[1]

Collision Avoidance using Fuzzy Logic

Paul Babala, Aditya Kothari

***Abstract*—Mobile robots have found application in a broad spectrum of fields such as agriculture, mining, military, space travel, transportation, and more. These robots are required to correctly identify and maneuver around static as well as dynamic objects maintaining safety of the people/ objects around. Our goal is to develop a collision avoidance algorithm for mobile robots working in a warehouse using Fuzzy Logic.**

***Index Terms*—Mobile robot, Warehouse robot, Fuzzy logic controller, Collision avoidance, Robot safety, Simulink, MATLAB, Gazebo, Lidar.**

I. Introduction

A mobile robot is an autonomous machine which can perform a predefined task with no to minimal human supervision. These tasks are generally categorized into four categories: perception, navigation, motion planning, and control. With these tasks, mobile robots have found application within a broad spectrum of fields such as agriculture, mining, space travel, transportation, and many more. A key aspect to these robots is the ability to correctly identify and appropriately maneuver around both static objects in environments as well as dynamic objects like personnel and other robots. Due to this complexity, there is an opportunity to evaluate if applying fuzzy logic control can provide an improvement when compared to traditional controllers in these environments.

II. Literature Review

A wealth of research has been done in the field of local path planning focused on mobility under complex unknown environments. Most of these algorithms have been grouped into four categories:

A. *Artificial Potential Field Based Algorithm*

The Artificial Potential Field (APF) algorithm for path planning was first introduced by [1]. In APF Algorithms the robot is considered to be moving in a virtual potential field. The target and obstacles around the robot are then described as attractive and repulsive potentials respectively. The APF algorithms are simple to formulate, easy to implement and fast enough for real-time applications. Major limitations of APF algorithms are: failure to find global minimum, failure to find a solution and oscillations in the found solutions. Much research has been done to overcome the previously mentioned limitation and to improve the performance of APF eg. [2][3][4] The practical limitation of APF based algorithms is that different tuning or completely different potential functions may be required for different scenarios.

B. *Population Involved Meta Heuristic Based Algorithm*

This group of algorithms are derived from studying social behavior of different species or objects found in nature. Key advantages of such algorithms are, ease of implementation, assurance of finding global minima and shortest path generation. Many algorithms based on Genetic Algorithms or modified Genetic Algorithms have been used for solving the problem of path planning eg. [5][6][7]. Other than Genetic Algorithms, Particle Swarm Optimization based algorithms [8][9] and biology inspired algorithms such as Ant Colony Optimization algorithm [10] have been used for path planning and collision avoidance. The major drawback of such algorithms is the compute power and time required to generate an optimal path.

C. *Artificial Neural Network Based Algorithm*

With recent advances in Machine Learning and Deep Learning algorithms, ANN algorithms have become a new benchmark in path planning and collision avoidance applications. In [11] authors have used ANN with Reinforcement Learning. In [12] authors propose a distance based SNN. The ANN based algorithms are capable of learning linear, non-linear and complex input-output relationships between the sensor readings and expected robot behavior, which makes them highly effective for path planning and collision avoidance application. The disadvantages of these algorithms include the amount of data required for training and the black-box nature of these algorithms.

D. *Fuzzy Logic Based Algorithm*

Inspired by human-like reasoning, Fuzzy Logic was introduced by Lofti Zadeh [13] Fuzzy Logic algorithms are good at handling uncertainty and imprecise information, which makes them a perfect candidate for path planning and collision avoidance applications. Many researchers have successfully used Fuzzy Logic based algorithms for path planning and collision avoidance [14][15][16]. The biggest disadvantage of a Fuzzy Logic based algorithm is the maintenance of a fuzzy rule base and making sure that the rule base is correct, complete and consistent.

III. Proposal

In this report we propose a Fuzzy Logic Control (FLC) for mobile robot collision avoidance in a warehouse. Our proposal was inspired by previous work in risk prediction for performing a lane change using a fuzzy logic method in [14]. The remaining section provides the design requirements for such a FLC.

A. *Requirements*

We proposed the following as our requirements for this study:

- Consider actors/people are not bounded by lane like markings while moving within the warehouse.
- Consider the relative angle between our vehicle and the target vehicle path.

[1]

- Reuse the existing algorithm for trajectory prediction and modify it to better navigate the dynamically variable environment of moving actors.

Our initial proposal included the need to provide lane assist functionality with other robots in the environment, but we moved away from this objective due to the following reasons:

- We are using the motion of the actors in the simulation to achieve the unpredictable dynamics. The actors are not bounded by structured lanes which we deemed to be more difficult and interesting to investigate.
- The original paper we reviewed for lane assist did not consider the relative angle between the primary vehicle and the target vehicle path [14]. For our application, this relative angle is important for increasing the speed and safety of our robot's controller.
- The original paper relied on predictable trajectories of the target vehicle [14] which cannot be applied to our scenario as we are considering actors walking across the intended route without the use of safety crosswalks.

B. *Software Process*

We used a standard waterfall approach to our software design. Our requirements were defined by our proposal. Initially we had still considered the lane assist requirement. A Simulink and Gazebo Simulation was used as our starting system design for the project. After testing with this code and simulation, it was determined that applicability was not necessary for our investigation. We amended the requirements to our project and reevaluated our system design. From here, the basic Fuzzy Logic Controller (FLC) was developed. While testing this design, some undesirable characteristics like a slower time to run through the entire route was seen. The FLC was optimized further and retested. The process was concluded after completing analysis and provided the required documentation for the project. Figure 1 shows the diagram for the typical Waterfall methodology that we followed.

