

## Лабораторна робота №3

### МОДЕЛЮВАННЯ ЕЛЕМЕНТІВ НЕЧІТКИХ МНОЖИН ТА ФОРМУВАННЯ НЕЧІТКИХ ПРАВИЛ

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

#### Хід роботи

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 будь-якого крану направо - це збільшити потік води відповідної температури.

Лістинг коду:

```
% Нечіткі функції належності для температури та тиску
x_range = linspace(-100, 100, 300); % Розширений діапазон значень для x (температура, тиск, кут)

% Трапецієподібна функція належності
def_trapmf = @(x, a, b, c, d) max(min((x - a) / (b - a), (d - x) / (d - c)), 0);
% Трикутна функція належності
def_trimf = @(x, a, b, c) max(min((x - a) / (b - a), (c - x) / (c - b)), 0);

% Функції належності для температури
temp_cold = def_trapmf(x_range, -15, -5, 10, 25);
temp_warm = def_trimf(x_range, 15, 35, 55);
temp_hot = def_trapmf(x_range, 50, 70, 90, 110);

% Функції належності для тиску
pressure_weak = def_trapmf(x_range, -2, 0, 2, 4);
pressure_normal = def_trimf(x_range, 3, 6, 9);
pressure_strong = def_trapmf(x_range, 8, 10, 12, 14);

% Тестові значення
test_temperature = 65; % Приклад температури
test_pressure = 6; % Приклад тиску

% Розрахунок значень належності для температури та тиску
membership_temp_cold = def_trapmf(test_temperature, -15, -5, 10, 25);
membership_temp_warm = def_trimf(test_temperature, 15, 35, 55);
membership_temp_hot = def_trapmf(test_temperature, 50, 70, 90, 110);
```

