

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Інтелектуальні біотехнічні системи. Лабораторний практикум

Навчальний посібник
з дисципліни «Інтелектуальні біотехнічні системи»
для здобувачів ступеня магістр за спеціальністю
163 Біомедична інженерія,
що навчаються за освітньою програмою
«Регенеративна та біофармацевтична інженерія»

*Рекомендовано Методичною радою
КПІ ім. Ігоря Сікорського*

Київ – 2021

Інтелектуальні біотехнічні системи. Лабораторний практикум: Навчальний посібник з дисципліни «Інтелектуальні біотехнічні системи» для здобувачів ступеня магістра за спеціальністю 163 Біомедична інженерія, що навчаються за освітньою програмою «Регенеративна та біофармацевтична інженерія» / Уклад.: О.Б. Бесараб, О.Ю. Галкін. – К.: КПІ ім. Ігоря Сікорського, 2021. – ... с. (електронне видання)

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського

Навчальне видання

Інтелектуальні біотехнічні системи. Лабораторний практикум

Навчальний посібник з дисципліни
«Інтелектуальні біотехнічні системи»
для здобувачів ступеня магістра за спеціальністю
163 Біомедична інженерія, що навчаються за освітньою програмою
«Регенеративна та біофармацевтична інженерія»

Укладачі: *Бесараб Олександр Борисович, к.т.н.*
Галкін Олександр Юрійович, д-р біол. наук, проф.

Відповідальний
редактор

Рецензенти:

Зміст

Вступ.....	4
Лабораторна робота 1. Ознайомлення із програмним пакетом MatLab. Відображення біомедичної інформації.....	5
Лабораторна робота 2. Функцій належності для представлення біомедичної інформації	17
Лабораторна робота 3. Система нечіткого виведення за алгоритмом Мамдані для визначення дозування препарату.....	29
Лабораторна робота 4. Проектування систем нечіткого виводу Сугено для вирішення задач біомедичної інженерії.....	36
Лабораторна робота 6. Автоматичне проектування систем нечіткого виводу Мамдані для систематизації біомедичної інформації	43
Лабораторна робота 7. Автоматичне проектування систем нечіткого виводу Мамдані із використанням кластерного аналізу для систематизації біомедичної інформації	44
Лабораторна робота 8. Дослідження впливу методу дефазифікації при оптимізації моделі інтерпретації біомедичної інформації на основі нечіткого логічного висновку.....	45
Лабораторна робота 9. Розробка біотехнічної системи із регулятором на базі нечіткого логічного висновку	46

Вступ

Дисципліна «Інтелектуальні біотехнічні системи» відноситься до нормативних дисциплін для здобувачів ступеня магістр за спеціальністю 163 Біомедична інженерія, спеціалізації «Регенеративна та біофармацевтична інженерія». *Лабораторний практикум*, дає можливість поглибленого вивчення окремих тем курсу які стосуються біотехнічних систем на базі нечіткої логіки.

При розв'язання великої кількості практичних задач має місце об'єктивна невизначеність даних, що ускладнює, а подекуди і унеможлиблює, застосування традиційних методів аналізу. Медичні дані, які отримуються із різних джерел (закладів), можуть мати певну суб'єктивність невизначеність, відсутність тих чи інших параметрів. Суб'єктивна невизначеність, у свою чергу, визначається надійністю суджень експертів – лікарів підчас діагностики. Так, недостатня компетенція, занижена або завищена авторитетність джерела знань може призводити до хибних або неефективних рішень. Одним із підходів аналізу медичних даних є застосування апарату нечіткої логіки. Нечіткий логічний контролер може використовуватися у системах керування різних біотехнічних систем.

Лабораторна робота 1. Ознайомлення із програмним пакетом MatLab.

Відображення біомедичної інформації

Мета роботи: ознайомитися із: 1) загальними поняттями системи MatLab, 2) особливостями програмування в середовищі MatLab.

Теоретичні відомості

Система MatLab складається з базової програми та декількох десятків пакетів (Toolbox), які забезпечують розв'язання системою завдань різної складності. В результаті запуску системи MatLab на екрані монітору з'явиться робочий інтерфейс програми MatLab (рис. 1.1). При цьому кожний запуск системи називається сеансом роботи з системою.

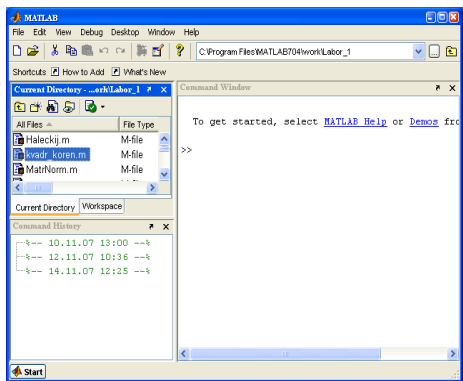


Рис. 1.1. Загальний вигляд графічного інтерфейсу системи

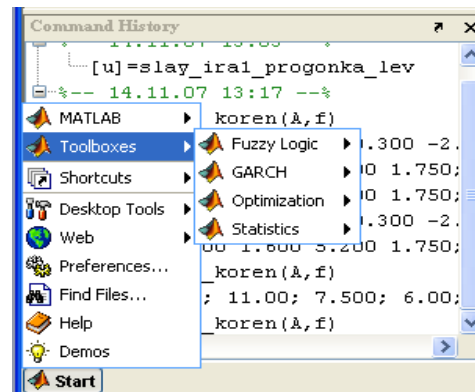


Рис. 1.2. Вікно доступу до компонентів системи (меню Start)

Панелью доступу до компонентів системи є меню Start (рис. 1.2), організоване у вигляді меню швидкого запуску «Пуск». Ця панель призначена для виклику потрібної довідки та спеціальних графічних засобів, перегляду демонстраційних прикладів тощо. Основну частину вікна MatLab займає вікно команд (рис. 1.3), призначене для взаємодії користувача з системою в режимі командного рядка. Вікно команд використовують для введення команд і функцій з необхідними

аргументами, для присвоєння змінним деяких значень і відображення результатів виконаних обчислень.

Застосування спеціальних графічних інтерфейсів користувача GUI (Graphic User Interface) для розв'язання окремих класів задач називають графічним або інтерактивним режимом роботи. Розглянемо вкладки, розташовані в лівій верхній частині головного вікна системи MatLab: вікно поточного каталогу або папки системи (Current Directory) та вікно перегляду робочої області системи (Workspace).

Вікно Workspace (рис. 1.4) надає можливість переглянути вміст робочої області та редагувати значення окремих змінних. При цьому варто пам'ятати, що всі змінні система розглядає як матриці або масиви. Тому у вікні перегляду робочої області вказується не тільки ім'я змінної, але й її розмір, об'єм пам'яті, який вона займає (в байтах), і тип цієї змінної.

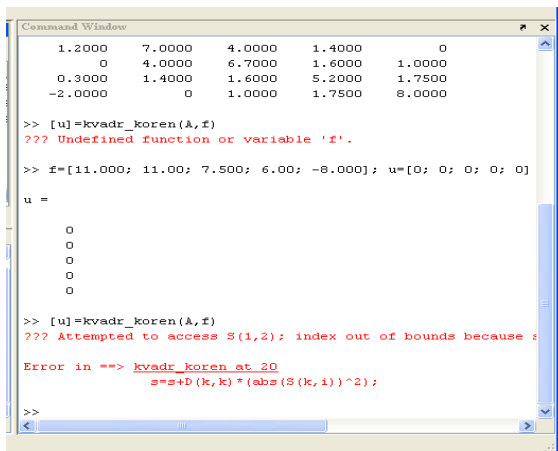


Рис. 1.3. Вікно команд системи

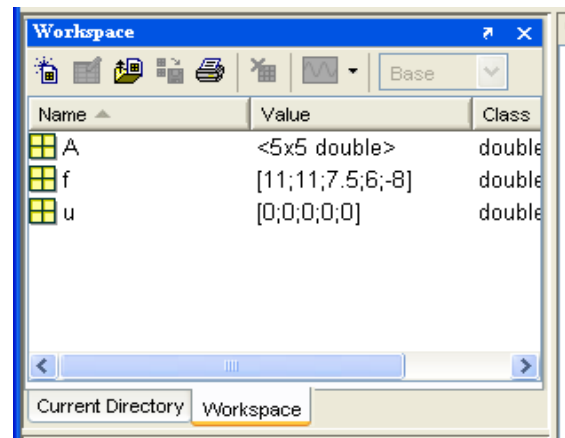


Рис. 1.4. Вікно перегляду робочої області системи

Вікно Workspace дозволяє видалити обрану змінну, очистити всю робочу область, завантажити змінні в робочу область із зовнішнього файлу та зберегти значення змінних в зовнішньому файлі.

Вікно поточного каталогу (рис. 1.5) відображає файли обраного каталогу системи MatLab. Файли з розширенням m, dat, fis є звичайними текстовими файлами, які можна переглядати й редагувати у будь-якому

ASCII-редакторі. Однак, більш зручним для цієї мети є вбудований редактор системи MatLab – редактор налагодження m-файлів Editor.

У лівій нижній частині головного вікна системи MatLab розташована вкладка, яка містить вікно історії команд Command History (рис. 1.6). Це вікно відображає всі команди, введені користувачем як під час поточного сеансу роботи із системою, так і під час попередніх сеансів.

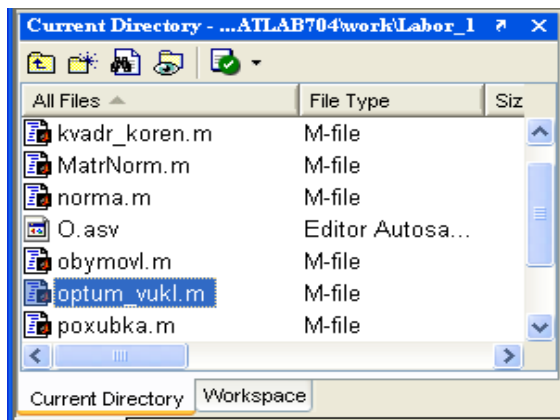


Рис. 1.5. Вікно поточного каталога системи MatLab

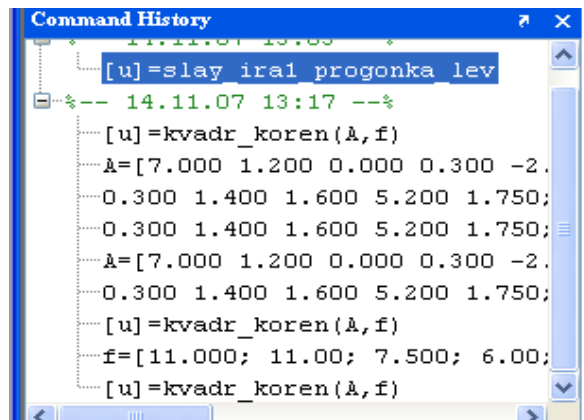


Рис. 1.6. Вікно історії команд

Для виконання однієї з цих команд достатньо двічі натиснути на її ім'я. Для редагування обраної команди її потрібно перемістити у вікно команд. Переглянути раніше виконані команди можна також скориставшись клавішами

«↑» і «↓» у вікні команд.

