

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Інститут автоматики, електроніки та комп'ютерних систем управління

(повне найменування інституту, назва факультету (відділення))

Кафедра комп'ютерних систем управління

(повна назва кафедри (предметної, циклової комісії))

Пояснювальна записка

до бакалаврської дипломної роботи

бакалавр

(освітньо-кваліфікаційний рівень)

на тему Ідентифікація залежностей за допомогою нечіткої бази знань з
правилами у різних форматах

Виконав: студент 4 курсу, групи 3СІ-106

Напрямок підготовки

6.050201- “Системна інженерія”

(шифр і назва напрямку підготовки, спеціальності)

Тилець Р. О.

(прізвище та ініціали)

Керівник Штовба С.Д.

(прізвище та ініціали)

Рецензент Папінов В. М.

(прізвище та ініціали)

Вінниця ВНТУ - 2014 року

Вінницький національний технічний університет

(повне найменування вищого навчального закладу)

Інститут автоматики, електроніки та комп'ютерних систем управління

Кафедра комп'ютерних систем управління

Освітньо-кваліфікаційний рівень бакалавр

Напрямок підготовки 6.050201 - "Системна інженерія"

(шифр і назва)

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ

В.М. Дубовой

“ 05 ” травня 2014 року

ЗАВДАННЯ НА БАКАЛАВРСЬКУ ДИПЛОМНУ РОБОТУ СТУДЕНТУ

Тилець Роман Олексійович

(прізвище, ім'я, по батькові)

1. Тема бакалаврської дипломної роботи Ідентифікація залежностей за допомогою нечіткої бази знань з правилами у різних форматах

керівник бакалаврської дипломної роботи Штовба С. Д., д.т.н., проф.,
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від “ 14 ” лютого 2014 року № 31

2. Строк подання студентом роботи 17 червня 2014

3. Вихідні дані до роботи а) Штовба С. Д. Проектирование нечетких систем средствами MATLAB / С. Д. Штовба – М.: Горячая линия – Телеком, 2007. – 288 с.- опис функціональних можливостей математичного середовища Матлаб для роботи з нечіткими базами знань; б) Штовба С. Д. Идентификация нелинейных зависимостей с помощью нечеткого логического вывода в системе MATLAB // С. Д. Штовба - Exponenta Pro: Математика в приложениях. – 2003. – № 2. – С. 9-15. – практичне застосування засобів матлаб для ідентифікації за допомогою нечіткої бази знань; в) Ротштейн А. П. Идентификация нелинейной зависимости нечеткой базой знаний с нечеткой обучающей выборкой // А. П. Ротштейн, С. Д. Штовба - Кибернетика и системный анализ. – 2006. – № 2. – С. 17-24. – настройка нечіткої бази знань г) Ротштейн А. П. Идентификация нелинейных зависимостей нечеткими базами знаний // А. П. Ротштейн, Д. И. Кательников - Кибернетика и системный анализ. – 1998. – № 5. - теорія по ідентифікації залежностей з допомогою нечітких БЗ.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Обґрунтування доцільності ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах; 2. Розробка математичної моделі нечіткої бази знань з правилами у різних форматах; 3. Розробка алгоритмічного та програмного забезпечення 4. Тестування та оцінка ефективності програмного забезпечення 5. Охорона праці

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень) графічна інтерпретація логічного виводу по базам знань з правилами у різних форматах, блок-схеми алгоритмів ідентифікації за допомогою баз знань з правилами у різних форматах, UML-діаграма розгортання програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах, UML-діаграма послідовності програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах, UML-діаграма варіантів використання програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах

6. Консультанти розділів бакалаврської дипломної роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1-4	Штовба С. Д., проф.		
5	Дембіцька С. В., доц.		

7. Дата видачі завдання 05 травня 2014 року

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської дипломної роботи	Строк виконання етапів роботи	Примітка
1.	Обґрунтування доцільності ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах	20 травня	
2.	Розробка математичної моделі нечіткої бази знань з правилами у різних форматах	28 травня	
3.	Розробка алгоритмічного та програмного забезпечення	10 червня	
4.	Тестування та оцінка ефективності програмного забезпечення	18 червня	
5.	Охорона праці	18 червня	

Студент

(підпис) (прізвище та ініціали)

Керівник бакалаврської дипломної роботи

(підпис) (прізвище та ініціали)

ЗМІСТ

АННОТАЦІЯ	6
ABSTRACT	7
ВСТУП	8
1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ІДЕНТИФІКАЦІЇ ЗАЛЕЖНОСТЕЙ ЗА ДОПОМОГОЮ НЕЧІТКОЇ БЗ З ПРАВИЛАМИ У РІЗНИХ ФОРМАТАХ.....	10
1.1 Аналіз нечітких множин та нечітких баз знань	10
1.2 Опис об'єкта проектування та постановка задачі.....	18
1.3 Варіантний огляд проектних рішень.....	18
1.4 Висновки до розділу	22
2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ НЕЧІТКОЇ БАЗИ ЗНАНЬ З ПРАВИЛАМИ У РІЗНИХ ФОРМАТАХ	24
2.1 Модель бази знань з правилами Мамдані та Ларсена	24
2.2 Модель бази знань з сингтонними правилами та правилами Сугено і Цукамото	25
2.3 Випадок зі змішаними правилами.....	26
2.3.1 Аналіз проблеми створення бази знань зі змішаними правилами....	26
2.3.2 Вирішення проблеми створення БЗ зі змішаними правилами	26
2.4 Висновки до розділу	27
3 РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.	28
3.1 Створення бази знань за допомогою вбудованих функцій Matlab	28
3.2 Розробка програмної реалізації бази знань з правилами в форматі Ларсена та Мамдані.....	33
3.3 Розробка програмної реалізації бази знань з сингтонними правилами та правилами у форматі Сугено.....	33

3.4 Розробка програмної реалізації бази знань з сингтонними правилами та правилами у форматі Сугено, Ларсена, Мамдані	34
3.5 Опис основних функцій використаних при розробці програмного забезпечення.....	34
3.5.1 Функція Evalfis	34
3.5.2 Функція Defuzz.....	36
3.5.3 FIS-редактор	36
3.5.4 Функція eval_mam_lars.....	37
3.5.5 Функція eval_mam_sug	37
3.5.6 Функція eval_singl_sug	38
3.6 Висновки до розділу	39
4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	40
4.1 Тестування та оцінка ефективності ідентифікації залежностей за допомогою бази знань з правилами типу Мамдані	40
4.2 Тестування та оцінка ефективності ідентифікації залежностей за допомогою нечіткої бази знань з правилами типу Мамдані та Сугено.....	44
4.3 Аналіз результатів та висновки до розділу	52
5 ОХОРОНА ПРАЦІ	54
5.1 Технічні рішення щодо безпечного виконання роботи.....	55
5.1.1 Обладнання приміщення	55
5.1.2 Ергономічні вимоги до робочого місця розробника	55
5.2. Технічні рішення з виробничої санітарії	57
5.2.1 Мікроклімат	57
5.2.2 Склад повітря робочої зони.....	58
5.2.3 Виробниче освітлення	59

	5
5.2.4 Виробничий шум.....	60
5.2.5 Виробничі випромінювання.....	61
5.3 Пожежна безпека.....	63
5.3.1 Технічні рішення системи запобігання пожежі	64
5.3.2. Технічні рішення системи протипожежного захисту	65
ВИСНОВКИ.....	67
ПЕРЕЛІК ПОСИЛАНЬ	69
ДОДАТКИ.....	72
Додаток А. Технічне завдання	73
Додаток Б. Графічна інтерпретація математичної моделі бази знань з правилами Мамдані та Ларсена	76
Додаток В. Графічна інтерпретація математичної моделі бази знань з сингтонними правилами та правилами Сугено і Цукамото.....	77
Додаток Г. Графічна інтерпретація математичної моделі бази знань зі змішаними правилами.....	78
Додаток Г. Лістинги програм	79
Додаток Д. Блок-схема програми ідентифікації залежностей за допомогою бази знань з правилами Ларсена та Мамдані	87
Додаток Ж. Блок-схема програми ідентифікації залежностей за допомогою бази знань з сингтонними правилами та правилами у форматі Сугено	88
Додаток К. Блок-схема програми ідентифікації залежностей за допомогою бази знань зі змішаними правилами.....	89
Додаток Л. UML-діаграма розгортання ПЗ	90
Додаток М. UML-діаграма послідовності ПЗ ідентифікації за допомогою баз знань з правилами у різних форматах	91
Додаток Н. UML - діаграма варіантів використання ПЗ.....	92

АННОТАЦІЯ

В роботі був проведений огляд нечітких множин та нечітких баз знань. Також був проведений загальний аналіз нечітких баз знань з різними типами правил (Сугено, Мамдані, Цукамото, синглтонна, Ларсена). Було проведено варіантний огляд і оцінку проектних рішень, в ході якого для розробки вибрано математичне середовище Matlab. Були розроблені алгоритми та програмне забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах. Програмне забезпечення було протестоване. Результати тестування підтверджують правильність роботи ПЗ.

ABSTRACT

In bachelor thesis was reviewed fuzzy sets and fuzzy knowledge bases. There was held a general analysis of fuzzy knowledge bases with different types of rules (Sugeno, Mamdani, Tsukamoto, singleton, Larsen). There have been a variant review and evaluate design solutions in which to develop the selected mathematical environment Matlab. Were developed algorithms and software to identify dependencies using fuzzy knowledge base of rules in different formats. The software has been tested. Test results confirm the correct work of the software.

ВСТУП

Актуальність теми роботи. Кожний рік людство виробляє і накопичує велику кількість знань. Для зручного зберігання і використання цієї інформації використовуються бази знань. База знань - це особливого роду база знань, розроблена для управління знаннями, тобто збором, зберіганням, пошуком і видачею знань. База знань — важливий компонент інтелектуальної системи. Найвідоміший клас таких програм — експертні системи. Вони призначені для знаходження способу вирішення специфічних проблем, базуючись на записах БЗ і на користувацькому описі ситуації [1].

Як правило певні категорії об'єктів належать певній множині з деяким ступенем належності. На цьому принципі базується нечітка логіка [2].

Проблеми які вирішують нечіткі бази знань — моделювання, ідентифікація залежностей, прийняття рішень. Використання нечіткого логічного виводу дозволяє значно зменшити обсяги обчислень [3].

Перевагою нечіткої бази знань яка використовує правила у різних форматах є те, що вона для ідентифікації залежності використовує ті правила, які найбільше підходять для певних типів вхідних даних і це дозволяє отримати більш точний вихід.

Нечіткі системи в даний час використовуються у багатьох технічних засобах, таких як: пральні машини, автомобілі, безпілотні літаючі засоби, космічні апарати та інші.

Мета роботи. Підвищення точності ідентифікації залежностей за допомогою використання бази знань з правилами у різних форматах.

Об'єктом дослідження є система управління з базою знань, яка включає правила у різних форматах.

Методи дослідження базуються на нечіткій логіці, обчислювальних методах.

Новизна роботи. Дана робота є першою роботою в якій проведена ідентифікація залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

Практичне значення роботи. Розроблене програмне забезпечення яке дозволяє автоматизувати складні процеси які пов'язані з соціологією, медициною, політикою (побудова соціальних моделей і т.п.).

Публікації. За темою дипломної бакалаврської роботи була проведена апробація та зроблена доповідь на «XLIII регіональній науково-технічній конференції професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області» [4].

Обсяг і структура роботи. Кількість розділів 5, рисунків 31, літературних джерел 37.

1 ОБҐРУНТУВАННЯ ДОЦІЛЬНОСТІ ІДЕНТИФІКАЦІЇ ЗАЛЕЖНОСТЕЙ ЗА ДОПОМОГОЮ НЕЧІТКОЇ БЗ З ПРАВИЛАМИ У РІЗНИХ ФОРМАТАХ

1.1 Аналіз нечітких множин та нечітких баз знань

В даному розділі, ґрунтуючись на [5, 6, 7, 8, 10, 11], наведені основні теоретичні відомості, які потрібні для подальшої розробки математичної моделі та програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

Основою теорії нечітких множин є ідея про те, що елементи з певної множини, які володіють деякою спільною властивістю, можуть володіти нею з різним ступенем, і, відповідно, належати до множини з різним ступенем [5]. При такому підході висловлювання типу «елемент належить певній множині» втрачає сенс, тому що необхідно вказати наскільки сильно, з яким ступенем елемент задовольняє властивостям множини.

Нечіткою множиною $\tilde{A}$ на універсальній множині U називається сукупність пар $(\mu_A(u), u)$, де $\mu_A(u)$ – ступінь належності елемента $u \in U$ до нечіткої множини $\tilde{A}$. Ступінь належності – це число з діапазону від 0 до 1 включно. Чим вище ступінь належності, тим більшою мірою елемент універсальної множини відповідає властивостям нечіткої множини [6].

Функцією належності називається така функція, яка дозволяє обчислити ступінь належності довільного елемента універсальної множини до нечіткої множини [7].

Якщо універсальна множина складається з кінцевого числа елементів, тоді $U = \{u_1, u_2, \dots, u_k\}$ нечітка множина $\tilde{A}$ записується так: $\tilde{A} = \left(\frac{\mu_A(u_1)}{u_1}, \frac{\mu_A(u_2)}{u_2}, \dots, \frac{\mu_A(u_k)}{u_k} \right)$.

Наприклад, нечітку множину «середня ціна на молоко у першій половині 2014 року» можна задати такою нечіткою множиною $\tilde{A} = \left(\frac{0,1}{6}, \frac{0,3}{6,5}, \frac{0,5}{7,1}, \frac{1}{8}, \frac{0,5}{8,9}, \frac{0,1}{10} \right)$.

При неперервній множині U використовують таке позначення: $\tilde{A} = \int \frac{\mu_A(u)}{u}$, де знак інтегрування означає сукупність пар $\mu_A(u)$ та u .

Основні операції які проводять над двома або більше нечіткими множинами: операція об'єднання ($\cup, \vee$), операція перетину ($\cap, \wedge$), операція доповнення. Інтерпретація цих операцій для нечітких множин проводиться за допомогою трикутних норм, найбільш вживаними є норми Заде, Ймовірнісні норми, норми Лукасевича. Найбільш уживані трикутні норми наведено у таблиці 1.1, в ній a, b – відповідні функції належності нечітких множин $\tilde{A}, \tilde{B}$.

Таблиця 1.1 – Найбільш уживані трикутні норми

t-норма: $a \wedge b$	s-норма: $a \vee b$	Назва
$\min(a, b)$	$\max(a, b)$	Норми Заде
$a * b$	$a + b - a * b$	Ймовірнісні норми
$\max(a + b - 1, 0)$	$\min(a + b, 1)$	Норми Лукасевича

Нечіткою базою знань називається сукупність нечітких правил «Якщо – тоді», які описують певну предметну область. Сьогодні найуживанішими є нечіткі сингтонна база знань, класифікаційна база знань, база знань Мамдані, база знань Сугено, база знань Ларсена та база знань Цукамото [8]. Антецедентами або посилками називають вхідні дані, а консеквентами – вихідні.

Для простого користувача база знань – це чорний ящик, який приймає вхідні дані і обробляючи їх формує результат, причому сам процес формування результату для користувача невідомий.

Сингтонну базу знань складають правила, антецеденти яких задані нечіткими множинами, а консеквенти – дійсними числами [10]:

$$\text{Якщо } (x_1 = \tilde{a}_{1j} \ominus_j x_2 = \tilde{a}_{2j} \ominus_j \dots \ominus_j x_n = \tilde{a}_{nj}) \text{ то } y = d_j, j = \overline{1, N}, \quad (1.1)$$

де $\tilde{a}_{ij}$ – нечіткий терм, яким оцінено фактор x_i в j -му правилі $i = \overline{1, n}$, $j = \overline{1, N}$;

N – кількість правил;

$\ominus_j$ – логічна операція (І чи АБО), яка пов'язує фрагменти антецедента j -го правила;

d_j – консеквент j -го правила, який задано дійсним числом.

В базі знань (1.1) нечіткий терм $\tilde{a}_{ij}$ обирається з деякої скінченної терм-множини $T_i = \{\tilde{t}_{i1}, \tilde{t}_{i2}, \dots, \tilde{t}_{iz}\}$.

Якщо в базі знань (1.1) є лише логічні операції І, тоді її зручно подати у формі таблиці 1.2. Для адекватного опису $y = f(x_1, x_2, \dots, x_n)$ залежності в деякій області факторного простору не завжди необхідно знання значень усіх вхідних змінних. Для цих випадків правила бази знань можна представити в «неповному форматі», вилучивши входи, які не впливають на вихідне значення. Для врахування такої особливості в нечіткій базі знань позначають символом «-» змінні, що можуть приймати довільні значення без порушення істинності відповідного правила. Інколи для цього використовують нечітку множину Don't care [10], функція належності якої дорівнює 1 на усьому діапазоні значень відповідної вхідної змінної.

Таблиця 1.2 – Нечітка база знань з логічними операціями І

Якщо				Тоді
x_1	x_2	...	x_n	y
$\tilde{a}_{11}$	$\tilde{a}_{21}$	...	$\tilde{a}_{31}$	d_1
...				
$\tilde{a}_{1N}$	$\tilde{a}_{2N}$	...	$\tilde{a}_{nN}$	d_N

Ступені належності поточного вхідного вектора $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ до числових значень $d_1 \dots d_N$ розраховують так:

$$\mu_{d_j}(X^*) = \mu_j(x_1^*) \chi_j \mu_j(x_2^*), \chi_j \dots \chi_j, \mu_j(x_n^*), j = \overline{1, N}, \quad (1.2)$$

де $\mu_j(x_i^*)$ – ступінь належності значення x_i^* нечіткому терму $\tilde{a}_{ij}$;

χ_j позначає t-норму якщо в (1.2) $\ominus_j = I$ та s-норму якщо $\ominus_j = \text{АБО}$. Перелік найуживаніших трикутних норм наведено в таблиці 1.1.

Чітке значення на виході моделі розраховують через дефазифікацію нечіткої множини $\tilde{y} = \frac{\mu_{d_1}(X^*)}{d_1}, \frac{\mu_{d_2}(X^*)}{d_2}, \dots, \frac{\mu_{d_N}(X^*)}{d_N}$ за методом центра тяжіння.

Таблиця 1.3 – Дефазифікація різними методами

Метод дефазифікації	Нечітка множина	
	На неперервному носії	На дискретному носії
Центр тяжіння	$\int_{\underline{u}}^{\bar{u}} \frac{u * \mu_A(u) du}{\mu_A(u)}$	$\frac{\sum_{i=1}^k u * \mu_A(u_i)}{\sum_{i=1}^k \mu_A(u_i)}$
Медіана	знайти таке число a , щоб: $\int_{\underline{u}}^a \mu_A(u) du = \int_a^{\bar{u}} \mu_A(u) du$	$\min(U_j) \forall j: \sum_{i=1}^j \mu_A(u_i) \geq \frac{1}{2} \sum_{i=1}^k \mu_A(u_i)$

Базу знань Сугено складають правила, антецеденти яких задані нечіткими множинами, а консеквенти – лінійними функціями від входів. Її можна розглядати як розбиття факторного простору на нечіткі зони, в кожній з яких діє свій закон «входи – вихід». Границі зон розмиті, тому в будь-якій точці факторного простору можуть виконуватися декілька законів «входи – вихід», але з різними ступенями [10].

$$\text{Якщо } (x_1 = \tilde{a}_{1j} \ominus_j x_2 = \tilde{a}_{2j} \ominus_j \dots \ominus_j x_n = \tilde{a}_{nj}) \text{ то } y = b_{j_0} + b_{j_1}x_1 + \dots + b_{j_n}x_n, \quad j = \overline{1, N}, \quad (1.3)$$

де $b_{j_0}, b_{j_1} \dots b_{j_n}$ – дійсні числа.

Ступені належності вектора $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ поточних значень факторів впливу до консеквентів $d_j = b_{j_0} + b_{j_1}x_1 + \dots + b_{j_n}x_n$ розраховують так:

$$\mu_{d_j}(X^*) = \mu_j(x_1^*) \chi_j \mu_j(x_2^*), \chi_j \dots \chi_j, \mu_j(x_n^*), j = \overline{1, N}. \quad (1.4)$$

Звідси зрозуміло, що нечітка база знань Сугено для кожної точки факторного простору продукує унікальну сингтонну базу знань [10].

В алгоритмі Сугено найчастіше використовують ймовірнісні норми. Для вхідного вектора X^* результатом логічного виведення за усією базою знань буде така нечітка множина:

$$\tilde{y} = \frac{\mu_{d_1}(X^*)}{d_1}, \frac{\mu_{d_2}(X^*)}{d_2}, \dots, \frac{\mu_{d_N}(X^*)}{d_N}. \quad (1.5)$$

Результуюче значення вихідної змінної знаходять як суперпозицію лінійних законів, які виконуються в точці X^* n -мірного факторного простору. Для цього (1.5) дефаззифікують за методом центра тяжіння (таблиця 1.3).

В базі знань Мамдані антецеденти і консеквенти задано нечіткими множинами. Її можна трактувати як розбиття факторного простору на зони з нечіткими межами, в кожній з яких функція відклику приймає нечітке значення. Кількість нечітких зон дорівнює числу правил. Нечітку базу знань Мамдані запишемо таким чином [10]:

$$\text{Якщо } (x_1 = \tilde{a}_{1j} \ominus_j x_2 = \tilde{a}_{2j} \ominus_j \dots \ominus_j x_n = \tilde{a}_{nj}) \text{ то } y = \tilde{d}_j \text{ з вагою} \quad (1.6)$$

$$w_j, j = \overline{1, N},$$

де $\tilde{d}_j$ – нечітке значення;

$w_j \in [0,1]$ – ваговий коефіцієнт, який віддзеркалює впевненість експерта в адекватності j -го правила.

Логічне виведення за нечіткою базою знань Мамдані здійснюють за схемою наведеною на рисунку 1.1. Спочатку розраховують ступінь виконання антецедента j -го правила для поточного вхідного вектора $X^* = (x_1^*, x_2^*, \dots, x_n^*)$:

$$\mu_j(X^*) = w_j * (\mu_j(x_1^*) \chi_j \mu_j(x_2^*), \chi_j \dots \chi_j, \mu_j(x_n^*)), j = \overline{1, N}. \quad (1.7)$$

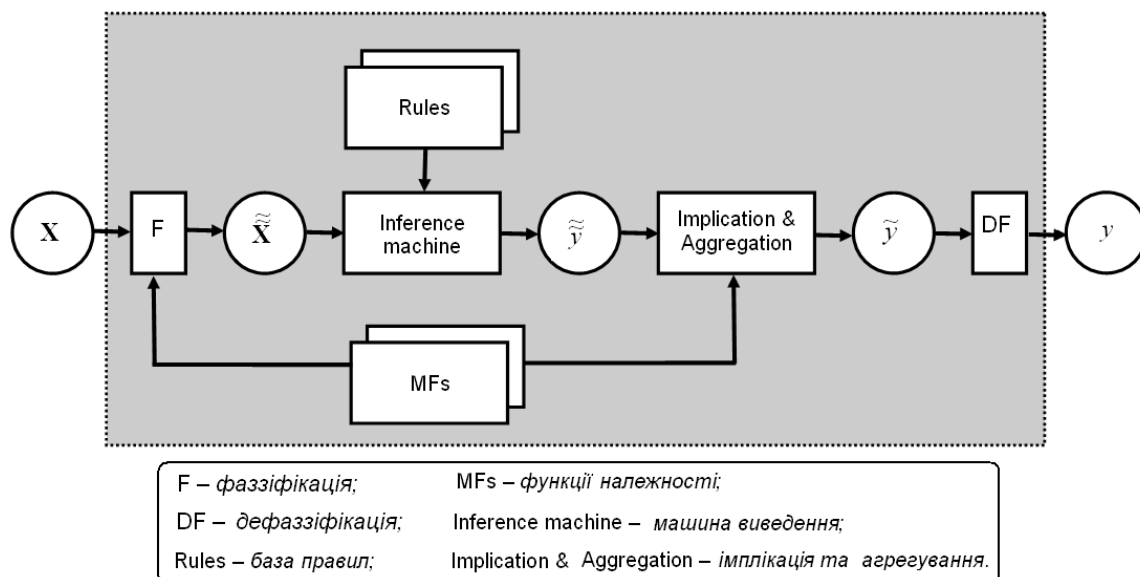


Рисунок 1.1 – Логічне виведення за нечіткою базою знань Мамдані

В алгоритмі Мамдані зазвичай використовують норми Заде. Результат логічного виведення можна записати у формі такої бінечіткої множини [11]:

$$\tilde{y}^* = \frac{\mu_{d_1}(X^*)}{\tilde{d}_1}, \frac{\mu_{d_2}(X^*)}{\tilde{d}_2}, \dots, \frac{\mu_{d_N}(X^*)}{\tilde{d}_N}. \quad (1.8)$$

Особливістю (1.8) є те, що елементи універсальною множини задані нечіткими множинами $\tilde{d}_1, \tilde{d}_2, \dots, \tilde{d}_N$. Для перетворення $\tilde{y}^*$ в звичайну нечітку множину виконаємо такі дії. Спочатку представимо результат виведення за j -им правилом бази знань у формі такої нечіткої множини:

$$\tilde{d}_j^* = \text{imp}(\tilde{d}_j, \mu_j(X^*)), j = \overline{1, N}, \quad (1.9)$$

де imp позначає імплікацію, яку реалізують операцією мінімуму.

