

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.

Git: <https://github.com/flekXD/SAI>

Завдання 1.

Побудова нечіткої моделі системи керування кранами гарячої і холодної води. При користуванні системою водопостачання на вхід змішувача подається холодна та гаряча вода по відповідним трубопроводам. Задача полягає у створенні моделі системи засобами Matlab Fuzzy Logic, яка б дозволила автоматизувати процес. Кран змішувача можна повертати наліво і направо (тобто, область визначення кута - це відрізок $[-90;90]$ градусів), керуючи тим самим температурою води і її напором. Нехай, повернення будьякого крану направо - це збільшити потік води відповідної температури.

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

temperature = ctrl.Antecedent(np.arange(0, 101, 1), 'temperature') # Температура
[0-100 °C]
pressure = ctrl.Antecedent(np.arange(0, 101, 1), 'pressure') # Напір [0-100
%]

hot_valve = ctrl.Consequent(np.arange(-90, 91, 1), 'hot_valve') # Кут крана
гарячої води
cold_valve = ctrl.Consequent(np.arange(-90, 91, 1), 'cold_valve') # Кут крана
холодної води

# Функції належності для температури
temperature['cold'] = fuzz.trimf(temperature.universe, [0, 0, 50])
temperature['cool'] = fuzz.trimf(temperature.universe, [0, 50, 75])
temperature['warm'] = fuzz.trimf(temperature.universe, [50, 75, 100])
temperature['hot'] = fuzz.trimf(temperature.universe, [75, 100, 100])

# Функції належності для напору
pressure['low'] = fuzz.trimf(pressure.universe, [0, 0, 50])
pressure['medium'] = fuzz.trimf(pressure.universe, [25, 50, 75])
pressure['high'] = fuzz.trimf(pressure.universe, [50, 100, 100])

# Функції належності для кутів
hot_valve['large_left'] = fuzz.trimf(hot_valve.universe, [-90, -90, -45])
```

| | | | | | | |
|------|------|---------------|--------|------|--|------|
| | | Кириченко О С | | | ДУ «Житомирська політехніка».21.121.00.000 – Лр1 | Арк. |
| | | Голенко М. Ю. | | | | 1 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

hot_valve['medium_left'] = fuzz.trimf(hot_valve.universe, [-60, -30, 0])
hot_valve['small_left'] = fuzz.trimf(hot_valve.universe, [-30, 0, 30])
hot_valve['no_change'] = fuzz.trimf(hot_valve.universe, [-15, 0, 15])
hot_valve['small_right'] = fuzz.trimf(hot_valve.universe, [0, 30, 60])
hot_valve['medium_right'] = fuzz.trimf(hot_valve.universe, [30, 60, 90])
hot_valve['large_right'] = fuzz.trimf(hot_valve.universe, [45, 90, 90])

cold_valve['large_left'] = fuzz.trimf(cold_valve.universe, [-90, -90, -45])
cold_valve['medium_left'] = fuzz.trimf(cold_valve.universe, [-60, -30, 0])
cold_valve['small_left'] = fuzz.trimf(cold_valve.universe, [-30, 0, 30])
cold_valve['no_change'] = fuzz.trimf(cold_valve.universe, [-15, 0, 15])
cold_valve['small_right'] = fuzz.trimf(cold_valve.universe, [0, 30, 60])
cold_valve['medium_right'] = fuzz.trimf(cold_valve.universe, [30, 60, 90])
cold_valve['large_right'] = fuzz.trimf(cold_valve.universe, [45, 90, 90])

# Правила нечіткої логіки
rules = [
    ctrl.Rule(temperature['hot'] & pressure['high'],
               (hot_valve['medium_left'], cold_valve['medium_right'])),
    ctrl.Rule(temperature['hot'] & pressure['medium'],
               cold_valve['medium_right']),
    ctrl.Rule(temperature['warm'] & pressure['high'],
               hot_valve['small_left']),
    ctrl.Rule(temperature['warm'] & pressure['low'],
               (hot_valve['small_right'], cold_valve['small_right'])),
    ctrl.Rule(temperature['cool'] & pressure['medium'],
               (hot_valve['no_change'], cold_valve['no_change'])),
    ctrl.Rule(temperature['cool'] & pressure['high'],
               (hot_valve['medium_right'], cold_valve['medium_left'])),
    ctrl.Rule(temperature['cool'] & pressure['low'],
               (hot_valve['medium_right'], cold_valve['small_left'])),
    ctrl.Rule(temperature['cold'] & pressure['low'],
               hot_valve['large_right']),
    ctrl.Rule(temperature['cold'] & pressure['high'],
               (hot_valve['medium_left'], cold_valve['medium_right'])),
    ctrl.Rule(temperature['warm'] & pressure['high'],
               (hot_valve['small_left'], cold_valve['small_left'])),
    ctrl.Rule(temperature['warm'] & pressure['low'],
               (hot_valve['small_right'], cold_valve['small_right']))
]

# Система керування
control_system = ctrl.ControlSystem(rules)
mixer = ctrl.ControlSystemSimulation(control_system)

# Візуалізація
def visualize_memberships():
    temperature.view()
    pressure.view()
    hot_valve.view()
    cold_valve.view()