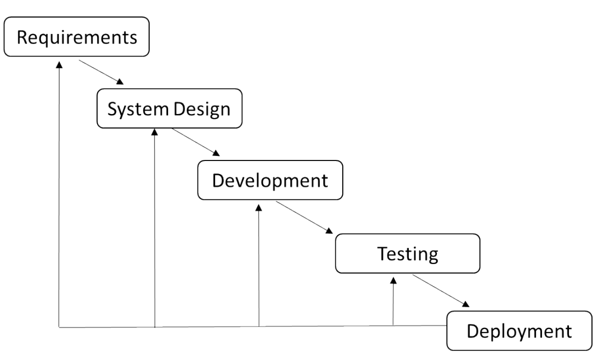

Fig. 1. Waterfall Software Process.

IV. DESIGN

Our design consisted of two portions: an preexisting Simulink traditional robot controller that is provided by MATHWORKS, and a custom developed FLC using MATHWORKS' Fuzzy Logic Toolbox. The original controller was provided by MATHWORKS as the simulateWarehouseRobotInGazebo Simulink solution. This code interfaced with the Gazebo Open Source libraries for sensor emulation and robot simulation which is detailed further in section V Simulation Setup.

A. *Traditional Controller*

The traditional controller consisted of four primary blocks: Sense, Schedule, Control, and Actuate. Figure 2 shows the top level architecture of the controller.

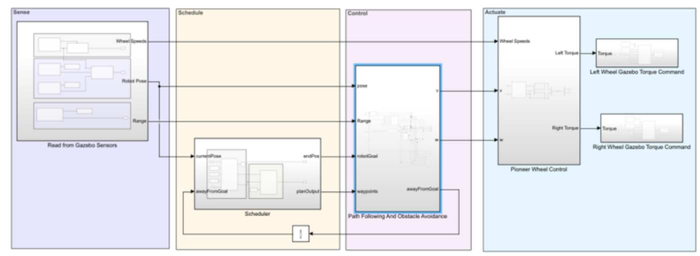

Fig. 2. Simulink Traditional Controller Simulate Warehouse Robot in Gazebo Top Level Design.

The Sense blocks consist of a sensor for Wheel Speed, Robot Pose (orientation), and Light Detection and Ranging (LIDAR) Range to an obstacle. The current Pose is sent to the Scheduler block within the Schedule block to generate waypoints and end position that the robot is required to drive to. These outputs are sent to the Path Following And Obstacle Avoidance block within the Control block in addition to the Range from an obstacle and the current Pose of the robot. The Control block uses the distance away from a given waypoint as the primary feedback into the controller. Figure 3 provides a more detailed view of the Path Following and Obstacle Avoidance block.

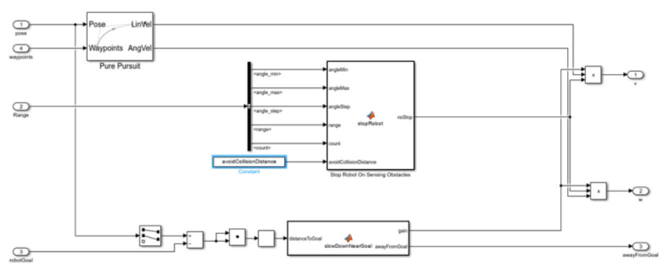

Fig. 3. Simulink Traditional Controller Path Following And Collision Avoidance.

This block consists of a few separate subsystems: Pure Pursuit, Stop Robot on Sensing Obstacles, and Slow Down Near Goal. The Pure Pursuit takes in the Pose of the Robot and the current waypoints and decomposes them into a linear velocity (v) and angular velocity (w). The Stop Robot is used to stop the robot before hitting an obstacle. A Boolean value is output to determine if the robot should stop or not. The FLC was inserted in between this subsystem which will be detailed later. The slowDownNearGoal is provided to avoid the robot from overshooting the required waypoint as the scenarios require the robot to turn around and return back to the origin location of the simulation. The outputs of the Control block, v and w, are sent to the Actuate block. The Simulink provides an emulation of a Pioneer 3-DX navigation robot to send wheel torques to.

B. Fuzzy Logic Controller

The FLC was inserted into the traditional controller of the simulink project in such a way that the outputs into the Actuator Block could be switched between the FLC or the traditional controller via a Simulink Switch block. Due to this, the FLC could more easily be inserted into the Simulink model since it uses and provides the same inputs/outputs.

Figure 4 details the Flowchart of the model for the FLC. The model is initialized and given a starting location and Pose. Each frame, the model evaluates two paths concurrently. The current Pose is read while the lidar sensor scans the simulation environment for obstacles. Using the current location, the Pure Pursuit Control block is used to determine the Actuator's desired linear velocity and angular velocity to drive the robot to the next waypoint in the route. Simultaneously, the FLC is computing if a Stop signal needs to be set to avoid any obstacles between the waypoints. If a collision is not deemed to be critical, the Pure Pursuit control's outputs will be passed through to the Actuate block. If a collision is imminent and is deemed critical, the FLC will set the NoStop signal low/0 (which indicates a stop action) and the FLC's output is selected. The FLC's output uses the slowDownNearGoal as an instantaneous stop is not realistic.

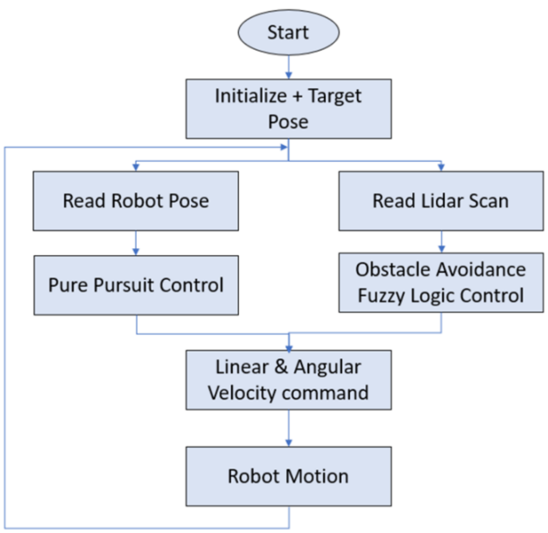

Fig. 4. Fuzzy Logic Algorithm Flowchart.

