

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ДОНЕЦЬКИЙ ДЕРЖАВНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
МАГДЕБУРГСЬКИЙ УНІВЕРСИТЕТ ім. ОТТО-ФОН-ГЕРІКЕ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

Калашніков В.І. , Паліс Ф. , Лозинський О.Ю.

**Основи теорії фази-логіки
та фази-регулювання**

Донецьк , Магдебург , Львів
2000

УДК 62-83-52-0313

“Основи теорії фази-логіки та фази-регулювання”: учбовий посібник/ Калашніков В.І. ,
Паліс Ф. , Лозинський О.Ю. - Донецьк, Магдебург ,Львів, - 2000 - 69с.

Розглянуто основи теорії фази-логіки та фази-регулювання стосовно впровадження інтелектуальних систем керування в галузі керованого електроприводу. Учбовий посібник спрямовано на придбання студентами навичок дослідження та проектування подібних систем за допомогою програмного пакета MATLAB / SIMULINK. Це дозволяє без будь-яких обмежень на лабораторне оснащення використовувати посібник як методичні матеріали для організації навчального процесу в будь-якому вищому навчальному закладі.

Навчальний посібник підготовлено в рамках програми інтеграції вищої школи України в європейську систему інженерної освіти і орієнтовано на студентів спеціальності “Електромеханічні системи автоматизації та електропривод”.

Автори виражають щире подяку студентам, магістрантам та науковим співробітникам Донецького технічного університету, Магдебургського ім. Отто-фон-Геріке університету та Львівського університету “Львівська політехніка”, які допомогали в підготовці посібника та його практичній реалізації в навчальному процесі вузів-партнерів.

Комп'ютерна верстка Александрової Ю. І.

Рецензенти:

Клепиков В.Б. – д.т.н., проф., завідувач кафедри “Автоматизовані електро-механічні системи” Харківського технічного університету.

Коцегуб П.Х. – д.т.н., проф., завідувач кафедри “Електропривод та автоматизація промислових установок” Донецького державного технічного університету.

ЗМІСТ

ВСТУП.....	4
1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ.....	7
1.1 Фази-множини та основні операції над ними.....	7
1.2 Фази-контролер.....	13
1.3 Основні етапи проектування фази-контролерів.....	25
1.4 Типи фази-контролерів.....	27
1.5 Варіанти побудови контурів регулювання із фази-регулятором.....	29
2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK.....	34
3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ ФАЗИ - ЛОГІКИ В СИСТЕМАХ РЕГУЛЮВАННЯ.....	44
3.1 Постановка завдання.....	44
3.2 Лабораторний практикум.....	48
Лабораторна робота 1. Використання неадаптивного фази-регулятора в системі автоматизації.....	48
Лабораторна робота 2. Адаптація фази-логіки в системах регулювання.....	59
ЛІТЕРАТУРА.....	69

ВСТУП

Давнім прагненням людства було створення приладів, що були б здатні полегшити людську працю. Спочатку йшлося про інструменти та машини, які механізували тільки фізичну працю, але пізніше постало питання розробки "розумних машин", які функціонували б подібно до людського мозку. Така постановка завдання викликає наступні принципові питання:

- ✓ чи є взагалі можливим створення машини, здатної до інтелектуальної діяльності;
- ✓ чим характеризується інтелектуальність людини та на чому вона ґрунтується;
- ✓ які головні принципи мають лягти в основі так званого "штучного інтелекту".

Спочатку треба пояснити, що означає поняття "інтелектуальність" стосовно до людини. Інтелектуальність (розумність) у такому контексті можна визначити як наявність комплексної когнітивної здатності, за допомогою якої людина впізнає об'єкти, явища та відповідно до ситуації на них реагує. Серед типових характерних рис інтелектуальності слід зазначити такі:

- ✓ усвідомлення ситуації, виходячи із багатозначної або суперечливої інформації;
- ✓ знаходження аналогій, здатність до класифікації;
- ✓ здатність до вивчення нових понять та встановлення зв'язків між різними поняттями.

Завданням теорії штучного інтелекту як області прикладної математики є знаходження засобів формалізації процесів усвідомлення та пізнання та переніс їх на мову алгоритмів. При цьому розумова діяльність постає як засіб обробки інформації. Це означає, що теорія штучного інтелекту виходить з того, що діяльність людського розуму в певній мірі може бути розглянута незалежно від функцій мозку, тобто достатньо знати основні принципи розумної поведінки, щоб записати їх у вигляді алгоритмів, зрозумілих для комп'ютера. Таким чином, інженерно-науковий аспект теорії штучного інтелекту полягає в проектуванні складних систем обробки інформації. Це завдання включає в себе розробку як методів обробки інформації, так і відповідних пристроїв та програмного забезпечення.

В даний час можна виділити чотири напрямки розвитку комп'ютерного інтелекту:

- ✓ фази-логіка;
- ✓ експертні системи;
- ✓ нейросітки;
- ✓ еволюційне моделювання.

ВСТУП

Перед теорією штучного інтелекту стоїть завдання створення комп'ютерних систем, які були б здатні вирішувати такі задачі:

1. **вирішення проблем.** Розв'язання знаходиться шляхом зведення наявної інформації до відомих елементарних висловлень та пошуку в базі знань необхідної інформації;
2. **розпізнавання та розуміння людського мовлення;**
3. **розпізнавання та обробка зображень,** тобто впізнавання із візуальної інформації окремих об'єктів та груп об'єктів. Цей процес вимагає класифікації об'єктів за певними характерними ознаками;
4. **навчання.** Знання має автоматично накопичуватись з метою покращення властивостей системи;
5. **експертні оцінки.** Специфічні для певної області знання мають бути збережені та оброблені комп'ютером таким чином, щоб система могла самостійно приймати рішення, подібно спеціалісту з цієї галузі;
6. **якісні висновки.** Ця область застосовується при аналізі процесів, які мають не чіткий кількісний, а якісний опис.

Вирішення першого, четвертого та шостого завдань тісно пов'язане із фази-логікою. Фази-логіка використовується в інформатиці, техніці регулювання та математиці для моделювання експертного знання та вирішення задач автоматизації. Експертні знання при цьому представляються у формі *якщо...то*-правил із якісними поняттями, такими, як *велика*, *маленька*. Застосування такого методу обробки інформації є доцільним при створенні систем регулювання для об'єктів, які описуються через велику кількість параметрів (особливо якщо ці параметри змінюються за нелінійними законами) і мають досить складну математичну модель.

Слід зазначити, що фази-логіка була взята інженерами на "озброєння" ще в сімдесяті роки. Звичайно, що сьогодні вона не відповідає сучасним вимогам, які ставляться до систем програмного керування, і тому застосування її у "чистому вигляді" є дуже рідкісним. Удосконалити фази-логіку, зробити її спроможною до адаптації в системах регулювання були покликані інші типи штучного інтелекту. Насамперед це нейросітки, експертні системи та еволюційні методи.

Дані методичні вказівки містять основи теорії фаззи-логіки та фаззи-регулювання, огляд засобів побудови на її основі фаззи-регуляторів, основних типів контурів регулювання

із фази-контролером. Практичному застосуванню фази-логіки в системах автоматизації присвячені дві лабораторні роботи.

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

1.1 Фази-множини та основні операції над ними

Фази-логіка – технологія розробки систем управління, яка базується на інтуїції експерта. Замість чіткого математичного опису фази-логіка використовує для опису поведінки системи лінгвістичні визначення (лінгвістичні змінні), значеннями яких є мовні конструкції, а не числові значення. Ці лінгвістичні змінні моделюються за допомогою так званих нечітких множин. Нечітка множина – множина, елементи якої належать до неї з тим чи іншим ступенем приналежності. Це означає, що кожному елементу цієї множини відповідає ступінь приналежності, що знаходиться в інтервалі від 0% до 100%. Математично нечітка множина записується таким чином:

$$\mathbf{A} = \{ [x_1, \mu_A(x_1)] \dots [x_n, \mu_A(x_n)] \} \quad (1.1)$$

Це означає, що величина x_1 належить до нечіткої множини \mathbf{A} із ступенем приналежності $\mu_A(x_1)$.

Елементи чіткої (класичної) множини можуть приймати тільки два значення ступеня приналежності – 0% та 100%. Математично чітка множина записується таким чином:

$$\mathbf{A} = \{ x_1, x_2 \dots x_n \} \quad (1.2)$$

Як правило для визначення фази-множини \mathbf{A} замість упорядкованої пари $x_1, \mu_A(x_1)$ використовують так звану функцію приналежності $\mu_A(x)$. Функція приналежності показує ступінь приналежності конкретного значення до певної нечіткої множини в області зміни вимірюваної величини, що підлягає розгляду. Функція приналежності може приймати різноманітні форми, найбільш поширені з них наведені в табл. 1.1.

Обробка фази-множин вимагає відповідних операторів, за допомогою яких реалізується зв'язок та модифікація фази-множин. Ці оператори розподіляються на дві групи:

- ✓ оператори, що пов'язують фази-множини єдиної базової множини;
- ✓ оператори, що обробляють фази-множини окремих базових множин.

Таблиця 1.1 – Функції приналежності

Тип функції	Математичне визначення	Графічне зображення
Монотонна (лінійна)	$\mu_A(x) = \begin{cases} 0 & \text{при } x < a \\ \frac{x-a}{b-a} & \text{при } a \leq x \leq b \\ 1 & \text{при } b \leq x \end{cases}$	
Трикутна	$\mu_A(x) = \begin{cases} 0 & \text{при } x < a \vee d \leq x \\ \frac{x-a}{m-a} & \text{при } a \leq x \leq m \\ \frac{x-d}{m-d} & \text{при } m \leq x \leq d \end{cases}$	
Трапецієподібна	$\mu_A(x) = \begin{cases} 0 & \text{при } x < a \vee d \leq x \\ \frac{x-a}{b-a} & \text{при } a \leq x \leq b \\ 1 & \text{при } b \leq x \leq c \\ \frac{x-d}{c-d} & \text{при } c \leq x \leq d \end{cases}$	
Прямокутна	$\mu_A(x) = \begin{cases} 0 & \text{при } x < a \vee b < x \\ 1 & \text{при } a \leq x \leq b \end{cases}$	
Функція Гауса	$\mu_A(x) = e^{-a \cdot (x-m)^2} \quad a > 0$	
Синглетон-функція	$\mu_A(x) = \begin{cases} 0 & \text{при } x = m \\ 1 & \text{при } x \neq m \end{cases}$	

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

В першій групі операцій із фази-множинами виділяють три основних операції:

- ✓ **I.** Реалізує перетин фази-множин. Перетин фази-множин $A_1 \cap A_2$ визначається таким чином:

$$\forall x \in X: \mu_{A_1 \cap A_2}(x) = \min \left\{ \mu_{A_1}(x), \mu_{A_2}(x) \right\} \quad (1.3)$$

- ✓ **АБО.** Реалізує поєднання фази-множин. Поєднання двох фази-множин $A_1 \cup A_2$ визначається таким чином:

$$\forall x \in X: \mu_{A_1 \cup A_2}(x) = \max \left\{ \mu_{A_1}(x), \mu_{A_2}(x) \right\} \quad (1.4)$$

- ✓ **III.** Реалізує логічне заперечення фази-множини. Логічне заперечення $\neg A$ фази-множини A визначається таким чином:

$$\forall x \in X: \mu_{\neg A}(x) = 1 - \mu_A(x) \quad (1.5)$$

На рис. 1.1 подається графічне зображення поданих вище визначень. Логічні операції над фази-множинами можуть бути розглянуті як узагальнення класичних операцій над множинами.

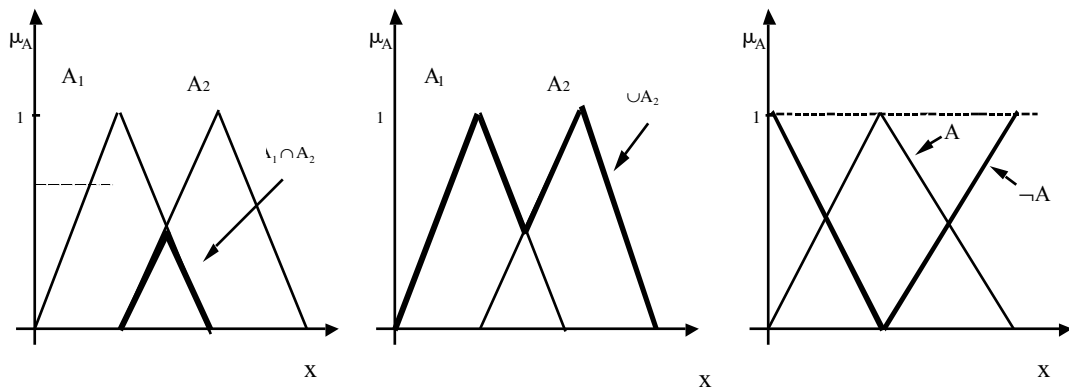


Рисунок 1.1 – Операції із фази-множинами

В процесі подальшого розвитку фази-логіки були визначені також інші оператори для фази-множин. Ці оператори розрізняються загальнозживаністю, сумісністю та насамперед ступенем вірогідності, що змінюється в межах від інтуїтивно аргументованих до аксіоматично обґрунтованих визначень. Розрізняють два основних класи таких операторів:

- ✓ Т-норми, що використовуються для моделювання перетину фази-множин;
- ✓ Т-Со-норми (також інколи позначаються як S-норми), що використовуються для моделювання поєднання фази-множин.

Функція $T: [0,1] \times [0,1] \rightarrow [0,1]$ називається T -нормою, якщо вона відповідає таким умовам:

1. $T(0,0) = 0$; $T(\mu_A(x), 1) = T(1, \mu_A(x)) = \mu_A(x) \quad \forall x \in X$
2. $T(\mu_{A1}(x), \mu_{A2}(x)) \leq T(\mu_{A3}(x), \mu_{A4}(x))$
якщо $\mu_{A1}(x) \leq \mu_{A3}(x)$ та $\mu_{A2}(x) \leq \mu_{A4}(x)$ (умова монотонності)
3. $T(\mu_{A1}(x), \mu_{A2}(x)) = T(\mu_{A2}(x), \mu_{A1}(x))$ (умова комутативності)
4. $T(\mu_{A1}(x), T(\mu_{A2}(x), \mu_3(x))) = T(T(\mu_{A1}(x), \mu_{A2}(x)), \mu_3(x))$ (умова асоціативності)

Функція $S: [0,1] \times [0,1] \rightarrow [0,1]$ називається T -Со-нормою або S -нормою, якщо вона відповідає таким умовам:

1. $S(1,1) = 1$; $S(\mu_A(x), 0) = S(0, \mu_A(x)) = \mu_A(x) \quad \forall x \in X$
2. $S(\mu_{A1}(x), \mu_{A2}(x)) \leq S(\mu_{A3}(x), \mu_{A4}(x))$
якщо $\mu_{A1}(x) \leq \mu_{A3}(x)$ та $\mu_{A2}(x) \leq \mu_{A4}(x)$ (умова монотонності)
3. $S(\mu_{A1}(x), \mu_{A2}(x)) = S(\mu_{A2}(x), \mu_{A1}(x))$ (умова комутативності)
4. $S(\mu_{A1}(x), T(\mu_{A2}(x), \mu_3(x))) = S(T(\mu_{A1}(x), \mu_{A2}(x)), \mu_3(x))$ (умова асоціативності)

Оператори обох цих класів можуть бути параметризованими та непараметризованими.

Табл. 1.2 надає огляд найбільш відомих непараметризованих T -норм та T -Со-норм.

Таблиця 1.2 – Непараметризовані оператори

Непараметризовані оператори	T -норма: $T(\mu_{A1}(x), \mu_{A2}(x)) =$	T -Со-норма або S -норма: $S(\mu_{A1}(x), \mu_{A2}(x)) =$
максимум/мінімум T_{\min}/S_{\max}	$\min\{\mu_{A1}(x), \mu_{A2}(x)\}$	$\max\{\mu_{A1}(x), \mu_{A2}(x)\}$
Обмежена різниця/ Обмежена сума $T_{\text{обмеж}}/S_{\text{обмеж}}$	$\max\{0, \mu_{A1}(x) + \mu_{A2}(x) - 1\}$	$\min\{1, \mu_{A1}(x) + \mu_{A2}(x)\}$

Продовження табл. 1.2

Алгебраїчний добуток/ Алгебраїчна сума $T_{алгебр}/S_{алгебр}$	$\mu_{A1}(x) \cdot \mu_{A2}(x)$	$\mu_{A1}(x) + \mu_{A2}(x) - \mu_{A1}(x) \cdot \mu_{A2}(x)$
Drastisches-добуток/ Drastisches-сума $T_{drastisch}/S_{drastisch}$	$\min\{\mu_{A1}(x), \mu_{A2}(x)\}$ якщо $\max\{\mu_{A1}(x), \mu_{A2}(x)\} = 1$ 0 інакше	$\max\{\mu_{A1}(x), \mu_{A2}(x)\}$ якщо $\min\{\mu_{A1}(x), \mu_{A2}(x)\} = 0$ 1 інакше
Hamacher-добуток/ Hamacher-сума $T_{Hamacher}/S_{Hamacher}$	$\frac{\mu_{A1}(x) \cdot \mu_{A2}(x)}{\mu_{A1}(x) + \mu_{A2}(x) - \mu_{A1}(x) \cdot \mu_{A2}(x)}$	$\frac{\mu_{A1}(x) + \mu_{A2}(x) - 2 \cdot \mu_{A1}(x) \cdot \mu_{A2}(x)}{1 - \mu_{A1}(x) \cdot \mu_{A2}(x)}$
Добуток Ейнштейна/ Сума Ейнштейна $T_{Ейншт}/S_{Ейншт}$	$\frac{\mu_{A1}(x) \cdot \mu_{A2}(x)}{2 - [\mu_{A1}(x) + \mu_{A2}(x) - \mu_{A1}(x) \cdot \mu_{A2}(x)]}$	$\frac{\mu_{A1}(x) + \mu_{A2}(x)}{1 + \mu_{A1}(x) \cdot \mu_{A2}(x)}$

Властивості параметризованих операторів на відміну від непараметризованих можуть змінюватися в широких межах залежно від області їх застосування за допомогою відповідного параметра. Деякі приклади таких операторів наведені в табл. 1.3.

Таблиця 1.3 – Параметризовані оператори

Параметризовані оператори	T-норма: $T(\mu_{A1}(x), \mu_{A2}(x), \gamma) =$	T-Co-норма або S-норма: $S(\mu_{A1}(x), \mu_{A2}(x), \gamma) =$
Hamacher-перетин/ поєднання $T_{Ham_перетин}/S_{Ham_поєднання}$	$\frac{\mu_{A1}(x) \cdot \mu_{A2}(x)}{\gamma + (1-\gamma) [\mu_{A1}(x) + \mu_{A2}(x) - \mu_{A1}(x) \cdot \mu_{A2}(x)]}$ $\gamma \geq 0$	$\frac{\mu_{A1}(x) + \mu_{A2}(x) - (2-\gamma) \mu_{A1}(x) \cdot \mu_{A2}(x)}{1 - (1-\gamma) \mu_{A1}(x) \cdot \mu_{A2}(x)}$ $\gamma \geq 0$
Yager-перетин/ поєднання $T_{Yager_перетин}/S_{Yager_поєднання}$	$1 - \min\left\{1, \left((1 - \mu_{A1}(x))^\gamma + (1 - \mu_{A2}(x))^\gamma\right)^{1/\gamma}\right\}$ $\gamma \geq 0$	$\min\left\{1, \left(\mu_{A1}(x)^\gamma + \mu_{A2}(x)^\gamma\right)^{1/\gamma}\right\}$ $\gamma \geq 0$

Ще більш узагальненим, ніж *фазу-I*- та *фазу-АБО*-оператори є γ -оператор (також в літературі зустрічається позначення $Z_{комп}$), який дозволяє здійснити певну компенсацію однієї величини іншою. Цей оператор становить комбінацію алгебраїчного добутку й

алгебраїчної суми. Можна сказати, що γ -оператор може бути безперервно визначений між перетином та поєднанням фаззи-множин:

$$\forall x \in X: Z_{\text{комп}}(\mu_{A_1}(x), \mu_{A_2}(x), \gamma) = (\mu_{A_1}(x) \cdot \mu_{A_2}(x))^{1-\gamma} \cdot (\mu_{A_1}(x) + \mu_{A_2}(x) - \mu_{A_1}(x) \cdot \mu_{A_2}(x))^{\gamma} \quad (1.6)$$

Змінюючи γ в інтервалі $[0,1]$ можна реалізувати моделювання від логічного **I** до логічного **АБО** за допомогою лише одного оператора:

$$\gamma = 0: \quad Z_{\text{комп}} \rightarrow T_{\text{алгебр}}$$

$$\gamma = 1: \quad Z_{\text{комп}} \rightarrow S_{\text{алгебр}}$$

До другої групи операторів, що визначають відношення між фаззи-множинами, які містять до себе різні базові елементи, належать фаззи-співвідношення. Щоб пояснити це поняття, необхідно звернутись до визначення векторного добутку.

Векторний добуток $A_1 \times \dots \times A_m$ фаззи-множин A_1, \dots, A_m із базовими множинами X_1, \dots, X_m визначається в просторі добутку $X_1 \times \dots \times X_m$ таким чином:

$$\forall x_i \in X_i: \mu_{A_1 \times \dots \times A_m}(x_1, \dots, x_m) = \min_{i=1}^m \left\{ \mu_{A_i}(x_i) \right\} = \min \left\{ \mu_{A_1}(x_1), \dots, \mu_{A_m}(x_m) \right\} \quad (1.7)$$

На основі поняття векторного добутку можна визначити фаззи-співвідношення. Фаззи-співвідношення можуть розглядатися як узагальнення співвідношень класичних множин. Фаззи-співвідношення являє собою фаззи-множину, яка є підмножиною простору добутку $X_1 \times \dots \times X_m$ базових множин X_1, \dots, X_m . Фаззи-співвідношення (нечітке співвідношення) R у просторі добутку $X_1 \times \dots \times X_m$ із базовими множинами X_1, \dots, X_m визначається таким чином:

$$R = R(X_1, \dots, X_m) = \left\{ (x_1, \dots, x_m), \mu_R(x_1, \dots, x_m) \mid (x_1, \dots, x_m) \subseteq X_1 \times \dots \times X_m \right\} \quad (1.8)$$

Зворотнє фаззи-співвідношення R^{-1} бінарного фаззи-співвідношення (1.8) визначається як

$$R^{-1} = \left\{ (x_1, x_2), \mu_R(x_2, \dots, x_1) \mid (x_1, x_2) \subseteq X_1 \times \dots \times X_2 \right\} \quad (1.9)$$

Нечіткі бінарні співвідношення $R_1(X_1, X_2)$ та $R_2(X_2, X_3)$ з різними просторами добутку та спільним базовим елементом X_2 поєднуються за допомогою оператора композиції (оператора з'єднання) " \circ ". З'єднання (зчеплення) $R_{12}(X_1, X_3)$ двох бінарних фаззи-співвідношень $R_1(X_1, X_2)$ та $R_2(X_2, X_3)$ із базовими множинами X_1, X_2, X_3

$$R_{12}(X_1, X_3) = R_1(X_1, X_2) \circ R_2(X_2, X_3) \quad \text{або} \quad R_{12} = R_1 \circ R_2 \quad (1.10)$$

може бути визначене за допомогою деяких операторів, найбільш відомі з них наведені у табл. 1.4.

