

Лабораторна робота №3

МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ НЕЧІТКИХ МНОЖИН ТА ФОРМУВАННЯ НЕЧІТКИХ ПРАВИЛ

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

Хід роботи

Репозиторій GITHUB: https://github.com/AnatoliiYarmolenko/SMI_1S4C

Завдання 3.1. Побудова нечіткої моделі системи керування кранами гарячої і холодної води

Для виконання завдання зкомпілюємо наступний код. Цей скрипт моделює систему нечіткої логіки для керування змішувачем води. Спочатку він визначає дві вхідні лінгвістичні змінні Температура та Напір, та дві вихідні змінні Кран гарячої і Кран холодної. Кожній змінній призначаються відповідні терми ("гаряча", "сильний", "повернути вліво"...) та їхні функції належності, які описують ступінь приналежності до цих термів. Потім скрипт створює базу знань, впроваджуючи евристичні правила із завдання. Наприкінці він об'єднує все це в єдину систему керування та будує 3D-графіки, які візуалізують, яке рішення прийме система для будь-якої комбінації вхідних даних:

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# --- 1. Визначення Всесвітів (Діапазонів) ---

# Вхідні змінні
# Припустимо, температура від 0 до 100 градусів
universe_temp = np.arange(0, 101, 1)
# Припустимо, напір - це умовна шкала від 0 до 10
universe_pressure = np.arange(0, 11, 1)
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.31.000 – Лр3				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Ярмоленко А.М.			Звіт з лабораторної роботи №3	Лім.	Арк.	Аркушів	
Перевір.		Маєвський О.В.					1	10	
Реценз.						ФІКТ, гр. ІПЗ-22-4			
Н. Контр.									
Зав.каф.		Вакалюк Т.А.							

```

# Вихідні змінні
# Кут повороту від -90 (макс. вліво) до +90 (макс. вправо)
universe_angle = np.arange(-90, 91, 1)

# --- 2. Створення Вхідних та Вихідних Змінних ---
temperature = ctrl.Antecedent(universe_temp, 'temperature')
pressure = ctrl.Antecedent(universe_pressure, 'pressure')

hot_tap = ctrl.Consequent(universe_angle, 'hot_tap')
cold_tap = ctrl.Consequent(universe_angle, 'cold_tap')

# --- 3. Визначення Функцій Належності (Membership Functions) ---

# Temperature Terms: 'cold', 'cool', 'warm', 'not_very_hot', 'hot'
temperature['cold'] = fuzz.trapmf(temperature.universe, [0, 0, 10, 25])
temperature['cool'] = fuzz.trimf(temperature.universe, [15, 30, 45])
temperature['warm'] = fuzz.trimf(temperature.universe, [40, 50, 60])
temperature['not_very_hot'] = fuzz.trimf(temperature.universe, [55, 70, 85])
temperature['hot'] = fuzz.trapmf(temperature.universe, [75, 90, 100, 100])

# Pressure Terms: 'weak', 'not_very_strong', 'strong'
pressure['weak'] = fuzz.trapmf(pressure.universe, [0, 0, 2, 4])
pressure['not_very_strong'] = fuzz.trimf(pressure.universe, [3, 5, 7])
pressure['strong'] = fuzz.trapmf(pressure.universe, [6, 8, 10, 10])

# Angle Terms: 'Large Left' (LL), 'Medium Left' (ML), 'Small Left' (SL),
# 'No Change' (NC), 'Small Right' (SR), 'Medium Right' (MR), 'Large Right' (LR)
# "вліво" = negative, "вправо" = positive [cite: 256, 257]
hot_tap['LL'] = fuzz.trimf(hot_tap.universe, [-90, -90, -60])
hot_tap['ML'] = fuzz.trimf(hot_tap.universe, [-75, -45, -15])
hot_tap['SL'] = fuzz.trimf(hot_tap.universe, [-30, -15, 0])
hot_tap['NC'] = fuzz.trimf(hot_tap.universe, [-10, 0, 10])
hot_tap['SR'] = fuzz.trimf(hot_tap.universe, [0, 15, 30])
hot_tap['MR'] = fuzz.trimf(hot_tap.universe, [15, 45, 75])
hot_tap['LR'] = fuzz.trimf(hot_tap.universe, [60, 90, 90])

# Використовуємо ті ж терміни для крана холодної води
cold_tap['LL'] = fuzz.trimf(cold_tap.universe, [-90, -90, -60])
cold_tap['ML'] = fuzz.trimf(cold_tap.universe, [-75, -45, -15])
cold_tap['SL'] = fuzz.trimf(cold_tap.universe, [-30, -15, 0])
cold_tap['NC'] = fuzz.trimf(cold_tap.universe, [-10, 0, 10])
cold_tap['SR'] = fuzz.trimf(cold_tap.universe, [0, 15, 30])
cold_tap['MR'] = fuzz.trimf(cold_tap.universe, [15, 45, 75])
cold_tap['LR'] = fuzz.trimf(cold_tap.universe, [60, 90, 90])

# Перегляд функцій належності
# temperature.view()
# pressure.view()
# hot_tap.view()

# --- 4. Визначення Експертних Правил ---

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