Геометричною інтерпретацією імплікації є «зрізання» (див. рисунок 1.2) графіка функції належності $\mu_{d_j}(y)$ по рівню $\mu_j(X^*)$, що математично запишемо так:

$$\tilde{d}_j^* = \int_y^{\bar{y}} \min(\mu_j(X^*), \mu_{d_j}(y)) / y, j = \overline{1, N}. \quad (1.10)$$

Результат виведення за усіма правилами знаходять агрегуванням нечітких множин (1.11):

$$\tilde{y}^* = agg(\tilde{d}_1^*, \tilde{d}_2^*, \dots, \tilde{d}_n^*), \quad (1.11)$$

де agg – агрегування нечітких множин, яку реалізують операцією максимуму.

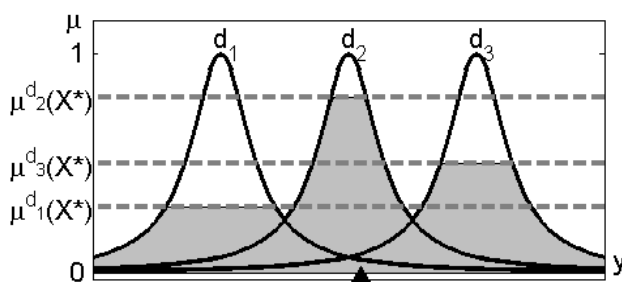


Рисунок 1.2 – Імплікація, агрегування та дефаззифікація в алгоритмі Мамдані

Чітке значення виходу y^* , яке відповідає вхідному вектору X^* , визначається через дефаззифікацію нечіткої множини $\tilde{y}^*$.

В базі знань Ларсена, як і в базі знань Мамдані [10], антецеденти і консеквенти задано нечіткими множинами. Відповідно, правила бази знань Ларсена можна представити в форматі (1.6). Початкові процедури логічного виведення для цих баз знань є тотожні. Відмінності полягають у виконанні імплікації. В базі знань Мамдані імплікація виконується через операцію мінімуму, тобто “зрізанням” функції належності $\mu_{d_j}(y)$ консеквента j -го правила по рівню $\mu_j(X^*)$, тобто по рівню виконання його антецедента. В алгоритмі Ларсена імплікація реалізується операцією множення, тобто функції належності $\mu_{d_j}(y)$ консеквента j -го правила множиться на значення $\mu_j(X^*)$ (рисунок 1.3). Далі, як і в алгоритмі Мамдані, проводиться агрегування нечітких висновків за усіма правилами і утворюється вихідна нечітка множина. Наступним кроком є дефаззифікація цієї вихідної нечіткої множини.

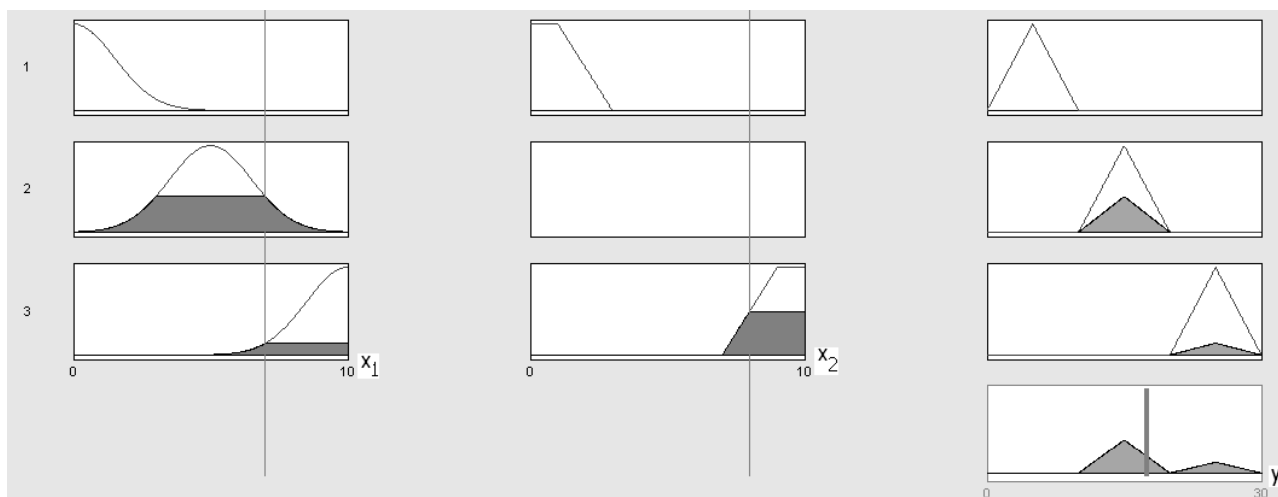


Рисунок 1.3 – Логічне виведення за алгоритмом Ларсена

В базі знань Цукамото, як і в базі знань Мамдані, антецеденти і консеквенти задано нечіткими множинами. Відмінність полягає у представленні консеквентів – їх функції належності мають бути монотонними. Зазвичай використовують 2 терма: «Низький» з монотонно спадаючою функцією належності та «Високий» з монотонно зростаючою функцією належності. Правила бази знань Цукамото можна представити в форматі (1.6). Початкові процедури логічного виведення – фаззіфікація та активація правил відбуваються так і для решти баз знань. Імплікація реалізується у такий спосіб [10]. Ступінь активації j -го правила - $\mu_j(X^*)$ ототожнюється із ступенем виконанням його антецедента. А чітке значення y_j^* поточного консеквента j -го правила знаходять з рівняння $\mu_{d_j}(y_j^*) = \mu_j(X^*)$.

Застосовуючи ці дії до усіх N правил, отримуємо таку нечітку множину на дискретному носії яка аналогічна до результуючої нечіткої множини в логічному виведенні за сингтонною нечіткою базою знань чи за базою знань Сугено:

$$\tilde{y} = \left(\frac{\mu_1(X^*)}{y_1^*}, \frac{\mu_2(X^*)}{y_2^*}, \dots, \frac{\mu_N(X^*)}{y_N^*} \right), \quad (1.12)$$

Чітке значення на виході отримуємо дефаззіфікацією нечіткої множини (1.12). Логічне виведення за нечіткою базою знань Цукамото показано на рисунку 1.4.

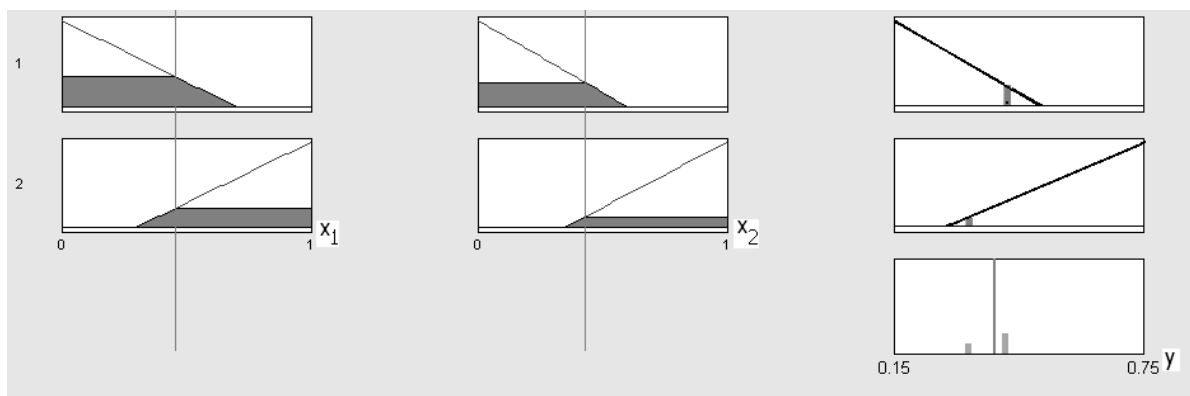


Рисунок 1.4 – Логічне виведення за алгоритмом Цукамото

1.2 Опис об'єкта проектування та постановка задачі

Згідно завдання до дипломного проекту потрібно розробити програмне забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

Існуючі методи ідентифікації працюють з базою знань з одним певним типом правил, приймають один тип вхідних даних.

Опис об'єкта проектування: існує багато задач де на практиці доцільніше використовувати вхідні змінні у різних форматах для зручності їх подання за допомогою висловлювань людини. Наприклад використання газу для опалення будинків в теплий період року можна описати нечіткою множиною «Відсутнє» (правила у форматі Мамдані), а в холодний період року використання газу буде описано певною залежністю (правила у форматі Сугено).

Постановка задачі: дано набір вхідних даних, потрібно розробити програмне забезпечення для ідентифікації залежностей за допомогою нечітких баз знань з правилами у різних форматах.

1.3 Варіантний огляд проектних рішень

Для вибору оптимального варіанту проектного рішення використаємо інтегральний критерій (1.13). В таблиці 1.4 подані ваги та оцінки критеріїв. Найкращий метод має мати найбільшу інтегральну оцінку.

$$I = \sum_{k=1}^n w_k * m_k , \quad (1.13)$$

де w_k – вага оцінки k ;

m_k - оцінка k ;

n – кількість критеріїв для оцінки.

Обраний метод можна реалізувати за допомогою кількох варіантів проектних рішень, які наведені у таблиці 1.4.

Опис середовищ розробки:

1) C++ — мова програмування високого рівня з підтримкою декількох парадигм програмування: об'єктно-орієнтованої, узагальненої та процедурної. Функції C++ [14]:

- підтримка об'єктно-орієнтованого програмування через класи;
- підтримка узагальненого програмування через шаблони;
- стандартної бібліотеки з вбудованими функціями;
- додаткові типи даних;
- обробка винятків;
- простори імен;
- вбудовані функції;
- перевантаження операторів;
- перевантаження імен функцій;
- посилення і оператори управління вільно розподіленою пам'яттю.

2) Java — об'єктно-орієнтована мова програмування. Синтаксис мови багато в чому схожий на C та C++. У офіційній реалізації, Java програми компілюються у байткод, який при виконанні інтерпретується віртуальною машиною для конкретної платформи [15].

Oracle надає компілятор Java та віртуальну машину Java, які задовольняють специфікації Java Community Process, під ліцензією GNU General Public License.

Передусім, Java розроблялась як платформи-незалежна мова, тому вона має менше низькорівневих можливостей для роботи з апаратним забезпеченням. За

необхідності таких дій java дозволяє викликати підпрограми, написані іншими мовами програмування.

Основні характеристики мови програмування Java:

- синтаксис мови повинен бути «простим, об'єктно-орієнтовним та звичним».
- реалізація має бути «безвідмовною та безпечною», а також «високопродуктивною».
- повинна зберегтися «незалежність від архітектури та портативність».
- мова має бути «динамічною, інтерпретованою та підтримувати мультиопрацювання».

3) MATLAB — пакет прикладних програм для числового аналізу, а також мова програмування, що використовується в даному пакеті. Система є зручним засобом для роботи з математичними матрицями, малювання функцій, роботи з алгоритмами [16].

MATLAB надає користувачеві велику кількість функцій для аналізу даних, які покривають майже всі області математики, зокрема:

- матриці та лінійна алгебра — алгебра матриць, лінійні рівняння, власні значення і вектори, сингулярності, факторизація матриць та інше;
- многочлени та інтерполяція — корені многочленів, операції над многочленами та їх диференціювання, інтерполяція та екстраполяція кривих;
- математична статистика та аналіз даних — статистичні функції, статистична регресія, цифрова фільтрація, швидке перетворення Фур'є та інші;
- обробка даних — набір спеціальних функцій, включаючи побудову графіків, оптимізацію, пошук нулів, чисельне інтегрування та інше;
- диференціальні рівняння — вирішення диференціальних і диференційно-алгебраїчних рівнянь, диференціальних рівнянь із запізнюванням, рівнянь з обмеженнями, рівнянь в часткових похідних та інше;

- розріджені матриці — спеціальний клас даних пакету MATLAB, що використовується у спеціалізованих додатках;
- цілочисельна арифметика — виконання операцій цілочисельної арифметики в середовищі MATLAB.

Опис критеріїв:

- 1) підтримувані операційні системи – критерій показує на скількох класах операційних систем можна запустити розроблений програмний продукт, вага оцінки обрана рівною 0,05 тому що наше програмне забезпечення планується запускати на класі операційних систем сімейства Windows і підтримка інших ОС не суттєва;
- 2) час запуску розробленого програмного забезпечення в системі – критерій є оцінкою часового проміжку між подачею команди на запуск ПЗ і кінцем завантаження його в пам'ять, так як головна мета роботи направлена на вирішення іншої поставленої задачі, то вага даного критерію 0,05;
- 3) зручність написання коду в середовищі розробки – критерій показує на скільки зручними є засоби редагування, перегляду та роботи з програмним кодом в середовищі розробки, критерій прямо впливає на швидкість та зручність розробки програмного забезпечення, тому його вага 0,2;
- 4) присутність вбудованих функцій для навчання нечітких баз знань – критерій вказує на присутність чи відсутність вбудованих функцій для навчання нечітких баз знань, критерій має вагу 0,3, так як він є важливим для досягнення вирішення завдання даної дипломної роботи;
- 5) швидкість написання ПЗ даного типу – критерій показує загальну оцінку часу написання програмного забезпечення пов'язаного з розробкою нечітких баз знань, наприклад наявність редакторів функцій належності, редакторів правил, засобів роботи з матрицями, це найважливіший критерій, тому що від нього залежить якість написаного програмного забезпечення, він має вагу 0,4.

З даних поданих у таблиці 1.4 робимо висновок що найкраще підійде проектне рішення розробки ПЗ для ідентифікації за допомогою нечіткої бази знань з правилами у різних форматах за допомогою Matlab так як воно отримало найбільшу

інтегральну оцінку, написання ПЗ займе найменше часу, тому що в даному пакеті присутні функції для навчання нечітких баз знань.

Таблиця 1.4 – Варіанти проектних рішень

Критерій оцінки	Вага оцінки	Проектне рішення		
		Розробка на C++ [14]	Розробка на Java [15]	Розробка на Matlab [16]
Підтримувані операційні системи	0,05	Windows (оцінка 1)	Багатоплатформенна (оцінка 2)	Багатоплатформенна (оцінка 2)
Час запуску розробленого ПЗ в системі	0,05	Швидкий (оцінка 3)	Середній (оцінка 2)	Середній (оцінка 1)
Зручність написання коду в середовищі розробки	0,2	Не зручно (оцінка 1)	Зручно (оцінка 2)	Зручно (оцінка 2)
Присутність вбудованих функцій для навчання нечітких баз знань	0,3	Не присутні (оцінка 1)	Не присутні (оцінка 1)	Присутні (оцінка 2)
Швидкість написання ПЗ даного типу	0,4	Низька (оцінка 1)	Середня (оцінка 2)	Швидка (оцінка 3)
Величина інтегрального критерію		1,1	1,7	2,35

1.4 Висновки до розділу

У ході проведення технічного обґрунтування було розглянуто область об'єкту управління, аналіз існуючих типів нечітких баз знань.

Розглянуто різні можливі способи виконання ідентифікації залежностей. Обґрунтовано вибір варіанту ідентифікації залежності за допомогою нечіткої бази знань з правилами у різних форматах.

Виконано технічний аналіз основних аспектів, пов'язаних з розробкою: цілі, можливості використання, необхідність створення, можливість розробки, існування відповідних технічних засобів розробки.

Порівняння варіантів методів і проектних рішень показали що доцільно розробити програму для ідентифікації залежностей нечіткою базою знань у різних форматах за допомогою математичного пакету Matlab – обране проектне рішення дозволяє суттєво знизити час на розробку програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

Отже розробка нового програмного продукту є вигідною і обґрунтованою.

2 РОЗРОБКА МАТЕМАТИЧНОЇ МОДЕЛІ НЕЧІТКОЇ БАЗИ ЗНАНЬ З ПРАВИЛАМИ У РІЗНИХ ФОРМАТАХ

В результаті опису між входами та виходами деякої залежності у нечіткій базі знань з правилами у різних форматах може виникнути три випадки формату логічного виведення:

1) випадок коли в результаті логічного виведення утворюється неперервна нечітка множина. Це означає що нечітка база знань містить правила у форматі Мамдані та Ларсена;

2) випадок коли в результаті логічного виведення утворюється чітка множина. Це означає що нечітка база знань містить правила у форматі Сугено та Цукамото та сингтонні правила;

3) випадок коли вихід є сукупністю першого та другого варіанту.

У кожному з трьох випадків виникають певні унікальні проблеми пов'язані з форматом об'єднання різних типів правил в одній базі знань.

2.1 Модель бази знань з правилами Мамдані та Ларсена

Спочатку проводимо фаззифікацію вхідних даних, далі в результаті імплікації за кожним правилом отримуємо нечітку множину на неперервному носію. Після цього проведемо об'єднання отриманих нечітких множин. І останньою процедурою буде дефаззифікація результуючої нечіткої множини.

Операція об'єднання є асоціативною, тому можна спочатку агрегувати результати виведення за правилами одного типу, наприклад, Мамдані, а потім – за правилами іншого типу – Ларсена. Далі об'єднаємо результати агрегування.

Графічна інтерпретація моделі бази знань з правилами Мамдані та Ларсена наведена в Додатку Б.

2.2 Модель бази знань з синглтонними правилами та правилами Сугено і Цукамото

Проводимо фаззіфікацію вхідних даних, далі в результаті імплікації за кожним правилом отримуємо пару чисел – кількісне значення вихідної змінної та ступінь належності. Далі об'єднуємо їх в одну нечітку множину на дискретному носії. Потім проводимо дефаззіфікацію результуючої нечіткої множини.

Операція об'єднання є асоціативною, тому можна, наприклад, спочатку агрегувати результати виведення за правилами Сугено, а потім – за правилами Цукамото, а в кінці – за синглтонними правилами.

Якщо база знань складається з правил Сугено, тоді за нечітких початкових даних $\tilde{X}^* = (\tilde{x}_1^*, \tilde{x}_2^*, \dots, \tilde{x}_n^*)$ змінюється не лише фаззіфікація, але і процедура формування поточних значень консеквентів. У цьому випадку нечіткі значення входів підставляємо в лінійні залежності в консеквентах і розраховуємо поточні значення консеквентів за правилами нечіткої арифметики:

$$\tilde{d}_j = b_{j1} * \tilde{x}_1^* + b_{j2} * \tilde{x}_2^* + \dots + b_{jn} * \tilde{x}_n^* + b_{j0}, j = \overline{1, N}. \quad (2.1)$$

Формула (2.1) означає, що для кожного вхідного вектора $\tilde{X}^*$ продукуються нечіткі значення консеквентів правил $\tilde{d}_j$. Це еквівалентно породженню для кожного вхідного вектора $\tilde{X}^*$ нечіткої бази знань Мамдані з унікальним набором консеквентів, тому подальше логічне виведення здійснимо за (1.7), (1.9), (1.11). В результаті імплікації за кожним правилом отримуємо нечітку множину на неперервному носії. За правилами інших типів (синглтонними та Цукамото) отримуємо нечітку множину на дискретному носії. Далі необхідно провести агрегування нечітких множин на різних носіях – дискретному та неперервному.

Графічна інтерпретація математичної моделі бази знань з синглтонними правилами та правилами Ларсена і Цукамото наведена в Додатку В.

2.3 Випадок зі змішаними правилами

2.3.1 Аналіз проблеми створення бази знань зі змішаними правилами

Розглянемо варіант коли в базі знань є правила у форматі Мамдані, Ларсена спільно з сингтонними правилами, або правилами у форматі Сугено чи Цукамото. За правилами Мамдані та Ларсена на виході отримуємо нечітку множину на неперервному носію (див. пункт 1.2). В зв'язку з тим що обробка відбувається на базі комп'ютера неперервний носій дискретизується у більш ніж ста точках. За правилами Сугено, Цукамото чи за сингтонними правилами на виході отримуємо нечітку множину на дискретному носію, величина якої дорівнює кількості правил. Зазвичай кількість таких правил не перевищує 10, тому і на цих 10 точках буде задана вихідна нечітка множина. Якщо об'єднати вихідні неперервну та дискретну нечіткі множини та провести дефаззифікацію, то її результат майже не залежатиме від правил Сугено, Цукамото чи сингтонних правил. Проблема виникає через те, що в результуючій нечіткій множині кількість елементів, що надійшли з дискретного носія буде на 1-2 порядки менше, ніж тих, що надійшли від неперервного носія.

2.3.2 Вирішення проблеми створення БЗ зі змішаними правилами

Щоб вирішити проблему об'єднання, а саме щоб зрівноважити вплив кожного виду правил у базі знань дефаззифікацію здійснимо у такий спосіб:

- 1) За логічним виведенням за кожним правилом Мамдані чи Ларсена отримуємо нечітку множину, над якою відразу проводимо дефаззифікацію.
- 2) Отримане чітке значення разом зі ступенем виконання правила утворює сингтонну вихідну нечітку множину, яка є аналогом результату логічного виведення за сингтонним правилом. Іншими словами, результати логічного виведення за правилами Мамдані чи Ларсена зводимо до результатів за форматами правил другої групи (Сугено, Цукамото та сингтонної баз знань). Це призводить до того, що змішанні неперервно-дискретні результати логічного виведення перетворюються до дискретних нечітких множин.

3) Далі об'єднуємо отримані множини та виконаємо дефаззифікацію за центром тяжіння (див. розділ 1.2).

Графічна інтерпретація математичної моделі бази знань зі змішаними правилами наведена в Додатку Г.

2.4 Висновки до розділу

В даному розділі проведено розробку трьох математичних моделей баз знань:

- 1) варіант коли в базу знань входять правила типу Мамдані та Ларсена;
- 2) варіант бази знань з сингтонними правилами та правилами Сугено і Цукамото;
- 3) варіант бази знань зі змішаними правилами.

Також подані графічні інтерпретації моделей для кращого розуміння принципів їх функціонування.

В ході аналізу проблеми яка виникає при створені бази знань зі змішаними правилами було представлено метод її вирішення.

3 РОЗРОБКА АЛГОРИТМІЧНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Розглянуті в другому розділі математичні моделі дозволяють перейти до програмної реалізації бази знань з правилами у різних форматах. Техніко-економічне обґрунтування показало що для виконання програмної реалізації використати математичне середовище Matlab.

Розробка ведеться за допомогою структурного підходу до створення алгоритмів. Структурний підхід [35] — це метод розробки алгоритму, який забезпечує легкість читання алгоритму, простоту перевірки правильності виконання алгоритму, зручність у його модифікації. Цей метод передбачає:

- конструювання алгоритму з використанням трьох базових алгоритмічних структур;
- використання методу покрокової деталізації, тобто подрібнення задачі на більш прості задачі, потім подрібнення цих задач на ще більш прості складові й т. д. (розробка алгоритму «зверху вниз»);
- аналіз алгоритму, тобто контроль правильності кожної структури алгоритму і взаємозв'язків структур.

Перевагою структурного підходу є його модульність. Модуль — це послідовність логічно пов'язаних команд, яка оформлена у вигляді окремого алгоритму. Ці допоміжні алгоритми можна конструювати й аналізувати окремо та незалежно, використовуючи їх потім в основному алгоритмі або інших допоміжних алгоритмах.

3.1 Створення бази знань за допомогою вбудованих функцій Matlab

Для створення бази знань в математичному пакеті Matlab можна використати текстові команди які вводяться у командному вікні, але такий підхід є більш затратним з погляду використання часових ресурсів, аніж створення бази знань за допомогою графічного редактору.

До складу програмних засобів Fuzzy Logic Toolbox входить редактор нечіткої системи виведення Fuzzy Inference System Editor (FIS Editor або FIS-редактор) разом з допоміжними програмами-редактором функцій приналежності (Membership Function Editor), редактор правил (Rule Editor), переглядач правил (Rule Viewer) і переглядач поверхні відклику (Surface Viewer).

Щоб почати роботу з графічним редактором потрібно ввести команду fuzzy в командному вікні математичного пакету Matlab. Після цього з'явиться головне вікно FIS Editor (рисунок 3.1). Схема FIS структури наведена на рисунку 3.2 [16].

