МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОДЗЕРЖИНСЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання практичних робіт з дисципліни
«Нові науково-технічні напрямки електронних систем»
для студентів спеціальності
8.090803 «Електроні системи»

Затверджено редакційно-видавничою
секцією науково-методичної ради ДДТУ
_____ 2012 р. протокол №____

Дніпродзержинськ
2012

Методичні вказівки до виконання практичних робіт з дисципліни «Нові науково-технічні напрямки електронних систем» для студентів спеціальності 8.090803 «Електронні системи»/ Укл.: к.т.н. доцент Багрій В.В., Дніпродзержинськ ,ДДТУ, 2012, ____с.

Укладач: к.т.н. доцент, Багрій В.В.
Відповідальний за випуск: д.т.н., проф. Бойко В.І.
Рецензент: д.т.н., проф. Довгалюк Б.П.

Затверджено на засіданні кафедри
електроніки і автоматики
протокол №___від_____

Коротка анотація видання. У методичних вказівках викладено ціль, задачі та порядок виконання практичних робіт до вивчення принципів побудови систем нечіткого виводу. В роботах наведено короткі відомості по теорії нечітких множин, завдання на практичну роботу і порядок її виконання за допомогою середовища MATLAB. Вказівки призначені для магістрів денної форми навчання спеціальності 8.090803 «Електронні системи».

Вступ

Нині спостерігається інтенсивний розвиток і практичного застосування нечітких систем для цілей управління і регулювання багатьох технічних об'єктів. Системи, засновані на нечіткій логіці, розроблені і успішно впроваджені в таких областях, як управління технологічними процесами, управління транспортом, управління побутовою технікою, медична і технічна діагностика.

Багато сучасних завдань управління просто не можуть бути вирішені класичними методами із-за дуже великої складності математичних моделей, які їх описують.

Поняття "*fuzzy-logic*" (у перекладі з англійською - нечітка, розмита логіка) введене американським математиком JI.A. Заді (L.A.Zadeh). який запропонував теорію нечітких множин, на основі якої можна побудувати нечіткі аналоги усіх математичних понять і створити необхідний формальний апарат для моделювання людських міркувань і людського способу рішення завдань. Нечітка множина (*fuzzy sel*) - сукупність елементів довільної природи, відносно яких не можна з повною визначеністю стверджувати, - чи належить той або інший елемент даній сукупності або ні.

Перевагами нечіткої логіки, які явно проявляються в нечіткому управлінні, полягають передусім в тому, що нечітка логіка дозволяє вдало представити мислення людини, а саме способи прийнята рішень людиною, і способи моделювання складних об'єктів засобами природної мови.

В результаті вивчення курсу студент повинен:

- знати основні теоретичні і методологічні напрями моделювання на нечітких множинах і нейронних мережах, сфери його застосування;
- уміти формалізувати технічну проблему і уміти її сформулювати в термінах нечіткої логіки і/або нейронних мереж а так само запропонувати адекватні методи для її моделювання і аналізу;
- мати навички практичного застосування методів нечіткої логіки і мережевого моделювання із застосуванням спеціалізованого програмного забезпечення

Практичне заняття 1.
Дослідження способів формування нечітких множин з використанням різних типів функцій приналежності

Мета роботи: вивчити методи побудови нечітких множин з використанням різних типів функцій приналежності.

Початковим поняттям нечіткої логіки є поняття елементарного нечіткого висловлювання.

Елементарним нечітким висловлюванням називається розповідне речення, що виражає закінчену думку, відносно якого ми можемо судити про її істинність або помилковість тільки з деякою мірою упевненості.

У нечіткій логіці міра істинності елементарного нечіткого висловлювання набуває значення з відрізку [0, 1], причому 0 та 1 є граничними значеннями міри істинності і співпадають зі значеннями "неправда" та "істина" відповідно.

Нечітка множина є сукупність елементів довільної природи, відносно яких не можна з повною визначеністю стверджувати, - чи належить той або інший елемент до даної сукупності цій множині або ні. Проте усі дані елементи повинні належати універсальній (базовій) множині U.

Таким чином, нечітка множина $A = \{ < x, \mu_A(x) > \}$ - це множина пар виду $< x, \mu_A(x) >$, де $x \in U$, а $\mu_A(x)$ - функція приналежності, яка ставить у відповідність кожному з елементів $x \in U$ деяке дійсне число з відрізку [0,1].

З цього визначення видно, що універсальна множина U нечіткої множини A визначено як область визначення функції приналежності μ_A. При цьому, якщо $\mu_A(x) = 1$ для деякого $x \in U$, то елемент x явно належить нечіткій множині A, а якщо $\mu_A(x) = 0$, то це означає, що елемент x явно не належить нечіткій множині A. Якщо для всякого $x \in U$ $\mu_A(x) = 1$, то A вважають співпадаючими з U, а якщо для всякого $x \in U$ $\mu_A(x) = 0$, то A вважають співпадаючими з $\varnothing$. Тому нечітке пусте і нечітке універсальне множини можна рахувати формально визначеними і співпадаючими з їх чіткими класичними аналогами.

При побудові функцій приналежності з кожною нечіткою множиною зручно асоціювати деяку нечітку властивість, нечітку ознаку або нечіткий атрибут, які характеризують елементи універсальної множини. Методи побудови функцій приналежності діляться на прямі і непрямі.

Прямі методи побудови функцій приналежності. У прямих методах експерти просто задають для кожного $x \in U$ значення функції приналежності $\mu_A(x)$. Як

правило, прямі методи побудови функцій приналежності використовуються для таких властивостей, які можуть бути виміряні за допомогою деякої кількісної шкали. Процес побудови нечіткої множини на основі деякого відомого заздалегідь кількісного значення вимірної ознаки дістав назву фазифікації.

Непрямі методи застосовуються в тих випадках, коли відсутні очевидні вимірні властивості, які можуть бути використані для побудови нечітких моделей які розглядається в даній предметною області.

Інструментарій нечіткої логіки у складі пакету Matlab містить 11 вбудованих типів функцій приналежності (ФП), що формуються на основі кусково-лінійних функцій, розподілу Гауса, сигмоїдної кривої, квадратичних і кубічних поліноміальних кривих. До найбільш простих ФП можна віднести трикутну і трапецієвидну. Найменування трикутної ФП - trimf (triangle membership function). У параметричному виді вона є не що інше, як набір трьох точок, що утворюють трикутник.

Синтаксис функції :

y = trimf(*x*, params)

Функція trimf має два вхідні аргументи:

- *x* - вектор, для координат якого розраховується ступень приналежності;
- *params* - вектор параметрів функції приналежності.

Порядок завдання параметрів : [*a b c*]. Величини *a* і *c* задають основу трикутника, *b* - його вершину. Параметри функції приналежності повинні задовольняти умові $a<b<c$.

У аналітичному виді трикутна ФП може бути задана таким чином:

$$f(x,a,b,c)=\begin{cases}0, x<a,\\ \dfrac{x-a}{b-a}, a\le x<b,\\ \dfrac{c-x}{c-b}, b\le x<c,\\ 0, x>c.\end{cases}$$

Функція trimf повертає міри приналежності координат вектору *x*.

Програма використання ФП trimf:

```
x= 0: 0.5: 10;
yl = trimf(x, [0 0 10]);
y2 = trimf(x, [0 3 10]);
y3 = trimf(x, [0 7 10]);
plot(x,y1,'o-',x,y2,'+:',x,y3,'.-')
title('trimf, a=0, b=0...7, c=10');
legend('b=0','b=3','b=7')
```

Одержаний результат показано на рис. 1.1.

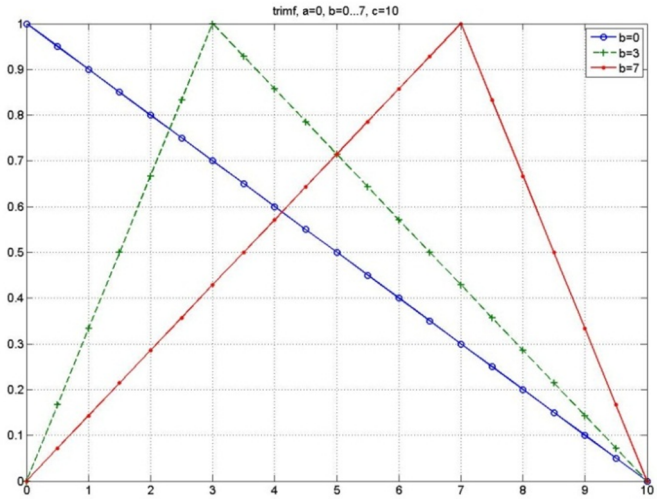


Рисунок 1.1- Трикутна ФП - trimf

Трапецієвидна ФП - trapmf (trapezoid membership function) - відрізняється від попередньої функції лише тим, що має верхню основу.

Синтаксис функції :

y = trapmf(*x, params*)

Функція trapmf має два вхідні аргументи:

- *x* – вектор для координат якого розраховується ступень приналежності;
- *params* - вектор параметрів функції приналежності.

Порядок завдання параметрів:[*abcd*]. Вони повинні задовольняти умові $a<b<c<d$. Параметри *a* і *d* - нижня основа трапеції; *b* і *c* - верхня основа трапеції.

Параметри трапецієвидної функції приналежності інтерпретуються так:

[*a, d*] - носій нечіткої множини (песимістична оцінка значення змінної);

[*b, c*] - ядро нечіткої множини (оптимістична оцінка значення змінної).

Функція trapmf застосовується для завдання асиметричних функцій приналежності нечітких множин, найбільш можливі значення яких досягаються на деякому інтервалі.

Аналітичний запис трапецієвидної функції має вигляд:

$$f(x,a,b,c,d)=\begin{cases}0, x<a,\\ \dfrac{x-a}{b-a}, a\le x<b,\\ 1, b<x\le c,\\ \dfrac{d-x}{d-c}, c<x\le b,\\ 0, x>d.\end{cases}$$

Функція trapmf повертає міри приналежності координат вектору *x*.

Програма використання ФП trapmf:

```
x = 0: 0.5: 10;
yl = trapmf(x, [0 0 1.5 10]);
y2 = trapmf(x, [0 2 3 10]);
y3 = trapmf(x, [0 4 7 10]);
plot(x, yl, 'o-', x, y2,'+:',x,y3, '.-')
title('trapmf, a=0, d=10')
legend('b=0, c=1.5', 'b=2, c=3', 'b=4, c=7')
```

Одержаний результат показано на рис. 1.2.

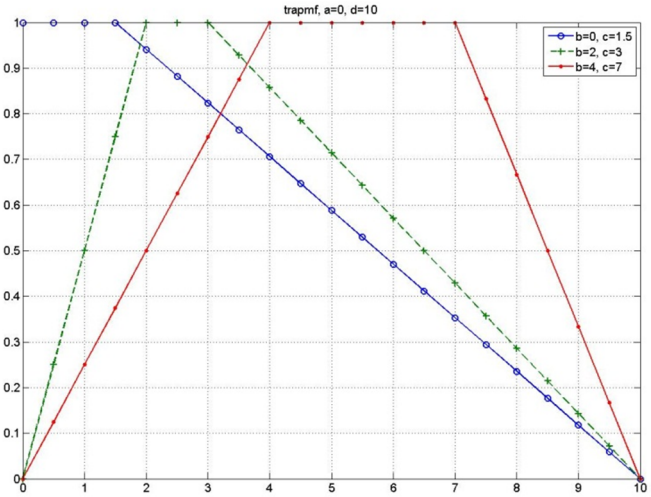


Рисунок 1.2- Трапецієвидна ФП - trapmf

Одне з основних переваг трикутних і трапецієвидних ФП - їх простота. На основі функції розподілу Гауса можна побудувати ФП двох видів : просту функцію приналежності Гауса і двосторонню, утворену за допомогою різних функцій розподілу Гауса.

Перша з них позначається gaussmf, а друга - gauss2mf.

Проста функція приналежності Гауса, *y* = gaussmf(*x, params*), задає функцію приналежності у вигляді симетричної гаусової кривої згідно з формулою:

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

Параметри функції приналежності геометрично інтерпретуються таким чином:

b - координата максимуму функції приналежності;

c - коефіцієнт концентрації функції приналежності.

Функція gaussmf застосовується для завдання гладких симетричних функцій приналежності. Функція gaussmf має два вхідні аргументи:

x - вектор, для координат якого розраховується ступені приналежності;

params - вектор параметрів функції приналежності.

Порядок завдання параметрів : [*c b*].

Функція gaussmf повертає міри приналежності координат вектору *x*.

Програма використання ФП gaussmf:

```
x = 0:0.5:10;
y1 = gaussmf(x,[0.5 5]);
y2 = gaussmf(x,[1 5]) ;
y3 = gaussmf(x,[2 5]);
y4 = gaussmf(x,[3 5]);
plot (x, y1, 'o-', x, y2,'+:',x, y3,'.-',x,y4,'-^')
title ('gaussmf,b=5, c=0.5...3');
legend('c=0.5','c=1', 'c=2', 'c=3')
```

Одержаний результат показано на рис. 1.3.

