

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ВОЛОДИМИРА ДАЛЯ

МЕТОДИЧНІ ВКАЗІВКИ
до виконання лабораторних робіт з дисципліни
«ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ УПРАВЛІННЯ»
(для здобувачів вищої освіти денної та заочної форми навчання
спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології)
(Електронне видання)

ЗАТВЕРДЖЕНО
на засіданні кафедри комп'ютерно-
інтегрованих систем управління
Протокол №4 від 02.12.2021 р.

СЄВЕРОДОНЕЦЬК 2021

УДК 681.518

Методичні вказівки до виконання лабораторних робіт з дисципліни «Інтелектуальні системи управління» (для здобувачів вищої освіти денної та заочної форми навчання спеціальності 151 Автоматизація та комп'ютерно-інтегровані технології) (Електронне видання) / Укладачі: О.І. Проказа, О.В. Кузнецова. – Сєверодонецьк: Вид-во СНУ ім. В. Даля, 2021. – 123 с.

У методичних вказівках викладені матеріали, необхідні для виконання лабораторних робіт з дисципліни «Інтелектуальні системи управління». Методичні вказівки містять в собі необхідні теоретичні відомості по темам лабораторних робіт, приклади рішення типових завдань в системі MatLab 6.5, що дозволяють одержати поглиблене уявлення про суть розглянутого питання, а також варіанти індивідуальних завдань та список літератури.

Укладачі:

О.І. Проказа, к.т.н., доц.

О.В. Кузнецова, ст. викл.

Рецензент

М.Г. Лорія, д.т.н., проф.

ЗМІСТ

ВСТУП.....	4
ЗАГАЛЬНІ ВКАЗІВКИ ДО ВИКОНАННЯ РОБІТ.....	6
ЛАБОРАТОРНА РОБОТА №1.....	7
ЛАБОРАТОРНА РОБОТА №2.....	27
ЛАБОРАТОРНА РОБОТА №3.....	44
ЛАБОРАТОРНА РОБОТА №4.....	64
ЛАБОРАТОРНА РОБОТА №5.....	84
ЛАБОРАТОРНА РОБОТА №6.....	94
ЛАБОРАТОРНА РОБОТА №7.....	108
СПИСОК ЛІТЕРАТУРИ.....	122

ВСТУП

Методичні вказівки складені відповідно до робочої програми навчальної дисципліни “Інтелектуальні системи управління” спеціальності 151 – Автоматизація та комп’ютерно–інтегровані технології.

«Інтелектуальні системи управління» – це вибіркова дисципліна, в якій вивчають технології побудови сучасних комп’ютерно-інтегрованих систем керування.

Дисципліна спрямована на формування теоретичних знань та практичних навичок з моделювання інтелектуальних систем управління, що передбачають використання нечітких логічних регуляторів, генетичних алгоритмів і сучасних технологій обчислень та обробки знань з метою покращення ефективності, стійкості та надійності процесів керування в об’єктах автоматизації.

При вивченні дисципліни здобувачі отримують навички використання технологій та інструментальних засобів інтелектуального аналізу даних для проектування інтелектуальних систем управління різними технологічними об’єктами в умовах невизначеності, а також застосування апарату нечіткої логіки та генетичних алгоритмів в автоматичних регуляторах і системах автоматичного керування.

У результаті вивчення дисципліни студент зобов’язаний знати:

- архітектуру системи нечіткого виведення;
- основні поняття і визначення, використовувані в сучасних нечітких системах;
- основні етапи створення гібридних інтелектуальних систем у середовищі MatLab;
- класифікацію функцій приналежності для побудови нечітких систем;
- алгоритми нечіткого виводу;
- призначення генетичних операторів;

вміти:

- використовувати пакет Fuzzy Logic середовища Matlab 6.5 для побудови нечітких систем;
- використовувати ANFIS-редактор для синтезування нейро-нечітких систем;
- створювати у середовищі Matlab 6.5 системи нечіткого виводу типу Мамдані та Сугено;
- тестувати отримані системи нечіткого виводу;
- застосовувати генетичні алгоритми для вирішення задач оптимізації.

ЗАГАЛЬНІ ВКАЗІВКИ ДО ВИКОНАННЯ РОБІТ

1. Порядок виконання лабораторних робіт

Для виконання лабораторних робіт необхідно уважно ознайомитися з теоретичними основами кожної роботи, приведеними в методичних вказівках. Для більш глибокого ознайомлення з матеріалом, необхідно звертатися до конспекту лекцій, а також до технічної літератури, на яку приводяться посилання у методичних вказівках.

Якщо під час вивчення дисципліни з'являться запитання, на які не знаходиться відповідь у конспекті лекцій або технічній літературі, необхідно звертатись за консультацією до ведучого лектора.

Лабораторні роботи необхідно виконувати за допомогою комп'ютера в системі MatLab 6.5.

Лабораторна робота складається з:

- одержання індивідуального завдання;
- одержання шуканих результатів;
- оформлення й захист звіту по лабораторній роботі.

2. Правила оформлення звітів з лабораторних робіт

Звіт повинен містити:

- титульний аркуш із найменуванням лабораторної роботи й даними виконавця;
- мета роботи;
- постановка задачі відповідно до варіанту;
- короткий опис виконаного завдання;
- аналіз результатів і висновки.

Всі матеріали звіту необхідно роздрукувати на листах формату А4, зброшурувати, сторінки пронумерувати.

Лабораторна робота № 1

Тема: ЗАСОБИ MATLAB ДЛЯ ПОБУДОВИ СИСТЕМ НЕЧІТКОГО ВИВОДУ

Мета роботи:

- закріпити знання по розділу нечіткі системи управління, систематизувати ці знання, навчитися реалізовувати моделі в системі Matlab;
- отримати навички створення системи нечіткого виведення типу Мамдані.

Постановка завдання: прогнозування засобами Matlab.

Короткі теоретичні відомості

Поняття нечіткої множини

Значний крок у напрямі розвитку теорії нечіткої множини зробив професор Каліфорнійського університету Лотфі А. Заді. Він розширив поняття множини, допустив, що характеристична функція (функція приналежності елементу множини) може набувати будь-яких значень в інтервалі $[0, 1]$. Така множина була названа ним нечіткими (fuzzy).

Хай E – універсальна множина, x – елемент E , P – деяка властивість. Звичайна (чітке) множина A універсальної множини E , елементи якої задовольняють властивості P , визначаються як множина впорядкованих пар:

$$A = \{\mu_A(x) / x\},$$

де μ_A – характеристична функція, що набуває значення 1, якщо x задовольняє властивості P , і 0 – інакше.

У теорії нечіткої множини для елементів x з E немає однозначної відповіді «так/ні» відносно властивості P . У зв'язку з цим нечітка множина A універсальної множини E визначається як множина впорядкованих пар з функцією

приналежності, що набуває значення в деякій впорядкованій множині M (наприклад, $M=[0, 1]$).

Функція принадлежности вказує міру (або рівень) принадлежности елемента x підмножині A . Множину M називають множиною принадлежности. Якщо $M=\{0, 1\}$, то нечітка підмножина A може розглядатися як звичайна або чітка множина.

Приклади запису нечіткої множини.

1. Хай $E=\{x_1, x_2, x_3, x_4, x_5\}$; A – нечітка множина, для якої $\mu_A(x_1)=0,3$; $\mu_A(x_2)=0$; $\mu_A(x_3)=1$; $\mu_A(x_4)=0,6$; $\mu_A(x_5)=0,9$.

Тоді A можна представити у вигляді

$$A = \{0,3/x_1; 0/x_2; 1/x_3; 0,6/x_4; 0,9/x_5\}.$$

2. Хай $E=\{1, 2, 3, \dots, 100\}$ і відповідає поняттю «віку», тоді нечітка множина «молодою» може бути визначена за допомогою виразу:

$$\mu_{\text{«молодий»}} = \begin{cases} 1, & x \in [1,25], \\ \frac{1}{1 + \left(\frac{x-25}{5}\right)^2}, & x > 25. \end{cases}$$

У наведених вище прикладах використані прямі методи визначення функцій принадлежности, коли експерт задає значення для кожного $x \in E$. Такий спосіб використовується для вимірюваних понять (швидкість, температура і так далі) або коли виділяють полярні значення. Різновидом прямого способу завдання функції принадлежности є груповий метод, коли групі експертів пропонують зробити оцінку того або іншого явища, наприклад, оцінити: «ця людина лиса» чи ні, - тоді кількість ствердних відповідей, що ділиться на загальне число експертів, дає значення $\mu_{\text{«лисий»}}$ для даної особи.

Окрім вказаних способів завдання функцій принадлежности використовують також типові форми функцій принадлежности (рис. 1).

Аналітичний запис деяких типових функцій принадлежности:

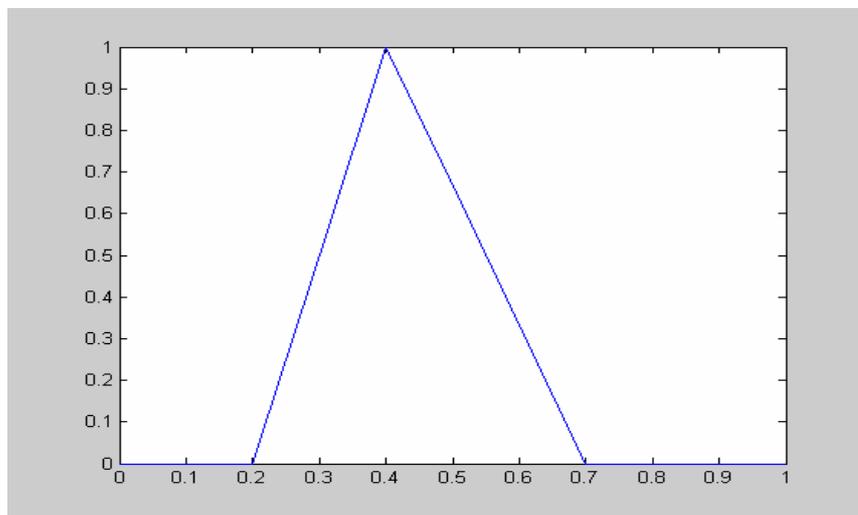
- трикутна - $\mu(x) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right)$, a, b, c - визначають

параметри функції;

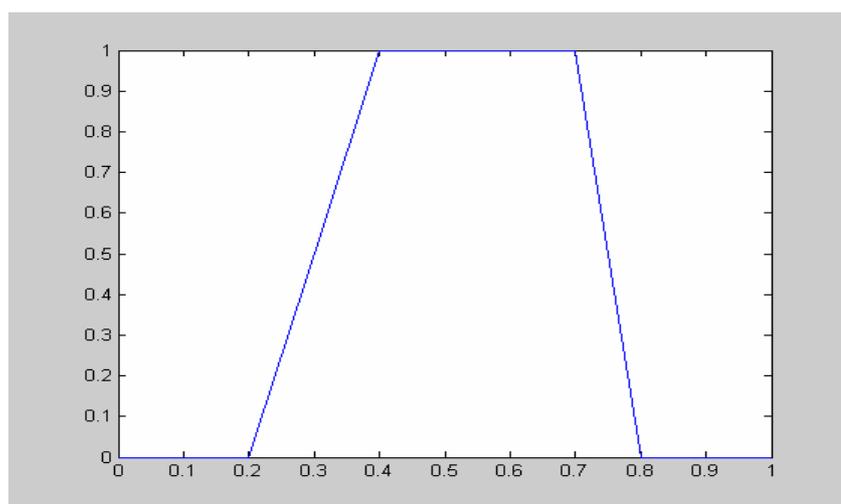
- гаус - $\mu(x) = e^{-\left(\frac{x-a}{b}\right)^2}$, a, b - параметри функції приналежності;

- сигмоїдна - $\mu(x) = \frac{1}{1 + \exp(-a(x-b))}$, a, b - параметри функції

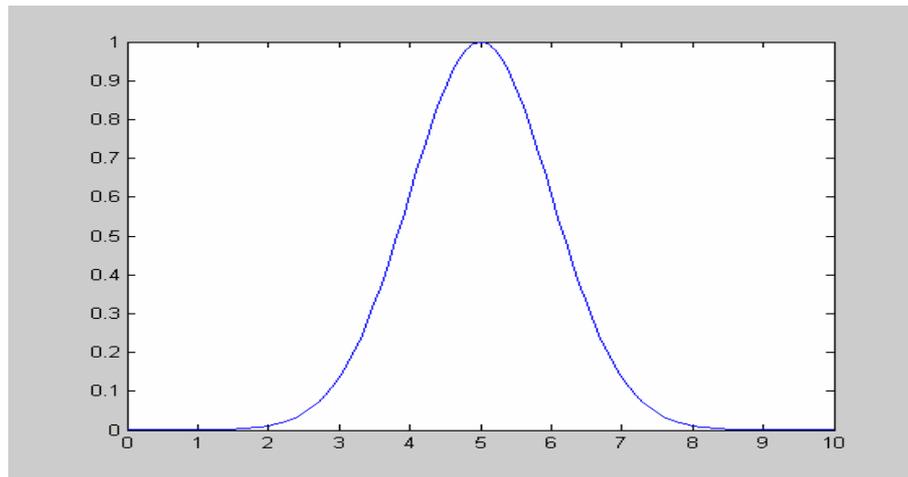
приналежності.



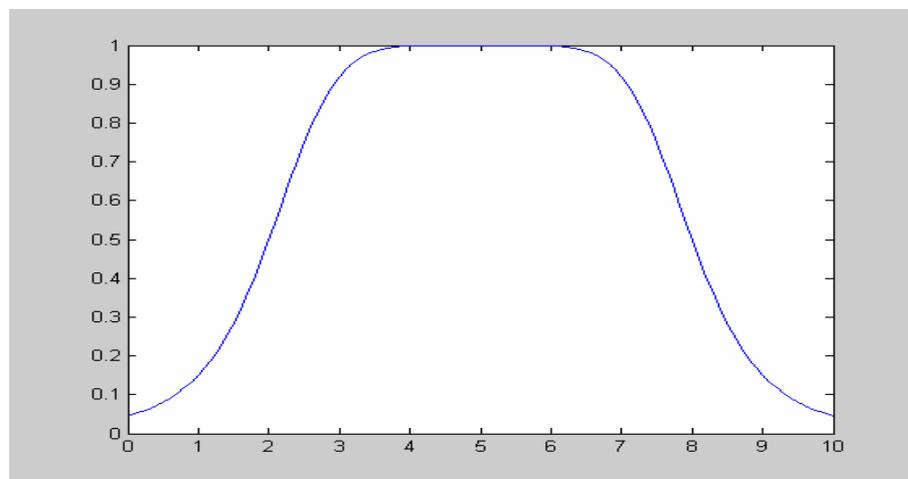
а) трикутна (trimf)



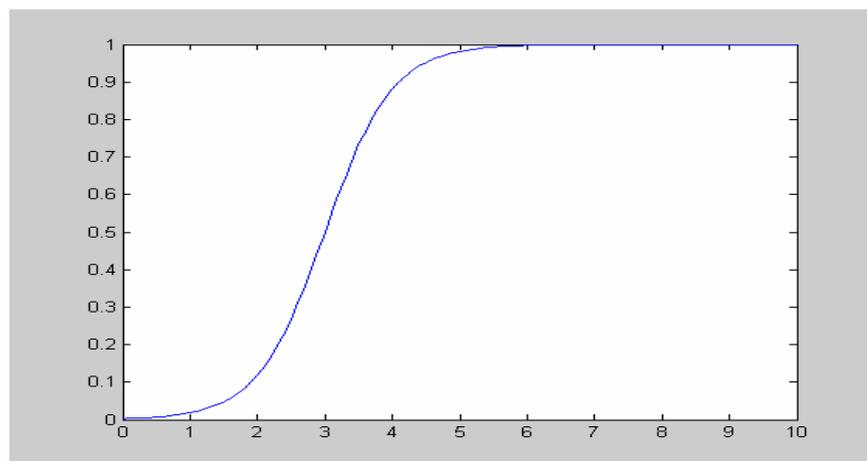
б) трапецеїдальна (trapmf)



в) гаусова (gaussmf)



г) узагальнена дзвоноподібна (gbellmf)



д) сигмоїдна (sigmf)

Рис. 1. Приклади типових форм функцій приналежності
(в дужках наводяться позначення функції в середовищі MATLAB)

Завдання 1. Створити систему нечіткого виводу для оцінки успішності студента за п'ятибальною шкалою на підставі даних про його відвідуваність (0-100%) і активність на заняттях (0-5 балів).

Порядок виконання роботи

1. Завантажити систему Matlab. Командою `fuzzy` з режиму командного рядка Matlab запусити пакет Fuzzy Logic

2. Перш ніж переходити до створення системи необхідно вибрати її тип: Мамдані (Mamdani) або Сугено (Sugeno). Вибираємо систему Mamdani (вона стоїть за умовчанням). Збережемо її на диску з ім'ям `lr1`. Для збереження опису системи нечіткого логіків використовуємо команду `File->Export->To Disk`.

3. Створити необхідну кількість входів і виходів системи. Спочатку в системи один вхід `Input1`. Аби додати другий вхід, необхідно скористатися меню `Edit->Add Variable.->Input`. Аналогічно можна додати в систему і вихід, але в даному випадку за умовчанням один вихід є (рис. 2).

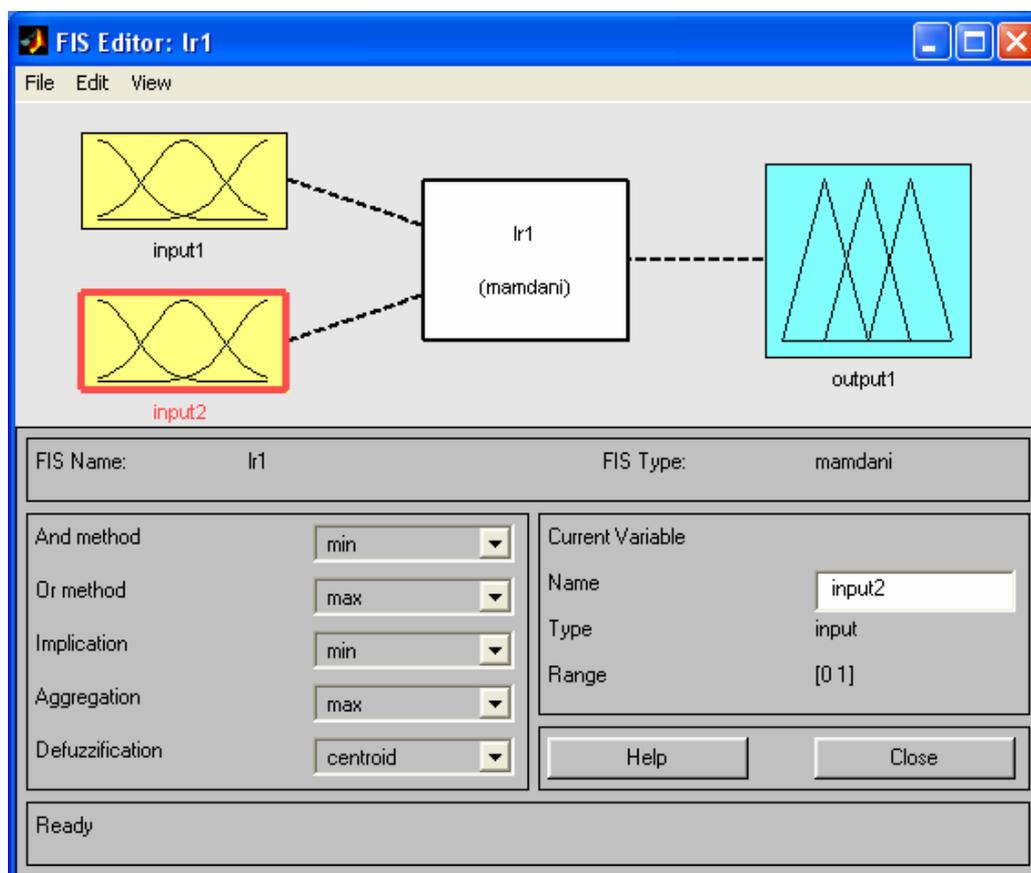


Рис. 2. Додавання входу

4. Перейменувати входи і виходи. Ця процедура обов'язкова і надалі необхідна для складання правил. Для того, щоб перейменувати вхід або вихід, потрібно виділити його (при цьому довкола з'явиться жирна червона рамка) і у вікні Name змінити назви (Input1, Input2, Output1) на необхідні нам (poseshaemost, aktivnost, oценка). Результати перейменування приведені на рис. 3.

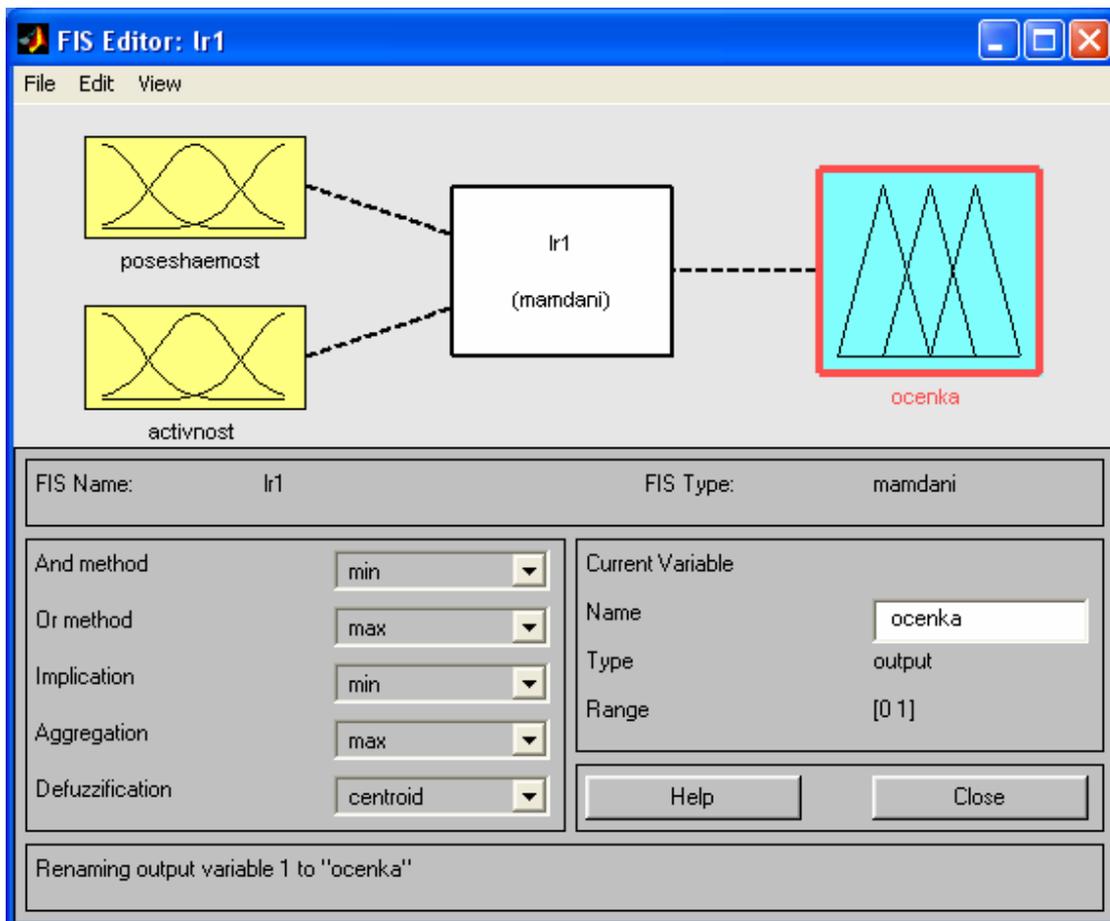


Рис. 3. Перейменовані входи і виходи системи нечіткого виводу

5. Відредагувати функції приналежності. Для редагування призначений редактор Membership Function Editor, увійти до якого можна або шляхом подвійного натиснення на будь-який вхід або вихід системи, або за допомогою меню Edit->Memberships Functions. Перш ніж редагувати будь-яку з функцій, її треба виділити. Хай це буде вхід «poseshaemost». При цьому рамка довкола нього стає червоного кольору. За умовчанням для входів і виходів використовуються трикутні функції помилки. Замінимо їх на функції Гауса. Для цього спочатку видалимо існуючі функції за допомогою меню Edit->Remove all MFs, потім

додамо 3 функції Гауса за допомогою меню Edit->Add MFs. У вікні, що з'явилося, слід вибрати тип функції (gaussmf) і їх кількість (3) і натиснути ОК (рис. 4).

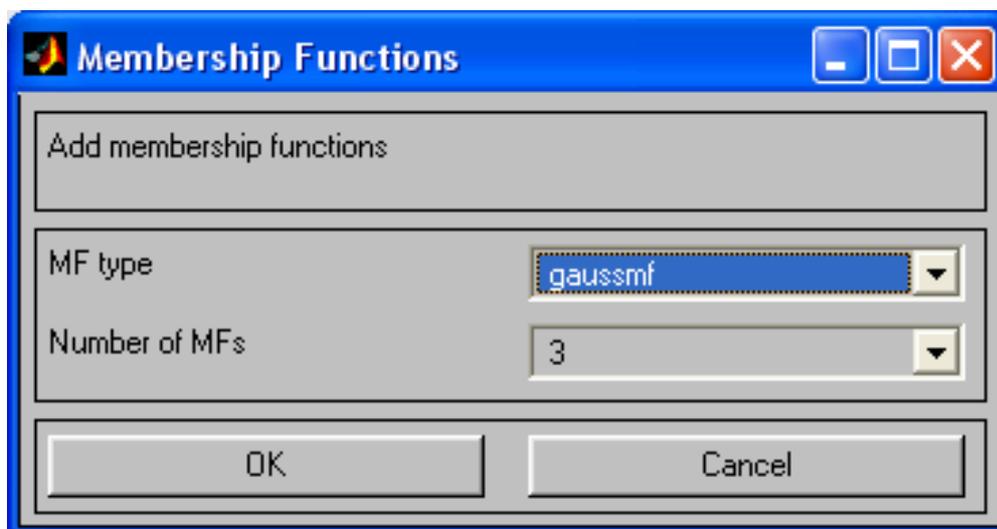


Рис. 4. Вікно установки типу і кількості функції приналежності

Потім слід задати діапазон зміни функцій. Для входу «poseshaemost» він буде від 0 до 100% (кількість відвідуваних лекцій у відсотках). Діапазон задається в лівій нижній частині вікна редактора функцій приналежності в полі Range.

Далі потрібно перейменувати функції приналежності. Перейменування здійснюється шляхом виділення функції (наприклад, mf1), що цікавить, і заміною в полі Name поточної назви на нове ім'я. Вочевидь, що максимум функції приналежності mf1 відповідає поганій («plohaya») відвідуваності, максимум функції mf2 – середньою («srednyaya»), максимум функції mf3 – хорошою («horoshaya»). Результати редагування функції приналежності входу «poseshaemost» приведені на рис. 5.

Аналогічні дії необхідно провести над входом «activnost» і виходом «ocenka».

При цьому діапазон функції приналежності біля входу «activnost» має бути від 1 до 5 (п'ятибальна оцінка) Кількість функцій приналежності біля входу «activnost» – 3 (plohaya, srednaya, horoshaya) (рис. 6).

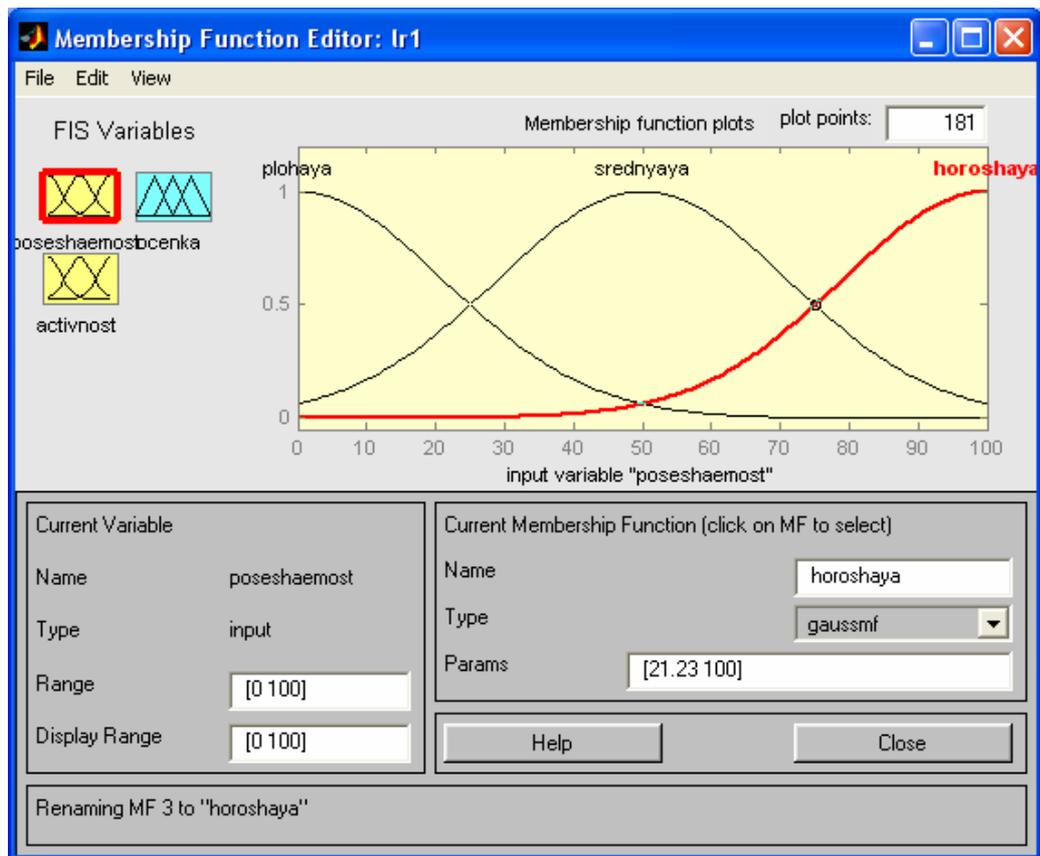


Рис. 5. Редагування функцій приналежності входу «poseshaemost»

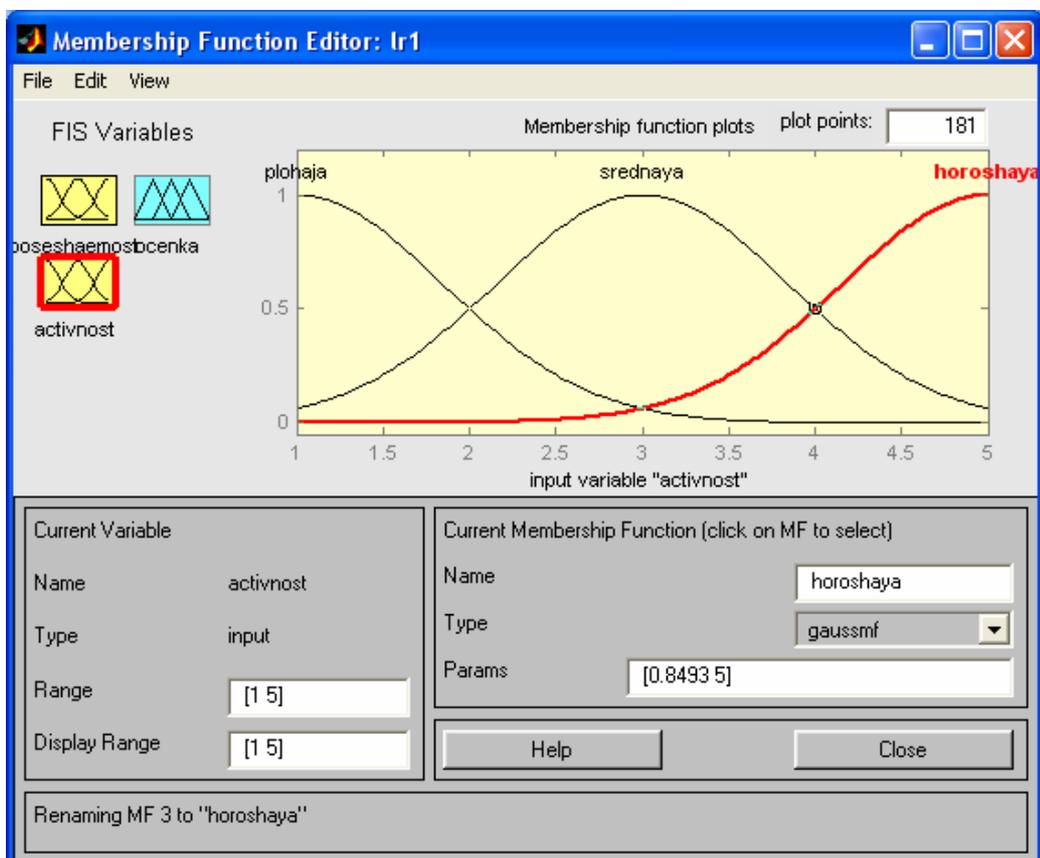


Рис. 6. Редагування функцій приналежності другого входу

Для виходу «осенка» додамо 4 трапецеїдальних функції типа (trapmf) (рис. 7).

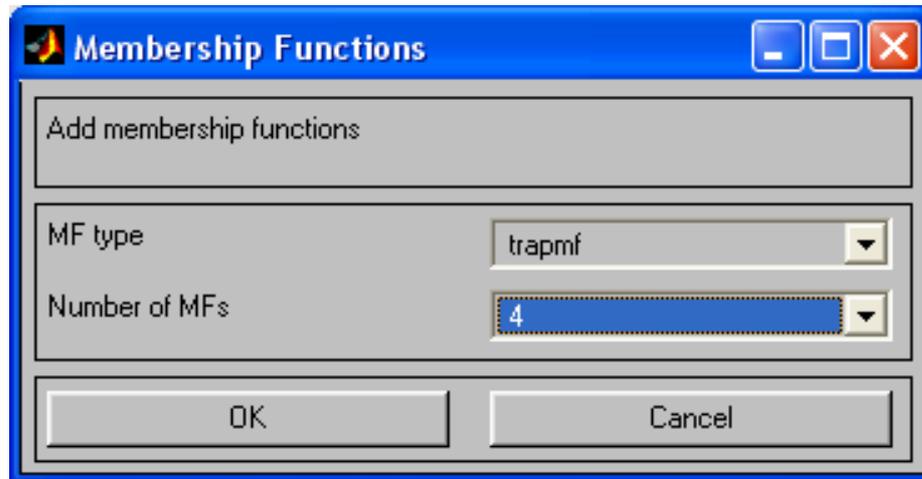


Рис. 7. Вікно установки типу і кількості функції приналежності

Перейменуємо функції приналежності біля виходу на «осенка» на «neudov», «udovlet», «horosho», «otlichno». Для них задамо діапазон від 2 до 5.

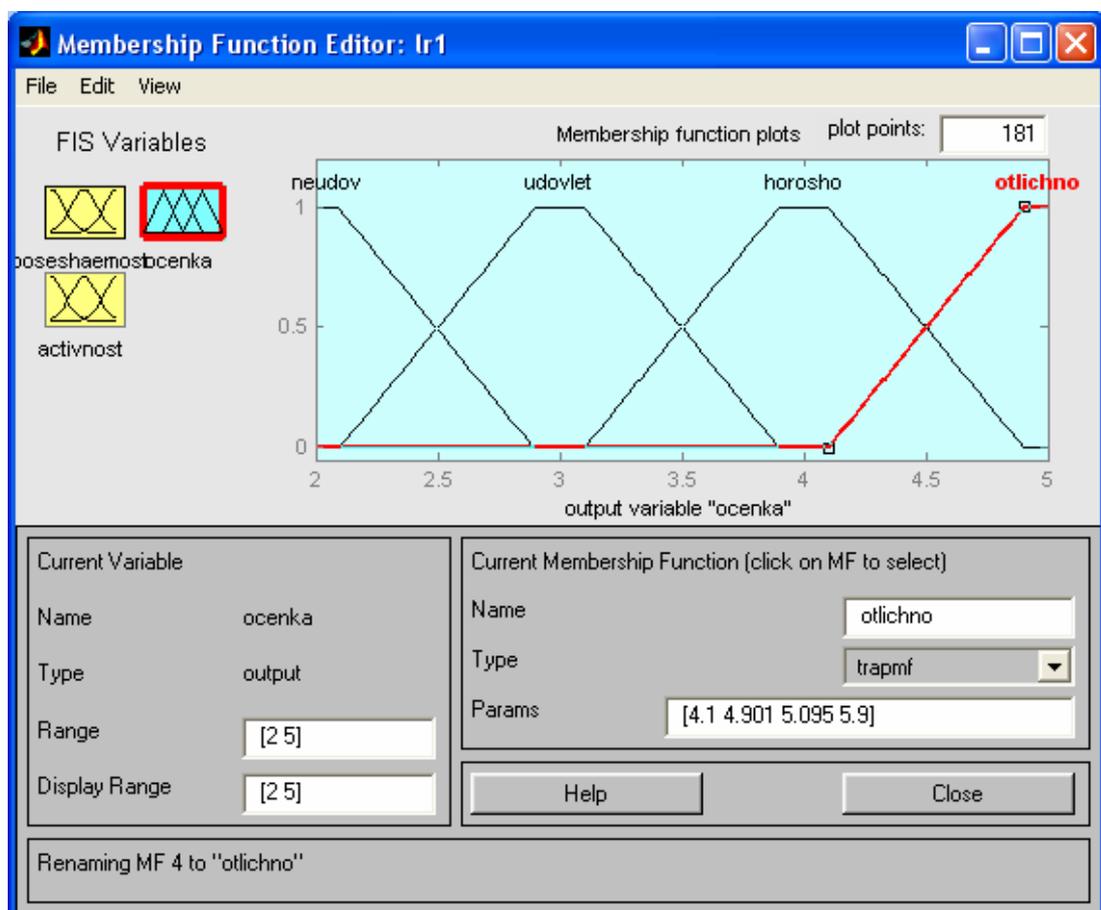


Рис. 8. Редагування функцій приналежності другого виходу

6. Створити базу правил. Правила створюються в редакторі правил Rule Editor, перейти до якого можна за допомогою меню Edit – >Rules.. Для створення першого правила необхідно виділити послідовно функції приналежності plohay (poseshaemost), plohay (activnost) і neudov (ocenka) і натискувати кнопку Add rule. Результат дії приведений на рис. 9, що відповідає правилу: «Якщо відвідуваність погана і активність погана, то оцінка neudov».

Аналогічним чином вводяться останні 8 правил (рис. 10).

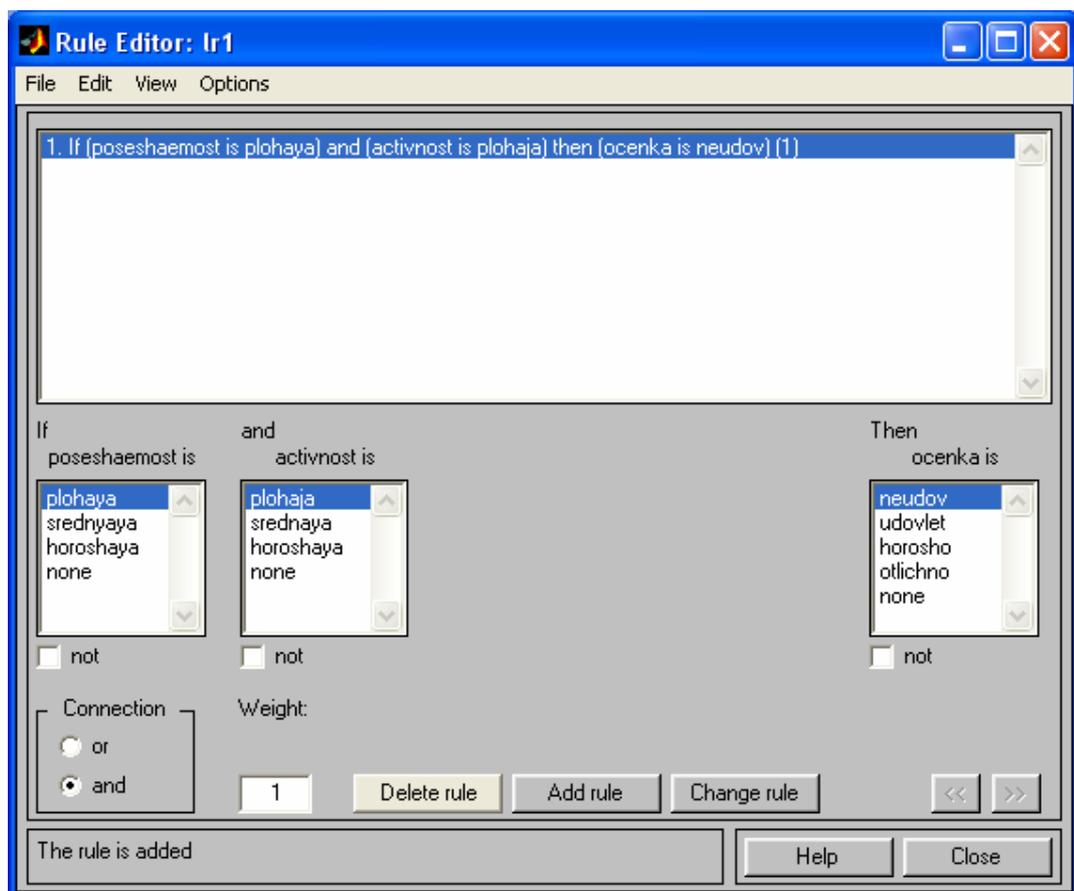


Рис. 9. Створення бази правил

7. Протестувати отриману систему нечіткого виводу.

Спостерігати роботу створеної системи можна в переглядачі правил Rule Viewer, перейти до якого можна за допомогою меню View->Rules (рис. 11).

Вертикальні червоні лінії у входів можна переміщати, отримуючи, таким чином, відповідь. У конкретному випадку видно, що при повній відсутності на лекціях і поганій активності система ставить оцінку 2,61 балу.

