Fuzzy Controller Project

Ray Chen

Person #: 50336524

rchen63@buffalo.edu

CSE 454

Fuzzy Logic System:

In this project, a Matlab app called Fuzzy Logic Designer is used to design the fuzzy logic system.

Fuzzy Logic System Input

- Distance to the destination
- Angle to the destination

Fuzzy Logic System Output:

- New Speed
- New Angular velocity
- Weights of each membership function

Constraints:

- The robot does not have a sensor that detect obstacles.
- Max speed is $10\ m/s$
- Max angular velocity is $90°/s$
- The destination won't be too far away from the starting point because power source is limited. That max distance is $5000\ meters$.

Things I define:

- The max speed of the robot is $10\ m/s$.
- Max angular velocity: The robot shouldn't rotate too fast because that may cause the robot to overturn. The max angular velocity is $90°/s$.
- The path will be ideally all flat.
- Frictions are not considered.
- The robot cannot be 100% accurate, so the robot will be considered reached when it reaches the destination within 1 meter.
- The robot carries a battery of $2500\ mAh$, ideally the average current flow would be $5000\ mA$, which means the robot can run 1800s after fully charged.

$$\frac{2500\ mAh}{5000\ mA} = 0.5\ h = 1800\ s$$

Physical equations:

- Distance to the destination:

$$dx = destination\ x$$
$$dy = destination\ y$$

$$cx = current\ location\ x$$
$$cy = current\ location\ y$$

$$distance\ to\ the\ destination = \sqrt{(dx - cx)^2 + (dy - cy)^2}$$

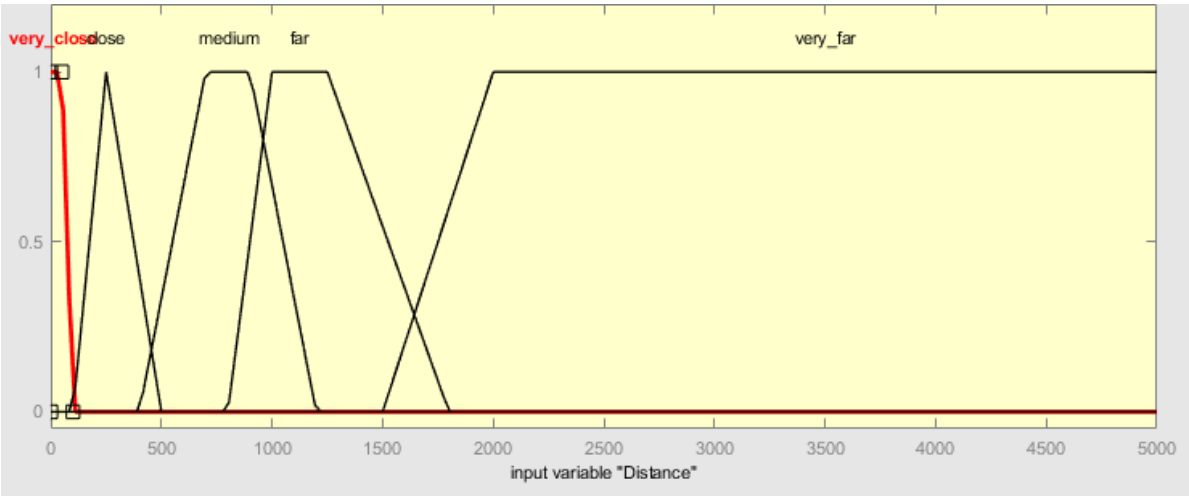
- Angle to the destination:

$$dx = destination\ x$$
$$dy = destination\ y$$
$$cx = current\ location\ x$$
$$cy = current\ location\ y$$
$$angle\ to\ destination = \arctan\left(\frac{dy - cy}{dx - dy}\right) - current\ angle$$

