Shri Vile Parle Kelavani Mandal's

DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING

(Autonomous College Affiliated to the University of Mumbai)

NAAC Accredited with "A" Grade (CGPA : 3.18)

Department of Computer Science and Engineering (Data Science)

Subject: Artificial Intelligence (DJ19DSC502)

AY: 2023-24

Jhanvi Parekh

60009210033

D11

Experiment 8

(Fuzzy Logic)

Aim: Design and Implement a Fuzzy Logic Controller.

Theory:

A **fuzzy control system** is a control system based on fuzzy logic—a mathematical system that analyzes analog input values in terms of logical variables that take on continuous values between 0 and 1, in contrast to classical or digital logic, which operates on discrete values of either 1 or 0 (true or false, respectively.

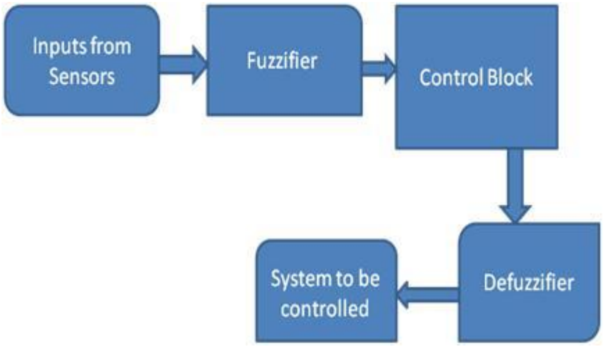
Fuzzy logic works on the concept of deciding the output based on assumptions. It works based on sets. Each set represents some linguistic variables defining the possible state of the output. Each possible state of the input and the degrees of change of the state are a part of the set, depending upon which the output is predicted. It works on the principle of If-else-the, i.e. If A AND B Then Z.

Suppose we want to control a system where the output can be anywhere in the set X, with a generic value x, such that x belongs to X. Consider a particular set A which is a subset of X such that all members of A belong to the interval 0 and 1. The set A is known as a fuzzy set and the value of $f_A(x)$ at x denotes the degree of membership of x in that set. The output is decided based on the degree of membership of x in the set. This assigning of membership depends on the assumption of the outputs depending on the inputs and the rate of change of the inputs.

These fuzzy sets are represented graphically using membership functions and the output is decided based on the degree of membership in each part of the function. The membership of the sets is decided by the IF-Else logic.

Fuzzy Control System

A fuzzy control system consists of the following components:



A Fuzzifier which transforms the measured or the input variable

A Controller performs the fuzzy logic operation of assigning the outputs based on the linguistic information. It performs approximate reasoning based on the human way of interpretation to achieve control logic. The controller consists of the knowledge base and the inference engine. The knowledge base consists of the membership functions and the fuzzy rules, which are obtained by knowledge of the system operation according to the environment.

The Defuzzifier converts this fuzzy output to the required output to control the system.

Lab Assignment to do:
Design a fuzzy Logic Controller to set the time of a washing machine.
LINK: https://colab.research.google.com/drive/1AkUFeIuZIYVjGsVcU9ECRidrnFdTnV3Y?usp=sharing

Installing libraries

```
!pip install fuzzywuzzy
```

```
Collecting fuzzywuzzy
  Downloading fuzzywuzzy-0.18.0-py2.py3-none-any.whl (18 kB)
Installing collected packages: fuzzywuzzy
Successfully installed fuzzywuzzy-0.18.0
```

```
!pip install scikit-fuzzy
```

```
Collecting scikit-fuzzy
  Downloading scikit-fuzzy-0.4.2.tar.gz (993 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 994.0/994.0 kB 7.9 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: numpy>=1.6.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.23.5)
Requirement already satisfied: scipy>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (1.11.3)
Requirement already satisfied: networkx>=1.9.0 in /usr/local/lib/python3.10/dist-packages (from scikit-fuzzy) (3.2.1)
Building wheels for collected packages: scikit-fuzzy
  Building wheel for scikit-fuzzy (setup.py) ... done
  Created wheel for scikit-fuzzy: filename=scikit_fuzzy-0.4.2-py3-none-any.whl size=894078 sha256=1acf51061952424e30726d3dd3d9f216686039
  Stored in directory: /root/.cache/pip/wheels/4f/86/1b/dfd97134a2c8313e519bcebd95d3fedc7be7944db022094bc8
Successfully built scikit-fuzzy
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.4.2
```