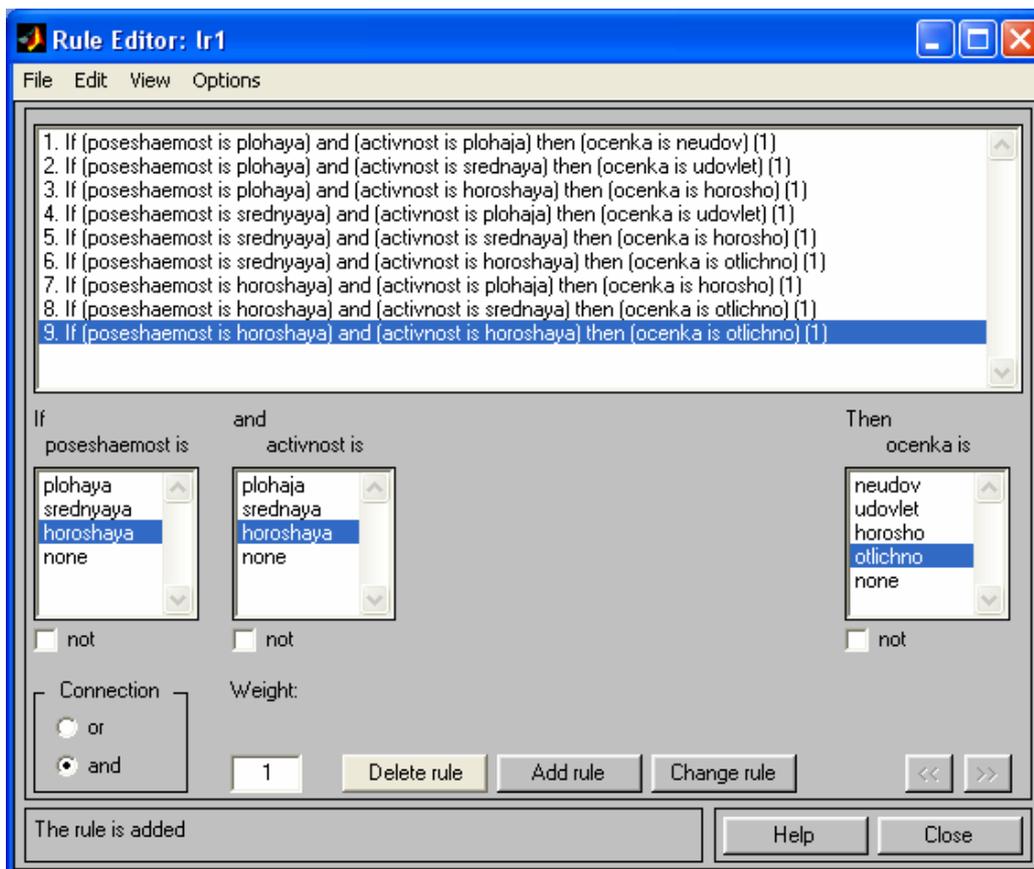


Рис. 10. Готова база правил

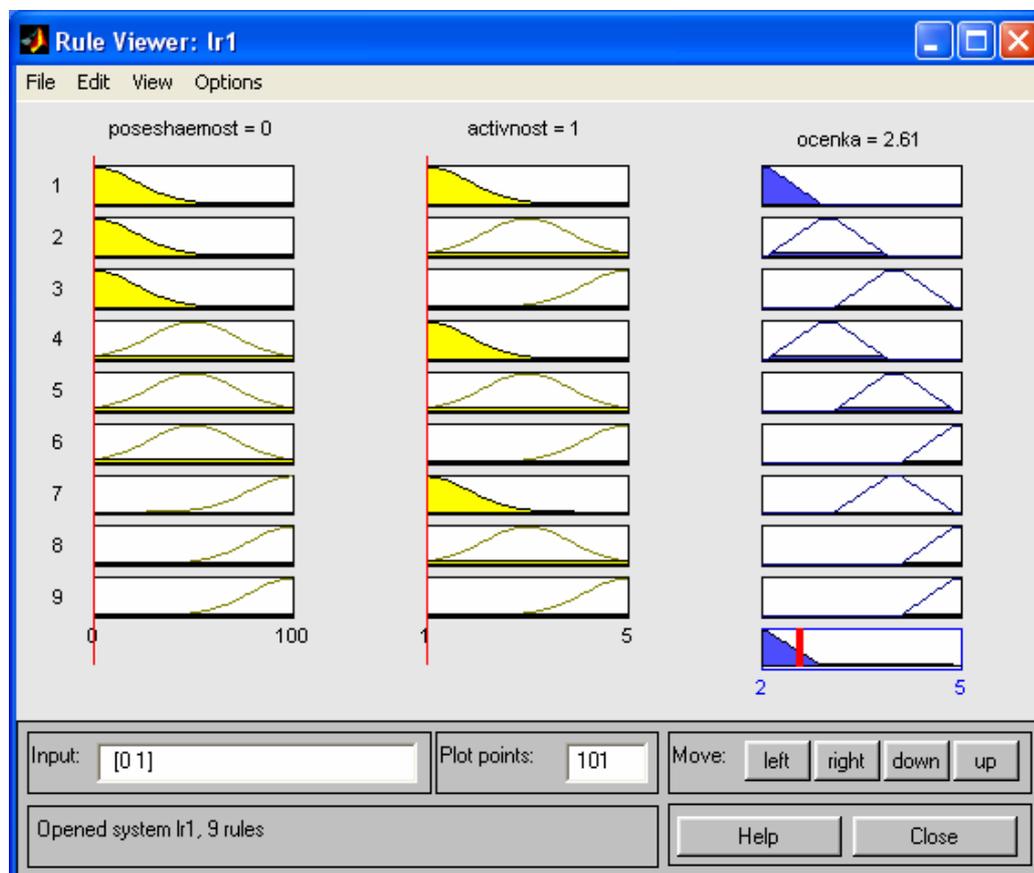
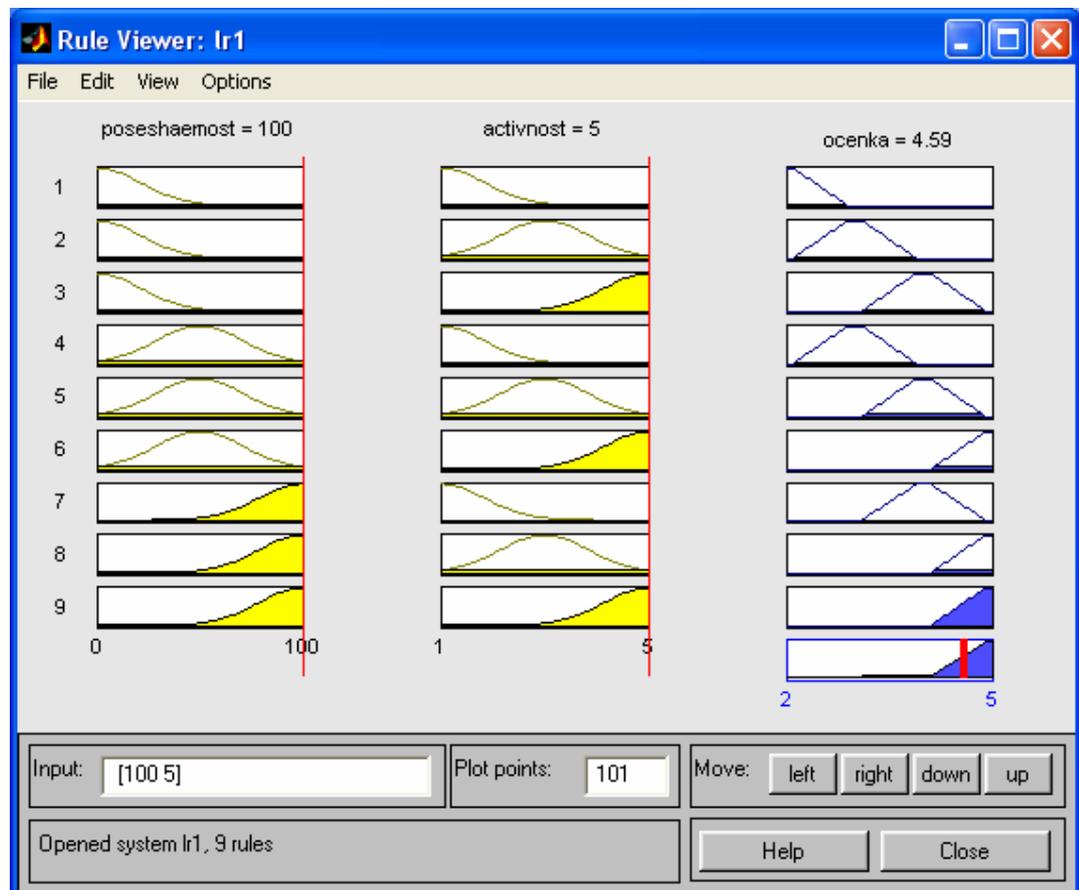
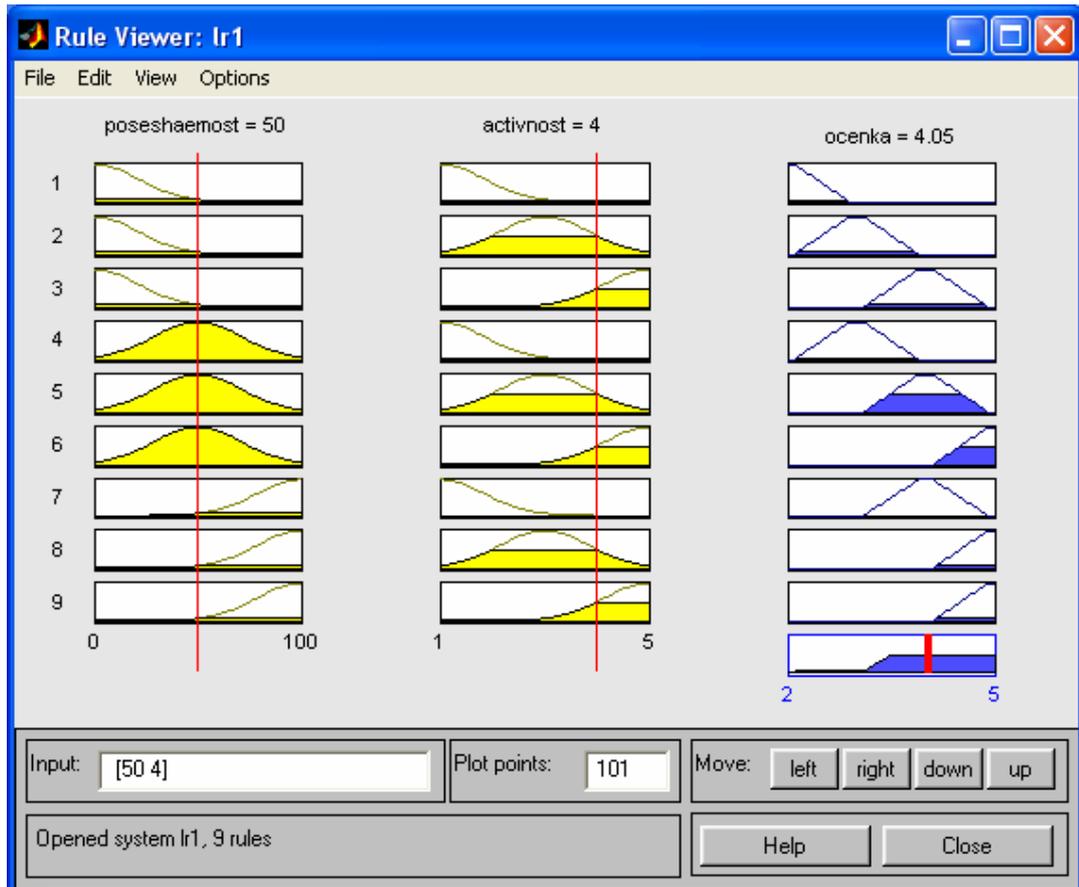


Рис. 11. Вікно переглядача правил

Інші варіанти тестування:



Варіанти завдання 1

Створити систему нечіткого виводу для оцінки успішності студента за стобальною шкалою на підставі даних про його відвідуваність (0-100%) і активність на заняттях (0-10 балів). Функції приналежності згідно варіанту вказані в таблиці 1.

Таблиця 1

Варіанти завдання

Варіант	1	2	3	4	5	6
mf input	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf	trimf
mf output	gaussmf	gauss2mf	gbellmf	dsigmf	trimf	trapmf

Варіант	7	8	9	10	11	12
mf input	gaussmf	gauss2mf	gbellmf	dsigmf	trimf	trapmf
mf output	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf	trimf

Варіант	13	14	15	16	17	18
mf input	gbellmf	gaussmf	gauss2mf	dsigmf	trimf	trapmf
mf output	trapmf	trimf	dsigmf	gaussmf	gbellmf	gauss2mf

Завдання 2. Створити двовимірну модель кондиціонера. Вхідні параметри: температура усередині приміщення (16-35°C) і швидкість зміни температури в приміщенні (від 0 до 2 °C у годину).

Вихідні параметри: швидкість обертання вентилятора (50-1000 об/хв) і витрата рідини, що охолоджує (0-1 мл/год).

Порядок виконання роботи

1. Створити модель з двома входами і двома виходами в системі Matlab:

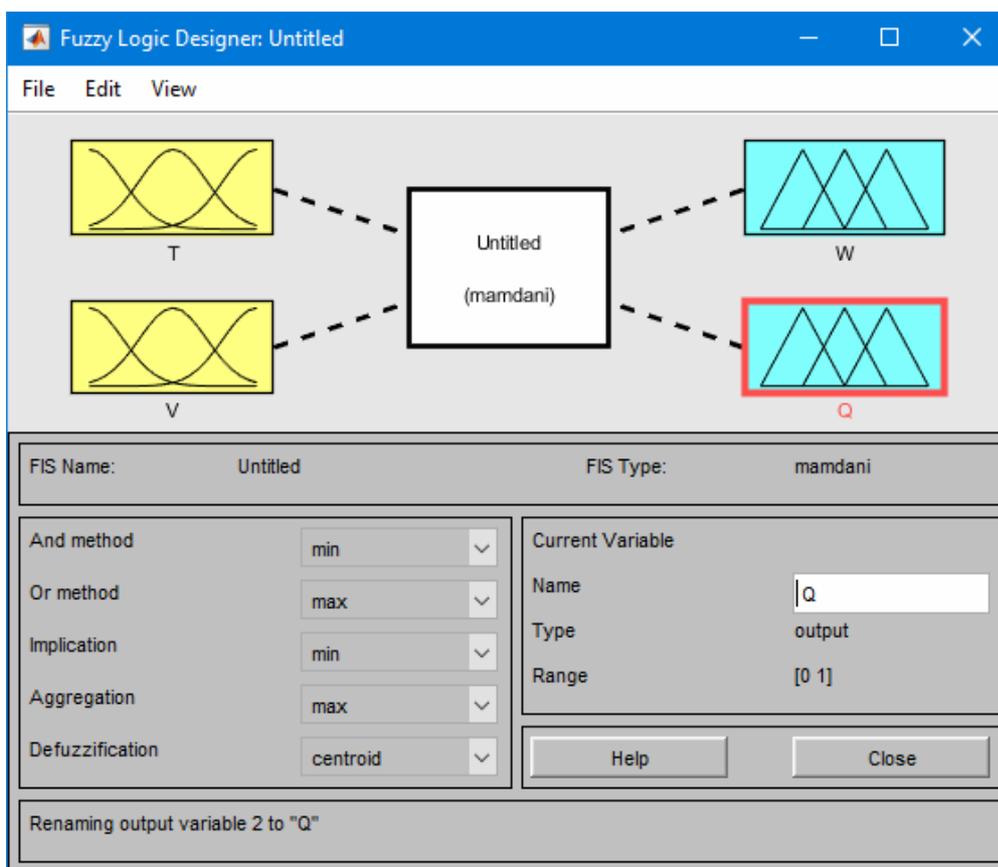


Рис. 12. Додавання входів і виходів

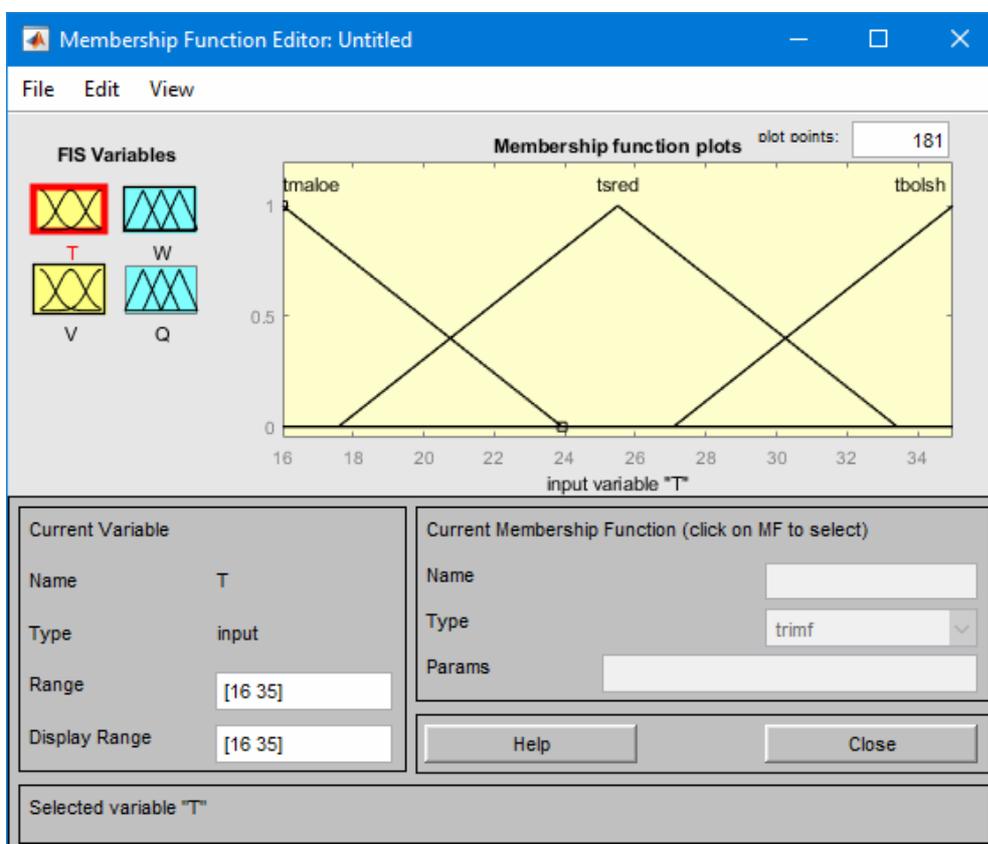


Рис. 13. Редагування функцій приналежності для вхідного сигналу «температура усередині приміщення»

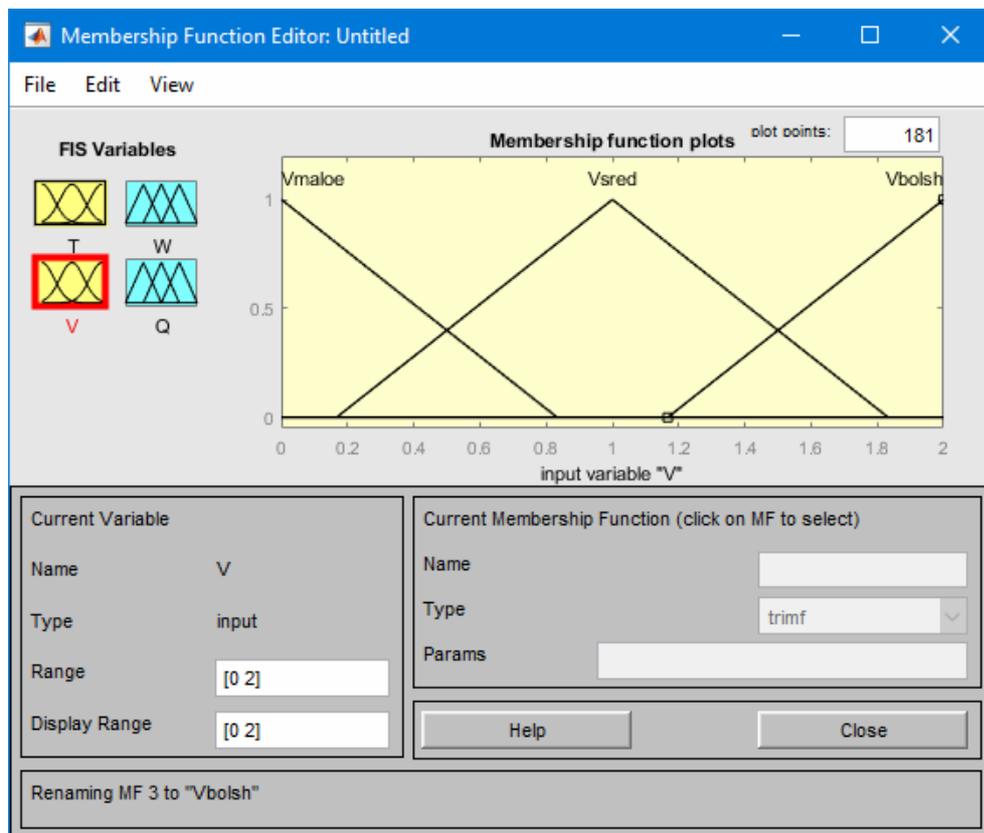


Рис. 14. Редагування функцій приналежності для вхідного сигналу «швидкість зміни температури в приміщенні»

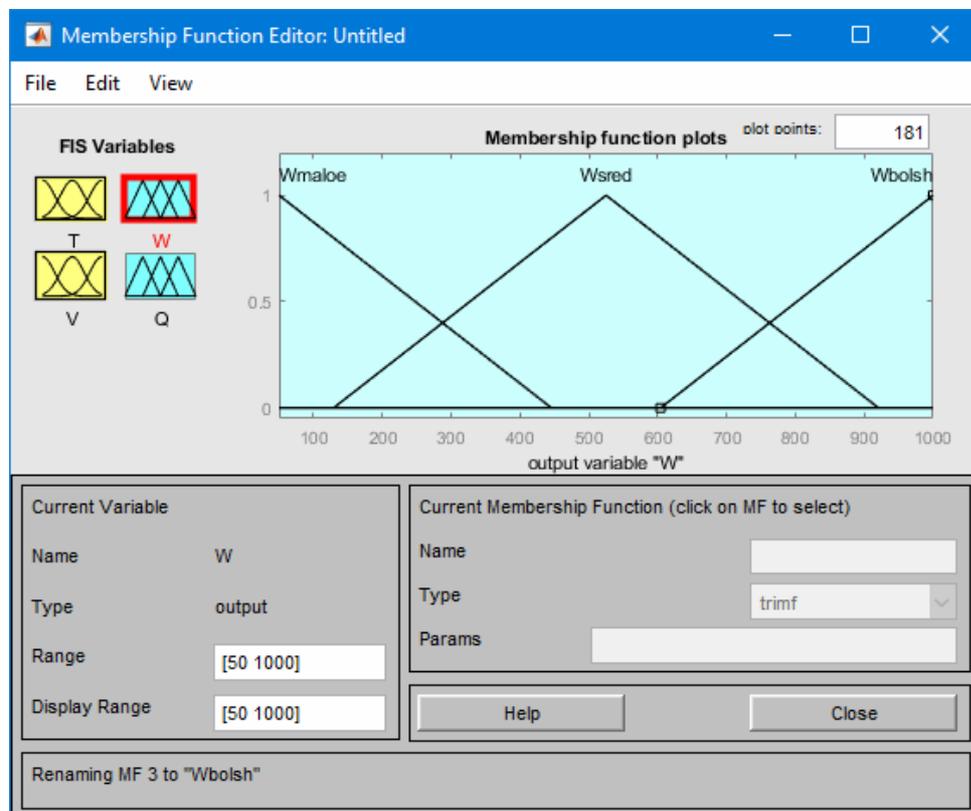


Рис. 15. Редагування функцій приналежності для вихідного сигналу «швидкість обертання вентилятора»

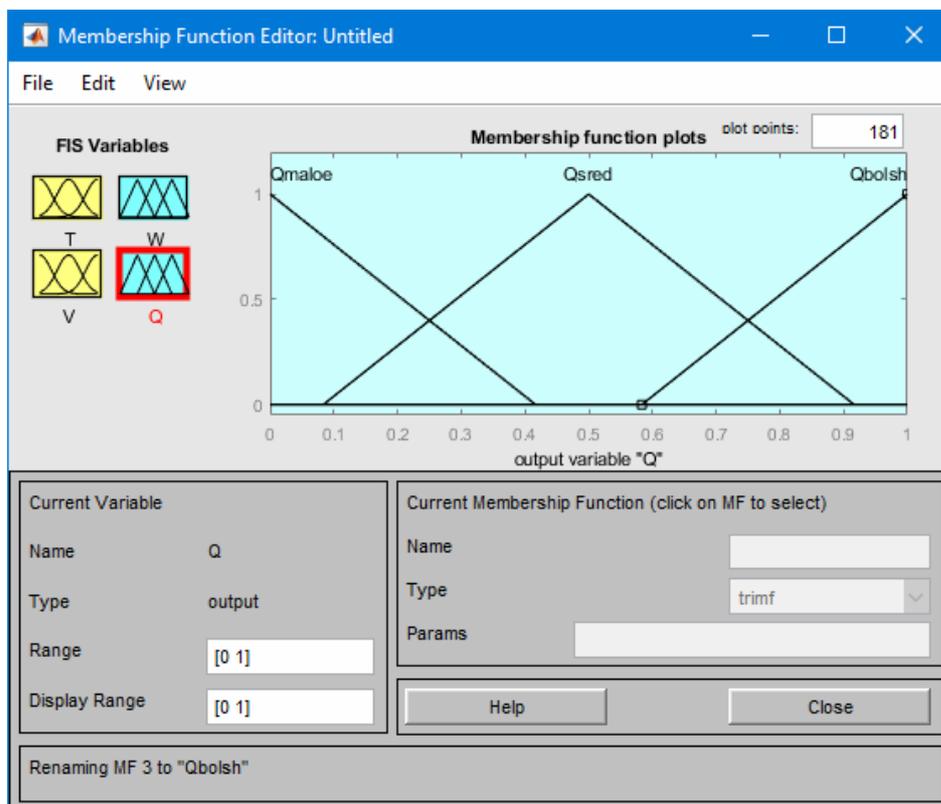


Рис. 16. Редагування функцій приналежності для вихідного сигналу «витрата рідини, що охолоджує»

2. Створити базу правил для завдання режимів роботи кондиціонера.

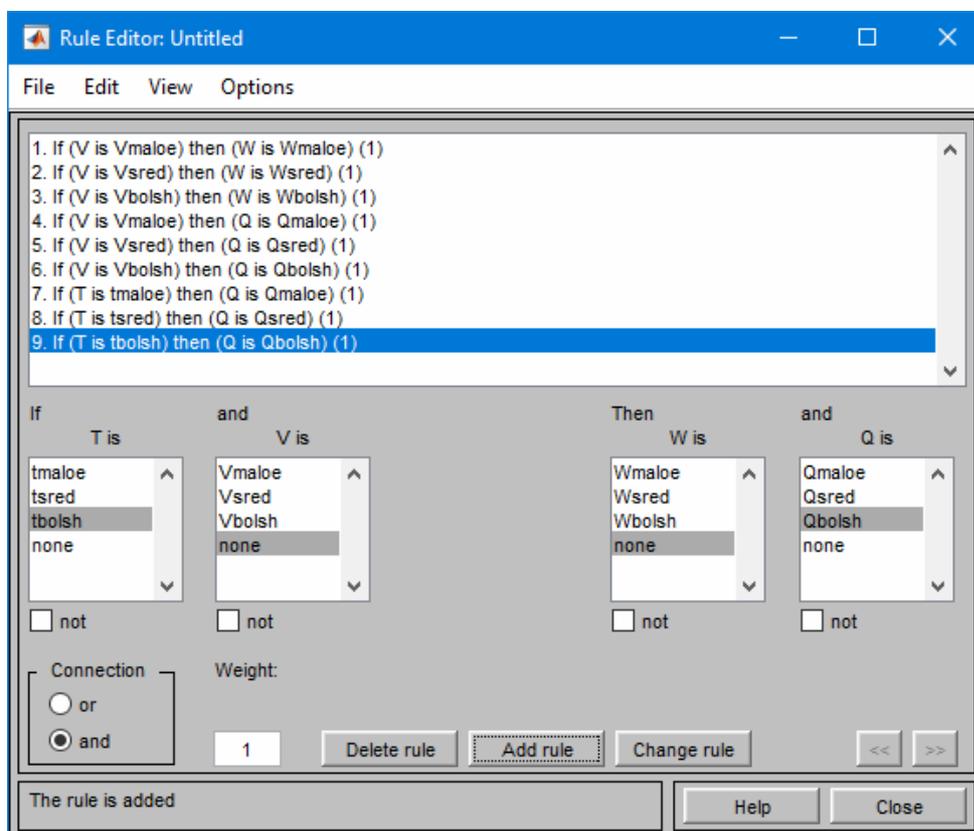


Рис. 17. Готова база правил

3. Протестувати отриману систему нечіткого виводу.

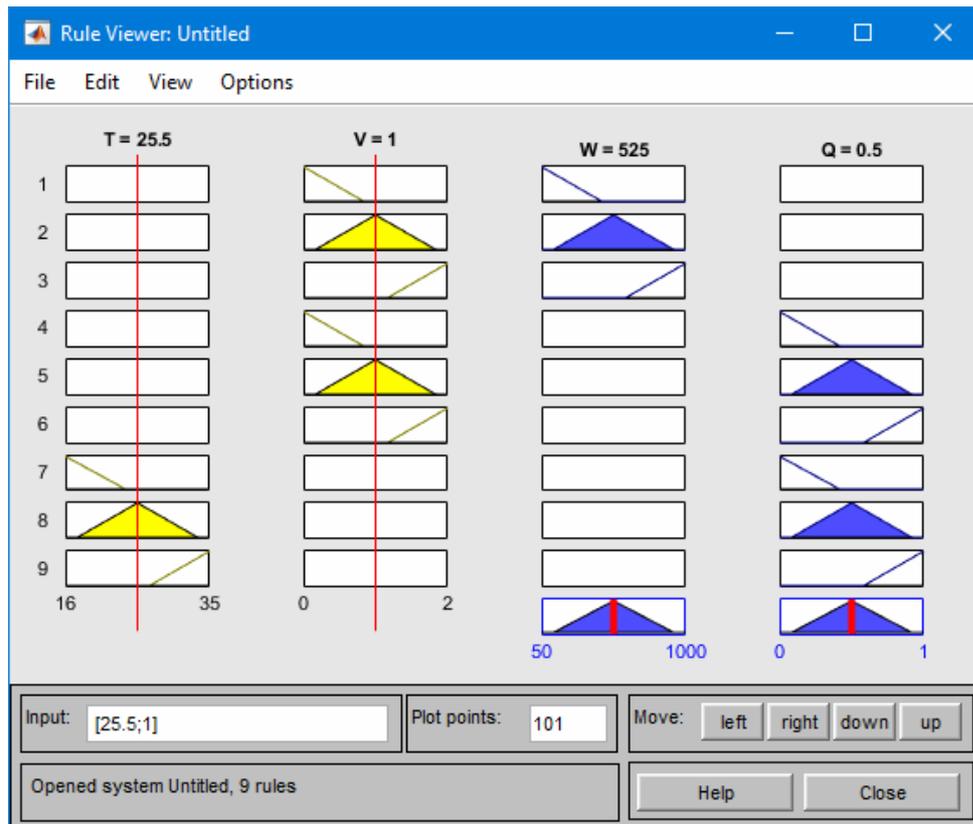


Рис. 18. Вікно переглядача правил

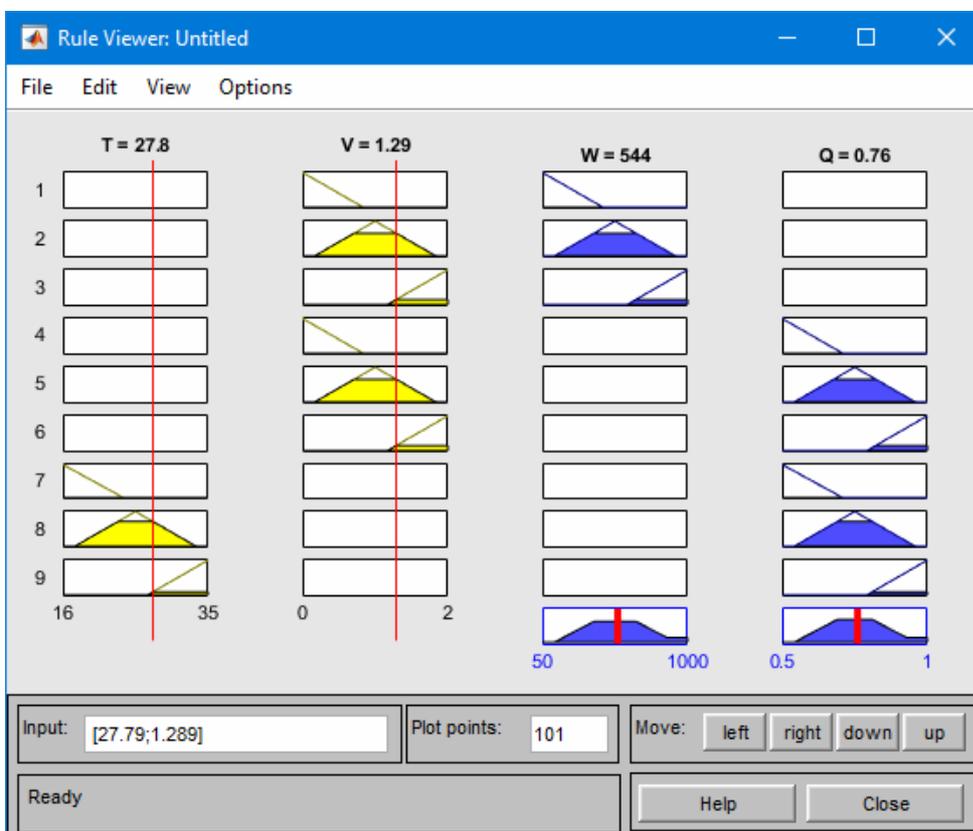


Рис. 19. Тестування системи при зміні вхідних параметрів

4. Аналіз отриманої моделі. Для побудови 3-х мірного графіка використовуємо команду View->Surface.

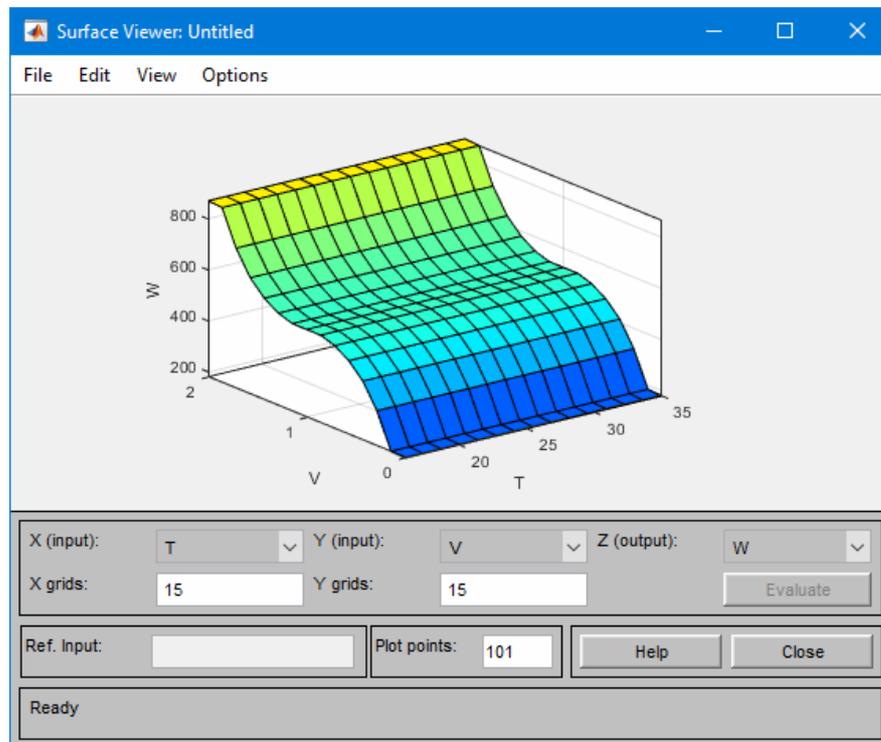


Рис. 20. Графік залежності швидкості обертання вентилятора від температури усередині приміщення і швидкості зміни температури

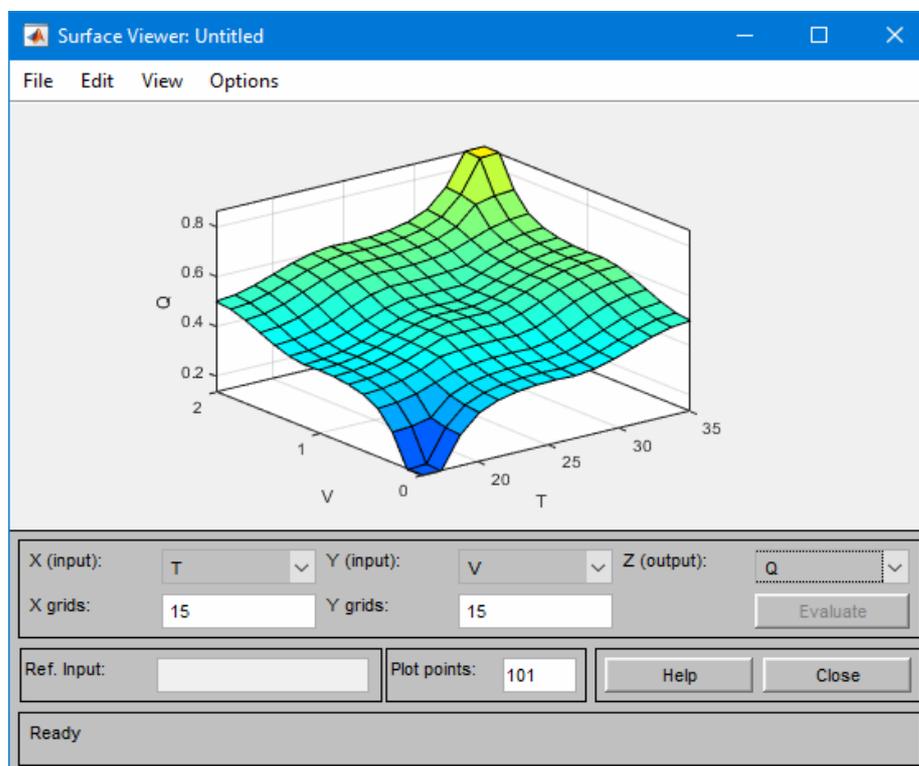


Рис. 21. Графік залежності витрати рідини, що охолоджує, від температури усередині приміщення і швидкості зміни температури

Варіанти завдання 2

Створити систему нечіткого виводу, що імітує модель роботи кондиціонера (варіант див. в таблиці 2).

Таблиця 2

Варіанти завдання

Варіант	1	2	3	4	5	6
mf input	gbellmf	gaussmf	gauss2mf	dsigmf	trimf	trapmf
mf output	trapmf	trimf	dsigmf	gaussmf	gbellmf	gauss2mf
T	11-32	13-30	10-30	14-33	15-34	12-31
V	0-1	1-3	1.5-2	0.5-1.5	0.75-2.5	2-3
W	60-800	40-700	70-900	80-1100	90-1200	100-1300
Q	0-0.7	0-0.8	0-0.9	0-1.1	0-1.2	0-1.3

Варіант	7	8	9	10	11	12
mf input	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf	trimf
mf output	gaussmf	gauss2mf	gbellmf	dsigmf	trimf	trapmf
T	14-32	11-30	12-30	15-33	13-34	10-31
V	0.2-1	1.5-3	1-2	0-1.5	0.25-2.5	0.5-3
W	55-700	45-800	65-900	75-1000	85-1100	95-1200
Q	0.1-0.7	0.2-0.8	0.3-0.9	0.4-1.1	0.5-1.2	0.6-1.3

Варіант	13	14	15	16	17	18
mf input	gaussmf	gauss2mf	gbellmf	dsigmf	trimf	trapmf
mf output	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf	trimf
T	15-34	13-32	10-31	14-33	15-35	12-30
V	0-1.3	1-2.5	1.4-2	0.2-1	0.7-2.	1.8-3
W	65-850	40-750	75-950	80-1150	90-1250	100-1050
Q	0-0.75	0-0.85	0-0.65	0-0.95	0-1.05	0-1.15

Контрольні питання

1. Що таке нечітка логіка?
2. Які застосування використовуються для моделювання нечітких систем?
3. Нечітка множина.
4. Способи завдання нечіткої множини.
5. Міра приналежності.
6. Функція приналежності.
7. Типи функцій приналежності.
8. Лінгвістична змінна, терм.

Лабораторна робота № 2

Тема: СИНТЕЗ СИСТЕМ НА ОСНОВІ НЕЧІТКОЇ ЛОГІКИ

Мета роботи: вивчення основних понять теорії нечіткої множини та її застосувань. Знайомство з пакетом програм MATLAB. Проектування системи нечіткого виводу у середовищі MATLAB

Постановка завдання: апроксимація функцій на основі нечіткої логіки.

Короткі теоретичні відомості

Нечіткі виводи

У експертних та керуючих системах механізм нечітких виводів в своїй основі має базу знань, що формується фахівцями предметної області у вигляді сукупності нечітких предикативних правил виду:

Π_1 : якщо $x \in A_1$, то $y \in B_1$

Π_2 : якщо $x \in A_2$, то $y \in B_2$

.

Π_n : якщо $x \in A_n$, то $y \in B_n$,

де x – вхідна змінна, y – змінна виводу, A і B – функції приналежності, визначені на x і y відповідно.

Знання експерта $A \rightarrow B$ відображає нечітке причинне відношення передумови і висновку, тому його називають нечітким відношенням:

$R = A \rightarrow B$, де « \rightarrow » - нечітка імплікація.

Відношення R можна розглядати як нечітку підмножину прямого добутку $X \times Y$ повної множини передумов X і висновків Y . Таким чином, процес здобуття (нечіткого) результату виводу B' з використанням даного спостереження A' і значення $A \rightarrow B$ можна представити у вигляді:

$B' = A' \bullet R = A' \bullet (A \rightarrow B)$.

Алгоритм нечіткого виводу

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

2. **Логічний вивід**. Обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це наводить до однієї нечіткої підмножини, яка буде призначена змінною виводу для кожного правила. Як правила логічного виводу використовуються лише операції \min (мінімуму) або prod (множення).

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

4. **Дефаззифікація** – приведення до чіткості (defuzzification). Перетворення нечіткого набору виводів в число.

Алгоритми нечіткого виведення Мамдані (Mamdani)

Хай задані два нечітких правила:

P_1 : якщо $x \in A_1$ і $y \in B_1$, тоді $z \in C_1$,

P_2 : якщо $x \in A_2$ і $y \in B_2$, тоді $z \in C_2$.

1. **Нечіткість**. Знаходять міри приналежності для передумов кожного правила: $A_1(x_0)$, $A_2(x_0)$, $B_1(y_0)$, $B_2(y_0)$.

2. **Нечіткий вивід**. Визначають рівні «відсікання» для передумов кожного правила (операція \min):

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

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

Визначають усічені функції приналежності:

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

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

3. **Композиція.** Проводиться об'єднання знайдених усічених функцій (операція max), отримують нечітку підмножину 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. **Дефазифікація.** Приведення до чіткості (визначення z_0), наприклад, центроїдним методом (як центр тяжіння для кривої):

$$z_0 = \frac{\int_{\Omega} z \mu_{\Sigma}(z) dz}{\int_{\Omega} \mu_{\Sigma}(z) dz}.$$

Алгоритм ілюструється на рисунку 1.

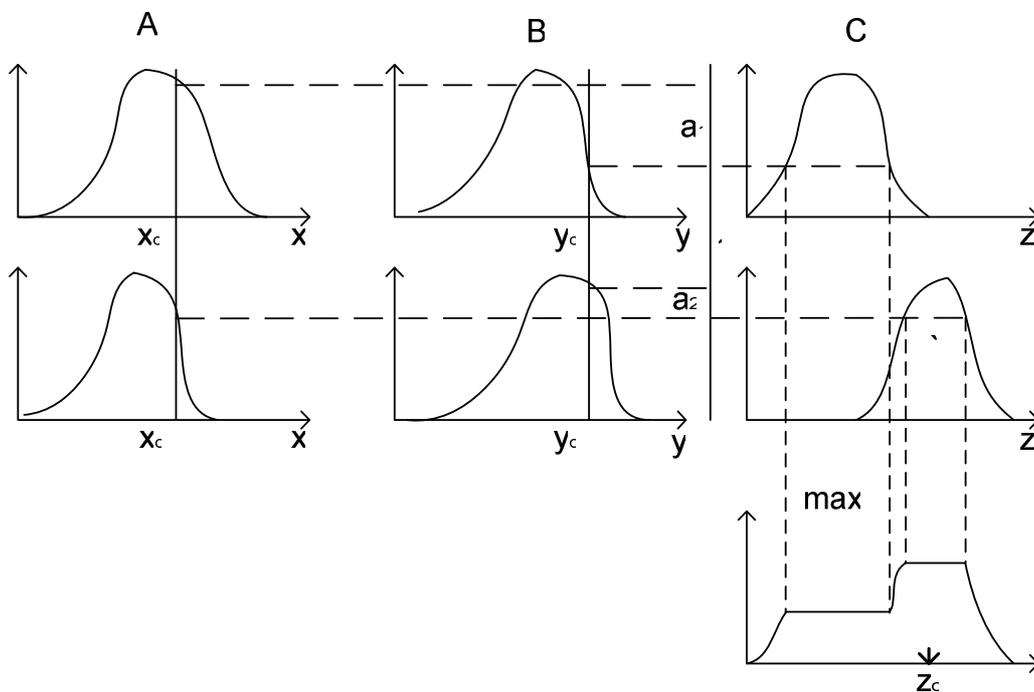


Рис. 1. Графічна інтерпретація алгоритму Мамдані

Алгоритми нечіткого виведення Сугено (Sugeno) і Такагі (Takagi)

Сугено (Sugeno) і Такагі (Takagi) використовували набір правил в наступній формі:

Π_1 : якщо $x \in A_1$ і $y \in B_1$, тоді $z = a_1x + b_1y$,

Π_2 : якщо $x \in A_2$ і $y \in B_2$, тоді $z = a_2x + b_2y$.

1. Перший етап – аналогічно алгоритму Мамдані.

2. Визначаються $\alpha_1 = A_1(x_0) \wedge B_1(y_0)$, $\alpha_2 = A_2(x_0) \wedge B_2(y_0)$ і індивідуальні виходи

правил:

$$z_1^* = a_1 x_0 + b_1 y_0,$$

$$z_2^* = a_2 x_0 + b_2 y_0.$$

3. Визначається значення змінної виводу:

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

Представлена форма правил ілюструє алгоритм Сугено 1-го порядку.

Якщо правила записані у формі:

П₁: якщо $x \in A_1$ і $y \in B_1$, тоді $z = c_1$

П₂: якщо $x \in A_2$ і $y \in B_2$, тоді $z = c_2$,

то заданий алгоритм Сугено 0-го порядку.

Ілюстрація алгоритму Сугено 0-го порядку представлена на рисунку 2.

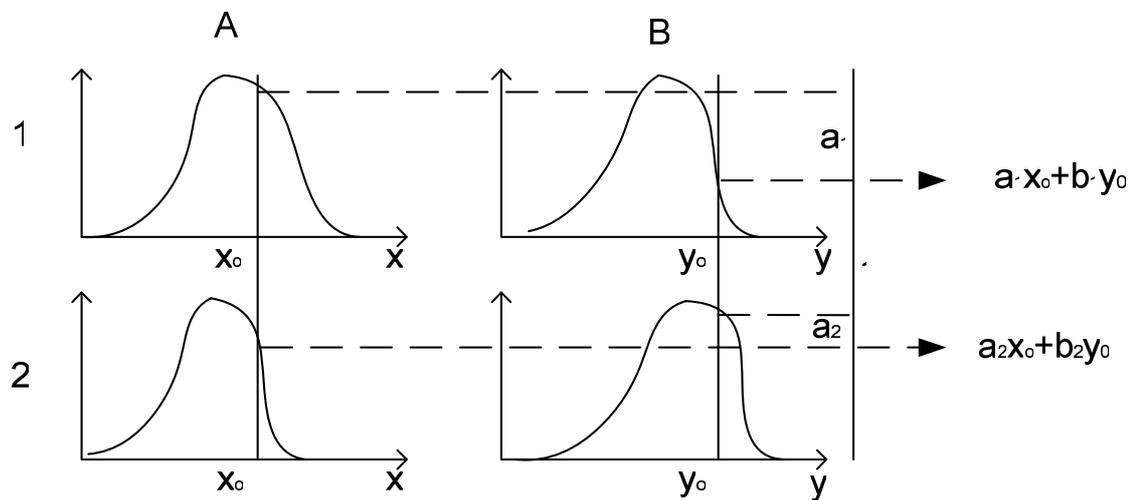


Рис. 2. Графічна інтерпретація алгоритму Сугено

Методи дефазифікації

Метод максимуму – вибирається той елемент нечіткої множини, який має найвищу міру приналежності цій множині.

Якщо цей елемент не є єдиним, тобто функція приналежності $\mu_A(y_1)$ має декілька локальних максимумів, або якщо є максимальне «плато», то вибір серед елементів, що мають найвищу міру приналежності множині, здійснюється на основі деякого критерію.

Метод лівого (правого) максимуму – вибирається найменше (найбільше) з чисел $y_1, y_2 \dots, y_n$, що мають найвищу міру приналежності нечіткій множині.

Метод середнього з максимумів – як шукане чітке значення y_0 приймається середнє арифметичне координат локальних максимумів $y_0 = \frac{1}{n} \sum_{i=1}^n y_i$.

Можливі методи дефазифікації і їх позначення в MATLAB приведені на рисунку 3: найменший максимум (smallest of max, som), найбільший максимум (largest of max, lom), середній максимум (mean of max, mom), бісекторний (bisector of area), центроїдний (centroid).

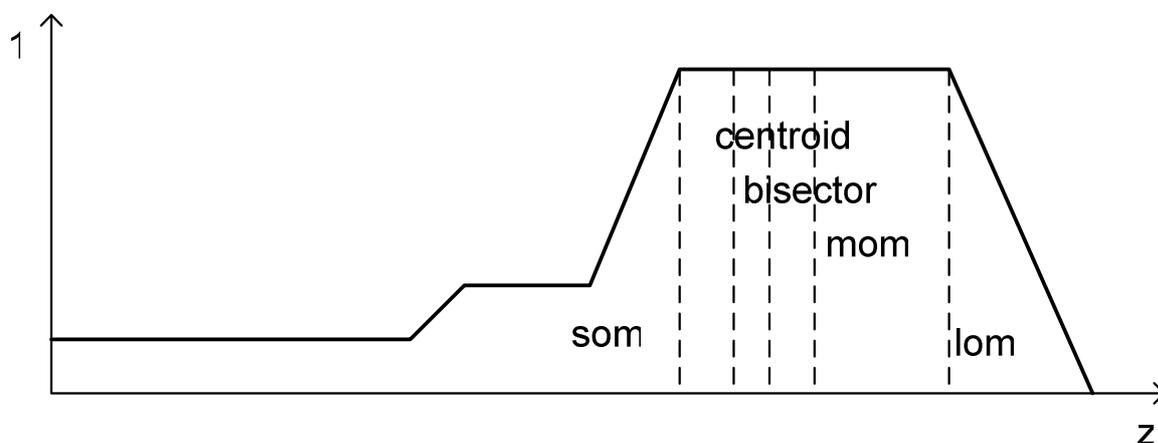


Рис. 3. Ілюстрація прикладів фазифікації z

Завдання. Побудувати нечітку апроксимуючу функцію $y=x^2$.

Вихідні дані для апроксимації представлені в таблиці 1.

Таблиця 1

Значення x і y

X	-1	-0,6	0,1	0,4	1
Y	1	0,36	0,01	0,16	1

Порядок виконання роботи

1. З режиму командного рядка командою `fuzzy` запустити основну інтерфейсну програму пакету Fuzzy Logic – редактор нечіткої системи виводу (рис. 4).

