Fuzzy Controller Project Report

By: Tanmayee Gujar
CSE 454, Fall 2021
https://tanmayeegujar.github.io/

Project Description: This Fuzzy controller works to make the agent (a robot with dimension 1*1 units) reach the user specified target, by using Fuzzy Logic. The space this robot is moving in can be considered to be a 20*20 grid. The user input is mapped to fuzzy membership functions of distance and theta, are then a rule-based approach is used to defuzzify the inputs and give out crisp output values in terms of the speed and change in angle. This output is then used in the next iteration of the controller, and so on, until the robot reaches the target. Each iteration has a timestamp of 100 ms.

1. The system inputs are:
   x and y coordinates of current location of robot
   Angle to target
   Distance to target

2. Each iteration, the system outputs:
   The speed of robot
   The change in angle

3. The constraints are:
   The time interval in each iteration (100 ms)
   Target location
   Distance (20*20 grid)

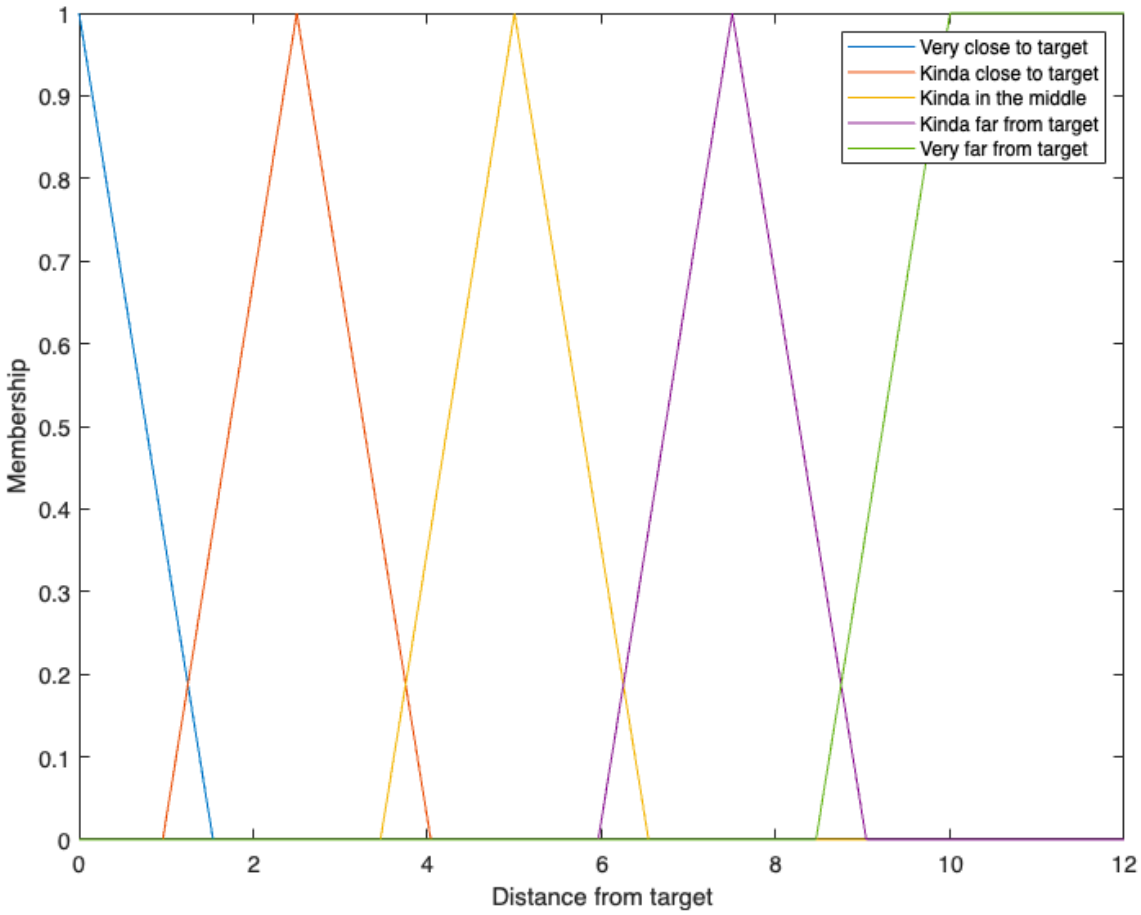
4. Things defined in project:
   The 20*20 gridspace
   The coordinate system used
   The membership functions for distance and theta
   The 5 fuzzy sets for distance and theta each
   Functions using weights to output speed and change in angle

