

Кабінет Міністрів України
Національний університет біоресурсів і природокористування України
Кафедра автоматики і робототехнічних систем
ім. акад. І.І. Мартиненка

Лисенко В.П., Штепа В.М., Заєць Н.А.

ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ
НЕЧІТКА ЛОГІКА

(ПАКЕТ FUZZY LOGIC TOOLBOX)

(навчальний посібник)

Для студентів напряму підготовки

6.050101 – «Комп'ютерні науки»

6.050202 – «Автоматизація та комп'ютерно-інтегровані технології»

6.100101 – «Енергетика та електротехнічні системи в агропромисловому комплексі»

(денна та заочна форма навчання)

Київ – 2010

УДК 604.94(075.8)

У навчальному посібнику викладено матеріал занять із дисципліни «Інтелектуальні системи» у розрізі використання прикладного програмного продукту Fuzzy Logic Toolbox системи MatLAB.

Навчальний посібник призначено для використання при підготовці фахівців напрямів підготовки: 6.050101 «Комп'ютерні науки», 6.050202 – «Автоматизація та комп'ютерно-інтегровані технології», 6.100101 – «Енергетика та електротехнічні системи в агропромисловому комплексі».

Рекомендовано до видання навчально-методичною радою факультету енергетики і автоматики Національного університету біоресурсів і природокористування України.

Укладачі: кандидат технічних наук, професор Лисенко В.П., кандидат технічних наук, старший викладач Штепа В.М., кандидат технічних наук, старший викладач Заєць Н.А.

Рецензенти: доктор технічних наук, професор Донченко М.І. (Національний технічний університет «КП»), доктор технічних наук, професор Котов Б.І. (Національний університет біоресурсів і природокористування України)

Навчальне видання
**ІНТЕЛЕКТУАЛЬНІ СИСТЕМИ НЕЧІТКА ЛОГІКА
(ПАКЕТ FUZZY LOGIC TOOLBOX)**
(Навчальний посібник)

ДЛЯ СТУДЕНТІВ НАПРЯМІВ ПІДГОТОВКИ:
6.050101 – «Комп'ютерні науки»
6.050202 – «Автоматизація та комп'ютерно-інтегровані технології»
6.100101 – «Енергетика та електротехнічні системи в агропромисловому комплексі»

Укладачі: ЛИСЕНКО Віталій Пилипович
ШТЕПА Володимир Миколайович
ЗАЄЦЬ Наталія Анатоліївна

Зав. видавничим центром НУБіП України А.П. Колесніков
Редактор

Підписано до друку
Ум. друк. арк.
Наклад 100 прим.
Видавничий центр НУБіП України
03041, Київ, вул. Героїв Оборони, 15

Формат 60×84 1/16
Обл.-вид. арк. 10
Зам. №

ЗМІСТ

Призначення і можливості пакету Fuzzy Logic Toolbox	4
Графічний інтерфейс Fuzzy Logic Toolbox	4
Робота Fuzzy Logic Toolbox у режимі командного рядка.....	16
Побудова нечіткої моделі з використанням блоків Simulink.....	34
Демонстраційні приклади роботи у пакеті Fuzzy Logic Toolbox.....	34
Література.....	37

Призначення і можливості пакету Fuzzy Logic Toolbox

Пакет Fuzzy Logic Toolbox (пакет нечіткої логіки) – це сукупність прикладних програм, що відносяться до теорії *розмитих* або *нечітких* множин, які дозволяють конструювати нечіткі експертні і керуючі системи.

Основні можливості пакета:

- побудова систем нечіткого висновку (експертних систем, регуляторів, апроксиматорів залежностей);
- побудова адаптивних нечітких систем;
- інтерактивне динамічне моделювання в Simulink.

Пакет дозволяє працювати:

- у режимі графічного інтерфейсу;
- у режимі командного рядка;
- с використанням блоків і прикладів пакету Simulink.

1. Графічний інтерфейс Fuzzy Logic Toolbox

1.1. Склад графічного інтерфейсу. До складу програмних засобів Fuzzy Logic Toolbox входять наступні основні програми, що дозволяють працювати в режимі графічного інтерфейсу: редактора нечіткої системи висновку Fuzzy Inference System Editor (FIS Editor або FIS-редактор) разом з допоміжними програмами – редактором функцій належності (Membership Function Editor), редактором правил (Rule Editor), браузером правил (Rule Viewer) і браузером поверхні відгуку (Surface Viewer).

Набір даних програм надає користувачеві максимальні зручності для створення, редагування і використання різних систем нечіткого висновку.

1.2. Побудова нечіткої апроксимуючої системи. Командою (функцією) **Fuzzy** з режиму командного рядка запускається основна інтерфейсна програма пакету Fuzzy Logic — редактор нечіткої системи висновку (Fuzzy Inference System Editor, FIS Editor, FIS-редактор). Вигляд вікна, що відкривається при цьому, приведено на рисунку 1.1.

Головне меню редактора містить позиції:

File – робота з файлами моделей (їхнє створення, збереження, зчитування і друк);

Edit – операції редагування (додавання і виключення вхідних і вихідних змінних);

View – перехід до додаткового інструментарію.

Сконструюємо нечітку систему, що відображає залежність між змінними x і y , задану за допомогою таблиці 1.1 (легко бачити, що представлені в таблиці дані відбивають залежність $y = x^2$).

Залежність y від x

x	-1	-0,6	0	0,4	1
y	1	0,36	0	0,16	1

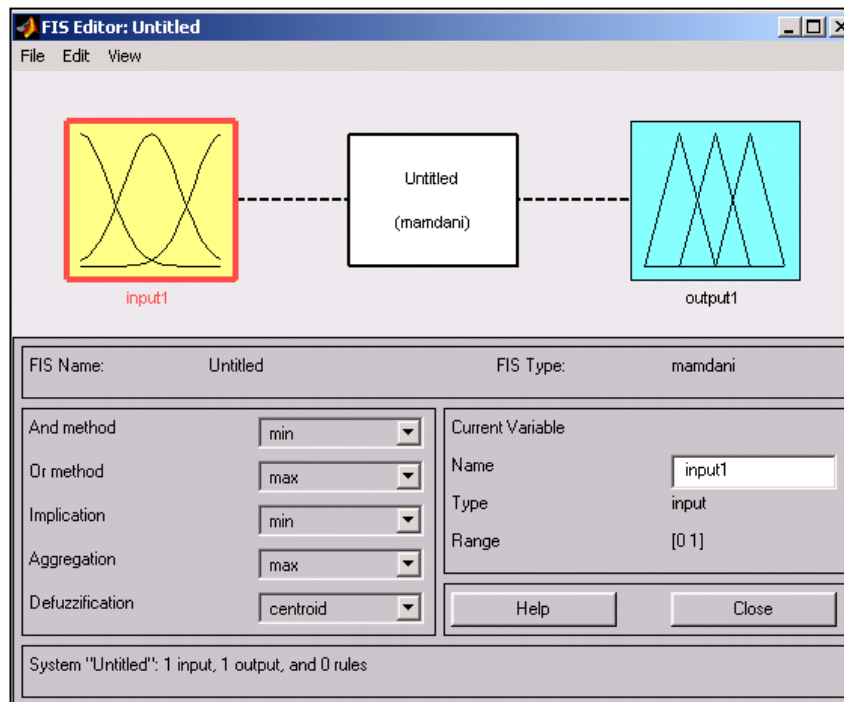


Рис. 1.1. Вид вікна FIS Editor

Необхідні дії відобразимо наступними пунктами.

1. У позиції меню **File** вибираємо опцію **New Sugeno FIS** (система типу Sugeno), при цьому в блоці зображеному білим квадратом, у верхній частині вікна редактора, з'явиться напис **Untitled2 (sugeno)**.

2. Клікнемо лівою кнопкою миші по блоці **input 1** (вході). Потім у правій частині редактора в поле **Name** (Ім'я), замість **input 1** введемо позначення нашого аргументу, тобто x . Зверніть увагу, що якщо тепер зробити поза блоками редактора одноразовий клік миші, то ім'я відзначеного блоку зміниться на x ; те ж досягається натисненням після введення клавіші **Enter**.

3. Двічі клікнемо на цьому блоці. Перед нами відкриється вікно редактора функцій належності – **Membership Function Editor** (рис. 1.2). Ввійдемо в позицію меню **Edit** редактора і виберемо в ньому опцію **Add MFs** (Add Membership Functions). Додати функції належності. При цьому з'явиться діалогове вікно (рис. 1.3), яке дозволяє задати тип (**MF type**) і кількість (**Number of MFs**) функцій належності (у даному випадку всі вони відносяться до вхідного сигналу, тобто до змінної x). Виберемо гауссівські функції належності (**gaussmf**), а їхню кількість задамо рівною п'яти – за числом значень аргументів у таблиці 1.1. Створенні за умовчанням функції належності видалемо. Підтвердимо введення інформації натисненням кнопки **OK**, після чого відбудеться повернення до вікна редактора функцій належності.

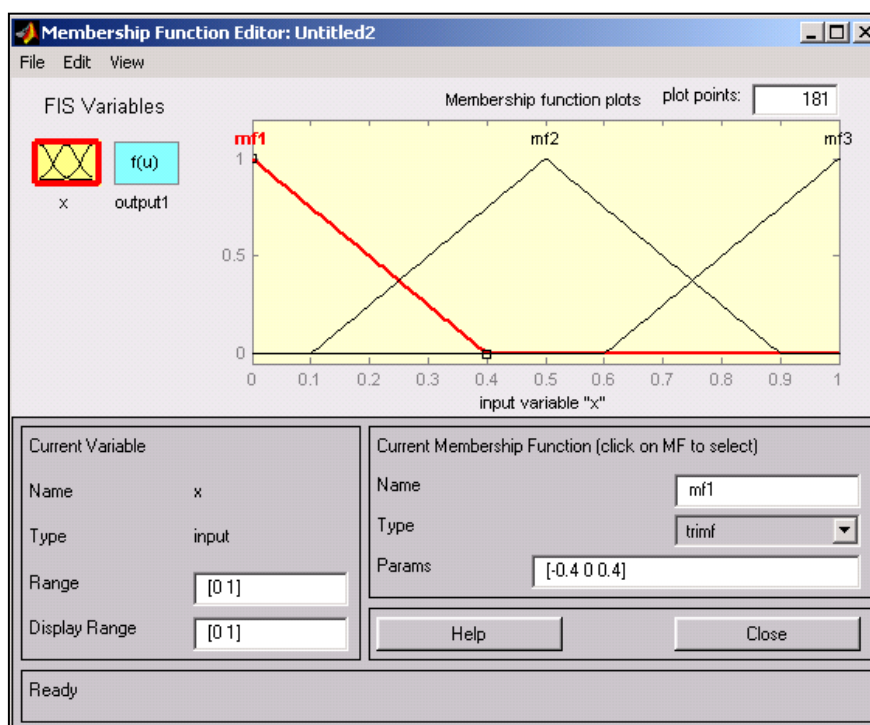


Рис. 1.2. Вікно редактора функцій належності

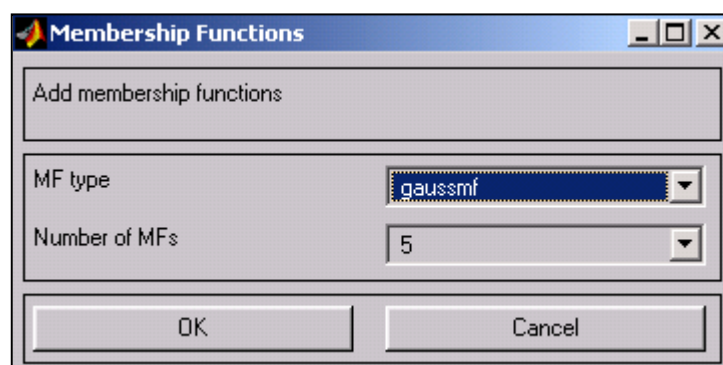


Рис. 1.3. Діалогове вікно задання типу і кількості функцій належності

4. У полі **Range** (Діапазон) встановимо діапазон зміни x від -1 до 1, тобто діапазон, що відповідає таблиці 1.1. Клікнемо потім лівою кнопкою миші у полі редактора (або нажмемо клавішу введення **Enter**). Зверніть увага, що після цього відбудеться відповідна зміна діапазону в полі **Display Range** (Діапазон дисплея).

5. Звернемося до графіків заданих нами функцій належності, зображеним у верхній частині вікна редактора функцій належності. Для успішного вирішення поставленої задачі необхідно, щоб ординати максимумів цих функцій збігалися з заданими значеннями аргументу x . Для лівої, центральної і правої функцій така умова виконана, але дві інші необхідно «підсунути» уздовж осі абсцис. «Пересування» робиться досить просто: підводимо курсор до потрібної кривої і клацаємо лівою кнопкою миші. Крива виділяється, офарблюючись в червоний колір, після чого за допомогою курсору її можна підсунути в потрібну сторону

(більш точну установку можна провести, змінюючи числові значення в полі **Params** (Параметри) – у даному випадку кожній функції належності відповідають два параметри, при цьому перший визначає розмах кривої, а другий – положення її центра). Для обраної кривої, крім цього, у полі **Name** можна змінювати ім'я (завершуючи введення кожного імені натисканням клавіші **Enter**). Зробимо необхідні переміщення кривих і задамо всім п'яти кривим нові імена:

- самій лівій – bn,
- наступній – n,
- центральній – z,
- наступній за нею (праворуч) – p,
- самій правій – bp.

Натиснемо кнопку **Close** і вийдемо з редактора функцій належності, повернувшись при цьому у вікно редактора нечіткої системи (**FIS Editor**).

6. Зробимо одноразовий клік лівою кнопкою миші на полі блоку **output1** (вихід). У вікні **Name** замінимо ім'я **output1** на **u**.

7. Двічі клікнемо по відзначеному блоці і перейдемо до редактора функцій належності. У позиції меню **Edit** виберемо опцію **Add MFs**. Діалогове вікно (рис. 1.4) дозволяє задати як функції належності тільки лінійні (linear) або постійні (constant) – у залежності від того, який алгоритм Sugeno (1-го або 0-го порядку) ми вибираємо. У розглянутій задачі необхідно вибрати постійні функції належності у кількості 4 (за числом різних значень u у таблиці 1.1). Підтвердимо введення даних натисканням кнопки **OK**, після чого відбудеться повернення у вікно редактора функцій належності.

