

Fuzzy Logic Simulation  
ECE 478 Homework #3

Phil Lamb  
pjl@pdx.edu

December 9, 2012

# Contents

|     |                            |   |
|-----|----------------------------|---|
| 1   | Introduction               | 3 |
| 1.1 | Fuzzy Logic . . . . .      | 3 |
| 2   | Simulation                 | 3 |
| 2.1 | Fuzzy Sets . . . . .       | 3 |
| 2.2 | Rules . . . . .            | 4 |
| 3   | Results                    | 4 |
| 4   | Implementation             | 5 |
| A   | Implementation Source Code | 6 |

# 1 Introduction

This report summarizes the implementation and results of a fuzzy logic simulation.

## 1.1 Fuzzy Logic

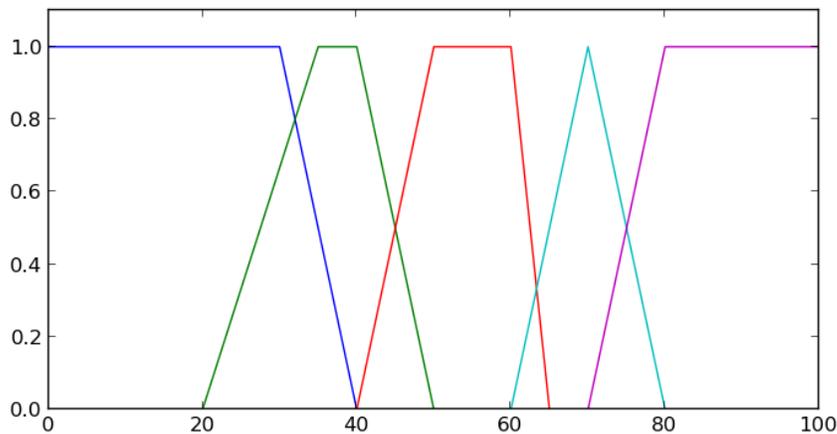
Fuzzy logic attempts to apply a more human-like reasoning to problems. Rather than implementing solutions in absolute yes or no values as in formal logic, fuzzy logic assigns levels of membership to fuzzy sets. For example, when determining whether an object is “near” or “far” the distinction between the two is not at some absolute boundry. Membership in fuzzy sets is often described by a triangular or trapezoidal function with the value of the variable as the independant variable, and the membership level (typically a value between 0 and 1 or 0% and 1%) on the dependant axis.

## 2 Simulation

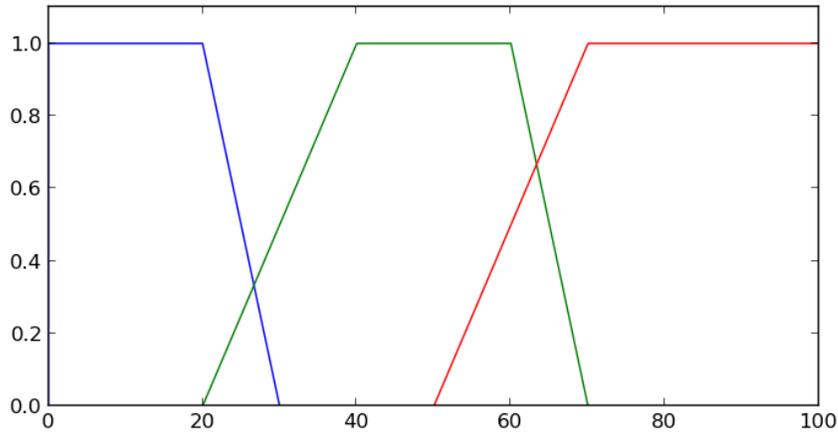
In this simulation I demonstrate a simple two-variable scenario with a single output variable: a heating system that can measure the current temperature and the level of activity in the heated area. The output is the output of the heater.

