

Лабораторна робота 3

МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ НЕЧІТКИХ МНОЖИН ТА ФОРМУ- ВАННЯ НЕЧІТКИХ ПРАВИЛ

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

ЗАВДАННЯ НА ЛАБОРАТОРНУ РОБОТУ

Завдання № 1.

Задача 1. Побудова нечіткої моделі системи керування кранами гарячої і холодної води

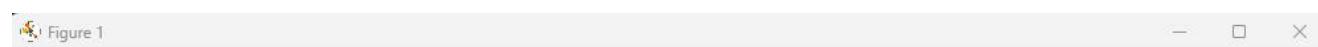


Рис. 1.1 Поверхня управління гарячим краном

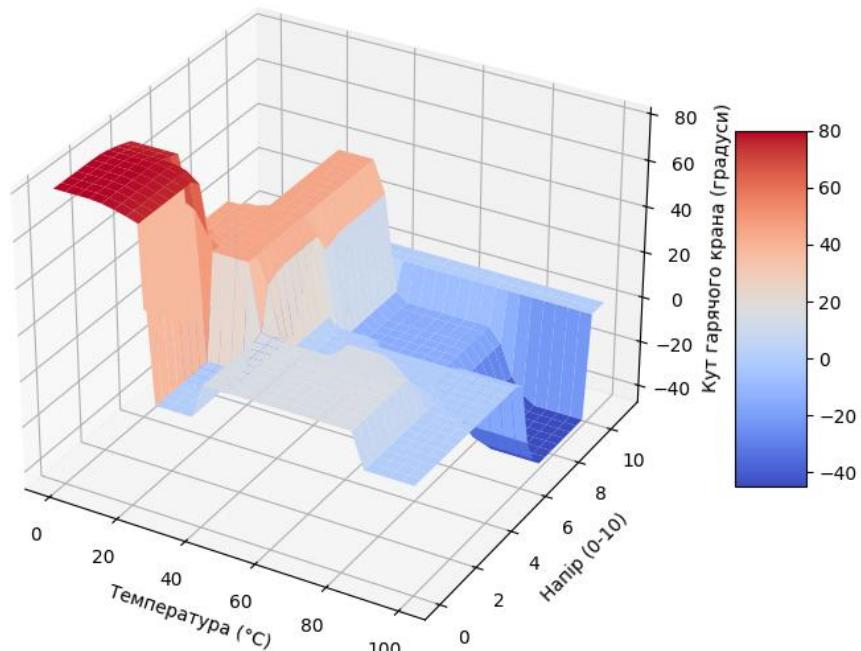


Рис. 1.1 Поверхня управління гарячим краном

Змн.	Арк.	№ докум.	Підпис	Дата	ДУ «Житомирська політехніка».25.121.14.003 – Пр3		
Розроб.	Кольцова Н.О.						
Перевір.	Маєвський О.В.						
Керівник							
Н. контр.							
Зав. каф.							
Звіт з лабораторної роботи					Lім.	Арк.	Аркушів
						1	13
					ФІКТ Гр. ІПЗ-22-4[1]		