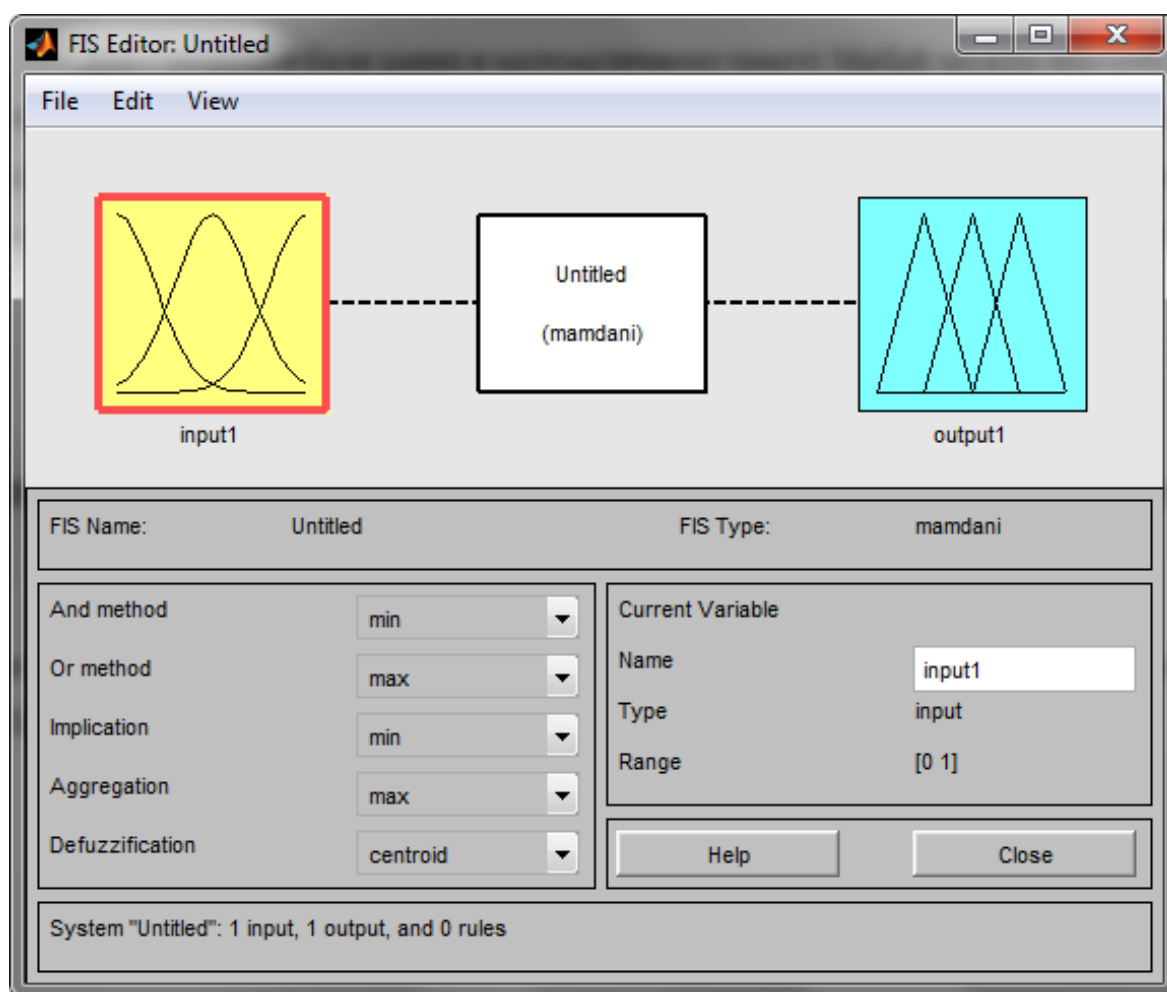


Рисунок 3.1 - Головне вікно FIS Editor

Головне меню редактора містить позиції:

- file - робота з файлами моделей (їх створення, збереження, зчитування і друк);

- edit - операції редагування (додавання і виключення вхідних і вихідних змінних);
- view - перехід до додаткового інструментарію.

Поля налаштувань призначені для збереження наступної інформації:

- andMethod - реалізація логічної операції «І» (запрограмовані реалізації: «min» – мінімум і «prod» – множення);
- orMethod - реалізація логічної операції «АБО» (запрограмовані реалізації: «max» – максимум і «probor» – імовірнісне "АБО");
- defuzzMethod - метод дефазифікації (запрограмовані методи для систем типу Мамдані: «centroid» – центр ваги; «bisector» – медіана; «lom» – найбільший з максимумів; «som» – найменший з максимумів; «mom» – середнє з максимумів; запрограмовані методи для систем типу Сугено: «wtaver» – зважене середнє і «wtsum» – зважена сума);
- impMethod - реалізація операції імплікації (запрограмовані реалізації: «min» – мінімум і «prod» – множення);
- aggMethod - реалізація операції об'єднання функцій приналежності вихідної змінної (запрограмовані реалізації: «max» – максимум; «sum» – сума і «probor» – імовірнісне «АБО»);

Типи функцій належності запрограмовані в Matlab:

- dsigmf - функція належності у виді різниці між двома сигмоїдними функціями;
- gauss2mf - двостороння гаусівська функція належності;
- gaussmf - гаусівська функція належності;
- gbellmf - узагальнена дзвоноподібна функція належності;
- pimf - пі-подібна функція належності;
- psigmf - добуток двох сигмоїдних функцій належності;
- sigmf - сигмоїдна функція належності;
- smf - s-подібна функція належності;
- trapmf - трапецієподібна функція належності;
- trimf - трикутна функція належності;

- zmf - z-подібна функція належності
- constatnt – константа (функція приналежності у виді синглтона);
- linear – лінійна комбінація вхідних змінних.

Таблиця 3.1 – Налаштування FIS Editor для створення баз знань різних типів

Тип правил нечіткої бази знань	Опції редактора FIS						Підтримувані типи функцій належності	
	Тип FIS (FIS type)	Метод «І» (And method)	Метод «АБО» (Or method)	Імплікація (Implication)	Агрегація (Aggregation)	Дефаззифікація (Defuzzification)	Вхідних змінних	Вихідних змінних
Мамдані	mamdani	min	max	min	max	centroid	dsigmf gauss2mf gaussmf gbellmf pimf psigmf sigmf smf trimf trapmf zmf	dsigmf gauss2mf gaussmf gbellmf pimf psigmf sigmf smf trimf trapmf zmf
Ларсена	mamdani	min	max	prod	max	centroid		
Цукамото	mamdani	min	max	custom (користувацька)	max	centroid		sigmf smf zmf
Сугено	sugeno	prod	probor	-	-	wtaver		constatnt linear
Синглтонна	sugeno	prod	probor	-	-	wtaver		

За допомогою інструментів FIS Editor за замовчуванням можна створити базу даних типу Сугено або базу даних типу Мамдані. Для створення баз даних типу

Цукамото, Ларсена та синглтонної потрібно змінити деякі настройки в редакторі (див. таблицю 3.1). Для більш докладного ознайомлення з можливостями програмних засобів Fuzzy Logic Toolbox див. документацію [16] і пункт 3.6.

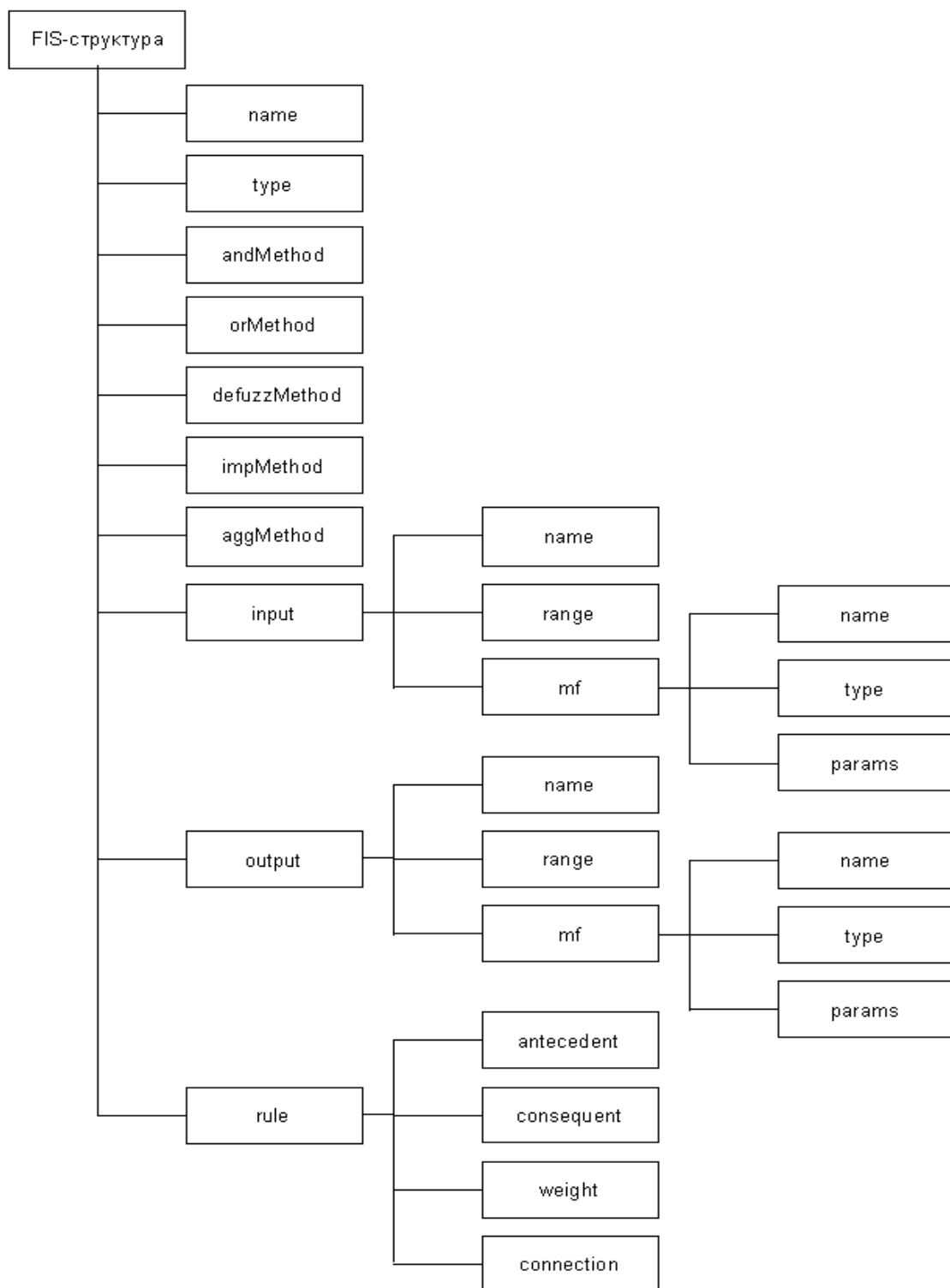


Рисунок 3.2 – Схема FIS-структури

3.2 Розробка програмної реалізації бази знань з правилами в форматі Ларсена та Мамдані

Під час програмної реалізації бази знань з правилами в форматі Ларсена та Мамдані потрібно виконати наступні кроки:

- 1) розбиваємо нечітку гібридну базу знань на базу знань з правилами в форматі Ларсена та на базу знань з правилами в форматі Мамдані;
- 2) формуємо вектор вхідних даних;
- 3) здійснюємо логічне виведення без дефаззифікації з кожної із баз знань за одного і того ж значення вхідного вектору;
- 4) отримані нечіткі множини об'єднуємо і проводимо дефаззифікацію;
- 5) результатом дефаззифікації є шукана залежність.

Опис користувацьких функцій наведений в розділі 3.5.

Блок схема алгоритму наведена в Додатку Д.

Лістинг програми наведений в Додатку Г.

3.3 Розробка програмної реалізації бази знань з сингלטонними правилами та правилами у форматі Сугено

Під час програмної реалізації розбиваємо нечітку гібридну базу знань на базу знань з сингלטонними правилами та правилами в форматі Сугено. Здійснюємо логічне виведення без дефаззифікації з кожної із баз знань за одного і того ж значення вхідного вектору. Отримані нечіткі множини об'єднуємо і проводимо дефаззифікацію. Результатом дефаззифікації є шукана залежність.

Блок схема алгоритму наведена в Додатку Ж.

Лістинг програми наведений в Додатку Г.

3.4 Розробка програмної реалізації бази знань з сингלטонними правилами та правилами у форматі Сугено, Ларсена, Мамдані

У другому розділі була розглянута проблема поєднання в одній базі знань правил які мають різні консеквенти. В такому випадку не можна об'єднувати логічні виводи за гомогенними правилами, а потім їх дефаззифікувати, тому що ми отримаємо вивід який не буде залежати від сингלטонних правил, правил Сугено чи Цукамото.

Тому для розробки програмної реалізації було використано наступний алгоритм:

- 1) за логічним виводом за кожним з правил Мамдані чи Ларсена тримаємо нечітку множину, над якою проводимо дефаззифікацію;
- 2) отримане чітке значення разом із ступенем виконання утворює сингלטонну нечітку множину;
- 3) отримана сингלטонна множина є аналогом до результатів отриманих за логічним виводом по базі знань з правилами типу Сугено, Цукамото та сингלטонних;
- 4) відповідно до дій які виконані у пунктах 1 -3 ми переходимо до дискретних рівно важливих нечітких множин;
- 5) проводимо об'єднання отриманих множин;
- 6) виконаємо дефаззифікацію результатів за центром тяжіння.

Блок схема алгоритму наведена в Додатку К.

Лістинг програми наведений в Додатку Г.

3.5 Опис основних функцій використаних при розробці програмного забезпечення

3.5.1 Функція Evalfis

Синтаксис функції Evalfis [13, 16]:

- 1) `output = evalfis(input, fis);`
- 2) `output = evalfis(input, fis, numPts);`

- 3) [output, IRR, ORR, ARR] = evalfis(input, fis);
- 4) [output, IRR, ORR, ARR] = evalfis(input, fis, numPts).

Опис функції: Виконує нечіткий логічний вивід.

Функція evalfis може приймати три вхідні аргумента, два з яких обов'язкові:

1) input – матриця значень вхідних змінних для яких потрібно виконати нечіткий логічний вивід. Матриця має мати розмір M на N, де N – кількість вхідних змінних; M – кількість вхідних даних. Кожна стрічка матриці представляє собою один вектор вхідних даних;

2) fis – ідентифікатор системи нечіткого логічного виводу;

3) numPts – кількість точок дискретизації функцій приналежності, необов'язковий аргумент.

Функція evalfis може мати чотири вихідних аргументи:

1) output - матриця значень вихідних змінних, що отримується в результаті нечіткого логічного виводу для вектора вхідних значень input. Матриця має розмір M на L, де M - кількість вхідних даних; L - кількість вихідних змінних в fis;

2) IRR – матриця розміром NR на N, де NR - кількість правил в fis; N - кількість вхідних змінних. Матриця містить ступені належності вхідних значень термам, що входять до бази знань;

3) ORR – матриця розміром numPts на (NR * L), де numPts - кількість точок дискретизації; NR - кількість правил в fis; L - кількість вихідних змінних в fis. Кожен стовпець матриці містить функцію належності вихідної змінної, одержувану в результаті виведення по одному правилу. Функція приналежності дискретизується на numPts точках і представляється у вигляді множини ступенів належності;

4) ARR – матриця розміром numPts на L, де numPts - кількість точок дискретизації; L - кількість вихідних змінних в fis. Матриця містить функції приналежності вихідних змінних, одержувані в результаті нечіткого логічного висновку по всій базі знань. Функції належності дискретизуються на numPts точках і представляються у вигляді множини ступенів належності.

Аргументи IRR, ORR і ARR є необов'язковими, вони містять проміжні результати нечіткого логічного виводу. У разі задання декількох вхідних даних

значення аргументів IRR, ORR і ARR будуть розраховані тільки для останнього вектора вхідних даних. Ці аргументи використовуються коли необхідно відстежити процес логічного висновку або коли необхідно реалізувати нестандартну процедуру нечіткого виводу.

3.5.2 Функція Defuzz

Синтаксис функції Defuzz [13, 16]:

1) `crisp = defuzz (x, mf, method).`

Опис функції: Виконує операцію дефаззифікації, тобто перетворення нечіткої множини в чітке число.

Функція defuzz приймає три вхідні аргументи:

- 1) `x` – універсальна множина, на якій задано нечітку множину, що підлягає дефаззифікації;
- 2) `mf` – вектор ступенів належності елементів множини `x` нечіткій множині, що підлягає дефаззифікації;
- 3) `method` – метод дефаззифікації. Допустимі значення:
 - `'centroid'` – центр ваг;
 - `'bisector'` – медіана;
 - `'mom'` – центр максимумів;
 - `'som'` – найменший із максимумів;
 - `'lom'` – найбільший із максимумів.

3.5.3 FIS-редактор

FIS-редактор [13, 16] призначений для створення, збереження, завантаження і виведення на друк систем нечіткого логічного виводу, а також для редагування наступних властивостей:

- тип системи;
- найменування системи;
- кількість вхідних і вихідних змінних;
- найменування вхідних і вихідних змінних;

- параметри нечіткого логічного висновку.

3.5.4 Функція eval_mam_lars

Синтаксис функції eval_mam_lars:

- 1) `outputval = eval_mam_lars(inputval, mamdanifis, larsenfis, numberPoints).`

Опис функції: виконує операцію логічного виводу по нечіткій базі знань з правилами типу Мамдані та Ларсена.

Функція eval_mam_lars приймає чотири вхідні аргументи:

- 1) `inputval` – матриця значень вхідних змінних для яких потрібно виконати нечіткий логічний вивід. Матриця має мати розмір M на N , де N – кількість вхідних змінних; M – кількість вхідних даних. Кожна стрічка матриці представляє собою один вектор вхідних даних;
- 2) `mamdanifis` – ідентифікатор системи нечіткого логічного виводу типу Мамдані;
- 3) `larsenfis` - ідентифікатор системи нечіткого логічного виводу типу Ларсена;
- 4) `numberPoints` – кількість точок дискретизації функцій приналежності.

Функція eval_mam_lars має один вихідний аргумент:

- 1) `outputval` - матриця значень вихідних змінних, що отримується в результаті нечіткого логічного виводу для вектора вхідних значень `inputval`. Матриця має розмір M на L , де M - кількість вхідних даних; L - кількість вихідних змінних в нечіткій базі знань.

Лістинг функції наведений в Додатку Г.

3.5.5 Функція eval_mam_sug

Синтаксис функції eval_mam_sug:

- 1) `outputval = eval_mam_sug(inputval, mamdanifis, sugenofis, numberPoints).`

Опис функції: виконує операцію логічного виводу по нечіткій базі знань з правилами типу Мамдані та Сугено.

Функція eval_mam_sug приймає чотири вхідні аргументи:

1) `inputval` – матриця значень вхідних змінних для яких потрібно виконати нечіткий логічний вивід. Матриця має мати розмір M на N , де N – кількість вхідних змінних; M – кількість вхідних даних. Кожна стрічка матриці представляє собою один вектор вхідних даних;

2) `mamdanifis` – ідентифікатор системи нечіткого логічного виводу типу Мамдані;

3) `sugenofis` - ідентифікатор системи нечіткого логічного виводу типу Сугено;

4) `numberPoints` – кількість точок дискретизації функцій приналежності.

Функція `eval_mam_sug` має один вихідний аргумент:

1) `outputval` - матриця значень вихідних змінних, що отримується в результаті нечіткого логічного виводу для вектора вхідних значень `inputval`. Матриця має розмір M на L , де M - кількість вхідних даних; L - кількість вихідних змінних в нечіткій базі знань.

Лістинг функції наведений в Додатку Г.

3.5.6 Функція `eval_singl_sug`

Синтаксис функції `eval_singl_sug`:

1) `outputval = eval_singl_sug(inputval, singletonfis, sugenofis, numberPoints)`.

Опис функції: виконує операцію логічного виводу по нечіткій базі знань з правилами типу синглтон та Сугено.

Функція `eval_singl_sug` приймає чотири вхідні аргументи:

1) `inputval` – матриця значень вхідних змінних для яких потрібно виконати нечіткий логічний вивід. Матриця має мати розмір M на N , де N – кількість вхідних змінних; M – кількість вхідних даних. Кожна стрічка матриці представляє собою один вектор вхідних даних;

2) `singletonfis` – ідентифікатор системи нечіткого логічного виводу типу синглтон;

3) `sugenofis` - ідентифікатор системи нечіткого логічного виводу типу Сугено;

4) `numberPoints` – кількість точок дискретизації функцій приналежності.

Функція `eval_mam_sug` має один вихідний аргумент:

1) `outputval` - матриця значень вихідних змінних, що отримується в результаті нечіткого логічного виводу для вектора вхідних значень `inputval`. Матриця має розмір M на L , де M - кількість вхідних даних; L - кількість вихідних змінних в нечіткій базі знань.

Лістинг функції наведений в Додатку Г.

3.6 Висновки до розділу

В даному розділі було обрано метод розробки програмного забезпечення, а саме структурний підхід, тому що за допомогою нього легко перевіряти окремі частини програмного забезпечення, зручно проводити модифікації.

Також розглянуті основні засоби та функції математичного пакету Matlab для реалізації логічного виводу нечіткої бази знань. Розглянуті настройки редактора нечітких баз знань.

Розроблено і описано пакет функцій і програм для ідентифікації залежностей за допомогою нечіткої бази знань в правилах у різних форматах.

4 ТЕСТУВАННЯ ТА ОЦІНКА ЕФЕКТИВНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для тестування та оцінки ефективності програмного забезпечення ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах виконаємо задачу прогнозування ефективності витрати палива. Прогноз будемо проводити по двом критеріям – маса та рік випуску.

Вибірка Auto MPG знаходиться на UC Irvine Machine Learning Repository [36]. Вибірка була поділена на навчальну та тестову та перевірена на репрезентативність за допомогою розробленої програми лістинг якої наведено в додатку Г.

Оцінка точності проведена за допомогою середньоквадратичної похибки (RMSE) [37].

Оптимізація параметрів проводилася за допомогою вбудованих функцій з Optimization Toolbox математичного пакету Matlab.

4.1 Тестування та оцінка ефективності ідентифікації залежностей за допомогою бази знань з правилами типу Мамдані

Для того щоб оцінити результати роботи розробленого в даній дипломній роботі методу, порівнюємо ефективність ідентифікації залежностей за допомогою бази знань з правилами у різних форматах та ефективність ідентифікації залежностей за допомогою бази знань з правилами типу Мамдані.

На рисунках 4.1 – 4.6 зображена створена база знань Мамдані: вхідні та вихідні змінні, діапазон в якому вони лежать, їхні функції належності, а також створені правила та залежність входи-вихід.

