

Потап О. Ю., Маслікова С. С., Кисляков В. Г.

СИСТЕМИ УПРАВЛІННЯ ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ

НЕЙРОМЕРЕЖНІ СИСТЕМИ УПРАВЛІННЯ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
УКРАЇНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ НАУКИ І ТЕХНОЛОГІЙ

О. Ю. Потап, С. С. Маслікова, В. Г. Кисляков

Системи управління зі штучним інтелектом.
Нейромережні системи управління



НАВЧАЛЬНИЙ ПОСІБНИК

ДНПРО
2024

УДК 004.8:681.5(075.8)

П 64

Авторський колектив:

Потап О. Ю., Маслікова С. С., Кисляков В. Г.

Рекомендовано Радою якості освітньої діяльності УДУНТ

Протокол № 6 від 20 лютого 2024 р.

П 64 **Потап, О. Ю.** Системи управління зі штучним інтелектом. Нейромережні системи управління : навч. посіб. / О. Ю. Потап, С. С. Маслікова, В. Г. Кисляков ; за ред. канд. техн. наук, проф. О. Ю. Потапа ; Укр. держ. ун-т науки і технологій. – Електрон. вид – Дніпро : УДУНТ, 2024. – 106 с.

У навчальному посібнику розглянуті загальні положення штучних нейронних мереж та їх використання в системах автоматичного управління. Розглянуто задачі ідентифікації статичних та динамічних об'єктів, стратегії нейромережного інверсного управління, ефективність застосування нейрорегуляторів у порівнянні з класичними ПД-регуляторами.

Призначений для опанування освітньої компоненти «Системи управління зі штучним інтелектом» за спеціальністю 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» для ОПП «Автоматизація та комп'ютерно-інтегровані технології» другого (магістерського) рівня вищої освіти.

УДК 004.8:681.5(075.8)

ЗМІСТ

ВСТУП	4
1. ОСНОВИ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	6
1.1. Біологічний нейрон як структурний елемент людського мозку.....	6
1.2. Штучні нейронні мережі – спрощена модель людського мозку.....	9
1.3. Модель нейрона	11
1.4. Архітектура нейронних мереж	15
1.4.1. Нейронні мережі прямого розповсюдження	16
1.4.2. Багатошарові мережі прямого розповсюдження	16
1.4.3. Рекурентні мережі	18
1.4.4. Повністю зв'язана мережа Хопфілда	19
1.5. Навчання штучних нейронних мереж	20
1.5.1. Пряме контрольоване навчання	21
1.5.2. Стимульоване навчання	22
1.5.3. Алгоритм зворотного розповсюдження помилки	24
1.5.4. Властивість узагальнення	32
1.5.5. Деякі особливості підготовки даних для навчання мережі	34
1.6. Радіально-базисні нейронні мережі	36
2. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В СИСТЕМАХ УПРАВЛІННЯ	43
2.1. Пряме інверсне управління	44
2.2. Пряме адитивне управління	46
2.3. Управління зі внутрішньою моделлю	47
3. ПРАКТИКУМ «НЕЙРОННІ МЕРЕЖІ В АВТОМАТИЗАЦІЇ»	50
3.1. Ідентифікація статичних лінійних об'єктів за допомогою нейронних мереж	50
3.2. Ідентифікація статичних нелінійних об'єктів за допомогою нейронних мереж	59
3.3. Ідентифікація динамічних об'єктів за допомогою нейронних мереж ...	69
3.4. Дослідження роботи АСР з нейрорегуляторами	80
БІБЛІОГРАФИЧНИЙ СПИСОК	95
ДОДАТОК. Традиційні методи автоматичного управління станом динамічних систем (короткий огляд)	96

ВСТУП

Застосування штучних нейронних мереж для вирішення задач управління різноманітними технічними та технологічними об'єктами стає наразі найбільш перспективним напрямком розвитку сучасних систем автоматизації. Разом з нечіткою логікою (fuzzy-logic), генетичними алгоритмами та низкою інших інформаційних технологій вони утворюють основу штучного інтелекту.

Серед задач, що вирішують штучні нейронні мережі, можна назвати задачі:

- 1) класифікації взірців – визначення приналежності взірця до одного або кількох заздалегідь визначених класів;
 - 2) кластеризацію – розподілення взірців на заздалегідь невизначені класи за якими-небудь ознаками;
 - 3) апроксимацію функцій – оцінювання невідомої залежності за експериментальними даними;
 - 4) прогнозування – визначення майбутнього стану процесу за його минулим станом;
 - 5) оптимізація – винайдення рішень, що максимізують або мінімізують певний критерій якості в умовах певних обмежень;
 - 6) управління – переведення та підтримання системи у заданому стані.
- З наведених задач задача управління є найбільш складною.

Цілком природно, що на сучасному етапі підготовка магістрів з автоматизації та комп'ютерних технологій передбачає набуття здобувачами вищої освіти професійних компетентностей як у галузі штучного інтелекту в цілому, так, зокрема, і штучних нейронних мереж.

Даний навчальний посібник призначений для студентів, які здобувають ступінь магістра в Українському державному університеті науки і технологій за спеціальністю 174 – автоматизація, комп'ютерно-інтегровані технології та робототехніка на освітній програмі «Автоматизація, комп'ютерно-інтегровані технології». Матеріал посібника охоплює відповідні розділи обов'язкової для вивчення професійної навчальної дисципліни «Системи управління зі штучним інтелектом», забезпечуючи, зокрема, формування таких визначених стандартом вищої освіти програмних компетентностей, як:

- здатність до проведення досліджень на відповідному рівні;
- здатність до абстрактного мислення, аналізу та синтезу;
- здатність здійснювати автоматизацію складних технологічних об'єктів та комплексів, створювати кіберфізичні системи на основі інтелектуальних методів управління та цифрових технологій з використанням баз даних, баз знань, методів штучного інтелекту, робототехнічних та інтелектуальних мехатронних пристроїв;

- здатність застосовувати методи моделювання та оптимізації для дослідження та підвищення ефективності систем і процесів керування складними технологічними та організаційно-технічними об'єктами;

- здатність застосовувати сучасні методи теорії автоматичного керування для розроблення автоматизованих систем управління технологічними процесами та об'єктами;

- здатність застосовувати спеціалізоване програмне забезпечення та цифрові технології для розв'язання складних задач і проблем автоматизації та комп'ютерно-інтегрованих технологій.

Відповідно до робочої програми засвоєння навчального матеріалу посібника забезпечує досягнення здобувачами вищої освіти таких очікуваних програмних результатів навчання:

- 1) уміння обґрунтовувати архітектуру штучних нейронних мереж, проводити їх навчання та здійснювати із їх допомогою ідентифікацію статичних та динамічних об'єктів;

- 2) уміння синтезувати системи автоматичного управління на базі нейрорегуляторів, моделювати та досліджувати їхню роботу з використанням спеціалізованих програмних засобів.

Навчальний посібник містить загальні положення штучних нейронних мереж та їх використання в системах автоматичного управління, зокрема у задачах ідентифікації статичних та динамічних об'єктів, нейромережного інверсного управління, визначення ефективності застосування нейрорегуляторів у порівнянні з класичними ПД-регуляторами.

Навчальний матеріал посібника міститься у трьох розділах.

У розділі 1 розглянуто найважливіші поняття нейронних мереж та методів машинного навчання, зокрема навчання перцептрона та нейронних мереж зі зворотним поширенням помилки. Окрему увагу приділено застосуванню радіально-базисних нейронних мереж для вирішення задач апроксимації функцій.

У розділі 2 описано найбільш відомі варіанти побудови нейромережних систем управління.

Розділ 3 містить 4 практичних роботи з дослідження особливостей ідентифікації статичних та динамічних об'єктів управління за допомогою нейронних мереж різної архітектури, а також ефективності різних схем застосування нейронних регуляторів для управління лінійними об'єктами із запізненням.

Викладення матеріалу супроводжується значною кількістю прикладів, що полегшує його сприйняття та засвоєння. Подано перелік запитань для самоконтролю.

1. ОСНОВИ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

1.1. Біологічний нейрон як структурний елемент людського мозку

Теорія штучних нейронних мереж є досить перспективним напрямком науки, що дуже активно розвивається Її виникнення пов'язане із дослідженнями в галузі штучного інтелекту, а саме зі спробами моделювання роботи людського мозку.

Сьогодні, як і сто років тому, не підлягає сумніву, що мозок працює більш ефективно й принципово іншим чином, ніж будь-яка створена людиною обчислювальна машина. Саме цей факт протягом кількох останніх десятиліть спонукує й направляє роботи вчених усього світу щодо створення й дослідження штучних нейронних мереж.

Питання, чи є живий мозок чимось на кшталт машини, вперше було порушене французьким філософом й математиком Рене Декартом (1596 - 1650), який жив у часи підйому механіки. За Декартом, усі дії, як у людини, так і у тварин, є відповіддю на події зовнішнього світу. Зовнішній подразник збуджує один з органів почуттів, збудження передається до мозку, який спрямовує його до відповідних м'язів. Таким чином, збудження органів почуттів призводить до скорочення м'язів, тобто до рефлекторного реагування на зовнішній подразник. Саме Декарт вперше увів до обігу термін «рефлекс».

Потужний поштовх розвитку теорії рефлексів надали роботи нейрофізіолога І. П. Павлова. Проте, до початку ХХ століття більшість нейрофізіологів вважали, що маршрут рефлексу пролягає крізь безперервні нитки нервової тканини. Поняття *синапсу*, щілини між нейронами, за допомогою якої вони мають взаємодіяти, є порівняно новим. Дослідження синапсу та його ролі у нервовій активності були здійснені на зламі віків англійцем Чарльзом Шеррінгтоном (1857 - 1952). Й, хоча робота Шаррінгтона зачипала не електрофізіологічний, а поведінковий рівень, з аналізу рефлекторних дій собак та мавп він зумів розгадати основні принципи роботи синапсу [1].

До перших спроб розкрити секрет реалізації роботи мозку можна віднести роботу Рамон-і-Кахаля (1911), у якій була висловлена ідея про

нейрон як структурну одиницю мозку. Застосувавши метод забарвлення нейронів солями срібла (срібло вибірково проникає до нейронів, але не просочується до інших клітин мозку), Кахаль побачив, що мозок має клітинну архітектуру. Крім нейронів до складу мозку входять різноманітні гліальні клітини, що виконують опорні функції. Кахаль описав нейрони як поляризовані клітини, які отримують сигнали сильно розгалуженими відростками, що отримали назву **дендритів**, а посилає інформацію нерозгалуженими відростками, що були названі **аксонами** (див. рис. 1.1).

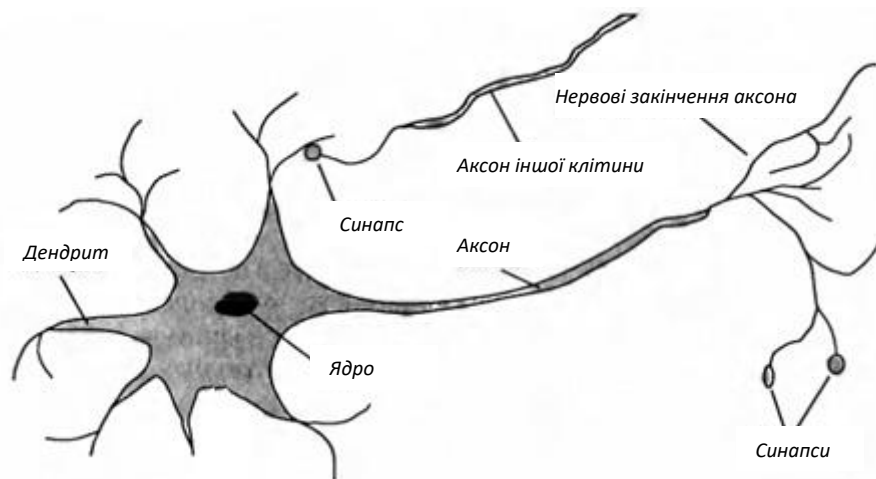


Рис. 1.1. Біологічний нейрон

В подальшому з'ясувалось, що, незважаючи на виявлену велику різноманітність нейронів, усі вони проводять інформацію однаково. Інформація передається аксонами у вигляді коротких електричних імпульсів, так званих потенціалів дії, амплітуда яких становить близько 100 мВ, а тривалість — 1мс. Виникнення імпульсів пов'язують з рухом позитивно заряджених іонів натрію крізь поверхневу клітинну мембрану з позаклітинної рідини всередину клітини, в її цитоплазму [1, 2].

Концентрація натрію в міжклітинному просторі майже в 10 разів більша за його внутрішньоклітинну концентрацію. У стані спокою підтримується трансмембранна різниця потенціалів близько – 70 мВ. При цьому іони натрію проникають всередину клітини повільно, оскільки доступ туди для них обмежений властивостями мембрани. Фізична або хімічна стимуляція деполяризує мембрану, збільшуючи її проникність для іонів

натрію. Потік натрію всередину клітини ще сильніше де поляризує мембрану, роблячи її все більш проникною.

Коли досягається певне критичне (порогове) значення потенціалу, відбувається регенеративний зсув, внаслідок якого знак різниці потенціалів змінюється на протилежний, тобто внутрішній вміст клітини стає позитивно зарядженим відносно зовнішньої середовища. Приблизно через 1 мс проникність мембрани для натрію падає й трансмембранний потенціал повертається до свого значення у стані спокою – 70 мВ. Після кожного такого «вибуху» впродовж кількох мілісекунд проникність мембрани не може змінюватись. Це обмежує частоту генерації імпульсів – не більше 200 разів на секунду.

Хоча аксони й схожі на електричні дроти, імпульси вони проводять інакше. Їх «кабельні характеристики» погані: електричний опір вздовж вісі занадто великий, а мембранний опір занадто малий. Позитивний заряд розсіюється вже за 1 - 2 мм. Щоб подолати відстань, яка іноді становить кілька сантиметрів, імпульси мають регенеруватись, тобто повторно підсилюватись, що обмежує максимальну швидкість розповсюдження нервового імпульсу вздовж аксону до 100 м/с.

Зв'язки між нейронами опосередковуються хімічними передавачами – нейромедіаторами, які виділяються у синапсах. Молекули нейромедіатора змінюють проникність мембрани. Ефект від їх дії може бути як збуджуючим, якщо зміна заряду наближає мембранний потенціал до порогу генерації імпульсу, так і гальмівним, якщо мембрана стабілізуватиметься на рівні потенціалу спокою. Отже, у синапсах сигнал аксона може підсилюватись або послаблятися.

Кожний синапс дає лише незначний ефект на активність аксону. Для переходу нейрона в активний стан, тобто для появи на його виході електричного імпульсу, кожний нейрон має безперервно інтегрувати до 1000 синаптичних входів.

Отже дендрити отримують сигнали від аксонів інших нейронів і передають їх до тіла клітини. Якщо сума отриманих сигналів більше деякого граничного значення, нейрон збуджується й генерує електричний імпульс, який передається аксоном до інших нейронів. Таким чином, біологічний нейрон здійснює приймання, елементарне перетворення й передачу інформації іншим нейронам.

Ще на початку ХХ століття нейрофізіологам стала зрозумілою виключно важлива роль синапсів у навчанні. Сигнали мозку, проходячи крізь

них, можуть підсилюватись і послаблятись. Звертає на себе увагу й такий факт: мозок новонародженого й мозок дорослої людини містять приблизно однакову кількість нейронів, але тільки мозок дорослого відрізняється упорядкованістю міжнейронних синапатичних зв'язків. Мабуть, навчання мозку й полягає у зміні архітектури нейронної мережі, яке супроводжується налаштуванням синапсів.

Однак біологічний нейрон має на 5-6 порядків меншу швидкість спрацьовування, ніж напівпровідниковий логічний елемент. Як показали подальші дослідження, секрет високої продуктивності мозку полягає у величезній кількості нейронів і взаємозв'язків між ними.

Мережа нейронів, що утворює людський мозок, являє собою високоефективну, комплексну, нелінійну, істотно паралельну систему обробки інформації. Вона здатна організувати свої нейрони таким чином, щоб реалізувати сприйняття образу, його розпізнання або управління рухом, у багато разів швидше, ніж ці задачі можуть бути вирішені найсучаснішими комп'ютерами.

1.2. Штучні нейронні мережі – спрощена модель людського мозку

Штучна нейронна мережа є спрощеною моделлю мозку. У літературі не існує для неї універсального визначення. Однак найчастіше під *штучною нейронною мережею* розуміють сукупність простих елементів – *нейронів*, з'єднаних між собою за допомогою зважених зв'язків.

Штучна нейронна мережа — це істотно паралельно розподілений процесор, що має природну схильність до збереження набутого знання й можливість надання його нам.

Привабливою властивістю штучних нейронних мереж є їхня здатність до *навчання*, тобто вони можуть навчатися рішення певної задачі. Інформація про те, як саме повинне вирішуватись завдання, зберігається у вагових коефіцієнтах (вагах) мережі, які настраюються в процесі навчання відповідно до певного алгоритму навчання.

Процедура, що використовується для здійснення процесу навчання, називається алгоритмом навчання. Її функція полягає в модифікації синапатичних ваг мережі певним чином так, щоб вона набула необхідні властивості.

Зі сказаного вище стає ясно, що нейронні мережі реалізують свою обчислювальну потужність, завдяки двом основним своїм властивостям: істотно паралельно розподіленій структурі та здатності навчатися й узагальнювати отримані знання. Під властивістю узагальнення розуміється здатність нейронних мереж генерувати правильні виходи для вхідних сигналів, які не були враховані в процесі навчання (тренування). Ці дві властивості роблять нейронні мережі системою переробки інформації, що вирішує складні багатомірні задачі, непосильні іншій техніці.

Зауважимо, що нейронні мережі можуть бути як програмного, так і апаратного виконання.

Дослідження роботи нейронних мереж, як правило, виконується шляхом програмного моделювання на звичайних комп'ютерах. Для цього використовують спеціально розроблені нейропакети, які емулюють середовище нейрокомп'ютера. Одним з найбільш відомих серед них є пакет прикладних програм Neural Network Toolbox для системи MATLAB. Він досить докладно описаний у [3, 4, 5].

Актуальність досліджень штучних нейронних мереж підтверджується безліччю їхніх застосувань у найрізноманітніших галузях людської діяльності – техніці, медицині, бізнесі. Нейронні мережі являють собою потужний апарат для вирішення таких задач, як ідентифікація, управління, розпізнавання образів, апроксимація функцій, прогнозування й оптимізація.

Основні принципи штучних нейронних мереж були закладені ще в 1940-х роках, а їх перші програмні й апаратні реалізації були виконані в 1950 - 1960-і роки.

Після того, як наприкінці 1960-х років були виявлені істотні обмеження ранніх моделей нейронних мереж (одношарових персептронів), дослідження в галузі штучних нейронних мереж практично припинилися.

Інтерес до штучних нейронних мереж відродився лише на початку 1980-х років частково завдяки розробці нових потужних нейромережних моделей (моделі багатошарового персептрона й рекурентної моделі Хопфілда) та методу навчання: зворотного розповсюдження помилки; а частково завдяки досягненням в галузі апаратних засобів, які дозволили реалізовувати нейронні мережі з дуже великою кількістю нейронів.

У промисловості нейронні мережі стали активно застосовуватися тільки наприкінці 1980-х років.

1.3. Модель нейрона

Штучний нейрон (або просто нейрон) є елементарним функціональним модулем, з певної кількості яких будуються штучні нейронні мережі. Він являє собою модель живого нейрона, однак лише в сенсі здійснюваних їм перетворень, а не способу функціонування. Існують логічні, безперервні й імпульсні моделі нейрона. Логічні моделі нейрона активно досліджувалися в 60-70-х роках, але не отримали подальшого розвитку. Імпульсні моделі є більш близькими до фізичної природи процесів, що відбуваються в нервовій клітині, однак їхня теорія не так розвинена як для безперервних, і вони усе ще не знаходять широкого застосування.

Безперервна модель нейрона працює у такий спосіб. Вхідні сигнали надходять на блоки, що реалізують функцію синапсів (рис.1.2). Кожний з них характеризується своїм ваговим коефіцієнтом (синаптичною вагою). Позитивні значення ваги w_{ij} відповідають збуджувальним синапсам, негативні – гальмівним. Зважені вхідні сигнали подаються на лінійний суматор, після чого результат їхнього додавання надходить на блок активаційної функції. Зазвичай активаційна функція обмежує вихідний сигнал нейрона в діапазоні $[0,1]$ або $[-1,1]$. Модель нейрона також містить у собі зсув b , що додається до вхідного сигналу блоку активаційної функції.

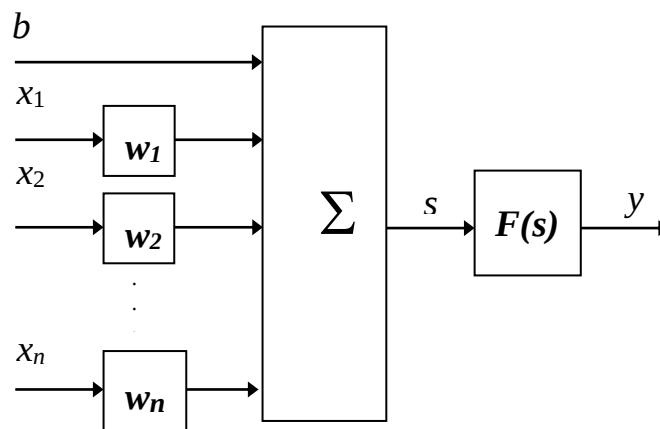


Рис. 1.2. Модель нейрона

Математична модель нейрона описується наступними залежностями:

$$s = \sum_i^n x_i w_i + b, \quad (1.1)$$

$$y = F(s), \quad (1.2)$$

де $x_1, x_2, \dots, x_n$ – вхідні сигнали нейрона, $w_1, w_2, \dots, w_n$ – синаптичні ваги, s – комбінований вхід (ввід) нейрона, b – зсув, $F(s)$ – функція активації, y – вихідний сигнал нейрона.

Цей запис можна спростити, увівши додатковий вхідний сигнал $x_0=1$ та вагу $w_0=b_0$. Тоді (1.1) матиме вигляд:

$$s = \sum_{i=0}^n x_i w_i \quad (1.3)$$

Функція активації нейрона $F(s)$ визначає нелінійне перетворення, яке здійснюється нейроном. Існує багато видів функцій активації, але найбільш поширені наступні чотири:

1. Гранична функція (рис. 1.3,а):

$$F(s) = \begin{cases} 1, & \text{якщо } s \geq 0 \\ 0, & \text{якщо } s < 0 \end{cases} \quad (1.4)$$

2. Кусочно-лінійна функція (рис. 1.3,б):

$$F(s) = \begin{cases} 1, & \text{якщо } s \geq a \\ s, & \text{якщо } a > s > -a \\ 0, & \text{якщо } s \leq -a \end{cases} \quad (1.5)$$

У наведеному випадку $a=1$, і коефіцієнт нахилу лінійної ділянки обраний одиничним, а вся функція може інтерпретуватися як апроксимація нелінійного підсилювача. При нескінченно великому коефіцієнті нахилу лінійної ділянки функція вироджується в граничну.

У більшості типів нейронних мереж використовуються нейрони з лінійною функцією активації $F(s)=s$, що представляє собою окремий випадок (1.5) з необмеженою лінійною ділянкою.

3. Сигмоїдальна функція (рис. 1.3,в). Це найбільш широко розповсюджений тип функції активації. Вона була уведена за аналогією із граничною функцією, але на відміну від неї є монотонно зростаючою, неперервною й диференційованою. Диференційованість є дуже важливою властивістю для аналізу мереж і деяких методів їхнього навчання.

У загальному вигляді сигмоїдальна активаційна функція описується залежністю:

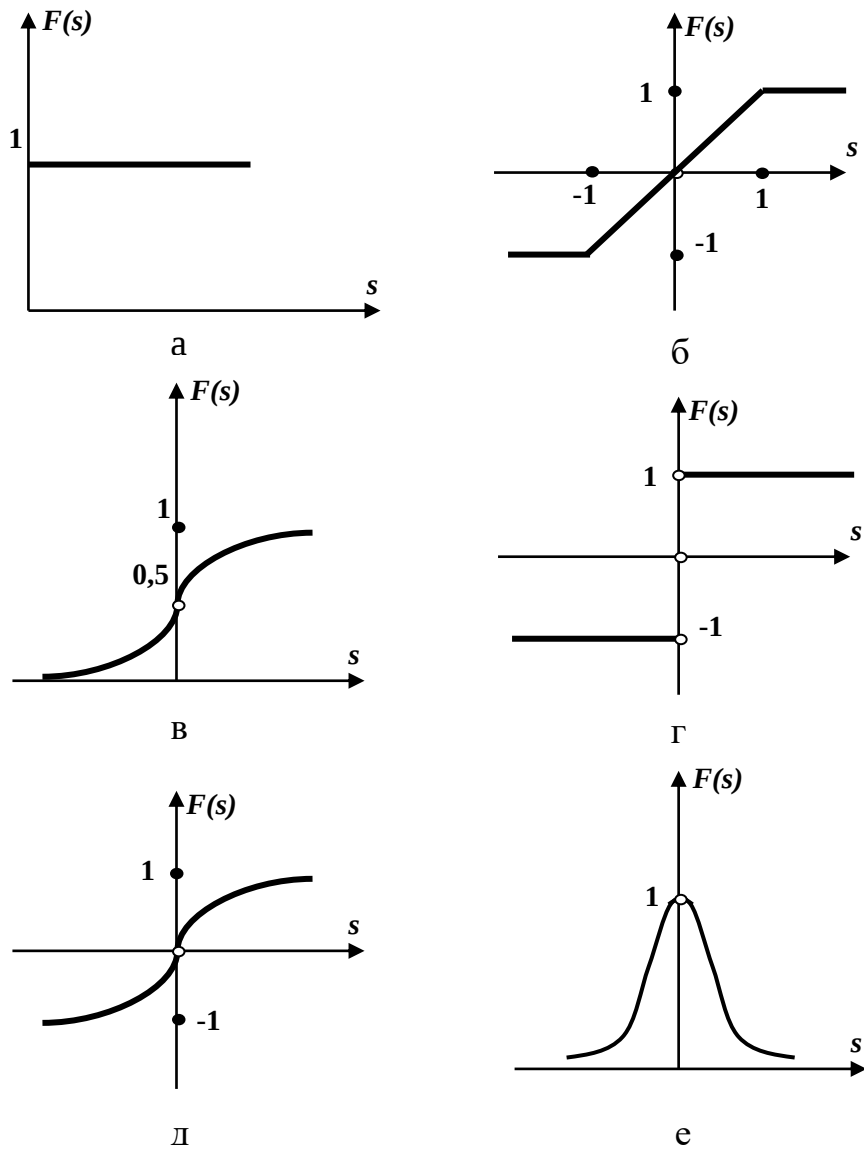


Рис. 1.3. Функції активації:
а), г) граничні; б) лінійна; в) сигмоїдальна; д) тангенціальна;
е) радіально-базисна

$$F(s) = \frac{1}{1 + e^{-as}}. \quad (1.6)$$

де a – параметр, що визначає нахил функції. Найчастіше використовується $a=1$. У випадку нескінченно великого a сигмоїдальна функція вироджується у граничну.

Крім перерахованих функцій, що змінюються в діапазоні $[0,1]$, існують також їхні аналоги з областю значень $[-1,1]$. Так наприклад (див. рис. 1.3,г), гранична функція може бути визначена як

$$F(s) = \begin{cases} 1, & \text{якщо } s > 0 \\ s, & \text{якщо } s = 0 \\ -1, & \text{якщо } s < 0 \end{cases}. \quad (1.7)$$

$$F(s) = th(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$$

Тобто

$$F(s) = sign(s).$$

Замість сигмоїдальної функції активації широко застосовується гіперболічний тангенс, що має аналогічні властивості (рис. 1.3,д)

$$F(s) = th(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}} \quad (1.8)$$

Непарність цієї функції робить її зручною для вирішення завдань управління.

4. У деяких нейронних мережах в якості активаційної застосовується функція Гауса (рис. 1.3,е)

$$F(s) = e^{-\frac{s^2}{\sigma^2}}. \quad (1.9)$$

Однак на відміну від (1.3) її аргумент обчислюється за формулою:

$$s = \|X - C\|, \quad (1.10)$$

де X – вектор вхідних сигналів нейрона, C – вектор координат центра вікна активаційної функції, σ – ширина вікна, норма $\|\dots\|$ – евклідова відстань. У теорії нейронних мереж активаційні функції цього типу називаються радіально-базисними функціями (РБФ), а засновані на них мережі – РБФ-мережами (*RBF – radial basis function*).

1.4. Архітектура нейронних мереж

У загальному випадку поняття «штучна нейронна мережа» охоплює ансамблі нейронів будь-якої структури, однак практичне застосування знайшли тільки деякі з них. Це пояснюється тим, що архітектура нейронної мережі безпосередньо пов'язана з методом її навчання. На різних етапах розвитку нейронних мереж з'являлись нові типи архітектури та спеціально опрацьовані для них методи навчання. Взагалі ж архітектура нейронної мережі визначається, як правило, особливостями та складністю задачі, що вирішується.

Зазвичай виділяють два класи нейронних мереж: нейронні мережі *прямого розповсюдження* (*feed-forward neural networks, FNN*) та *рекурентні* нейронні мережі (*recurrent neural networks, RNN*).

До нейронних мереж прямого розповсюдження відносяться: одношарова нейрона мережа (чи одношаровий персептрон), багатошарова нейрона мережа (або багатошаровий персептрон) та радіальна базисна нейронна мережа.

Рекурентні нейронні мережі, на відміну від мереж прямого розповсюдження, мають хоча б один зворотний зв'язок. Наприклад, мережа Хопфілда, мережа Елмана тощо.

В літературі наводяться такі орієнтовні дані щодо використання різних типів нейронних мереж: багатошаровий персептрон – 81,2%, мережа Кохонена – 8,3 %, мережа Хопфілда – 5,4 %, інші – 5,1 %.

Далі, як приклад, розглянемо декілька нейронних мереж з найбільш розповсюдженими різновидами архітектури.

1.4.1. Нейронні мережі прямого розповсюдження

Шаровою називається нейронна мережа, що складається із груп нейронів, розділених на шари. Мережа, що містить k шарів, називається k -шаровою. Якщо сигнали передаються мережею тільки в напрямку від попереднього шару до наступного, така мережа називається мережею прямого розповсюдження.

На рис. 1.4 зображена одношарова мережа прямого розповсюдження, яка містить у собі шар вхідних нейронів та шар нейронів на виході. Нейрони вхідного шару просто ретранслюють сигнали до нейронів на виході, не перетворюючи їх, тому вхідний шар при визначенні типу мережі не враховується і наведена мережа називається одношаровою. У шарі нейронів на виході відбувається перетворення сигналів і формування реакції мережі. Необхідно відзначити, що нейрони також іноді називаються вузлами, елементами або обчислювальними модулями. Кількість нейронів в одному шарі визначає розмір шару.

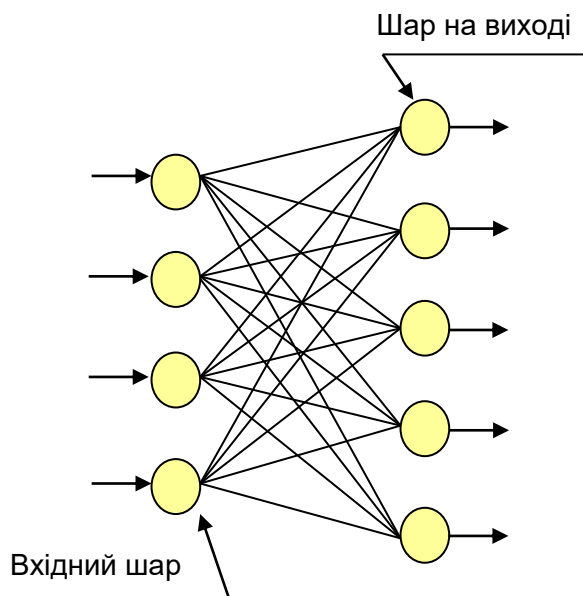


Рис. 1.4. Одношарова мережа прямого розповсюдження

1.4.2. Багатшарові мережі прямого розповсюдження

Такі мережі характеризуються наявністю одного або декількох прихованих шарів, які здійснюють перетворення інформації. Нейрони прихованого шару називаються прихованими нейронами або прихованими

вузлами. Використання прихованих шарів дозволяє мережі здійснювати нелінійні перетворення вхід-вихід будь-якої складності. Ці унікальні властивості багатошарових мереж особливо проявляються при високій розмірності простору вхідних сигналів.

На рис. 1.5 наведена схема двошарової мережі прямого розповсюдження з одним прихованим шаром. Для опису такої мережі використовується запис $NN\ 4-5-2$. Тут 4 – розмір вхідного шару мережі, 5 – прихованого шару, 2 – шару на виході. У загальному випадку мережа прямого розповсюдження із g вхідними нейронами, q вихідними нейронами та n прихованими шарами розміру h_i позначається $NN\ g-h_1-h_2-...-h_n-q$.

Нейрони вхідного шару у таких мережах просто ретранслюють вхідні сигнали на перший прихований шар. У прихованих нейронах послідовно, шар за шаром, відбувається нелінійне перетворення сигналів. Сигнали з останнього прихованого шару надходять на нейрони вихідного шару, які формують реакцію мережі.