Таблиця 1.4 – Оператори з'єднання

$R_{12} = R_1 \circ R_2$	$\mu_{R_{12}}(x_1, x_3)$
max-min (максимум мінімуму)	$\max_{x_2 \in X_2} \left\{ \min \left\{ \mu_{R_1}(x_1, x_2), \mu_{R_2}(x_2, x_3) \right\} \right\}$
max-prod (максимум добутку)	$\max_{x_2 \in X_2} \left\{ \mu_{R_1}(x_1, x_2) \cdot \mu_{R_2}(x_2, x_3) \right\}$
max-average (максимум середнього значення)	$\frac{1}{2} \cdot \max_{x_2 \in X_2} \left\{ \mu_{R_1}(x_1, x_2) + \mu_{R_2}(x_2, x_3) \right\}$

1.2 Фази-контролер

У попередньому розділі були представлені основні поняття та механізми фази-теорії без огляду на практичні аспекти її використання в техніці регулювання. В цьому розділі розглядаються засоби побудови на основі нечіткої логіки фази-контролерів та їх практичне застосування в межах техніки автоматизації.

Основну структуру фази-контролера можна представити у вигляді трьох блоків (рис. 1.2):

- ✓ вхідний фільтр;
- ✓ фази-блок;
- ✓ вихідний фільтр.

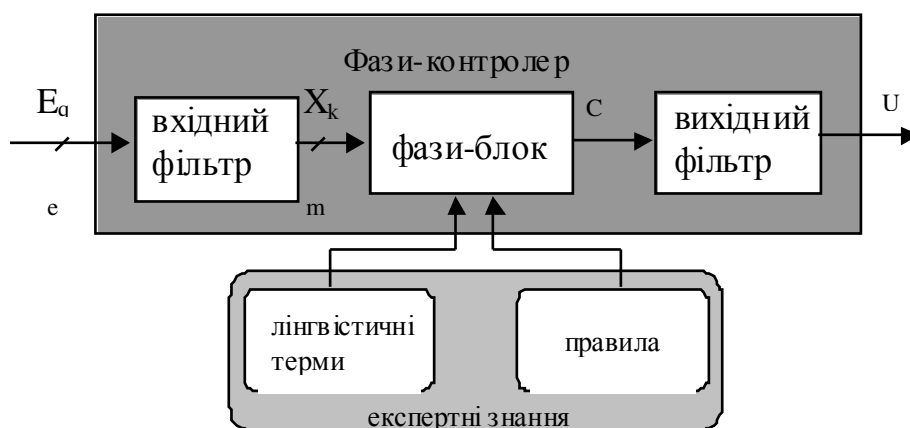


Рисунок 1.2 – Структура фази-контролера

Вхідні та вихідні фільтри фази-контролера служать для узгодження параметрів об'єкта регулювання із параметрами фази-блока. Вхідні та вихідні фільтри можуть включати в себе

динамічні компоненти фази-контролера (наприклад, інтегратори та ланки диференціювання) або складатися із чисто статичних компонентів (наприклад, суматори та підсилювачі). Фази-блок утворює центральну частину фази-контролера. Передавальна функція фази-блока є чисто статичною, фази-блок не містить у собі жодних динамічних компонентів та внутрішніх змінних стану.

Завданням вхідного фільтра є перетворення вимірюваних величин системи $E_1 \dots E_e$ на вхідні величини $X_1 \dots X_m$ фази-блока. У випадку, якщо для утворення значення однієї вхідної величини потрібні значення декількох змінних системи, кількість e величин системи $E_1 \dots E_e$ може відрізнятись від кількості m вхідних величин $X_1 \dots X_m$ фази-блока. Найпростішим прикладом цього є використання помилки регулювання як вхідної величини фази-блока. Помилка регулювання X_d обчислюється виходячи із заданого значення величини r , що регулюється, та поточного значення цієї ж величини y ($X_d = r - y$).

Завданням вихідного фільтра є узгодження значення вихідної величини фази-блока із керуючою величиною перетворювача або безпосередньо об'єкта регулювання. Принципово вихідний фільтр може містити будь-які статичні та динамічні функції, проте як правило він містить тільки підсилювач або інтегратор. Інтегратор у вихідному фільтрі здійснює перетворення форми керуючого сигналу.

Ядро фази-контролера утворює фази-блок. Вище було зазначено, що фази-блок не містить жодних динамічних компонентів. Це означає, що будь-який фази-блок можна представити за допомогою статичної функції $F: \underline{X} \rightarrow \underline{C}$, що пов'язує базові множини X_1, \dots, X_m вхідних величин із базовою множиною вихідної величини \underline{C} . Цей зв'язок здійснюється за допомогою якісних знань експерта. Це означає, що функція $c = F(\underline{x})$, де $\underline{x} \in \underline{X}$, $c \in \underline{C}$, не записується математично в аналітичній формі, а встановлюється за допомогою лінгвістичних термів (фази-множин) та правил на основі інших ступенів свободи. Функція $F(\underline{x})$ є як правило нелінійною. Це викликає питання: що власне нового привносить фази-регулювання? Адже нелінійна передавальна функція із багатьма вхідними змінними може бути реалізована за допомогою багатовимірного регулятора. Новим є метод розробки системи регулювання, вербальне формулювання експертних знань за допомогою лінгвістичних правил.

Структура фази-блока показана на рис. 1.3. Фази-блок складається з трьох частин:

- ✓ блок фазифікації;
- ✓ блок інференції;
- ✓ блок дефазифікації.

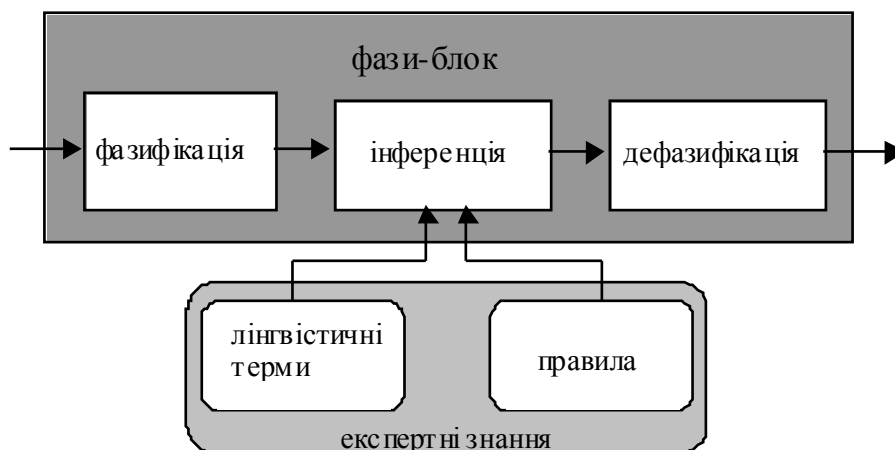


Рисунок 1.3 – Структура фазі-блока

У техніці регулювання можуть бути виміряні тільки однозначні величини, і тільки однозначні керуючі величини можуть бути задані об'єкту регулювання; тому використання фазі-множин безпосередньо як технічного сигналу неможливе. Фазифікація та дефазифікація здійснюють перетворення відповідно однозначних вхідних величин на вхідні фазі-множини та вихідних фазі-множин на однозначні вихідні величини.

У процесі фазифікації кожному елементарному висловленню *ЯКЩО-частини* правила ставиться до відповідності ступінь приналежності. Ступінь приналежності розраховується із функції приналежності лінгвістичного терма та актуального вхідного значення елементарного висловлення. Графічне зображення фазифікації надається на рис. 1.4.

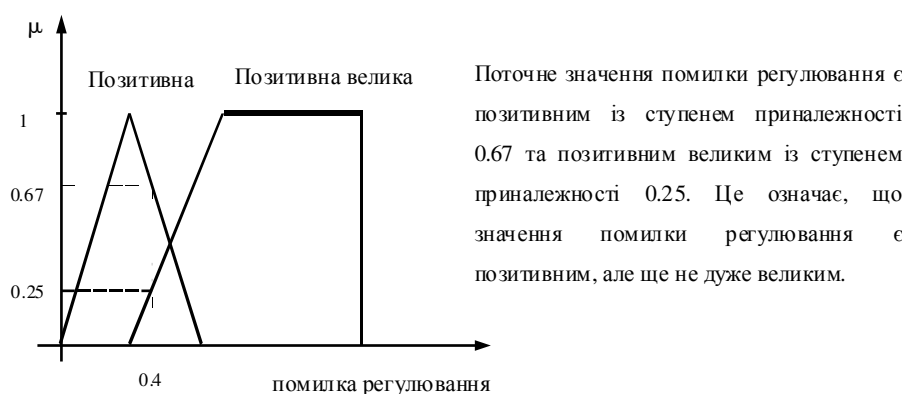


Рисунок 1.4 – Фазифікація

Завданням інференції є перетворення якісного експертного знання на закон регулювання, що може бути реалізований технічно. Вхідними даними для процесу інференції є фазі-правила та ступінь приналежності, з яким є справедливим елементарне

висловлення $E_{k,i}(x_k)$. На основі результатів фазифікації та фази-правил в процесі інференції відбувається формування вихідної фази-множини B^* . Інференцію можна умовно поділити на три етапи:

- ✓ агрегація;
- ✓ імплікація;
- ✓ акумуляція.

У процесі агрегації вибираються різні елементарні висловлення, що належать до одного правила, та утворюється ступінь приналежності, з яким є справедливим елементарне висловлення *ЯКЩО-частини* фази-правила. Кожна *ЯКЩО-частина* правила складається з декількох елементарних висловлень, які зводяться за допомогою логічних зв'язків до єдиного комплексного висловлення. Як результат для кожного активного правила визначається ступінь приналежності, з яким висловлення у *ЯКЩО-частині* є справедливим. Операторами агрегації можуть бути будь-які з наведених вище операторів, але в техніці регулювання використовується як правило лише І-оператор. І-оператор може бути реалізований двома способами, тому можна виділити два типи реалізації оператора агрегації:

- ✓ *мінімум-агрегація*:

$$E_i(\underline{x}) = \min\{E_{1,i}(x_1), \dots, E_{m,i}(x_m)\} = \min_{k=1}^m E_{k,i}(x_k) \quad \text{при } i \in \{1, \dots, n\} \quad (1.11)$$

- ✓ *добуток-агрегація*:

$$E_i(\underline{x}) = E_{1,i}(x_1) \cdot \dots \cdot E_{m,i}(x_m) = \prod_{k=1}^m E_{k,i}(x_k) \quad (1.12)$$

Графічне зображення мінімум-агрегації надається на рис. 1.5. Розглядається правило, *ЯКЩО-частина* якого має вигляд:

ЯКЩО Помилка регулювання позитивна **ТА** Похідна від помилки регулювання позитивна велика,...

Помилка регулювання є позитивною зі ступенем приналежності 0.7 та *Похідна від помилки регулювання* є позитивною великою зі ступенем приналежності 0.6. Відповідно до формули (1.11), агрегація дає результат – ступінь приналежності, з яким виконується *ЯКЩО-частина* правила, – 0.6.

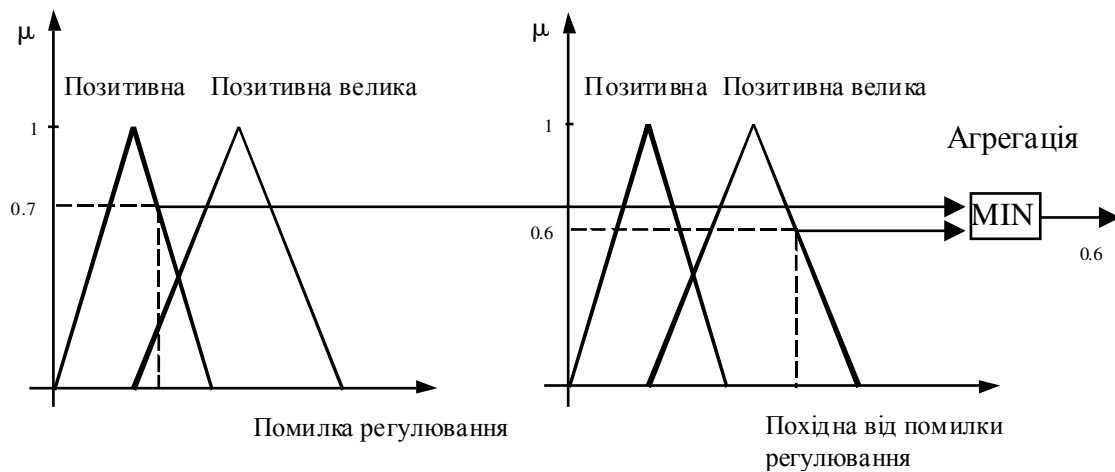


Рисунок 1.5 – Мінімум-агрегація

В процесі імплікації для кожного активного правила залежно від ступеня приналежності $E_i(\underline{x})$, з яким воно виконується, визначається фази-множина \mathbf{B}_i^* . Для здійснення імплікації використовуються оператори композиції. Фази-множина \mathbf{B}_i^* є складовою частиною базової множини \mathbf{C}_i вихідної величини \mathbf{C} фази-блока. Існує декілька методів імплікації, найбільш відомі з них наведені в табл. 1.5. Найчастіше як оператор імплікації в техніці регулювання використовується І-оператор. Графічне уявлення імплікації, що здійснюється за допомогою І-оператора, представлено на рис. 1.6.

Таблиця 1.5 – Методи імплікації

Оператор імплікації	$\forall c \in C: \mu_{\mathbf{B}_i^*}(c) =$
Zadeh	$\max \left\{ \min \left\{ E_i(\underline{x}), \mu_{\mathbf{B}_i}(c) \right\}, 1 - E_i(\underline{x}) \right\}$
Lukasiewicz	$\min \left\{ 1, 1 - E_i(\underline{x}) + \mu_{\mathbf{B}_i}(c) \right\}$
Minimum Mamdani	$\min \left\{ E_i(\underline{x}), \mu_{\mathbf{B}_i}(c) \right\}$
Kleene-Dienes	$\max \left\{ 1 - E_i(\underline{x}), \mu_{\mathbf{B}_i}(c) \right\}$
Produkt	$E_i(\underline{x}) \cdot \mu_{\mathbf{B}_i}(c)$

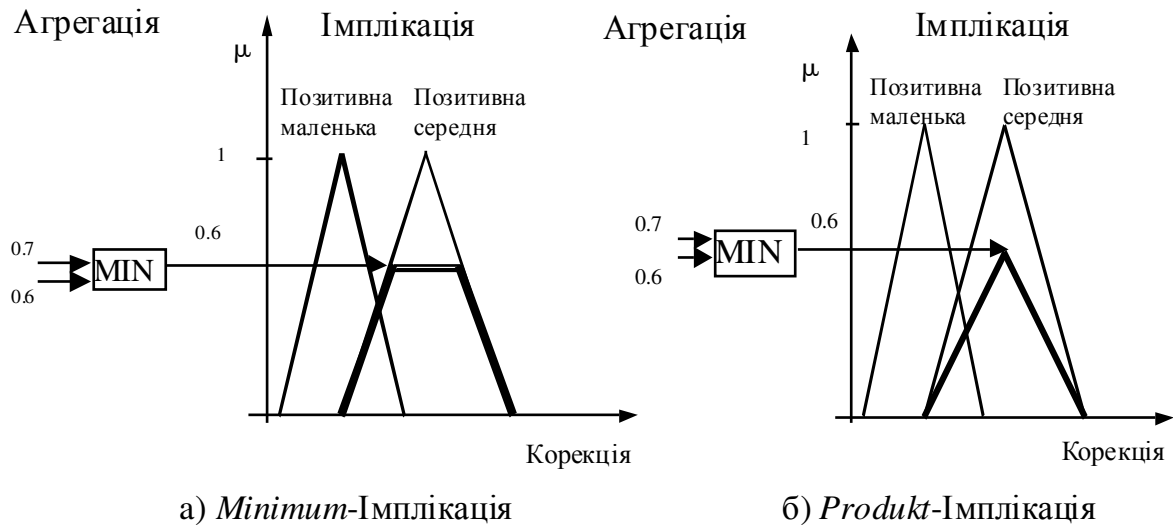


Рисунок 1.6 – Імплікація

Останнім етапом інференції є акумуляція. Завданням акумуляції є об'єднання результатів інференції окремих фаз-правил. Результатом акумуляції є фазі-множина \mathbf{V}^* , що утворюється шляхом об'єднання окремих фазі-множин \mathbf{V}_i^* :

$$\mathbf{V}^* = \bigcup_{i=1}^n \mathbf{V}_i^* \quad (1.13)$$

Акумульована фазі-множина \mathbf{V}^* являє собою остаточний результат процесу інференції. Функція приналежності результуючої фазі-множини утворюється за допомогою оператора акумуляції *Akk*:

$$\forall c \in C: \mu_{\mathbf{V}^*}(c) = \text{Akk} \mu_{\mathbf{V}_i^*}(c) \quad (1.14)$$

Оператор акумуляції *Akk* здійснює об'єднання фазі-множин, тому для його реалізації можна застосовувати будь-який з розглянутих вище операторів для фазі-множин. Оператори, що є найбільш поширеними, наведені в табл. 1.6.

Таблиця 1.6 – Оператори акумуляції

Оператори акумуляції <i>Akk</i>	T-Со-норма bzw. S-норма
Максимум (за Мамдані)	S_{\max}
Обмежена сума	$S_{\text{обмеж}}$

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

Найчастіше як оператор акумуляції виступає максимум-оператор. У цьому випадку акумульована фази-множина визначається таким чином:

$$\forall c \in C \quad \mu_{B^*}(c) = \max \left\{ \mu_{B_1^*}(c), \dots, \mu_{B_n^*}(c) \right\} = \max_{i=1}^n \mu_{B_i^*}(c) \quad (1.15)$$

Функція приналежності результуючої фази-множини, що визначається за допомогою максимум-оператора, наведена на рис. 1.7.

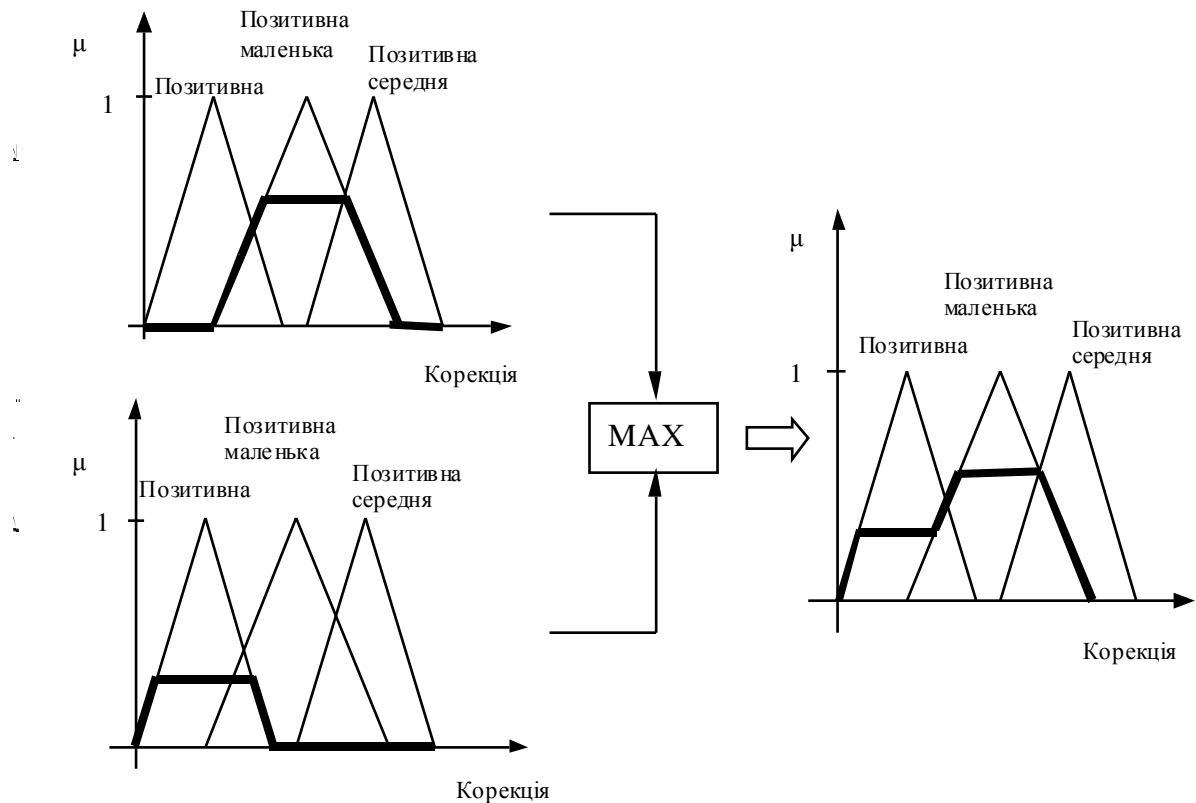


Рисунок 1.7 – Акумуляція

У техніці регулювання як керуючі сигнали можуть бути використані лише однозначні величини, тому акумульована у процесі інференції фази-множина не може бути використана як керуюча величина об'єкту регулювання; необхідна дефазифікація отриманого результату. У процесі дефазифікації відбувається перетворення акумульованої фази-множини на однозначну вихідну величину. Існує велика кількість методів дефазифікації, метою яких є визначення такої однозначної вихідної величини, яка б як можна точніше представила акумульовану фази-множину. Серед найбільш поширених слід зазначити такі методи дефазифікації:

- ✓ **метод мінімуму**, коли як вихідне значення виступає максимальне значення множини S^{\min} , яку утворюють елементи, що належать до терму із мінімальними значеннями функції приналежності акумульованої фази-множини (рис. 1.8).
- ✓ **метод максимуму**, коли як вихідне значення виступає максимальне значення множини S^{\max} , яку утворюють елементи, що належать до терму із максимальними значеннями функції приналежності акумульованої фази-множини (рис. 1.8).
- ✓ **гравітаційний метод**, коли розраховується центр ваги площі функції приналежності вихідної величини, яка складається зі всіх лінгвістичних змінних (рис. 1.8). Вихідна величина визначається таким чином:

$$X_S = \frac{\int c \cdot \mu(c) dc}{\int \mu(c) dc} \quad (1.16)$$

- ✓ **сінглетон - метод** – спрощений варіант гравітаційного метода. Застосування цього метода дозволяє зменшити витрати часу на розрахунки, тобто збільшити швидкість обчислювальних процесів та зменшити витрати пам'яті. Відхилення результатів, отриманих за допомогою цього метода, від результатів розрахунків за гравітаційним методом незначні.