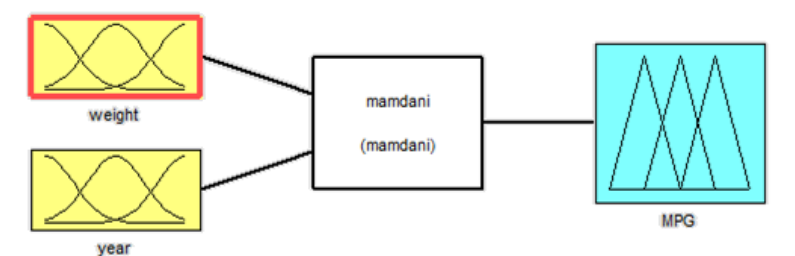


Рисунок 4.1 – Входи та вихід бази знань з правилами типу Мамдані

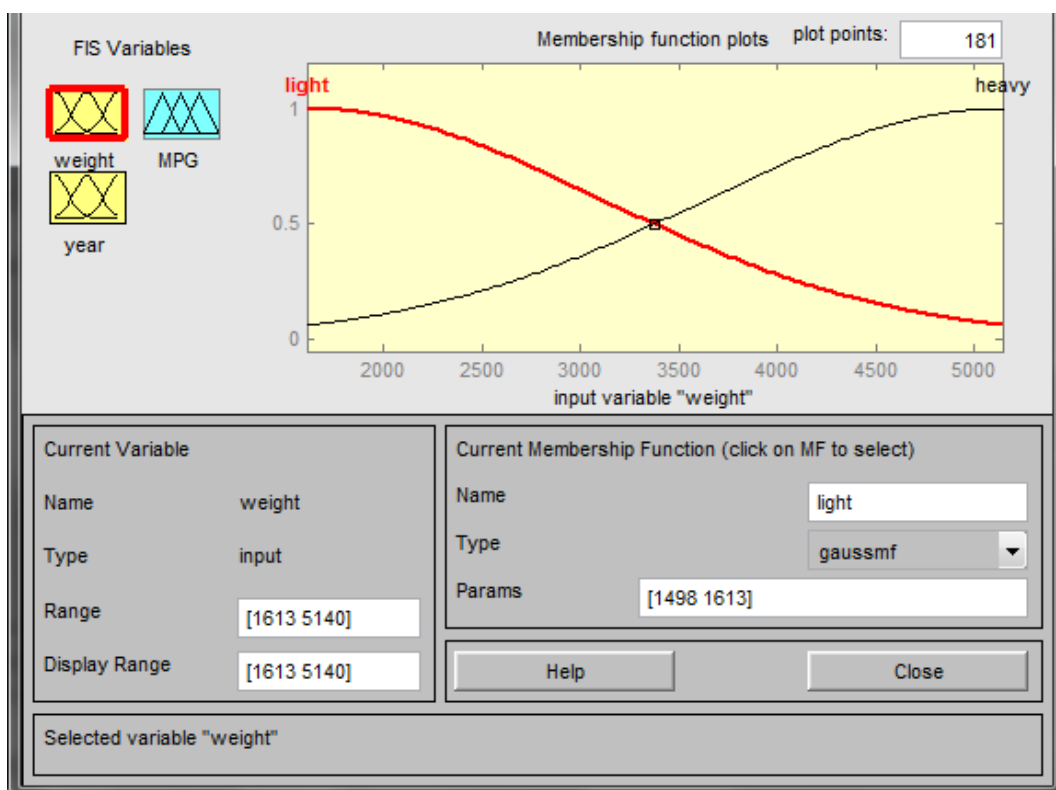


Рисунок 4.2 – Функції належності вхідної змінної weight

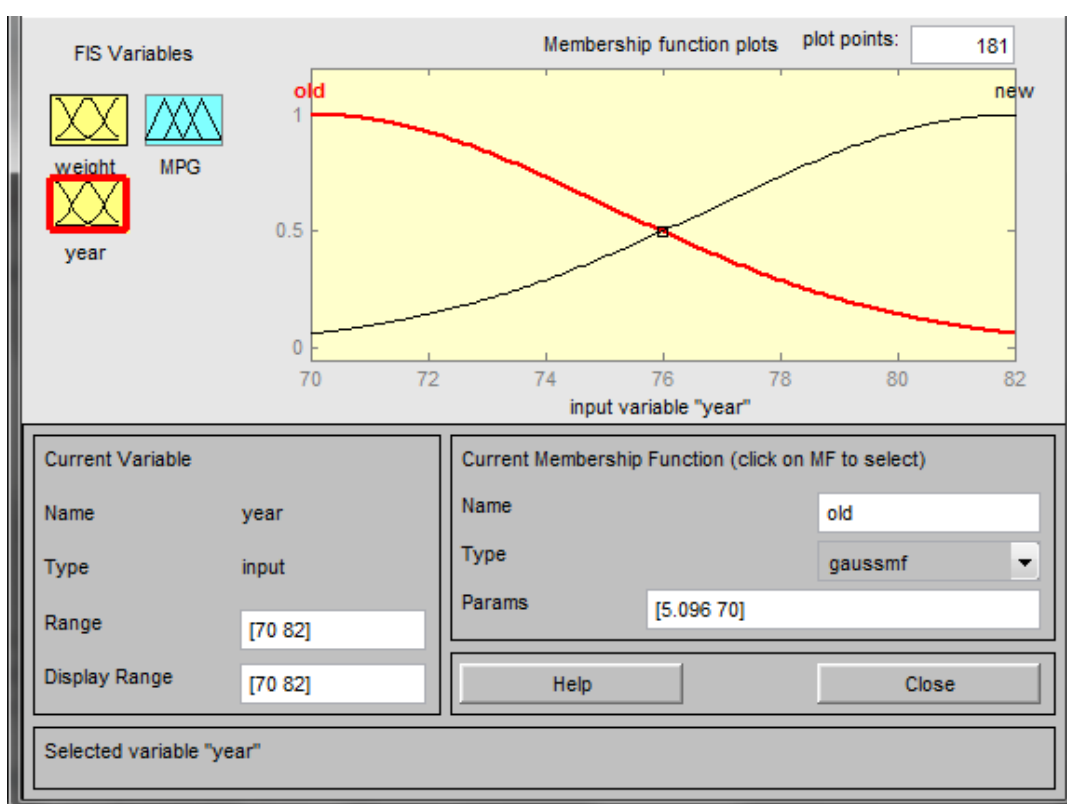


Рисунок 4.3 – Функції належності вхідної змінної year

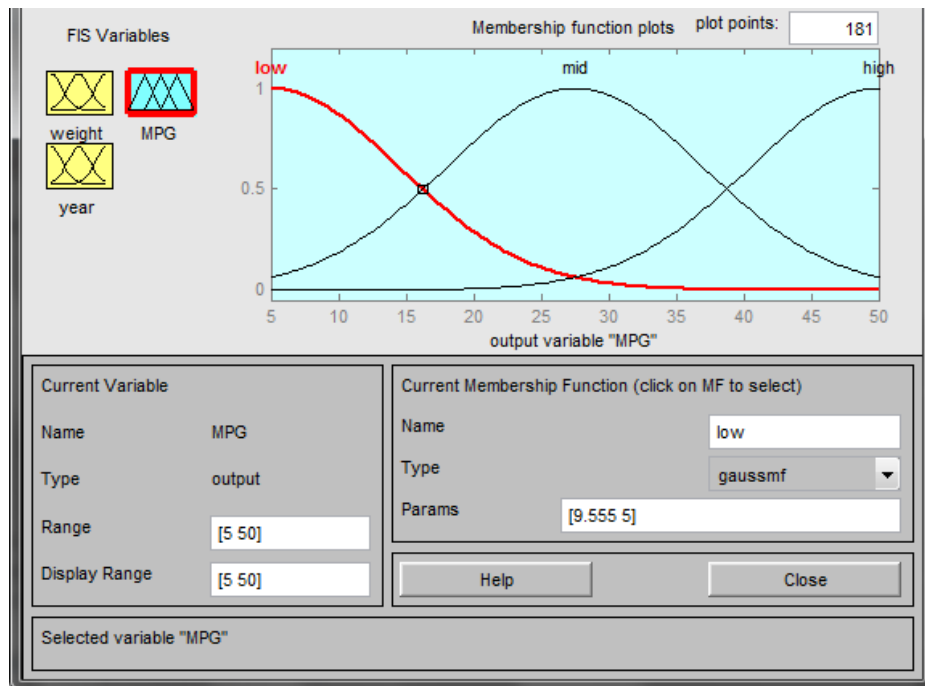


Рисунок 4.4 – Функції належності вихідної змінної MPG

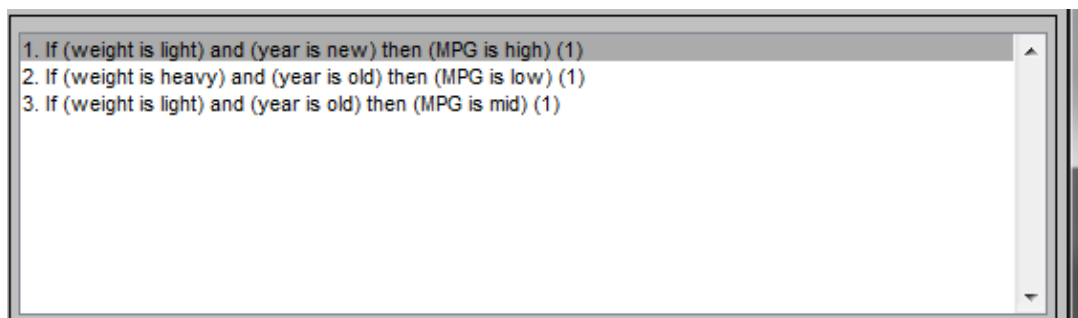


Рисунок 4.5 – Правила нечіткої бази знань типу Мамдані

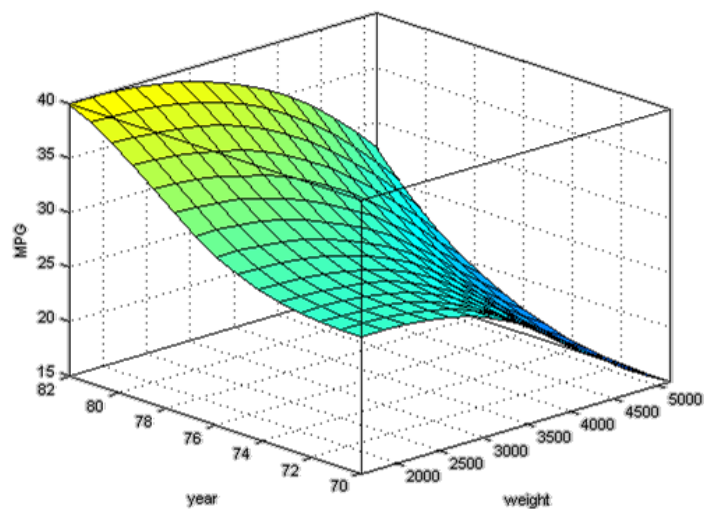


Рисунок 4.6 – Нечітка модель залежності MPG від year та weight

Проведемо настройку одинадцяти параметрів нечіткої бази знань: сім коефіцієнтів центрації функцій належності вхідних та вихідних змінних, координату максимуму функції приналежності вихідного терма *mid*, та три вагові коефіцієнта створених правил.

Лістинг програми настройки та тестування бази знань Мамдані наведений в Додатку Г.

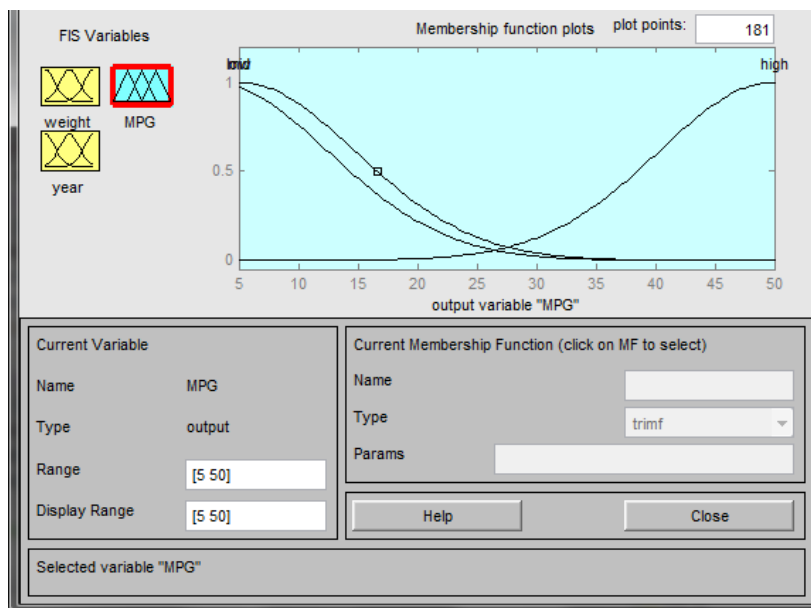


Рисунок 4.7 - Функції належності вихідної змінної MPG після навчання нечіткої бази знань Мамдані

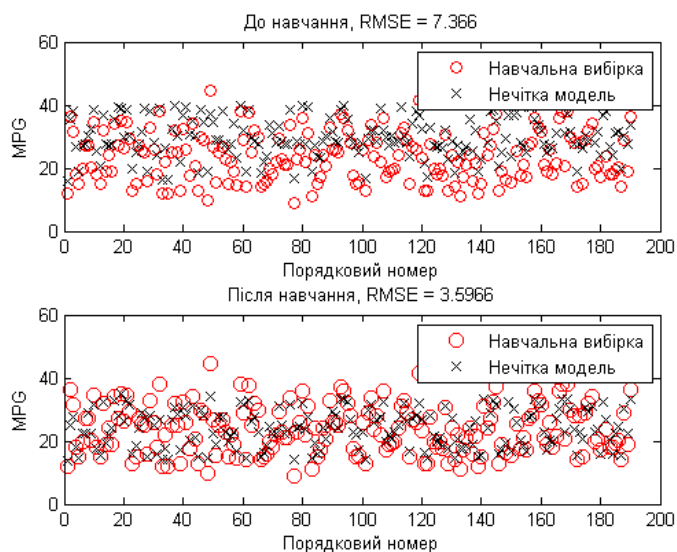


Рисунок 4.8 – Графік розкиду виходу нечіткої моделі до та після навчання

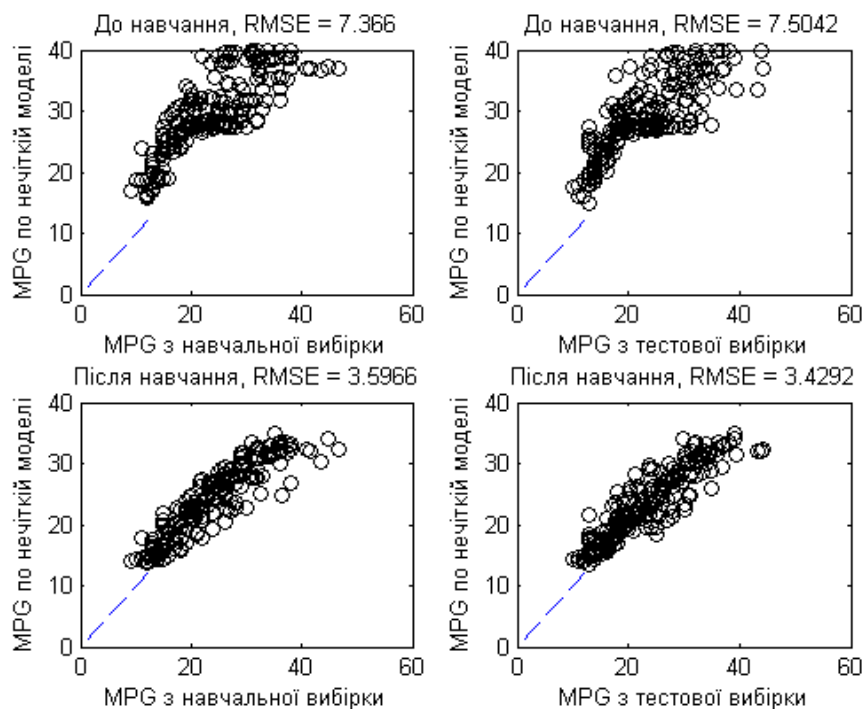


Рисунок 4.9 – Графік залежності виходу по нечіткій базі знань Мамдані до і після навчання та виходу тестової вибірки

Результати ідентифікації за допомогою бази знань Мамдані наведені в таблиці 4.1.

4.2 Тестування та оцінка ефективності ідентифікації залежностей за допомогою нечіткої бази знань з правилами типу Мамдані та Сугено

Для тестування нечітка база знань з правилами типу Мамдані та Сугено була розділена на дві бази знань з гомогенними правилами.

На рисунках 4.10 – 4.16 зображена створена база знань Мамдані: вхідні та вихідні змінні, діапазон в якому вони лежать, їхні функції належності, а також створені правила та залежність входи-вихід.

На рисунках 4.17 – 4.21 зображена створена база знань Сугено: вхідні та вихідні змінні, діапазон в якому вони лежать, їхні функції належності, а також створені правила та залежність входи-вихід.

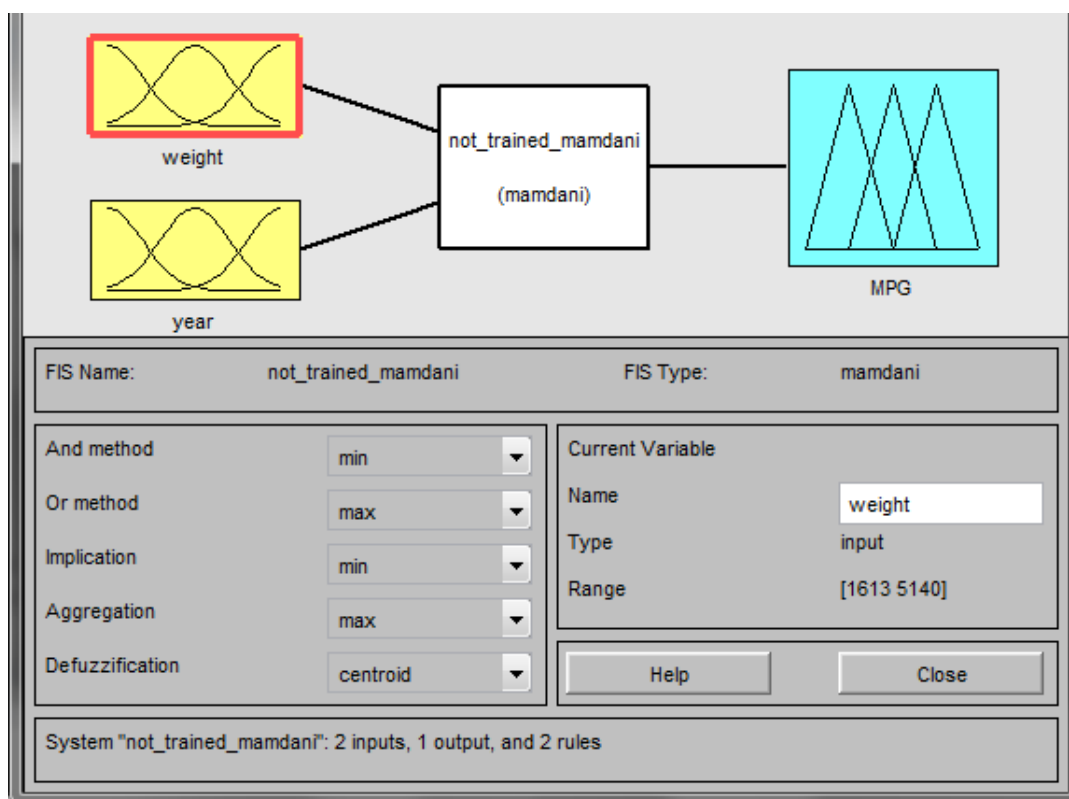


Рисунок 4.10 – Входи та вихід бази знань з правилами типу Мамдані

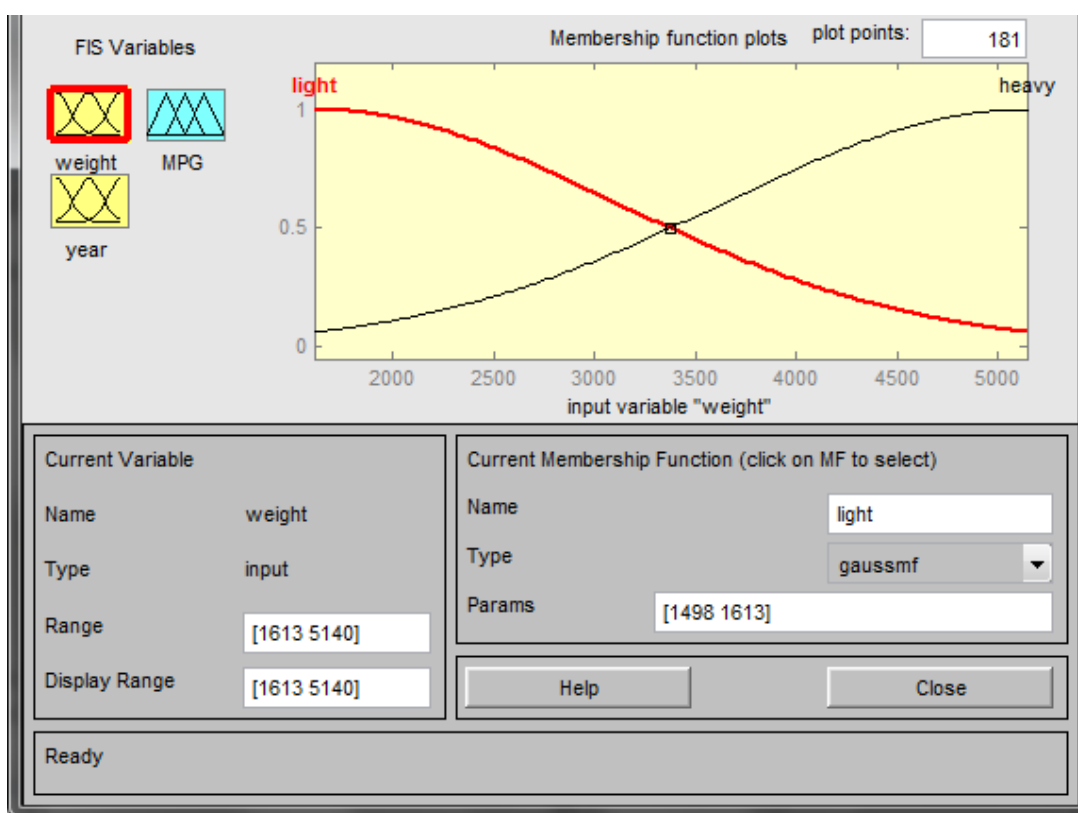


Рисунок 4.11 – Функції належності вхідної змінної weight в базі знань з правилами типу Мамдані

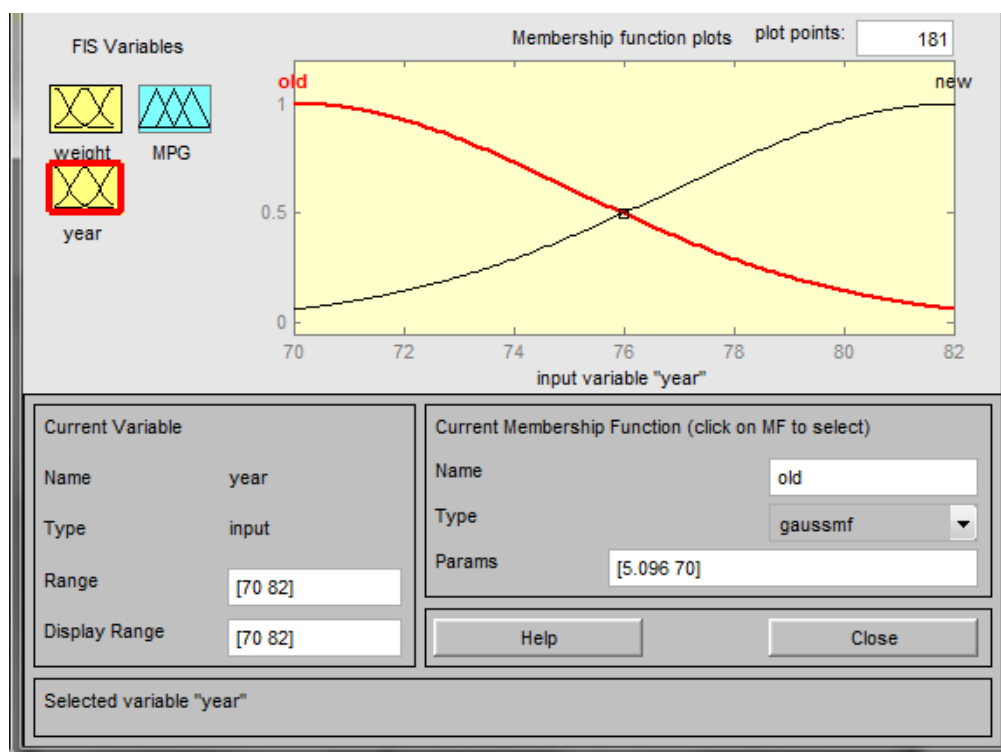


Рисунок 4.12 – Функції належності вхідної змінної year в базі знань з правилами типу Мамдані

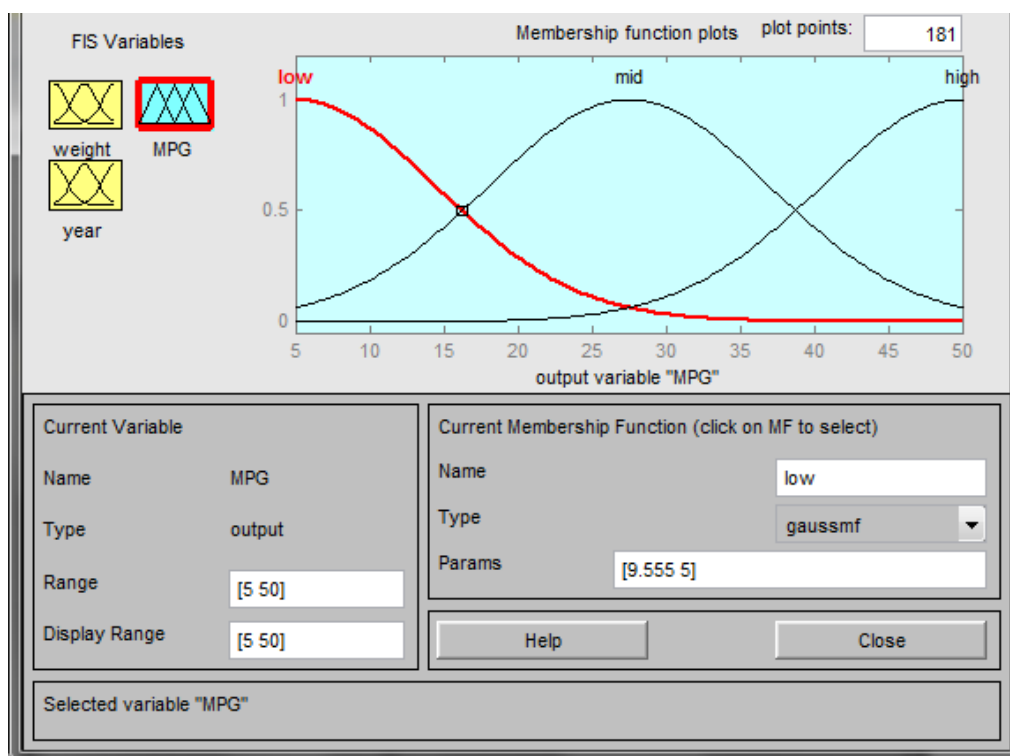


Рисунок 4.13 – Функції належності вихідної змінної MPG в базі знань з правилами типу Мамдані

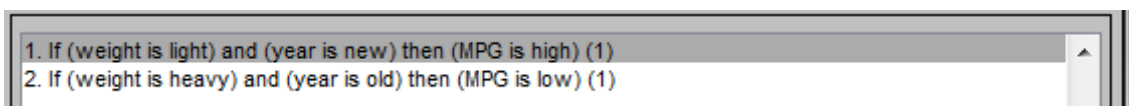


Рисунок 4.14 – Правила нечіткої бази знань типу Мамдані

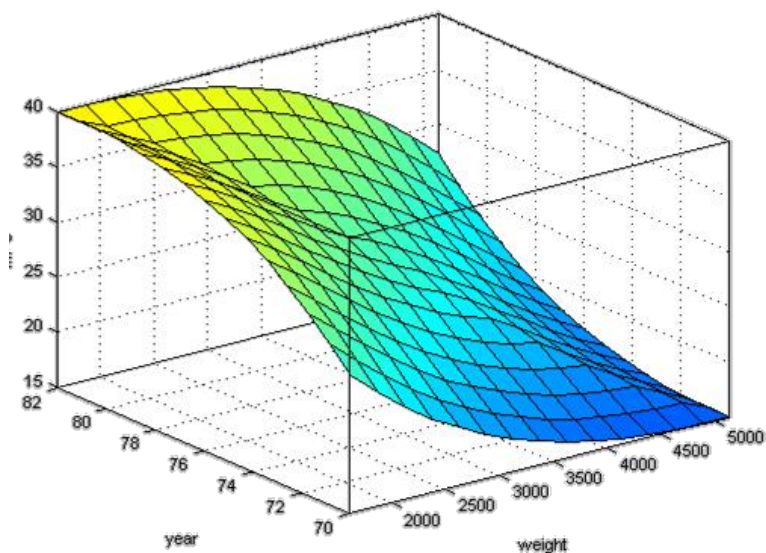


Рисунок 4.15 – Нечітка модель залежності MPG від year та weight в базі знань з правилами типу Мамдані

