ЗМІСТ

Скорочення та умовні познаки 5
Вступ 7
1 Моделі нечіткого логічного виведення 8
1.1 Основні поняття нечіткої логіки 8
1.2 Операції над нечіткими множинами 13
1.3 Нечітке логічне виведення 14
1.4 Процес нечіткого умовного виведення 16
1.4.1 Етап фазифікації 18
1.4.2 Визначення відповідного правила нечіткого виведення 19
1.4.3 Згортка висловлювань і обробка нечітких правил 20
1.4.4 Методи дефазифікації 21
1.5 Огляд алгоритмів нечіткого виведення 23
1.5.1 Алгоритм Мамдані 23
1.5.2 Алгоритм Сукамото 24
1.5.3 Алгоритм Ларсена 26
1.5.4 Алгоритм Сугено 27
2 Порівняльний аналіз сучасних програмних засобів нечіткого моделювання та керування 29
2.1 Засоби нечіткого моделювання та керування в пакеті MatLab 29
2.2 Пакет проектування нечітких систем Fuzzy Tech 32
2.3 Нечітке керування у системі Трейс Моуд 34
3 Проектування інтелектуальної системи на основі нечітких знань в пакеті MatLab 36
3.1 Моделювання нечіткої системи засобами Fuzzy Logic Toolbox 36
3.1.1 Редактор систем нечіткого виведення 36
3.1.2 Редактор функцій належності 38
3.1.3 Редактор правил системи нечіткого виведення 39
3.1.4 Програма перегляду поверхні системи нечіткого виведення 41

3.2 Моделювання роботи світлофора з нечіткою логікою 42
3.2.1 Постановка завдання 42
3.2.2 Моделювання системи 45
3.2.3 Результати моделювання..... 48
Висновки 52
Перелік джерел посилання 53
Додаток А Програмний код моделі в MATLAB 53

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ЕОМ – електронна обчислювальна машина

НМ – нечітка множина

ПЗ – програмне забезпечення

САК – системи автоматичного керування

ПП – програмний продукт

ТЗ – транспортні засоби

ФН – функція належності

CG – Centre of Gravity – центр ваги

FIS – Fuzzy Inference System

ВСТУП

В останні десятиліття при побудові моделей різних економічних, політичних, біржових ситуацій активно застосовуються методи нечіткої логіки. На відміну від класичної математичної логіки, що базується на чітких поняттях типу "істина" і "хибність", в нечіткій логіці немає однозначної відповіді на питання, чи задовольняє деякий елемент розглянутої множини заданій властивості. Кожному елементу множини за допомогою функції приналежності зіставляється ступінь істинності, з якої він може задовольняти заданій властивості. Таким чином задаються нечіткі множини, потім вводяться нечіткі відносини і алгоритми нечіткого виведення. Методи нечіткої логіки забезпечують швидкість проведення розрахунків в умовах неповноти і неточності вхідної інформації.

Процес нечіткого моделювання являє собою послідовність взаємопов'язаних етапів, кожен з яких виконується з метою використання нечіткої моделі системи для рішення вихідної проблеми. Апарат теорії нечітких множин, покладений в основу розробленої в даній роботі моделі, дозволяє знайти рішення цієї проблеми в умовах невизначеності, а також слабкої структурованості оціночних показників, не вдаючись до застосування експертних оцінок. Основна перевага використання даного апарату полягає в можливості створення кількісних оцінок для лінгвістичних змінних, а також ефективного відображення залежності між цими змінними у вигляді нечітких правил.

Метою даної дипломної роботи є побудова експертної системи нечіткого виведення за алгоритмом Мамдані з використанням основних функцій пакету Fuzzy Logic Toolbox програмного середовища MatLab для керування світлофором на перехресті.

Для досягнення цієї мети були поставлені наступні завдання:

– дослідження основних понять математичного апарату нечітких множин та нечіткої логіки;

– вивчення методів побудови бази знань і вибір найкращого;

– подання продукційної моделі побудови бази знань;

– дослідження механізмів логічного виведення;

– реалізація експертної системи з використанням основних функцій пакету Fuzzy Logic Toolbox програмного середовища MatLab.

1 МОДЕЛІ НЕЧІТКОГО ЛОГІЧНОГО ВИВЕДЕННЯ

1.1 Основні поняття нечіткої логіки

Фундаментом нечіткої логіки є поняття нечіткої множини. Нехай X – універсальна множина, x – елемент X, а G – деяка властивість. Нечіткою підмножиною A універсальної множини E називається множе впорядкованих пар:

$$A = \{\mu_A(x) / x\}, \tag{1.1}$$

де $\mu_A(x)$ – функція належності, яка приймає значення в деякій заданій множині M (наприклад, M = [0,1]). Функця належності $\mu_A(x)$ вказує ступінь істинності, з яким елемент x з X належить підмножині A. Нечітка підмножина A може бути також названо нечіткою множиною на універсальній множині X. Якщо множина M складається тільки з двох чисел 0 та 1, то нечітка підмножина A стає звичайною підмножиною універсальної множини X [1][1)]

Нечітка підмножина A може бути записана перерахуванням складаючих його впорядкованих пар:

$$A = \{\mu(x_1) / x_1; \mu(x_2) / x_2; ...; \mu(x_n) / x_n\}, \tag{1.2}$$

[1)] [1] Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. 432 с.

де $x_1, x_2, ..., x_n$ – елементи універсальної множини X. Можливий також наступний запис:

$$A = \mu(x_1)/x_1 + \mu(x_2)/x_2 + ... + \mu(x_n)/x_n, \quad (1.3)$$

в якому знак «+» не є знаком операції додавання, а обозначає об'єднання перерахованих пар.

Значення функції належності $\mu_A(x)$ можуть бути взяті тільки з апріорних знань або опитування експертів. Значення експертних оцінок ступені істинності належності елемента нечіткій множині зручно представляти числами з відрізка [0,1].

Для опису об'єктів та явищ за допомогою нечіткої множини вводяться поняття нечіткої та лінгвістичної змінних [2][1)].

Нечітка змінна описується тройкою параметрів $<\alpha, X, A>$, де α – найменування змінної, X – універсальна множина (область визначення α), A – нечітка множина на X, що описує обмеження на значення нечіткої змінної α.

Лінгвістична змінна характеризується набором параметрів $<\beta, T, X, G, P>$, де β – назва лінгвістичної змінної; T – множина її значень, що представляють найменування нечітких змінних, визначенних на універсальній множині X, при цьому T називається базовою терм – множиною лінгвістичної змінної. G – синтаксична процедура, за допомогою якої генеруються нові терми (значення); P – семантична процедура, яка кожному значенню лінгвістичної змінної ставить у відповідність нечітку підмножину множини X.

Існує більше десятка типових форм кривих для завдання функцій належності. Найбільше поширення отримали: трикутна, трапецієвидна та гаусова функції належності (табл.1.1).

[1)] [2] Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.:Мир, 1976. 165 с.

Таблиця 1.1 – Основні функції належності

Найменування функції	Аналітичний вираз	Інтерпретація параметрів
Трикутна	$f_{\triangle}(x;a;b;c)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ \frac{c-x}{c-b}, & b\le x\le c\\ 0, & c\le x\end{cases}$	a, c – носій нечіткої множини; b – координата максимуму
Трапецієвидна	$f_T(x;a;b;c;d)=\begin{cases}0, & x\le a\\ \frac{x-a}{b-a}, & a\le x\le b\\ 1, & b\le x\le c\\ \frac{d-x}{d-c}, & c\le x\le d\\ 0, & d\le x\end{cases}$	a, d – носій нечіткої множини; b, c – ядро нечіткої множини
Гаусова	$f_{\gamma}(x;\sigma;c)=e^{-\frac{(x-c)^2}{2\sigma^2}}$,	c – координата максимума; σ – коефіцієнт концентрації
Сигмоїдальна	$f_{sig}(x;a;b)=\frac{1}{1+e^{-a(x-b)}}$,	a – коефіцієнт крутизни; b – координата переходу через 0,5
Сінглтон	$f_{sing}(x;a)=\begin{cases}1, & x=a\\ 0, & x\ne a\end{cases}$,	a – нечітке число, що представлене нечіткою множиною

Основними характеристиками нечітких множин є наступні характеристики.

Носій нечіткої множини A – чітка множина $\tilde{A}$ таких точок в U, для яких величина $\mu_A(x)$ додатня, тобто $\tilde{A}=\{x\,|\,\mu_A(x)>0\}$.

Висота нечіткої множини A – верхня межа її функції належності: $\sup_U \mu_A(x)$.

Нормальна нечітка множина A, якщо $\sup_U \mu_A(x) = 1$. В іншому випадку воно називається субнормальною нечіткою множиною [3][1)].

Порожня нечітка множина, якщо $\forall x \in U \ (\mu_A(x) = 0)$. Зрозуміло, що в даному універсумі U існує єдина порожня нечітка множина. Непорожню субнормальну нечітку множину можна привести до нормальної (нормалізувати) за формулою:

$$\mu'_A(x) = \frac{\mu_A(x)}{\sup_U \mu_A(x)} \tag{1.4}$$

Ядро нечіткої множини – чітке підмножина універсальної множини U, елементи якої мають ступені належності рівні одиниці. Ядро субнормальної нечіткої множини пусте.

Множина рівня α (α-зріз) нечіткої множини A – чітка підмножина універсальної множини U, що визначається за формулою:

$$A_\alpha = \{x \mid \mu_A(x) \geq \alpha\}, \ \ if \ \ \alpha \in [0,1] \tag{1.5}$$

Множина суворого рівня визначається у вигляді $A_\alpha = \{x \mid \mu_A(x) > \alpha\}$

Зокрема, носієм нечіткої множина є множина елементів, для яких $\mu_A(x) > 0$. Поняття множини рівня є розширенням поняття інтервалу. Рис. 1.1 ілюструє визначення носія, ядра, α-зрізу і α-рівня нечіткої множини.

[1)] [3] Нечеткие множества и теория возможностей. Последние достижения. Под ред. Р.Р. Ягера. М.: Радио и связь, 1986. 408 с.

Точка переходу нечіткої множини A – це такий елемент $x \in U$, для якого $\mu_A(x) = 0.5$.

Чітка множина Ã, найближча до нечіткої множини A, визначається наступним чином:

$$\mu_A(x) = \begin{cases} 0, if\ \ \mu_A(x) < 0.5 \\ 1, if\ \ \mu_A(x) > 0.5 \\ 0\ or\ \ 1,\ \ else \end{cases} \tag{1.6}$$

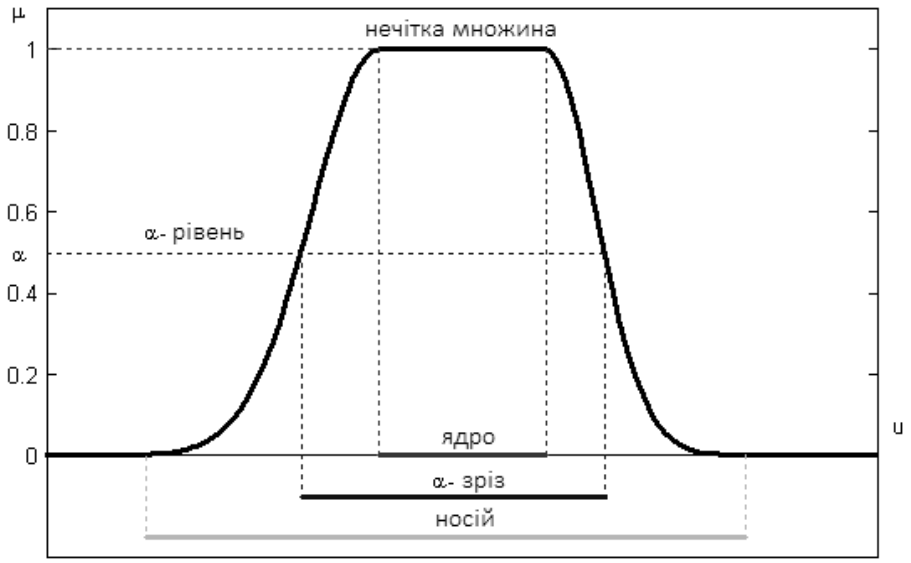

Рисунок 1.1 – Ядро, носій і α-зріз нечіткої множини

Опукла нечітка множина Ã – якщо:

$$\mu_A(\lambda u_1 + (1-\lambda)u_2) \geq min(\mu_A(u_1), \mu_A(u_2)),\ \ u_1, u_2 \in U\ \ \lambda \in [0,1] \tag{1.7}$$

На рис.1.2 наведені приклади опуклої і неопуклої нечіткої множини.