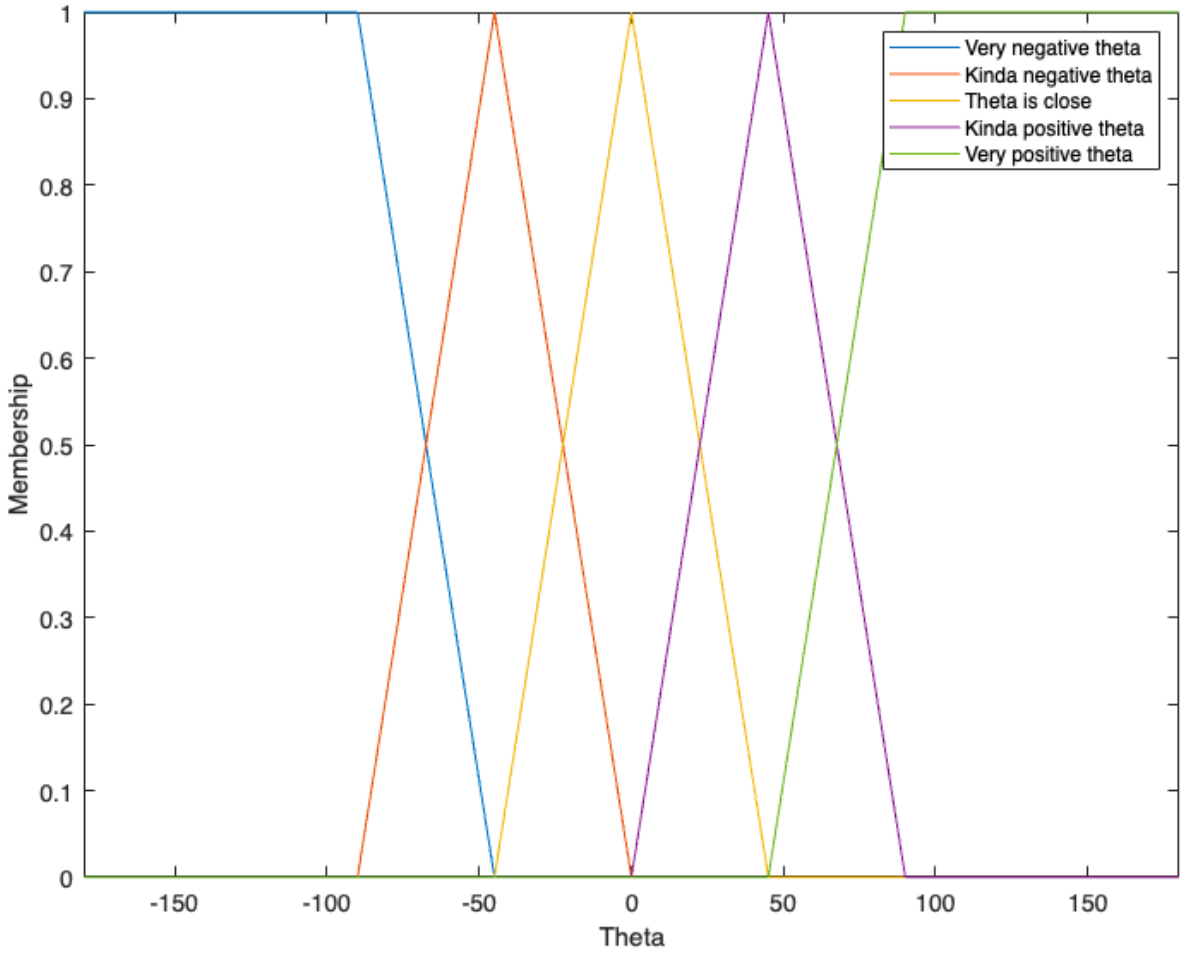
5. Physics equations used to describe the robot's movements:
   For calculating position from distance (r) and angle (θ):
   $x = r * sin(\theta)$ and $y = r * sin(\theta)$
   For calculating distance: $Distance = Speed * Time$
   To calculate distance between two points (a,b) and (x,y):
   $Distance = \sqrt{(x-a)^2 + (y-b)^2}$

To find angle between current location and target:

$\theta = arctan(\frac{\Delta x}{\Delta y})$

6. The fuzzifier was created by having membership functions for the distance between robot and target, and angle between robot and target. The fuzzy sets for distance membership function were: Very close, Kinda close, Kinda in the middle, Kinda far and Very far. The fuzzy membership sets for angles are: Very negative, Kind negative, Close to the angle, Kinda positive, Very positive.





7. The fuzzy inference engine was created using an expert rule-based system. For the distance membership functions, the speed was proportional to the distance, so the farther the robot from the target, the faster the speed, and it slows down as it gets closer, stopping as it reaches the target. Similarly for angle, the change in angle is large if the angle between robot's direction and target is large, vise versa.

8. The formula for the defuzzifier used is the sum of all memberships times the values assigned as an expert system.
   $Output = \Sigma(memberships * expert\ value)$

9. Results:
   For x = 0; y = 0; angle = 30°, distance = 8 units:
   The target was reached in 78 iterations.
   For x = 7, y = -3, angle = -17°, distance = 13 units:
   The target was reached in 97 iterations.
   For x = -10, y = 9, angle = 46°, distance = 0 units:
   Target was already reached, no iterations needed.

The weights of memberships, speed and change in angle at each iteration are output in the command window of the MATLAB code.

10. Ethical issue: The biggest one taken into consideration is for the robot being able to slow down and stop when there is an obstacle in the path, in this case, the target. This is especially important in real life situations since if this robot was deployed, it would be of utmost importance that the robot does not run into and harm anyone.
    Another ethical issue that can be taken into account in the future regarding this robot is the efficiency with which the robot is able to reach its destination. At this point, it is not very efficient at doing that, but the program could be tweaked to make its path optimal.