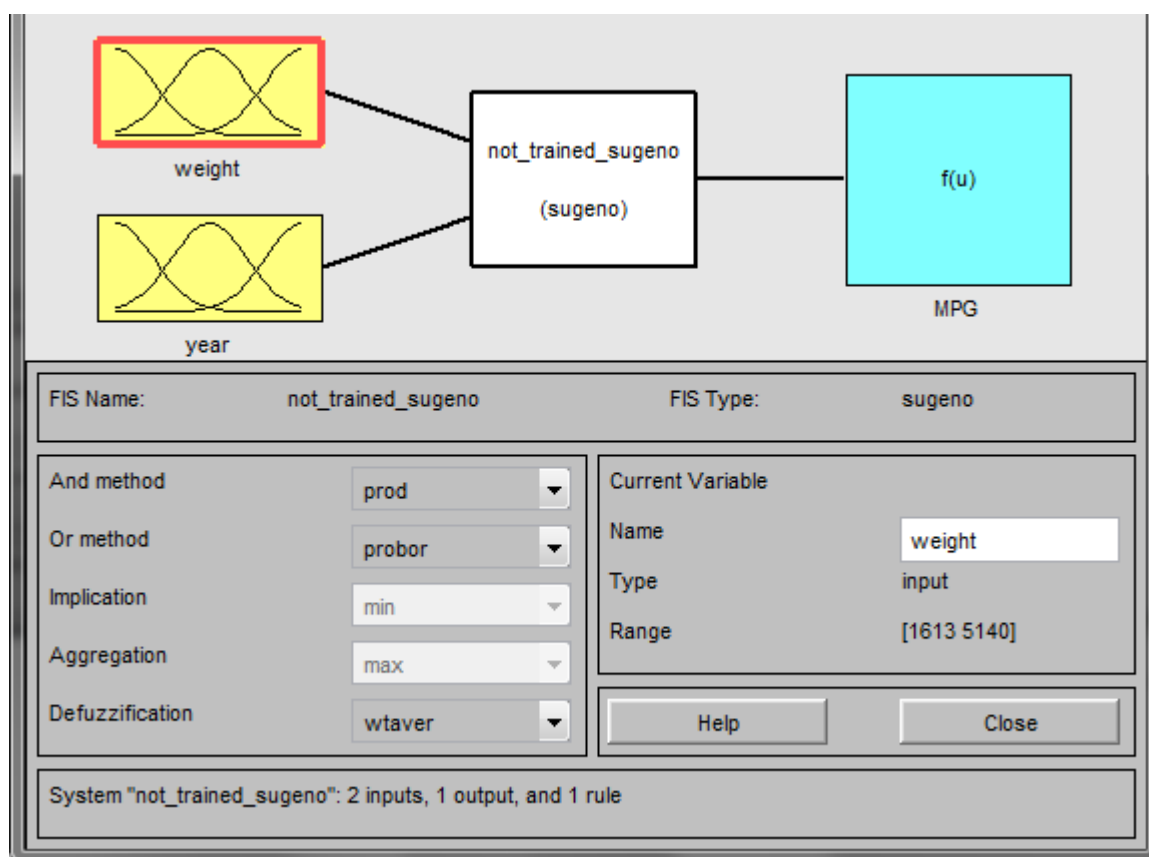


Рисунок 4.16 – Входи та вихід бази знань з правилами типу Сугено

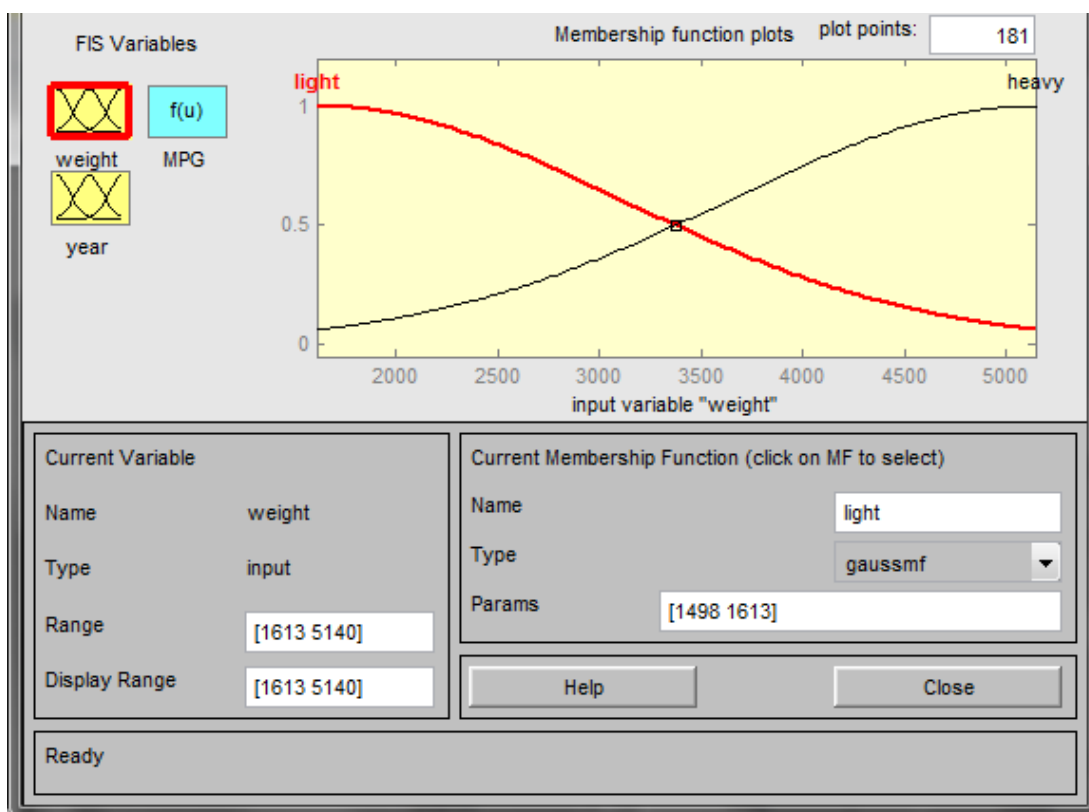


Рисунок 4.17 – Функції належності вхідної змінної weight в базі знань з правилами типу Сугено

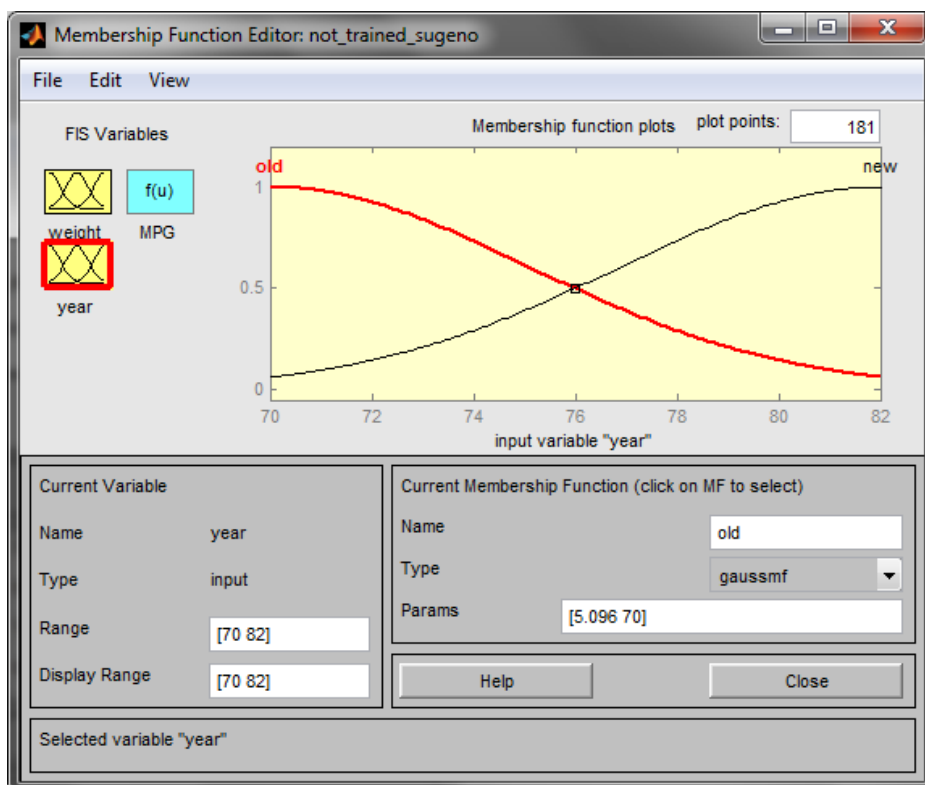


Рисунок 4.18 – Функції належності вхідної змінної year в базі знань з правилами типу Сугено

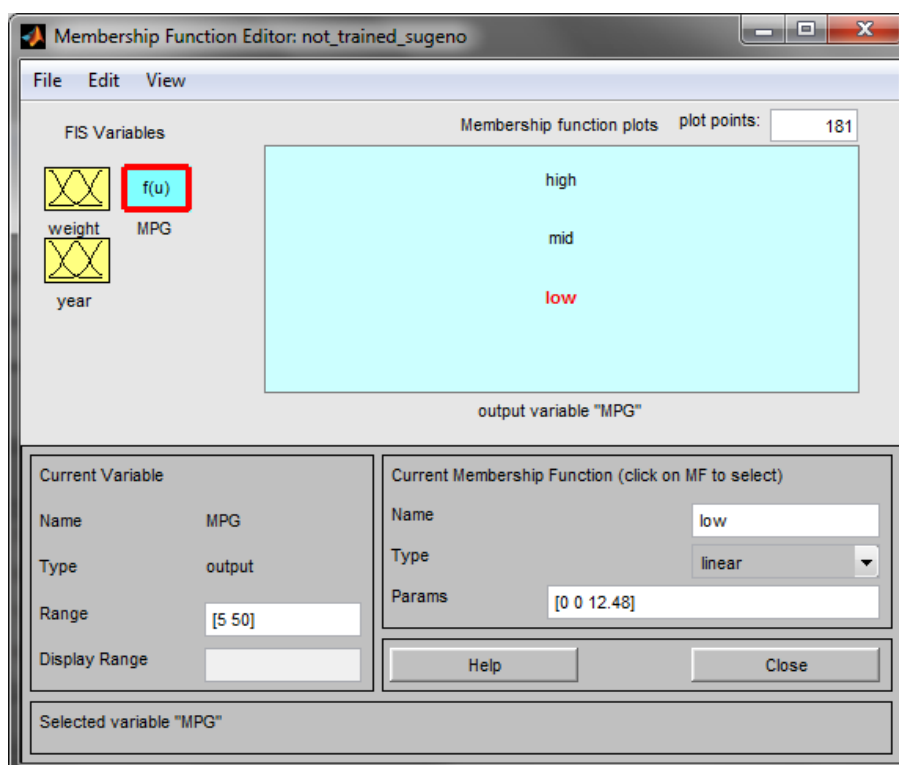


Рисунок 4.19 – Функції належності вихідної змінної MPG в базі знань з правилами типу Сугено

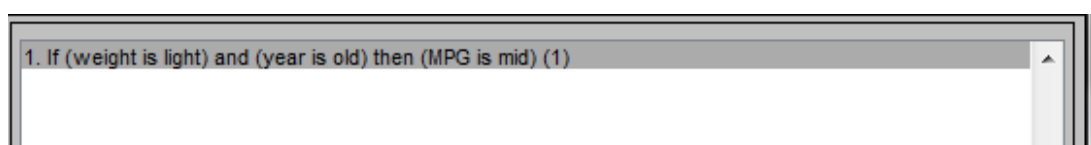


Рисунок 4.20 – Правила нечіткої бази знань типу Сугено до навчання

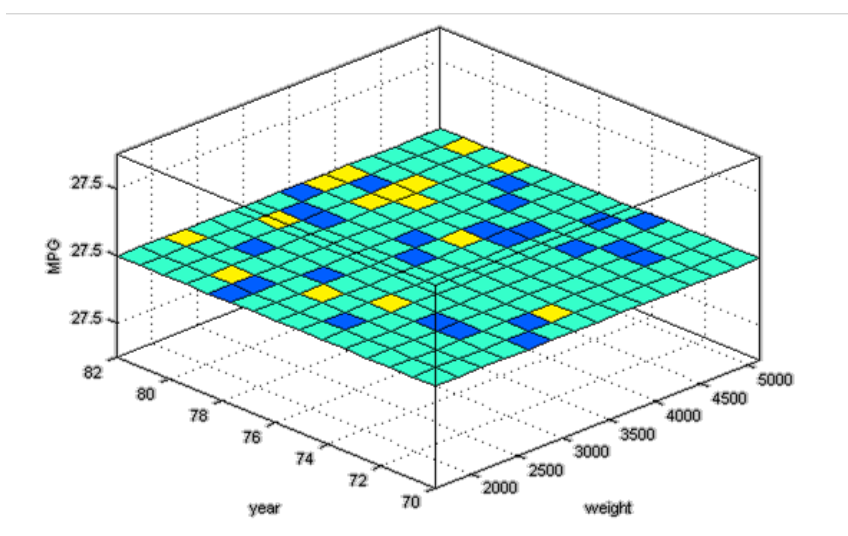


Рисунок 4.21 – Нечітка модель залежності MPG від year та weight в базі знань з правилами типу Сугено

Проведемо настройку десяти параметрів нечіткої бази знань Мамдані: сім коефіцієнтів центрації функцій належності вхідних та вихідних змінних, координату максимуму функції приналежності вихідного терма *mid*, та два вагові коефіцієнта створених правил.

Проведемо настройку дев'яти параметрів нечіткої бази знань Сугено: сім коефіцієнтів центрації функцій належності вхідних та вихідних змінних, координату максимуму функції приналежності вихідного терма *mid*, та один ваговий коефіцієнта створеного правила.

Лістинг програми настройки нечіткої бази знань з правилами Мамдані та Сугено наведені в Додатку Г.

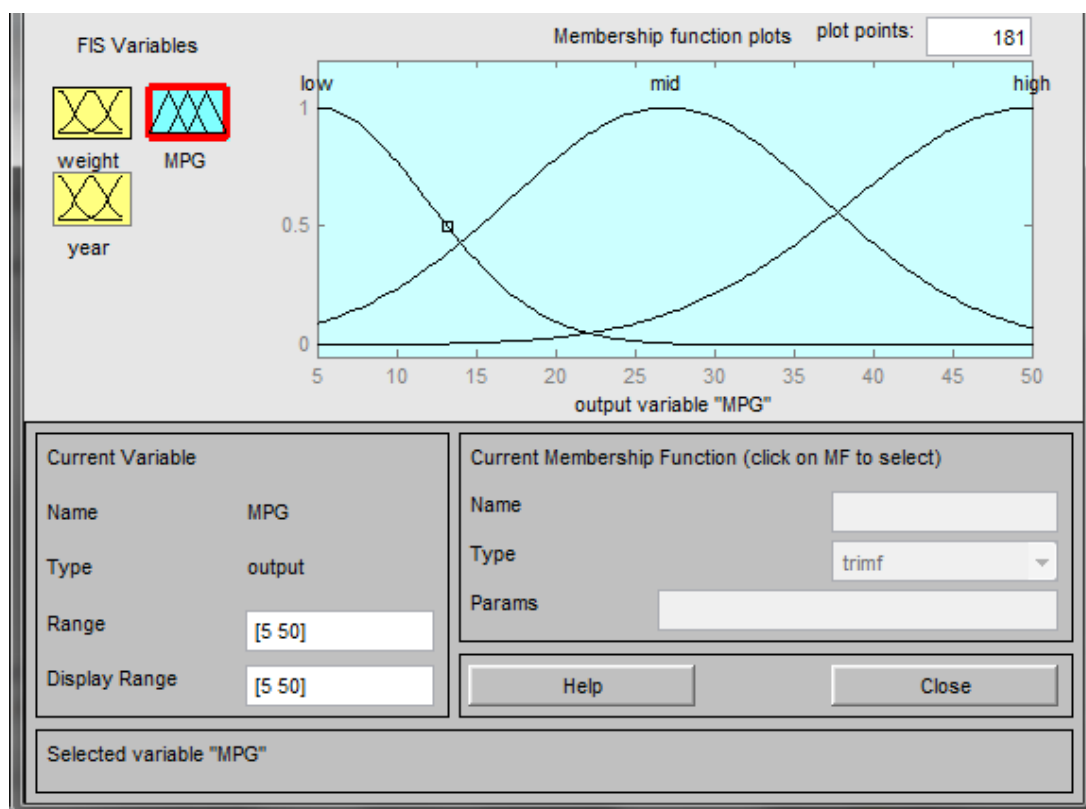


Рисунок 4.22 - Функції належності вихідної змінної MPG після навчання нечіткої бази знань Мамдані

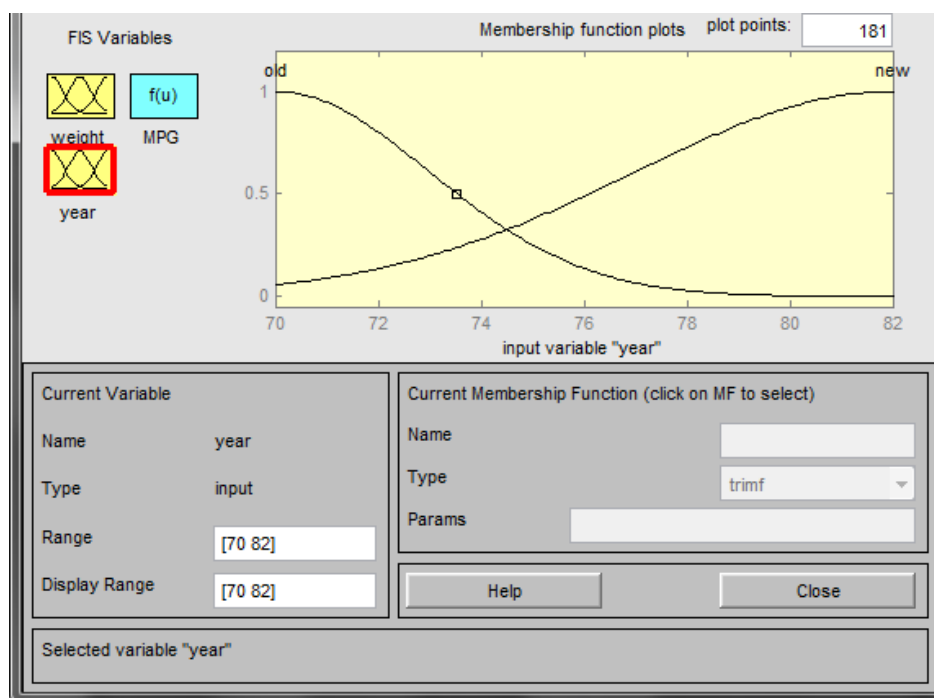


Рисунок 4.23 - Функції належності вхідної змінної year після навчання нечіткої бази знань Сугено

Результати ідентифікації за допомогою бази знань з правилами типу Мамдані та Сугено наведені в таблиці 4.1.

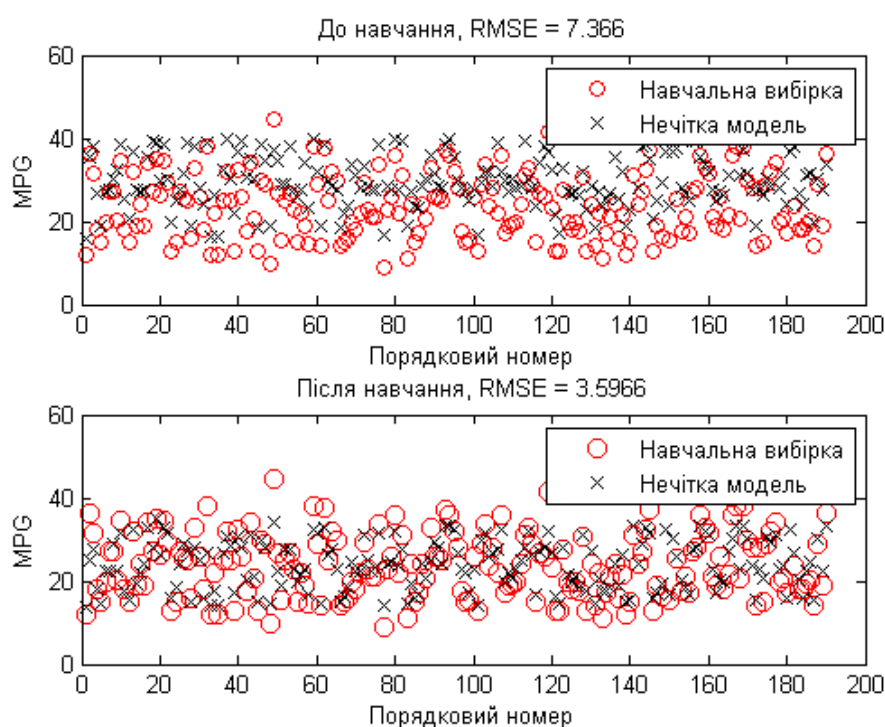


Рисунок 4.24 – Графік розкиду виходу нечіткої моделі до та після навчання

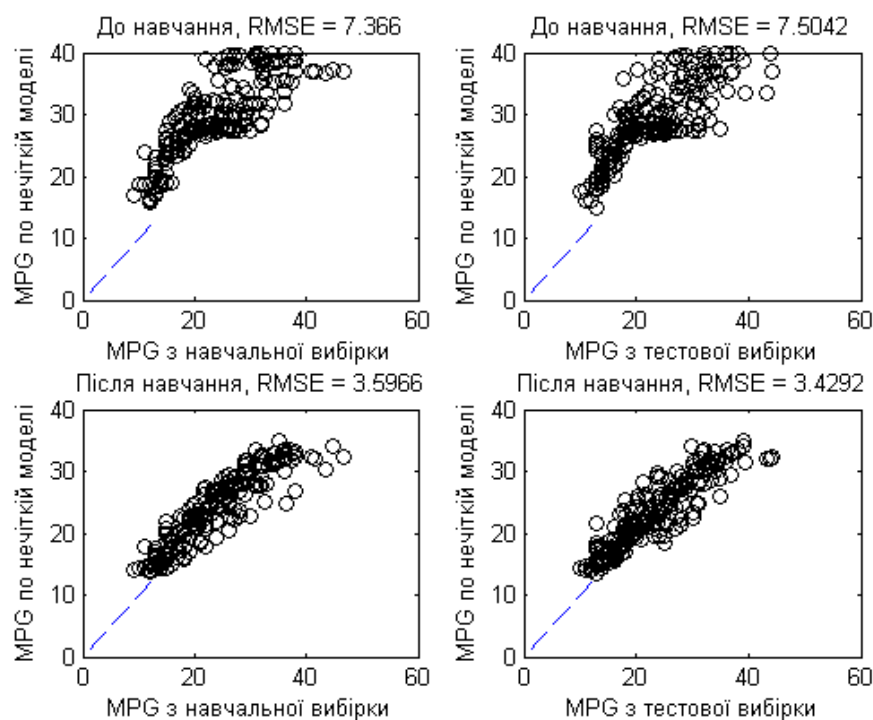


Рисунок 4.25 – Графік залежності виходу по нечіткій базі знань з правилами типу Мамдані та Сугено до і після навчання та виходу тестової вибірки

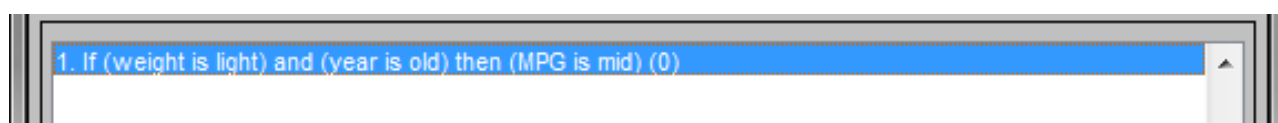


Рисунок 4.26 – Правила нечіткої бази знань типу Сугено після навчання

4.3 Аналіз результатів та висновки до розділу

В ході тестування та оцінки ефективності програмного забезпечення було проведено ідентифікацію залежності за допомогою нечіткої бази знань з правилами типу Мамдані, а також ідентифікацію за допомогою нечіткої бази знань з правилами типу Мамдані та Сугено. Тестування показало що програмне забезпечення працює відповідно до вимог поставлених до нього. Згідно результатам які наведені у таблиці 4.1 робимо висновок, що наша розробка є найоптимальнішою для ідентифікації залежностей, тому що вона дає найменшу похибку за найменшу кількість ітерацій.

Результати роботи розробки були протестовані за допомогою вибірки Auto MPG з UC Irvine Machine Learning Repository [36, 37].

Таблиця 4.1 – Результати ідентифікації за допомогою баз знань

Метод ідентифікації	RMSE на навчальній вибірці	RMSE на тестовій вибірці	Кількість ітерацій затрачених на навчання
За допомогою нечіткої бази знань з правилами Мамдані до навчання	7,3660	7,5042	26
За допомогою нечіткої бази знань з правилами Мамдані після навчання	3,4292	3,5966	
За допомогою нечіткої бази знань з правилами Мамдані та Сугено до навчання	7,3200	7,4149	5
За допомогою нечіткої бази знань з правилами Мамдані та Сугено після навчання	2,0222	1,9683	
За допомогою нечіткої бази знань з правилами Мамдані (конкурент)	2,7979	2,8842	Немає даних
За допомогою нечіткої бази знань з правилами Сугено (конкурент)	2,6965	2,9779	
Лінійна модель на 2 входи (конкурент)	3,5007	3,3373	

5 ОХОРОНА ПРАЦІ

Однією із характерних особливостей сучасного розвитку суспільства є зростання сфер діяльності людини, в яких використовуються інформаційні технології. Широке розповсюдження отримали персональні комп'ютери.

