University of Glasgow

James Watt School of Engineering

ENG5009_Advanced Control 5 -Assignment

Title: **Movement of the robot using controller**

Submitted by:

Chandra Sai Kamal Nadendla

2719689N

Table of Contents

1.Introduction

The path selection and obstacle avoidance is an important criteria in any automotive vehicle or robots. The appropriate control system should be used to achieve the optimal path, and in many cases control system is designed in very complex way to satisfy all the conditions.

As a part of this assignment the controller is developed in such way that it can guide a robot to move successfully in defined path and avoid the obstacles in the way. To achieve this controlled behaviour of the robot I developed a controller system using fuzzy logic and this fuzzy logic mimics or replicates the way human reasons in approximate way rather than in a precise way. The main reason for using this fuzzy logic controller is it generally outperforms other controllers' algorithm in non-linear, complex, and undefined systems.

This fuzzy logic controller design starts with the set of input and output membership functions and set of rules are applied to the membership functions to get an output value or crisp output value. The two sensors are used (left sensor and right sensor) to detect the obstacles in the path and the data collected by the sensors are used by the controller to help the robot to estimate the distance of the obstacle and change its movement accordingly. In the same way as sensors, two motors are used (motor-left and motor-right) to initiate the voltages and along with that calculated heading angle is used to estimate the change of angle for predicting the direction of checkpoints or to navigate from one checkpoint to another checkpoint.

The sensor data and change in heading angle data are used by the controller as the input values and voltages are the expected output value.

The complete modelling and simulation are carried out in the MATLAB and fuzzy controller is developed by the fuzzy logic toolbox in MATLAB.

2.Methodology

The main aim of this assignment is to move a robot in given checkpoints and avoid obstacles in its path. To achieve this, I used two types of controllers, one for moving in described path based on heading angle which can be referred as navigation controller and another controller is to avoid obstacles which is known as obstacle avoidance controller

The steps involved in developing the above-mentioned controllers are

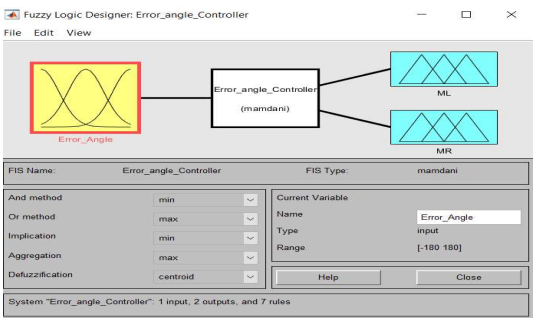
Navigation Controller

Need to record the data of new heading angle which is obtained using the compute heading angle function and current heading angle from State (24)

The difference between the new heading angle and current heading angle gives the error angle (the angle in which robot should move)

The fuzzy system is developed by taking error angle as input fuzzy set and voltage left and voltage right as output fuzzy sets

The certain set of rules are added to alter voltage left and voltage right whenever there is a positive or negative change in the error angle

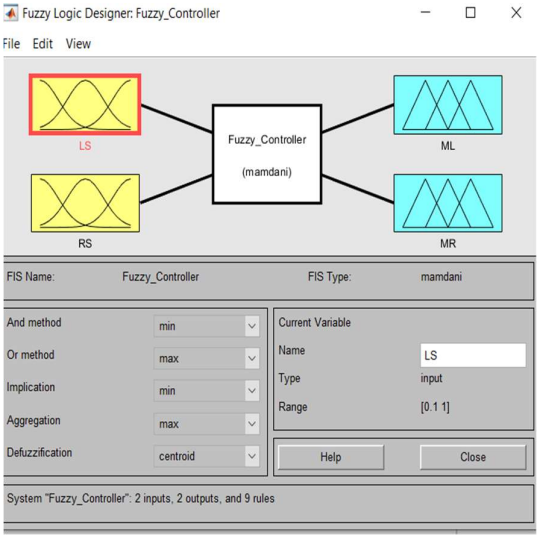


Obstacle Avoidance Controller

To develop a fuzzy controller two input data sets and output data sets are required

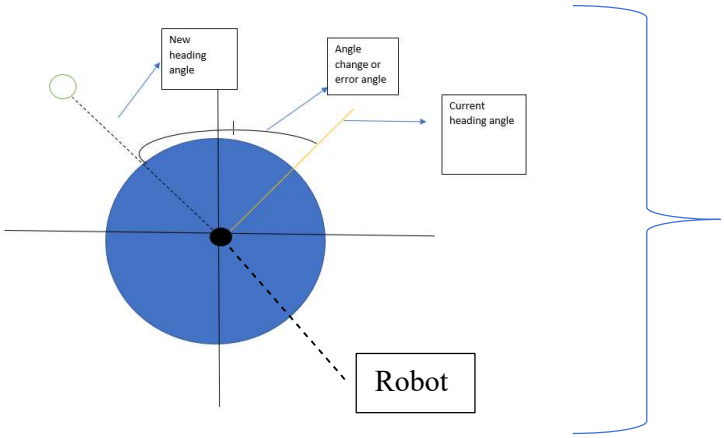
The sensor outputs are considered as the input fuzzy sets and voltage left and voltage right data is taken as output fuzzy sets

The fuzzy system is developed in such a way that whenever an obstacle a close or near or too close the voltages will be altered respectively



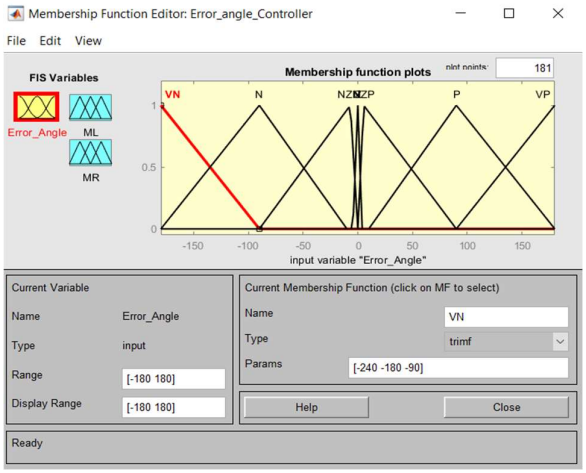
3.Navigation Controller

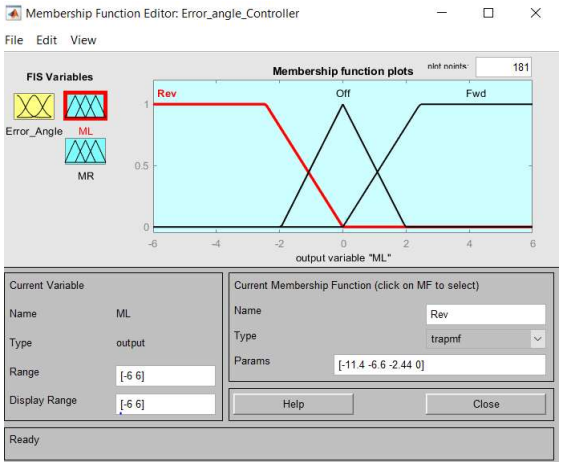
3.1 Concept of Navigation Controller



Whenever the robot is moving towards a certain checkpoint it travels in a certain angle with respect to x axis which can be referred as current heading angle. The given checkpoint of robot is at different position then the imaginary angle made by the checkpoint direction and x axis can be referred as new heading angle. So, the difference between current heading angle and new heading is error angle. This error angle can be used as the input fuzzy sets for the controller.

