Design and Simulation of a Fuzzy Trajectory Tracking Controller for Linear graph Four Wheel Skid-steer Mobile-Robot with Obstacle Avoidance

Ali Pahlavan[1] , Saeed Khankalantary[*1]

[1]*Department of Electrical Engineering, K.N. Toosi University of Technology, Tehran, Iran*
(s.kalantary@kntu.ac.ir)

Abstract:

This article presents a Fuzzy Trajectory Tracking Controller for a Linear graph model of Four four-wheel skid-steer mobile robots by leveraging a state space model derived from McCormick's work focusing on navigation and obstacle avoidance. The study commences with designing The fuzzy logic controller which is meticulously detailed, focusing on its input parameters, which include metrics like distance to the target, proximity to obstacles, target relative angle, and obstacle relative angle. These inputs guide the controller in making decisions that directly influence the velocities of the Mobile Robot's left and right wheels by adjusting their voltages. Fuzzy controller outputs are voltages of the left and right wheels of the mobile robot. The research methodology encompasses three distinct scenarios, each one challenges the Mobile Robot to navigate towards a target while encountering static and dynamic obstacles with disturbance. The results of these simulations, complete with trajectory plots, angles, velocity profiles, and the distance of the robot to the obstacles and the target, clearly demonstrate the proficiency and robustness of the developed fuzzy logic controller in orchestrating a safe, adaptive, and efficient mobile robot movement.

Keywords:

four-wheel skid-steer mobile robot, fuzzy logic controller, obstacle avoidance, trajectory tracking, disturbance

1. Introduction

Mobile robots are a type of robots that can move from one location to another location. These robots are known as the unmanned vehicles. Four-wheel skid-steer mobile robots are widely used in industrial automation, intelligent inspection, and outdoor exploration [1]. In recent works, various methods of system modeling have been developed. but most of them have a common defect and that is a lack of appropriate focus on a system by mechatronics vision [2]. To solve this problem some methods have been introduced.In particular, In the 1960s, Henry M. Paynter introduced two distinct graphical techniques for modeling dynamic systems: linear graphs (LGs) and bond graphs (BGs). [3]. In 2007 W. Borutzky presented an innovative graphical, computer-aided modeling methodology tailored to the concurrent design of multidisciplinary systems, focusing on engineering systems encompassing mechanical, electrical, hydraulic, or pneumatic components, and considering the intricate interactions of physical effects across diverse energy domains [4].

In [5], the authors focus on modeling and simulating a robotic system with four wheels using bond graph modeling. They introduce a dynamic model that spans different physical domains, and this model is simulated using the BGV_20 toolbox in MATLAB Simulink. To demonstrate the effectiveness of their dynamic modeling technique, they validate their simulated results by comparing them with those obtained using the classical Newton-Euler method. In [6] a comprehensive study focused on the kinematics modeling of a non-holonomic mobile robot with a unique configuration: a four-wheeled design featuring two active wheels and two passive wheels, each driven by independent DC motors.

In [7] a systematic journey was conducted to establish a mathematical model for a 4-wheel skid-steering mobile robot, dissecting it into distinct levels of kinematics, dynamics, and drive. In [8] they present the development of an innovative constrained model predictive control approach tailored for the dynamic path tracking of an off-road mobile robot equipped with a double steering axle. In another article [9] , the authors tackle the challenge of dynamics modeling for a mobile robot equipped with four wheels where the wheels do not steered. and servo-driven propulsion. They consider three drive system configurations, derive a

dynamics model that accounts for various factors like tire-ground contact, friction, and motor characteristics, and emphasize its utility in developing and simulating control systems, including robust and adaptive control designs.

In [10], the authors create a mathematical model for a magnetic-wheeled wall-climbing robot and develop an intelligent trajectory-tracking control algorithm. They optimize the control algorithm using genetic algorithms and demonstrate its effectiveness in simulations, highlighting its advantages over a neural network-based approach. In [11], the authors develop an efficient navigation algorithm called BBNA for four-wheel steering (FWS) mobile robots used in industrial settings. BBNA combines various behaviors, including 'Goal-to-Goal' and 'Obstacle Avoidance,' and introduces a 'Follow-Wall' behavior to reduce chattering. An article [12] details the implementation of a TurtleBot Burger 3 mobile robot in a 3D environment for testing autonomous navigation algorithms. They simplify the robot's mathematical description with a simulator based on its geometry, exploring different trajectory tracking schemes and optimizing path tracking with random obstacles to improve understanding and digital representation of the system.

In [13] , the focus is on addressing the challenges of navigating dynamic and unknown environments with rapidly changing obstacles. The solution involves the development of an Artificial Intelligence System (AIS) that emulates the behavior of Intelligent Autonomous Vehicles (IAVs) like humans. This AIS employs methods based on Artificial Neural Networks (ANNs) and navigation techniques to train datasets and make decisions for obstacle avoidance. The study emphasizes the adaptability of ANNs to handle new constraints and external environmental data, aiming to improve navigation performance in real-life scenarios.The article [14] involves the study and enhancement of the movement control of four-wheeled car robots within a two-dimensional space for path-following tasks. The study compares the performance of P and Fuzzy controllers and suggests the potential for further development in this model and control system based on the results obtained.

In [15], the authors work on obstacle avoidance for mobile robots to address road accidents. They develop a hybrid algorithm combining line following, wall following, and tangent bug strategies, and

propose a fuzzy logic-based obstacle avoidance controller. Both methods are implemented and compared in simulations on a bicycle vehicle model equipped with ultrasonic sensors, aiming to improve safe navigation. In another work [16], the authors discuss the control of DC motors for Wheeled Mobile Robots (WMRs) using a Proportional-Integral-Derivative (PID) controller. They apply the Taguchi method to optimize the PID controller parameters for path tracking, aiming to improve tracking performance in terms of overshoot, stability, and tracking time.

In another work [17], the authors tackle precise control for mobile robots by creating a motion model through mechanism analysis. They introduce a fuzzy PID controller for trajectory tracking, a combination of a PID controller and a fuzzy inference unit, to adjust parameters based on error and error rate. Simulations in Simulink compare the robot's performance using the fuzzy PID controller and a traditional PID controller, showing that the fuzzy PID controller enables faster tracking and reduced overshoot by 40% while maintaining stability, anti-interference capability, and tracking precision.

In [18], the authors introduce a controller employing a Singleton Type-1 Fuzzy Logic System (T1-SFLS) in conjunction with the Fuzzy-WDO (Wind Driven Optimization) algorithm to enable autonomous navigation and collision avoidance for mobile robots in unfamiliar surroundings. They use the WDO algorithm to optimize the fuzzy controller's parameters. Through computer simulations and real-time experiments with the Khepera-III mobile robot, they found that the Fuzzy-WDO approach performs better than the T1-SFLS controller for mobile robot navigation.

In [19], the authors design and implement a controller designed for a four-wheel mobile robot that utilizes fuzzy logic for navigation and obstacle avoidance. They use a single fuzzy logic controller for both navigation and obstacle avoidance in the dynamic model of the robot. The dynamic model of the robot is created using bond graphs and then transformed into a SIMULINK block. The fuzzy controller takes into account input parameters such as distance from the target, angle, and proximity to obstacles, and generates control signals for the left and right wheels. Various simulations are conducted to evaluate the controller's performance, demonstrating its efficiency in minimizing the path traveled in different scenarios.

In [20], the main objective is to design a fuzzy control system for path planning and controlling a moving robot in a social environment with obstacles. The proposed control algorithm established an appropriate path to reach the target without collision with obstacles by receiving the target position frequently. When the obstacles are examined, it is assumed that fixed and moving obstacles have existed in the environment. In another work [21], research explores a four-wheeled omnidirectional robot of the meconium type. The analysis delves into the kinematics and dynamics of the robot, highlighting the impact of parameters and models on equations. Behavior control for the robot is achieved by employing kinematic equations, allowing it to navigate to its intended position even in the presence of obstacles.

In the present study, in methodology firstly in section I the model of mobile robots will be completely explained then the approach of control of this study will be debated. Then the results of simulations will be delivered and finally, with a summary of work in the conclusion part, it will finish. Our innovations in this article are as

follows:

- Making a fuzzy logic control for a linear graph model that has been made for a four-wheel skid-steer mobile robot, has previously been done in McCormick's work [2] but no article works on designing a controller for this linear graph model
- Develop fuzzy rules with aiming target reaching, obstacle avoidance and disturbance
- Using dynamic obstacles in the second and third scenarios to test the ability of fuzzy logic controllers for evasion.

The paper is organized as follows: In Section II, the Model of the mobile robot is explained in detail. Then in Section III, the control of the mobile robot is discussed, followed by a discussion of the Simulation and Results in Section IV. Section V contains simulations supporting the paper's goals. Finally, the concluding remarks are presented in the last section of the Conclusion.

2. Mobile-Robot Model

Linear graph system models offer a visual depiction of a system's physical model and how its components are interconnected. From this graphical representation, one can derive a complete set of differential and algebraic equations that define the system accurately. There is a specific procedure in Linear graph modeling. First, define the system boundary and analyze the physical system to obtain the roles of each component. Mostly they are inputs, outputs, or energy domains. The next step is identifying across-variables that define the linear graph nodes and draw a set of nodes. Then determining appropriate nodes for each lumped element is important. Finally, assigning linear graph signs according to the sign convention have to do.[22]. In Fig. 1, an example of drawing a linear graph for a mechanical system is shown.

