

## МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт

з дисципліни **МОДЕЛІ І МЕТОДИ НЕЧІТКОЇ ЛОГІКИ**

для студентів III курсу

Рівень вищої освіти – «Бакалавр»

Спеціальність – 122 – «Комп'ютерні науки»

ЗАТВЕРДЖЕНО

на засіданні групи забезпечення  
спеціальності 122 Комп'ютерні науки від  
« 20 » 05 2021 року протокол  
№ 13

Голова групи  (Мещеряков В.І.)

Затверджено на засіданні кафедри

Інформаційних технологій

Протокол № 9 від 18.05.21

В.О. Зав. кафедри  Казакова Н.Ф.

Одеса, 2021

МЕТОДИЧНІ ВКАЗІВКИ  
до виконання лабораторних робіт  
з дисципліни **МОДЕЛІ І МЕТОДИ НЕЧІТКОЇ ЛОГІКИ**  
для студентів III курсу  
Рівень вищої освіти – «Бакалавр»  
Спеціальність – 122 – «Комп'ютерні науки»

Одеса, 2021

Методичні вказівки до виконання лабораторних робіт з дисципліни «Моделі і методи нечіткої логіки» для студентів 3-го курсу рівня вищої освіти «Бакалавр», за спеціальністю 122 – «Комп'ютерні науки»

Укладачі:

Вохменцева Т.Б., ст.викладач кафедри інформаційних технологій

## ЗМІСТ

ПЕРЕДМОВА.....	5
1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ДИСЦИПЛІНУ.....	6
ЛАБОРАТОРНА РОБОТА №1	
ОСНОВИ ПРОГРАМУВАННЯ В СИСТЕМІ МАТЛАБ.....	8
ЛАБОРАТОРНА РОБОТА №2	
ДОСЛІДЖЕННЯ НЕЧІТКИХ ОПЕРАЦІЙ ПРИ ПОБУДОВІ ФУНКЦІЙ НАЛЕЖНОСТІ.....	20
ЛАБОРАТОРНА РОБОТА №3	
ДОСЛІДЖЕННЯ СПОСОБІВ ФОРМУВАННЯ НЕЧІТКИХ МНОЖИН ТА ОПЕРАЦІЙ НАД НИМИ В FUZZY LOGIC TOOLBOX.....	31
ЛАБОРАТОРНА РОБОТА №4	
ПРОЕКТУВАННЯ СИСТЕМИ ТИПУ МАМДАНІ НА ПРИКЛАДІ ПОБУДОВИ НЕЧІТКОЇ АПРОКСИМУЮЧОЇ СИСТЕМИ .....	45
ЛАБОРАТОРНА РОБОТА №5	
ПРОЕКТУВАННЯ СИСТЕМИ ТИПУ СУГЕНО НА ПРИКЛАДІ ПОБУДОВИ НЕЧІТКОЇ АПРОКСИМУЮЧОЇ СИСТЕМИ .....	58
ЛАБОРАТОРНА РОБОТА №6	
ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ НА ОСНОВІ НЕЧІТКИХ ЗНАНЬ .....	66
ЛІТЕРАТУРА .....	82

## ПЕРЕДМОВА

Дисципліна «Моделі і методи нечіткої логіки» викладається для рівня вищої освіти – «Бакалавр» спеціальності 122 – «Комп'ютерні науки» входить до складу вибіркової частини навчального плану підготовки бакалаврів. Викладається відповідно до робочого навчального плану підготовки бакалаврів.

Мета методичних вказівок – забезпечити отримання студентами теоретичних знань і практичних навичок щодо використання новітніх інформаційні технологій для створення студентами математичних моделей процесів і явищ з використанням нечіткої логіки. Знайомство з моделями управління на базі нейронечітких систем, вивчення методів формалізації процесів і явищ в понятійному апараті нечіткої логіки.

В процесі навчання основна увага направлена на використання програмних засобів для розробки моделей нечіткої логіки.

Ці методичні вказівки містять теоретичні відомості, приклади виконання лабораторних робіт з дисципліни «Моделі і методи нечіткої логіки» та завдання. В методичних вказівках розглядаються питання, які відповідають навчальній програмі дисципліни.

# 1 ЗАГАЛЬНІ ВІДОМОСТІ ПРО ДИСЦИПЛІНУ

## 1.1 Мета дисципліни та її місце у навчальному процесі

Метою викладання дисципліни є формування навичок та вмій створення студентами математичних моделей процесів і явищ з використанням нечіткої логіки, знайомство з моделями управління на базі нейронечітких систем, вивчення методів формалізації процесів і явищ в понятійному апараті нечіткої логіки.

У результаті вивчення дисципліни студенти повинні **знати**:

основні визначення нечіткої логіки; області застосування нечітких множин і логіки; програмні засоби для моделювання нечітких множин; інструментальні інтегровані програмні середовища розробників для застосування моделей нечітких множин; приклади моделювання нечіткої логіки і мову нечітких формальних методів вирішення прикладних завдань.

### ***вміти:***

застосовувати програмні засоби розробки моделей нечіткої логіки, використовувати інструментальні функції інтегрованих програмних середовищ розробників нечіткої логіки; користуватися апаратними засобами моделювання нечітких множин; застосовувати програми нечіткої логіки для вирішення прикладних завдань.

По кожній лабораторній роботі студент повинен скласти звіт, якій містить в собі:

- Назву роботи. Мету.
- Умову завдання згідно варіанту.
- Хід виконання роботи.
- Відповіді на контрольні питання.

Оформлений звіт захищається студентом усно.

Види контролю поточних знань – усне опитування під час лекцій та лабораторних занять, контрольні роботи, модульний контроль, залік. Вид контролю залишкових знань – тестові завдання.

Оцінювання лабораторних робіт:

За кожну лабораторну роботу встановлена максимальна оцінка – 10 балів.

Контроль по кожній лабораторній роботі проводиться в формі:

- *усного опитування* при підготовці до кожної лабораторної роботи з метою допуску до її виконання (кількість запитань – до 5, максимальна кількість балів – 5),

- *захисту результатів* лабораторної роботи наведених у звіті до лабораторної роботи (кількість запитань залежить від ходу виконання студентом роботи і якості

звіту, максимальна кількість балів – 5).

Для кожної лабораторної роботи, якщо студент за *усне опитування* одержав 2 і менше балів він не допускається до виконання роботи, а якщо більше – допускається.

Для кожної лабораторної роботи при *захисті результатів* студент може одержати від 1 до 5 балів.

Підсумковою оцінкою за кожну лабораторну роботу буде сума балів за *усне опитування* і *захист результатів*.

Загальний обсяг навчального часу для денної форми навчання – 120 годин: лекцій – 30 годин, лабораторних занять – 30 годин, самостійна робота – 60 годин.

#### Правила техніки безпеки та охорона праці

Згідно з «Правилами техніки безпеки в лабораторіях інформатики» студентам забороняється:

- з'являтися та знаходитись приміщенні в нетверезому стані;
- ставити поруч з клавіатурою ємності з рідиною;
- перебувати в приміщенні в верхній одежі та завалювати нею робочі столи та стільці;
- працювати в лабораторії більше 6-ти годин на день (для вагітних жінок – більше 4-х годин);
- за власною ініціативою змінювати закріплені за ними робочі місця та знаходитись в приміщенні під час роботи іншої учбової групи;
- самостійно виконувати вмикання електроживлення лабораторії та заміну складових частин ПК, що вийшли із ладу.

У випадку виявлення несправностей обчислювальної техніки студент повинен сповістити про це викладача чи будь-кого з навчально-допоміжного персоналу лабораторії.

## ОСНОВИ ПРОГРАМУВАННЯ В СИСТЕМІ МАТЛАБ

**Мета роботи:** Засвоїти принципи роботи та програмування в системі Матлаб

### Теоретична частина

Після запуску системи MatLAB на екрані з'явиться вікно, показане на рис.1. Представлене вікно містить три області: Command Window, призначене для введення команд і виведення результатів; Command History, що містить історію всіх виконаних в Command Window команд і Workspace, де відображаються всі створені в робочій області змінні.

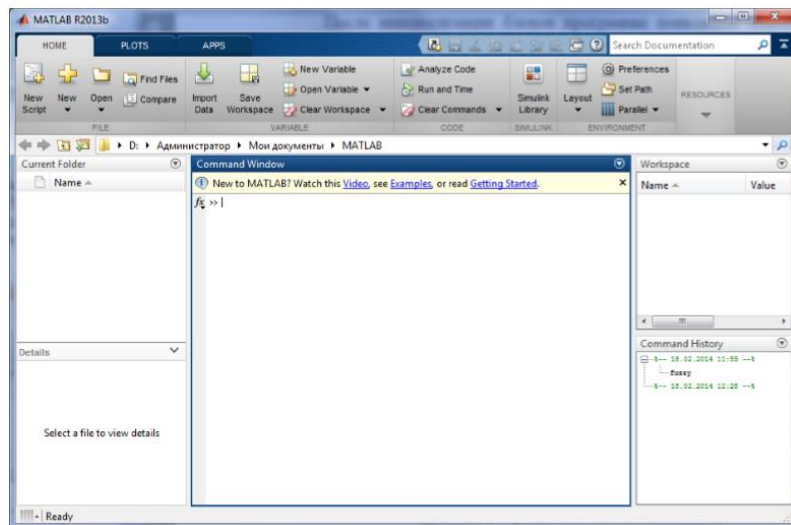


Рисунок 1 – Командне вікно системи MatLAB

Знак «>>» показує готовність системи до виконання введених команд.

### Умовний оператор if

Для того, щоб мати можливість реалізовувати логіку в програмі використовуються умовні оператори. Синтаксис умовного оператора має вигляд:

```
if <вираз1>
<оператори1>           % виконуються, якщо істинний вираз1
elseif <вираз2>
<оператори2>           % виконуються, якщо істинний вираз2
...
elseif <виразN>
<операториN>           % виконуються, якщо істинний виразN
end
```

### Приклад 1

```
function my_sign
x = 5;
if x > 0
    disp(1);           % виконується, якщо x > 0
elseif x < 0
```



```

disp(-1);      % виконується, якщо x < 0
else
disp(0);      % виконується, якщо x = 0
end

```

Для реалізації складених умов в MatLab використовуються логічні оператори:

& - логічне І  
| - логічне АБО  
~ - логічне НЕ

### Приклад 2.

```

function my_if
x = 1;
if (x >= -5 & x <= 5) & (x < 0 | x > 1)
disp('x належить [-5, 5], але не входить в [0, 1]');
else
disp('x або не входить в [-5, 5] або в [0, 1]');
end

```

Пріоритет логічних операцій наступний: НЕ (~); І (&); АБО (|).

### Умовний оператор switch

У деяких задачах програмування потрібно виконувати перевірку на рівність деякої змінної константним значенням. Для вирішення таких завдань зручніше користуватися умовним оператором switch, який повинен виглядати так:

```

switch expr
case case_expr,
<оператори1>
case {case_expr1, case_expr2, case_expr3,...}
<оператори2>
...
otherwise,
<оператори>
end

```

Тут expr - змінна, значення якої перевіряється на рівність тих чи інших констант; case\_expr - константи, з яким порівнюється значення змінної; otherwise - ключове слово, для виконання операторів, при всіх помилкових умовах.

Наведемо приклад роботи даного оператора для перетворення малих літер латинського алфавіту в заголовні.

### Приклад 3.

```

function upper_symbol
ch='c';
switch ch
case 'a', ch='A';
case 'b', ch='B';
case 'c', ch='C';

```

```

    case 'd', ch='D';
    case 'e', ch='E';
    ...
    case 'z', ch='Z';
end
disp(ch);

```

У даній програмі задається символічна змінна *ch* зі значенням *s*. Потім, за допомогою оператора `switch` перевіряється її значення з усіма можливими малими буквами латинського алфавіту від *a* до *z*. Як тільки одна з умов спрацювала, оператор `switch` завершує свою роботу і виконання програми переходить на функцію `disp()`, яка відображає значення змінної *ch* на екрані.

## Оператор циклу `while`

Мова програмування `MatLab` має два оператора циклу: `while` і `for`. З їх допомогою, наприклад, виконується програмування рекурентних алгоритмів, підрахунку суми ряду, перебору елементів масиву і багато іншого. У найпростішому випадку цикл в програмі організовується за допомогою оператора `while`, який повинен виглядати так:

```

while <умова>
    <оператори>
end

```

Тут `<умова>` означає умовний вираз подібний до того, який застосовується в операторі `if`, і цикл `while` працює до тих пір, поки ця умова істинна. Слід звернути увагу на те, що якщо умова буде помилковою до початку виконання циклу, то оператори, що входять в цикл, не будуть виконані жодного разу.

### Приклад 4

$$S = \sum_{i=1}^{20} i$$

Обчислити суму ряду:

```

function sum_i
S = 0; % початкове значення суми
i=1; % лічильник суми
while i <= 20 % цикл (працює доки i <= 20)
    S=S+i; % підраховується сума
    i=i+1; % лічильник збільшується на 1
end % кінець циклу
disp(S); % відображення результату сумування на екрані

```

Роботу будь-якого оператора циклу, в тому числі і `while`, можна примусово завершити за допомогою оператора `break`.

### Приклад 5.

```
function sum_i
S = 0; % початкове значення суми
i=1; % лічильник
while i <= 20 % цикл (працює доки i<=20 )
    S=S+i; % підраховується сума
    i=i+1; % лічильник збільшується на 1
    if S > 20 % якщо S > 20,
        break; % то цикл завершується
    end
end % кінець циклу
disp(S); % відображення результату суми на екрані
```

В даному прикладі друга умова завершення циклу, коли  $S$  буде більше 20, записано в самому циклі і за допомогою оператора `break` здійснюється вихід з циклу на функцію `disp()`, що стоїть відразу після циклу `while`.

Другий оператор управління виконанням циклу `continue` дозволяє пропускати виконання фрагмента програми, що стоїть після нього. Наприклад, потрібно підрахувати суму елементів масиву  $a = [1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]$ , виключаючи елемент з індексом 5.

### Приклад 6.

```
function sum_array
S = 0; % початкове значення суми
a = [1 2 3 4 5 6 7 8 9]; % масив
i=0; % лічильник індексів масиву
while i < length(a) % цикл (працює доки i менше
                    % довжини масиву a)
    i=i+1; % лічильник збільшується на 1
    if i == 5 % якщо індекс дорівнює 5
        continue; % то його не враховуємо
    end
    S=S+a(i); % підраховується сума
end % кінець циклу
disp(S); % відображення результату
```

## Оператор циклу for

Часто при організації циклу потрібно перебирати значення лічильника в заданому діапазоні значень і з заданим кроком зміни. Синтаксис оператора циклу `for` має наступний вигляд:

```
for <лічильник> = <початкове значення>:<крок>:<кінцеве значення>
    <оператори циклу>
end
```

Розглянемо роботу даного циклу на прикладі реалізації алгоритму пошуку максимального значення елемента у векторі.

### Приклад 7.

```
function search_max
a = [3 6 5 3 6 9 5 3 1 0];
m = a(1); % поточне максимальне значення
for i=1:length(a) % цикл від 1 до кінця вектора з
                % кроком 1 (за замовчуванням)
    if m < a(i) % якщо a(i) > m,
```

```

        m = a(i);           % то m = a(i)
    end
end
disp(m);                  % кінець циклу for

```

В даному прикладі цикл for задає лічильник  $i$  і змінює його значення від 1 до 10 з кроком 1. Зверніть увагу, що якщо величина кроку не вказується явно, то він береться за замовчуванням рівним 1.

## Робота з графіками в MatLab

MatLab надає багатий інструментарій по візуалізації даних. Використовуючи внутрішню мову, можна виводити двовимірні і тривимірні графіки в декартових і полярних координатах, виконувати відображення зображень з різною глибиною кольору і різними колірними картами, створювати просту анімацію результатів моделювання в процесі обчислень і багато іншого.

### Функція plot

#### Приклад 8. Побудова графіку функції $y=\sin(x)$

```

x = 0:0.01:pi;
y = sin(x);
plot(x,y);

```

В результаті отримаємо графік, представлений на рис.2.

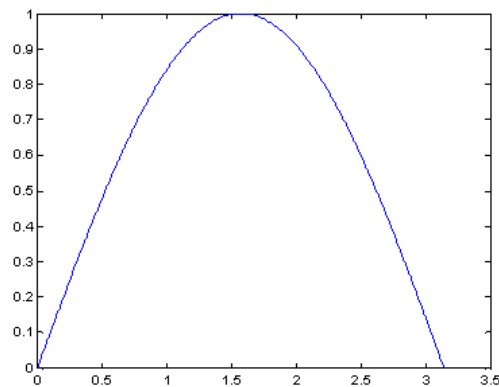


Рисунок 2 – Графік функції  $\sin(x)$

#### Приклад 9. Побудова графіків функцій $y=\sin(x)$ та $y=\cos(x)$

```

x = 0:0.01:pi;
y1 = sin(x);
y2 = cos(x);
plot(x,y1,x,y2);

```

Результат представлений на рис.3.

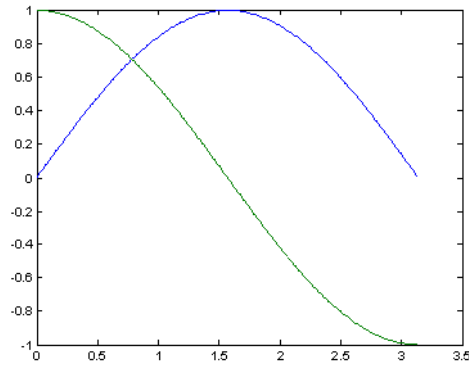


Рисунок 3 – Відображення двох графіків в одних координатних вісях.

**Приклад 10.** Побудова графіків функцій  $y=\sin(x)$  та  $y=\cos(x)$  в двох графічних вікнах

```
x1 = 0:0.01:2*pi;
y1 = sin(x1);
x2 = 0:0.01:pi;
y2 = cos(x2);
plot(x1, y1);           % відображення першого графіка
figure;                 % створення 2-го графічного вікна
plot(x2, y2);           % відображення 2-го графіка у 2-му вікні
```

**Приклад 11.** Побудова графіків функцій  $y=\sin(x)$  та  $y=\cos(x)$  в одному графічному вікні

```
x1 = 0:0.01:2*pi;
y1 = sin(x1);
x2 = 0:0.01:pi;
y2 = cos(x2);
figure(1);
subplot(2,1,1);        % поділяємо вікно на 2 рядки та 1 стовпець
plot(x1,y1);           % відображення першого графіка
subplot(2,1,2);        % будуємо 2-у координатну вісь
plot(x2,y2);           % відображення другого графіка
```

Результат показаний на рис. 4.

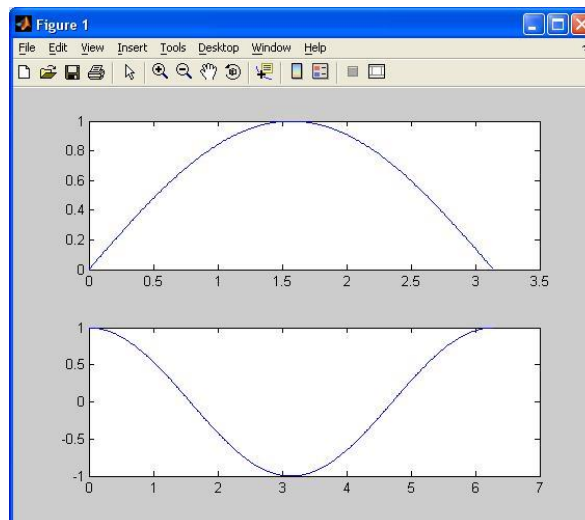


Рисунок 4 - Приклад роботи функції subplot

## Оформлення графіків

Функція `plot ()` дозволяє змінювати колір і тип лінії. Для цього, використовуються додаткові параметри, які записуються в такий спосіб:

```
plot (<x>, <y>, < 'колір лінії, тип лінії, маркер точок'>);
```

Зверніть увагу, що третій параметр записується в апострофах і має позначення, наведені в таблицях 1-3.

Таблиця 1 – Позначення кольору лінії графіку

Маркер	Колір лінії
c	блакитний
m	фіолетовий
y	жовтий
r	червоний
g	зелений
b	синій
w	білий
k	чорний

Таблиця 2 - Позначення типу лінії графіку

Маркер	Тип лінії
-	неперервна
--	штрихова
:	пунктирна
-.	штрих-пунктирна

Таблиця3 - Позначення типу точок графіку

Маркер	Тип точок
.	крапка
+	плюс
*	зірочка
o	коло
x	хрестик

Нижче показані приклади запису функції `plot()` з різним набором маркерів.

### Приклад 12.

```
x = 0:0.1:2*pi;  
y = sin(x);
```

```

subplot(2,2,1); plot(x,y,'r-');
subplot(2,2,2); plot(x,y,'r-',x,y,'ko');
subplot(2,2,3); plot(y,'b--');
subplot(2,2,4); plot(y,'b--+');

```

Результат роботи наведено на рис.5.

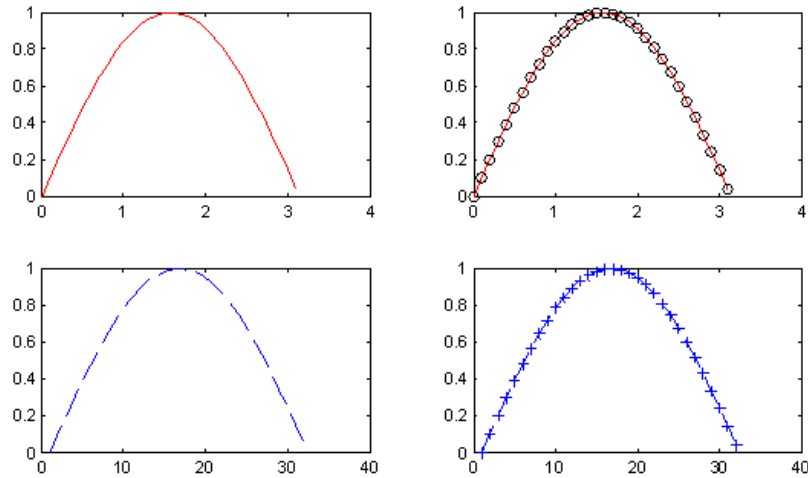


Рисунок 5 – Приклади відображення графіків з різним набором маркерів

Розглянемо можливість створення надписів графіків, осей та відображення сітки на графіку. Для цього використовуються функції мови MatLab, перелічені в табл.4

Таблиця 4 - Функції оформлення графіків

Назва	Опис
grid [on, off]	Вмикає/вимикає сітку на графіку
title('заголовок графіка')	Створює заголовок графіка
xlabel('підпис вісі Ох')	Створює підпис вісі Ох
ylabel('підпис вісі Оу')	Створює підпис вісі Оу
text(x,y,'текст')	Створює текстовий підпис в координатах (x,y).

### Приклад 13

```

x = 0:0.1:2*pi;
y = sin(x);
plot(x,y);
axis([0 2*pi -1 1]);
grid on;
title('The graphic of sin(x) function');
xlabel('The coordinate of Ох');

```

```
ylabel('The coordinate of Oy');
text(3.05,0.16,'\leftarrow sin(x)');
```

Результат наведено на рис.6

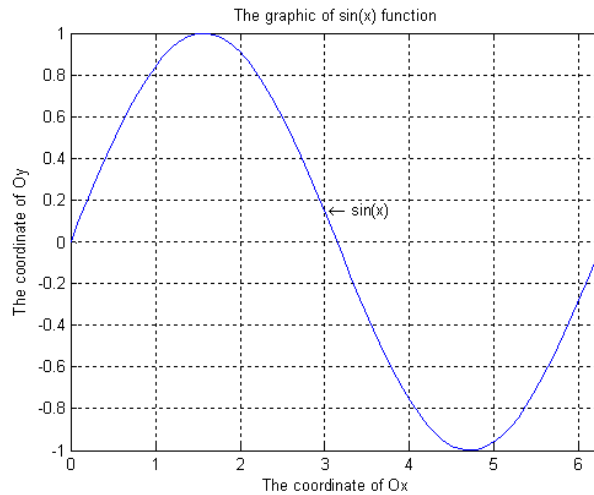


Рисунок 6 - Приклад оформлення графіка

Для візуалізації графіка в тривимірних координатних вісях, використовується функція **plot3 (X, Y, Z)**, яка в якості перших двох аргументів приймає матриці з координатами точок по вісях Ox і Oy відповідно, а в якості третьої аргументу передається матриця значень точок по вісі Oz.

#### Приклад 14.

Побудувати графік функції  $z(x, y) = \exp(-x^2 - y^2)$ , при  $x = -1, -0.9, \dots, 1$  и  $y = -2, -1.9, \dots, 2$ .

```
x=-1:0.1:1;           % координати точок по вісі Ox
y=-2:0.1:2;          % координати точок по вісі Oy
[X,Y]=meshgrid(x,y); % матриці координат точок по вісям Ox и Oy
Z=exp(-X.^2-Y.^2);
plot3(X,Y,Z);
```

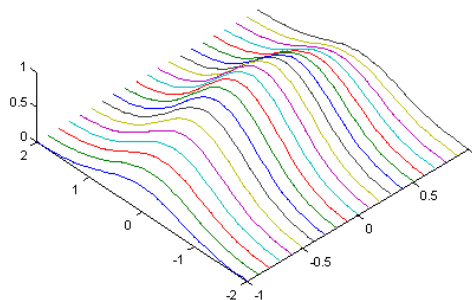


Рисунок 7 - Приклад відображення графіка за допомогою функції plot3()



З наведеного рисунку видно, що функція `plot3 ()` відображує графік у вигляді набору ліній, кожна з яких відповідає перетину графіка функції вздовж вісі  $Oy$ .

Більш наглядну візуалізацію можна отримати, використовуючи функцію `mesh (X, Y, Z)`. В результаті отримаємо наступний вигляд тривимірного графіка (рис.8).

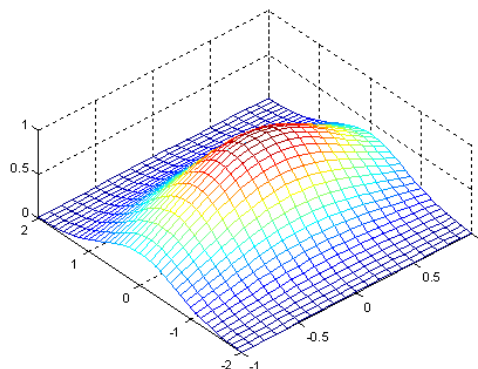


Рисунок 8 - Результат роботи функції `mesh()`

Для візуалізації неперервної поверхні в тривимірних осях використовується функція `surf (X, Y, Z)`.