3.2 Navigation Controller membership functions





3.3 Navigation Controller Rules

Input Fuzzy Set	Output fuzzy set
Error angle is VN	ML is Rev and MR is Fwd
Error angle is N	ML is Rev and MR is Fwd
Error angle is NZP	ML is Fwd and MR is Off
Error angle is P	ML is Fwd and MR is Rev
Error angle is VP	ML is Fwd and MR is Rev
Error angle is NZN	ML is Off and MR is Fwd
Error angle is Z	ML is Fwd and MR is Fwd

VN: Very Negative

N: Negative

NZP: Near to zone positive

P: Positive

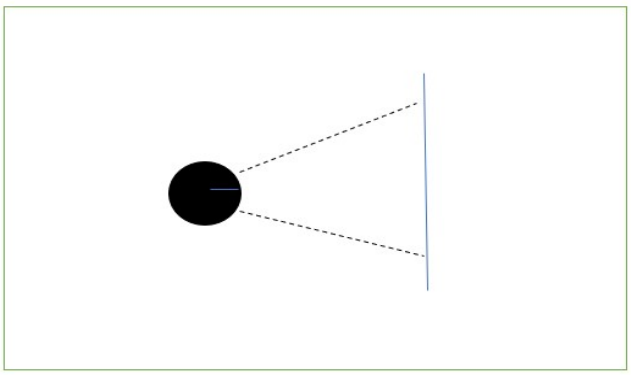
VP: Very Positive

NZN: Near Zone Negative

Z: Zero

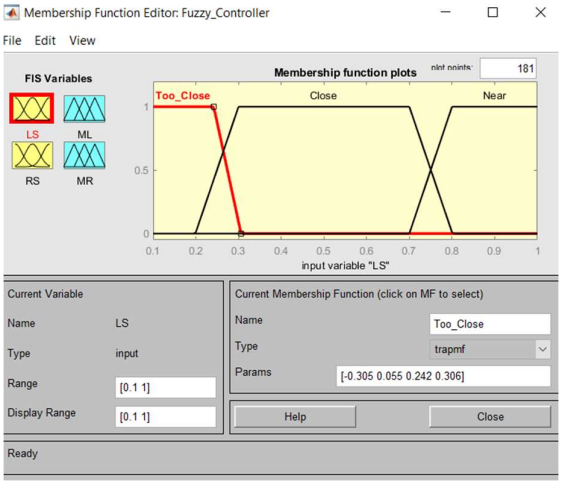
4.Obstacle Avoidance Controller

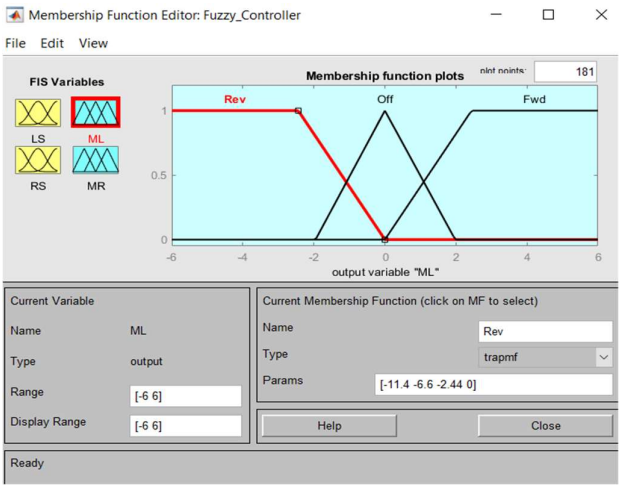
4.1 Concept of obstacle avoidance controller



Whenever robot moves sensor connected to it keeps on checking for an obstacle within the range of 0.9 m and when it finds the obstacle the data collected by the sensor will be used as input fuzzy sets in the controller which is used to alter the left and right voltages based the ranges of left and right sensors

4.2 Obstacle Avoidance Controller Membership Functions





4.3 Obstacle Avoidance Controller Rules

LS	RS	ML	MR
N	N	F	F
N	C	O	F
N	TC	R	F
C	N	O	F
C	C	R	F
C	TC	O	F
TC	N	F	R
TC	C	F	R
TC	TC	R	R

N: Near

C: Close

TC: Too close

F: Forward

R: Reverse

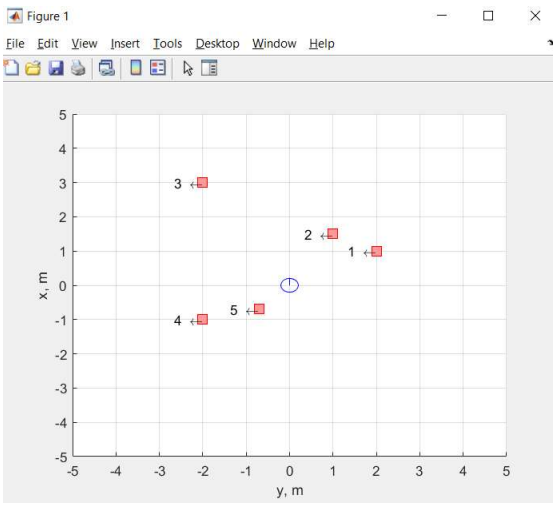
O: Off

5.Results

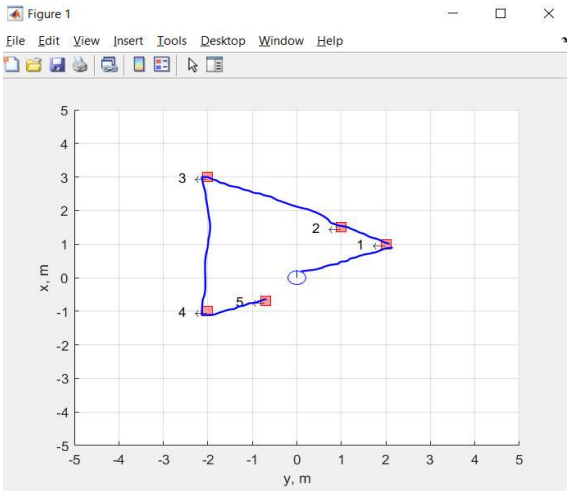
5.1 Task 1: To develop a controller that can be able to drive a robot to a selected point.