Основні прийоми роботи у вікні команд. Введення команди здійснюється набором відповідних символів у окремому рядку активного вікна команд після символів запрошення «>>». Аналогічно вводяться й окремі значення змінних, при цьому після імені змінної необхідно поставити символ

«=», який в системі відіграє роль знака присвоєння. Символи верхнього і нижнього регістрів сприймаються системою як різні. Якщо ім'я команди або значення змінної занадто довге, то варто використати символ продовження введення «...». При цьому всі набрані в новому рядку символи

вважаються продовженням символів попереднього рядка.

Якщо після введення імені команди (або значення змінної) поставлено символ «;», то результат її виконання у вікні команд не відображається.

Якщо результат виконання окремої команди набуває деякого значення, яке не присвоєно жодній змінній, то це значення присвоюється, за замовчуванням, змінній з ім'ям `ans`. Тип кожної змінної визначається її значенням.

Масиви і матриці є базовими об'єктами мови MatLab. З часом ця мова поповнювалася новими об'єктами (наприклад, об'єктами `net` для опису нейронних мереж, `fis` для опису систем нечіткого логічного виведення).

Для визначення значень елементів матриці використовуються символи:

«[» – для початку і «]» – для закінчення окремої матриці. При цьому матриці записуються по рядках: елементи матриці відокремлюють один від одного прогалиною, а рядки – символом «;». При введенні матриць розміром 1×1 знаки початку та закінчення матриці можна опустити. Наприклад, якщо необхідно присвоїти змінній з ім'ям `X` значення 10, то у вікні команд після символів «>>» треба ввести такий рядок: `X=10`. Задати матрицю можна таким чином: `M=[1 0.1 0.2; 0.8 0.9 1; 0.7 0.8 0.5; 1 0.5 0.2]`

До цієї матриці можуть бути застосовані різні унарні функції (транспонування, знаходження оберненої матриці, мінімальних і максимальних елементів у кожному стовпчику матриці тощо). Наприклад, для знаходження мінімальних елементів у кожному стовпчику матриці `M` необхідно скористатися функцією `min`:

```
>> min (M)
ans = 0.7000 0.1000 0.2000
```

Для звернення до окремих елементів матриці треба після її імені вказати в круглих дужках індекси відповідного елемента: номер рядка та номер стовпчика, які розділені комами (наприклад, `M(1, 2)`).

У деяких випадках для присвоєння значень елементам масивів і матриць зручним є використання оператора двокрапка («:»), дія якого аналогічна присвоєнню значень змінній в циклі типу for. При цьому форма запису має вигляд: «a:b:c», де a – значення першого елемента масиву, b – крок зміни елементів, c – значення останнього елемента масиву. Наприклад, при побудові графіків функцій необхідно визначити діапазони значень відповідних змінних. Це зручно зробити таким чином: X=0:0.1:10. В результаті введення цього рядка буде визначено масив X, який складається зі 101 елемента, першим з яких є число 0, а останнім – число «10».

Для введення матриць в системі передбачені окремі функції, деякі з яких наведено в табл. 1.1. Наприклад, щоб визначити матрицю Z розміром 3*4, яка

складається з нулів, у вікні команд потрібно ввести ім'я відповідної функції zeros та її параметри: Z=zeros(3, 4). При написанні алгебраїчних виразів MatLab дозволяє використовувати традиційні знаки арифметичних операцій.

Таблиця 1.1

Найпростіші функції для введення матриць

Ім'я функції	Призначення функції
ones	введення матриці, яка складається з одиниць
zeros	введення матриці, яка складається з нулів
rand	введення матриці, елементи якої розподілені за нормальним законом на проміжку [0, 1]
randn	введення матриці, елементи якої розподілені за нормальним законом
eye	введення матриці, головна діагональ якої складається з одиниць
magic	введення “магічної” матриці, сума елементів якої по рядкам, по стовпчикам і по діагоналі є однаковою.

Інформація про всі функції системи MatLab міститься в довіднику

системи.

1. Графічні можливості системи MatLab [3]

Система має зручні та потужні засоби візуалізації та графічного зображення різноманітних математичних об'єктів типу кривих, поверхонь і діаграм на площині та у 3-вимірному просторі. При цьому використовуються різні системи координат, стилі та способи виділення кольором зображень.

Розглянемо основні функції системи, які можуть бути використані для візуалізації окремих властивостей нечітких моделей.

Для побудови графіка функції однієї змінної насамперед необхідно визначити множину (масив) значень незалежної змінної та відповідну множину значень залежної (функціональної) змінної. Нижче наведено послідовність команд для побудови графіка функції $y=\sin(x)$ за допомогою функції `plot`: `X=- 10:0.1:10; Y=sin(X); plot(X,Y)`. Після виконання даної послідовності команд виникне нове вікно з графіком цієї функції (рис. 1.7), яке має власне головне меню та панель команд. Відповідні команди дозволяють виконати редагування властивостей отриманого графіка, наприклад, внести додатковий текст (легенду), змінити колір, тип ліній і масштаб зображення, виконати поворот зображення.

Для побудови графіків 3-вимірних поверхонь і кривих, які є зображеннями функцій двох змінних, можна скористатися такими функціями системи MatLab: 1) `plot3` – функція побудови зображень 3-вимірних поверхонь лініями; 2) `mesh` – функція побудови зображень 3-вимірних поверхонь із функціональним розфарбовуванням ліній контурної сітки; 3) `surf` – функція побудови зображень 3-вимірних поверхонь з функціональним розфарбовуванням комірок контурної сітки.

При цьому для присвоєння значень незалежним змінним зручно скористатися функцією `meshgrid`: `[X,Y]=meshgrid(x,y)`, яка перетворює

область, визначену векторами x і y , в масиви X і Y . Нижче наведено послідовність команд, необхідних для побудови графіка поверхні $z=\min(x,y)$ за допомогою функції `surf()` (рис. 1.8):

```
>> [x,y]=meshgrid([-1:0.1:1]); z=min(x,y); surf(x,y,z)
```

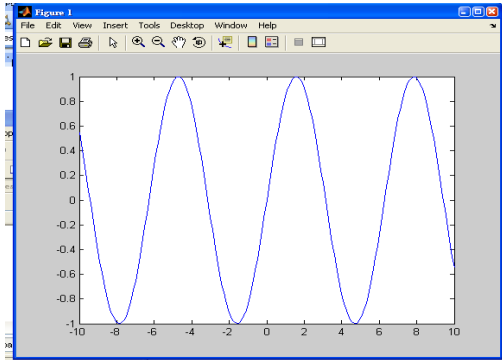


Рис. 1.7. Вікно із побудованим графіком функції $y = \sin(x)$

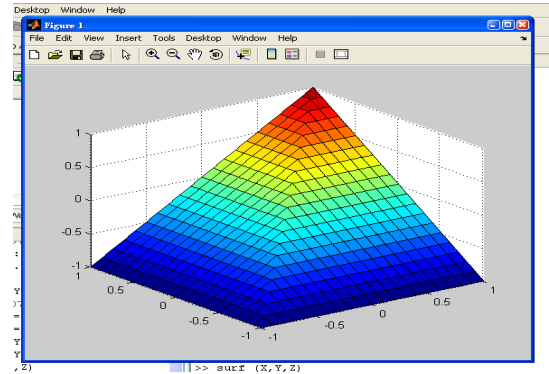


Рис. 1.8. Поверхня $z=\min(x, y)$, побудована за допомогою функції `surf`

2. Основи програмування в системі MatLab

M-файли в системі MatLab (сценарії та функції). Файли, які містять команди та оператори системи MatLab, називаються m-файлами. Розрізняють два типи m-файлів: сценарії (m-сценарії) та функції (m-функції). Вони можуть бути створені в будь-якому текстовому редакторі та збережені у файлі з розширенням `m`. Однак система MatLab має особистий редактор m-файлів – Editor, який можна викликати:

- 1) за допомогою опції `New M-File` з меню `File`;
- 2) з командного рядка системи MatLab, ввівши команду `edit`;
- 3) за допомогою меню швидкого запуску «Пуск».

Зазвичай, при написанні m-функцій ім'я m-файлу, в якому запам'ятовується програма, співпадає з ім'ям функції.

M-сценарії дозволяють автоматизувати виконання послідовності операторів. В них можна звертатися до будь-яких команд та функцій системи MatLab, при цьому не потрібно оголошувати імена та типи змінних.

Сценарії можуть виконуватися як безпосередньо з файлу, так і шляхом копіювання всього сценарію або його фрагментів у командне вікно. Вони оперують даними з робочої області Workspace й можуть створювати нові дані з метою їхньої подальшої обробки. Декілька сценаріїв можуть виконуватися послідовно один за одним, передаючи дані наступному сценарію через робочий простір або запам'ятовуючи проміжні результати в файлах. Один m-файл може включати в себе декілька m-сценаріїв.

Сценарій, зазвичай, розпочинається із заголовку `script`, за яким розміщено рядок коментарю, який пояснює призначення сценарію. Коментар може приймати вигляд окремого рядка або розміщуватися після будь-якого оператора, за умови, що йому передує символ «%». Наведемо приклад сценарію (файл `example.m`), який обчислює радіус-вектор `rho` для тригонометричної функції `sin` в залежності від кута `theta` та будує відповідний графік в полярних координатах:

сценарії відсутні вхідні та вихідні аргументи: програма сама створює змінні, які зберігаються в робочій області системи. Коли виконання сценарію завершено, змінні `theta` та `rho` залишаються в робочій області. Для того, щоб переглянути їх, треба відкрити вікно робочої області Workspace або скористатися командою `whos`. Запуск сценаріїв на виконання може також здійснюватися і в середовищі текстового редактора Editor: 1) за допомогою опції `Run` меню `Debug` (або клавіші `F5`), 2) за допомогою опції `Evaluate Selection` меню `Text`.

Команда Evaluate Selection забезпечує обчислення виділеного фрагменту m-сценарію, якщо в поточний момент відомі значення всіх змінних, які входять до нього.

Результат даних обчислень відображається у вікні Command Window.

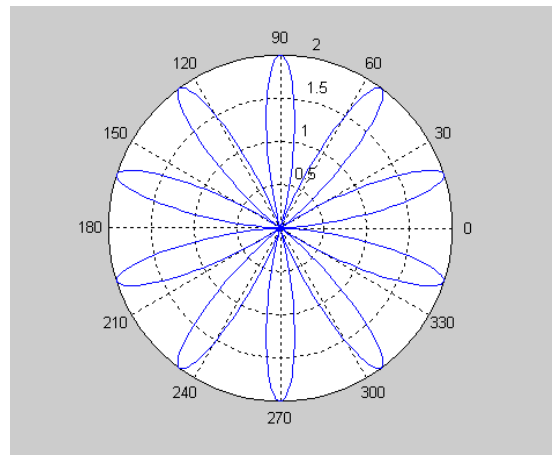


Рис.1.9. Результат виконання команди `exemple`

M-функції дозволяють створювати нові бібліотеки m-функцій і пакети прикладних програм. Вони, на відміну від сценаріїв, мають вхідні та вихідні аргументи та створюють локальні змінні в робочій області m-функції, яка відрізняється від робочої області системи MatLab.