Поверхня управління холодним краном

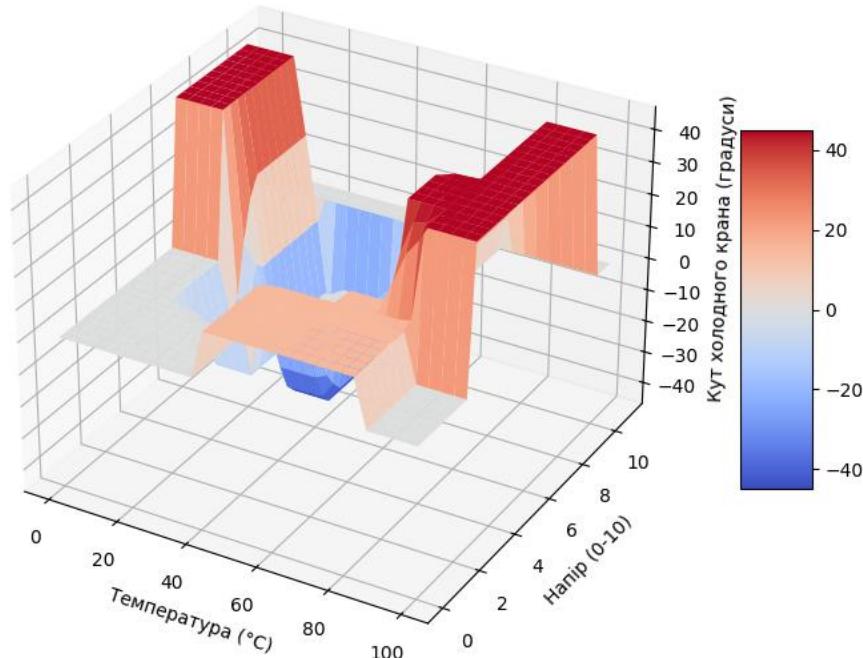


Рис. 1.2 Поверхня управління холодним краном

--- Приклад 1: Температура 90°C, Напір 9 ---

Поворот гарячого крана: -45.00 градусів

Поворот холодного крана: 45.00 градусів

--- Приклад 2: Температура 20°C, Напір 4 ---

Поворот гарячого крана: 45.00 градусів

Поворот холодного крана: -15.00 градусів

--- Приклад 3: Температура 55°C, Напір 7 ---

Поворот гарячого крана: -15.00 градусів

Поворот холодного крана: -15.00 градусів

--- Приклад 4: Температура 10°C, Напір 2 ---

Поворот гарячого крана: 80.00 градусів

Поворот холодного крана: 0.00 градусів

--- Приклад 5: Температура 45°C, Напір 8 ---

Поворот гарячого крана: -15.00 градусів

Поворот холодного крана: -15.00 градусів

--- Приклад 6: Температура 80°C, Напір 3 ---

Поворот гарячого крана: 15.00 градусів

Поворот холодного крана: 15.00 градусів

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

Рис. 1.3 Результати як система керує кранами залежно від температури води та її напору

Модель нечіткої логіки для керування кранами гарячої та холодної води працює коректно.

Графіки на рисунках 1.1 і 1.2 показують, як система адаптує кути відкриття кожного з кранів залежно від температури води і напору. На рисунку 1.1 (поверхня управління гарячим краном) видно, що при високій температурі та сильному напорі кран гарячої води відкривається на більший кут, а при низьких значеннях температури або слабкому напорі кран гарячої води залишається закритим або в нейтральному положенні. Analogічно, на рисунку 1.2 (поверхня управління холodним краном) показано, як кран холодної води відкривається в залежності від температури та напору.

Результати симуляції Рис. 1.3 підтверджують, що система коректно реагує на різні варіанти температури та напору, регулюючи кути відкриття кранів відповідно до заданих правил. Наприклад, для гарячої води з сильним напором система відкриває кран гарячої води на -45 градусів, а холодний кран відкривається на 45 градусів, що відповідає реальному процесу змішування води.

Нечітка модель забезпечує ефективне керування кранами і може бути використана для автоматизації систем водопостачання в реальних умовах.

Завдання № 2

Задача 2. Нечітка модель керування кондиціонером повітря в приміщенні.

