FUZZY CONTROLLER

Bodhiswattwa Basu

OCTOBER 1, 2021

CSE 454: CI FOR CEN

About the System

Robby the Robot™ is a robot based on fuzzy logic control. Fuzzy logic provides a way of dealing with imprecision and nonlinearity in complex control situations. There is a significant process to fuzzy logic – fuzzification, defuzzification and fuzzy rule inference.

The two inputs that Robby™ takes in are the x- and y-coordinates of the destination. The system will output the linear velocity converted to m/s that Robby™ would be travelling at to reach the destination and the forward force required for Robby™ to reach the destination. The system will also output the weight of the membership function as Robby™ approaches his destination. Once Robby™ has reached his destination, he will announce that he has!

The constraints that this system has established is that its output is inherently inaccurate because it does not utilize crisp inputs to establish a Boolean relationship. This causes the system to be unreliable in more complication applications and in some cases, unsafe.

Process of Fuzzification

Physics Equations for Movement of Robby™

The coordinates that are inputted by the user are then used for further calculation of the distance to the destination.

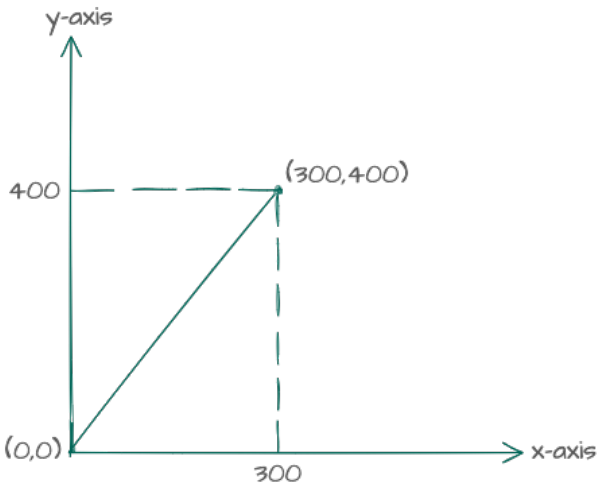


Figure 1 - Coordinates of Example Destination

To calculate the shortest distance from the initial point to the destination, the Pythagorean theorem can be applied.

$$(shortest\ distance\ to\ destination)^2 = (x - coordinates)^2 + (y - coordinates)^2$$

To calculate the forward force that Robby™ requires to move to the destination is determined by Newton's Second Law of Motion, which is denoted by,

$$Force = Mass * Acceleration$$

The assumption here is made about Robby's™ weight as two kilograms.

Design of the Fuzzifier

The fuzzifier was designed by fuzzifying the user inputs. Once the user inputs the x- and y-coordinates, the shortest distance to the destination is calculated by the Pythagorean theorem, and that value is then fuzzified into linguistic variables as displayed in the table below.

RANGE OF DISTANCE	LINGUISTIC VARIABLE
0 m – 250 m	Very Close (VC)
225 m – 475 m	Close (C)
425 m – 675 m	Average (A)
525 m – 875 m	Far (F)
850 m – 1050 m	Very Far (VF)

Table 1 - Membership Functions

These membership functions are trapezoidal functions which displays the best correlation.

The membership function 'Very Close (VC)' is a R-type trapezoidal function which is represented by the following piecewise relation,

$$\mu_{VC}(x) = \begin{cases} 0, & x > d \\ \dfrac{d-x}{d-c}, & c \leq x \leq d \\ 1, & x < c \end{cases}$$

The membership functions for 'Close (C)', 'Average (A)' and 'Far (F)' are trapezoidal functions which is represented by the following piecewise relation,