В результаті отримаємо графік (рис.9).

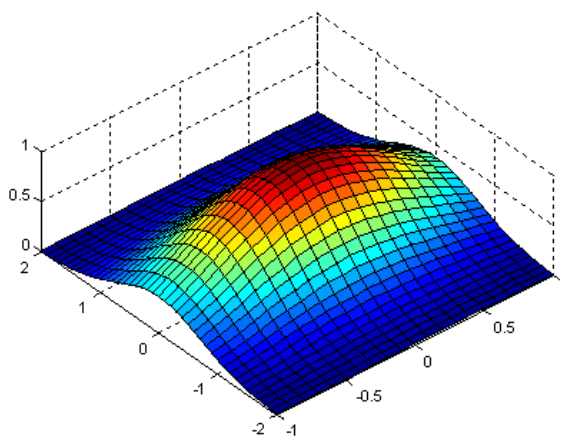


Рисунок 9 - Результат роботи функції `surf()`

Слід зазначити, що всі три функції `plot3 ()`, `mesh ()` і `surf ()` можуть бути використані і з одним аргументом  $Z$ , який інтерпретується як матриця зі значеннями точок по вісі  $Oz$ .

Для масштабування окремих ділянок тривимірних графіків, також як і у випадку з двовимірними графіками, використовується функція `axis ([xmin xmax ymin ymax zmin zmax])` зі зрозумілим набором параметрів.

Для оформлення тривимірних графіків можна користуватися описаними раніше функціями: `text`, `xlabel`, `ylabel`, `zlabel`, `title`, `grid [on/off]`, `subplot`.

## Варіанти завдань

### Завдання 1. Побудувати графік функції із заданими параметрами

Номер варіанту	Функція	Інтервал побудови	Крок аргументу	Параметри графіка		
				Колір	Тип точок	Тип лінії
1	$y = 2x^3 - 3x^2 + x + 5$	[-5;5]	0.1	червоний	крапка	неперервна
2	$y = \frac{x^3}{x^2 + 1}$	[0;10]	0.2	блакитний	плюс	штрихова
3	$y = \frac{x^2 + x}{x^2 - 3x + 2}$	[5;10]	0.25	зелений	зірочка	пунктирна
4	$y = (x^2 - 2x)e^x$	[-5;5]	0.4	чорний	коло	штрих-пунктирна
5	$y = \sin^3 x + \cos^3 x$	[0;10]	0.5	синій	хрестик	неперервна
6	$y = x^3 - 3x^2 - 9x + 11$	[5;10]	0.1	фіолетовий	крапка	штрихова
7	$y = \sqrt[3]{(x^2 - 1)}$	[1;10]	0.2	червоний	плюс	пунктирна
8	$y = -\frac{1}{4}(x^3 - 3x^2 + 4)$	[-4;4]	0.25	блакитний	зірочка	штрих-пунктирна
9	$y = \frac{x}{\sqrt{x^2 + x}}$	[1;5]	0.04	зелений	коло	неперервна
10	$y = \frac{e^x}{x}$	[1;6]	0.5	фіолетовий	хрестик	штрихова
11	$y = \frac{x^3}{x^2 - 4}$	[3;7]	0.1	чорний	крапка	пунктирна
12	$y = \frac{x^3}{2(x+5)^2}$	[0;6]	0.2	блакитний	плюс	штрих-пунктирна
13	$y = \frac{x^3}{x^2 - 1}$	[2;10]	0.25	червоний	зірочка	неперервна
14	$y = \frac{x^3 - 1}{4x^2}$	[3;10]	0.04	блакитний	коло	штрихова
15	$y = \ln \frac{x+1}{x+2}$	[5;10]	0.5	червоний	хрестик	пунктирна

### Завдання 2. Написати програму для виконання завдання згідно варіанту

1. Дано дійсне число  $x$ . Обчислити

$$x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots - \frac{x^n}{n!} + \dots$$

(прийняти  $x = 2.5$ ,  $n = 15$ )

2. Дано натуральне  $n$ , дійсне  $x$ . Обчислити  $\sin x + \sin^2 x + \dots + \sin^n(x)$   
(прийняти  $x = 2.5$ .)
3. Дано дійсне число  $x$ . Обчислити  $(x-2)(x-4)(x-8) \dots (x-64)$   
(прийняти  $x = 2.5$ )
4. Дано натуральне  $n$ , дійсне  $x$ . Обчислити  $\sin x + \sin(\sin x) + \dots + \sin \sin \dots \sin x$   
(прийняти  $x = 2.5, n = 15$ )
5. Обчислити суму  $1 + \frac{1}{3!} + \frac{1}{5!} + \frac{1}{7!} + \dots + \frac{1}{10!}$
6. Обчислити суму  $\frac{1}{13} + \frac{1}{23} + \frac{1}{33} + \dots + \frac{1}{503}$
7. Дано два цілих числа  $A$  і  $B$  ( $A < B$ ). Знайти суму всіх цілих чисел від  $A$  до  $B$  включно. (прийняти  $A = 38, B = 51$ )
8. Дано натуральні числа від 20 до 50. Надрукувати ті з них, які діляться на 3, але не діляться на 5.
9. Знайти добуток двозначних непарних чисел, кратних 13.
10. Дано два цілих числа  $A$  і  $B$  ( $A < B$ ). Знайти суму квадратів всіх цілих чисел від  $A$  до  $B$  включно. (прийняти  $A = 1, B = 40$ ).
11. Знайти суму двозначних чисел, кратних 12.
12. Знайти добуток двозначних чисел, кратних 12.
13. Обчислити суму перших 30 натуральних чисел.
14. Обчислити добуток перших 10 натуральних чисел.
15. Дано натуральні числа від 20 до 50. Обчислити суму тих з них, які діляться на 3, але не діляться на 5.

### Контрольні питання до захисту

1. Яка функція будує графік?
2. Які типи графіків може будувати система MatLab?
3. Для чого призначений subplot?
4. Наведіть синтаксис умовного оператора.
5. Коли використовуються оператори циклу?
6. Чим відрізняються цикли while і for?
7. Для чого призначений оператор break?
8. Для чого призначений оператор continue?
9. Які логічні операції використовуються при заданні умов?

ДОСЛІДЖЕННЯ НЕЧІТКИХ ОПЕРАЦІЙ ПРИ ПОБУДОВІ ФУНКЦІЙ  
НАЛЕЖНОСТІ

**Мета:** Вивчити операції, які виконуються над нечіткими множинами

**Теоретична частина**

**Нечіткі множини та нечітка логіка**

З точки зору характеристичної функції нечіткі множини є природне узагальнення звичайних множин, коли ми відмовляємося від бінарного характеру цієї функції та припускаємо, що вона може приймати будь-які значення на відрізьку  $[0, 1]$ . В теорії нечітких множин характеристична функція називається функцією приналежності, а її значення  $\mu_A(x)$  - ступінню приналежності елемента  $x$  нечіткій множині  $A$ .

Більш строго, нечіткою множиною  $A$  називається сукупність пар

$$A = \{ \langle x, \mu_A(x) \rangle \mid x \in U \}$$

де  $\mu_A(x)$  – функція приналежності, тобто

$$\mu_A(x) : U \rightarrow [0,1].$$

**Операції над нечіткими множинами**

Над нечіткими множинами можна робити різні операції, при цьому необхідно визначити їх так, щоб в окремому випадку, коли множина є чіткою, операції переходили в звичайні операції теорії множин, тобто операції над нечіткими множинами повинні узагальнювати відповідні операції над звичайними множинами. При цьому узагальнення може бути реалізовано різними способами, через що будь-якої операції над звичайними множинами може відповідати кілька операцій в теорії нечітких множин. Для визначення перетину і об'єднання нечітких множин найбільшою популярністю користуються такі три групи операцій:

1. Максимінні:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\},$$

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}.$$

2. Алгебраїчні:

$$\mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x)\mu_B(x),$$

$$\mu_{A \cap B}(x) = \mu_A(x)\mu_B(x).$$

3. Обмежені:

$$\mu_{A \cup B}(x) = \min\{1, \mu_A(x) + \mu_B(x)\},$$

$$\mu_{A \cap B}(x) = \max\{0, \mu_A(x) + \mu_B(x) - 1\}.$$

Доповнення нечіткої множини у всіх трьох випадках визначається однакою

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x).$$

### Кусково-лінійні функції приналежності

Найбільш характерним прикладом таких функцій є "трикутна" (рис. 10, а) і "трапецієвидна" (рис. 10, б) функції приналежності

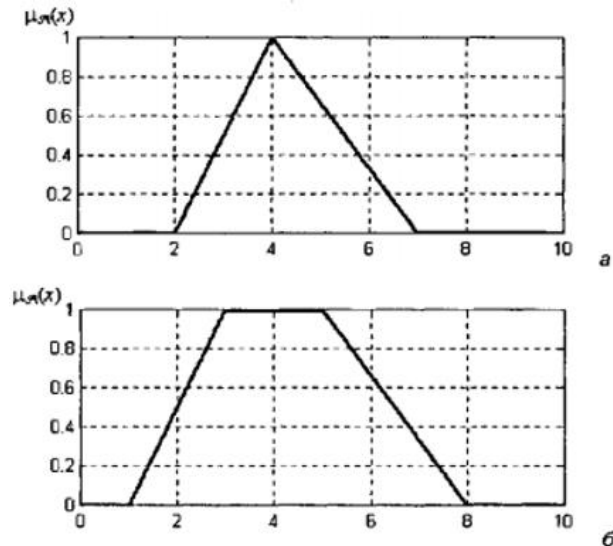


Рисунок 10 – Графіки функцій приналежності

Перша з цих функцій приналежності в загальному випадку може бути задана аналітично наступним виразом:

$$f_{\Delta}(x; , a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq c \leq x \\ 0, & c \leq x \end{cases},$$

де  $a, b, c$  - деякі числові параметри, які приймають довільні дійсні значення і впорядковані нерівністю:  $a \leq b \leq c$ . Як неважко помітити, параметри  $a$  і  $c$  характеризують основу трикутника, а параметр  $b$  – його вершину.

Трапецієвидна функція приналежності в загальному випадку може бути задана аналітично наступним виразом:

$$f_T(x; , a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases},$$

де  $a, b, c, d$ - деякі числові параметри, які приймають довільні дійсні значення і впорядковані відношенням:  $a \leq b \leq c \leq d$ .

Як окремі випадки Z- і S-образних кривих зручно розглядати так звану лінійну Z-подібну функцію (рис. 11, а) і лінійну S-подібну функцію (рис. 11, б). Перша з цих функцій в загальному випадку може бути задана аналітично наступним виразом:

$$f_1(x; a, b) = \begin{cases} 1, & x \leq a \\ \frac{b-x}{b-a}, & a < x < b \\ 0, & b \leq x \end{cases},$$

де  $a, b$  – деякі числові параметри, які приймають довільні дійсні значення і впорядковані відношенням:  $a < b$ . Друга з цих функцій в загальному випадку може бути задана аналітично наступним виразом:

$$f_2(x; a, b) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x < b \\ 1, & b \leq x \end{cases},$$

де  $a, b$  – деякі числові параметри, які приймають довільні дійсні значення і впорядковані відношенням:  $a < b$ .

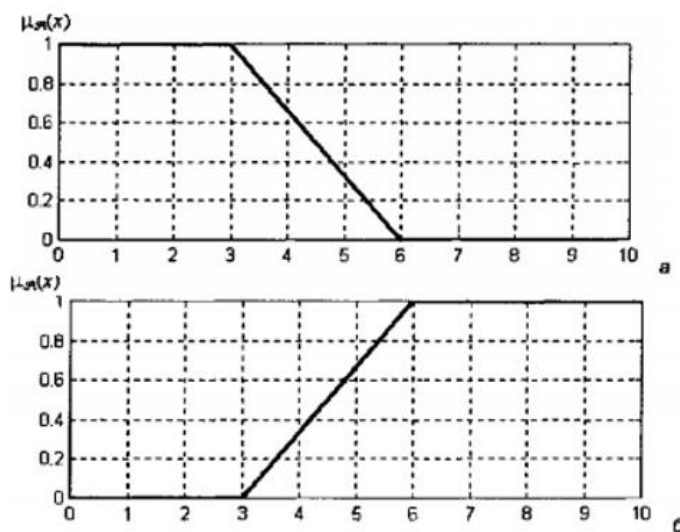


Рисунок 11 – Графіки лінійної Z-подібної функції (а) і лінійної S-подібної функції (б) приналежності для значень параметрів  $a = 3, b = 6$

## Варіанти завдань

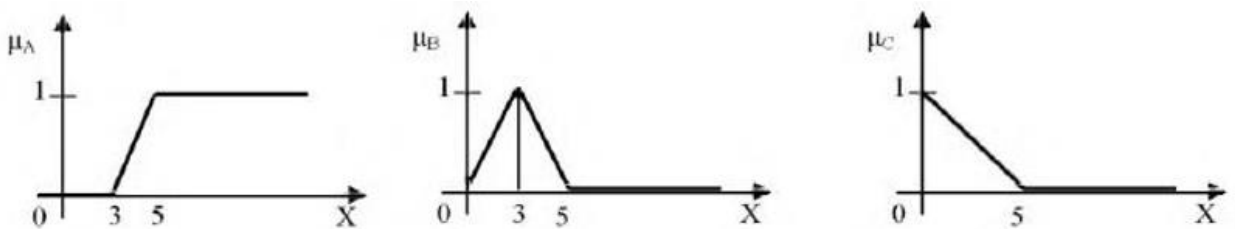
### Загальна постановка задачі

Дано 3 нечітких множини в формі функцій належності. Необхідно побудувати функцію приналежності заданої нечіткої множини  $D$  згідно Вашого варіанту і визначити ступінь приналежності одного елемента множини  $D$ , використовуючи один з підходів. При цьому необхідно розрахувати і побудувати графіки:

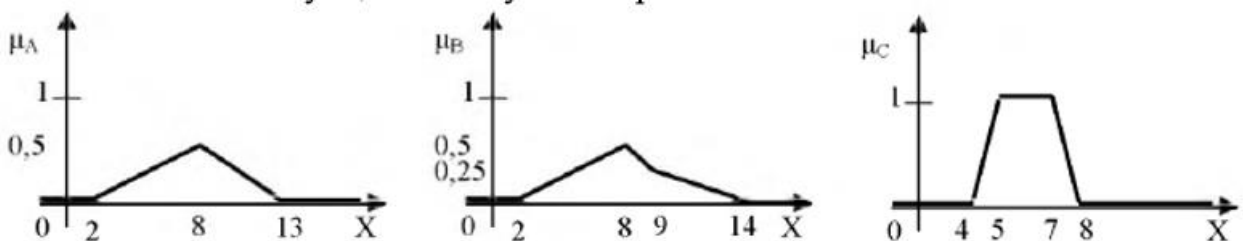
- 1)  $A \cap B$ ;  $A \cap C$ ;  $C \cap B$ ;  $A \cap C \cap B$ ;
- 2)  $A \cup B$ ;  $A \cup C$ ;  $C \cup B$ ;  $A \cup C \cup B$ ;
- 3)  $A \setminus B$ ;  $A \setminus C$ ;  $C \setminus B$ ;  $C \setminus A$ ;  $B \setminus A$ ;  $B \setminus C$ ;
- 4)  $\bar{A}$ ;  $\bar{C}$ ;  $\bar{B}$ ;  $\bar{D}$ ;
- 5)  $\sup_U \mu_A(x)$ ;  $\sup_U \mu_B(x)$ ;  $\sup_U \mu_C(x)$ ;  $\sup_U \mu_D(x)$ .

### Варіанти

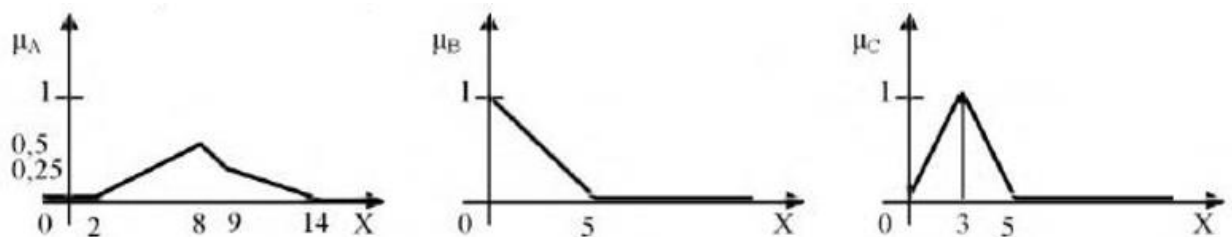
1. Дано 3 нечітких множини  $A, B, C$  (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap C$  і визначити ступінь приналежності одного елемента множини  $D$ , використовуючи максимінний спосіб.



2. Дано 3 нечітких множини  $A, B, C$  (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap C$  і визначити ступінь приналежності одного елемента множини  $D$ , використовуючи алгебраїчний спосіб.

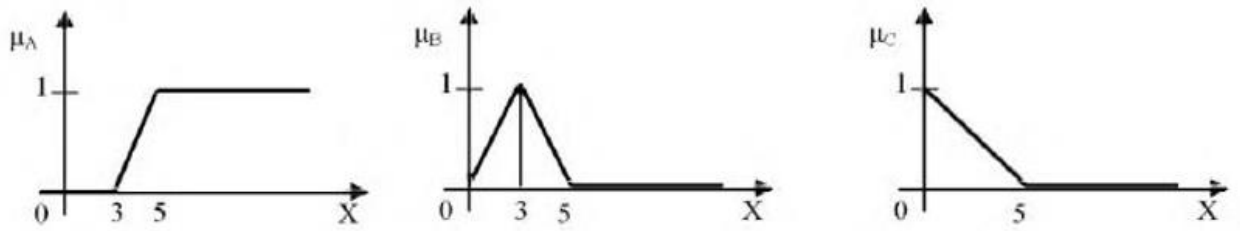


3. Дано 3 нечітких множини  $A, B, C$  (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap C$  і визначити ступінь приналежності одного елемента множини  $D$ , використовуючи спосіб обмежень.

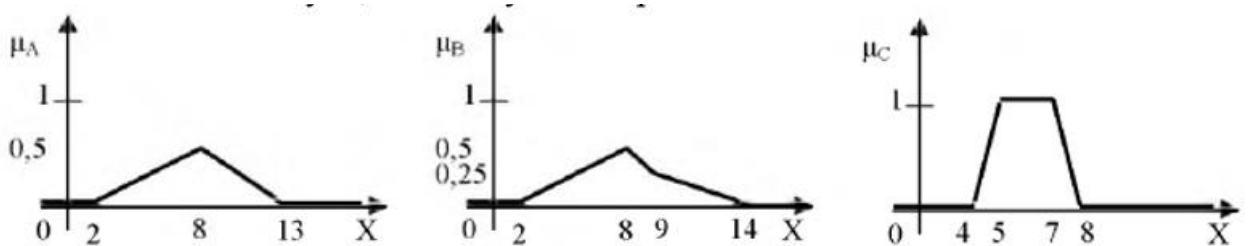


4. Дано 3 нечітких множини  $A, B, C$  (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup \bar{B} \cap C$  і визначити

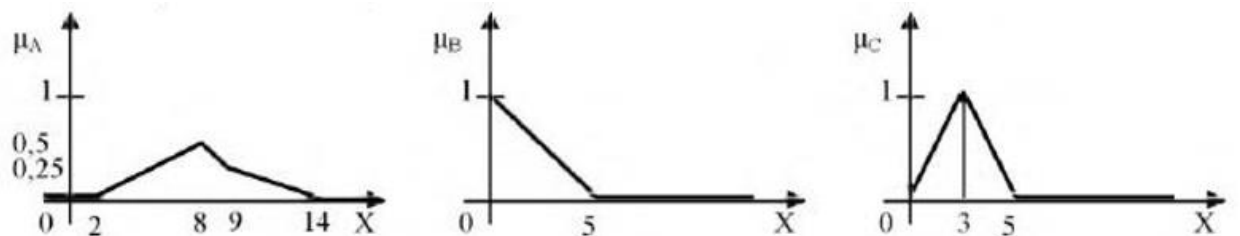
ступінь приналежності одного елемента множини D, використовуючи максимінний спосіб.



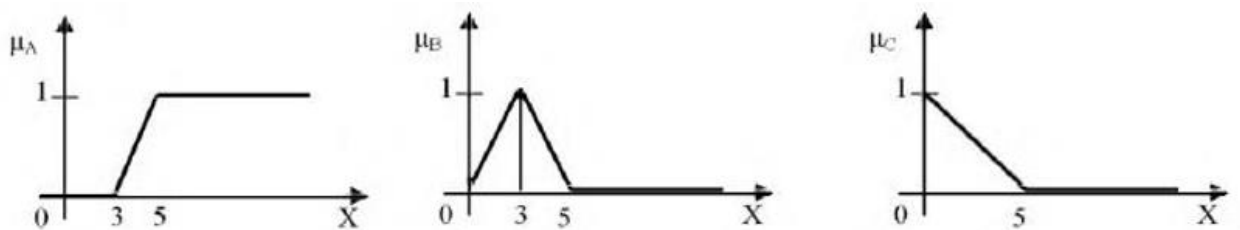
5. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup \bar{B} \cap C$  і визначити ступінь приналежності одного елемента множини D, використовуючи алгебраїчний спосіб.



6. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup \bar{B} \cap C$  і визначити ступінь приналежності одного елемента множини D, використовуючи спосіб обмежень.



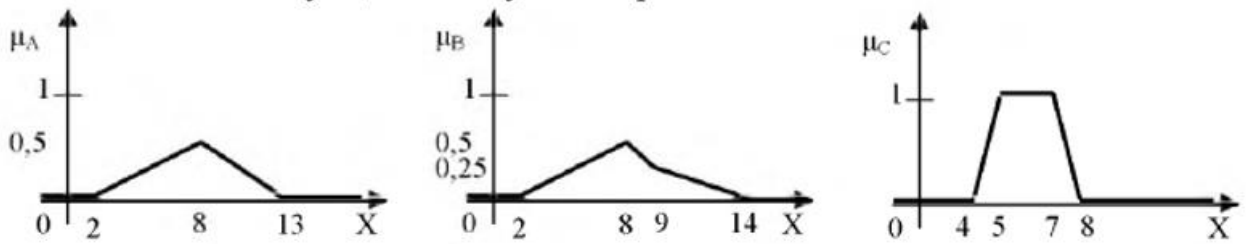
7. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap \bar{C}$  і визначити ступінь приналежності одного елемента множини D, використовуючи максимінний спосіб.



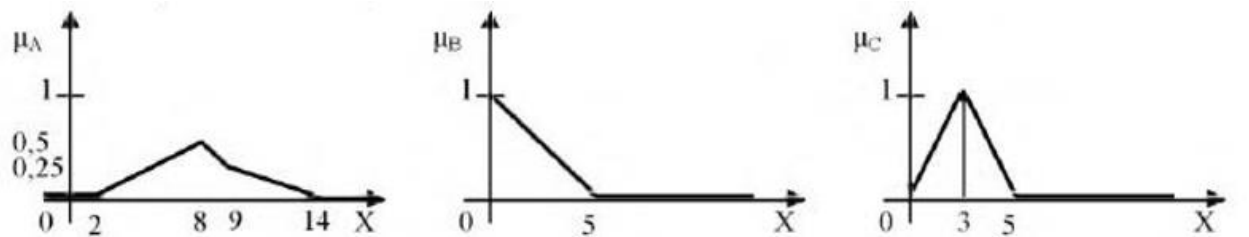
8. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap \bar{C}$  і визначити



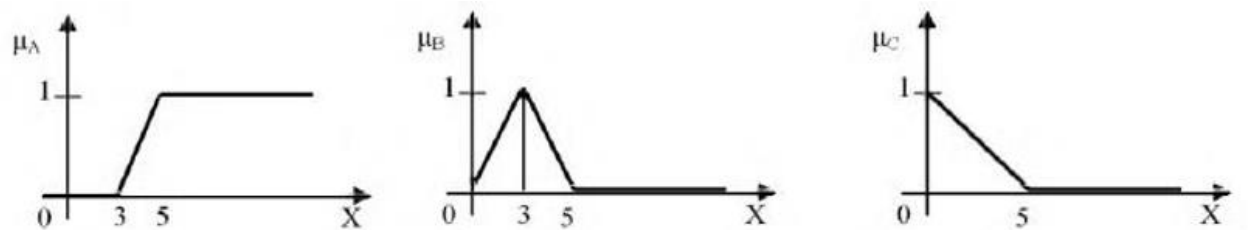
ступінь приналежності одного елемента множини D, використовуючи алгебраїчний спосіб.



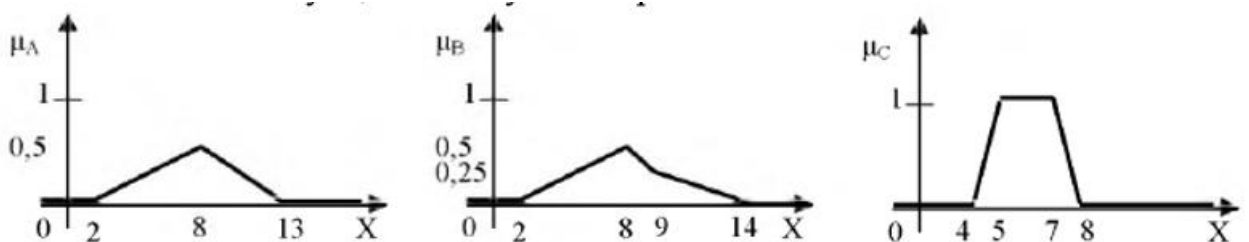
9. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=A \cup B \cap \bar{C}$  і визначити ступінь приналежності одного елемента множини D, використовуючи спосіб обмежень.



10. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=\bar{A} \cup B \cap C$  і визначити ступінь приналежності одного елемента множини D, використовуючи максимінний спосіб.

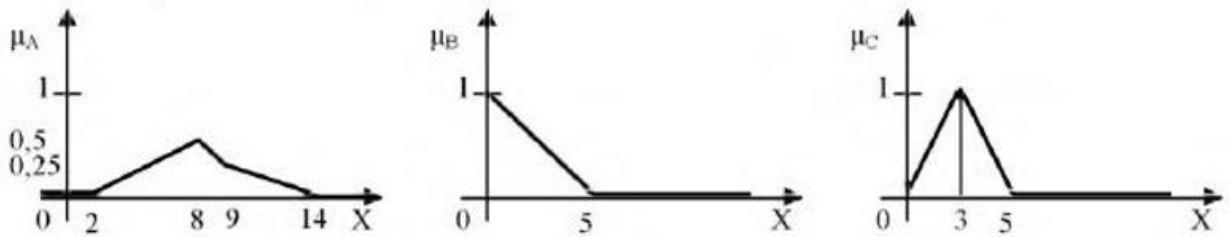


11. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=\bar{A} \cup B \cap C$  і визначити ступінь приналежності одного елемента множини D, використовуючи алгебраїчний спосіб.