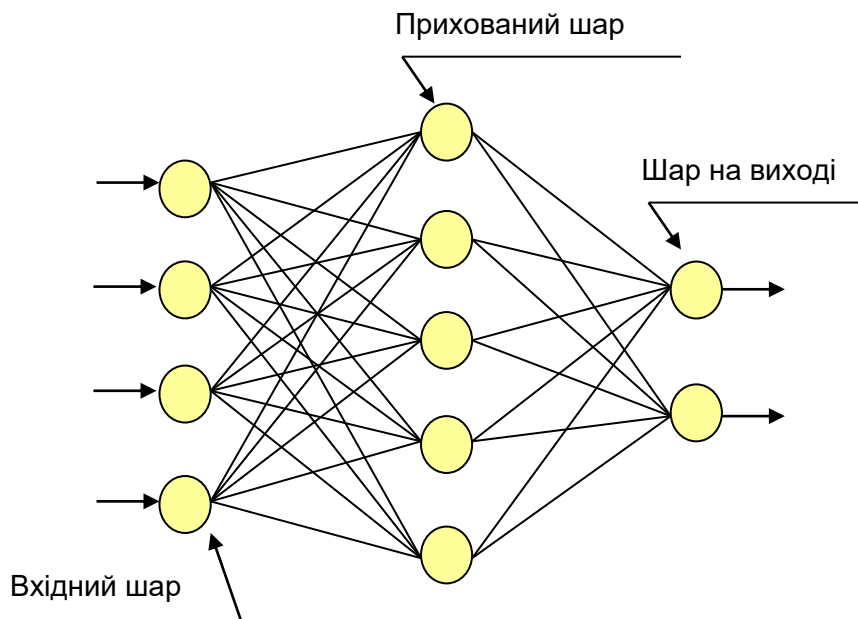


Рис. 1.5. Повністю зв'язана двошарова мережа прямого розповсюдження

Формально, не існує обмежень на типи активаційних функцій нейронів різних шарів мережі або навіть одного шару, однак зазвичай усі приховані нейрони обираються одного типу. Шар на виході мережі може складатися з нейронів з таким самим типом активаційної функції, що й у нейронів прихованого шару, але найпоширенішою є модель мережі з лінійними нейронами на виході. Нейронні мережі цього типу з активаційними

функціями нейронів прихованого шару (1.6) або (1.8) називаються багатошаровими персептронами. Багатошарові персептрони знайшли широке застосування при вирішенні різноманітних задач і є одним з головних об'єктів теоретичних досліджень.

Зображена на рис. 1.5 мережа називається повністю зв'язаною мережею прямого розповсюдження, тому що кожен нейрон попереднього шару зв'язаний з усіма нейронами наступного шару. Застосовуються також частково зв'язані мережі прямого розповсюдження, у яких нейрони одного шару зв'язані тільки з певною частиною нейронів наступного шару. Така архітектура дозволяє закласти в мережу апріорні знання про бажаний закон обробки сигналів.

Двошарові мережі прямого розповсюдження широко використовуються для вирішення задач класифікації, розпізнавання образів, ідентифікації та управління.

1.4.3. Рекурентні мережі

Рекурентні мережі відрізняються існуванням зворотних зв'язків та елементів часової затримки сигналу.

На рис.1.6 наведена рекурентна мережа Джордана, що містить прихований шар нейронів. У цьому випадку кожен нейрон прихованого шару отримує, крім вхідних сигналів, ще й всі сигнали з виходу мережі.

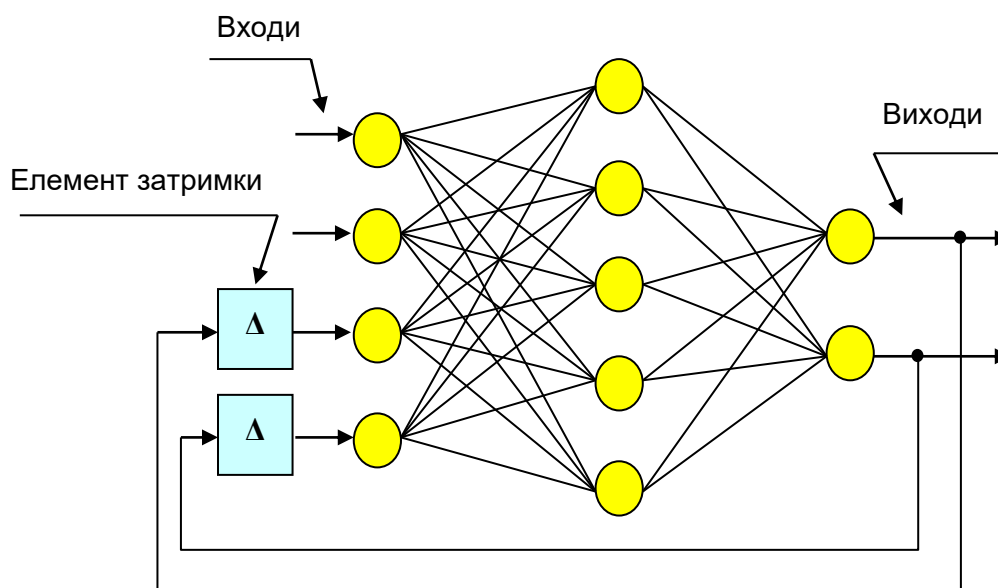


Рис. 1.6. Рекурентна мережа Джордана з одним прихованим шаром

В нейронній мережі Елмана (рис. 1.7) застосовується такий самий підхід, проте рекурентні зв'язки зі затримкою на декілька тактів подаються не з вихідних нейронів, а з прихованого шару багатошарового персептрону.

Наявність зворотних зв'язків та елементів часової затримки сигналів надає рекурентним мережам власні нелінійні динамічні властивості. Це також впливає на їхню здатність до навчання. Тренування рекурентних мереж вимагає урахування їх динамічних властивостей.

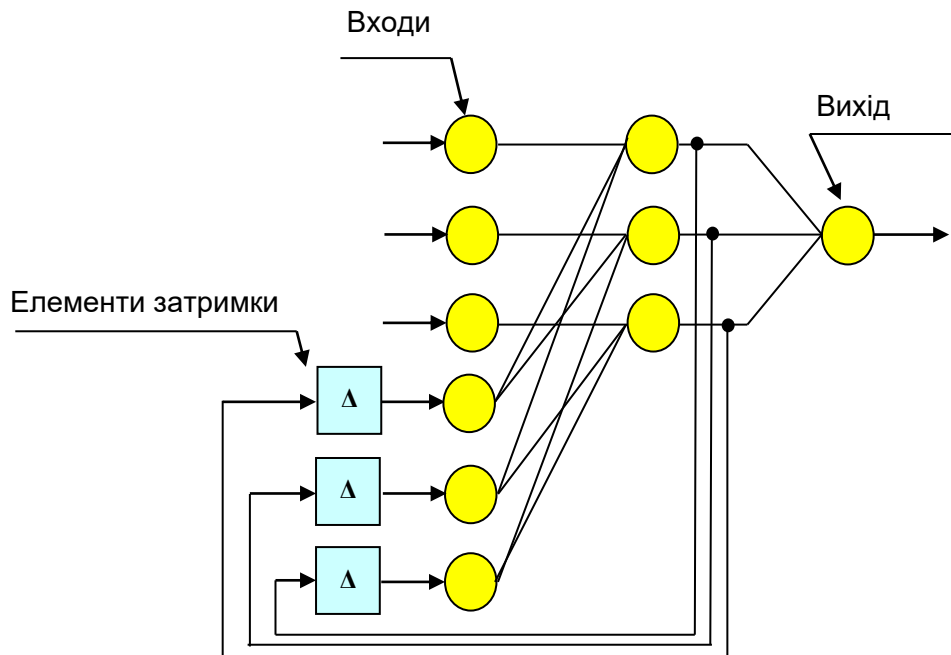


Рис. 1.7. Рекурентна мережа Елмана

Одним з головних застосувань рекурентних нейронних мереж є нейроемулятори динамічних об'єктів, тобто їх нейромережні моделі. Такі мережі можуть також використовуватись для вирішення задач апроксимації часових послідовностей, класифікації, розпізнавання образів та управління.

1.4.4. Повністю зв'язана мережа Хопфілда

У 1982 Джон Хопфілд запропонував цікавий варіант одношарової нейромережі, який дозволяв досить просто вирішувати деякі завдання оптимізації або надавати механізм автоасоціативної пам'яті. У цій мережі (рис. 1.8) є тільки вхідні нейрони, кожен з яких з'єднаний з усіма іншими (тобто мережа є повнозв'язаним графом).

Тут кожен нейрон має двосторонні зв'язки з усіма іншими нейронами мережі. У загальному випадку мережа Хопфілда має симетричну кільцеву

структуру, у ній не можна виділити приховані нейрони і єдиний напрямок розповсюдження сигналів. Робота повністю зв'язаної мережі і обмін даними в ній контролюється одним головним нейроном.

Мережа Хопфілда має властивості динамічної мережі, що ґрунтується на принципах самоорганізації, хоча у ній і не використовуються в явному вигляді елементи часової затримки.

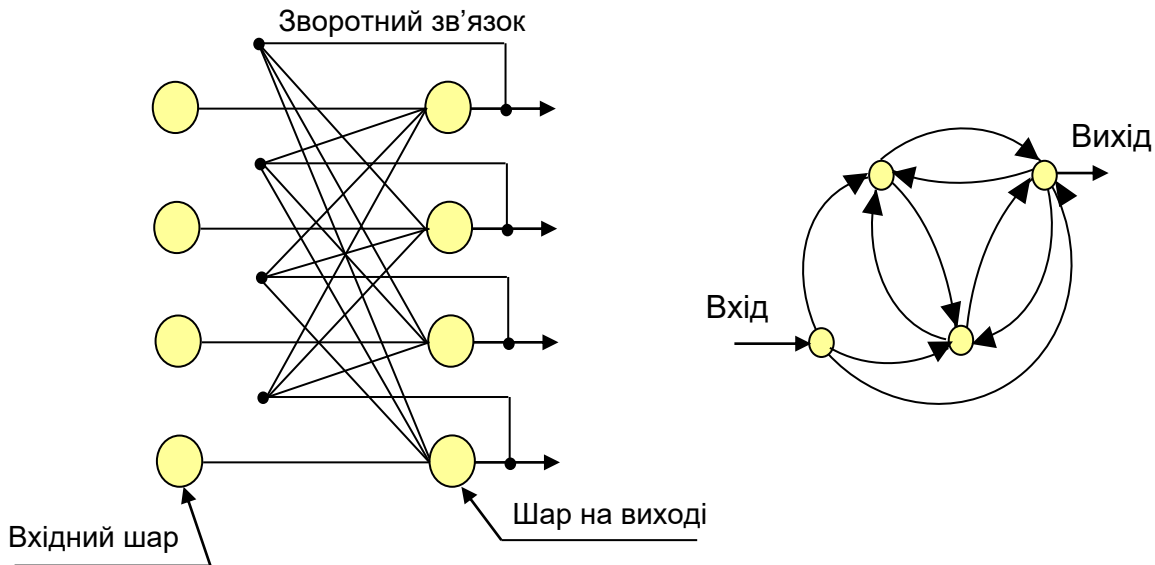


Рис. 1.8. Повністю зв'язана мережа Хопфілда

1.5. Навчання штучних нейронних мереж. Алгоритм зворотного розповсюдження помилки

Визначальною особливістю штучних нейронних мереж є їхня спроможність навчатися, тобто покращувати свою роботу під впливом оточуючого середовища, яке змінює їхні параметри. Існує багато визначень терміну «навчання» стосовно нейронних мереж, проте, найбільш підходящим можна вважати наступне:

Навчання штучної нейронної мережі – це цілеспрямований процес змінення ваги міжшарових синаптичних зв'язків, який ітеративно повторюється, доки мережа не набуде необхідних властивостей.

В літературі крім терміну «навчання» використовують також рівноправні поняття «тренування мережі» та «налаштування параметрів мережі».

Можна виділити два види навчання: **контрольоване навчання** (*supervised learning*) та **самонавчання** (*self-organized learning*). Перший вид має на увазі наявність «вчителя», який спостерігає реакцію мережі й направляє змінення її параметрів. В другому випадку мережа самоорганізується під впливом зовнішнього середовища та вивчає його самостійно без допомоги «вчителя». Самонавчання властиве задачам розпізнавання образів та класифікації. При вирішенні задач управління зазвичай використовують контрольоване навчання нейронних мереж.

Контрольоване навчання, у свою чергу, має два різновиди: **пряме контрольоване навчання** та **ситмльоване навчання** (*reinforcement learning*). Оскільки перший різновид з'явився раніш за другий і є більш розповсюдженим, на нього зазвичай посилаються просто як на контрольоване навчання.

1.5.1. Пряме контрольоване навчання

На рис. 1.9 наведена схема прямого контрольованого навчання. Присутній на ній «вчитель» володіє знаннями про зовнішнє середовище, які представлена у вигляді набору взірців «вхід-вихід». Пара таких взірців називається тренувальним шаблоном, що містить, відповідно, шаблон входів та шаблон виходів. Сукупність цих пар даних називається тренувальним набором шаблонів.

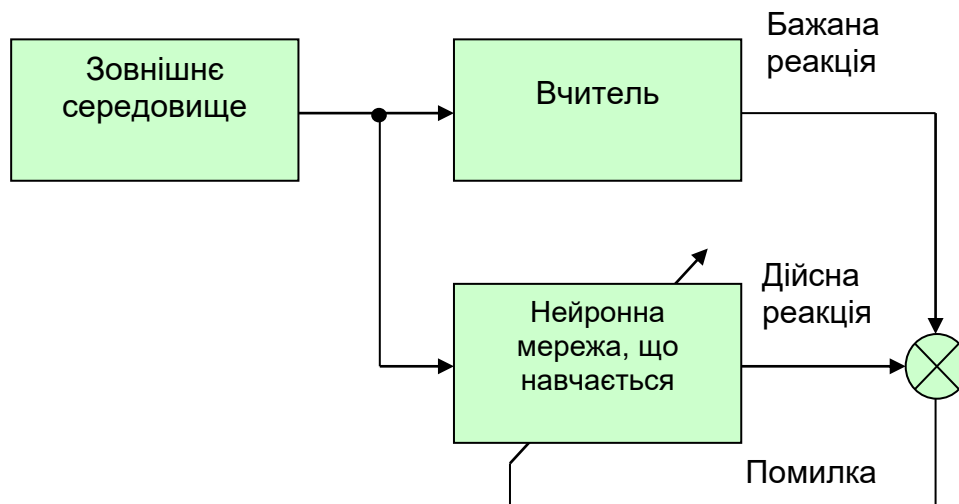


Рис. 1.9. Схема прямого контрольованого навчання

Нейронна мережа у вихідному стані ніякими знаннями не володіє. В процесі навчання «вчитель» та мережа піддаються впливу з боку зовнішнього

середовища, тобто на їхні входи поступає тренувальний сигнал, що співпадає з одним із вхідних шаблонів. «Вчитель» повідомляє мережу про те, якою має бути правильна (бажана) реакція на вплив, що надійшов, надаючи відповідний шаблон виходу. Виходячи з помилки між реальним та бажаним виходами мережі, відбувається налаштування її синаптичних зв'язків за певним правилом. Повторюючи цей процес ітеративно, можна налаштувати мережу так, що вона буде емулювати «вчителя», тобто знання «вчителя» про зовнішнє середовище перейдуть до мережі.

Типовими задачами, що вирішуються за допомогою прямого контрольованого навчання, є апроксимація невідомої функції, що описується набором даних, та ідентифікація динамічного об'єкта. У таких задачах зазвичай заздалегідь відомі вхідні сигнали та правильна реакція на них, тобто тренувальний набір шаблонів.

Найбільш поширеним методом прямого контрольованого навчання багатопарових нейронних мереж прямого розповсюдження є **алгоритм зворотного розповсюдження помилки** (backpropagation algorithm), який являє собою узагальнення методу найменших квадратів. Але, перш ніж перейти до його більш детального розгляду, доречно охарактеризувати другий різновид контрольованого навчання – стимульоване навчання.

1.5.2. Стимульоване навчання

Стимульоване навчання не використовує знання «вчителя» про бажаний вихід мережі. Замість цього навчання проводиться за результатом оцінювання якості перетворення вхід-вихід, що провадить сама мережа. Оцінювання здійснюється на підставі аналізу реакції зовнішнього оточення після подачі на вхід мережі тренувального вхідного впливу (рис. 1.10). При цьому налаштування параметрів мережі здійснюється так, щоб надати максимум значенню певного чисельного критерію, який називається стимулом (reinforcement signal).

Ідея такого способу виходить з реального процесу навчання, що відбувається у живих істот. У психології він відомий як *закон дії Торндайка*. Стосовно стимульованого навчання нейронних мереж цей закон можна сформулювати наступним чином:

Якщо внаслідок модифікації синаптичних зв'язків, яка відбулась в процесі навчання, нейронна мережа дає покращений (більш задовільний)

результат, то тенденція коригувати синаптичні зв'язки у тому ж напрямку збільшується (система стимулюється). У протилежному випадку тенденція зменшується.

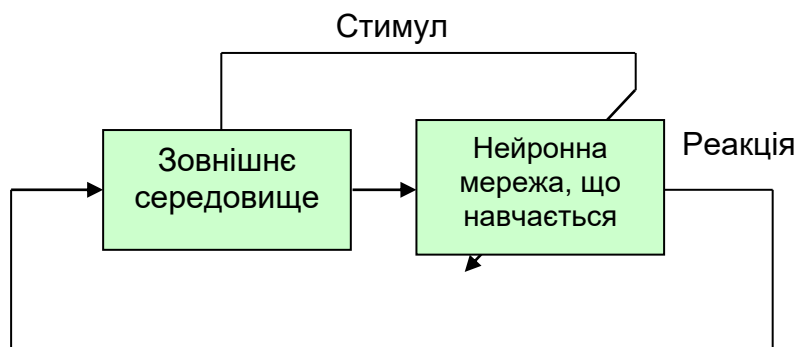


Рис. 1.10. Схема стимульованого навчання

Найбільш характерним прикладом використання стимульованого навчання є створення адаптивних систем управління. У ній частиною, що навчається, є контролер, а об'єкт управління, зовнішні впливи й сигнали завдання виступають його зовнішнім середовищем. У результаті впливу з боку цього середовища контролер виробляє певний сигнал управління, що переводить об'єкт до нового стану. При цьому якість управління можна оцінити лише за сигналом на виході об'єкта. Оскільки необхідна реакція контролера, що забезпечує заданий стан об'єкта, заздалегідь невідома, не можна сформувати тренувальний набір шаблонів. А, отже, через неможливість застосування прямого контрольованого навчання у цьому випадку необхідно використовувати стимульоване навчання контролера за якістю роботи всієї системи управління в цілому, тобто за оцінкою стану зовнішнього середовища.

Стимульоване навчання може реалізовуватись двома способами: з безпосереднім стимулюванням та із затриманим стимулюванням. У першому випадку оцінювання роботи й відповідне їй налаштування параметрів нейронної мережі здійснюється на кожному кроці роботи системи навчання. У другому випадку, при затриманому стимулюванні, налаштування мережі виконується так, щоб максимізувати кумулятивну оцінку роботи системи на певній послідовності кроків.

Хоча безпосереднє стимулювання більш властиве класичним схемам адаптивного управління, останнім часом велика увага приділяється методам

затриманого стимулювання. Для збіжності методу безпосереднього стимулювання необхідно, щоб кожний наступний стан зовнішнього середовища визначався лише його попереднім станом та попереднім впливом з боку нейронної мережі. При затриманому ж навчанні таке обмеження усувається. З іншого боку, набагато простіше сформулювати критерій, що визначає оптимальне поводження системи управління в цілому за результатами якоїсь дії, ніж критерій, що направляє її рух на кожному кроці.

Загалом, порівнюючи пряме контрольоване та стимульоване навчання, можна стверджувати, що система, яка навчається лише за результатами своєї взаємодії із зовнішнім середовищем, є більш інтелектуальною за систему, що використовує додаткову інформацію «вчителя».

1.5.3. Алгоритм зворотного розповсюдження помилки

Як уже зазначалось, найбільш розвиненою формою навчання нейронних мереж є контрольоване навчання зі «вчителем». Зазвичай на початку навчання вагові коефіцієнти синаптичних зв'язків устанавлюються випадковим чином і мають невеликі значення. Отже, після пред'явлення мережі першого тренувального взірця поява на її виході вірного результату виявляється малоімовірною. Розбіжність між тим, що дає мережа, й тим, що має бути за даними взірця, складає помилку δ , яка може бути використана для коригування ваги. При цьому значення вагового коефіцієнту $\Delta w(n+1)$ на черговому $(n+1)$ -му кроці визначається шляхом додавання до того значення $\Delta w(n)$, що було на попередньому n -му кроці, добавки Δw :

$$w(n+1) = w(n) + \Delta w. \quad (1.11)$$

Правило, за яким найчастіше визначається добавка Δw , називається *алгоритмом зворотного розповсюдження помилки*. Цей алгоритм широко використовується для навчання багатошарових повнозв'язаних нейронних мереж прямого розповсюдження, тобто мереж, в яких сигнали передаються лише в напрямку від вхідного шару до шару на виході, а елементи (нейрони) попереднього шару з'єднані з усіма елементами наступного шару.

Для певного j -го нейрону останнього шару такої мережі, що зображений на рис. 1.11, комбінований ввід s_j становить

$$s_j = \sum_i x_i w_{ij} + w_{0j} \quad (1.12)$$

Вихідний сигнал o_j за умов використання логістичної функції активації становить

$$o_j = \frac{1}{1 + e^{-s_j}}. \quad (1.13)$$

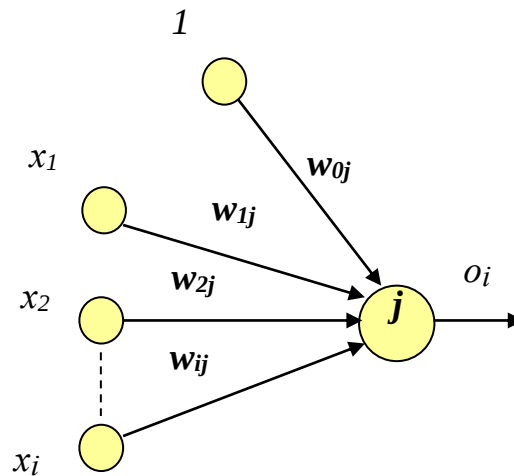


Рис. 1.11. Окремий нейрон мережі прямого зв'язку

Помилка на виході нейрона:

$$\delta_j = (t_j - o_j), \quad (1.14)$$

де t_j – бажаний сигнал на виході нейрона, який визначається відповідним зразком тренувального набору даних.

Квадратична помилка на виході нейрона:

$$E_j = \frac{1}{2} \delta_j^2 = \frac{1}{2} (t_j - o_j)^2 \quad (1.15)$$

(множник $\frac{1}{2}$ включено до формули з метою спрощення подальших математичних перетворень, що жодним чином не впливає на їхню строгість).

Добавка Δw_{ij} , яка коригуватиме синаптичну вагу w_{ij} у протилежному зростанню помилки напрямку визначається за формулою

$$\Delta w_{ij} = -\eta \frac{\partial E_j}{\partial w_{ij}}, \quad (1.16)$$

де η – норма навчання: мале дійсне число, яке обирається з діапазону $[0,1]$ і, обмежуючи швидкість змінення ваги, забезпечує збіжність процедури навчання.

Тобто

$$w_{ij}(n+1) = w_{ij}(n) - \eta \frac{\partial E_j}{\partial w_{ij}}. \quad (1.17)$$

Таким чином, коригування певного вагового коефіцієнта здійснюється пропорційно чутливості квадратичної помилки до цього коефіцієнта. Інакше кажучи, чим відчутніше квадратична помилка реагує на змінення відповідного коефіцієнта, тим більший крок у бік його корекції здійснює алгоритм навчання, і навпаки: чим менше змінення певного коефіцієнта впливає на зменшення помилки, тим меншою буде відповідна корекція.

Представимо чутливість квадратичної помилки до синаптичної ваги у вигляді

$$\frac{\partial E_j}{\partial w_{ij}} = \frac{\partial E_j}{\partial o_j} \cdot \frac{\partial o_j}{\partial s_j} \cdot \frac{\partial s_j}{\partial w_{ij}}. \quad (1.18)$$

Визначимо кожен з часткових похідних, які входять до цього виразу:

$$\text{а) } \frac{\partial E_j}{\partial o_j} = \frac{\partial}{\partial o_j} \left[\frac{1}{2} (t_j - o_j)^2 \right] = -(t_j - o_j) = -\delta_j; \quad (1.19)$$

$$\text{б) } \frac{\partial o_j}{\partial s_j} = \frac{\partial}{\partial s_j} \left(\frac{1}{1 + e^{-s_j}} \right) = \frac{e^{-s_j}}{\left(1 + e^{-s_j} \right)^2} =$$

$$= \frac{1}{1+e^{-s_j}} \left(1 - \frac{1}{1+e^{-s_j}} \right) = o_j(1-o_j);$$

(1.20)

$$в) \quad \frac{\partial s_j}{\partial w_{ij}} = x_i. \quad (1.21)$$

Отже

$$\frac{\partial E_j}{\partial w_{ij}} = -(t_j - o_j) o_j(1-o_j) x_i. \quad (1.22)$$

Після підстановки виразу (1.22) до формули (1.17) отримуємо

$$w_{ij}(n+1) = w_{ij}(n) + \eta (t_j - o_j) o_j(1-o_j) x_i. \quad (1.23)$$

Зважаючи на вигляд формул (1.17) та (1.23), у коригуванні синаптичних вагових коефіцієнтів для нейронів, що розміщуються в останньому шарі мережі, не виникатиме жодних проблем, адже для них цільові значення активності t_j є відомими. Проблема виникає з визначенням помилки на виходах нейронів прихованих шарів, у яких помилка немає безпосереднього зв'язку з цільовим значенням на виході.

Для вирішення цієї проблеми будемо виходити з того, що синаптичні коефіцієнти прихованих нейронів необхідно коригувати пропорційно їхньому «внеску» до помилки наступного шару. Тобто в мережі з прихованими шарами (рис. 1.12) помилка кожного нейрона останнього шару δ_j вносить свою частку μ_j до помилки δ_i кожного нейрона прихованого передостаннього шару

$$\delta_i = \sum_j \delta_j \mu_j.$$

Для визначення тієї частки помилки i -го нейрона прихованого шару, що обумовлена помилкою j -го нейрона наступного шару, приймемо, що вона дорівнює чутливості помилки δ_j до помилки δ_i

$$\delta_{ij} = \delta_j \frac{\partial \delta_j}{\partial \delta_i},$$

тобто внесок помилки δ_j до помилки δ_i тим більший, чим сильнішим є вплив δ_i помилки на помилку δ_j .

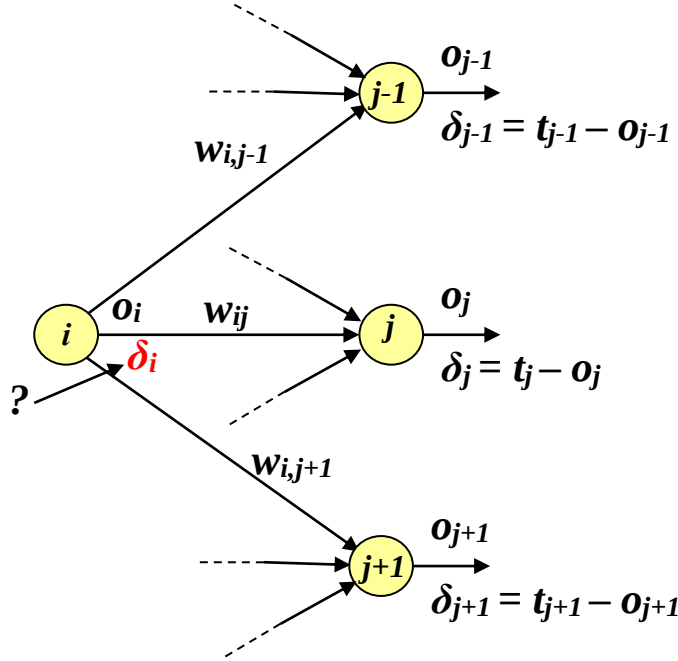


Рис. 1.12. До визначення помилки нейрона у прихованому шарі

Беручи до уваги, що

$$\partial \delta_i = \partial(t_i - o_i) = -\partial o_i,$$

можемо записати

$$\delta_{ij} = \delta_j \frac{\partial \delta_j}{\partial \delta_i} = -\delta_j \frac{\partial o_j}{\partial \delta_i} = -\delta_j \frac{\partial o_j}{\partial s_i} \cdot \frac{\partial s_j}{\partial \delta_i}. \quad (1.24)$$

Похідна $\frac{\partial o_j}{\partial \delta_i}$ в цьому виразі визначається за формулою (1.20), а

похідна $\frac{\partial s_j}{\partial \delta_i} = -\frac{\partial s_j}{\partial o_i} = -\frac{\partial w_{ij} o_i}{\partial o_i} = -w_{ij}$. Підстановка до (1.24) дає

$$\delta_{ij} = \sum_j \delta_j o_j (1 - o_j) w_{ij}. \quad (1.25)$$

Отже помилка δ_i нейрона передостаннього прихованого шару може бути визначена наступним чином

$$\delta_i = \sum_j \delta_j o_j (1 - o_j) w_{ij}. \quad (1.26)$$

Розглянемо алгоритм зворотного розповсюдження помилки на прикладі тришарової повнозв'язної нейронної мережі (рис. 1.13) [6], яка має два нейрони у вхідному шарі та по два нейрони у кожному з двох прихованих шарів. Останній шар містить один нейрон.

Припустимо, що мережі представлений такий поточний тренувальний шаблон: $x_1 = 0,1$; $x_2 = 0,9$ (вхідні сигнали) та $t_j = 0,9$, (цільовий вихід). Поточні значення вагових коефіцієнтів проставлені над стрілками відповідних зв'язків. Активність допоміжних елементів, які забезпечують надання нейронам зсуву, дорівнює 1.

Алгоритм зворотного розповсюдження помилки передбачає два потоки в мережі: прямий потік від вхідного шару до шару на виході (рис. 1.13,а) та зворотний потік – від шару на виході до вхідного (рис. 1.13,б). Прямий потік просуває крізь мережу вхідну комбінацію, після чого на виході мережі з'являється результат. На рис. 1.13,а під кожним нейроном наведене значення його комбінованого входу, а над ним – значення його активності.

Зворотний прохід просуває назад по мережі значення помилок. На рис.1.13,б над кожним нейроном наведене значення його помилки δ , яка обчислена за формулою (1.26).

На рис. 1.14 наведені оновлені вагові коефіцієнти, які обчислюються за формулою (1.23). При цьому прийнята норма навчання $\eta = 0,8$. На рис. 1.14 під кожним нейроном наведене значення його активності o , а над ним – помилки δ .

Після оновлення вагових коефіцієнтів на вхід мережі подається наступний тренувальний шаблон (взірець) з тренувального набору і процедура корекції вагових коефіцієнтів повторюється. Повний цикл опрацювання усіх шаблонів тренувального набору називається **епохою**.

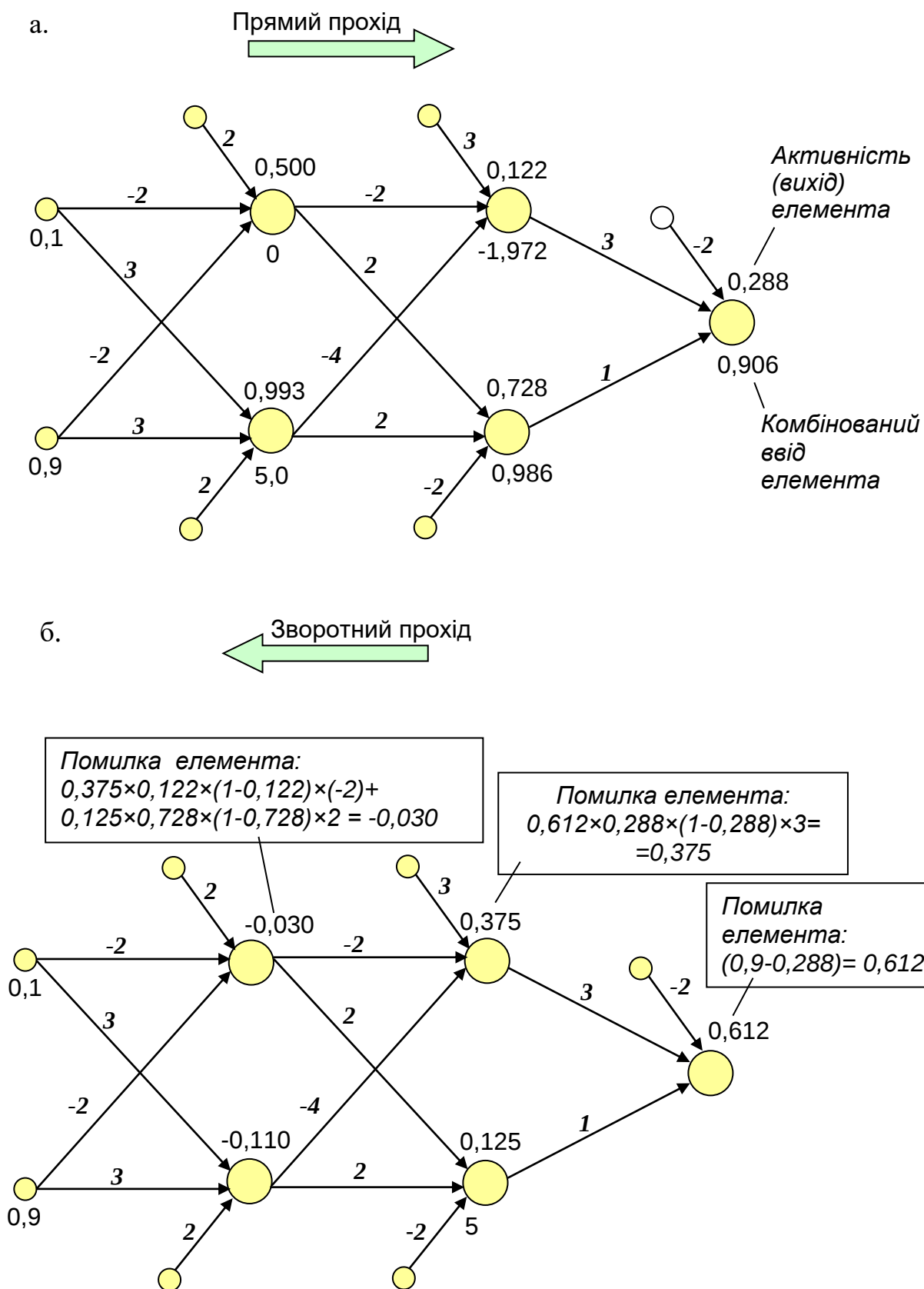


Рис.1.13. Приклад прямого та зворотного проходів у тришаровій мережі прямого розповсюдження

Взірці подаються на опрацювання мережею знов і знов, епоха за епохою, доки на протязі однієї епохи помилки усіх нейронів на виході мережі для кожного взірця не попадуть у припустимі межі. Іноді навчання триває доки зміна усередненої квадратичної помилки при переході від однієї епохи до іншої не стане меншою за певне припустиме значення. Наприклад, припустиме значення 0,01 означає, що усереднені квадратичні помилки сусідніх епох відрізняються не більше, ніж $\pm 0,01$.