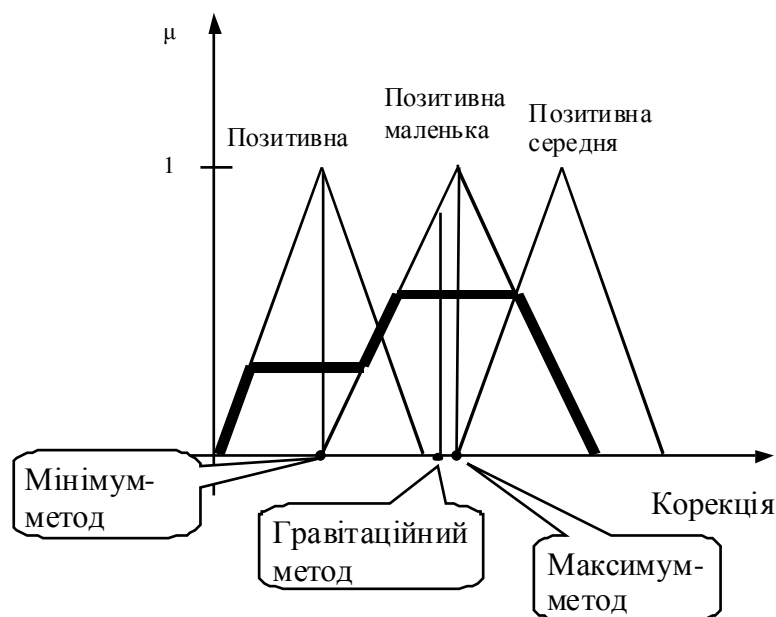


Рисунок 1.8 - Методи дефазифікації

Розглянемо докладніше найпростіший і тому найбільш поширений метод дефазифікації – сінглетон-метод. Цей метод базується на гравітаційному методі і тому передбачає розрахунок проєкції центра ваги фігури, яка утворюється шляхом накладення

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

усіх усічених площ, що відповідають результату кожного активного правила. Звичайно, це стосується випадку наявності декількох активних термів вихідної змінної.

Як було зазначено вище, центр ваги X_S результуючої площі визначається шляхом інтегрування результативної функції $\mu(c)$ (рис. 1.9).

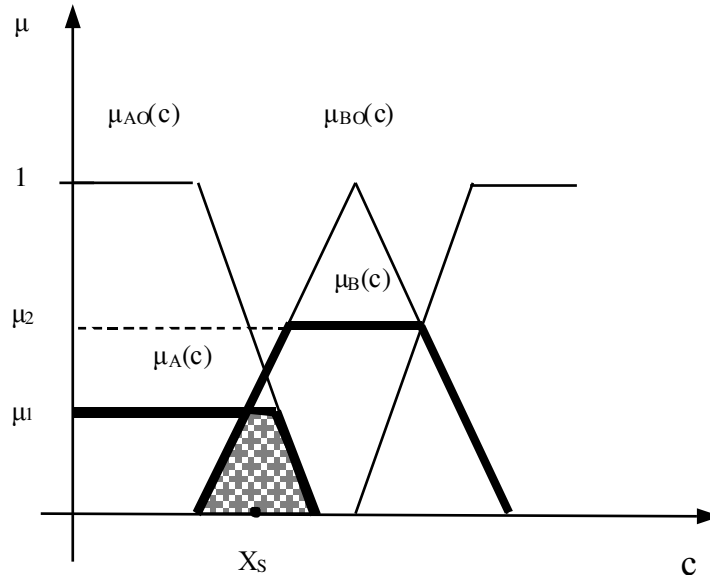


Рисунок 1.9 - Визначення центра ваги результуючої площі

Якщо усічені функції приналежності розглядати окремо, а значення усічених функцій визначати шляхом множення

$$\mu_A(c) = \mu_1 \cdot \mu_{AO}(c) \quad (1.17)$$

$$\mu_B(c) = \mu_2 \cdot \mu_{BO}(c) \quad (1.18),$$

то методика значно спрощується:

$$X_S = \frac{\mu_1 \cdot \int c \cdot \mu_{AO}(c) dc + \mu_2 \cdot \int c \cdot \mu_{BO}(c) dc}{\mu_1 \cdot \int \mu_{AO}(c) dc + \mu_2 \cdot \int \mu_{BO}(c) dc} \quad (1.19)$$

Якщо розглянути останнє рівняння, то можна помітити, що тепер під інтегралом стоять тільки функції $\mu_{AO}(c)$ та $\mu_{BO}(c)$, а ці функції не залежать від процесу, тобто площа під ними може бути розрахована ще під час проектування.

Інтеграл в чисельнику дробі відповідає обертальному моменту M , а інтеграл в знаменнику відповідає усіченій площі A . Відповідно до цього рівняння спрощується ще:

$$X_S = \frac{\mu_1 \cdot M_{AO} + \mu_2 \cdot M_{BO} + \dots}{\mu_1 \cdot A_{AO} + \mu_2 \cdot A_{BO} + \dots} \quad (1.20)$$

З цього рівняння видно, що форма функцій приналежності вихідної фази-змінної не впливає на кінцевий результат. Зараз важливими є тільки площі та їхнє розташування на осі. Якщо вибрати функції приналежності $\mu_{AO}(c)$ і $\mu_{BO}(c)$ незалежними від c , наприклад, прямокутники з висотою, яка дорівнює одиниці, то ці функції можна винести за знак інтеграла і скоротити. У цьому випадку ми отримаємо формулу розрахунку центра ваги за сінглетон - методом:

$$X_S = \frac{\mu_1 \cdot c_A + \mu_2 \cdot c_B + \dots}{\mu_1 + \mu_2 + \dots} \quad (1.21)$$

На рис. 1.10 показані сінглетон-функції усічених функцій приналежності $\mu_{AO}(c)$ і $\mu_{BO}(c)$.

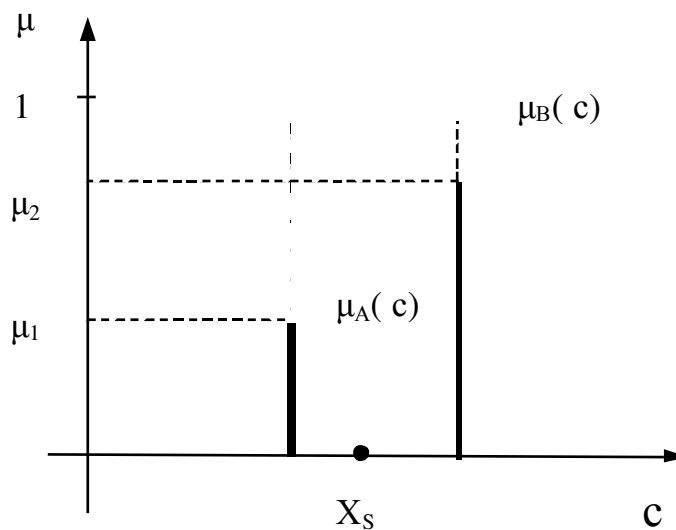


Рисунок 1.10 - Сінглетон-функції усічених функцій приналежності

Усі ступені свободи фази-блока можна розподілити на два класи:

1. Параметричні:

- ✓ лінгвістичні терми (форма та параметри функцій приналежності);
- ✓ фази-правила;
- ✓ вагові коефіцієнти.

2. Алгоритмічні:

- ✓ оператори агрегації;
- ✓ оператори імплікації;

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

- ✓ оператори акумуляції;
- ✓ метод дефазифікації.

Визначальну роль при настроюванні фази-блока грають параметричні ступені свободи, тому що саме база правил та лінгвістичні терми представляють експертне знання. При розробці фази-блока оператори інференції та метод дефазифікації завжди розглядаються підпорядковано, головна увага приділяється визначенню параметричних ступенів свободи. Значно легше помітити зміни в роботі системи, що викликані зміною лінгвістичних термів або правил, ніж зміни, що спричиняються варіацією методу дефазифікації або операторів інференції. Алгоритмічні ступені свободи здебільшого встановлюються виходячи із вимог до швидкодії (це стосується насамперед до вибору методу дефазифікації) та сталості системи. В техніці регулювання існують установки, яким традиційно надається перевага при проектуванні фази-блоків. Як оператори інференції, наприклад, найчастіше вживаються *Max-Min*- та *Max-Prod*-інференція. Огляд найбільш поширених операторів інференції представлений в табл. 1.7.

Таблиця 1.7 – Оператори інференції

Метод інференції	Агрегація (I)	Імплікація	Акумуляція
<i>Max-Min</i>	Мінімум або добуток	Мінімум	Максимум
<i>Max-Prod</i>	Мінімум або добуток	Добуток	Максимум
<i>Sum-Prod</i>	Мінімум або добуток	Добуток	Сума
<i>Sum-Min</i>	Мінімум або добуток	Мінімум	Сума

Математичний опис операторів інференції у вигляді формул надається в табл. 1.8.

Таблиця 1.8 – Формульне представлення деяких операторів інференції

Метод інференції	Формула
<i>Max-Min</i>	$\forall c \in C: \mu_{B^*}(c) = \max_{i=1}^n \left\{ \min \left[\mu_{B_i}(c), E_i(\underline{x}) \right] \right\}$
<i>Max-Prod</i>	$\forall c \in C: \mu_{B^*}(c) = \max_{i=1}^n \left\{ \mu_{B_i}(c) \cdot E_i(\underline{x}) \right\}$

Продовження табл. 1.8

<i>Sum-Prod</i>	$\forall c \in C: \mu_{B^*}(c) = \sum_{i=1}^n \left\{ \mu_{B_i}(c) \cdot E_i(x) \right\}$
<i>Sum-Min</i>	$\forall c \in C: \mu_{B^*}(c) = \sum_{i=1}^n \left\{ \min \left[\mu_{B_i}(c), E_i(x) \right] \right\}$

Загальна структура фаззи-контролера зображена на рис. 1.11. Ця схема може служити допоміжним матеріалом при проектуванні фаззи-контролерів.

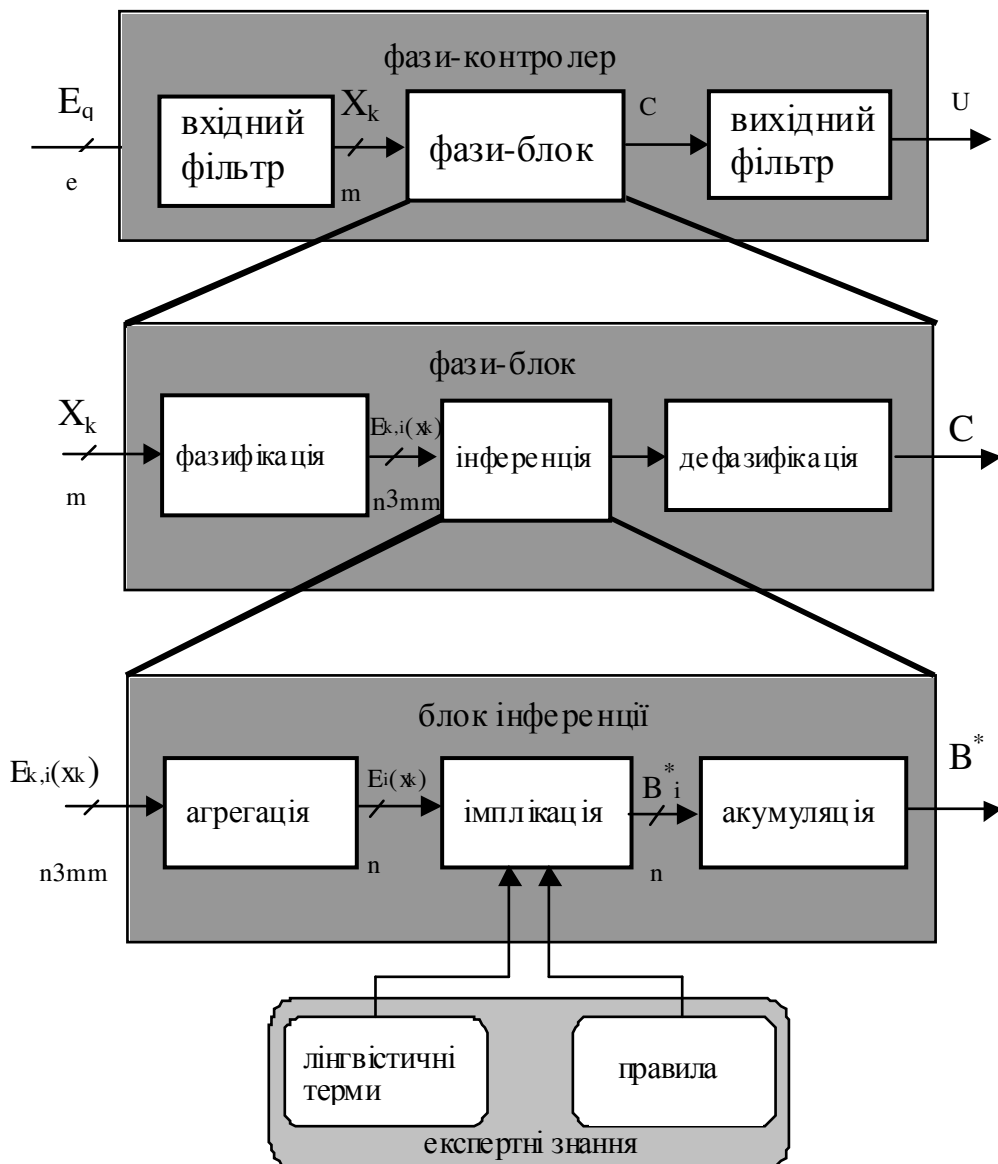


Рисунок 1.11 – Загальна структура фаззи-контролера

1.3 Основні етапи проектування фази-контролерів

Процес проектування фази-контролера складається з чотирьох етапів.

1. *Аналіз системи.* На початку проектування обов'язково треба провести аналіз системи. Процедура аналізу включає в себе як вибір структури контуру регулювання, так і вибір вимірюваних величин та вхідних величин фази-контролера, що визначаються виходячи із результатів вимірювання; визначення вихідних величин фази-контролера, тобто керуючих величин об'єкта регулювання. На цьому етапі можуть також бути встановленими вхідні й вихідні фільтри, що вимагає вірного добору коефіцієнтів підсилювання або масштабних коефіцієнтів у вхідних та вихідних фільтрах. Ця процедура є обов'язковою при застосуванні нормованого фази-блока, тому що в такому випадку необхідно узгодити величини системи із вхідними та вихідними величинами фази-блоку, що змінюються в одиничному інтервалі (наприклад, $[-1,1]$ або $[1,100]$). Використання нормованих фази-блоків дозволяє інколи стандартизувати їх, використовувати той самий фази-блок в різних системах керування із різними параметрами, змінюючи лише масштабні коефіцієнти вхідних та вихідних фільтрів. Масштабні коефіцієнти вхідних та вихідних фільтрів грають таку ж важливу роль, як і коефіцієнти підсилення в класичних регуляторах. Перший етап проектування є дуже відповідальним і потребує глибоких знань в області класичних методів регулювання та цифрової обробки сигналу. Помилки, що виникають на цьому етапі, набагато складніше виправити, ніж ті, що виникають на наступних етапах.
2. *Встановлення алгоритмічних ступенів свободи.* На цьому етапі здійснюється визначення основних властивостей фази-блока. Наприклад, метод дефазифікації визначає швидкодію фази-блока та його сталість.
3. *Встановлення параметричних ступенів свободи.* Цей етап є центральним в усьому процесі проектування, його можна розподілити на три кроки:
 - ✓ встановлення можливих інтервалів зміни вхідних та вихідних величин. При цьому припускається, що на попередньому етапі були визначені властивості об'єкта регулювання та масштабні коефіцієнти фільтрів;
 - ✓ встановлення лінгвістичних термів та їх функцій приналежності для всіх лінгвістичних змінних. Як правило у техніці регулювання обираються трикутні, трапецієподібні та синглетон-формені функції приналежності. Слід

зауважити, що наявність невизначених областей є небажаною, всі функції приналежності вхідних фази-множин обов'язково мають перекриватись;

- ✓ складання бази правил. На жаль, для цього не існує жодного систематизованого алгоритму, тому центральний етап проектування фази-регулятора не має такої загальноприйнятої систематики, як при проектуванні класичних регуляторів. Це значно ускладнює процес проектування - дуже часто трапляються випадки, коли різні настройки фази-блока зумовлюють майже ідентичні його передавальні властивості.
4. *Моделювання спроектованого контуру регулювання.* На цьому етапі відбувається перевірка розробленого фази-контролера. У випадку, якщо результати моделювання є незадовільними, потрібно змінити певні параметри фази-контролера.

На рис. 1.12 процес розробки фази-контролера представляється у наочній формі.

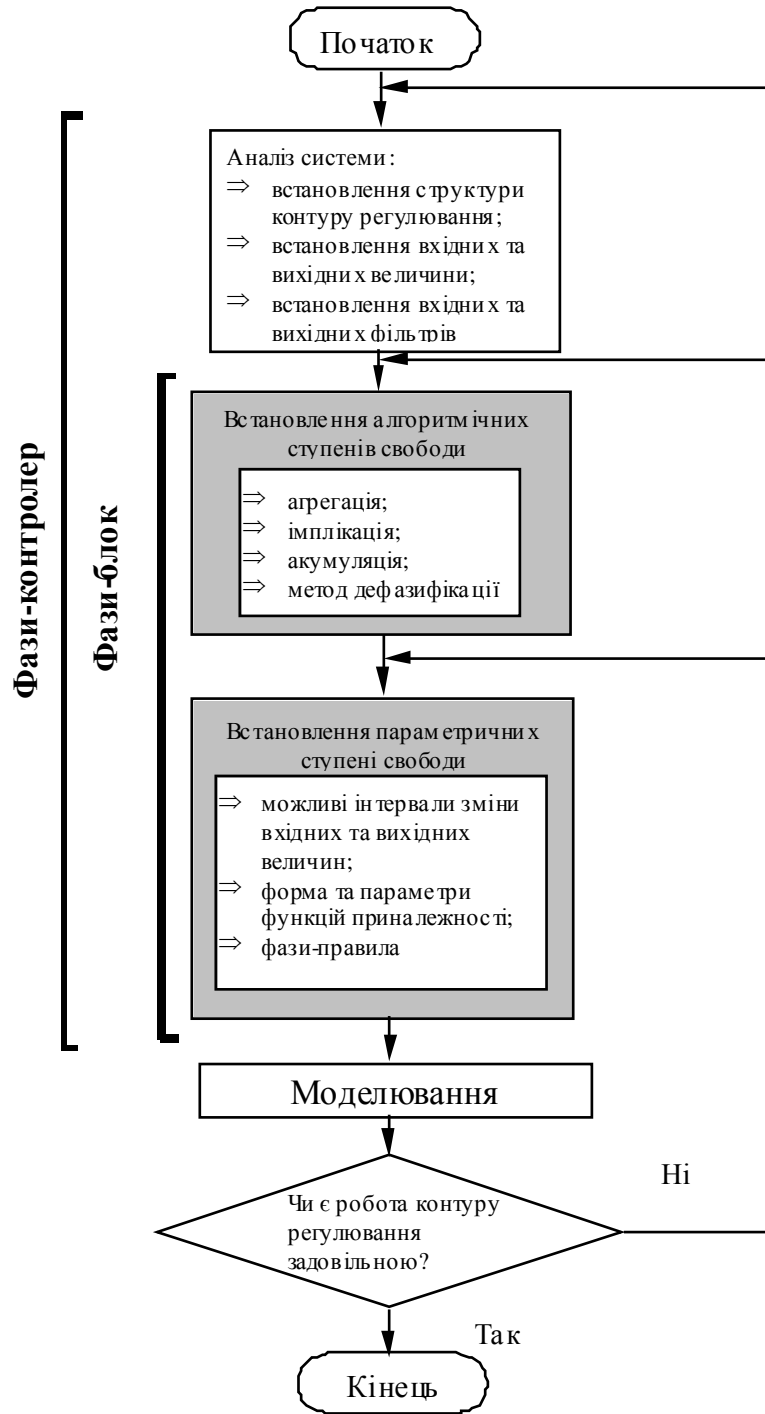


Рисунок 1.12 – Процес розробки фази-контролера

1.4 Типи фази-контролерів

Фази-контролери, як і класичні регулятори, можуть бути реалізовані у вигляді П-регуляторів, ПІ-регуляторів, ПД-регуляторів та ПІД-регуляторів [4]. Проте слід пам'ятати,

що ця аналогія не є повною, і фазі-П-регулятор, наприклад, не є аналогом класичного П-регулятора. Нижче розглянуті основні типи фазі-контролерів.

1. Фазі-П-регулятор

Фазі-П-регулятором називається фазі-контролер з однією вхідною величиною, причому при зростанні вхідної величини e вихідна величина u також зростає або принаймні залишається незмінною, тобто:

$$u = F(e)$$

Це визначення збігається з вимогами до монотонної (проте не строго монотонної) передавальної функції та являє собою значне пом'якшення поняття пропорційності. Як фазі-П-регулятор може розглядатися, наприклад, зона нечутливості.

2. Фазі-ПІ-регулятор

Фазі-ПІ-регулятором називається фазі-контролер, вхідними величинами якого є помилка регулювання e та швидкість її змінювання (тобто перша похідна) \dot{e} , а вихідною величиною є похідна від керуючого сигналу \dot{u} , причому як при зростанні помилки регулювання e , так і при зростанні її похідної \dot{e} вихідна величина \dot{u} зростає або принаймні залишається незмінною:

$$\dot{u} = F(e, \dot{e})$$

3. Фазі-ПІД-регулятор

Фазі-ПІД-регулятором називається фазі-контролер, вхідними величинами якого є помилка регулювання e та швидкість її зміни (тобто перша похідна) \dot{e} , а вихідною величиною є керуючий сигнал u , причому як при зростанні помилки регулювання e , так і при зростанні її похідної \dot{e} вихідна величина u зростає або принаймні залишається незмінною:

$$u = F(e, \dot{e})$$

4. Фазі-ПІІД-регулятор

Фазі-ПІІД-регулятор може бути реалізований за допомогою двох алгоритмів – алгоритму встановлення керуючої величини та алгоритму встановлення швидкості зміни керуючої величини:

- ✓ при реалізації за першим алгоритмом фазі-ПІІД-регулятором називається фазі-контролер, вхідними величинами якого є помилка регулювання e , її перша похідна \dot{e} та інтеграл від помилки регулювання $\int e dt$, а вихідною величиною є керуючий сигнал u :

$$u = F(e, \int e dt, \dot{e})$$

- ✓ при реалізації за другим алгоритмом фази-ПІД-регулятором називається фази-контролер, вхідними величинами якого є помилка регулювання e , її перша \dot{e} та друга \ddot{e} похідні, а вихідною величиною є похідна керуючого сигналу \dot{u} .

$$\dot{u} = F(e, \dot{e}, \ddot{e})$$

1.5 Варіанти побудови контурів регулювання із фази-регулятором

Існує велика кількість різноманітних варіантів побудови контурів регулювання із фази-регулятором. Цей розділ присвячений розгляду найбільш поширених з них.