					ДУ «Житомирська політехніка».24.121.16.000 – Лр3			
Змн.	Арк.	№ докум.	Підпис	Дата	Звіт з лабораторної роботи  ФІКТ Гр. ІПЗ-21-5			
Розроб.		Некритий В.Ю.						
Перевір.		Іванов Д.А.						
Керівник								
Н. контр.								
Зав. каф.								
					Літ.	Арк.	Аркуші	
						1	4	

```

membership_pressure_weak = def_trapmf(test_pressure, -2, 0, 2, 4);
membership_pressure_normal = def_trapmf(test_pressure, 3, 6, 9);
membership_pressure_strong = def_trapmf(test_pressure, 8, 10, 12, 14);

% Виведення значень належності для тестових значень
disp("Належність температури:");
disp(["Холодна вода: ", num2str(membership_temp_cold)]);
disp(["Тепла вода: ", num2str(membership_temp_warm)]);
disp(["Гаряча вода: ", num2str(membership_temp_hot)]);

disp("Належність тиску:");
disp(["Слабкий тиск: ", num2str(membership_pressure_weak)]);
disp(["Нормальний тиск: ", num2str(membership_pressure_normal)]);
disp(["Сильний тиск: ", num2str(membership_pressure_strong)]);

% Графіки функцій належності
subplot(2, 2, 1);
plot(x_range, temp_cold, 'r--', x_range, temp_warm, 'g--', x_range, temp_hot, 'b--');
title('Функції належності температури');
legend('Холодна', 'Тепла', 'Гаряча');
xlabel('Температура');
ylabel('Рівень належності');

title('Функції належності тиску');
subplot(2, 2, 2);
plot(x_range, pressure_weak, 'm-.', x_range, pressure_normal, 'c-.', x_range,
pressure_strong, 'k-.');
title('Функції належності тиску');
legend('Слабкий', 'Нормальний', 'Сильний');
xlabel('Тиск');
ylabel('Рівень належності');

% Оцінка правил
hot_valve_angle = 0;
cold_valve_angle = 0;

% Правила для гарячої води
hot_valve_angle = max(hot_valve_angle, min(membership_temp_cold,
membership_pressure_strong) * -40); % Змінене правило 1
hot_valve_angle = max(hot_valve_angle, min(membership_temp_hot,
membership_pressure_normal) * -50); % Змінене правило 2
hot_valve_angle = max(hot_valve_angle, min(membership_temp_warm,
membership_pressure_strong) * -25); % Змінене правило 3
hot_valve_angle = max(hot_valve_angle, min(membership_temp_warm, membership_pressure_weak)
* 20); % Змінене правило 4

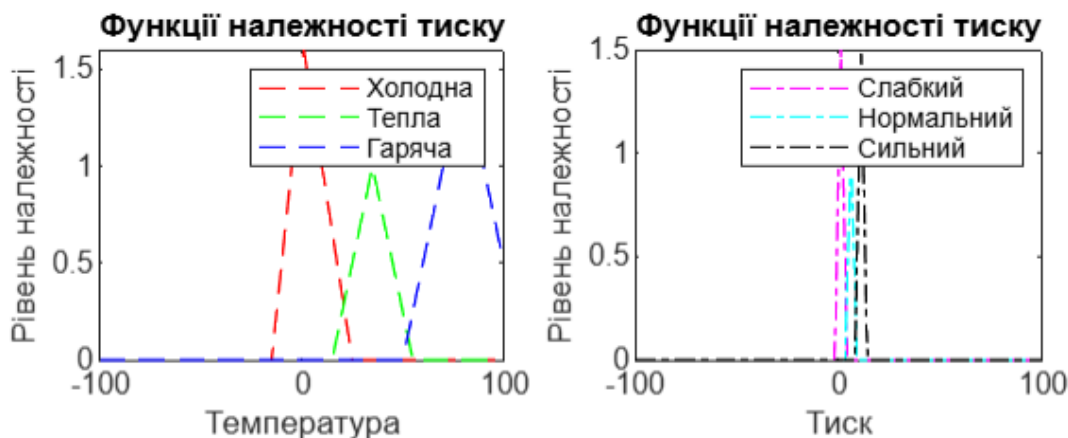
% Правила для холодної води
cold_valve_angle = max(cold_valve_angle, min(membership_temp_cold,
membership_pressure_strong) * 50); % Змінене правило 5
cold_valve_angle = max(cold_valve_angle, min(membership_temp_warm,
membership_pressure_strong) * -10); % Змінене правило 6
cold_valve_angle = max(cold_valve_angle, min(membership_temp_hot,
membership_pressure_weak) * 25); % Змінене правило 7
cold_valve_angle = max(cold_valve_angle, min(membership_temp_hot,
membership_pressure_strong) * -35); % Змінене правило 8

% Виведення кінцевих результатів
disp('Кут відкриття гарячого крану:');
disp(hot_valve_angle);
disp('Кут відкриття холодного крану:');
disp(cold_valve_angle);

```

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – ПрЗ	Арк.
		Іванов Д.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		2

Результат виконання:



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.

Лістинг коду:

```
% Основний код для визначення режиму роботи кондиціонера на основі температури та її зміни

% Початкові значення температури та швидкості її зміни
temperature = 18; % Температура в приміщенні (°C)
temp_change_rate = -1; % Швидкість зміни температури (°C/хв)

% Ініціалізація змінних для режиму роботи кондиціонера та часу
ac_state = zeros(1, 11); % Збереження режимів для побудови графіка (11 точок часу)
time_points = 0:10; % Час для побудови графіка (від 0 до 10 хв)

% Цикл для моделювання змін у режимі роботи кондиціонера
for t = 1:length(time_points)
    % Правила керування залежно від температури та швидкості її зміни
    if temperature > 30 && temp_change_rate > 0
        ac_state(t) = 70; % Великий кут вліво для охолодження
    elseif temperature > 30 && temp_change_rate < 0
        ac_state(t) = 40; % Малий кут вліво для охолодження
    elseif temperature > 20 && temp_change_rate > 0
        ac_state(t) = 70; % Великий кут вліво для охолодження
    elseif temperature > 20 && temp_change_rate < 0
        ac_state(t) = 10; % Вимкнути кондиціонер
    elseif temperature < 15 && temp_change_rate < 0
        ac_state(t) = 90; % Великий кут вправо для обігріву
    elseif temperature < 15 && temp_change_rate > 0
        ac_state(t) = 50; % Малий кут вправо для обігріву
    elseif temperature < 20 && temp_change_rate < 0
        ac_state(t) = 70; % Великий кут вліво для охолодження
    elseif temperature < 20 && temp_change_rate > 0
        ac_state(t) = 10; % Вимкнути кондиціонер
    end
end
```