Figure 5 shows the addition of the FLC in the original Simulink model. The signals going off the figure are datapumps that are used for analysis of the controllers. The following values are saved off each frame into a csv file: time_sec, poseX_m, poseY_m, poseTheta_rad, noStop, relRange_ft, relSpeed_ft, relAngle_rad, and fuzzyOutput.

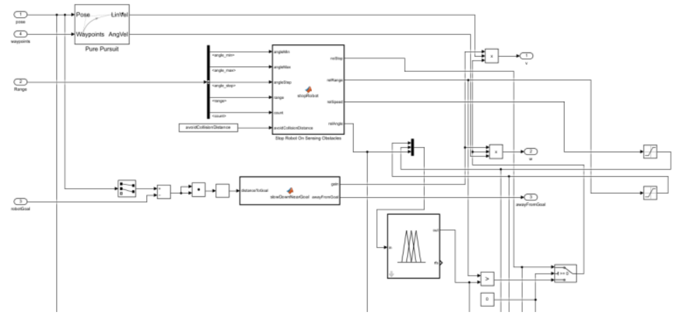

Fig. 5. Fuzzy Logic Controller In Simulink.

Figure 6 details the Fuzzy Logic Controller Block Diagram. Three inputs, relative distance to distance, relative speed to target, and relative angle to target are fed into a Mamdani type 1 Fuzzy Logic. The output of this controller is threshold distance.

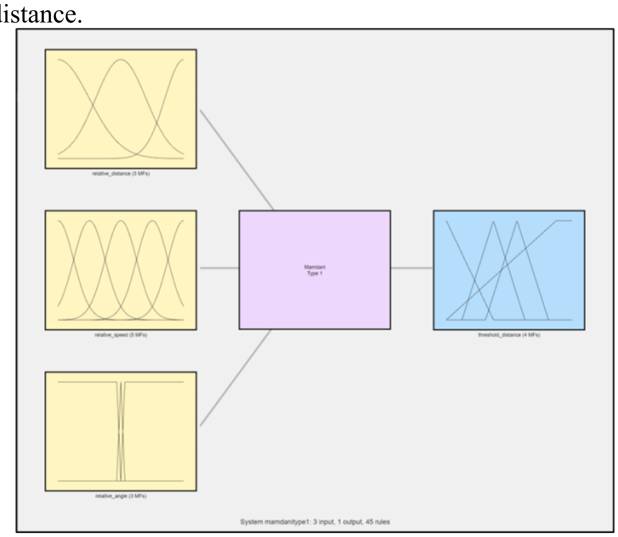

Fig. 6. Fuzzy Logic Controller Block Diagram.

The Relative distance membership function is shown in figure 7. It consists of three linguistic values: low, medium and high.

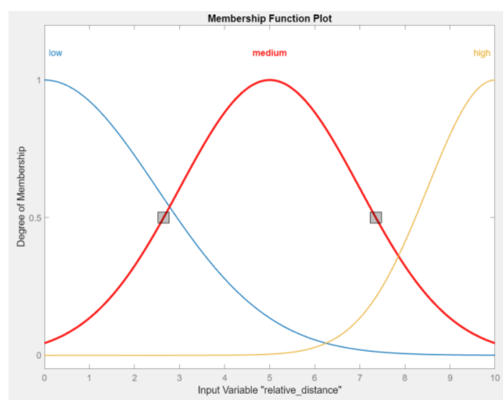

Fig. 7. Relative Distance Membership Function.

Table I provides the details for each of the linguistic values used in the Fuzzy Logic toolbox.. All three values follow a Gaussian membership function. Generally, a low/close distance is set to be between 0 to 3 meters, medium distance is 3 to 8 meters, and high/far distance is 8 to 10 meters.

TABLE I
RELATIVE DISTANCE DEGREE OF MEMBERSHIP

Low	Gaussian	[2.5, 0]
Medium	Gaussian	[2, 5]
High	Gaussian	[1.5, 10]

The Relative speed membership function is shown in figure 8. It consists of five linguistic values: negative, small_negative, zero, small_positve and positive.

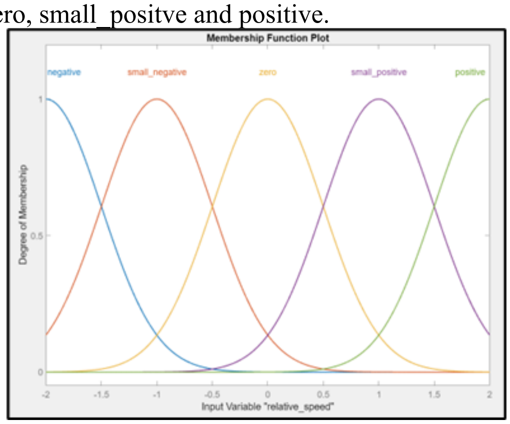

Fig. 8. Relative Speed Membership Function.

Table II provides the details for each of the linguistic values used in the Fuzzy Logic toolbox.. All five values follow a Gaussian membership function. Generally, a negative speed is defined as -2 to -1.5 meters per second (mps), small_negative is -1.5 to .5 mps, zero is -.5 to .5 mps, small_positive is .5 to 1.5 mps, and positive is 1.5 to 2 mps.

TABLE II
RELATIVE SPEED DEGREE OF MEMBERSHIP

Negative	Gaussian	[0.5, -2]
Small Negative	Gaussian	[0.5, -1]
Zero	Gaussian	[0.5, 0]
Small Positive	Gaussian	[0.5, 1]
Positive	Gaussian	[0.5, 2]

The relative angle membership function is shown in figure 9. It consists of three linguistic values: negative, zero and positive.