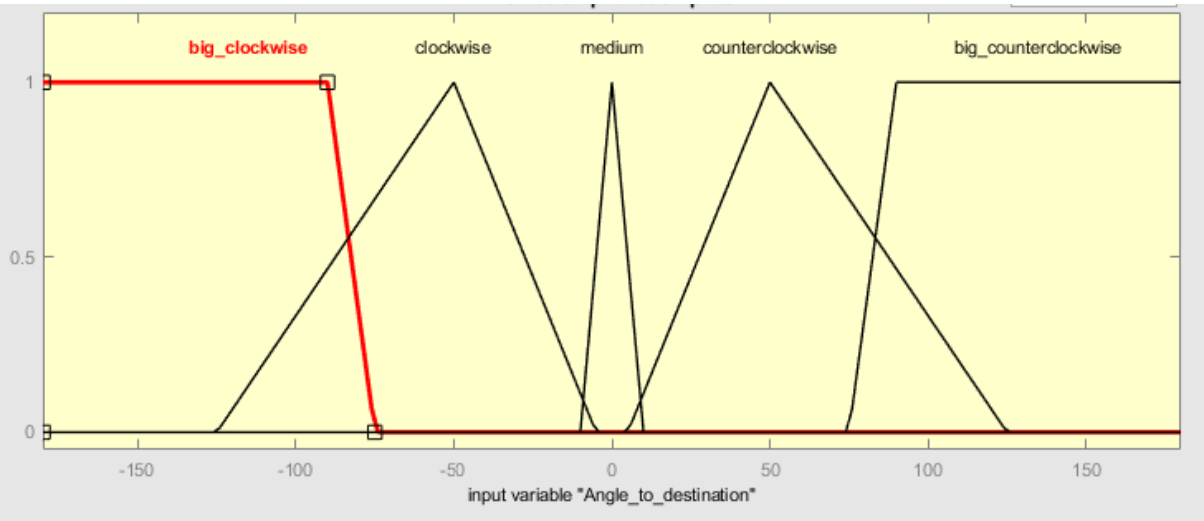
The angle to destination ranges from $-180°$ to $180°$. If $\Delta y > 0$, $\arctan\left(\frac{dy-cy}{dx-dy}\right)$ ranges from $0°\ to\ 180°$, else if $\Delta y \leq 0$, $\arctan\left(\frac{dy-cy}{dx-dy}\right)$ ranges from $0°\ to\ -180°$.

Fuzzifier:

- First membership function is about the current distance to the destination.
  - Very close
  - Close
  - Medium
  - Far
  - Very Far



- Second member ship function is about the current angle to the destination.
  - Big clockwise
  - clockwise
  - Medium
  - Counterclockwise
  - Big counterclockwise



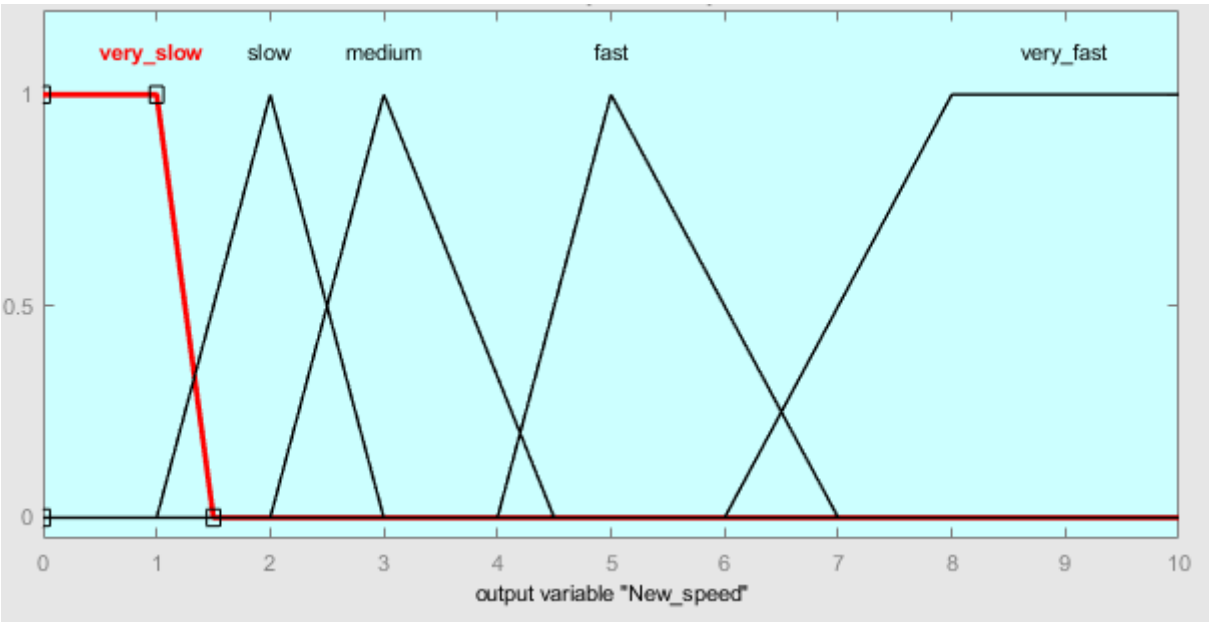
Fuzzy inference engine:

- When the distance to destination is very close, no matter the angle to destination, the speed will be very slow, because a high speed will cause the robot to run over the destination.
- When the angle to destination is big, no matter the distance to destination, the speed will be very slow, because when the angle is big, the robot runs away from the destination.
- When the angle is clockwise to the destination, the robot will turn clockwise and when the angle is counterclockwise to the destination, the robot will turn counterclockwise
  - When the angle is big, angular velocity will be fast because the robot can turn quickly to the right direction.
  - When the angle is small, angular velocity will be slow because the robot would turn away from the destination if the angular velocity is high.

