

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

УПРАВЛІННЯ БАГАТОМАСОВИМИ ЕЛЕКТРОМЕХАНІЧНИМИ
СИСТЕМАМИ ОСНОВНИХ МЕХАНІЗМІВ РОТОРНОГО
ЕКСКАВАТОРА З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ.
ТЕМА КОМПЛЕКСНА. СПЕЦІАЛЬНА ЧАСТИНА.
НЕЙРОМЕРЕЖЕВА СИСТЕМА УПРАВЛІННЯ МЕХАНІЗМОМ
РОЗВОРОТУ РОТОРНОГО ЕКСКАВАТОРА

(тема кваліфікаційної роботи)

Виконав: студент 2 курсу, групи ДЕА-А24мг
спеціальності: 174 Автоматизація, комп'ютерно-
інтегровані технології та робототехніка

(код і найменування спеціальності)

_____ / Іван НАДТОЧИЙ
(підпис) (ім'я та прізвище)

Керівник _____ / Геннадій КАНЮК
(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 – 2025

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)
« » 2025р.

ЗАВДАННЯ

на кваліфікаційну роботу магістерського рівня вищої освіти

студенту (ці) Іван НАДТОЧИЙ
(ім'я, прізвище)

1. Тема роботи Управління багатомасовими електромеханічними системами основних механізмів роторного екскаватора з застосуванням нейронних мереж. Тема комплексна. Спеціальна частина. Нейромережева система управління механізмом розвороту роторного екскаватора

затверджена наказом по університету № 4801-5/3664 від “06” жовтня 2025 р.

2. Термін здачі закінченої роботи “5” грудня 2025 р.

Теоретична продуктивність по рихлій масі $Q=3900\text{ м}^3/\text{год}$; діаметр роторного колеса $D_p=12\text{ м}$; бічне зусилля різання $F_6=12\cdot 10^4\text{ Н}$; питоме зусилля копання $k_f=1.1\cdot 10^6\text{ кг/м}^2$; радіус різання $R_0=37\text{ м}$; кутова швидкість повороту платформи $\omega_{пл}=0.6\cdot 10^{-2}\text{ с}^{-1}$; маса поворотної платформи $m_{пл}=1000\cdot 10^3\text{ кг}$; частота обертання ротора $\omega_p=5\text{ рад/хв}$; момент інерції поворотної платформи $J_{пл}=88\cdot 10^7\text{ кг.м}^2$; діаметр роликвого круга $D_{кр}=14\text{ м}$. 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 розвороту роторного екскаватора. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити статичні і динамічні характеристики системи. Розробити математичну модель двомасової системи. Виконати моделювання на ЕОМ та провести аналіз показників якості перехідних процесів змінних стану двомасової системи. 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. Розворот екскаватора. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 6. Нейромережева ідентифікація. NARX моделі. 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 2025 р.

Керівник

(підпис)

Геннадій КАНЮК.
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Іван НАДТОЧИЙ
(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 вико- нання за- вдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму розвороту роторного екскаватора.	09.09.2025- 18.09.2025	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження статичних та динамічних характеристик системи.	19.09.2025- 15.10.2025	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ПЕОМ.	16.10.2025- 05.11.2025	
4	Синтез нейромережової системи управління на ПЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2025- 25.11.2025	
5	Оформлення роботи	26.11.2025- 05.12.2025	

Студент (ка)

(підпис)

Іван НАДТОЧИЙ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 226 сторінок, 75 рисунків, 2 таблиці, 80 використаних джерел, 5 додатків.

Метою дослідження є розроблення підходів до синтезу системи керування електроприводом механізму обертання стріли роторного екскаватора з орієнтацією на підвищення якості роботи системи шляхом застосування нейромережевих регуляторів.

Параметри системи керування визначено аналітично з використанням основних принципів теорії автоматичного регулювання. Для побудови моделей багатомасової електромеханічної системи застосовано методи математичного моделювання. На основі апарату штучних нейронних мереж виконано синтез нейрорегулятора, призначеного для керування двомасовою електромеханічною структурою. За допомогою пакета MATLAB проведено комп'ютерне моделювання та досліджено динамічні властивості нейромережевої системи.

У межах магістерської роботи обґрунтовано доцільність використання нейромережевих технологій для керування багатомасовою електромеханічною системою механізму розвороту роторного екскаватора. Розроблено систему регулювання координат електропривода, визначено параметри автоматичної системи керування та виконано її комп'ютерне моделювання.

Створено математичну модель електропривода з урахуванням пружних елементів механічної частини та отримано динамічні показники двомасової системи.

Проаналізовано архітектуру нейромережевої системи та реалізовано синтез нейрорегулятора типу NN Predictive Controller у середовищі MATLAB. Проведене моделювання підтвердило високі якісні характеристики синтезованої системи керування.

Результати та основні напрацювання роботи впроваджуються у навчальний процес кафедри АМЕТ ННІ «УПА» ХНУ ім. В.Н. Каразіна під час викладання лекцій, розроблення курсових проєктів і проведення практичних занять.

Наукові висновки та отримані результати представлені на студентській науково-практичній конференції ННІ «УПА» (Харків, 2025 рік).

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

ABSTRACT

An explanatory message consists of: 226 pages, 75 figures, 2 tables, 80 used sources, 5 additions.

The purpose of the study is to develop approaches to the synthesis of the control system for the electric drive of the rotary excavator boom rotation mechanism with an orientation towards improving the quality of the system's operation through the use of neural network regulators.

The parameters of the control system are determined analytically using the basic principles of the theory of automatic regulation. Mathematical modeling methods are used to build models of the multi-mass electromechanical system. Based on the apparatus of artificial neural networks, a neural regulator designed to control a two-mass electromechanical structure was synthesized. Using the MATLAB package, computer modeling was performed and the dynamic properties of the neural network system were investigated.

Within the framework of the master's thesis, the feasibility of using neural network technologies to control the multi-mass electromechanical system of the rotary excavator turning mechanism is substantiated. The system for regulating the coordinates of the electric drive was developed, the parameters of the automatic control system were determined and its computer simulation was performed.

A mathematical model of the electric drive was created taking into account the elastic elements of the mechanical part and the dynamic indicators of the two-mass system were obtained.

The architecture of the neural network system was analyzed and the synthesis of a neural regulator of the NN Predictive Controller type was implemented in the MATLAB environment. The simulation confirmed the high quality characteristics of the synthesized control system.

The results and main developments of the work are implemented in the educational process of the Department of AMET of the Education and Research Institute "UIPA" of the V.N. Karazin KhNU during lectures, development of course projects and practical classes.

Scientific conclusions and the results obtained are presented at the student scientific and practical conference of the Education and Research Institute "UIPA" of the V.N. Karazin KhNU (Kharkiv, 2025).

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

P_n – потужність, що відповідає номінальному режиму роботи;

U_n – напруга, встановлена як номінальна;

I_n – струм у номінальних умовах;

M_n – номінальний момент;

n_n – швидкість обертання при номінальному навантаженні;

Φ_n – магнітний потік у номінальному режимі;

R – активний опір елемента;

L – величина індуктивності;

J – момент інерції вузла;

k_d – коефіцієнт підсилення електричного двигуна;

T_e – електромагнітна стала часу електропривода;

T_m – електромеханічна стала часу електропривода;

$W(p)$ – передавальна функція;

c – коефіцієнт жорсткості;

β – коефіцієнт внутрішнього в'язкого тертя;

A – матриця, що описує стан системи;

B – матриця, яка визначає вплив керуючих дій;

C – матриця виходу;

$\vec{X}(t)$ – вектор, що характеризує стан системи;

$\vec{U}(t)$ – вектор керувальних впливів;

$\vec{Y}(t)$ – вектор вихідних сигналів;

САУ – система автоматичного управління;

М – електродвигун.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. ОГЛЯД СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	12
1.1 Сучасні області застосування нейронних мереж.....	12
1.2 Огляд структур нейромережевих систем регулювання т а обґрунтування вибору архітектури	20
1.3 Загальні відомості про роторні екскаватори	41
1.4 Аналіз факторів, що визначають вибір системи управління електроприводом механізму розвороту роторного екскаватора.....	44
1.5 Постановка завдання синтезу нейромережевої системи управління ...	47
ВИСНОВКИ ДО РОЗДІЛУ 1	50
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ РОЗВОРОТУ РОТОРНОГО ЕКСКАВАТОРА	52
2.1 Основні технічні характеристики роторного екскаватора та вимоги до системи управління.....	52
2.2 Розрахунок потужності та вибір електродвигунів механізму розвороту роторного екскаватора	54
2.3 Розрахунок швидкісної діаграми механізму розвороту	58
2.4 Аналіз навантажувальної діаграми електроприводу та перевірка вибору двигунів	61
2.5 Вибір силового електроустаткування електроприводу	67
2.6 Аналіз і розрахунок статичних характеристик розімкненої та замкненої системи електроприводу.....	70
2.10 Дослідження динамічних характеристик системи	74
ВИСНОВКИ ДО РОЗДІЛУ 2	96
РОЗДІЛ 3 ХАРАКТЕРИСТИКИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ.....	97
3.1 Математична модель двомасової системи	97
3.2 Розрахунок перехідних процесів двомасової системи.....	104

ВИСНОВКИ ДО РОЗДІЛУ 3	111
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	112
4.1 Регулятори, реалізовані засобами пакета Neural Network Toolbox у середовищі MATLAB	112
4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller	114
4.3 Проєктування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB	117
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	162
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	164
ВИСНОВКИ ДО РОЗДІЛУ 4	170
ВИСНОВКИ	172
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	174
ДОДАТОК А. СТРУКТУРА ШТУЧНОГО НЕЙРОНА	182
ДОДАТОК Б. КЛАСИФІКАЦІЯ НЕЙРОННИХ МЕРЕЖ І ЇХ ВЛАСТИВОСТІ.....	188
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	196
В.1 Сутність процесу навчання	196
В.2 Алгоритми навчання нейронних мереж.....	198
В.3 Метод зворотного розповсюдження помилки.....	202
В.4 Перенавчання і узагальнення нейронних мереж.....	212
ДОДАТОК Г. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ.....	215
Г.1 Моделі систем, використовувані при нейромережевій ідентифікації	215
Г.2 Нейронні мережі як універсальні моделі	220
Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж	223
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	226

ВСТУП

Актуальність теми. Сучасний етап розвитку гірничодобувної галузі характеризується активним розширенням відкритих способів видобутку корисних копалин, що зумовлено необхідністю підвищення продуктивності та оптимізації витрат. Центральною ланкою технологічного ланцюга при відкритих розробках виступає роторний екскаватор – високопродуктивна машина безперервної дії, здатна забезпечити значні обсяги переміщення гірничої маси. Робота екскаватора є результатом злагодженої взаємодії кількох електромеханічних механізмів: роторного колеса з багатоковшовою системою, підйомного механізму стріли, виконавчого механізму повороту, ходової частини, а також розгалуженої транспортної системи стрічкових конвеєрів.