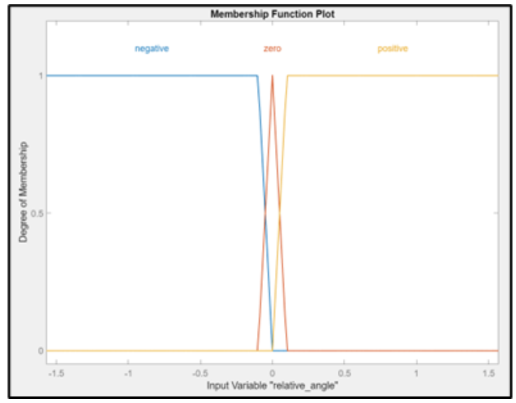

Fig. 9. Relative Angle Membership Function.

Table III provides the details for each of the linguistic values used in the Fuzzy Logic toolbox.. The negative and positive values use trapezoidal membership function, while zero uses a triangular membership function. Generally, negative is -1.5 to -.1 radians, zero is -.1 to .1 radians, and positive is .1 to 1.5 radians.

TABLE III
RELATIVE ANGLE DEGREE OF MEMBERSHIP

Negative	Trapezoidal	[-1.57, -1.57, -0.1, 0]
Zero	Triangular	[-0.1, 0, 0.1]
Positive	Trapezoidal	[0, 0.1, 1.57, 1.57]

The threshold distance membership function is shown in figure 9. It consists of four linguistic values: high, medium, low, and critical.

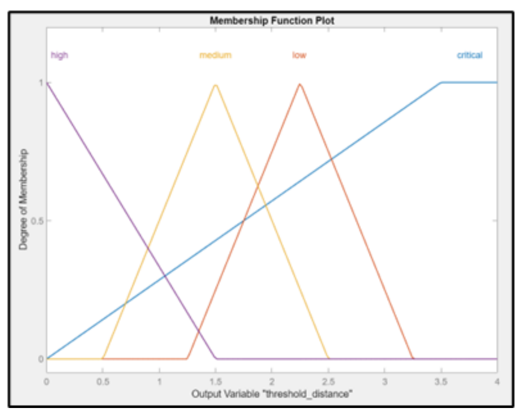

Fig. 10. Threshold Distance Membership Function.

Tablet IV provides the details for each of the linguistic values used in the Fuzzy Logic Toolbox. Critical and High values used trapezoidal membership functions while Low and Medium used triangular membership functions. Generally, a high threshold distance is between 0 and .75, medium is .75 to 2, low is 2 to 2.75, and critical is a linear function that becomes most relevant from 2.75 to 4.

TABLE IV

THRESHOLD DISTANCE DEGREE OF MEMBERSHIP

Critical	Trapezoidal	[0, 3.5, 4, 4]
Low	Triangular	[1.25, 2.25, 3.25]
Medium	Triangular	[0.5, 1.5, 2.5]
High	Trapezoidal	[0, 0, 0, 1.5]

A total of 45 rules were created for the FLC. All rules are shown in Figure 11.

	Rule	Weight	Name
1	If relative_distance is low and relative_speed is negative and relative_angle is negative then threshold_distance is low	1	rule1
2	If relative_distance is medium and relative_speed is negative and relative_angle is negative then threshold_distance is medium	1	rule2
3	If relative_distance is high and relative_speed is negative and relative_angle is negative then threshold_distance is medium	1	rule3
4	If relative_distance is low and relative_speed is small_negative and relative_angle is negative then threshold_distance is medium	1	rule4
5	If relative_distance is medium and relative_speed is small_negative and relative_angle is negative then threshold_distance is high	1	rule5
6	If relative_distance is high and relative_speed is small_negative and relative_angle is negative then threshold_distance is high	1	rule6
7	If relative_distance is low and relative_speed is zero and relative_angle is negative then threshold_distance is medium	1	rule7
8	If relative_distance is medium and relative_speed is zero and relative_angle is negative then threshold_distance is medium	1	rule8
9	If relative_distance is high and relative_speed is zero and relative_angle is negative then threshold_distance is medium	1	rule9
10	If relative_distance is low and relative_speed is small_positive and relative_angle is negative then threshold_distance is medium	1	rule10
11	If relative_distance is medium and relative_speed is small_positive and relative_angle is negative then threshold_distance is high	1	rule11
12	If relative_distance is high and relative_speed is small_positive and relative_angle is negative then threshold_distance is high	1	rule12
13	If relative_distance is low and relative_speed is positive and relative_angle is negative then threshold_distance is high	1	rule13
14	If relative_distance is medium and relative_speed is positive and relative_angle is negative then threshold_distance is high	1	rule14
15	If relative_distance is high and relative_speed is positive and relative_angle is negative then threshold_distance is high	1	rule15
16	If relative_distance is low and relative_speed is negative and relative_angle is zero then threshold_distance is critical	1	rule16
17	If relative_distance is medium and relative_speed is negative and relative_angle is zero then threshold_distance is low	1	rule17
18	If relative_distance is high and relative_speed is negative and relative_angle is zero then threshold_distance is medium	1	rule18
19	If relative_distance is low and relative_speed is small_negative and relative_angle is zero then threshold_distance is low	1	rule19
20	If relative_distance is medium and relative_speed is small_negative and relative_angle is zero then threshold_distance is medium	1	rule20
21	If relative_distance is high and relative_speed is small_negative and relative_angle is zero then threshold_distance is high	1	rule21
22	If relative_distance is low and relative_speed is zero and relative_angle is zero then threshold_distance is high	1	rule22
23	If relative_distance is medium and relative_speed is zero and relative_angle is zero then threshold_distance is high	1	rule23
24	If relative_distance is high and relative_speed is zero and relative_angle is zero then threshold_distance is high	1	rule24
25	If relative_distance is low and relative_speed is small_positive and relative_angle is zero then threshold_distance is high	1	rule25
26	If relative_distance is medium and relative_speed is small_positive and relative_angle is zero then threshold_distance is high	1	rule26
27	If relative_distance is high and relative_speed is small_positive and relative_angle is zero then threshold_distance is high	1	rule27
28	If relative_distance is low and relative_speed is positive and relative_angle is zero then threshold_distance is high	1	rule28
29	If relative_distance is medium and relative_speed is positive and relative_angle is zero then threshold_distance is high	1	rule29
30	If relative_distance is high and relative_speed is positive and relative_angle is zero then threshold_distance is high	1	rule30
31	If relative_distance is low and relative_speed is negative and relative_angle is positive then threshold_distance is low	1	rule31
32	If relative_distance is medium and relative_speed is negative and relative_angle is positive then threshold_distance is medium	1	rule32
33	If relative_distance is high and relative_speed is negative and relative_angle is positive then threshold_distance is medium	1	rule33
34	If relative_distance is low and relative_speed is small_negative and relative_angle is positive then threshold_distance is medium	1	rule34
35	If relative_distance is medium and relative_speed is small_negative and relative_angle is positive then threshold_distance is high	1	rule35
36	If relative_distance is high and relative_speed is small_negative and relative_angle is positive then threshold_distance is high	1	rule36
37	If relative_distance is low and relative_speed is zero and relative_angle is positive then threshold_distance is medium	1	rule37
38	If relative_distance is medium and relative_speed is zero and relative_angle is positive then threshold_distance is medium	1	rule38
39	If relative_distance is high and relative_speed is zero and relative_angle is positive then threshold_distance is medium	1	rule39
40	If relative_distance is low and relative_speed is small_positive and relative_angle is positive then threshold_distance is medium	1	rule40
41	If relative_distance is medium and relative_speed is small_positive and relative_angle is positive then threshold_distance is high	1	rule41
42	If relative_distance is high and relative_speed is small_positive and relative_angle is positive then threshold_distance is high	1	rule42
43	If relative_distance is low and relative_speed is positive and relative_angle is positive then threshold_distance is high	1	rule43
44	If relative_distance is medium and relative_speed is positive and relative_angle is positive then threshold_distance is high	1	rule44
45	If relative_distance is high and relative_speed is positive and relative_angle is positive then threshold_distance is high	1	rule45

