PATH PLANNING AND OPTIMIZATION OF DIFFERENTIAL DRIVE WHEEL USING FUZZY LOGIC

A Minor Project Report

submitted in partial fulfillment of the
requirements of VII-Semester for the degree
of
Bachelor of Technology
in

COMPUTER SCIENCE & ENGINEERING

by

K. Raj Kamal (Reg. No. 16115030)

G. Jayanth Kumar (Reg. No. 16115022)

K. Gautham (Reg. No. 16115029)

Under the guidance of

Dr. Preeti Chandrakar

Assistant Professor

NIT-RAIPUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
RAIPUR, CG (INDIA)

NOVEMBER 2019

DECLARATION

I hereby declare that the work described in this thesis, entitled **“PATH PLANNING AND OPTIMIZATION OF DIFFERENTIAL DRIVE WHEEL USING FUZZY LOGIC”** which is being submitted by me in partial fulfillment for the VII-Semester of the degree of Bachelor of Technology in the Department of **Computer Science and Engineering** to the National Institute of Technology Raipur is the result of investigations carried out by me under the guidance of **Dr. Preeti Chandrakar**

The work is original and has not been submitted for any Degree/Diploma of this or any other Institute/university.

Signature

Name of the Candidate

G. Jayanth Kumar
Roll No.: 16115022
K. Raj Kamal
Roll No :16115030
K. Gautham
Roll No: 16115029

Place: Raipur,

Date: 23-11-2019.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY

CERTIFICATE

This is to certify that the project entitled **"PATH PLANNING AND OPTIMIZATION OF DIFFERENTIAL DRIVE WHEEL USING FUZZY LOGIC",** that is being submitted by **K. Raj Kamal (Roll No. 16115030), G. Jayanth (Roll No. 16115022), K. Gautham (Roll No. 16115029)** in partial fulfilment for VII-Semester of the degree of **Bachelor of Technology in Computer Science & Engineering** to National Institute of Technology Raipur is a record of bonafide work carried out by them under my guidance and supervision.

The matter presented in this project document has not been submitted by her for the award of any other degree elsewhere.

Signature of Supervisor
Dr. Preeti Chandrakar
Assistant Professor
Department of Computer Science & Engineering
National Institute of Technology, Raipur (CG.)

Project Coordinator
Dr. K. Jairam Naik
Department of Computer Science & Engineering
National Institute of Technology, Raipur (CG.)

H.O.D
Dr. Pradeep Singh
Department of C.S.E
NIT Raipur (CG.)

ACKNOWLEDGEMENT

We would like to acknowledge my college **National Institute of Technology, Raipur** for providing a holistic environment that nurtures creativity and research-based activities.

We express our sincere thanks to **Dr. Preeti Chandrakar, Assistant Professor CSE Department**, **NIT Raipur,** the guide of the project for guiding and correcting throughout the process with attention and care. She has frequently suggested our creative ideas and guided through the major hurdles that occurred during the duration of the project.

We would also thank **Dr. Pradeep Singh, Head of the department** and all the faculty members without whom this project would be a distant reality. We also extend my heartfelt thanks to our family and friends who supported us.

G. Jayanth Kumar
Roll No.: 16115022

K. Raj Kamal
Roll No: 16115030

K. Gautham
Roll No: 16115029

ABSTRACT

. The path planning is a task which is defined as the process of computation of the motion sequence enabling the robot from one position to another position automatically without any human actions. Robot uses sensors to gather information from the environment and passes this input to some unit to find the next position of the robot. There is a lot of imprecision and uncertainty in the path planning implementation using fuzzy logic . “Fuzzy logic is an approach to computing based on degree of truth rather than the usual true or false. We aggregate data and form a number of partial truths which we aggregate further into higher truths which in turn, when certain thresholds are exceeded, cause certain further results such as motor reactions”.

The WDO (Wind Driven Optimization) algorithm is working based on the “atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain”. .We are going to propose an approach for designing optimized fuzzy controller using WDO which allows the robot to move in both dynamic and static environments.

In this approach we are designing the pipeline so that the robot will be able to sustain in a dynamic environment in which the robot doesn’t have any prior knowledge about obstacles and environmental conditions.

We are implementing the designed approach through Pioneer 3DX Robot which is a commonly used differential drive robots with 8 sensors and a Caster wheel to balance the robot. We are simulating this robot in a virtual environment designed using Simulink. The possible states of the robot are represented using the State Space Structures in simulink. Unlike many other models available today we are not just predicting the velocity but in turn optimizing it based on the Environmental conditions through Wind-Driven Optimization (WDO).

CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Path planning is a primitive for an autonomous mobile robot that lets the robots find the shortest path between two points and avoid the obstacles. The path planning is a task which is defined as the process of computation of the motion sequence enabling the robot from one position to another position automatically without any human actions. Robot uses sensors to gather information from the environment and passes this input to some unit to find the next position of the robot. Fuzzy logic System(FLS) is used here to find the path planning between source and destination. By considering the dynamic environment this path is optimized by using WDO (Wind Driven Optimization)[1]. WDO is used to optimize and tune the velocities that we get from Fuzzy System. The WDO is an algorithm working based on the "atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain".

1.2 OBJECTIVES AND IMPORTANCE OF PROJECT

Path planning is a primitive for an autonomous mobile robot that lets the robots find the shortest path between two points and avoid the obstacles.

The main objectives are:

To get the estimated velocities of the two motors (left and right).

To optimize the estimated velocities using WDO algorithm.

1.3 SCOPE AND MOTIVATION

Path planning is a primitive for autonomous mobile robot. Efficiency, Accuracy and Safety are the three important concerns which should be kept in mind regarding robot navigation problems .The most important factor among this is efficiency since destination should be reached in shortest path and time.

There is a lot of imprecision and uncertainty in path planning algorithms since they are not accurate. Some algorithms for path planning are provided for just static environment and not applicable when the obstacles are moving. In the case of dynamic environment, the approach should be different since the environment is changing by time.

1.4 ORGANIZATION OF THE PROJECT REPORT

The Overall project has been explained in 7 chapters. Each chapter has been considered as an individual component contributing to the overall project. A brief introduction of all the chapters is as follows:

In the first chapter we explained the importance, Objectives, Scope and Motivation of the project. In the second chapter we explained the literature used along with comparisons with various methods and models currently existing.

In the third chapter we basically described the techniques we followed such as Fuzzy logic including brief introduction about Fuzzy logic and its components such as membership functions, Fuzzification and Defuzzification., Wind-Driven Optimization and we explained the pseudo code of the algorithm along with various forces being considered.

In the fourth chapter we explained the Architecture of the pipeline along with Technical Specifications such as Robot and Sensor Configuration used to simulate the approach. We discussed the procedure of the velocity prediction and its optimization. Finally, we explained the end to end pipeline using the above-mentioned techniques and the acquired results are taken through simulation in Matlab and Simulink are also presented in subsequent chapters.

In next chapter we described the Environmental setup such as configurations of the computing machine and technologies used such as MATLAB, Simulink, SolidWorks that are required for simulating the model. For certain binary maps we shown the navigated path from given source to destination in the simulation results. In performance evaluation subchapter we discussed the metrics being used to evaluate the model along with their importance and we are also comparing the results of the proposed approach with currently existing techniques such as Genetic Algorithm.

In next chapters we concluded the project and with stating the future work and other opportunities available to extend the proposed approach.

CHAPTER-2

LITERATURE OVERVIEW

2.1 OVERVIEW

Path planning and control of an autonomous mobile robot in an unknown dynamic environment is one of the most challenging jobs. Fuzzy logic is one of the most used techniques for path planning of mobile robot between source and destination. Our main aim to achieve path planning in both static and dynamic environments. This is done by getting input from sensors of robot and passing these inputs to fuzzy system and predicting left and right motor velocities and these velocities are being optimized using WDO algorithm.

In a paper published in 2014 by Pandey and Parhi this algorithm along with angle parameter considers use the ensemble model fuzzy system to predict the velocity. This approach uses the particle swarm optimization which is a kind of outdated and performs poorly than Wind-Driven Optimization.[3].

In a paper published in 2012 path planning is done by calculating the distance of left obstacle, right obstacle and front obstacle along with angle as parameter. Since it considers angle as parameter it gives the better predicted velocity. But this fails to optimize the path in dynamic environment.[2]

In a paper published in 2017 path planning algorithm is developed by considering left obstacle, right obstacle and front obstacle distance as parameter and this algorithm is even stable in dynamic environments. This algorithm won't emphasize much on path planning. This mainly focuses on optimizing using Wind-Driven Optimization and does not consider angle as parameter.[1].

In the article published by Zhai et al in 2019 path planning is done by considering distance of the obstacle from left, right and front. This algorithm makes use of the Optimal Transport theory and Fokker-Plank Equation. It doesn't use any optimization algorithm to optimize the path further.[6].

In the article published by Khaksar, Torreson. The distance of the obstacle from the front, left, right are assumed as parameters. Along with Domain-specific inputs labeled data is also taken. The algorithm is Ensemble model of both Fuzzy and Neural network [8].