Якщо під час навчання помилка в мережі попадає в межі припустимого змінення, говорять, що спостерігається **збіжність**.

На завершення процесу навчання мережа піддається тестуванню на даних, які не пред'являлися їй на етапі тренування.

1.5.4. Властивість узагальнення

У процесі контрольованого навчання вагові синаптичні коефіцієнти мережі налаштовуються так, щоб вона апроксимувала вихідні дані, що містяться в пропонованих їй тренувальних шаблонах. Наприкінці процесу навчання мережа запам'ятовує всі тренувальні дані, тобто правильно реагує на тестові входи. Однак крім здатності пам'ятати, мережа має набути ще й властивість узагальнювати тренувальні дані.

Використаний тут термін **узагальнення** запозичений зі психології й для натренованої нейронної мережі означає явище, коли здійснюване нею перетворення вхід-вихід є вірним як для тренувальних даних, так і для тестових даних, що не використовувались при тренуванні. Узагальнення виникає в нейронній мережі як наслідок гарної нелінійної інтерполяції вхідних даних, головним чином завдяки безперервності й гладкості активаційної функції самих нейронів.

На рис. 1.15 проілюстрована властивість узагальнення для гіпотетичного тришарового персептрона з одним входом та одним виходом. Крива, що зображена на рисунку, відображає нелінійне перетворення, яке здійснюється мережею. Точки, позначені як «тренувальні дані», використовувались під час навчання мережі, інші точки, позначені як «узагальнення», отримані як результат інтерполяції, що проведена мережею.

Але нейронна мережа не завжди може узагальнювати дані. На рис. 1.16 зображене перетворення, що його здійснила мережа, яка навчалась на такому ж самому тренувальному наборі. Видно, що в цьому випадку мережа не може коректно інтерполювати вхідні дані. Це пояснюється тим, що мережа була

перетренована, тобто вона запам'ятала тренувальні дані як таблицю й видає коректні дані лише при точній вказівці на «адресу» у ній.

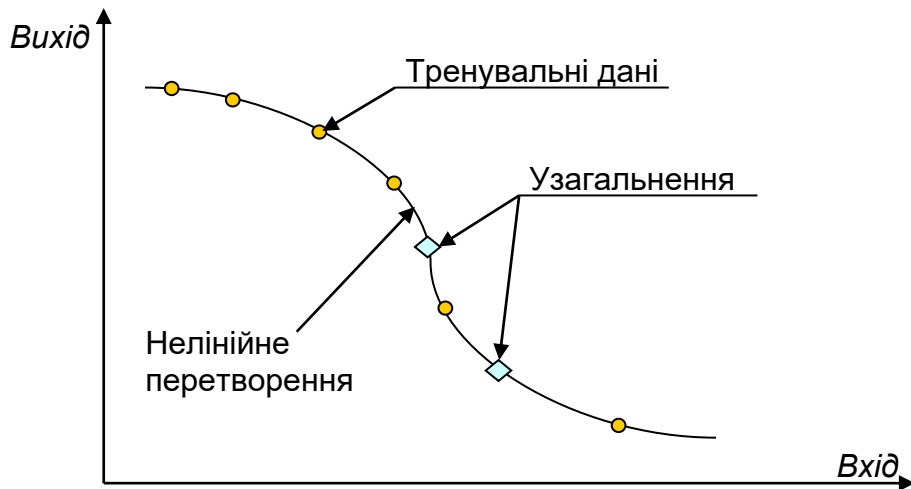


Рис. 1.15. Явище узагальнення в нейронній мережі

Можна виділити дві необхідних, але не достатніх умови існування узагальнення в нейронних мережах. Перша – це гладкість того нелінійного перетворення, якому мережа повинна навчитись. Тобто малим змінам вхідних сигналів повинні відповідати малі ж зміни сигналів на виході. Ця умова не завжди виконується, особливо якщо тренувальні дані отримані в ході реальних експериментів і містять велику кількість перешкод.

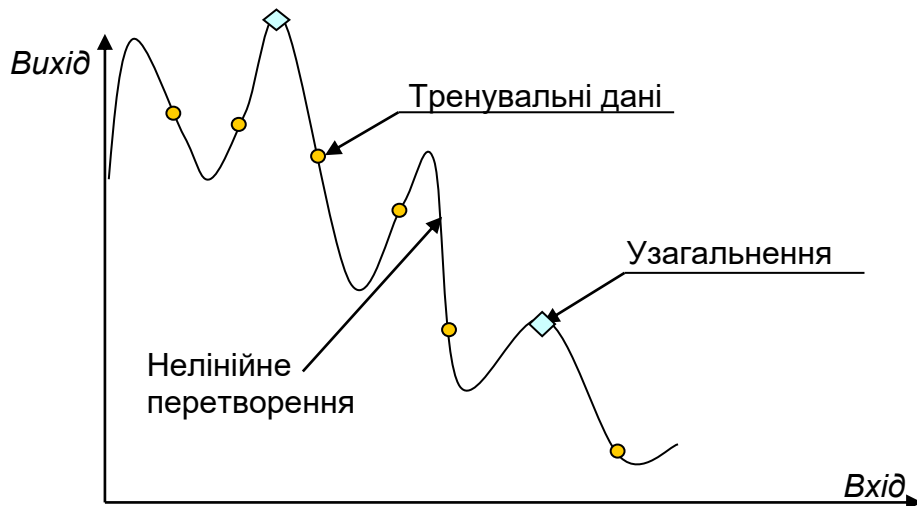


Рис. 1.16. Низький рівень узагальнення в перетренованій нейронній мережі

Другою умовою є достатня репрезентативність тренувального набору даних. Зазвичай задача, яку розв'язуватиме нейронна мережа, містить обсяг

даних, який набагато перевищує обчислювальні можливості комп'ютерів, тому для її навчання мережі використовується тільки певна тренувальна вибірка. Відбір даних у тренувальний набір необхідно проводити так, щоб, з одного боку, його можна було опрацювати за прийнятний час, а з іншого – він достатньо повно описував вихідне завдання.

Підходячи спрощено, можна класифікувати проведене мережею узагальнення, як інтерполяцію або екстраполяцію залежно від того, як у вхідному просторі розміщуються тестові дані відносно тренувальних шаблонів. З огляду на те, що інтерполяція виконується простіше й точніше, ніж екстраполяція, можна стверджувати, що мережа тим краще узагальнює, чим більш рівномірно й щільніше розташовані тренувальні дані у вхідному просторі. Якщо тестові дані завжди будуть опинятись між близько розташованими тренувальними шаблонами, мережа зможе здійснювати коректне узагальнення, інтерполюючи, а не екстраполуючи, вхідні дані.

Величезний вплив на якість узагальнення має власне процес навчання. Перша з погроз втрати узагальнення пов'язана із можливістю недотренувати мережу, припинивши навчання відразу після досягнення прийнятної помилки на тренувальному наборі. У цьому випадку в мережі можуть залишитися нейрони, що не задіяні в запам'ятовуванні тренувальних даних, і їхній внесок у перетворення, здійснюване мережею, може заважати правильній інтерполяції. Друга погроза полягає у ризику перетренувати нейронну мережу. Мається на увазі, що кожний нейрон перетренованої мережі налаштований так, щоб максимально точно відповідати всім тренувальним шаблонам, але в результаті цього дія активаційних функцій нейронів стає локальною, а інтерполяція за ними – неможливою. Фактично, перетренована мережа перетворюється на таблицю, як на рис. 1.16.

Виходом у даній ситуації є постійний контроль в процесі навчання за здатністю мережі узагальнювати. Для цього можна використовувати розширений набір тестових даних, який містить крім тренувальних ще й близькі до них додаткові шаблони.

1.5.5. Деякі особливості підготовки даних для навчання мережі

Якість роботи нейронної мережі значною мірою залежить від якості даних, які пред'являються їй під час навчання. Тренувальний набір повинен містити дані, які є типовими для задачі, яку має розв'язувати мережа. При

цьому дуже важливо, щоб дані охоплювали усю область припустимих значень, усі можливі стани проблеми, що вирішується.

Зазвичай дані, які використовуються для навчання нейронної мережі, заздалегідь розділяють на дві категорії: перші використовують для тренування (коректування ваги синаптичних зв'язків), інші – для тестування.

Насправді, виявлення реальних властивостей нейронної мережі та з'ясування її спроможності до узагальнення можливе лише під час тестування, коли аналізується реакція мережі на дані, що є хоча й аналогічними до тренувальних, але все ж таки відрізняються від них.

Дані необхідно перевірити на вірогідність. Як правило, у великому наборі завжди знаходяться неузгодженості та помилки. Іноді невірні дані вносяться оператором. Іноді набір містить сторонні дані. Але тут вимагається увага, адже іноді сторонні значення можуть нести корисну інформацію, скажімо, про аварійні ситуації.

Дані зазвичай потребують масштабування. Наприклад, сигнали на виході нейронів із сигмоїдальною функцією активності лежать між 0 та 1, оскільки такою є область значень сигмоїдальної функції. Кожний сигнал може потребувати свого масштабування. Припустимо, що один вхідний сигнал змінюється в діапазоні від 300 до 2000, а другий – від 5 до 130. Тоді перший сигнал буде домінувати над другим, бо матиме більший вплив на комбінований ввід відповідного нейрона.

Найпростішим методом масштабування є ділення поточного значення сигналу на його максимальне значення. В результаті найбільше значення масштабованого сигналу буде обмежене значенням «1».

Для сигмоїдальної функції активності з метою запобігання переходу мережі до режиму насичення часто застосовують масштабування до діапазону 0,1 – 0,9 за допомогою формули

$$x' = \frac{0,9-0,1}{x_{\max} - x_{\min}} x + \left(0,9 - \frac{0,9-0,1}{x_{\max} - x_{\min}} x_{\max} \right), \quad (1.27)$$

де x' – нове масштабоване значення сигналу x .

Відомі й інші методи масштабування, наприклад, із визначенням для кожного вхідного сигналу математичного очікування m_x та середнього квадратичного відхилення σ_x :

$$x' = \frac{x - m_x}{\sigma_x}. \quad (1.28)$$

1.6. Радіально-базисні нейронні мережі

Радіально базисна нейронна мережа (*Radial Basis Function network*) являє собою двошарову нейронну мережу, що складається з прихованого шару радіальних базисних нейронів та вихідного лінійного шару. Як зазначалось у розд. 1.3, функція активації нейронів прихованого шару RBF-мережі має вигляд функції Гауса (рис. 1.17)

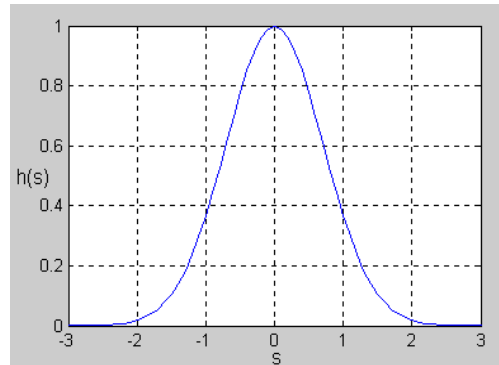


Рис. 1.17. Графік функції активації нейронів RBF-мережі (функції Гауса)

Доведено, що RBF-мережі є майже ідеальними апроксиматорами нелінійних залежностей.

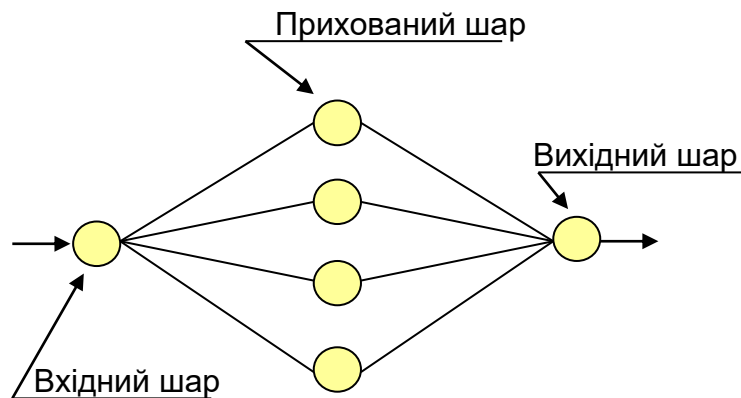


Рис. 1.18. RBF-мережа з одним вхідним та одним вихідним нейроном

Якщо кількість нейронів вхідного шару та шару на виході RBF-мережі дорівнює одному (рис. 1.18), вона може застосовуватись для апроксимації нелінійної функції одного аргументу. Таку апроксимацію нелінійної кривої сумою багатьох функцій Гауса

$$f_i(x) = \exp - \left(\frac{(x - C_i)^2}{2\sigma^2} \right), \quad (1.29)$$

які мають різні центри та однакову ширину вікна, ілюструє рис. 1.19.

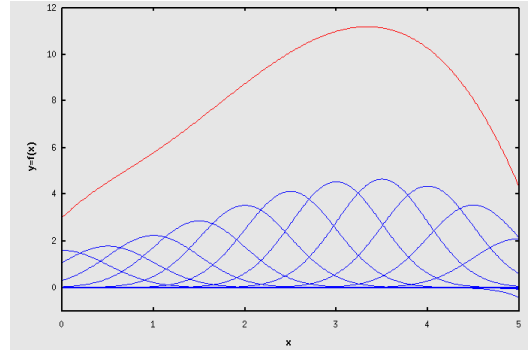


Рис. 1.19. Апроксимація нелінійної функції одного аргументу RBF-мережею

На рис. 1.20 наведена схема RBF-мережі з двома вхідними нейронами.

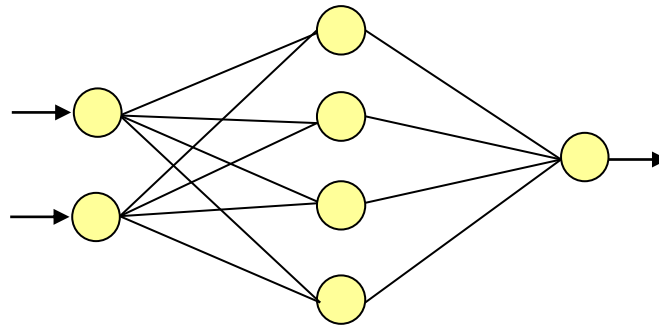


Рис. 1.20. RBF-мережа з двома вхідними та єдиним вихідним нейроном

У такій мережі кожний нейрон прихованого шару здійснює перетворення вхідного вектору $X [x_1, x_2]$ за формулою

$$f_i(x) = \exp - \left(\frac{(x_1 - C_{i,1})^2 + (x_2 - C_{i,2})^2}{2\sigma^2} \right), \quad (1.30)$$

формуючи відповідну поверхню (рис. 1.21).

Розглянемо приклад [7] використання радіально-базисної мережі для розв'язання задачі апроксимації функції однієї змінної (рис. 1.22) на інтервалі $[-1, 1]$.

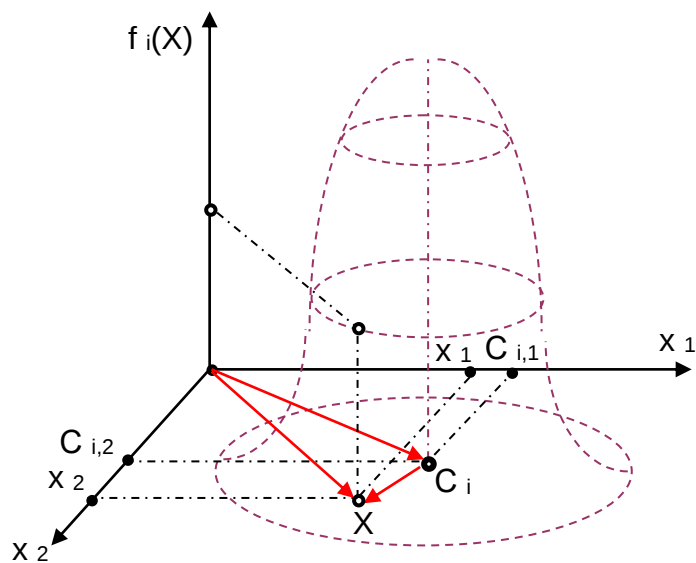


Рис. 1.21. Поверхня функції активації нейронів RBF-мережі з двома вхідними нейронами

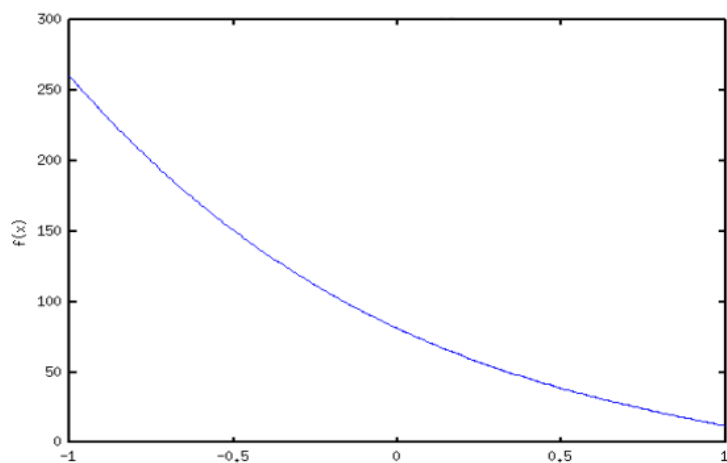


Рис. 1.22. Графік функції, що апроксимується

Нехай відомі 9 точок кривої, що описується функцією

$$f(x) = (x-3)^4 - 4x^3. \quad (1.31)$$

Будемо апроксимувати цю криву за допомогою дев'ятох зважених функцій Гауса (рис. 1.23)

$$f(x) = \sum_{i=1}^9 \lambda_i f_i(x),$$

$$f_i = e^{-\frac{(x-c_i)^2}{2\sigma^2}}, i = 1, 2, \dots, 9,$$

де C_i – координати центрів цих функцій, σ – параметр, що визначає крутість дзвоноподібних графіків (ширину вікна) функцій.

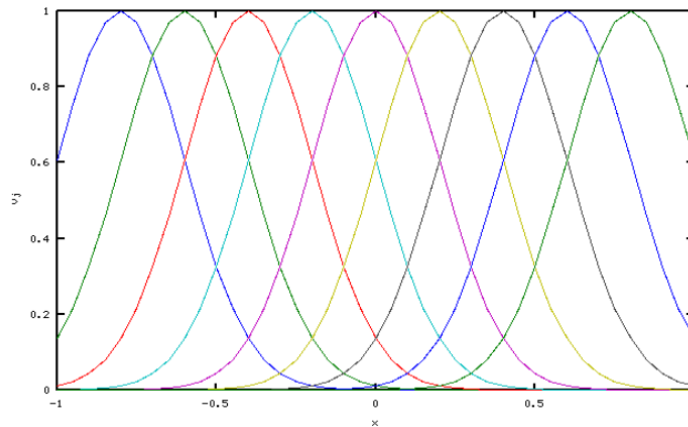


Рис. 1.23. Множина радіальних функцій на інтервалі $[-1, 1]$

Нейронна мережа матиме вигляд, що наведений на рис. 1.24. Вона має один скалярний вхід, значення якого передається на дев'ять радіально-базисних нейронів, зважені виходи яких формують скалярний вихід мережі.

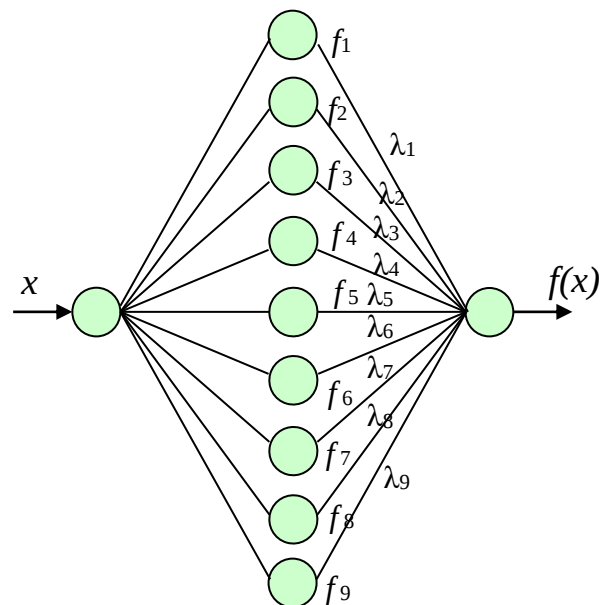


Рис. 1.24. RBF-мережа для апроксимації функції (1.31)

Кількість нейронів прихованого шару, розташування їх центрів, а також ширина вікна σ радіальної функції підбираються експертом під конкретні умови. Від цих параметрів залежить точність наближення.

Не маючи певних міркувань щодо розташування центрів радіальних функцій, розмістимо їх рівномірно на осі всередині обраного інтервалу апроксимації у точках $[-0.8; -0.6; -0.4; -0.2; 0; 0.2; 0.4; 0.6; 0.8]$.

Ширину «вікна» функції призначимо $\sigma=0,04$

Крива апроксимується зваженою сумою (лінійною комбінацією) базисних функцій, що реалізуються нейронами прихованого шару. Таким чином, ці функції є базисом для побудови апроксимуючої кривої (рис. 1.23).

Необхідно знайти вагові коефіцієнти λ_i цих функцій, які мають забезпечувати отримання значень апроксимованої функції в точках навчальної вибірки, тобто:

$$\begin{cases} f_1(x_1)\lambda_1 + f_2(x_1)\lambda_2 + \dots + f_9(x_1)\lambda_9 = f(x_1) \\ f_1(x_2)\lambda_1 + f_2(x_2)\lambda_2 + \dots + f_9(x_2)\lambda_9 = f(x_2) \\ \dots \\ f_1(x_9)\lambda_1 + f_2(x_9)\lambda_2 + \dots + f_9(x_9)\lambda_9 = f(x_9) \end{cases} \quad (1.32)$$

Звідси видно, що кількість точок навчальної вибірки має бути не менше числа нейронів прихованого шару. Тоді знайти вагові коефіцієнти λ_i прихованого шару можна буде, вирішивши систему рівнянь (1.32).

Нехай, наприклад, точками навчальної вибірки будуть такі:

$$x = [-0,87; -0,64; -0,22; 0,07; 0,16; 0,42; 0,5; 0,77; 0,96],$$

а реакції мережі (значення функції) для кожної з цих точок будуть такими:

$$f(x) = [226,94; 176,6; 107,55; 73,70; 65,04; 44,01; 38,56; 22,90; 13,78]$$

Вирішивши систему рівнянь (1.30), отримаємо вектор вагових коефіцієнтів:

$$\lambda = [280,24; -122,16; 200,44; -59,18; 91,31; -14,13; 42,63; -5,23; 19,23].$$

Тоді з урахуванням отриманих вагових коефіцієнтів радіальні функції матимуть вигляд, що наведений на рис. 1.25.

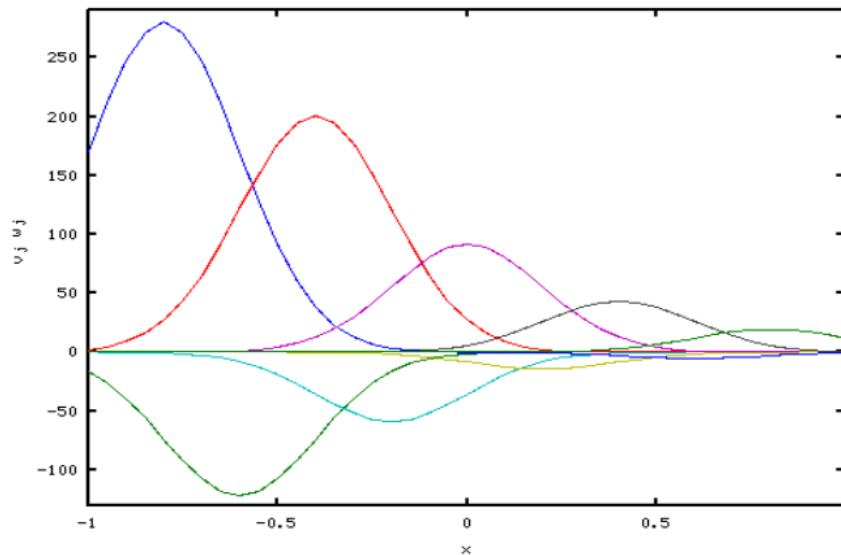


Рис. 1.25. Масштабовані радіально-базисні функції

Сума цих функцій дає нам апроксимуючу криву, що проходить через усі точки навчальної вибірки (рис. 1.26).

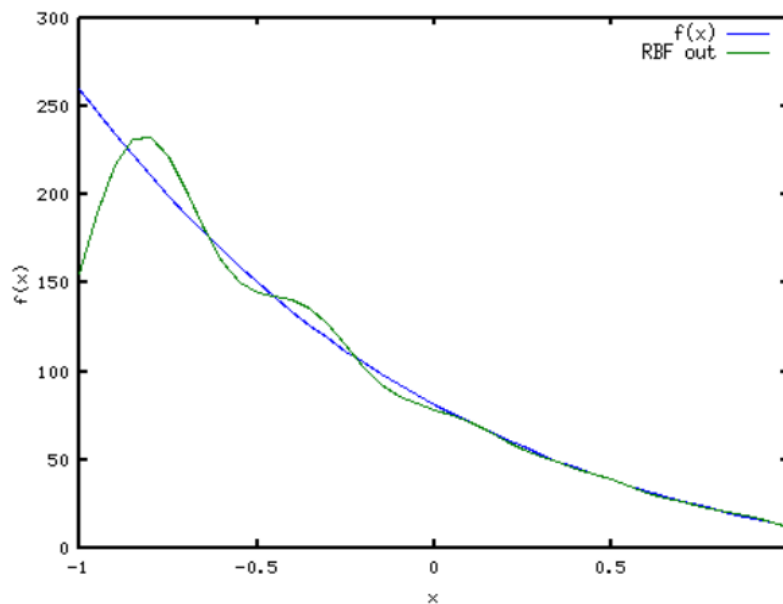


Рис. 1.26. Вихідна функція $f(x)$ та крива, що її апроксимує

Таким чином, навчання радіально-базисної мережі у класичному розумінні можна уникнути. Воно може бути замінене простим розв'язанням системи лінійних алгебраїчних рівнянь.

Контрольні запитання та завдання

1. Наведіть структуру штучного нейрона
2. Які функції активації найчастіше використовується в моделях нейронних мереж?
3. Що розуміють під архітектурою нейронної мережі?
4. Наведіть структуру багатошарового перцептрона.
5. Поясніть, чому рекурентні нейронні мережі набувають властивості відображати динамічні процеси?
6. Що розуміють під навчанням нейронної мережі і що є результатом навчання?
7. У чому полягає навчання нейронної мережі з учителем? В яких випадках воно може застосовуватись?
8. Як відбувається навчання нейронної мережі без учителя? Наведіть приклади задач автоматизації, для вирішення яких воно може застосовуватись.
9. За яких умов може бути завершене (або припинене) навчання нейронної мережі з учителем?
10. Опишіть сутність навчання нейронних мереж за алгоритмом зворотного розповсюдження помилки. В чому полягає ідея цього методу?
11. До якого типу мереж відноситься класичний багатошаровий перцептрон, що навчається за методом зворотного розповсюдження помилки?
12. У чому полягає властивість узагальнення нейронної мережі?
13. З яких причин нейронна мережа може не набути властивості узагальнення?
14. Які особливості підготовки даних для навчання нейронних мереж вам відомі?
15. Навіщо застосовують масштабування (нормалізацію) даних під час навчання нейронних мереж?

2. ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В СИСТЕМАХ УПРАВЛІННЯ

До початку ХХІ століття у побудові регуляторів в системах автоматичного управління домінував традиційний підхід, що передбачав синтез структури та параметрів регулятора з використанням моделі об'єкта управління (ОУ), отриманої за результатом ідентифікації. Короткий огляд класичних (традиційних) методів побудови систем автоматичного управління можна знайти у Додатку 1. Проте, регулятори, побудовані на базі такого підходу, не завжди дозволяють здійснювати якісне управління складними нестационарними об'єктами. Саме це стало підставою для розвитку нового напрямку – інтелектуальних систем управління.

Інтелектуальні системи управління – це системи, що здатні до «розуміння» й навчання відносно ОУ, збурень, зовнішнього середовища та умов функціонування. Головна їх відмінність від традиційних систем – наявність механізму отримання, зберігання та обробки знань про ОУ для реалізації власних функцій.

Серед сучасних інформаційних технологій, що дозволяють створювати інтелектуальні системи управління, особливе місце посідають штучні нейронні мережі, яким властиві притаманні людині прийоми навчання.

Структурно інтелектуальні системи управління на базі штучних нейронних мереж містять додаткові блоки, що здійснюють системну обробку даних, що надходять від ОУ. Ці блоки можуть виконуватись як надбудова над звичайним регулятором, налаштовуючи потрібним чином його параметри, або безпосередньо включатися до замкненого контуру управління.

Загалом задача управління за допомогою нейронних мереж зводиться до синтезу нейромережного контролера, який має формувати такий сигнал управління, що призводить до необхідного (бажаного) поведіння ОУ.

Використання нейронної мережі в якості нейромережного контролера можливе у двох варіантах. За першим – нейронна мережа навчається в режимі *on-line* і одночасно формує сигнал управління. За другим – робота мережі складається із двох етапів: попереднього етапу навчання в режимі *off-line* заданій оптимальній функції управління та етапу відтворення цієї

функції в режимі управління об'єктом за тих самих або близьких до них умов.

Вибір конкретного підходу до навчання мережі (on-line або off-line) залежить від специфіки завдання й визначає конкретний вид алгоритму навчання. Так, наприклад, у галузях промисловості, де накопичені величезні масиви даних щодо поведінки досліджуваного технологічного об'єкта, доцільно використовувати off-line навчання. У тих випадках, коли технічні характеристики об'єкта змінюються в процесі експлуатації, тобто технологічний об'єкт є нестационарним, доцільнішим є застосування on-line навчання.

2.1. Пряме інверсне управління

Пряме *інверсне управління* (*direct inverse control*, DIC) являє собою послідовне з'єднання нейромережного контролера, що є інверсною моделлю об'єкта управління, і безпосередньо самого об'єкта управління (рис. 2.1).

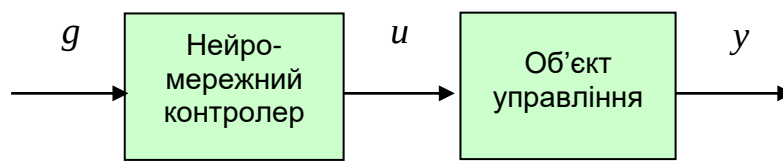


Рис. 2.1. Схема прямого інверсного нейромережного управління: g – сигнал завдання; u – сигнал управління; y – сигнал на виході

За допомогою схеми прямого інверсного управління, в якій нейромережний контролер (НК) реалізує точну зворотну (інверсну) модель ОУ

$$W_{НК}(p) = \frac{1}{W_{ОУ}},$$

теоретично можна досягти ідеального управління, за яким вихід об'єкта завжди дорівнює завданню.

Для отримання нейромережної інверсної моделі об'єкта управління необхідно навчити нейронну мережу його зворотній динаміці. Із цією метою на вхід нейронної мережі подається сигнал з виходу об'єкта, потім її вихід порівнюється зі вхідним сигналом об'єкта, а сигнал помилки використовується для навчання мережі (рис. 2.2). Навчена у такий спосіб нейронна мережа використовується потім у схемі на рис. 2.1.

Якщо параметри ОУ можуть змінюватись в процесі функціонування, застосовують схему, що допускає оперативне підстроювання НК (рис.2.3), при цьому НК просто періодично копіює нейромережну інверсну модель.

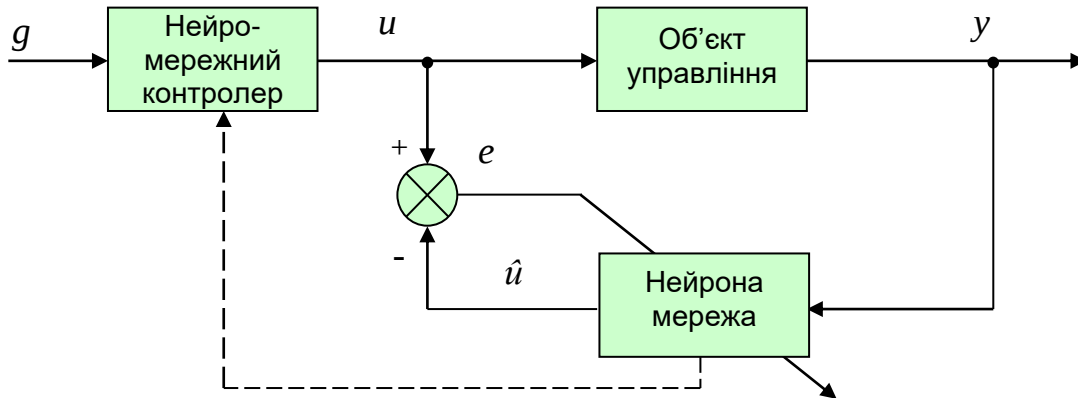


Рис. 2.2. Схема навчання нейронної мережі інверсній динаміці об'єкта:
 u – сигнал управління; y – сигнал на виході об'єкта; e – сигнал помилки; $\hat{u}$ – оцінка сигналу управління (сигнал на виході нейронної мережі)

З рис. 2.3 видно, що схема прямого інверсного управління є розімкненою. Отже, вона прийнятна лише для управління стійкими стаціонарними об'єктами за відсутності відчутних збурювань.