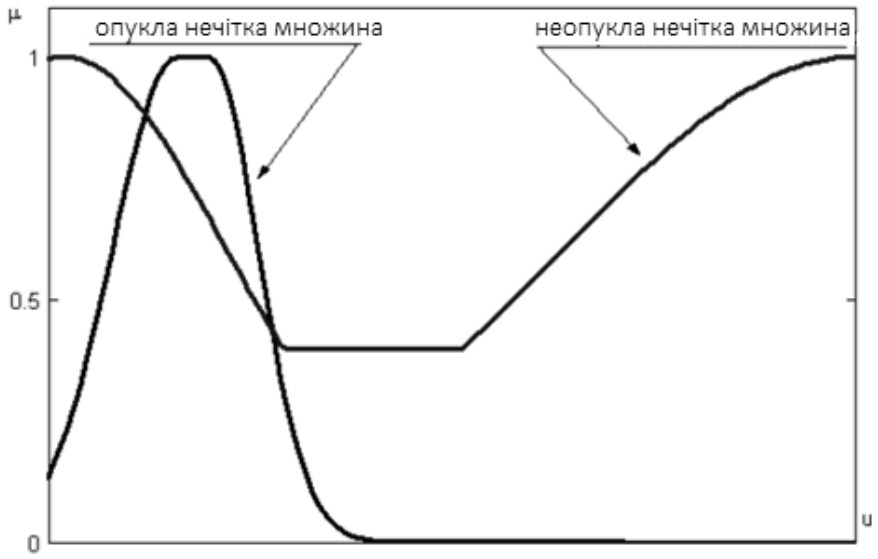

Рисунок 1.2 – Приклад опуклої і неопуклої нечіткої множини

1.2 Операції над нечіткими множинами

Над нечіткими множинами можна робити різні операції, при цьому необхідно визначити їх так, щоб в окремому випадку, коли множина є чіткою, операції переходили в звичайні операції теорії множин, тобто операції над нечіткими множинами повинні узагальнювати відповідні операції над звичайними множинами. При цьому узагальнення може бути реалізовано різними способами, через що будь-якої операції над звичайними множинами може відповідати кілька операцій в теорії нечітких множин. Для визначення перетину і об'єднання нечітких множин найбільшою популярністю користуються такі три групи операцій [4][1)]:

1) Максимінні:

$$\begin{aligned} \mu_{A\cup B}(x) &= max\{\mu_A(x),\mu_B(x)\}, \\ \mu_{A\cap B}(x) &= min\{\mu_A(x),\mu_B(x)\}. \end{aligned} \tag{1.8}$$

[1)] [4] Нечеткие множества в моделях управления и искусственного интеллекта. Под ред. Д.А. Поспелова. М.: Наука, 1986. 312 с.

2) Алгебраїчні:

$$\begin{aligned}\mu_{A\cup B}(x)&=\mu_A(x)+\mu_B(x)-\mu_A(x)\mu_B(x),\\ \mu_{A\cap B}(x)&=\mu_A(x)\mu_B(x).\end{aligned} \tag{1.9}$$

3) Обмежені:

$$\begin{aligned}\mu_{A\cup B}(x)&=min\{1,\mu_A(x)+\mu_B(x)\},\\ \mu_{A\cap B}(x)&=max\{0,\mu_A(x)+\mu_B(x)-1\}.\end{aligned} \tag{1.10}$$

Доповнення нечіткої множини у всіх трьох випадках визначається однаково:

$$\mu_{\bar{A}}(x)=1-\mu_A(x) \tag{1.11}$$

В теорії нечітких множин оператор доповнення не є єдиним. Крім загальновідомого існує цілий набір операторів доповнення нечіткої множини.

1.3 Нечітке логічне виведення

Нечіткою імплікацією $\tilde{A}\Rightarrow\tilde{B}$ називається бінарна логічна операція, результатом якої є нечітке висловлювання, значення істинності якого може визначаться однієї з наступних формул [5,6][1)]:

1) Класична імплікація Л. Заде:

$$\tilde{A}\Rightarrow\tilde{B}=max\left\{min\left[\mu_{\tilde{A}}(u),\mu_{\tilde{B}}(u),1-\mu_{\tilde{A}}(u)\right]\right\} \tag{1.12}$$

[1)] [5] Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. Вопросы анализа и процедуры принятия решений. М: Мир, 1976. С.172 – 215.

[6] Макеев А.В. Основы нечеткой логики. Учебное пособие для вузов. Н.Новгород:ВГИПУ, 2009. 59с.

2) Класична нечітка імплікація для випадку $\mu_{\tilde{A}}(u) \geq \mu_{\tilde{B}}(u)$:

$$\tilde{A} \Rightarrow \tilde{B} = max\left\{min\left[\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u), 1 - \mu_{\tilde{A}}(u)\right]\right\} \quad (1.13)$$

3) Нечітка імплікація І. Мамдані:

$$\tilde{A} \Rightarrow \tilde{B} = min\left\{\mu_{\tilde{A}}(u), \mu_{\tilde{B}}(u)\right\} \quad (1.14)$$

Нечітка імплікація грає важливу роль в процесі обробки нечітких логічних суджень. Так само, як і в класичній логіці, перший її операнд називають посилкою, або антецедентом, а другий – ув'язненням, або консеквентом. Хоча класична нечіткая імплікація знаходить найбільше застосування при вирішенні прикладних задач і залишається справедливою в разі звичайних висловлювань класичної логіки, проте інші способи обчислення нечіткої імплікації в окремих випадках виявляються більш ефективними з обчислювальної точки зору.

У системах нечіткого виведення умови і заключення формулюються у вигляді нечітких висловлювань щодо тих чи інших лінгвістичних змінних. Поняття нечіткого лінгвістичного висловлювання має фундаментальне значення для систем нечіткого виведення.

Нечіткими висловлюваннями називають висловлювання виду [7][1)]:

1) Висловлювання <β є α>, де β – найменування лінгвістичної змінної, що представляє певний об'єкт або параметр реальної дійсності, щодо якої висловлюється твердження α, що є її нечіткої оцінкою.
2) Висловлювання виду <β є *m*α>, <β є *Q*α>, <*Q*β є *m*α>, <*m*β є *Q*α>, при цьому *m* називається модифікатором (йому відповідає такі слова, як ДУЖЕ, БІЛЬШЕ АБО МЕНШ, СЕРЕДНІЙ та ін.), *Q* – квантіфікатор

[1)] [7] Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.

(йому відповідають слова типу БІЛЬШІСТЬ, БАГАТО, КІЛЬКА, ДУЖЕ МАЛО та ін.).

3) Висловлювання, що утворені з висловлювань 1-го і 2-го видів і сполучників І, АБО, ЯКЩО ..., ТО ..., ЯКЩО ... ТО ..., ІНАКШЕ ....

Нечітке висловлювання 3-го виду представляє окремий випадок нечіткої продукції, назвемо нечітким умовним висловлюванням. Деякаузгоджена безліч окремих нечітких умовних висловлювань можна розглядати як систему нечітких правил виведення [8][1)]. Основне завдання нечіткого виведення полягає в тому, щоб на основі деяких нечітких висловлювань з відомою ступеню істинності, які знаходяться в умовній частині правил виведення, оцінити ступінь істинності інших нечітких висловлювань, що є виведенням (наслідком) даного правила.

Нехай нечітка множина $\tilde{A} = \{\mu_{\tilde{A}}(x)\}$, $x \in X$ інтерпретується як умова деякого нечіткого умовного висловлювання, а нечітка множина $\tilde{A} = \{\mu_{\tilde{B}}(y)\}$, $y \in Y$ – як висновок цього ж висловлювання. При цьому універсальні множини *X* і *Y* будемо розглядати як підмножини універсальної множини *U*. Для переважної більшості практичних завдань з достатнім ступенем строгості нечітке умовне висловлювання можна розглядати як нечітку імплікацію. Тоді задача обробки нечіткого умовного висловлювання зводиться до задачі обробки нечіткої імплікації по однією з відомих формул, наприклад, (1.14).

1.4 Процес нечіткого умовного виведення

Системи нечіткого виведення призначені для перетворення значень вхідних змінних процесу управління у вихідні змінні на основі використання правил нечіткого умовного виведення. Інформація, яка надходить на вхід си-

---

[1)] [8] Сявавко М., Рибицька О. Математичне моделювання за умов невизначеності. Львів: Українські технології, 2000. 320 с.

стеми нечіткого виведення, може надходити в різній формі. У системах керування – це виміряні деяким чином вхідні змінні, що відповідають реальним змінним процесу керування. На виході системи нечіткого виведення повинні бути сформовані вихідні змінні, що відповідають керуючим змінним процесу керування.

Отриманні висновки в системах нечіткого виведення базуються на поділі процесу виведення на ряд послідовних етапів, реалізація яких виконується на основі основних положень нечіткої логіки (рис. 1.3) [9][1)].

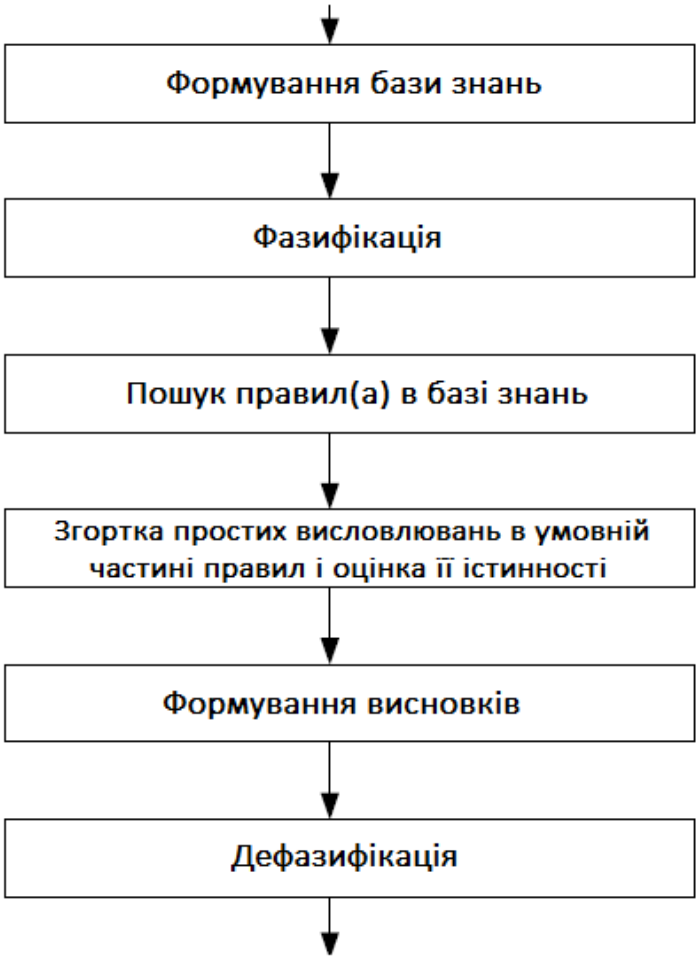

Рисунок 1.3 – Діаграма процесу нечіткого виведення

[1)] [9] Борисов А.Н., Крумберг О.А., Федоров И.П. Принятие решений на основе нечетких моделей. Примеры использования. Рига:Зинатне, 1990. 184 с.

Основною компонентою, яка багато в чому визначає одержувану якість керування (прийняття рішення), буде база знань системи нечіткого умовного виведення, що являє собою множину узгоджених правил нечіткого умовного виведення. При побудові бази знань повинні бути виконані наступні умови:

1) Правила, що утворюють базу знань системи, не повинні бути суперечливими.

2) Система правил повинна бути повною і ненадмірною.

1.4.1 Етап фазифікації

Мета етапу фазифікації – встановлення відповідності між конкретним (зазвичай числовим) значенням конкретної вхідної змінної системи нечіткого виведення і її відповідним лінгвістичним значенням, представленим функцією належності.

До початку етапу фазифікації визначаються області визначення всіх вхідних змінних $D(x_i)$, де $x_i \in D(x_i)$ – вхідна змінна системи, що є по суті універсальними множинами, на яких будуть визначатися лінгвістичні значення і відповідні їм нечіткі множини.

У загальному випадку вхідний змінної $x_i \in D(x_i)$ може бути поставлено у відповідність терм-множина лінгвістичних значень T_x, множина імен лінгвістичної змінної L_{xi} і відповідні нечіткі множини $\tilde{A}$ [10][1)].

Формально процедура фазифікації полягає в перевірці виконання умов «α є β», що містяться в правилах умовного нечіткого виведення, що знаходяться в базі знань системи. Більш просто і наочно її можна уявити в графічній формі (рис. 1.4).

Значення функції належності можна інтерпретувати як оцінку істинності виконання нечіткого висловлювання, тому в даному випадку (рис. 1.5) іс-

---

[1)] [10] Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия Телеком, 2007. 452 с.

тинність висловлювання «x_i є l_j» більше, ніж істинність висловлювання «x_i є l_{j+1}» $\mu_{lj}(\alpha_t)>\mu_{lj+1}$.

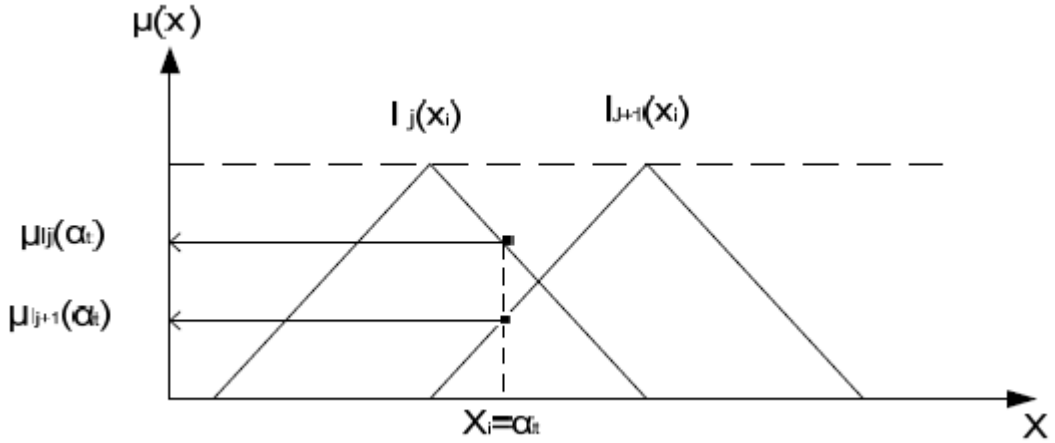

Рисунок 1.4 – Приклад фазифікації

Таким чином, при фазифікації можуть бути отримані два результата, що визначає можливість використання декількох правил нечіткого умовного виведення при отриманні остаточного заключення. Тому при побудові системи повинно бути прийнято рішення, як будуть застосовуватися результати фазифікації: буде використуватися один результат або обидва.