Застосування індивідуальних електроприводів для кожного технологічного вузла дозволяє не тільки автоматизувати процеси, але й забезпечує стабільну роботу обладнання, своєчасне виявлення аварійних ситуацій, зменшення навантаження на персонал та істотне підвищення загальної ефективності виробничого циклу.

Підвищення рівня механізації, збільшення потужності гірничих машин та впровадження систем комплексної автоматизації є ключовою передумовою переходу до принципово нових форм організації виробництва. У таких умовах роль оператора зводиться переважно до функцій контролю та прийняття управлінських рішень, тоді як рутинні процеси делегуються автоматичним системам керування.

Упродовж останніх десятиліть у теорії та практиці автоматизації з'явилася велика кількість підходів до побудови систем керування, включаючи методи, що базуються на адаптивних моделях. Розвиток теорії стійкості та регулювання дозволив створити системи, які змінюють свої властивості залежно від зовнішніх умов і поточного стану об'єкта. Проте ефективність адаптивного керування визначається точністю математичної моделі, яку у більшості випад-

ків не вдається побудувати з необхідною повнотою через складність об'єкта та зовнішні невизначеності.

Одним із шляхів подолання цих обмежень є інтеграція методів адаптивного керування з технологіями штучних нейронних мереж. Нейронні мережі природно пристосовані до роботи з нелінійними залежностями, здатні до самонавчання, узагальнення досвіду та функціонування в умовах неповноти інформації.

Їх ключовими перевагами є:

- паралельна обробка сигналів, яка забезпечує високу швидкодію;
- стійкість до пошкоджень окремих елементів мережі та можливість компенсації їхнього впливу на результат;
- здатність до навчання та адаптації, що робить такі структури ефективними у задачах керування складними динамічними об'єктами;
- можливість апаратної реалізації за допомогою технологій надвеликої інтеграції, що відкриває перспективи створення універсальних обчислювальних модулів з однаковою структурою.

З поширенням високопродуктивних обчислювальних систем інтерес до нейронних мереж різко зріс, а методи їх застосування стали основою новітніх рішень у машинному баченні, прогнозуванні, адаптивному регулюванні, аналізі сигналів та управлінні складними технічними комплексами [1–12]. У системах керування такі моделі можуть використовуватися як ідентифікатори стану, додавачі адаптації, оптимізатори параметрів або безпосередні регулятори.

Враховуючи складність та багатомасовість механізму розвороту роторного екскаватора, використання нейромережевих методів у системах його керування є науково обґрунтованим та технологічно доцільним. Це визначає актуальність і практичну значущість даної наукової роботи.

Мета і завдання дослідження. Головною метою є розроблення методології синтезу системи керування електроприводом механізму розвороту роторного екскаватора із застосуванням нейромережевих регуляторів, що здатні забезпечити підвищені показники точності, стійкості та швидкодії.

Для досягнення поставленої мети необхідно виконати такі **завдання**:

1. Провести аналіз сучасних систем керування та оцінити можливості впровадження нейромережевих технологій.
2. Розробити та уточнити математичну модель динаміки об'єкта керування.
3. Сформулювати структурну схему системи керування з використанням нейромережевого регулятора.
4. Виконати синтез нейромережевої системи управління із застосуванням прогнозуючого нейрорегулятора.
5. Змодельовати роботу системи на ЕОМ та провести всебічний аналіз отриманих результатів.

Об'єкт дослідження – динамічні процеси в системі керування електроприводом механізму повороту роторного екскаватора, що містить пружні елементи.

Предмет дослідження – методи побудови та синтезу нейромережевих систем управління двомасовими електромеханічними системами механізму розвороту.

Методи дослідження. Робота спирається на положення теорії автоматичного керування, математичне моделювання динамічних систем та інструменти теорії штучних нейронних мереж. Для розроблення алгоритмів керування застосовано методи нейроуправління, а якість функціонування системи оцінювалася за допомогою імітаційного моделювання в середовищі MATLAB.

Наукова новизна. У процесі виконання роботи отримано такі результати, що містять елементи наукової новизни:

1. Вперше обґрунтовано доцільність застосування нейромережевих алгоритмів для керування багатомасовою системою поворотного механізму роторного екскаватора.
2. Розроблено та синтезовано нову нейромережеву систему керування електроприводом механізму розвороту з урахуванням пружних ланок.

3. Запропоновано нову математичну модель системи у формі багатошарового персептрона.

4. Отримано нові результати щодо моделювання двомасових електромеханічних систем із складними кінематичними зв'язками.

5. Розвинено методи синтезу нейромережевих структур прогнозного типу для керування складними нелінійними об'єктами.

Практичне значення. Практична цінність роботи полягає у створенні нейромережевої системи керування двомасовим електроприводом, яка забезпечує високу точність регулювання та може бути впроваджена в структуру керування роторного екскаватора. Результати дослідження застосовуються у навчальному процесі кафедри АМЕТ ННІ «УІПА» ХНУ ім. В.Н. Каразіна.

Особистий внесок здобувача. Усі результати, представлені до захисту, отримані автором самостійно. До індивідуальних напрацювань належать: синтез системи регулювання координат привода, створення математичної моделі з урахуванням пружних елементів, побудова структурної схеми нейромережевої системи, розроблення нейрорегулятора NN Predictive Controller, моделювання всієї системи, економічне обґрунтування її впровадження та розробка заходів із забезпечення безпеки праці.

Апробація результатів. Основні результати та наукові положення оприлюднені на студентській науковій конференції ННІ «УІПА» ХНУ ім. В.Н. Каразіна (Харків, 2025 р.).

Структура та обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів, висновків, додатків та списку використаних джерел. Загальний обсяг становить 226 сторінок. Робота містить 75 рисунки, з яких 42 наведено в основному тексті, а 42 – на 36 окремих сторінках; 2 таблиці, з яких 1 на 2 окремих сторінках, 5 додатків обсягом 45 сторінок та список літературних джерел із 80 позицій, розміщений на 8 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Сучасні області застосування нейронних мереж

Поняття «нейронні мережі» сформувалося ще у 40-х роках ХХ століття в колі дослідників, які намагалися зрозуміти принципи роботи та організацію природних нервових структур. Основоположні результати в цій сфері належать таким відомим американським науковцям, як У. Маккалох, Д. Хебб, Ф. Розенблатт, М. Мінські, Дж. Хопфілд та іншим дослідникам, чиї праці стали фундаментом сучасної теорії штучного інтелекту [13,14].

Штучні нейронні мережі являють собою математично-обчислювальні моделі, побудовані за аналогією з природними нейронними структурами. Їхня функція полягає в опрацюванні інформації та формуванні реакцій на основі навчання, що здійснюється шляхом аналізу взаємозв'язків між входами та виходами системи. Ключовим структурним компонентом будь-якої такої мережі є штучний нейрон – елемент, концепція якого запозичена з функціонування біологічних нервових клітин. Саме сукупність таких елементів, об'єднаних у шари та зв'язки з певними ваговими коефіцієнтами, забезпечує мережі здатність до адаптації, узагальнення інформації та формування складних залежностей, що важко або неможливо описати традиційними алгоритмічними методами.

Штучні нейронні мережі (ШНМ) є одним із ключових напрямів розвитку сучасних інтелектуальних технологій, що знаходять широке застосування у науці, техніці, промисловості, економіці та соціальній сфері [15-17]. Основна перевага нейронних мереж полягає у здатності до навчання, узагальнення, адаптації до змін зовнішніх умов і роботи з неповною або зашумленою інформацією. Завдяки цим властивостям вони ефективно розв'язують задачі, які складно або неможливо формалізувати традиційними аналітичними методами.

Апроксимація функцій.

Встановлення універсальних апроксимуючих властивостей ШНМ стало важливим етапом у формуванні загальної теорії нейронних мереж і стимулювало подальші дослідження в цій галузі. Різні дослідники довели, що нейронна мережа з одним прихованим і одним вихідним шаром здатна з довільною заданою точністю апроксимувати будь-яку неперервну функцію на компактній множині.