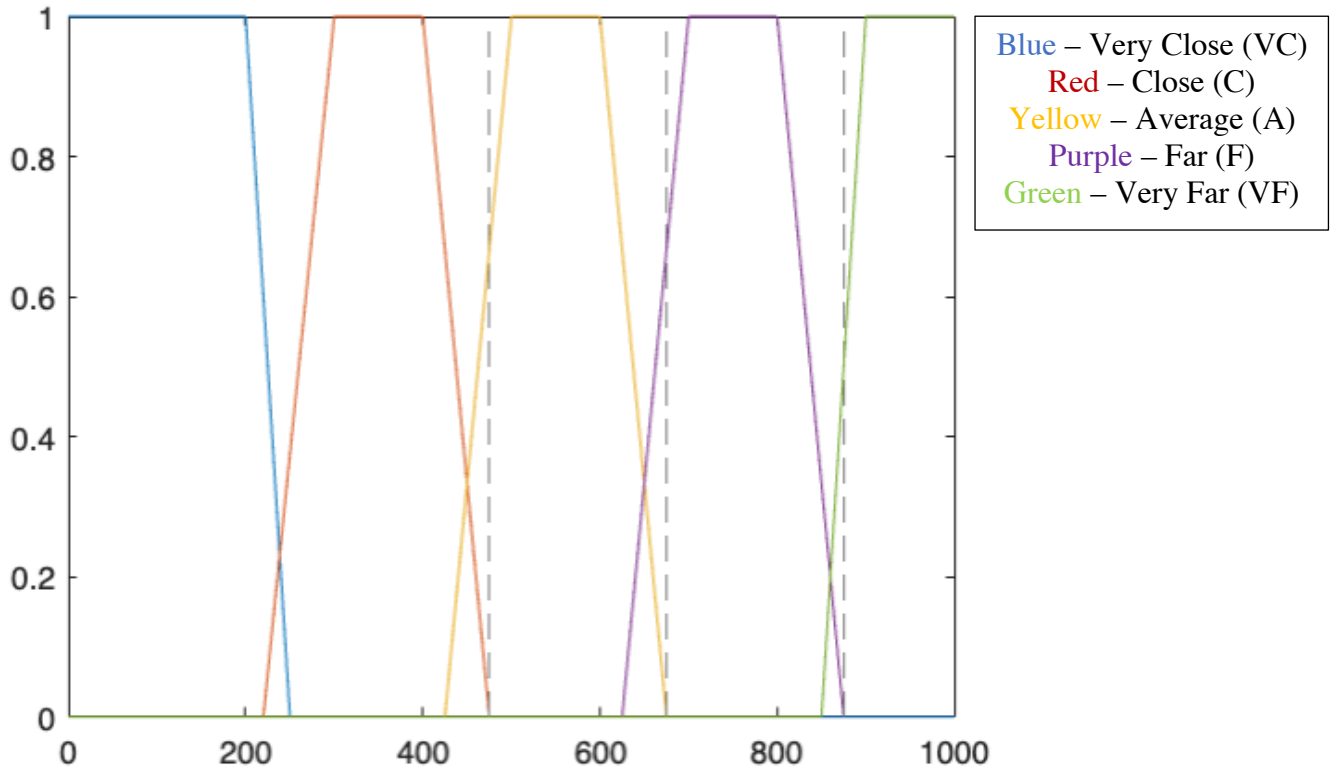
$$\mu_{C,A,F}(x) = \begin{cases} 0, & x > d \\ \dfrac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \dfrac{d-x}{d-c}, & c \leq x \leq d \end{cases}$$

The membership function for 'Very Far (VF)' is a L-type trapezoidal function which is represented by the following piecewise relation,

$$\mu_{VF}(x) = \begin{cases} 0, & x < a \\ \dfrac{x-a}{b-a}, & a \leq x \leq b \\ 1, & x > b \end{cases}$$

Please note the following for the above piecewise relations: a = Lower limit; b = Lower support limit; c = Upper support limit; d = Upper limit;

The membership functions are graphed as below:



Fuzzy Inference Engine

The fuzzy inference engine helps apply the inference rules to the fuzzified inputs in order to generate an output.