Структура m-функції характеризується наступними компонентами:

1. Рядок визначення функції (заголовок функції) задає її ім'я та послідовність вхідних і вихідних аргументів (змінних).

Можливі такі варіанти оформлення заголовків функцій:

`function f1`

`function y=f3(x1, x2,...)`

`function f2(x1, x2,...)`

`function [y1, y2,...]=f4(x1, x2,...),`

де `f1`, `f2`, `f3`, `f4` – заголовки функцій, `(x1, x2, ...)` – вхідні аргументи; `(y1, y2, ...)` – вихідні аргументи. Якщо m-функція має більше одного вихідного аргумента, то вони розміщуються в квадратних дужках і відокремлюються один від одного

комами. Вхідні аргументи m-функції, якщо їх більше одного, розміщуються в круглих дужках.

2. Перший рядок коментарю, зазвичай, визначає призначення функції. Він використовується командою `help <ім'я каталогу>` для виведення на екран загальної інформації про всі m-функції заданого каталогу.

3. Тіло функції – це програмний код на мові MatLab, який реалізує обчислення та присвоює значення вихідним змінним, наприклад:

Заголовок функції:	Перший	<code>function f=factorial (n)</code>
рядок коментарю:		<code>% FACTORIAL – обчислення факторіалу</code>
Коментар:		<code>% factorial(n) визначає значення n!</code>
Тіло функції:		<code>f=prod(1:n);</code>

M-файли можуть містити коди декількох функцій. Перша функція в файлі – основна функція, яка викликається за ім'ям файлу. Інші функції являють собою підфункції, які доступні лише основній функції та іншим підфункціям цього файлу.

Кожна підфункція має свій власний заголовок. Підфункції розташовуються одна за одною й можуть викликатися в довільному порядку в тілі основної функції, наприклад:

Основна функція	<code>function [avg,med]= newstats(u)</code> <code>% NEWSTATS знаходить середнє значення та медіану</code> <code>% для елементів вектора u з використанням підфункцій</code> <code>n=length(u);</code> <code>avg=mean_a(u,n);</code> <code>med=median_a(u,n);</code>
Підфункції	<code>function m=mean_a(v,n)</code> <code>% ОБЧИСЛИТИ</code> <code>середнє m=sum(v)/n;</code> <code>function m=median_a(v,n) % ОБЧИСЛИТИ медіану</code> <code>w=sort(v);</code> <code>if rem(n,2)==1</code> <code> m=w((n+1)/2);</code> <code>else m=(w(n/2)+w(n/2+1))/2;</code> <code>end</code>

MatLab дозволяє використовувати «анонімні» функції, які не мають особистого імені, а замість нього використовують вказівку на функцію.

Формат оголошення таких функцій приймає наступний вигляд:

`h_F=@ (список_параметрів) формула`

Функції такого типу корисно застосовувати при роботі з формулами. Наприклад, “анонімна” функція піднесення до квадрату з вказівкою `sqr` може бути описана та використана наступним чином:

```
>>sqr=@(x)x.^2; sqr([1 2 3])  
ans = 1 4 9
```

Вказівка на анонімну функцію може бути передана іншим функціям як параметр. Наприклад, ми можемо звернутися до функції обчислення визначеного інтеграла від x^2 на інтервалі від 0 до 1 наступним чином:

```
>> quad (sqr, 0, 1)  
ans = 0.3333
```

M-функції викликаються аналогічно вбудованим функціям системи MatLab. Їх також можна, на відміну від сценаріїв, застосовувати в арифметичних виразах.

Розглянемо m-файл, який обчислює середнє значення елементів вектора.

```
function y=average(x)  
% AVERAGE(x) обчислює середнє значення елементів вектора  
% Якщо вхідний аргумент не є вектором, генерується  
помилка [m,n]=size(x);  
if (~((m==1)|(n==1))|(m==1&n==1))  
    error('Вхідний масив повинен бути  
    вектором')  
end  
y=sum(x)/length(x  
);
```

Оператори в тілі функції можуть складатися з виклику функцій,

програмних конструкцій для управління потоком команд, інтерактивного вводу-виводу, обчислень, операторів присвоєння, коментарів і порожніх рядків. Звернутися до функції average можна таким чином:

```
>>y=average
```

```
(1:99)
```

```
y = 50
```

Завдання роботи

Розробити скрипти побудови дво- та тривимірних графіків функцій для відображення медичної інформації наданої викладачем у табличному вигляді.

Лабораторна робота 2. Функцій належності для представлення біомедичної інформації

Мета роботи: ознайомитися із застосуванням функцій належності, для представлення біомедичної інформації на прикладі віку пацієнта. Опанувати операції із нечіткими множинами.

Теоретичні відомості

В основі апарату нечіткої логіки покладено уявлення про те, що множина елементів, що мають спільну властивість, можуть мати цю властивість у різній мірі. Міра належності елемента множині формалізується за допомогою функції належності.

Функцією належності (ФН) називається функція, яка дозволяє для довільного елемента універсальної множини розрахувати ступінь його належності нечіткій множині.

Універсальна множина (універсум) – це множина, яка містить всі можливі елементи в рамках контексту визначеної предметної області.

Нечіткою множиною \tilde{A} на універсальній множині U називається сукупність пар $(\mu_A(u), u)$, де $\mu_A(u)$ – ступінь належності елемента $u \in U$ нечіткій множині \tilde{A} . Ступінь належності – число із діапазону $[0; 1]$. Чим більше ступінь належності, тим більшою мірою елемент універсальної множини відповідає властивостям нечіткої множини. Таким чином можна стверджувати, що нечітка множина описується її ФН, а операції із нечіткими множинами зводяться до операцій з їх ФН.

Розглянемо вбудовані функції належності Matlab.

1. Трикутна ФН описується залежністю:

$$\mu(x, [a, b, c]) = \begin{cases} 0, & \text{якщо } x \leq a; \\ \frac{x-a}{c-a}, & \text{якщо } a < x \leq c; \\ \frac{b-x}{b-c}, & \text{якщо } c < x < b; \\ 0, & \text{якщо } x \geq b. \end{cases}$$

де x – величина для якої визначається ступінь належності, $[a, b, c]$ – вектор параметрів функції належності ($a < b < c$).

Графік трикутної ФН із параметрами $[0.2 \ 0.5 \ 0.6]$ наведено на рис 2.1.

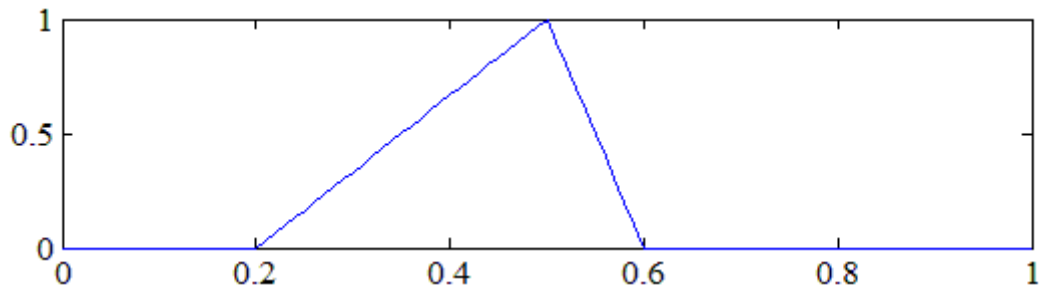


Рис. 2.1. Графік трикутної ФН із параметрами $[0.2 \ 0.5 \ 0.6]$

Для побудови даного графіку функції було створено та виконано m-файл із наступним вмістом:

```
x = [0:0.01:1]; %формуємо масив значень від 0 до 1 із кроком 0.01
y = trimf(x, [0.2 0.5 0.6]); %розраховуємо масив значень трикутної ФН за значеннями масиву x
plot(x, y) % будуємо двовірний графік функції y(x)
```

Для подальшого використання отриманого зображення його можна зберегти, наприклад у вигляді точкового рисунку *.bmp. Для цього у меню File, необхідно обрати пункт Save As (Рис. 2.2), та у стандартному діалоговому вікні збереження файлу обрати місце збереження, задати ім'я та тип файлу.

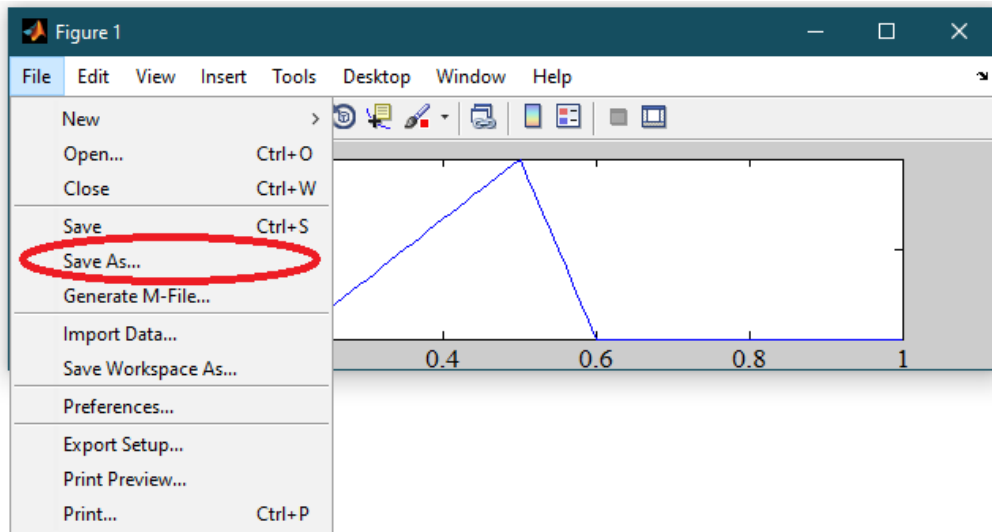


Рис. 2.2. Збереження графічного зображення ФН

2. Трапецієвидна ФН описується залежністю:

$$\mu(x, [a, b, c, d]) = \begin{cases} 0, & \text{якщо } x \leq a; \\ \frac{x - a}{b - a}, & \text{якщо } a < x < b; \\ 1, & \text{якщо } b \leq x \leq c; \\ \frac{c - x}{c - d}, & \text{якщо } c < x < d; \\ 0, & \text{якщо } x \geq d. \end{cases}$$

Графік трапецієвидної ФН із параметрами [0.2 0.4 0.5 0.6] наведено на рис 2.3.