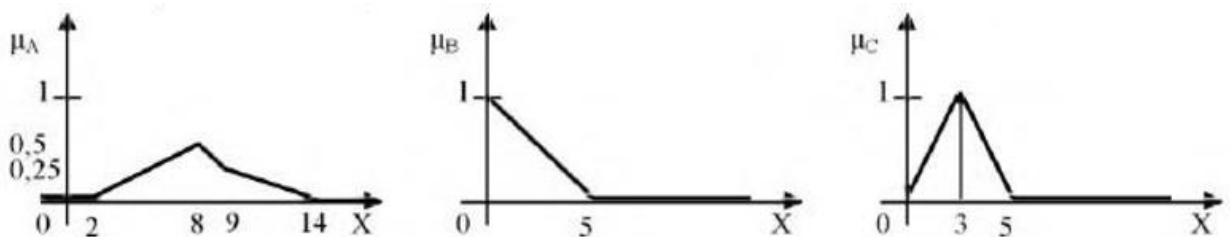


12. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=\bar{A} \cup B \cap C$  і визначити

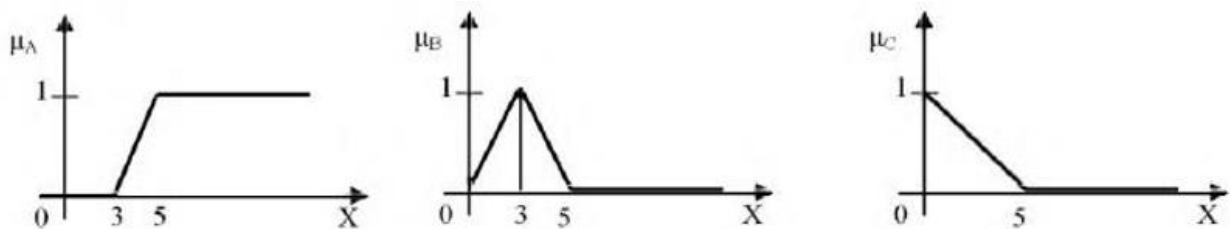
ступінь приналежності одного елемента множини D, використовуючи спосіб обмежень.



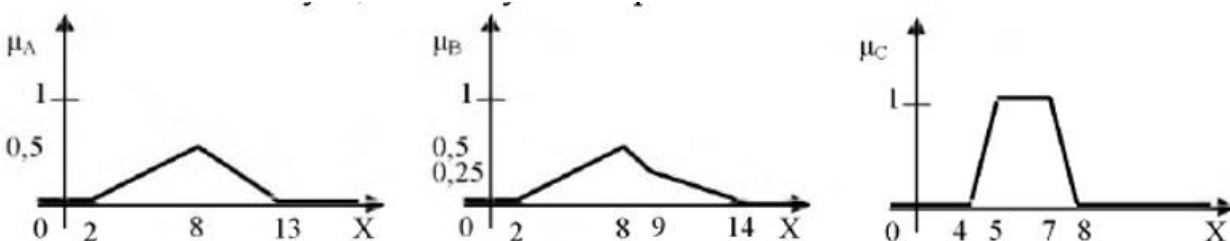
13. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=(A \cup \bar{B}) \cap C$  і визначити ступінь приналежності одного елемента множини D, використовуючи спосіб обмежень.



14. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=(A \cup B) \cap \bar{C}$  і визначити ступінь приналежності одного елемента множини D, використовуючи максимінний спосіб.



15. Дано 3 нечітких множини A, B, C (задані їх функції приналежності). Побудувати функцію приналежності нечіткої множини  $D=D=(A \cup B) \cap \bar{C}$  і визначити ступінь приналежності одного елемента множини D, використовуючи алгебраїчний спосіб.

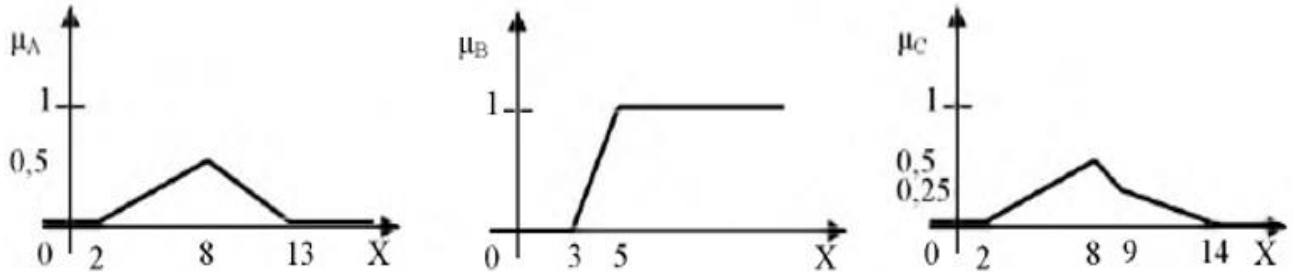


## Приклад виконання роботи

Дано 3 нечітких множини А, В, С (задані їх функції приналежності).  
Побудувати функцію приналежності нечіткої множини

$$D = \bar{A} \cap (A \cup C \cup B)$$

і визначити ступінь приналежності одного елемента множини D, використовуючи метод обмежень.



```
t = 0:0.01:20; % определяем число значений по оси абсцисс
a=2; b=8; c=13; %Параметры функции принадлежности M_A
a1=3; b1=5; %Параметры функции принадлежности M_B
a2=2; b2=8; c2=9; d2=14; %Параметры функции принадлежности M_C
% Рассчитываем значения оси ординат функций принадлежности M_A,
M_B, M_C
```

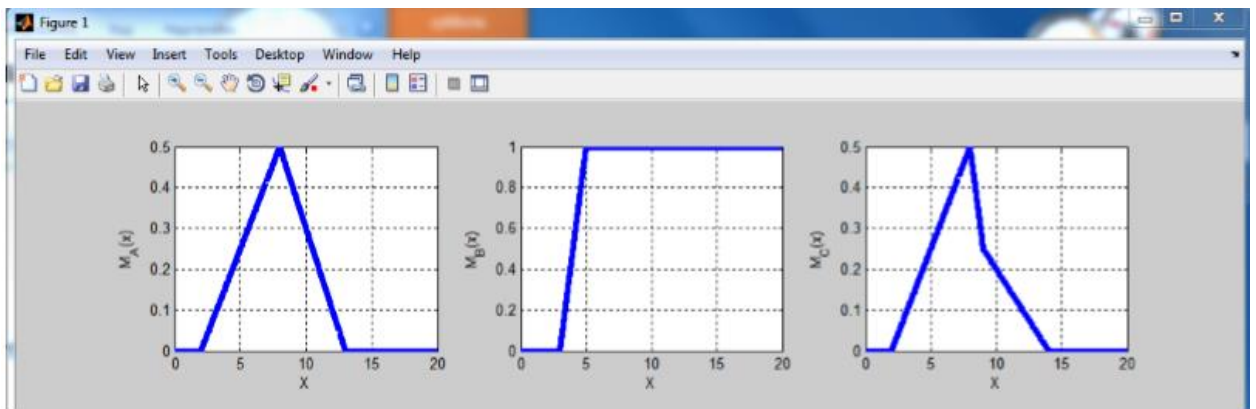
```
j=1;
for i = 0:0.01:20
    if i<=a M_A=0;
    elseif (i>a)&(i<=8) M_A(j)=0.5*(i-a)/(b-a);
    elseif (i>b)&(i<=13) M_A(j)=0.5*(c-i)/(c-b);
    elseif (i>c) M_A(j)=0; end;

    if i<=a1 M_B=0;
    elseif (i>a1)&(i<=b1) M_B(j)=(i-a1)/(b1-a1);
    elseif (i>b1) M_B(j)=1; end;

    if i<=a2 M_C=0;
    elseif (i>a2)&(i<=b2) M_C(j)=0.5*(i-a2)/(b2-a2);
    elseif (i>b2)&(i<=c2) M_C(j)=-0.25*i+2.5;
    elseif (i>c2)&(i<=d2) M_C(j)=0.25*(d2-i)/(d2-c2);
    elseif (i>d2) M_C(j)=0; end

    j=j+1;
end
```

```
% Строим график функции принадлежности M_A, M_B, M_C
subplot(2, 3, 1)
plot(t, M_A, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_A(x)');
subplot(2, 3, 2)
plot(t, M_B, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_B(x)');
subplot(2, 3, 3)
plot(t, M_C, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_C(x)');
```



### Опис процесу розв'язання

Для побудови функції приналежності нової множини необхідно:

- 1) Визначити послідовність виконання операцій у формулі.
- 2) Побудувати на окремих графіках проміжні множини, відповідно до визначеної послідовності дій. Звести проміжні множини на одному графіку і визначити підсумкову функцію приналежності.
- 3) Використовуючи певний в завданні метод, визначити аналітично ступінь приналежності елемента, що входить в ядро підсумкової множини.
- 4) Перевірити аналітичні обчислення по побудованій графіком функції приналежності.

### Розв'язання

- 1) Множина має вигляд

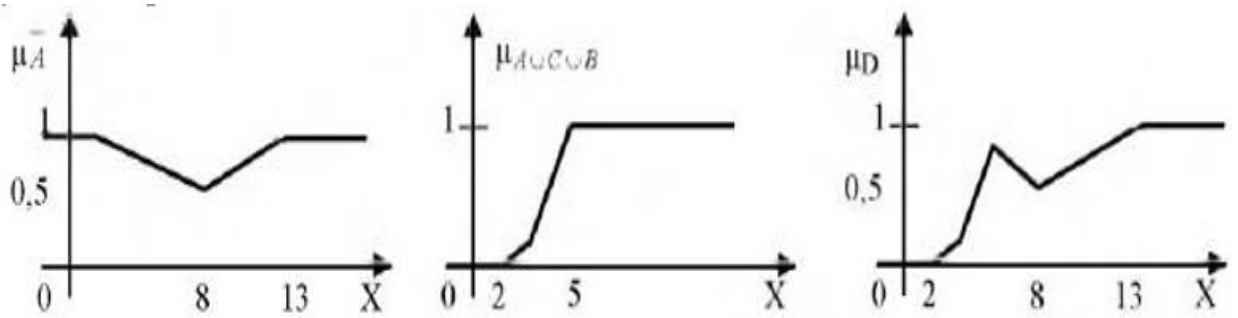
$$D = \bar{A} \cap (A \cup C \cup B)$$

Отже, послідовність операцій буде наступна:

$$\bar{A}, A \cup C \cup B, \bar{A} \cap (A \cup C \cup B).$$

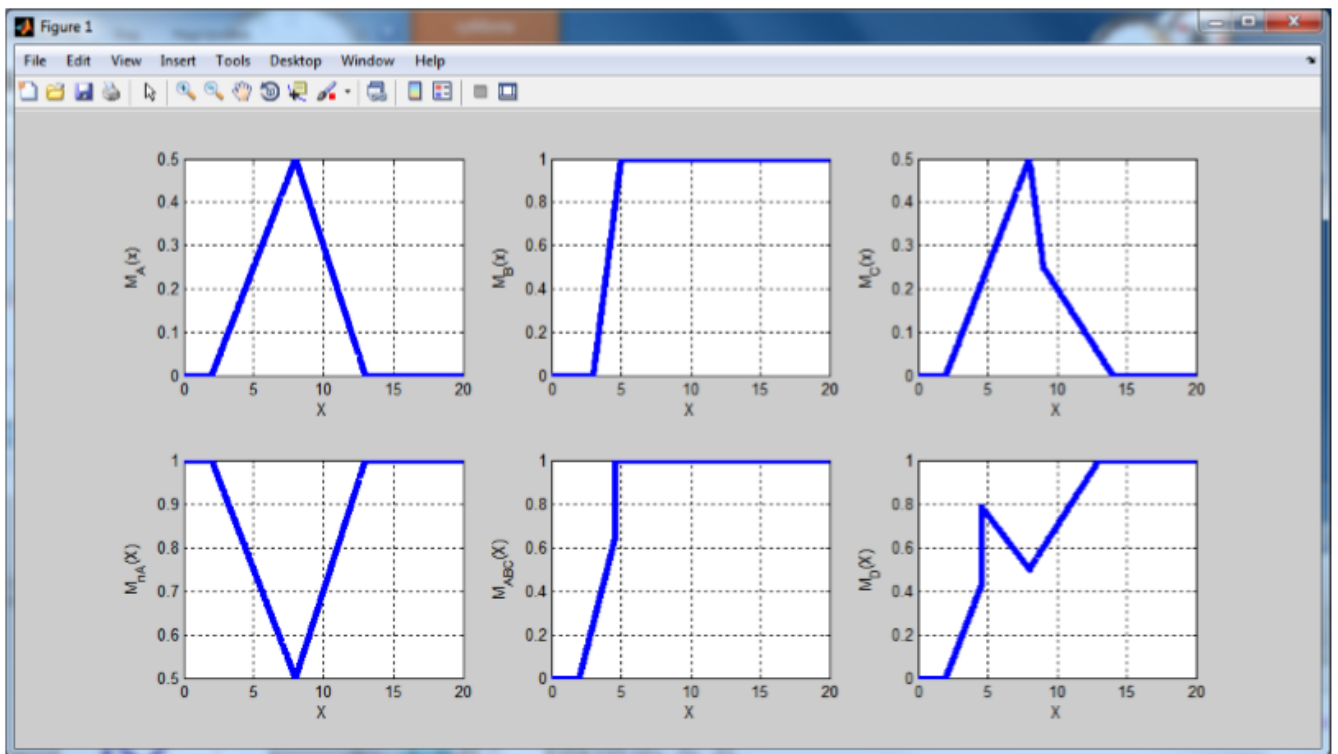
```
% Рассчитываем значения последовательности операций
j=1;
for i = 0:0.01:20
    M_n_A(j)=1-M_A(j); % Рассчитываем значение отрицания A
    (M_n_A)
    if (M_A(j)+M_B(j))<1 M_A_B(j)=M_A(j)+M_C(j); else
M_A_B(j)=1; end;% Рассчитываем значение объединения A и B
(M_A_B)
    if (M_A_B(j)+M_C(j))<1 M_A_B_C(j)=M_A_B(j)+M_C(j); else
M_A_B_C(j)=1; end;% Рассчитываем значение объединения (M_A_B_C)
    if (M_A_B_C(j)+M_n_A(j)-1)>0 M_D(j)=M_A_B_C(j)+M_n_A(j)-1;
else M_D(j)=0; end; % Рассчитываем значение M_D
    j=j+1;
end
```

- 2) Побудуємо відповідно до цієї послідовності операцій графіки функцій приналежності:



% Строим графики функций рассчитанных функций принадлежности

```
subplot(2, 3, 4)
plot(t, M_n_A, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_n_A(X)');
subplot(2, 3, 5)
plot(t, M_A_B_C, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_A_B_C(X)');
subplot(2, 3, 6)
plot(t, M_D, 'LineWidth', 4); grid on; xlabel('X');
ylabel('M_D(X)');
```



3) Носій множини D складається з елементів з інтервалу  $[2, 15]$ . Ядро множини D складається з елементів  $[13, 14, 15]$ . Виберемо елемент 8.

$$\mu_A(8) = 0,5;$$

$$\mu_B(8) = 1;$$

$$\mu_C(8) = 0,5;$$

$$\mu_{\bar{A}}(8) = 1 - \mu_A(8) = 1 - 0,5 = 0,5;$$

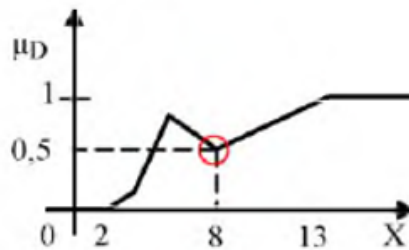
$$\mu_{\bar{C}}(8) = 1 - \mu_C(8) = 1 - 0,5 = 0,5;$$

$$\mu_{A \cup C}(8) = \min\{1, \mu_C(8) + \mu_A(8)\} = \min\{1, 1 + 1\} = 1;$$

$$\mu_{A \cup C \cup B}(8) = \min\{1, \mu_{A \cup C}(8) + \mu_B(8)\} = \min\{1, 1 + 1\} = 1;$$

$$\mu_{\bar{A} \cap (A \cup C \cup B)}(8) = \max\{0, \mu_{\bar{A}}(8) + \mu_{A \cup C \cup B}(8) - 1\} = \max\{0, 0,5 + 1 - 1\} = 0,5.$$

$$4) \mu_D(8) = 0,5.$$



### Контрольні питання до захисту

1. Яким чином задається нечітка множина?
2. Назвіть операції над нечіткими множинами?
3. Що таке функція приналежності?
4. Яким чином задається функція приналежності?
5. Чи може функція приналежності змінюватися в інтервалі  $[-1, +1]$ ?

ЛАБОРАТОРНА РОБОТА №3  
ДОСЛІДЖЕННЯ СПОСОБІВ ФОРМУВАННЯ НЕЧІТКИХ МНОЖИН ТА  
ОПЕРАЦІЙ НАД НИМИ В FUZZY LOGIC TOOLBOX

**Мета:** Вивчити методи побудови нечітких множин з використанням різних типів функцій належності. Ознайомитися з найбільш поширеними логічними операціями над нечіткими множинами.

**Теоретична частина**

**Вбудовані функції належності пакета Fuzzy Logic Toolbox**

Fuzzy Logic Toolbox включає 11 вбудованих функцій належності. Для зручності імена всіх вбудованих функцій належності закінчуються на **mf**. Виклик функції належності здійснюється наступним чином: **namemf(x, params)**,

де **namemf** – найменування функції належності;

**x** – вектор, для координат якого необхідно розрахувати значення функції належності;

**params** – вектор параметрів функції належності.

Вбудовані функції належності використовують такі основні функції:

1. Кусково-лінійні функції належності:

1.1. **trimf** – трикутна функція належності. Порядок параметрів: [a, b, c].

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases}$$

1.2. **trapmf** – трапецієвидна функція належності. Порядок параметрів: [a, b, c, d].

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases}$$

1.3. **smf** – s-подібна функція належності. Порядок параметрів: [a b].

$$\mu(x) = \begin{cases} 0, & x \leq a \\ \text{нелинейная аппроксимация}, & a < x < b \\ 1, & x \geq b \end{cases}$$

1.4. **zmf** – z-подібна функція належності. Порядок параметрів: [a b].

$$\mu(x) = \begin{cases} 1, & x \leq a \\ \text{нелинейная аппроксимация}, & a < x < b \\ 0, & x \geq b \end{cases}$$

2. Функції належності, побудовані на підставі гаусовського розподілу:

2.1. **gaussmf** – симетрична гаусовська функція належності. Порядок параметрів: [c b].

$$\mu(x) = e^{-\frac{(x-b)^2}{2c^2}}$$

2.2. **gauss2mf** – двостороння гаусовська функція належності. Порядок параметрів: [a1 c1 a2 c2].

$$\mu(x) = \begin{cases} \exp\left(\frac{(x-c_1)^2}{-2a_1^2}\right), & x < c_1 \\ 1, & c_1 \leq x \leq c_2 \\ \exp\left(\frac{(x-c_2)^2}{-2a_2^2}\right), & x > c_2 \end{cases}$$

якщо  $c_1 < c_2$ , то

$$\mu(x) = \begin{cases} \exp\left(\frac{(x-c_1)^2}{-2a_1^2}\right), & x < c_2 \\ \exp\left(\frac{(x-c_1)^2}{-2a_1^2}\right) * \exp\left(\frac{(x-c_2)^2}{-2a_2^2}\right) & c_2 \leq x \leq c_1 \\ \exp\left(\frac{(x-c_2)^2}{-2a_2^2}\right), & x > c_1 \end{cases}$$

якщо  $c_1 > c_2$ , то

3. Функції належності, побудовані на підставі сигмоїдної кривої:

3.1. **sigmf** – сигмоїдна функція належності. Порядок параметрів: [a c].

$$\mu(x) = \frac{1}{1 + e^{-a(x-c)}}$$

3.2. **psigmf** – добудок двох сигмоїдних функцій належності. Порядок параметрів: [a1 c1 a2 c2].

$$\mu(x) = \frac{1}{1 + e^{-a_1(x-c_1)}} \cdot \frac{1}{1 + e^{-a_2(x-c_2)}}$$

3.3. **dsigmf** – функція належності у вигляді різниці між двома сигмоїдними функціями. Порядок параметрів: [a1 c1 a2 c2].

$$\mu(x) = \frac{1}{1 + e^{-a_1(x-c_1)}} - \frac{1}{1 + e^{-a_2(x-c_2)}}$$

4. Функції належності, побудовані на підставі квадратичної і кубічної кривих:

4.1. **gbellmf** – узагальнена колокообразна функція належності. Порядок параметрів: [a b c].



$$\mu(x) = \frac{1}{1 + \left| \frac{x - c}{a} \right|^{2b}}$$

4.2. **pimf** – пі-подібна функція належності. Порядок параметрів: [a b c d], [a d] – носій нечіткої множини; [b c] – ядро нечіткої множини. Є добутком smf і zmf функцій.

Найпростіші функції належності трикутна (**trimf**) і трапецієвидна (**trapmf**) формується з використанням кусково-лінійної апроксимації. Трапецієвидна функція належності є узагальнення трикутної, вона дозволяє задавати ядро нечіткої множини у вигляді інтервалу. У разі трапецієподібної функції належності можлива наступна зручна інтерпретація: ядро нечіткої множини – оптимістична оцінка; носій нечіткої множини – песимістична оцінка. Дві функції належності – симетрична гаусовська (**gaussmf**) і двостороння гаусовська (**gauss2mf**) формується з використанням гаусовського розподілу. Функція **gaussmf** дозволяє задавати асиметричні функції належності. Узагальнена колоколообразна функція належності (**gbellmf**) за своєю формою схожа на гаусові. Ці функції належності часто використовуються в нечітких системах, так як на всій області визначення вони є гладкими і приймають ненульові значення. Функції приналежності **sigmf**, **dsigmf**, **psigmf** засновані на використанні сігмоїдної кривої. Ці функції дозволяють формувати функції належності, значення яких починаючи з деякого значення аргументу і до  $+\infty$  рівні 1. Такі функції зручні для завдання лінгвістичних термів типу «високий» або «низький». Поліноміальна апроксимація застосовується при формуванні функцій **zmf**, **pimf** і **smf**, графічні зображення яких схожі на функції **sigmf**, **dsigmf**, **psigmf**, відповідно.

На рис.12 наведені графічні зображення функцій належності, отримані за допомогою демонстраційного сценарію **mfdemo**. Як видно з рисунку, вбудовані функції належності дозволяють задавати різноманітні нечіткі множини.

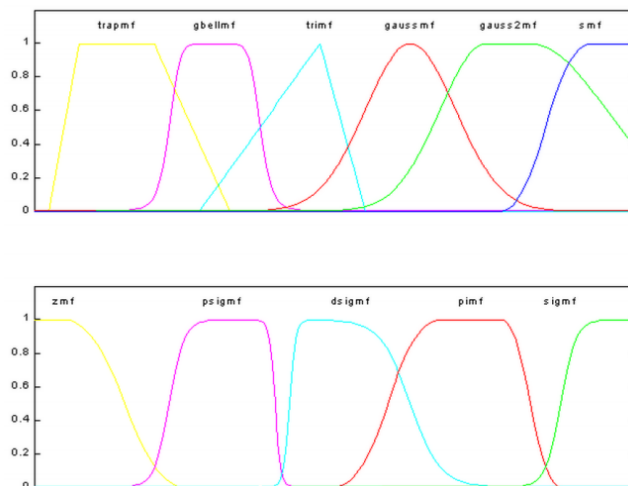


Рисунок 12 – Вбудовані функції належності

У Fuzzy Logic Toolbox передбачена можливість для користувача створення власної функції належності. Для цього необхідно створити m-функцію, яка містить два вхідних аргументу – вектор, для координат якого необхідно розрахувати значення функції належності і вектор параметрів функції належності. Вихідним аргументом функції повинен бути вектор ступенів належності. Нижче приведена m-функція, що реалізує дзвінообразну функцію належності (рис. 13):

$$\mu(x) = \frac{1}{1 + \left(\frac{x-b}{a}\right)^2}$$

```
x=0:0.1:5;
```

```
a=1;
```

```
b=2;
```

```
mu=1./(1+ ((x-b)/a).^2);
```

```
plot(x, mu);
```

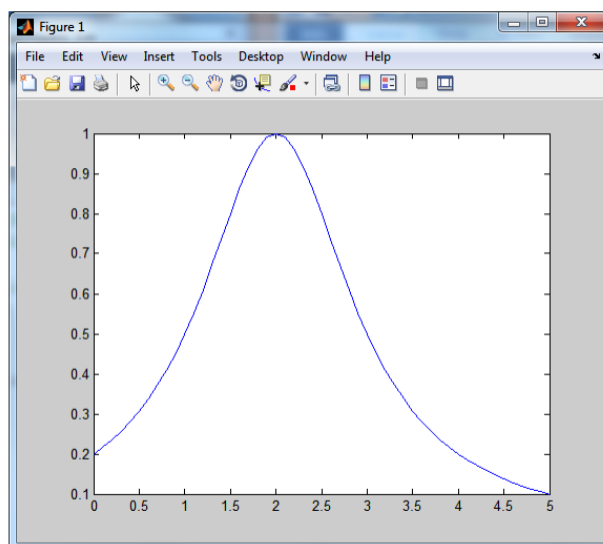


Рисунок 13 – Дзвінообразна функція належності

### Операції з нечіткими множинами

Виділяють три основні логічні операції з нечіткими множинами: кон'юнкцію, диз'юнкцію і логічне заперечення. У середовищі Matlab існує можливість визначати кон'юнктивні і диз'юнктивні оператори з точки зору мінімаксної і ймовірнісної інтерпретацій.

Розглянемо мінімаксне інтерпретацію логічних операторів, в якій кон'юнктивний оператор представляє знаходження мінімуму (Min), а диз'юнктивний – максимуму (max):

$$y = \min ([y1; y2]).$$

$$y = \max ([y1; y2]).$$

Параметри  $y_1$  і  $y_2$  представляють собою вихідні ФН. функція  $\min$  працює зі списком ФН. У Matlab список оформляється квадратними дужками, а елементи списку розділяються крапкою з комою.