The physical mobile robot (ClearPath Husky) is designed, along with a diagram illustrating its subsystems in Fig. 2, The LG model of this robot comprises several subsystems that cover various physical domains and functionalities within the robotic system. These subsystems include the electrical system, which incorporates DC motors powered by a voltage source (battery), with a drivetrain system consisting of front and rear axles, as well as wheels for each front and rear powertrain. Additionally, there are translational and rotational systems, representing the linear and rotational movements of the entire mobile robot. The results of McCormick's work [2] have been used as the model of a four-wheel skid steer mobile robot for this research. The complete model is as Fig. 1. The state space model of this LG model is as follows:

$$\frac{dx}{dt} = A\begin{bmatrix} \omega_{JLW} \\ \omega_{JRW} \\ v_{MH} \\ \omega_{JH} \\ i_{L1} \\ i_{L2} \end{bmatrix} + B\begin{bmatrix} v_{s1} \\ v_{s2} \end{bmatrix} \tag{1}$$

$$y = C\begin{bmatrix} \omega_{JLW} \\ \omega_{JRW} \\ v_{MH} \\ \omega_{JH} \\ i_{L1} \\ i_{L2} \end{bmatrix} + D\begin{bmatrix} v_{s1} \\ v_{s2} \end{bmatrix} \tag{2}$$

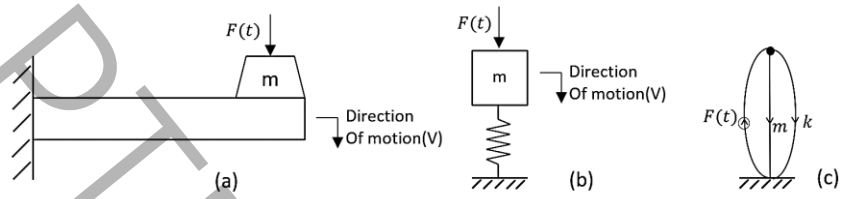

Fig 1. (a) Mechanical system (b) Schematic (c) Linear graph model

$$A = \begin{bmatrix} \frac{-B_{FL}-B_{RL}}{J_L} & 0 & \frac{B_{FL}TF_3-B_{RL}TF_1}{J_L} & \frac{B_{FL}TF_4-B_{RL}TF_2}{J_L} & \frac{T_{ML}}{J_L} & 0 \\ 0 & \frac{-B_{FR}-B_{RR}}{J_R} & \frac{B_{FR}TF_5-B_{RR}TF_7}{J_R} & \frac{B_{FR}TF_6-B_{RR}TF_8}{J_R} & 0 & \frac{T_{MR}}{J_R} \\ \frac{B_{FL}TF_3+B_{RL}TF_1}{M_H} & \frac{B_{FR}TF_5+B_{RR}TF_7}{M_H} & \frac{-B_{RL}TF_1^2-B_{FL}TF_3^2-B_{FR}TF_5^2-B_{RR}TF_7^2}{M_H} & \frac{-B_{FL}TF_3TF_4-B_{FR}TF_5TF_6-B_{RL}TF_1TF_2-B_{RR}TF_7TF_8}{M_H} & 0 & 0 \\ \frac{B_{FL}TF_4+B_{RL}TF_2}{J_H} & \frac{B_{FR}TF_6+B_{RR}TF_8}{J_H} & \frac{-B_{FL}TF_3TF_4-B_{FR}TF_5TF_6-B_{RL}TF_1TF_2-B_{RR}TF_7TF_8}{J_H} & \frac{-B_{RL}TF_2^2-B_{FL}TF_4^2-B_{FR}TF_6^2-B_{RR}TF_8^2-B_H}{J_H} & 0 & 0 \\ -\frac{T_{ML}}{L_1} & 0 & 0 & 0 & -\frac{R_1}{L_1} & 0 \\ 0 & -\frac{T_{MR}}{L_2} & 0 & 0 & 0 & -\frac{R_2}{L_2} \end{bmatrix} \tag{3}$$

$$B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{L_1} & 0 \\ 0 & \frac{1}{L_2} \end{bmatrix} \tag{4}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \tag{5}$$

$$D = [0]_{4\times 2} \tag{6}$$

Eqs. 1,2 represent the dynamics of the system in state space model of the system. The robot's dynamic parameters include both known and unknown values essential for simulating the Husky robot's behavior. Known parameters, such as motor resistance (R_1, R_2), motor inductance (L_1, L_2), motor torque constant (k_t), gear ratio (GR), robot mass (M_{Husky}), and the inertia of the wheels and robot ($J_{LW}, J_{RW}, J_{Husky}$), are obtained from system documentation. Unknown parameters, such as drivetrain damping coefficients ($B_{RL}, B_{FL}, B_{FR}, B_{RR}$) and rotational damping (B_{Husky}), are estimated using algorithms like genetic algorithms (GA) for more accurate simulations. Key parameters also include r_w (wheel radius), a and b (distances from the robot's center of gravity to the front and rear axles), and c (half of the robot's width), which influence the robot's speed, stability, and movement control. Additionally, the rotational speeds of the left and right wheels ($\omega_{JLW}, \omega_{JRW}$) are crucial for calculating the robot's linear and rotational velocities (v_{MH}, ω_{JH}), while (i_L) represents the current in the inductive element, influencing the robot's electrical response.

These parameters together model the robot's dynamics across both mechanical and electrical domains. In addition to these, the system also considers the velocities of the left and right wheels, V_{S1} and V_{S2} which are related to the linear velocities of the wheels and affect the robot's movement and trajectory. These parameters, together with the motor currents (i_{L1}, i_{L2}), contribute to the overall dynamic model that governs the robot's behavior in various operating conditions. Eqs. 3-6 are weight matrices of State space equations. Diagram illustrating a mobile robot navigating towards a target with a single obstacle, alongside the displayed parameters. In Fig. 3, (x_r, y_r) is The present location of the mobile robot's center of gravity. The target position is represented by (x_t, y_t), and the distance from the robot to the target is indicated by d_t , which is also obtained in Eq. 7 using the Euclidean distance formula between two points.

The robot's orientation angle (heading angle) is symbolized by θ_r . The velocity of the right and left wheels is denotedted by V_r and V_l , respectively. The width of the mobile robot is indicated by W. ϕ_t represents the angle between the target and the robot's heading angle, which is obtained using Eq. ١٠. ϕ_0 represents the angle between the obstacle and the robot's heading angle, which is obtained using Eq. ٩. The fuzzy logic controller's output is supplied as voltage to the right and left wheels of the mobile robot. Notably, in the scenarios discussed in the Results and Simulation section, the system consists of a target and obstacles, which may be static or dynamic. Utilizing the robot's radar and ultrasonic sensors, the positions of the target and obstacles are detected, while the robot's position and heading angle are internally known. This data enables the calculation of the Euclidean distance to both the target and the obstacles. Similarly, the robot's distance to each obstacle (d_o) is computed using the Euclidean distance, just like its distance to the target, as obtained in Eq. 8. The angles between the robot and the target (θ_t) , as well as between the robot and each obstacle (θ_o) , are also calculated using the inverse tangent formula, as shown in Eqs. 1١,1٢.

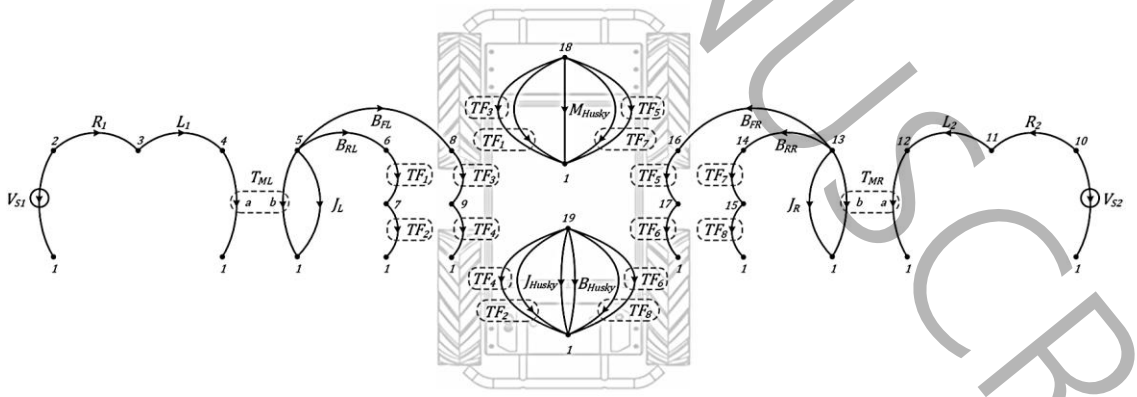

Fig 2. The comprehensive linear graph (LG) representation of the ClearPath Husky robot.[2]

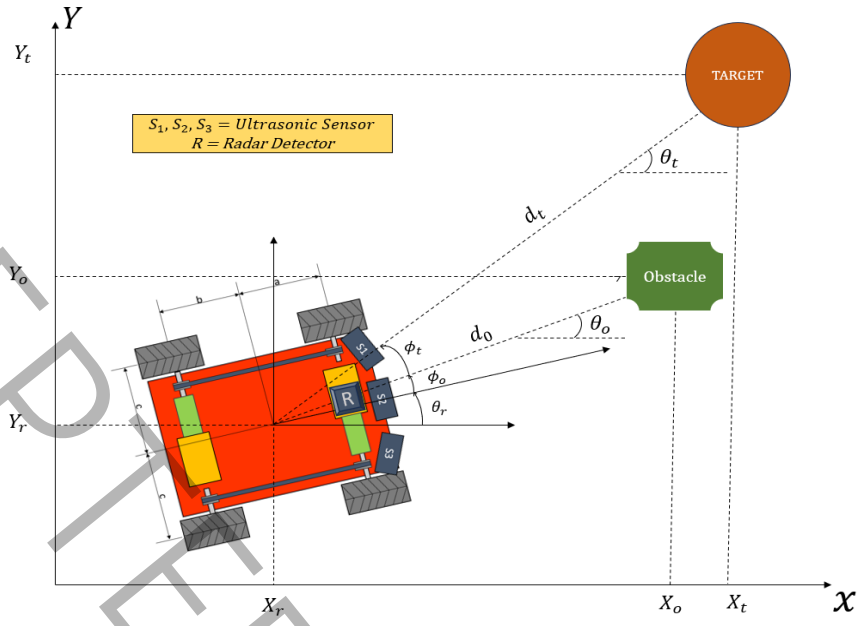

Fig 3. Schematic diagram of four-wheel Skid-Steer Mobile Robot along target with one obstacle

$$d_t = \sqrt{(X_t - X_r)^2 + (Y_t - Y_r)^2} \tag{7}$$

$$d_o = \sqrt{(X_o - X_r)^2 + (Y_o - Y_r)^2} \tag{8}$$

$$\phi_o = \theta_0 - \theta_r \tag{9}$$

$$\phi_t = \theta_t - \theta_r \tag{10}$$

Where:

$$\theta_t = \tan^{-1}\left(\frac{Y_t - Y_r}{X_t - X_r}\right) \tag{11}$$

$$\theta_O = \tan^{-1}\left(\frac{Y_O - Y_r}{X_O - X_r}\right) \tag{12}$$

3. CONTROL ALGORITHM

The fuzzy logic control is a range-to-point or range-to-range control. FLCs are multivalued rationality based and include approximation instead of defined and accurate values. These controllers are highly suitable for multivariable and inconsistent processes, where the exact measurement of input and its effect is difficult. FLCs may have single or multiple inputs and outputs depending upon the application. Fuzzy sets and fuzzy reasoning are used in an FLC for its design and execution. The design of an FLC includes "if … then …" rules, which are based on available data history and/or expert knowledge. These rules are a linguistic description of systems' behavior and characteristics with their imprecise numeric value. [23]

