ECE 478
Homework #1: Fuzzy logic simulation
Professor: Marek Perkowski
Team #2: Samuel Salin
Jelon Anderson
Justin Morgan
Surendra Madulla
Saly Hakkoum
Date: November 1st, 2016

Description:

Fuzzy logic control has became an important technique in many areas. In the fuzzy logic control inputs are processed in three steps: Fuzzification, Inference and Deffuzification,

This document was originally written to introduce a Jimmy robot controller using Fuzzy logic rules.

Fuzzy logic controller:

In order to avoid obstacles in unknown dynamic environments, the inputs of our obstacle avoidance fuzzy logic controller are the distance between the robot and the obstacles and the angle between the robot and the obstacle. These distances and angles are acquired by a sensor (camera).

Fuzzy Sets:

The direction variable is divided between the sets: {Left, Middle, and Right}.

The distance variable is divided between the sets: {VeryFar, Far, Middle, Close, VeryClose}.

The final output will be one of the following commands: {MOVEFORWARD, TURN LEFT, TURN RIGHT, STOP, SLOW DOWN}.

Fuzzy Rules:

1. RULE 1
   IF (Angle is Right) AND (Distance is VERY FAR)
   THEN MOVE FORWARD

2. RULE 2
   IF (Angle is Right) AND (Distance is FAR)
   THEN MOVE FORWARD

3. RULE 3
   IF (Angle is Right) AND (Distance is CLOSE)
   THEN SLOW DOWN

4. RULE 4
   IF (Angle is Right) AND (Distance is VERY CLOSE)
   THEN TURN RIGHT

5. RULE 5
   IF (Angle is Right) AND (Distance is ZERO)
   THEN STOP

6. RULE 6
   IF (Angle is Middle) AND (Distance is VERY FAR)
   THEN MOVE FORWARD

7. RULE 7
   IF (Angle is Middle) AND (Distance is FAR)
   THEN Move Forward

8. RULE 8
   IF (Angle is Middle) AND (Distance is CLOSE)
   THEN SLOW DOWN

9. RULE 9
   IF (Angle is Middle) AND (Distance is VERY CLOSE)
   THEN SLOW DOWN

10. RULE 10
    IF (Angle is Middle) AND (Distance is ZERO)
    THEN Stop

11. RULE 11
    IF (Angle is Left) AND (Distance is VERY FAR)
    THEN MOVE FORWARD

12. RULE 12
    IF (Angle is Left) AND (Distance is FAR)
    THEN MOVE FORWARD

13. RULE 13
    IF (Angle is Left) AND (Distance is CLOSE)
    THEN SLOW DOWN

14. RULE 14
    IF (Angle is Left) AND (Distance is VERY CLOSE)
    THEN TURN RIGHT

15. RULE 15
    IF (Angle is Left) AND (Distance is ZERO)
    THEN STOP

<u>Fuzz table:</u>

	Very Close	Close	Middle	Far	Very Far
Right	Stop	Turn Left	Turn Left	Move Forward	Move Forward
Middle	Stop	Slow Down	Slow Down	Move Forward	Move Forward
Left	Stop	Turn Right	Turn Right	Move Forward	Move Forward

Fuzzy membership function:

Our fuzzy membership function is divided up into two section. The first section converts the distance value to a fuzzy value. We chose 100 cm to be the maximum distance between the robot and the obstacle to be detected by our sensor. Then we divided up this maximum into five different ranges. Values in each range gets a membership value of 1/5.

Membership value for range 0-20 cm = 1
Membership value for range 20-40 cm = 4/5
Membership value for range 40-60 cm = 3/5
Membership value for range 60-80 cm = 2/5
Membership value for range 80-100 cm = 1/5

For any value that does not belong to any of the ranges, the membership value will be equal to zero.

The second section converts the angle value to a fuzzy value. We chose 180 degrees to be the maximum angle between the robot and the final destination to be detected by our sensor. Then we divided up this maximum into three different ranges.

