

Лабораторна робота №3

МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ НЕЧІТКИХ МНОЖИН ТА ФОРМУВАННЯ НЕЧІТКИХ ПРАВИЛ

Мета: дослідити можливості ПППІ MATLAB щодо проектування систем керування на основі алгоритмів нечіткого виводу

Хід роботи

Завдання 1: Побудова нечіткої моделі системи керування кранами.

Спроектувати систему нечіткого виводу (СНВ) для керування змішувачем, яка приймає на вхід температуру та напір води, а на виході видає кути повороту кранів.

- Вхідні: Temperature (0-100°C), Pressure (0-10 атм).
- Вихідні: Hot_Valve (-90°...90°), Cold_Valve (-90°...90°).

Для змінних використано трикутні функції належності (trimf).

```
=== ЗАВДАННЯ 1: Керування змішувачем ===  
Вхід: T=90°C, P=8 атм  
Вихід Гарячий кран: -60.00°  
Вихід Холодний кран: 60.00°  
Генерування 3D графіку: Task 1: Hot Valve Control Surface...
```

Рис.1.1 – Результат виконання завдання.

Під час виконання програми було сформовано функції належності для вхідних та вихідних змінних.

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.21.121.10.000 – Лр.1		
Змн.	Арк.	№ докум.	Підпис	Дата			
Розроб.		Ломоносов І.О.			Звіт з лабораторної роботи №1	Літ.	Арк.
Перевір.		Маєвський О.В.					1
Реценз.						ФІКТ, гр. ІПЗ-22-4	
Н. Контр.							
Зав.каф.		Єфіменко А.А.					

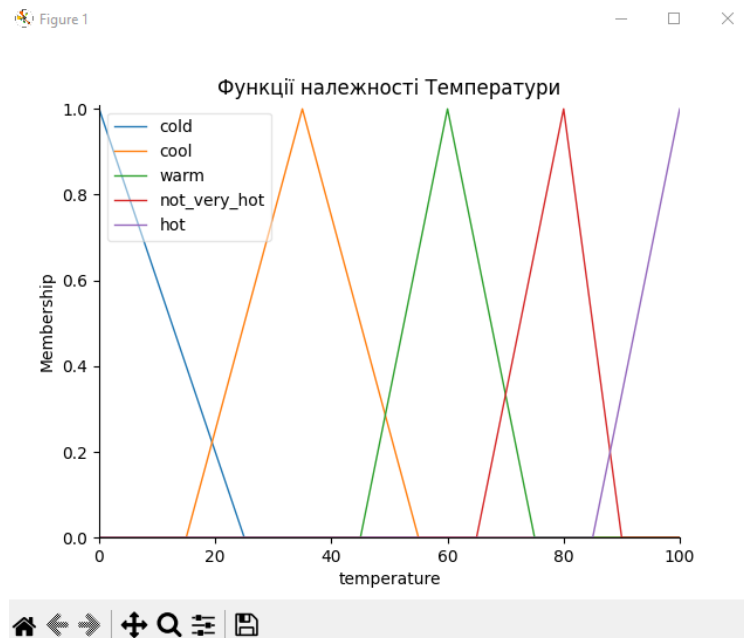


Рис.1.2 – Функції належності для температури.

На діаграмі зображено 5 термів (Cold, Cool, Warm, Not_Very_Hot, Hot). Використання трикутних функцій належності, що перекриваються, дозволяє системі плавно реагувати на зміну температури, уникаючи різких стрибків у керуванні.

