

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені В.Н.КАРАЗІНА

ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ І НЕЧІТКОГО МОДЕЛЮВАННЯ В СИСТЕМАХ УПРАВЛІННЯ

У двох частинах

Частина 2

Навчально-методичний посібник
для здобувачів вищої освіти другого (магістерського) рівня
за спеціальністю 174 «Автоматизація, комп'ютерно-
інтегровані технології та робототехніка»

Електронний ресурс

Харків – 2025

Рецензенти:

Р. М. ТРИЩ – доктор технічних наук, професор, завідувач кафедри мехатроніки та електротехніки Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут»;

О. О. Литвин – доктор фізико-математичних наук, професор, завідувач кафедри ХТЛПід Навчально-наукового інституту «Українська інженерно-педагогічна академія».

*Затверджено до розміщення в мережі Інтернет рішенням Науково-методичної ради
Харківського національного університету імені В. Н. Каразіна
(протокол № 8 від 28 березня 2025 року)*

Т 38

Технології нейронних мереж і нечіткого моделювання в системах управління. У 2-х частинах. Ч. 2 : навчально-методичний посібник для здобувачів вищої освіти другого (магістерського) рівня за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка [Електронний ресурс] / уклад. Г. І. Канюк, Т. Ю. Василець О. О. Варфоломійєв. – Харків : ХНУ імені В. Н. Каразіна, 2025. – (PDF 107 с.)

У другій частині навчально-методичного посібника викладено апарат нечіткої логіки і гібридних нейронних мереж. Розглянуто основні поняття нечіткої логіки, операції над нечіткими множинами і відносинами, алгоритми нечітких висновків, алгоритми навчання і використання гібридних мереж.

Значна увага приділена використанню систем нечіткого висновку в задачах управління. Детально розглянуті методи синтезу нечітких систем управління з застосуванням системи MATLAB. Наведено приклад, що ілюструє ефективність застосування нечітких регуляторів в системах управління, і приклад розробки нечіткої моделі гібридної мережі для вирішення завдання ідентифікації двомасової електромеханічної системи.

Видання призначене здобувачам вищої освіти освітнього ступеня «магістр», що навчаються за освітньо-професійною програмою «Автоматизація та комп'ютерно-інтегровані технології» зі спеціальності 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

УДК 658.5.011.56 : 658.589 (075.8)

© Харківський національний університет
імені В. Н. Каразіна, 2025

© Канюк Г. І., Василець Т. Ю.,
Варфоломійєв О. О., уклад., 2025

ЗМІСТ

ГЛАВА 5 ОСНОВИ ТЕОРІЇ НЕЧІТКИХ МНОЖИНИ І НЕЧІТКОЇ ЛОГІКИ.....	5
5.1 Ефективність систем прийняття рішень,що використовують методи нечіткої логіки	6
5.2 Основні поняття нечіткої логіки.....	7
5.2.1 Визначення нечіткої множини.....	7
5.2.2 Характеристики нечітких множин	8
5.2.3 Функції приналежності нечіткої логіки	10
5.2.4 Нечіткість і вірогідність	11
5.3 Операції над нечіткими множинами і відносинами	12
5.3.1 Логічні операції	12
5.3.2 Алгебраїчні операції з нечіткими множинами.....	16
5.3.3 Нечіткі відносини	19
5.3.4 Операції над нечіткими відносинами.....	20
5.3.5 Нечітка імплікація	23
5.4 Нечіткі і лінгвістичні змінні. Нечіткі числа	25
5.4.1 Визначення нечіткої і лінгвістичної змінних	25
5.4.2 Нечіткі числа.....	27
5.4.3 Операції над нечіткими числами	28
5.4.4 Нечіткі числа (L – R)-типа	28
5.5 Нечіткі висновки	31
5.5.1 Основні етапи нечітких висновків.....	31
5.5.2 Алгоритми нечіткого висновку.....	34
5.5.3 Методи приведення до чіткості	41
5.6 Використання систем нечіткого висновку в задачах управління	42
Питання для самоконтролю.....	45
ГЛАВА 6 ГІБРИДНІ МЕРЕЖІ.....	46
6.1 Основні поняття і визначення гібридних мереж.....	47
6.2 Алгоритми навчання і використання гібридних мереж	49
6.3 Нечіткий гібридний класифікатор.....	56
Питання для самоконтролю.....	57
ГЛАВА 7 СИНТЕЗ НЕЧІТКИХ РЕГУЛЯТОРІВ З ВИКОРИСТАННЯМ СИСТЕМИ MATLAB.....	58
7.1 Призначення і можливості пакету Fuzzy Logic Toolbox	58
7.2 Графічний інтерфейс Fuzzy Logic Toolbox	58
7.3 Розробка моделі системи управління з нечітким регулятором	59
7.4 Формування бази правил системи нечіткого висновку	64

7.5 Розробка системи нечіткого висновку	66
7.6 Моделювання двомасової електромеханічної системи з нечітким регулятором	76
Питання для самоконтролю.....	80
ГЛАВА 8 ПОБУДОВА МОДЕЛЕЙ ДИНАМІЧНИХ ОБ'ЄКТІВ	
З ВИКОРИСТАННЯМ ГІБРИДНИХ МЕРЕЖ.....	81
8.1 Реалізація ANFIS в середовищі MATLAB	81
8.2 Розробка нечіткої моделі гібридної мережі для вирішення завдання ідентифікації двомасової електромеханічної системи.....	84
Питання для самоконтролю.....	101
СПИСОК ЛІТЕРАТУРИ.....	102

ОСНОВИ ТЕОРІЇ НЕЧІТКИХ МНОЖИН І НЕЧІТКОЇ ЛОГІКИ

Однією з найбільш вражаючих властивостей людського інтелекту є здатність ухвалювати правильні рішення в обстановці неповної і нечіткої інформації. Побудова моделей наближених міркувань людини і використання їх в комп'ютерних системах майбутніх поколінь представляє сьогодні одну з найважливіших проблем науки.

Значне просування в цьому напрямі зроблене професором Каліфорнійського університету (Берклі) Лотфі А. Заде (Lotfi A. Zadeh). Його робота «Fuzzy Sets», що з'явилася в 1965 р. в журналі *Information and Control* №8 [1], заклала основи моделювання інтелектуальної діяльності людини і явилася початковим поштовхом до розвитку нової математичної теорії.

Л. Заде розширив класичне канторовське поняття множини, допустивши, що характеристична функція (функція приналежності елементу множини) може приймати будь-які значення в інтервалі $[0; 1]$, а не тільки значення 0 або 1. Такі множини були названі ним нечіткими (fuzzy). Він визначив також ряд операцій над нечіткими множинами і запропонував узагальнення відомих методів логічного виведення *modus ponens* і *modus tollens*.

Ввівши потім поняття лінгвістичної змінної і допустивши, що в якості її значень (термів) виступають нечіткі множини, Л. Заде створив апарат для опису процесів інтелектуальної діяльності, включаючи нечіткість і невизначеність виразів.

Подальші роботи професора Л. Заде і його послідовників заклали міцний фундамент нової теорії і створили передумови для впровадження методів нечіткого управління в інженерну практику. Ці питання висвітлюються в багатьох роботах, наприклад [2-60].

Діапазон застосування нечітких методів з кожним роком розширюється, охоплюючи такі області, як проектування промислових роботів і побутових електроприладів, управління доменними печами і рухом потягів метро, автоматичне розпізнавання мови і зображень. При цьому нечіткі системи дозволяють підвищити якість продукції при зменшенні ресурсо- і енерговитрат і забезпечують вищу стійкість до дії заважаючих факторів в порівнянні з традиційними системами автоматичного управління.

Нові підходи дозволяють розширити методи синтезу систем автоматичного управління за межі застосування класичної теорії. У цьому плані цікава точка зору Л. Заде: «Я вважаю, що зайве прагнення до точності стало надавати дію, що зводить нанівець теорію управління і теорію систем, оскільки воно призводить до того, що дослідження в цій області зосереджуються на тих і лише тих проблемах, які піддаються точному рішення. В результаті багато класів важливих проблем, в яких дані, цілі і обмеження є дуже складними або погано визначеними для того, щоб допусти-

ти точний математичний аналіз, залишалися і залишаються осторонь з тієї причини, що вони не піддаються математичному трактуванню. Для того, щоб сказати щонебудь істотне для проблем подібного роду, ми повинні відмовитися від наших вимог точності і допустити результати, які є декілька розмитими або невизначеними».

Зсув центру досліджень нечітких систем у бік практичних застосувань привів до постановки цілого ряду проблем, таких як нова архітектура комп'ютерів для нечітких обчислень, елементна база нечітких комп'ютерів і контролерів, інструментальні засоби розробки, інженерні методи розрахунку і розробки нечітких систем управління і багато що інше.

Математична теорія нечітких множин дозволяє описувати нечіткі поняття і знання, оперувати цими знаннями і робити нечіткі висновки. Нечітке управління виявляється особливо корисним, коли технологічні процеси є дуже складними для аналізу за допомогою загальноприйнятих кількісних методів або коли доступні джерела інформації інтерпретуються якісно, неточно або невизначено. Нечітка логіка, на якій засновано нечітке управління, ближче по духу до людського мислення і природних мов, ніж традиційні логічні системи. Нечітка логіка в основному забезпечує ефективні засоби відображення невизначеностей і неточностей реального миру. Наявність математичних засобів відображення нечіткості початкової інформації дозволяє побудувати модель, адекватну реальності.

5.1 Ефективність систем прийняття рішень, що використовують методи нечіткої логіки

Можливість використання апарату нечіткої логіки базується на наступних результатах.

1. У 1992 р. Wang довів теорему: *для кожної речової безперервної функції $G(x)$, заданої на компактi U , і для довільного $\varepsilon > 0$ існує нечітка експертна система, що формує вихідну функцію $F(x)$ таку, що*

$$\sup_{x \in U} \|G(x) - F(x)\| \leq \varepsilon,$$

де $\| \cdot \|$ – символ прийнятої відстані між функціями.

2. У 1995 р. Castro показав, що логічний контролер Mamdani при певних умовах також є універсальним апроксиматором.
3. Згідно знаменитої теореми FAT (Fuzzy Approximation Theorem), доведеної Б. Коско (B. Kosko) у 1993 році, *будь яка математична система може бути апроксимована системою, заснованою на нечіткій логіці.*

Системи з нечіткою логікою доцільно застосовувати в наступних випадках:

- 1) для складних процесів, коли немає простої математичної моделі;
- 2) якщо експертні знання про об'єкт або про процес можна сформулювати тільки в лінгвістичній формі.

Відзначимо, що поряд з безумовними перевагами, системи з нечіткою логікою мають і деякі *недоліки* пов'язані в основному з тим, що:

- 1) початковий набір постулюємих нечітких правил формулюється експертом-людиною і може виявитися неповним або суперечливим;
- 2) вигляд і параметри функцій приналежності, що описують вхідні і вихідні змінні системи, вибираються суб'єктивно і можуть виявитися такими, що не цілком відображають реальну дійсність.

5.2 Основні поняття нечіткої логіки

5.2.1 Визначення нечіткої множини

Нечітка множина (fuzzy set) – це сукупність елементів довільної природи, щодо яких не можна з повною визначеністю стверджувати: чи належить той або інший елемент даній сукупності даній множині, чи ні. Іншими словами, нечітка множина відрізняється від звичайної множини тим, що для всіх або частини її елементів не існує однозначної відповіді на питання: «Належить або не належить той або інший елемент даній нечіткій множині?» Можна це питання поставити і по-іншому: «Чи володіють його елементи деякою характеристичною властивістю, яка може бути використана для завдання цієї нечіткої множини?»

Формально нечітка множина A визначається як множина впорядкованих пар або кортежів вигляду: $\langle x, \mu_A(x) \rangle$, де x є елементом деякої **універсальної множини** або **універсуму** E , а $\mu_A(x)$ – **функція приналежності** (або ступінь приналежності), яка ставить у відповідність кожному з елементів $x \in E$ деяке дійсне число з інтервалу $[0, 1]$, тобто дана функція визначається у формі відображення

$$\mu_A(x): E \rightarrow [0,1].$$

При цьому $\mu_A(x) = 1$ для деякого $x \in E$ означає, що елемент x безумовно належить нечіткій множині, а значення $\mu_A(x) = 0$ означає, що елемент x безумовно не належить нечіткій множині A .

Функція приналежності може бути визначена явним чином у вигляді функціональної залежності, наприклад

$$\mu_A(x) = \exp\left(-\left(\frac{x-1}{2}\right)^2\right),$$

або дискретно – шляхом завдання кінцевої послідовності значень $x \in (x_i)$ у вигляді

$$A = \{\mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n\},$$

або у вигляді

$$A = \left\{ \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} \right\},$$

при цьому коса і горизонтальна межа служить просто роздільником, а знак «+» позначає не арифметичну суму, а теоретико-множинне об'єднання окремих елементів. Нескінченні нечіткі множини іноді записують із знаком інтеграла у вигляді

$$\int \mu_A(x) / x.$$

Все це скоріше дань традиції, ніж щось, що має змістовний сенс. Тим паче, що сам знак інтеграла може бути сприйнятий як нечіткий інтеграл, чим він тут ніяк не є: Бажаючи підкреслити або явно вказати, що множина A є нечіткою, іноді часто записують нечітку множину із знаком тильда « $\sim$ » вгорі, тобто у формі $\tilde{A}$.

Приклад завдання нечіткої множини. Нехай $E = \{x_1, x_2, x_3, x_4, x_5\}$; A – нечітка множина, для якої $\mu_A(x_1) = 0,3$, $\mu_A(x_2) = 0$, $\mu_A(x_3) = 1$, $\mu_A(x_4) = 0,5$, $\mu_A(x_5) = 0,9$.

Тоді A можна представити, наприклад, у вигляді

$$A = 0,3/x_1 + 0/x_2 + 1/x_3 + 0,5/x_4 + 0,9/x_5,$$

або у вигляді:

	x_1	x_2	x_3	x_4	x_5
$A =$	0,3	0	1	0,5	0,9

5.2.2 Характеристики нечітких множин

Нехай A – нечітка множина з елементами з універсальної множини E , що приймає свої значення на відрізку $[0, 1]$.

- Величина $\sup_{x \in E} \mu_A(x)$ називається *висотою* нечіткої множини A . Нечітка множина A *нормальна*, якщо її висота рівна 1 ($\sup_{x \in E} \mu_A(x) = 1$). При $\sup_{x \in E} \mu_A(x) < 1$ нечітка множина називається *субнормальною*.

- Нечітка множина *порожня*, якщо $\forall x \in E \mu_A(x) = 0$, тобто це множина, яка не містить жодного елементу (позначається через $\emptyset$). Формально порожня множина визначається як така нечітка множина, функція приналежності якої тотожно рівна нулю для всіх без виключення елементів. Непорожню субнормальну множину можна нормалізувати по формулі

$$\mu_A(x) := \frac{\mu_A(x)}{\sup_{x \in E} \mu_A(x)}.$$

- Нечітка множина *унімодальна*, якщо $\mu_A(x) = 1$ тільки на одному x з E .
- Кожна нечітка множина має певний *носіє* (англ.: support). Носієм множини $Supp(A)$ є підмножина тих елементів A , для яких ступінь приналежності до A не рівна нулю, тобто $Supp(A) = \{x, \mu_A(x) > 0\}$.
- Елементи $x \in E$, для яких $\mu_A(x) = 0,5$, називаються *точками переходу* множини A .
- Так званий *універсум*, що позначається через E , вже був використаний вище як звичайна множина, що містить в рамках деякого контексту всі можливі елементи. Формально зручно вважати, що функція приналежності універсуму як нечіткої множини тотожно рівна одиниці для всіх без виключення елементів: $\mu_E = 1$.

Приклади нечітких множин.

1. Нехай $E = \{0, 1, 2, \dots, 10\}$. Нечітку множину «декілька» можна визначити таким чином:

$$\text{«декілька»} = 0,5/3 + 0,8/4 + 1/5 + 1/6 + 0,8/7 + 0,5/8.$$

Її характеристики: *висота*=1; *носіє*= $\{3, 4, 5, 6, 7, 8\}$; *точки переходу* – $\{3, 8\}$

2. Нехай $E = \{1, 2, 3, \dots, n, \dots\}$. Нечітку множину «малий» можна визначити за допомогою функції приналежності

$$\mu_{\text{«малий»}}(n) = \frac{1}{1 + \left(\frac{n}{10}\right)^2}.$$

3. Нехай $E = \{1, 2, 3, \dots, 100\}$ і відповідає поняттю «вік»; тоді нечітка множина «молодий» може бути визначена за допомогою функції приналежності

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

Носієм множини «молодий» є множина років в інтервалі від 1 до 100.

4. Нехай E – множина цілих чисел: $E = \{-8, -5, -3, 0, 1, 2, 4, 6, 9\}$; тоді нечітку підмножину чисел, що по абсолютній величині близькі до нуля, можна визначити, наприклад, так:

$$A = \{0/-8 + 0,5/-5 + 0,6/-3 + 1/0 + 0,9/1 + 0,8/2 + 0,6/4 + 0,3/6 + 0/9\}.$$

5.2.3 Функції приналежності нечіткої логіки

Вищенаведене формальне визначення нечіткої множини не накладає ніяких обмежень на вибір конкретної функції приналежності для її представлення. Проте на практиці зручно використовувати ті з них, які допускають аналітичне представлення у вигляді деякої простої математичної функції. Це спрощує не тільки відповідні чисельні розрахунки, але і скорочує обчислювальні ресурси, необхідні для зберігання окремих значень цих функцій приналежності. Необхідність типізації окремих функцій приналежності також обумовлена наявністю реалізацій відповідних функцій у відомих інструментальних засобах (наприклад, в системі MATLAB).

Найчастіше в нечіткій логіці для завдання функцій приналежності використовуються наступні типові форми приведених нижче функцій (рис. 5.1): трикутна (trimf); трапецеїдальна (trapmf); гаусова (gaussmf); подвійна гаусова (gauss2mf); узагальнена коло колоподібна (gbellmf); сигмоїдальна (sigmf); подвійна сигмоїдальна (dsigmf); добуток двох сигмоїдальних функцій (psigmf); Z-функція (zmf); *Pi*-функція (pimf); *S*-функція (smf).

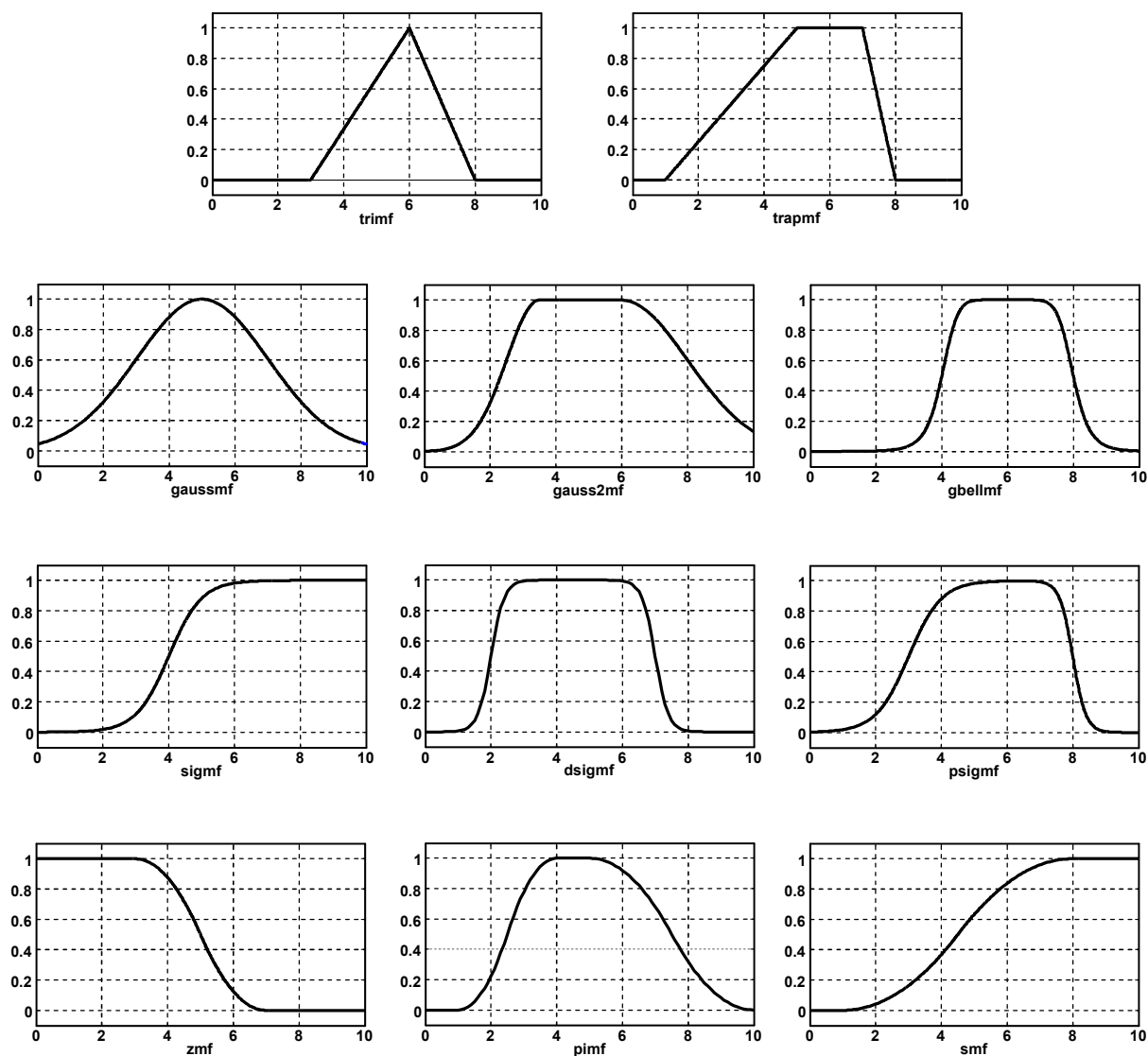


Рис. 5.1. Типові функції приналежності нечітких множин

Конкретний вид даних функцій визначається значеннями параметрів, що входять в їх аналітичні представлення, наприклад:

$$\text{trimf}(x, a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{x-a}{b-a}\right), 0\right);$$

$$\text{trapmf}(x, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{d-a}{c-b}\right), 0\right);$$

$$\text{gaussmf}(x, \sigma, c) = e^{-\left(\frac{x-c}{\sigma}\right)^2};$$

$$\text{bellmf}(x, a, b, c) = \frac{1}{1 + \left|\frac{x-c}{a}\right|^{2b}};$$

$$\text{sigmf}(x, a, c) = \frac{1}{1 + \exp(-a(x-c))} \text{ і т.д.}$$

5.2.4 Нечіткість і вірогідність

Добре відомий той факт, що стохастична (імовірнісна) і лінгвістична невизначеності мають різний характер. *Стохастична невизначеність* має справу з невизначеністю того, чи відбудеться деяка добре описана подія в майбутньому, а теорія вірогідності дозволяє дати на це питання ту або іншу відповідь.

Лінгвістична невизначеність пов'язана з неточністю опису самої ситуації або події незалежно від часу їх розгляду. Теорія вірогідності не може використовуватися для вирішення подібних проблем, оскільки уявлення про суб'єктивні категорії, присутні в процесах мислення людини, повною мірою не узгоджуються з її аксіомами.

Проте, деякі з фахівців, що інтенсивно працюють з теорією вірогідності і математичною статистикою, довгий час заперечували саму можливість застосування нечіткої логіки в додатках. Ці фахівці часто стверджували, що всі види невизначеності можуть бути виражені в поняттях теорії вірогідності. В той же час, очевидно, що як теорію вірогідності, так і теорію нечітких множин доцільно використовувати для моделювання різних аспектів невизначеності, що відрізняються за своєю природою.

Поняття нечіткої множини здатне забезпечити адекватною інформацією щодо неточного опису тих або інших ситуацій. По суті, цей підхід найбільш застосовний для вирішення таких проблем, в яких невизначеність характеризується відсутністю добре певних критеріїв, що дозволяють однозначно судити про приналежність елементів тому або іншому класу. Саме у цьому виявляється відмінність між нечіткістю і випадковістю. В той же час нечіткі моделі не є заміною моделей, розроблених в те-

орії, вірогідності. Кожна чітка множина є нечіткою, але зворотне твердження не вірне.

Тому нечіткі моделі узагальнюють традиційні і більш знайомі нам математичні моделі. Іноді вони працюють краще, а іноді немає. Врешті-решт, ефективність моделі виявляється в її здатності адекватно вирішити ту або іншу конкретну проблему.

Як правило, складна проблема в тому або іншому ступені пов'язана з невизначеністю. Досягнуті останнім часом вражаючі успіхи в додатку нечітких технологій для вирішення найрізноманітніших практичних завдань дозволяють стверджувати, що нечітке моделювання реальних складних систем – ефективна альтернатива традиційним математичним моделям і методам.

5.3 Операції над нечіткими множинами і відносинами

5.3.1 Логічні операції

Основні логічні операції, які можливі з нечіткими множинами, наступні.

1. *Включення.* Нехай A і B – нечіткі множини на універсальній множині E .

Говорять, що A міститься в B , якщо $\forall x \in E \mu_A(x) \leq \mu_B(x)$

Позначення: $A \subset B$.

Іноді використовують термін «домінування», тобто у разі, коли $A \subset B$, говорять, що B домінує над A .

2. *Рівність.* A і B рівні, якщо $\forall x \in E \mu_A(x) = \mu_B(x)$.

Позначення: $A = B$.

3. *Доповнення.* Нехай A і B – нечіткі множини, задані на E . A і B доповнюють одна одну, якщо $\forall x \in E \mu_A(x) = 1 - \mu_B(x)$.

Позначення: $B = \bar{A}$ або $A = \bar{B}$.

Очевидно, що $\bar{\bar{A}} = A$.

4. *Перетин.* $A \cap B$ – найбільша нечітка підмножина, що міститься одночасно в A і B :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x).$$

5. *Об'єднання.* $A \cup B$ – найменша нечітка підмножина, що включає як A , так і B , з функцією приналежності:

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x).$$

6. *Різниця.* $A - B = A \cap \bar{B}$ з функцією приналежності:

$$\mu_{A-B}(x) = \mu_{A \cap \bar{B}}(x) = \min(\mu_A(x), 1 - \mu_B(x)).$$

7. Диз'юнктивна сума $A \oplus B = (A - B) \cup (B - A) = (A \cap \bar{B}) \cup (\bar{A} \cap B)$ з функцією приналежності:

$$\mu_{A \oplus B}(x) = \max(\min(\mu_A(x), 1 - \mu_B(x)); \min(1 - \mu_A(x), \mu_B(x))).$$

Приклади.

Нехай

$$A = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4,$$

$$B = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4,$$

$$C = 0,1/x_1 + 1/x_2 + 0,2/x_3 + 0,9/x_4.$$

Тут:

1) $A \subset B$, тобто A міститься в B або B домінує над A . C незрівняна ні з A , ні з B , тобто пари $\{A, C\}$ і $\{A, B\}$ – пари недомінуємих нечітких множин.

$$2) A \neq B \neq C.$$

$$3) \bar{A} = 0,6/x_1 + 0,8/x_2 + 1/x_3 + 0/x_4;$$

$$\bar{B} = 0,3/x_1 + 0,1/x_2 + 0,9/x_3 + 0/x_4.$$

$$4) A \cap B = 0,4/x_1 + 0,2/x_2 + 0/x_3 + 1/x_4.$$

$$5) B \cup A = 0,7/x_1 + 0,9/x_2 + 0,1/x_3 + 1/x_4.$$

$$6) A - B = A \cap \bar{B} = 0,3/x_1 + 0,1/x_2 + 0/x_3 + 0/x_4;$$

$$B - A = \bar{A} \cap B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4.$$

$$7) A \oplus B = 0,6/x_1 + 0,8/x_2 + 0,1/x_3 + 0/x_4.$$

Наглядне представлення логічних операцій над нечіткими множинами. Для нечітких множин можна будувати візуальне уявлення. Розглянемо прямокутну систему координат, на осі ординат якої відкладаються значення $\mu_A(x)$, а на осі абсцис в довільному порядку розташовані елементи E . Якщо E за своєю природою впорядковано, то цей порядок бажано зберегти в розташуванні елементів на осі абсцис. Таке уявлення робить наочними прості логічні операції над нечіткими множинами (рис. 5.2).

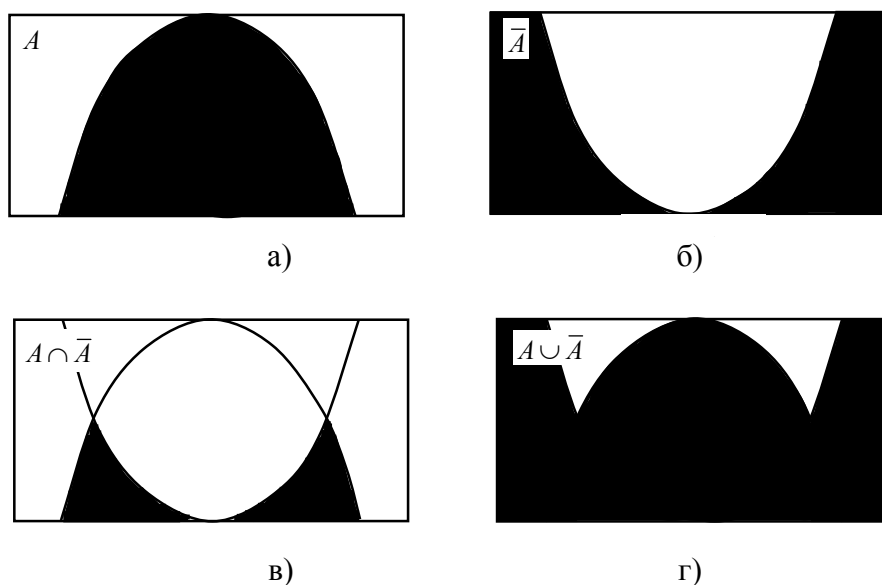


Рис. 5.2. Графічна інтерпретація логічних операцій
а – нечітка множина A ; б – нечітка множина $\bar{A}$; в – $A \cap \bar{A}$; г – $A \cup \bar{A}$

На рисунку 5.2а затемнена область відповідає нечіткій множині A і, якщо говорити точно, зображає область значень A і всіх нечітких множин, що містяться в A . На рис. 5.2 б,в,г показані $\bar{A}$, $A \cap \bar{A}$, $A \cup \bar{A}$.

Властивості операцій об'єднання і перетину.

Нехай A , B , C – нечіткі множині, тоді виконуються наступні властивості.

- 1) $\left. \begin{array}{l} A \cap B = B \cap A \\ A \cup B = B \cup A \end{array} \right\}$ – комутативність;
- 2) $\left. \begin{array}{l} (A \cap B) \cap C = A \cap (B \cap C) \\ (A \cup B) \cup C = A \cup (B \cup C) \end{array} \right\}$ – асоціативність;
- 3) $\left. \begin{array}{l} A \cap A = A \\ A \cup A = A \end{array} \right\}$ – ідемпотентність
- 4) $\left. \begin{array}{l} A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \\ A \cup (B \cap C) = (A \cup B) \cap (A \cup C) \end{array} \right\}$ – дистрибутивність;
- 5) $A \cup \emptyset = A$, де $\emptyset$ – порожня множина, тобто $\mu_{\emptyset} = 0 \forall x \in E$;

$$6) A \cap \emptyset = \emptyset;$$

$$7) A \cap E = A \text{ де } E - \text{універсальна множина.}$$

$$8) A \cup E = E;$$

$$9) \left. \begin{array}{l} \overline{A \cap B} = \bar{A} \cup \bar{B} \\ \overline{A \cup B} = \bar{A} \cap \bar{B} \end{array} \right\} - \text{теореми де Моргана.}$$

На відміну від чітких множин, для нечітких множин в загальному випадку:

$$A \cap \bar{A} \neq \emptyset, \quad A \cup \bar{A} \neq E,$$

(що, зокрема, проілюстровано вище в прикладі наочного представлення нечітких множин).

Примітка. Введені вище операції над нечіткими множинами засновані на використанні операцій $\max$ і $\min$. У теорії нечітких множин розробляються питання побудови узагальнених, параметризованих операторів перетину, об'єднання і доповнення, що дозволяють врахувати різноманітні смислові відтінки відповідних ним зв'язок «І», «АБО», «НЕ».

Один з підходів до операторів перетину і об'єднання полягає в їх визначенні в класі трикутних норм і коном.

Трикутною нормою (T -нормою) називається двомісна дійсна функція $T : [0,1] \times [0,1] \rightarrow [0,1]$, що задовольняє наступним умовам:

- 1) $T(0,0) = 0$; $T(\mu_A, 1) = \mu_A$; $T(1, \mu_A) = \mu_A$ – обмеженість;
- 2) $T(\mu_A, \mu_B) \leq T(\mu_C, \mu_D)$, якщо $\mu_A \leq \mu_C$, $\mu_B \leq \mu_D$ – монотонність;
- 3) $T(\mu_A, \mu_B) = T(\mu_B, \mu_A)$ – комутативність;
- 4) $T(\mu_A, T(\mu_B, \mu_C)) = T(T(\mu_A, \mu_B), \mu_C)$ – асоціативність.

Простим випадком трикутних норм є:

- $\min(\mu_A, \mu_B)$;
- добуток $\mu_A \cdot \mu_B$;
- $\max(0, \mu_A + \mu_B - 1)$.

Трикутною конормою (T -конормою) називається двомісна дійсна функція, з властивостями:

- 1) $S(1, 1) = 1; S(\mu_A, 0) = \mu_A; S(0, \mu_A) = \mu_A$ – обмеженість;
- 2) $S(\mu_A, \mu_B) \geq S(\mu_C, \mu_D)$, якщо $\mu_A \geq \mu_C, \mu_B \geq \mu_D$ – монотонність;
- 3) $S(\mu_A, \mu_B) = S(\mu_B, \mu_A)$ – комутативна;
- 4) $S(\mu_A, S(\mu_B, \mu_C)) = S(S(\mu_A, \mu_B), \mu_C)$ – асоціативність;

Приклади T -конорм:

- $\max(\mu_A, \mu_B)$;
- $\mu_A + \mu_B - \mu_A \cdot \mu_B$;
- $\min(1, \mu_A + \mu_B)$.

5.3.2 Алгебраїчні операції з нечіткими множинами

1. Алгебраїчний добуток A і B позначається $A \cdot B$ і визначається так:

$$\forall x \in E \quad \mu_{A \cdot B}(x) = \mu_A(x) \cdot \mu_B(x).$$

2. Алгебраїчна сума цих множин позначається $A \hat{+} B$ і визначається так:

$$\forall x \in E \quad \mu_{A \hat{+} B}(x) = \mu_A(x) + \mu_B(x) - \mu_A(x) \cdot \mu_B(x).$$

Для операцій $\{\cdot, \hat{+}\}$ виконуються властивості:

- 1) $\left. \begin{array}{l} A \cdot B = B \cdot A \\ A \hat{+} B = B \hat{+} A \end{array} \right\}$ – комутативність;
- 2) $\left. \begin{array}{l} (A \cdot B) \cdot C = A \cdot (B \cdot C) \\ (A \hat{+} B) \hat{+} C = A \hat{+} (B \hat{+} C) \end{array} \right\}$ – асоціативність;
- 3) $A \cdot \emptyset = \emptyset, A \hat{+} \emptyset = A, A \cdot E = A, A \hat{+} E = E$;
- 4) $\left. \begin{array}{l} \overline{A \cdot B} = \overline{A} \hat{+} \overline{B} \\ \overline{A \hat{+} B} = \overline{A} \cdot \overline{B} \end{array} \right\}$ – теореми де Моргана.

Не виконуються:

$$1) \left. \begin{array}{l} A \cdot A = A \\ A \hat{+} A = A \end{array} \right\} - \text{ідемпотентність}$$

$$2) \left. \begin{array}{l} A \cdot (B \hat{+} C) = (A \cdot B) \hat{+} (A \cdot C) \\ A \hat{+} (B \cdot C) = (A \hat{+} B) \cdot (A \hat{+} C) \end{array} \right\} - \text{дистрибутивність}$$

$$3) \text{ а також } A \cdot \bar{A} = \emptyset, \quad A \hat{+} \bar{A} = E.$$

Примітка. При сумісному використанні операцій $\{\cup, \cap, \hat{+}, \cdot\}$ виконуються властивості:

$$1) A \cdot (B \cup C) = (A \cdot B) \cup (A \cdot C).$$

$$2) A \cdot (B \cap C) = (A \cdot B) \cap (A \cdot C).$$

$$3) A \hat{+} (B \cup C) = (A \hat{+} B) \cup (A \hat{+} C).$$

$$4) A \hat{+} (B \cap C) = (A \hat{+} B) \cap (A \hat{+} C).$$

На основі операції алгебраїчного добутку визначається операція *піднесення до ступеня α* нечіткої множини A , де α – позитивне число. Нечітка множина A^α визначається функцією приналежності $\mu_A^\alpha = \mu_A^\alpha(x)$. Окремими випадками піднесення до ступеня є:

$$1) \text{CON}(A) = A^2 - \text{операція концентрації (ущільнення)};$$

$$2) \text{DIL}(A) = A^{0.5} - \text{операція розтягування},$$

які використовуються при роботі з лінгвістичними невизначеностями (рис. 5.3).

$$3. \text{Множення на число. Якщо } \alpha - \text{позитивне число, таке, що } \alpha \max_{x \in A} \mu_A(x) \leq 1,$$

то нечітка множина αA має функцію приналежності:

$$\mu_{\alpha A}(x) = \alpha \mu_A(x).$$

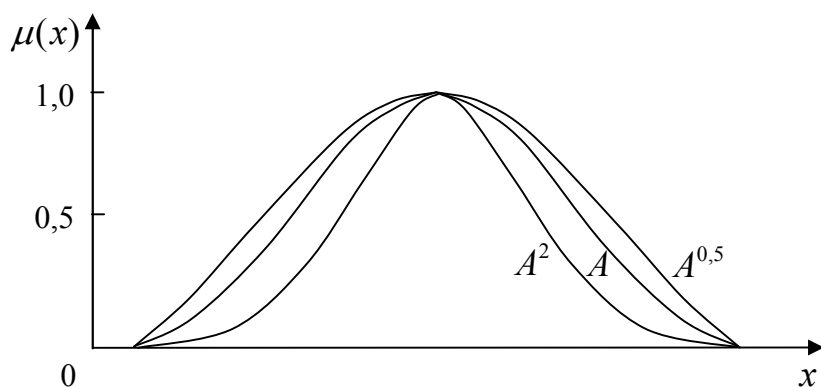


Рис. 5.3. Ілюстрація до поняття операцій концентрування (ущільнення) і розтягування

4. *Опукла комбінація нечітких множин.* Нехай $A_1, A_2, \dots, A_n$ – нечіткі множини універсальної множини E , а $\omega_1, \omega_2, \dots, \omega_n$ – ненегативні числа, сума яких рівна 1.

Опуклою комбінацією $A_1, A_2, \dots, A_n$ називається нечітка множина A з функцією приналежності:

$$\forall x \in E \quad \mu_A(x_1, x_2, \dots, x_n) = \omega_1 \mu_{A_1}(x) + \omega_2 \mu_{A_2}(x) + \dots + \omega_n \mu_{A_n}(x).$$

5. *Декартовий (прямий) добуток нечітких множин.* Нехай $A_1, A_2, \dots, A_n$ – нечіткі підмножини універсальних множин $E_1, E_2, \dots, E_n$ відповідно. Декартовий, або прямий добуток $A = A_1 \times A_2 \times \dots \times A_n$ є нечіткою підмножиною множини $E = E_1 \times E_2 \times \dots \times E_n$ з функцією приналежності:

$$\mu_A(x_1, x_2, \dots, x_n) = \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n)).$$

6. *Оператор збільшення нечіткості* використовується для перетворення чітких множин в нечіткі і для збільшення нечіткості нечіткої множини.

Нехай A – нечітка множина, E – універсальна множина і для всіх $x \in E$ визначені нечіткі множини $K(x)$. Сукупність всіх $K(x)$ називається ядром оператора збільшення нечіткості Φ . Результатом дії оператора Φ на нечітку множину A є нечітка множина вигляду

$$\Phi(A, K) = \bigcup_{x \in E} \mu_A(x) K(x),$$

де $\mu_A(x) K(x)$ – добуток числа на нечітку множину.

Приклад. Нехай

$$E = \{1, 2, 3, 4\};$$

$$A = 0,8/1 + 0,6/2 + 0/3 + 0/4;$$

$$K(1) = 1/1 + 0,4/2; K(2) = 1/2 + 0,4/1 + 0,4/3; K(3) = 1/3 + 0,5/4; K(4) = 1/4.$$

Тоді

$$\begin{aligned}\Phi(A, K) &= \mu_A(1)K(1) \cup \mu_A(2)K(2) \cup \mu_A(3)K(3) \cup \mu_A(4)K(4) = \\ &= 0,8(1/1 + 0,4/2) \cup 0,6(1/2 + 0,4/1 + 0,4/3) = 0,8/1 + 0,6/2 + 0,24/3.\end{aligned}$$

7. *Чітка множина α -рівня (або рівня α).* Множиною α -рівня нечіткої множини A універсальної множини E називається чітка підмножина A_α універсальної множини E , визначувана у вигляді

$$A_\alpha = \{x / \mu_A(x) \geq \alpha\},$$

де $\alpha \leq 1$.