Процес фазифікації для конкретної вхідної змінної вважається закінченим, коли будуть перевірені всі відповідні їй можливі лінгвістичні значення. Однак в цьому немає необхідності, так як якщо дотримуються правила побудови вихідних функцій належності, то поточне значення вхідної змінної може ідентифікуватися тільки з двома сусідніми лінгвістичними значеннями.

1.4.2 Визначення відповідного правила нечіткого виведення

Для побудови висновків в базі знань системи необхідно вибрати ті правила умовного нечіткого виведення, у яких в умовній частини містяться ви-

словлювання, в яких присутня вхідні змінна з встановленим в процесі фазифікації лінгвістичним значенням.

Залежно від того, яку угоду прийнято щодо використання результатів фазифікації, буде вибрано різне число правил виведення, які потенційно можуть розглядатися як відповідні для отримання висновку. Необхідно відзначити, що завдання пошуку правил має самостійне значення, так як ефективність алгоритму пошуку безпосередньо позначиться на ефективности всієї системи в цілому. Особливо важливо час пошуку правил для нечітких систем управління технічними об'єктами, коли завдання управління вирішується в реальному масштабі часу і час, що відводиться на вироблення керуючого впливу, обмежений умовами динаміки керованого процесу.

1.4.3 Згортка висловлювань і обробка нечітких правил

У загальному випадку в умовній частині правил нечіткого виведення знаходиться складене нечітке висловлювання. Крім того, внутри простих висловлювань можливе використання логічних зв'язок, модифікаторів і квантіфікаторов. За правилами нечіткої логіки повинна бути виконана згортка висловлювань з метою отримання інтегрального умови.

На етапі фазифікації для простого висловлювання виходить оцінка його істинності у вигляді значення деякої функції належності $\mu_{lj}(\alpha_t)$. В результаті виконання згортки простих висловлювань буде отримана інтегральна оцінка істинності виконання умовної частини правил $m_1, m_2, ..., m_p$, де p – число обраних для побудови укладення правил. Для формування висновку може бути використано одне правило з максимальним значенням істинності виконання умовної частини або ж кілька правил, на основі яких буде вироблений інтегральний висновок. Кількість правил може бути зменшена, якщо встановити певний поріг істинності.

У будь-якому випадку останнім етапом буде обчислення нечіткої імплікації. В результаті буде отримана деяка множина нечітких висновків, що

випливають з конкретних правил виведення. Інтегральний висновок буде також представлено в формі нечіткої множини, на якому потрібно буде вибрати єдиний висновок. Це завдання вирішується на етапі дефаззифікації.

1.4.4 Методи дефазифікації

У загальному випадку після обчислення імплікацій буде отримано деякий набір нечітких множин $\tilde{W} = \{w_1, w_2, ..., w_p\}$ $M = \{\mu_{\tilde{w}_1}(z), \mu_{\tilde{w}_2}(z), ..., \mu_{\tilde{w}_p}(z)\}$, де p – число оброблених нечітких правил, який для кожної з вихідних змінних являє множину допустимих значень виведення (управління).

В сучасних системах управління виконавчі пристрої здатні сприймати команди управління в традиційній кількісній формі, тому на отриманій нечіткій множині допустимих керувань потрібно вибрати єдине значення, представлене в кількісної формі.

Дефазифікація в системах нечіткого виведення – це процедура знаходження звичайного (НЕ нечіткого) значення, яке може бути використано зовнішніми по відношенню до системи нечіткого виведення елементами. Для реалізації дефаззифікації використовуються декілька методів.

Найбільш простим і часто використовуваним вважається метод максимуму функції належності (рис. 1.5).

В цьому випадку для унімодальної функції належності результат дефаззифікації визначається по координаті її максимуму (рис. 1.6, а), для трапецеїдальних результат визначається по середині верхньої основи (рис. 1.6, б). Для функцій належності з декількома екстремумами метод максимуму не дає однозначного рішення.

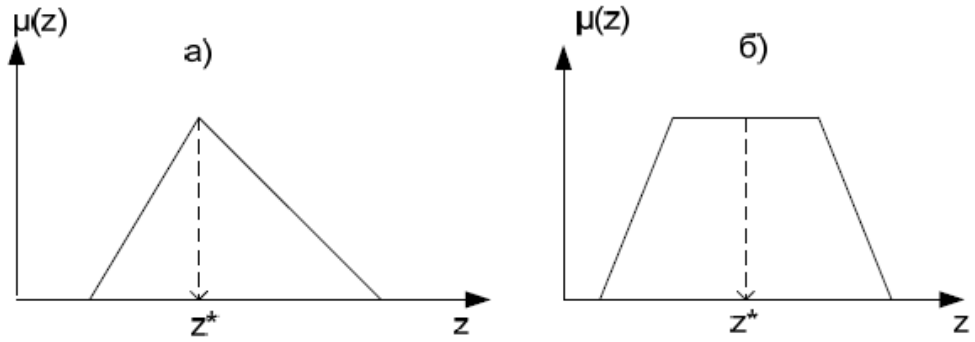

Рисунок 1.5 – Приклад дефазифікації для трикутної функції належності (а) і для трапецієвидної функції належності (б)

Метод центру площі. Шукане значення z* визначається з рівняння

$$\int_{\min Z}^{Z^*} \mu(z)dz = \int_{Z^*}^{\max Z} \mu(z)dz \ . \qquad (1.15)$$

Іншими словами, визначається абсциса прямої, що ділить площу по кривій функції належності на дві рівні частини. Метод центру площ досить незручний при реалізації, не може бути використаний, якщо функція належності задана дискретними значеннями, і є неоднозначною, так як можлива побудова декількох прямих (біссектрис площі), що поділяють площа на рівні частини.

Метод центру ваги. Ідея цього методу полягає в тому, що функцію належності розглядають як систему матеріальних точок, маси яких дорівнюють значенням функції належності. Відомо, що координата центра ваги є узагальненою характеристикою системи матеріальних точок .

Для дискретних функцій належності координата центру ваги (CG – Centre of Gravity) визначається співвідношенням

$$CG = z^* = \frac{\sum_{i=1}^{N} z_i \mu(z_i)}{\sum_{i=1}^{N} \mu(z_i)}. \qquad (`1.16)$$

Розглянуті вище етапи нечіткого виведення можуть бути реализовані різним чином, оскільки включають в себе окремі параметри, які повинні бути фіксовані чи специфіковані.

1.5 Огляд алгоритмів нечіткого виведення

Розглянемо найбільш часто використовувані алгоритми нечіткого виведення. Вважаємо, для простоти, що база знань системи складається з двох нечітких правил:

П1: якщо x є A_1 і y є B_1, то z є C_1;

П2: якщо x є A_2 і y є B_2, то z є C_2,

де x і y імена вхідних змінних, z – ім'я змінної виведення, A_1, A_2, B_1, B_2 – лінгвістичні значення з відповідними функціями належності. При цьому чітке значення виведення Z^* визначається на основі наведених правил і значень вхідних зміних x_0 і y_0 [11][1)].

1.5.1 Алгоритм Мамдані

1) Введення нечіткості. Знаходяться ступеня істинності для передумов кожного правила: $\mu_{A_1}(x_0), \mu_{A_2}(x_0), \mu_{B_1}(y_0), \mu_{B_2}(y_0)$.

2) Логічний висновок. Знаходяться рівні «відсікання» для передумов кожного правила з використанням операції min

[1)] [11] Асаи К., Ватада Д., Иваи С. Прикладные нечеткие.М.: Мир, 1993. 368 с.

$$\alpha_1 = min\left\{\mu_{A_1}(x_0),\mu_{B_1}(y_0)\right\},\ \alpha_2 = min\left\{\mu_{A_2}(x_0),\mu_{B_2}(y_0)\right\}.$$

Потім знаходяться функції належності

$$\mu_1(z) = min\left\{\alpha_1,\mu_{C_1}(z)\right\}\ i\ \mu_2(z) = min\left\{\alpha_2,\mu_{C_2}(z)\right\}. \quad (1.17)$$

3) Композиція. Співвідношення (1.17) визначають нечіткі множини, що представляють допустимі висновки, відповідні правилам П1 і П2. Для отримання інтегрального виведення, що задовольняє обидвам правилам, будується об'єднання

$$\mu'(z) = max\left\{\mu_1(z),\mu_2(z)\right\} = max\left\{min\left[\alpha_1,\mu_{C_1}(z)\right],min\left[\alpha_2,\mu_{C_2}(z)\right]\right\}. \quad (1.18)$$

4) Деффазіфікація. Знаходження *Z** виконується за допомогою одного з описаних методів дефаззифікації, наприклад по центру ваги.

Графічне представлення алгоритму нечіткого виведення Мамдані представлено на рис. 1.6.

1.5.2 Алгоритм Сукамото

Вихідні передумови ті ж, як і у алгоритму Мамдані, але передбачається, що функції $\mu_{C_1}(z)$ і $\mu_{C_2}(z)$ монотонні (рис. 1.7). Остання умова не є строго обов'язковою.

1) Введення нечіткості. Виконується так само, як і в алгоритмі Мамдані.

2) Логічний висновок. Знаходяться рівні «відсікання» для передумо кожного правила з використанням операції min:

$$\alpha_1 = min\left\{\mu_{A_1}(x_0),\mu_{B_1}(y_0)\right\},\ \alpha_2 = min\left\{\mu_{A_2}(x_0),\mu_{B_2}(y_0)\right\}.$$

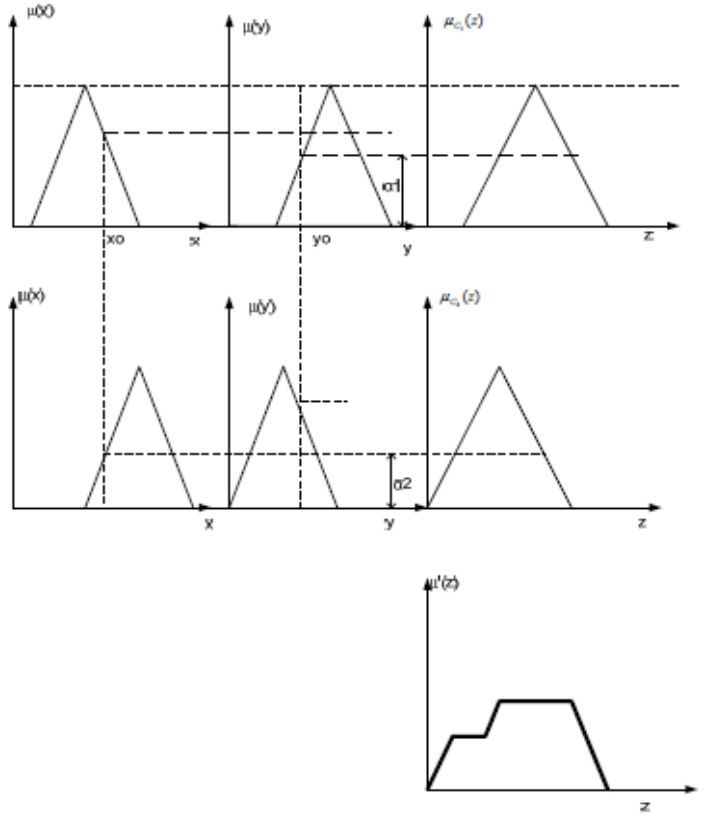

Рисунок 1.6 – Приклад виконання алгоритму Мамдані

Потім рішенням рівнянь $\alpha_1 = \mu_{C_1}(z)$ і $\alpha_2 = \mu_{C_2}(z)$ визначаються чіткі значення виведення z_1^* і z_2^* для кожного з правил П1 і П2. Інтегральний чіткий висновок знаходиться як середньозважене $\mu_{C_1}(z)$ $\mu_{C_2}(z)$:

$z^* = \dfrac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}$ або в загальному випадку

$$z^* = \frac{\sum_{j=1}^{n} \alpha_j z_j^*}{\sum_{j=1}^{n} \alpha_j}. \qquad (1.19)$$

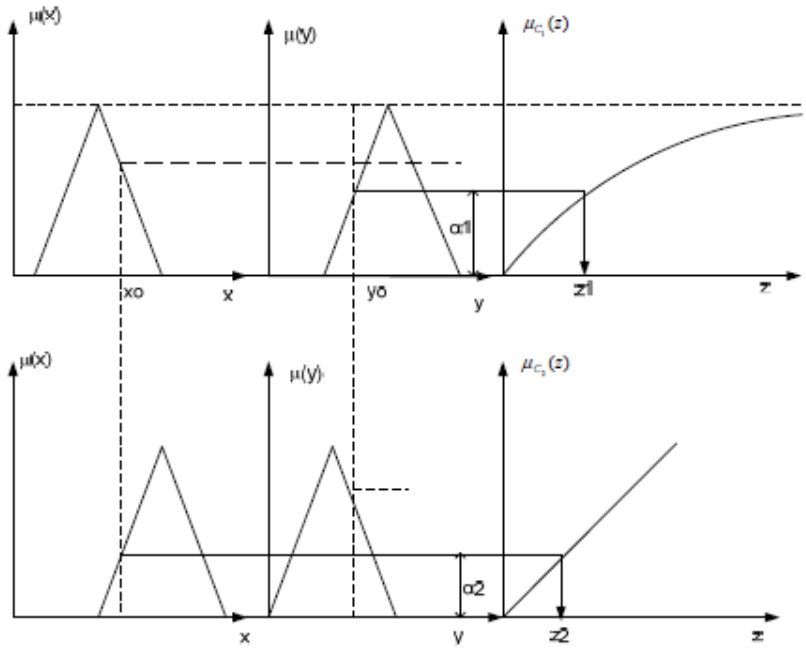

Рисунок 1.7 – Приклад виконання алгоритму Сукамото

1.5.3 Алгоритм Ларсена

1) Введення нечіткості. Виконується так само, як і в алгоритмі Мамдані.