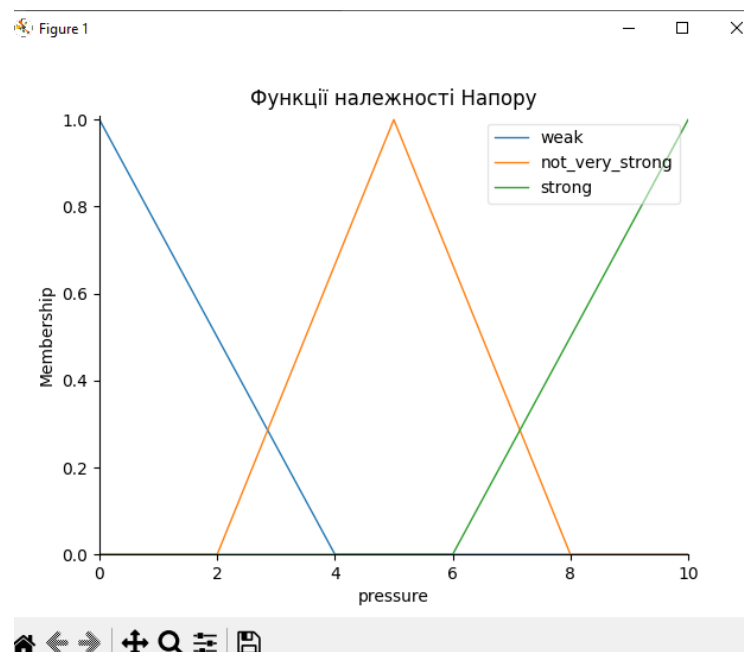


Рис.1.3 – Функції незалежності для напору.

Змінна "Напір" має 3 терми: Weak, Not_Very_Strong, Strong. Це дозволяє враховувати силу потоку води при прийнятті рішень щодо повороту кранів (наприклад, при сильному напорі кут повороту може коригуватися, щоб уникнути розбризкування або гідроударів).

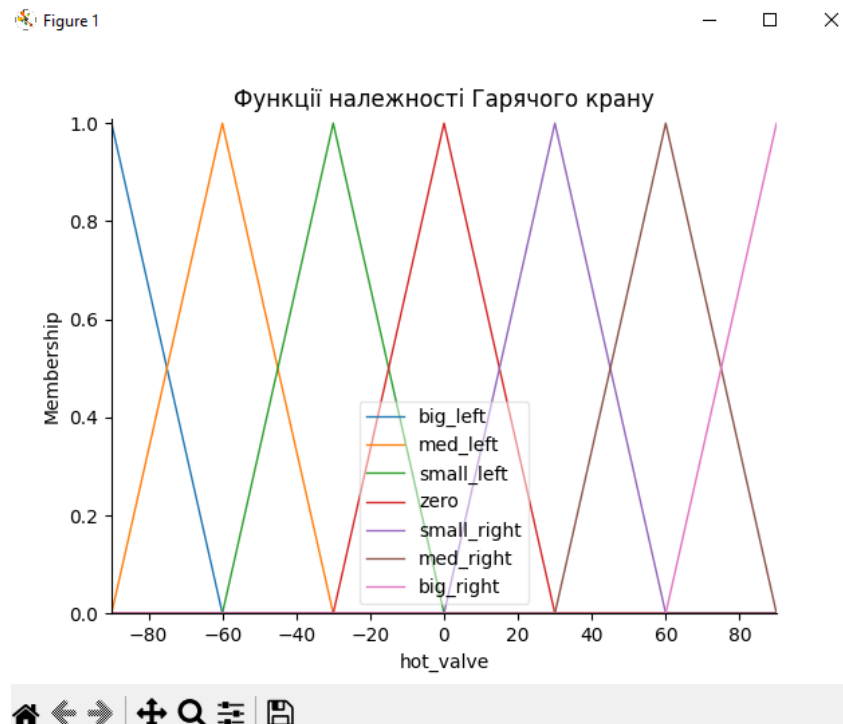


Рис.1.4 – Функції незалежності для гарячого крану.

Вона описує можливі напрямки й величини повороту крана гарячої води. Від’ємні значення відповідають руху в бік закриття (вліво), а додатні — у бік відкриття (вправо). Така організація дозволяє системі плавно й точно регулювати положення клапана відповідно до змін ситуації.