Fig. 11. Fuzzy Logic Rules.

Table V provides the primary rules that will most affect the control of the robot. Rules 1, 16, 17, 19, and 31 are shown in the tablet along with their corresponding threshold distance.

TABLE V
THRESHOLD DISTANCE CRITICALITY

Relative Distance	Relative Speed	Relative Angle	Operator	Threshold Distance
LOW	NEGATIVE	NEGATIVE	AND	LOW
LOW	NEGATIVE	ZERO	AND	CRITICAL
MEDIUM	NEGATIVE	ZERO	AND	LOW
LOW	SMALL NEGATIVE	ZERO	AND	LOW
LOW	NEGATIVE	POSITIVE	AND	LOW

A low distance with a negative relative speed and negative relative angle angle was chosen primarily because of proximity. Rule 16 was chosen as critical because the closure rate is positive between the robot and an obstacle and the angle is directed toward it giving it a higher chance for collision.. Rules 17, 19, and 31 were similar to the rationale for rule 1 with only small changes in range and relative angle.

V. SIMULATION SETUP

The setup of the simulation consists of the previously detailed Simulink robot controller in combination with the Gazebo collection of open libraries. The overall architecture of the simulation is detailed in figure 12.

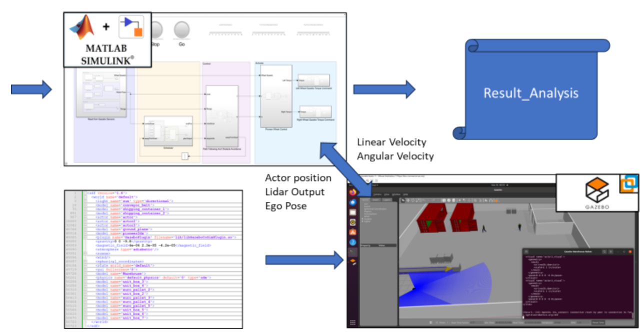

Fig. 12. Simulation Architecture.

The Simulink Controller provides the position and pose to the Gazebo simulation. Gazebo provides more than just a visual representation of the robot, however. It has a Lidar sensor emulator and can be visually shown in figure 13.

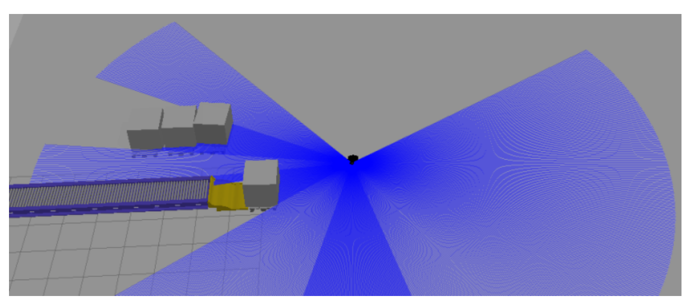

Fig. 13. Gazebo Lidar Scan.

The sensor is capable of emulating a, TODO (Emulation Details). The sensor output is sent back to the Simulink controller via UDP. Gazebo provides an Ubuntu VM to help to interface the environment with Simulink. Gazebo uses XML text files to create custom worlds. These files are used to define the parameters of the Occupancy grid, create static objects within the grid, and create both static and moving actors within the grid. Figure 14 shows the MATLAB representation of the environment created within the Gazebo world file. The world consists of multiple static objects from cargo containers to the walls of the environment.