Приклад.

Нехай $A = 0,2/x_1 + 0/x_2 + 0,5/x_3 + 1/x_4$, тоді $A_{0,3} = \{x_3, x_4\}$, $A_{0,7} = \{x_4\}$.

Досить очевидна властивість: якщо $\alpha_1 \geq \alpha_2$, то $A_{\alpha_1} \leq A_{\alpha_2}$.

5.3.3 Нечіткі відносини

Нехай $E = E_1 \times E_2 \times \dots \times E_n$ – прямий добуток універсальних множин. Нечітке n -арне відношення визначається як нечітка підмножина R на E , що приймає свої значення на відрізку $[0, 1]$. У випадку $n = 2$ нечітким відношенням R між множинами $X = E_1$ і $Y = E_2$ називатиметься функція $R: (X, Y) \rightarrow [0, 1]$, яка ставить у відповідність кожній парі елементів $(x, y) \in X \times Y$ величину $\mu_R(x, y) \in [0, 1]$.

Позначення: нечітке відношення на $X \times Y$ запишеться у вигляді:

$$x \in X, y \in Y: xRy.$$

У разі, коли $X = Y$, тобто X і Y співпадають, нечітке відношення $R: X \times X \rightarrow [0, 1]$ називається нечітким відношенням на множині X .

Приклади.

1. Нехай $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3, y_4\}$, $M = [0, 1]$. Нечітке відношення $R = XRY$ може бути задане, наприклад, табл. 5.1.

Таблиця 5.1

Завдання нечіткого відношення

R	y_1	y_2	y_3	y_4
x_1	0	0	0,1	0,3
x_2	0	0,8	1	0,7
x_3	2	0,5	0,6	1

2. Нехай $X = Y = (-\infty, \infty)$, тобто множина всіх дійсних чисел. Відношення $x \gg y$ (x набагато більше y) можна записати наступною функцією приналежності:

$$\mu_R(x, y) = \begin{cases} 0, & \text{якщо } x \leq y, \\ \frac{1}{1 + \frac{1}{(x-y)^2}}, & \text{якщо } x > y. \end{cases}$$

3. Відношення R , для якого $\mu_R(x, y) = e^{-k(x-y)^2}$, при достатньо великих k можна інтерпретувати так: « x і y – близькі один до одного числа».

5.3.4 Операції над нечіткими відносинами

1. *Об'єднання двох відносин* R_1 і R_2 . Об'єднання двох відносин позначається $R_1 \cup R_2$ і визначається виразом

$$\mu_{R_1 \cup R_2}(x, y) = \mu_{R_1}(x, y) \vee \mu_{R_2}(x, y) = \max[\mu_{R_1}(x, y), \mu_{R_2}(x, y)].$$

2. *Перетин двох відносин*. Перетин двох відносин R_1 і R_2 позначається $R_1 \cap R_2$ і визначається виразом

$$\mu_{R_1 \cap R_2}(x, y) = \mu_{R_1}(x, y) \wedge \mu_{R_2}(x, y) = \min[\mu_{R_1}(x, y), \mu_{R_2}(x, y)].$$

3. *Алгебраїчний добуток двох відносин*. Алгебраїчний добуток двох відносин R_1 і R_2 позначається $R_1 \cdot R_2$ і визначається виразом

$$\mu_{R_1 \cdot R_2}(x, y) = \mu_{R_1}(x, y) \cdot \mu_{R_2}(x, y).$$

4. *Алгебраїчна сума двох відносин.* Алгебраїчна сума двох відносин R_1 і R_2 позначається $R_1 \hat{+} R_2$ і визначається виразом

$$\mu_{R_1 \hat{+} R_2}(x, y) = \mu_{R_1}(x, y) + \mu_{R_2}(x, y) - \mu_{R_1}(x, y) \cdot \mu_{R_2}(x, y).$$

5. *Доповнення відношення.* Доповнення відношення R позначається $\bar{R}$ і визначається функцією приналежності:

$$\mu_{\bar{R}}(x, y) = 1 - \mu_R(x, y).$$

6. *Звичайне відношення, найближче до нечіткого.* Нехай R – нечітке відношення з функцією приналежності $\mu_R(x, y)$. Звичайне відношення, найближче до нечіткого, позначається $\underline{R}$ і визначається виразом

$$\mu_{\underline{R}}(x, y) = \begin{cases} 0, & \text{якщо } \mu_R(x, y) < 0,5, \\ 1, & \text{якщо } \mu_R(x, y) > 0,5, \\ 0 \text{ або } 1, & \text{якщо } \mu_R(x, y) = 0,5. \end{cases}$$

За домовленістю приймають $\mu_{\underline{R}}(x, y) = 0$ при $\mu_R(x, y) = 0,5$.

7. *Композиція (згортка) двох нечітких відносин.* Нехай R_1 – нечітке відношення $R_1: (X \times Y) \rightarrow [0, 1]$ між X і Y ; R_2 – нечітке відношення $R_2: (Y \times Z) \rightarrow [0, 1]$ між Y і Z . Нечітке відношення між X і Z , що позначається $R_1 \bullet R_2$, визначене через R_1 і R_2 виразом

$$\mu_{R_1 \bullet R_2}(x, z) = \bigvee_y (\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z)) = \max[\min[(\mu_{R_1}(x, y), \mu_{R_2}(y, z))]],$$

де символом « $\vee$ » позначена операція вибору найбільшого по y значення, називається (max-min)-композицією ((max-min)-згорткою) відносин R_1 і R_2 .

Приклад.

Нехай відношення R_1 і R_2 визначаються наступним чином:

R_1	y_1	y_2	y_3
x_1	0,1	0,7	0,4
x_2	1	0,5	0

R_2	z_1	z_2	z_3	z_4
y_1	0,9	0	1	0,2
y_2	0,3	0,6	0	0,9
y_3	0,1	1	0	0,5

Тоді

$R_1 \cdot R_2$	z_1	z_2	z_3	z_4
x_1	0,3	0,6	0,1	0,7
x_2	0,9	0,5	1	0,5

При цьому

$$\begin{aligned}\mu_{R_1 \cdot R_2}(x_1, z_1) &= [\mu_{R_1}(x_1, y_1) \wedge \mu_{R_2}(y_1, z_1)] \vee [\mu_{R_1}(x_1, y_2) \wedge \mu_{R_2}(y_2, z_1)] \vee \\ &\quad \vee [\mu_{R_1}(x_1, y_3) \wedge \mu_{R_2}(y_3, z_1)] = \\ &= (0,1 \wedge 0,9) \vee (0,7 \wedge 0,3) \vee (0,4 \wedge 0,1) = 0,1 \vee 0,3 \vee 0,1 = 0,3;\end{aligned}$$

$$\mu_{R_1 \cdot R_2}(x_1, z_2) = (0,1 \wedge 0) \vee (0,7 \wedge 0,6) \vee (0,4 \wedge 1) = 0 \vee 0,6 \vee 0,4 = 0,6;$$

$$\mu_{R_1 \cdot R_2}(x_1, z_3) = 0,1;$$

...

$$\mu_{R_1 \cdot R_2}(x_2, z_4) = 0,5.$$

Примітка. В даному прикладі спочатку використаний «аналітичний» спосіб композиції відносин R_1 і R_2 , тобто i -а строка R_1 , «перемножається» на j -й стовець R_2 з використанням операції $\wedge$, отриманий результат «згортається» з використанням операції $\vee$ в $\mu(x_i, z_j)$.

8. (max-*)-композиція.

У виразі $\mu_{R_1 \cdot R_2}(x, z) = \bigvee_y (\mu_{R_1}(x, y) \wedge \mu_{R_2}(y, z))$ для (max-min)-композиції відносин R_1 і R_2 операцію $\wedge$ можна замінити будь-якою іншою, для якої виконуються ті ж обмеження, що і для $\wedge$: асоціативність і монотонність (у сенсі неубування) по кожному аргументу. Тоді

$$\mu_{R_1 \cdot R_2}(x, z) = \bigvee_y (\mu_{R_1}(x, y) * \mu_{R_1}(y, z)).$$

Зокрема, операція $\wedge$ може бути замінена алгебраїчним добутком, тоді говорять про (max – prod)-композицію.

5.3.5 Нечітка імплікація

Нехай маємо звичайні («чіткі») вислови вигляду: $p = \langle x \in A \rangle$ і $q = \langle y \in B \rangle$. Тоді імплікацією (у звичайній, «чіткій» логіці) «якщо p , то q » називається вираз, який помилковий тоді і тільки тоді, коли p істинно, а q помилково.

Такий вид відносин між висловами зазвичай позначається як $p \rightarrow q$ і розуміється як затвердження « p спричиняє за собою q ».

Якщо перейти до бінарної (булевої) алгебри логіки, де з поняттям «істина» зіставляється 1, а з поняттям «брехня» – 0, то імплікацію можна представити таблицею істинності

p	q	$p \rightarrow q$
1	1	1
0	1	1
0	0	1
1	0	0

Повніше трактування поняття імплікації означає, що істинність $p \rightarrow q$ – це те ж, що істинність твердження «ступінь істинності q не менший, ніж ступінь істинності p », тобто

$$p \rightarrow q = \begin{cases} 1 & \text{якщо } \tau(p) \leq \tau(q), \\ 0 & \text{в протилежному випадку,} \end{cases}$$

де $\tau(p)$ і $\tau(q)$ – вказані «ступені істинності» (у чіткій логіці приймаючи значення лише 0 або 1).

Приклад.

Нехай $p = \langle x \text{ більше } 5 \rangle$ і $q = \langle x \text{ більше } 4 \rangle$. Легко бачити, що в даному випадку імплікація $p \rightarrow q$ є істинною, оскільки з нерівності $x > 5$ виходить нерівність $x > 4$.

Нечітка імплікація, в принципі, зберігає той же сенс, що і імплікація чіткої логіки. Відмінність полягає лише в тому, що в цьому випадку «ступені істинності» можуть мати будь-яке значення між 0 і 1.

Нечітка імплікація визначається зазвичай таким чином.

Вважатимемо, що задані універсальні множини X і Y , що містять кінцеве число елементів. Під способом визначення нечіткої імплікації «якщо A , то B », де A і B – нечіткі множини на X і Y відповідно розумітимемо спосіб завдання нечіткого відношення R на $X \times Y$, що відповідає даному вислову.

Таке відношення можна задати по-різному, тому для математичного представлення нечіткої імплікації запропоновано велике число різних формул, деякі з яких разом з прізвищами авторів, що запропонували їх, приведені нижче.

1) Класична нечітка імплікація, запропонована Л.Заде:

$$\mu_R(x, y) = \max \{ \min \{ \mu_A(x), \mu_B(y) \}, 1 - \mu_A(x) \}.$$

Цю нечітку імплікацію називають також нечіткою імплікацією Заде.

2) Класична нечітка імплікація для випадку $\mu_A(x) \geq \mu_B(y)$:

$$\mu_R(x, y) = \max \{ 1 - \mu_A(x), \mu_B(y) \}.$$

3) Нечітка імплікація, запропонована Е.Мамдані:

$$\mu_R(x, y) = \min \{ \mu_A(x), \mu_B(y) \}.$$

Цю форму нечіткої імплікації називають нечіткою імплікацією Мамдані або нечіткою імплікацією мінімуму кореляції. Можна відзначити, що у випадку $\mu_A(x) \geq 0,5$ і $\mu_B(y) \geq 0,5$ класична нечітка імплікація перетворюється в нечітку імплікацію Мамдані.

4) Нечітка імплікація, запропонована Я.Лукасевичем:

$$\mu_R(x, y) = \min \{ 1, 1 - \mu_A(x) + \mu_B(y) \}$$

Цю форму нечіткої імплікації називають нечіткою імплікацією Лукасевича.

5) Нечітка імплікація, запропонована Дж. Гогеном:

$$\mu_R(x, y) = \min \{ 1, \mu_B(y) / \mu_A(x) \}, \text{ де } \mu_A(x) > 0.$$

Цю форму нечіткої імплікації називають нечіткою імплікацією Гогена.

6) Нечітка імплікація по формулі граничної суми::

$$\mu_R(x, y) = \min \{ 1, \mu_A(x) + \mu_B(y) \}.$$

7) Нечітка імплікація по формулі добутку:

$$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y).$$

8) Нечітка імплікація, запропонована Н. Вади:

$$\mu_R(x, y) = \max \{1 - \mu_A(x), \mu_A(x) \cdot \mu_B(y)\}.$$

9) Нечітка імплікація Брауера:

$$\mu_R(x, y) = \begin{cases} 1, & \text{якщо } \mu_A(x) \leq \mu_B(y); \\ \mu_B(y) & \text{в протилежному випадку.} \end{cases}$$

Загалом, якої-небудь переваги однієї формули над іншими немає, тому всі вони (а до теперішнього часу відомо декілька десятків подібних формул) мають право на існування.

Нечітка імплікація грає важливу роль в процесі нечітких логічних міркувань. Також, як і в математичній логіці, перший її операнд (нечіткий вислів) називається посилкою або антецедентом, а другий – висновком, або консеквентом.

5.4 Нечітка і лінгвістична змінні. Нечіткі числа

5.4.1 Визначення нечіткої і лінгвістичної змінних

Поняття нечіткої і лінгвістичної змінних використовується при описі об'єктів і явищ за допомогою нечітких множин.

Нечітка змінна характеризується трійкою $\langle \alpha, X, A \rangle$, де

α – найменування змінної;

X – універсальна множина (область визначення α);

A – нечітка множина на X , що описує обмеження (тобто $\mu_A(x)$) на значення нечіткої змінної α .

Лінгвістичною змінною називається набір $\langle \beta, T, X, G, M \rangle$, де

β – найменування лінгвістичної змінної;

T – множина її значень (терм-множина), що є найменуваннями нечітких змінних, областю визначення кожною з яких є множина X . Множина T називається базовою *терм-множиною* лінгвістичної змінної;

G – синтаксична процедура, що дозволяє оперувати елементами терм-множини T , зокрема, генерувати нові терми (значення). Множина $T \cup G(T)$, де $G(T)$ – множина згенерованих термів, називається розширеною терм-множиною лінгвістичної змінної;

M – семантична процедура, що дозволяє перетворити кожне нове значення лінгвістичної змінної, що утворюється процедурою G , в нечітку змінну, тобто сформувати відповідну нечітку множину.

Зауваження. Щоб уникнути великої кількості символів:

1) символ β використовують як для назви самої змінної, так і для всіх її значень;

2) користуються одним і тим же символом для позначення нечіткої множини і її назви, наприклад терм «Молодий» являється значенням лінгвістичної змінної β =«вік», одночасно є і нечіткою множиною M («Молодий»).

Присвоєння декількох значень символам припускає, що контекст дозволяє вирішити можливі невизначеності.

Приклад. Нехай експерт визначає товщину виробу, що випускається, за допомогою понять «Мала товщина», «Середня товщина» і «Велика товщина», при цьому мінімальна товщина рівна 10 мм, а максимальна – 80 мм.

Формалізація такого опису може бути проведена за допомогою наступної лінгвістичної змінної $\langle \beta, T, X, G, M \rangle$, де

β – товщина виробу;

$T = \{ \text{«Мала товщина», «Середня товщина», «Велика товщина»} \};$

$X = [10, 80];$

G – процедура утворення нових термів за допомогою зв'язок «і», «або» і модифікаторів типу «дуже», «не», «злегка» і т.п. Наприклад: «Мала або середня товщина», «Дуже мала товщина» і т.д.;

M – процедура завдання на $X = [10, 80]$ нечітких підмножин A_1 =«Мала товщина», A_2 =«Середня товщина», A_3 =«Велика товщина», а також нечітких множин для термів із $G(T)$ відповідно до правил трансляції нечітких зв'язок і модифікаторів «і», «або», «не», «дуже», трохи і інших операцій над нечіткими множинами виду: $A \cap B$, $A \cup B$, $\text{CON } A = A^2$, $\text{DIL } A = A^{0.5}$ і т.д.

Зауваження. Разом з розглянутими вище базовими значеннями лінгвістичної змінної «Товщина» ($T = \{ \text{«Мала товщина», «Середня товщина», «Велика товщина»} \}$) можливі значення, залежні від області визначення X . В даному випадку значення лінгвістичної змінної «Товщина виробу» можуть бути визначені як «біля 20 мм», «біля 50 мм», «біля 70 мм», тобто у вигляді нечітких чисел.

Терм-множину і розширену терм-множину в умовах прикладу можна характеризувати функціями приналежності, приведеними на рис. 5.4 і 5.5.

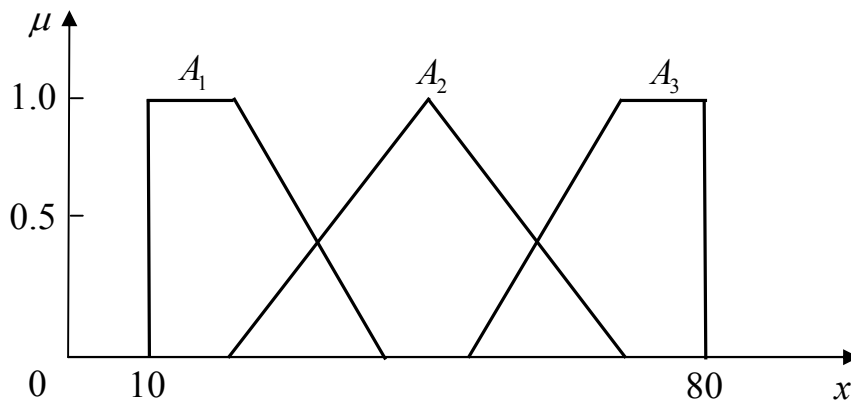


Рис. 5.4. Функції приналежності нечітких множин: A_1 = «Мала товщина», A_2 = «Середня товщина», A_3 = «Велика товщина»

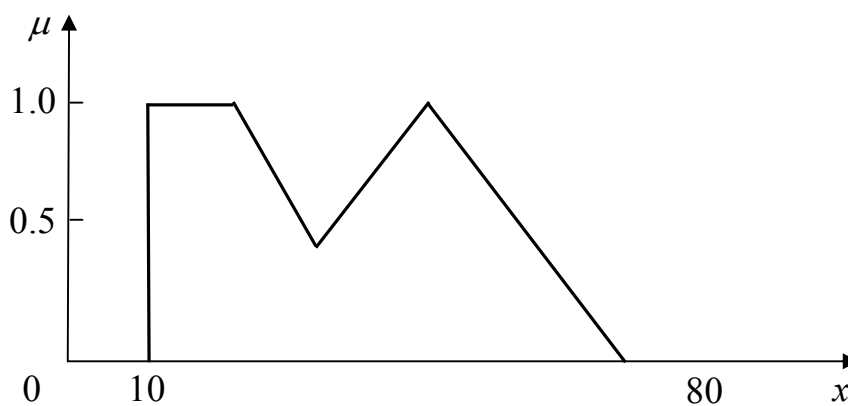


Рис. 5.5. Функція приналежності нечіткої множини $A_1 \cup A_2$ = «Мала або середня товщина»

5.4.2 Нечіткі числа

Нечіткі числа – нечіткі змінні, визначені на числовій осі, тобто нечітке число визначається як нечітка множина A на множині дійсних чисел $\mathbb{R}$ з функцією приналежності $\mu_A(x) \in [0, 1]$, де x – дійсне число, тобто $x \in \mathbb{R}$.

Нечітке число A *нормальне*, якщо $\max_x \mu_A(x) = 1$; *опукле*, якщо для будь-яких $x \leq y \leq z$ виконується

$$\mu_A(x) \geq \mu_B(y) \wedge \mu_A(z).$$

Множина α -рівня нечіткого числа A визначається як

$$A\alpha = \{x / \mu_A(x) \geq \alpha\}.$$

Підмножина $S_A \subset \mathbb{R}$ називається носієм нечіткого числа A , якщо

$$S_A = \{x / \mu_A(x) > 0\}.$$

Нечітке число A *унімодальне*, якщо умова $\mu_A(x)=1$ справедлива тільки для однієї точки дійсної осі.

Опукле нечітке число A називається *нечітким нулем*, якщо

$$\mu_A(x) = \sup_x (\mu_A(x)).$$

Нечітке число A *позитивне*, якщо $\forall x \in S_A, x > 0$ і *негативне*, якщо $\forall x \in S_A, x < 0$.

5.4.3 Операції над нечіткими числами.

Розширені бінарні арифметичні операції (складання, множення і ін.) для нечітких чисел визначаються через відповідні операції для чітких чисел з використанням принципу узагальнення таким чином.

Нехай A і B – нечіткі числа, і $\tilde{*}$ – нечітка операція, відповідна довільній алгебраїчній операції $*$ над звичайними числами. Тоді (використовуючи тут і надалі позначення $\bigvee_x$ замість $\max_x$ і $\bigwedge_x$ замість $\min_x$) можна записати

$$C = A \tilde{*} B - \mu_C(z) = \bigvee_{z=X*Y} (\mu_A(x) \wedge \mu_B(y)).$$

Звідси

$$C = A \tilde{+} B - \mu_C(z) = \bigvee_{z=X+Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{-} B - \mu_C(z) = \bigvee_{z=X-Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{\cdot} B - \mu_C(z) = \bigvee_{z=X \cdot Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = A \tilde{:} B - \mu_C(z) = \bigvee_{z=X \div Y} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = \tilde{\max}(A, B) - \mu_C(z) = \bigvee_{z=\max(X,Y)} (\mu_A(x) \wedge \mu_B(y)),$$

$$C = \tilde{\min}(A, B) - \mu_C(z) = \bigvee_{z=\min(X,Y)} (\mu_A(x) \wedge \mu_B(y)).$$

5.4.4 Нечіткі числа (L – R)-типа

Нечіткі числа (R-L)-типа – це різновид нечітких чисел спеціального вигляду, тобто таких, що задаються за певними правилами з метою зниження об'єму обчислень при операціях над ними.

Функції приналежності нечітких чисел (R-L)-типа задаються за допомогою незростаючих на множині невід'ємних дійсних чисел функцій дійсного змінного $L(x)$ і $R(x)$, що задовольняють властивостям:

- а) $L(-x)=L(x)$, $R(-x)=R(x)$;
- б) $L(0)=R(0)$.

Очевидно, що до класу (R-L)-функцій відносяться функції, графіки яких мають вигляд, приведений на рис. 5.6.

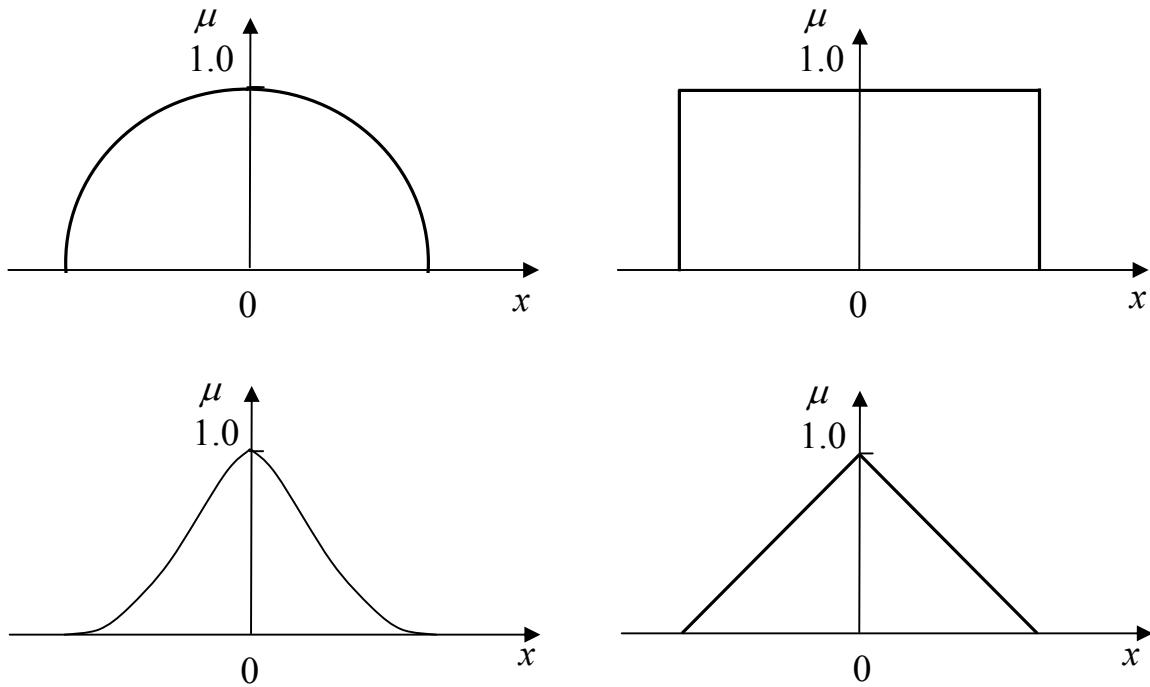


Рис. 5.6. Можливий вигляд (R-L)-функцій

Прикладами аналітичного завдання (R-L)-функцій можуть бути

$$L(x)=e^{-|x|^p}, \quad p \geq 0; \quad R(x)=\frac{1}{1+|x|^p}, \quad p \geq 0,$$

і т.д.

Нехай $L(y)$ і $R(y)$ – функції (R-L)-типа (конкретні). Унімодальне нечітке число A з модою a (тобто $\mu_A(a)=1$) з допомогою $L(y)$ і $R(y)$ задається таким чином:

$$\mu_A(x) = \begin{cases} L\left(\frac{a-x}{\alpha}\right) & \text{при } x \geq a, \\ R\left(\frac{x-a}{\beta}\right) & \text{при } x < a, \end{cases}$$

де a – мода; $\alpha > 0$, $\beta > 0$ – лівий і правий коефіцієнти нечіткості.

Таким чином, при заданих $L(y)$ і $R(y)$ нечітке число (унімодальне) задається трійкою $A = (a, \alpha, \beta)$.

Толерантне нечітке число задається, відповідно, четвіркою параметрів $A = (a_1, a_2, \alpha, \beta)$, де a_1 і a_2 – межі толерантності, тобто в проміжку $[a_1, a_2]$ значення функції приналежності рівне 1.

Приклади графіків функцій приналежності нечітких чисел (R-L)-типа приведені на рис. 5.7.

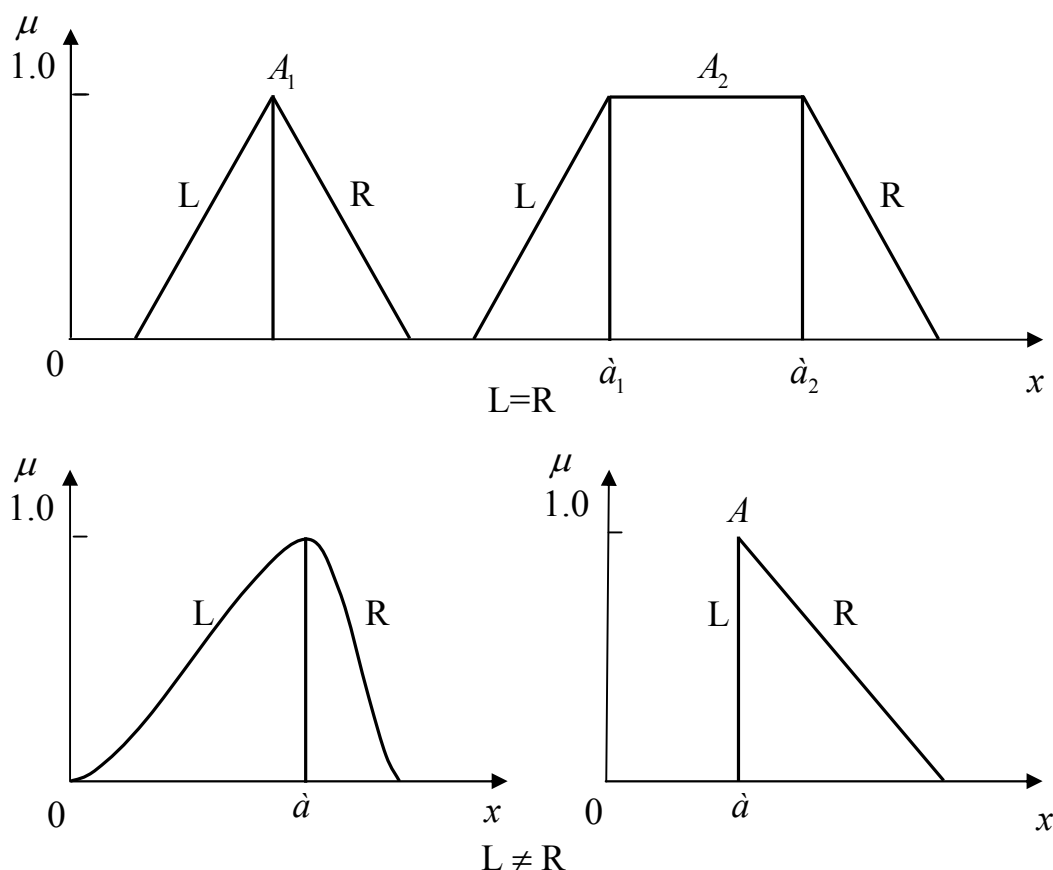


Рис. 5.7. Приклади графіків функцій приналежності нечітких чисел (R-L)-типа

Відзначимо, що в конкретних ситуаціях функції $L(y)$, $R(y)$, а також параметри α , β нечітких чисел (a, α, β) і $(a_1, a_2, \alpha, \beta)$ повинні підбиратися так, щоб результат операції (складання, віднімання, ділення і т.д.) був точно або приблизно рівний нечіткому числу з тими ж $L(y)$ і $R(y)$, а параметри α і β результату не виходили за рамки обмежень на ці параметри для початкових нечітких чисел, особливо якщо результат надалі братиме участь в операціях.

Зауваження. Рішення задач математичного моделювання складних систем із застосуванням апарату нечітких множин вимагає виконання великого об'єму операцій над різного роду лінгвістичними і іншими нечіткими змінними. Для зручності виконання операцій, а також для введення-висновку і зберігання даних, бажано працювати з функціями приналежності стандартного вигляду.

Нечіткі множини, якими доводиться оперувати в більшості завдань, є, як правило, унімодальними і нормальними. Одним з можливих методів апроксимації унімодальних нечітких множин є апроксимація за допомогою функцій (R-L)-типа.

5.5 Нечіткі висновки

5.5.1 Основні етапи нечітких висновків

Використовуваний в різного роду експертних і управляючих системах механізм нечітких висновків в своїй основі має базу знань, що формується фахівцями предметної області у вигляді сукупності нечітких предикативних правил вигляду:

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

P_2 : якщо $x \in A_2$, тоді $y \in B_2$,

.....

P_n : якщо $x \in A_n$, тоді $y \in B_n$,

де x – вхідна змінна (ім'я для відомих значень даних),
 y – змінна висновку (ім'я для значення даних, яке буде обчислено);
 A_i і B_i – нечіткі множини, визначені відповідно на X і Y .

Приклад подібного правила.

Якщо x – «низьке», то y – «високе».

Приведемо детальніше пояснення. Знання експерта $A \rightarrow B$ відображає нечітке причинне відношення передумови і висновку, тому його можна назвати нечітким відношенням і позначити через R :

$$R = A \rightarrow B,$$

де, як і раніше, символ « $\rightarrow$ » означає нечітку імплікацію.

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

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

де « $\bullet$ » – введена вище операція згортки.

Як операцію композиції, так і операцію імплікації в алгебрі нечітких множин можна реалізовувати по-різному (при цьому, природно, різнитиметься і підсумковий

отримуваний результат), але у будь-якому випадку загальний логічний висновок здійснюється за наступні чотири етапи.

1. *Нечіткість (введення нечіткості, фазифікація, fuzzification)*. Функції приналежності, визначені на вхідних змінних застосовуються до їх фактичних значень для визначення ступеня істинності кожної передумови кожного правила.
2. *Логічний висновок*. Обчислене значення істинності для передумов кожного правила застосовується до висновків кожного правила. Це приводить до однієї нечіткої підмножини, яка буде призначена кожній змінній висновку для кожного правила. Як правила логічного висновку, зазвичай використовуються тільки операції $\min$ (МІНІМУМ) або prod (МНОЖЕННЯ). У логічному виведенні МІНІМУМУ функція приналежності висновку «відсікається» по висоті, відповідній обчисленому ступеню істинності передумови правила (нечітка логіка «І»). У логічному виведенні МНОЖЕННЯ функція приналежності висновку масштабується за допомогою обчисленого ступеня істинності передумови правила.
3. *Композиція*. Всі нечіткі підмножини, призначені до кожної змінної висновку (у всіх правилах), об'єднуються разом, щоб формувати одну нечітку підмножину для кожної змінної висновку. При подібному об'єднанні зазвичай використовуються операції $\max$ (МАКСИМУМ) або sum (СУМА). При композиції МАКСИМУМУ комбіноване виведення нечіткої підмножини конструюється як поточечний максимум по всіх нечітких підмножинах (нечітка логіка «АБО»). При композиції СУМИ комбіноване виведення нечіткої підмножини конструюється як поточечна сума по всіх нечітких підмножинах, призначених змінній висновку правилами логічного висновку.
4. На закінчення (додатково) – *приведення до чіткості (дефазифікація, defuzzification)*, яке використовується, коли корисно перетворити нечіткий набір висновків в чітке число. Є велика кількість методів приведення до чіткості, деякі з яких розглянуті нижче.

Приклад.

Нехай деяка система описується наступними нечіткими правилами:

P_1 : якщо $x \in A$, тоді $w \in D$,

P_2 : якщо $y \in B$, тоді $w \in E$,

P_3 : якщо $z \in C$, тоді $w \in F$,

де x, y і z – імена вхідних змінних,

w – ім'я змінної висновку;

A, B, C, D, E, F – нечіткі множини, визначені на множині дійсних чисел з функціями приналежності відповідно $\mu_A, \mu_B, \mu_C, \mu_D, \mu_E, \mu_F$.

Процедура отримання логічного висновку ілюструється рис. 5.8.

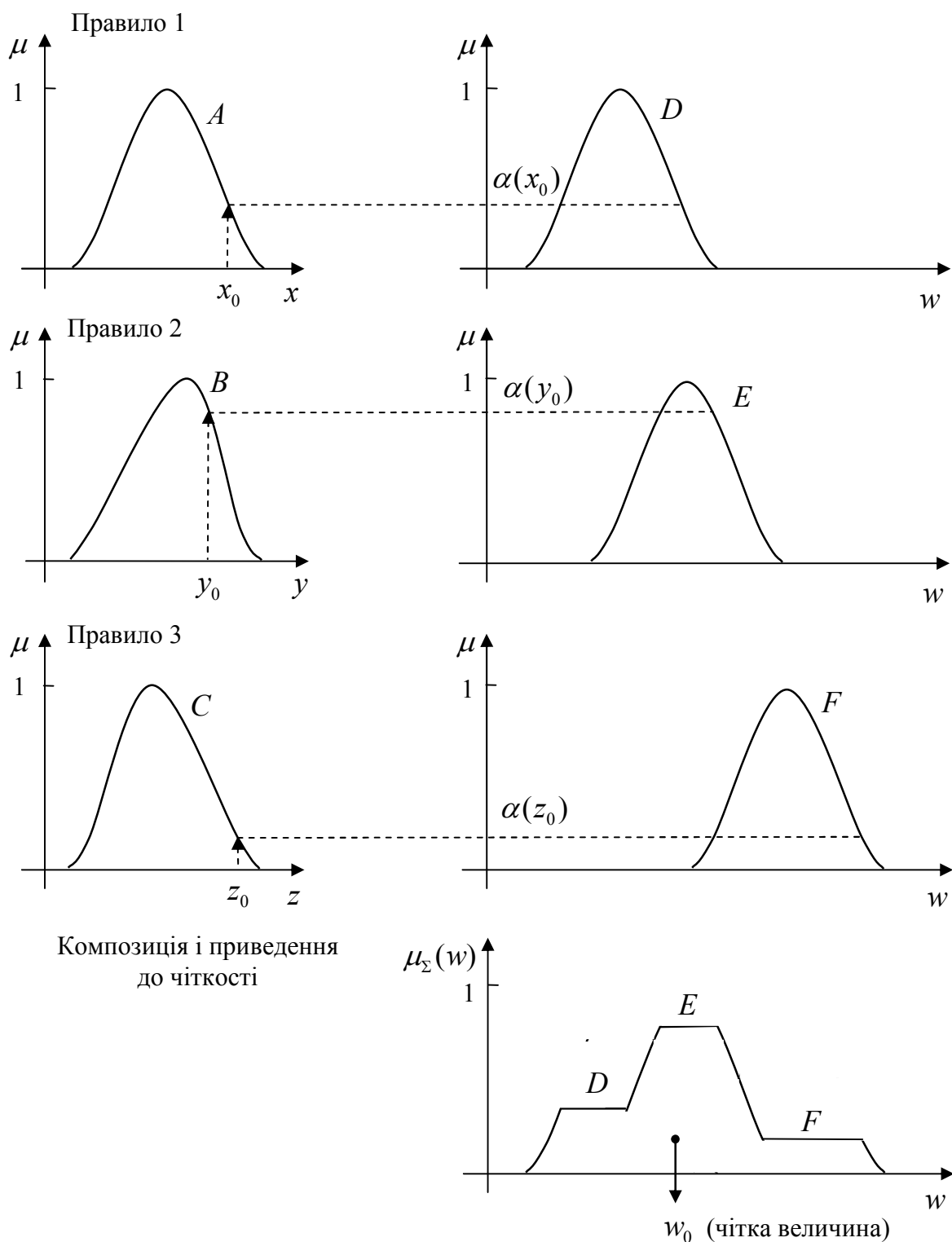


Рис. 5.8. Ілюстрація до процедури логічного висновку

Передбачається, що входні змінні прийняли деякі конкретні (чіткі) значення – x_0 , y_0 і z_0

Відповідно до приведених етапів, на етапі 1 для даних значень і виходячи з функцій приналежності A , B і C , знаходяться ступені істинності $\alpha(x_0)$, $\alpha(y_0)$ і $\alpha(z_0)$ для передумов кожного з трьох приведених правил (див. рис. 5.8).

На етапі 2 відбувається «відсікання» функцій приналежності висновків правил (тобто μ_D, μ_E, μ_F) на рівнях $\alpha(x_0), \alpha(y_0)$ і $\alpha(z_0)$.

На етапі 3 розглядаються усічені на другому етапі функції приналежності і проводиться їх об'єднання з використанням операції $\max$, внаслідок чого виходить комбінована нечітка підмножина, що описується функцією приналежності $\mu_\Sigma(w)$ і відповідна логічному висновку для вихідної змінної w .

Нарешті, на 4-му етапі – при необхідності – знаходиться чітке значення вихідної змінної, наприклад, із застосуванням центроїдного методу: чітке значення вихідної змінної визначається як центр тяжіння для кривої $\mu_\Sigma(w)$, тобто

$$w_0 = \frac{\int_{\Omega} w \mu_\Sigma(w) dw}{\int_{\Omega} \mu_\Sigma(w) dw}.$$

Тут Ω – область визначення $\mu_\Sigma(w)$.

5.5.2 Алгоритми нечіткого висновку

Розглянемо наступні найбільш часто використовувані модифікації алгоритму нечіткого висновку, вважаючи, для простоти, що базу знань організовують два нечітких правила вигляду:

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

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

де x і y – імена вхідних змінних;

z – ім'я змінної висновку;

$A_1, A_2, B_1, B_2, C_1, C_2$ – деякі нечіткі множини, задані функціями приналежності $\mu_{A_1}, \mu_{A_2}, \mu_{B_1}, \mu_{B_2}, \mu_{C_1}, \mu_{C_2}$.

Чітке значення z_0 необхідно визначити на основі приведеної інформації і чітких значень x_0 і y_0 .

Алгоритм Mamdani

Одним з найбільш поширених алгоритмів нечіткого висновку є *алгоритм Мамдані* (Mamdani.). Даний алгоритм відповідає розглянутому вище прикладу (див. рис. 5.8), а також рис. 5.9.