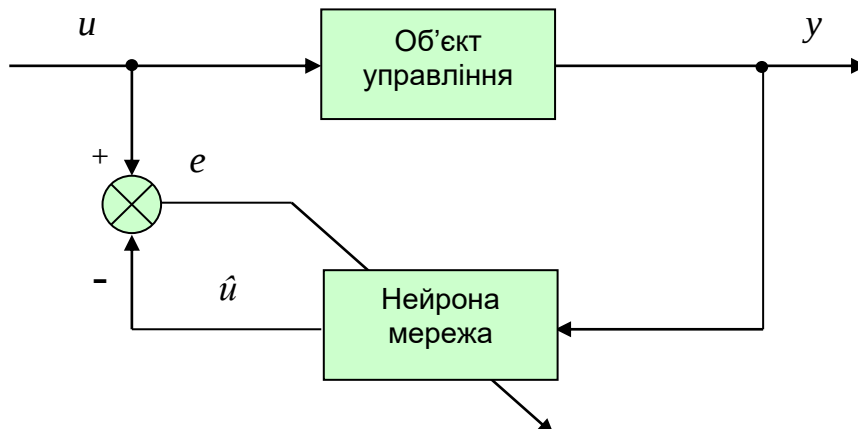


Рис. 2.3. Схема оперативного переналаштування нейроконтролера

Її головним достоїнством є простота реалізації й відсутність проблем зі стійкістю. Проте, для забезпечення достатньої якості управління нейромережний контролер має достатньо точно описувати інверсну (зворотну) динаміку об'єкта.

2.2. Пряме адитивне управління

Пряме *адитивне управління* (*additive feed-forward control*, AFFC) полягає в паралельному підключенні до незадовільно працюючого класичного контролера додаткового нейромережного контролера (рис. 2.4), який є навченим інверсній динаміці об'єкта у відповідності до схеми на рис. 2.2.

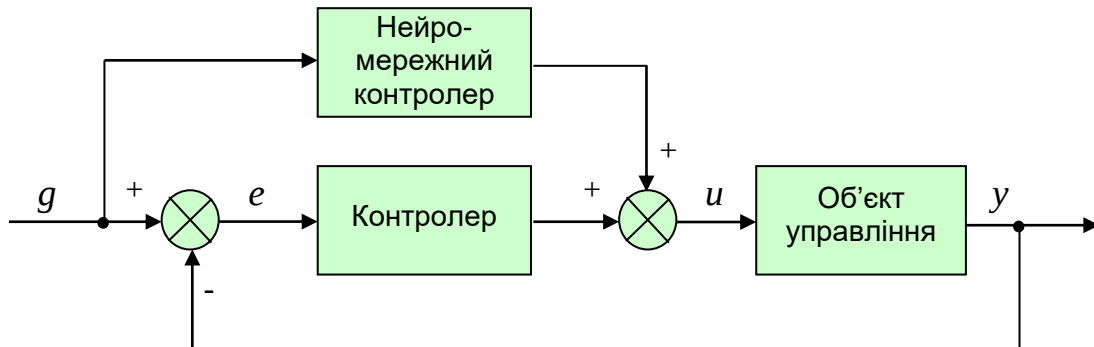


Рис. 2.4. Схема прямого адитивного управління:

g – сигнал завдання; e – сигнал помилки; u – сигнал управління;
 y – сигнал на виході об'єкта

У даній схемі використовується ідея так званого комбінованого управління: поряд з регулюванням за відхиленням, що його реалізує незадовільно працюючий класичний контролер, здійснюється регулювання за завданням, яке реалізує нейромережний контролер.

Покажемо, що в розглянутій системі управління (рис.2.4) вихід об'єкта управління дорівнює завданню й помилка регулювання дорівнює нулю, якщо нейромережний контролер точно описує інверсну динаміку об'єкта.

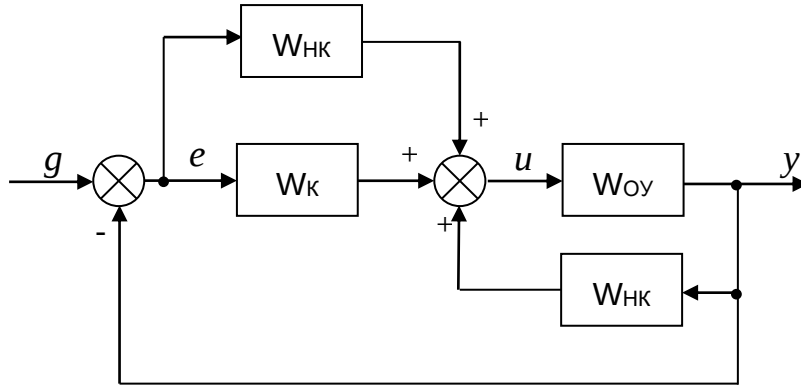
Нехай об'єкт управління і нейромережний контролер, як і раніше, описуються функціями передачі $W_{ov}(p)$ і $W_{нк}(p)$ відповідно, а функція передачі класичного контролера – $W_k(p)$.

Перенесемо вхід нейромережного контролера на схемі рис. 2.4 через суматор. При цьому схема матиме вигляд, що наведений на рис. 2.5.

Запишемо функцію передачі системи управління:

$$W_c(p) = \frac{W_{нк}(p) \cdot W_{ov}(p) + W_k(p) \cdot W_{ov}(p)}{1 + W_k(p) \cdot W_{ov}(p)} \quad (2.1)$$

Рис. 2.5. Перетворена схема прямого адитивного управління



Як відомо, вихід об'єкта управління точно відтворює сигнал завдання (тобто $y = g$), якщо функція передачі замкнутої системи за завданням дорівнює одиниці. Очевидно, що функція передачі (2.1) дорівнює 1 за умови

$$W_{HK}(p) \cdot W_{OY}(p) = 1 \quad \text{або} \quad W_{HK}(p) = 1 / W_{OY}(p).$$

Отже, нейромережний контролер повинен відтворювати інверсну модель об'єкта управління, що й було потрібно показати.

Достоїнство схеми прямого адитивного управління полягає у тому, що дані для навчання нейромережного контролера можуть бути отримані без розмикання існуючого зворотного зв'язку. Крім того, наявність у схемі (рис. 2.4) замкнутого контуру регулювання за відхиленням уможливорює управління нестійким об'єктом.

2.3. Управління зі внутрішньою моделлю

Пряме адитивне нейромережне управління так саме, як і пряме інверсне управління, на відміну від класичного ПІД-управління, не дозволяє управляти нестационарними об'єктами. Для подібних об'єктів використовують управління **зі внутрішньою моделлю** (*internal model control*, ІМС). Таке управління добре зарекомендувало себе в умовах дії неконтрольованих збурювань.

Схема управління зі внутрішньою моделлю включає нейромережний контролер, нейромережну модель об'єкта та робастний фільтр (рис.2.6).

Нейромережний контролер, як правило, являє собою інверсну модель об'єкта управління й включається з ним послідовно. Нейромережна модель об'єкта включається паралельно реальному об'єкту й використовується для оцінки збурювань на його виході. В якості робастного фільтру зазвичай застосовують ланку першого порядку, постійна часу якої обирається таким

чином, щоб забезпечити стійкість замкнутої схеми щодо неконтрольованих збурювань.

Розглянемо роботу схеми нейромережного управління зі внутрішньою моделлю. Керуючий сигнал u , що сформований нейромережним контролером, надходить на вхід об'єкта управління та на вхід його нейромережної моделі. Вихід об'єкта y порівнюється із виходом моделі $\hat{y}$, а сигнал помилки e подається на вхід суматора, де він віднімається від сигналу завдання g . Скоригований й відфільтрований сигнал завдання подається на вхід нейромережного контролера, який формує сигнал управління u .

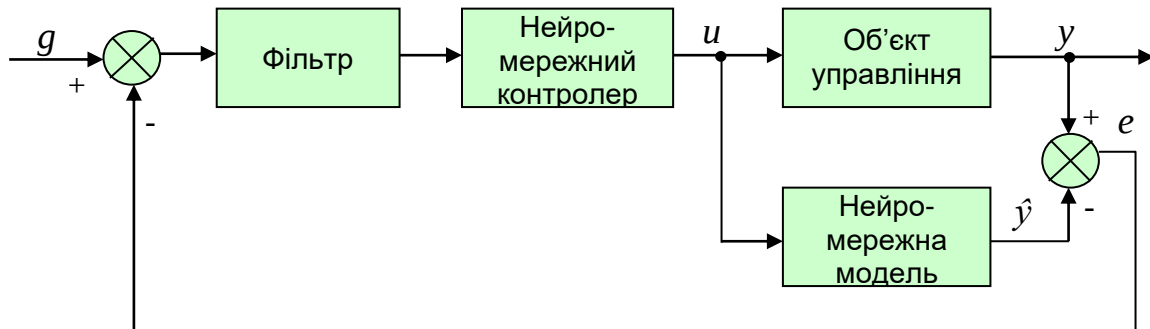


Рис. 2.6. Схема управління зі внутрішньою моделлю:

g – сигнал завдання; u – сигнал управління; y – сигнал на виході об'єкта; $\hat{y}$ – оцінка виходу об'єкта; e – сигнал помилки між виходом об'єкта й моделі

Якщо на об'єкт управління не діють неконтрольовані збурювання, помилка e є мірою невідповідності моделі об'єкту. У тому ж випадку, якщо нейромережна модель є точною моделлю об'єкта управління, помилка e відповідає реакції об'єкта на невідоме (неконтрольоване) збурювання.

Таким чином, сигнал помилки між виходом об'єкта управління й виходом його моделі може розцінюватися як додаткова інформація, що відсутня в моделі, й використовуватись для покращення якості управління. Теоретично, використовуючи схему управління зі внутрішньою моделлю, можна досягти ідеального управління.

Запровадимо наступні позначення: $W_{НК}(p)$ – функція передачі нейромережного контролера; $W_{ОУ}(p)$ – функція передачі об'єкта управління; $W_{НМ}(p)$ – функція передачі нейромережної моделі об'єкта. Перетворимо схему рис. 2.6 до вигляду на рис. 2.7.

Тоді функція передачі системи управління зі внутрішньою моделлю, що наведена на рис. 2.6, без урахування фільтра може бути записана у такий спосіб

$$W_C(p) = \frac{W_{OY}(p) \cdot W_{HK}(p)}{1 - W_{HM}(p) \cdot W_{HK}(p) + W_{OY}(p) \cdot W_{HK}(p)}. \quad (2.2)$$

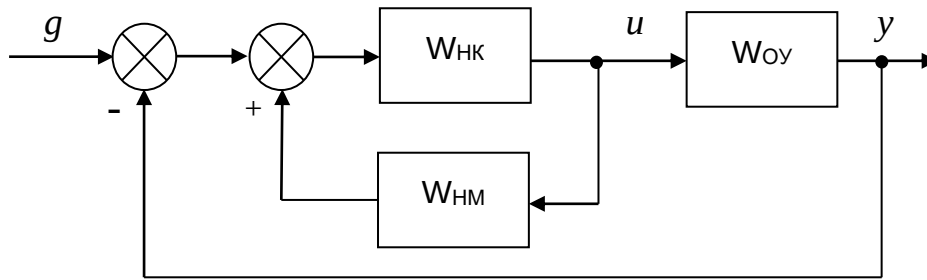


Рис. 2.7. Перетворена схема управління зі внутрішньою моделлю

Для того, щоб вихідний сигнал об'єкта точно відтворював сигнал завдання, функція передачі (2.2) має дорівнювати «1». Це стає можливим за умови

$$W_{HK}(p) \cdot W_{HM}(p) = 1 \quad \text{або} \quad W_{HK}(p) = 1 / W_{HM}(p),$$

тобто коли нейромережний контролер є точною інверсією прямої нейромережної моделі об'єкта управління. У цьому випадку матиме місце ідеальне відстеження сигналу завдання ($y = g$) й усунення впливу збурювання.

Очевидно, що для одержання високої якості управління достатньо, щоб нейроконтролер точно описував інверсну динаміку прямої нейромережної моделі об'єкта управління, а не власне об'єкта управління.

Детальніше ознайомитись з особливостями застосування нейроконтролерів в системах автоматичного управління можна під час виконання практикуму (див. розд. 3.4).

Контрольні запитання та завдання

1. Опишіть ідею прямого інверсного управління. Які обмеження існують для її застосування?
2. В чому полягає навчання нейрорегулятора зворотній динаміці об'єкта управління?
3. Доведіть, що пряме адитивне управління є ефективним за умови точного відтворення нейрорегулятором зворотної динаміки об'єкта.
4. Які переваги схеми управління зі внутрішньою моделлю перед прямим адитивним навчанням?

3. ПРАКТИКУМ «НЕЙРОННІ МЕРЕЖІ В АВТОМАТИЗАЦІЇ»

3.1. Ідентифікація статичних лінійних об'єктів за допомогою нейронних мереж

Мета роботи:

1. Опанування основних прийомів роботи з GUI-інтерфейсом NNTool ППП Neural Network Toolbox (NNT) – MATLAB.

2. Створення, навчання й застосування одношарової нейронної мережі для ідентифікації лінійного статичного об'єкта. Дослідження точності нейронної моделі.

3.1.1. Вихідні дані

Застосування нейронних мереж для ідентифікації статичних лінійних об'єктів розглянемо на прикладі об'єкта, що описується алгебраїчним рівнянням

$$y = a_1x_1 + a_2x_2 + a_3x_3, \quad (3.1)$$

де y – вихід об'єкта; x_1 , x_2 , x_3 – його входи, а a_1 , a_2 , a_3 – постійні коефіцієнти. Параметри об'єктів, які необхідні для індивідуального виконання роботи, наведені у таблиці 3.1.

Таблиця 3.1

Вихідні дані для виконання роботи

№ варіанту	a_1	a_2	a_3
1	1,34	2,85	6,93
2	0,66	7,19	-2,51
3	3,98	-2,42	5,03
4	9,01	-3,23	-4,17
5	1,40	5,47	2,31
6	1,34	-2,85	0,32
7	2,13	1,15	3,38

№ варіанту	a_1	a_2	a_3
8	8,22	1,65	-4,14
9	8,83	1,69	3,11
10	-1,34	2,85	9,63
11	2,13	-1,15	3,38
12	8,83	1,69	-3,11

3.1.2. Формування навчальної множини

Для вирішення задачі ідентифікації, насамперед, необхідно сформулювати навчальну множину, що складається зі значень вхідних і вихідних сигналів об'єкта.

На практиці для навчання нейронної мережі використовуються дані, які отримують під час проведення експериментів на реальному об'єкті. На вхід об'єкта подають відомі вхідні сигнали й вимірюють відповідні вихідні сигнали об'єкта. Частку отриманих у такий спосіб даних використовують для навчання нейронної мережі, а частку – резервують для її тестування по закінченні навчання.

Оскільки в лабораторній роботі як об'єкт розглядається математична модель з відомим рівнянням, наприклад,

$$y = 1,34x_1 + 2,85x_2 + 1,63x_3,$$

навчальну множину можна сформулювати в системі структурного моделювання Simulink, використовуючи схему, що наведена на рис. 3.1.

Як вхідні сигнали можна використати, наприклад, синусоїдальні сигнали одиничної амплітуди з різною частотою та різними фазовими зсувами. Ці сигнали моделюються за допомогою блоків *Sine Wave*, *Sine Wave1* та *Sine Wave2*.

Сигнали на вході й виході об'єкта варто записати до робочої області системи MATLAB за допомогою блоків *To Workspace* і *To Workspace1* відповідно.

Перед моделюванням у відповідних діалогових вікнах необхідно задати параметри усіх блоків.

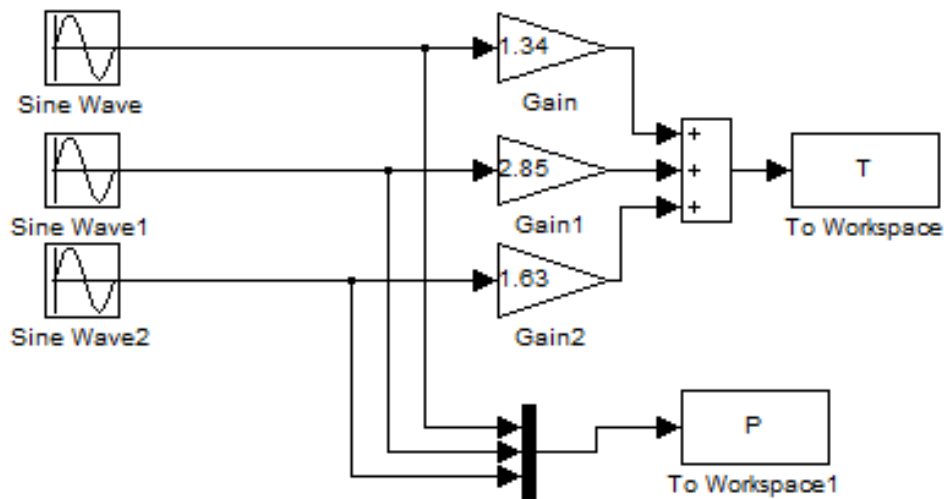


Рис. 3.1. Схема для формування навчальних даних

Для блоків *To Workspace* та *To Workspace1* потрібно змінити задані за замовчуванням імена змінних *simout* та *simout1* на *P* та *T* відповідно. Крок дискретизації (*sample time*) слід установити рівним 0.2, що відповідатиме фіксації сигналів, що надходять на вхід блоку, через кожні 0,2 с. Форматом, у якому будуть записуватись дані (*save format*), слід обрати масив (*array*). Значення інших параметрів залишити такими, що задаються за замовчуванням.

Для блоків *Sine Wave*, *Sine Wave1*, *Sine Wave2* значення частоти (*frequency*) і фази (*phase*) призначити, наприклад, такими: 5 та 0; 1 та 0.8; 0.5 та 0.2 відповідно. Інші параметри цих блоків залишити без зміни.

Тривалість моделювання можна обмежити 10 секундами. По закінченні моделювання в робочу область системи MATLAB будуть виведені два числові масиви формату *double* *P* и *T* розміром 51×3 і 51×1 відповідно: *P* – масив значень вхідних сигналів об'єкта; *T* – масив значень вихідного (цільового) сигналу.

Для подальшого використання в ППП Neural Network Toolbox (NNT) масиви *P* и *T* необхідно перетворити, застосувавши операцію транспонування:

```
>> P = P'; T = T';
```

У результаті будуть отримані масиви *P* и *T* розміром 3×51 і 1×51, які вже можна використовувати для навчання нейронних мереж.

На практиці часто виникає необхідність у масштабуванні масивів даних, тобто в приведенні їхніх елементів до певного діапазону значень, про що йшлося у п. 1.5.5 [6]. Як правило, значення елементів масивів зводять до

інтервалу $[-1 \ 1]$. У ППП NNT є М-функції, що дозволяють здійснювати таке перетворення [5, С.96-98, С.410-414].

Ми ж для спрощення, щоб не вдаватися до процедури масштабування, вважатимемо, що вхідні сигнали розглянутого об'єкта перебувають в межах від -1 до 1.

3.1.3. Створення й навчання нейронної мережі

Оскільки статичний лінійний об'єкт, що описується рівнянням (3.1), має три входи й один вихід, для побудови його моделі скористаємось статичною лінійною одношаровою нейронною мережею, що складається всього з одного нейрона, вихід якої у описується рівнянням

$$y = w_1x_1 + w_2x_2 + w_3x_3 + b, \quad (3.2)$$

де x_1, x_2, x_3 – входи мережі; w_1, w_2, w_3 – вагові коефіцієнти; b – зсув.

Структурна схема такої нейронної мережі наведена на рис. 3.2.

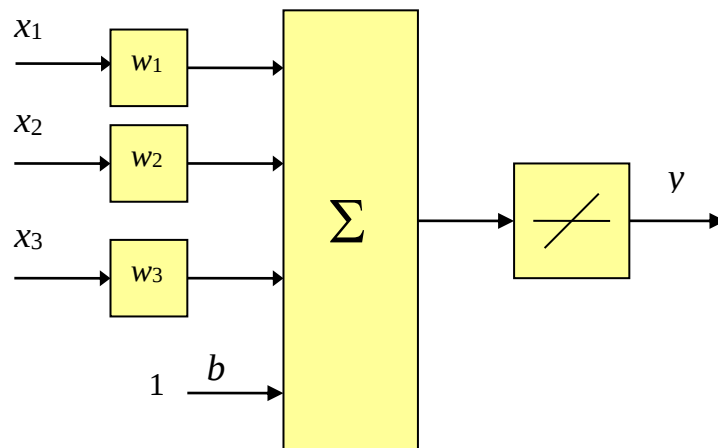


Рис. 3.2. Структура лінійної одношарової мережі, що містить один нейрон

Для створення й навчання лінійної одношарової нейронної мережі скористаємось GUI-інтерфейсом NNTool, що активується в результаті виконання однойменної команди в командному вікні системи MATLAB:

```
>> nntool
```

Щоб створити лінійний шар, у головному вікні, що відкрилося (рис. 3.3), NNTool Network/Data Manager /Керування мережею/даними необхідно натиснути кнопку New ... / Нова мережа У діалоговому вікні

Create New Network / *Створити нову мережу* вибрати у полі Network type / *Тип мережі* Linear layer (train) / *Лінійний шар (навчання)*.

Після цього задати наступні параметри лінійного шару: діапазони вектора входу (*input ranges*): [-1 1; -1 1; -1 1]; кількість нейронів (*number of neurons*): 1; швидкість навчання (*learning rate*): 0.01. Ім'я нейронної мережі *network1* залишити без зміни (рис. 3.4).

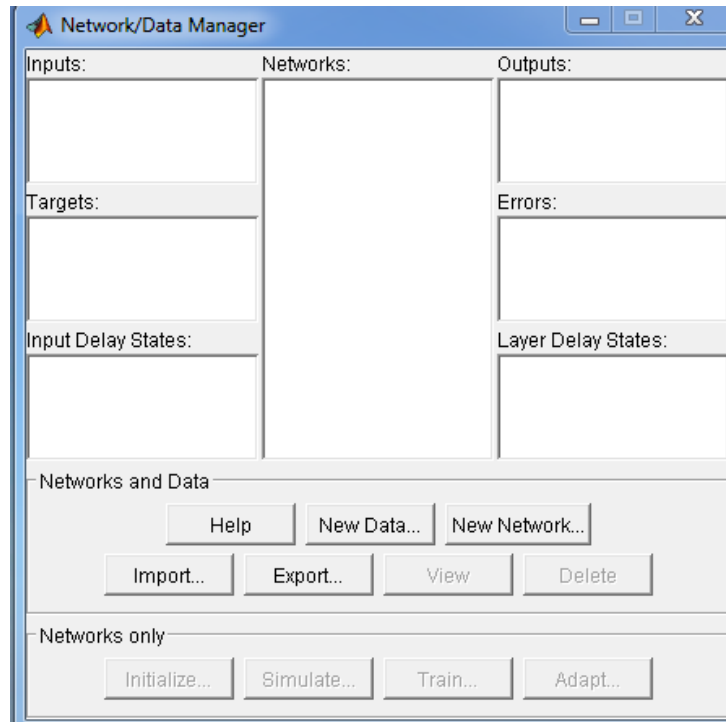


Рис. 3.3. GUI-інтерфейсом NNTool

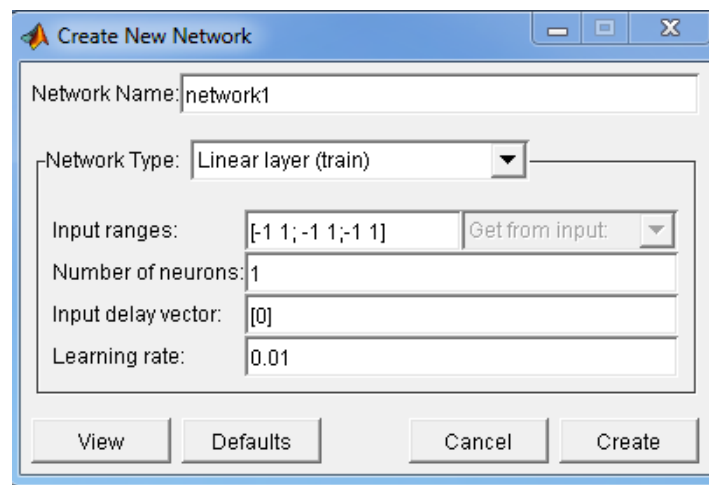


Рис. 3.4. Створення лінійної одношарової нейронної мережі з одним нейроном

Перш ніж приступитися до навчання нейронної мережі, потрібно імпортувати з робочої області MATLAB навчальні послідовності даних. Для цього можна скористатись вікном Import or Load to Network/Data Manager, що викликається кнопкою Import / *Імпортувати* вікна Network/Data Manager (рис. 3.5).

Як джерело (*source*), з якого будуть імпортуватися дані, необхідно вказати робочу область, вибравши Import from MATLAB workspace. Змінну P імпортувати як послідовність входів (*import as inputs*), а змінну T – як послідовність цілей (*import as targets*).

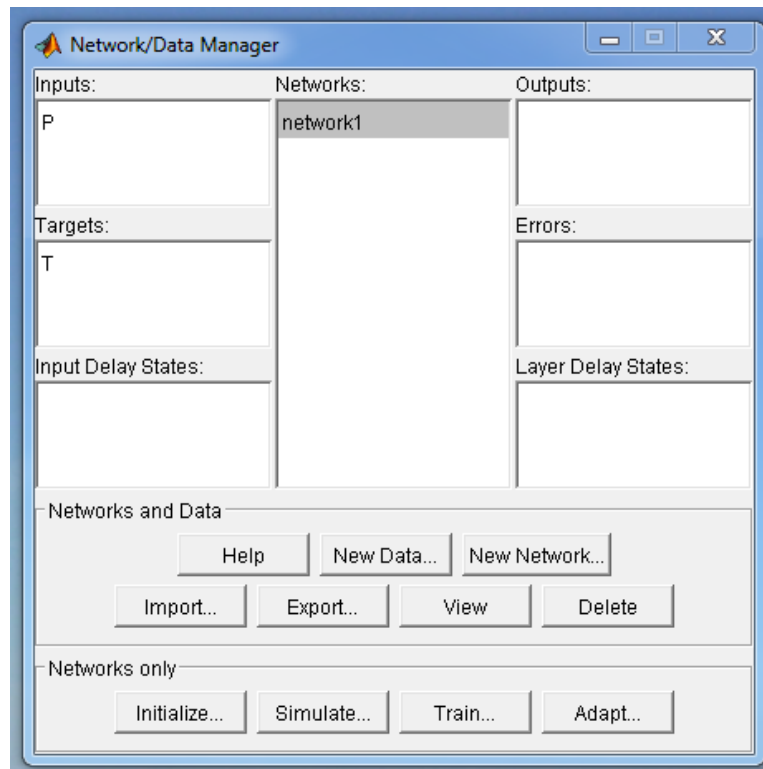


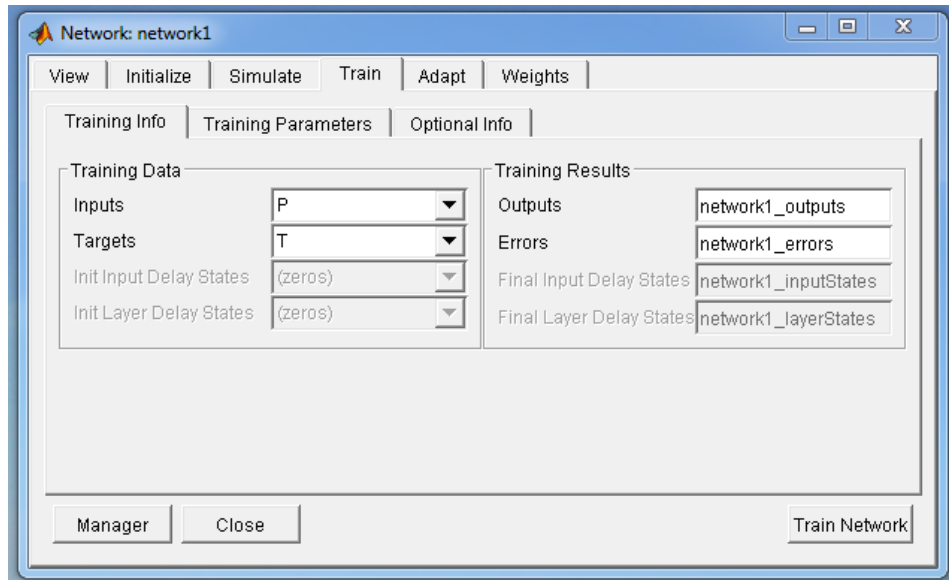
Рис. 3.5. Імпортування навчальні послідовності даних з робочої області MATLAB

Зазвичай перед навчанням виконують ініціалізацію нейронної мережі, тобто привласнюють її ваговим коефіцієнтам деякі невеликі випадкові початкові значення й тим самим задають початковий стан мережі. Зауважимо, що в GUI-інтерфейсі NNTool при ініціалізації лінійного шару усім вагам за замовчуванням привласнюються нульові початкові значення.

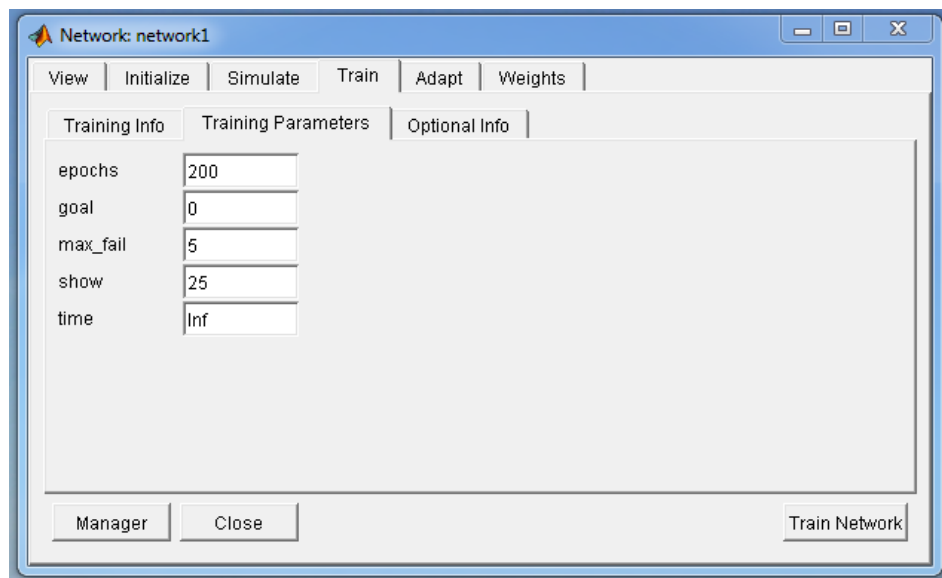
Щоб навчити нейронну мережу, необхідно вибрати її й натиснути кнопку Train / *Навчити*. У результаті відкривається діалогова панель із трьома вкладками: Training Info / *Інформація про навчальні послідовності*; Training Parameters / *Параметри навчання*; Optional Info / *Додаткова інформація* (рис.3.6,а).

На закладці Training Info зі спадаючих меню Inputs (*Входи*) і Targets (*Цілі*) вибираємо послідовності *P* и *T* відповідно.

На закладці Training Parameters (рис. 3.6,б) задаємо граничну кількість епох навчання (*epochs*), наприклад, 200, а значення інших параметрів залишаємо без зміни.



а



б

Рис. 3.6. Визначення параметрів навчання мережі

Далі виконуємо навчання нейронної мережі, натиснувши кнопку *Train Network*. По закінченні навчання у вікні Training with TRAINS (рис. 3.7) можна побачити графік змінення помилки навчання за епохами, а на закладці Weights (рис. 3.8) – значення вагових коефіцієнтів мережі, що

встановились за результатами навчання. Як впливає з рис. 3.8, кінцева помилка навчання мережі майже відсутня. Після 145 епох вона набула зневажливо малого значення $5,38171 \cdot 10^{-31}$. При цьому синаптичні вагові коефіцієнти навченої мережі (див. рис. 3.8) набули значень, що точно дорівнюють відповідним коефіцієнтам рівняння об'єкта (3.1). Така висока відповідність між коефіцієнтами пояснюється повною відповідністю структури використаної для ідентифікації нейронної мережі структурі об'єкта.

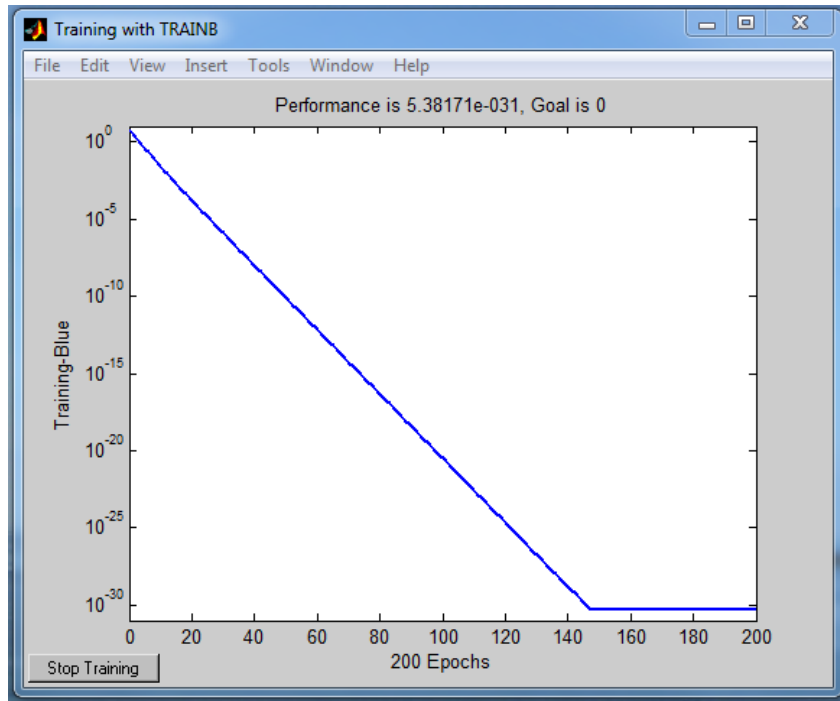


Рис. 3.7. Змінення помилки мережі впродовж навчання

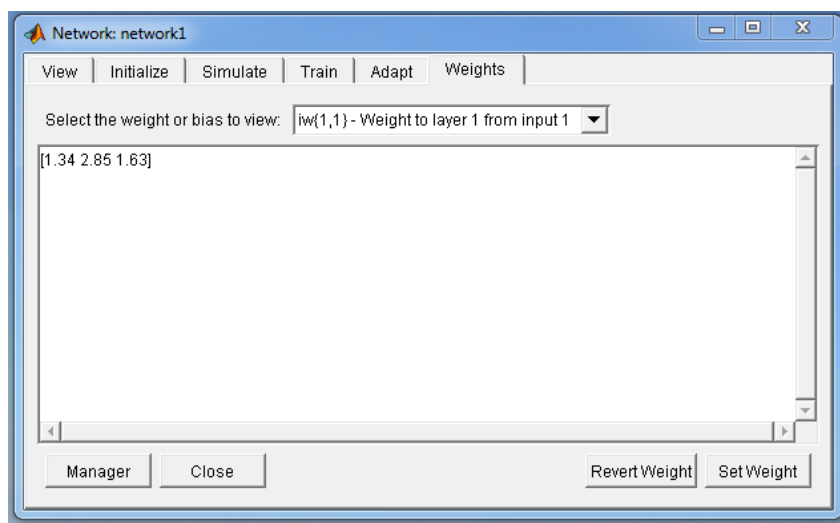


Рис. 3.8. Вагові коефіцієнти навченої мережі

3.1.4. Тестування нейромережі й аналіз точності ідентифікації

Для тестування навченої нейронної мережі *network1*, потрібно сформувати її Simulink-модель (*S-модель*). З огляду на те, що нейронна мережа була створена в середовищі NNTool, її необхідно попередньо експортувати до робочої області MATLAB.

