HARSHIT SHARMA D20B/51

HARSHIT SHARMA D20B/51

Introduction

Traditional control systems such as PID controllers and Boolean logic based systems are effective when the system model is precise and the environment is predictable. However in real life many systems are uncertain, nonlinear, and imprecise. Examples include air conditioner control, washing machines, car braking systems, and medical diagnosis.

To handle uncertainty and vagueness a fuzzy control system is used.

What is Fuzzy Logic

Fuzzy logic was introduced by Lotfi Zadeh in 1965. Unlike classical logic which works only with true or false values, fuzzy logic allows intermediate values between zero and one. It is based on linguistic variables such as low, medium, and high instead of exact numbers. For example, a temperature of thirty degrees can be described as partly warm and partly hot at the same time.

Components of a Fuzzy Control System

A fuzzy control system has four major components.

1. Fuzzification

This process converts crisp numerical input values into fuzzy sets. Membership functions are used to represent fuzzy sets. For example, a temperature of thirty may belong partly to the set warm and partly to the set hot.

2. Rule Base or Knowledge Base

This contains a collection of if then rules defined using expert knowledge. For example If temperature is high and humidity is high then fan speed is fast If temperature is low then fan speed is slow

3. Inference Engine

This part evaluates the rules and decides how to combine them to produce a fuzzy output. Two common inference methods are Mamdani method and Sugeno method.

4. Defuzzification

This process converts the fuzzy output into a crisp numerical value. The most common method is the centroid or center of gravity method. Other methods include mean of maximum and weighted average.

Advantages of Fuzzy Control Systems

Can handle uncertainty and imprecision

Easy to implement using expert knowledge without needing a precise mathematical model

Suitable for nonlinear and complex systems

Flexible and adaptable to multiple applications

Applications of Fuzzy Control Systems

Consumer electronics such as washing machines, air conditioners, and microwave ovens

Healthcare such as disease diagnosis and medical decision support

Automobile industry such as anti lock braking and automatic transmission

Industrial control such as robotics and process control

Smart systems such as smart homes, IoT devices, and renewable energy management

Example: Fan Speed Control

Inputs are temperature and humidity, outputs are fan speed levels. Rules can be If temperature is high and humidity is high then fan speed is fast If temperature is low then fan speed is slow

This way the fan speed changes smoothly instead of abrupt on or off operation.

Conclusion

A fuzzy control system is an intelligent system that mimics human decision making in uncertain environments. It works in four stages namely fuzzification, rule evaluation, inference, and defuzzification. It has wide applications in automation, healthcare, industry, and consumer products. Using libraries such as scikit fuzzy in Python, fuzzy control systems can be easily designed and simulated in Google Colab.

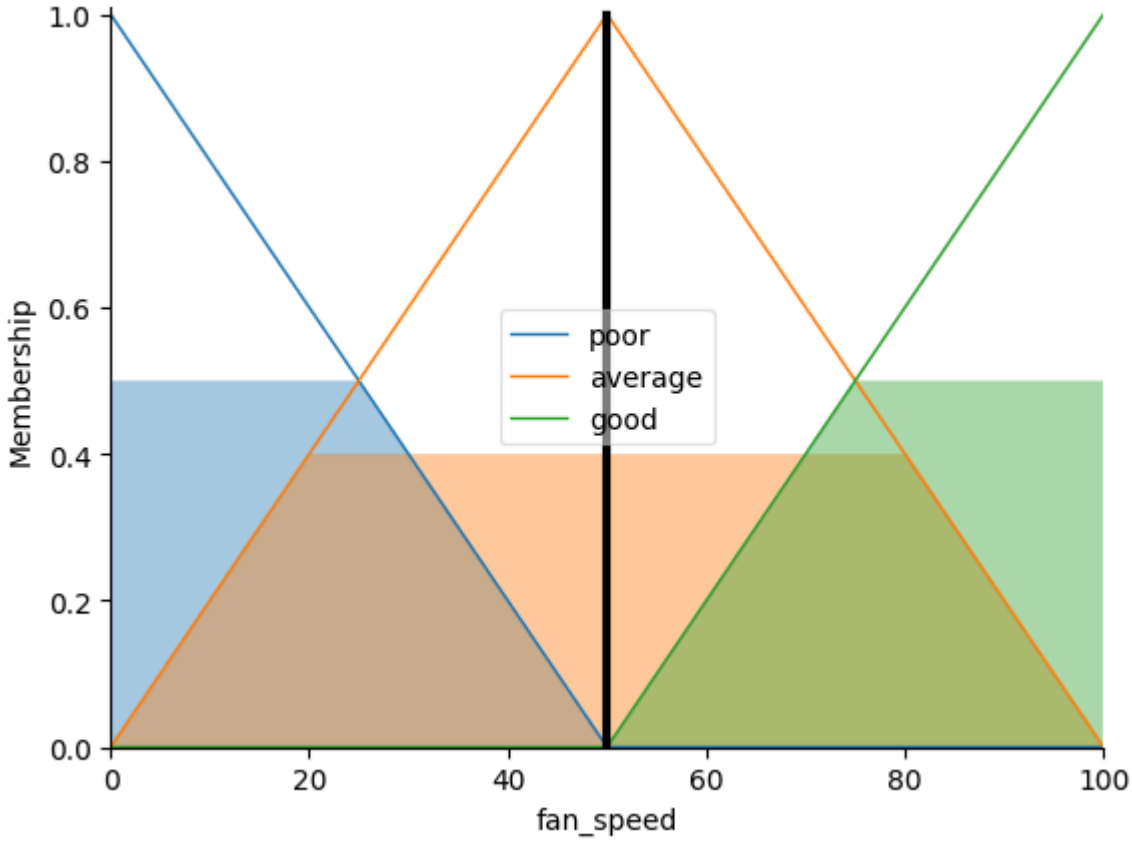
```
!pip install scikit-fuzzy matplotlib
```

```
Collecting scikit-fuzzy
  Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl.metadata (2.6 kB)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-package
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-pa
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-pack
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-pac
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (f
Downloading scikit_fuzzy-0.5.0-py2.py3-none-any.whl (920 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 920.8/920.8 kB 9.6 MB/s eta 0:00:00
Installing collected packages: scikit-fuzzy
Successfully installed scikit-fuzzy-0.5.0
```

```
import numpy as np
import skfuzzy as fuzz
```

```
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt



# Define fuzzy variables
temperature = ctrl.Antecedent(np.arange(0, 41, 1), 'temperature')
humidity = ctrl.Antecedent(np.arange(0, 101, 1), 'humidity')
fan_speed = ctrl.Consequent(np.arange(0, 101, 1), 'fan_speed')

# Define membership functions automatically
temperature.automf(3)  # Low, Medium, High
humidity.automf(3)     # Low, Medium, High
fan_speed.automf(3)    # Low, Medium, High



rule1 = ctrl.Rule(temperature['poor'] | humidity['good'], fan_speed['average'])
rule2 = ctrl.Rule(temperature['average'] & humidity['average'], fan_speed['good'])
rule3 = ctrl.Rule(temperature['good'] | humidity['poor'], fan_speed['poor'])



fan_ctrl = ctrl.ControlSystem([rule1, rule2, rule3])
fan = ctrl.ControlSystemSimulation(fan_ctrl)



# Provide inputs
fan.input['temperature'] = 30
fan.input['humidity'] = 70

# Compute output
fan.compute()

print("Fan Speed Output:", fan.output['fan_speed'])

# Show result graph
fan_speed.view(sim=fan)
plt.show()
```

```
Fan Speed Output: 50.00000000000006
```



Start coding or generate with AI.