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		3

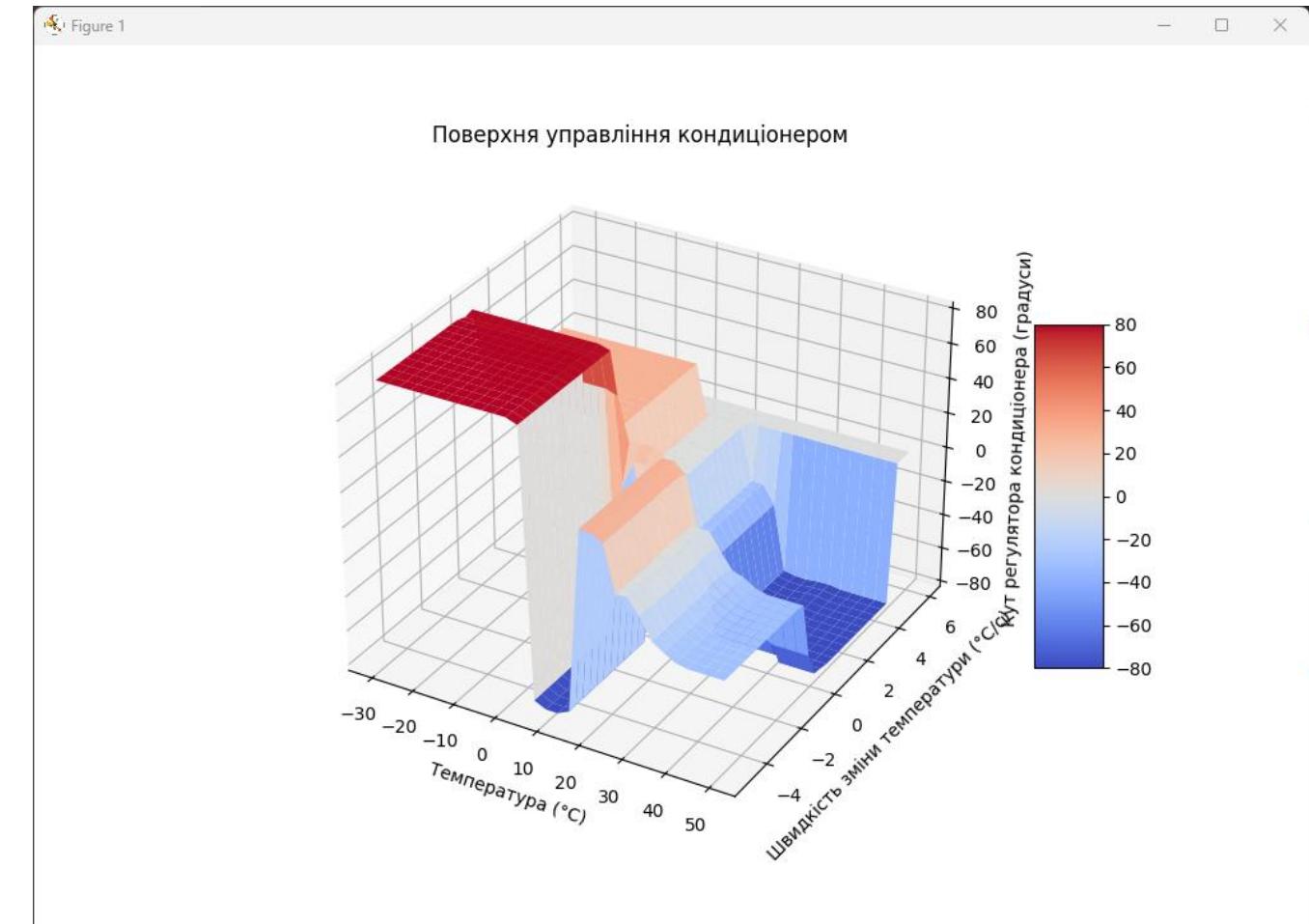


Рис. 1.4 Поверхня управління кондиціонером

```

--- Приклад 1: Температура 35°C, Швидкість зміни 1°C/с ---
Поворот кондиціонера: -78.33 градусів

--- Приклад 2: Температура 25°C, Швидкість зміни -1°C/с ---
Поворот кондиціонера: -0.00 градусів

--- Приклад 3: Температура -5°C, Швидкість зміни -2°C/с ---
Поворот кондиціонера: 80.00 градусів

--- Приклад 4: Температура 20°C, Швидкість зміни 0°C/с ---
Поворот кондиціонера: 0.00 градусів

--- Приклад 5: Температура 10°C, Швидкість зміни 2°C/с ---
Поворот кондиціонера: 0.00 градусів

```

Рис. 1.5 Результати симуляції

Вхідними змінними є температура, яка може бути дуже холодною, теплою або гарячою, і швидкість зміни температури, що може бути від'ємною (охолодження), нульовою або позитивною (нагрівання).