```

# (ML: середній вліво, MR: середній вправо, SL: невеликий вліво, SR: невеликий
вправо, LR: великий вправо, NC: не змінювати)

rule1 = ctrl.Rule(temperature['hot'] & pressure['strong'],
                  (hot_tap['ML'], cold_tap['MR'])) #
rule2 = ctrl.Rule(temperature['hot'] & pressure['not_very_strong'],
                  (hot_tap['NC'], cold_tap['MR'])) # [cite: 259]
rule3 = ctrl.Rule(temperature['not_very_hot'] & pressure['strong'],
                  (hot_tap['SL'], cold_tap['NC'])) # [cite: 260]
rule4 = ctrl.Rule(temperature['not_very_hot'] & pressure['weak'],
                  (hot_tap['SR'], cold_tap['SR'])) # [cite: 261]
rule5 = ctrl.Rule(temperature['warm'] & pressure['not_very_strong'],
                  (hot_tap['NC'], cold_tap['NC'])) #
rule6 = ctrl.Rule(temperature['cool'] & pressure['strong'],
                  (hot_tap['MR'], cold_tap['ML'])) # [cite: 263]
rule7 = ctrl.Rule(temperature['cool'] & pressure['not_very_strong'],
                  (hot_tap['MR'], cold_tap['SL'])) # [cite: 264]
rule8 = ctrl.Rule(temperature['cold'] & pressure['weak'],
                  (hot_tap['LR'], cold_tap['NC'])) #
rule9 = ctrl.Rule(temperature['cold'] & pressure['strong'],
                  (hot_tap['ML'], cold_tap['MR'])) # [cite: 266]
rule10 = ctrl.Rule(temperature['warm'] & pressure['strong'],
                  (hot_tap['SL'], cold_tap['SL'])) # [cite: 267]
rule11 = ctrl.Rule(temperature['warm'] & pressure['weak'],
                  (hot_tap['SR'], cold_tap['SR'])) # [cite: 268]

# --- 5. Створення Системи Керування ---
tap_control_system = ctrl.ControlSystem(
    [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10, rule11]
)
tap_simulation = ctrl.ControlSystemSimulation(tap_control_system)

# --- 6. Запуск Симуляції (Приклад) ---

# Приклад 1: Вода дуже гаряча (95) і напір сильний (9)
tap_simulation.input['temperature'] = 95
tap_simulation.input['pressure'] = 9
tap_simulation.compute()

print("--- Приклад 1: Гаряча вода (95), Сильний напір (9) ---")
print(f"Поворот гарячого крана: {tap_simulation.output['hot_tap']:.2f} градусів")
print(f"Поворот холодного крана: {tap_simulation.output['cold_tap']:.2f} градусів")
# Очікуваний результат: гарячий кран - вліво (негативне), холодний - вправо
(позитивне)

# Приклад 2: Вода холодна (10) і напір слабкий (2)
tap_simulation.input['temperature'] = 10
tap_simulation.input['pressure'] = 2
tap_simulation.compute()

print("\n--- Приклад 2: Холодна вода (10), Слабкий напір (2) ---")