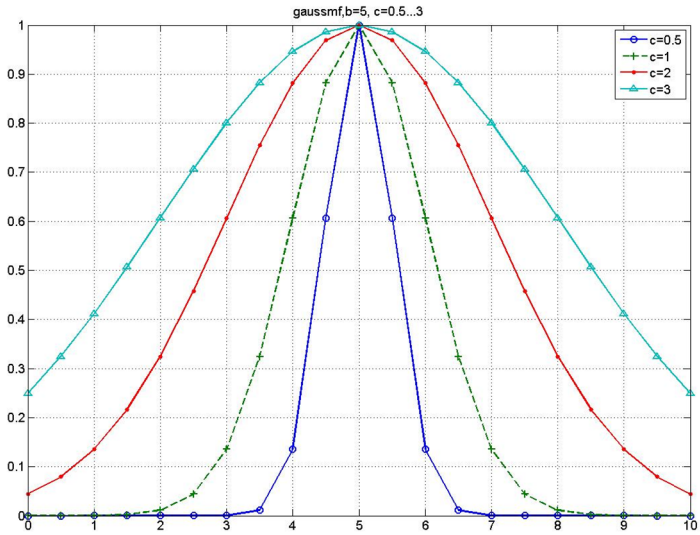


Рисунок 1.3- Проста ФП – gaussmf

Двостороння функція приналежності Гауса, *y* = gauss2mf(*x*, params), задасть функцію приналежності у вигляді наступної комбінації двох кривих Гауса:

якщо $b_1 < b_2$,

то $\mu(x) = \begin{cases} exp((x-b_1)^2/(-2c_1^2)), x < b_1; \\ 1, b_1 \le x \le b_2; \\ exp((x-b_2)^2/(-2c_2^2)), x > b_2; \end{cases}$

якщо $b_1 > b_2$,

то $\mu(x) = \begin{cases} exp((x-b_1)^2/(-2c_1^2)), x < b_2; \\ exp((x-b_1)^2/(-2c_1^2))exp((x-b_2)^2/(-2c_2^2)), b_2 \le x \le b_1; \\ exp((x-b_2)^2/(-2c_2^2)), x > b_1; \end{cases}$

Якщо $b_1 < b_2$, то параметри функції приналежності геометрично інтерпретуються таким чином:

b_1 і b_2 - нижня і верхня межі ядра нечіткої множини;

c_1 і c_2 - коефіцієнти концентрацій лівої і правої гілок графіку функції приналежності.

При $b_1 > b_2$ нечітка множина виходить субнормальною. Функція gauss2mf застосовується для завдання гладких асиметричних функцій приналежності.

Функція gauss2mf має два вхідні аргументи:

x - вектор, для координат якого розраховується ступені приналежності;

params - вектор параметрів функції приналежності.

Порядок завдання параметрів : [c_1 b_1 c_2 b_2].

Програма використання ФП gauss2mf :

```
x = 0: 0.5: 10;
yl = gauss2mf(x, [2 1 1 3]);
y2 = gauss2mf (x,[2 4 1 5]);
y3 = gauss2mf(x, [2 6 1 6]);
plot(x, yl, 'o-', x, y2,'+:',x, y3,'.-');
title('gauss2mf, cl=2, c2=l');
legend('bl=l, b2=3', 'bl=4, b2=5','bl=6, b2=6')
```

Одержаний результат показано на рис. 1.4.

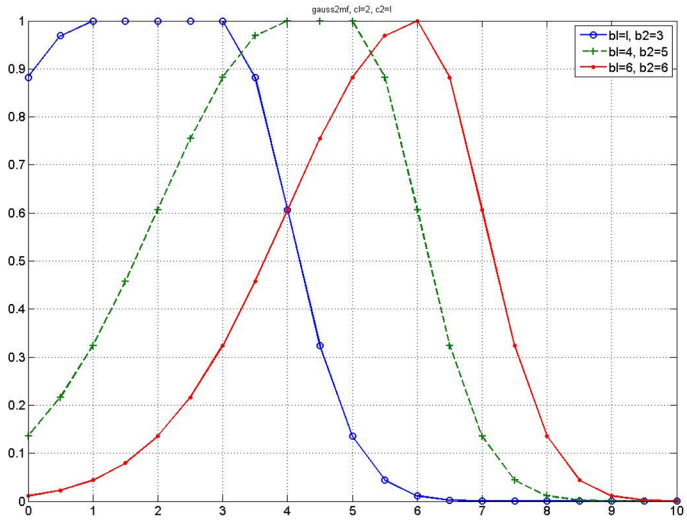


Рисунок 1.4- ФП – gauss2mf

Наступною функцією, яка дозволяє представляти нечіткі суб'єктивні переваги, є ФП "узагальнений дзвін" і позначається gbellmf (generalized bell shape membership function). Її відмінність від розглянутих раніше ФП полягає в додаванні третього параметра, що дозволяє здійснювати плавний перехід між нечіткими множинами.

Узагальнена дзвоноподібна функція приналежності, y = gbellmf(x, params) задає функцію приналежності у вигляді симетричної кривої у формі дзвону.

Ця функція задається формулою

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

параметри якої геометрично інтерпретуються наступним чином:

a - коефіцієнт концентрації функції приналежності;

b - коефіцієнт крутизни функції приналежності (b > 0);

c - координата максимуму функції приналежності.

Функція gbellmf застосовується для завдання гладких симетричних функцій приналежності. Функція gbellmf має два вхідних аргументи:

x - вектор, для координат якого розраховується ступені приналежності;

params - вектор параметрів функції приналежності.

Порядок завдання параметрів : [*a b c*].

Функція gbellmf повертає міри приналежності координат вектору *x*.

Програма використання ФП gbellmf :

```
x = 0: 0.5: 10;
yl = gbellmf(x, [3 1 5]);
y2 = gbellmf (x, [3 2 5]);
y3 = gbellmf(x, [3 3 5]) ;
plot(x, yl,'o-',x, y2,'+:',x, y3,'.-')
title('gbellmf, a=3, b=l, . .., 3, c=5');
legend('b=l', 'b=2', 'b=3')
```

Одержаний результат показано на рис. 1.5.

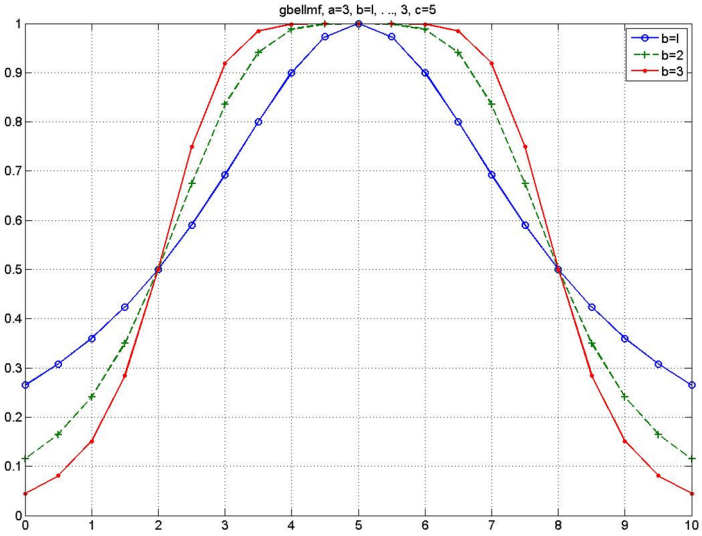


Рисунок 1.5- ФП – gbellmf

Функції приналежності на основі функції розподілу Гауса і ФП "узагальнений дзвін" відрізняються гладкістю та простотою запису і є найбільш використовуваними при описі нечітких множин. Незважаючи на те що гаусові і дзвіноподібні ФП мають властивість гладкості, вони не дозволяють формувати асиметричні ФП.

Для цих цілей передбачений набір сигмоідних функцій, які можуть бути відкриті або ліворуч, або справа залежно від типу функції. Симетричні і закриті функції синтезують з використанням двох додаткових сигмоід. Основна сигмоідна ФП позначається sigmf, а додаткові – dsigmf та psigmf.

Сигмоідна функція приналежності, y = sigmf(x, params) задає сигмоїдну функцію приналежності по формулі:

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

Параметри функції приналежності геометрично інтерпретуються так:

a - коефіцієнт крутизни функції приналежності;

c - координата перегину функції приналежності.

Функція sigmf застосовується для завдання монотонних функцій приналежності. Функція sigmf має два вхідні аргументи:

x - вектор, для координат якого розраховується ступені приналежності;

params - вектор параметрів функції приналежності.

Порядок завдання параметрів : [a c].

Програма використання ФП sigmf:

```
x = 0: 0.5: 10;
yl = sigmf(x, [-2 4]);
y2 = sigmf (x,[-1 4]);
y3 = sigmf (x,[1 4]);
y4 = sigmf(x,[2 4]);
plot(x, yl,'o-',x, y2,'+:',x, y3,'.-', x,y4,'-^')
title('sigmf: a=-2,..., 2, c=4');
legend('a=-2','a=-1','a=l', 'a=2')
```

Одержаний результат показано на рис. 1.6.

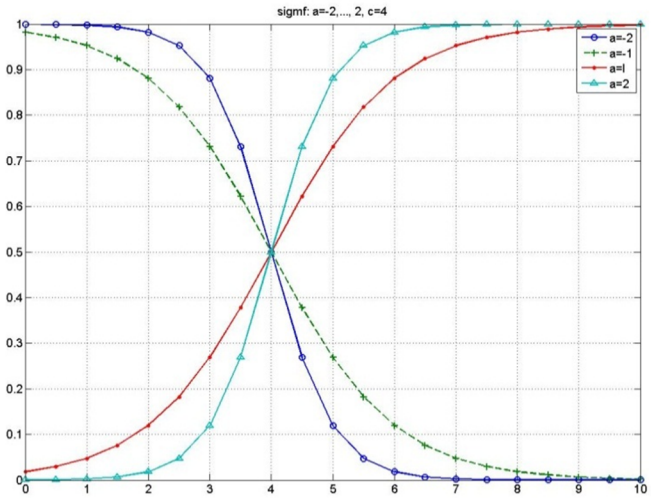


Рисунок 1.6- ФП – sigmf

Залежно від знаку параметра *a* дана ФП буде відкрита або справа або ліворуч.

Функція приналежності у вигляді різниці двох сигмоідних функцій, y = dsigmf(x, params) задає функцію приналежності у вигляді різниці двох сигмоідних функцій по формулі:

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

Застосовується для завдання гладких асиметричних функцій приналежності.

Функція dsigmf має два вхідні аргументи:

x - вектор, для координат якого розраховується ступені приналежності;

params - вектор параметрів функції приналежності.

Порядок завдання параметрів : [a1 c1 a2 c2].

Програма використання ФП dsigmf:

```
x = 0: 0.25: 10;
yl = dsigmf(x,[5 1 8 7]);
y2 = dsigmf(x, [5 4 5 7]);
y3 = dsigmf(x, [5 6 2 7]);
plot (x , yl, 's-', x, y2,'.:',x, y3,'+-') ;
ylim( [0 1.05]);
title('dsigmf: a1=5, c2=7');
legend('cl=l, a2=8', 'c1=4, a2=5' , 'cl=6, a2=2')
```

Одержаний результат показано на рис. 1.7.

Функція приналежності у вигляді добутку двох сигмоідних функцій, y = psigmf(x, params),задає функцію приналежності у вигляді наступного добутку двох сигмоідних функцій:

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

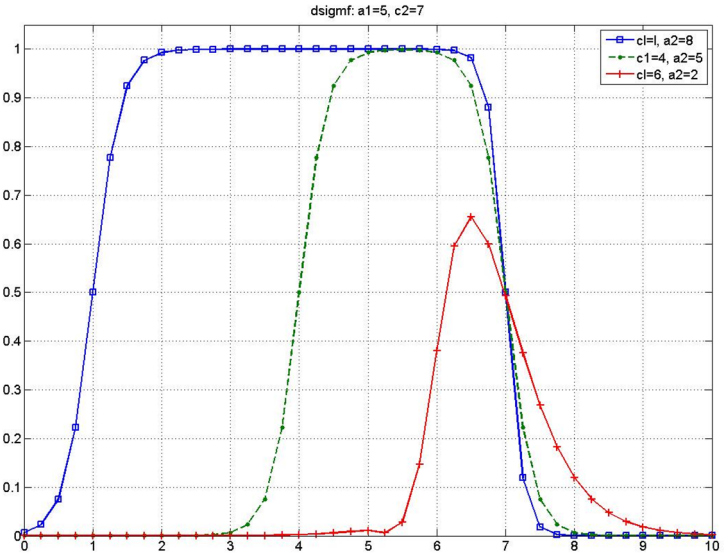


Рисунок 1.7- ФП – dsigmf

Функція psigmf застосовується для завдання гладких асиметричних функцій приналежності. Порядок завдання параметрів : [*a1 c1 a2 c2*].

Програма використання ФП psigmf :

```
x = 0: 0.5: 10;
yl = psigmf(x, [2 1 -1 7]);
y2 = psigmf(x, [2 2 -4 7]);
```

```
y3 = psigmf(x, [2 3 -8 7]);
plot(x, yl,'o-',x, y2, ' +:', x, y3,'.-');
title('psigmf, a=2, d=7');
legend ('b=l, c=-l', 'b=2, c=-4', 'b=3, c=-8')
```

Одержаний результат показано на рис. 1.8.

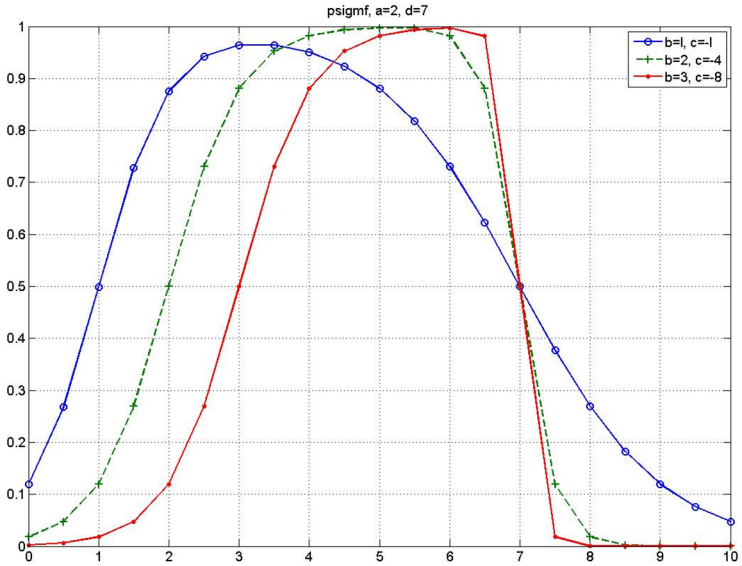


Рисунок 1.8- ФП – psigmf

Інструментарій нечіткої логіки (fuzzy logic toolbox) у складі Matlab надає можливість формування ФП на основі поліноміальних кривих. Відповідні функції називаються Z - функції (zmf), PI - функції (pimf) і S - функції (smf).

