 145–146, doi: 10.1109/ISOCC59558.2023.10396118.

[5] S. The, E. E. Mathias, J. F. Valesco, K. W. Noverianto, R. Elta, B. Juarto, 2023. Accuracy Evaluation of Infrared and Ultrasonic Sensors for Car Parking Systems with Confusion Matrix and RMSE. In *Proc. - 2023 3rd Int. Conf. Electron. Electr. Eng. Intell. Syst. Responsible Technol. Sustain. Humanit. ICE3IS 2023*, pp. 448–453, doi: 10.1109/ICE3IS59323.2023.10335280.

[6] A. De La Fuente-Martinez, L. M. Garcia-Casas, J. R. Garcia-Martinez, E. E. Cruz-Miguel, O. A. Barra-Vazquez, L. F. Olmedo-Garcia, 2023. Fuzzy Logic-Based Temperature Controller for a PLA Storage Container. In *2023 IEEE Int. Conf. Eng. Veracruz, ICEV 2023*, pp. 13–17, doi: 10.1109/ICEV59168.2023.10329744.

[7] "GitHub - alvesoaj/eFLL: eFLL (Embedded Fuzzy Logic Library) is a standard library for Embedded Systems." [Online]. Available: https://github.com/alvesoaj/eFLL [Accessed Jul. 07, 2025].