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		3

```

print(f"Поворот гарячого крана: {tap_simulation.output['hot_tap']:.2f} градусів")
print(f"Поворот холодного крана: {tap_simulation.output['cold_tap']:.2f} градусів")
# Очікуваний результат: гарячий кран - сильно вправо (позитивне)

# --- 7. Побудова 3D Поверхонь Керування (аналог Рис. 7 ) ---

# Готуємо 2D сітку для вхідних даних
temp_range = np.linspace(universe_temp.min(), universe_temp.max(), 30)
pres_range = np.linspace(universe_pressure.min(), universe_pressure.max(), 30)
temp_grid, pres_grid = np.meshgrid(temp_range, pres_range)

# Готуємо 2D сітку для вихідних даних
hot_tap_output = np.zeros_like(temp_grid)
cold_tap_output = np.zeros_like(temp_grid)

# Розрахунок виходів для кожної точки сітки
for i in range(temp_grid.shape[0]):
    for j in range(temp_grid.shape[1]):
        tap_simulation.input['temperature'] = temp_grid[i, j]
        tap_simulation.input['pressure'] = pres_grid[i, j]

        try:
            tap_simulation.compute()
            hot_tap_output[i, j] = tap_simulation.output['hot_tap']
            cold_tap_output[i, j] = tap_simulation.output['cold_tap']
        except:
            # Обробка випадків, де правила не спрацьовують
            hot_tap_output[i, j] = 0
            cold_tap_output[i, j] = 0

# Побудова 3D-графіка для гарячого крана
fig1 = plt.figure(figsize=(10, 7))
ax1 = fig1.add_subplot(111, projection='3d')
surf1 = ax1.plot_surface(temp_grid, pres_grid, hot_tap_output, cmap='viridis')
ax1.set_xlabel('Temperature (°C)')
ax1.set_ylabel('Pressure (0-10)')
ax1.set_zlabel('Hot Tap Angle (degrees)')
ax1.set_title('Hot Tap Control Surface')
fig1.colorbar(surf1, shrink=0.5, aspect=5)

# Побудова 3D-графіка для холодного крана
fig2 = plt.figure(figsize=(10, 7))
ax2 = fig2.add_subplot(111, projection='3d')
surf2 = ax2.plot_surface(temp_grid, pres_grid, cold_tap_output, cmap='plasma')
ax2.set_xlabel('Temperature (°C)')
ax2.set_ylabel('Pressure (0-10)')
ax2.set_zlabel('Cold Tap Angle (degrees)')
ax2.set_title('Cold Tap Control Surface')
fig2.colorbar(surf2, shrink=0.5, aspect=5)

plt.show()

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

В результаті отримаємо наступні графіки та вивід в консоль тестових даних:

```

d:/02 Polytexn/SMI_1S4C/Lab_3/LK_3_Task_1.py
--- Приклад 1: Гаряча вода (95), Сильний напір (9) ---
Поворот гарячого крана: -45.00 градусів
Поворот холодного крана: 45.00 градусів

--- Приклад 2: Холодна вода (10), Слабкий напір (2) ---
Поворот гарячого крана: 80.00 градусів
Поворот холодного крана: 0.00 градусів

```

Рис. 1 Тестові дані

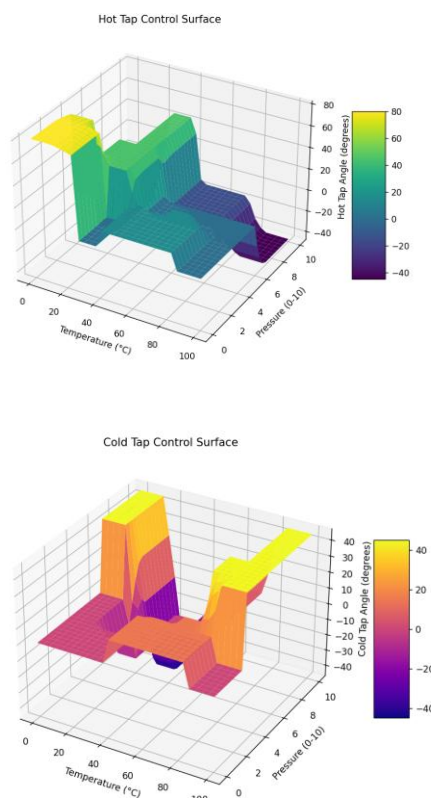


Рис. 2 Графічний вивід

Поверхні керування це 3D-візуалізація всіх можливих рішень, які прийме нечітка система. Горизонтальні осі представляють діапазони вхідних змінних — Температури та Напору, а вертикальна вісь показує кінцеве, чітке рішення для вихідної змінної — Кут повороту крана. На графіку Гарячий кран ми бачимо, що система повертає кран вправо (додатні значення Z, жовтий колір) при низькій температурі та вліво (від'ємні Z, фіолетовий колір) при високій. Графік Холодний кран демонструє зворотну логіку: він повертається вправо (додатні Z, жовтий) при

високій температурі, щоб остудити воду, і вліво (від'ємні Z, синій) при прохолодній, виконуючи правила, закладені експертом.

Завдання 3.2. Нечітка модель керування кондиціонером повітря в приміщенні

Аналогічно представимо код. На відміну від попередньої задачі про змішувач, ця система використовує інші вхідні лінгвістичні змінні: поточну Температуру в приміщенні та Швидкість її зміни. Ця друга змінна додана для врахування інертності системи, дозволяючи контролеру реагувати не лише на поточний стан, але й на динаміку процесу. Вихідна змінна також одна – Дія регулятора, яка визначає потужність нагріву або охолодження.