2) Логічний висновок. Знаходяться рівні «відсікання» для передумов кожного правила з використанням операції min:

$$\alpha_1 = min\left\{\mu_{A_1}(x_0),\mu_{B_1}(y_0)\right\},\ \alpha_2 = min\left\{\mu_{A_2}(x_0),\mu_{B_2}(y_0)\right\}.$$

Потім визначаються нечіткі множини, що відповідають правилам П1 $\mu_1'(z) = \alpha_1 \cdot \mu_{C_1}(z)$ і П2 $\mu_2'(z) = \alpha_2 \cdot \mu_{C_2}(z)$.

3) Композиція. Підсумкове нечітка множина будується з допомогою операції об'єднання:

$$\mu_{\Sigma}(z) = max\left\{\mu_1'(z), \mu_2'(z)\right\} = max\left\{\alpha_1 \cdot \mu_{C_1}(z), \alpha_2 \cdot \mu_{C_2}(z)\right\}.$$

4) Дефазифікація. Виконується будь-яким з відомих методів.

Графічне представлення алгоритму нечіткого виведення Мамдані представлено на рис. 1.8.

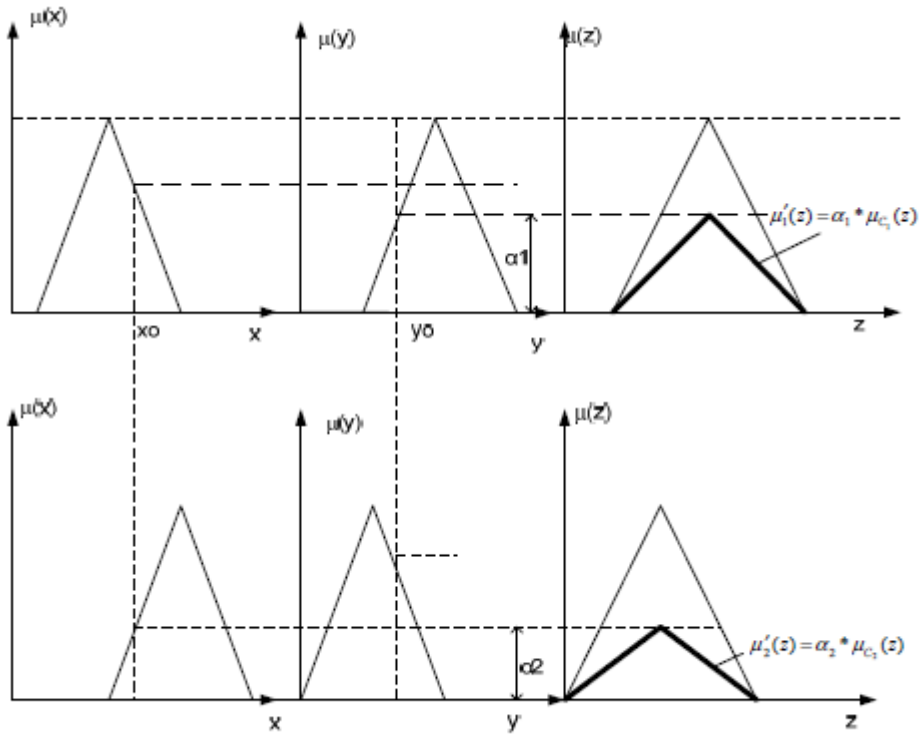

Рисунок 1.8 – Прикад виконання алгоритму Ларсена

1.5.4 Алгоритм Сугено

У цьому алгоритмі на відміну від інших використовується набір правил такого вигляду:

П1: якщо x є A_1 і y є B_1, то $z = a_1x + b_1y$;

П2: якщо x є A_2 і y є B_2, то $z = a_2x + b_2y$.

1) Введення нечіткості. Виконується так само, як і в алгоритмі Мамдані (рис.1.9).

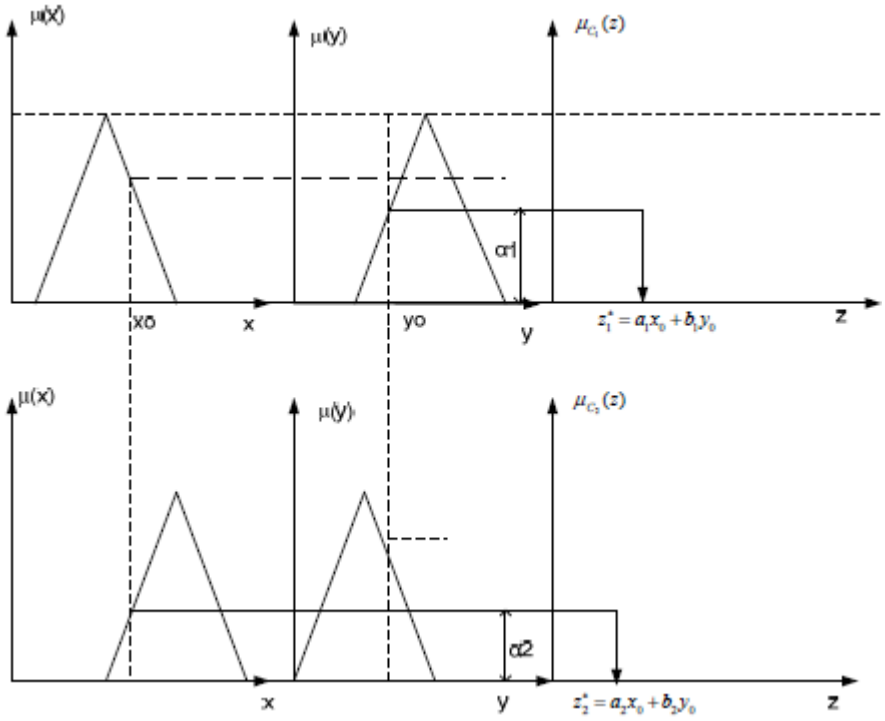

Рисунок 1.9 – Прикад виконання алгоритму Сугено

2) Логічний висновок. Знаходяться рівні «відсікання» для передумов кожного правила з використанням операції min:

$$\alpha_1 = min\left\{\mu_{A_1}(x_0),\mu_{B_1}(y_0)\right\},\ \alpha_2 = min\left\{\mu_{A_2}(x_0),\mu_{B_2}(y_0)\right\}.$$

Потім визначаються висновки по кожному з правил $z_1^* = a_1 x_0 + b_1 y_0$ і $z_2^* = a_2 x_0 + b_2 y_0$.

3) Підсумковий висновок $z^* = \dfrac{\alpha_1 z_1^* + \alpha_2 z_2^*}{\alpha_1 + \alpha_2}$

4) Дефазифікація. Виконується будь-яким з відомих методів.

2 ПОРІВНЯЛЬНИЙ АНАЛІЗ СУЧАСНИХ ПРОГРАМНИХ ЗАСОБІВ НЕЧІТКОГО МОДЕЛЮВАННЯ ТА КЕРУВАННЯ

2.1 Засоби нечіткого моделювання та керування в пакеті MatLab

Пакет Matlab компанії MathWoks (США) містить широкий набір готових функцій, які використовуються для нечіткого моделювання. У пакеті є набір алгоритмів, що утворюють так званий інструментарій, який може використовуватися для проектування, аналізу і моделювання САК. Крім цього в пакеті Matlab є набір блоків Simulink, що дозволяє в графічній формі виконувати моделювання досить складних систем, в тому числі і з використанням блоків, що реалізують нечітке керування.

Для вирішення завдань методами теорії нечітких множин в пакеті Matlab передбачений пакет нечіткої логіки Fuzzy Logic Toolbox [12][1)]. Основні можливості пакету:

– побудова систем нечіткого виводу (експертних систем, регулятора, апроксиматорів залежностей);

– побудова адаптивних нечітких систем (гібридних нейронних мереж);

– інтерактивне динамічне моделювання в середовищі Simulink.

Пакет забезпечує роботу:

– в режимі графічного інтерфейсу;

– в режимі командного рядка;

– з використанням блоків і прикладів пакета Simulink.

Моделювання нечіткого управління виконується за допомогою системи нечіткого виведення FIS (Fuzzy Inference System) (рис. 2.1), що включає редактор системи нечіткого виведення (FIS- Editor), редактор функцій

[1)] [12] Леоненков А. В. Нечеткое моделирование в MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005. 736 с.

належності (The Member Ship Function Editor), редактор правил (The Rule Editor), підсистему для перегляду правил і схем нечітких висновків (The Rule Viewer), отриманих поверхонь (The Surface Viewer).

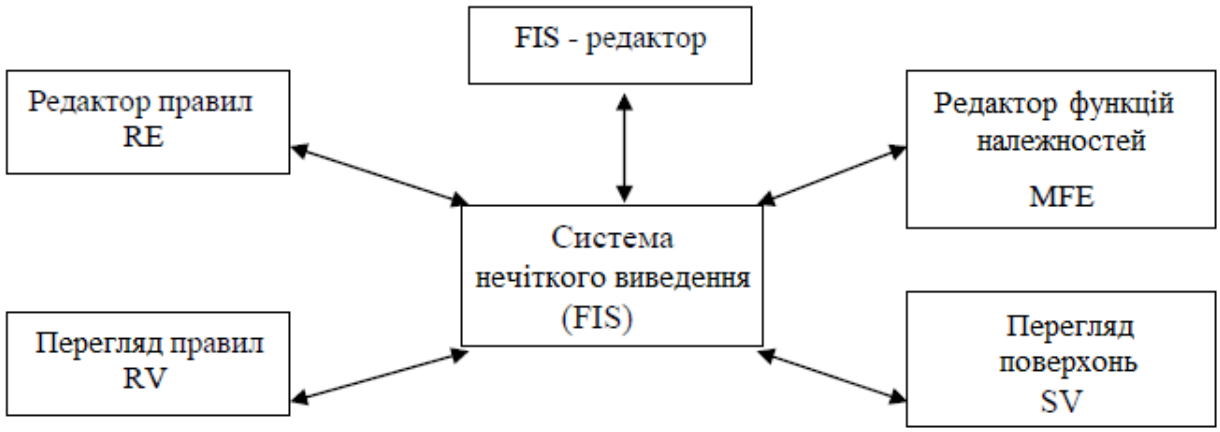

Рисунок 2.1 – Система нечіткого виведення FIS

FIS-редактор забезпечує високий рівень спілкування з системою, не має обмежень на кількість вхідних і вихідних змінних, яке обмежується лише доступним обсягом пам'яті ЕОМ.

Редактор функцій належності використовується для завдання виду функцій належності для кожної змінної. Редактор правил застосовується для редагування тексту правил умовного логічного виведення при описі поведінки модельованої системи [13][1)].

Переглядачі правил і поверхонь необхідні для візуального контролю. Переглядач правил відображає схему нечіткого виведення на останньому етапі і використовується як засіб діагностики. З його допомогою можна, наприклад, побачити які правила активні, або оцінити вплив форми окремої функції належності на результат.

[1)] [13] Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия Телеком, 2007. 288 с.

Переглядач поверхонь використовується для представлення на екрані залежності одного виходу від другого або двох входів, а також генерації та побудови картини поверхні виходу для системи.

Всі компоненти FIS можуть взаємодіяти і обмінюватися даними в процесі моделювання. У пакеті Matlab можливо використування шести видів функції належності:

– трикутної (trimf);

– трапецеідальної (tramf);

– функції належності у вигляді кривої Гаусса (gaussmf) або складеної з двох кривих Гаусса (gauss2mf);

– колокоподібної (bellmf);

– сигма-функції, призначеної для відтворення несиметричних функцій належності: sigmf – функція належності, відкрита справа, dsigmf – закрита функція належності, складена з різниці двох сигма-функцій, psigmf – закрита функція належності, що утворена з добутку двох сигма-функцій;

– трьох функцій належності, заснованих на поліноміальних кривих: zmf – несиметрична функція належності, відкрита зліва, smf – несиметрична функція надлежности, відкрита справа, pmf – закрита функція надлежности.

Крім цього в пакеті Matlab є можливість для користувача конструювати власні функції належності. Система нечіткого моделювання підтримує два основних оператора «І» та «АБО». Імплікація реалізується через оператор «І», який представлений в двох видах: min і добуток (prod), «АБО» – max і probor – оператор ймовірного «АБО», відомий ще як алгебраїчна сума і який вираховується за рівнянням $probor\ (a, b) = a + b - ab$.

Крім цих операцій в пакеті нечіткої логіки Fuzzy Logic Toolbox представлені операції концентрування і розтягування. Пакет нечіткої логіки підтримує також всі відомі операції над нечіткими відносинами. Для реалізації нечітких висновків використовуються алгоритми Мамдані (Mamdani) і Сугено (Sugeno).

