

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут «Українська інженерго-педагогічна академія»
Кафедра Автоматизації, метрології та енергоефективних технологій

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

СИСТЕМИ УПРАВЛІННЯ ОСНОВНИМИ МЕХАНІЗМАМИ
РОТОРНОГО ЕКСКАВАТОРА З НЕЙРОРЕГУЛЯТОРОМ NN
PREDICTIVE CONTROLLER. ТЕМА КОМПЛЕКСНА. СПЕЦІАЛЬНА
ЧАСТИНА. СИНТЕЗ І ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖЕВОЇ
СИСТЕМИ УПРАВЛІННЯ МЕХАНІЗМОМ ОБЕРТАННЯ КОЛЕСА
РОТОРНОГО ЕКСКАВАТОРА

(тема кваліфікаційної роботи)

Виконав: студент 2 курсу, групи ДЕА-А23мг
спеціальності: 174 Автоматизація, комп'ютерно-
інтегровані технології та робототехніка

(код і найменування спеціальності)

_____/ Артем ФОМЕНКО
(підпис) (ім'я та прізвище)

Керівник _____/ Тетяна ВАСИЛЕЦЬ
(підпис) (ім'я та прізвище)

Рецензент _____/ Анатолій ТАРАСЕНКО
(підпис) (ім'я та прізвище)

«До захисту допущено»

Завідувач кафедри _____/ Геннадій КАНЮК
(підпис) (ім'я та прізвище)

Нормоконтроль _____/ Євген КЛЮЧКА
(підпис) (ім'я та прізвище)

Секретар ЕК _____/ Євген КЛЮЧКА
(підпис) (ім'я та прізвище)

Харків – 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний університет імені В.Н.Каразіна

Навчально-науковий інститут «Українська інженерно-педагогічна академія»
Кафедра Автоматизації, метрології та енергоефективних технологій
рівень вищої освіти другий (магістрський)
Спеціальність 174 Автоматизація, комп'ютерно-інтегровані технології та
робототехніка
Освітня програма Автоматизація та комп'ютерно-інтегровані технології

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., проф. Геннадій КАНЮК
(науковий ступінь, вчене звання, ім'я, прізвище)
« » 2024р.

ЗАВДАННЯ
на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Артем ФОМЕНКО
(ім'я, прізвище)

1. Тема роботи Системи управління основними механізмами роторного екскаватора з нейрорегулятором NN Predictive Controller. Тема комплексна. Спеціальна частина. Синтез і дослідження нейромережевої системи управління механізмом обертання колеса роторного екскаватора

затверджена наказом по університету № 4801-5/3704 від “21” листопада 2024 р.

2. Термін здачі закінченої роботи “ 5 ” грудня 2024 р.

3. Вихідні дані до роботи Теоретична продуктивність по рихлій масі $Q=1700 \text{ м}^3 \cdot \text{год}$; діаметр роторного колеса $D_p=8 \text{ м}$; бічне зусилля різання $F_b=1,2 \cdot 10^6 \text{ Н}$; питоме зусилля копання $k_f=3,8 \cdot 10^4 \text{ кг/м}^2$; радіус різання $R_0=23,3 \text{ м}$; кутова швидкість повороту платформи $\omega_{пл}=2,9 \cdot 10^{-2} \text{ с}^{-1}$; маса поворотної платформи $m_{пл}=500 \cdot 10^3 \text{ кг}$; маса приймальної частини $m_{пч}=127 \cdot 10^3 \text{ кг}$; маса відвальної частини $m_{вч}=120 \cdot 10^3 \text{ кг}$; частота обертання ротора $\omega_p=6 \text{ рад/хв}$; момент інерції поворотної платформи $J_{пл}=17 \cdot 10^7 \text{ кг} \cdot \text{м}^2$; діаметр роликів $D_{кр}=8,7 \text{ м}$. Перерегулювання не більше 10%, час регулювання до 5 с.

4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму обертання колеса роторного екскаватора. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити статичні і динамічні характеристики системи. Розробити математична модель двомасової системи. Виконати моделювання на ЕОМ. Розробити структурну схему нейромережевої системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання нейромереженої системи.

5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістерська кваліфікаційна робота. Формальні ознаки. 2. Механізм обертання колеса. . Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейронні мережі. Ідентифікація об'єктів. 6. Системи нейрорегулювання. Структурні схеми. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси. 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14-15. Магістерська кваліфікаційна робота. Висновки.

6. Консультант:

Розділ	Консультант	Підпис, дата		Оцінка (бали)
		Завдання видав	Завдання прийняв	

7. Дата видачі завдання “ 9 ” вересня 2024 р.

Керівник

(підпис)

Тетяна ВАСИЛЕЦЬ.
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Артем ФОМЕНКО
(ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму обертання колеса роторного екскаватора.	09.09.2024-18.09.2024	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження статичних і динамічних характеристик системи.	19.09.2024-15.10.2024	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2024-05.11.2024	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2024-25.11.2024	
5	Оформлення роботи	26.11.2024-05.12.2024	

Студент (ка)

(підпис)

Артем ФОМЕНКО
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 190 сторінок, 61 рисунок, 4 таблиці, 47 використаних джерела, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму обертання колеса роторного екскаватора для підвищення показників якості функціонування системи на основі нейрорегуляторів.

Розрахунковим шляхом з застосуванням фундаментальних положень теорії автоматичного управління визначаються параметри системи управління. Методи математичного моделювання використані для побудови математичних моделей роботи багатомасової системи. З використання методів теорії штучних нейронних мереж виконується синтез нейрорегулятора для управління двомасовою електромеханічною системою. Методом комп'ютерного моделювання із застосуванням пакету прикладних програм MATLAB досліджуються динамічні характеристики нейромережевої системи.

В результаті виконання магістерської кваліфікаційної роботи науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму обертання колеса роторного екскаватора. Розроблена система управління координат електроприводу механізму. Розраховані параметри системи автоматичного управління. Проведено моделювання системи на ЕОМ.

Розроблена математична модель системи з урахуванням пружних елементів механічної частини електроприводу. Розраховані динамічні характеристики двомасової системи.

Розроблена структура нейромережевої системи управління і виконаний синтез нейрорегулятора NN Predictive Controller в системі MATLAB. Проведено моделювання нейромережевої системи управління з нейрорегулятором на ЕОМ. Встановлено, що синтезована нейромережева система має високі показники якості функціонування.

Основні положення магістерської роботи використовуються у навчальному процесі кафедри АМЕТ ННІ «УПА» ХНУ ім. В.Н. Каразіна при читанні лекцій, виконанні курсового проєкту, проведенні практичних занять.

Основні положення та наукові результати апробовані на науково-практичній конференції студентів УПА (Харків, 2024 р.).

Ключові слова: ШТУЧНИЙ НЕРОН, НЕЙРОННІ МЕРЕЖІ, РОТОРНИЙ ЕКСКАВАТОР, ЕЛЕКТРОПРИВОД МЕХАНІЗМУ ОБЕРТАННЯ КОЛЕСА, СИСТЕМА УПРАВЛІННЯ З ПІДСУМОВУЮЧИМ ПІДСИЛЮВАЧЕМ, ОДНОМАСОВА І ДВОМАСОВА СИСТЕМИ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОРЕГУЛЯТОР

ABSTRACT

An explanatory message consists of: 190 pages, 61 figures, 4 tables, 47 used sources, 5 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the mechanism of rotation of the wheel of a rotary excavator to improve the quality indicators of the system functioning on the basis of neuroregulators.

By calculation with the use of fundamental concepts of automatic control theory determines the parameters of the control system. Mathematical modeling techniques used to build mathematical models multimass system. With the use of methods of theory of artificial neural networks the synthesis of neuroregulators is executed for a control the two-mass electromechanics system. The method of computer simulation using MATLAB application package investigates the dynamic characteristics of neural systems.

As a result of the master's qualification work scientifically grounded perspective the use of neural network technology to control multimass electromechanical system of the mechanism of rotation of the wheel of a rotary excavator. The control system of coordinates of mechanism electric drive is worked out. The parameters of automatic control system are calculated. The simulation of the system is executed on a computer.

Worked out the mathematical model of the system which takes into account the elastic elements of the mechanical part of the electric drive is developed. Dynamic descriptions of the two-mass system are calculated.

Worked out the structure of the network control system and the synthesis of neuroregulator NN Predictive Controller is executed in the system of MATLAB. The simulation of neural network control system with neuroregulation is conducted on a computer. It is set that synthesized the neural network system has high indexes of quality of functioning.

The main provisions of the master's thesis are used in the educational process of the Department of AMET of the Scientific and Technological Research Institute "UIPA" of the V.N. Karazin KhNU when giving lectures, completing a course project, and conducting practical classes.

Basic provisions and scientific results tested at the scientific-practical conference of students of the UIPA (Kharkov, 2024 y.).

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, ROTARY EXCAVATOR, ELECTRIC DRIVE OF THE MECHANISM OF ROTATION OF THE WHEEL, CONTROL SYSTEM WITH SUMMARIZING AMPLIFIER, ONE-MASS AND TWO-MASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

- P_n – номінальна потужність;
 U_n – номінальна напруга;
 I_n – номінальний струм;
 M_n – номінальний момент;
 n_n – номінальна швидкість обертання;
 Φ_n – номінальний магнітний потік;
 R – активний опір;
 L – індуктивність;
 C – ємність;
 J – момент інерції;
 k_d – коефіцієнт посилення двигуна;
 T_e – електромагнітна постійна часу електроприводу;
 T_m – електромеханічна постійна часу електроприводу;
 $W(p)$ – передатна функція;
 c – коефіцієнт жорсткості;
 β – коефіцієнт внутрішнього в'язкого тертя;
 A – матриця стану;
 B – матриця управління;
 C – матриця виходу;
 $\vec{X}(t)$ – вектор стану системи;
 $\vec{U}(t)$ – вектор управління;
 $\vec{Y}(t)$ – вектор виходу;
САУ – система автоматичного управління;
М – двигун.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ В СИСТЕМАХ УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	13
1.1 Основні поняття і визначення нейронних мереж	13
1.2 Огляд структур систем нейрорегулювання. Вибір структури системи	19
1.3 Загальні відомості про роторні екскаватори	30
1.4 Обґрунтування вибору системи електроприводу роторного колеса екскаватора	34
1.5 Нейромережевий підхід до управління. Постановка задачі дослідження	36
ВИСНОВКИ ДО РОЗДІЛУ 1	40
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ОБЕРТАННЯ КОЛЕСА РОТОРНОГО ЕКСКАВАТОРА ...	42
2.1 Основні технічні характеристики роторного екскаватора. Вимоги до системи управління	42
2.2 Розрахунок потужності і вибір електродвигуна механізму обертання колеса роторного екскаватора	44
2.3 Вибір основного електроустаткування електроприводу	46
2.4 Розрахунок і побудова статичних характеристик розімкненої і замкненої систем	47
2.5 Дослідження динамічних характеристик системи	60
ВИСНОВКИ ДО РОЗДІЛУ 2	74
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ	75
3.1 Математична модель двомасової системи	76
3.2 Розрахунок перехідних процесів двомасової системи	79
ВИСНОВКИ ДО РОЗДІЛУ 3	86

РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	87
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	87
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	89
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	92
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	136
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	138
ВИСНОВКИ ДО РОЗДІЛУ 4	144
ВИСНОВКИ	146
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	148
ДОДАТОК А. МОДЕЛІ НЕЙРОНА.....	153
А.1 Структурна схема нейрона.....	153
А.2 Функції активації.....	154
А.3 Нейрон з векторним входом.....	157
ДОДАТОК Б. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ.....	159
Б.1 Одношарові мережі.....	159
Б.2 Багатошарові мережі	161
Б.3 Мережі з прямою передачею сигналу	164
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	166
В.1 Сутність процесу навчання	166
В.2 Алгоритми навчання нейронних мереж.....	168
В.3 Метод зворотного розповсюдження помилки.....	172
В.4 Перенавчання і узагальнення нейронних мереж.....	182
ДОДАТОК Г. ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	185
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	190

ВСТУП

Актуальність теми. Однією з провідних галузей народного господарства є гірничодобувна промисловість. Підвищенням продуктивності праці в гірничодобувній промисловості є одним з основних завдань розвитку економіки нашої країни. У зв'язку з цим на відкритих гірських роботах в даний час здійснюється широка програма по технічному переозброєнню і оснащенню виробництва більш довершеною технікою.

Роторний екскаватор є основною машиною, що входить в технологічний комплекс розробки корисних копалини відкритим способом. В даний час накопичений великий досвід проектування, виробництва, експлуатації і модернізації електроприводів основних і допоміжних механізмів роторних екскаваторів. Проте комплексне рішення задачі підвищення ефективності роторних екскаваторів неможливе без вдосконалення систем управління електроприводами. Саме якість систем управління електроприводами визначає можливість реалізації оптимальних режимів роботи механічного устаткування, забезпечення максимальної продуктивності і безперебійної роботи всіх механізмів..

Зростаючі вимоги до тактико-технічних характеристик та якості систем управління змушують шукати нові підходи до вдосконалення автоматичних систем. У результаті інтенсивних досліджень останніх десятиліть широке застосування отримали методи адаптивного та робастного регулювання. Ці методи забезпечують стійкість систем із змінною динамікою, базуються на відомій моделі об'єкта та не потребують точного знання її параметрів. Методи адаптивного регулювання налаштовують регулятор за рахунок оцінки параметрів невідомого об'єкта, тоді як робастні регулятори залишаються постійними і забезпечують стійкість системи лише в межах допустимих змін параметрів об'єкта. Проте проблема синтезу адаптивних систем регулювання для невідомих, нелінійних і змінних у часі об'єктів залишається нерозв'язаною як у теоретичному, так і в практичному аспектах. Область застосування методів робастного регулювання обмежується, переважно, лінійними системами та деякими спеціальними кла-

сами нелінійних систем. Тому, незважаючи на наявний прогрес, слід визнати, що універсальний ідеальний регулятор поки що не існує.

Сьогодні спостерігається стрімкий розвиток теорії нейронних мереж. Розроблено велику кількість алгоритмів для функціонування та навчання нейронних мереж, а також проводяться роботи з оптимізації їх структури для підвищення швидкодії, якості результатів та скорочення часу навчання. Нейромереві технології знайшли своє застосування в економіці, медицині, промисловості та багатьох інших сферах науки і техніки, здатні ефективно вирішувати практично будь-які завдання, що пов'язані з моделюванням, прогнозуванням та оптимізацією.

Нейронні мережі пропонують універсальний метод для вирішення багатьох нестандартних завдань, оскільки вони можуть виявляти складні нелінійні зв'язки, удосконалюватися та навчатися під час експлуатації. Системи, створені на основі нейронних мереж, є ефективними, відносно простими у використанні та економічно вигідними. Це пояснює зростаючий інтерес до цієї технології та розширення її сфери застосування, що призвело до розвитку численних підходів, алгоритмів навчання, програмних рішень і варіантів використання.

Основна частина досліджень у сфері нейронних технологій зосереджена на створенні систем управління для виробничих процесів. Проблема полягає в тому, що виробничі процеси зазвичай мають великий набір динамічно взаємодіючих параметрів, що ускладнює створення точних аналітичних моделей для об'єктів управління. Крім того, у сучасних умовах виробництва завдання стають все складнішими, що, по-перше, вимагає значного часу на розробку та перевірку аналітичних моделей, що призводить до затримок у впровадженні нових технологічних рішень, а по-друге, часто створює систему, яка є своєрідним «чорним ящиком», тобто системою, для якої відсутні чіткі аналітичні залежності між вхідними та вихідними параметрами.

Також слід зазначити, що деякі аналітичні математичні моделі, які добре описують процес, виявляються непридатними через високу обчислювальну складність. У цьому контексті все більше уваги приділяється новим альтерна-

тивним підходам до моделювання виробничих процесів, зокрема використанню емпіричних моделей, таких як нейронні мережі, генетичні алгоритми, еволюційні стратегії, еволюційне програмування та нечітка логіка. Використання нейроподібних мереж відкриває величезні можливості для створення моделей, які працюють у реальному часі з мінімальними похибками та мають здатність до навчання під час експлуатації.

Таким чином, використання нейромережових технологій в такій наукоємкій і багатогранній області як системи управління, надає величезні можливості для пошуку ефективних рішень самих різних задач, багато з яких раніше не мали рішення із задовільною точністю. Тому тема роботи, яка присвячена розробці нейромережової системи управління електроприводом механізму повороту стріли роторного екскаватора з урахуванням пружних елементів є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління механізмом обертання колеса роторного екскаватора для підвищення показників якості функціонування системи на основі нейромережових регуляторів.

Досягнення поставленої мети здійснюється рішенням наступних основних задач.

1. Аналіз існуючих систем, а також можливостей і перспектив застосування нейромережових систем.
2. Розвиток математичної моделі динаміки об'єкту управління системи.
3. Розробка структурної схеми системи управління з нейромережовим регулятором.
4. Синтез нейромережової системи із застосуванням нейрорегулятора з прогнозом, яка має високі показники якості функціонування.
5. Моделювання нейромережової системи на ЕОМ і аналіз отриманих результатів.

Об'єктом дослідження є динамічні процеси в системі управління механізмом обертання колеса роторного екскаватора з урахуванням пружних елементів.

Предметом дослідження є моделі і методи синтезу нейромережевої системи управління двомасовою електромеханічною системою механізму обертання колеса роторного екскаватора.

Методи дослідження. Всі теоретичні положення роботи базуються на фундаментальних положеннях теорії автоматичного управління. Методи математичного моделювання вроботи використані для побудови математичних моделей роботи системи. Методи теорії штучних нейронних мереж дозволили синтезувати нейромережеву модель об'єкту управління і вибрати процедуру її навчання. Методи нейроуправління стали основою синтезу нейромережевої системи. Імітаційне моделювання дозволило проаналізувати якість функціонування систем і підтвердило ефективність отриманих результатів.

Наукова новизна одержаних результатів. Наукова новизна результатів, одержаних в процесі виконання роботи, полягає в наступному.

1. Вперше отримано наукове обґрунтування доцільності застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму обертання колеса роторного екскаватора.
2. Вперше запропоновано і синтезовано нейромережеву систему управління механізмом обертання колеса роторного екскаватора з урахуванням пружних елементів, що має високі показники якості функціонування.
3. Вперше запропонована математична модель системи управління механізмом обертання колеса роторного екскаватора з урахуванням пружних елементів у вигляді багатошарової прямонаправленої мережі – багатошарового персептрона.
4. Здійснений подальший розвиток математичних моделей об'єктів управління систем із складними кінематичними зв'язками у вигляді двомасових електромеханічних систем.
5. Одержали подальший розвиток методи синтезу нейромережевих структур з прогнозом для управління складними динамічними об'єктами з математичними моделями у вигляді двомасових електромеханічних систем.

Практичне значення одержаних результатів полягає в розробці нейромережевої системи управління двомасовою електромеханічною системою механізму обертання колеса роторного екскаватора, що забезпечує високоякісне регулювання.

Основні положення магістерської роботи використовуються у навчальному процесі кафедри АМЕТ ННІ «УПА» ХНУ ім. В.Н. Каразіна при читанні лекцій, виконанні курсового проєкту, проведенні практичних занять.

Особистий внесок здобувача. Всі основні результати роботи, які винесені на захист, одержані магістром особисто.

Серед них: синтезована система підлеглого регулювання координат електроприводу; розроблена модель системи з урахуванням пружних властивостей підйомного канату, розроблена структурна схема нейромережевої системи; синтезований нейрорегулятор NN Predictive Controller з використанням пакету прикладних програм Neural Network Toolbox системи MATLAB; виконано моделювання нейромережевої систем і проведено аналіз отриманих результатів; виконано економічне обґрунтування автоматизованої системи управління; розроблено заходи забезпечення безпеки праці при розробці нейромережевої системи управління механізмом обертання колеса роторного екскаватора.

В роботі наведені посилання на авторів і відповідні джерела при використанні відомих залежностей, експериментальних даних і наукових положень.

Апробація результатів роботи. Основні положення та наукові результати були апробовані на науковій конференції студентів ННІ «УПА» ХНУ ім. В.Н. Каразіна (Харків, 2024 р.).

Структура і об'єм роботи Робота складається з вступу, 4 розділів основної частини, висновків, додатків і списку використаних літературних джерел. Повний об'єм роботи складає 190 сторінок, серед них: 41 рисунок за текстом, 20 рисунків на 21 окремих сторінках, 4 таблиці, 5 додатків на 38 сторінках, список літературних джерел з 47 найменування на 5 сторінках.

РОЗДІЛ 1

ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ В СИСТЕМАХ УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Основні поняття і визначення нейронних мереж

1.1.1 Визначення нейронної мережі.

Теорія нейронних мереж охоплює широкий спектр питань з різних галузей науки, таких як біофізика, математика, інформатика, схемотехніка та технології. Визначення поняття "Нейронні мережі" є вельми складним через такий різноманітний спектр аспектів.

Штучні нейронні мережі (НМ) є моделями біологічних нейронних систем. Такі мережі складаються з елементів – штучних нейронів – які пов'язані між собою синаптичними зв'язками. Ці мережі обробляють вхідну інформацію і змінюють свій стан в часі, що відображається у вихідних сигналах.

Робота таких мереж полягає у трансформації вхідних сигналів з плином часу, що веде до зміни внутрішнього стану мережі та формування вихідних дій. Зазвичай нейронні мережі операціонують з цифровими, а не символьними значеннями.

Більшість моделей нейронних мереж потребують навчання. Узагальнюючи, навчання це процес вибору параметрів мережі, за якого мережа найефективніше розв'язує поставлені завдання. Навчання представляє собою задачу багатовимірної оптимізації, для розв'язання якої існує багато алгоритмів. Штучні нейронні мережі представляють собою набір математичних та алгоритмічних методів, які використовуються для вирішення широкого спектру завдань. Основні характеристики цих мереж, які визначають їх як універсальний інструмент для розв'язання завдань, можна виділити наступним чином:

1. Нейромережі надають можливість кращого розуміння організації нервової системи людини і тварин на середніх рівнях, таких як пам'ять, обробка сенсорної інформації та моторика.

2. Нейромережі є ефективним інструментом для обробки інформації, а саме:
 - а) гнучка модель для нелінійної апроксимації багатовимірних функцій; б) засіб прогнозування в часі для процесів, що залежать від багатьох змінних; в) класифікатор за багатьма ознаками, що дозволяє розділити вхідний простір на різні області; г) засіб розпізнавання образів; д) інструмент для пошуку за асоціаціями; е) модель для виявлення закономірностей в масивах даних.
3. Нейронні мережі уникнули обмежень традиційних комп'ютерів завдяки паралельній обробці і сильній взаємодії між нейронами.
4. В перспективі, нейромережі мають потенціал розкрити основні принципи, на яких ґрунтуються вищі функції нервової системи, такі як свідомість, емоції та мислення.

Біофізичні аспекти займають значну частину теорії нейронних мереж. Для створення адекватної математичної моделі необхідно ретельно вивчати роботу біологічних нервових клітин і мереж з точки зору хімії, фізики, теорії інформації і синергетики. Це передбачає вирішення основних питань:

Як працює біологічний нейрон - нервова клітина? Необхідно розробити математичну модель, що відображає інформаційні процеси в нейроні. Які аспекти функціонування нейрона є ключовими для моделювання, а які менш важливі?

Як відбувається передача інформації через з'єднання між нейронами - синапси? Як змінюється провідність синапсу в залежності від сигналів, що проходять через нього?

Якими законами визначається взаємозв'язок між нейронами у мережі? Як клітина "визнає", з якими сусідами встановлювати з'єднання?

Як біологічні нейронні мережі навчаються розв'язувати завдання? Як обираються параметри мережі для генерації вірних вихідних сигналів? Який сигнал вважається правильним, а який помилковим?

1.1.2 Основні характеристики біологічних нейронних мереж.

1. Паралельна обробка інформації. Кожен нейрон генерує свій вихідний сигнал, спираючись на свої вхідні дані та внутрішній стан, з урахуванням загальних механізмів регуляції нервової системи.
2. Способність до комплексної обробки інформації. Нейронні мережі здатні вирішувати всі завдання, відомі людині. До таких властивостей належать асоціативність (здатність відновлювати повний образ за його частинами), класифікація, узагальнення, абстрагування та інші.
3. Самоорганізація. в біологічних нейронних мережах виявляється у здатності системи самостійно вчитися та розв'язувати різноманітні завдання під впливом зовнішнього середовища. Біологічні нейронні мережі не мають принципових обмежень у складності завдань, які вони можуть вирішувати. Нервова система постійно формує та ускладнює свої алгоритми протягом життя. Усе це підносить питання про те, чому системи, які розвиваються, здатні до самоускладнення, виключно властиві живим організмам, зокрема, біологічним нейромережам. Це призводить до запитання, яку ключову умову не врахувала людина у спробах створити системи, що могли б самостійно ускладнюватися.
4. Біологічні нейронні мережі працюють за аналоговим принципом. Інформація надходить у мережу через велику кількість каналів і кодується за просторовим принципом: тип інформації визначається номером нервового волокна, яким вона передається. Амплітуда вхідного сигналу кодується частотою нервових імпульсів, що проходять по цьому волокну.
5. Надійність. Біологічні нейронні мережі мають неймовірну надійність: вихід з ладу навіть 10% нейронів у нервовій системі не порушує її функціонування. У порівнянні з послідовними обчислювальними машинами, побудованими за архітектурою фон-Неймана, де збій одного елемента пам'яті чи вузла апаратури призводить до збою системи.

Сучасні штучні нейронні мережі демонструють цінні властивості, що свідчать про їхню складність і "інтелект":

1. Здатність до навчання. Вибравши відповідну модель нейронної мережі, створивши мережу і виконавши алгоритм навчання, можна натренувати мережу на виконання конкретної задачі. Хоча немає гарантій, що процес буде успішним при певних налаштуваннях мережі, алгоритму та завданні, за правильного підходу навчання зазвичай завершується успішно.
2. Здатність до узагальнення. Після навчання мережа стає стійкою до незначних змін вхідних сигналів (шуму або варіацій вхідних даних) і забезпечує правильний вихідний результат.
3. Здатність до абстрагування. Якщо мережі подати кілька спотворених варіантів одного й того ж вхідного образу, вона може створити на виході ідеальний образ, з яким не мала попереднього знайомства.

1.1.3 Паралельна обробка та здійсненність нейронних мереж.

Швидкодія сучасних комп'ютерних систем значно поступається швидкодії людського мозку, що, зокрема, зумовлено його здатністю до паралельної обробки інформації. Тому для підвищення ефективності обчислювальних систем необхідно перейти від архітектури фон-Неймана до паралельної обробки даних. Проте паралельні комп'ютери ще не набули широкого поширення через кілька ключових причин:

1. Обмеження через складність зв'язків. Кожен процесор в паралельній системі з'єднується з численними іншими, що призводить до значної кількості зв'язків, яка перевищує кількість самих процесорів. Створення таких щільних з'єднань є технічно складним у звичайних інтегральних схемах.
2. Тривимірна структура зв'язків між процесорами. У паралельних системах потрібні тривимірні зв'язки між процесорами, що на даний момент є технологічно складним завданням.
3. Складність програмування. Не розроблені універсальні методи програмування паралельних обчислювальних систем, а також інструменти для написання програм.

Попри перспективність паралельних комп'ютерних систем і нейронних мереж, наразі не існує достатньої елементної бази для їх створення. Тому більшість досліджень у цій галузі проводиться двома основними способами:

1. моделювання нейронних мереж на звичайних послідовних комп'ютерах;
2. створення спеціалізованих нейроплат та нейропроцесорів для прискорення обробки даних з нейронними мережами.

Перший спосіб призводить до значних втрат у швидкодії навіть порівняно з традиційними комп'ютерами, а другий обмежує можливість переходу між різними моделями нейронних мереж, оскільки модель визначається специфічною нейроплатою чи нейропроцесором, і зміна моделі потребує заміни апаратної частини.

Спроби використовувати альтернативні технології, такі як оптичні, хімічні та біологічні системи, для створення нейронних мереж, хоч і обіцяють перспективу, наразі не мають практичного застосування.

1.1.4 Місце нейронних мереж серед інших методів вирішення завдань.

Нейронні мережі демонструють перевагу над традиційними комп'ютерами у виконанні завдань, де людині вдається обійти машини. Завдання, що потребують значних обчислювальних потужностей або високої точності, зазвичай краще виконуються звичайними ЕОМ.

На поточному етапі розвитку нейронні мережі успішно вирішують наступні завдання:

- розпізнавання зорових та слухових образів; це має широке застосування, від розпізнавання тексту і виявлення цілей на радарі до систем голосового управління;
- асоціативний пошук інформації та створення асоціативних моделей, синтез мови, формування природної мови;
- побудова моделей та прогнозування розвитку складних нелінійних систем, що важко описуються математично;

- використання в промисловості, прогнозування розвитку циклонів і інших природних явищ, передбачення змін валютних курсів і інших фінансових процесів;
- системи управління та регулювання з прогнозуванням, управління роботами та іншими складними пристроями;
- різні види автоматів, включаючи системи масового обслуговування, комутаційні та телекомунікаційні системи;
- прийняття рішень та діагностика, де логічний аналіз недостатній; це особливо важливо у сферах, де відсутні чіткі математичні моделі, таких як медицина, криміналістика та фінансова сфера.

Унікальна риса нейронних мереж – їх універсальність. Хоча для більшості зазначених завдань існують ефективні математичні методи і, попри те що нейронні мережі поступаються спеціалізованим підходам для конкретних задач, їх універсальність і потенціал для розв’язання глобальних проблем, таких як створення штучного інтелекту та моделювання процесів мислення, роблять їх важливим напрямом для подальших досліджень.

У додатках А-Г наведені основні поняття теорії нейронних мереж, які являються базою для розуміння виконаних в роботі досліджень.

У даній роботі для синтезу нейромережевої системи використовуються багат шарові прямонаправлені мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямку без зворотних зв'язків всередині мережі – багат шаровий персептрон. В додатку А розглянуто структуру штучного нейрона., а в додатку Б - структуру багат шарових мереж.

Огляд методів навчання і вибір методу навчання нейронних мереж, що використовується в роботі, наведено в додатку В.

Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання нелінійних динамічних систем є отримання їх адекватних математичних моделей. Питання ідентифікації динамічних систем за допомогою нейромереж розглянуто в додатку Г.

1.2 Огляд структур систем нейрорегулювання. Вибір структури системи

На сьогоднішній день в спеціальній літературі по нейрорегулюванню часто згадуються терміни „пряме” і „неаряме” регулювання в різних інтерпретаціях. Тому відразу слід конкретизувати обидва терміни і одноманітно застосовувати їх далі в тексті. Загальноприйнято розрізняти методи регулювання з прямою і непрямою адаптацією залежно від того, яким чином ведеться підстроювання параметрів регулятора. При прямій адаптації зміна параметрів регулятора проводиться безпосередньо, щоб забезпечити виконання певного критерію якості. При непрямій адаптації спочатку здійснюється ідентифікація параметрів моделі, які потім використовуються для синтезу регулятора. Цей же критерій покладений в основу представленого нижче огляду методів нейрорегулювання.

Ще одна комбінація, що часто вживається в літературі, – „online”, „offline” тренування. Online тренування тісно пов'язане з поняттям „регулювання в реальному часі” (*real-time control*). У системах реального часу введення, обробка і виведення інформації відбувається в межах одного періоду дискретності, величина якого визначається в першу чергу характеристиками об'єкту регулювання, а не наявною обчислювальною потужністю. Аналогічно online тренування відповідає випадку, коли тренування мережі (як мінімум одна епоха) і регулювання, що виконується цією ж мережею, проходять одночасно і в межах одного періоду дискретності.

Проведення online тренування в системах регулювання реального часу уточнюється діаграмою на рис. 1.1 [13].

Кількість структур систем нейрорегулювання, запропонованих і випробуваних на сьогоднішній день, дуже велика. Одні системи нейрорегулювання використовують аналітичний закон регулювання і нейромоделі. Нейронна мережа відновлює при цьому елементи матриць моделі об'єкту. Це так звані структуровані методи регулювання. Проте повністю використовувати весь потенціал ап-

роксимації нейронних мереж дозволяють неструктуровані методи регулювання. У наступних нижче підрозділах будуть стисло розглянуті структури систем нейрорегулювання, актуальні на сьогоднішній день.

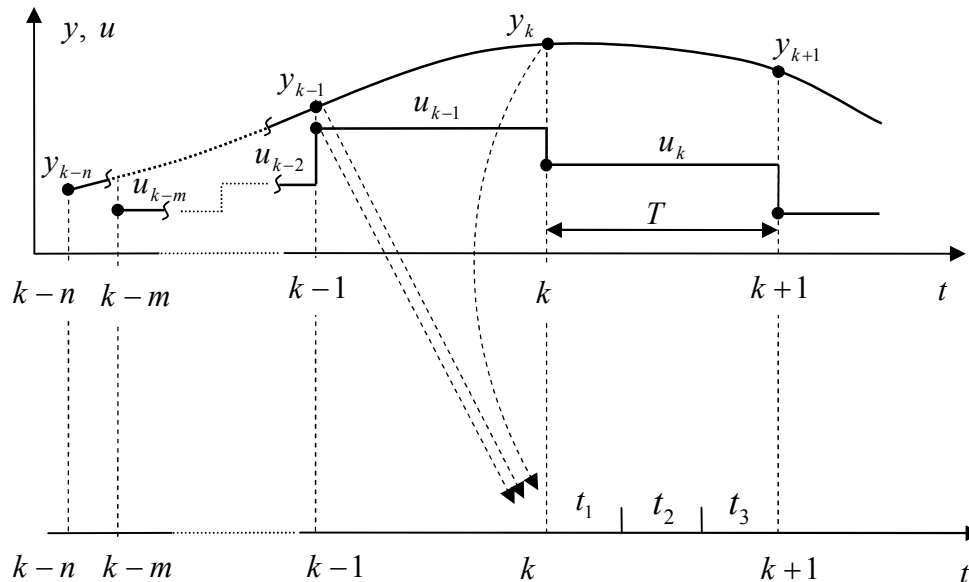


Рисунок. 1.1 – Регулювання і тренування мережі в реальному часі

На рисунку прийняті наступні позначення: y – регульована координата; u – дія на об'єкт; T – період дискретності; t_1 – попередня обробка даних датчиків; t_2 – тренування мережі; t_3 – обчислення і видача дії на об'єкт; m, n – число попередніх значень змінних.

1.2.1 Пряме інверсне регулювання.

Термін «*Direct inverse control*» був введений Вербосом [14]. Пряме інверсне нейрорегулювання – це метод, найчастіше вживаний на сьогоднішній день. Тренування мережі при регулюванні цим методом може проводитися як offline, так і online. Згідно класифікації Псалтиса і ін. [15], ці методи названі «*General learning*» і «*specialized learning*». Відмінності структур прямого інверсного регулювання з offline і online тренуванням розглядаються нижчим.

Offline тренування. Пряме інверсне регулювання з offline тренуванням засноване на пропозиції, що інверсне відображення простору виходів в простір

входів існує. При цьому спочатку в режимі offline формують модель інверсної динаміки об'єкту регулювання **NC** (перемикачі на рис. 1.2 в нижньому положенні). Потім отримана інверсна модель включається послідовно з об'єктом (перемикачі на рис. 1.2 у верхньому положенні) і використовується як регулятор. Назва методу "Пряме інверсне регулювання" згідно введеного визначення тут повністю підтверджується.

Очевидний недолік даної структури регулювання – недостатня робастність через відсутність явного зворотного зв'язку. Тому умова реалізації цієї структури регулювання – стійкість об'єкту регулювання. На практиці ця умова не завжди виконується.

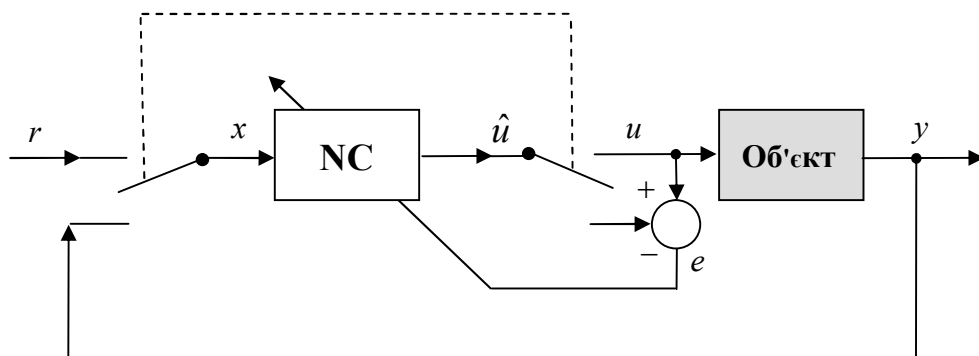


Рисунок 1.2 – Пряме інверсне регулювання з offline тренуванням

Online тренування. Структура системи прямого інверсного регулювання з тренуванням мережі в режимі online представлена на Рис. 1.3.

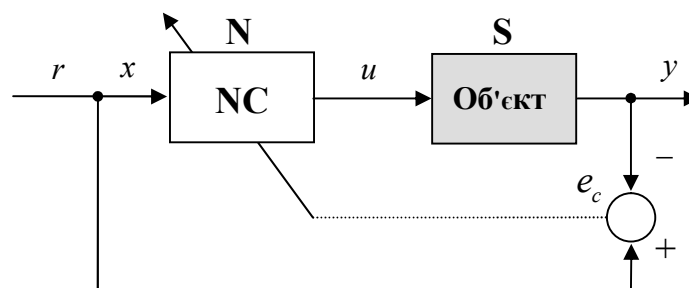


Рисунок 1.3 – Пряме інверсне регулювання з online тренуванням

Кінцева мета тренування нейрорегулятора **NC** вважається досягнутою, якщо він генерує такий сигнал дії на об'єкт, який приводить до відстежування

регульованою координатою y сигналу завдання r . Регулятор навчається безперервно і одночасно з процесом регулювання, тому цей метод застосовний для змінних в часі об'єктів регулювання.

Істотний недолік тренування мережі в режимі online – необхідність коректування сигналу помилки регулювання e_c оскільки e_c формується на виході об'єкту регулювання, а не на виході регулятора. Пунктирна лінія на рис. 1.2 відображає виникаючий пропуск. Формули обчислення градієнта приведені в таблиці 1.1.

Таблиця 1.1 – Обчислення градієнта при регулюванні з online тренуванням

Вихідний сигнал мережі	$u = \mathbf{N}(x, \mathbf{w})$
Регульована координата	$y = \mathbf{S}(u)$
Помилка	$e_c = r - y$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e_c} \frac{\partial e_c}{\partial \mathbf{w}} = -e_c \frac{\partial y}{\partial \mathbf{w}} = -e_c \frac{\partial y}{\partial u} \frac{\partial u}{\partial \mathbf{w}}$

Похідна $\frac{\partial u}{\partial \mathbf{w}}$ може бути завжди обчислена. Складову $\frac{\partial y}{\partial u}$ називають матрицею Якобі. Для відомих об'єктів регулювання матриця Якобі може бути отримана аналітично, а її елементи – розраховані. На жаль, об'єкт регулювання найчастіше точно невідомий. Приблизно похідна може бути обчислена за допомогою різницевого рівняння [15]:

$$\frac{\partial y}{\partial u} \approx \frac{\Delta y}{\Delta u} = \frac{y(k) - y(k-1)}{u(k) - u(k-1)}.$$

По аналогії з лінійними методами адаптивного регулювання запропоновано також використовувати замість величини похідної її знак [16], якщо про об'єкт відома якісна інформація. Проте при зашумлених даних обидва методи не здатні ефективно вирішити вказану проблему.

Слід зазначити, що регулювання в перший момент часу нетренованим регулятором в розімкненій системі може привести до неконтрольованої поведінки

системи. Щоб подолати цей недолік, слід проводити грубе попереднє тренування нейрорегулятора.

1.2.2 Непряме інверсне нейрорегулювання.

У літературі зустрічаються різні назви цього методу регулювання, наприклад, *feedforward inverse control* у Наренда [17] або *specialized inverse learning* у Ханта [18]. Структурна схема системи регулювання представлена на рис. 1.4.

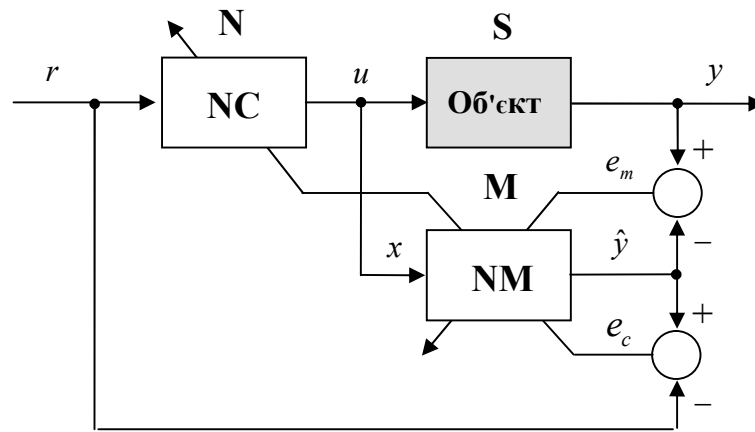


Рисунок 1.4– Непряме інверсне нейрорегулювання

Непряме нейрорегулювання – це двохетапний процес. На першому етапі відбувається тренування нейромоделі об'єкту **NM** у режимі *offline*.. Після успішного тренування на основі нейромоделі у момент часу τ правилом підстановки можна обчислити матрицю Якобі:

$$\left. \frac{\partial y}{\partial u} \right|_{t=\tau} \approx \left. \frac{\partial \hat{y}}{\partial u} \right|_{x(k)} = \frac{\partial y^{(n)}}{\partial n^{(n)}} \frac{\partial n^{(n)}}{\partial a^{(n-1)}} \frac{\partial a^{(n-1)}}{\partial n^{(n-1)}} \cdots \frac{\partial y^{(1)}}{\partial n^{(1)}} \frac{\partial n^{(1)}}{\partial x_0},$$

де $n^{(i)}, a^{(i)}$ – зважена сума і вихід i -го шаруючі мережі $1 \leq i \leq L$;

$a^{(L)} = \hat{y}$ – вихід нейромоделі;

$x_0 = u$ – один з входів нейромоделі.

Завдяки цьому помилка e_c може бути використана для підстроювання вагових коефіцієнтів нейрорегулятора **NC** у другій фазі – фазі регулювання.

Проблематичною тут виявляється ситуація, коли нейрорегулятор генерує дію на об'єкт поза діапазоном, в якому тренувалася нейромодель об'єкту. У роботах Наренди і ін. [17] досліджувався варіант непрямого нейрорегулювання, в якому нейромодель **NM** підстроювалася в режим online. Ця концепція системи регулювання має проблеми забезпечення стійкості в перші моменти часу тренування нейрорегулятора. Явно виражений зворотний зв'язок в системі також відсутній. Ішигуро [19] запропонували ефективнішу структуру „*neural compensator*”. У ній нейронна мережа формує сигнал компенсації паралельно класичному регулятору, а сигнал помилки для тренування мережі формується побічно як різниця між сигналом дії на об'єкт і вихідним сигналом аналітичної моделі об'єкту.

1.2.3 Регулювання з внутрішньою моделлю.

У системі регулювання з внутрішньою моделлю (*Internal model Control*, ІМС) використовується як пряма, так і інверсна модель об'єкту. Схема вперше була запропонована Гарсія і Морарі [20] з аналітичними нелінійними моделями. Застосування нейромоделей може служити яскравим прикладом того, як нейротехнології можуть бути інтегровані в класичні адаптивні системи регулювання. Структурна схема системи представлена на рис. 1.5.

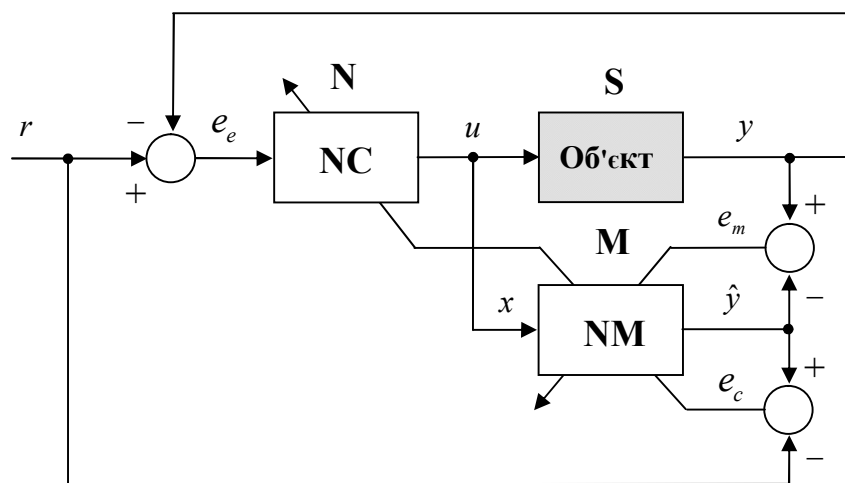


Рисунок 1.5 – Система регулювання з внутрішньою моделлю

Аналогічно непрямому нейрорегулюванню, спочатку здійснюється тренування нейромоделі **NM** у режимі offline. З допомогою **NM** формується матриця Якобі для корекції сигналу помилки e_n щоб забезпечити тренування інверсного нейрорегулятора **НС** у режимі online. На відміну від непрямого регулювання обидві нейронні моделі тут є складовими частинами контура регулювання, оскільки сигнал помилки між моделлю об'єкту і самим об'єктом e_m замикається на вхід нейрорегулятора.

Дана структура має одну цікаву особливість. Якщо модель об'єкту ідеальна ($\mathbf{M} = \mathbf{S}$), то ІМС стає ідентичною непрямому нейрорегулюванню з розімкненим контуром ($e_m = 0$), завдяки чому стає можливим доказ стійкості. Примітно також, що при асимптотично стійких збуреннях і сигналі завдання відсутня статична помилка регулювання ($y = r$) за умови, що нейрорегулятор є інверсною ланкою моделі об'єкту ($\mathbf{N} = \mathbf{M}^{-1}$). Якість моделі об'єкту при цьому – другорядна. Це витікає з наступних виразів:

$$\hat{y} = r_e \cdot \mathbf{N} \cdot \mathbf{M}; \quad r_e = r - y + \hat{y}.$$

Метод ІМС застосовний, проте, тільки до стійких об'єктів регулювання. Щоб підтримувати стійкість контура регулювання зазвичай потрібний фільтр низьких частот на вході нейрорегулятора, оскільки в реальних умовах ідеальні моделі практично не реалізуються. В принципі, дана концепція регулювання має ті ж слабкі місця, що і непряме інверсне нейрорегулювання. В порівнянні з класичними ІМС нейронні мережі нічого не додають окрім спрощення потрібній ідентифікації нелінійних динамічних систем.

1.2.4 Feedforward-Feedback-Error регулювання.

Дана концепція регулювання, в принципі, теж відображає структуру прямого інверсного регулювання. Як видно з рис. 1.6, в систему додатково інтегровані ще один зворотний зв'язок і третя нейронна мережа – *Feedback-Error-регулятор*.

Обидва нейрорегулятора NC_{FF} і NC_{FB} тренуються спочатку в режимі offline роздільно і з різними функціоналами якості. Модель об'єкту NM повинна бути також натренована в режимі offline. Після того, як модель об'єкту NM отримана, процес регулювання може бути початий, а помилка e_c скоректована за допомогою прямої нейромоделі для отримання сигналу підстроювання нейрорегуляторів. NC_{FF} виконує роль нейронного компенсатора в прямому каналі, як і при непрямому інверсному регулюванні. За рахунок Feedback-Error-регулятора NC_{FB} утворюється замкнутий контур, що позитивно впливає на якість регулювання. Даний метод регулювання застосовано, наприклад в [21, 22].

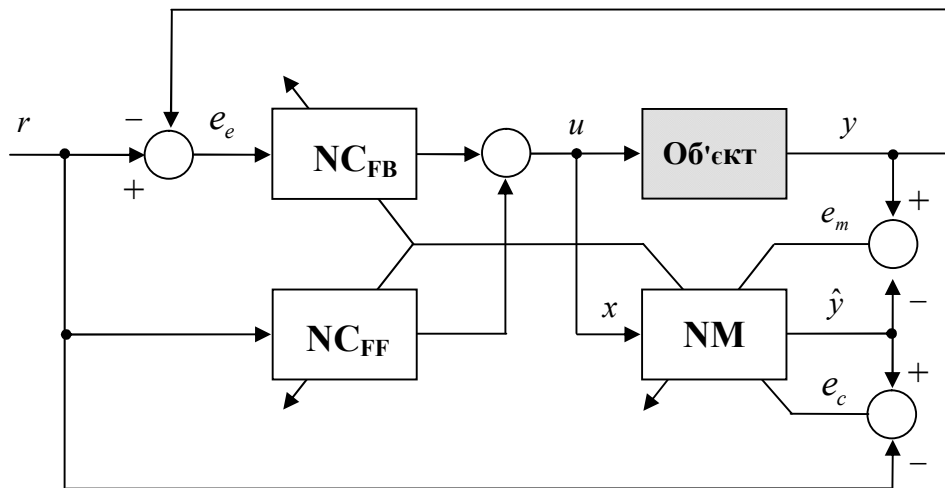


Рисунок 1.6 – Feedforward-Feedback-Error-регулятор

1.2.5 Регулювання з Feedback-Error-Learning

На рис. 1.7 представлена структура системи регулювання з «*Feedback-Error-Learning*» (FEL) Кавато і ін. [23]. По суті, дана структура аналогічна прямому інверсному регулюванню з класичним, наприклад, ПД-регулятором.

Проте, на відміну від всіх раніше згаданих прямих методів нейрорегулювання, метод FEL використовує вихідний сигнал ПД-регулятора як сигнал помилки, що вимагає мінімізації e_c для підстроювання вагових коефіцієнтів нейрорегулятора. Завдяки цьому відпадає необхідність коректувати помилку за допомогою моделі об'єкту. Іншими словами, тут використовується інший фун-

кціонал якості з метою відмови від обчислення матриці Якобі. В результаті online тренування нейрорегулятор **NC** все більше і більше замінює ПД-регулятор. У результаті $e_c = 0$, що означає, що об'єкт регулюється тільки за допомогою нейрорегулятора, що є інверсне об'єкту регулювання ланка. Випадкові відхилення регульованої координати можуть при цьому коректуватися за допомогою ПД-регулятора. Крім того, ПД-регулятор забезпечує стійкість системи в початковій фазі тренування мережі.

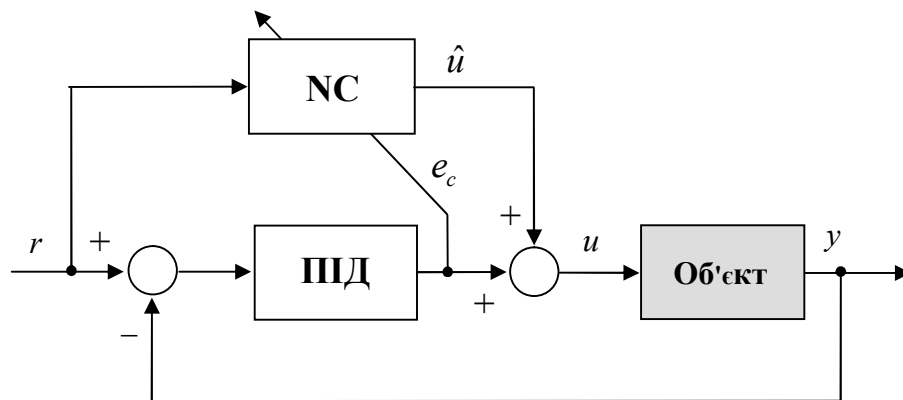


Рисунок 1.7 – Регулювання з Feedback-Error-Learning

До недоліків FEL регулювання відноситься сильна залежність швидкості збіжності від параметрів ПД-регулятора [24]. Слід зазначити, що твердження про те, що нейрорегулятор є інверсною об'єкту ланкою, не буде справедливим, якщо тренувальні дані не є глобальними. Вагові коефіцієнти слід повторно підстроювати, якщо завдання лежатиме в іншому робочому діапазоні.

1.2.6 Регулювання методом компенсації завдання.

Регулювання методом компенсації завдання (*Reference Compensation Technique*, RCT), структурна схема якого представлена на рис. 1.8, була запропонована Юнгом і Хсіа [25].

Як і при регулюванні FEL тут також не проводиться обчислення матриці Якобі завдяки тому, що для тренування нейросети використовується спеціально вибраний сигнал помилки e_c (див. таблицю 1.2).

Особливість методу RCT в тому, що компенсація наявних нелінійностей здійснюється на рівні сигналів завдання, а не сигналів дії на об'єкт. Перевага даної техніки – простота вбудовування в готовий контур регулювання. Складність методу полягає у виборі початкових значень параметрів мережі, оскільки інакше можливе формування небезпечних сигналів завдання.

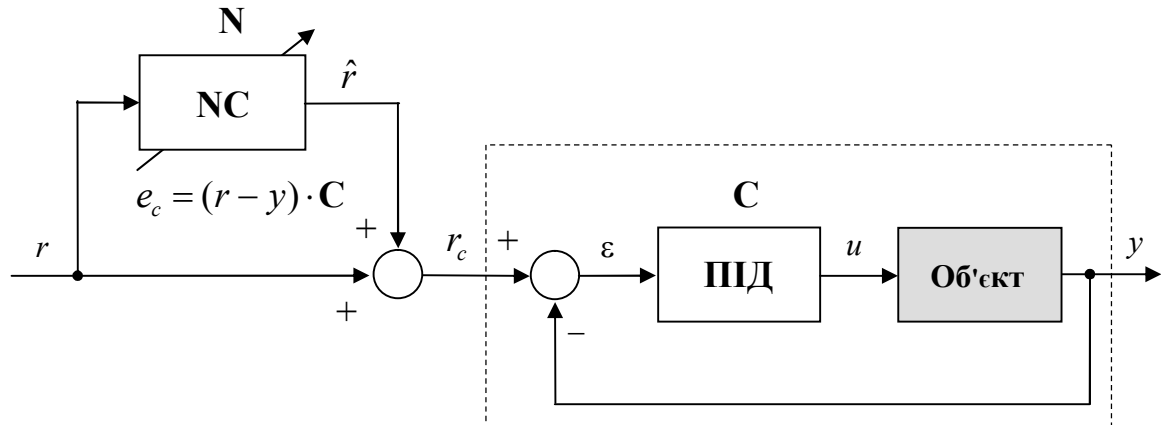


Рисунок 1.8 – Регулювання методом компенсації завдання

Таблиця 1.2 – Обчислення градієнта при FEL регулюванні

Вихідний сигнал мережі	$\hat{r} = \mathbf{N}(r, \mathbf{w})$
Помилка	$e_c = (r - y) \cdot \mathbf{C} = u - \hat{r} \cdot \mathbf{C}$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e_c} \frac{\partial e_c}{\partial \mathbf{w}} = -e_c \frac{\partial \hat{r}}{\partial \mathbf{w}} \mathbf{C}$

1.2.7 Instantaneous linearization.

Головна ідея цього методу полягає в лінеаризації нейронної моделі об'єкту в поточній робочій точці [26].

Хай, наприклад, нелінійний динамічний об'єкт

$$y(k) = \phi(y(k-1) \dots y(k-n), u(k) \dots u(k-m))$$

моделюється за допомогою NARX-нейромоделі

$$\hat{y}(k) = \mathbf{M}(\mathbf{x}(k), \mathbf{w}),$$

де $\mathbf{x}(k) = [y(k-1) \dots y(k-n_y), u(k-n_k) \dots u(k-n_u-n_k+1)]^T$ – вектор стану;
 $\mathbf{w}$ – вагові коефіцієнти мережі.