Для цього в головному вікні NNTool слід натиснути кнопку *Export/Експортувати*. Потім у діалоговому вікні, що з'явилося, *Export or Save from Network/Data Manager* вибрати *мережа network1* і знову натиснути кнопку *Export*. У результаті мережа буде скопійована в робочу область MATLAB.

Для формування *S-моделі* нейронної мережі в ППП NNT є М-функція *gensim (net, TS)*, де *net* – ім'я мережі; *TS* – крок дискретизації, що задається для моделювання динамічних нейронних мереж.

Оскільки наша нейронна мережа *network1* є статичною, при створенні її *S-моделі* параметру *TS* потрібно надати значення -1 , що відповідає невизначеному такту дискретності:

```
>> gensim (network1, -1)
```

Для тестування отриманої нейромережної моделі необхідно зібрати схему, що наведена на рис. 3.9.

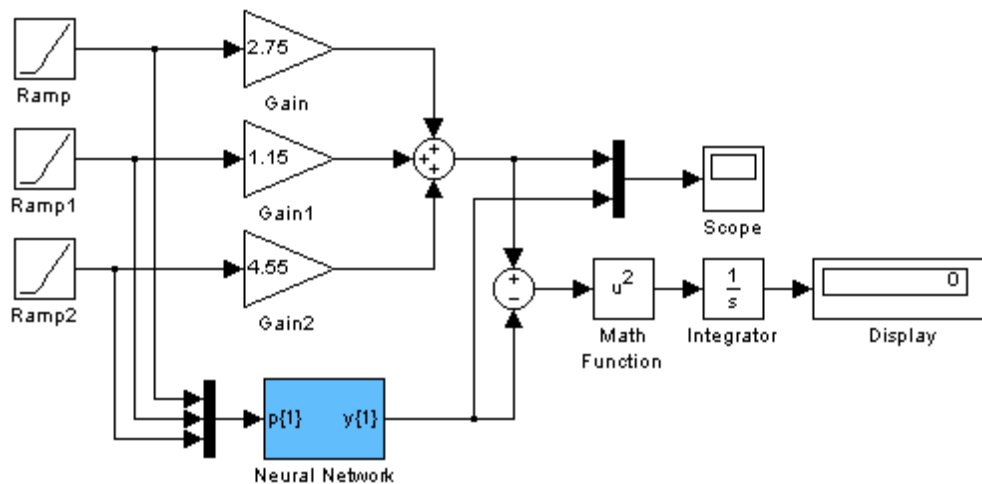


Рис. 3.9. Схема тестування нейромережної моделі статичного лінійного об'єкта

На входи нейромережної моделі та об'єкта слід подати сигнали, які не використовувались під час навчання. Такими сигналами можуть бути, наприклад, сигнали з виходів блоків *Ramp*. Для оцінювання ступеня

адекватності отриманої моделі об'єкту будемо обчислювати інтегральну квадратичну помилку.

Параметри блоків Ramp задамо, наприклад, у такий спосіб:

Ramp: slope = 0.075; start time = 0; initial output = 0.15;

Ramp1: slope = 0.06; start time = 0; initial output = 0.5;

Ramp2: slope = 0.025; start time = 0; initial output = 0.1.

Тривалість моделювання встановимо – 10 с.

Результати моделювання слід вивести на осцилограф (рис. 3.10) й оцінити за інтегральним критерієм адекватності моделі та точність ідентифікації.

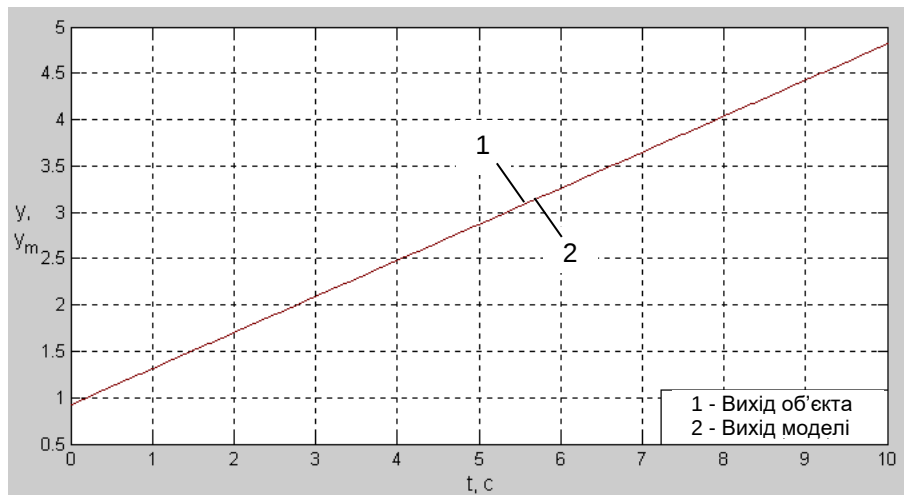


Рис. 3.10. Результати ідентифікації статичного лінійного об'єкта (графіки з виходу об'єкта і нейромережної моделі на рисунку співпадають)

3.2. Ідентифікація статичних нелінійних об'єктів за допомогою нейронних мереж

Мета роботи:

1. Створити, навчити й застосувати двошарову нейронну мережу для ідентифікації нелінійного статичного об'єкта.
2. Дослідити вплив параметрів мережі на точність нейронної моделі.

3.2.1. Вихідні дані

Застосування нейронних мереж для ідентифікації статичних нелінійних об'єктів розглянемо на прикладі об'єкта, зв'язок між виходом y та входом x якого описується рівнянням

$$y = b_1x^3 + b_2x^2 + b_3x, \quad (3.3)$$

де y – вихід об'єкта; x – його вхід, а b_1 , b_2 та b_3 – постійні коефіцієнти. Параметри об'єктів, які необхідно використати при індивідуальному виконанні лабораторної роботи, наведені в табл. 3.2.

Таблиця 3.2

Вихідні дані для виконання роботи

№ вар.	b_1	b_2	b_3
1	3,81	-2,16	4,32
2	1,14	5,26	2,04
3	5,22	3,78	-6,19
4	2,44	-1,01	7,14
5	3,56	2,18	4,03
6	1,95	2,14	3,55
7	-1,95	2,14	3,55
8	4,23	2,87	1,45
9	2,12	-1,44	-1,75
10	4,30	1,06	4,36
11	2,63	1,25	4,74
12	7,11	5,31	-4,00

Для ідентифікації таких об'єктів можна використати нейронні мережі прямого розповсюдження, що навчають методом зворотного розповсюдження помилки та радіальних базисних нейронних мереж.

3.2.2. Формування навчальної множини

Нехай, наприклад, нелінійний об'єкт описується рівнянням

$$y = 1,95x^3 + 2,75x^2 + 1,25x. \quad (3.4)$$

Насамперед, необхідно сформувати навчальні послідовності даних. Із цією метою в ППП Simulink можна зібрати схему, що наведена на рис. 3.11.

Як вхідний сигнал можна використати, наприклад, сигнал, що лінійно зростає і моделюється за допомогою блоку *Ramp*. Припускаючи, що вхідний сигнал об'єкта може змінюватись у межах від «0» до «1», а тривалість моделювання дорівнює 10 с, задамо такі параметри блоку *Ramp*: *slope* = 0.1; *start time* = 0; *initial output* = 0.

За допомогою блоків *To Workspace* та *To Workspace1* записати до робочої області MATLAB значення вхідного й вихідного сигналів об'єкта з кроком 0,2 с у вигляді масивів.

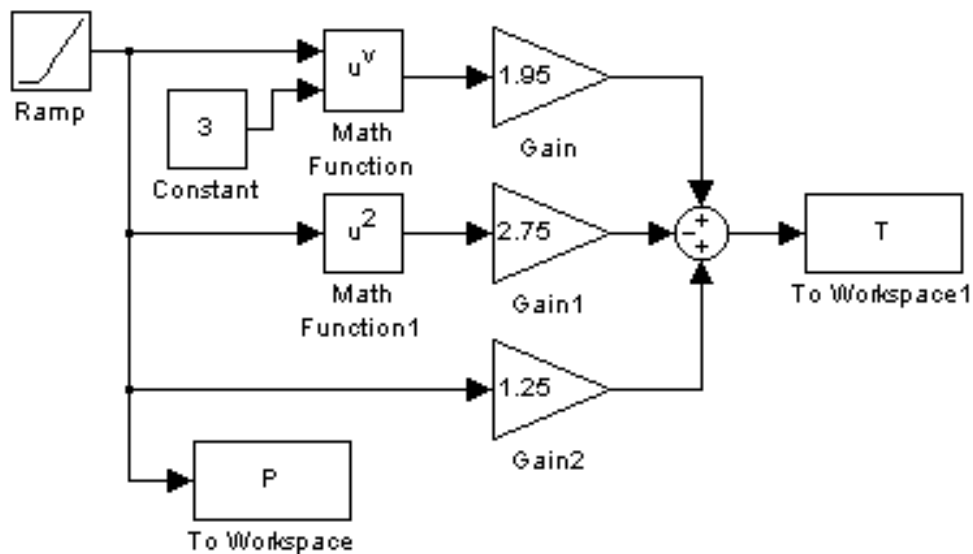


Рис. 3.11. Схема для формування навчальних даних

По закінченні моделювання в робочій області MATLAB будуть створені масиви даних *P* і *T*, які необхідно перетворити у такий спосіб:

```
>> P = P', T = T'.
```

Спочатку вирішимо задачу ідентифікації статичного нелінійного об'єкта, що описується рівнянням (3.3), за допомогою багатошарової нейронної мережі прямого розповсюдження

3.2.3. Ідентифікація за допомогою багатошарової нейронної мережі прямого розповсюдження

Багатошарова нейронна мережа прямого розповсюдження (multi-layer feed-forward network), або *багатошаровий перцептрон* (multi-layered

perceptron, MLP) являє собою нейронну мережу, що не має зворотних зв'язків, сигнали в якій передаються тільки в прямому напрямку: від входу мережі до її виходу.

Багатошарові нейронні мережі прямого розповсюдження мають шар вхідних нейронів, щонайменше один шар прихованих нейронів та вихідний шар. При визначенні кількості шарів вхідний шар зазвичай не враховується.

Структурна схема двошарової нейронної мережі прямого розповсюдження наведена на рис. 3.12.

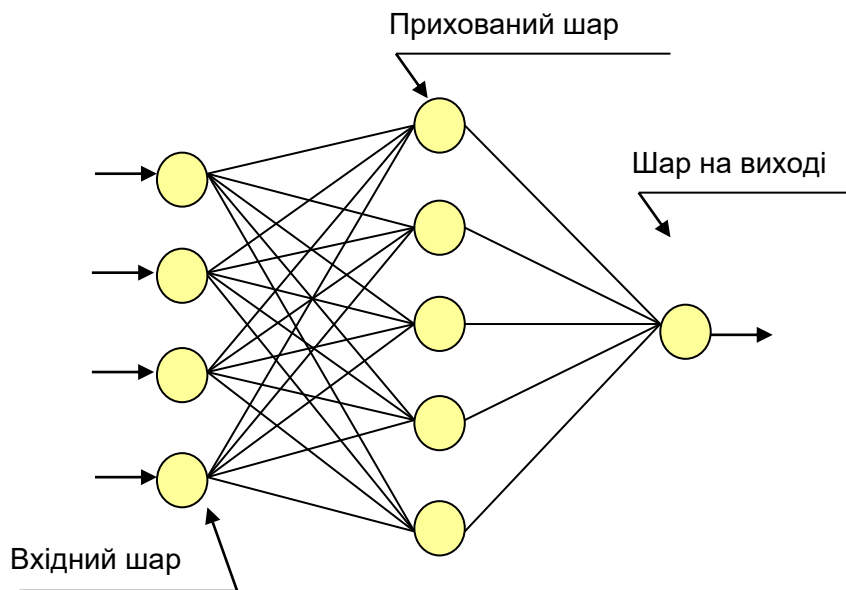


Рис. 3.12. Структурна схема двошарової нейронної мережі прямого розповсюдження

Як правило, для прихованих нейронів використовують сигмоїдальні або гіперболічні тангенціальні функції активації, а для нейронів вихідного шару – лінійні. У цьому випадку виходи мережі можуть набувати довільних значень.

Якщо з якихось причин необхідно обмежити значення виходів мережі, як функції активації нейронів вихідного шару варто застосовувати сигмоїдальні або гіперболічні тангенціальні функції, що дозволяють одержувати значення виходів мережі в межах від 0 до 1 та від -1 до 1 відповідно.

Для налаштування ваг у багатошарових мережах прямого розповсюдження зазвичай застосовують алгоритми, що реалізують метод зворотного розповсюдження помилки (*error backpropagation method*).

Для навчання багат шарових мереж невеликого й середнього розмірів у літературі рекомендується користатись алгоритмом Левенберга-Марквардта, або методом Ньютона, а для навчання великих мереж – алгоритмом методу сполучених градієнтів [5].

Багат шарові нейронні мережі з нелінійними функціями активації прихованих нейронів, вважаються універсальними апроксиматорами, що здатні з достатньою точністю апроксимувати складні нелінійні функції. Ця властивість обумовлює можливість їхнього застосування для ідентифікації статичних нелінійних об'єктів.

Перш ніж приступати до побудови дво шарових нейронних мереж прямого розповсюдження, імпортуємо з робочої області MATLAB до середовища NNTool сформовані раніше навчальні послідовності P та T .

У нашому випадку для ідентифікації статичного нелінійного об'єкта, що описується рівнянням (3.3), будемо використовувати дво шарові нейронні мережі прямого розповсюдження. Створювати їх будемо за допомогою GUI-інтерфейсу NNTool. Нейронним мережам даного типу в NNTool відповідає мережа *feed-forward backprop*, що створюється за замовчуванням.

Для дослідження впливу кількості нейронів у прихованому шарі на тривалість навчання та на якість ідентифікації, сформуємо кілька подібних нейронних мереж прямого розповсюдження з різною кількістю нейронів у прихованому шарі.

При створенні нейронних мереж типу *feed-forward backprop* діапазон входів будемо задавати таким, що становить $[0 \ 1]$. Як функцію навчання (*training function*) будемо використовувати М-функцію *trainlm*, що реалізує алгоритм Левенберга-Марквардта LM , а як функцію оцінки якості навчання (*performance function*) – *mse* (критерій середньої квадратичної помилки). Кількість шарів (*number of layers*) призначаємо такою, що дорівнює 2.

На панелі *Properties for* необхідно задати властивості шарів: кількість нейронів у шарі (*number of neurons*) та функцію активації шару (*transfer function*).

Для нейронів першого (прихованого) шару вибираємо логістичну сигмоїдальну функцію активації *logsig*, а для нейронів другого (вихідного) шару – лінійну функцію *purelin*. Оскільки об'єкт (3.3) має тільки один вихід, то кількість нейронів у вихідному шарі дорівнюватиме 1.

Усі параметри навчання, за винятком граничної кількості епох (*epochs*), залишаємо тими ж, що задані за замовчуванням. Значення параметра *epochs*

будемо підбирати в процесі навчання відповідно до характеру змінення критерію якості навчання (mse).

Навчені нейронні мережі експортуємо в робочу область MATLAB й формуємо їхні S-моделі.

Тестування отриманих нейронних мереж здійснюємо за допомогою схеми, яка наведена на рис. 3.12.

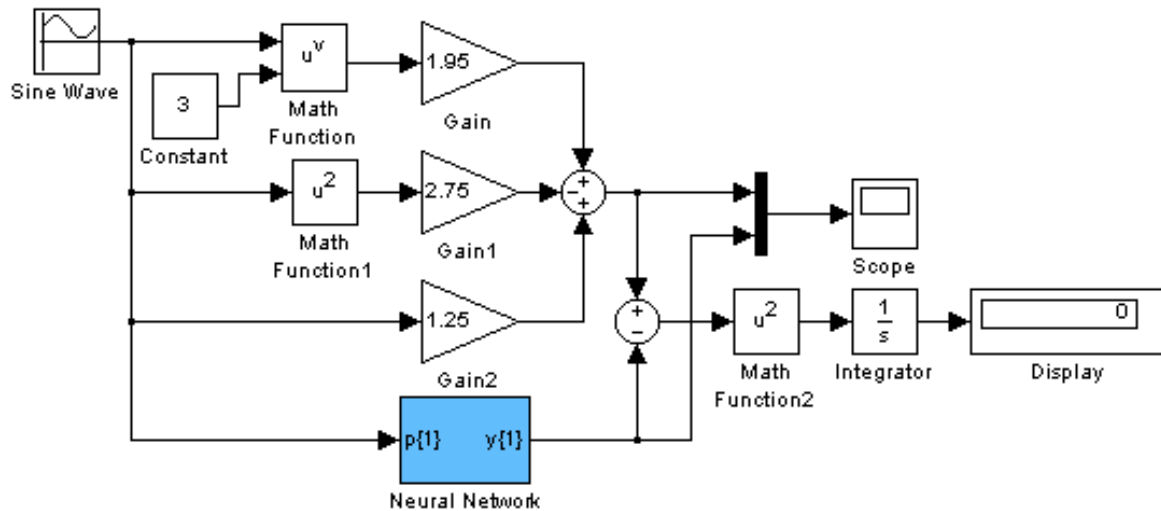


Рис. 3.12. Схема для тестування нейромережних моделей статичного нелінійного об'єкта

Таблиця 3.3

Результати навчання та тестування двошарових нейронних мереж прямого розповсюдження

Ім'я мережі	Кількість нейронів у прихованому шарі	Кількість епох (<i>epochs</i>)	Критерій оцінки якості навчання (<i>MSE</i>)	Інтегральний квадратичний критерій
1	2	3	4	5
net1	1	100		
		2000		
		6000		
net2	2	100		
		2000		
		6000		

Продовження табл. 3.3

1	2	3	4	5
net3	3	100		
		2000		
		6000		
net4	4	100		
		2000		
		6000		

За результатами навчання й тестування необхідно заповнити табл. 3.3, навівши у звіті відповідні графіки як, наприклад, на рис. 3.13.

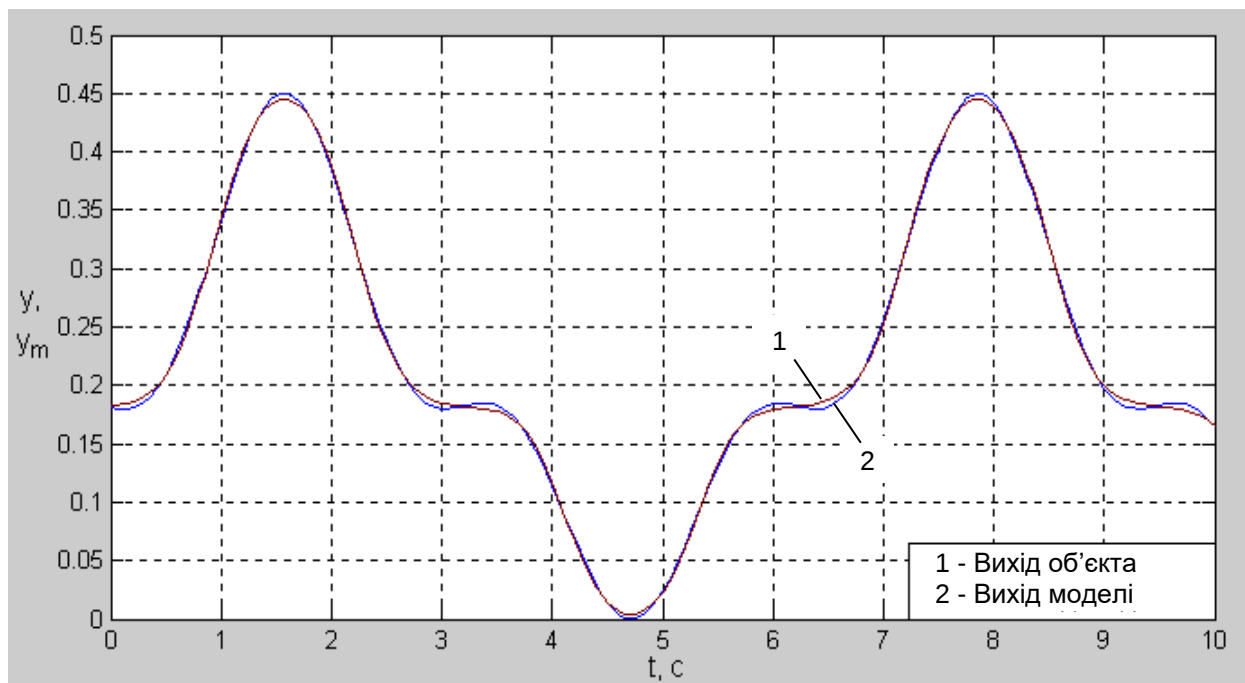


Рис. 3.13. Результати ідентифікації статичного нелінійного об'єкта за допомогою двошарової нейронної мережі прямого розповсюдження (варіант, за яким з кількістю нейронів у прихованому шарі дорівнює 2)

За результатами аналізу даних, що наведені у табл. 3.3, необхідно зробити висновок щодо впливу кількості нейронів у прихованому шарі на точність нейромережної моделі та швидкість її навчання.

Відзначимо, що, хоча деякі дослідники й переконані, що підбирати оптимальну кількість нейронів у прихованому шарі треба винятково

експериментальним шляхом, на практиці досить часто для її визначення використовують теорему Хехт-Колмогорова (Hecht - Kolmogorov's theorem).

Теорема Хехт-Колмогорова стверджує, що оптимальна кількість нейронів у прихованому шарі дорівнює подвоєній кількості вхідних нейронів плюс один [1]. Таким чином, згідно з даною теоремою для ідентифікації статичних нелінійних об'єктів з одним входом необхідно застосовувати двошарові нейронні мережі із трьома нейронами у прихованому шарі.

3.2.4. Ідентифікація за допомогою радіальної базисної нейронної мережі

Радіальна базисна нейронна мережа (Radial Basis Function network, RBF-мережа) являє собою двошарову нейронну мережу, що складається з прихованого шару радіальних базисних нейронів та вихідного лінійного шару (рис. 3.14) [5, С. 131-140; 6].

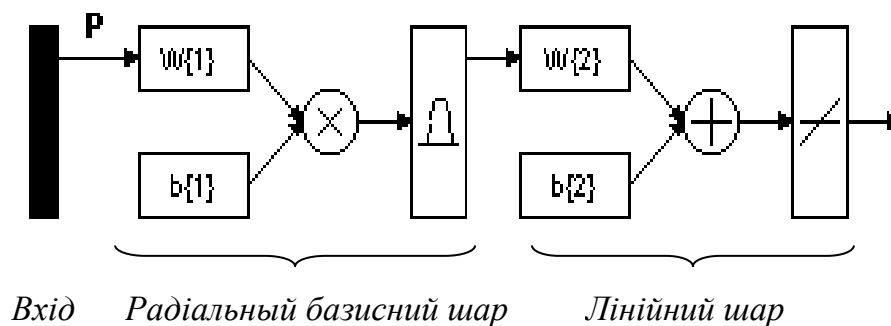


Рис. 3.14. Структура радіальної базисної нейронної мережі:

$b\{2\}$ – зсуви першого та другого шарів відповідно

P – вектор входу; $W\{1\}$, $W\{2\}$ – матриці ваг першого та другого шарів відповідно; $b\{1\}$,

Радіальний базисний шар складається з радіальних базисних нейронів, вихід яких описується формулою

$$f(P) = h\left(\frac{\|P - C\|}{\sigma}\right), \quad (3.5)$$

де P – вхідний сигнал; h – функція активації; C – значення центру радіального базисного елемента (нейрона); σ – відхилення.

Як функція активації зазвичай використовується функція Гауса (див. рис. 1.17)

На виході радіального базисного нейрона формується значення, що відповідає близькості вектора входу до вектора ваг нейрона.

Якщо вектор входу є близьким до вектора ваг, то вихід нейрона буде близьким до «1» й з відповідною вагою передаватиметься до нейронів лінійного шару; якщо ж вони відчутно відрізняються, вихід нейрона буде близьким до «0» і його вплив на виходи лінійних нейронів буде незначним.

Необхідно відзначити, що при навчанні RBF-мережі спочатку визначаються центри й відхилення для радіальних базисних нейронів, а потім настраюються вагові коефіцієнти лінійного шару.

Для RBF-мереж доведено низку теорем, згідно якими при виконанні певних умов ці мережі можуть апроксимувати довільну гладку функцію. Отже, їх можна використовувати для вирішення задач статичної ідентифікації нелінійних об'єктів.

У даній роботі скористаємось для створення радіальних базисних нейронних мереж GUI-інтерфейсом NNTool, який дає змогу створювати два типи таких мереж:

- *radial basis (exact fit)* - радіальну базисну мережу з нульовою помилкою на навчальній множині [5, С.134-135];

- *radial basis (fewer neurons)* - радіальну базисну мережу, що дозволяє управляти кількістю нейронів у прихованому шарі [5, С.135-137].

Перш ніж приступати до побудови радіальних базисних нейронних мереж, імпортуємо з робочої області MATLAB до середовища NNTool сформовані раніше навчальні послідовності P та T .

Розглянемо побудову радіальних базисних нейронних мереж з керованою кількістю нейронів у прихованому шарі. Як тип мережі варто вибрати *radial basis (fewer neurons)* і додатково задати значення припустимої середньоквадратичної помилки (*performance goal*).

Як вхідні дані (*input data*) слід вибрати масив P , а як вихідні (*target data*) – масив T . Значення параметра *spread*, що впливає на гладкість апроксимації, призначимо рівним 1.0.

Сформована радіальна базисна нейронна мережа, як відомо (див. розд. 1.6) не вимагає навчання, тому відразу ж після створення її можна експортувати до робочої області MATLAB і сформувавши S-модель.

Для тестування створеної нейромережної моделі статичного нелінійного об'єкта (3.3) пропонується скористатись схемою, що наведена на рис. 3.12.

Як вхідний тестовий сигнал можна використати, наприклад, синусоїдальний сигнал з такими параметрами: амплітуда (*amplitude*) = 0.5; зсув (*bias*) = 0.5; частота (*frequency*) = 1 рад/с; фаза (*phase*) = 0 рад.

Для оцінки адекватності нейромережної моделі об'єкту, як і раніше, зручно застосувати інтегральний квадратичний критерій. Тривалість моделювання прийняти – 10 с.

Для дослідження впливу кількості нейронів у прихованому шарі на якість моделі необхідно створити кілька подібних нейронних мереж типу *radial basis (fewer neurons)* з різним значенням параметра *performance goal* (табл. 3.4) та провести їхнє тестування. Звернімо увагу, що залежно від заданої точності (*performance goal*) пакет *nntool* створюватиме мережі з різною кількістю радіальних базисних нейронів у прихованому шарі.

Таблиця 3.4

Результати тестування нейронних мереж типу *radial basis (fewer neurons)*

Ім'я мережі	Значення параметра <i>goal</i>	Кількість радіальних базисних нейронів	Значення інтегрального квадратичного критерію
net1	0.1	2	
net2	0.01	3	
net3	0.001	3	
net4	0.0001	5	
net5	0.00001	6	
net6	0.000001	6	
net7	0.0000001	6	
net8	0.00000001	7	
net9	0.000000001	8	
net10	0.0000000001	51	

За результатами тестування (характерна осцилограма наведена на рис. 3.15) необхідно визначити й занести до таблиці значення інтегрального квадратичного критерію та зробити висновок щодо впливу кількості радіальних базисних нейронів на точність нейромережної моделі.

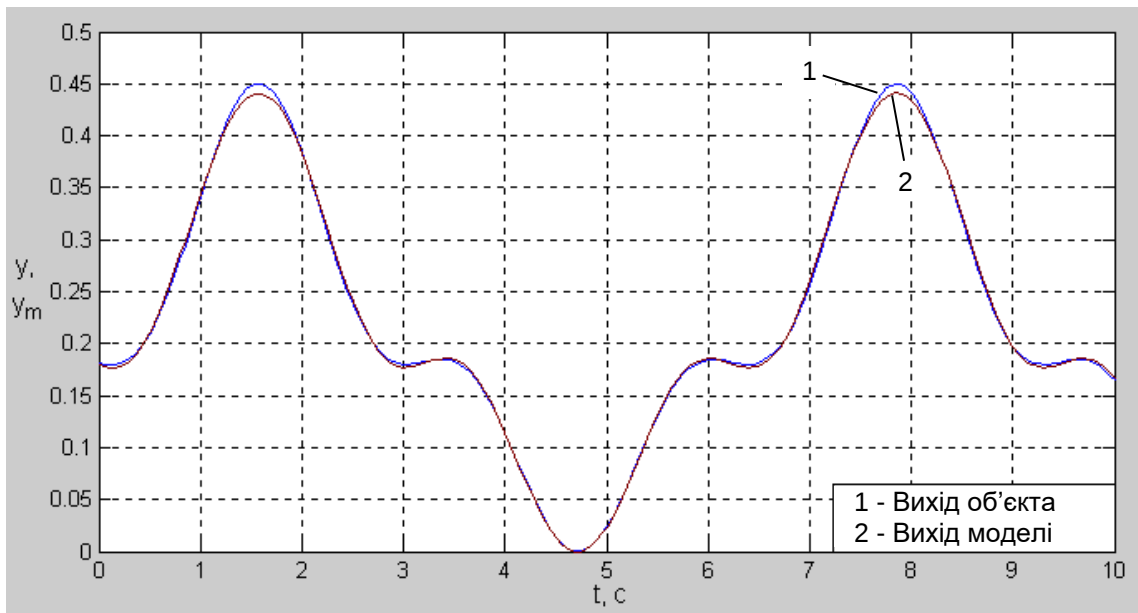


Рис. 3.15. Результати ідентифікації статичного нелінійного об'єкта за допомогою радіальної базисної мережі з керованою кількістю нейронів у прихованому шарі (варіант з 3-ма радіальними базисними нейронами)

3.3. Ідентифікація динамічних об'єктів за допомогою нейронних мереж

Мета роботи:

1. Створити, навчити та застосувати одношарову лінійну нейронну мережу прямого розповсюдження для ідентифікації лінійного динамічного об'єкта. Дослідити вплив кроку дискретизації та глибини лінії затримки на точність нейронної моделі.

2. Створити, навчити та застосувати рекурентну одношарову лінійну нейронну мережу для ідентифікації лінійного динамічного об'єкта. Дослідити вплив кроку дискретизації та глибини лінії затримки на точність нейронної моделі.

3.3.1. Вихідні дані

Ідентифікацію динамічних об'єктів за допомогою нейронних мереж розглянемо на прикладі об'єкта управління, що описується ланкою першого порядку із запізнюванням

$$W_0(p) = \frac{k_0 \cdot e^{-\tau_0 \cdot p}}{T_0 \cdot p + 1}, \quad (3.6)$$

де k_0 – коефіцієнт передачі об'єкта; T_0 – постійна часу об'єкта; τ_0 – час чистого запізнювання.

У табл. 3.5 наведені параметри різних динамічних об'єктів зазначеного типу як варіанти вихідних даних для виконання лабораторної роботи.

Нехай функція передачі об'єкта управління має вигляд

$$W_0(p) = \frac{3,5 \cdot e^{-10p}}{30 \cdot p + 1}, \quad (3.7)$$

Для ідентифікації динамічного об'єкта (3.6) будемо застосовувати динамічні одношарові лінійні нейронні мережі прямого розповсюдження та рекурентні одношарові лінійні нейронні мережі.

Як критерій для порівняння моделей між собою й для оцінки їхньої адекватності об'єктові будемо використовувати інтегральний квадратичний критерій.

Таблиця 3.5

Вихідні дані для виконання роботи

№ вар.	k_0	$T_0, \text{с}$	$\tau_0, \text{с}$
1	1,0	15	12
2	2,6	10	15
3	5,3	25	14
4	4,8	40	20
5	7,2	17	10
6	2,9	33	8
7	14,0	6	16
8	1,8	20	4
9	6,4	12	26
10	10,2	7	17
11	0,8	14	5
12	3,9	19	11

3.3.2. Формування навчальної множини

Оскільки динамічний об'єкт (3.6) є безперервним, а нейронні мережі, що його відтворюватимуть, є дискретними, постає питання про вибір такту дискретності (кроку дискретизації) нейронних мереж.

З метою дослідження впливу кроку дискретизації на якість динамічної нейромережної моделі об'єкта створимо нейронні мережі з тактом дискретності 5,0 с, 2,5 с та 1,0 с.