Зокрема, Р. Хехт-Нільсен [18,19], використовуючи теорему Колмогорова-Арнольда [20], що описує подання неперервних функцій декількох змінних у вигляді суперпозиції неперервних функцій однієї змінної, довів можливість апроксимації функцій багатьох змінних загального вигляду за допомогою двошарової нейронної мережі з прямими повними зв'язками, фіксованою кількістю нейронів і заданими обмеженими функціями активації.

Дж. Цибенкот і К. Хорник [21-23], спираючись на теорему Хана-Банаха, показали, що кінцева лінійна комбінація фіксованих одновимірних функцій може точно апроксимувати будь-яку неперервну функцію кількох дійсних змінних на заданому гіперкубі за умови достатньо м'яких обмежень на функції однієї змінної.

У [24] розглянуто теоретичні основи апроксимаційних можливостей багатшарових перцептронів (MLP). Автор доводить універсальну здатність таких нейронних мереж наближати довільні неперервні функції за певних умов щодо архітектури та активаційних функцій. Робота пояснює математичні принципи, що лежать в основі ефективності нейронних мереж у задачах нелінійного моделювання.

Оптимізаційні завдання.

Більшість практично важливих завдань можуть бути сформульовані як оптимізаційні, які доставляють екстремум деякому наперед вибраному критерію. Тут, проте, йтиметься тільки про одну таку задачу, задачу про комівояжера, яке має величезне практичне значення і рішення якої вимагає значних обчислювальних витрат. Це завдання полягає в тому, що комівояжер повинен відві-

дати задану кількість місць, вибравши для цього найкоротший маршрут, причому в кожному місці він повинен побувати не більше одного разу. Таке завдання виникає, наприклад, під час створення автоматичних технологічних ліній (свердлення отворів в друкованих платах, трасування друкованих плат і ін.), визначення оптимальних маршрутів перевезення, тощо.

Доведено, що це завдання відноситься до класу «NP-повних» (недетерміністськи поліноміальних), кращим методом рішення яких є повний перебір можливих варіантів. Оскільки у разі n місць існує $n!$ варіантів обходу, очевидно, що із збільшенням кількості місць складність рішення завдання різко зростає. Одне з можливих ефективних рішень цієї задачі за допомогою ШНМ, що вимагає значно менших обчислювальних витрат і при цьому сходиться з рішенням, отриманим повним перебором, було запропоновано в роботі [25].

У роботі [26] розглянуто сучасні методи оптимізації, що застосовуються в навчанні нейронних мереж. Порівнюються стохастичні градієнтні методи, адаптивні алгоритми та їхні модифікації. Аналізуються проблеми збіжності, швидкодії та стабільності під час навчання глибоких моделей.

Автоматичне керування та технічна діагностика

Однією з провідних сфер використання нейронних мереж є автоматичне керування складними динамічними системами. Нейронні регулятори застосовуються у системах керування електроприводами, робототехнічними комплексами, технологічними процесами, транспортними об'єктами. Вони забезпечують адаптацію до змін параметрів об'єкта, компенсацію нелінійностей та невідзначеностей, а також підвищення точності регулювання порівняно з традиційними ПІД-регуляторами.

Велике значення має використання нейронних мереж у задачах технічної діагностики - для розпізнавання несправностей, прогнозування відмов та оцінювання залишкового ресурсу обладнання. У статті [27] досліджено використання методів глибокого навчання для моніторингу технічного стану обладнання. Описано архітектури нейронних мереж для діагностики несправностей на

основі вібраційних сигналів. Розглядаються приклади застосувань у промислових системах і перспективи розвитку галузі.

Оброблення сигналів і зображень

Нейронні мережі широко застосовуються в задачах цифрового оброблення сигналів для фільтрації, прогнозування, класифікації та реконструкції даних у системах зв'язку, акустики, радіолокації.

У галузі комп'ютерного зору глибокі згорткові нейронні мережі (CNN) забезпечують розпізнавання об'єктів, виявлення дефектів, ідентифікацію осіб, аналіз медичних зображень, а також формування систем технічного зору для автономних роботів і транспортних засобів. Стаття [28] містить узагальнений огляд згорткових нейронних мереж (CNN) у задачах розпізнавання зображень. Розглянуто еволюцію архітектур CNN, техніки регуляризації та методи підвищення точності. Підкреслюється значення CNN для сучасного комп'ютерного зору.

Інтелектуальні системи прогнозування

У фінансових, енергетичних та промислових системах нейронні мережі успішно використовуються для прогнозування часових рядів. Вони забезпечують точне передбачення споживання електроенергії, попиту на продукцію, рівня завантаження обладнання, курсових коливань тощо.

Рекурентні нейронні мережі (RNN), а також модифікації типу LSTM (Long Short-Term Memory) і GRU (Gated Recurrent Unit) дозволяють враховувати часову структуру сигналів та забезпечують довготривалі залежності між подіями. У статті [29] подано огляд моделей глибокого навчання, що застосовуються для прогнозування часових рядів. Порівнюються рекурентні мережі, CNN та трансформери. Окрему увагу приділено проблемам надмірного навчання та оцінці якості прогнозів

Кібербезпека та оброблення даних

У сфері інформаційної безпеки нейронні мережі використовуються для виявлення мережових атак, аналізу трафіку, ідентифікації шкідливого програмного забезпечення та виявлення аномальної поведінки користувачів.

Глибокі автоенкодери, графові нейронні мережі (GNN) та моделі з підкріпленням застосовуються для побудови систем виявлення вторгнень у реальному часі й прогнозування потенційних кіберзагроз. У роботах [30-32] автори класифікують підходи за типами архітектур та аналізують їхню ефективність у виявленні кіберзагроз. Окрему увагу приділено проблемам дисбалансу даних, продуктивності моделей у реальному часі та можливостям подальшої інтеграції ШІ в системи кіберзахисту

Робототехніка та автономні системи

У сучасній робототехніці нейронні мережі забезпечують сенсорну обробку інформації, розпізнавання середовища, планування траєкторій руху, навігацію та самонавчання поведінки.

Поєднання глибокого навчання з навчанням із підкріпленням (Reinforcement Learning) дає змогу створювати системи, які здатні самостійно формувати стратегії управління на основі досвіду. Такі технології використовуються у безпілотних транспортних засобах, мобільних роботах і дронах.

У [33] надано огляд присвячено використанню навчання з підкріпленням у робототехніці. Розглянуто підходи до управління рухом, навігації та взаємодії з навколишнім середовищем. Наведено практичні приклади навчання роботів автономної поведінки.

Медицина та біоінформатика

Нейронні мережі успішно застосовуються для автоматичного аналізу медичних зображень (рентген, МРТ, КТ), розпізнавання патологій, прогнозування перебігу захворювань і персоналізованої діагностики.

У біоінформатиці вони використовуються для аналізу геномних даних, моделювання білкових структур, оптимізації лікарських препаратів і пошуку нових біоактивних сполук.

У роботах [34,35] проведено огляд основних алгоритмів глибокого навчання та їхнього застосування у сфері охорони здоров'я. Автори розглядають архітектури CNN, RNN, автоенкодерів та гібридних моделей, аналізуючи їхню ефективність у задачах діагностики, класифікації медичних зображень і прогно-

зування захворювань. Також обговорюються питання якості медичних даних, інтерпретованості результатів і перспективи впровадження ШІ у клінічну практику.

Природномовна обробка інформації (NLP)

Останні роки характеризуються стрімким розвитком трансформерних архітектур, таких як BERT, GPT, T5, які забезпечують якісно новий рівень розуміння природної мови.

Вони застосовуються для автоматичного перекладу, аналізу текстів, генерації осмислених повідомлень, створення чат-ботів, інтелектуальних систем підтримки користувачів і засобів автоматизованого програмування.

У роботі [36] дається огляд сучасних методаів глибокого навчання в обробці природної мови. Пояснено принципи роботи рекурентних мереж, моделей уваги та трансформерів. Показано приклади застосувань у машинному перекладі, аналізі текстів і чат-ботах.

Енергетика та транспорт

У системах енергопостачання нейронні мережі використовують для прогнозування навантажень, керування розподілом потужностей, оптимізації роботи відновлюваних джерел енергії та підвищення ефективності енергоспоживання.

У транспортних системах нейромережеві моделі допомагають у прогнозуванні трафіку, розпізнаванні дорожніх ситуацій, запобіганні аваріям і реалізації автономного керування.

У роботах [37-39] автори класифікують основні типи моделей і аналізують їхню ефективність у коротко- та довгостроковому прогнозуванні. Розглянуто фактори, що впливають на точність моделей, включно з якістю даних, сезонністю та зовнішніми умовами. Також визначено ключові напрями подальших досліджень у галузі енергетичного моделювання на основі ШІ.

Промисловість і виробничі процеси

У промисловості нейронні мережі застосовуються для оптимізації технологічних процесів, регулювання параметрів режимів роботи, контролю якості продукції та прогнозування технічного стану машин.

Поєднання нейронних мереж із нечіткою логікою, генетичними алгоритмами та системами нечітких регуляторів дозволяє створювати гібридні інтелектуальні системи керування нового покоління.

У роботах [40-42] здійснено ґрунтовний огляд сучасних методів глибокого навчання, що застосовуються для інтелектуальної діагностики та прогнозування стану промислових систем. Автори аналізують архітектури нейронних мереж та їх ефективність у виявленні й класифікації технічних несправностей. Розглянуто переваги використання ШІ над традиційними методами аналізу сигналів, а також визначено основні виклики й напрями подальших досліджень у галузі промислової діагностики.

Творчі індустрії та медіа

Завдяки генеративним нейронним мережам (GAN) і моделям дифузії (Diffusion Models) активно розвивається напрям генеративного мистецтва - створення зображень, музики, відео та текстів.