2.2 Пакет проектування нечітких систем Fuzzy Tech

Пакет Fuzzy Tech, розроблений фірмою Inform Software Corporation, призначений для проектування нечітких систем. Кінцевий продукт при розробці системи – генерується за допомогою пакету програмний модуль. У пакеті є два редактора:

– редактор для створення і роботи з лінгвістичними змінними;

– редактор для роботи з базою нечітких правил.

Робота з редактором змінних. Кожна лінгвістична змінна відповідає певній вихідній змінній, яка називається базовою. Для кожної базової змінної вводяться діапазон зміни, а також значення, які їй присвоюються в разі, якщо це значення не визначене для вхідної змінної або не обчислено для вихідної. Воно позначається словом «Default» і використовується, наприклад, в разі, якщо є помилки або пропуски у вхідних даних.

Для визначення лінгвістичної змінної задається її ім'я і терм-множина можливих значень (Term name). Редактор для роботи зі змінними дозволяє графічно визначити для кожної з можливих лінгвістичних змінних функцію належності (скорочена назва MBF).

У пакеті Fuzzy Tech функції належності визначаються завданням координат точок визначення (Definition poit), які об'єднуються лінійними або нелінійними функціями. Найбільш часто використовують чотири стандартні функції належності: *Z*-типу, *λ*-типу, *P*-типу і *S*-типу.

Функції *S*-типу і *Z*-типу використовуються для крайнього правого і крайнього лівого лінгвістичних значень. Нелінійні функції належності в пакеті Fuzzy Tech представляються кубічними сплайнами. На використання цього типу функцій належності вказує параметр «Shape».

Дефазафікація виконується одним з трьох методів.

1) CoM – метод центру максимуму, який використовується, коли результатом нечіткого логічного висновку може бути кілька термів вихідної змінної. Даний метод найбільш компромісний;

2) MoM – метод максимуму;

3) CoA – метод центру ваги.

Редактор для робіт з базою правил. Блоки правил використовуються для реалізації стратегій керування. Кожен блок правил містить правила для певного рішення. Основні операції для обробки правил – це операції MAX, MIN.

Розробка нечткой системи керування виконується з допомогою CAD-системи Fuzzy Tech і включає наступні етапи, що схематично показані на рис. 2.2.

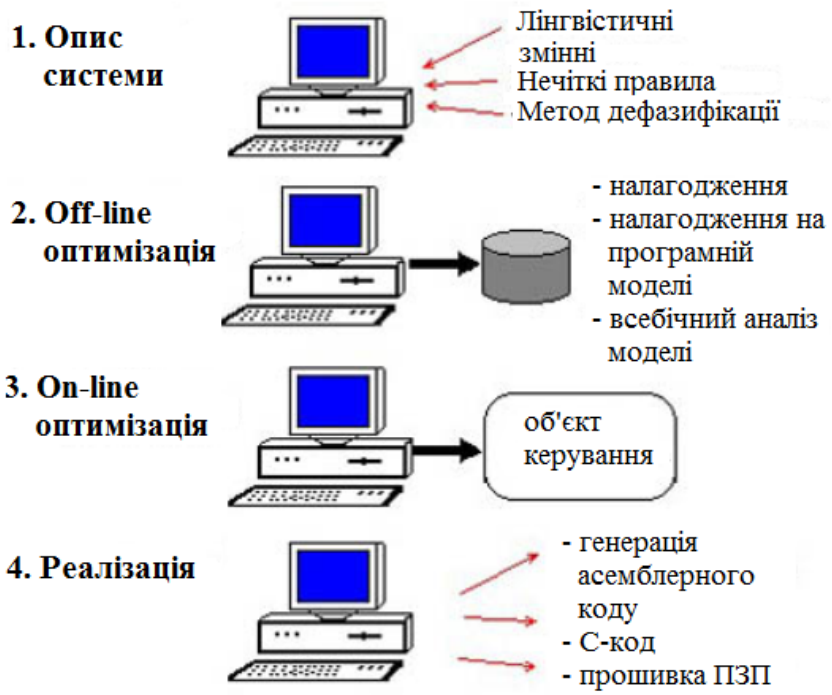

Рисунок 2.2 – Процес розробки проекту нечіткої системи керування на fuzzy TECH

1) Опис системи. На цьому етапі за допомогою засобів, доступних в пакеті, задача формалізується. Тут описуються лінгвістичні змінні, якими будуть використуватися і відповіді функції належності. Стратегія керування

описується нечіткими правилами, які об'єднані в базу знань системи. На цьому етапі використовуються CASE- технології, на основі яких побудований пакет.

2) Off-line-оптимізація. Виконується перевірка працездатності створеної системи. Для цієї мети замість реального об'єкта можна використовувати його програмну модель. Для зв'язку системи керування з моделлю використовується спеціально розроблений протокол зв'язку fTlink, в основу якого покладена концепція обміну повідомленнями Windows.

3) On-line-оптимізація. На цьому кроці розробляється система керування і реальний об'єкт керування об'єднуються в єдину систему. В цьому випадку в реальних умовах спостерігають за поведінкою системи, а також, в разі необхідності вносяться зміни в систему керування. Після налагодження створюється остаточний варіант коду для конкретного мікроконтролера. Основу програмного коду, що генерується пакетом, становить аппаратно-орієнтоване на конкретний тип контролера ядро. Генеруємий код складається з трьох основних частин:

– код бібліотечних функцій;
– сегмент бази правил і функцій належності;
– функції нечіткої системи.

2.3 Нечітке керування у системі Трейс Моуд

Пакет Трейс Моуд. (Trace Mode) фірми Адастра призначений для розробки автоматизованих систем керування технологічними процесами крім засобів розробки проектів систем, які використовують традиційні алгоритми керування, надає можливості для реалізації нечіткого керування.

Нечіткий регулятор FZCTR в системі Трейс Моуд реалізований як функціональний блок бібліотеки регулювання (рис. 2.3). Регулятор має три функціональних входу і два виходу. На вхід IP треба подавати регульоване значення; другий вхід призначений для введення значення уставки.

Для вибору типу об'єкта керування використовується вхід IC, який може встановлюватися в стан:

– "0", коли характеристики системи регулювання налаштовуються за допомогою функціональних блоків Gд, NД, SД;

– "1" – виконується регулювання малоінерційних параметрів, для яких значення постійних часу лежать в діапазоні (2 ÷ 5) ºС;

– "2" – виконується регулювання параметрів середньої інерційності з постійними часу в межах від 5 до 60 °С;

– "3" – для регулювання сільноінерціонних параметрів, значення постійних часу яких можуть бути в межах від 1 до 30 хв.

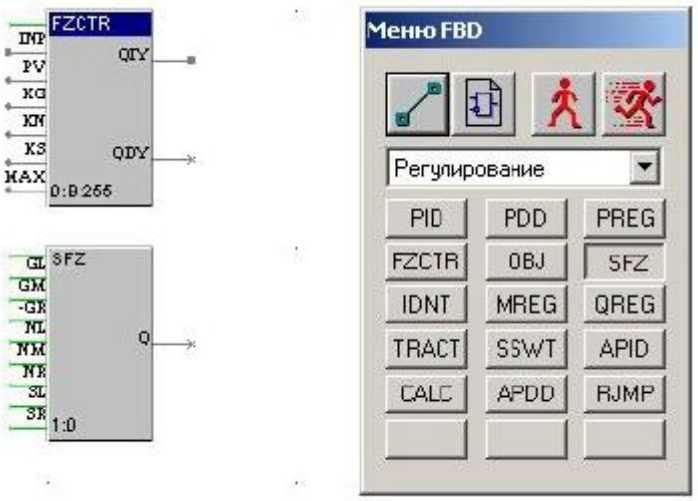

Рисунок 2.3 – Функціональний блок бібліотеки регулювання системи Трейс Моуд

На виході *Q* формується величина керуючого впливу, вихід *dQ* призначений для формування величини приросту керуючого впливу на поточному кроці керування.

Блоків Gд, NД і SД передають задані в них настройки всім присутнім в даному вузлі нечітким логічним регуляторам. Тому перед кожним з регуляторів, що мають індивідуальні настройки, слід передбачити набір цих блоків.

При цьому їх треба також розмістити в програмі, щоб вони виконувалися перед відповідним блоком нечіткого логічного регулятора.

Блок настройки FZCTR «сильне відхилення» (Gд) призначен для настройки діапазонів, які сприймаються як сильне відхилення регульованої величини від уставки і завдання коефіцієнтов для розрахунку керуючого впливу в цьому випадку.

Блок настройки FZCTR «середнє відхилення» (NД) призначен для настройки діапазонів, які сприймаються як середнє відхилення регульованої величини від уставки і завдання коефіцієнтов для розрахунку керуючого впливу в цьому випадку.

Блок настройки FZCTR «слабке відхилення» (SД) призначений для настройки діапазонів, які сприймаються як слабкі відхиленя регульованої величини від уставки і завдання коефіцієнтів для розрахунку керуючого впливу в цьому випадку.

3 ПРОЕКТУВАННЯ ІНТЕЛЕКТУАЛЬНОЇ СИСТЕМИ НА ОСНОВІ НЕЧІТКИХ ЗНАНЬ В ПАКЕТІ MATLAB

3.1 Моделювання нечіткої системи засобами Fuzzy Logic Toolbox

3.1.1 Редактор систем нечіткого виведення

Редактор систем нечіткого виводу FIS (або просто редактор FIS) є основним засобом, що використовується для створення або редагування систем нечіткого виводу в графічному режимі (рис.3.1). Редактор FIS може бути відкритий за допомогою введення функції fuzzy у вікні команд.

Редактор FIS має головне меню, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т.д.

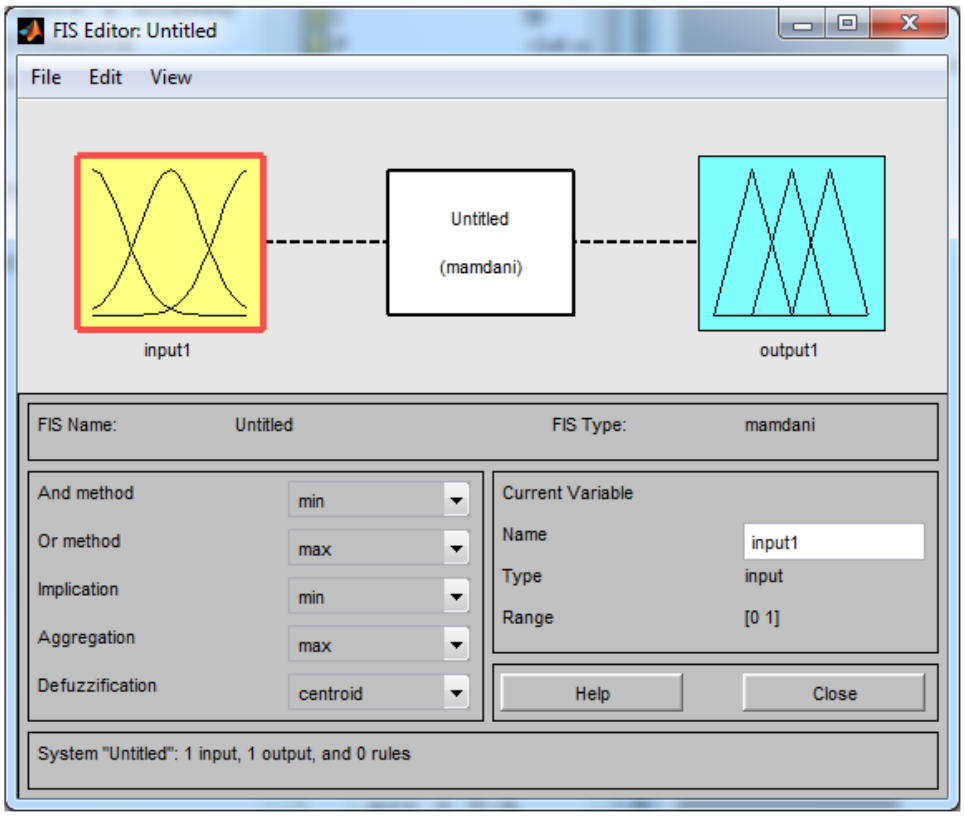

Рисунок 3.1 – Вікно редактора систем нечіткого виведення

Пункт меню File (Файл) редактора FIS дозволяє вибрати тип системи нечіткого виведення: Mamdani – типу Мамдані або Sugeno – типу Сугено, завантажити в редактор FIS існуючу систему нечіткого виведення та зберегти редаговану систему нечіткого виводу [14][1)].

Пункт меню Edit (Редагування) дозволяє викликати редактор функцій належності і редактор правил нечіткого виведення.

Пункт меню View (Вид) дозволяє викликає програму перегляду правил нечіткого виводу і програму перегляду поверхні нечіткого виведення.

У лівій нижній частині робочого інтерфейсу редактора FIS є 5 спливаючих меню:

1) And method (Метод логічної кон'юнкції) – дозволяє задати метод для виконання логічної кон'юнкції в умовах нечітких правил.

[1)] [14] Дьяконов В. MATLAB 6: учебный курс. СПб.: Питер, 2001. 592 с.

2) Or method (Метод логічної диз'юнкції) – дозволяє задати метод для виконання логічної диз'юнкції в умовах нечітких правил.

3) Implication method (Метод виведення висновку) – дозволяє задати метод для виконання (активізації) логічного висновку в кожному з нечітких правил (не використовується для систем типу Сугено).