A navigation and obstacle avoidance model based on fuzzy logic has been developed specifically for regulating the motor speed of a four-wheel Skid-Steer Mobile Robot. The complete model of a fuzzy logic controller for this robot is disclosed in Fig. 4. The membership functions employed in this controller are a blend of triangular, Linear S-shaped, Linear Z-shaped, and Gaussian functions. The ranges of each membership function of the fuzzy sets used in the controller vary, and their selection has relied on trial-and-error experience, as illustrated in Fig. 5. The inputs to the fuzzy logic controller include the distance from the target, the distance from obstacles, the angle of the target, and the angle of obstacles. The outputs of the fuzzy logic controller are the velocities of the right (V_{S1}) and left (V_{S2}) wheels of the mobile robot. Fig. 5 illustrates the block diagram of the navigation and obstacle avoidance system for the model of the four-wheel skid-steer mobile robot.

The inputs of fuzzy logic controller are shown in Figs. 6-9. Distance from target which is shown in Fig. 6 has 5 membership functions consisting VS, S, M, L, VL .the range of distance from target is from -1.13 to 30 meters. Distance from other obstacle which is shown in Fig. 7 has 3 membership functions (far,m medium near and the range is 0 to 3 meters. a single target (ϕ_t) which is shown in Fig. 8 has 5 membership functions (VN,N,Z,P,VP) and its range is $(-2\pi, 2\pi)$ radian. angle of obstacle (ϕ_0) which is disclosed in Fig. 9 has 5 membership functions (VN,N,Z,P,VP) and its range is from $-\pi \; to \; \pi$ radian. the outputs of the fuzzy logic

controller are shown in Fig. 10 and 11. The velocities of the left and right wheels serve as outputs, ranging from -0.28 volts to 14 volts.

Three ultrasonic/range sensors (US-100) are utilized for obstacle detection. so from the data obtained from them, the distance from the robot to the obstacles will be calculated (when a distance of the robot with an obstacle is in range of 0 to 3 meters, for more distance, it is 'none'). as it is clear that, the maximum measurement range of these ultrasonics is 4 meters so we consider this on a range of some of our inputs in the fuzzy logic controller. In addition, the mobile robot is equipped with Radar which gives us the distance to the target.we assume the HB100 Microwave Doppler Radar Detector is a radar sensor.

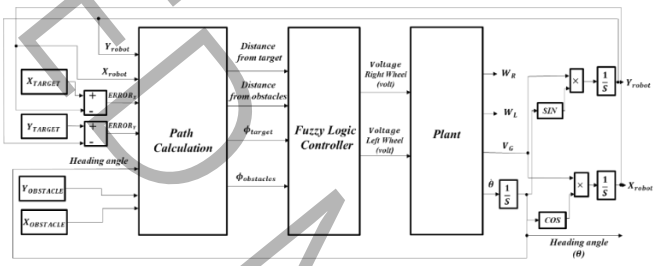

Fig 4. Simulink model for controlling a four-wheel Skid-Steer Mobile Robot

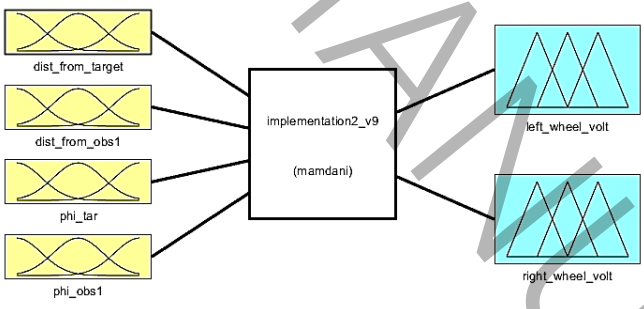

Fig 5. Fuzzy Logic Controller

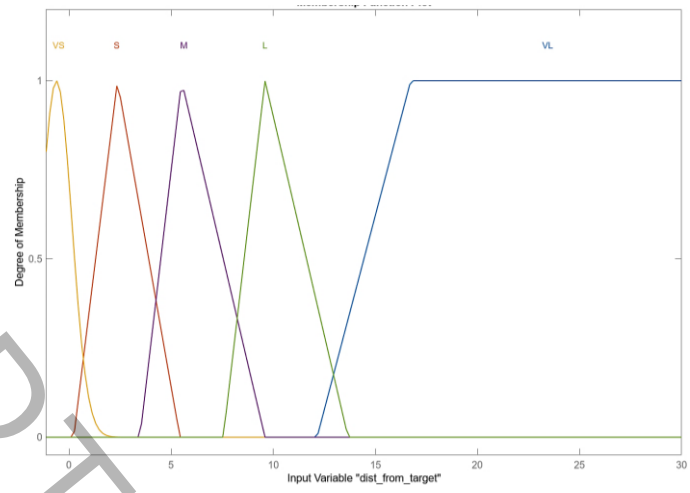

Fig 6. Distance from a target (Input membership function)

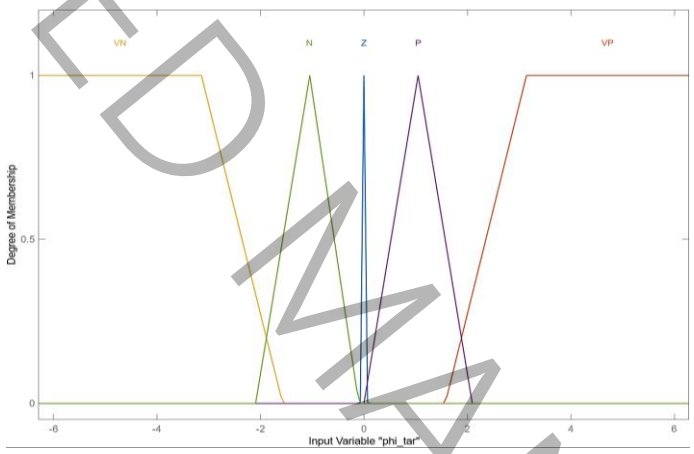

Fig 7. Angle of a target (Input membership function)

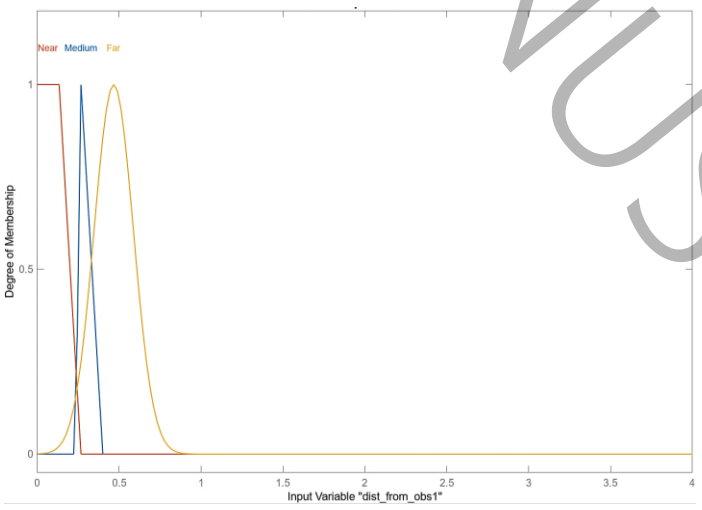

Fig 8. Distance from obstacle (Input membership function)

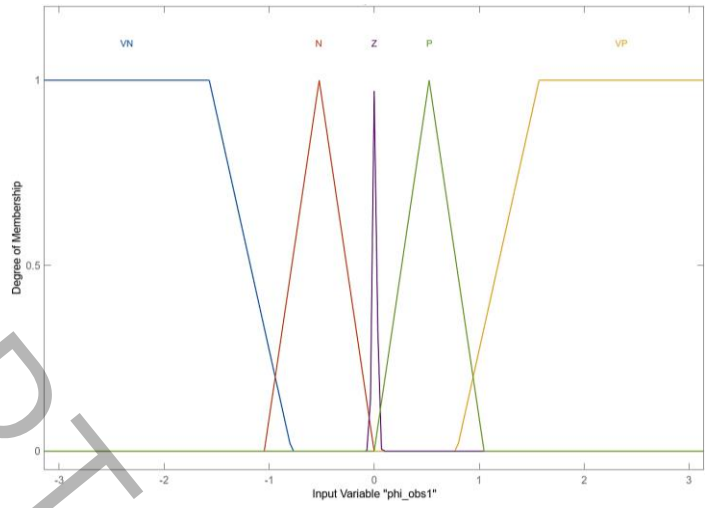

Fig 9. Angle of obstacle (Input membership function)