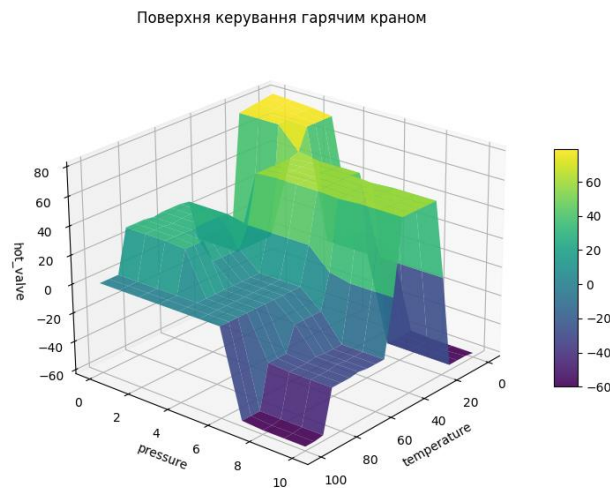


Рис.1.5 – Поверхня керування гарячим краном.

Отримана 3D-поверхня візуалізує базу правил. Ми бачимо, що:

- У зоні низьких температур (ліва частина) система формує сигнал на відкриття гарячого крану (додатні кути).
- У зоні високих температур (права частина) гарячий кран закривається.
- Напір впливає на "крутизну" поверхні, корегуючи інтенсивність реакції. Це підтверджує, що спроектована нечітка система поводить себе логічно та адекватно

Завдання 2: Нечітка модель керування кондиціонером повітря в приміщенні.

Розробити систему керування кондиціонером, яка враховує не тільки поточну температуру в приміщенні, але й швидкість її зміни (інерційність), щоб забезпечити більш точне та енергоефективне підтримання клімату.

- Вхідні: Room_Temp (Температура в кімнаті), Delta_Temp (Швидкість зміни).
- Вихідні: AC_Control (Режим роботи: Холод/Тепло/Вимк).

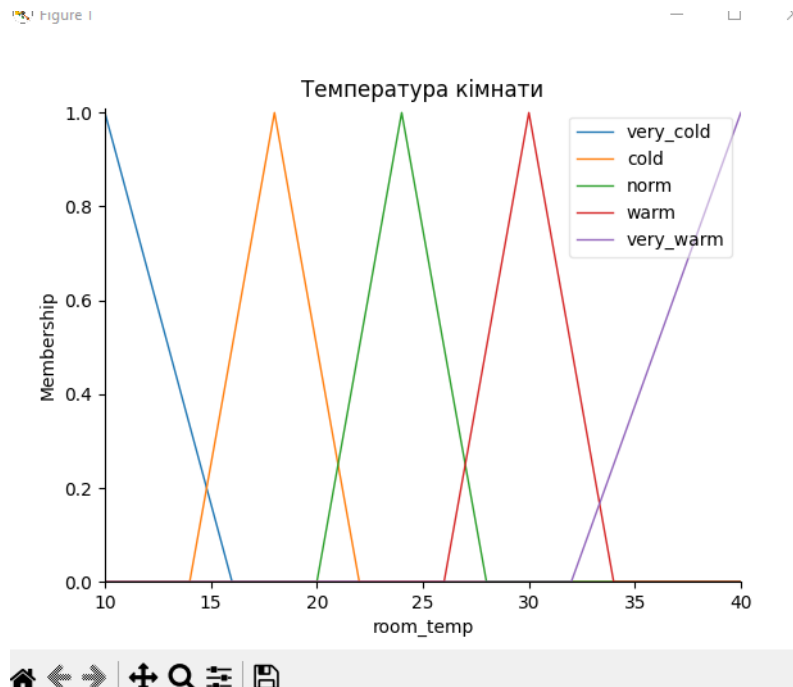


Рис.1.6 – Функції належності для температури в кімнаті.

На графіку функцій належності для температури при значенні наприклад 30 °С помітно, що точка має високу міру належності до терма warm і лише незначну — до сусідніх термів. Це свідчить про те, що система розпізнає такий стан як теплий.