8. Діапазон (**Range**) зміни, встановлюється за замовчуванням, змінювати його не потрібно. Змінимо лише імена функцій належності (їхній графіки при використанні алгоритму Sugeno для вихідних змінних не приводяться), наприклад, задавши їх як відповідні числові значення u , тобто: 0, 0,16, 0,36, 1; одночасно ці ж числові значення введемо в поле **Params** (рис. 1.4). Потім закриємо вікно натисканням кнопки **Close** і повернемося у вікно FIS-редактора.

9. Двічі клікнемо лівою кнопкою миші на середньому блоці, при цьому розкриється вікно ще однієї програми – редактора правил (**Rule Editor**). Введемо відповідні правила. При введенні кожного правила необхідно позначити відповідність між кожною функцією належності аргументу x і числовим значенням u . Крива, позначена нами bn, відповідає $x = -1$, тобто $u = 1$. Виберемо, тому в лівому полі (із заголовком x is) bn, а в правому 1 і натиснемо кнопку **Add rule** (Додати правило). Введене правило з'явиться у вікні правил і буде являти собою запис: *1. If (x is bn) then (y is 1) (1)*. Аналогічно зробимо для всіх інших значень x , у результаті чого сформується набір з 5 правил (рис. 1.5). Закриємо вікно редактора правил і повернемося у вікно FIS-редактора. Побудова системи закінчена і можна почати її дослідження. Відмітимо, що більшість опцій вибиралося нами за замовчуванням.

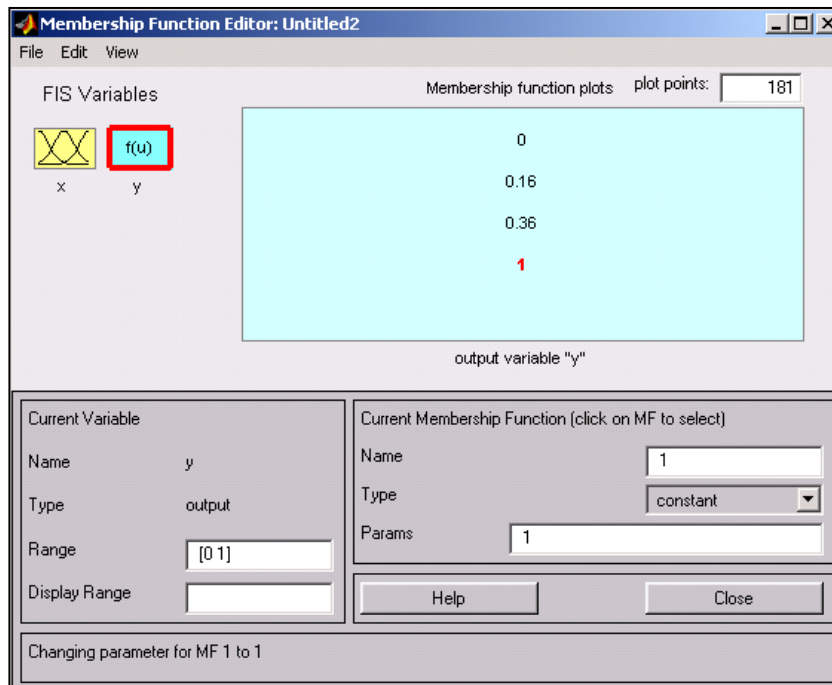


Рис. 1.4. Параметри функцій належності змінної y

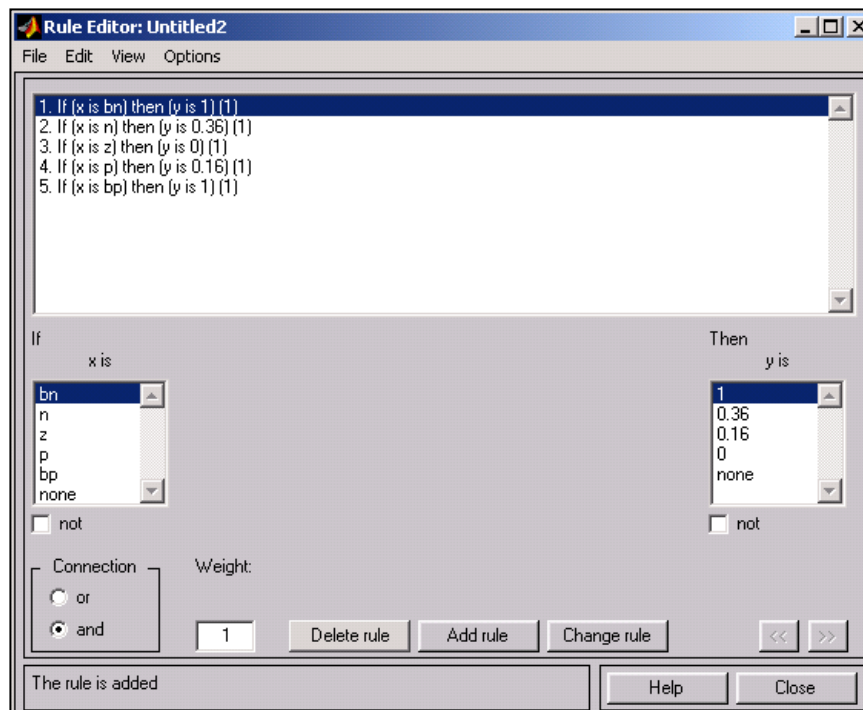


Рис. 1.5. Редактор правил

10. Попередньо збережемо на диску (використовуючи пункти меню **File/Save to disk as...**) створену систему під яким-небудь іменем, наприклад, **Proba**.

11. Виберемо позицію меню **View**. Як видно з підменю, за допомогою пунктів **Edit membership functions** і **Edit rules** можна зробити перехід до двох вищерозглянутих програм – редакторів функцій належності і правил (те ж можна зробити і натисканням клавіш **Ctrl+2** або **Ctrl+3**). Виберемо пункт **View**

rules, при цьому відкриється вікно (рис. 1.6) ще однієї програми – перегляду правил (**Rule Viewer**).

12. У правій частині вікна в графічній формі представлені функції належності аргументу x , у лівій – змінної виходу y з поясненням механізму ухвалення рішення. Червона вертикальна лінія, яку можна переміщувати за допомогою курсору, дозволяє змінювати значення змінної входу (це ж можна робити, задаючи числові значення в поле **Input** (Вхід)), при цьому відповідно змінюються значення y у правій верхній частині вікна. Задамо, наприклад, $x = 0,5$ у поле **Input** і натиснемо потім клавішу введення (**Enter**). Значення y відразу зміниться і стане рівним **0,202**. Таким чином, за допомогою побудованої моделі і вікна перегляду правил можна вирішувати задачу інтерполяції. Зміна аргументу шляхом переміщення червоної вертикальної лінії наочно демонструє, як система визначає значення виходу.

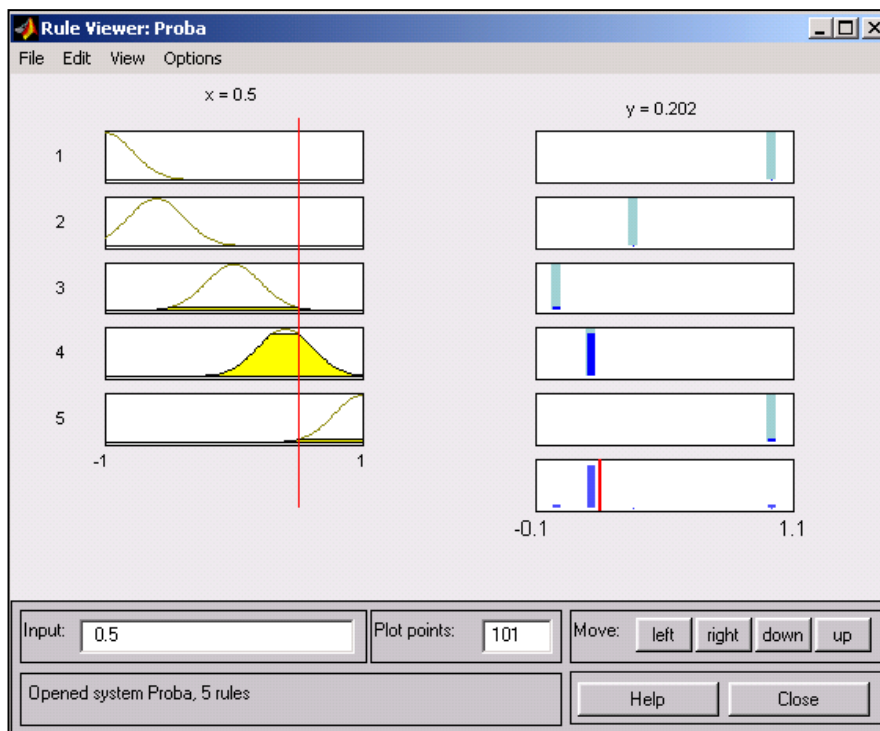


Рис. 1.6. Вікно перегляду реалізації правил

13. Закриємо вікно перегляду правил і вибором пункту меню **View/View surface** перейдемо до вікна перегляду поверхні відгуку (виходу), у нашому випадку – до перегляду кривої $y(x)$ (рис. 1.7). Видно, що змодельоване системою за таблицею даних (табл. 1.1) відображення не дуже нагадує функцію x^2 . Нічого дивного в цьому немає: число експериментальних точок невелике, та й параметри функцій належності (для x) обрані, швидше за все, неоптимальним образом. Нижче ми розглянемо можливість поліпшення якості подібної моделі.

На закінчення розгляду прикладу відзначимо, що за допомогою вищевказаних програм-редакторів на будь-якому етапі проектування нечіткої моделі в неї можна внести необхідні корективи, аж до задання особливої функції належності.

З опцій, встановлюваних у FIS-редакторі за замовчуванням при використанні алгоритму Sugeno, можна відзначити:

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

Використовуючи відповідні поля в лівій нижній частині вікна FIS-редактори, дані опції можна, при бажанні, змінити.

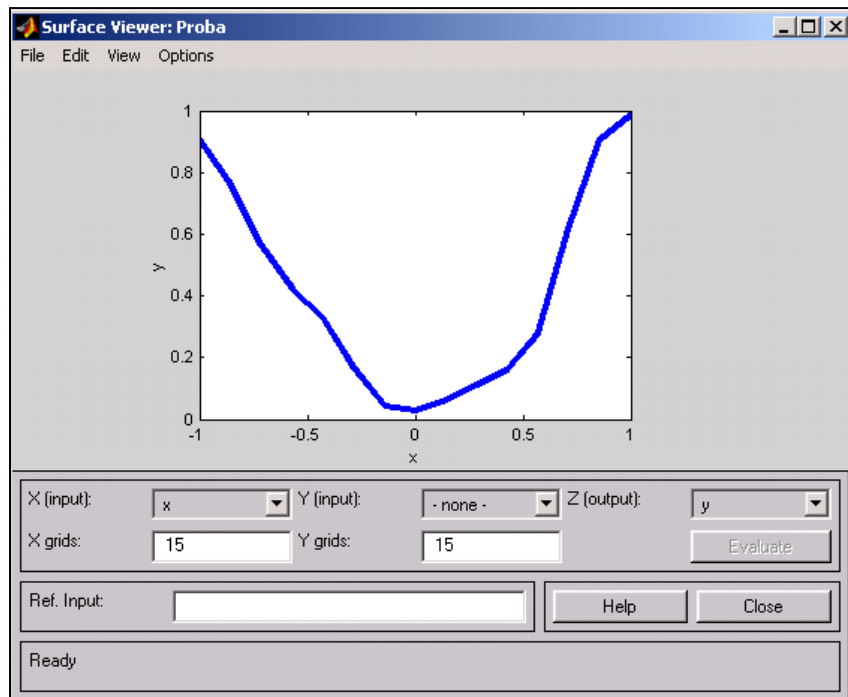


Рис. 1.7. Вікно перегляду поверхні відгуку

Завдання

1. Реалізувати наведений приклад.
2. Згідно власного варіанту, створити нечітку експертну систему, яка опрацьовуватиме формулу: $y = N \cdot x^3$ (N – номер варіанту).

1.3. Побудова експертної системи: скільки дати «на чай»?

Розглянемо тепер методику побудови нечіткої експертної системи, яка повинна допомогти користувачеві з відповіддю на питання: скільки дати «на чай» офіціантові за обслуговування в ресторані? (Припустимо, мова йде про місце, де такі чайові прийняте давати, наприклад, у ресторанах Парижа або Ріо-де-Жанейро).

Ґрунтуючись на звичаях і інтуїтивних представленнях, приймемо, що задача про чайові може бути описана наступними пропозиціями:

1. Якщо обслуговування погане або їжа підгоріла, то чайові — **малі**.
2. Якщо обслуговування гарне, то чайові — **середні**.
3. Якщо обслуговування відмінне або їжа чудова, то чайові — **щедрі**.

Якість обслуговування і їжі будемо оцінювати за 10-бальною шкалою (0 – найгірша оцінка, 10 – найкраща).

Припустимо, що малі чайові складають близько 5% від вартості обіду, середні – близько 15%, щедрі – приблизно 25%.

Представленої інформації, в принципі, досить для проектування нечіткої експертної системи. Така система буде мати 2 входи (які умовно можна назвати «сервіс» і «їжа»), один вихід («чайові»), три правила типу «якщо ... то» (відповідно до трьох приведених пропозицій) і по три значення (відповідно: 0 балів, 5 балів, 10 балів і 5%, 15%, 25%) для центрів функцій належності входів і виходу. Побудуємо дану систему, використовуючи алгоритм висновку Mamdani.