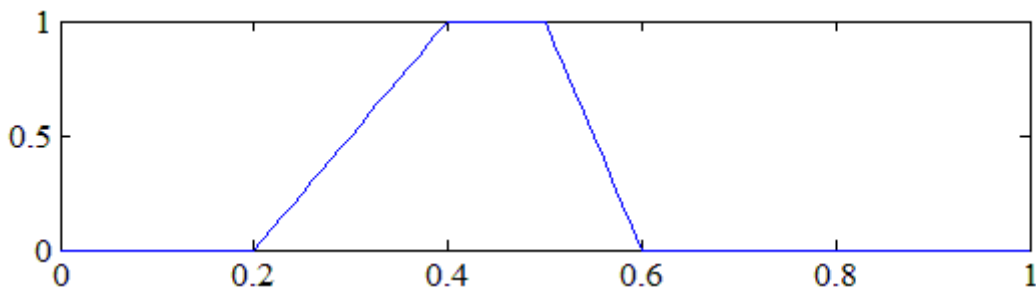


Рис. 2.3. Графік трапецієвидної ФН із параметрами [0.2 0.4 0.5 0.6]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \mathbf{trapmf}(x, [0.2 \ 0.4 \ 0.5 \ 0.6]);$

3. Дзвоноподібна ФН описується залежністю:

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

де a – коефіцієнт концентрації, b – коефіцієнт крутизни, c – координата максимуму.

Графік дзвоноподібної ФН із параметрами [0.2 1 0.5] наведено на рис 2.4.

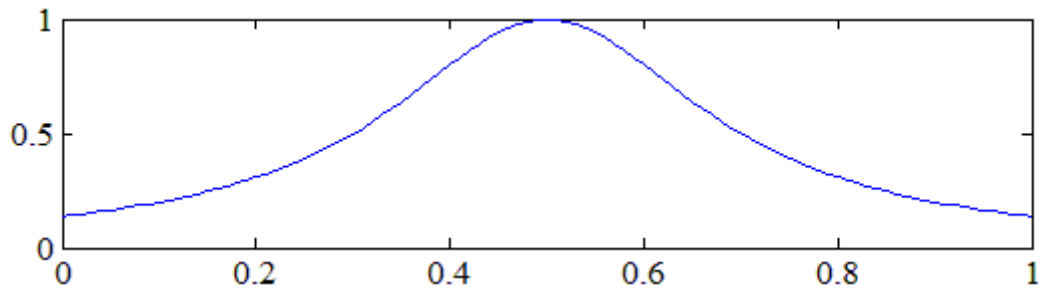


Рис. 2.4. Графік дзвоноподібної ФН із параметрами [0.2 1 0.5]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \mathbf{gbellmf}(x, [0.2 \ 1 \ 0.5]);$

4. Гаусова ФН описується залежністю:

$$\mu(x, [a, c]) = \exp\left[-\frac{(x - c)^2}{2a^2}\right]$$

де a – коефіцієнт концентрації, c – координата максимуму.

Графік гаусової ФН із параметрами [0.2 0.5] наведено на рис 2.5.

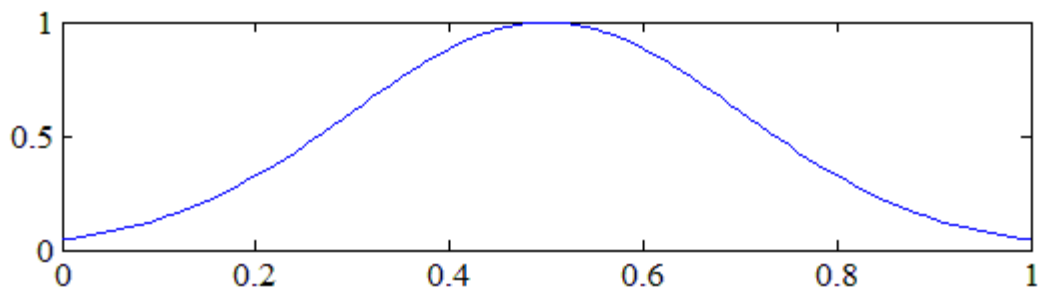


Рис. 2.5. Графік гаусової ФН із параметрами [0.2 0.5]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:

5. Двостороння гаусова ФН описується залежністю:

Якщо $c_1 < c_2$, то

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

Якщо $c_2 < c_1$, то

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

Коефіцієнти a та c , як і для гаусової ФН показують концентрації та координати максимумів лівої (індекс 1) та правої (індекс 2) частин графіку.

Графік двосторонніх гаусових ФН наведено на рис 2.6.

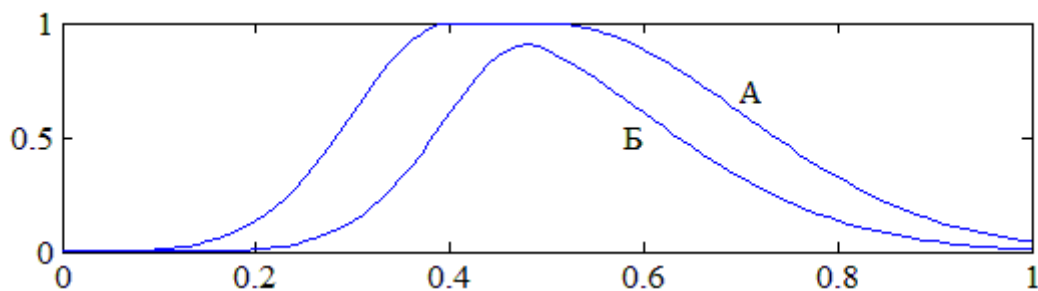


Рис. 2.6. Графік двосторонніх гаусових ФН

(А із параметрами [0.1 0.4 0.2 0.5], Б із параметрами [0.1 0.5 0.2 0.4])

Для побудови даного графіку функції було створено та виконано m-файл із наступним вмістом:

```
x = [0:0.01:1];
y = gauss2mf(x, [0.1 0.4 0.2 0.5]);
plot(x, y)
hold on
y = gauss2mf(x, [0.1 0.5 0.2 0.4]);
plot(x, y)
hold off
```

6. Сигмоїдальна ФН описується залежністю:

$$\mu(x, [a, b]) = \frac{1}{1 + \exp(-a(x - b))}$$

де a – коефіцієнт (при від’ємних значеннях ФН монотонно спадаюча) ширини переходу між мінімальним та максимальним значеннями, b – координата центру переходу.

Графік сигмоїдальної ФН з параметрами [20 0.5] наведено на рис 2.7.

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \text{sigmf}(x, [20 \ 0.5]);$

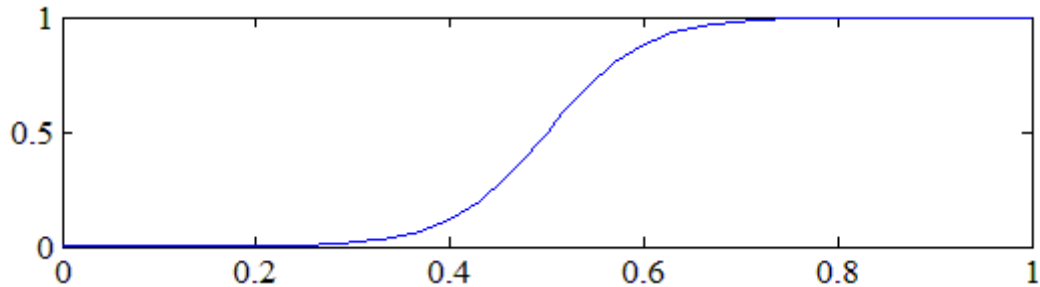


Рис. 2.7. Графік сигмоїдальної ФН з параметрами [20 0.5]

7. Двостороння сигмоїдальна ФН різниці описується залежністю:

$$\mu(x, [a_1, b_1, a_2, b_2]) = \frac{1}{1 + \exp(-a_1(x - b_1))} - \frac{1}{1 + \exp(-a_2(x - b_2))}$$

Графік двосторонньої сигмоїдальної ФН різниці з параметрами [20 0.3 30 0.8] наведено на рис 2.8.

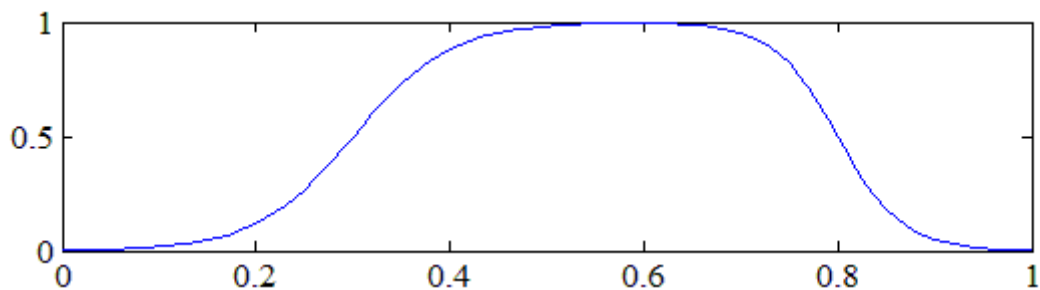


Рис. 2.8. Графік двосторонньої сигмоїдальної ФН різниці з параметрами [20 0.3 30 0.8]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \text{dsigmf}(x, [20 \ 0.3 \ 30 \ 0.8]);$

8. Двостороння сигмоїдальна ФН добутку описується залежністю:

$$\mu(x, [a_1, b_1, a_2, b_2]) = \frac{1}{1 + \exp(-a_1(x - b_1))} * \frac{1}{1 + \exp(-a_2(x - b_2))}$$

Графік двосторонньої сигмоїдальної ФН добутку з параметрами [20 0.3 -30 0.8] наведено на рис 2.9.

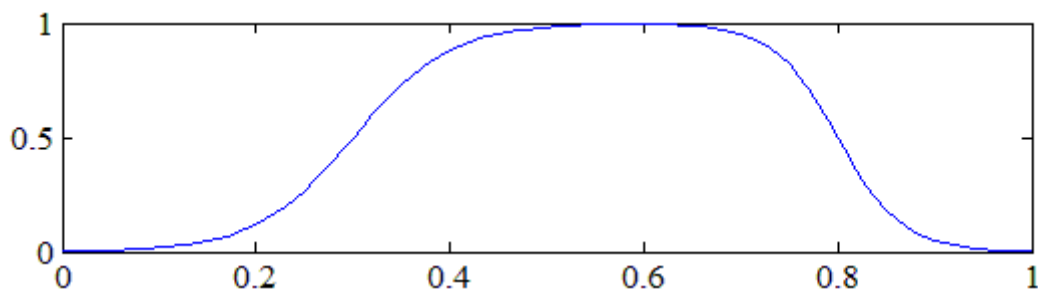


Рис. 2.9. Графік двосторонньої сигмоїдальної ФН добутку з добутку [20 0.3 -30 0.8]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \text{psigmf}(x, [20\ 0.3\ -30\ 0.8]);$

9. Z-функція належності описується залежністю:

$$\mu(x, a, b) = \begin{cases} 1, & \text{якщо } x \leq a; \\ 1 - \frac{2(x - a)^2}{(b - a)^2}, & \text{якщо } a < x \leq \frac{a + b}{2}; \\ \frac{2(x - a)^2}{(b - a)^2}, & \text{якщо } \frac{a + b}{2} < x < b; \\ 0, & \text{якщо } x \geq b. \end{cases}$$

де a – найбільша координата максимального (1) значення ФН, b – найменша координата мінімального (0) значення ФН.