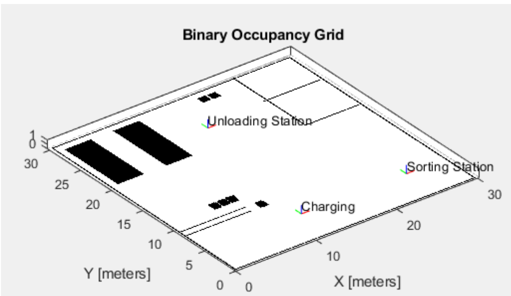

Fig. 14. Simulation Warehouse Environment.

A total of 7 scenarios were considered. Tablet VI details the Scenario, the number of actors in the scenario, and the key focus of it.

TABLE VI

Scenario	# of Actors	Key Focus
1) Baseline scenario	2	FLC tunning and implementation
2) Stationary actor starts to move away from the bot	3	Stop and go performance; target following performance
3) Stationary actor starts to move towards the bot	3	Critical scenario identification, urgent stopping
4) Stationary actor starts to move alongside the travel path of bot	3	No stopping required as the actor is not directly in path
5) Two actors crossing the path of robot perpendicularly, one after another	4	Target threat resolution
6) Two actors crossing the path of robot at an angle, one after another	4	Target threat resolution
7) Crowded environment - 1	6	Performance comparison between legacy control and FLC

SCENARIOS

The general trend for the scenarios was a build-up approach where more and more dynamically moving actors were added to the simulation. The first scenario was provided by the Simulink/Gazebo integration tutorial. The second scenario focused on stop and go performance and target following performance. The actor moving away from the robot helped to verify the fuzzy rules permit the robot to not be as influenced by an obstacle that is becoming less likely to interfere with the target path. The third scenario tests the opposite to verify the controller can determine a significant threat of collision. The fourth scenario is used to verify that false positive collisions do not occur. The fifth scenario is a simple stress test to verify the controller is actively tracking obstacles that occur one after another. The sixth scenario is similar to the fifth, but the angle was adjusted to directly target the relative angle membership function of the controller. Lastly, the seventh scenario was a full stress test by adding the most actors to the scenario and is used for the final comparison between the traditional and our FLC.

VI. Results

A. Scenario 1

The FLC was slightly faster than the traditional controller. The FLC appeared to stop at the same locations in addition to one other stop. The FLC spent less time not moving which increased its overall speed throughout the run Notably, the FLC and Traditional Controller turned opposite directions at the 35 second mark of the simulation which is the primary reason why the FLC was quicker. However, the FLC also got close to colliding, but managed to stop before contact. Figure 15 details the FLC vs Traditional Controller for Relative Range, Relative Speed, Relative Angle, and NoStop signals.

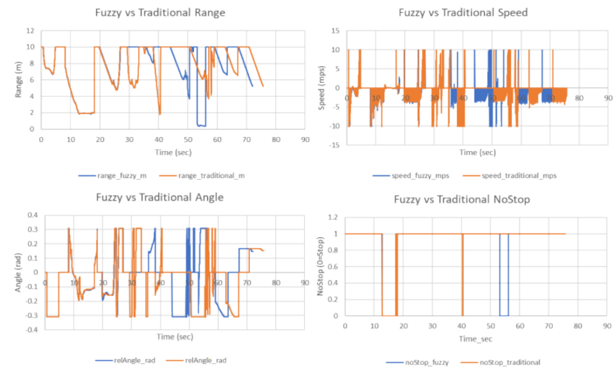

Fig. 15. Scenario 1 Results Plots.

B. Scenario 2

The FLC took roughly the same amount of time as the traditional controller to reach the goal. The responses of the controllers were very similar. Both avoided the obstacles at nearly the same distance. There were only minor differences between the relative angle and relative speed of the controllers. Figure 16 details the FLC vs Traditional Controller.

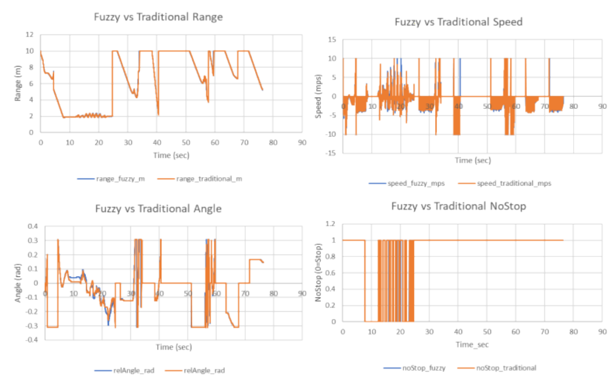

Fig. 16. Scenario 2 Results Plots.

C. Scenario 3

The FLC took nearly the same amount of time as the traditional controller to reach the goal. The responses of the controllers were very similar. In this scenario, both controllers collided with an object in the simulation. There were only minor differences between the relative angle and relative speed of the robots. The traditional controller's path did not appear to be influenced to the same degree as the FLC to the moving actors between 15 and 30 seconds into the simulation.

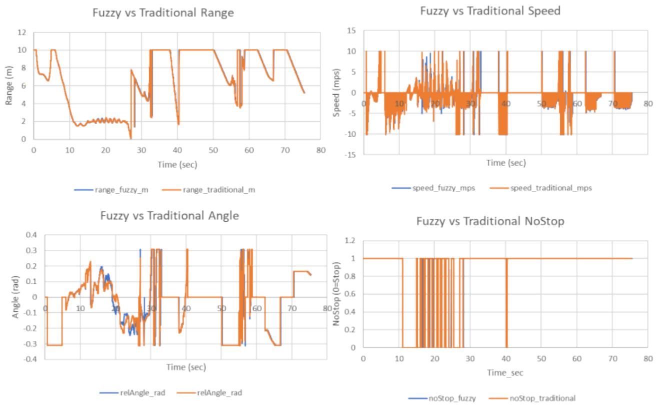

Fig. 17. Scenario 3 Results Plots.

D. Scenario 4

The differences between the controllers were nearly identical. The FLC took a wider turn around 55 seconds which resulted in the FLC solution reaching the goal sooner, but by only .17 seconds which is small enough to not be notable. Figure 18 shows the Fuzzy vs Traditional controller plots.