		<i>Кольцова Н.О.</i>			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		4

Для кожної змінної визначено функції належності, що дозволяють точно оцінювати параметри. Наприклад, температура поділена на категорії: дуже гаряча, тепла, нормальнa, холодна, дуже холодна. Для швидкості зміни температури створено три категорії: від'ємна, нульова і позитивна. Керування кондиціонером визначено п'ятьма положеннями: великі та малі кути вліво (охолодження), нейтральне (вимкнений режим), малі та великі кути вправо (нагрівання).

Система містить 15 правил, які визначають поведінку кондиціонера в залежності від температури та її зміни. Наприклад, якщо температура дуже гаряча і швидкість зміни температури позитивна, кондиціонер включається в режим охолодження з великим кутом вліво. Якщо температура нормальнa і швидкість зміни температури нульова, кондиціонер вимикається.

Після створення системи було проведено симуляції для різних умов. Наприклад, при температурі 35°C і швидкості зміни температури $1^{\circ}\text{C}/\text{s}$ кондиціонер працює на великому куті вліво (охолодження). Коли температура складає 25°C , а швидкість зміни температури $-1^{\circ}\text{C}/\text{s}$, кондиціонер працює на малому куті вліво (легке охолодження). В інших випадках, коли температура в приміщенні холодна або зміни температури незначні, кондиціонер або вимикається, або включає відповідний режим нагріву чи охолодження.

Візуалізація результатів показала, як змінюється кут регулятора кондиціонера в залежності від температури та швидкості зміни температури, надаючи повний контроль над режимом кондиціонера.

Ця модель ефективно автоматизує роботу кондиціонера, підтримуючи комфортні умови в приміщенні, і може бути адаптована для врахування інших факторів чи змін у роботі кондиціонера.

Висновок: За результатами класифікації побудована модель віднесла тестову точку до категорії доходу $\leq 50\text{K}$, що вказує на нижчий рівень заробітку згідно з наявними ознаками. Оцінка продуктивності моделі демонструє збалансовану якість: показник F1-міри становить 76.01%, що свідчить про достатньо узгоджене співвідношення точності та повноти. Значення accuracy (79.66%) також підтверджує

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Пр3	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

загальну надійність моделі та її здатність коректно класифікувати більшість прикладів.

Високий показник precision (78.88%) вказує на те, що модель робить відносно мало хибних позитивних передбачень, хоча при цьому recall (79.66%) демонструє збалансоване виявлення позитивних випадків. Сукупно ці метрики підтверджують, що модель має досить хорошу узагальнювальну здатність і може використовуватися для прогнозування доходів на основі заданих ознак. У разі необхідності підвищення точності для конкретного класу можливе додаткове покращення моделі шляхом зміни гіперпараметрів або застосування інших алгоритмів.

Посилання на github: <https://github.com/KoltcovaNadiia/Artificial-intelligence-systems-2025>

Висновок: У завданні були розроблені нечіткі моделі для керування кранами гарячої та холодної води, а також для кондиціонера. Моделі ефективно реагують на зміни входних параметрів, таких як температура і напір води або температура та швидкість її зміни. Графіки та результати симуляцій показали, що система коректно регулює кути відкриття кранів та зміни режиму роботи кондиціонера, забезпечуючи оптимальні умови. Ці моделі можуть бути використані для автоматизації водопостачальних систем та кондиціонерів у реальних умовах.