Функція приналежності $y = \text{zmf}(x,\text{params})$ задає z - подібну двохпараметричну функцію приналежності.

Функція zmf є асиметричною поліноміальною кривою, відкритою ліворуч Параметри функції приналежності визначають інтервал, усередині якого функція нелінійно убуває від 0 до 1. Функція zmf застосовується для завдання незростаючих функцій приналежності з насиченням, що представляють нечітку безліч типу "дуже низький".

Порядок завдання параметрів : [*a, b*], які задають інтервал убування функції приналежності. Якщо $b<a$, те функція приналежності виходить у вигляді одиничної сходинки, що проходити через точку $(a + b)/ 2$.

Програма використання ФП zmf:

```
x = 0: 0.5: 10;
yl = zmf(x, [2 3]);
y2 = zmf(x, [2 5]);
y3 = zmf(x, [2 9]);
plot(x, yl,'o-',x, y2,'+:',x, y3,'.-')
title ('zmf,    a=2, b=3,...9')
legend('b=3', 'b=5', 'b=9')
```

Побудова графіків z -подібних функцій приналежності з параметрами [2 3], [2 5] і [2 9] показано на рис. 1.9.

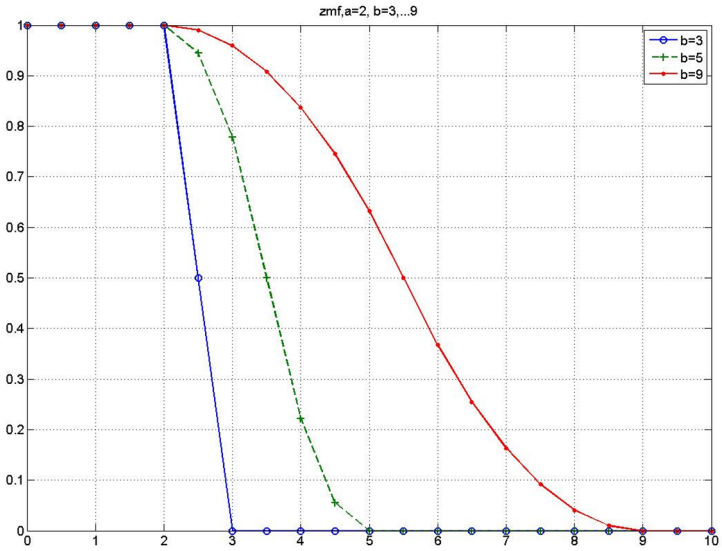


Рисунок 1.9- ФП – zmf

Функція приналежності,

y = smf(x, params)

задає *S* - подібну двохпараметричну функцію приналежності. Параметри функції приналежності задають інтервал, усередині якого функція нелінійно зростає від 0 до 1. Функція smf застосовується, для завдання неубутних функцій приналежності з насиченням, що представляють нечіткі множини типу "дуже високий".

Порядок завдання параметрів : [a, b]. Якщо b<a, то функція приналежності виходить у вигляді одиничного сходинки, що проходить через точку $(a + b) / 2$. Функція smf - дзеркальне відображення функції zmf.

Програма використання ФП smf:

```
x = 0: 0.5: 10;
yl =smf(x, [1 2]);
y2 = smf(x, [1 5]);
y3 = smf(x, [1 9]);
plot (x, yl,'o-',x, y2,'+:',x, y3, '.-')
title('smf, a=l, b=2,9');
legend('b=2','b=5', 'b=9')
```

Побудова графіків s - подібних функцій приналежності з параметрами [1 2], [1 5] і [1 9] показано на рис. 1.10.

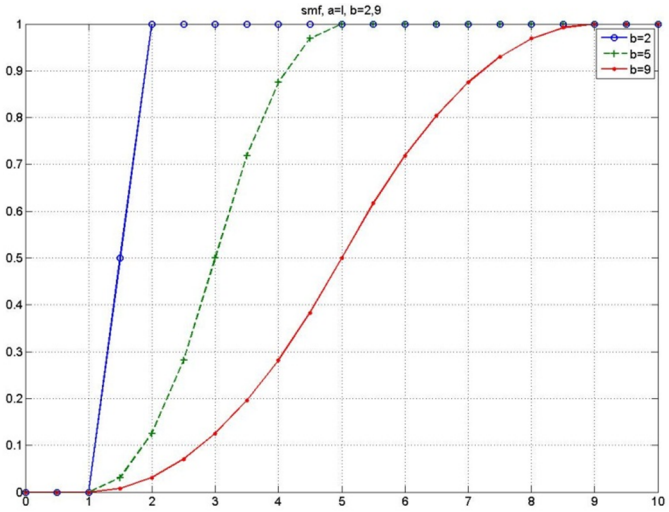


Рисунок 1.10- ФП – smf

Пі- подібна функція приналежності.

y = pimf(x, params)

задає функцію приналежності у вигляді криволінійної трапеції. Ця функція задається як добуток s - і z - подібних функцій приналежності :

pimf(x, [a, Ь, з, d]) = smf (x, [a, b]). *zmf(x, [c, d])

Якщо b > c, то параметри функції приналежності інтерпретуються так:

[a, d] - носій нечіткої множини;

[b, c] - ядро нечіткої множини.

При b > c нечітка множина виходить субнормальною. Відповідно функція pimf дорівнює нулю в правій і лівій межах та набуває значення одиниці в середині деякого відрізку.

Функція pimf застосовується для завдання асиметричних функцій приналежності з плавним переходом від песимістичної до оптимістичної оцінки нечіткого числа.

Порядок завдання параметрів [a b з d]. Параметри a і d задають переходження функції в нульове значення, а параметри b і c - в одиничне.

Програма використання ФП pimf:

```
x = 0: 0.5: 10;
yl = pimf(x, [0 0.5 3 9]);
y2 = pimf(x, [0 4 5.5 9]);
y3 = pimf(x, [0 6 7 9]);
plot(x, yl, 'o-', x, y2,'+:',x, y3,'.-')
title('pimf, a=0, d=9')
legend('b=0.5, c=3','b=4,c=5,5', 'b=6, c=7')
```

Побудова графіку пі- подібної функції приналежності для нечітких множин з ядрами [0,5, 3], [4, 5,5] і [6, 7] показано на рис. 1.11.

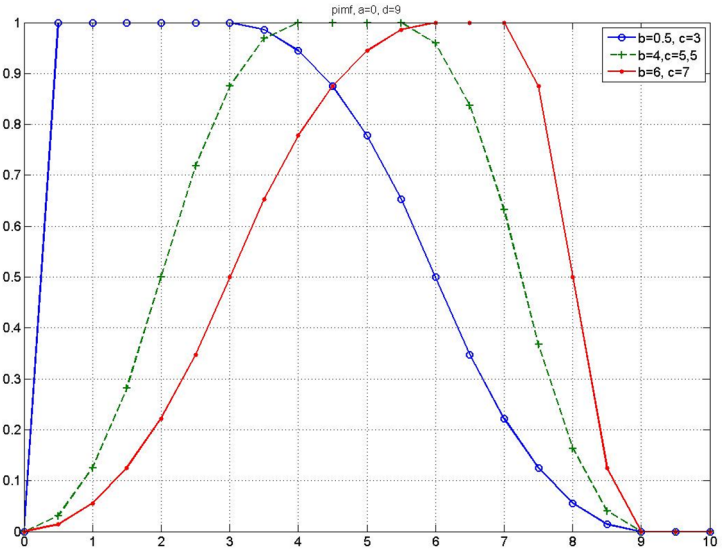


Рисунок 1.11- ФП – pimf

Окрім розглянутих функцій, що дозволяють представляти нечіткі множини, в Matlab є можливість формувати власні ФП або модифікувати вбудовані.

Контрольні питання

1. Що таке нечітка логіка? У яких випадках вона використовується?
2. Що таке нечітка множина і яка його основна відмінність від звичайної (чіткої) множини?
3. Що таке функція приналежності?
4. Наведіть поширені типи функцій приналежності

Практичне заняття 2.

Дослідження операцій над нечіткими множинами

Мета роботи: ознайомитися з найбільш поширеними логічними операціями над нечіткими множинами.

Нехай A і B - довільні нечіткі множини, задані на *U*, тобто елементи класу *P(U)* усіх нечітких множин. Тоді для всякого $x \in U$ операції об'єднання, перетину, абсолютного доповнення, відносного доповнення і симетричної різниці можуть бути задані таким чином:

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

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

$$\overline{A} = \{x \in U, \mu_{\overline{A}}(x) = 1 - \mu_A(x)\};$$

$$A \setminus B = A \cap \overline{B};$$

$$A \oplus B = (A \setminus B) \cup (B \setminus A)$$

Розглянуті операції над нечіткими множинами набули найбільшого поширення при рішенні практичних задач нечіткого моделювання. Це пов'язано з тим, що побудована таким чином алгебра нечітких множин найбільш близька до булевої алгебри і є її узагальненням. Проте, наприклад, для операцій об'єднання, перетину і доповнення нечітких множин можливі і інші коректні способи їх визначення, а доцільність їх застосування обумовлена специфічними особливостями конкретних прикладних завдань.

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 імовірнісної інтерпретацій.

Розглянемо мінімаксну інтерпретацію логічних операторів, в якій кон'юнктивний оператор представляє знаходження мінімуму, - *min*, а диз'юнктивний - максимуму - *max*.

Опис кон'юнктивної функції :

$y = min([y_1; y_2])$

Опис диз'юнктивної функції :

$y = max([y_1; y_2])$

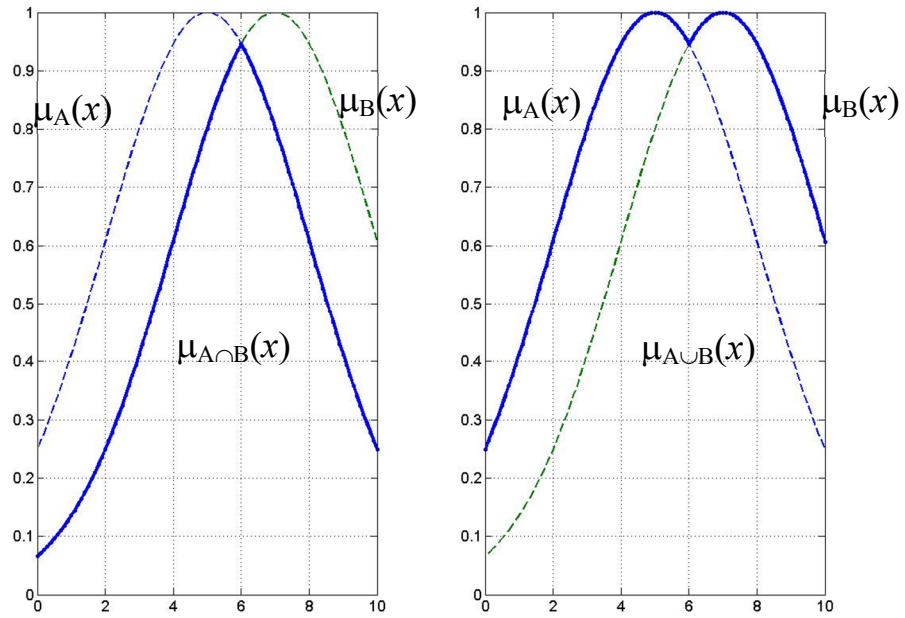
Параметри y_1 і y_2 є початковими ФП. Функція *min* працює із списком ФП. У Matlab список оформляється квадратними дужками, а елементи списку розділяються крапкою з комою.

Програма використання операцій *min* і *max*:

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

```
hold on;
plot (x, y4);
hold off.
```

На отриманих графіках (рис. 2.1) пунктирною лінією зображені початкові ФП, а суцільною лінією - результат дії логічних операторів.



а) *б)*

Рисунок 2.1 - Перетин (*а*) і об'єднання (*б*) нечітких множин (мінімаксна інтерпретація)

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.

У рамках цієї інтерпретації кон'юнктивний оператор є оператор обчислення алгебраїчного добутку - *prod*, а диз'юнктивний оператор - оператор обчислення алгебраїчної суми - *probor*.

Опис функції *prod*:

$$y = prod\ ([y_1\ ;\ y_2\ ]).$$

Опис функції *probor*:

$$y = probor\ ([y_1\ ;\ y_2\ ]).$$

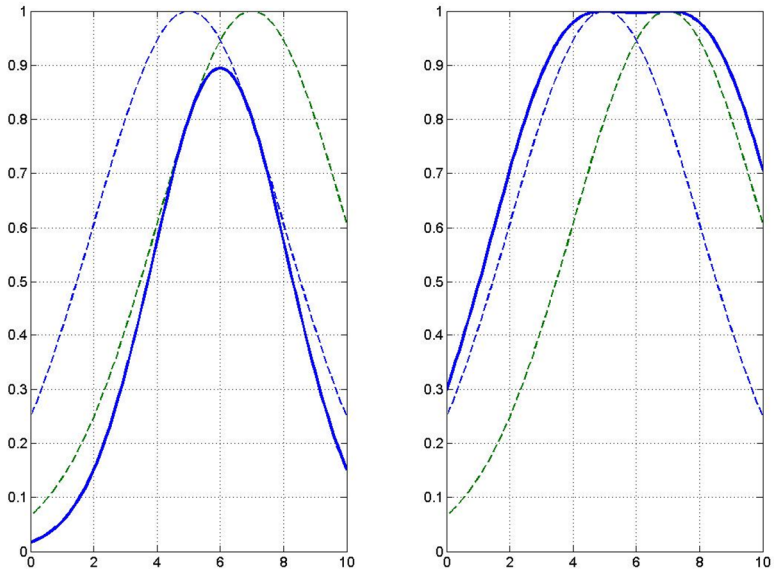
Параметри y_1 і y_2 є початковими ФП.

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

```
x = 0 : 0.1 : 10;
subplot (1, 2, 1) ;
yl = gaussmf(x, [3 5]);
```

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

На отриманих графіках (рис. 2.2) пунктирною лінією зображені початкові ФП, а суцільною лінією - результат дії логічних операторів.