4) Aggregation method (Метод агрегування) – дозволяє задати метод для агрегування значень функції приналежності кожної з вихідних змінних у висновках нечітких правил.

5) Defuzzification method (Метод дефазифікації) – дозволяє задати метод для виконання дефазифікації вихідних змінних в системі нечіткого виведення типу Мамдані або типу Сугено.

3.1.2 Редактор функцій належності

Редактор функцій належності (рис.3.2) призначений для завдання і редагування функцій приналежності окремих термів системи нечіткого виведення в графічному режимі. Програму-редактор функцій належності можна відкрити трьома способами: через пункт меню View → Edit membership functions ...; подвійним клацанням лівої кнопки миші по іконці, що відображає відповідну змінну; натисканням клавіш Ctrl + 2.

Редактор надає користувачеві не тільки можливість вибрати будь-яку з 11 вбудованих функцій належності, але і задати власну функцію належності. Для кожної функції належності можна змінити її ім'я, тип і параметри.

Для відображення графіків функцій належності слід вибрати необхідну змінну в лівій частині графічного інтерфейсу редактора під заголовком FIS Variables (Змінні FIS).

Щоб вибрати потрібну функцію належності, слід клацнути на ній або її мітці в основному вікні з графіками функцій належності.

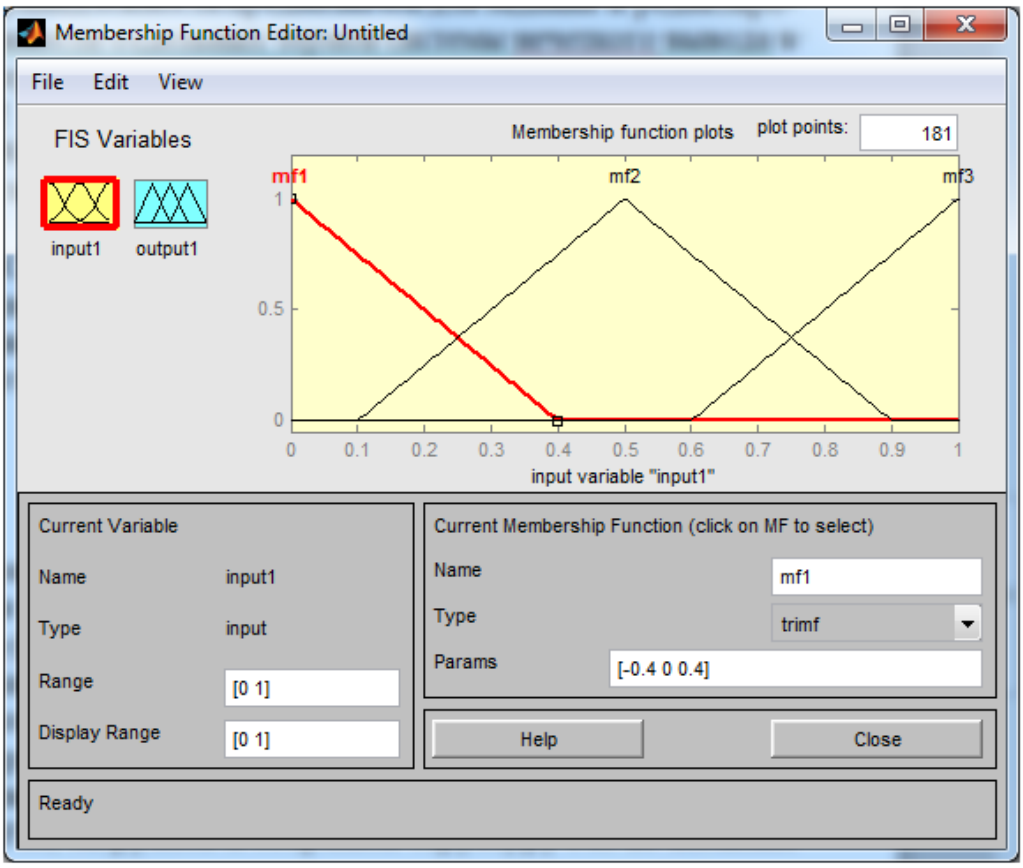

Рисунок 3.2 – Вікно редактору функцій належності

Редактор функцій належності має головне меню програми, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т. д.

3.1.3 Редактор правил системи нечіткого виведення

Редактор правил системи нечіткого виведення (рис.3.3) призначений для завдання і редагування окремих правил системи нечіткого виведення в графічному режимі. Редактор правил може бути відкритий за допомогою головного меню редактора FIS командою меню Edit → Rules ... або натисканням клавіш Ctrl + 3.

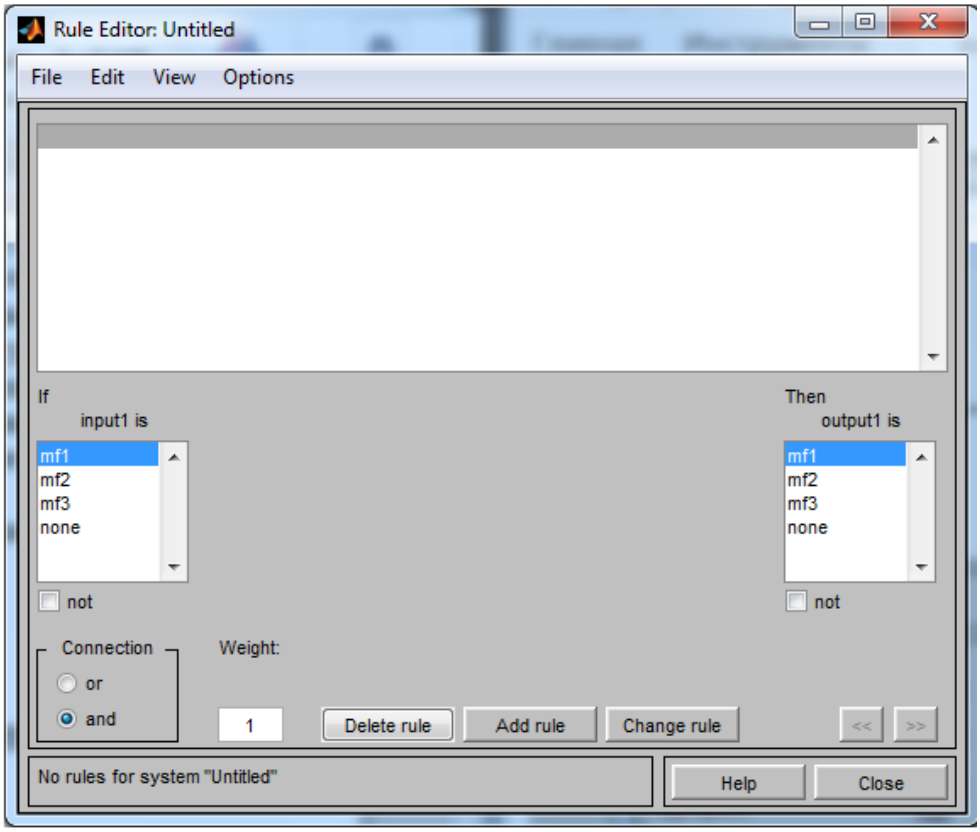

Рисунок 3.3 – Вікно редактору правил

Щоб використовувати цей редактор для створення правил, необхідно попередньо визначити все вхідні і вихідні змінні, для чого можна скористатися редактором системи нечіткого виведення FIS і редактором функцій належності. При цьому задати правила можна за допомогою вибору відповідних значень термів вхідних і вихідних змінних.

Поля введення в середній частині графічного інтерфейсу редактора правил дозволяють задати нове правило в системі нечіткого виведення. Для цього необхідно виділити ім'я терма відповідної змінної, яка повинна бути попередньо визначена за допомогою редактора функцій належності. Якщо деякий терм не входить в правило, то для нього слід вибрати значення none. Якщо в умові правила використовується логічне заперечення деякого терма, то для цього терма слід зазначити відповідний прапорець з міткою not.

Редактор правил дозволяє також задати логічні зв'язки для підумови правила (перемикач Connection) і вага правила (поле введення Weight). Кноп-

ки в нижній частині графічного інтерфейсу редактора правил, як випливає з їх назв, служать для видалення виділеного у вікні правила (Delete rule), додавання створеного правила в систему (Add rule) і внесення змін в виділене в вікні правило (Change rule). У правому нижньому кутку знаходяться кнопки виклику вбудованої довідкової системи MATLAB (Help) і кнопка закриття редактора правил (Close).

3.1.4 Програма перегляду поверхні системи нечіткого виведення

Програма перегляду поверхні системи нечіткого виведення (рис.3.4) дозволяє переглядати поверхню системи нечіткого виведення і візуалізувати графіки залежності вихідних змінних від окремих вхідних змінних. Графічний інтерфейс програми перегляду правил може бути відкритий за допомогою головного меню редактора FIS, редактора функцій належності або редактора правил командою меню View → Surface або натисканням клавіш Ctrl + 6.

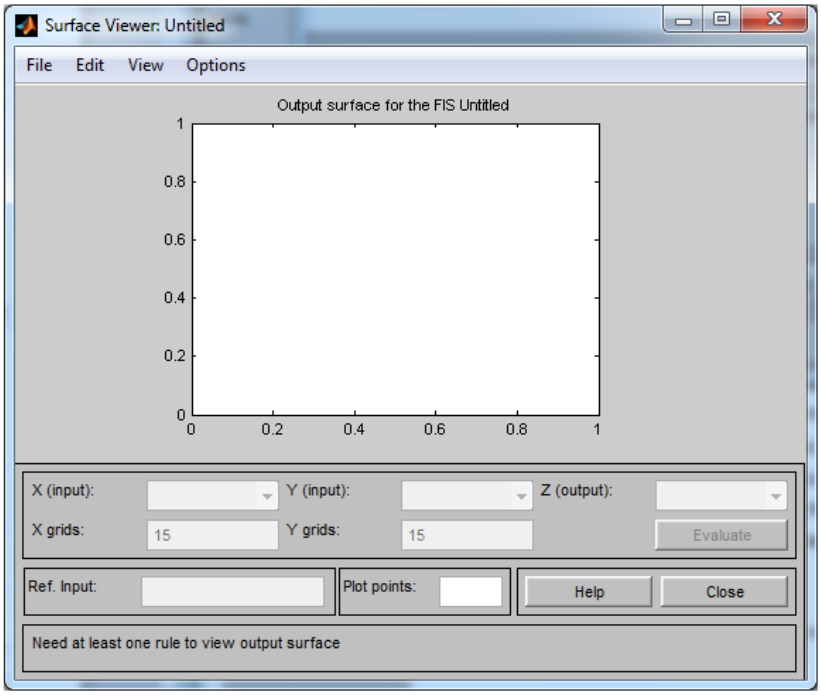

Рисунок 3.4 – Вікно програми перегляду поверхні

Програма перегляду поверхні виведення має головне меню, яке дозволяє користувачеві викликати інші графічні засоби роботи з системою нечіткого виведення FIS, завантажувати і зберігати структуру FIS в зовнішніх файлах і т.д.

Програма перегляду поверхні виведення не дозволяє вносити зміни в систему нечіткого виведення і відповідну їй структуру FIS. Використовуючи головне меню програми, користувач може вибрати вхідні змінні і відповідні їм горизонтальні осі системи координат (X і Y), а також вихідну змінну, якій відповідає вертикальна вісь системи координат (Z).

Натиснувши і утримуючи ліву кнопку миші на осях графіка поверхні, за допомогою подальшого переміщення курсора миші в тому чи іншому напрямку можна змінити кут перегляду поверхні виведення. Якщо розглядається система нечіткого виведення з більш ніж двома вхідними змінними, то для невізуалізіруемих вхідних змінних слід задати деякі постійні значення (константи).

3.2 Моделювання роботи світлофора з нечіткою логікою

3.2.1 Постановка завдання

Виконаємо розробку нечіткої системи роботи світлофора, встановленого на перехресті, при різних інтенсивностях потоків автомашин.

Для кількісної та якісної характеристики роботи світлофора існують поняття такту, фази і циклу регулювання. Тактом регулювання називається період дії певної комбінації світлофорних сигналів. Такти бувають основні і проміжні. В період основного такту дозволено, а в конфлікті напрямку заборонено, рух певної групи транспортних і пішохідних потоків. Під час проміжного такту виїзд на перехрестя заборонений за винятком ТЗ, водії яких не змогли своєчасно зупинитися у стоп-лінії. Відбувається підготовка перехрестя до переходу права на рух наступної групи потоків. Перехрестя повино бути звільнено від транспортних засобів і пішоходів, які мали право на рух під

час попереднього такту. Проміжний такт служить для безпеки руху в перехідний період, коли рух попередньої групи потоків вже заборонено, а подальша група дозвіл на рух ще не отримала. Фаза регулювання – сукупність основного та наступного за ним проміжного такту. Зазвичай число фаз регулювання відповідає числу найбільш завантажених конфліктних напрямків руху на перехресті. Цикл регулювання – сукупність всіх фаз, що періодично повторюється. Під режимом світлофорного регулювання розуміється тривалість циклу, а також число, порядок чергування і тривалість складових цикл тактів і фаз [15][1)].

В аналітичному вигляді режим світлофорного регулювання можна представити у вигляді виразу :

$$T_ц=t_{О1}+t_{П1}+ t_{О2}+t_{П2}+\dots +t_{Оn}+t_{Пn} \quad (3.1)$$

де $T_ц$ – цикл регулювання (сек);

$t_{О1}$, $t_{О2}$, $t_{Оn}$ – тривалість основних тактів (сек);

$t_{П1}$, $t_{П2}$, $t_{Пn}$ – тривалість проміжних тактів (сек);

n – число фаз регулювання.