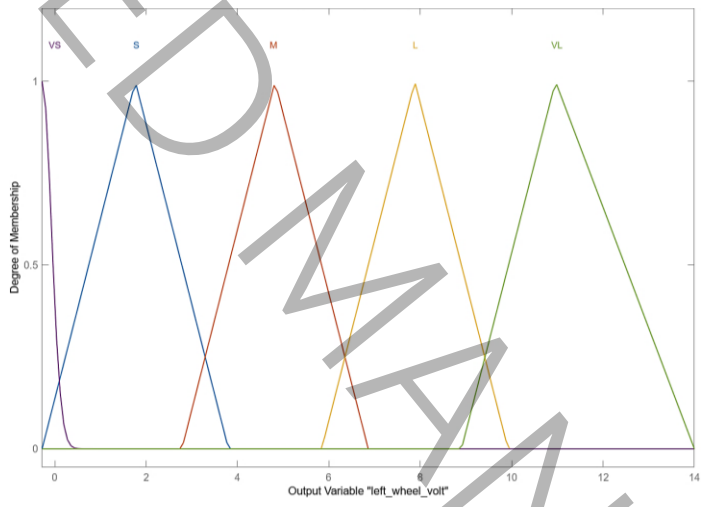

Fig 10. Left wheel velocity (Output membership function)

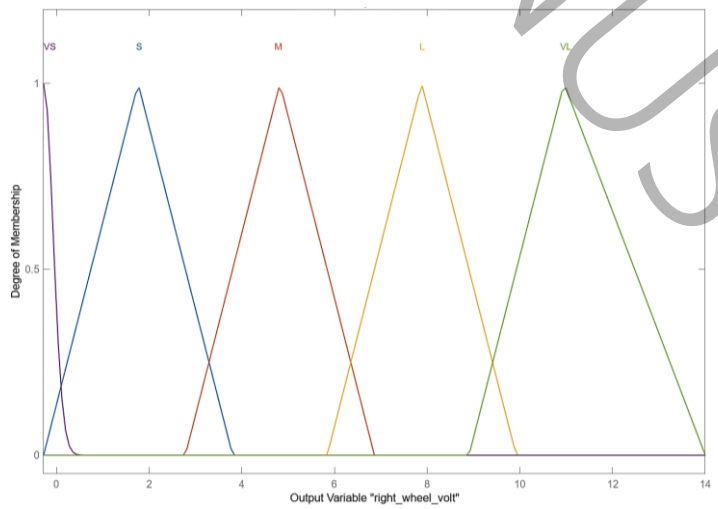

Fig 11. Right wheel velocity (Output membership function)

The essential aspect of a fuzzy controller is its collection of linguistic rules. Frequently, it is simple to translate an expert's knowledge and experience into these rules. In certain scenarios, these rules can be established through trial-and-error techniques. However, there is a key approach to determining these rules. In general control of the mobile robot can divide into 2 parts. First, when the mobile robot is far away from the obstacle (more than 3 meters) and second when a mobile robot is near to the obstacle. So separating these states can help to have the best estimation while choosing the wheel speeds(outputs) for each rule and input.

Generally, when ϕ_t is positive, it means that mobile robot is on the left side of the target so the right wheel should have more speed than the left wheel therefore mobile robot can return to $\phi_t = 0$. In this position, heading angle of robot and line that represent the angle with the target are colinear and this disclose that robot is going toward the target (all these descriptions are reversed when a mobile robot is on the right side of the target). fuzzy logic controller's rules are disclosed in Table. I. Each row in the table represents a rule that governs the robot's behavior based on its distance from the target and obstacles, as well as the angular direction of the target and obstacle. These rules are applied to adjust the velocities of the left and right wheels in order to achieve the desired navigation and obstacle avoidance.

Table I. Rules of Fuzzy Logic Controller

NO	DISTANCE FROM TARGET	DISTANCE FROM OBSTACLE	PHI TARGET	PHI OBSTACLE	LEFT WHEEL VELOCITY	RIGHT WHEEL VELOCITY	WEIGHT
1	VL	NONE	VN	NONE	L	VS	1
2	VL	NONE	N	NONE	M	VS	1
3	VL	NONE	Z	NONE	VL	VL	1
4	VL	NONE	P	NONE	VS	M	1
5	VL	NONE	VP	NONE	VS	L	1
6	L	NONE	VN	NONE	L	VS	1
7	L	NONE	N	NONE	M	VS	1
8	L	NONE	Z	NONE	L	L	1
9	L	NONE	P	NONE	VS	M	1
10	L	NONE	VP	NONE	VS	L	1
11	M	NONE	VN	NONE	M	VS	1
12	M	NONE	N	NONE	S	VS	1
13	M	NONE	Z	NONE	M	M	1
14	M	NONE	P	NONE	VS	S	1
15	M	NONE	VP	NONE	VS	M	1

16	S	NONE	VN	NONE	M	VS	1
17	S	NONE	N	NONE	S	VS	1
18	S	NONE	Z	NONE	S	S	1
19	S	NONE	P	NONE	VS	S	1
20	S	NONE	VP	NONE	VS	M	1
21	VS	NONE	VN	NONE	S	VS	1
22	VS	NONE	N	NONE	S	VS	1
23	VS	NONE	Z	NONE	VS	VS	1
24	VS	NONE	P	NONE	VS	S	1
25	VS	NONE	VP	NONE	VS	S	1
26	NONE	NEAR	NONE	VN	S	S	1
27	NONE	NEAR	NONE	N	VS	S	1
28	NONE	NEAR	NONE	Z	VS	M	1
29	NONE	NEAR	NONE	P	M	VS	1
30	NONE	NEAR	NONE	VP	S	S	1
31	NONE	MEDIUM	NONE	VN	M	M	1
32	NONE	MEDIUM	NONE	N	VS	S	1
33	NONE	MEDIUM	NONE	Z	VS	M	1
34	NONE	MEDIUM	NONE	P	M	VS	1
35	NONE	MEDIUM	NONE	VP	M	M	1
36	NONE	FAR	NONE	VN	L	L	1
37	NONE	FAR	NONE	N	VS	S	1
38	NONE	FAR	NONE	Z	VS	M	1
39	NONE	FAR	NONE	P	S	VS	1
40	NONE	FAR	NONE	VP	L	L	1

4. SIMULATION AND RESULTS

This section examines two scenarios about the Mobile Robot, obstacle, and target have considered, the first scenario is Moving the Mobile Robot toward a target with 3 static obstacles and the second scenario is Moving toward a target with 1 dynamic and 1 static obstacle. Also, constant values used in the simulation of the Mobile Robot are shown in Table III. The parameters of the simulation are as Table II,III.

Table II. State-space parameters [2]

Description	***Parameter***	***Value***	***Units***
Voltage Inputs	V_{s1}, V_{s2}	± 24	V
Internal Motor Resistance	R_1, R_2	0.46	Ω
Internal Motor Inductance	L_1, L_2	0.22	mH
Motor Torque Constant	k_t	0.044488	N.m/A

Gear Ratio	GR	78:71:1	Gear Ratio
Motor Transformer Ratio	T_{ML}, T_{MR}	$k_t \times GR$	N.m/A
Drivetrain Inertia	J_{LW}, J_{RW}	0.08	kg / m^2
Drivetrain Damping	$B_{RL,FL,FR,RR}$	Unknown	rad/(N.m.s)
Power Conversion	TF_{odd}	$\frac{1}{rw}$	– –
Transformer Ratios	TF_{even}	$\pm\cos(\theta_{\omega_i}).r_{c_i}.\frac{1}{rw}$	– –
Husky Mass	M_{Husky}	48.39	kg
Husky Rotational Damping	B_{Husky}	Unknown	rad/(N.m.s)
Husky Inertia	J_{Husky}	3.0556	kg / m^2
Radius of wheel	rw	0.18	m
distance of the front axle from robot C.G.	a	0.270	m
distance of rear axle from robot C.G	b	0.270	m
Half of the width of the robot	c	0.8961	m

Table III. Variables adapted by the ga and the resulting values [2]

Variable	$B_{RL,FL,FR,RR}$	B_{Husky}	c
Upper Bounds	1	1	0.75
Lower Bounds	100	100	1.00
Results	1.3016	12.8650	0.8961

In all scenarios, a safe distance of 25 centimeters from obstacles has been considered.

4-1- The first scenario: Robot reaching its target with three stationary obstacles.

In this section, we have three obstacles in front of the robot. The coordinates of the obstacles are (4, 4), (5.7, 5), and (7.5, 6.4). The target coordinates are (8, 8). Fig. 12 illustrates the mobile robot starting from the origin. It successfully navigates the first obstacle located at coordinates (4, 4), bypassing it at a distance of 46 centimeters. The second obstacle positioned at coordinates (5.7, 5) is slightly more challenging to pass but still maintains a safe distance, circumventing it from 29 centimeters away. The third obstacle at coordinates (7.5, 6.4) poses a significant challenge to pass without collision, yet the fuzzy algorithm manages

to successfully navigate around it with excellent maneuverability. Subsequently, it efficiently takes the shortest path towards the target and swiftly reaches it.

Fig. 12 demonstrates that the mobile robot successfully navigates past obstacles without any collisions. This claim can be substantiated by examining the plot in Fig. 13, showing that none of the obstacles have collided with the mobile robot. In Fig. 14, the first plot is the relative angle of the target with the robot. After several oscillations, it finally reaches zero at the end of the simulation, and there is a place where the robot has reached the target. In the second plot, the changes in the orientation angle of the robot can be seen. The sloping lines There are places where the robot goes around and the lines with almost zero slope are the places where the robot moved straight. Fig. 15 is the relative angle of obstacles with the robot. The angle of the robot with each obstacle should not be zero, because in this case, the robot is moving directly towards the obstacle and will collide with the object. But as it can be seen from the plot, at first the mobile robot was going towards the obstacles, but the change of angle is done well and the collision with the obstacles is prevented.