Мінімаксна інтерпретація є найбільш поширеною при побудові нечітких систем. Проте на практиці досить часто використовується альтернативна імовірнісна інтерпретація кон'юнктивних і диз'юнктивних операторів. Matlab містить відповідні функції.

В рамках даної інтерпретації кон'юнктивний оператор є оператором обчислення алгебраїчного добутку –  $\text{prod}$ , а диз'юнктивний оператор – оператор обчислення алгебраїчної суми –  $\text{probor}$ :

$$y = \text{prod} ([y2; y2])$$

$$y = \text{probor} ([y2; y2])$$

Параметри  $y_1$  і  $y_2$  представляють собою вихідні ФН.

Доповнення нечіткої множини є не що інше, як математичне уявлення вербального вираження «НЕ А», де А – нечітка множина, яка описує деяке розмите судження. Опис функції доповнення:

$$y = 1 - y^*, \text{ де } y^* - \text{вихідна ФП.}$$

## Варіанти завдань

### Загальна постановка задачі

Дано 3 нечітких множин у формі функцій належності. Необхідно побудувати функцію належності заданої нечіткої множини D згідно Вашого варіанту і визначити ступінь належності одного елемента множини D, використовуючи один з підходів.

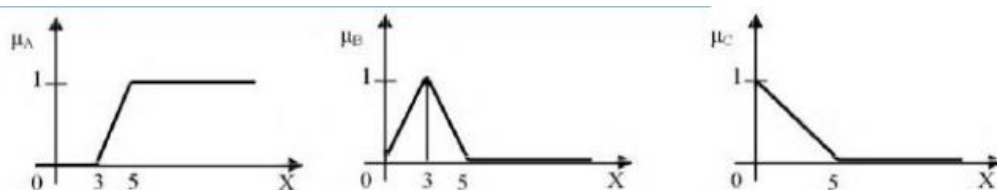
При цьому необхідно розрахувати і побудувати графіки:

- 1)  $A \cap B$ ;  $A \cap C$ ;  $C \cap B$ ;  $A \cap C \cap B$ ;
- 2)  $A \cup B$ ;  $A \cup C$ ;  $C \cup B$ ;  $A \cup C \cup B$ ;
- 3)  $A \setminus B$ ;  $A \setminus C$ ;  $C \setminus B$ ;  $C \setminus A$ ;  $B \setminus A$ ;  $B \setminus C$ ;
- 4)  $\bar{A}$ ;  $\bar{C}$ ;  $\bar{B}$ ;  $\bar{D}$ .

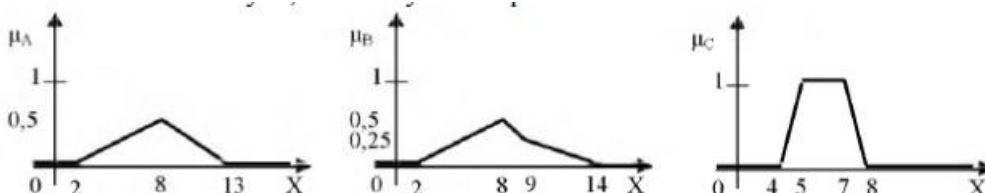
### Варіанти

1. Дано 3 нечітких множини А, В, С (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap C$  і визначити ступінь належності одного елемента множини D, використовуючи максимінний спосіб. При цьому функції належності нечітких множин А, В, С задати за

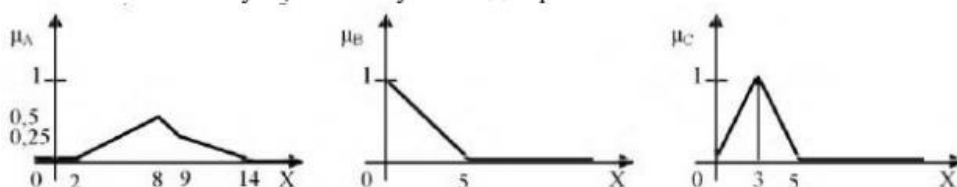
допомогою вбудованих функцій fuzzy logic:  $A = \text{«створити дві функції і за допомогою логічних операцій отримати вихідну множину»}$ ;  $B = \text{gaussmf}$ ;  $C = \text{zmf}$ .



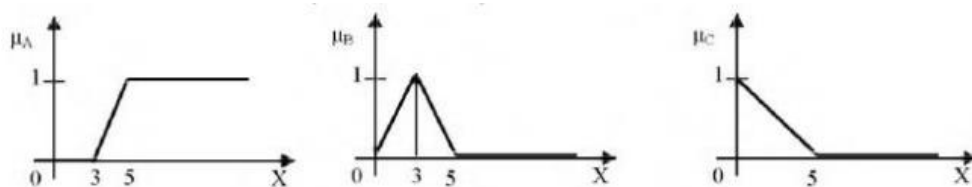
2. Дано 3 нечітких множини  $A, B, C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap C$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи алгебраїчний спосіб. При цьому функції належності нечітких множин  $A, B, C$  задати за допомогою вбудованих функцій fuzzy logic:  $A = \text{trimf}$ ;  $B = \text{«створити дві функції і за допомогою логічних операцій отримати вихідну множину»}$ ;  $C = \text{gbellmf}$ .



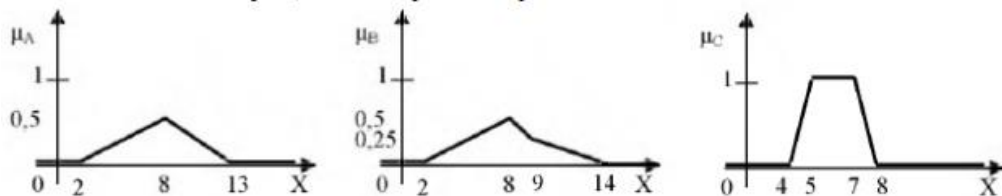
3. Дано 3 нечітких множини  $A, B, C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap C$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи метод обмежень. При цьому функції належності нечітких множин  $A, B, C$  задати за допомогою вбудованих функцій fuzzy logic:  $A = \text{«створити дві функції і за допомогою логічних операцій отримати вихідну множину»}$ ;  $B = \text{zmf}$ ;  $C = \text{gaussmf}$ .



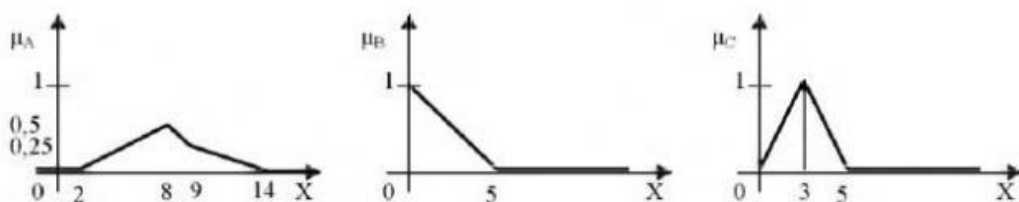
4. Дано 3 нечітких множини  $A, B, C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup \bar{B} \cap C$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи максимінний спосіб. При цьому функції належності нечітких множин  $A, B, C$  задати за допомогою вбудованих функцій fuzzy logic:  $A = \text{smf}$ ;  $B = \text{«створити дві функції і за допомогою логічних операцій отримати вихідну множину»}$ ;  $C = \text{zmf}$ .



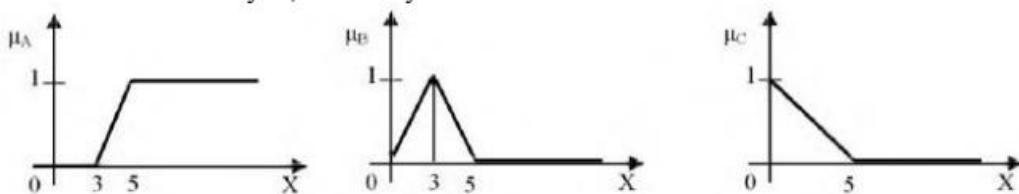
5. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup \bar{B} \cap C$  і визначити ступінь належності одного елемента множини D, використовуючи алгебраїчний спосіб. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A = trimf; B = «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; C = psigmf.



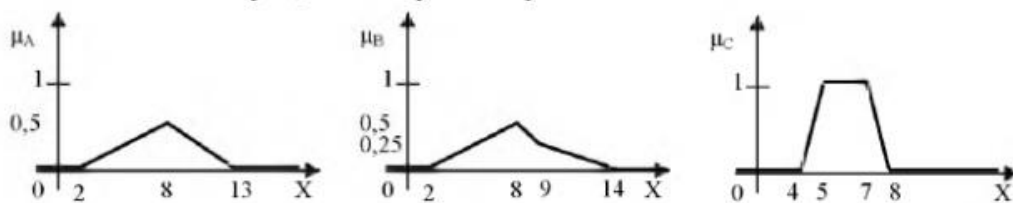
6. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup \bar{B} \cap C$  і визначити ступінь належності одного елемента множини D, використовуючи метод обмежень. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A = «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; B = zmf; C = trimf.



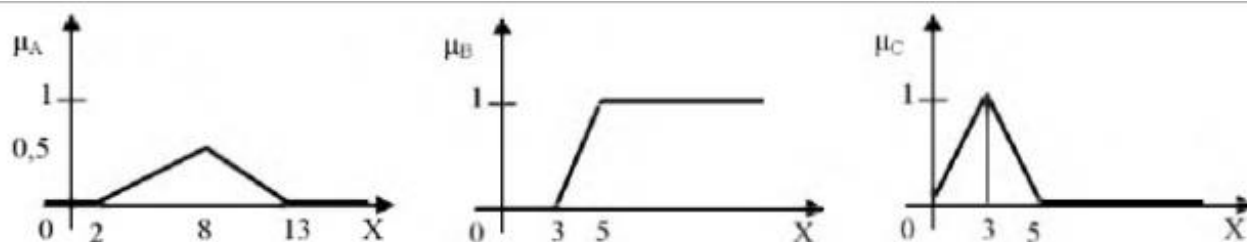
7. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap \bar{C}$  і визначити ступінь належності одного елемента множини D, використовуючи максимінний спосіб. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A = «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; B = trimf; C = zmf.



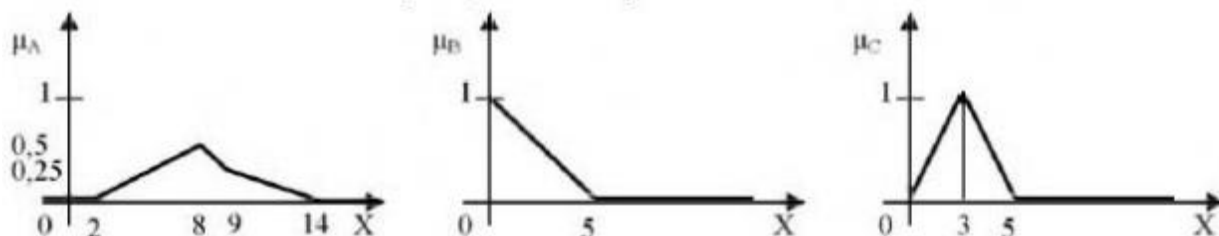
8. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap \bar{C}$  і визначити ступінь належності одного елемента множини D, використовуючи алгебраїчний спосіб. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A = trimf; B = «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; C = trapmf.



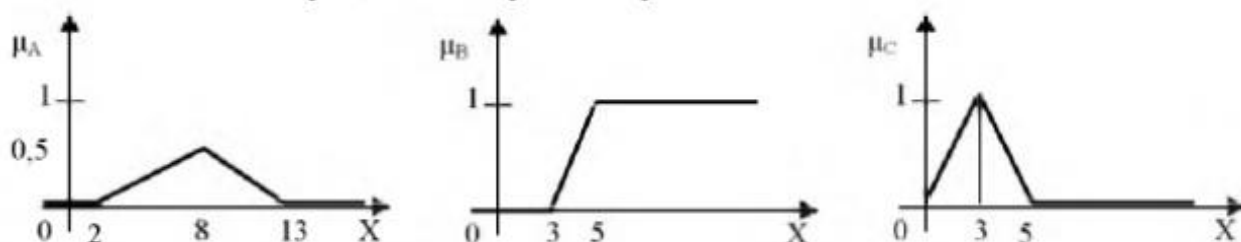
9. Дано 3 нечітких множини  $A$ ,  $B$ ,  $C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap \bar{C}$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи метод обмежень. При цьому функції належності нечітких множин  $A$ ,  $B$ ,  $C$  задати за допомогою вбудованих функцій fuzzy logic:  $A = \text{trimf}$ ;  $B = \text{smf}$ ;  $C =$  «створити дві функції і за допомогою логічних операцій отримати вихідну множину».



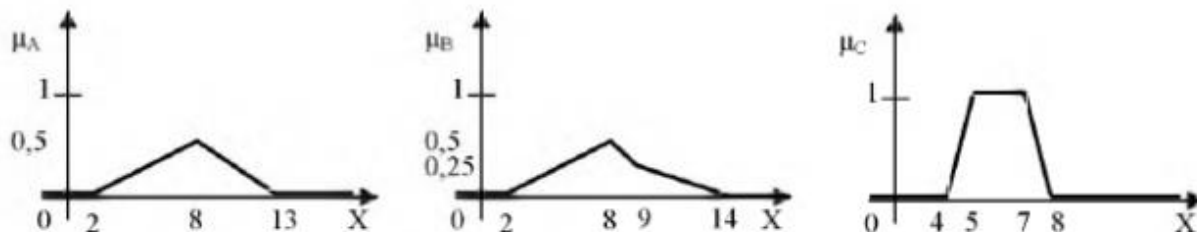
10. Дано 3 нечітких множини  $A$ ,  $B$ ,  $C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = \bar{A} \cup B \cap C$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи максимінний спосіб. При цьому функції належності нечітких множин  $A$ ,  $B$ ,  $C$  задати за допомогою вбудованих функцій fuzzy logic:  $A =$  «створити дві функції і за допомогою логічних операцій отримати вихідну множину»;  $B = \text{zmf}$ ;  $C = \text{trimf}$ .



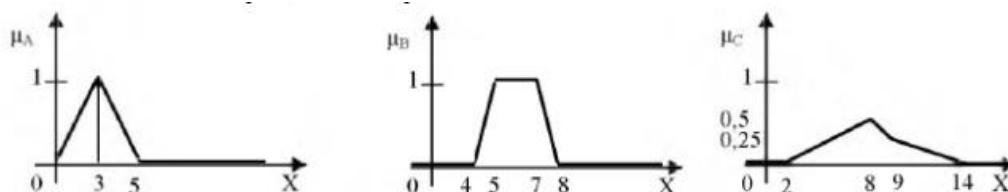
11. Дано 3 нечітких множини  $A$ ,  $B$ ,  $C$  (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = \bar{A} \cup B \cap C$  і визначити ступінь належності одного елемента множини  $D$ , використовуючи алгебраїчний спосіб. При цьому функції належності нечітких множин  $A$ ,  $B$ ,  $C$  задати за допомогою вбудованих функцій fuzzy logic:  $A = \text{trimf}$ ;  $B = \text{smf}$ ;  $C =$  «створити дві функції і за допомогою логічних операцій отримати вихідну множину».



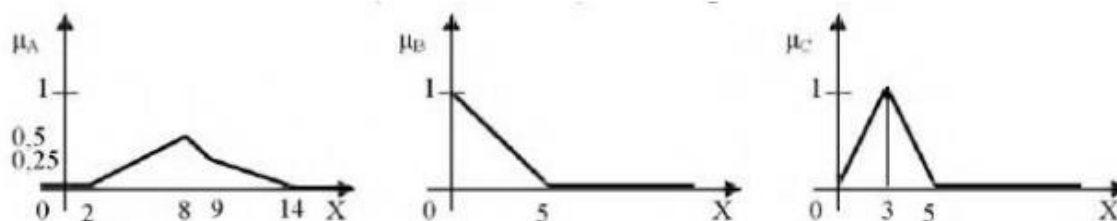
12. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = \bar{A} \cup B \cap C$  і визначити ступінь належності одного елемента множини D, використовуючи метод обмежень. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A=trimf; B= «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; C=dsigmf.



13. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = (\bar{A} \cup B) \cap \bar{C}$  і визначити ступінь належності одного елемента множини D, використовуючи максимінний спосіб. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A=trimf; B= trapmf; C= «створити дві функції і за допомогою логічних операцій отримати вихідну множину».

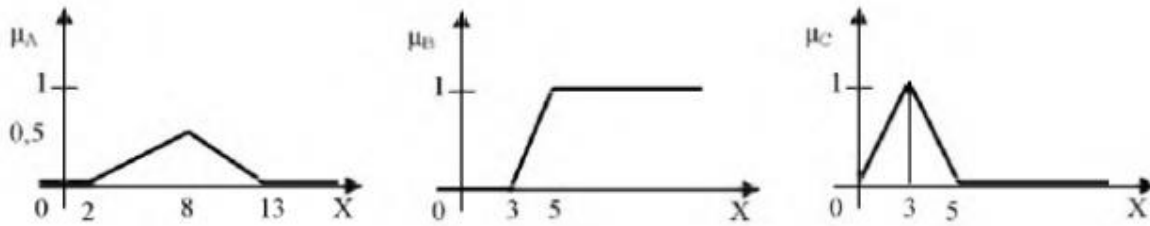


14. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = (\bar{A} \cup B) \cap \bar{C}$  і визначити ступінь належності одного елемента множини D, використовуючи алгебраїчний спосіб. При цьому функції належності нечітких множин A, B, C задати за допомогою вбудованих функцій fuzzy logic: A= «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; B=zmf; C= «створити дві функції і за допомогою логічних операцій отримати вихідну множину».



15. Дано 3 нечітких множини A, B, C (задані їх функції належності). Побудувати функцію належності нечіткої множини  $D = A \cup B \cap C$  і визначити ступінь належності одного елемента множини D, використовуючи метод обмежень. При цьому функції належності нечітких множин A, B, C задати за допомогою

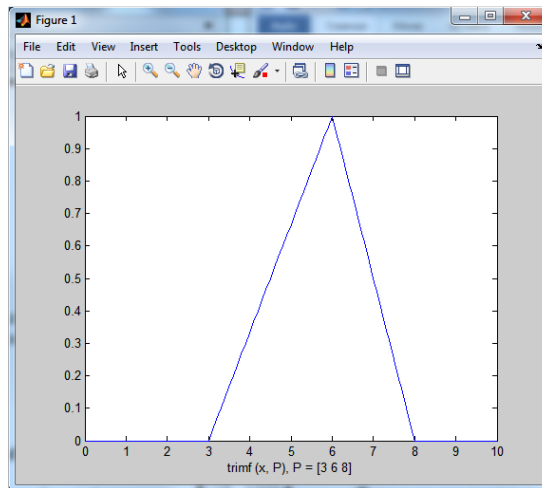
вбудованих функцій fuzzy logic: A= «створити дві функції і за допомогою логічних операцій отримати вихідну множину»; V=sigmf; C=trimf.



### Приклад виконання роботи

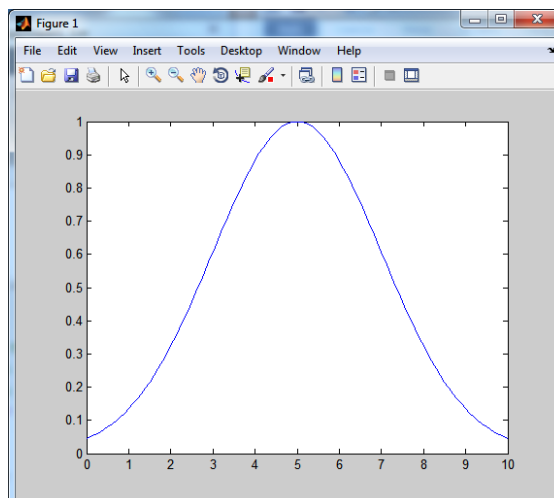
Програма використання ФН trimf:

```
x=0:0.1:10; % Задається базове множество  
y =trimf (x,[3 6 8]); % Определяется треугольная ФП  
plot (x,y); % Выводится график функции  
xlabel('trimf (x, P), P = [3 6 8]');
```



Програма використання ФН gaussmf:

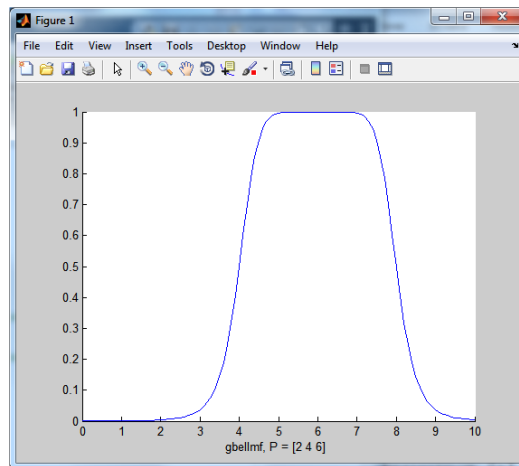
```
x = 0 : 0.1 :10;  
y = gaussmf (x, [2 5]);  
plot (x, y);
```





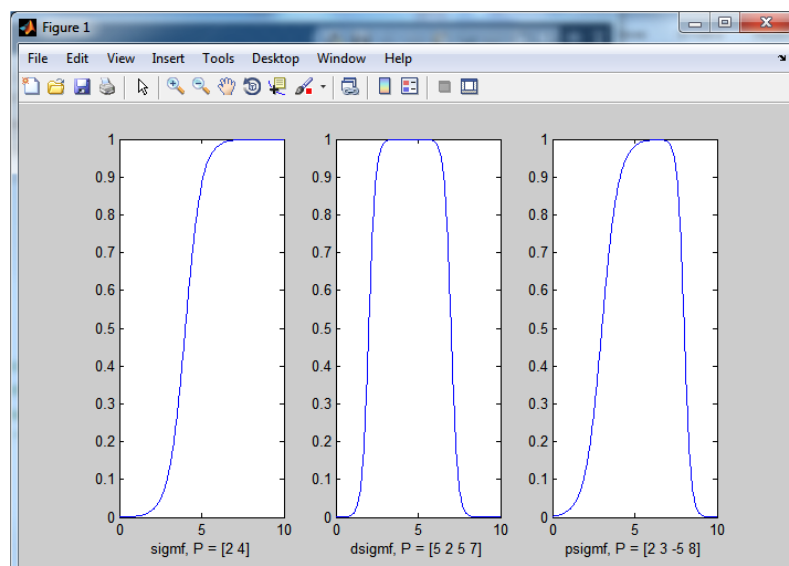
## Програма використання ФН gbellmf:

```
x = 0:0.1:10;  
y = gbellmf (x, [2 4 6]);  
plot (x, y);  
xlabel('gbellmf, P = [2 4 6]')
```



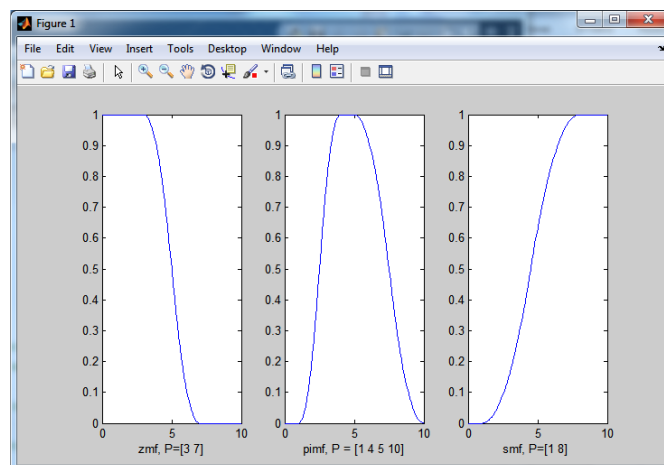
## Програма використання сігмодних функцій:

```
x = 0:0.1:10; % определяется базовое множество  
subplot (1,3,1) ; % формируется матрица графиков (3 x 1) – первый элемент –  
текущий  
y=sigmf (x, [2 4]) ;  
plot (x, y); % выводится график в первый элемент матрицы  
xlabel ('sigmf, P = [2 4]');  
subplot (1,3,2); % выбирается второй текущий элемент  
y = dsigmf (x, [5 2 5 7]) ;  
plot (x, y) ; % выводится график во второй элемент матрицы  
xlabel ('dsigmf, P = [5 2 5 7]');  
subplot (1,3,3); % выбирается третий текущий элемент  
y = psigmf (x, [2 3 -5 8]);  
plot (x, y); % выводится график в третий элемент матрицы  
xlabel ('psigmf, P = [2 3 -5 8]');
```



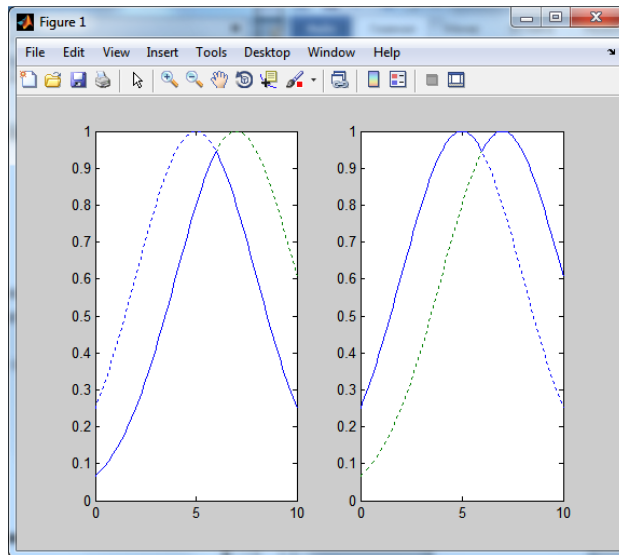
## Програма використання поліноміальних кривих:

```
x = 0: 0.1:10;  
subplot (1,3,1);  
y = zmf (x, [3 7] ) ;  
plot (x, y);  
xlabel ('zmf, P=[3 7]');  
subplot (1, 3, 2);  
y = pimf (x,[1 4 5 10]) ;  
plot (x, y);  
xlabel ('pimf, P = [1 4 5 10]');  
subplot (1, 3, 3);  
y = smf (x, [1 8]) ;  
plot (x, y);  
xlabel ('smf, P=[1 8] ')
```



## Програма використання максимічних операцій min і max:

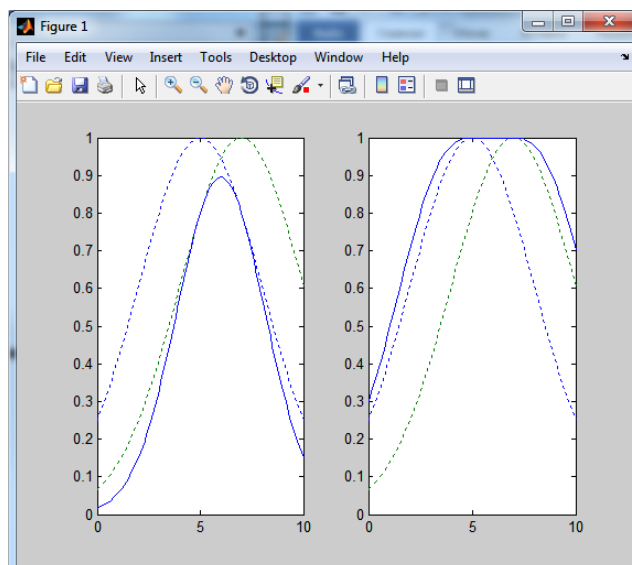
```
x = 0:0.1:10;  
subplot(1, 2, 1);  
y1 = gaussmf (x, [3 5] ) ;  
y2 = gaussmf (x, [3,7]);  
y3 = min ( [y1; y2]);  
plot (x, [y1 ; y2] , ':' ) ; % построение исходных ФП пунктирной линией  
hold on;% включение механизма добавления кривой в текущей график  
plot (x, y3);  
hold off; % выключение механизма добавления кривой в текущей график  
subplot(1, 2, 2);  
y4 = max ( [y1; y2]);  
plot (x, [y1 ; y2] , ':' ) ;  
hold on;  
plot (x, y4);  
hold off;
```



Пунктирною лінією на графіках зображені вихідні ФН, а суцільною лінією – результат дії логічних операторів.