Зауважимо, що такт дискретності нейронної мережі повинен відповідати кроку дискретизації навчальних послідовностей.

Для формування навчальних послідовностей у ППП Simulink зберемо схему, що наведена на рис. 3.16.

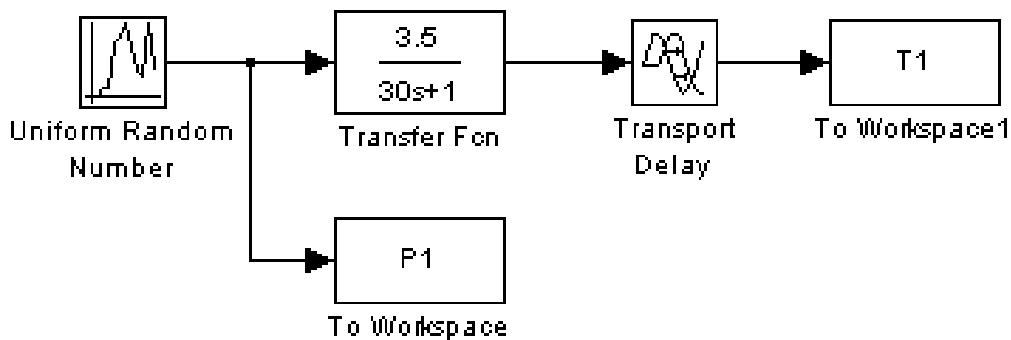


Рис. 3.16. Схема для формування навчальних послідовностей

На вхід об'єкта управління подаватимемо сигнал з блоку *Uniform Random Number*, який представляє собою рівномірно розподілену на певному інтервалі випадкову дискретну у часі величину

Задамо наступні параметри блока *Uniform Random Number*: *minimum* = 0; *maximum* = 1; *initial seed* = 121; *sample time* = 200. Параметри *minimum* та *maximum* визначають відповідно мінімальне та максимальне значення вихідного сигналу блока; параметр *initial seed* дозволяє формувати різні реалізації випадкового сигналу; параметр *sample time* визначає крок дискретизації, з яким змінюється сигнал на виході блока.

За допомогою блоків *To Workspace* та *To Workspace1* запишемо значення вхідного й вихідного сигналів об'єкта до робочої області MATLAB з певним кроком у вигляді масивів

Здійснимо трикратне моделювання зібраної схеми (рис. 3.16), задаючи щоразу параметри блоків *To Workspace* й *To Workspace1* та тривалість моделювання згідно з табл. 3.6.

Таблиця 3.6

Формування навчальних послідовностей

Блок <i>To Workspace</i>		Блок <i>To Workspace 1</i>		Час моделювання, с
<i>Variable name</i>	<i>Sample time, с</i>	<i>Variable name</i>	<i>Sample time, с</i>	
P1	5	T1	5	995
P2	2.5	T2	2.5	997.5
P3	1	T3	1	599

В результаті у робочій області MATLAB будуть створені масиви даних *P1* і *T1*, *P2* і *T2*, *P3* і *T3*, які для подальшого використання транспонуємо й представимо у вигляді послідовності комірок:

```
>> P1 = con2seq (P1'); T1 = con2seq (T1');
>> P2 = con2seq (P2'); T2 = con2seq (T2');
>> P3 = con2seq (P3'); T3 = con2seq (T3');
```

Імпортуємо сформовані навчальні послідовності даних до GUI-інтерфейсу NNTool й перейдемо до створення власне нейронних мереж.

3.3.3. Ідентифікація за допомогою динамічної одношарової лінійної нейронної мережі прямого розповсюдження

Спочатку розглянемо ідентифікацію об'єкта (3.6) за допомогою динамічної одношарової лінійної нейронної мережі прямого розповсюдження.

Структурна схема такої мережі, що містить один нейрон та $(N-1)$ блоків запізнювання в лінії затримки, наведена на рис. 3.17. Її вихід може бути описаний співвідношенням:

$$y(k) = \sum_{i=1}^N w_{1i} \cdot u(k-i+1) + b, \quad (3.8)$$

де $y(k)$ – значення виходу в момент часу k ; $u(k-i)$ – значення входу в момент часу $(k-i)$; $w_{11}, w_{12}, \dots, w_{1N}$ – вагові коефіцієнти; b – зсув.

Для створення й навчання динамічних одношарових лінійних нейронних мереж прямого розповсюдження скористаємось GUI-інтерфейсом NNTool. Як тип нейронної мережі вибираємо *Linear layer (train)* / *Лінійний шар (навчання)* і задаємо такі параметри: діапазони вектора входу (*input ranges*); кількість нейронів (*number of neurons*); вектор затримки входу (*input delay vector*); швидкість навчання (*learning rate*).

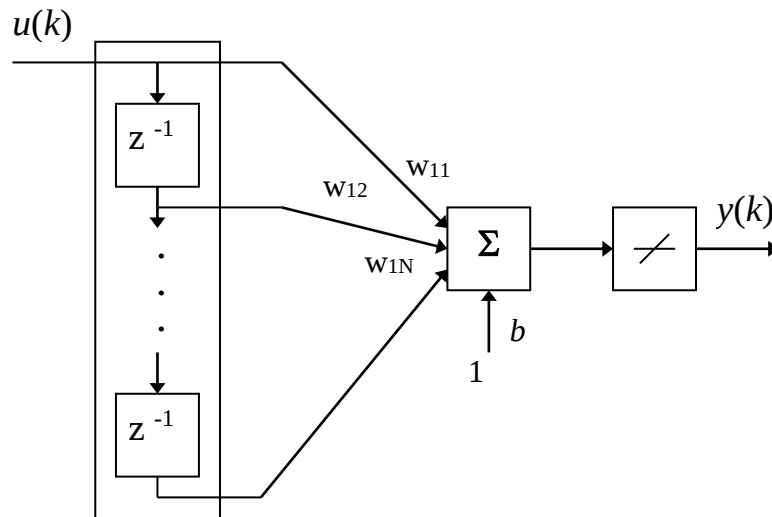


Рис. 3.17. Структурна схема динамічної одношарової лінійної нейронної мережі прямого розповсюдження, що містить один нейрон

Вважаючи, що вхідний сигнал об'єкта управління може набувати значень в межах від «0» до «1», задамо для всіх створюваних нейронних мереж діапазон вектора входу – $[0\ 1]$. Кількість нейронів встановимо – 1. Прийнятне значення швидкості навчання визначатимемо в процесі навчання.

Відзначимо – якщо при спробі навчити нейронну мережу, значення критерію навчання починає необмежено зростати, це свідчить про обрання занадто великої швидкості навчання. У цьому випадку необхідно створити нову аналогічну нейронну мережу з меншим значенням параметра *learning rate* і повторити спробу навчання.

Створимо кілька подібних нейронних мереж з різною глибиною лінії затримки входу, навчимо їх за допомогою сформованих раніше навчальних послідовностей і дослідимо вплив такту дискретності мережі й глибини лінії затримки на якість нейромережної моделі об'єкта.

Навчені нейронні мережі експортуємо в робочу область MATLAB і сформуємо їхні *S-моделі*, використовуючи функцію *gensim (net, TS)*. Оскільки

створені нейронні мережі є дискретними, значення параметра TS при формуванні S-моделі призначатимемо рівним такту дискретності мережі.

Для тестування нейромережних моделей варто використати схему, наведену на рис. 3.18.

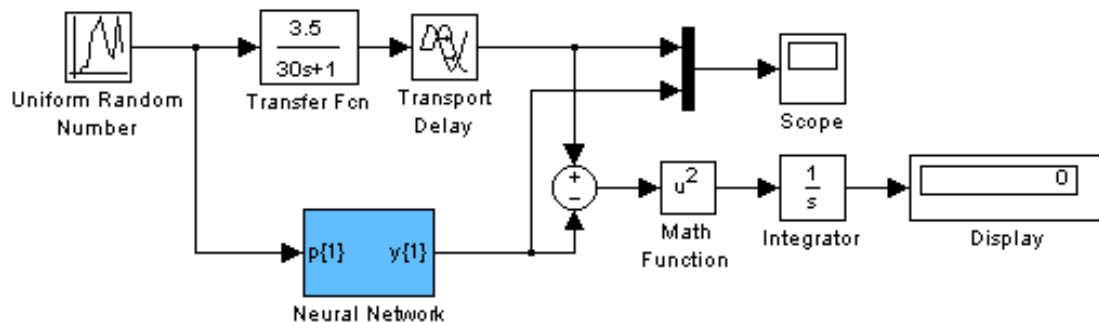


Рис. 3.18. Схема для тестування динамічних нейромережних моделей

На вхід об'єкта управління та його нейромережної моделі подаємо сигнал, сформований блоком *Uniform Random Number* із такими значеннями параметрів: $minimum = 0$; $maximum = 1$; $initial\ seed = 2122$; $sample\ time = 200$. Тривалість моделювання задаємо – 1000 с.

Результати навчання й тестування динамічних одношарових лінійних нейронних мереж зведемо до табл. 3.7, а також представимо у вигляді осцилограм (рис. 3.19).

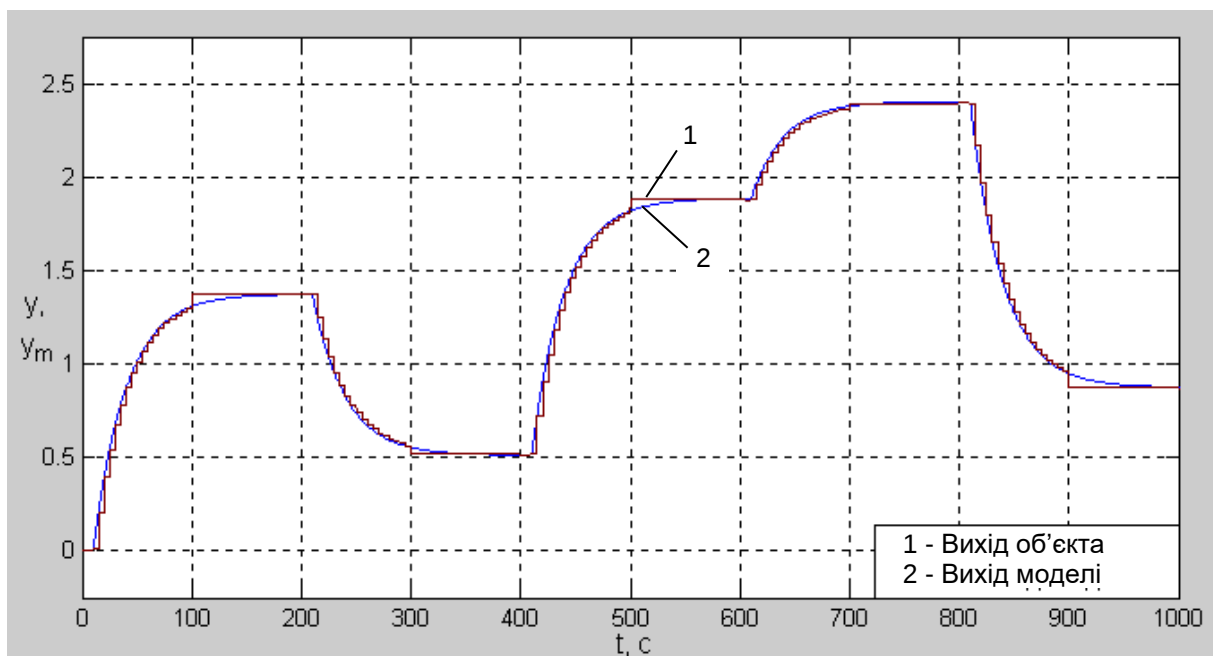


Рис. 3.19. Результати ідентифікації динамічного об'єкта за допомогою динамічної одношарової лінійної нейронної мережі з тактом дискретності 5 с та глибиною лінії затримки на 20 тактів

**Результати навчання й тестування динамічних одношарових лінійних
нейронних мереж прямого розповсюдження**

Ім'я мережі	Вектор затримки входу	Швидкість навчання (<i>learning rate</i>)	Кількість епох (<i>epochs</i>)	Критерій оцінки якості навчання (<i>MSE</i>)	Інтегральний квадратичний критерій
Sample time = 5,0 c					
n1_1	[0 1...10]	0001	5000		
n1_2	[0 1...20]	0001	10000		
n1_3	[0 1...30]	0001	30000		
n1_4	[0 1...40]	0001	40000		
Sample time = 2.5 c					
n2_1	[0 1...20]	00001	7500		
n2_2	[0 1...40]	00001	50000		
n2_3	[0 1...60]	00001	75000		
n2_4	[0 1...80]	00001	250000		
Sample time = 1,0 c					
n3_1	[0 1...50]	0000001	5000		
n3_2	[0 1 ... 100]	0000001	5000		
n3_3	[0 1 ... 150]	0000001	50000		
n3_4	[0 1 ... 200]	0000001	600000		

На підставі даних табл. 3.19 слід зробити висновки щодо впливу кількості блоків запізнювання в лінії затримки на час навчання нейронної мережі та її точність. Звернути увагу на співвідношення між тактом мережі, дискретністю й глибиною лінії затримки.

Зауваження. На рис. 3.19 наведені типові графіки, що отримані в результаті тестування нейромережних моделей із різною глибиною лінії затримки й тактом дискретності 5с. Видно, що нейромережна модель досить точно описує криву перехідного процесу об'єкта на часовому інтервалі, якій дорівнює добутку такту дискретності моделі на кількість блоків запізнювання в лінії затримки, після чого вихід моделі стрибком набуває значення, що є близьким до значення вихідного сигналу об'єкта по

закінченні перехідного процесу, а не “стежить” за кривою перехідного процесу.

Це пояснюється тим, що розглянуті нейромережні моделі мають кінцеву пам'ять, що визначається їхнім тактом дискретності й глибиною лінії затримки. Для того, щоб нейромережна модель була адекватною динамічному об'єкту, вона повинна мати достатню глибину лінії затримки. Експериментально величину цієї затримки можна оцінити за кривою перехідного процесу об'єкта як відношення часу перехідного процесу до такту дискретності нейронної мережі.

Істотним недоліком динамічних лінійних нейронних мереж є те, що для одержання адекватної нейромережної моделі динамічного об'єкта управління може знадобитись занадто велика кількість блоків запізнювання в лінії затримки й занадто тривалий час навчання.

3.3.4. Ідентифікація за допомогою рекурентної одношарової лінійної нейронної мережі

Структурна схема такої рекурентної нейронної мережі, що містить один охоплений зворотним зв'язком нейрон з лініями затримки вхідного сигналу та сигналу зворотного зв'язку, наведена на рис. 3.20.

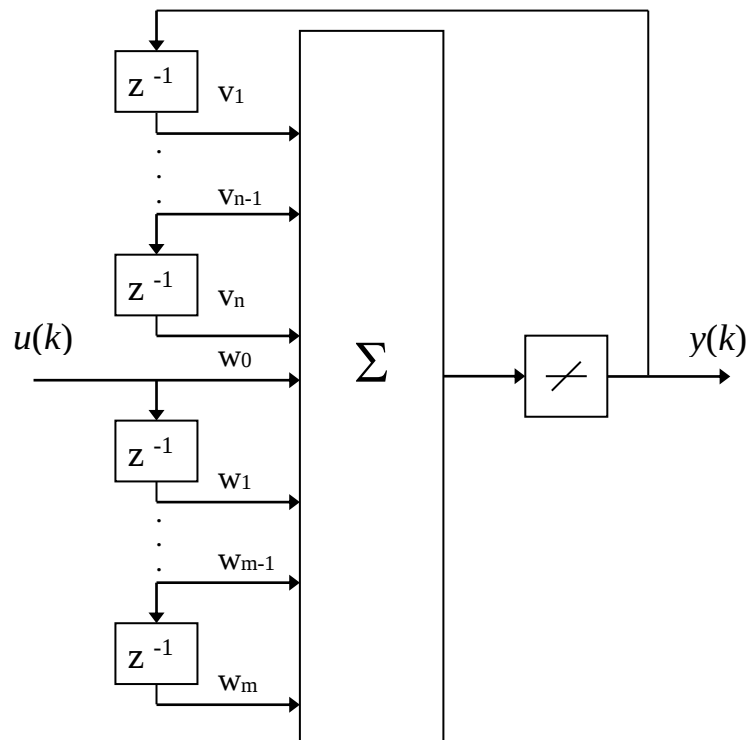


Рис. 3.20. Структурна схема рекурентної одношарової лінійної нейронної мережі, що містить один нейрон

Вихід нейрона може бути описаний таким співвідношенням

$$y(k) = \sum_{i=0}^m w_i \cdot u(k-i) + \sum_{j=1}^n v_j \cdot y(k-j), \quad (3.9)$$

де $u(k-i)$ – вхід мережі в момент часу $(k-i)$; $w_0, w_1, \dots, w_m, v_1, \dots, v_n$ – вагові коефіцієнти; $y(k-j)$ – вихід мережі в момент часу $(k-j)$; m – глибина лінії затримки вхідного сигналу; n – глибина лінії затримки зворотного зв'язку.

Побудова нейронних мереж з подібною структурою в рамках GUI-інтерфейсу NNTool не є можливою. Тому для їхнього формування скористаємось М-функцією *network*, що дозволяє створювати нейронні мережі довільної структури і має такий синтаксис:

net = network (numInputs, numLayers, biasConnect, inputConnect, layerConnect, outputConnect, targetConnect),

де *net* – ім'я нейронної мережі; *numInputs* – кількість входів; *numLayers* – кількість шарів; *biasConnect* – матриця зв'язності для зсувів; *inputConnect* – матриця зв'язності для входів; *layerConnect* – матриця зв'язності для шарів; *outputConnect* – матриця зв'язності для виходів; *targetConnect* – матриця зв'язності для цілей [5, С.228–230, С.245–248].

Створимо кілька подібних нейронних мереж, що відрізняються кількістю блоків запізнювання в лінії затримки входу, і дослідимо вплив глибини лінії затримки вхідного сигналу на точність нейромережної моделі об'єкта. Глибину лінії затримки зворотного зв'язку приймаємо рівною 1.

Як приклад розглянемо формування спрощеної нейронної мережі з двома блоками запізнювання в лінії затримки вхідного сигналу (рис. 3.21).

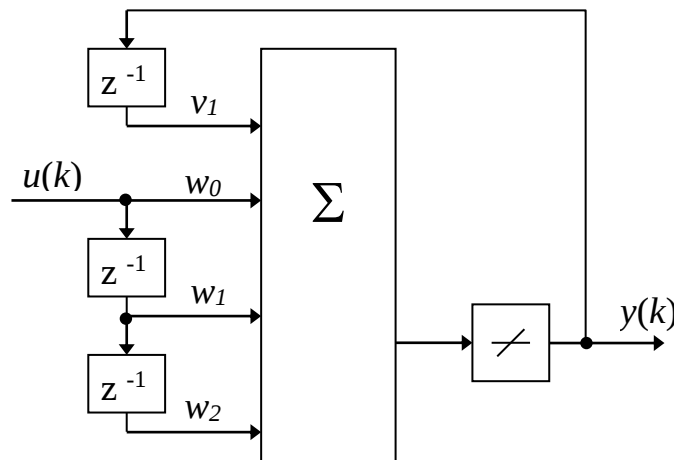


Рис. 3.21. Структурна схема спрощеної рекурентної нейронної мережі

Насамперед, необхідно створити шаблон рекурентної одношарової лінійної нейронної мережі, що містить один нейрон без зсуву:

```
>> net1_1 = network (1, 1, 0, 1, 1, 1, 1);
```

Далі вводимо лінії затримки вхідного сигналу й сигналу зворотного зв'язку:

```
>> net1_1.inputWeights{1, 1}.delays = [0 1 2];
```

```
>> net1_1.layerWeights{1, 1}.delays = [1];
```

Задаємо діапазон значень вхідного сигналу:

```
>> net1_1.inputs{1}.range = [0 1];
```

Для навчання нейронної мережі використаємо М-функцію *trainlm*, що реалізує алгоритм Левенберга-Марквардта LM.

```
>> net1_1.trainFcn = 'trainlm';
```

Для оцінювання якості навчання встановимо критерій середньої квадратичної помилки:

```
>> net1_1.performFcn = 'mse';
```

Перед початком навчання створену нейронну мережу бажано ініціювати, привласнивши їй вагам випадкові початкові значення. Це можна зробити, наприклад, у такий спосіб:

```
>> net1_1.initFcn = 'initlay';
```

```
>> net1_1.layers{1}.initFcn = 'initwb';
```

```
>> net1_1.inputWeights{1,1}.initFcn = 'rands';
```

```
>> net1_1.layerWeights{1,1}.initFcn = 'rands';
```

```
>> net1_1 = init (net1_1);
```

Проте, щоб мати можливість відтворити отримані результати навчання, ми не будемо ініціювати створені нейронні мережі й будемо здійснювати їхнє навчання з нульовими початковими значеннями ваг.

Для навчання та тестування рекурентних одношарових лінійних нейронних мереж будемо використовувати ті ж самі дані, що й у випадку одношарових динамічних лінійних нейронних мереж прямого розповсюдження (див. табл. 3.6).

Результати навчання й тестування рекурентних лінійних нейронних мереж слід звести до табл. 3.8, а також представити у вигляді осцилограм (рис. 3.22).

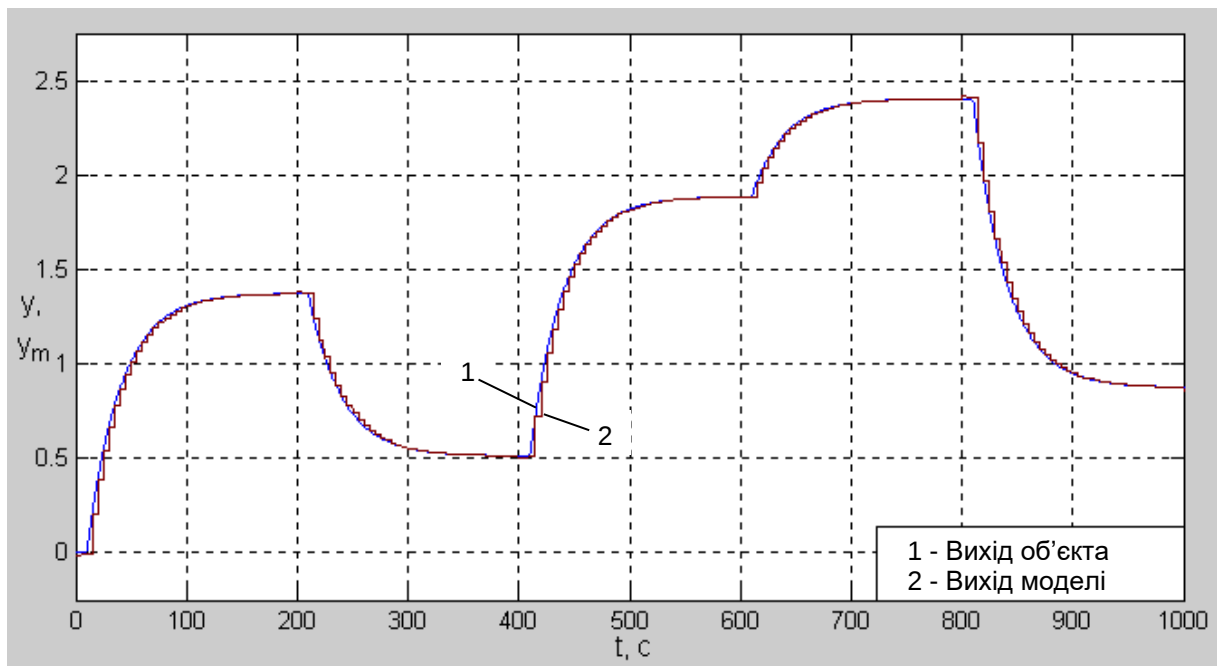


Рис. 3.22. Результати ідентифікації динамічного об'єкта за допомогою рекурентної одношарової лінійної нейронної мережі з тактом дискретності 5,0 с і глибиною лінії затримки вхідного сигналу на 4 такти

Таблиця 3.8

Результати навчання та тестування рекурентних одношарових лінійних нейронних мереж

Ім'я мережі	Вектор затримки входу	Кількість епох (<i>epochs</i>)	Критерій оцінки якості навчання (<i>MSE</i>)	Інтегральний квадратичний критерій
1	2	3	4	5
Sample time = 5,0с				
net1_1	[0 1...2]	110		
net1_2	[0 1...3]	53		
net1_3	[0 1...4]	66		
Sample time = 2.5с				
net2_1	[0 1...4]	36		
net2_2	[0 1...5]	116		
net2_3	[0 1...6]	109		
Sample time = 1,0с				
net3_1	[0 1...10]	82		
net3_2	[0 1...11]	40		
net3_3	[0 1...12]	36		

Проаналізувавши дані табл. 3.8 та графіки, що отримані за результатами тестування нейромережних моделей, необхідно зробити висновки щодо впливу глибини затримки вхідного сигналу на точність нейромережної моделі. Звернути увагу на зв'язок між кількістю блоків запізнювання в лінії затримки вхідного сигналу, яка необхідна для адекватного відтворення об'єкта, тактом дискретності моделі та часом чистого запізнювання об'єкта.

3.4. Дослідження роботи АСР з нейрорегулятором

Мета роботи:

1. Синтезувати та дослідити роботу нейромережних регуляторів, що представляють собою інверсну модель об'єкта управління, а саме:

- пряме інверсне управління,
- пряме адитивне управління,
- управління із внутрішньою моделлю.

2. Порівняти якість роботи нейромережного та класичного ПД-регуляторів при управлінні об'єктом першого порядку із запізнюванням.

3.4.1. Вихідні дані

У даній роботі як об'єкт управління будемо використовувати ланку першого порядку із запізнюванням з попередньої роботи, який описується функцією передачі (3.6) з параметрами, що наведені у таблиці 3.5.

На практиці для управління подібними технологічними об'єктами зазвичай застосовуються ПД-регулятори. Тому побудовані нами нейромережні системи доцільно порівнювати саме з ПД-управлінням. Якість управління в обох випадках оцінюватимемо за допомогою інтегрального квадратичного критерію.

3.4.2. Пряме інверсне управління

Пряме інверсне управління (*direct inverse control, DIC*) являє собою послідовне з'єднання нейромережного контролера (нейроконтролера), який є інверсною моделлю об'єкта, і власне об'єкта управління (див. рис. 2.1).

За допомогою схеми прямого інверсного управління, у якій нейроконтролер реалізує точну зворотну (інверсну) модель об'єкта

управління, теоретично можна досягти ідеального управління, при якому вихід об'єкта завжди дорівнює завданню.

Схема прямого інверсного управління є розімкненою. Отже, вона прийнятна тільки для управління стійкими стаціонарними об'єктами за відсутності збурювань.

Для одержання інверсної нейромережної моделі об'єкта управління необхідно навчити нейронну мережу його зворотній динаміці [8]. Із цією метою на вхід нейронної мережі подається вихідний сигнал об'єкта, потім її вихід порівнюється із вхідним сигналом об'єкта, а сигнал помилки використовується для навчання мережі (див. рис. 2.2).

Для прикладу розглянемо побудову дискретної нейромережної інверсної моделі об'єкта управління (3.6), що має такт дискретності 1с.

За допомогою схеми з попередньої роботи (рис. 3.16) сформуємо навчальні послідовності із кроком дискретизації 1с.

Параметри блоку *Uniform Random Number* задамо у такий спосіб: *minimum* = 0; *maximum* = 1; *initial seed* = 0; *sample time* = 200. Тривалість моделювання встановимо – 599 с.

В результаті моделювання в робочій області MATLAB будуть створені масиви даних P і T , які для подальшого використання транспонуємо й представимо у вигляді послідовності чарунок:

```
>> T = con2seq (T'); P = con2seq (P');
```

Після цього в ППП Neural Network Toolbox створимо рекурентну одношарову лінійну нейронну мережу з лінією затримки вхідного сигналу на один такт і навчаємо її інверсній динаміці об'єкта (3.6) за допомогою сформованих навчальних послідовностей.

Нагадаємо, що під час навчання нейронної мережі як вхідну послідовність необхідно вказати послідовність значень вихідного сигналу об'єкта T , а як цільову послідовність – послідовність значень вхідного сигналу об'єкта P .

По завершенні навчання нейронної мережі сформуємо її S-модель, яку надалі при побудові нейромережних систем управління будемо використовувати в якості нейроконтролера.

Для проведення дослідження необхідно зібрати в системі структурного моделювання Simulink схему, що дозволяє порівняти пряме інверсне нейромережне управління із класичним ПД-управлінням. На рис. 3.25

наведена така схема для випадку, коли об'єкт управління описується функцією передачі

$$W_o(p) = \frac{3,5 \cdot e^{-10 \cdot p}}{30 \cdot p + 1}.$$

У цій схемі параметри ПД-регулятора необхідно підібрати за відомою з ТАУ інженерною методикою так, щоб перехідний процес був аперіодичним з мінімальним часом регулювання. Для наведеної схеми (рис. 3.23) ці параметри мають такі значення: $k_{\Pi} = 0.34$; $k_{\text{И}} = 0.01$; $k_{\text{Д}} = 1$.

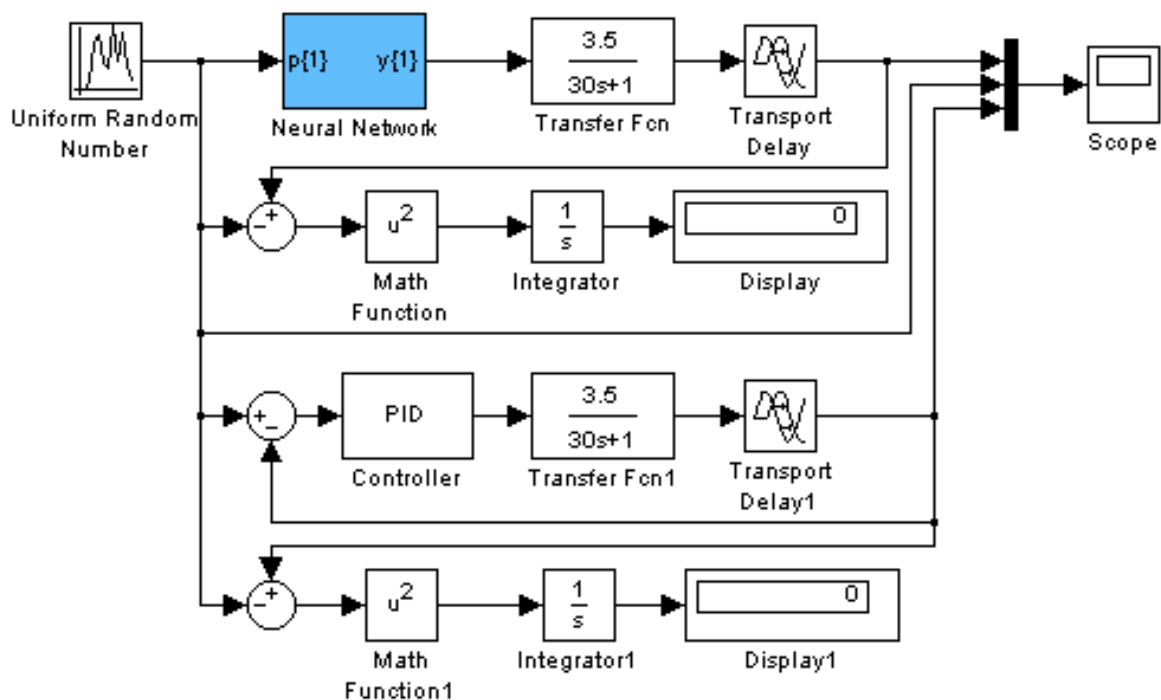


Рис. 3.23. Схема порівняння прямого інверсного неймережного управління з ПД-управлінням

Для тестування схеми (рис. 3.23), пропонується використати тестовий сигнал завдання, що являє собою послідовність прямокутних імпульсів тривалістю 250 с з випадковою амплітудою, яка рівномірно розподілена в інтервалі від 0 до 1. Для формування такого сигналу можна скористатись блоком *Uniform Random Number*, параметри якого треба задати у такий спосіб: *minimum*=0; *maximum* = 1; *initial seed* = 2122; *sample time* = 250.

Тривалість моделювання схеми встановити 1000с.

На рис. 3.24 як приклад наведені результати моделювання схеми (рис. 3.23).

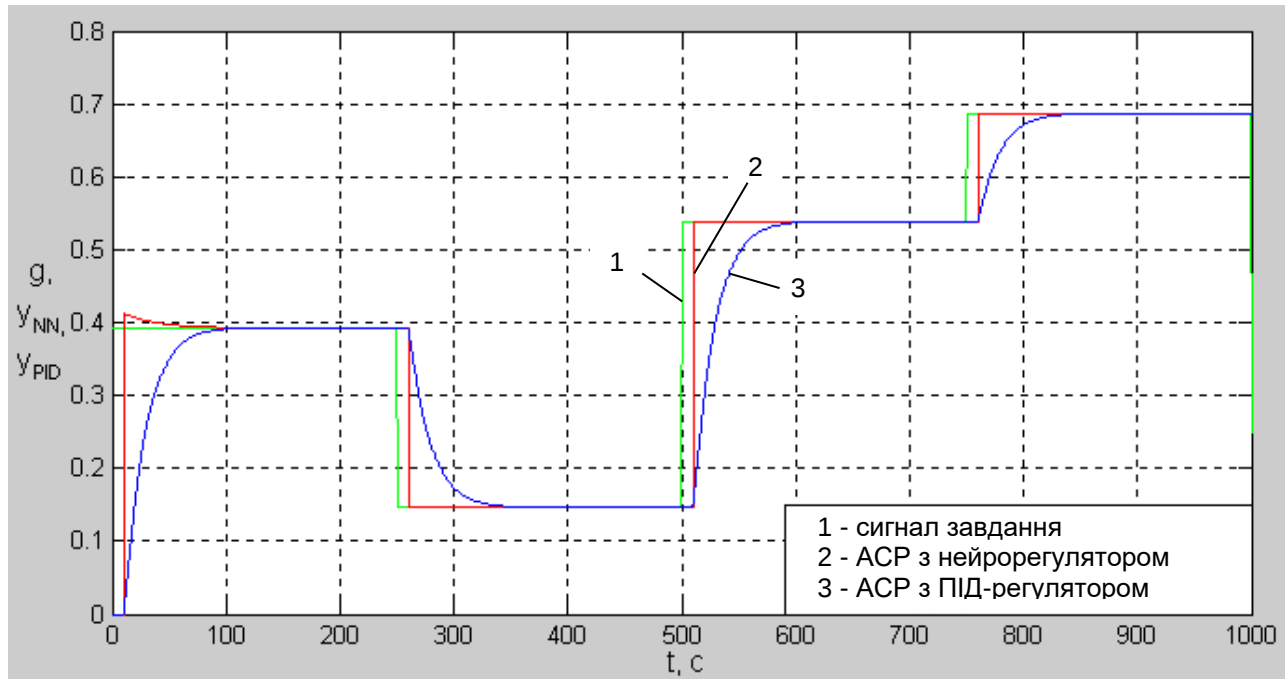


Рис. 3.24. Результати порівняння прямого інверсного нейромережного управління з ПІД-управлінням

В результаті тестування схеми (рис. 3.23) необхідно одержати значення критерію якості управління й, ґрунтуючись на ньому, порівняти пряме інверсне управління із традиційним ПІД-управлінням.