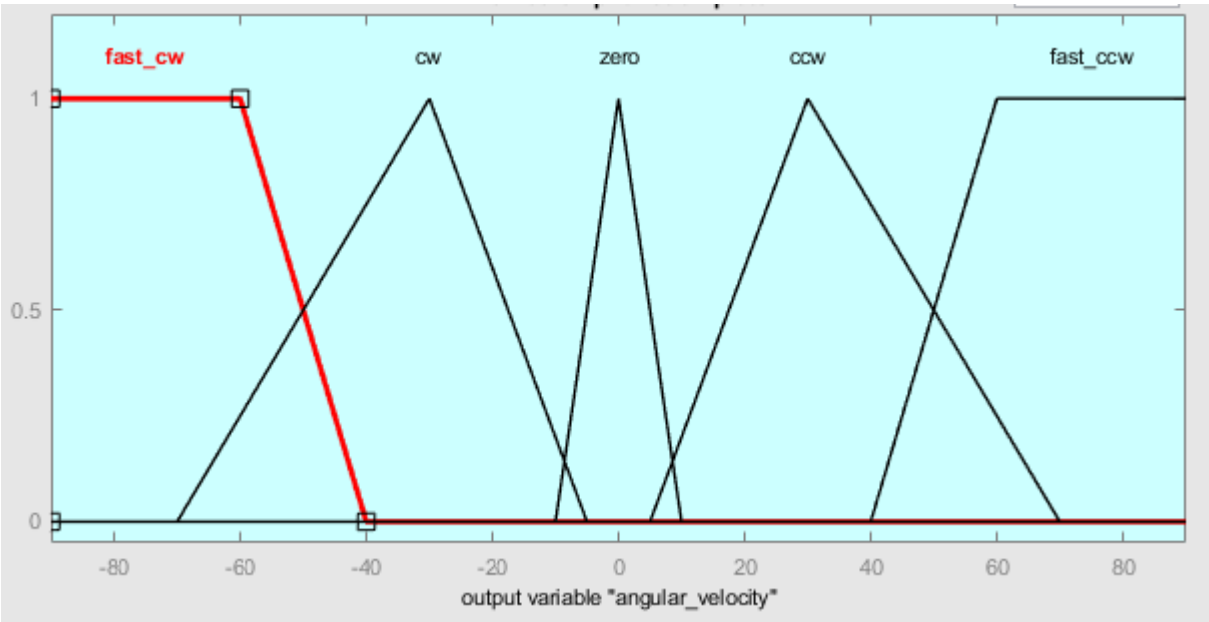
Defuzzifier design:

Mamdani is used to determine the output of the FIE. In Mamdani, the output of each rule to be a fuzzy logic set. There are two output membership functions, each with 5 sets.

- Speed
  - Very slow
  - Slow
  - Medium
  - Fast
  - Very fast



- Angular Velocity
  - Fast clockwise
  - Clockwise
  - Slow
  - Counterclockwise
  - Fast counterclockwise



The defuzzification method is centroid of gravity.

Result of the fuzzy system:

- First test case:
  - starting point (0, 0)
  - destination (10, 0)

  Result:

  Time: 0
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
  New Speed = 0.608 m/s
  New angular velocity = -70.040 degree/s

  Time: 0.1000
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
  New Speed = 0.608 m/s
  New angular velocity = -70.040 degree/s

  Time: 0.2000
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
  New Speed = 0.608 m/s
  New angular velocity = -70.040 degree/s

  Time: 0.3000
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
  New Speed = 0.608 m/s
  New angular velocity = -70.040 degree/s

  Time: 0.4000
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
  New Speed = 0.608 m/s
  New angular velocity = -70.040 degree/s

  Skip…

  Time: 16.2000
  Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
  Weight of angle membership function = [0.000 0.009 0.461 0.000 0.000]
  New Speed = 0.667 m/s
  New angular velocity = -2.665 degree/s

Time: 16.3000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.009 0.459 0.000 0.000]
New Speed = 0.667 m/s
New angular velocity = -2.817 degree/s