In the article published by Saraswathi, Bala Murali in 2018 domain-specific heuristics, and sensor data for next obstacle distance are considered. This algorithm makes use of advantages of both the bat algorithm and the Cuckoo search algorithm. It won't make use of any other optimization algorithm for handling local minima.[9]

In the article "Motion Planning of the mobile robot using Fuzzy-GA method along with three path concept in dynamic environment" published by Singh, KM Singh in 2018, the distance of the obstacle from the front, left, right were taken as parameters and the environment in the form of a bit map. This method combines the Fuzzy logic with the Genetic Algorithm.[11]

Nguyen, Trangerq published an article "A study on Building Optimal path planning algorithms for mobile robot navigation" in 2017 which considers input parameters such as next obstacle distance, domain-Specific heuristic knowledge and labeled data. It compares the various algorithms and their advantages and disadvantages.[12]

2.2 SUMMARY OF LITERATURE

In summary we explained the end to end pipeline including velocity prediction and optimization using the Fuzzy Inference System (FIS) and Wind Driven Optimization (WDO) and we also present the simulated results acquired through simulation of above techniques in MATLAB and Simulink. Through the below table the current advancements along with their results can be seen.

In the below table the current advancements in the field of path planning and optimization are mentioned along with the authors and published date.

S.NO	Paper Title	Authors	Year of Publication	Parameters considered	Advantages	Limitations
1.	“Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm”	Anish Pandey , Dayal R. Parhi	2017	Distance of the obstacle from front, left, right.	The algorithm is stable even in a dynamic environment.	This approach won’t emphasize much on the path planning, this mainly focuses on optimizing the path using Wind Driven Optimization. And this won’t consider the angle parameter.
2.	“A Fuzzy-Logic-Based Approach for Mobile Robot Path Tracking”	Gianluca Antonelli, Et al.	2007	Distance of the obstacle from front, left, right along with angle parameter.	This algorithm considers the angle parameter, so this performs better velocity prediction.	Fails to optimize the path(velocity) in a dynamic environment.

3.	"Fuzzy logic-based robot path planning In unknown environment ."	MENG, et al.	2014		This algorithm along with the angle parameter considers uses the ensemble model fuzzy system to predict the velocity.	This approach uses the particle swarm optimization, which is kind of outdated and performs poorly than the Wind Driven optimization.
4	"Path Planning in Unknown Environments Using Optimal Transport Theory"	Zhai, Haoyan, Egerstedt	2019	The distance of the obstacle from the front, left, right.	Makes use of the Optimal Transport theory and Fokker-Plank Equation Uses velocity optimization techniques	It doesn't use any optimization algorithm to optimize the path further.
5	"Advancement in navigational path planning of robots using various artificial and computing techniques"	Parhi	2018	Review Paper	Compares the current work done in the field of mobile robot navigation	Compares the current work done in the field of mobile robot navigation
6	"Neuro-Fuzzy sampling: Safe and fast multi-query randomized path planning for mobile robots"	Khaksar, Torreson	2018	The distance of the obstacle from the front, left, right. Domain-specific inputs along with labeled data.	Ensemble model of both Fuzzy and Neural network	Won't use any optimization algorithm to further optimize the path and velocities.

7	“Optimal path planning of mobile robot using the hybrid Cuckoo search and bat algorithm”	Saraswathi, Bala Murali	2018	Domain-specific heuristics, and sensor data for next obstacle distance.	Makes use of advantages of both the bat algorithm and the Cuckoo search algorithm	It won’t make use of any other optimization algorithm for handling local minima.
8	“Improved Fuzzy Logic rules and effective searching box method for mobile robot navigation”	Liu, Yuan, Chen.	2018	The distance of the obstacle from the front, left, right. Labeled data.	This approach makes use of ensemble modeling of both fuzzy logic and neural network	This approach won’t use any other optimization algorithm to optimize predicted velocity
9	“Motion Planning of the mobile robot using Fuzzy-GA method along with three path concept in dynamic environment”	Singh, KM singh	2018	The distance of the obstacle from the front, left, right and the environment in the form of a bit map	This combines both the Fuzzy logic with the Genetic Algorithm	------
10	“A study on Building Optimal path planning algorithms for mobile robot navigation”	Nguyen, Trangerq	2017	Considers input parameters such as next obstacle distance, domain-Specific heuristic knowledge and labeled data.	Review Paper - Compares the various algorithms and their advantages	Review Paper - Compares the various algorithms and their disadvantages

Table 2.1 Summary of Literature

CHAPTER - 3

TECHNIQUES USED

3.1 Fuzzy Logic

Fuzzy Logic is similar to human decision taking pattern and deals with the information which is uncertain. “Any event, process, or function that is changing continuously cannot always be defined as either it is true or false”, that means we should define such activities in a fuzzy manner.

“It is a gross oversimplification of the real-world problems and based on degrees of truth rather than usual true/false or Boolean logic.”

The fuzzy logic is used in the fields of Aerospace, automotive, defence, electronics, transportation, pattern recognition and classification.

The Fuzziness of the fuzzy logic can be known by its membership function. membership function better determines the degree of truthiness of fuzzy logic. We group the data given and form number of partial truths which results in the motor reaction.

3.1.1 MEMBERSHIP FUNCTIONS

Fuzziness of a fuzzy system is better defined by its membership functions. It represents the degree of truth in fuzzy logic. Here are some important points related to membership function:

1. Fuzziness is grouped by member functions, whether the elements in fuzzy sets are not continuous or continuous .
2. Membership functions are those functions which can be defined as a technique to solve practical problems by considering experience rather than knowledge.
3. They are represented in graphical forms.
4. “Rules for defining fuzziness are fuzzy too”.

It is mathematically represented as:

$$\tilde{A} = \{(y, \mu_{\tilde{A}}(y)) \mid y \in U\} \quad [13]$$

3.1.2 FUZZY RULES SET

Fuzzy rules are those which are used to infer an output based on input variables.

Table 1
Fuzzy rules set.

If (d_f is Far) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Far) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Near) **and** (d_r is Far) **then** (m_r is Low) **and** (m_l is High)
If (d_f is Near) **and** (d_l is Far) **and** (d_r is Near) **then** (m_r is High) **and** (m_l is Low)
If (d_f is Far) **and** (d_l is Near) **and** (d_r is Near) **then** (m_r is Low) **and** (m_l is High)

Table 3.1 Rule Set [1]

Here

1. Pre-processing the raw data, read from sensors. (dl, dr, df)

Where,

dl – Distance of the next obstacle on left

dr – Distance of the next obstacle on right

df – Distance of the next obstacle on front

2. Evaluating the inputs(dl, dr, df) on the ruleset of Fuzzy Logic.
3. Estimating the velocities (ml, mr) using Fuzzy Logic.

Where,

ml – left motor velocity

mr – right motor velocity

3.1.3 FUZZY INFERENCE SYSTEM

The main system of fuzzy logic is fuzzy inference system. Its primary work is decision making.

The main characteristics of fuzzy system are

1.the output is fuzzy set which is independent of input is whether fuzzy or crisp.

2.There should be a fuzzy output when it is used like controller.

3.There should be a defuzzification unit with fuzzy system to modify the fuzzy set into crisp values

The main blocks of fuzzy inference system [13]

1.Rule Base

2.DataBase

3.Decision-making unit

4. “Fuzzification Interface unit: It converts the fuzzy values into crisp values.”

5. “Defuzzification Interface unit: It converts the crisp values into fuzzy values.”

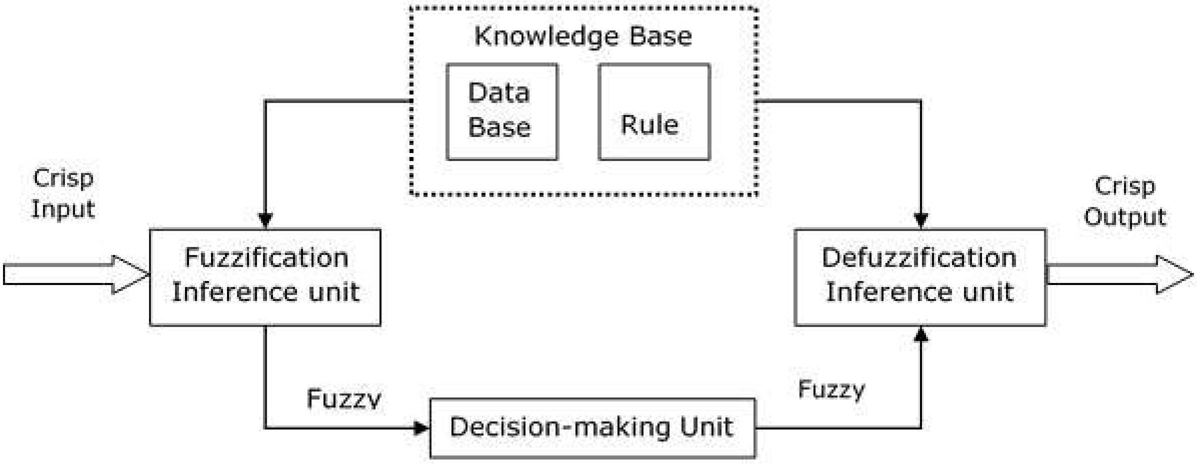


Figure 3.1 Fuzzy Inference System [13]

3.2 WIND-DRIVEN OPTIMIZATION

3.2.1 OVERVIEW

WDO is a new globally accepted particle-based optimization on nature inspired atmospheric motions. "A population of very huge amount of small air packets navigates over a n-dimensional object following Newton's second law of motion, this is also used to describe the motion of air parcels within the earth's atmosphere". The WDO (Wind Driven Optimization) [1] algorithm primarily works on the principle of Newton's second law.

