Caroline Hart
CSE 454 - Fuzzy Project Writeup

GitHub Repository

https://github.com/cmhart2/CSE454-Fuzzy_Project#readme

Inputs

X-coordinate: [0, 100]
Y-coordinate: [0, 100]
θ: [$-\pi$, π]

Outputs

Linear velocity: [0, 10]
Angular velocity: [-0.5, 0.5] *radians/s*

Constraints

- Every 100 ms, the robot will update both its linear and angular velocity.
- Assume the change in θ and speed happen instantaneously
- The robot must start at (x, y) coordinates inputted by the user
- The robot must travel to the destination coordinates

Things I Defined

For this project, I confined myself to a 100 x 100, XY grid for the robot's movement. This means that the robot is forbidden to move outside this area.

Physics Equations for the Robot's Movement

$$d = \sqrt{\left(current_x - destination_x\right)^2 + \left(current_y - destination_y\right)^2}$$

To compute the angle between the robot and the destination, first, we must vectorize the components

$$currentVector = \langle current_x + cos(\theta),\ current_y + sin(\theta)\rangle$$

Caroline Hart
CSE 454 - Fuzzy Project Writeup

$$destinationVector = \langle destination_x - current_x, destination_y - current_y \rangle$$

$$\phi = cos^{-1}\left(\frac{currentVector \bullet destinationVector}{||currentVector||\ ||destinationVector||}\right)$$

Fuzzifier Design

$f(x;a,b,c,d)$=max(min(
$x-a$
$b-a$
,1,$d-x$
$d-c$),0)

$$distance_{MS}(x;\ -31.95, -3.55, 3.55, 31.95) = max(min(\frac{x+31.95}{-3.55+31.95}, 1, \frac{31.95-x}{31.95-3.55}), 0)$$

$$distance_{SS}(x;\ 3.17, 31.6, 38.7, 67.1) = max(min(\frac{x-3.17}{31.6-3.17}, 1, \frac{67.1-x}{67.1-38.7}), 0)$$

$$distance_{H}(x;\ 39.05, 67.45, 74.55, 103) = max(min(\frac{x-39.05}{67.45-39.05}, 1, \frac{103-x}{103-74.55}), 0)$$

$$distance_{SF}(x;\ 74.55, 103, 110, 138.4) = max(min(\frac{x-74.55}{103-74.55}, 1, \frac{138.4-x}{138.4-110}), 0)$$

$$distance_{MF}(x;\ 110, 138.4, 145.5, 174) = max(min(\frac{x-110}{138.4-110}, 1, \frac{174-x}{174-145.5}), 0)$$

$$phi_{PF}(x;\ -4.556, -3.299, -2.985, -1.728) = max(min(\frac{x+4.556}{-3.299+-4.556}, 1, \frac{-1.728-x}{-1.728+2.985}), 0)$$

$$phi_{PS}(x;\ -2.985, -1.728, -1.414, -0.1571) = max(min(\frac{x+2.985}{-1.728+2.985}, 1, \frac{-0.1571-x}{-0.1571+1.414}), 0)$$

$$phi_{Z}(x;\ -1.414, -0.1571, 0.1571, 1.414) = max(min(\frac{x+1.414}{-0.1571+1.414}, 1, \frac{1.414-x}{1.414-0.1571}), 0)$$

$$phi_{NS}(x;\ 0.1571, 1.414, 1.728, 2.985) = max(min(\frac{x-0.1571}{1.414-0.1571}, 1, \frac{2.985-x}{2.985-1.728}), 0)$$

$$phi_{NF}(x;\ 1.728, 2.985, 3.299, 4.556) = max(min(\frac{x-1.728}{2.985-1.728}, 1, \frac{4.556-x}{4.556-3.299}), 0)$$

Fuzzy Inference Engine

For this project, I utilized MATLAB's Fuzzy Logic Designer for a simplified setup and use. The fuzzifier was designed with distance and phi being the fuzzy inputs. I used Mamdani fuzzy logic for my inference engine.

Fuzzy Rules

The fuzzy rules can be organized into a fuzzy inference matrix, as seen below.

Phi Distance	PF	PS	Z	NS	NF
MS	MS & PF	MS & PS	MS & Z	MS & NS	MS & NS
SS	MS & PS	SS & PS	H & Z	SS & NS	MS & NF
H	MS & PS	SS & PS	SF & Z	SS & NS	MS & NF
SF	MS & PS	H & PS	MF & Z	H & NS	MS & NF
MF	MS & PS	H & PS	MF & Z	H & NS	MS & NF

I designed these rules so that the robot would slow down both its linear and angular velocities as it nears its destination. This way, the robot has time to stop at exactly the destination, rather than risk the possibility of driving past the destination and needing to turn around.

Defuzzifier Design

For the design of the defuzzifier, I decided to keep things simple with a centroid defuzzification method. I attempted to utilize different defuzzification methods during the implementation process, but in certain cases it did not yield the desired result of having the robot reach its destination.

$$distance_{MS}(x;\ -2.25, -0.25, 0.25, 2.25) = max(min(\frac{x+2.25}{-0.25+2.25}, 1, \frac{2.25-x}{0.25-2.25}), 0)$$
$$distance_{SS}(x;\ 0.25, 2.25, 2.75, 4.75) = = max(min(\frac{x-0.25}{2.25-0.25}, 1, \frac{4.75-x}{2.75-4.75}), 0)$$
$$distance_{H}(x;\ 2.75, 4.75, 5.25, 7.25) = = max(min(\frac{x-2.75}{4.75-2.75}, 1, \frac{7.25-x}{5.25-7.25}), 0)$$
$$distance_{SF}(x;\ 5.25, 7.25, 7.75, 9.75) = = max(min(\frac{x-5.25}{7.25-5.25}, 1, \frac{9.75-x}{7.75-9.75}), 0)$$
$$distance_{MF}(x;\ 7.75, 9.75, 10.25, 12.25) = = max(min(\frac{x-7.75}{9.75-7.75}, 1, \frac{12.25-x}{10.25-12.25}), 0)$$

$$phi_{PF}(x;\ -0.725, -0.525, -0.475, -0.275) = max(min(\frac{x+0.725}{-0.525+0.725}, 1, \frac{-0.275-x}{-0.475+0.275}), 0)$$
$$phi_{PS}(x;\ -0.475, -0.275, -0.225, -0.025) = max(min(\frac{x+0.475}{-0.275+0.475}, 1, \frac{-0.025-x}{-0.225+0.025}), 0)$$
$$phi_{Z}(x;\ -0.25, 0, 0.25) = max(min(\frac{x+0.25}{0+0.25}, \frac{0.25-x}{0.25-0}), 0)$$
$$phi_{NS}(x;\ 0.025, 0.225, 0.275, 0.475) = max(min(\frac{x-0.025}{0.225-0.025}, 1, \frac{0.475-x}{0.275-0.475}), 0)$$
$$phi_{NF}(x;\ 0.275, 0.475, 0.525, 0.725) = max(min(\frac{x-0.275}{0.475-0.275}, 1, \frac{0.725-x}{0.525-0.725}), 0)$$

Caroline Hart
CSE 454 - Fuzzy Project Writeup

System Results

For this project, the ethical concern about whether the robot would encounter an object within the grid was studied. It was addressed by adjusting the angular velocity when the linear velocity is nonzero, but the robot's coordinates are not changing.

In future iterations of this project, I would like to study the ethical concern of the robot colliding with the walls. Hopefully, by studying this, we can make the robot self-correct its direction of motion if/when it collides with a wall/border. Then, this same methodology can be applied to other objects that happen to be within the field.