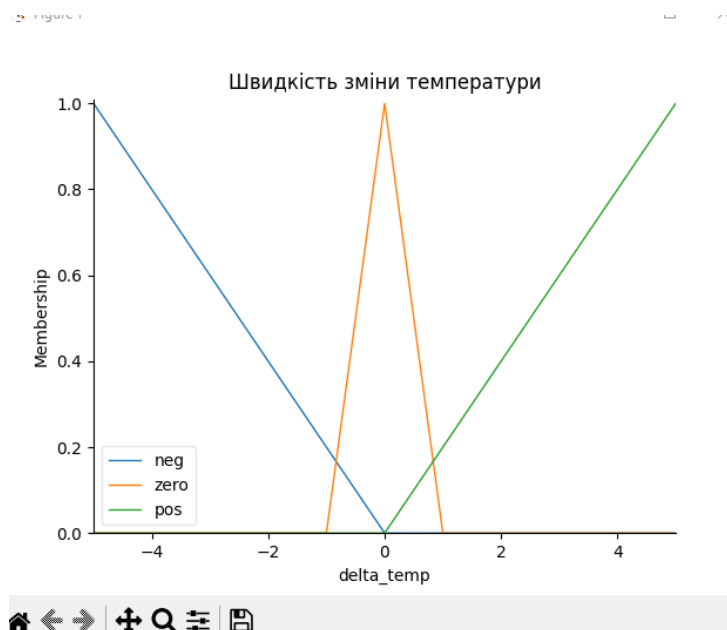


Рис.1.7 – Функції незалежності швидкості зміни температури.

Введення змінної "Швидкість зміни температури" (speed_change) є критично важливим. Вона має терми "Negative" (холодає), "Zero" (стабільно) та "Positive" (теплішає). Це дозволяє системі працювати на випередження: наприклад, якщо температура ще в нормі, але швидко росте ("Positive"), кондиціонер почне охолодження раніше.

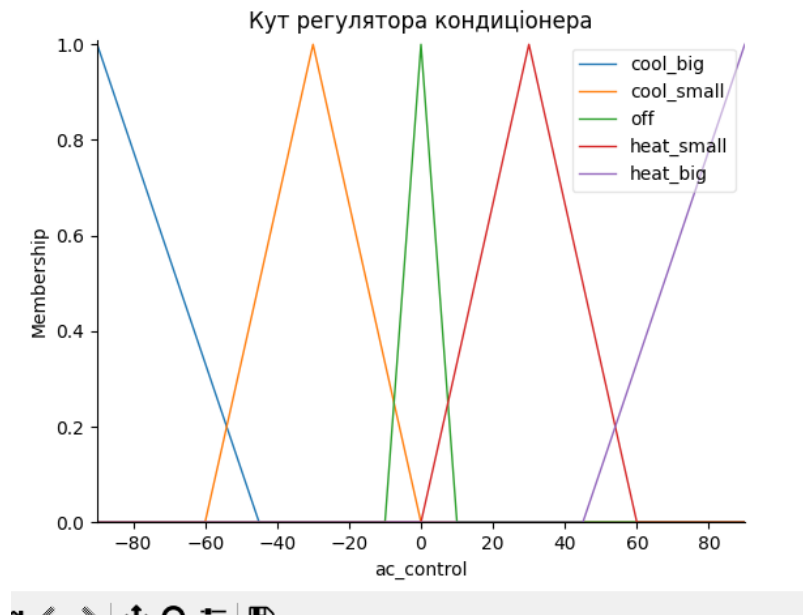


Рис.1.8 – Функції незалежності кута регулятора кондиціонера.

Вихідна змінна визначає режим роботи: охолодження (від'ємні кути) або обігрів (додатні кути). Стан "Turn Off" (0) відповідає ситуації, коли клімат у нормі.