Membership value for range 0-20 cm = 0.3
Membership value for range 20-40 cm = 1
Membership value for range 40-60 cm = 0.7

For any value that does not belong to any of the ranges, the membership value will be equal to zero

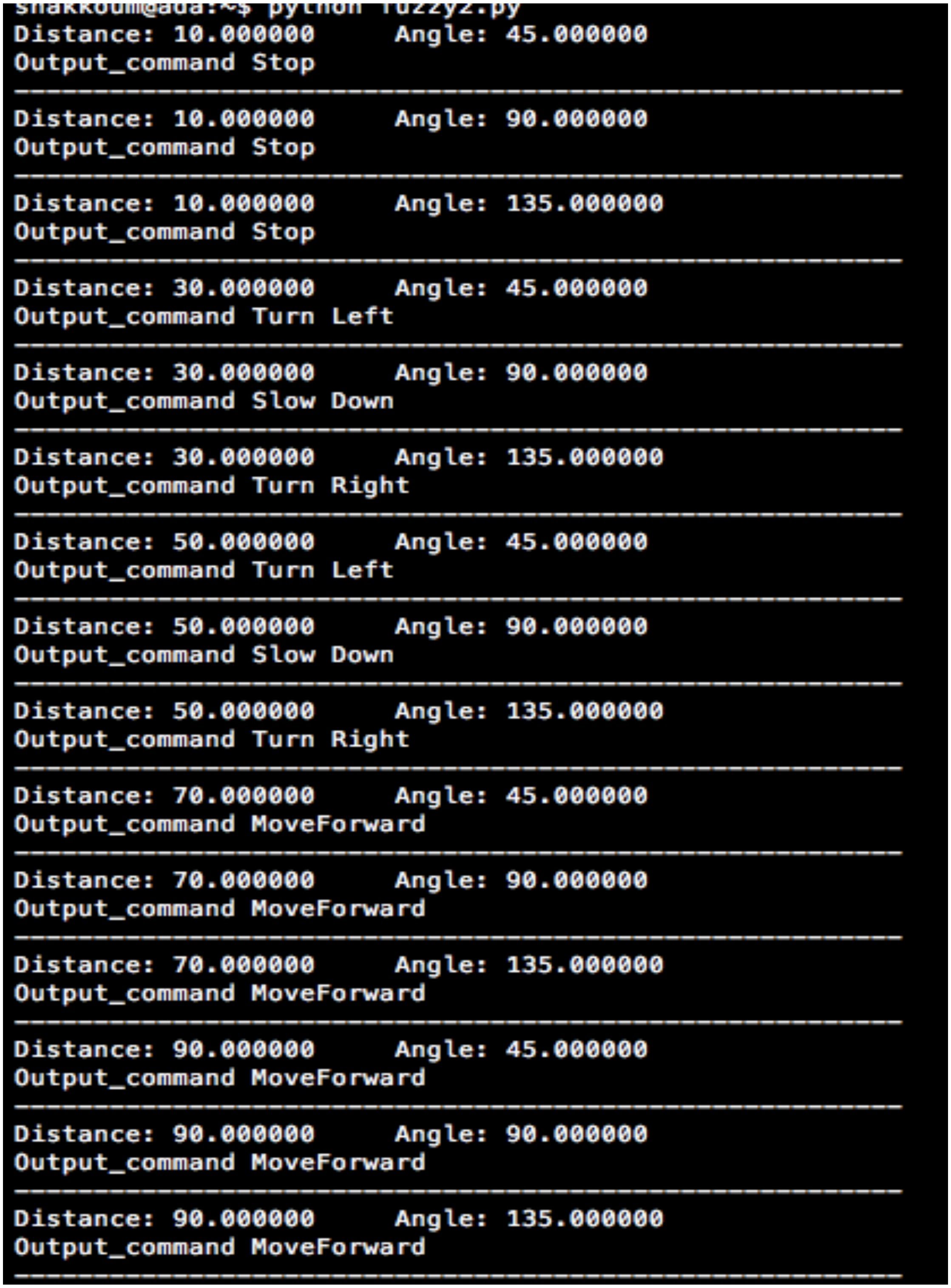
Implementation:

```
from __future__ import division

class FuzzySet(object):
    def __init__(self, x1, x2):
        self.x1 = x1
        self.x2 = x2

    def membership(self, x, y):
        temp = int(y)
        if (temp == 100):
            if (x < self.x1 or x > self.x2):
                return 0.0
            elif (x > 0 and x <= 20):
                return 1
            elif (x > 20 and x <= 40):
                return 4/5
            elif (x > 40 and x <= 60):
                return 3/5
            elif (x > 60 and x <= 80):
                return 2/5
            elif (x > 80 and x <= 100):
                return 1/5

        if (temp == 180):
            if (x < self.x1 or x > self.x2):
                return 0.0
            if (x > 0 and x <= 60):
                return 0.3
            if (x > 60 and x <= 120):
                return 1.0
            if (x > 120 and x <= 180):
                return 0.7

def Main():
# Settin up the input variables
# Distance are measured in centimeters
    VeryClose       = FuzzySet(0, 20)
    Close           = FuzzySet(20, 40)
    Medium              = FuzzySet(40, 60)
    Far             = FuzzySet(60, 80)
    VeryFar             = FuzzySet(80,100)
```

```
# Angles are measured in degrees
    Right   = FuzzySet(0, 60)
    Middle  = FuzzySet(60, 120)
    Left    = FuzzySet(120, 180)

    data = {}
    count = 0
    with open("fuzzy_data.txt","r") as fin:
        for line in fin:
            line = line.strip('\n')
            parts = line.split(',')
            temp = {}
            temp['Distance'] = int(parts[0])
            temp['Angle'] = int(parts[1])
            data[count] = temp
            count += 1

    for i in range(0,15):
        Output_command = ""
        Output = 0.0

        Angle = float(data[i]['Angle'])
        Distance = float(data[i]['Distance'])
        # Fuzzification Rules
        Output += Middle.membership(Angle,180) * VeryFar.membership(Distance,100)
        Output += Middle.membership(Angle,180) * Far.membership(Distance,100)
        Output += Middle.membership(Angle,180) * Medium.membership(Distance,100)
        Output += Middle.membership(Angle,180) * Close.membership(Distance,100)
        Output += Middle.membership(Angle,180) * VeryClose.membership(Distance,100)

        Output += Right.membership(Angle,180) * VeryFar.membership(Distance,100)
        Output += Right.membership(Angle,180) * Far.membership(Distance,100)
        Output += Right.membership(Angle,180) * Medium.membership(Distance,100)
        Output += Right.membership(Angle,180) * Close.membership(Distance,100)
        Output += Right.membership(Angle,180) * VeryClose.membership(Distance,100)

        Output += Left.membership(Angle,180) * VeryFar.membership(Distance,100)
        Output += Left.membership(Angle,180) * Far.membership(Distance,100)
        Output += Left.membership(Angle,180) * Medium.membership(Distance,100)
        Output += Left.membership(Angle,180) * Close.membership(Distance,100)
        Output += Left.membership(Angle,180) * VeryClose.membership(Distance,100)
        # Defuzzification
        if Output != 0:
            if Output == 0.42 or Output == 0.5599999999999999:
                Output_command = "Turn Right"
            elif Output == 0.24 or Output == 0.18:
                Output_command = "Turn Left"
            elif Output == 0.6 or Output == 0.8:
                Output_command = "Slow Down"
            elif Output == 0.3 or Output == 1.0 or Output == 0.7:
                Output_command = "Stop"
            else:
                Output_command = "MoveForward"

        print ("Distance: %f\tAngle: %f\nOutput_command %s" %(Distance, Angle, Output_command))
        print ("-------------------------------------------------------")
if __name__ == "__main__":
    Main()
```