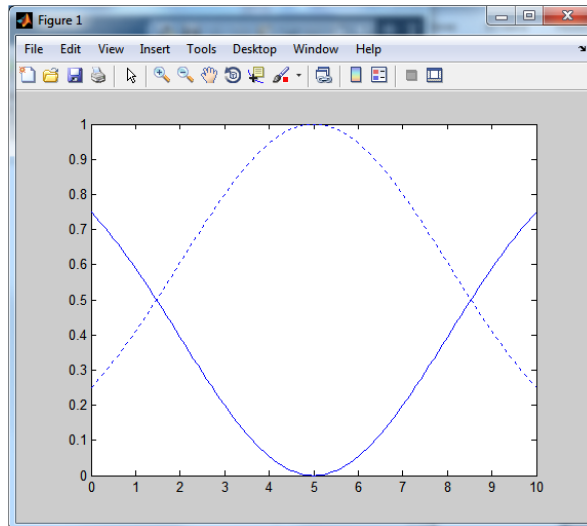
Програма використання імовірнісних операторів кон'юнкції і диз'юнкції:

```
x = 0:0.1:10;
subplot (1, 2, 1);
y1 = gaussmf (x, [3 5]);
y2 = gaussmf (x, [3 7]);
y3 = prod ([y1; y2]);
plot (x, [y1; y2] , ':' )
hold on;
plot (x, y3) ;
hold off;
subplot(1, 2, 2);
y4 = probor ( [y1; y2]);
plot (x, [y1; y2] , ':' ) ;
hold on;
plot (x, y4) ;
hold off
```



## Програма використання операції доповнення:

```
x = 0 : 0.1 : 10;  
y1 = gaussmf (x, [3 5]);  
y = 1 - y1;  
plot (x, y1, ': ');  
hold on;  
plot(x, y);  
hold off;
```



### Контрольні питання до захисту

1. Назвіть вбудовані функції належності пакета fuzzy logic?
2. Що таке функція належності?
3. Які кон'юнктивні і диз'юнктивні оператори ви знаєте?
4. Назвіть операції з нечіткими множинами, вбудовані в пакет fuzzy logic?

ЛАБОРАТОРНА РОБОТА №4  
ПРОЕКТУВАННЯ СИСТЕМИ ТИПУ МАМДАНІ НА ПРИКЛАДІ ПОБУДОВИ  
НЕЧІТКОЇ АПРОКСИМУЮЧОЇ СИСТЕМИ

**Мета:** вивчити засоби нечіткого моделювання в пакеті MatLab, спроектувати нечітку апроксимуючу систему типу Мамдані.

**Теоретичні відомості**

**Засоби нечіткого моделювання та керування в пакеті MatLab**

Пакет Matlab компанії MathWorks (США) містить широкий набір готових функцій, які використовуються для нечіткого моделювання. У пакеті є набір алгоритмів, що утворюють так звану інструментарій, який може використовуватися для проектування, аналізу і моделювання. Крім цього в пакеті Matlab є набір блоків Simulink, що дозволяє в графічній формі виконувати моделювання досить складних систем, в тому числі і з використанням блоків, що реалізують нечітке керування. Для вирішення завдань методами теорії нечітких множин в пакеті Matlab передбачений пакет нечіткої логіки Fuzzy Logic Toolbox. Основні можливості пакету:

- побудова систем нечіткого виводу (експертних систем, регулятора, апроксиматорів залежностей);
- побудова адаптивних нечітких систем (гібридних нейронних мереж);
- інтерактивне динамічне моделювання в середовищі Simulink.

Пакет забезпечує роботу:

- в режимі графічного інтерфейсу;
- в режимі командного рядка;
- з використанням блоків і прикладів пакета Simulink.

Моделювання нечіткого управління виконується за допомогою системи нечіткого виведення FIS (Fuzzy Inference System) (рис.14), що включає редактор системи нечіткого виведення (FIS- Editor), редактор функцій належності (The Member Ship Function Editor), редактор правил (The Rule Editor), підсистему для перегляду правил і схем нечітких висновків (The Rule Viewer), отриманих поверхонь (The Surface Viewer).

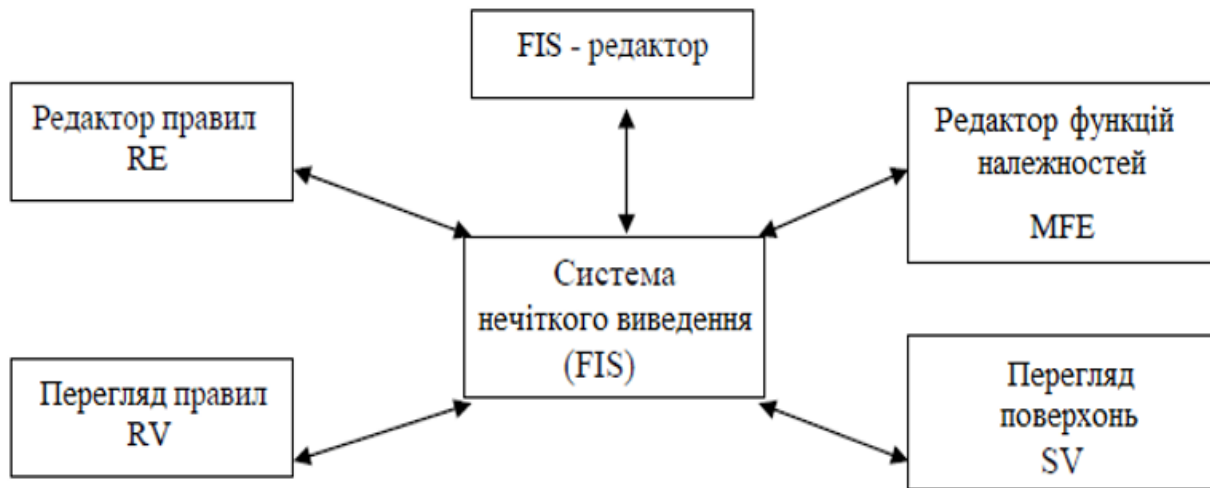


Рисунок 14 - Система нечіткого виведення FIS

FIS-редактор забезпечує високий рівень спілкування з системою, не має обмежень на кількість вхідних і вихідних змінних, яке обмежується лише доступним обсягом пам'яті ЕОМ.

Редактор функцій належності використовується для завдання виду функцій належності для кожної змінної. Редактор правил застосовується для редагування тексту правил умовного логічного виведення при описі поведінки модельованої системи.

Переглядачі правил і поверхонь необхідні для візуального контролю. Переглядач правил відображає схему нечіткого виведення на останньому етапі і використовується як засіб діагностики. З його допомогою можна, наприклад, побачити які правила активні, або оцінити вплив форми окремої функції належності на результат. Переглядач поверхонь використовується для представлення на екрані залежності одного виходу від другого або двох входів, а також генерації та побудови картини поверхні виходу для системи.

Всі компоненти FIS можуть взаємодіяти і обмінюватися даними в процесі моделювання.

У пакеті Matlab можливо використання шести видів функції належності:

- трикутної (trimf);
- трапецеїдальної (trmf);
- функції належності у вигляді кривої Гаусса (gaussmf) або складеної з двох кривих Гаусса (gauss2mf);
- колоколоподібної (bellmf);
- сигма-функції, призначеної для відтворення несиметричних функцій належності: sigmf – функція належності, відкрита справа, dsigmf – закрыта функція

належності, складена з різниці двох сигма-функцій,  $\text{psigmf}$  – закрита функція належності, що утворена з добутку двох сигма-функцій;

– трьох функцій належності, заснованих на поліноміальних кривих:  $\text{zmf}$  – несиметрична функція належності, відкрита зліва,  $\text{smf}$  – несиметрична функція належності, відкрита справа,  $\text{pmf}$  – закрита функція належності.

Крім цього в пакеті Matlab є можливість для користувача конструювати власні функції належності. Система нечіткого моделювання підтримує два основних оператора «І» та «АБО». Імплікація реалізується через оператор «І», який представлений в двох видах:  $\min$  і добуток ( $\text{prod}$ ), «АБО» –  $\max$  і  $\text{probor}$  – оператор ймовірного «АБО», відомий ще як алгебраїчна сума і який вираховується за рівнянням  $\text{probor}(a, b) = a + b - ab$ .

Крім цих операцій в пакеті нечіткої логіки Fuzzy Logic Toolbox представлені операції концентрування і розтягування. Пакет нечіткої логіки підтримує також всі відомі операції над нечіткими відносинами.

Для реалізації нечітких висновків використовуються алгоритми Мамдані (Mamdani) і Сугено (Sugeno).

## Моделювання нечіткої системи засобами Fuzzy Logic Toolbox

### Редактор систем нечіткого виведення

Редактор систем нечіткого виводу FIS (або просто редактор FIS) є основним засобом, що використовується для створення або редагування систем нечіткого виводу в графічному режимі (рис.15). Редактор FIS може бути відкритий за допомогою введення функції `fuzzy` у вікні команд. Редактор FIS має головне меню, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т.д.

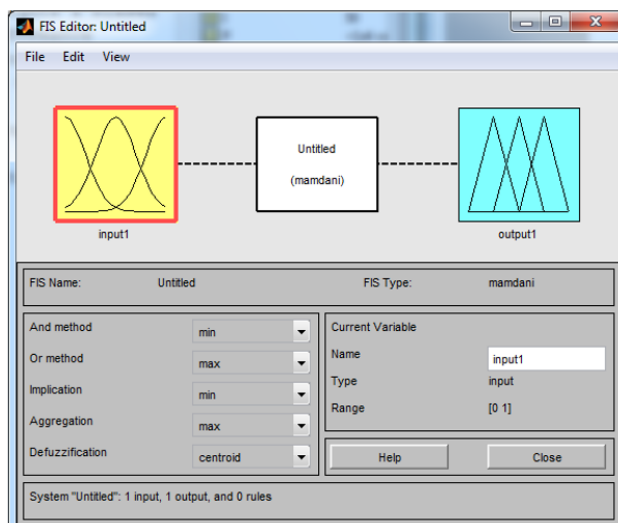


Рисунок 15 - Вікно редактора систем нечіткого виведення

Пункт меню File (Файл) редактора FIS дозволяє вибрати тип системи нечіткого виведення: Mamdani – типу Мамдані або Sugeno – типу Сугено, завантажити в редактор FIS існуючу систему нечіткого виведення та зберегти редаговану систему нечіткого виводу.

Пункт меню Edit (Редагування) дозволяє викликати редактор функцій належності і редактор правил нечіткого виведення.

Пункт меню View (Вид) дозволяє викликає програму перегляду правил нечіткого виводу і програму перегляду поверхні нечіткого виведення. У лівій нижній частині робочого інтерфейсу редактора FIS є 5 спливаючих меню:

1) And method (Метод логічної кон'юнкції) – дозволяє задати метод для виконання логічної кон'юнкції в умовах нечітких правил.

2) Or method (Метод логічної диз'юнкції) – дозволяє задати метод для виконання логічної диз'юнкції в умовах нечітких правил.

3) Implication method (Метод виведення висновку) – дозволяє задати метод для виконання (активізації) логічного висновку в кожному з нечітких правил (не використовується для систем типу Сугено).

4) Aggregation method (Метод агрегування) – дозволяє задати метод для агрегування значень функції приналежності кожної з вихідних змінних у висновках нечітких правил.

5) Defuzzification method (Метод дефазифікації) – дозволяє задати метод для виконання дефазифікації вихідних змінних в системі нечіткого виведення типу Мамдані або типу Сугено.

### **Редактор функцій належності**

Редактор функцій належності (рис.16) призначений для завдання і редагування функцій приналежності окремих термів системи нечіткого виведення в графічному режимі. Програму-редактор функцій належності можна відкрити трьома способами: через пункт меню View → Edit membership functions ...; подвійним клацанням лівої кнопки миші по іконці, що відображає відповідну змінну; натисканням клавіш Ctrl + 2.

Редактор надає користувачеві не тільки можливість вибрати будь-яку з 11 вбудованих функцій належності, але і задати власну функцію належності. Для кожної функції належності можна змінити її ім'я, тип і параметри.

Для відображення графіків функцій належності слід вибрати необхідну змінну в лівій частині графічного інтерфейсу редактора під заголовком FIS Variables (Змінні FIS).



Щоб вибрати потрібну функцію належності, слід клацнути на ній або її мітці в основному вікні з графіками функцій належності.

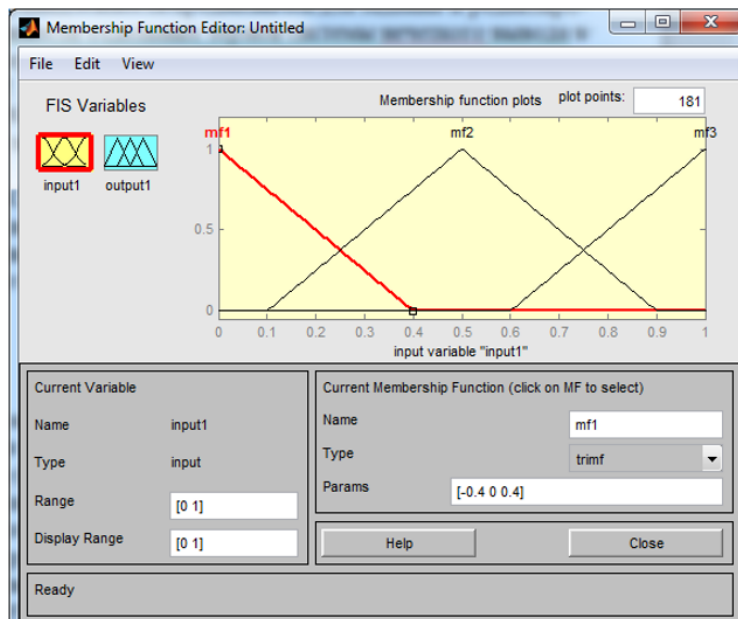


Рисунок 16 - Вікно редактору функцій належності

Редактор функцій належності має головне меню програми, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т. д.

### Редактор правил системи нечіткого виведення

Редактор правил системи нечіткого виведення (рис.17) призначений для завдання і редагування окремих правил системи нечіткого виведення в графічному режимі. Редактор правил може бути відкритий за допомогою головного меню редактора FIS командою меню Edit → Rules ... або натисканням клавіш Ctrl + 3.

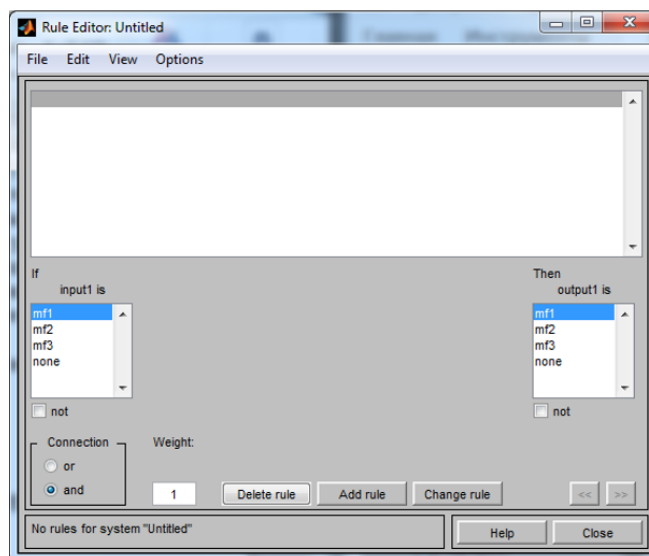


Рисунок 17 - Вікно редактору правил

Щоб використовувати цей редактор для створення правил, необхідно попередньо визначити все вхідні і вихідні змінні, для чого можна скористатися редактором системи нечіткого виведення FIS і редактором функцій належності. При цьому задати правила можна за допомогою вибору відповідних значень термів вхідних і вихідних змінних. Поля введення в середній частині графічного інтерфейсу редактора правил дозволяють задати нове правило в системі нечіткого виведення. Для цього необхідно виділити ім'я терма відповідної змінної, яка повинна бути попередньо визначена за допомогою редактора функцій належності. Якщо деякий терм не входить в правило, то для нього слід вибрати значення none. Якщо в умові правила використовується логічне заперечення деякого терма, то для цього терма слід зазначити відповідний прапорець з міткою not. Редактор правил дозволяє також задати логічні зв'язки для підумови правила (перемикач Connection) і вага правила (поле введення Weight). Кнопки в нижній частині графічного інтерфейсу редактора правил, як випливає з їх назв, служать для видалення виділеного у вікні правила (Delete rule), додавання створеного правила в систему (Add rule) і внесення змін в виділене в вікні правило (Change rule). У правому нижньому кутку знаходяться кнопки виклику вбудованої довідкової системи MATLAB (Help) і кнопка закриття редактора правил (Close).

### Програма перегляду поверхні системи нечіткого виведення

Програма перегляду поверхні системи нечіткого виведення (рис.18) дозволяє переглядати поверхню системи нечіткого виведення і візуалізувати графіки залежності вихідних змінних від окремих вхідних змінних. Графічний інтерфейс програми перегляду правил може бути відкритий за допомогою головного меню редактора FIS, редактора функцій належності або редактора правил командою меню View → Surface або натисканням клавіш Ctrl + 6.

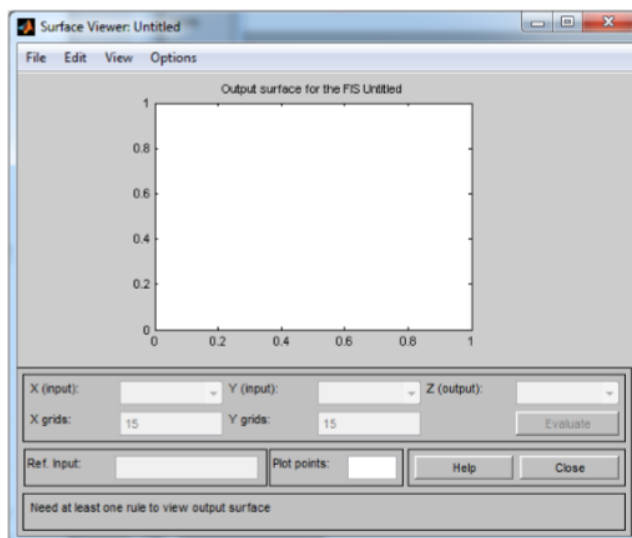


Рисунок 18 - Вікно програми перегляду поверхні

Програма перегляду поверхні виведення має головне меню, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т.д.

Програма перегляду поверхні виведення не дозволяє вносити зміни в систему нечіткого виведення і відповідну їй структуру FIS. Використовуючи головне меню програми, користувач може вибрати вхідні змінні і відповідні їм горизонтальні осі системи координат (X і Y), а також вихідну змінну, якій відповідає вертикальна вісь системи координат (Z).

Натиснувши і утримуючи ліву кнопку миші на осях графіка поверхні, за допомогою подальшого переміщення курсора миші в тому чи іншому напрямку можна змінити кут перегляду поверхні виведення. Якщо розглядається система нечіткого виведення з більш ніж двома вхідними змінними, то для невізуалізуємих вхідних змінних слід задати деякі постійні значення (константи).

## Варіанти завдань

### Загальна постановка задачі

Спроектувати систему типу Мамдані засобами пакета Fuzzy Logic Toolbox на прикладі побудови нечіткої апроксимуючої системи. Обрану нелінійну функцію описати базою правил для лінгвістичних змінних, що описують  $z(x, y)$ , визначених на безлічі з п'яти, семи, дев'яти і одинадцяти термів.

Примітка. Якщо  $z$  виходить за задані межі, то  $z$  набуває значення, яке дорівнює значенню мінімальної або максимальної межі, відповідно.

№ вар.	Функція	Область изменення		
		x	y	z
1	$z = x^2 + \sin(y - \pi/2)$	$-2 < x < 2$	$-\pi < y < \pi$	$-1 < z < 4$
2	$z = \cos(x) + \sin(y - \pi/2)$	$-\pi < x < \pi$	$-\pi < y < \pi$	$-2 < z < 2$
3	$z = x * \sin(y)$	$-\pi < x < \pi$	$-\pi < y < \pi$	$-3 < z < 3$
4	$z = x^2 - y^2$	$-4 < x < 4$	$-4 < y < 4$	$-4 < z < 4$
5	$z = x^2 * y^2$	$-3 < x < 3$	$-3 < y < 3$	$0 < z < 4$
6	$z = x * y$	$-3 < x < 3$	$-3 < y < 3$	$-4 < z < 4$
7	$z = (x - y) * y + 1$	$-3 < x < 3$	$-3 < y < 3$	$-4 < z < 3$
8	$z = y * \sin(x + y)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$-0.5 < z < 1.5$
9	$z = \sin(2 * x / \pi) * \sin(2 * y / \pi)$	$-5 < x < 5$	$5 < y < 5$	$1 < z < 1$
10	$z = y * \sin(x)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$1 < z < 1.5$
11	$z = y * \cos(x)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$-1 < z < 1.5$
12	$z = x * \cos(x) + y * \sin(y)$	$-2 < x < 2$	$-\pi/2 < y < \pi/2$	$-0.6 < z < 2$

## Приклад виконання роботи

Розглянемо основні етапи проектування систем типу Мамдані на прикладі створення системи нечіткого логічного висновку, що моделює залежність

$$y = x_1^2 \cdot \sin(x_2 - 1), \quad x_1 \in [-7; 3], \quad x_2 \in [-4.4; 1.7]$$

Проектування системи нечіткого логічного висновку будемо проводити на основі графічного зображення зазначеної залежності.

Для побудови тривимірного зображення функції  $y = x_1^2 \cdot \sin(x_2 - 1)$  в області  $x_1 \in [-7; 3]$ ,  $x_2 \in [-4.4; 1.7]$  складемо наступну програму:

```
%Построение графика функции y=x1^2*sin(x2-1)
%в области x1e[-7,3] и x2e[-4.4,1.7].
n=15;
x1=-7:10/(n-1):3;
x2=-4.4:6.1/(n-1):1.7;
y=zeros(n,n);
for j=1:n
y(j,:)=x1.^2*sin(x2(j)-1);
end
surf(x1,x2,y)
xlabel('x1')
ylabel('x2')
zlabel('y')
title('Target');
```

В результаті виконання програми отримаємо графічне зображення, наведене на рис. 19. Проектування системи нечіткого логічного висновку, що відповідає наведеним графіком, полягає у виконанні наступної послідовності кроків.

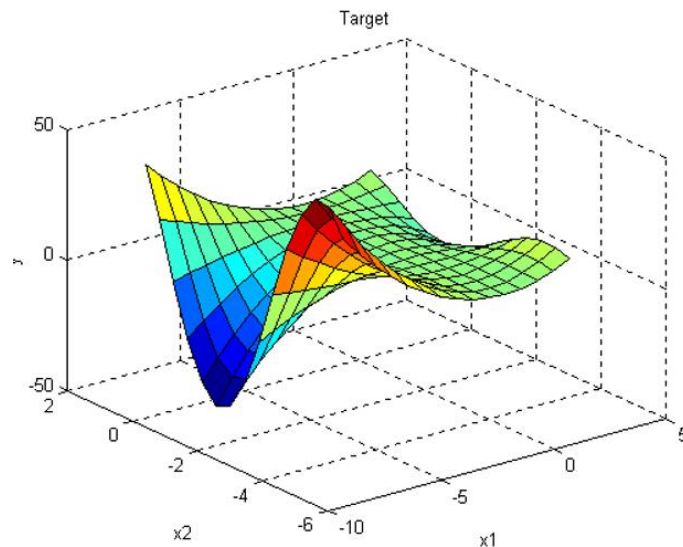


Рисунок 19 – Еталонна поверхня

**Крок 1.** Для завантаження основного `fis`-редактора надрукуємо слово **fuzzy** в командному рядку.

**Крок 2.** Додамо другу вхідну змінну. Для цього в меню **Edit** вибираємо команду **Add input**.

**Крок 3.** перейменуємо першу вхідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **input1**, введемо нове позначення **x1** в полі редагування імені поточної змінної і натиснемо **<Enter>**.

**Крок 4.** перейменуємо другу вхідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **input2**, введемо нове позначення **x2** в поле редагування імені поточної змінної і натиснемо **<Enter>**.

**Крок 5.** перейменуємо вихідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **output1**, введемо нове позначення **y** в поле редагування імені поточної змінної і натиснемо **<Enter>**.

**Крок 6.** Задамо ім'я системи. Для цього в меню **File** вибираємо в підменю **Export** команду **To disk** і вводимо ім'я файлу, наприклад, **first**.

**Крок 7.** Перейдемо в редактор функцій належності. Для цього зробимо подвійне клацання лівою кнопкою миші на блоці **x1**.

**Крок 8.** Задамо діапазон зміни змінної **x1**. Для цього надрукуємо **-7 3** в поле **Range** (рис.20) і натиснемо **<Enter>**.