Цей об'єкт може бути лінеаризован в момент $k = \kappa$ наступним чином:

$$\hat{y}(k)|_{k=\kappa} = y(k-1) - a_1 \Delta y(k-1) \dots - a_{n_y} \Delta y(k-n_y) + \\ + b_0 \Delta u(k-n_k) + b_{n_u} \Delta u(k-n_u-n_k+1),$$

де

$$a_i = - \left. \frac{\partial \mathbf{M}(\mathbf{x}(k), \mathbf{w})}{\partial y(k-i)} \right|_{k=\kappa}, \quad b_j = - \left. \frac{\partial \mathbf{M}(\mathbf{x}(k), \mathbf{w})}{\partial u(k-j)} \right|_{k=\kappa}, \quad i = 1 \dots n_y, \quad j = 0 \dots n_u + n_k - 1,$$

$$\Delta y(k-i) = y(k-i) - y(k-i-1), \quad \Delta u(k-j) = u(k-j) - u(k-j-1).$$

Приватні похідні можуть бути обчислені з нейромоделі **NM** за допомогою вже відомого правила підстановки. Звідси один крок до синтезу лінійного регулятора. Концепція регулювання представлена на рис. 1.9 і аналогічна лінійним методам регулювання з непрямою адаптацією.

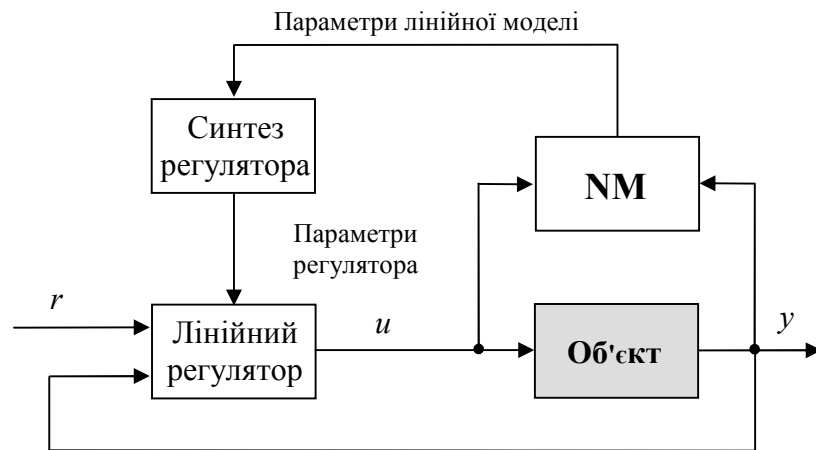


Рисунок 1.9 – Instantaneous linearization і синтез лінійного регулятора

1.2.8 Нелінійне предиктивное регулювання.

Принцип предиктивного методу регулювання на основі моделі (*model predictive control*, MPC) полягає у формуванні такої послідовності сигналів дії на об'єкт, яка мінімізує різницю між завданням і що передбачається моделлю

процесу вихідним сигналом в майбутньому (на «горизонті»), див. пунктирні лінії на рис. 1.10.

Регулятор в даному випадку є не що інше, як ітераційний алгоритм оптимізації [27]:

- мінімізація здійснюється чисельними методами оптимізації;
- мінімізується функціонал якості, що описує бажаний критерій якості;
- для дії на об'єкт використовується тільки перший елемент оптимізованої послідовності;
- вся процедура оптимізації повторюється на кожному кроці дискретності.

Перевага методу MPC – висока якість регулювання навіть у разі складних нелінійностей в об'єкті. Недоліком методу є дуже трудомісткі обчислення і високі вимоги до якості нелінійної моделі об'єкту. У даній роботі використовуємо нелінійне предиктивне регулювання.

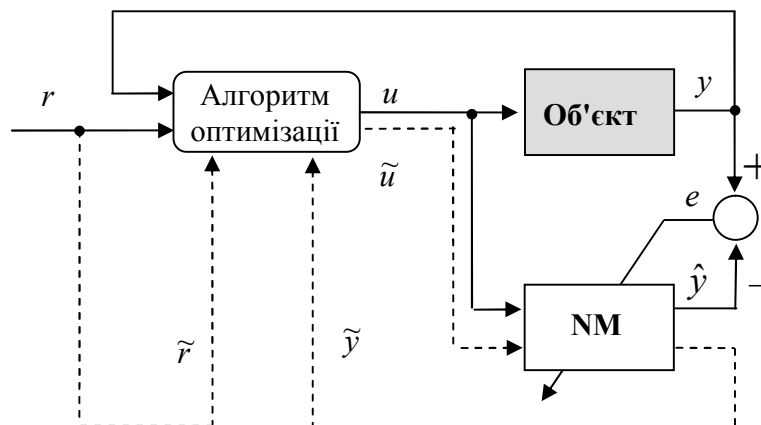


Рисунок 1.10 – Предиктивне регулювання

1.3 Загальні відомості про роторні екскаватори

Головним елементом гірничодобувних комплексів є роторний екскаватор, що являє собою складну систему, яка включає понад десяти взаємодіючих силових механізмів та містить як основне, так і допоміжне обладнання для транс-

портування видобутої породи до інших машин комплексу. На рисунку 1.13 зображена схема компоновання роторного екскаватора.

Конструкція екскаватора складається з поворотної і неповоротної частин, розділених опорно-поворотним пристроєм 1, яке забезпечує поворот верхньої частини машини щодо вертикальної осі. До складу поворотної частини входить платформа, на якій закріплена стріла ротора 4 з робочим роторним колесом 3 і встановленими по його периферії ковшами 2. При обертанні роторного колеса з урізуванням ріжучих крамок ковшів і поворотному русі стріли відбувається відділення природи, що поступає на розташований усередині фермової конструкції стріли конвейєр 16.

Потім порода подається в центральну частину екскаватора, де встановлений додрабливаючий пристрій, перетинає площину опорно-поворотного пристрою 1, поступає на конвейєр нижньої частини машини, потім на конвейєр, розташований усередині розвантажувальної консолі 10.

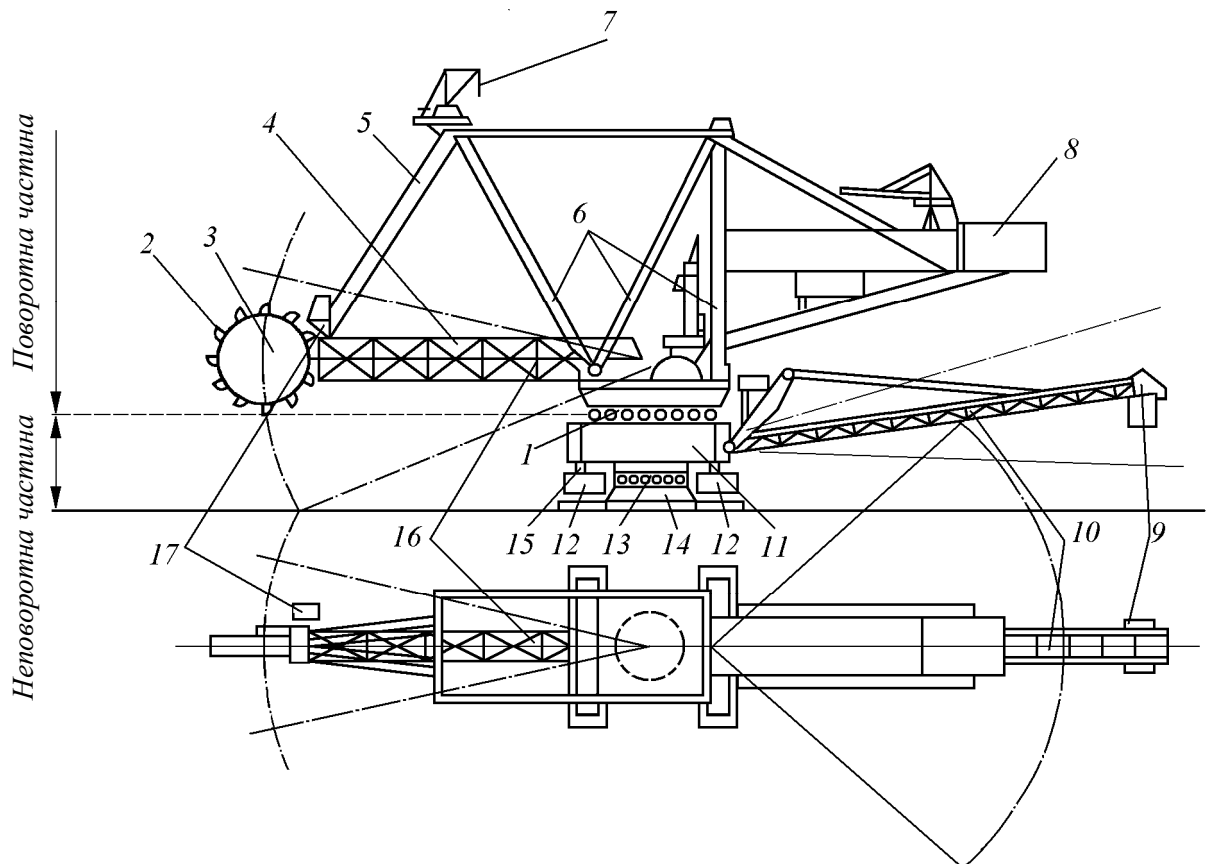


Рисунок 1.13 – Компоновальна схема роторного екскаватора

Для зміни положення роторного колеса у вертикальній площині стріла екскаватора може встановлюватися в робочому положенні в межах деякого сектора канатною підвіскою 5 і повертатися навколо осі, розташованої поворотної частини екскаватора. Верхнє положення стріли визначає найбільшу висоту оброблюваного уступу, нижнє- максимальну глибину копання нижче рівня стояння машини. Верхні блоки підвіски стріли закріплені на одному з порталів 6. За допомогою підкошувань, винесених убік, протилежну роторній стрілі, і встановлених на консольній частині будов 8, створюється противага, компенсуюча перекидаючий момент стріли.

Нижче за опорно-поворотний пристрій розташований могутній силовий пояс 11, сполучений внизу з опорною базою екскаватора 14. До нерухомої частини з обох боків кріпляться могутні гідродомкрати 15, здатні встановлювати конструкцію з опорной на ґрунт через базу або через шагающе-рейкове устаткування (лижі) Розвантажувальна консоль 10 разом з конвейєром може встановлюватися під різними кутами (в межах 80°) по відношенню до напрямку переміщень екскаватора, що забезпечує автономність її просторового положення в процесі розвантаження. Так само, як і стріла ротора, розвантажувальна консоль має можливість різної установки у вертикальній площині.

Управління робочими процесами екскаватора здійснюється з кабін управління, розташованих у роторного колеса 17, на кінці розвантажувальної консолі 10 і на нижній рамі.

Для розворотів екскаватора при зміні напрямку перестановки нерухома його частина разом з поворотною можуть повертатися щодо бази, для чого є ще один поворотний пристрій 13.

Зручність обслуговування елементів конструкції і устаткування екскаватора забезпечується установкою в трьох місцях 7- на поворотній платформі, верхній будові(надбудові) і похилому порталі- підйомних кранів вантажопідйомністю від 5 до 20 т.

Основними механізмами, що забезпечують робочий процес екскаватора, є наступні: механізм приводу ротора; механізм повороту стріли екскаватора; ме-

ханізм розвороту консолі; механізм розвороту екскаватора; механізм підйому; механізм ходу; механізм конвейєра стріли; механізм конвейєра розвантажувальної консолі.

Компоновка індивідуальних електроприводів роторного екскаватора показана на рис. 1.14.

Безпосередньо у роторного колеса встановлюються електродвигуни і редуктори приводу обертання 1, а також електропривод барабанного живильника 2, що направляє потік породи, що відділяється від ковшів, на конвеєр стріли ротора. Електропривод конвеєра 3 розташовується в зоні осі гойдання стріли.

На поворотній частині опорно-поворотного механізму розташовані електроприводи 6, що забезпечують взаємний поворот нижньої і верхньої частин машини. Привід підйому стріли ротора 5 встановлений в надбудові протипаги, в нижній частині якого розміщено вентиляційне устаткування 4.

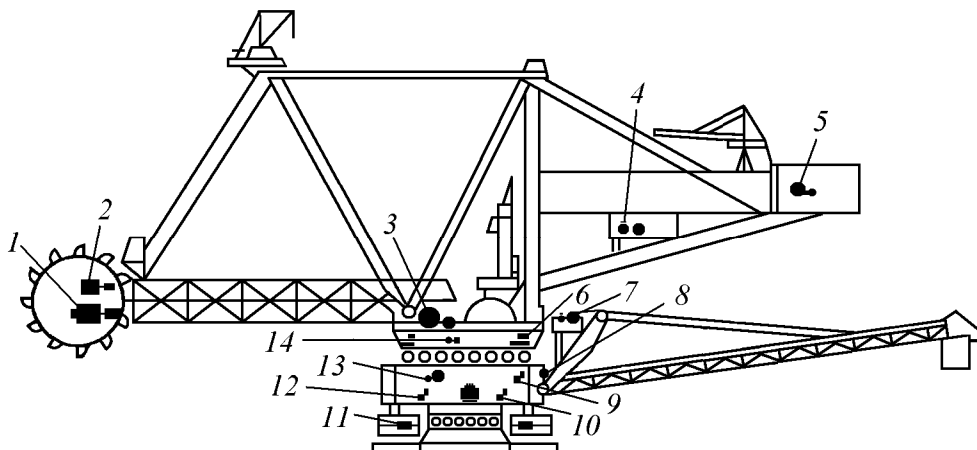


Рисунок 1.14 - Компоновка індивідуальних електроприводів роторного екскаватора

У верхньому поясі в зоні установки воронки, що обертаються, або додблюючого пристрою розташовується і їх приводи 14.

Переміщення розвантажувальної консолі у вертикальній площині забезпечується електроприводами підйому 7, встановленими на спеціальному майданчику, нижче за горизонтальну площину коштує привід повороту консолі 9,

аналогічний приводу повороту верхньої будови, але меншій потужності, і привід конвеєра розвантажувальної консолі 8.

У неповоротній частині машини розташовуються електроприводи механізму розвороту екскаватора 12, конвеєра нижньої рами 13, а також електроустаткування введення електроенергії: струмоприймачі, вимикачі, трансформатори, розподільні пристрої високої і низької напруги 10. Безпосередньо у металоконструкціях лиж рейковий-крокуючого ходу коштують електроприводи ходових лебідок 11.

1.4 Обґрунтування вибору системи електроприводу роторного колеса екскаватора

Роторне колесо виконує головний робочий рух, що забезпечує різання ґрунту, і його електропривод повинен працювати в тривалому режимі. У зв'язку відсутністю технологічної необхідності зміни напрямку обертання роторного колеса електропривод може бути нерeverсивним.

Для забезпечення раціональних режимів різання різних по міцності ґрунтів і ефективного використання конструктивних можливостей екскаватора електропривод повинен забезпечувати зниження кутової швидкості роторного колеса на 30 – 60% по відношенню до номінальної. Отже, необхідний діапазон регулювання D , який визначається як відношення середніх значень максимальної швидкості до мінімальної [28]

$$D = \omega_{\text{ср max}} / \omega_{\text{ср min}} \quad (1.15)$$

лежать в межах $(1,4 - 1,7) : 1$.

Для усунення можливих резонансних коливань металоконструкцій, що виникають при несприятливих поєднаннях швидкостей роторного колеса, пружних властивостей системи ротор – стріла – підвіска стріли, фортеці породи і т.п., бажаним є безступінчате регулювання швидкості. Характер перехідних процесів повинен бути оптимальним, таким, що забезпечує обмеження динамічних навантажень механізму і металоконструкцій.

Особливістю механізму є необхідність регулювання швидкості привода практично тільки в процесі різання, тобто під навантаженням. При істотних обмеженнях на динамічні навантаження, а значить, і на прискорення електропривод, як при розгоні, так і при гальмуванні працюватиме тільки в руховому режимі. Отже, забезпечення електроприводом рекуперативного гальмування не є обов'язковим.

Механічна характеристика електроприводу, що представляє залежність кутової швидкості електродвигунів від електромагнітного моменту $\omega = f(M)$, повинна мати екскаваторну форму. Для електроприводу роторного колеса потрібна, можливо, вища жорсткість робочої ділянки механічної характеристики. При недостатній жорсткості коливання зусилля різання викличуть відповідні коливання швидкості електродвигунів і роторного колеса, які будуть тим більше, чим м'якше характеристика. Коливання швидкості можуть додатково посилюватися унаслідок обернено пропорційної залежності між лінійною швидкістю різання і площею поперечного перетину стружок, що зрізаються, а, отже, і моментом навантаження електроприводу. Це може привести до небажаних коливань роторної стріли з небезпекою виникнення резонансних явищ в металоконструкціях екскаватора.

Можливість технологічних перевантажень і режимів стопоріння при зустрічі ковшів з непереборною перешкодою зумовила вимогу обмеження моменту, що розвивався двигунами, на допустимому рівні. Ділянка обмеження моменту механічної характеристики зазвичай має коефіцієнт відсічення, рівний:

$$k_{\text{відс}} = M_{\text{відс}} / M_{\text{стоп}} = 0,8, \quad (1.16)$$

де $M_{\text{відс}}$ - момент відсічення, перевищення якого супроводжується переходом з робочої ділянки механічної характеристики на ділянку обмеження моменту;

$M_{\text{стоп}}$ - момент стопоріння, відповідає максимально допустимому за умовами міцності механізму моменту.

Таке значення $k_{\text{відс}}$ дає можливість машиністові екскаватора, електроприводи якого не мають автоматичного регулювання швидкості повороту стріли

ротора залежно від навантаження приводу роторного колеса, за швидкістю приводу оцінювати його навантаження і своєчасно управляти електроприводами роторного колеса і повороту стріли ротора, не допускаючи їх перевантаження і стопоріння.

Використання екскаваторної механічної характеристики у вживаних електроприводах роторного колеса не виключає в режимах стопоріння неприпустимих динамічних навантажень механічного устаткування, тому в кінематичний ланцюг механізму вводиться ланка з обмеженим граничним передаваним моментом.

Електродвигуни приводу роторного колеса встановлені на оголовці стріли і працюють в умовах підвищеної вібрації і трясіння, в зоні інтенсивної освіти пороши, при значних диферентах, пов'язаних з різним положенням стріли у вертикальній площині. У зв'язку з цим електродвигуни разом з передавальним і виконавчим механізмами повинні мати мінімальну масу і габарити. Крім того, електродвигуни повинні бути закритого виконання, допускати роботу при вібраціях і в похилому положенні, а також оснащуватися пристроями очищення від пилу повітря, використовуваного для охолодження електродвигунів.

Складність вимог, що пред'являються до регульованих електроприводів основних механізмів роторних екскаваторів, зумовила для більшості з них застосування системи електроприводу постійного струму керований перетворювач – двигун. Як керований перетворювач використовується або електромашинний перетворювач (система Г-Д), або реверсивний теристорний перетворювач (система ТП-Д). У магістерській роботі для електроприводу роторного колеса вибираємо систему генератор – двигун.

1.5 Нейромережевий підхід до управління. Постановка задачі дослідження

Однією з основних проблем у алгоритмічному та структурному синтезі систем управління на базі нейромереж є розумний вибір необхідної інформації

для тренування багатошарових нейронних мереж, які формують оптимальні закони управління та встановлюють умови для досягнення цілей управління для класу нелінійних об'єктів. Алгоритмічний синтез охоплює вибір оптимального алгоритму для навчання нейронної мережі в реальному часі та аналіз можливості досягнення поставлених цілей управління, що має важливе практичне значення.

Для ефективного використання нейромережевого підходу необхідно, передусім, розробити математичну модель системи. Однак, математичне моделювання, яке базується на припущеннях про лінійність системи, не завжди відображає її реальні фізичні характеристики. Навіть якщо вдається створити складні математичні моделі, які точно відображають фізичні взаємозв'язки між входом і виходом системи, вони можуть виявитися непридатними для управлінських цілей. Для нелінійних систем особливо важко забезпечити низьку чутливість моделі до параметрів, що є критично важливим.

Застосування статистичних моделей дозволяє оцінювати параметри процесу через аналіз частотних характеристик за допомогою методів класичної теорії управління. Проте ефективність цього підходу може бути недостатньою для досягнення необхідної точності в управлінні. Основною причиною цього є спроба адаптувати лінійні моделі до складних нелінійних систем. Окрім того, ці методи базуються на моделях «чорного ящика», де параметри часто не мають фізичного значення. Через це їх застосування в управлінських практиках є обмеженим.

Використання адаптивних методів управління припускає наявність математичної моделі, заснованої на фізичних явищах, і методів оцінки параметрів цієї моделі. Потім будується закон управління, направлений на досягнення деякої мети, а математична модель, розглядається як реальна система. Даний підхід також ґрунтується на теорії лінійних систем. При виникненні змін в об'єкті управління або в зовнішніх умовах необхідно перебудовувати параметри моделі і задавати новий закон управління. Це вимагає втручання оператора для перевірки адекватності моделі і реальної системи.

Отже, можна зробити наступний висновок: для ефективного застосування алгоритмів управління вони повинні бути адаптивними, стійкими, нелінійними, а також легкими у реалізації та розумінні. Саме тому штучні нейронні мережі (ШНМ), які володіють усіма зазначеними властивостями, набули широкого застосування в управлінських задачах.

Управління на основі нейронних мереж базується на використанні повністю визначених ШНМ для генерації необхідних управлінських сигналів. Високий інтерес до цього напрямку в теорії управління обумовлений наступними факторами:

- ШНМ є адаптивними системами, які здатні навчатися розв'язувати складні завдання. Для ефективного навчання необхідно правильно обрати структуру мережі та забезпечити достатній об'єм навчальної інформації.
- Нейромережеві технології управління допомагають подолати багато труднощів, які виникають при роботі з нелінійними об'єктами або об'єктами невідомої структури, які не можна розв'язати за допомогою традиційних методів адаптивного управління. Здатність ШНМ реалізовувати складні нелінійні відображення забезпечується використанням сигмоїдальних функцій активації (або інших нелінійних функцій загального вигляду) у нейронах прихованих шарів.
- Здібність ШНМ до самонавчання дозволяє застосовувати нейрорегулятори навіть за значних невизначеностей, тоді як для традиційних методів адаптивного управління необхідно мати великий обсяг апріорної інформації про об'єкт управління.
- Висока швидкодія та надійність нейрорегуляторів забезпечуються високим ступенем паралельності ШНМ. Простота реалізації нейронних мереж та їх здатність до навчання роблять їх особливо привабливими для управління складними нелінійними об'єктами в реальному часі.

Проведений аналіз показав доцільність використання нейромережевих технологій для синтезу нейромережевої системи управління механізмом обертання колеса роторного екскаватора. Це зумовлено в першу чергу тим фактором, що при несприятливих поєднаннях швидкостей роторного колеса, пружних властивостей системи ротор-стріла-підвіска стріли, міцності породи і т.п.

можуть виникати резонансні коливання металоконструкцій, що несприятливо впливає на роботу механізму. Характер перехідних процесів повинен бути оптимальним, таким, що забезпечує обмеження динамічних навантажень механізму і металоконструкцій. Традиційні регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

Метою роботи є розробка нейромережевої системи управління механізмом обертання колеса роторного екскаватора з урахуванням пружних властивостей системи ротор-стріла-підвіска стріли.

Для досягнення поставленої мети необхідно вирішити наступні завдання.

1. Виконати вибір системи електроприводу механізму обертання колеса роторного екскаватора, вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу. Виконати синтез системи управління електроприводом і провести моделювання системи на ЕОМ.

2. Розробити математичну моделі системи управління з урахуванням пружних властивостей системи ротор-стріла-підвіска стріли, провести моделювання багатомасової системи на ЕОМ і виконати аналіз отриманих результатів.

3. Розробити структурну схему нейромережевої системи управління, що реалізує високоякісне регулювання з врахуванням кінцевої жорсткості елементів механічної частини електроприводу.

4. Синтезувати за допомогою багатошарової прямонаправленої мережі модель багатомасової системи з високою точністю ідентифікації.

5. Використовуючи принцип предикативного методу регулювання на основі нейромережевої моделі об'єкта синтезувати нейромережевий регулятор, що забезпечує високу швидкодію і якість управління системи.

6. Виконати моделювання нейромережевої систем і провести аналіз отриманих результатів.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Оцінено фактори, що визначають ефективність застосування нейромережових технологій у управлінських задачах. Широке використання нейронних мереж та значні інвестиції в розробку програмного забезпечення й апаратних засобів для їх реалізації свідчать про високий інтерес до створення штучних нейронних мереж. ШНМ є адаптивними системами, які здатні навчатися розв'язувати складні завдання. Для успішного навчання необхідно обрати правильну структуру нейромережової моделі та мати достатню кількість даних для навчання.

2. Нейромережові технології управління допомагають вирішувати багато труднощів, які виникають при роботі з нелінійними об'єктами або об'єктами з невідомою структурою, що не піддаються звичайним методам адаптивного управління. Здатність ШНМ моделювати складні нелінійні відображення забезпечується використанням сигмоїдальних функцій активації (або інших нелінійних функцій загального вигляду) у нейронах прихованих шарів.

3. Особливою характеристикою нейронних мереж є їх універсальність. Завдяки цій властивості та перспективності для вирішення глобальних завдань, таких як створення штучного інтелекту і моделювання процесу мислення, вони є важливою сферою дослідження, яка потребує глибокого вивчення.

4. Показана ефективність використання багатоварових мереж прямого розповсюдження для побудови нейромережової системи управління. Двошарова мережа з прямою передачею сигналу може достатньо точно відтворити будь-яку функцію з кінцевим числом точок розриву, якщо задати достатнє число нейронів прихованого шару. Розглянуті питання навчання нейронних мереж, вибрано метод навчання нейронної мережі регулятора.

5. Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання динамічних систем є отримання їх адекватних математичних моделей. Показано, що завдання ідентифікації нелінійних динамічних об'єктів успішно вирішуються за допомогою штучних нейронних мереж. Найбільше за-

стосування для ідентифікації отримали багатошаровий персептрон і радіально-базисні мережі.

6. Для оптимізації роботи об'єкта в необхідна корекція параметрів регуляторів відповідно до зміни робочих умов. У зв'язку з цим доцільною є розробка систем керування на основі адаптивного підходу в поєднанні з методами теорії ШНМ, що дозволить, з одного боку, побудувати досить прості нейромереві моделі, а з іншого – адаптивно корегувати параметри як цих моделей, так і регулятора відповідно до умов, що змінюються.

7. В результаті аналізу структур систем нейрорегулювання, актуальних в даний час, вибраний принцип нелінійного предиктивного регулювання при побудові нейромережевого регулятора системи.

8. Показана доцільність використання нейромережевих технологій для синтезу нейромережевої системи управління механізмом обертання колеса роторного екскаватора. При несприятливих поєднаннях швидкостей роторного колеса, пружних властивостей системи ротор-стріла-підвіска стріли, фортеці породи і т.п. можуть виникати резонансні коливання металоконструкцій, що несприятливо впливає на роботу механізму. Характер перехідних процесів повинен бути оптимальним, таким, що забезпечує обмеження динамічних навантажень механізму і металоконструкцій. Традиційні системи управління не можуть забезпечити необхідну якість функціонування систем управління механізмом обертання колеса роторного екскаватора.

9. На підставі аналізу технічних вимоги, що пред'являється до електроприводів основних механізмів роторних екскаваторів для електроприводу механізму обертання колеса вибраний електропривод постійного струму, виконаний за системою Г–Д.

10. Проаналізовані нейрмережеві підходи до рішення задачі управління. Сформульовано задачу дослідження, що складається в розробці нейромережевої системи управління механізмом обертання колеса роторного екскаватора з урахуванням пружних механічних зв'язків.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ОБЕРТАННЯ КОЛЕСА РОТОРНОГО ЕКСКАВАТОРА

2.1 Основні технічні характеристики роторного екскаватора. Вимоги до системи управління

Вихідні дані для розрахунку електроприводу механізму обертання колеса роторного екскаватора і синтезу системи управління наведені в табл. 2.1.

Таблиця 2.1 – Технічні дані роторного екскаватора

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
1	Теоретична продуктивність по рих- лій масі	$m^3 \cdot год$	Q	1700
2	Діаметр роторного колеса	m	D_p	8
3	Бічне зусилля різання	H	F_{δ}	$1,2 \cdot 10^6$
4	Питоме зусилля копання	$кг/м^2$	k_f	$3,8 \cdot 10^4$
5	Радіус різання	m	R_0	23,3
6	Кутова швидкість повороту платфо- рми	c^{-1}	$\omega_{пл}$	$2,9 \cdot 10^{-2}$
7	Момент площ поворотної платфор- ми, що піддається дії вітру щодо осі обертання	$m^2 \cdot m$	$M_{пв}$	430
8	Швидкість руху транспортної стріч- ки	m/c	V_t	3,5
9	Маса поворотної платформи	$кг$	$m_{пл}$	$500 \cdot 10^3$

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
10	Маса приймальної частини	кг	$m_{пч}$	$127 \cdot 10^3$
11	Маса відвальної частини	кг	$m_{вч}$	$120 \cdot 10^3$
12	Діаметр роликового круга	м	$D_{кр}$	8,7
13	Діаметр опорних роликів	м	$d_{тк}$	0,8
14	Момент інерції поворотної платформи	кг · м ²	$J_{пл}$	$17 \cdot 10^7$
15	Частота обертання ротора	рад/хв	ω_p	6
16	Щільність ґрунту	кг/м ³	γ_p	$1,8 \cdot 10^3$
17	Коефіцієнт аеродинамічного опору		$k_{ад}$	1,4
18	Максимально допустиме вітрове навантаження	Па	$\lambda_{вт}$	300
19	Максимально подоланий ухил		γ	10°
20	Число двигунів механізму повороту		k	2
21	Максимально допустимі прискорення і уповільнення платформи	с ⁻²	$\omega_{пл}$	0,005
22	ККД редуктора роторного колеса		η_p	0,8
23	ККД редуктора механізму повороту		$\eta_{п}$	0,45 – 0,65
24	Максимальний кут повороту		$\alpha_{п \max}$	±75°

Вимоги до показників якості функціонування системи управління.

Система управління повинна бути астатичною.

Перерегулювання швидкості при ступінчастій входній дії не більше 10%, час регулювання не більше 5с.

2.2 Розрахунок потужності і вибір електродвигуна механізму обертання колеса роторного екскаватора

Режим роботи електроприводу роторного колеса тривалий, при цьому потужність на валу двигуна визначається співвідношенням:

$$P_d = \frac{P_{p\Sigma}}{\eta_p}, \quad (2.1)$$

де η_p – ККД редуктора роторного колеса;

$P_{p\Sigma}$ – сумарна потужність на валу роторного колеса яка визначається наступним чином

$$P_{p\Sigma} = k_t \cdot (P_p + P_{\pi}) \quad (2.2)$$

де $k_t = 1,03 \div 1,05$ – коефіцієнт, що враховує навантаження на валу роторного колеса, від тертя ковшів об породу і навантаження, пов'язане з повідомленням окремої від масиву породи кінематичної енергії;

P_p – потужність копання

$$P_p = \frac{Q \cdot k_f}{3600 \cdot k_p} \cdot 10^{-3} \quad (2.3)$$

де Q – продуктивність екскаватора;

k_f – коефіцієнт питомого опору копанню;

k_p – коефіцієнт розпушування ґрунту; $k_p = 1,1 \div 1,5$, приймаємо $k_p = 1,3$

$$P_p = \frac{1700 \cdot 1,2 \cdot 10^6}{3600 \cdot 1,2} \cdot 10^{-3} = 472 \text{ кВт}$$

P_{π} – потужність, необхідна для здійснення роботи по подоланню сили тяжіння при підйомі породи після її відділення від масиву:

$$P_{\text{п}} = Q \cdot \frac{\gamma_{\text{п}} \cdot g \cdot D_{\text{р}}}{k_{\text{р}} \cdot 7200} \cdot 10^{-3} \quad (2.4)$$

де γ – щільність породи;
 g – прискорення вільного падіння;
 $D_{\text{р}}$ – діаметр роторного колеса

$$P_{\text{п}} = 1700 \cdot \frac{1,8 \cdot 10^{-3} \cdot 9,81 \cdot 8,7}{1,2 \cdot 7200} \cdot 10^{-3} = 30,22 \text{ кВт}$$

Сумарна потужність

$$P_{\text{р}\Sigma} = 1,04 \cdot (472 + 30,22) = 522,3 \text{ кВт}$$

Потужність на валу двигуна

$$P_{\text{д}} = \frac{522,3}{0,8} = 653 \text{ кВт}$$

Вибираємо електродвигун постійного струму типу ПСЕ-174-7К з наступними технічними даними: номінальна потужність $P_{\text{н}} = 860 \text{ кВт}$, тривалість включення ПВ=100%; номінальна напруга $U_{\text{н}} = 480 \text{ В}$, номінальна швидкість обертання $n_{\text{н}} = 560 \text{ об / хв}$, номінальний струм $I_{\text{н}} = 1900 \text{ А}$, опір обмотки якоря $R_{\text{я д}} = 0,00229 \text{ Ом}$, опір обмотки додаткових полюсів $R_{\text{дп д}} = 0,00091 \text{ Ом}$, опір компенсаційної обмотки $R_{\text{ко д}} = 1,23 \cdot 10^{-3} \text{ Ом}$, номінальний магнітний потік полюса $\Phi_{\text{н д}} = 0,0185 \text{ Вб}$, число пар полюсів $2p = 8$, число пар паралельних гілок обмотки якоря $2a = 12$, ККД $\eta_{\text{д}} = 0,93$, момент інерції $J_{\text{д}} = 380 \text{ кг} \cdot \text{м}^2$

Оскільки двигун механізму обертання роторного колеса працює в тривалому режимі і періоди пуску і гальмування складають незначну частину від часу роботи, то перевірка двигуна по нагріву не потрібна.

У перехідних режимах динамічний момент обмежується відповідним вибором прискорення електроприводу.

2.3 Вибір основного електроустаткування електроприводу

Потужність генератора в системі Г-Д визначаються з умови

$$P_{нг} \geq \frac{P_{нд}}{\eta_d}, \quad (2.5)$$

$$P_{нг} \geq \frac{860}{0,93} = 925 \text{ кВт}$$

Повинні також виконуватись співвідношення: $I_{нг} \geq I_{нд}$; $U_{нг} \geq U_{нд}$.

Вибираємо генератор постійного струму типу генератор ПЕ-172-7К. Технічні дані генератора: номінальна потужність $P_n = 950 \text{ кВт}$; номінальна напруга $U_n = 500 \text{ В}$; номінальна швидкість обертання $n_n = 1000 \text{ об/хв}$; номінальний струм $I_n = 1900 \text{ А}$; опір обмотки якоря $R_{яг} = 0,0229 \text{ Ом}$, опір обмотки додаткових полюсів $R_{дпг} = 0,7 \cdot 10^{-3} \text{ Ом}$; опір компенсаційної обмотки $R_{ког} = 0,00123 \text{ Ом}$; число пар полюсів $2p = 8$; номінальна напруга збудження $U_{нзб} = 220 \text{ В}$; опір обмотки збудження $R_{збг} = 5,76 \text{ Ом}$; номінальний магнітний потік полюса $\Phi_{нг} = 69,5 \text{ мВб}$; число витків обмотки збудження $W_{зб} = 336$; ККД $\eta_r = 0,938$.

Струм обмотки збудження:

$$I_{зг} = \frac{U_{нз}}{R_{збг}} \quad (2.6)$$

$$I_{зг} = \frac{220}{5,76} = 38,19 \text{ А}.$$

Для живлення обмотки збудження генератора вибираємо тиристорний перетворювач – збудник БТУ-3601-3627. Технічні дані тиристорного збудника: номінальна потужність $P_{тзн} = 9,2 \text{ кВт}$; номінальна випрямлена напруга $U_{дн} = 230 \text{ В}$; номінальний випрямлений струм $I_{дн} = 40 \text{ А}$; схема випрямлення – реверсивна трифазна нульова; управління групами тиристорів – роздільне.

Анодний трансформатор типа ТСЗП-25/0,7-74, втрати холостого ходу $P_{хх} = 170 \text{ Вт}$, втрати короткого замикання $P_{кз} = 1300 \text{ Вт}$.

2.4 Розрахунок і побудова статичних характеристик розімкненої і замкненої систем

Після розробки системи управління механізмів роторного колеса необхідно провести детальне дослідження статичних і динамічних характеристик системи.

На рис. 2.1 зображена схема системи Г-Д.

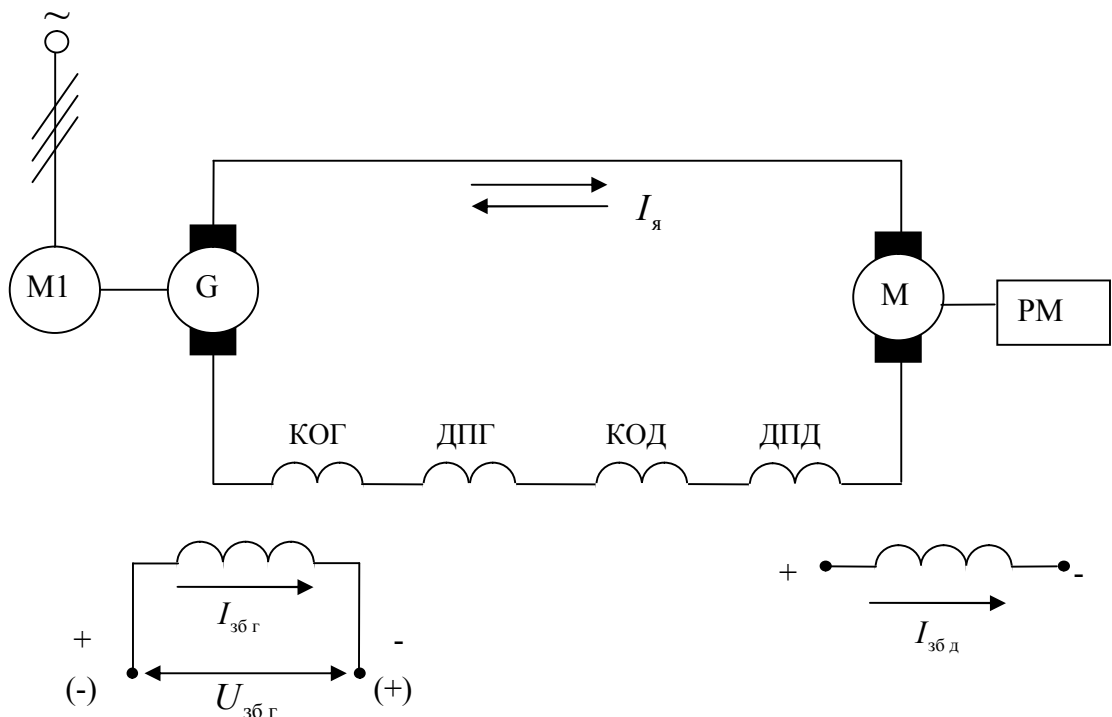


Рисунок 2.1 – Принципова схема системи Г-Д

На схемі прийняті наступні позначення: M , G , M – двигун, генератор і приводний двигун генератора;

$КОГ$, $ДПД$, $ДПГ$ – компенсаційна обмотка генератора і обмотки додаткових полюсів двигуна і генератора відповідно;

$I_я$ – струм якорного ланцюга;

$I_{зб\ д}$, $I_{зб\ г}$ – струми в обмотках збудження двигуна M і в незалежній обмотці збудження генератора G

PM – робочий механізм.

Схема рис 6.1 відповідає числу приводів $k = 1$ оскільки для електроприводу механізму обертання роторного колеса завжди використовується один двигун.

Як впливає з схеми, якірні обмотки двигуна і генератора сполучені послідовно і утворюють якірний ланцюг. Якщо розглядати ЕДС генератора E_r як джерело живлення двигуна, то рівняння електромеханічної і механічної характеристик двигуна матиме вигляд:

$$\omega_d = \frac{E_r}{k k_d \Phi_{нд}} - I_{я} \frac{R_{я\Sigma}}{k k_d \Phi_{нд}} = \omega_0 - \Delta\omega_{ст} ; \quad (2.7)$$

$$\omega_d = \frac{E_r}{k k_d \Phi_{нд}} - M_d \frac{R_{я\Sigma}}{k (k_d \Phi_{нд})^2} = \omega_0 - \Delta\omega_{ст} \quad (2.8)$$

де M_d – момент, що розвивається двигуном;

k_d – конструктивний коефіцієнт двигуна;

$\Phi_{нд}$ – номінальний магнітний потік двигуна;

$R_{я\Sigma}$ – сумарний опір якірного ланцюга

$$R_{я\Sigma} = R_{я\Sigma д} + R_{я\Sigma Г} ; \quad (2.9)$$

$R_{я\Sigma д}$ – сумарний опір якірного ланцюга двигуна, Ом,

$$R_{я\Sigma д} = k_t (R_{я д} + R_{дп д} + R_{ко д}) ; \quad (2.10)$$

$R_{я д}$, $R_{дп д}$, $R_{ко д}$ – опір обмотки якоря, компенсаційної і додаткових полюсів двигуна, Ом;

k_t – температурний коефіцієнт; у каталогах приведений опір обмоток електричних машин при $20^\circ C$, отже, при розрахунках його слід приводити до робочої температури, тобто помножити на температурний коефіцієнт $k_t = 1,36$

$$R_{\Sigma \Gamma} = 1,36 \cdot (0,00229 + 0,00091 + 0,00123) = 4,352 \cdot 10^{-2} \text{ Ом},$$

$R_{\Sigma \Gamma}$ – сумарний опір якірного ланцюга генератора, Ом:

$$R_{\Sigma \Gamma} = k_t (R_{\Gamma} + R_{\text{дп } \Gamma} + R_{\text{ко } \Gamma}), \quad (2.11)$$

$$R_{\Sigma \Gamma} = 1,36 \cdot (0,0229 + 0,00123 + 0,0007) = 3,377 \cdot 10^{-2} \text{ Ом},$$

$$R_{\Sigma} = 4,352 \cdot 10^{-2} + 3,377 \cdot 10^{-2} = 3,812 \cdot 10^{-2} \text{ Ом}.$$

Добуток k_d на $\Phi_{\text{нд}}$ (позначимо надалі як c) можна визначити з виразу для природної електромеханічної характеристики двигуна:

$$c = k_d \Phi_{\text{нд}} = \frac{U_{\text{нд}} - I_{\text{нд}} R_{\Sigma \Gamma}}{\omega_{\text{нд}}}; \quad (2.12)$$

$U_{\text{нд}}, I_{\text{нд}}$ – номінальні напруга і струм двигуна;

$\omega_{\text{н}}$ – номінальна швидкість обертання двигуна c^{-1}

$$\omega_{\text{н}} = \frac{\pi \cdot n_{\text{н}}}{30}, \quad (2.13)$$

$$\omega_{\text{н}} = \frac{3,14 \cdot 560}{30} = 58,6 c^{-1}.$$

$$c = k_d \Phi_{\text{нд}} = \frac{480 - 1900 \cdot 4,352 \cdot 10^{-3}}{58,6} = 8,02 \text{ В} \cdot c / \text{рад}$$

При побудові механічних і електромеханічних характеристик системи Г-Д в якості максимальної ЕДС генератора E_{Γ} приймаємо номінальну напругу двигуна, тобто $E_{\Gamma} = U_{\text{н}} = 480 \text{ В}$.

Підставляємо числові значення параметрів в (2.7) і (2.8), отримаємо:

$$\omega_d = \frac{E_r}{8,02} - I_{\text{я}} \frac{0,03812}{8,02} = E_r \cdot 0,1246 - I_{\text{я}} \cdot 4,75 \cdot 10^{-3}; \quad (2.14)$$

$$\omega_d = \frac{E_r}{8,02} - M_d \frac{0,03812}{(8,02)^2} = E_r \cdot 0,1246 - M_d \cdot 0,59 \cdot 10^{-3}. \quad (2.15)$$

Оскільки характеристики мають вид прямих ліній, то для їх побудови досить визначити координати двох точок, наприклад точки холостого ходу і точки, що відповідає номінальному навантаженню. Крім номінального значення $E_r = U_n = 480 \text{ В}$, будуємо характеристики також при

$$E_{r2} = 0,75 \cdot E_{3 \text{ max}} = 0,75 \cdot 480 = 360 \text{ В},$$

$$E_{r3} = 0,5 \cdot E_{3 \text{ max}} = 0,5 \cdot 480 = 240 \text{ В},$$

$$E_{r4} = 0,25 \cdot E_{3 \text{ max}} = 0,25 \cdot 480 = 120 \text{ В}$$

$$E_{r5} = 0 \text{ (характеристика динамічного гальмування).}$$

1 точка – точка ідеального холостого ходу

$$I_{\text{я}} = 0, \quad M = 0$$

$$\omega_{01} = E_{r1} \cdot 0,1246 = 480 \cdot 0,1246 = 59,8 \text{ с}^{-1}$$

$$\omega_{02} = E_{r2} \cdot 0,1246 = 360 \cdot 0,125 = 44,9 \text{ с}^{-1}$$

$$\omega_{03} = E_{r3} \cdot 0,1246 = 240 \cdot 0,1246 = 29,9 \text{ с}^{-1}$$

$$\omega_{04} = E_{r4} \cdot 0,125 = 120 \cdot 0,1246 = 15,6 \text{ с}^{-1}$$

$$\omega_{05} = 0$$

2 точка – відповідає номінальному режиму

$$I_{\text{я}} = I_{\text{нд}} = 1900 \text{ А},$$

$$M = M_{\text{н}} = k_{\text{д}} \Phi_{\text{нд}} \cdot I_{\text{нд}} = 8,023 \cdot 1900 = 15244 \text{ Нм}$$

$$\omega_1 = 480 \cdot 0,12469 - 1900 \cdot 4,75 \cdot 10^{-3} = 55,3 \text{ Ом}$$

$$\omega_2 = 360 \cdot 0,12469 - 1900 \cdot 4,75 \cdot 10^{-3} = 40,4 \text{ Ом}$$

$$\omega_3 = 240 \cdot 0,12469 - 1900 \cdot 4,75 \cdot 10^{-3} = 25,4 \text{ Ом}$$

$$\omega_4 = 120 \cdot 0,12469 - 1900 \cdot 4,75 \cdot 10^{-3} = 11,08 \text{ Ом}$$

$$\omega_5 = 0 \cdot 0,12469 - 1900 \cdot 4,75 \cdot 10^{-3} = -4,514 \text{ с}^{-1}$$

Графіки електромеханічних і механічних характеристик розімкненої системи приведені на рис. 2.2 (пунктирні прямі). Для від'ємних значень напруга характеристики являються симетричними відносно початку координат.

Оскільки сумарний опір якірного ланцюга в системі Г-Д збільшується в порівнянні з опором якірного ланцюга двигуна, то жорсткість механічної характеристики $\beta = \frac{(k_{\text{д}} \Phi_{\text{нд}})^2}{R_{\text{я}\Sigma}}$ у системі Г-Д нижче за жорсткість природної характеристики двигуна.

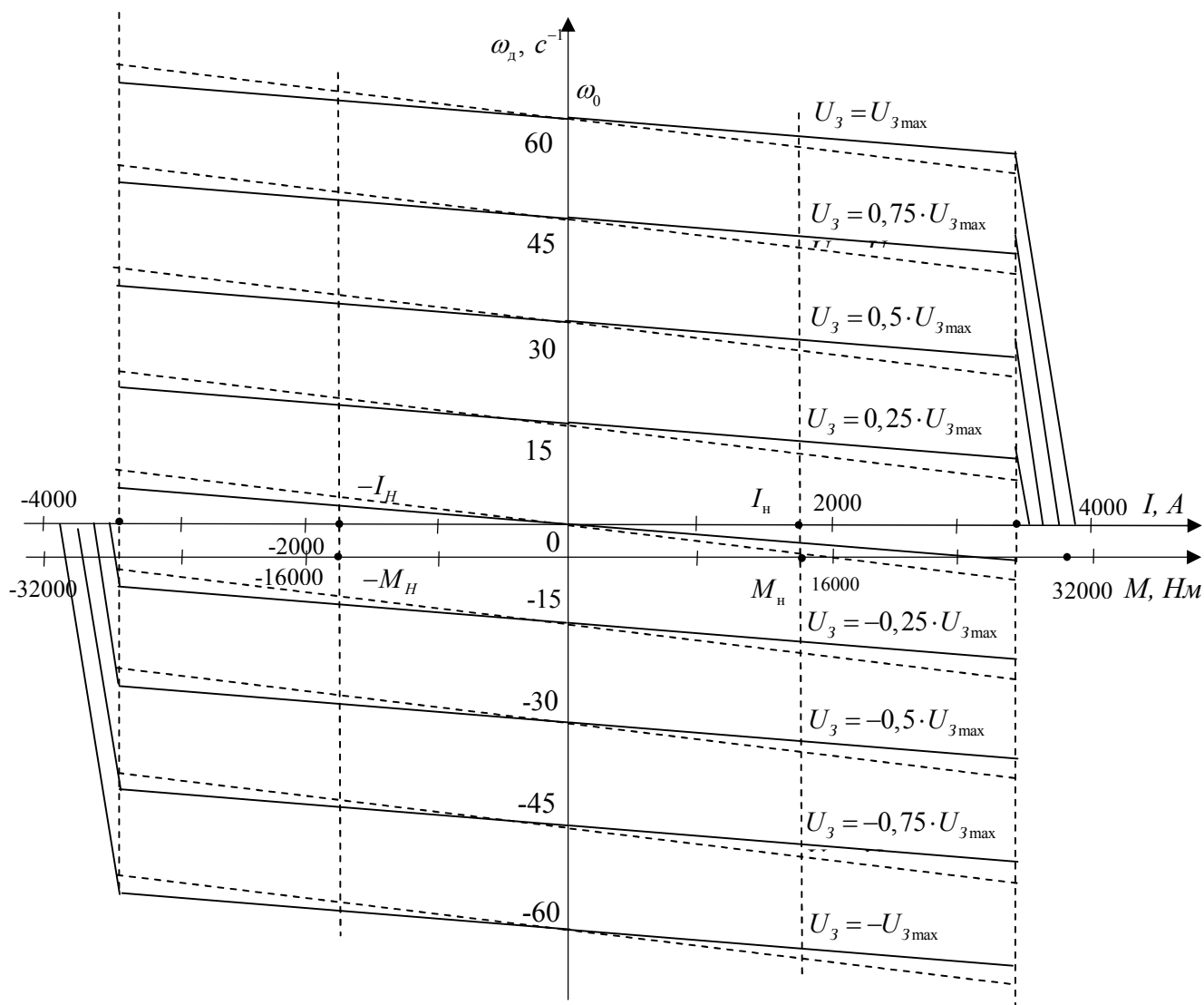


Рисунок 2.2 – Механічні і електромеханічні характеристики розімкненої (пунктирні лінії) і замкненої (суцільні лінії) системи управління при різних значеннях напруги

До електроприводів більшості основних механізмів роторних екскаваторів пред'являються підвищені вимоги відносно жорсткості робочої частини механічних характеристик і обмеження максимального значення моменту, що розвивається, тобто необхідності формування «екскаваторної» характеристики, показаної на рис. 2.3.

На робочій ділянці 1 у міру зростання навантаження потрібне аж до струму відсічення $I_{\text{я}} = I_{\text{відс}}$ (відповідає моменту відсічення $M_{\text{відс}}$) пропорційне струму $I_{\text{я}}$ збільшення ЕДС генератора, $E_{\text{г}}$ чим компенсується зростаюче падін-

ня напруги на опорі R_{Σ} і забезпечується підвищення жорсткості робочої ділянки характеристики. Подальше збільшення навантаження повинне супроводжуватися пропорційним різниці $\Delta I_{\text{я}} = I_{\text{я}} - I_{\text{відс}}$ зниженням $E_{\text{г}}$ і, отже, швидкості приводу, яка досягає нуля $\omega = 0$ при струмі $I_{\text{я}} = I_{\text{стоп}}$ (відповідає моменту сто-

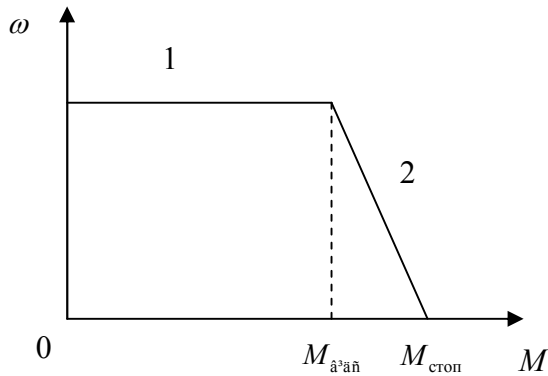


Рисунок 2.3 – Екскаваторна механічна характеристика

поріння $M_{\text{д}} = M_{\text{стоп}}$) і ЕДС генератора $E_{\text{г}} = I_{\text{стоп}} R_{\Sigma}$ (ділянка 2 характеристики).

Забезпечити формування такої характеристики можна тільки в замкнутих САУ електроприводом. На роторних екскаваторах використовуються САУ, що мають структуру з підсумовуючим підсилювачем, на вході якого підсумовуються сигнали завдання і сигнали зворотних зв'язків. Зворотні зв'язки можуть

бути різних типів: жорсткі і гнучкі ЗЗ по напрузі генератора, ЗЗ за швидкістю двигуна, ЗЗ по струму з відсіченням.

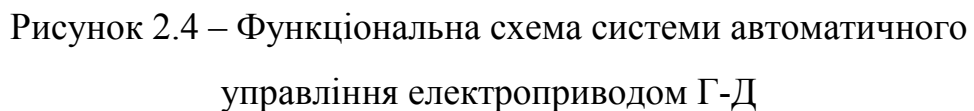
У роботі для електроприводу роторного колеса використана САУ із зворотним зв'язком за швидкістю, гнучким ЗЗ по струму і ЗЗ по струму з відсіченням. Функціональна схема системи показана нарис. 2.4.

На схемі прийняті наступні позначення: AJ – задатчик інтенсивності; AW – підсумовуючий підсилювач; UL – збудник генератора; UA – датчик струму; BR – датчик швидкості (тахогенератор); HE – нелінійний елемент; DE – диференціюючий елемент.

Задатчик інтенсивності AJ забезпечує в динаміці при довільній формі вхідного сигналу $U_{\text{вх}}$ сигнал, що лінійно змінюється в часі $U_{\text{з}}$ до тих пір, поки не буде досягнута рівність

$$U_{\text{з}} = k' \cdot U_{\text{вх}}, \quad (2.16)$$

де k' – коефіцієнт пропорційності.



Задатчик інтенсивності і гнучкості 33 не беруть участь у формуванні статичних характеристик приводу. Істотно для формування характеристик нелінійності має нелінійний елемент HE , що стоїть в ланцюзі 33 по струму з відсіченням. Його характеристика вхід-вихід містить зону нечутливості, за допомогою якої створюється затримка введення сигналу 33 до тих пір, поки струм якоря лежить в межах

У даному приводі зазвичай витримується рівність

$$I_{\text{вїдс1}} = |I_{\text{вїдс2}}| = I_{\text{вїдс}} = 2 I_{\text{н д}} . \quad (2.18)$$

54

$$\begin{aligned}\omega_{\text{д}} &= U_3 \frac{k_{\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} - 1(\Delta I) (I_{\text{я}} - I_{\text{відс}}) \frac{k_{\text{т відс}} k_{\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} = \\ &= \omega_0 - \Delta \omega_{\text{ст 1}} - 1(\Delta I) \Delta \omega_{\text{ст 2}} ,\end{aligned}\quad (2.19)$$

де k_{Σ} – сумарний коефіцієнт посилення, рівний добутку коефіцієнтів посилення елементів AW , UL і G на лінійній частині їх характеристики вхід-вихід;

$$k_{\Sigma} = k_{AW} k_{UL} k_{\Gamma}; \quad (2.20)$$

$k_{\text{ззш}}$, $k_{\text{т відс}}$ – коефіцієнти ЗЗ за швидкістю і струму з відсіченням;

$1(\Delta I)$ – одинична функція, рівна 0 при $|I_{\text{я}}| \leq I_{\text{відс}}$ і рівна 1 при $|I_{\text{я}}| > I_{\text{відс}}$.