а) *б)*

Рисунок 2.2 - Перетин (*а*) і об'єднання (*б*) нечітких множин (імовірнісна інтерпретація)

Доповнення нечіткої множини є не що інше, як математичне представлення вербального вираження "НЕ А", де А - нечітка множина, що описує деяке розмите судження.

Опис функції доповнення : y=1- y*

де y* — початкова ФП.

Програма використання операції доповнення.

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

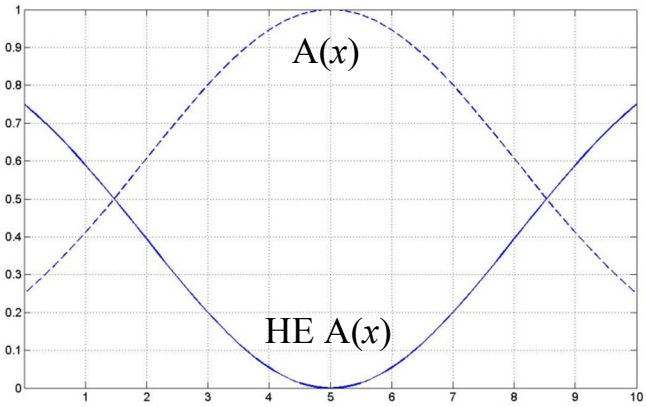


Рисунок 2.3- Доповнення нечіткої множини A(*x*)

Виконання операцій нечіткої арифметики в Matlab виконується за допомогою функції fuzarith.

Опис функції: C = fuzarith(x, A, B, operator)

Використовуючи операції з нечіткими інтервалами, функція повертає нечітку множину C в якості результату виконання бінарної операції, визначуваним рядком operator, над опуклою нечіткою множиною A та B. При цьому нечіткі множини A і B заздалегідь мають бути задані опуклими функціями приналежності на універсумі *x*.

Аргументами цієї функції є:

A, B, x - вектори однакової розмірності;

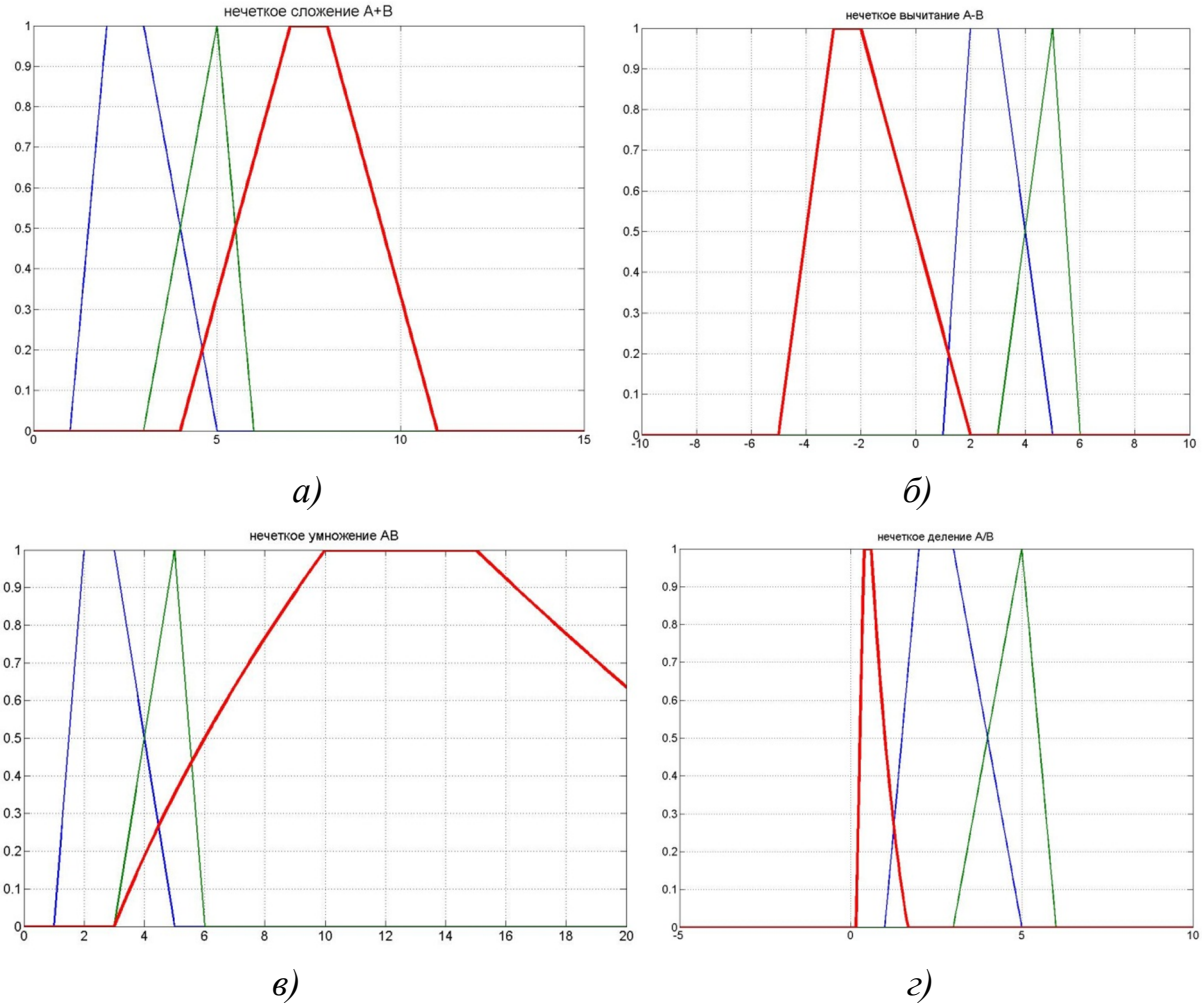
operator - рядок, що приймає одне з наступних значень: 'sum', 'sub', 'prod', 'div'.

Як результат функція повертає C- вектор-стовпець тієї ж розмірності, що і вектор універсуму x.

Програма використання функції fuzarith:

```
point_n = 1001; % определяет количество точек функции принадлежности
min_x = -20; max_x = 20;% универсум нечеткого множества[min_x,max_x]
x=linspace (min_x, max_x, point_n)';
A = trapmf(x, [1 2  3  5]); % A — трапециевидный нечеткий интервал
B = trimf(x, [3 5 6]); % B — треугольное нечеткое число
C1 = fuzarith(x, A, B, 'sum');
plot (x, A, x, B, x, C1);
title('нечеткое сложение A+B');
C2 = fuzarith(x, A, B, 'sub');
plot(x, A, x, B, x, C2);
title('нечеткое вычитание A-B');
C3 = fuzarith(x, A, B, 'prod');
plot(x, A, x, B, x, C3) ;
title('нечеткое умножение AB');
C4 = fuzarith(x, A, B, 'div');
plot(x, A,x, B, x, C4);
title('нечеткое деление A/B');
```

Результат виконання цієї послідовності команд зображений на рис. 2.4.



а) *б)*

в) *г)*

Рисунок 2.4 - Результати виконання операцій з нечітким інтервалом А і нечітким числом В

Контрольні питання

1. Поясніть основні операції над нечіткими множинами.
2. Яким операціям булевої алгебри відповідають процедури фазі-об’єднання і фазі-перетин нечітких множин?
3. За допомогою яких процедур знаходять ФП фазі-об’єднання і фазі-перетин, якщо відомі ФП початкових множин?

Практичне заняття 3.
Моделювання нечіткої системи засобами інструментарію нечіткої логіки

Мета роботи: вивчити метод побудови нечіткої системи засобами інструментарію нечіткої логіки (ІНЛ).

Алгоритми нечіткого виводу розрізняються головним чином видом використовуваних правил, логічних операцій і різновидом методу дефазифікації. Розроблені моделі нечіткого виведення Мамдані, Сугено, Ларсена, Цукамото.

В загальному випадку логічний вивід здійснюється за чотири етапи:

1. Нечіткість (фазифікація, fuzzification). Функції приналежності, визначені для вхідних змінних, застосовується до їх фактичних значень для визначення міри істинності кожної передумови кожного правила).

2. Логічний вивід. Вичислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це призводить до однієї нечіткої підмножини, яка буде призначена змінній виводу для кожного правила. В якості правил логічного виводу зазвичай використовуються операції *min* (мінімуму) або *prod* (множення).

3. Композиція. Нечіткі підмножини, призначені для кожної змінної виводу (в усіх правилах), об'єднуються разом, щоб сформувати одну нечітку підмножину для кожної змінної виводу. При подібному об'єднанні зазвичай використовуються операції *max* (максимум) або *sum* (сума).

4. Дефазифікація - приведення до чіткості (defuzzification). Перетворення нечіткого набору виводів в число.

Процедура дефазифікації аналогічна знаходження характеристик положення (математичного очікування, моди, медіани) випадкових величин в теорії вірогідності. Простим способом виконання процедури дефазифікації являється вибір чіткого числа, відповідного максимуму функції приналежності. Проте придатність цього способу обмежується лише одноекстремальними функціями приналежності.

Для дефазифікації багатоекстремальних функцій приналежності Fuzzy Logic Toolbox використовує п'ять вбудованих методів: *centroid* (координата - абсциса - центру тяжіння фігури під кривою), *bisector* (координата - абсциса – що поділяє площу фігури під кривою навпіл), *mom* (mean of the maximum - середина інтервалу максимальних значень), *lom* (largest of maximum - верхня межа інтервалу максимальних значень) і *som* (smallest of maximum - нижня межа інтервалу мак-

симальних значень). Можливі методи дефазифікації і їх позначення в MatLab приведені на рисунку 3.1.

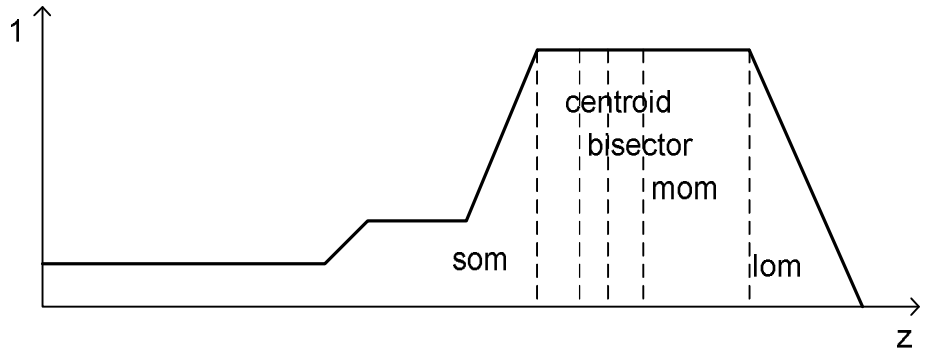


Рисунок 3.1- Можливі методи дефазифікації

Нехай *x, y* - вхідні змінні, що мають чіткі значення x_0, y_0; *z* - вихідна змінна.

Задані функції приналежності A_1, A_2, B_1, B_2, C_1, C_2 і два правила:

П1: ЯКЩО $x \in A_1$ і $y \in B_1$, ТО $z \in C_1$,

П2: ЯКЩО $x \in A_2$ і $y \in B_2$, ТО $z \in C_2$

Алгоритм Мамдані (Mamdani)

У системах типу Мамдані база знань будується з нечітких висловлювань виду "$\beta \in \alpha$" за допомогою зв'язок "І", "ЯКЩО - ТО".

Етапи нечіткого виводу реалізуються наступним чином:

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.

Потім знаходяться усічені функції приналежності :

$$C'_1(z) = (\alpha_1 \wedge C_1(z)),$$

$$C'_2(z) = (\alpha_2 \wedge C_2(z))$$

3) Композиція: з використанням операції максимум (позначається як "$\vee$") виконується об'єднання знайдених усічених функцій, що призводить до отримання підсумкової нечіткої підмножини для змінної виходу з функцією приналежності :

$$\mu_\Sigma(z) = C(z) = C_1'(z) \vee C_2'(z) = (\alpha_1 \wedge C_1(z)) \vee (\alpha_2 \wedge C_2(z))$$

4) Приведення до чіткості для отримання z_0 виконується методом центру тяжіння (центроїдним методом).

Алгоритм ілюструється на рисунку 3.2.

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 :

$$z_0 = \frac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}$$

Алгоритм ілюструється на рисунку 3.3.

Операції з нечіткою логікою у пакеті MatLab дозволяє виконувати модуль *Fuzzy Logic Toolbox*. Він дозволяє створювати системи нечіткого логічного виведення і нечіткої класифікації в рамках середовища MatLab, з можливістю їхнього інтегрування в Simulink.

У складі Matlab є п'ять основних засобів графічного інтерфейсу користувача (ГІК). які забезпечують доступ до ІНЛ: редактори системи нечіткого виводу (СНВ), функції приналежності, правил виводу, а також засоби перегляду правил і поверхні виводу. Ці засоби пов'язані між собою динамічно і зроблені зміни в одному з них тягнуть зміни в інших.

Редактор СНВ (*FIS-редактор*) призначений для створення, збереження, завантаження і виведення у друк систем нечіткого логічного виводу, а також для редагування наступних властивостей: тип системи; найменування системи; кількість вхідних і вихідних змінних; найменування вхідних і вихідних змінних; параметри нечіткого логічного виводу.