Найпростіша структура показана на рис. 1.13. Ця структура представляє собою одноконтурну систему регулювання, аналогічну класичній одноконтурній системі. Замість класичного регулятора контур містить фази-регулятор, що має одну вхідну та одну вихідну величини.

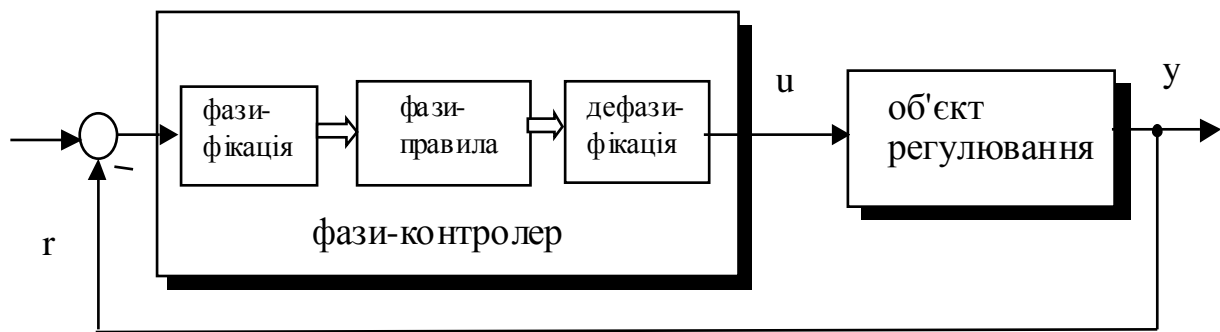


Рисунок 1.13 – Одноконтурна система регулювання з фази-регулятором

Трохи ускладнений варіант цієї структури є система, показана на рис. 1.14. Фази-контролер в цьому випадку має більш, ніж одну вхідну величину, тому в системі з'являється додатковий блок підготовки вимірюваних величин. Його завданням є розрахунок вхідних величин регулятора, виходячи із значення помилки регулювання та надання отриманих величин регулятору. Принципово при реалізації цього блока можливе використання будь-яких динамічних та статичних операторів.

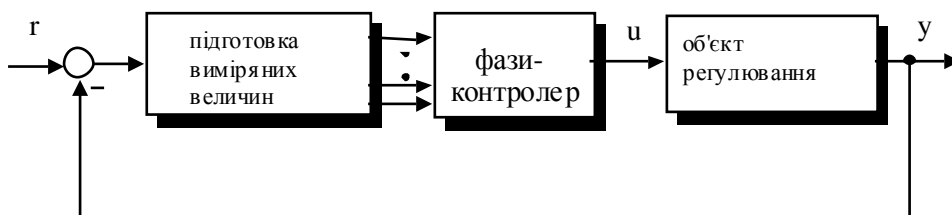


Рисунок 1.14 - Одноконтурна система регулювання з фази-регулятором

Слід зауважити, що структури, подібні до наведених, на практиці вживаються дуже обмежено. Причиною є те, що проєктувальники ще "не довіряють" фазі-контролеру, теорія фазі-регулювання в даний час ще не має однозначних простих методів оцінки фазі-систем регулювання. Тому досить часто структур із одним лише фазі-регулятором намагаються уникнути. В більшості випадків проєктувальники систем фазі-регулювання прагнуть поєднати достоїнства фазі-регуляторів та класичних регуляторів. Дуже часто використовуються структури, в яких до випробуваного класичного контуру регулювання (наприклад, до контуру з ПІД-регулятором) підключається фазі-регулятор, що має компенсувати недоліки класичного регулятора або втрутитися у разі особливих режимів роботи. Це дає можливість переключатись залежно від поточного режиму роботи системи між двома концепціями регулювання: при незначних відхиленнях від робочої точки є доцільним використання класичного регулятора (наприклад, ПІД-регулятора), при значних швидких змінах збудження або завдання хороші результати дає фазі-регулятор. Одним із варіантів реалізації такого підходу є структура, наведена на рис. 1.15. Контур регулювання містить фазі-контролер, що підключається паралельно до класичного регулятора. Вхідною величиною фазі-контролера може бути, наприклад, помилка регулювання та похідна від неї, на виході фазі-контролер утворює коригувальний сигнал, що складається із сигналом на виході класичного регулятора.

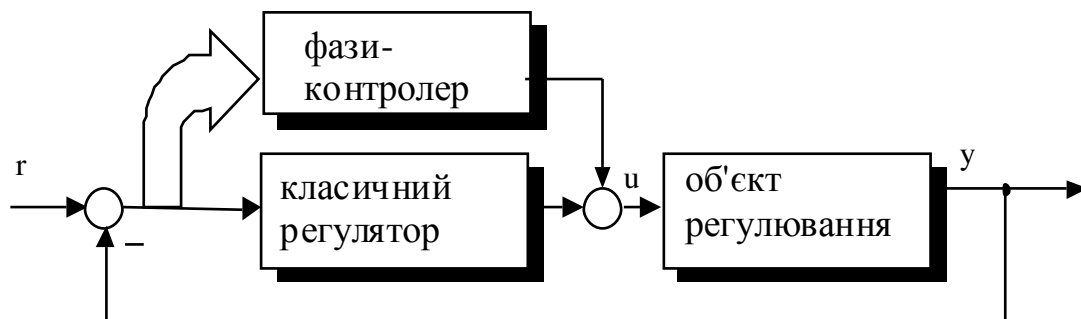


Рисунок 1.15 – Контур регулювання з паралельною фазі-корекцією

Широкі можливості для фазі-регулювання відкриваються в області адаптивного регулювання. Йдеться про *on-line*-покращення властивостей контуру регулювання, що є, наприклад, доцільним у системах, в яких вид перехідного процесу залежить від моменту його початку. В таких системах шляхом модифікації структури регулювання розробляється фазі-контролер, здатний до самоорганізації, який через зміну своїх параметрів залежно від стану системи функціонує можливо оптимально. Найпростіша структура регулювання такого типу надається на рис. 1.16.

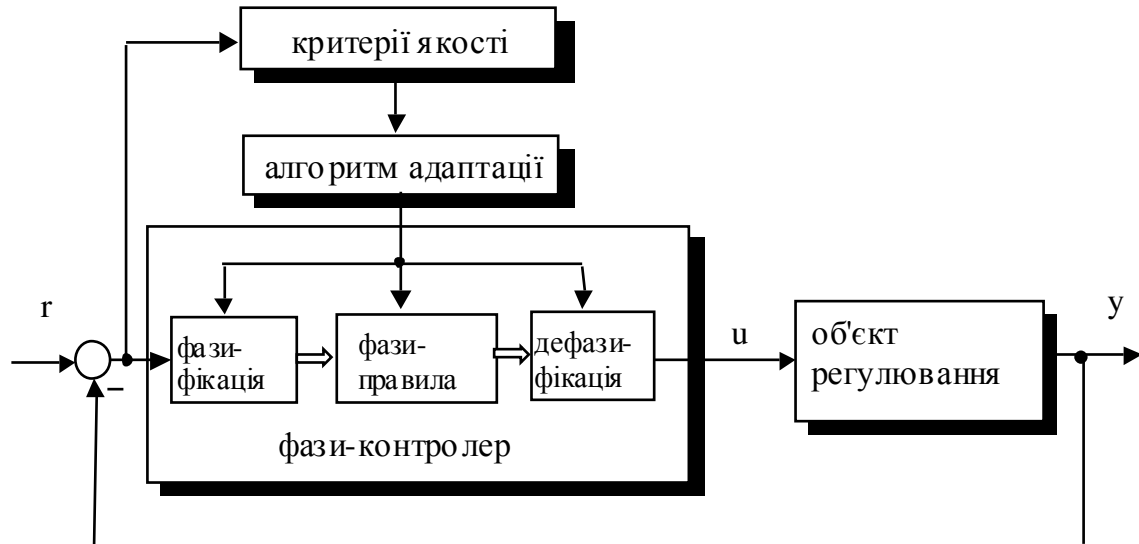


Рисунок 1.16 – Структура контуру регулювання з адаптивним фазі-контролером (пряма адаптація)

Основою для адаптації фазі-контролера служать в цьому випадку спостереження за величиною помилки регулювання, виходячи з якої визначається через певний критерій якості індекс якості, що характеризує якість регулювання. "Розум" адаптивного регулятора міститься у блоці *Алгоритм адаптації* у вигляді різноманітних команд, що виконуються залежно від значення індексу якості. Таким чином, алгоритм адаптації проектується як фазі-алгоритм у формі *ЯКЩО...ТО*-правил. Ця структура являє собою так звану пряму адаптацію. Більш складною є структура непрямої адаптації. Найпростіша форма такого контуру регулювання наведена на рис. 1.17. В цьому випадку структура алгоритму адаптації ускладнюється. Блок *Алгоритм адаптації* складається з двох блоків: *Ідентифікація системи* та *Проектувальник фазі-контролера*. Протягом процесу адаптації виконується аналіз поточних значень регульованої координати та заданого значення цієї ж координати. На основі даних аналізу відбувається зміна параметрів фазі-контролера. Здійснення зміни параметрів є завданням блока *Проектувальник фазі-контролера*.

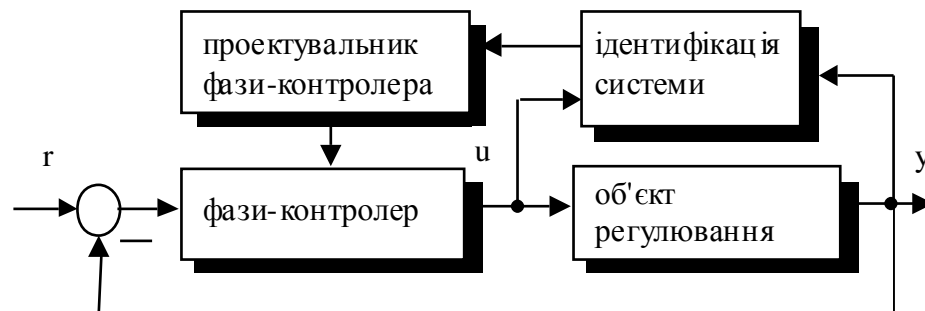


Рисунок 1.17 – Структура контуру регулювання з адаптивним фазі-контролером (непряма адаптація)

Така стратегія адаптації може бути реалізована в концепції регулювання, що ґрунтується на моделі процесу. Структура контуру регулювання такого типу представлена на рис. 1.18. В цьому випадку зміна параметрів фази-контролера відбувається на основі порівняння процесів, що протікають в реальній системі, з процесами в ідеалізованій моделі (так званої моделі процесу). Відхилення регульованої координати, що вимірюється в реальній системі, від значення цієї ж координати в моделі процесу (у разі необхідності може також аналізуватися похідна від нього) використовується як сигнал навчання. Це відхилення та його похідні (у більшості випадків до уваги приймається лише перша похідна) можуть бути проаналізовані за допомогою ще одного фази-контролера (не адаптивного), але це значно ускладнює структуру контуру регулювання.

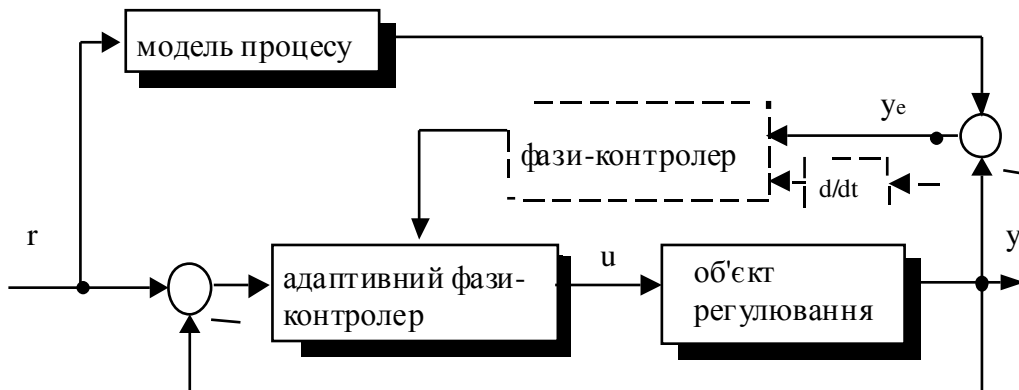


Рисунок 1.18 – Реалізація концепції регулювання, що ґрунтується на моделі процесу

Більш консервативна структура адаптивного контуру регулювання надається на рис. 1.19. В цьому випадку власне адаптивний регулятор є традиційним (наприклад, ПІД-регулятор), і зміна його параметрів відбувається за допомогою фази-алгоритму.

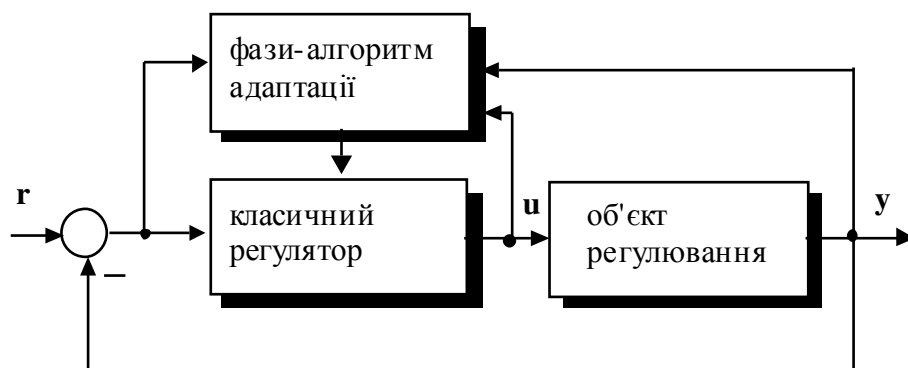


Рисунок 1.19 – Контур регулювання з адаптивним класичним регулятором

1 ОСНОВИ ТЕОРІЇ ФАЗИ-ЛОГІКИ ТА ФАЗИ-РЕГУЛЮВАННЯ

Вибір структури контуру регулювання є одним з найбільш відповідальних кроків у розробці системи регулювання. Складна система із адаптивним фази-контролером або адаптивним класичним регулятором, настроювання якого відбувається за допомогою фази-алгоритму, демонструє кращі результати, ніж одноконтурна система (рис. 1.13). Слід зауважити, що взагалі практичне застосування одноконтурної системи із фази-контролером є дуже обмеженим через проблеми у забезпеченні необхідного ступеня стабільності. Проте правильне настроювання складних систем потребує дуже великих витрат часу (насамперед через відсутність загальноживаних методів розробки фази-систем). Задовільні результати можуть бути отримані при вдалій настройці в системі із паралельною фази-корекцією (рис. 1.15). Настроювання такої системи не потребує значних витрат часу у порівнянні з настроюванням адаптивних систем, але така система має відносно високу надійність: при "випадінні" із контуру регулювання фази-контролера система продовжує задовільно функціонувати.

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

Як зазначалося вище, важливим етапом розробки системи із фази-регулятором є моделювання контуру регулювання. В даний час розроблювачі програмного забезпечення пропонують різні моделюючі програми, що дозволяють проводити розробку та тестування фази-блоків. Деякі з них дають можливість інтегрувати модель фази-регулятора до загальної моделі системи регулювання та дослідити роботу системи в цілому, інші дозволяють лише перевірити роботу фази-блока окремо від системи, до якої він інтегрується, при різних значеннях вхідних величин. Велика кількість проектувальників користується при моделюванні програмним пакетом Matlab та моделюючою програмою Simulink, що входить до його складу. Simulink дає можливість створювати наочні математичні моделі систем регулювання та досліджувати їх функціонування.

Для моделювання фази-блоків в програмному пакеті Matlab/Simulink було розроблено Fuzzy Logic Toolbox, що являє собою набір функцій, створених у числовому обчислювальному середовищі Matlab. Fuzzy Logic Toolbox містить в собі засоби створення та редагування фази-блоків в межах структури Matlab. Fuzzy Logic Toolbox дає можливість створювати фази-блоки трьох типів [6]:

- ✓ система інференції Мамдані (*Mamdani's fuzzy inference method*) – фази-блок, вхідні та вихідні функції приналежності якого можуть мати будь-яку з наведених в табл. 1.1 форму);
- ✓ система інференції Сугено (*Sugeno's fuzzy inference method*) – фази-блок, вхідні та вихідні функції приналежності якого можуть бути лише константні або лінійні);
- ✓ адаптивна нейро-фаззи система інференції (*adaptive neuro-fuzzy inference system*) – фази-блок, добір параметрів функцій приналежності якого відбувається на основі заданих бажаних значень вхідних та вихідних величин фази-блока за допомогою методів тренування нейросіток.

В цих методичних вказівках розглядається моделювання систем лише першого типу.

Створені за допомогою Fuzzy Logic Toolbox фази-системи можуть бути промодельовані моделюючою програмою Simulink. Недоліком Fuzzy Logic Toolbox є відсутність можливості використання у системі інференції Мамдані функцій приналежності із змінними параметрами, тобто промодельовати фази-блок, параметри функцій приналежності якого динамічно змінюються в процесі моделювання. Промодельовати в Matlab/Simulink такий адаптивний фази-блок можна, лише написавши функцію користувача,

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

що містить алгоритм обробки інформації таким регулятором, у середовищі Matlab або C та помістивши цю функцію до Simulink-моделі за допомогою блока **Matlab Fcn** (функція Matlab) або **S-Function** (S-функція).

Засоби, що входять до складу Fuzzy Logic Toolbox, можна поділити на дві групи. До першої групи входять редактор фази-блоків (**FIS Editor**), редактор правил (**Rule Editor**) та редактор функцій приналежності (**Membership Function Editor**). Ці засоби служать для встановлення вхідних та вихідних змінних фази-блока, їх функцій приналежності та фази-правил. Друга група являє собою *read-only Tools*, тобто інструменти, які не дозволяють вносити зміни в фази-систему, а використовуються тільки для діагностики. До цієї групи належать переглядач тривимірної функції (**Surface Viewer**) та переглядач правил (**Rule Viewer**).

Для того, щоб викликати Fuzzy Logic Toolbox, необхідно набрати в командному рядку Matlab *fuzzy* та натиснути клавішу *Enter*. Після цього відкривається вікно, вигляд якого представлений на рис. 2.1.

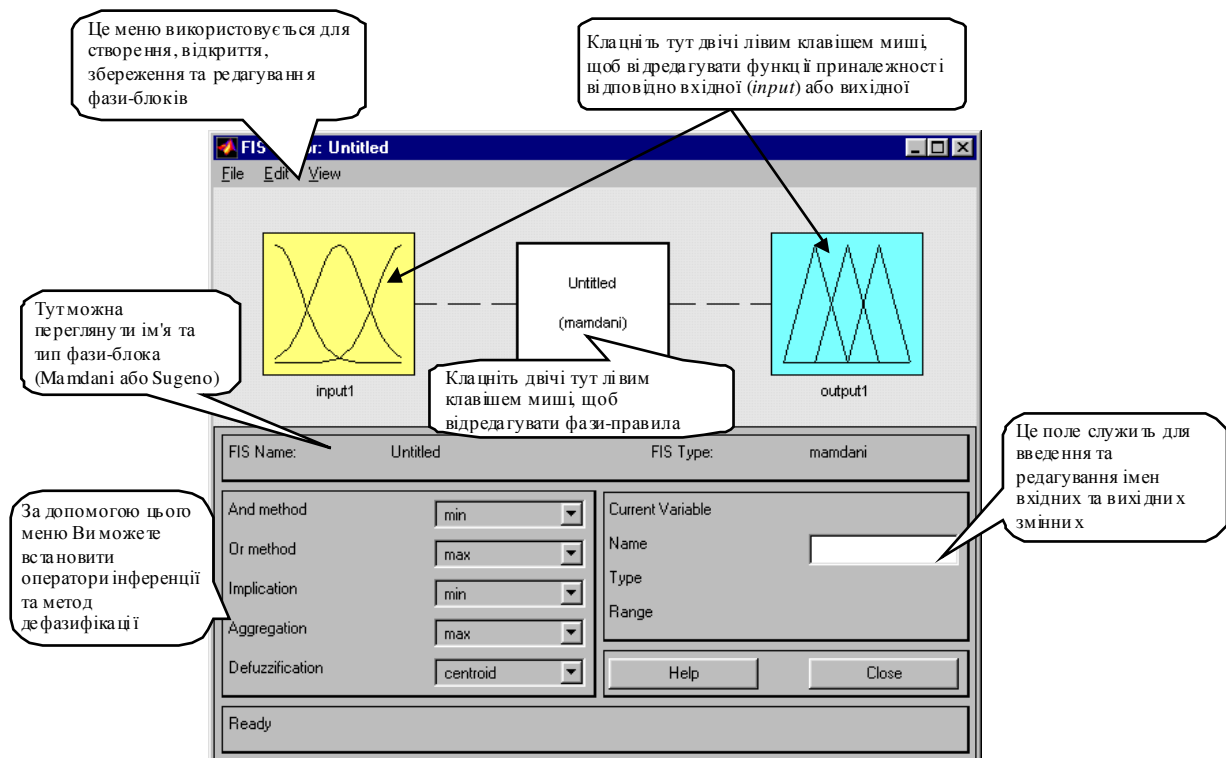


Рисунок 2.1 – Редактор фази-блоків

В цьому вікні Ви можете створити власний фази-блок. В тому випадку, якщо фази-блок містить більш, ніж одну вхідну чи вихідну змінну, треба вибрати в меню відповідно

Edit→**Add input** або **Edit**→**Add output**. Щоб відредагувати функції приналежності, треба клацнути двічі лівим клавiшем миші на іконці вхідної або вихідної змінної або вибрати в меню **View**→**Edit Membershipfunctions**. Після цього на екрані з'явиться вікно редактора функцій приналежності (рис. 2.2), за допомогою якого можна задати функції приналежності певної лінгвістичної змінної або відредагувати вже існуючі.

