

Article

Obstacles Avoidance for Mobile Robot Using Type-2 Fuzzy Logic Controller

Mohammad Al-Mallah¹, Mohammad Ali² and Mustafa Al-Khawaldeh^{1,*} ¹ Department of Mechatronics Engineering, Philadelphia University, P.O. Box 1, Amman 19392, Jordan² Department of Electrical Engineering, Philadelphia University, P.O. Box 1, Amman 19392, Jordan

* Correspondence: malkhawaldeh@philadelphia.edu.jo

Abstract: Intelligent mobile robots need to deal with different kinds of uncertainties in order to perform their tasks, such as tracking predefined paths and avoiding static and dynamic obstacles until reaching their destination. In this research, a Robotino[®] from Festo Company was used to reach a predefined target in different scenarios, autonomously, in a static and dynamic environment. A Type-2 fuzzy logic controller was used to guide and help Robotino[®] reach its predefined destination safely. The Robotino[®] collects data from the environment. The rules of the Type-2 fuzzy logic controller were built from human experience. They controlled the Robotino[®] movement, guiding it toward its goal by controlling its linear and angular velocities, preventing it from colliding obstacles at the same time, as well. The Takagi–Sugeno–Kang (TSK) algorithm was implemented. Real-time and simulation experimental results showed the capability and effectiveness of the proposed controller, especially in dealing with uncertainty problems.

Keywords: mobile robot; Robotino[®]; static and dynamic obstacle-avoidance environment; Type-2 fuzzy logic controller; wireless sensor network



Citation: Al-Mallah, M.; Ali, M.; Al-Khawaldeh, M. Obstacles Avoidance for Mobile Robot Using Type-2 Fuzzy Logic Controller. *Robotics* **2022**, *11*, 130. <https://doi.org/10.3390/robotics11060130>

Received: 27 September 2022

Accepted: 8 November 2022

Published: 16 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, robots are an inseparable part of our life. Robots with movement ability impose themselves through many applications, including medical facilities, hospitality, entertainment, package delivery, space, and military. Recently, mobile robots have been a controlling contributor to human development and one of the fastest growth fields of scientific research. They have displayed their abilities in helping and substituting humans in many applications with high efficiency [1].

The obstacle avoidance is an important feature in mobile robots that enables them to reach their destination point collision free. This necessitates providing them with a decision-making capability for planning their path autonomously and reacting to the hazards that may hinder their movements. However, this is no longer easily achieved by using classical control approaches without prior information available about the environment and using intelligent control [1]. Furthermore, some of the intelligent control methods cannot handle the high level of uncertainties of sensors, actuators, and environment [2].

For achieving autonomous obstacle avoidance, numerous control strategies have been developed, among which is the Type-2 fuzzy logic control. Fuzzy logic control is considered the most vastly used technique for designing controllers that manage suitable performance in many real-world applications [3]. For example, it was used to design a controller capable of introducing a safe Robotino[®] and tracking its predefined target, as in Ref. [4]. A fuzzy logic controller with 153 fuzzy rules was utilized for controlling the Robotino[®] path-tracking issue, while another fuzzy logic controller with 27 fuzzy rules was applied for the Robotino[®] obstacle-avoidance feature, using the Sugeno fuzzy algorithm. Many real-time experiments reflected good abilities of the proposed controllers. Moreover, an autonomous mobile robot was designed and implemented by using a fuzzy logic controller,

along with an ultrasonic sensor, in Ref. [5]. The results proved the effectiveness of the algorithm when taking the settling time and the overshoot as a performance index.

A Type-2 fuzzy system was used to design a robust controller, using C# language [6]. The simulation of obstacle avoidance has been performed, showing that the algorithm has the shortest path among other algorithms. Santiago and C. Chiu [7] applied an interval Type-2 fuzzy logic controller, along with a PID controller, for autonomous obstacle avoidance and wall following, using eight sonar sensors, in the presence of environmental uncertainties. The results proved the robustness of the proposed approach in an unknown environment and following the wall with good stability. A comparison between the Type-1 fuzzy logic controller and Type-2 fuzzy logic controller in mobile robot obstacle avoidance, using 12 infrared sensors, was presented by Ref. [8]. The experimental simulation results showed that both types of the fuzzy controllers behave similarly in most cases, but in the case of increasing uncertainties, the Type-2 fuzzy logic controller gives a better performance in terms of rapidity and precision.

A Type-2 fuzzy logic controller as a high-level mobile robot controller was also applied in Ref. [9]. In addition to orientation angle measurements, three sensors (i.e., left, front, and right) have been used as input to the controller. They help to determine the response action taken for linear and angular output velocities for tracking and positioning of the mobile robot in an unknown environment. The results proved smooth performance and decision-making on the correct acceleration and braking when considering the direction change of the target. Type-2 fuzzy logic controller and optical flow approach were also utilized for visual obstacle avoidance based on video acquisition and an image-processing algorithm to extract fruitful information about the environment from a robot's camera [10]. The simulation results have shown a good capability for the Type-2 fuzzy logic controller in obstacle avoidance. A Type-1 fuzzy logic and interval Type-2 fuzzy logic controllers were designed and implemented for a swarm robot system in Ref. [11]. Each of the mobile robots contained one compass sensor, three distance sensors, and one X-Bee communication module. The experimental simulation results have shown that the complex problem modeling Type-2 fuzzy logic has better performance than Type-1 fuzzy logic. Hierarchical fuzzy controllers based on Type-2 Fuzzy Logic System for mobile robot in a partially known environment with eight sensors were presented by Ref. [12]. A sliding mode was used to make the mobile robot chase the dynamic target and avoid obstacles simultaneously. The simulation results have proved the robustness of the performance of the Type-2 fuzzy logic controller.

A fuzzy-logic-based obstacle avoidance technique was used in Ref. [13]. This system could be controlled remotely and switched to avoid obstacles in an indoor environment. The omnidirectional wheel was adapted in the power system to perform translating and spinning movements for avoiding obstacles fast and flexibly in paths. The experiments proved the good performance of the proposed system in service environments. In Ref. [14], a fuzzy logic rules set was used to calculate the position of the robot concerning the road lane center during the movement. The Haar-cascade-classifier-based machine-learning technique has been utilized to detect different types of obstacles facing the robot in its path from source to destination. The simulation results of many experimentation scenarios have revealed effective performance of the proposed approach.

A Mamdani fuzzy reasoning system was used in Ref. [15] for designing a controller that was capable of reducing the amount of time spent on the programming of the controller. Twenty-five rules are used to cover a robot's possible interactions with various obstacles. The results of the MATLAB/Simulink simulation tools have verified the algorithms with real-time constraints and demonstrated the superior performance of the proposed controller with high adaptability to the environment and a sufficient level of accuracy. A system using fuzzy logic was proposed in Ref. [16] for detecting and avoiding different types of obstacles. Data coming from a ZED stereo camera were mainly exploited for making decisions based on fuzzy predicates. The simulated environment has proved that the proposed technique is robust.

The review of the literature reveals the available control techniques used for avoiding obstacles avoidance. The present study contributes to the existing literature through

building a simple control algorithm that can guide Robotino[®] safely and collision free to its predefined destination point.

The article is organized as follows: Section 2 presents the materials and methods. Section 3 presents the results of the simulation. The conclusion of the study is presented in Section 4.

2. Materials and Methods

The backbone of this research is to apply Type-2 fuzzy logic controller on mobile robot; thus, this section concentrates on giving a detailed idea about these main parts, as shown below:

2.1. Mobile Robot System Description

Robotino[®], shown in Figure 1, is a mobile robot made by Festo company with its omnidirectional drive, interfaces, learning system for training, sensors, and a platform for research and development [4].



Figure 1. Robotino[®] of Festo Didactic GmbH & Co.KG.

Robotino[®] has a high-performance controller that provides an intelligence system to use, numerous sensors, and an omnidirectional drive that consists of three independent wheels arranged at angles of 120° . All of these components are separately controllable, thus enabling the Robotino[®] to travel freely in all directions. Furthermore, it has an open electrical interface to easily install any additional sensors or actuators [17].

2.1.1. Unit Drive

Three individual omnidirectional-drive units are attached at an angle of 120° to drive the Robotino[®]. Each of the three drive units, as shown in the Figure 2, consists of the following elements: DC motor with nominal speed of 3600 rpm and nominal current of 0.9 A, all-way roller that has the capability of traveling in any desired direction, gear unit with a gear ratio of 16:1, toothed belt with gear wheels, and incremental encoder.

The interaction of the three drive units with the multidirectional wheel gives possibilities to produce orientation of motion. This gives the advantage of roam-free traveling in any direction, without the demand for tuning with different speed-control methods of the motor. The three D.C. motors are used with their speed measurements collected by three shaft encoders mounted on each motor, while their speed control is made by their low-level individual PID controller used for this purpose [18].

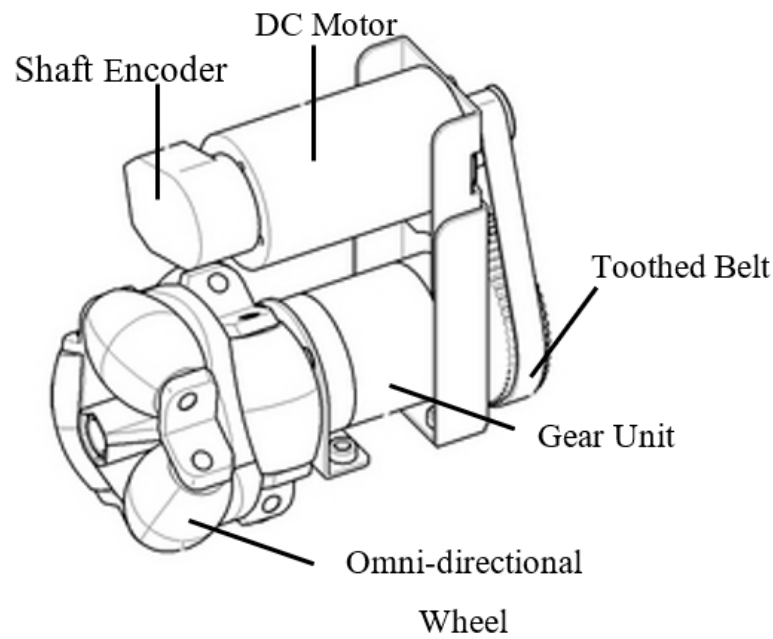


Figure 2. Robotino® Driving Unit.

2.1.2. Infrared (IR) Distance Sensors

Three infrared distance sensors are integrated around the circumference of the chassis of Robotino® for anti-collision purposes and detecting targets at an angle of 40° between each other. These sensors have the ability to measure an object’s relative distance of (4 to 30) cm [19]. Each of the infrared distance sensors includes connection to one analog output signal and consists of an emitter and receiver. The infrared distance sensor works on the principle of the reflected light waves with power supply. Figure 3 shows the output distance characteristics of the infrared sensor.

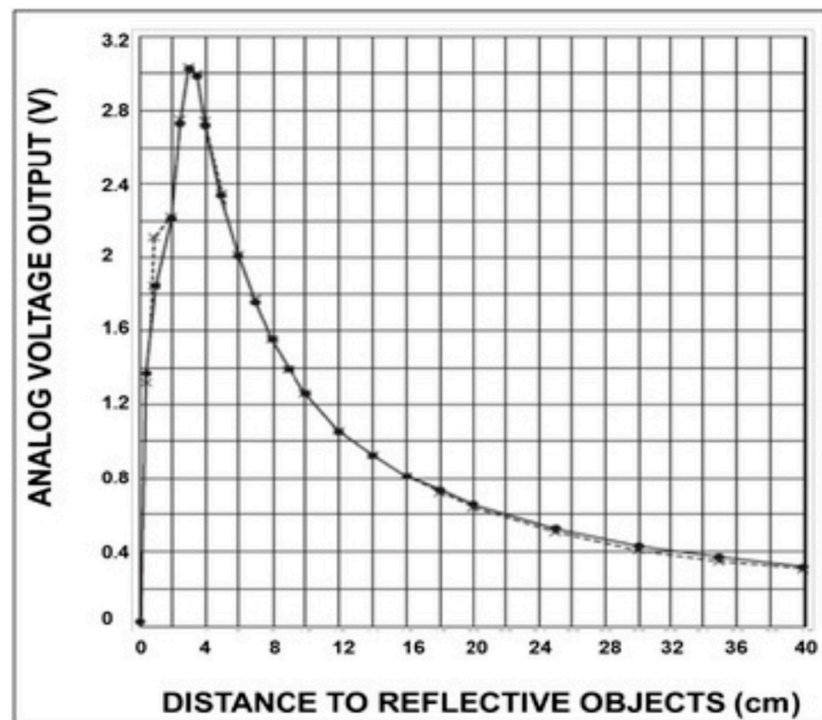


Figure 3. IR Relationship Between Distance and Voltages.