Графік Z-функції належності з параметрами [0.3 0.8] наведено на рис 2.10.

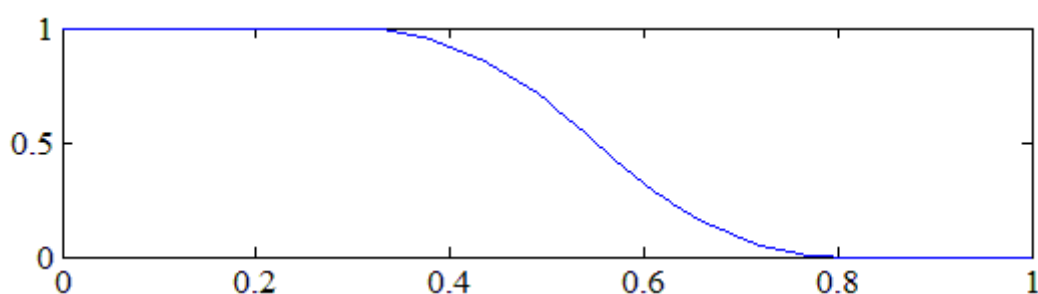


Рис. 2.10. Графік Z-функції належності з параметрами [0.3 0.8]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \text{zmf}(x, [0.3\ 0.8]);$

10. S-функція належності є дзеркальною до Z-функції та описується залежністю:

$$\mu(x, a, b) = \begin{cases} 0, & \text{если } x \leq a; \\ \frac{2(x-a)^2}{(b-a)^2}, & \text{если } a < x \leq \frac{a+b}{2}; \\ 1 - \frac{2(x-a)^2}{(b-a)^2}, & \text{если } \frac{a+b}{2} < x < b; \\ 1, & \text{если } x \geq b. \end{cases}$$

Графік S-функції належності з параметрами [0.3 0.8] наведено на рис 2.11.

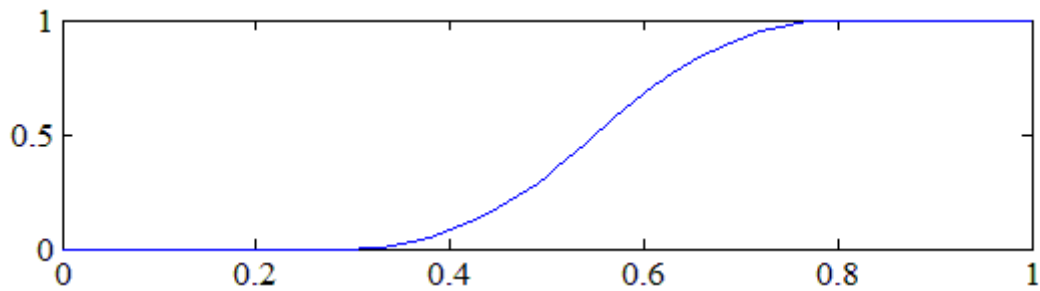


Рис. 2.11. Графік S-функції належності з параметрами [0.3 0.8]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \mathbf{smf}(x, [0.3\ 0.8]);$

11. PI-функція належності визначається як алгебраїчних добуток значень Z-функції та S-функції. Графік PI-функції належності з параметрами [0.2 0.3 0.5 0.8] наведено на рис 2.12.

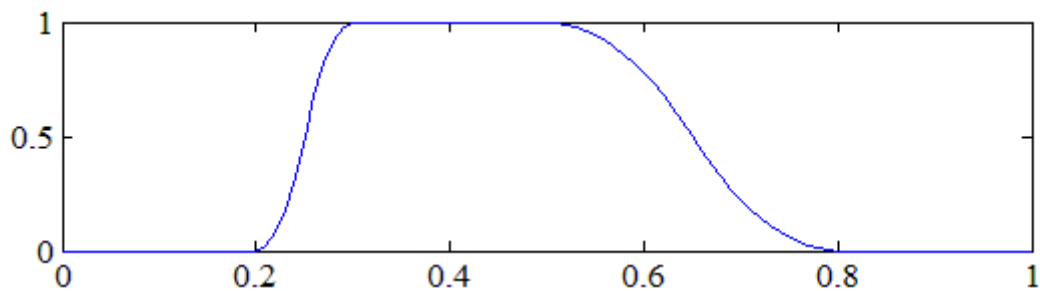


Рис. 2.12. Графік PI-функції належності з параметрами [0.2 0.3 0.5 0.8]

Для побудови даного графіку функції було використано m-файл для побудови графіку трикутної ФН у якому замінено другий рядок на наступний:
 $y = \mathbf{pimf}(x, [0.2\ 0.3\ 0.5\ 0.8]);$

Завдання роботи

Завдання 1. На універсальній множині від 0 до 100, що відображає вік людини, сформуванати нечіткі множини, що відповідають віковим групам: «діти»,

«юнаки», «молодь», «дорослі», «літні люди», «довгожителі». Тип функцій належності нечітких множин вибрати відповідно до варіанту із таблиці 1.1.

Таблиця 1.1. Варіанти перших завдань до лабораторної роботи № 1

Варіант	Нечіткі множини / № ФН					
	діти	юнаки	молодь	дорослі	літні люди	довгожителі
1	6	3	11	5	8	6
2	9	4	2	7	5	6
3	9	11	4	7	3	10
4	9	5	11	1	11	6
5	6	8	5	1	2	10
6	9	7	2	1	4	6
7	6	4	8	1	1	10
8	6	8	11	2	2	6
9	6	5	1	11	1	6
10	9	4	5	5	3	6
11	9	8	11	2	1	10
12	9	5	5	4	8	6
13	6	1	8	3	8	6
14	9	5	5	1	2	10
15	9	2	1	5	7	10
16	6	8	3	2	11	6
17	9	2	4	5	2	10
18	9	7	2	3	8	10
19	6	2	5	2	7	6
20	9	3	7	11	4	6
21	6	2	8	5	8	10
22	6	5	2	7	4	10
23	6	8	5	4	7	10
24	9	2	4	3	5	10
25	9	1	8	3	1	6
26	6	8	2	3	11	6
27	9	7	4	5	5	10
28	9	1	7	1	8	10
29	6	2	5	4	4	6
30	6	2	1	3	4	10

Номери ФН: 1 – трикутна; 2 – трапецієвидна; 3 – дзвоноподібна; 4 – гаусова; 5 – двостороння гаусова; 6 – сигмоїдальна; 7 – двостороння сигмоїдальна (різниці); 8 – двостороння сигмоїдальна (добутку); 9 – Z-функція; 10 – S-функція; 11 – PI-функція.

Приклад 1. На універсальній множині від 10 до 40, що відображає температуру повітря у приміщенні в °С, сформувані нечіткі множини, що відповідають температурним діапазнам «холодно», «комфортно», «спекотно». Для нечіткої множини «комфортно» задано трикутну ФН, а для інших – сигмоїдальні ФН. Нехай вважатимемо температуру від 20 до 25 комфортною. Від 10 до 15 низькою, а температуру від 30 до 40 жаркою. Для значень температур від 15 до 20 та від 25 до 30 складно однозначно визначити до якої із нечітких множин їх віднести, тому віднесемо ці значення одночасно до обох сусідніх нечітких множин. Коефіцієнти ФН підберемо таким чином, щоб ФН сусідніх нечітких множин перетинались на рівні 0,5 (або близькому до нього).

Для побудови графіку сформуємо наступний m-файл

```
x = [10:0.1:40]; % формуємо вектор X зі значеннями від 10 до 40
із кроком 0,1
y = sigmf(x, [-0.5 17.5]); % розраховуємо значення ФН множини
«холодно»
plot(x, y) % побудова графіку
hold on % команда зберігає графіки в поточній системі координат
так, щоб нові графіки, додані до осей, не видаляла існуючі
графіки
y = trimf(x, [13 22.5 32]); % розраховуємо значення ФН множини
«комфортно»
plot(x, y) % побудова графіку
y = sigmf(x, [0.5 27.5]); % розраховуємо значення ФН множини
«спекотно»
plot(x, y) % побудова графіку
hold off % відміна команди hold on
```

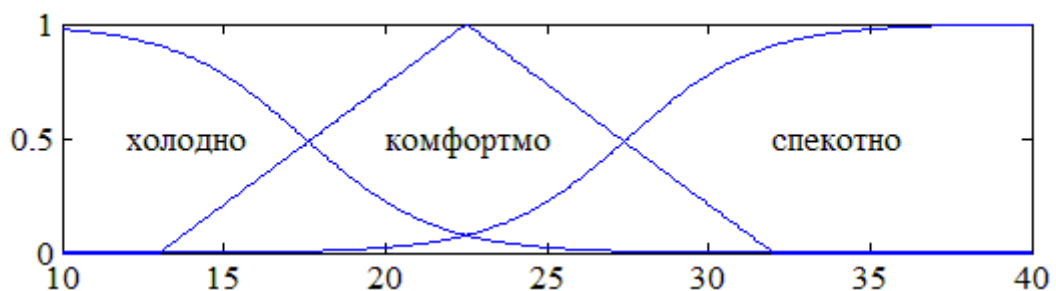


Рис. 2.13. Графік до прикладу 1

Завдання 2. Побудувати графіки ФН нечітких множин, що отримані в результаті операцій із нечіткими множинами із завдання 1 відповідно до варіантів із табл. 1.2.

«діти», «юнаки», «молодь», «дорослі», «літні люди», «довгожителі»

Таблиця 1.2. Варіанти перших завдань до лабораторної роботи № 1

Вар	Заперечення	Перетин множин	Об'єднання множин
1, 16	«НЕ діти»	«діти» І «юнаки»	«діти» АБО «молодь»
2, 17	«НЕ юнаки»	«юнаки» І «молодь»	«діти» АБО «літні люди»
3, 18	«НЕ дорослі»	«молодь» І «дорослі»	«діти» АБО «довгожителі»
4, 19	«НЕ літні люди»	«дорослі» І «літні люди»	«діти» АБО «юнаки»
5, 20	«НЕ довгожителі»	«літні люди» І «довгожителі»	«дорослі» АБО «діти»
6, 21	«НЕ молодь»	«діти» І «юнаки»	«дорослі» АБО «юнаки»
7, 22	«НЕ діти»	«юнаки» І «молодь»	«дорослі» АБО «молодь»
8, 23	«НЕ юнаки»	«молодь» І «дорослі»	«дорослі» АБО «літні люди»
9, 24	«НЕ дорослі»	«дорослі» І «літні люди»	«довгожителі» АБО «діти»
10, 25	«НЕ літні люди»	«літні люди» І «довгожителі»	«довгожителі» АБО «юнаки»
11, 26	«НЕ довгожителі»	«діти» І «юнаки»	«юнаки» АБО «дорослі»
12, 27	«НЕ молодь»	«юнаки» І «молодь»	«юнаки» АБО «літні люди»
13, 28	«НЕ діти»	«молодь» І «дорослі»	«юнаки» АБО «молодь»
14, 29	«НЕ юнаки»	«дорослі» І «літні люди»	«молодь» АБО «дорослі»
15, 30	«НЕ дорослі»	«літні люди» І «довгожителі»	«довгожителі» АБО «діти»

Приклад 2. Побудувати графіки ФН нечітких множин, що отримані в результаті операцій із нечіткими множинами із прикладу 1.