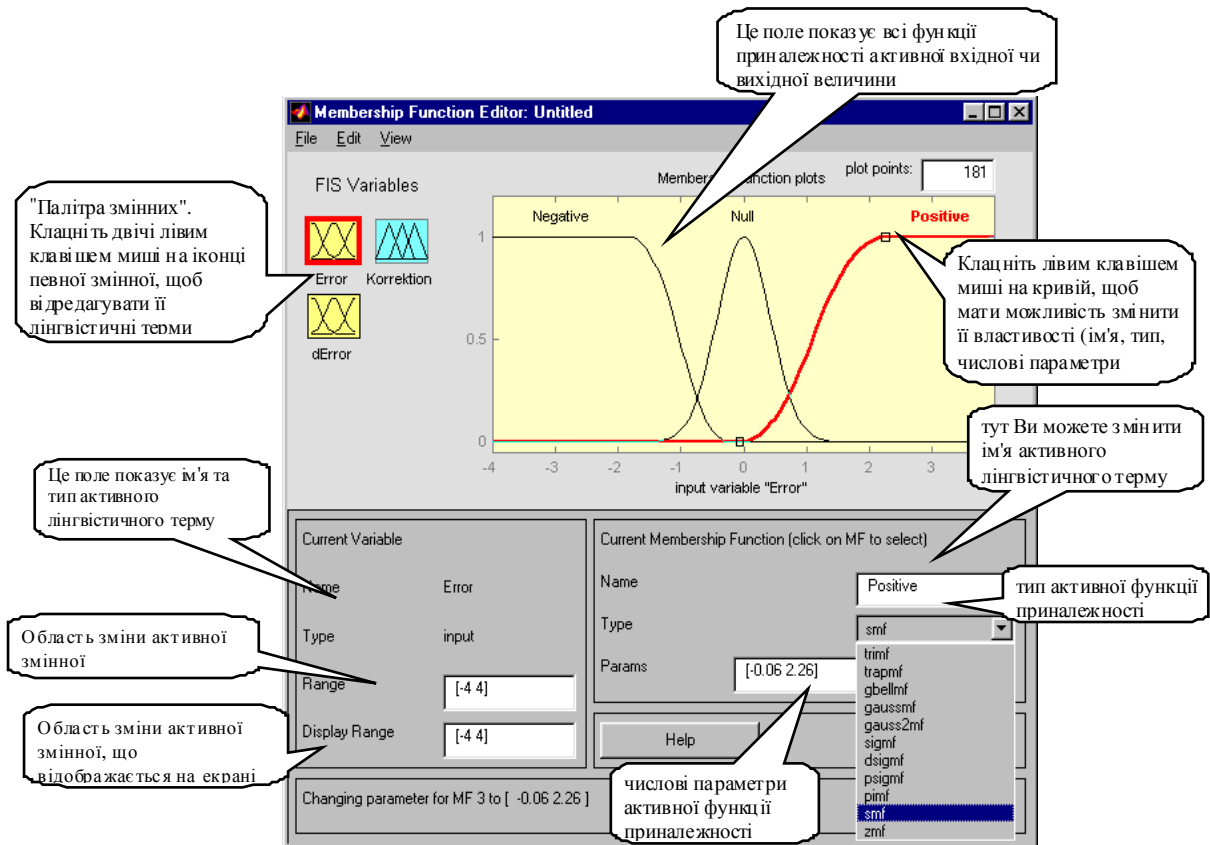


Рисунок 2.2 – Вікно редактора функцій приналежності

Для того, щоб ввести нову функцію приналежності, необхідно вибрати в меню **Edit**→**Add MFs...** або **Edit**→**Add custom MFs...**. В першому випадку відкривається таке вікно:

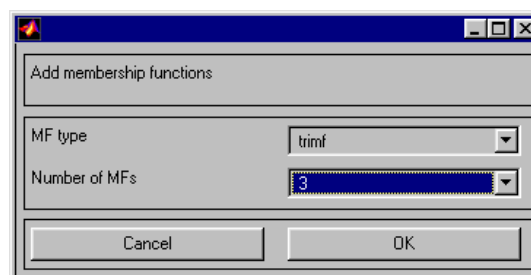


Рисунок 2.3 – Вікно параметрів стандартної функції приналежності

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

В другому випадку відкривається таке вікно:

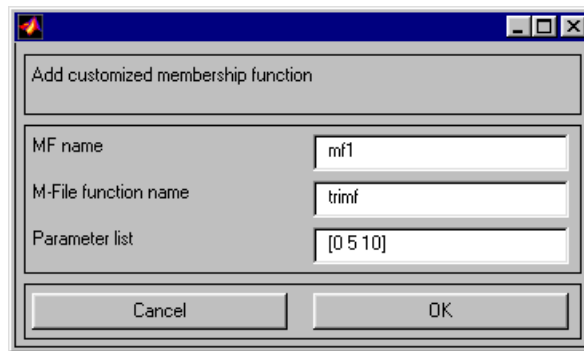


Рисунок 2.4 – Вікно параметрів функції приналежності користувача

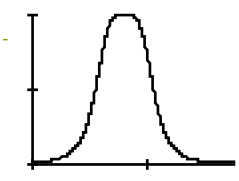
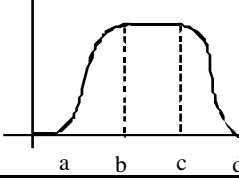


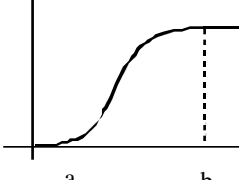
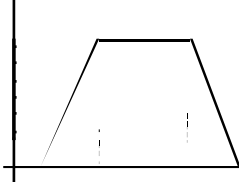
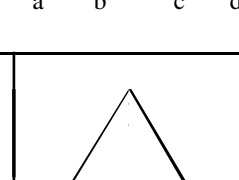
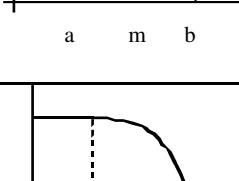
В цьому вікні можна ввести будь-яке ім'я терму, тип та межі функції приналежності.

Як видно із рис. 2.2, Fuzzy Logic Toolbox пропонує для системи інференції Мамдані 11 типів функцій приналежності. Їх назви та стислий опис представлені в табл. 2.1.

Таблиця 2.1 – Тип функцій приналежності лінгвістичних термів в Matlab

Тип функції приналежності	Опис	Математичне визначення	Графічне зображення
$dsigmf(a_1, c_1, a_2, c_2)$	різниця двох сигмоїдальних функцій приналежності	$\frac{1}{1 + e^{-a_1(x-c_1)}} - \frac{1}{1 + e^{-a_2(x-c_2)}}$	
$gaussmf(\sigma, c)$	симетрична функція Гауса	$e^{-\frac{(x-c)^2}{2\cdot\sigma^2}}$	
$gauss2mf(\sigma_1, c_1, \sigma_2, c_2)$	несиметрична крива, утворена двома функціями Гауса	$e^{-\frac{(x-c_1)^2}{2\cdot\sigma_1^2}}$, якщо $a \leq x \leq d$ $e^{-\frac{(x-c_2)^2}{2\cdot\sigma_2^2}}$, якщо $d \leq x \leq b$	

Продовження табл. 2.1

$gbellmf(a,b,c)$	дзвіноподібна крива	$\frac{1}{1 + \left \frac{x-c}{a} \right ^{2b}}, b > 0$	
$pimf(a,b,c,d)$	сплайнова крива, що проходить через точки a, b, c, d	-	
$psigmf(a_1, c_1, a_2, c_2)$	добуток двох сигмоїдальних функцій приналежності	$\frac{1}{1 + e^{-a_1(x-c_1)}} \cdot \frac{1}{1 + e^{-a_2(x-c_2)}}$	
$sigmf(a,c)$	сигмоїдальна функція приналежності	$\frac{1}{1 + e^{-a(x-c)}}$	
$smf(a,b)$	S-подібна сплайнова крива, що проходить через точки a, b	-	
$trapmf(a,b,c,d)$	трапецієподібна функція приналежності	$\begin{cases} 0 & \text{при } x < a \vee d \leq x \\ \frac{x-a}{b-a} & \text{при } a \leq x \leq b \\ 1 & \text{при } b \leq x \leq c \\ \frac{x-d}{c-d} & \text{при } c \leq x \leq d \end{cases}$	
$trimf(a,b,c)$	трикутна функція приналежності	$\begin{cases} 0 & \text{при } x < a \vee d \leq x \\ \frac{x-a}{m-a} & \text{при } a \leq x \leq m \\ \frac{x-d}{m-d} & \text{при } m \leq x \leq d \end{cases}$	
$zmf(a,b)$	Z-подібна сплайнова крива, що проходить через точки a, b	-	

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

Для введення фази-правил використовується редактор правил. Щоб його відкрити, треба вибрати в меню **View**→**Edit rules**. Після цього відкривається вікно редактора правил (рис. 2.5). Складання фази-правил проводиться автоматично на основі опису вхідних та вихідних лінгвістичних змінних, який був зроблений за допомогою редактора фази-блоків. Для того, щоб утворити нове правило, треба натиснути на кнопку **Add rule**, після чого обрати значення кожної вхідної змінної, логічний зв'язок між вхідними змінними та значення вихідної змінної, що відповідає створеній *ЯКЩО-частині* фази-правила. У випадку, якщо правило не містить якоїсь вхідної величини, треба обрати як вхідну величину **none**. Якщо змінна має бути проінвертована, треба поставити прапорець напроти **not**.

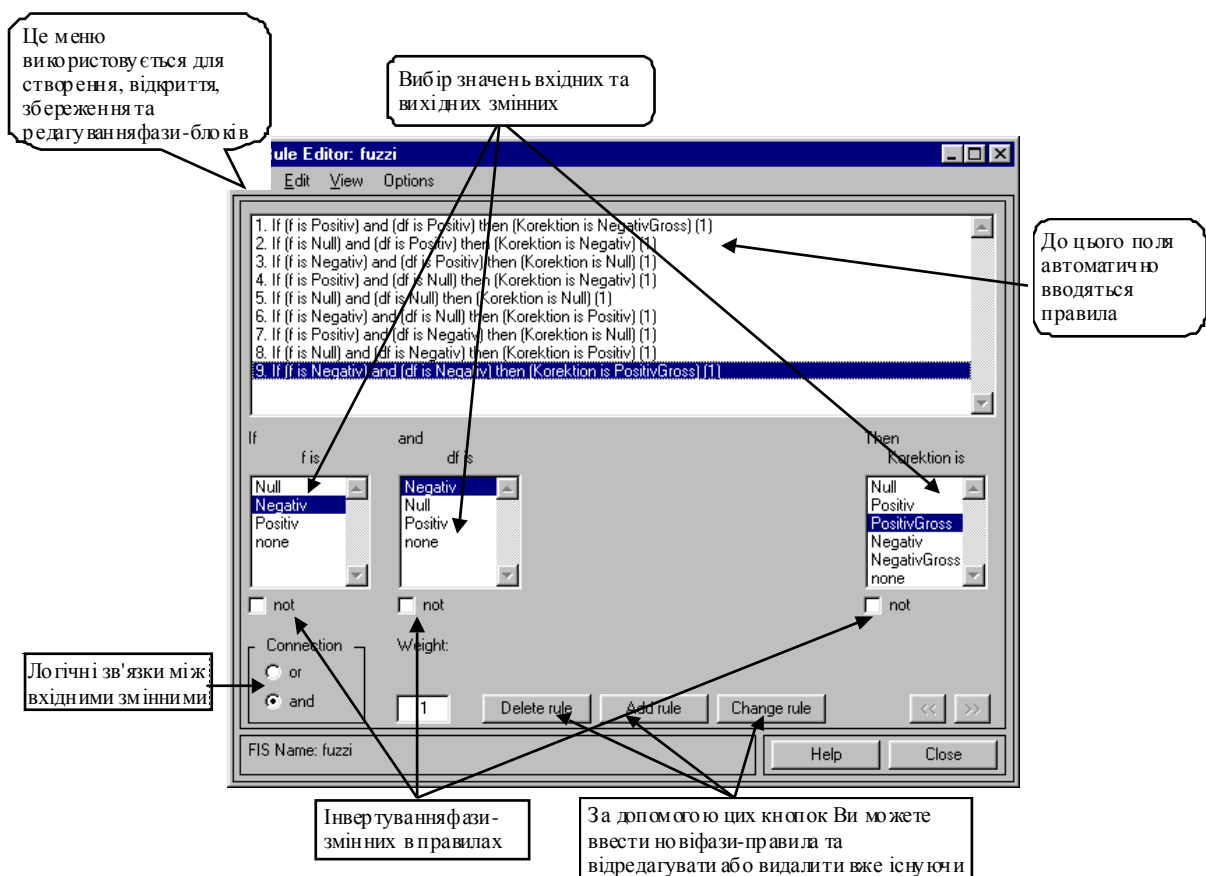


Рисунок 2.5 – Вікно введення фази-правил

Діагностика розробленого фази-блока здійснюється за допомогою переглядача тривимірної функції (**Surface Viewer**) та переглядача правил (**Rule Viewer**). Для того, щоб викликати вікно переглядача правил, треба обрати в меню **View**→**View rules**, після чого на екрані з'явиться вікно:

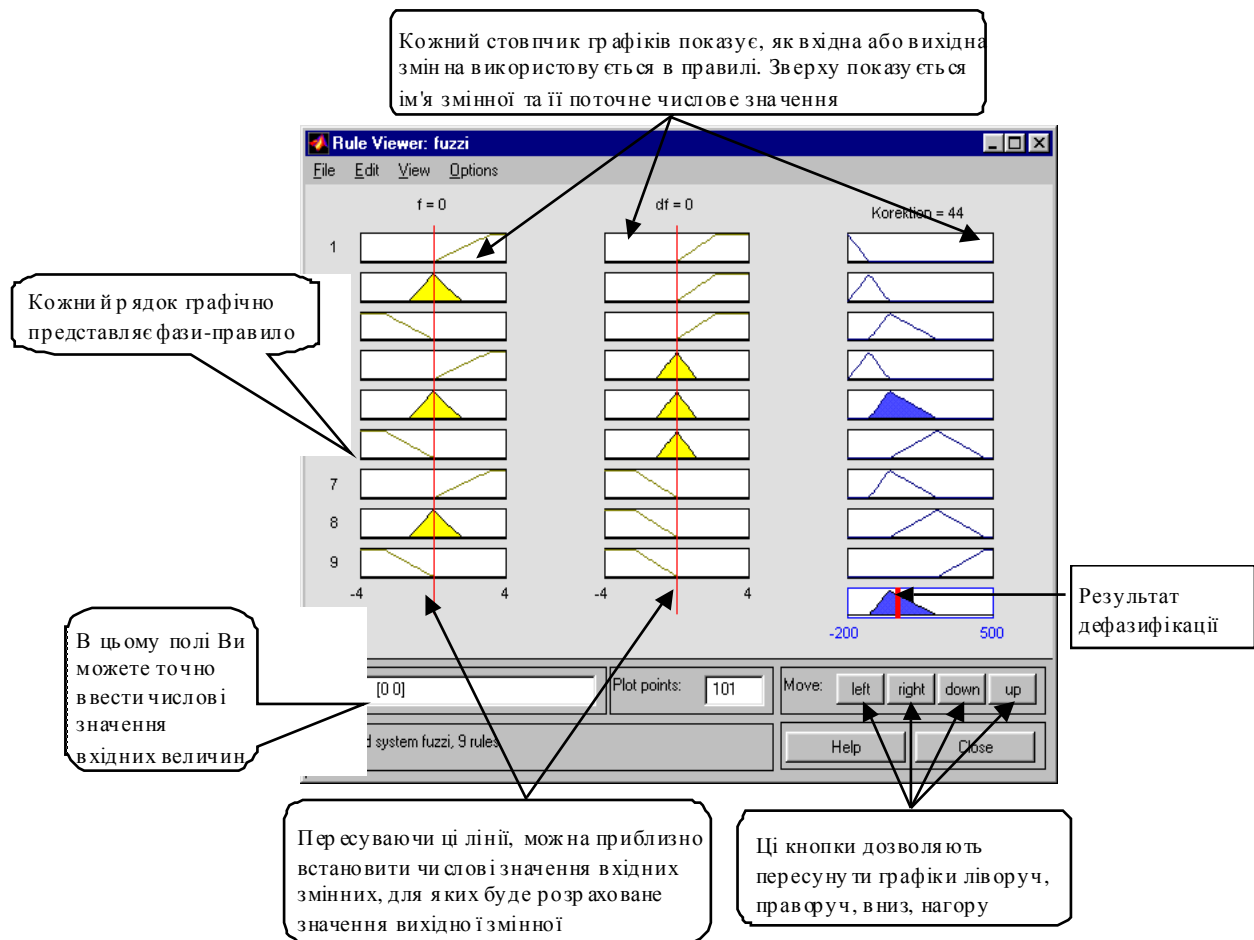


Рисунок 2.6 – Вікно тестування фазі-правил

Переглядач правил дає можливість для будь-якої комбінації числових значень вхідних змінних (звичайно, із області зміни цих змінних!) визначити відповідне числове значення вихідної змінної.

Переглядач тривимірної функції дає можливість представити залежність вихідної змінної від вхідних змінних у вигляді тривимірного графіка; особливо добре це дозволяє проаналізувати фазі-блоки, що мають дві вхідні та одну вихідну змінні. У випадку, якщо фазі-блок має більш, ніж одну вихідну змінну, або більш, ніж дві вхідні змінні, можна обрати таку комбінацію двох вхідних та однієї вихідної змінної, яка дозволяє найліпше представити передавальні властивості фазі-блока. Для того, щоб викликати переглядач тривимірної функції, необхідно обрати в меню **View**→**View surface**, після чого на екрані з'явиться вікно:

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

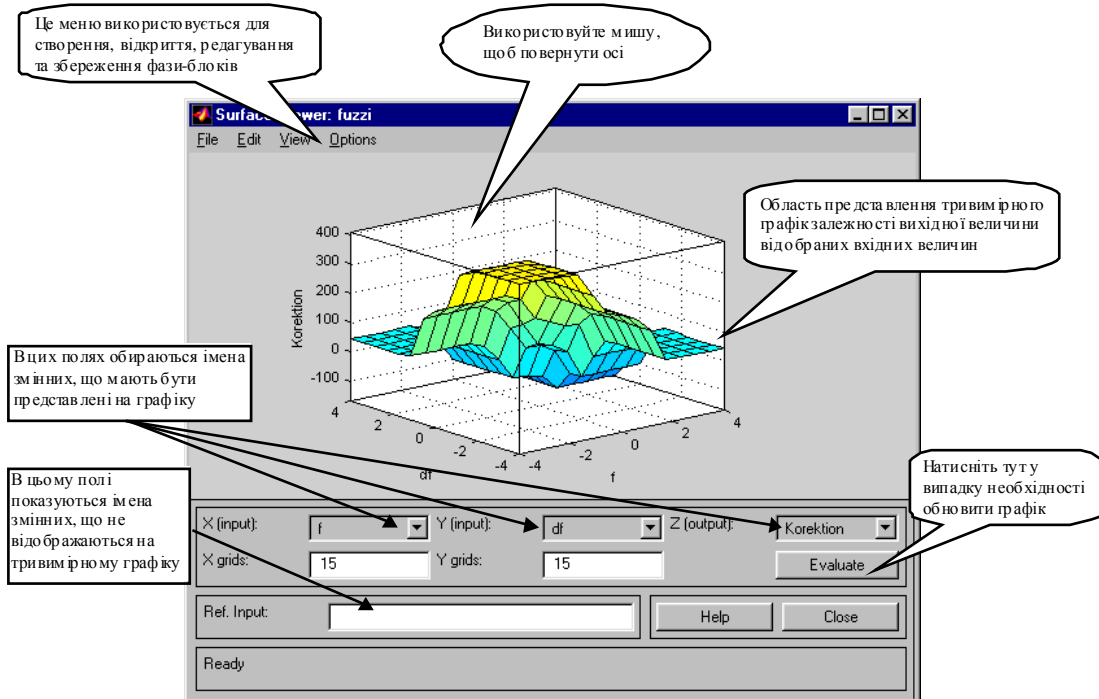


Рисунок 2.7 – Вікно переглядача тривимірного графіка передавальної функції фази-блока

Для того, щоб дати користувачу можливість переглянути установки фази-блока, що знаходиться в робочому просторі Matlab, у більш зручній формі, ніж це пропонує редактор фази-блоків, із можливістю копіювання інформації до буферу обміну для подальшого редагування за допомогою інших програм, було розроблено ряд функцій виводу на екран (**System Display Functions**). Огляд цих функцій представлений в табл. 2.2.

Таблиця 2.2 – Функції виводу на установок фази-блока екран

Ім'я та параметри функції	Опис функцій
<i>plotfis (fismat)</i>	Створює в вікні фігури діаграму фази-блока, що міститься в робочому просторі Matlab під ім'ям <i>fismat</i> . На діаграмі позначаються вхідні та вихідні змінні, вигляд їх термів, кількість фази-правил та тип системи інференції.
<i>plotmf (fismat,varType,varIndex)</i>	Виводить на екран графік функції приналежності лінгвістичної змінної фази-блока <i>fismat</i> . У списку параметрів вказуються тип змінної ('Input' або 'Output') та її індекс.

Продовження табл. 2.2

<i>gensurf (fismat)</i>	Виводить на екран тривимірний графік передавальної функції фази-блока <i>fismat</i> .
<i>getfis (fismat)</i>	Виводить на екран установки фази-блока <i>fismat</i> – тип фази-блока, кількість вхідних та вихідних змінних, їх імена, тип дефазифікації, тип операторів інференції.

Для того, щоб інтегрувати розроблений фази-блок до Simulink-моделі, що має містити фази-регулятор, та промодельювати цю систему регулювання, треба спочатку переконатись, що фази-блок, створений за допомогою редактора фази-блоків, є збереженим до робочого простору Matlab під іменем, яке використовується при параметризації Simulink-блока **Fuzzy-Logic-Controller**. Для того, щоб зберегти фази-блок до робочого простору Matlab, треба обрати в меню редактора фази-блоків **File**→**Save to workspace** або **File**→**Save to workspace as**. У випадку, якщо фази-блок був раніше спроектований та збережений на твердому диску, цю операцію можна виконати безпосередньо в робочому просторі Matlab. Для цього в командному рядку необхідно набрати команду:

$$fuz = readfis('Controller')$$

fuz – ім'я FIS-матриці – матриці, що відображує фази-блок у робочому просторі Matlab;

Controller - ім'я файлу, який містить фази-блок.