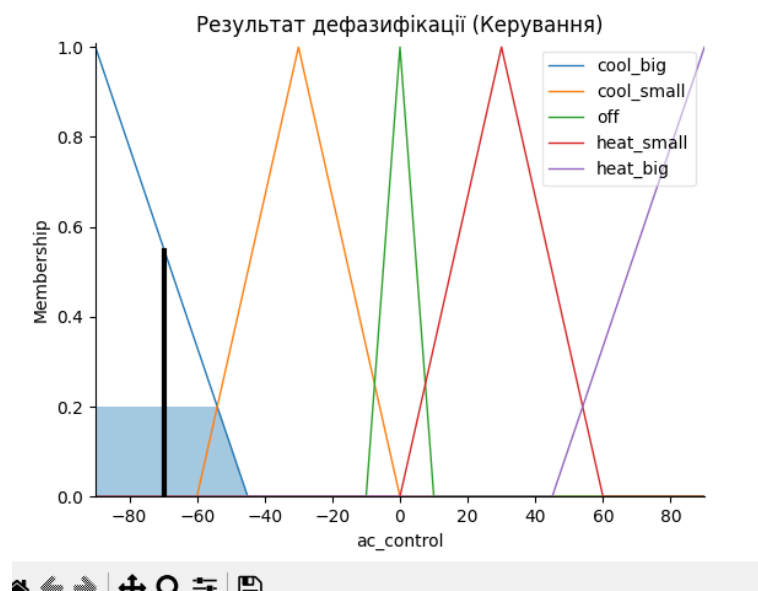


Рис.1.9 – Результат дефазифікації.

Цей графік ілюструє процес формування чіткого вихідного значення (кут повороту регулятора). Після агрегації активних правил ми отримуємо результуючу фігуру (зафарбована область). Центр тяжіння цієї фігури (вертикальна лінія) вказує на кінцеве значення (наприклад, -70 значний поворот вліво, що відповідає інтенсивному охолодженню large_left). Це підтверджує, що система адекватно компенсує перегрів приміщення.

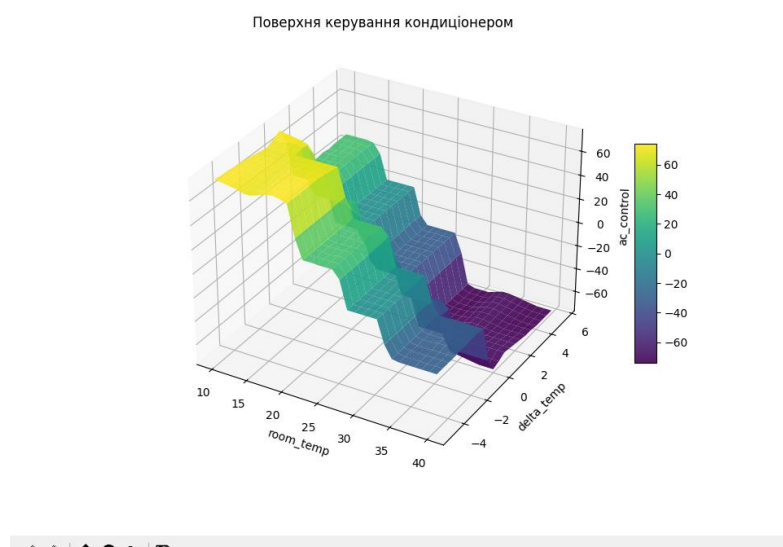


Рис.1.10 – Поверхня керування кондиціонером.

Інтерактивна 3D-поверхня «температура — швидкість зміни — кут регулятора» ілюструє загальну поведінку моделі:

- У області високих температур і додатної швидкості зміни (червона зона) вихід переходить у значні від'ємні значення (активне охолодження).
- За нормальної температури та нульової швидкості зміни система формує малі кути або переходить у стан turn_off (зона плато).
- Поверхня має шматково-гладкий характер, що підтверджує коректну роботу бази правил і плавні переходи між режимами охолодження та нагрівання.

Лістинг 1.1

```
import matplotlib

try:
    matplotlib.use('TkAgg')
except:
    pass

import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt

def run_task_1_water_mixer():
    print("\n=== ЗАВДАННЯ 1: Керування змішувачем ===")

    temp = ctrl.Antecedent(np.arange(0, 101, 1), 'temperature')
    pressure = ctrl.Antecedent(np.arange(0, 11, 1), 'pressure')
    hot_valve = ctrl.Consequent(np.arange(-90, 91, 1), 'hot_valve')
    cold_valve = ctrl.Consequent(np.arange(-90, 91, 1), 'cold_valve')

    temp['cold'] = fuzz.trimf(temp.universe, [0, 0, 25])
    temp['cool'] = fuzz.trimf(temp.universe, [15, 35, 55])
    temp['warm'] = fuzz.trimf(temp.universe, [45, 60, 75])
    temp['not_very_hot'] = fuzz.trimf(temp.universe, [65, 80, 90])
    temp['hot'] = fuzz.trimf(temp.universe, [85, 100, 100])

    pressure['weak'] = fuzz.trimf(pressure.universe, [0, 0, 4])
    pressure['not_very_strong'] = fuzz.trimf(pressure.universe, [2, 5, 8])
    pressure['strong'] = fuzz.trimf(pressure.universe, [6, 10, 10])

    angle_names = ['big_left', 'med_left', 'small_left', 'zero', 'small_right',
                  'med_right', 'big_right']
    hot_valve.automf(names=angle_names)
    cold_valve.automf(names=angle_names)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.20.121.10.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8