1. Командою **fuzzy** запускаємо FIS-редактор. За замовчуванням, алгоритм висновку – типу Mamdani (про що говорить напис у центральному блоці). Змін вносити не потрібно, але в системі повинно бути два входи, тому через пункт меню **Edit/Add input** додаємо в систему другий вхід (у вікні редактора з'являється другий жовтий блок з ім'ям **input2**). Роблячи далі одноразовий клік лівою кнопкою миші по блоці **input1**, змінюємо його ім'я на «**Service**», завершуючи введення нового імені натисканням клавіші **Enter**. Аналогічним чином встановлюємо ім'я «**Food**» блокові **input2** і «**tips**» – вихідному блокові (праворуч угорі) **output1**. Присвоюємо відразу ж і ім'я всій системі, наприклад, «**TIP**» (по-англійському це і є чайові), виконавши пункт меню **File/Save to workspace as...** Вигляд вікна редактора після зазначених дій приведено на рисунку 1.8.

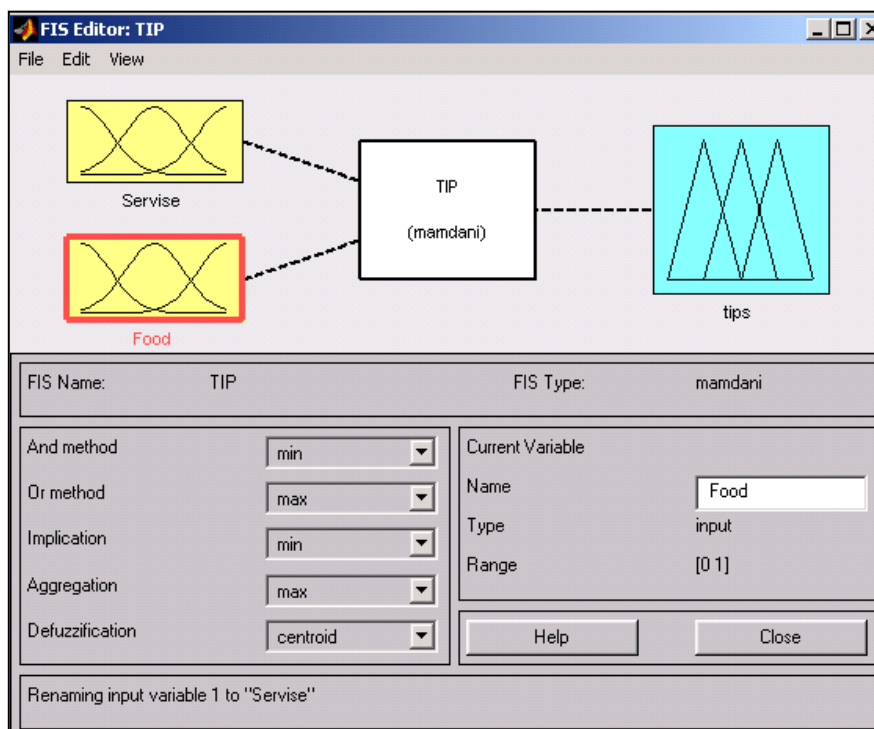


Рис. 1.8. Вигляд вікна FIS-редактори після задання структури системи

2. Задамо тепер функції належності змінних. Програму-редактор функцій належності можна відкрити трьома способами:

- через пункт меню **View/Edit membership functions...**,
- подвійним кліком лівої кнопки миші по іконці,
- натисканням клавіш **Ctrl+2**.

Задання і редагування функцій належності почнемо зі змінної «**Service**». Спочатку в полях **Range** і **Display Range** встановимо діапазон зміни і відображення цієї змінної – від 0 до 10 (балів), підтверджуючи введення натисканням клавіші **Enter**. Потім через пункт меню **Edit/Add MFs** перейдемо до діалогового вікна, де задамо функції належності гауссівського типу (**gaussmf**) із загальним числом **3**. Натиснемо кнопку **OK** і повернемося у вікно редактора функцій належності. Не змінюючи розмах і положення заданих функцій, замінимо тільки їхні імена на «**bad**», «**normal**» і «**excellent**» (як у пункті 5 попереднього прикладу).

Кліком лівої кнопки миші по іконці «**Food**» ввійдемо у вікно редагування функцій належності цієї змінної. Задамо спочатку діапазон її зміни від 0 до 10, а потім задамо дві функції належності трапецеїдної форми з параметрами, відповідно: $[0\ 0\ 1\ 3]$ і $[7\ 9\ 10\ 10]$ і іменами «**bad**» і «**nice**».

Для вихідної змінної «**tips**» вкажемо спочатку діапазон зміни – від 0 до 30, потім задамо три функції належності трикутної форми з іменами «**small**», «**middle**», «**big**» (рис. 1.9). Можна, зрозуміло, задати інші функції або вибрати для них інші параметри.

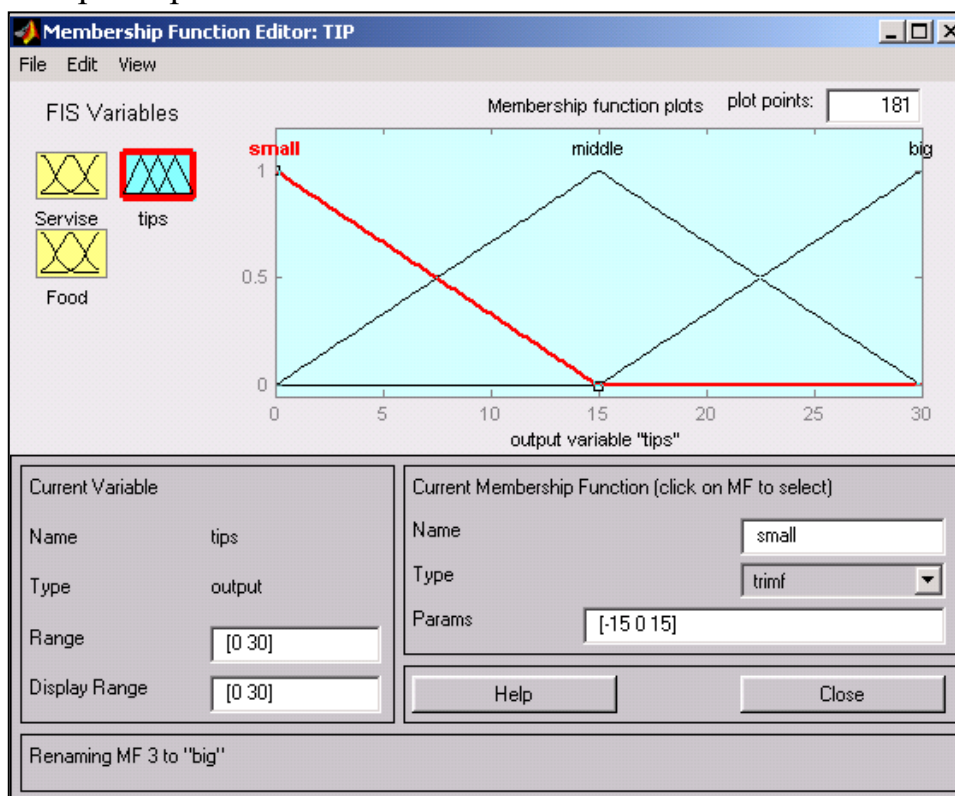


Рис. 1.9. Функції належності змінної «tips»

3. Перейдемо до конструювання правил. Для цього виберемо пункт меню **View/Edit rules...** Далі введення правил здійснюється так же, як у пункті 9

попереднього прикладу. Відмітимо, що в першому і третьому правилах у якості «зв'язування» у передумовах правила необхідно використовувати не «І» (and), а «АБО» (or); при введенні другого правила, де відсутня змінна «Food», для неї вибирається опція **none**. Підсумковий набір правил відображено на рисунку 1.10:

1. *If (Service is bad) or (Food is bad) then (tips is small) (1)*
2. *If (Service is normal) then (tips is middle) (1)*
3. *If (Service is excellent) or (Food is nice) then (tips is big) (1)*

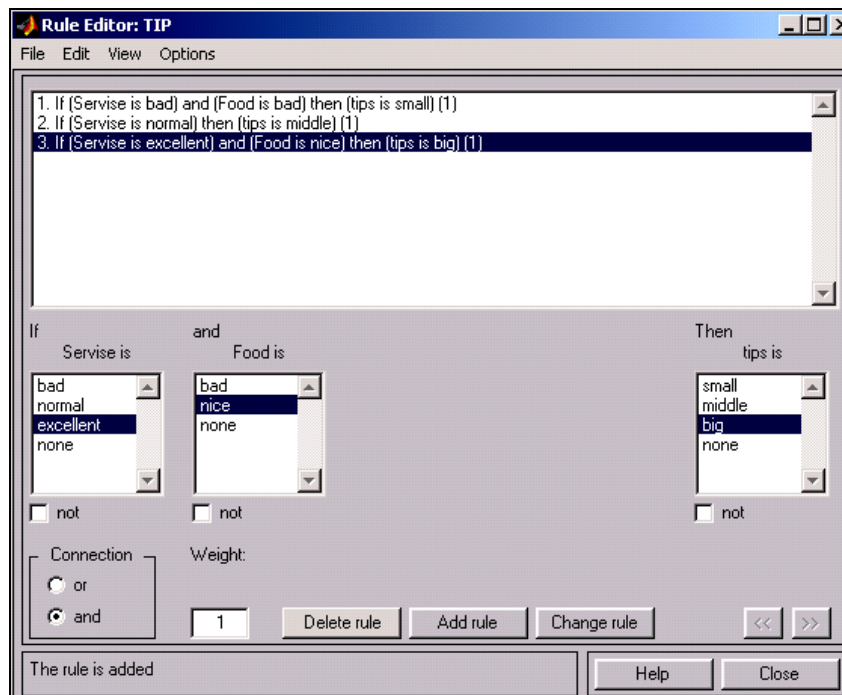


Рис. 1.10. Підсумковий набір правил

Запис представляється досить зрозумілим: одиниця в дужках після кожного правила вказує «вагу» (**Weight**), тобто значимість правила. Вагу можна змінювати, використовуючи відповідне поле в лівій нижній частині вікна редактори правил. Правила представляються і в інших формах: символічній (**symbolic**) і індексній (**indexed**), при цьому перехід від однієї форми до іншої відбувається через опції пункту меню редактори правил **Options/Format**.

На цьому конструювання експертної системи закінчено.

4. Відкриємо (через пункт меню **View/View rules...**) вікно перегляду правил і встановимо значення змінних (рис. 1.11): Service = 0 (тобто нікуди не придатний), Food = 10 (тобто чудова). Побачимо відповідь: tips = 15 (середні).

Підтвердженням відзначеної залежності вихідної змінної від вхідних може слугувати вигляд поверхні відгуку, що представляється при виборі пункту меню **View/View surface** (рис. 1.12); зверніть увагу, що за допомогою мишки графік можна повертати.

5. У вікні, що відкрилося, змінюючи імена змінних у полях введення X(input) і Y(input), можна задати і перегляд одномірних залежностей.

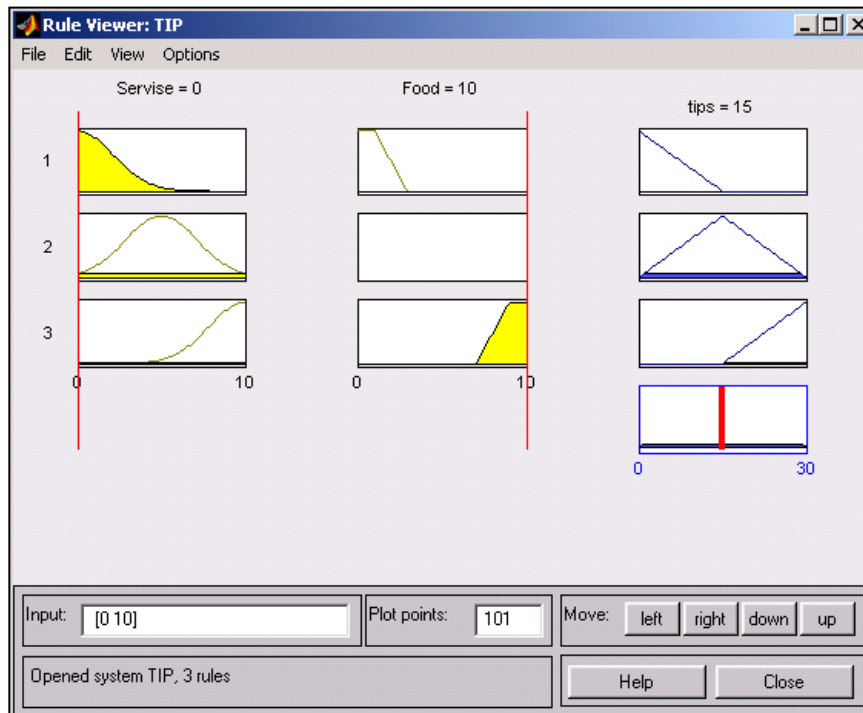


Рис. 1.11. Вікно перегляду правил

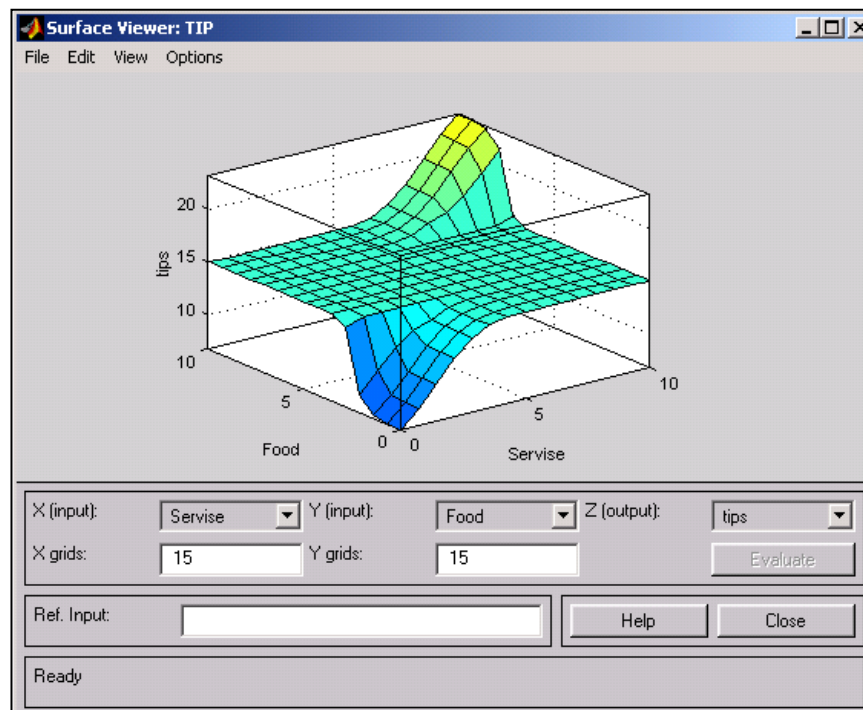


Рис. 1.12. Графічний вигляд залежності вихідної змінної від вхідних

Очевидно, із графічної залежності, що система надто негнучка. Цього можна досягнути шляхом корекції вигляду функцій належності вхідних та вихідної змінних.