Fig. 16 is the control signals (left and right wheel voltages) and as it is shown clearly, their change patterns are almost the same. But in some places where the robot is rotated, the voltage of one wheel is higher than the other. There are some peaks in target angle. The reason is to avoid collision with obstacles. In Fig. 15, the Maximum voltage of the left and right wheels is under 12 volts so saturation blocks work properly. we put saturation blocks after the fuzzy logic controller for output signals. Because we assume that the maximum voltage of the motors of the robot's wheels is 12 v.

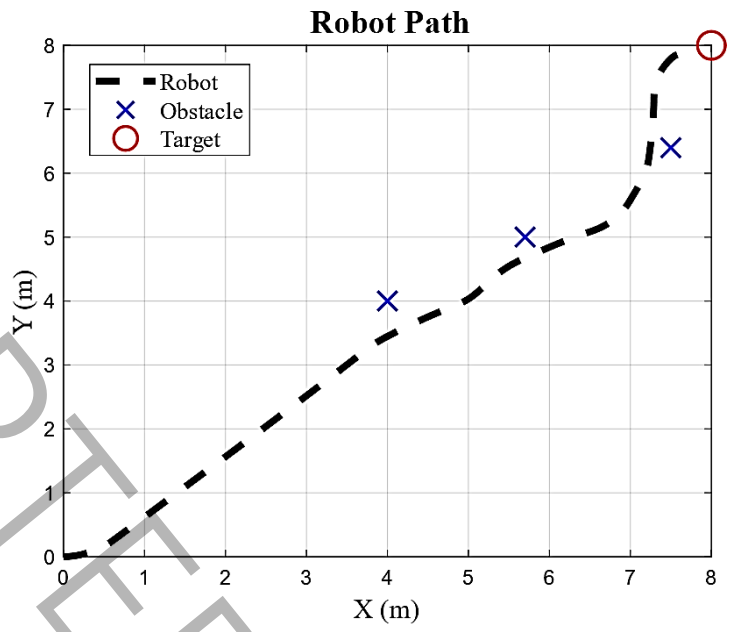

Fig 12. trajectory of a robot while the target reaching

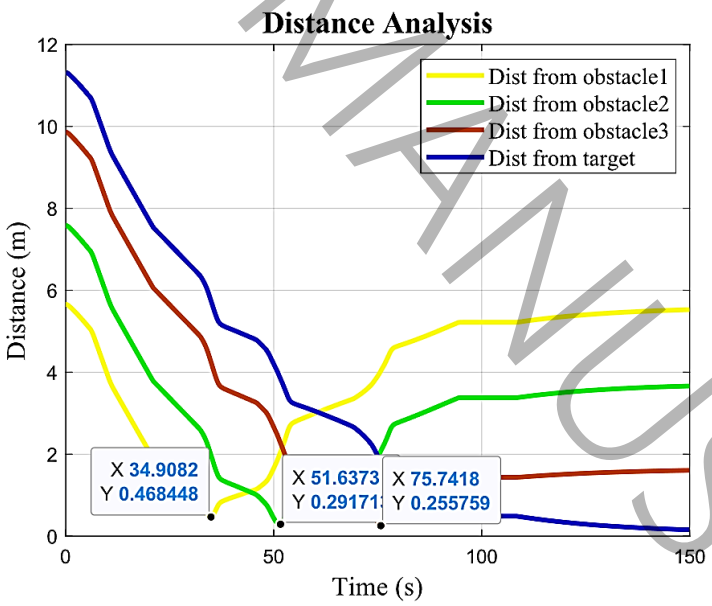

Fig 13. distance from the target and distance from obstacles while the target reaching

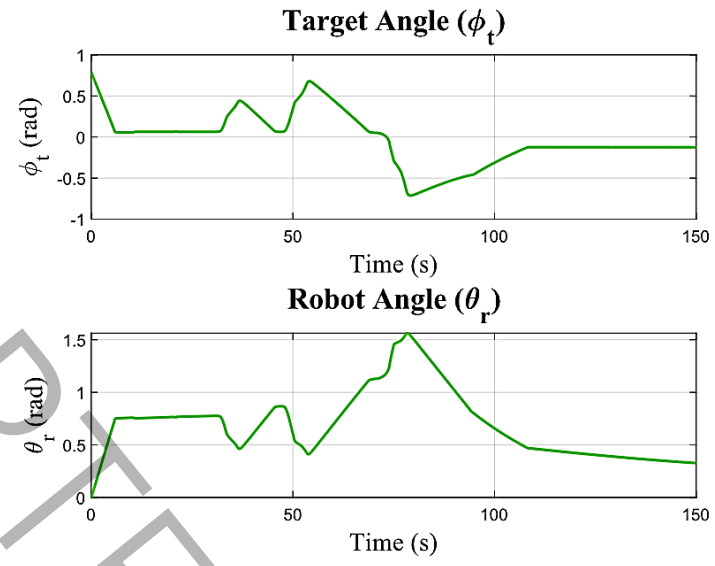

Fig 14. Relative Angle of the robot with target and robot heading angle

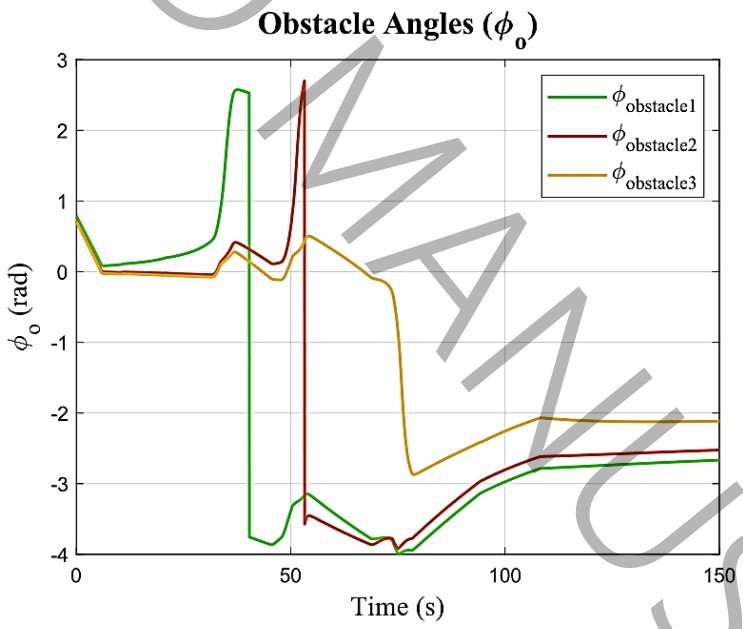

Fig 15. Angles of the robot with obstacles

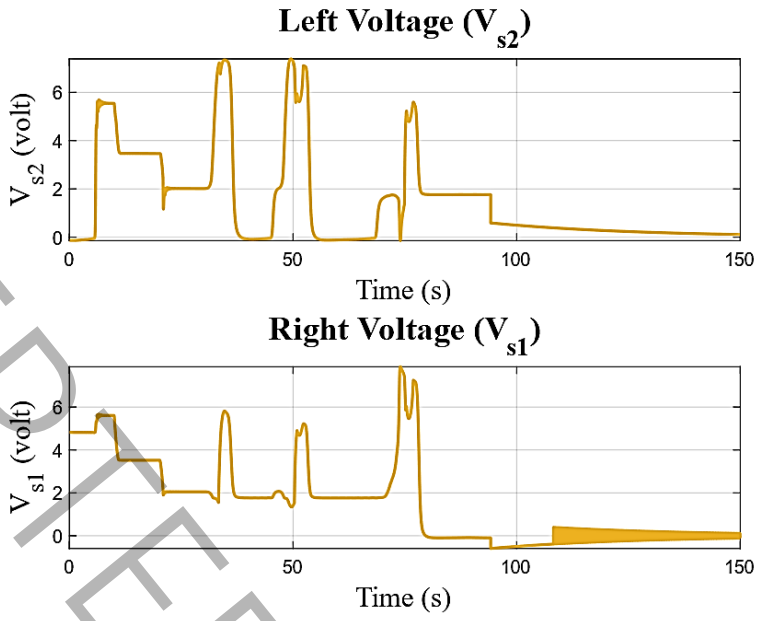

Fig 16. Control signals to the plant (left and right motor voltage)

4-2- The second scenario: Target reaching of a robot with one dynamic obstacle and one static obstacle

Similarly to the previous scenario, In this section, a dynamic obstacle (it could be another robot) with constant acceleration is considered, and its dynamic equations are given by Eqs. 13,14, starts going from the target coordinate and comes to the mobile robot and also, a static obstacle with coordinates of (4,3.58), exist along the trajectory of the robot to the target. So the mobile robot avoids collision with dynamic and static obstacles simultaneously. As it is clear from Fig. 17, the mobile robot bypasses the static obstacle in t = 29 seconds with a distance of 38 cm. It also passes the dynamic obstacle in t = 61 seconds with a distance of 72 cm and then towards the target. Also, as it is clear from Fig. 18, the target angle will eventually reach zero. The peaks that appear in the target angle plot are related to preventing collision with static and dynamic obstacles. The dynamic of an obstacle is as follows: The initial position of the dynamic obstacle is the target position ($x_0 = x_{target}; y_0 = y_{target}$). The initial speed is zero. The acceleration of the axis is -0.001 and the acceleration of the axis is -0.00123.

$$x_{obs} = 0.5a_x t^2 + v_{0x} t + x_0 \tag{13}$$

$$y_{obs} = 0.5a_y t^2 + v_{0y} t + y_0 \tag{14}$$

In Fig. 21, the control signals for both the left and right wheels do not reach 12 volts, meaning that the saturation block's effect cannot be directly observed in this scenario. The saturation block is designed to limit the control signal at 12 volts, but since the applied control inputs remain below this threshold, its functionality is not explicitly demonstrated here. However, in more demanding scenarios, such as the third scenario, the effect of the saturation block becomes evident. As shown in Fig. 26, the right wheel voltage attempts to exceed 12 volts at a certain moment, but the saturation block effectively constrains it to the 12-volt limit, confirming its proper operation. In Fig. 19, the relative angle of the robot with the target peaks at 29 seconds and 61 seconds. While the angle was fixed before these peaks, the reason is that the robot encounters obstacles and tries to avoid the collision. Finally, the relative angle of the robot with the target reaches zero. The heading angle is changing and is finally fixed. In general, from the comparison of the images of the plots, it can be said that the heading is the opposite of the relative angle of the target.