Results:

The output of the python script will look like this:



```
shakkoum@ada:~$ python fuzzy2.py
Distance: 10.000000     Angle: 45.000000
Output_command Stop
--------------------------------------------------------
Distance: 10.000000     Angle: 90.000000
Output_command Stop
--------------------------------------------------------
Distance: 10.000000     Angle: 135.000000
Output_command Stop
--------------------------------------------------------
Distance: 30.000000     Angle: 45.000000
Output_command Turn Left
--------------------------------------------------------
Distance: 30.000000     Angle: 90.000000
Output_command Slow Down
--------------------------------------------------------
Distance: 30.000000     Angle: 135.000000
Output_command Turn Right
--------------------------------------------------------
Distance: 50.000000     Angle: 45.000000
Output_command Turn Left
--------------------------------------------------------
Distance: 50.000000     Angle: 90.000000
Output_command Slow Down
--------------------------------------------------------
Distance: 50.000000     Angle: 135.000000
Output_command Turn Right
--------------------------------------------------------
Distance: 70.000000     Angle: 45.000000
Output_command MoveForward
--------------------------------------------------------
Distance: 70.000000     Angle: 90.000000
Output_command MoveForward
--------------------------------------------------------
Distance: 70.000000     Angle: 135.000000
Output_command MoveForward
--------------------------------------------------------
Distance: 90.000000     Angle: 45.000000
Output_command MoveForward
--------------------------------------------------------
Distance: 90.000000     Angle: 90.000000
Output_command MoveForward
--------------------------------------------------------
Distance: 90.000000     Angle: 135.000000
Output_command MoveForward
--------------------------------------------------------
```

Where the test values are imported from a text file.