**Name: Akash Kumar**

**PRN: 1032233339**

**9. Water Level Control in a Tank: Create a fuzzy logic controller to regulate the water level in a tank. Input variables could include inflow rate, outflow rate, and current water level, while the output is the control signal to a pump or valve.**

**Code:**

import numpy as np

import skfuzzy as fuzz

from skfuzzy import control as ctrl

import matplotlib.pyplot as plt

# Define the universe of discourse for each variable

inflow\_rate = ctrl.Antecedent(np.arange(0, 11, 1), 'inflow\_rate')

outflow\_rate = ctrl.Antecedent(np.arange(0, 11, 1), 'outflow\_rate')

current\_level = ctrl.Antecedent(np.arange(0, 101, 1), 'current\_level')

control\_signal = ctrl.Consequent(np.arange(-50, 51, 1), 'control\_signal')

# Define custom membership functions

inflow\_rate['very\_low'] = fuzz.trimf(inflow\_rate.universe, [0, 0, 2.5])

inflow\_rate['low'] = fuzz.trimf(inflow\_rate.universe, [2, 3.5, 5])

inflow\_rate['moderate'] = fuzz.trimf(inflow\_rate.universe, [4.5, 5.5, 6.5])

inflow\_rate['high'] = fuzz.trimf(inflow\_rate.universe, [6, 7.5, 9])

inflow\_rate['very\_high'] = fuzz.trimf(inflow\_rate.universe, [8.5, 10, 10])

outflow\_rate['very\_low'] = fuzz.trimf(outflow\_rate.universe, [0, 0, 2.5])

outflow\_rate['low'] = fuzz.trimf(outflow\_rate.universe, [2, 3.5, 5])

outflow\_rate['moderate'] = fuzz.trimf(outflow\_rate.universe, [4.5, 5.5, 6.5])

outflow\_rate['high'] = fuzz.trimf(outflow\_rate.universe, [6, 7.5, 9])

outflow\_rate['very\_high'] = fuzz.trimf(outflow\_rate.universe, [8.5, 10, 10])

current\_level['very\_low'] = fuzz.trimf(current\_level.universe, [0, 0, 25])

current\_level['low'] = fuzz.trimf(current\_level.universe, [20, 25, 50])

current\_level['moderate'] = fuzz.trimf(current\_level.universe, [45, 50, 55])

current\_level['high'] = fuzz.trimf(current\_level.universe, [50, 75, 100])

current\_level['very\_high'] = fuzz.trimf(current\_level.universe, [75, 100, 100])

control\_signal['decrease'] = fuzz.trimf(control\_signal.universe, [-50, -25, 0])

control\_signal['maintain'] = fuzz.trimf(control\_signal.universe, [-25, 0, 25])

control\_signal['increase'] = fuzz.trimf(control\_signal.universe, [0, 25, 50])

# Define rules

rule1 = ctrl.Rule(inflow\_rate['very\_low'] | outflow\_rate['very\_high'], control\_signal['increase'])

rule2 = ctrl.Rule(inflow\_rate['very\_high'] & current\_level['very\_low'], control\_signal['decrease'])

rule3 = ctrl.Rule(outflow\_rate['very\_low'] & current\_level['very\_high'], control\_signal['maintain'])

rule4 = ctrl.Rule(inflow\_rate['very\_high'] & current\_level['low'], control\_signal['increase'])

rule5 = ctrl.Rule(outflow\_rate['very\_low'] & current\_level['very\_high'], control\_signal['decrease'])

rule6 = ctrl.Rule(inflow\_rate['moderate'] & outflow\_rate['moderate'], control\_signal['maintain'])

rule7 = ctrl.Rule(inflow\_rate['low'] & current\_level['very\_low'], control\_signal['increase'])

rule8 = ctrl.Rule(outflow\_rate['high'] & current\_level['high'], control\_signal['decrease'])

rule9 = ctrl.Rule(inflow\_rate['very\_low'] & outflow\_rate['very\_high'], control\_signal['maintain'])

# Create control system and simulation

control\_rules = [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9]

water\_level\_ctrl = ctrl.ControlSystem(control\_rules)

water\_level = ctrl.ControlSystemSimulation(water\_level\_ctrl)

# User input

inflow\_input = float(input("Enter inflow rate (0-10): "))

outflow\_input = float(input("Enter outflow rate (0-10): "))

current\_level\_input = float(input("Enter current water level (0-100): "))

# Setting the inputs

water\_level.input['inflow\_rate'] = inflow\_input

water\_level.input['outflow\_rate'] = outflow\_input

water\_level.input['current\_level'] = current\_level\_input

# Crunch the numbers

water\_level.compute()

print(f"Control Signal: {water\_level.output['control\_signal']}")

control\_signal.view(sim=water\_level)

plt.show() # This will display the plot inline within the Jupyter Notebook cell