Given checkpoints: [1 2; 1.5 1; 3 -2; -1 -2; -0.7 -0.7]

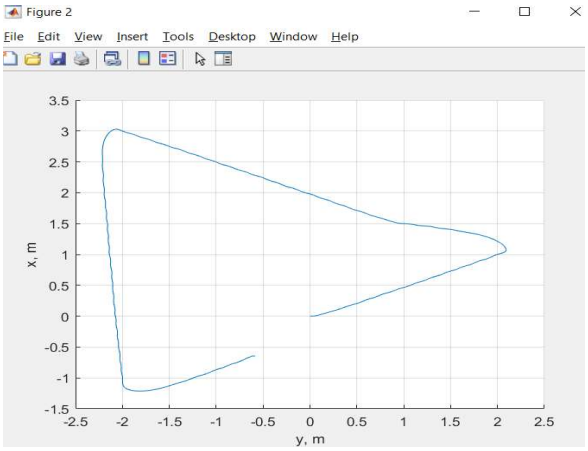
Plotting of checkpoint and positioning of robot



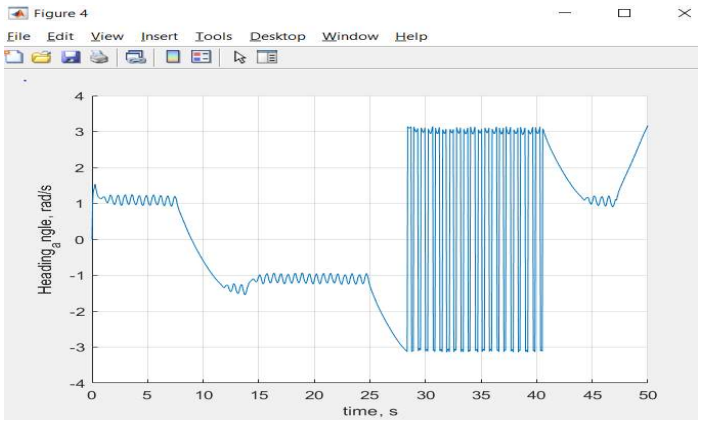
Estimated movement of the robot



Resultant movement of the robot with the help of controller



Here the robot moves from zero to all the checkpoints as estimated in the above chart. The line in the graph is linear without any disturbances from one checkpoint to another checkpoint which explains the movement of robot is smooth and work perfectly as expected

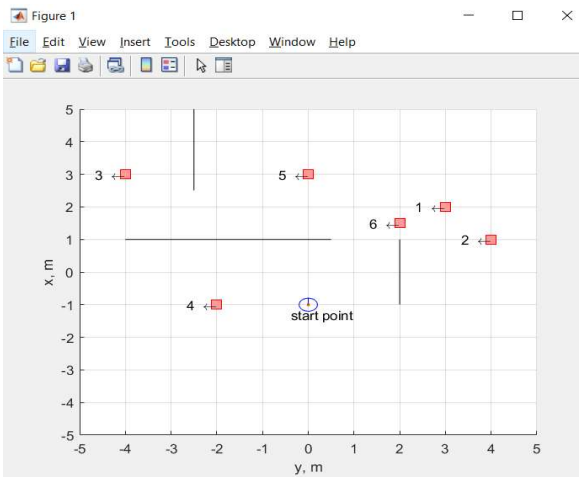


This graph shows the change in heading angle with respect to the time.

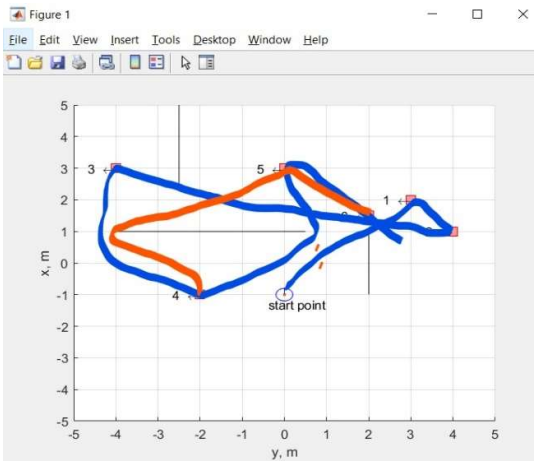
5.2 Task 2: To develop a controller that can be able to drive a robot to a selected point and avoid obstacles

Given checkpoints: [2 3; 1 4; 3 -4; -1 -2;3 0;1.5 2]

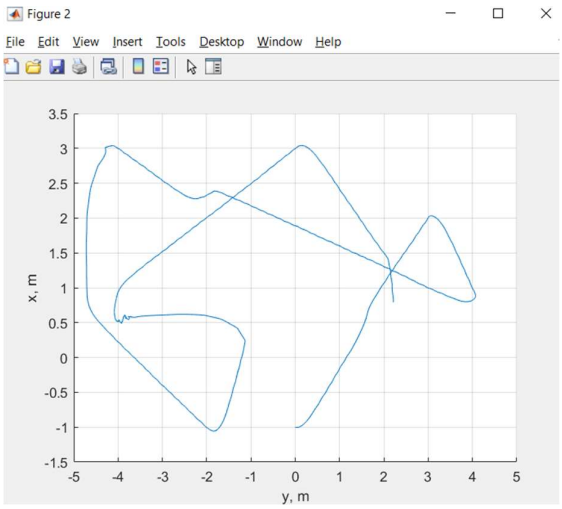
Plotting of checkpoints and positioning of robot



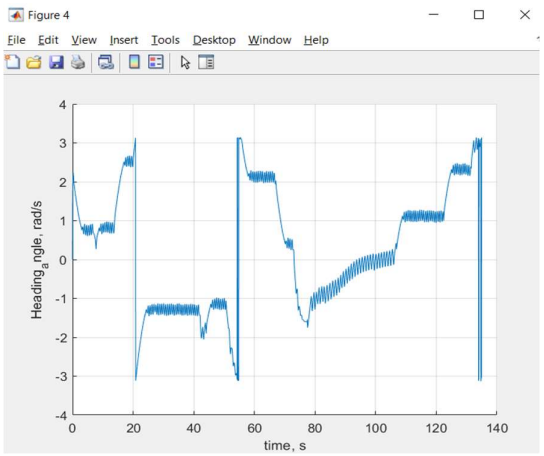
Estimated movement of the robot



Resultant movement of robot with the help of combination of both the fuzzy controllers