```

```
def simulate_and_visualize(temp, pres):
    mixer.input['temperature'] = temp
    mixer.input['pressure'] = pres
    mixer.compute()
    hot_valve.view(sim=mixer)
    cold_valve.view(sim=mixer)

visualize_memberships()

simulate_and_visualize(70, 80)

plt.show()
```

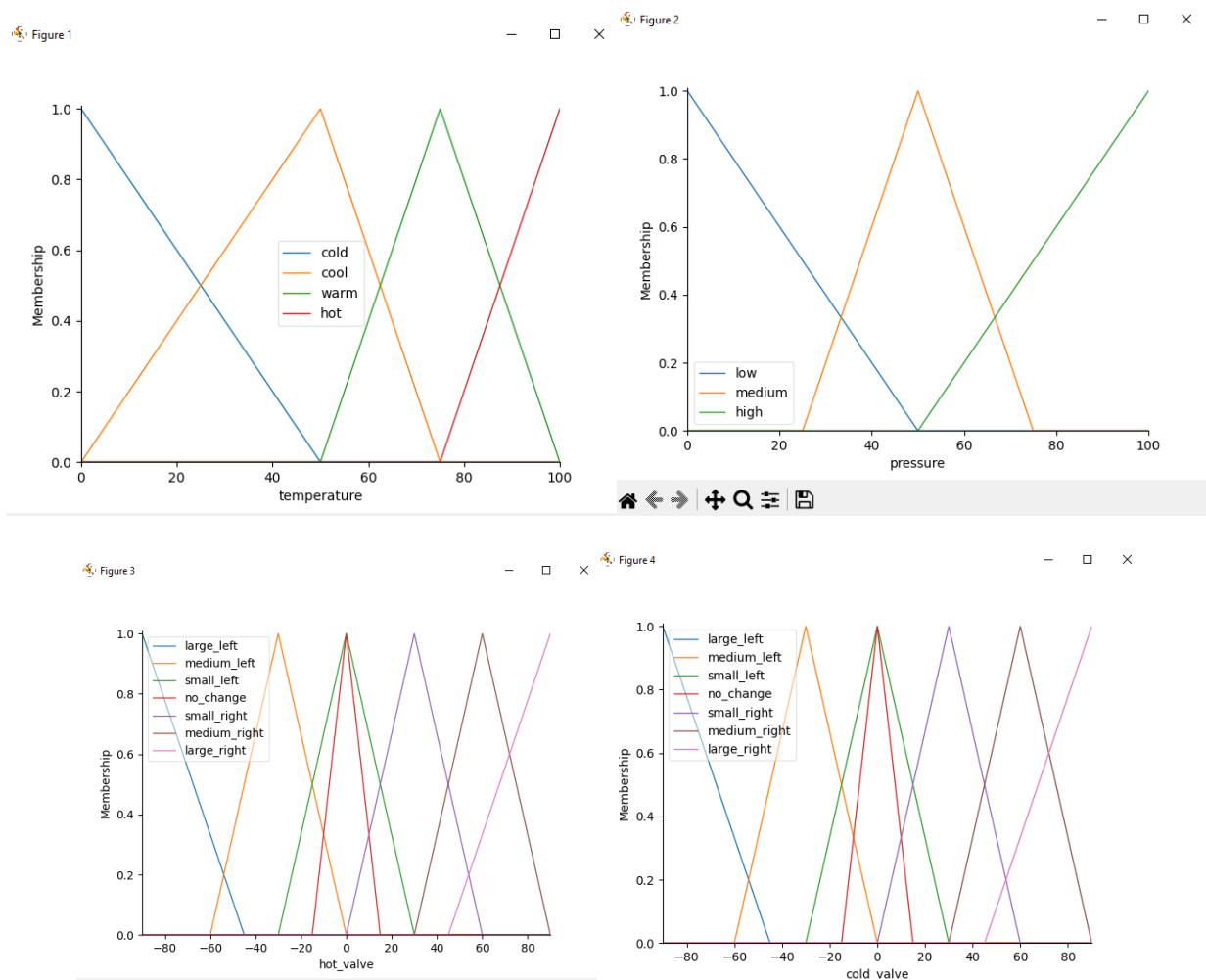


Figure 5

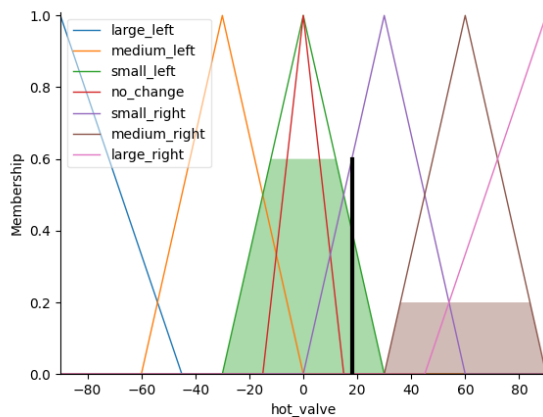
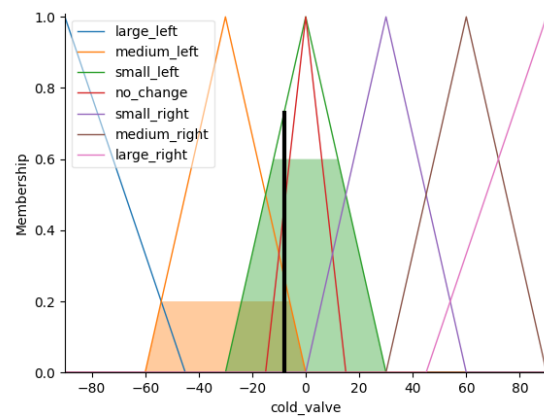


Figure 6



Модель дозволяє врахувати комбіновану дію двох факторів (температури та напору) для точного регулювання кранів.

Використання нечіткої логіки забезпечує плавність переходу між станами та можливість адаптації до різних ситуацій.

Завдання 2.

Нечітка модель керування кондиціонером повітря в приміщенні. Нехай, в приміщенні встановлений кондиціонер, який дозволяє регулювати (нагрівати чи охолоджувати) температуру. Найбільш комфортні умови складаються при встановленні деякої заданої комфортної температури. Задача полягає у розробці АСУ, яка б змогла автоматизувати роботу кондиціонера при коливанні температури приміщення через різні зовнішні дестабілізуючі фактори.

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

temperature = ctrl.Antecedent(np.arange(0, 41, 1), 'temperature') # Температура (°C)
temp_change = ctrl.Antecedent(np.arange(-5, 6, 1), 'temp_change') # Швидкість зміни температури (°C/хв)

ac_regulator = ctrl.Consequent(np.arange(-90, 91, 1), 'ac_regulator') # Кут регулятора кондиціонера

# Функції належності для температури
temperature['very_cold'] = fuzz.trimf(temperature.universe, [0, 0, 10])
temperature['cold'] = fuzz.trimf(temperature.universe, [5, 10, 20])
temperature['normal'] = fuzz.trimf(temperature.universe, [15, 20, 25])
```

```

temperature['warm'] = fuzz.trimf(temperature.universe, [20, 30, 35])
temperature['very_warm'] = fuzz.trimf(temperature.universe, [30, 40, 40])