Застосування методів охорони праці дозволяє зберегти власне та суспільне здоров'я, застосовувати нові підходи до організації робочих місць, забезпечує проведення профілактичних заходів для запобігання розвитку негативних наслідків впливу ПК на здоров'я користувачів [17].

Предметом проектування є розробка програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

Небезпечні й шкідливі виробничі фактори стандартом ГОСТ 12.0.003-74 [18] поділяються на фізичні, хімічні, біологічні й психофізіологічні. Останні за характером впливу на людину підрозділяються на фізичні й нервово-психічні перевантаження, а інші - на конкретні небезпечні й шкідливі виробничі фактори.

Також небезпечні та шкідливі фактори описані в «Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин» ДСанПІН 3.3.2.007-98 [19].

При роботі з комп'ютером на користувача ПК можуть впливати такі небезпечні й шкідливі виробничі фактори:

- відсутність чи нестача природного світла;
- недостатня освітленість робочої зони;
- знижена контрастність об'єктів в порівнянні з фоном;
- перевантаження (статичні й динамічні) і нервово-психічні чинники (емоційні перевантаження, перенапруга аналізаторів, розумова перенапруга, монотонність праці).

Робота з комп'ютером вимагає значної розумової напруги і супроводжується нервово-емоційним навантаженням користувача, високою напругою зорової роботи і досить значним навантаженням на м'язи рук під час роботи з клавіатурою ПК.

5.1 Технічні рішення щодо безпечного виконання роботи

5.1.1 Обладнання приміщення

Під час розробки програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах згідно з ДСанПІН 3.3.2.007-98 дотримані наступні норми.

Площа на одного робочого, який працює за ПК, складає 5,0 кв. м.. Площа приміщень з ПК розраховуватись на 3 чоловік.

Робочі столи з ВДТ розташовані так, що відстані між бічними поверхнями ВДТ 1,2 м, відстань від тильної поверхні одного ВДТ до екрана іншого ВДТ - 2,5 м.

Стіни, стеля і підлога та обладнання кабінетів комп'ютерної техніки мають покриття із матеріалів з матовою фактурою з коефіцієнтом відбиття: стін - 40%, стелі - 80 %, підлоги - 25 %, предметів обладнання - 50% (робочого столу -50%, корпусу дисплею та клавіатури - 50 %, шаф та стелажів - 60 %).

Поверхня підлоги має антистатичне покриття та є зручною для вологого прибирання.

5.1.2 Ергономічні вимоги до робочого місця розробника

Основним обладнанням робочого місця з ПК є монітор, клавіатура, принтер, робочий стіл, стілець, допоміжним - пюпітр, підставка для ніг та ін.

Конструкція робочого місця робітника забезпечує підтримання оптимальної робочої пози.

Конструкція робочого столу відповідає сучасним вимогам ергономіки і забезпечує оптимальне розміщення на робочій поверхні використовуваного обладнання (дисплея, клавіатури, принтера) і документів.

Висота робочої поверхні робочого столу з ВДТ регулюється в межах 680-800 мм, а ширина і глибина - забезпечує можливість виконання операцій у зоні досяжності моторного поля (ширина: 1000 мм, глибина - 900 мм).

Робочий стіл має простір для ніг заввишки 700 мм, завширшки 500 мм, завглибшки (на рівні колін) 550 мм..

Робочий стілець є підйомно-поворотним, регульованим за висотою, з кутом і нахилу сидіння та спинки і за відстанню від спинки до переднього краю сидіння поверхня сидіння є плоскою, передній край - заокруглений. Регулювання за кожним із параметрів здійснюється незалежно, легко і надійно фіксується.

Висота поверхні сидіння регулюється в межах 400-500 мм, а ширина і глибина становить 500 мм. Кут нахилу сидіння - до 15 град. вперед і до 5 град. назад.

Висота спинки стільця становить 310 мм, ширина - 380 мм, радіус кривизни горизонтальної площини - 400 мм. Кут нахилу спинки регулюється в межах 1-30 град. від вертикального положення. Відстань від спинки до переднього краю сидіння регулюється в межах 260- 400 мм.

Поверхність сидіння і спинки стільця є напівм'якою з нековзним, повітронепроникним покриттям, що легко чиститься і не електризується.

Екран ВДТ розташований на оптимальній відстані від очей користувача, що становить 650 мм.

Розташування екрана ВДТ забезпечує зручність зорового спостереження у вертикальній площині під кутом +30 град. до нормальної лінії погляду працюючого.

Клавіатура розташована на поверхні столу на відстані 150 мм від краю, звернутого до працюючого. У конструкції клавіатури передбачений опорний пристрій, який дає змогу змінювати кут нахилу поверхні клавіатури у межах 5-15°. Висота середнього рядка клавіш 25 мм. Поверхня клавіатури матова з коефіцієнтом відбиття 0,4.

Розташування пристрою введення - виведення інформації забезпечує добру видимість екрана ВДТ, зручність ручного керування в зоні досяжності моторного поля і за висотою - 1000 мм, за шириною 450 мм.

Робоче місце з ВДТ обладнане пюпітром [21] для документів, що легко переміщуються.

Для забезпечення захисту і досягнення нормованих рівнів комп'ютерних випромінювань застосовуються приєкранні фільтри, локальні світлофільтри ці засоби пройшли лабораторні випробування і мають відповідний сертифікат.

Робочого місця з ВДТ оснащено лазерним принтером [22] параметри лазерного випромінювання відповідають вимогам СанПіН № 5804-91.

5.2. Технічні рішення з виробничої санітарії

5.2.1 Мікроклімат

Відповідно до ДСанПІН 3.3.2.007-98 [19] у виробничих приміщеннях на робочих місцях мають забезпечуватись оптимальні значення параметрів мікроклімату: температури, відносної вологості й рухливості повітря. Вони нормуються ГОСТ 12.1.005-88 [24]. Рівні позитивних і негативних іонів у повітрі приміщень нормуються санітарно-гігієнічними нормами № 2152-80 [25].

Параметри мікроклімату, що нормуються: температура (t °С) і відносна вологість повітря (W , %), швидкість його переміщення ($\frac{M}{c}$), потужність теплових випромінювань ($\frac{Bt}{m^2}$).

Категорія робіт при розробці програмного забезпечення 1а.

Допустимі параметри мікроклімату для умов, що розглядаються (категорія робіт та період року) наведені в таблиці 5.1.

Таблиця 5.1 – Параметри мікроклімату

Період Року	Допустимі		
	$t, ^\circ C$	$W, \%$	$V, \frac{M}{c}$
Теплий	22-28	55 при 28°C	0,1-0,2
Холодний	21-25	75	не більш 0,1

Для забезпечення необхідних за нормативами параметрів мікроклімату проектом передбачено:

1. Кондиціонування повітря за допомогою кондиціонера для приміщень Panasonic Standart CS/CU-YW07MKD: холодовиробництво - 2100 Вт, тепло виробництво - 2100 Вт, Рівень шуму 24 Дб [23].

2. Підтримання тепла приміщення в холодний період року за допомогою автономної системи опалення.

3. Використання екологічних теплоізоляційних матеріалів.

5.2.2 Склад повітря робочої зони

Склад повітря у робочій зоні нормується за ГОСТ 12.1.005-88 [26]. Забруднення повітря робочої зони регламентується гранично-допустимими концентраціями (ГДК) в мг/м^3 .

В умовах, що розглядаються в проекті, можливими забруднювачами повітря можуть бути: пил нетоксичний, озон, формальдегід, вуглекислий газ (таблиця 5.2).

Рівні іонізації повітря приміщень: кількість іонів в 1 см куб. повітря $n+ n-3000 - 5000$.

Для забезпечення складу повітря робочої зони проектом передбачені такі рішення:

1. Видалення шкідливих речовин, що потрапляють у повітря робочої зони, за рахунок вентиляції, аспірації, очищення і нормалізації повітря за допомогою кондиціонерів.

2. Для зменшення запиленості використовується розпилювач води.

Таблиця 5.2 – Гранично допустимі концентрації шкідливих речовин для повітря атмосфери (ГДК)

Назва речовини	ГДК, $\frac{\text{мг}}{\text{м}^3}$		Клас небезпечності
	Максимально разова	Середньо добова	
Озон	0,16	0,03	4
Вуглець (окис CO)	3	1	4
Формальдегід	0,035	0,003	2
Пил нетоксичний	0,5	0,15	4

5.2.3 Виробниче освітлення

Природне освітлення.

Природне освітлення нормується коефіцієнтом природного освітлення - (КПО) (формула 5.1).

$$e = \frac{E_{\text{вн}}}{E_{\text{зов}}} * 100\%, \quad (5.1)$$

де $E_{\text{вн}}$ – внутрішня природна освітленість у приміщенні в місці, що розглядається, лк; $E_{\text{зов}}$ – зовнішня природна освітленість дифузним світлом всього небосхилу, виміряна одночасно з $E_{\text{вн}}$, лк.

Для умов, що розглядаються в проекті (розряд роботи - III), система природнього освітлення (бокове), пояс світлового клімату (IV-V)), нормативне значення коефіцієнта $e_{\text{min}}^{\text{III}}$, для III-го поясу світлового клімату наведені в таблиці 5.3.

Таблиця 5.3 – Норми освітленості при штучному освітленні та КЕО (для III поясу світлового клімату СНГ) при природньому та сумісному освітленні

Характеристика зорової роботи	Найменший розмір об'єкта розрізнювання	Розряд зорової роботи	Підрозряд зорової роботи	Контраст об'єкта розрізнення з фоном	Характеристика фона	Освітленість, лк		КЕО, e_n , %			
						Штучне освітлення		Природне освітлення		Сумісне освітлення	
						Комбіноване	Загальне	Верхнє або верхнє і бокове	Бокове	Верхнє або верхнє і бокове	Бокове
Високої точності	Вище 0,3 до 0,5	III	б	Середній	Середній Темний	1000	300	5	2	3	1,2

Для світлових поясів IV, V використовується формула 5.2.

$$e^{IV,V} = e^{III} \cdot m \cdot c, \quad (5.2)$$

де m і c – відповідно, коефіцієнти світлового та сонячного клімату.

Для забезпечення нормативного значення e_{min} передбачено використання ролетів для вікон.

Штучне освітлення.

За ДСанПІН 3.3.2.007-98 величина освітленості E нормується в люксах [19].

Для умов, що розглядаються в проекті (розряд робіт III), підрозряд робіт 6, система освітлення бічна, тип джерела освітлення – люмінесцентні лампи типу ЛБ-40, потужністю 40 Вт. Для забезпечення наведеного значення E передбачено підбір люмінесцентних ламп потужністю 40Вт.

5.2.4 Виробничий шум

Рівні звукового тиску в октавних смугах частот, рівні звуку та еквівалентні рівні звуку на робочих місцях, обладнаних ПЕОМ мають відповідати вимогам ГОСТ 12.1.003-83 [27].

Допустимі рівні звукового тиску нормуються (формула 5.3), допустимі рівні звуку нормуються (формула 5.4).

$$L = 20 \lg \left(\frac{P_1}{P_0} \right), \quad (5.3)$$

де P_1 – середньоквадратичне значення звукового тиску, за період часу, що розглядається, Па;

P_0 – значення звукового тиску на нижньому порозі чутливості в октавній смузі зі середньо-геометричною частотою 1000 Гц залежно від частоти, характеру робіт і характеру шуму (нормування за граничними спектрами – ГС).

$$L_A = 20 \lg\left(\frac{P_A}{P_0}\right), \quad (5.4)$$

де P_A – середньоквадратичне значення звукового тиску з урахуванням корекції А шумоміра залежно від характеру робіт і характеру шуму.

Для умов, що розглядаються в проєкті, чи умов виконання роботи (характер робіт і характер шуму) допустимі рівні звукового тиску повинні відповідати ГС, а рівні звуку L_A не повинні перевищувати 50 дБА - дивись таблицю 5.4.

Джерелами шуму в умовах, що розглядаються в роботі, є комп'ютер, а саме його рухомі частини: куллери, головка жорсткого диска, мотор оптичного привода.

Для забезпечення допустимих параметрів шуму (поліпшення шумового клімату) в приміщенні проєктом передбачено:

1. Використання комп'ютерів з пасивним охолодженням.
2. Використання шумопоглинаючих гумових чохлів для жорсткого диску та оптичного привода.
3. Розташування комп'ютера у спеціальних відсіках робочого столу.

Таблиця 5.4 – Допустимі рівні звукового тиску і рівні звуку для постійного широкопasmового шуму

Характер робіт	Допустимі рівні звукового тиску (дБ) в стандартизованих октавних смугах зі середньгеометричними частинами (Гц)									Допустимий рівень звуку, дБА
	32	63	125	250	500	1000	2000	4000	8000	
Виробничі приміщення	86	71	61	54	49	45	42	40	38	50

5.2.5 Виробничі випромінювання

В даній роботі діє електромагнітне випромінювання(ЕМВ) яке утворює ПК. ЕМВ через свою негативну дію на генофонд і здоров'я людини підлягають гігієнічному нормуванню [19].

Значення напруженості електростатичного поля на робочих місцях з ВДТ (як у зоні екрана дисплея, так і на поверхнях обладнання, клавіатури, друкувального пристрою) мають не перевищувати гранично допустимих за ГОСТ 12.1.045-84 [28]. Порогові інтенсивності теплової дії електромагнітних хвиль на організм нормуються залежно від діапазону частот за електричною і магнітною складовими ЕМВ.

Напруженість ЕМВ вимірюється у вольтах на метр ($\frac{В}{м}$), а напруженість магнітного поля — в амперах на метр ($\frac{А}{м}$). У хвильовій зоні інтенсивність поля оцінюється величиною енергії, яка припадає на одиницю поверхні і вимірюється у $\frac{Вт}{м^2}$ або у похідних одиницях $\frac{мВт}{м^2}$.

У виробничих приміщеннях, де є джерела ЕМВ радіочастотного діапазону, допустимі значення ЕМВ контролюються шляхом вимірювання напруженості Н та Е на робочих місцях і місцях можливого перебування працюючих. Контроль параметрів ЕМВ проводиться один раз на рік, а також при введенні в експлуатацію нових установок або таких, що пройшли ремонтні роботи чи відбулася зміна технологічного процесу. Результати вимірювань заносять в спеціальний журнал.

Гранично допустиму щільність потоку енергії ЕМВ встановлюють виходячи з допустимого значення енергетичного навантаження на органом людини й часу перебування її у зоні опромінення. У всіх випадках це навантаження не може перевищувати $10 \frac{Вт}{м^2}$.

Допустимі рівні вищі, але не більше яку 2 рази, у випадку, коли час дії ЕМВ на персонал не перевищує 50 % тривалості робочого часу.

Для забезпечення безпеки від дії ЕМВ використовують комплекс засобів захисту, які умовно поділяються на організаційні, інженерно-технічні, лікувально-профілактичні.

Організаційні заходи: оптимальне розміщення технологічного устаткування, дотримання гігієнічно-обґрунтованих режимів праці та відпочинку, зменшення часу перебування у зоні опромінення.

Технічні заходи: віддалення робочого місця від джерела випромінювання.

5.3 Пожежна безпека

Причини пожеж у приміщенні, де розробляється ПЗ може бути наступними [29]:

- несправність виробничого устаткування;
- несправність і перевантаження (перегрівання) електричних пристроїв;
- необережне поводження з вогнем (паління і застосування відкритого вогню в заборонених місцях, залишення без нагляду електронагрівальних приладів).

Дана робота не пов'язана роботою з роботою з пожежонебезпечними речовинами – категорія приміщень Г (таблиця 5.5) і класи приміщень і зон з вибухо- і пожежонебезпеки 0 (таблиця 5.6) [30].

Таблиця 5.5 – Категорія приміщення за вибухо- та пожежонебезпекою

Категорія приміщення	Характеристика речовин та матеріалів, що знаходяться у приміщенні
Г	Негорючі речовини і матеріали в гарячому, розжареному або розплавленому стані, процес обробки яких супроводжується виділенням променевої теплоти, іскор та полум'я, а також горючі гази, рідини і тверді речовини, які спалюються або утилізуються у вигляді палива.

Таблиця 5.6 – Клас приміщень і зон за вибуховою і пожежною небезпекою

Клас приміщень	Фактори вибухо- та пожежонебезпечності, інші умови
0	Простір, у якому вибухонебезпечне середовище присутнє постійно або протягом тривалого часу (може мати місце тільки в межах корпусів технічного обладнання).

5.3.1 Технічні рішення системи запобігання пожежі

Приміщення, в якому експлуатується ПЕОМ повинне відповідати вимогам з пожежної безпеки, зазначеними у НАПБ.А.01.001-2004 [31].

Приміщення повинні своєчасно очищатися від горючого сміття і постійно утримуватися в чистоті.

Придбані за кордоном машини, механізми, устаткування, технологічне обладнання вводяться в експлуатацію лише за умови відповідності їх діючим в Україні нормативно-правовим актам з пожежної безпеки.

Противопожежні системи, установки, устаткування приміщень (протидимовий захист, пожежна автоматика, протипожежне водопостачання, протипожежні двері, клапани, інші захисні пристрої у протипожежних стінах і перекриттях тощо) повинні постійно утримуватися у справному робочому стані.

Отвори у протипожежних стінах, перегородках та перекриттях повинні бути обладнані захисними пристроями (протипожежні двері, вогнезахисні клапани, водяні завіси тощо) проти поширення вогню та продуктів горіння.

Не допускається встановлювати будь-які пристрої, що перешкоджають нормальному зачиненню протипожежних та проти-димних дверей.

Дерев'яні конструкції повинні піддаватися вогнезахисній обробці за винятком вікон, дверей, підлоги, вбудованих меблів, стелажів, якщо в будівельних нормах не зазначені інші вимоги.

Пошкодження вогнезахисних покриттів (штукатурки, спеціальних фарб, лаків, обмазок тощо) будівельних конструкцій, горючих оздоблювальних і теплоізоляційних матеріалів, повітроводів, металевих опор та перегородок повинні негайно усуватися.

У приміщеннях забороняється:

- прибирати приміщення і прати одяг із застосуванням бензину, гасу та інших ЛЗР та ГР, а також відігрівати замерзлі труби паяльними лампами та іншими засобами із застосуванням відкритого вогню;

- розкидати й залишати неприбраними промаслені обтиральні матеріали. Їх необхідно прибирати в металеві ящики, щільно закривати кришками і після

закінчення роботи видаляти з приміщення у спеціально відведені за межами будівель місця, забезпечені негорючими збірниками з кришками, які щільно закриваються.

5.3.2. Технічні рішення системи протипожежного захисту

Технічні рішення системи протипожежного захисту мають за мету, на випадок виникнення пожежі, обмежити її розповсюдження, виявити пожежу, забезпечити умови для її ліквідації, захистити працюючих від небезпечних і шкідливих факторів, пов'язаних з пожежею, а матеріальні цінності – від знищення.

Дана будівля відноситься до будівлі з штучними і захисними конструкціями з природних та штучних кам'яних матеріалів, бетону або залізобетону із застосуванням листових та плиткових негорючих матеріалів – I ступінь вогнестійкості. Мінімальні межі вогнестійкості будівельних конструкцій і максимальні межі розповсюдження полум'я по них для даного ступеня вогнестійкості будівлі подано в таблиці 5.7 [32]. У даній таблиці, у чисельнику вказуються межі вогнестійкості будівельних конструкцій; у знаменнику - межі розповсюдження полум'я по них.

Таблиця 5.7 – Мінімальні межі вогнестійкості будівельних конструкцій (у год.) і максимальні межі розповсюдження полум'я по них (у см) для I ступеня вогнестійкості будівлі

Ступінь вогнестійкості будівлі	Стіни				Колони	Сходові площадки, балки, марші сходових кліток	Плити, настили, (з утеплювачем), інші несучі конструкції перекриттів	Елементи перекриттів	
	Несучі	Самонесучі	Зовнішні не несучі	Внутрішні не несучі (перегородки)				Плити, настили, прогони	Балки, ферми, арки, рами
I	$\frac{2.5}{0}$	$\frac{1.25}{0}$	$\frac{0.5}{0}$	$\frac{0.5}{0}$	$\frac{2.5}{0}$	$\frac{1}{0}$	$\frac{1}{0}$	$\frac{0.5}{0}$	$\frac{0.5}{0}$

Протипожежна перешкода – протипожежні стіни, які мають 1 тип протипожежних перешкод та мінімальні межі вогнестійкості протипожежних перешкод – 2.5 год. Будівля має 10 поверхів. Будівля має загальну систему вентиляції та опалення. Відстань між спорудами не нормується так як ступінь вогнестійкості використовуваної будівлі та навколишніх будівель – І [33].

Також, для даної категорії споруд (категорія Г) не нормується відстань до евакуаційного виходу. Будівля має загальну систему вентиляції та опалення. Також встановлений кондиціонер. Кількість людей на 1 метр ширини евакуаційного виходу становить 100 [32].

Для забезпечення первинного захисту потрібно забезпечити будівлю вогнегасниками. Основною причиною пожежі є електроприлади, тому можна стверджувати що клас пожеж Е. Осередки пожежі можуть виникнути на невеликих площах, тому використовуємо переносні, а не пересувні вогнегасники. Для гасіння даного класу пожеж використовують вуглекислотні, хладонові, порошкові вогнегасники (таблиця 5.8) [31].

Таблиця 5.8 – Рекомендовані первинні засоби гасіння пожеж

Категорія приміщення	Гранична захищувана площа, м ²	Клас пожежі	Пінні та водні вогнегасники місткістю 10л	Порошкові вогнегасники			Хладонові вогнегасники місткістю 2(3)л	Вуглекислотні вогнегасники місткістю, л	
				2	3	10		2(3)	5(8)
Г	1800	(Е)	-	2+	2++	1+	2+	4+	2++

Крім вогнегасника, для гасіння пожеж, передбачено використання покривала розміром 1м². На поверсі, де розташоване дане приміщення, встановлений один пожежний щит на площу 500м².

ВИСНОВКИ

У результаті виконання бакалаврської дипломної роботи було розроблено програмне забезпечення для ідентифікації залежностей з правилами у різних форматах. Досягнуто мети даної роботи, а саме проведено розробку програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

У ході проведення технічного обґрунтування було розглянуто область об'єкту управління, аналіз існуючих типів нечітких баз знань.

Розглянуто різні можливі способи виконання ідентифікації залежностей. Обґрунтовано вибір варіанту ідентифікації залежності за допомогою нечіткої бази знань з правилами у різних форматах.

Виконано технічний аналіз основних аспектів, пов'язаних з розробкою: цілі, можливості використання, необхідність створення, можливість розробки, існування відповідних технічних засобів розробки.

Порівняння варіантів методів і проектних рішень показали що доцільно розробити програму для ідентифікації залежностей нечіткою базою знань у різних форматах за допомогою математичного пакету Matlab – обране проектне рішення дозволяє суттєво знизити час на розробку програмного забезпечення для ідентифікації залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

В другому розділі проведено розробку трьох математичних моделей баз знань:

- 1) варіант коли в базу знань входять правила типу Мамдані та Ларсена;
- 2) варіант бази знань з сингтонними правилами та правилами Сугено і Цукамото;
- 3) варіант бази знань зі змішаними правилами.

Також подані графічні інтерпретації моделей для кращого розуміння принципів їх функціонування.

В третьому розділі було обрано метод розробки програмного забезпечення, а саме структурний підхід, тому що за допомогою нього легко перевіряти окремі частини програмного забезпечення, зручно проводити модифікації.

Також розглянуті основні засоби та функції математичного пакету Matlab для реалізації логічного виводу нечіткої бази знань. Розглянуті настройки редактора нечітких баз знань.

В ході тестування та оцінки ефективності програмного забезпечення було проведено ідентифікацію за допомогою нечіткої бази знань з правилами типу Мамдані та Сугено. Тестування показало що програмне забезпечення працює відповідно до вимог поставлених до нього. Згідно отриманим результатам отримали висновок, що наша розробка є найоптимальнішою для ідентифікації залежностей, тому що вона дає найменшу похибку за найменшу кількість ітерацій.

У п'ятому розділі були розглянуті основні вимоги до охорони праці при виконанні роботи. Було визначено які небезпечні та шкідливі фактори можуть діяти при виконанні роботи. Всі умови праці дотримані відповідно ДСанПН 3.3.2.007-98 та іншим нормативним документам.

Отже розробка нового програмного продукту є вигідною і обґрунтованою.

В ході аналізу проблеми яка виникає при створенні бази знань зі змішаними правилами було представлено метод її вирішення.

Розроблено і описано пакет функцій і програм для ідентифікації залежностей за допомогою нечіткої бази знань в правилах у різних форматах.

Розроблене програмне забезпечення відповідає критеріям поставленим у технічному завданні.

За темою дипломної бакалаврської роботи була проведена апробація та зроблена доповідь на «XLIII регіональній науково-технічній конференції професорсько-викладацького складу, співробітників та студентів університету з участю працівників науково-дослідних організацій та інженерно-технічних працівників підприємств м. Вінниці та області».