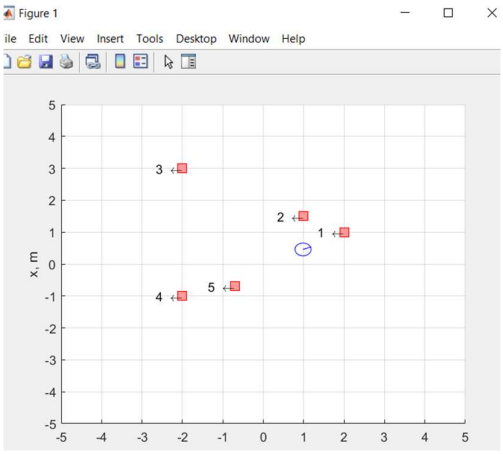
Here it is observed that the robot is moving through the given checkpoints and avoiding the obstacles along the way.

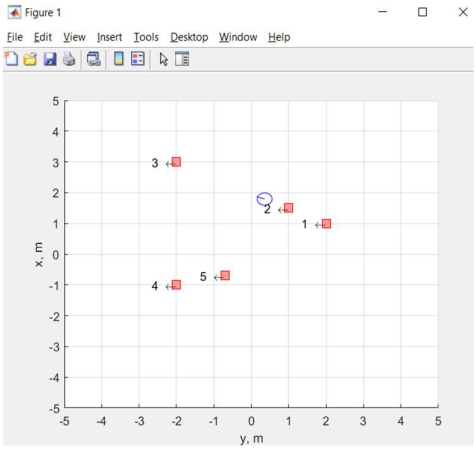


It is graph of current heading angle with respect to time.

6. Overview of the results obtained

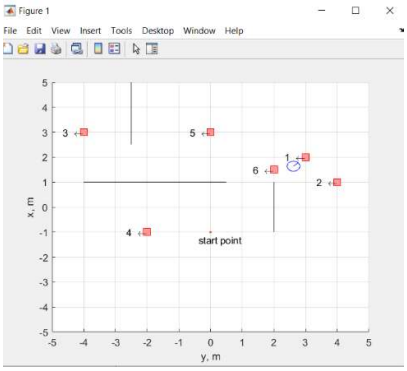
6.1 Overview of task 1 results

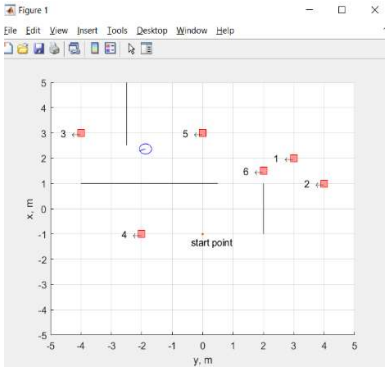


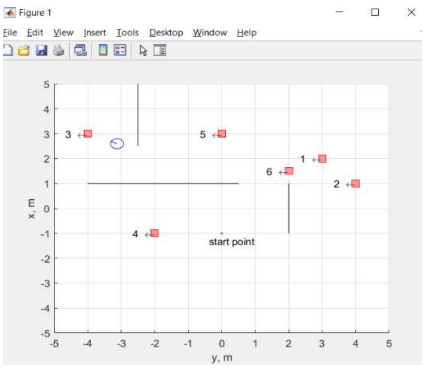


In task 1the robot starting at (0,0), heads towards first checkpoint (1,2) by changing it angle later from checkpoint 1 it heads to checkpoint 2 followed by checkpoint 3 by making its position based on the controller. In this way the robot completes the path by covering all the checkpoint within a simulation time of 50 seconds. Here only one fuzzy controller (Navigation fuzzy controller) is used as there are no obstacles to avoid. The x and y plot mentioned in the results of task 1 shows the path travelled by the robot.

6.2 Overview of task 2 results







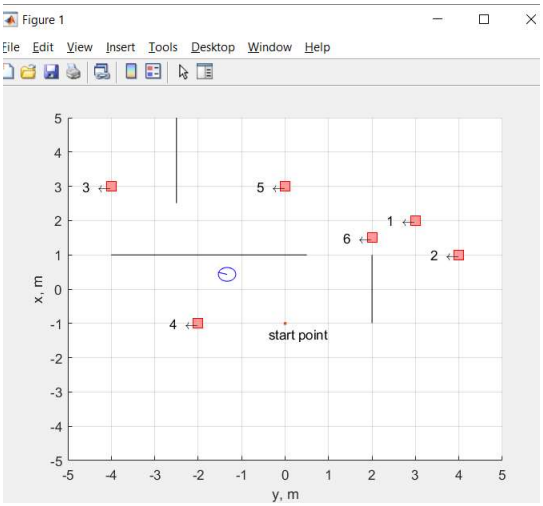
In the task 2 robot is starting from (-1,0), heads towards checkpoint 1(2,3) and move towards checkpoint 2 and checkpoint 3 respectively but while moving towards checkpoint 3 the robot encounters an obstacle and change its path and again continues towards checkpoint 3. In this way robot completes the path by covering all the checkpoints within a simulation time of 135 seconds. Here two fuzzy controllers are used (Navigation fuzzy controller and obstacle avoidance fuzzy controller), one is used for navigation and other is used to avoid obstacles. The x and y plot mentioned in the results of task 2 shows the path travelled by the robot.

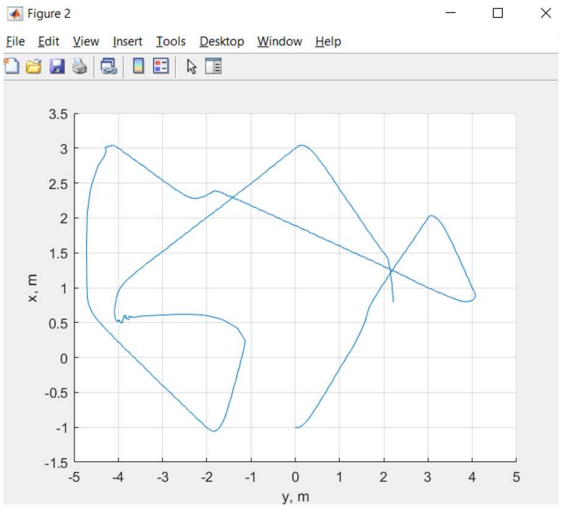
7.Further Improvisations

The fuzzy controller developed for this assignment works perfectly for the given checkpoints and travels in the desired direction. whereas in some test cases other than the given checkpoints I found that the robot is taking long distance routes which is not an optimal path that it suppose be travelled.