The matrix below helps explain the rules of the fuzzy inference engine.

Goal\Actual	VC	C	A	F	VF
VC	VS	VS	S	AS	F
C	S	VS	VS	AS	F
A	AS	S	VS	S	AS
F	F	AS	S	VS	S
VF	VF	F	AS	S	VS

Matrix 1 - Rule Inference Engine Matrix

Where VS, S, A, F and VF describe different velocities at which Robby™ should be travelling at. These values associated with the variables are arbitrary to an extent as they also consider the maximum distance that Robby™ can travel and the safety of the speeds that Robby™ should be allowed to reach due to safety concerns.

Speed (m/s)	LINGUISTIC VARIABLE
5	Very Slow (VS)
6.5	Slow (C)
8	Average Speed (AS)
9.5	Fast (F)
12	Very Fast (VF)

Table 2 - Speed Associated Linguistic Variables

The rules that were put in place were based on human reasoning. If the initial location is far (F) from the distance, the speed that Robby™ will employ will be fast (F), and so on.

Testing the System

Test 1:

User input:

- *x-coordinate = 300*
- *y-coordinate = 400*

Output:

```
>> FuzzyControllerProject
Enter the x-coordinates of the destination:
300
Enter the y-coordinates of the destination:
400
These are the starting coordinates == (0,0)
    "Weight of the membership function: "    "Average distance from the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
------------------------------------------------------------------------
    "100m Checkpoint Number "    "1"

    "Weight of the membership function: "    "Close to the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
------------------------------------------------------------------------
    "100m Checkpoint Number "    "2"

    "Weight of the membership function: "    "Close to the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
------------------------------------------------------------------------
    "100m Checkpoint Number "    "3"

    "Weight of the membership function: "    "Close to the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
```

```
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "4"

    "Weight of the membership function: "    "Close to the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
--------------------------------------------------------------------------
    "Weight of the membership function: "    "Close to the destination"

Speed of the robot: 8.000000 m/s
Forward Force: 16.000000 N
The destination (300.000000, 400.000000) is reached!
```

Test 2:

User input:

- *x-coordinate = 680*
- *y-coordinate = 250*

```
Enter the x-coordinates of the destination:
680
Enter the y-coordinates of the destination:
250
These are the starting coordinates == (0,0)
    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "1"

    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
```

```
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "2"

    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "3"

    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "4"

    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
--------------------------------------------------------------------------
    "100m Checkpoint Number "    "5"

    "Weight of the membership function: "    "Far from the destination"

Speed of the robot: 9.500000 m/s
Forward Force: 19.000000 N
--------------------------------------------------------------------------
     "100m Checkpoint Number "    "6"

     "Weight of the membership function: "    "Far from the destination"

 Speed of the robot: 9.500000 m/s
 Forward Force: 19.000000 N
 --------------------------------------------------------------------------
     "Weight of the membership function: "    "Far from the destination"

 Speed of the robot: 9.500000 m/s
 Forward Force: 19.000000 N
 The destination (680.000000, 250.000000) is reached!
```

Test 3:

User input:

- *x-coordinate = 800*
- *y-coordinates = 1000*

Output:

```
>> FuzzyControllerProject
Enter the x-coordinates of the destination:
800
Enter the y-coordinates of the destination:
1000
Coordinates are too large, please run program again and re-enter coordinates.
```

Ethical Issues and Considerations

Of the ethical issues taken into consideration, one of them is the maximum distance Robby™ should be allowed to move, due to the portability and movability factors, battery life and power consumption factors and more complicated traffic patterns and unknown pedestrian walking patterns.

Other ethical issues that are taken into consideration are the weather effects of Robby™ -- if it would be able to withstand harsh weather conditions.

It does not have a system that helps with obstacle detection which is inherently dangerous for pedestrians or wildlife.

Since Robby™ weighs about 2kg in this system, it will not be feasible to let it out on a non-test field because it would not have the capacity to align with the speed of society.