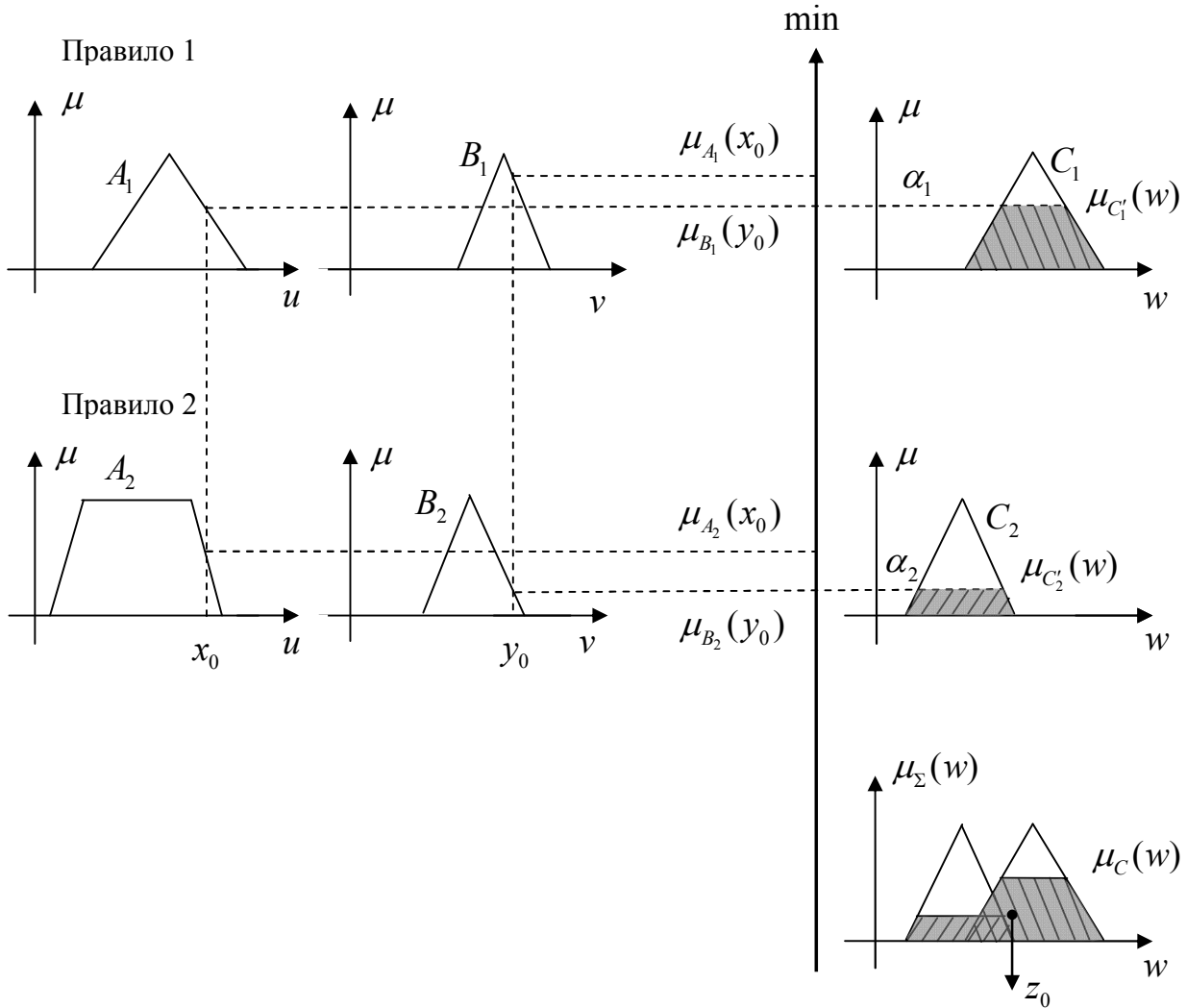


Рис. 5.9. Ілюстрації до алгоритму Mamdani

У даній ситуації він математично може бути описаний таким чином.

1. Нечіткість: знаходяться ступені 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. Нечіткий висновок: знаходяться рівні «відсічення» для передумов кожного з правил (з використанням операції МІНІМУМ)

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

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

де через « $\wedge$ » позначена операція логічного мінімуму (min), потім знаходяться «усічені» функції приналежності

$$\mu_{C_1}(w) = (\alpha_1 \wedge \mu_{C_1}(w)),$$

$$\mu_{C_2}(w) = (\alpha_2 \wedge \mu_{C_2}(w)).$$

3. Композиція: з використання операції МАКСИМУМ (max, що далі позначається як « $\vee$ ») проводиться об'єднання знайдених усічених функцій, що приводить до отримання підсумкової нечіткої підмножини для змінній виходу з функцією приналежності

$$\mu_{\Sigma}(w) = \mu_C(w) = \mu_{C_1}(w) \vee \mu_{C_2}(w) = (\alpha_1 \wedge \mu_{C_1}(w)) \vee (\alpha_2 \wedge \mu_{C_2}(w)).$$

4. Приведення до чіткості (для знаходження z_0) проводиться, наприклад, центроїдним методом.

Алгоритм Tsukamoto

Початкові посилки – як у попереднього алгоритму, але в даному випадку передбачається, що функції $\mu_{C_1}(z)$, $\mu_{C_2}(z)$ є монотонними.

1. Перший етап – такий же, як в алгоритмі Mamdani.

2. На другому етапі спочатку знаходяться (як в алгоритмі Mamdani) рівні «відсікання» z_1 і z_2 , а потім – за допомогою рішення рівнянь

$$\alpha_1 = \mu_{C_1}(z_1), \quad \alpha_2 = \mu_{C_2}(z_2),$$

чіткі значення (z_1 і z_2) для кожного з вихідних правил.

3. Визначається чітке значення змінної висновку (як зважене середнє z_1 і z_2):

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2};$$

у загальному випадку (дискретний варіант центроїдного методу)

$$z_0 = \frac{\sum_{i=1}^n \alpha_i z_i}{\sum_{i=1}^n \alpha_i}.$$

Приклад. Нехай маємо $\mu_{A_1}(x_0)=0,7$, $\mu_{A_2}(x_0)=0,6$, $\mu_{B_1}(y_0)=0,3$, $\mu_{B_2}(y_0)=0,8$.

Відповідні рівні відсікання

$$\alpha_1 = \min(\mu_{A_1}(x_0), \mu_{B_1}(y_0)) = \min(0,7; 0,3) = 0,3,$$

$$\alpha_2 = \min(\mu_{A_2}(x_0), \mu_{B_2}(y_0)) = \min(0,6; 0,8) = 0,6$$

і значення $z_1 = 8$ і $z_2 = 4$, знайдені в результаті рішення рівнянь

$$\mu_{C_1}(z_1) = 0,3, \quad \mu_{C_2}(z_2) = 0,6.$$

При цьому чітке значення змінної висновку (див. рис. 5.10)

$$z_0 = (8 \cdot 0,3 + 4 \cdot 0,6) / (0,3 + 0,6) = 6.$$

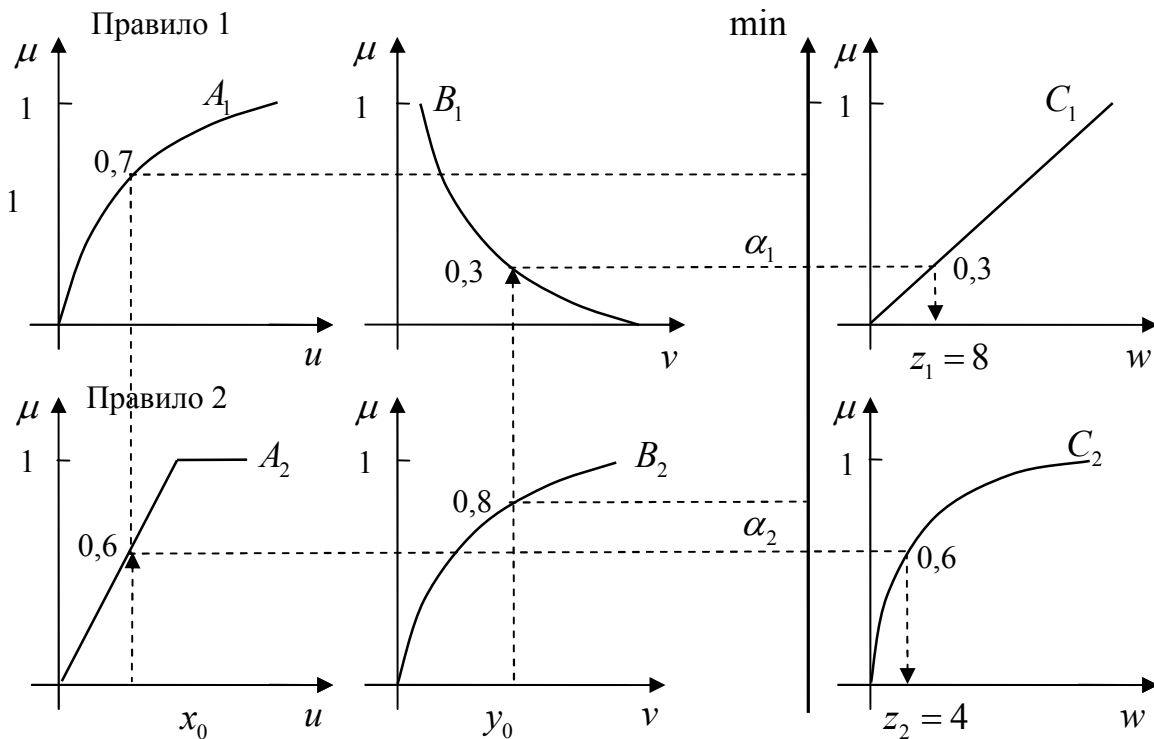


Рис. 5.10. Ілюстрації до алгоритму Tsukamoto

Алгоритм Sugeno

Sugeno і Takagi використовували набір правил в наступній формі (як і раніше, приводиться приклад двох правил):

Π_1 : якщо $x \in A_1$ і $y \in B_1$, тоді $z_1 = a_1x + b_1y$,

Π_2 : якщо $x \in A_2$ і $y \in B_2$, тоді $z_2 = a_2x + b_2y$.

Представлення алгоритму.

1. Перший етап – як в алгоритмі Mamdani.

2. На другому етапі знаходяться $\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$, $\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$ і індивідуальні виходи правил:

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

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

3. На третьому етапі визначається чітке значення змінної висновку:

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

Ілюструє алгоритм рис. 5.11.

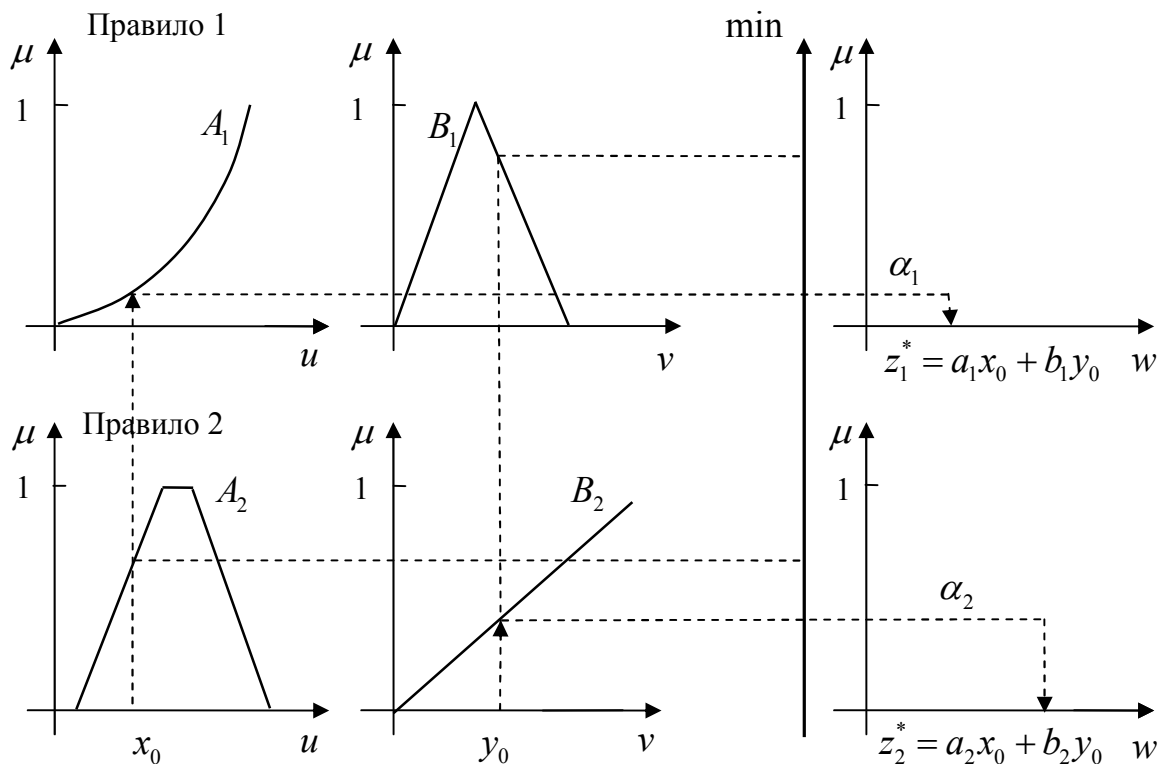


Рис. 5.11. Ілюстрація до алгоритму Sugeno

Алгоритм Larsen

У алгоритмі Larsen нечітка імплікація моделюється з використанням оператора множення.

Опис алгоритму.

1. Перший етап – як в алгоритмі Mamdani.

2. На другому етапі, як в алгоритмі Mamdani спочатку знаходяться значення

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

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

а потім – часткові нечіткі підмножини

$$\alpha_1 \mu_{C_1}(w), \quad \alpha_2 \mu_{C_2}(w).$$

3. Знаходиться підсумкова нечітка підмножина з функцією приналежності

$$\mu_{\Sigma}(w) = \mu_C(w) = (\alpha_1 \mu_{C_1}(w)) \vee (\alpha_2 \mu_{C_2}(w))$$

(у загальному випадку n правил $\mu_{\Sigma}(w) = \mu_C(w) = \bigvee_{i=1}^n (\alpha_i \mu_{C_i}(w))$).

4. При необхідності проводиться приведення до чіткості (як в раніше розглянутих алгоритмах).

Алгоритм Larsen ілюструється рис. 5.12.

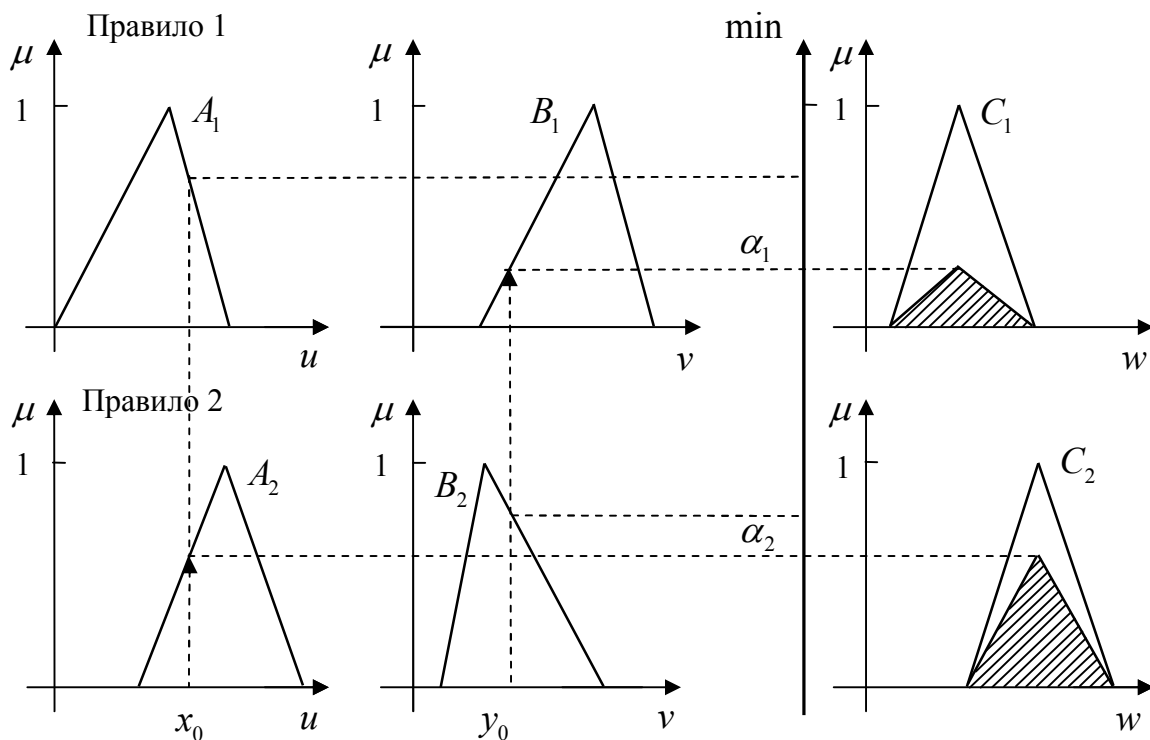


Рис. 5.12. Ілюстрація алгоритму Larsen

Спрощений алгоритм нечіткого висновку

Початкові правила в даному випадку задаються у вигляді:

P_1 : якщо $x \in A_1$ і $y \in B_1$, тоді $z_1 = c_1$,

P_2 : якщо $x \in A_2$ і $y \in B_1$, тоді $z_2 = c_2$,

де c_1 і c_2 – деякі звичайні (чіткі) числа.

Опис алгоритму

1. Перший етап – як в алгоритмі Mamdani.

2. На другому етапі знаходяться числа $\alpha_1 = \mu_{A_1}(x_0) \wedge \mu_{B_1}(y_0)$,
 $\alpha_2 = \mu_{A_2}(x_0) \wedge \mu_{B_2}(y_0)$.

3. На третьому етапі знаходиться чітке значення вихідної змінної по формулі:

$$z_0 = \frac{\alpha_1 c_1 + \alpha_2 c_2}{\alpha_1 + \alpha_2},$$

або в загальному випадку при наявності n правил – по формулі:

$$z_0 = \frac{\sum_{i=1}^n \alpha_i c_i}{\sum_{i=1}^n \alpha_i}.$$

Ілюстрація алгоритму приведена на рис. 5.13.

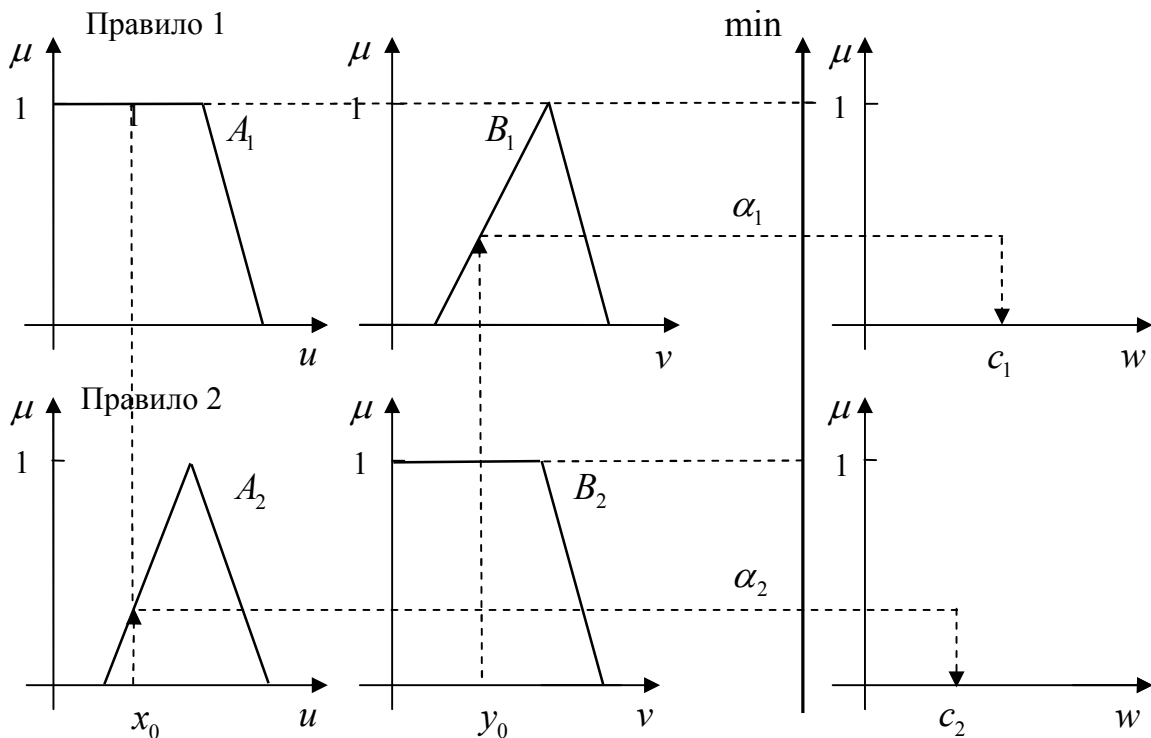


Рис. 5.13. Ілюстрація спрощеного алгоритму нечіткого висновку

5.5.3 Методи приведення до чіткості

1. Вище вже був розглянутий один з даних методів – центроїдний. Приведемо відповідні формули ще раз. Для безперервного варіанту:

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

для дискретного варіанту:

$$z_0 = \frac{\sum_{i=1}^n \mu_{\Sigma}(z_i) z_i}{\sum_{i=1}^n \mu_{\Sigma}(z_i)}.$$

2. Перший максимум (First-of-Maxima). Чітка величина змінної висновку знаходиться як найменше значення, при якому досягається максимум підсумкової нечіткої множини, тобто (див. рис. 5.14 а)

$$z_0 = \min(z \mid \mu_{\Sigma}(z) = \max_z \mu_{\Sigma}(z)).$$

3. Середній максимум (Middle-of-Maxima). Чітке значення знаходиться по формулі

$$z_0 = \frac{\int_G z dz}{\int_G dz},$$

де G – підмножина елементів, що максимізували μ_{Σ} (див. рис. 5.14 б).

Дискретний варіант (якщо μ_{Σ} – дискретне):

$$z_0 = \frac{1}{N} \sum_{j=1}^N z_j$$

4. Критерій максимуму (Max-Criterion). Чітке значення вибирається довільно серед множини елементів, що доставляють максимум, тобто

$$z_0 \in \{z \mid \mu_{\Sigma}(z) = \max_z \mu_{\Sigma}(z)\}.$$

5. Висотна дефазифікація (Height defuzzification). Елементи області визначення, для яких значення функції приналежності менше, ніж деякий рівень α в розрахунок не приймаються, і чітке значення розраховується по формулі

$$z_0 = \frac{\int_{C_\alpha} z \mu_\Sigma(z) dz}{\int_{C_\alpha} \mu_\Sigma(z) dz},$$

де C_α – нечітка множина α – рівня (див. вище).

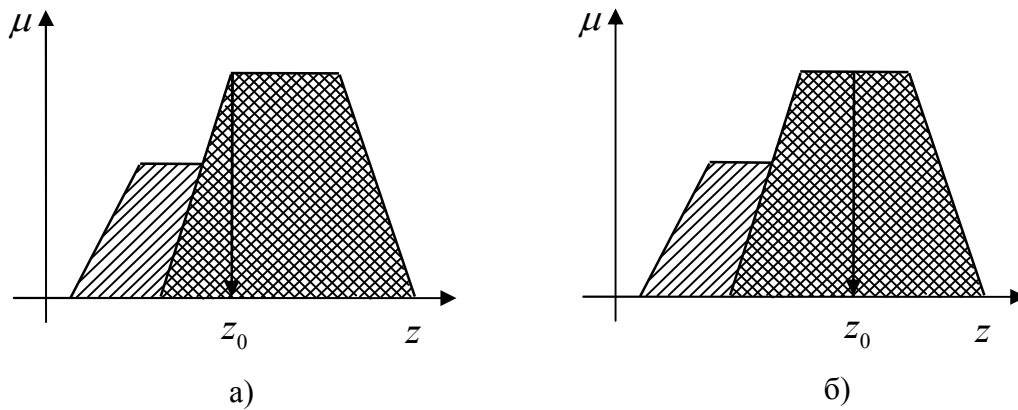


Рис. 5.14. Ілюстрація до методів приведення до чіткості:
а – перший максимум; б – середній максимум

5.6 Використання систем нечіткого висновку в задачах управління

Одним з основних напрямів практичного використання систем нечіткого висновку є рішення задач управління різними об'єктами або процесами. В цьому випадку побудова нечіткої моделі ґрунтується на формальному представленні характеристик досліджуваної системи в термінах лінгвістичних змінних. Оскільки окрім алгоритму управління, основними поняттями систем управління є вхідні і вихідні змінні, то саме вони розглядаються як лінгвістичні змінні при формуванні бази правил в системах нечіткого висновку.

У загальному випадку мета управління полягає в тому, щоб на основі аналізу поточного стану об'єкту управління визначити значення управляючих змінних, реалізація яких дозволяє забезпечити бажану поведінку або стан об'єкту управління. В даний час для вирішення відповідних завдань використовується загальна теорія управління, в рамках якої розроблені різні алгоритми знаходження оптимальних законів управління об'єктами різної фізичної природи.

Не вдаючись до детального обговорення концепцій класичної теорії управління, розглянемо лише основні визначення, необхідні для розуміння особливостей і місця систем нечіткого висновку при рішенні задач управління. Базова архітектура або модель класичної теорії управління ґрунтується на представленні об'єкту і процесу управління у формі деяких систем (рис. 5.15). При цьому об'єкт управління ха-

рактизується деякою кінцевою множиною вхідних параметрів і кінцевою множиною вихідних параметрів. На вхід системи управління поступають деякі вхідні змінні, які формуються за допомогою кінцевої множини датчиків. На виході системи управління з використанням деякого алгоритму управління формується множина значень вихідних змінних, які ще називають управляючими змінними або змінними процесу управління. Значення цих вихідних змінних поступають на вхід об'єкту управління і, комбінуючись із значеннями вхідних параметрів об'єкту управління, змінюють його поведінку в бажаному напрямі.

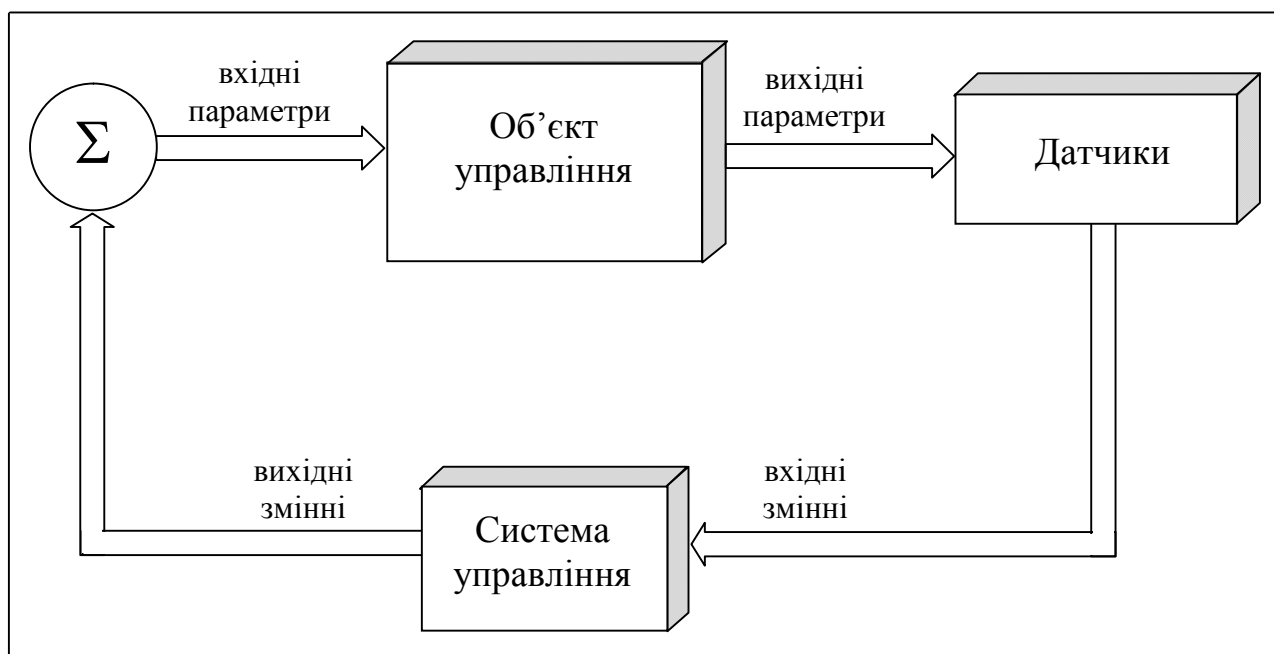


Рис. 5.15. Архітектура компонентів процесу управління із зворотним зв'язком

Як відомо, розглянута архітектура називається процесом управління із зворотним зв'язком, а використовувані для управління технічними об'єктами системи управління – контролерами.

Найбільш типовим прикладом розглянутої моделі управління є так званий інтегрально-диференціюючий контролер або PID-контролер (proportional-integral-derivative controller). Алгоритм його управління заснований на порівнянні вихідних параметрів об'єкту управління з деякими заданими параметрами і визначенні величини розбіжності між ними або помилки. Після цього розраховуються величини вихідних змінних у формі адитивної суми величини цієї помилки, значення інтеграла і похідної за часом протягом деякого проміжку часу.

Один з недоліків PID-контролерів полягає в припущенні про лінійний характер залежності вхідних і вихідних змінних процесу управління, що істотно знижує адекватність цієї моделі при рішенні окремих практичних задач. Інший недолік моделі пов'язаний з складністю виконання відповідних розрахунків, що може привести до неприпустимих затримок в реалізації управляючих дій при оперативному управлінні об'єктами з високою динамікою зміни вихідних параметрів.

Архітектура або модель нечіткого управління заснована на заміні класичної системи управління системою нечіткого управління, у якості якої використовуються системи нечіткого висновку. В цьому випадку модель нечіткого управління (рис. 5.16) будується з урахуванням необхідності реалізації всіх етапів нечіткого висновку, а сам процес висновку реалізується на основі одного з розглянутих вище алгоритмів нечіткого висновку.

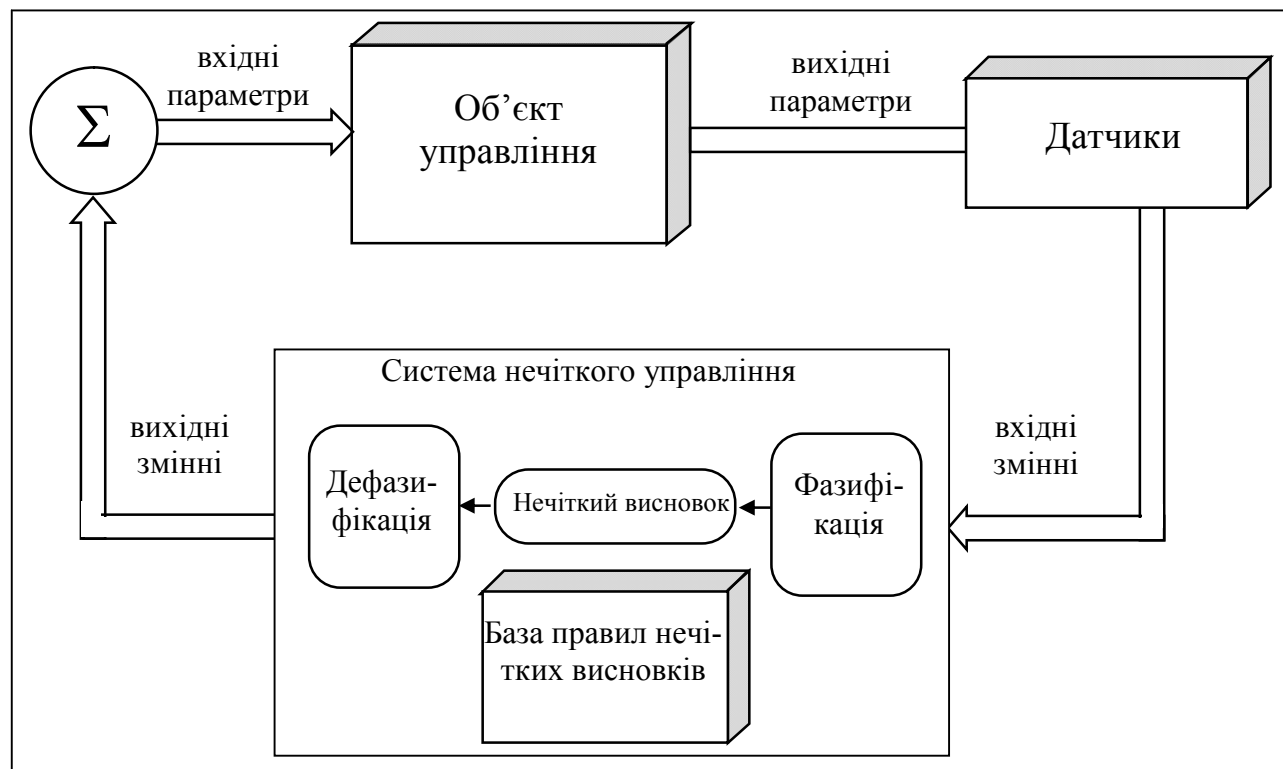


Рис. 5.16. Архітектура компонентів процесу нечіткого управління

Питання для самоконтролю.

1. У яких випадках доцільно використовувати нечітке управління?
2. Які теореми обґрунтовують можливість використання апарату нечіткої логіки в задачах управління?
3. Перечисліть основні недоліки систем з нечіткою логікою.
4. Дайте визначення нечіткої множини. Наведіть приклади завдання нечіткої множини.
5. Перечисліть і поясніть основні характеристики нечітких множин.
6. Дайте визначення функції приналежності нечіткої логіки.
7. Які типові форми функцій найчастіше використовуються в нечіткій логіці для завдання функцій приналежності? Наведіть їх рівняння і графіки.
8. Перечисліть основні логічні операції, які можливі з нечіткими множинами. Дайте їх графічну інтерпретацію.
9. Перечисліть алгебраїчні операції з нечіткими множинами.
10. Дайте визначення нечіткого n -арного відношення. Які способи його завдання?
11. Перечисліть операції над нечіткими відносинами.
12. Дайте визначення нечіткої імплікації.
13. Дайте визначення нечіткої і лінгвістичної змінних. Наведіть приклад.
14. Дайте визначення нечітких чисел.
15. Які операції можливі з нечіткими числами?
16. Що таке нечіткі числа (R-L)-типа? Наведіть приклади.
17. Перечисліть етапи формування логічного висновку. Наведіть приклади формування нечітких висновків.
18. Наведіть алгоритми нечіткого висновку: Mamdani, Tsukamoto, Sugeno, Larsen, спрощений алгоритм нечіткого висновку.
19. Перечисліть методи приведення до чіткості. Наведіть відповідні формули.
20. Наведіть структурну схему системи управління з нечітким регулятором.

ГІБРИДНІ МЕРЕЖІ

Кожен різновид систем штучного інтелекту – як нейронні мережі, так і системи, в яких використовується нечітка логіка мають достоїнства і недоліки. В даний час розроблені структури, які об'єднують якнайкращі властивості обох методів, і в той же час вільні від їх проблем. За певних умов нечітка система може бути представлена у формі багатошарової мережі з прямим розповсюдженням сигналу, яку можна навчати по методу зворотного розповсюдження помилки.

Нейронні мережі і нечіткі системами, як було зазначено вище, вельми успішно справляються із завданнями, які традиційні системи регулювання вирішують не самим кращим чином. Якщо виникає необхідність управляти об'єктом, який має неодноточні властивості, опис якого неповний або не може бути зведений до простої математичної моделі, то доводиться шукати рішення, альтернативні «звичайним» способам управління, і найчастіше вибираються нейронні мережі або системи з нечіткою логікою. Відзначимо ще раз, чим характеризуються ці методи.

Найважливішим достоїнством нейронних мереж вважається можливість їх навчання і адаптації. При використанні нейронних мереж не потрібні повні знання про об'єкт управління (наприклад, його математична модель). На основі вхідних і заданих (еталонних) сигналів нейронна мережа може навчитися управляти об'єктом. Нейронні мережі (див. гл. 1) складаються з великої кількості взаємозв'язаних простих елементів (нейронів), що в результаті дає величезну обчислювальну потужність при використанні паралельної обробки інформації. На жаль, спосіб проектування таких систем ґрунтується скоріше на інтуїції, ніж на існуючих закономірностях. До теперішнього часу невідомий алгоритм розрахунку кількості шарів мережі і кількості нейронів в кожному шарі для конкретних застосувань. Проте, після закінчення навчання нейронні мережі стають незамінними засобами рішення задач управління, розпізнавання образів, апроксимації, оптимізації або класифікації. З іншого боку, накопичені нейронною мережею знання виявляються розподіленими між всіма її елементами, що робить їх практично недоступними для спостерігача.

Цього недоліку позбавлені системи управління з нечіткою логікою. Проте в даному випадку знання про спосіб управління необхідні вже на стадії проектування управляючих модулів, причому вони повинні виходити від експертів і, отже, можливість навчання відсутня. Проте і в такій ситуації повні знання (що описують в математичному вигляді функціональну залежність між входами і виходами системи) не потрібні. На відміну від звичайних модулів управління, використовуються не кількісні («скільки?»), а якісні («як?») знання. Система ухвалює рішення на основі правил, записаних у формі імплікації «якщо-то». Простий підхід до проектування таких систем полягає у формулюванні правил управління і функції приналежності за наслідками спостереження за процесом управління, здійснюваного людиною або

вже існуючим регулятором, з подальшим оцінюванням коректності функціонування такої системи. Якщо проект виявляється невдалим, то функцію приналежності і/або правила управління можна легко модифікувати. Як вже наголошувалося, основний недолік подібних систем – це неможливість адаптації і навчання.

Об'єднання обох підходів дозволяє, з одного боку, привнести здібність до навчання і обчислювальну потужність нейронних мереж в системи з нечіткою логікою, а з іншого боку – підсилити інтелектуальні можливості нейронних мереж власними «людському» способу мислення нечіткими правилами вироблення рішень.

Спроби такого об'єднання стали предметом вельми інтенсивних досліджень і розглядаються в багатьох роботах, наприклад [61-73]. Їх результатом можна вважати системи вироблення рішень, що різною мірою реалізують ідею нечіткого мислення в комплексі із запозиченою від нейронних мереж здібністю до навчання.

6.1 Основні поняття і визначення гібридних мереж

Для пояснення суті гібридних мереж, розглянемо ще раз просту нейронну мережу, що має два входи і лише один нейрон (рис. 6.1).

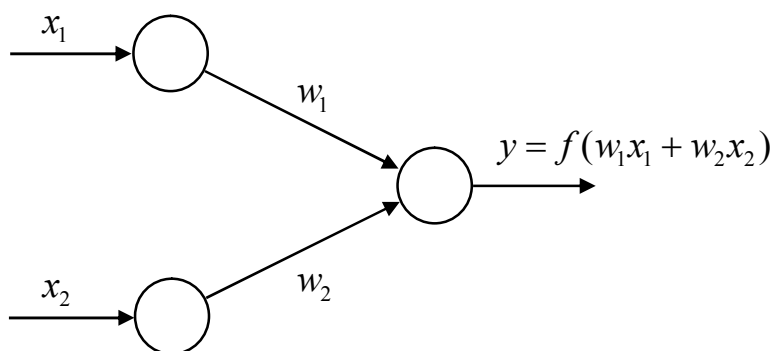


Рис. 6.1. Елементарна нейронна мережа

Тут вхідні сигнали x_i «взаємодіють» з вагами w_i , утворюючи добутки

$$p_i = x_i w_i, \quad i = 1, 2.$$

Така приватна інформація (добутки) об'єднуються з використанням операції підсумовування, утворюючи вхід net нейрона:

$$net = p_1 + p_2 = w_1x_1 + w_2x_2.$$

Вихід нейрона утворюється в результаті перетворення входу net деякою активаційною функцією:

$$y = f(net) = f(w_1x_1 + w_2x_2),$$

наприклад, сигмоїдного типу

$$f(x) = \frac{1}{1 + e^{-x}}.$$

Приведену одношарову нейронну мережу, в якій використовуються операції множення, підсумовування і сигмоїдна функція активації, називається стандартною нейронною мережею.

У разі застосування інших операцій, таких як T -норма або T -конорма (див. гл. 4), утворюється нейронна мережа, яка називається гібридною.

Гібридна нейронна мережа є багатошаровою нейронною мережею спеціальної структури без зворотних зв'язків, в якій використовуються звичайні (не нечеткі) сигнали, ваги і функції активації, а виконання операцій підсумовування засноване на використанні T -норми, T -конорми або деякої іншої безперервної операції. При цьому значення входів, виходів і вагів гібридної нейронної мережі є дійсними числами, що належать відрітку $[0, 1]$.

Розглянемо наступні приклади елементарних гібридних нейронних мереж.