In Fig. 20, the change of the relative angle of the robot with the obstacles can be seen, which should not be zero. Therefore, we can see that the angle of the robot with the obstacles went to zero at first. But after seeing the obstacles, we have a peak in the plot. The reason is, as mentioned before, to prevent collision with obstacles. Fig. 21 shows the voltages of the left and right wheels of the robot. In approx. Most of the time, the pattern of the left and right wheel voltage signals are the same, but in some cases, they differ because of the rotation of the robot.

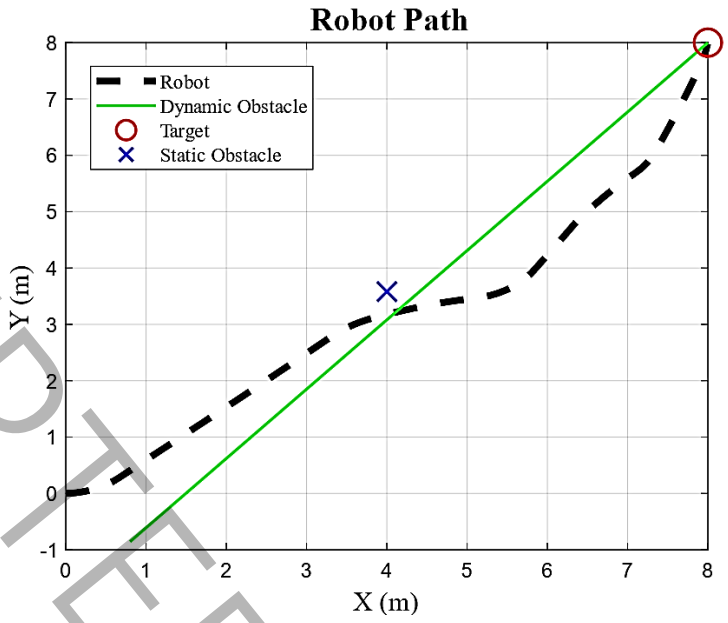

Fig 17. trajectory of the robot while the target reaching

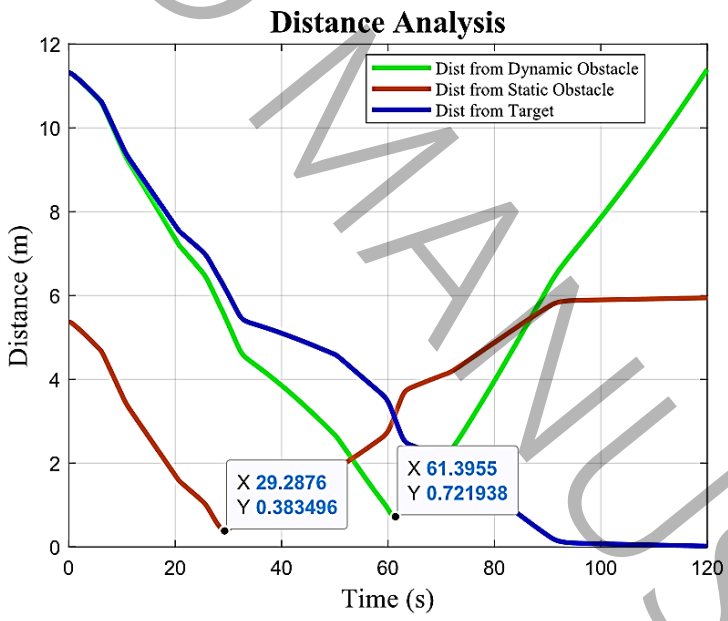

Fig 18. distance from the target and distance from obstacles while the target reaching

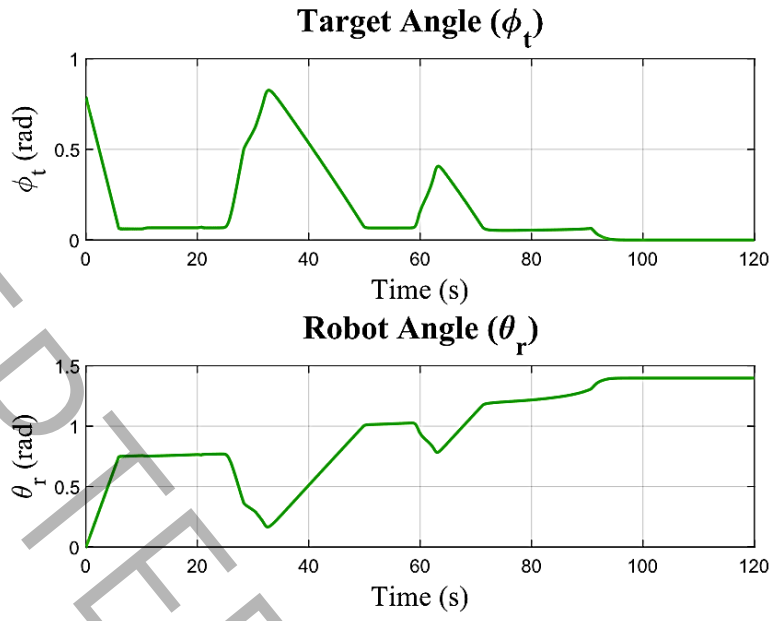

Fig 19. Relative Angle of the robot with target and robot heading angle

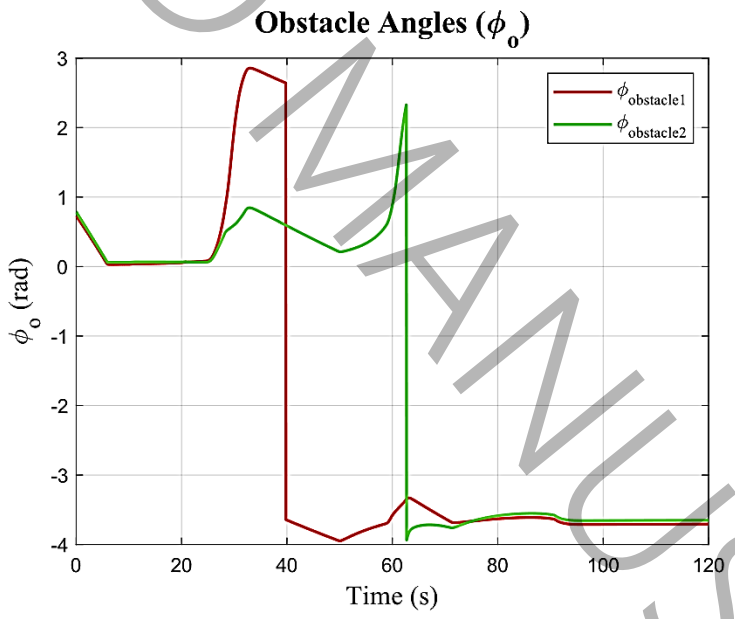

Fig 20. Angles of the robot with obstacles while the target reaching

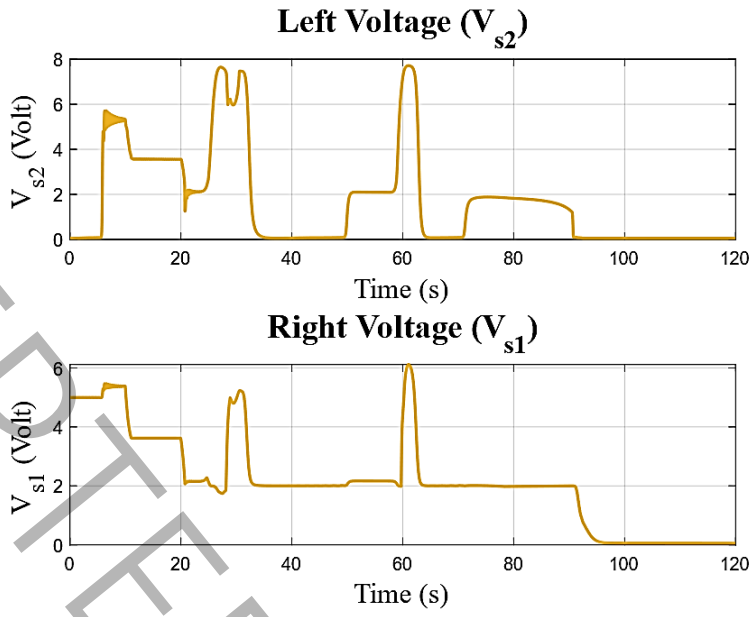

Fig 21. Control signals to the plant (left and right motor voltage)

4-3-The third scenario: Target reaching of a robot with one dynamic obstacle and one static obstacle with disturbance

In this scenario, it can be seen in Fig. 22 that first, the robot leaves the origin for the destination(Target). In the period of 25 seconds to 35 seconds, the fuzzy inputs are disturbed. Disturbance value is sin(2t). It was at this time that in the previous scenario, the robot reached the static obstacle(the static obstacle exists in (4,3.58)). But it can be seen that the controller has solved the disturbance well and after this period has taken over the control of the robot and has managed to guide the mobile robot to the target again. It can also be seen from Fig. 23 that the amount of time spent for The arrival of the robot to the target has not changed significantly compared to the previous scenario.

It can be seen in Fig. 23 that the robot reaches the first static obstacle at t = 33 seconds and bypasses it with a distance of 63 cm. Then at t = 57 seconds, it also reaches the dynamic obstacle and bypasses it with a distance of 73 cm. Also, In Fig. 24, the changes in the relative angle of the robot with the target and the changes in the orientation angle of the robot can be seen. It did not cause the instability of the robot, and even the robot was controllable after that. Fig. 25 also shows the relative angle change of the robot with each of the obstacles. Again, the peaks happened in the moments when the robot faced the obstacles. Fig.