Newton's second law: "It states the air parcel accelerates in the direction with an acceleration in the direction of resultant force applied."

$\rho.a = \sum Fi$

These forces mainly consists of:

1. Gravitational force.

2. Pressure gradient force.

3. Frictional force.

4. Corolis force.

3.2.2 Pseudo code of the WDO algorithm

1.Start

2.Set the group of air packet, number of dimensions for the optimization problem, coefficients, pressure function, maximum number of iterations, upper and lower boundaries of optimization problem.

3.Determine a position(random) and velocity of the air parcels.

4.Analyze the fitness(pressure) values of each air parcel, at its current position respectively.

5.After the evaluation of the expected pressure values, the ranking of population is done in ascending order of these pressure values and the velocity is updated according to the equation along with the restrictions given.

6.Revise the position of air parcel for the next iteration and the boundaries of the air parcel are checked.

7.After achieving maximum number of iterations the process is terminated, else go to step 4.

3.2.3 Forces Acting:

The forces acting on the robot mainly consists of

1.Gravitational Force.

2.Frictional Force.

3.Corolis Force.

4.Pressure Gradient Force.

The equations of the forces are:

$$F_{PG} = -\nabla P \cdot \delta V \qquad F_G = \rho \cdot \delta V \cdot g$$

$$F_F = -\rho \cdot \alpha \cdot u \qquad F_C = -2 \cdot \Omega \cdot u$$

The wind driven optimization works on the principle of Newton's second law, which is:

$$\rho \cdot a = \sum F_i$$

Substituting the forces in Newton's second law:

$$\rho \cdot \frac{\Delta u}{\Delta t} = (\rho \cdot \delta V \cdot g) + (-\nabla P \cdot \delta V) + (-\rho \cdot \alpha \cdot u) + (-2 \cdot \Omega \cdot u)$$

Now in order to consider the temperature of the room, consider the ideal gas equation:

$$P = \rho RT$$

Now combining the above equations:

$$\rho \cdot \Delta u = (\rho \cdot g) + (-\nabla P) + (-\rho \cdot \alpha \cdot u) + (-2 \cdot \Omega \cdot u)$$

$$u_{\text{new}} = (1-\alpha) \cdot u_{\text{cur}} - g \cdot x_{\text{cur}} + \left(RT \left| \frac{1}{i} - 1 \right| (x_{\text{opt}} - x_{\text{cur}}) \right) + \left(\frac{c \cdot u_{\text{cur}}^{\text{other dim}}}{i} \right)$$

The generated U is in a single iteration, this goes on for n number of iterations:

$$x_{\text{new}} = x_{\text{cur}} + (u_{\text{new}} \cdot \Delta t)$$

The generated velocity is added to the previous step as above.

The generated velocity should not be more than the allowed maximum velocity, if it’s more then , replace it with the maximum specified velocity.

$$u^{*}_{\text{new}} = \begin{cases} u_{\text{max}} & \text{if } u_{\text{new}} > u_{\text{max}} \\ -u_{\text{max}} & \text{if } u_{\text{new}} < -u_{\text{max}} \end{cases}$$

All the above mathematical equations have been taken from [1].

CHAPTER - 4

PROPOSED METHODOLOGY

4.1 Architecture

Block Diagram:

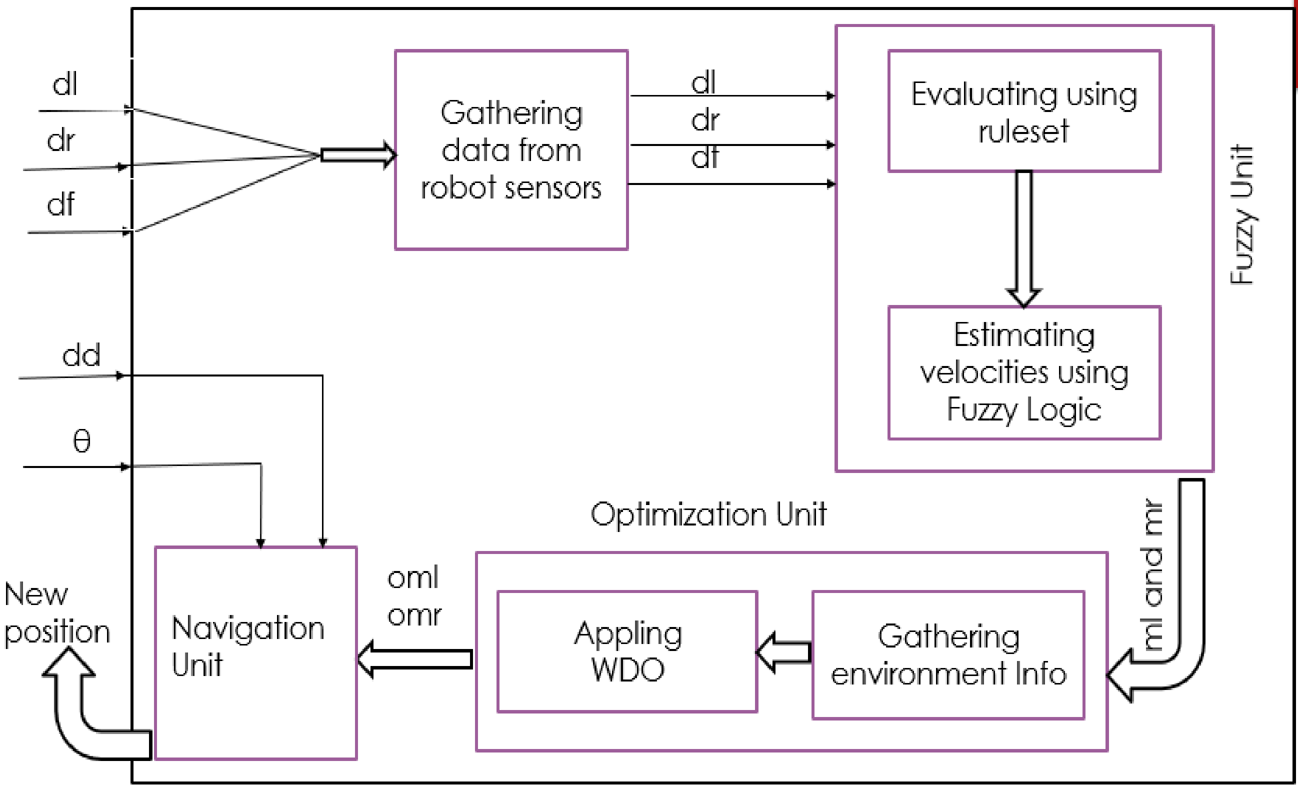


Fig 4.1 Architecture.

Pre-processing the raw data, read from sensors. (dl, dr, df)

Where,

dl – Distance of the next obstacle on left

dr – Distance of the next obstacle on right

df – Distance of the next obstacle on front

dd – Distance of robot from the destination

θ – Angle included between X-axis and destination

Apart from merely considering the next obstacle distances we are even considering at what angle those obstacles are varying according to current position of robot.

4.1.1 FUZZY UNIT:

Evaluating the inputs(dl,dr,df) on the ruleset of Fuzzy Logic.

Estimating the velocities (ml, mr) using Fuzzy Logic.

Where,

ml – left motor velocity

mr – right motor velocity

4.1.2 OPTIMIZATION UNIT

Gathering Environment Info:

The WDO (Wind Driven Optimization) algorithm primarily works on the principle of Newton's second law.

Newton's second law: It states the air parcel accelerates in the direction with an acceleration in the direction of resultant force applied.

$\rho.a = \sum Fi$

These forces mainly consists of:

1.Gravitational force

2.pressure gradient force

3. frictional force

4.Coriolis force

4.2 ROBOT AND SENSORS USED

The robot used here is Pioneer 3DX.It is a robot used for research and applications. It has two motors and 16 sensors(8 on front and 8 facing backward) . 8 sensors on the front align from 90 degrees to +90 degrees and the angle included between two consecutive sensors is 23.5 degrees.

Fig 4.2 PIONEER 3DX ROBOT

4.3 VELOCITY PREDICTION

The left and right motor velocities are predicted by passing inputs to Fuzzy Inference System (FIS). For every instance all input data is collected and passed to FIS.FIS uses rule set to predict two motor velocities. All input data include robot current position , 8 sensors data ,angle with X-axis and distance from destination.