Нечіткий нейрон «І». Сигнали x_i і ваги w_i в даному випадку об'єднуються за допомогою трикутної конорми:

$$p_i = S(x_i, w_i), \quad i = 1, 2,$$

а вихід утворюється із застосуванням трикутної норми (рис. 6.2):

$$y = \text{AND}(p_1, p_2) = T(p_1, p_2) = T(S(w_1, x_1), S(w_2, x_2)).$$

Якщо прийняти

$$T = \min, \quad S = \max,$$

тоді нечіткий нейрон «І» реалізує композицію $\min$ – $\max$:

$$y = \min(w_1 \vee x_1, w_2 \vee x_2).$$

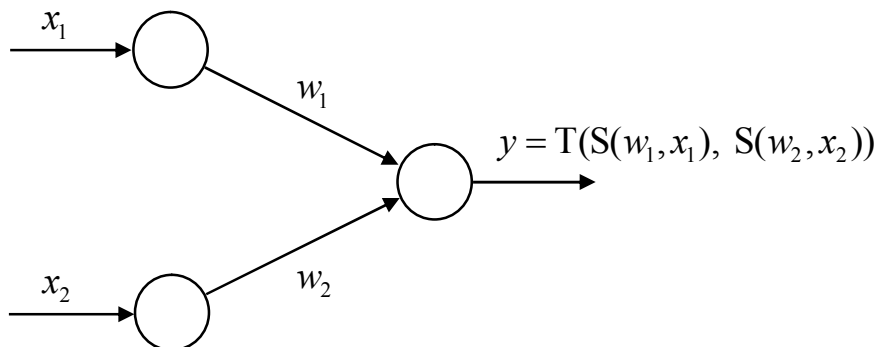


Рис. 6.2. Структура гібридного нейрона «І»

Нечіткий нейрон «АБО». Сигнали x_i і ваги w_i тут об'єднуються за допомогою трикутної норми:

$$p_i = T(x_i, w_i), \quad i = 1, 2,$$

а вихід утворюється із застосуванням трикутної конорми (див. рис. 6.3):

$$y = OR(p_1, p_2) = S(p_1, p_2) = T(T(w_1, x_1), T(w_2, x_2)).$$

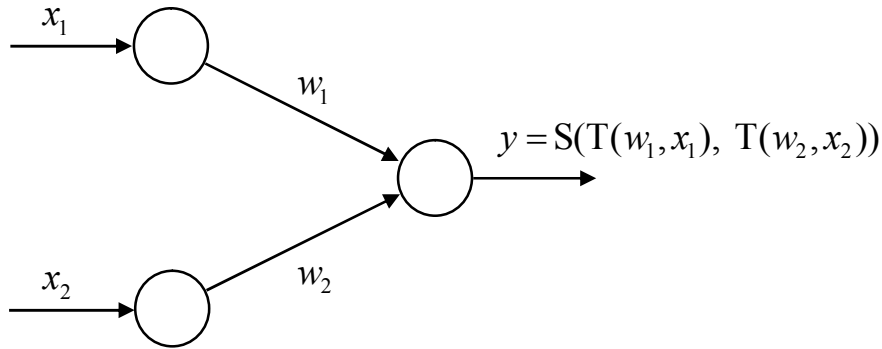


Рис. 6.3. Нечіткий нейрон «АБО»

Якщо прийняти

$$T = \min, \quad S = \max,$$

тоді нечіткий нейрон «АБО» реалізує композицію $\max - \min$:

$$y = \max(w_1 \wedge x_1, w_2 \wedge x_2).$$

6.2 Алгоритми навчання і використання гібридних мереж

Опишемо типовий підхід до побудови алгоритмів навчання і використання гібридних нейронних мереж.

Припустимо, що гібридною мережею повинно бути реалізоване (невідоме) відображення

$$y^k = f(\mathbf{x}^k) = f(x_1^k, x_2^k, \dots, x_n^k), \quad k = 1, 2, \dots, N,$$

за наявності навчальної множини

$$\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \dots, (\mathbf{x}^N, y^N)\}.$$

Для моделювання невідомого відображення f використовуємо раніше розглянутий спрощений алгоритм нечіткого висновку, застосовуючи наступну форму запису предикативних правил:

Π_i : якщо $x_1 \in A_{i1}$, і $x_2 \in A_{i2}$, і ... і $x_n \in A_{in}$, тоді $y \in z_i$, $i = 1, 2, \dots, m$,

де A_{ij} – нечіткі числа трикутної форми;

z_i – дійсні числа, що визначають ступінь істинності i -го правила за допомогою операції множення (Larsen):

$$\alpha_i = \prod_{j=1}^n A_{ij}(x_j^k),$$

(тут можна використовувати і інші уявлення для моделювання логічного оператора «І») і визначаючи вихід o^k нечіткої системи дискретним аналогом центроїдного методу:

$$o^k = \frac{\sum_{i=1}^m \alpha_i z_i}{\sum_{i=1}^m \alpha_i}.$$

Введення функції помилки для k -го пред'явленого зразка виду

$$E_k = \frac{1}{2}(o^k - y^k)^2$$

дозволяє, далі, як в звичайних (стандартних) нейронних мережах використовувати градієнтний метод для підстроювання параметрів заданих предикативних правил. Так, величини z_i можна коректувати по співвідношенню

$$z_i := z_i - \eta \frac{\partial E_k}{\partial z_i} = z_i - \eta(o^k - y^k) \frac{\alpha_i}{\alpha_1 + \alpha_2 + \dots + \alpha_m}, \quad i = 1, 2, \dots, m$$

де η – константа, що характеризує швидкість навчання.

Детальніше алгоритм настройки розглянемо на прикладі системи, що включає два правила:

Π_1 : якщо $x \in A_1$, тоді $y = z_1$,

Π_2 : якщо $x \in A_2$, тоді $y = z_2$,

при цьому передбачається, що нечіткі поняття A_1 і A_2 мають сигмоїдні функції приналежності

$$A_1(x) = \frac{1}{1 + e^{b_1(x-a_1)}}, \quad A_2(x) = \frac{1}{1 + e^{b_2(x-a_2)}},$$

що характеризуються параметрами a_1, a_2, b_1, b_2 .

Ступені істинності правил визначаються в даному випадку співвідношеннями

$$\alpha_1 = A_1(x) = \frac{1}{1 + e^{b_1(x-a_1)}}, \quad \alpha_2 = A_2(x) = \frac{1}{1 + e^{b_2(x-a_2)}},$$

а вихід системи – виразом

$$o = \frac{\alpha_1 z_1 + \alpha_2 z_2}{\alpha_1 + \alpha_2} = \frac{A_1(x)z_1 + A_2(x)z_2}{A_1(x) + A_2(x)}.$$

Припустимо, що є навчальна множина $\{(\mathbf{x}^1, y^1), (\mathbf{x}^2, y^2), \dots, (\mathbf{x}^N, y^N)\}$, що відображає невідому функцію f .

Потрібно: здійснити таку настройку параметрів системи $a_1, a_2, b_1, b_2, z_1, z_2$, при якій забезпечується найкраща апроксимація даної функції.

Рішення. В даному випадку функція помилки може бути записана у формі

$$E_k = E_k(a_1, a_2, b_1, b_2, z_1, z_2) = \frac{1}{2}(o^k(a_1, a_2, b_1, b_2, z_1, z_2) - y^k)^2.$$

Використовуючи далі той же підхід, що і при виведенні алгоритму зворотного розповсюдження помилки, запишемо:

$$z_1 := z_1 - \eta \frac{\partial E_k}{\partial z_1} = z_1 - \eta(o^k - y^k) \frac{\alpha_1}{\alpha_1 + \alpha_2} = z_1 - \eta(o^k - y^k) \frac{A_1(x^k)}{A_1(x^k) + A_2(x^k)},$$

$$z_2 := z_2 - \eta \frac{\partial E_k}{\partial z_2} = z_2 - \eta(o^k - y^k) \frac{\alpha_2}{\alpha_1 + \alpha_2} = z_2 - \eta(o^k - y^k) \frac{A_2(x^k)}{A_1(x^k) + A_2(x^k)}.$$

Аналогічним шляхом можуть бути отримані розгорнені вирази для корекції коефіцієнтів a_1, a_2, b_1, b_2 . Вихідні співвідношення наступні

$$a_1 := a_1 - \eta \frac{\partial E_k}{\partial a_1}, \quad a_2 := a_2 - \eta \frac{\partial E_k}{\partial a_2},$$

$$b_1 := b_1 - \eta \frac{\partial E_k}{\partial b_1}, \quad b_2 := b_2 - \eta \frac{\partial E_k}{\partial b_2}.$$

Кінцеві вирази є достатньо громіздкими, але можуть бути спрощені у випадку, якщо функції приналежності мають вигляд

$$A_1(x) = \frac{1}{1 + e^{-b(x-a)}}, \quad A_2(x) = \frac{1}{1 + e^{b(x-a)}},$$

Дані функції характеризуються всього двома параметрами (a і b), у певному значенні є симетричними (див. рис. 6.4) і задовольняють рівнянню

$$A_1(x) + A_2(x) = 1.$$

Відмітимо, що з останнього і раніше отриманих рівнянь слідує:

$$z_1 := z_1 - \eta \frac{\partial E_k}{\partial z_1} = z_1 - \eta(o^k - y^k) \frac{\alpha_1}{\alpha_1 + \alpha_2} = z_1 - \eta(o^k - y^k) A_1(x^k),$$

$$z_2 := z_2 - \eta \frac{\partial E_k}{\partial z_2} = z_2 - \eta(o^k - y^k) \frac{\alpha_2}{\alpha_1 + \alpha_2} = z_2 - \eta(o^k - y^k) A_2(x^k).$$

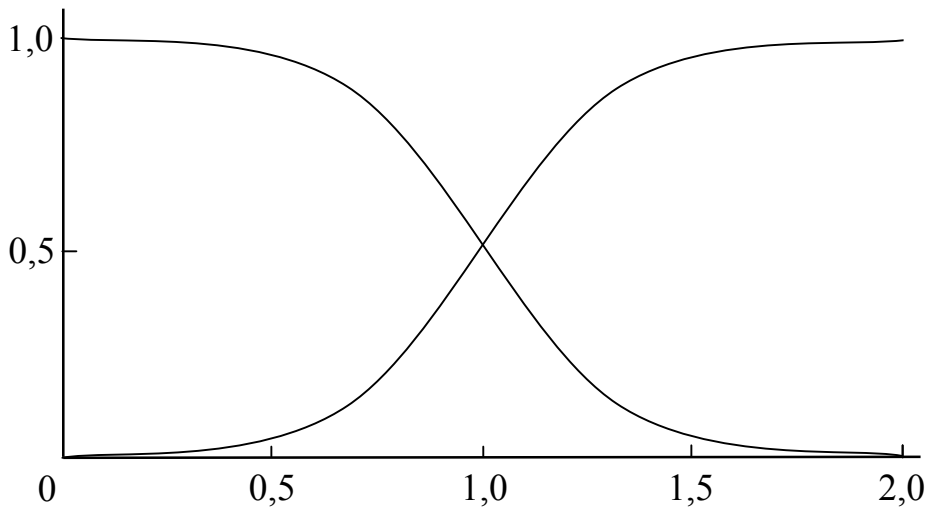


Рис. 6.4. Симетричні функції приналежності

Подальші викладення такі

$$a := a - \eta \frac{\partial E_k(a, b)}{\partial a},$$

де

$$\begin{aligned} \frac{\partial E_k(a, b)}{\partial a} &= (o^k - y^k) \frac{\partial o^k}{\partial a} = (o^k - y^k) \frac{\partial}{\partial a} (z_1 A_1(x^k) + z_2 A_2(x^k)) = \\ &= (o^k - y^k) \frac{\partial}{\partial a} (z_1 A_1(x^k) + z_2 (1 - A_1(x^k))) = \\ &= (o^k - y^k) (z_1 - z_2) \frac{\partial A_1(x^k)}{\partial a} = (o^k - y^k) (z_1 - z_2) b \frac{e^{b(x^k - a)}}{(1 + e^{b(x^k - a)})^2} = \\ &= (o^k - y^k) (z_1 - z_2) b A_1(x^k) (1 - A_1(x^k)) = (o^k - y^k) (z_1 - z_2) b A_1(x^k) A_2(x^k) \end{aligned}$$

$$b := b - \eta \frac{\partial E_k(a, b)}{\partial b},$$

$$\begin{aligned} \frac{\partial E_k(a, b)}{\partial b} &= (o^k - y^k)(z_1 - z_2) \frac{\partial}{\partial b} \frac{1}{1 + e^{b(x^k - a)}} = \\ &= (o^k - y^k)(z_1 - z_2)(x^k - a) A_1(x^k)(1 - A_1(x^k)) = \\ &= (o^k - y^k)(z_1 - z_2)(x^k - a) A_1(x^k) A_2(x^k). \end{aligned}$$

Приведені викладення, як представляється, повністю ілюструють ідеї алгоритмів навчання і використання гібридної мережі.

Іншим прикладом може служити система, що має наступну базу знань:

Π_1 : якщо $x_1 \in L_1$, і $x_2 \in L_2$, і $x_3 \in L_3$ тоді $y \in H$,

Π_2 : якщо $x_1 \in H_1$, і $x_2 \in H_2$, і $x_3 \in L_3$ тоді $y \in M$,

Π_3 : якщо $x_1 \in H_1$, і $x_2 \in H_2$, і $x_3 \in H_3$ тоді $y \in S$,

де x_1, x_2, x_3 – вхідні змінні;

y – вихід системи;

$L_1, L_2, L_3, H_1, H_2, H_3, H, M, S$ – деякі нечіткі множини з функціями приналежності сигмоїдного типу

$$L_j(t) = \frac{1}{1 + e^{b_j(t - c_j)}}, \quad H_j = \frac{1}{1 + e^{b_j(t - c_j)}}, \quad j = 1, 2, 3,$$

$$H(t) = \frac{1}{1 + e^{-b_4(t - c_4 + c_5)}}, \quad M(t) = \frac{1}{1 + e^{-b_4(t - c_4)}}, \quad S(t) = \frac{1}{1 + e^{b_4(t - c_4)}}.$$

Для визначення вихідної змінної використовується алгоритм виведення Tsukamoto (див. вище), тобто:

1) підраховуються значення істинності передумов для кожного правила:

$$\alpha_1 = L_1(a_1) \wedge L_2(a_2) \wedge L_3(a_3),$$

$$\alpha_1 = H_1(a_1) \wedge H_2(a_2) \wedge L_3(a_3),$$

$$\alpha_1 = H_1(a_1) \wedge H_2(a_2) \wedge H_3(a_3),$$

де в даному випадку a_1, a_2, a_3 – поточні значення входів системи;

2) для кожного правила визначаються приватні виходи:

$$z_1 = B^{-1}(\alpha_1) = c_4 + c_5 + \frac{1}{b_4} \ln \frac{1 - \alpha_1}{\alpha_1},$$

$$z_2 = B^{-1}(\alpha_2) = c_4 + \frac{1}{b_4} \ln \frac{1 - \alpha_2}{\alpha_2},$$

$$z_3 = B^{-1}(\alpha_3) = c_4 + \frac{1}{b_4} \ln \frac{1 - \alpha_3}{\alpha_3};$$

3) знаходиться загальний вихід системи:

$$z_0 = \frac{\alpha_1 z_1 + \alpha_2 z_2 + \alpha_3 z_3}{\alpha_1 + \alpha_2 + \alpha_3}.$$

Викладений процес ілюструється рис. 6.5.

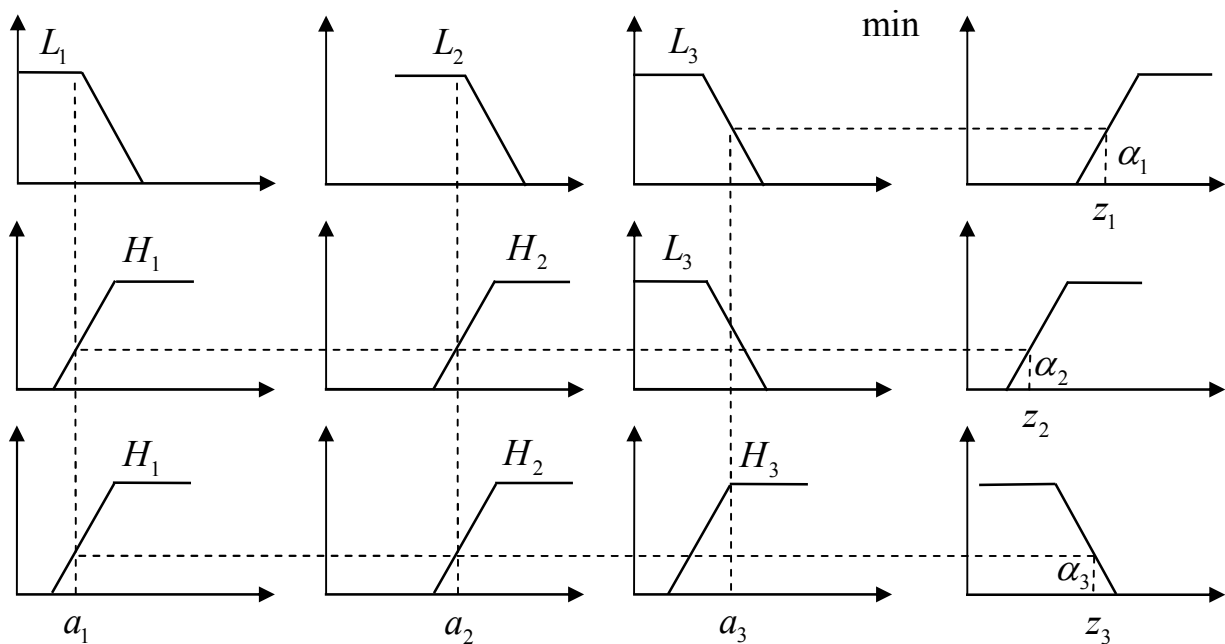


Рис. 6.5. Ілюстрація алгоритму висновку Tsukamoto

Гібридна нейронна мережа, що відображає приведений механізм висновку, представлена на рис. 6.6. Відмітимо, що мережі з подібною архітектурою в англо-мвній літературі отримали назву ANFIS (Adaptive Neuro-Fuzzy Inference System).

Дана мережа може бути описана таким чином.

Шар 1 {Layer 1}. Виходи вузлів цього шару представляють собою значення функцій приналежності при конкретних (заданих) значеннях входів.

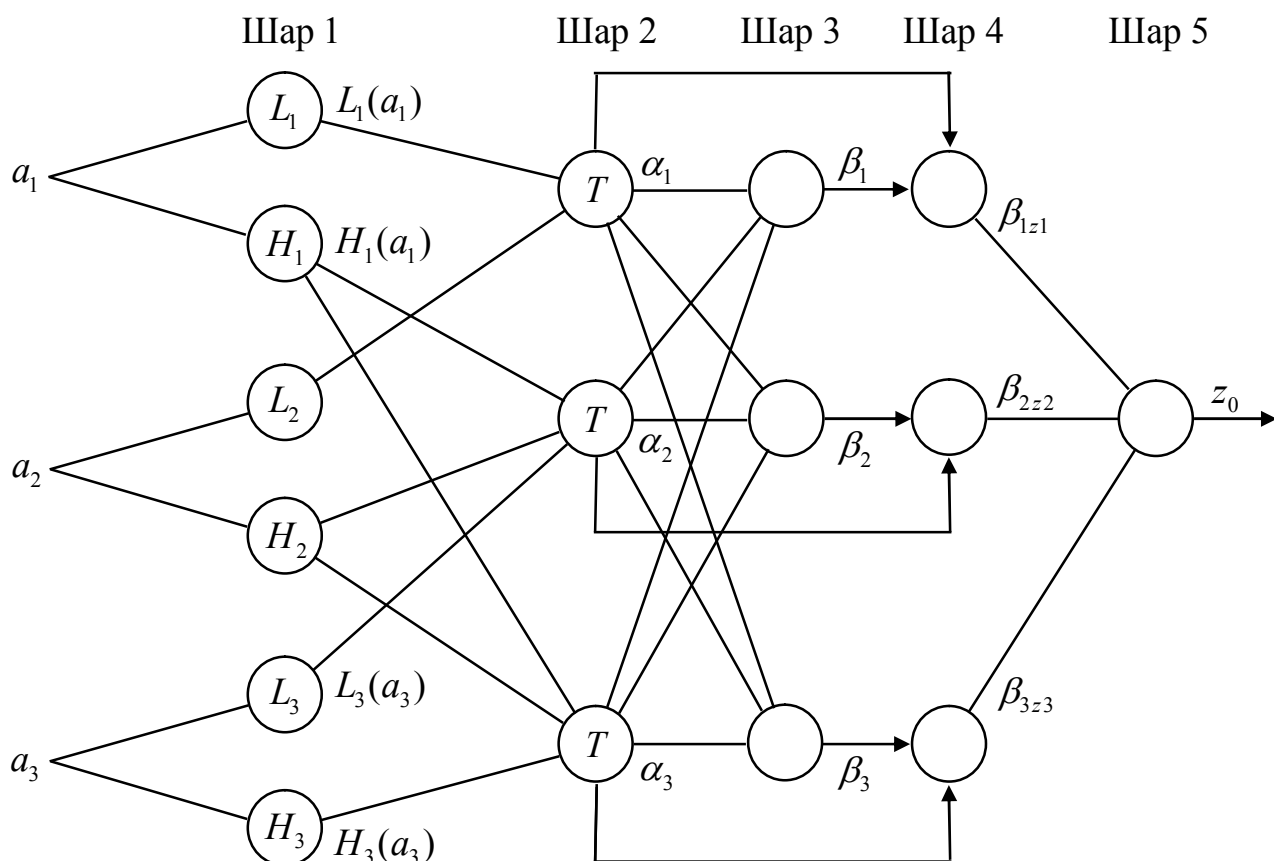


Рис. 6.6. Структура гібридної нейронної мережі (архітектура ANFIS)

Шап 2 {Layer 2}. Виходами нейронів цього шару є ступені істинності передумов кожного правила бази знань системи, що обчислюються за формулами:

$$\alpha_1 = L_1(a_1) \wedge L_2(a_2) \wedge L_3(a_3),$$

$$\alpha_1 = H_1(a_1) \wedge H_2(a_2) \wedge L_3(a_3),$$

$$\alpha_1 = H_1(a_1) \wedge H_2(a_2) \wedge H_3(a_3).$$

Всі нейрони цього шару позначені буквою T , що означає, що вони можуть реалізовувати довільну T -норму для моделювання операції «І».

3 *Шап 3 {Layer 3}*. Нейрони цього шару (позначені буквою N) обчислюють величини:

$$\beta_1 = \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \beta_2 = \frac{\alpha_2}{\alpha_1 + \alpha_2 + \alpha_3}, \quad \beta_3 = \frac{\alpha_3}{\alpha_1 + \alpha_2 + \alpha_3}.$$

4. *Шап 4 (Layer 4)*. Нейрони даного шару виконують операції:

$$\beta_1 z_1 = \beta_1 H^{-1}(\alpha_1), \quad \beta_2 z_2 = \beta_2 M^{-1}(\alpha_2), \quad \beta_3 z_3 = \beta_3 S^{-1}(\alpha_3)$$

5. *Шар 5 (Layer 5)*. Єдиний нейрон цього шару обчислює вихід мережі:

$$z_0 = \beta_1 z_1 + \beta_2 z_2 + \beta_3 z_3$$

Коректування параметрів системи тут проводиться відповідно до раніше розглянутого підходу. Так, наприклад, настройка коефіцієнтів b_4 , c_4 і c_5 – по формулам:

$$b_4 := b_4 - \eta \frac{\partial E_k}{\partial b_4} = b_4 - \frac{\eta}{b_4^2} \delta_k \frac{\alpha_1 + \alpha_2 - \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3},$$

$$c_4 := c_4 - \eta \frac{\partial E_k}{\partial c_4} = c_4 - \eta \delta_k \frac{\alpha_1 + \alpha_2 + \alpha_3}{\alpha_1 + \alpha_2 + \alpha_3} = c_4 + \eta \delta_k,$$

$$c_5 := c_5 - \eta \frac{\partial E_k}{\partial c_5} = c_5 - \eta \delta_k \frac{\alpha_1}{\alpha_1 + \alpha_2 + \alpha_3},$$

де $\delta_k = y^k - o^k$.

Відповідні вирази можуть бути отримані і для решти коефіцієнтів.

6.3 Нечіткий гібридний класифікатор

Розглянемо тепер, як за допомогою гібридної системи вирішується завдання класифікації, тобто віднесення об'єкту, що характеризується набором ознак, до деякого класу.

Одна з можливих структур для вирішення подібного завдання приведена на рис. 6.7.

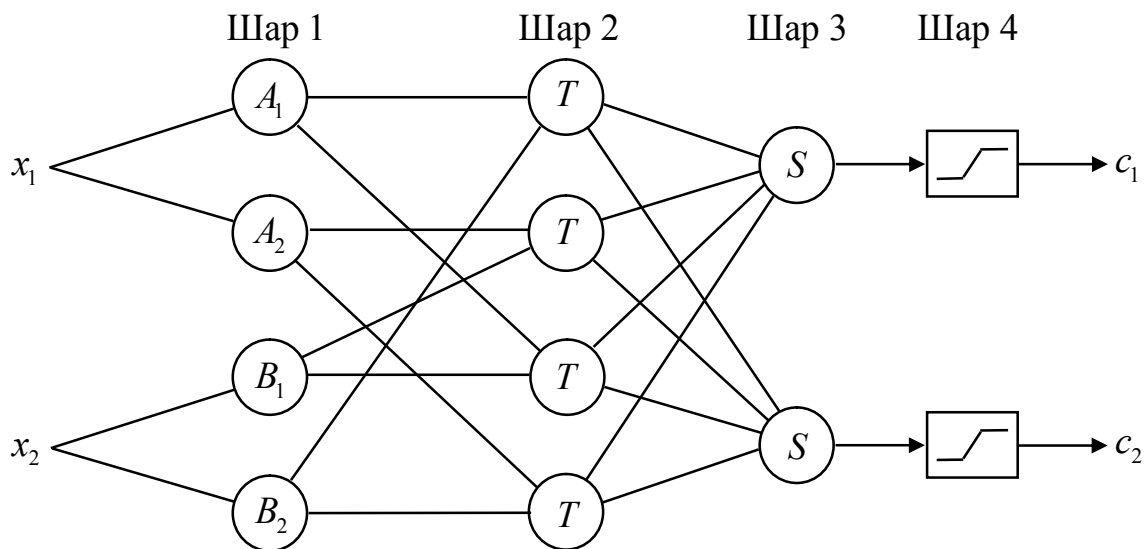


Рис. 6.7. Гібридна мережа для вирішення завдання класифікації

Передбачається, що тут об'єкт характеризується двома кількісними ознаками x_1 і x_2 і відноситься до одного з двох класів – c_1 або c_2 . Кожен вхід представляється двома лінгвістичними поняттями, що дозволяє обмежитися всього чотирма правилами.

Мережа може бути описана таким чином.

1. *Шар 1 (Layer 1)*. Виходи вузлів даного шару – це ступені приналежності вхідних змінних визначенням для них нечітким множинам A_1, A_2, B_1, B_2 .

В даному випадку вибрані функції приналежності колоколоподібного вигляду

$$A_i(t) = \exp(-(t - a_{i1} / b_{i1})^2 / 2),$$

$$B_i(t) = \exp(-(t - a_{i2} / b_{i2})^2 / 2),$$

з набором параметрів $a_{i1}, a_{i1}, a_{i1}, a_{i1}$.

Значення даних параметрів коректуються в процесі навчання мережі (заснованому на градієнтному методі).

2. *Шар 2 (Layer 2)*. Кожен нейрон цього шару є нейроном типу розглянутого вище гібридного (нечіткого) нейрона «І».

3. *Шар 3 (Layer 3)*. Нейрони даного шару є звичайними (стандартними) нейронами, входами яких є лінійні (зважені) комбінації виходів нейронів попереднього шару, а виходи формуються з використанням активаційних функцій сигмоїдного типу. Ці виходи трактуються як ступені приналежності пред'явленого об'єкту першому або другому класу.

Алгоритм навчання даної мережі, в принципі, не відрізняється від розглянутих.

Питання для самоконтролю.

1. Дайте визначення гібридної мережі.
2. Наведіть структуру і основні рівняння гібридного нейрона «І» і «АБО».
3. Поясніть алгоритм навчання гібридних мереж.
4. Наведіть приклади, що ілюструють ідеї алгоритмів навчання і використання гібридних мереж.
5. Проілюструйте алгоритму висновку Tsukamoto.
6. Наведіть приклад структурної схеми гібридної нейронної мережі.
7. Поясніть призначення і структуру нечіткого гібридного класифікатора.
8. Зобразіть структуру гібридної нейронної мережі для вирішення завдання класифікації.

СИНТЕЗ НЕЧІТКИХ РЕГУЛЯТОРІВ З ВИКОРИСТАННЯМ СИСТЕМИ MATLAB

7.1 Призначення і можливості пакету Fuzzy Logic Toolbox

Пакет нечіткої логіки Fuzzy Logic Toolbox – це пакет прикладних програм, що відносяться до теорії розмитих або нечітких множин і дозволяє конструювати так звані нечіткі експертні і управляючі системи [74-81].

Основні можливості пакету:

- побудова систем нечіткого висновку (експертних систем, регуляторів, аппроксиматоров залежностей);
- побудова адаптивних нечітких систем (гібридних нейронних мереж);
- інтерактивне динамічне моделювання систем з нечіткою логікою в середовищі пакету блокового моделювання Simulink.

Пакет забезпечує роботу:

- у режимі командного рядка;
- у вікнах графічного інтерфейсу;
- у середовищі пакету розширення Simulink.

Пакет Fuzzy Logic Toolbox може працювати в операційному середовищі системи MATLAB. Спільно з Fuzzy Logic Toolbox можуть використовуватися і інші пакети розширення, зокрема, пакет Simulink, що значно підвищує наочність рішень.

7.2 Графічний інтерфейс Fuzzy Logic Toolbox

Більшість додатків системи MATLAB мають найбільш наочну реалізацію при використанні засобів графічного інтерфейсу користувача GUI. До складу програмних засобів Fuzzy Logic Toolbox входять наступні основні програми, що дозволяють працювати в режимі GUI:

- редактор нечіткої системи виведення **Fuzzy Inference System Editor (FIS Editor** або **FIS-редактор**) разом з допоміжними програмами – редактором функцій приналежності (**Membership Function Editor**), редактором правил (**Rule Editor**), переглядачем правил (**Rule Viewer**) і переглядачем поверхні нечіткого висновку (**Surface Viewer**);

- редактор гібридних систем (**ANFIS Editor, ANFIS-редактор**);
- програма знаходження центрів кластерів (програма **Clustering** – кластеризація).

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

7.3 Розробка моделі системи управління з нечітким регулятором

Синтез нечітких регуляторів розглянемо на прикладі управління двомасовою електромеханічною системою механізму підйому мостового крану. У якості привідного двигуна застосовано двигун постійного струму незалежного збудження. Якірна обмотка живиться від тиристорного підсилювача потужності. Систем управління електроприводом побудована за принципом підлеглого регулювання і містить два контури: контур регулювання струму і контур регулювання швидкості із зворотним зв'язком по ЕДС двигуна. Замкнений контур струму настроєний на модульний оптимум, а контур ЕДС – на симетричний оптимум.

Система диференціальних рівнянь двомасової системи підлеглого регулювання швидкості з внутрішнім контуром струму і зовнішнім контуром ЕДС має вид:

$$\left\{ \begin{array}{l} \frac{d\omega_m}{dt} = \frac{1}{J_m} M_{np} + \frac{\beta}{J_m} \omega_d - \frac{\beta}{J_m} \omega_m - \frac{1}{J_m} M_{ct}; \frac{dM_{np}}{dt} = c_{12} \omega_d - c_{12} \omega_m; \\ \frac{d\omega_d}{dt} = \frac{1}{J_{d\Sigma}} M_d - \frac{1}{J_{d\Sigma}} M_{np} - \frac{\beta}{J_{d\Sigma}} \omega_d - \frac{\beta}{J_{d\Sigma}} \omega_m; \\ \frac{dM_d}{dt} = \frac{1}{k_d} P; \\ \frac{dP}{dt} = \frac{k_{pe}}{2T_{\mu T}^2 k_T} E_{3e} - \frac{k_{pe}}{2T_{\mu T}^2 k_T} U_{33e} - \frac{k_d}{2T_{\mu T}^2} M_d - \frac{1}{T_{\mu T}} P + \frac{1}{T_{\mu T}^2 k_T} U_{ie}; \\ \frac{dE_{3e}}{dt} = \frac{1}{T_a} U_{3e} - \frac{1}{T_a} E_{3e}; \\ \frac{dU_{33e}}{dt} = \frac{k_n}{T_a k_d} \omega_d - \frac{1}{T_a} U_{33e}; \\ \frac{dU_{ie}}{dt} = k_{ie} E_{3e} - k_{ie} U_{33e}. \end{array} \right.$$

де $\omega_d(t)$, $\omega_m(t)$ – кутові швидкості двигуна і механізму;

$M_d(t)$ – обертаючий момент двигуна;

$M_c(t)$ – момент статичного навантаження;

$M_{np}(t)$ – момент пружності;

$P(t) = dM_d(t) / dt$ – ривок;

$U_{зе}(t)$, $E_{зе}(t)$ – напруга і ЕДС завдання;
 $U_{ззе}(t)$ – напруга зворотного зв'язку по ЕДС;
 $U_{іе}(t)$ – вихідна напруга інтегратора ПІ регулятора ЕДС;
 $T_{\mu\tau}$ – мала некомпенсуєма постійна часу контуру струму;
 T_a – електромеханічна постійна часу електродвигуна;
 k_d – коефіцієнт посилення двигуна;
 k_{τ} – коефіцієнт посилення зворотного зв'язку по току;
 k_n – коефіцієнт посилення зворотного зв'язку по напрузі;
 $k_{пе}$, $k_{іе}$ – коефіцієнти підсилення пропорційної і інтегральної частин ПІ регулятора контура ЕДС;
 β – коефіцієнт внутрішнього в'язкого тертя;
 c_{12} – коефіцієнт жорсткості передачі;
 $J_{д\Sigma}$ – сумарний момент інерції двигуна і жорстко пов'язаних з ним елементів системи;
 J_m – момент інерції механізму.

Алгоритмічна схема двомасової системи зображена на рис.7.1.

Визначимо мету управління як забезпечення задовільних показників якості функціонування двомасової електромеханічної системи. Будемо оцінювати якість функціонування системи по величині миттєвої помилки регулювання $\varepsilon(t)$, тобто різниці між заданим значенням швидкості обертання механізму $\omega_z(t)$ і реальною швидкістю обертання механізму $\omega_m(t)$. Очевидно, в регулятор, що забезпечує досягнення мети управління, повинна поступати інформація про величину помилки регулювання, при цьому регулятор повинен виробляти управляючий сигнал $U_{рег}(t)$, що подається на двомасову електромеханічну систему.

Оскільки двомасова система є інерційною, то при регулюванні слід враховувати похідну від помилки регулювання, в якості якої можна використовувати момент пружності $M_{пр}(t)$ в кінематичному ланцюзі електроприводу. Тобто, на вхід регулятора слід подавати також сигнал $M_{пр}(t)$.

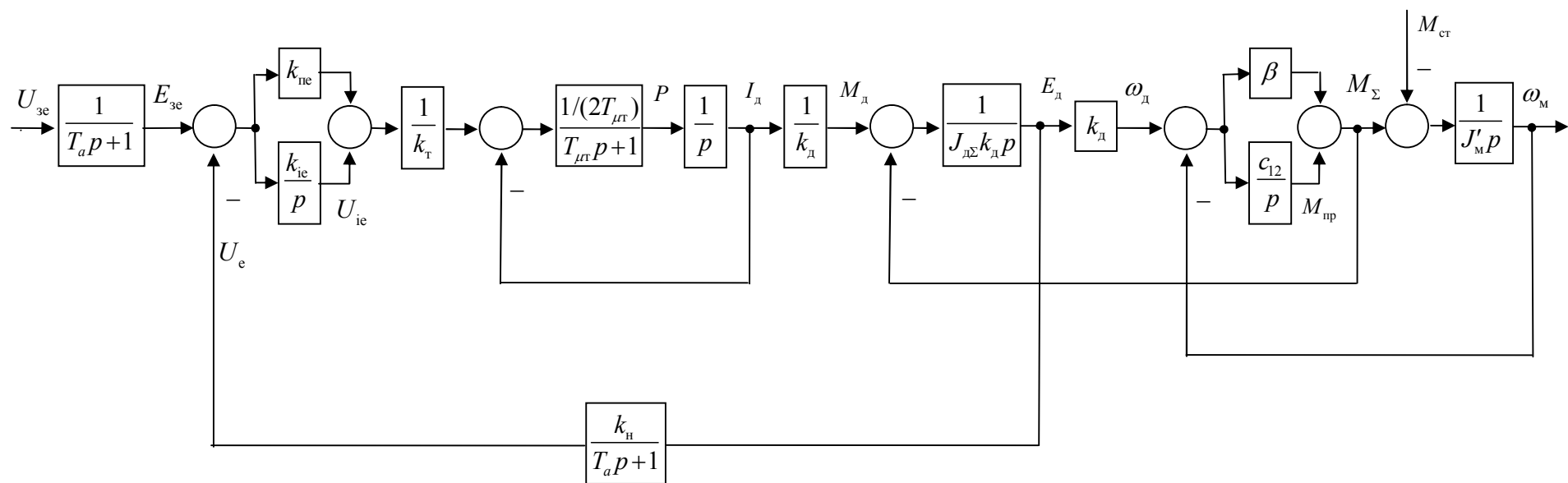


Рис. 7.1. Алгоритмічна схема двомасової системи

Системи нечіткого висновку, створені за допомогою пакету Fuzzy Logic Toolbox, допускають інтерпретацію з інструментальними засобами Simulink. Модель системи управління двомасовою електромеханічною системою з нечітким регулятором, побудована в Simulink показана на рис. 7.2. Схема містить блок **Fuzzy Logic Controller**, який співставляється з системою нечіткого висновку, записаною у файлі Kran.fis (порядок формування системи нечіткого висновку наведено далі) і підсистеми Subsystem, тобто моделі двомасової електромеханічної системи (рис. 7.3). Дана модель з'являється у відповідному вікні при активізації блоку Subsystem подвійним клацанням лівої кнопки миші.

Як зазначалося, на входи нечіткого регулятора подається помилка регулювання $\varepsilon(t)$ і момент пружності $M_{\text{пр}}(t)$ (точніше – різниця між моментом пружності $M_{\text{пр}}(t)$ і моментом статичного навантаження $M_c(t)$, тобто $M_{\text{пр}}(t) - M_c(t)$). За допомогою регулятора формується управляюча дія $U_{\text{пер}}(t)$, що алгебраїчно підсумовується з напругою завдання $U_3(t)$ і визначається напруга $U_{31}(t)$, що діє на двомасову систему.

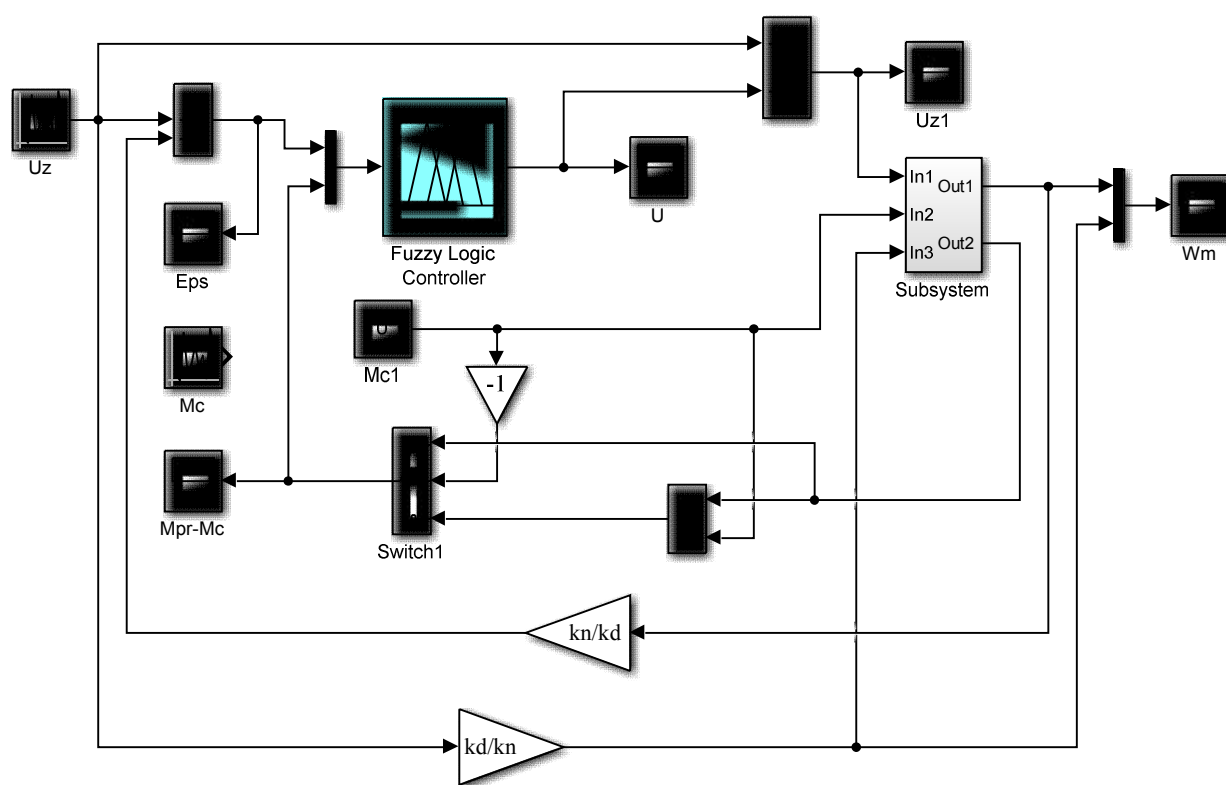


Рис. 7.2. Модель системи управління з нечітким регулятором

Виконаємо розробку системи нечіткого висновку в інтерактивному режимі за допомогою графічних засобів пакету Fuzzy Logic Toolbox.