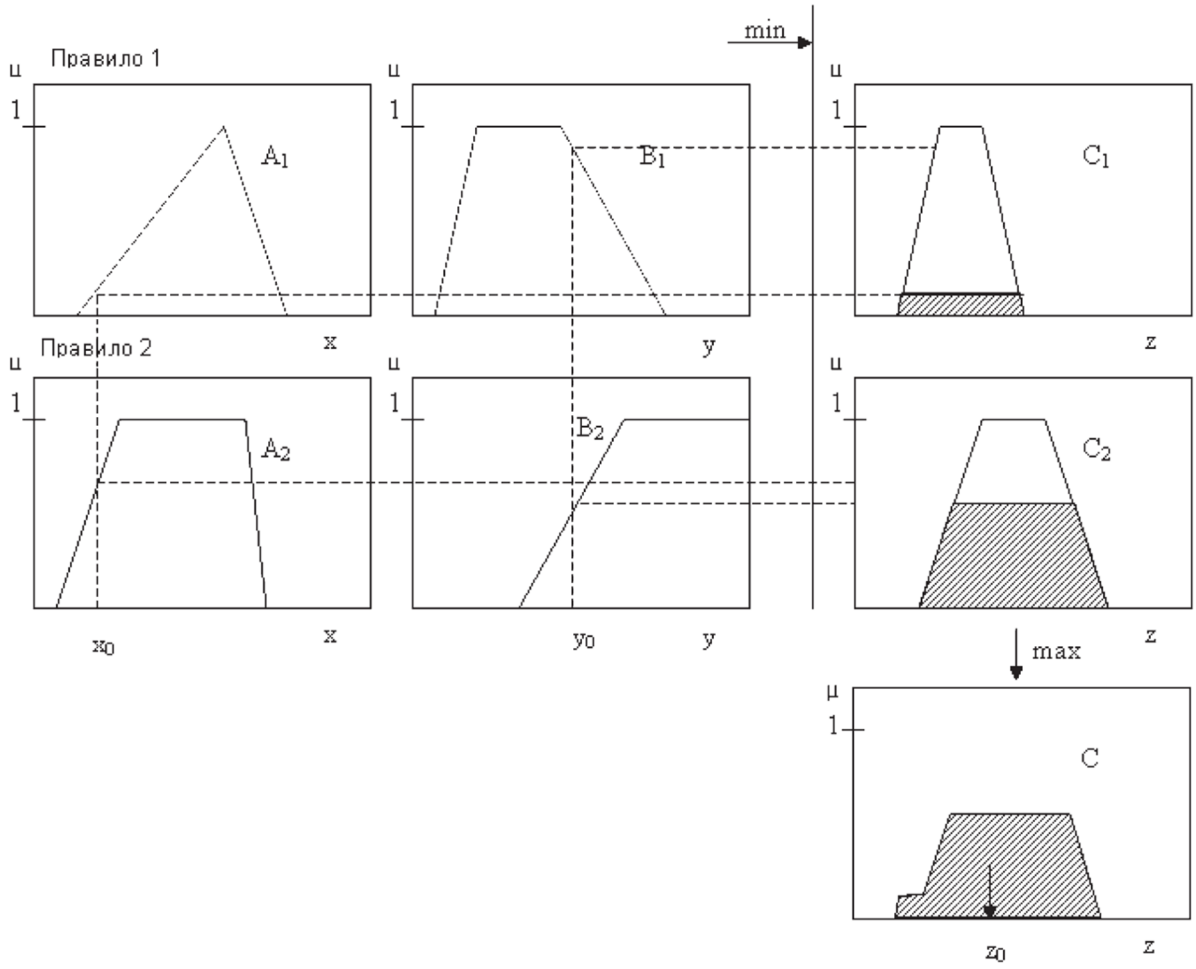


Рисунок 3.2- Алгоритм Mamdani

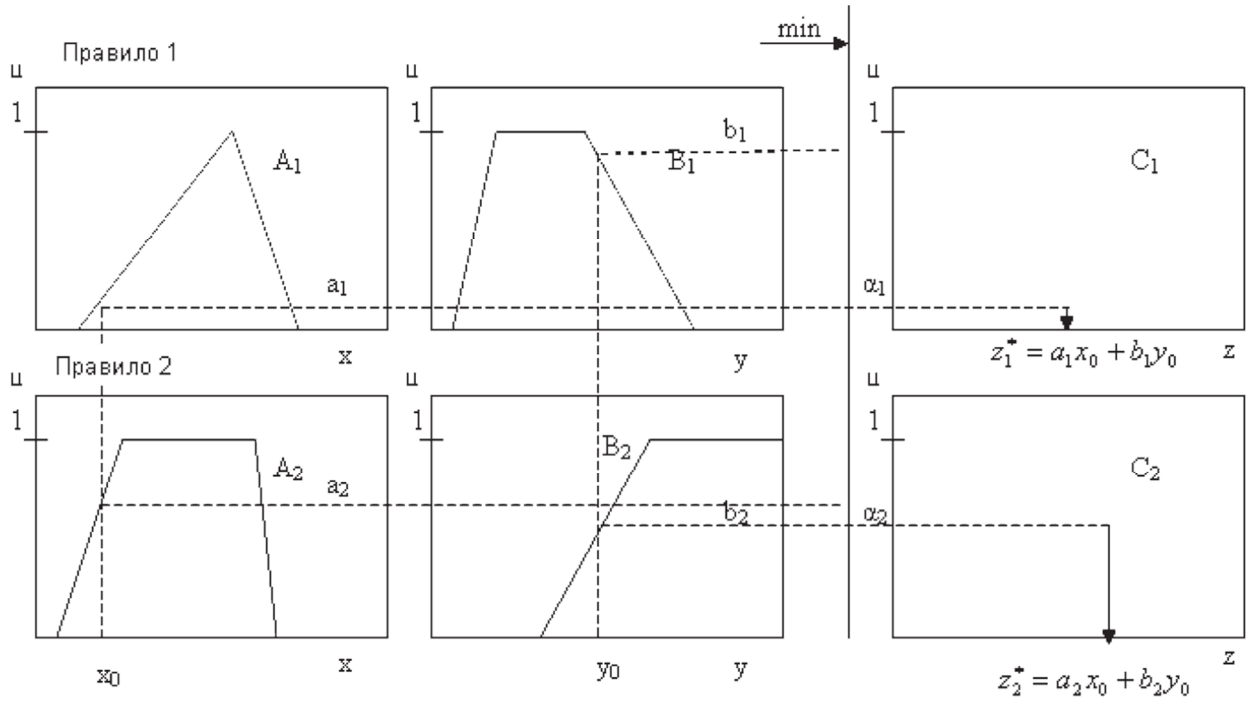


Рисунок 3.3 - Алгоритм Sugeno

Для побудови створюваної системи в командному рядку основного вікна Matlab необхідно набрати команду / *fuzzy*. Вікно *FIS*-редактора нової СНВ містить вхідну, позначену *input1* та вихідну - *output1* змінні. За умовчанням ІНЛ пропонує створювати СНВ типу Мамдані (рис 3.4).

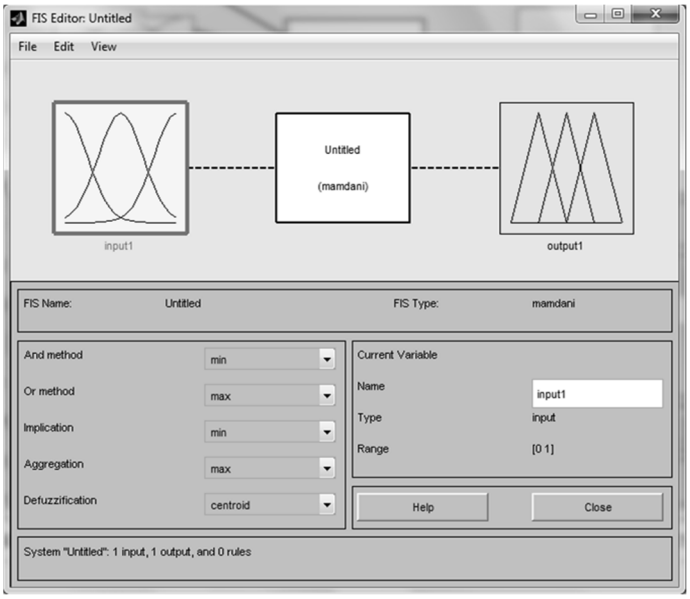


Рисунок 3.4 - Графічне вікно FIS-редактора

FIS-редактор містить 8 меню. Це три загальносистемних меню - *File*, *Edit*, *View*, і п'ять меню для вибору параметрів нечіткого логічного виводу – *And Method, Or Method, Implication, Aggregation і Defuzzification.*

Меню File - це загальне меню для всіх GUI-модулів використовуваних із системами нечіткого логічного виводу.

За допомогою команди *New FIS...* користувач має можливість створити нову систему нечіткого логічного виведення. При виборі цієї команди з'являться дві альтернативи: Mamdani і Sugeno, що визначають тип створюваної системи.

Команда *Add Variable...* дозволяє додати в систему нечіткого логічного виведення ще одну змінну. При виборі цієї команди з'являться дві альтернативи *Input* і *Output,* що дозволяють додати вхідну і вихідну змінну, відповідно.

Команда *Remove Selected Variable* видаляє поточну змінну із системи. Ознакою поточної змінної є червона окантовка її прямокутника. Призначення поточної змінної відбувається за допомогою однократного щиглика лівої кнопки миші по її прямокутнику. Видалити поточну змінну можна також за допомогою натискання Ctrl+X.

Команда *Membership Function...* відкриває редактор функцій приналежності. Ця команда може бути також виконана натисканням Ctrl+2.

Команда *Rules*…відкриває редактор бази знань. Ця команда може бути також виконана натисканням Ctrl+3.

Для того, щоб додати нову змінну, необхідно вибрати в меню *Edit* відповідний пункт (для вхідної змінної – *Add input*, для вихідної - *Add output*). Зміна найменування змінної відбувається по кроках.

- Крок 1. Відзначається змінна, яку необхідно перейменувати.
- Крок 2. У полі редагування змінюється найменування змінної за умовчанням на ім'я, яке запропоноване користувачем.

Збереження проектованої системи в робочий простір середовища Matlab здійснюється з допомогою пункту меню *file - Save to workspace as*. В цьому випадку дані зберігаються до закінчення сеансу роботи з Matlab. Для збереження даних на диску після закінчення сеансу роботи застосовується відповідний пункт того ж меню - *Save to disk as*.

Редактор ФП (Membership Function Editor)**.** Наступним кроком в побудові нечіткої моделі засобами ІНЛ є асоціювання набору ФП з кожною вхідною і вихідною змінною. Ця операція виконується у редакторі ФП трьома способами, активізувати які можна:

1. вибором в меню *View* пункту *Edit Membership Functions*..
2. подвійним клацанням миші на зображенні відповідної змінної (вхідної або вихідної);
3. набором в командному рядку оператора *mfedit*.

За допомогою редактора ФП (рис. 3.5) можна відображувати і редагувати будь-які ФП, що асоціюються (пов'язані) з усіма вхідними і вихідними змінними СНВ.

Зв'язування ФП з ім'ям змінної відбувається наступним чином:

1. вибирається змінна по імені з набору графічних об'єктів вікна редактора ФП :
2. вказується діапазон зміни значень для базової і видимий діапазон для поточних змінних: у меню *Edit* вибирається пункт *AddMFs*.. У вікні, що з'явилося, вибирають вид ФП і їх кількість.

Редагують ФП поточної змінної двома способами: використовуючи графічне вікно ФП або змінюючи характеристики ФП (найменування, тип і числові параметри).

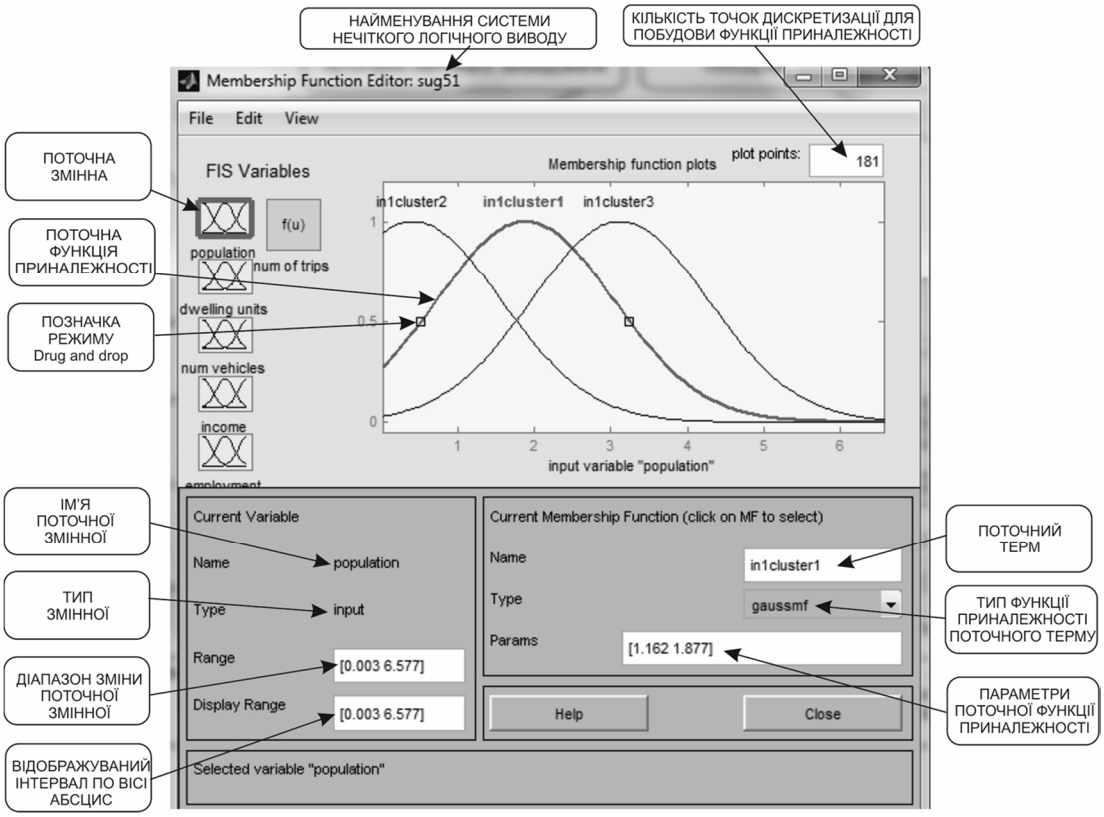


Рисунок 3.5 - Редактор функцій приналежності

Редактор функцій приналежності містить чотири меню - File, Edit, View, Type і чотири вікна введення інформації – Range, Display Range, Name і Params. Ці чотири вікна призначені для завдання діапазону зміни поточної змінної, діапазону виведення функцій приналежності, найменування поточного лінгвістичного терму і параметрів його функції приналежності, відповідно.