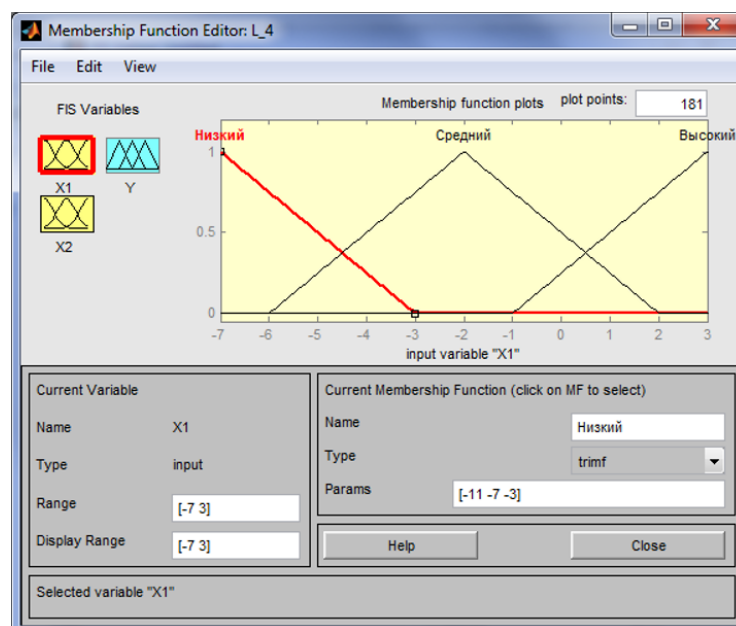


Рисунок 20 - Функції приналежності змінної **x1**

**Крок 9.** Задамо функції належності змінної **x1**. Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями належності. Для цього в меню **Edit** виберемо команду **Add MFs ...** В результаті з'явиться діалогове вікно вибору типу і кількості функцій приналежності. За замовчуванням це 3 терма з трикутними функціями належності. Тому просто натискаємо **<Enter>**.

**Крок 10.** Задамо найменування термів змінної **x1**. Для цього робимо одне клацання лівою кнопкою миші по графіку першої функції приналежності. Потім

вводимо найменування терму, наприклад, **Низький**, в поле **Name** і натиснемо **<Enter>**. Потім робимо одне клацання лівою кнопкою миші по графіку другої функції приналежності і вводимо найменування терму, наприклад, **Середній**, в поле **Name** і натиснемо **<Enter>**. Ще раз робимо одне клацання лівою кнопкою миші по графіку третьої функції приналежності і вводимо найменування терму, наприклад, **Високий**, в поле **Name** і натиснемо **<Enter>**.

**Крок 11.** Задамо функції належності змінної **x2**. Для лінгвістичної оцінки цієї змінної будемо використовувати 5 термів з гаусовими функціями належності. Для цього активізуємо змінну **x2** за допомогою клацання лівої кнопки миші на блоці **x2**. Задамо діапазон зміни змінної **x2**. Для цього надрукуємо **-4.4 1.7** в поле **Range** (рис.8) і натиснемо **<Enter>**. Потім в меню **Edit** виберемо команду **Add MFs ....** У діалоговому вікні вибираємо тип функції належності **gaussmf** в поле **MF type** і **5** термів в полі **Number of MFs**. Після цього натискаємо **<Enter>**.

**Крок 12.** За аналогією з кроком 10 задамо наступні найменування термів змінної **x2**: **Низький, Нижче середнього, Середній, Вище середнього, Високий**. В результаті отримаємо графічне вікно, зображене на рис.21.

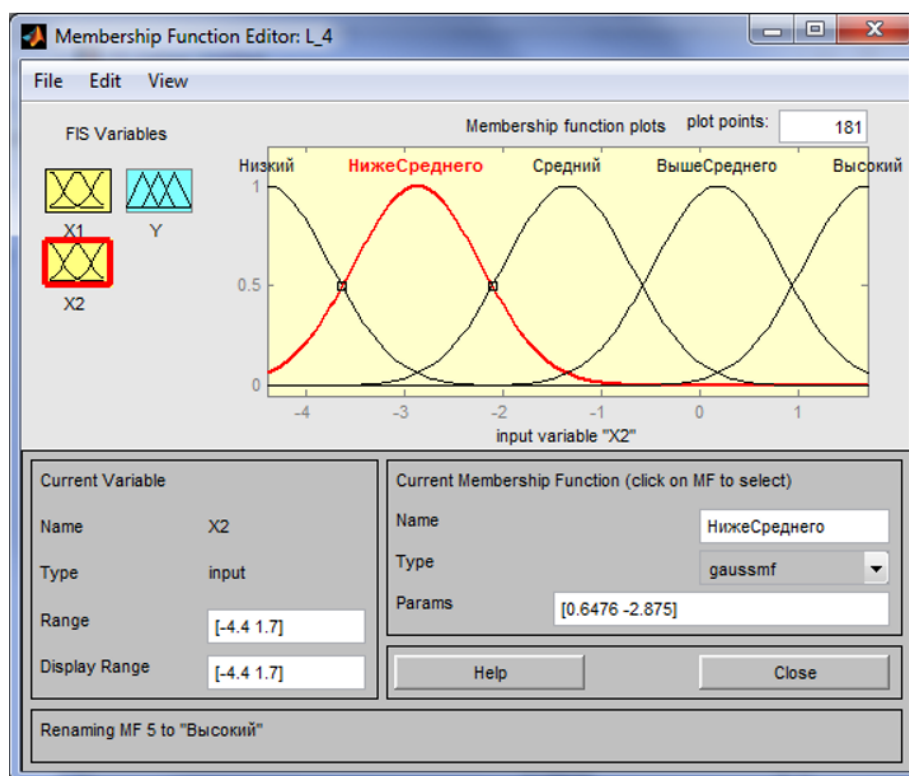


Рисунок 21 - Функції приналежності змінної **x2**

**Крок 13.** Задамо функції належності змінної **y**. Для лінгвістичної оцінки цієї змінної будемо використовувати **5** термів з трикутними функціями належності. Для цього активізуємо змінну **y** за допомогою клацання лівої кнопки миші на блоці **y**. Задамо діапазон зміни змінної **y**. Для цього надрукуємо **-50 50** в поле **Range** (Рис.9)



і натиснемо **<Enter>** .Потім в меню **Edit** виберемо команду **Add MFs ...** . У діалоговому вікні вибираємо **5** термів в полі **Number of MFs**. Після цього натискаємо **<Enter>**.

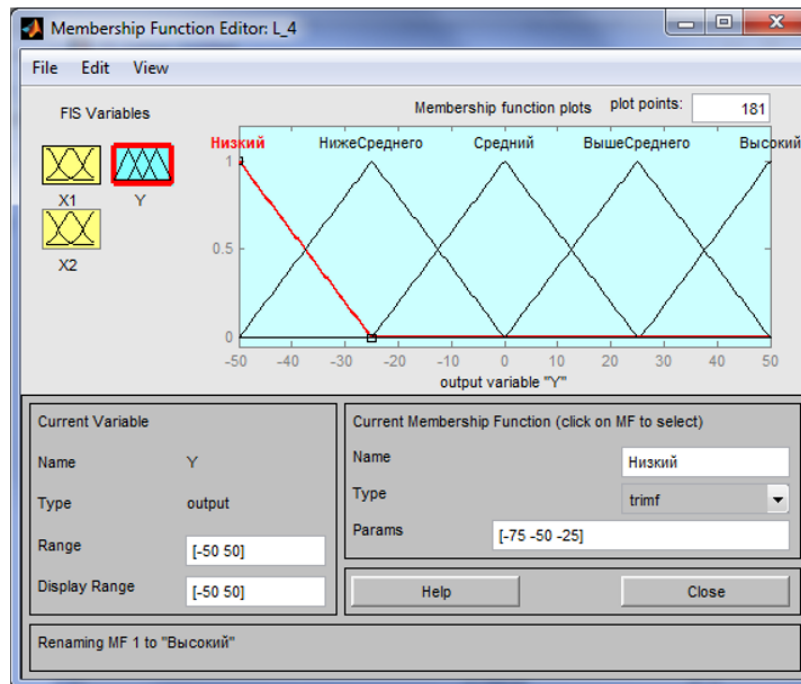


Рисунок 22 - Функції приналежності змінної у

**Крок 14.** За аналогією з кроком 10 задамо наступні найменування термів змінної у: **Низький, Нижче середнього, Середній, Вище середнього, Високий**. В результаті отримаємо графічне вікно, зображене на рис.22.

**Крок 15.** Перейдемо в редактор бази знань **RuleEditor**. Для цього виберемо в меню **Edit** виберемо команду **Edit rules ...**

**Крок 16.** На основі візуального спостереження за графіком, зображеним на рис. 7 сформулюємо наступні дев'ять правил:

**Якщо x1 = Середній, то у = Середній;**

**Якщо x1 = Низький і x2 = Низький, то у = Високий;**

**Якщо x1 = Низький і x2 = Високий, то у = Високий;**

**Якщо x1 = Високий і x2 = Високий, то у = Вище Середнього;**

**Якщо x1 = Високий і x2 = Низький, то у = Вище Середнього;**

**Якщо x1 = Високий і x2 = Середній, то у = Середній;**

**Якщо x1 = Низький і x2 = Середній, то у = Низький;**

**Якщо x1 = Високий і x2 = Вище Середнього, то у = Середній;**

**Якщо x1 = Високий і x2 = Нижче Середнього, то у = Середній.**

Для введення правила необхідно вибрати в меню відповідну комбінацію термів і натиснути кнопку Add rule. На рис.23 зображено вікно редактора бази

знань після введення всіх дев'яти правил. Число, наведене в дужках в кінці кожного правила є ваговим коефіцієнтом відповідного правила.

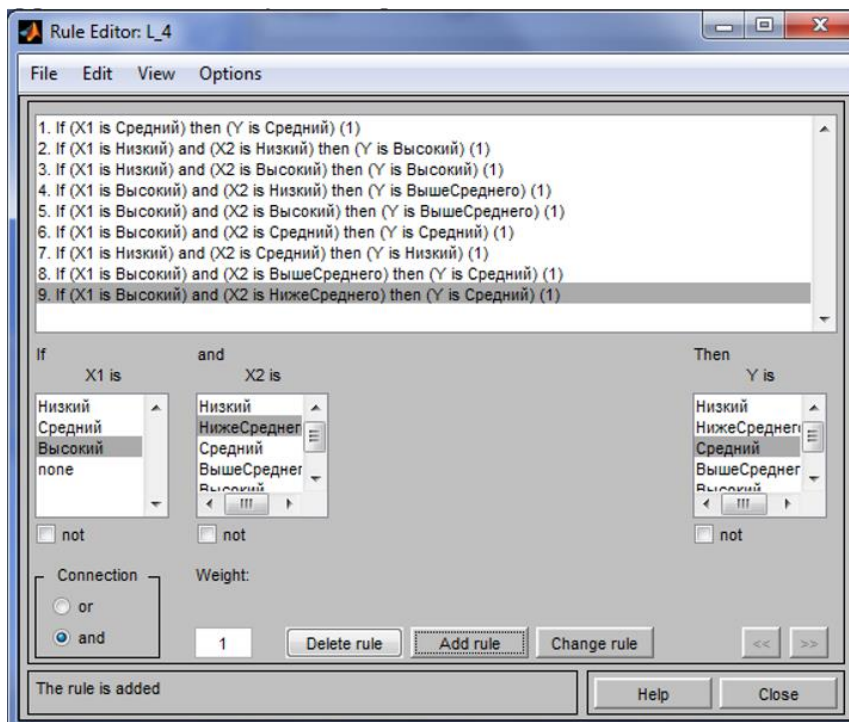


Рисунок 23 - База знань в RuleEditor

**Крок 17.** Збережемо створену систему. Для цього в меню **File** вибираємо в підменю **Export** команду **To disk**. На рис.24 наведено вікно візуалізації нечіткого логічного висновку. Це вікно активізується командою **View rules ...** меню **View**. В поле **Input** вказуються значення входних змінних, для яких виконується логічний висновок.

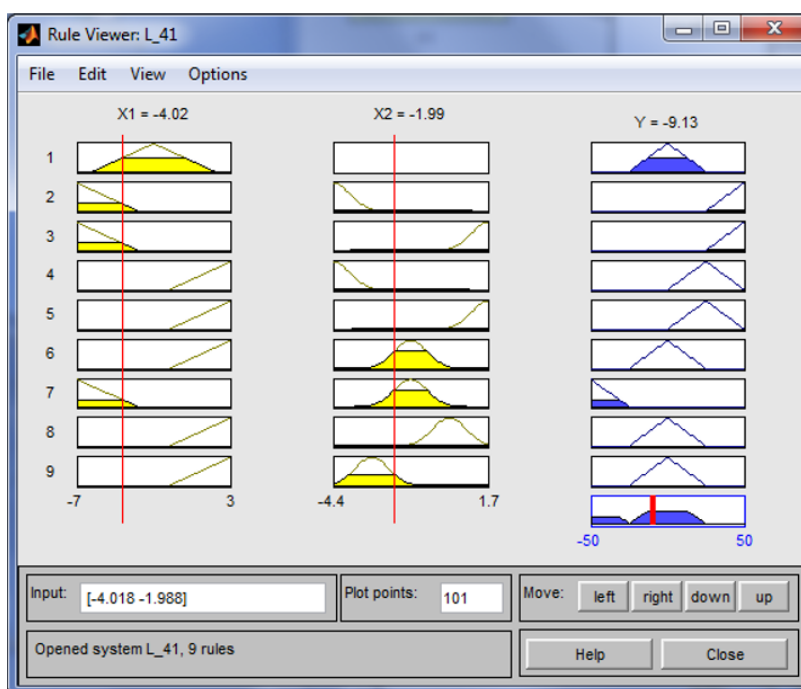


Рисунок 24 - Візуалізація нечіткого логічного виведення в RuleViewer



На рис.25 наведена поверхня «входи-вихід», відповідна синтезованої нечіткої системі. Для виведення цього вікна необхідно використовувати команду **View surface ...** меню **View**. Порівнюючи поверхні на рис. 6 і на рис.12 можна зробити висновок, що нечіткі правила досить добре описують складну нелінійну залежність.

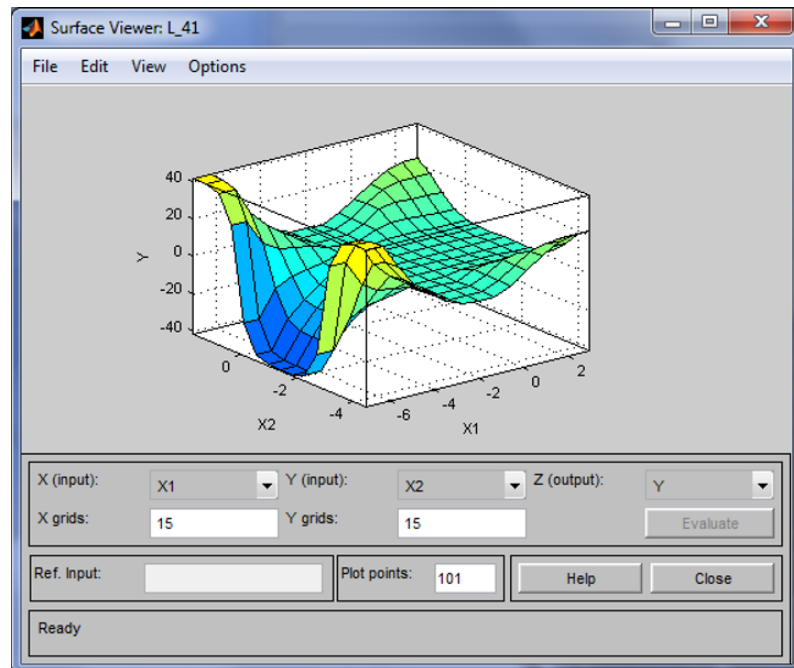


Рисунок 25 - Поверхня «входи-вихід» у вікні SurfaceViewer

### Контрольні питання до захисту

1. Для чого призначений FIS-редактор?
2. Назвіть основні елементи вікна FIS-редактора?
3. Опишіть редактор функцій належності?
4. Якого типу є функції приналежності в пакеті fuzzy logic?
5. Опишіть вікно редактора баз знань?
6. Яким чином можна зробити візуалізацію нечіткого логічного висновку?
7. Що таке візуалізація поверхні типу «входи-виходи»?

ЛАБОРАТОРНА РОБОТА №5  
ПРОЕКТУВАННЯ СИСТЕМИ ТИПУ СУГЕНО НА ПРИКЛАДІ ПОБУДОВИ  
НЕЧІТКОЇ АПРОКСИМУЮЧОЇ СИСТЕМИ

**Мета:** вивчити засоби нечіткого моделювання в пакеті MatLab, спроектувати нечітку апроксимуючу систему типу Сугено.

**Теоретичні відомості**

Модуль Fuzzy Logic Toolbox дозволяє будувати нечіткі системи двох типів - Мамдані та Сугено. Основна відмінність між системами Мамдані та Сугено полягає в різних способах завдання значень вихідної змінної в правилах, що утворюють базу знань. У системах типу Мамдані значення вихідної змінної задаються нечіткими термами, у системах типу Сугено - як лінійна комбінація вхідних змінних. Робота в середовищі MatLab аналогічна лабораторній роботі №4.

**Алгоритм Сугено (Sugeno)**

У алгоритмі Сугено база знань будується з правил в наступній формі:

$$\text{ЯКЩО } x \in A_1 \text{ і } y \in B_1, \text{ ТО } z_1 = a_1x + b_1y,$$

$$\text{ЯКЩО } x \in A_2 \text{ і } y \in B_2, \text{ ТО } z_2 = a_2x + b_2y$$

Етапи нечіткого виводу :

1. Фазифікація: знаходяться міри істинності для передумов кожного правила :  
 $A_1(x_0), A_2(x_0), B_1(y_0), B_2(y_0)$ .
2. Вивід: знаходяться рівні відсікання для передумов кожного з правил з використанням операції мінімум:

$$\alpha_1 = A_1(x_0) \wedge B_1(y_0),$$

$$\alpha_2 = A_2(x_0) \wedge B_2(y_0),$$

де  $\wedge$  - операція логічного мінімуму.

Знаходяться індивідуальні виходи правил :

$$z^*_1 = a_1x + b_1y,$$

$$z^*_2 = a_2x + b_2y$$

3. Визначається чітке значення змінної виводу :

$$z_0 = \frac{\alpha_1 z^*_1 + \alpha_2 z^*_2}{\alpha_1 + \alpha_2}$$

## Варіанти завдань

### Загальна постановка задачі

Спроектувати систему типу Сугено засобами пакета Fuzzy Logic Toolbox на прикладі побудови нечіткої апроксимуючої системи. Обрану нелінійну функцію описати базою правил для лінгвістичних змінних, що описують  $z(x, y)$ , визначених на множині з п'яти, семи, дев'яти і одинадцяти термів.

Примітка. Якщо  $z$  виходить за задані межі, то  $z$  набуває значення, яке дорівнює значенню мінімальної або максимальної межі, відповідно.

№ вар.	Функція	Область изменения		
		x	y	z
1	$z = x^2 + \sin(y - \pi/2)$	$-2 < x < 2$	$-\pi < y < \pi$	$-1 < z < 4$
2	$z = \cos(x) + \sin(y - \pi/2)$	$-\pi < x < \pi$	$-\pi < y < \pi$	$-2 < z < 2$
3	$z = x * \sin(y)$	$-\pi < x < \pi$	$-\pi < y < \pi$	$-3 < z < 3$
4	$z = x^2 - y^2$	$-4 < x < 4$	$-4 < y < 4$	$-4 < z < 4$
5	$z = x^2 * y^2$	$-3 < x < 3$	$-3 < y < 3$	$0 < z < 4$
6	$z = x * y$	$-3 < x < 3$	$-3 < y < 3$	$-4 < z < 4$
7	$z = (x - y) * y + 1$	$-3 < x < 3$	$-3 < y < 3$	$-4 < z < 3$
8	$z = y * \sin(x + y)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$-0.5 < z < 1.5$
9	$z = \sin(2 * x / \pi) * \sin(2 * y / \pi)$	$-5 < x < 5$	$5 < y < 5$	$1 < z < 1$
10	$z = y * \sin(x)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$1 < z < 1.5$
11	$z = y * \cos(x)$	$-\pi/2 < x < \pi/2$	$-\pi/2 < y < \pi/2$	$-1 < z < 1.5$
12	$z = x * \cos(x) + y * \sin(y)$	$-2 < x < 2$	$-\pi/2 < y < \pi/2$	$-0.6 < z < 2$

### Приклад виконання роботи

Розглянемо основні етапи проектування систем типу Сугено на прикладі створення системи нечіткого логічного виведення, що моделює залежність

$$y = x_1^2 \cdot \sin(x_2 - 1), \quad x_1 \in [-7; 3], \quad x_2 \in [-4.4; 1.7]$$

Моделювання цієї залежності будемо здійснювати за допомогою наступної бази знань:

1. Якщо  $x_1$ =Середній, то  $y=0$ ;
2. Якщо  $x_1$ =Високий та  $x_2$ =Високий, то  $y=2x_1+2x_2+1$ ;
3. Якщо  $x_1$ =Високий та  $x_2$ =Низький, то  $y=4x_1-x_2$ ;
4. Якщо  $x_1$ =Низький та  $x_2$ =Середній, то  $y=8x_1+2x_2+8$ ;
5. Якщо  $x_1$ =Низький та  $x_2$ =Низький, то  $y=50$ ;
6. Якщо  $x_1$ =Низький та  $x_2$ =Високий, то  $y=50$ .

Проектування системи нечіткого логічного виведення типу Сугено складається у виконанні наступної послідовності кроків.

**Крок 1.** Для завантаження основного **fis**-редактора надрукуємо слово **fuzzy** у командному рядку. Після цього відкриється нове графічне вікно.

**Крок 2.** Виберемо тип системи. Для цього в меню **File** вибираємо в підменю **New fis...** команду **Sugeno**.

**Крок 3.** Додамо другу вхідну змінну. Для цього в меню **Edit** вибираємо команду **Add input**.

**Крок 4.** Перейменуємо першу вхідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **input1**, уведемо нове позначення **x1** у поле редагування імені поточної змінної і натиснемо **[Enter]**.

**Крок 5.** Перейменуємо другу вхідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **input2**, уведемо нове позначення **x2** у поле редагування імені поточної змінної і натиснемо **[Enter]**.

**Крок 6.** Перейменуємо вихідну змінну. Для цього зробимо одне клацання лівою кнопкою миші на блоці **output1**, уведемо нове позначення **y** у поле редагування імені поточної змінної і натиснемо **[Enter]**.

**Крок 7.** Задамо ім'я системи. Для цього в меню **File** вибираємо в підменю **Export** команду **To disk** і введемо ім'я файлу, наприклад, **FirstSugeno**.

**Крок 8.** Перейдемо в редактор функцій приналежності. Для цього зробимо подвійного клацання лівою кнопкою миші на блоці **x1**.

**Крок 9.** Задамо діапазон зміни змінної **x1**. Для цього надрукуємо **-7 3** у полі **Range** і натиснемо **[Enter]**.

**Крок 10.** Задамо функції приналежності змінної **x1**. Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями приналежності, що встановлені за замовчуванням. Задамо найменування термів змінної **x1**. Для цього робимо одне клацання лівою кнопкою миші на графіку першої функції приналежності. Потім надрукуємо найменування терму **Низький** у поле **Name**. Потім робимо одне клацання лівою кнопкою миші на графіку другої функції приналежності і вводимо найменування терму **Середній** у поле **Name**. Ще раз робимо одне клацання лівою кнопкою миші на графіку третьої функції приналежності і вводимо найменування терму **Високий** у поле **Name** і натиснемо **[Enter]** (рис.26).

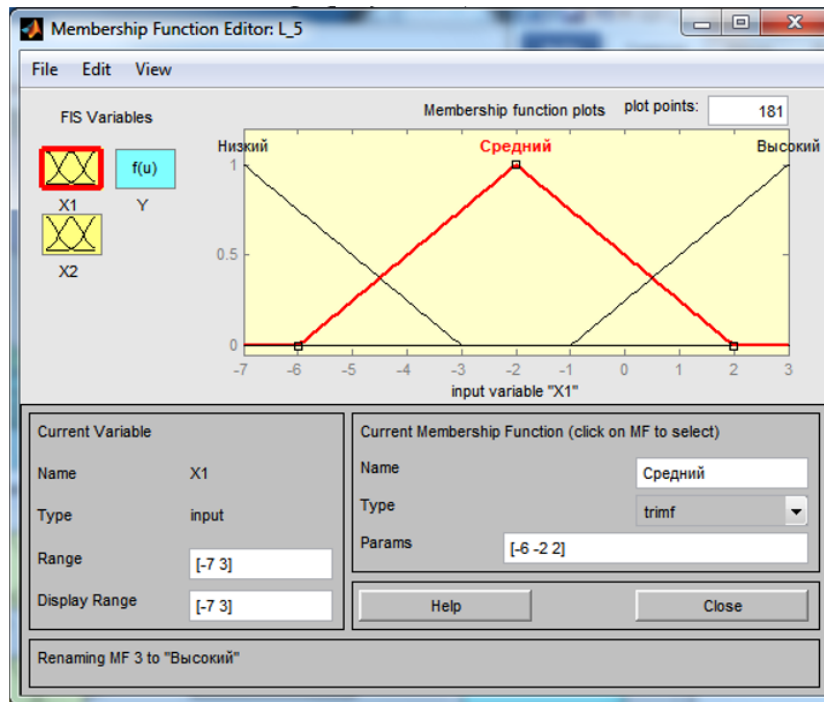


Рисунок 26 – Функції приналежності змінної  $x_1$

**Крок 11.** Задамо функції приналежності змінної  $x_2$ . Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями приналежності, що встановлені за замовчуванням. Для цього активізуємо змінну  $x_2$  за допомогою клацання лівої кнопки миші на блоці  $x_2$ . Задамо діапазон зміни змінної  $x_2$ . Для цього надрукуємо **-4.4 1.7** у полі **Range** і натиснемо **[Enter]**. За аналогією з попереднім кроком задамо наступні найменування термів змінної  $x_2$ : **Низький, Середній, Високий** (рис.27).

