```
==========================================
Fuzzy Systems: Water Quality Monitoring
==========================================

"""
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
# THe process of obtaining imput has been explained in Supplementary_Material_2.pdf
# Antecedent/Consequent for water quality monitoring
# functions
pH = ctrl.Antecedent(np.arange(0, 14, 1), 'pH')
DO = ctrl.Antecedent(np.arange(0, 12, 1), 'DO')
EC = ctrl.Antecedent(np.arange(0, 1100, 50), 'EC')
ORP = ctrl.Antecedent(np.arange(0, 1100, 50), 'ORP')
Temperature= ctrl.Antecedent(np.arange(0, 40, 2.5), 'Temperature')
Water_Quality = ctrl.Consequent(np.arange(0, 10, 1), 'Water_Quality')
# Generate Fuzzy membership function for antecedent
pH['NA'] = fuzz.trimf(pH.universe, [0, 0, 5.7])
pH['ADE'] = fuzz.trimf(pH.universe, [2.9, 4.0, 5.1])
pH['HACC'] = fuzz.trimf(pH.universe, [6.5, 7.6, 8.7])
pH['NA'] = fuzz.trimf(pH.universe, [8.5, 14, 14.1])
DO['NA'] = fuzz.trimf(DO.universe, [0, 0, 3])
DO['ADE'] = fuzz.trimf(DO.universe, [5.3, 6, 6.7])
DO['HACC'] = fuzz.trimf(DO.universe, [5.1, 8, 11.1])
DO['NA'] = fuzz.trimf(DO.universe, [11, 12, 12.1])
EC['NA'] = fuzz.trimf(EC.universe, [0, 0, 300])
EC['ADE'] = fuzz.trimf(EC.universe, [290, 400, 510])
EC['HACC'] = fuzz.trimf(EC.universe, [650, 740, 820])
EC['NA'] = fuzz.trimf(EC.universe, [800, 1000, 1100])
ORP['NA'] = fuzz.trimf(ORP.universe, [0, 0, 550])
ORP['ADE'] = fuzz.trimf(ORP.universe, [530, 600, 670])
ORP['HACC'] = fuzz.trimf(ORP.universe, [650, 740, 820])
ORP['NA'] = fuzz.trimf(ORP.universe, [800, 1000, 1100])
Temperature['NA'] = fuzz.trimf(Temperature.universe, [0, 0, 2])
Temperature['ADE'] = fuzz.trimf(Temperature.universe, [1.9, 5, 10])
Temperature['HACC'] = fuzz.trimf(Temperature.universe, [9, 21, 36])
Temperature['NA'] = fuzz.trimf(Temperature.universe, [35, 37.5, 40])
#Custom membership function for antecedent
# Pythonic API
Water_Quality['NA'] = fuzz.trimf(Water_Quality.universe, [0, 0, 4])
```

```
Water_Quality['ADE'] = fuzz.trimf(tip.universe, [4, 5.5, 7])
Water_Quality['HACC'] = fuzz.trimf(tip.universe, [7, 8.5, 10])
# You can see how these look with .view()for example
pH['ADE'].view()
# Fuzzy Rule Generation (For Convineance only one rule have been shown here, rest of the rules can be referred from supplementary material and can be implemented
#through same fashion
#If (pH is ADE) OR (D.O. is HACC) OR (E.C. is HACC) OR (O.R.P is HACC) OR (Temperature is ADE) then (Water Quality is ADE)
rule1 = ctrl.Rule(pH['ADE'] | DO['HACC'] | EC['HACC'] | ORP['HACC'] | Temperature['ADE'], Water_Quality['ADE'])
rule1.view()
#To create a control for one rule, However to change the rules control has to change
WaterQ= ctrl.ControlSystem([rule1])
# Simulate control
WaterQ = ctrl.ControlSystemSimulation(Water_Quality_ctrl)
# Input Process can be understood from Supplementary_Material_2, However, Input have to be averaged before supply to Fuzzy framework.
#To understand the process let consider arbitrary values
WaterQ.input['pH'] = 5.95
WaterQ.input['DO'] = 8.08
WaterQ.input['EC'] = 406
WaterQ.input['ORP'] = 735
WaterQ.input['Temperature'] = 22.8
WaterQ.compute()
print WaterQ.output['Water_Quality']
Water_Quality.view(sim=WaterQ)
```