Параметри функції приналежності можна підбирати й у графічному режимі, шляхом зміни форми функції приналежності за допомогою технології “Drug and drop”. Для цього необхідно позиціонувати курсор миші на знаку режиму “Drug and drop”, натиснути на ліву кнопку миші і не відпускаючи її змінювати форму функції приналежності. Параметри функції приналежності будуть перераховуватися автоматично.

Таким чином, при побудові СНВ необхідно за допомогою редактора ФП визначити відповідні функції для кожної з вхідних і вихідних змінних.

Редактор правил виводу (Rule Editor). Після того, як вказано кількість вхідних і вихідних змінних, визначені їх найменування та побудовані відповідні ФП, в СНВ необхідно включити правила виводу. Для цього в меню *View* вибирається пункт *Edit Rules.*. або в командному рядку *Matlab* набирається команда *ruleedit*.

Загальний вид редактора правил виводу або редактора бази знань із указівкою функціонального призначення основних полів графічного вікна приведений на рис. 3.6.

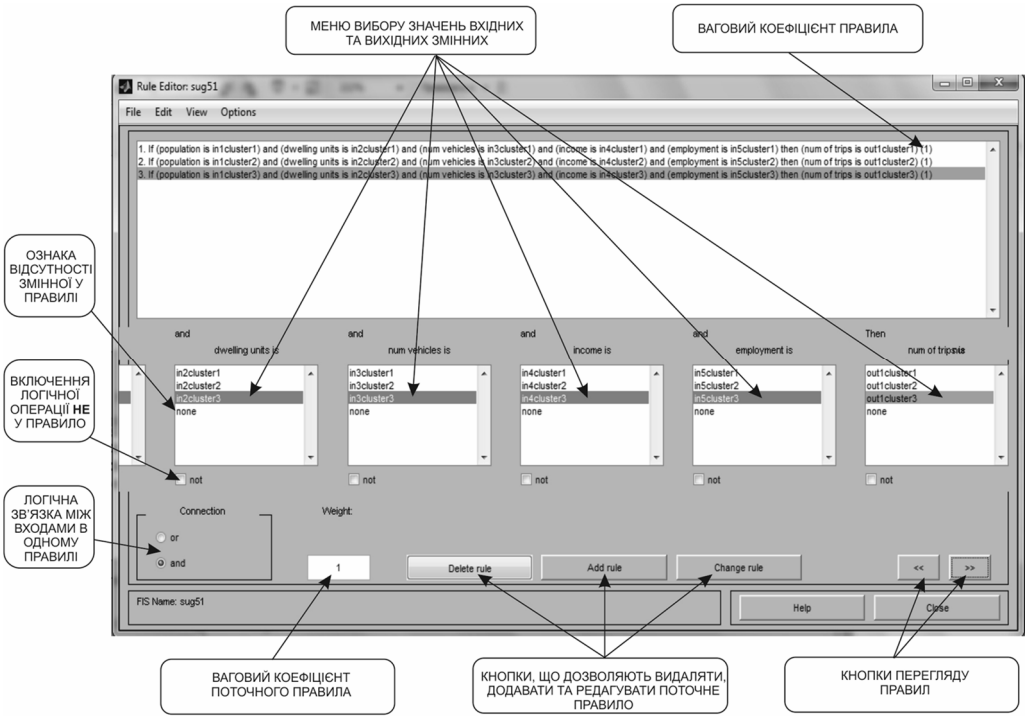


Рисунок 3.6 – Редактор правил виводу (редактор бази знань)

Засіб перегляду правил виводу використовується в цілях діагностики і може показувати, наприклад, активність правил або форму впливу окремих ФП на результат нечіткого виводу.

Ґрунтуючись на описах вхідних і вихідних змінних, визначених в редакторі ФП, редактор правил виводу формує структуру правил автоматично. Від користувача вимагається лише зв'язати значення вхідних і вихідних змінних, вибираючи із списку ФП заданих раніше та визначити логічні зв'язки між ними. Також допускається використання логічного заперечення (НЕ) і зміна ваг правил в діапазоні від 0 до 1.

Правила виводу можуть відображуватися у вікні в різних форматах, які визначаються шляхом вибору відповідного пункту підміню *Format* меню *Options*. За умовчанням використовується розширений формат відображення правив виводу (*verbose form*) :

if (input_lis[not]mf_1j$_1$,) <and,or>...
(input_i is [not]mf_ij$_1$)...<and, or>
(input_n is [not]mf_nj$_n$)then
(output_1 is[notJmf_n+ j$_{n+1}$)<and, or>...

(output_k is[not]mf_ k+ nj$_{k+n}$) <and,or>…

(output_m is[not)mf_m +nj$_{m+n}$) (w),

де

i - номер вхідної змінної;

j - номер ФП *i* -ої змінної;

k - номер вихідний змінної;

n - кількість вхідних змінних;

m - кількість вихідних змінних;

w - вага правила.

Круглі дужки включають в собі обов'язкові параметри, квадратні, - необов'язкові. а кутові - альтернативні параметри (один на вибір).

Окрім формату по замовченню існують ще два види форматів відображення правил : символьний (*symbolic form*) і індексний (*indexed form*). Символьний формат має наступний вигляд:

(input_1<~=,==>mf_1j$_1$,)<&,|>…

(input_i<~=,==>mf_1j$_1$,)…<&,|>

(input_<~=,==>mf_nj$_n$)=>

(output_1 <~=,==>mf_n+ 1j$_{n+1}$)… <&,|>

(output_k <~=,==>mf_ k+ nj$_{k+n}$)<&,|> …

(output_m <~=,==>mf_m +nj$_{m\neq n}$) (w)

Відмінно символьного формату від розширеного полягає в тому, що замість словесної інтерпретації зв'язка використовується символьна (символи "&" і "|" - відповідно визначає логічне І та логічне АБО, символ «~» - логічного заперечення, а символ "=>" є роздільником умовної і завершальної частин правила (антецендена та консеквента).

Загальний опис правила виводу в індексному форматі може бути представлений в наступному виді:

[-]1j_1…[-]ij_1…[-]nj_n[-]n+1j_{n+1}…[-]k+nj_{k+1}[-]m+njm+$n(w)$:<1,2>.

Тут порядок дотримання чисел відповідає черзі змінних, що вводяться, причому символ «,» розділяє правило на умовну та завершальну частини. До двокрапки записується порядковий номер відповідної ФП, потім двокрапка - вид логічної зв'язки («1» - логічне І. «2» - логічне АБО). Логічне заперечення задається символом «-».

Після визначення правил виводу в однойменному редакторі можна стверджувати, що СНВ повністю створена.

Засіб перегляду правил виводу. Візуалізація нечіткого логічного виведення здійснюється за допомогою GUI-модуля *Rule Viewer*. Цей модуль дозволяє проілюструвати хід логічного виведення за кожним правилом, одержання результуючої нечіткої множини і виконання процедури дефазифікації. Rule Viewer може бути викликаний з будь-якого GUI-модуля, використовуваного із системами нечіткого логічного виводу, командою View rules … меню View чи натисканням клавіш Ctrl+4. Вид Rule Viewer для системи логічного виведення із указівкою функціонального призначення основних полів графічного вікна приведений на рис. 3.7.

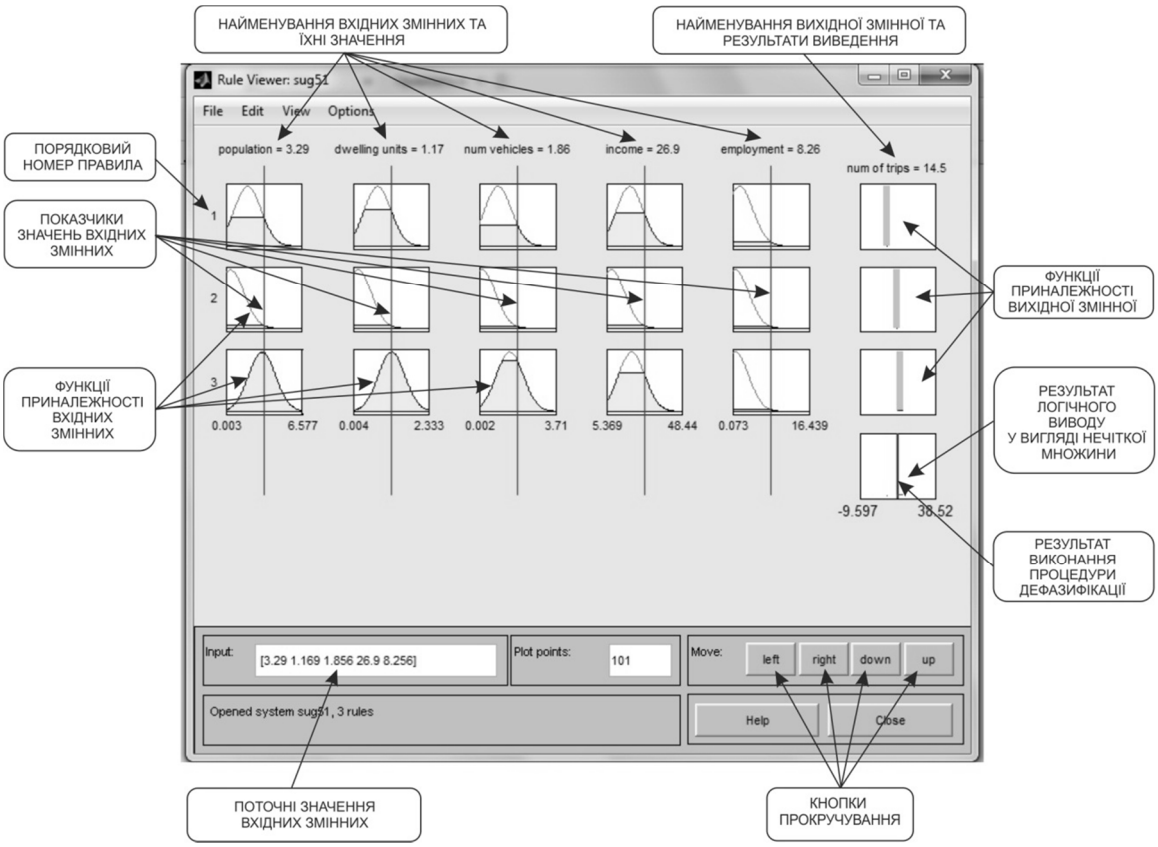


Рисунок 3.7 – Візуалізація логічного виведення для системи sug51 за допомогою Rule Viewer

Головне вікно засобу перегляду складається з декількох графічних вікон, що розташовуються по рядках і стовпцях. Кількість рядків відповідає числу правил нечіткого виводу, а кількість стовпців - числу вхідних і вихідних змінних, заданих в СНВ, що розробляється. Додаткове графічне вікно служить для відображення результату нечіткого виводу і операції дефазифікації. У кожному вікні відображується відповідна ФП, рівень її зрізу (для вхідних змінних) і вклад окремої ФП в загальний результат (для вихідних змінних).

У нижній частині головного вікна можна відобразити номери правил виводу в різних форматах виводу, відмічаючи їх мишею. Для зміни формату в меню Options вибирається пункт Rule display format.

Зміна значень вхідних змінних допустима двома способами:

1) шляхом введення в поле Input запису вхідного вектору, розмірність якого дорівнює кількості вхідних змінних;

2) клацанням миші в будь-якому графічному вікні, яке відноситься до вхідної змінної.

Вхідний вектор в кожному з цих варіантів визначення початкових даних задаватиме набір червоних вертикальних прямих.

Засіб перегляду поверхні виводу. Засіб перегляду поверхні виводу дозволяє будувати тривимірну поверхню як залежність одного з вихідних змінних від двох вхідних. Вибір вхідних і вихідних змінних здійснюється за допомогою спадаючих меню головного вікна даного програмного засобу. Кількість ліній, що виводяться, по осях X а Y визначається в полях введення X grids, Y grids. Приклад поверхні нечіткого виводу, відповідна правилам виведення показана на рис. 3.8.

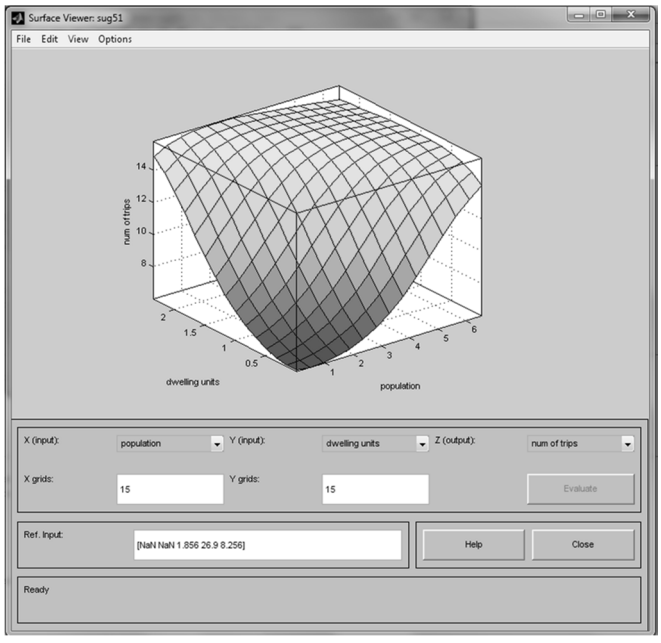


Рисунок 3.8-Візуалізація поверхні нечіткого виводу

Побудова нечітких систем типу Сугено. Розглянемо побудову СНВ двома редакторами - СНВ і ФП. Для побудови СНВ типу Сугено необхідно в меню File вибрати пункт New Sugeno FIS. Кількість вхідних і вихідних змінних визначається так само, як і при побудові СНВ типу Мамдані.