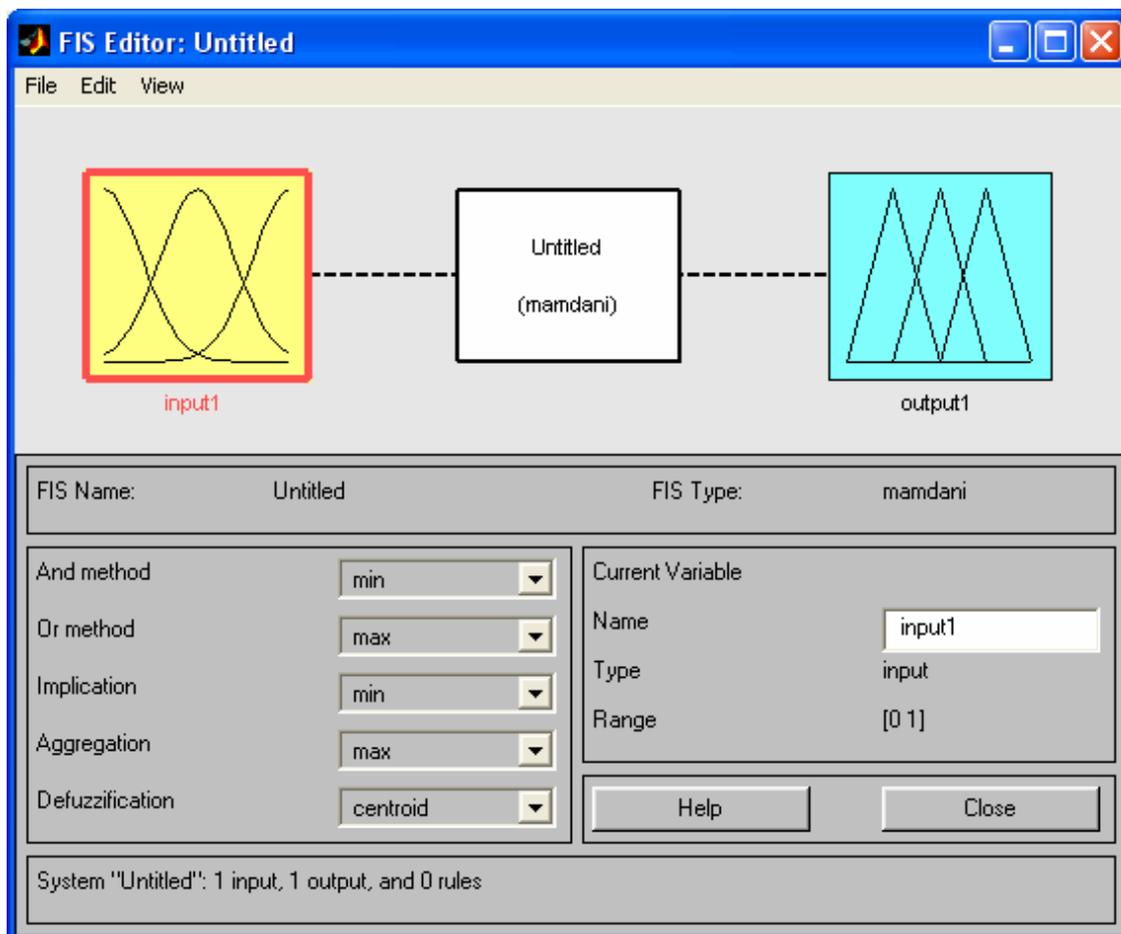


Рис. 4. Вікно редактора нечіткої системи виводу

2. За замовчуванням програма пропонує алгоритм Мамдані, потрібно змінити його на алгоритм Сугено. Для цього, в меню File вибрати команду New FIS... - Sugeno (рис. 5).

3. Зберегти на диску систему за допомогою команд: File->Export->To Disk під ім'ям, наприклад, lr2.

Вибрати вхідний елемент системи `input1` і ввести в поле Name позначення вхідної змінної (`x`), потім виділити вихідний елемент системи `output1` і ввести в поле Name позначення вихідної змінної (`y`) (рис. 6).

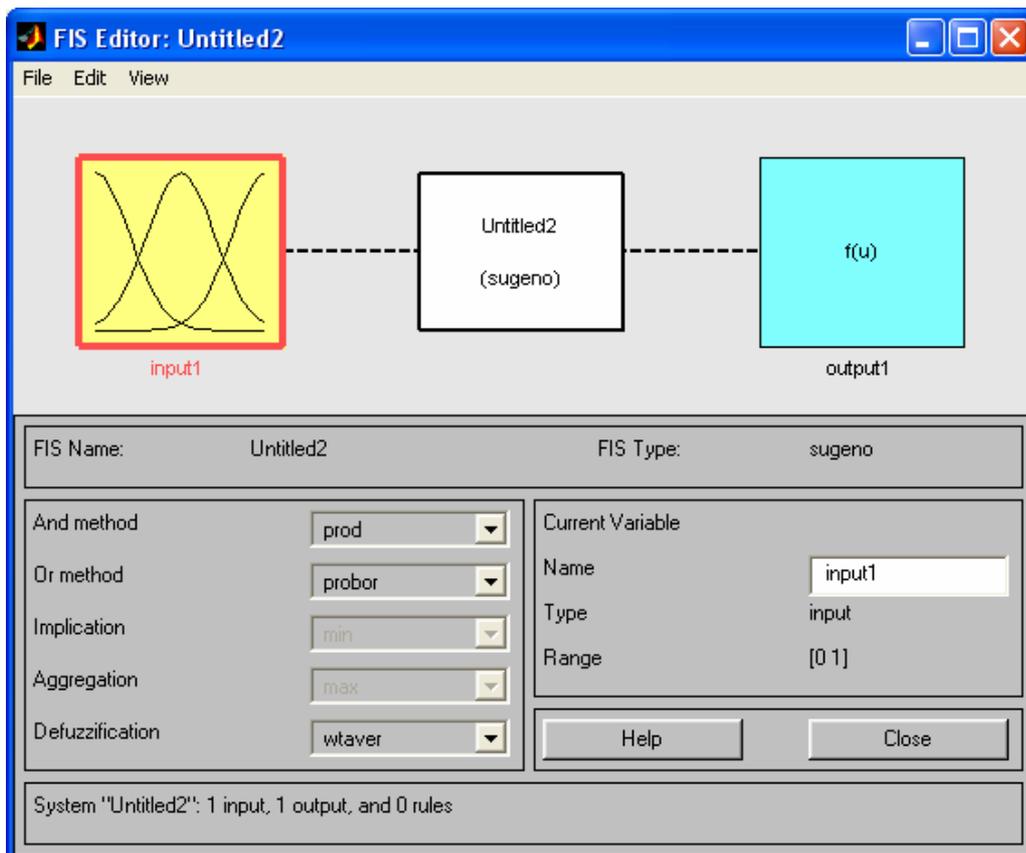


Рис. 5. Вікно алгоритму Сугено

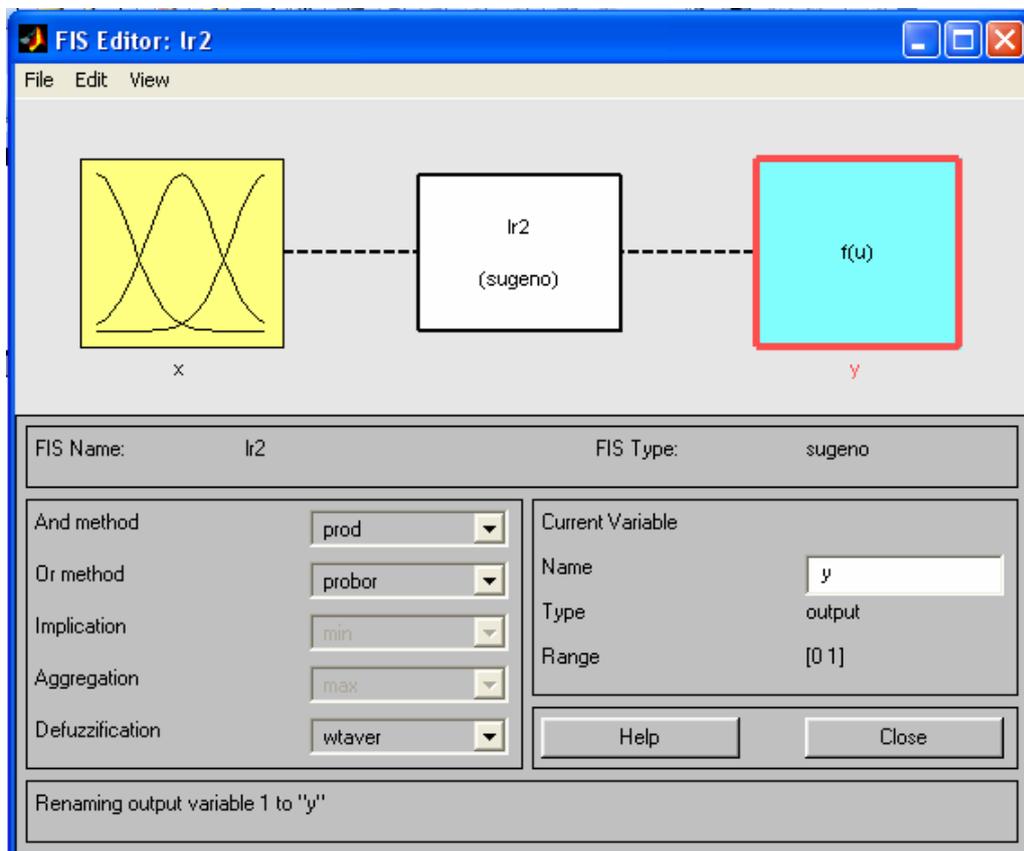


Рис. 6. Перейменування вхідних і вихідних елементів

4. Увійти до режиму редагування функції приналежності вхідного елементу x – Membership Function Editor (подвійним клацанням лівою клавшею миші) (рис. 7).

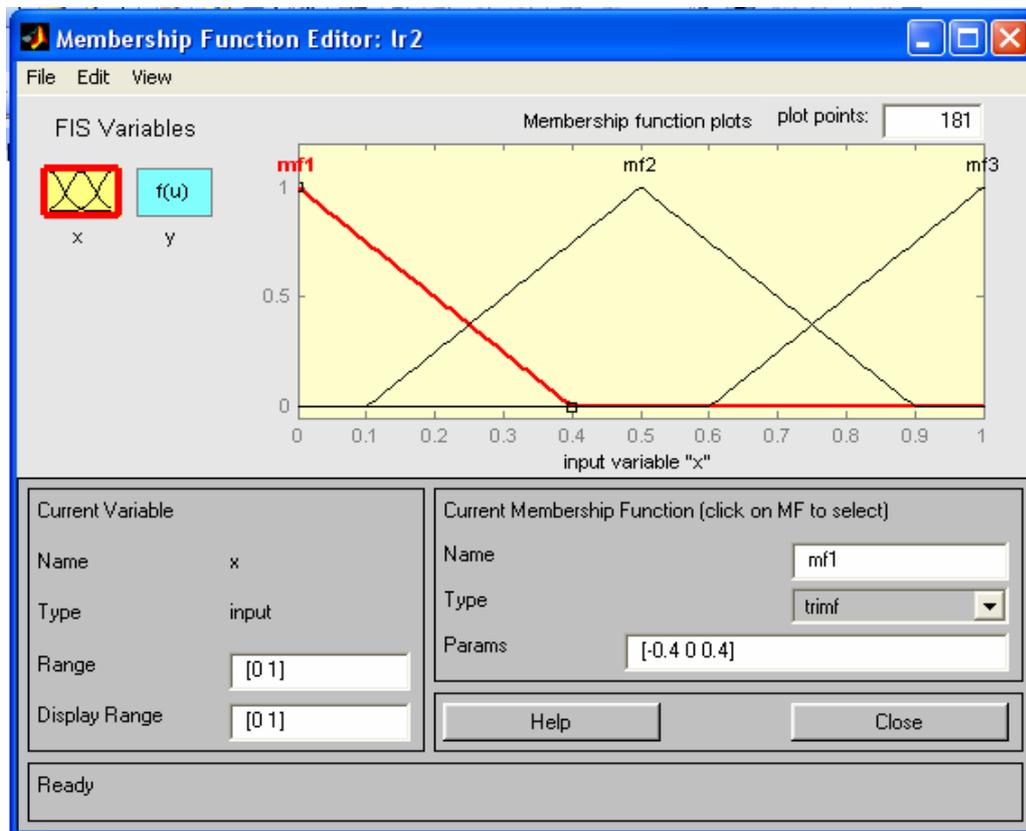


Рис. 7. Режим редагування функції приналежності вхідного елементу

Міняємо заданий за замовчуванням тип функції приналежності. Для цього спочатку необхідно очистити вікно Membership Function plots командами Edit - Remove all MFs, потім командами Edit - Add MFs (додати функцію приналежності) задати тип функції приналежності (gaussmf) і кількість (5) (рис. 8).



Рис. 8. Вікно функції приналежності

В результаті вікно Membership Function Editor для вхідного елемента набере вигляду, показаного на рис. 9.

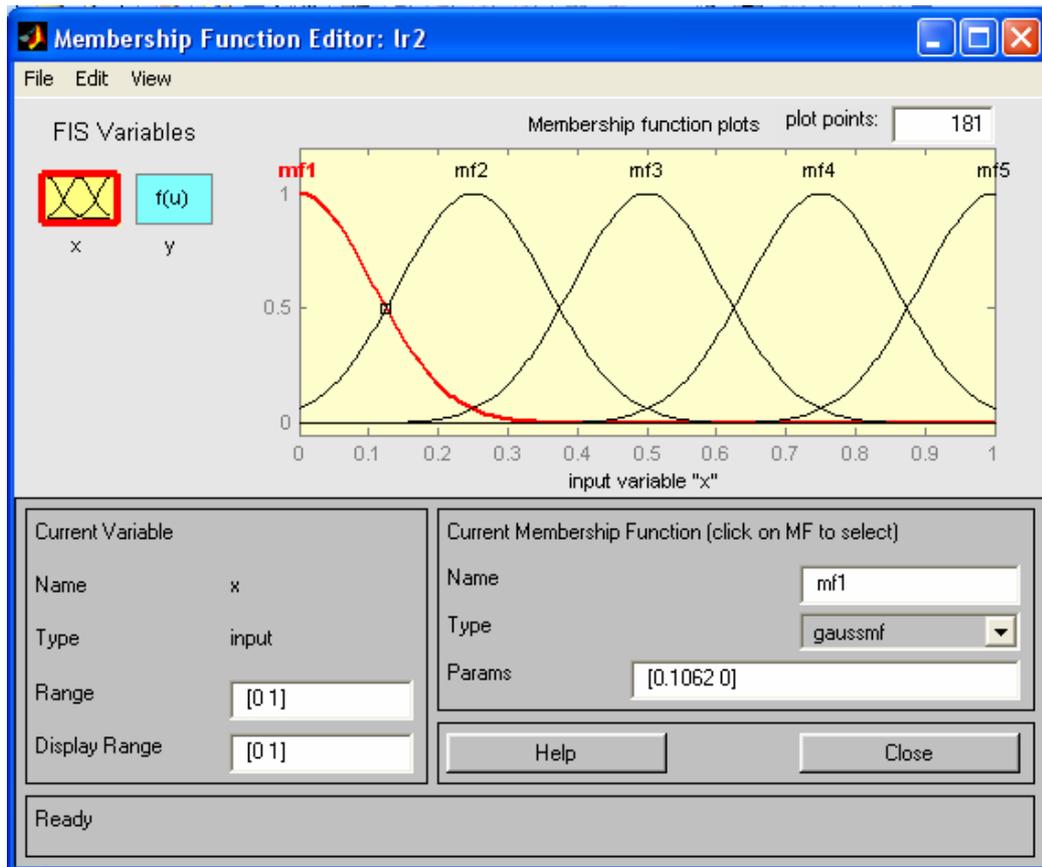


Рис. 9. Вікно редактора вибраної функції приналежності

5. Встановити діапазон (Range) зміни x -1 до 1 (визначається заданим діапазоном x) (рис. 10).

6. Поєднати ординати максимумів функцій приналежності із значеннями аргументу x .

Крок виконується двома способами:

- виділити редагуєму функцію приналежності і перетягнути мишею криву функції приналежності, аби її максимум збігся із значенням на осі x (згідно варіанту завдання);

- точнішу установку можна провести шляхом завдання числових значень параметрів функції приналежності в полі Params (перше значення визначає розмах кривої, друге – положення центру).

У поле Name вводиться ім'я кожної по порядку функції приналежності (mf1- b_n , mf2 – n , mf3 (центральна) – z , mf4 – p , mf5 – b_p) (рис. 10).

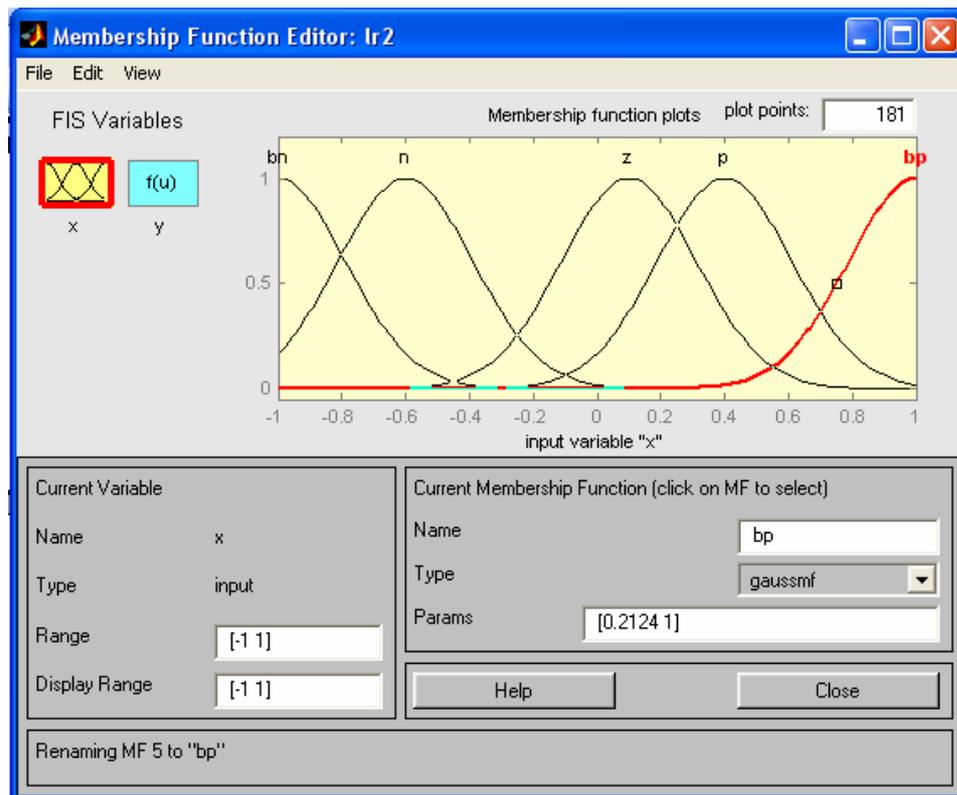


Рис. 10. Результат зміни функцій приналежності

Вийти з редактора функцій приналежності, натиснувши Close.

7. Перейти до вихідного елементу y , виділивши його (рис. 11).

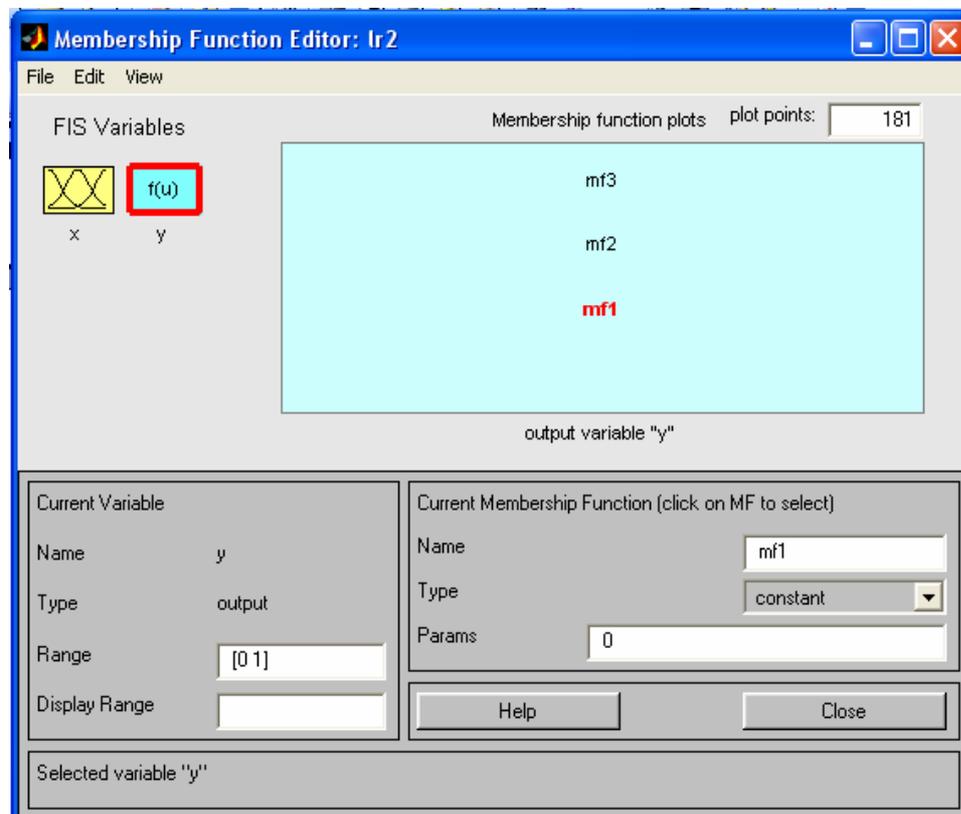


Рис. 11. Вибір вихідний змінної

8. Увійти до режиму редагування функцій приналежності (Edit - Remove all MFs, потім Edit - Add MFs). Задати вигляд функції приналежності для вихідної змінної. Пропонується вибрати в якості функції приналежності лінійні (linear) або постійні (constant) – залежно від алгоритму Сугено (1-го або 0-го порядку). У поставленому завданні необхідно вибрати постійні функції приналежності із загальним числом 4 (по числу різних значень y). Натиснути ОК (Рис. 12).

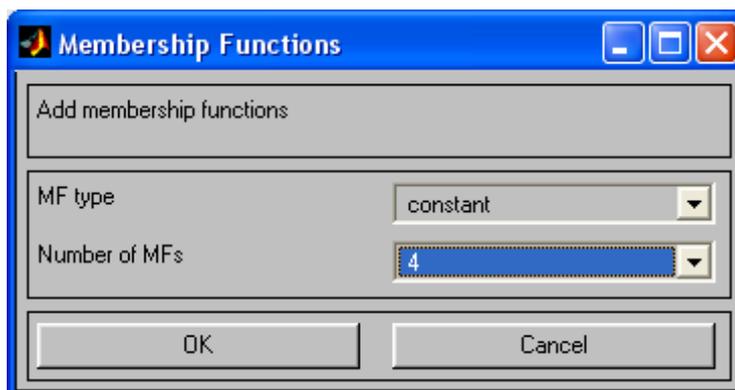


Рис. 12. Зміна типа функцій приналежності

В результаті вікно Membership Function Editor для вихідний змінної набере вигляду, показано на рис. 13.

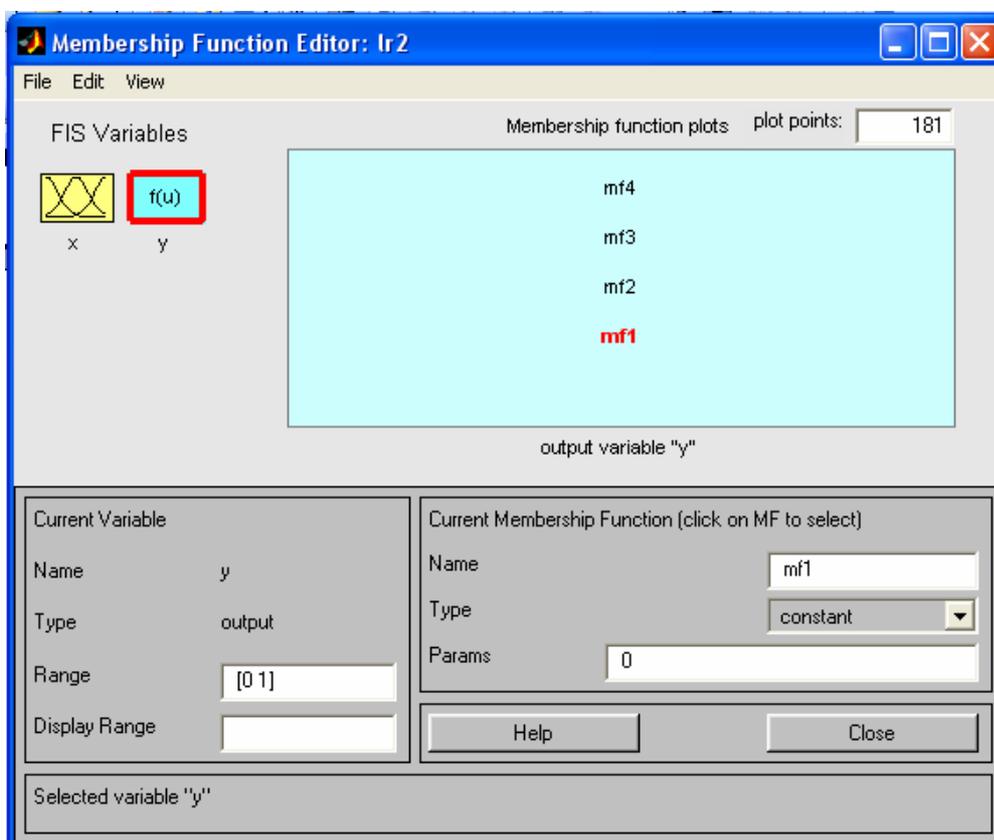


Рис. 13. Функції приналежності вихідної змінної

9. Діапазон (Range) не змінюємо – $[0, 1]$. Змінити імена (Name) функцій приналежності відповідно: 0.01; 0.16; 0.36; 1; одночасно ці ж числові значення вводимо в поле Params (рис. 14). Закрити вікно редактора, натисненням кнопки Close і повернутися у вікно FIS-редактора.

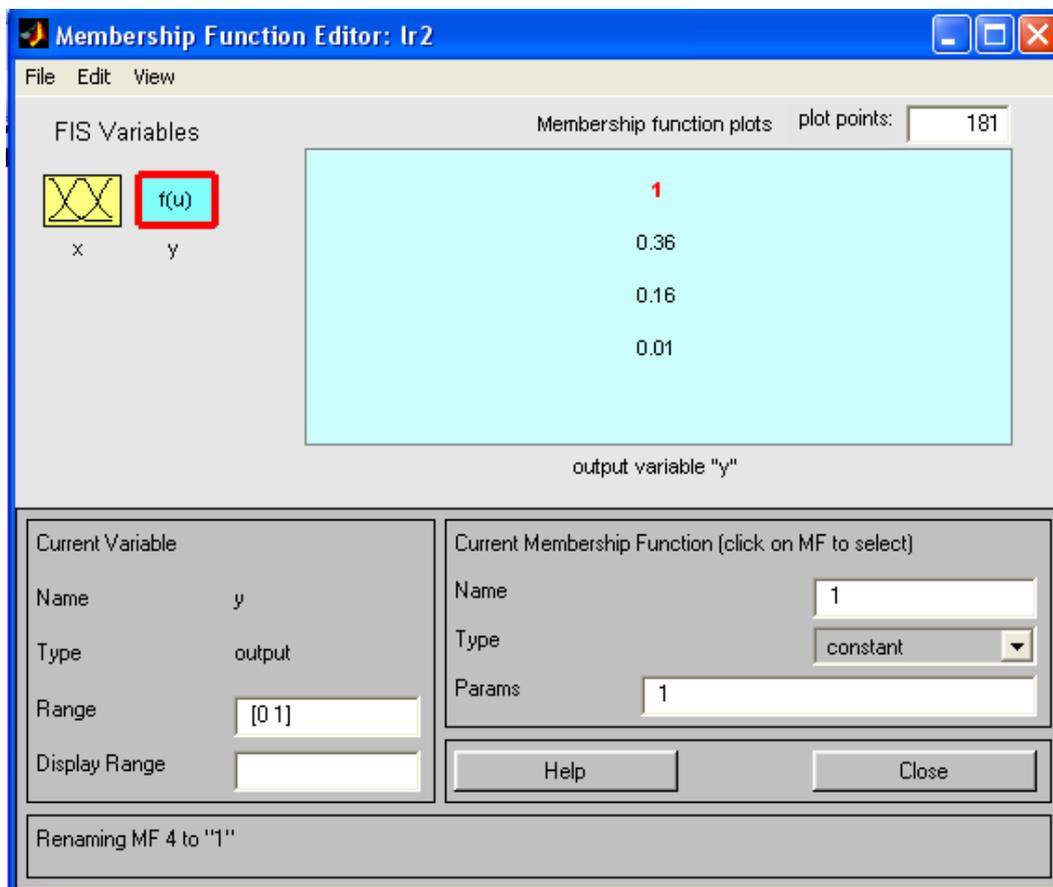


Рис. 14. Зміна функцій приналежності вихідній змінній

10. Перейти в редактор правил (Rule Editor) (двічі клацнути на середній білий квадрат структури системи нечіткого виводу, що розробляється, з ім'ям lr2). При введенні кожного правила необхідно позначити відповідність між кожною функцією приналежності аргументу x і числовим значенням. Крива, позначена b_1 , відповідає $y=1$, для чого вибирається в лівому полі (із заголовком x is) варіант b_1 , а в правому – 1 і натискується кнопка Add rule внизу вікна редактора.

Введене правило з'явиться у вікні правил Rule Editor у вигляді, показаному на рисунку 15.

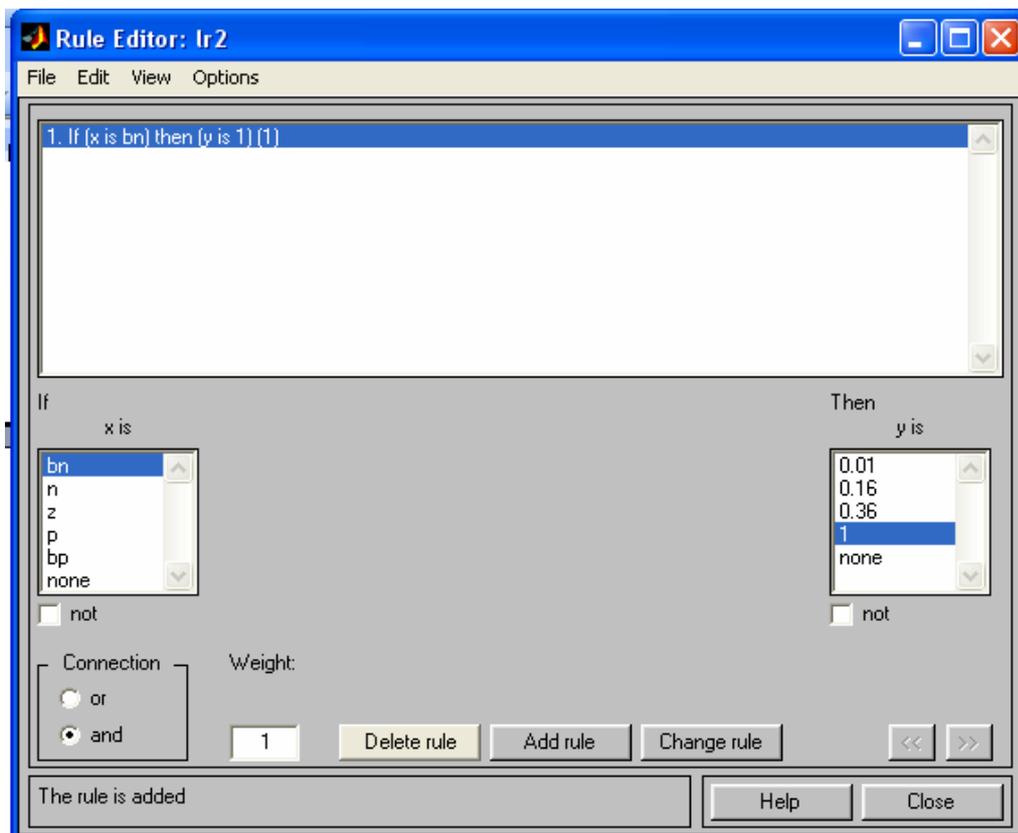


Рис. 15. Введення першого правила

Аналогічно вводяться всі п'ять правил даної системи (рис. 16).

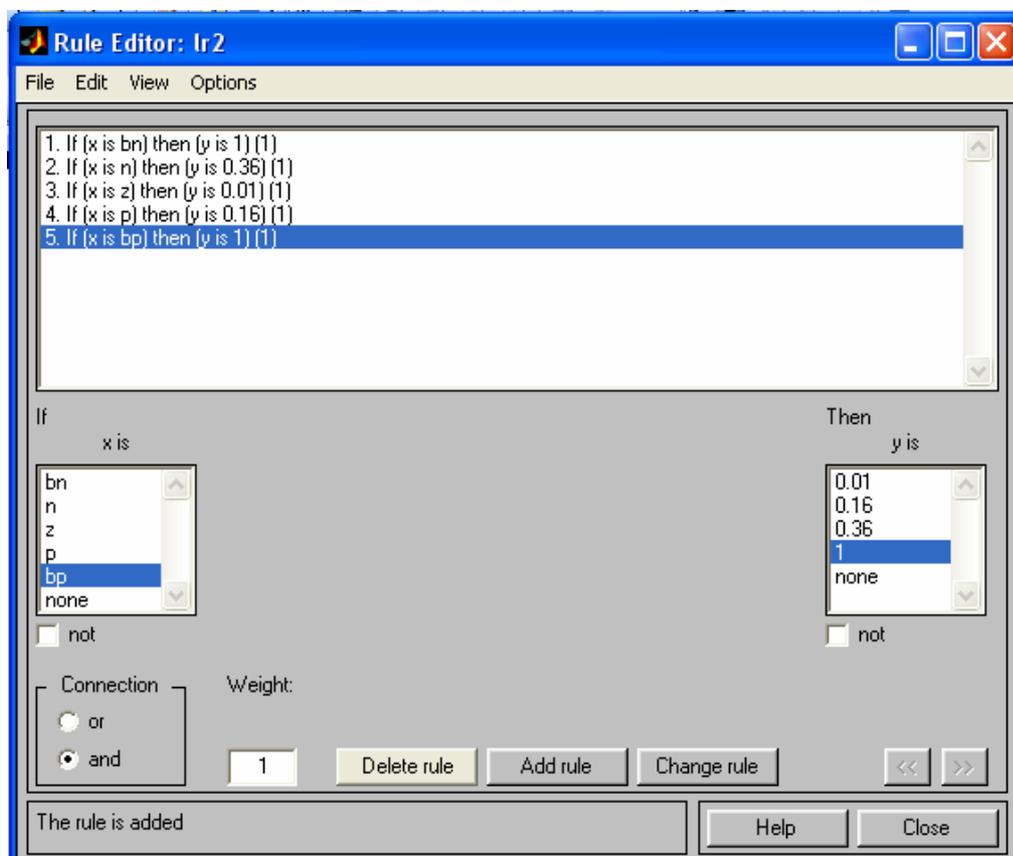


Рис. 16. Вікно редактора правил після введення всіх правил

Закрити вікно редактора правив і повернутися у вікно FIS-редактора.

Побудова системи закінчена. Необхідно зберегти систему.

11. Перейти в редактор функцій приналежності (Edit - Membership function.).

З вікна редактора командою можна перейти у вікно перегляду правил (View - Rules) та перегляду поверхні (View - surface).

У вікні перегляду правил (рис.17) ілюструється процес ухвалення рішення (обчислення y).

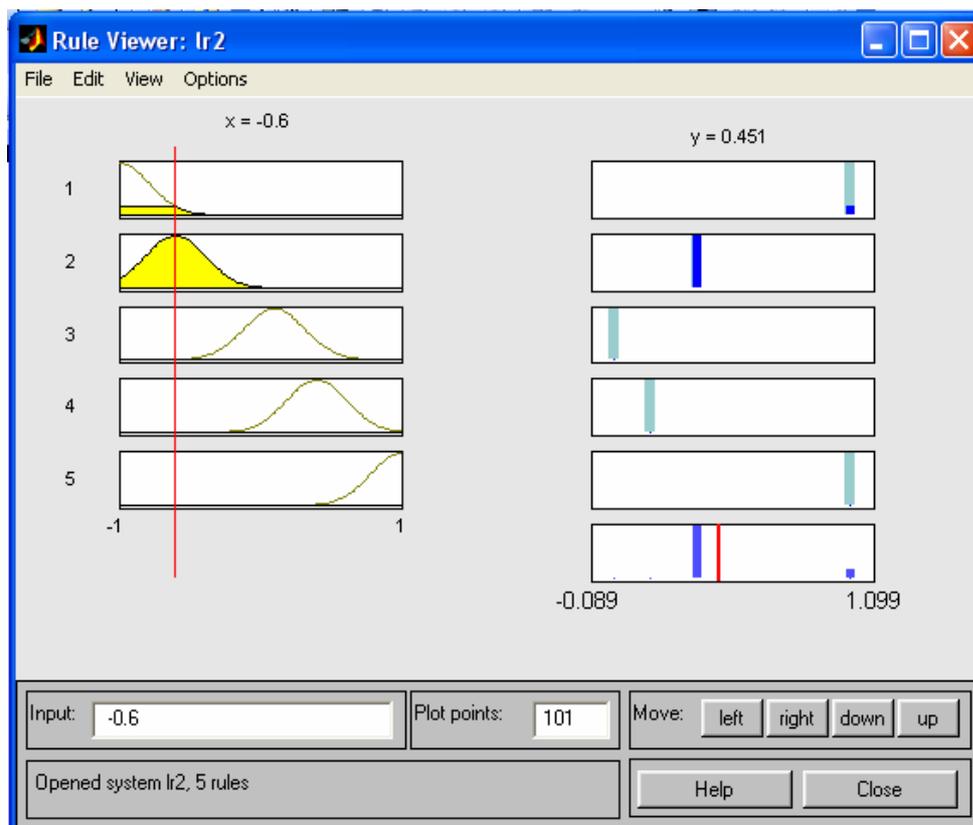


Рис. 17. Вікно перегляду правил

Червона вертикальна риса, що пересікає графіки в лівій частині вікна (її можна переміщати за допомогою миші), дозволяє змінювати значення змінної входу (або можна ввести значення змінної з клавіатури в полі Input), при цьому відповідно змінюється значення виходу y .

Знаходимо значення y при всіх значеннях вхідної змінної x . Результати апроксимації функції заносимо в таблицю.

Результати апроксимації функції $y=x^2$

x	-1	-0,6	0,1	0,4	1
y	1	0,36	0,01	0,16	1
y_{apr}	0,908	0,451	0,0514	0,131	0,985
σ	0,092	0,253	4,14	0,181	0,015

Похибка апроксимації розраховується за формулою: $\sigma=|y-y_{apr}|/y$.

Велика похибка апроксимації заданої залежності пояснюється малим числом експериментів.

Перегляд кривої $y(x)$ здійснюється командою View/View surface (рис. 18).

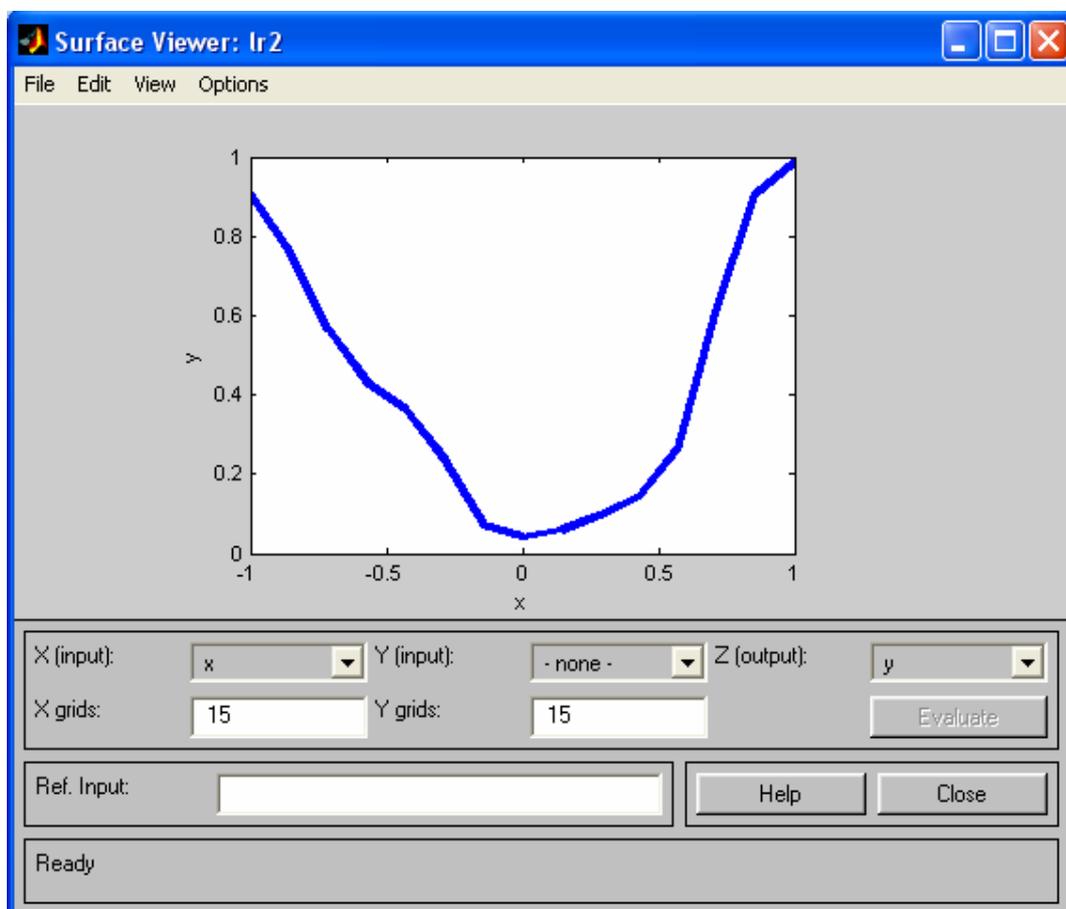


Рис. 18. Вікно перегляду поверхні

За допомогою розглянутих редакторів на будь-якому етапі проектування нечіткої моделі можна внести необхідні корективи, наприклад завдання, призначеної для користувача, функції приналежності.

У розглянутому прикладі використані наступні операції, що встановлюються за замовчуванням в алгоритмі Сугено (рис. 19):

- логічний вивід організується за допомогою операції логічного множення (prod);
- композиція організується за допомогою операції логічної суми (імовірнісного АБО, probor);
- приведення до чіткості (дефазифікація) організується дискретним варіантом центроїдного методу (зваженим середнім, wtaver).

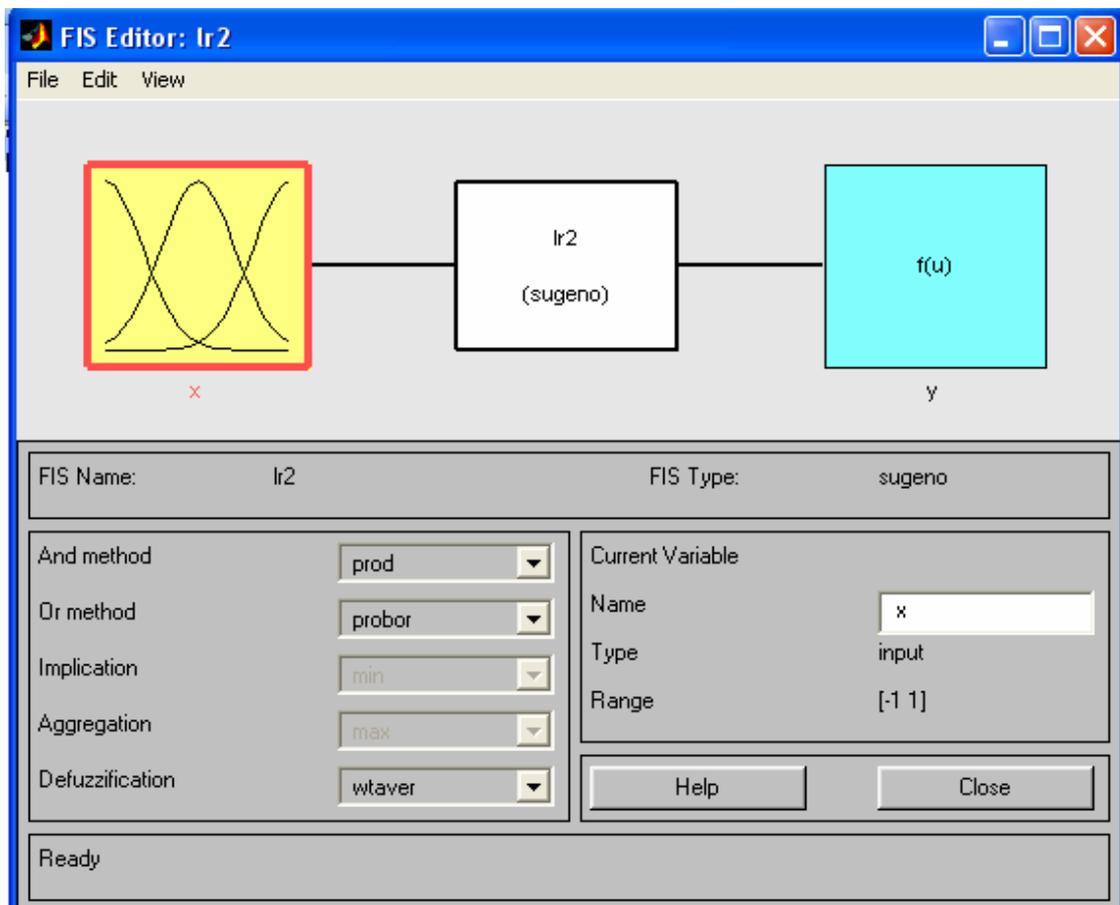


Рис. 19. Операції алгоритму Сугено

Варіанти завдання

Створити систему нечіткого виводу для апроксимації функції $y=k*x^2$ (варіант див. в таблиці 2). Для апроксимації взяти 5 значень функції.

Таблиця 2

Варіанти завдання для апроксимації функції

Варіант	1	2	3	4	5	6
k	1.5	1.8	1.3	1.9	2	2.2
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Варіант	7	8	9	10	11	12
k	1.4	2.5	2.3	2.4	3.5	3.8
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Варіант	13	14	15	16	17	18
k	3.3	3.9	3	3.2	3.4	4
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Контрольні питання

1. Фазифікація і дефазифікація.
2. Методи дефазифікації.
3. Нечітка база знань.
4. Нечіткий логічний вивід.
5. Система нечіткого виводу.
6. Алгоритми нечіткого виводу Мамдані і Сугено.
7. Запуск пакету Fuzzy Logic.
8. Як створити систему нечіткого виводу?
9. Як вибрати функцію приналежності?
10. Як задати правила нечіткого виводу?
11. Для чого призначені вікна перегляду правил і перегляду поверхні, як їх відкрити?

Лабораторна робота № 3

Тема: РОЗРОБКА ГІБРИДНИХ ІНТЕЛЕКТУАЛЬНИХ СИСТЕМ В СЕРЕДОВИЩІ MATLAB

Мета роботи: вивчення основних властивостей і основ роботи з гібридними інтелектуальними системами в програмному середовищі MatLab 6.5.

Постановка завдання: апроксимація функцій за допомогою адаптивної мережі нечіткого виводу.

Короткі теоретичні відомості

ANFIS-редактор

ANFIS - є аббревіатурою Adaptive Neuro-Fuzzy Inference System (адаптивна нейро-нечітка система). ANFIS-редактор дозволяє автоматично синтезувати з експериментальних даних нейро-нечіткої мережі. Нейро-нечітку мережу можна розглядати як один з різновидів систем нечіткого логічного виведення 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`. В результаті виконання цієї команди з'являється графічне вікно, показане на рис. 1.