2.1.3. Odometry Mechanism

The odometry means the use of data from motion sensors to estimate (not determine) any change in position relative to starting point over time. The odometry gives the possibility of obtaining the actual position (X, Y) of the Robotino[®] with its orientation (θ) by integrating the linear velocity and angular velocity (Vx and Vy, respectively).

2.2. Type-2 Fuzzy Logic Controller Design for Robotino[®] Guiding

Far away from the details of the Type-2 fuzzy logic system design, the main design issues are highlighted in this section. A Type-2 fuzzy logic controller was designed by obtaining a comprehensive knowledge about the Robotino[®] dynamic system and its ability to accept a high-level algorithm to make the Robotino[®] reach its predefined path safely.

2.2.1. Type-2 Fuzzy Logic System

The typical architecture of a Type-2 fuzzy logic controller is similar to that of a Type-1 fuzzy logic controller, but with an extra step, which is called type reducer, as shown in Figure 4. It contains five components: a fuzzifier, a fuzzy rule base, an inference engine, a type-reducer, and a defuzzifier [20].

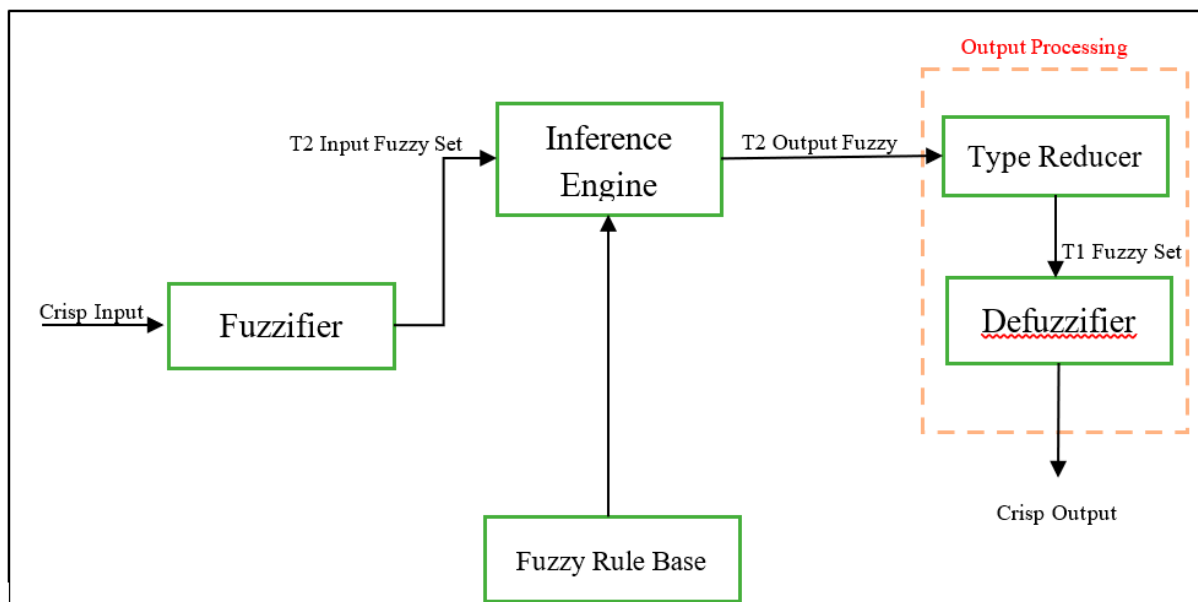


Figure 4. Type-2 Fuzzy Logic Controller Structure.

The main functions of each block of Figure 4 can be explained as follows:

- Fuzzifier: It converts crisp inputs into Type-2 fuzzy sets.
- Fuzzy rule base: The rules in a Type-2 fuzzy logic stay the same as in a Type-1 fuzzy logic, but antecedents and consequents are performed by Type-2 fuzzy sets.
- Fuzzy inference engine: It assigns membership functions by using the rules in the rule base and operators such as union and intersection fuzzy operation to drive the output.
- Type-reducer: The Type-2 fuzzy outputs of the inference engine are transformed into Type-1 fuzzy sets. To obtain the scalar crisp control action, one must apply the defuzzification process based on one-dimensional fuzzy control action vector. Thus, the type-reducer brings this vector out of the two dimensions, using the following equations:

$$Y_{\text{left}}(z) = \min_{\forall z^i \in [f^i(z), \bar{f}^i(z)]} \left[\frac{\sum_{i=1}^I f^i Y^i(z)}{\sum_{i=1}^I f^i} \right] \tag{1}$$

$$Y_{\text{right}}(z) = \max_{\forall z^i \in [f^i(z), \bar{f}^i(z)]} \left[\frac{\sum_{i=1}^I f^i \bar{Y}^i(z)}{\sum_{i=1}^I f^i} \right] \tag{2}$$

$$\underline{Y}^i(z) = \underline{a}_0^i + \underline{a}_1^i z_1 + \dots + \underline{a}_p^i z_p \tag{3}$$

$$\bar{Y}^i(z) = \bar{a}_0^i + \bar{a}_1^i z_1 + \dots + \bar{a}_p^i z_p \tag{4}$$

$$A = \frac{1}{2} (Y_{\text{left}}(z) + Y_{\text{right}}(z)) \tag{5}$$

where \bar{Y}^i and \underline{Y}^i are the output of Type-2 upper and lower fuzzy sets, respectively; f^i and \bar{f}^i are a singleton value replacing the output firing set by defined interval; \underline{a}_p^i and \bar{a}_p^i are crisp values representing the center of gravity (the consequent Type-1 fuzzy set); z_p is the area of the upper and lower membership functions; and A is the average of Y_{left} and Y_{right} of the reduced set [12].

Defuzzification calculates the average to obtain the defuzzified value in a crisp manner by using the center of gravity.

Type-2 fuzzy logic systems are able to model levels of uncertainty that fuzzy logic Type-1 cannot.

2.2.2. Robotino® Guiding Type-2 Fuzzy Controller

The system block diagram is demonstrated in Figure 5. At the beginning, Robotino® is given a destination point coordinate to reach. The instant Robotino® position and orientation are obtained from different sensors and compared with the desired values. The sensors' different readings are used to adjust the parameters of the Type-2 fuzzy logic to bring out the required distance and orientation that should be followed to meet the predefined goal.

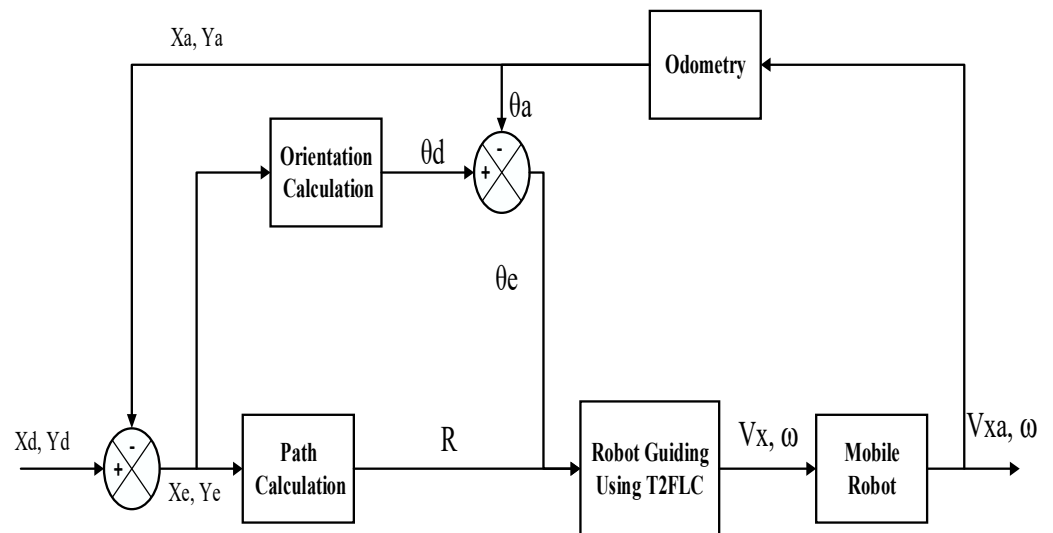


Figure 5. System Block Diagram for Robotino® Guiding Using Type-2 Fuzzy Control. X_d, Y_d , and θ_d are the desired values in X and Y coordinates' position and orientation, respectively. X_a, Y_a , and θ_a are the actual values in X and Y coordinates' position and orientation, respectively. X_e, Y_e , and θ_e are the errors and are calculated by taking the difference between the desired and the actual coordinates, using a comparator, as shown below:

$$X_e = X_d - X_a \tag{6}$$

$$Y_e = Y_d - Y_a \tag{7}$$

$$\theta_e = \theta_d - \theta_a \tag{8}$$

$$\theta_d = \tan^{-1}\left(\frac{Y_e}{X_e}\right) \tag{9}$$

$$R = \sqrt{(X_e)^2 + (Y_e)^2} \tag{10}$$

where R is the needed path resultant distance that is calculated in Equation (8).

Both the resultant instant distance (R) and orientation error (θ_e) are the input variables for the Type-2 fuzzy logic controller.

The Type-2 fuzzy logic controller block has three main processing parts:

Fuzzification Process

Since nobody can determine the completeness of the fuzzy rules which are related to specific applications, a set of 63 fuzzy production rules were used. This is based on the design requirements (i.e., these rules can be extended according to more requirements' design).

After many trials to obtain the best fuzzy set and its intervals for (R) and (θ_e) based on their possible variation, two fuzzifiers were obtained. The R fuzzifier has five Type-2 fuzzy sets spread over a universe of discourse of (0–7) m, with an interval for [0.8,0.25], which represents the footprint of uncertainty for both the (R) and (θ_e), as shown in Figures 6 and 7, respectively.

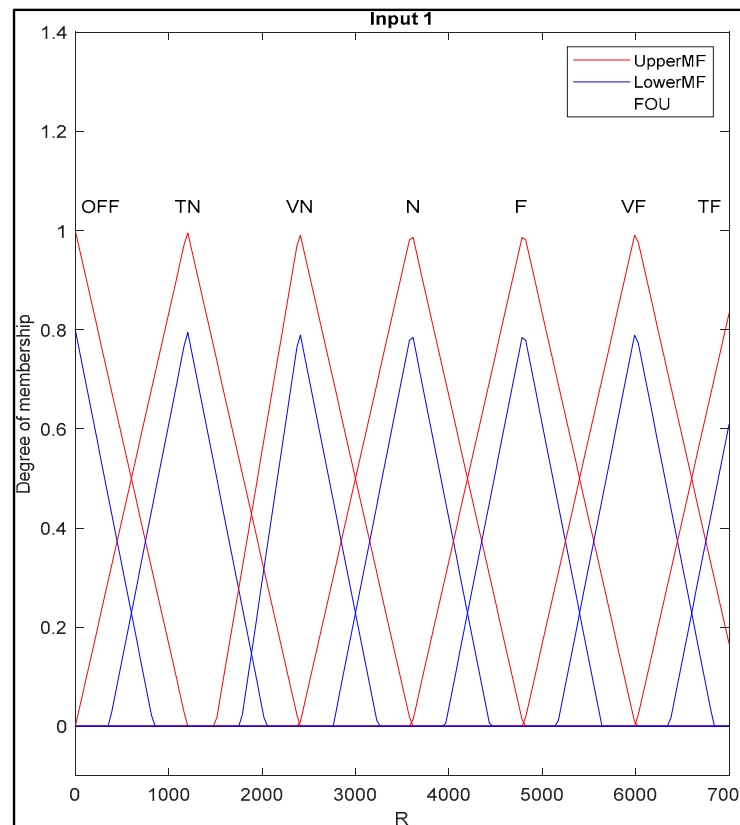


Figure 6. Resultant Type-2 Fuzzy Sets.