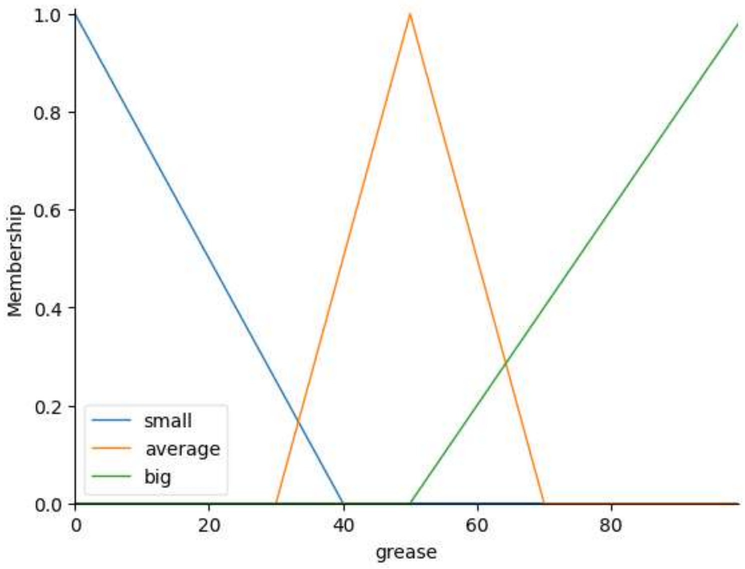
```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
```

DESCRIPTORS FOR INPUT - DIRT, GREASE AND WASH TIME

```
grease = ctrl.Antecedent(np.arange(0, 100, 1), 'grease')
dirt = ctrl.Antecedent(np.arange(0, 100, 1), 'dirt')
time = ctrl.Consequent(np.arange(0, 100, 1), 'washing time')
```

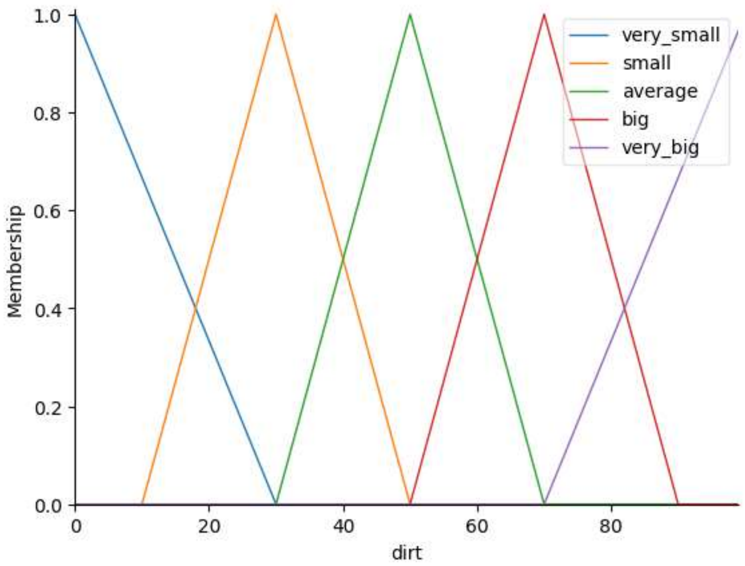
MEMEBERSHIP FUNCTION:

```
grease['small'] = fuzz.trimf(grease.universe, [0, 0, 40])
grease['average'] = fuzz.trimf(grease.universe, [30, 50, 70])
grease['big'] = fuzz.trimf(grease.universe, [50, 100, 100])
grease.view()
```