4.4 VELOCITY OPTIMIZATION

By collecting robot current position, 8 sensors data, angle with X-axis and distance from destination we are predicting left and right motors velocity. These velocities are passed to WDO algorithm to get optimized left and right motor velocities.

4.5 END-END PIPELINE

This process of collecting the data from the sensors, predicting the velocities and optimizing the velocity is done continuously for every moment of the robot until it reaches its destination.

1. Initially we imported the CAD model of pioneer 3DX robot into matlab/simulink in xml format and performed unit testing to ensure the CAD model is working fine.

2. In this step we created the virtual environment of a room using Simulink.
3. We integrated the robot CAD model into the created virtual environment.
4. In the backend the Fuzzy inference system along with wind driven optimization for velocity prediction and velocity optimization is used.
5. From the front end in the virtual environment through the robot sensors the next left, front, right obstacle distances are gathered and sent back to the backend.
6. In the backend these are given to the Fuzzy Inference System and the velocity is predicted. This predicted velocity is given to the Wind Driven optimization to optimize the velocity according to the environmental conditions so that the robot can even sustain in the dynamic environment.
7. The optimized velocity from the previous step along with the pure pursuit algorithm is used to find the next position of the robot.
8. Steps 4 to 7 are repeated in a loop until destination is reached.

Code for initialization:

In this part of code we first specify the map to be read, source position and destination position ,initial heading direction of the robot ,length and breadth ,regular speed ,maximum speed ,safety distance of the robot ,threshold distance, maximum acceleration robot can attain.

The fuzzy outputs to turn are restricted to -1 and 1. These are magnified here. Maximum turn can be 60 degrees.

```
map=int16(imbinarize(imread('map2.bmp'))); % Specify the map to be read
source=[20 20]; % source position in Y, X format
goal=[480 480]; % goal position in Y, X format
robotDirection=pi/4; % initial heading direction of the robot
robotSize=[10 10]; %length and breadth(The robot dimensions)
robotSpeed=10; % specify the regular speed of the robot
maxRobotSpeed=10; % Specify the maximum speed of the robot
S=10; % safety distance
distanceThreshold=30; % a threshold distace. points within this threshold can be taken as same.
maxAcceleration=10; % maximum speed change per unit time
directionScaling=60*pi/180; % fuzzy outputs to turn are restriect to -1 and 1. these are magnifi
```

Fig 4.3 Initialization

Code for loading the fuzzy system, checking whether the destination is feasible or not, maximum speed of the robot, regular speed of the robot:

The goal point of the robot is not feasible if goal is out of range or goal is on obstacle. this piece of code is used to find whether destination is feasible or not.

```
fuz=readfis('fuzzyBase.fis'); % fuzzy inference system used.
distanceScaling=(size(map,1)^2+size(map,2)^2)^0.5; % all inputs are scaled by this number so th
currentPosition=source; % position of the centre of the robot
currentDirection=robotDirection; % direction of orientation of the robot
robotHalfDiagonalDistance=((robotSize(1)/2)^2+(robotSize(2)/2)^2)^0.5; % used for distance calc
pathFound=false; % has goal been reached
prevTurn=0; % preffered turn at the previous time step, used for turning heuristic, see variabl
prevDistanceLeftDiagonal=distanceScaling; % diagonal distance at the previous time step, used f
prevDistanceRightDiagonal=distanceScaling; % diagonal distance at the previous time step, used
pathCost=0;
t=1;
imshow(map==1);
rectangle('position',[1 1 size(map)-1],'edgecolor','k');
pathLength=0;
if ~plotRobot(currentPosition,currentDirection,map,robotHalfDiagonalDistance)
     error('source lies on an obstacle or outside map');
end
M(t)=getframe;
t=t+1;

if ~feasiblePoint(goal,map), error('goal lies on an obstacle or outside map'); end

tic;
```

Fig 4.4 Feasibility Check

The while loop to get inputs and predict next position and angle, until we reach destination

In the while loop inputs are read from 8 sensors, previous position of robot, inclination with x-axis and distance from destination and generates next position and angle of the robot until the goal is reached.

```
while ~pathFound

    % calculate distance from obstacle at front
    for i=robotSize(1)/2+1:distanceScaling
        x=int16(currentPosition+i*[sin(currentDirection) cos(currentDirection)]);
        if ~feasiblePoint(x,map), break; end
    end
    distanceFront=(i-robotSize(1)/2)/distanceScaling; % robotSize(1)/2 distance included in i was i

    % calculate distance from obstacle at front-left diagonal
    for i=robotHalfDiagonalDistance+1:distanceScaling
        x=int16(currentPosition+i*[sin(currentDirection-pi/4) cos(currentDirection-pi/4)]);
        if ~feasiblePoint(x,map), break; end
    end
    distanceFrontLeftDiagonal=(i-robotHalfDiagonalDistance)/distanceScaling;

    % calculate distance from obstacle at front-right diagonal
    for i=robotHalfDiagonalDistance+1:distanceScaling
        x=int16(currentPosition+i*[sin(currentDirection+pi/4) cos(currentDirection+pi/4)]);
        if ~feasiblePoint(x,map), break; end
    end
    distanceFrontRightDiagonal=(i-robotHalfDiagonalDistance)/distanceScaling;

    % calculate angle deviation to goal
     slopeGoal=atan2(goal(1)-currentPosition(1),goal(2)-currentPosition(2));
     angleGoal=slopeGoal-currentDirection;
     while angleGoal>pi, angleGoal=angleGoal-2*pi; end % check to get the angle between -pi and pi
```

Fig 4.5 End-End Pipeline

Code for plotting:

```
% plotting robot
if ~plotRobot(currentPosition,currentDirection,map,robotHalfDiagonalDistance)
    error('collission recorded');
end
M(t)=getframe;t=t+1;
```

Fig 4.6 Plotting

CHAPTER 5

ENVIRONMENTAL SETUP AND PERFORMANCE EVALUATION

5.1 ENVIRONMENTAL SETUP

Following are the environmental setup's used:

Laptop configuration:

RAM: 8GB,
ROM: 1TB.

Simulation tools used:

1. Matlab.
2. Simulink.
3. Solidworks.

The whole project was done in Matlab and Simulink.

1. The CAD model of the pioneer 3DX robot has been taken from the manufacturer's website and loaded into Solid works ide and exported as an XML file.
2. The exported xml CAD model is imported into Simulink/Matlab and unit tests are performed to check whether CAD model is working or not.
3. For the design of the virtual environment, we made use of state flow diagrams in Simulink.
4. To generate the final assemble model integrating the virtual environment with the cad model we made use of Simulink package library.
5. To generate binary maps we made use of built-in software in windows called paint.

5.2 SIMULATION RESULTS

Assemble Block diagram:

The assemble block diagram represents the overall workflow of the project, from importing the CAD model to simulation of results in the virtual environment. This contains the pioneer 3DX robot CAD model and the virtual environment along with the connecting subsystem.

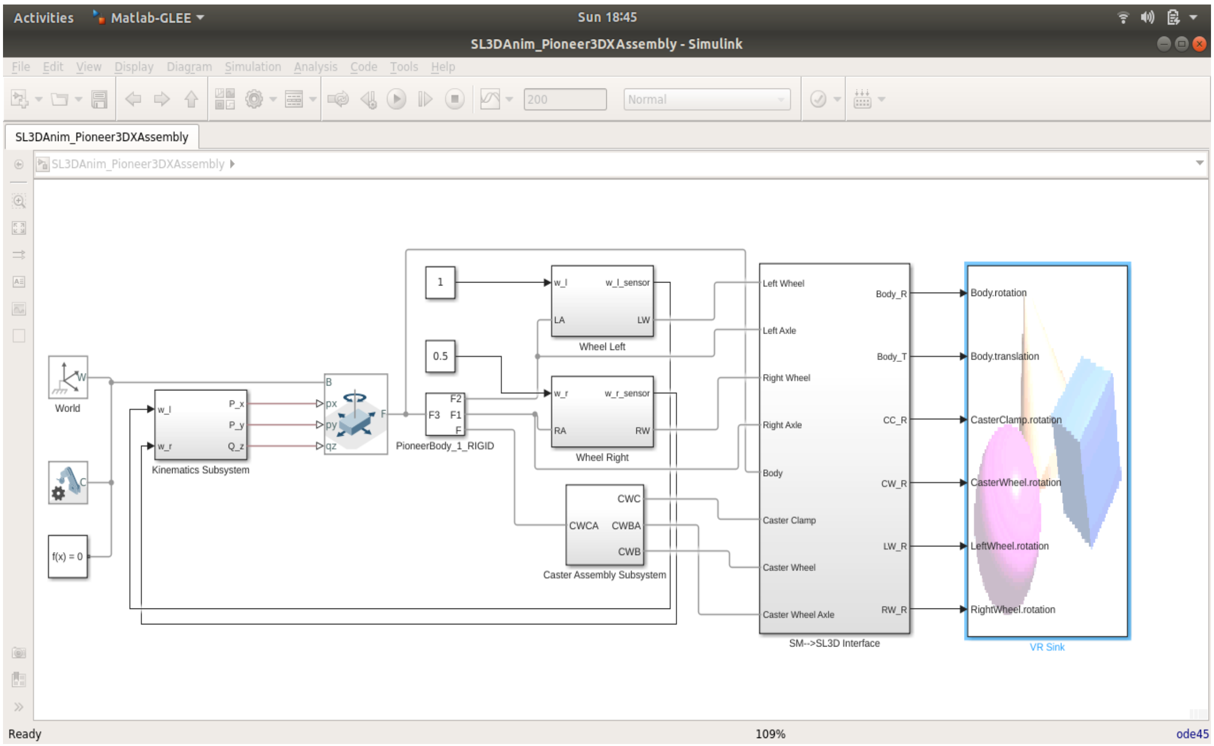


Fig 5.1 Assemble Block Diagram

State flow Diagrams:

The state flow diagrams are a high-level simulation tool available in simulink. The state flow represents all the available states of the robot in its course of action. The robot in order to move from one state to another state has to satisfy some constraints, these constraints are mentioned along the flowing arrow.