ПЕРЕЛІК ПОСИЛАНЬ

1. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень: Навчальний посібник / С. О. Субботін — Запоріжжя: ЗНТУ, 2008. — 341 с.
2. Zadeh L. A. et al. Fuzzy Sets, Fuzzy Logic, Fuzzy Systems / L. A. Zadeh - World Scientific Press, 1996. - ISBN 981-02-2421-4.
3. Mamdani E. H. Application of fuzzy algorithms for the control of a simple dynamic plant / E. H. Mamdani - In Proc IEEE, 1974, p. 121–159.
4. Конференції ВНТУ [Електронний ресурс] // Ідентифікація залежностей за допомогою нечіткої бази знань з правилами у різних форматах <http://conf.vntu.edu.ua/allvntu/2014/inaeksu/txt/Tylets.pdf>
5. Жоль К. К. Вступ до сучасної логіки / К. К. Жоль — К.: Либідь, 2002. — 152 с.
6. Волошин О. Ф. Моделі та методи прийняття рішень : навч. посіб. для студ. вищ. навч. закл. / О. Ф. Волошин, С. О. Мащенко. – 2-ге вид., перероб. та допов. – К. : Видавничо-поліграфічний центр "Київський університет", 2010. – 336 с. - ISBN 978-966-439-267-6
7. Рутковская Д. Нейронные сети, генетические алгоритмы и нечеткие системы: Пер. с польского И. Д. Рудинского / Д. Рутковская, М. Пилиньский, Л. Рутковский — М.: Горячая линия — Телеком, 2004. — 452 с — ISBN 5-93517-103-1
8. Нечіткі технології в брендинзі [Електронний ресурс] // КНЕУ. – Режим доступу: <http://ir.kneu.edu.ua:8080/bitstream/2010/3312/1/Shtovba.pdf>
9. An Approach to Fuzzy Default Reasoning for Function Approximation [Електронний ресурс] // Osaka University. - Режим доступу: http://www.cs.osakafu-u.ac.jp/ci/Papers/pdf_file/fuzzy/Final-Version-Ishibuchi-Soft-Computing-Journal.pdf
10. Штовба С. Д. Введение в теорию нечетких множеств и нечеткую логику [Електронний ресурс] / Exponenta. - Режим доступу: <http://matlab.exponenta.ru/fuzzylogic/book1/index.php>

11. Леоненков А. В. Нечеткое моделирование в среде MATLAB и fuzzyTECH / А. В. Леоненков — СПб: БХВ-Петербург, 2003. — 736 с.
12. Hisao Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining / Hisao, Ishibuchi, Tomoharu, Nakashima, Manabu, Nii - XI, 2005, 307 p.
13. Штовба С. Д. Идентификация нелинейных зависимостей с помощью нечеткого логического вывода в системе MATLAB // Штовба С. Д. - Exponenta Pro: Математика в приложениях. - 2003.- №2.- С.9-15.
14. C++ [Электронный ресурс] // Wikipedia - Режим доступа: <http://en.wikipedia.org/wiki/C%2B%2B>
15. Java tutorials [Электронный ресурс] // Oracle - Режим доступа: <http://docs.oracle.com/javase/tutorial/>
16. Matlab [Электронный ресурс] // Mathworks - Режим доступа: <http://www.mathworks.com/products/matlab/>
17. Гандзюк М. П. Основи охорони праці : підручник / М.П. Гандзюк, Є.П. Желібо, М.О. Халімовський. 3-є вид. - К. : Каравела, 2006. - 392 с.
18. Запорожець О. І. Основи охорони праці. Підручник / О. І. Запорожець, О. С. Протоєрейський, Г. М. Франчук, І. М. Боровик - К.:Центр учбової літератури, 2009. - 264с.
19. ДСанПІН 3.3.2.007-98. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин.
20. Ролета віконна [Електронний ресурс] // Jysk - Режим доступу: http://jysk.ua/product_info.php?products_id=100276
21. Пюпітр [Електронний ресурс] // Wikipedia - Режим доступу: <http://uk.wikipedia.org/wiki/Пюпітр>
22. SENSYS LBP6000 [Електронний ресурс] // Canon - Режим доступу: http://www.canon.ua/For_Home/Product_Finder/Printers/Laser/i-SENSYS_LBP6000/
23. CS-CU-YW-07-MKD [Електронний ресурс] // Теплоклімат - Режим доступу: <http://teplo-klimat.com.ua/kondicion/condicionery/CS-CU-YW-07-MKD.html>

24. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
25. ГН 2152-80. Санитарно-гигиенические нормы допустимых уровней ионизации воздуха производственных и общественных помещений.
26. ГОСТ 12.1.005-88. Система стандартов безопасности труда. Общие санитарно-гигиенические требования к воздуху рабочей зоны.
27. ГОСТ 12.1.003-83. Система стандартов безопасности труда. Шум. Общие требования безопасности.
28. ГОСТ 12.1.045-84. Электростатические поля. Допустимые уровни на рабочих местах и требования к проведению контроля.
29. Москальова В. М. Охорона праці (питання та відповіді) : довід. / В. М. Москальова, В. А. Батлук, С. Л. Кусковець, В. Л. Филипчук - Л. : Магнолія 2011. - 442 с.
30. ОНТП 24—86. Определение категорий помещений и зданий по взрывопожарной и пожарной опасности.
31. НАПБ.А.01.001-2004. Правила пожежної безпеки в Україні.
32. СНиП 2.01.02-85. Противопожарные нормы проектирования зданий и сооружений.
33. СНиП 2.09.01—85. Производственные здания.
34. ISO №3941-77. Пожары. Классификация.
35. Структурний підхід до створення алгоритмів [Електронний ресурс] // Golomah A.V. - Режим доступу: http://nastyuschichka.blogspot.com/2012/04/blog-post_7898.html
36. UC Irvine Machine Learning Repository [Електронний ресурс] // Auto MPG - Режим доступу: <https://archive.ics.uci.edu/ml/datasets/Auto+MPG>
37. GIS-Lab [Електронний ресурс] // Среднеквадратичная ошибка (RMSE) - Режим доступу: [http://wiki.gis-lab.info/w/Среднеквадратичная_ошибка_\(RMSE\)](http://wiki.gis-lab.info/w/Среднеквадратичная_ошибка_(RMSE))

ДОДАТКИ

Додаток А
(обов'язковий)
ВНТУ

ЗАТВЕРДЖУЮ

Завідувач кафедри КСУ
д.т.н., проф. В.М. Дубовой

« 16 » травня 2014 р.

ТЕХНІЧНЕ ЗАВДАННЯ
на бакалаврську дипломну роботу
ІДЕНТИФІКАЦІЯ ЗАЛЕЖНОСТЕЙ ЗА ДОПОМОГОЮ НЕЧІТКОЇ БАЗИ
ЗНАНЬ З ПРАВИЛАМИ У РІЗНИХ ФОРМАТАХ

08-01.БДР.016.00.000 ТЗ

Керівник: д.т.н., професор кафедри КСУ

_____ С. Д. Штовба

“ _____ ” _____ 2014 р.

Розробив: студент гр. ЗСІ-106

_____ Р. О. Тилець

“ _____ ” _____ 2014 р.

Вінниця 2014

1. Назва та галузь застосування

1.1. Назва – Ідентифікація залежностей за допомогою нечіткої бази знань з правилами у різних форматах.

1.2. Галузь застосування – Медицина, економіка, політика.

2. Підстава для проведення розробки.

Тема бакалаврської дипломної роботи затверджена наказом по ВНТУ № 31 від 14 лютого 2014 р. та на засіданні кафедри КСУ протокол № 11 від 21.01.2014 р.

3. Мета та призначення розробки.

Метою бакалаврської дипломної роботи є збільшення точності ідентифікації залежностей за допомогою використання бази знань з правилами у різних форматах.

4. Вихідні дані для проведення розробки.

Бакалаврська дипломна робота виконується вперше. В ході проведення розробки повинні використовуватись такі документи:

- а) Штовба С. Д. Проектирование нечетких систем средствами MATLAB / С. Д. Штовба – М.: Горячая линия – Телеком, 2007. – 288 с.
- б) Штовба С. Д. Идентификация нелинейных зависимостей с помощью нечеткого логического вывода в системе MATLAB // С. Д. Штовба - Exponenta Pro: Математика в приложениях. – 2003. – № 2. – С. 9-15.
- в) Ротштейн А. П. Идентификация нелинейной зависимости нечеткой базой знаний с нечеткой обучающей выборкой // А. П. Ротштейн, С. Д. Штовба - Кибернетика и системный анализ. – 2006. – № 2. – С. 17-24.
- г) Ротштейн А. П. Идентификация нелинейных зависимостей нечеткими базами знаний // А. П. Ротштейн, Д. И. Кательников - Кибернетика и системный анализ. – 1998. – № 5. – С. 53-61.

5. Вимоги до розробки.

5.1. Перелік головних функцій:

- забезпечення виводу за допомогою нечітких баз знань з правилами у різних форматах;
- забезпечення точності виконань не менше 95 %;
- інтерфейс введення даних та виведення результатів;
- збереження результатів;

5.2. Основні технічні вимоги до розробки.

5.2.1. Вимоги до програмної платформи:

- WINDOWS XP або вище;
- JVM;
- MATLAB 10.0 або вище.

5.2.2. Умови експлуатації системи:

- робота на стандартних ПЕОМ в приміщеннях зі стандартними умовами;
- можливість цілодобового функціонування системи;
- текст програмного забезпечення системи є цілком закритим.

6. Стадії та етапи розробки.

6.1 Пояснювальна записка:

- техніко-економічне обґрунтування: «20» травня 2014 р.
- розробка структурної схеми та схеми даних: «22» травня 2014 р.
- розробка алгоритмів функціонування с-ми: «24» травня 2014 р.
- розробка програмного забезпечення с-ми: «30» червня 2014 р.

6.2 Графічні матеріали:

- структурна схема системи: «14» червня 2014 р.
- схема даних: «14» червня 2014 р.
- схеми алгоритмів функціонування системи: «15» червня 2014 р.
- плакати демонстраційні: «19» червня 2014 р.

7. Порядок контролю і приймання.

7.1. Хід виконання роботи контролюється керівником роботи.

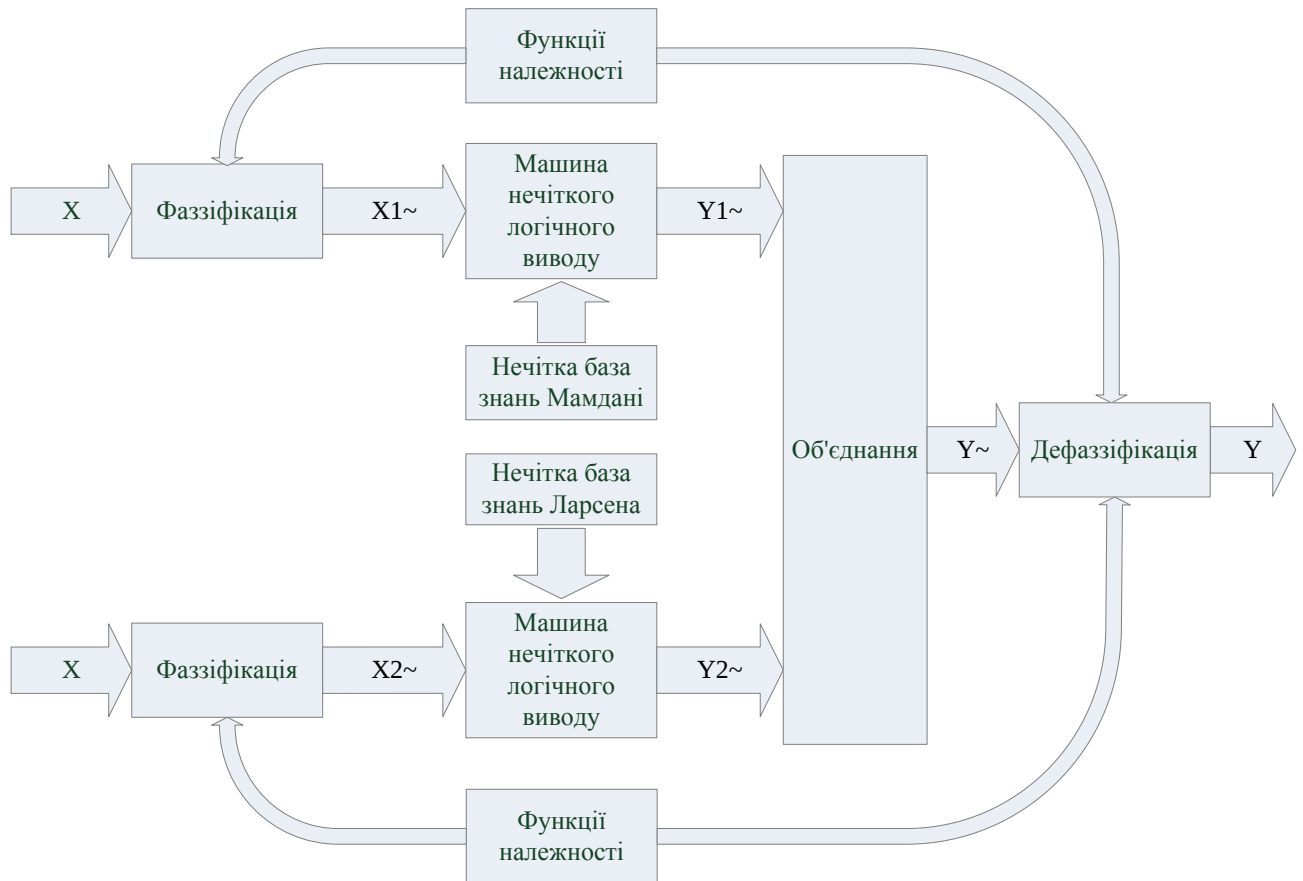
Рубіжний контроль провести до «16» червня 2014 р.

7.2. Атестація проекту здійснюється на попередньому захисті. Попередній захист бакалаврської дипломної роботи провести до «18» червня 2014 р.

7.3. Підсумкове рішення щодо оцінки якості виконання роботи приймається на засіданні ДЕК. Захист бакалаврської дипломної роботи провести до «20» червня 2014 р.

Додаток Б
(довідниковий)

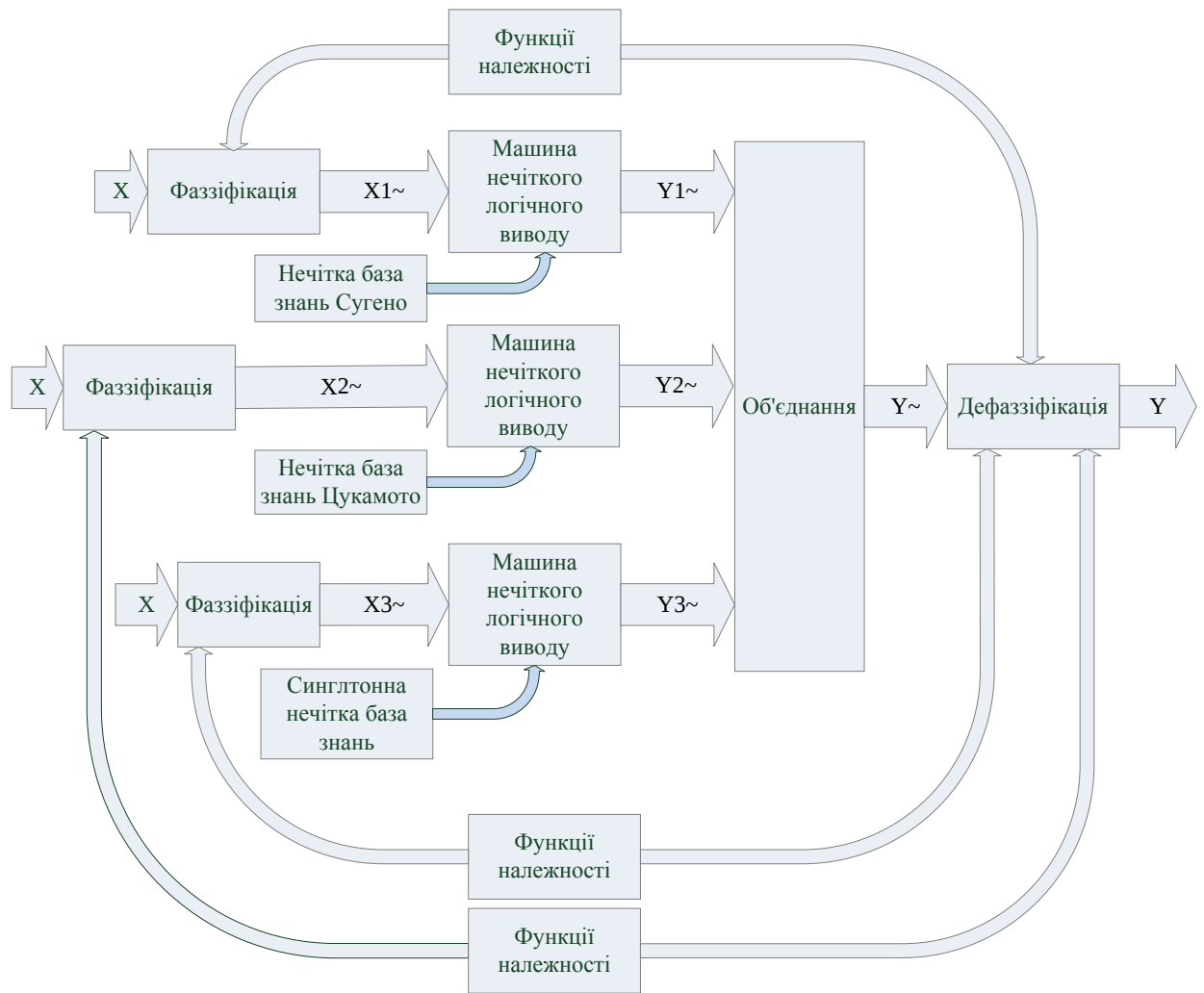
Графічна інтерпретація математичної моделі бази знань з правилами Мамдані та Ларсена



Додаток В

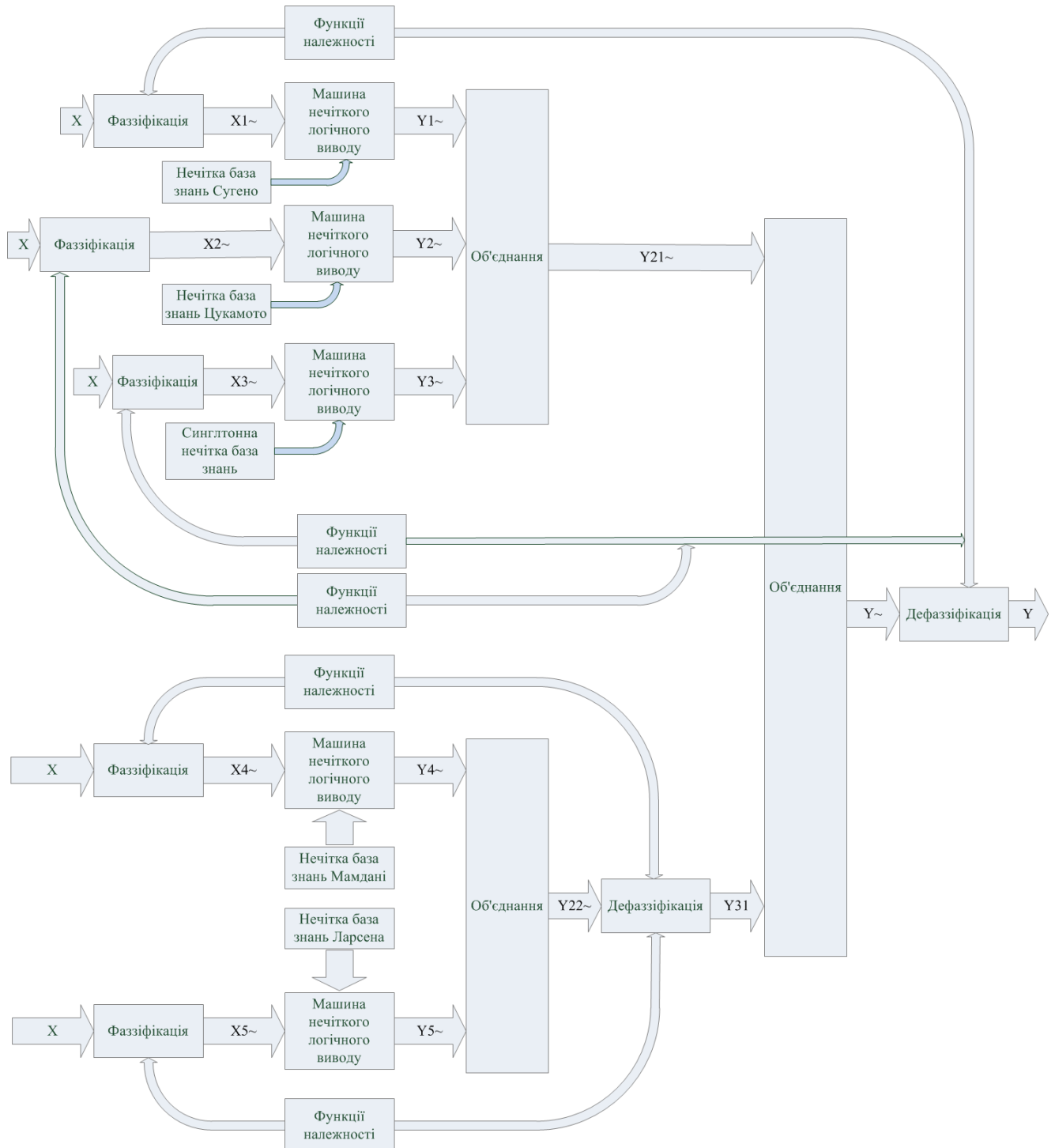
(довідниковий)

Графічна інтерпретація математичної моделі бази знань з сингтонними правилами та правилами Сугено і Цукамото



Додаток Г (довідниковий)

Графічна інтерпретація математичної моделі бази знань зі змішаними правилами



Додаток Г

(обов'язковий)

Лістинги програм

```

eval_mam_lars.m

function outputval = eval_mam_lars(inputval, mamdanifis, larsenfis,
numberPoints)

%logic output of hybrid data base with Mamdani and Larsen rules

%Mamdni FIS
mamfis = mamdanifis;

%Larsen FIS
larfis = larsenfis;

%output range
outrange = getfis(mamfis, 'outrange');

%number points of discretisation
numPts = numberPoints;

% universal - universal set for output variable
universal = (outrange(1, 1):outrange(1, 2)/(numPts-1):outrange(1,2))';

%iterator
i = 1;

%input vector
for x = inputval

%logic output for Mamdani DB
[mamoutput, mamirr, mamorr, mamarr] = evalfis(x, mamfis, numPts);

%logic output for Larsen DB
[laroutput, larirr, larorr, lararr] = evalfis(x, larfis, numPts);

%union of logic outputs of Mamdani and Larsen
union_mf = max(lararr, mamarr);

%deffuzyfication of union
defuzzRes(1,i) = defuzz(universal, union_mf, 'centroid');

i = i+1;
end

outputval = defuzzRes;

prog_mam_lars.m

%fuzzy interpretation function  $y = x^2$ 

clear
clc

%reading mamdni FIS
mamfis = readfis('MamdaniNF');
```

```

%reading Larsen FIS
larfis = readfis('LarsenNF');

%forming input vector
x = -2:0.1:2;

%number points of discretisation
nmrPts = 101;

output = eval_mam_lars(x, mamfis, larfis, nmrPts);

plot(x, output);
    eval_singl_sug.m
function outputval = eval_singl_sug(inputval, singletonfis, sugenofis,
numberPoints)

%logic output of hybrid data base with Mamdani and Larsen rules

%Singleton FIS
singlfis = singletonfis;

%Sugeno FIS
sugfis = sugenofis;

%number points of discretisation
numPts = numberPoints;

%iterator
i = 1;

%input vector
for x = inputval

%logic output for Singleton DB
[singloutput, singlirr, singlorr, singlarr] = evalfis(x, singlfis);

%logic output for Sugeno DB
[sugoutput, sugirr, sugorr, sugarr] = evalfis(x, sugfis);

%union of logic outputs of Singleton and Sugeno
output_union = [singlarr; sugarr]';

%universal - universal set for output variable
universal = [singlorr; sugorr]';

%defuzzyfication of union
defuzzRes(1,i) = defuzz(universal, output_union, 'centroid');
i = i+1;

end

outputval = defuzzRes;

    prog_singl_sug.m
%fuzzy interpretation function  $y = x^2$ 

clear
clc

%reading Singleton FIS
singlfis = readfis('SingletonNF');

```

```

%reading Sugeno FIS
sugfis = readfis('SugenoNF');

%forming input vector
x = -2:0.1:2;

%number points of discretisation
nmrPts = 101;

output = eval_singl_sug(x, singlfis, sugfis, nmrPts);

plot(x, output);
    eval_mam_sug.m
function outputval = eval_mam_sug(inputval, mamdanifis, sugenofis, numberPoints)

%logic output of hybrid data base with Mamdani and Larsen rules

%Mamdni FIS
mamfis = mamdanifis;

%Sugeno FIS
sugfis = sugenofis;

%output range
outrange = getfis(mamfis, 'outrange');

mam_num_rules = getfis(mamfis, 'NumRules');

%number points of discretisation
numPts = numberPoints;

% universal - universal set for output variable
universal = (outrange(1, 1):outrange(1, 2)/(numPts-1):outrange(1,2))';

%iterator
i = 1;

%input vector
for x = inputval

%logic output for Mamdani DB
[mamoutput, mamirr, mamorr, mamarr] = evalfis(x, mamfis, numPts);

%logic output for Sugeno DB
[sugoutput, sugirr, sugorr, sugarr] = evalfis(x, sugfis);

% defuzzification of logic output for every rule of Mamdani DB
% number of rules in Mamdani = mam_num_rules
for k = 1:1:mam_num_rules
    mamdefuzz(k) = defuzz(universal, mamorr(:, k), 'centroid');
end

%union of logic outputs of Mamdani and Sugeno
union_mf = [min(mamirr', [], 1) sugarr'];

output_universal = [mamdefuzz sugorr'];

%defuzzyfication of union
defuzzRes(i) = defuzz(output_universal, union_mf, 'centroid');

i = i+1;