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

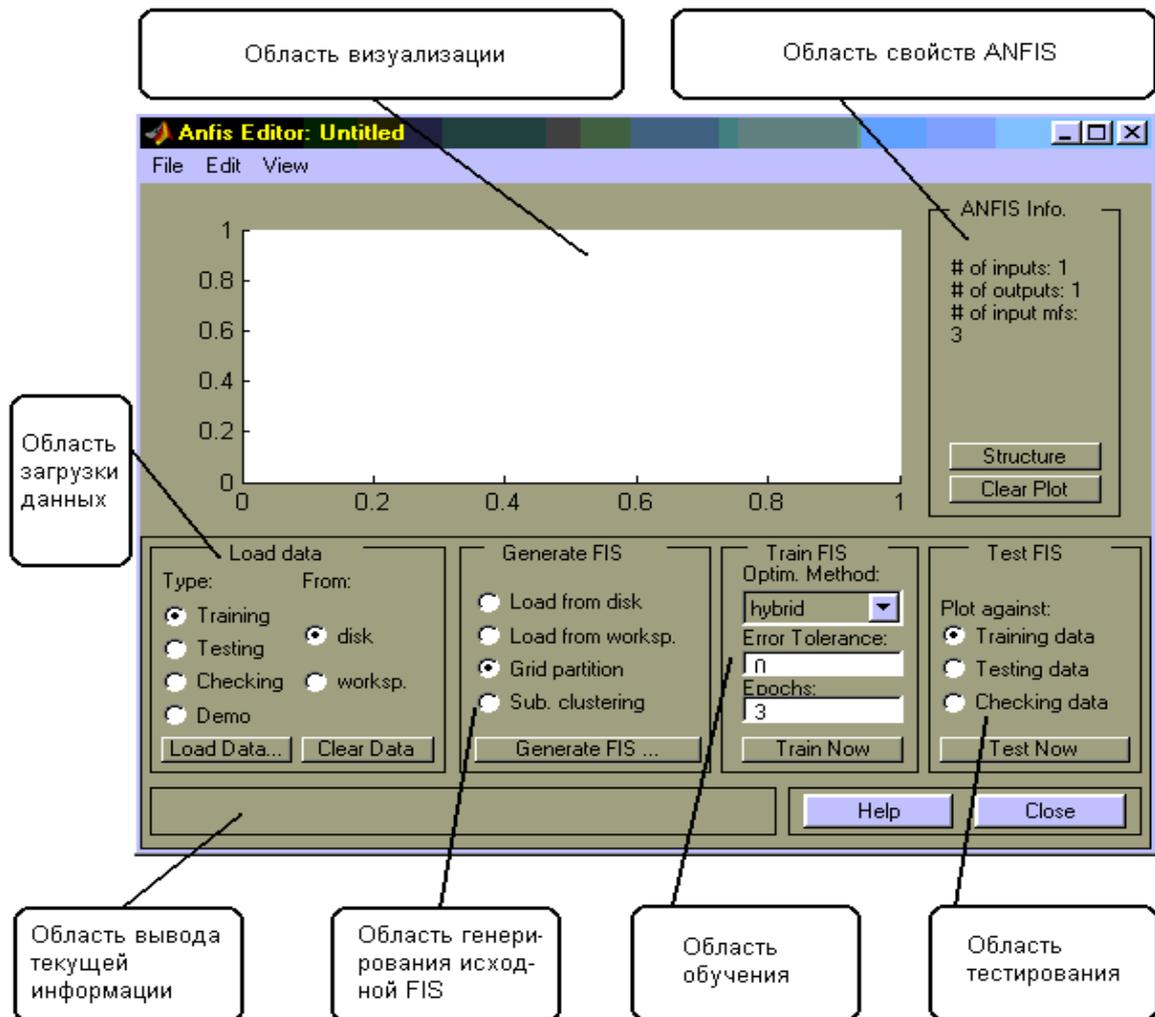


Рис. 1. Основне вікно ANFIS-редактора

Меню File і View однакові для всіх GUI-модулей, використовуваних з системами нечіткого логічного виводу.

Меню Edit

Загальний вигляд меню приведений на рис.2.

Undo	Ctrl+Z
FIS Properties...	Ctrl+1
Membership Functions...	Ctrl+2
Rules...	Ctrl+3
Anfis...	Ctrl+4

Рис. 2. Меню Edit

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

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

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

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

Команда Anfis відкриває ANFIS-редактор. Ця команда може бути також виконана натисненням клавіш Ctrl+3. Відмітимо, що дана команда, запущена з ANFIS-редактора, не приводить до виконання яких-небудь дій, оскільки цей редактор вже відкритий. Проте, в меню Edit інших GUI-модулей, використовуваних в системах нечіткого логічного виводу, додається команда Anfis., що дозволяє відкрити ANFIS-редактор з цих модулів.

Область візуалізації

У цій області виводиться два типи інформації:

- При навчанні системи – крива навчання у вигляді графіка залежності помилки навчання від порядкового номера ітерації.
- При завантаженні даних і тестуванні системи – експериментальні дані і результати моделювання.

Експериментальні дані і результати моделювання виводяться у вигляді множини крапок у двовимірному просторі. При цьому по осі абсцис відкладається порядковий номер рядка даних у вибірці (навчальній, тестуючій або контрольній), а по осі ординат - значення вихідної змінної для даного рядка вибірки.

Використовуються наступні маркери:

- блакитна крапка (.) – тестуюча вибірка;
- блакитне коло (o) – навчальна вибірка;
- блакитний плюс (+) – контрольна вибірка;
- червона зірочка (*) – результати моделювання.

Область властивостей ANFIS

В області властивостей ANFIS (ANFIS info) виводиться інформація про кількість вхідних і вихідних змінних, про кількість функцій приналежностей для кожної вхідної змінної, а також про кількість рядків у вибірках. У цій області розташовано дві кнопки: Structure і Clear Plot.

Натиснення кнопки Structure відкриває нове графічне вікно, в якому система нечіткого логічного виводу представляється у вигляді нейро-нечіткої мережі. На рис. 3 приведена нейро-нечітка мережа, що містить чотири вхідних змінних і одну вихідну. У цій системі по три лінгвістичні терми використовуються для оцінки кожної з вхідних змінних і чотири терми для вихідної.

Натиснення кнопки Clear Plot дозволяє очистити область візуалізації.

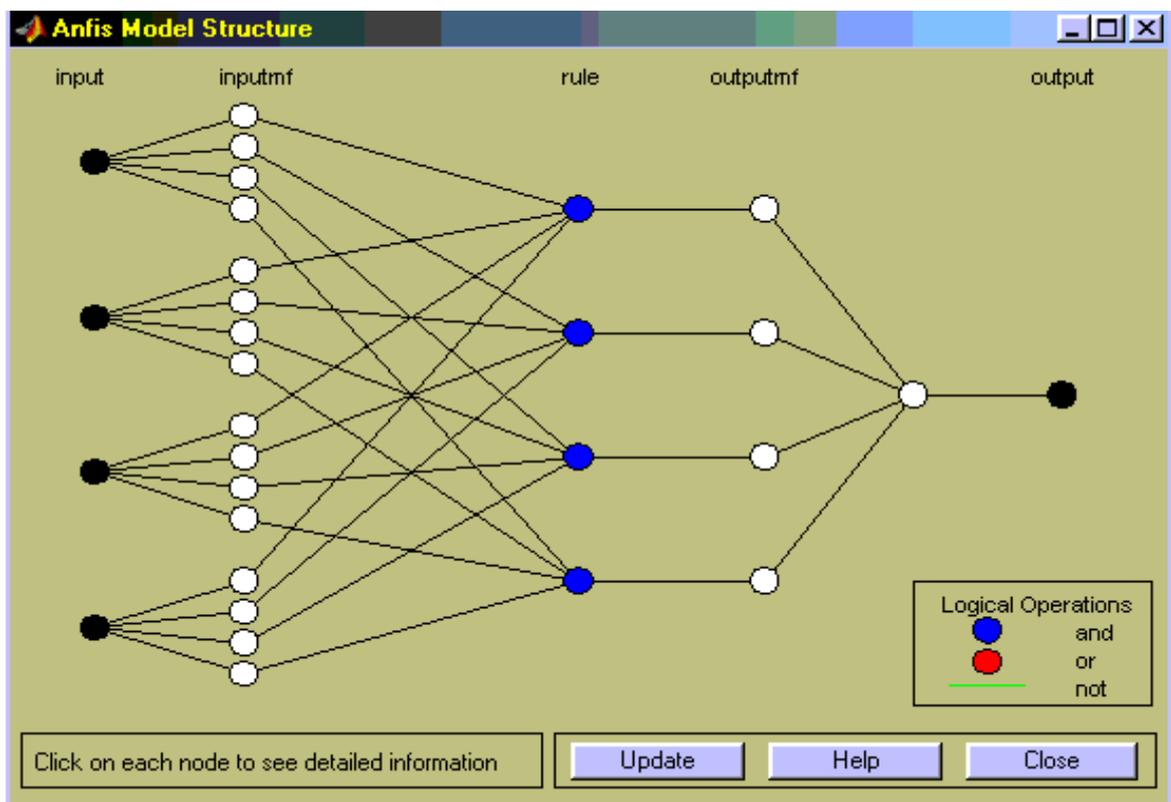


Рис. 3. Приклад структури нейро-нечіткої мережі

Область завантаження даних

В області завантаження даних (Load data) розташовані:

- меню вибору типу даних (Type), що містить наступні альтернативи:
 - **Traning** - навчальна вибірка;

- **Testing** - тестуюча вибірка;
- **Checking** - контрольна вибірка;
- **Demo** - демонстраційний приклад;
- меню вибору джерела даних (From), що містить наступні альтернативи:
 - **disk** – диск;
 - **worksp.** – робоча область MatLab;
- 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;
 - кнопка очищення даних Clear Data.

Примітка. Протягом одного сеансу роботи ANFIS-редактора можна завантажувати дані одного формату, тобто кількість вхідних змінних у вибірках має бути однаковою.

Область генерування вихідної системи нечіткого логічного виводу

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 решітки (без кластеризації);
- **Sub. clustering** – генерування системи по методу субкластеризації.

У цій області також розташована кнопка Generate, по натисненню якої генерується вихідна система нечіткого логічного виводу.

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 виводу.

При виборі Grid partition з'являється вікно введення параметрів методу решітки (рис. 4), в якому потрібно вказати кількість термів для кожної вхідної змінної і типи функцій приналежності для вхідних і вихідних змінних.

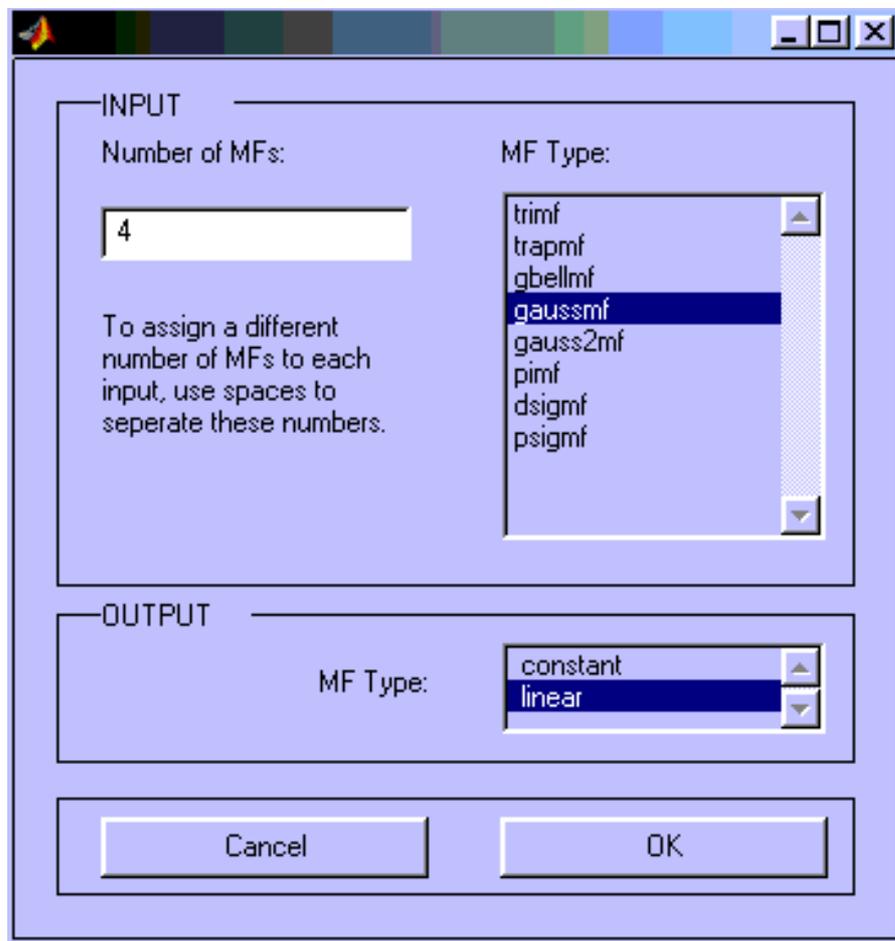


Рис. 4. Вікно введення параметрів для методу решітки

При виборі Sub. clustering з'являється вікно введення наступних параметрів методу субкластеризації (рис. 5):

Range of influence – рівні впливу вхідних змінних;

Squash factor – коефіцієнт пригнічення;

Accept ratio – коефіцієнт, що встановлює у скільки разів потенціал даної крапки має бути вище за потенціал центру першого кластера для того, щоб центром одного з кластерів була призначена дана крапка;

Reject ratio – коефіцієнт, потенціал даної крапки, що встановлює у скільки разів, має бути нижче за потенціал центру першого кластера, аби дана крапка була виключена з можливих центрів кластерів.

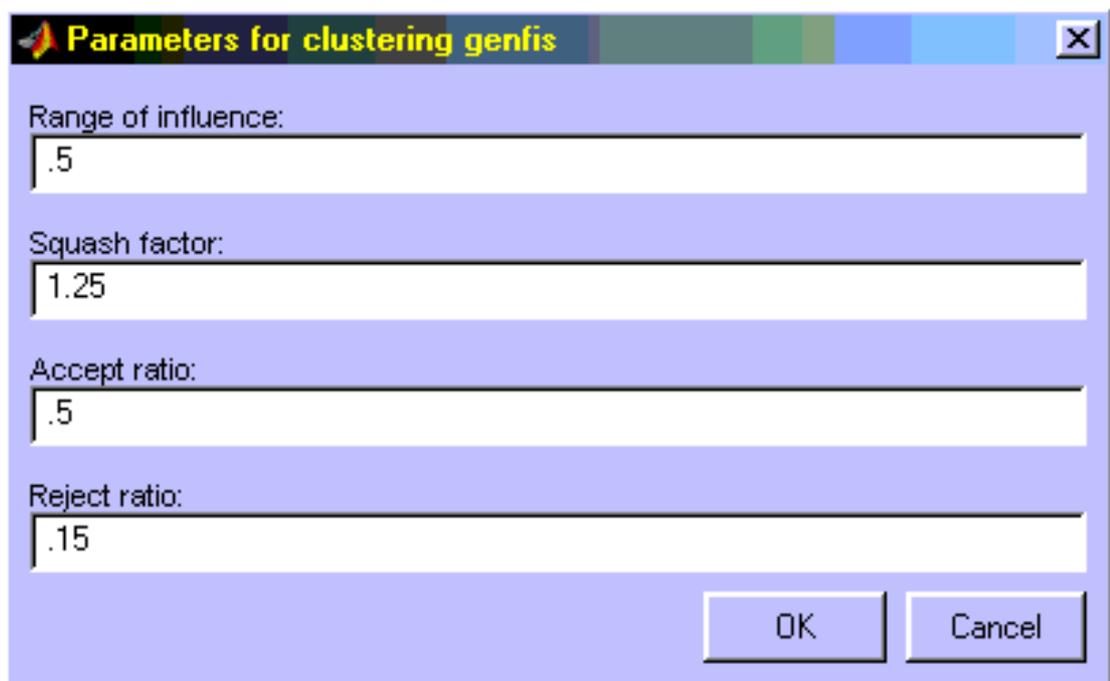


Рис. 5. Вікно введення параметрів для методу субкластеризації

Область навчання

В області навчання (Train FIS) розташовані меню вибору методу оптимізації (Optim. method), поле завдання необхідній точності навчання (Error tolerance), поле завдання кількості ітерацій навчання (Epochs) і кнопка Train Now, натиснення якої запускає режим навчання. Проміжні результати навчання виводяться в область візуалізації і в робочу область MatLab. У ANFIS-редакторі реалізовано два методи навчання:

- **backpropa** – метод зворотного поширення помилки, заснований на ідеях методу найшвидшого спуску;
- **hybrid** – гібридний метод, об'єднуючий метод зворотного поширення помилки з методом найменших квадратів.

Область тестування

В області тестування (Test FIS) розташовано меню вибору вибірки і кнопку Test Now, по натисненню якої відбувається тестування нечіткої системи з виведенням результатів в область візуалізації.

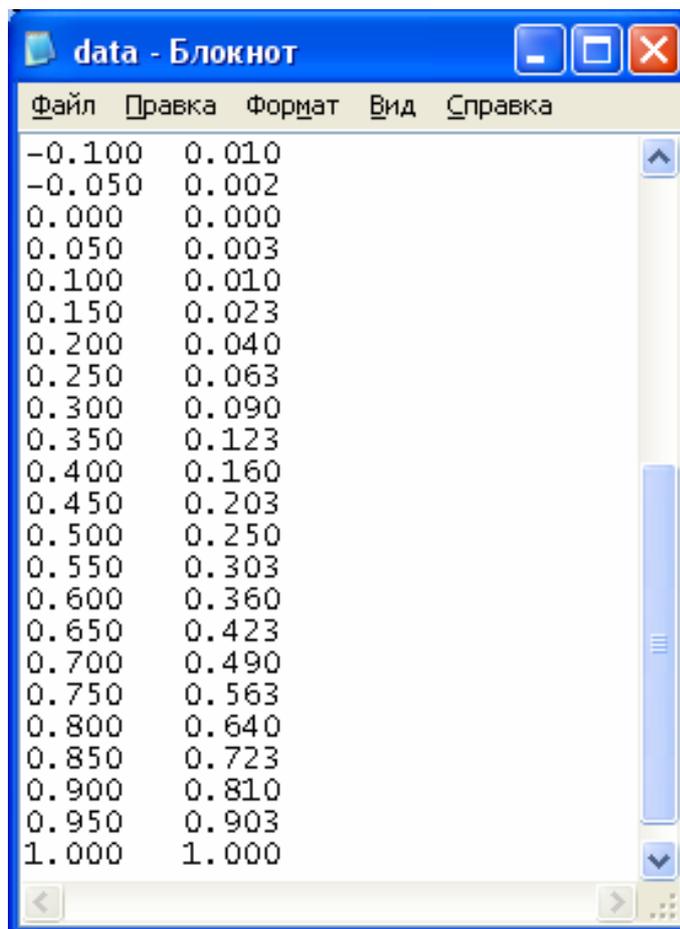
Область виведення поточної інформації

У цій області виводиться найбільш істотна поточна інформація, наприклад, повідомлення про закінчення виконання операцій, значення помилки навчання або тестування і тому подібне.

Завдання. За допомогою адаптивної мережі нечіткого виводу апроксимувати функцію: $y=x^2$, використовуючи функцію приналежності gaussmf.

Порядок виконання роботи

1. Створити за допомогою Excel і зберегти у вигляді текстового файлу вихідні дані для навчання нейро-нечіткої мережі (рис. 6). Перший стовпчик – значення аргументу x в діапазоні від -1 до 1 з кроком 0.05, другий – відповідні значення функції y згідно своєму варіанту (табл. 5).



Файл	Правка	Формат	Вид	Справка
-0.100	0.010			
-0.050	0.002			
0.000	0.000			
0.050	0.003			
0.100	0.010			
0.150	0.023			
0.200	0.040			
0.250	0.063			
0.300	0.090			
0.350	0.123			
0.400	0.160			
0.450	0.203			
0.500	0.250			
0.550	0.303			
0.600	0.360			
0.650	0.423			
0.700	0.490			
0.750	0.563			
0.800	0.640			
0.850	0.723			
0.900	0.810			
0.950	0.903			
1.000	1.000			

Рис. 6. Вихідні дані для навчання нейро-нечіткої мережі

2. У Matlab у вікні Current Directory натиснути кнопку (Browse for Folder) і вказати дорогу до папки з вихідними даними, в якій також буде збережена нейро-нечітка система.

3. Командою `anfisedit` з режиму командного рядка Matlab запусити ANFIS-редактор адаптивної нейро-нечіткої системи (рис. 7).

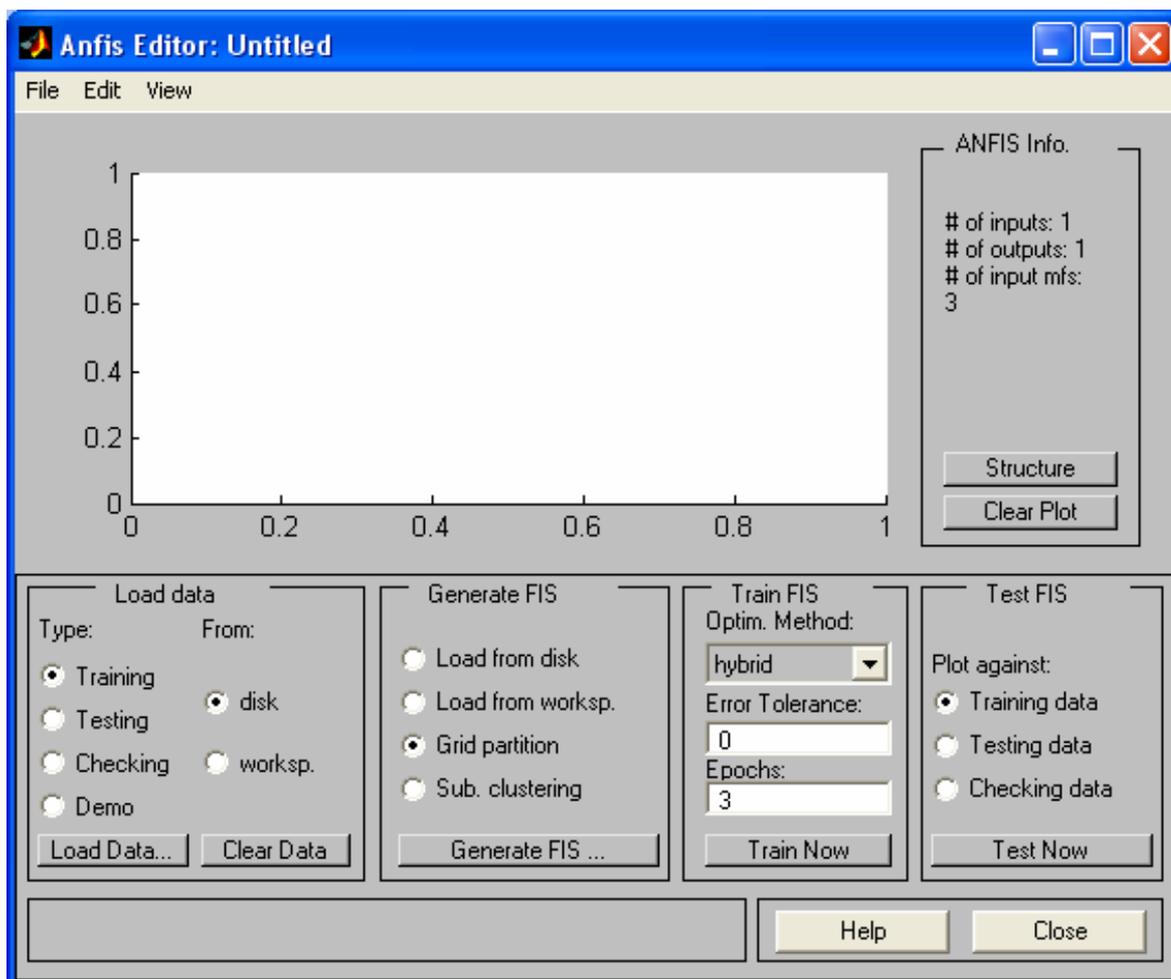


Рис. 7. Вихідне вікно ANFIS-редактора

4. Завантажити навчальну вибірку. Для цього в полі Load data при вибраних за замовчуванням опціях Training і disk натиснути кнопку Load data.. У вікні Select File of Open, що з'явилося, вказати тип файлів: All Files (*.*) і відкрити файл з вихідною вибіркою (рис. 8).

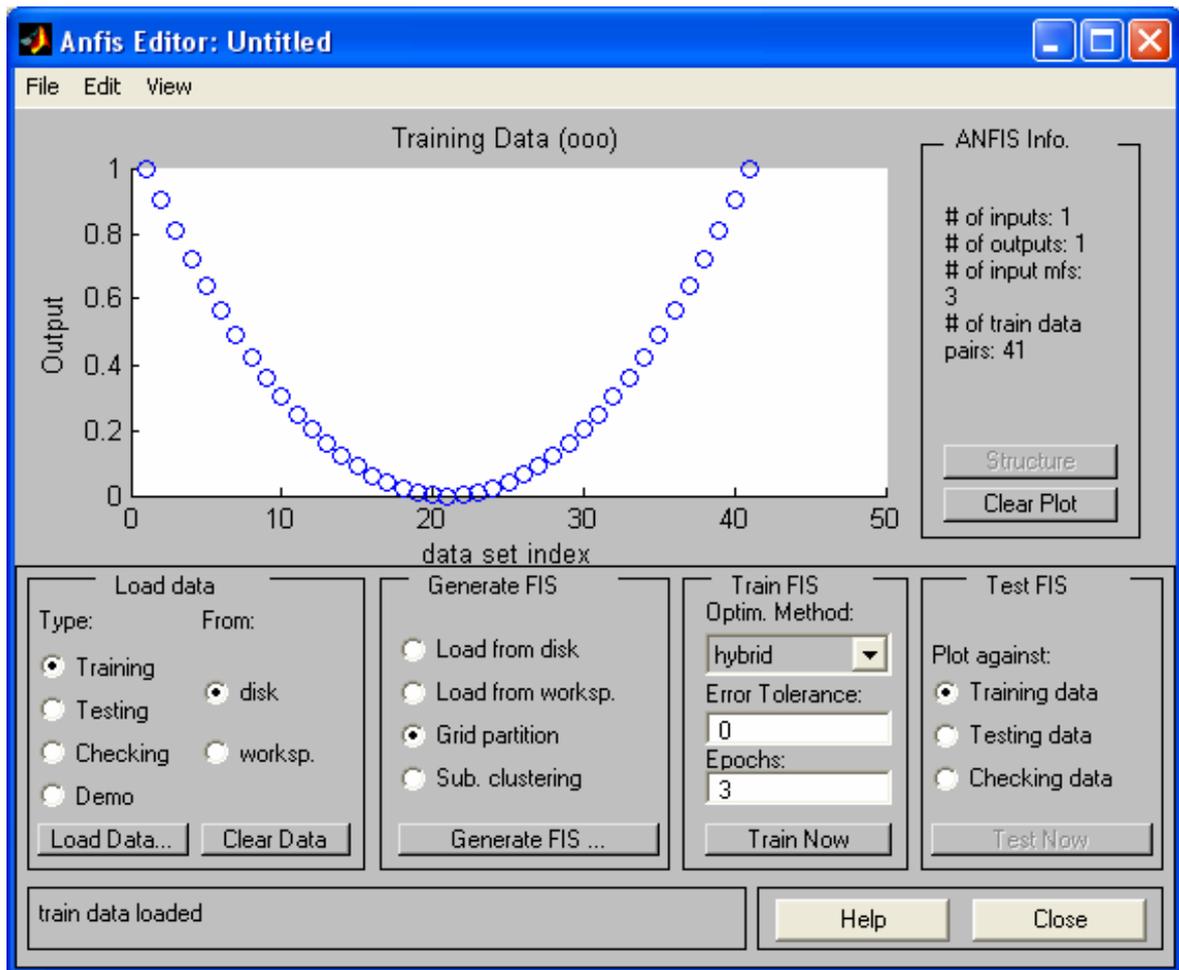


Рис. 8. Результат завантаження навчальної вибірки

5. Створити вихідну систему нечіткого виводу, натиснув кнопку Generate FIS, використовуючи за замовчуванням метод решітчастого розбиття Grid partition. У вікні, що з'явилося, вибираємо тип функції приналежності для вхідної змінної згідно варіанту, кількість функцій приналежності – 5, вихідний параметр – за замовчуванням. Натиснути ОК (рис. 9).

З'явиться вікно з новою нечіткою системою типа Сугено (рис.10).

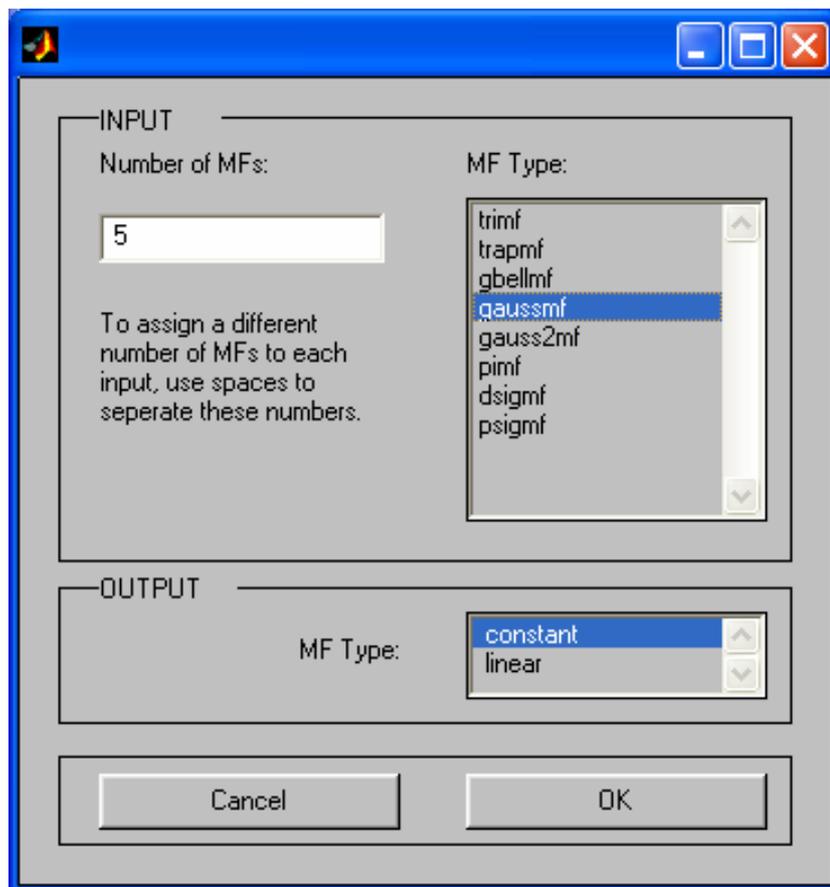


Рис. 9. Вікно введення параметрів для методу решітки

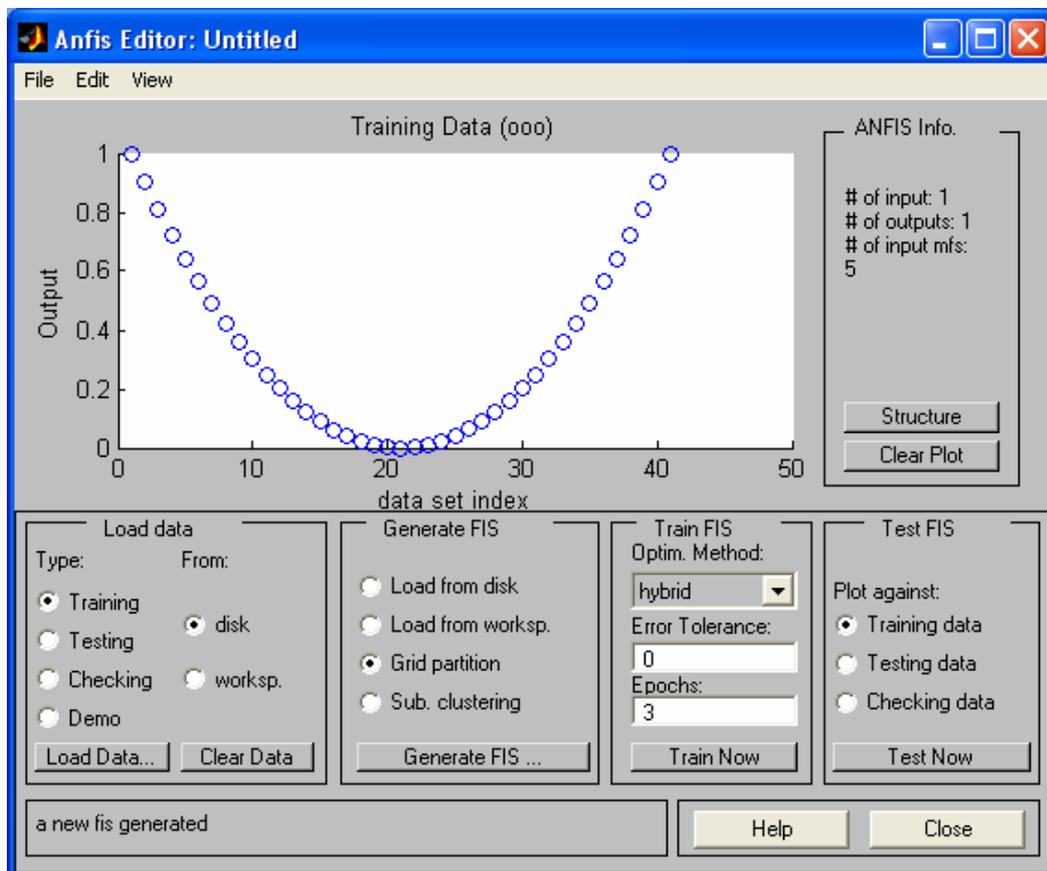


Рис. 10. ANFIS-редактор після генерації FIS-системи

6. Для зображення синтезованої системи у вигляді нейро-нечіткої мережі необхідно натиснути кнопку Structure в полі ANFIS Info (рис. 11).

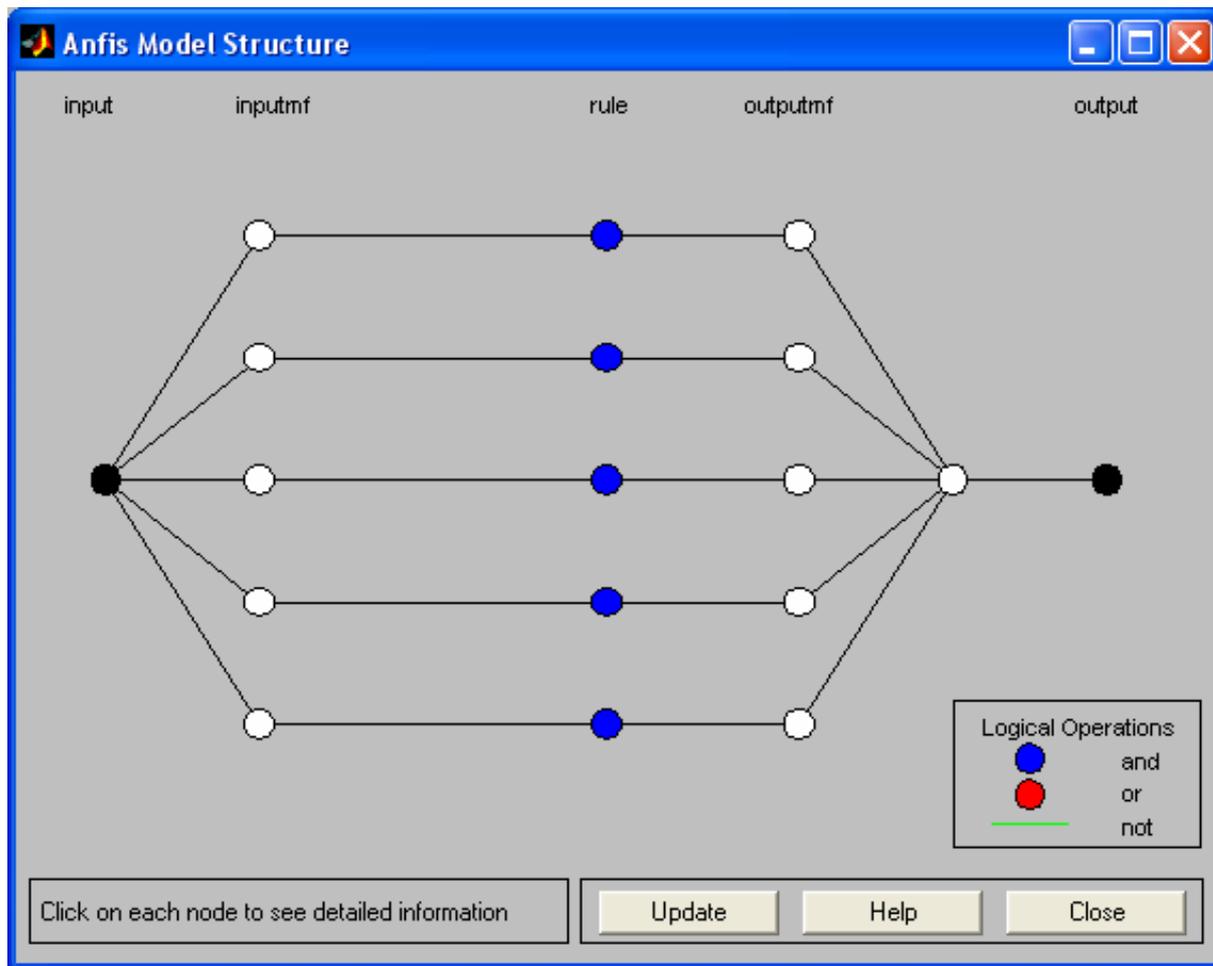


Рис. 11. Структура нейро-нечіткої мережі

7. Провести навчання нейро-нечіткої системи впродовж 10 ітерацій. Для цього в полі Epochs: меню Train FIS набрати 10 і натиснути кнопку Train Now. Динаміка навчання буде виведена в основному вікні ANFIS-редактора (рис. 12).

Число epoch	10
Значення помилки навчання після налаштування	0.017783

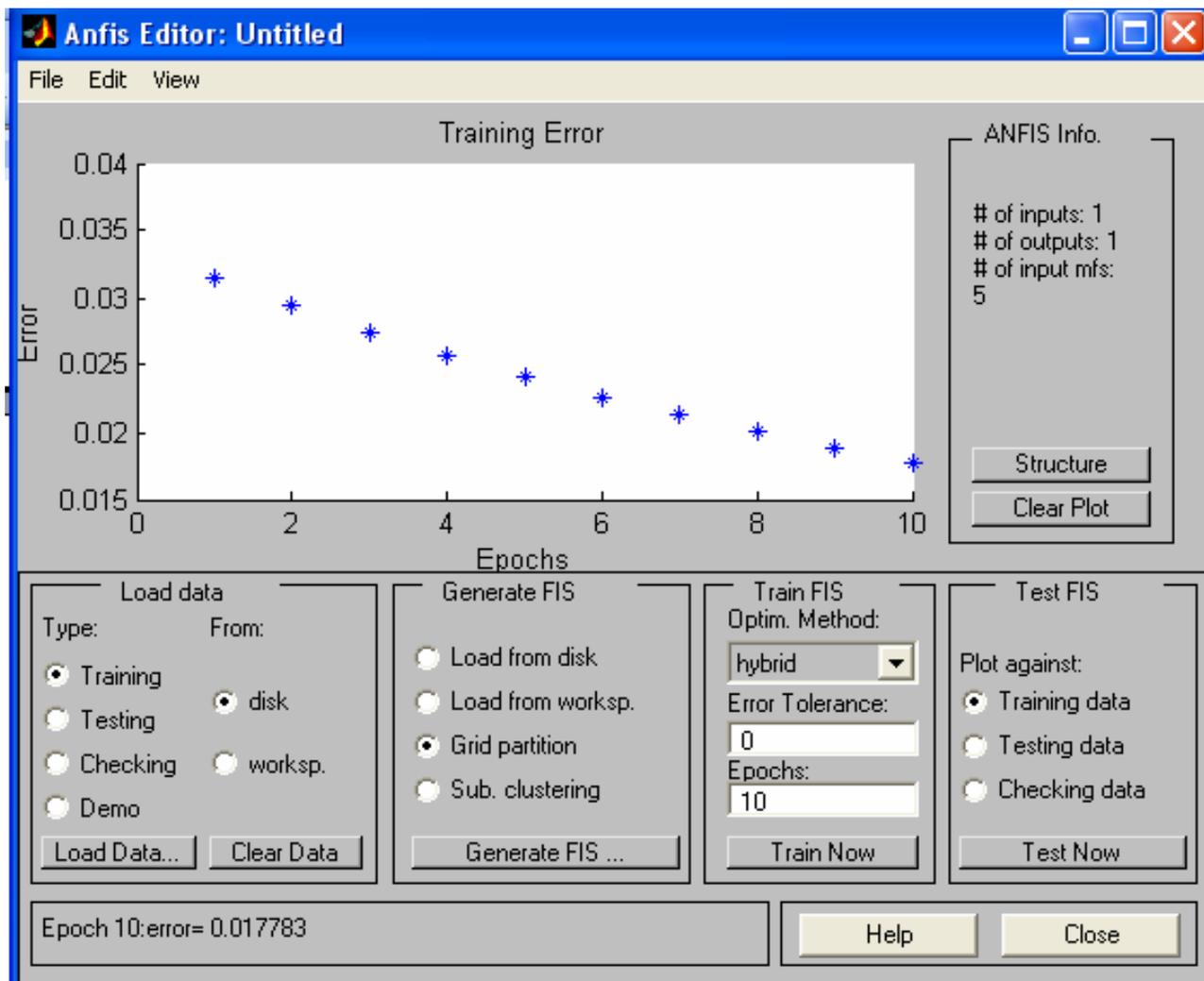


Рис. 12. Результат навчання нейро-нечіткої системи

8. Зберегти отриману нечітку систему: File > Export > To Disk.

9. З отриманою нечіткою системою можна працювати, використовуючи GUI-модулі пакету Fuzzy Logic Toolbox. Для перегляду поверхні системи нечіткого виводу потрібно вибрати в меню View команду Surfase (рис. 13).

10. Для визначення результатів апроксимації функції потрібно відкрити вікно правил системи нечіткого виводу: View > Rules (рис. 14). Результати апроксимації і похибку звести в табл. 1., де y^* вибирається з вікна Rule Viewer (рис. 14), зі стовпчика Output при заданих значеннях x у стовпчику Input згідно табл. 1. (можна задавати x у рядку Input:), $\sigma = |y - y^*|/y^* \cdot 100\%$ – відносна похибка у відсотках.

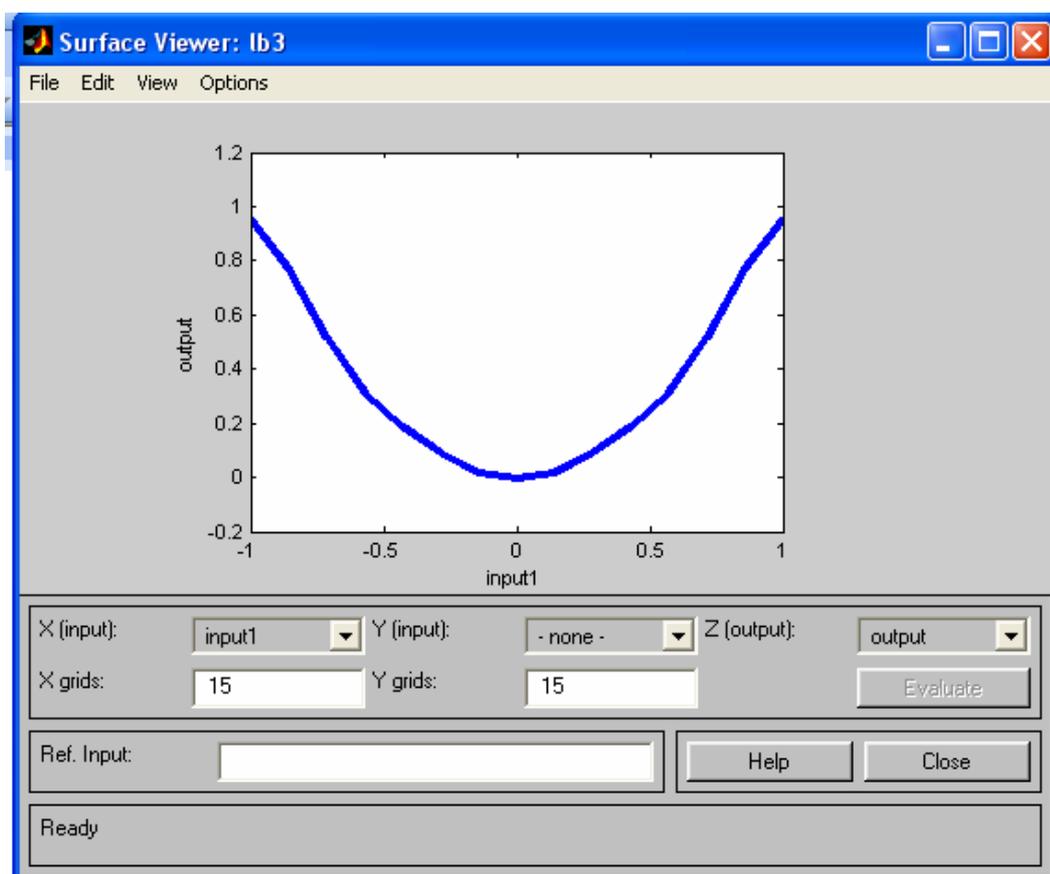


Рис. 13. Перегляд поверхні системи нечіткого виводу

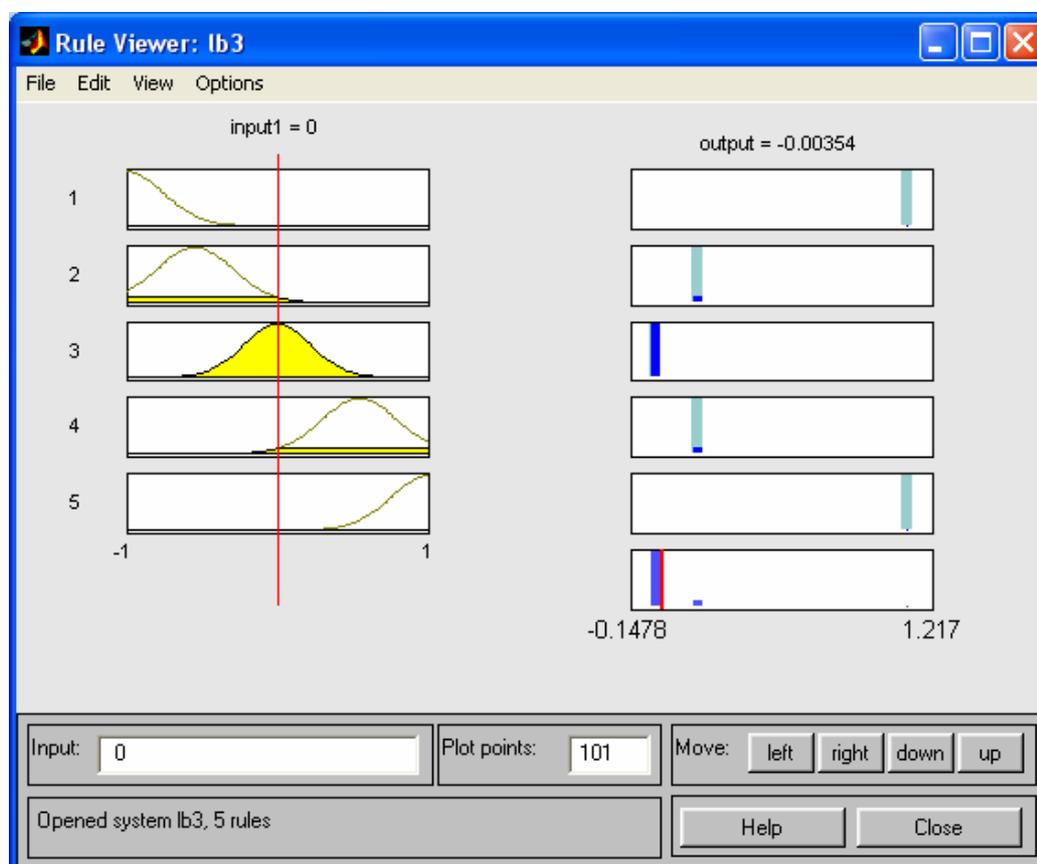


Рис. 14. Правила системи нечіткого виводу

Таблиця 1

Результати апроксимації за допомогою адаптивної мережі

x	y	Вихідна мережа		
		y^*	$ y - y^* $	$\sigma, \%$
-1	1	0.95	0.05	5
-0.8	0.64	0.667	0.027	4.21875
-0.5	0.25	0.24	0.01	4
-0.2	0.04	0.0416	0.0016	4
0.1	0.01	0.00746	0.00254	25.4
0.3	0.09	0.0977	0.0077	8.555556
0.6	0.36	0.345	0.015	4.166667
0.9	0.81	0.831	0.021	2.592593
1	1	0.95	0.05	5

11. Визначити параметри функцій приналежності вхідної змінної: Edit > Membership Functions. (рис. 15), результати занести в таблицю 2.