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		6

Лістинг програми:

LR_3_task_1.py

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

# Діапазони змінних
universe_temp = np.arange(0, 101, 1) # Температура води [0, 100] °C
universe_pressure = np.arange(0, 11, 0.1) # Тиск [0, 10] одиниць
universe_angle = np.arange(-90, 91, 1) # Кут повороту кранів від -90 до 90 градусів

# Створення нечітких змінних
temperature = ctrl.Antecedent(universe_temp, 'temperature')
pressure = ctrl.Antecedent(universe_pressure, 'pressure')
hot_tap = ctrl.Consequent(universe_angle, 'hot_tap', defuzzify_method='centroid')
cold_tap = ctrl.Consequent(universe_angle, 'cold_tap', defuzzify_method='centroid')

# Функції належності для температури води
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])

# Функції належності для тиску
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])

# Функції належності для кутів кранів
hot_tap['LL'] = fuzz.trimf(hot_tap.universe, [-90, -90, -60]) # Кран гарячої води сильно закритий
hot_tap['ML'] = fuzz.trimf(hot_tap.universe, [-75, -45, -15]) # Кран гарячої води середньо закритий
hot_tap['SL'] = fuzz.trimf(hot_tap.universe, [-30, -15, 0]) # Кран гарячої води слабо закритий
hot_tap['NC'] = fuzz.trimf(hot_tap.universe, [-10, 0, 10]) # Кран гарячої води нейтральний
hot_tap['SR'] = fuzz.trimf(hot_tap.universe, [0, 15, 30]) # Кран гарячої води слабо відкритий
hot_tap['MR'] = fuzz.trimf(hot_tap.universe, [15, 45, 75]) # Кран гарячої води середньо відкритий
```

		Кольцова Н.О.			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

```

hot_tap['LR'] = fuzz.trimf(hot_tap.universe, [60, 90, 90]) # Кран гарячої води сильно відкритий

cold_tap['LL'] = hot_tap['LL'].mf
cold_tap['ML'] = hot_tap['ML'].mf
cold_tap['SL'] = hot_tap['SL'].mf
cold_tap['NC'] = hot_tap['NC'].mf
cold_tap['SR'] = hot_tap['SR'].mf
cold_tap['MR'] = hot_tap['MR'].mf
cold_tap['LR'] = hot_tap['LR'].mf

# Створення правил для системи
rule1 = ctrl.Rule(temperature['hot'] & pressure['strong'],
                  (hot_tap['ML'], cold_tap['MR'])) # Гаряча вода і сильний напір
rule2 = ctrl.Rule(temperature['hot'] & pressure['not_very_strong'],
                  (hot_tap['NC'], cold_tap['MR'])) # Гаряча вода і не дуже сильний напір
rule3 = ctrl.Rule(temperature['not_very_hot'] & pressure['strong'],
                  (hot_tap['SL'], cold_tap['NC'])) # Не дуже гаряча вода і сильний напір
rule4 = ctrl.Rule(temperature['not_very_hot'] & pressure['weak'],
                  (hot_tap['SR'], cold_tap['SR'])) # Не дуже гаряча вода і слабкий напір
rule5 = ctrl.Rule(temperature['warm'] & pressure['not_very_strong'],
                  (hot_tap['NC'], cold_tap['NC'])) # Тепла вода і не дуже сильний напір
rule6 = ctrl.Rule(temperature['cool'] & pressure['strong'],
                  (hot_tap['MR'], cold_tap['ML'])) # Прохолодна вода і сильний напір
rule7 = ctrl.Rule(temperature['cool'] & pressure['not_very_strong'],
                  (hot_tap['MR'], cold_tap['SL'])) # Прохолодна вода і не дуже сильний напір
rule8 = ctrl.Rule(temperature['cold'] & pressure['weak'],
                  (hot_tap['LR'], cold_tap['NC'])) # Холодна вода і слабкий напір
rule9 = ctrl.Rule(temperature['cold'] & pressure['strong'],
                  (hot_tap['ML'], cold_tap['MR'])) # Холодна вода і сильний напір
rule10 = ctrl.Rule(temperature['warm'] & pressure['strong'],
                  (hot_tap['SL'], cold_tap['SL'])) # Тепла вода і сильний напір
rule11 = ctrl.Rule(temperature['warm'] & pressure['weak'],
                  (hot_tap['SR'], cold_tap['SR'])) # Тепла вода і слабкий напір