Такі системи знаходять застосування у дизайні, кінематографії, моделюванні, розробленні ігор та медіаконтенту. Водночас порушуються питання авторського права, етики й достовірності створюваних матеріалів.

У статті [43] подано систематичний огляд генеративних моделей штучного інтелекту, що застосовуються в мистецтві, дизайні та медіа. Розглянуто архітектури GAN і дифузійні моделі. Автори аналізують етичні аспекти й перспективи використання ШІ у творчій діяльності.

Отже, штучні нейронні мережі посідають провідне місце серед сучасних методів оброблення інформації та керування. Їхнє використання охоплює широкий спектр галузей - від технічних систем і енергетики до медицини, економіки та творчих індустрій.

Подальший розвиток нейромережевих технологій спрямований на підвищення інтерпретованості моделей, зниження енергоспоживання, розроблення універсальних архітектур і забезпечення кібербезпеки, що визначатиме новий етап еволюції інтелектуальних систем.

У додатках А–Г подано матеріали, що пояснюють фундаментальні концепції теорії штучних нейронних мереж, які слугують теоретичною основою для виконаних у роботі досліджень та розробок. Ці відомості мають допоміжний характер і покликані забезпечити глибше розуміння принципів функціонування нейромережевих систем, застосованих під час синтезу алгоритмів управління.

У межах даної роботи для формування нейромережевої системи управління використано багат шарові прямонаправлені архітектури, що передбачають односторонню передачу інформації між нейронами, тобто відсутність внутрішніх зворотних зв'язків. Такий тип мереж відомий як багат шаровий перцептрон, який є однією з найпоширеніших структур у задачах апроксимації та класифікації. У додатку А подано докладний розгляд будови окремого штучного нейрона, тоді як у додатку Б висвітлено структуру та принципи функціонування багат шарових нейронних мереж.

Методи навчання, їх характеристика, відмінності та обґрунтування вибору конкретного алгоритму, застосованого у межах цього дослідження, наведені у додатку В. Саме коректний вибір методу навчання значною мірою визначає здатність нейромережевої моделі відтворювати необхідні властивості об'єкта керування.

Одним із ключових етапів під час вирішення задач управління, прогнозування стану системи та моделювання її поведінки є побудова математичної моделі, яка адекватно відображає властивості реального об'єкта. Питання ідентифікації динамічних систем за допомогою нейронних мереж, а також особливості формування моделей нелінійних об'єктів наведено в додатку Г. Цей матеріал дозволяє краще зрозуміти процес формування нейромережевої моделі, що використовується під час синтезу системи управління.

1.2 Огляд структур нейромережевих систем регулювання та обґрунтування вибору архітектури

На сьогодні у науковій літературі та практиці управління накопичено значну кількість різноманітних структур систем нейроуправління, кожна з яких має свої переваги, особливості побудови й сферу застосування. У частині з них нейронні мережі слугують доповненням до традиційних методів регулювання: використовується певний аналітичний закон керування, а нейронна модель застосовується для відновлення невідомих або змінних параметрів об'єкта. Такий підхід прийнято відносити до класу структурованих методів керування, оскільки рахунок ведеться на математичну модель, а не на повну свободу апроксимаційних властивостей мережі.

На противагу цьому, неструктуровані методи дають нейромережі можливість працювати у ролі універсального наближення без прив'язки до строго визначеної структури рівнянь об'єкта. Завдяки цьому вони дозволяють максимально використати її здатність до апроксимації складних нелінійних залежностей. У цьому підрозділі наведено огляд структур нейромережевих систем керування, які мають найбільшу практичну значущість.

1.2.1 Послідовне (інверсно-пряме) нейрокерування

Одна з найдавніших і концептуально простих структур нейроуправління – це схема, у якій нейронна мережа виконує роль інверсної моделі об'єкта, тобто відтворює відображення, зворотне до закону функціонування керованої системи. Саме тому таку архітектуру інколи позначають терміном «*інверсно-пряме управління*». Її ключовою ідеєю є забезпечення такого керуючого впливу, щоб об'єкт реагував відповідно до бажаної моделі.

Під час навчання мережі мінімізується певний цільовий функціонал помилки. При цьому необхідно знати або обчислювати Якобіан об'єкта, оскільки він визначає чутливість виходу системи до зміни керуючого сигналу. Процес навчання припиняється у разі, коли досягається достатньо мала похибка – це

свідчить про те, що нейронна мережа навчилася відтворювати інверсні властивості об'єкта. Такий підхід називають спеціалізованим нейроуправлінням із зворотним відображенням (рис. 1.1).

Одновимірний об'єкт та двошаровий перцептрон як регулятор. Для більш наочного пояснення структури розглянемо одно-вхідний та одно-вихідний об'єкт керування, який описується деякою невідомою функцією перетворення. Як нейронний регулятор прийнемо двошаровий перцептрон із одним нейроном на виході та логістичною активацією у прихованому шарі:

$$f_{log}(u) = \frac{1}{1 + e^{-u}}, \quad (1.1)$$

похідна якої задається виразом

$$f'_{log}(u) = f_{log}(1 - f_{log}(u)). \quad (1.2)$$

Схему такої мережі наведено на рис. 1.2.

Метою регулювання виступає мінімізація квадратичної функції помилки

$$E = \frac{1}{2} e^2 = \frac{1}{2} (d(k) - y(k))^2 \quad (1.3)$$

де $y(k) = f_0(u(k))$, f_0 невідома функція перетворення.

Вихід мережі визначається рівняннями:

$$u = y_1^2 = f_{log}(a_1^2), \quad (1.4)$$

$$a_1^2 = \sum_{j=1}^{N^1} w_{1j}^2 y_j^1 + b_{10}^2 n_0^1, \quad (1.5)$$

$$y_j^1 = f_{log}(a_j^1), \quad (1.6)$$

$$a_j^1 = w_{j1}^1 d + b_{j0}^1 n_0^0. \quad (1.7)$$

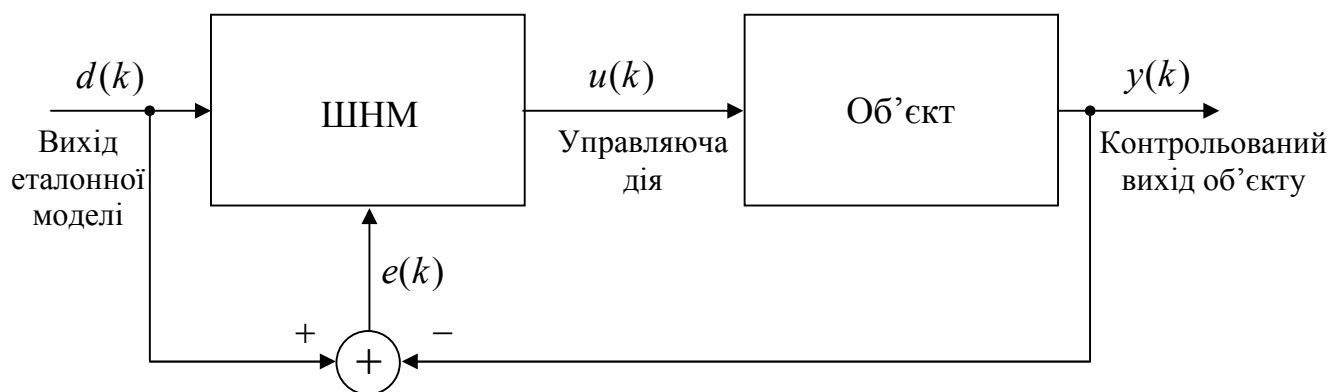


Рисунок 1.1 – Приклад реалізації спеціалізованого підходу до нейронного керування

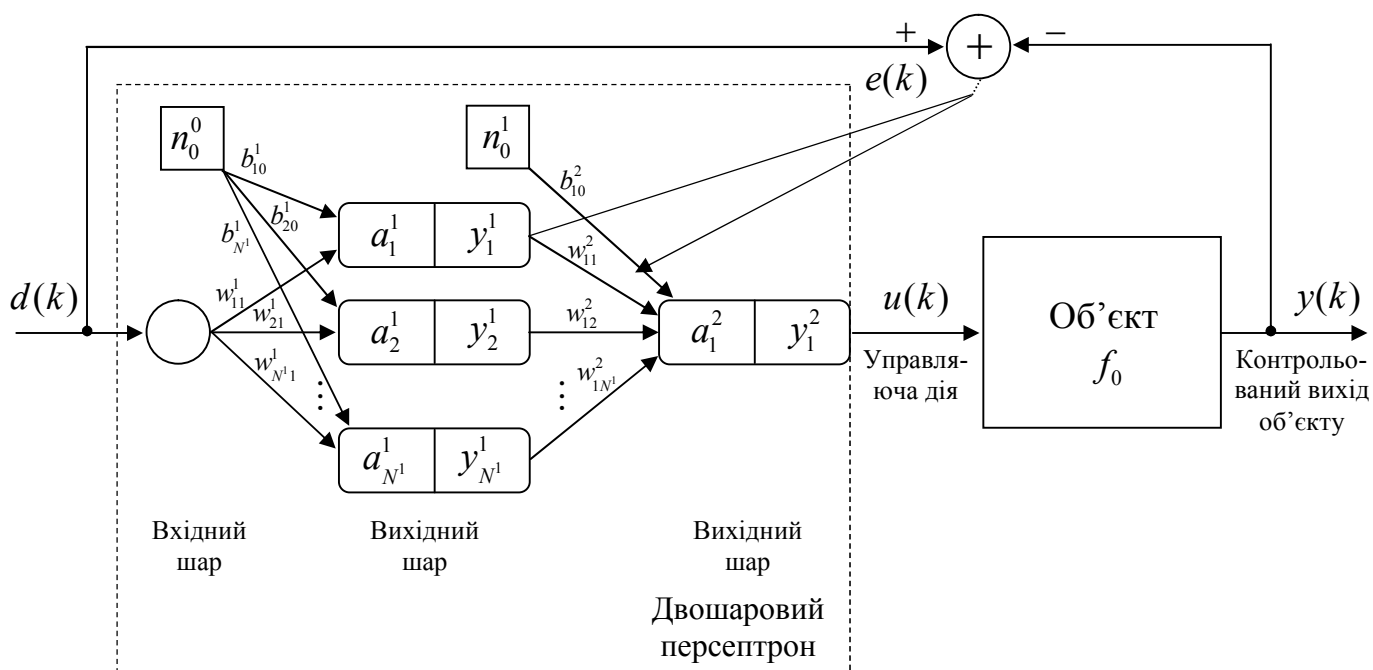


Рисунок 1.2 – Схематична будова нейрорегулятора, побудованого на двошаровому персептроні