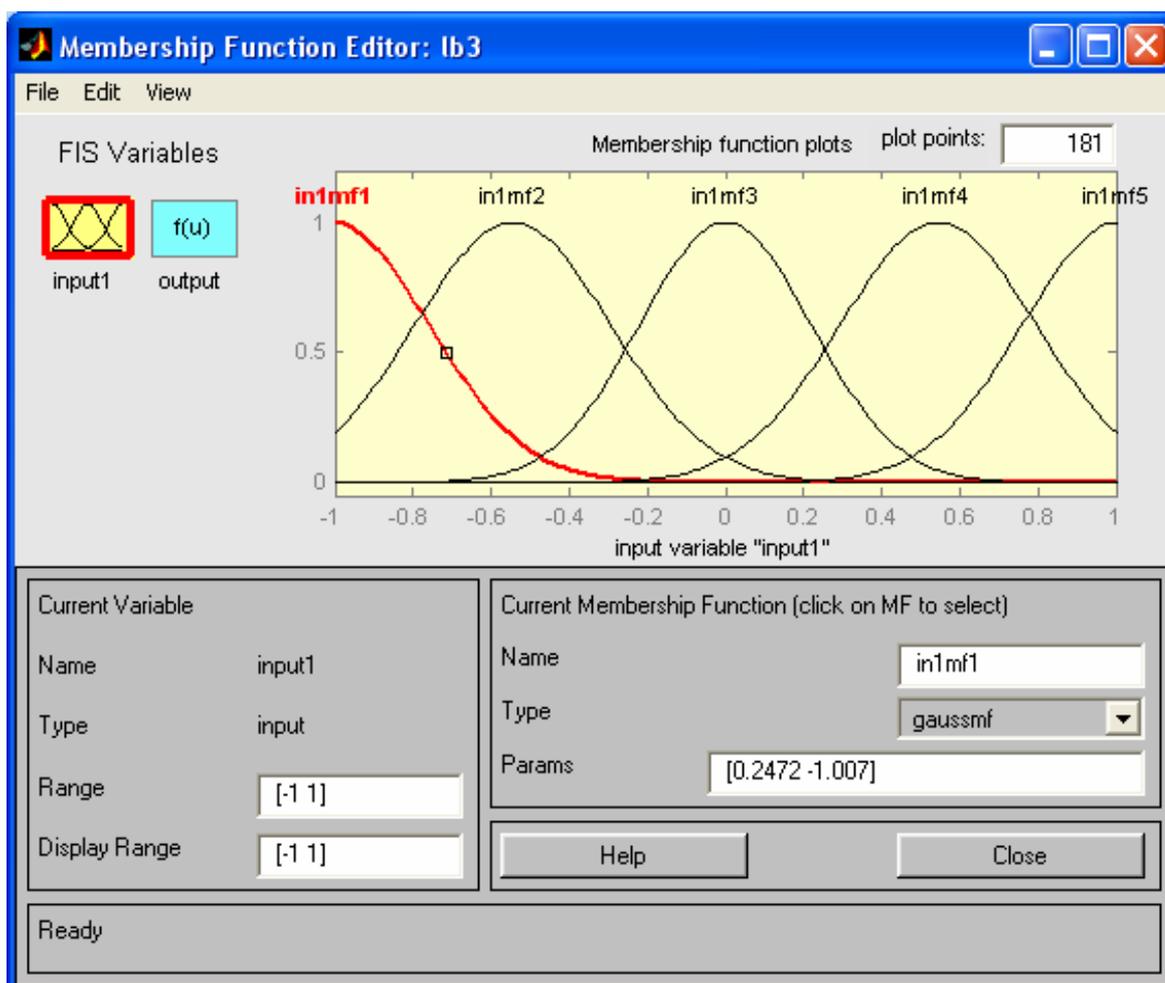


Рис. 15. Вікно редактора функції приналежності

Таблиця 2

Зміна функції приналежності

Функція приналежності	Вихідні параметри	
	A1	A2
in1mf1	0.2472	-1.007
in1mf2	0.251	-0.5448
in1mf3	0.2203	-7.783e-005
in1mf4	0.2512	0.5445
in1mf5	0.2468	1.007

12. Отримати скоректовану нечітку систему. Для цього у вікні ANFIS-редактора провести налаштування системи впродовж 40 ітерацій (рис. 16).

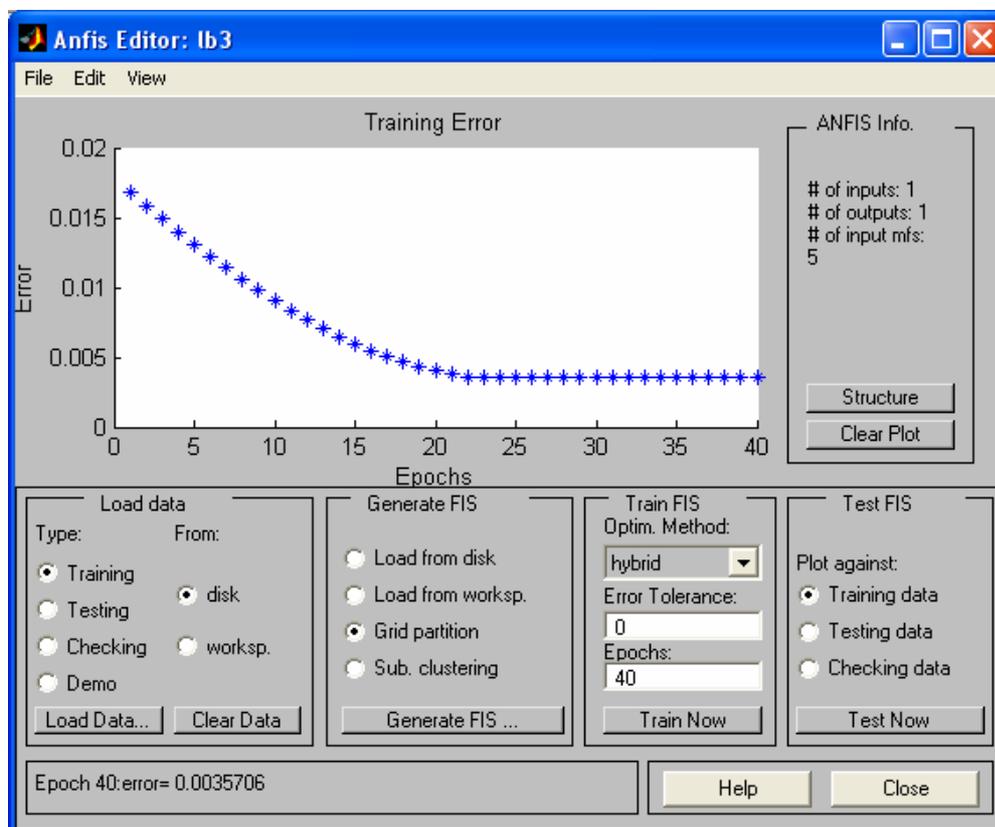


Рис. 16. Динаміка скоректованої нечіткої системи

Число епох	40
Значення помилки навчання після налаштування	0.0035706

Перегляд поверхні скоректованої системи нечіткого виводу показаний на рис.17.

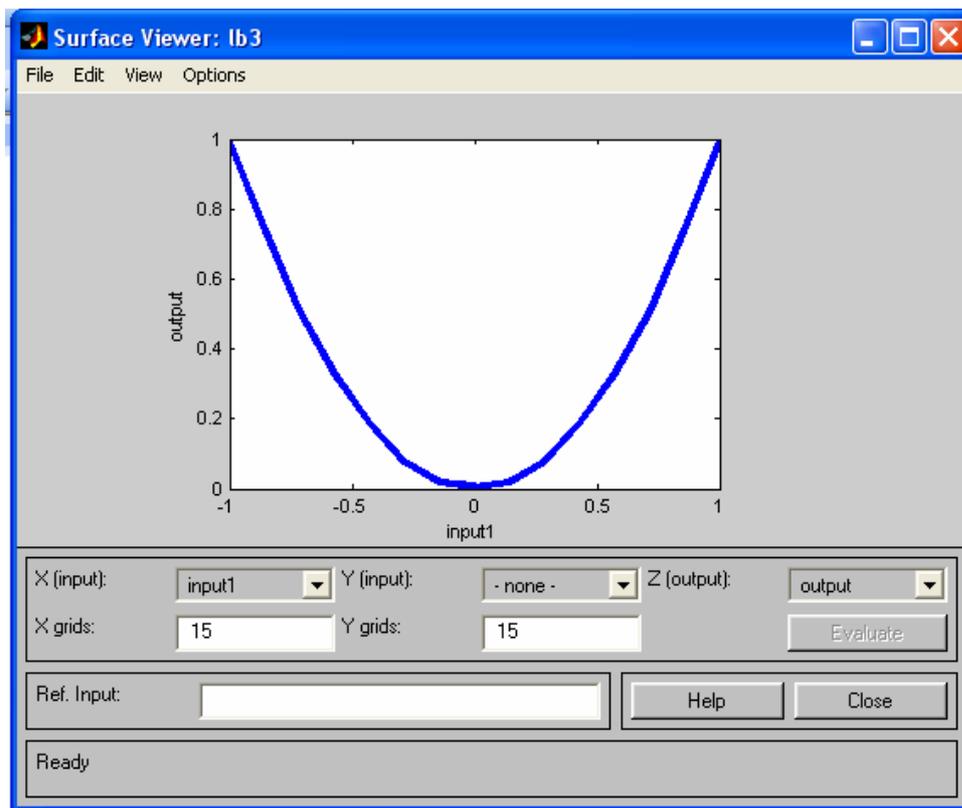


Рис. 17. Скооректована поверхня системи нечіткого виводу

13. По отриманих правилах скооректованої системи нечіткого виводу (рис. 18) визначити результати апроксимації функції і занести в таблицю 3.

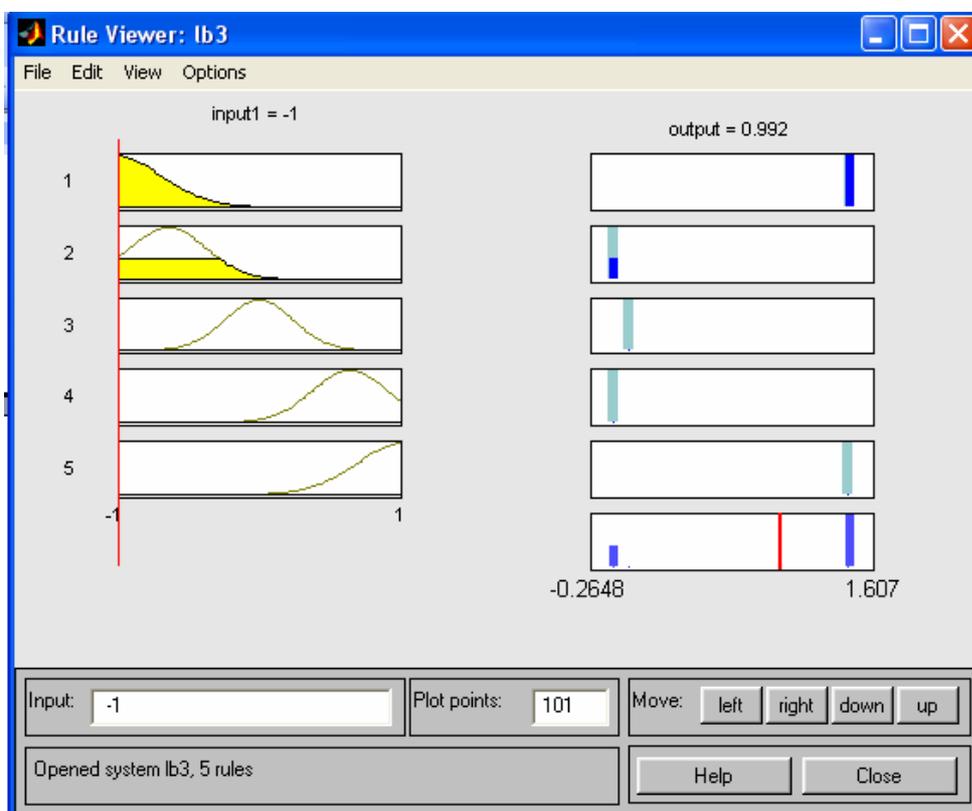


Рис. 18. Правила скооректованої нечіткої системи

Таблиця 3

Результати апроксимації за допомогою скоректованої адаптивної мережі

x	y	Вихідна мережа			Скоректована мережа		
		y^*	$ y - y^* $	$\sigma, \%$	y^*	$ y - y^* $	$\sigma, \%$
-1	1	0.95	0.05	5	0.992	0.008	0.8
-0.8	0.64	0.667	0.027	4.21875	0.642	0.002	0.3125
-0.5	0.25	0.24	0.01	4	0.253	0.003	1.2
-0.2	0.04	0.0416	0.0016	4	0.0351	0.0049	12.25
0.1	0.01	0.00746	0.00254	25.4	0.0121	0.0021	21
0.3	0.09	0.0977	0.0077	8.555556	0.0875	0.0025	2.777778
0.6	0.36	0.345	0.015	4.166667	0.358	0.002	0.555556
0.9	0.81	0.831	0.021	2.592593	0.816	0.006	0.740741
1	1	0.95	0.05	5	0.994	0.006	0.6

14. Визначити параметри функцій приналежності з вікна редактора (рис. 19) та занести їх в таблицю 4.

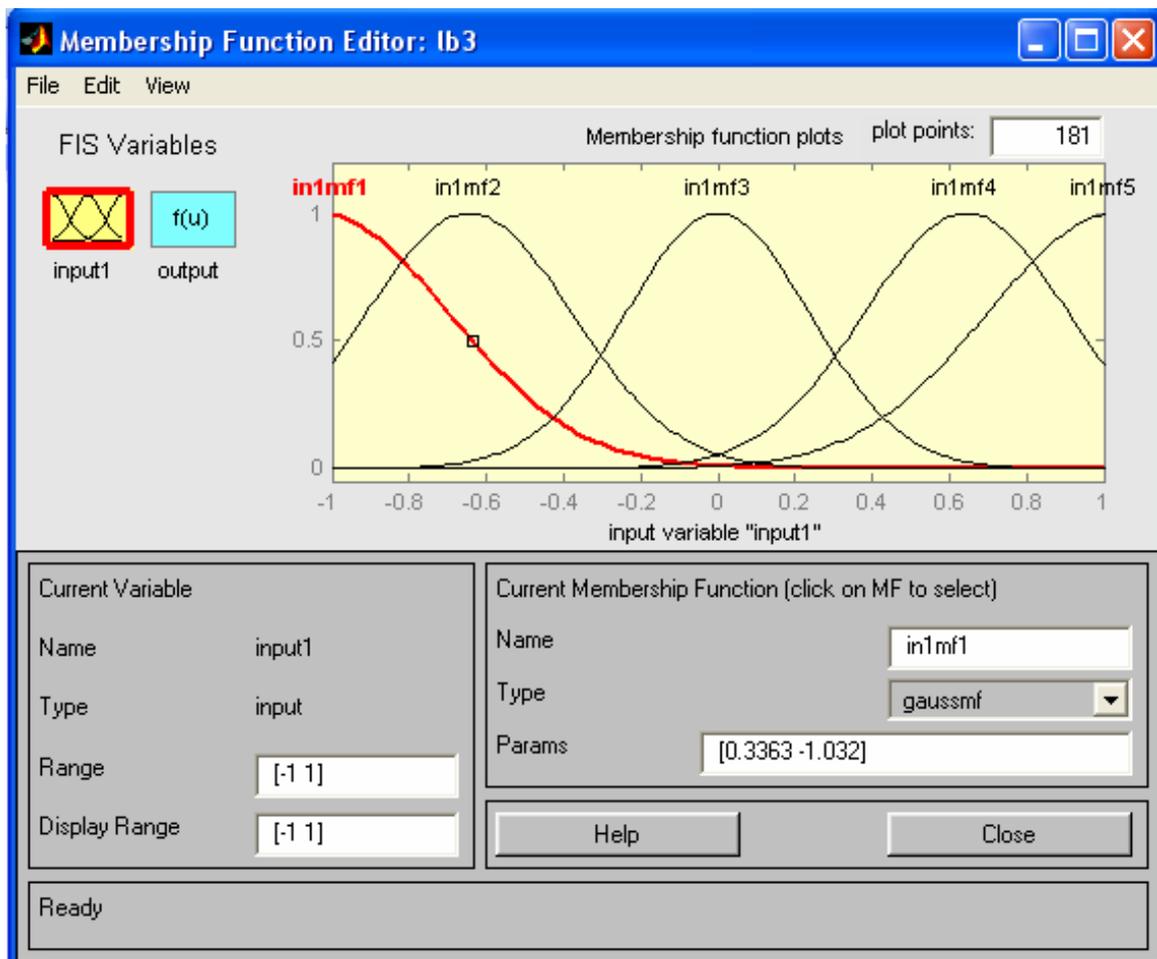


Рис. 19. Вікно редактора функції приналежності

Таблиця 4

Зміна функції приналежності

Функція приналежності	Вихідні параметри		Скоректовані параметри	
	A1	A2	A1	A2
in1mf1	0.2472	-1.007	0.3363	-1.032
in1mf2	0.251	-0.5448	0.2675	0.6451
in1mf3	0.2203	-7.783e-005	0.2388	-0.0007073
in1mf4	0.2512	0.5445	0.2659	0.6438
in1mf5	0.2468	1.007	0.338	1.03

15. Зберегти скоректовану нечітку систему.

16. Проаналізувати результати апроксимації залежності, що описується деякою математичною функцією, за допомогою адаптивної системи нейро-нечіткого виводу. Порівняти з результатом апроксимації функції, виконаної у попередній лабораторній роботі.

Варіанти завдання

Створити систему нечіткого виводу для апроксимації функції $y=k*x^2$ (варіант див. в таблиці 5). Для апроксимації взяти 7 значень функції.

Таблиця 5

Варіанти завдання для апроксимації функції

Варіант	1	2	3	4	5	6
k	1.5	1.8	1.3	1.9	2	2.2
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Варіант	7	8	9	10	11	12
k	1.4	2.5	2.3	2.4	3.5	3.8
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Варіант	13	14	15	16	17	18
k	3.3	3.9	3	3.2	3.4	4
mf input	trimf	trapmf	gbellmf	gaussmf	gauss2mf	dsigmf

Контрольні питання

1. ANFIS-редактор, призначення, завантаження.
2. Дайте визначення нейро-нечіткої мережі.
3. Меню Edit.
4. Область візуалізації.
5. Область властивостей ANFIS.
6. Область завантаження даних.
7. Область генерування вихідної системи нечіткого логічного виводу.
8. Область навчання.
9. Область тестування.
10. Область виводу поточної інформації.
11. Охарактеризуйте структуру нейро-нечіткої мережі.
12. Опишіть процес розробки нейро-нечіткої мережі в середовищі MATLAB.
13. Як перевірити адекватність побудованої нейро-нечіткої мережі?

Лабораторна робота № 4

Тема: ДОСЛІДЖЕННЯ АВТОМАТИЧНОЇ СИСТЕМИ УПРАВЛІННЯ З FUZZY-РЕГУЛЯТОРОМ ПІ-ТИПА

Мета роботи: придбання навиків роботи в пакеті Fuzzy Logic Toolbox з блоками Simulink.

Постановка завдання: дослідження системи автоматичного регулювання з ПІ-подібним нечітким регулятором.

Короткі теоретичні відомості

Системи нечіткого виводу, створені тим або іншим чином за допомогою пакету Fuzzy Logic Toolbox, допускають інтеграцію з інструментами пакету Simulink, що дозволяє виконувати моделювання систем в рамках останнього.

Для створення і моделювання систем нечіткого виводу використовуються блоки Fuzzy Logic Controller або Fuzzy Logic Controller with Ruleviewer, які містяться в бібліотеці Fuzzy Logic Toolbox (рис. 1).

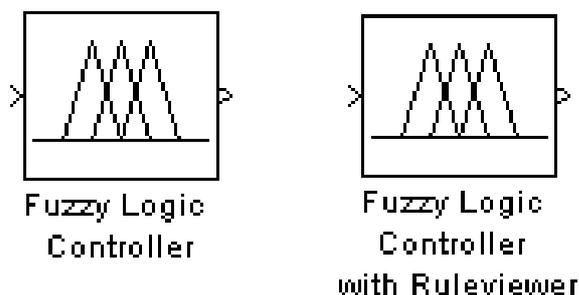


Рис. 1. Вміст бібліотеки Fuzzy logic Toolbox

Нечіткі логічні регулятори (НЛР), нелінійні за своєю суттю, можуть управляти лінійними об'єктами краще, ніж класичні регулятори, а також управляти нелінійними об'єктами, для яких лінійні регулятори непридатні.

Нечіткі регулятори можуть розглядатися як природний розвиток ідей ПІД-управління. При правильному проектуванні вони забезпечують кращі експлуатаційні показники, тобто можуть зменшити перерегулювання, час перехідного процесу і сталу похибку.

Структури нечітких регуляторів

Зазвичай НЛР включаються послідовно з об'єктом управління, подібно до традиційних регуляторів (рис. 2).

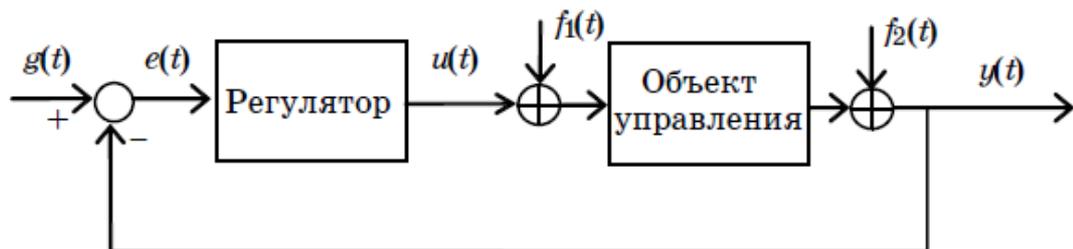


Рис. 2. Система управління із зворотним зв'язком

Вхідні і вихідні сигнали НЛР піддаються перетворенню (рис. 3).

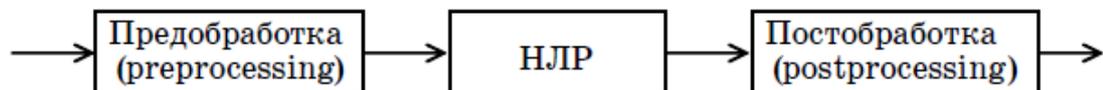


Рис. 3. Обробка сигналів на вході і виході НЛР

У загальному випадку передобробка вхідного сигналу може включати такі операції, як:

- квантифікація (округлення);
- нелінійне шкалювання;
- фільтрація і видалення шумів;
- диференціювання і (або) інтеграція;
- усереднення на деякому інтервалі часу, екстраполяція, інтерполяція;
- обчислення довільних функцій над комбінаціями вхідних змінних для здобуття семантично значимих значень.

Проте найчастіше передобробка зводиться до нормалізації (N) вхідного сигналу – приведення його до діапазону $[-1, 1]$. Тоді постобробка зводиться до денормалізації (DN) вихідного сигналу НЛР (рис. 4).

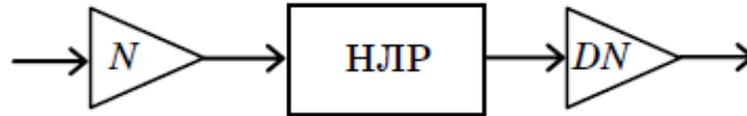


Рис. 4. Нормалізація і денормалізація сигналів на вході і виході НЛР

Якщо x – вхідна змінна НЛР, то нормалізація відбувається шляхом множення x на масштабуючий коефіцієнт:

$$x_N = xN = \frac{x}{|x_{\max}|}.$$

Денормалізація вихідної величини y зводиться до зворотної операції:

$$y = y_N DN = y_N |y_{\max}|,$$

де y_{\max} – максимальне значення управління, що подається на об'єкт.

Кількість і вигляд вхідних і вихідних сигналів НЛР залежать від вирішуваного завдання, проте найчастіше входами НЛР служать (у різних комбінаціях) помилка управління, її похідна і інтеграл. Відповідно розрізняють НЛР П-типу, ПД-типу, ІІ-типу або ІІД-типу (рис. 5-8).

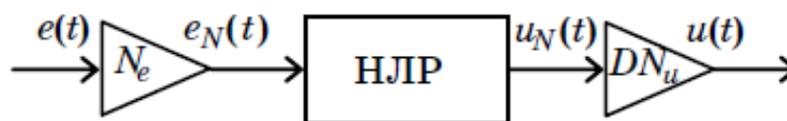


Рис. 5. НЛР П-типу (НЛР_П)

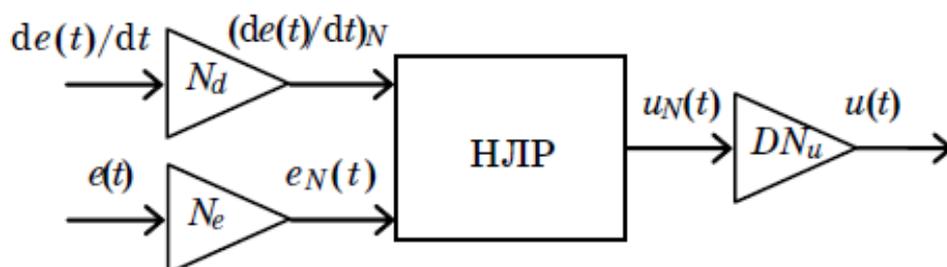


Рис. 6. НЛР ПД-типу (НЛР_ПД)

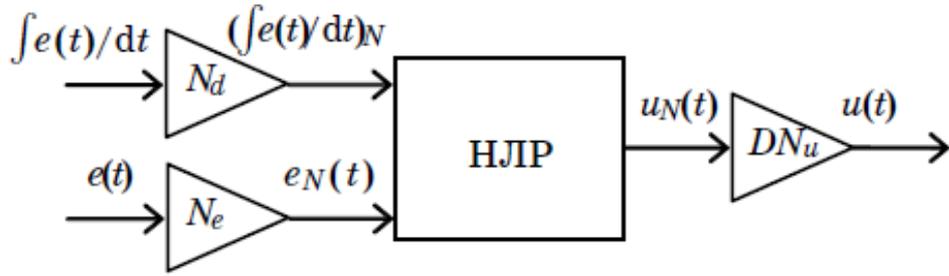


Рис. 7. НЛР ПІ-типу (НЛР_ПІ)

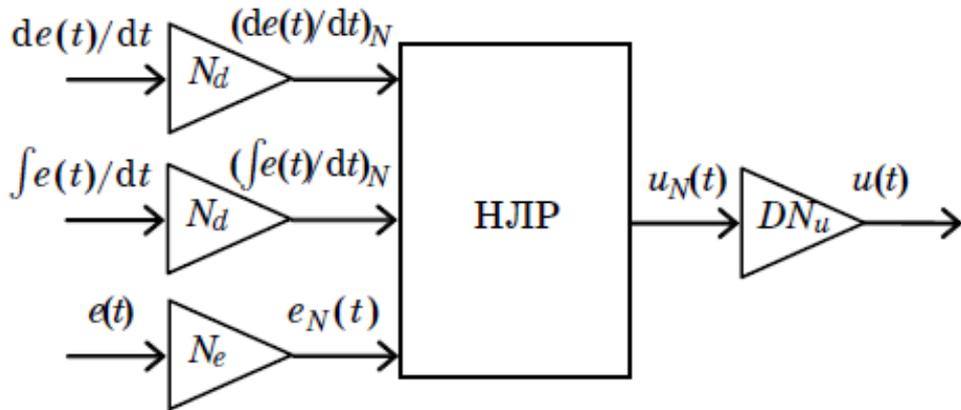


Рис. 8. НЛР ПІД-типу (НЛР_ПІД)

Позначимо через E вхідний вектор НЛР (рис. 9). Він перетвориться в нечітку форму E^* (блок фазифікації F), потім виконується нечіткий логічний вивід в базі правил, внаслідок чого виходить нечітка вихідна змінна u^* . Після дефазифікації (блок DF) на об'єкт управління поступає «чіткий» сигнал управління u .

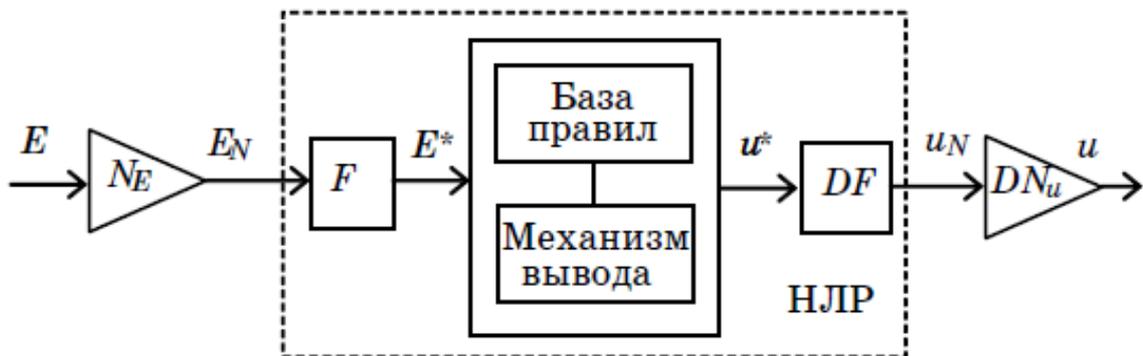


Рис. 9. НЛР ПІ-типу

Операції фазифікації і дефазифікації залежать від параметрів термів, що описують вхідні і вихідні змінні. Таким чином, можна сказати, що окрім механізму виводу в НЛР існує база знань, що включає опис термів і правил (рис. 10).

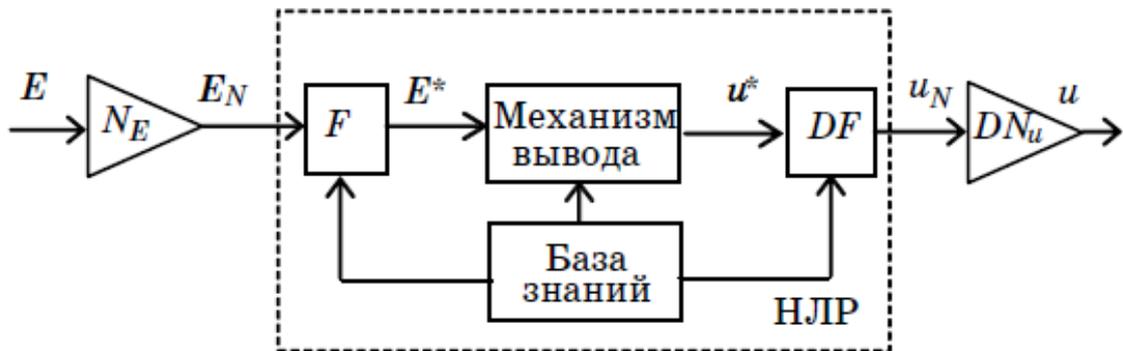


Рис. 10. Уточнений варіант опису НЛР_П

Запишемо k -е правило управління НЛР_ПД у вигляді

$$\text{Если } (x_1 = A_1^k) \text{ и } (x_2 = A_2^k) \text{ и } (x_3 = A_3^k), \text{ то } y = B_k,$$

де x_1 , x_2 і x_3 – відповідно помилка управління, її похідна і інтеграл;

y – сигнал управління;

A_i і B_k – терми, що входять в k -е правило.

На практиці нечіткий вивід реалізується зазвичай в декілька кроків:

1. Виконується фазифікація – обчислюються значення $\alpha_{i,k}$ – міра відповідності i -ї вхідної змінної i -й посилю к-го правила:

$$\alpha_{i,k} = x_i^* \cap A_i^k,$$

де x_i^* – нечітка множина (синглетон), утворена з x_i .

2. Визначається міра відповідності всіх посилок k -го правила поточним вхідним даним:

$$\beta_k = \bigwedge_{i=1}^N \alpha_{i,k},$$

де N – кількість посилок правила; T – використовується T -норма;

β_k – міра спрацьовування (запуску) правила.

3. Визначається вихідний сигнал кожного правила:

$$\mu_{B'_k}(y) = I(\beta_k, \mu_{B_k}(y)),$$

де I – використувувана імплікація; $\mu_{B_k}(y)$ – функція приналежності висновку k -го правила.

4. Загальний вихідний сигнал системи правил B' визначається шляхом агрегації вихідних сигналів окремих правил:

$$\mu_{B'}(y) = \bigcup_k \mu_{B'_k}(y).$$

Таким чином, закон управління реалізується як сукупна дія всіх правил, записаних в базі правил:

$$R = R_1 \vee R_2 \vee R_3 \dots \vee R_k.$$

Відповідно до рис. 10 синтез НЛР пов'язаний з визначенням наступних параметрів:

- 1) коефіцієнтів нормалізації і денормалізації;
- 2) функцій приналежності термів лінгвістичних змінних;
- 3) керуючих правил НЛР.

Можливі різні постановки завдання синтезу НЛР, але для додання їй реалістичного характеру бажано мінімізувати кількість параметрів.

Залежно від кількості входів міняється кількість посилок нечітких керуючих правил.

Для НЛР_П кількість правил дорівнює кількості термів, що описують лінгвістичну змінну «Помилка управління». Для інших типів НЛР потенційна кількість правил дорівнює декартову добутку потужностей терм-множин вхідних змінних. Якщо вважати, що кожна вхідна змінна має однакову кількість термів, то число правил можна оцінити відповідно до таблиці. 1.

Потенційна кількість правил для різних типів НЛР

Кількість термов	Кількість правил		
	НЛР_П	НЛР_ПД (ПИ)	НЛР_ПД
3	3	9	27
5	5	25	125
7	7	49	343

Завдання. Дослідити АСР з аналоговим ПІ-регулятором і fuzzy-регулятором. Як модель об'єкту прийняти аперіодичну ланку першого порядку із запізнюванням. Передавальна функція об'єкту управління має вигляд ($\tau=1$ с):

$$W_o(s) = \frac{1.5}{1.5 \cdot s + 1} \cdot e^{-s}$$

Порядок виконання роботи

1. Запустити пакет Simulink, натиснувши кнопку Simulink на панелі інструментів командного вікна Matlab.

2. У пакеті Simulink створити модель (File > New > Model) і зберегти її у файлі model_1.mdl. Для зображення структури АСР вибрати з бібліотеки системи Simulink наступні блоки (таблиця 2) і розташувати їх у вікні моделі (рис. 11). При цьому напрям стрілок деяких блоків змінити за допомогою контекстного меню Format > Flip block і перемістити назви у деяких блоків за допомогою команд контекстного меню Format > Flip name.

Блоки Simulink для побудови АСР

Найменування блоків	Розділи бібліотеки Simulink
Constant	Sources
Sum	Math Operators
Transfer Fcn	Continuous
Transport Delay	Continuous
Integrator	Continuous

Fuzzy Logic Controller	Fuzzy Logic Toolbox
Scope	Sinks
Mux	Signal Routing

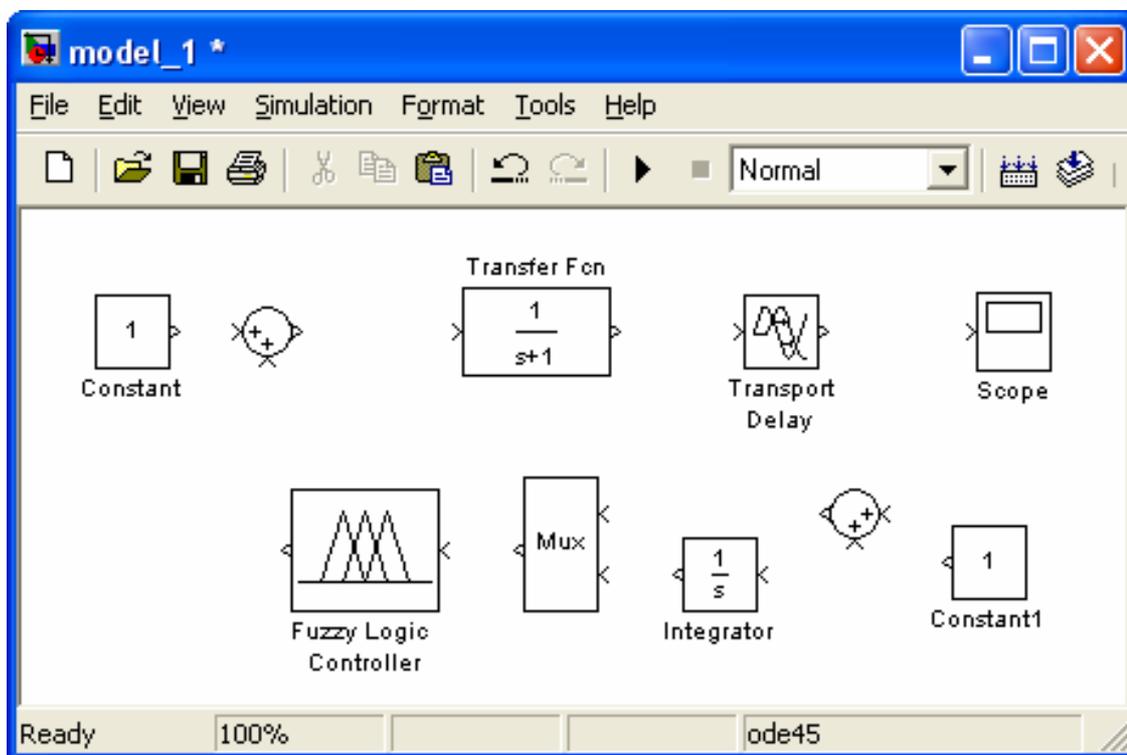


Рис. 11. Вікно моделі, що містить блоки для АСР

3. З'єднати блоки, змінити їх назви і задати параметри суматорів, передавальної функції і завдання як показано на рис. 12.

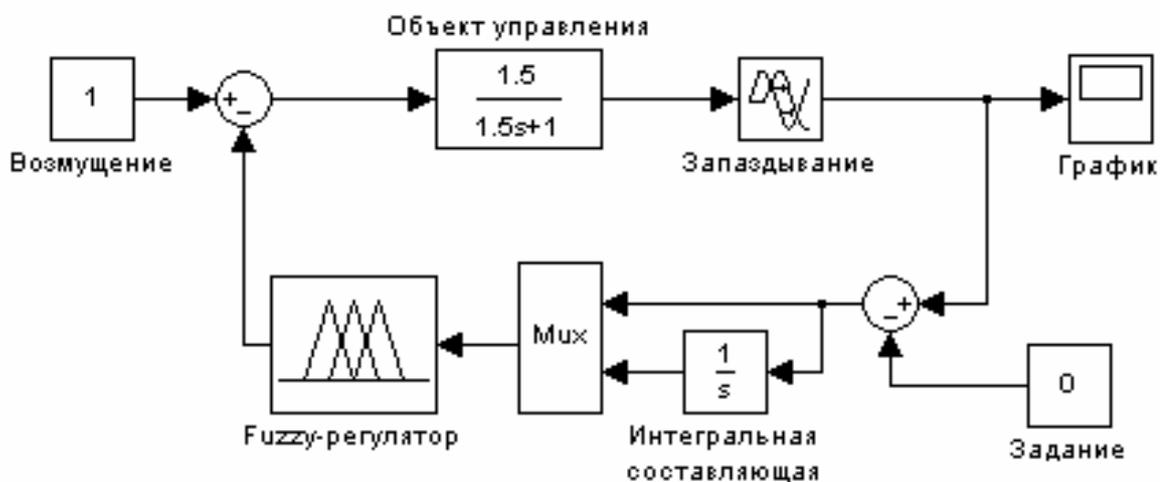


Рис. 12. Модель одноконтурної системи автоматичного регулювання з подібним для ПІІ fuzzy-регулятором

4. За допомогою інструментів графічного інтерфейсу користувача (GUI) пакету "Fuzzy Logic Toolbox" можна створити нечітку систему, що реалізовує типовий аналоговий ПІ-регулятор. Відмітимо, що за допомогою пакету "Fuzzy Logic Toolbox" можна будувати нечіткі системи двох типів – Мамдані і Сугено. Зупинимося на системі типа Мамдані.

Командою `fuzzy` у вікні Matlab викликати вікно редактора фазі-інференційної системи (Fuzzy Inference System Editor). Вибрати тип системи – Мамдані, задати два входи (Edit > Add Variable... > Input) - для пропорційної і інтегральної складових. Дати назви вхідним змінним, наприклад, x_1 і x_2 , а вихідній – y (рис. 13).

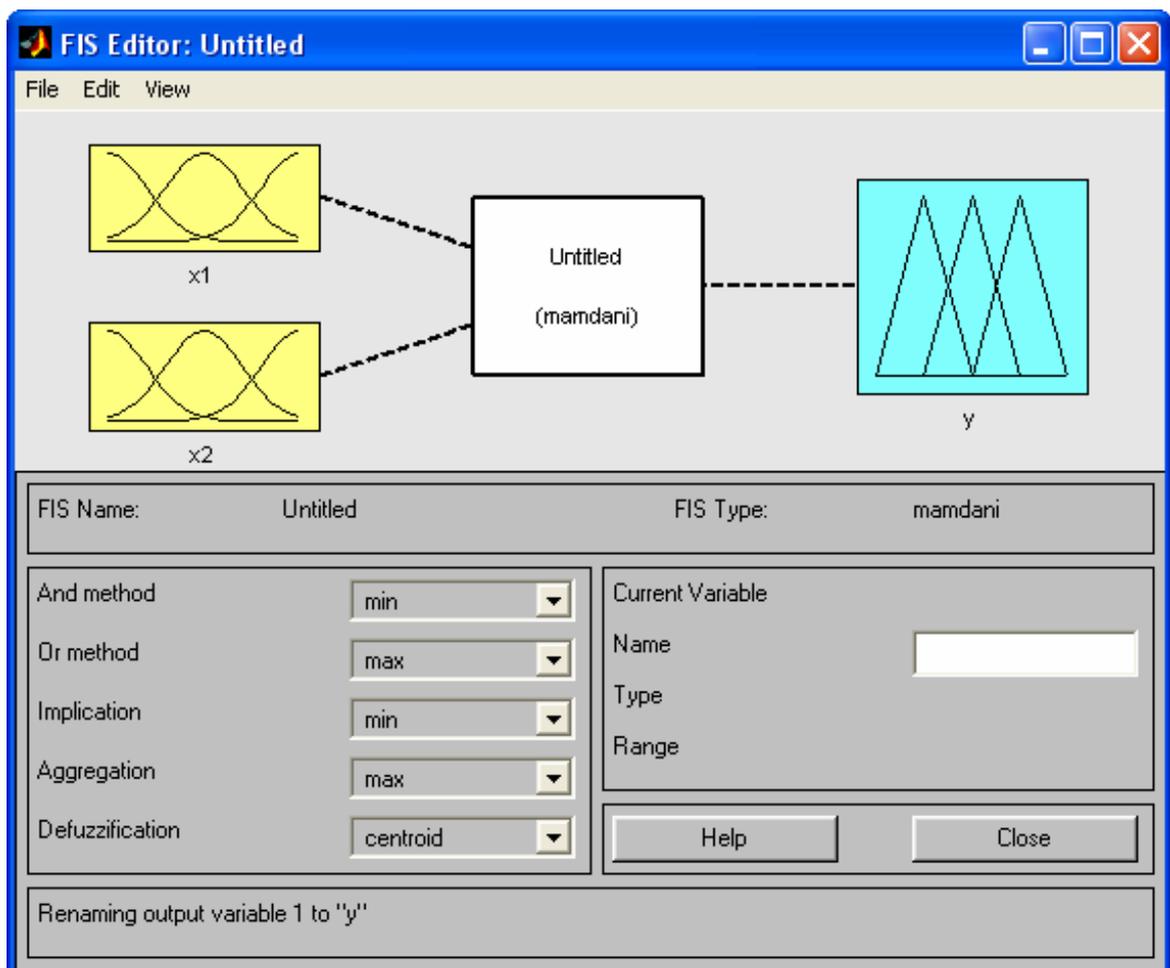


Рис. 13. Вікно редактора FIS

5. Викликати вікно редактора функцій приналежності Membership Function Editor, клацнувши подвійним клацанням миші по зображенню змінної x_1 або за допомогою меню Edit. Для лінгвістичного опису кожної змінної вибрати сім трикутних термів (NB, NM, NS, ZE, PS, PM, PB) за допомогою команди Edit > Add MFs, заздалегідь видаливши вихідні терми (Edit > Remove All MFs).

6. У цьому ж вікні задати діапазони зміни змінних. Для вхідних змінних регулятора рекомендуються симетричні діапазони зміни, при цьому:

$$x_1 \in \left[-\frac{1}{K_1}; \frac{1}{K_1} \right] \quad \text{і} \quad x_2 \in \left[-\frac{1}{K_0}; \frac{1}{K_0} \right],$$

де K_1 , K_0 – оптимальні налаштування пропорційної і інтегруючої частин ПІ-регулятора відносно якогось критерію.

Значення параметрів налаштування ПІ-регулятора для даного об'єкту наступні: $K_1=0.6$ $K_0=0.667$ (згідно таблиці 5), тоді $x_1 \in [-1.667; 1.667]$ і $x_2 \in [-1.5; 1.5]$ (рис. 14).

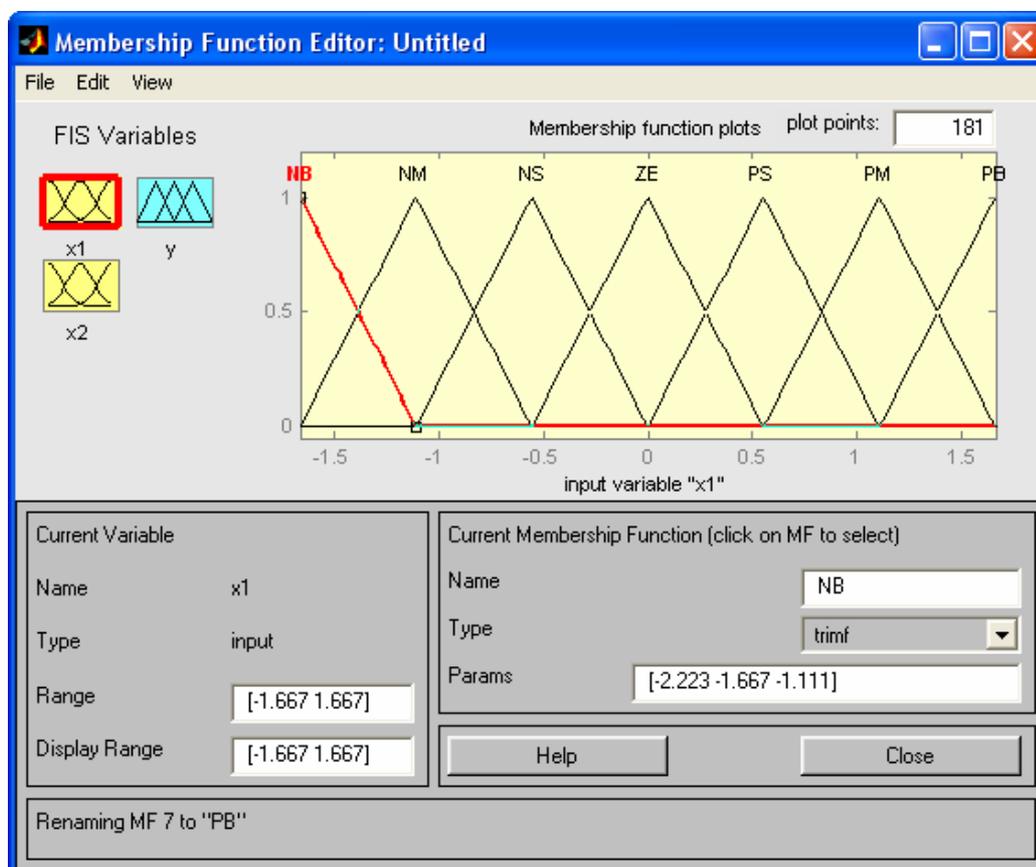


Рис. 14. Вікно редактора функцій приналежності змінної x_1

Аналогічним чином налаштуємо вхідну змінну x_2 (рис. 15).