```

temp.view()
plt.title('Функції належності Температури')
plt.show()

pressure.view()
plt.title('Функції належності Напору')
plt.show()

hot_valve.view()
plt.title('Функції належності Гарячого крану')
plt.show()

rules = [
    ctrl.Rule(temp['hot'] & pressure['strong'], (hot_valve['med_left'],
cold_valve['med_right'])),
    ctrl.Rule(temp['hot'] & pressure['not_very_strong'],
cold_valve['med_right']),
    ctrl.Rule(temp['not_very_hot'] & pressure['strong'],
hot_valve['small_left']),
    ctrl.Rule(temp['not_very_hot'] & pressure['weak'],
(hot_valve['small_right'], cold_valve['small_right'])),
    ctrl.Rule(temp['warm'] & pressure['not_very_strong'], (hot_valve['zero'],
cold_valve['zero'])),
    ctrl.Rule(temp['cool'] & pressure['strong'], (hot_valve['med_right'],
cold_valve['med_left'])),
    ctrl.Rule(temp['cool'] & pressure['not_very_strong'],
(hot_valve['med_right'], cold_valve['small_left'])),
    ctrl.Rule(temp['cold'] & pressure['weak'], hot_valve['big_right']),
    ctrl.Rule(temp['cold'] & pressure['strong'], (hot_valve['med_left'],
cold_valve['med_right'])),
    ctrl.Rule(temp['warm'] & pressure['strong'], (hot_valve['small_left'],
cold_valve['small_left'])),
    ctrl.Rule(temp['warm'] & pressure['weak'], (hot_valve['small_right'],
cold_valve['small_right']))
]

water_ctrl = ctrl.ControlSystem(rules)
water_sim = ctrl.ControlSystemSimulation(water_ctrl)

water_sim.input['temperature'] = 90
water_sim.input['pressure'] = 8
water_sim.compute()
print(f"Кут гарячого крану: {water_sim.output.get('hot_valve', 0):.2f}")
print(f"Кут холодного крану: {water_sim.output.get('cold_valve', 0):.2f}")

plot_surface_3d(water_sim, np.arange(0, 101, 5), np.arange(0, 11, 1),
                'temperature', 'pressure', 'hot_valve',
                'Поверхня керування гарячим краном')

def run_task_2_air_conditioner():
    print("\n=== ЗАВДАННЯ 2: Керування кондиціонером ===")

    room_temp = ctrl.Antecedent(np.arange(10, 41, 1), 'room_temp')
    delta_temp = ctrl.Antecedent(np.arange(-5, 6, 1), 'delta_temp')
    ac_control = ctrl.Consequent(np.arange(-90, 91, 1), 'ac_control')

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.20.121.10.000 – Лр.1	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

room_temp['very_cold'] = fuzz.trimf(room_temp.universe, [10, 10, 16])
room_temp['cold'] = fuzz.trimf(room_temp.universe, [14, 18, 22])
room_temp['norm'] = fuzz.trimf(room_temp.universe, [20, 24, 28])
room_temp['warm'] = fuzz.trimf(room_temp.universe, [26, 30, 34])
room_temp['very_warm'] = fuzz.trimf(room_temp.universe, [32, 40, 40])

delta_temp['neg'] = fuzz.trimf(delta_temp.universe, [-5, -5, 0])
delta_temp['zero'] = fuzz.trimf(delta_temp.universe, [-1, 0, 1])
delta_temp['pos'] = fuzz.trimf(delta_temp.universe, [0, 5, 5])