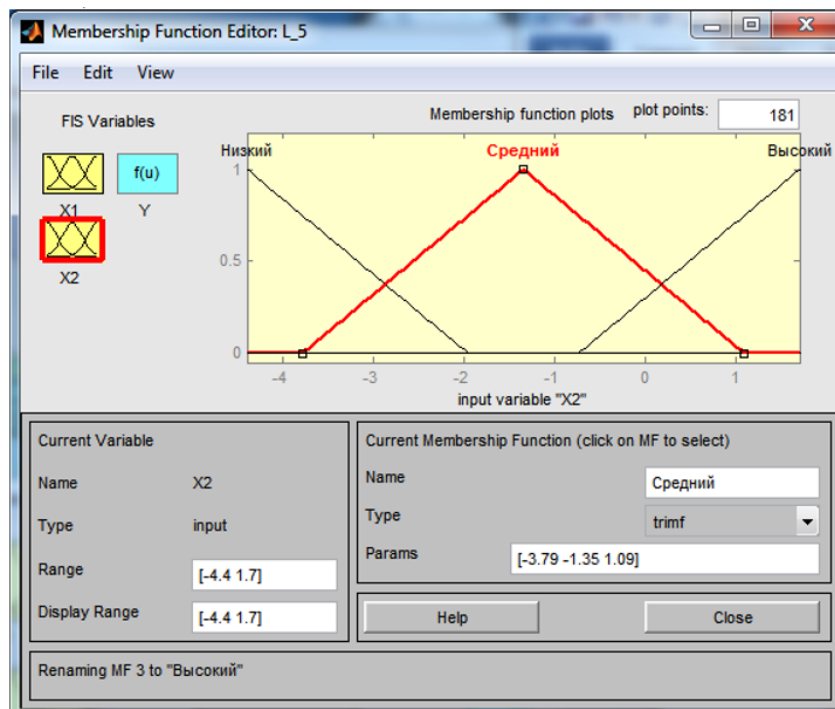


Рисунок 27 – Функції приналежності змінної  $x_2$

**Крок 12.** Задамо лінійні залежності між входами і виходом, приведені в базі знань. Для цього активізуємо змінну  $y$  за допомогою клацання лівої кнопки миші на блоці  $y$ . У правому верхньому куті з'явилося позначення трьох функцій приналежності, кожна з яких відповідає одній лінійній залежності між входами і виходом.

У базі знань зазначені 5 різних залежностей:

$$y=50; y=4x_1-x_2; y=2x_1+2x_2+1; y=8x_1+2x_2+8; y=0.$$

Тому додамо ще дві залежності шляхом вибору команди **Add Mfs...** меню **Edit**. У діалоговому вікні, що з'явилося, в поле **Number of MFs** вибираємо **2** і натискаємо кнопку **ОК**.

**Крок 13.** Задамо найменування і параметри лінійних залежностей. Для цього робимо одне клацання лівою кнопкою миші по найменуванню першої залежності **mf1**. Потім друкуємо найменування залежності, наприклад **50**, у полі **Name**, і встановлюємо тип залежності – константа шляхом вибору опції **Constant** у меню **Type**. Після цього вводимо значення параметра – **50** у полі **Params**. Аналогічно для другої залежності **mf2** уведемо найменування залежності, наприклад **8+8x1+2x2**. Потім укажемо лінійний тип залежності шляхом вибору опції **Linear** у меню **Type** і введемо параметри залежності **8 2 8** у полі **Params**.

Для лінійної залежності порядок параметрів наступний: перший параметр – коефіцієнт при першій змінній, другий – при другій і т.д., і останній параметр – вільний член залежності.

Аналогічно для третьої залежності **mf3** уведемо найменування залежності, наприклад **1+2x1+2x2**, укажемо лінійний тип залежності і введемо параметри залежності **2 2 1**. Для четвертої залежності **mf4** уведемо найменування залежності, наприклад **4x1- x2**, укажемо лінійний тип залежності і введемо параметри залежності **4 -1 0**. Для п'ятої залежності **mf5** уведемо найменування залежності, наприклад **0**, укажемо тип залежності - константа і введемо параметр залежності **0**.

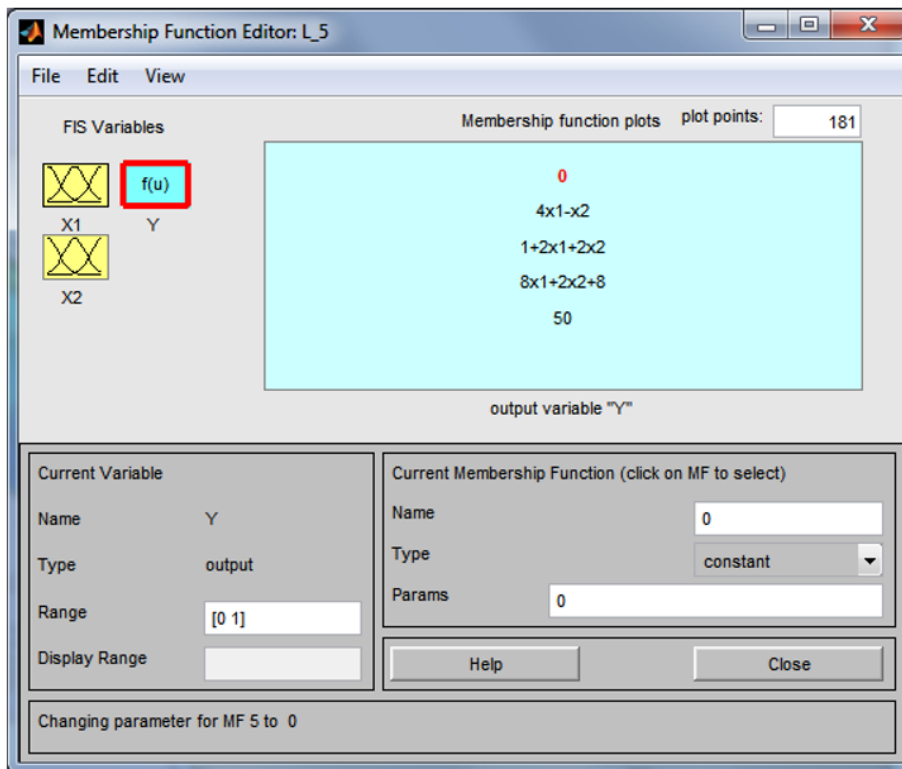


Рисунок 28 - Вікно лінійних залежностей «входи-вихіди»

**Крок 14.** Перейдемо в редактор бази знань **RuleEditor**. Для цього виберемо в меню **Edit** команду **Edit rules....** і введемо правила бази знань. Для введення правила необхідно вибрати відповідну комбінацію термів і залежностей і натиснути кнопку **Add rule** (рис.29).

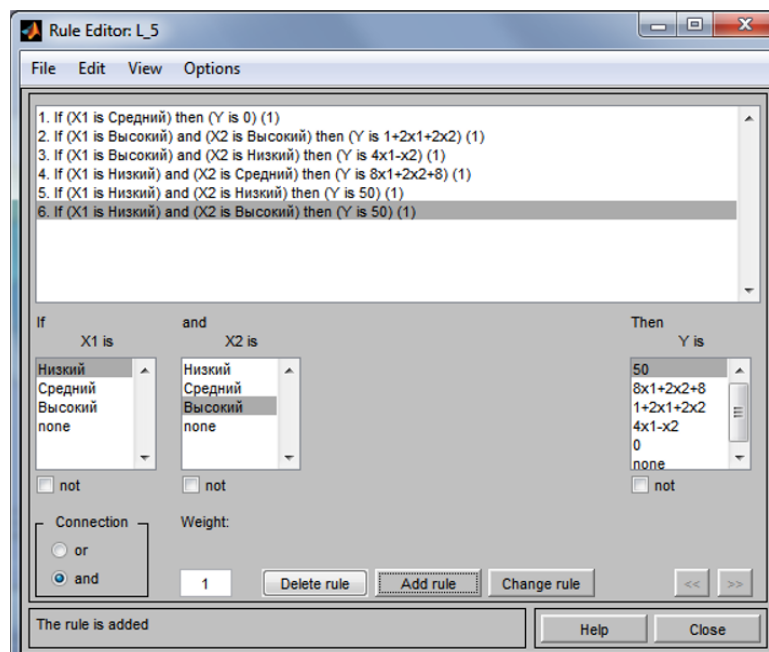


Рисунок 29 – Нечітка база знань для системи типу Сугено

Вікно візуалізації нечіткого логічного виведення активізується командою **View rules...** меню **View**. У поле **Input** указуються значення вхідних змінних, для

яких виконуються логічне виведення. Значення вихідної змінної розраховується як середнє зважене значення результатів виведення за кожним правилом (рис.30).

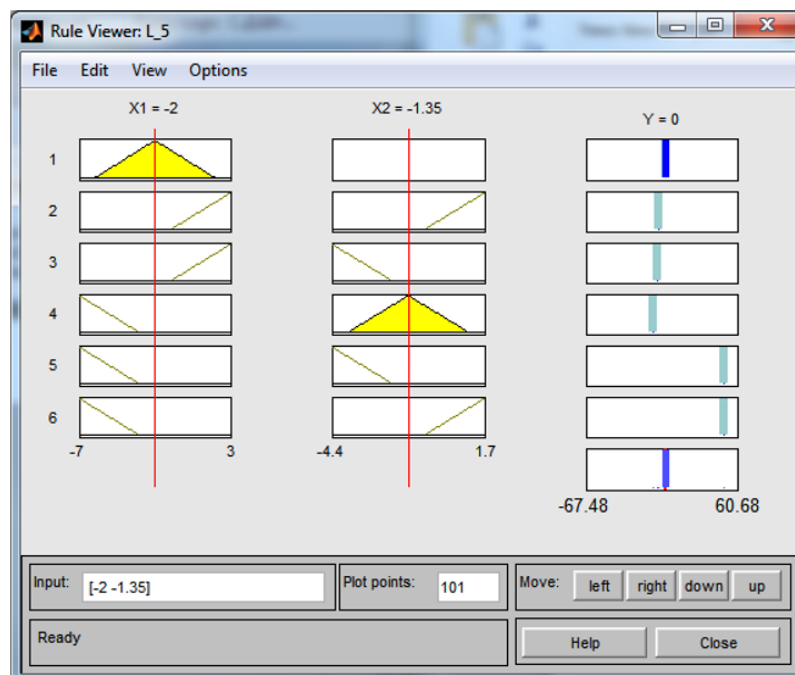


Рисунок 30 - Візуалізація нечіткого логічного виведення для системи типу Сугено

Для виведення вікна поверхні «вхід-вихід», що відповідає синтезованій нечіткій системі, необхідно використовувати команду **View surface...** меню **View** (рис.31).

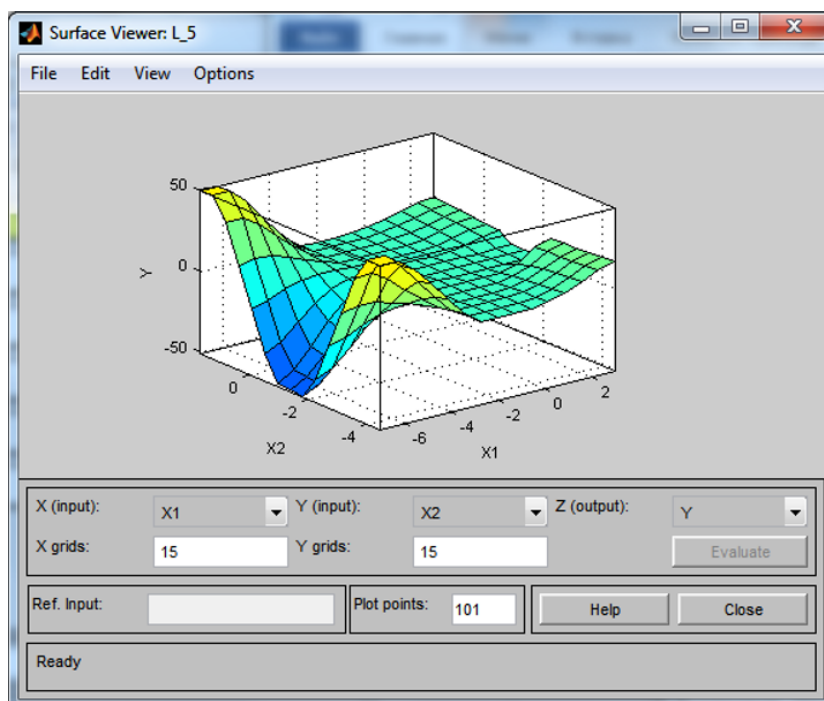


Рисунок 31 - Поверхня «входи-вихід» для системи типу Сугено



Порівнюючи еталонні поверхні (побудована в лабораторній роботі 4) і на рисунку 31 можна зробити висновок, що нечіткі правила досить добре описують складну нелінійну залежність. При цьому, модель типу Сугено більш точна. Перевага моделей типу Мамдані полягає в тому, що правила бази знань є прозорими і інтуїтивно зрозумілими, тоді як для моделей типу Сугено не завжди ясно які лінійні залежності «входи-вихід» необхідно використовувати.

### **Контрольні питання до захисту**

1. Для чого призначений FIS-редактор?
2. Якого типу є функції приналежності в пакеті fuzzy logic?
3. Яким чином можна зробити візуалізацію нечіткого логічного висновку?
4. Що таке візуалізація поверхні типу «входи-виходи»?
5. Назвіть ключові відмінності підходів проектування нечітких систем Мамдані та Сугено.

ЛАБОРАТОРНА РОБОТА №6  
ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ НА ОСНОВІ НЕЧІТКИХ  
ЗНАНЬ

**Мета:** Розробити нечітку модель інтелектуальної системи та дослідити її роботу.

**Теоретична частина**

Побудова нечіткої моделі інтелектуальної системи керування базується на формальному представленні характеристик об'єкту в термінах лінгвістичних змінних. В системі керування в якості лінгвістичних змінних розглядаються змінні входу і виходу системи. Ціль керування полягає у визначенні значень керуючих змінних (вхідних) змінних, реалізація яких забезпечує бажану поведінку чи бажаний стан об'єкту керування. Таким чином, нечітка модель системи керування може бути описана за допомогою апарату системи нечіткого виводу.

Для керування системами з вбудованими елементами штучного інтелекту в умовах невизначеності необхідно побудувати людино-машинний інтерфейс на основі якісних, вербальних категорій. Такі категорії визначаються в термінах нечітких множин. Для ефективного використання нечітких систем необхідно адекватно визначити нечіткі множини величин, побудувати правила виведення, правила агрегування виходів, здійснити перетворення чітких входів у нечіткі й нечітких виходів у чіткі. Забезпечити адекватність функцій належності нечітких величин можна їх динамічним підстроюванням за допомогою адаптивних методів.

Якість прийняття рішень в основному визначається базою нечітких правил. Такі правила визначаються експертним методом і тому можуть бути суб'єктивними, неповними або суперечливими. Подолання суперечливості правил і, відповідно, підвищення інтелектуального рівня системи нечіткого логічного виведення досягається поповненням і вдосконаленням бази правил у процесі навчання.

Розглянемо приклад того, як обробляються нечіткі правила виведення в експертній системі, що управляє вентилятором кімнатного кондиціонера. Завдання кондиціонера - підтримувати оптимальну температуру повітря в кімнаті, охолоджуючи його, коли жарко, і нагріваючи, коли холодно. Нехай, змінюючи швидкість обертання вентилятора, що проганяє повітря через охолоджуючий елемент, ми можемо змінювати температуру повітря, тоді алгоритм роботи кондиціонера може бути заданий наступними правилами:

- 1) якщо температура повітря в кімнаті висока, то швидкість обертання вентилятора висока;
- 2) якщо температура повітря в кімнаті середня, то швидкість обертання вентилятора середня;
- 3) якщо температура повітря в кімнаті низька, то швидкість обертання вентилятора низька.

Для того щоб система могла обробляти ці правила, треба налаштувати установки приналежності для нечітких підмножин, визначених на значеннях температури ( $t$ ) і швидкості обертання вентилятора ( $v$ ). Нехай температура повітря в кімнаті знаходиться в межах від  $0^\circ\text{C}$  до  $60^\circ\text{C}$  - в іншому випадку кондиціонер навряд чи допоможе. Функцію приналежності для нечіткого підмножини «низька», певну на інтервалі зміни температури, можна задати, наприклад, так (рис.32):

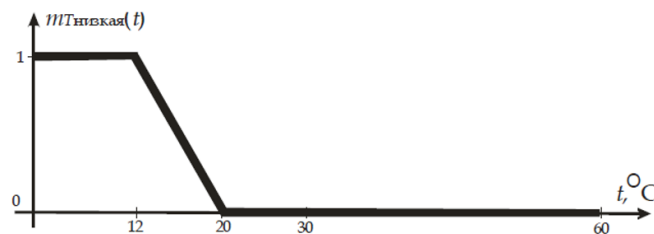


Рисунок 32 – Нечітка множина «низька», визначена на множині значень температури

Якщо температура менше  $12^\circ\text{C}$ , то це - напевно низька температура для кімнати. Температуру вище  $20^\circ\text{C}$  ніяк не можна назвати низькою. У проміжку між цими значеннями функція приналежності лінійно убиває - зі збільшенням температури зменшується істинність твердження «температура повітря в кімнаті низька». Аналітично виражається наступним чином:

$$m_{T_{низька}}(t) = \begin{cases} 1, & t \leq 12 \\ \frac{20-t}{8}, & 12 < t < 20 \\ 0, & t \geq 20 \end{cases}$$

Подібні міркування дозволяють нам налаштувати установки приналежності для решти множин: середня (рис.6.2) і висока (рис.33).

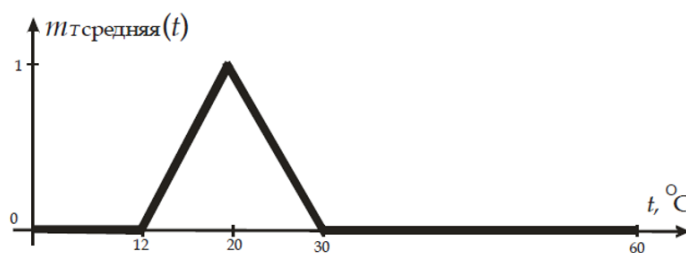


Рисунок 33 – Нечітка множина «середня», визначена на множині значень температури

$$m_{T_{\text{середня}}}(t) = \begin{cases} 0, t < 12 \text{ или } t \geq 30 \\ \frac{t-12}{8}, 12 \leq t < 20 \\ \frac{30-t}{10}, 20 \leq t < 30 \end{cases}$$

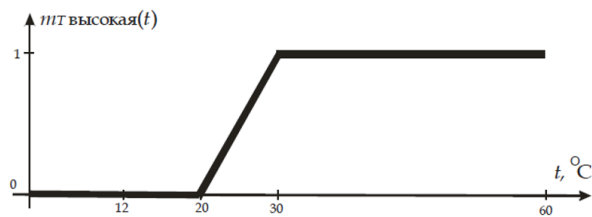


Рисунок 34 – Нечітка множина «висока», визначена на множині значень температури

$$m_{T_{\text{високая}}}(t) = \begin{cases} 0, t < 20 \\ \frac{t-20}{10}, 20 \leq t < 30 \\ 1, t > 30 \end{cases}$$

Визначимо нечіткі підмножини для швидкості обертання вентилятора. Нехай вона може змінюватися від 0 до 1000 об / хв. Цілком допустимим буде наступний варіант визначення функцій приналежності для нечітких підмножин низька, середня і висока (рис. 35, 36, 37).

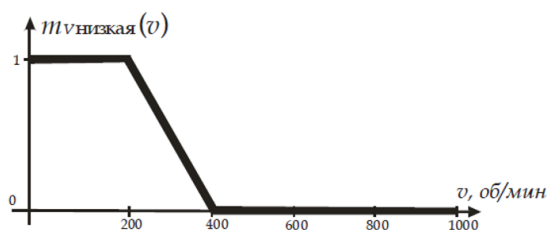


Рисунок 35 – Нечітка множина «низька», визначена на множині значень швидкості обертання вентилятора кондиціонера

$$m_{V_{\text{низкая}}}(v) = \begin{cases} 1, v < 200 \\ \frac{400 - v}{200}, 200 \leq v \leq 400 \\ 0, v > 400 \end{cases}$$

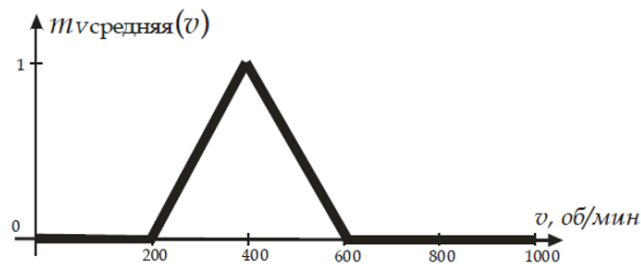


Рисунок 36 – Нечітка множина «середня», визначена на множині значень швидкості обертання вентилятора кондиціонера

$$m_{V_{\text{средняя}}}(v) = \begin{cases} 0, v < 200 \text{ или } v > 600 \\ \frac{v - 200}{200}, 200 \leq v < 400 \\ \frac{600 - v}{200}, 400 \leq v \leq 600 \end{cases}$$

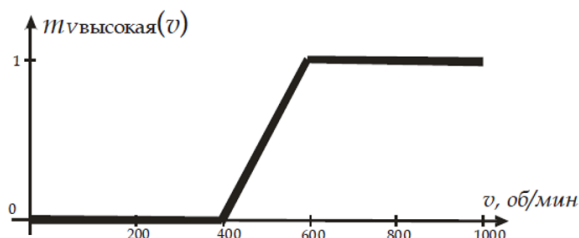


Рисунок 37 – Нечітка множина «висока», визначена на множині значень швидкості обертання вентилятора кондиціонера

$$m_{V_{\text{высокая}}}(v) = \begin{cases} 0, v < 400 \\ \frac{v - 400}{200}, 400 \leq v < 600 \\ 1, v \geq 600 \end{cases}$$

Розглянемо тепер, як нечітка експертна система визначає швидкість обертання вентилятора в залежності від температури повітря в кімнаті. Нехай ця температура дорівнює 22°C. Спочатку експертної системі треба визначити істинність лівих частин правил виведення при підстановці в них поточного значення температури. Для цього вона повинна знайти ступінь входження  $t = 22^\circ \text{C}$

в кожну із зазначених зліва нечітких підмножин. У лівих частинах правил вказані три підмножини, заданих на інтервалі значень температури: висока, низька і середня. Ступінь входження знаходимо, обчислюючи значення функцій приналежності кожного з підмножин від  $t = 22^\circ \text{C}$ :

$$\begin{aligned} m_{T \text{ высокая}}(22) &= 0.2; \\ m_{T \text{ средняя}}(22) &= 0.8; \\ m_{T \text{ низкая}}(22) &= 0. \end{aligned}$$

Значення істинності лівій частині кожного правила використовуються для модифікації нечіткої множини, зазначеного в його правій частині. На рис.38 зображено, як трансформуються множини, які знаходяться в правих частинах правил.

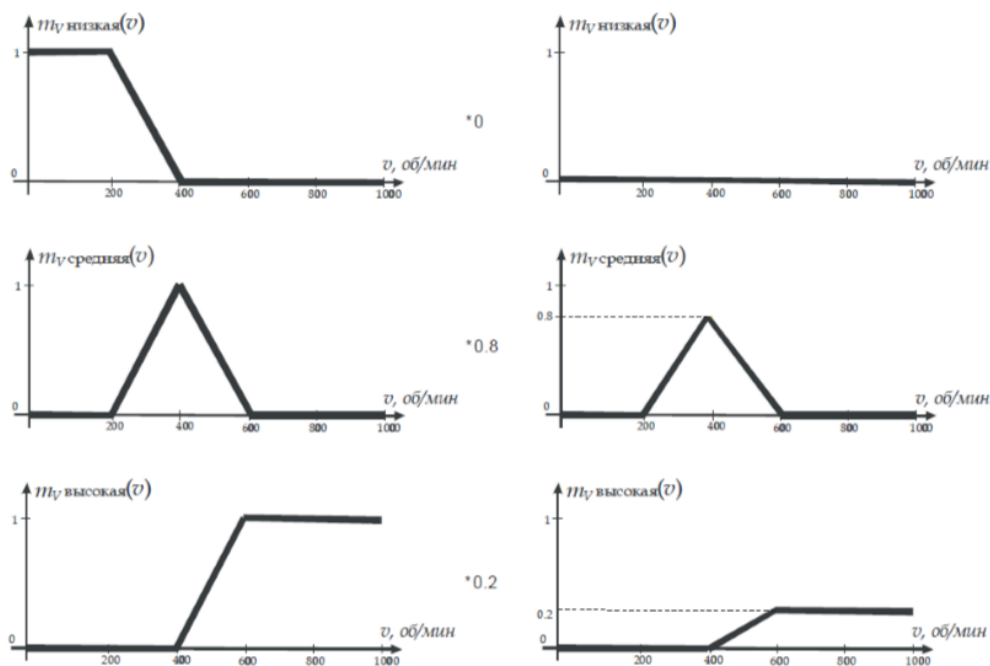


Рисунок 38 - Модифікація нечітких підмножин, визначених на інтервалі зміни швидкості обертання вентилятора

Далі нечіткої експертної системі необхідно узагальнити результати дії всіх правил виведення, тобто зробити суперпозицію отриманих нечітких множин. Результат об'єднання нечітких множин показаний на рис 39.

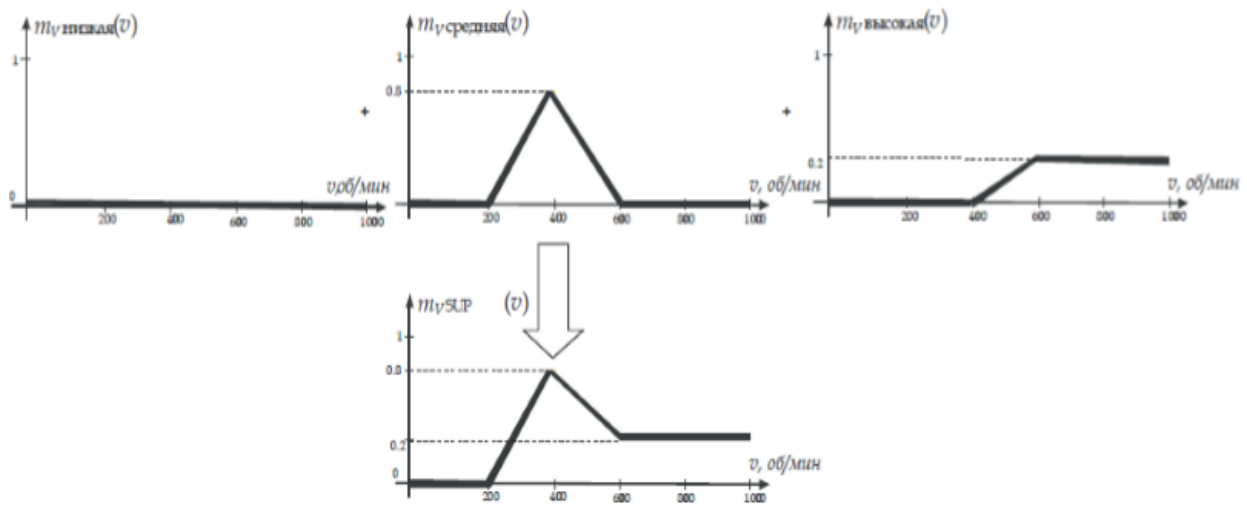


Рисунок 39 – Результат суперпозиции нечетких множеств

Тепер необхідно здійснити перехід від суперпозиції множин до скалярного значення. Скаляризації зробимо методом "центра ваги". Ілюстрація того, як виходить результат, представлена на рис.40.