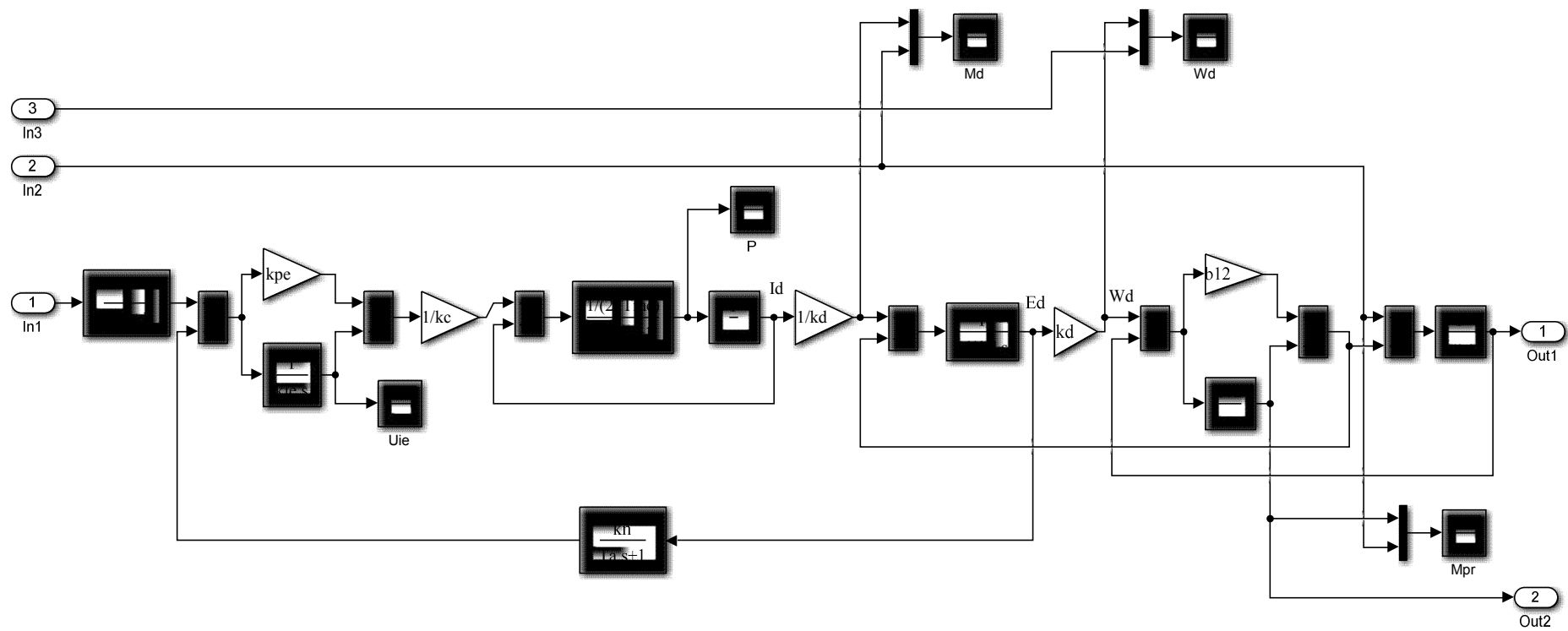


Рис. 7.3 – Модель керованого об'єкту нечіткого регулятора

7.4 Формування бази правил системи нечіткого висновку

Для формування бази правил системи нечіткого висновку необхідно заздалегідь визначити вхідні і вихідні лінгвістичні змінні. Очевидно, в даному випадку вхідні лінгвістичні змінні – це помилка регулювання $\varepsilon(t)$: ε – "помилка регулювання", і момент пружності $M_{\text{пр}}(t)$: $M_{\text{пр}}$ – "момент пружності". Вихідна лінгвістична змінна – напруга на виході регулятора $U_{\text{рег}}(t)$: $U_{\text{рег}}$ – "напруга регулятора".

Як терм-множини першої лінгвістичної змінної – помилка регулювання $\varepsilon(t)$ – використовуватимемо множину $T_1 = \{\text{"негативна велика помилка", "негативна мала помилка", "близька до нуля помилка", "позитивна мала помилка", "позитивна велика помилка"}\}$, яку можна скорочено записати в символічному виді $T_1 = \{\text{NGE, NSE, ZRE, PSE, PGE}\}$. В якості терм-множини другої лінгвістичної змінної – моменту пружності $M_{\text{пр}}(t)$ – використовуватимемо множину $T_2 = \{\text{"негативний великий момент", "негативний малий момент", "близький до нуля момент", "позитивний малий момент", "позитивний великий момент"}\}$, яка записується в символічному виді $T_2 = \{\text{NGM, NSM, ZRM, PSM, PGM}\}$. В якості терм-множини вихідної лінгвістичної змінної – напруги на виході регулятора $U_{\text{рег}}(t)$ – використовуватимемо множину: $T_3 = \{\text{"негативна велика напруга", "негативна мала напруга", "близька до нуля напруга", "позитивна мала напруга", "позитивна велика напруга"}\}$, яка записується в символічному виді $T_3 = \{\text{NGU, NSU, ZRU, PSU, PGU}\}$.

Функції приналежності кожного терма кожної множини задаються далі (див. рис. 7.6, рис. 7.8 і рис. 7.9).

На підставі аналізу експериментальних перехідних процесів при пуску і набросі навантаження сформуємо базу правил системи нечіткого висновку. Оскільки розглядаються дві вхідні лінгвістичні змінні, кожна з яких має 5 термів, то система нечіткого висновку міститиме 25 правил нечітких висновків наступного вигляду:

- ПРАВИЛО_1: ЯКЩО " $\varepsilon \in \text{NGE}$ " І " $M_{\text{пр}} \in \text{NGM}$ " ТО " $U_{\text{рег}} \in \text{PGU}$ "
ПРАВИЛО_2: ЯКЩО " $\varepsilon \in \text{NGE}$ " І " $M_{\text{пр}} \in \text{NSM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "
ПРАВИЛО_3: ЯКЩО " $\varepsilon \in \text{NGE}$ " І " $M_{\text{пр}} \in \text{ZRM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "
ПРАВИЛО_4: ЯКЩО " $\varepsilon \in \text{NGE}$ " І " $M_{\text{пр}} \in \text{PSM}$ " ТО " $U_{\text{рег}} \in \text{ZRU}$ "
ПРАВИЛО_5: ЯКЩО " $\varepsilon \in \text{NGE}$ " І " $M_{\text{пр}} \in \text{PGM}$ " ТО " $U_{\text{рег}} \in \text{NSU}$ "
ПРАВИЛО_6: ЯКЩО " $\varepsilon \in \text{NSE}$ " І " $M_{\text{пр}} \in \text{NGM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "
ПРАВИЛО_7: ЯКЩО " $\varepsilon \in \text{NSE}$ " І " $M_{\text{пр}} \in \text{NSM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "
ПРАВИЛО_8: ЯКЩО " $\varepsilon \in \text{NSE}$ " І " $M_{\text{пр}} \in \text{ZRM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "
ПРАВИЛО_9: ЯКЩО " $\varepsilon \in \text{NSE}$ " І " $M_{\text{пр}} \in \text{PSM}$ " ТО " $U_{\text{рег}} \in \text{ZRU}$ "
ПРАВИЛО_10: ЯКЩО " $\varepsilon \in \text{NSE}$ " І " $M_{\text{пр}} \in \text{PGM}$ " ТО " $U_{\text{рег}} \in \text{NSU}$ "
ПРАВИЛО_11: ЯКЩО " $\varepsilon \in \text{ZRE}$ " І " $M_{\text{пр}} \in \text{NGM}$ " ТО " $U_{\text{рег}} \in \text{PSU}$ "

ПРАВИЛО_12: ЯКЩО " $\varepsilon \in ZRE$ " І " $M_{np} \in NSM$ " ТО " $U_{per} \in ZRU$ "
 ПРАВИЛО_13: ЯКЩО " $\varepsilon \in ZRE$ " І " $M_{np} \in ZRM$ " ТО " $U_{per} \in ZRU$ "
 ПРАВИЛО_14: ЯКЩО " $\varepsilon \in ZRE$ " І " $M_{np} \in PSM$ " ТО " $U_{per} \in ZRU$ "
 ПРАВИЛО_15: ЯКЩО " $\varepsilon \in ZRE$ " І " $M_{np} \in PGM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_16: ЯКЩО " $\varepsilon \in PSE$ " І " $M_{np} \in NGM$ " ТО " $U_{per} \in PSU$ "
 ПРАВИЛО_17: ЯКЩО " $\varepsilon \in PSE$ " І " $M_{np} \in NSM$ " ТО " $U_{per} \in ZRU$ "
 ПРАВИЛО_18: ЯКЩО " $\varepsilon \in PSE$ " І " $M_{np} \in ZRM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_19: ЯКЩО " $\varepsilon \in PSE$ " І " $M_{np} \in PSM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_20: ЯКЩО " $\varepsilon \in PSE$ " І " $M_{np} \in PGM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_21: ЯКЩО " $\varepsilon \in PGE$ " І " $M_{np} \in NGM$ " ТО " $U_{per} \in PSU$ "
 ПРАВИЛО_22: ЯКЩО " $\varepsilon \in PGE$ " І " $M_{np} \in NSM$ " ТО " $U_{per} \in ZRU$ "
 ПРАВИЛО_23: ЯКЩО " $\varepsilon \in PGE$ " І " $M_{np} \in ZRM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_24: ЯКЩО " $\varepsilon \in PGE$ " І " $M_{np} \in PSM$ " ТО " $U_{per} \in NSU$ "
 ПРАВИЛО_25: ЯКЩО " $\varepsilon \in PGE$ " І " $M_{np} \in PGM$ " ТО " $U_{per} \in NGU$ "

Базу правил системи нечіткого висновку представимо для зручності у вигляді таблиці 7.1.

Таблиця 7.1

База правил системи нечіткого висновку

		Помилка регулювання				
		NGW	NSW	ZRW	PSW	PGW
Момент пружності	NGM	PGU	PSU	PSU	PSU	PSU
	NSM	PSU	PSU	ZRU	ZRU	ZRU
	ZRM	PSU	PSU	ZRU	NSU	NSU
	PSM	ZRU	ZRU	ZRU	NSU	NSU
	PGM	NSU	NSU	NSU	NSU	NGU

Представленої інформації досить для проектування нечіткої системи. Така система матиме 2 входи, один вихід, двадцять п'ять правил типу "якщо... то" і по п'ять функцій приналежності входів і виходу. Побудуємо дану систему, використовуючи алгоритм виведення Mamdani.

7.5 Розробка системи нечіткого висновку

Процес розробки системи нечіткого висновку в інтерактивному режимі полягає у виконанні наступної послідовності дій.

1. Викликати редактор систем нечіткого виведення **FIS**, для чого у вікні команд набрати ім'я відповідної функції fuzzy. Після виконання цієї команди на екрані з'явиться графічний інтерфейс редактора **FIS** з ім'ям системи нечіткого виведення **Untitled** і типом системи нечіткого висновку (Мамдані), запропонованими за умовчанням (рис. 7.4).

2. Оскільки в прикладі, що розглядається система нечіткого висновку з двома входами, необхідно додати в систему **FIS**, що розробляється, ще одну вхідну змінну. Для цього слід виконати команду меню **Edi\Add Variable...\Input**. В результаті виконання цієї команди на діаграмі системи нечіткого висновку з'явиться новий жовтий прямокутник з ім'ям другої вхідної змінної: **input2**.

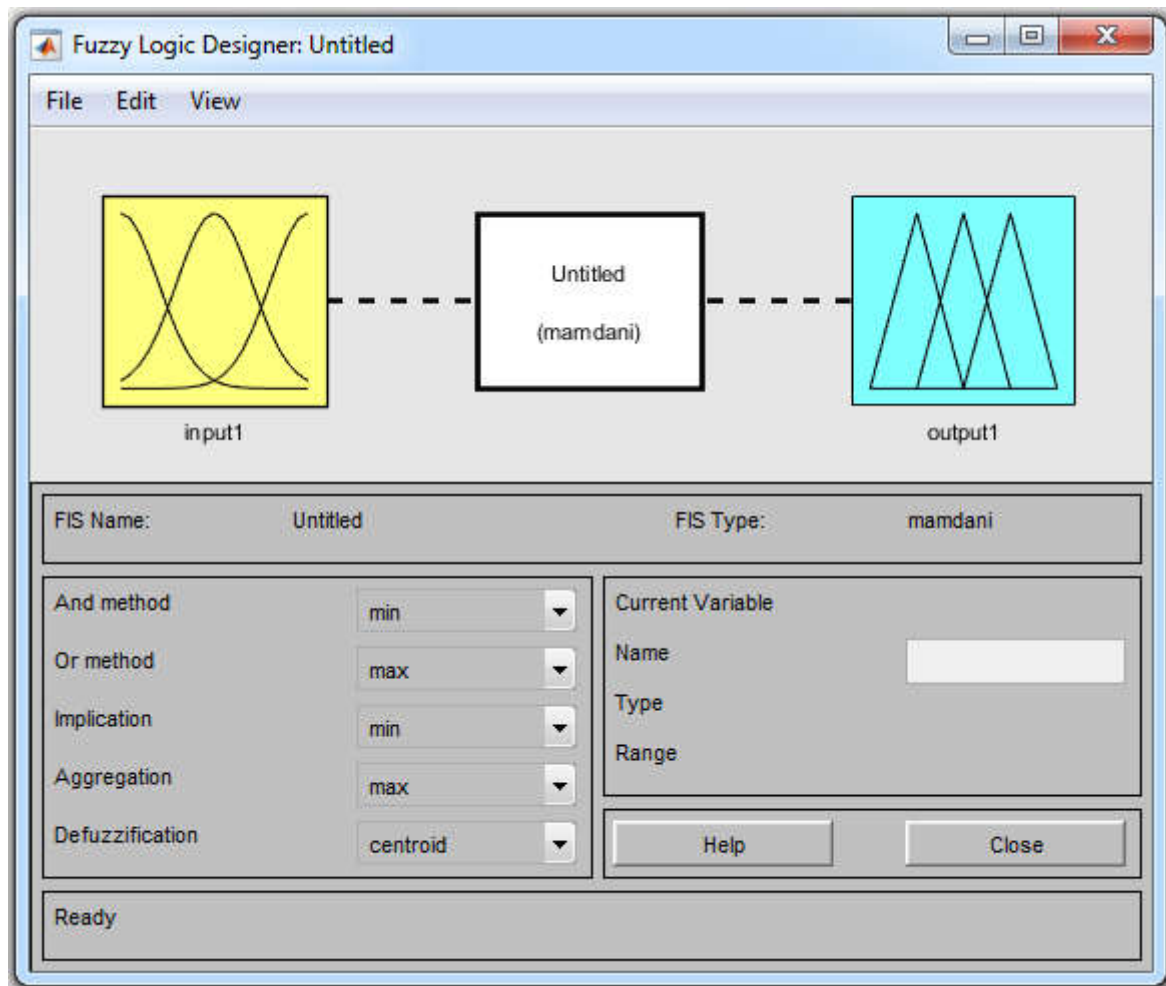


Рис. 7.4. Вид редактора **FIS Editor** після додавання другої вхідної змінної

3. Змінимо імена вхідних і вихідних змінних, запропонованих системою MATLAB за умовчанням. Для цього необхідно виділити прямокутник з ім'ям відповідної змінної, виконавши клацання на його зображенні на діаграмі (сторони виділеного прямокутника мають червоний колір). Після чого слід набрати нове ім'я змінної в полі введення **Name** в правій частині редактора **FIS**. В прикладі, що розглядається, роблячи одноразове клацання на блоку input 1, міняємо його ім'я на «Epsilon» (помилка регулювання $\varepsilon(t)$), завершуючи введення нового імені натисненням клавіші Enter. Аналогічним чином встановлюємо ім'я «Mpr» ($M_{\text{пр}}(t)$) блоку input2 і «U» ($U(t)$) – вихідному блоку (справа вгорі) output 1. Вид вікна редактора після вказаних дій приведений на рис. 7.5.

Залишимо без зміни запропоновані системою MATLAB за умовчанням: метод нечіткого логічного **I (And method)** – значення "min", метод нечіткого логічного **АБО (Or method)** – значення "max", метод імплікації (**Implication**) – значення "min", метод агрегації (**Aggregation**) – значення "max" ' і метод дефазифікації (**Defuzzification**) – значення "centroid". Очевидно, ці значення можуть бути змінені при необхідності.

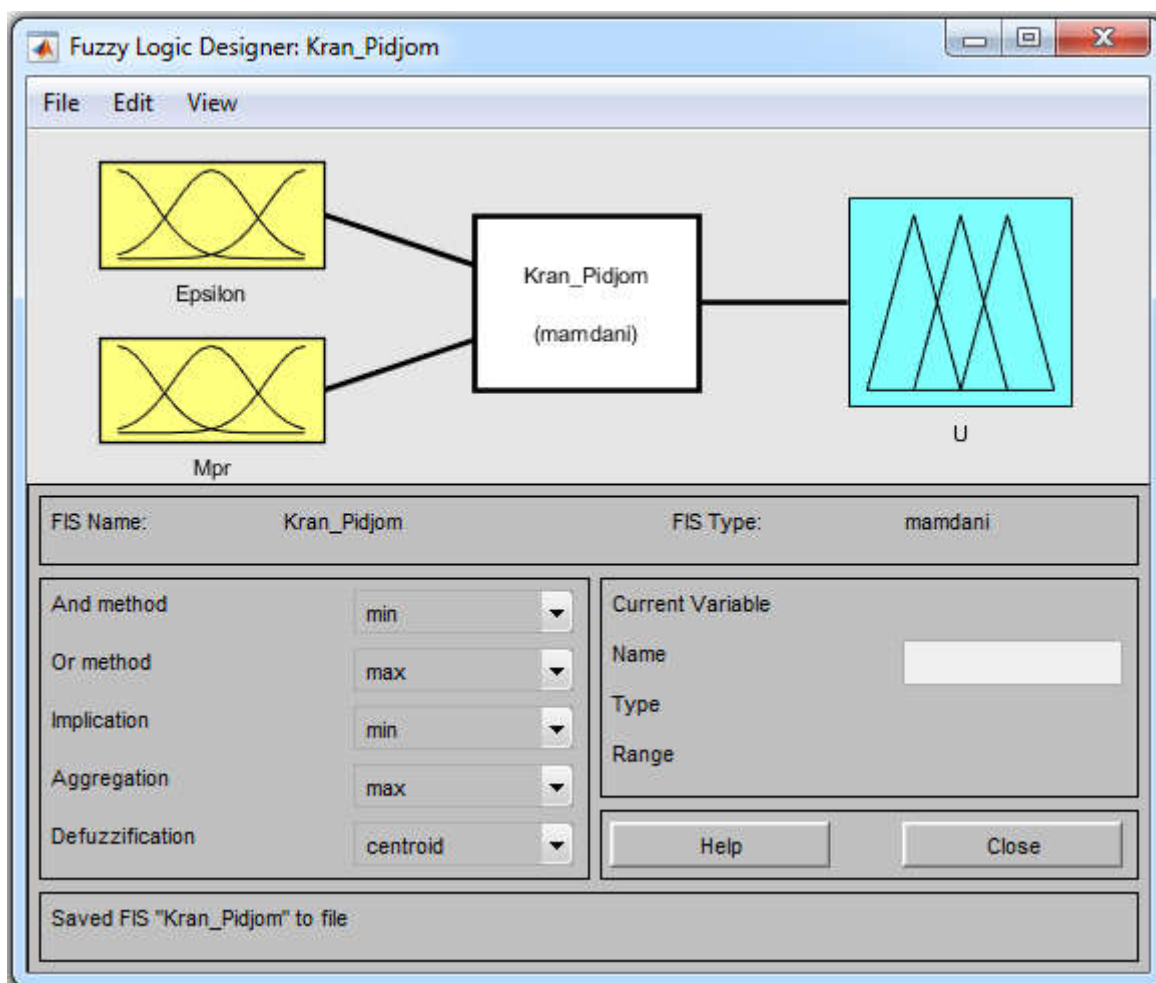


Рис. 7.5. Вид вікна **FIS Editor** після завдання структури системи

4. Змінимо ім'я системи нечіткого висновку (Untitled), запропоноване за умовчанням. Для цього збережемо створювану структуру **FIS** з ім'ям **Kran_Pidjom.fis**, в робочому просторі командою меню **File\Export\To Workspace...**, або на диску, виконавши команду меню **File\Export\To Disk...**. При цьому буде викликано стандартне діалогове вікно збереження файлу, в якому користувачеві пропонується ввести ім'я відповідного файлу (розширення файлу приписується автоматично).

5. Задамо тепер терми і їх функції приналежності для вхідних і вихідних змінних системи нечіткого висновку. Вкажемо, що програму – редактор функцій приналежності – можна відкрити трьома способами:

- через пункт меню **Edit\Membership functions**;
- подвійним клацанням на значку, що відображає відповідну змінну;
- натисненням клавіш **Ctrl+2**.

Будь-яким з приведених способів перейдемо до даної програми.

Завдання і редагування функцій приналежності почнемо із змінної «Epsilon» (рис. 7.6).

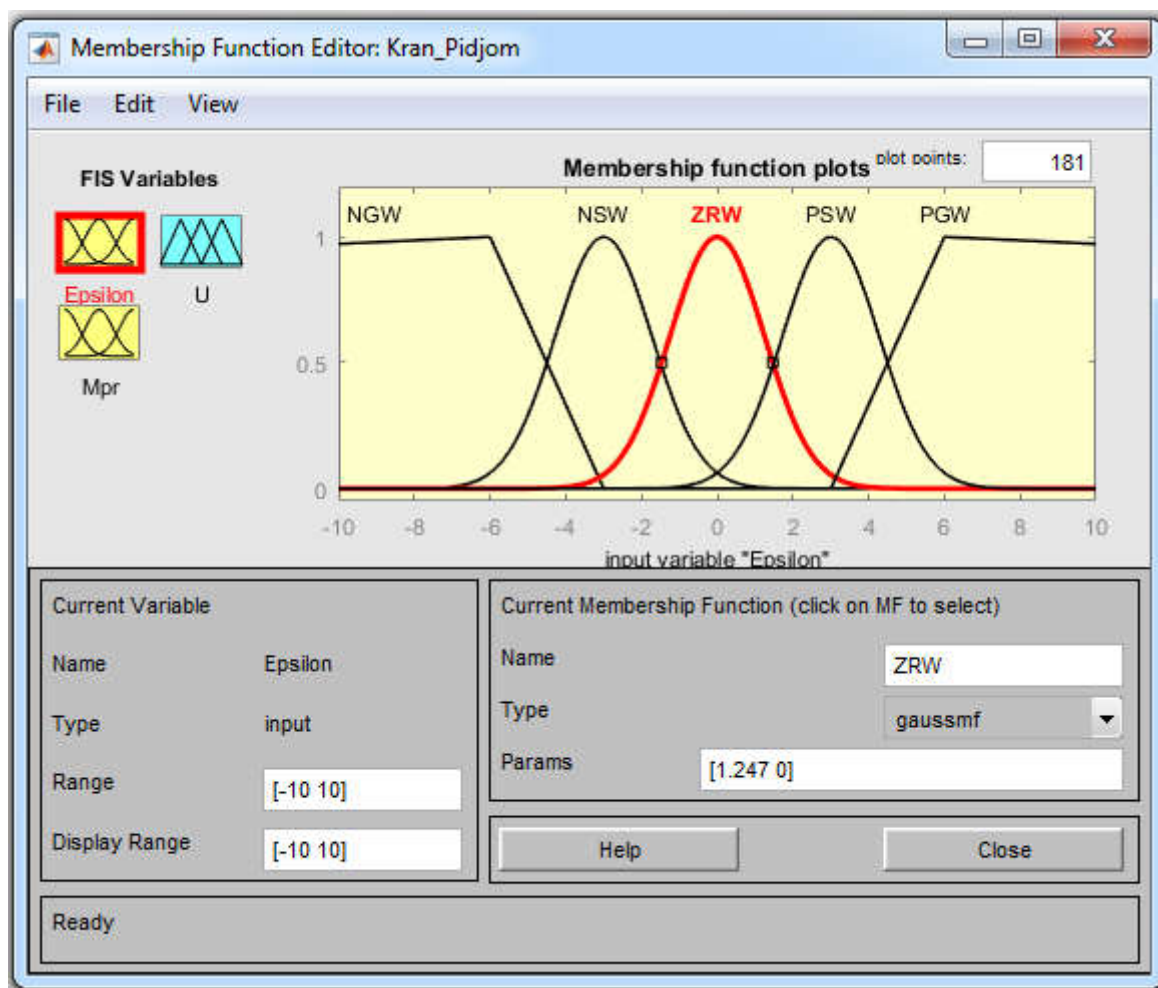


Рис. 7.6. Функції приналежності змінної «Epsilon» (помилка регулювання $\varepsilon(t)$)

Спочатку в полях **Range i Display Range** встановимо діапазон зміни і відображення цієї змінної від -10 до $+10$, підтверджуючи введення натисненням клавіші **Enter**. Потім через пункти меню **Edit\Remove All MFs**, **Edit\Add MFs** перейдемо до діалогового вікна виду рис. 7.7 і задамо в ньому п'ять функцій приналежності трикутної форми (trimf) (або гауссова типу (gaussmf)).

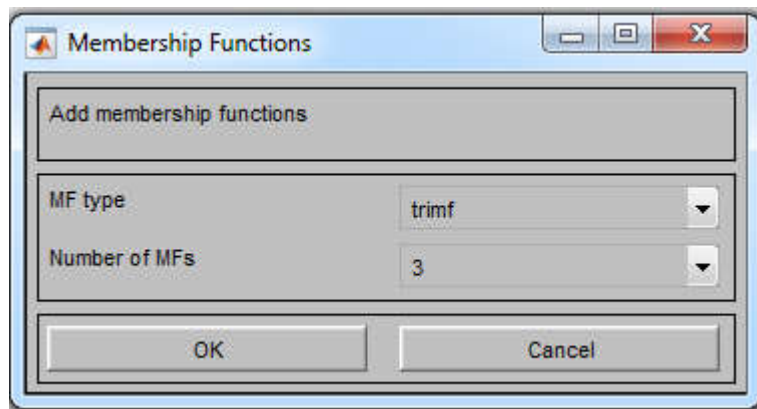


Рис. 7.7. Вид вікна завдання типа і кількості функцій приналежності

Натиснемо кнопку **OK** і повернемося у вікно редактора функцій приналежності. Замінімо імена заданих функцій, для чого підводимо покажчик до потрібної кривої і клацаємо лівою кнопкою миші. Крива вибирається, забарвлюючись в червоний колір, після чого за допомогою миші її можна посунути в потрібну сторону (точнішу установку можна провести, змінюючи числові значення в полі **Params** (Параметри) – в даному випадку кожній функції приналежності відповідають два параметри, при цьому перший визначає розмах кривої, а другий – положення її центру). У полі **Name** змінимо ім'я для вибраної кривої (завершуючи введення кожного імені натисненням клавіші **Enter**). Задамо всім п'яти кривим нові імена, наприклад, так:

- найлівіший – NGE (негативна велика для помилки);
- наступній – NSE (негативна мала для помилки);
- центральний – ZRE (дуже мала для помилки);
- наступній за нею справа – PSE (позитивна мала для помилки);
- найправіший – PGE (позитивна велика для помилки).

Тип, розмах і положення заданих функцій вибирається в процесі настройки Fuzzy регулятора шляхом багатократного розрахунку перехідних процесів багатомасової системи при різних типах і параметрах функцій приналежності і вибору варіанту, при якому показники якості перехідних процесів мають найкращі значення.

Клацанням на значку «Mpr» ($M_{mp}(t)$) увійдемо до вікна редагування функцій приналежності для цієї змінної (рис.7.8). Задамо спочатку діапазон її зміни від -60 до $+60$, а потім, поступаючи як раніше, задамо п'ять функцій приналежності трикутної форми (trimf) з іменами: «NGM» («негативна велика для моменту»), «NSM» («негативна мала для моменту»), ZRM («дуже мала для моменту»), «PSM» («позитивна мала для моменту») і «PGM» («позитивна велика для моменту»).

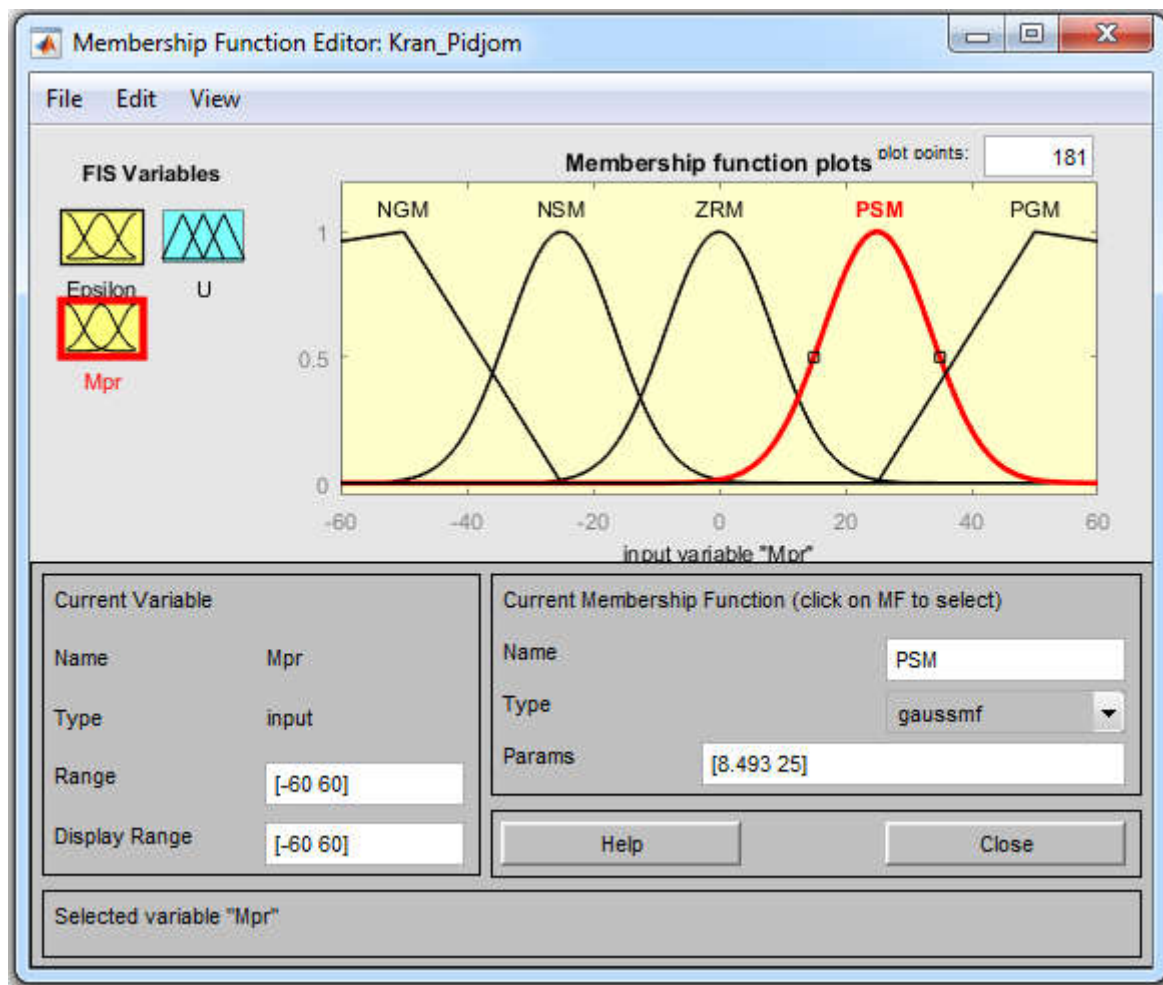


Рис. 7.8. Функції приналежності змінної «Mpr» (моменту пружності $M_{\text{пр}}(t)$)

Для вихідної змінної U ($U(t)$) вкажемо спочатку діапазон зміни (від -5 до $+5$), потім задамо п'ять функцій приналежності гауссова типу: для центральної функції `gauss2mf`, а для інших – `gaussmf` з іменами «NGU» («негативна велика для вихідної змінної $U(t)$ »), «NSU» («негативна мала для вихідної змінної $U(t)$ »), ZRU («дуже мала для вихідної змінної $U(t)$ »), «PSU» («позитивна мала для вихідної змінної $U(t)$ ») і «PGU» («позитивна велика для вихідної змінної $U(t)$ ») так, як це представлено на рис. 7.9.

Як указувалося раніше, тип і параметри функцій уточнюються в процесі настройки Fuzzy регулятора.

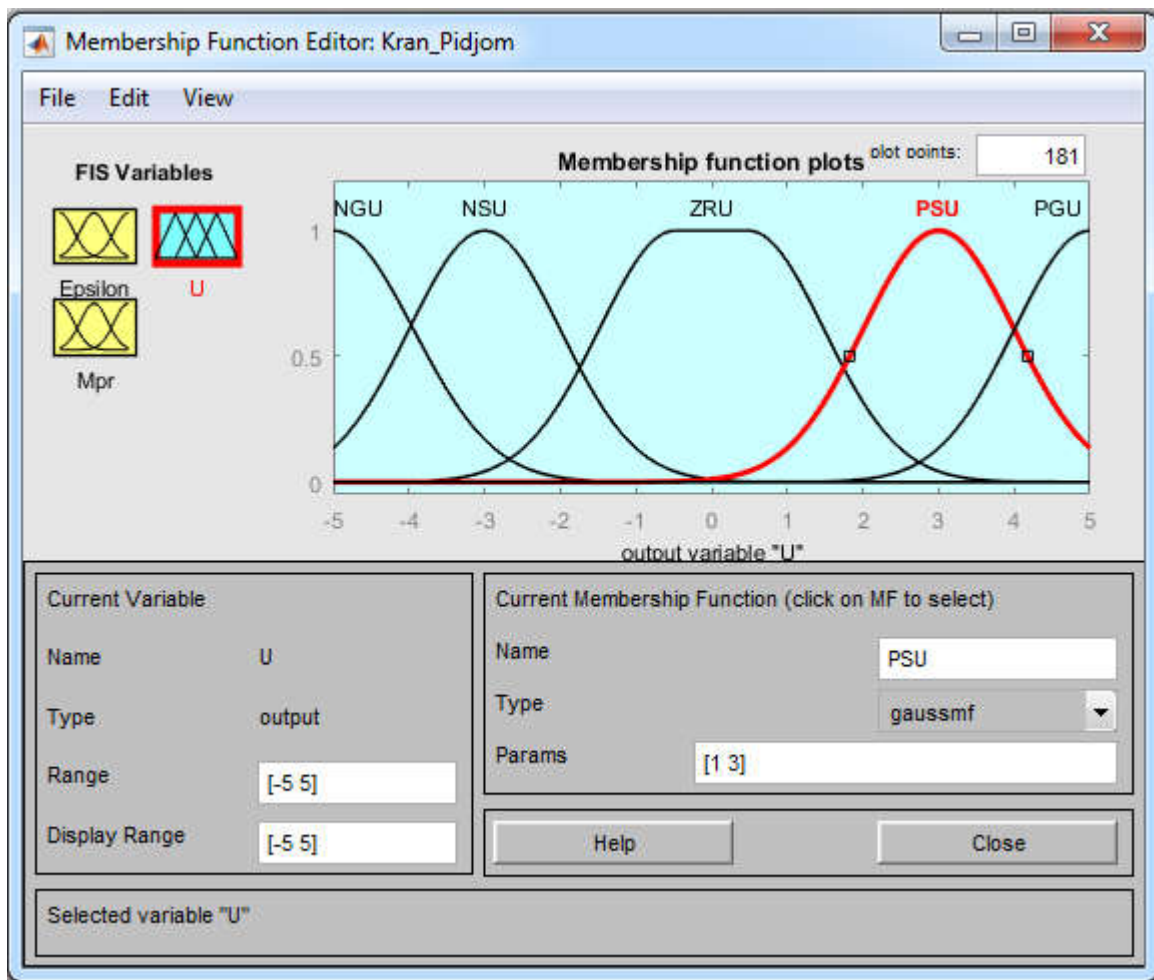


Рис. 7.9. Функції приналежності «U» (вихідної змінної $U(t)$)

6. Тепер настала черга визначити правила нечіткого висновку для нечіткої системи, що розроблялася. Для цієї мети слід скористатися редактором правил, який може бути викликаний одним з наступних способів:

- подвійним клацанням на значку квадрата в центрі з ім'ям створюваної системи нечіткого висновку (Kran);
- командою меню **Edit\Rules...**;
- натисненням клавіш **Ctrl+3**.

Оскільки спершу база правил нечіткого висновку порожня, то після виклику редактора правил центральне поле введення не містить ніяких правил. Для їх визначення слід використовувати поля меню і перемикачі в нижній частині графічного інтерфейсу редактора правил. Для завдання першого правила слід залишити виділене за умовчанням поле з ім'ям терма "NGE" для першої вхідної змінної «Epsilon», поле з ім'ям терма "NGM" для другої вхідної змінної «Mpr» і поле з ім'ям терма "PGU" для вихідної змінної «Ureg». Перемикач **Connection** слід залишити в положенні **and** (логічне **I**) і натиснути на кнопку **Add rule**. Після цього перше правило відобразиться у першому рядку вікна.

Для завдання другого правила, для якого слід виділити імена термів "NGE", "NSM" і "PSU". Аналогічним чином задаються інші правила. Вид вікна редактора правил після їх визначення зображений на рис. 7.10.

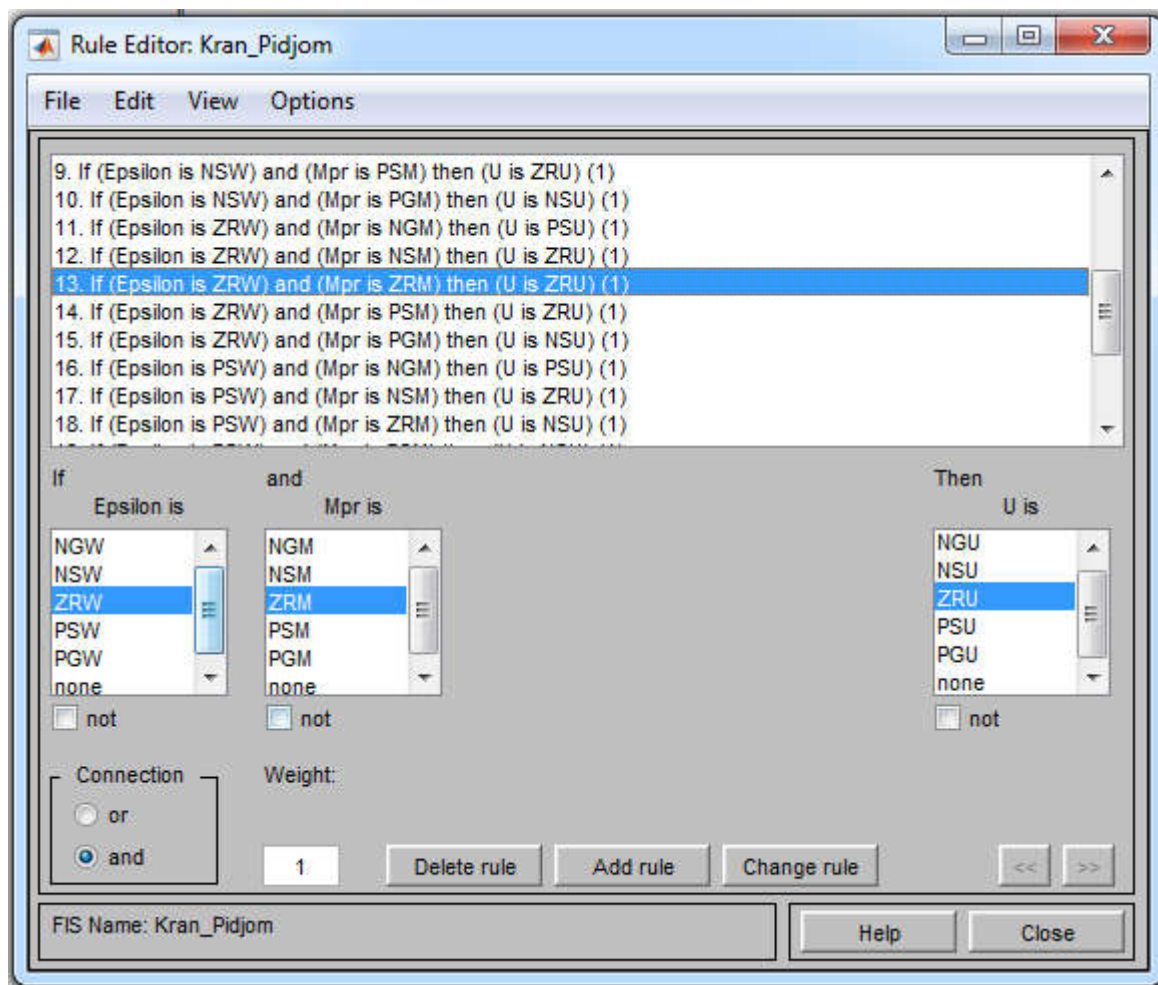


Рис. 7.10. Вікно редактора правил

Відмітимо, що в полі введення **Weight** відображається вага кожного правила, яку можна змінювати в межах інтервалу $[0, 1]$ (залишимо без зміни його значення за умовчанням, рівне 1 для всіх правил). Ця ж вага правил записується в круглих дужках у вікні правил після кожного з правил нечіткого висновку.