The five fuzzy Type-2 fuzzy sets have indicative linguistics that are named as follows: off (O), too near (TN), very near (VN), near (N), far (F) very far (VF), and too far (TF). Moreover, θ_e has a universe of discourse of ($-180^\circ, 180^\circ$), as shown in Figure 7.

The nine fuzzy Type-2 fuzzy sets have indicative linguistics: too negative (TN), very negative (VN), medium negative (MN), negative small (NS), zero (Z), small positive (SP), medium positive (MP), very positive (VP), and too positive (TP).

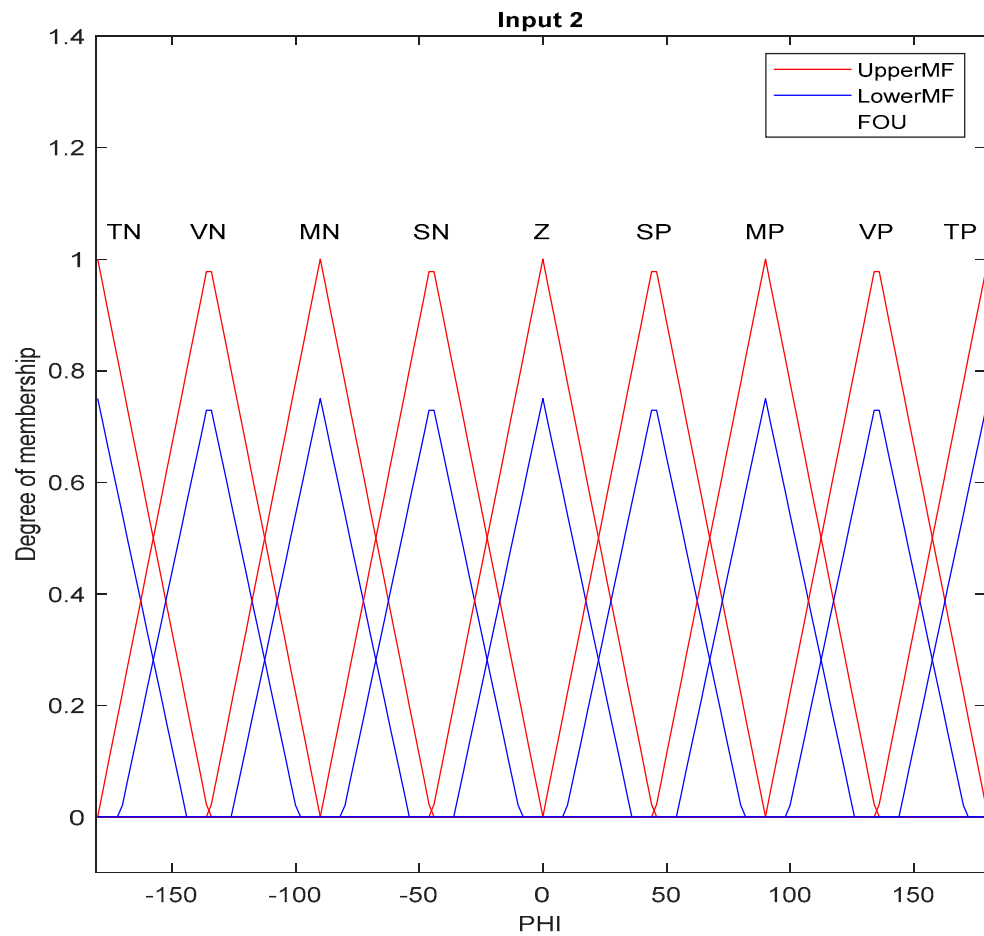


Figure 7. Orientation Type-2 Fuzzy Sets.

Fuzzy Production Rules

A fuzzy production rule has the following form:
 IF R is TN AND θ_e is PS THEN V_x is OFF, ω is SP.
 That is:

If the resultant R is too near and the error is positive small, then the linear velocity in the x-axis is off and the angular velocity is small positive. So, based on fuzzification, the process of inputting 63 fuzzy production rules was extracted, as listed in Table 1:

Table 1. System Type-2 Fuzzy Production Rules for Robotino® Guiding.

	θ_eR	TN	VN	MN	NS	Z	SP	MP	VP	TP								
R	V_x	Ω	V_x	ω	V_x	ω	V_x	ω	V_x	ω								
OFF	O	N	O	N	O	N	O	SN	O	Z	O	SP	O	P	O	P	O	P
TN	O	N	O	N	O	N	O	SN	O	Z	O	SP	O	P	O	P	O	P
VN	O	N	O	N	O	N	O	SN	SL	Z	O	SP	O	P	O	P	O	P
N	O	N	O	N	O	N	O	SN	F	Z	O	SP	O	P	O	P	O	P
F	O	N	O	N	O	N	O	SN	F	Z	O	SP	O	P	O	P	O	P
VF	O	N	O	N	O	N	O	SN	F	Z	O	SP	O	P	O	P	O	P
TF	O	N	O	N	O	N	O	SN	F	Z	O	SP	O	P	O	N	O	P

The Output Processing

In this paper, the center-of-sets-type reduction method is used to compute Y_{left} and Y_{right} based on the Karnik–Mendel algorithm (KMA) by applying Equations (1) and (2). Then the defuzzification process is implemented by using the average of Y_{left} and Y_{right} as in Equation (3).

As was described in Section 2, the output crisp value is obtained by using the Sugano action. Thus, for the linear velocity in the x -axis, V_x has three set of actions, “OFF (O), Slow (SL) and Fast (F)”, which are related to speed (0 mm/s, 250 mm/s, and 350 mm/s, respectively), as shown in Figure 8. They were chosen to guarantee enough time to take an action to avoid any detected obstacle, besides maintaining the smooth movement of the Robotino®.

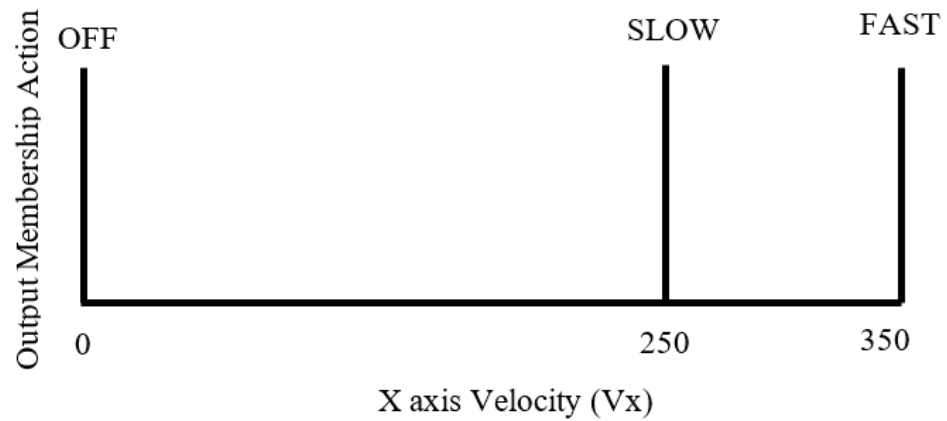


Figure 8. TSK Type-2 Fuzzy Action for X-Axis Linear Velocity.

The fuzzy action of the angular velocity is presented in Figure 9. It has five TSK actions: negative (N), small negative (SN), zero (O), small positive (SP), and positive (P). They are related to (−18 mm/s, −8 mm/s, 0 mm/s, 8 mm/s, and 18mm/s), respectively, as shown in Figure 9. They were chosen to guarantee that Robotino® rotates toward the needed orientation.

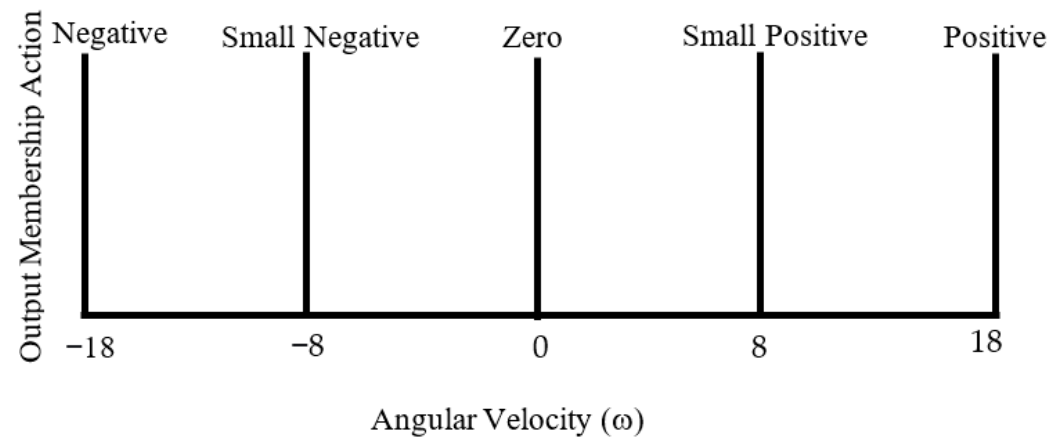


Figure 9. TSK Type-2 Fuzzy Action for Angular Velocity.

2.3. Type-2 Fuzzy Logic Controller Design for Obstacle Avoidance

It is necessary to include obstacle-avoidance capability to ensure that the robot reaches the predefined path safely. Using the available sensor measurements mounted on the Robotino® (right, left, and front) with 40° between each apart, with a range of (4–40) cm. If an obstacle exists at the center, it is recommended that Robotino® moves left, based on the pre-programmed algorithm. Knowing that, a scale conversion of (1 v = 12 cm) is used:

- If output voltage from infrared sensor ≥ 1 V,
- Then analog voltage value is set to (1);
- Otherwise, set it to (0).

The input twenty-seven fuzzy production rules are extracted as listed in Table 2.

Table 2. Obstacle-Avoidance Type-2 Fuzzy Rule.

Rule No.	Situation	Right Sensor	Operator	Front Sensor	Operator	Left Sensor	Consequent	Action
1	If	CL	And	CL	And	CL	Then	L
2	If	CL	And	CL	And	N	Then	L
3	If	CL	And	CL	And	F	Then	L
4	If	CL	And	N	And	CL	Then	None
5	If	CL	And	N	And	N	Then	L
6	If	CL	And	N	And	F	Then	L
7	If	CL	And	F	And	CL	Then	None
8	If	CL	And	F	And	N	Then	L
9	If	CL	And	F	And	F	Then	L
10	If	N	And	CL	And	CL	Then	R
11	If	N	And	CL	And	N	Then	R
12	If	N	And	CL	And	F	Then	L
13	If	N	And	N	And	CL	Then	R
14	If	N	And	N	And	N	Then	R
15	If	N	And	N	And	F	Then	L
16	If	N	And	F	And	CL	Then	R
17	If	N	And	F	And	N	Then	R
18	If	N	And	F	And	F	Then	L
19	If	F	And	CL	And	CL	Then	R
20	If	F	And	CL	And	N	Then	R
21	If	F	And	CL	And	F	Then	R
22	If	F	And	N	And	CL	Then	R
23	If	F	And	N	And	N	Then	R
24	If	F	And	N	And	F	Then	R
25	If	F	And	F	And	CL	Then	R
26	If	F	And	F	And	N	Then	R
27	If	F	And	F	And	F	Then	R

As mentioned above, the center-of-sets type reduction method is used to compute Y_{left} and Y_{right} based on the Karnik–Mendel algorithm (KMA). Then the defuzzification process is implemented by using the average of Y_{left} and Y_{right} , as mentioned in Equations (6) and (7).