Алгоритм навчання регулятора. Для зміни вагових коефіцієнтів використовується метод найшвидшого спуску. Застосувавши правило диференціювання складних функцій, отримаємо формулу для корекції вагів вихідного шару.

$$\Delta w_{1j}^2 = -\eta \frac{\partial E}{\partial w_{1j}^2} = -\eta \frac{\partial E}{\partial a_1^2} \frac{\partial a_1^2}{\partial w_{1j}^2}.$$

З (1.5) маємо

$$\frac{\partial a_1^2}{\partial w_{1j}^2} = y_j^1.$$

Враховуючи (1.3), (1.1), (1.2) і, вповзначивши локальну помилку вихідного шару $\delta^2 = \frac{\partial E}{\partial a_1^2}$, отримаємо

$$\delta^2 = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial a_1^2} = e \frac{\partial y_0}{\partial u} \frac{\partial u}{\partial a_1^2} = ef'_0(u)u(1-u). \quad (1.8)$$

Загальна формула налаштування вагів вихідного шару

$$\Delta w_{1j}^2 = -\eta \delta^2 y_j^1. \quad (1.9)$$

Для прихованого шару локальна помилка визначається:

$$\Delta w_{j1}^1 = -\eta \delta_j^1 d, \quad (1.10)$$

де

$$\delta_j^1 = \frac{\partial E}{\partial a_1^2} \frac{\partial a_1^2}{\partial y_j^1} \frac{\partial y_j^1}{\partial a_j^1} = \delta^2 w_{1j}^2 y_j^1 (1 - y_j^1). \quad (1.11)$$

Загальний алгоритм функціонування. Алгоритм навчання нейромережевого регулятора можна представити у вигляді такої послідовності:

1. Задати початкові ваги w і зсуви b , та значення коефіцієнта навчання η .
2. Обчислити помилку керування та локальні помилки шарів.
3. Внести корекції у ваги вихідного шару за (1.9).
4. Оновити ваги прихованого шару за (1.10).

Повторювати кроки 2–4 до досягнення критерія зупинки.

Особливості застосування. Описана структура ефективна передусім для статичного керування. Якщо ж об'єкт має динамічні властивості, регулятор повинен отримувати не лише поточні значення сигналів, а й їх попередні реалізації. У цьому випадку доцільно використовувати багат шаровий персептрон із розширеним вектором входів.

Основна трудність такого підходу полягає у необхідності визначення Якобіана об'єкта $f'_0(u(k))$. Оскільки реальна залежність, як правило, невідома, використовують її апроксимацію, наприклад:

$$f'_0(u(k)) \approx \frac{f_0(u(k))}{\Delta u(k)},$$

де

$$\Delta f_0(u(k)) = f_0(u(k-1)) - f_0(u(k-2)),$$

$$\Delta u(k) = u(k-1) - u(k-2),$$

що дає змогу оцінити чутливість виходу до зміни керуючого сигналу.

1.2.2 Система нейрокерування із зворотним розповсюдженням помилки в часі

Ще один підхід до побудови систем нейроуправління ґрунтується на використанні нейромережевої моделі об'єкта – так званого нейроемулятора, що виконує функції ідентифікатора динамічної системи. Робота системи в цьому випадку поділяється на два окремі етапи. Спершу формується інверсна модель об'єкта: нейронна мережа-регулятор навчається відтворювати таку динаміку, яка компенсує поведінку реального об'єкта. Після завершення цього етапу сформований регулятор використовується безпосередньо для управління та для передачі похибки назад через нейромережеву модель.

Використання емулятора дозволяє отримувати оціночні значення Якобіана об'єкта, замінюючи сам реальний об'єкт його нейромережею. Це дає можливість обчислити:

$$f'_0(u(k)) \approx \frac{\partial \hat{f}_c(u(k))}{\partial u(k)},$$

де $\hat{f}_c$ – відображення, сформоване нейронною моделлю об'єкта.

Добитися Підвищити швидкодію навчання нейромережевого ідентифікатора можна, якщо замість “чистого” виходу об'єкта застосовувати сигнал, утворений шляхом сумування виходу деякої апріорної математичної моделі. У разі розбіжностей між прогнозованим і фактичним значенням вихідного сигналу відбувається корекція параметрів мережі, спрямована на мінімізацію похибки ідентифікації. Таким чином, нейроемулятор розглядається як комбінація певної моделі та мережі, де налаштуванню підлягають лише вагові коефіцієнти останньої (рис. 1.3). Такий підхід добре зарекомендував себе в практичних застосуваннях систем автоматичного керування.

Найбільш ефективними в таких схемах є моделі, орієнтовані на тривале прогнозування поведінки об'єкта: AR, ARX, ARMAX, OE, BJ та інші типи параметричних моделей. Їхнім завданням є опис лінійної частини динаміки, тоді як нейромережа компенсує нелінійні зв'язки, що виникають у процесі роботи системи.

Узагальнена структура нейромережевої системи керування нелінійним динамічним об'єктом подана на рис. 1.4. Нейронна мережа ШНМ2 виступає в ролі ідентифікатора (рис. 1.5), який формує оцінку Якобіана, необхідну для навчання основного нейрорегулятора – ШНМ1. Останній реалізує інверсну динаміку об'єкта керування.

Процес навчання регулятора здійснюється за принципом зворотного розповсюдження помилки, який проходить через нейромережеву модель об'єкта. Через це розглянута структура отримала назву «метод зворотного розповсюдження в часі». У літературі він також описується у роботах [26–48].