26 also shows the voltages of the left and right wheels. As it was said and it is clear that between 25 and 35 seconds the control voltage is noisy, but after that, the control is done well again.

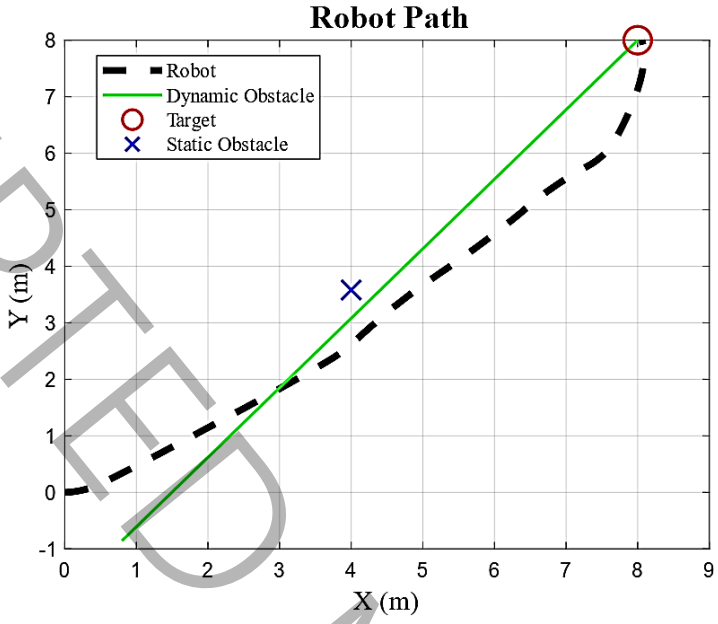

Fig 22. the trajectory of a robot while target reaching

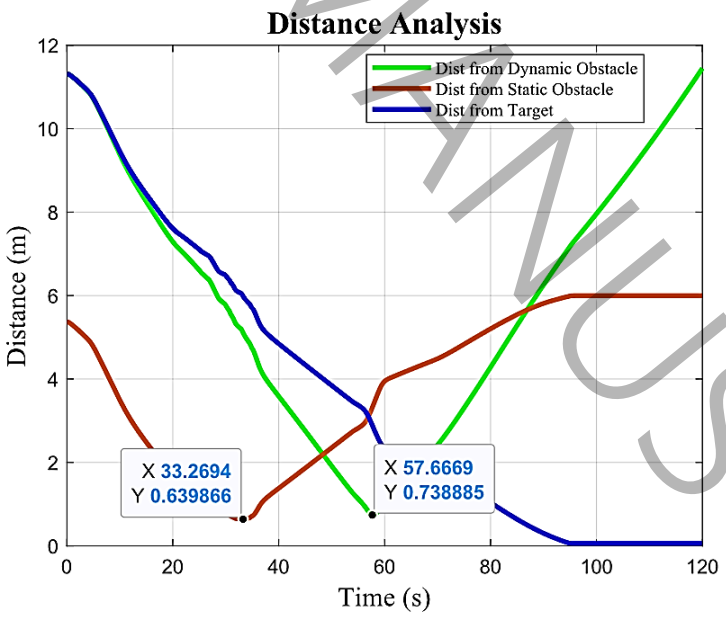

Fig 23. distance from target and distance from obstacles while the target reaching

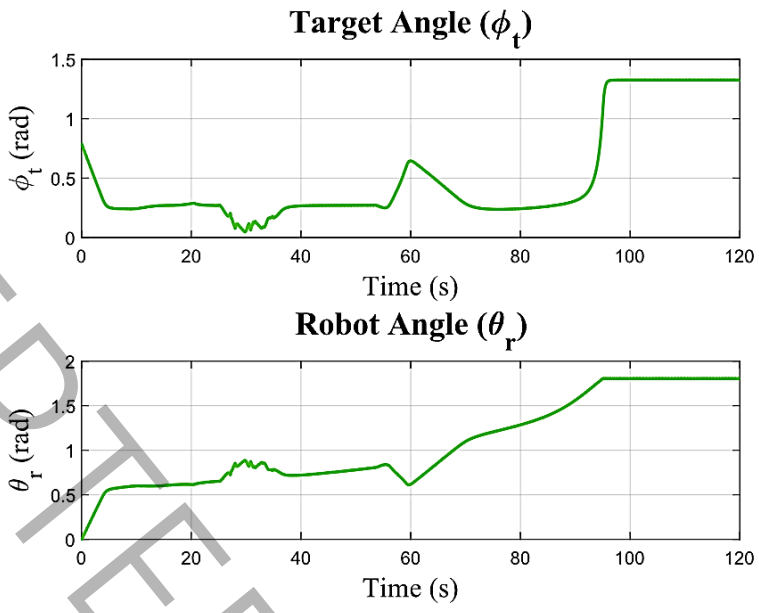

Fig 24. Relative Angle of the robot with target and robot heading angle

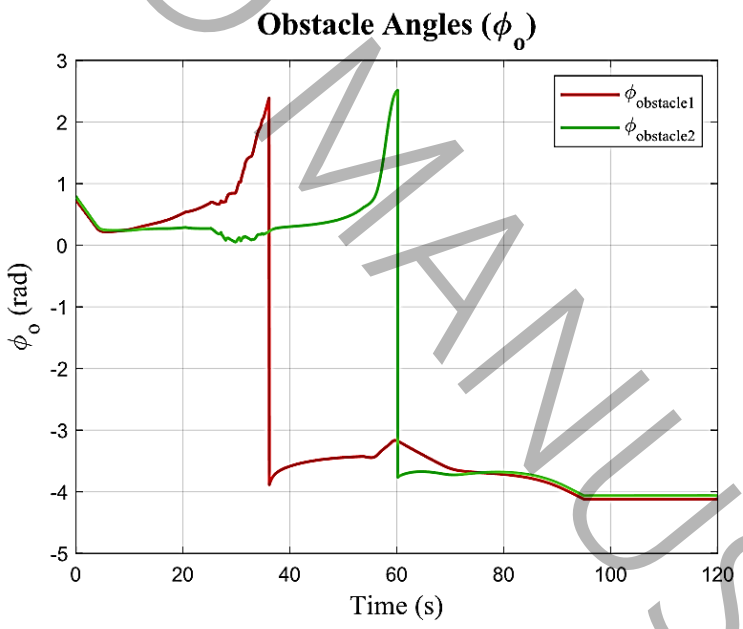

Fig 25. Angles of the robot with obstacles while the target reaching

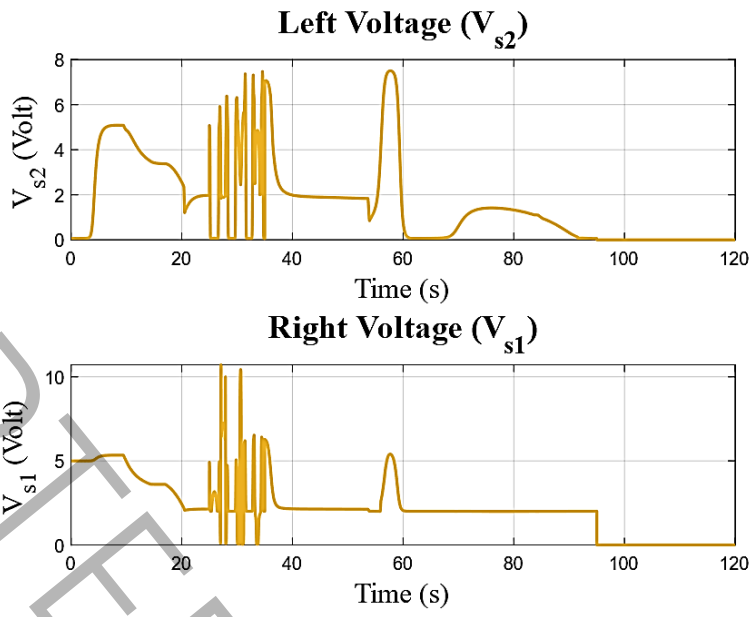

Fig 26. Control signals to the plant (left and right motor voltage)

In Fig. 26 we can see that the control signals are in good condition, they are both under 12 volts. so the saturation block works properly.

4-4- Comparative Analysis of the Proposed Method and Method in [19]

In this section, the performance of the proposed method is compared with the approach presented in [19]. The robot's trajectory in the 2D plane (x,y) was plotted for both methods, and key parameters were analyzed. The comparison criteria include Total path length, Mean absolute deviation (MAD), Root mean square error (RMSE) and Minimum distance to obstacles.

Performance Evaluation Metrics

1. **Total Path Length :**

   This metric represents the total distance traveled by the robot from the start to the target. A shorter path length indicates a more efficient movement, as given by Eq. 15

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{15}$$

2. **Mean Absolute Deviation (MAD):**

This metric measures the average absolute deviation between the actual trajectory and the ideal path. A lower value indicates a more stable and smoother trajectory, as computed using Eq. 16.

$$MAD = \frac{1}{N}\sum_{i=1}^{N}\left|d_i - d_{ideal}\right| \tag{16}$$

where d_i is the distance of the ith point from the ideal path.

3. **Root Mean Square Error (RMSE)**

RMSE quantifies the deviation of the actual trajectory from the ideal path. A lower RMSE value indicates fewer fluctuations and a more precise movement, as computed using Eq. 17.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(d_i - d_{ideal}\right)^2} \tag{17}$$

4. **Minimum Distance to Obstacles**

This metric represents the shortest distance between the robot and obstacles during its motion. A smaller value may indicate that the robot moves closer to obstacles, which could pose safety challenges.