A Type-2 fuzzy logic control for avoiding obstacles was added. This controller receives readings from the three IR sensors detecting any possible obstacle existence, along with its distance. The fuzzy set is shown in Figure 10 below:

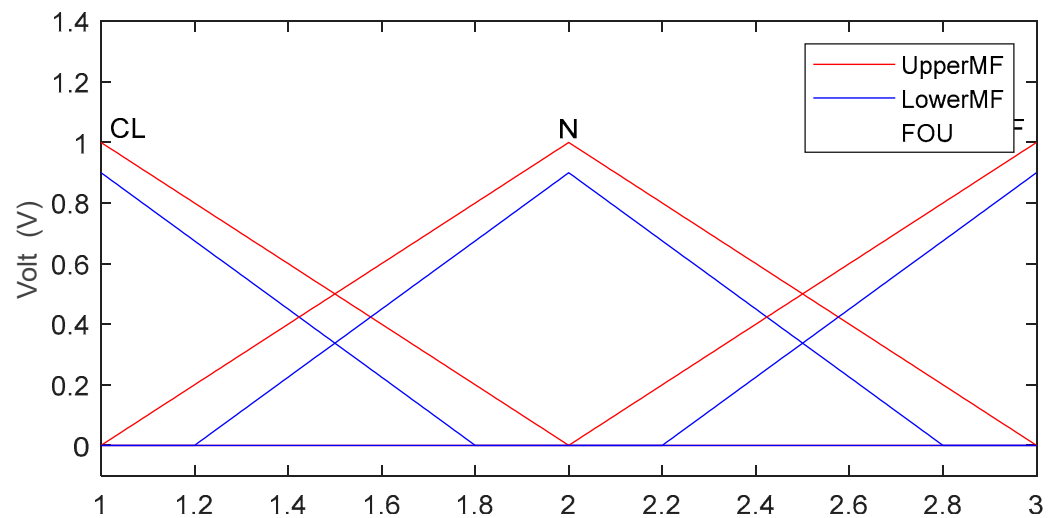


Figure 10. IR Measured Voltage Fuzzification Process.

As described in Section 2, the output crisp value is obtained using Sugano action. Thus, for the linear velocity in Y coordinate V_y , two sets of action “Left (L) and Right(R)”

are related to speed (-250 mm/s and 300 mm/s), respectively, as shown in Figure 11, through many trails at different velocity chosen values:

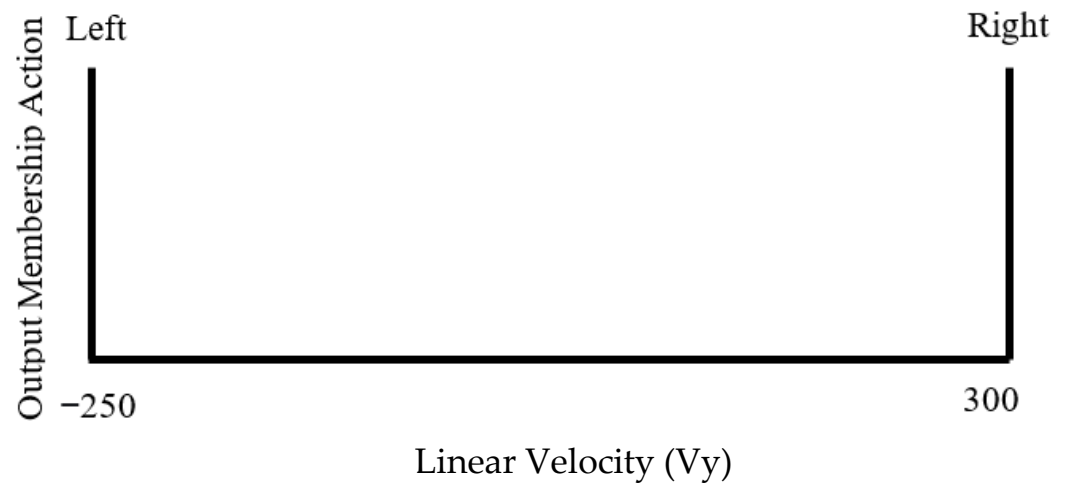


Figure 11. TSK Type-2 Fuzzy Action for Y-Axis Linear Velocity.

2.4. The Overall System Block Diagram

The objective of this research was to design and implement a control algorithm that would enable Robotino[®] to safely reach its predefined goal, collision free. To achieve the objective, the designed control algorithm was split into two main parts. The guiding mechanism that is responsible for driving Robotino[®] toward its destination and the obstacle-avoidance mechanism that makes Robotino[®] move collision free. The overall system block diagram shown in Figure 12 unifies these parts.

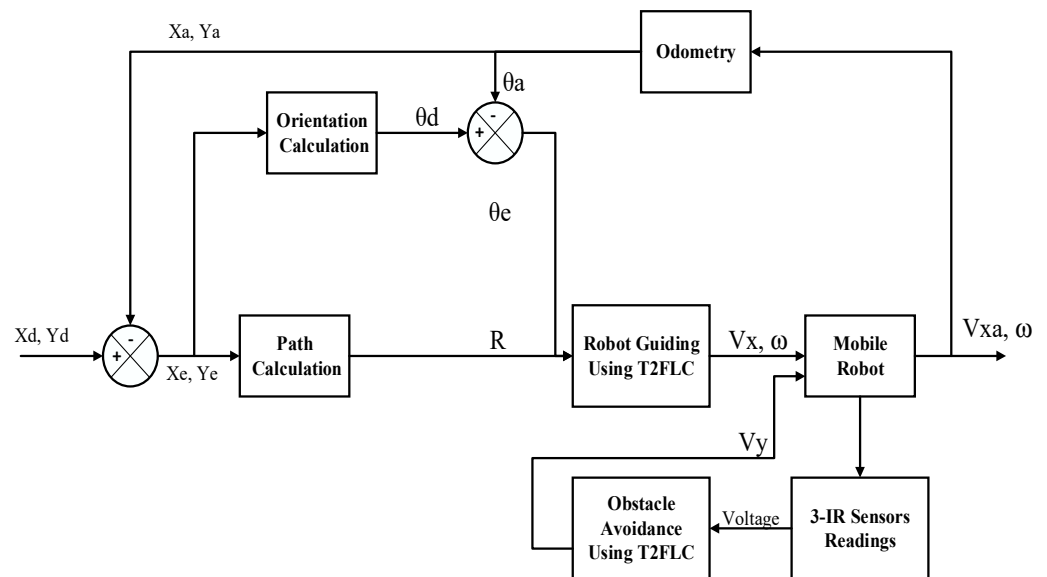


Figure 12. The Overall System Block Diagram.

2.5. MATLAB Interface Block

The available Robotino[®] was designed to use MATLAB 2021b version, with the toolbox related to the Robotino[®] for making use of the interface block elements, as illustrated in Figure 13. The communication block is responsible for setting up wireless communication with Robotino[®] via a suitable IP address. The constant block is specifically used for choosing the desired coordinates (X_d, Y_d). The analog displays show the actual coordinates and orientation, as well. The subsystem block, named the MATLAB interface block, is explained in detail in Figure 14; it consists of a subsystem for three infrared-distance-measured signals.

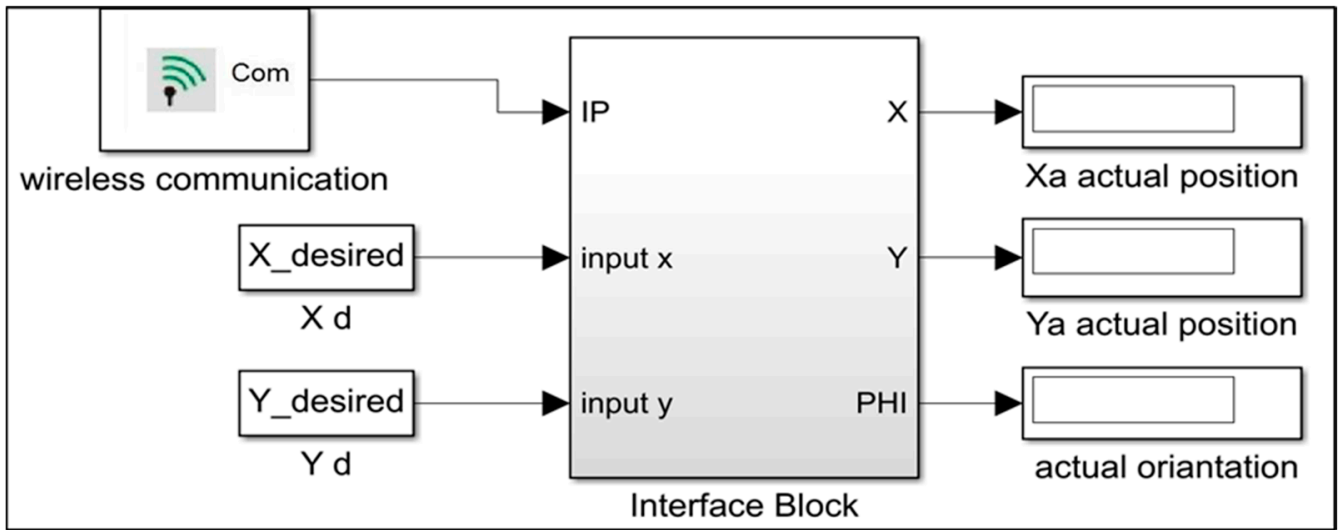


Figure 13. MATLAB Interface Block.

The IP block is utilized for limiting the three infrared-distance-measured voltage values and two subsystems for Robotino® Guiding Type-2 fuzzy logic controller and obstacle avoidance Type-2 fuzzy logic controller, as shown below, in Figures 15 and 16, respectively.

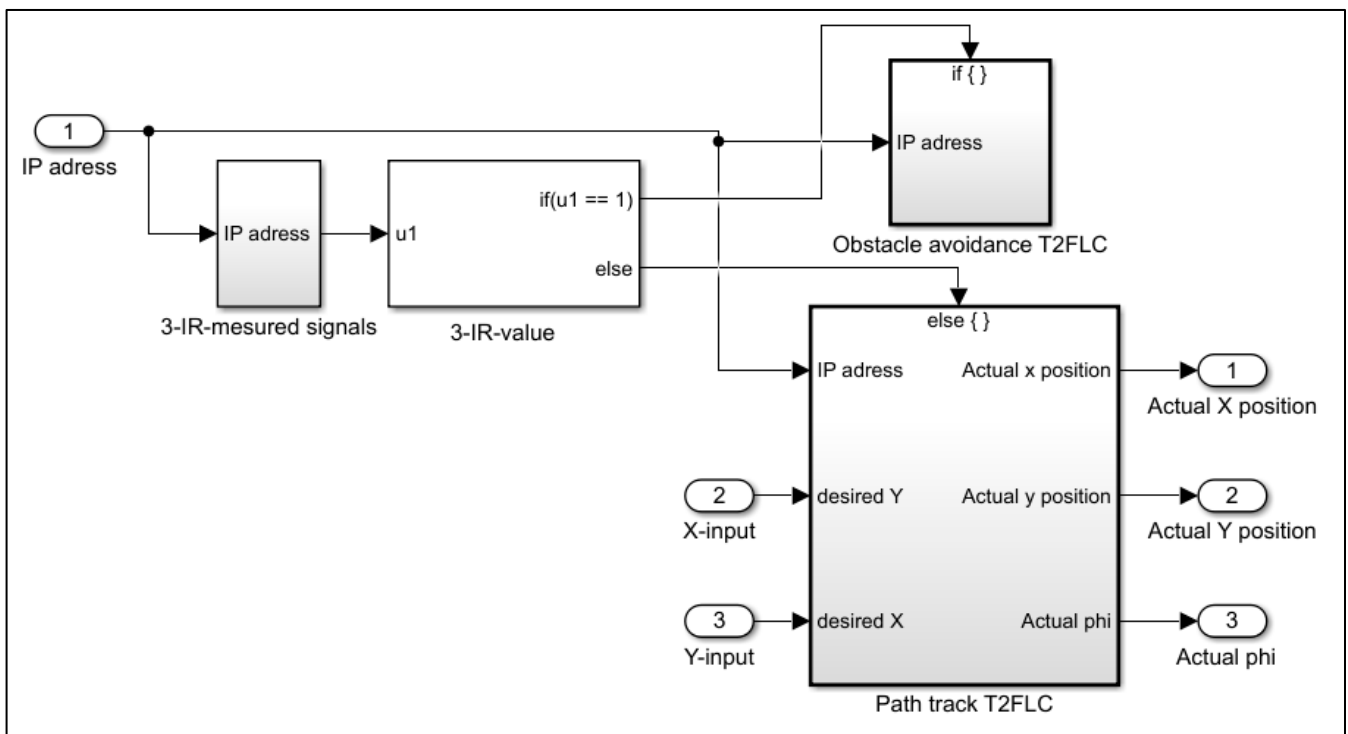


Figure 14. MATLAB Interface Block Elements.