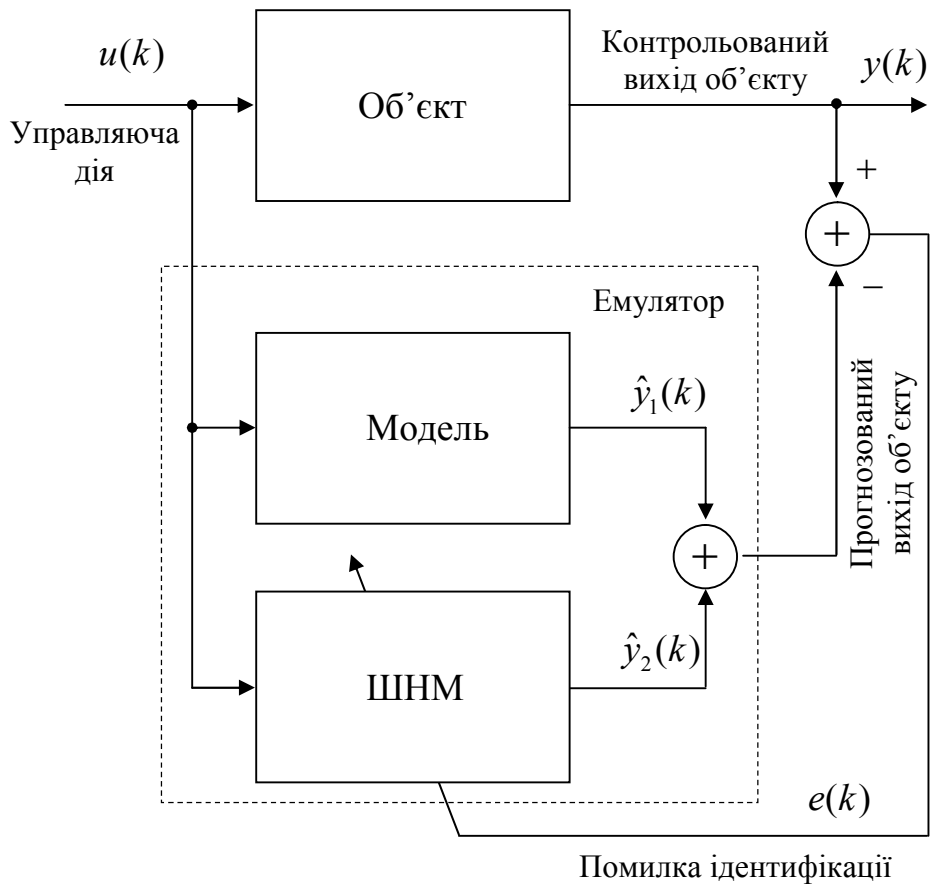


Рисунок 1.3 – Конфігурація нейроемулятора з використанням умовної моделі

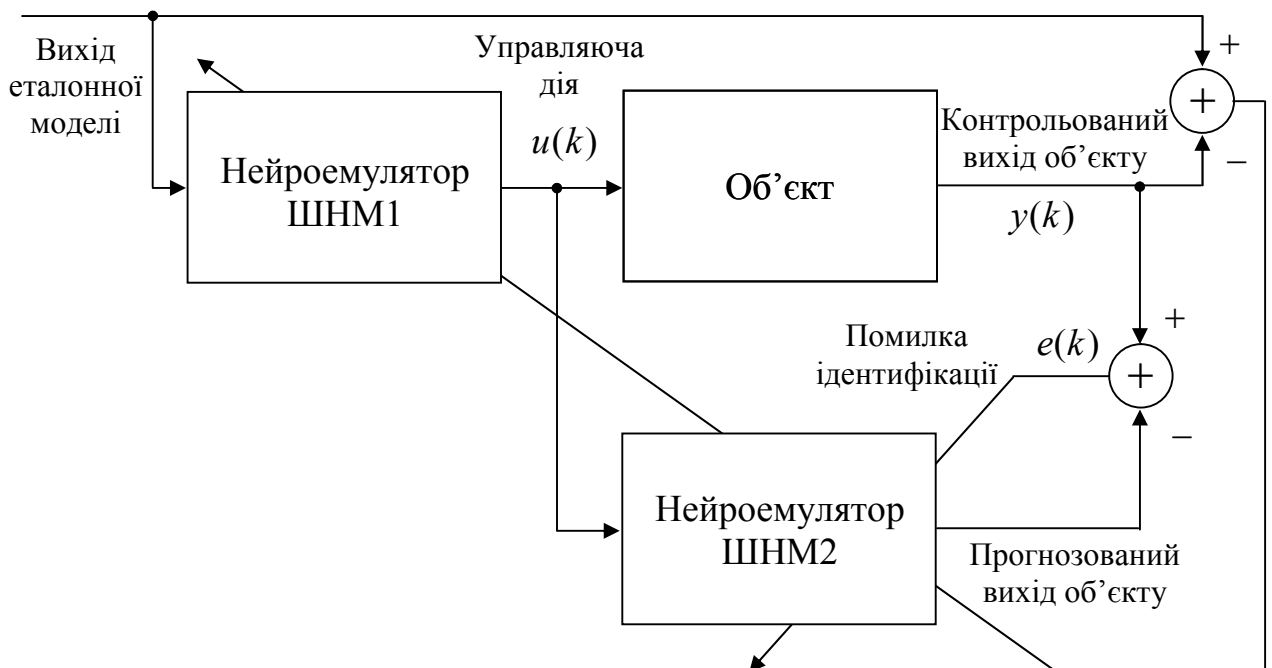


Рисунок 1.4 – Узагальнена схема нейроуправління на основі інверсної динаміки

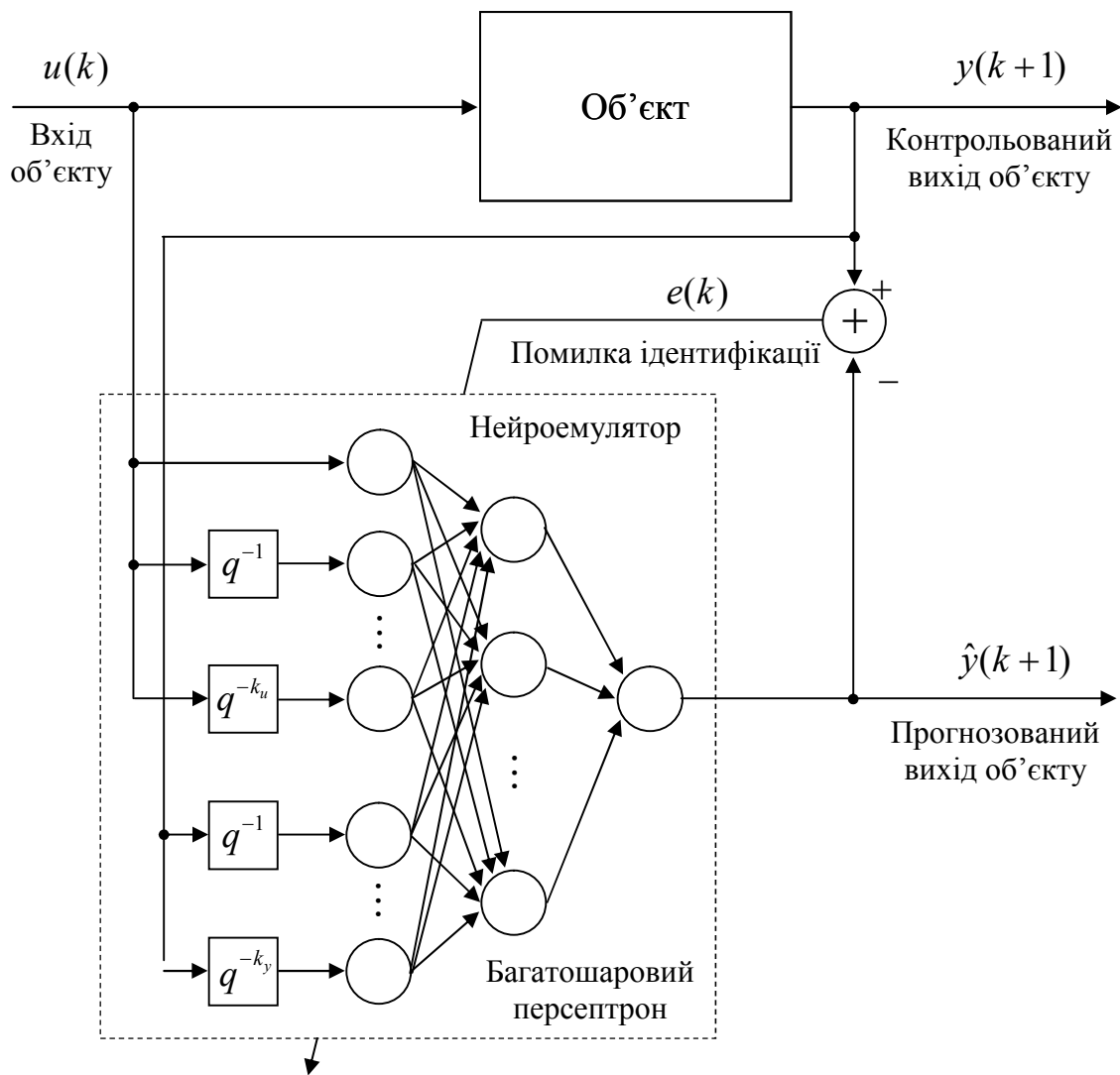


Рисунок 1.5 – Структура нейроемулятора як ідентифікатора об'єкта

У праці [49] запропоновано модифікований варіант схеми інверсного керування (рис. 1.6). У цій конфігурації нейронна модель об'єкта ШНМ2 попередньо навчається відтворювати інверсну динаміку об'єкта. Після завершення процесу навчання її параметри копіюються до нейрорегулятора ШНМ1. Процедура такого копіювання виконується після повної ідентифікації системи. Навчання нейроемулятора може здійснюватися як у автономному режимі, так і шляхом подачі на вхід об'єкта випадкових або тестових сигналів. Для побудови моделі ШНМ2 може використовуватися структура, наведена на рис. 1.7.

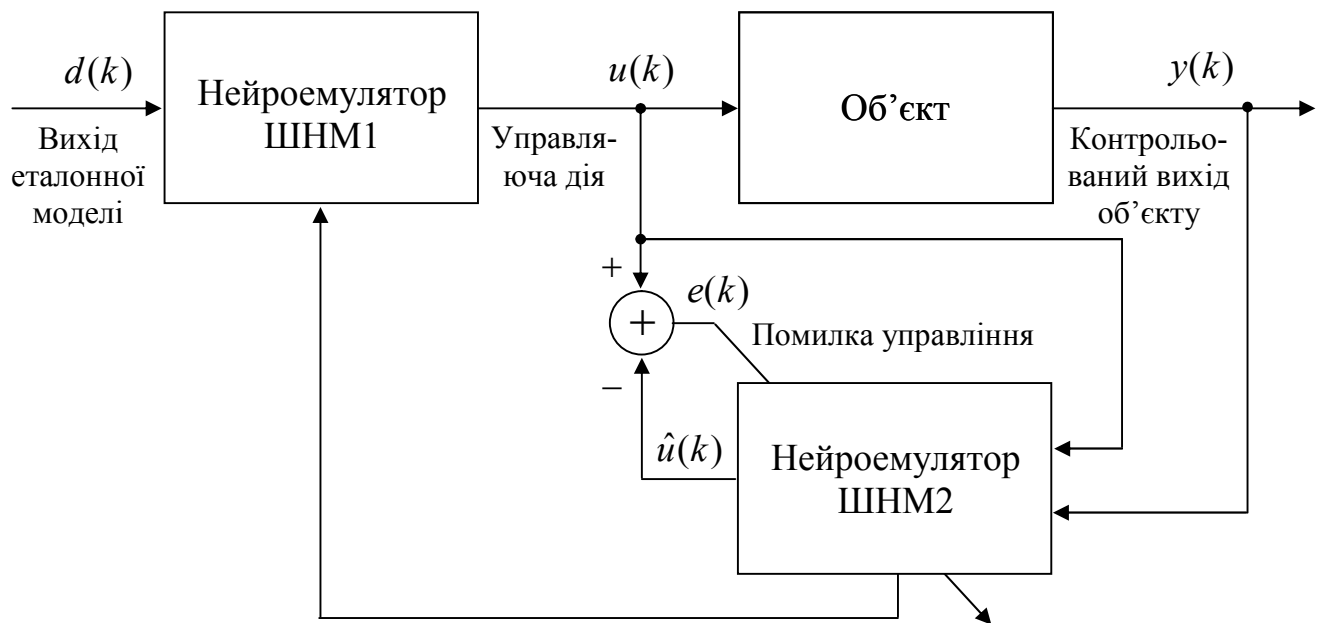


Рисунок 1.6 – Схема зворотного відображення для реалізації інверсної динаміки об'єкта