Завдання

1. Реалізувати наведений приклад.
2. Досягнути більшої гнучкості нечіткої експертної системи «нарахування чайових».
3. Згідно з завданням викладача реалізувати приклад побудови нечіткої експертної системи.

1.4. Створення власних функцій належності. Якщо з якихось причин вас не влаштовує жодна з вбудованих функцій належності, ви можете створити і використовувати власну функцію. Така функція повинна бути створена як M-файл зі значеннями від 0 до 1 і з числом аргументів не більшим за 16.

Етапи створення такої функції під деяким ім'ям **fun**.

1. Створюється відповідний M-файл з ім'ям **fun.m**.
2. Вибирається пункт **Edit/Add custom MF** (Редагування/Додати власну функцію належності) у меню редактора функцій належності.
3. У поле **M-File function name** діалогового вікна, що з'являється, **Add customized membership function** вводиться ім'я створеного нового M-файлу (**fun**).
4. У поле **Parameter list** даного вікна вводяться необхідні числові параметри.
5. Нарешті, у поле **MF name** (Ім'я функції належності) вводиться ім'я функції, що задається (наприклад, **fun**).
6. Зазначене введення підтверджується натисканням кнопки **OK** (рис. 1.14).

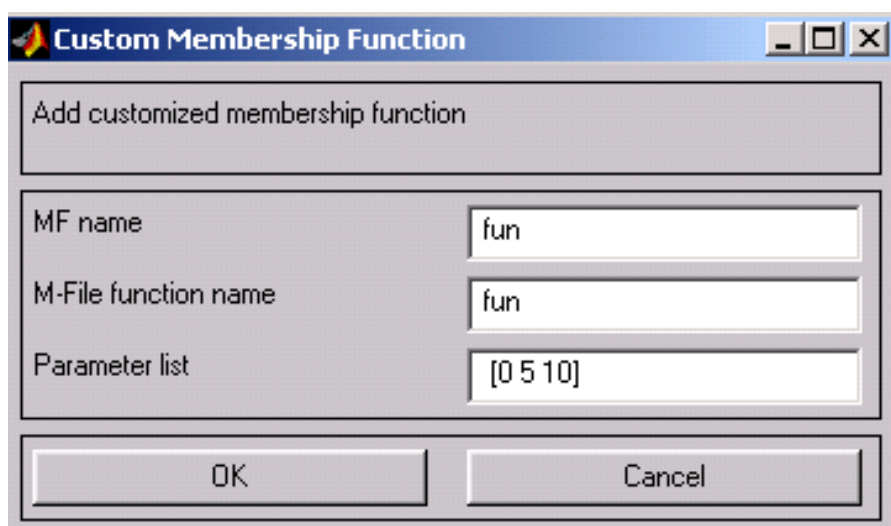


Рис. 1.14. Вікно завдання функції приналежності користувача

Нижче приведено приклад M-файлу деякої функції належності дискретного типу, що має ім'я **testmfl**, залежної від 8 числових параметрів (кожний — з діапазону [0 101]):

```

function out = testmf1(x, params)
for i = 1:length(x)
if x(i)<params(1)
y(i) = params(1);
elseif x(i)<params(2)
y(i) = params(2);
elseif x(i)<params(3)
y(i) = params(3);
elseif x(i)<params(4)
y(i) = params(4);
elseif x(i)<params(5)
y(i) = params(5);
elseif x(i)<params(6)
y(i) = params(6);
elseif x(i)<params(7)
y(i) = params(7);
elseif x(i)<params(8)
y(i) = params(8);
else
y(i) = 0;
end
end
out =.1*y'

```

Завдання

1. Реалізувати наведений приклад.
2. Згідно з завданням викладача створити власну функцію належності.

2. Робота Fuzzy Logic Toolbox у режимі командного рядка

2.1. Можливості роботи в режимі командного рядка. Пакет Fuzzy Logic має у своєму розпорядженні великий набір функцій, які виконуються з командного рядка MATLAB і дозволяють, в принципі, не використовувати при роботі із системами нечіткого висновку розглянуті програми графічного інтерфейсу. Усі такі функції діляться на групи:

1. виклику програм графічного інтерфейсу;
2. задання функцій належності;
3. створення, редагування, перегляду, відкриття і збереження систем нечіткого висновку;
4. додаткові;
5. виклику діалогових вікон інтерфейсу;
6. виклику блоків Simulink;
7. демонстрації можливостей пакета.

2.2. Функції виклику програм графічного інтерфейсу. До цієї групи відносяться функції:

Fuzzy – виклик FIS-редактора;

Mfedit – виклик редактора функцій належності;

Ruleedit – виклик редактора правил;

Ruleview – виклик програми перегляду правил;

Surfview – виклик програми перегляду поверхні відгуку;

Anfisedit – виклик ANFIS-редактора (тільки для систем, що використовують алгоритм Sugeno і мають одну вихідну змінну);

Findcluster – виклик програми кластеризації.

Використання перших шести функцій з аргументом (наприклад, **fuzzy (a)**, де **a** — ім'я змінної робочого простору, присвоєне системі нечіткого висновку), відкриває відповідну програму з одночасним завантаженням у неї розглянутої системи.

Функція **findcluster (ім'я файлу)** відкриває програму кластеризації з одночасним завантаженням зазначеного файлу даних.

2.3. Задання функцій належності. У дану групу включені 11 функцій (за числом функцій належності, використовуваних у пакеті Fuzzy Logic).

1. Функція **dsigmf**.

Запис: $y = \text{dsigmf}(x, [a1 \ c1 \ a2 \ c2])$

Опис. Задається функція належності, обумовлена як різниця двох сигмоїдальних функцій. Сигмоїдальна функція, як відомо, описується:

$$f = (x, a, c) = \frac{1}{1 + \exp(-a(x - c))} \quad (2.1)$$

і залежить від двох числових параметрів a і c . Описувана функція, як відзначено, є різницею двох сигмоїдальних:

$$f_1(x, a_1, c_1) - f_1(x, a_2, c_2), \quad (2.2)$$

і залежить від чотирьох параметрів a_1, c_1, a_2, c_2 .

Приклад

```
>> x=0:0.1:10;
```

```
>> y=dsigmf(x,[5 2 5 7]);
```

```
>> plot(x,y)
```

```
>> xlabel('dsigmf, P=[5 2 5 7]') (рис.2.1)
```

2. Функція **gauss2mf**.

Запис: $y = \text{gauss2mf}(x, [\text{sig1} \ c1 \ \text{sig2} \ c2])$

Опис. Задається функція належності, що є різницею двох гауссівських функцій, обумовлена співвідношенням:

$$f(x, \sigma_1, c_1, \sigma_2, c_2) = \exp(-(x - c_1)^2 / \sigma_1^2) - \exp(-(x - c_2)^2 / \sigma_2^2), \quad (2.3)$$

і залежну від чотирьох параметрів a_1, c_1, a_2, c_2 .

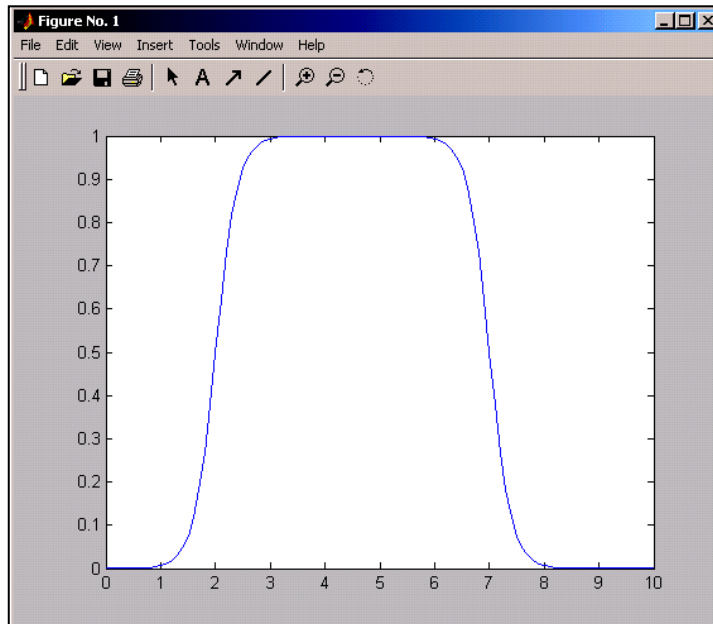


Рис. 2.1. Вигляд функції `dsigmf(x,[5 2 5 7])`

Приклад

```
>> x=(0:0.1:10)';
>> y1=gauss2mf(x,[2 4 1 8]);
>> y2=gauss2mf(x,[2 5 1 7]);
>> y3=gauss2mf(x,[2 6 1 6]);
>> y4=gauss2mf(x,[2 7 1 5]);
>> y5=gauss2mf(x,[2 8 1 4]);
>> plot(x,[y1 y2 y3 y4 y5]); (рис. 2.2)
```

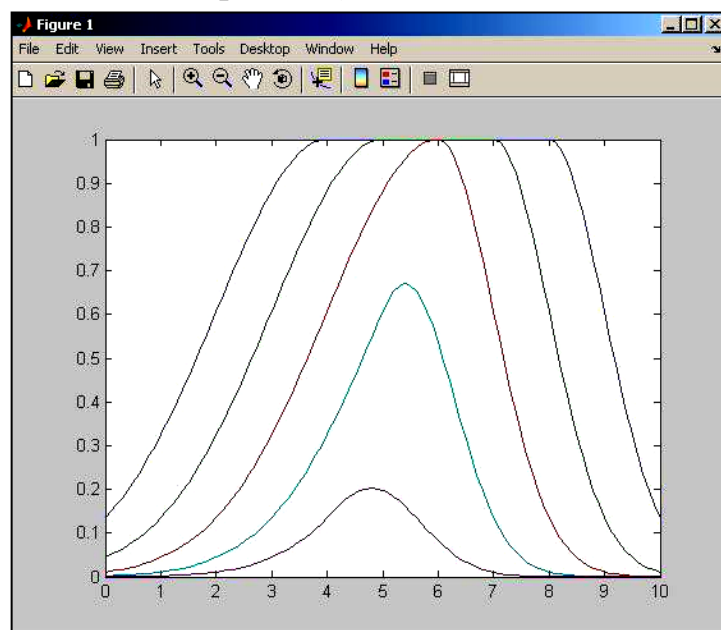


Рис. 2.2. Сімейство кривих, заданих функцією `gauss2mf`

3. Функція `gaussmf`.

Запис: $y = \text{gaussmf}(x, [\text{sig } c])$.

Опис. Задається функція належності гауссівського типу, що залежить від двох параметрів ([sig c]).

Приклад

```
>> x=(0:0.1:10);  
>> y=gaussmf(x,[2 5]);  
>> plot(x,y)  
>> xlabel('gaussmf, P=[2 5]') (рис. 2.3)
```

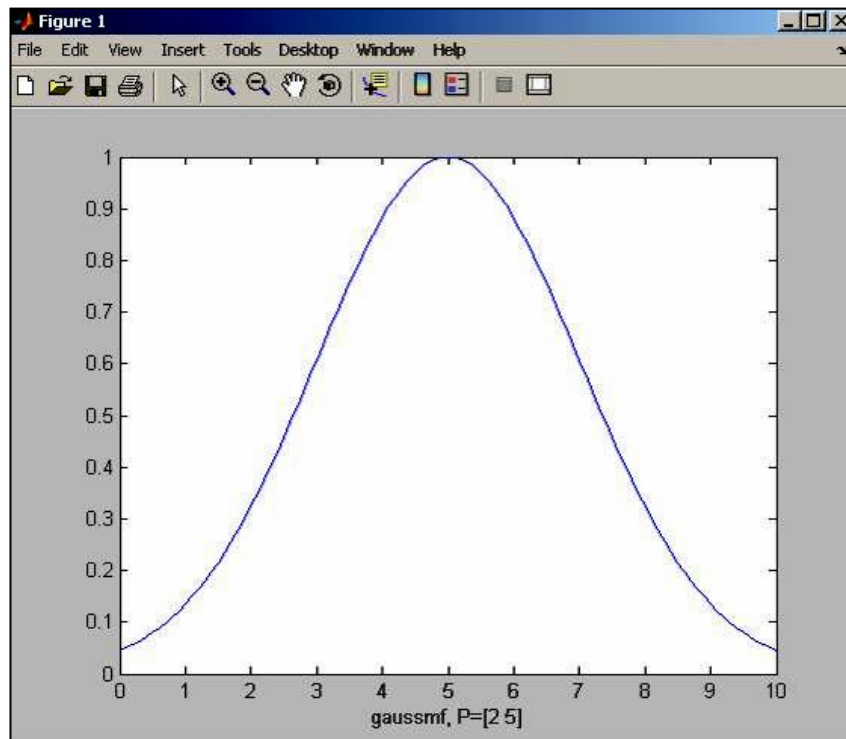


Рис. 2.3. Функція належності типу гауссівської кривої

4. Функція gbellmf.