```
import numpy as np
import skfuzzy as fuzz
from skfuzzy import control as ctrl
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# --- 1. Визначення Всесвітів (Діапазонів) ---

# Вхідні змінні
# Припустимо, температура - це відхилення від "комфортної" (0)
# від -10 (дуже холодно) до +10 (дуже тепло)
universe_temp = np.arange(-10, 11, 1)
# Швидкість зміни: від -5 (швидко холоднішає) до +5 (швидко теплішає)
universe_speed = np.arange(-5, 6, 0.5)

# Вихідна змінна
# Дія регулятора: від -100 (макс. холод) до +100 (макс. тепло)
universe_action = np.arange(-100, 101, 10)

# --- 2. Створення Вхідних та Вихідних Змінних ---
temperature = ctrl.Antecedent(universe_temp, 'temperature')
speed = ctrl.Antecedent(universe_speed, 'speed')
action = ctrl.Consequent(universe_action, 'action')

# --- 3. Визначення Функцій Належності (Terms) ---

# Temperature Terms: 5 термінів
temperature['very_cold'] = fuzz.trapmf(temperature.universe, [-10, -10, -8, -6])
temperature['cold'] = fuzz.trimf(temperature.universe, [-7, -5, -3])
temperature['normal'] = fuzz.trimf(temperature.universe, [-4, 0, 4])
temperature['hot'] = fuzz.trimf(temperature.universe, [3, 5, 7])
temperature['very_hot'] = fuzz.trapmf(temperature.universe, [6, 8, 10, 10])

# Speed Terms: 3 терміни
```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

```

speed['negative'] = fuzz.trapmf(speed.universe, [-5, -5, -2, 0])
speed['zero'] = fuzz.trimf(speed.universe, [-1, 0, 1])
speed['positive'] = fuzz.trapmf(speed.universe, [0, 2, 5, 5])

# Action Terms: 5 термінів
# Вліво = 'холод' (негативні значення), Вправо = 'тепло' (позитивні)
action['LL'] = fuzz.trapmf(action.universe, [-100, -100, -80, -60]) # Великий вліво
action['SL'] = fuzz.trimf(action.universe, [-70, -50, -30]) # Невеликий вліво
action['Off'] = fuzz.trimf(action.universe, [-20, 0, 20]) # Вимкнуті
action['SR'] = fuzz.trimf(action.universe, [30, 50, 70]) # Невеликий вправо
action['LR'] = fuzz.trapmf(action.universe, [60, 80, 100, 100]) # Великий вправо

# Перегляд функцій належності
# temperature.view()
# speed.view()
# action.view()

# --- 4. Визначення Експертних Правил ---
# (LL: великий вліво, SL: невеликий вліво, Off: вимкнуті, SR: невеликий вправо, LR:
великий вправо)

rule1 = ctrl.Rule(temperature['very_hot'] & speed['positive'], action['LL'])
rule2 = ctrl.Rule(temperature['very_hot'] & speed['negative'], action['SL'])
rule3 = ctrl.Rule(temperature['hot'] & speed['positive'], action['LL'])
rule4 = ctrl.Rule(temperature['hot'] & speed['negative'], action['Off'])
rule5 = ctrl.Rule(temperature['very_cold'] & speed['negative'], action['LR'])
rule6 = ctrl.Rule(temperature['very_cold'] & speed['positive'], action['SR'])
rule7 = ctrl.Rule(temperature['cold'] & speed['negative'], action['LL']) #див. аналіз
rule8 = ctrl.Rule(temperature['cold'] & speed['positive'], action['Off'])
rule9 = ctrl.Rule(temperature['very_hot'] & speed['zero'], action['LL'])
rule10 = ctrl.Rule(temperature['hot'] & speed['zero'], action['SL'])
rule11 = ctrl.Rule(temperature['very_cold'] & speed['zero'], action['LR'])
rule12 = ctrl.Rule(temperature['cold'] & speed['zero'], action['SR'])
rule13 = ctrl.Rule(temperature['normal'] & speed['positive'], action['SL'])
rule14 = ctrl.Rule(temperature['normal'] & speed['negative'], action['SR'])
rule15 = ctrl.Rule(temperature['normal'] & speed['zero'], action['Off'])