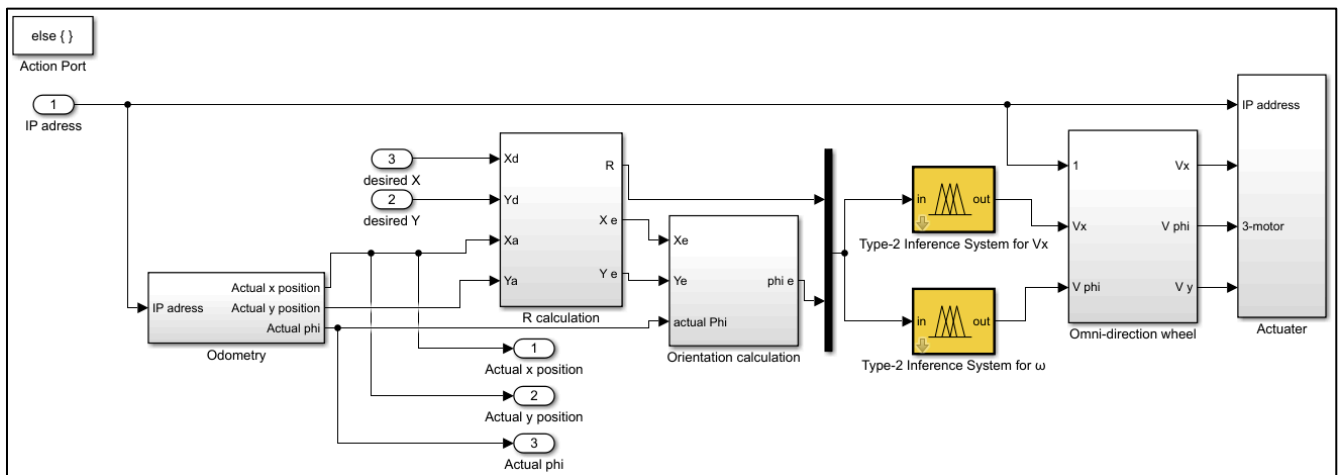


Figure 15. Robotino® Guiding Type-2 Fuzzy Logic Controller Block.

Figure 15 represents the Odometry block, which is responsible for sending the calculated odometric information to Robotino® motors. Two subsystem blocks are used: one for the calculation of the resultant (R) and another one for the orientation (θ) calculation. A Multiplexer is used for selecting input–output crisp variables, the x -axis linear velocity is V_x , and the angular velocity (ω) is used for the Type-2 fuzzy logic controller. The omni-drive object corresponds to the Robotino® omni-drive unit to calculate motors’ speeds in rpm. Finally, a subsystem block consists of the three motors’ actuating signals.

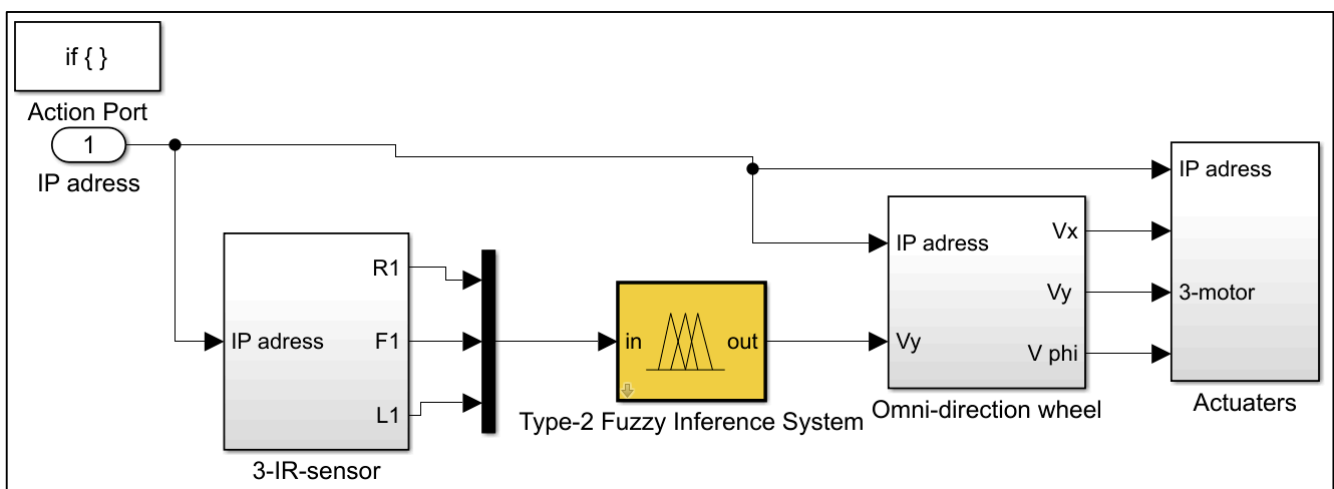


Figure 16. Obstacle Avoidance Type-2 Fuzzy Logic Controller Block.

Figure 16 explains the obstacle-avoidance Type-2 fuzzy logic controller which receives its 3-sensor reading voltages and produces an action of y -axis linear velocity to Robotino®.

2.6. The Control Algorithm Flowchart

The procedural algorithmic flowchart is explained in Figure 17.

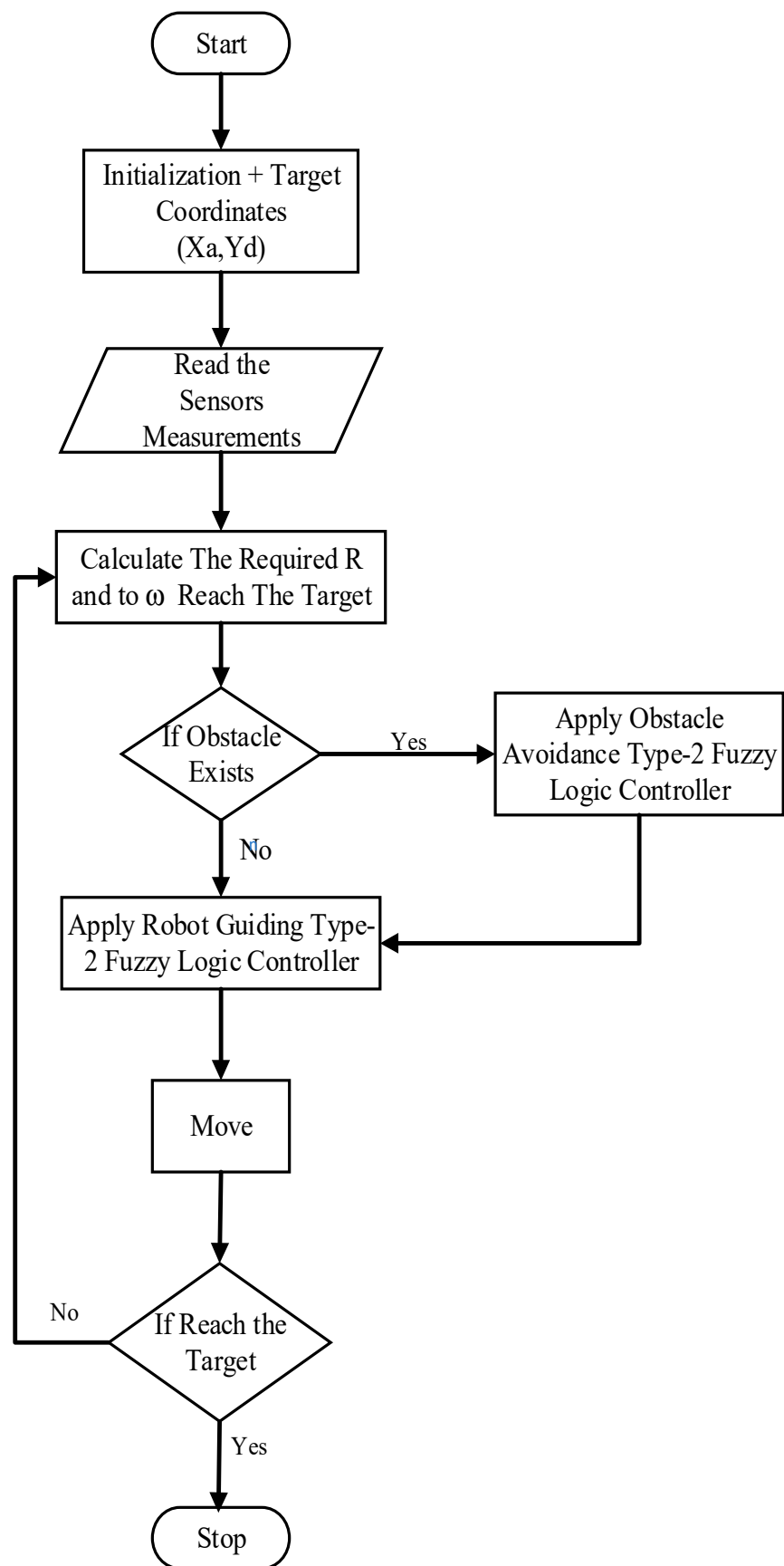


Figure 17. Algorithm Procedure Flowchart.

3. Results

One of the challenges is moving the Robotino[®] safely. Thus, multiple obstacles were put into consideration to test the Type-2 fuzzy logic controller's ability for handling this challenge. It needs more time and energy to avoid an obstacle to keep moving toward its destination point. Thus, IR sensors have the ability to detect such an obstacle distance enough to avoid it completely. Figure 18 shows the behavior of the Robotino[®] when reaching its predefined destination, avoiding such obstacles and following certain steps of movement taken based on two Type-2 fuzzy controllers.

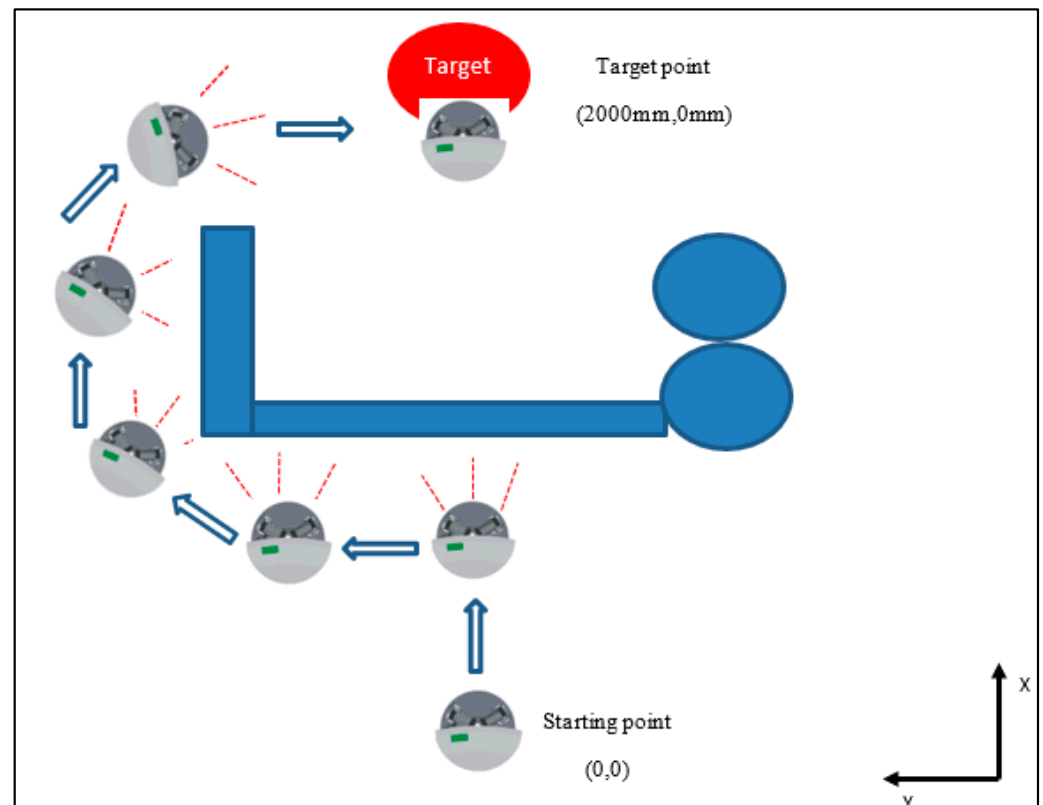


Figure 18. Robotino[®] Response in U-Shape Environment.