Запис: $y = \text{gbellmf}(x, \text{params})$

Опис. Задається функція належності узагальненого звоноподібного типу з аналітичним описом, що залежить від трьох числових параметрів:

$$(x, a, b, c) = \frac{1}{1 + |(x - c) / a|^{2b}}. \quad (2.4)$$

Приклад

```
>> x=(0:0.1:10);  
>> y=gbellmf(x,[2 4 6]);  
>> plot(x,y)  
>> xlabel('gbellmf, P=[2 4 6]') (рис. 2.4)
```

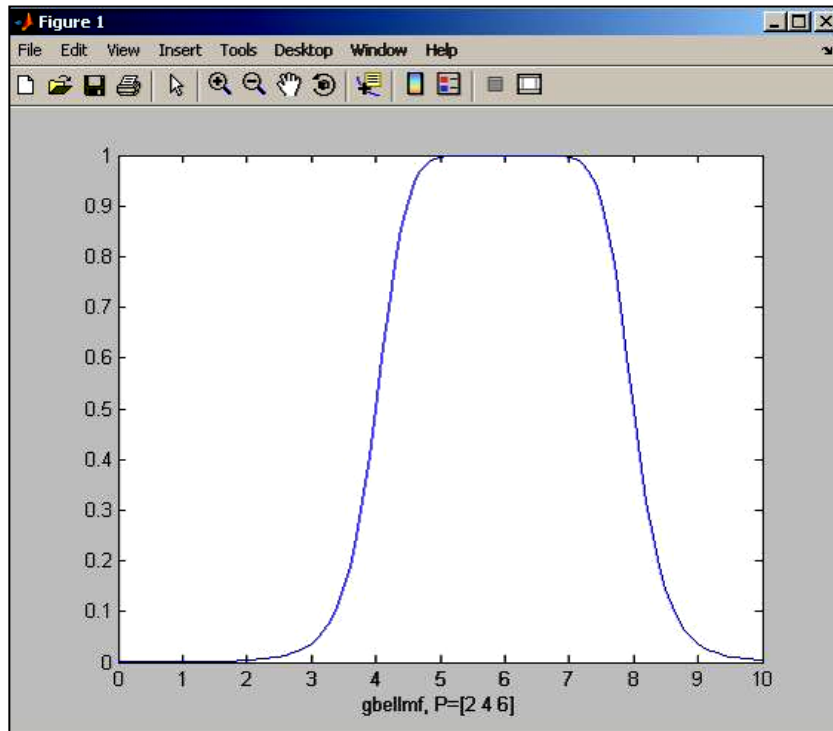


Рис. 2.4. Графік узагальненої звоноподібної функції

5. Функція **pimf**.

Запис: $y = \text{pimf}(x, [a \ b \ c \ d])$

Опис. Задається π -подібна функція належності, яка одержала свою назву через своєрідну форму. Функція обчислюється з використанням сплайн апроксимації за чотирьма точками, що задається вектором параметрів. Параметри a і d визначають основу кривої, а параметри b і c – положення плоскої вершини.

Приклад

```
>> x=(0:0.1:10);
>> y=pimf(x,[1 4 5 10]);
>> plot(x,y)
>> xlabel('pimf, P=[1 4 5 10]') (рис. 2.5)
```

6. Функція **psigmf**.

Запис: $y = \text{psigmf}(x, [a1, c1, a2, c2])$

Опис. Задається функція належності, добуток двох сигмоїдальних функцій, залежна від чотирьох параметрів **a1, c1, a2, c2**.

Приклад

```
>> x=(0:0.1:10);
>> y=psigmf(x,[2 3 -5 8]);
>> plot(x,y)
>> xlabel('psigmf, P=[2 3 -5 8]') (рис. 2.6)
```

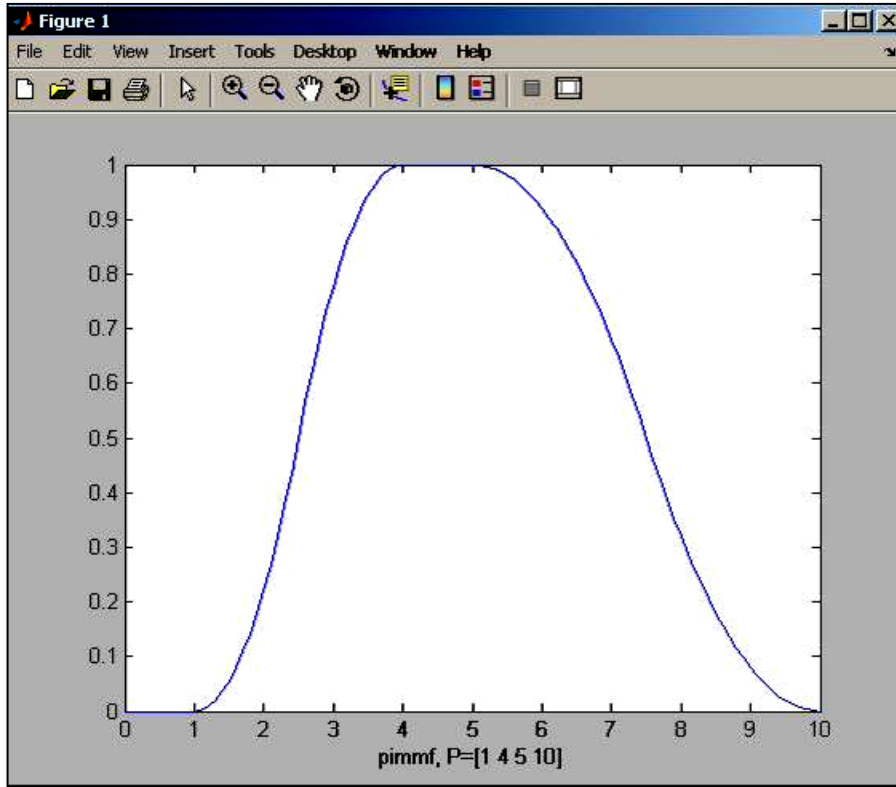


Рис. 2.5. Графік π -подібної функції

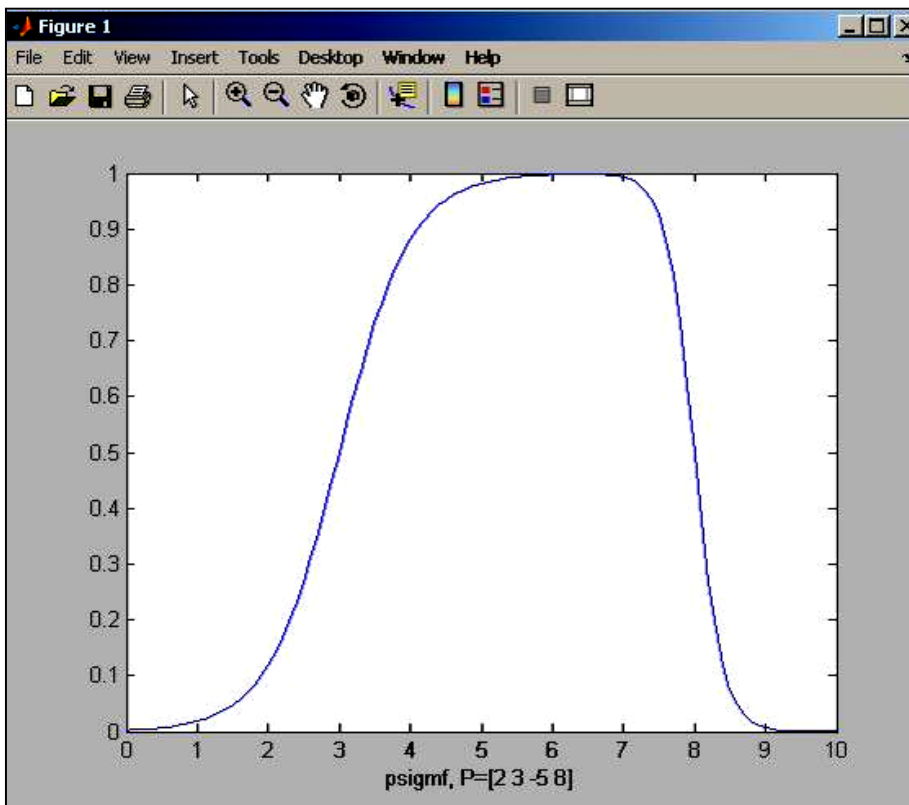


Рис. 2.6. Приклад використання функції **psignif**

7. Функція **smf**.

Запис: $y = \text{smf}(x, [a \ b])$.

Опис. Задається залежна від двох параметрів S-подібна функція належності. Параметри a і b визначають діапазон значень аргументу, де функція зростає.

Приклад

```
>> x=(0:0.1:10);  
>> y=smf(x,[1 8]);  
>> plot(x,y)  
>> xlabel('psigmf, P=[1 8]') (рис. 2.7)
```

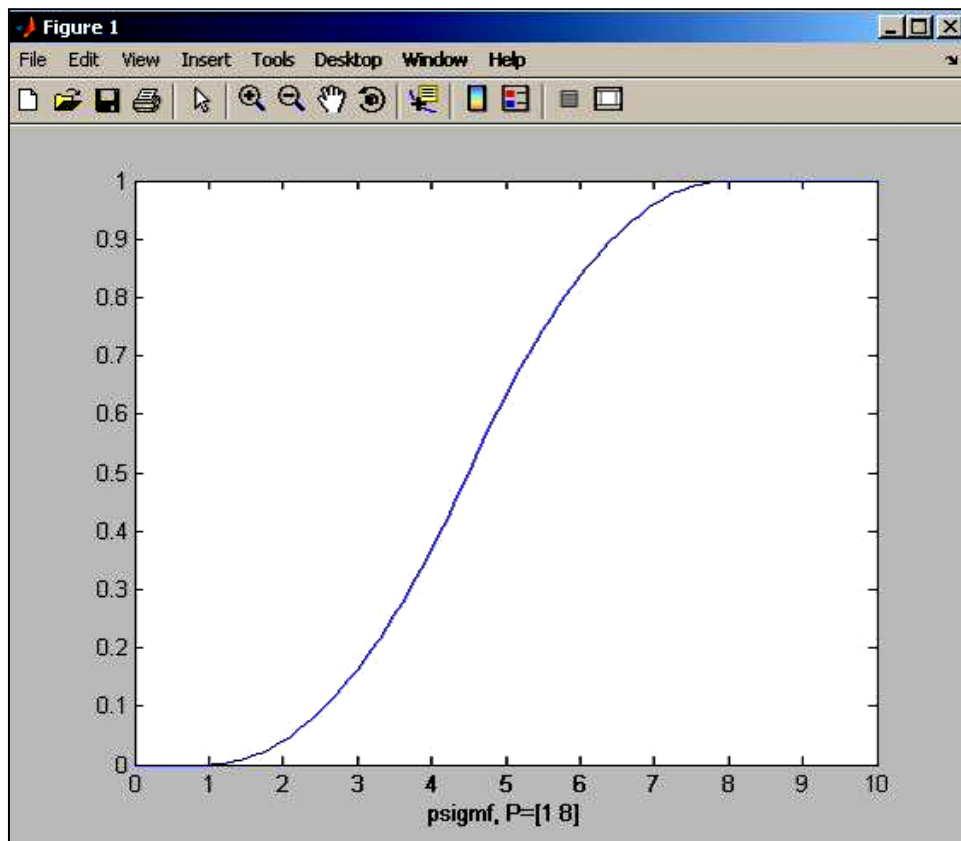


Рис. 2.7. Графік S-подібної функції

8. Функція **sigmf**.

Запис: $y = \text{sigmf}(x, [a \ c])$.

Опис. Задається сигмоїдальної функція належності, залежить від двох параметрів.

Приклад

```
>> x=(0:0.1:10);  
>> y=sigmf(x,[2 4]);  
>> plot(x,y)
```

```
>> xlabel('sigmf, P=[2 4]') (рис. 2.8)
```

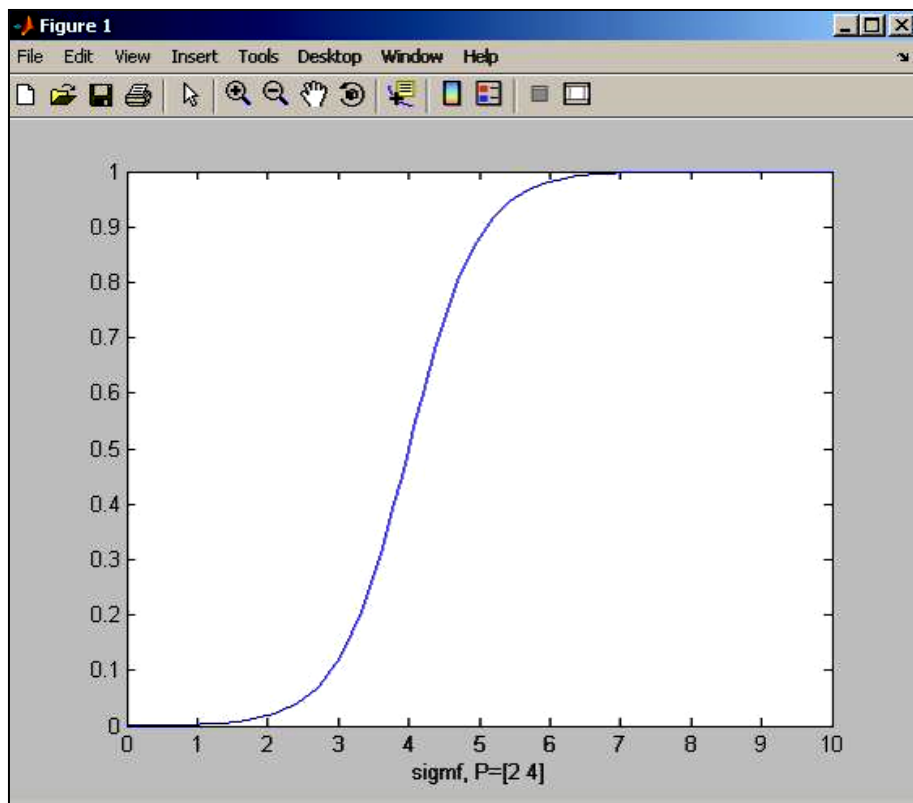


Рис. 2.8. Графік сигмоїдної функції

9. Функція **trapmf**.

Запис: **y = trapmf(x,[a b c d])**.

Опис. Задається трапецеїдальна функція належності, яка залежить від чотирьох параметрів і визначена формулою:

$$f(x, a, b, c, d) = \left. \begin{array}{l} 0, x \leq a \\ \frac{x-a}{b-a}, a \leq x \leq b \\ 1, b \leq x \leq c \\ \frac{d-x}{d-c}, c \leq x \leq d \\ 0, d \leq x \end{array} \right\}, \quad (2.5)$$

при цьому параметри a і d визначають основу кривою, b і c – положення вершини.