# --- 5. Створення Системи Керування ---
ac_control_system = ctrl.ControlSystem(
    [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9,
     rule10, rule11, rule12, rule13, rule14, rule15]
)
ac_simulation = ctrl.ControlSystemSimulation(ac_control_system)

# --- 6. Запуск Симуляції (Приклади) ---

# Приклад 1: "Дуже тепло" (+8) і "стає тепліше" (+3)
ac_simulation.input['temperature'] = 8
ac_simulation.input['speed'] = 3
ac_simulation.compute()
print("--- Приклад 1: Дуже тепло (+8), стає тепліше (+3) ---")

```

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7


```

print(f"Дія регулятора: {ac_simulation.output['action']:.2f} (Очікується: 'Великий вліво')")

# Приклад 2: "Нормальна" (0) і "стає холодніше" (-2)
ac_simulation.input['temperature'] = 0
ac_simulation.input['speed'] = -2
ac_simulation.compute()
print("\n--- Приклад 2: Нормальна (0), стає холодніше (-2) ---")
print(f"Дія регулятора: {ac_simulation.output['action']:.2f} (Очікується: 'Невеликий вправо')")

# Приклад 3: "Дуже холодно" (-9) і "стає холодніше" (-4)
ac_simulation.input['temperature'] = -9
ac_simulation.input['speed'] = -4
ac_simulation.compute()
print("\n--- Приклад 3: Дуже холодно (-9), стає холодніше (-4) ---")
print(f"Дія регулятора: {ac_simulation.output['action']:.2f} (Очікується: 'Великий вправо')")

# --- 7. Побудова 3D Поверхні Керування ---
temp_range = np.linspace(universe_temp.min(), universe_temp.max(), 30)
speed_range = np.linspace(universe_speed.min(), universe_speed.max(), 30)
temp_grid, speed_grid = np.meshgrid(temp_range, speed_range)

action_output = np.zeros_like(temp_grid)

for i in range(temp_grid.shape[0]):
    for j in range(speed_grid.shape[1]):
        ac_simulation.input['temperature'] = temp_grid[i, j]
        ac_simulation.input['speed'] = speed_grid[i, j]

        try:
            ac_simulation.compute()
            action_output[i, j] = ac_simulation.output['action']
        except:
            action_output[i, j] = 0

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(temp_grid, speed_grid, action_output, cmap='coolwarm')

ax.set_xlabel('Temperature (Відхилення від норми)')
ax.set_ylabel('Speed (Швидкість зміни)')
ax.set_zlabel('Action (Дія регулятора)')
ax.set_title('Air Conditioner Control Surface')
fig.colorbar(surf, shrink=0.5, aspect=5)

plt.show()

```

В результаті отримаємо наступні графіки та вивід в консоль тестових даних:

					ЖИТОМИРСЬКА ПОЛІТЕХНІКА.25.121.2.000 – Лр.3	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		


```

--- Приклад 1: Дуже тепло (+8), стає тепліше (+3) ---
Дія регулятора: -84.44 (Очікується: 'Великий вліво')

--- Приклад 2: Нормальна (0), стає холодніше (-2) ---
Дія регулятора: 50.00 (Очікується: 'Невеликий вправо')