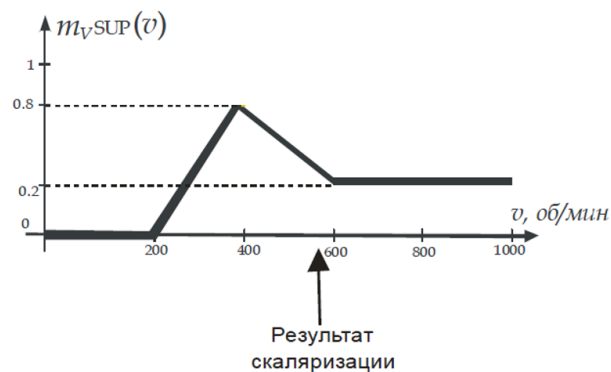


Рисунок 40 - Отримання скалярного значення швидкості обертання вентилятора методом «центру ваги»

Центр ваги фігури на рис.40 знаходиться в точці  $v = 520$ . Це і буде значенням швидкості обертання вентилятора, яке видасть експертна система при температурі повітря в кімнаті дорівнює  $22^{\circ} \text{C}$ . При інших значеннях температури функція приналежності узагальненого результату виконання всіх правил, зображена на рис.40, буде змінюватися.

## ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Задання нечітких функцій приналежності. У прикладному пакеті Fuzzy logic toolbox програми MATLAB створимо новий проект і поставимо нечіткі функції приналежності для температури як вхідні параметри (рис.41).

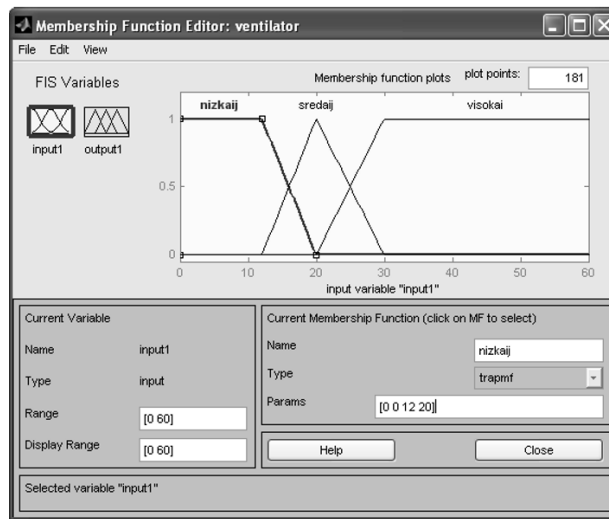


Рисунок 41 - Вікно завдання нечітких функцій приналежності для значень температури

Задамо нечіткі функції приналежності для швидкості обертання вентилятора як вхідні параметри (рис.42).

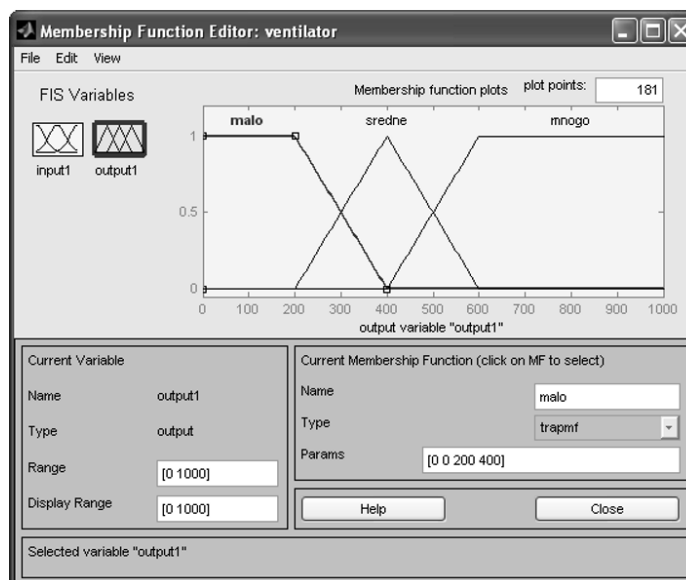


Рисунок 42 - Вікно завдання нечітких функцій приналежності для значень швидкості обертання вентилятора

- Задання правил виведення. Правила виведення в створеній комп'ютерній моделі нечіткі правила виведення задаються за допомогою вкладки Edit → Rules. Вікно редактора після завдання правил виведення матиме вигляд, представлений на рис. 43.



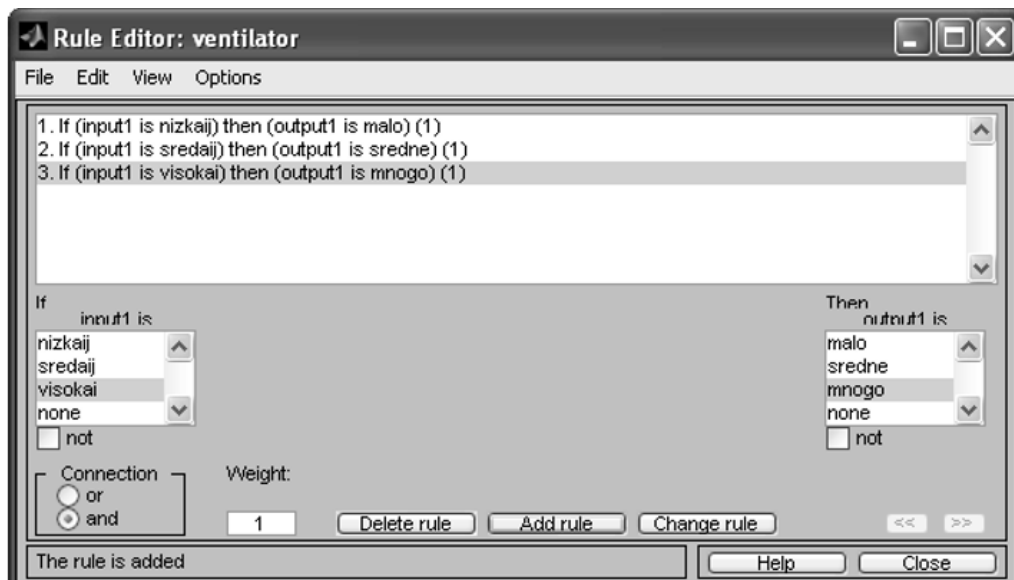


Рисунок 43 – Задання правил виведення

3. Отримання відгуку системи. Послідовність обробки нечітких знань для конкретного значення температури можна переглянути у вікні перегляду правил View → Rules. На рис.44 представлена послідовність обробки нечітких знань для температури  $t = 22^{\circ} \text{C}$ , при цьому вентилятор кондиціонера повинен обертатися зі швидкістю  $v = 520$  об / хв.

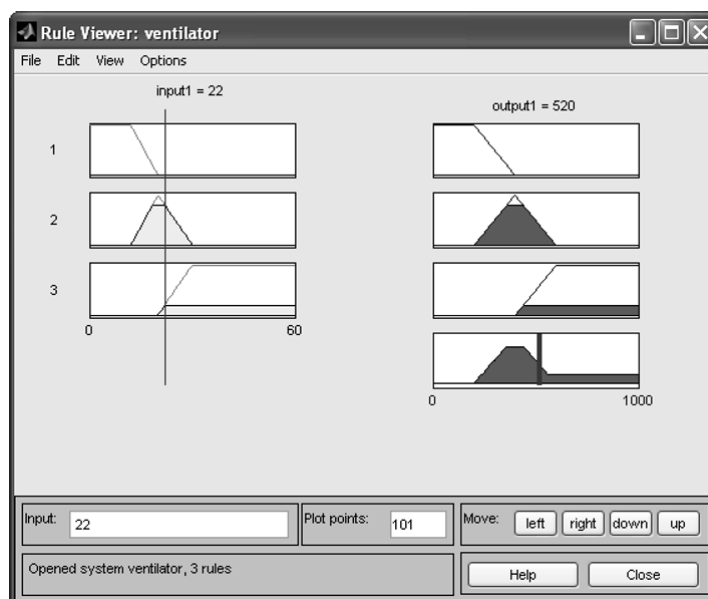


Рисунок 44 – Обробка нечітких знань в експертній системі

Залежність швидкості обертання кондиціонера від температури для даної моделі можна переглянути за допомогою команди View → Surface (рис.45).

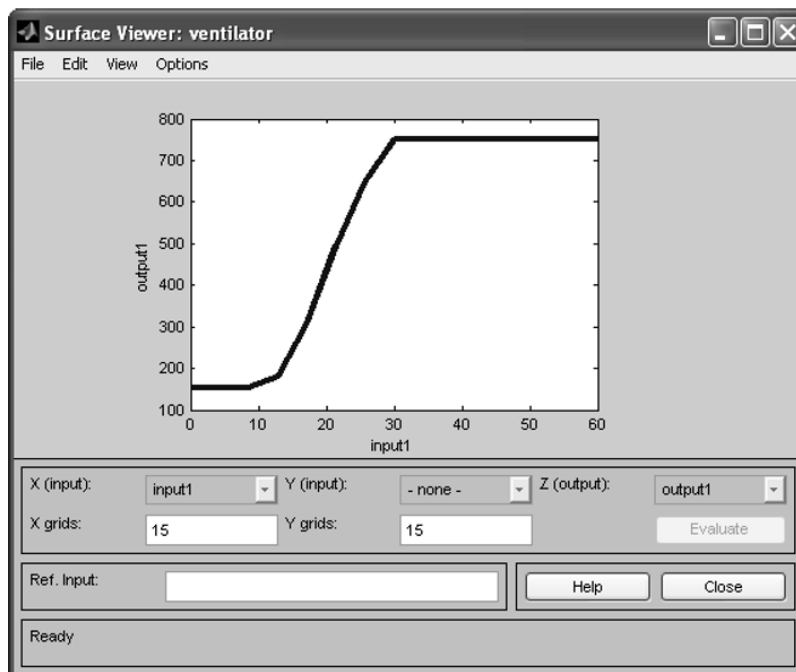


Рисунок 45 – Відгук системи

### Індивідуальні завдання

#### **Варіант 1. Побудова системи нечіткого керування процесом прийому на роботу у фірму нових співробітників.**

Нехай, у фірми є дві вакансії: програміст і системний адміністратор. Тому система нечіткого виведення має дві вихідні лінгвістичні змінні  $U_1, U_2$  з множинами-носіями у вигляді відрізків  $[0,1]$ , які інтерпретуються як імовірність прийняття деякої особи на роботу відповідно на першу чи другу посаду.

Вхідні змінні мають характеризувати особу, яка подала резюме про себе в фірми з метою влаштуватися на роботу. Наприклад,  $X_1$  - досвід роботи в сфері комп'ютерних технологій,  $X_2$  - освіта,  $X_3$  - ділові якості спілкування.

Менеджер по кадрам даної фірми розробив такі прості правила прийому на роботу:

1. Якщо досвід роботи високий і професійна освіта висока, то ймовірність прийняття на посаду програміста висока
2. Якщо досвід роботи високий і ділові якості гарні, то ймовірність прийняття на посаду системного адміністратора висока
3. Якщо досвід роботи і освіта високі, а ділові якості гарні, то ймовірність прийняття на роботу даної особи висока на обидві посади
4. Якщо досвід роботи невеликий, але освіта та ділові якості високі, тоді можливо прийняти на роботу таку особу в якості системного адміністратора

5. Якщо особа не має освіти і досвіду роботи, тоді прийняття її на роботу малоймовірно на обидві посади

6. Якщо професійна освіта висока, але досвід роботи і ділові якості невеликі тоді можна прийняти цю особу на посаду програміста і малоймовірно її прийняти на посаду системного адміністратора

7. Якщо освіти у особи немає, тоді малоймовірно її прийняття на роботу

8. Якщо особа не має досвіду роботи і її ділові якості не високі, тоді малоймовірно її прийняття на роботу на обидві посади

9. Якщо освіта особи висока, досвід роботи невеликий, то можливо прийняття її на роботу на посаду програміста

10. Якщо особа не має освіти і ділових якостей, то прийняття її на роботу малоймовірно на обидві посади

11. Якщо особа не має освіти, досвід і її ділові якості невисокі, тоді малоймовірно її прийняття на роботу на обидві посади

12. Якщо ділові якості особи невеликі, освіта професійна і досвід роботи високі, тоді малоймовірно її прийняти на посаду системного адміністратора і можливо її прийняти на посаду програміста.

Для розробки системи слід виконати наступні кроки.

**Крок 1.** Визначити вхідні і вихідні змінні. Очевидно, що для СНВ у якості вхідних змінних потрібно взяти

- 1) досвід роботи  $x_1$  з множиною-носієм від 0 до 30 років;
- 2) освіти особи  $x_2$  з множиною-носієм, наприклад,  $[0,10]$ ;
- 3) ділові якості  $x_3$  з множиною-носієм  $[0,1]$ .

Далі слід задати терми цих змінних. Наприклад, для вхідної змінної досвіду роботи можуть бути задані наступні терми: «немає», «невеликий», «високий»

**Крок 2.** Фазифікація вхідних змінних. На цьому кроці слід задати функції належності для всіх термів вхідних змінних, а в якості області визначення – їх множини-носії

**Крок 3.** Задати функції належності термів вихідних змінних (ймовірність прийняття на посаду)

**Крок 4.** Ввести правила у базу правил.

**Крок 5.** Використання моделі. Для цього розглянути приклад роботи системи керування при різних значеннях вхідної змінної. Для цього слід відкрити вікно правил *View* → *Rules* і переглянути можливі значення вихідної змінної прийняття рішення про прийом на роботу в залежності від зміни значень вхідних змінних.

## **Варіант 2. Побудова нечіткої моделі керування кранами гарячої і холодної води при прийнятті душу**

При прийнятті душу на вхід змішувача подається холодна та гаряча вода по відповідним трубопроводам. Для комфортного прийняття душа, користувач задає на виході бажану комфортну теплу температуру і бажаний напір води. Оскільки, під час прийняття душу спостерігається нерівномірне використання води і температура на виході змішувача буде весь час коливатися, виникає необхідність ручного керування кранами відкриття гарячої чи холодної води. Задача полягає у створенні системи нечіткого виведення, яка б дозволила автоматизувати цей 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. Тоді досвід прийняття душа дозволяє сформулювати декілька евристичних правил.

1. Якщо вода гаряча і її напір сильний, тоді необхідно повернути кран гарячої води на середній кут вліво, а кран холодної води на середній кут вправо

2. Якщо вода гаряча і її напір не дуже сильний, слід повернути кран холодної води на середній кут вправо

3. Якщо вода не дуже гаряча і її напір сильний, тоді необхідно повернути кран гарячої води на невеликий кут вліво

4. Якщо вода не дуже гаряча і її напір слабкий, тоді слід повернути крани гарячої і холодної води на невеликий кут вправо

5. Якщо вода тепла і її напір не дуже сильний, тоді слід залишити кран змішувача в своєму положенні

6. Якщо вода прохолодна і її напір сильний, тоді необхідно повернути кран гарячої води на середній кут вправо, а кран холодної води на середній кут вліво

7. Якщо вода прохолодна і її напір не дуже сильний, тоді слід повернути кран гарячої води на середній кут вправо, а кран холодної води на невеликий кут вліво

8. Якщо вода холодна і її напір слабкий, тоді слід повернути кран гарячої води на великий кут вправо

9. Якщо вода холодна і її напір сильний, тоді слід повернути кран гарячої води на середній кут вліво, а кран гарячої води на середній кут вправо

10. Якщо вода тепла і її напір сильний, тоді слід повернути крани гарячої і холодної води на невеликий кут вліво .

11. Якщо вода тепла і її напір слабкий, тоді слід повернути крани гарячої і холодної води на невеликий кут вправо .

Для розробки системи слід виконати наступні кроки.

**Крок 1.** Визначити вхідні і вихідні змінні. Очевидно, що для СНВ у якості вхідних змінних потрібно взяти температуру води на виході змішувача і її напір. У якості вихідних змінної слід взяти кути повороту кранів гарячої і холодної води.

Далі слід задати терми цих змінних. Наприклад, для вхідної змінної температури – «вода гаряча», «вода не дуже гаряча», «вода тепла», «вода прохолодна», «вода холодна»

**Крок 2.** Фазифікація вхідних змінних. На цьому кроці слід задати функції належності для всіх термів вхідних змінних, а в якості області визначення – інтервал можливої температури води і інтервал кількісної оцінки напору. Наприклад, відповідно, в градусах Цельсія від 0 до 100 і в нормованих одиницях – від 0(напору немає) до 1(напір дуже сильний).

**Крок 3.** Задати функції належності термів вихідної змінної (кута повороту крана гарячої води) з інтервалом в області визначення від -90 до 90 гр.

**Крок 4.** Ввести правила у базу правил.

**Крок 5.** Використання моделі. Для цього розглянути приклад роботи системи керування при різних значеннях вхідної змінної.

### **Варіант 3. 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 системи нечіткого виведення, яка б змогла автоматизувати роботу кондиціонера при коливанні температури приміщення через різні зовнішні дестабілізуючі фактори.

Досвід використання побутових кондиціонерів показує деяку інертність в процесі нагріву чи охолодження повітря. Наприклад, після включення режиму «холод», відбувається нагнітання холодного повітря, через що температура в приміщенні поступово спадає.

При цьому, при виключенні цього режиму, температура все рівно деякий час продовжує знижуватися. Аналогічна картина спостерігається при включенні режиму «тепло». Щоб врахувати цю властивість, потрібно задати як вхідну змінну не тільки температуру приміщення, але і швидкість її зміни. В такому випадку, досвід показує адекватність наступних правил керування кондиціонеру:

1. Якщо температура повітря дуже тепла і швидкість зміни температури додатня, то потрібно включити режим «холод», повернувши регулятор кондиціонеру на великий кут вліво.

2. Якщо температура повітря дуже тепла, а швидкість зміни температури від'ємна, тоді необхідно включити режим «холод», повернувши регулятор кондиціонеру на невеликий кут вліво.
3. Якщо температура повітря тепла, а швидкість зміни температури додатня, тоді потрібно включити режим «холод», повернувши регулятор кондиціонеру на великий кут вліво.
4. Якщо температура повітря тепла, а швидкість зміни температури від'ємна, тоді потрібно включити режим «холод», повернувши регулятор кондиціонеру слід вимкнути.
5. Якщо температура повітря дуже холодна, а швидкість зміни температури від'ємна, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонеру на великий кут вправо.
6. Якщо температура повітря дуже холодна, а швидкість зміни температури додатня, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонеру на невеликий кут вправо.
7. Якщо температура повітря холодна, а швидкість зміни температури від'ємна, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонеру на великий кут вліво.
8. Якщо температура повітря холодна, а швидкість зміни температури додатня, тоді потрібно виключити кондиціонер.
9. Якщо температура повітря дуже тепла, а швидкість зміни температури дорівнює 0, тоді потрібно включити режим «холод», повернувши регулятор кондиціонеру на великий кут вліво.
10. Якщо температура повітря тепла, а швидкість зміни температури дорівнює 0, тоді потрібно включити режим «холод», повернувши регулятор кондиціонеру на невеликий кут вліво.
11. Якщо температура повітря дуже холодна, а швидкість зміни температури дорівнює 0, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонеру на великий кут вправо.
12. Якщо температура повітря холодна, а швидкість зміни температури дорівнює 0, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонеру на невеликий кут вправо.
13. Якщо температура повітря в нормі, а швидкість зміни температури додатня, тоді потрібно включити режим «холод», повернувши регулятор кондиціонеру на невеликий кут вліво.

14. Якщо температура повітря в нормі, а швидкість зміни температури від'ємна, тоді потрібно включити режим «тепло», повернувши регулятор кондиціонера на невеликий кут вправо.
15. Якщо температура повітря в нормі, а швидкість зміни температури дорівнює 0, тоді потрібно виключити кондиціонер.
16. Тому, для розробки системи слід виконати наступні кроки.

**Крок 1.** Визначити вхідні і вихідні змінні. Очевидно, що для СНВ у якості вхідних змінних потрібно взяти температуру повітря в приміщенні та швидкість її зміни. У якості вихідної змінної слід взяти кут повороту регулятора кондиціонера.

Далі слід задати терми цих змінних. Наприклад, для 1 вхідної змінної – «повітря дуже холодне», «повітря холодне», «повітря в нормі», «повітря тепле», «повітря дуже тепле»

**Крок 2.** Фазифікація вхідних змінних. На цьому кроці слід задати функції належності для всіх термів вхідної змінної, а в якості області визначення – інтервал можливої температури повітря та її швидкості. Наприклад, в градусах Цельсія від -10 до 40.

**Крок 3.** Задати функції належності термів вихідної змінної (кута повороту регулятора кондиціонера) з інтервалом в області визначення від -90 до 90 гр.

**Крок 4.** Ввести правила у базу правил.

**Крок 5.** Використання моделі. Для цього розглянути приклад роботи системи керування при різних значеннях вхідної змінної.

#### **Варіант 4. Нечітка модель керування контейнерним краном.**

Контейнерні крани використовуються при виконанні вантажних робіт в портах. Крани тросом підіймають контейнер з вантажем і далі ідуть на спеціальних рейсах до місця призначення і там розвантажуються. Коли контейнер піднімається до крану і кран починає рухатися по рейсам, вантаж починає розхитувати і відхилятися від свого строго вертикального положення під краном. Проблема полягає в тому, що допоки контейнер хитається, він не може бути опущений на платформу цілі, в якості яких використовуються деякі транспортні засоби.

Аналіз дій крановиків-операторів, які виконують дії по управлінню краном, показує, що вони в своїй роботі використовують наступні евристичні правила:

Почати роботу, потрібно, коли контейнер вже завантажено грумом і виконувати наступні правила:

1. Якщо контейнер знаходиться на горизонталі завантаження чи цілі (наприклад, це позиція 0), тоді слід підняти контейнер на рівень руху (наприклад, позиція 1).

2. Починати рух потрібно зі середньою потужністю
3. Якщо рух вже почався, кабіна знаходиться далеко від цілі і контейнер несильно хитається, тоді потрібно задати потужність руху велику
4. Якщо рух вже почався, кабіна знаходиться далеко від цілі і контейнер сильно хитається, тоді потрібно задати потужність руху невелику
5. Якщо кабіна знаходиться близько від цілі, слід зменшити швидкість руху до малої
6. Якщо контейнер знаходиться дуже близько до цілі, тоді слід виключити потужність двигуна
7. Коли контейнер знаходиться прямо над позицією цілі, тоді слід зупинити двигун.
8. Якщо контейнер знаходиться над ціллю, але хитання великі, тоді слід не опускати контейнер на ціль і чекати
9. Якщо контейнер знаходиться над ціллю, він не хитається, тоді слід опускати його
10. Якщо контейнер зустрівся з ціллю, тоді слід зупинити його опускання

**Крок 1.** Визначити вхідні і вихідні змінні. Очевидно, що для СНВ у якості вхідних змінних потрібно взяти відстань до цілі, кут хитання контейнеру від строго вертикальної позиції під кабіною крана і вертикальну відстань контейнеру від цілі.. У якості вихідних змінних слід взяти потужність двигуна крана при русі рейсами і швидкість опускання, чи піднімання крана.

Далі слід задати терми цих змінних. Наприклад, для 1 вхідної змінної – «відстань далека», «відстань середня», «відстань близька», «позиція над ціллю», для 2 вхідної змінної – «кут=0», «кут хитання малий», «кут хитання великий», для 3 вхідної змінної – «контейнер в позиції 0», «контейнер в позиції 1», для 1 вихідної змінної – «потужність середня», «потужність велика», «потужність мала», «потужність нульова», для 2 вихідної змінної – «піднімати контейнер», «опускати контейнер», «не змінювати вертикальну позицію контейнера»

**Крок 2.** Фазифікація вхідних змінних. На цьому кроці слід задати функції належності для всіх термів вхідної змінної, а в якості області визначення – інтервал можливих значень.

**Крок 3.** Задати функції належності термів вихідних змінних.

**Крок 4.** Ввести правила у базу правил.

**Крок 5.** Використання моделі. Для цього розглянути приклад роботи системи керування при різних значеннях вхідної змінної.



### **Контрольні питання до захисту**

1. Назвіть і охарактеризуйте етапи здійснення нечіткого логічного виведення?
2. Що таке фазифікація?
3. Яким набором параметрів характеризується лінгвістична змінна?
4. У чому відмінність нечіткої змінної від лінгвістичної?
5. Назвіть логічні зв'язки.
6. Наведіть приклади лінгвістичних змінних.
7. Які завдання необхідно вирішувати, використовуючи апарат нечіткої логіки?

## ЛІТЕРАТУРА

### Основна:

1. Спеціальні розділи математики: конспект лекцій [Текст] / В.І.Вербицький. – Х.: «ХНАДУ», 2017. – 45 с. (електр.вид)
2. Інтелектуальні системи керування: конспект лекцій [Текст] / В. О. Апостолук, О. С. Апостолук. – К.: НТУУ «КПІ», 2008. – 88 с. (електр.вид)
3. Нечеткая логика и нейронные сети: учебное пособие / сост. Р.Г. Асадуллаев. – Белгород, 2017. – 309 с. (електр.вид)

### Додаткова:

1. Основы теории нечетких множеств : учеб. пособие / В.Г. Чернов ; Владим. гос. ун-т.- Владимир : Изд-во Владим. гос. ун-та, 2010. – 96 с. – ISBN 978-5-9984-0055-1.
2. Заде, Л. А. Понятие лингвистической переменной и его применение к принятию приближенных решений / Л.А. Заде: пер. с англ. – М.: Мир, 1976. – 165 с.
3. Борисов, А. Н. Модели принятия решений на основе лингвистической переменной / А.Н. Борисов [и др.]. – М.: Знание, 1982. – 170 с.