Теоретично при прагненні k_{Σ} або $k_{\text{ззш}}$ до ∞ досягається нескінченна жорсткість характеристик. Практично зростання жорсткості механічної характеристики обмежене можливістю втрати стійкості САУ і фізичною реалізовується великих значень k_{Σ} , $k_{\text{ззш}}$, U_3 .

Коефіцієнт посилення генератора

$$k_{\Gamma} = \frac{E_{\text{н}\Gamma}}{U_{\text{н зб}\Gamma}} = \frac{U_{\text{н}\Gamma} + I_{\text{н}\Gamma} R_{\text{я}\Sigma\Gamma}}{U_{\text{н зб}\Gamma}}, \quad (2.21)$$

де $U_{\text{н}\Gamma}$, $U_{\text{н зб}\Gamma}$ – номінальна напруга на якорі і номінальна напруга збудження генератора;

$I_{\text{н}\Gamma}$ – номінальний струм генератора;

$E_{\text{н}\Gamma}$ – номінальна ЕДС генератора;

$$E_{\text{н}\Gamma} = U_{\text{н}\Gamma} + I_{\text{н}\Gamma} R_{\text{я}\Sigma\Gamma}. \quad (2.22)$$

$$E_{\Gamma} = 500 + 1900 \cdot 3,377 \cdot 10^{-2} = 564 \text{ В}$$

$$k_{\Gamma} = \frac{564}{220} = 2,56.$$

Коефіцієнт посилення тиристорного збудника приблизно можна визначити як

$$k_{UL} = \frac{U_{днзб}}{U_y}, \quad (2.23)$$

де $U_{днзб}$ – номінальна випрямлена напруга збудника, приймаємо $U_{днзб} = U_{нзбг}$;

U_y – напруга управління на вході тиристорного збудника; оскільки в каталозі значення U_y не вказані, приймаємо $U_y = 10 В$.

$$k_{UL} = \frac{230}{10} = 23.$$

Коефіцієнт посилення суматора приймаємо $k_{AW} = 0,1$.

$$k_{\Sigma} = 0,1 \cdot 23 \cdot 2,56 = 5,89.$$

Коефіцієнт зворотного зв'язку за швидкістю $k_{ззш}$ розраховуємо, виходячи з необхідного статичного перепаду швидкості $\Delta\omega_{ст1}$ при номінальному струмі двигуна:

$$\Delta\omega_{ст1} = I_{нд} \frac{R_{я\Sigma}}{c + k_{\Sigma} k_{ззш}}, \quad (2.24)$$

звідки

$$k_{ззш} = I_{нд} \frac{R_{я\Sigma}}{k_{\Sigma} \Delta\omega_{ст1}} - \frac{c}{k_{\Sigma}}. \quad (2.25)$$

Значення $\Delta\omega_{ст1}$ приймаємо рівним половині статичного перепаді швидкості в розімкненій системі $\Delta\omega_{ст}$ при номінальному навантаженні

$$\Delta\omega_{ст1} = 0,5 \Delta\omega_{ст} = 0,5(\omega_0 - \omega_n) = 0,5 \frac{I_n R_{я\Sigma}}{k_d \Phi_{нд}}, \quad (2.26)$$

$$\Delta\omega_{\text{ст1}} = 0,5 \cdot \frac{1900 \cdot 3,812 \cdot 10^{-2}}{8,02} = 4,514 \text{ c}^{-1},$$

$$k_{\text{ззш}} = 1900 \cdot \frac{3,812 \cdot 10^{-2}}{5,89 \cdot 4,514} - \frac{8,02}{5,89} = 1,33.$$

Розрахунок коефіцієнта відсічення по струму $k_{\text{т відс}}$ проводиться в цілях отримання заданого максимального струму стопоріння по формулі:

$$k_{\text{т відс}} = \frac{U_{\text{з max}} - I_{\text{стоп}} R_{\Sigma} / k_{\Sigma}}{I_{\text{стоп}} - I_{\text{відс}}}. \quad (2.27)$$

Значення $I_{\text{відс}}$ і $I_{\text{стоп}}$ приймаємо

$$I_{\text{відс}} = 2I_{\text{нд}}; \quad I_{\text{стоп}} = 2,3I_{\text{нд}}. \quad (2.28)$$

$$I_{\text{відс}} = 2I_{\text{нд}} = 2 \cdot 1900 = 3800 \text{ A},$$

$$I_{\text{стоп}} = 2,3I_{\text{нд}} = 2,3 \cdot 1900 = 4370 \text{ A}.$$

Величину напруги завдання $U_{\text{з max}}$ визначаємо з умови отримання при $U_{\text{з max}}$ швидкості ідеального холостого ходу розімкненої системи при $E_{\Gamma} = k U_{\text{нд}}$.

$$\omega_0 = U_{\text{з max}} \frac{k_{\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} = \frac{U_{\text{нд}}}{c}, \quad (2.29)$$

звідки

$$U_{\text{з max}} = \frac{U_{\text{нд}} (c + k_{\Sigma} k_{\text{ззш}})}{c k_{\Sigma}}, \quad (2.30)$$

$$U_{\text{з max}} = \frac{480(8,02 + 5,89 \cdot 1,33)}{8,02 \cdot 5,89} = 161 \text{ B}.$$

$$U_{\text{т відс}} = \frac{161 - 4370 \cdot 3,812 \cdot 10^{-2} / 5,89}{4370 - 3800} = 0,287.$$

Як видно з виразу (2.27), забезпечення абсолютно м'якої характеристики на ділянці відсічення (при $I_{\text{відс}} = I_{\text{стоп}}$) зажадало б нескінченно великого коефіцієнта посилення $k_{\text{т відс}}$. Практично максимум $k_{\text{т відс}}$ обмежений фізичною реалізуємою величиною коефіцієнта посилення $k_{\text{т відс}}$ і необхідністю збереження стійкості САУ в режимі стопоріння.

Будуємо статичні характеристики замкненої системи. Оскільки характеристики є прямі лінії, то для їх побудови досить дві точки на кожній ділянці роботи: точки ідеального холостого ходу, номінальної точки і точки стопоріння.

Оскільки момент двигуна пропорційний струму: $M_{\text{д}} = k_{\text{д}} \Phi_{\text{нд}} I_{\text{д}} \Rightarrow M_{\text{д}} \equiv I_{\text{д}}$, то механічні характеристики не розраховуємо, а досить розрахувати тільки електромеханічні характеристики $\omega_{\text{д}} = f(I_{\text{я}})$.

Підставимо числові значення у рівняння електромеханічної характеристики замкненої системи (2.19)

$$\begin{aligned} \omega_{\text{д}} &= U_3 \frac{k_{\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} - 1(\Delta I)(I_{\text{я}} - I_{\text{відс}}) \frac{k_{\text{т відс}} k_{\Sigma}}{c + k_{\Sigma} k_{\text{ззш}}} = \\ &= U_3 \frac{5,89}{8,02 + 5,89 \cdot 1,33} - I_{\text{я}} \frac{3,812 \cdot 10^{-2}}{8,02 + 5,89 \cdot 1,33} - 1(\Delta I)(I_{\text{я}} - 3800) \cdot \frac{0,287 \cdot 5,89}{8,02 + 5,89 \cdot 1,33} = \\ &= U_3 \cdot 0,37 - I_{\text{я}} \cdot 0,24 \cdot 10^{-2} - 1(\Delta I)(I_{\text{я}} - 3800) \cdot 0,105. \end{aligned}$$

Використовуючи отримане рівняння, будуємо характеристики замкнутої системи для декількох значень напруги завдання U_3 , а саме, для значень U_3 , рівних $U_3 = (1; 0,75; 0,5; 0,25; 0; -0,25; -0,5; -0,75; -1) U_{3 \text{ max}}$.

$$U_3 = U_{3\max} = 192 \text{ В}.$$

1 точка (точка холостого ходу).

$$I_{\text{я}} = 0, \quad \omega_{\text{д}} = 192 \cdot 0,37 - 0 \cdot 0,24 \cdot 10^{-2} - 0 \cdot (-3800) \cdot 0,105 = 59,8 \text{ с}^{-1}.$$

2 точка (номінальна точка).

$$I_{\text{я}} = I_{\text{нд}} = 1900 \text{ А}, \quad \omega_{\text{д}} = 192 \cdot 0,37 - 1900 \cdot 2,4 \cdot 10^{-3} - 0 = 53,7 \text{ с}^{-1}$$

3 точка (точка, що відповідає режиму стопоріння).

$$\omega_{\text{д}} = 0, \quad I_{\text{я}} = I_{\text{стоп}} = 4370 \text{ А}.$$

Аналогічно проводимо розрахунки для $U_3 = 0,75U_{3\max}$; $0,5U_{3\max}$; $0,25U_{3\max}$; 0 . Для $-0,25U_{3\max}$; $-0,5U_{3\max}$; $-0,75U_{3\max}$ характеристики будуються по тих же точках, але з від'ємними значеннями.

Результати розрахунків зводимо в таблицю 2.1.

Таблиця 2.1 – Розрахунок статичних характеристик системи

U_3	$U_3, \text{В}$	$\omega_{\text{д}}, \text{с}^{-1}$ при $I_{\text{я}} = 0$	$\omega_{\text{д}}, \text{с}^{-1}$ при $I_{\text{я}} = I_{\text{нд}}$	$I_{\text{стоп}}, \text{А}$ $\omega_{\text{д}} = 0$
$U_{3\max}$	192	59,8	55,7	4370
$0,75U_{3\max}$	143,7	44,9	40,6	4206
$0,5U_{3\max}$	95,8	29,9	25,3	4042
$0,25U_{3\max}$	47,9	15,6	10,07	3879
0	0	0	-4,5	-

Графіки електромеханічних і механічних характеристик замкнутої системи приведені на рис.2.2 (суцільні лінії). Для від'ємних значень напруги характеристики являються симетричними відносно початку координат.

2.5 Дослідження динамічних характеристик системи

Окрім статичних характеристик, для будь-якої системи необхідний аналіз її динамічних характеристик; а саме: реакція системи на дві зовнішні дії: управляючу дію – напругу завдання U_3 і збурюючу дію – момент $M_{ст}$ ($I_{ст}$ струм) статичного навантаження. Для дослідження динамічних властивостей необхідно розглянути алгоритмічну схему рис.2.5, побудовану відповідно до функціональної схеми. У алгоритмічній схемі не приведений ЗЗ по струму з відсіченням, оскільки він не бере участь в робочих режимах, і задатчик інтенсивності, оскільки його вплив можна врахувати шляхом подачі сигналу завдання $U_3(t)$, що лінійно змінюється в часі.

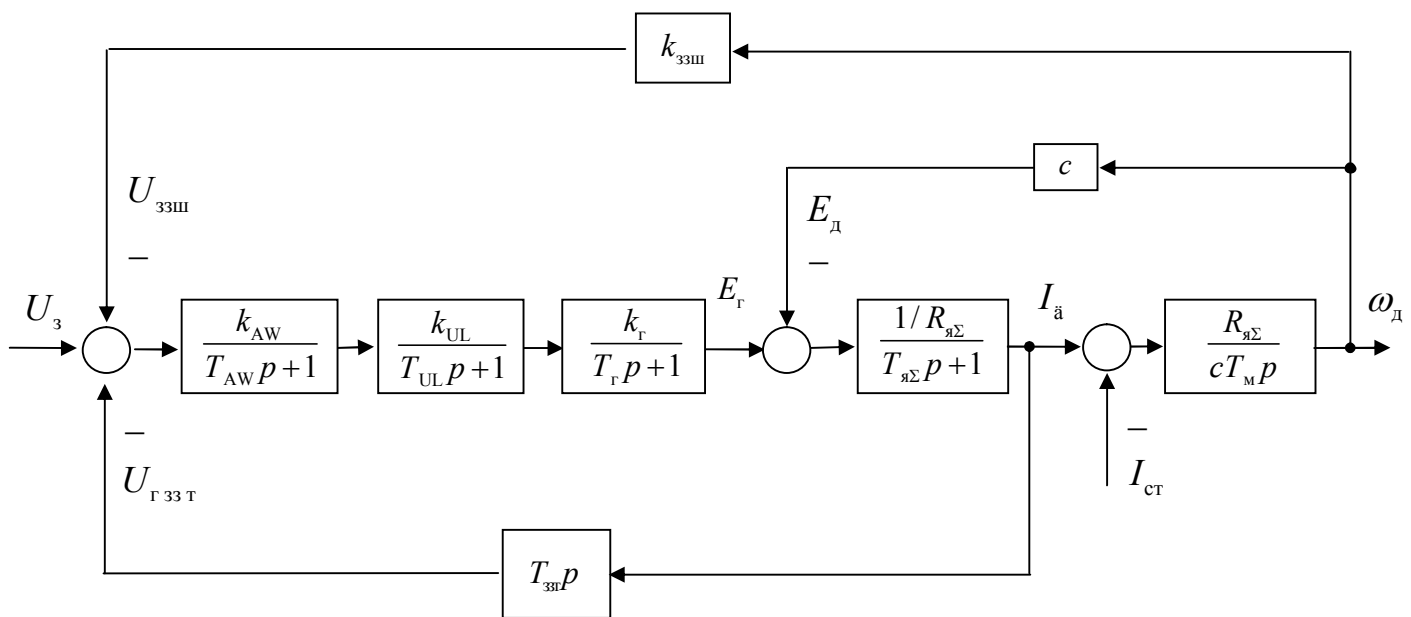


Рисунок 2.5 – Алгоритмічна схема системи

Визначимо значення параметрів системи.

При визначенні постійної часу генератора $T_Г$ вважаємо, що вона приблизно рівна постійній часу незалежної обмотки збудження (це справедливо за відсутності паралельної і послідовної обмотки і зневазі вихровими струмами), тобто

$$T_{\Gamma} = T_{3\phi \Gamma} = \frac{2 p_{\Gamma} W_{3\phi} k_{\phi}}{k_t R_{3\phi}}, \quad (2.31)$$

де $W_{3\phi}$, $R_{3\phi}$ – число витків на полюс і активний опір, Ом, незалежної обмотки збудження;

p_{Γ} – число пар полюсів генератора;

k_{ϕ} – коефіцієнт нахилу кривої намагнічення для лінійної ділянки кривої намагнічення, можна прийняти $k_{\phi} = \Delta\Phi / \Delta I_{3\phi} = 0,29 \cdot 10^{-2} \text{ Ом} \cdot \text{с}$.

$$T_{\Gamma} = T_{3\phi \Gamma} = \frac{4 \cdot 930 \cdot 0,29 \cdot 10^{-2}}{1,36 \cdot 5,4} = 0,29 \text{ с}.$$

Електромеханічна постійна часу електроприводу

$$T_{\text{м}} = \frac{J_{\Sigma} R_{\text{я}\Sigma}}{(k \text{ с})^2}; \quad (2.32)$$

де J_{Σ} – сумарний момент інерції якоря двигуна і приведений до його валу момент інерції механізму. У розрахунках можна прийняти $J_{\Sigma} = 1,3 J_{\text{д}}$

$$T_{\text{м}} = \frac{1,3 \cdot 380 \cdot 0,03812}{(2 \cdot 8,02)^2} = 0,093 \text{ с}.$$

Електромагнітна постійна часу електроприводу

$$T_{\text{я}} = \frac{L_{\text{я}\Sigma}}{R_{\text{я}\Sigma}}, \quad (2.34)$$

де $L_{\text{я}\Sigma}$ – сумарна індуктивність якірного ланцюга, Гн

$$L_{\text{я}\Sigma} = k L_{\text{яд}} + L_{\text{яг}}, \quad (2.35)$$

$L_{\text{яд}}$, $L_{\text{яг}}$ – індуктивність двигуна і генератора відповідно; дані індуктивності можуть бути визначені по наближеній формулі, Гн:

$$L_{\text{я д}(\Gamma)} = k_k \frac{U_{\text{н д}(\Gamma)}}{p_{\text{д}(\Gamma)} \omega_{\text{н д}(\Gamma)} I_{\text{н д}(\Gamma)}}, \quad (2.36)$$

k_k – коефіцієнт, рівний $k_k = 0,25$ для компенсованої машини і $k_k = 0,6$ для некомпенсованої;

$U_{\text{н д}(\Gamma)}$, $I_{\text{н д}(\Gamma)}$, $\omega_{\text{н д}(\Gamma)}$ – номінальні величини напруги, B , струму, A , і швидкості, c^{-1} , двигуна і генератора відповідно;

$p_{\text{д}(\Gamma)}$ – число пар полюсів двигуна і генератора.

Номінальна швидкість генератора:

$$\omega_{\text{н Г}} = \frac{\pi n_{\text{н Г}}}{30}, \quad (2.37)$$

$$\omega_{\text{н Г}} = \frac{3,14 \cdot 1000}{30} = 104 \text{ } c^{-1}.$$

$$L_{\text{я д}} = \frac{0,25 \cdot 305}{2 \cdot 78,5 \cdot 360} = 0,001349 \text{ Гн}.$$

$$L_{\text{я Г}} = \frac{0,6 \cdot 630}{2 \cdot 104 \cdot 500} = 0,003635 \text{ Гн}.$$

$$L_{\text{я}\Sigma} = 2 \cdot 0,001349 + 0,003635 = 0,006333 \text{ Гн}.$$

$$T_{\text{я}} = \frac{0,006333}{0,1028} = 0,06 \text{ с}.$$

Постійну часу зворотного зв'язку по току $T_{\text{ззГ}}$ можна визначити таким чином: використовуючи критерій Гурвіца, знайти мінімальне значення $T_{\text{ззГ min}}$, при якому система стійка. Потім провести розрахунок перехідних характеристик для значень $T_{\text{ззГ}} \geq T_{\text{ззГ min}}$ і вибрати те значення, при якому перехідний процес оп-

тимальний. Якщо система стійка при будь-якому значенні $T_{ззт}$, то слід задатися рядом значень $T_{ззт}$ і вибрати оптимальне значення.

Для дослідження динамічних характеристик отримаємо вирази для передавальних функцій системи по задаючій і збурюючій діях. Для отримання передавальних функцій системи проведемо перетворення алгоритмічної схеми рис. 2.5. Для спрощення розрахунків без істотної погрішності можна виключити з розгляду постійні часу напівпровідникових суматорів AW і тиристорних збудників UL , постійні часу яких в 50 разів менше T_r . За цієї умови об'єднаємо передавальні функції вказаних елементів

$$W_1(p) = \frac{k_{AW}}{T_{AW}p + 1} \cdot \frac{k_{UL}}{T_{UL}p + 1} \cdot \frac{k_r}{T_r p + 1} = \frac{k_\Sigma}{T_r p + 1}. \quad (2.38)$$

Отримаємо алгоритмічну схему, показану на рис.2.6. Користуючись спрощеною алгоритмічною схемою, рис. 2.8, можна отримати передавальні функції САУ для швидкості і струму по задаючій і збурюючій діях. Для спрощення подальших перетворень всі ланки пронумеровані від 1 до 6.

Перемикаємо зворотний зв'язок по ЕДС двигуна з точки a в точку b , включаючи при цьому в зворотний зв'язок ланку з передатною функцією

$$W_7(p) = \frac{1}{W_1(p)} = \frac{1}{\frac{k_\Sigma}{T_{тп}p + 1}} = \frac{T_{тп}p + 1}{k_\Sigma}. \quad (2.39)$$

Отримаємо алгоритмічну схему, показану на рис.2.7 із зворотними зв'язками, що не перехрещуються.

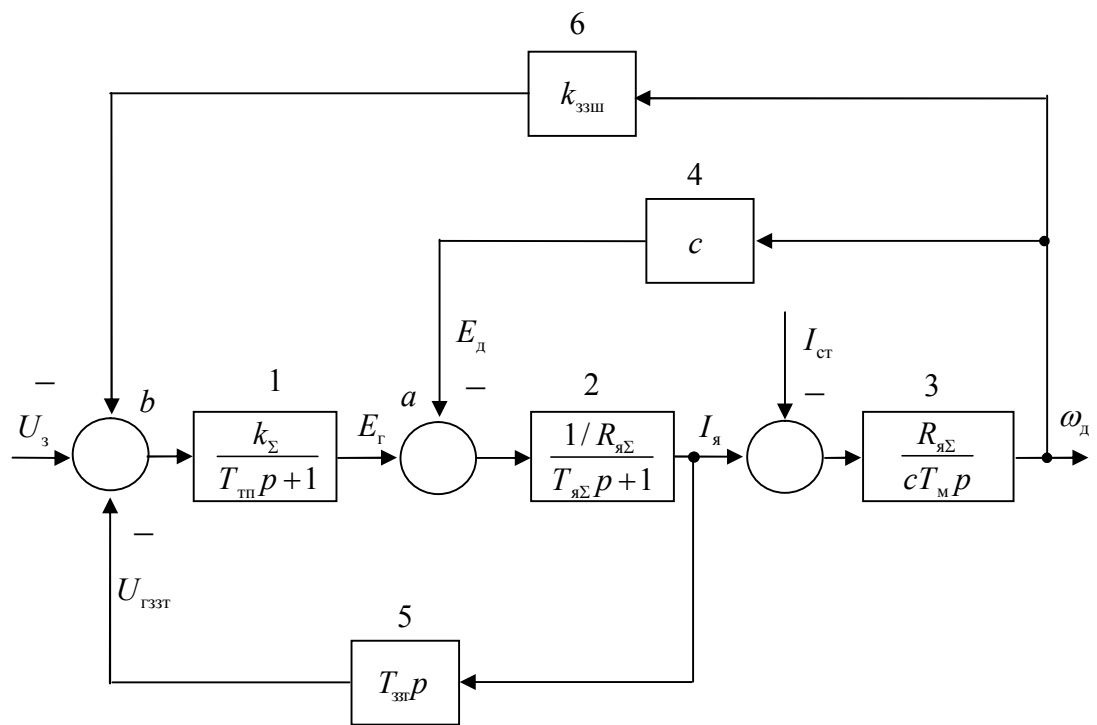


Рисунок 2.6 – Перетворена алгоритмічна схема системи,
показаної на рис. 2.7

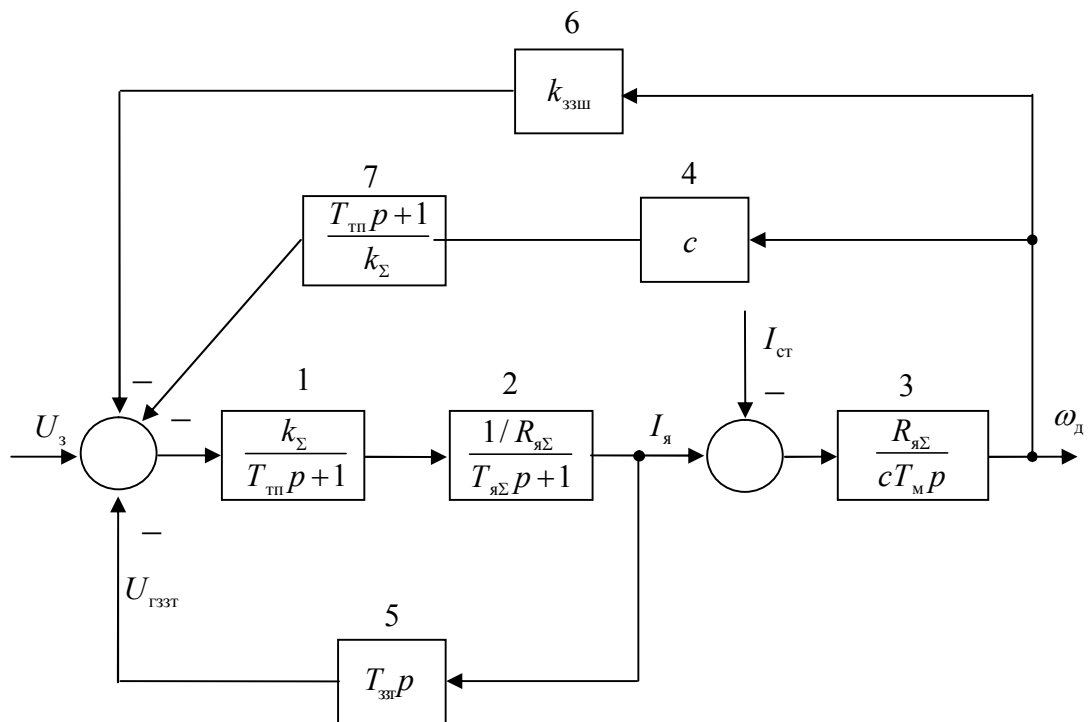


Рисунок 2.7 - Перетворена алгоритмічна схема системи,
показаної на рис. 2.8

Об'єднаємо ланки 1 і 2, а також 4 і 7

$$W_{1,2}(p) = W_1(p) \cdot W_2(p) = \frac{k_{\Sigma}}{T_{\text{тп}}p + 1} \cdot \frac{1/R_{\text{я}\Sigma}}{T_{\text{я}\Sigma}p + 1} = \frac{k_{\Sigma}/R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1)}, \quad (2.40)$$

$$W_{4,7}(p) = W_4(p) \cdot W_7(p) = c \cdot \frac{T_{\text{тп}}p + 1}{k_{\Sigma}} = \frac{c(T_{\text{тп}}p + 1)}{k_{\Sigma}}. \quad (2.41)$$

Одержимо схему, показану на рис.2.8.

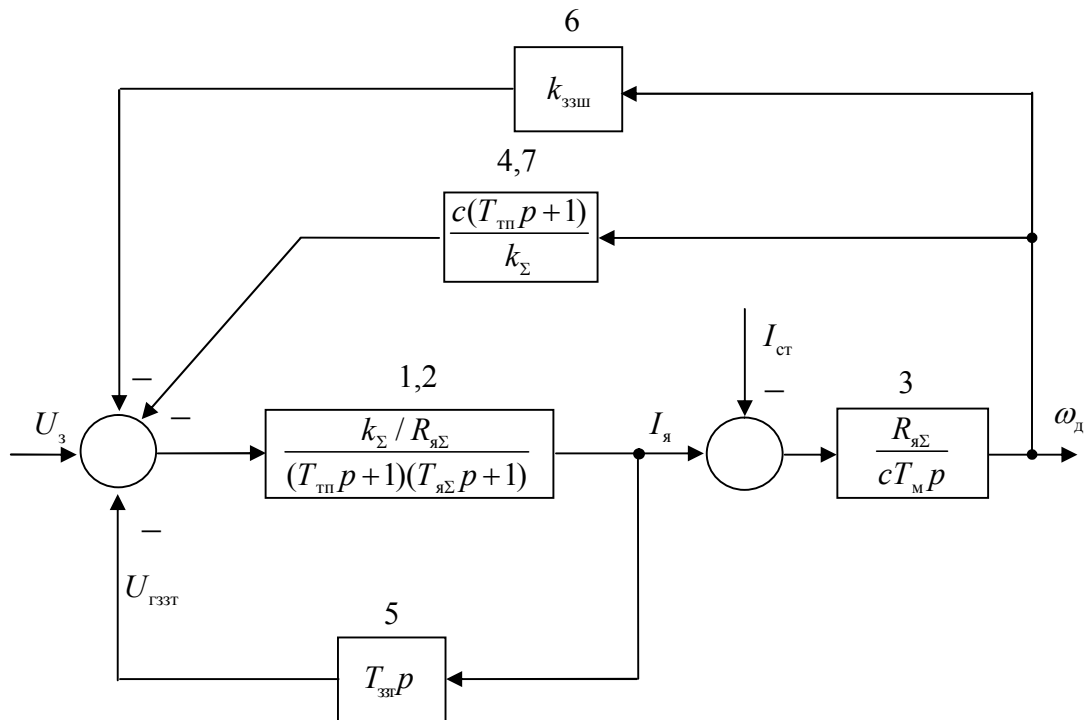


Рисунок 2.8- Перетворена алгоритмічна схема системи,
показаної на рис. 2.9

Об'єднаємо ланки 4,7 і 6 , а також виконаємо з'єднання ланок 1,2 і 5

$$W_{4,7,6}(p) = W_{4,7}(p) + W_6(p) = \frac{c(T_{\text{тп}}p + 1)}{k_{\Sigma}} + k_{33\text{ш}}. \quad (2.42)$$

$$W_{1,2,5}(p) = \frac{W_{1,2}(p)}{1 + W_{1,2}(p) \cdot W_5(p)} = \frac{\frac{k_{\Sigma}/R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1)}}{1 + \frac{k_{\Sigma}/R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1)} \cdot T_{3\text{тп}}p} =$$

$$= \frac{\frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1)}}{\frac{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1)}} = \frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}}. \quad (2.43)$$

Одержимо схему, показану на рис.2.9.

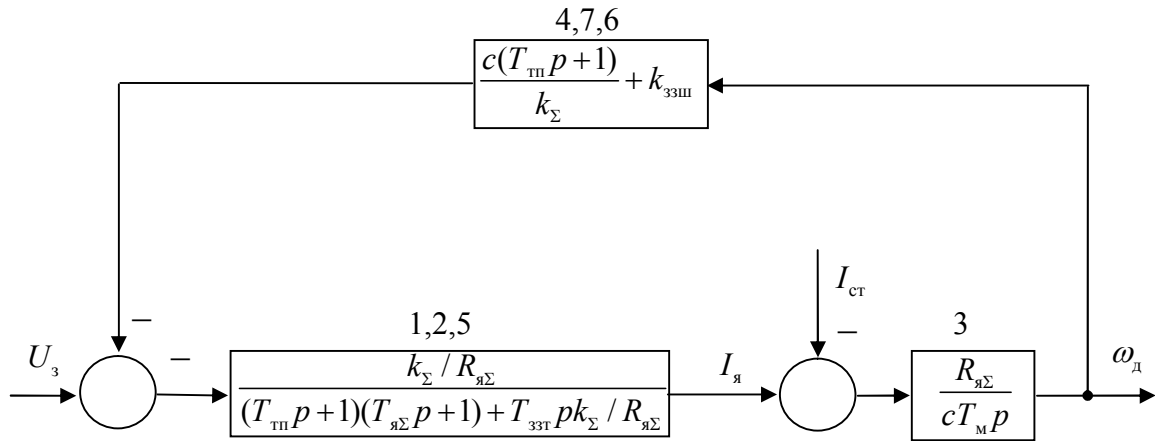


Рисунок 2.9- Перетворена алгоритмічна схема системи,
показаної на рис. 2.10

Використовуючи схему рис. 2.10 знайдемо передатні функції системи. При виведенні передавальних функцій по задаючій дії вважаємо $I_{\text{ст}} = 0$, а при виведенні передавальних функцій по збурюючій дії вважаємо $U_3 = 0$.

Передавальна функція для швидкості по задаючій дії.

$$W_{\omega, U_3}(p) = \frac{\omega(p)}{U_3(p)} = \frac{\frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}} \cdot \frac{R_{\Sigma}}{cT_{\text{м}} p}}{1 + \frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}} \cdot \frac{R_{\Sigma}}{cT_{\text{м}} p} \cdot \left[\frac{c(T_{\text{тп}} p + 1)}{k_{\Sigma}} + k_{\text{ззш}} \right]} =$$

$$= \frac{\frac{k_{\Sigma}}{[(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}] \cdot cT_{\text{м}} p}}{cT_{\text{м}} p \cdot k_{\Sigma} \cdot (T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + cT_{\text{м}} p \cdot k_{\Sigma} \cdot T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma} + c(T_{\text{тп}} p + 1) + k_{\Sigma}^2 \cdot k_{\text{ззш}}} =$$

$$\frac{k_{\Sigma}}{[(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\Sigma}] \cdot cT_{\text{м}} p \cdot k_{\Sigma}}$$

$$\begin{aligned}
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p \cdot R_{я\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p \cdot p \cdot k_{\Sigma} + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} \cdot cT_{\Sigma} p + R_{я\Sigma} \cdot c} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p \cdot R_{я\Sigma} (T_{\Sigma} p \cdot T_{я\Sigma} p + 1) + cT_{\Sigma} p \cdot T_{33\text{т}} p \cdot k_{\Sigma} + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} \cdot cT_{\Sigma} p + R_{я\Sigma} \cdot c} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p R_{я\Sigma} T_{\Sigma} p T_{я\Sigma} p + cT_{\Sigma} p R_{я\Sigma} T_{\Sigma} p + cT_{\Sigma} p R_{я\Sigma} T_{я\Sigma} p + cT_{\Sigma} p \cdot R_{я\Sigma} + cT_{\Sigma} p T_{33\text{т}} p k_{\Sigma} + \\
&\quad + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} cT_{\Sigma} p + R_{я\Sigma}} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cR_{я\Sigma} T_{\Sigma} T_{\Sigma} T_{я\Sigma} p^3 + c(R_{я\Sigma} T_{\Sigma} T_{\Sigma} + R_{я\Sigma} T_{\Sigma} T_{я\Sigma} + k_{\Sigma} T_{\Sigma} T_{\Sigma}) p^2 + cR_{я\Sigma} (T_{\Sigma} + T_{\Sigma}) p + R_{я\Sigma} (k_{\Sigma} k_{33\text{ш}} + c)} = \\
&= \frac{k_{\Sigma} k_{\Sigma}}{T_{\Sigma} T_{\Sigma} T_{я\Sigma} p^3 + \left(T_{\Sigma} T_{\Sigma} + T_{я\Sigma} T_{\Sigma} + \frac{k_{\Sigma} T_{\Sigma} T_{33\text{т}}}{R_{я\Sigma}} \right) p^2 + (T_{\Sigma} + T_{\Sigma}) p + 1 + k_{\Sigma} k_{\Sigma} k_{33\text{ш}}}. \quad (2.44)
\end{aligned}$$

При виведенні передавальної функції прийнято позначення $k_{\Sigma} = \frac{1}{c}$ – коефіцієнт посилення еквівалентного двигуна.

Передавальна функція для швидкості по збурюючій дії.

$$\begin{aligned}
W_{\omega, I_{\Sigma}}(p) &= \frac{\omega(p)}{I_{\Sigma}(p)} = \frac{\frac{R_{я\Sigma}}{cT_{\Sigma} p}}{1 + \frac{R_{я\Sigma}}{cT_{\Sigma} p} \cdot \frac{k_{\Sigma} / R_{я\Sigma}}{(T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + T_{33\text{т}} p k_{\Sigma} / R_{я\Sigma}} \cdot \left[\frac{c(T_{\Sigma} p + 1)}{k_{\Sigma}} + k_{33\text{ш}} \right]} = \\
&= \frac{\frac{R_{я\Sigma}}{cT_{\Sigma} p}}{\frac{cT_{\Sigma} p R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p + R_{я\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{я\Sigma} k_{\Sigma} cT_{\Sigma} p + R_{я\Sigma} k_{\Sigma} c}{cT_{\Sigma} p \cdot R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p}} = \\
&= \frac{R_{я\Sigma}}{cT_{\Sigma} p} \cdot \frac{cT_{\Sigma} p \cdot R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p}{cT_{\Sigma} p \cdot cT_{\Sigma} p R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p + R_{я\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{я\Sigma} k_{\Sigma} cT_{\Sigma} p + R_{я\Sigma} k_{\Sigma} c} = \\
&= \frac{R_{я\Sigma}^2 k_{\Sigma} (T_{\Sigma} p R_{я\Sigma} p + T_{\Sigma} p + T_{я\Sigma} p + 1) + k_{\Sigma}^2 T_{33\text{т}} p}{cT_{\Sigma} p R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p T_{я\Sigma} p + T_{\Sigma} p + T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p + R_{я\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{я\Sigma} k_{\Sigma} cT_{\Sigma} p + R_{я\Sigma} k_{\Sigma} c} =
\end{aligned}$$

$$\begin{aligned}
&= \frac{R_{\Sigma}^2 k_{\Sigma} T_{\Sigma} p + T_{\Sigma} p + R_{\Sigma}^2 k_{\Sigma} T_{\Sigma} p + R_{\Sigma}^2 k_{\Sigma} T_{\Sigma} p + R_{\Sigma}^2 k_{\Sigma} + k_{\Sigma}^2 T_{\Sigma} p}{c T_{\Sigma} p R_{\Sigma} k_{\Sigma} T_{\Sigma} p^3 + c k_{\Sigma} (R_{\Sigma} T_{\Sigma} T_{\Sigma} + T_{\Sigma} T_{\Sigma} R_{\Sigma} + k_{\Sigma} T_{\Sigma} T_{\Sigma}) p^2 + c R_{\Sigma} k_{\Sigma} (T_{\Sigma} + T_{\Sigma}) p +} \\
&\quad + R_{\Sigma} k_{\Sigma} (k_{\Sigma} k_{\Sigma} + c) \\
&= \frac{R_{\Sigma}^2 T_{\Sigma} T_{\Sigma} p^3 + (R_{\Sigma}^2 T_{\Sigma} + R_{\Sigma}^2 T_{\Sigma} + k_{\Sigma} T_{\Sigma}) p + R_{\Sigma}^2}{c R_{\Sigma} T_{\Sigma} T_{\Sigma} T_{\Sigma} p^3 + c (R_{\Sigma} T_{\Sigma} T_{\Sigma} + T_{\Sigma} R_{\Sigma} + k_{\Sigma} T_{\Sigma} T_{\Sigma}) p^2 + c R_{\Sigma} (T_{\Sigma} + T_{\Sigma}) p + R_{\Sigma} (k_{\Sigma} k_{\Sigma} + c)} \\
&= \frac{(R_{\Sigma} T_{\Sigma} T_{\Sigma} p^2 + (R_{\Sigma} T_{\Sigma} + R_{\Sigma} T_{\Sigma} + k_{\Sigma} T_{\Sigma}) p + R_{\Sigma}) k_{\Sigma}}{T_{\Sigma} T_{\Sigma} T_{\Sigma} p^3 + \left(T_{\Sigma} T_{\Sigma} + T_{\Sigma} T_{\Sigma} + \frac{k_{\Sigma} T_{\Sigma} T_{\Sigma}}{R_{\Sigma}} \right) p^2 + (T_{\Sigma} + T_{\Sigma}) p + 1 + k_{\Sigma} k_{\Sigma} k_{\Sigma}}. \quad (2.45)
\end{aligned}$$

Передавальна функція для струму по задаючій дії.

$$\begin{aligned}
W_{I_{\Sigma}, U_{\Sigma}}(p) &= \frac{I_{\Sigma}(p)}{U_{\Sigma}} = \frac{\frac{k_{\Sigma} / R_{\Sigma}}{(T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + T_{\Sigma} p k_{\Sigma} / R_{\Sigma}}}{1 + \frac{k_{\Sigma} / R_{\Sigma}}{(T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + T_{\Sigma} p k_{\Sigma} / R_{\Sigma}} \cdot \left[\frac{c(T_{\Sigma} p + 1)}{k_{\Sigma}} + k_{\Sigma} \right] \cdot \frac{R_{\Sigma}}{c T_{\Sigma} p}} = \\
&= \frac{\frac{k_{\Sigma}}{(T_{\Sigma} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{\Sigma} p k_{\Sigma}}}{\frac{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p + R_{\Sigma} k_{\Sigma}^2 k_{\Sigma} + R_{\Sigma} k_{\Sigma} c T_{\Sigma} p + R_{\Sigma} k_{\Sigma} c}{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p}} = \\
&= \frac{k_{\Sigma}}{(T_{\Sigma} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{\Sigma} p k_{\Sigma}} \cdot \frac{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p}{k_{\Sigma} \left[R_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma} c T_{\Sigma} p T_{\Sigma} p + R_{\Sigma} k_{\Sigma} k_{\Sigma} + R_{\Sigma} c T_{\Sigma} p + R_{\Sigma} c \right]} = \\
W_{I_{\Sigma}, U_{\Sigma}}(p) &= \frac{I_{\Sigma}(p)}{U_{\Sigma}} = \frac{\frac{k_{\Sigma} / R_{\Sigma}}{(T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + T_{\Sigma} p k_{\Sigma} / R_{\Sigma}}}{1 + \frac{k_{\Sigma} / R_{\Sigma}}{(T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + T_{\Sigma} p k_{\Sigma} / R_{\Sigma}} \cdot \left[\frac{c(T_{\Sigma} p + 1)}{k_{\Sigma}} + k_{\Sigma} \right] \cdot \frac{R_{\Sigma}}{c T_{\Sigma} p}} = \\
&= \frac{\frac{k_{\Sigma}}{(T_{\Sigma} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{\Sigma} p k_{\Sigma}}}{\frac{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p + R_{\Sigma} k_{\Sigma}^2 k_{\Sigma} + R_{\Sigma} k_{\Sigma} c T_{\Sigma} p + R_{\Sigma} k_{\Sigma} c}{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p}} = \\
&= \frac{k_{\Sigma}}{(T_{\Sigma} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{\Sigma} p k_{\Sigma}} \cdot \frac{R_{\Sigma} k_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\Sigma} p T_{\Sigma} p}{k_{\Sigma} \left[R_{\Sigma} c T_{\Sigma} p (T_{\Sigma} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma} c T_{\Sigma} p T_{\Sigma} p + R_{\Sigma} k_{\Sigma} k_{\Sigma} + R_{\Sigma} c T_{\Sigma} p + R_{\Sigma} c \right]} =
\end{aligned}$$

$$= \frac{k_{\Sigma} T_{\text{м}} p / R_{\text{я}\Sigma}}{T_{\text{тп}} T_{\text{м}} T_{\text{я}\Sigma} p^3 + \left(T_{\text{тп}} T_{\text{м}} + T_{\text{я}\Sigma} T_{\text{м}} + \frac{k_{\Sigma} T_{\text{м}} T_{\text{ззт}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\text{г}} + T_{\text{м}}) p + 1 + k_{\Sigma} k_{\text{д}} k_{\text{ззш}}}. \quad (2.46)$$

Запишемо передавальну функцію для струму по збуджуючій дії, заздалегідь перетворивши алгоритмічну схему, показану на рис.2.9, до вигляду, показаному на рис. 2.10.

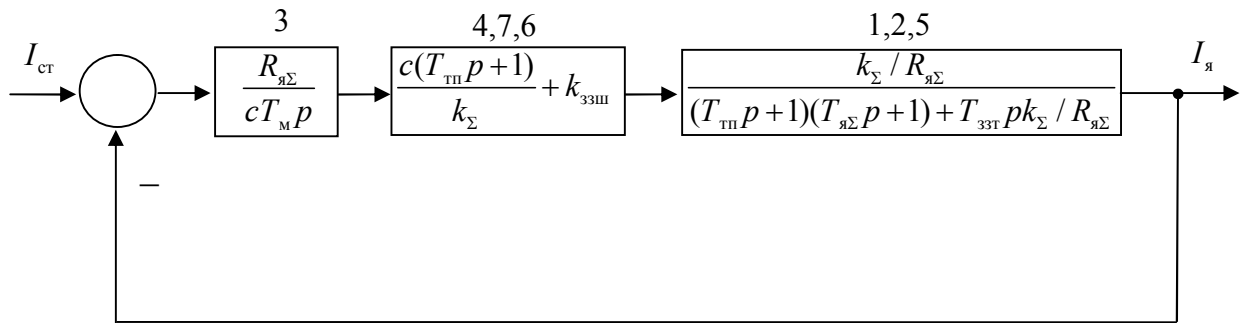


Рисунок 2.10- Перетворена алгоритмічна схема рис. 2.11 для визначення

$$\text{передавальної функції } W_4(p) = \frac{I(p)}{I_{\text{ст}}(p)}$$

$$\begin{aligned} W_{I_{\text{я}}, I_{\text{ст}}}(p) &= \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)} = \frac{\frac{R_{\text{я}\Sigma}}{c T_{\text{м}} p} \cdot \left[\frac{c(T_{\text{тп}} p + 1)}{k_{\Sigma}} + k_{\text{ззш}} \right] \cdot \frac{k_{\Sigma} / R_{\text{я}\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\text{я}\Sigma}}}{1 + \frac{R_{\text{я}\Sigma}}{c T_{\text{м}} p} \cdot \left[\frac{c(T_{\text{тп}} p + 1)}{k_{\Sigma}} + k_{\text{ззш}} \right] \cdot \frac{k_{\Sigma} / R_{\text{я}\Sigma}}{(T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) + T_{\text{ззт}} p k_{\Sigma} / R_{\text{я}\Sigma}}} = \\ &= \frac{R_{\text{я}\Sigma} k_{\Sigma} k_{\text{ззш}} + R_{\text{я}\Sigma} c T_{\text{тп}} p + R_{\text{я}\Sigma} c}{c T_{\text{м}} p (T_{\text{тп}} p + 1)(T_{\text{я}\Sigma} p + 1) R_{\text{я}\Sigma} + c T_{\text{м}} T_{\text{ззт}} p^2 \cdot k_{\Sigma} + R_{\text{я}\Sigma} k_{\Sigma} k_{\text{ззш}} + R_{\text{я}\Sigma} c T_{\text{тп}} p + R_{\text{я}\Sigma} c} = \\ &= \frac{R_{\text{я}\Sigma} (k_{\Sigma} k_{\text{ззш}} + c T_{\text{тп}} p + c)}{c T_{\text{м}} p (T_{\text{тп}} T_{\text{я}\Sigma} p^2 + T_{\text{я}\Sigma} p + T_{\text{тп}} p + 1) R_{\text{я}\Sigma} + c T_{\text{м}} T_{\text{ззт}} p^2 \cdot k_{\Sigma} + R_{\text{я}\Sigma} c T_{\text{тп}} p + R_{\text{я}\Sigma} (k_{\Sigma} k_{\text{ззш}} + c)} = \\ &= \frac{R_{\text{я}\Sigma} (k_{\Sigma} k_{\text{ззш}} + c T_{\text{тп}} p + c)}{R_{\text{я}\Sigma} c T_{\text{м}} T_{\text{тп}} T_{\text{я}\Sigma} p^3 + c (R_{\text{я}\Sigma} T_{\text{м}} T_{\text{я}\Sigma} + R_{\text{я}\Sigma} T_{\text{м}} T_{\text{тп}} + k_{\Sigma} T_{\text{м}} T_{\text{ззт}}) p^2 + R_{\text{я}\Sigma} c (T_{\text{м}} + T_{\text{тп}}) p + R_{\text{я}\Sigma} (k_{\Sigma} k_{\text{ззш}} + c)} = \end{aligned}$$

$$= \frac{T_{\text{тп}} p + 1 + k_{\Sigma} k_{\text{ззш}} k_{\text{д}}}{T_{\text{тп}} T_{\text{м}} T_{\text{я}\Sigma} p^3 + \left(T_{\text{тп}} T_{\text{м}} + T_{\text{я}\Sigma} T_{\text{м}} + \frac{k_{\Sigma} T_{\text{м}} T_{\text{ззт}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\text{г}} + T_{\text{м}}) p + 1 + k_{\Sigma} k_{\text{д}} k_{\text{ззш}}} . \quad (2.47)$$

Використовуючи отримані передавальні функції були розраховані графіки перехідних процесів, приведені на рис.2.12 і рис.2.13. Розрахунки проводилися по ЕОМ з використанням пакету прикладних програм MATLAB.

Крива 1 відповідає $T_{\text{ззт}} = 0,003 \text{ с}$, крива 2 - $T_{\text{ззт}} = 0,01 \text{ с}$, крива 3 - $T_{\text{ззт}} = 0,2 \text{ с}$. Як видно з графіків, оптимальним є значення $T_{\text{ззт}} = 0,005 \text{ с}$, при якому перехідні процеси мають кращі показники якості.

Для отримання графіків перехідних процесів використовувалася також модель системи, розроблена за допомогою пакету MATLAB в режимі SIMULINK. Сама модель показана на рисунку 2.11. Графіки, побудовані за допомогою моделі, повністю співпадають з графіками, отриманими по передавальних функціях, що свідчить про правильність отриманих результатів.

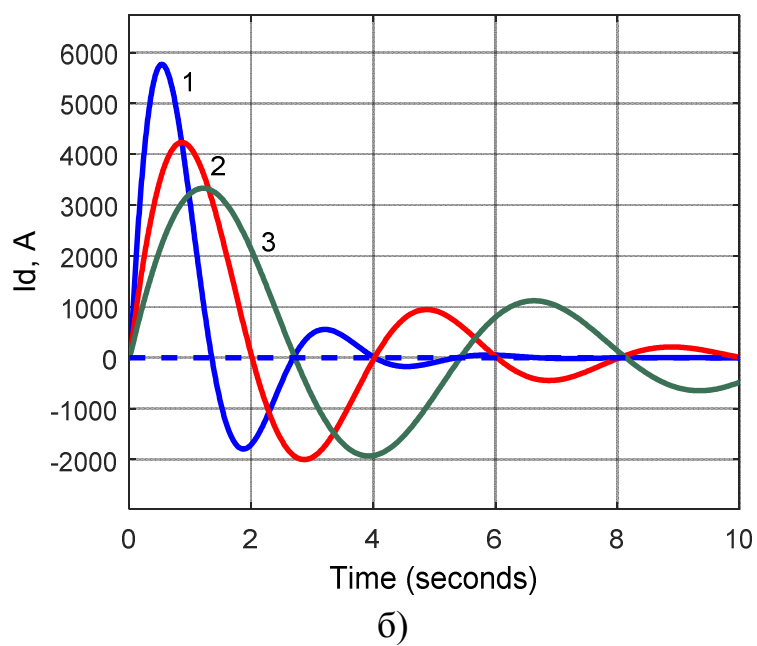
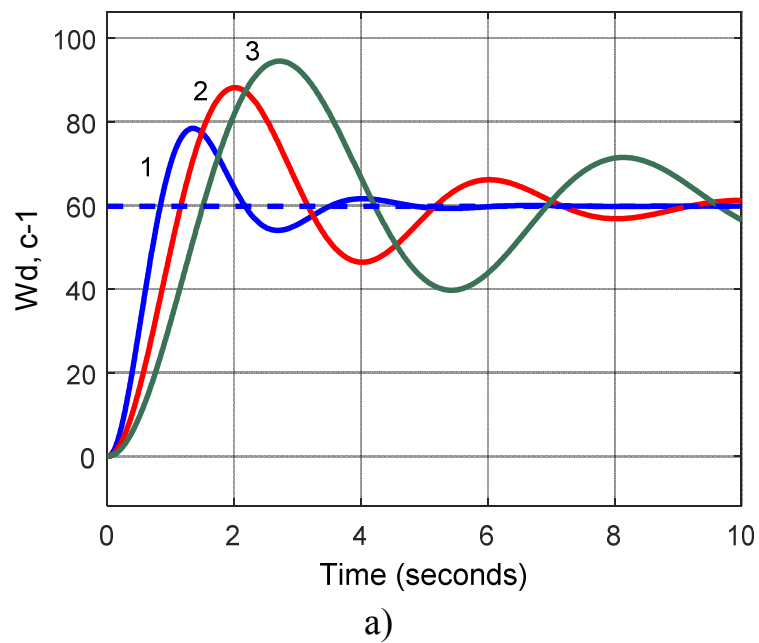
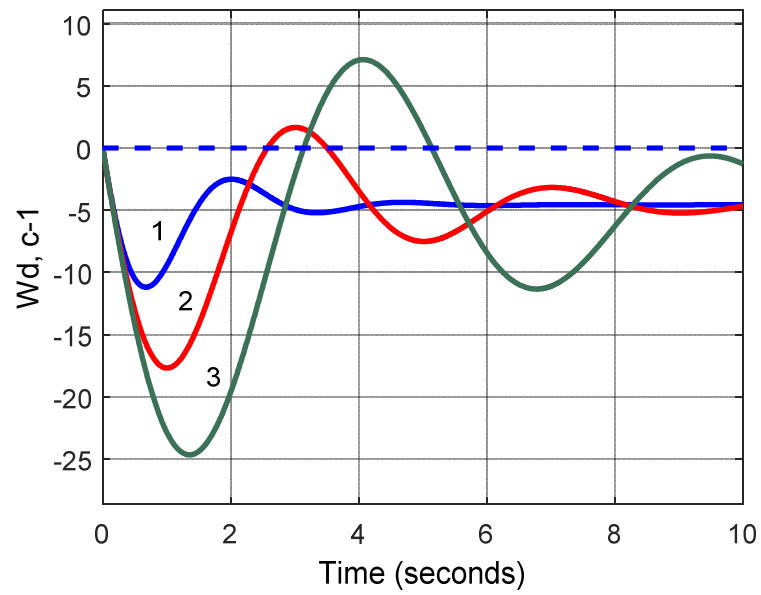
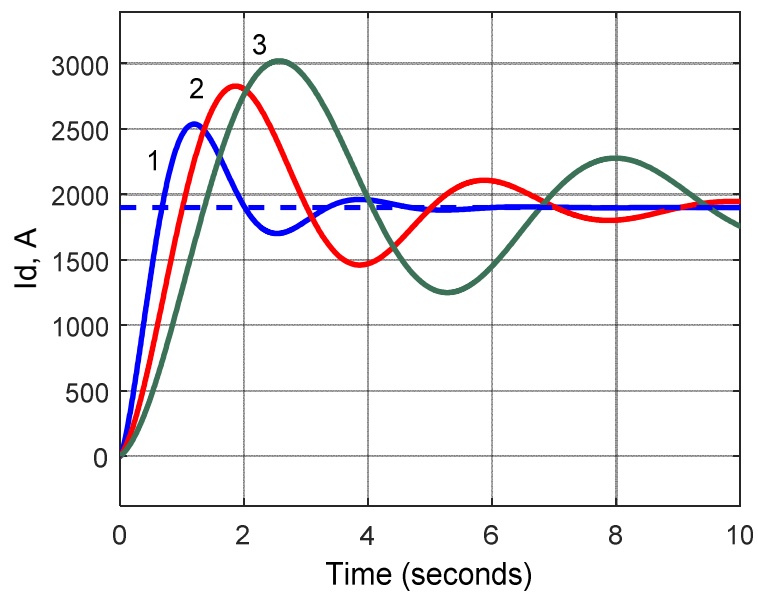


Рисунок 2.12– Графіки перехідних процесів швидкості а) і струму б)
по задаючій дії:

$$1 - T_{33T} = 0,005 \text{ c}; \quad 2 - T_{33T} = 0,015 \text{ c}; \quad 3 - T_{33T} = 0,03 \text{ c}$$



a)



б)

Рисунок 2.13 – Графіки перехідних процесів швидкості а) і струму б)

по збурюючій дії:

$$1 - T_{\text{ззт}} = 0,005 \text{ c}; \quad 2 - T_{\text{ззт}} = 0,015 \text{ c}; \quad 3 - T_{\text{ззт}} = 0,03 \text{ c}$$

ВИСНОВКИ ДО РОЗДІЛУ 2

1. Виконано вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу.
2. Виконано розрахунок параметрів силового ланцюга електроприводу, що використовуються при розрахунку системи автоматичного управління.
3. Проведено вибір структури системи автоматичного управління. До електроприводів більшості основних механізмів роторних екскаваторів пред'являються підвищені вимоги відносно жорсткості робочої частини механічної характеристики і обмеження максимального значення моменту, що розвивається.. Згідно рекомендацій [28] вибрано систему із зворотним зв'язком за швидкістю, гнучким зворотним зв'язком по струму і зворотним зв'язком по струму з відсіченням.
4. Проведений розрахунок статичних характеристик розімкненої і замкненої системи. Складено алгоритмічну схему системи та визначені передавальні функції замкненої системи по задаючій і збурюючій діях.
5. Проведено моделювання синтезованої системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблена схема моделі системи, що дозволяє виконати дослідження динамічних характеристик системи. Шляхом моделювання визначено значення постійної часу зворотного зв'язку по струму, при якій перехідні процеси змінних стану системи мають задовільні показники якості.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

У попередніх розділах механічна частина електроприводу представлялася жорсткою приведеною ланкою, що не враховує впливу пружності реальних механічних зв'язків. Із-за кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, прикладення до якої дій, що управляють (моменту двигуна) або збурюючого (навантаження), викликають коливання в'язаних мас, що збільшує максимальні навантаження зв'язків і ускладнює точність відпрацювання необхідних рухів робочого органу машини.