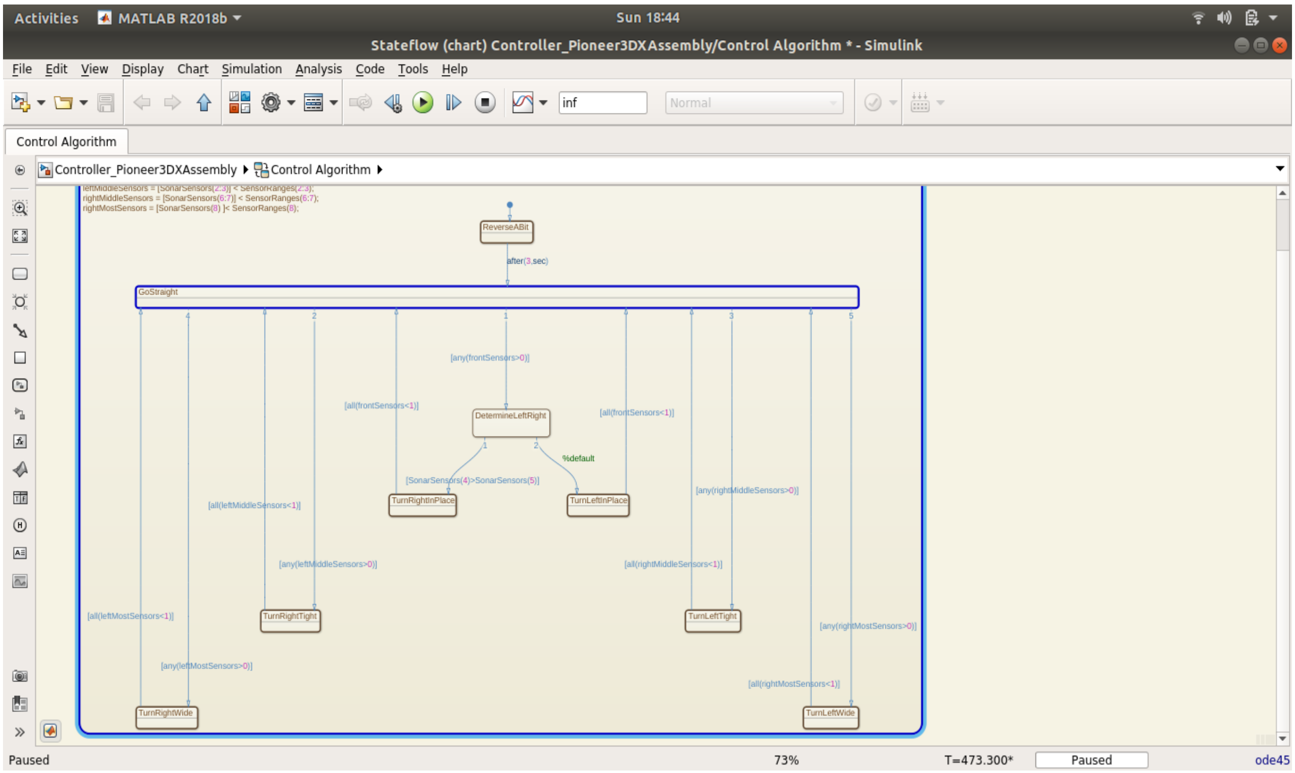


Fig 5.2 State Flow Diagram

Kinematic subsystem:

The kinematic subsystem takes input as the velocities and predicts the angular velocity along with the orientation angle and the next position of the robot.

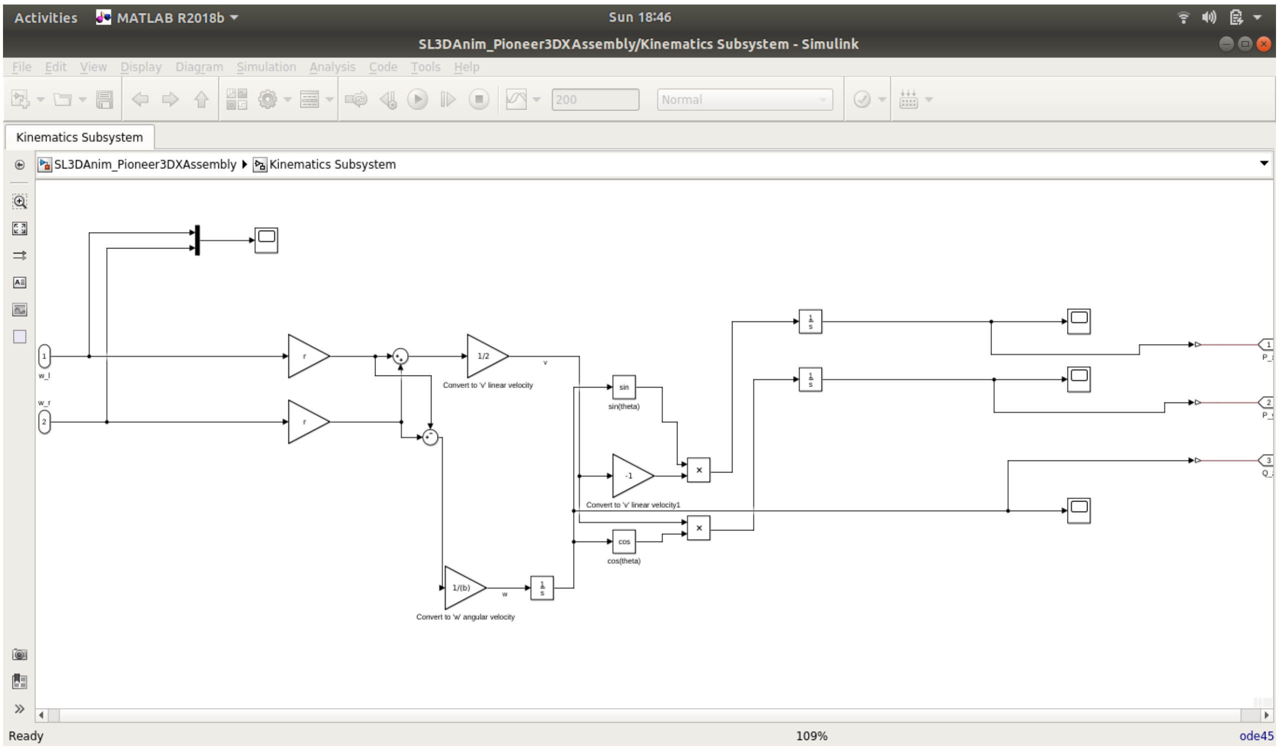


Fig 5.3 Kinetic Sub-System.

Virtual environment:

The virtual environment has been designed using Simulink, the designed virtual environment is a room, with a sofa and regular objects as obstacles.