In this study, the ideal path is considered as the straight-line trajectory from the start point to the target**.** This represents the shortest possible distance between the start and target, assuming there are no obstacles in the environment. However, due to the presence of obstacles, the robot inevitably deviates from the ideal path. This deviation is a natural part of the navigation process. The key objective is to ensure effective maneuverability around obstacles while maintaining a smooth and controlled trajectory**.** Therefore, comparing metrics such as MAD and RMSE helps assess the quality of the robot's motion. The table below presents the results for both methods :

Table IV. Numerical Comparison Results Between the Proposed Method and the Method in [19]

Metric	Method in [19]			Proposed Method		
	Scenario .1	Scenario .2	Scenario .3	Scenario .1	Scenario .2	Scenario .3
Total Path Length	12.4996	20.7641	22.0387	11.8332	11.8672	11.8263
Mean Absolute Deviation (MAD)	0.7037	2.6611	2.8844	0.3653	0.5727	0.6298
Root Mean Square Error (RMSE)	0.8964	3.2797	3.4752	0.4825	0.7449	0.7474
Minimum Distance to Obstacles	0.3891	2.22	2.3	0.2557	0.3834	0.6398

Based on Table IV and Fig. 2٧, the results indicate that the proposed method provides a more effective trajectory compared to the method in [19]. To ensure a fair comparison, we implemented the controller from [19] on our plant. While their method may have performed well on its original plant, which was modeled using a bond graph approach, its effectiveness on our system is more limited. The total path length is lower in our method, meaning the robot traveled a shorter distance to reach the target. The lower MAD value suggests a more stable and smoother trajectory with fewer unnecessary deviations, while the lower RMSE indicates more precise navigation with reduced deviation from the ideal path.

Although the minimum distance to obstacles is smaller in our method, it remains within an acceptable range, demonstrating good maneuverability and effective obstacle avoidance. These results highlight the advantages of our approach in the context of our plant. Thus, our method has successfully achieved these improvements—the robot travels a shorter distance, follows a more stable trajectory, and exhibits reduced deviation from the ideal path.

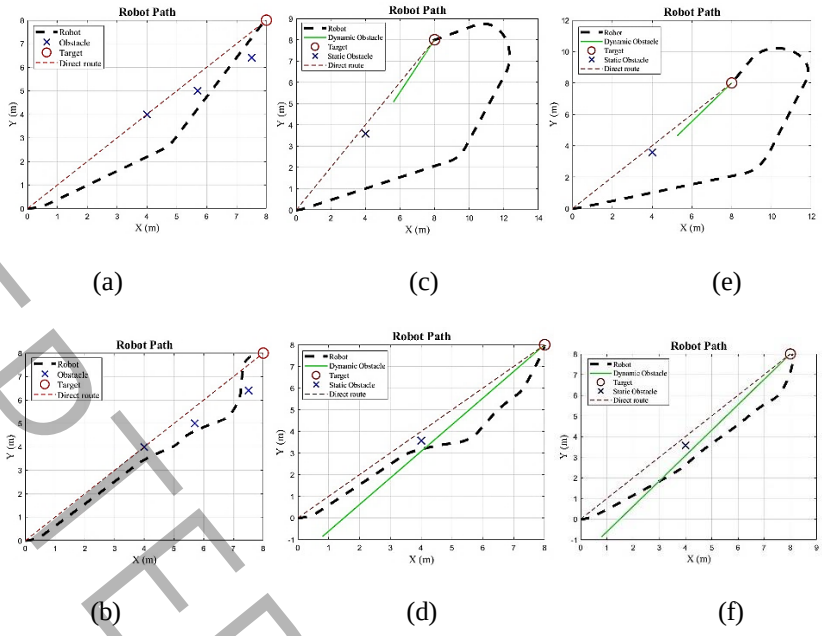

Fig 27. Trajectory of the robot while reaching the target: (a) method from [19] (scenario .1), (b) proposed method (scenario .1), (c) method from [19] (scenario .2), (d) proposed method (scenario .2), (e) method from [19] (scenario .3), (f) proposed method (scenario .3)

5. Conclusion

This study introduced and applied a fuzzy logic controller to navigate a four-wheeled skid-steering mobile robot based on McCormick's previous work. Despite the lack of specific research on controlling this robot model according to McCormick's paper, we successfully designed a fuzzy logic controller tailored to this linear model. We established fuzzy rules to guide the robot in reaching its target and avoiding obstacles effectively. By simulating various scenarios, we ensured the controller's resilience against uncertainties, including dynamic obstacles and disturbances. Our experiments demonstrated the controller's ability to navigate around static obstacles with precision and to adapt to dynamic obstacles, albeit with slightly delayed target reaching. Even in the presence of disturbances, the controller maintained its effectiveness, guiding the robot safely to its destination without collisions. In conclusion, this study showcases the efficacy of a fuzzy logic controller in maneuvering a mobile robot, paving the way for further research in intelligent control systems such as fuzzy-PID and neural-fuzzy controllers for enhanced performance in real-world applications.

References

[1] J. Liao, Z. Chen, B. Yao, Model-Based Coordinated Control of Four-Wheel Independently Driven Skid Steer Mobile Robot with Wheel–Ground Interaction and Wheel Dynamics, IEEE Transactions on Industrial Informatics, 15(3) (2019) 1742-1752.
[2] E. McCormick, H. Lang, C.W. de Silva, Dynamic Modeling and Simulation of a Four-Wheel Skid-Steer Mobile Robot Using Linear Graphs, Electronics, 11(15) (2022).
[3] H.M. Paynter, Analysis and design of engineering systems, MIT press, (1961).
[4] W. Borutzky, Bond graph modelling and simulation of multidisciplinary systems – An introduction, Simulation Modelling Practice and Theory, 17(1) (2009) 3-21.
[5] S.R. Sahoo, Dynamic modeling of four-wheel skid mobile robot by unified bond graph approach, in: International Conference on Robotics: Current Trends and Future Challenges (RCTFC), 2016.
[6] H.N. Nur Uddin, Auralius Manurung,Hendi Hermawan, Teddy Mohamad Darajat, Kinematics Modeling and Motions Analysis of Non-holonomic Mobile Robot, in: International Conference on Information and Communications Technology (ICOIACT), 2022.
[7] K. KOZŁOWSKI, Modeling and control of a 4-wheel skid-steering mobile robot, International journal of applied mathematics and computer science, 14(4) (2004) 477-496.
[8] Mohamed Fnadi , Fr´ed´eric Plumet ,Fa¨ız Benamar, Model predictive control based dynamic path tracking of a four-wheel steering mobile robot, in: International Conference on Intelligent Robots and Systems (IROS), 2019.
[9] M. Trojnacki, Dynamics Model of a Four-Wheeled Mobile Robot for Control Applications – A Three-Case Study, in: Proceedings of the 7th IEEE International Conference Intelligent Systems, 2015, pp. 99-116.
[10] S. Dian, H. Fang, T. Zhao, Q. Wu, Y. Hu, R. Guo, S. Li, Modeling and Trajectory Tracking Control for Magnetic Wheeled Mobile Robots Based on Improved Dual-Heuristic Dynamic Programming, IEEE Transactions on Industrial Informatics, 17(2) (2021) 1470-1482.
[11] B. Saeedi, Implementation of Behavior-Based Navigation Algorithm on Four-Wheel Steering Mobile Robot, Journal of Computational and Applied Mechanics, 52(4) (2021) 619-641.
[12] Khaled Khnissi, Chiraz Ben Jabeur, Hassene Seddik, 3D simulator for navigation of a mobile robot using simscape-SIMULINK, in: International Conference on Control, Automation and Diagnosis (ICCAD), 2019.
[13] K.K.A. Farag, H.H. Shehata, H.M. El-Batsh, L. Fortuna, Mobile Robot Obstacle Avoidance Based on Neural Network with a Standardization Technique, Journal of Robotics, 2021 (2021) 1-14.
[14] Kecskés, István, Z. Balogh, P. Odry, Modeling and fuzzy control of a four-wheeled mobile robot, in: IEEE 10th Jubilee International Symposium on Intelligent Systems and Informatics, 2012, pp. 205-210.
[15] R. Singh, T.K. Bera, Obstacle avoidance of mobile robot using fuzzy logic and hybrid obstacle avoidance algorithm, in: IOP Conference Series: Materials Science and Engineering, 2019.
[16] C.T. Lee, B.-R. Su, C.-H. Chang, T.-Y. Hsu, W.-D. Lee, Applications of Taguchi method to PID control for path tracking of a wheeled mobile robot, in: IEEE International Conference on Applied System Invention (ICASI), 2018, pp. 453-456.
[17] Q. Xu, J. Kan, S. Chen, S. Yan, Fuzzy PID Based Trajectory Tracking Control of Mobile Robot and its Simulation in Simulink, International Journal of Control and Automation, 7 (2014) 233-244.
[18] A. Pandey, D.R. Parhi, Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm, Defence Technology, 13(1) (2017) 47-58.
[19] R. Singh, T.K. Bera, Fuzzy Logic Controller for Obstacle Avoidance of Mobile Robot, International Journal of Nonlinear Sciences and Numerical Simulation, 20(1) (2019) 51-62.
[20] N. Tanha, Seyed Davood, M.H. Korayem, S. Fathollahi Dehkordi, Path Design and Control of a Moving Social Robot in an Environment with Moving Obstacles to Reach a Moving Target through Fuzzy Control, Amirkabir Journal of Mechanical Engineering, 53(2) (2021) 993-1014.

[21] A. Ghasemi Zade, M. Sadedel, Behavior-Based Control of Mecanum Four-Wheeled Omnidirectional Robot, Amirkabir Journal of Mechanical Engineering, 54(2) (2022) 309-332.
[22] D. Rowell, Advanced System Dynamics and Control, Massachusetts Institite of Technology. Department of Mechanical Engineering, (22) (2004) 1-41.
[23] V. Mahajan, P. Agarwal, H.O. Gupta, Power quality problems with renewable energy integration, in: Power quality in modern power systems, Elsevier, 2021, pp. 105-131.