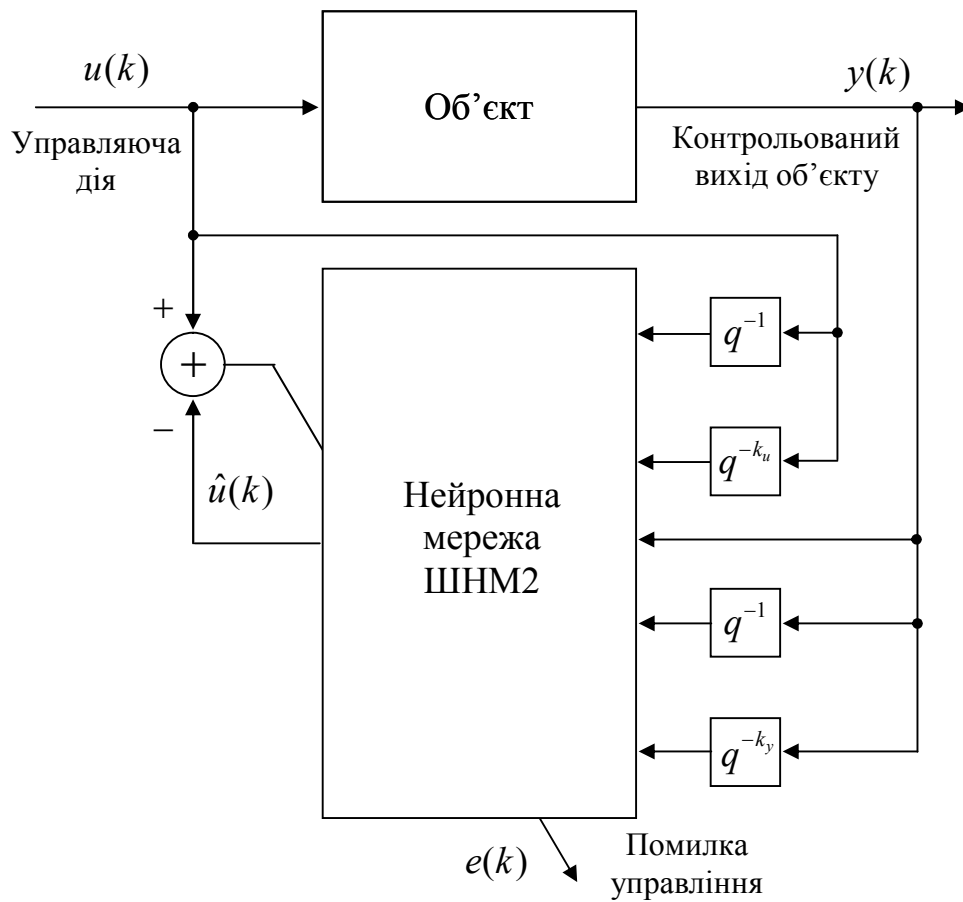


Рисунок 1.7 – Структура нейроемулятора для схеми зворотного відображення

1.2.3 Паралельне нейроуправління

Широке застосування отримали схеми паралельного нейроуправління. У цьому випадку сигнал керування формується як сумарний вихід звичайного регулятора (наприклад, ПІД-регулятора) та нейрорегулятора (рис. 1.8).

Роль нейрорегулятора полягає у корекції сигналу стандартного регулятора у тих випадках, коли його вихід не забезпечує потрібної якості управління. Один із варіантів паралельної схеми, що поєднує інверсну та пряму моделі об'єкта, запропонований у роботах [50, 51] (рис. 1.9).

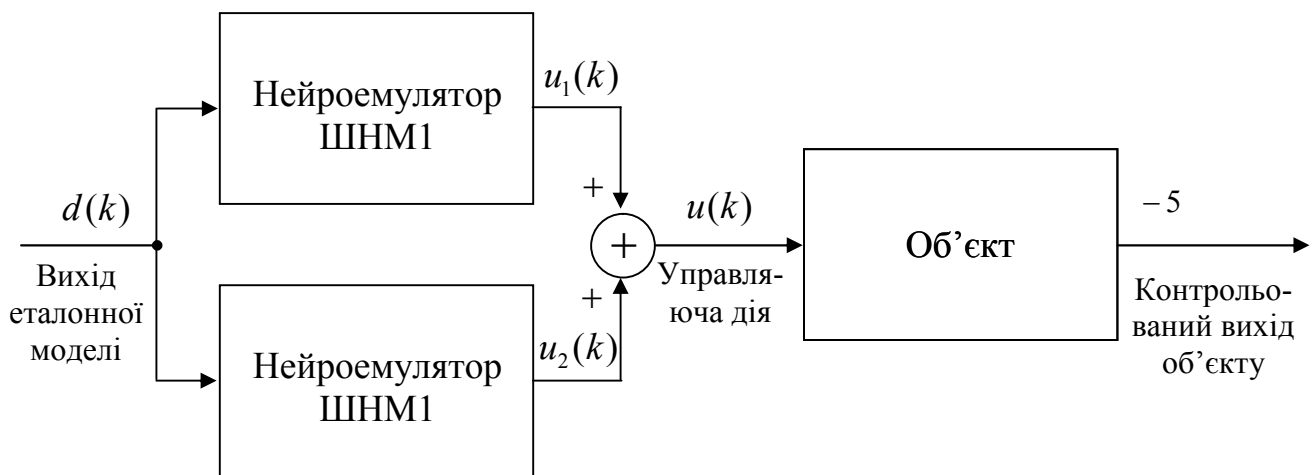


Рисунок 1.8 – Паралельна схема керування з нейрорегулятором

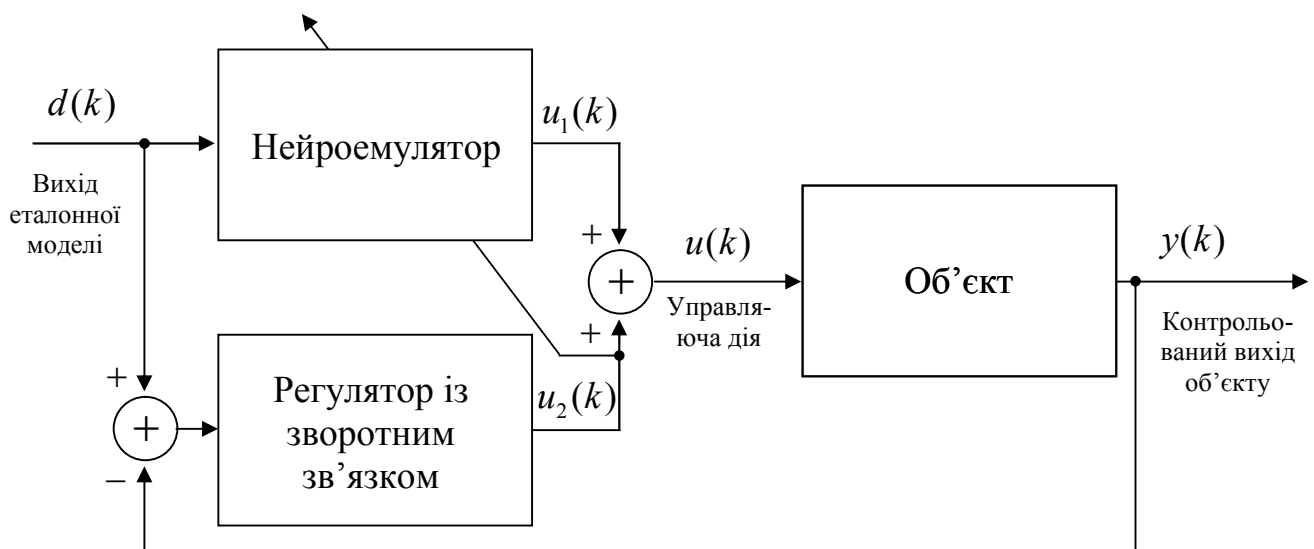


Рисунок 1.9 – Нейроуправління на основі помилки зворотного зв'язку

Для налаштування параметрів нейрорегулятора використовується алгоритм навчання на основі помилки зворотного зв'язку. Мережа навчається через багаторазове повторення циклів бажаної траєкторії до досягнення необхідної точності управління. При цьому сигнал з виходу паралельного регулятора використовується як помилка зворотного зв'язку. Після завершення навчання нейрорегулятор повністю бере на себе керування об'єктом, виключаючи традиційний регулятор.

Для загального розгляду можна використати модифікований варіант схеми нейроуправління по помилці зворотного зв'язку, представлений на рис. 1.10 [52].