The implemented procedure is as below:

Step 1: Robotino[®] determines the starting point coordinates (0, 0, 0); in this case, there is no need to change its orientation.

Step 2: Robotino[®] starts to move forward at X-direction, with no change in Y-direction.

Step 3: Robotino[®] IR sensor detects an obstacle less than 40 cm. Thus, an action is taken to move it to the left, as planned before for priority, i.e., change in Y direction only.

Step 4: When no obstacle is detected, Robotino[®] corrects its rotational angle to move toward the destination from its instant position.

Step 5: Algorithm modifies Robotino's[®] position in the same manner as in Step 3 and Step 4, until it reaches its target with almost zero steady-state error, with a time response as shown in Figure 19, where a 3602-Sample was acquired by using a period of 36 s; therefore, the sampling time was 10 ms.

The infrared sensor readings are shown in Figure 20, whereas the D.C. motors' current consumption is demonstrated in Figure 21. The motors' linear velocities, which clearly reflect the response of Robotino[®] to avoid U-shape obstacles, are shown in Figure 22. This indicates that all motors work to guide Robotino[®] to the right direction to avoid collision.

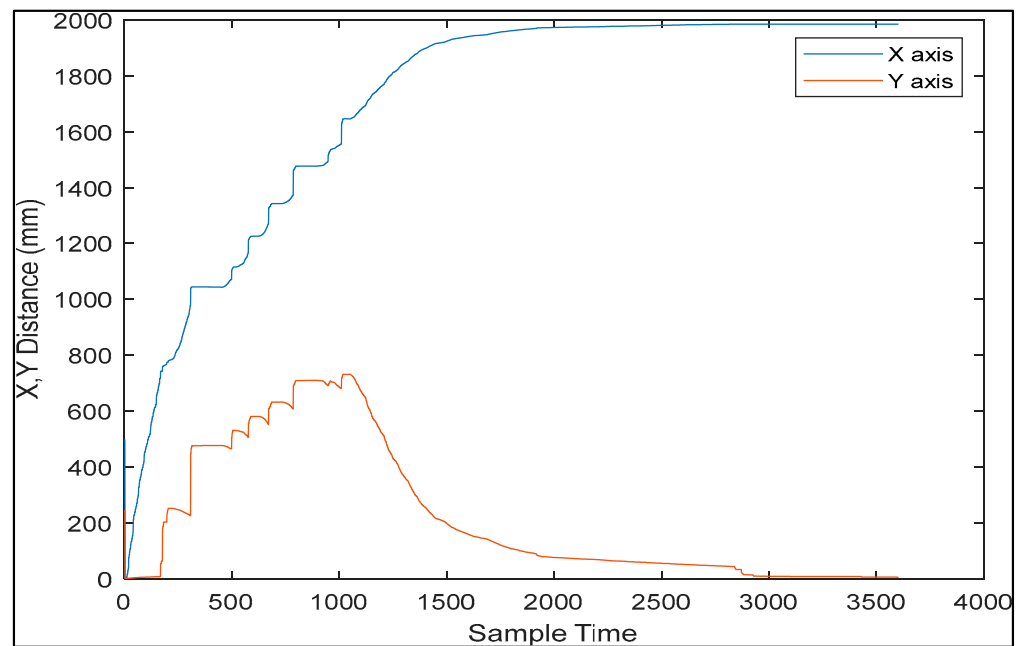


Figure 19. Robotino® Time Response in Obstacle U-Shape Scenario.

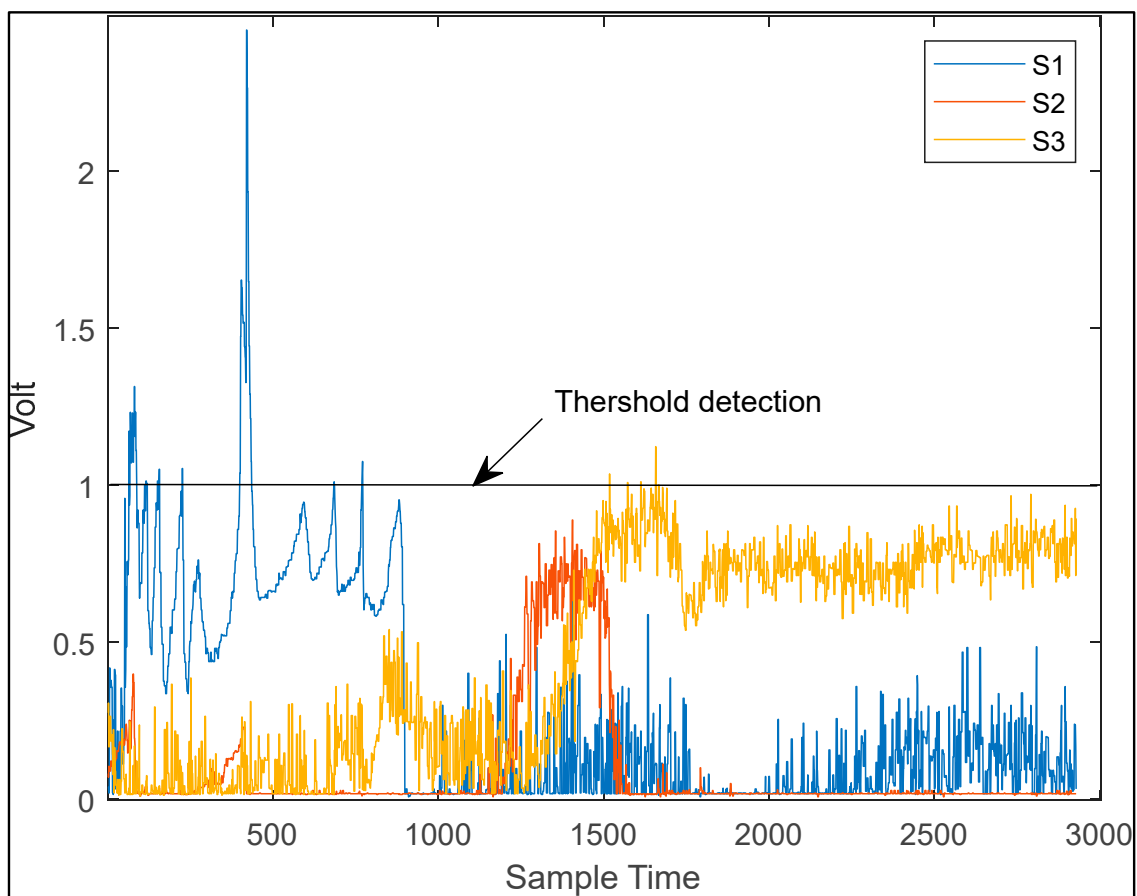


Figure 20. The Three IR Sensors Readings.

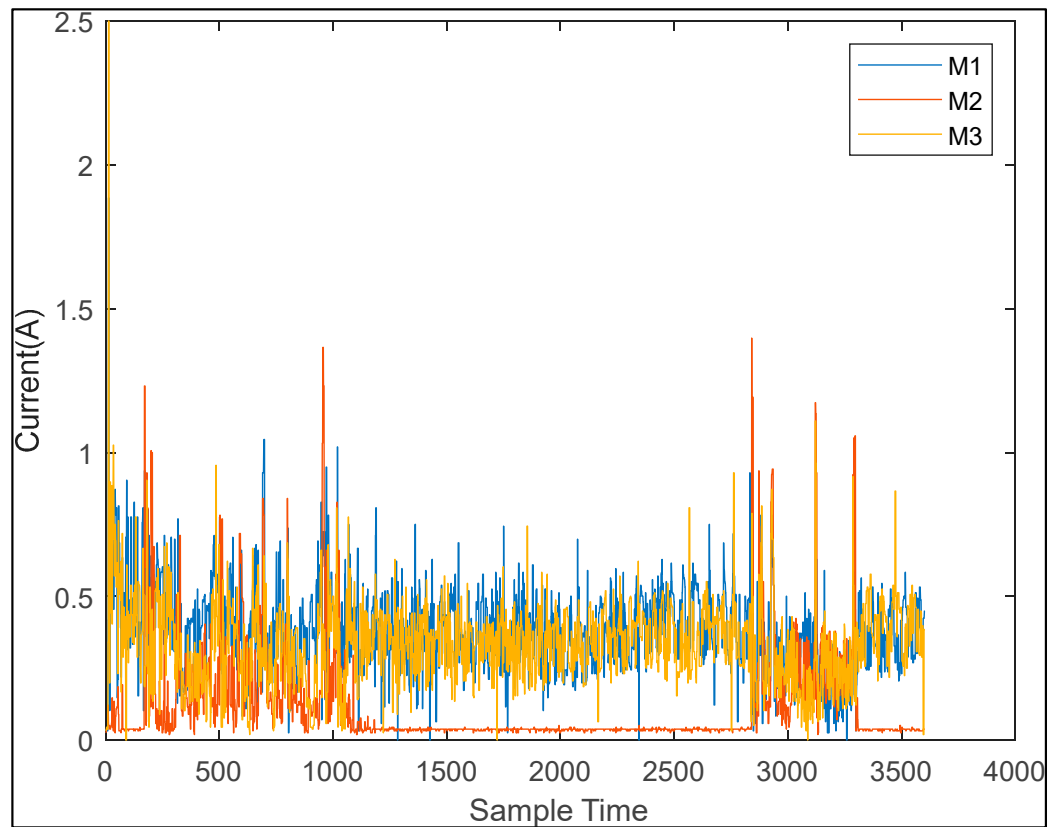


Figure 21. The Current for Three Robotino® D.C. Motors.

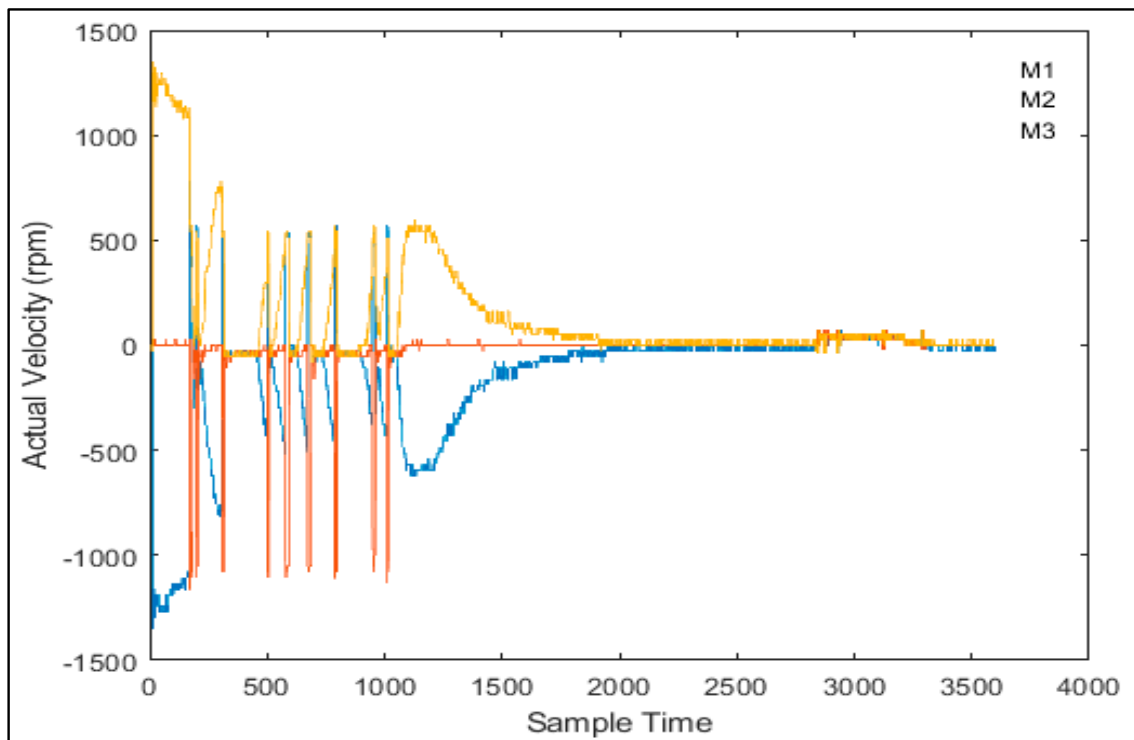


Figure 22. The Actual Linear Velocities for Three D.C. Motors.