Якщо виділити основні маси, які визначають рух системи, і найменші жорсткості зв'язків між ними, то кінематична схема механізму обертання роторного колеса може бути приведена до двомасової системи (рис. 3.1).

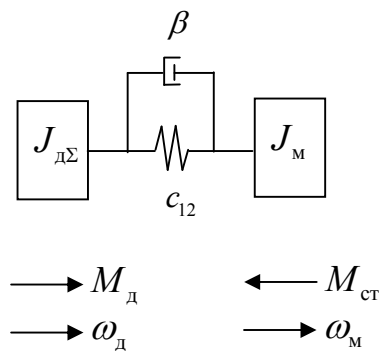


Рисунок 3.1– Двомасова розрахункова схема

Параметрами двомасової системи є сумарний момент інерції якоря двигуна і жорстко пов'язаних з ним елементів $J_{дΣ}$, сумарний приведений до швидкості двигуна момент інерції роторного колеса (механізму) $J_{м}$ і безінерційний пружний зв'язок між ними, що характеризується приведеною еквівалентною жорсткістю c_{12} і коефіцієнтом внутрішнього в'язкого тертя між масами β .

Якщо привести всі рухомі маси і жорсткості електроприводу механізму обертання колеса роторного екскаватора до кутової швидкості двигуна, то для двомасової системи отримаємо наступну систему диференціальних рівнянь:

$$\begin{cases} J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma} - M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_{\text{д}} - \omega_{\text{м}}); \\ J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = k\Phi_{\text{н}} I_{\text{я}} - M_{\Sigma}, \end{cases} \quad (3.1)$$

де M_{Σ} сумарний момент, що передається пружною передачею, який рівний сумі пружного моменту $M_{\text{пр}}$ і моменту в'язкого тертя $M_{\text{вт}} = \beta(\omega_{\text{д}} - \omega_{\text{м}})$;

$$M_{\Sigma} = M_{\text{пр}} + \beta(\omega_{\text{д}} - \omega_{\text{м}}), \quad (3.2)$$

β – коефіцієнт внутрішнього в'язкого тертя між масами;

$\omega_{\text{д}}$, $\omega_{\text{м}}$ - кутова швидкість двигуна і механізму відповідно;

$M_{\text{ст}}$ - момент статичного навантаження, прикладений до другої маси;

Запишемо систему (3.1) з урахуванням (3.2) у формі Коші

$$\begin{cases} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}. \end{cases} \quad (3.3)$$

Диференціальні рівняння, що описують електромеханічні процеси в розробленій в попередніх розділах одномасовій системі мають наступну форму

$$\left\{ \begin{array}{l} T_{\text{м}} \frac{d\omega_{\text{д}}}{dt} = \frac{R_{\text{я}\Sigma}}{c} (I_{\text{я}} - I_{\text{ст}}); \\ T_{\text{я}\Sigma} \frac{dI_{\text{я}}}{dt} + I_{\text{я}} = \frac{1}{R_{\text{я}\Sigma}} (E_{\Gamma} - c\omega_{\text{д}}); \\ T_{\Gamma} \frac{dE_{\Gamma}}{dt} + E_{\Gamma} = k_{\Sigma} U_3 - k_{33\text{ш}} k_{\Sigma} \omega_{\text{д}} - k_{\Sigma} U_{\Gamma 33\Gamma}; \\ T_{33\Gamma} \frac{dI_{\text{я}}}{dt} = U_{\Gamma 33\Gamma}. \end{array} \right. \quad (3.4)$$

Запишемо систему (3.4) у формі Коші, враховуючи, що $I_{\text{ст}} = M_{\text{ст}}/c$, $T_{\text{м}} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}$. Об'єднаємо системи (3.3) і (3.4) і проведемо деякі нескладні перетворення. Отримаємо наступну систему рівнянь стану двомасової системи:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = +\frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{c}{J_{\text{д}\Sigma}} I_{\text{я}}; \\ \frac{dM_{\text{пр}}}{dt} = -c_{12} \omega_{\text{м}} + c_{12} \omega_{\text{д}}; \\ \frac{d\omega_{\text{м}}}{dt} = -\frac{\beta}{J_{\text{м}}} \omega_{\text{м}} + \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dI_{\text{я}}}{dt} = -\frac{c}{T_{\text{я}\Sigma} R_{\text{я}\Sigma}} \omega_{\text{д}} - \frac{1}{T_{\text{я}\Sigma}} I_{\text{я}} + \frac{1}{T_{\text{я}\Sigma} R_{\text{я}\Sigma}} E_{\Gamma}; \\ \frac{dE_{\Gamma}}{dt} = \left(\frac{ck_{\Sigma} T_{33\Gamma}}{T_{\text{я}\Sigma} R_{\text{я}\Sigma} T_{\Gamma}} - \frac{k_{33\text{ш}} k_{\Sigma}}{T_{\Gamma}} \right) \omega_{\text{д}} + \frac{k_{\Sigma} T_{33\Gamma}}{T_{\text{я}\Sigma} T_{\Gamma}} I_{\text{я}} + \left(-\frac{k_{\Sigma} T_{33\Gamma}}{T_{\text{я}\Sigma} R_{\text{я}\Sigma} T_{\Gamma}} - \frac{1}{T_{\Gamma}} \right) E_{\Gamma} + \frac{k_{\Sigma}}{T_{\Gamma}} U_3 \end{array} \right. \quad (3.5)$$

Введемо вектор стану системи

$$\vec{X}(t) = [\omega_{\text{м}}(t); M_{\text{пр}}(t); \omega_{\text{д}}(t); I_{\text{я}}(t); E_{\Gamma}(t)]^T \quad (3.6)$$

і вектор зовнішніх дій

$$\vec{U}(t) = [U_3(t); M_{\text{ст}}(t)]^T. \quad (3.7)$$

Система, записана у векторний-матричній формі і доповнена рівнянням виходу, має вид:

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.8)$$

де $\mathbf{A}$ - матриця стану системи;

$\mathbf{B}$ - матриця управління;

$\mathbf{C}$ - матриця виходу;

$\vec{Y}(t)$ - вектор виходу.

Матриці стану $\mathbf{A}$ і управління $\mathbf{B}$ мають структуру:

$$\mathbf{A} = \begin{array}{ccccc} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & I_{\text{я}} & E_{\text{г}} \\ \hline -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 \\ -c_{12} & 0 & c_{12} & 0 & 0 \\ \frac{\beta}{J_{\text{м}}} & -\frac{1}{J_{\text{м}}} & -\frac{\beta}{J_{\text{м}}} & \frac{c}{J_{\text{м}}} & 0 \\ 0 & 0 & -\frac{c}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}} & -\frac{1}{T_{\text{я}\Sigma}} & \frac{1}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}} \\ 0 & 0 & \frac{ck_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\Gamma}} - \frac{k_{33\text{ш}}k_{\Sigma}}{T_{\Gamma}} & \frac{k_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}T_{\Gamma}} & \frac{-k_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\Gamma}} - \frac{1}{T_{\Gamma}} \end{array} \quad \mathbf{B} = \begin{array}{cc} U_3 & M_{\text{ст}} \\ \hline 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_{\Sigma}}{T_{\Gamma}} & 0 \end{array}$$

Алгоритмічна схема двомасової системи показана на рис.3.2.

3.2 Розрахунок перехідних процесів двомасової системи

Використовуючи математичну модель системи у вигляді рівнянь стану були проведені розрахунки перехідних процесів по задаючій і збурюючій діях. Текст файлу, що містить початкові дані, матриці **A**, **B**, **C** приведений в додатку Г. Дослідження проводилися також за допомогою схеми моделі системи, приведеної на рис. 3.3. Всі розрахунки виконувалися на ЕОМ за допомогою пакету прикладних програм MATLAB.

Графіки перехідних процесів показані на рис. 3.4 – рис. 3.7. На рисунках приведені перші чотири змінні стани системи. Як видно з графіків, мають місце значні коливання основних координат системи, що несприятливо позначається на роботі механізму, збільшуючи навантаження на елементи системи в перехідних режимах.

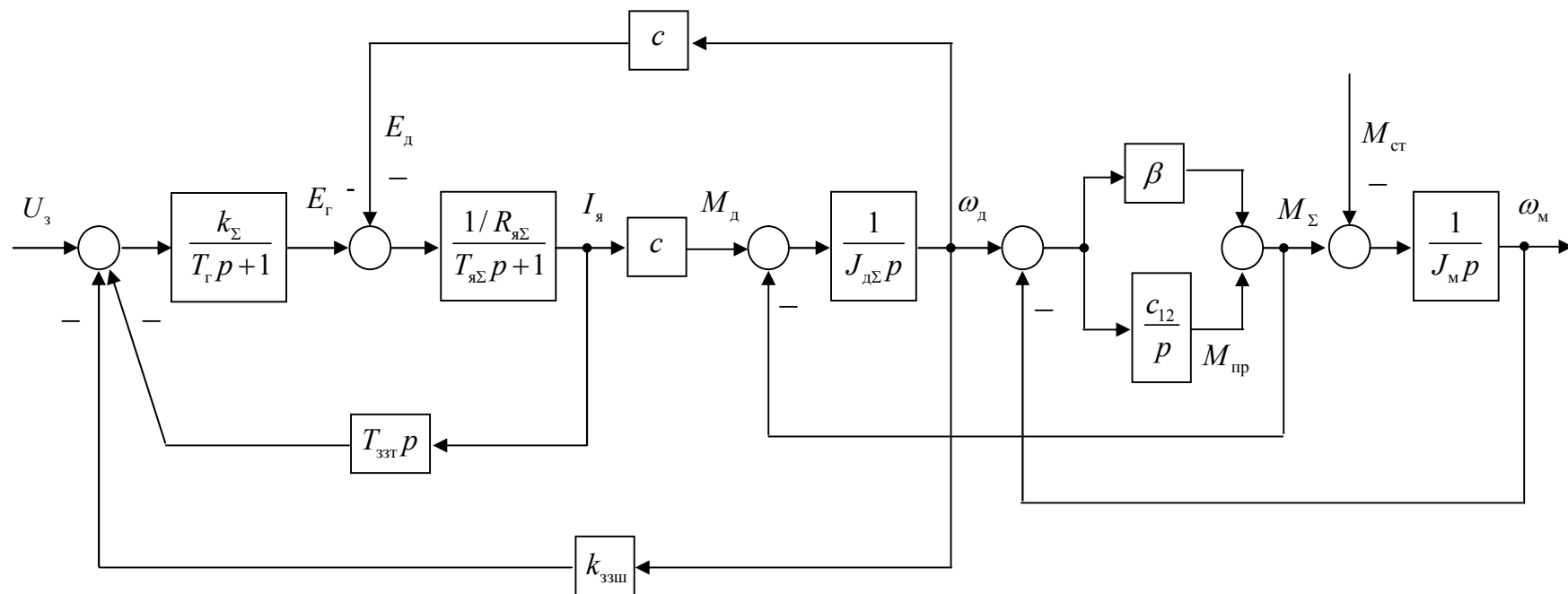


Рисунок 3.2 - Алгоритмічна схема двомасової системи.

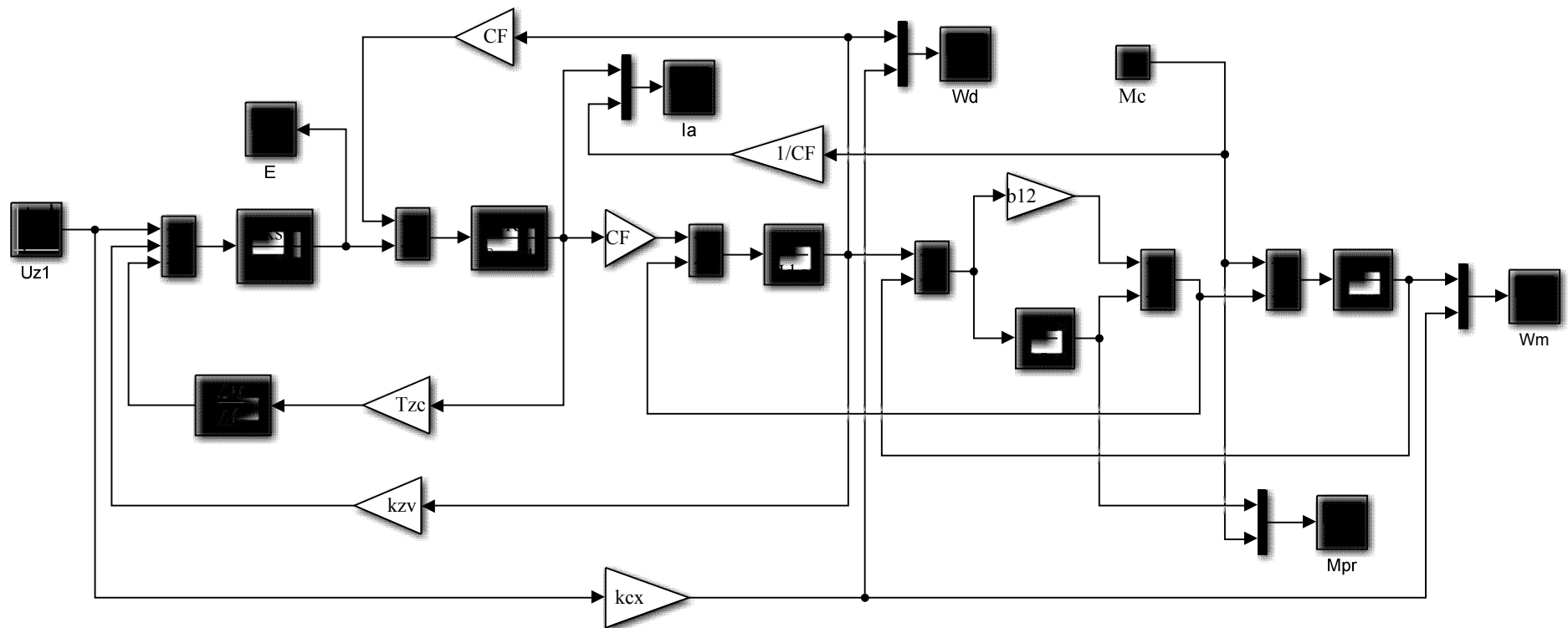
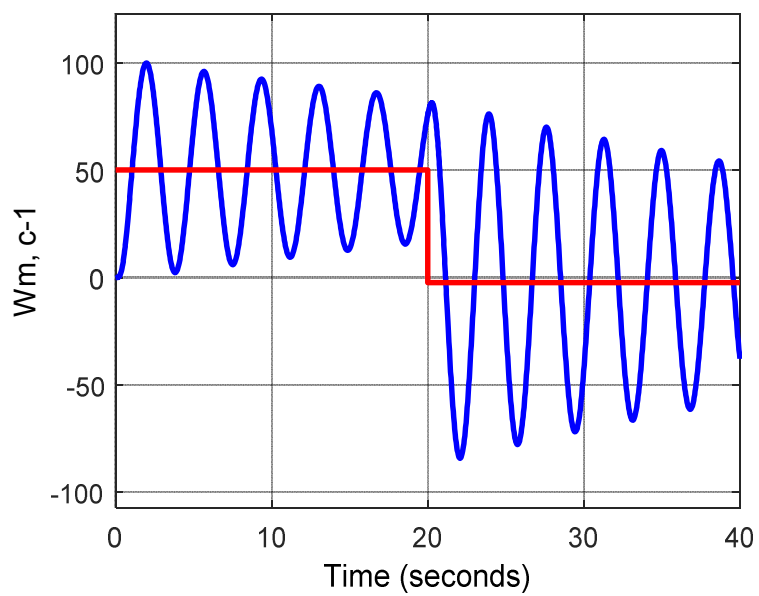
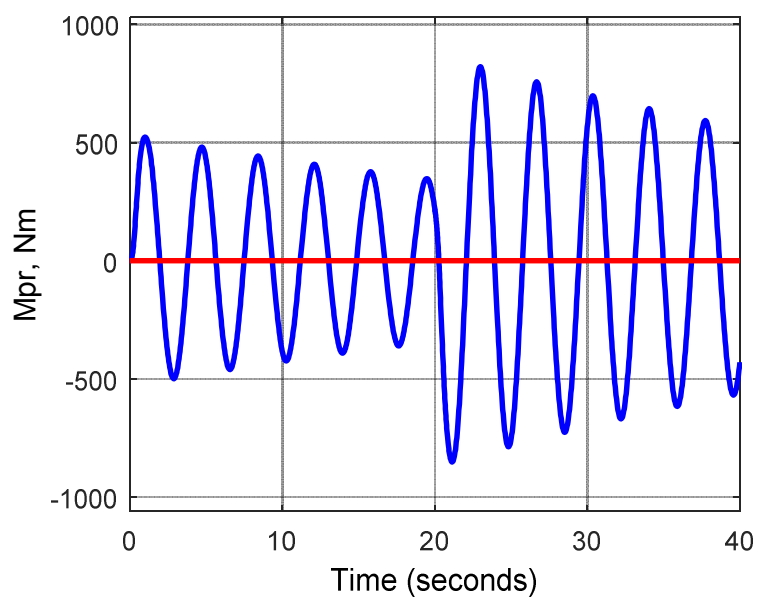


Рисунок 3.3 - Схема моделі двомасової системи, розроблена на ЕОМ



a)



б)

Рисунок 3.4 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (a) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

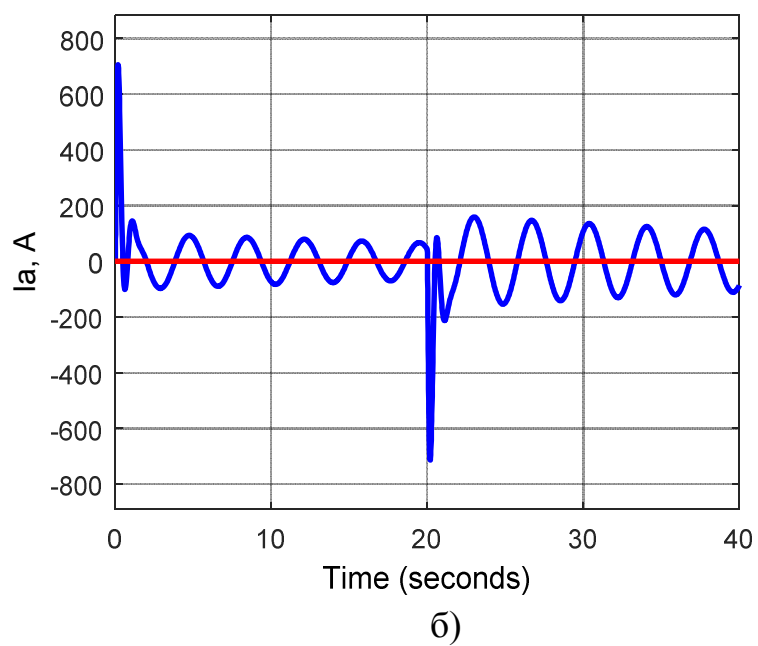
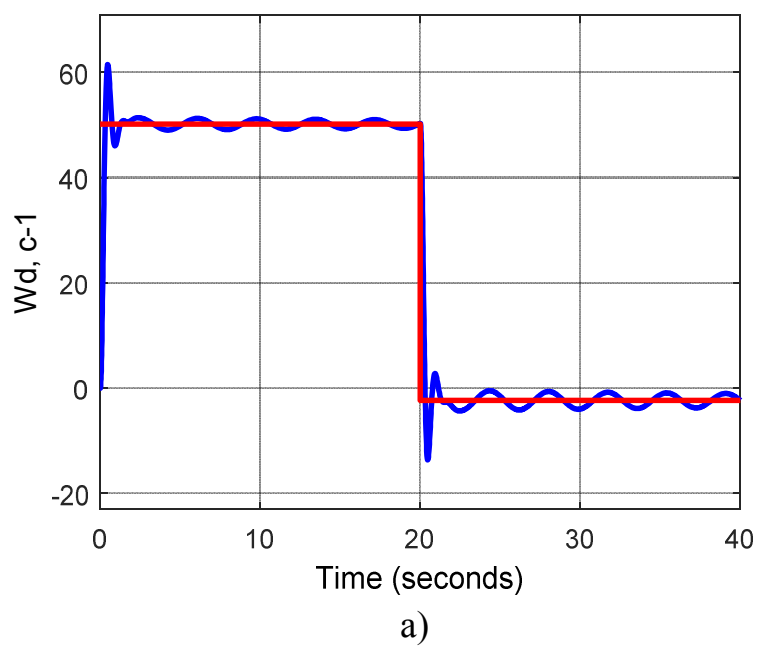


Рисунок 3.5 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по задаючій дії

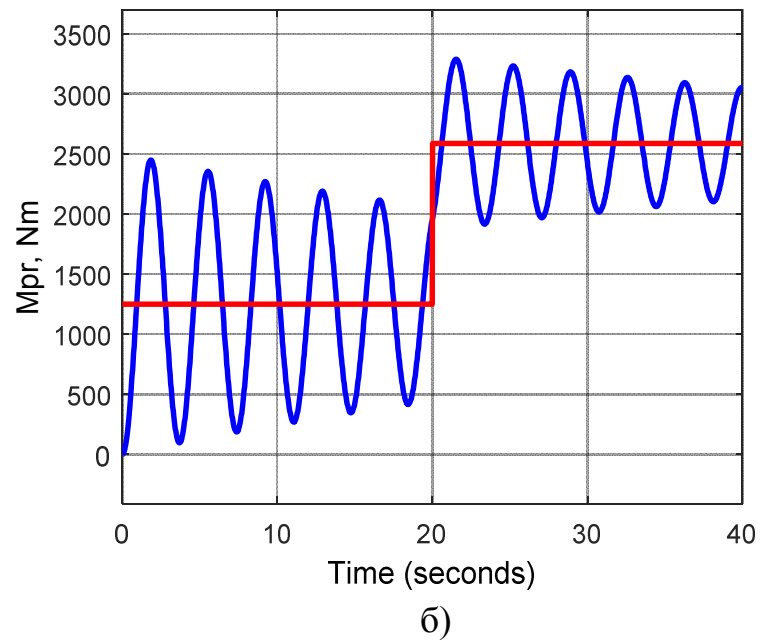
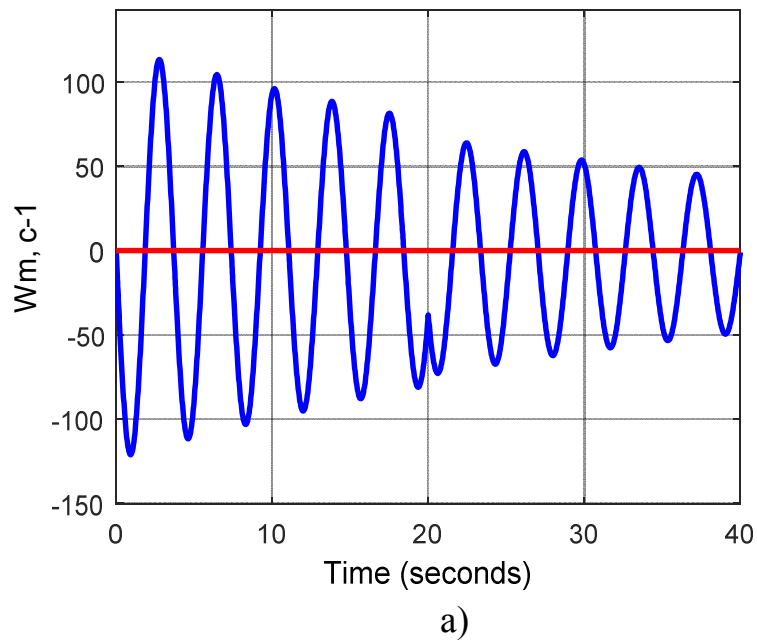


Рисунок 3.6 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{\text{пр}} = f(t)$ (б) двомасової системи по збурюючій дії

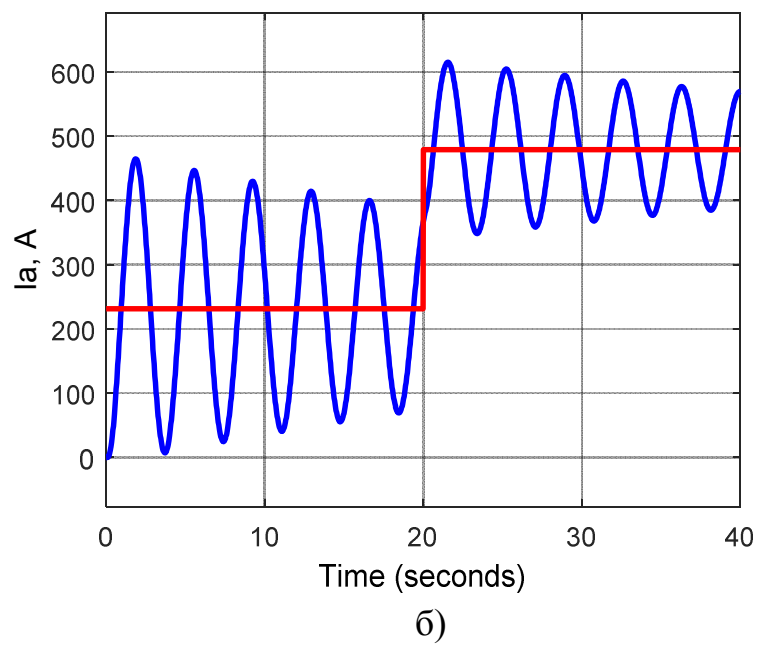
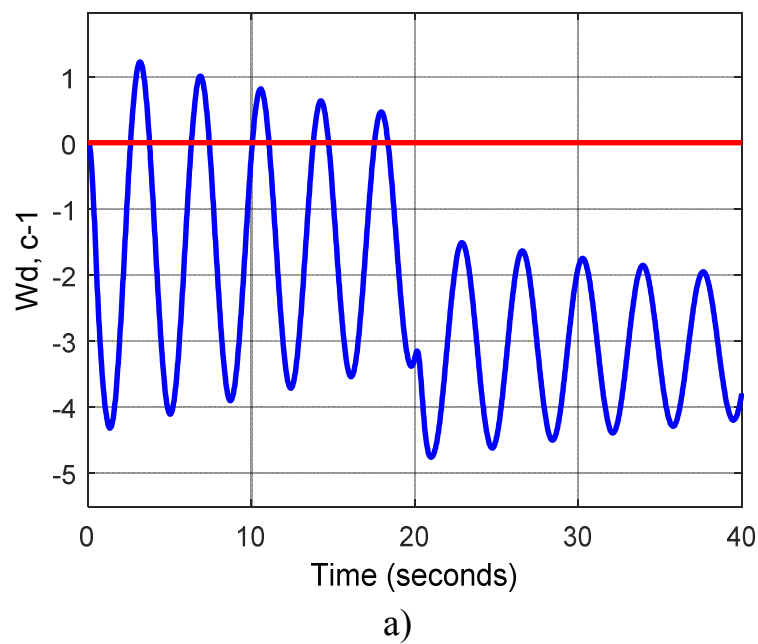


Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму обертання колеса роторного екскаватора за наявності пружних властивостей системи ротор-стріла- підвіска стріли, фортеці породи і т.п. При несприятливих поєднаннях швидкостей роторного колеса і зазначених факторів можуть виникати резонансні коливання металоко-нструкцій, що несприятливо впливає на роботу механізму. Показано, що кіне-матична схема механічної частини системи управління механізмом обертання колеса роторного екскаватора з урахуванням пружності механічних зв'язків може розглядатися як двомасова.

2. Виконано розробку математичної моделі. Отримано рівняння стану двомасової системи і рівняння стану двомасової системи у векторному – матри-чній формі, визначені елементи вектора стану, тобто змінні стану системи; ви-значений вектор зовнішніх дій, а також матриці стану **A** і управління **B**. Розро-блена алгоритмічна схема двомасової системи і спрощена алгоритмічна схема двохмасової системи з оптимізованим по модульному критерію контуром стру-му.

3. Проведено моделювання двомасової системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблено схему моделі двомасової системи. Як тестовий сигнал на вхід сис-теми подавався ступінчастий сигнал з випадковою амплітудою, який знахо-диться в межах від $+U_{z \max}$ до $-U_{z \max}$, що відповідає зміні швидкості в межах номінальних значень. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи мають незадовільний (коливальний) характер.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

4.1 Регулятори, реалізовані в пакеті прикладних програм **Neural Network Toolbox** системи **MATLAB**

Синтез системи управління проводився за допомогою **Neural Network Toolbox** в системі **MATLAB**. Далі випадку коротко описується цей інструментарій, процес синтезу нейрорегулятора та висвітлені результати моделювання системи управління.

В [53] розглянуто три типи нейронних мереж, які вбудовані в **Neural Network Toolbox MATLAB** і застосовуються у наступних типах регуляторів:

- Регулятор з прогнозом (**NN Predictive Controller**);
- Регулятор на основі моделі авторегресії з ковзаючим середнім (**NARMA-L2 Controller**);
- Регулятор на основі еталонної моделі (**Model Reference Controller**).

Використання нейронних мереж у задачах управління може бути розкладено на два етапи проектування:

- Етап ідентифікації об'єкта управління;
- Етап синтезу правила управління.

На етапі ідентифікації створюється модель керованого об'єкту у вигляді нейронної мережі, яка потім використовується на етапі синтезу для розробки регулятора. Для кожної з трьох архітектур використовується той самий процес ідентифікації, проте етапи синтезу суттєво відрізняються.

При управлінні з прогнозом, модель керованого об'єкту використовується для прогнозування його майбутньої поведінки, а алгоритм оптимізації застосовується для обчислення управління, яке мінімізує різницю між бажаними та фактичними змінами виходу моделі.

При управлінні на основі моделі авторегресії з ковзаючим середнім, регулятор є простою реконструкцією моделі керованого об'єкту.

При управлінні на основі еталонної моделі, регулятор - це навчена нейронна мережа, яка управляє об'єктом так, щоб він відтворював поведінку еталонної моделі. У цьому випадку модель керованого об'єкту активно використовується для налаштування параметрів самого регулятора.

Динамічні моделі систем управління з нейромережевими регуляторами розміщені в спеціальному розділі **Control Systems Neural Network Toolbox** (див. рис. 4.1).

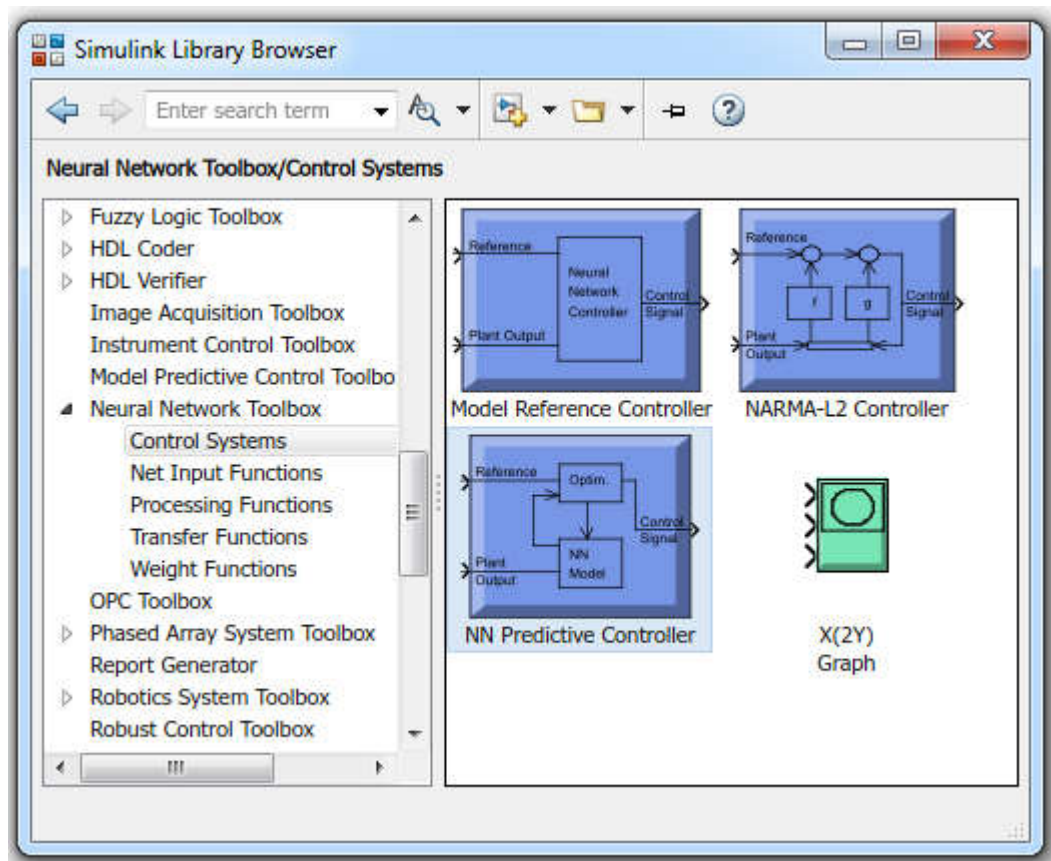


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

Регулятор з прогнозом використовує модель керованого процесу у вигляді нейронної мережі для передбачення майбутньої реакції процесу на випадкові управляючі сигнали. Алгоритм оптимізації визначає керуючі сигнали, які мінімізують відмінність між очікуваними та фактичними змінами виходу моделі, тим самим оптимізуючи управління процесом. Побудова моделі керованого процесу автономно виконується за допомогою нейронної мережі, яка навчається

ся в груповому режимі за одним з алгоритмів навчання. Цей тип регулятора потребує значних обчислювальних ресурсів, оскільки оптимізація відбувається на кожному такті управління.

Регулятор на основі моделі авторегресії з ковзаючим середнім вимагає найменшої кількості обчислень серед всіх типів регуляторів. Цей регулятор є простою реконструкцією нейромережевої моделі керованого об'єкту, яка була отримана під час автономної ідентифікації. Його обмеження полягає у тому, що модель об'єкту повинна бути визначена у канонічній формі простору стану, що відповідає певній матриці, що може призводити до обчислювальних похибок.

Регулятор на основі еталонної моделі вимагає обчислювальних ресурсів, що порівнюються з попереднім типом. Однак архітектура цього регулятора потребує навчання нейронної мережі як для об'єкту управління, так і для регулятора. Навчання регулятора виявляється досить складним, оскільки воно ґрунтується на динамічному варіанті методу зворотного розповсюдження помилок. Перевагою регуляторів на основі еталонної моделі є їхнє застосування до різних класів об'єктів управління.

У даній роботі для використання в системі був обраний регулятор з прогнозом, оскільки використання регулятора на основі моделі авторегресії з ковзаючим середнім і регулятора на основі еталонної моделі не призвело до позитивних результатів.

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

Регулятор NN Predictive Controller, що реалізований у пакеті Neural Network Toolbox, використовує нейронну мережу для моделювання нелінійної динаміки керованого об'єкту з метою прогнозування його майбутньої поведінки. Більше того, цей регулятор розраховує сигнал управління, що оптимізує реакцію об'єкту протягом визначеного інтервалу часу.

4.2.1 Ідентифікація об'єкту управління.

Схема підсистеми ідентифікації, яка показана на рисунку 4.2, включає в себе модель об'єкту управління у формі нейронної мережі, яка повинна пройти автономне навчання для мінімізації різниці між реакціями об'єкту та моделлю на послідовність тестових сигналів.

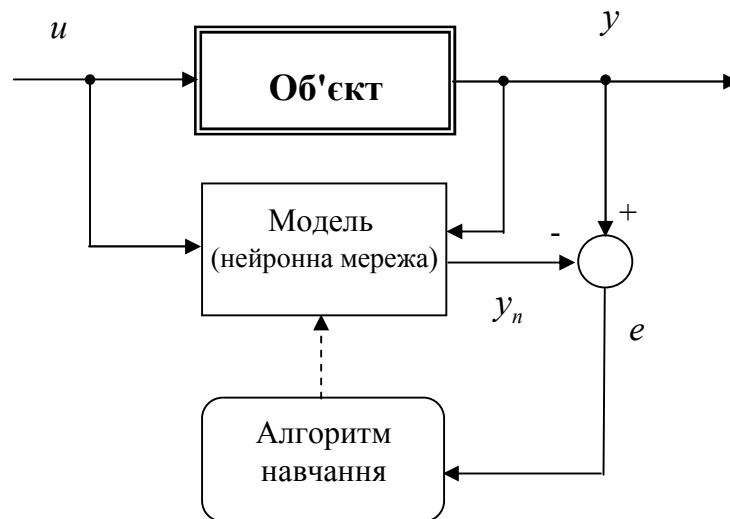


Рисунок 4.2 – Схема підсистеми ідентифікації.

Нейронна мережа, що представлена на рисунку 4.3 і використовується у регуляторі об'єкту управління, має два шари нейронів та використовує лінії за-тримки (ЛЗ) для збереження попередніх значень входів та виходів об'єкту з метою передбачення майбутніх значень виходу.

Наведення параметрів цієї мережі виконується автономно за допомогою методу групового навчання, використовуючи дані, накопичені під час експериментів з реальним об'єктом. Для навчання мережі може бути використано будь-який із доступних алгоритмів навчання для нейронних мереж. У даній роботі був використаний алгоритм Левенберга-Марквардта, реалізований у вигляді М-функції `trainlm` у пакеті **Neural Network Toolbox**.

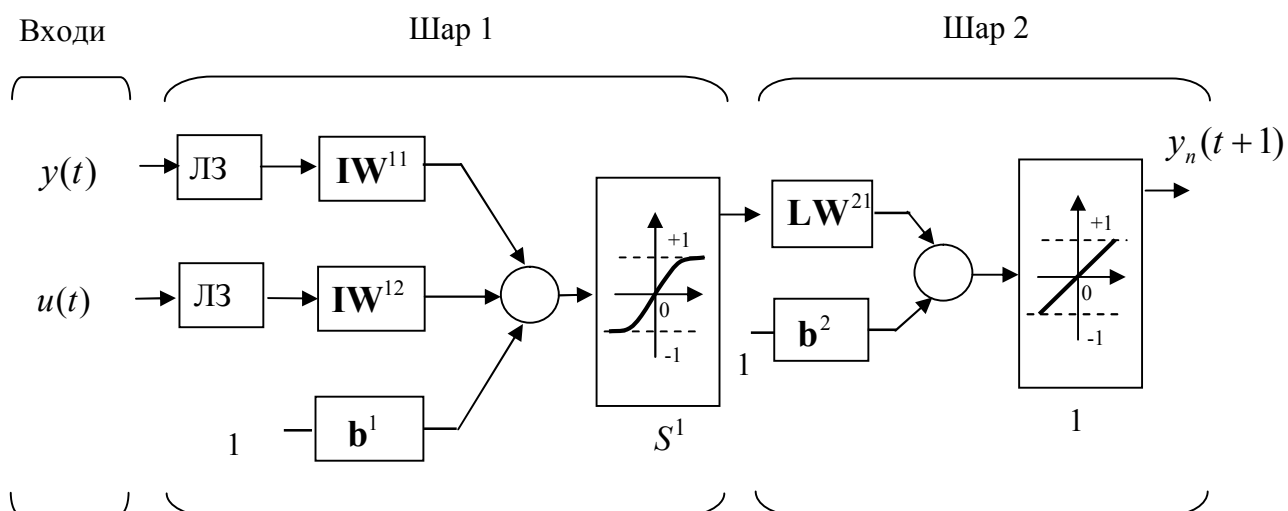


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

Управління з прогнозом використовує принцип передбачення реакції керованого об'єкту на певний майбутній інтервал часу за допомогою нейромережевої моделі. Прогноз використовується у чисельній оптимізації для розрахунку сигналу управління, який мінімізує критерій якості, що визначений таким чином:

$$J = \sum_{j=N_1}^{N_2} [y_r(t+j) - y_m(t+j)]^2 + \rho \sum_{j=1}^{N_4} [u'(t+j-1) - u'(t+j-2)]^2, \quad (4.1)$$

де константи N_1 , N_2 і N_4 визначають межі, в яких обчислюються помилки стеження та потужність управляючого сигналу. Перемінна u' сигнал завдання, що подається не нейромережеву модель, y_r – очікувана, а y_m – фактична реакція моделі системи. Змінна ρ визначає внесок потужності управління у критерій якості. На рисунку 4.4 показано структурну схему процесу управління з прогнозом, де регулятор складається з нейромережевої моделі керованого об'єкту і блоку оптимізації. Блок оптимізації визначає значення управління, яке мінімізує критерій якості управління, що управляє процесом.

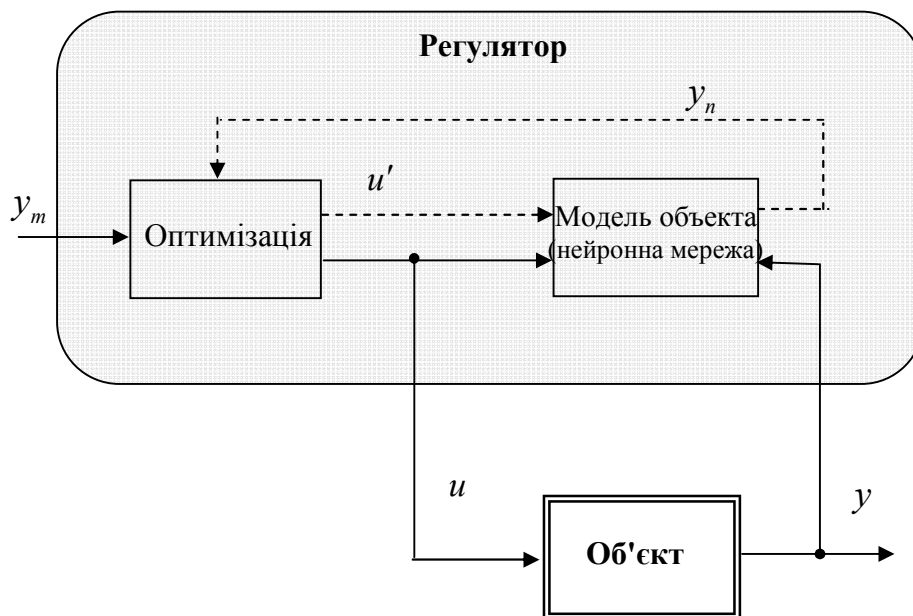


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB

При синтезі нейрорегулятора NN Predictive Controller використовуються наступні файли, розміщені в каталозі toolbox/nnet/nncontrol системи Simulink.

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

Csrchbre – метод Брента, об'єднуючий методи золотого перетину і квадратичної інтерполяції;

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

Prest3sim2 – нейромережева модель управління використовувана М-функцією `deport` для прогнозу процесу в майбутньому.

Допоміжні функції:

Nncontrolutil — утиліта, яка забезпечує доступ до приватних функцій у середовищі Simulink;

Nnident.m — головна функція для ідентифікації об'єкта, розташована в каталозі **private**. Вона надає інтерфейс GUI для користувача, виконує генерацію навчальної вибірки, створює та тренує нейронну мережу.

У вікні системи SIMULINK формується схема системи із структурою, показаною на рисунку 4.5.

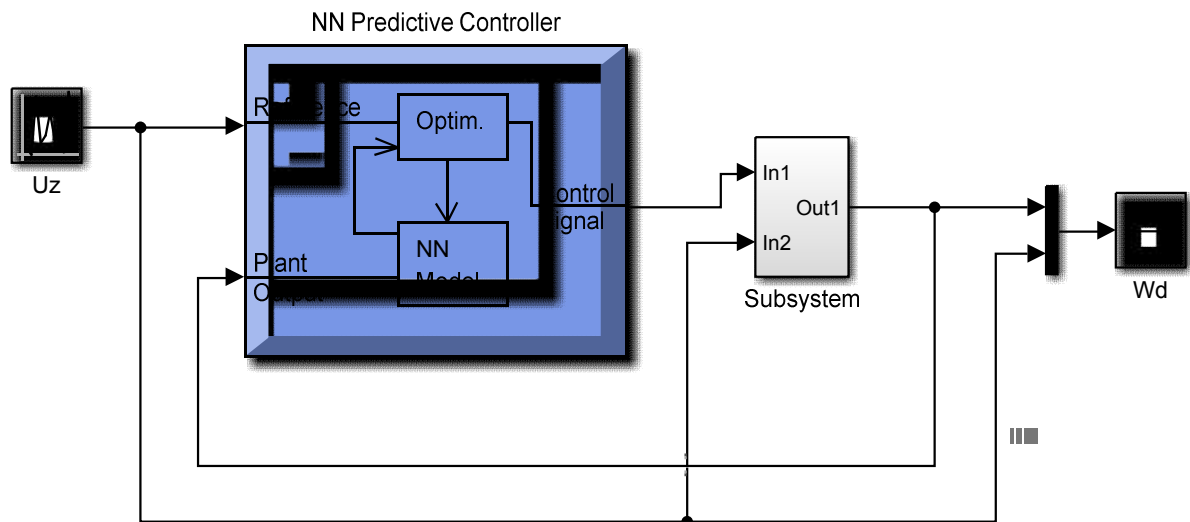


Рисунок 4.5 – Схема системи управління з нейрорегулятором

NN Predictive Controller

Ця структура включає блок керованого об'єкту **Subsystem** і блок регулятора **NN Predictive Controller**, а також блоки генерації еталонного ступінчастого сигналу з випадковою амплітудою **Random Reference**, блок побудови графіків. Особливість цієї структури полягає в тому, що вона виконується не тільки функції блок-схеми системи SIMULINK, але і функції графічного інтерфейсу користувача GUI.

Структурна схема нейрорегулятора **NN Predictive Controller** представле-

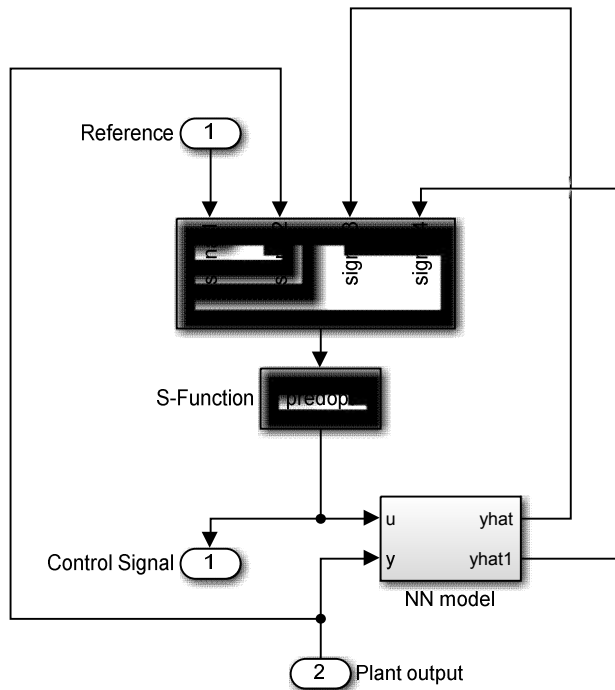


Рисунок 4.6 – Структурна схема нейрорегулятора

на на рисунку 4.6. Ця схема відображається у відокремленому вікні, яке вибирається через пункт меню **Look Under Mask** блоку **NN Predictive Controller**.

Процес створення нейрорегулятора починається з активації блоку **NN Predictive Controller**. З'являється вікно, показане на рис. 4.7. Це вікно виконує функції графічного інтерфейсу користувача.

У рамці вікна рис. 4.3 відображається повідомлення, що вказує на подальші кроки: *"Perform plant identification before controller configuration"*. Це означає, що перед конфігуруванням регулятора необхідно спершу провести ідентифікацію керованого об'єкту, тобто створити його нейромережеву модель, використовуючи спеціальну процедуру **Plant Identification**.

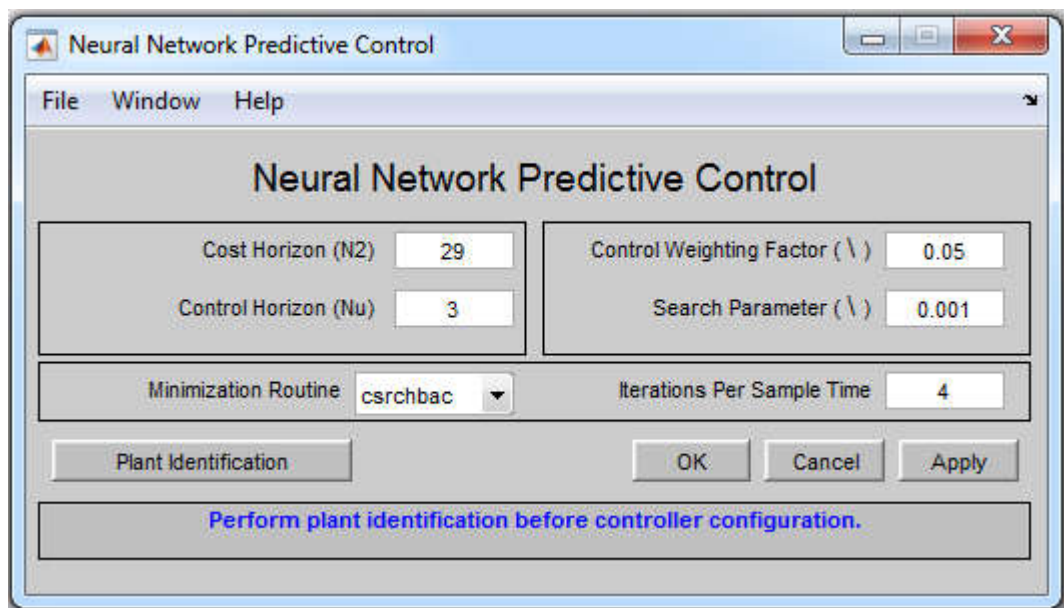


Рисунок 4.7 – Вікно завдання параметрів регулятора

Вікно **Plant Identification** зображено на рисунку 4.8. Ця структура є універсальною і може бути використана для розробки нейромережових моделей для будь-якого динамічного об'єкту, який описаний за допомогою моделі SIMULINK. В завданні, що розглядається, моделлю є двомасова система.

Процедура ідентифікації дозволяє побудувати нейронну мережу, яка відтворюватиме динаміку керованого об'єкту. Оскільки ця модель має застосовуватися під час налаштування регулятора, її слід створити перед розпочатком процесу налаштування регулятора

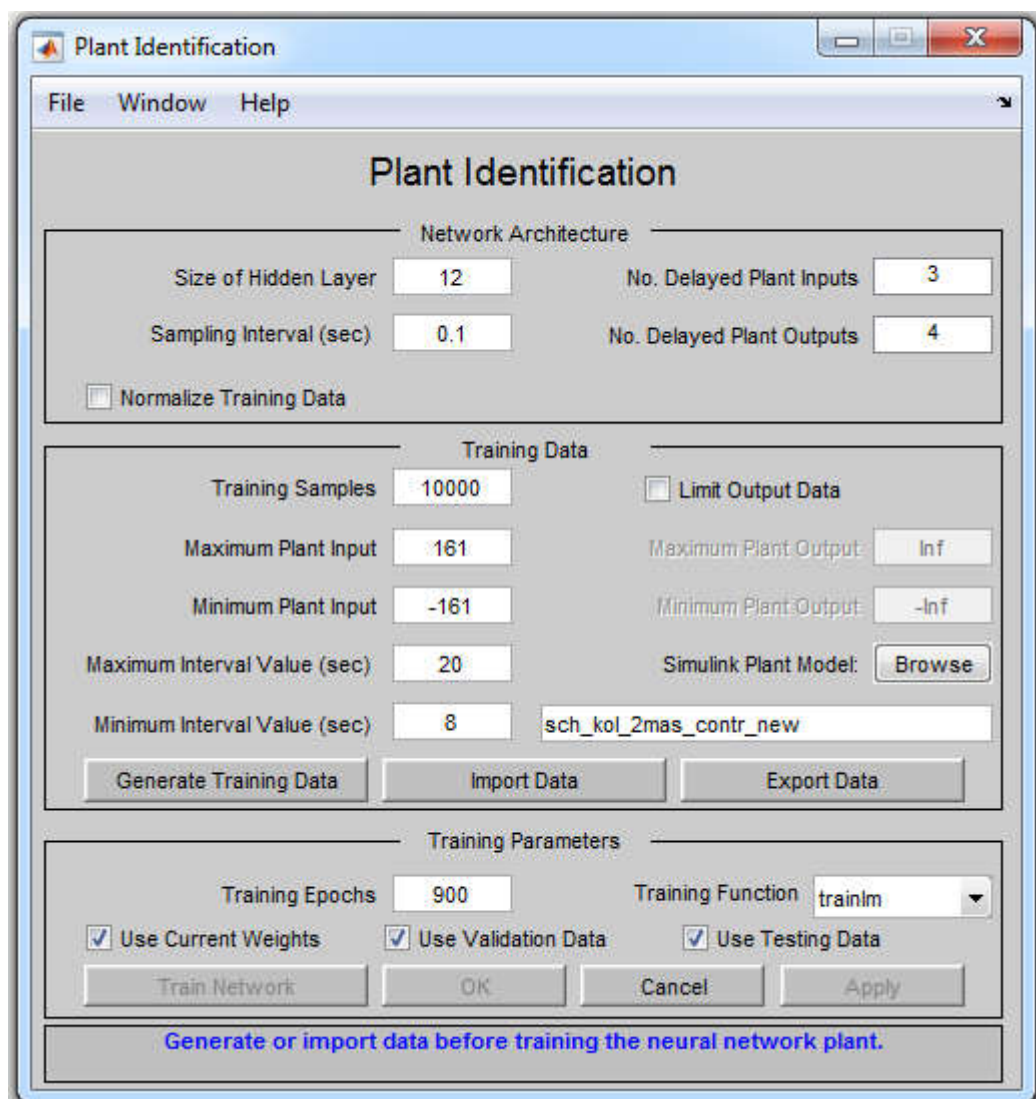


Рисунок 4.8 – Вікно ідентифікації керованої системи

Процедура ідентифікації вимагає завдання ряду параметрів, які задаються з використанням вікна Plant Identification. Дане вікно містить кілька секцій.

Опис вікна Plant Identification

Секція Network Architectuer (Архітектура нейронної мережі). Параметри, що задаються в цій секції визначають архітектуру нейронної мережі, яка використовується для ідентифікації і моделювання динаміки системи. Вибір правильних параметрів секції Network Architecture є ключовим етапом в ідентифікації системи, оскільки від них залежить здатність моделі адекватно відтворювати динаміку досліджуваної системи.

Size of Hidden Layer (розмір прихованого шару). Параметр Size of Hidden Layer вказує на кількість нейронів у прихованому шарі нейронної мережі, яку ви використовуєте для моделювання або ідентифікації об'єкта або системи. Вибір розміру прихованого шару є важливим параметром при проектуванні нейронної мережі, оскільки він впливає на потужність і здатність мережі моделювати складність взаємозв'язків у ваших даних. Зазвичай, збільшення розміру прихованого шару може допомогти нейронній мережі навчатися більш складним залежностям у даних, але воно також може призвести до перенавчання (overfitting), тобто погіршення загальної здатності мережі генералізувати результати на нові дані.

Розмір прихованого шару часто є гіперпараметром, який може бути експериментально налаштований для досягнення найкращих результатів у конкретній задачі.