Для СНВ типу Сугено зміни торкаються тільки схеми визначення ФП для вихідних змінних. ІНЛ в середовищі Matlab дозволяє розробляти два види нечітких моделей. Перша модель - це нечітка модель Сугено нульового порядку. Нечітке правило виводу має наступний вигляд:

If x is A and y is B then z = k

де A і B ~ нечітку безліч антецедента;

k ~ чітко задана константа консеквента.

Для побудови такої моделі при додаванні ФП необхідно вибрати тип - константа (constant) і задати як параметр ФП чисельне значення відповідної константи. Друга модель - нечітка модель Сугено першого

порядку. Для неї нечітке правило виводу записується таким чином:

ifx as A and y is B then z = p - x+ q - y + r

де p, q і r - константи.

В даному випадку тип ФП - лінійна залежність (linear). Для визначення параметрів ФП необхідно ввести вектор, елементи якого відповідають чисельним значенням констант консеквента.

Робота з редактором правил виводу, а також із засобами перегляду правил і поверхні виводу виконується аналогічно випадку побудови СНВ по Мамдані.

Контрольні питання

1. Яка структура типової системи нечіткого виводу?

2. У чому відмінність методу нечіткого виводу по Сугено від методу нечіткого виводу по Мамдані?

3. Яким чином формуються антецеденты і консеквенты нечітких правил в Matlab?

4. Які можливості по візуалізації результатів моделювання надає система MATLAB?

5.Чому при використанні методу Сугено немає процедури дефазифікації ?

Практичне заняття 4.

Синтез нейро - нечіткої мережі в середовищі MATLAB

Мета*:* вивчити принципи побудови нейро - нечітких мереж в пакеті прикладних програм ANFIS в середовищі системи MATLAB.

Гібридна мережа є багатошаровою нейронною мережею спеціальної структури без зворотних зв'язків, в якій використовуються звичайні (не нечіткі) сигнали, ваги і функції активації, а виконання операції підсумовування засноване на використанні фіксованої Т-норми, Т- конорми або деякої іншої безперервної операції. При цьому значення входів, виходів і ваг гібридної нейронної мережі є дійсні числа з відрізку [0, 1].

Основна ідея, покладена в основу моделі гібридних мереж, полягає у тому, щоб використовувати існуючу вибірку даних для визначення параметрів функцій приналежності, яка краще всього відповідає деякій системі нечіткого виводу. При цьому для знаходження параметрів функцій приналежності використовуються відомі процедури навчання нейронних мереж.

У пакеті Fuzzy Logic Toolbox системи MATLAB гібридні мережі реалізовані у формі так званої адаптивної системи нейро-нечіткого виводу ANFIS. З одного боку, гібридна мережа ANFIS є нейронною мережею з єдиним виходом і декількома входами, які є нечіткими лінгвістичними змінними. При цьому терми вхідних лінгвістичних змінних описуються стандартними для системи MATLAB функціями приналежності, а терми вихідної змінної представляються лінійною або постійною функцією приналежності.

З іншого боку, гібридна мережа ANFIS є системою нечіткого виводу FIS типу Сугено нульового або першого порядку, в якій кожне з правил нечіткої продукції має постійну вагу, яка дорівнює 1. ANFIS є абревіатурою Adaptive Neuro - Fuzzy Inference System - (адаптивна нейро-нечітка система). ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткі мережі. Нейро-нечітку мережу можна розглядати як один з різновидів систем нечіткого логічного виводу типу Сугено. При цьому функції приналежності синтезованих систем налагоджені (навчені) так, щоб мінімізувати відхилення між результатами нечіткого моделювання і експериментальними даними. Завантаження ANFIS-редактора здійснюється по команді anfisedit. У результаті виконання цієї команди з'явиться графічне вікно (рисунок 4.1.). На цьому ж рисунку вказані функціональні області ANFIS-редактора, опис яких наведено нижче.

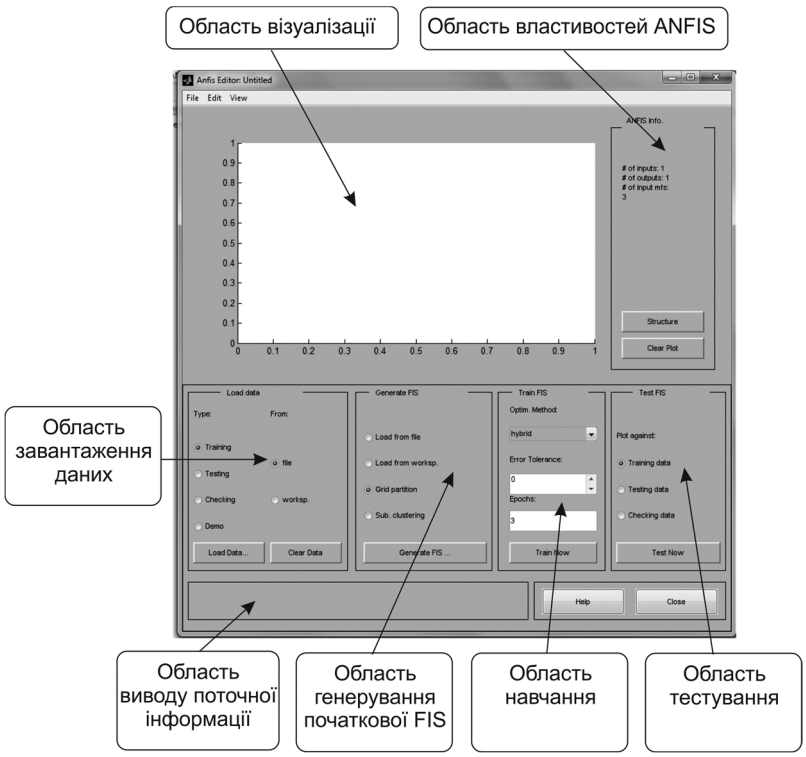


Рисунок 4.1- Основне вікно ANFIS-редактора

ANFIS-редактор містить 3 верхні меню - *File, Edit і View*, область візуалізації, область властивостей ANFIS, область завантаження даних, область генерування початкової системи нечіткого логічного висновку, область навчання, область тестування, область виведення поточної інформації, а також кнопки Help і Close, які дозволяють викликати вікно довідки і закрити ANFIS-редактор, відповідно.

Меню File і View однакові для усіх GUI-модулів, використовуваних з системами нечіткого логічного висновку.

Загальний вигляд меню приведений на рис. 4.2.

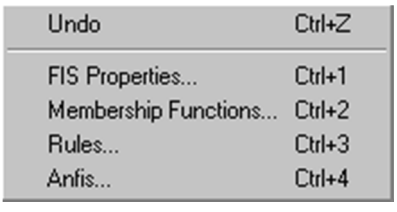


Рисунок 4.2 - Меню Edit

Команда Undo відміняє раніше здійснену дію. Виконується також по натисненню Ctrl+Z.

Команда FIS Properties відкриває FIS-редактор. Ця команда може бути також виконана натисненням Ctrl+1.

Команда Membership Functions. відкриває редактор функцій приналежності. Ця команда може бути також виконана натисненням Ctrl+2.

Команда Rules відкриває редактор бази знань. Ця команда може бути також виконана натисненням Ctrl+3.

Команда Anfis відкриває ANFIS -редактор. Ця команда може бути також виконана натисненням Ctrl+3. Помітимо, що ця команда, запущена з ANFIS-редактора не призводить до виконання яких-небудь дій, так цей редактор вже відкритий. Проте, в меню Edit інших GUI-модулів, використовуваних з системами нечіткого логічного висновку, додається команда Anfis, що дозволяє відкрити ANFIS-редактор з цих модулів.

Загальна послідовність процесу розробки моделі гібридної мережі може бути представлена в наступному виді.

1. Підготовка файлу з навчальними даними. Доцільно скористатися редактором електронних таблиць MS Excel. Навчальну вибірку необхідно зберегти в зовнішньому файлі з розширенням *.dat.
2. Відкрити редактор ANFIS. Завантажити файл з навчальними даними.

Кнопка завантаження даних **Load Data**, по натисненню якої з'являється діалогове вікно вибору файлу, якщо завантаження даних відбувається з диска, або вікно введення ідентифікатора вибірки, якщо завантаження даних походить з робочої області;

Зовнішній вигляд редактора ANFIS із завантаженими навчальними даними зображений на рис. 4.3.

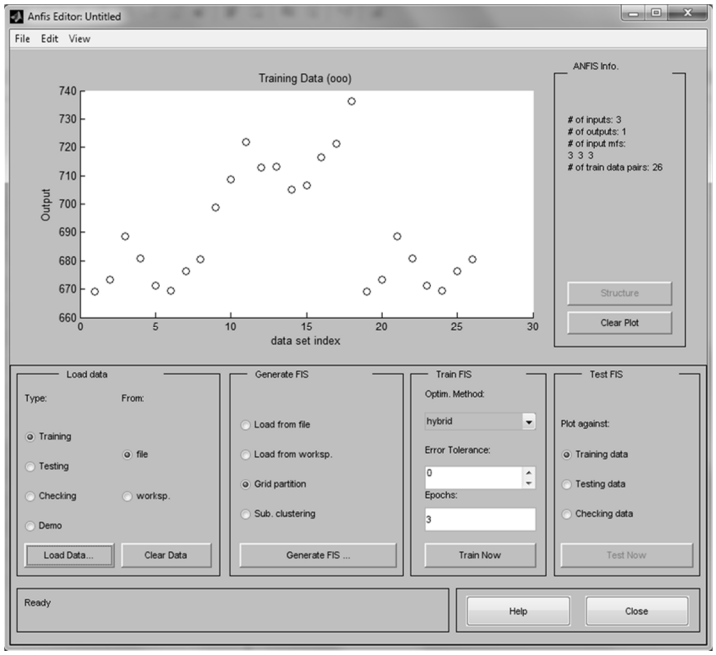


Рисунок 4.3 - Графічний інтерфейс редактора ANFIS після загрузки навчальних даних

В області завантаження даних (Load data) розташовані: меню вибору типу даних (Type), що містить альтернативи (Traning - навчальна вибірка; Testing - тес-

това вибірка; Checking - контрольна вибірка; Demo - демонстраційний приклад); меню вибору джерела даних (From), що містить альтернативи (disk – диск; worksp. - робоча область MatLab); кнопка завантаження даних Load Data…, по натисканню якої з'являється діалогове вікно вибору файлу, якщо завантаження даних відбувається з диска, або вікно введення ідентифікатора вибірки, якщо завантаження даних походить з робочої області; кнопка очищення даних Clear Data.

В області генерування (Generate FIS) розташовані меню вибору способу створення вихідної системи нечіткого логічного виводу. Меню містить наступні альтернативи: *Load from disk* – завантаження системи з диска; *Load from worksp*. – завантаження системи з робочої області MatLab; *Grid partition* - генерування системи по методу ґрат (без кластеризації); *Sub. clustering* – генерування системи за методом субкластеризації.

В області також розташована кнопка Generate, по натисканню якої генерується вихідна система нечіткого логічного виведення.

При виборі *Load from disk* з'являється стандартне діалогове вікно відкриття файлу.

При виборі *Load from worksp.* з'являється стандартне діалогове вікно введення ідентифікатора системи нечіткого логічного виведення.

При виборі Grid partition з'являється вікно введення параметрів методу ґрат, у якому потрібно вказати кількість термів для кожен вхідний змінної і тип функцій приналежності для вхідних і вихідної змінних.

При виборі Sub. clustering з'являється вікно введення наступних параметрів методу субкластеризації: *Range of influence* – рівні впливу вхідних змінних; *Squash factor* – коефіцієнт пригнічення; *Accept ratio* – коефіцієнт, що встановлює у скільки разів потенціал даної точки повинний бути вище потенціалу центра першого кластера для того, щоб центром одного з кластерів була призначена розглянута точка; *Reject ratio* - коефіцієнт, що встановлює у скількох разів потенціал даної точки повинний бути нижче потенціалу центра першого кластера, щоб розглянута точка була виключена з можливих центрів кластерів.

В області навчання (Train FIS) розташовані меню вибору методу оптимізації (Optim. method), поле завдання необхідної точності навчання (Error tolerance), поле завдання кількості ітерацій навчання (Epochs) і кнопка Train Now, натискання якого запускає режим навчання. Проміжні результати навчання виводяться в область візуалізації й у робочу область MatLab. У ANFIS-редакторі реалізовані два методи навчання: backprora – метод зворотного поширення помилки, заснований

на ідеях методу найшвидшого спуска; hybrid – гібридний метод, що поєднує метод зворотного поширення помилки з методом найменших квадратів.

В області тестування (Test FIS) розташовані меню вибору вибірки і кнопка Test Now, по натисканню по який відбувається тестування нечіткої системи з виведення результатів в область візуалізації.

3. Після підготовки і завантаження навчальних даних можна згенерувати структуру системи нечіткого виводу FIS типу Сугено, яка являється моделлю гібридної мережі в системі Matlab. Для цієї мети слідує скористатися кнопкою **Generate FIS** в нижній частині робочого вікна редактора.

Після натиснення кнопки **Generate FIS** викликається діалогове вікно з вказівкою числа і типу функцій приналежності для окремих термів вхідних змінних і вихідної змінної (рис. 4.4).

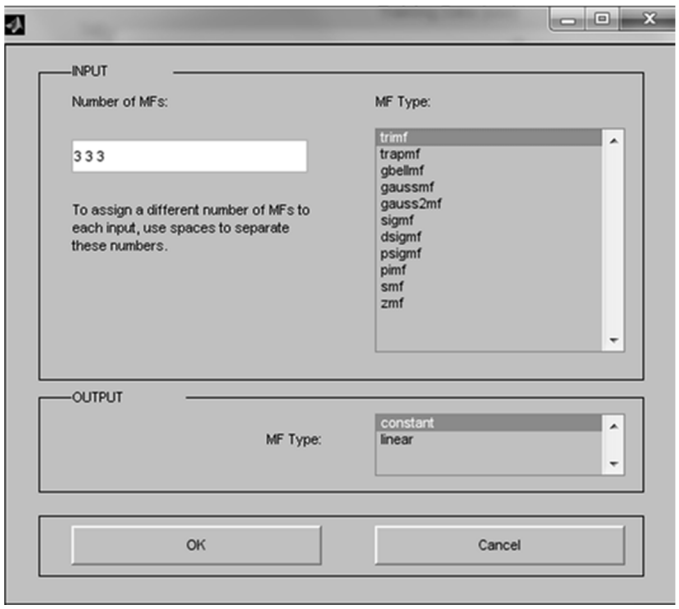


Рисунок 4.4- Діалогове вікно для завдання кількості і типу функцій приналежності

4. Після генерації структури гібридної мережі можна відобразити її структуру, для чого слід натиснути кнопку **Structure** в правій частині графічного вікна (рис. 4.5). Для наведеного прикладу система нечіткого виводу містить три вхідних змінних з трьома термами кожна, 27 правил нечітких продукції, одну вихідну змінну з 27 термами.