Зауваження. Варто звернути увагу на те, що дослідження проводилось за відсутності неконтрольованих збурювань, що впливають на об'єкт управління.

3.4.3. Пряме адитивне управління

Ідея **прямого адитивного управління** (additive feed-forward control, AFFC) полягає в паралельному підключенні до незадовільно працюючого контролера додаткового нейроконтролера, інверсного щодо об'єкту (рис. 2.4).

Для простоти й наочності в ролі незадовільно працюючого контролера використаємо П-регулятор, що має єдиний параметр налаштування – коефіцієнт пропорційності k_P . Його значення для отримання аперіодичного перехідного процесу з мінімальним часом регулювання визначаємо за формулою

$$k_P = \frac{0.3}{k_0 \cdot \tau_0 / T_0},$$

де k_0 - коефіцієнт передачі об'єкта; T_0 - постійна часу об'єкта; τ_0 - час чистого запізнювання. (Для нашого прикладу $k_P = \frac{0.3}{3.5 \cdot \frac{10}{30}} = 0.257$.)

Для порівняння прямого адитивного нейромережного управління з ПІД-управлінням за відсутності неконтрольованих збурювань, які впливають на об'єкт управління, необхідно зібрати схему, наведену на рис. 3.25.

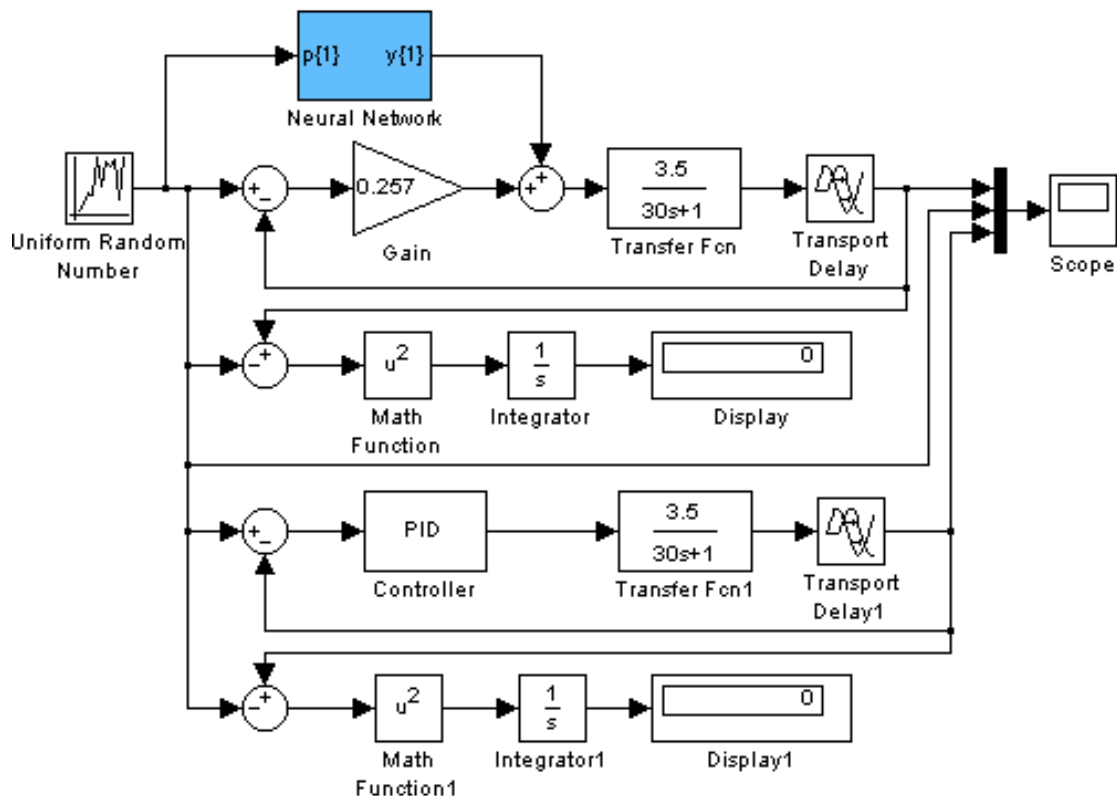


Рис. 3,25. Схема порівняння прямого адитивного нейромережного управління з ПІД-управлінням за відсутності неконтрольованих збурювань (перехідний процес коливальний)

Щоб мати можливість порівнювати розглянуті в даній роботі нейромережні стратегії між собою, будемо для їх тестування використовувати ті ж тестові сигнали завдання, що й для схеми порівняння прямого інверсного нейромережного управління з ПІД- управлінням (див. розд. 3.4.2). Як приклад на рис. 3.26 наведені результати моделювання схеми

(рис. 3.25) з такими параметрами ПІД-регулятора: $k_I = 0.6$; $k_D = 1$.

В результаті тестування схеми (рис. 3.25) необхідно отримати значення критерію якості управління й, ґрунтуючись на ньому, порівняти пряме адитивне управління із традиційним ПІД- управлінням.

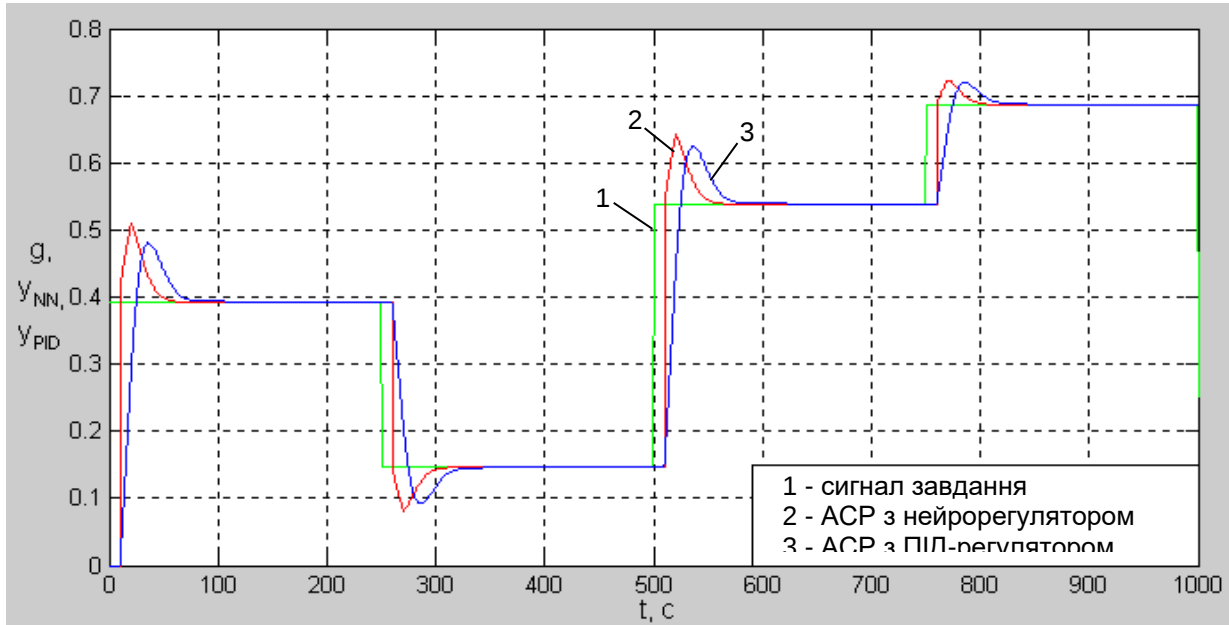


Рис. 3.26. Результати порівняння прямого адитивного нейромережного управління з ПІД-управлінням

Схема, що наведена на рис. 3.25 дозволяє оцінити якість управління тільки за відсутності неконтрольованих збурювань, що впливають на об'єкт управління. Для того ж, щоб порівняти пряме адитивне нейромережне управління з ПІД-управлінням за наявності неконтрольованих збурювань, необхідно цю схему «доопрацювати» (рис. 3.27).

Для моделювання впливу, що збурює (рис. 3.27) використаємо блок *Random Number*, що формує на своєму виході нормально розподілений випадковий сигнал і має наступні параметри: *mean* (середнє значення): 0; *variance* (дисперсія): 0.025; *initial seed* (початковий стан): 0; *sample time* (крок дискретизації): 0.

Приблизний вигляд осцилограм, які можуть бути отримані в результаті тестування схеми за рис. 3.27, наведено на рис. 3.28.

На підставі отриманих значень інтегрального квадратичного критерію якості управління необхідно порівняти роботу класичного ПІД-регулятора з роботою нейромережного адитивного регулятора як за відсутності, так і за наявності неконтрольованих збурювань.

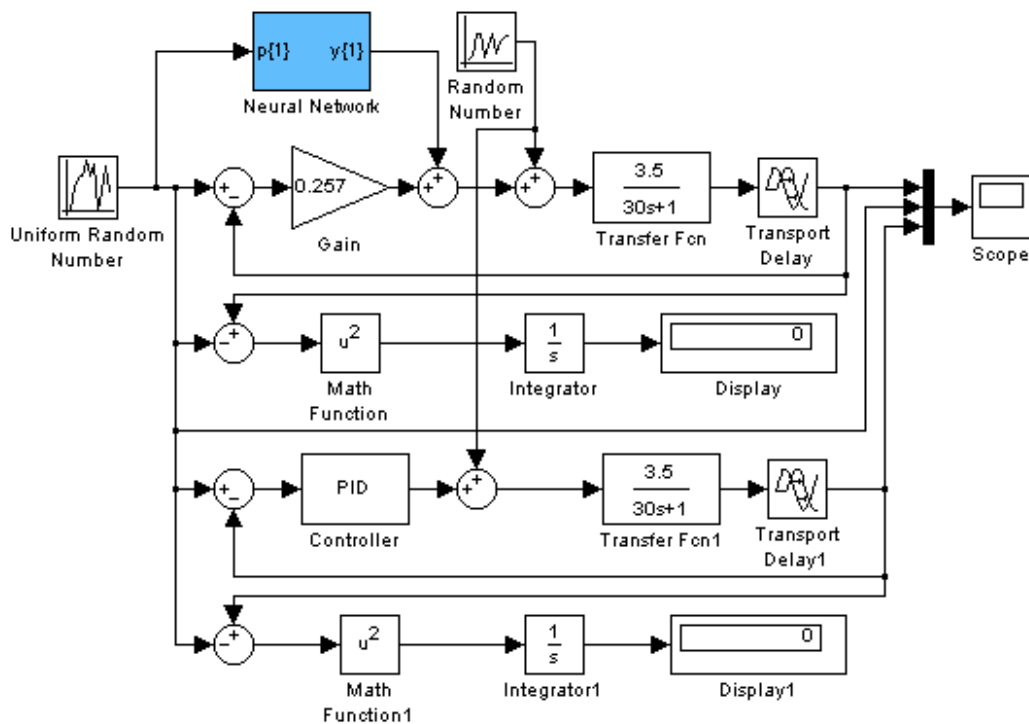


Рис. 3.27. Схема порівняння прямого адитивного нейромережного управління з ПІД-управлінням за наявності неконтрольованих збурювань, що впливають на об'єкт управління

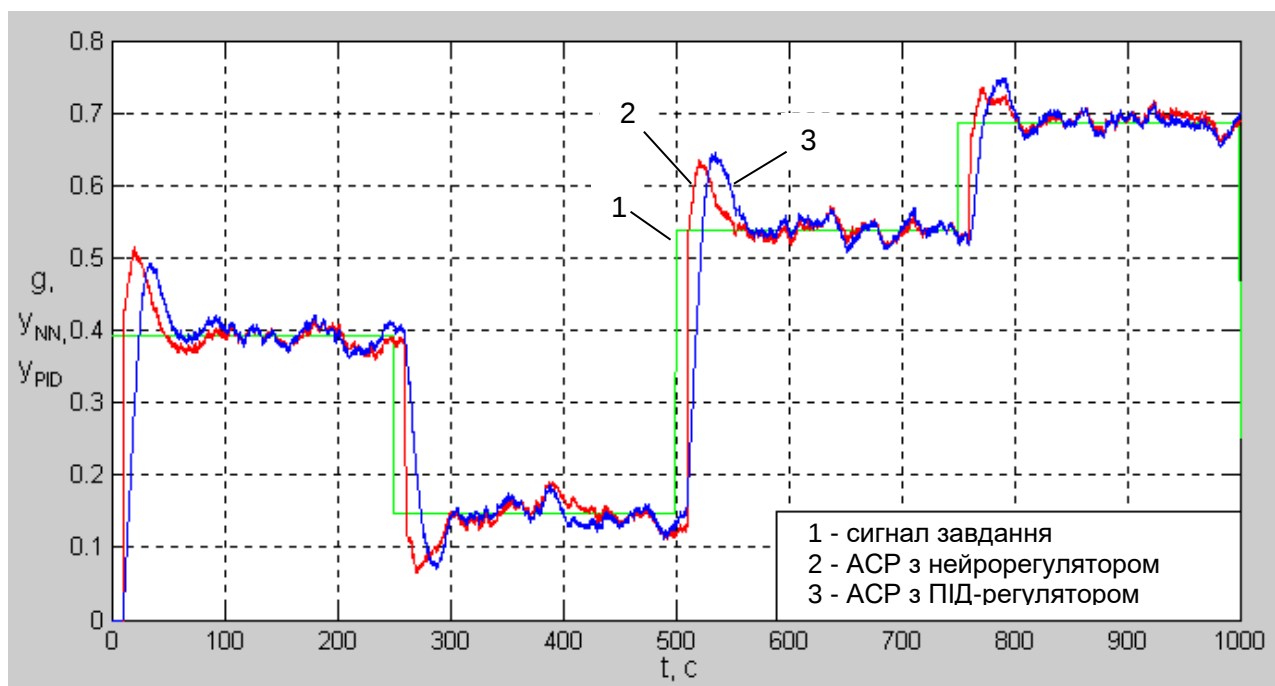


Рис. 3.28. Результати порівняння прямого адитивного нейромережного управління з ПІД-управлінням за наявності неконтрольованих збурювань, що впливають на об'єкт управління

Як додаткове самостійне дослідження пропонується порівняти стратегії прямого інверсного й прямого адитивного нейромережного управління. Для цього варто зробити спільне тестування схем за рис. 3.23 та рис. 3.27. При цьому до контуру адитивного нейроуправління необхідно послідовно з нейрорегулятором включити фільтр у вигляді аперіодичної ланки першого порядку (це забезпечить аперіодичний вигляд перехідного процесу й дозволить здійснити порівняння коректно).

3.4.4. Управління зі внутрішньою моделлю

Пряме адитивне нейромережне управління так саме, як і пряме інверсне управління, на відміну від ПІД-управління, не дозволяє управляти нестационарними об'єктами. Для подібних об'єктів використовують управління **зі внутрішньою моделлю** (*internal model control, ІМС*). Схеми управління із внутрішньою моделлю містять нейроконтролер, нейромережну модель об'єкта та робастного фільтра (див. рис. 2.6).

Розглянемо два варіанти побудови схеми нейромережного управління із внутрішньою моделлю, що відрізняються способом побудови нейромережної моделі об'єкта управління.

Перший спосіб полягає у використанні для побудови моделі динамічного об'єкта динамічної нейронної мережі. Для надання нейронній мережі динамічних властивостей можуть додатково вводитися лінії затримки вхідних сигналів (нейронна мережа з лінією затримки вхідних сигналів) та/або зворотні зв'язки (рекурентна нейронна мережа).

Структурна схема ідентифікації об'єкта управління в цьому випадку має вигляд, як на рис. 1.9. (Її застосування для побудови нейромережної моделі об'єкта було розглянуте раніше у розд. 3.3.). У даному дослідженні використаємо одну з динамічних нейромережних моделей *п3_3*, *п3_4* (див. табл. 3.7), або *net3_2*, *net3_3* (див. табл. 3.8), що були створені під час попереднього дослідження (див. розд. 3.3.3 та розд. 3.3.4).

Для прикладу побудуємо систему управління зі внутрішньою моделлю, в якій нейромережну модель об'єкта представлено рекурентною одношаровою лінійною нейронною мережею *net3_2* із глибиною лінії затримки вхідного сигналу на 11 тактів.

Для порівняння якості нейромережного управління зі внутрішньою моделлю й ПІД-управління за відсутності та за наявності неконтрольованих

збурювань, що впливають на об'єкт управління, у ППП Simulink необхідно зібрати схеми, наведені на рис. 3.29 та рис. 3.30 відповідно.

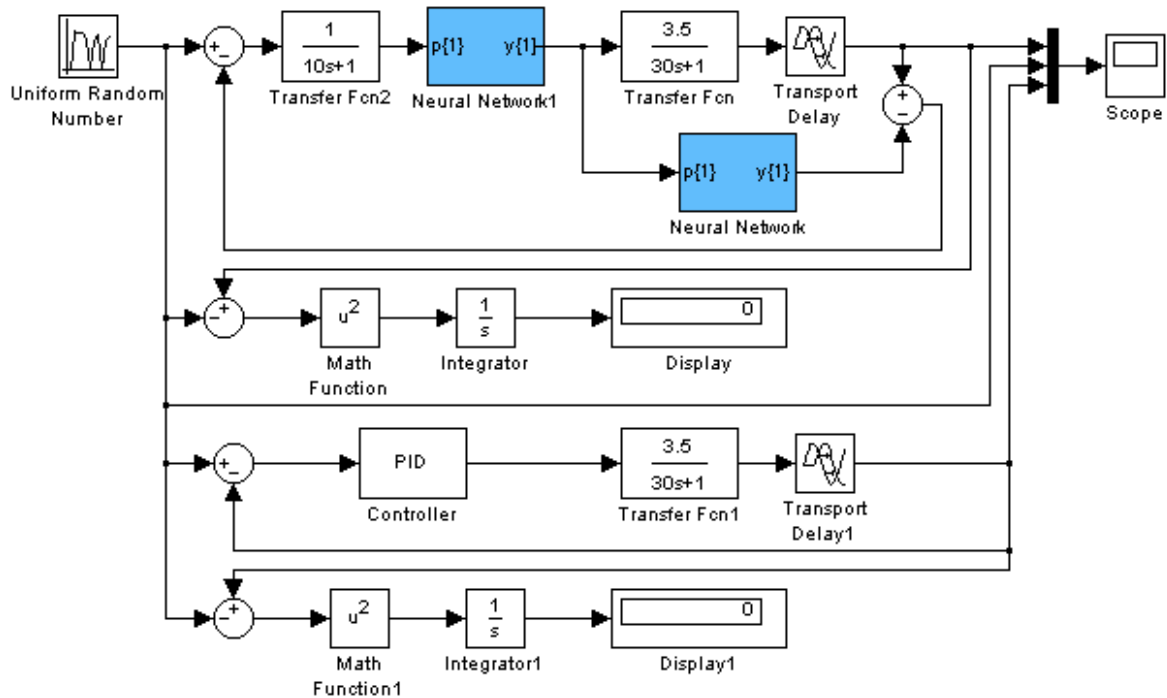


Рис. 3.29. Схема порівняння нейромережного управління зі внутрішньою моделлю та ПІД-управління за відсутності неконтрольованих збурювань (пряма нейромережна модель реалізована динамічною нейронною мережею)

При моделюванні схем, що наведені на рис. 3.29 та 3.30, будемо використовувати ті ж тестові сигнали завдання, що й для інших аналогічних схем порівняння.

Параметри збурювання в схемі (рис. 3.30) задаємо такі ж, як і в схемі (рис. 3.25), щоб мати можливість порівнювати отримані результати.

Відзначимо, що при тестуванні усіх схем, що порівнюють управління зі внутрішньою моделлю та ПІД-управління, будемо використовувати ПІД-регулятор з тими ж параметрами, що на рис. 3.23, рис. 3.25 та рис. 3.27, який забезпечуватиме аперіодичний перехідний процес із мінімальним часом регулювання.

Зразковий вигляд осцилограм, які можуть бути отримані в результаті тестування схеми за рис. ЛР4.13, представлені на рис. ЛР4.14.

На підставі отриманих значень інтегрального квадратичного критерію якості управління необхідно порівняти роботу класичного ПІД-регулятора з роботою системи нейромережного управління із внутрішньою моделлю як за відсутності, так і за наявності неконтрольованих збурювань.

Занести значення критерію якості управління, що отримані в результаті тестування схем, що наведені на рис. 3.29 та рис. 3.30, до табл. 3.9.

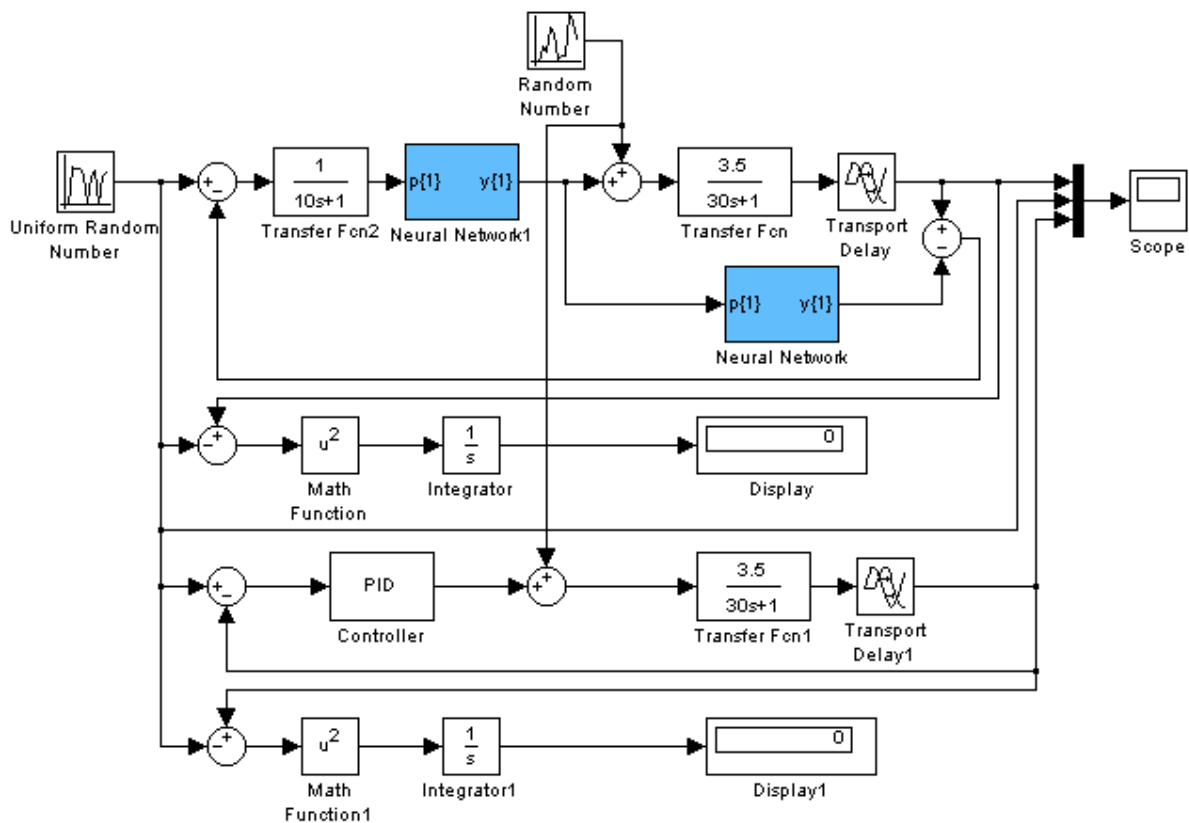


Рис. 3.30. Схема порівняння нейромережного управління зі внутрішньою моделлю та ПІД-управління за наявності неконтрольованих збурювань (пряма нейромережна модель реалізована динамічною нейронною мережею)

Другий спосіб побудови системи управління із внутрішньою моделлю полягає у використанні для побудови динамічної моделі об'єкта статичної нейронної мережі із зовнішньою лінією затримки сигналів [8]. Схема ідентифікації об'єкта управління в цьому випадку буде мати вигляд як на рисунку 3.32.

Для формування навчальних даних у ППП Simulink збираємо схему, яку наведено на рис. 3.33. Тривалість моделювання встановлюємо 2200 с.

Перед моделюванням задаємо наступні значення параметрів блоків: *Uniform Random Number*: minimum = 0; maximum = 1; initial seed = 0; sample time = 200; *To Workspace* і *To Workspace1*: limit data points to last: 2000; decimation: 1; sample time: 1; save format: *array* (масив).

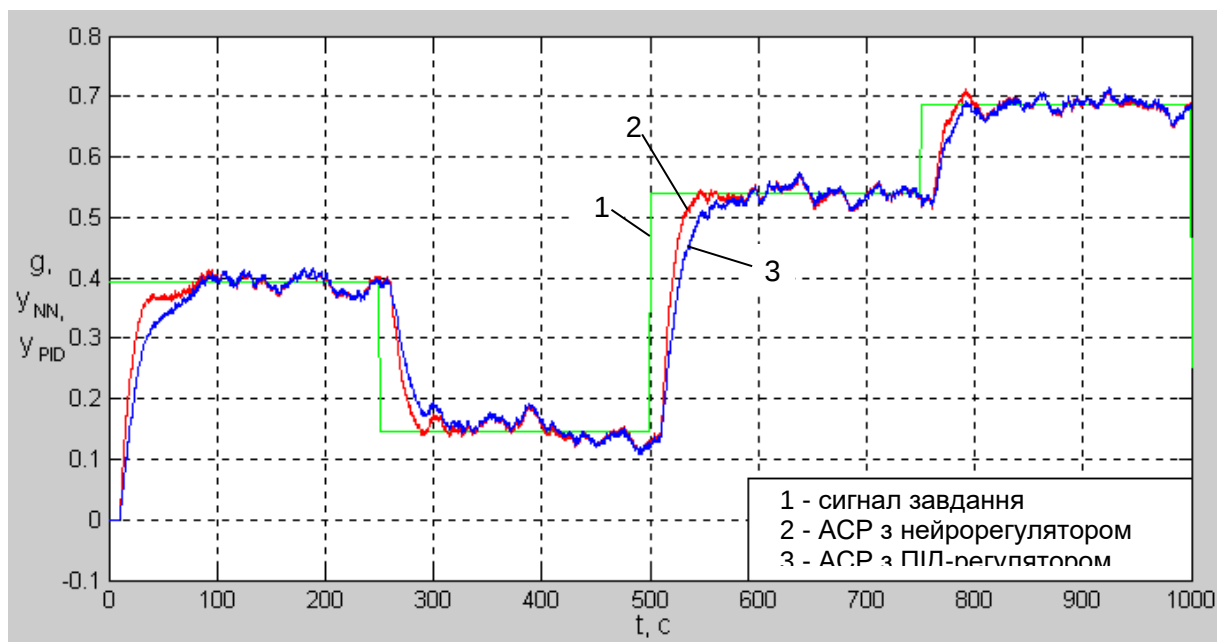


Рис. 3.31. Результати порівняння нейромережного управління зі внутрішньою моделлю (І спосіб) та ПІД-управління за наявності неконтрольованих збурювань, що впливають на об'єкт управління

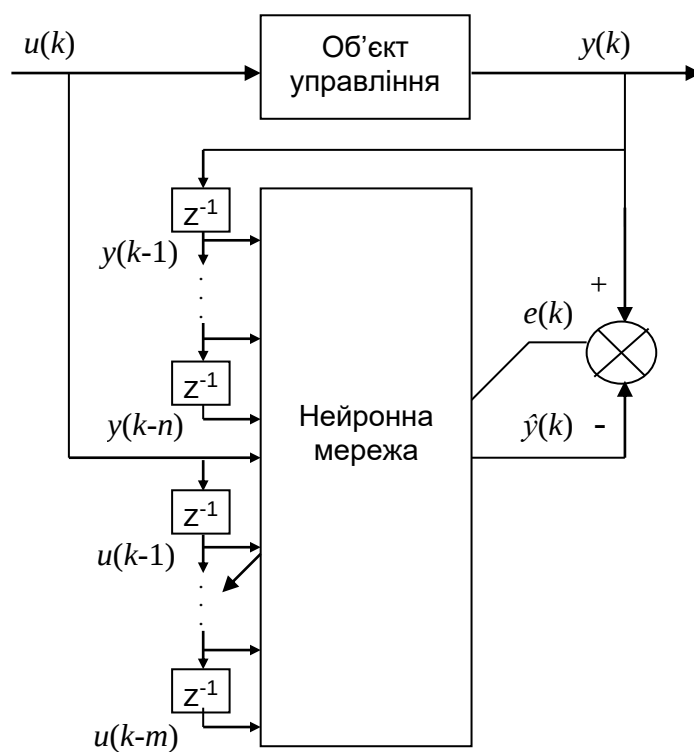


Рис. 3.32. Схема навчання статичної нейронної мережі прямій динаміці об'єкта управління: $u(k)$ – керуючий сигнал; y – вихідний сигнал; $\hat{y}(k)$ – оцінка виходу об'єкта; $e(k)$ – сигнал помилки; m – глибина лінії затримки вхідного сигналу; n – глибина затримки зворотного зв'язку за виходом

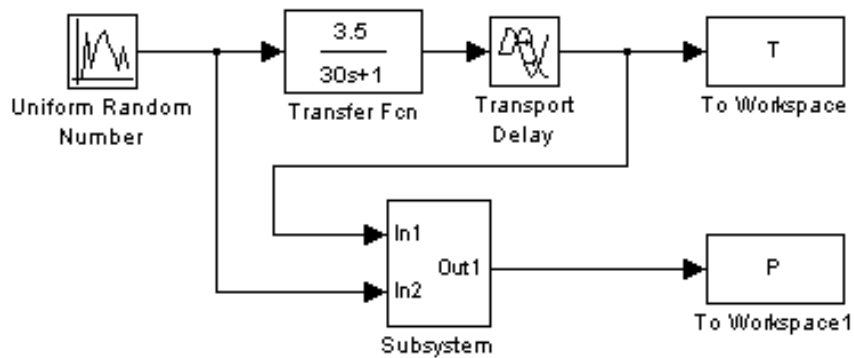


Рис. 3.33. Схема формування даних для навчання статичної нейронної мережі
прямій динаміці об'єкта управління

Параметри усіх блоків затримки *Unit Delay* (рис. 3.34) задаємо у такий спосіб: *initial conditions* (початкові умови): 0; *sample time* (крок дискретизації): 1.

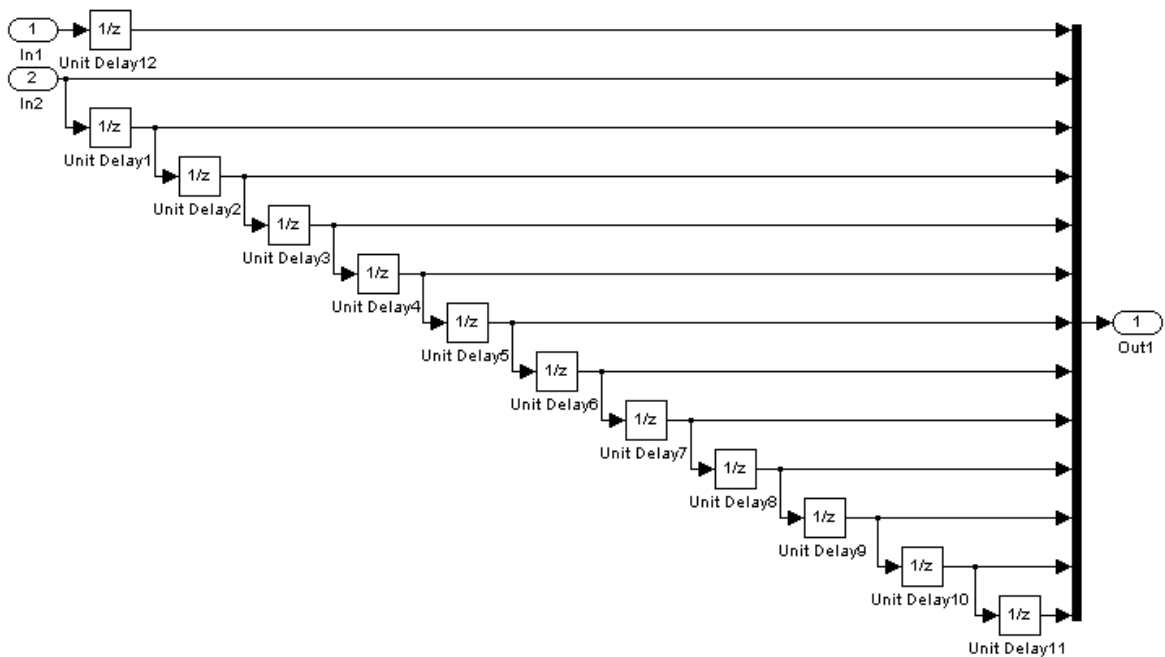


Рис. 3.34. Внутрішня структура блоку *Subsystem* (див. рис. 3.33)

У результаті моделювання схеми (рис. ЛР4.16) у робочій області MATLAB будуть створені масиви *P* і *T*, які для подальшого використання перетворюємо, застосувавши операцію транспонування:

```
>> P=P'; T=T';
```

Потім за допомогою GUI-інтерфейсу NNTool створимо лінійну одношарову статичну нейронну мережу, що складається з одного нейрона, і навчимо її на сформованих навчальних даних. При цьому значення параметра *learning rate* призначимо рівним 0.0001, а кількість епох (*epochs*) задамо рівною 550000.