Приклад

```
>> x=(0:0.1:10);  
>> y=trapmf(x,[1 5 7 8]);  
>> plot(x,y)
```

>> xlabel('trapmf, P=[1 5 7 8]') (рис. 2.9)

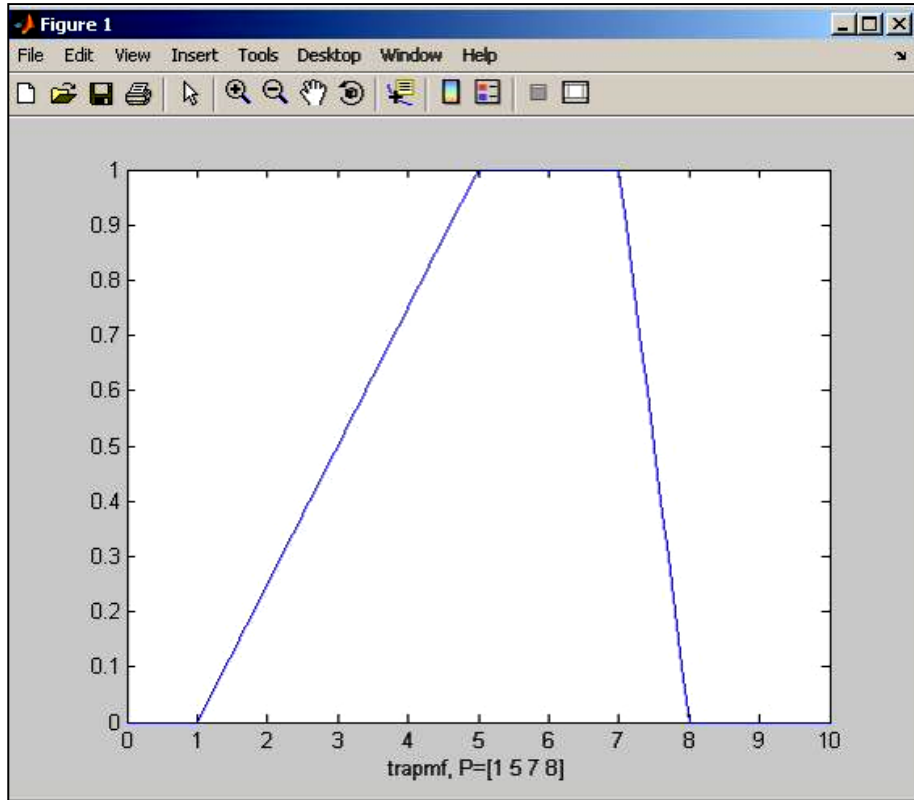


Рис. 2.9. Графік трапецеїдальної функції

Завдання

1. Реалізувати наведені приклади.
2. Згідно з завданням викладача реалізувати функцію належності.

2.3. Функції збереження, відкриття і використання створеної системи.

Читання, використання і збереження на диску створеної системи нечіткого висновку в режимі командного рядка здійснюється функціями:

```
readfis('ім'я_файлу'),  
evalfis(вектор параметрів, ім'я),  
writefis(HMfl) або writefis(ім'я, 'ім'я_файлу')
```

Тут **ім'я_файлу** – найменування файлу з записаною системою (без вказівки розширення), **ім'я** – ідентифікатор, що визначений системі в робочому середовищі MATLAB, **вектор параметрів** – набір значень входів, для яких потрібно розрахувати вихід (можливо, матриця параметрів, тоді результат розрахунків – вектор у випадку однієї вихідної змінної або матриця, при декількох таких змінних).

Приклади (стосовно до раніше створеного і збереженого на диску файлу з ім'ям **tip**):


```

>> a=readfis('TIP');
>> evalfis([1 2], a)
ans =
    9.3281
<< writefis(a)
ans =
    TIP

```

Завдання

1. Реалізувати наведені приклади.
2. Згідно з завданням викладача активувати fis-файл.

2.4. Функції використання графічного вікна. Наступні три функції дозволяють використовувати елементи графічних зображень поза програмами з графічним інтерфейсом.

1. Команда (функція) **plotfis(a)** викликає появу графічного вікна MATLAB із мнемонічним представленням розробленої системи (рис. 2.10).

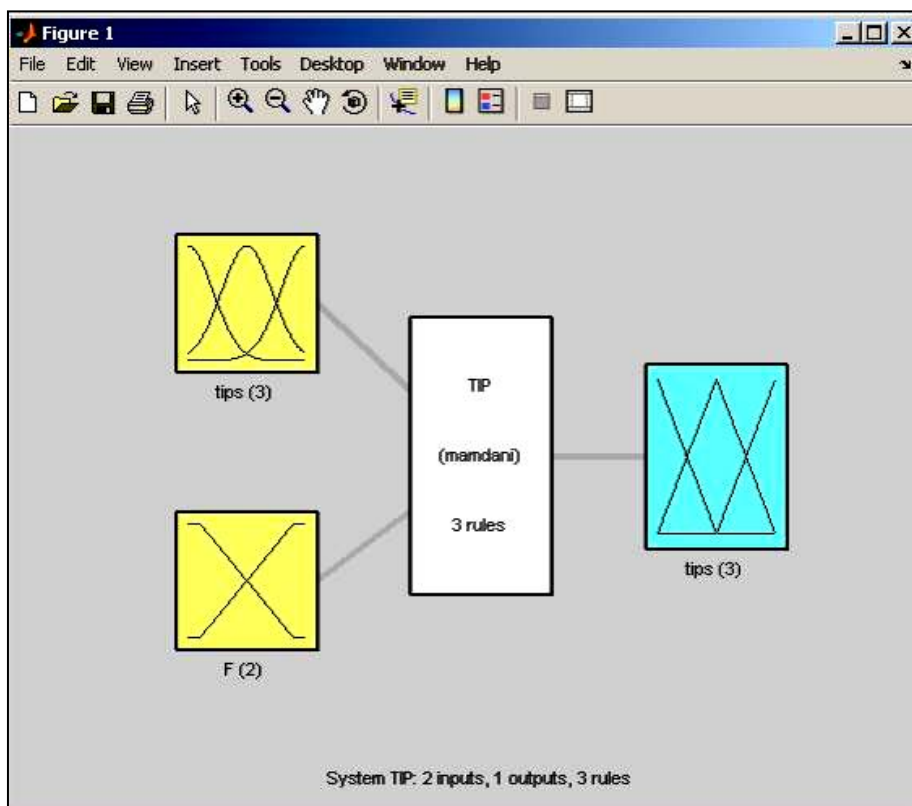


Рис. 2.10. Мнемонічне представлення системи нечіткого висновку

2. Команда (функція) **plotmf** виконує аналогічну операцію, але стосовно графіків функцій належності. На рисунку 2.11. приведено результат виконання функції **plotmf(a, 'input', 1)**.

3. Нарешті, команда (функція) **gensurf(a)** робить те ж саме, але стосовно поверхні відгуку (рис. 2.12).

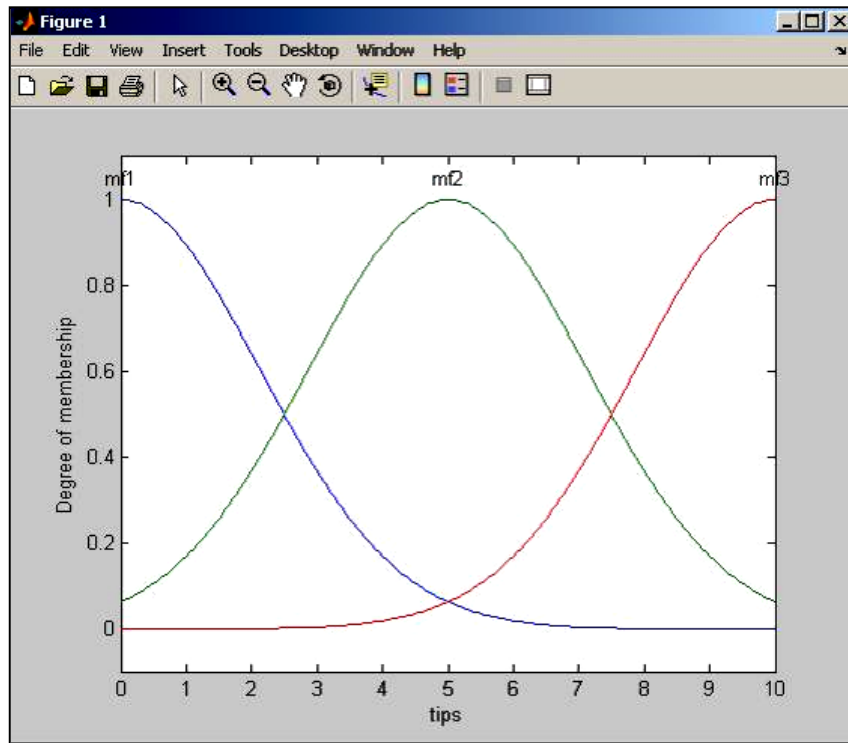


Рис. 2.11. Результат виконання функції **plotmf(a 'input', 1)**

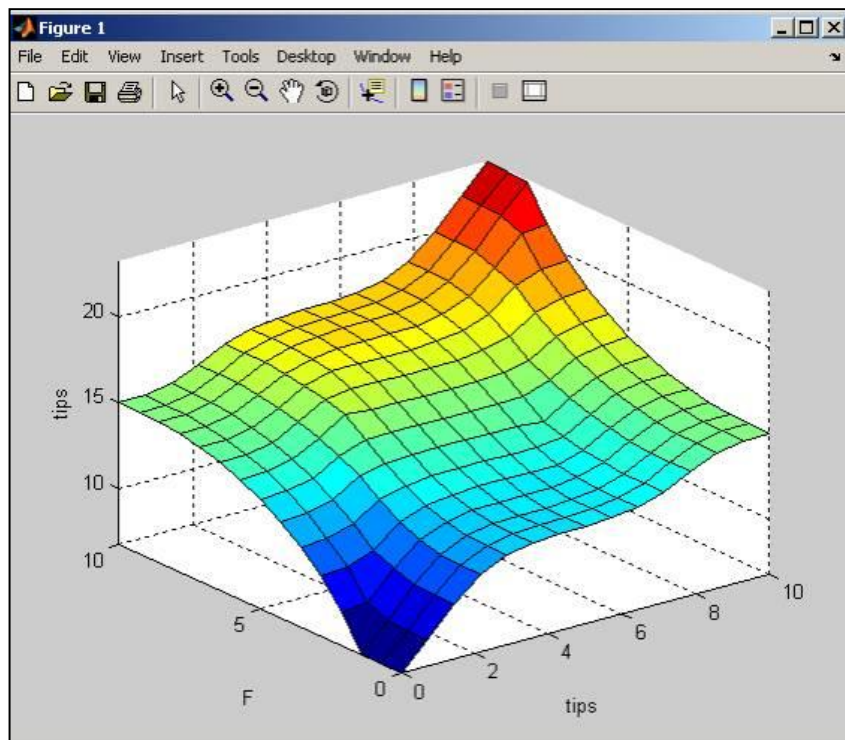


Рис. 2.12. Результат виконання функція **gensurf(a)**

2.5. Функції створення, перегляду структури і редагування систем нечіткого висновку. Взагалі, при бажанні можна зовсім обійтися без програм графічного інтерфейсу і, використовуючи функції **newfis** (нова система), **addvar** (додати змінну), **addmf** (додати функцію належності) і **addrule** (додати правило), сконструювати систему нечіткого висновку у режимі командного рядка MATLAB. Процес цей, треба сказати, більш трудомісткий, чим із застосуванням зазначених програм.

При необхідності після перегляду елементів структури (системи з ідентифікатором **a**) у режимі командного рядка варто ввести команду виду **a.мітка**, наприклад:

```
>> a.type
    одержимо відповідь:
>> a.type

ans =
    mamdani
```

Одержання інформації з всіх елементів структури забезпечується функцією **getfis(a)**. Результат її виконання:

```
>> getfis(a)
Name      = TIP
Type      = mamdani
NumInputs = 2
InLabels  =
    tips
    F
NumOutputs = 1
OutLabels =
    tips
NumRules  = 3
AndMethod = min
OrMethod  = max
ImpMethod = min
AggMethod = max
DefuzzMethod = centroid

ans =
    TIP
```

Повний перегляд структури системи нечіткого висновку здійснюється функцією **showfis(a)**.

Перегляд правил, включених у систему нечіткого висновку, здійснюється за допомогою функції **showrule**.

Опис. У функції **ім'я** – це ідентифікатор розглянутої системи в середовищі MATLAB, **номера_правил** – список правил, які потрібно показати, формат – змінна, що має значення **'verbose'**, **'symbolic'** або **'indexed'**, **мова** – змінна із

значеннями 'english', 'français' або 'deutsch' (налаштування за замовчуванням – 'verbose' і 'english').

Приклад

```
>> a=readfis('TIP');
>> showrule(a,1)
ans =
1. If (tips is mf1) or (F is mf1) then (tips is mf1) (1)
>> showrule(a,2)
ans =
2. If (tips is mf2) then (tips is mf2) (1)
>> showrule(a,3)
ans =
3. If (tips is mf3) or (F is mf2) then (tips is mf3) (1)
```

Функція **rmmf** використовується для видалення функції належності зі складу системи.

Опис. Параметри функції: **ім'я** – ідентифікатор системи в середовищі MATLAB, **varType** – змінна зі значеннями 'input' або 'output', **varIndex** – порядковий номер змінної (за списком змінних входу або виходу), **mf** – змінна, що задає функцію належності, **mfIndex** – порядковий номер даної функції.

Функція **rmvar** видаляє змінну зі складу системи.

Опис. Тут змінні **varType** і **varIndex** мають той же зміст, що й у попередній функції, змінна **errorStr** дозволяє записати будь-яке повідомлення про помилку, **нове_ім'я** — нове ім'я системи.

Функція приведення до чіткості (дефазифікація) **defuzz** дозволяє за заданою функцією належності визначити відповідне чітке значення.

Запис: **out = defuzz(x, mf, type)**

Опис. Тут **x** – позначення числового аргументу, **mf** – тип функції належності, **type** – метод приведення до чіткості.