Sampling Interval (sec) (інтервал вибірки в секундах). Параметр Sampling Interval (sec) визначає інтервал між двома послідовними моментами знімання даних. Параметр важливий при зборі даних для моделювання або ідентифікації системи, оскільки впливає на точність та роздільну здатність зібраних даних. Для точної роботи моделі ідентифікації важливо визначити правильний інтервал вибірки, який враховує динаміку системи і вимоги до точності результатів ідентифікації.

No. Delayed Plant Inputs (кількість затриманих входів системи). Параметр No. Delayed Plant Inputs вказує на кількість запізнених (затриманих) входів системи, які враховуються при ідентифікації динаміки системи. Запізнені входи

враховують затримки у вхідних сигналах системи і допомагають створити більш точну модель системи.

Запізнені входи особливо важливі в ідентифікації систем, де існують затримки між вхідними сигналами та відповідними виходами. Наприклад, в автоматичному керуванні часто є затримки в активних процесах, і ідентифікація таких систем може вимагати врахування цих затримок.

Кількість No. Delayed Plant Inputs вказує, скільки вхідних сигналів системи вважаються запізненими і будуть використовуватися при створенні моделі системи. Від правильного визначення цього параметра залежить точність ідентифікації системи, особливо в умовах існування затримок між сигналами.

Цей параметр важливий для налаштування ідентифікаційного процесу та допомагає створити більш адекватну модель системи, яка може враховувати реальні умови та характеристики системи керування.

No. Delayed Plant Outputs (кількість затриманих виходів системи). Параметр No. Delayed Plant Outputs вказує на кількість запізнених (затриманих) виходів системи, які враховуються при ідентифікації динаміки системи. Запізнені входи враховують затримки у вихідних сигналах системи і допомагають створити більш точну модель системи.

Цей параметр дозволяє визначити, скільки затриманих виходів системи вважаються при ідентифікації і впливає на точність та адекватність створеної моделі. Затримані входи дозволяють враховувати затримки в вихідних сигналах, які можуть бути важливими для ефективної ідентифікації та моделювання системи.

Вибір кількості No. Delayed Plant Outputs залежить від конкретних характеристик досліджуваної системи та вимог до точності ідентифікації.

Normalize Training Data (нормалізація даних навчання). Вікна контролю нормування навчальних даних. Normalize Training Data вказує на необхідність нормалізації даних під час навчання моделі системи. Нормалізація даних - це процес масштабування вхідних та вихідних даних системи до певного діапазону або стандартної форми перед подачею їх на вхід моделі для навчання.

Якщо вікно увімкнено, то вхідні та вихідні дані будуть нормалізовані перед початком процесу навчання моделі. Нормалізація даних може включати в себе масштабування даних до діапазону від 0 до 1, використання стандартного відхилення та середнього значення для центрування даних, або інші методи нормалізації.

Якщо вікно не увімкнено, то нормалізація даних не буде застосована, і дані будуть використовуватися без змін.

Мета нормалізації даних під час навчання полягає в тому, щоб забезпечити стабільність та швидкість навчання моделі, особливо коли вхідні дані мають різний масштаб або діапазон значень. Нормалізація може полегшити збіжність навчання та поліпшити точність ідентифікації системи.

Вибір варіанта Normalize Training Data повинен здійснюватися відповідно до характеру та потреб конкретної задачі ідентифікації системи

Секція Training Data (дані навчання). Training Data - це набір даних, які використовуються для навчання моделі ідентифікації системи. Ці дані представляють собою вхідні та вихідні сигнали системи, і їх використовують для створення математичної моделі системи.

Основна мета Training Data полягає в навчанні моделі системи таким чином, щоб вона могла адекватно відтворювати динаміку реальної системи. Дані включають в себе виміряні значення входів та виходів системи протягом певного періоду часу або при певних умовах.

Зазвичай Training Data містять дані з різних режимів роботи системи, щоб модель була здатна відтворювати різноманітні аспекти системи. Важливо, щоб ці дані були репрезентативними та відображали різноманітні умови функціонування системи.

На основі цих даних проводиться процес ідентифікації, під час якого створюється модель, яка намагається знайти математичні зв'язки між вхідними та вихідними сигналами системи. Точність та адекватність цієї моделі залежать від якості та репрезентативності Training Data.

Training samples (навчальні зразки). Довжина навчальної вибірки (кількість точок знімання інформації). Навчальна вибірка включає в себе вхідні та вихідні дані, які використовуються для створення моделі системи. Довжина навчальної вибірки грає важливу роль у процедурі ідентифікації об'єкта і створенні адекватної моделі системи.

Maximum Plant Input (максимальне значення вхідного сигналу). Параметр Maximum Plant Input вказує на максимальне значення вхідного сигналу (величини, яка керує системою) під час ідентифікації динаміки системи. Цей параметр встановлює верхню межу діапазону значень вхідного сигналу, які будуть використовуватися під час навчання моделі.

Максимальне значення вхідного сигналу важливо вказати для того, щоб навчальний процес був обмежений реалістичними значеннями вхідного сигналу. Вказане значення може відображати фізичні обмеження системи або діапазони значень, які найчастіше використовуються в реальних умовах.

Важливо правильно налаштувати цей параметр, оскільки надмірно велике значення може призвести до надмірного навчання моделі на діапазонах вхідних сигналів, які надто віддалені від реальних умов. Це може призвести до некоректних прогнозів та погіршення якості моделі системи.

Minimum Plant Input (мінімальне значення вхідного сигналу). Параметр Minimum Plant Input вказує на мінімальне значення вхідного сигналу (величини, яка керує системою) під час ідентифікації динаміки системи. Цей параметр встановлює нижню межу діапазону значень вхідного сигналу під час навчання моделі.

Максимальне і мінімальне значення вхідного сигналу важливо вказати для того, щоб навчальний процес був обмежений реалістичними значеннями вхідного сигналу. Вказане значення може відображати фізичні обмеження системи або діапазони значень, які найчастіше використовуються в реальних умовах. Неправильне встановлення цих параметрів може призвести до некоректних прогнозів та погіршення якості моделі системи.

Maximum Interval Value (sec) (максимальне значення інтервалу). Параметр Maximum Interval Value (sec) вказує на максимальну довжину інтервалу часу в секундах, протягом якого використовуються вхідні та вихідні дані для ідентифікації динаміки системи, тобто максимальний інтервал ідентифікації. Цей параметр дозволяє обмежити часовий період, на який поширюється навчальний набір даних, використовуваний під час ідентифікації.

Вказане максимальне значення в секундах служить для обмеження часового діапазону аналізу інформації. Це корисно в ситуаціях, коли вхідні та вихідні дані мають значення протягом певного часового інтервалу, і деякий період часу може бути менш важливим для ідентифікації системи.

Зазвичай, під час встановлення параметру Maximum Interval Value, враховуються характеристики конкретної ідентифікаційної задачі. Навчальний набір даних може включати в себе декілька циклів або періодів роботи системи, і цей параметр допомагає обмежити аналіз до конкретного періоду часу.

Minimum Interval Value (sec) (мінімальне значення інтервалу). Параметр Minimum Interval Value (sec) вказує на мінімальну довжину інтервалу часу в секундах, протягом якого використовуються вхідні та вихідні дані для ідентифікації динаміки системи, тобто мінімальний інтервал ідентифікації. Цей параметр дозволяє обмежити часовий період, на який поширюється навчальний набір даних, використовуваний під час ідентифікації.

Мінімальне значення вказує на мінімальний період, який вважається допустимим для аналізу даних. Налаштування Minimum Interval Value (sec) допомагає контролювати, які частини часового ряду вважати допустимими для використання під час ідентифікації, і може допомогти враховувати тільки ті інтервали, які мають певну мінімальну тривалість або важливість.

Цей параметр важливий для налаштування ідентифікаційного процесу відповідно до конкретних вимог і особливостей досліджуваної системи.

Limit Output Data (обмеження вихідних даних). Вікно контролю, що дозволяє обмежити об'єм вихідних даних, тільки при включеному вікні будуть доступні 2 наступних вікна редагування тексту.

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

Параметри Maximum Plant Output і Minimum Plant Output вказують на обмеження вихідних даних, які використовуються під час ідентифікації динаміки системи. Ці параметри можуть бути використаними для вибору конкретного діапазону вихідних даних для аналізу та навчання моделі системи.

Параметри Maximum Plant Output і Minimum Plant Output корисні, коли потрібно провести аналіз лише на певному діапазоні вихідних даних, або коли дослідника цікавить певний аспект системи, що відбувається в певний період часу.

Параметри Maximum Plant Output і Minimum Plant Output дозволяють зменшити обсяг даних, які використовуються для ідентифікації, і спростити процес аналізу та навчання моделі системи.

Simulink Plant Model (Simulink модель об'єкту). Завдання моделі Simulink з вказівкою вхідних і вихідних портів, використовуваних при побудові нейромережевої моделі керованої системи. Вказується модель об'єкта або системи, яка використовується для ідентифікації.

За допомогою кнопки **Browser** вибирається модель Simulink об'єкту управління; в даній роботі це схема моделі, показана на рис. 4.9.

Generate Training Data (генерація навчальних даних). Кнопка запуску процесу генерації навчальної послідовності, тобто створення навчальних даних для ідентифікації динаміки системи на основі параметрів та моделі системи. Ця функція дозволяє симулювати дані для навчання моделі, коли реальні навчальні дані не доступні.

Import Data. Імпорт навчальної послідовності з робочої області або файлу даних.

Export Data. Експорт даних, що згенерували, в робочу область або MAT – файл.



Секція Training Parameters (параметри навчання). Training Parameters

- це набір налаштувань, які визначають, як саме буде проводитися процес навчання моделі системи на основі навчальних даних. Ці параметри дозволяють керувати процесом навчання ідентифікаційної моделі та впливати на його результати.

Training Epochs (епохи навчання). Training Epochs вказують на кількість ітерацій (повних циклів навчання) для навчання моделі або алгоритму ідентифікації. Кожна епоха представляє собою одну ітерацію, під час якої модель навчається на підготовчих даних та оновлює свої внутрішні параметри, щоб покращити свою точність.

Кількість Training Epochs визначає, скільки разів модель або алгоритм пройде через навчальні дані для покращення своєї здатності до відтворення динаміки системи. Зазвичай навчання включає багато епох, і після кожної епохи модель оцінюється на тестових даних, щоб визначити, наскільки добре вона справляється з завданням ідентифікації.

Вибір кількості Training Epochs може бути важливим параметром при налаштуванні процесу навчання. Занадто мала кількість епох може призвести до недоосвіченої моделі, яка не здатна адекватно відтворити динаміку системи, тоді як занадто велика кількість епох може призвести до перенавчання (overfitting) моделі на навчальних даних, що погіршує її здатність до узагальнення на нові дані. Точна кількість епох може варіюватися в залежності від конкретної задачі ідентифікації.

Training function (навчальна функція). Training Function вказує на функцію, яка використовується для навчання нейромережевої моделі об'єкта або системи. Ця функція визначає, яким чином модель навчається на навчальних даних та які параметри моделі оптимізуються під час навчання.

Вибір правильної Training Function важливий для успішного навчання моделі. У MATLAB і Neural Network Toolbox, доступні різні Training Functions для нейронних мереж та інших моделей ідентифікації, наприклад: `traingd` - градієнтний спуск; `traingdm` - градієнтний спуск з моментом; `trainlm` - метод найменших квадратів Левенберга-Марквардта; `trainbr` - байєсівський регулятор.

Кожна з цих функцій має свої особливості та параметри, які можна налаштувати для досягнення найкращих результатів в конкретній задачі ідентифікації. Training Function визначає, як модель або алгоритм оновлює внутрішні параметри під час навчання та яким чином він збільшує точність та здатність моделі до адекватного відтворення динаміки системи.

Use Current Weights (використовувати поточні ваги). Вікно контролю, що дозволяє підтвердити використання поточних вагів нейронної мережі.

Параметр Use Current Weights вказує, чи слід використовувати поточні ваги для ідентифікації системи під час навчання моделі. Цей параметр впливає на початкову точку або ініціалізацію процесу навчання.

Якщо Use Current Weights встановлено в «Так» (або включено), то попередні ваги (параметри моделі) вважаються початковими значеннями для нового процесу навчання. Це означає, що процес навчання розпочнеться з вагами, які вже були встановлені або навчені під час попередніх ітерацій.

Якщо Use Current Weights встановлено в «Ні» (або виключено), то навчання розпочнеться зі звичайними початковими значеннями, які зазвичай є випадковими або ініціалізованими за допомогою певних методів.

Використання поточних ваг при навчанні може бути корисним, якщо необхідно покращити або доповнити існуючу модель, або якщо відомо, що поточні ваги вже набули певного рівня адекватності. Важливо розуміти, як цей параметр впливає на процес навчання та на результати ідентифікації системи.

Use Validation Data (використовувати дані валідації). Вікно контролю, що дозволяє вказати, чи слід використовувати окремий набір даних для валідації моделі під час навчання ідентифікаційної системи. Валідація - це процес оцінки якості побудованої моделі на основі навчальних даних.

Якщо Use Validation Data встановлено в «Так» (або включено), то під час навчання моделі буде використовуватися окремий набір даних, який не брав участь в процесі навчання. Ці дані використовуються для оцінки того, наскільки добре модель узгоджується з даними, які вона раніше не бачила. Викорис-

тання валідаційних даних допомагає виявити ознаки перенавчання моделі та визначити, наскільки добре модель узгоджується з новими вхідними даними.

Якщо Use Validation Data встановлено в «Ні» (або виключено), то валідація не буде використовуватися, і модель буде оцінюватися лише на основі навчальних даних. Використання валідаційних даних рекомендується для кращої оцінки якості моделі та попередження перенавчання, але варіант без валідації може бути використаний, коли недоступні додаткові дані для цієї цілі.

Use Testing Data (використовувати тестові дані). Вікно контролю, що дозволяє вказати, чи слід використовувати окремий набір тестових даних для оцінки якості побудованої моделі системи після її навчання. Тестові дані - це дані, які модель не бачила під час навчання або валідації.

Якщо Use Testing Data встановлено в «Так» (або включено), то після завершення навчання і валідації моделі буде використовуватися окремий набір тестових даних для оцінки якості моделі. Ці дані допомагають визначити, наскільки добре модель здатна передбачати реальну динаміку системи, якщо у неї немає попередніх знань про ці дані.

Якщо Use Testing Data встановлено в «Ні» (або виключено), то тестові дані не використовуються, і модель оцінюється лише на основі навчальних і, можливо, валідаційних даних.

Використання тестових даних важливе для об'єктивної оцінки якості моделі та підтвердження її ефективності на нових, незалежних вибірках даних. Тестові дані допомагають уникнути перенавчання та підтвердити, наскільки добре модель може працювати на нових вхідних даних.

Виклик процедури **Generate Training Data** призведе до запуску програми для створення навчальної послідовності. Ця програма генерує навчальні дані шляхом впливу послідовності випадкових ступінчастих сигналів на модель Simulink керованого об'єкту. Графіки вхідного та вихідного сигналів об'єкту управління відображаються на екрані (див. рис. 4.11). Після завершення процесу створення навчальної послідовності користувач має можливість прийняти (**Accept Data**) або відхилити (**Reject Data**) згенеровані дані.

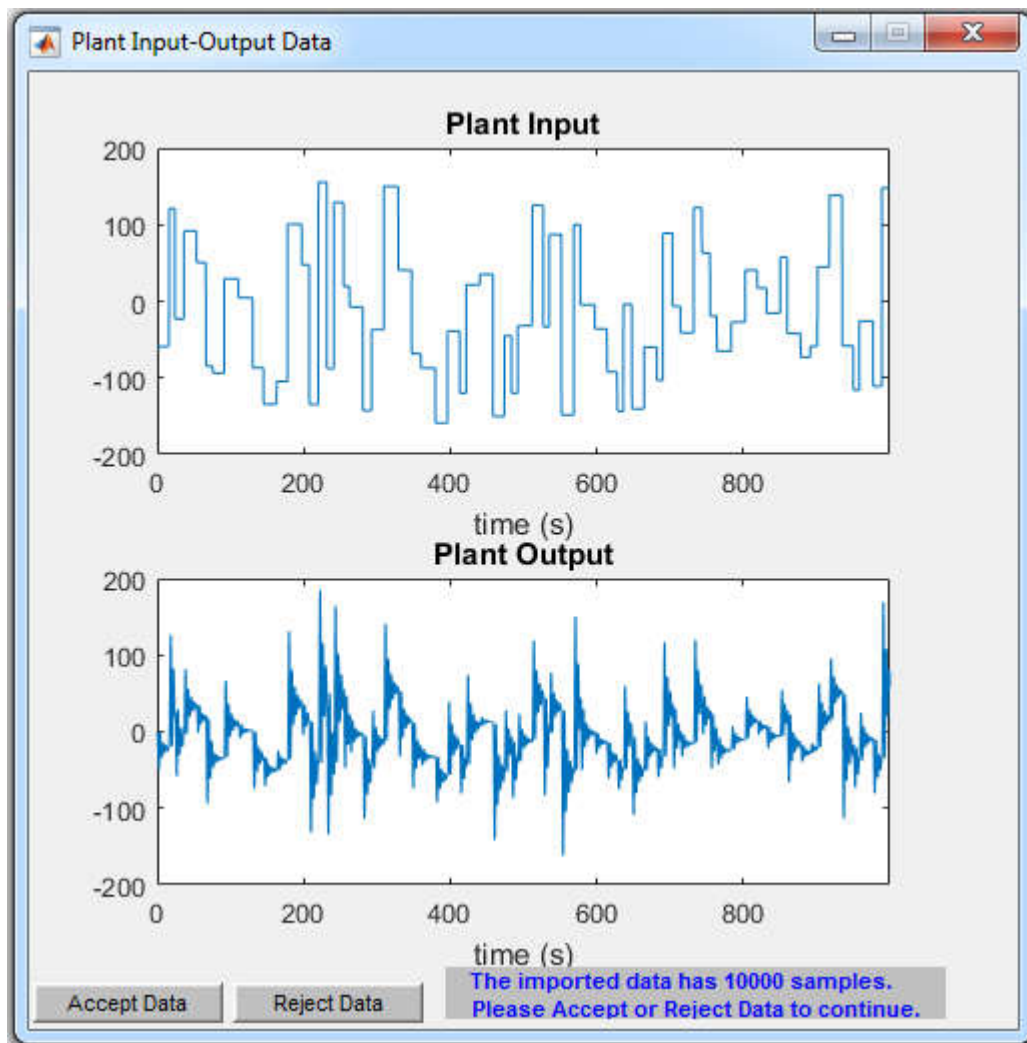


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

Якщо дані прийняті, програма відкриє модифіковане вікно **Plant Identification** (див. рис. 4.12). У цьому вікні деякі елементи стають недоступними, а кнопка **Generate Training Data** замінюється кнопкою **Erase Generate Data**, яка дає можливість видалити згенеровані дані.

Після натискання на кнопку **Train Network** відбувається створення мережі з прямим розподілом сигналу за допомогою М-функції **newff**. Ця функція не лише створює мережу з ім'ям **netn**, але й ініціалізує її ваги та зсуви, готуючи нейронну мережу до навчання. Модель нейронної мережі може бути відображена у системі Simulink за допомогою оператора **gensim(netn)** (див. рис. 4.13)

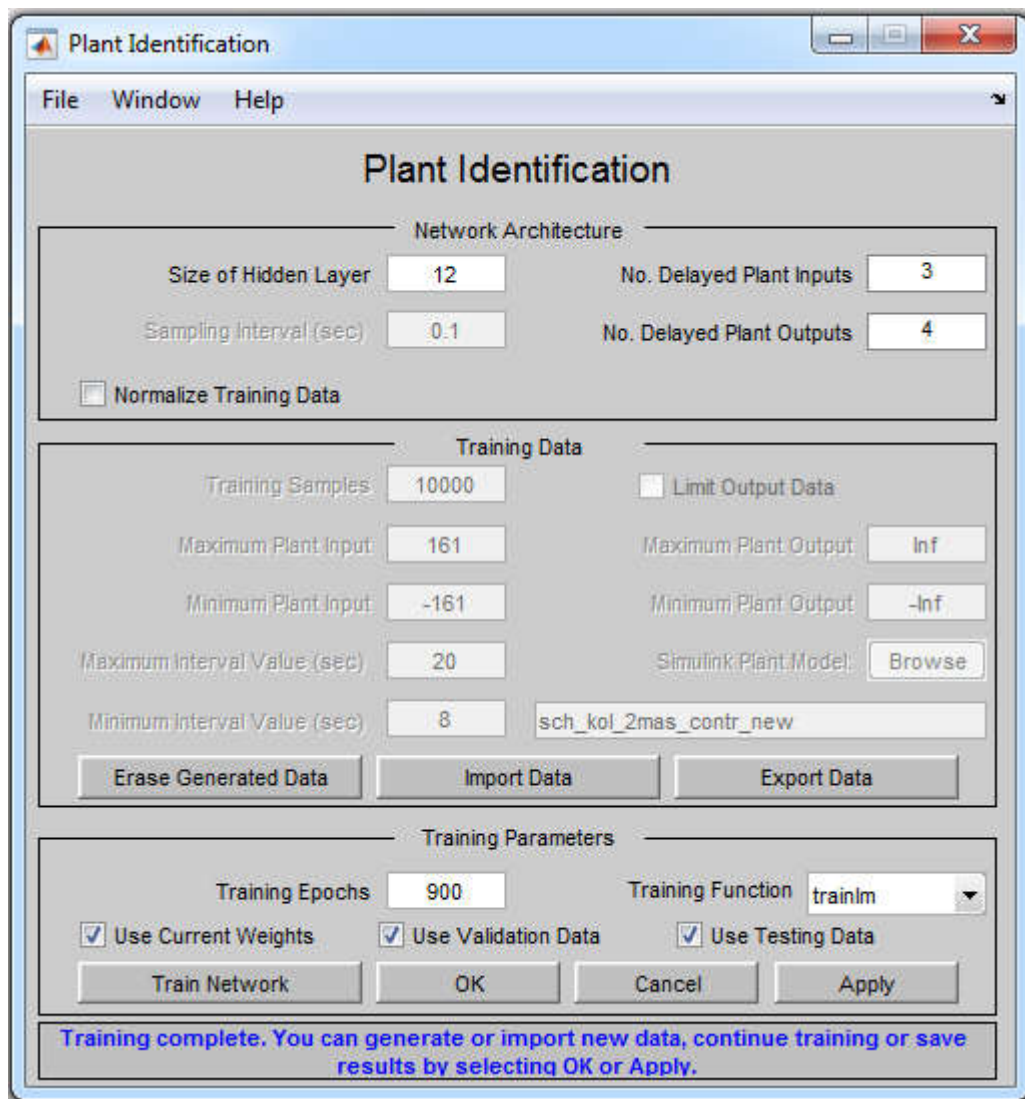


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

Елементи нейронної мережі відповідають заданим параметрам: розмір прихованого шару, кількість затримок на вході моделі та кількість затримок на виході моделі. Кожен наступний елемент стає доступним у новому вікні після подвійного клацання на попередньому елементі. На основі цих елементів у системі Simulink можна побудувати схему мережі, як показано на рис. 4.13.

Ця мережа є статичною (статичні мережі не містять елементів затримки та зворотних зв'язків). Вона використовує один вектор входу з шістьма елементами, має два шари: задана кількість нейронів у першому (прихованому) шарі та один нейрон у другому (вихідному) шарі. Активаційні функції: гіперболіч-

ний тангенс (tansig) у першому шарі та лінійна функція (purelin) у другому шарі.

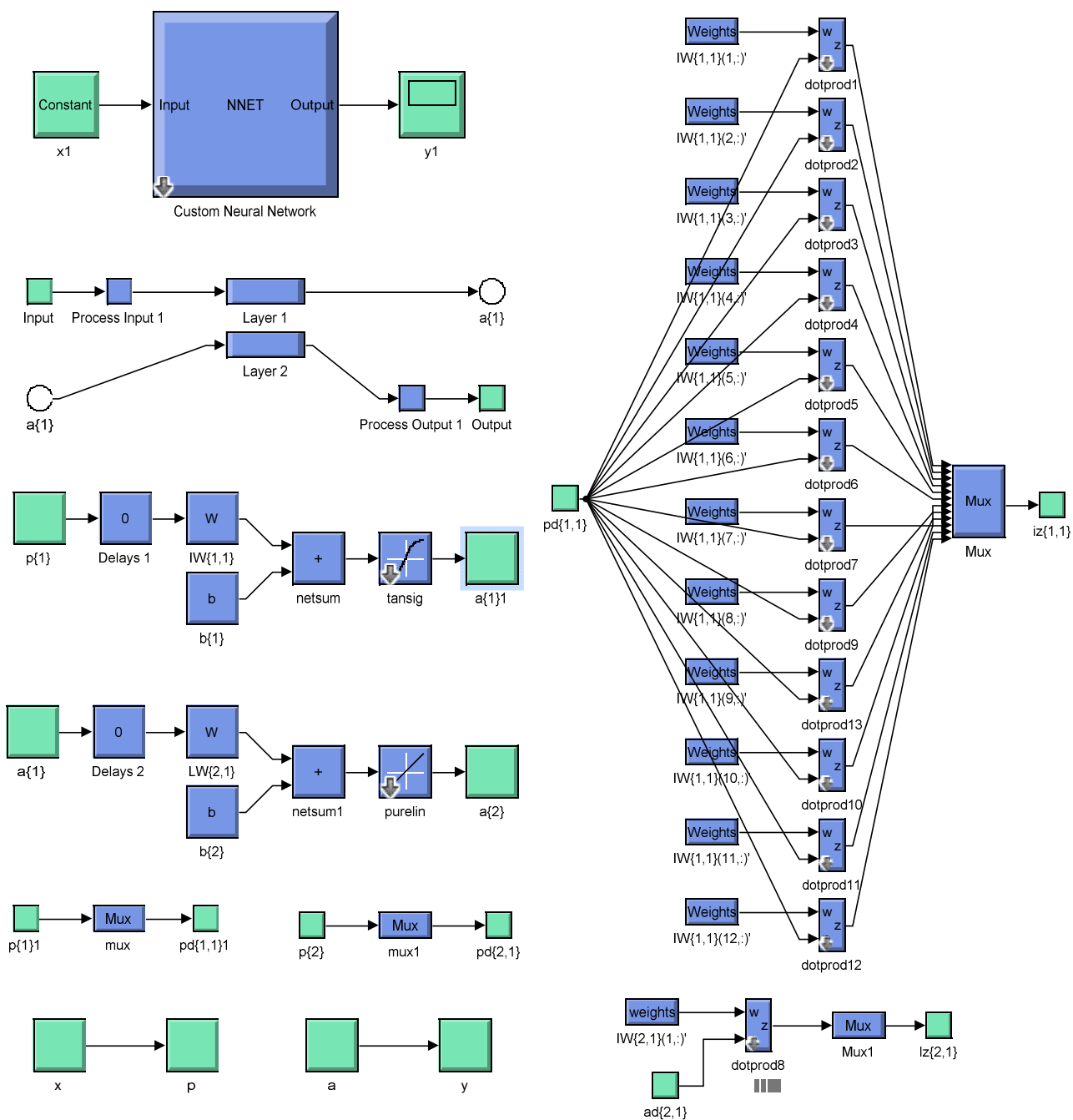


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

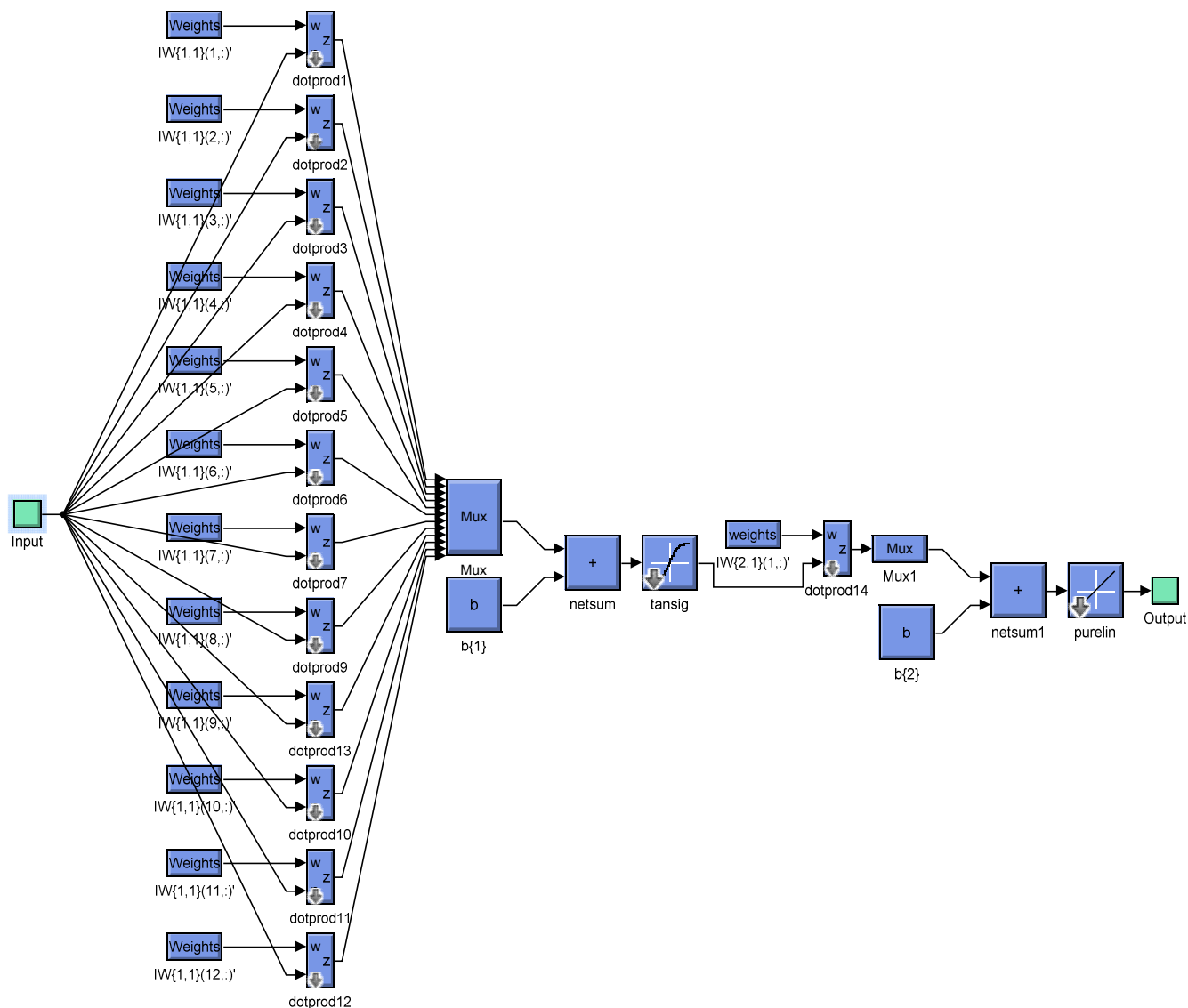


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

Після створення мережі починається процес її навчання. Процес та результати навчання наочно демонструються за допомогою діалогового вікна **Neural Network Training (nntraintool)**, показаного на рис.4.14. Наведене вікно - це інтерфейс для навчання нейронних мереж у Neural Network Toolbox в MATLAB. Воно є потужним інструментом для навчання і налаштування нейронних мереж з візуалізацією та можливістю моніторингу процесу навчання.

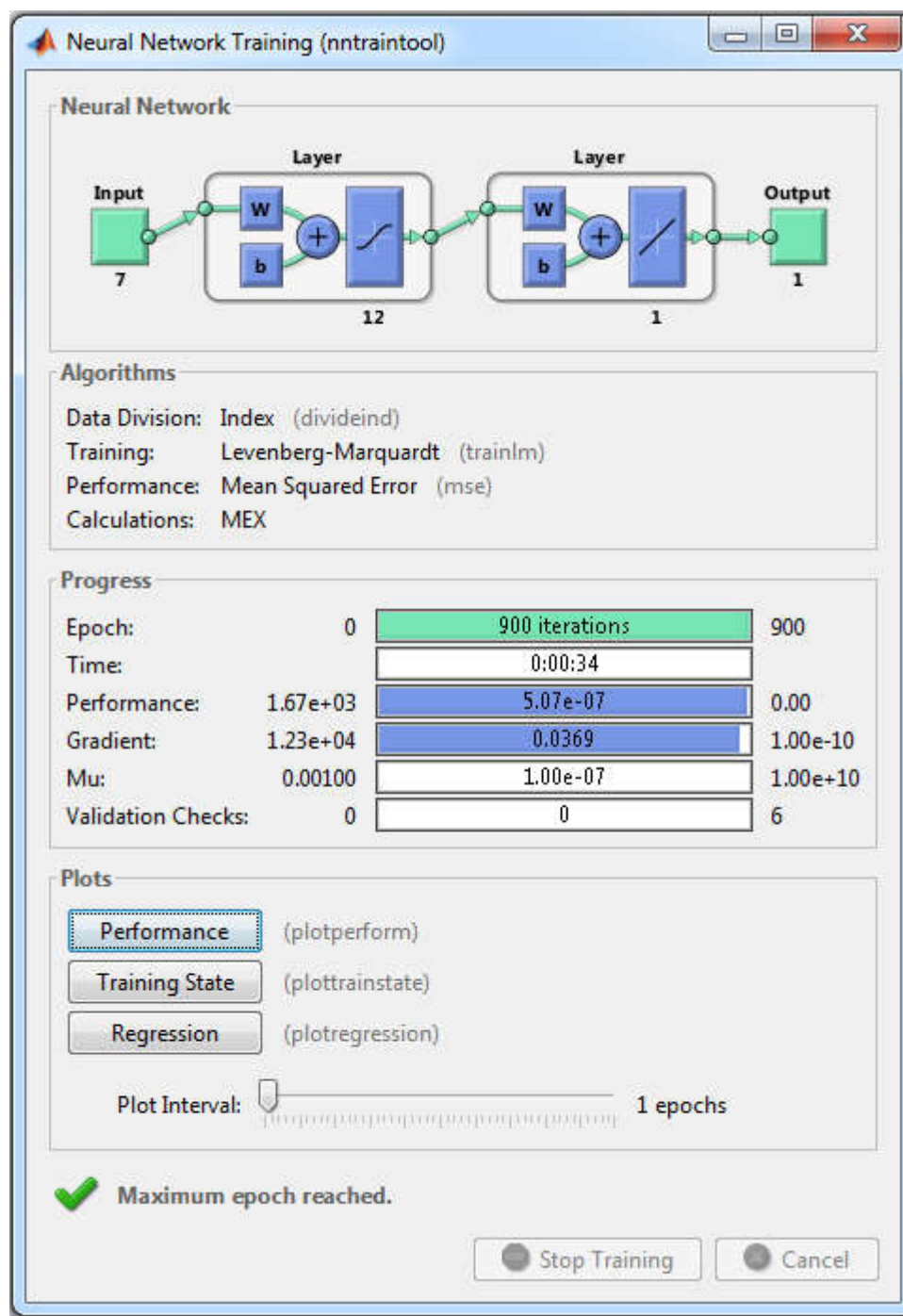


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

Секція Neural Network (нейронна мережа). У секції наводиться структура згенерованої нейронної мережі.

Секція Algorithms (алгоритми). Секція призначена для відображення різних алгоритмів та параметрів, які використовуються під час навчання нейронної мережі.

Data Division (ділення даних). Параметр Data Division: Index (поділ даних: індекс) вказує на конкретні індекси прикладів даних, які використовуються для навчання, перевірки та тестування нейронної мережі. Цей параметр використовується для визначення, як саме дані розділяються на навчальні, валідаційні та тестові набори під час навчання мережі. Якщо ділення даних не відбувається, то рядок Data Division у вікні Neural Network Training (nntraintool) відсутній.

Training (навчання). Функція навчання, яка визначає, як нейронна мережа оновлює свої ваги та зсуви під час навчання. Популярні функції включають в себе Backpropagation, Levenberg-Marquardt, Bayesian Regularization та інші.

Встановлення Training: Levenberg-Marquardt (trainlm) вказує на використання алгоритму тренування, відомого як алгоритм Левенберга-Марквардта (Levenberg-Marquardt, позначається як trainlm), для навчання нейронної мережі в середовищі MATLAB. Алгоритм Левенберга-Марквардта - це широко використовуваний метод оптимізації для навчання нейронних мереж. Цей метод комбінує в собі ідеї з методу найменших квадратів і методу Ньютона для оптимізації функції втрати мережі. Основна ідея алгоритму Левенберга-Марквардта полягає в швидкому спаді значення функції втрати і пошуку локального мінімуму шляхом оновлення вагових коефіцієнтів мережі. Він добре справляється зі складними задачами навчання, де інші методи можуть справлятися гірше.

Performance (виконання). Термін Performance вказує на показник якості навчання нейронної мережі на основі обраної функції втрат (loss function) та даних для перевірки (validation dataset). Цей показник відображає, наскільки добре нейронна мережа працює під час навчання на навчальних та валідаційних даних і допомагає визначити, коли навчання мережі має бути завершено.

Параметр Performance: Mean Squared Error (mse) вказує на використання середньоквадратичної помилки (MSE) для оцінки точності нейронної мережі під час навчання. Середньоквадратична помилка (Mean Squared Error, MSE) є однією з найпоширеніших метрик у задачах регресії. Вона обчислюється як середнє значення квадратів різниці між прогнозованими значеннями моделі (у цьому випадку нейронної мережі) та відомими або фактичними значеннями у наборі даних. Формула MSE виглядає наступним чином:

$$MSE = \frac{\sum_{i=1}^N (y_{actual_i} - y_{predicted_i})^2}{N},$$

де N - кількість прикладів даних в наборі; y_{actual_i} - фактичні значення; $y_{predicted_i}$ - прогнозовані значення, отримані від нейронної мережі.

MSE вимірює середню квадратичну відстань між фактичними і прогнозованими значеннями. Чим менше значення MSE, тим краще модель адаптується до даних. Під час навчання нейронної мережі, оптимізаційний алгоритм намагається мінімізувати MSE, тобто підганяти прогнозовані значення, щоб вони були ближчі до фактичних.

Calculations (розрахунки). Параметр відноситься до кількості обчислень або розрахунків, які виконуються під час навчання нейронної мережі. Цей параметр може бути важливим для моніторингу продуктивності та визначення часу, який потрібен для завершення процесу навчання.

Параметр **Calculations: MEX** вказує на використання MEX-функцій під час навчання нейронної мережі в середовищі MATLAB. MEX (MATLAB Executable) - це механізм, який дозволяє виконувати оптимізований обчислювальний код, написаний на C, C++ або Fortran, безпосередньо в середовищі MATLAB.

В контексті навчання нейронних мереж Calculations: MEX вказує, чи використовувати MEX-функції для підвищення швидкості обчислень під час тренування. MEX-функції можуть бути більш швидкими за інтерпретовані мови,

такі як MATLAB, і зазвичай вони використовуються для виконання обчислень на рівні мови програмування C або C++.

Секція Progress (прогрес). У секції вказується на інформація про прогрес навчання нейронної мережі під час процесу навчання. Містить інформацію, яка дозволяє в режимі реального часу відстежувати прогрес та якість навчання моделі. Це допомагає контролювати процес навчання та вживати необхідні дії для поліпшення якості моделі.

Epoch (епоха). Вказує на один повний прохід через весь навчальний набір даних під час навчання нейронної мережі. Під час кожної епохи навчання всі приклади з навчального набору даних проходять через нейронну мережу один раз, і ваги мережі оновлюються відповідно до результатів навчання на цій епохі. Епохи допомагають відстежувати процес навчання та роблять його більш керованим.

Time (час). Вказує на час, який вже було витрачено на процес навчання нейронної мережі. Цей параметр служить для відстеження тривалості процесу навчання і може бути корисним для оцінки часових аспектів навчання моделі.

Performance (виконання). Вказує на показник якості навчання нейронної мережі на основі обраної функції втрат (loss function) та даних для перевірки (validation dataset). Цей показник відображає, наскільки добре нейронна мережа працює під час навчання на навчальних та валідаційних даних. Для задач регресії зазвичай використовується середньоквадратична помилка (Mean Squared Error, MSE), яка визначає середню квадратичну помилку між прогнозованими значеннями і справжніми значеннями цільової змінної.

Gradient (градієнт). Вказує на градієнт функції втрат (loss function) відносно параметрів нейронної мережі під час навчання. Градієнт - це вектор, який вказує на те, як потрібно змінювати ваги та зсуви (biases) нейронної мережі, щоб мінімізувати функцію втрат і покращити її ефективність.

Під час навчання нейронної мережі використовується метод зворотного поширення помилки (backpropagation), і градієнт відіграє важливу роль у цьому процесі. Градієнт обчислюється для кожного параметра (ваги та зсуви) і пока-

зує, в якому напрямку і наскільки слід змінити ці параметри, щоб зменшити значення функції втрат.

Mu. Відноситься до тау-параметра, який використовується в алгоритмах оптимізації для регулювання швидкості навчання нейронної мережі. Тау-параметр (μ) використовується в контексті адаптивних методів оптимізації, таких як метод зворотного розповсюдження і метод Бройдена-Флетчера-Гольдфарба-Шанно (BFGS), для регулювання швидкості навчання. Він дозволяє алгоритму оптимізації адаптуватися до поточної динаміки процесу навчання та забезпечити збіжність навчання навіть при складних функціях втрат або підвищених градієнтах.

Зазвичай значення μ обчислюється алгоритмом оптимізації під час навчання і може змінюватися на кожній ітерації. Цей параметр є однією з ключових частин процесу оптимізації нейронної мережі та допомагає покращити збіжність та стабільність навчання.

Validation Checks (перевірки підтвердження). Вказується кількість перевірок (checks) якості моделі, які виконуються під час навчання нейронної мережі з використанням набору даних для перевірки (validation dataset). Validation Checks допомагає контролювати процес навчання та визначити, коли навчання слід припинити або зупинити, щоб уникнути перенавчання.

Основна ідея полягає в тому, що під час навчання нейронної мережі може виникнути перенавчання (overfitting), коли модель вивчає «на пам'ять» навчальний набір даних і втрачає здатність узагальнювати на нові дані. Для запобігання перенавчанню, використовуються перевірки якості моделі на окремому наборі даних, який не використовується під час навчання, а називається набором даних для перевірки (validation dataset).

Validation Checks вказують, як часто виконувати перевірки якості моделі під час навчання. Наприклад, якщо встановлено 6 перевірок, то під час навчання нейронної мережі програма буде регулярно перевіряти, наскільки добре модель працює на наборі даних для перевірки після кожних 6 епох навчання (де епоха - це один прохід через весь навчальний набір даних). Якщо якість моделі

на наборі даних для перевірки перестає покращуватися або погіршується, то це може бути індикатором перенавчання, і навчання може бути припинене.

Секція Plots (графіки). Містить графіки та візуалізацію прогресу навчання нейронної мережі під час процесу навчання. Ця секція допомагає візуально аналізувати якість та хід навчання мережі.

Кнопка **Performance** використовується для відображення показників якості під час навчання нейронної мережі. При натисканні на цю кнопку відкривається вікно **Neural Network Training Performance**, показане на рис. 4.15.

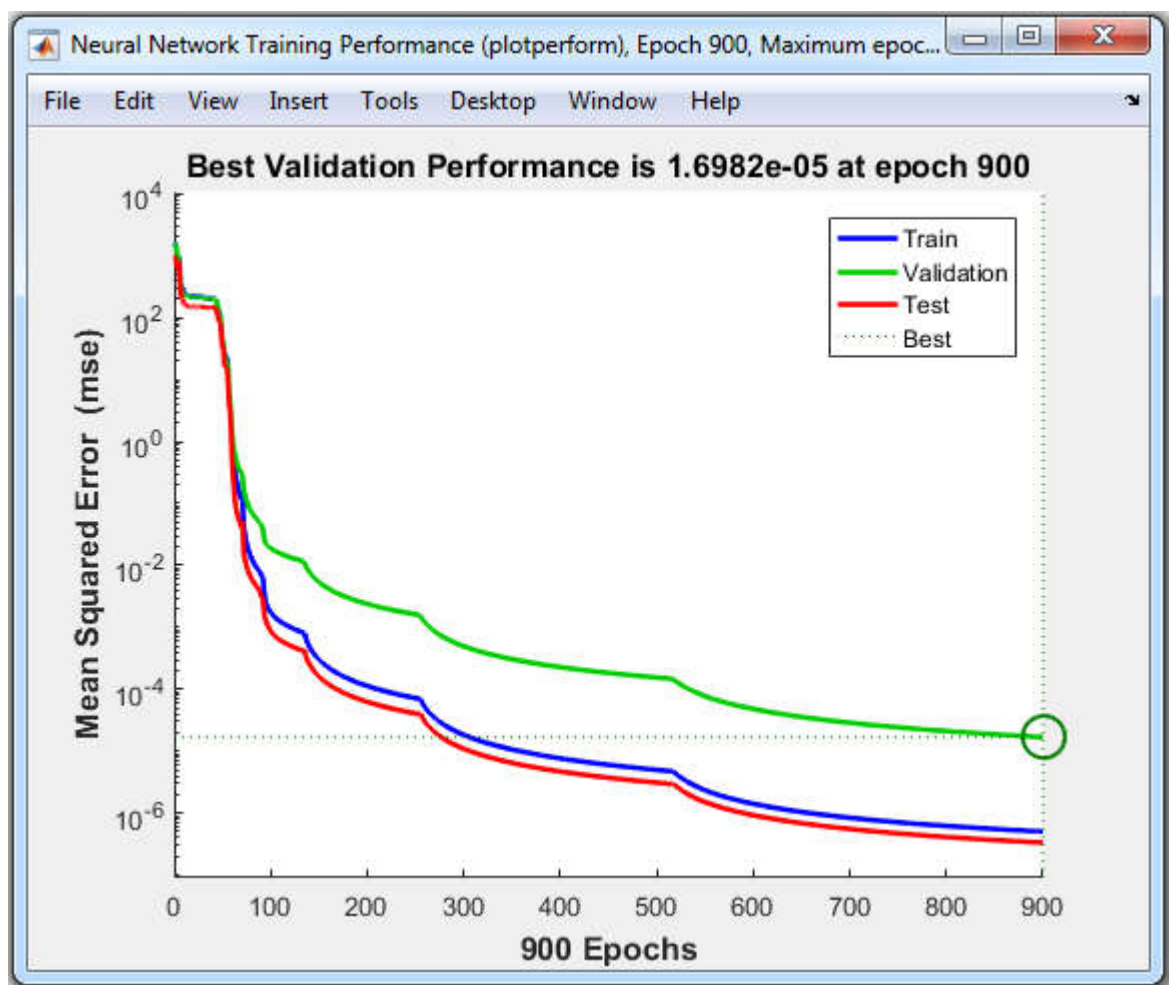


Рисунок 4.15 – Вікно контролю процесу навчання

У вікні Neural Network Training Performance відображаються графіки **Mean Squared Error (mse)**, які показують, як змінюється середньоквадратична помилка на навчальних, валідаційних (перевірочних) та тестових наборах даних

під час кожної ітерації навчання. Графіки MSE показують, як змінюється точність мережі під час навчання і чи є ознаки перенавчання (overfitting), коли помилка на валідаційних та тестових даних починає зростати. Метою є мінімізувати MSE на усіх наборах даних для досягнення найкращої моделі.

Кнопка **Training State (стан навчання)**. При натисканні на кнопку Training State відкривається вікно **Neural Network Training State** (рис. 4.16) в якому відображається інформацію про стан навчання нейронної мережі на певний момент часу під час процесу навчання.

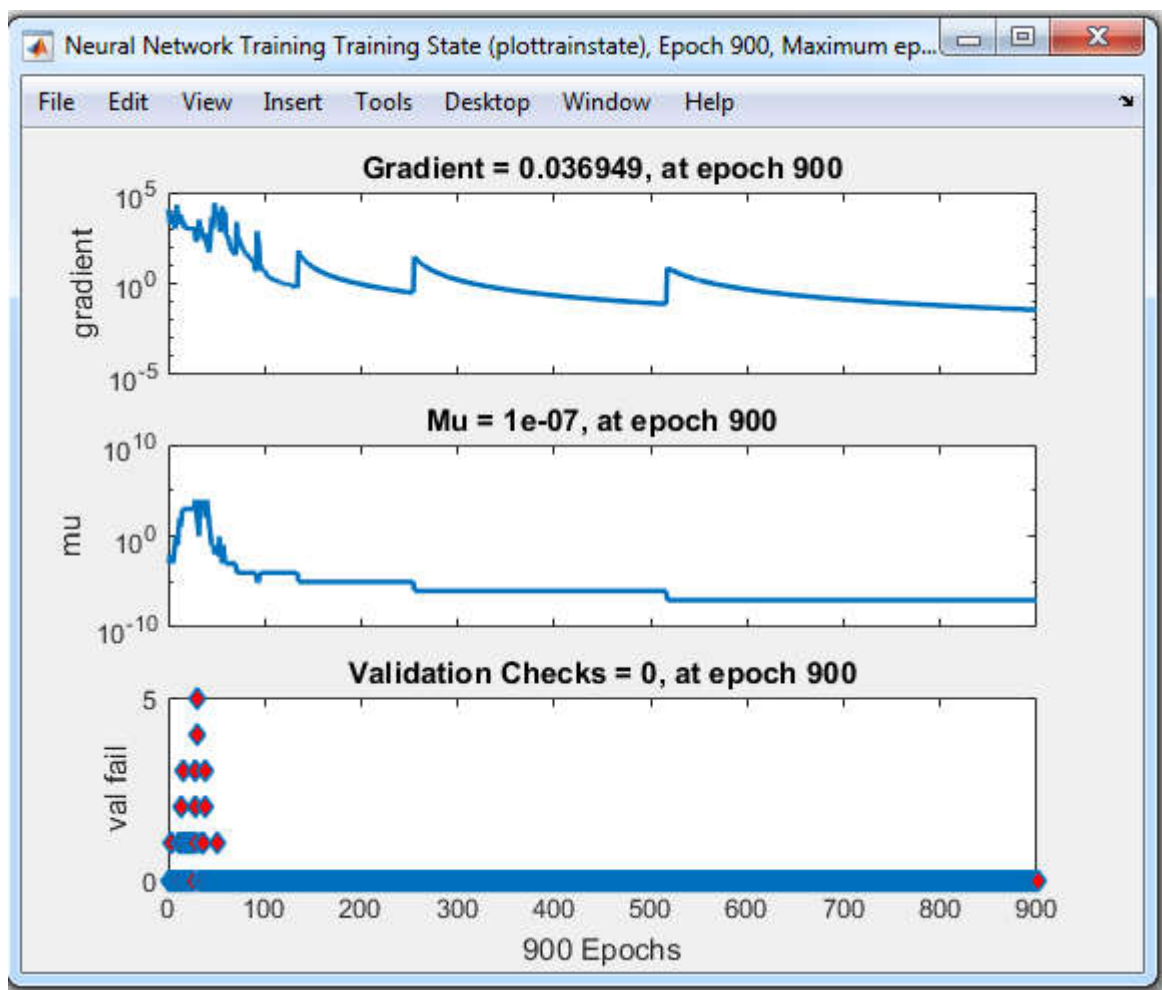


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

Графік Gradient показує зміну значення градієнта під час навчання нейронної мережі. Градієнт в контексті навчання нейронних мереж є вектором,

який вказує напрямок та міру найшвидшого зростання функції втрати (або помилки) мережі в точці навчання. Градієнт використовується для оновлення вагових коефіцієнтів мережі під час оптимізації.

Основна ідея використання градієнта полягає в тому, що шляхом зміни вагових коефіцієнтів мережі в напрямку, протилежному до градієнта, можна найбільше зменшити значення функції втрати. Градієнт допомагає визначити, в якому напрямку і наскільки потрібно вносити зміни в вагові коефіцієнти для покращення продуктивності мережі.

На графіку Gradient відображається, як змінюється величина градієнта під час процесу навчання. Зазвичай, мережа спочатку навчається з великими градієнтами, і з часом ці градієнти зменшуються, коли мережа наближається до оптимальних вагових коефіцієнтів.

Моніторинг графіка Gradient може бути корисним для визначення, чи існують проблеми з градієнтним зникненням або вибуванням (gradient vanishing/exploding), а також для визначення, чи виникають проблеми зі збіжністю навчання. Великі або надто малі градієнти можуть вказувати на необхідність оптимізації параметрів навчання.

Графік μ показує зміну значення параметра μ під час навчання нейронної мережі. Параметр μ відноситься до швидкості навчання та регуляризації мережі, і його зміна вказує на те, як адаптується швидкість навчання під час тренування.

Зазвичай, під час навчання нейронної мережі використовуються методи оптимізації, такі як стохастичний градієнтний спуск (SGD) або його модифікації. Параметр μ в цьому контексті часто вказує на швидкість навчання (learning rate) та регуляризацію.

Зміна μ показує, як модифікується швидкість навчання під час процесу навчання. Зазвичай, μ може збільшуватися або зменшуватися в залежності від реакції нейронної мережі на дані. Наприклад, велике значення " μ " може вказувати на те, що швидкість навчання збільшилася для прискорення збіжності,

але при цьому може виникнути ризик нестабільності. Зменшення ти може вказувати на те, що навчання сталося занадто швидко або навіть зупинилося.

Моніторинг графіка ти може бути корисним для контролю за швидкістю навчання та стабільністю процесу навчання. Великий вміст інформації в графіку ти може вказувати на те, що вибрані параметри навчання потребують подальшої налаштування.

Графік Validation Failure (або val fail) (помилка підтвердження) показує, як змінюється величина, що відображає кількість невдач (помилки) під час перевірки (валідації) нейронної мережі під час навчання. Головною метою валідації є визначення, наскільки добре навчена мережа узгоджується з набором даних, які не використовувалися під час навчання, і чи існують ознаки перенавчання. Зазвичай, мінімізація цього показника є однією з цілей тренування мережі. Якщо кількість помилок на валідації росте, це може бути ознакою перенавчання, і може бути необхідно прийняти заходи для покращення продуктивності мережі, такі як зупинка навчання раніше або зміна гіперпараметрів.

Продовження опису вікна Neural Network Training (nntraintool)

Кнопка **Regression (регресія)** використовується для вибору та налаштування параметрів навчання нейронної мережі для завдань регресії. Як відомо, завдання регресії - це один з типів завдань в машинному навчанні, де основною метою є передбачення числових значень або кількісних величин на основі вхідних даних. У регресійних завданнях модель намагається знайти функціональну залежність між вхідними ознаками (параметрами) і вихідними числовими значеннями, які зазвичай називаються цільовими або відгуком. .

При натисканні на кнопку Regression відкривається вікно **Neural Network Training Regression** (рис. 4.17), у якому показані графіки та інформація, які допомагають в оцінці та аналізі продуктивності нейронної мережі під час навчання. Для наочної оцінки якості навчання приведені графіки регресії для Training, Validation та Test наборів даних.