Після цього треба відкрити Simulink-бібліотеку **fuzblock**. У результаті відкривається вікно:

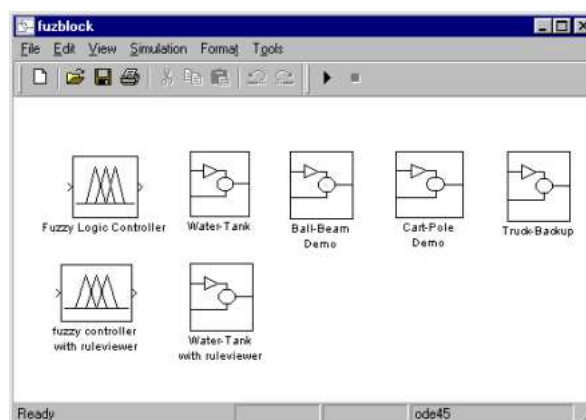


Рисунок 2.8 – Simulink-бібліотека фази-блоків

З цієї бібліотеки до Simulink-моделі копіюється один із таких блоків:

2 МОДЕЛЮВАННЯ СИСТЕМ ФАЗИ-РЕГУЛЮВАННЯ ЗА ДОПОМОГОЮ ПРОГРАМНОГО ПАКЕТА MATLAB/SIMULINK

- ✓ **Fuzzy Logic Controller;**
- ✓ **Fuzzy Controller with ruleviewer.**

Використання блока **Fuzzy Controller with ruleviewer** дає можливість у процесі моделювання спостерігати за змінами вхідних та вихідних величин фази-блока за допомогою переглядача правил.

Обраний блок необхідно параметризувати. Для цього треба двічі клацнути лівим клавшем миші на іконці блока. Якщо використовується блок **Fuzzy Logic Controller**, то після цього на екрані з'явиться вікно:

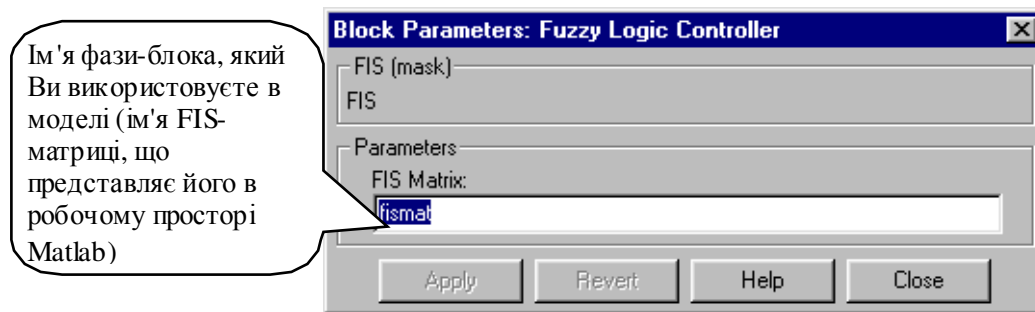


Рисунок 2.9 – Вікно параметрів блока **Fuzzy Logic Controller**

Блок **Fuzzy Controller with ruleviewer** має досить складну структуру, яка представлена на рис. 2.10.

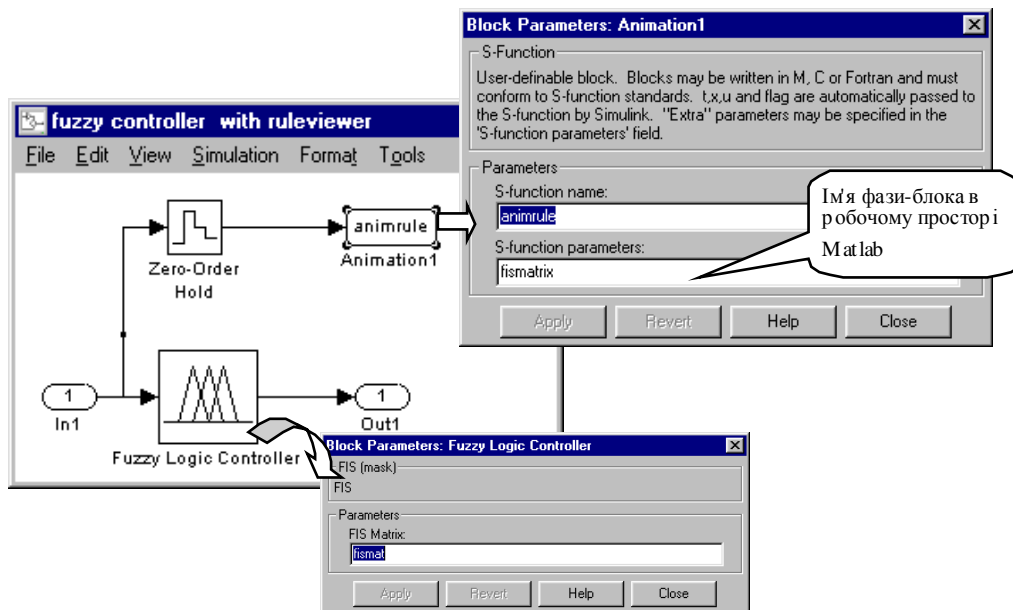


Рисунок 2.10 – Структура блока **Fuzzy Controller with ruleviewer**

Параметризація блока **Fuzzy Controller with ruleviewer** здійснюється, як показано на рис. 2.10.

3 ПРАКТИЧНЕ ЗАСТОСУВАННЯ ФАЗИ-ЛОГІКИ В СИСТЕМАХ РЕГУЛЮВАННЯ

3.1 Постановка завдання

Прикладом об'єкта практичного застосування фазис-логіки в системах регулювання є електропривод із класу низькошвидкісних слідкуючих приводів. Такі приводи мають низку характерних ознак:

- ✓ велике значення передавального числа в редукторі, яке може сягати десятків та сотень, призводить до виникнення розбіжностей на виході редуктора;
- ✓ масивна та відносно еластична механічна конструкція у поєднанні з великим значенням передавального числа в редукторі робить можливою появу низькочастотних механічних коливань;
- ✓ коефіцієнт моменту пружності

$$\gamma = 1 + \frac{J_2}{J_1} \quad (3.1)$$

прагне досягти одиниці, тобто електродвигун досить слабо демпфює виникаючі механічні коливання. Затухання коливань здійснюється завдяки дії сил сухого тертя в механічній частині приводу. Слід зауважити, що здатність системи до коливань зменшується із зростанням величини γ . Це ілюструє рис. 3.1, на якому наведені графіки перехідних процесів у двомасовій системі електроприводу, що враховує наявність зазору в редукторі, при трьох значеннях коефіцієнту моменту пружності. Найбільшу схильність до коливань демонструє система з $\gamma = 1.2$, при $\gamma = 10$ перехідний процес носить аперіодичний характер.

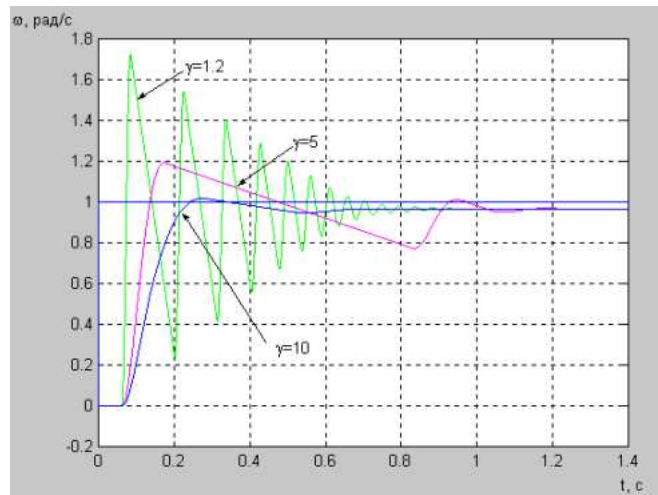


Рисунок 3.1 – Відпрацювання завдання на швидкість двомасовою системою електроприводу

Такі ознаки властиві цілій низці слідкуючих приводів у широкому діапазоні потужностей. До цього класу належать, наприклад, електроприводи радіотелескопів, антен, сонячних печей. При складанні математичних моделей таких приводів досить часто потрібно враховувати вплив таких явищ, як зазор у редукторі, дія сил сухого тертя, наявність еластичності механічного вала.

Наявність еластичності притаманна багатьом типам передавальних ланцюгів – довгим валам, пружинам, пасовим передачам. Ці ланцюги мають еластичні й коливальні властивості, які не можна ігнорувати. При наявності таких ланцюгів в механічній частині електроприводу при переносі моменту виникає перекручування вала. Це явище призводить, з одного боку, до появи коливань, з іншого, до перевантаження передавальних пристроїв та самого електродвигуна. В системах регулювання положення будь-які коливання вихідної величини взагалі неприпустимі, тому що ці системи мають здебільшого досить високі вимоги до точності.

На рис. 3.2 наведено зображення моделі двомасової системи. Електродвигун та робоча машина поєднані між собою за допомогою еластичного вала.

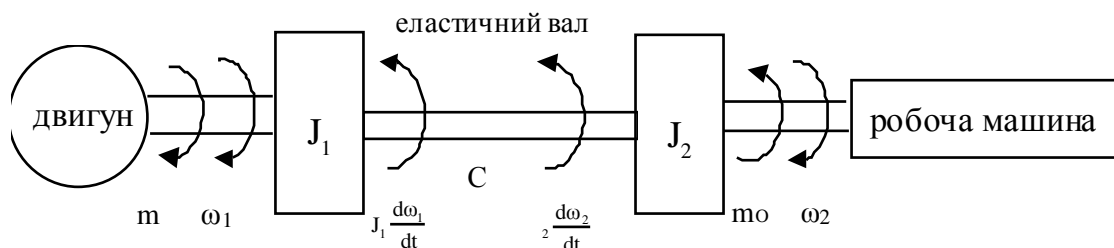


Рисунок 3.2 – Двомасова система електроприводу

Припустимо, що маса вала дорівнює нулю. Рівняння руху системи має вигляд:

$$m - J_1 \frac{d\omega_1}{dt} - J_2 \frac{d\omega_2}{dt} - m_0 = 0 \quad (3.2)$$

де J_1 – сумарний момент інерції з боку двигуна;

J_2 – сумарний момент інерції з боку робочої машини, приведений до швидкості двигуна;

m – момент двигуна;

m_0 – момент опору;

ω_1 – швидкість двигуна;

ω_2 – швидкість робочої машини, приведена до швидкості двигуна.

Еластичний вал з коефіцієнтом жорсткості C переносить обертаючий момент m_{12} :

$$m_{12} = C(\alpha_1 - \alpha_2) = m - J_1 \frac{d\omega_1}{dt} = m_0 + J_2 \frac{d\omega_2}{dt} \quad (3.3)$$

Циклічна частота власних незатухаючих коливань системи [3]:

$$\omega_0 = \sqrt{C \cdot \frac{J_1 + J_2}{J_1 \cdot J_2}} \quad (3.4)$$

Досить часто при аналізі систем електроприводу необхідно враховувати вплив сил сухого тертя та залежного від швидкості моменту демпфірування. Обертаючий момент сухого тертя визначається за формулою:

$$M_T = K_T \cdot \text{sign}\omega \quad (3.5)$$

де K_T – коефіцієнт сухого тертя.

Момент демпфірування визначається за формулою:

$$M_D = K_D \cdot (\omega_1 - \omega_2) \quad (3.6)$$

В деяких механічних передавальних ланцюгах досить часто спостерігаються зазори. При цьому в динамічному режимі не відбувається жорсткий перенос моменту в межах кута $\pm \Delta\alpha$. Результатом цього є те, що при лінійній зміні вхідної величини вихідна величина є нелінійною, внаслідок чого виникає часовий зсув фаз між вхідною та вихідною величинами та усталена помилка (рис. 3.3).

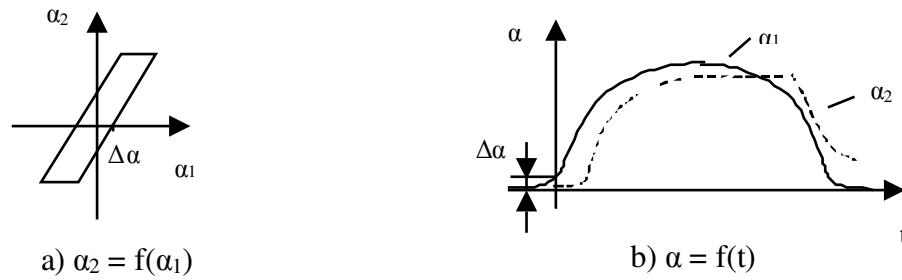


Рисунок 3.3 – Вплив зазорів

Структурна схема механічної частини системи електроприводу, складена згідно з рівняннями (3.2), (3.3) з врахуванням дії моменту сухого тертя (3.5), моменту демпфювання (3.6) та зазору в редукторі наведена на рис. 3.4 [3].

Зважаючи на усі вказані особливості двомасової системи, слід зазначити, що побудова класичних регуляторів для неї, які б враховували вплив нелінійностей, є досить складним завданням, яке не завжди має однозначне рішення. Методи фаззи-логіки, які засновані на обробці нечіткої інформації, дають при регулюванні значно кращі результати, особливо коли до контуру регулювання входять також і класичні регулятори, оптимізовані за традиційними методами (модульний оптимум, симетричний оптимум, настроювання за мінімумом резонансного максимуму), що дозволяє поєднати достоїнства класичної концепції регулювання та методів, які засновані на теорії штучного інтелекту. Треба зауважити, що фаззи-логіка в "чистому вигляді" використовується зараз дуже обмежено, набагато перспективнішим є її поєднання із нейросітками, що дозволяє створювати гнучкі системи регулювання, здатні до самонастроювання.

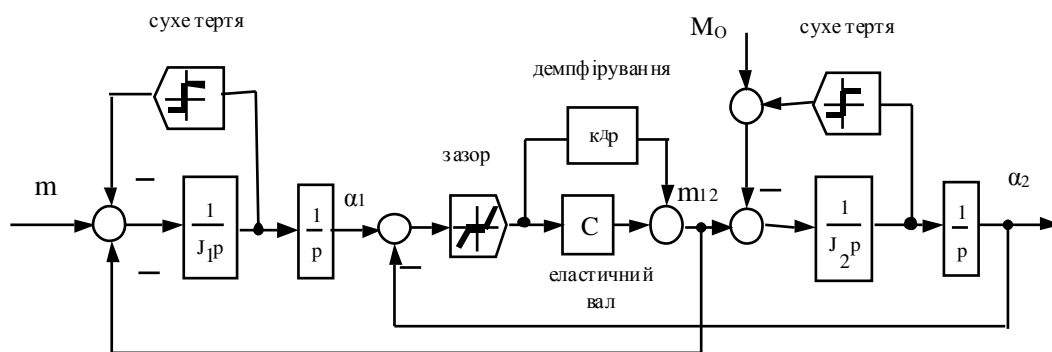


Рисунок 3.4 – Структурна схема механічної частини двомасової системи електроприводу

3.2 Лабораторний практикум

Лабораторна робота 1. Використання неадаптивного фазо-регулятора в системі автоматизації

1 Постановка завдання

Об'єктом дослідження є електропривід постійного струму, що приводить до руху робочу машину (параметри системи представлені в табл. 3.1).

Таблиця 3.1 – Параметри системи, що підлягає дослідженню

Найменування параметру	Числове значення	Одиниця виміру
Номінальна потужність електродвигуна P	160	кВ
Кількість пар полюсів $2p$	6	–
Номінальний струм електродвигуна I	390	А
Номінальна напруга електродвигуна U	440	В
Номінальна частота обертання електродвигуна n	980	об/хв
Величина опору якірного ланцюга $R_{Я}$	0,025	Ом
Момент інерції електродвигуна J_1	58,7	кг·м ²
Стала двигуна k	4,18	В·с
Стала часу якірного ланцюга $T_{Я}$	82	мс
Механічна стала часу T_M	84	мс
Коефіцієнт підсилення перетворювача $K_{П}$	68,35	В
Стала часу перетворювача $T_{П}$	1,7	мс
Коефіцієнт жорсткості вала C	$2,06 \cdot 10^5$	Н·м
Момент інерції робочої машини J_2	29,35	кг·м ²

Математична модель системи складається з урахуванням еластичних властивостей довгого вала, наявності зазору в редукторі та дії моменту сил сухого тертя в двигуні та робочому механізмі. Система містить класичні регулятори струму, швидкості двигуна та

положення, які оптимізовані за модульним оптимумом (табл. 3.2). Структурна схема системи, що підлягає дослідженню, надається на рис. 3.6.

Таблиця 3.2 – Розрахунок системи підпорядкованого регулювання положення

Регулятор	Розрахунок параметрів
ПІ-регулятор струму	$T_{PC} = T_{Я} = 0.082 \text{ с}$ $T_O = 2 \cdot T_{\Sigma} \cdot K_C = 2 \cdot T_{II} \cdot K_{II} \cdot \frac{1}{R_{Я}} = 2 \cdot 0.0017 \cdot 68.35 \cdot \frac{1}{0.025} =$
П-регулятор швидкості	$K_{PШ} = \frac{T_M}{2 \cdot T_{\Sigma Ш} \cdot K_{Ш}} = \frac{T_M}{2 \cdot T_{\Sigma Ш} \cdot \frac{R_{Я}}{k}} = \frac{0.084}{2 \cdot 0.0034 \cdot \frac{0.025}{4.18}} = 2065$
П-регулятор положення	$K_{PП} = \frac{1}{2 \cdot T_{\Sigma П} \cdot V_{П}} = \frac{1}{2 \cdot 0.0068 \cdot 1} = 73.529$

Задане значення вихідної величини системи змінюється за лінійним законом.

Система, що підлягає розгляду, має високі вимоги до точності, тому що йдеться про регулювання положення, а будь-які коливання та усталена помилка в цьому випадку є неприпустимими. Класичні регулятори не здатні забезпечити задовільну точність (рис. 3.5). Одним із засобів покращення передавальних властивостей системи є застосування фази-регулятора. Пропонується використати неадаптивний фази-регулятор, що підключається паралельно до класичного регулятора положення (рис. 1.15). Така система регулювання є досить стабільною та дозволяє значно покращити вид перехідного процесу, отримавши на виході системи майже експоненціальну криву вихідної величини – кута повороту робочого механізму.

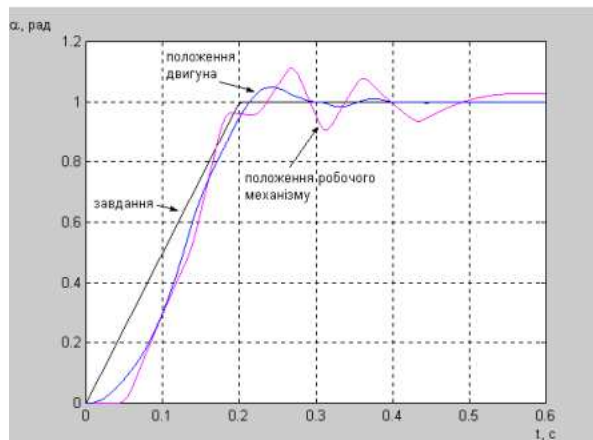


Рисунок 3.5 – Перехідні процеси в двомасовій системі електроприводу підпорядкованого регулювання з класичними регуляторами

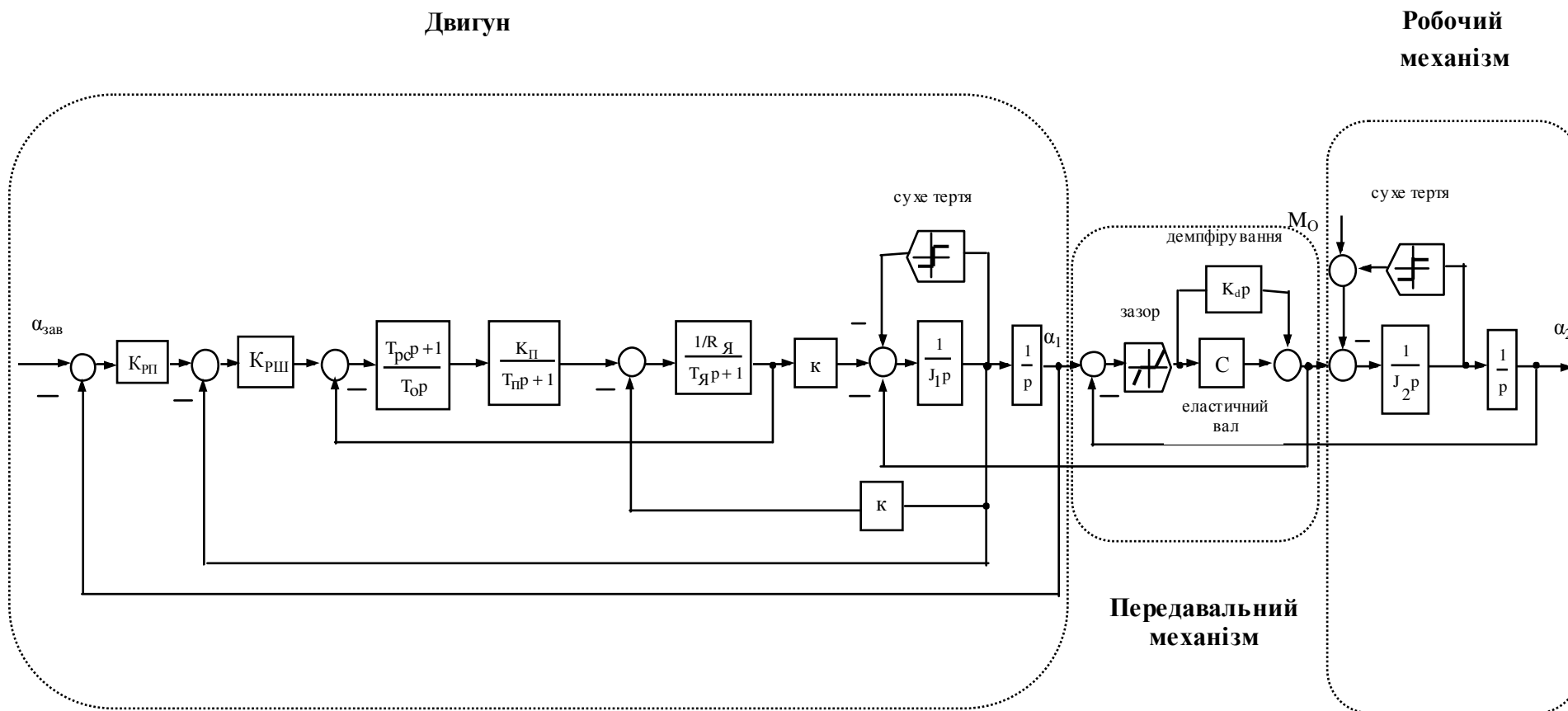


Рисунок 3.6 – Структурна схема системи електроприводу постійного струму з врахуванням еластичних властивостей довгого вала, наявності зазору в редукторі та дії моменту сил сухого тертя в двигуні та робочому механізмі

2 Мета роботи

Мета лабораторної роботи полягає в дослідженні системи електроприводу постійного струму за допомогою програмного пакета Matlab/Simulink з метою порівняння передавальних властивостей системи з підпорядкованим регулюванням положення з класичними ПІ-регулятором струму і П-регуляторами швидкості і положення та системи з поєднанням класичного підпорядкованого регулювання і фази-корекції.