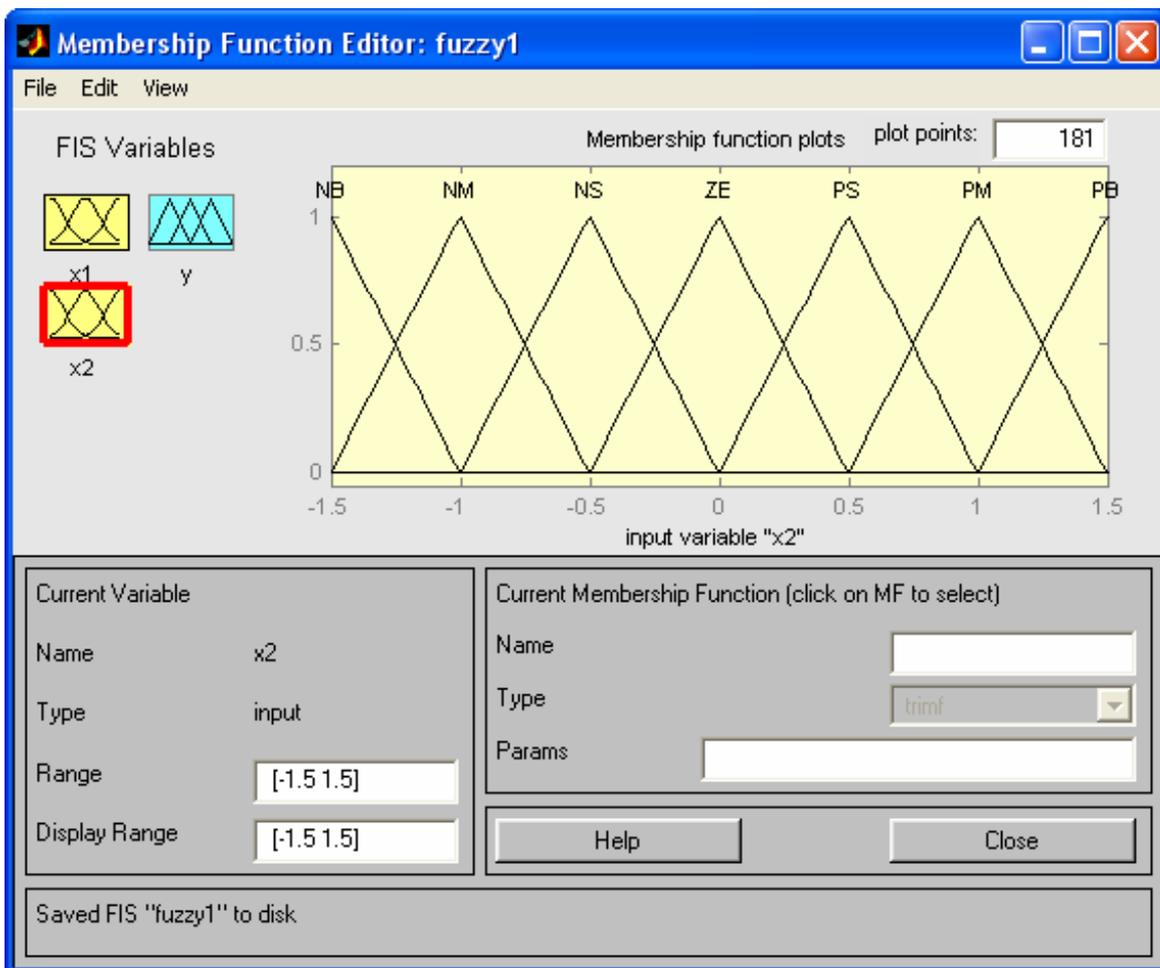


Рис. 15. Вікно редактора функцій приналежності змінної x_2

7. Для вихідної змінної регулятора діапазон зміни рекомендується брати у вигляді $y \in [0; C]$, де верхня межа C при одиничній ступінчастій дії варіюється від 1.1 до 2, аби вихідний сигнал регулятора міг компенсувати це збурення (рис. 16). По мірі збільшення значення C зменшується динамічна помилка, але зростають час регулювання і число коливань перехідного процесу. Тому рекомендується C приймати рівним 2, коли спостерігається оптимальне співвідношення між величиною динамічної помилки, часом регулювання і кількістю коливань.

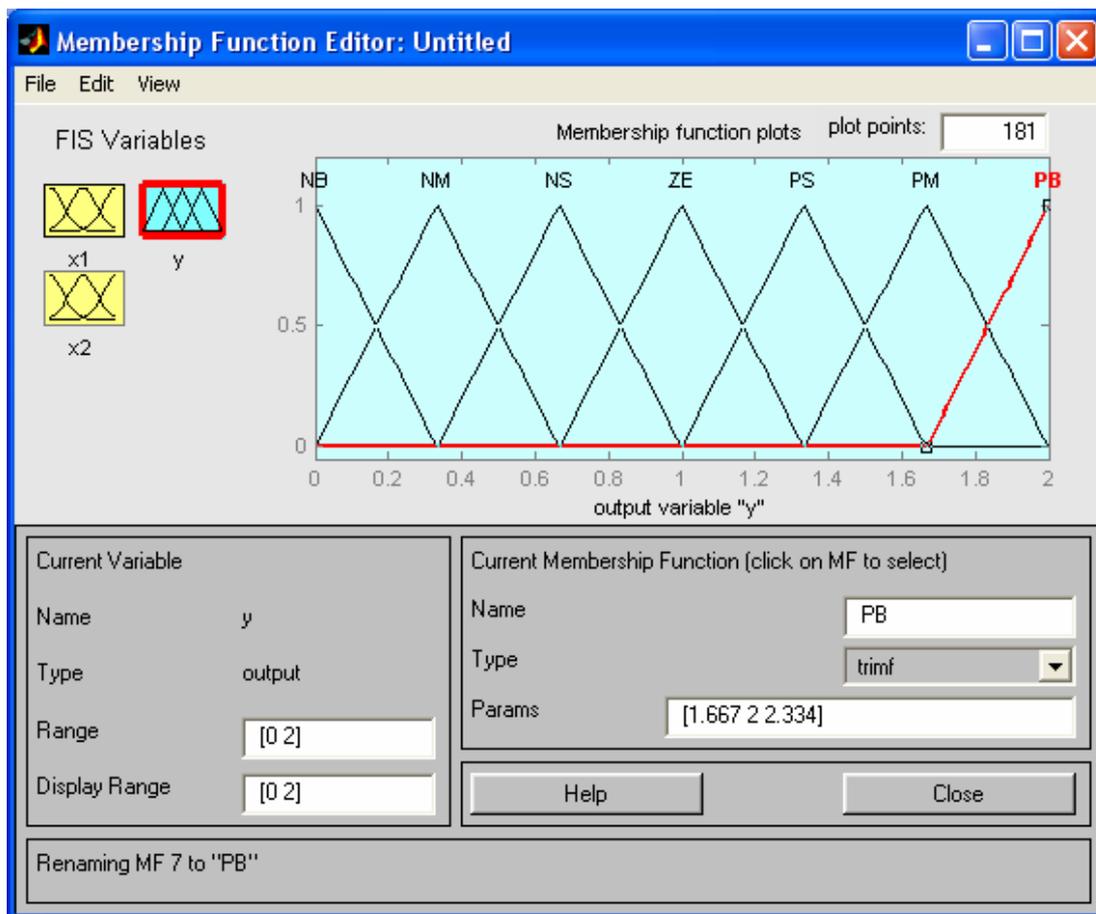


Рис. 16. Вікно редактора функцій приналежності вихідної змінної y

8. Тепер необхідно сформулювати базу правил fuzzy-регулятора. Лінійний безперервний ПІ-регулятор з диференціальним рівнянням

$$y(t) = k_D \cdot \varepsilon(t) + \frac{1}{T_E} \int_0^t \varepsilon(t) \cdot dt$$

можна замінити близьким за стратегією і логікою управління fuzzy-регулятором, якщо в якості його вихідної змінної розглядати приріст керуючої дії Δy . Тоді ПІ-закон регулювання можна представити в наступній диференціальній формі:

$$\frac{dy(t)}{dt} = k_D \cdot \frac{d\varepsilon(t)}{dt} + \frac{1}{T_E} \cdot \varepsilon(t)$$

або в різницевій формі:

$$\Delta y(k) = y(k) - y(k-1) = k_D \cdot \Delta \varepsilon(k) + \frac{\Delta t}{T_E} \cdot \varepsilon(k)$$

Таким чином, для вхідних змінних $\varepsilon(k)$ і $\Delta\varepsilon(k)$ і вихідної $\Delta y(k)$ може бути синтезований fuzzy-регулятор, що реалізовує нелінійний закон, еквівалентний в певному значенні ПІ-регулятору:

$$\Delta y(k) = \mathcal{F} [\Delta \varepsilon(k), \varepsilon(k)]$$

Для нашого випадку x_1 відповідає сигналу неузгодження $\varepsilon(k)$, x_2 відповідає приросту сигналу неузгодження $\Delta\varepsilon(k)$, а y відповідає $\Delta y(k)$. Лінгвістичні правила для такого ПІ-подібного fuzzy-регулятора приведені в таблиці 3.

Таблиця 3

Лінгвістичні правила для нечіткого ПІ-регулятора

ε (x1) $\Delta\varepsilon$ (x2)	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

9. Викликати вікно редактора правил (Rule Editor) за допомогою меню Edit > Rules... або натисненням Ctrl+3 і заповнити список правилами з таблиці 3. Правила формуються за типом: ЯКЩО ...І ..., ТО. (рис. 17).

10. Можна переглянути простір управління, викликавши вікно перегляду простору управління (Surface Viewer) з меню View або комбінацією клавіш Ctrl+6 (рис. 18).

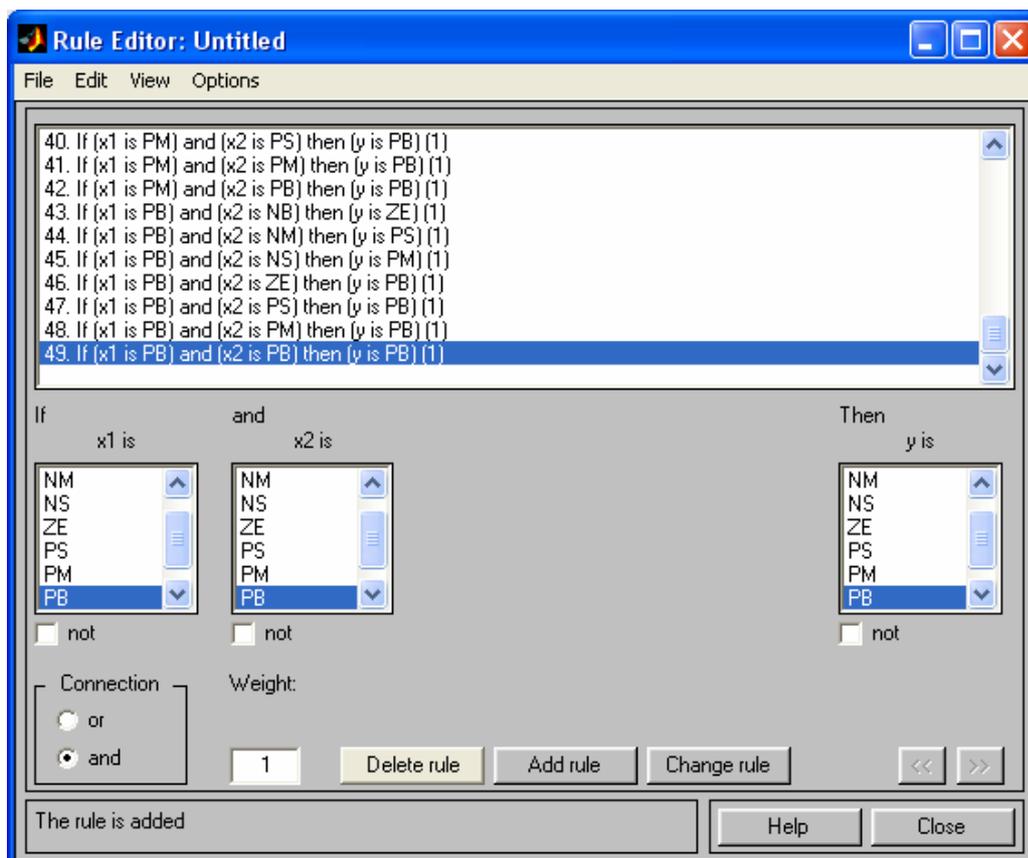


Рис. 17. Вікно редактора правил

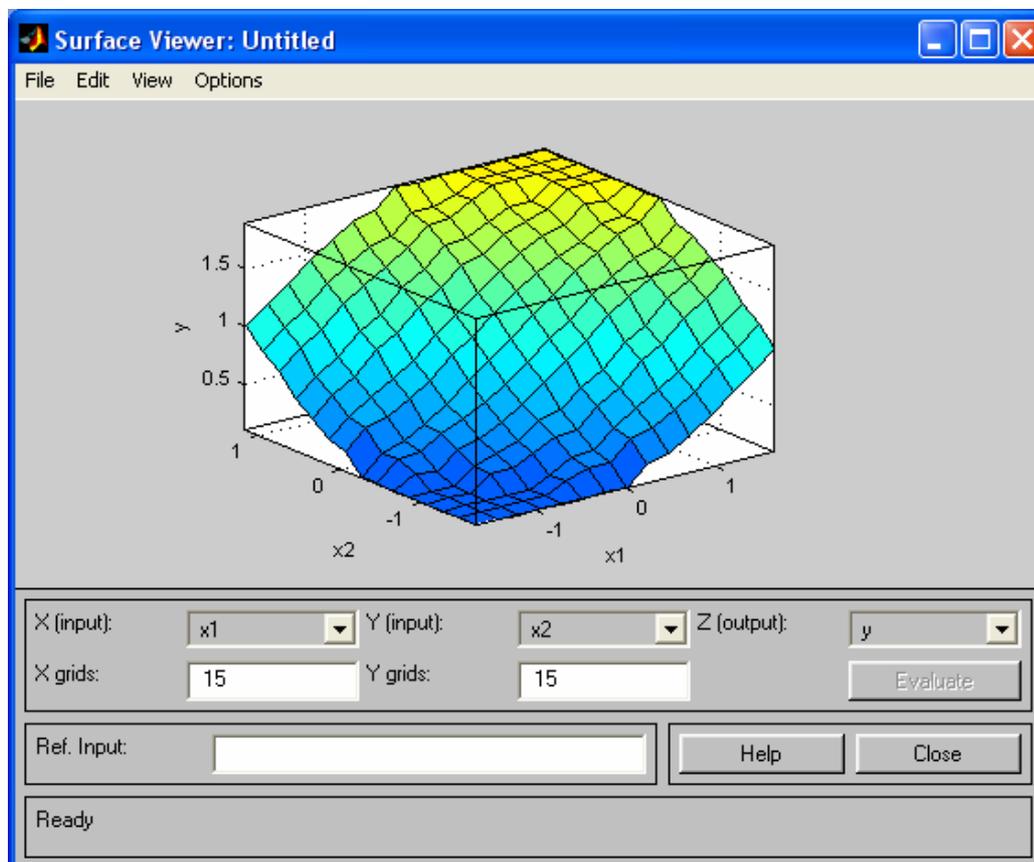


Рис. 18. Вікно перегляду простору управління

Отриманий файл зберегти як fuzzy1.fis (File >Export >To Disk.).

11. Перейти у вікно моделі. У вікні параметрів блоку Fuzzy Logic Controller в рядку FIS File or Structure вказати ім'я файлу – fuzzy1 (рис. 19).

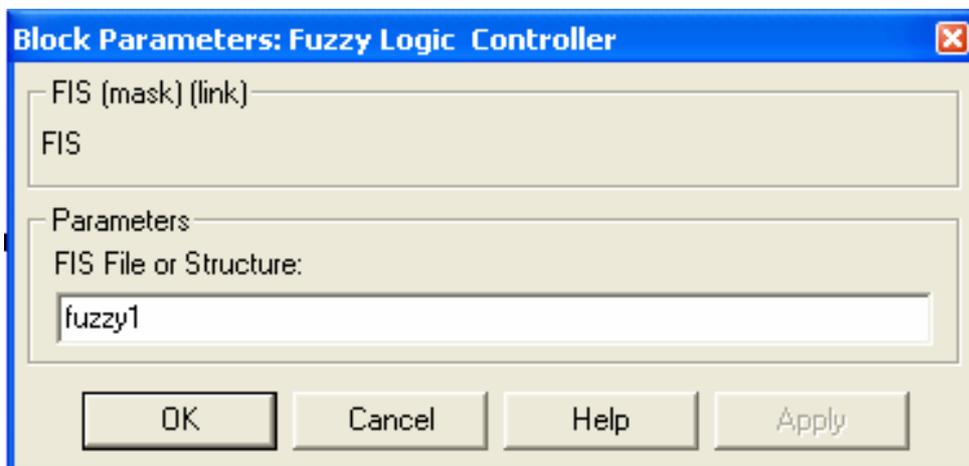


Рис. 19. Завдання параметра моделі

12. У вікні моделі в меню File вибрати пункт Model Properties. У вікні, що відкрилося, вибрати вкладку CallBacks і в полі Model pre-load function написати:

```
fuzzy1=readfis('fuzzy1')
```

Натиснути Ok.

Дана команда кожного разу при відкритті файлу моделі поміщатиме файл fuzzy1.fis у Workspace (робочий простір системи Matlab). Це необхідно для нормального функціонування моделі. Варто відмітити, що при внесенні змін до fis-файлу потрібно поміщати його виправлену версію в Workspace або за допомогою пункту Export>To Workspace меню File, або комбінацією клавіш Ctrl+T, або кожного разу закриттям і відкриттям файлу моделі.

13. У меню Simulation в діалоговому вікні Simulation Parameters у вкладці Advanced для опції Boolean logic signals необхідно встановити значення off (рис. 20). При цьому блоки логіки допускать змінні у формі з плаваючою крапкою.

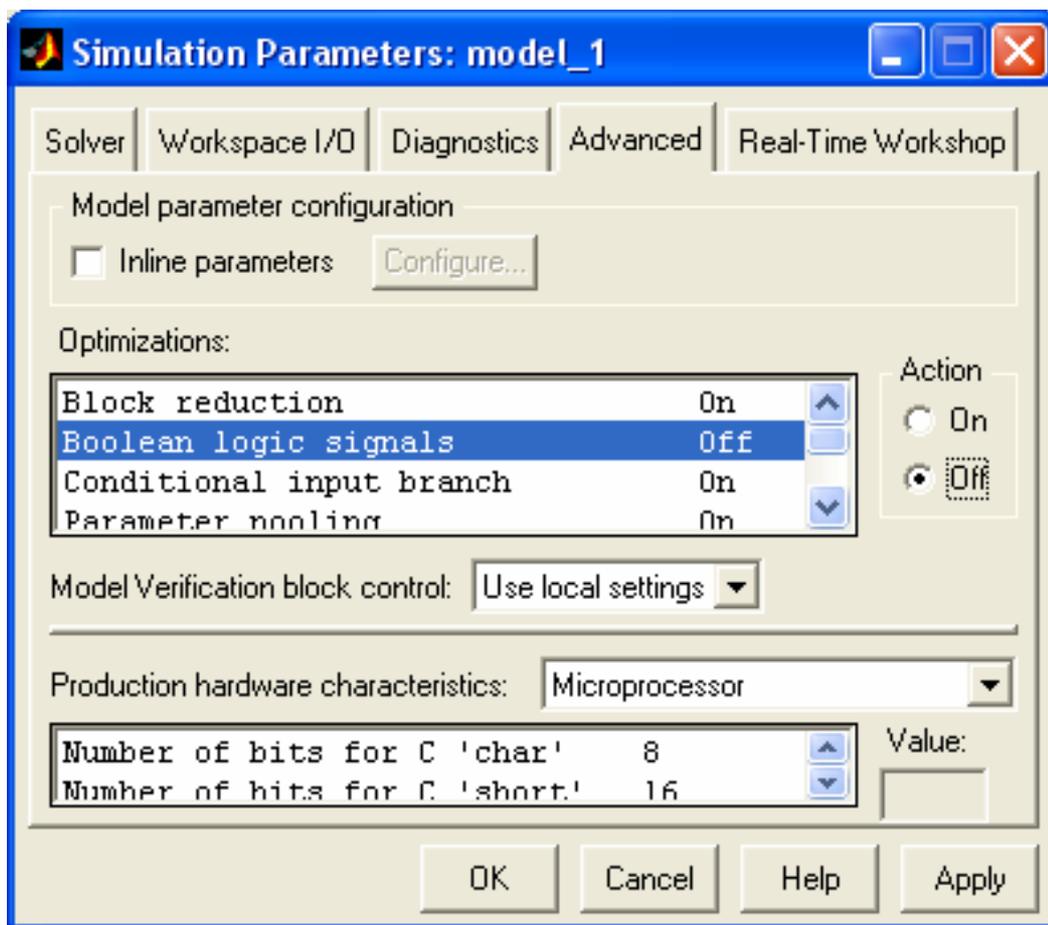


Рис. 20. Налаштування параметрів моделі

14. Завантажити файл з правилами нечіткого виводу за допомогою команди: File>Import From Disk>fuzzy1.fis.

Виконати запуск моделі – Simulation/Start (Ctrl+T).

15. Роботу розробленого fuzzy-регулятора можна порівняти з аналоговою системою регулювання. Для цього модель з fuzzy-регулятором можна доповнити блоком PID з розділу Simulink Extras>група Additional Linear>блок PID Controller.

Для дослідження ПІ-регулятора в блоці налаштувань регулятора необхідно в полі параметра T_d (налаштування диференціюючої частини ПІД-регулятора) встановити значення 0 (рис. 21).

16. Вставити в модель ручний перемикач Manual Switch з групи Signal Routing розділу Simulink. В результаті модель набере наступного вигляду (рис. 22).

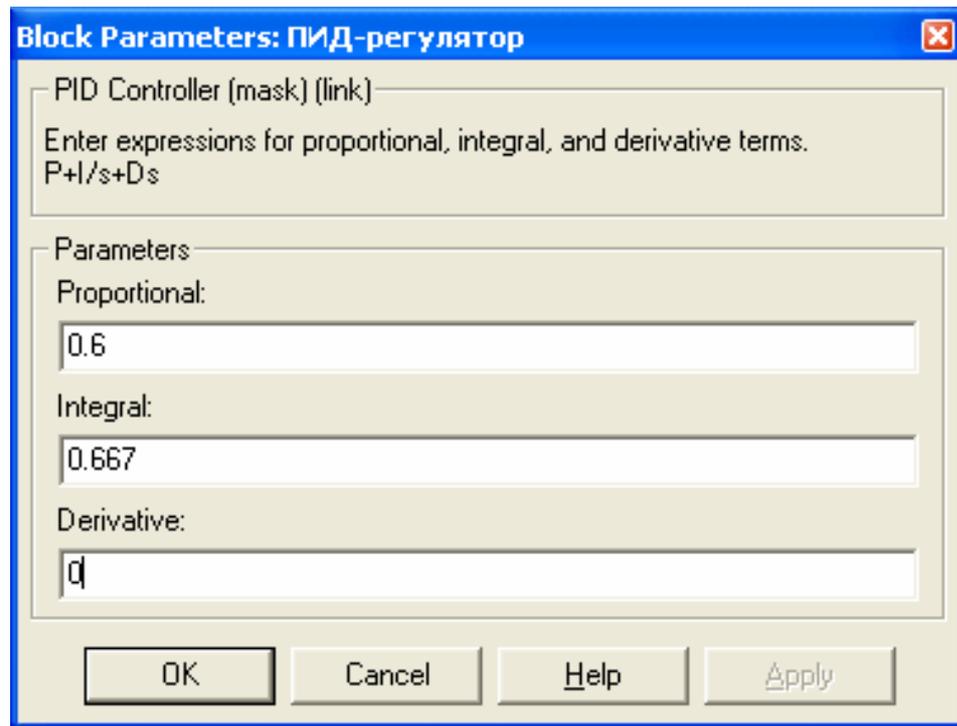


Рис. 21. Параметры ПИ-регулятора

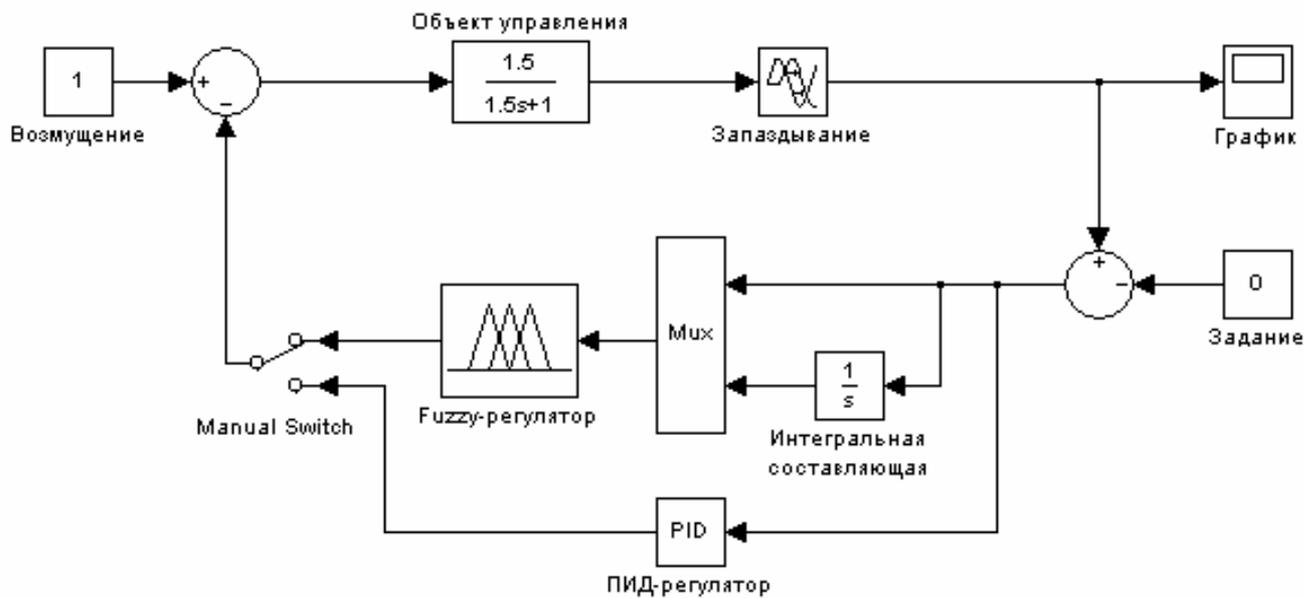


Рис. 22. Модель АСР з аналоговим ПІ-регулятором і fuzzy-регулятором

17. Графік перехідних процесів для моделі з аналоговим ПІ-регулятором і моделі з ПІ-подібним fuzzy-регулятором представлені на рисунках 23 і 24.

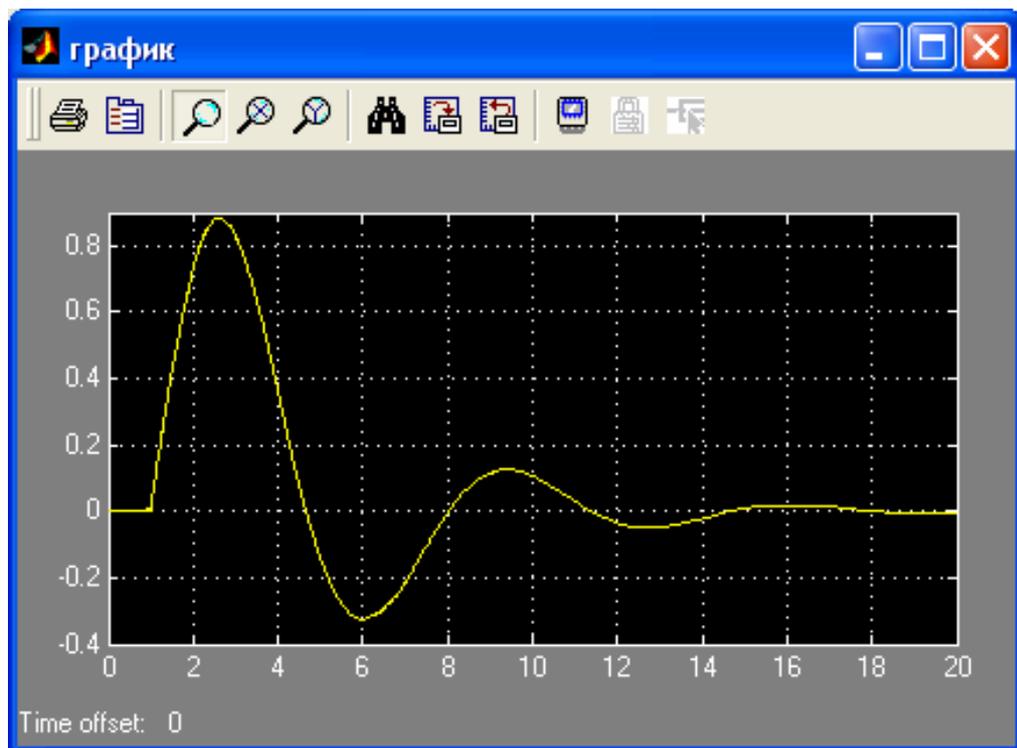


Рис. 23. Графік перехідного процесу для моделі з аналоговим ПІ-регулятором

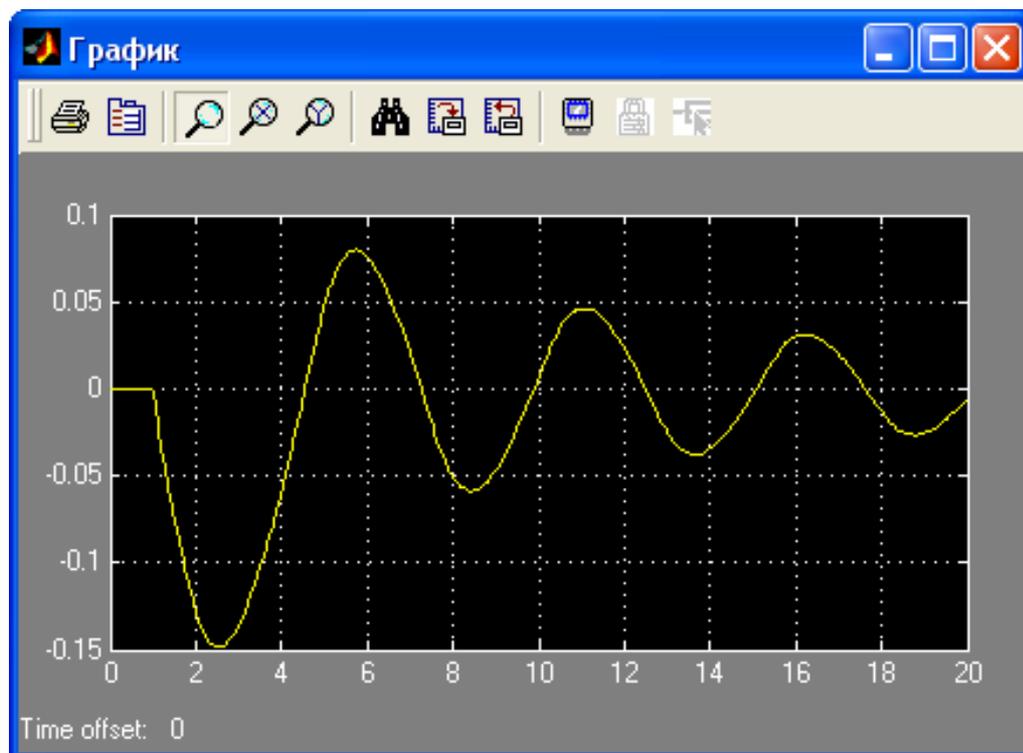


Рис. 24. Графік перехідного процесу для моделі з ПІ-подібним fuzzy-регулятором

Як бачимо з рисунків час регулювання і число коливань більше для системи з fuzzy-регулятором, проте динамічна помилка значно зменшилася.

Варіанти завдання

Дослідити АСР з аналоговим ПІ-регулятором і fuzzy-регулятором. Як модель об'єкту прийняти аперіодичну ланку, яка описується передавальною функцією:

$$W(s) = \frac{K}{Ts+1} e^{-\tau \cdot s}$$

Параметри об'єкту взяти з таблиці 4, для всіх варіантів $\tau=1$ с.

Таблиця 4

Параметри об'єкту

Варіант	К	Т	Варіант	К	Т
1	1.5	1.8	11	3.5	1.8
2	1.8	1.8	12	3.8	1.5
3	1.3	1.5	13	3.3	1.5
4	1.9	1.5	14	3.9	1.5
5	2	1.3	15	3	1.8
6	2.2	1.5	16	3.2	1.2
7	1.4	1.5	17	3.4	1.4
8	2.5	1.6	18	4	1.9
9	2.3	1.4	19	2.6	1.6
10	2.4	1.7	20	2.8	1.7

Оптимальні налаштування пропорційної (K_1) і інтегруючої (K_0) частин ПІ-регулятора визначити по формульному методу для аперіодичного процесу (таблиця. 5).

Параметри налаштування регуляторів

Регулятор	Аперіодичний процес	Процес з перерегулюванням 20 %	Процес з мінімальним часом регулювання
ПІ	$K_1 = \frac{0,6 \cdot T}{K \cdot \tau}$ $K_0 = \frac{1}{K \cdot \tau}$	$K_1 = \frac{0,7 \cdot T}{K \cdot \tau}$ $K_0 = \frac{1}{K \cdot \tau}$	$K_1 = \frac{T}{K \cdot \tau}$ $K_0 = \frac{1}{K \cdot \tau}$

Контрольні питання

1. Структурна схема системи керування з нечітким регулятором.
2. Переваги нечітких логічних регуляторів.
3. Що таке нечіткий алгоритм?
4. У чому полягає процедура синтезу нечітких регуляторів?
5. Експертні системи на основі використання нечіткої логіки.
6. Гібридні мережі, їхнє застосування.
7. Що таке нечіткий регулятор Такагі-Сугено?

Лабораторна робота № 5

Тема: ДОСЛІДЖЕННЯ АВТОМАТИЧНОЇ СИСТЕМИ УПРАВЛІННЯ З FUZZY-РЕГУЛЯТОРОМ ПІД-ТИПУ

Мета роботи: закріплення навичок роботи Fuzzy Logic з блоками Simulink.

Постановка завдання: дослідження системи автоматичного регулювання з ПІД-подібним нечітким регулятором.

Короткі теоретичні відомості

Кількість і вигляд вхідних і вихідних сигналів нечітких логічних регуляторів (НЛР) ПІД-типу залежить від вирішуваного завдання, проте найчастіше входами НЛР служать (у різних комбінаціях) помилка управління, її похідна і інтеграл (рис. 1).

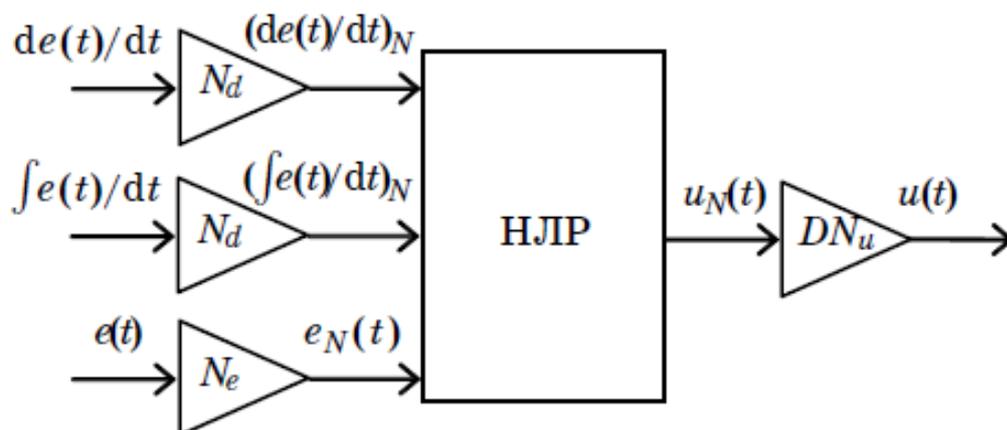


Рис. 1. НЛР ПІД-типу (НЛР_ПІД)

Для спрощення завдання проектування НЛР_ПІД використовуються різні варіанти запису рівняння (таблиця. 1).

Потенційна кількість правил для різних типів НЛР

Кількість термов	Кількість правил		
	НЛР_П	НЛР_ПД (ПИ)	НЛР_ПІД
3	3	9	27
5	5	25	125
7	7	49	343

Запишемо рівняння для ПІД-регулятора у вигляді:

$$u(t) = \left(\frac{1}{2} k_p e(t) + k_d \frac{de(t)}{dt} \right) + \left(\frac{1}{2} k_p e(t) + k_i \int e(t) dt \right).$$

Перша скобка описує ПД-регулятор, друга – ПІ-регулятор, тобто ПІД-регулятор можна представити у вигляді паралельного з'єднання цих регуляторів. Відповідно, НЛР_ПІД можна представити у вигляді структури, показаної на рис. 2.

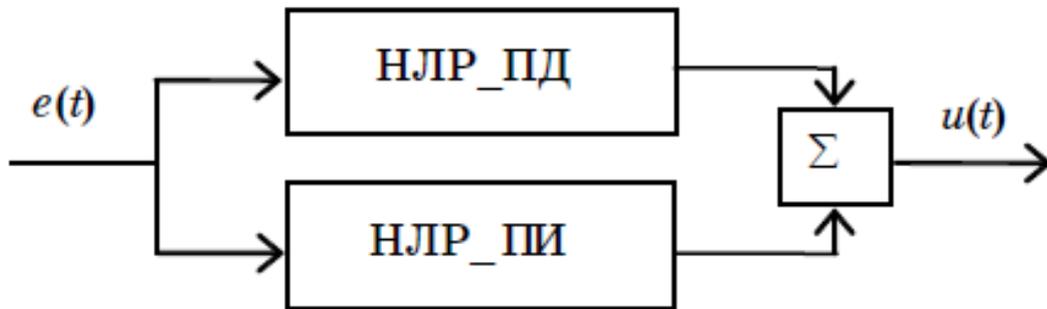


Рис. 2. Спрощення структури НЛР_ПІД

Інший варіант реалізації ґрунтується на уявленні (рис. 1) у вигляді:

$$u(t) = k_p \left(e(t) + \frac{k_d}{k_p} \frac{de(t)}{dt} \right) + k_i \int e(t) dt.$$

Тут перша скобка описує ПД-регулятор, що відповідає за час наростання і перерегулювання, а друга скобка – І-регулятор, який служить для зменшення статичної помилки. Таким чином, ПД-регулятор повинен передавати управління І-регулятору, коли $e(t)$ і $de(t)/dt$ стають малими (рис. 3).

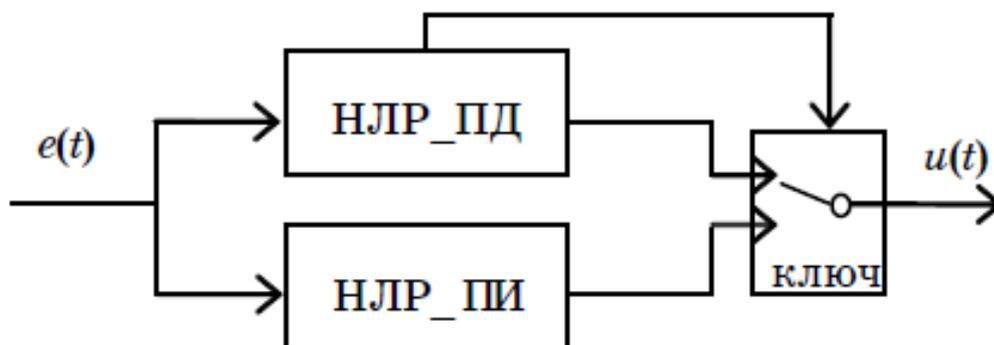


Рис. 3. Варіант структури НЛР_ПД

Завдання. Дослідити АСР з аналоговим ПД-регулятором і ПД-подібним fuzzy-регулятором. Як модель об'єкту прийняти аперіодичну ланку першого порядку із запізнюванням. Передавальна функція об'єкту управління має вигляд ($\tau=1$ с):

$$W_o(s) = \frac{1.5}{1.5 \cdot s + 1} \cdot e^{-s}$$

Порядок виконання роботи

1. У пакеті Simulink застосування Matlab відкрити модель model_1.mdl, яка була створена у лабораторній роботі №4. Зберегти її як model_2.mdl.

Для реалізації ПД-подібного fuzzy-регулятора необхідно в структурну схему моделі додати диференціальну складову сигналу неузгодження, використовуючи блок диференціювання Derivative з групи Continuous розділу Simulink (рис. 4).

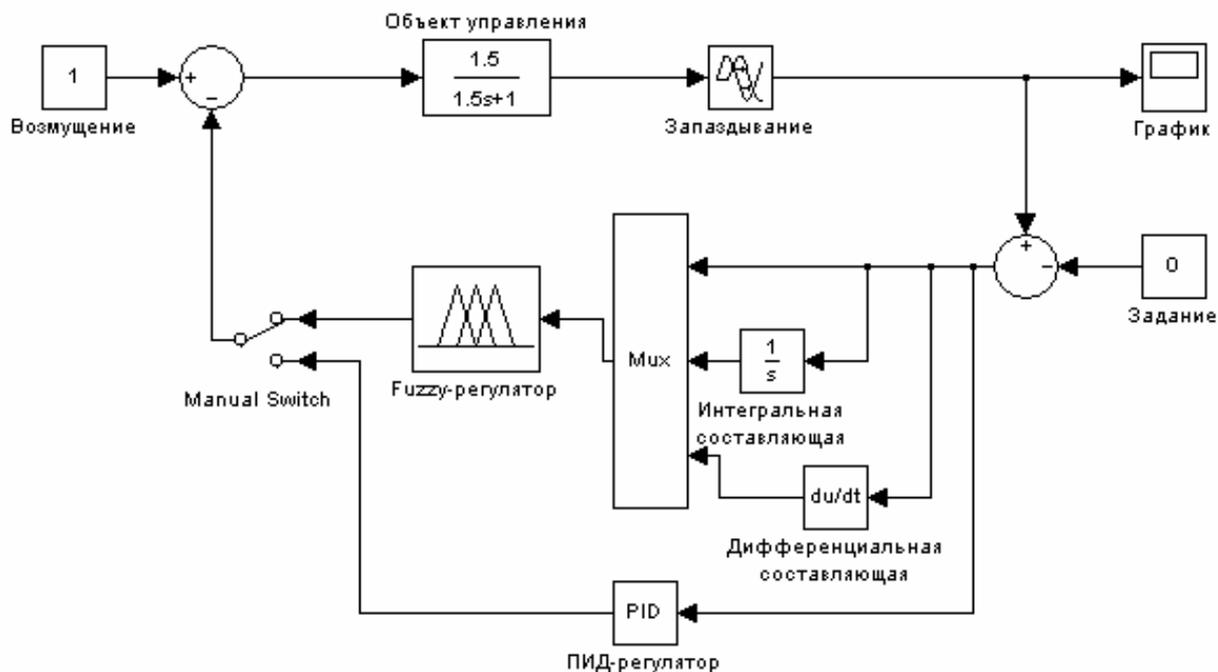


Рис. 4. Модель одноконтурної системи автоматичного регулювання з ПІД-подібним fuzzy-регулятором

2. Встановити значення параметрів налаштування ПІД-регулятора для даного об'єкту: $K_1=0.95$, $K_0=0.4$, $K_2=T_D=0.38$ (рис. 5).

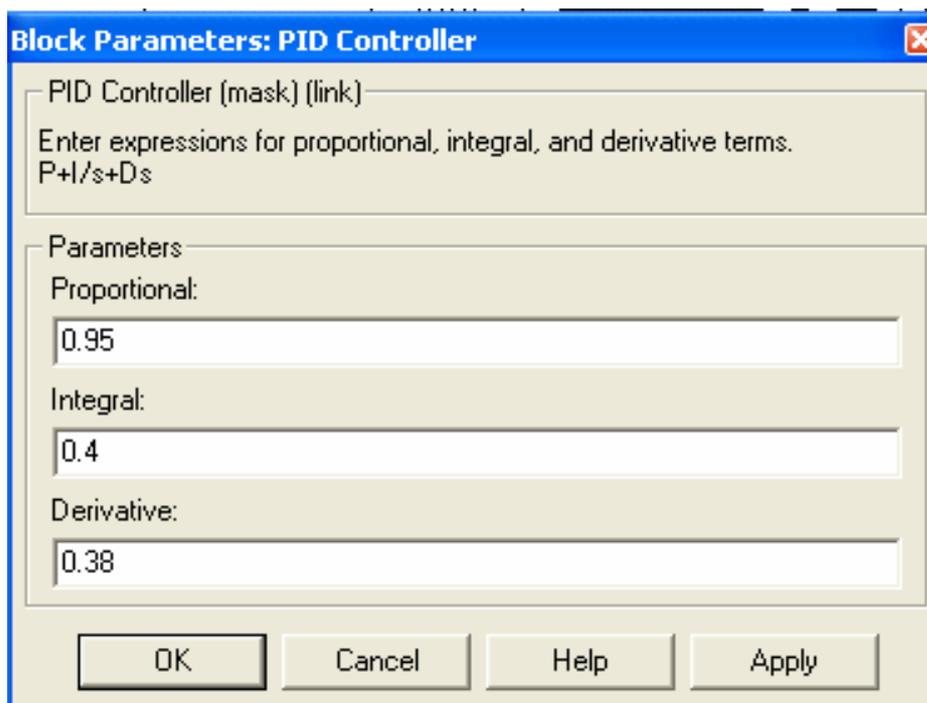


Рис. 5. Параметри ПІД-регулятора

Диференціальне рівняння ПД-регулятора має вигляд:

$$y(t) = k_D \cdot \varepsilon(t) + \frac{1}{T_E} \int_0^t \varepsilon(t) \cdot dt + T_D \cdot \frac{d\varepsilon(t)}{dt}.$$

3. За основу fis-файла, що реалізовує ПД-закон регулювання, може бути узятий файл fuzzy1.fis. Необхідно у вікні редактора фазі-інференційної системи додати ще одну вхідну змінну x3, для якої у вікні редактора функцій приналежності задати три трикутні терми – N, ZE і P.