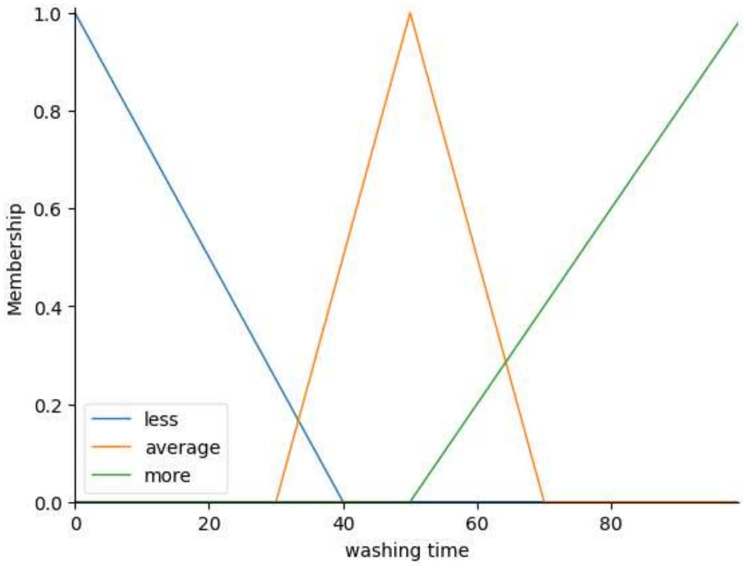


```
dirt['very_small'] = fuzz.trimf(dirt.universe, [0, 0, 30])
dirt['small'] = fuzz.trimf(dirt.universe, [10, 30, 50])
dirt['average'] = fuzz.trimf(dirt.universe, [30, 50, 70])
```

```
dirt['big'] = fuzz.trimf(dirt.universe, [50, 70, 90])
dirt['very_big'] = fuzz.trimf(dirt.universe, [70, 100, 100])
dirt.view()
```



```
time['less'] = fuzz.trimf(time.universe, [0, 0, 40])
time['average'] = fuzz.trimf(time.universe, [30, 50, 70])
time['more'] = fuzz.trimf(time.universe, [50, 100, 100])
time.view()
```



```
rule1 = ctrl.Rule(antecedent=(dirt['very_small'] & grease['small']), consequent=time['less'])
rule2 = ctrl.Rule(antecedent=(dirt['very_small'] & grease['average']), consequent=time['average'])
rule3 = ctrl.Rule(antecedent=(dirt['very_small'] & grease['big']), consequent=time['more'])
rule4 = ctrl.Rule(antecedent=(dirt['small'] & grease['small']), consequent=time['average'])
rule5 = ctrl.Rule(antecedent=(dirt['small'] & grease['average']), consequent=time['more'])
rule6 = ctrl.Rule(antecedent=(dirt['small'] & grease['big']), consequent=time['more'])
rule7 = ctrl.Rule(antecedent=(dirt['average'] & grease['small']), consequent=time['average'])
rule8 = ctrl.Rule(antecedent=(dirt['average'] & grease['average']), consequent=time['more'])
rule9 = ctrl.Rule(antecedent=(dirt['average'] & grease['big']), consequent=time['more'])
rule10 = ctrl.Rule(antecedent=(dirt['big'] & grease['small']), consequent=time['average'])
rule11 = ctrl.Rule(antecedent=(dirt['big'] & grease['average']), consequent=time['more'])
rule12 = ctrl.Rule(antecedent=(dirt['big'] & grease['big']), consequent=time['more'])
rule13 = ctrl.Rule(antecedent=(dirt['very_big'] & grease['small']), consequent=time['more'])
rule14 = ctrl.Rule(antecedent=(dirt['very_big'] & grease['average']), consequent=time['more'])
rule15 = ctrl.Rule(antecedent=(dirt['very_big'] & grease['big']), consequent=time['more'])
```

```
washing_machine = ctrl.ControlSystem([rule1, rule2, rule3,rule4,rule5,rule6,rule7,rule8,rule9,rule10,rule11,rule12,rule13,rule15,rule15])
washing_machine = ctrl.ControlSystemSimulation(washing_machine)
```

```
washing_machine.input['dirt'] = 20
washing_machine.input['grease'] = 30
washing_machine.compute()
print(f"\nWASHING TIME = {washing_machine.output['washing time']}\n")
```

```
WASHING TIME = 33.7776111944028
```

```
washing_machine.input['dirt'] = 40
washing_machine.input['grease'] = 90
washing_machine.compute()
print(f"\nWASHING TIME = {washing_machine.output['washing time']}\n")
```

```
WASHING TIME = 80.0365296803653
```

```
washing_machine.input['dirt'] = 30
washing_machine.input['grease'] = 60
washing_machine.compute()
print(f"\nWASHING TIME = {washing_machine.output['washing time']}\n")
```

```
WASHING TIME = 80.0365296803653
```