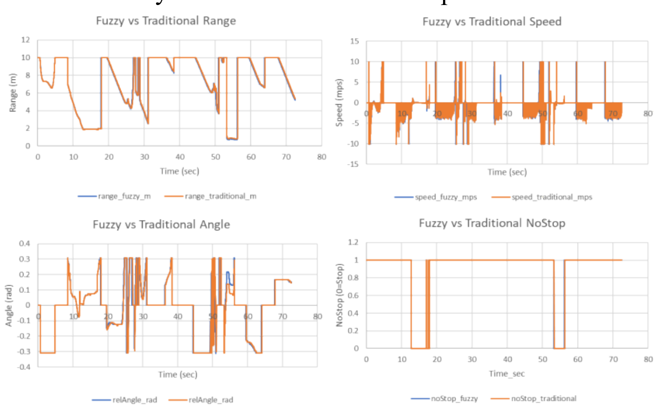

Fig. 18. Scenario 4 Results Plots.

E. Scenario 5

The FLC completed the route quicker than the traditional controller. The FLC took an opposite turn as compared to the traditional controller around 35 seconds into the simulation. As a result, the FLC was close to colliding with an obstacle as seen between 50 and 60 seconds. The FLC needed to make an additional stop between 50 to 60 seconds into the simulation. Figure 19 shows the Fuzzy vs Traditional controller plots.

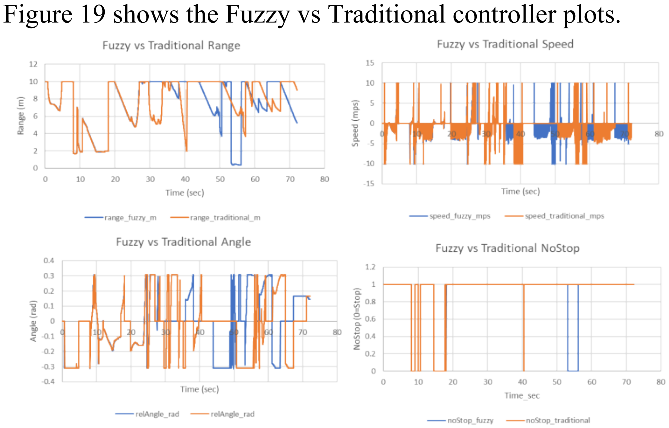

Fig. 19. Scenario 5 Results Plots.

F. Scenario 6

The controllers performed nearly identically. The shapes of each of the plots are the same and there are only minor deviations with the FLC's speed and angle being only slightly larger. Figure 20 shows the Fuzzy vs Traditional controller plots.

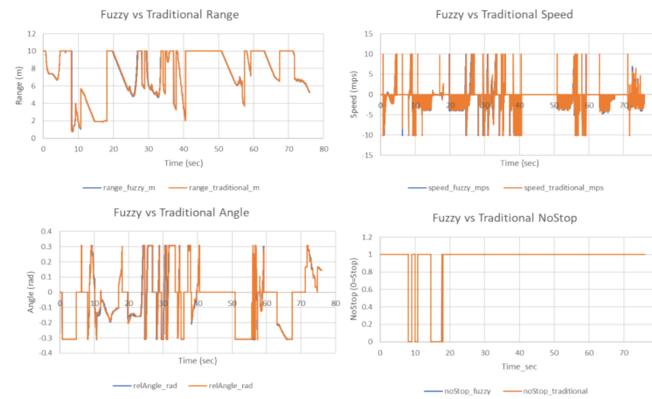

Fig. 20. Scenario 6 Results Plots.

G. Scenario 7

The FLC took longer than the traditional controller to reach the goal. Despite this longer path, the traditional controller appeared to hit an obstacle as the relative range reached 0 to an obstacle. The FLC spent more time turning to avoid obstacles, specifically around the 60 to 70 second mark which is where the traditional controller crashed. The FLC is shown to stop more times than the traditional controller. Figure 21 shows the Fuzzy vs Traditional controller plots.

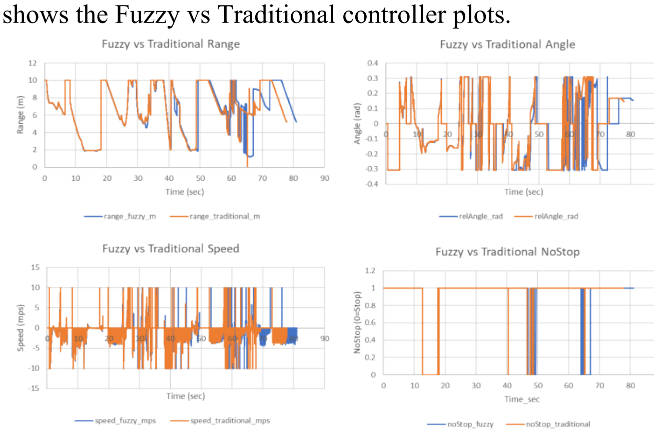

Fig. 21. Scenario 7 Results Plots.

VII. CONCLUSION

From the presented results it can be concluded that in most cases the Fuzzy Logic Controller and Traditional Controller have very similar performance. But in one of the scenarios, the Traditional Controller hit an obstacle where the Fuzzy Logic Controller could safely avoid hitting the same obstacle but reached the target location later. This particular scenario confirms that the Fuzzy Logic Controller is designed to prioritize safety rather than performance.

REFERENCES

[1] Khatib O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. The International Journal of Robotics Research. 1986;5(1):90-98. doi:10.1177/027836498600500106

[2] Min Gyu Park, Jae Hyun Jeon and Min Cheol Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing," ISIE 2001. 2001 IEEE International Symposium on Industrial Electronics Proceedings (Cat. No.01TH8570), Pusan, South Korea, 2001, pp. 1530-1535 vol.3, doi: 10.1109/ISIE.2001.931933.