4. Діапазони зміни змінних визначаються згідно варіанту завдання в таблиці

3. Для даного прикладу:

- $x1=[-1.053;1.053]$;
- $x2=[-2.5;2.5]$;
- $x3=[-1;1]$ (спочатку);
- $y=[0;2]$.

Діалогові вікна з налаштуванням діапазону значень для вхідних змінних показані на рис. 6-8.

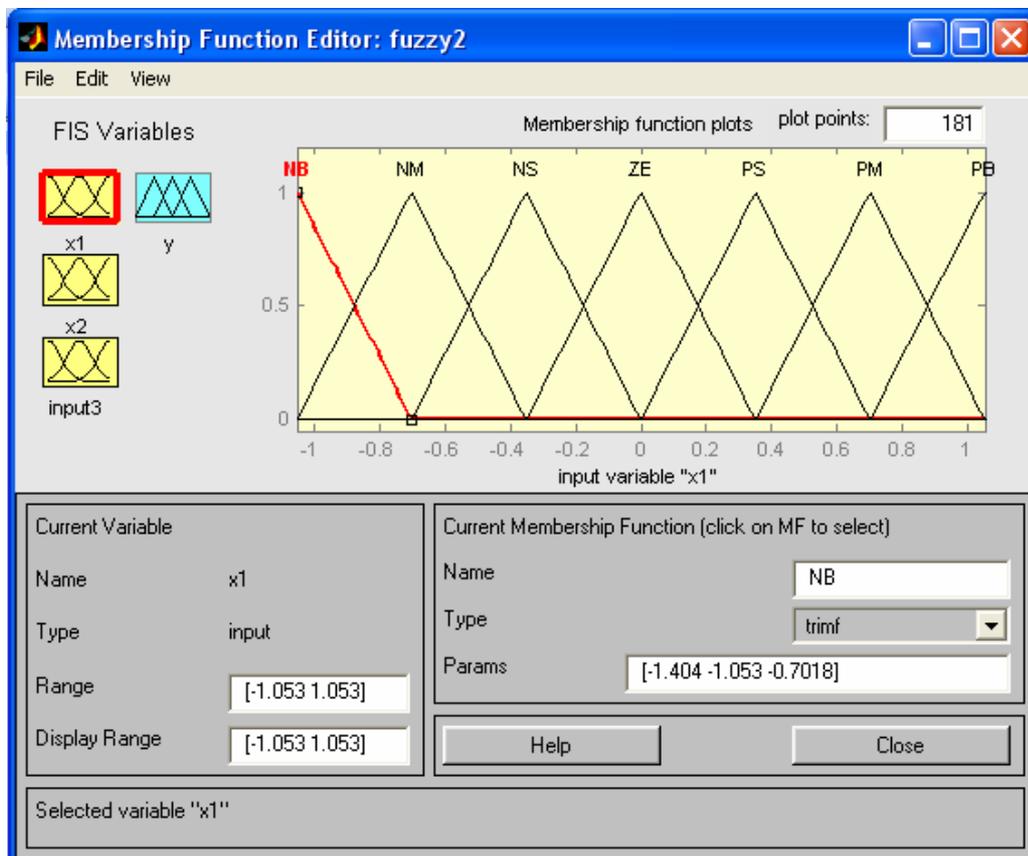


Рис. 6. Вікно редактора функцій приналежності змінної x1

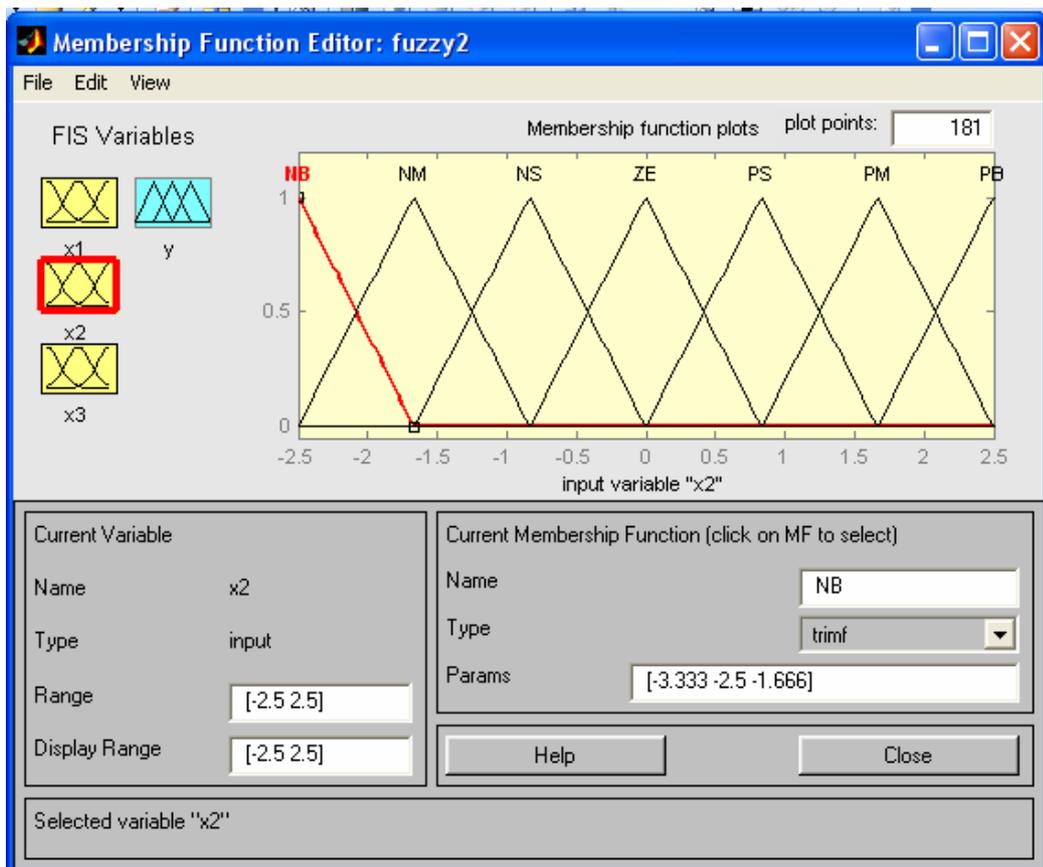


Рис. 7. Вікно редактора функцій приналежності змінної x2

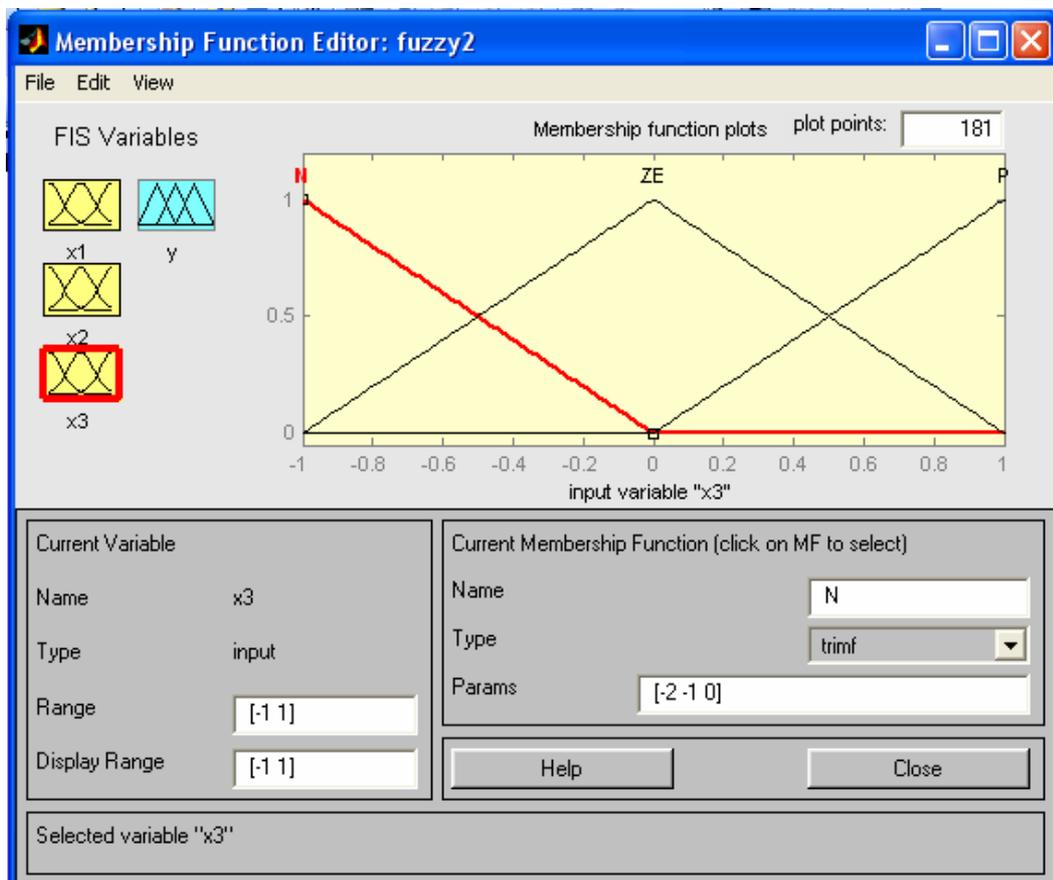


Рис. 8. Вікно редактора функцій приналежності змінної x3

5. Для вихідної змінної додати два терми N і P (рис. 9).

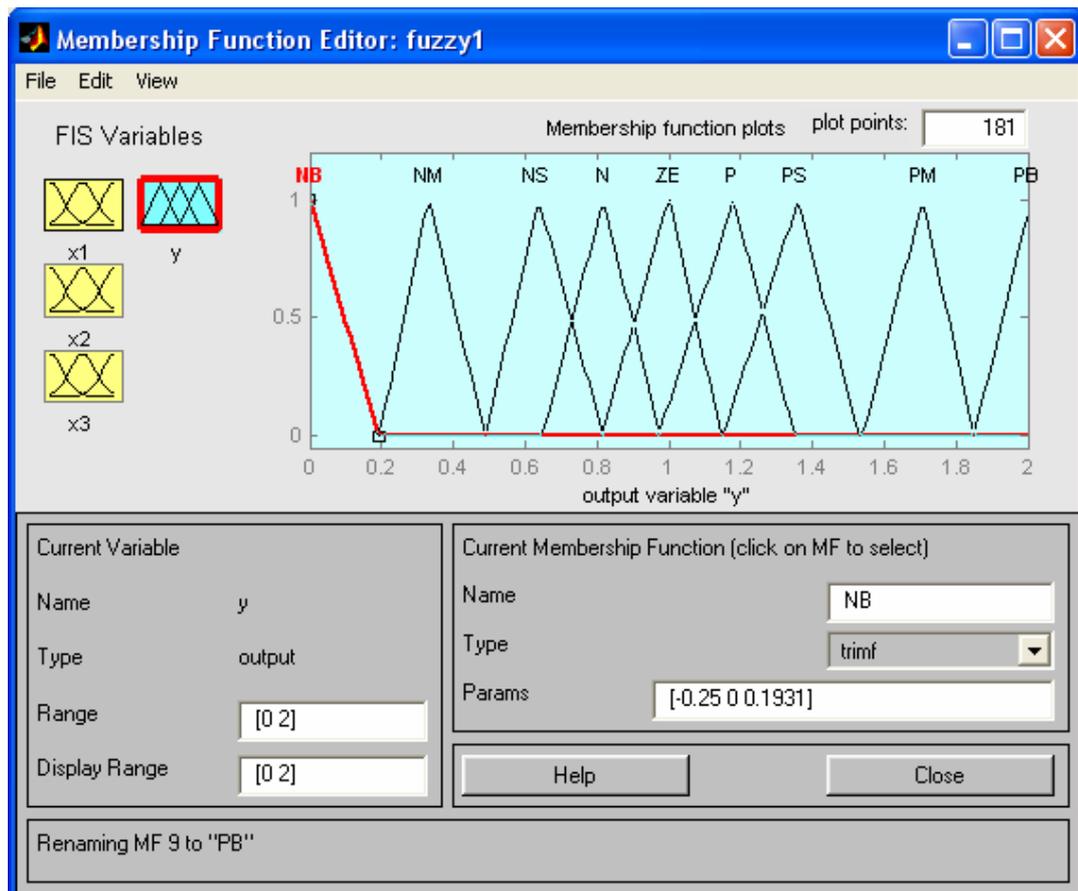


Рис. 9. Терми вихідної змінної ПІД-подібного fuzzy-регулятора

6. Викликати вікно редактора правил і доповнити список правил наступними трьома правилами:

- IF (x3=N) THEN (y=N)
- IF (x3=ZE) THEN (y=ZE)
- IF (x3=P) THEN (y=P)

Дійсно, якщо швидкість позитивна, то вихідний сигнал регулятора повинен збільшитися, аби попередити збільшення сигналу неузгодження, і, навпаки.

7. Зберегти відредагований файл як fuzzy2.fis.

8. У вікні параметрів блоку Fuzzy Logic Controller моделі вказати ім'я файлу fuzzy2. У вікні моделі в меню File вибрати пункт Model Properties.

У вікні, що відкрилося, вибрати вкладку CallBacks і в полі Model pre-load function написати:

```
fuzzy2=readfis('fuzzy2')
```

У діалоговому вікні Simulation Parameters меню Simulation у вкладці Advanced для опції Boolean logic signals необхідно встановити значення off.

9. Точніше налаштування діапазону зміни Д-складової здійснюється методом підбору. Наприклад, для даного об'єкту [-0.16;0.16].

Графіки перехідних процесів для моделі з аналоговим регулятором і моделі з ПД-подібним fuzzy-регулятором представлені на рисунку 10.

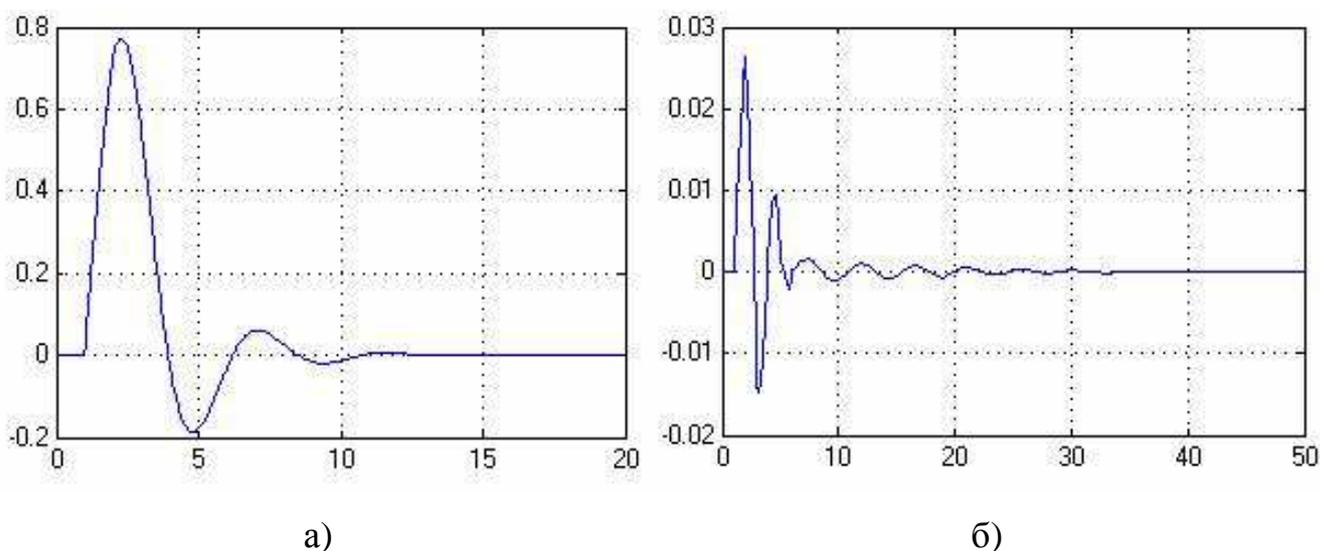


Рис. 10. Графіки перехідних процесів для моделі з аналоговим регулятором (а) і моделі з ПД-подібним fuzzy-регулятором (б)

З рисунку 10 бачимо, що час регулювання і число коливань більше для системи з fuzzy-регулятором, проте динамічна помилка значно зменшилася.

На практиці найбільше використовується ПІ-регулятор, оскільки Д-складову потрібно точно налаштувати, інакше в системі виникнуть коливання.

Варіанти завдання

Дослідити АСР з аналоговим ПІД-регулятором і fuzzy-регулятором. Як модель об'єкту прийняти аперіодичну ланку, яка описується передавальною функцією:

$$W(s) = \frac{K}{Ts+1} e^{-\tau \cdot s}$$

Параметри об'єкту узяти з таблиці 2, для всіх варіантів $\tau = 1$ с.

Таблиця 2

Параметри об'єкту

Варіант	К	Т	Варіант	К	Т
1	1.5	1.8	11	3.5	1.8
2	1.8	1.8	12	3.8	1.5
3	1.3	1.5	13	3.3	1.5
4	1.9	1.5	14	3.9	1.5
5	2	1.3	15	3	1.8
6	2.2	1.5	16	3.2	1.2
7	1.4	1.5	17	3.4	1.4
8	2.5	1.6	18	4	1.9
9	2.3	1.4	19	2.6	1.6
10	2.4	1.7	20	2.8	1.7

Налаштування ПІД-регулятора визначити по формульному методу (таблиця.

3). При цьому:

$$x1 \in \left[-\frac{1}{K_1}; \frac{1}{K_1} \right], \quad x2 \in \left[-\frac{1}{K_0}; \frac{1}{K_0} \right], \quad x3 \in \left[-\frac{1}{K_2}; \frac{1}{K_2} \right],$$

де K_1 , K_0 , K_2 – оптимальні налаштування пропорційної, інтегруючої і диференціюючої частин ПІД-регулятора відносно якогось критерію.

Таблиця 3

Параметри налаштування регуляторів

Регулятор	Аперіодичний процес	Процес з перерегулюванням 20 %	Процес з мінімальним часом регулювання
ПІД	$K_1 = \frac{0,95 \cdot T}{K \cdot \tau}$ $K_0 = \frac{0,4 \cdot T}{K \cdot \tau^2}$ $K_2 = \frac{0,38 \cdot T}{K}$	$K_1 = \frac{1,2 \cdot T}{K \cdot \tau}$ $K_0 = \frac{0,6 \cdot T}{K \cdot \tau^2}$ $K_2 = \frac{0,48 \cdot T}{K}$	$K_1 = \frac{1,4 \cdot T}{K \cdot \tau}$ $K_0 = \frac{1,08 \cdot T}{K \cdot \tau^2}$ $K_2 = \frac{0,7 \cdot T}{K}$

Контрольні питання

1. У чому полягає ідея нечіткого управління?
2. Структурна схема системи нечіткого управління.
3. Основні операції над нечіткою множиною.
4. Особливості побудови адаптивної САУ з еталонною моделлю на основі нечіткої логіки.
5. У чому відмінність методу Максимуму-Мінімуму від методу Максимуму-Добутку?
6. Чим відрізняється алгоритм нечіткого виведення Мамдані від алгоритму Сугено?

Лабораторна робота № 6

Тема: ОПТИМІЗАЦІЯ БАГАТОВИМІРНИХ ФУНКЦІЙ З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

Мета роботи: вивчення особливостей використання генетичного алгоритму для знаходження мінімуму функцій з багатьма екстремумами.

Постановка завдання: пошук мінімуму двовимірної функції за допомогою генетичного алгоритму.

Короткі теоретичні відомості

Під оптимізацією функції слід розуміти знаходження екстремального значення якої-небудь залежності на всьому робочому діапазоні.

На відміну від точних методів математичного програмування еволюційні методи дозволяють знаходити рішення, близькі до оптимальних, за прийнятний час. На відміну від відомих евристичних методів оптимізації еволюційні методи характеризуються істотно меншою залежністю від особливостей застосування (тобто більш універсальні) і, в більшості випадків, забезпечують кращу міру наближення до оптимального рішення.

Найважливішим окремим випадком еволюційних методів є генетичні методи, засновані на пошуку кращих рішень за допомогою успадкування і посилення корисних властивостей множини об'єктів певного застосування в процесі імітації їх еволюції. Загальне завдання безперервної оптимізації може представлятися як:

$$\min_{x \in D} f(x), \text{ де } D = \{x = (x_1, x_2, \dots, x_N) \mid x_i \text{ на } [a_i, b_i], i=1,2,\dots,N\}, \quad (1)$$

де $f(x)$ - скалярна багатопараметрична (цільова) функція, що мінімізується, яка може мати декілька глобальних екстремумів; прямокутна область D – зона пошуку.

Мається на увазі, що про функцію $f(x)$ відомо лише те, що вона визначена в будь-якій точці області D . Жодна додаткова інформація про характер функції і її властивості (диференційованість, безперервність і так далі) не враховується в процесі пошуку.

Під рішенням задачі (1) розумітимемо вектор $x = (x_1, x_2, \dots, x_N)$. Оптимальним рішенням задачі (1) вважатимемо вектор x^* , при якому цільова функція $f(x)$ набуває максимального значення. Виходячи з припущення про можливу багатоекстремальність $f(x)$, оптимальне рішення може бути не єдиним.

Стандартні генетичні алгоритми, спочатку запропоновані як ефективний засіб для вирішення різних оптимізаційних завдань, будуються згідно узагальненій структурі, яка показана на рис. 1, і описуються набором наступних основних параметрів:

$$GA = (P^T, d, l, s, r, f, t),$$

де $P^T = (a_1^T, \dots, a_d^T)$ – деяка підмножина потенційних рішень, яка називається популяцією хромосом;

$T = (0, 1, 2, \dots)$ – номер популяції;

d – розмір популяції;

a_i^T – хромосома, що є послідовністю генів, значення яких визначають закодований вигляд потенційного рішення деякої поставленої задачі;

l – довжина хромосоми, що визначає число генів кожної особи;

s – оператор відбору;

r – оператор рекомбінації хромосом;

f – функція оцінки міри корисності (пристосованості) хромосом;

t – критерій зупинки.



Рис. 1. Узагальнена структура стандартного генетического алгоритму

Практичне застосування генетичних алгоритмів до завдань оптимізації передбачає необхідність кодування простору пошуку рішень. Вибраний спосіб кодування визначатиме структуру хромосоми, як набір генів, що є деякою бінарною послідовністю. При цьому, унікальна кодова комбінація, записана в кожній окремій хромосомі, може розглядатися як потенційне рішення відповідної оптимізаційної задачі.

Методика виконання роботи

Методику виконання роботи покажемо на прикладі оптимізації функції, для якої x змінюється в діапазоні $[-10;25]$:

$$f(x) = \begin{cases} 3 \sin(8x) - \exp(-(\frac{x}{20})^2), & \text{для } x \leq 20 \\ -\exp(-1) + (x-20)(x-22), & \text{для } x > 20 \end{cases}$$

Для візуального відображення вигляду функції $f(x)$ необхідно створити код в m -файле. Для цього потрібно виконати команду Home>New>Script і у вікні, що відкрилося, створити наступний код:

% Код 1

```

y = [];
x = -10:0.2:25; NT = length(x);
for t = 1:NT
if x(t)<20
y(t) = 3*sin(2*x(t))-exp(-(x(t)/20).^2);
else
y(t) = -exp(-1)+(x(t)-20)*(x(t)-22);
end
end
plot(x,y)

```

Запустити код на виконання за допомогою команди EDITOR>Run.

На отриманому графіку (рис. 2) можна візуально оцінити точку глобального мінімуму і зафіксувати відповідне значення координати по x.

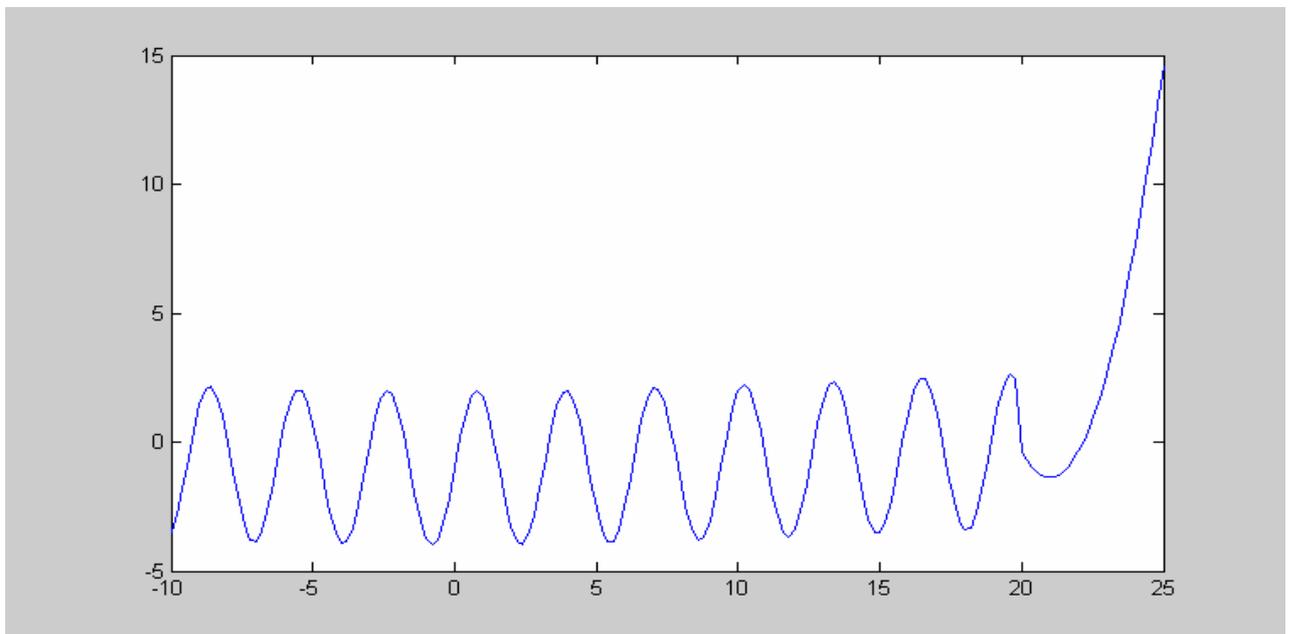


Рис. 2. Вигляд функції $f(x)$

Для обчислення точного значення мінімуму можна використати функцію min. Для цього у вікні Command Window потрібно набрати:

```
[ymin,imin] = min(y);
ymin, x(imin)
```

Результат відобразиться в командному рядку:

```
ymin = -3.9971
x = -0.8000.
```

Примітка

Вживання функції $\min(f(x))$ можливо, якщо відомі всі значення функції для всього простору аргументів, тобто при рішенні задачі мінімізації повним перебором.

При використанні оптимізатора GATool необхідно за допомогою команди HOME>New>Function створити функцію (окремий m-файл) з наступним кодом, що обчислює значення функції $f(x)$:

%код 2

```
function y = func_0(x)
if x<20
y = 3*sin(2*x)-exp(-(x/20).^2);
else
y = -exp(-1)+(x-20)*(x-22);
end
```

Функцію потрібно зберегти в робочу (або поточну) теку як **func_0.m**.

Відкрити майстер генетичного алгоритму за допомогою вкладки APPS і у вікні Matlab вибрати Optimization. У полі Solver вибрати: ga – Genetic Algorithm. У вікні майстра генетичного алгоритму, що з'явилося, в рядку «*Fitness function*» задати ім'я функції для оптимізації з наступним префіксом: @**func_0**.

У рядку «*Number of variables*» задати число змінних (x_1, \dots, x_n), за якими здійснюється пошук, в даному випадку - 1. У рядку «**Initial Range**» задати діапазон пошуку [-10; 25]. Після цього можна запустити алгоритм пошуку за допомогою кнопки Start.

Результат виконання оптимізації відображується у вікні Final Point. Для заданого прикладу результат буде $x=-0.793$. Значення функції $y=-3.998051851160963$.

В результаті аналізу знайдених координат за допомогою генетичного алгоритму і координат, знайдених за допомогою функції min (перебором всіх значень) можна зробити висновок, що вони збігаються з точністю до третього знаку після коми.

Завдання. Знайти мінімум двовимірної функції Z, для якої параметри змінюються в діапазоні $X=[-2:2]$ і $Y=[-2:2]$, а N - номер варіанту:

$$Z = 2 * N * \sin(2 * \pi * N * X) * \exp(-X^2 - Y^2).$$

Порядок виконання роботи

1. Записати код для обчислення мінімуму функції по сітці X,Y (повним перебором):

%Код 3

N = 18; % N=номер варіанта;

dx = 0.01;

[X,Y] = meshgrid(-2:dx:2); % задание матриц значений

Z = 2*N*sin(2*N*pi*(X)).* exp(-X.^2 - Y.^2); % задание функции

surf(X,Y,Z) % построение графика

[Zmin] = min(min(Z)) % минимум полным перебором

Зберегти код в поточній папці.

```

Editor - D:\cod3.m
cod3.m x +
1 %Код 3
2 - N = 18; % N=номер варианта;
3 - dx = 0.01;
4 - [X,Y] = meshgrid(-2:dx:2); % задание матриц значений
5 - Z = 2*N*sin(2*N*pi*(X)).* exp(-X.^2 - Y.^2); % задание функции
6 - surf(X,Y,Z) % построение графика
7 - [Zmin] = min(min(Z)) % минимум полным перебором

```

Рис. 3. Створений m-файл з кодом програми

2. Запустити код на виконання за допомогою команди EDITOR>Run. В результаті отримуємо графік функції Z (рис. 4).

У вікні Command Window отримуємо результат – значення мінімуму функції Z:

Zmin =
-35.7533

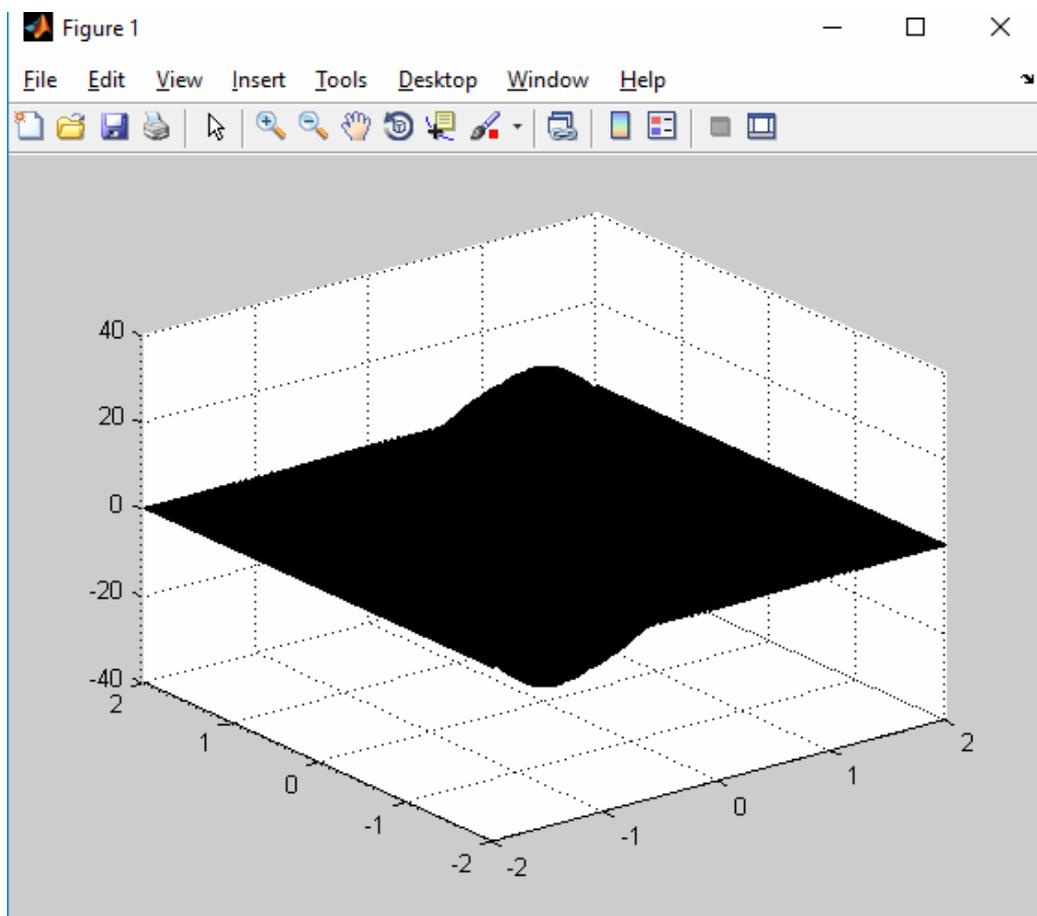


Рис. 4. Тривимірний графік функції Z

3. За допомогою команди HOME>New>Function створити m-файл з функцією розрахунку Z для GATool, де $X=x(1)$, $Y=x(2)$:

% код 4

function Z = func_1(x) % объявление функции

N = 18; % Номер варианта

Z = 2*N*sin(2*N*pi*(x(1))).* exp(-x(1).^2 - x(2).^2); % Расчет функции

Зберегти файл в робочу (або поточну) папку як **func_1.m**.

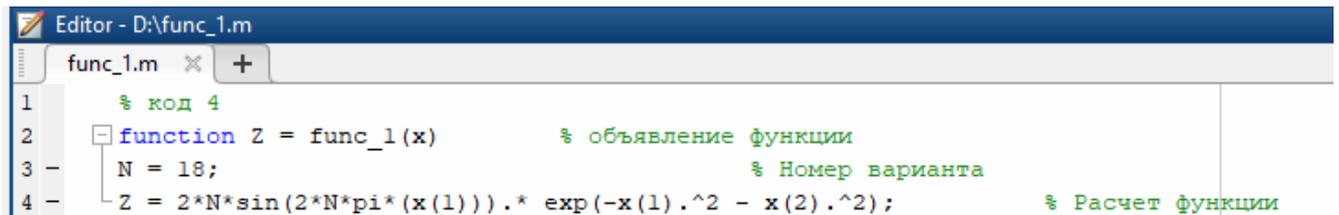


Рис. 5. Створений m-файл з функцією розрахунку Z

4. Відкрити майстер генетичного алгоритму за допомогою вкладки APPS у вікні Matlab і вибрати Optimization. У полі Solver вибрати: ga – Genetic Algorithm (рис. 6).

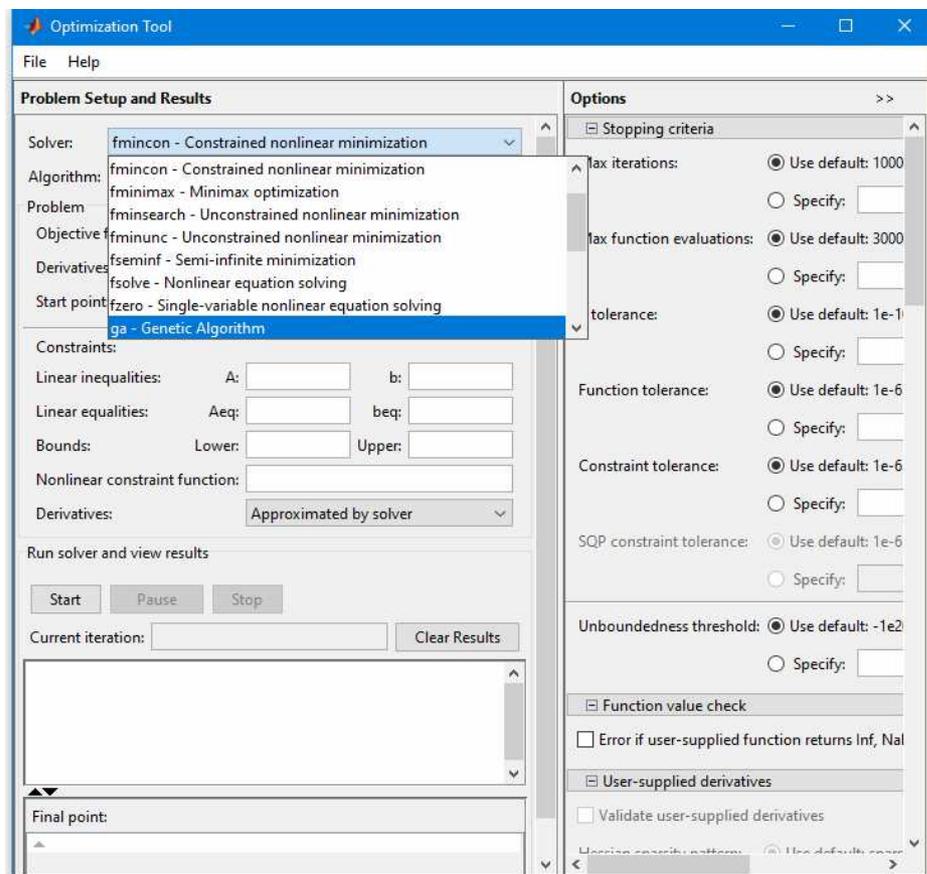


Рис. 6. Вікно майстра генетичного алгоритму

5. У вікні майстра «генетичного алгоритму», що з'явилося, в рядку «*Fitness function*» задати ім'я функції для оптимізації з наступним префіксом: **@func_1**.

У рядку «*Number of variables*» задати число змінних (x_1, x_n), за якими здійснюється пошук, в даному випадку - **2**.

У рядку «*Initial Range*» задати діапазон пошуку [-2 -2; 2 2] (рис. 7).

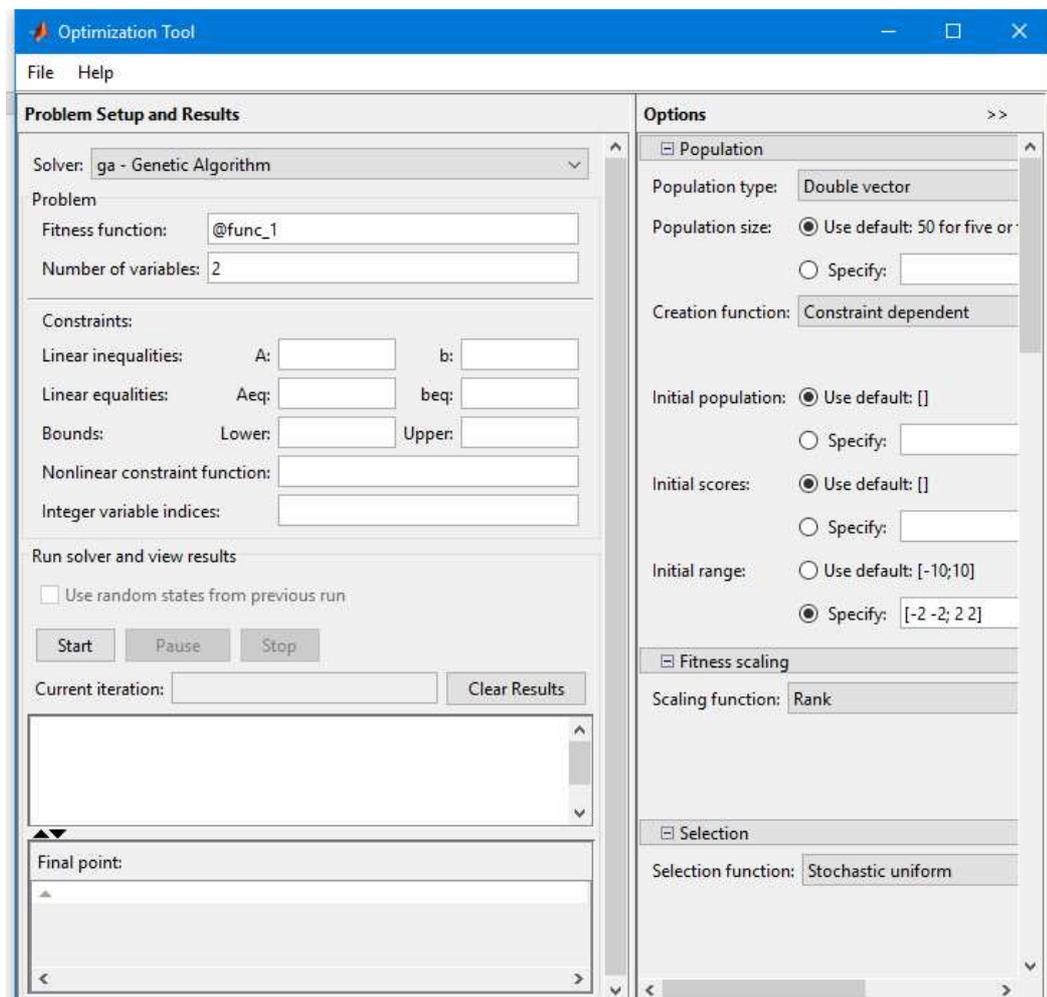


Рис. 7. Завдання параметрів обчислення

6. Запустити алгоритм пошуку за допомогою кнопки Start. В результаті отримаємо мінімум функції Z і значення аргументів функції (рис. 8).

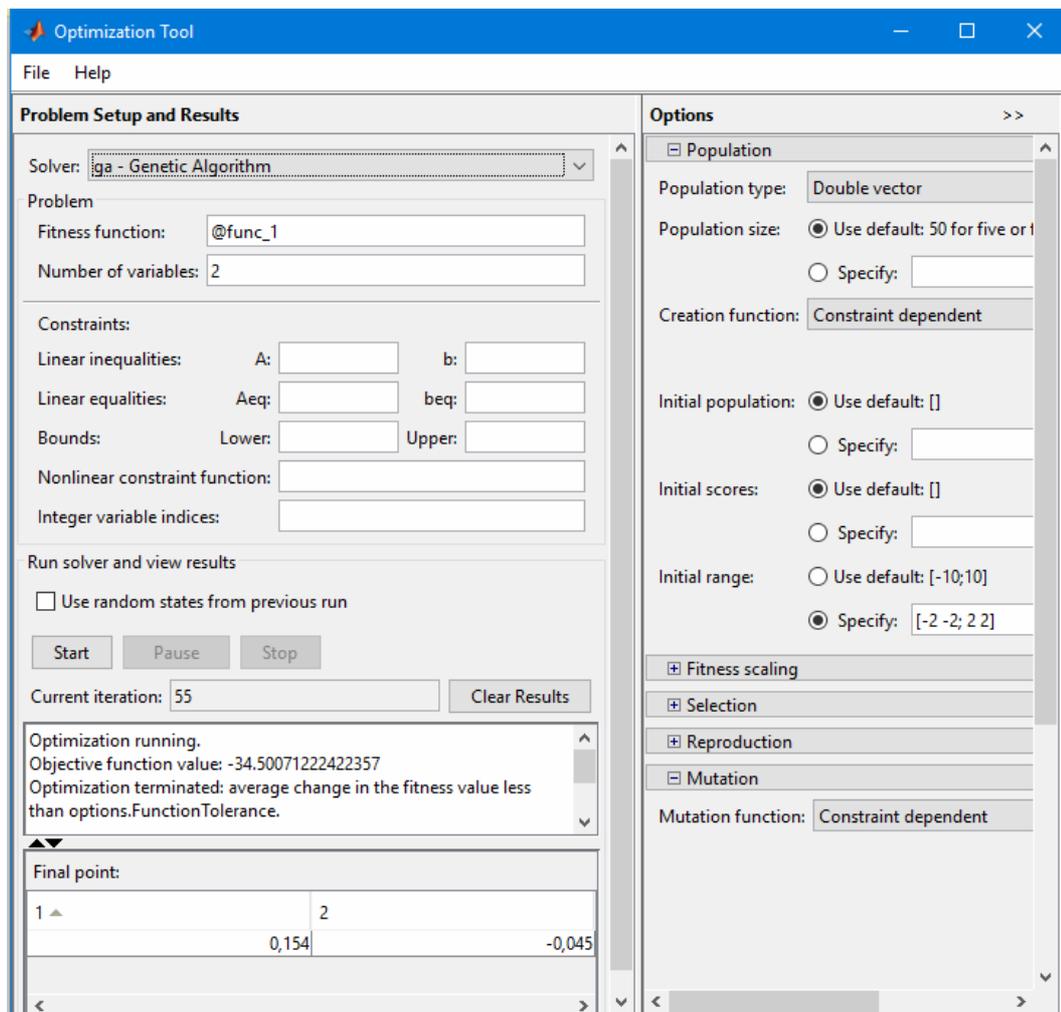


Рис. 8. Результат пошуку рішення у вікні Optimization Tool

8. Повторити пошук три рази (рис. 9-11). Порівняти, знайдені за допомогою генетичного алгоритму, значення мінімуму функції Z із значеннями, знайденими за допомогою функції `min` (перебором всіх значень).

В результаті аналізу знайденого мінімуму функції Z за допомогою генетичного алгоритму і за допомогою функції `min` (перебором всіх значень) можна зробити висновок, що вони збігаються з точністю до другого знаку після коми.