7. Після завдання правил нечіткого висновку виявляється можливим отримати результат нечіткого висновку (значення вихідної змінної) для конкретних значень вхідних змінних. З цією метою необхідно відкрити програму перегляду правил одним з наступних способів:

- командою меню **View\Rules** редактора **FIS**;
- командою меню **View\Rules** редактора функцій приналежності;
- командою меню **View\Rules** редактора правил;
- натисненням клавіш **Ctrl+5**.

Після виклику програми перегляду правил для системи нечіткого висновку, що розглядається в прикладі, відкриється вікно (рис. 7.11). У правій частині вікна в

графічній формі представлені функції приналежності аргументів «Epsilon» (помилки регулювання $\varepsilon(t)$), і «Mpr» (моменту пружності $M_{пр}(t)$), в лівій – функції приналежності змінної виходу «U» ($U(t)$) з поясненням механізму ухвалення рішення.

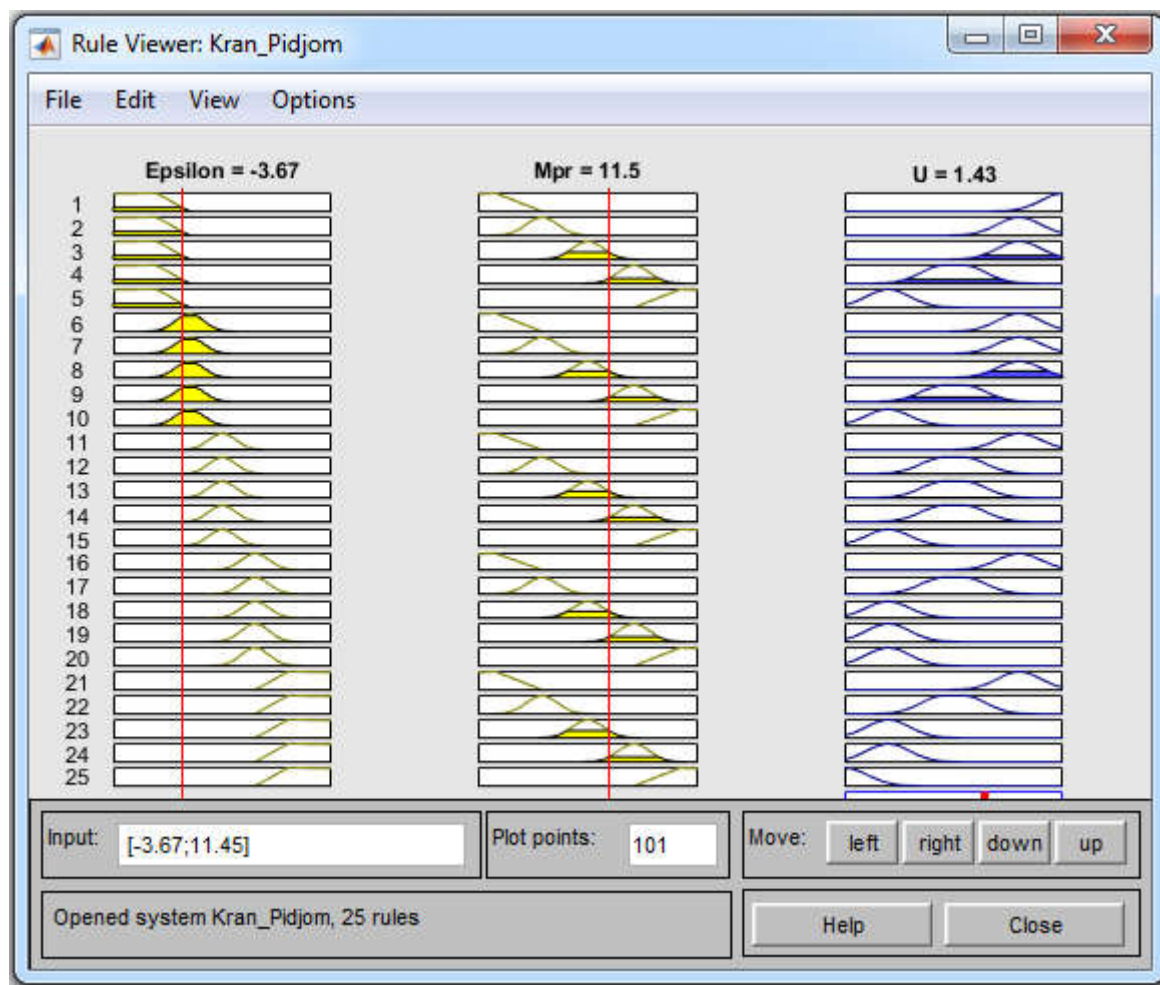


Рис. 7.11. Вікно перегляду правил

Червона вертикальна межа, що перетинає графіки в правій частині вікна, яку можна переміщати за допомогою миші, дозволяє змінювати значення змінних входу (це ж можна робити, задаючи числові значення в полі **Input** (Вхід)), при цьому відповідно змінюються значення «Epsilon» і «Mpr» в правій верхній частині вікна і, відповідно, значення вихідної змінної «Ureg» ($U_{рег}(t)$). Таким чином, за допомогою побудованої моделі і вікна перегляду правил можна вирішувати задачу синтезу Fuzzy регулятора. Зміна аргументу шляхом переміщення червоної вертикальної лінії дуже наочно демонструє, як система визначає значення виходу.

8. Для остаточного аналізу розробленої нечіткої моделі може виявитися корисною програма перегляду поверхні нечіткого висновку, яка може бути викликана одним з наступних способів:

- командою меню **View\Surface** з редактора **FIS**;
- командою меню **View\Surface** редактора функцій приналежності;
- командою меню **View\Surface** редактора правил;

командою меню **View\Surface** програми перегляду правил;
натисненням клавіш Ctrl+6.

Графічний інтерфейс програми перегляду поверхні нечіткого висновку для розробленої нечіткої моделі зображений на рис. 7.12.

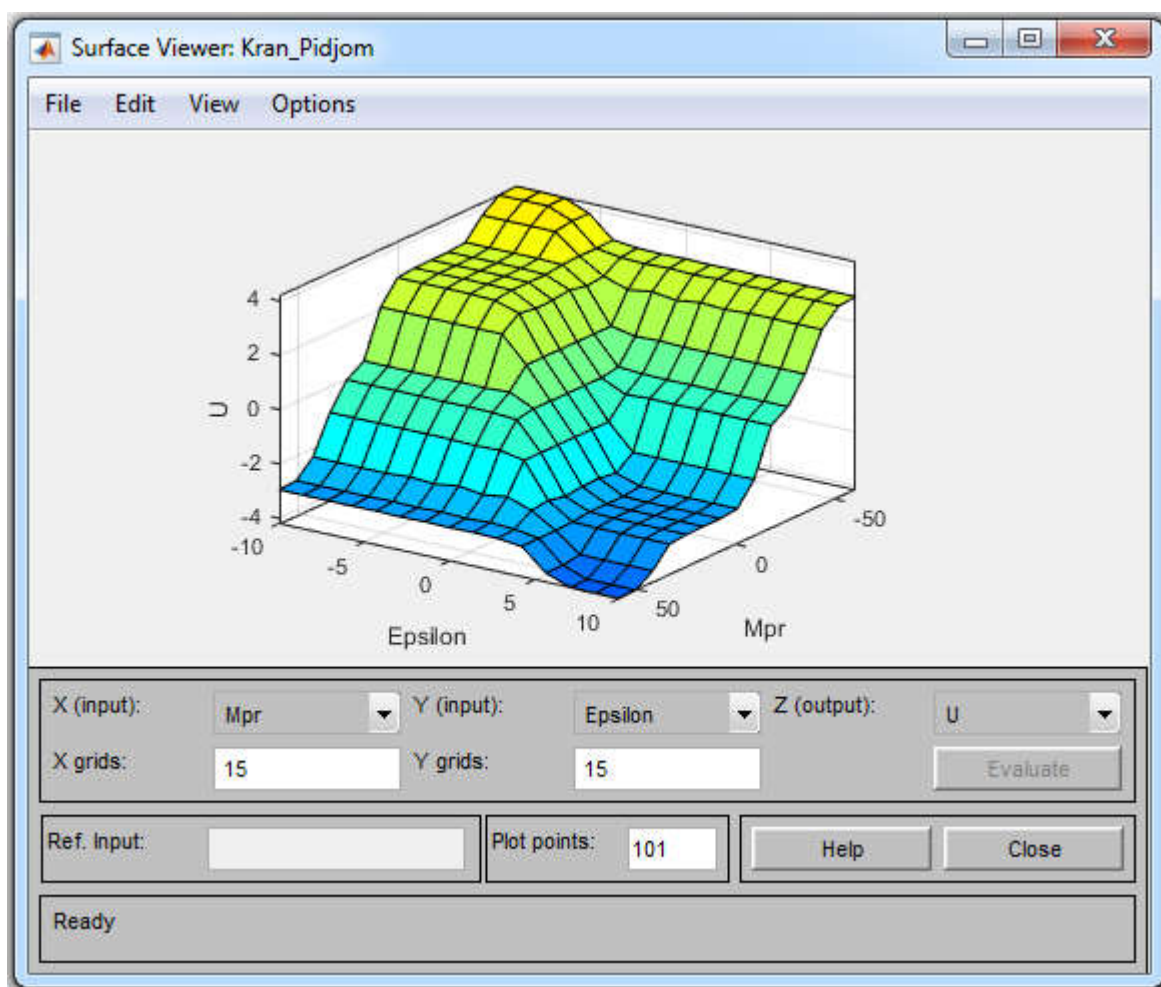


Рис. 7.12. Графічний вид залежності вихідної змінної від вхідних

Ця програма служить для загального аналізу адекватності нечіткої моделі, дозволяючи оцінити вплив зміни значень вхідних нечітких змінних на значення вихідної змінної. За допомогою мишки графік можна повертати на всі боки.

У разі потреби можна отримати графік залежності вихідної змінної від однієї з вхідних змінних. Для цього необхідно вибрати потрібну змінну в списку, що розкривається, **X (input)**, а в списку, що розкривається, **Y (input)**, вибрати значення - **none**-. Наприклад, графік залежності $U_{reg}(U_{per}(t))$ від $M_{pr}(M_{pr}(t))$ зображений на рис. 7.13.

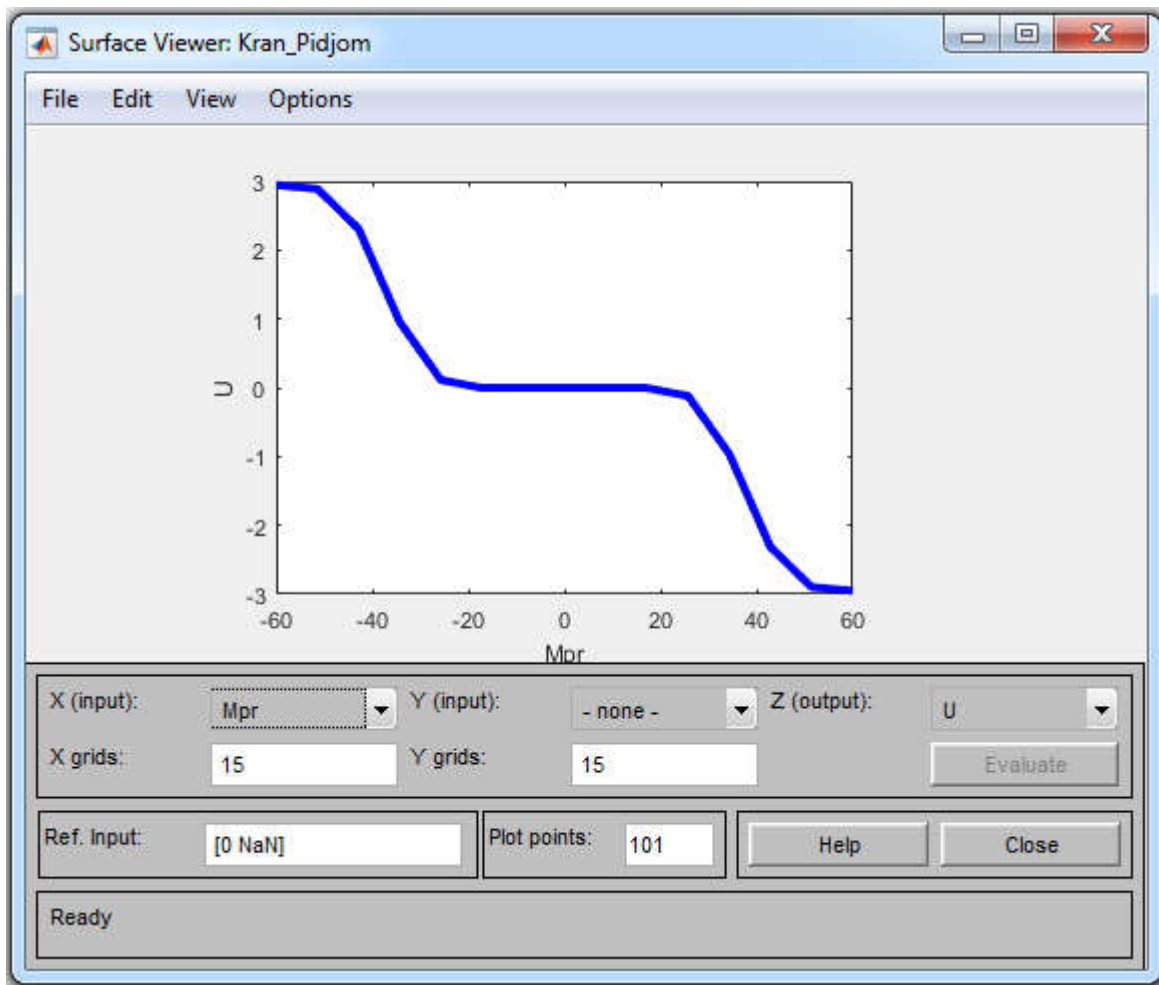


Рис. 7.13. Одновимірну залежність вихідної змінної «U» ($U(t)$) від вхідної – моменту пружності «Mpr» ($M_{\text{пр}}(t)$)

Як показали дослідження, змінюючи тип і параметри функцій приналежності, діапазон їх зміни **Range** можна сконструювати нечіткий регулятор, що дозволяє понизити динамічні навантаження в багатомасовій системі.

Закінчуючи розгляд процесу розробки простої системи нечіткого висновку в інтерактивному режимі, слід відмітити, що найбільш ефективним цей спосіб виявляється для складних нечітких моделей з великим числом змінних і правил нечіткого висновку. В цьому випадку завдання змінних і функцій приналежності їх термів в графічному режимі, а також візуалізація правил дозволяють істотно зменшити трудомісткість розробки нечіткої моделі, понизити кількість можливих помилок і скоротити загальний час нечіткого моделювання.

В той же час слід пам'ятати, що кількість змінних і правил в нечіткій моделі, які можуть бути візуалізовані, обмежена. Зокрема, якщо число вхідних змінних перевищує 7, то їх відображення у відповідних графічних редакторах відбувається із спотвореннями.

7.6 Моделювання двомасової електромеханічної системи з нечітким регулятором

Для моделювання двомасової системи з нечітким регулятором можуть бути використані моделі Simulink, приведені на рис.7.2 або 7.14. Перш ніж почати моделювання, необхідно зберегти створену структуру **FIS** з ім'ям **Kran.fis** в робочому просторі командою меню **File\Export\To Workspace....** Потім необхідно співставити розроблену систему нечіткого висновку, записану у файлі **Kran.fis** з блоком **Fuzzy Logic Controller**. Для цього шляхом подвійного клацання мишею на зображенні блока відкриємо вікно завдання параметрів блоку. Єдиним параметром **FIS File or Structure**, що задається в вікні, є ім'я системи нечіткого висновку, в даному випадку **Kran**.

На рис. 7.15–7.16 показані графіки перехідних процесів основних змінних стану системи з Fuzzy регулятором в режимі пуску і набросу навантаження. Як видно з графіків, перехідні процеси мають задовільний характер, що істотно знижує динамічні навантаження двомасової системи. (Перехідні процеси двомасової системи, що розглядається, без регулятора мають характер слабо затухаючих коливань).

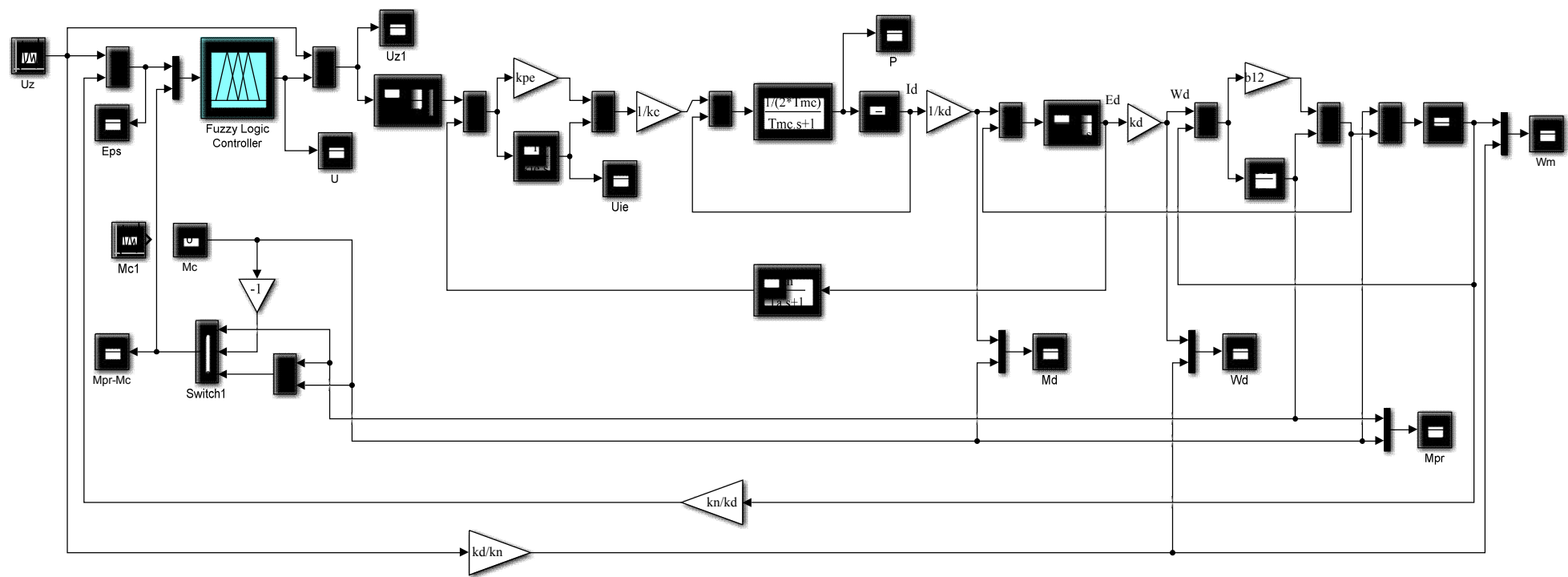
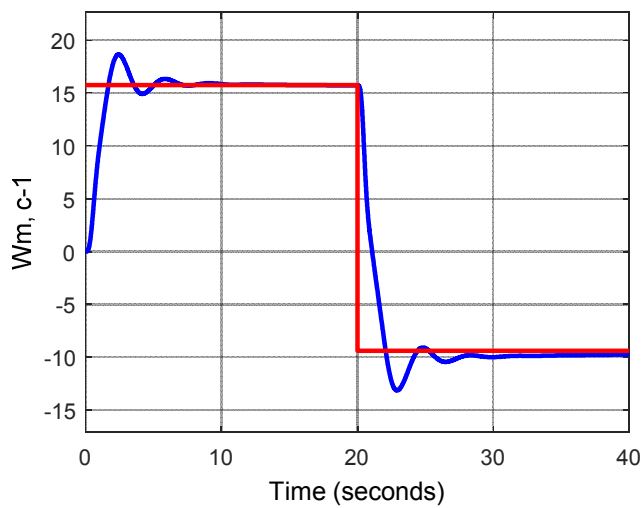
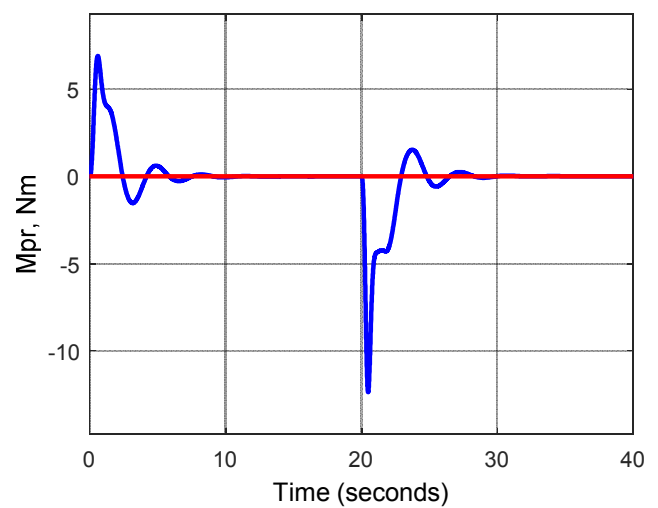


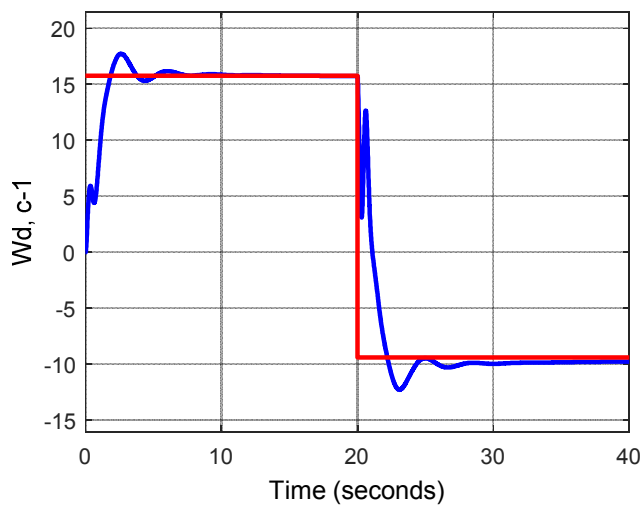
Рис. 7.14. Схема моделі двомасової системи з нечітким регулятором



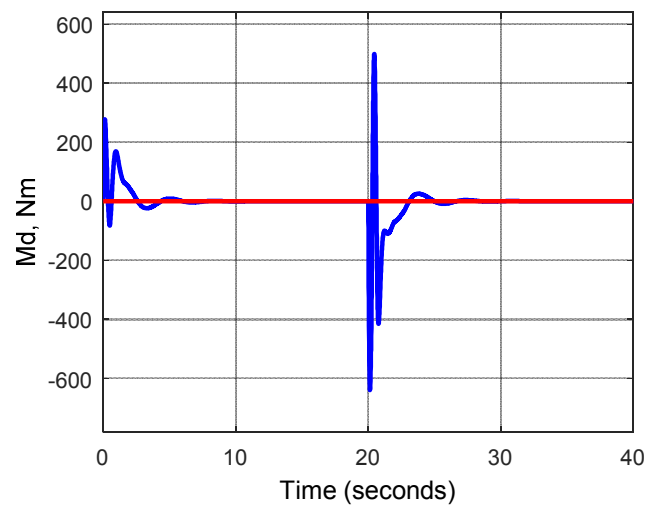
а)



б)



в)



г)

Рис. 7.15. Графіки перехідних процесів двомасової системи з нечітким регулятором по задаючій дії

а) швидкість механізму $\omega_m = f(t)$; б) момент пружності $M_{np} = f(t)$;

в) швидкість двигуна $\omega_d = f(t)$; г) момент двигуна $M_d = f(t)$

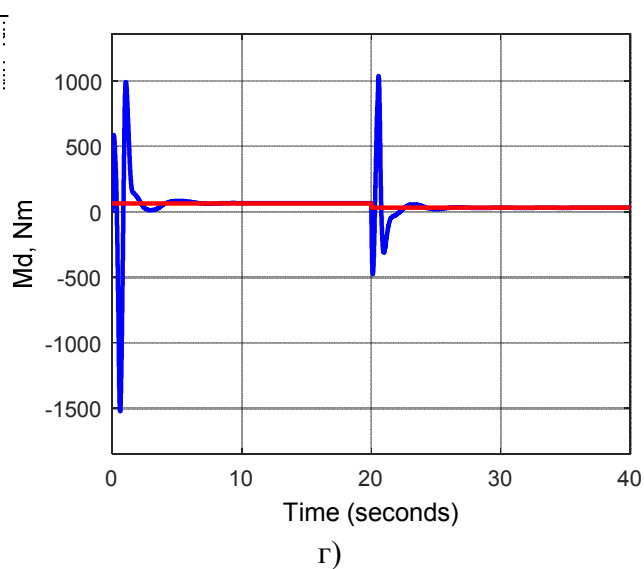
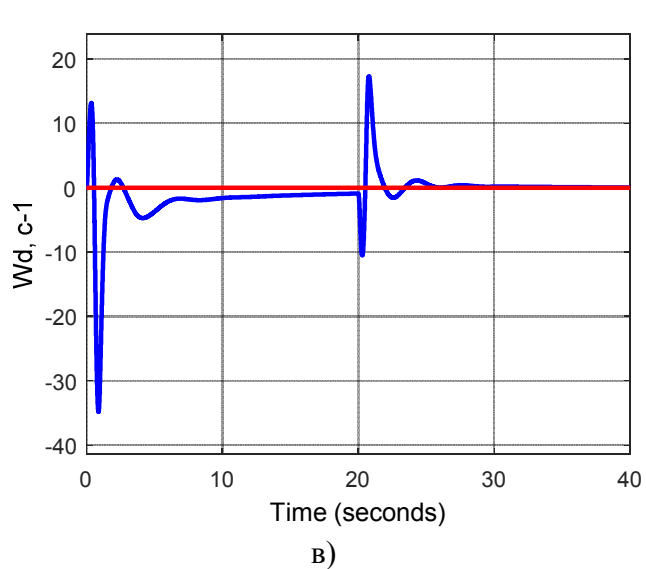
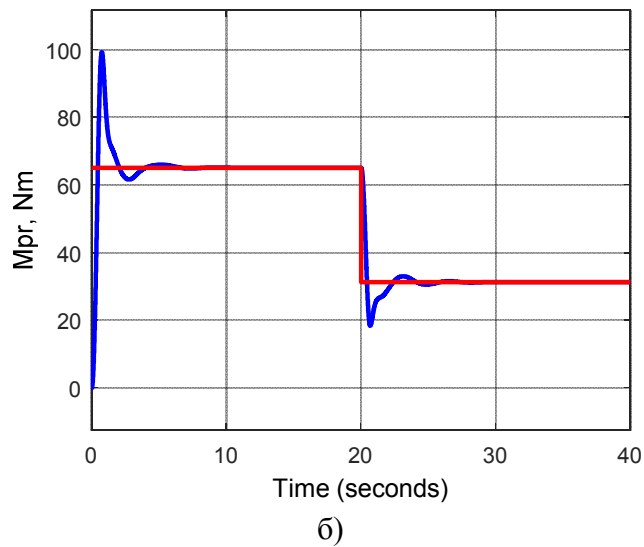
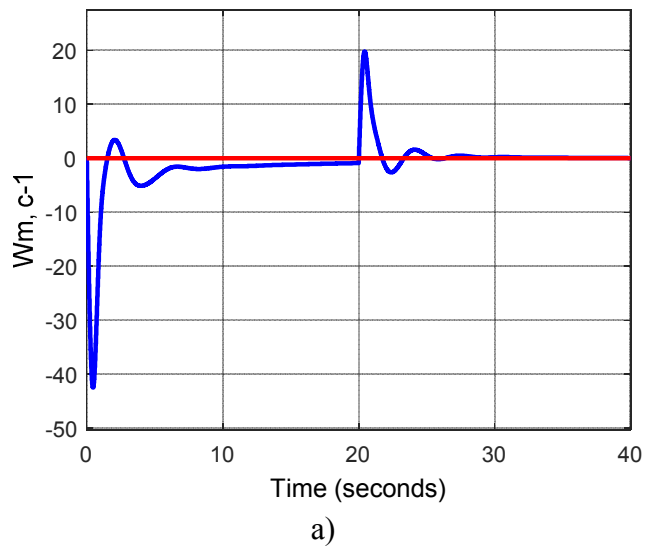


Рис. 7.16 .Графіки перехідних процесів двомасової системи
з нечітким регулятором по обурюючій дії

- а) швидкість механізму $\omega_m = f(t)$; б) момент пружності $M_{пр} = f(t)$;
в) швидкість двигуна $\omega_d = f(t)$; г) момент двигуна $M_d = f(t)$

Питання для самоконтролю.

1. Зобразіть схему системи управління двомасовою електромеханічною системою з нечітким регулятором?
2. Які сигнали слід подавати на вхід нечіткого регулятора?
3. Як визначаються вхідні і вихідні лінгвістичні змінні системи нечіткого висновку?
4. Які множини використовуються як терм-множини всіх лінгвістичних змінних?
5. Як формується база правил системи нечіткого висновку? Як визначається їх кількість?
6. Поясніть порядок синтезу нечітких регуляторів в середовищі MATLAB.
7. Як задаються терми і їх функції приналежності для вхідних і вихідних змінних системи нечіткого висновку?
8. Як задаються діапазони зміни вхідних і вихідних змінних?
9. Як виконується настройка функцій приналежності вхідних і вихідних змінних?
10. Як задаються правила нечіткого висновку?
11. Якими способами може бути викликана програма перегляду поверхні нечіткого висновку?
12. Як можна отримати графік залежності вихідної змінної від однієї з вхідних змінних?
13. Як виконується моделювання системи управління з нечітким регулятором?

ПОБУДОВА МОДЕЛЕЙ ДИНАМІЧНИХ ОБ'ЄКТІВ З ВИКОРИСТАННЯМ ГІБРИДНИХ МЕРЕЖ

8.1 Реалізація ANFIS в середовищі MATLAB

У пакеті Fuzzy Logic Toolbox системи MATLAB гібридні мережі реалізовані у формі адаптивних систем нейро-нечіткого виведення **ANFIS**. При цьому розробка і дослідження гібридних мереж виявляється можливою:

- у інтерактивному режимі за допомогою спеціального графічного редактора адаптивних мереж, що отримав назву редактора **ANFIS**;
- у режимі командного рядка за допомогою введення імен відповідних функцій з необхідними аргументами безпосередньо у вікно команд системи MATLAB. Для роботи в режимі командного рядка призначені спеціальні функції [76, 77].

Редактор **ANFIS** дозволяє створювати або завантажувати конкретну модель адаптивної системи нейро-нечіткого висновку, виконувати її навчання, візуалізувати її структуру, змінювати і налаштовувати її параметри, а також використовувати налаштовану мережу отримання результатів нечіткого висновку.

Графічний інтерфейс редактора **ANFIS** викликається функцією **anfisedit** командного рядка (рис. 8.1).

Головне меню редактора **ANFIS** достатньо просте і призначено для роботи із заздалегідь створеною системою нечіткого висновку. Основну частину графічного інтерфейсу займає вікно візуалізації даних, яке розташоване нижче за головне меню. Для новостворюваної гібридної мережі це вікно не містить ніяких даних.

Для створення гібридної мережі необхідно завантажити дані. Для цієї мети слід скористатися кнопкою **Load Data** в лівій нижній частині графічного вікна. При цьому дані можуть бути завантажені із зовнішнього файлу (**disk**) або з робочої області (**workspace**). У першому випадку необхідно заздалегідь створити файл з початковими даними (файл з розширенням **.dat**), який є звичайним текстовим файлом. При цьому початкові дані є звичайною числовою матрицею розмірності $m \times (n + 1)$, у якій кількість рядків m відповідає об'єму вибірки N_B , перші n стовпців – значенням вхідних змінних моделі, а останній стовпець – значенню вихідної змінної. Згідно правилам системи MATLAB окремі значення матриці відділяються пропусками, а кожен рядок матриці завершується символом «переведення каретки» (клавіша **Enter**).

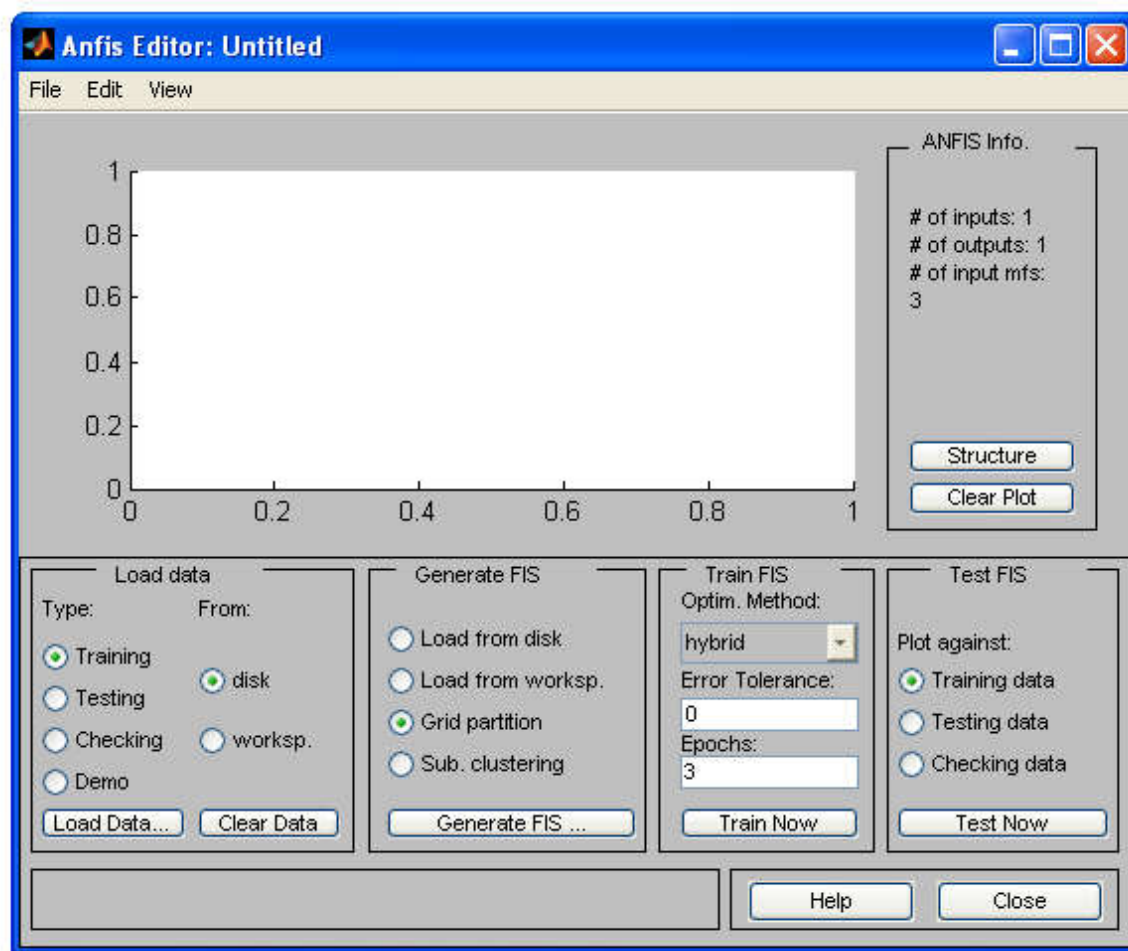


Рис. 8.1. Графічний інтерфейс редактора **ANFIS**, що викликається функцією **anfisedit**

Примітка. Хоча по кількості рядків матриці початкових даних не існує формальних рекомендацій, прийнято вважати, що якість навчання гібридної мережі, а, отже, і точність отримуваних результатів пропорційно залежить від об'єму навчальної вибірки. Що стосується кількості стовпців матриці початкових даних, то слід зазначити можливі проблеми з працездатністю системи MATLAB, якщо кількість вхідних змінних перевищує 5 – 6.

Завантажувані початкові дані можуть бути одного з наступних типів:

- навчальні дані (**Training**) – обов'язкові дані, які використовуються для побудови гібридної мережі;
- тестові дані (**Testing**) – необов'язкові дані, які використовуються для тестування побудованої гібридної мережі з метою перевірки якості функціонування побудованої гібридної мережі;
- перевірочні дані (**Checking**) – необов'язкові дані, які використовуються для перевірки побудованої гібридної мережі з метою з'ясування факту перенавчання мережі.

Пояснимо пункти меню і опції редактора.

Пункти меню **File** і **View**, загалом ідентичні аналогічним пунктам редактора нечіткої системи виведення (**FIS**-редактора) пакету **Fuzzy Logic** системи MATLAB, за тим виключенням, що тут робота може відбуватися тільки з алгоритмом нечіткого виведення **Sugeno**. Пункт меню **Edit** містить єдиний підпункт – **Undo** (Відмінити виконану дію).

Набір опцій **Load Data** (Завантажити дані) в нижній лівій частині вікна редактора розглянутий вище і включає:

- тип (**Type**) завантажуваних даних (для навчання – **Training**, для тестування – **Testing**, для перевірки – **Checking**, демонстраційні – **Demo**);
- місце, звідки повинні завантажуватися дані (з диска – **disk** або з робочої області MATLAB – **workspace**).

До даних опцій відносяться дві кнопки, натиснення на яких приводить до необхідних дій – **Load Data...** (Завантажити дані) і **Clear Data** (очистити, тобто стерти введені дані).

Наступна група опцій (в середині нижньої частини вікна **ANFIS**-редактора) об'єднана під ім'ям **Generate FIS** (Створення нечіткої системи висновку). Дана група включає опції:

- завантаження структури системи з диска (**Load from disk**);
- завантаження структури системи з робочої області MATLAB (**Load from worksp.**);
- розбиття (ділення) областей визначення вхідних змінних (аргументів) на підобласті – незалежно для кожного аргументу (**Grid partition**);
- розбиття всієї області визначення аргументів (вхідних змінних) на підобласті – в комплексі для всіх аргументів (**Subtract clustering** або **Sub. clustering**), а також кнопку **Generate FIS**, натиснення якої приводить до процесу створення гібридної системи з точністю до ряду параметрів.

Наступна група опцій – **Train FIS** (навчання нечіткої системи висновку) – дозволяє визначити метод «навчання» (**Optim. Method**) системи (тобто метод настройки її параметрів) – гібридний (**hybrid**) або зворотного розповсюдження помилки (**backpropa**), встановити рівень поточної сумарної (по всіх зразках) помилки навчання (**Error Tolerance**), досягши якого процес навчання закінчується і кількість циклів навчання (**Epochs**), тобто кількість «прогонів» всіх зразків (або прикладів) навчальної вибірки; процес навчання, таким чином закінчується або досягши відміченого рівня помилки навчання, або при проведенні заданого кількості циклів.

Кнопка **Train Now** (почати навчання) дозволяє почати процес навчання, тобто процес настройки параметрів гібридної мережі.

У правому верхньому кутку вікна **ANFIS**-редактора видається інформація (**ANFIS Info.**) про проєктовану систему: про кількість входів, виходів, функцій при-

належності входів; натиснення кнопки **Structure** (Структура) дозволяє побачити структуру мережі. Кнопка **Clear** (очистити) дозволяє стерти всі результати.

Опції **Test FIS** в правому нижньому кутку вікна дозволяють провести перевірку і тестування створеної і навченої системи з виведенням результатів у вигляді графіків (відповідні графіки для навчальної вибірки – **Training data**, тестуючої вибірки – **Testing data** і перевіркової вибірки – **Checking data**. Кнопка **Test Now** дозволяє запустити вказані процеси.

8.2 Розробка нечіткої моделі гібридної мережі для вирішення завдання ідентифікації двомасової електромеханічної системи

Як приклад побудови адаптивної системи нейро-нечіткого висновку розглянемо процес розробки нечіткої моделі гібридної мережі для вирішення завдання ідентифікації двомасової електромеханічної системи управління механізмом підйому мостового. Цей приклад дозволяє не тільки уточнити зміст і послідовність етапів розробки, але і оцінити точність отриманої нечіткої моделі за допомогою порівняння прогнозованих модельних значень із значеннями, отриманими шляхом моделювання двомасової електромеханічної системи.

Алгоритмічна схема двомасової системи наведена в главі 7 (див. рис.7.1).