Front view:

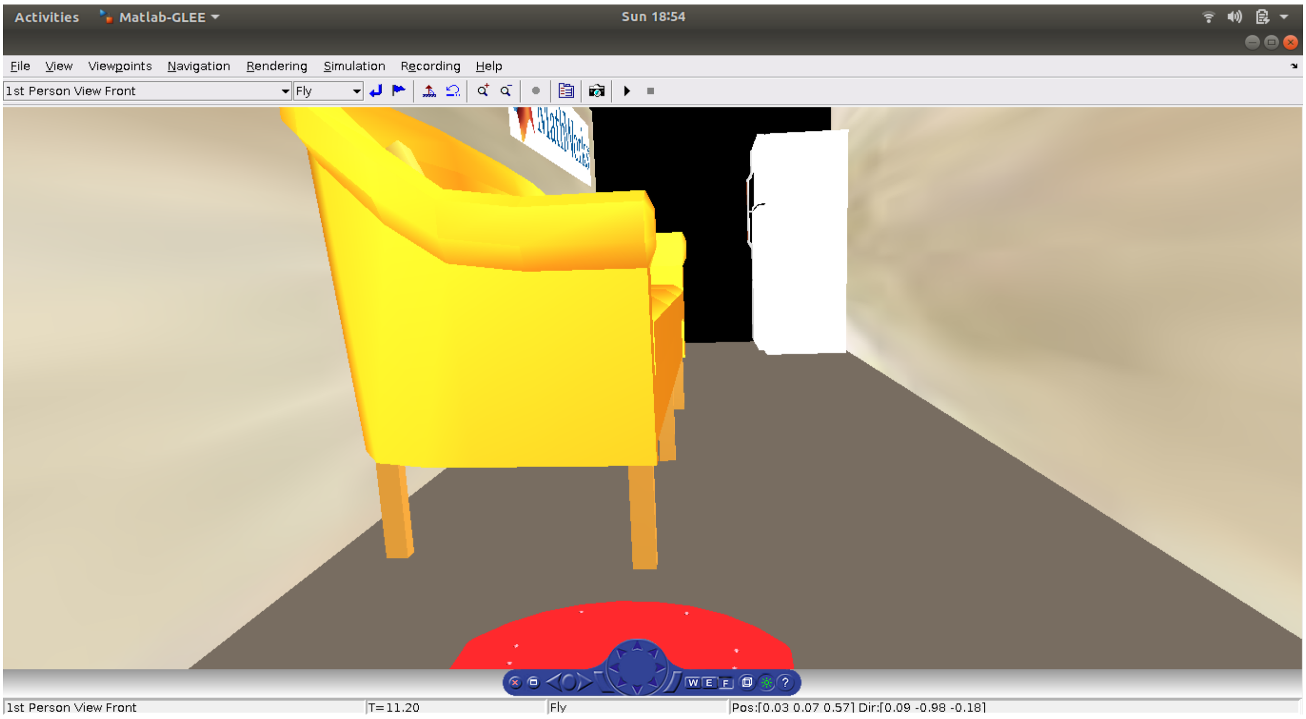


Fig 5.4.1 Virtual Environment Front View

Top view:

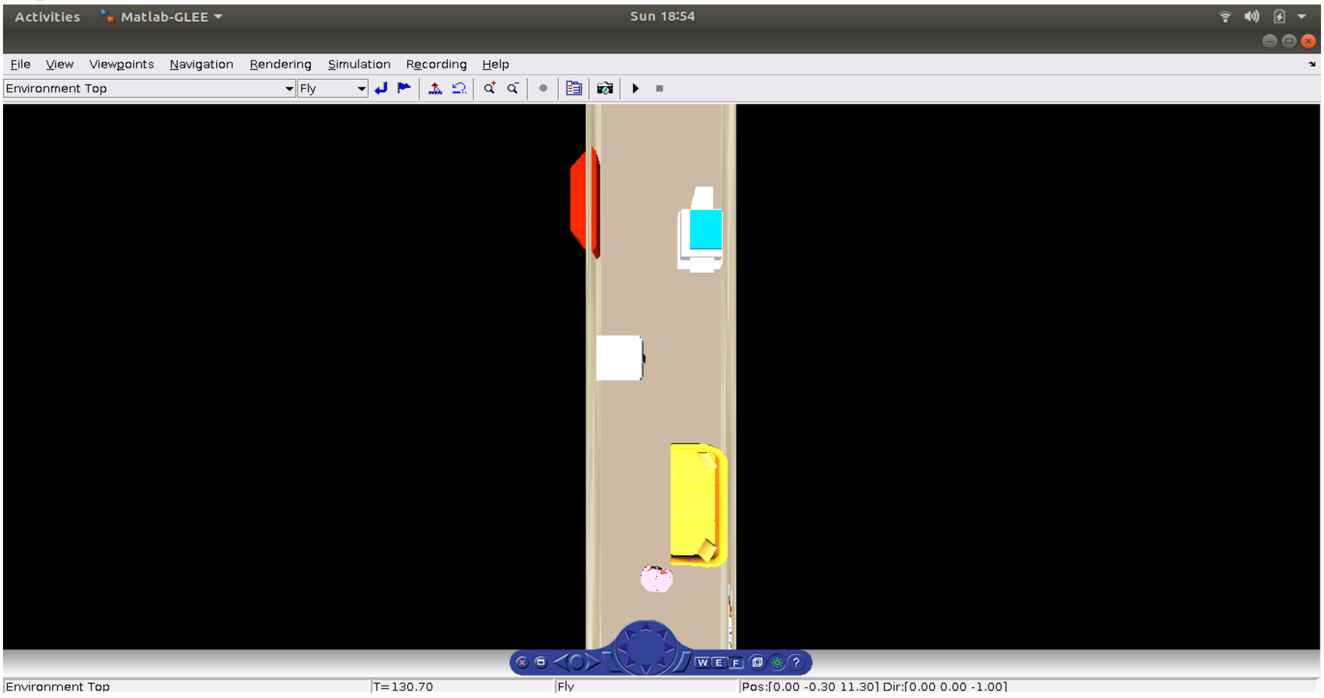


Fig 5.4.2 Virtual Environment Top View

Outcomes for some sample binary maps:

Binary maps are those which contains obstacles represented by 1 and empty represented by 0. These maps are designed using maps software.

Below given are outcomes for some sample binary maps:

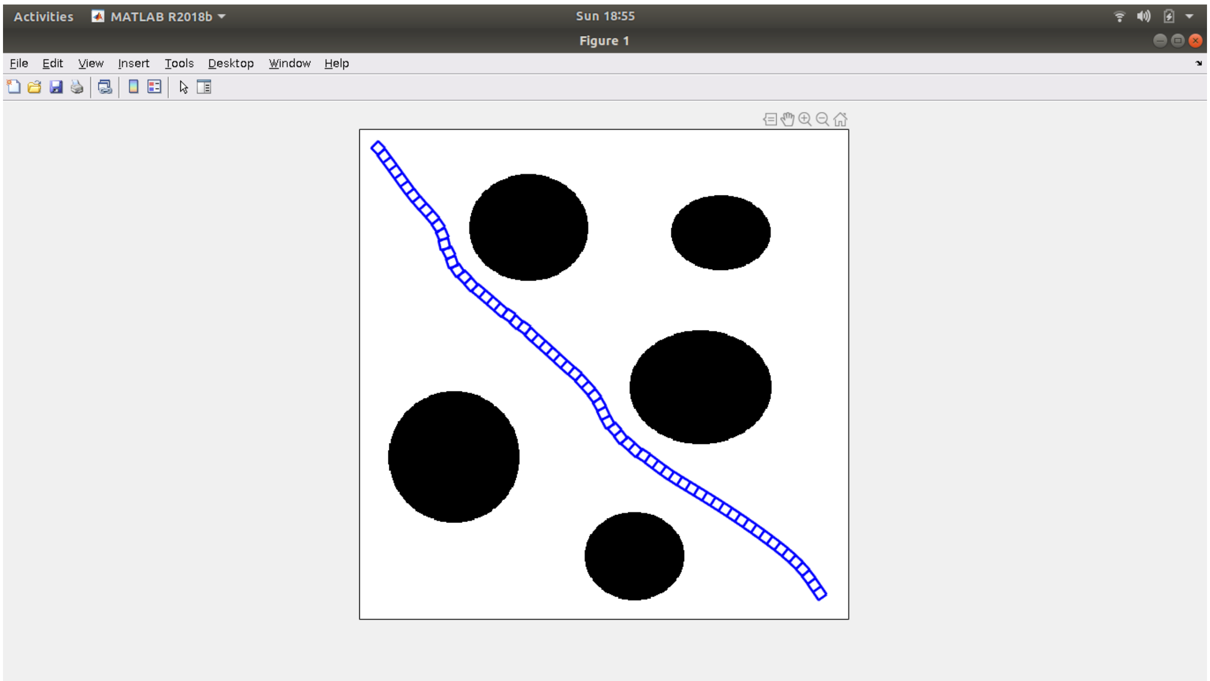


Fig 5.5.1 Bit Map -1

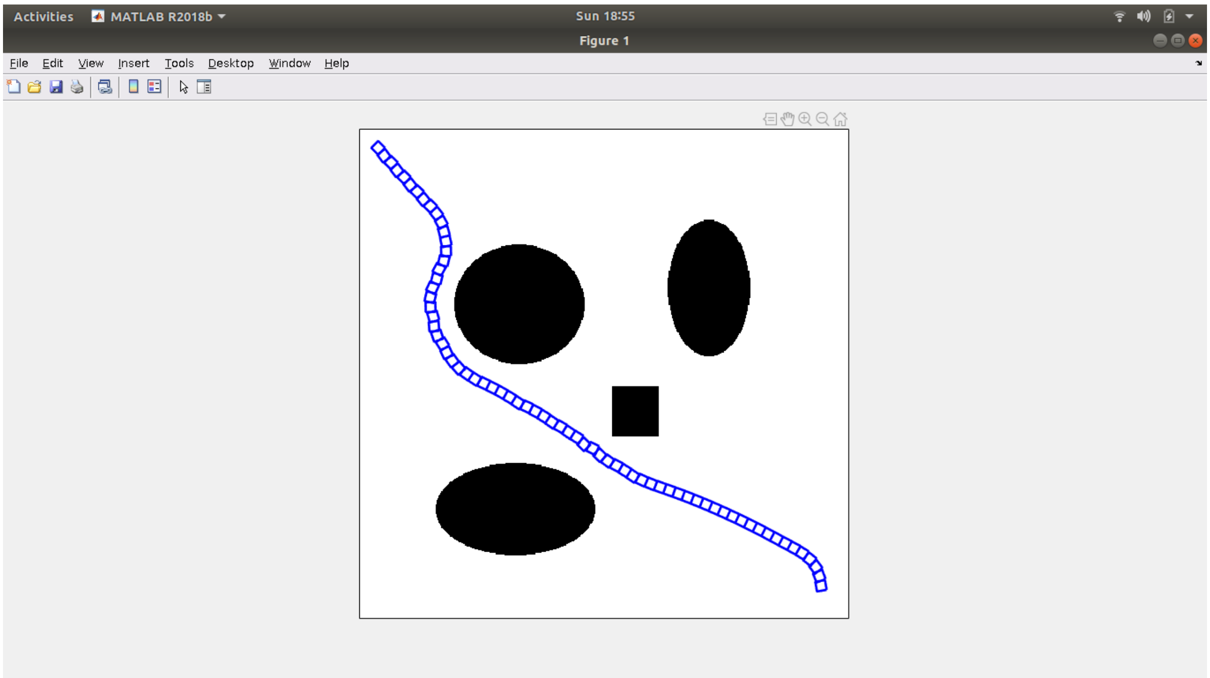


Fig 5.5.2 Bit Map -2

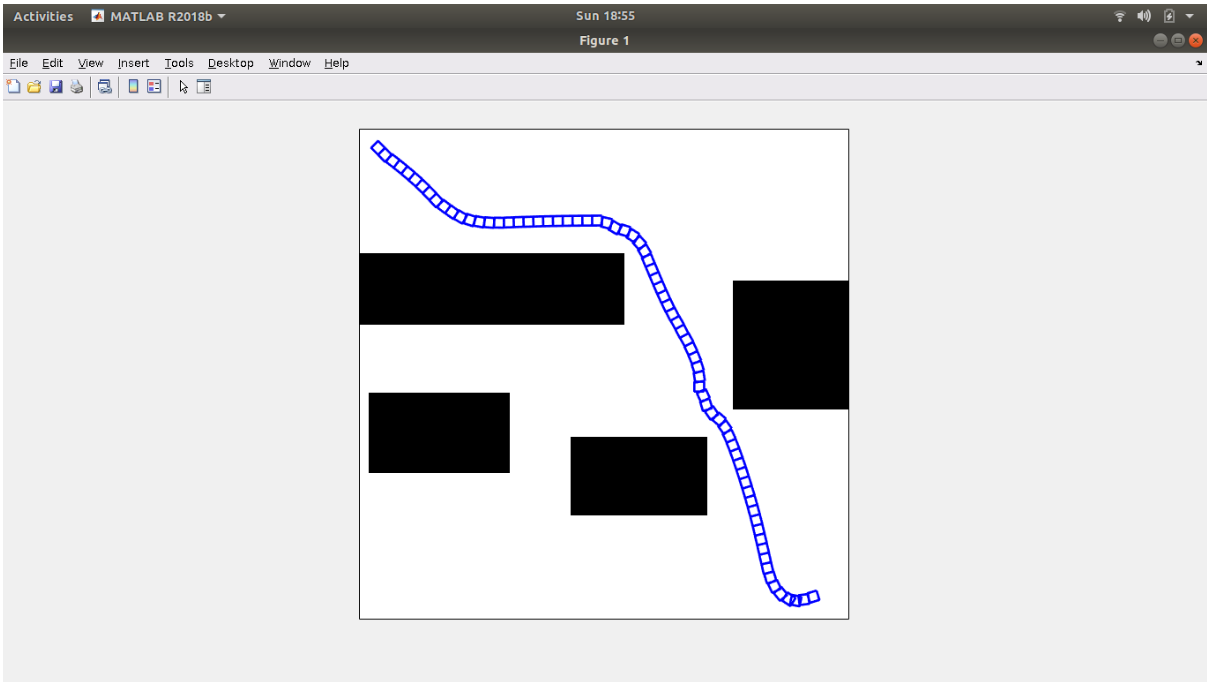


Fig 5.5.3 Bit Map -3

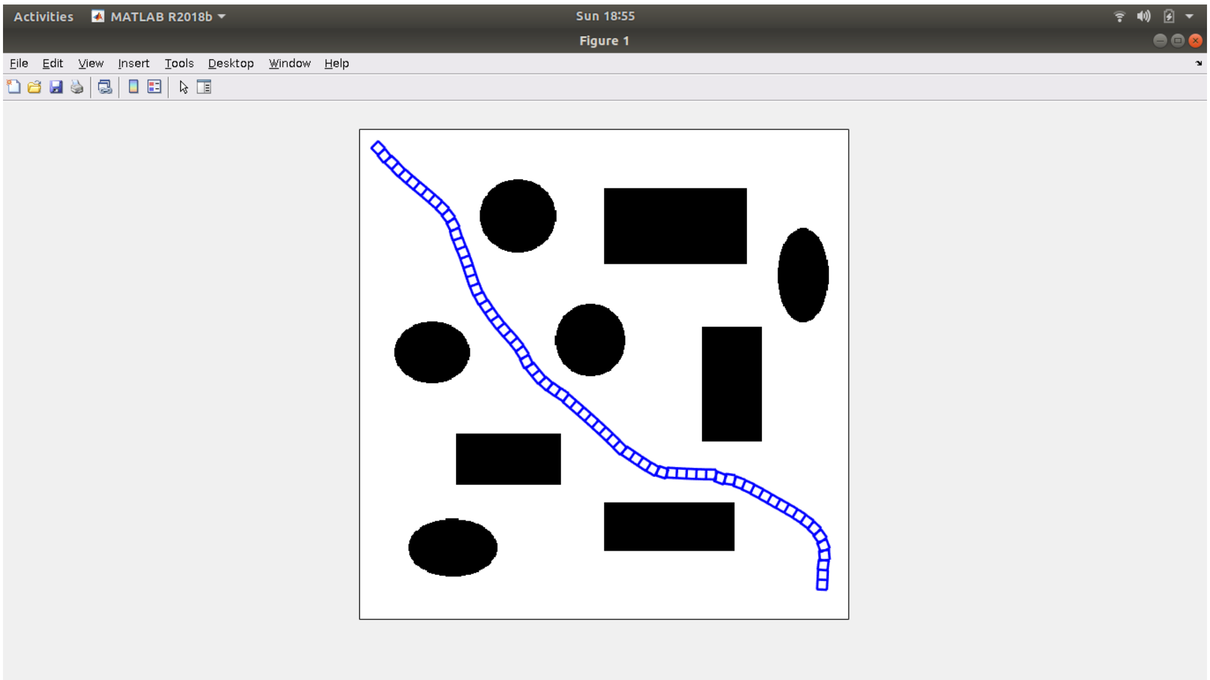


Fig 5.5.4 Bit Map -4

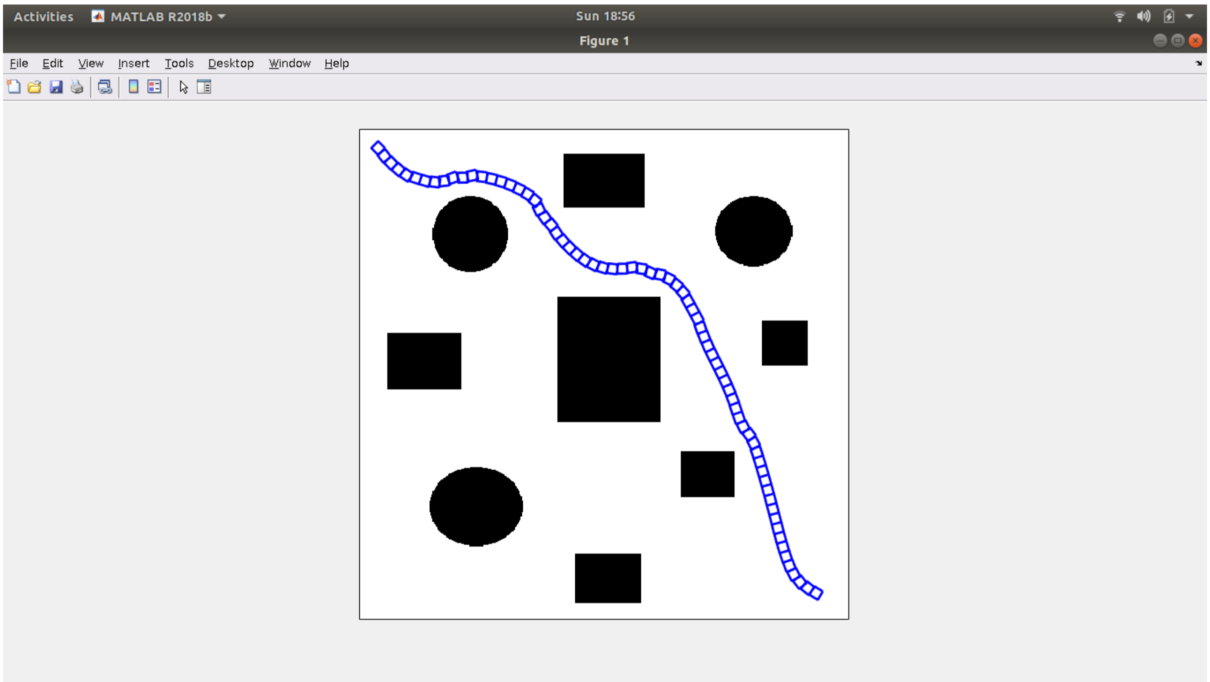


Fig 5.5.5 Bit Map -5

5.3 PERFORMANCE EVALUATION

For the performance evaluation, we are calculating the time taken to reach the destination from the source and we are also calculating the cost of the path (The distance from the source to destination).