# Створення системи керування
tap_control_system = ctrl.ControlSystem(
    [rule1, rule2, rule3, rule4, rule5, rule6, rule7, rule8, rule9, rule10, rule11]
)

# Симуляція
tap_simulation = ctrl.ControlSystemSimulation(tap_control_system)

```

Змн.	Арк.	№ докум.	Підпис	Дата
------	------	----------	--------	------

```

# Визначимо приклади для симуляції
examples = [
    {"temperature": 90, "pressure": 9}, # Гаряча вода, сильний напір
    {"temperature": 20, "pressure": 4}, # Прохолодна вода, слабкий напір
    {"temperature": 55, "pressure": 7}, # Тепла вода, середній напір
    {"temperature": 10, "pressure": 2}, # Холодна вода, слабкий напір
    {"temperature": 45, "pressure": 8}, # Тепла вода, сильний напір
    {"temperature": 80, "pressure": 3}, # Гаряча вода, не дуже сильний напір
]

# Для кожного прикладу виконуємо симуляцію
for i, example in enumerate(examples, 1):
    tap_simulation.input['temperature'] = example["temperature"]
    tap_simulation.input['pressure'] = example["pressure"]
    tap_simulation.compute()

    print(f"\n--- Приклад {i}: Температура {example['temperature']}°C, Напір
{example['pressure']} ---")
    print(f"Поворот гарячого крана: {tap_simulation.output['hot_tap']:.2f} градусів")
    print(f"Поворот холодного крана: {tap_simulation.output['cold_tap']:.2f} граду-
сів")

# Візуалізація результатів (поверхні управління)
temp_range_plot = np.linspace(universe_temp.min(), universe_temp.max(), 30)
pres_range_plot = np.linspace(universe_pressure.min(), universe_pressure.max(), 30)
temp_grid, pres_grid = np.meshgrid(temp_range_plot, pres_range_plot)

hot_tap_output = np.zeros_like(temp_grid)
cold_tap_output = np.zeros_like(temp_grid)

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]
        tap_simulation.compute()
        hot_tap_output[i, j] = tap_simulation.output.get('hot_tap', 0) # Без помилки,
якщо ключ відсутній
        cold_tap_output[i, j] = tap_simulation.output.get('cold_tap', 0) # Без помилки,
якщо ключ відсутній

# Побудова графіків
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='coolwarm')

```

		<i>Кольцова Н.О.</i>				Арк.
		<i>Маєвський О.В.</i>				ДУ «Житомирська політехніка».25. 121.14..000 – ПрЗ
Змн.	Арк.	№ докум.	<i>Підпись</i>	<i>Дата</i>		9

```

ax1.set_xlabel('Температура (°C)')
ax1.set_ylabel('Напір (0-10)')
ax1.set_zlabel('Кут гарячого крана (градуси)')
ax1.set_title('Поверхня управління гарячим краном')
fig1.colorbar(surf1, shrink=0.5, aspect=5)

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='coolwarm')
ax2.set_xlabel('Температура (°C)')
ax2.set_ylabel('Напір (0-10)')
ax2.set_zlabel('Кут холодного крана (градуси)')
ax2.set_title('Поверхня управління холодним краном')
fig2.colorbar(surf2, shrink=0.5, aspect=5)

plt.show()

```

LR_3_task_2.py

```

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

# Діапазони змінних
universe_temp = np.arange(-30, 51, 1) # Температура повітря [ °C]
universe_rate = np.arange(-5, 6, 0.1) # Швидкість зміни температури [ °C/c ]
universe_angle = np.arange(-90, 91, 1) # Кут повороту регулятора кондиціонера