Зазвичай проміжний позначається жовтим сигналом у напрямку де раніше (під час основного такту) здійснюється рух. Тривалість жовтого сигналу $t_ж$ не повинна бути менше 3 сек. З іншого боку з позиції безпеки руху, його тривалість не повинна бути більше 4 сек.

У звичайному режимі роботи світлофора час роботи зеленого і червоного світла, а також час циклу фіксовані. Для нашої моделі час циклу звичайного світлофору буде дорівнювати 64 сек., де фаза зеленого складає $t_О+t_П$ =30 сек + 4 сек, фаза червоного – 27 сек + 3 сек. Це створює певні труднощі

[1)] [15] Структура светофорного цикла. URL:https://studopedia.ru/2_100793_ struktura-svetofornogo-tsikla.html (дата звернення 13.05.2019)

в русі машин, особливо, при зміні їх потоків в години пік, що досить часто призводить до появи автомобільних пробок.

У пропонованому нечіткому світлофорі час циклу залишається постійним, проте, час його роботи в режимі зеленого світла повинно змінюватися в залежності від кількості машин, що під'їжджають до перехрестя.

Для роботи нечіткого світлофора на перехресті вулиць Осипова та Успенської необхідно встановити 8 детекторів (рис.3.5), які рахують машини, що проїхали повз них.

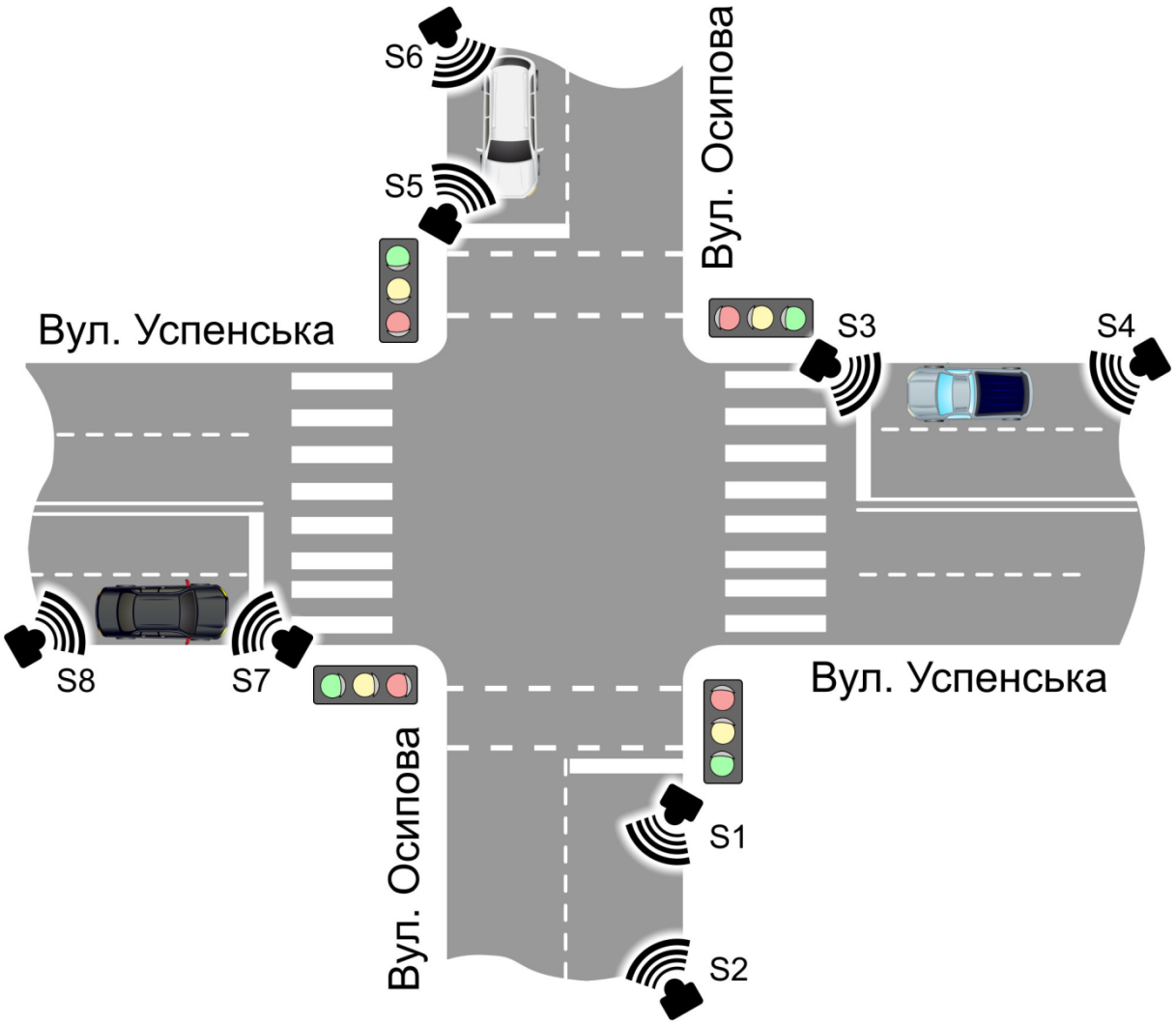

Рисунок 3.5 – Розташування детекторів на перехресті

На перехресті розташовуються два типи детекторів:

– детектори, що розташовують на під'їзді до перехрестя, вони рахують автомобілі в момент проїзду через них,

– детектори, які необхідні для обліку автомобілів, які покидають перехрестя, і розташовуються безпосередньо на кордоні перехрестя.

Світлофор використовує різниці показань чотирьох пар детекторів: (S1 – S2), (S3 – S4), (S5 – S6) і (S7 – S8). Таким чином, якщо для вул. Осипова горить зелене світло, машини проїжджають перехрестя і показання двох пар датчиків рівні: S1 = S2, S5 = S6, а, отже, їх різниця дорівнює нулю. В цей же час на вул. Успенська перед світлофором зупиняються машини, які встигли проїхати тільки S4 і S8.

В результаті можна розрахувати сумарну кількість автомобілів на вул. Успенська:

$$(S4 – S3) + (S8 – S7) = (S4 – 0) + (S8 – 0) = S4 + S8.$$

Для порівняння роботи обох світлофорів введемо показник ефективності, в якості якого будемо розглядати число машин, що не проїхали перехрестя за один цикл світлофора.

3.2.2 Моделювання системи

Для вирішення поставленого завдання використовується пакет Matlab, тому що він має в своєму складі fuzzy-додаток, необхідне для моделювання роботи нечіткого світлофора.

При проектуванні нечіткої підпрограми повинні бути визначені всі входи і виходи.

Оскільки робота світлофора залежить від числа машин на обох вулицях і поточного часу зеленого світла, для нашої підпрограми пропонується використовувати 3 входу: число машин на вул. Успенська по закінченню чергово-

го циклу, число машин на вул. Осипова по закінченню циклу і час зеленого світла нечіткого світлофора.

Тепер для кожної змінної треба задати лінгвістичні терми, відповідні деяким діапазонами чітких значень. Так, для змінної час зеленого світла пропонується використовувати три терма (рис.3.6):

– мале (10-25сек.);

– середнє (20-40сек.);

– велике (35-50сек.).

Ступінь належності чітких значень термам задається за допомогою функцій належності (в нашому випадку ці функції мають форму трапеції).

Аналогічно, терми для двох змінних, що залишилися, будуть наступні (рис.3.7):

– дуже мале (0-18);

– мале (16-36);

– середнє (34-56);

– велике (54-76);

– дуже велике (72-90).

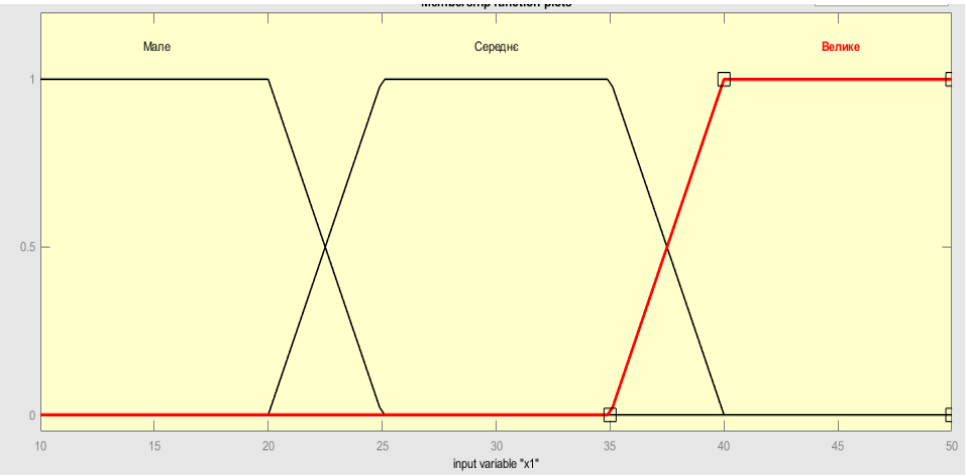

Рисунок 3.6 – Функція належності першої вхідної змінної

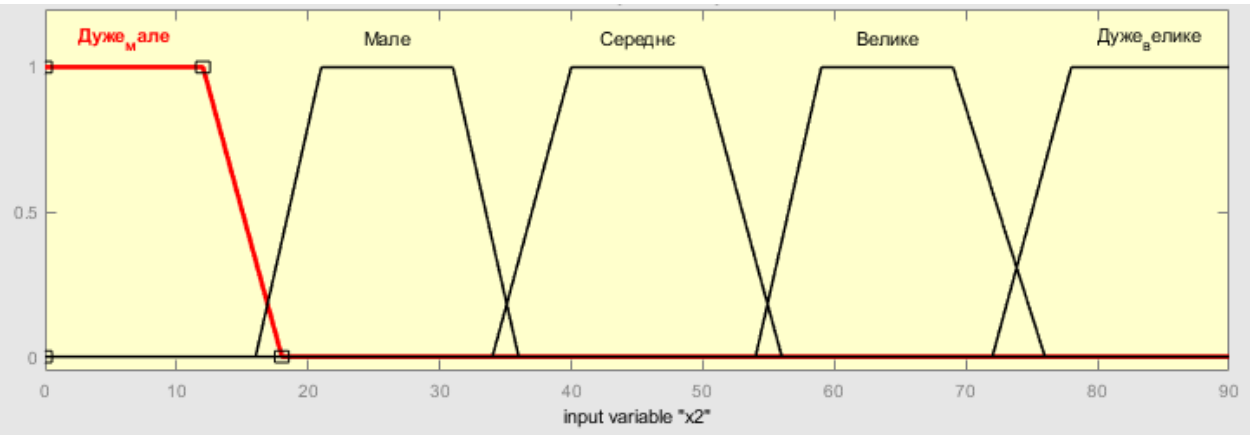

Рисунок 3.7 – Функція належності другої та третьої вхідних змінних

Функції приналежності тут також мають форму трапеції.

Так як суть роботи світлофора полягає в зміні часу зеленого світла, в якості вихідного параметра пропонується використовувати величину цієї зміни. Терми в цьому випадку будуть наступні (рис.3.8):

– зменшити (-20-0сек.);

– не змінювати (-15-15сек.);

– збільшити (0-20сек.).

Функції приналежності мають форму Гаусса.

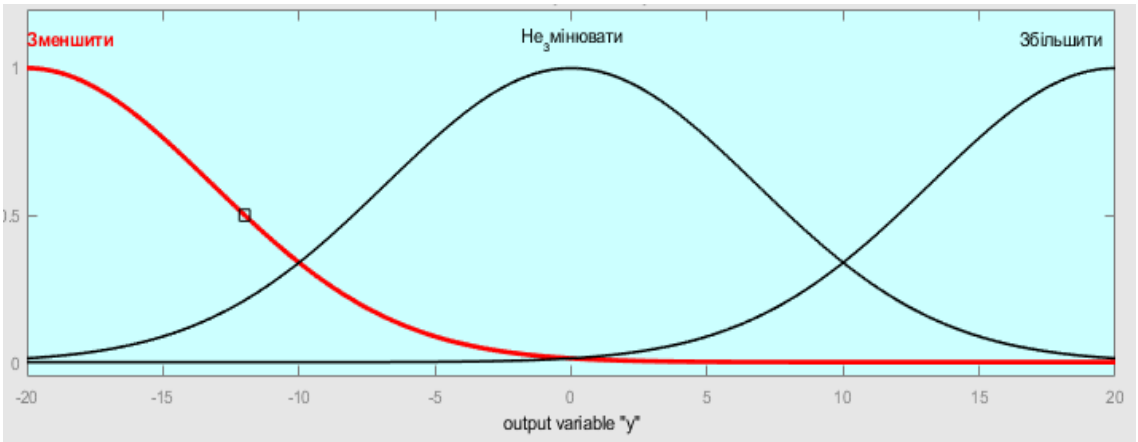

Рисунок 3.8 – Функція належності вихідної змінної

Крім того, в підпрограму записується таблиця правил на основі умовних висловлювань, яка формує вихідне значення виходячи з величин вхідних параметрів, наприклад: Якщо (число машин на вул. Успенська = мале) і (число машин на вул. Осипова = велике) і (час зеленого світла на вул. Успенська = велике), то (час зеленого світла = зменшити). Таблиця правил наведена на рис.3.9. На рис. 3.10 наведено вікно візуалізації нечіткого логічного виведення.