Загальна послідовність процесу розробки моделі гібридної мережі може бути представлена в наступному вигляді.

I. Спершу підготуємо навчальні **Training**, тестові **Testing** і перевірочні **Checking** дані. Ці дані можуть бути сформовані різними способами. Розглянемо один з них.

Створимо у вікні Simulink схему моделі системи (рис.8.2), а в редакторі-відладчику m-файлів сформуємо файл вихідних даних системи і збережемо його на диску під іменем “**Dat_Kran**”. За допомогою команд меню **Debug\Run** введемо початкові дані в робочу область MATLAB.

У схемі моделі рис.8.2 використаний блок **To Workspace** для запису вхідного і вихідного сигналів системи (в даному випадку напруги завдання $U_{\text{сд}}(t)$ і швидкості механізму $\omega_1(t)$) у робочу область системи MATLAB. У вікні завдання параметрів блоку слід задати наступні параметри.

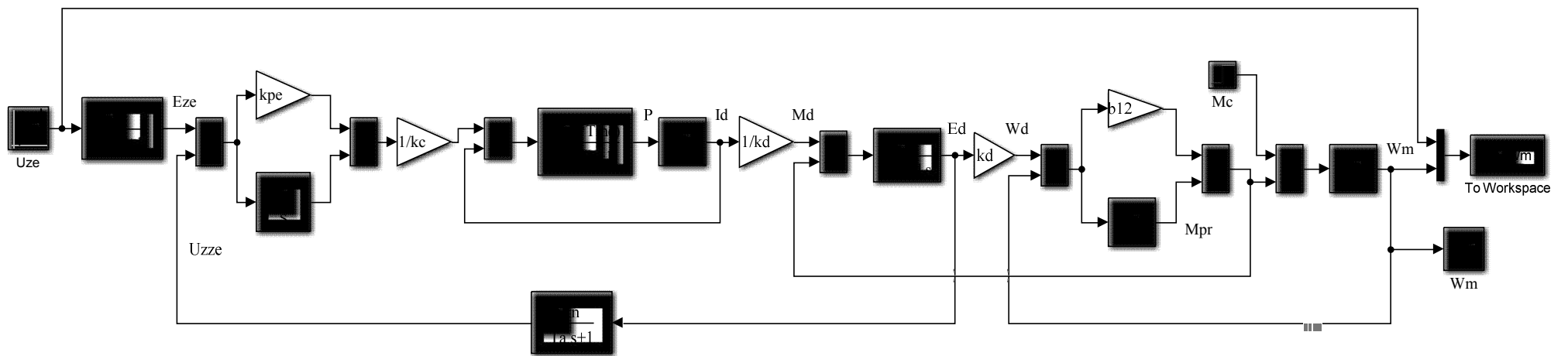


Рис. 8.2. Схема моделі двомасової системи, побудованої в Simulink

Variable name [ім'я змінної]. Ім'я масиву, в який записуватимуться дані. Встановимо **Uz_Wm**.

Sample time [такт дискретності]. Величина такту дискретності встановлюється з тих же міркувань, що і при розробці нейромережевої моделі (див. главу 6). Для даного прикладу встановимо такт дискретності $\Delta t = 0,05c$.

Save format [формат даних, що зберігаються]. З пропонованого списку вибираємо формат **Array** – масив. Дані зберігаються як масив, в якому число рядків визначається числом розрахункових точок в часі, а число стовпців – розмірністю вектора, що подається на вхід блоку. У даному прикладі масив міститиме два стовпці: перший стовпець відповідає вхідному сигналу системи $U_{\varphi A}(t)$, другий – вихідному сигналу $\omega_i(t)$.

Решту параметрів блоку залишимо без зміни.

Як джерело сигналу в схемі рис. 8.2 використаний блок **Uniform Random Number** – джерело сигналу з рівномірним розподілом. У вікні завдання параметрів даного блоку слід задати такі ж параметри, що і при розробці нейромережевої моделі системи, а саме:

Minimum [мінімальний рівень сигналу], **Maximum** [максимальний рівень сигналу] – мінімальну і максимальну величину напруги завдання $U_{\varphi A}(t)$ встановлюємо $+10B$ і $-10B$ відповідно.

Sample time [такт дискретності] – встановлюємо $5c$.

Задамо у вікні Simulink моделі системи час моделювання $100c$ і промодельємо систему.

У вікні **Workspace** системи MATLAB виводяться імена всіх змінних, які знаходяться в робочій області, у тому числі і масиву, що містить значення вхідний і вихідний сигнал системи **Uz_Wm**. Масив має два стовпці і 2001 рядків.

Збережемо масив **Uz_Wm** у файлі з назвою **Dat_Uz_Wm** і розмістимо файл в папку **Fuzzy_Neural_Work** на диску E. Команда **save** дозволяє зберегти вміст робочої області в двійковому MAT-файлі, який можна потім викликати командою **load**. Проте для вирішення даного завдання ідентифікації це не прийнятно, оскільки, як було вказано вище, файл початкових даних повинен бути звичайним текстовим файлом з розширенням **.dat**.

Для збереження масиву **Uz_Wm** в ASCII-форматі в командному вікні системи MATLAB слід задати команду:

```
save('E:\FuzzyNeuralWork\Dat_Uz_Wm','Uz_Wm','-ascii')
```

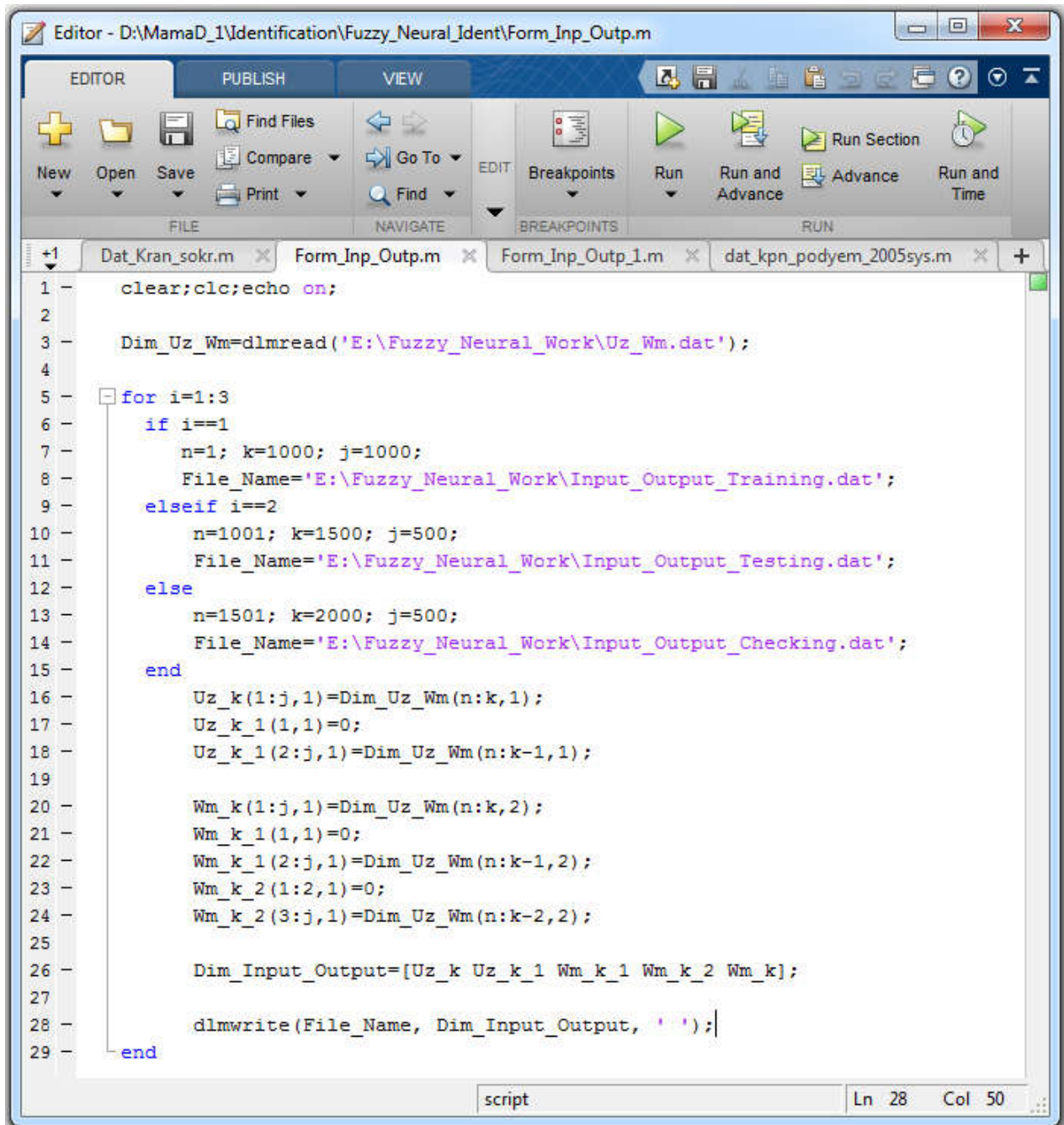
або

```
save E:\FuzzyNeuralWork\Dat_Uz_Wm Uz_Wm -ascii
```

Використовуємо надалі перші 1000 рядків масиву **Uz_Wm** для формування навчальних даних **Input_Output_Training**, наступні 500 рядків для формування тестових даних **Input_Output_Testing** і потім 500 рядків для формування перевірочних

даних **Input_Output_Checking**.

Як було вказано вище, для побудови моделі динамічного об'єкту, необхідно вхідну послідовність задати на основі поточного значення вхідного сигналу об'єкту, і ряду попередніх значень вхідного і вихідного сигналів. Кількість затримок по вхідному і вихідному сигналах заздалегідь вибираються на підставі попередніх знань про об'єкт ідентифікації, а потім корегуються в процесі побудови моделі об'єкту.



```
1 clear;clc;echo on;
2
3 Dim_Uz_Wm=dlmread('E:\Fuzzy_Neural_Work\Uz_Wm.dat');
4
5 for i=1:3
6     if i==1
7         n=1; k=1000; j=1000;
8         File_Name='E:\Fuzzy_Neural_Work\Input_Output_Training.dat';
9     elseif i==2
10        n=1001; k=1500; j=500;
11        File_Name='E:\Fuzzy_Neural_Work\Input_Output_Testing.dat';
12    else
13        n=1501; k=2000; j=500;
14        File_Name='E:\Fuzzy_Neural_Work\Input_Output_Checking.dat';
15    end
16    Uz_k(1:j,1)=Dim_Uz_Wm(n:k,1);
17    Uz_k_1(1,1)=0;
18    Uz_k_1(2:j,1)=Dim_Uz_Wm(n:k-1,1);
19
20    Wm_k(1:j,1)=Dim_Uz_Wm(n:k,2);
21    Wm_k_1(1,1)=0;
22    Wm_k_1(2:j,1)=Dim_Uz_Wm(n:k-1,2);
23    Wm_k_2(1:2,1)=0;
24    Wm_k_2(3:j,1)=Dim_Uz_Wm(n:k-2,2);
25
26    Dim_Input_Output=[Uz_k Uz_k_1 Wm_k_1 Wm_k_2 Wm_k];
27
28    dlmwrite(File_Name, Dim_Input_Output, ' ');
29 end
```

Рис. 8.3. Текст m-файлу формування масивів навчальних, тестових і перевірочних даних

У даному прикладі сформуємо вхідну послідовність на основі поточного значення вхідного сигналу системи $U_{3E}(k)$ і вхідного сигналу, затриманого на один крок дискретності $U_{3E}(k-1)$, а також два затриманих на один і два кроки вихідних сигналів, тобто $\omega_m(k-1)$ і $\omega_m(k-2)$ відповідно.

Для формування масивів вказаних даних може бути використаний будь-який текстовий редактор, наприклад редактор-відладчик m-файлів системи MATLAB (рис. 8.3). У вікні відладчика приведений текст m-файлу формування масивів навчальних, тестових і перевірочних даних. **Form_Inp_Outp.m**. Вказані масиви збережені у файлах з назвами **Input_Output_Training.dat**, **Input_Output_Testing.dat** і **Input_Output_Checking.dat** відповідно.

Дані масиви містять 5 стовпців. Перші 4 стовпці відповідають значенням входних сигналів моделі ($U_{3E}(k)$, $U_{3E}(k-1)$, $\omega_m(k-1)$, $\omega_m(k-2)$), а 5 стовпець – значенням вихідного сигналу $\omega_m(k)$. У вікні, зображеному на рис.8.4, приведені перші 21 рядки масиву **Input_Output_Training**.

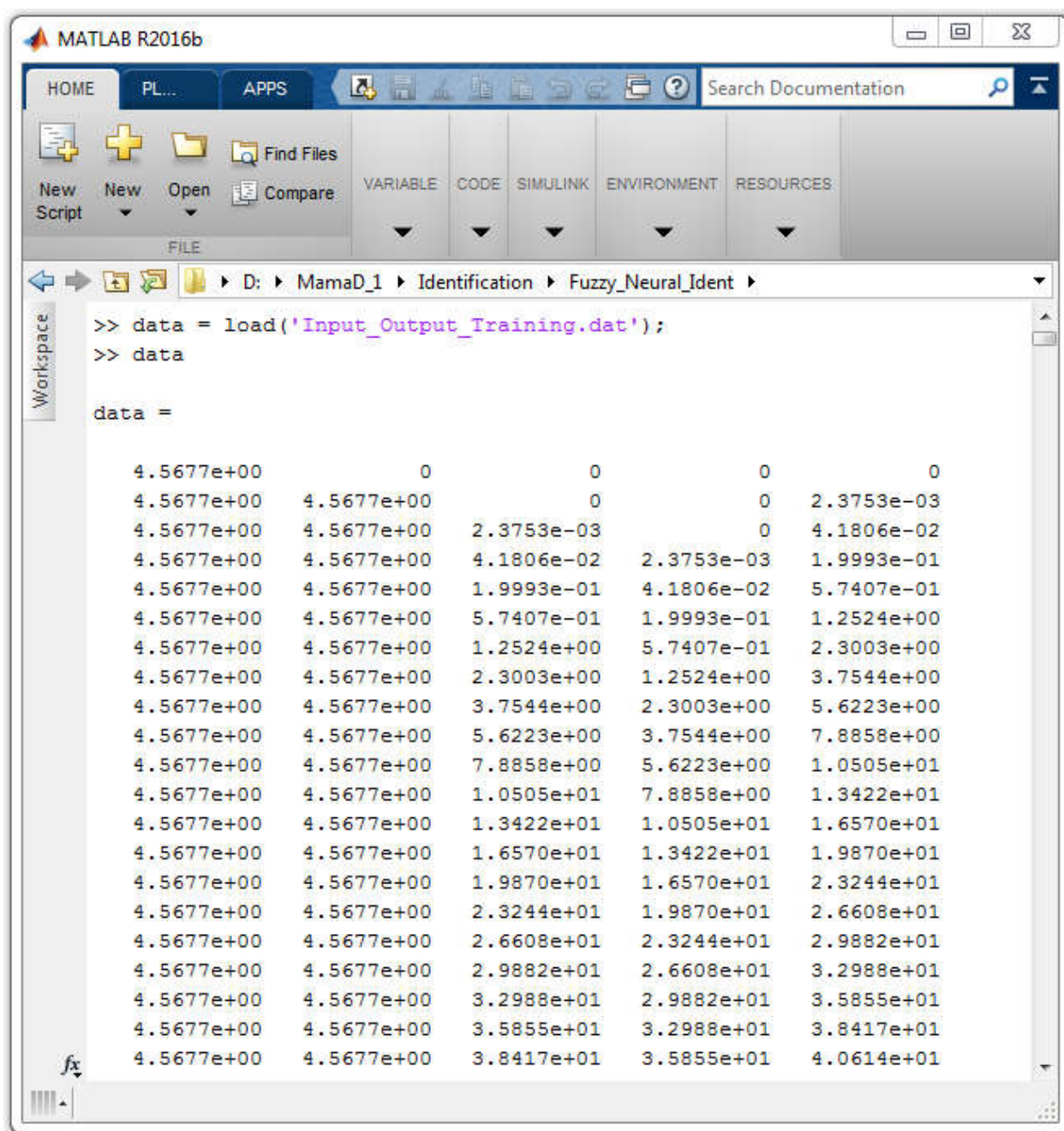


Рис. 8.4. Навчальні дані **Input_Output_Training**

Отже, в папці **E:\FuzzyNeuroWork** є три файли **Input_Output_Training.dat**, **Input_Output_Testing.dat** і **Input_Output_Checking.dat**, які можуть бути використані для розробки гібридної мережі за допомогою редактора **ANFIS**.

Завантажимо навчальні дані з файлу **Input_Output_Training.dat**, скориставшись кнопкою **Load Data** графічного вікна редактора **ANFIS**, встановивши тип завантажуваних даних **Training**. Після завантаження навчальних даних їх структура буде відображена в робочому вікні редактора (рис. 8.5). При цьому кожному рядку даних відповідає окрема точка графіка, яка для навчальних даних зображується кружком. На горизонтальній осі указуються порядковий номер (індекс) окремого рядка даних, а вертикальна вісь служить для вказівки значень вихідної змінної. У даному прикладі використовується 1000 точок навчальних даних.

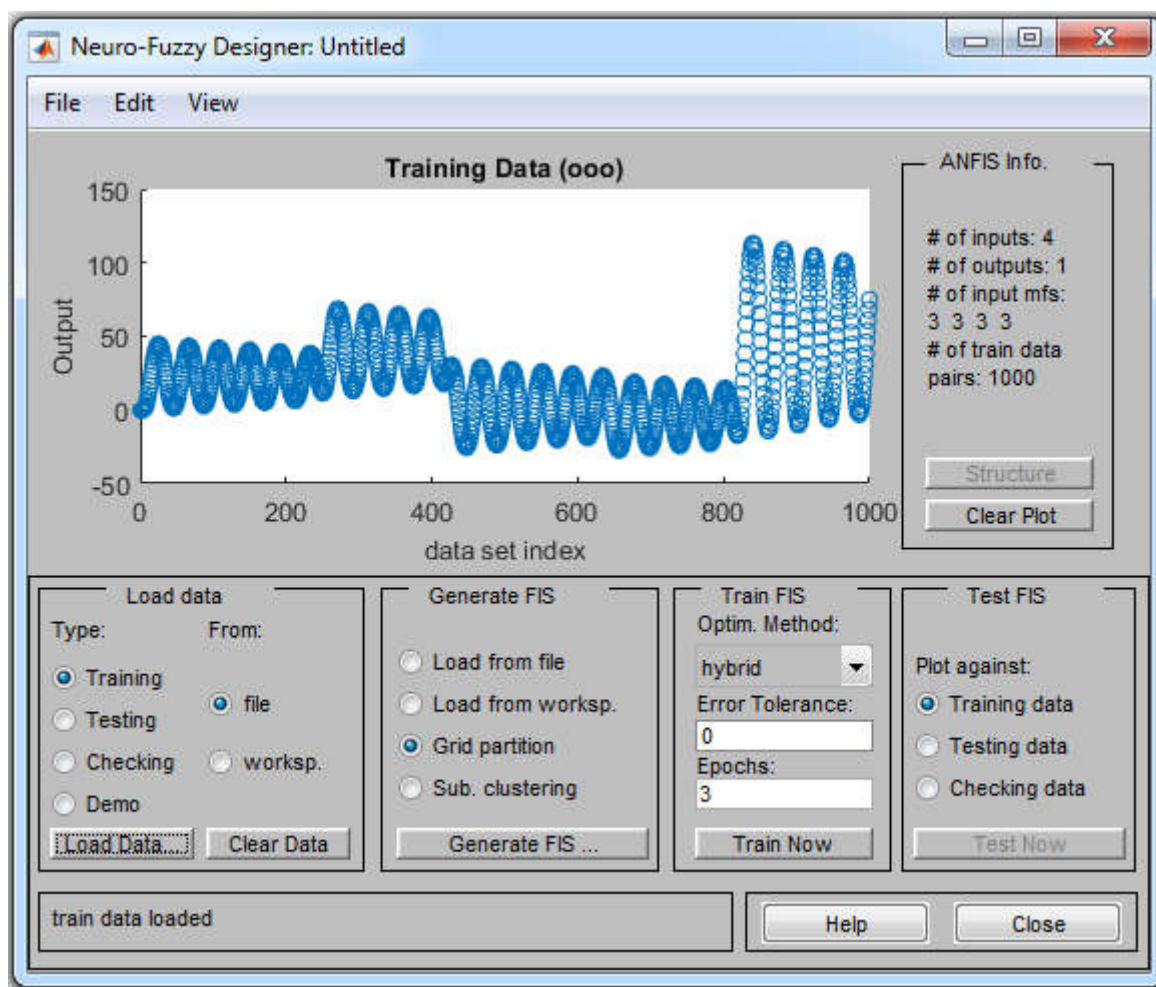


Рис. 8.5. Графічний інтерфейс редактора **ANFIS** після завантаження файлу з навчальними даними **Input_Output_Training.dat**

Завантажимо тестові дані з файлу **Input_Output_Testing.dat**. Для цього змінимо тип завантажуваних даних на **Testing** і виберемо ім'я вказаного файлу. В результаті тестові дані будуть завантажені в редактор **ANFIS** і зображені в робочому вікні (рис.8.6). В цьому випадку початковий графік буде доповнений 500 точками тестових даних, кожному рядку яких відповідає окрема точка графіка, зображена плюсом.

Аналогічно завантажимо перевірочні дані з файлу **Input_Output_Checking.dat**, встановивши тип завантажуваних даних **Checking**. В результаті дані будуть завантажені в редактор **ANFIS** і зображені в робочому вікні (рис. 8.7). Початковий графік доповнюється 500 точкою перевірочних даних, кожному рядку яких також відповідає окрема точка графіка, зображена жирною точкою.

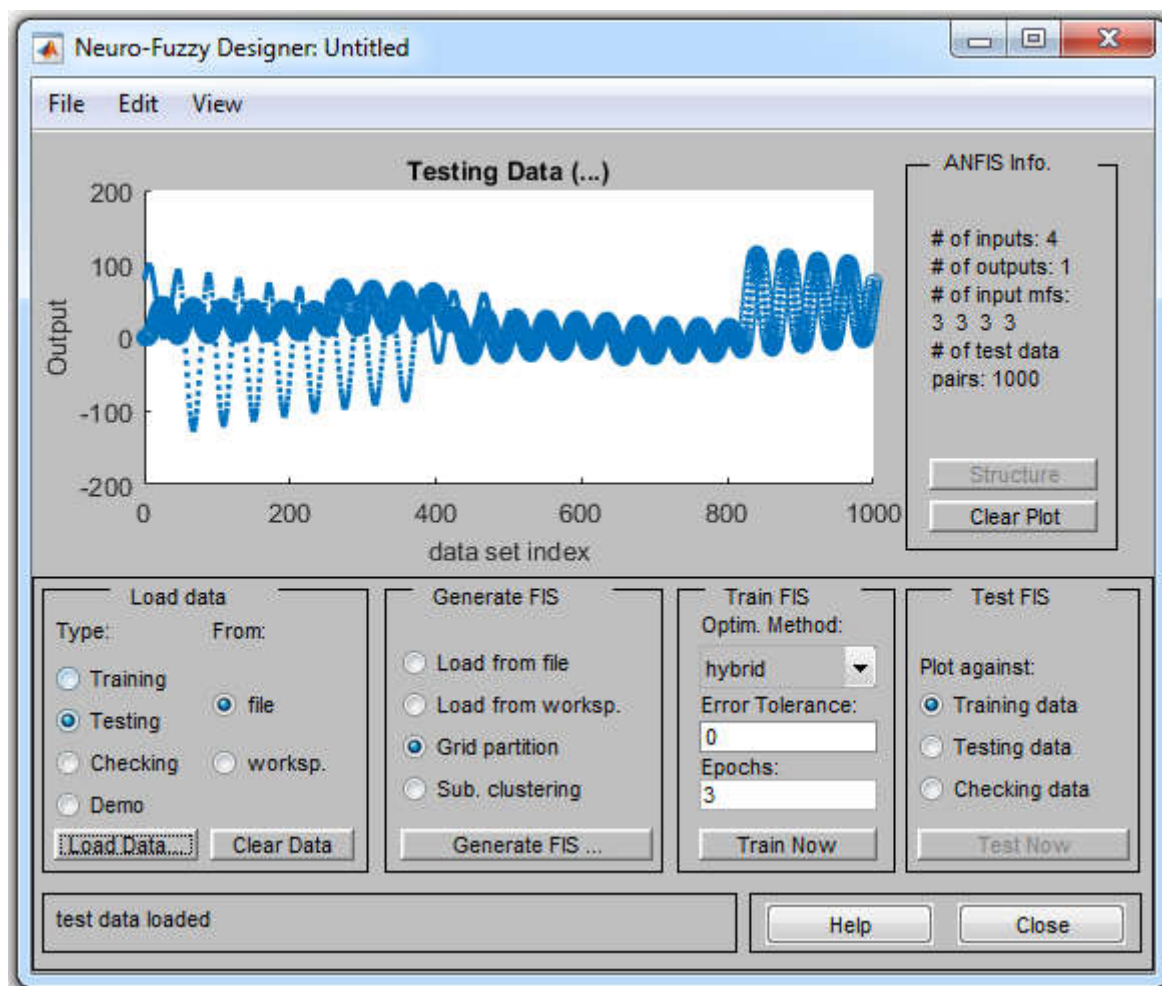


Рис.8.6 .Графічний інтерфейс редактора **ANFIS** після завантаження файлу з тестовими даними **Input_Output_Testing.dat**

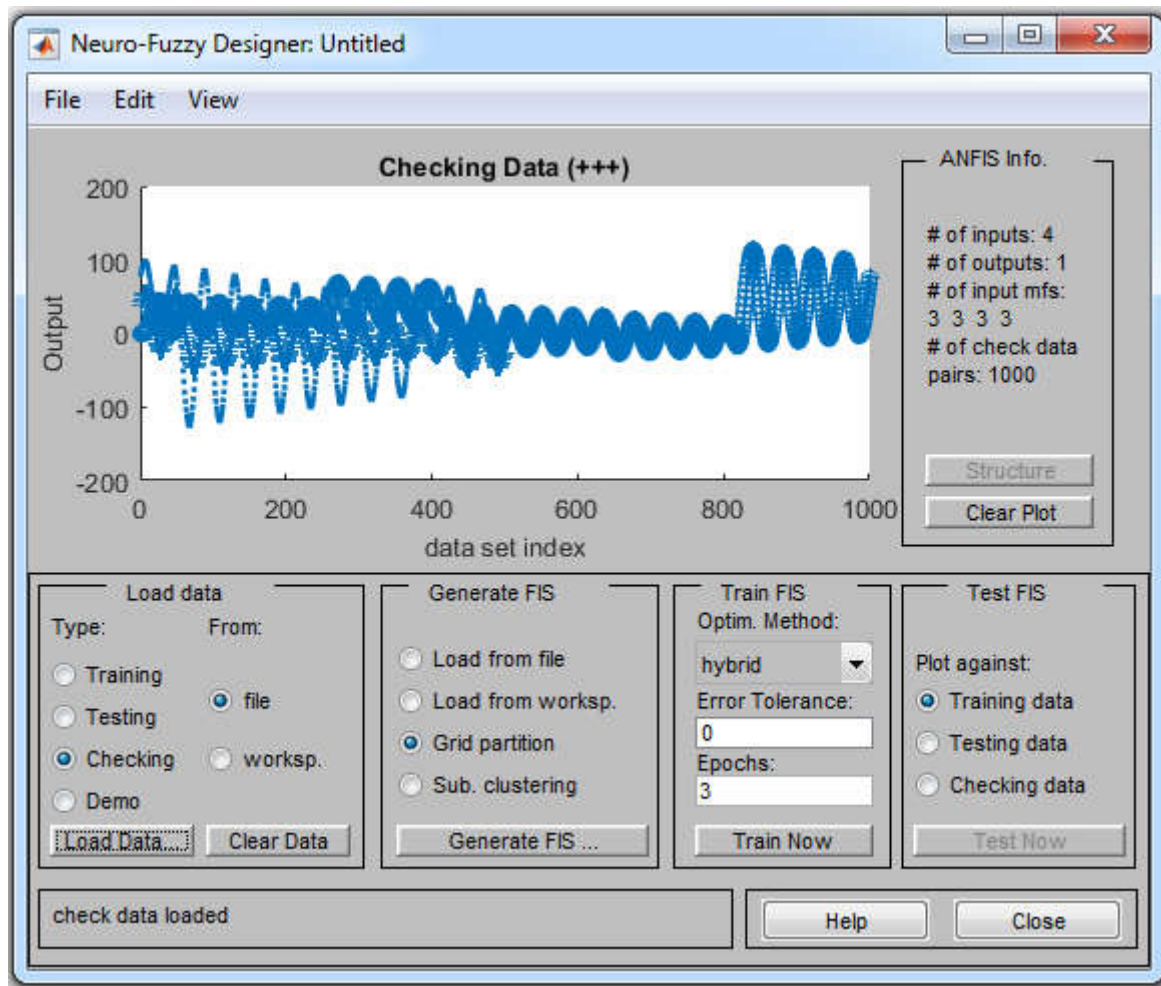


Рис. 8.7. Графічний інтерфейс редактора ANFIS після завантаження файлу з перевірочними даними **Input_Output_Checking.dat**

Після підготовки і завантаження даних можна згенерувати структуру системи нечіткого виведення **FIS** типа **Сугено**, яка є моделлю гібридної мережі в системі MATLAB. Для цієї мети слід скористатися кнопкою **Generate FIS** в нижній частині робочого вікна редактора. При цьому дві перші опції відносяться до заздалегідь створеної структури гібридної мережі, а дві останні – до форми розбиття вхідних змінних моделі.

Завантажити структуру вже створеної **FIS** можна або з диска (**Load from disk**), або з робочої області (**Load from worksp.**). При створенні структури нової **FIS** можна незалежно розбити всі вхідні змінні на області їх значень (**Grid partition**) або скористатися процедурою субтрактивної кластеризації для попереднього розбиття значень вхідних змінних на кластери близьких значень (**Sub. clustering**).

Натисненням кнопки **Generate FIS** викликається діалогове вікно з вказівкою числа і типу функцій приналежності для окремих термів вхідних змінних і вихідної змінної (рис. 8.8). В цьому випадку можна вибрати будь-який тип функцій приналежності з реалізованих в системі MATLAB і розглянутих раніше (див. главу 4).

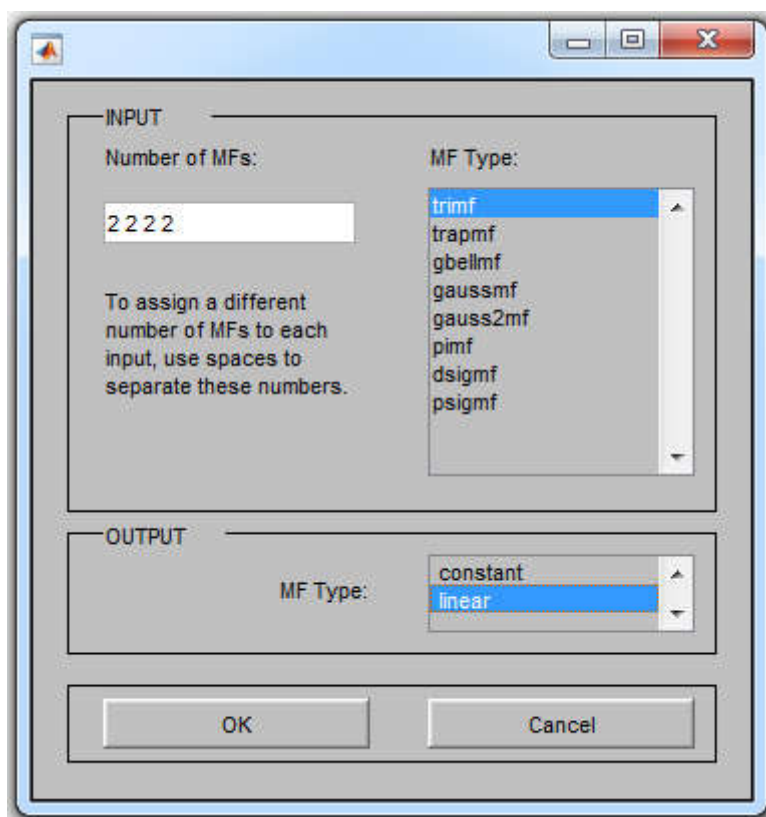


Рис. 8.8 .Діалогове вікно для завдання кількості і типу функцій приналежності

Після генерації структури гібридної мережі можна візуалізувати її структуру, для чого слід натиснути кнопку **Structure** в правій частині графічного вікна. Структура отриманої в результаті системи нечіткого виведення **FIS** відображається в окремому вікні (рис. 8.9).

Для даного прикладу найбільша точність ідентифікації отримана для випадку, коли система нечіткого висновку містить 4 вхідних змінних з 4 термами, 4 правила нечітких висновків, одну вихідну змінну з 4 термами. На рис.8.9 для наочності показана структура системи нечіткого висновку з 2 термами. Як тип їх функцій приналежності вибрані трикутні функції (встановлені системою MATLAB за умовчанням). Як функція приналежності вихідної змінної задана лінійна функція. Компоненти системи **FIS** зображуються вузлами відповідного кольору.

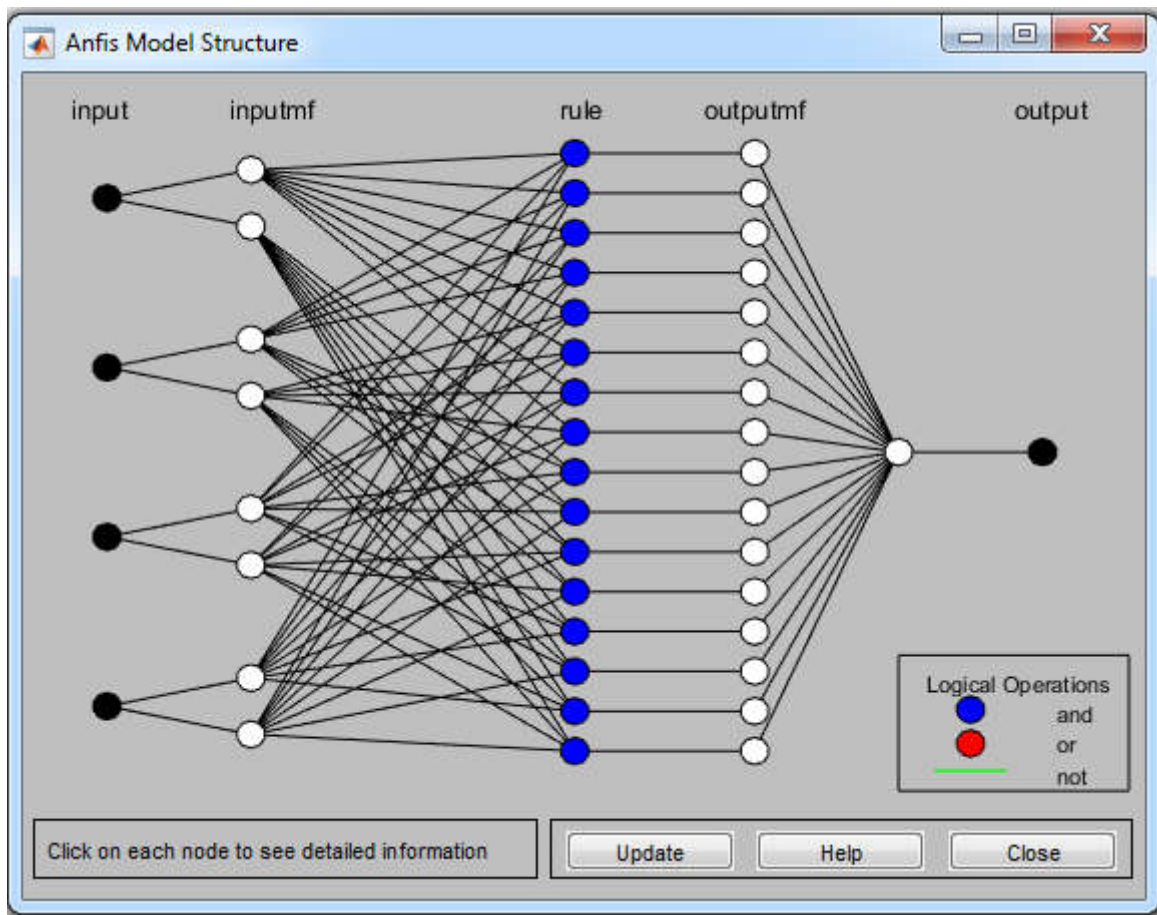


Рис. 8.9. Структура згенерованої системи нечіткого висновку

Перед навчанням гібридної мережі необхідно задати параметри навчання, для чого слід скористатися наступною групою опцій в правій нижній частині робочого вікна:

Вибрати метод навчання гібридної мережі – зворотного розповсюдження (**backpropo**) або гібридний (**hybrid**), що є комбінацією методу найменших квадратів і методу убування зворотного градієнта.

Встановити рівень помилки навчання (**Error Tolerance**) – за умовчанням значення 0 (змінювати не рекомендується).

Задати кількість циклів навчання (**Epochs**) – за умовчанням значення 3 (рекомендується збільшити і для даного прикладу задамо його значення, рівним 300).

Для навчання мережі слід натиснути кнопку **Train Now**. При цьому хід процесу навчання ілюструється у вікні візуалізації у формі графіка залежності помилки від кількості циклів навчання (рис. 8.10).

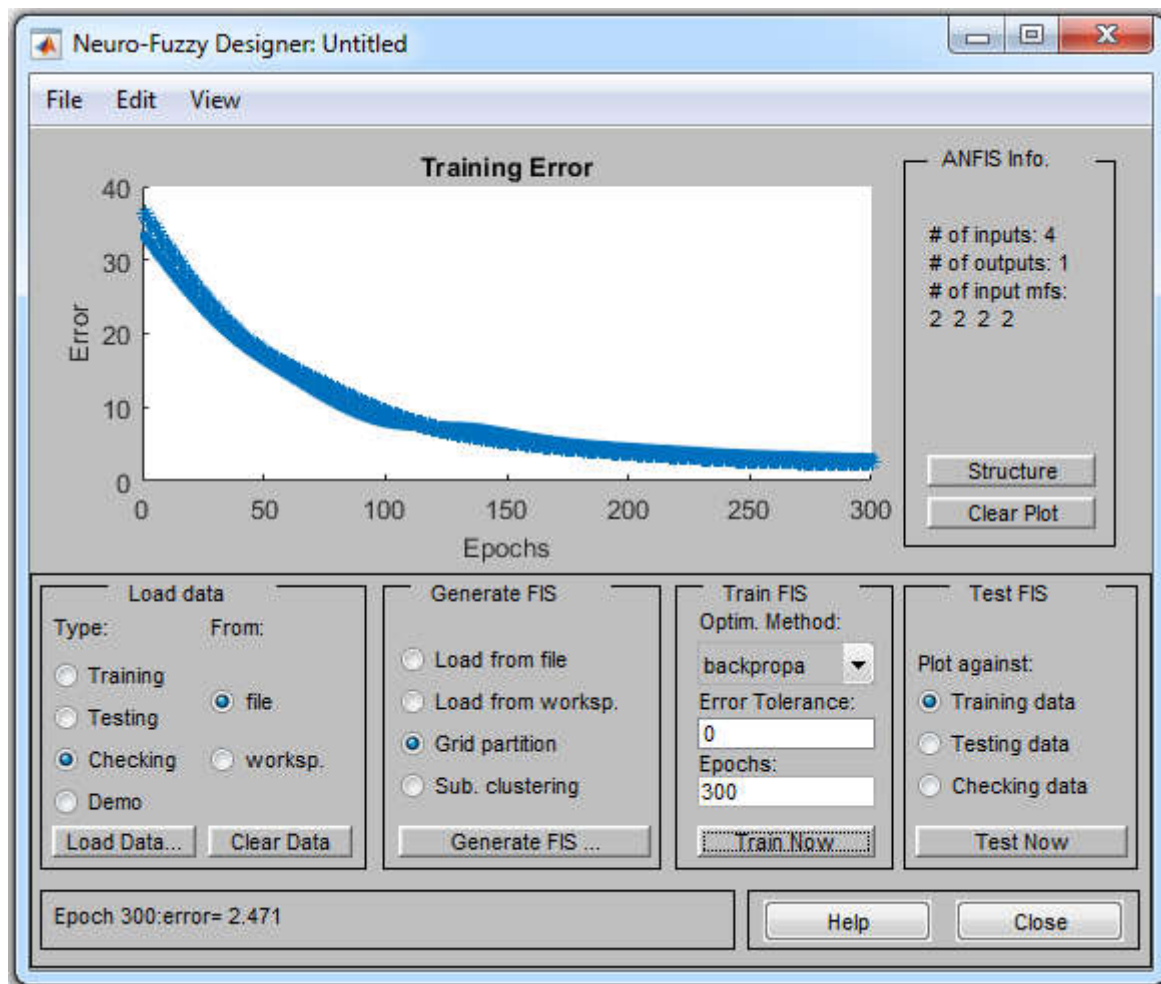


Рис. 8.10. Графік залежності помилок навчання і перевірки від кількості циклів навчання

В цьому випадку на графіці зображеному знаком «*», приведена залежність помилки навчання від кількості циклів навчання, а на графіку, зображеному «•», показана залежність помилки перевірки від кількості циклів навчання.