Приклад

```
>> x=-10:0.1:10;
>> mf=trapmf(x,[-10 -8 -4 7]);
>> xx=defuzz(x, mf, 'centroid')
xx =
-3.2857
```

Функція **evalmf** обчислює значення заданої функції належності.

Запис: **y = evalmf(x, mfParams, mfType)**

Опис. Тут **x** – позначення числового аргументу, **mf-Params** – вектор необхідних числових параметрів, **mfType** – тип функції належності.

Приклад

```
>> x=-10:0.1:10;
```

```
>> mfparams=[2 4 6];
>> mftype='gbellmf';
>> y=evalmf(x, mfparams,mftype);
>> plot(x,y)
>> xlabel('gbellmf, P=[ 2 4 6]') (рис. 2.13)
```

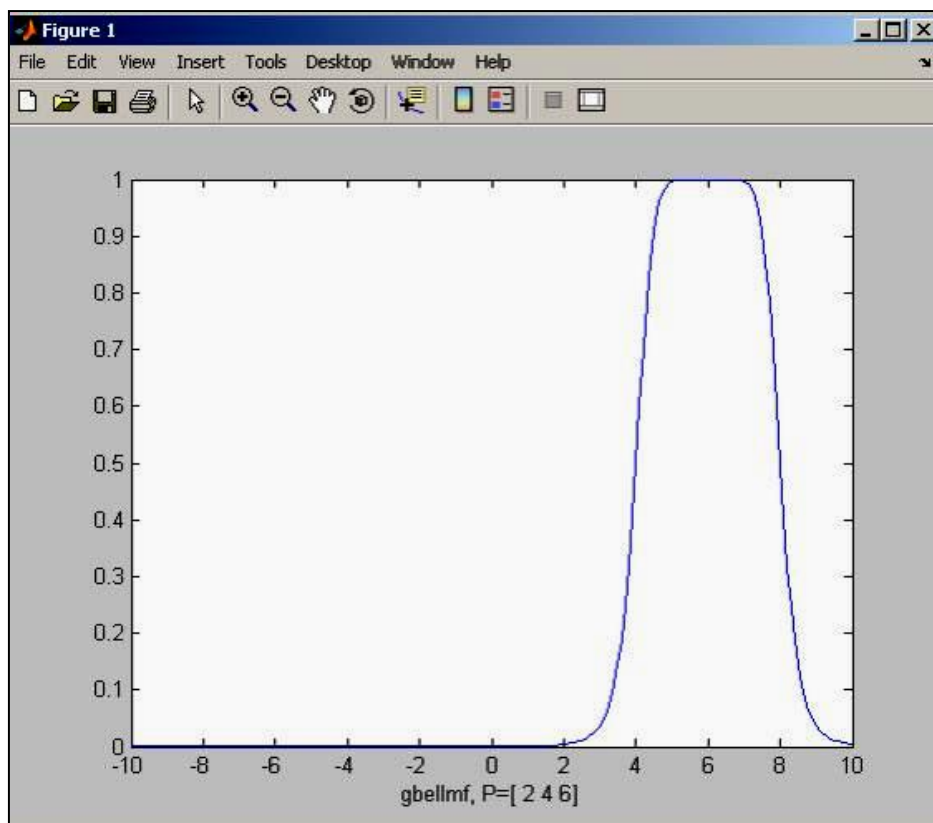


Рис. 2.13. Графічне представлення результату виконання команди **evalmf(x,mfParams,mfType)**

Функція **mftmf** дозволяє замінити одну функцію належності близькою їй за формою іншою з деяким «еквівалентним» набором числових параметрів.

Запис: **outParams = mJ2mf(inParams,inType,outType)**

Опис. **InParams** – набір параметрів вихідної функції, **inType** – тип вихідної функції належності, **outType** – тип еквівалентної функції належності, **outParams** – набір перетворених параметрів.

Приклад

```
>> x=-10:0.1:10;
>> mfp1=[1 2 3];
>> mfp2=mf2mf(mfp1,'gbellmf','trimf')
mfp2 =
    1    3    5
>> plot(x, gbellmf(x, mfp1), x, trimf(x, mfp2)) (рис. 2.14)
```

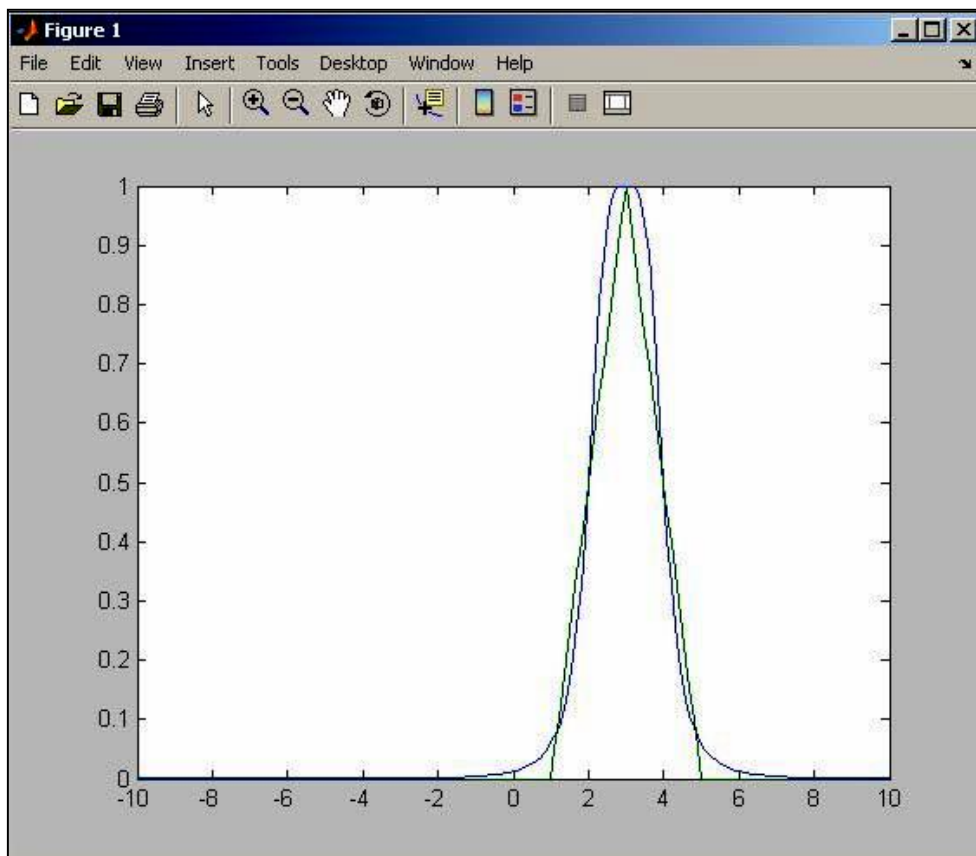


Рис. 2.14. Графіки вихідної (звоноподібної) і еквівалентної (трикутної) функцій належності

Функцію **parsrule** можна віднести до числа сервісних.

Запис: **нове_ім'я = parsrule (ім'я, текст_правил) нове_ім'я = parsrule (ім'я, текст_правил, формат) нове_ім'я = parsrule (ім'я, текст_правил, формат, мова)**

Опис. У даній функції ідентифікатори **ім'я** і **нове_ім'я** відносяться до вихідної і модернізованої систем з нечітким висновком, рядкові змінні, **формат** і **мова** мають той же зміст, що й у розглянутій раніше функції **showrule**.

Завдання

1. Реалізувати наведені приклади.
2. Згідно з завданням викладача трансформувати функції належності.

2.6. Інші функції. До цієї групи відносяться:

Convertfis – забезпечує конвертацію FIS-матриці пакета Fuzzy Logic версії 1.0 у FIS-структуру пакета Fuzzy Logic версії 2.0: **ім'я_нове = convertfis (ім'я_старе)**.

Discfis – перетворює безперервні функції належності системи нечіткого висновку в дискретні (задаються наборами точок).

Evalmmf – обчислює значення функцій належності. Є багатомірним аналогом функції **evalmf**.

Fstrvcat – поєднує задані вектори і матриці в одну матрицю.

Функція повертає матрицю Y , яка представляє собою об'єднання матриць-аргументів при цьому для вирівнювання розмірів підсумкової матриці вона доповнюється нулями. Елементи аргументів можуть бути рядкового типу, при цьому їхні символи перетворюються в числову форму.

Fuzarith – виконує алгебраїчні операції над нечіткими множинами.

Findrow – повертає номери аргументів функції **fstrvcat** за їх іменами.

Genparam – повертає параметри заданих функцій належності, визначаючи їх за набором вхідних експериментальних даних. Такі результати можуть бути використані як початкові наближення при наступному застосуванні команди **anfis**.

Mam2sug – перетворює систему нечіткого висновку з алгоритмом Mamdani у систему, що використовує алгоритм Sugeno.

Приклад

```
>> a=readfis('TIP');
>> b=mam2sug(a);
>> subplot(2,1,1);
>> gensurf(a);
>> title('Mamdani');
>> subplot(2,1,2);
>> gensurf(b);
>> title('Sugeno'); (рис. 2.15)
```

Зауваження. Перевірка показала, що розглянута функція виконується некоректно і приводить до непрацюючої системи висновку (спроби використання системи Sugeno викликають повідомлення про помилку). Даний недолік можна усунути, змінивши в передостанньому рядку файлу mam2sug.m (у директорії Matlab/toolbox/fuzzy/fuzzy/) **defuzzmethod** на **defuzzMethod**.

Probor – виконує операцію ймовірнісного АБО над стовпцями матриці-аргументу: $Y = \text{probor}(X)$

Якщо матриця-аргумент X має два стовпці, A і B , то функція повертає матрицю $Y = A + Y - AB$; якщо аргумент містить тільки один стовець, то $Y = X$.

Sugmax – дозволяє визначити максимально можливі діапазони зміни вихідних змінних у заданій системі нечіткого висновку, що використовує алгоритм Sugeno.

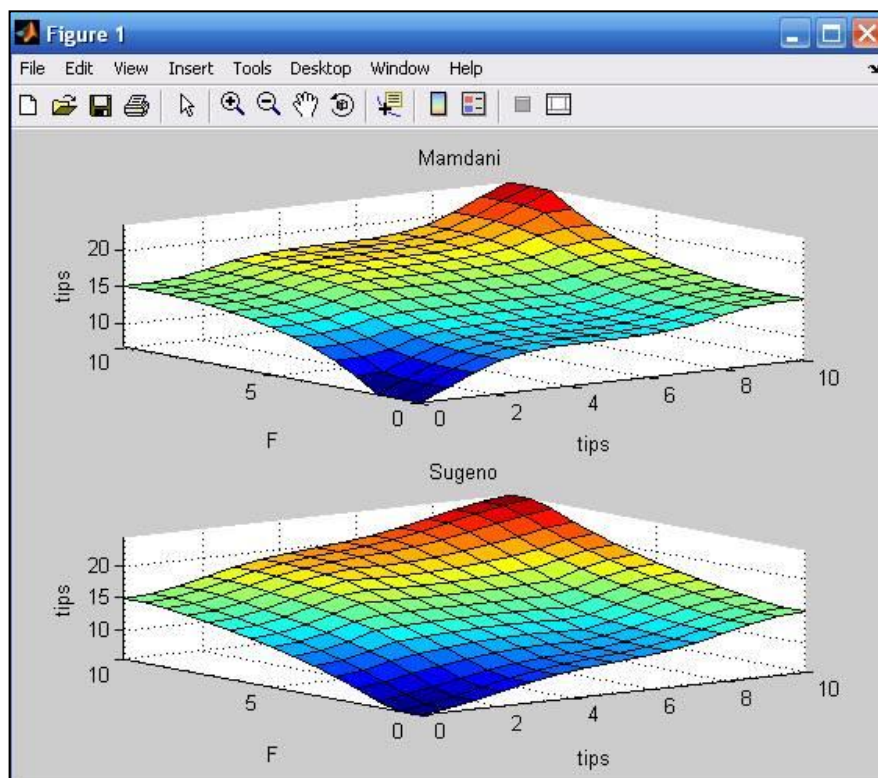


Рис. 2.15. Ілюстрація виконання функції **mam2sug**

Завдання

1. Реалізувати наведений приклад.
2. Згідно з завданням викладача реалізувати функції FIS-редактора.

2.7. Функції виклику діалогових вікон інтерфейсу. Групу утворюють 12 функцій:

Cmfdlg – викликає діалогове вікно для завдання функції належності користувача.

Cmthdlg – викликає діалогове вікно для завдання алгоритму нечіткого висновку.

Fisgui – дозволяє генерувати елементи графічного інтерфейсу користувача.

Gfmfdlg – викликає діалогове вікно для завдання числа і типу функцій належності при проектуванні системи типу Sugeno (з наступним застосуванням функції **anfis**).

Mfdlg – викликає діалогове вікно для завдання додаткової функції належності,

Mfdrag – забезпечує «пересування» функції належності з використанням миші,

Popundo – повертає зі стеку результати скасування попередньої дії,

Pushundo – обновлює в стек дані щодо скасування дії.

Savedlg – викликає діалогове вікно, що видає запит на збереження результатів.

Statmsg – розміщує повідомлення в поле стану вікна.

Updtfis – оновлює елементи графічного інтерфейсу користувача.

Wsdlg – викликає діалогове вікно для виконання оператора збереження в робочому просторі.