Заперечення, логічне «НЕ». ФН нечіткої множини «НЕ холодно» може бути описана співвідношенням $\mu_{\text{НЕ холодно}}(x) = 1 - \mu_{\text{холодно}}(x)$

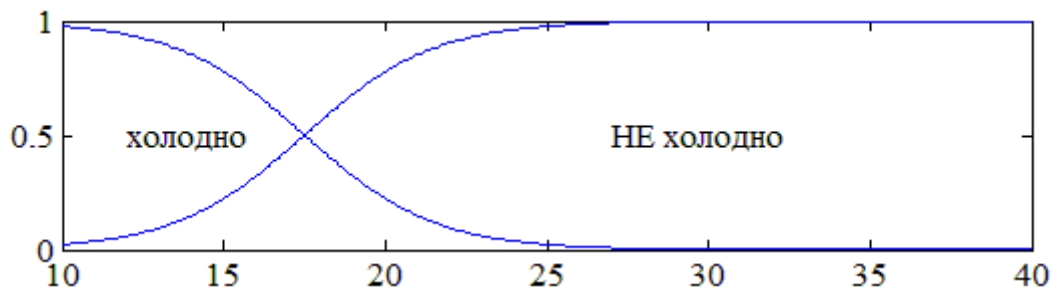


Рис. 2.14. Графік ФН нечітких множин «холодно» та «НЕ холодно»

Для побудови графіку сформовано та виконано наступний m-файл

```
x = [10:0.1:40]; % формуємо вектор X зі значеннями від 10 до 40
із кроком 0,1
y = sigmf(x, [-0.5 17.5]); % розраховуємо значення ФН множини
«холодно»
plot(x, y) % побудова графіку
hold on % команда зберігає графіки в поточній системі координат
так, щоб нові графіки, додані до осей, не видаляла існуючі
графіки
y1 = 1-y; % розраховуємо значення ФН множини «не холодно»
plot(x, y1) % побудова графіку
text (12, 0.5, 'холодно'); %виведення тексту на графіку
```

```
text (27, 0.5, 'НЕ холодно');
hold off % відміна команди hold on
```

Перетин нечітких множин, логічне «І». ФН нечіткої множини «холодно

І комфортно» може бути описана співвідношенням:

$$\mu_{\text{холодно І комфортно}}(x) = \text{MIN}(\mu_{\text{холодно}}(x); \mu_{\text{комфортно}}(x))$$

Для побудови графіку сформуємо та виконаємо наступний m-файл:

```
x = [10:0.1:40]; % формуємо вектор X зі значеннями від 10 до 40
із кроком 0,1
y1 = sigmf(x, [-0.5 17.5]); % розраховуємо значення ФН множини
«холодно»
y2 = trimf(x, [13 22.5 32]); % розраховуємо значення ФН
множини «комфортно»
y=min(y1,y2) % розраховуємо значення ФН множини «холодно І
комфортно»
plot(x, y) % побудова графіку
```

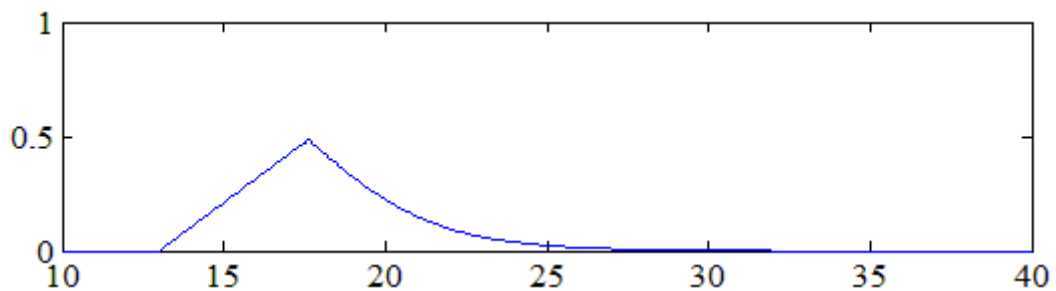


Рис. 2.15. Графік ФН нечіткої множини «холодно І комфортно»

Об'єднання нечітких множин, логічне «АБО». ФН нечіткої множини

«холодно АБО комфортно» може бути описана співвідношенням:

$$\mu_{\text{холодно АБО комфортно}}(x) = \text{MAX}(\mu_{\text{холодно}}(x); \mu_{\text{комфортно}}(x))$$

Для побудови графіку використаємо m-файл для операції «І» у якому замінимо функцію MIN на функцію MAX:

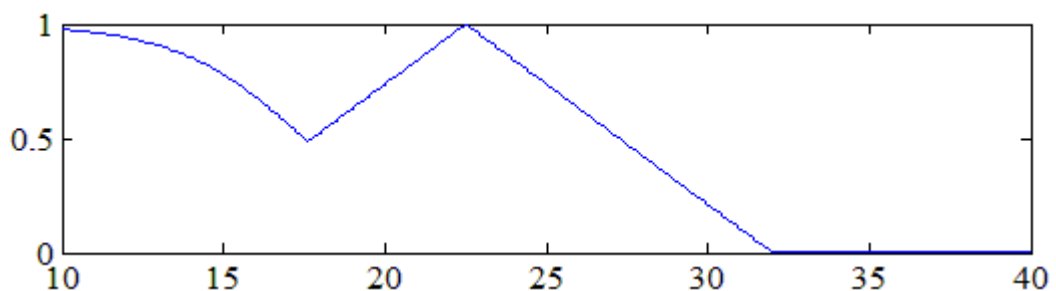


Рис. 2.16. Графік ФН нечіткої множини «холодно АБО комфортно»

Лабораторна робота 3. Система нечіткого виведення за алгоритмом Мамдані для визначення дозування препарату

Мета роботи: ознайомитися з алгоритмом нечіткого виведення Мамдані для визначення дозування препарату

Теоретичні відомості

Одним із напрямків розвитку систем штучного інтелекту є розробка лінгвістичних моделей, що відтворюють спосіб мислення людини. Основоположником таких моделей є американський математик і логік азербайджанського походження, засновник теорії нечітких множин і нечіткої логіки Лотфі Заде. Найбільш поширеним алгоритмом нечіткого логічного виведення є алгоритм запропонований у 1974 році вченим в галузі математики, комп'ютерних наук, телекомунікацій та штучного інтелекту Ебрахімом Мамдані.

Розглянемо задачу визначення дозування препарату із використанням алгоритму нечіткого логічного виведення Мамдані. Рішення про величину дозування препарату за основі двох критеріїв:

1. Температура;
2. Вага пацієнта.

Кожному критерію буде відповідати своя лінгвістична змінна. Для обох вхідних лінгвістичних змінних приймемо діапазон від 0 до 1. Обидві вхідні лінгвістичні змінні розділимо на 2 терм множини: мала та велика (small, big).

Кожній терм множині відповідатиме своя функція належності. Для вхідних змінних приймемо трапецієвидні функції належності (trapmf) із наступними параметрами зазначеними у табл. 3.1.

Таблиця 3.1. Параметри функцій належності вхідних змінних «Температура» та «Вага»

Терми	Вхідні лінгвістичні змінні	
	Температуру	Вага
small	[-1 -1 0.1 0.9]	[-1 -1 0.1 0.9]
big	[0.1 0.9 2 2]	[0.1 0.9 2 2]

У якості вихідної змінної буде доза. Універсум даної лінгвістичної змінної розділимо на три терм множини, що відповідають малій, середній та великій дозам (Small, Medium, Big). Кожній терм множині відповідатиме своя функція належності, у нашому випадку прийняті трикутні функції належності.

Приймаємо, що розмір дози змінюватиметься в інтервалі від 0 до 10 одиниць, а діапазон для вихідної лінгвістичної змінної приймемо від -5 до 15, таким чином, щоб він включав носії усіх функцій належності даної змінної.

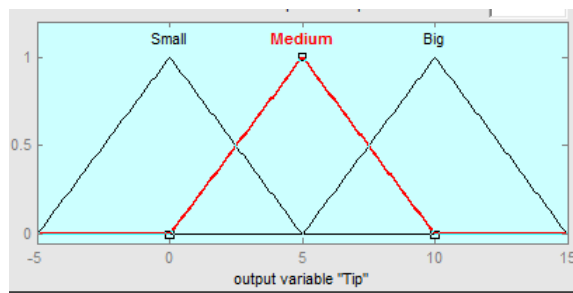


Рис. 3.1. Графічне відображення функцій належності вихідної змінної «доза»

Параметри функцій належності вихідної змінної зведемо до табл 3.2.

Таблиця 3.2. Параметри функції належності змінної «Доза»

Small	Medium	Big
[-5 0 5]	[0 5 10]	[5 10 15]

Правила будемо формувати у вигляді: ЯКЩО «Температура» ТА «Вага», то «Доза». Базу правил сформуємо у вигляді табл 3.3.

Таблиця 3.3. Табличний вигляд бази правил

Номер правила	Температура	Вага	Доза
1	small	small	Small
2	small	big	Medium
3	big	small	Medium
4	big	big	Big

У випадку двох вхідних змінних базу правил зручно показувати у вигляді матриці, табл. 3.4.

Таблиця 3.4. Матричний вигляд бази правил

		Вага	
		small	big
Температура	small	Small	Medium
	big	Medium	Big

Розглянемо етапи нечіткого логічного виведення за розробленою нечіткою моделлю. Для прикладу, приймемо значення температури на рівні 0,3, а ваги – 0,6.

Етап 1. Фазифікація вхідних змінних – приведення їх до нечіткості: інтерпретація точного значення вхідної змінної як нечіткого. Метою виконання даного етапу є отримання значень істинності для відповідних підумов із бази правил.