За допомогою опції **Test Now** в правому нижньому кутку вікна виконаємо перевірку і тестування створеної і навченої системи з виведенням результатів у вигляді графіків (відповідні графіки для навчальної вибірки – **Training data**, тестової вибірки, – **Testing data** і перевірконої вибірки – **Checking data** приведені на рис. 8.11-8.13.

Як видно з графіків, значення сигналів на виході двомасової системи електро-механічної системи управління механізмом підйому мостового крана і на виході нечіткої моделі гібридної мережі практично повністю співпадають.

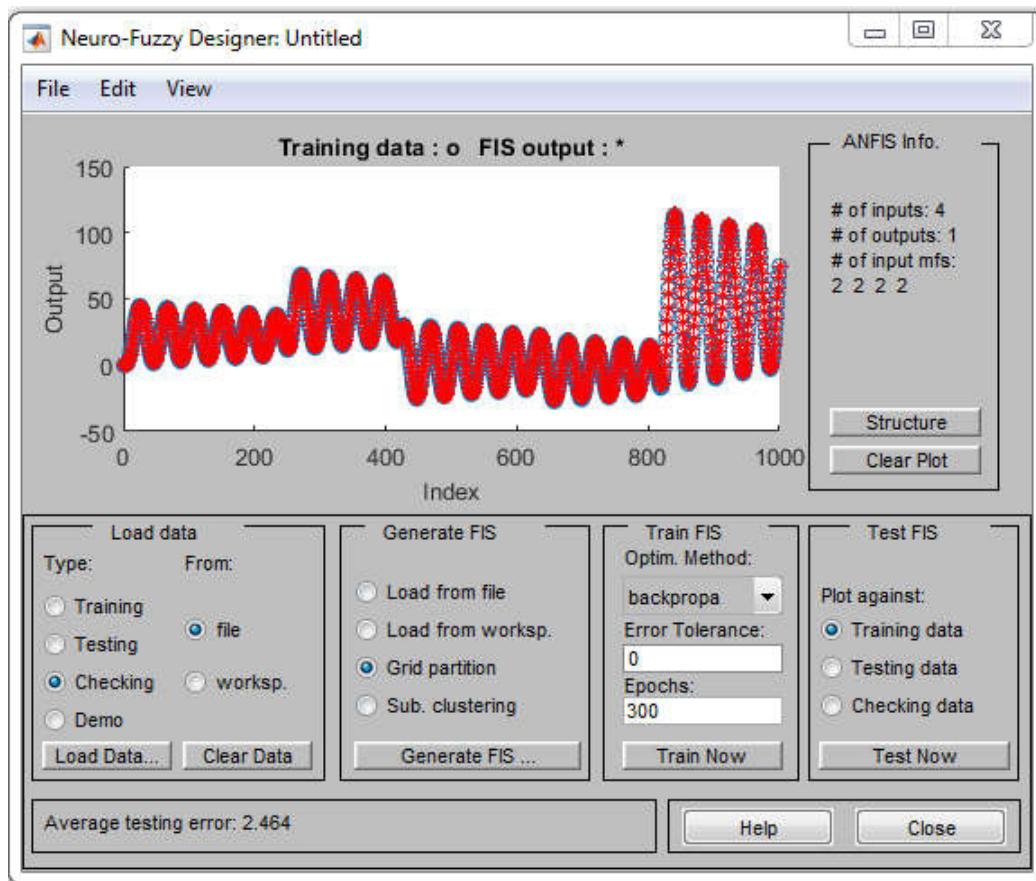


Рис. 8.11. Графіки перевірки створеної і навченої системи для навчальної вибірки – **Training data**

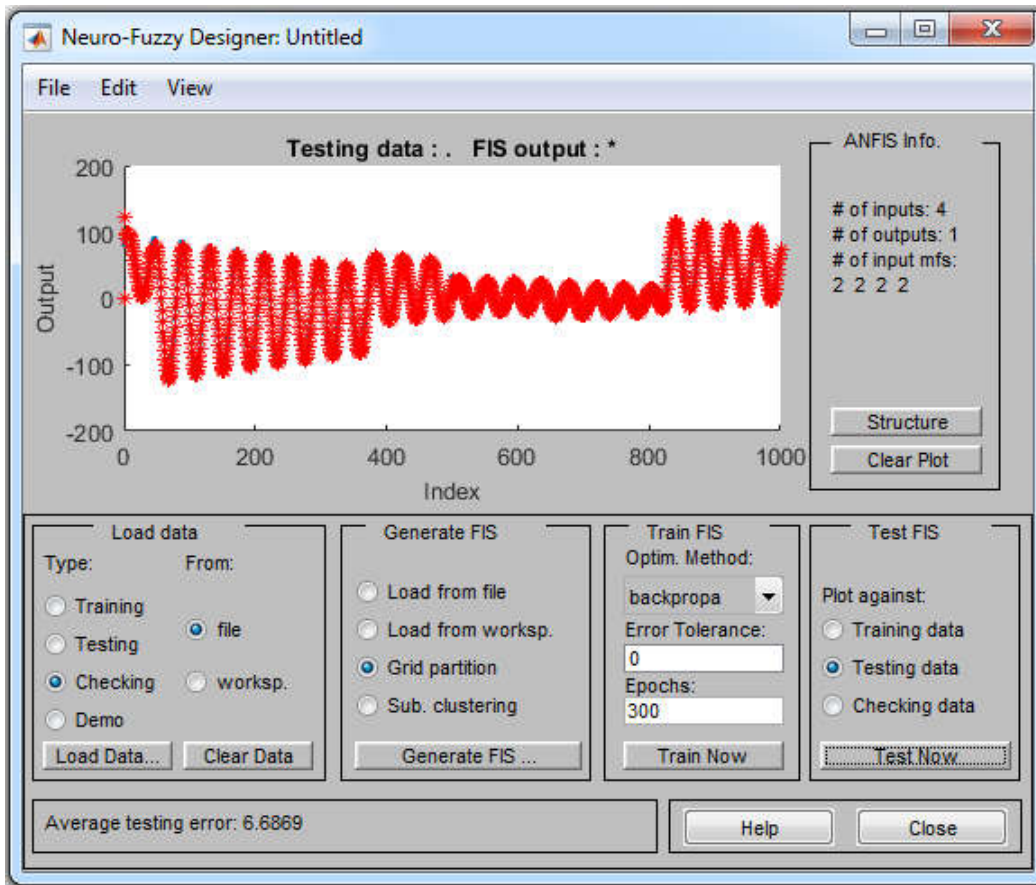


Рис. 8.12 . Графіки перевірки створеної і навченої системи для тестової вибірки – **Testing data**.

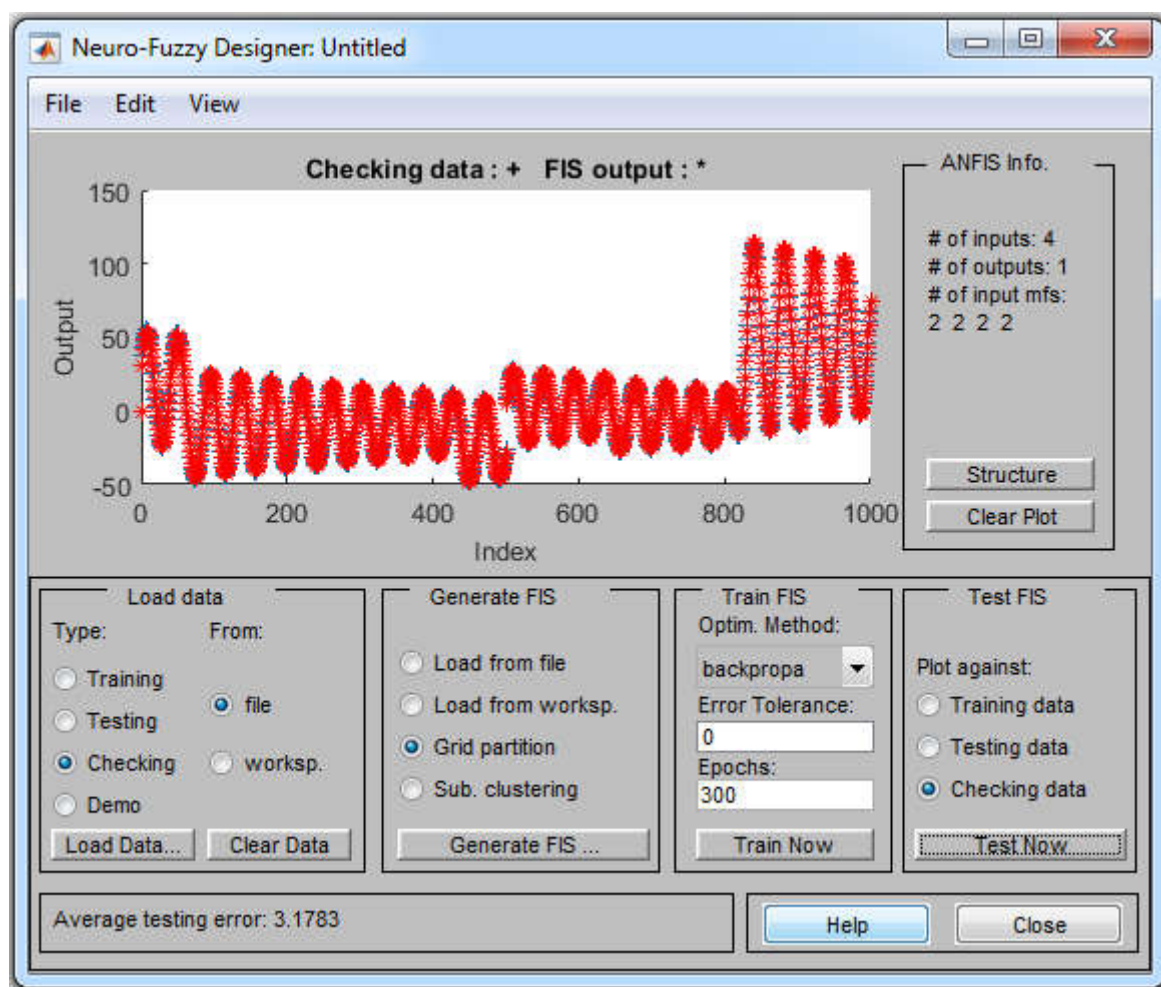


Рис. 8.13. Графіки перевірки створеної і навченої системи для перевірконої вибірки – **Checking data**.

При необхідності подальша настройка параметрів побудованої і навченої гібридної мережі може бути виконана за допомогою стандартних графічних засобів пакету Fuzzy Logic Toolbox (див. главу 7). Для цього слід зберегти створену систему нечіткого висновку в зовнішньому файлі з розширенням **fis** (збережемо у файлі з ім'ям **ANFIS_model.fis**), після чого необхідно завантажити цей файл редактор систем нечіткого виведення **FIS** (рис. 8.14).

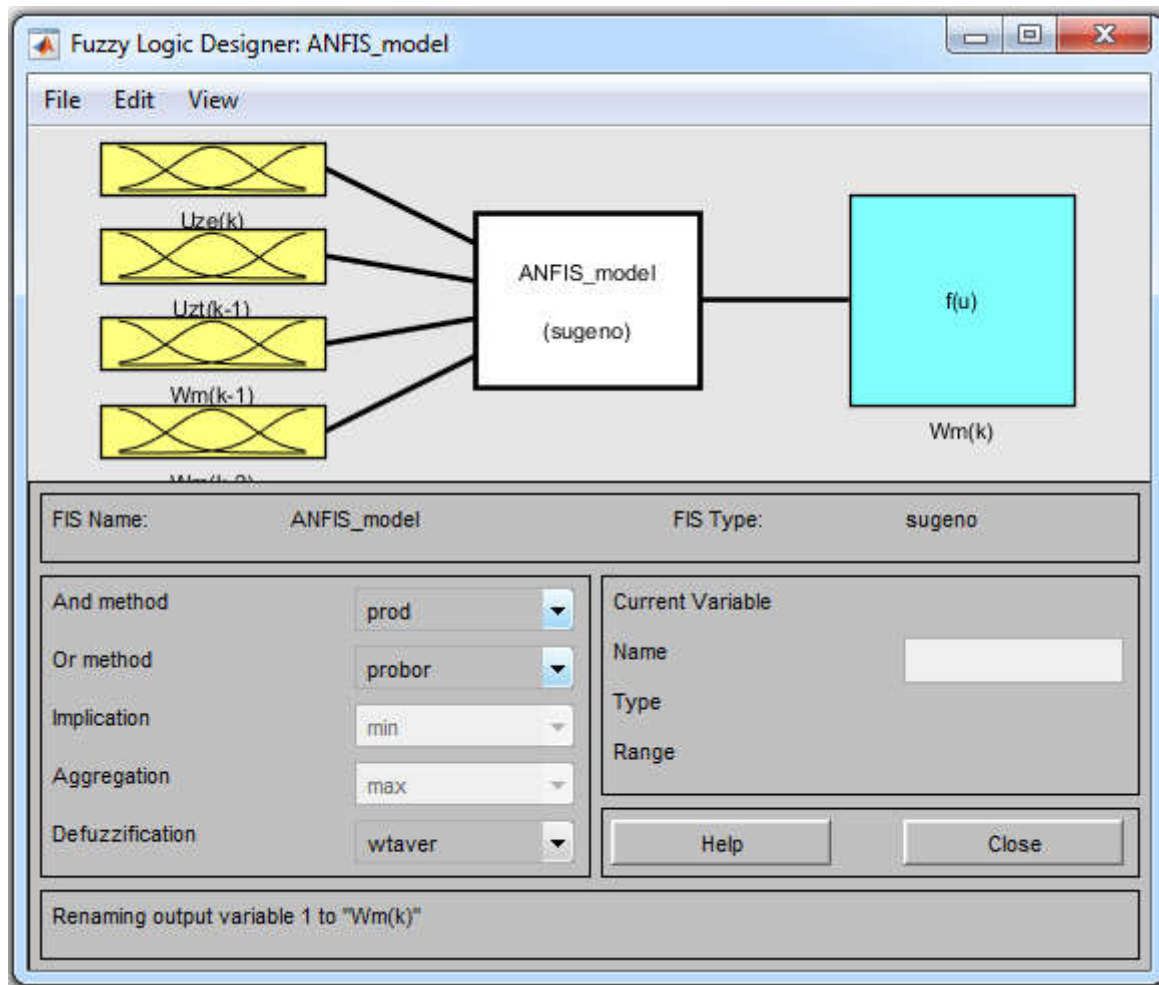


Рис. 8.14. Графічний інтерфейс редактора FIS для згенерованої системи нечіткого висновку

При цьому також стають доступними редактор функцій приналежності системи нечіткого висновку (**Membership Function Editor**), редактор правил системи нечіткого висновку (**Rule Editor**), програма перегляду правил системи нечіткого висновку (**Rule Viewer**) і програма перегляду поверхні системи нечіткого висновку (**Surface Viewer**).

Примітка. У останніх версіях системи MATLAB виклик графічних засобів редагування і перегляду моделей FIS можливий безпосередньо з головного меню редактора ANFIS.

Для дослідження побудованої моделі гібридної мережі можна скористатися програмою перегляду правил (**Rule Viewer**). Для одержання буд-якого значення вихідної змінної, необхідно задати конкретні значення вхідних змінних аналогічно загальним рекомендаціям систем нечіткого висновку. При цьому на графіці функцій приналежності вихідної змінної буде вказане шукане значення вихідної змінної (рис. 8.15).



Рис. 8.15. Графічний інтерфейс перегляду правил згенерованої системи нечіткого висновку

На додаток до цього можна виконати візуальний аналіз поверхні висновку для побудованої гібридної мережі, яка також дозволяє оцінити значення вихідної змінної. Виконати аналіз навченої і настроєної гібридної мережі можна за допомогою візуалізації поверхні нечіткого висновку. Для цієї мети слід скористатися програмою перегляду поверхні відгуку системи нечіткого висновку (рис. 8.16).

Зображена поверхня висновку може бути інтерпретована як графік функціональної залежності вихідної змінної від будь-яких двох вхідних змінних.

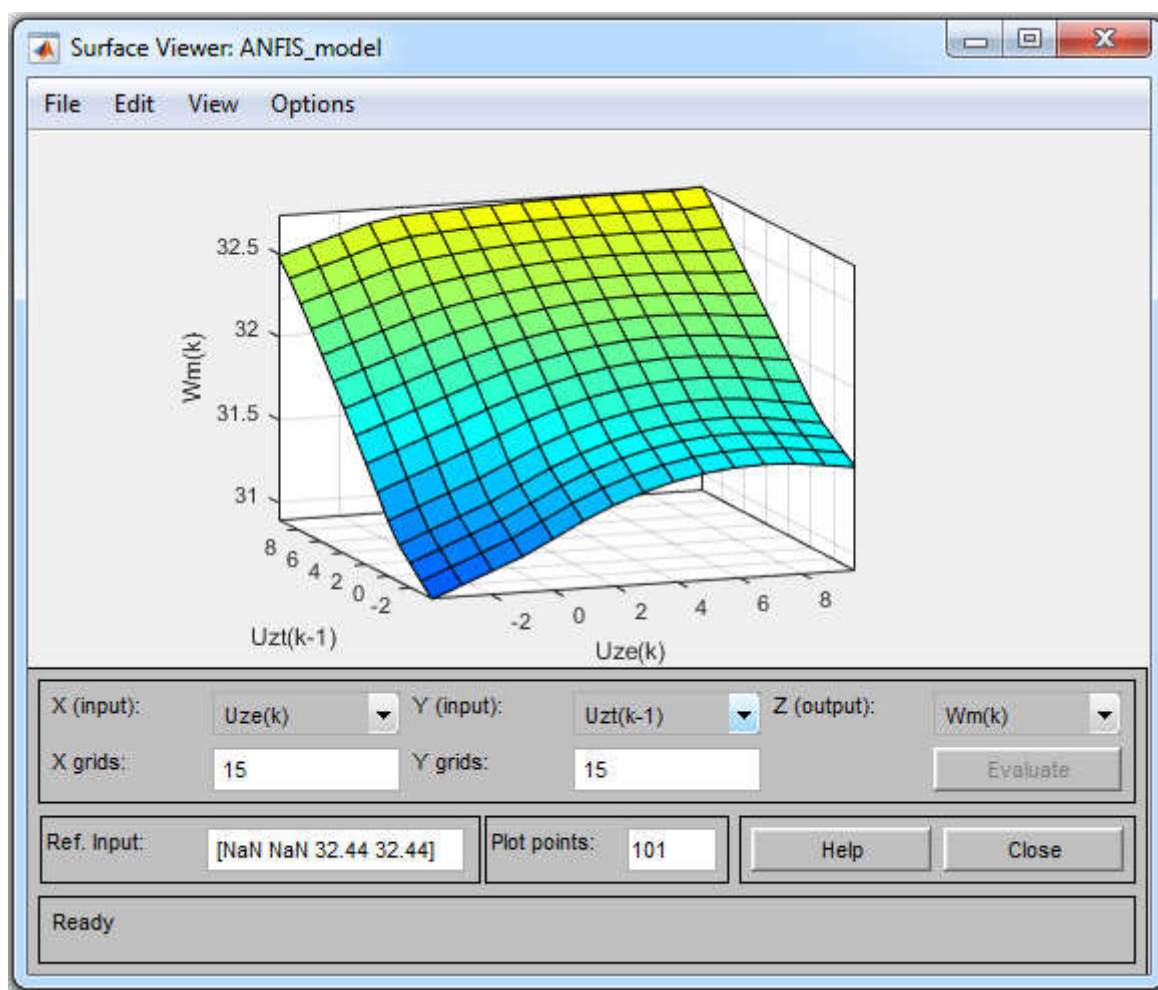


Рис. 8.16. Графічний інтерфейс перегляду поверхні висновку згенерованої системи нечіткого висновку

Виконаємо перевірку адекватності побудованої нечіткої моделі гібридної мережі. Для цієї мети використаємо схему, наведену на рис.7.11 в розділі 7.3.

У вікні завдання параметрів блоку **Fuzzy Logic Controller** у полі **FIS File or Structure** задамо ім'я згенерованої системи нечіткого виводу **ANFIS_model**.

Перед початком моделювання слід завантажити в робочу область MATLAB початкові дані двомасової системи (файл **Dat_Kran**) і розроблену нечітку модель гібридної мережі **ANFIS_model**, використовуючи команду меню **File\Export\To Workspace** у вікні редактора **ANFIS** (див. рис. 8.1). Результати моделювання представлені на рис. 8.17 і рис. 8. 18.

На рис. 8.17 приведені графіки вихідної координати двомасової електромеханічної системи управління механізмом підйому мостового крана – швидкості механізму $\omega_m(t)$. Графік 1 відповідає заданому значенню швидкості $\omega_{m_3}(t)$, графік 2 – швидкості на виході моделі двомасової системи $\omega_{m_{dc}}(t)$ і графік 3 – вихідній координ-

наті побудованої нечіткої моделі гібридної мережі $\omega_{\text{МНМ}}(t)$. Як бачимо, графіки 2 і 3 практично співпадають.

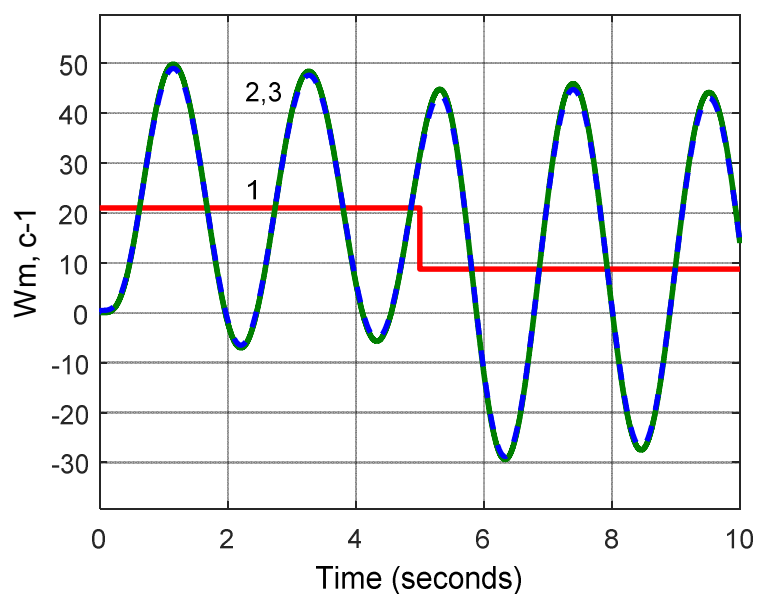


Рис. 8.17. Графіки швидкості механізму двомасової електромеханічної системи: 1 – задане значення швидкості $\omega_{\text{М}_3}(t)$; 2 – швидкість на виході моделі двомасової системи $\omega_{\text{М}_\text{дс}}(t)$; 3 – вихідна координата нечіткої моделі гібридної мережі $\omega_{\text{МНМ}}(t)$.

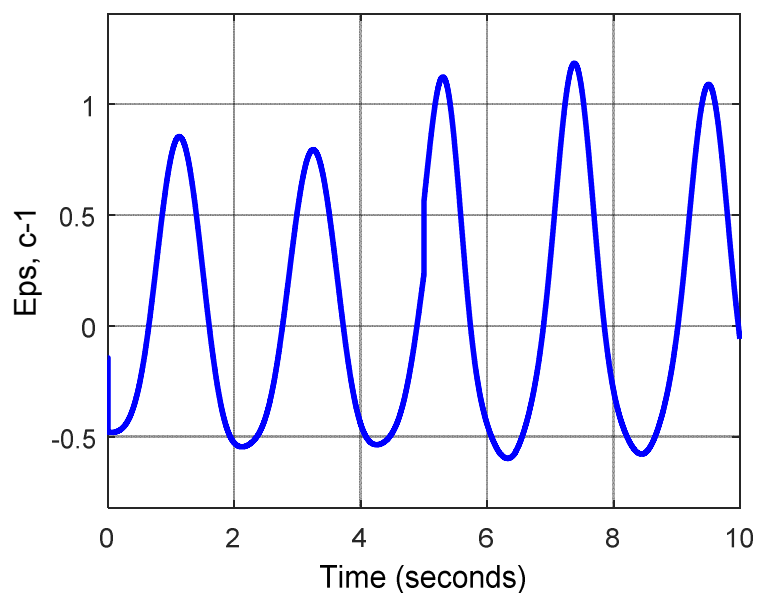


Рис. 8.18. Графік помилки ідентифікації $\varepsilon(t) = \omega_{\text{М}_\text{дс}}(t) - \omega_{\text{МНМ}}(t)$

На рис. 8.18 зображений графік різниці вказаних швидкостей: $\varepsilon(t) = \omega_{\text{мдс}}(t) - \omega_{\text{мнм}}(t)$, тобто графік помилки ідентифікації. З аналізу графіків виходить, що при зміні швидкості механізму в межах від $+50\text{с}^{-1}$ до -30с^{-1} значення $\varepsilon(t)$ знаходиться в межах від $+1,5\text{с}^{-1}$ до $-0,5\text{с}^{-1}$, тобто помилка ідентифікації не перевищує 4%.

Таким чином, перевірка побудованої нечіткої моделі гібридної мережі показує достатньо високий ступінь її адекватності реальним даним, що дозволяє зробити висновок про можливість її практичного використання для вирішення завдань ідентифікації. В цьому випадку нечіткі моделі адаптивних систем нейро-нечіткого висновку можуть вважатися новим і конструктивним інструментом побудови моделей динамічних об'єктів з складними кінематичними ланцюгами.

Розглянутий підхід є перспективним напрямом для побудови і використання відповідних нечітких моделей динамічних об'єктів і систем. Основним достоїнством даного інструменту з позицій технічного аналізу є відсутність апріорних припущень про структуру і параметри об'єктів, що цілком узгоджується з початковими передумовами побудови нечітких моделей адаптивних систем нейро-нечіткого висновку.

Питання для самоконтролю.

1. Як реалізовані гібридні мережі в пакеті прикладних програм Fuzzy Logic Toolbox системи MATLAB?
2. Поясніть призначення і можливості редактора ANFIS.
3. Перечисліть і поясніть пункти меню редактора ANFIS.
4. Наведіть методику синтезу гібридної мережі для побудови моделі двомасової електромеханічної системи.
5. Як формуються навчальні, тестові і перевірочні дані?
6. Як вибирається кількість термів вхідних і вихідного сигналів системи нечіткого висновку?
7. Як можна візуалізувати систему нечіткого висновку?
8. Які методи використовуються для навчання гібридної мережі?
9. За допомогою яких засобів можна виконувати корекцію параметрів гібридної мережі?
10. Як виконується моделювання нечіткої моделі гібридної мережі в Simulink?

СПИСОК ЛІТЕРАТУРИ

1. Zadeh L. A., Fuzzy Sets, Information and Control, 1965, Vol. 8. – P. 338-353.
2. Желдак Т.А. Нечіткі множини в системах управління та прийняття рішень: навч. посіб. / Т.А. Желдак, Л.С. Коряшкіна, С.А. Ус, за редакцією С.А. Ус ; М-во освіти і науки України, Нац. техн. ун-т «Дніпровська політехніка». – Дніпро : НТУ «ДП», 2020. – 387 с.
3. Кирик В. В. К43 Математичний апарат штучного інтелекту в електроенергетичних системах: підручник / В. В. Кирик.– Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка» 2019.– 224с.
4. Антоненко В. М., Мамченко С.Д., Рогушина Ю.В. Сучасні інформаційні системи і технології: управління знаннями: навчальний посібник. – Ірпінь : Національний університет ДПС України, 2016. – 212 с.
5. Глибовець М.М., Отецький О.В. Штучний інтелект. – К.: Вид. дім «КМ Академія», 2002. – 366 с.
6. Ямпольський Л. С. Системи штучного інтелекту в плануванні, моделюванні та управлінні : підруч. для студ. вищ. навч. закл. / Л. С. Ямпольський, Б. П. Ткач, О. І. Лісовиченко. - К. : ДП «Вид. дім «Персонал», 2011. - 544 с.
7. Методи та системи штучного інтелекту: Теорія і практика: Навчальний посібник / О.С. Булгаков, В.В. Зосімов, В.О. Поздєєв. – Одеса. : Олді плюс, 2020. – 356с.
8. Методи та системи штучного інтелекту: Навчальний посібник—/ Уклад. : А.С. Савченко, О. О. Синельников. – К. : НАУ, 2017. – 190 с.
9. Методи та системи штучного інтелекту: Навчальний посібник / Уклад.: І.М. Удовик, Г.М. Коротенко, Л.М. Коротенко, В.О. Трусів, А.Т. Харь. – Д.: Державний ВНЗ «Національний гірничий університет», 2017. – 105 с.
10. Методи та системи штучного інтелекту: навч. посіб. / укл.Д.В. Лубко, С.В. Шаров. – Мелітополь: ФОП Однорог Т.В., 2019. – 264 с.
11. Паламар М. І. Комп'ютерні технології штучного інтелекту для прецизійного управління у мехатронних системах : навч. посіб. / Михайло Паламар, Михайло Стрембіцький ; Тернопіл. нац. техн. ун-т ім. Івана Пулюя. Тернопіль : Тернопіл. нац. техн. ун-т ім. Івана Пулюя, 2018. 127 с.
12. Литвин В.В. Інтелектуальні системи : підручник / В.В. Литвин, В.В. Пасічник, Ю.В. Яцишен. – Львів: Новий світ, 2009. – 405 с.
13. Троцько В.В., Методи штучного інтелекту: навчально-методичний і практичний посібник. / Троцько В.В. – Київ: Університет економіки та права «КРОК», 2020. – 86 с.
14. Куклін В. М. К Подання знань і операції над ними; навчальний посібник. / В. М. Куклін. Харків: ХНУ імені В. Н. Каразіна, 2019 – 164 с.
15. Снитюк В. Є. Прогнозування. Моделі, методи, алгоритми / В. Є. Снитюк. – Київ : Маклаут, 2008. – 364 с.
16. Івахів, Орест Васильович. Основи побудови систем керування з нечіткою логікою : навчальний посібник /О. Івахів, М. Наконечний.– Львів :Растр-7,2017. – 129 с.

- 17.Коротка, Лариса Іванівна. Обчислювальний інтелект : теорія нечітких множин: навчальний посібник / Коротка Л.І., Зеленцов Д.Г., Науменко Н.Ю.,Ляшенко О.А., Солодка Н.О.– Дніпро :ДВНЗ УДХТУ,2020. – 161 с.
- 18.Шушура О.М. Методологічні основи побудови інформаційних технологій для автоматизації управління складними системами на принципах нечіткої логіки : дис. ... доктора технічних наук : 05.13.06 / Шушура Олексій Миколайович. К., 2018. 332 с.
- 19.Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications* (3rd ed.). Chichester: John Wiley & Sons Ltd. 852 p. ISBN 978-0-470-74376-8.
- 20.Zimmermann, H.-J. (2001). *Fuzzy Set Theory and Its Applications* (4th ed.). Boston: Kluwer Academic Publishers. 514 p. ISBN 978-0-7923-7435-0.
- 21.Pedrycz, W., & Gomide, F. (2007). *Fuzzy Systems Engineering: Toward Human-Centric Computing*. Hoboken: Wiley-IEEE Press. 544 p. ISBN 978-0-471-78800-7.
- 22.Klir, G. J., & Yuan, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Upper Saddle River: Prentice Hall. 592 p. ISBN 978-0-13-101171-7.
- 23.Klir, G. J., & Wierman, M. J. (2009). *Fuzzy Logic: A Practical Approach* (2nd ed.). Prentice Hall. 650 p.
- 24.Mendel, J. M. (2001). *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River: Prentice Hall PTR. 624 p. ISBN 978-0-13-040969-0.
- 25.Kosko, B. (1992). *Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence*. Englewood Cliffs: Prentice Hall. 449 p. ISBN 978-0-13-611435-2.
- 26.Bojadziev, G., & Bojadziev, M. (2007). *Fuzzy Logic for Business, Finance, and Management* (2nd ed.). Singapore: World Scientific Publishing Company. 256 p. ISBN 978-981-270-649-9.
- 27.Karnik, N. N., & Mendel, J. M. (2001). *Centroid of a Fuzzy Set*. *IEEE Transactions on Fuzzy Systems*, 9(5), 769–778. doi:10.1109/91.940962.
- 28.Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Prentice Hall. ISBN 978-0-13-261066-7.
- 29.Chen, G., & Pham, T. T. (2001). *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. Boca Raton: CRC Press. 328 p. ISBN 978-0367397883.
- 30.Vukadinovic, D. (2010). *Fuzzy Control Systems*. New York: Nova Science Publishers. 161 p. ISBN 978-1613244881.
- 31.Pedrycz, W., & Gomide, F. (2007). *An Introduction to Fuzzy Sets: Analysis and Design*. MIT Press. 450 p.
32. Pedrycz, W., & Gomide, F. (2023). *Modern Adaptive Fuzzy Control Systems*. Cham: Springer. 375 p. ISBN 978-3031173929.
- 33.Mendel, J. M. (2017). *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions* (2nd ed.). Cham: Springer. 684 p. ISBN 978-3319513690.

34. Tsoukalas, L. H., & Uhrig, R. E. (1997). *Fuzzy and Neural Approaches in Engineering*. New York: John Wiley & Sons. 587 p. ISBN 978-0471160038.
35. Mendel, J. M. (2001). *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*. Upper Saddle River: Prentice Hall PTR. 624 p. ISBN 978-0-13-040969-0.
36. Kuo, B. C., & Golnaraghi, F. (2003). *Automatic Control Systems* (8th ed.). Wiley-Interscience. 500 p.
37. Ross, T. J. (2010). *Fuzzy Logic with Engineering Applications* (3rd ed.). Wiley. 672 p.
38. Mamdani E. H. Applications of fuzzy logic to approximate reasoning using linguistic synthesis. – IEEE Transactions on Computers, Vol. 26, №. 12, 1977. – P. 1182-1191.
39. Ross T. J. Fuzzy logic with engineering applications. – McGraw-Hill, 1995. – 600 p.
40. Takagi T., Sugeno M. Fuzzy identification of systems and its applications to modeling and control. – IEEE Transactions on Systems, Man, and Cybernetics, Vol. 15, №. 1, 1985. – P. 116-132.
41. Bezdek J. C, Pal S. K, (red.), Fuzzy Models for Pattern Recognition, IEEE, New York, 1992.
42. Brown M., Harris C, Neurofuzzy Adaptive Modelling and Control, Prentice Hall, New York, 1994.
43. Cox L., The Fuzzy Systems Handbook, Academic Press, London, 1994.
44. De Silva C. W., Intelligent Control: Fuzzy Logic Applications, CRC Press, Boca Raton, 1995.
45. Driankov D., Hellendoom H., Reinfrank M. An Introduction to Fuzzy Control, Springer-Verlag, Berlin, 1993.
46. Harris C. J., Moore C. G., Brown M., Intelligent Control: Aspects of fuzzy logic and neural nets. World Scientific, Singapore, 1993.
47. Soger R., Fuzzy Logic in Control, Thesis Technische Universiteit Delft Delft, 1995.
48. Jamshidi M., Vadiiee N. Ross T. J., (red.), Fuzzy Logic and Control, Prentice Hall, Englewood, 1993.
49. Kosko B., Neural Networks and Fuzzy Systems, Prentice Hall, Englewood Cliffs, 1992.
50. Kruse R., Gebhardt J., Klawonn R, Foundations of Fuzzy Systems, John Wiley, Chichester, 1994.
51. Lee C. C, Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part 1, IEEE Transactions on Systems, Man and Cybernetics, 1990, Vol. 20, № 2. – P. 404-418.
52. Lee C. C, Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part II, IEEE Transactions on Systems, Man and Cybernetics, 1990, Vol. 20, №. 2. – P. 419-435.
53. Nie J., Linkens D., Fuzzy-Neural Control, Prentice Hall, New York, 1995.
54. Pedrycz V, Fuzzy Control and Fuzzy Systems, New York, John Wiley, 1993.
55. Slupecki J., Halkowska K, Pirog-Rzepecka K, Logika i teoria mnogosci, PWN, Warszawa, 1994.
56. Terano T, Asai K, Sugeno M. Fuzzy Systems Theory and its Applications, Academic Press, London, 1992.

57. Wang L. X. Adaptive Fuzzy Systems and Control – Design and Stability Analysis, Prentice Hall, Englewood Cliffs, 1994.
58. Yager R. R., Filev D. P. Podstawy modelowania i sterowania rozmytego, WNT, Warszawa, 1995.
59. Yan J., Ryan M., Power J., Using Fuzzy Logic, Prentice Hall, London, 1994.
60. Zimmermann H. J., Fuzzy Set Theory, Kluwer Academic Publishers, Boston // Dordrecht/London, 1994.
61. Korbicz J., Obuchowicz A., Ucinski D. Sztuczne sieci neuronowe. Podstawy i zastosowania, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994.
62. Osowsta S., Sieci Neuronowe, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa, 1994.
63. Stacewicz R. Zastosowania sieci neuronowych do wnioskowan rozmytych, Prace Instytutu Podstaw Informatyki PAN, Warszawa, Lipiec 1995.
64. Tadeusiewicz R. Sieci neuronowe, Akademicka Oficyna Wydawnicza RM, Warszawa 1993.
65. Wang L. X. Adaptive Fuzzy Systems and Control. PTR Prentice Hall. Englewood Cliffs, 1994.
66. Zurada J. Introduction to Artificial Neural Systems, West Publishing Company, Boston, 1994.
67. Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River: Prentice Hall. 614 p. ISBN 978-0-13-261066-7.
68. Lin, C.-T., & Lee, C. S. G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River: Prentice Hall. 800 p. ISBN 978-0-13-235169-0.
69. Kasabov, N. K. (2007). *Evolving Connectionist Systems: The Knowledge Engineering Approach*. London: Springer. 400 p. ISBN 978-1-84628-346-8.
70. Angelov, P., Filev, D., & Kasabov, N. (2010). *Evolving Intelligent Systems: Methodology and Applications*. Hoboken: Wiley. 440 p. ISBN 978-0-470-28719-4.
71. Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence*. Upper Saddle River: Prentice Hall. 614 p. ISBN 978-0-13-261066-7.
72. Lin, C.-T., & Lee, C. S. G. (1996). *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Upper Saddle River: Prentice Hall. 800 p. ISBN 978-0-13-235169-0.
73. Abraham, A. (2005). *Adaptation of Fuzzy Inference System Using Neural Learning*. Y *Fuzzy Systems Engineering: Theory and Practice* (ред. N. Nedjah, L. de Macedo Mourelle), с. 53–83. Berlin: Springer. ISBN 978-3-540-25055-4.
74. Дьяконов В. П., Круглов В. В. MATLAB 6.5 SP1/SP2 + Simulink 5/6 Инструменты искусственного интеллекта и биоинформатики. Серия «Библиотека профессионала». – М. : СОЛОН-ПРЕСС, 2006. – 456 с.
75. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

- 76.Круглов В.В., Дли М.И. Интеллектуальные информационные системы: компьютерная поддержка систем нечёткой логики и нечёткого вывода. – М.: Физматлит, 2002. – 252 с.
- 77.Леоненко А. В. Нечёткое моделирование в среде MATLAB и fuzzy TECH – СПб. : БХП – Петербург, 2003. – 736 с.
- 78.Штовба С.Д. Проектирование нечетких систем средствами MATLAB / С.Д. Штовба – М.: "Горячая линия - Телеком", 2007. – 288 с. – 284 с.
- 79.MathWorks. (2021). *Fuzzy Logic Toolbox User's Guide*. The MathWorks, Inc. Доступно за посиланням: https://www.mathworks.com/help/pdf_doc/fuzzy/fuzzy.pdf
- 80.Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрьоменко. Харків, ХНАДУ, 2012. – 284 с.
- 81.Модельовання систем управління в SIMULINK : навч. посібник / В. О. Богомоллов, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьев, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

Електронне навчальне видання комбінованого використання
Можна використовувати в локальному мережному режимі

Канюк Геннадій Іванович
Василець Тетяна Юхимівна
Варфоломієв Олексій Олексійович

ТЕХНОЛОГІЇ НЕЙРОННИХ МЕРЕЖ І НЕЧІТКОГО МОДЕЛЮВАННЯ В СИСТЕМАХ УПРАВЛІННЯ

У двох частинах

Частина 2

Навчально-методичний посібник
для здобувачів вищої освіти другого (магістерського) рівня
за спеціальністю 174 «Автоматизація, комп'ютерно-
інтегровані технології та робототехніка»

В авторській редакції

Підписано до розміщення 28.03.2025. Гарнітура Times New Roman.
Ум. друк. арк. 2,42. Обсяг 2,800 Кб. Зам. № 112/25.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2009
Видавництво ХНУ імені В. Н. Каразіна