3 Зміст роботи

3.1 Дослідження системи електроприводу з класичною структурою підпорядкованого регулювання:

- ✓ запустити середовище програмування Matlab 5.2;
- ✓ ввійти до програмного пакету Simulink, набравши в командному рядку команду *simulink*;
- ✓ додати директорію **x:\..\alex** до списку активних директорій, для чого вибрати в меню **File**→**Set Path**. Після цього відкриється вікно списку активних директорій (**Path Browser**), в якому слід вибрати в меню **Path**→**Add to Path** та ввести шлях до директорії **x:\..\alex**;
- ✓ відкрити в середовищі Simulink файл **x:\..\alex\system.mdl**;
- ✓ ввести параметри системи, що підлягає дослідженню, до робочого простору Matlab 5.2, для чого клацнути двічі лівим клавiшем миші на блоці **Enter Parameter**. Встановлені параметри можна переглянути в робочому просторі Matlab;
- ✓ промодельовати систему, для чого в меню Simulink вибрати **Simulation**→**Start**;
- ✓ проаналізувати результати моделювання, для чого клацнути двічі лівим клавiшем миші на блоці **Plot**, який виконує функції виводу на екран графіків величин, що мають бути проаналізовані – завдання на положення, положення двигуна, положення робочого механізму, помилки регулювання й її похідної, швидкості двигуна та переданого через редуктор моменту m_{12} . За допомогою графіків визначити величину перерегулювання положення робочого механізму, інтервал зміни швидкості двигуна, помилки регулювання й її похідної;
- ✓ зафіксувати результати моделювання.

3.2 Дослідження системи електроприводу з класичною структурою підпорядкованого регулювання та паралельною фази-корекцією

3.2.1 Розробка фази-регулятора

Процес розробки фази-регулятора здійснюється за алгоритмом, розглянутим у розділі 1.3 (рис. 1.12).

1. Проведення аналізу системи. Встановлення структури контуру регулювання та вхідних і вихідних величин фази-регулятора.

Обирається одна з найбільш стабільних та простих для настроювання структур – контур з паралельною фази-корекцією, загальна структура якого представлена на рис. 1.15). Фази-регулятор підключається паралельно до класичного регулятора положення. Вхідними величинами фази-регулятора є помилка регулювання положення робочого механізму ($\alpha_{зав} - \alpha_2$) та похідна від неї ($\dot{(\alpha_{зав} - \alpha_2)}$). Вихідною величиною фази-контролера є коригувальний сигнал, який складається з вихідним сигналом традиційного пропорційного регулятора положення. Сума цих сигналів утворює завдання на швидкість двигуна.

2. Встановлення алгоритмічних ступенів свободи.

Практика показує, що на етапі моделювання тип фази-інференції не грає великої ролі (це можна легко перевірити, проекспериментувавши за допомогою редактора фази-блоків Fuzzy Logic Toolbox). Вибір операторів інференції має велике значення лише при практичній реалізації фази-блока, тому що він визначає швидкодію контуру регулювання та стабільність передавальних властивостей фази-блока. Проте рекомендується дотримуватись певних рекомендацій і використовувати загальноприйняті сполучення операторів окремих етапів інференції (табл. 1.7). Для проєктованого фази-блока пропонується використати такі установки:

- ✓ оператор імплікації – мінімум-оператор;
- ✓ оператор агрегації – максимум-оператор.

Рекомендується використати найбільш популярний метод дефазифікації - гравітаційний (*centroid*).

3 Встановлення параметричних ступенів свободи.

Цей етап є центральним кроком процесу розробки фази-регулятора, він складається з трьох кроків.

3.1 Встановлення можливих інтервалів зміни вхідних та вихідних величин

Результати моделювання системи з класичною системою регулювання дозволяють визначити інтервали зміни вхідних та вихідних величин. Помилка регулювання змінюється в інтервалі $[-0.5, 0.5]$, похідна від неї змінюється в інтервалі $[-10, 10]$. Сумісний аналіз кривих

зміни помилки регулювання, її похідної та швидкості двигуна дають можливість визначити інтервал зміни вихідної величини як $[-100,100]$.

3.2 Встановлення форми та параметрів функції приналежності

Вхідним змінним **Error** (помилка регулювання) та **D-Error** (похідна помилки регулювання) відповідає п'ять лінгвістичних термів:

- ✓ **PL** – Positive Large (позитивна велика);
- ✓ **P** – Positive (позитивна);
- ✓ **Z** – Zero (нуль);
- ✓ **N** – Negative (негативна);
- ✓ **NL** – Negative Large (негативна велика).

Форма граничних термів **PL** та **NL** приймається трапецієподібною (прямокутна трапеція, прямий кут якої відповідає точці на межі області визначення змінної). Форма термів **P**, **Z**, **N** приймається трикутною. Трикутна та трапецієподібна форми функцій приналежності вживаються в техніці регулювання найчастіше, тому що це дозволяє зменшити витрати машинного часу.

Границі термів були спочатку встановлені приблизно в межах інтервалів зміни вхідних величин та визначені остаточно після моделювання системи. Остаточно прийняті для лінгвістичних змінних **Error** та **D-Error** функції приналежності зображені на рис. 3.7 та рис. 3.8, а параметри їх термів представлені в табл. 3.3-3.4.

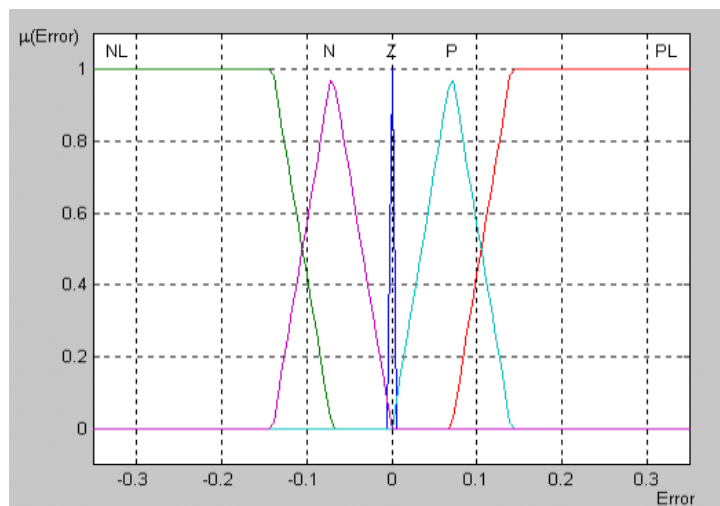
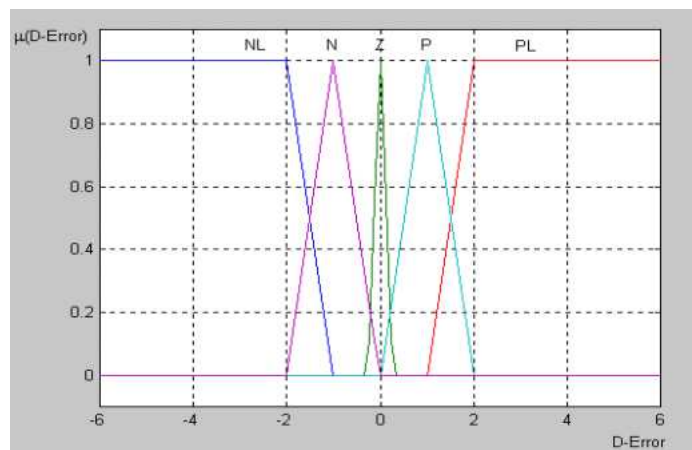


Рисунок 3.7 – Функція приналежності вхідної змінної **Error**

Таблиця 3.3 – Параметри функції приналежності вхідної змінної **Error**

Терм	Форма терму	Характерні точки
NL (Negative Large)	трапеція	[-0.5, -0.5, -0.14, -0.07]
N (Negative)	трикутник	[-0.14, -0.07, 0]
Z (Zero)	трикутник	[-0.005, 0, 0.005]
P (Positive)	трикутник	[0, 0.07, 0.14]
PL (Positive Large)	трапеція	[0.07, 0.14, 0.5, 0.5]

Рисунок 3.8 – Функція приналежності вхідної змінної **D-Error**Таблиця 3.4 – Параметри функції приналежності вхідної змінної **D-Error**

Терм	Форма терму	Характерні точки
NL (Negative Large)	трапеція	[-10, -10, -2, -1]
N (Negative)	трикутник	[-2, -1, 0]
Z (Zero)	трикутник	[-0.25, 0, 0.25]
P (Positive)	трикутник	[0, 1, 2]
PL (Positive Large)	трапеція	[1, 2, 10, 10]

Вихідній змінній **Korrektion** (корекція) відповідає дев'ять лінгвістичних термів:

- ✓ **PL** – Positive Large (позитивна велика);
- ✓ **PM** - Positive Middle (позитивна середня);
- ✓ **PS** - Positive Small (позитивна маленька);
- ✓ **P** – Positive (позитивна);
- ✓ **Z** – Zero (нуль);

- ✓ **N** – Negative (негативна);
- ✓ **NS** – Negative Small (негативна маленька);
- ✓ **NM** – Negative Middle (негативна середня);
- ✓ **NL** – Negative Large (негативна велика).

Як і у випадку вхідних змінних, форма граничних термів **PL** та **NL** приймається трапецієподібною, а форма "внутрішніх" термів **PM**, **PS**, **P**, **Z**, **N**, **NS**, **NM** - трикутною. Обрати параметри термів для вихідної змінної складніше, ніж для вхідних змінних, точно встановити їх можна лише після багаторазового моделювання. Остаточно приймається функція приналежності, що зображена на рис. 3.9. Параметри функції приналежності лінгвістичної змінної **Korrektion** подаються в табл. 3.5.

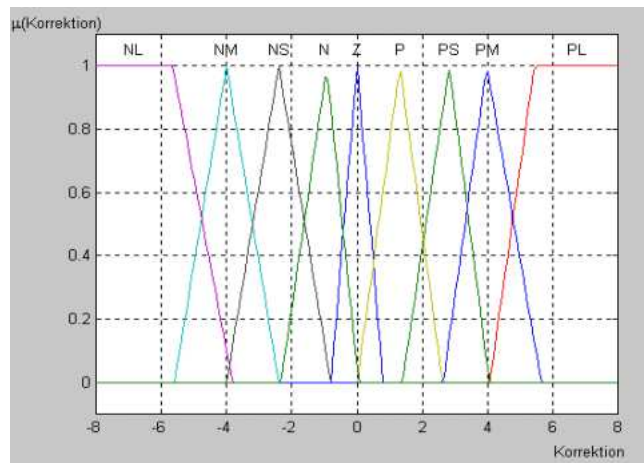


Рисунок 3.9 – Функція приналежності вихідної змінної **Korrektion**

Таблиця 3.5 – Параметри функції приналежності вихідної змінної **Korrektion**

Терм	Форма терму	Характерні точки
NL (Negative Large)	трапеція	[-8, -8, -5.61, -3.77]
NM (Negative Middle)	трикутник	[-5.6, -4, -2.4]
NS (Negative Small)	трикутник	[-4, -2.4, -0.8]
N (Negative)	трикутник	[-2.38, -0.8, 0]
Z (Zero)	трикутник	[-0.8, 0, 0.8]
P (Positive)	трикутник	[0, 1.312, 2.624]
PS (Positive Small)	трикутник	[1.39, 2.83, 4.05]
PM (Positive Middle)	трикутник	[2.64, 3.97, 5.65]
PL (Positive Large)	трапеція	[4.064, 5.44, 8, 8]

3.3 Складання бази фази-правил

Правила складаються на основі інтуїтивних знань експерта. В нашому випадку як експертні знання використовуються результати аналізу перехідних процесів, що були отримані при моделюванні системи з класичною системою регулювання. Деякі правила після моделювання системи з паралельною корекцією можуть бути уточненими.

Принцип складання фази-правил ілюструється за допомогою рис. 3.10, на якому представлений перехідний процес у системі з класичним регулятором. Як зазначалося вище, завдання паралельної фази-корекції полягає в покращенні вигляду перехідного процесу, зменшенні величини перерегулювання в ідеальному випадку до нуля. Тому найбільш небажаною є ділянка перехідного процесу, коли величина кута повороту перевищує величину завдання на кут повороту. На рис. 3.10 виділена саме ця ділянка та наведені у масштабі збільшення відповідні фрагменти кривих зміни помилки регулювання, її похідної та швидкості двигуна.

Розглянемо точки 1, 2, 3. В точці 1 помилка регулювання позитивна, а її похідна негативна велика. Це означає, що помилка регулювання має тенденцію до зменшення, тому корекція має бути незначна. Знак корекції +, тому що із кривої швидкості двигуна видно, що поблизу точки 1 швидкість зменшується, тобто рух сповільнюється, а його треба прискорити, щоб зменшити величину помилки. Отримуємо таке правило:

ЯКЩО Помилка регулювання Позитивна **I** Похідна помилки регулювання Негативна Велика, **ТО** Корекція Позитивна.

В точці 2 помилка регулювання негативна та має досить велике числове значення, а її похідна майже дорівнює нуля, тобто помилка регулювання поблизу точки 2 змінюється незначно. Це означає, що корекція не має бути досить сильною. Знак корекції визначимо, проаналізувавши графік швидкості двигуна. Поблизу точки 2 швидкість двигуна негативна, тому для компенсації корекція має бути позитивною – **Р**. Таким чином, отримуємо правило:

ЯКЩО Помилка регулювання Негативна **I** Похідна помилки регулювання Нульова, **ТО** Корекція Позитивна.

В точці 3 помилка регулювання позитивна, а її похідна позитивна та має досить велике числове значення, тобто поблизу точки 3 помилка регулювання має тенденцію до подальшого зростання. Виходячи з цього, можна стверджувати, що корекція має бути негативною з метою обмеження швидкості. Аналіз кривої швидкості показує, що величина цієї корекції не має бути дуже великою, тому що швидкість на цій ділянці вже є негативною і, надмірно її знизивши, можна отримати велике значення перерегулювання. Таким чином отримуємо два правила:

ЯКЩО Помилка регулювання Позитивна **I** Похідна помилки регулювання Позитивна,
ТО Корекція Негативна Маленька.

ЯКЩО Помилка регулювання Позитивна **I** Похідна помилки регулювання Позитивна
Велика, **ТО** Корекція Негативна Середня.

Складені таким чином та уточнені після моделювання правила наведені в табл. 3.6.

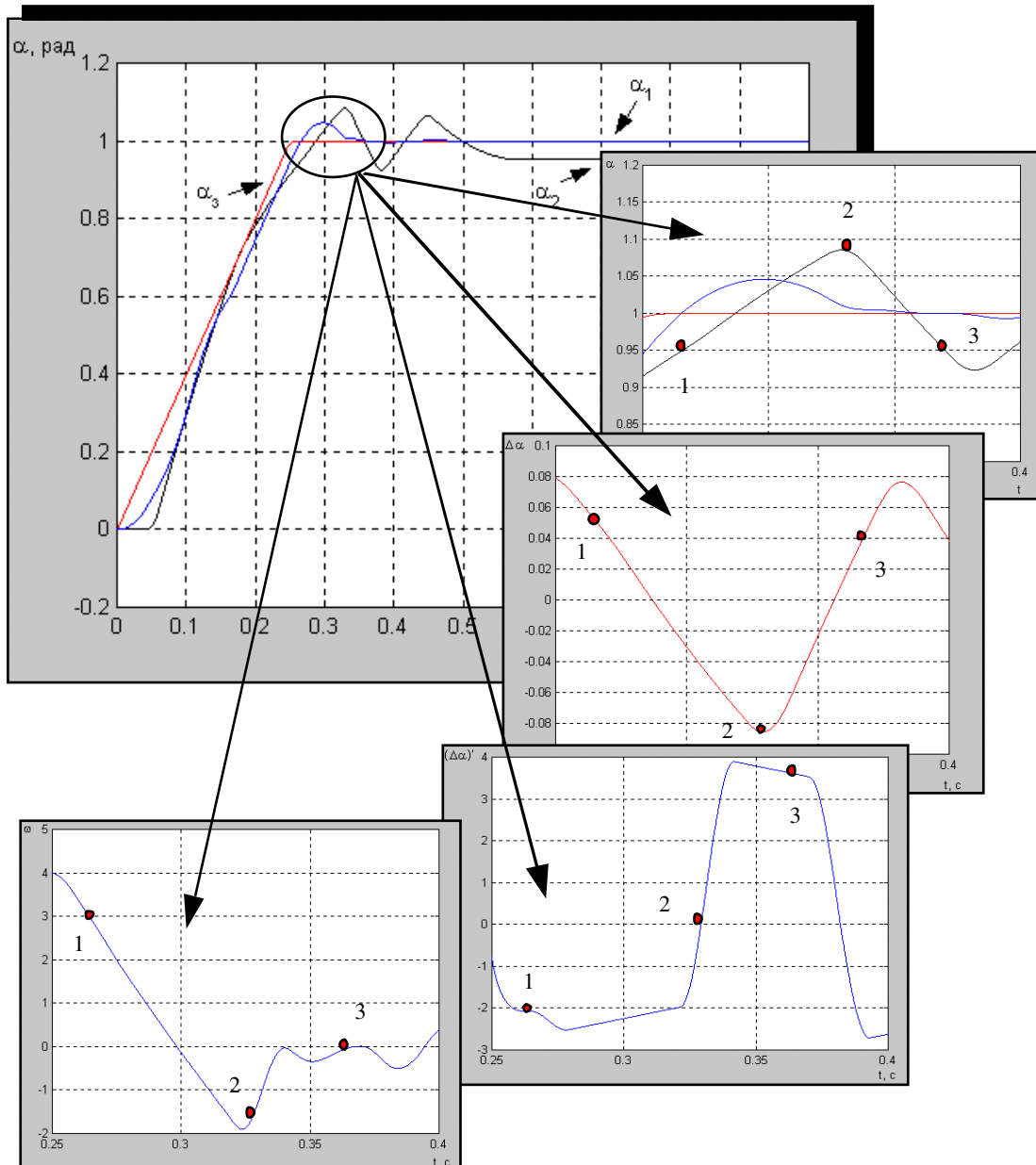


Рисунок 3.10 – Перехідні процеси в двомасовій системі електроприводу постійного струму

Таблиця 3.6 – Фази правила

		Error				
		PL	P	Z	N	NL
D-Error	PL	NL	NM	Z	N	Z
	P	NM	NS	Z	Z	P
	Z	PM	PS	Z	P	PS
	N	N	Z	Z	PS	PM
	NL	Z	P	Z	PM	PL

3.2.2 Моделювання системи електроприводу з класичною системою підпорядкованого регулювання та паралельною фази-корекцією:

- ✓ створити за допомогою Fuzzy Logic Toolbox фази-блок з параметрами, що наведені в пункті 3.2.1. Утворену FIS-матрицю зберегти на жорсткий диск та в робочий простір Matlab;
- ✓ вставити до Simulink-моделі, що міститься в файлі `x:\..\alex\system.mdl`, паралельно до пропорційного регулятора положення фази-блок (**Fuzzy Logic Controller** або **Fuzzy Controller with ruleviewer**). Запараметрувати цей блок;
- ✓ зберегти на жорсткий диск створену Simulink-модель під іменем **system1.mdl**;
- ✓ промоделювати систему;
- ✓ за допомогою блока **Plot** вивести на екран графіки завдання на положення, положення двигуна, положення робочого механізму, помилки регулювання та її похідної, швидкості двигуна й переданого через редуктор моменту **m₁₂**;
- ✓ зафіксувати результати моделювання;
- ✓ порівняти результати, що були отримані в першій частині лабораторної роботи, з результатами моделювання системи з паралельною фази-корекцією, зробити висновки.

4 Зміст звіту

В звіті повинні бути представлені:

- ✓ структурна схема системи, що підлягала дослідженню;
- ✓ параметри фази-регулятора для паралельної корекції – функції приналежності лінгвістичних змінних, фази-правила, тривимірний графік передавальної функції фази-блока;
- ✓ графіки зміни завдання на положення, положення двигуна, положення робочого механізму, помилки регулювання й її похідної, швидкості двигуна та переданого через редуктор моменту m_{12} для системи електроприводу з класичною підпорядкованою системою регулювання та для системи з паралельною фази-корекцією;
- ✓ аналіз результатів.

Лабораторна робота 2. Адаптація фази-логіки в системах регулювання

1 Постановка завдання

Розглянута в попередній лабораторній роботі система регулювання з неадаптивним фази-регулятором має декілька недоліків. Насамперед це її негнучкість та відсутність особливої точності. Це є її основною вадою, яка зумовлює неможливість використання фази-систем у “чистому вигляді” для виконання такого, наприклад, завдання, як точне позиціонування.

Щоб зробити фази-систему більш досконалою, було запропоновано об’єднати її з іншими формами штучного інтелекту, які дозволяють зробити систему фази-регулювання більш гнучкою. В цій роботі підлягає розгляду подібне об’єднання фази-системи з нейросітками [9], а точніше з алгоритмами навчання та тренування нейросіток.

Об’єктом досліджень є система електроприводу підпорядкованого регулювання швидкості. Як і в попередній роботі, з метою покращення якості роботи в систему вводиться паралельна фази-корекція, що зумовлена необхідністю демпфірування внутрішніх коливань в системі. Проте така система має невелику швидкодію, особливо коли йдеться про відпрацьовування стрибкоподібного завдання. Причиною є класичний ПІ-регулятор швидкості, який настроюється таким чином, щоб забезпечити аперіодичний перехідний процес в лінійній системі. Система відпрацьовує завдання без коливань, проте потребує багато часу, що не є бажаним в слідкуючих системах. З метою виправлення цього недоліку в

систему вводяться 2 пропорційноінтегральні фази-регулятори, що здійснюють адаптацію класичного ПІ-регулятора швидкості. Один із них є неадаптивним фази-регулятором (регулятор сталої часу), а інший (фази-регулятор коефіцієнту підсилення) має спроможність до самоадаптації. Причиною того, що фази-регулятор сталої часу не потребує адаптації, є те, що зміна сталої часу пропорційноінтегрального регулятора у відносно малих межах несуттєво впливає на процес регулювання, тому цим впливом можна знехтувати.