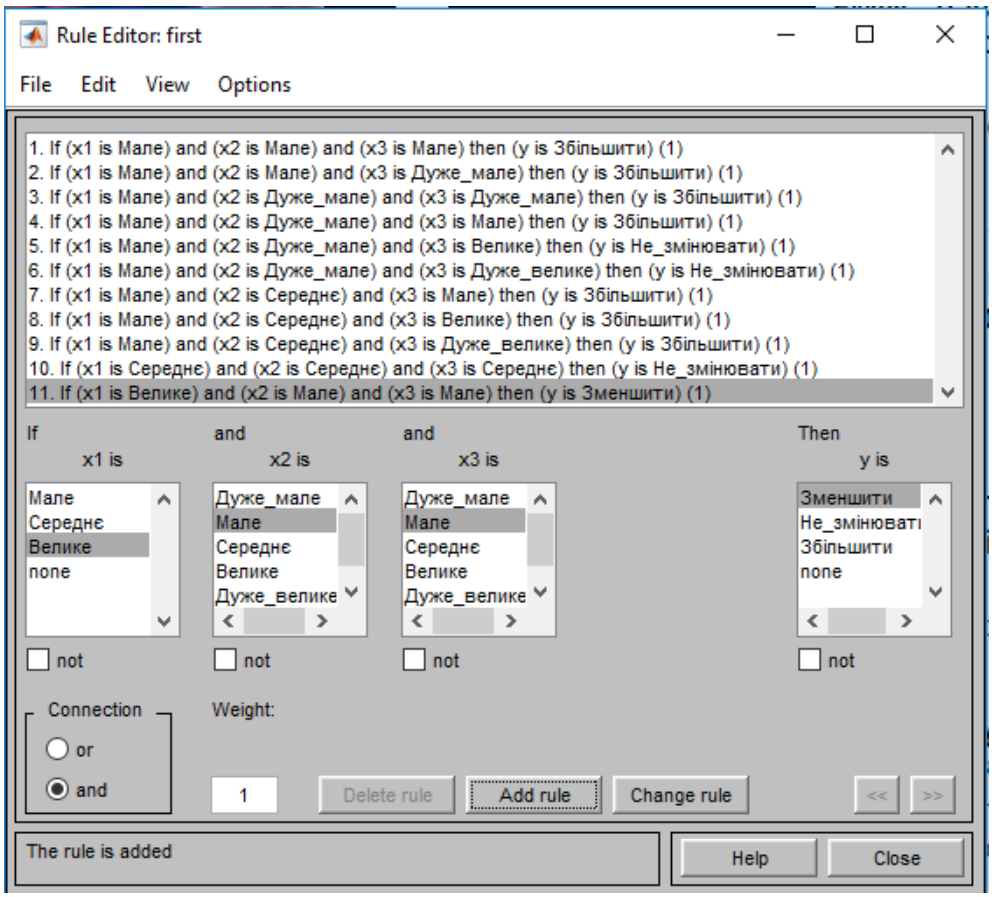

Рисунок 3.9 – Правила виведення для нечіткої системи

3.2.3 Результати моделювання

Основна програма працює наступним чином: за допомогою вбудованого в Matlab генератора випадкових чисел відбувається генерування числа машин за один цикл світлофора для вулиць Успенська і Осипова.

Частина машин з цього числа встигає проїхати на зелене світло, інші зупиняються перед перехрестям, чекаючи закінчення дії червоного світла світлофора. Всі ті автомобілі, які залишилися стояти перед світлофором після одного циклу, вважаються такими, що не обслуговані заявками.

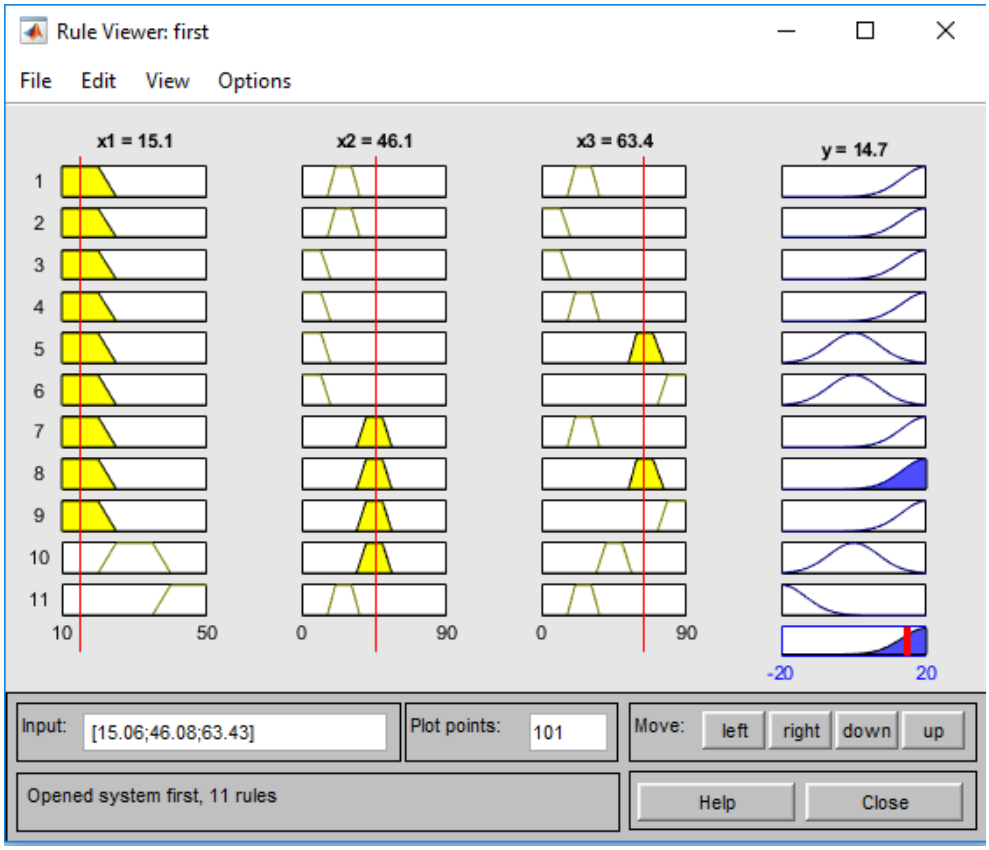

Рисунок 3.10 – Вікно візуалізації нечіткого логічного виведення

За показник ефективності даної системи приймається середнє число не обслугованих заявок за задану кількість циклів світлофора. Відповідно, чим менше значення має показник ефективності, тим більша кількість машин пропускає світлофор.

Число циклів світлофора не повинно бути занадто малим, тому що в цьому випадку не виходить об'єктивної інформації, або занадто великим, тому що програма буде дуже довго обчислювати необхідну величину. Рекомендована кількість циклів – 100.

Таким чином, алгоритм програми наступний: на світлофор з детекторів надходить інформація про кількість автомобілів на двох вулицях. Ці дані переводяться в нечіткий формат згідно із заданими функціями належності, далі, всередині підпрограми відбувається їх обробка, отримане значення зміни часу зеленого світла дефазифікується (тобто переводиться назад в чіткий формат) і надходить у вигляді керуючого сигналу на світлофор. Відповідно до цього сигналу час зеленого світла світлофора в наступному циклі буде іншим.

Результат обчислень представлений у вигляді графіка рис.3.9. На цьому графіку показано кількість машин, які не встигли проїхати перехрестя за певну кількість циклів при заданих умовах. Синім зображено кількість відмов при роботі звичайного світлофора, рожевим – при роботі нечіткого світлофора. Програма моделювання в MatLab наведена у додатку А.

Таким чином, що нечітка логіка, в деяких найпростіших випадках, дозволяє поліпшити якість керування об'єктами. Вирішальну роль в оптимізації показників ефективності грають експерти, які визначають кількість вхідних і вихідних змінних, число термів для кожної змінної, види функцій належності, тому що зміна цих параметрів призводить до поліпшення або погіршення процесу керування об'єктом.

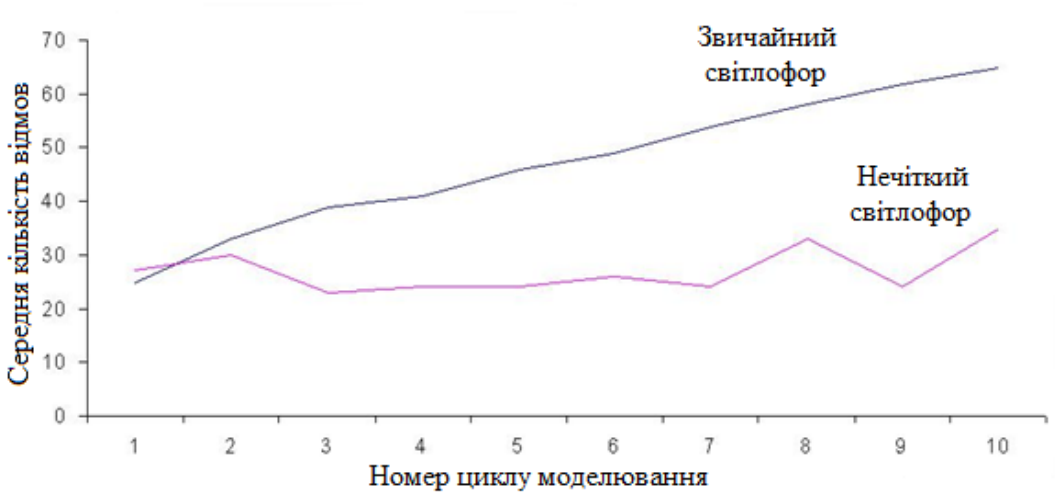

Рисунок 3.9 – Результати моделювання

Найважливішим недоліком нечіткої логіки є відсутність єдиного методу моделювання систем, тобто для кожного випадку доводиться заново проектувати нечітку підпрограму, визначаючи крок за кроком всі параметри і будуючи свою таблицю рішень.

ВИСНОВКИ

В результаті виконання даної роботи була розроблена експертна система нечіткого виведення за алгоритмом Мамдані з використанням основних функцій пакету Fuzzy Logic Toolbox програмного середовища MatLab для керування світлофором на перехресті.

В результаті проведеного дослідження також були вирішені такі завдання:

- розглянуті основні поняття теоретичні відомості, що стосуються питань керування об’єктами, які важко формалізувати, на основі нечіткої логіки;
- виконаний аналіз методів побудови бази знань та різних механізмів логічного виведення;
- досліджено можливості середовища MatLab по створенню і редагуванню систем керування нечіткими об'єктами;
- виявлено нові закономірності роботи світлофора з нечітким керуванням;
- розроблено додаток MatLab, яка реалізує роботу нечіткого світлофора, виконано тестування та аналіз отриманих результатів для різних циклів роботи звичайного та нечіткого світлофорів.

Слід відзначити, що нечітка логіка, в деяких найпростіших випадках, дозволяє поліпшити якість керування об'єктами. Але значну роль в оптимізації показників ефективності грає досвід та вміння експертів, які визначають кількість вхідних і вихідних змінних, число термів для кожної змінної, види функцій належності. Також недоліком нечіткої логіки є відсутність єдиного методу моделювання систем, тобто для кожного випадку доводиться заново проектувати нечітку підпрограму, визначаючи крок за кроком всі параметри і будуючи свою таблицю рішень.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Заде Л.А. Понятие лингвистической переменной и его применение к принятию приближенных решений. М.:Мир, 1976. 165 с.
2. Беллман Р., Заде Л. Принятие решений в расплывчатых условиях. Вопросы анализа и процедуры принятия решений. М: Мир, 1976. С.172 – 215.
3. Кофман А. Введение в теорию нечетких множеств. М.: Радио и связь, 1982. 432 с.
4. Нечеткие множества в моделях управления и искусственного интеллекта. Под ред. Д.А. Поспелова. М.: Наука, 1986. 312 с.
5. Нечеткие множества и теория возможностей. Под ред. Р.Р. Ягера. М.: Радио и связь, 1986. 408 с.
6. Макеев А.В. Основы нечеткой логики. Учебное пособие для вузов. Н.Новгород:ВГИПУ, 2009. 59с.
7. Субботін С. О. Подання й обробка знань у системах штучного інтелекту та підтримки прийняття рішень. Навчальний посібник. Запоріжжя: ЗНТУ, 2008. 341 с.
8. Сявавко М., Рибицька О. Математичне моделювання за умов невизначеності. Львів: Українські технології, 2000. 320 с.
9. Борисов А.Н., Крумберг О.А., Федоров И.П. Принятие решений на основе нечетких моделей. Примеры использования. Рига:Зинатне, 1990. 184 с.
10. Рутковская Д., Пилиньский М., Рутковский Л. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Горячая линия Телеком, 2007. 452 с.
11. Асаи К., Ватада Д., Иваи С. Прикладные нечеткие системы. М.: Мир, 1993. 368 с.
12. Леоненков А. В. Нечеткое моделирование в MATLAB и fuzzyTECH. СПб.: БХВ-Петербург, 2005. 736 с.

13. Штовба С.Д. Проектирование нечетких систем средствами MATLAB. М.: Горячая линия Телеком, 2007. 288 с.
14. Дьяконов В. MATLAB 6: учебный курс. СПб.: Питер, 2001. 592 с.
15. Структура светофорного цикла. URL:https://studopedia.ru/2_100793_struktura-svetofornogo-tsikla.html (дата звернення 13.05.2019)

ДОДАТОК А

Програмний код моделі в MATLAB