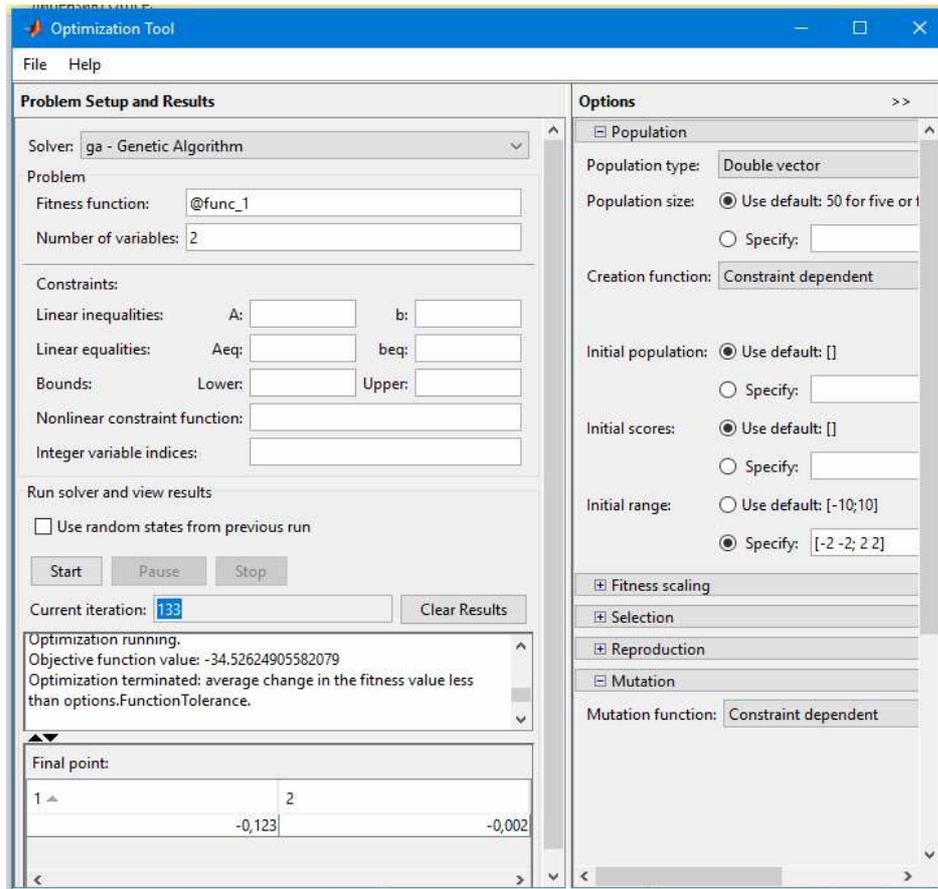


Рис. 9. Другий результат пошуку рішення у вікні Optimization Tool

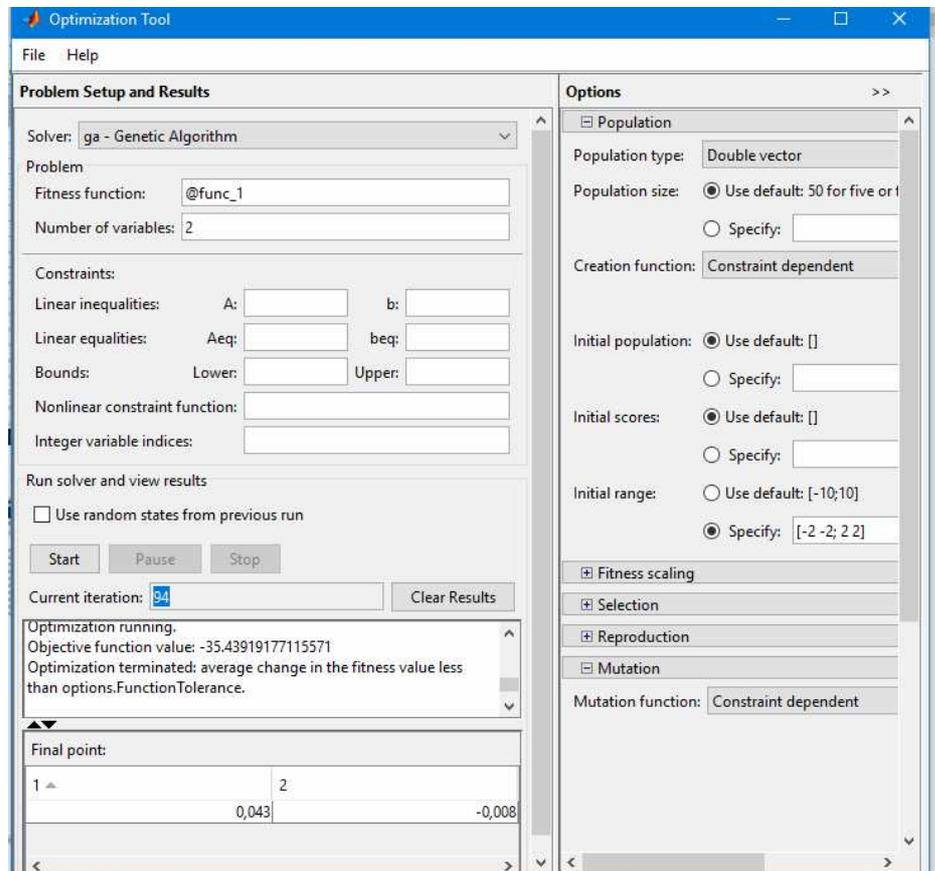


Рис. 10. Третій результат пошуку рішення у вікні Optimization Tool

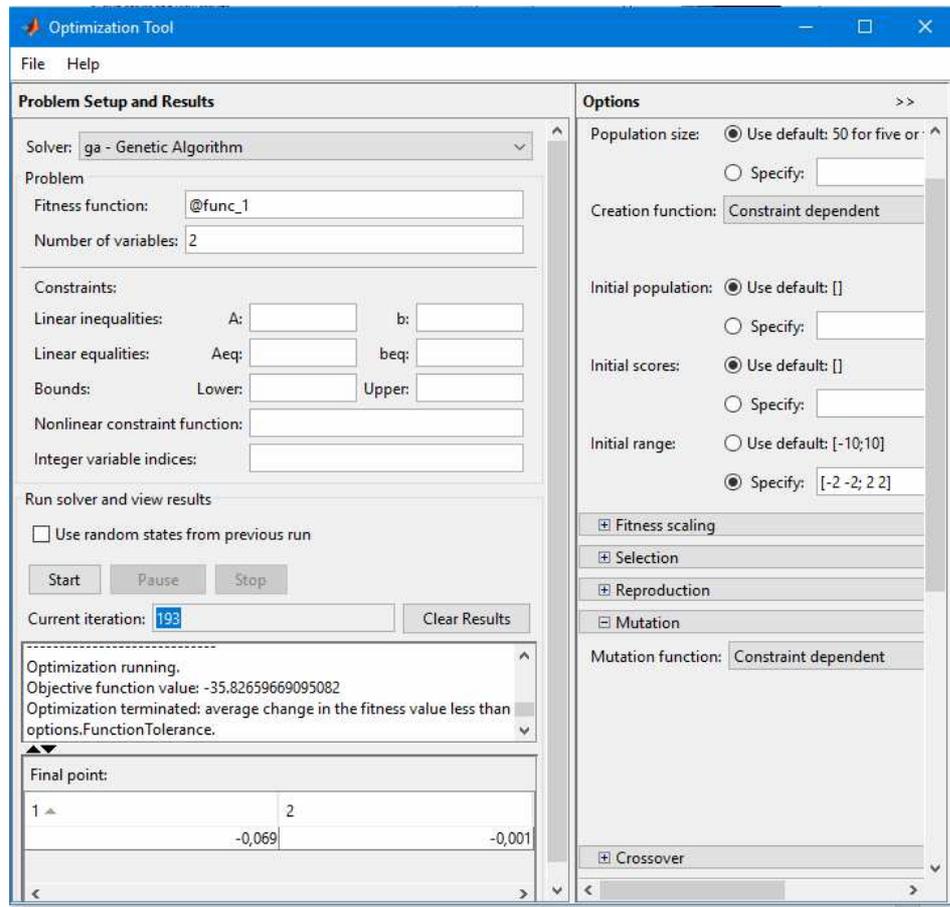
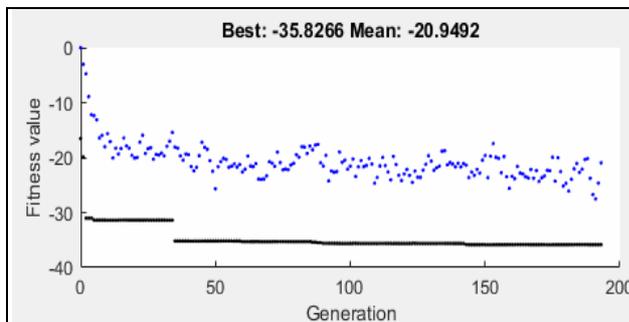
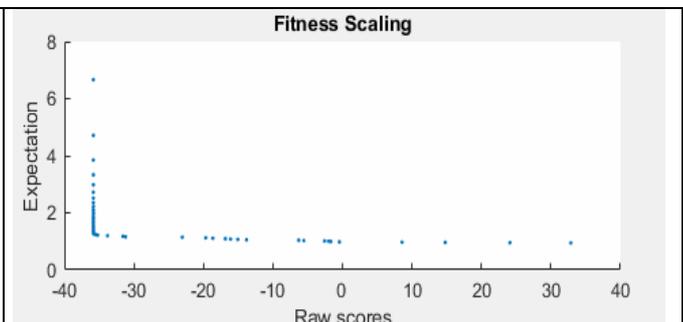


Рис. 11. Четвертий результат пошуку рішення у вікні Optimization Tool

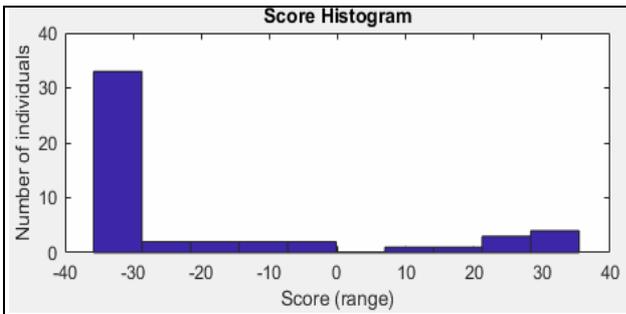
9. Провести візуалізацію результатів роботи генетичного алгоритму за допомогою панелі Plot function. Отримані графіки дозволяють аналізувати і підвищувати ефективність роботи генетичного алгоритму.



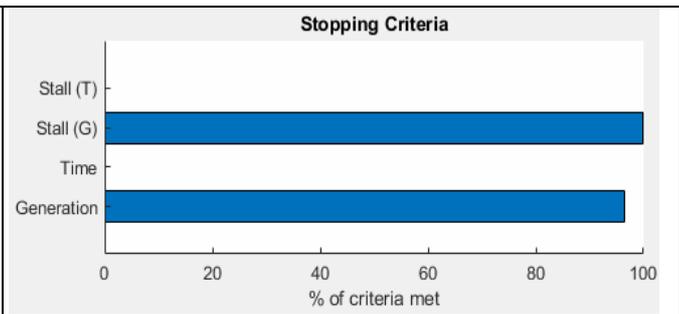
Графіки найкращої і середньої
придатності



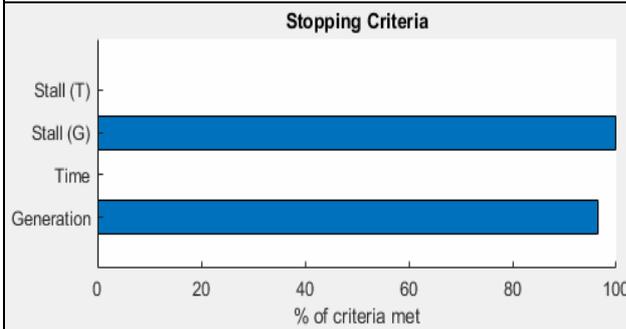
Масштабована придатність



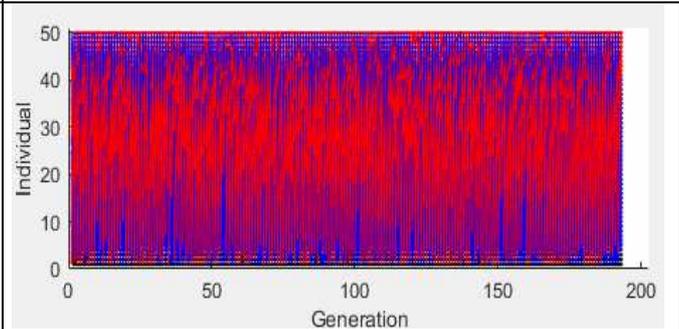
Графік значень придатності хромосом



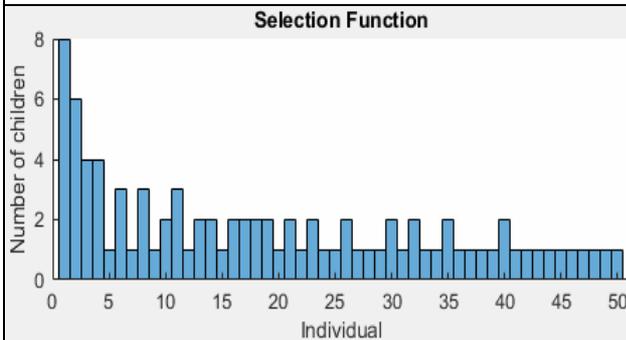
Графік виконання критеріїв зупинки



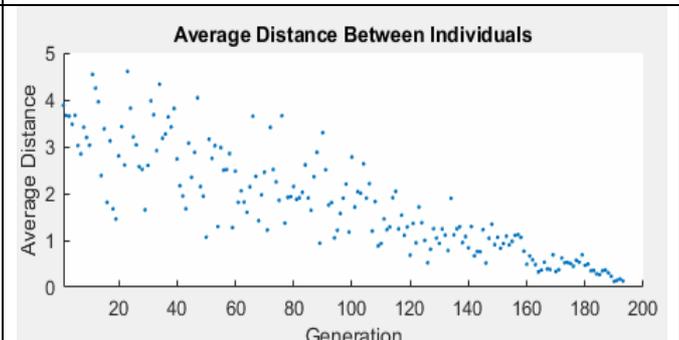
Номер і значення кращої хромосоми



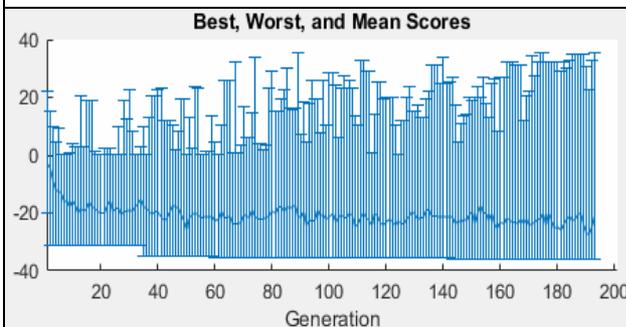
Генеалогія хромосом



Число нащадків хромосом



Відстань між хромосомами популяції



Оцінка придатності хромосом

6. Згенерувати m-файл (File->Generate code) з налагодженими параметрами (мутація схрещування, міграція і тому подібне) і включити його в звіт з лабораторної роботи (рис.12).

```

Editor - D:\code.m
func_1.m code.m +
1 function [x,fval,exitflag,output,population,score] = code(nvars,InitialPopulationRange_Data)
2 %% This is an auto generated MATLAB file from Optimization Tool.
3
4 %% Start with the default options
5 options = optimoptions('ga');
6 %% Modify options setting
7 options = optimoptions(options,'InitialPopulationRange', InitialPopulationRange_Data);
8 options = optimoptions(options,'Display', 'off');
9 options = optimoptions(options,'PlotFcn', { @gaplotbestf @gaplotbestindiv @gaplotdistance @gaplotexpectation @gaplotgenealogy});
10 [x,fval,exitflag,output,population,score] = ...
11 ga(@func_1,nvars,[],[],[],[],[],[],[],[],options);
12

```

Рис. 12. Сгенерований файл з налагодженими параметрами

Контрольні питання

1. Для вирішення яких завдань застосовується генетичний алгоритм?
2. Які генетичні оператори застосовуються у стандартному генетичному алгоритмі?
3. Як здійснюється кодування величин в хромосому?
4. Як впливають операції генетичного алгоритму на процес оптимізації?
5. У чому особливість генетичного алгоритму і його відмінність від звичайних алгоритмів пошуку оптимальних рішень (перебір, бульбашковий метод і так далі)?
6. Що таке функція корисності?

Лабораторна робота № 7

Тема: МІНІМІЗАЦІЯ БАГАТОВИМІРНИХ ФУНКЦІЙ З ОБМЕЖЕННЯМИ

Мета роботи: вирішити завдання мінімізації багатовимірної функції з обмеженнями методом генетичного алгоритму.

Постановка завдання: пошук мінімуму багатовимірних функцій з лінійними і нелінійними обмеженнями.

Короткі теоретичні відомості

На практиці часом складно, а деколи і неможливо, зафіксувати властивості функціональної залежності вихідних параметрів від вхідних величин, ще складніше привести аналітичний опис такої залежності. Ця обставина значно ускладнює використання на стадії проектування класичних методів оптимізації, оскільки більшість з них ґрунтуються на використанні апріорної інформації про характер поведінки цільової функції, а завдання визначення функції приналежності тому або іншому класу порівнянне по складності з початковим. У зв'язку з цим встає завдання побудови таких методів оптимізації, які були б здатні відшукувати рішення практично при повній відсутності припущень про характер досліджуваної функції. Одними з таких методів є, так звані, еволюційні методи пошуку і, зокрема, генетичні алгоритми (ГА), що моделюють процеси природної еволюції.

Генетичні алгоритми є одними з еволюційних алгоритмів, які використовуються для пошуку глобального екстремуму функції багатьох змінних. Принцип роботи генетичних алгоритмів заснований на моделюванні деяких механізмів генетики популяції: маніпулювання хромосомним набором при формуванні генотипу нової біологічної особини шляхом спадкоємства ділянок хромосомних наборів батьків (кросинговер), випадкова зміна генотипу, відома в

природі як мутація. Іншим важливим механізмом, запозиченим у природи, є процедура природного відбору, направлена на поліпшення від покоління до покоління пристосованості членів популяції шляхом більшої здібності до "виживання" особин, що володіють певними ознаками. Реалізацію базового генетичного алгоритму можна представити як ітераційний процес, що включає декілька етапів:

1. Генерація початкової популяції.
2. Відтворення "нащадків":
 - вибір батьківської пари;
 - вибір і реалізація одного з операторів кросовера;
 - вибір і реалізація одного з операторів мутації.
3. Створення репродукційної групи.
4. Процедура відбору і формування на його основі нового покоління.
5. Якщо не виконана умова зупинки, то перейти до п.2.

Завдання 1. Знайти мінімум функції, що має лінійні обмеження:

$$F(x) = 8x_1 + 9x_2 + 13x_3 \rightarrow \min$$

$$\begin{cases} x_1 + 6x_2 + 20x_3 \geq 90 \\ 3x_1 + 3x_2 + x_3 \geq 70 \\ x_1 + x_2 + x_3 \geq 25 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$$

Порядок виконання роботи

1. За допомогою команди HOME>New>Function створити m-файл з функцією розрахунку цільової функції s:

```
function s = test10(x)
s = 8*x(1)+9*x(2)+13*x(3)
end
```

Зберегти файл в робочу (або поточну) папку як **test10.m**.



The screenshot shows a MATLAB Editor window titled 'Editor - D:\test10.m'. The code in the editor is as follows:

```
1 function s = test10(x)
2   s = 8*x(1)+9*x(2)+13*x(3)
3   end
4
```

Рис. 1. Створений m-файл з функцією розрахунку цільової функції s

2. Задані обмеження є лінійними, тому їх потрібно привести до вигляду

$$Ax \leq b; A_{eq}x \leq b_{eq}.$$

Виконуючи перетворення, отримуємо:

```
>> A = [-1 -6 -20; -3 -3 -1; -1 -1 -1];
```

```
>> B = [-90; -70; -25];
```

```
>> Lb = [0; 0; 0];
```

3. Запустити генетичний алгоритм за допомогою команди:

```
>> [x fval reason] = ga(@test10,3,A,B,[],[],Lb,[])
```

Генетичний алгоритм легко знаходить рішення задачі:

```
Optimization terminated: average change in the fitness value less than options.FunctionTolerance.
x =
    11.7422    13.2567     0.0001

fval =
    213.2490

reason =
     1
```

Рис. 2. Результат пошуку мінімуму функції s

4. Перевірити отриманий результат за допомогою MATHCAD (рис. 3).

$$\begin{aligned}
 &F(x_1, x_2, x_3) := 8 \cdot x_1 + 9 \cdot x_2 + 13 \cdot x_3 \\
 &\underline{x_1} := 0 \quad \underline{x_2} := 0 \quad \underline{x_3} := 0 \\
 &\text{Given} \\
 & \quad x_1 + 6 \cdot x_2 + 20 \cdot x_3 \geq 90 \\
 & \quad 3 \cdot x_1 + 3 \cdot x_2 + x_3 \geq 70 \\
 & \quad x_1 + x_2 + x_3 \geq 25 \\
 & \quad x_1 \geq 0 \\
 & \quad x_2 \geq 0 \\
 & \quad x_3 \geq 0 \\
 & R := \text{Minimize}(F, x_1, x_2, x_3) \quad R = \begin{pmatrix} 12 \\ 13 \\ 0 \end{pmatrix} \\
 & \underline{x_1} := 12 \quad \underline{x_2} := 13 \quad \underline{x_3} := 0 \\
 & F(x_1, x_2, x_3) = 213
 \end{aligned}$$

Рис. 3. Перевірка отриманого результату за допомогою MATHCAD

5. Виконати мінімізацію за допомогою майстра генетичного алгоритму. Для цього необхідно використати пакет Optimization Tool.

Ввести в командному вікні команду:

>> optimtool

На екрані з'явиться основне графічне вікно пакету Optimization Tool.

Вибрати в рядку Solver: **ga – Genetic Algorithm**. Всі інші поля заповнити, як показано на рис. 4.

6. Запустити алгоритм пошуку за допомогою кнопки Start. В результаті отримаємо мінімум функції і значення аргументів функції (рис. 5).

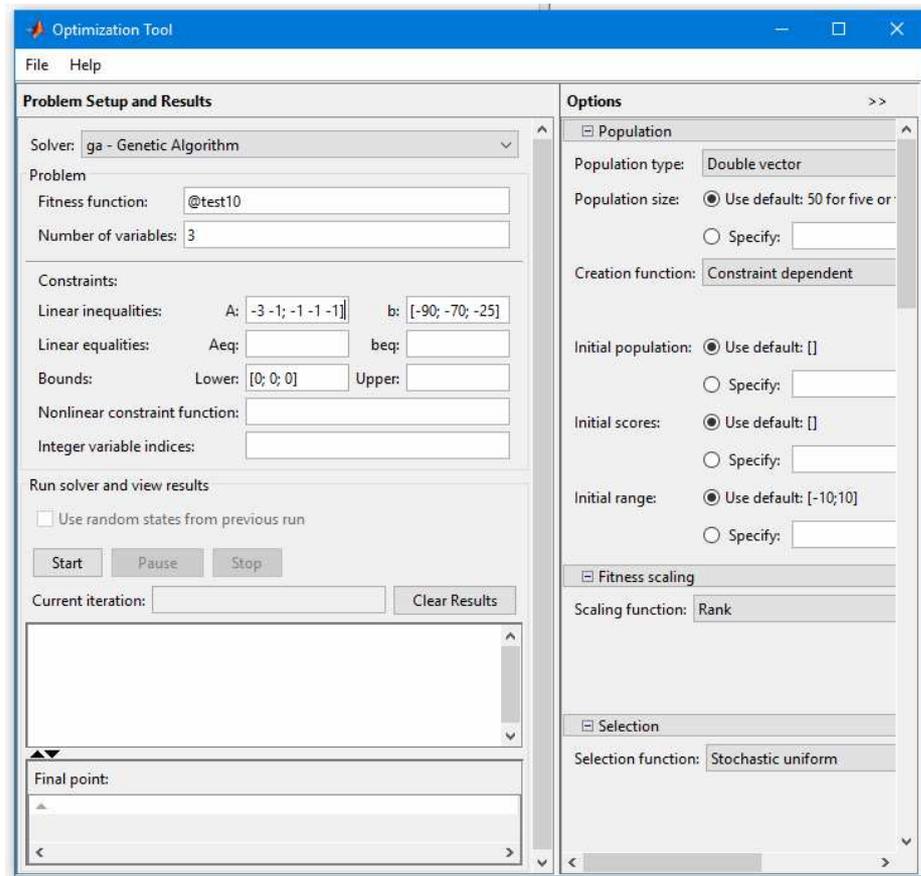


Рис. 4. Завдання параметрів обчислення

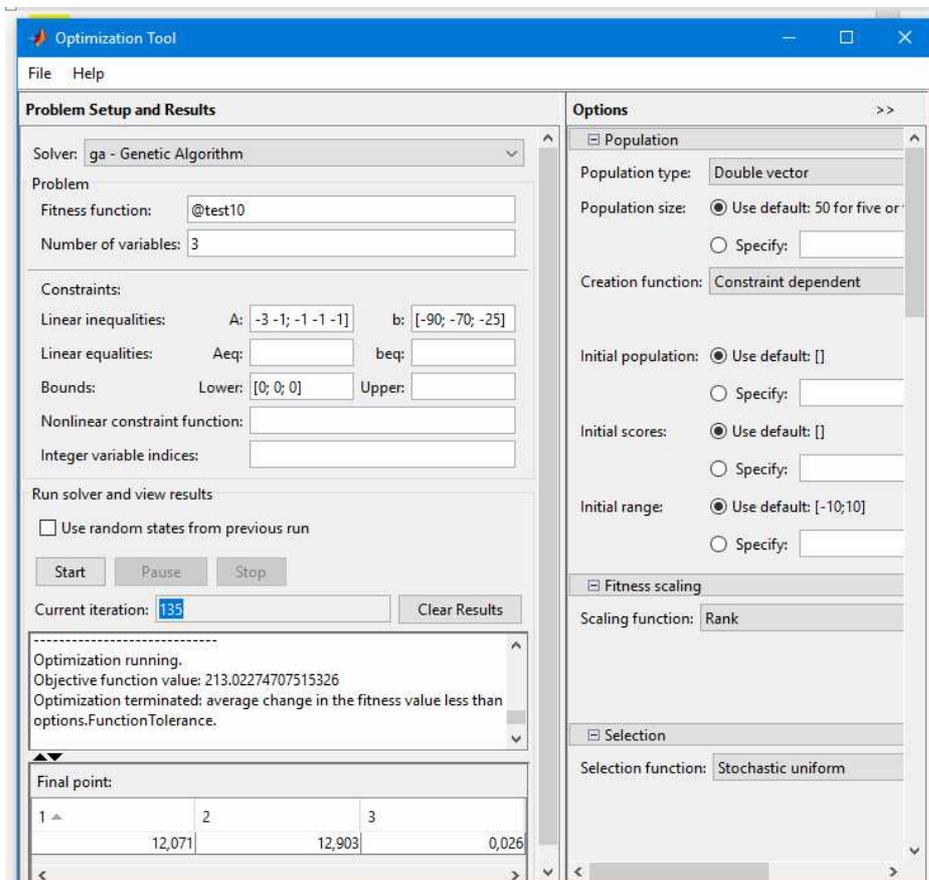


Рис. 5. Результат пошуку рішення у вікні Optimization Tool

7. Порівняти знайдені значення мінімуму функції за допомогою майстра генетичного алгоритму із значеннями, знайденими за допомогою функції (пункт 3) та значеннями, розрахованими за допомогою MATHCAD (пункт 4).

В результаті аналізу отриманих результатів мінімуму функції можна зробити вивід, що вони майже збігаються.

Варіанти завдання 1

№ варіанту	Функція	Обмеження
1	2	3
1	$F(x) = -2x_1 + x_2 - 3x_4 \rightarrow \min$	$\begin{cases} x_1 + 2x_2 - 3x_3 + 2x_4 \geq 8 \\ -x_1 + x_2 + 2x_3 - x_4 \geq 2 \\ 2x_1 - x_3 + 3x_4 \geq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \\ x_4 \geq 0 \end{cases}$
2	$F(x) = x_1 - 2x_2 \rightarrow \min$	$\begin{cases} x_1 - x_2 \geq 1 \\ x_1 + x_2 \geq 2 \\ x_1 - x_2 \geq 0 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$
3	$F(x) = 40x_1 + 36x_2 \rightarrow \min$	$\begin{cases} 5x_1 + 3x_2 \geq 45 \\ 9x_1 + x_2 \geq 4 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$
4	$F(x) = x_1 + 2x_2 \rightarrow \min$	$\begin{cases} x_1 + 2x_2 \geq 10 \\ x_1 - x_2 \geq 1 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$

1	2	3
5	$F(x) = 2x_1 + x_2 + 5x_3 \rightarrow \min$	$\begin{cases} -x_1 + 6x_2 + 20x_3 \geq 90 \\ 3x_1 + 3x_2 + x_3 \geq 70 \\ x_1 + x_2 + x_3 \geq 25 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$
6	$F(x) = 8x_1 + 9x_2 + 13x_3 \rightarrow \min$	$\begin{cases} x_1 + 6x_2 + 20x_3 \geq 90 \\ 3x_1 + 3x_2 + x_3 \geq 70 \\ x_1 + x_2 + x_3 \geq 25 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$
7	$F(x) = 4x_1 + 2x_2 \rightarrow \min$	$\begin{cases} -x_1 + 3x_2 \geq 9 \\ 2x_1 + 3x_2 \geq 28 \\ 2x_1 - x_2 \geq 10 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{cases}$
8	$F(x) = x_1 - 2x_2 + 3x_3 \rightarrow \min$	$\begin{cases} x_1 + x_2 + x_3 \geq 7 \\ x_1 - x_2 + x_3 \geq 2 \\ 3x_1 - x_2 - 2x_3 \geq -5 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$
9	$F(x) = -3x_1 + x_2 + x_3 \rightarrow \min$	$\begin{cases} x_1 - 2x_2 + x_3 \geq 11 \\ -4x_1 + x_2 + 2x_3 \geq 3 \\ 2x_1 - x_3 \geq -1 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$

1	2	3
10	$F(x) = 10x_1 + 6x_2 + 4x_3 \rightarrow \min$	$\begin{cases} x_1 + x_2 + x_3 \geq 100 \\ 10x_1 + 4x_2 + 5x_3 \geq 150 \\ 2x_1 + 2x_2 + 6x_3 \geq 300 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{cases}$

Завдання 2. Знайти мінімум функції, що має нелінійні обмеження:

$$f(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 \rightarrow \min$$

$$\begin{cases} x_1x_2 + x_1 - x_2 + 1,5 \leq 0 \\ 10 - x_1x_2 \leq 0 \\ 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 13 \end{cases}$$

Порядок виконання роботи

1. За допомогою команди HOME>New>Function створити m-файл з функцією розрахунку цільової функції s:

```
function s = test20(x)
s = 100*(x(1)^2-x(2))^2+(1-x(1))^2;
end
```

Зберегти файл в робочу (або поточну) папку під ім'ям **test20.m** (рис. 6).

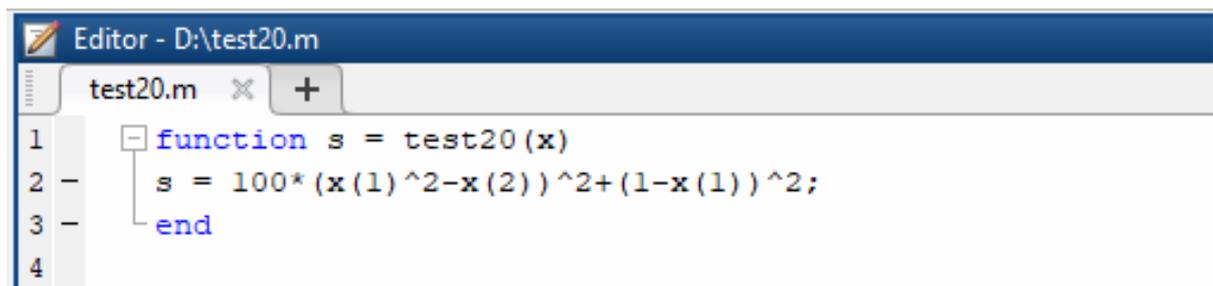
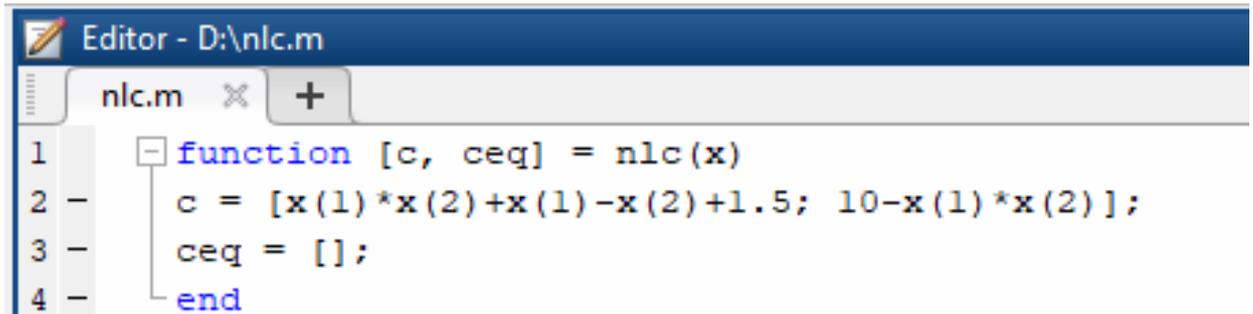


Рис. 6. Створений m-файл з функцією розрахунку цільової функції s

2. Задані нелінійні обмеження у вигляді нерівностей можна описати за допомогою файлу-функції:

```
function [c, ceq] = nlc(x)
c = [x(1)*x(2)+x(1)-x(2)+1.5; 10-x(1)*x(2)];
ceq = [];
end
```

Зберегти файл в робочу (або поточну) теку під ім'ям **nlc.m** (рис. 7).



```
Editor - D:\nlc.m
nlc.m x +
1 function [c, ceq] = nlc(x)
2 c = [x(1)*x(2)+x(1)-x(2)+1.5; 10-x(1)*x(2)];
3 ceq = [];
4 end
```

Рис. 7. Створений m-файл з функцією завдання нелінійних обмежень

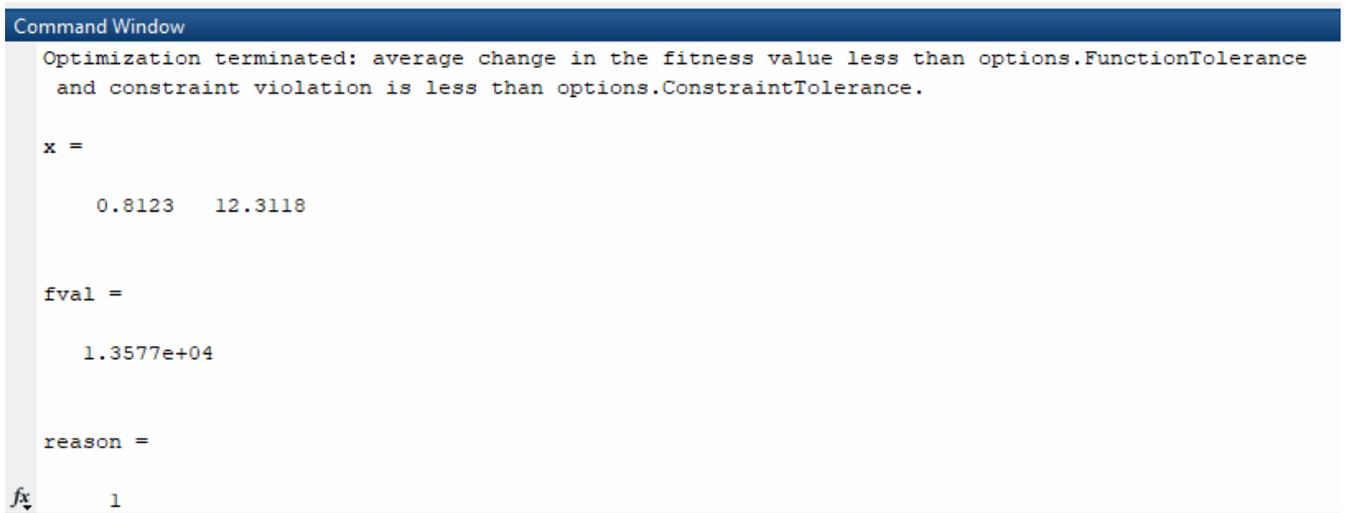
3. Задати межі зміни змінних:

```
>> Lb = [0; 0];
>> Ub = [1; 13];
```

4. Запустити генетичний алгоритм за допомогою команди:

```
>> [x fval reason] = ga(@test20,2,[],[],[],[],Lb,Ub,@nlc)
```

Отримуємо рішення задачі:



```
Command Window
Optimization terminated: average change in the fitness value less than options.FunctionTolerance
and constraint violation is less than options.ConstraintTolerance.

x =

    0.8123    12.3118

fval =

    1.3577e+04

reason =

    1
```

Рис. 8. Результат пошуку мінімуму функції s

5. Перевірити отриманий результат за допомогою MATHCAD (рис. 9).

$$F(x1, x2) := 100 \cdot (x1^2 - x2)^2 + (1 - x1)^2$$

$$x1 := 0 \quad x2 := 0$$

Given

$$x1 \cdot x2 + x1 - x2 + 1.5 \leq 0$$

$$10 - x1 \cdot x2 \leq 0$$

$$0 \leq x1 \leq 1$$

$$0 \leq x2 \leq 13$$

$$R := \text{Minimize}(F, x1, x2) \quad R = \begin{pmatrix} 0.8122 \\ 12.3122 \end{pmatrix}$$

$$x1 := 0.8122 \quad x2 := 12.3122$$

$$F(x1, x2) = 1.358 \times 10^4$$

Рис. 9. Перевірка отриманого результату за допомогою MATHCAD

6. Виконати мінімізацію за допомогою майстра генетичного алгоритму. Для цього необхідно використати пакет Optimization Tool.

Ввести в командному вікні команду:

>> optimtool

На екрані з'явиться основне графічне вікно пакету Optimization Tool.

Вибрати в рядку Solver: **ga – Genetic Algorithm**. Всі інші поля заповнити, як показано на рис. 10.

7. Запустити алгоритм пошуку за допомогою кнопки Start. В результаті отримаємо мінімум функції Z і значення аргументів функції (рис. 11).

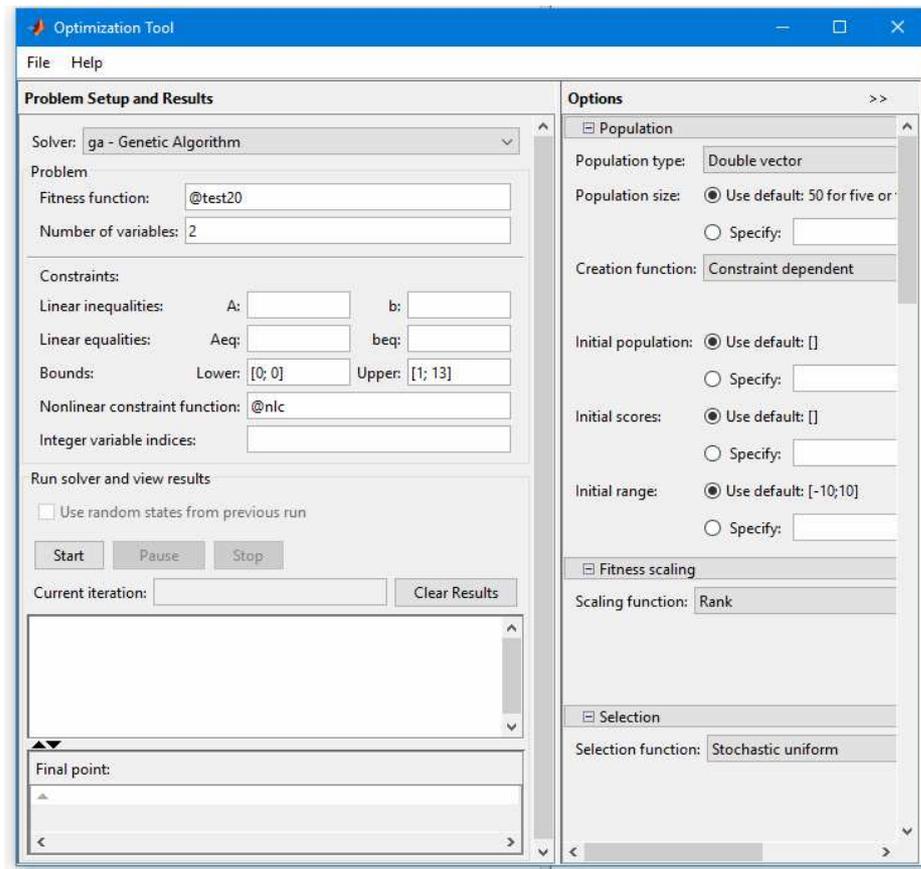


Рис. 10. Завдання параметрів обчислення

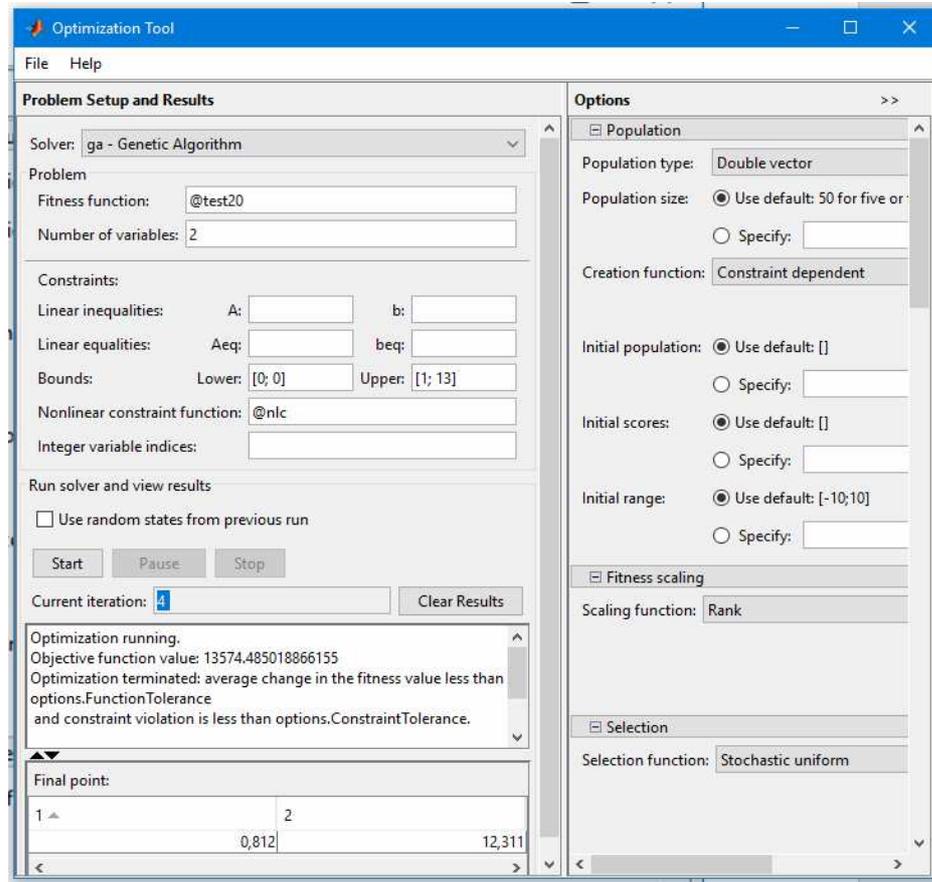


Рис. 11. Результат пошуку рішення у вікні Optimization Tool

8. Порівняти, знайдені значення мінімуму функції s із значеннями, знайденими за допомогою функції (пункт 4) і значеннями, розрахованими за допомогою MATHCAD (пункт 5).

В результаті аналізу знайденого мінімуму функції s за допомогою генетичного алгоритму і за допомогою функції можна зробити вивід, що вони майже збігаються.

Варіанти завдання 2

№ варіанту	Функція	Обмеження
1	2	3
1	$f(x) = 2x_1^2 + x_2^2 \rightarrow \min$	$\begin{cases} x_1^2 - x_2 \leq 0 \\ x_1 + x_1x_2 \leq 0 \\ 0 \leq x_1 \leq 2 \\ 1 \leq x_2 \leq 10 \end{cases}$
2	$f(x) = 1 - 2x_1 - 4x_1x_2 \rightarrow \min$	$\begin{cases} x_1 + 4x_1x_2 \leq 0 \\ 2x_1x_2 - x_2 \leq 0 \\ 0 \leq x_1 \leq 3 \\ 2 \leq x_2 \leq 5 \end{cases}$
3	$f(x) = 3x_1^2 - 2x_2 \rightarrow \min$	$\begin{cases} 2x_1 + x_2 \leq 0 \\ x_1^2 + x_2^2 \leq 0 \\ 1 \leq x_1 \leq 2 \\ 0 \leq x_2 \leq 7 \end{cases}$
4	$f(x) = x_1^2 - 7x_2^2 \rightarrow \min$	$\begin{cases} (x_1 - 1)^2 - x_2^3 \leq 0 \\ x_1 + x_1x_2 \leq 0 \\ 0 \leq x_1 \leq 0,6 \\ 0 \leq x_2 \leq 1 \end{cases}$

1	2	3
5	$f(x) = x_1 + 3x_2^2 \rightarrow \min$	$\begin{cases} 2 - x_1 - x_2^2 \leq 0 \\ x_1^2 - x_2 \leq 0 \\ 2 \leq x_1 \leq 5 \\ 1 \leq x_2 \leq 2 \end{cases}$
6	$f(x) = x_1^4 + x_2 \rightarrow \min$	$\begin{cases} 9 - 2x_1^2 - 3x_2 \leq 0 \\ x_1 + x_2^2 \leq 0 \\ 0 \leq x_1 \leq 2 \\ 1 \leq x_2 \leq 7 \end{cases}$
7	$f(x) = (x_1 - 2)^2 + (x_2 - 2)^2 \rightarrow \min$	$\begin{cases} 3 - x_1 - x_2 \leq 0 \\ 10x_1 + x_2^2 - 2 \leq 0 \\ 1 \leq x_1 \leq 3 \\ 1 \leq x_2 \leq 2 \end{cases}$
8	$f(x) = 2x_1 - 3(x_2 - 4)^2 \rightarrow \min$	$\begin{cases} 10 - x_1^2 - x_2^2 \leq 0 \\ 9 - x_1^2 - (x_2 - 4)^2 \leq 0 \\ 0 \leq x_1 \leq 1 \\ 0 \leq x_2 \leq 1 \end{cases}$
9	$f(x) = x_1 - 2x_2^2 \rightarrow \min$	$\begin{cases} 25 - x_1^2 - x_2^2 \leq 0 \\ 10 - x_1^2 + x_2 \leq 0 \\ 0 \leq x_1 \leq 5 \\ 0 \leq x_2 \leq 10 \end{cases}$
10	$f(x) = x_1^2 - 11x_2 \rightarrow \min$	$\begin{cases} 8x_1 + 3x_2 \leq 0 \\ 2x_1^2 + 3x_2^2 \leq 0 \\ 0 \leq x_1 \leq 10 \\ 0 \leq x_2 \leq 8 \end{cases}$

Контрольні питання

1. Основні цілі та завдання генетичних алгоритмів.
2. Критерії ефективності генетичних алгоритмів.
3. Формування початкової популяції. Основні принципи.
4. Генетичні оператори. Наведіть основні види цих операторів.
5. Що таке генетичне програмування?
6. Методика генетичного програмування.

СПИСОК ЛІТЕРАТУРИ

1. Доля В. Г. Комп'ютерні системи штучного інтелекту. - Київ, 2011. – 295 с.
2. Субботін С.О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Навчальний посібник. – Запоріжжя: ЗНТУ, 2008. – 341 с.
3. Нікольський Ю.В., Пасічник В.В., Щербина Ю.М. Системи штучного інтелекту: Навч. посіб./За наук. ред. В.В. Пасічника. - Львів: Магнолія 2006, 2010. – 279с.
4. Дубровін В.І., Субботін С.О. Методи оптимізації та їх застосування в задачах навчання нейронних мереж: Навчальний посібник. – Запоріжжя: ЗНТУ, 2003. – 136 с.
5. Томашевський В.М. Моделювання систем: Підручник. – К: Видавнича група ВНУ, 2005. – 352 с.
6. Гайна Г.А. Системи штучного інтелекту: Навчальний посібник. – К. : «КНУБА», 2007. – 208 с.
7. Леоненков А.В. Нечеткое моделирование в среде MATLAB и fuzzyTECH.–СПб.: БХВ-Петербург, 2003.
8. Харазов В.Г. Интегрированные системы управления технологическими процессами. – СПб.: Профессия, 2009. – 592 с.
9. Денисенко В.В. Компьютерное управление технологическим процессом, экспериментом, оборудованием. – М.: Горячая линия – Телеком, 2009. – 608 с.
10. Васильев В.И., Ильясов Б.Г. Интеллектуальные системы управления. Теория и практика: учебное пособие. – М.: Радиотехника, 2009. - 392 с.
11. Прикладные нечеткие системы / под ред. Т. Тэрано, К. Асаи, М. Сугэно. – М.: Мир, 1993. – 368 с.
12. Штовба С. Д. Проектирование нечетких систем средствами MATLAB. – М.: Горячая линия – Телеком, 2007. – 288 с.
13. Ярушкина Н. Г. Основы теории нечетких и гибридных систем. – М.: Финансы и статистика, 2004. – 320 с.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт з дисципліни

«ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ УПРАВЛІННЯ»

(для здобувачів вищої освіти заочної форми навчання

спеціальності 151 Автоматизація та комп'ютерно–інтегровані технології)

(Електронне видання)

Укладачі:

ПРОКАЗА Олена Іванівна

КУЗНЕЦОВА Олена Володимирівна

Оригінал - макет О.І. Проказа

Підписано до друку _____

Формат 60×84 $\frac{1}{16}$. Папір типограф. Гарнітура Times.

Друк офсетний. Умовн. друк. арк. _____. Облік.-видавн.арк._____.

Тираж ____ екз. Вид. №_____. Замовл. №_____. Ціна договірна.

Видавництво Східноукраїнського національного університету імені
імені Володимира Даля

Адреса видавництва: м. Сєверодонецьк, просп. Центральний, 59а

Телефон: +38 (050) 218 04 78, факс (064 52) 4 03 42

E-mail: vidavnictvosnu.ua@gmail.com