Для адаптивного регулятора сталої часу використовується непряма стратегія адаптації, що ґрунтується на моделі процесу (рис. 1.18). Як модель процесу виступає цільова функція, яку необхідно отримати на виході системи після певної кількості тренувань системи.

Структурна схема системи регулювання, що підлягає дослідженню, надається на рис. 3.11.

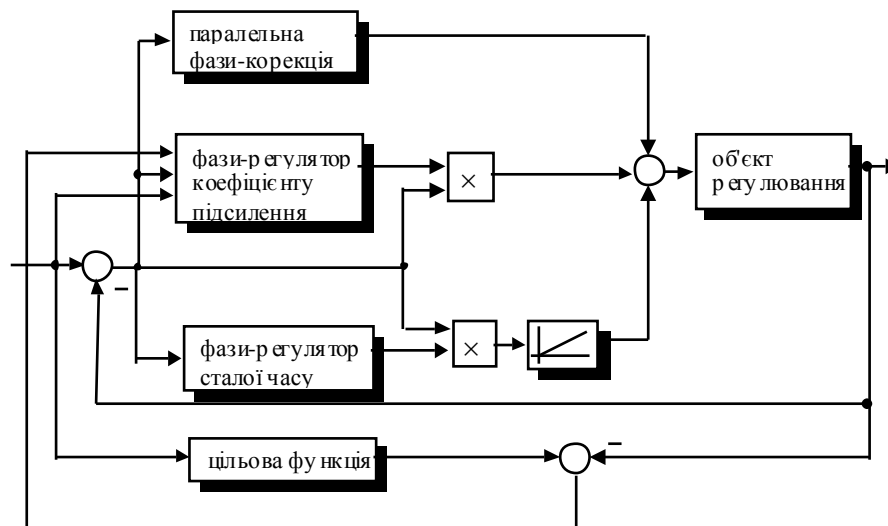


Рисунок 3.11 – Структурна схема контуру регулювання системи електроприводу з адаптивним класичним ПІ-регулятором та паралельною фази-корекцією

2 Мета роботи

Мета лабораторної роботи полягає в дослідженні системи електроприводу постійного струму за допомогою програмного пакета Matlab/Simulink з метою порівняння ефективності використання двох різних алгоритмів навчання нейросіток (методу Відроу – Хоффа та методу Кохонена) для адаптації фази-регуляторів в системах регулювання електроприводів.

3 Теоретична частина

3.1 Основи теорії нейросіток

3.1.1 Біологічна та математична моделі нейрона

Нейросітки - це системи, що використовують деякі з відомих принципів організації людського мозку і базуються на біологічних моделях [7]. Докладний розгляд цих біологічних моделей дозволяє порівняти мозок людини або високо розвиненої тварини з комп'ютером. Найважливішим результатом такого порівняння є висновки відносно паралельної структури як першого, так і останнього, бо це робить можливим здійснювати у комп'ютері імітацію роботи біологічних нервових систем, а це, в свою чергу, дозволяє надати комп'ютеру таку властивість, як спроможність до навчання.

Першою спроможною до навчання нейросіткою був перцептрон, який базувався на аспектах відпрацьовування та збереження інформації. За допомогою правил навчання після певної кількості тренувань перцептрон ставив у відповідність вхідним величинам бажані вихідні. Проте ця нейросітка не була спроможна вирішувати більш важливі проблеми, що спонукало до розробки нових більш потужних нейросіток та пошуку нових правил навчання.

У протиріччя до біології або до нейрології для нейрона не використовується модель, яка чітко описує всі його аспекти, а тільки та модель, що сама по собі є дуже грубим узагальненням [10]. Традиційна модель нейрона, яка представлена на рис. 3.12, розглядається тут як різновид суматора з граничним значенням на його виході.

Біологічний нейрон складається з тіла нейрона, до якого входить велика кількість дендритів (від 10000 до 100000) і виходить тільки один аксон. Якщо сума вхідних сигналів перевищує певну межу (у випадку біологічного нейрона - граничний потенціал), то нейрон активізується і посилає через вихідний провідник (аксон) короткочасний електричний імпульс до інших нейронів [11]. Таким чином, нейрон являє собою процесор з єдиною спроможністю - посылати електричні імпульси залежно від суми вхідних сигналів.

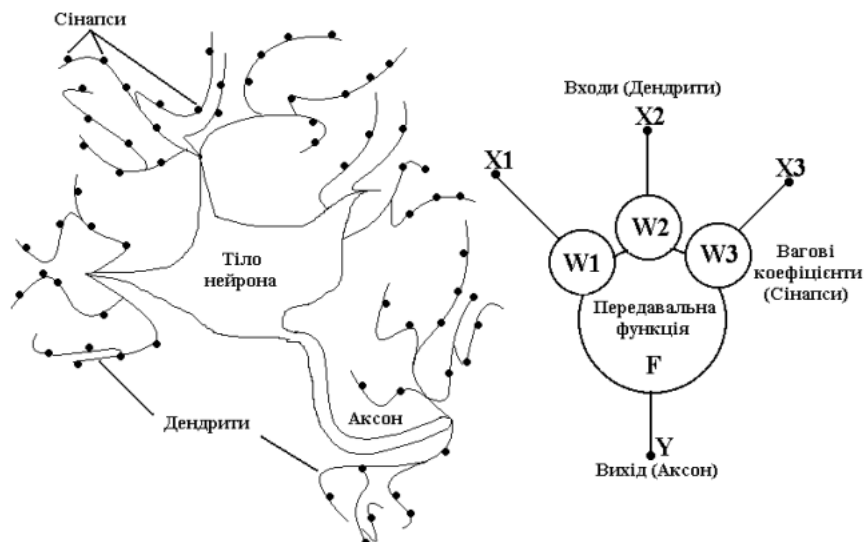


Рисунок 3.12 - Біологічна та математична моделі нейрона

Між входом одного нейрона (дендритом) та виходом попереднього (аксоном) є ще одна клітина, яка може підсилювати або подавляти сигнал, який до неї надходить. Ця клітина називається синапс і знаходиться найчастіше просто на тілі нейрона. Для процесу навчання синапс має дуже велике значення, тому що з часом він може змінювати свою властивість підсилювати або подавляти сигнал.

Комунікація між нейронами завжди відбувається тільки в такому напрямку:

Сінапс → Дендрит → Тіло нейрона → Аксон → Сінапс і т. д.

Штучний нейрон можна представити як процесор, який приймає тільки два стани - активний та пасивний. Тому відповідно до бінарної логіки надамо активному стану значення **1**, а пасивному стану значення **0**. Вхідними сигналами нейрона є імпульси, що передаються через дендрити, які, в свою чергу, є виходами інших нейронів. Проте перед тим, як досягти тіла нейрона, вхідні сигнали проходять через синапси. Цього можна легко досягти шляхом множення вхідних сигналів на позитивне (ефект підсилення) або негативне (ефект ослаблення) число. Ці числа отримали назву вагових коефіцієнтів.

Сигнал на виході нейрона з'являється тільки тоді, коли сума всіх вхідних сигналів перевищує певне граничне значення. При побудові моделі нейрона це досягається шляхом порівняння суми множення всіх вхідних сигналів на величини синапсів або, що одне й те саме, на відповідні вагові коефіцієнти з граничним значенням. Якщо сума більша, ніж граничне значення, то на виході нейрона буде встановлено одиницю, а якщо ні - то нуль. Наочно модель нейрона представлена на рис. 3.13.

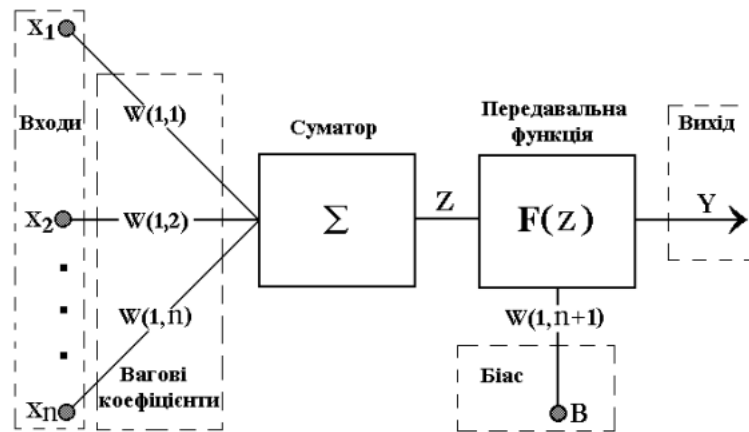


Рисунок 3.13 - Функціональна модель нейрона

Для урахування граничного значення при підсумовуванні вхідних сигналів вводиться допоміжний вхід **B** – біас.

Вхідні сигнали X_1, \dots, X_n та **B** проходять через вагові коефіцієнти $W_{(1,1)}, \dots, W_{(1, n+1)}$, підсумовуються і проходять через передавальну функцію **F** на вихід. Вхідне значення біасу завжди дорівнює одиниці.

Виходячи із схеми рис. 3.13, значення на виході суматора розраховується за формулою:

$$Z(w, x) = \sum_{i=1}^n (x_i \cdot w_{1,i} - B \cdot w_{1,i+1}) \quad (3.7)$$

Ця функція називається функцією активності [7].

Значення на виході нейрона є функцією від функції активності:

$$Y = F(Z) = F(w, x, Z) \quad (3.8)$$

Як передавальна функція може бути використана одна із стандартних функцій або будь-яка інша, на розсуд проектувальника нейросітки.

3.1.2 Нейросітка

Для того, щоб отримати найпростішу нейросітку, треба об'єднати між собою декілька нейронів. На рис. 3.14 представлена нейросітка з двома нейронами, яка має три входи (X_1 , X_2 та **B**) та два виходи (Y_1 та Y_2).

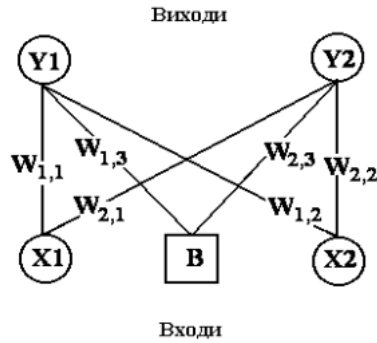


Рисунок 3.14 - Схема нейросітки

Щодо параметрів нейронів, з яких будується нейросітка, немає жодних усталених законів або правил. Наприклад, передавальні функції нейронів можуть бути однакові за своїм виглядом та параметрами або відрізнятися одна від одної. Це стосується також і зв'язків між нейронами, що є дуже актуальним при розгляді багатошарових нейросіток.

3.1.3 Правила навчання і тренування нейросіток

Найважливішою складовою частиною моделі нейросітки є процес навчання. Метою процесу навчання є визначення таких вагових коефіцієнтів $W_{(1,1)}, \dots, W_{(1, n+1)}$, при яких сітка буде реагувати заданими вихідними значеннями на задані вхідні значення. Щоб досягти цього, необхідно на кожному етапі тренування порівнювати дійсне вихідне значення нейросітки з бажаним і на підставі цього змінювати вагові коефіцієнти таким чином, щоб на подальших етапах тренування дійсне та бажане вихідне значення сходилися [10].

Існує велика кількість правил тренування нейросіток, саме від них залежить швидкість та точність навчання, звичайно, за інших рівних умов. Тому виникає проблема пошуку оптимального правила навчання, яке б задовольняло вимогам швидкості та точності процесу навчання.

Одним з алгоритмів навчання є перцептрон. Правило перцептрон відповідає Δ -правилу або правилу Відроу-Хоффа і має такий математичний опис:

$$w_{i,j}^k = w_{i,j}^{k-1} + x_j \cdot (a_i - y_i), \quad (3.9)$$

де $w_{i,j}^k, w_{i,j}^{k-1}$ – нове і попереднє значення вагових коефіцієнтів;

x_j – вхідний сигнал;

a_i – бажаний вихідний сигнал;

y_i – дійсний вихідний сигнал.

Інші правила є більш досконалішими, вони містять в собі параметр швидкості навчання або коефіцієнт навчання і належать до так званого другого покоління правил тренування. Одним з цих правил є правило Відроу - Хоффа, яке має такий вигляд:

$$w_{i,j}^k = w_{i,j}^{k-1} + lr \cdot x_j \cdot (a_i - y_i) \quad (3.10)$$

де lr - коефіцієнт навчання.

Другий клас алгоритмів навчання містить низку правил, що мають спроможність до самоадаптації. Найвідомішим з них є правило Кохонена:

$$w_{i,j}^k = w_{i,j}^{k-1} + lr \cdot (x_j - w_{i,j}^{k-1}) \quad (3.11)$$

3.3 Адаптація фаззи-логіки в системах регулювання

Метою об'єднання фаззи-логіки з іншими формами штучного інтелекту є удосконалення фаззи-логіки. В цій роботі розглядається об'єднання фаззи-системи з алгоритмами навчання та тренування нейросіток.

Як зазначалося при розгляді теоретичних основ фаззи-регулювання, форма термів вхідної фаззи-змінної, площа термів вихідної фаззи-змінної та значення діапазонів зміни цих змінних визначаються спеціалістом, виходячи з його досвіду та знань в галузі, де застосовується фаззи-система. Але спеціаліст - це людина, а людині властиво помиляться. Тому необхідно ввести певну корекцію вихідного значення фаззи-регулятора у вигляді змінних коефіцієнтів, щоб не змінювати в процесі роботи самі лінгвістичні правила.

Формула розрахунку вихідного значення фаззи-регулятора має такий вигляд:

$$X_S = \frac{\mu_1 \cdot X_D \cdot W_1 + \mu_2 \cdot X_E \cdot W_2 + \dots}{\mu_1 + \mu_2 + \dots}, \quad (3.12)$$

де W_1, W_2 - корегувальні коефіцієнти.

Пропонується змінювати коригувальні коефіцієнти за правилами тренування нейросіток. При застосуванні, наприклад, алгоритму навчання Відроу-Хоффа коригувальні коефіцієнти вихідного значення фаззи-регулятора розраховуються за формулою:

$$w_{i,j}^k = w_{i,j}^{k-1} + lr \cdot x_{\text{зав}} \cdot (x_{\text{ціль}} - x) \quad (3.13)$$

де $X_{\text{зав}}$ - завдання на вході системи регулювання;

$X_{\text{ціль}}$ - цільовий процес;

X - дійсний процес на виході системи (рис. 3.15).

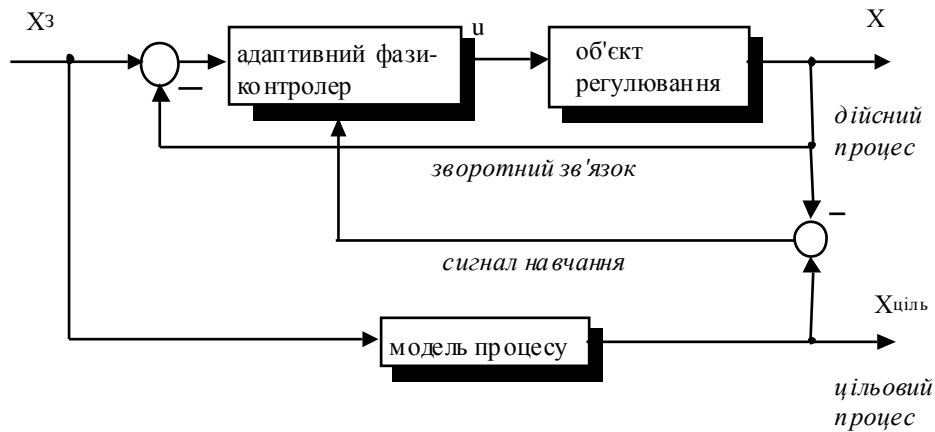
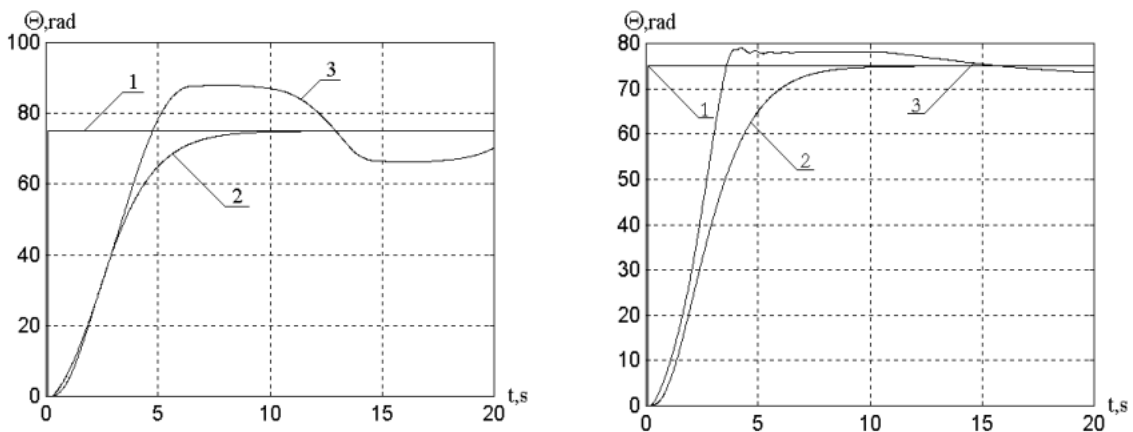


Рисунок 3.15 - Структурна схема адаптивної фазі-системи електропривода

Результати роботи адаптивної системи можна оцінити за допомогою рис. 3.16.



1 - Вхідна дія; **2** - Цільовий процес;
3 - Ненавчена система

1 - Вхідна дія; **2** - Цільовий процес;
3 - П'ята епоха тренування коефіцієнту підсилення (навчена система)

Рисунок 3.16 – Перехідні процеси в системі електроприводу з адаптивною системою регулювання

Запропонована методика самоадаптації фазі-регулятора в системах регулювання є дуже ефективною в тому плані, що дозволяє автоматично коригувати всі недоліки і вади, зроблені при проектуванні тієї чи іншої фазі-системи, не змінюючи самої фазі-системи. Слід також додати, що завдяки об'єднанню алгоритмів навчання нейросіток і фазі-логіки остання стала більш досконалою і набула суттєвого значення для вирішення широкого спектру проблем в електроприводі на рівні з іншими формами штучного інтелекту.

4 Зміст роботи

4.1 Дослідження ненавченої системи регулювання:

- ✓ запустити середовище програмування Matlab 5.2;
- ✓ ввійти до програмного пакету Simulink, набравши в командному рядку команду *simulink*;
- ✓ перейти до директорії (**x:\..\Kollarov**) командним шляхом, набравши в командному рядку:

```
>cd x:\..\Kollarov;
```

- ✓ відкрити в середовищі Matlab файл **x:\..\Kollarov\Gain.m**. Впевнитись, що перед відповідними рядками файла **Gain.m**, які наведені далі, стоять коментарі " % ":

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * u(1) * u(4);
```

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * (u(1)-w_KPI(pg,qg));
```

- ✓ ввести параметри регуляторів та всі інші дані, які знаходяться у файлі **daten.m**, в середовище Matlab 5.2;
- ✓ виконати одноразове моделювання системи і зафіксувати результати.

4.2 Дослідження правила навчання Відроу - Хоффа:

- ✓ ввести параметри регуляторів та всі інші дані, які знаходяться у файлі **daten.m**, в середовище Matlab 5.2;

- ✓ змінити рядки файла **Gain.m**

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * u(1) * u(4);
```

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * (u(1)-w_KPI(pg,qg));
```

на рядки

```
w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * u(1) * u(4);
```

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * (u(1)-w_KPI(pg,qg));
```

- ✓ виконати послідовно чотири епохи тренування системи шляхом чотирікратного запуску моделювання, зберігаючи результати після першого та четвертого моделювання;

4.3 Дослідження правила навчання Кохонена:

- ✓ ввести параметри регуляторів та всі інші дані, які знаходяться у файлі **daten.m**, в середовище Matlab 5.2;

- ✓ змінити рядки файла **Gain.m**:

```
w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * u(1) * u(4);
```

```
% w_KPI(pg,qg) = w_KPI(pg,qg) + LR_KPI * (u(1)-w_KPI(pg,qg));
```

на рядки:

$$\% \quad w_{KPI}(pg, qg) = w_{KPI}(pg, qg) + LR_{KPI} * u(1) * u(4);$$

$$w_{KPI}(pg, qg) = w_{KPI}(pg, qg) + LR_{KPI} * (u(1) - w_{KPI}(pg, qg));$$

- ✓ виконати послідовно десять епох тренування системи шляхом десятикратного запуску моделювання, зберігаючи результати після першого, четвертого та десятого моделювання.

5 Зміст звіту

В звіті повинні бути представлені:

- ✓ графік перехідного процесу в ненавченій системі, графік вхідної дії та цільового процесу на одному рисунку;
- ✓ графік перехідного процесу при першій та четвертій епохах тренування за правилом Відроу-Хоффа на одному рисунку з вхідною дією;
- ✓ графік перехідного процесу при першій, четвертій та десятій епохах тренування за правилом Кохонена на одному рисунку з вхідною дією;
- ✓ порівняльні висновки ефективності використання цих двох алгоритмів навчання нейросіток для адаптації фази-регуляторів у системах регулювання електроприводів у плані швидкості і точності адаптації.

ЛІТЕРАТУРА

1. Drechsel, D. Regelbasierte Interpolation und Fuzzy Control. - Braunschweig/Wiesbaden: Friedr. Vieweg&Sohn Verlagsgesellschaft mbH. - 1996.
2. Passino K. M., Yurkovitch. Fuzzy Control. - ADDISON-WESLEY. - 1998.
3. Vogel J. u.a. Elektrische Antriebstechnik. – Berlin: Verlag Technik GmbH. - 1991.
4. Kahlert J., Frank H. Fuzzy-Logik und Fuzzy-Control. Eine anwendungsorientierte Einführung mit Begleitsoftware. - Braunschweig/Wiesbaden: Friedr. Vieweg&Sohn Verlagsgesellschaft mbH. - 1993.
5. Калашников В.И., Палис Ф. Введение в интеллектуальные системы программного управления – Донецк: ДонГТУ. - 1997.
6. Fuzzy Logic Toolbox User's Guide: MathWorks. - 1998.
7. Brause R. Neuronale Netze: eine Einführung in die Neuroinformatik. - Stuttgart: Teubner. - 1991.
8. Kruse R., Gebhardt J., Klawonn F. Fuzzy-Systeme.- Stuttgart: Teubner. - 1993.
9. Nauk D., Klawonn F., Kruse R. Neuronale Netze und Fuzzy-Systeme, Grundlagen des Konnektionismus, Neuronaler Fuzzy-Systeme und der Kombination mit wissensbasierten Methoden. - Braunschweig: Teubner. - 1992.
10. Калашников В.И., Денисенко И.В. Теория нейросетей. - Донецк: ДонГТУ. - 1997.