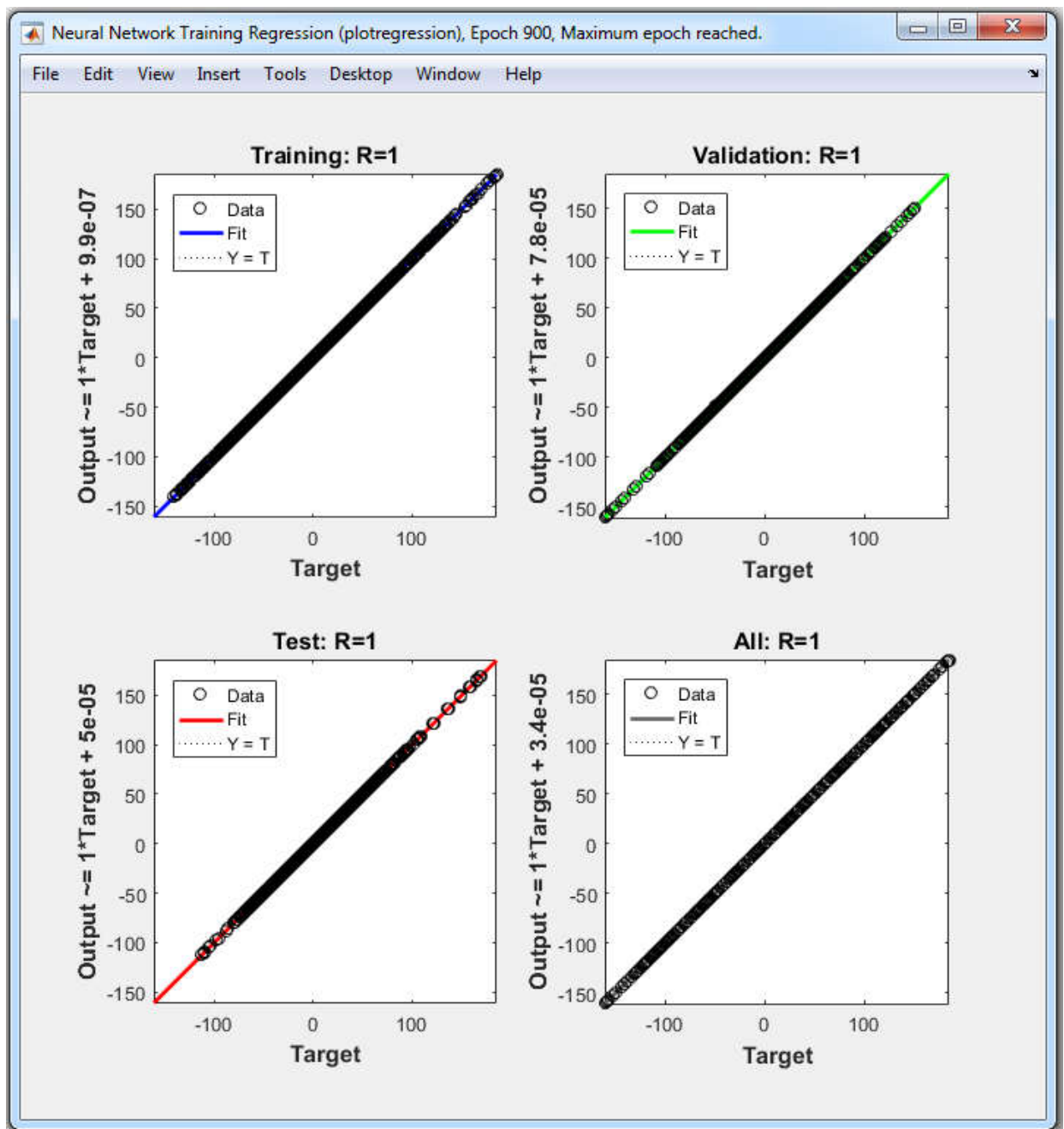


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис вікна Neural Network Training Regression

Графіки **Training** показують, наскільки точно модель відповідає навчальним даним, тобто даним, на яких вона була навчена.

Графіки **Validation** демонструють, наскільки точно модель працює на наборі даних для перевірки. Дані для перевірки використовуються для валідації результатів моделі та визначення, наскільки вона загальніє на нових даних.

Графіки **Test** включають тестові дані, які незалежно від використовувалися для оцінки точності моделі на даних, які модель раніше не бачила. Тестові дані надають об'єктивну оцінку роботи моделі на нових даних.

Графіки **All** показують регресійні результати для всіх трьох наборів даних: Training, Validation та Test. Графіки All об'єднують і відображають прогнози моделі та фактичні значення для всіх трьох наборів даних на одному графіку з метою порівняння. Ці графіки забезпечують користувача загальним зоровим порівнянням точності моделі на різних наборах даних і допомагають визначити, чи існують різниці в її поведінці на тренувальних, валідаційних та тестових даних.

Графіки **Data (дані)** для Training, Validation, Test та All даних у вікні Neural Network Training Regression представляють собою графічну інформацію про те, як навчальна модель нейронної мережі відповідає даним на різних наборах даних. Ці графіки показують на горизонтальній вісі фактичні значення (дані з набору даних), а на вертикальній вісі - відповідні прогнозовані значення, отримані від моделі нейронної мережі, тобто кожній парі з тренувальних, валідаційних та тестових даних відповідає одна точка (у вікні Neural Network Training Regression позначено кружечками) з координатою по осі абсцис, яка дорівнює бажаному значенню, а по осі ординат – значенню на виході штучної нейронної мережі. Якщо всі точки лежать близько до ідеальної (діагональної) лінії (це графіки $Y = T$, позначені точками у вікні Neural Network Training Regression), то це означає, що модель добре передбачає дані. Якщо є велика розсіюваність точок, це може свідчити про недосконалість моделі або шум у даних.

Графіки **dat** допомагають візуально оцінити точність моделі на трьох різних наборах даних і визначити, наскільки добре модель може узгоджуватися з фактичними даними. Вони можуть бути корисними для виявлення аномалій, оцінки загальної продуктивності моделі і визначення необхідності подальшої оптимізації.

Графіки **Fit (прогноз)** – це графіки лінійної регресії, які відображають прогнозовані значення, які надає нейронна мережа на основі вхідних даних. Гра-

фіки Fit демонструють відповідність прогнозованих значень, отриманих від моделі, фактичним значенням з наборів даних. Вони допомагають візуально оцінити, наскільки точно модель регресії передбачає числові значення на основі вхідних даних. Якщо прогнозована лінія добре відповідає фактичним даним і проходить через багато точок, це свідчить про високу точність моделі. Напротив, якщо лінія регресії відхиляється від фактичних значень, це може вказувати на неадекватність моделі і необхідність налаштування параметрів нейронної мережі.

Вирази для лінійної регресії, а також значення коефіцієнта кореляції R наведені у вікні Neural Network Training Regression для кожного набору даних: Training, Validation, Test та All. Якщо графік Fit добре узгоджується з графіком $Y=T$, то це свідчить про високу точність моделі регресії.

Коефіцієнт кореляції R у вікні Neural Network Training Regression (plotregression) є мірою кореляції між фактичними та прогнозованими значеннями у завданні регресії. Цей коефіцієнт допомагає визначити, наскільки добре модель регресії відповідає фактичним даним і наскільки точним є її прогноз.

Зазвичай, коефіцієнт кореляції R може приймати значення в діапазоні від -1 до 1. Його інтерпретація наступна:

- Коефіцієнт кореляції R дорівнює 1: Це означає, що прогнозовані значення і фактичні значення абсолютно лінійно залежні одне від одного, і модель регресії ідеально передбачає фактичні значення.
- Коефіцієнт кореляції R дорівнює 0: Це вказує на жодну лінійну залежність між прогнозованими і фактичними значеннями, і модель не може передбачити дані з жодною точністю.
- Коефіцієнт кореляції R дорівнює -1: Це означає, що прогнозовані значення і фактичні значення є лінійно залежними, але зворотним способом, і модель регресії передбачає їх ідеально, але з зміненим напрямком.

Кореляційний коефіцієнт R близький до 1 вказує на високу точність моделі регресії, тоді як значення близьке до 0 або від'ємне вказує на низьку точ-

ність. Цей коефіцієнт дуже корисний для візуальної оцінки точності моделі та порівняння прогнозованих та фактичних значень у завданнях регресії.

Продовження опису вікна Neural Network Training (nntraintool)

Plot Interval (інтервал графіка). Цей параметр контролює частоту оновлення графіків та відображення результатів під час навчання нейронної мережі для задачі регресії. Plot Interval визначає, через скільки ітерацій навчання (або epoch) має бути відображена інформація на графіках.

Зазвичай вибір Plot Interval допомагає контролювати кількість інформації, яка відображається під час навчання, особливо якщо навчання триває багато epoch і графіки можуть бути дуже завантажені. Встановлення меншого значення Plot Interval означатиме, що графіки оновлюватимуться частіше, що може бути корисним для більш детального аналізу, але збільшить обсяг відображеної інформації. Навпаки, встановлення більшого значення Plot Interval зменшить частоту оновлень графіків і зменшить обсяг відображеної інформації.

Зазвичай, налаштування Plot Interval є важливою опцією під час навчання моделі для того, щоб зручно аналізувати процес навчання та вплив варіацій параметрів моделі на результати.

Кнопки **Stop Training** і **Cancel** у вікні Neural Network Training (nntraintool) в середовищі MATLAB використовуються для керування процесом навчання нейронної мережі, але вони мають різні функції:

Кнопка **Stop Training (зупинка навчання)** призначена для негайної зупинки процесу навчання нейронної мережі. Вона припиняє навчання без можливості продовження або збереження поточного стану моделі. Використовується, коли ви бачите, що навчання триває дуже довго, ви визнали, що навчання не дає задовільних результатів, або ви просто хочете негайно припинити процес навчання з будь-якої причини.

Кнопка **Cancel (скасувати)** використовується для припинення процесу навчання, але зазвичай вона надає можливість підтвердження та збереження поточного стану моделі. Вона дозволяє вам контролювати ітерації навчання,

перевіряти результати, змінювати параметри та вирішувати, коли припинити навчання з можливістю збереження результатів навчання.

Після завершення навчання результати відображаються на графіках у вікні **Training data for NN Predictive Control**, яке показано на рис.4.18.

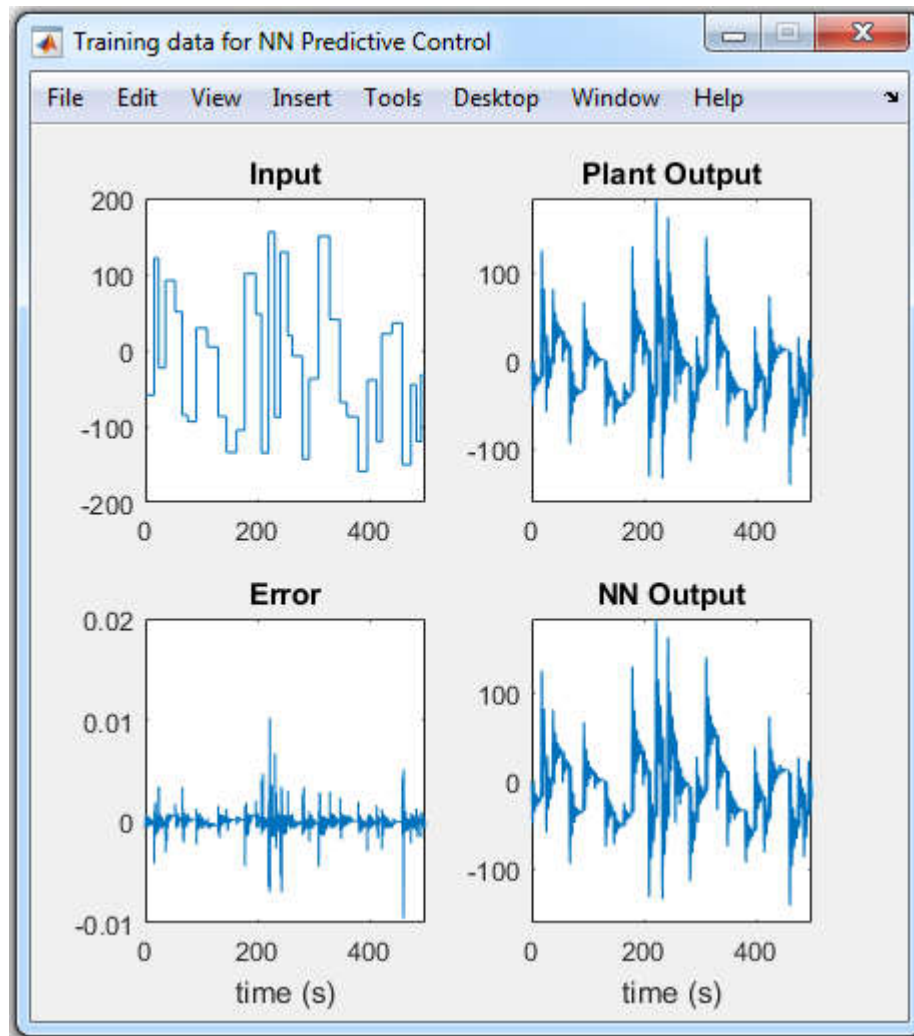


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

Input (вхід). Графік Input відображає вхідні сигнали, які подаються на систему і нейромережеву модель під час навчання для генерації відповіді.

Plant Output (вихід системи). Графік Plant output відображає фактичні вихідні дані системи (Plant) під час навчання. Ці дані представляють реальну динаміку системи під впливом вхідних сигналів.

NN Output (вихід нейромережі). Графік NN Output відображає вихідні дані, що згенеровані нейромережевою моделлю під час навчання. Ці дані представляють прогнозовану відповідь моделі, яка намагається відтворити динаміку системи.

Error (Помилка). Графік Error показує різницю між прогнозованими відповідями нейромережевої моделі (NN Output) і фактичними відповідями системи (Plant Output). Ця помилка вказує на те, наскільки добре модель відтворює динаміку системи під час навчання.

На рис. 4.19, 4.20 показані вікна **Validation data for NN Predictive Control** і **Testing data for NN Predictive Control**, в яких наведені аналогічні графіки, побудовані в результаті перевірки на контрольній і текстовій множині.

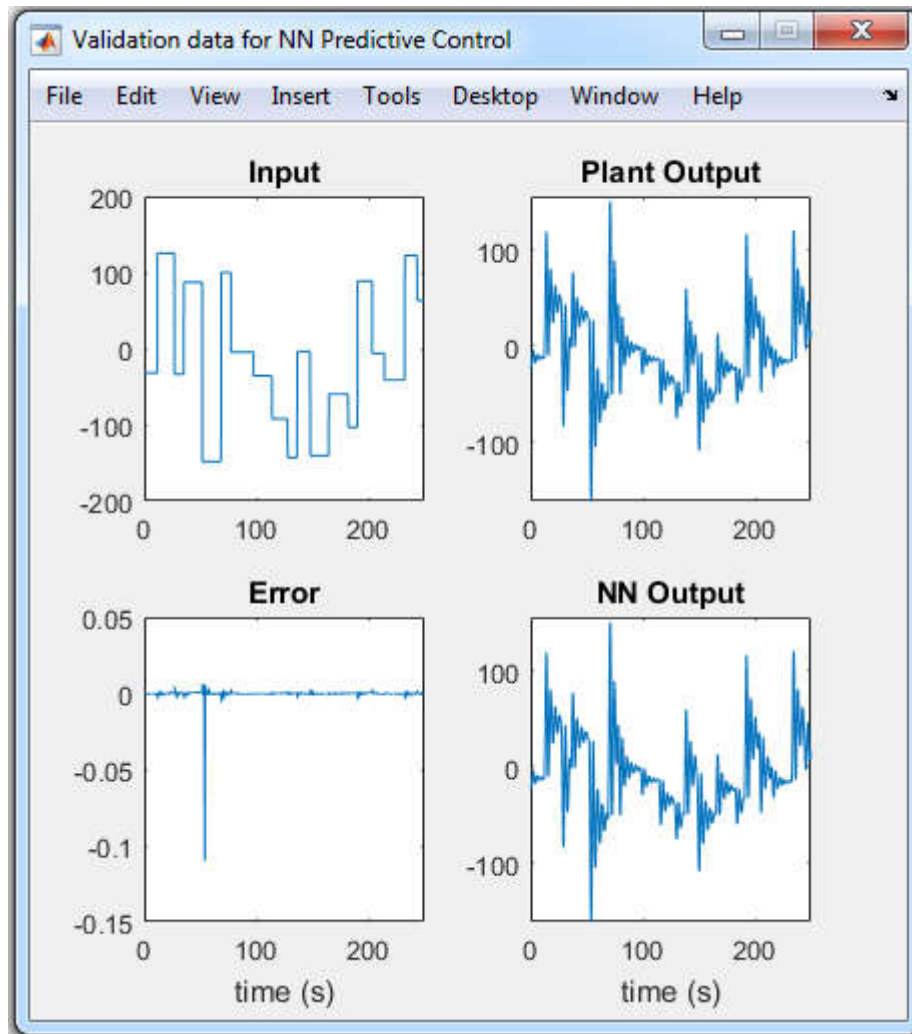


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

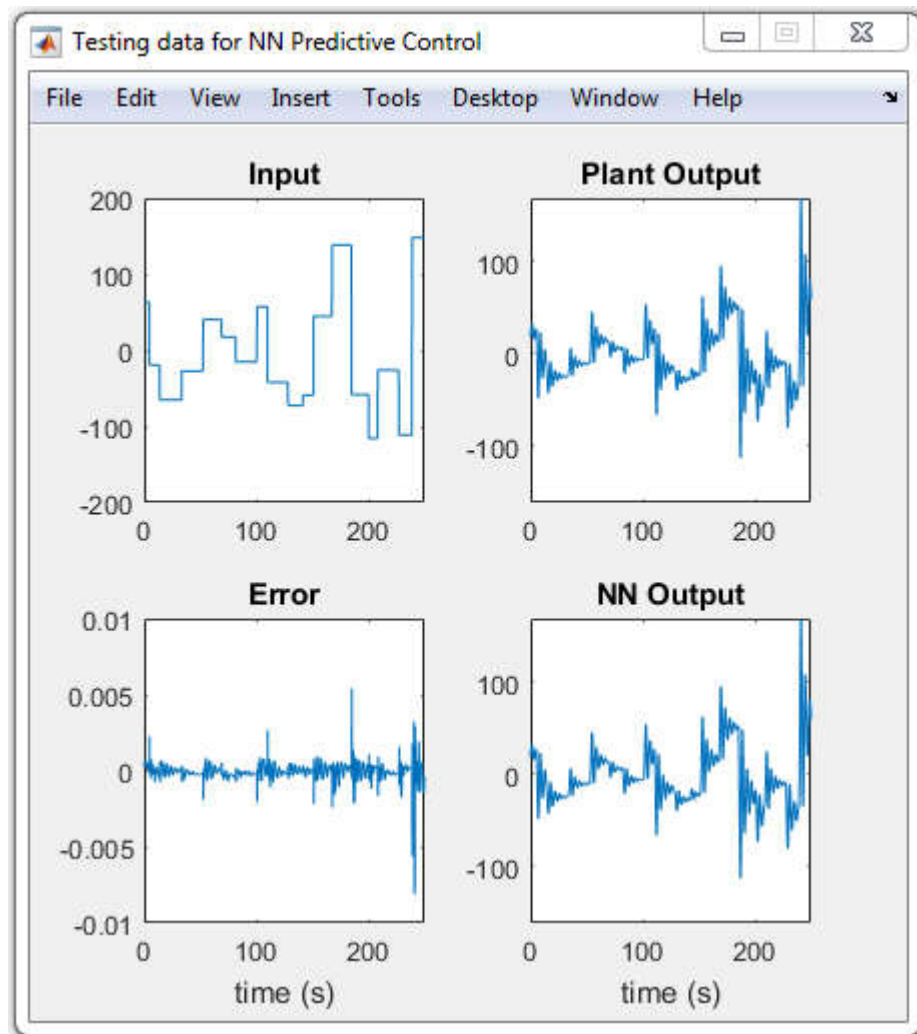


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

Поточний стан відмічений у вікні Plant Identification повідомленням *«Training complete. You can generate or import new data, continue training or save results by selecting OK or Apply»* (Навчання завершено. Можна згенерувати або імпортувати нові дані, продовжити навчання або зберегти отримані результати, вибравши кнопки OK або Apply).

Після завершення навчання М-функція **Nnident** формує динамічну мережу **netn2** із визначеною кількістю затримок на вході та виході моделі, не змінюючи при цьому набутих значень вагів і зсувів нейронів шарів. Згорнута схема динамічної мережі і моделі елементів мережі показані на рис.4.21.

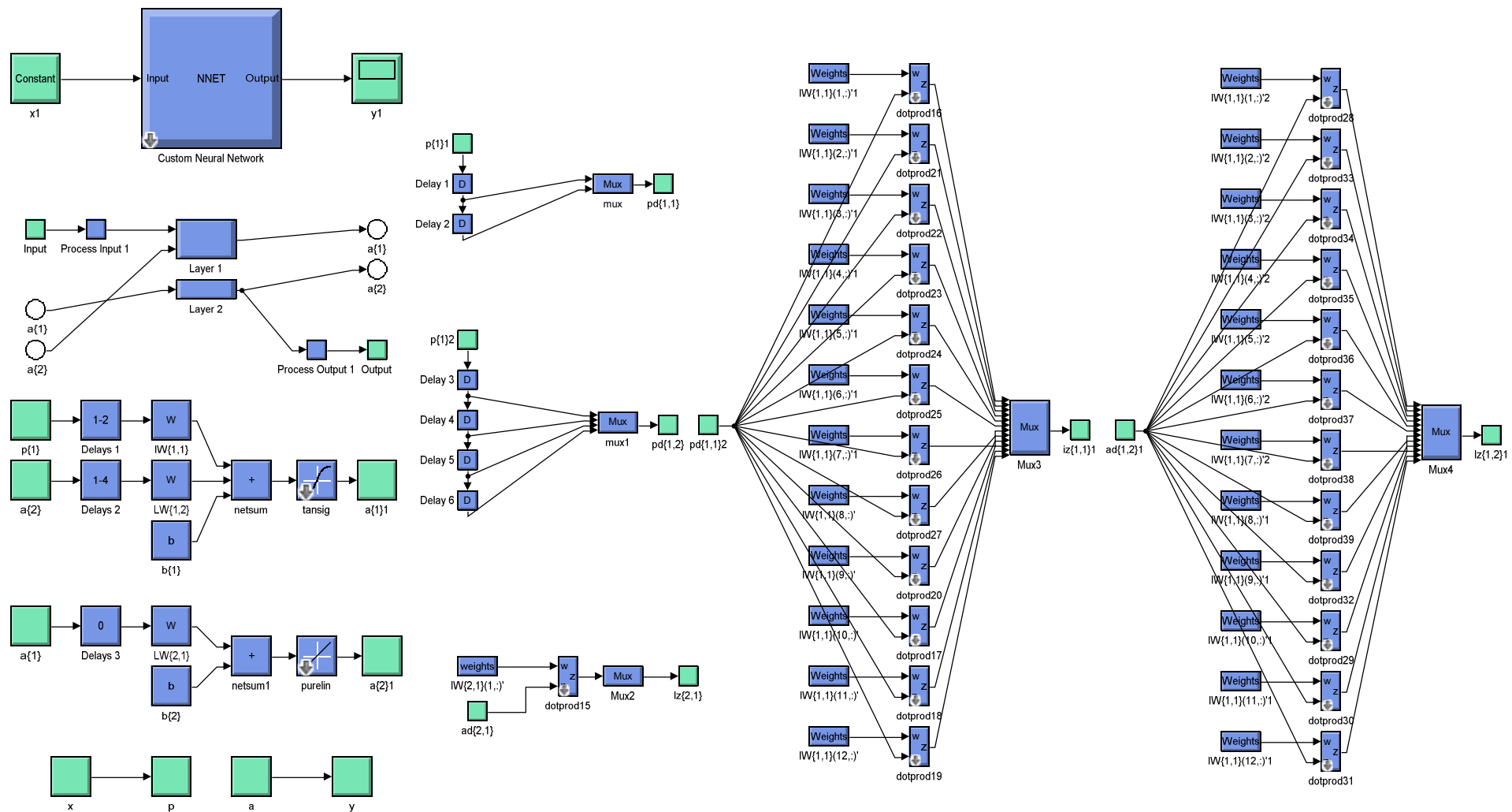


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

Модель динамічної нейронної побудована в системі Simulink за допомогою оператора gensim(netn2). Кожен подальший елемент з'являється в окремому вікні при активізації попереднього подвійним клацанням миші. З даних елементів в системі Simulink побудована схема динамічної мережі, показана на рис.4.22.

Слід зазначити, що для побудови моделей мереж netn і netn2 треба перервати виконання програми шляхом попереднього встановлення **Breakpoint** у відповідному місці функції Nnident.m, оскільки після закінчення виконання зазначеної функції побудова мереж стає неможливою.

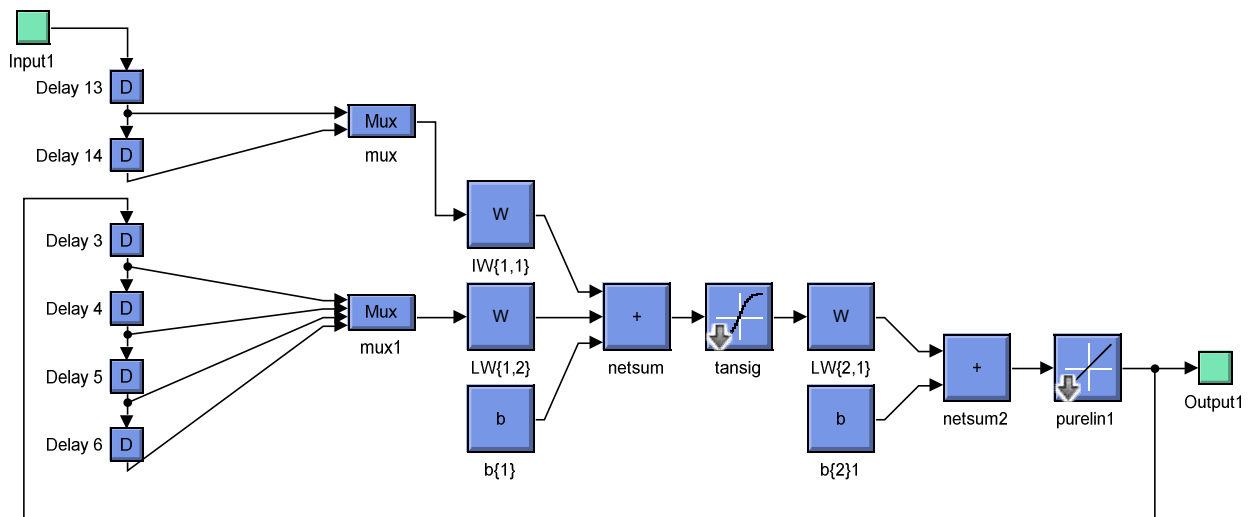


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

В результаті параметри нейромережевої моделі керованого об'єкту вводяться в блок NN Predictive Controller системи Simulink. У системі Simulink дана мережа представляється у вигляді структурної схеми, показаної на рис.4.23.

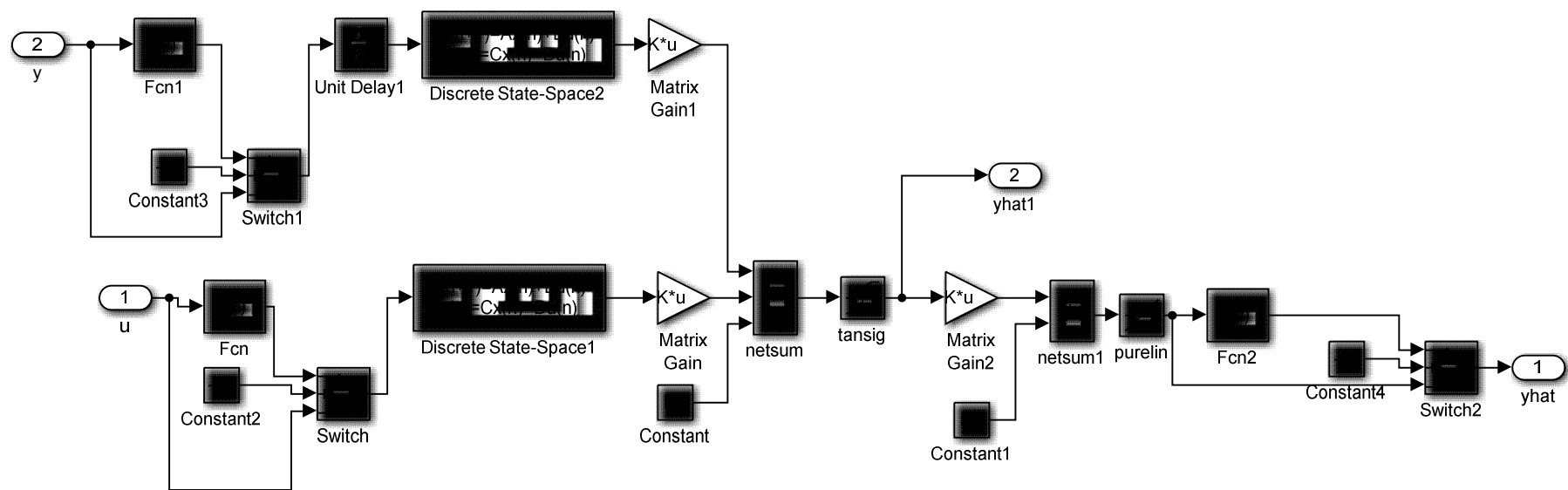


Рисунок 4.23 – Структурна схема нейронмережевої моделі об'єкту регулювання

Блоки Matrix Gain і Matrix Gain 1 відповідають матрицям $IW\{1,1\}$ і $LW\{1,2\}$ відповідно. Блоки Constant (B1) Constant1 (B2) відносяться до зсувів нейронів першого і другого шарів. Елементи затримок моделюються за допомогою блоків Discrete State Space 1 і Discrete State Space 2

$$\mathbf{y}(n) = \mathbf{C}\mathbf{x}(n) + \mathbf{D}\mathbf{u}(n) ;$$

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{B}\mathbf{u}(n).$$

Для випадку $N_i = 2$ і $N_j = 5$ чисельні значення матриць $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ і $\mathbf{D}$ вказаних блоків наступні

Discrete State Space 1

$$\mathbf{A} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}.$$

Discrete State Space 2

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \mathbf{D} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

У системі Simulink формується також схема ptest3sim2, показана на рис.4.24. Після активізації блоку Subsystem відкривається вікно з схемою рис.4.25. Дана схема теж є нейромережевою моделлю об'єкту управління, що має додаткові виходи, і використовується М-функцією predopt для прогнозу процесу в майбутньому.

Перетворена структурна схема нейрорегулятора, отримана в результаті об'єднання схем рис.4.6 і рис.4.23 показана на рис 4.26.

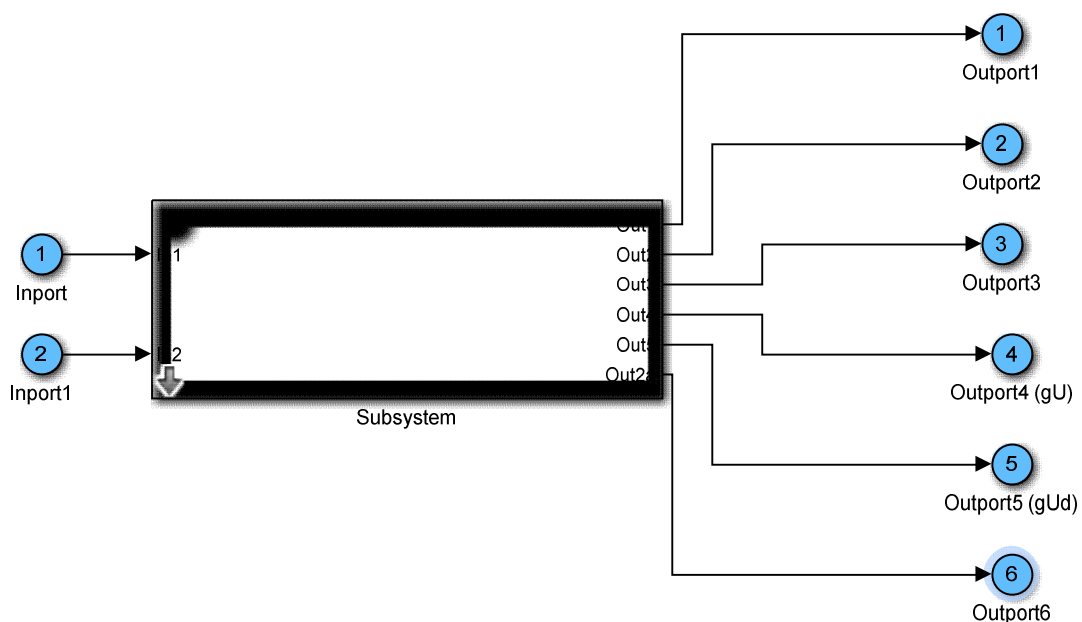


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

Після створення нейромережевої моделі керованого об'єкта здійснюється повернення до вікна **Neural Network Predictive Controller** (рис. 4.7), де задаються параметри оптимізації.

Продовження опису вікна Neural Network Predictive Controller

Const Horizon (N2). Верхня межа підсумовування в показнику якості (нижня межа N_1 фіксована і рівна 1). Цей параметр визначає горизонт прогнозу, тобто кількість кроків у майбутньому, на які алгоритм прогнозованого керування буде робити прогнози. Вибір значення Const Horizon (N2) важливий, оскільки він впливає на те, наскільки точно і передбачувано система буде керована за допомогою алгоритму NNPC.

Зазвичай, збільшення значення Const Horizon (N2) дозволяє отримати більший горизонт прогнозу, що означає більш далекі випереджальні прогнози, але водночас може збільшити обчислювальну складність і затримки у системі керування. Обране значення "Const Horizon" повинно бути ретельно підібрано з урахуванням конкретних вимог і характеристик вашого керовального завдання і системи.

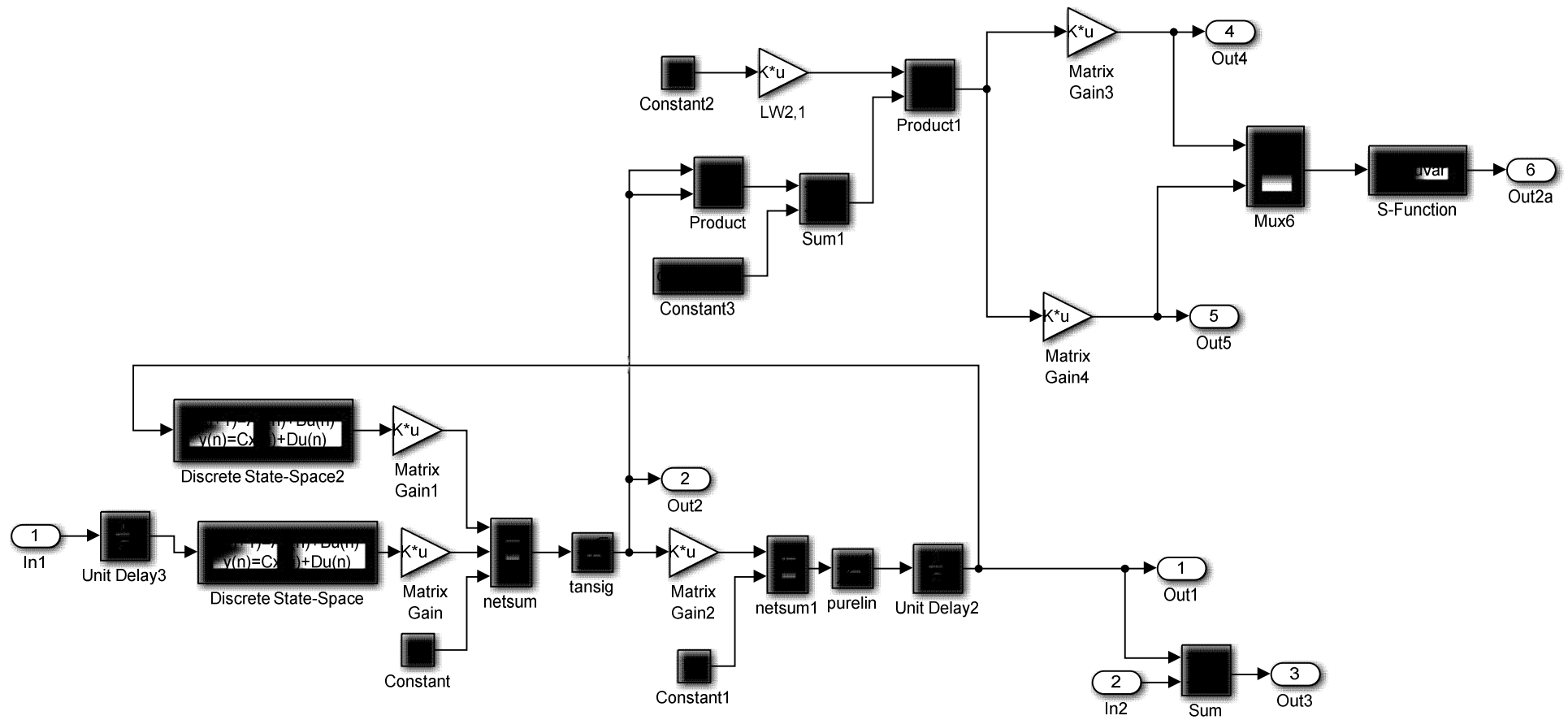


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

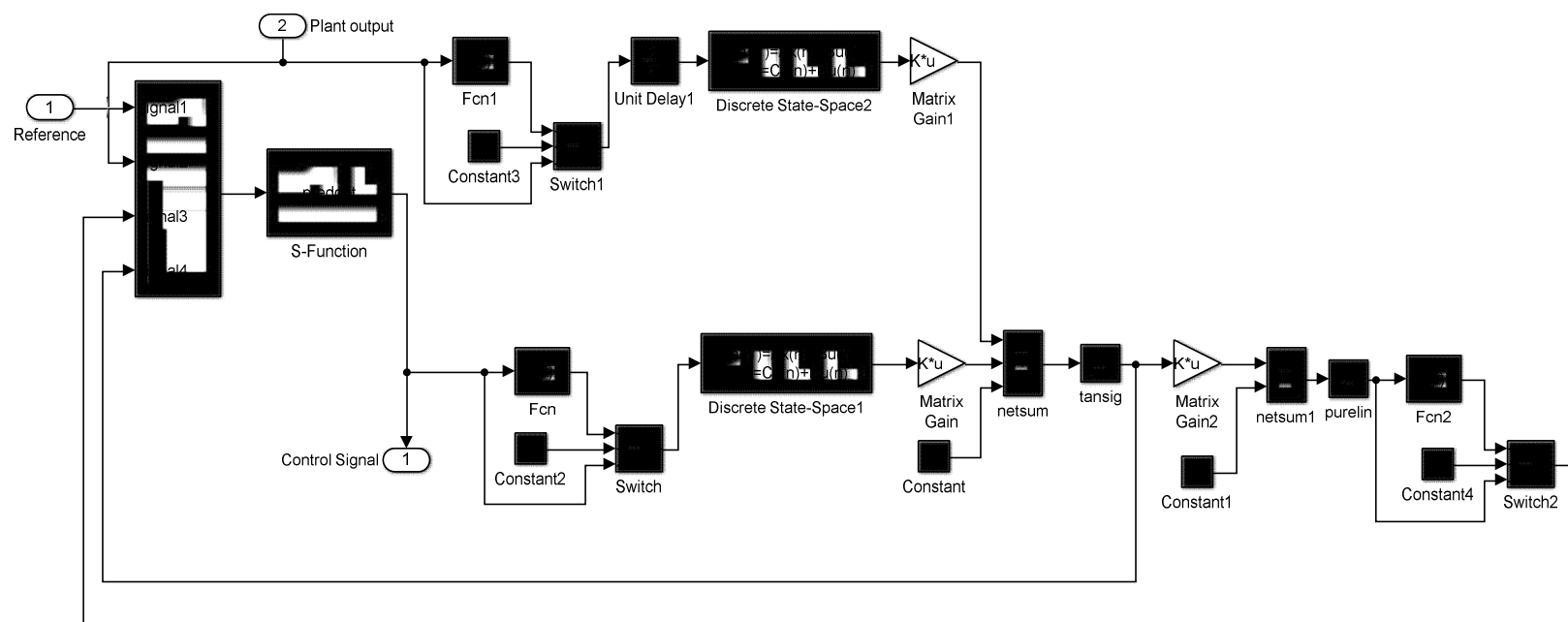


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

Control Horizon (Nu). Верхня межа підсумовування при оцінці потужності управління N_u . Цей параметр грає важливу роль при плануванні оптимального керування системою, і він визначає кількість кроків у майбутньому, на які буде розрахована потужність управління в рамках контролю.

Основна ідея цього параметру полягає в тому, що велике значення Control Horizon (Nu) дозволяє оптимізаційному алгоритму керування урахувати більшу кількість кроків у майбутньому при плануванні керування. Це може бути корисним для досягнення бажаного стану системи та забезпечення стабільності, а також враховувати обмеження на керування.

Однак велика кількість кроків в Control Horizon (Nu) також може призвести до складніших обчислювальних задач і збільшити вимоги до ресурсів, тому вибір значення параметра Control Horizon (Nu) повинен бути обґрунтованим і базуватися на конкретних вимогах та характеристиках системи керування.

Control Weighting Factor (ρ). Коефіцієнт ваги для потужності управління. Цей параметр дозволяє вам налаштувати важливість мінімізації керуючих сигналів в обчисленнях контролера.

Контролер має оптимізувати вихідні сигнали системи, забезпечуючи бажаний режим роботи. Однак інколи деякі сигнали керування можуть бути дорогою або необхідною величиною. Наприклад, в деяких ситуаціях, зниження витрат на керування може бути важливим завданням.

Control Weighting Factor визначає, як обчислювати вартість або функцію витрат управління під час оптимізації. Якщо цей фактор великий, то контролер надає велику вагу мінімізації витрат на керування, і, отже, старається робити керуючі сигнали якнайбільш економічними. Зменшення фактора зробить керуючі сигнали менш важливими у функції витрат, і контролер буде акцентуватися на досягненні бажаного режиму роботи системи.

Користувачі можуть регулювати Control Weighting Factor в залежності від своїх конкретних вимог та цілей управління. Цей параметр дозволяє балансувати між досягненням бажаного результату і вартістю управління в контексті нейронного мережевого прогнозного управління.

Search parameter (α). Параметр одновимірного пошуку, задаючий поріг зменшення показника якості. Цей параметр використовується для контролю процесу оптимізації і для визначення, коли оптимізація може бути завершена, оскільки зменшення показника якості досягло заданого порогу.

Одновимірний пошук (або одновимірна оптимізація) - це метод оптимізації, де здійснюється пошук оптимального значення одного параметра, тримаючи інші параметри незмінними. В контексті контролера з використанням нейронної мережі це може бути важливим для налаштування певних параметрів контрольної стратегії, які впливають на якість та ефективність керування.

Поріг зменшення показника якості вказує на те, наскільки мінімальне зменшення величини, яку слід оптимізувати, вважати прийнятним для завершення процесу оптимізації. Якщо зменшення показника якості стає меншим за встановлений поріг, то оптимізація завершується, і визначені параметри вважаються оптимальними.

Цей параметр дозволяє контролеру зупинити оптимізацію, коли досягнуто задовільного рівня покращення показника якості, що може бути корисним для зекономлення обчислювальних ресурсів та забезпечення швидкої збіжності оптимізації.

Minimization Routine. Вибір процедури одновимірного пошуку.

Цей параметр визначає, який алгоритм оптимізації буде використовуватися для налаштування параметрів контрольної стратегії. Вибір підходящого алгоритму оптимізації може суттєво впливати на якість та ефективність контрольної системи. Вибір конкретного "Minimization Routine" залежить від характеристик завдання керування, обмежень, доступності обчислювальних ресурсів та інших факторів.

Iterations Per Sample Time. Параметр визначає кількість ітерацій оптимізації, які виконуються протягом кожного відрізка часу (Sample Time) при розрахунку оптимального керування системою. Цей параметр дозволяє встановити, скільки ітерацій оптимізації виконується в межах кожного відрізка часу, і впливає на швидкість та точність оптимізації контролера. Зазвичай, більша кі-

лькість ітерацій дозволяє отримати більш точну оптимальну стратегію керування, але водночас може збільшити обчислювальну складність та вимоги до ресурсів.

Важливо збалансувати кількість ітерацій *Iterations Per Sample Time* так, щоб контролер зберігав ефективність та швидкість реакції на зміни у системі керування. Занадто багато ітерацій може призвести до затримок в реакції контролера, тоді як занадто мало ітерацій може призвести до недостатньої точності оптимізації.

Загальною практикою є експериментальне налаштування цього параметра в залежності від конкретних вимог та обмежень системи керування для досягнення оптимального балансу між точністю та швидкістю оптимізації.

Після установки параметрів оптимізації вони вводяться в блок *NN Predictive controller* системи *Simulink*.

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

При синтезі регулятора варіюються значення параметрів N_2 , N_u і ρ (при цьому N_1 фіксоване та дорівнює 1), а також параметра одновимірного пошуку α , який визначає поріг зменшення показника якості та кількість ітерацій на один такт дискретності γ . Додатково задається процедура одновимірного пошуку.

Дослідження показали, що параметри N_u , ρ і α мають незначний вплив на результати синтезу, тому були обрані значення: $N_u=2$, $\rho=0,05$, $\alpha=0,001$. Як процедуру одновимірного пошуку використано **csrchbac**.

Параметри N_2 і γ значно впливають на роботу регулятора. Збільшення цих значень підвищує точність, проте істотно зростає обсяг обчислень на кожному такті дискретності. Оптимальні значення для розв'язуваної задачі знаходяться в межах $N_2 = 15 \div 25$, $\gamma = 2 \div 3$.

При ідентифікації об'єкта керування ключовим питанням є вибір кількості нейронів у прихованому шарі N . Недостатня кількість нейронів унеможливилює виконання поставлених завдань, а надмірна кількість призводить до перенавчання та збільшення обсягу обчислень. Для даної задачі оптимальне значення $N = 7 \div 12$, при цьому помилки навчання, а також контрольної та тестової вибірок ε знаходиться в межах $10^{-3} \div 10^{-5}$.

Успіх навчання мережі суттєво залежить від довжини навчальної вибірки N_b та такту дискретності Δt , який визначає інтервал між двома послідовними моментами збору даних. Оптимальні значення: $N_b = 10000$, $\Delta t = 0,05 \div 0,1$ с. Збільшення Δt знижує точність обчислень і зменшує різницю між помилкою на навчальній множині та помилками на контрольній і тестовій множинах. Зменшення Δt вимагає збільшення N_b , що призводить до значного зростання часу навчання без суттєвого зменшення помилки.

Для отримання репрезентативної вибірки важливо правильно встановити мінімальне та максимальне значення інтервалу ідентифікації. Ці параметри залежать від характеристик об'єкта **Subsystem**. У даній роботі обрані межі: $t_{\min} = 4 \div 5$ с, $t_{\max} = 10 \div 20$ с.

Під час синтезу нейромережевої моделі системи задається кількість елементів затримки на вході z_1 та виході z_2 моделі. Найкращі результати досягнуті при $z_1 = 2$, $z_2 = 2 \div 5$.

Результат навчання мережі також залежить від початкових значень ваг w_{ij} та кількості навчальних циклів $N_{\text{ц}}$. Для досягнення глобального мінімуму процес навчання слід повторювати багаторазово з різними початковими значеннями w_{ij} та різними значеннями $N_{\text{ц}}$. У даній роботі для кожної конфігурації мережі було обрано кілька сотень початкових точок, а кількість навчальних циклів, після яких помилка переставала зменшуватись, становила $300 \div 600$.

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

Для побудови графіків перехідних процесів двомасової системи з нейрорегулятором NN Predictive Controller використовуємо моделі Simulink, приведені на рис. 4.5 або 4.27.

На рис. 4.28 – 4.31 приведені графіки перехідних процесів основних змінних стану синтезованої системи. З аналізу отриманих характеристик виходить, що реакція системи на ступінчасті дії з випадковою амплітудою цілком задовільна, має коливальний характер з невеликим перерегулюванням

Отже, синтезований нейромережевий регулятор з прогнозом NN Predictive Controller може бути використаний для управління трьохмасовою електромеханічною системою, що розглядається в роботі.

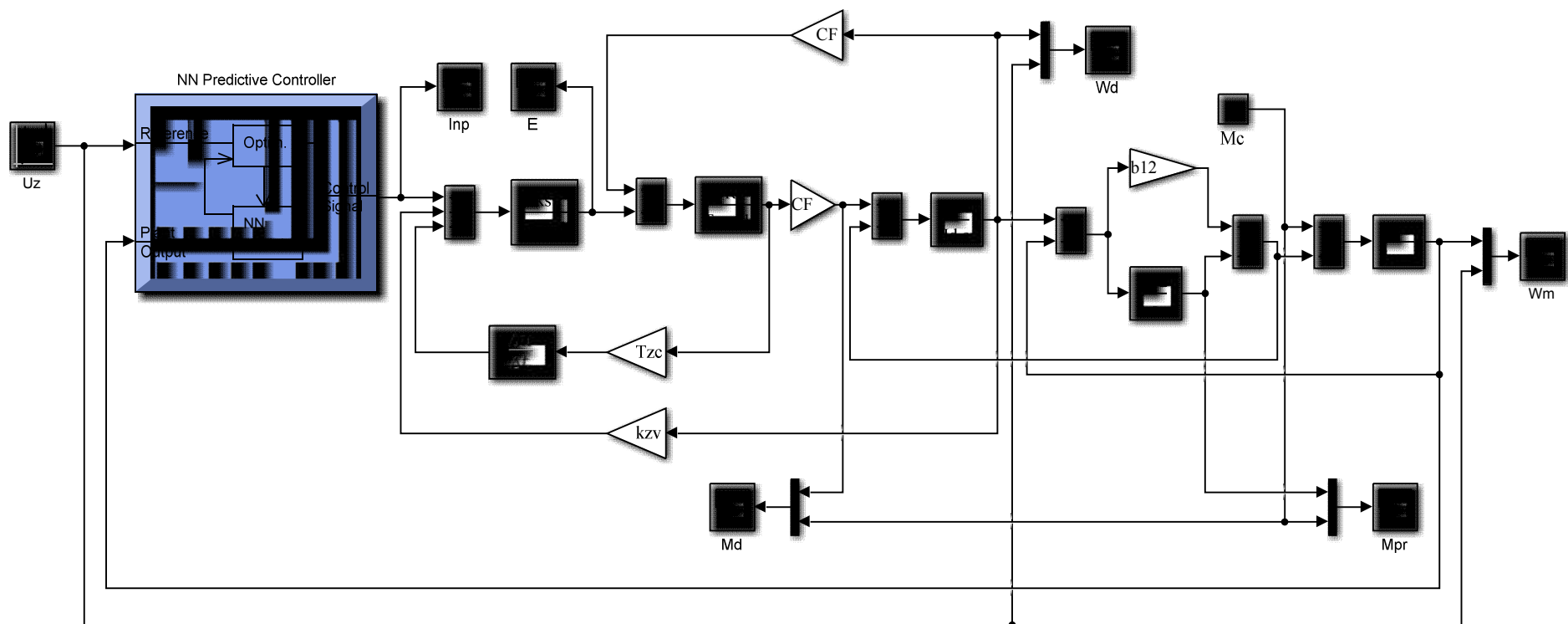
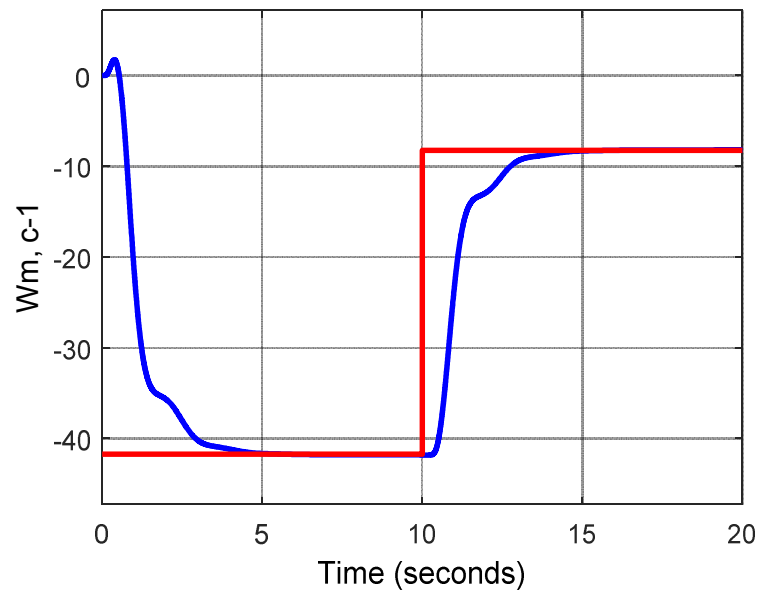
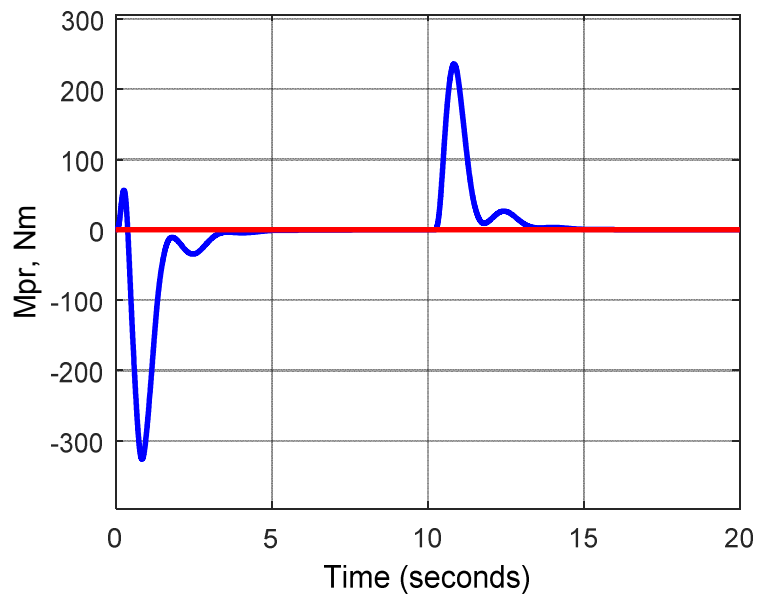


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором



a)



б)

Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

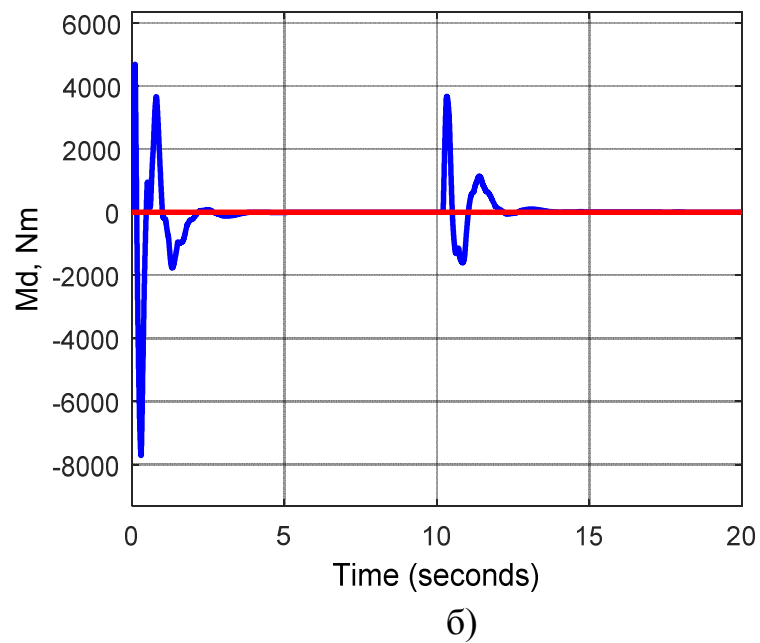
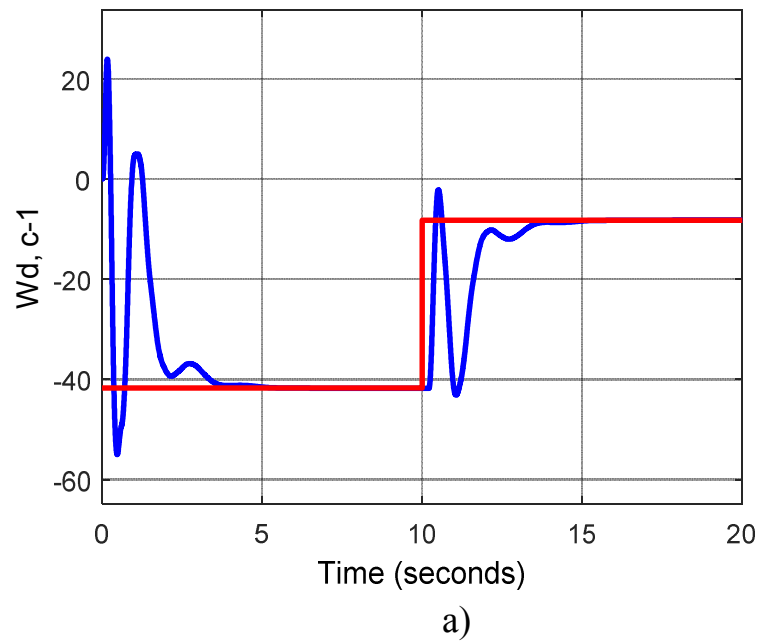
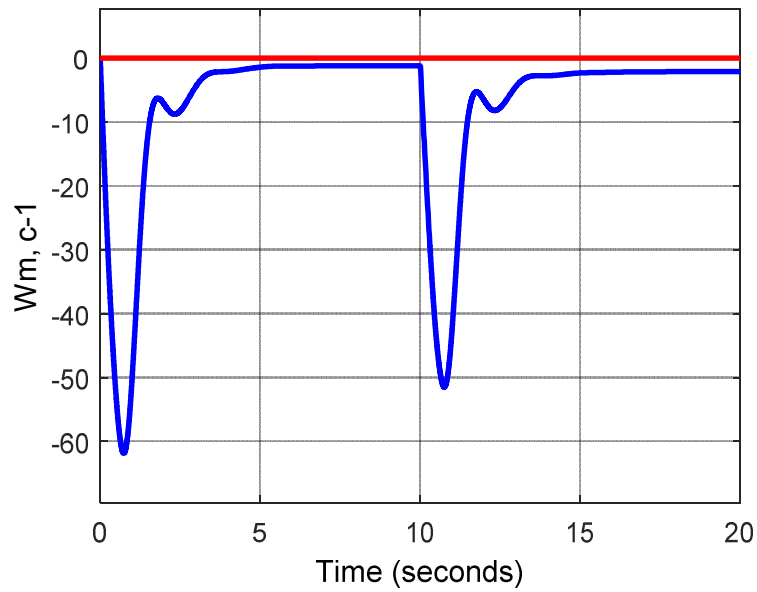
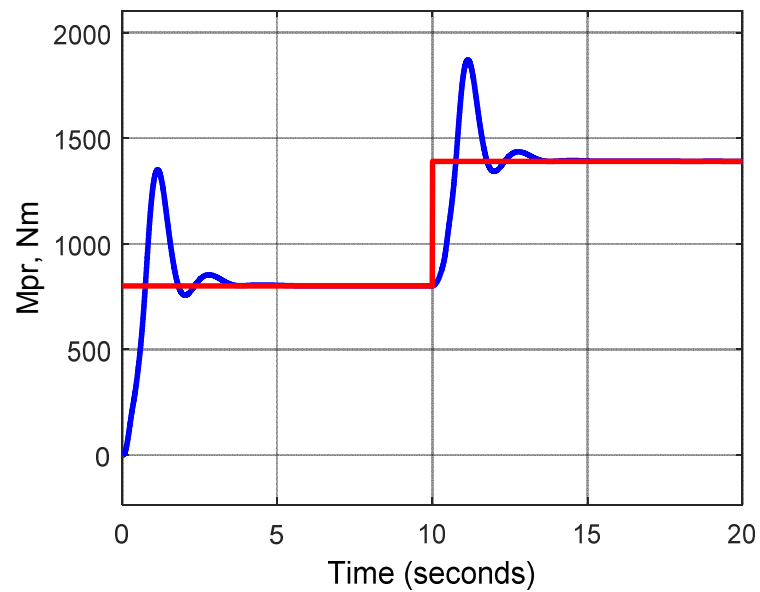


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

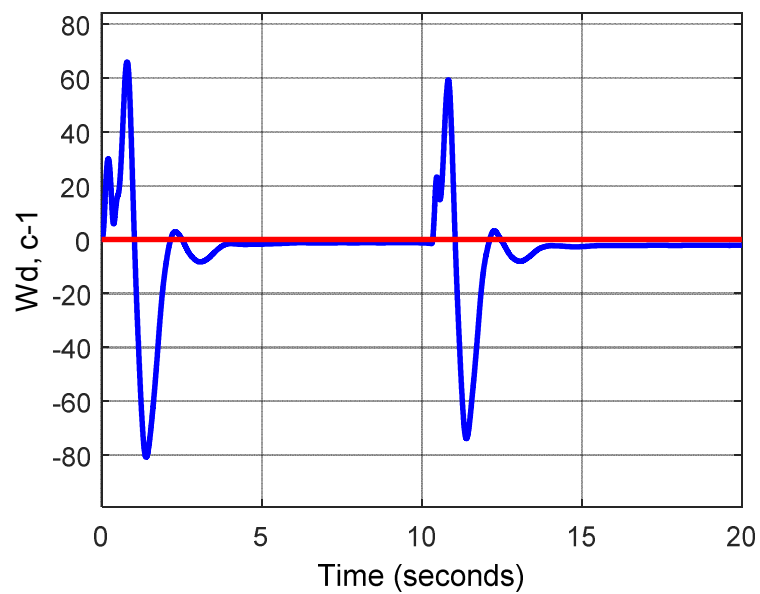


a)

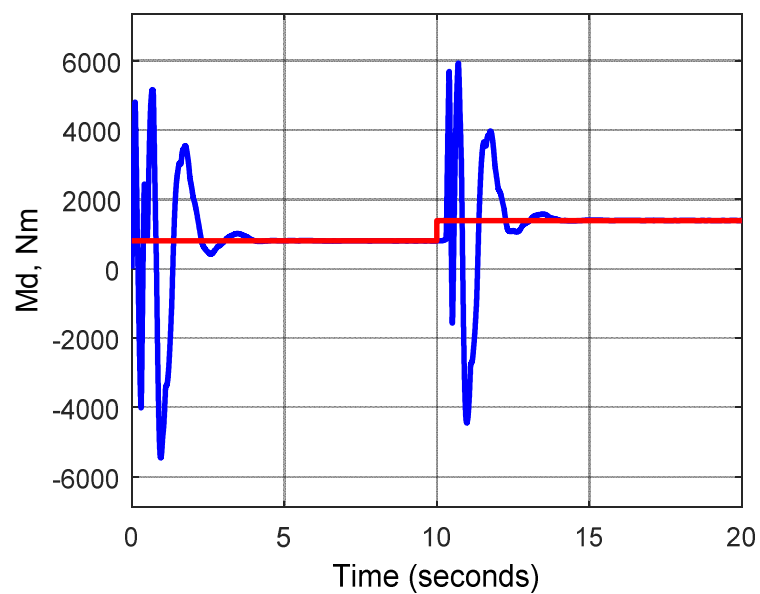


б)

Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{\text{пр}} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії



a)



б)

Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проаналізовано принципи побудови нейрорегулятора, реалізованих в пакеті прикладних програм (ППП) Neural Network Toolbox системи MATLAB: регулятор з прогнозом (NN Predictive Controller); регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) і регулятор на основі еталонної моделі (Model Reference Controller). Для системи управління, що розглядається в даній роботі, обраний регулятор з прогнозом NN Predictive Controller. Розглянуто структуру та порядок синтезу нейрорегулятора.