The reason for this is the rules of obstacle avoidance controller and the membership function of Navigation controller. Due to the limitation in the range of sensor, it is hard to find the obstacle length. So, the best possible way of fixing this is to adjust the rules and membership functions.

This limitation of controller even effected the task 2 when the robot is traveling from checkpoint 4 to checkpoint 5.





Here the robot while moving from checkpoint 4 to checkpoint 5 it took complete opposite turn and travelled to checkpoint 5.

The main aim of the controller is to make robot to move the robot to travel to given checkpoints by avoiding the obstacles, which works perfectly as it observed from the above graph.

8.Conclusion

The robot is moved successfully in the given checkpoints and avoided obstacles along the path with the help of Fuzzy controller system. The fuzzy controller controlled the behaviour of the robot by making it travel from one checkpoint to another with and without obstacles. The time taken by the controller, to complete all the checkpoints without obstacles is 80 seconds and time consumed by the robot to travel along the defined with obstacle in the path is 135 seconds. This simulation time describes the quick movement and orientation of the robot.

To achieve this controlled behaviour of robot, two simple fuzzy controllers are used, one is navigation controller and obstacle avoidance controller. The navigation controller navigates the robot through checkpoints until the sensor detects the obstacle. Once sensor detects the obstacle the navigation controller is switched to the obstacle avoidance controller and makes the robot to avoids the obstacle successfully. In this the robot is moved in defined path and avoided the obstacles.

9.References

De Silva C.W. 'Fuzzy Adaptation and Control of a Class of Dynamic Systems.' Proceedings of the Fifth IEEE International Symposium on Intelligent Control, Philadelphia, 1990, pp. 304–9

C.-Y. Chang, "Adaptive fuzzy controller of the overhead cranes with nonlinear disturbance," IEEE Trans. Ind. Informat., vol. 3, no. 2, pp. 164– 172, May 2007

Ungering A.P. and Goser K. 'Architecture of a 64-bit Fuzzy Inference Processor.' In: Proceedings of the Third IEEE Conference on Fuzzy Systems, 26–29 June, Orlando, USA, IEEE Press, 1994, 3, pp. 1776–80

Wang Li-Xin 'A Supervisory Controller for Fuzzy Control Systems that Guarantees Stability.' IEEE Transactions on Automatic Control, 39 (9), 1994, pp. 1845–7

Abdi, H. (1994). A neural Network Primer. Journal of Biological System, 2(3), 247{283

10.Appendices

A: Laboratory 1

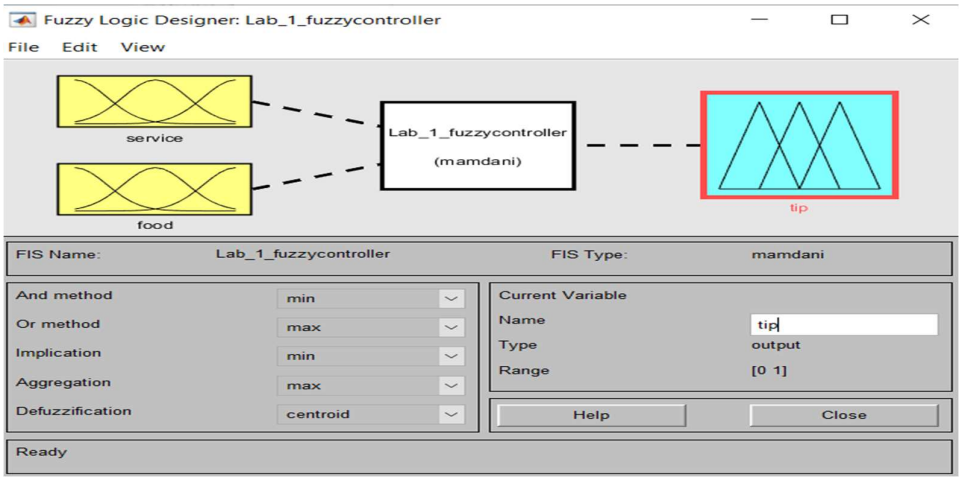
I) Task 1 is to develop a fuzzy logic Toolbox using the standard settings and to provide the tipping level for the given inputs

- Given inputs

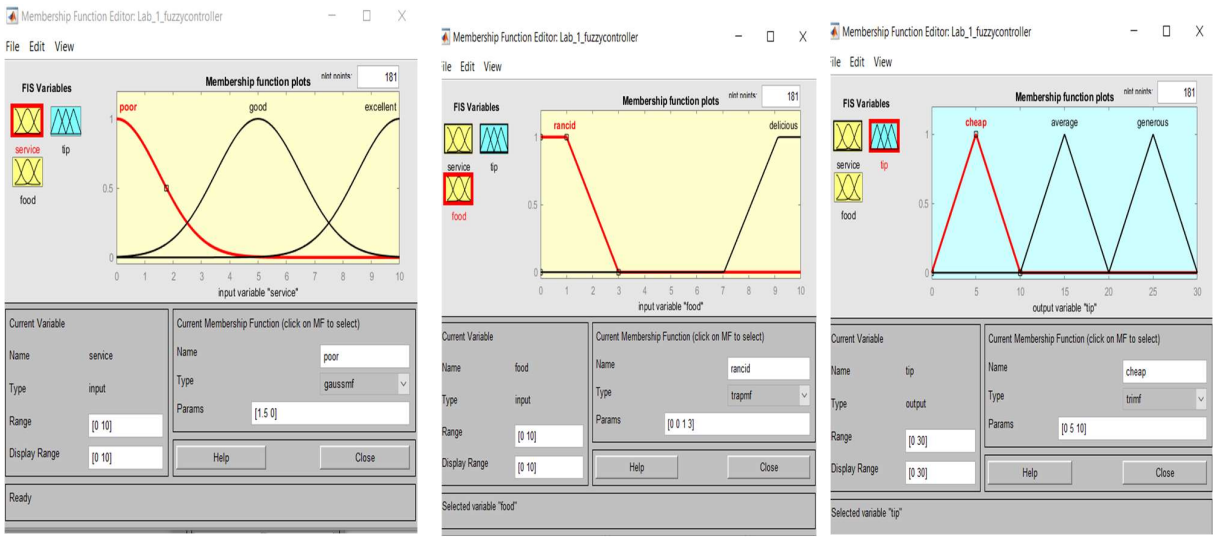
Service	Food
5	5
7	8
10	2

- Required to alter the defuzzification method to mean of maximum and bisector

- The required Fuzzy logic model



- Membership functions for the above model



- Rules of fuzzy logic model

Service	Food	Tip
Poor	Rancid	Cheap
Good	None	Average
Excellent	Delicious	Generous