Як приклад, приведемо виконання функції **gfmfdlg** (рис. 2.16):

```
>> a=readfis('TIP');  
>> gfmfdlg('#init',a)
```

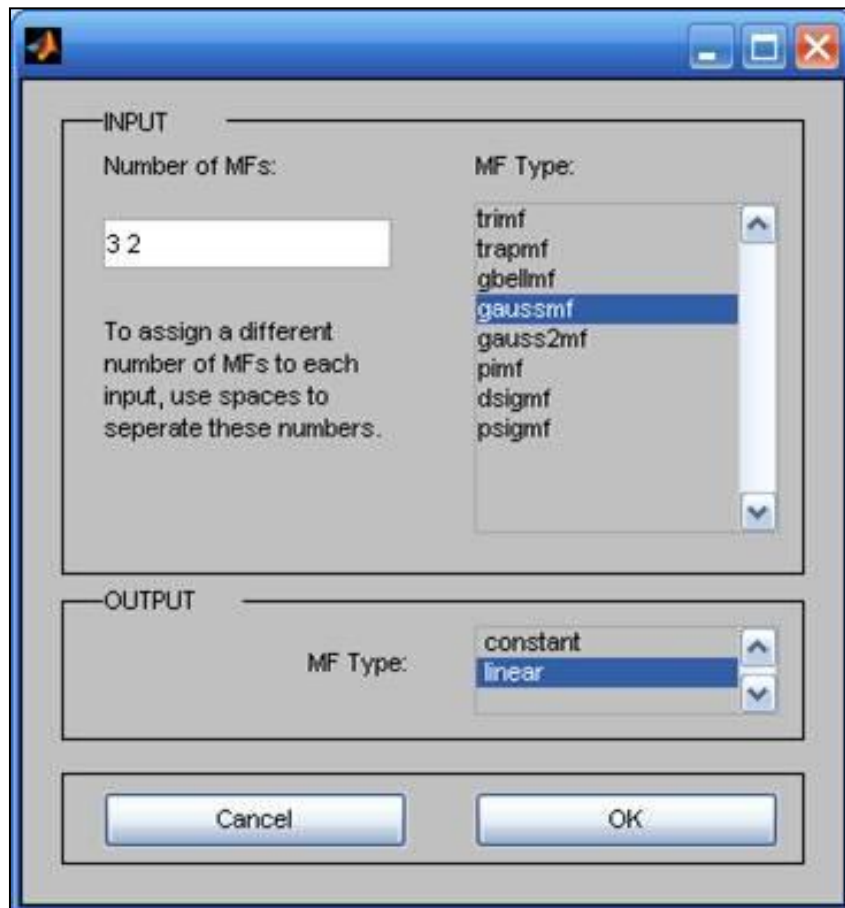


Рис. 2.16. Результат виконання команди **gfmfdlg**

Системи нечіткого висновку, створені тим або іншим способом за допомогою пакета Fuzzy Logic Toolbox, допускають інтеграцію з інструментами пакета Simulink, що дозволяє виконувати моделювання систем у рамках останнього.

Завдання

1. Реалізувати наведений приклад.
2. Згідно з завданням викладача реалізувати функції FIS-редактора виклику інтерфейсних вікон.

3. Побудова нечіткої моделі з використанням блоків Simulink

Для побудови власної моделюючої системи з використанням засобів нечіткої логіки і блоків Simulink рекомендується просто скопіювати блок Fuzzy Logic Controller з системи **sltank** (або якого-небудь іншого демонстраційного прикладу MATLAB) і помістити його в блок-діаграму розроблювальної системи. З командного рядка командою **fuzblock** (рис. 3.1) можна також відкрити бібліотеку нечітких блоків пакету Simulink, що можуть бути використані точно так само, при необхідності – з додатковим редагуванням.

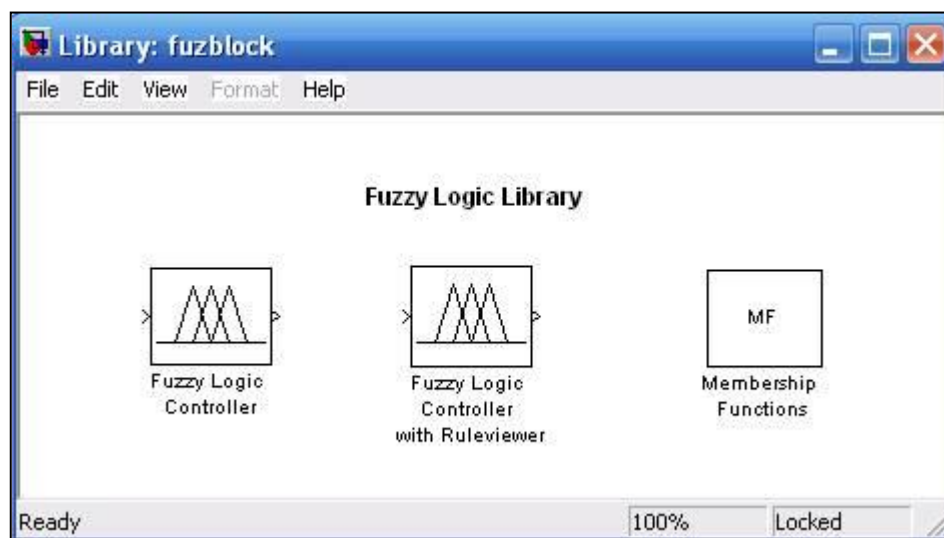


Рис. 3.1. Ілюстрація виконання команди **fuzblock**

Відзначимо, що функціонування зазначених блоків здійснюється з використанням системної S-функції **sfis.mex**. Запис цієї функції: **output = sfns(t,x,u,flag,fismat)**, де: **output** – вихід нечіткого регулятора, **t**, **x** і **flag** – стандартні аргументи системної функції Simulink, **fismat** – ідентифікатор (ім'я) нечіткої системи висновку, **i** — вхідний сигнал (вектор вхідних сигналів) регулятора.

За змістом дана функція аналогічна розглянутій функції **evalfis**, але вона оптимізована для роботи в середовищі Simulink.

Завдання

1. Реалізувати наведений приклад.
2. Згідно з завданням викладача реалізувати нечітку модель у **Simulink**.

4. Демонстраційні приклади роботи з пакетом Fuzzy Logic Toolbox

Для ознайомлення з пакетом Fuzzy Logic Toolbox можна використовувати наступні функції (команди) у режимі командного рядка:

defuzzdm – огляд методів приведення до чіткості (дефазифікації),

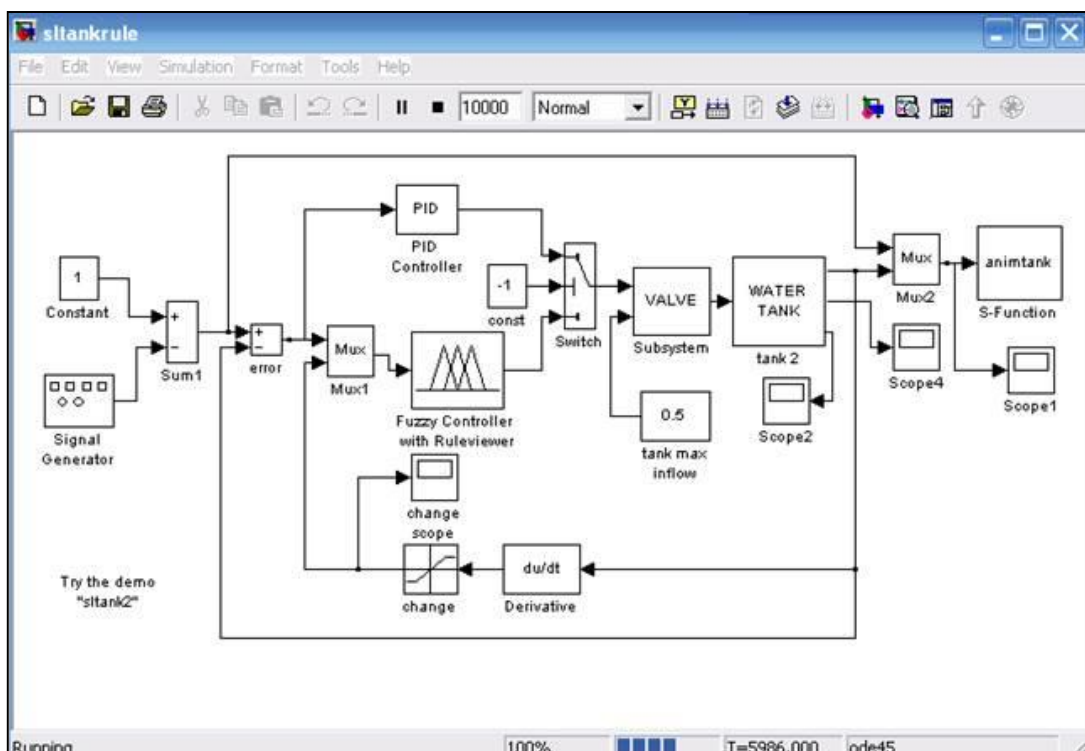
fcmdemo – демонстрація алгоритму кластеризації Fuzzy c-means (2-D графіка),

fuzdemos – демонстрація графічного інтерфейсу пакета Fuzzy Logic,
gasdemo – демонстрація використання апарату гібридних мереж для рішення задачі про вибір автомобіля, найбільш економічного за витратами палива,
juggler – демонстрація системи жонглювання м'ячем з використанням нечіткого регулятора,
invkine – демонстрація нечіткого керування рухом робота-маніпулятора,
irisfcm – демонстрація алгоритму кластеризації Fuzzy c-means,
noisedm – демонстрація рішення задачі фільтрації на основі методів нечіткої логіки,
slbb – демонстрація задачі «куля на гойдалці»,
slcp – демонстрація нечіткої системи керування маятником,
sltank – демонстрація системи керування рівнем води в баці з нечітким регулятором,
sltankrule – те ж, що в попередньому випадку, але з додатковим переглядом нечітких правил (рис. 4.1),
sltbu – демонстрація функціонування нечіткої системи керування вантажівкою.

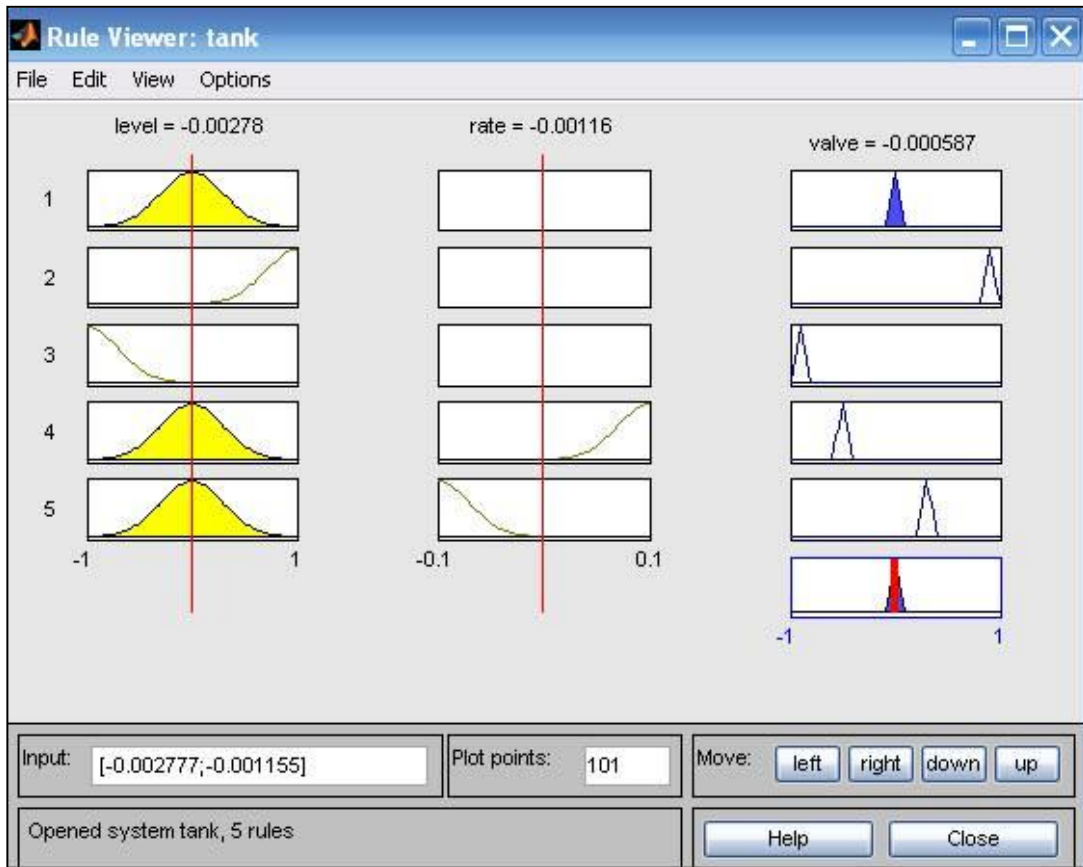
З кожним з цих прикладів зв'язаний М-файл, fis-файл або/і блок-діаграма Simulink, запуск яких здійснюється за допомогою однієї з зазначених команд. Текст пояснень у прикладах – англійською мовою.

Приклади доступні і через головне меню MATLAB (пункт **Help/Examples and Demos, Toolboxes/Fuzzy Logic**).

Додаткову інформацію можна одержати, використовуючи команду **help fuzzy**.



a)



б)

Рис.4.1. Приклад реалізації нечіткого регулятора рівня води в баці (**sltankrule**):
 а – мнемосхема у пакеті **Simulink**;
 б – набір правил

ЛІТЕРАТУРА

1. Корчемний М.О. Інтелектуальні системи. Частина 1. Нечіткі системи / М.О. Корчемний, В.П. Лисенко, М.В. Чапний. – К.: НАУ, 2006. – 75 с.
2. Лисенко В.П. Спеціальні розділи вищої математики (Нечіткі множини) / В.П. Лисенко, Б.В. Кузьменко. – К.: НАУ, 2004 р. – 83 с.
3. Леоненков А.В. Нечёткое моделирование в среде MATLAB и fuzzyTECH-СПб: БХВ-Петербург, 2003 г. – 736 с.
4. Kevin M. Passino Fuzzy Control / Kevin M. Passino, Stephen Yurkovich. – Ohio State University, 2001 y. – 572 p.
5. Ross T.J. Fuzzy logic with engineering applications. – McGraw-Hill, 1995. – 600p.
6. Асаи К. Прикладные нечеткие системы / К. Асаи, Д. Ватада, С. Иваи./Под ред. Т. Тэрано, К. Асаи, М. Сугено.- М.: Мир, 1993. – 368 с.
7. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер с польск / Д. Рутковская, М. Пилиньский, Л. Рутковский – М.: Горячая линия - Телеком, 2004. – 452 с.
8. Митюшкин Ю.И. Soft Computing: идентификация закономерностей нечеткими базами знаний / Ю.И. Митюшкин, Б.И. Мокин, А.П. Ротштейн. – Винница: УНИВЕРСУМ-Винница, 2002. – 145 с.