2. У режимі SIMULINK системи MATLAB розроблена системи управління з нейрорегулятора NN Predictive Controller, яка включає блок керованого об'єкта, в якості якого використовується схема моделі двомасової системи управління механізмом обертання колеса роторного екскаватора і блок регулятора NN Predictive Controller.

3. Шляхом варіювання параметрів навчальної послідовності, з використанням моделі виконавчого пристрою, виконана генерація навчальної послідовності, яка дозволила виконати тренування мережі з високим ступенем точності. Оптимальні параметри навчальної послідовності наступні: довжина навчальної вибірки 10000; інтервал між двома послідовними моментами знімання даних 0,5с; мінімальний інтервал ідентифікації 8 с; максимальний інтервал ідентифікації 20 с.

4. Створено двошарову нейронну мережу з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-6} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 0,05 .

5. Виконано розробку нейрорегулятора з передбаченням NN Predictive Controller. В ході варіювання параметрів нейрорегулятора в широких межах визначено, які параметри суттєво впливають на ефективність регулювання, та встановлено їх оптимальні значення. Застосування нейромережевої моделі об'є-

кта управління, що забезпечує високий рівень ідентифікації, дозволило створити нейрорегулятор, який демонструє високі показники якості системи.

6. Проведено моделювання системи з синтезованим нейрорегулятором. Аналіз моделювання показав, що перехідні процеси змінних стану системи за участю нейрорегулятора NN Predictive Controller при запуску та накиді навантаження мають задовільні характеристики. Перерегулювання відсутнє, час регулювання складає 4 с, що відповідає завданню.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розв'язана актуальна науково-практична задача розробки нейромережевої системи управління багатомасовою електромеханічною системою механізму обертання колеса роторного екскаватора, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління основних механізмів роторного екскаватора, що забезпечують високу якість регулювання і з урахуванням вимог, що пред'являються до сучасних систем управління, науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму обертання колеса роторного екскаватора.

2. На підставі аналізу вимог до систем електроприводів основних механізмів роторних екскаваторів для електроприводу механізму обертання колеса вибраний електропривод постійного струму, виконаний за системою Г–Д. Виконано вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу.

3. Проведений вибір і розрахунок системи автоматичного управління параметрами електроприводу. Для забезпечення необхідної жорсткості робочої частини механічної характеристики і обмеження максимального значення моменту, що розвивається, вибрано систему із зворотним зв'язком за швидкістю, гнучким зворотним зв'язком за струмом і зворотним зв'язком за струмом з відсіченням. Проведено розрахунок статичних характеристик розімкненої і замкненої системи. Складено алгоритмічну схему системи та визначені передавальні функції системи по задаючій і збурюючій діях. Проведено моделювання системи на ЕОМ з використанням пакету прокладних програм MATLAB.

4. Розроблена математична модель двомасової системи, що враховує пружність механічних з'єднань. Виконано моделювання системи за допомогою комп'ютерних засобів. Дослідження виявили, що в двомасовій системі

перехідні процеси основних координат супроводжуються значними коливаннями.

5. Для додання системі бажаних динамічних характеристик в роботі виконаний синтез нейромережевої системи управління з нейрорегулятором NN Predictive Controller, реалізованим в пакеті прикладних програм Neural Network Toolbox системи MATLAB. Розглянутий принцип побудови і функціонування нейромережевого регулятора, приведений порядок синтезу регулятора, побудовані процеси в двомасовій системі з нейрорегулятором. Аналіз результатів моделювання показав, що реакція системи на ступінчасту дію з випадковою амплітудою цілком задовільна, має коливальний характер з достатньо швидким загасанням. Система є астатичною. Перерегулювання відсутнє, час регулювання складає 4 с, що відповідає завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології нейронних мереж і нечіткого моделювання в системах управління: підруч. для здобувачів вищої освіти спец. 151 Автоматизація та комп'ютерно-інтегровані технології / Г.І. Канюк, Б.І. Кузнецов, Т.Ю. Василець, А.Ю. Мезеря, О.О. Варфоломійєв. – Харків : Друкарня Мадрид, 2020. – 306 с.
2. Нейромережеві технології в системах управління: Підручник для візів./ Б. І. Кузнецов, Т.Ю. Василець, Т.Б. Нікітіна, В. В. Коломиєць, О.О. Варфоломійєв; Укр. інж.-пед. акад.. - Харків: УПА, 2014. - 232 с.
3. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
4. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
5. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г.Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006, 404 с.
6. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
7. Тимошук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. -444 с.
8. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
9. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
10. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
11. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344с.

12. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления. – М.: ИПРЖР, 2002. – 480 с.
13. Sen P., Hearn G.E., Zhang Y. Adaptive Neural Controller // In Neural Network Systems Techniques and Applications. Hrsg. Leondes C.T. – 1998. – Vol.4. – P. 274-343.
14. Werbos P.J. Backpropagation and neurocontrol: A review and prospects // Proc. of International Joint Conf. On Neural Networks. – Washington DC. – 1989. – Vol. 1. – P. 209-216.
15. Psaltis A.; Sideris A., Yamamura A. A multi-layered neural network controller // IEEE Control Systems Mag. – 1988. – №8. – P. 17-21.
16. Saerens M.; Soquet A. A neural controller // Proceedings of the 1st IEE International Conference on Artificial Neural Networks, London. – 1989. – P. 211-215.
17. Narendra K.S. Neural networks for control: theory and practice // Proc. IEEE. – 1996. – №84. – P. 1385-1406.
18. Neural networks for control systems: A survey K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop // Automatica. – 1992. – Vol. 28. – № 6. P. 1083-1112.
19. Ishiguro A. et al. A neural network compensator for uncertainties of robotic manipulators // IEEE Trans, on Industrial Electronics. – 1992. – 339. – P. 565-569.
20. Garcia C.E.; Morari M. Internal Model Control: Unifying Review and Some New Results // Ind.Eng.Chem.Process Des.Dev. – 1982. – №21. – 308-323.
21. Park Y.-M., Choi M.S.; Lee Y. An optimal tracking neuro-controller for non-linear dynamic systems // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1099-1110.
22. Mistry, S.I.; Nair S.S. Indirect control of a class of nonlinear dynamic systems. // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1015-1023.
23. Miyamoto H., Kawato M., Setoyama, T., Suzuki R. Feedback error learning neural network for trajectory control of robotic manipulator // IEEE Trans, on Neural Networks. – 1988. – Vol. 1. – P. 251-265.
24. Jung S. Neural Network Controllers for Robot Manipulators. PhD thesis, University of California. – Davis. – 1996. – 132 p.

25. Jung S.; Hsia T.C. A new neural network control technique for robot manipulators // *Robotica*. – 1995. – Vol. 13. – P. 477-484.
26. Sorensen O. Neural Networks in Control Applications. PhD thesis. Aalborg University. Department of Control Engineering. – 1994. – 187 p.
27. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // *Automatica*. – 1989. – Vol.25. – P.335-348.
28. Системы электропривода и электрооборудования роторных экскаваторов. / Ю.Т. Калашников, А.О. Горнов, В.Н. Остриров и др. – М.; Энергоатомиздат, 1988. – 312 с.
29. Подерни Р.Ю. Горные машины и комплексы для открытых работ в 2 томах - М.; МГГУ, Издание 4, 2001 – 754 с.
30. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
31. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
32. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
33. Ладанюк А.П. Теорія автоматичного керування технологічними об'єктами : навч. посіб. / Ладанюк А.П., Архангельська К.С., Власенко Л.О. – К.: НУХТ, 2014 – 274 с.
34. Сорока К.О. Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина перша. Основи теорії систем автоматичного керування. : навч. посібник. – Харків : ФОП Тімченко, 2010. – 218 с.
35. Сорока К.О. Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина друга. Аналіз систем автоматичного керування засобами комп'ютерного моделювання : навч. посібник. – Харків : ФОП Тімченко, 2010. – 156 с.
36. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.

37. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
38. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
39. Автоматизований електропривод ч. 2 [Електронний ресурс]: навчальний посібник для студентів освітньої програми «Електромеханічні системи автоматизації, електропривод та електромобільність» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / В.І. Теряєв. – Київ: КПІ ім. Ігоря Сікорського, 2022.– 204 с
40. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч1.
41. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч2.
42. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» денної і заочної форми навчання)/ – Краматорськ: ДДМА, 2018. – 225 с.
43. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоменко. Харків, ХНАДУ, 2012. – 284 с.
44. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
45. Дьяконов В.П., Круглов В.В. MATLAB 6.5 SP1/SP2 + Simulink 5/6 инструменты искусственного интеллекта и биоинформатики. Серия «Библиотека профессионала». – М.: СОЛОН-ПРЕСС, 2006.– 456 с.

46. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. – СПб: Питер, 2001. –488 с.
47. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

ДОДАТОК А

МОДЕЛЬ НЕЙРОНА

А.1 Структурна схема нейрона

Елементарним осередком нейронної мережі є нейрон. Структура нейрона з єдиним скалярним входом показана на рис. А.1, а.

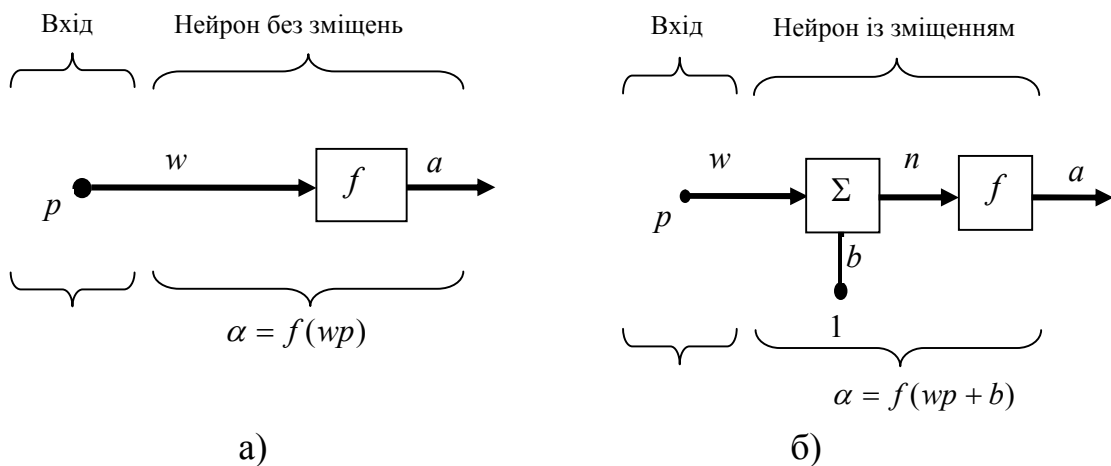


Рисунок А.1 – Структура простого нейрона

а) нейрон з одним скалярним входом;

б) нейрон з скалярним зсувом

Скалярний вхідний сигнал p множиться на скалярний ваговий коефіцієнт w і результуючий зважений вхід $w \cdot p$ є аргументом функції активації нейрона, яка породжує скалярний вихід a .

Нейрон, показаний на рис. А.1 б, доповнений скалярним зсувом b . Зсув підсумовується із зваженим входом $w \cdot p$ і приводить до зрушення аргументу функції f на величину b . Дію зсуву можна звести до схеми зважування, якщо уявити, що нейрон має другий вхідний сигнал із значенням, рівним 1. Вхід n функції активації нейрона як і раніше залишається скалярним і рівним сумі зваженого входу і зсуву b . Ця сума є аргументом функції активації f ; виходом функції активації є сигнал a . Константи w і b є скалярними параметрами нейрона. Основний принцип роботи нейронної мережі полягає в настройці параме-

трів нейрона так, щоб поведінка мережі відповідала деякій бажаній поведінці. Регулюючи ваги або параметри зсуву, можна навчити мережу виконувати конкретну роботу; можливо також, що мережа сама коректуватиме свої параметри, щоб досягти необхідного результату.

Рівняння нейрона із зсувом має вигляд

$$a = f(w \cdot p + b \cdot 1) \quad (\text{A.1})$$

Як вже наголошувалося, зсув b - скалярний параметр нейрона, який не є входом, що настроюється, а константа 1, яка управляє зсувом, розглядається, як вхід і може бути врахована у вигляді лінійної комбінації векторів входу

$$a = [w \ p] \begin{bmatrix} b \\ 1 \end{bmatrix}. \quad (\text{A.2})$$

А.2 Функції активації нейронів

Функції активації (передавальні функції) нейрона можуть мати найрізноманітніший вигляд. Функція активації f , як правило, належить до класу сигмоїдальних функцій (сигмоїдальною (S-образною) функцією називається безперервна функція, що має дві горизонтальні асимптоти і одну точку перегину) з аргументом n і виходом a .

Розглянемо три найбільш поширені форми функції активації.

Одинична функція активації з жорстким обмеженням `hardlim`. Ця функція описується співвідношенням $a = \text{hardlim}(n) = 1(n)$ і показана на рис.А.2. Вона рівна 0, якщо $n < 0$, і 1, якщо $n \geq 0$.

До складу пакету прикладних програм (ППП) `Neural Network Toolbox` системи `MATLAB` входить М-функція `hardlim`, що реалізовує функцію активації з жорсткими обмеженнями.

Лінійна функція активації `purelin`. Ця функція описується співвідношення $a = \text{purelin}(n) = n$ і показана на рис. А1.3.

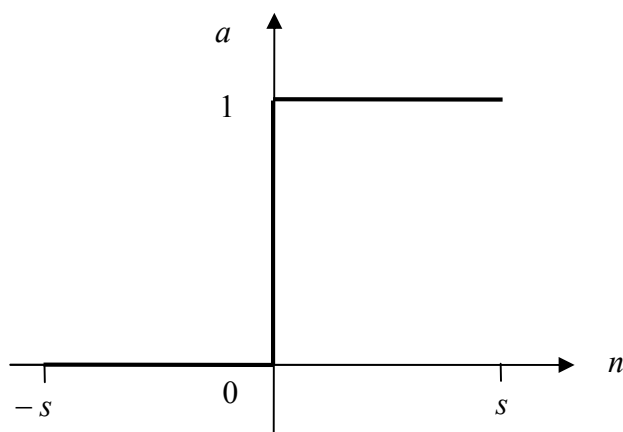


Рисунок А.2 – Одинична функція активації

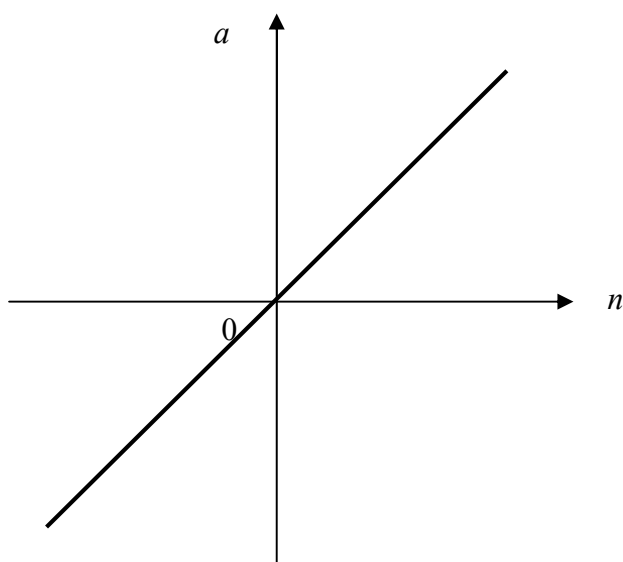


Рисунок А.3 – Лінійна функція активації

Логістична функція активації logsig . Ця функція описується співвідношення $a = \text{logsig}(n) = 1/(1 + \exp(-n))$ і показана на рис. А.4. Вона належить до класу сигмоїдальних функцій, і її аргумент може приймати будь-яке значення в діапазоні від $-\infty$ до $+\infty$ а вихід змінюється в діапазоні від 0 до 1. У ППП Neural Network Toolbox вона представлена М-функцією logsig . Завдяки властивості диференцируемости ця функція часто використовується в мережах з навчанням на основі методу зворотного розповсюдження помилки.

У багатошарових мережах часто застосовуються нелінійні сигмоїдальні функції активації типу гіперболічного тангенса (рис. А.5).

У ППП Neural Network Toolbox включені і інші функції активації. Використовуючи мову MATLAB, користувач може створювати і свої власні унікальні функції.

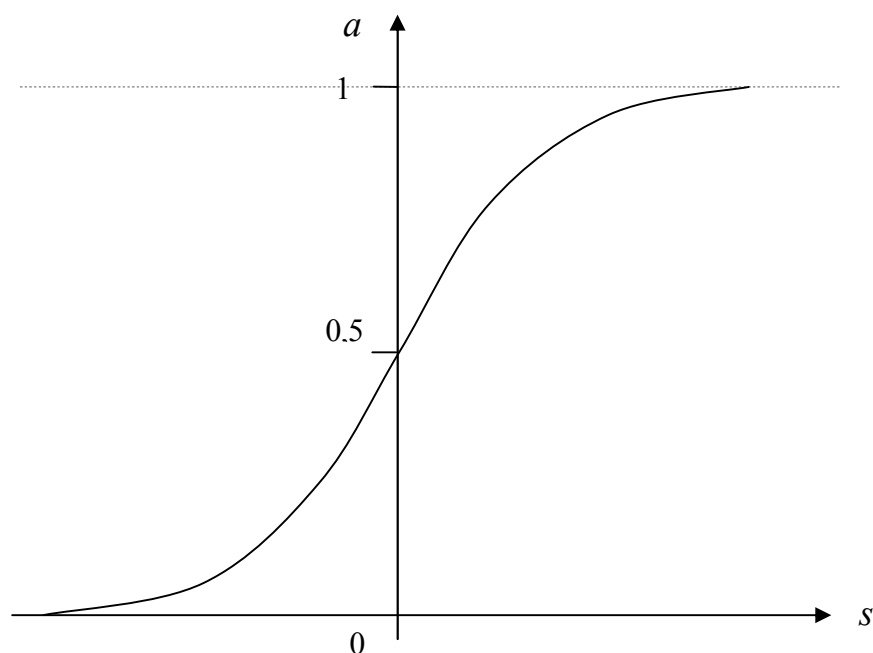


Рисунок А.4 – Логістична функція активації

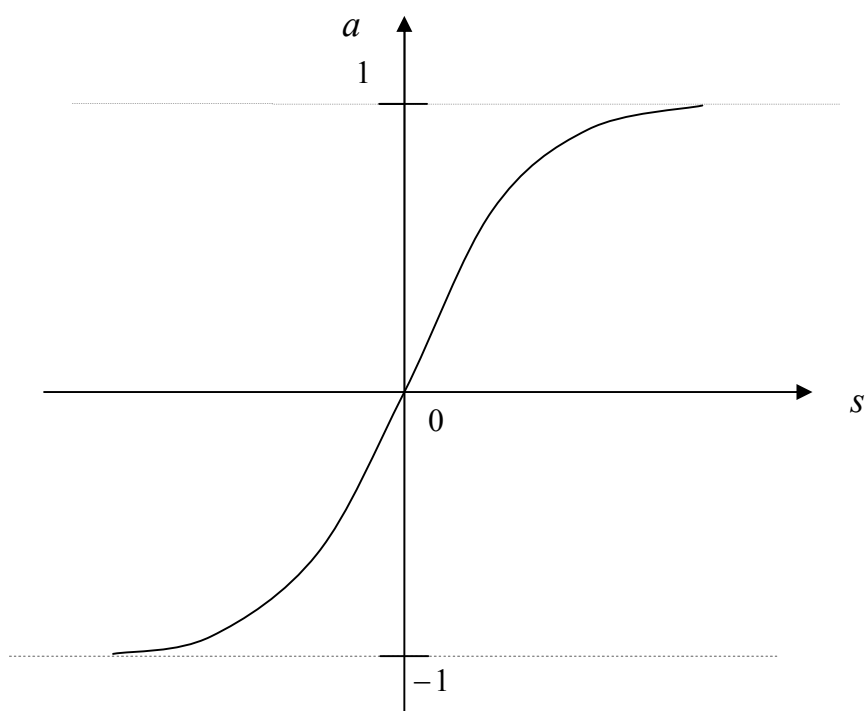


Рисунок А.5 – Функція активації типу гіперболічного тангенса

А.3 Нейрон з векторним входом

Нейрон з одним вектором входу $\mathbf{p}$ з R елементами $p_1, p_2, \dots, p_R$ показаний на рис. А.6. Тут кожен елемент входу умножається на ваги $w_{11}, w_{12}, \dots, w_{1R}$ відповідно і зважені значення передаються на суматор. Їх сума рівна скалярному твору вектора - рядки $\mathbf{W}$ на вектор входу $\mathbf{p}$.

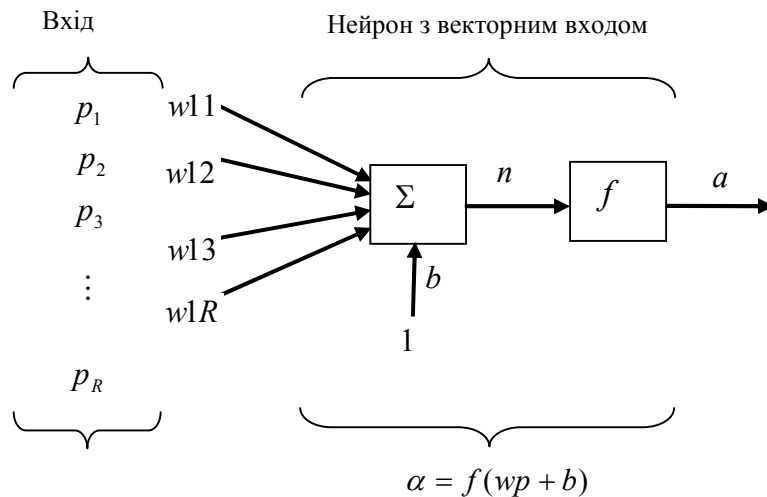


Рисунок А.6 – Схема нейрона з векторним входом

Нейрон має зсув b , який підсумовується із зваженою сумою входів. Результуюча сума n рівна

$$n = w_{11} \cdot p_1 + w_{12} \cdot p_2 + \dots + w_{1R} \cdot p_R + b \quad (1.3)$$

і служить аргументом функції активації f . У нотації мови MATLAB цей вираз записується так:

$$n = \mathbf{W} \cdot \mathbf{p} + b \quad (1.4)$$

Структура нейрона, показана вище, містить багато зайвих деталей. При розгляді мереж, що складаються з великого числа нейронів, в ППП Neural Network Toolbox використовується укрупнена структурна схема нейрона (рис. А.7).

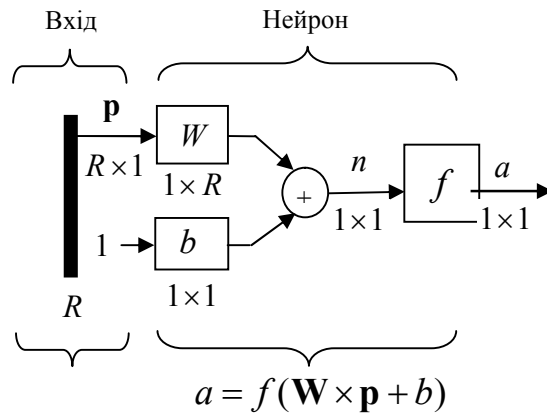


Рисунок А.7 – Укрупнена структурна схема нейрона

Вхід нейрона зображається у вигляді темної вертикальної межі, під якою вказується кількість елементів входу R . Розмір вектора входу $\mathbf{p}$ вказується нижче за символ $\mathbf{p}$ і рівний R . Вектор входу множиться на вектор-рядок $\mathbf{W}$ довжини R . Як і раніше, константа 1 розглядається як вхід, який множиться на скалярний зсув b . Входом n функції активації нейрона служить сума зсуву b і добутки $\mathbf{W} \cdot \mathbf{p}$. Ця сума перетвориться функцією активації f на виході якої отримуємо вихід нейрона a , який в даному випадку є скалярною величиною.

ДОДАТОК Б

АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

Б.1 Одношарові мережі

Реальна нейронна мережа може містити один або більша кількість шарів і відповідно характеризуватися як одношарова або як багатошарова.

Розгорнута схема мережі з одного шару з R вхідними елементами і S нейронами показана на рис. Б.1

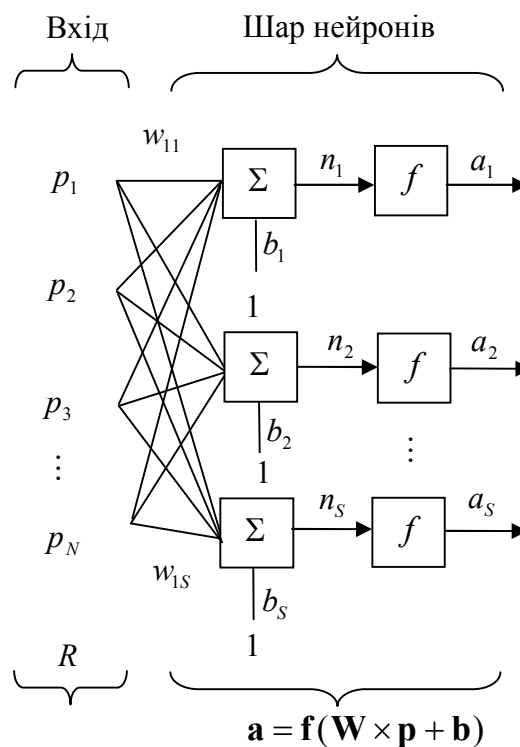


Рисунок Б.1 – Схема одношарової мережі

У цій мережі кожен елемент вектора входу сполучений зі всіма входами нейрона і це з'єднання задається матрицею вагів $\mathbf{W}$; при цьому кожен i -й нейрон включає елемент, що підсумовує, який формує скалярний вихід $n(i)$. Сукупність скалярних функцій $n(i)$ об'єднується в S -елементний вектор входу $\mathbf{n}$ функції активації шаруючи. Виходи шару нейронів формують вектор-стовпець $\mathbf{a}$, і, таким чином, опис шару нейронів має вигляд

$$\mathbf{a} = \mathbf{f}(\mathbf{W} \cdot \mathbf{p} + \mathbf{b})$$

Кількість входів R в шарі може не співпадати з кількістю нейронів S . У кожному шарі, як правило, використовується одна і та ж функція активації. Проте можна створювати складені шари нейронів з використанням різних функцій активації, сполучаючи мережі, подібні зображеною на рис. Б.1, паралельно. Обидві мережі матимуть ті ж самі входи, і кожна мережа генеруватиме певну частину виходів. Елементи вектора входу передаються в мережу через матрицю вагів $\mathbf{W}$, що має вигляд

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1R} \\ w_{21} & w_{22} & \dots & w_{2R} \\ \dots & \dots & \dots & \dots \\ w_{S1} & w_{S2} & \dots & w_{SR} \end{bmatrix}$$

Відмітимо, що індекси рядків матриці $\mathbf{W}$ указують адресатів (пункти призначення) вагів нейронів, а індекси стовпців - яке джерело є входом для цієї ваги. Таким чином, елемент матриці вагів $w_{12} = \mathbf{W}(1,2)$ визначає коефіцієнт, на який множиться другий елемент входу при передачі його на перший нейрон.

Для одношарової мережі з S нейронами укрупнена структурна схема показана рис.Б.2.

На рисунку позначен: $\mathbf{p}$ - вектор входу розміру $R \times 1$; $\mathbf{W}$ - вагова матриця розміру $S \times R$; $\mathbf{a}$, $\mathbf{b}$, $\mathbf{n}$ - вектори розміру $S \times 1$.

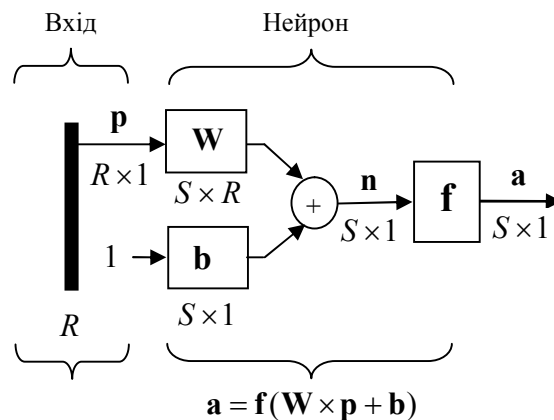


Рисунок Б.2 – Укрупнена структурна схема одношарової мережі

Б.2. Багатошарові мережі

Розглянемо мережі, що мають декілька шарів. Вагові матриці, сполучені з входами, називаються вагами вхідного шару, а вагові матриці для сигналів, витікаючі з шару, називаються вагами вихідного шару. Далі, будемо використовувати верхні індекси, щоб вказати джерело і адресат для різних вагів і інших елементів нейронної мережі. Щоб пояснити це, розглянемо спочатку тільки один, перший шар багатошарової мережі (рис. Б.3)

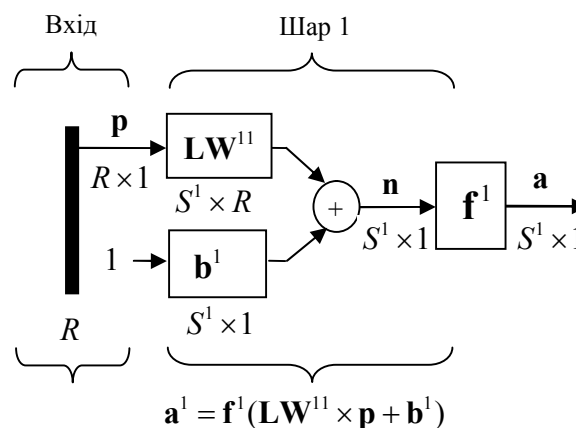


Рисунок Б.3 – Структурна схема першого шару багатошарової мережі

Позначимо вагову матрицю, пов'язану з входами, через $\mathbf{IW}^{11}$, верхні індекси якої указують, що джерелом входів є перший шар (другий індекс) і адреси том є також перший шар (перший індекс). Елементи цього шару, такі, як зсув $\mathbf{b}^1$, вхід функції активації $\mathbf{n}^1$ і вихід шару $\mathbf{a}^1$ мають верхній індекс 1, щоб позначити, що вони пов'язані з першим шаром. Надалі для матриць вагів входу і виходу шару будуть використані позначення $\mathbf{IW}$ (Input Weight) і $\mathbf{LW}$ (Layer Weight) відповідно.

Коли мережа має декілька шарів, то кожен шар має свою матрицю вагів $\mathbf{W}$, вектор зсуву $\mathbf{b}$ і вектор виходу $\mathbf{a}$. Щоб розрізняти вагові матриці, вектори виходу і т.д. для кожного з цих шарів, введемо номер шару як верхній індекс для тієї, що представляє інтерес, змінної. Використання цієї системи позначень для мережі з трьох шарів можна бачити на показаній нижче структурній схемі і з рівнянь, приведених в нижній частині рис. Б.4.

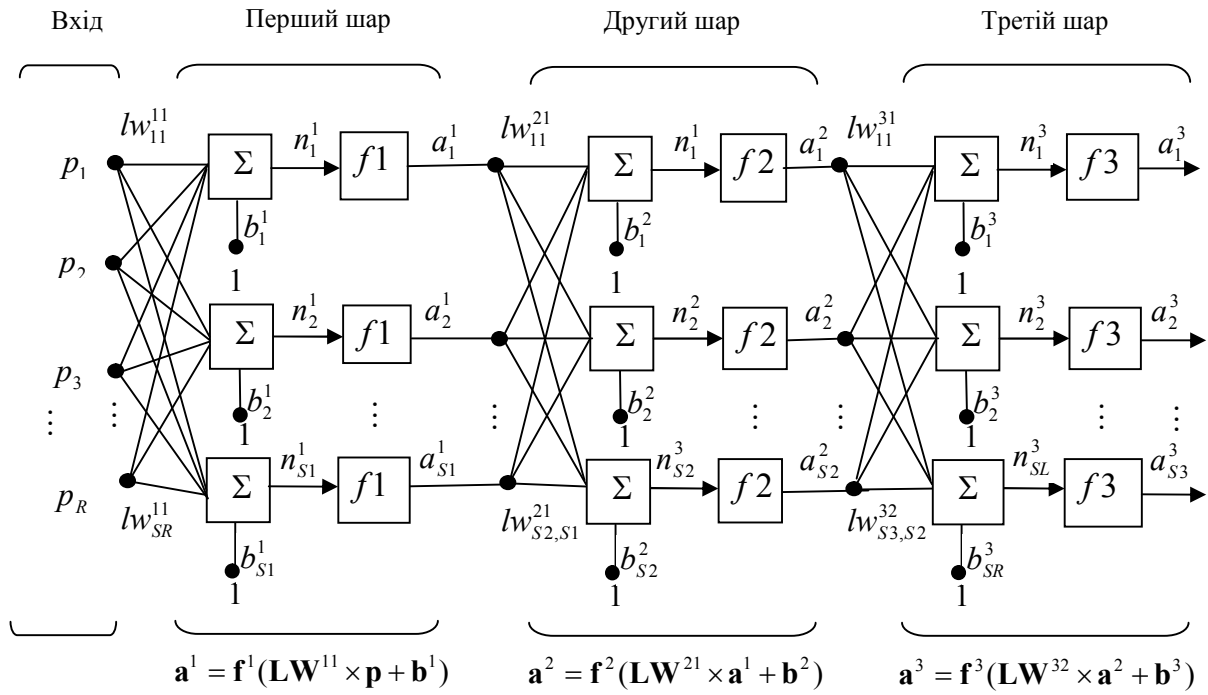


Рисунок Б.4 – Структурна схема багатошарової мережі

Мережа, показана вище, має R входів, S^1 нейронів в першому шарі, S^2 нейронів в другому шарі і т.д. Для спільності вважатимемо, що різні шари мають різне число нейронів. На зсуви для кожного нейрона поданий постійний вхідний сигнал 1. Відмітимо, що виходи кожного проміжного шару служать входами для наступного шару. Таким чином, шар 2 може бути розглянутий як один шар мережі з S^1 входами S^2 нейронами і $S^1 \times S^2$ матрицею вагів $\mathbf{W}_2$. Вхід до шару 2 є 1, а вихід - 2. Тепер, коли позначені всі вектори і матриці шару 2, можна трактувати його як самостійну одношарову мережу. Такий підхід може бути використаний до будь-якого шару мережі.

Шари багатошарової мережі мають різні призначення. Шар, який утворює вихід мережі, називається шаром виходу. Всі інші шари називаються прихованими шарами. Тришарова мережа, показана вище, має вихідний шар (шар 3) і 2 прихованих шари (шар 1 і шар 2). Ця ж тришарова мережа може бути представлена у вигляді укрупненої структурної схеми (рис. Б.5).

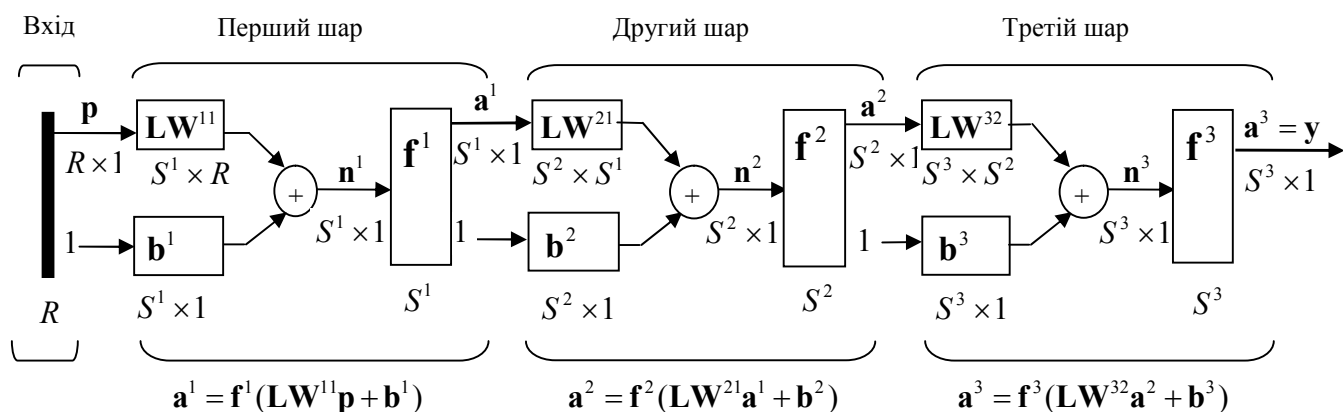


Рисунок Б.5 – Укрупнена структурна схема багатошарової мережі

Відмітимо, що вихід третього шару a^3 позначений через y . Ця зроблено для того, щоб підкреслити, що вихід останнього шару є виходом мережі.

Багатошарові мережі володіють вельми потужними можливостями. Наприклад, двошарова мережа, в якій перший шар містить сигмоїдальну, а другий шар - лінійну функцію активації, може бути навчена апроксимувати з довільною точністю будь-яку функцію з кінцевим числом точок розриву.

На закінчення можна сформулювати наступні висновки. Вхід функції активації нейрона визначається зсувом і сумою зважених входів. Вихід нейрона залежить як від входів нейрона, так і від виду функції активації. Один нейрон не може вирішувати складні завдання, проте декілька нейронів, об'єднаних в один або декілька шарів, володіють великими можливостями.

Архітектура мережі складається з опису того, скільки шарів має мережу, кількості нейронів в кожному шарі, виду функції активації кожного шару і інформації про з'єднання шарів. Архітектура мережі залежить від того конкретного завдання, яке повинна вирішувати мережа.

Робота мережі полягає в обчисленні виходів мережі на основі відомих входів з метою формування бажаного відображення вхід/вихід. Конкретне завдання визначає число входів і число виходів мережі. Окрім числа нейронів у вихідному шарі мережі, для проектувальника важливе число нейронів в кожному шарі. Більша кількість нейронів в прихованих шарах забезпечує могутнішу

мережу. Якщо повинне бути реалізоване лінійне відображення, то слід використовувати нейрони з лінійними функціями активації. При цьому треба пам'ятати, що лінійні нейронні мережі не можуть формувати нелінійні відображення. Використання нелінійних функцій активації дозволяє набудувати нейронну мережу на реалізацію нелінійних зв'язків між входом і виходом.

Мережі із зсувом дозволяють формувати складніші зв'язки між входами і виходами, чим мережі без зсуву. Наприклад, нейрон без зсуву, коли всі входи нульові, завжди задаватиме вхід функції активації рівним нулю, проте нейрон із зсувом може бути навчений так, щоб за тих же умов задати вхід функції активації довільної форми.

У багатошарових мережах часто застосовуються нелінійні сигмоїдальні функції активації типа логістичної (див. рис. А.4) або гіперболічного тангенса (див. рис. А.5).

Якщо останній шар багатошарової мережі використовує такі функції активації, то виходи мережі будуть обмежені. Коли у вихідному шарі використовуються лінійні нейрони, то виходи мережі можуть приймати довільні значення. У ППП NNT передбачені М-функції, що дозволяють обчислювати похідні функцій активації.

Б.3 Мережі з прямою передачею сигналу

Одношарова мережа з S нейронами з функціями активації logsig , що має R входів, показана на рис. Б.6.

Ця мережа, що не має зворотних зв'язків, називається мережею з прямою передачею сигналу. Такі мережі часто мають один або більше прихованих шарів нейронів з сигмоїдальними функціями активації, тоді як вихідний шар містить нейрони з лінійними функціями активації. Мережі з такою архітектурою можуть відтворювати вельми складні нелінійні залежності між входом і виходом мережі.

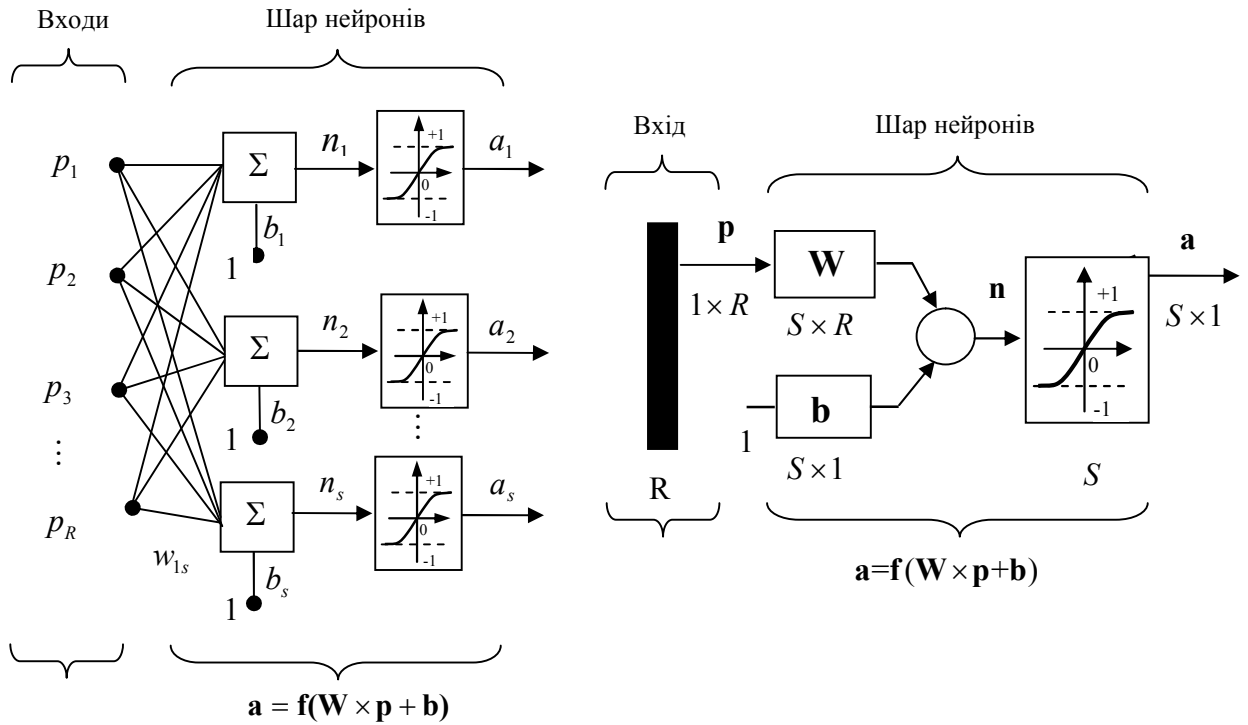


Рисунок Б.6 – Структурна схема одношарової мережі з R входами

Двошарова мережа, показана на рис Б.7, може бути використана для апроксимації функцій. Вона може достатньо точно відтворити будь-яку функцію з кінцевим числом точок розриву, якщо задати достатнє число нейронів прихованого шару.

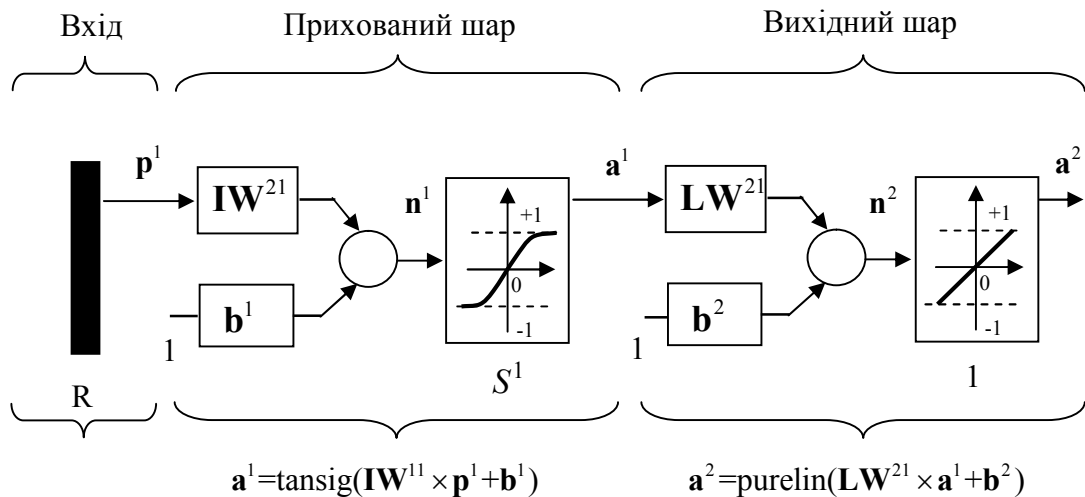


Рисунок Б.7 – Структурна схема багатошарової мережі з R входам

ДОДАТОК В

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

В.1 Сутність процесу навчання

Теорема про повноту. Будь-яка безперервна функція на замкнутій обмеженій множині може бути рівномірно наближена функціями, обчислюваними нейронними мережами, якщо функція активації нейрона двічі безперервно диференціюється і нелінійна (існують і інші варіанти теорем про обличчя).

Таким чином, нейронні мережі є універсальними апроксимуючими системами.

Очевидно, що процес функціонування ШНМ, тобто суть дій, які вона здатна виконувати, залежить від величин синаптичних зв'язків, тому, задавшись певною структурою ШНМ, що відповідає якому-небудь завданню, розробник мережі повинен знайти оптимальні значення всіх вагових коефіцієнтів.

Цей етап називається навчанням ШНМ, і від того, наскільки якісно він буде виконаний, залежить здатність мережі вирішувати поставлені перед нею проблеми під час функціонування.

Навчання штучною нейронною полягає в настройці вагових коефіцієнтів $w_{i,j}$ її базових процесорних елементів, результатом чого є виконання мережею конкретних завдань – розпізнавання, оптимізації, апроксимації, управління. Досягнення подібної мети формалізується критерієм якості Q , мінімальне значення $\min Q = Q^*$ якого відповідає якнайкращому рішення поставленої задачі.

Навчання мережі відбувається відповідно до схеми, показаної на рис.В.1.

При навчанні мережі є деяка база даних, що містить приклади. Пред'являючи перший приклад на вхід мережі, отримуємо від неї деяку відповідь, не обов'язково вірну. Відома і вірна (бажаний) відповідь. Обчислюючи різницю між бажаною відповіддю і реальною відповіддю мережі, отримуємо вектор помилки. Алгоритм навчання – це набір формул, який дозволяє по вектору поми-

лки обчислити необхідні поправки для вагів мережі. Один і той же приклад може пред'являтися мережі багато раз.

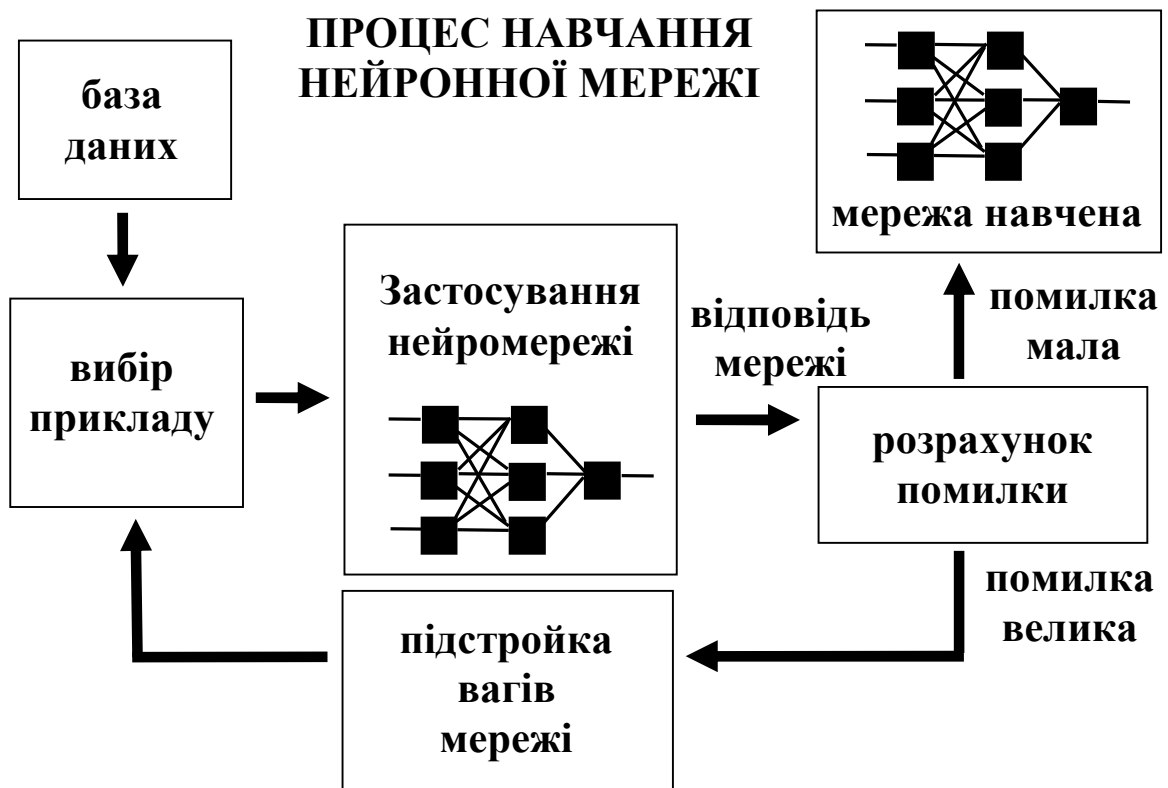


Рисунок В.1 – Ілюстрація процесу навчання ШНМ

Виявляється, що після багатократного пред'явлення прикладів ваги мережі стабілізуються, причому мережа дає правильні відповіді на всі (або майже все) приклади з бази даних. У такому разі говорять, що «мережа вивчила всі приклади», «мережа навчена», або «мережа натренована». У програмних реалізаціях можна бачити, що в процесі навчання функція помилки (наприклад, сума квадратів помилок по всіх виходах) поступово зменшується. Коли функція помилки досягає нуля або прийнятного малого рівня, тренування зупиняють, а отриману мережу вважають натренованою і готовою до застосування на нових даних.