- The current of motor No. 2 is almost zero since Robotino® is moving forward; then motors No. 1 and No. 3 worked.

- All motor currents are indicated with a value of about 1.5 amps at the 10th and the 28th second, and this is related to the action taken in Step 3, as described before.
- At the end of the period, all motors were stopped, and zero current values were recorded.

Moreover, many case studies are tested, and some of them are described as below:

3.1. Dynamic Obstacles Scenario

It is proposed that a multi-position dynamic obstacle moves closely toward Robotino[®] in its path at destination (0 cm, 400 cm). The rules in the control algorithm related to the 3-IR readings that are explained in Table 2 were executed effectively, as shown in Figure 23.

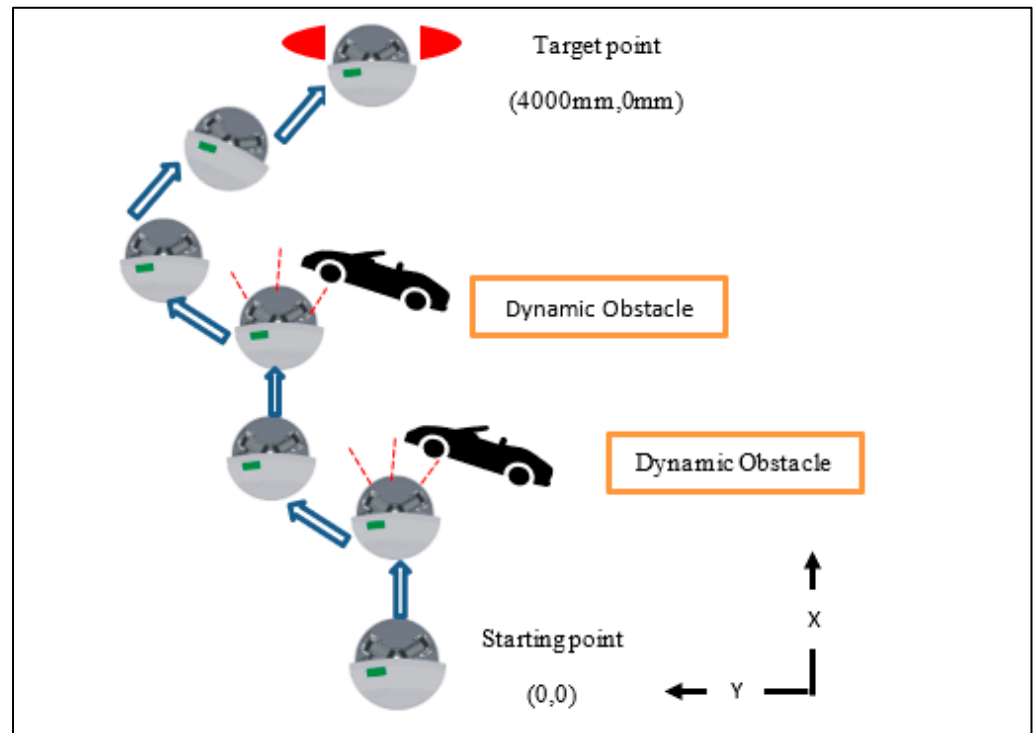


Figure 23. Robotino[®] Response for Dynamic-Obstacles Environment.

Time response is taken to confirm the effectiveness of the proposed control algorithm in the sense of position control. Real readings were collected by the aid of the available Robotino[®] facilities. This brought out the time responses shown in Figures 24 and 25. These responses show how Robotino[®] behaved to avoid the dynamic obstacle, in addition to reaching its predefined target (0, 400 cm), with somehow negligible steady-state errors in both Y and X coordinates, as shown in Table 3 below:

Table 3. Time-Response Specifications for Dynamic Obstacle Scenario.

Coordinate (cm)	Settling Time (Ts), s	Steady State Error (5%)
Yd = 0	-	-
Ya = 3.2	32	0.003
Xd = 400	-	-
Xa = 400.1	35	0.001

The steady-state error performance index shows the accurate obtained position based on the acceptable tolerance of (5%). Note that the target of the controller is used to help Robotino[®] reach its predefined destination effectively, regardless of the time taken.

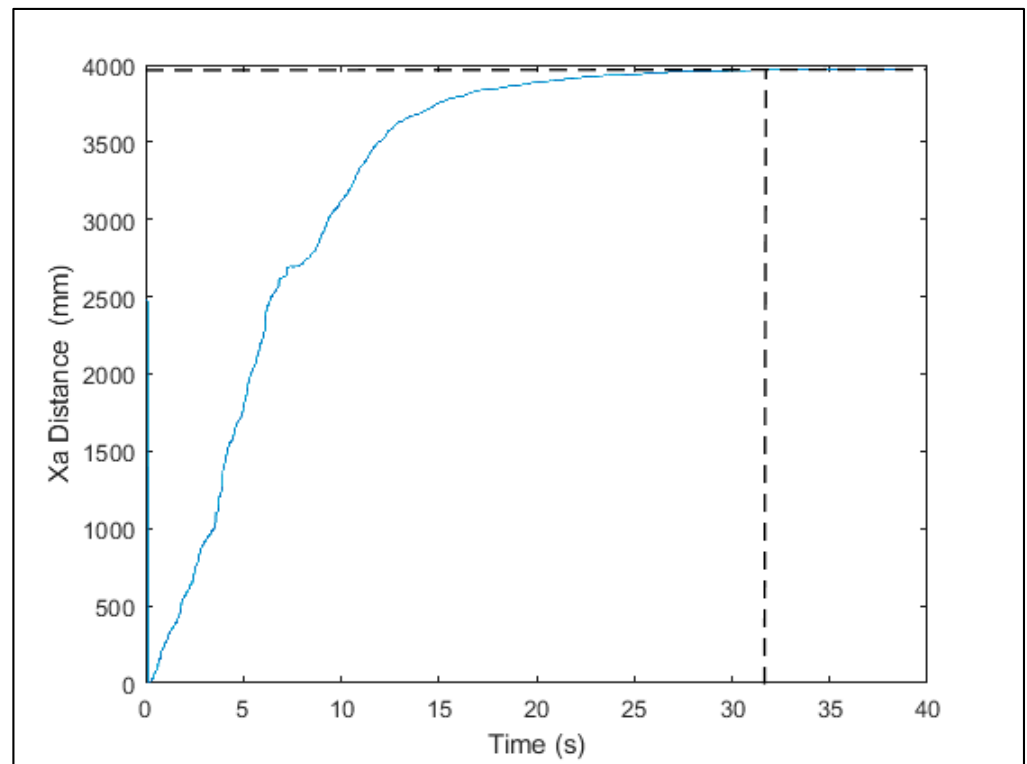


Figure 24. Time Response in X-Coordinate.

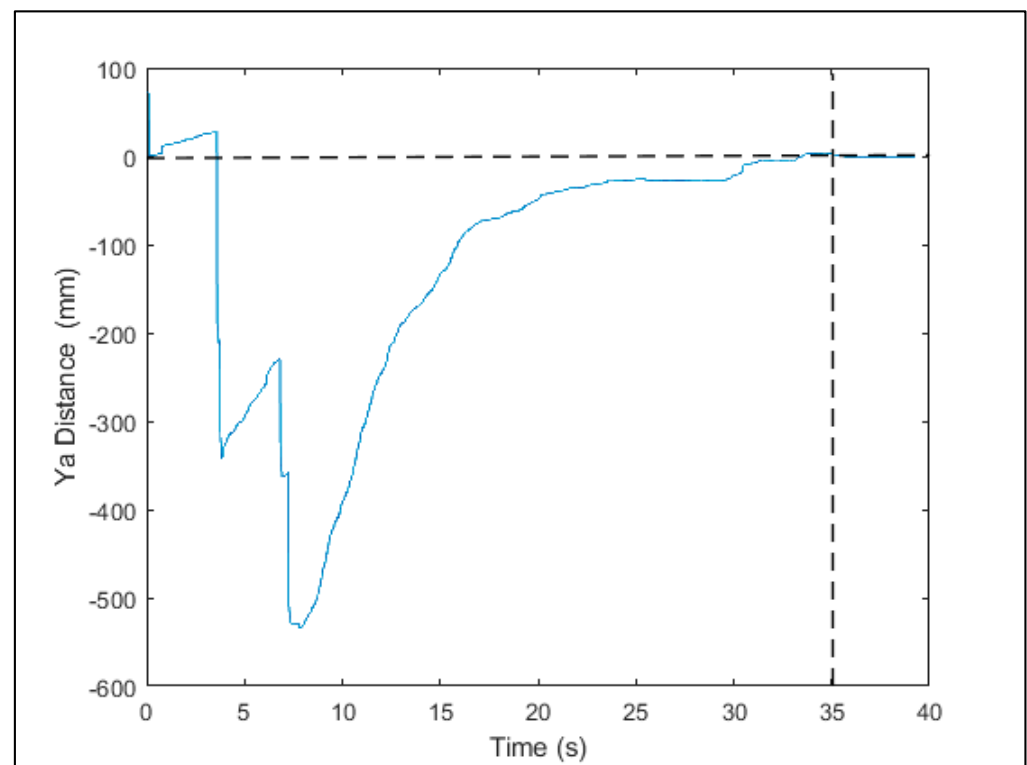


Figure 25. Time Response in Y-Coordinate.

3.2. Comparison Study

A research study presented in Ref. [19] was taken as a comparative study to measure the capabilities of the proposed Type-2 fuzzy logic controllers for the following reasons:

The same model of the Robotino[®] was used. A fuzzy logic controller of 153 fuzzy rules was implemented to control the Robotino[®] path trajectory, while 63 fuzzy production rules were used for the same purpose in this research. The common environment was easy to be implemented, as used in Ref. [19], for accurate environmental comparison. A real-time experiment was achieved with the exact operating conditions as in Reference [19] to reach the desired goal of (3500 mm, 1500 mm), using the underlying Type-2 fuzzy logic controllers. A smooth response was obtained by employing a Type-2 fuzzy logic controller; a time delay of (1257 ms) took place by using the proposed Type-2 fuzzy logic controller with a robust response. Meanwhile, (2280 ms) was taken when using the Type-1 fuzzy logic controller with a slightly slow response. Anyway, the two responses reached the desired destination, with zero steady-state errors, as shown in Figures 26 and 27.