- Answer Grid for Lab 1

Defuzzification Method	Service	Food	Tipping Value
Centroid	5	5	15
Centroid	7	8	20.2
Centroid	10	2	16.4
Mean of Maximum	5	5	15
Mean of Maximum	7	8	25.1
Mean of Maximum	10	2	24.9
Bisector	5	5	15
Bisector	7	8	21
Bisector	10	2	22.5

II)Task 2 is to develop fuzzy system using command line

- Required to find the values for given inputs

Service	Food
5	5
7	8
10	2

- The MATLAB code

```
tipper = readfis('Lab_1_fuzzycontroller');
>> input = [5 5;7 8;10 2]

input =

    5     5
    7     8
   10     2

>> output = evalfis(tipper,input)

output =

   15.0000
   27.6000
   24.9000
```

- Output Answer grid of lab 1

Defuzzification method used: Largest of Maximum

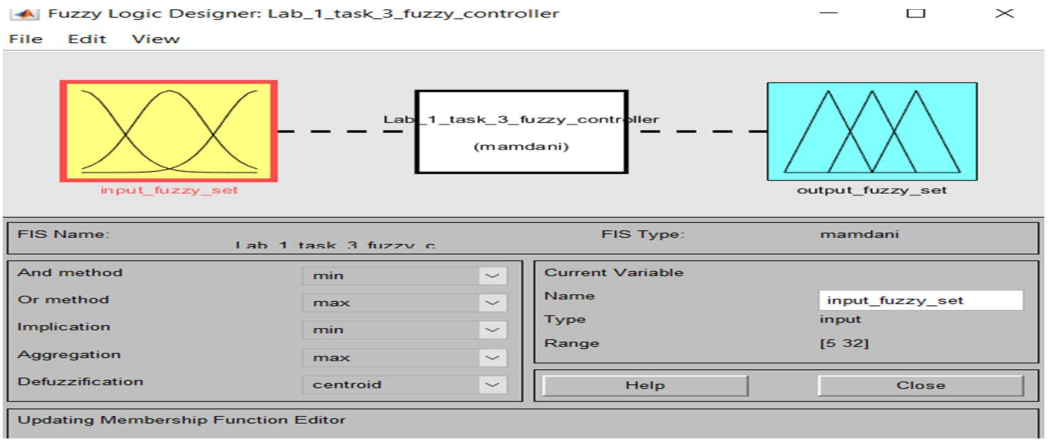
Service	Food	Value
5	5	15.000
7	8	27.600
10	2	24.900

III)Task 3 is to develop a AC system using fuzzy logic system

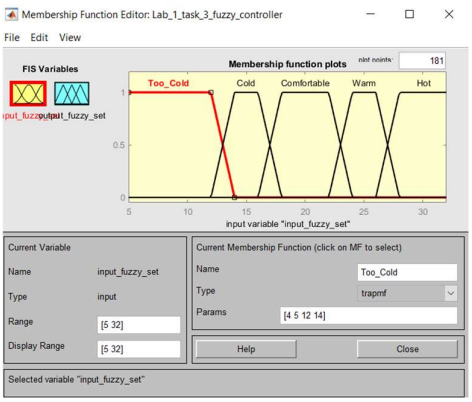
- Given input values are

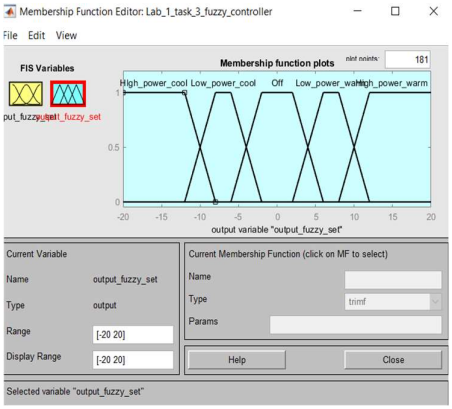
Input Temperature, °C
9
13.1
26.5
20

- The fuzzy logic model developed is



- Membership functions for the developed fuzzy system is





- Rules for fuzzy system are

Temperature	AC Power
Too Cold	High Power Cool
Cold	Low Power Cool
Comfortable	OFF
Warm	Low Power Warm
Hot	High Power Warm

- Output answer grid for lab 1 task 3

Input Temperature, °C	**Output**
9	-15
13.1	-11.2
26.5	9.56
20	-9.44e-17 ~ 0

B: Laboratory 2

I)Task 1: It is to provide basic operation for the robot

- Robot should move in straight line

MATLAB code

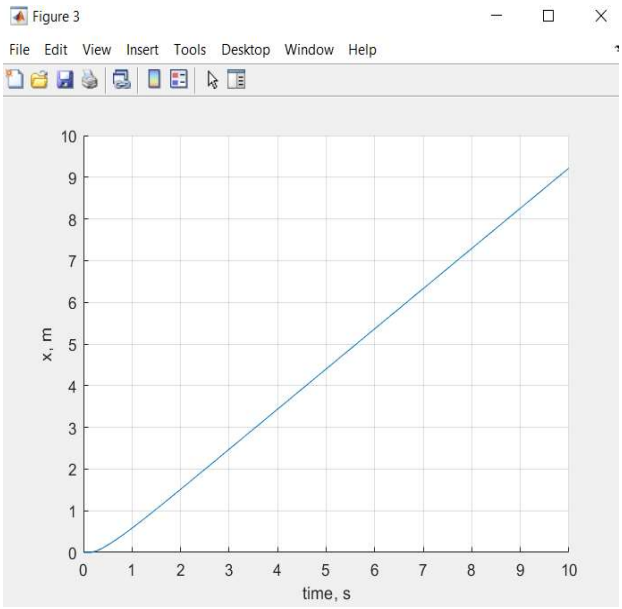
Simulation setup

```
simulationTime_total = 5;
stepSize_time = 0.05;

voltage_left = 0;
voltage_right = 0;
```

Main Simulation

```
timesteps total = simulationTime_total/stepSize_time;
state = state initial;
time = 0;
for timestep = 1: timeSteps_total
   voltages = [voltage_left; voltage_left; voltage_right; volt-
   age_right];
         voltage_left = 6;
         voltage_right = 6;

end
```