--- Приклад 3: Дуже холодно (-9), стає холодніше (-4) ---
Дія регулятора: 84.44 (Очікується: 'Великий вправо')

```

Рис. 3 Тестові дані

Інтерпретація тестових даних показує логіку роботи системи. Так, на першому прикладі, коли в кімнаті дуже тепло (+8) і температура продовжує зростати (+3), система реагує максимальним охолодженням ("Великий вліво" -83.58).

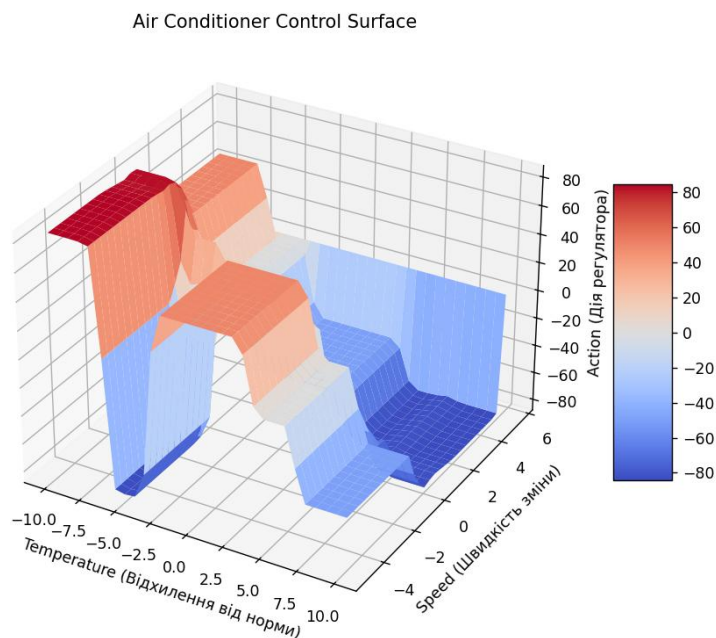


Рис. 4 Графічний вивід

Отримана поверхня керування є 3D-картою рішень кондиціонера. Горизонтальні осі представляють входи Температура та Швидкість зміни, а вертикальна вісь вихідну Дію регулятора. Кольори візуалізують цю дію: червоні "гори" означають сильний нагрів (позитивні значення "вправо"), а сині "ями" – сильне охолодження (негативні значення "вліво"), тоді як білі зони біля нуля відповідають вимкненому стану.

Варто зауважити невелику аномалію в зоні "холодно" та "від'ємна швидкість", де з'являється синя "яма" охолодження замість очікуваного нагріву. Це виникає через моделювання правила 7 з завдання:

«Якщо температура повітря холодна, а швидкість зміни температури від'ємна, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонера на великий кут вліво.»

Можливо в завданні наявна помилка і замість повороту «вліво» повинен бути поворот «вправо», що повинно виправити ситуацію:

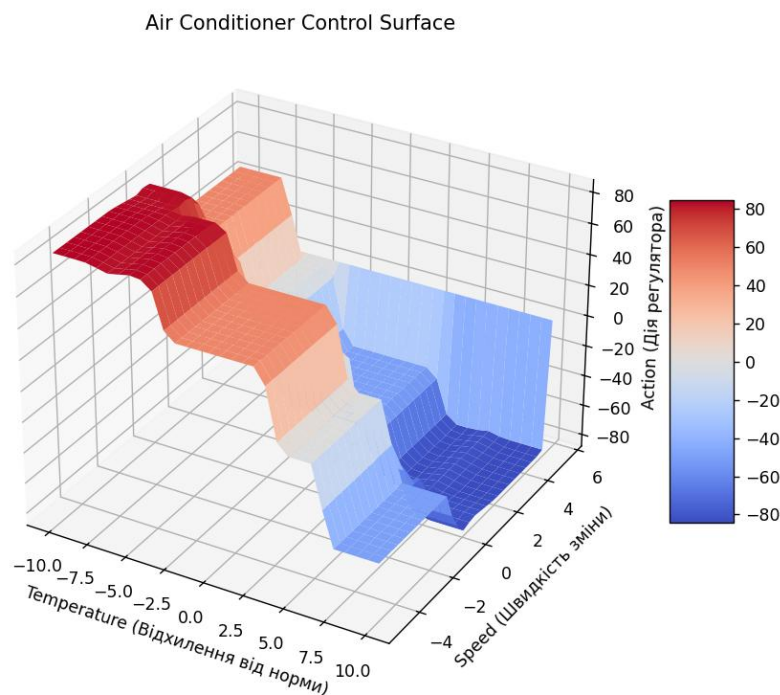


Рис. 5 Графічний вивід після зміни правила

Загалом, модель адекватно відтворює поведінку "розумного" контролера, який враховує не лише поточний стан, але й тенденцію його зміни, що є ключовою перевагою нечіткої логіки для керування інертними процесами.

Висновок: Під час цієї лабораторної роботи ми успішно змоделювали дві системи нечіткого керування: змішувач води та кондиціонер. Ключовим моментом стало використання "швидкості зміни" температури для кондиціонера, що дозволило врахувати інертність системи. Побудовані 3D-поверхні керування наочно візуалізували логіку рішень і навіть допомогли виявити потенційну помилку в одному з експертних правил.