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – Лр3	Арк.
		Іванов Д.А.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

elseif temperature == 0 && temp_change_rate == 0
    ac_state(t) = 10; % Вимкнути кондиціонер
else
    ac_state(t) = 10; % За замовчуванням: вимкнути кондиціонер
end

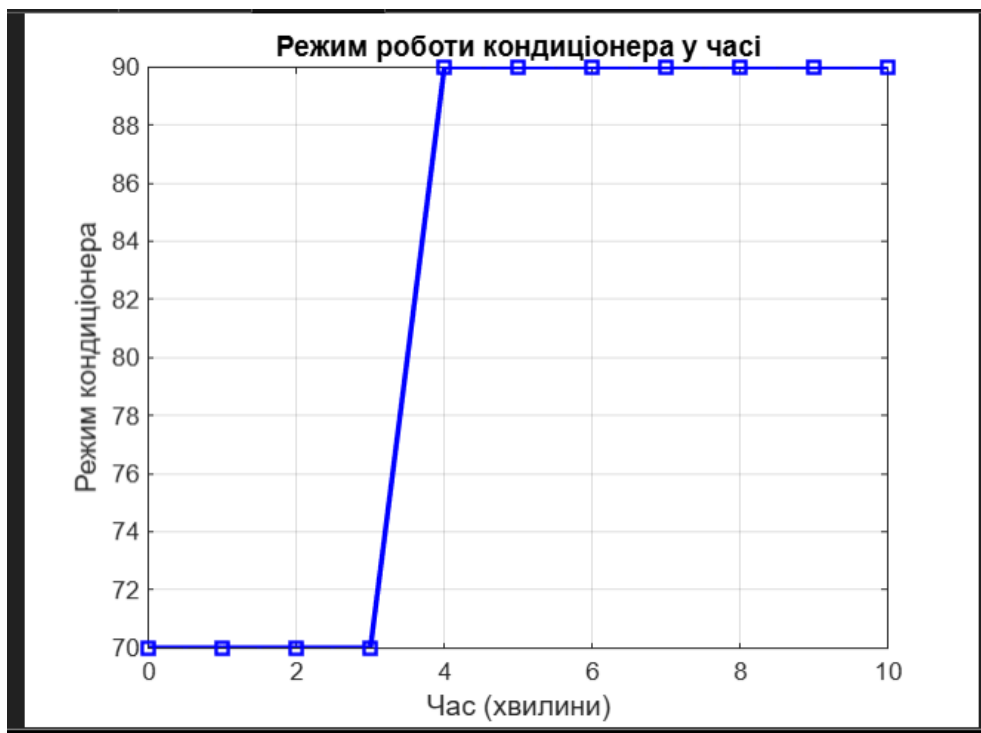
% Оновлення температури та швидкості її зміни
temperature = temperature + temp_change_rate; % Зміна температури
end

% Побудова графіка режиму роботи кондиціонера у часі
figure;
plot(time_points, ac_state, 'LineWidth', 2, 'Marker', 's', 'Color', 'b'); % Графік із
синім кольором і маркерами
xlabel('Час (хвилини)');
ylabel('Режим кондиціонера');
title('Режим роботи кондиціонера у часі');
grid on;

% Виведення результату для останнього моменту часу
fprintf('Остаточний режим роботи кондиціонера: %.2f\n', ac_state(end));

```

Результат виконання:



**Висновок:** дослідив можливості ППП MATLAB щодо проектування систем керування на основі алгоритмів нечіткого виводу.

		Некритий В.Ю.			ДУ «Житомирська політехніка».24.121.16.000 – Лр3	Арк.
		Іванов Д.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		4