ac_control['cool_big'] = fuzz.trimf(ac_control.universe, [-90, -90, -45])
ac_control['cool_small'] = fuzz.trimf(ac_control.universe, [-60, -30, 0])
ac_control['off'] = fuzz.trimf(ac_control.universe, [-10, 0, 10])
ac_control['heat_small'] = fuzz.trimf(ac_control.universe, [0, 30, 60])
ac_control['heat_big'] = fuzz.trimf(ac_control.universe, [45, 90, 90])

room_temp.view()
plt.title('Температура кімнати')
plt.show()

delta_temp.view()
plt.title('Швидкість зміни температури')
plt.show()

ac_control.view()
plt.title('Кут регулятора кондиціонера')
plt.show()

rules = [
    ctrl.Rule(room_temp['very_warm'] & delta_temp['pos'],
ac_control['cool_big']),
    ctrl.Rule(room_temp['very_warm'] & delta_temp['neg'],
ac_control['cool_small']),
    ctrl.Rule(room_temp['warm'] & delta_temp['pos'], ac_control['cool_big']),
    ctrl.Rule(room_temp['warm'] & delta_temp['neg'], ac_control['off']),
    ctrl.Rule(room_temp['very_cold'] & delta_temp['neg'],
ac_control['heat_big']),
    ctrl.Rule(room_temp['very_cold'] & delta_temp['pos'],
ac_control['heat_small']),
    ctrl.Rule(room_temp['cold'] & delta_temp['neg'], ac_control['heat_big']),
    ctrl.Rule(room_temp['cold'] & delta_temp['pos'], ac_control['off']),
    ctrl.Rule(room_temp['very_warm'] & delta_temp['zero'],
ac_control['cool_big']),
    ctrl.Rule(room_temp['warm'] & delta_temp['zero'],
ac_control['cool_small']),
    ctrl.Rule(room_temp['very_cold'] & delta_temp['zero'],
ac_control['heat_big']),
    ctrl.Rule(room_temp['cold'] & delta_temp['zero'],
ac_control['heat_small']),
    ctrl.Rule(room_temp['norm'] & delta_temp['pos'],
ac_control['cool_small']),
    ctrl.Rule(room_temp['norm'] & delta_temp['neg'],
ac_control['heat_small']),
    ctrl.Rule(room_temp['norm'] & delta_temp['zero'], ac_control['off'])
]

ac_system = ctrl.ControlSystem(rules)
ac_sim = ctrl.ControlSystemSimulation(ac_system)

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.20.121.10.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

```

ac_sim.input['room_temp'] = 38
ac_sim.input['delta_temp'] = 1
ac_sim.compute()

ac_control.view(sim=ac_sim)
plt.title('Результат дефазифікації (Керування)')
plt.show()

plot_surface_3d(ac_sim, np.arange(10, 41, 1), np.arange(-5, 6, 0.5),
                'room_temp', 'delta_temp', 'ac_control',
                'Поверхня керування кондиціонером')

def plot_surface_3d(simulation, x_range, y_range, input_x_name, input_y_name,
                    output_name, title):
    x, y = np.meshgrid(x_range, y_range)
    z = np.zeros_like(x)

    print(f"Генерування 3D графіку: {title}...")

    for i in range(y_range.size):
        for j in range(x_range.size):
            simulation.input[input_x_name] = x[i, j]
            simulation.input[input_y_name] = y[i, j]
            try:
                simulation.compute()
                z[i, j] = simulation.output[output_name]
            except:
                z[i, j] = 0

    fig = plt.figure(figsize=(10, 7))
    ax = fig.add_subplot(111, projection='3d')
    surf = ax.plot_surface(x, y, z, cmap='viridis', edgecolor='none', alpha=0.9)

    ax.set_xlabel(input_x_name)
    ax.set_ylabel(input_y_name)
    ax.set_zlabel(output_name)
    ax.set_title(title)
    fig.colorbar(surf, shrink=0.5, aspect=10)
    plt.show()

if __name__ == "__main__":
    run_task_1_water_mixer()
    run_task_2_air_conditioner()

```

Висновок: Ми дослідили можливості ППП MATLAB щодо проектування систем керування на основі алгоритмів нечіткого виводу.

Посилання на git: <https://github.com/IgorLomonosov/SAI>

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.20.121.10.000 – Лр.1	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11