# Створення нечітких змінних
temperature = ctrl.Antecedent(universe_temp, 'temperature')
rate_of_change = ctrl.Antecedent(universe_rate, 'rate_of_change')
ac_control = ctrl.Consequent(universe_angle, 'ac_control',
defuzzify_method='centroid')

# Функції належності для температури
temperature['very_hot'] = fuzz.trapmf(temperature.universe, [30, 40, 50, 50]) # Дуже гаряча
temperature['hot'] = fuzz.trimf(temperature.universe, [20, 30, 40]) # Тепла
temperature['normal'] = fuzz.trimf(temperature.universe, [15, 20, 25]) # Нормальна
temperature['cold'] = fuzz.trimf(temperature.universe, [5, 10, 15]) # Холодна
temperature['very_cold'] = fuzz.trapmf(temperature.universe, [-30, -30, 0, 5]) # Дуже холодна

# Функції належності для швидкості зміни температури

```

		<i>Кольцова Н.О.</i>			<i>ДУ «Житомирська політехніка».25. 121.14..000 – Прз</i>	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		10

```

rate_of_change['negative'] = fuzz.trapmf(rate_of_change.universe, [-5, -5, -1, 0]) # Від'ємна швидкість
rate_of_change['zero'] = fuzz.trimf(rate_of_change.universe, [-0.5, 0, 0.5]) # Швидкість 0
rate_of_change['positive'] = fuzz.trapmf(rate_of_change.universe, [0, 1, 5, 5]) # Позитивна швидкість

# Функції належності для контролю кондиціонером
ac_control['large_left'] = fuzz.trimf(ac_control.universe, [-90, -90, -60]) # Великий кут вліво (холод)
ac_control['small_left'] = fuzz.trimf(ac_control.universe, [-60, -30, 0]) # Малий кут вліво (холод)
ac_control['neutral'] = fuzz.trimf(ac_control.universe, [-10, 0, 10]) # Нейтральний
ac_control['small_right'] = fuzz.trimf(ac_control.universe, [0, 30, 60]) # Малий кут вправо (тепло)
ac_control['large_right'] = fuzz.trimf(ac_control.universe, [60, 90, 90]) # Великий кут вправо (тепло)

# Створення правил для системи
rule1 = ctrl.Rule(temperature['very_hot'] & rate_of_change['positive'],
ac_control['large_left']) # Дуже гаряча і позитивна зміна -> великий кут вліво
rule2 = ctrl.Rule(temperature['very_hot'] & rate_of_change['negative'],
ac_control['small_left']) # Дуже гаряча і від'ємна зміна -> малий кут вліво
rule3 = ctrl.Rule(temperature['hot'] & rate_of_change['positive'],
ac_control['large_left']) # Тепла і позитивна зміна -> великий кут вліво
rule4 = ctrl.Rule(temperature['hot'] & rate_of_change['negative'],
ac_control['neutral']) # Тепла і від'ємна зміна -> вимкнути
rule5 = ctrl.Rule(temperature['very_cold'] & rate_of_change['negative'],
ac_control['large_right']) # Дуже холодна і від'ємна зміна -> великий кут вправо
rule6 = ctrl.Rule(temperature['very_cold'] & rate_of_change['positive'],
ac_control['small_right']) # Дуже холодна і позитивна зміна -> малий кут вправо
rule7 = ctrl.Rule(temperature['cold'] & rate_of_change['negative'],
ac_control['large_left']) # Холодна і від'ємна зміна -> великий кут вліво
rule8 = ctrl.Rule(temperature['cold'] & rate_of_change['positive'],
ac_control['neutral']) # Холодна і позитивна зміна -> вимкнути
rule9 = ctrl.Rule(temperature['very_hot'] & rate_of_change['zero'],
ac_control['large_left']) # Дуже гаряча і швидкість 0 -> великий кут вліво
rule10 = ctrl.Rule(temperature['hot'] & rate_of_change['zero'],
ac_control['small_left']) # Тепла і швидкість 0 -> малий кут вліво
rule11 = ctrl.Rule(temperature['very_cold'] & rate_of_change['zero'],
ac_control['large_right']) # Дуже холодна і швидкість 0 -> великий кут вправо
rule12 = ctrl.Rule(temperature['cold'] & rate_of_change['zero'],
ac_control['small_right']) # Холодна і швидкість 0 -> малий кут вправо
rule13 = ctrl.Rule(temperature['normal'] & rate_of_change['positive'],
ac_control['small_left']) # Нормальна і позитивна зміна -> малий кут вліво
rule14 = ctrl.Rule(temperature['normal'] & rate_of_change['negative'],
ac_control['small_right']) # Нормальна і від'ємна зміна -> малий кут вправо
rule15 = ctrl.Rule(temperature['normal'] & rate_of_change['zero'],
ac_control['neutral']) # Нормальна і швидкість 0 -> вимкнути

```

		<i>Кольцова Н.О.</i>							
		<i>Маєвський О.В.</i>							
Змн.	Арк.	№ докум.	Підпис	Дата					Арк.