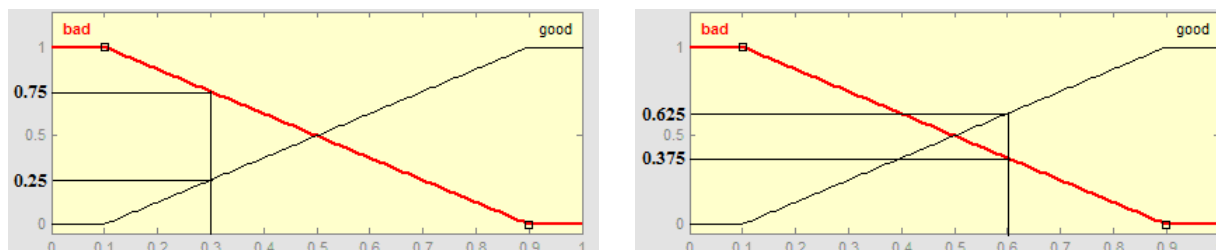


Рис. 3.2. Графічне відображення фазифікації вхідних змінних для заданих вхідних значень

Також представимо результати фазифікації у вигляді табл. 3.5

Таблиця 3.5. Результат фазифікації вхідних змінних

Чисельне значення температури = 0,3		
Терм	small	big
Відповідність чисельного значення терму	0,75	0,25
Чисельне значення ваги = 0,6		
Терм	small	big
Відповідність чисельного значення терму	0,375	0,625

Етап 2. Активація підумов – композиція вхідної змінної і умовної частини правила: обчислення рівня придатності правила до ситуації. У нашому випадку підумови у кожному правилі зв’язані оператором кон’юнкції (функція мінімуму), а метою виконання даного етапу є визначення істинності виконання правила, шляхом віднаходження мінімального значення істинності усіх його підумов. Результати зведемо до таблиці 3.6.

Таблиця 3.6. Результат етапу активації

Номер правила	Температура	Вага	Результат активації
1	small (0,75)	small (0,375)	0,375
2	small (0,75)	big (0,625)	0,625
3	big (0,25)	small (0,375)	0,25
4	big (0,25)	big (0,625)	0,25

Етап 3. Імплікація. На даному етапі здійснюється перехід від умов до підвисновків. Для кожного підвисновку розраховується значення істинності як добуток істинності правила на його ваговий коефіцієнт.

Таблиця 3.7. Результат етапу імплікації

Номер правила	Доза	Результат активації	Вагові коефіцієнти правил	Результат імплікації	Графічний результат
1	Small	0,375	1	0,375	
2	Medium	0,625	1	0,625	
3	Medium	0,25	1	0,25	
4	Big	0,25	1	0,25	

Етап 4. Агрегування – акумуляція підвисновків: побудова нечіткого значення виходу за результатами попередніх етапів. Для вихідної лінгвістичної змінної розраховується значення функції належності як максимум значень підвисновків.

Таблиця 3.8. Результат етапу агрегування

Доза	Результати імплікації	Результат агрегування
Small	0,375	0,375
Medium	0,625; 0,25	0,625
Big	0,25	0,25

В результаті агрегації отримуємо нечітке значення виходу, представимо його у графічній формі:

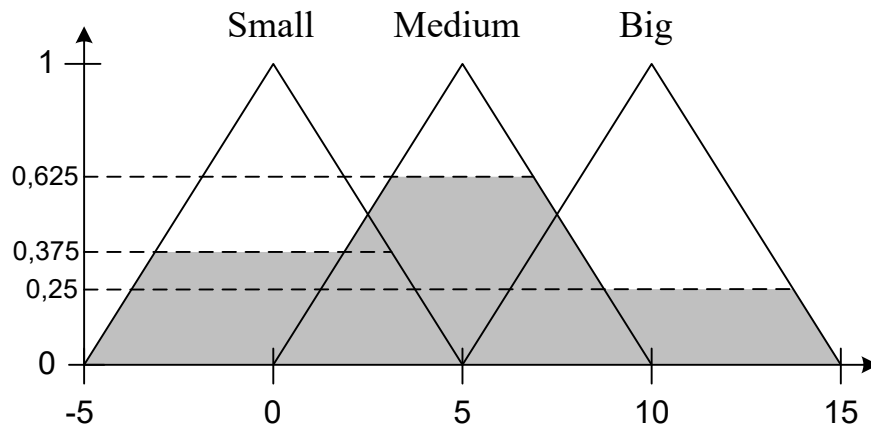


Рис. 3.3. Графічне відображення агрегування

Етап 5. Дефазифікація вихідного значення полягає у отриманні кількісного значення вихідної змінної за її нечітким значенням, отриманим в результаті агрегації. Дефазифікацію виконаємо за методом центру ваги (знайдемо координату центру ваги плоскої фігури).

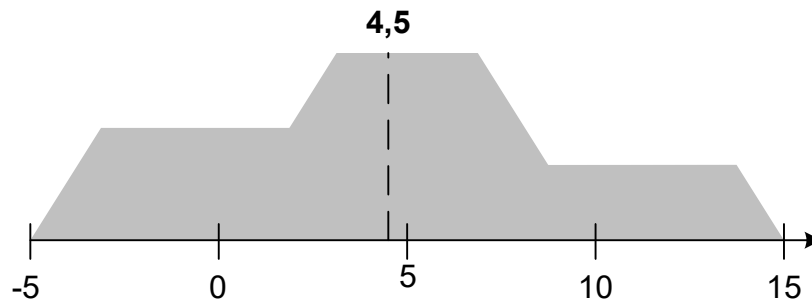


Рис. 3.4. Графічне відображення дефазифікації

Результатом дефазифікації у нашому випадку буде значення **4,5**.

Завдання роботи

Розробити модель нечіткого логічного виведення за алгоритмом Мамдані для передбачення розвитку цукрового діабету другого типу.

Лабораторна робота 4. Проектування систем нечіткого виводу Сугено для вирішення задач біомедичної інженерії

Мета роботи: освоїти методику проектування системи нечіткого виводу на основі розробки та використання баз знань продукційних правил з використанням алгоритму Сугено для вирішення задач біомедичної інженерії.

Теоретичні відомості

Розглянемо основні етапи проектування систем нечіткого виводу за алгоритмом Сугено на прикладі задачі візуалізації поверхні, яка реалізує деяку залежність параметрів дозування препарату $y = (x_1^2 - 8) \cos x_2$ на відрізку $x_1 \in [0, 4]$; $x_2 \in [0, 4]$. Відмінність СНВ алгоритму Сугено полягає у проектуванні вихідних змінних. Формування бази правил систем нечіткого виводу наступного формату:

Правило <#>: **Якщо** «змінна_1=значення_A» **і** «змінна_2=значення_B»
Тоді « $y = k_1A + k_2B + k_0$ »

або

Правило <#>: **Якщо** «змінна_1 = значення_A» **і** «змінна_2 = значення_B»
Тоді «змінна_u=значення_C»

Метод проектування та використання систем за алгоритмом Сугено

Моделювання заданої поверхні будемо реалізовувати за допомогою наступних правил бази знань:

1. Якщо x_1 та x_2 низькі, тоді $y = -10$

2. Якщо x_1 низьке, тоді $y = 3,75x_2 - 10$
3. Якщо x_1 низьке і x_2 вище середнього, тоді $y = 7$
4. Якщо x_1 та x_2 високі, тоді $y = -10$
5. Якщо x_2 низьке, тоді $y = 4x_1 - 10$
6. Якщо x_1 високе, тоді $y = 15 - 3,75x_2$
7. Якщо x_2 високе, тоді $y = 15 - 3,75x_1$
8. Якщо x_1 низьке і x_2 високе, тоді $y = 7$
9. Якщо x_1 середнє і x_2 середнє, тоді $y = 7$
10. Тоді проектування системи нечіткого виводу типу Сугено лежить у виконанні наступної послідовності кроків.

Крок 1. Для завантаження основного fis-редактору надрукуємо слово fuzzy у командному рядку Matlab.

Крок 2. Оберемо тип системи. Для цього в меню File в підменю New fis... оберемо команду Sugeno.

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

Крок 4. Перейменуємо першу вхідну змінну. Для цього зробимо одне натиснення лівої кнопки миші на блоці input1, введемо нове позначення x_1 в полі редагування імені поточної змінної і натиснемо <Enter>.

Крок 5. Перейменуємо другу вхідну змінну. Для цього зробимо одне натиснення лівою кнопкою миші на блоці input2, введемо нове позначення x_2 в полі редагування імені поточної змінної і натиснемо <Enter>.

Крок 6. Перейменуємо вихідну змінну. Для цього зробимо одне натиснення лівою кнопкою миші на блоці output1, введемо нове позначення y в полі редагування імені поточної змінної і натиснемо <Enter>.

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

Крок 8. Перейдемо в редактор функцій належності. Для цього зробимо швидке подвійне натиснення лівої кнопки миші на блоці x_1 .

Крок 9. Задаємо діапазон змін змінної x_1 . Для цього надрукуємо 0 4 в полі Range і натиснемо <Enter> (див. рис.4.1).

Крок 10. Задамо функції належності змінної x_1 . Для лінгвістичної оцінки цієї змінної будемо використовувати 3 терми з трикутними функціями належності, які встановлені за замовченням. Задамо найменування термів змінної x_1 . Для цього робимо одне натиснення лівою кнопкою миші по графіку першої функції належності.

Крок 11. Аналогічно задамо функції належності змінної x_2 . Для лінгвістичної оцінки цієї змінної будемо використовувати 4 терми з трикутними функціями належності. Для цього активізуємо змінну x_2 за допомогою натиснення лівої кнопки миші на блоці x_2 . Задамо діапазон змін x_2 . Для цього надрукуємо 0 4 в полі Range і натиснемо <Enter>. Задамо найменування 4 термів (наприклад, L Низький, A Середній, HA Вище середнього, H Високий).

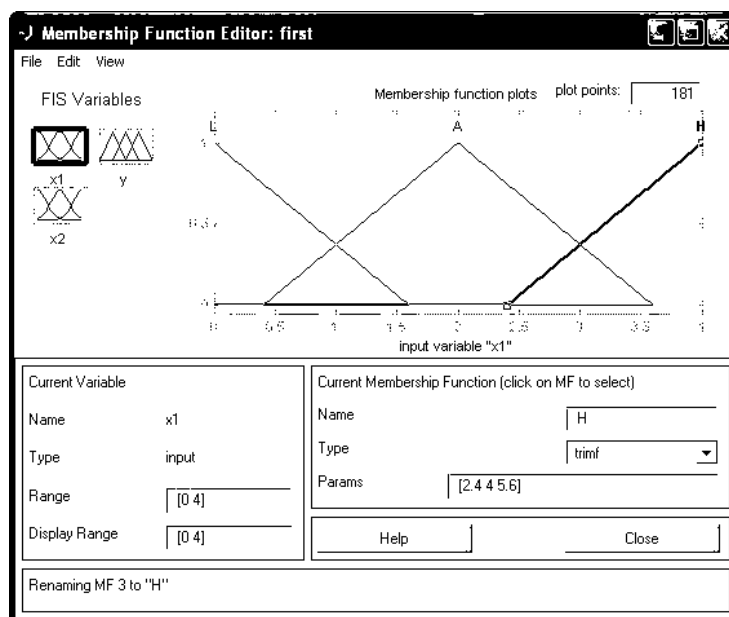


Рис 4.1. Функції належності змінної x_1

Крок 12. Задамо лінійні залежності між входами і виходом, яке наведене в базі знань. Для цього активуємо змінну y за допомогою натиснення лівої кнопки на блоці y . В правому верхньому куті можуть з'явитися позначення функцій належності, кожна з яких відповідає однієї лінійної залежності між входами і виходом. В базі знань, яка наведена на початку файлу, вказані 6 різних залежностей: $y=-10$; $y=7$; $y=3,75 x_1 -10$; $y=4 x_1 -10$; $y=-3,75 x_1 +15$; $y=-3,75 x_2 +15$. Додамо ще необхідну кількість функцій залежності шляхом обирання команди Add Mfs... меню Edit.