### 2.1 Fuzzy Sets

The temperature variable is divided between the sets: Hot, Warm, Cool, Cold, and Freezing. The following figure shows the values for these sets.



The activity variable is divided into: Busy, Medium, and Quiet. The following figure shows the values for these sets.



## 2.2 Rules

In a fuzzy logic system rules are implemented in a way similar to natural language. The rules for this system are listed below:

- If **hot** turn heater off
- If **warm** turn heater off
- If **cool** turn heater on low
- If **cold** turn heater on medium
- If **freezing** turn heater on high
- If **busy** turn heater on low
- If **medium** (activity) turn heater on medium
- If **quiet** turn heater off

In actual implementation the membership value of a variable is multiplied by the heater level.

## 3 Results

Below is a sample output of the program with a number of test cases:

```

Temperature: 70.000000 Activity: 2.000000 Heater: 0.000000
Temperature: 20.000000 Activity: 98.000000 Heater: 8.500000
Temperature: 50.000000 Activity: 40.000000 Heater: 3.500000
Temperature: 60.000000 Activity: 65.300000 Heater: 2.410000
Temperature: 31.200000 Activity: 1.300000 Heater: 5.033333
Temperature: 63.900000 Activity: 77.100000 Heater: 1.220000

```

Temperature: 25.100000 Activity: 39.900000 Heater: 6.662500

## 4 Implementation

The simulation was implemented in python. The source code can be found in appendix 1.

## A Implementation Source Code

```
#####d#####
# Phil Lamb
# ECE478 Homework 3
# Fuzzy Logic Simulation
#####

class FuzzySet(object):
    def __init__(self, x1, x2, x3, x4):
10     self.x1 = x1
        self.x2 = x2
        self.x3 = x3
        self.x4 = x4

    def membership(self, x):
        if x < self.x1 or x > self.x4:
            return 0.0
        if x >= self.x2 and x <= self.x3:
            return 1.0
20     if x < self.x2:
            return (self.x2 - x) / (self.x2 - self.x1)
        if x > self.x3:
            return (self.x4 - x) / (self.x4 - self.x3)

# set up variables
#temperature in degrees F
temperature = 0.0
30 hot = FuzzySet(70, 80, 200, 200)
warm = FuzzySet(60, 70, 70, 80)
cool = FuzzySet(40, 50, 60, 65)
cold = FuzzySet(20, 35, 40, 50)
freezing = FuzzySet(-100, -100, 30, 40)

#activity in arbitrary units between 0 and 100
activity = 0.0
busy = FuzzySet(50, 70, 100, 100)
medium = FuzzySet(20, 40, 60, 70)
40 quiet = FuzzySet(0, 0, 20, 30)

#heater level
heater = 0.0
```

```

heater_off = 0.0
heater_low = 1.0
heater_med = 2.5
heater_high = 5.0

#test cases (temperature, activity)
50 testcases = (
    (70.0, 2.0),
    (20.0, 98.0),
    (50.0, 40.0),
    (60.0, 65.3),
    (31.2, 1.3),
    (63.9, 77.1),
    (25.1, 39.9),
    )

60 #main logic
for t in testcases:
    heater = 0.0
    #if hot, turn heater off
    heater += heater_off * hot.membership(t[0])
    #if warm, turn heater off
    heater += heater_off * warm.membership(t[0])
    #if cool, turn heater on low
    heater += heater_low * cool.membership(t[0])
    #if cold, turn heater on medium
70 heater += heater_med * cold.membership(t[0])
    #if freezing, turn heater on high
    heater += heater_high * freezing.membership(t[0])

    #if busy, turn heater on low
    heater += heater_low * busy.membership(t[1])
    #if medium activity, turn heater on med
    heater += heater_med * medium.membership(t[1])
    #if quiet, turn heater off
    heater += heater_off * quiet.membership(t[1])

80 print("Temperature: %f\tActivity: %f\tHeater: %f" % (t[0], t[1], heater))

```