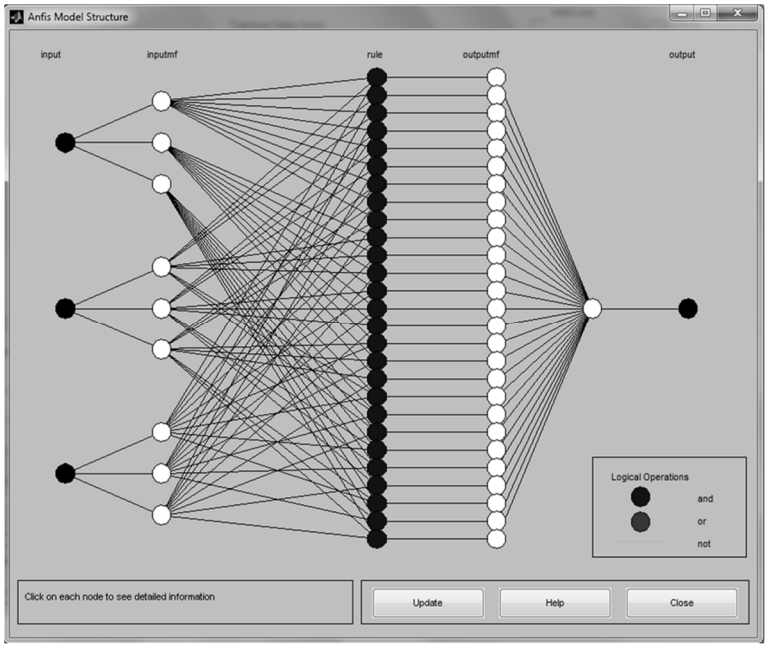


Рисунок 4.5- Структура згенерованої системи нечіткого виводу

5. Перед навчанням гібридної мережі необхідно задати параметри навчання, для чого слід скористатися наступною групою опцій в правій нижній частині робочого вікна :

– вибрати метод навчання гібридної мережі - зворотного поширення (*backpropo*) або гібридний (*hybrid*), такий, що є комбінацією методу найменших квадратів і методу убування зворотного градієнта.

– встановити рівень помилки навчання (*Error Tolerance*) - за умовчанням значення 0. (змінювати не рекомендується)

– задати кількість циклів навчання (*Epochs*) - за умовчанням значення 3 (рекомендується збільшити).

Для навчання мережі слід натиснути кнопку **Train now**. При цьому хід процесу навчання ілюструється у вікні візуалізації у формі графіку залежності помилки від кількості циклів навчання (рис. 4.6).

Подальше налаштування параметрів побудованої і навченої гібридної мережі може бути виконане за допомогою стандартних графічних засобів пакету Fuzzy Logic Toolbox. Для цього рекомендується зберегти створену систему нечіткого виводу в зовнішньому файлі з розширенням *.fis, після чого слід завантажити цей файл редактор систем нечіткого виведення FIS.

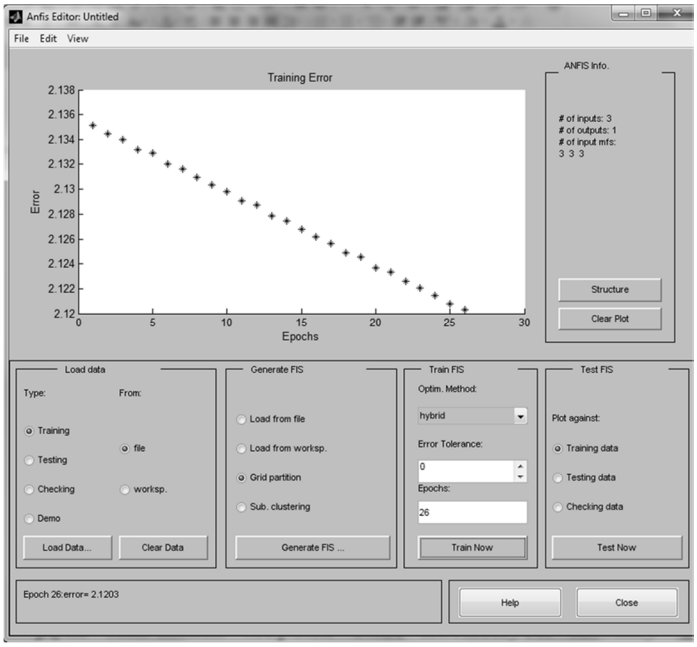


Рисунок 4.6 - Візуалізація графіку залежності помилки від кількості циклів навчання

6. Перевірку адекватності побудованої нечіткої моделі гібридної мережі. Для вирішення цього завдання необхідно скористатися функцією evalfis.

Порядок виконання роботи

1. Підготувати файл з повчальними даними з розширенням *.dat.
2. Завантажити файл з навчальними даними в редактор ANFIS.
3. Згенерувати структуру системи нечіткого виведення FIS типу Сугено
4. Здійснити навчання нейро-нечіткої мережі, заздалегідь задавши параметри навчання
5. Перевірити адекватність побудованої нечіткої моделі гібридної мережі.

Контрольні питання

1. Дайте визначення нейро-нечіткої мережі.
2. Яке призначення мереж нейро-нечіткого виводу?
3. У чому переваги використання нейро-нечітких мереж?
4. Охарактеризуйте структуру нейро-нечіткої мережі.
5. Поясніть процес розробки нейро-нечіткої мережі в середовищі MATLAB.
6. Як впливає задана точність навчання на тривалість навчання?
7. Які вимоги мають пред’являтися до навчаючої вибірки та як це вплине на процес навчання?

ЛІТЕРАТУРА

1. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH. — СПб.: БХВ-Петербург, 2005. — 736 с.

2. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. - М.: Горячая линия - Телеком, 2007. - 288 с.

3. Кофман А. Введение в теорию нечетких множеств. – М.: Радио и связь, 1982. – 432 с.

4. Дьяконов В., Круглов В. Математические пакеты расширения MATLAB: специальный справочник. – СПб.: Питер, 2001. – 480 с.

5. Гостев В.И. Нечеткие регуляторы в системах автоматического управления. - К.: "Радіоаматор". 2008.-972 с.

Додаток А
FIS-структура

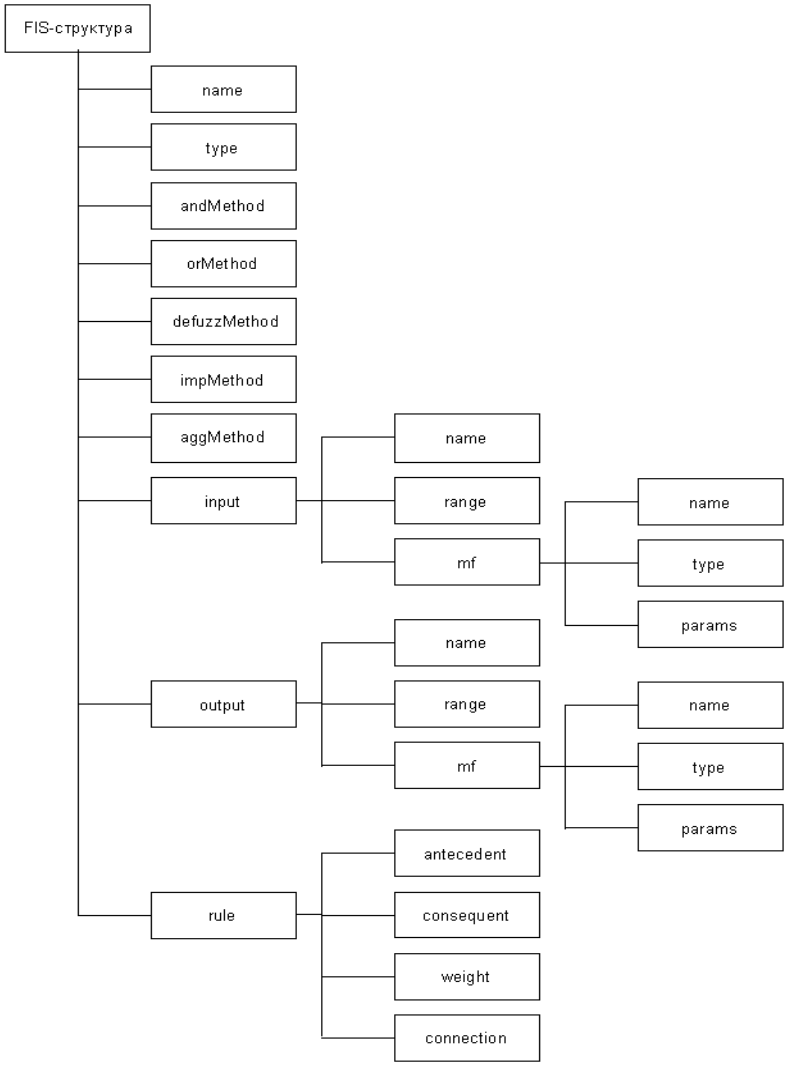


Рисунок А1 - FIS-структура

Поля структури системи нечіткого логічного виводу призначені для збереження наступної інформації:

– *name* - найменування системи нечіткого логічного виводу;

– *type* - тип системи (припустимі значення 'Mamdani' та 'Sugeno');

– *andMethod* - реалізація логічної операції "І" (запрограмовані реалізації: '*min*' – мінімум і '*prod*' – множення);

– *orMethod* - реалізація логічної операції "АБО" (запрограмовані реалізації: 'max' – максимум і 'probor' – імовірнісне "АБО");

– *defuzzMethod* - метод дефазифікації (запрограмовані методи для систем типу Мамдані: '*centroid*' – центр ваги; *'bisector*' – медіана; '*lom*' – найбільший з максимумів; *'som*' – найменший з максимумів; *'mom*' – середнє з максимумів; запрограмовані методи для систем типу Сугено: '*wtaver*' – зважене середнє і '*wtsum*' – зважена сума);

– *impMethod* - реалізація операції імплікації (запрограмовані реалізації: '*min*' – мінімум і 'prod' – множення);

– *aggMethod* - реалізація операції об'єднання функцій приналежності вихідної змінної (запрограмовані реалізації: '*max*' – максимум; '*sum*' – сума і '*probor*' – імовірнісне "АБО");

– *input* - масив вхідних змінних;

– *input.name* - найменування вхідної змінної;

– *input.range* - діапазон зміни вхідної змінної;

– *input.mf* - масив функцій приналежності вхідної змінної;

– *input.mf.name* - найменування функції приналежності вхідної змінної;

– *input.mf.type* - модель функції приналежності вхідної змінної (запрограмовані моделі: dsigmf - функція приналежності у виді різниці між двома сигмоїдними функціями; gauss2mf - двостороння гаусівська функція приналежності; gaussmf - гаусівська функція приналежності; gbellmf - узагальнена колоколообразна функція приналежності; pimf - пі-подібна функція приналежності; psigmf - добуток двох сигмоїдних функцій приналежності; sigmf - сигмоїдна функція приналежності; smf - s-подібна функція приналежності; trapmf - трапецієподібна функція приналежності; trimf - трикутна функція приналежності; zmf - z-подібна функція приналежності);

– *input.mf.params* - масив параметрів функції приналежності вхідної змінної;

– *output* - масив вихідних змінних;

– *output.name* - найменування вихідної змінної;

– *output.range* - діапазон зміни вихідної змінної;

– *output.mf* - масив функцій приналежності вихідної змінної;

– *output.mf.name* - найменування функції приналежності вихідної змінної;

– output.mf.type - модель функції приналежності вихідної змінної (запрограмовані моделі для системи типу Мамдані: dsigmf - функція приналежності у виді різниці між двома сигмоїдними функціями; gauss2mf - двостороння гаусівська функція приналежності; gaussmf - гаусівська функція приналежності; gbellmf - узагальнена колоколообразна функція приналежності; pimf - пі-подібна функція приналежності; psigmf - добуток двох сигмоїдних функцій приналежності; запрограмовані моделі для системи типу Сугено: constatnt – константа (функція приналежності у виді сінглтона); linear – лінійна комбінація вхідних змінних);

– *output.mf.params* - масив параметрів функції приналежності вихідної змінної;

– *rule* - масив правил нечіткої бази знань;

– *rule.antecedent* - посилки правила (вказуються порядкові номери термів у порядку запису вхідних змінних. Число 0 указує на те, що значення відповідної вхідний змінної не впливає на істинність правила);

– *rule.consequent* - висновок правила (вказуються порядкові номери термів у порядку запису вихідних змінних. Число 0 указує на те, що правило не поширюється на відповідну вихідну змінну);

– *rule.weight* - вага правила. Задається числом з діапазону [0, 1];

– *rule.connection* - логічне зв'язування змінних усередині правила: 1 – логічне "І"; 2 – логічне "АБО".

Методичні вказівки до виконання практичних робіт з дисципліни «Нові науково-технічні напрямки електронних систем» для студентів спеціальності 8.090803 «Електронні системи»/ Укл.: к.т.н. доцент Багрій В.В., Дніпродзержинськ ,ДДТУ, 2012, ____с.

Укладач: к.т.н. доцент, Багрій В.В.

Підписано до друку _____2010
Формат Обсяг др. аркуш.
Тираж екз. Замовлення
51918, м. Дніпродзержинськ
вул.. Дніпробудівська, 2