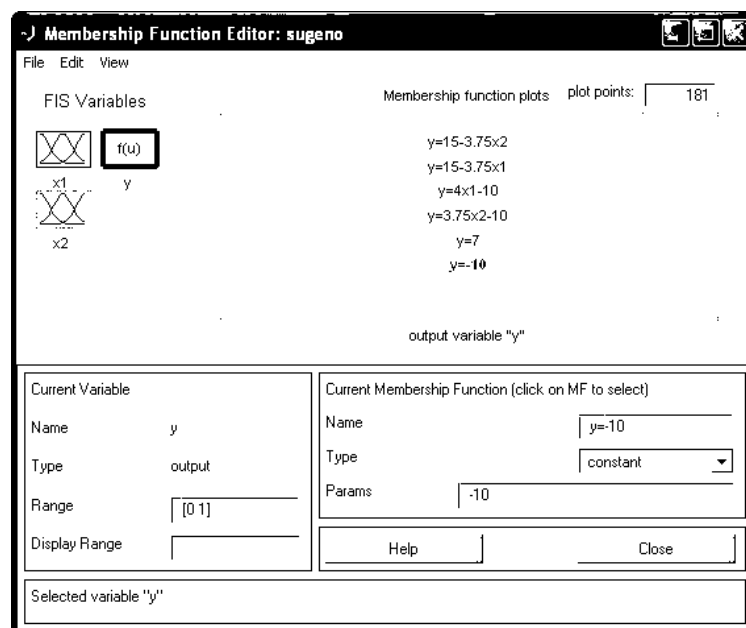


Рис 4.2. Вікно лінійних залежностей «входи-вихід»

Крок 13. Задамо найменування і параметри цих залежностей. Для цього робимо одне натиснення лівою кнопкою миші по імені першої залежності $mf1$. Потім друкуємо назву залежності, наприклад $y=-10$, в полі Name, і встановлюємо тип залежності – константа шляхом обирання опції Constant в меню Type. Після цього вводимо значення параметру – -10 в полі Params. Аналогічну процедуру робимо для другої змінної $y=7$.

Для третьої функції $mf3$ введемо найменування, наприклад, $y=3.75x_1-10$. Потім вкажемо лінійний тип залежності шляхом вибору опції Linear в меню Type і введемо параметри залежності $3.75\ 0\ -10$ в полі Params. Для лінійної

залежності порядок параметрів наступний: перший параметр – коефіцієнт при першій змінній, другий – при другій і т.д., останній параметр – вільний член залежності. Таким ж чином введемо назви і параметри для всіх 6 функцій належності змінної y .

В результаті отримуємо графічне вікно, яке представлено на рис. 4.2.

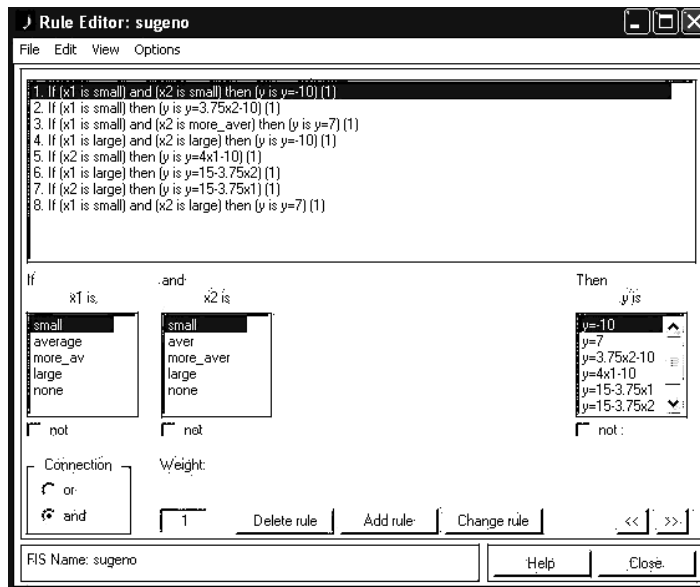


Рис 4.3. Нечітка база знань для системи типу Сугено

Крок 14. Перейдемо в редактор бази знань RuleEditor. Для цього оберемо в меню Edit команду Edit rules.... і введемо правила. Для вводу правила необхідно обрати відповідну комбінацію термів і залежностей і натиснути кнопку Add rule. На рис. 4.3 зображене вікно редактору бази знань після введення усіх 6 правил.

На рис. 4.4 приведено вікно візуалізації нечіткого логічного виводу. Це вікно активується командою View rules... меню View. В полі Input вказуються значення вхідних змінних, для яких виконується логічний вивід. Як можна побачити з рисунку, значення вихідної змінної, розраховується як середнє зважене значення результатів виходу за кожним правилом.

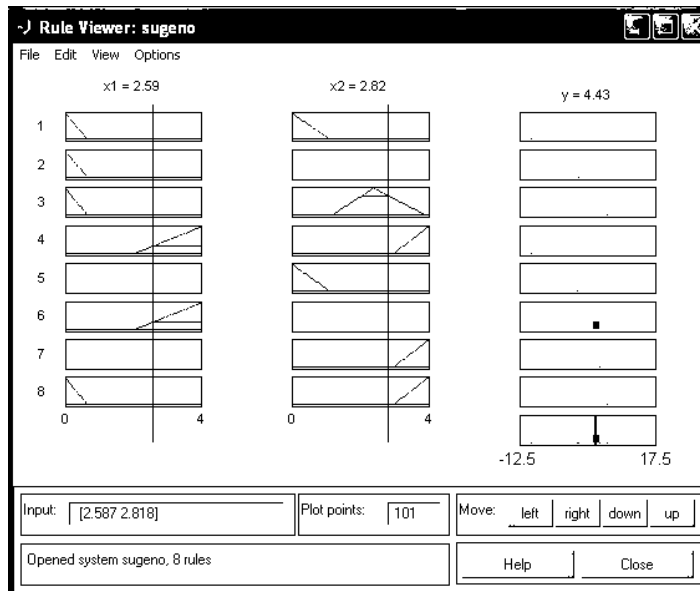


Рис. 4.4 Візуалізація нечіткого логічного виводу для системи типу Сугено

На рис. 4.5 приведена поверхня «входи-вихід», яка відповідає синтезованій нечіткій системі. Для виведення цього вікна необхідно використати команду View surface... меню View. Перевага моделей типу Мамдані полягає у тому, що правила бази знань є прозорі і інтуїтивно зрозумілі, таді як для моделей типу Сугэно не завжди ясно які лінійні залежності «входи-вихід» необхідно використовувати.

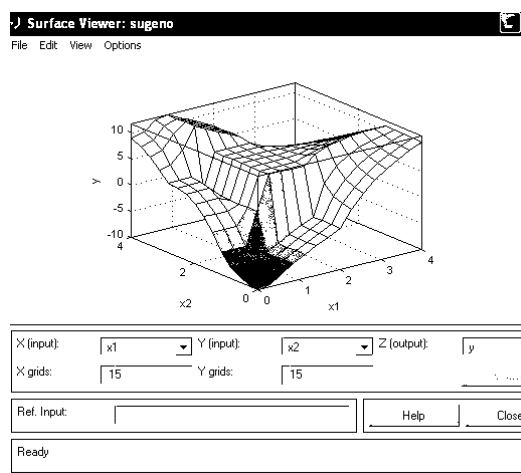


Рис 4.5. Поверхня «вхід-вихід» для системи алгоритму Сугено

Завдання

Розробити модель нечіткого логічного виведення за алгоритмом Сугено для вирішення завдання із попередньої лабораторної роботи.

Лабораторна робота 6. Автоматичне проектування систем нечіткого виводу Мамдані для систематизації біомедичної інформації

Мета роботи: освоїти методику автоматичного проектування системи нечіткого виводу Мамдані для систематизації біомедичної інформації

Теоретичні відомості

Одна із найскладніших задач дослідження причинно-наслідкових зв'язків розвитку захворювань полягає у їх формалізації. Нараз існує велика кількість репозиторіїв біомедичної інформації із вільним доступом. Інформація у таких системах може характеризуватися значною неточністю, тому класичні методи її апроксимації можуть мати значні неточності. Апарат нечіткої логіки у даному випадку використовувати доцільно, адже він дозволяє невілювати неточності.

Завдання даної лабораторної роботи полягає у розробці сценарію автоматичного синтезу моделі нечіткого логічного виведення за алгоритмом Мамдані на основі біомедичних даних у табличному вигляді.

Лабораторна робота 7. Автоматичне проектування систем нечіткого виводу Мамдані із використанням кластерного аналізу для систематизації біомедичної інформації

Мета роботи: освоїти методику автоматичного проектування системи нечіткого виводу Мамдані із формуванням термів лінгвістичних змінних на основі кластерного аналізу для систематизації біомедичної інформації

Теоретичні відомості

Рівномірне розташування функцій належності лінгвістичних змінних, як було запропоновано у попередній лабораторній роботі, призводить до не оптимального формування баз правил та термів лінгвістичних змінних. Терми необхідно розташовувати відповідно до концентрації табличних значень на універсумі. Із цією метою пропонується розподіл термів лінгвістичних змінних виконувати на базі кластерного аналізу.

Лабораторна робота 8. Дослідження впливу методу дефазифікації при оптимізації моделі інтерпретації біомедичної інформації на основі нечіткого логічного висновку

Мета роботи: ознайомитися із впливом методів дефазифікації у моделі нечіткого логічного виведення за алгоритмом Мамдані на її можливості апроксимації біомедичної інформації із використанням пакету Optimization toolbox. Вивчити застосування пакету Optimization toolbox для налаштування можелей за алгоритмами Сугено для апроксимації біомедичної інформації.

Лабораторна робота 9. Розробка біотехнічної системи із регулятором на базі нечіткого логічного висновку

Мета роботи: навчитися моделювати біотехнічні системи із інтелектуальними регуляторами на базі нечіткої логіки.

Теоретичні відомості

Завдання лабораторної роботи полягає у розробці моделі системи керування на базі нечіткої логіки до складу якої входить біологічний об'єкт. На прикладі регулятора подачі кисню у маску пацієнта від сатурації для підтримки останньої на заданому рівні.