```

```

end

outputval = defuzzRes;

    prog_mam_sug.m
%fuzzy interpretation function  $y = x^2$ 

clear
clc

%reading mamdani FIS
mamfis = readfis('MamdaniNF');

%reading Sugeno FIS
larfis = readfis('SugenoNF');

%forming input vector
x = -2:0.1:2;

%number points of discretisation
nmrPts = 101;
output = eval_mam_sug(x, mamfis, larfis, nmrPts);
plot(x, output);

    optimisation_main.m
clc;
clear;

%      1. mpg:           continuous
%      2. cylinders:     multi-valued discrete
%      3. displacement:  continuous
%      4. horsepower:    continuous
%      5. weight:         continuous
%      6. acceleration:  continuous
%      7. model year:     multi-valued discrete
%      8. origin:         multi-valued discrete

mpg_test = load('testSet.txt');
mpg_tr = load('trainingSet.txt');

[testnmRws testnmAttr] = size(mpg_test);
[trnmRws trnmAttr] = size(mpg_tr);

in_attr_start = 2; %from what column input attribute start
out_attr_start = 1;%from what column output attribute start

fis = readfis('mamdani_full');

fis_num_rules = getfis(fis,'numrules');

vlb_c = [750 750 3 3 3 3 3];
c0 = [1500 1500 5 5 10 10 10];
vub_c = [2000 2000 8 8 20 20 20];

vlb_b = 1.5;
b0 = 2.7;
vub_b = 4;

```

```

vlb_w(1:fis_num_rules) = 0;
w0(1:fis_num_rules) = 1;
vub_w(1:fis_num_rules) = 1;

vlb = [vlb_c vlb_b vlb_w];
x0 = [c0 b0 w0];
vub = [vub_c vub_b vub_w];

options = [];
options = optimset('Display','iter');
options.DiffMinChange = 0.01;
options.DiffMaxChange = 0.1;
options.LargeScale='off';
options.Algorithm='active-set';
options.MaxIter=250;
options.MaxFunEvals=555;
options.TolCon = 0;

[xopt, delta_tr]=fmincon(@ob_fun_duty, x0, [], [], [], [], vlb, vub, [],
options, fis, mpg_tr(:,[5 7]), mpg_tr(:, out_attr_start));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Тестирование на тестовой выборке:
fis_opt= change_fis_duty(xopt, fis);
delta_test = ob_fun_duty(xopt, fis, mpg_test(:,[5 7]), mpg_test(:,
out_attr_start))

y_tr_initial = evalfis(mpg_tr(:,[5 7]), fis);
y_tr_opt = evalfis(mpg_tr(:,[5 7]), fis_opt);
y_test_initial = evalfis(mpg_test(:,[5 7]), fis);
y_test_opt = evalfis(mpg_test(:,[5 7]), fis_opt);

RMSE_tr_initial = rms(y_tr_initial-mpg_tr(:, out_attr_start));
RMSE_tr_opt = rms(y_tr_opt-mpg_tr(:, out_attr_start));
RMSE_test_initial = rms(y_test_initial-mpg_test(:, out_attr_start));
RMSE_test_opt = rms(y_test_opt - mpg_test(:, out_attr_start));

figure(1)
subplot(2,1,1)
plot(1:trnmRws, mpg_tr(:, out_attr_start), 'ro', 'markersize', 7)
hold on
plot(1:trnmRws, y_tr_initial, 'kx', 'markersize', 7)
legend('Навчальна вибірка', 'Нечітка модель')
title('До навчання, RMSE=')
xlabel('Порядковий номер')
ylabel('Ризик')
subplot(2,1,2)
plot(1:trnmRws, mpg_tr(:, out_attr_start), 'ro', 'markersize', 8)
hold on
plot(1:trnmRws, y_tr_opt, 'kx', 'markersize', 8)
legend('Навчальна вибірка', 'Нечітка модель')
title('Після навчання, RMSE=')
xlabel('Порядковий номер')
ylabel('Ризик')

figure(2)
subplot(2,2,1)
plot(mpg_tr(:, out_attr_start), y_tr_initial, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)

```

```

title('До навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,2)
plot(mpg_test(:, out_attr_start), y_test_initial, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('До навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,3)
plot(mpg_tr(:, out_attr_start), y_tr_opt, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('Після навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,4)
plot(mpg_test(:, out_attr_start), y_test_opt, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('Після навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')

```

optimisation_prog.m

```

clc;
clear;

% SET ATTRIBUTES:
% 1. mpg: continuous
% 2. cylinders: multi-valued discrete
% 3. displacement: continuous
% 4. horsepower: continuous
% 5. weight: continuous
% 6. acceleration: continuous
% 7. model year: multi-valued discrete
% 8. origin: multi-valued discrete

in_attr_start = 2; %from what column input attribute start
out_attr_start = 1;%from what column output attribute start

numberPoints = 101;

%Load test and training sets
mpg_test = load('testSet.txt');
mpg_tr = load('trainingSet.txt');

test_x = mpg_test(:, [5 7]);
test_y = mpg_test(:, out_attr_start);

tr_x = mpg_tr(:, [5 7]);
tr_y = mpg_tr(:, out_attr_start);

[testnmRws testnmAttr] = size(test_x);
[trnmRws trnmAttr] = size(tr_x);

fis_mam = readfis('not_trained_mamdani');
fis_mam_num_rules = getfis(fis_mam, 'numrules');

fis_sug = readfis('not_trained_sugeno');
fis_sug_num_rules = getfis(fis_sug, 'numrules');

```



```

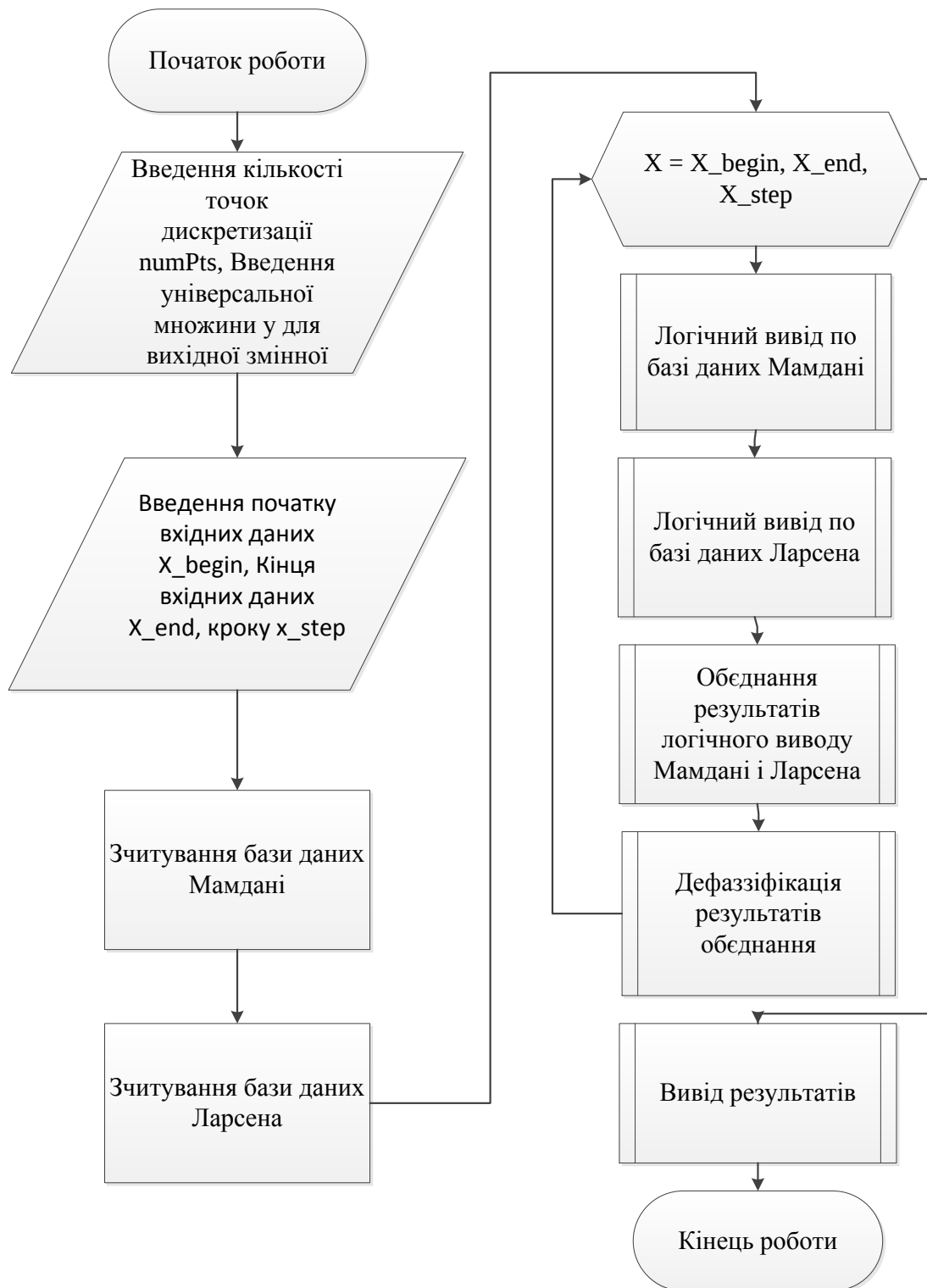
figure(1)
subplot(2,1,1)
plot(1:trnmRws, tr_y, 'ro', 'markersize', 7)
hold on
plot(1:trnmRws, y_tr_initial, 'kx', 'markersize', 7)
legend('Навчальна вибірка', 'Нечітка модель')
title('До навчання, RMSE=')
xlabel('Порядковий номер')
ylabel('Ризик')
subplot(2,1,2)
plot(1:trnmRws, tr_y, 'ro', 'markersize', 8)
hold on
plot(1:trnmRws, y_tr_opt, 'kx', 'markersize', 8)
legend('Навчальна вибірка', 'Нечітка модель')
title('Після навчання, RMSE=')
xlabel('Порядковий номер')
ylabel('Ризик')

figure(2)
subplot(2,2,1)
plot(tr_y, y_tr_initial, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('До навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,2)
plot(test_y, y_test_initial, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('До навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,3)
plot(tr_y, y_tr_opt, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('Після навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')
subplot(2,2,4)
plot(test_y, y_test_opt, 'ko', 'markersize', 7)
hold on
plot([0 13], [0 13], 'b--', 'linewidth', 1)
title('Після навчання, RMSE=')
xlabel('Ризик з тестової вибірки')
ylabel('Ризик по нечіткій моделі')

```

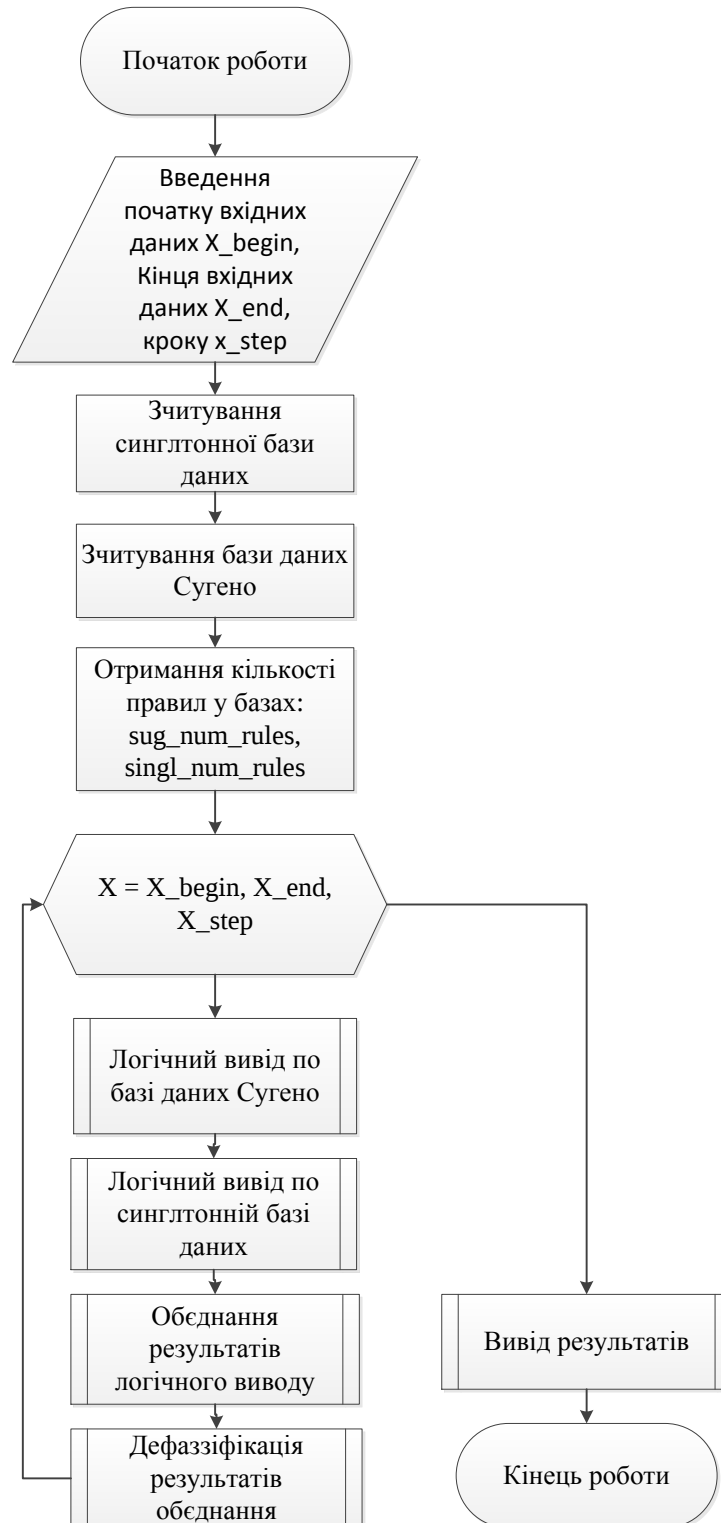
Додаток Д
(обов'язковий)

Блок-схема програми ідентифікації залежностей за допомогою бази знань з
правилами Ларсена та Мамдані



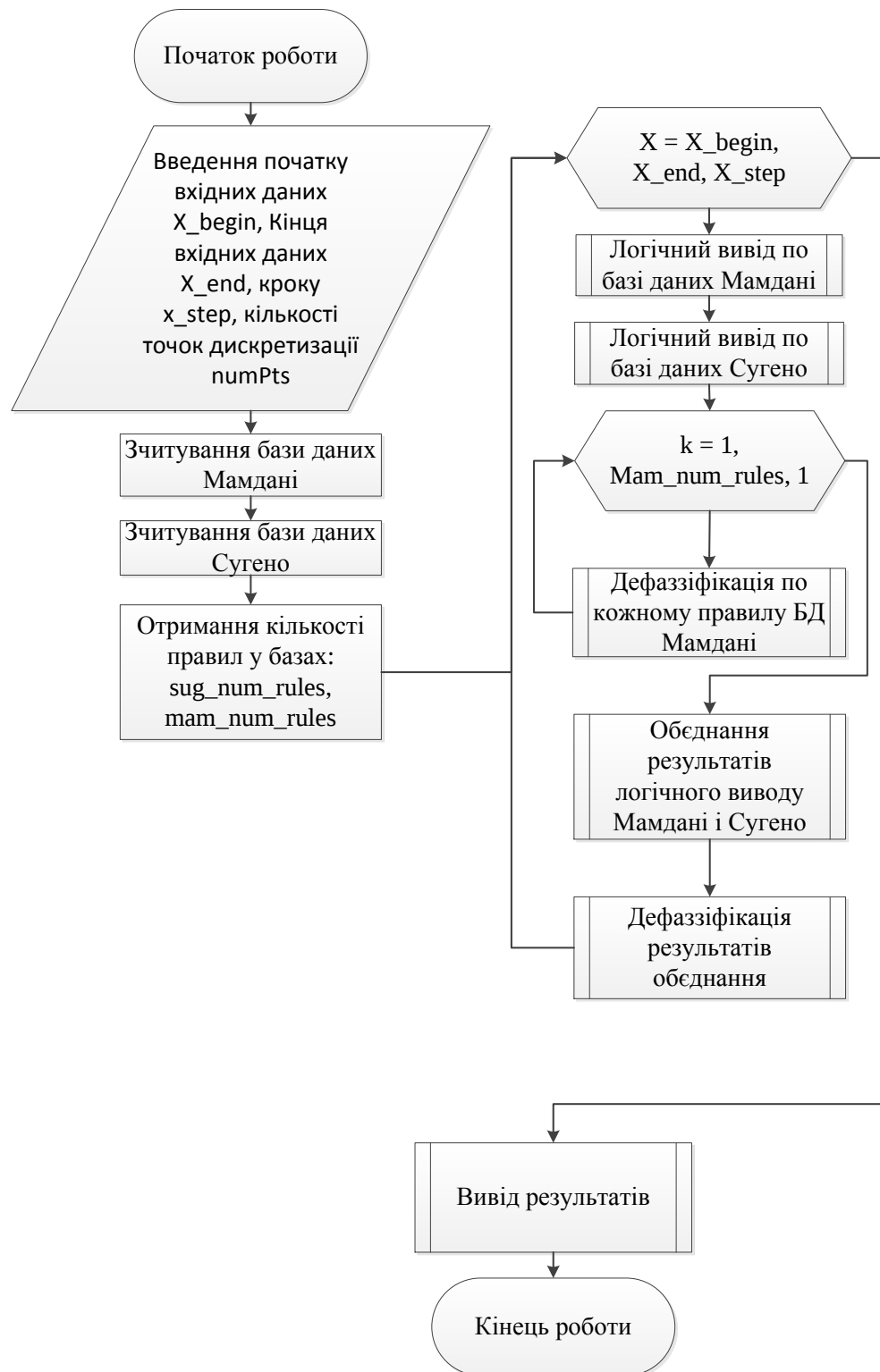
Додаток Ж
(обов'язковий)

Блок-схема програми ідентифікації залежностей за допомогою бази знань з
синглтонними правилами та правилами у форматі Сугено



Додаток К
(обов'язковий)

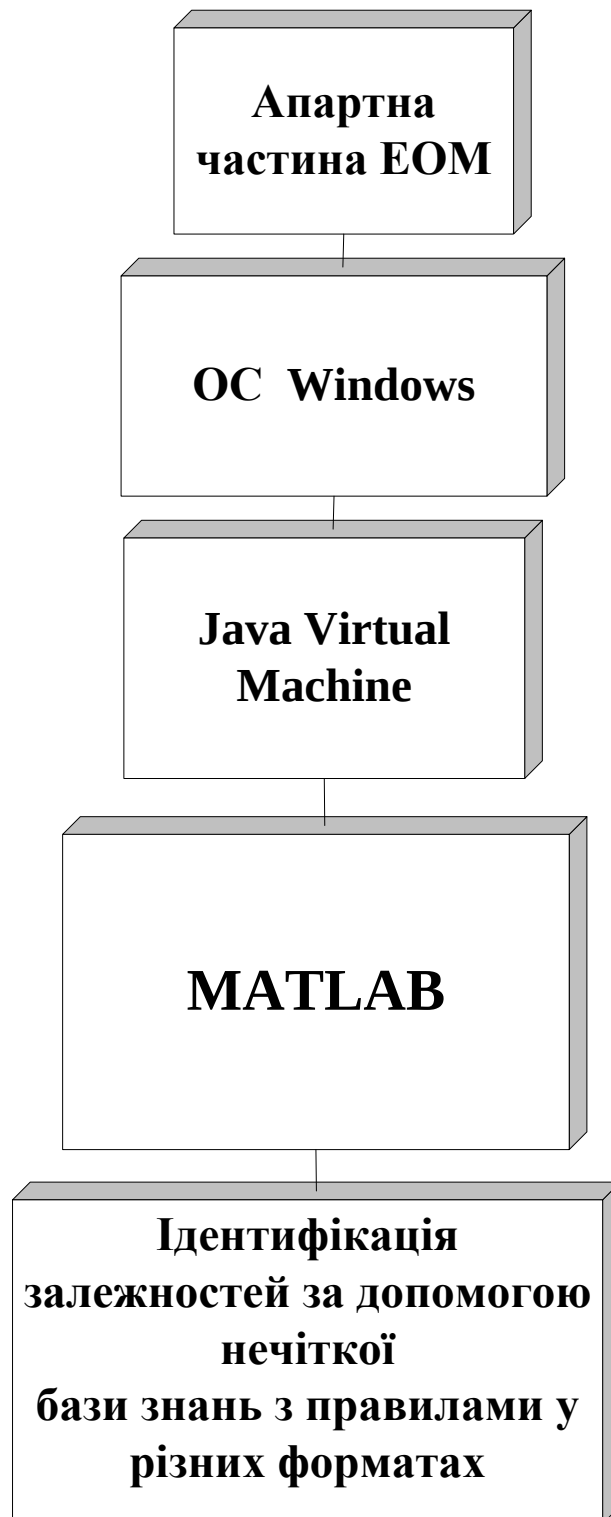
Блок-схема програми ідентифікації залежностей за допомогою бази знань зі змішаними правилами



Додаток Л

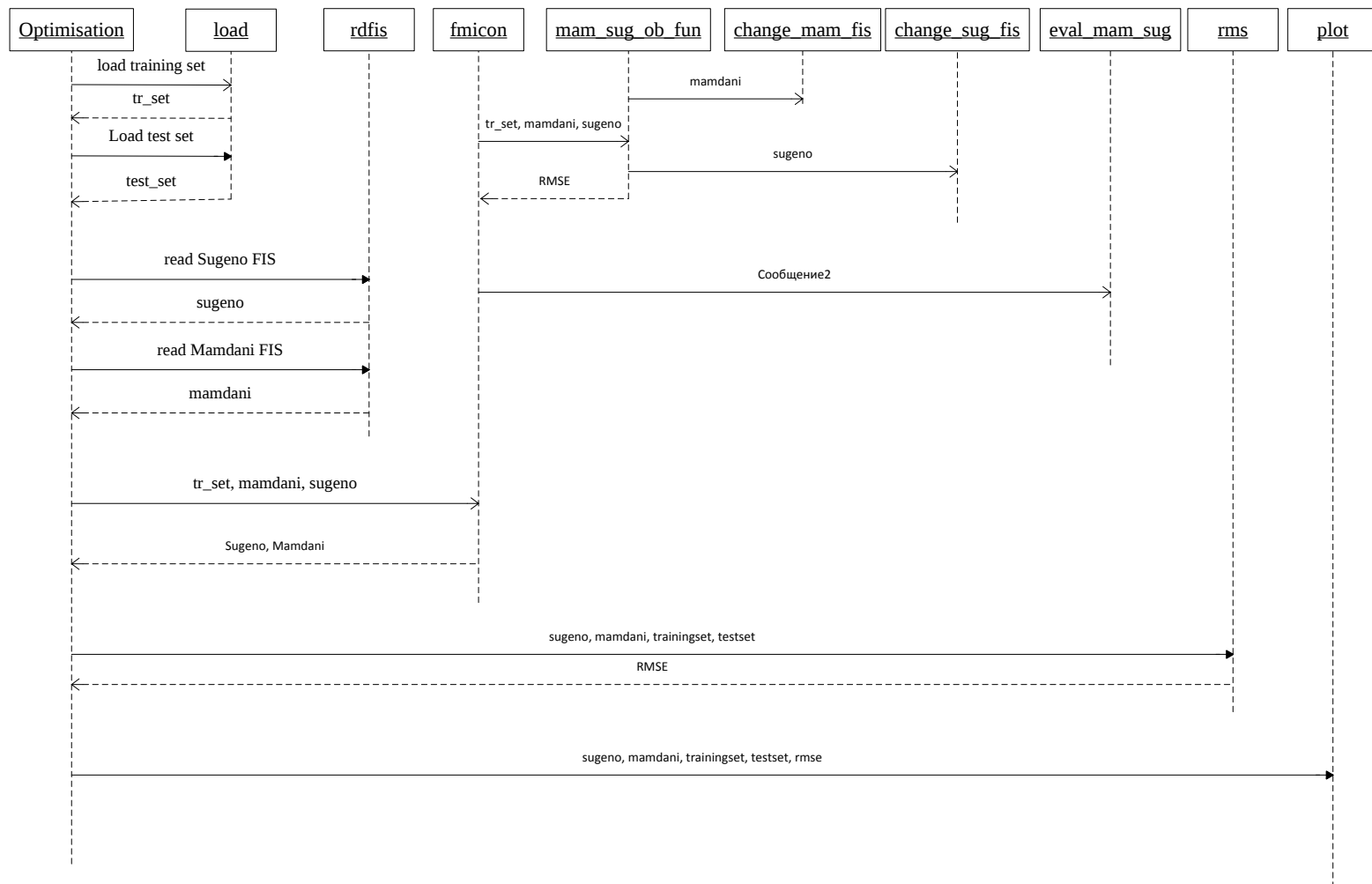
(обов'язковий)

UML-діаграма розгортання ПЗ ідентифікації за допомогою баз знань з правилами у різних форматах



Додаток М
(обов'язковий)

UML-діаграма послідовності ПЗ ідентифікації за допомогою баз знань з правилами у різних форматах



Додаток Н
(обов'язковий)

UML-діаграма варіантів використання ПЗ ідентифікації за допомогою баз знань
з правилами у різних форматах