Після цього необхідно сформувати S-модель навченої нейронної мережі, зібрати схеми, які наведено на рис. 3.35 та рис. 3.36, та здійснити тестування на тих самих сигналах, що використовувались для тестування схем за рис. 3.29 та рис. 3.30.

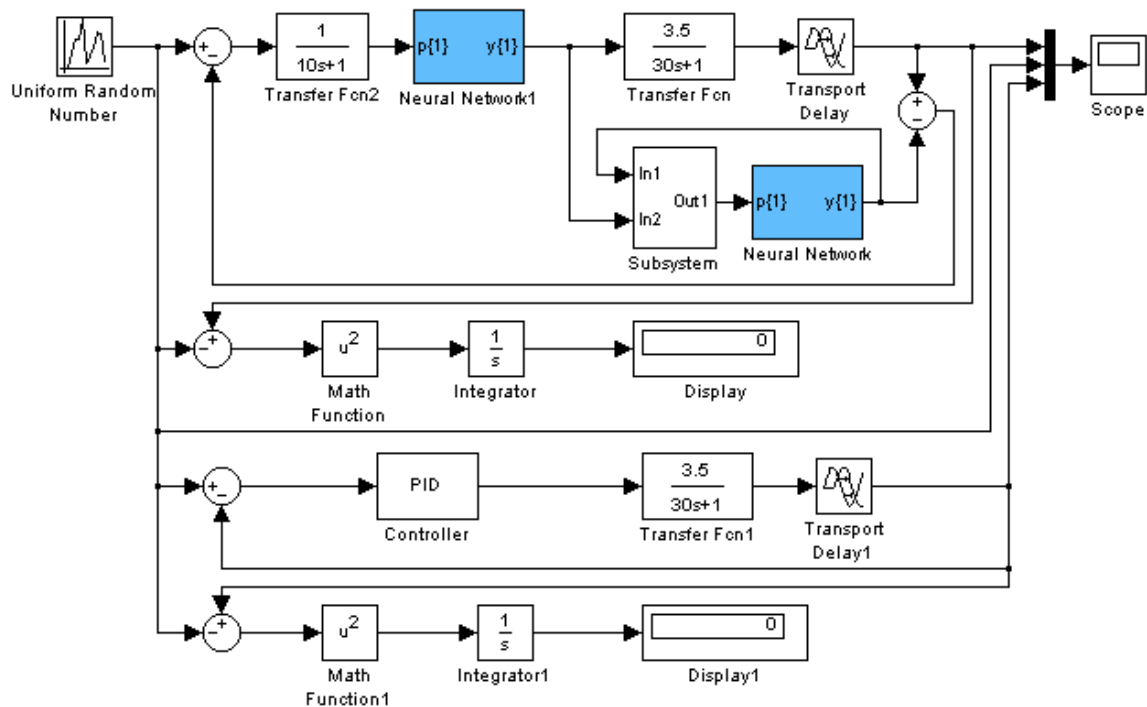


Рис. 3.35. Схема порівняння нейромережного управління зі внутрішньою моделлю і ПІД-управління за відсутності неконтрольованих збурювань

Графіки, що отримані в результаті тестування, повинні бути подібні отриманим в ході тестування схем за рис. 3.29 та 3.30.

Результати тестування схем, що наведені на рис. 3.35 та рис. 3.36, слід занести до табл. 3.9, порівняти їх із результатами тестування систем за рис. 3.29 та рис. 3.30 і зробити відповідні висновки.

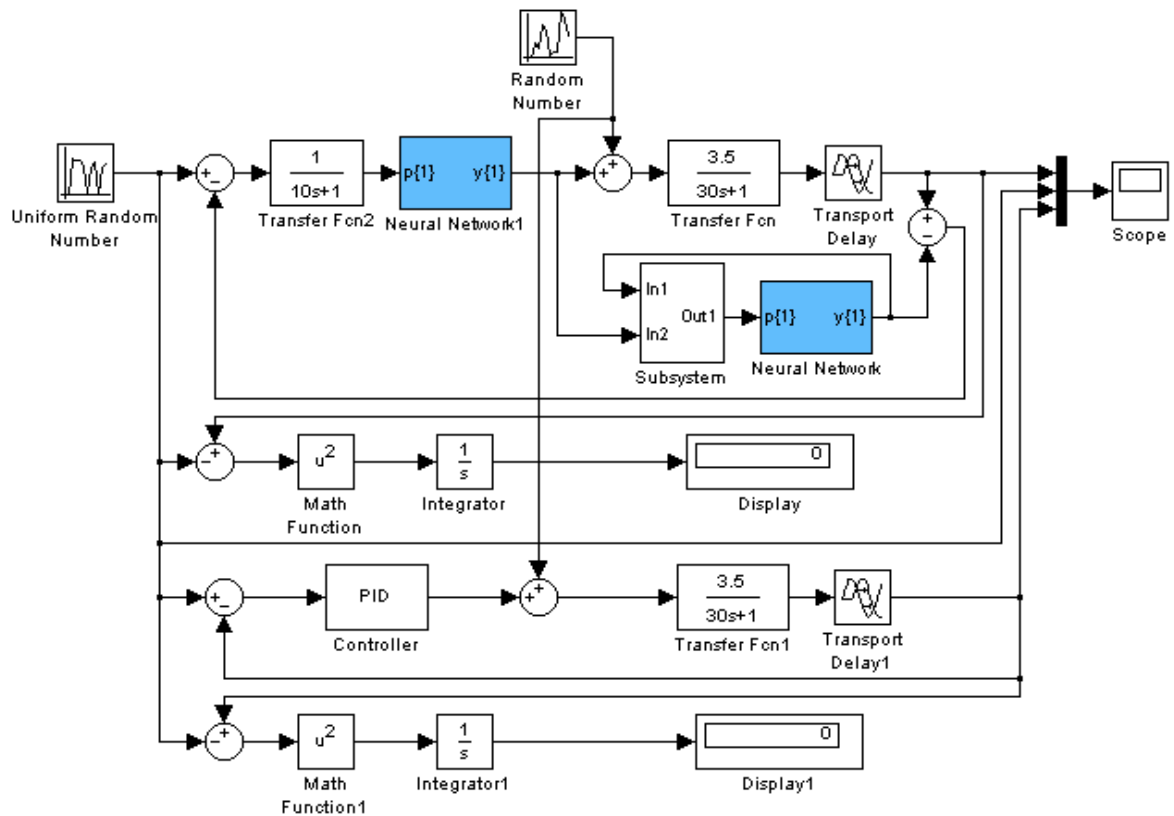


Рис. 3.36. Схема порівняння нейромережного управління зі внутрішньою моделлю і ПІД- управління за наявності неконтрольованих збурювань

Таблиця 3.9

Порівняння нейроуправління зі внутрішньою моделлю та ПІД- управління за відсутності та за наявності збурювань

Значення критерію якості управління		
Управління із внутрішньою моделлю		ПІД - управління
I спосіб	II спосіб	
За відсутності збурювань		
За наявності збурювань		

Зауваження. На практиці найбільше поширення одержав другий спосіб побудови прямої нейромережної моделі об'єкта управління, реалізація якого не викликає принципових утруднень. Це обумовлено тим, що для навчання статичної нейронної мережі успішно можуть застосовуватись

широко розповсюджені алгоритми навчання, які реалізують метод зворотного розповсюдження помилки.

Необхідно також відзначити, що схема управління із внутрішньою моделлю цілком підходить й для управління нестационарними об'єктами, зокрема об'єктами, у яких значення коефіцієнта передачі k_0 згодом стає відмінним від того значення, яке мало місце при одержанні їхніх нейромережних прямих та інверсних моделей.

Інакше кажучи, схему нейромережного управління із внутрішньою моделлю, на відміну від інших розглянутих схем нейромережного управління, цілком можна використовувати для управління не тільки стаціонарними, але й нестационарними об'єктами, причому як за відсутності неконтрольованих збурювань, що впливають на об'єкт управління, так і за їхньої наявності.

Бажаючи можуть переконатися в цьому, для чого пропонується на додаток до вже отриманих даних (табл. 3.9) здійснити моделювання роботи схем, що наведені на рис. 3.35 та рис. 3.36, для значень коефіцієнта передачі об'єкта $k_0 = 2,5$ та $k_0 = 4,5$, занести результати до табл. 3.10 та порівняти їх.

Таблиця 3.10

Порівняння нейроуправління зі внутрішньою моделлю й ПІД-управління за відсутності та за наявності збурювань в умовах зміни параметрів об'єкта

Значення критерію якості управління					
Управління з внутрішньою моделлю			ПІД - управління		
$k_0 = 2,5$	$k_0 = 3,5$	$k_0 = 4,5$	$k_0 = 2,5$	$k_0 = 3,5$	$k_0 = 4,5$
За відсутності збурювань					
За наявності збурювань					

БІБЛІОГРАФІЧНИЙ СПИСОК

1. Новіков Ф. А. Символічний штучний інтелект: математичні основи представлення знань. 2018. URL: <https://stud.com.ua/116384/prirodovnavstvo/praktikum> (дата звернення: 15.02.2024)
2. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности : монография / Г. К. Вороновский и др. Харьков : Основа, 1997. 112 с.
3. Субботін С. О. Нейронні мережі: теорія та практика : навч. посіб. Житомир : Вид. О. О. Євенок, 2020. 184 с.
4. Lopez C. P. Neural network architectures. Examples with MATLAB : Scientific Books, 2020. 209 p.
5. Медведев В. С., Потемкин В. Г. Нейронные сети. MATLAB 6 / под общ. ред. В. Г. Потемкина. Москва : ДИАЛОГ-МИФИ, 2002. 496 с.
6. Callan R. The Essence of Neural Networks. New Jersey : Prentice Hall Europe, 1999. 232 p.
7. Osowski S. Sieci neuronowe do przetwarzania informacji. Warszawa : Wydawnicza Politechniki Warszawskiej, 2013. 422 s.
8. Руденко О. Г., Бодянський Є. В. Штучні нейронні мережі. Харків : Компанія СМІТ, 2006. 404 с.
9. Тимошук П. В. Штучні нейронні мережі : навч. посіб. Львів : Львівська політехніка, 2011. 444 с.
10. Растрингин Л. А. Современные принципы управления сложными объектами. Москва : Сов. радио, 1980. 232 с.

ТРАДИЦІЙНІ МЕТОДИ АВТОМАТИЧНОГО УПРАВЛІННЯ СТАНОМ ДИНАМІЧНИХ СИСТЕМ (КОРОТКИЙ ОГЛЯД)

Д.1. Загальна концепція автоматичного управління

Процес управління у класичному розумінні полягає в організації цілеспрямованого впливу на певний об'єкт управління (ОУ), що переводить його у потрібний (цільовий) стан, якщо, звісно, ОУ піддається такому цілеспрямованому впливу, тобто є керованим.

Реалізацію управління здійснює система управління (СУ). Під СУ розуміють усі необхідні алгоритми обробки інформації та засоби їх реалізації. Узагальнена схема СУ наведена на рис. Д.1 [10].

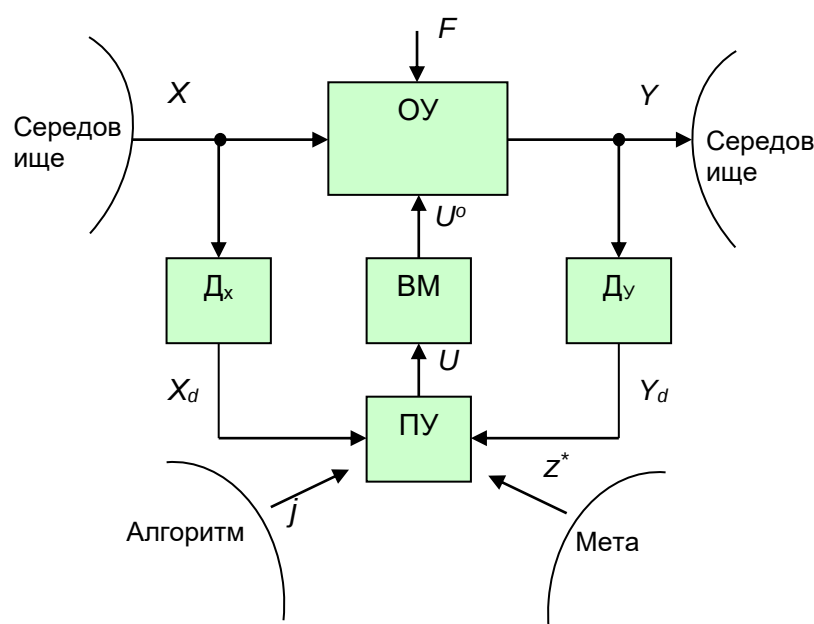


Рис.Д.1. Узагальнена схема системи управління

На даному рисунку застосовані такі позначення: D_x та D_y – датчики, що вимірюють стан середовища X та об'єкта управління Y відповідно; ПУ – пристрій управління, що виробляє керуючий вплив U ; ВМ – виконавчий механізм, що формує згідно із сигналом пристрою управління власне вплив

на об'єкт U^0 ; F – неконтрольовані збурення, тобто усі неконтрольовані внутрішні та зовнішні фактори, що впливають на стан об'єкта.

Для цілеспрямованого функціонування ПУ йому крім інформації від D_x та D_u необхідно також задати мету управління Z^* (тобто чого слід прагнути в процесі управління) та алгоритм управління j (спосіб досягнення заданої мети).

Під алгоритмом управління розуміють чітке, однозначне правило, що необхідно робити, щоб досягти мети Z^* у ситуації $I = \{X_d Y_d\}$, що склалась.

Основними факторами, що притаманні будь-якій СУ, є:

- мета управління Z^* ;
- інформація про стан об'єкта та середовища I ;
- вплив на об'єкт (власне управління) U ;
- алгоритм управління j .

Усі труднощі побудови СУ визначаються складністю об'єкта. Наведемо основні ознаки складного ОУ [10].

1). **Відсутність математичного опису**, який надає алгоритм обчислення стану Y об'єкта за його входами: керованим U та некерованим, але таким, що спостерігається X .

2). **Недетермінованість (стохастичність)**, що обумовлена не стільки наявністю джерел випадкових перешкод в об'єкті, скільки складністю об'єкта та пов'язаними з цим неминучим розмаїттям всякого роду другорядних (з огляду на управління) процесів, якими нехтують при побудові математичної моделі.

3). **Нестационарність**, яка проявляється у дрейфі характеристик об'єкта, тобто у його еволюції в часі.

Структура СУ складним об'єктом зазвичай містить стратегічний, тактичний та виконавчий рівні. На виконавчому, нижньому рівні СУ найчастіше вирішується задача регулювання – задача змінення потрібним чином (або утримання незмінними) якихось координат ОУ. Пристрій управління в даному випадку називається регулятором й у сукупності з ОУ утворює систему автоматичного регулювання (САР).

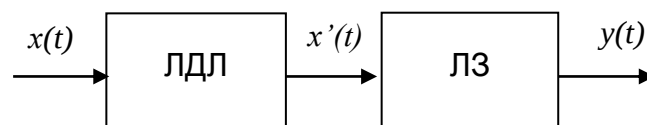


Рис.Д.2. Типова структура математичної моделі ОУ

На практиці точна модель ОУ зазвичай є невідомою. Проте, для оцінних розрахунків достатньо мати модель, що хоча б приблизно описувала властивості ОУ. У першому наближенні математичний опис багатьох промислових об'єктів (електричних, електромеханічних, теплотехнічних тощо) можна уявити у вигляді послідовного з'єднання лінійної динамічної ланки (ЛДЛ) та ланки чистого запізнювання (ЛЗ) (рис. Д2).

Функція передачі ЛДЛ зазвичай має вигляд

$$W(p) = \frac{k \cdot e^{-p\tau}}{(T_1 p + 1)(T_2 p + 1)}. \quad (\text{Д.1})$$

Розглянемо різні варіанти побудови регуляторів автоматичних систем, які не потребують знання точної моделі ОУ.

Д1.2. Релейне регулювання

Розглянемо замкнену одноконтурну систему автоматичного регулювання з релейним регулятором (рис. Д.3). Релейний регулятор являє собою статичну ланку з релейною характеристикою.

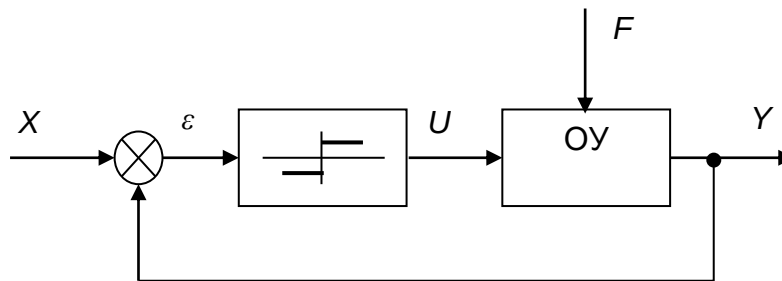


Рис.Д.3. Структура САР з релейним регулятором

Деякі варіанти найпростіших характеристик релейних регуляторів наведені на рис. Д.4.

Релейні САР можна розділити на два класи. Режимом нормального функціонування систем першого класу є режим автоколивань, тоді як у системах другого класу автоколивання відсутні завдяки наявності інтегратора у складі ОУ та використання трипозиційних регуляторів із зоною нечутливості.

Приклади релейних систем першого класу: стабілізатори температури, що застосовуються в електронагрівальних приладах, електричних прасках та

холодильниках. Приклад релейної системи другого класу: система регулювання кута повороту, що відстежує.

Відомо, що релейні системи можуть стабілізувати ОУ тільки у разі дотримання таких умов:

- керуючі впливи та збурення є обмеженими;
- система не містить ланок запізнювання (або запізнення значно менше за сталі часу ОУ);
- лінійна частина системи є мінімально-фазовою і має відносний порядок (різницю між порядком чисельника і знаменника функції передачі) не вище двох.

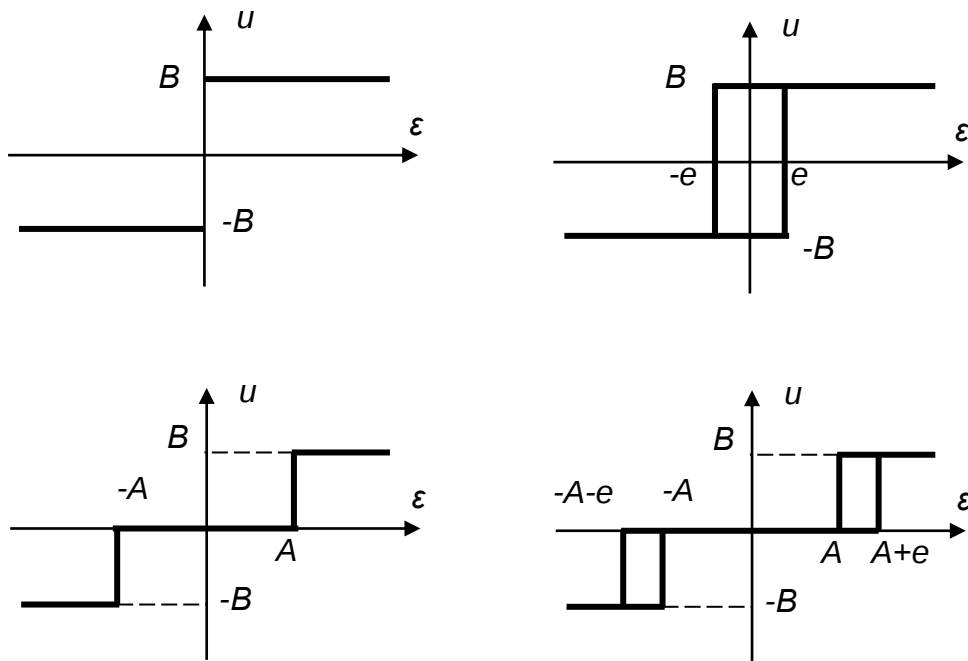


Рис.Д.4. Характеристики релейних регуляторів

У релейних САР керуючі впливи завжди мають граничні значення. Через це такі САР мають велику швидкість змінення керованих величин, а отже іноді можуть бути оптимальними за швидкодією.

Відмінними рисами релейних регуляторів є простота, надійність, часто не дуже високі показники точності регулювання, в іноді взагалі нездатність стабілізувати ОУ, що обмежує їх застосування.

Д1.3. Системи з лінійними регуляторами

Лінійні регулятори являють собою динамічні ланки, що описуються лінійними диференціальними рівняннями (рис. Д.5). Найбільш розповсюдженими регуляторами є ПІД-регулятори, що мають такий математичний опис

$$u(t) = K_p \varepsilon(t) + K_I \int_0^t \varepsilon(t) \cdot dt + K_D \frac{d}{dt} \varepsilon(t), \quad (\text{Д.2})$$

де $u(t)$ та $\varepsilon(t)$ – сигнали на виході та на вході регулятора відповідно; K_p , K_I та K_D – коефіцієнти пропорційної, інтегральної та диференціальної складових відповідно.

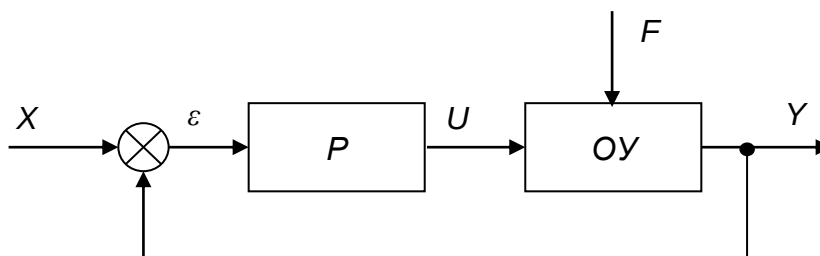


Рис.Д.5. Структура САР з лінійним регулятором

Окремим випадком ПІД-регуляторів є І-регулятори (для яких, на відміну від ПІД-регуляторів, K_p та K_D дорівнюють нулю), П-регулятори (K_I та K_D дорівнюють нулю) та ПД-регулятори (K_I дорівнює нулю).

Ідея ПІД-регулятора є емпіричною: пропорційна складова відображає помилку системи у поточний момент часу, інтегральна складова – історію змінення помилки САР, а диференціальна складова – передбачення майбутньої помилки управління.

Опрацьована велика кількість достатньо ефективних методів налаштування ПІД-регуляторів.

До ПІД-регулятора можна прийти, наприклад, наступним чином. Нехай ОУ описується функцією передачі

$$W_{oy}(p) = \frac{k \cdot e^{-p\tau}}{(T_1 p + 1)(T_2 p + 1)},$$

параметри об'єкта незмінні, а параметр τ є достатньо малим. Функція передачі регулятора згідно з (Д.2) має вигляд

$$W_p(p) = K_p + \frac{K_I}{p} + pK_D. \quad (\text{Д.3})$$

Вимагатимемо, щоб функція передачі замкненої системи відповідала б функції передачі ланки чистого запізнювання зі сталою часу τ , тобто щоб система відтворювала на виході ОУ задане значення керованої величини з запізненням τ

$$W_{cy}(p) = \frac{W_{ov}(p)W_p(p)}{1 + W_{ov}(p)W_p(p)} = e^{-p\tau}. \quad (\text{Д.4})$$

Тоді функція передачі регулятора такої майже ідеальної СУ визначатиметься через функцію передачі ОУ виразом

$$W_p(p) = \frac{e^{-p\tau}}{(1 - e^{-p\tau})W_{ov}(p)}. \quad (\text{Д.5})$$

Беручи до уваги (Д.2) та те, що за умови малого τ

$$e^{-p\tau} \approx 1 - p\tau,$$

отримуємо

$$\begin{aligned} W_p(p) &= \frac{1 - p\tau}{(1 - 1 + p\tau)W_{ov}(p)} = \frac{1 - p\tau}{p\tau \cdot W_{ov}(p)} = \\ &= \frac{(1 - p\tau)(T_1p + 1)(T_2p + 1)}{p\tau \cdot k \cdot (1 - p\tau)} = \frac{(T_1p + 1)(T_2p + 1)}{p\tau k} = \\ &= p \frac{T_1T_2}{\tau \cdot k} + \frac{1}{p\tau \cdot k} + \frac{T_1 + T_2}{\tau \cdot k}. \end{aligned} \quad (\text{Д.6})$$

Отже, з огляду на (Д.3) квазіідеальний регулятор матиме такі параметри:

$$K_p = \frac{T_1 \cdot T_2}{\tau \cdot k}, \quad K_I = \frac{1}{\tau \cdot k}, \quad K_D = \frac{T_1 + T_2}{\tau \cdot k}. \quad (\text{Д.7})$$

ПД-регулятори довели свою ефективність в управлінні різноманітними процесами. Більше 80% регуляторів, що наразі застосовуються в промисловості, є саме ПД-регуляторами.

Проте, необхідно зазначити, що для об'єктів зі змінними параметрами, значним часовим запізненням, суттєвими нелінійностями та великими перешкодами ПД-регулятори не завжди можуть забезпечити необхідну якість управління.

Д1.4. Системи регулювання з додатковими інформаційними каналами

Застосування лінійних регуляторів певної структури досяжна динамічна точність САР є обмеженою і не завжди задовольняє висунуті вимоги. У випадку, коли подальше збільшення динамічної точності регулювання не може бути досягнуте покращенням налаштувань регулятора, доводиться вживати заходи, спрямовані на формування додаткових керуючих впливів завдяки використанню більш повної та своєчасної інформації про стан ОУ.

Найчастіше використовують можливість реагування на зміни збурень, які піддаються контролю (вимірювання). У таких САР можна організувати так зване комбіноване управління з додатковим каналом управління за збуренням, як це ілюструє рис. Д.6. За таким методом побудови САР одночасно з контуром регулювання за відхиленням, що реалізується завдяки від'ємному зворотному зв'язку, функціонує також розімкнений контур регулювання за збуренням.

Можна довести, що за умови

$$\varphi(p) = \frac{1}{W_{ov}^f(p)}, \quad (Д.4)$$

де $W_{ov}^f(p)$ - функція передачі ОУ за збуренням, помилка ε не залежить від впливу збурення f , тобто система є інваріантною відносно збурення.

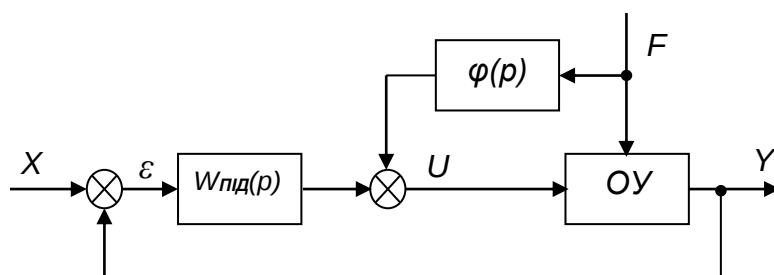


Рис.Д.6. Структура САР комбінованого управління з додатковим каналом регулювання за збуренням

Зважаючи на те, що інваріантність реалізується за умови (Д.4), а функція $\varphi(p)$ може бути визначена лише для окремих збурень, на практиці може бути здійснена не повна, а лише часткова інваріантність, що у низці випадків зводить нанівець переваги інваріантних систем.

Д1.5. Нелінійні регулятори

В реальних умовах розглянуті раніше релейні та лінійні регулятори не завжди дозволяють досягти бажаної якості управління, що призвело до розробки регуляторів, які реалізують складні нелінійні динамічні оператори.

Перешкодою широкому застосуванню нелінійних регуляторів є відсутність універсальних методів їх синтезу. Тому використання нелінійних регуляторів на практиці обмежене окремими випадками, для яких доведена їх ефективність.

Розглянемо традиційні підходи до побудови нелінійних регуляторів.

Найпростіші нелінійні регулятори були розглянуті у розділі Д.2. Їх логічним продовженням є СУ зі змінною структурою, в яких відбувається стрибкоподібне зміння зв'язків між функціональними елементами регулятора залежно від змінних стану СУ.

На рис. Д.7 наведений приклад СУ зі змінною структурою. Вона містить ОУ, два регулятори P_1 та P_2 , а також пристрій управління ПУ, який стрибкоподібно змінює структуру системи після зміння сигналу неузгодженості ε .

Застосування принципу змінної структури дозволяє отримати СУ, які набувають нових унікальних властивостей. Наприклад, застосовуючи в СУ на рис. Д.7 регулятор P_1 , що забезпечує перехідний процес малої тривалості, але з більшим перерегулюванням, та регулятор P_2 , що забезпечує аперіодичний перехідний процес з великою тривалістю, можна отримати систему з аперіодичним перехідним процесом малої тривалості.

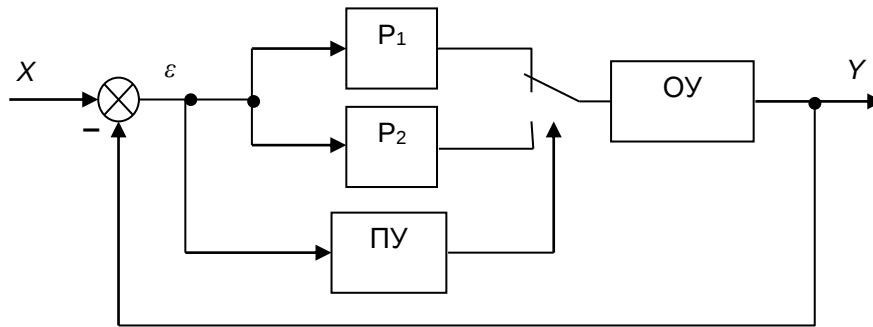


Рис.Д.7. Схема системи управління зі змінною структурою

Принцип змінної структури забезпечує високоякісне управління об'єктами з незмінними параметрами, а у випадку змінення параметрів об'єкта в широкому діапазоні або за наявності суттєвих збурень дозволяє побудувати СУ малочутливу до таких змінень.

Д1.6. Адаптивні системи

Зазвичай під час проектування СУ передбачається, що математична модель ОУ є відомою (принаймні з точністю до параметрів). У результаті визначають структуру та усі параметри САР, що задовольняють висунуті вимоги. Проте не завжди існує така визначеність. Часто параметри ОУ, а іноді й структуру його математичного опису не можна вважати достовірними. У таких випадках неможливо створити якісну СУ з визначеними фіксованими параметрами й виникає необхідність у створенні **адаптивної** САР, параметри і навіть структура якої автоматично налаштовуються в процесі функціонування.

Адаптивні системи на відміну від звичайних систем містять контур само налаштування. При цьому можливими є два варіанти:

- 1) після налаштування регулятора контур само налаштування більше не бере участі у роботі системи й відключається;
- 2) параметри об'єкта і збурень постійно змінюються і, якщо ці змінення є значними, то для отримання необхідної якості управління синхронно змінюються параметри регулятора.

На рис. Д.8 наведена схема адаптивної СУ зі замкненим контуром самоналаштування. Регулятор адаптується залежно від параметрів нестационарного ОУ, які визначаються за допомогою блока ідентифікації БІ.

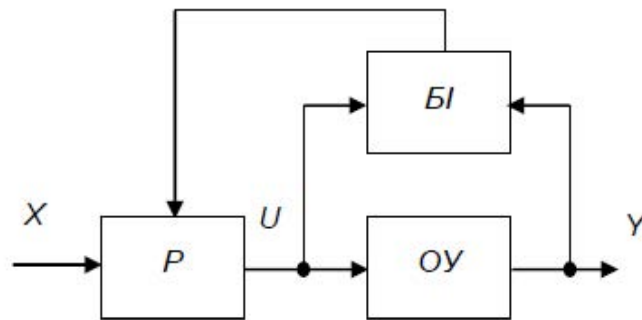


Рис.Д.8. Схема адаптивної СУ

Суттєвим є спосіб визначення параметрів ОУ: шляхом подання на ОУ спеціальних (пробних) впливів або шляхом спостереження природного ходу процесу управління. У зв'язку з цим розрізняють системи пошукового самоналаштування або аналітичні системи само налаштування.

До недоліків адаптивних систем відносять те, що більшість алгоритмів адаптації отримані для умов відсутності неконтрольованих збурень під час ідентифікації. До того ж практично усі алгоритми адаптації є ефективними, якщо виконується гіпотеза квазістаціонарності ОУ впродовж налаштування регулятора та відсутні збурення, що зникають. Необхідно також зазначити, що існуючі алгоритми адаптації є доволі складними в реалізації.

Розглянуті СУ з релейними та лінійними регуляторами, системи зі змінною структурою, системи з ПІД-регуляторами тощо відносять до традиційних СУ. У них не застосовуються сучасні інформаційні технології, такі, як нейронні мережі, нечітка логіка, генетичні алгоритми, експертні системи тощо, і це можна вважати їх відмінною особливістю. Застосування зазначених технологій в СУ може суттєво покращити якість управління складними об'єктами.

Навчальне видання

Потап Олег Юхимович, Маслікова Світлана Сергіївна,

Кисляков Володимир Геннадійович

СИСТЕМИ УПРАВЛІННЯ ЗІ ШТУЧНИМ ІНТЕЛЕКТОМ. НЕЙРОМЕРЕЖНІ СИСТЕМИ УПРАВЛІННЯ

Навчальний посібник

Електронне видання

Відповідальний редактор О. Ю. Потап

Комп'ютерна верстка О. Ю. Потап

Дизайн обкладинки О. Ю. Потап

Експертний висновок склав канд. техн. наук, М. Д. Зінченко

Зареєстровано НМВ УДУНТ (№ 709 від 14.03.2024)

Формат 60x84 $\frac{1}{16}$. Ум. друк. 6,16. Обл.-вид. арк. 3,2.
Зам. № 26

Видавець: Український державний університет науки і технологій.
вул. Лазаряна, 2, ауд. 2216, ауд. 263 м. Дніпро, 49010.

Свідоцтво суб'єкта видавничої справи ДК № 7709 від 14.12.2022