```

# Створення системи керування
ac_system = ctrl.ControlSystem([rule1, rule2, rule3, rule4, rule5, rule6, rule7,
rule8, rule9, rule10, rule11, rule12, rule13, rule14, rule15])

# Симуляція
ac_simulation = ctrl.ControlSystemSimulation(ac_system)

# Визначимо декілька прикладів для симуляції
examples = [
    {"temperature": 35, "rate_of_change": 1}, # Дуже гаряча, позитивна зміна
    {"temperature": 25, "rate_of_change": -1}, # Тепла, від'ємна зміна
    {"temperature": -5, "rate_of_change": -2}, # Дуже холодна, від'ємна зміна
    {"temperature": 20, "rate_of_change": 0}, # Нормальна, швидкість 0
    {"temperature": 10, "rate_of_change": 2}, # Холодна, позитивна зміна
]

# Для кожного прикладу виконуємо симуляцію
for i, example in enumerate(examples, 1):
    ac_simulation.input['temperature'] = example["temperature"]
    ac_simulation.input['rate_of_change'] = example["rate_of_change"]
    ac_simulation.compute()

    print(f"\n--- Приклад {i}: Температура {example['temperature']}°C, Швидкість зміни {example['rate_of_change']}°C/c ---")
    print(f"Поворот кондиціонера: {ac_simulation.output['ac_control']:.2f} градусів")

# Візуалізація результатів
temp_range_plot = np.linspace(universe_temp.min(), universe_temp.max(), 30)
rate_range_plot = np.linspace(universe_rate.min(), universe_rate.max(), 30)
temp_grid, rate_grid = np.meshgrid(temp_range_plot, rate_range_plot)

ac_output = np.zeros_like(temp_grid)

for i in range(temp_grid.shape[0]):
    for j in range(temp_grid.shape[1]):
        ac_simulation.input['temperature'] = temp_grid[i, j]
        ac_simulation.input['rate_of_change'] = rate_grid[i, j]
        ac_simulation.compute()
        ac_output[i, j] = ac_simulation.output.get('ac_control', 0) # Без помилки, якщо ключ відсутній

# Побудова графіку
fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')
surf = ax.plot_surface(temp_grid, rate_grid, ac_output, cmap='coolwarm')
ax.set_xlabel('Температура (°C)')
ax.set_ylabel('Швидкість зміни температури (°C/c)')
ax.set_zlabel('Кут регулятора кондиціонера (градуси)')
ax.set_title('Поверхня управління кондиціонером')
fig.colorbar(surf, shrink=0.5, aspect=5)

```

		Кольцова Н.О.		
		Маєвський О.В.		
Змн.	Арк.	№ докум.	Підпис	Дата

```
plt.show()
```

		<i>Кольцова Н.О.</i>			ДУ «Житомирська політехніка».25. 121.14..000 – Прз	Арк.
		Маєвський О.В.				
Змн.	Арк.	№ докум.	Підпис	Дата		13