[3] Enxiu Shi, Tao Cai, Changlin He and Junjie Guo, "Study of the New Method for Improving Artificial Potential Field in Mobile Robot Obstacle Avoidance," 2007 IEEE International Conference on Automation and Logistics, Jinan, China, 2007, pp. 282-286, doi: 10.1109/ICAL.2007.4338572.

[4] H. Li, Z. Wang and Y. Ou, "Obstacle Avoidance of Manipulators Based on Improved Artificial Potential Field Method," 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 2019, pp. 564-569, doi: 10.1109/ROBIO49542.2019.8961506.

[5] R. Kala, A. Shukla, R. Tiwari, S. Rungta and R. R. Janghel, "Mobile Robot Navigation Control in Moving Obstacle Environment Using Genetic Algorithm, Artificial Neural Networks and A* Algorithm," 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 705-713, doi: 10.1109/CSIE.2009.854.

[6] Elhoseny, Mohamed, Shehab, Abdulaziz, and Yuan, Xiaohui. 'Optimizing Robot Path in Dynamic Environments Using Genetic Algorithm and Bezier Curve'. 1 Jan. 2017 : 2305 – 2316.

[7] M. Gao, J. Xu, J. Tian and H. Wu, "Path Planning for Mobile Robot Based on Chaos Genetic Algorithm," 2008 Fourth International Conference on Natural Computation, Jinan, China, 2008, pp. 409-413, doi: 10.1109/ICNC.2008.627.

[8] Baofang Wang, Sheng Li, Jian Guo, Qingwei Chen, Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm, Neurocomputing, Volume 282, 2018, Pages 42-51, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2017.12.015.

[9] P.K. Das, H.S. Behera, Swagatam Das, H.K. Tripathy, B.K. Panigrahi, S.K. Pradhan, A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment, Neurocomputing, Volume 207, 2016, Pages 735-753, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2016.05.057.

[10] Michael Morin, Irène Abi-Zeid, Claude-Guy Quimper, Ant colony optimization for path planning in search and rescue operations, European Journal of Operational Research, Volume 305, Issue 1, 2023, Pages 53-63, ISSN 0377-2217, https://doi.org/10.1016/j.ejor.2022.06.019.

[11] Feng, S.; Sebastian, B.; Ben-Tzvi, P. A Collision Avoidance Method Based on Deep Reinforcement Learning. Robotics 2021, 10, 73. https://doi.org/10.3390/robotics10020073

[12] Xiuqing Wang, Zeng-Guang Hou, Anmin Zou, Min Tan, Long Cheng, A behavior controller based on spiking neural networks for mobile robots, Neurocomputing, Volume 71, Issues 4–6, 2008, Pages 655-666, ISSN 0925-2312, https://doi.org/10.1016/j.neucom.2007.08.025.

[13] L.A. Zadeh, The concept of a linguistic variable and its application to approximate reasoning-III, Information Sciences, Volume 9, Issue 1, 1975, Pages 43-80, ISSN 0020-0255, https://doi.org/10.1016/0020-0255(75)90017-1.

[14] A. Mechernene, V. Judalet, A. Chaibet and M. Boukhnifer, "Lane Change Decision Algorithm based on Risk Prediction and Fuzzy Logic Method", 2021 25th International Conference on System Theory, Control and Computing (ICSTCC), Iasi, Romania, 2021, pp. 707-713, doi: 10.1109/ICSTCC52150.2021.9607228.

[15] Faisal M, Hedjar R, Al Sulaiman M, Al-Mutib K., "Fuzzy Logic Navigation and Obstacle Avoidance by a Mobile Robot in an Unknown Dynamic Environment", International Journal of Advanced Robotic Systems. 2013;10(1). doi:10.5772/54427

[16] B. K. Oleiwi, A. Mahfuz and H. Roth, "Application of Fuzzy Logic for Collision Avoidance of Mobile Robots in Dynamic-Indoor Environments," 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), DHAKA, Bangladesh, 2021, pp. 131-136, doi: 10.1109/ICREST51555.2021.9331072.

[17] Yosif, Z.M., Mahmood, B.S., Saeed, S.Z. (2022). Artificial techniques based on neural network and fuzzy logic combination approach for avoiding dynamic obstacles. Journal Européen des Systèmes Automatisés, Vol. 55, No. 3, pp. 339-348. https://doi.org/10.18280/jesa.550306v

[18] S. Hu, C. Cao and J. Pan, "Deep-learned pedestrian avoidance policy for robot navigation," 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 2017, pp. 338-343, doi: 10.1109/ROBIO.2017.8324440.

[19] L. E. Zarate, M. Becker, B. D. M. Garrido and H. S. C. Rocha, "An artificial neural network structure able to obstacle avoidance behavior used in mobile robots," IEEE 2002 28th Annual Conference of the Industrial Electronics Society. IECON 02, Seville, Spain, 2002, pp. 2457-2461 vol.3, doi: 10.1109/IECON.2002.1185358.

[20] R. Kala, A. Shukla, R. Tiwari, S. Rungta and R. R. Janghel, "Mobile Robot Navigation Control in Moving Obstacle Environment Using Genetic Algorithm, Artificial Neural Networks and A* Algorithm," 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 705-713, doi: 10.1109/CSIE.2009.854.

[21] B. Li, J. Chang and C. Wu, "A potential function and artificial neural network for path planning in dynamic environments based on self-reconfigurable mobile robot system," 2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), College Station,

TX, USA, 2012, pp. 1-6, doi: 10.1109/SSRR.2012.6523900.
[22] X. Zhang, A. Liniger and F. Borrelli, "Optimization-Based Collision Avoidance," in IEEE Transactions on Control Systems Technology, vol. 29, no. 3, pp. 972-983, May 2021, doi: 10.1109/TCST.2019.2949540.