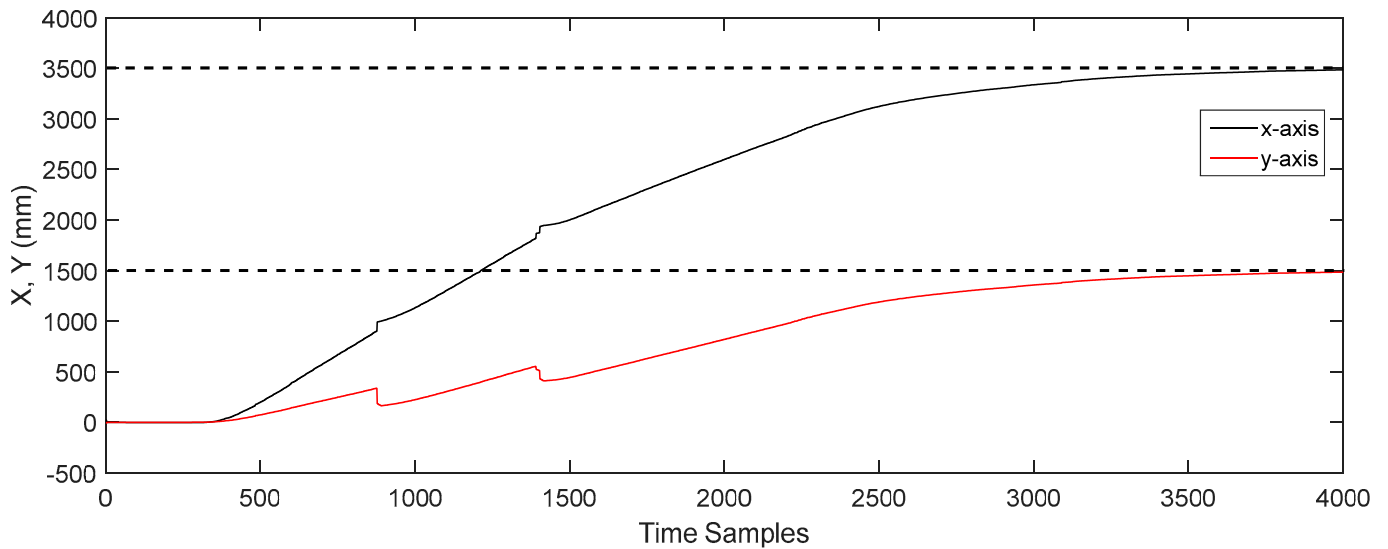


Figure 26. Robotino[®] Time Response Using Type-1 Fuzzy Logic Controller [19].

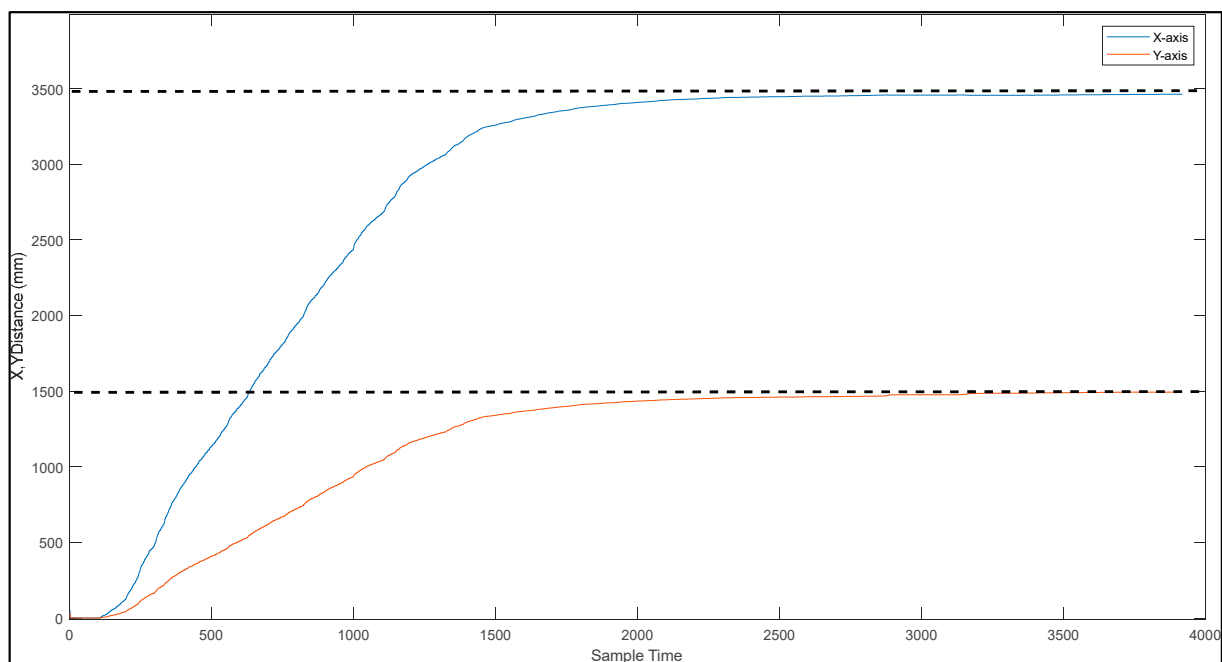


Figure 27. Robotino[®] Time Response Using Type-2 Fuzzy Logic Controller.

The simulation results proved the robust manner of the Type-2 fuzzy logic controllers over the Type-1 fuzzy logic controller when the same operating conditions were used regarding the obstacle-avoidance case.

4. Conclusions

A Type-2 fuzzy logic controller algorithm was proposed to detect and avoid static or dynamic obstacles, using three infrared sensors. The required data for the algorithm from the three sensors are analog voltages which reflect the distance of the obstacle in cm, with a scale of (1 V = 12 cm). The sensors' readings were fed to the obstacle-avoidance Type-2 fuzzy logic controller as crisp voltage inputs that were fuzzified into three fuzzy sets each. Its action based on the TSK Type-2 fuzzy algorithm was used to control the linear Robotino[®] velocity at the y-direction.

A Type-2 Fuzzy logic controller with 63 fuzzy rules was designed to control Robotino[®] guiding. It received two inputs: the resultant distance and orientation errors. So, when they were appropriately controlled, Robotino[®] moved toward its predefined path in a controllable manner. The fuzzy actions based on the TSK Type-2 fuzzy algorithm were used to control the linear velocity along the x-direction, along with the angular velocity.

An overall controller design which is capable of avoiding both static and dynamic obstacles was implemented effectively. The effectiveness of the Type-2 fuzzy controller was tested successfully by using many scenarios in real-time experiments by extracting meaningful information to represent the results to help in the evaluation of the proposed Type-2 fuzzy controller responses.

The designed Type-2 fuzzy logic controller was used in this research as a high-level controller, which showed a high degree of performance in the sense of both speed and steady-state error for both static and dynamic obstacle avoidance, as the experimental result proved. The resultant hardware system is also flexible, with a good dynamic response, which resulted in good accuracy.

There are some suggestions for promising future research using laser sensors. So, Robotino[®] can detect small near obstacles that an infrared sensor cannot because of its features. Moreover, other control strategies could be introduced, such as Neuro-Type-2 fuzzy controllers. This will help to decide which controller is fit to control Robotino[®] in both Robotino[®] guiding and obstacle-avoidance cases, compared with the designed Type-2 fuzzy logic controller responses. It is also worth it to apply the Mamdani Type-2 fuzzy logic algorithm and compare its results with those of the TSK Type-2 fuzzy algorithm applied in this research. The Global Position System (GPS) concept could be introduced for guiding an outdoor mobile robot. A time-of-light camera can be added to create an accurate top-view map.

Author Contributions: Conceptualization, M.A. and M.A.-K.; formal analysis, M.A.; methodology, M.A.-M., M.A., and M.A.-K.; software, M.A.-M.; supervision, M.A. and M.A.-K.; validation, M.A.-M., M.A. and M.A.-K.; writing—original draft, M.A.-M.; writing—review and editing, M.A. and M.A.-K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: This work was achieved with the support of research laboratory equipment's at Mechatronics Engineering Department/Faculty of Engineering/Philadelphia University/Amman/Jordan.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1–22. [[CrossRef](#)]
2. Pandey, A.; Pandey, S.; Parhi, D.R. Mobile robot navigation and obstacle avoidance techniques: A review. *Int. Rob. Auto J.* **2017**, *2*, 00022. [[CrossRef](#)]
3. Mendel, J.; Hagaras, H.; Tan, W.W.; Melek, W.W.; Ying, H. *Introduction to Type-2 Fuzzy Logic Control: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2014.

4. Al-Dahhan, M.R.H.; Ali, M.M. Path tracking control of a mobile robot using fuzzy logic. In Proceedings of the 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD), Leipzig, Germany, 21–24 March 2016; pp. 82–88.
5. Khairudin, M.; Refalda, R.; Yatmono, S.; Pramono, H.S.; Triatmaja Adhy Kurnia Shah, A. The Mobile Robot Control in Obstacle Avoidance Using Fuzzy Logic Controller. *Indones. J. Sci. Technol.* **2020**, *5*, 334–353. [[CrossRef](#)]
6. Abiyev, R.H.; Erin, B.; Denker, A. Navigation of mobile robot using type-2 fuzzy system. In *International Conference on Intelligent Computing*; Springer: Cham, Switzerland, 2017; pp. 15–26.
7. Santiago, C.; Chiu, C.S. Interval Type-2 Fuzzy and PID Dual-Mode Controller for an Autonomous Mobile Robot. In Proceedings of the 2018 International Conference on System Science and Engineering (ICSSE), New Taipei, Taiwan, 28–30 June 2018; pp. 1–6.
8. Azzoug, Y.; Pusca, R.; Sahraoui, M.; Ammar, A.; Romary, R.; Cardoso, A.J.M. A single observer for currents estimation in sensor's fault-tolerant control of induction motor drives. In Proceedings of the 2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID), Elazig, Turkey, 25–27 September 2019; Volume 1, pp. 1–6.
9. Cuevas, F.; Castillo, O.; Cortés-Antonio, P. Design of a Control Strategy Based on Type-2 Fuzzy Logic for Omnidirectional Mobile Robots. *J. Mult.-Valued Log. Soft Comput.* **2021**, *37*, 107–136.
10. Nadour, M.; Boumehraz, M.; Cherroun, L.; Puig Cayuela, V. Hybrid type-2 fuzzy logic obstacle avoidance system based on horn-schunck method. *Electroteh. Electron. Autom.* **2019**, *67*, 45–51.
11. Nyayu, L.H.; Handayani, A.S.; Nurmaini, S.; Yani, I. Application of type-1 and type-2 fuzzy logic controller for the real swarm robot. *ijOE* **2019**, *15*, 83–98.
12. Ayedi, D.; Boujelben, M.; Rekik, C. Hybrid type-2 fuzzy-sliding mode controller for navigation of mobile robot in an environment containing a dynamic target. *J. Robot.* **2018**, *2018*, 8421848. [[CrossRef](#)]
13. Tseng, S.P.; Chen, C.W.; Kuan, T.W.; Hsu, Y.T.; Wang, J.F. Fuzzy Obstacle Avoidance for the Mobile System of Service Robots. *Wirel. Commun. Mob. Comput.* **2020**, *2020*, 8887547. [[CrossRef](#)]
14. Singh, R.; Bera, T.K.; Chatti, N. A real-time obstacle avoidance and path tracking strategy for a mobile robot using machine-learning and vision-based approach. *Simulation* **2022**, *98*, 789–805. [[CrossRef](#)]
15. Khudaverdiyeva, M. Modeling of Mobile Robot with Obstacle Avoidance Using Fuzzy Controller. *Adv. Inf. Syst.* **2022**, *6*, 21–25.
16. Andrade, A.; Silva, T.T.A.; Gomes, I.P.; Gonçalves, L.M.G. Fuzzy System for Obstacle Detection and Avoidance for Mobile Robot. In Proceedings of the 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE), Rio Grande, Brazil, 23–25 October 2019; pp. 180–185. [[CrossRef](#)]
17. Jebur, M.H.; Ali, M.M. Safe navigation and target recognition for a mobile robot using neural networks. In Proceedings of the 2017 14th International Multi-Conference on Systems, Signals & Devices (SSD), Marrakech, Morocco, 28–31 March 2017; pp. 705–712. [[CrossRef](#)]
18. Ali, M.M.; Ali, T.Y. Obstacles avoidance for omnidirectional mobile robot using line trajectory adaptation. In Proceedings of the 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD), Leipzig, Germany, 21–24 March 2016; pp. 96–101.
19. Ali, T.Y.; Ali, M.M. Robotino obstacles avoidance capability using infrared sensors. In Proceedings of the 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT), Amman, Jordan, 3–5 November 2015; pp. 1–6. [[CrossRef](#)]
20. Castillo, O.; Melin, P.; Kacprzyk, J.; Pedrycz, W. Type-2 fuzzy logic: Theory and applications. In Proceedings of the 2007 IEEE International Conference on Granular Computing (GRC 2007), Fremont, CA, USA, 2–4 November 2007; p. 145.