Важливо відзначити, що вся інформація, яку мережа має про завдання, міститься в наборі прикладів. Тому якість навчання мережі безпосередньо залежить від кількості прикладів в навчальній вибірці, а також від того, наскільки

повно ці приклади описують дане завдання. Так, наприклад, безглуздо використовувати мережу для прогнозу фінансової кризи, якщо в навчальній вибірці криз не представлено. Вважається, що для повноцінного тренування потрібно хоч би декілька десятків (а краще за сотні) прикладів.

Математично процес навчання можна описати таким чином.

В процесі функціонування нейронна мережа формує вихідний сигнал Y відповідно до вхідного сигналу X , реалізуючи деяку функцію $Y = G(X)$. Якщо архітектура мережі задана, то вид функції G визначається значеннями синаптических вагів і зсувів мережі.

Хай рішенням деякої задачі є функція $Y = F(X)$, задана парами вхідних, – вихідних даних $(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)$, для яких $Y^k = F(X^k)$ ($k = 1 \dots N$).

Навчання полягає в пошуку (синтезі) функції G , близької до F в сенсі деякої функції помилки E .

Якщо вибрана множина навчальних прикладів – пара (X^k, Y^k) (де $k = 1 \dots N$) і спосіб обчислення функції помилки E , то навчання нейронної мережі перетворюється на завдання багатовимірної оптимізації, що має дуже велику розмірність, при цьому, оскільки функція E може мати довільний вигляд, навчання в загальному випадку – багатоекстремальне неопукле завдання оптимізації.

В.2 Алгоритми навчання нейронних мереж

3.2.1 Класифікація алгоритмів навчання

Для вирішення цього завдання можуть бути використані наступні (ітеративні) алгоритми:

1 – алгоритми локальної оптимізації з обчисленням приватних похідних першого порядку:

градієнтний алгоритм (метод швидкого спуску);

методи з одновимірною і двовимірною оптимізацією цільової

функції у напрямі антиградієнта;
метод зв'язаних градієнтів;
методи, що враховують напрям антиградієнта на декількох
кроках алгоритму;

2 – алгоритми локальної оптимізації з обчисленням приватних похідних першого і другого порядку:

метод Ньютона;
методи оптимізації з розрідженими матрицями Гессе;
квазіньютонівські методи;
метод Гаусса-Ньютона;
метод Левенберга-Марквардта і ін.

3 – стохастичні алгоритми оптимізації:

пошук у випадковому напрямі
імітація відпалу
метод Монте-Карло (чисельний метод статистичних випробувань);

4) алгоритми глобальної оптимізації (завдання глобальної оптимізації вирішуються за допомогою перебору значень змінних, від яких залежить цільова функція).

В.2.2 Алгоритм зворотного розповсюдження.

Розглянемо ідею одного з найпоширеніших алгоритмів навчання – алгоритму зворотного розповсюдження помилки (БР) (back propagation). Це ітеративний градієнтний алгоритм навчання, який використовується з метою мінімізації середньоквадратичного відхилення поточного виходу від бажаного виходу в багатошарових нейронних мережах.

Алгоритм зворотного розповсюдження використовується для навчання багатошарових нейронних мереж з послідовними зв'язками. Як зазначено вище, нейрони в таких мережах діляться на групи із загальним вхідним сигналом – шари, при цьому на кожен нейрон першого шару подаються всі елементи зовнішнього вхідного сигналу, а всі виходи нейронів q -го шару подаються на ко-

жен нейрон шару $(q + 1)$. Нейрони виконують зважене (з синаптичними вагами) підсумовування елементів вхідних сигналів; до даної суми додається зсув нейрона. Над отриманим результатом потім виконується нелінійне перетворення за допомогою активаційної функції. Значення функції активації є вихід нейрона.

У багатошарових мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, невідомі, і трьох- або більш шаровий персептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах ШНМ. Найбільш прийнятним варіантом навчання в таких умовах виявився градієнтний метод пошуку мінімуму функції помилки з розглядом сигналів помилки від виходів ШНМ до її входів, тобто в напрямі, зворотному прямому розповсюдженню сигналів в звичайному режимі роботи. Цей алгоритм навчання ШНМ отримав назву процедури зворотного розповсюдження.

У даному алгоритмі функція помилки є сумою квадратів розузгодження (помилки) бажаного виходу мережі і реального. При обчисленні елементів вектора градієнта використаний своєрідний вид похідних функцій активації сигмоїдального типу. Алгоритм діє циклічно (ітеративно), і його цикли прийнято називати епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні спостереження, вихідні значення мережі порівнюються з цільовими значеннями і обчислюється помилка. Значення помилки, а також градієнта поверхні помилок використовується для коректування вагів, після чого всі дії повторюються. Початкова конфігурація мережі вибирається випадковим чином, і процес навчання припиняється або коли пройдена певна кількість епох, або коли помилка досягне деякого певного рівня трохи, або коли помилка перестане зменшуватися (користувач може сам вибрати потрібну умову зупинки).

Приведемо словесний опис алгоритму.

Крок 1. Вагам мережі присвоюються невеликі початкові значення.

Крок 2. Вибирається чергова навчальна пара (X, Y) з навчальної множини; вектор X подається на вхід мережі.

Крок 3. Обчислюється вихід мережі.

Крок 4. Обчислюється різниця між необхідним Y (цільовим) і реальним (обчисленим) виходом мережі.

Крок 5. Ваги мережі коректуються так, щоб мінімізувати помилку (спочатку ваги вихідного шару, потім, з використанням правила диференціювання складної функції і відміченого своєрідного виду похідної сигмоїдальної функції, – ваги попереднього шару і т. п.).

Крок 6. Кроки з 2-го по 5-ий повторюються для кожної пари навчальної множини до тих пір, поки помилка на всій множині не досягне прийнятної величини.

Кроки 2 і 3 подібні до тих, які виконуються у вже навченій мережі.

Обчислення в мережі виконуються пошарово. На кроці 3 кожний з виходів мережі віднімається з відповідного компоненту цільового вектора з метою отримання помилки. Ця помилка використовується на кроці 5 для корекції вагів мережі.

Кроки 2 і 3 можна розглядати як «прохід вперед», оскільки сигнал розповсюджується по мережі від входу до виходу. Кроки 4 і 5 складають «зворотний прохід», оскільки тут обчислюваний сигнал помилки розповсюджується назад по мережі і використовується для підстроювання вагів.

Класичний метод зворотного розповсюдження відноситься до алгоритмів з лінійною збіжністю. Його відомими недоліками є: невисока швидкість збіжності (велике число ітерацій, потрібних для досягнення мінімуму функції помилки), можливість сходитися не до глобального, а до локальних рішень (локальним мінімумам відміченої функції). Можливий також параліч мережі, при якому більшість нейронів функціонують при дуже великих значеннях аргументу функції активації, тобто на її пологій ділянці (оскільки помилка пропорційна похідній, яка на даних ділянках мала, то процес навчання практично завмирає).

Для усунення цих недоліків були запропоновані численні модифікації алгоритму зворотного розповсюдження, які зв'язані з використанням різних функцій помилки, різних процедур визначення напрямку і величини кроку і т.п. Розглянемо один з них.

В.3 Метод зворотного розповсюдження помилки

Як було зазначено вище, метод зворотного розповсюдження помилки – метод ВР розроблений для навчання статичних нелінійних багатошарових нейронних мереж і використовує технологію послідовної і пошарової настройки базових процесорних елементів, починаючи з останнього, вихідного шару і закінчуючи настройкою елементів першого шару. «Урок» навчання штучної нейронної мережі може бути повторений необхідне число разів. Для настройки вагових коефіцієнтів базових елементів застосовується δ -правило в його загальній формі: у вигляді алгоритму градієнтного методу мінімізації критерію навчання $Q(\mathbf{e}_u)$. Тут $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)}$ – вектор помилки навчання мережі щодо бажаного (еталонного) виходу $\mathbf{u}^*$ при нелінійному згідно (1.10) перетворенні ввідного вектора $\mathbf{r}$ у вихідний вектор $\mathbf{u} = \mathbf{q}^{(K)}$ останнього шару K . Помилка $\mathbf{e}_u$, явним чином залежить від коефіцієнтів $w_{i,j}^{(K)}$ і може бути використана як аргумент функціонала Q для настройки базових елементів K -го шару. Явної залежності вектора $\mathbf{e}_u$ і функції $Q(\mathbf{e}_u)$ від вагових коефіцієнтів $w_{i,j}^{(l)}$ ($l = \overline{1, K-1}$) немає. Тому у разі використання критерію навчання $Q(\mathbf{e}_u)$ для настройки коефіцієнтів $w_{i,j}^{(l)}$ помилка $\mathbf{e}_u$ в процесі навчання багатошарової нейронної мережі перераховується послідовно в узагальнені помилки $\boldsymbol{\sigma}^{(l)}$, що явно залежать від значень $w_{i,j}^{(l)}$. Настройка коефіцієнтів $w_{i,j}^{(l)}$ здійснюється за δ -правилом, де використовуються узагальнені помилки $\boldsymbol{\sigma}^{(l)}$. Очевидно, що для шару K помилка $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$. У методі ВР далі використовується локальний квадратичний функціонал $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$. Мінімізація $Q(\boldsymbol{\sigma})$ по вагових коефіцієнтах $w_{i,j}^{(l)}$ ($l = K-1, \dots, 1$), що настроюються, методом градієнтного спуску з перерахунком помилки $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$ для шару K в помилку $\boldsymbol{\sigma}^{(l)} = \mathbf{e}(\mathbf{w}^{(l)})$ для шарів $l = K-1, K-2, \dots, 1$ визначає сутність ВР-технології навчання багатошарових нейронних мереж.

В.3.1 Налаштування коефіцієнтів вихідного шару.

Для шару K процедура налаштування вектора всіх вагових коефіцієнтів в K -му шарі $\mathbf{w}^{(k)}$ відповідає зазвичай алгоритму градієнтного спуску:

$$\mathbf{w}^{(k)}(k+1) = \mathbf{w}^{(K)} - \gamma \nabla_{\mathbf{w}} Q(\mathbf{e}_u(\mathbf{w}^{(k)}(k))) \quad (\text{B.1})$$

де:

$$\nabla_{\mathbf{w}} Q(\cdot) = \text{col} \left(\frac{\partial Q(\cdot)}{\partial \mathbf{w}_1^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_j^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_{n_k}^{(K)}} \right)$$

вектор розмірністю $(n_k \times 1)$. Вектор $\mathbf{w}^{(k)}$ – складний:

$$\mathbf{w}^{(K)} = \text{col}(\mathbf{w}_1^{(K)}, \dots, \mathbf{w}_i^{(K)}, \dots, \mathbf{w}_{n_K}^{(K)})$$

де:

$$\mathbf{w}^{(K)} = \text{col}(w_{i,1}^{(K)}, \dots, w_{i,m}^{(K)}, \dots, w_{n_{K-1}}^{(K)})$$

$\mathbf{q}^{(K-1)}(k)$ – вектор розмірністю $((n_{K-1} + 1) \times 1)$ базових елементів в шарі K :

$$\mathbf{q}^{(K-1)}(k) = \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}).$$

Сигнал ініціалізації $q_0^{(K-1)}$ зазвичай приймається рівним +1,

Градієнт

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = \mathbf{e}_n^T \frac{\partial \mathbf{e}_u}{\partial \mathbf{w}} = -\mathbf{e}_u^T \frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}}$$

де похідна:

$$\frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}} = \frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{w}^{(K)}} = \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right)$$

є добуток матриць Якобі. перша з них, така, що розкривається в наступному вигляді:

$$\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{w}_1^{(K)}} \\ \frac{\partial}{\partial \mathbf{w}_2^{(K)}} \\ \dots \\ \dots \\ \frac{\partial}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} \times [s_1^{(K)}, s_2^{(K)}, \dots, s_{n_K}^{(K)}] = \begin{bmatrix} \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_1^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_1^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_i^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_i^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} = \text{diag} \left(\frac{\partial s_i^{(K)}}{\partial \mathbf{w}_i^{(K)}} \right) -$$

квадратна матриця, кожен елемент якої на діагоналі є вектор-стовпець розмірністю $[(n_{K-1} + 1) \times 1]$, отже повна розмірність матриці рівна $(n_K(n_{K-1} + 1) \times n_K)$. В матриці $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ всі елементи, окрім діагональних, рівні нулю, а приватні похідні $\partial s_i^{(K)} / \partial \mathbf{w}_i^{(K)}$ на діагоналі – всі однакові і рівно вектору $\mathbf{q}^{(K-1)}$. Це витікає з визначення дискримінантної функції $s_i^{(K)}$

$$s_i^{(K)} = \sum_{j=0}^{n_{K-1}} w_{i,j}^{(K)} q_j^{(K-1)} = \mathbf{w}_i^{(K)} \mathbf{q}^{(K-1)}.$$

Інша матриця Якобі розкривається у вигляді діагональної матриці $(n_K \times n_K)$

$$\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} = \text{diag}(f'_s(\mathbf{s}^{(K)})) = \begin{bmatrix} f'_s(\mathbf{s}_1^{(K)}) & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & f'_s(\mathbf{s}_{n_K}^{(K)}) \end{bmatrix}.$$

З урахуванням приведених матриць Якобі градієнт функціонала настройки K -го шару багат шарової нейронної мережі можна записати таким чином:

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right) \mathbf{e}_u = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \text{diag}(f'_s(\mathbf{s}^{(K)})) \right) \mathbf{e}_u,$$

а алгоритм настройки вагових коефіцієнтів K -го шару має вигляд

$$\mathbf{w}^{(K)}(k+1) = \mathbf{w}^{(K)}(k) + \gamma \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}(k)} \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) \right) \mathbf{e}_u(k+1) \quad (\text{B.2})$$

Рекурентний алгоритм (B.2) можна розділити на n_K алгоритмів навчання i -х базових елементів, $i = \overline{1, n_K}$, оскільки діагональна матриця $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ складається з однакових елементів $\mathbf{q}^{(K-1)}$.

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i(k+1) \mathbf{q}^{(K-1)}(k+1). \end{aligned} \quad (\text{B.3})$$

Узагальнена помилка σ , навчання мережі для шару K рівна $f'_s(s_i^{(K)})e_{ui}$. У багатошарових мережах останній вихідний шар виконується зазвичай у вигляді мадалини з лінійними функціями активації і $f'_s(s) = 1$. Тоді $\sigma_i = e_{ui}$ і алгоритм настройки вагових коефіцієнтів шару K приймає вигляд:

$$\mathbf{w}_i^{(K)}(k+1) = \mathbf{w}_i^{(K)}(k) + \gamma e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) \quad (\text{B.4})$$

У алгоритмах (B.2)-(B.4) $k = 1, 2, \dots$ – номер інтеграції. Початкові значення вагових коефіцієнтів $\mathbf{w}^{(K)}(0)$ задаються довільно, але бажано в околиці значень $\mathbf{w}^{*(K)}$, для яких $Q(\mathbf{w}_{(K)}^*) = \min_{\mathbf{w}} Q(\mathbf{w}^{(K)})$.

В.3.2. Налаштування вагових коефіцієнтів прихованих шарів нейронної мережі

Для прихованих шарів $l = \overline{K-1, 1}$ помилки навчання в явному вигляді не можуть бути визначені. Проте очевидно, що якщо $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)} \neq 0$, то через архітектуру мережі існує ненульова узагальнена помилка і для всіх попередніх шарів. Суть методу ВР якраз і полягає в послідовному визначенні «прихованих» помилок, починаючи з обчислення помилки $\sigma^{(K)}$ у вихідному шарі K , потім помилки $\sigma^{(K-1)}$ в шарі $K-1$ і так до вхідного шару 1. Всі приховані по-

милки $\sigma^{(l)} = \sigma(\mathbf{w}_i^l)$ залежать явно від вагових коефіцієнтів, що настраюються $\mathbf{w}_i^l$. Розглянемо суть необхідних перетворень в δ -правилі (В.2), яке використовує ідею обчислення прихованих помилок в рекурентній процедурі настройки вагових коефіцієнтів в шарах $l = \overline{K-1, 1}$. З цією метою складемо алгоритм настройки вагових коефіцієнтів для окремих i -х базових елементів в шарі l :

$$\mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, \dots, w_{i,j}^{(l)}, \dots, w_{i,n_l}^{(l)}).$$

При виведенні алгоритму настройки вагових коефіцієнтів в шарах $K-1, K-2, \dots, 1$ вкажемо, що в загальному випадку градієнт функціонала Q щодо вектора вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ можна обчислити як:

$$\nabla \mathbf{w}_i^{(l)} Q = \frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \frac{\partial Q}{\partial \mathbf{q}^{(l)}}.$$

Тут, як і для шару K , вважаємо для спільності і для спрощення подальших перетворень вектор $\mathbf{q}^{(l)} = \text{col}(q_0^{(l)}, q_1^{(l)}, \dots, q_{n_l}^{(l)})$.

У практичних застосуваннях $q_0^{(l)} = +1$. У подальшому виводі вважатимемо, що ініціалізуючі сигнали $q_0^{(l)}$ є виходами фіктивних, «нульових» базових елементів з функціями активації $f(s_0^{(l)})$ того ж типу, що і в реальних елементах. Тоді випадку $q_0^{(l)} = +1$ відповідає робоча точка $f(s_0^{(l)}) = 1$.

З урахуванням зробленого припущення розмірність матриці Якобі $\partial \mathbf{s}^{(l)} / \partial \mathbf{w}_i^{(l)}$, де $\mathbf{s}^{(l)} = \text{col}(s_0^{(l)}, s_1^{(l)}, \dots, s_{n_l}^{(l)})$, рівна $(n_{l-1} + 1) \times (n_l + 1)$, відповідно розмірність матриці Якобі $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)} = \dim \text{diag}(f'_s(s_0^{(l)})) = (n_l + 1) \times (n_l + 1)$. Якщо покласти $f'_s(s_0^{(l)}) = 1$, то верхній діагональний елемент матриці $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)}$ буде рівний нулю. Отже, матриці

$$\left(\frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \right) = \left\| \frac{\partial s_i^{(l)}}{\partial w_{i,j}^{(l)}} \right\|, \quad i = \overline{0, n_l}, \quad j = \overline{0, n_{l-1}}; \quad \left(\frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \right) = \text{diag}(f'_s(s^{(l)}))$$

складені. Позначимо першу з них символом $\mathbf{S}^{(l)}$, другу $\mathbf{F}^{(l)}$. Елементами матриці служать приватні похідні $\partial s_i^{(l)} / \partial w_{i,j}^{(l)}$. Оскільки

$$s_i^{(l)} = \sum_{j=0}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)},$$

матриця $\mathbf{S}^{(l)}$ має наступну структуру:

$$\mathbf{S}^{(l)} = \begin{bmatrix} q_0^{(l-1)} & q_0^{(l-1)} & \dots & q_0^{(l-1)} \\ q_1^{(l-1)} & q_1^{(l-1)} & \dots & q_1^{(l-1)} \\ \dots & \dots & \dots & \dots \\ q_{n_{j-1}}^{(l-1)} & q_{n_{j-1}}^{(l-1)} & \dots & q_{n_{j-1}}^{(l-1)} \end{bmatrix} = [\mathbf{q}^{(l-1)} : \mathbf{q}^{(l-1)} : \dots : \mathbf{q}^{(l-1)}],$$

розмірність $\mathbf{S}^{(l)}$ рівна $(n_{l-1} + 1) \times (n_{l-1} + 1)$ і $\mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$.

Елементами діагональної $\mathbf{F}^{(l)}$ матриці є приватні похідні функцій активації $f(s)$ по своєму аргументу $s_i (i = \overline{0, n_l})$. Можна показати, що добуток матриць $\mathbf{S}^{(l)}$ і $\mathbf{F}^{(l)}$ зводиться до зовнішнього добутку векторів $\mathbf{q}^{(l-1)}$ і $\mathbf{f}'_s(s^{(l)}) = (f'_s(s_0^{(l)}), \dots, f'_s(s_{n_l}^{(l)}))$.

$$\mathbf{S}^{(l)} \mathbf{F}^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T.$$

Похідна скалярного функціонала $Q(\boldsymbol{\sigma})$ по векторному аргументу $\mathbf{q}^{(l)} (l = \overline{K-1, 1})$ по своєму сенсу є вектор чутливості функціонала Q до зміни вектора виходу $\mathbf{q}^{(l)}$ l -го шару мережі. Для шару l введемо вектор чутливості функціонала Q :

$$\boldsymbol{\Lambda}^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \text{col} \left(\frac{\partial Q}{\partial q_0^{(l)}}, \frac{\partial Q}{\partial q_1^{(l)}}, \dots, \frac{\partial Q}{\partial q_{n_l}^{(l)}} \right) = \text{col}(\lambda_0^{(l)}, \lambda_1^{(l)}, \dots, \lambda_{n_l}^{(l)}).$$

Приватні функції чутливості $\lambda_i^{(l)}$ обчислюються, по-перше, відповідно до правила диференціювання складних функцій; по-друге, відповідно до структури багатошарової нейронної мережі і мають вигляд:

$$\lambda_i^{(l)} = \frac{\partial Q}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \frac{\partial Q}{\partial q_j^{(l+1)}} \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \lambda_j^{(l+1)} d_{j,i}^{(l+1)} = (\boldsymbol{\Lambda}^{(l+1)})^T \mathbf{d}_i^{(l+1)}, \quad i = \overline{1, n_l}.$$

Повний вектор функцій чутливості функціонала Q до зміни вектора виходу нейронів l -го шару є

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \mathbf{D}^{(l+1)} \Lambda^{(l+1)}, \quad l = \overline{K-1, 1}. \quad (\text{B.5})$$

Матрицю $\mathbf{D}^{(l+1)}$ назвемо матрицею переходу у зворотному напрямі, оскільки для розрахунку функції $\Lambda^{(l)}$ використовуються функції чутливості $\Lambda^{(l+1)}$ і $\mathbf{D}^{(l+1)}$ подальшого шару. Матриця $\mathbf{D}^{(l+1)}$ складена з вектор-рядків $(\mathbf{d}_i^{(l+1)})^T$ де $i = \overline{0, n_l}$. Нагадаємо, що за умовчанням $q_0^{(l)} = +1$, а вагові коефіцієнти ініціалізуючих входів $w_{j,0}^{(l+1)}$ всіх базових елементів в шарі $l+1$ також настроюються. Елементами векторів за визначенням є приватні похідні:

$$\frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \frac{\partial}{\partial q_i^{(l)}} f \left(\sum_{j=0}^{n_l} w_{j,i}^{(l+1)} q_i^{(l)} \right) = \left(\frac{\partial f(s_j^{(l+1)})}{\partial s_j^{(l+1)}} \right) \left(\frac{\partial s_j^{(l+1)}}{\partial q_i^{(l)}} \right) = f'_s(s_j^{(l+1)}) w_{j,i}^{(l+1)}.$$

Для всієї множини індексів $i = \overline{0, n_l}, j = \overline{1, n_{l+1}}$ матриця переходу $\mathbf{D}^{(l+1)}$ має наступну структуру:

$$\mathbf{D}^{(l+1)} = \frac{\partial \mathbf{q}^{(l+1)}}{\partial \mathbf{q}^{(l)}} = \begin{bmatrix} \frac{\partial}{\partial q_0^{(l)}} \\ \frac{\partial}{\partial q_1^{(l)}} \\ \dots \\ \frac{\partial}{\partial q_{n_l}^{(l)}} \end{bmatrix} \times (q_0^{(l+1)}, q_1^{(l+1)}, \dots, q_{n_{l+1}}^{(l+1)}) = \begin{bmatrix} \frac{\partial q_0^{(l+1)}}{\partial q_0^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_0^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_0^{(l)}} \\ \frac{\partial q_0^{(l+1)}}{\partial q_1^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_1^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_1^{(l)}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial q_0^{(l+1)}}{\partial q_{n_l}^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_{n_l}^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_{n_l}^{(l)}} \end{bmatrix}.$$

Відзначимо, що розмірність матриці переходу $\mathbf{D}^{(l+1)}$ рівна $[(n_{l+1} + 1) \times (n_l + 1)]$. Приватні похідні в матриці $\mathbf{D}^{(l+1)}$ визначені раніше. Тепер можна скласти вектор-рядки $(\mathbf{d}_i^{(l+1)})^T$:

$$(\mathbf{d}_i^{(l+1)})^T = (f'_s(s_0^{(l+1)})w_{0,i}^{(l+1)}, f'_s(s_1^{(l+1)})w_{1,i}^{(l+1)}, \dots, f'_s(s_{n_{l+1}}^{(l+1)})w_{n_{l+1},i}^{(l+1)})$$

розмірністю $[1 \times (n_{l+1} + 1)]$. Введемо матриці:

а)

$$\mathbf{W}^{(l+1)} = [\mathbf{w}_0^{(l+1)} : \mathbf{w}_1^{(l+1)} : \dots : \mathbf{w}_{n_{l+1}}^{(l+1)}] = \|\mathbf{w}_{i,j}^{(l)}\|, \quad i = \overline{0, n_{l+1}}; \quad j = \overline{0, n_l},$$

тобто

$$\dim \mathbf{W}^{(l+1)} = (n_l + 1) \times (n_{l+1} + 1),$$

де

$$\mathbf{w}_i^{(l+1)} = \text{col}(w_{i,0}^{(l+1)}, \dots, w_{i,j}^{(l+1)}, \dots, w_{i,n_l}^{(l+1)});$$

б)

$$\mathbf{F}^{(l+1)} = \text{diag}(f'_s(\mathbf{s}^{(l+1)})) = \begin{bmatrix} f'_s(s_0^{(l+1)}) & \dots & 0 & 0 \\ 0 & f'_s(s_1^{(l+1)}) & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & f'_s(s_{n_{j-1}}^{(l+1)}) \end{bmatrix}$$

$$\dim \mathbf{F}^{(l+1)} = (n_{l+1} + 1) \times (n_{l+1} + 1).$$

Відмітимо, що для фіктивного «нульового» базового елементу $q_0^{(l+1)} = +1$ і елемент матриці $\mathbf{F}^{(l+1)} f'_s(s_0^{(l+1)}) = 0$. З урахуванням введених матриць $\mathbf{W}^{(l+1)}$ і $\mathbf{F}^{(l+1)}$ матрицю $\mathbf{D}^{(l+1)}$ запишемо у вигляді:

$$\mathbf{D}^{(l+1)} = \left\| \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} \right\|_{i=\overline{0, n_l}}^{j=\overline{0, n_{l+1}}} = (\mathbf{W}^{(l+1)})^T \mathbf{F}^{(l+1)} = (\mathbf{W}^{(l+1)})^T \text{diag}(f'_s(s^{(l+1)})). \quad (\text{B.6})$$

Тепер можна записати формулу обчислення градієнта функціонала навчання багат шарової нейронної мережі $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$ по вагових коефіцієнтах будь-якого шару $l = \overline{K-1, 1}$:

$$\nabla \mathbf{w}_i^{(l)} Q = \mathbf{S}^{(l)} \mathbf{F}^{(l)} \Delta^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T \Lambda^{(l)} \quad (\text{B.7})$$

Повний алгоритм навчання багатошарової нейронної мережі по методу ВР включає початковий алгоритм настройки вагових коефіцієнтів K -го шару (В.4) і алгоритми настройки базових елементів шарів $l = \overline{K-1, 1}$ використовують формулу (В.7).

В.3.3 Алгоритм навчання багатошарової нейронної мережі по методу зворотного розповсюдження помилки.

Приведемо алгоритм навчання нейронної мережі методом зворотного розповсюдження помилки в одній з можливих форм, витікаючої з прийнятої методики виводу алгоритму:

а)

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k+1))e_{ui}(k+1)\mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i^{(K)}(k+1)\mathbf{q}^{(K-1)}(k+1); \\ \gamma &> 0; \quad k = 0, 1, 2, \dots; \quad e_{ui} = u_i^* - u_i; \quad i \in \overline{1, n_K}; \\ \mathbf{q}^{(K-1)} &= \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_j^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}); \end{aligned} \quad (\text{В.8})$$

б)

$$\begin{aligned} \mathbf{w}_i^{(l)}(k+1) &= \mathbf{w}_i^{(l)}(k) - \gamma \mathbf{q}^{(l-1)}(k+1)(\mathbf{f}'_s(s_i^{(l)}(k+1)))^T \Lambda^{(l)}(k); \\ \Lambda^{(l)}(k) &= \mathbf{D}^{(l+1)} \Lambda^{(l+1)}(k); \\ \mathbf{D}^{(l+1)}(k) &= (\mathbf{W}^{(l+1)}(k))^T \text{diag}(f'_s(\mathbf{s}^{(l)}(k))); \\ \gamma &> 0; \quad l = \overline{K-1, 1}; \quad \mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, q_1^{(l-1)}, \dots, q_{l_{K-1}}^{(l-1)}); \quad i = \overline{0, n_l} \end{aligned} \quad (\text{В.9})$$

Зокрема, алгоритм настройки БПЕ шару $K-1$ у відповідності до пункту б) має вид

$$\begin{aligned} \mathbf{w}_i^{(K-1)}(K+1) &= \mathbf{w}_i^{(K-1)}(K) - \gamma \mathbf{q}^{(K-2)}(K+1)(\mathbf{f}'_s(s_i^{(K-1)}(K+1)))^T \Lambda^{(K-1)}(K), \\ \Lambda^{(K-1)}(K) &= \mathbf{D}^{(K)}(K) \Lambda^{(K)}(K); \\ \mathbf{D}^{(K)}(K) &= \mathbf{W}^{(K)}(K) \text{diag}(f'_s(\mathbf{s}^{(K)}(K))) = \mathbf{W}^{(K)}(K); \\ \Lambda^{(K)}(K) &= \text{col}\left(\frac{\partial Q}{\partial \mathbf{q}_1^{(K)}}, \dots, \frac{\partial Q}{\partial \mathbf{q}_{n_K}^{(K)}}\right) = -\mathbf{e}_u(K). \end{aligned}$$

тоді

$$\begin{aligned}\Lambda^{(K-1)}(k) &= -\mathbf{W}^{(K)}(k)\mathbf{e}_u(k), \\ (\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k) &= \\ &= -(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \mathbf{W}^{(K)}(k)\mathbf{e}_u(k) = \sigma_1^{(K-1)}(k+1)\end{aligned}$$

– скалярна узагальнена помилка навчання нейросети для i -го базового елемента в шарі $K-1$. З її використанням отримуємо алгоритм настройки, формою в точності відповідний δ -правилу:

$$\begin{aligned}\mathbf{w}_i^{(K-1)}(k+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \sigma_i^{(K-1)}(k) \mathbf{q}^{(K-2)}(k+1); \\ \sigma_i^{(K-1)}(k+1) &= -(\mathbf{f}'_s(s_i^{(K-1)}(k)))^T \mathbf{W}^{(K)}(k) \mathbf{e}^{(K)}(k+1); \\ \mathbf{e}_u^{(K)}(k) &= \mathbf{u}^*(k) - \mathbf{q}^{(K)}(k).\end{aligned}\tag{B.10}$$

Аргумент k позначає момент використання відповідних значень заданих множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$ на k -й ітерації обчислень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ на «уроках» навчання, число яких залежить від числа шарів мережі і кількості нейронів в шарах.

Фаза навчання завершена, коли досягнутий для заданих в своєму класі множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$. Мережа фіксує цей стан в значеннях своїх вагових коефіцієнтів і відповідних «віртуальних» зсувах центрів симетрії функцій активації і їх похідних активованих базових елементів і зберігає цей стан після зняття сигналу $\mathbf{q}^{(0)} = \mathbf{r}$ на входах мережі а виді віртуальної статичної нелінійної характеристики, зібраної з фрагментів сигмоїдних функцій. Якщо тепер па входах пред'явити ту ж множину або близьку до неї з того ж класу, то нейронна мережа практично миттєво відновить на виході ту ж множину $\mathbf{u} = \mathbf{q}^{(K)}$, що і після фази навчання. Цей етап функціонування мережі називають фазою спогаду. Мережа, таким чином, виконує функцію асоціативної пам'яті. Якщо ж множини $\mathbf{u}^*$, $\mathbf{q}^{(0)} = \mathbf{r}$ є функціями часу: $\mathbf{q}^{(0)} = \mathbf{r}(t)$, $\mathbf{u}^* = \mathbf{u}^*(t)$, то стандартний алгоритм ВР

стає неавтономним і вимагає завдання всієї множини значень змінних $\mathbf{r}(t)$ і $\mathbf{u}^*(t)$. Неминуче зростання числа шарів і базових елементів в них і пов'язане з цим збільшення тривалості навчання викликає необхідність забезпечення стійкості процесу настройки мережі по алгоритму (В.8) (В.9).

Очевидно, що в завданнях управління $\mathbf{u}^*(t)$ – функція оптимального управління динамічним об'єктом. Управління за допомогою багатошарової нейронної мережі в двохетапному режимі можливо, якщо тільки функція оптимального управління наперед розрахована і може бути визначена вся множина значень помилки навчання мережі $\mathbf{e}_u(t)$ на інтервалі її визначення. Використання іншої вимірjuвальної інформації, наприклад помилки відтворення заданої функції $\mathbf{r}(t)$ на виході динамічного об'єкту, вносить запізнювання до алгоритму настройки і приводить до нестійкого процесу. Виникає проблема синтезу динамічних алгоритмів навчання багатошарової нейронної мережі, що враховують вплив інерційності, що вноситься до процесу настройки по алгоритму ВР.

В.4 Перенавчання і узагальнення нейронних мереж

Одна з найбільш серйозних труднощів викладеного підходу полягає в тому, що таким чином ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати, – помилку, яку можна чекати від мережі, коли їй подаватимуться абсолютно нові спостереження. Інакше кажучи, ми хотіли б, щоб нейронна мережа володіла здатністю узагальнювати результат на нові спостереження. Насправді мережа навчається мінімізувати помилку на навчальній множині, і у відсутність ідеальної і нескінченно великої навчальної множини це зовсім не те ж саме, що мінімізувати «справжню» помилку на поверхні помилок в наперед невідомій моделі явища.

Найсильніше ця відмінність виявляється в проблемі перенавчання. При дуже близькій підгонці явище простіше буде продемонструвати не для нейронної мережі, а на прикладі апроксимації за допомогою поліномів, при цьому суть явища абсолютно та ж.

Поліном (або многочлен) – це вираз, що містить тільки константи і цілі ступені незалежної змінної. Графіки поліномів можуть мати різну форму. Чим вище ступінь многочлена (і, тим самим, ніж більше членів в нього входить), тим більше складною може бути ця форма. Якщо у нас є деякі дані, ми можемо поставити мету підігнати до них поліноміальну криву (модель) і отримати, таким чином, пояснення для наявної залежності. Проте наші дані можуть бути зашумлені, тому не можна вважати, що сама краща модель задається кривою, яка в точності проходить через все наявні крапки. Поліном низького порядку може бути недостатньо гнучким засобом для апроксимації даних, тоді як поліном високого порядку може виявитися занадто гнучким і буде точно слід даним, приймаючи при цьому хитромудру форму, що не має ніякого відношення до форми справжньої залежності.

Нейронна мережа стикається з точно такою ж трудностю. Мережі з великим числом вагів моделюють складніші функції і, отже, схильні до перенавчання. Мережа ж з невеликим числом вагів може опинитися недостатньо гнучкою, щоб змодельовати наявну залежність. Наприклад, мережа без проміжних шарів насправді моделює звичайну лінійну функцію.

Як же вибрати «правильний» ступінь складності для мережі? Складніша мережа майже завжди дає меншу помилку, але це може свідчити не про хорошу якість моделі, а про перенавчання.

Відповідь полягає в тому, щоб використовувати механізм контрольної перекресної перевірки (крос-перевірки), при якому частина навчальних спостережень резервується і не використовується в процесі навчання по алгоритму зворотного розповсюдження. Натомість у міру роботи алгоритму вона використовується для незалежної перевірки результату. На самому початку роботи помилка мережі на навчальній і перевірочній множинах буде однаковою, але у міру того як мережа навчається, помилка навчання, природно, убиває, і, поки навчання зменшує дійсну функцію помилок, помилка на контрольній (перевірочній) множині також убиватиме. Якщо ж контрольна помилка перестала убивати або навіть стала рости, це указує на те, що мережа почала дуже близько апроксимувати дані і навчання слід зупинити.

Відмічене явище занадто точної апроксимації в процесі навчання і називається перенавчанням. Якщо таке трапляється, то рекомендується зменшити число прихованих елементів і/або шарів, бо мережа є дуже могутньою для даного завдання. Якщо ж мережа, навпаки, була узята недостатньо складною для того, щоб моделювати наявну залежність, то перенавчання, швидше за все, не відбудеться, і обидві помилки – навчання і перевірки – не досягнуть достатнього рівня трохи.

Описані проблеми з локальними мінімумами і вибором розміру мережі призводять до того, що при практичній роботі з нейронними мережами, як правило, доводиться експериментувати з великим числом різних мереж, деколи навчаючи кожную з них по кілька разів (щоб не бути введеним в оману локальними мінімумами) і порівнюючи отримані результати. Головним показником якості результату тут є контрольна помилка. При цьому, відповідно до загальнонаукового принципу, згідно якому при інших рівних слід віддати перевагу простішій моделі, з двох мереж з приблизно рівними помилками контролю має сенс вибрати ту, яка менше.

Необхідність багатократних експериментів веде до того, що перевірюча множина починає грати ключову роль у виборі моделі, тобто стає частиною процесу навчання. Тим самим ослабляється її роль як незалежного критерію якості моделі – при великому числі експериментів є ризик вибрати «вдалу» мережу, що дає добрий результат на перевірючій (контрольному) множині.

Для того, щоб додати остаточній моделі належну надійність, часто (принаймні, коли об'єм навчальних даних це дозволяє) поступають так: резервують ще одну – тестову множину спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на навчальній і контрольній множинах, реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки один раз: якщо її використовувати повторно для коректування процесу навчання, то вона фактично перетвориться на контрольну множину.

ДОДАТОК Г

ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання нелінійних динамічних систем є отримання їх адекватних математичних моделей, який базується, як правило, на теоретичному і експериментальному аналізі властивостей цих систем. Теоретичний аналіз, заснований на вивченні фізичних і хімічних процесів, які відбуваються в системі, дозволяє отримати математичний опис, наприклад, диференціальних рівнянь. При експериментальному аналізі на основі спостережень вхідних і початкових сигналів системи отримують або її параметричну, або непараметричну модель. Найбільшого розповсюдження придбали параметричні моделі, які вимагають рішення задач структурної і параметричної ідентифікації і що використовують обмежену кількість параметрів. Не дивлячись на величезну кількість робіт, різноманіття видів нелінійностей не дозволяє створити єдину теорію ідентифікації нелінійних систем. Вживаний найчастіше класичний підхід заснований на апроксимації нелінійностей, наприклад рядами Вольтерра, Гаммерштейна, Вінера, поліномами Колмогорова-Габора і ін., проте сфера застосування таких моделей обмежена. Крім того, додаткові труднощі отримання адекватного математичного опису зумовлює наявність в реальних сигналах завад, яка вимагає попередньої фільтрації сигналів.

Розглянемо завдання ідентифікації нелінійного динамічного об'єкту, що описується NARMAX-моделлю.

$$\tilde{y}(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n), k] + \xi(k), \quad (\text{Г.1})$$

де $\tilde{y}(i)$, $u(i)$ – вихідний і вхідний сигнали об'єкту у момент часу i відповідно;

m , n – порядки запізнювання по вихідним і вхідним каналами відповідно;

$f(\cdot)$ – невідома нелінійна функція;

ξ – завада вимірювання вихідного сигналу.

Введемо вектор узагальненого вхідного сигналу розмірністю $(n + m) \times 1$ так, що:

$$x(k) = [\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-m), u(k-1), u(k-2), \dots, u(k-n)]. \quad (\Gamma.2)$$

Тоді рівняння (Г.1) може бути записане таким чином:

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (\Gamma.3)$$

Завдання ідентифікації полягає в оцінюванні функції $f(\cdot)$ по вимірюваннях вхідних $u(k)$ і вихідних $\tilde{y}(k)$ змінних.

Схема ідентифікації нелінійного об'єкту (Г.1) на підставі ШНМ приведена на рис. Г.1.

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (Г.1) сьогодні отримали багат шаровий персептрон (рис. Г.2) і радіально-базисні мережі (РБМ) (рис. Г.3), що використовують апроксимацію нелінійного оператора $f(\cdot)$ нейронною мережею.

Рівняння БП має вигляд

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (\Gamma.4)$$

де q – кількість куль в мережі;

ω^i – вектор вагових параметрів нейронів i -го кулі мережі;

$f(\cdot)$ – функція активації i -го шару.

Модель у вигляді радіально базової мережі РБМ, яка використовує для апроксимації $f(\cdot)$ деякі базисні функції $\Phi_i(x)$ може бути представлена таким чином:

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (\Gamma.5)$$

де c_i – вагові коефіцієнти, які підлягають визначенню.

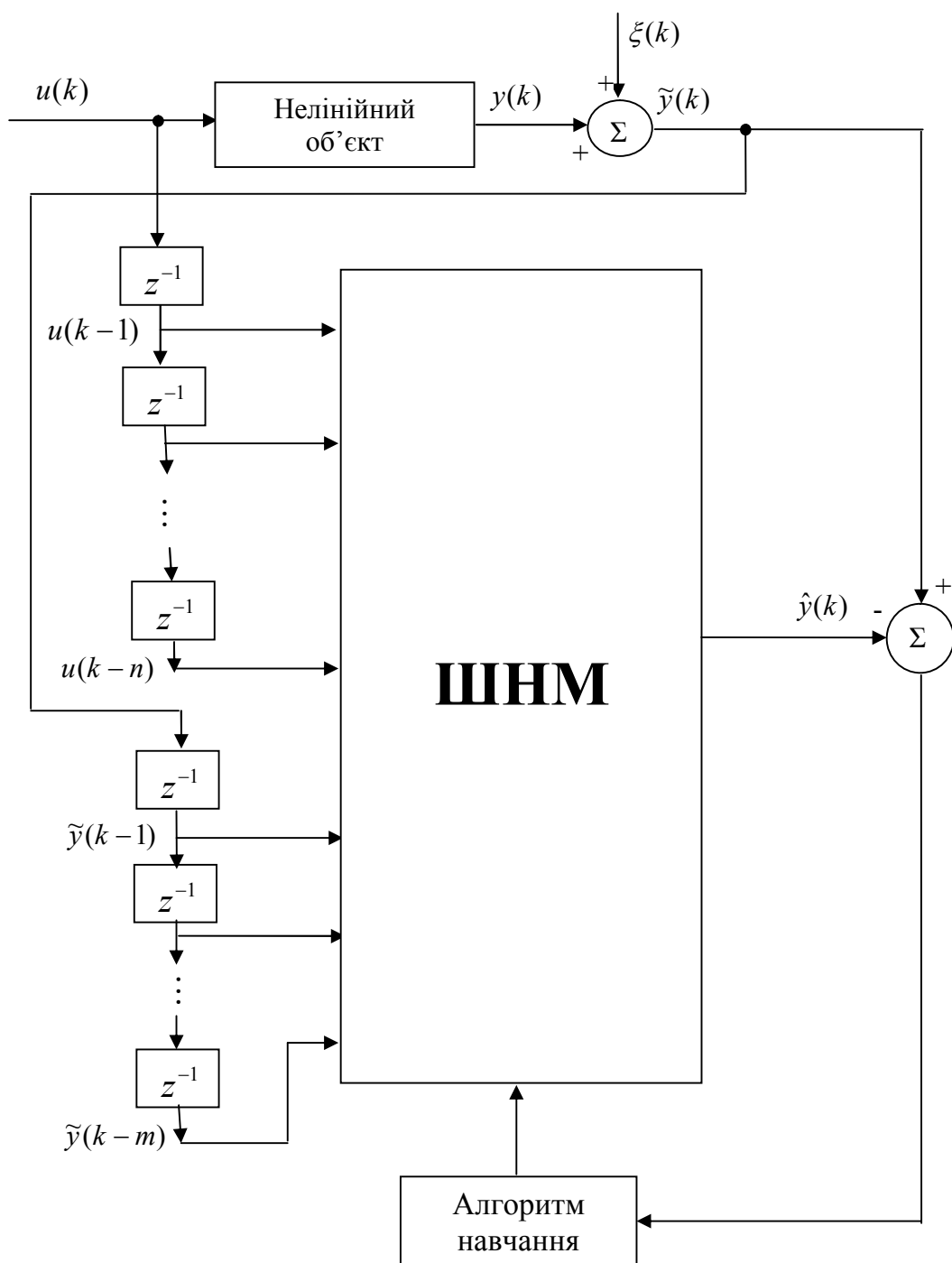


Рисунок Г.1 – Схема ідентифікації динамічного об'єкту

Якщо як базисну функцію $\Phi_i(x)$ вибрати

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (\text{Г.6})$$

де μ_i, σ_i – центри і радіуси базисних функцій відповідно;

$\|\cdot\|$ – евклідова норма, то визначенню підлягає вектор

$$\omega(k) = (c_0(k), c_1(k), \mu_1^T(k), \sigma_1(k), \dots, c_N(k), \mu_N^T(k), \sigma_N(k))^T,$$

який містить всі невідомі параметри мережі (зазвичай приймають $\Phi_0(x) = 1$).

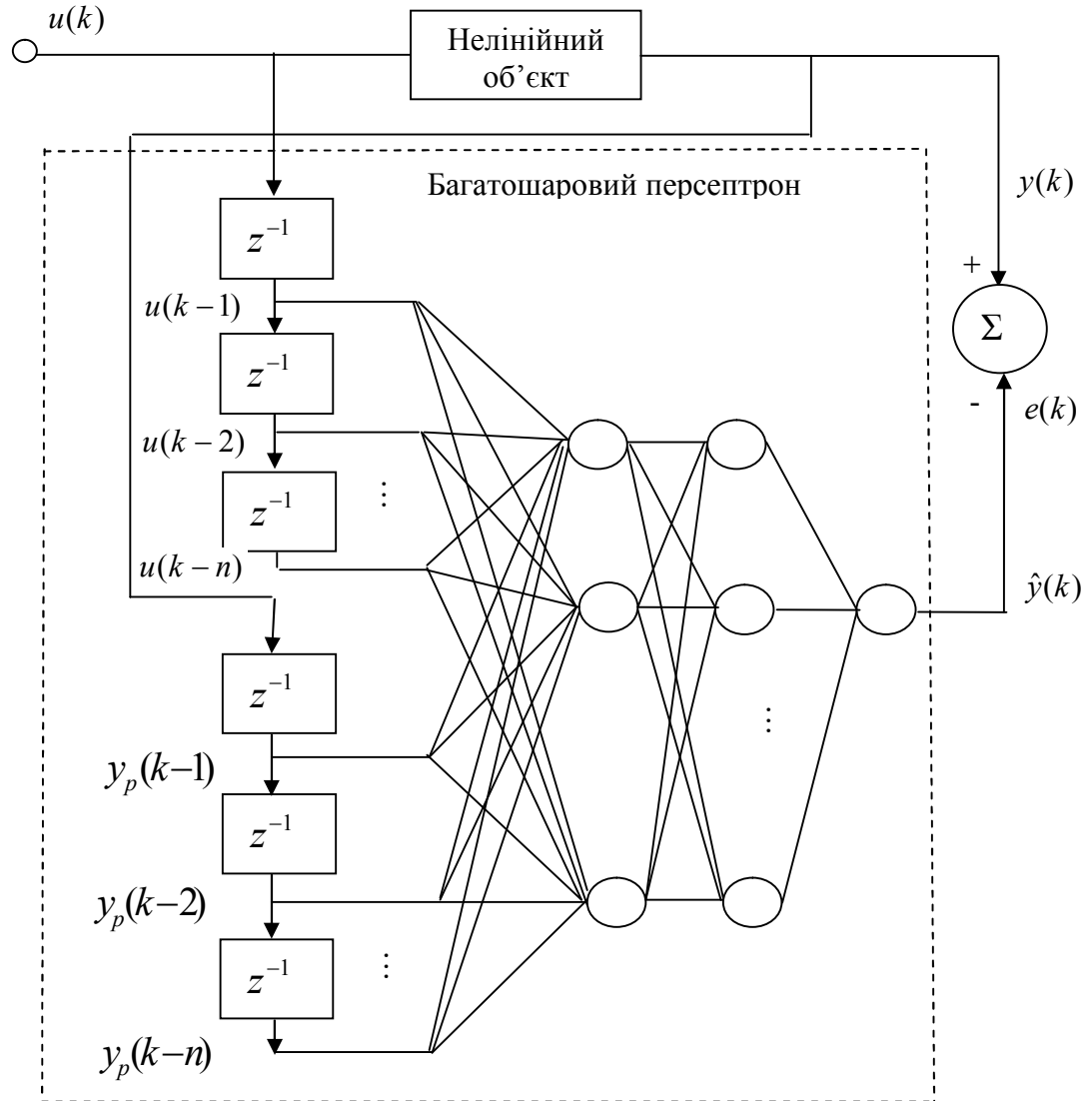


Рис Г.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

Завдання ідентифікації полягає в навчанні мережі, тобто в такому на-
строюванні її параметрів, які б забезпечували мінімум функціоналу

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (\Gamma.7)$$

Навчання мережі забезпечується за допомогою якого-небудь методу мінімізації функціонала якості.

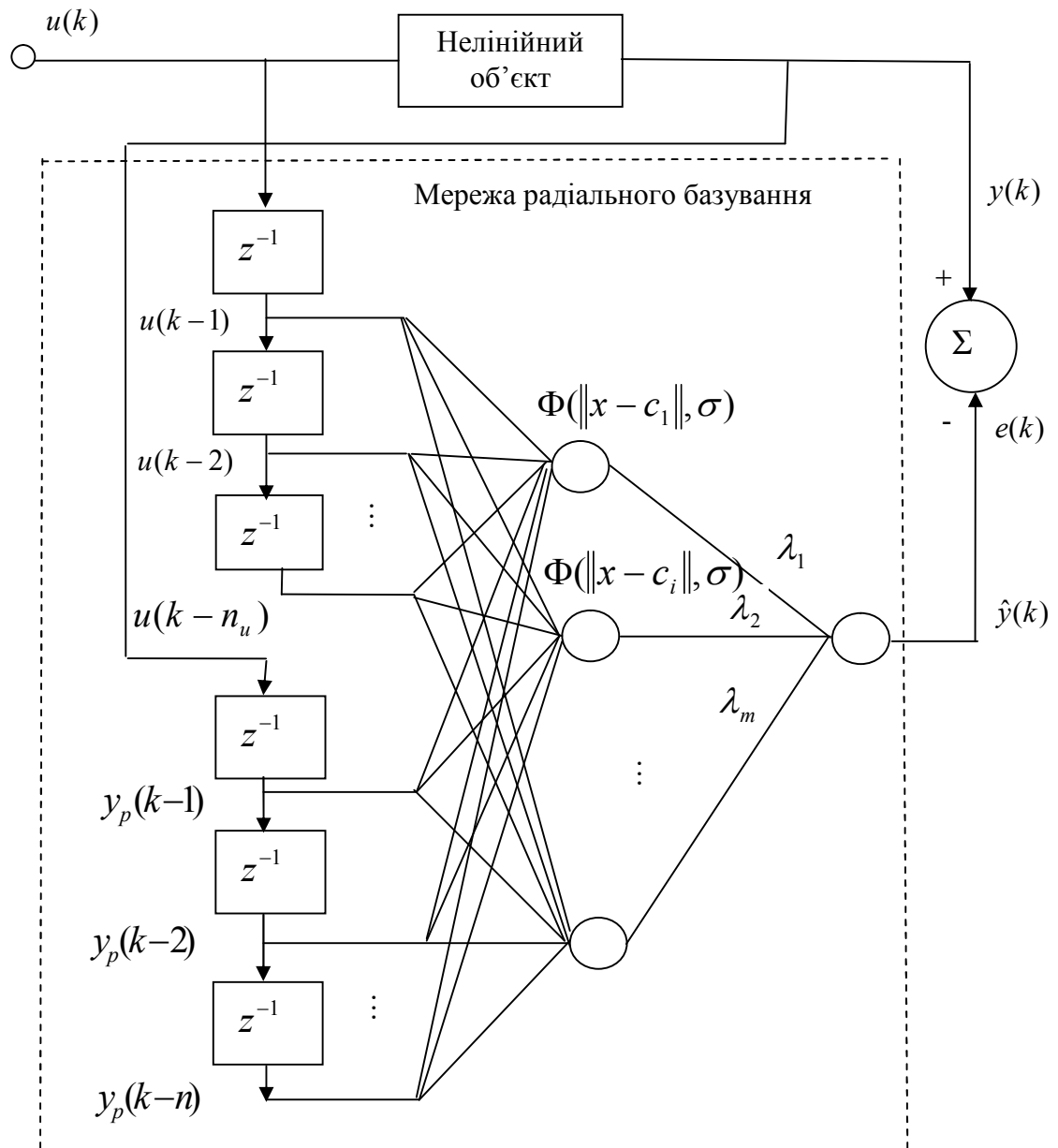


Рисунок Г.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом обертання
% роторного колеса екскаватора
% 2023 рік

% Вихідні дані
kzv=1.33;
ks=5.89;
Tg=0.55;
Tzc=0.01;
Te=0.084;
Re=0.03812;
CF=8.02;
In=380;
Mn=CF*In;
Jd=380;
Jm=104;
Js=Jd+Jm;
w0=2;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);
b12=0.1;
Uz=184;
kcx=ks/(CF+ks*kzv);

% Матриці
A=[-b12/Jm 1/Jm b12/Jm 0 0;
-c12 0 c12 0 0;
b12/Jd -1/Jd -b12/Jd CF/Jd 0;
0 0 -CF/(Te*Re) -1/Te 1/(Te*Re) ;
0 0 (CF*ks*Tzc/(Te*Re*Tg)-kzv*ks/Tg) ks*Tzc/(Te*Tg) (-
ks*Tzc/(Te*Re*Tg)-1/Tg)];
B=[0 -1/Jm;
zeros(3,2);
ks/Tg 0];
C=zeros(5,5);
C(1,1)=1;C(2,2)=1; C(3,3)=1;C(4,4)=1;C(5,5)=1;
D=zeros(5,2);
```