- Graph plots



The Graph indicates the distance travelled by the robot in x direction with respect to time of 5s

- Robot should move in counter clockwise direction

  MATLAB code

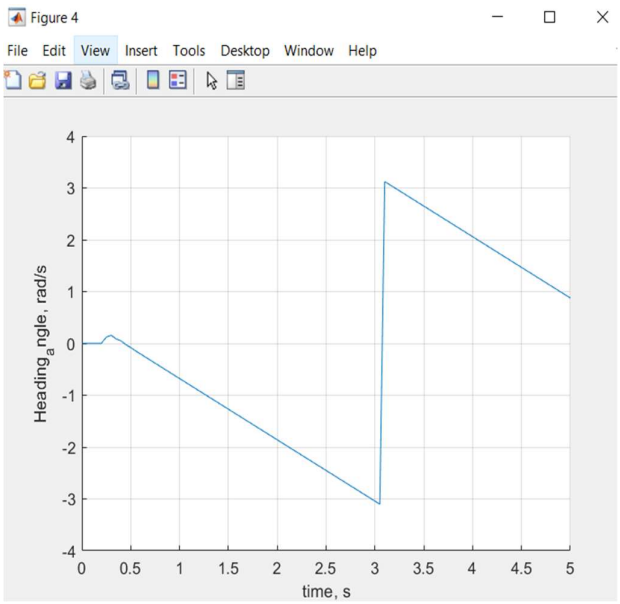
  Simulation setup

```
simulationTime_total = 5.
stepSize_time = 0.05.

voltage_left = 0.
voltage_right = 0.
```

  Main Simulation

```
timesteps total = simulationTime_total/stepSize_time;
state = state initial.
time = 0;
for timeStep = 1:timeSteps_total
   voltages = [voltage_left; voltage_left; voltage_right; volt-
   age_right];

          voltage_left  = -6;
          voltage_right = 6;


end
```

- Graph plots



When the robot is moved in counter clockwise direction the heading angle graph with respect to time of 5s

- Robot should move in clockwise direction

MATLAB code

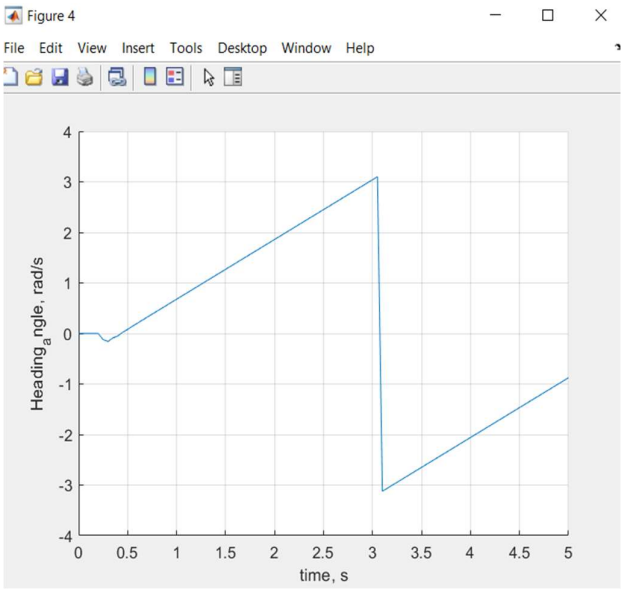
Simulation setup

```
simulationTime_total = 5;
stepSize_time = 0.05.

voltage_left = 0;
voltage_right = 0;
```

Main Simulation

```
timesteps total = simulationTime_total/stepSize_time;
state = state initial.
time = 0;
for timestep = 1:timeSteps_total
   voltages = [voltage_left; voltage_left; voltage_right; volt-
   age_right];
          voltage_left  = 6;
          voltage_right = -6;


end
```

- Output Graph



When the robot is moved in clockwise direction the heading angle graph with respect to time of 5s

- Robot should Complete a short forward, turn left, forward, turn right

MATLAB code

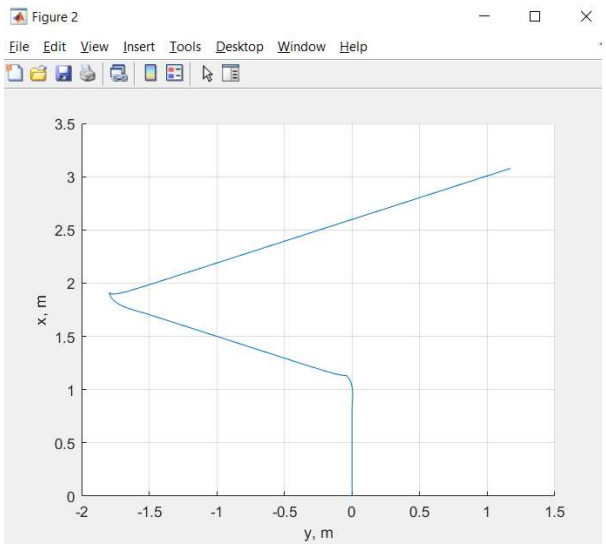
Simulation setup

```
simulationTime_total = 5;
stepSize_time = 0.05;

voltage_left = 0;
voltage_right = 0;
```

Main Simulation

```
timesteps total = simulationTime_total/stepSize_time;
state = state initial.
time = 0;
for timeStep = 1:timeSteps_total
   voltages = [voltage_left; voltage_left; voltage_right; volt-
   age_right];
   if timeStep == 1/stepSize_time
          voltage_left = -6;
           voltage_right = 6;
   elseif timeStep == 2/stepSize_time
          voltage_left = 6;
          voltage_right = 6;
    elseif timeStep == 4/stepSize_time
          voltage_left = 6;
          voltage_right = -6;
    elseif timeStep == 6/stepSize_time
          voltage_left = 6;
          voltage_right = 6;
   end
```