From the below images we can see the time is taken and the cost of the path is calculated:

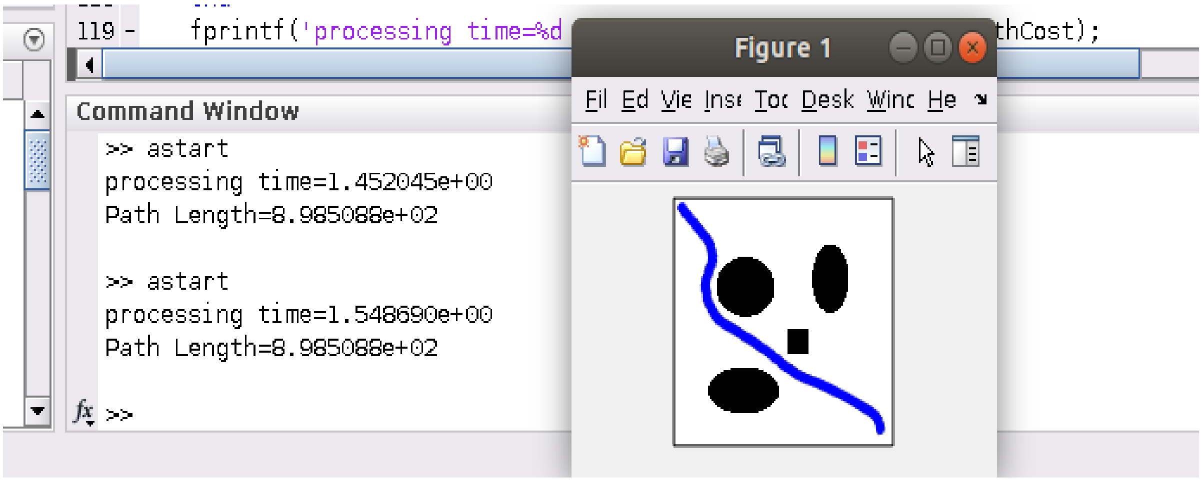


Fig 5.6 Performance

Using these metrics, we can compare how efficiently we are calculating the path, compared to other path planning algorithms. For any selected algorithm the path cost should be minimum and the computing or processing time should be minimum.

Comparison with Genetic Algorithm:

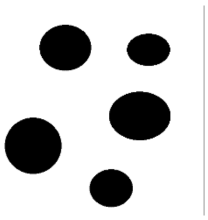
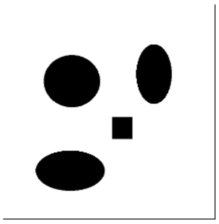
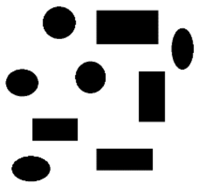
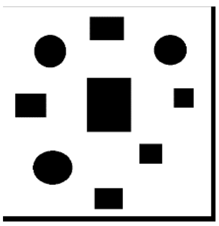
Map	Fuzzy Time	Fuzzy Distance	GA Time	GA Distance
	3.481758e+00	903	6.129890e+01	861
	1.728770e+00	898	4.398107+e01	882
	3.021180e+00	900	4.124277e+01	880
	2.850881e+00	897	4.675294e+01	875
	3.647383e+00	900	5.450715e+01	877

Table 5.1 Comparison with GA.

6. CONCLUSION

The path planning is a task which is defined as the process of computation of the motion sequence enabling the robot from one position to another position automatically without any human actions. Robot uses sensors to gather information from the environment and passes this input to some unit to find the next position of the robot. There is a lot of imprecision and uncertainty in the path planning implementation using fuzzy logic, “Fuzzy logic is an approach to computing based on degree of truth rather than the usual true or false. We aggregate data and form a number of partial truths which we aggregate further into higher truths which in turn, when certain thresholds are exceeded, cause certain further results such as motor reactions”.

The WDO (Wind Driven Optimization) algorithm is working based on the “atmospheric motion of infinitesimal small air parcels navigates over an N-dimensional search domain”. .We are going to propose an approach for designing optimized fuzzy controller using WDO which allows the robot to move in both dynamic and static environments.

The fuzzy logic takes input taken from the sensors and sends it into the fuzzy inference system and releases velocities as output.These velocities are then optimized by applying WDO algorithm on the velocities and optimize them since it is a dynamic environment.This process of collecting the data from the sensors,predicting the velocities and optimizing the velocity is done continuously for every moment of the robot until it reaches its destination.

Finally, velocities of left and right motor are predicted by using Fuzzy Inference system and we optimized these velocities using Wind Driven Optimization algorithm. Finally, we made robot to navigate in a dynamic environment by reading data from sensors .

7. FUTURE SCOPE

Now a days path planning problem has become one of the important research fields in which researchers are interested in. if path planning technology is implemented properly, we can benefit a lot of advantages like reduction of the wear and capital investment of mobile robot and reduction of time.

Future research should include:

1. Path planning of mobile robots in high dimensional environment should be researched. The research in path planning of mobile robots should be done in high dimensional environment.
2. We should also research on air robots and water robots.
3. The enhancement in path planning algorithm can be done on different scales like speed and computational complexity.
4. To have the knowledge of already visited places we save the maps on a central server using additional memory which can be used in future for sharing among multiple mobile robots.
5. It requires more power to run with an additional hardware.

REFERENCES AND USEFUL LINKS

[1] “Pandey, Anish, and Dayal R. Parhi”. “Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind DrivenOptimization algorithm”.*Defence Technology* 13.1 (2017): 47-58”.

[2] “Antonelli, Gianluca, Stefano Chiaverini, and Giuseppe Fusco”. “A fuzzy-logic-based approach for mobile robot path tracking”. “ *IEEE Transactions on Fuzzy Systems* 15.2 (2007): 211-221”.

[3] Pandey, Anish, et al. "Path planning navigation of mobile robot with obstacles avoidance using fuzzy logic controller." “*2014 IEEE 8th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2014”.

[4] “Wang, Meng. "Fuzzy logic-based robot path planning in unknown environment." “*2005 International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, 2005”.

[5] “Pandey, A., & Parhi, D. R. (2014). MATLAB simulation for mobile robot navigation with hurdles in cluttered environment using minimum rule based fuzzy logic controller. *Procedia Technology*, *14*, 28-34”.

[6] “Zhai, Haoyan, Magnus Egerstedt, and Haomin Zhou. "Path Planning in Unknown Environments Using Optimal Transport Theory." *arXiv preprint arXiv:1909.11235* (2019)”.

[7] “Parhi, D. R. "Advancement in navigational path planning of robots using various artificial and computing techniques." *Int Rob Auto J* 4.2 (2018): 133-136”.

[8] “Khaksar, Weria, Md Zia Uddin, and Jim Torresen”. “Neuro-Fuzzy Sampling: Safe and Fast Multi-query Randomized Path Planning for Mobile Robots”. “ *International Conference on Robot Intelligence Technology and Applications*. Springer, Singapore, 2018”.

[9] “Saraswathi, Mbl, Gunji Bala Murali, and B. B. V. L. Deepak”. “Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm”. “ *Procedia computer science* 133 (2018): 510-517”.

[10] “Liu, Zubing, et al. "Improved Fuzzy Logic Rules and Effective Searching Box Method for Mobile Robot Obstacle Avoidance." *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, 2018”.

[11]“ Singh, Sapam Jitu, et al.” “Motion planning of mobile robot using Fuzzy-GA method along with Three Path concept in dynamic environment.” “*Journal of Intelligent & Fuzzy Systems*35.2 (2018): 1445-1457”.

[12]“ Nguyen, Quoc My, Lam Ngoc Minh Tran, and Tri Cong Phung. "A Study on Building Optimal Path Planning Algorithms for Mobile Robot”. “ *2018 4th International Conference on Green Technology and Sustainable Development (GTSD)*. IEEE, 2018”.

[13] https://www.tutorialspoint.com/fuzzy_logic/.