Time: 16.4000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.010 0.456 0.000 0.000]
New Speed = 0.667 m/s
New angular velocity = -2.988 degree/s

Time: 16.5000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.010 0.453 0.000 0.000]
New Speed = 0.668 m/s
New angular velocity = -3.180 degree/s

Total Time:
16.6000

- Second test case
  - Starting location: (100, 100)
  - Destination: (-100, -100)

Result:

Time: 0
Weight of distance membership function = [0.000 0.869 0.000 0.000 0.000]
Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
New Speed = 0.621 m/s
New angular velocity = -69.490 degree/s

Time: 0.1000
Weight of distance membership function = [0.000 0.868 0.000 0.000 0.000]
Weight of angle membership function = [1.000 0.000 0.000 0.000 0.000]
New Speed = 0.621 m/s
New angular velocity = -69.490 degree/s

Time: 0.2000
Weight of distance membership function = [0.000 0.868 0.000 0.000 0.000]
Weight of angle membership function = [1.000 0.052 0.000 0.000 0.000]
New Speed = 0.621 m/s
New angular velocity = -67.322 degree/s

Time: 0.3000
Weight of distance membership function = [0.000 0.868 0.000 0.000 0.000]
Weight of angle membership function = [1.000 0.141 0.000 0.000 0.000]
New Speed = 0.621 m/s
New angular velocity = -64.141 degree/s

Time: 0.4000
Weight of distance membership function = [0.000 0.868 0.000 0.000 0.000]
Weight of angle membership function = [1.000 0.227 0.000 0.000 0.000]
New Speed = 0.621 m/s
New angular velocity = -61.677 degree/s

Skip…

Time: 242.7000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.009 0.458 0.000 0.000]
New Speed = 0.667 m/s
New angular velocity = -2.846 degree/s

Time: 242.8000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.010 0.456 0.000 0.000]
New Speed = 0.667 m/s
New angular velocity = -3.020 degree/s

Time: 242.9000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.011 0.453 0.000 0.000]
New Speed = 0.668 m/s
New angular velocity = -3.216 degree/s

Total Time:
243.0000

- Third test case
  - Starting location: (0, 0)
  - Destination: (-2345, 3456)

Result:

Time: 0
Weight of distance membership function = [0.000 0.000 0.000 0.000 1.000]
Weight of angle membership function = [0.000 0.000 0.000 0.011 1.000]
New Speed = 0.661 m/s
New angular velocity = 69.575 degree/s

Time: 0.1000
Weight of distance membership function = [0.000 0.000 0.000 0.000 1.000]
Weight of angle membership function = [0.000 0.000 0.000 0.104 1.000]
New Speed = 1.025 m/s
New angular velocity = 66.273 degree/s

Time: 0.2000
Weight of distance membership function = [0.000 0.000 0.000 0.000 1.000]
Weight of angle membership function = [0.000 0.000 0.000 0.192 1.000]
New Speed = 1.263 m/s
New angular velocity = 63.704 degree/s

Time: 0.3000
Weight of distance membership function = [0.000 0.000 0.000 0.000 1.000]
Weight of angle membership function = [0.000 0.000 0.000 0.277 1.000]
New Speed = 1.428 m/s
New angular velocity = 61.657 degree/s

Time: 0.4000
Weight of distance membership function = [0.000 0.000 0.000 0.000 1.000]
Weight of angle membership function = [0.000 0.000 0.000 0.359 1.000]
New Speed = 1.547 m/s
New angular velocity = 60.002 degree/s

Skip…

Time: 912.8000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.000 0.461 0.009 0.000]
New Speed = 0.667 m/s
New angular velocity = 2.629 degree/s

Time: 912.9000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.000 0.459 0.009 0.000]
New Speed = 0.667 m/s
New angular velocity = 2.776 degree/s

Time: 913.0000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.000 0.457 0.010 0.000]
New Speed = 0.667 m/s
New angular velocity = 2.942 degree/s

```
-------------------------------------------------
Time:      913.1000
Weight of distance membership function = [1.000 0.000 0.000 0.000 0.000]
Weight of angle membership function = [0.000 0.000 0.454 0.010 0.000]
New Speed = 0.668 m/s
New angular velocity = 3.128 degree/s
-------------------------------------------------
Total Time:
   913.2000
```

Ethical concern:

The speed of the robot should not be too high, because we don't want the robot to hurt people and this protects the robot itself. Therefore, I set the max speed to be 10 m/s. The robot should slow down to prevent accident whenever it detects obstacles near it, but the robot does not have the sensor that detects obstacles, so it can not slow down in this situation. For future improvement, the robot should be able to sense obstacles.