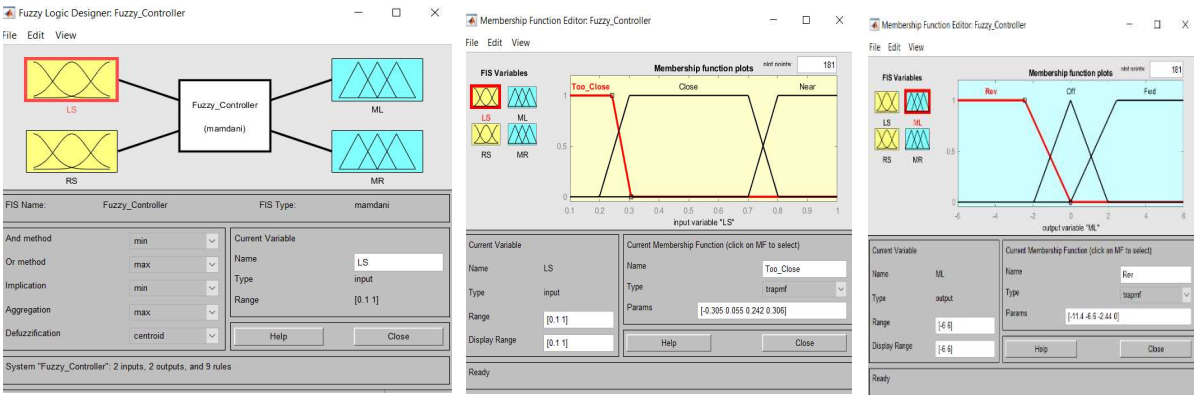
- Output Graph



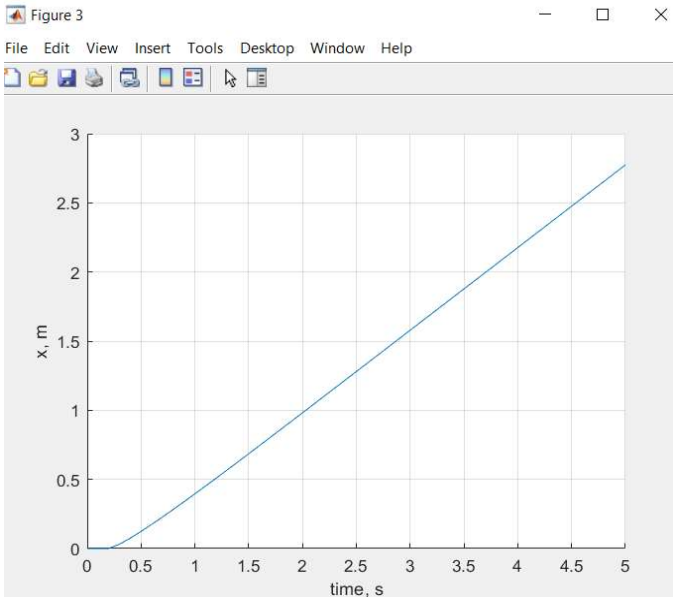
The robot first moves forward for 1 second later it lakes left turn and moves forward 2 seconds and afterwards it takes right turn and move forward

II)<u>Task 2:</u> To run a model using fuzzy controller

- <u>Fuzzy controller model</u>



- <u>Graph of (x, y) when there is no obstacle</u>

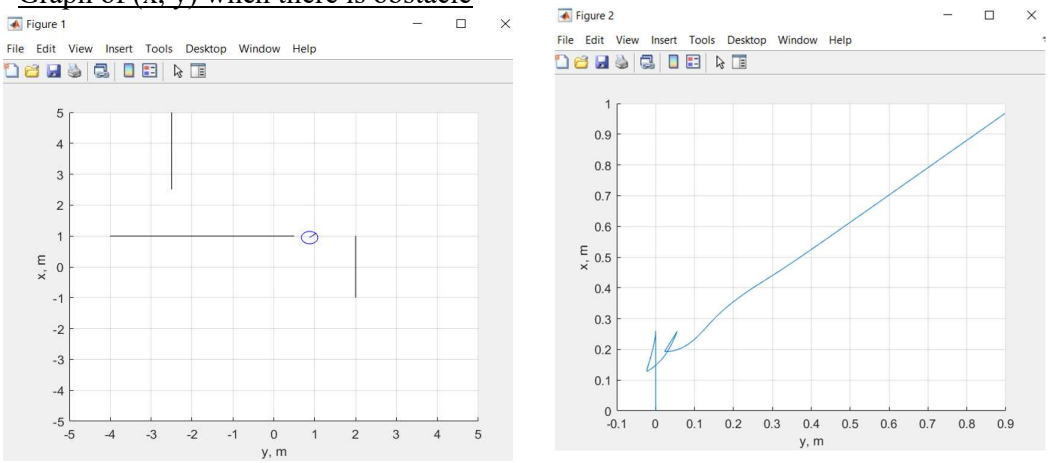


<u>Behaviour of the system</u>:

Here from the x and time graph it is observed that robot is moved in straight line motion with 6 as initial voltages of both the motors and with no obstacles.

There is no change in the robot path when there are no obstacles

- <u>Graph of (x, y) when there is obstacle</u>



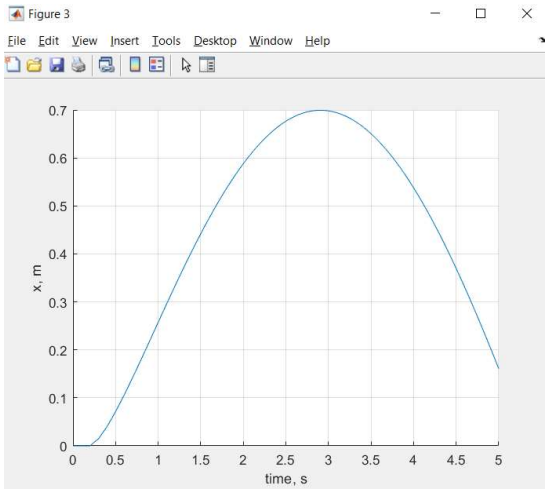
<u>Behaviour of the system</u>: From the above x and y graph it is observed that the robot takes the diversion to avoid the obstacle, the change of angle in robot is because of fuzzy rules

III)Task 3: To run a model using Neural network

MATLAB code:

```
    simulationTime_total = 5;
    stepSize_time = 0.05;
    voltage_left  = 0;
    voltage_right = 0;
    weight1 = -1.4;
   weight2 = 1.2;
   weight3 = 1.25;
   weight4 = -1;
   threshold1 =  0;
  threshold2 =  0;

%% Calculate nodes
 nodeLeft  = sensorOutput(1) * weight1 + sensorOutput(2) * weight3;
 nodeRight = sensorOutput(1) * weight2 + sensorOutput(2) * weight4;
 %% Compare with tresholds and prepare return variables
if nodeLeft >= threshold1
      voltage_left  = 6;
end
if nodeRight >= threshold2
      voltage_right = 6;
end
```

- Output Graph



Behaviour of the system:

It is observed that robot moves in the circular path, as the simulation mentioned here is 10 seconds the robot takes half turn.

From x and time graph it is observed that the distance travelled by the robot is like curve with respect to time