# Функції належності для швидкості зміни температури
temp_change['negative'] = fuzz.trimf(temp_change.universe, [-5, -5, 0])
temp_change['zero'] = fuzz.trimf(temp_change.universe, [-1, 0, 1])
temp_change['positive'] = fuzz.trimf(temp_change.universe, [0, 5, 5])

# Функції належності для регулятора
ac_regulator['large_left'] = fuzz.trimf(ac_regulator.universe, [-90, -90, -45])
ac_regulator['small_left'] = fuzz.trimf(ac_regulator.universe, [-45, -15, 0])
ac_regulator['no_change'] = fuzz.trimf(ac_regulator.universe, [-15, 0, 15])
ac_regulator['small_right'] = fuzz.trimf(ac_regulator.universe, [0, 15, 45])
ac_regulator['large_right'] = fuzz.trimf(ac_regulator.universe, [45, 90, 90])

# Правила нечіткої логіки
rules = [
    ctrl.Rule(temperature['very_warm'] & temp_change['positive'],
ac_regulator['large_left']),
    ctrl.Rule(temperature['very_warm'] & temp_change['negative'],
ac_regulator['small_left']),
    ctrl.Rule(temperature['very_warm'] & temp_change['zero'],
ac_regulator['large_left']),
    ctrl.Rule(temperature['warm'] & temp_change['positive'],
ac_regulator['large_left']),
    ctrl.Rule(temperature['warm'] & temp_change['negative'],
ac_regulator['no_change']),
    ctrl.Rule(temperature['warm'] & temp_change['zero'],
ac_regulator['small_left']),
    ctrl.Rule(temperature['very_cold'] & temp_change['negative'],
ac_regulator['large_right']),
    ctrl.Rule(temperature['very_cold'] & temp_change['positive'],
ac_regulator['small_right']),
    ctrl.Rule(temperature['very_cold'] & temp_change['zero'],
ac_regulator['large_right']),
    ctrl.Rule(temperature['cold'] & temp_change['negative'],
ac_regulator['large_right']),
    ctrl.Rule(temperature['cold'] & temp_change['positive'],
ac_regulator['no_change']),
    ctrl.Rule(temperature['cold'] & temp_change['zero'],
ac_regulator['small_right']),
    ctrl.Rule(temperature['normal'] & temp_change['positive'],
ac_regulator['small_left']),
    ctrl.Rule(temperature['normal'] & temp_change['negative'],
ac_regulator['small_right']),
    ctrl.Rule(temperature['normal'] & temp_change['zero'],
ac_regulator['no_change'])
]

# Система керування
control_system = ctrl.ControlSystem(rules)
ac_simulation = ctrl.ControlSystemSimulation(control_system)

```

| | | | | | | |
|------|------|---------------|--------|------|--|------|
| | | Кириченко О С | | | ДУ «Житомирська політехніка».21.121.00.000 – Лр1 | Арк. |
| | | Голенко М. Ю. | | | | 5 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |

```

# Візуалізація
def visualize_memberships():
    temperature.view()
    temp_change.view()
    ac_regulator.view()

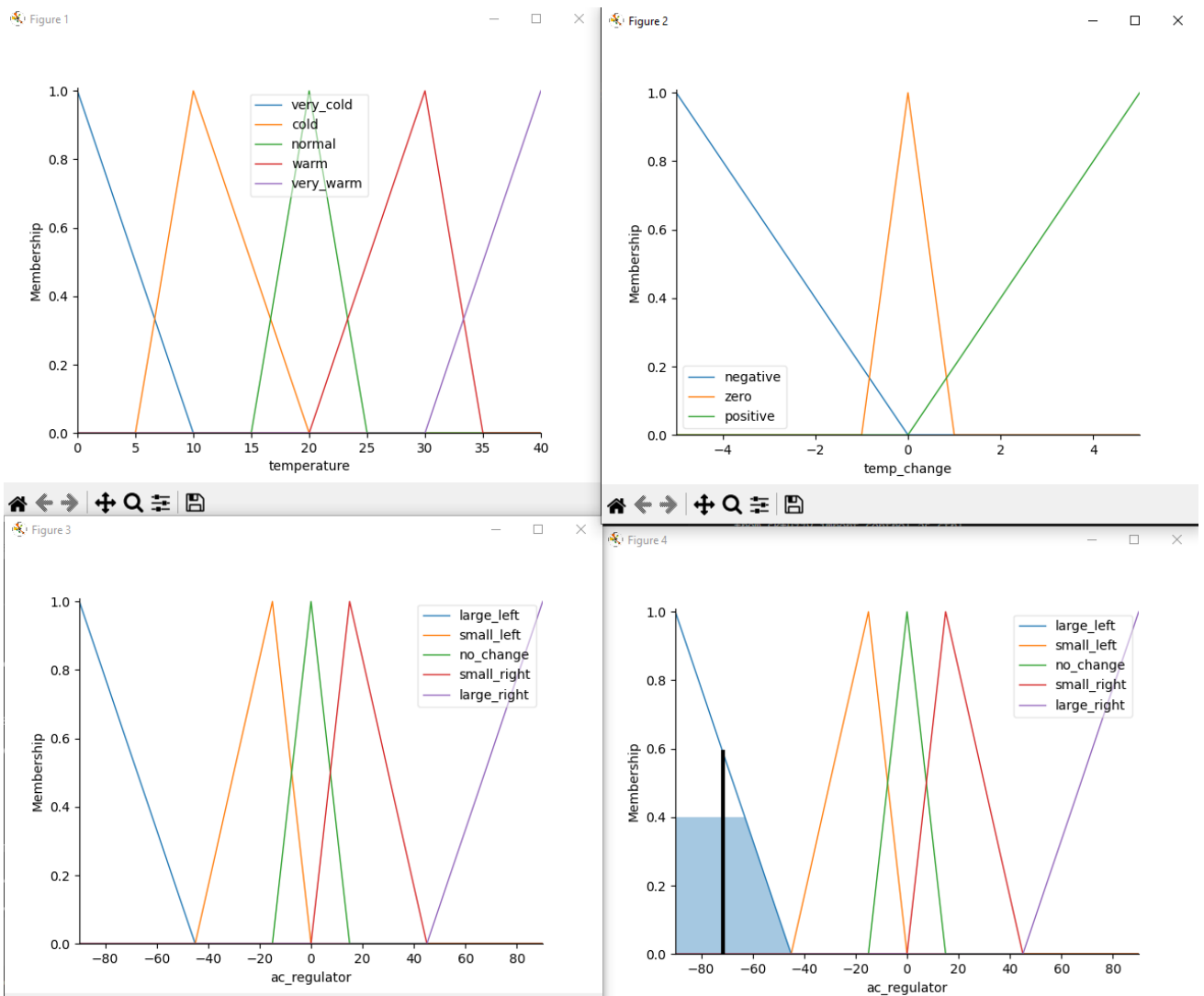
def simulate_and_visualize(temp, temp_rate_change):
    ac_simulation.input['temperature'] = temp
    ac_simulation.input['temp_change'] = temp_rate_change
    ac_simulation.compute()
    ac_regulator.view(sim=ac_simulation)

# Візуалізація функцій належності
visualize_memberships()

# Тестування моделі з візуалізацією
simulate_and_visualize(28, 2)

plt.show()

```



Використання нечіткої логіки забезпечує плавну та адаптивну роботу системи в умовах змінних зовнішніх впливів.

Модель враховує інертність кондиціонера, що дозволяє уникнути надмірного регулювання.

Ефективне керування кондиціонером можливе навіть за складних умов, що включають зростання чи спад температури.

Висновок:

Використання нечіткої логіки у двох завданнях дозволяє автоматизувати процеси регулювання з урахуванням багатofакторних умов. Гнучкість нечітких моделей робить їх придатними для задач, де складно сформулювати чіткі математичні залежності. Завдяки візуалізації функцій належності та результатів можна легко інтерпретувати дії моделей. Для обох задач можна додатково покращити точність шляхом оптимізації функцій належності або додавання нових правил.

| | | | | | | |
|------|------|---------------|--------|------|--|------|
| | | Кириченко О С | | | ДУ «Житомирська політехніка».21.121.00.000 – Лр1 | Арк. |
| | | Голенко М. Ю. | | | | 7 |
| Змн. | Арк. | № докум. | Підпис | Дата | | |