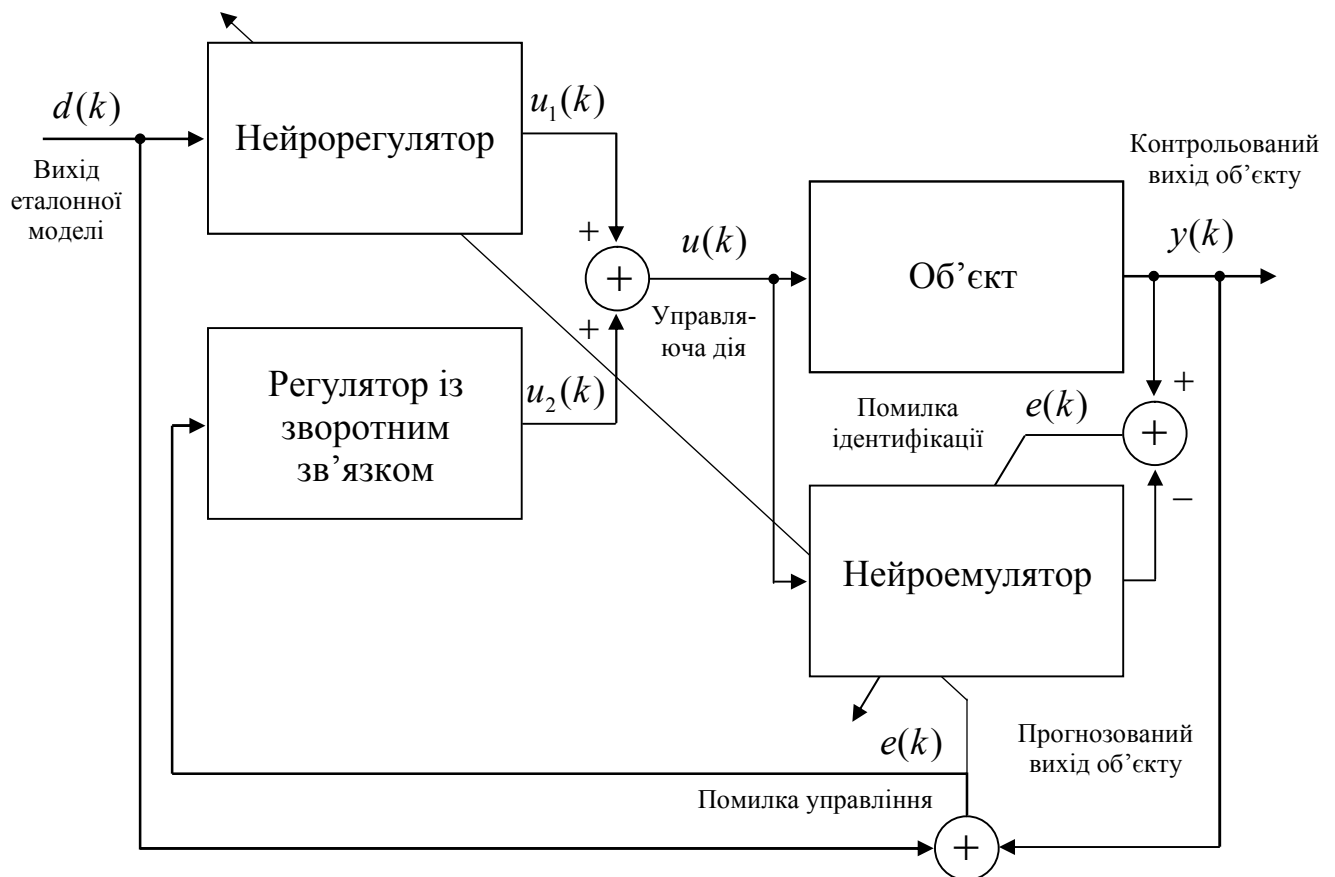


Рисунок 1.10 – Узагальнена схема нейроуправління за помилкою зворотного зв'язку

Система управління у цьому випадку складається з трьох частин. Першу частину (виділено жирними лініями) складає традиційний контур зворотного зв'язку (зовнішній контур) [53]. Він базується на помилці між виходом ета-

лонної моделі та виходом об'єкта управління $e(k) = d(k) - y(k)$. Як регулятор зворотного зв'язку зазвичай застосовується ПД-регулятор.

Другу частину формує нейроемулятор, що моделює динаміку об'єкта та навчання якого здійснюється за помилкою ідентифікації $e(k) = y(k) - \hat{y}(k)$. Нейроемулятор утворює внутрішній контур зворотного зв'язку, що забезпечує швидку реакцію системи.

Третя частина включає нейрорегулятор, який реалізує інверсну динаміку об'єкта. Датчики зворотного зв'язку використовуються лише на етапі навчання, оскільки через затримки сигналів накладаються обмеження на максимальний коефіцієнт передачі традиційного регулятора. Під час навчання нейрорегулятор підлаштовується під динамічну модель об'єкта, а після завершення навчання бере на себе основне управління, забезпечуючи високу швидкість керування навіть при наявності зовнішніх завад.

Для виведення алгоритму навчання розглянемо об'єкт управління з часовою затримкою, що описується рівнянням

$$\begin{aligned} y(k+1) &= f_0(u(k)) = f_0(u_1(k)) - u_2(k); \\ u_2(k) &= \Phi(k). \end{aligned} \tag{1.12}$$

Критерієм якості служить функція помилки (1.3), а для налаштування вагів нейрорегулятора застосовується метод найшвидшого спуску. Структура нейрорегулятора показана на рис. 1.11.

Корекція вагів нейронів вихідного шару здійснюється за правилом:

$$\Delta w_{1j}^2(k+1) = -\eta \frac{\partial E}{\partial w_{1j}^2(k)}.$$

Корекція вагів нейронів прихованого шару визначається формулою:

$$\Delta w_{j1}^1(k+1) = -\eta \frac{\partial E}{\partial w_{j1}^1(k)}.$$

Використовуючи позначення для локальної помилки вихідного шару

$\delta^2 = \frac{\partial E}{\partial a_1^2}$ та співвідношення вхід-вихід об'єкта управління (1.12), отримуємо:

$$\begin{aligned}\delta^2 &= \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial a_1^2} = \\ &= e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \left(\frac{\partial u_1(k)}{\partial a_1^2} + \frac{\partial u_2(k)}{\partial a_1^2} \right).\end{aligned}$$

З урахуванням виду активаційної функції і те, що $\frac{\partial u_2(k)}{\partial a_1^2} = 0$, отримаємо

$$\delta^2 = (d(k+1) - y(k+1)) \frac{\partial y(k+1)}{\partial u(k)} u_2(k)(1 - u_2(k)). \quad (1.13)$$

Для нейронів прихованого шару, використовуючи формули (1.5), (1.6) та локальні помилки (1.13), правило корекції вагів має вигляд:

$$\Delta w_{j1}^1 = -\eta \delta_j^1 d,$$

де

$$\delta_j^1 = \delta^2 w_{1j}^2 y_j^1 (1 - y_j^1). \quad (1.14)$$

Алгоритм налаштування вагів нейрорегулятора для паралельної схеми:

1. Встановлюються початкові значення вагів w і зсувів b , задається норма навчання η .
2. На вхід мережі подається опорний сигнал $d(k)$ і обчислюються локальні помилки вихідного δ^2 та прихованого δ_j^1 шарів за формулами (1.13) та (1.14).
3. Налаштовуються ваги вихідного шару $\Delta w_{1j}^2(k+1) = \Delta w_{1j}^2(k) - \eta \delta^2 y_j^1$.
4. Налаштовуються ваги прихованого шару $\Delta w_{j1}^1(k+1) = \Delta w_{j1}^1(k) - \eta^2 \delta_j^1 d$.
5. Повторюються дії пп. 2-4 до досягнення критеріїв зупинки навчання.

Як і у випадку послідовної схеми, на кожній ітерації навчання необхідне обчислення Якобіана об'єкта управління. Його можна апроксимувати або оцінити за допомогою нейроемулятора на етапі ідентифікації (рис. 1.11).

1.2.4 Пряме адаптивне нейроуправління

Розглядається нелінійний об'єкт, який описується моделлю NARX:

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)] \quad (1.15)$$

де $y(k)$ – вихід об'єкта;

$u(k)$ – управляючий сигнал;

$f(\cdot)$ – невідома нелінійна функція;

k_u, k_y – порядки запізнювання по вихідному каналу та каналу керування.

Для побудови нейромережевої моделі об'єкта використовується двошаровий персептрон. Вагові коефіцієнти мережі коригуються за допомогою алгоритму зворотного розповсюдження помилки. У прихованому шарі застосовується функція активації гіперболічний тангенс:

$$f_{th}(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}},$$

Похідна цієї функції має вид

$$f'_{th}(u) = 1 - f_{th}^2(u).$$

Для нейрона вихідного шару обрана лінійна активація.

Вхідні сигнали нейроемулятора формуються у вигляді вектора:

$$p = [y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)].$$

Нейромережева модель об'єкту (НММ) має вид:

$$\hat{y}(k+1) = \hat{f}[y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)], \quad (1.16)$$

де $\hat{y}(k+1)$ – вихідний сигнал штучної нейронної мережі;

$\hat{f}[\cdot]$ – оцінкою невідомої функції $f[\cdot]$.

Оскільки алгоритм навчання мережі мінімізує квадрат помилки ідентифікації $e^2(k+1) = [y(k+1) - \hat{y}(k+1)]^2$, вихід НММ $\hat{y}(k+1)$ може бути використаний як прогнозований вихід об'єкта управління. Тому метою управління є мінімізація розходження між виходами нейромережевої та еталонної моделей. Функціонал якості управління має вигляд:

$$J = \frac{1}{2} e^2(k+1), \quad (1.17)$$

де

$$e(k+1) = d(k+1) - \hat{y}(k+1). \quad (1.18)$$

Керуюча дія $u(k)$ обирається таким чином, щоб мінімізувати функціонал J .

Використовуючи нейромережеву модель, рівняння (1.16) можна записати у вигляді:

$$\hat{y}(k+1) = w^{2^T} [\tanh(\mathbf{W}^1 p + b^1)] + b^2, \quad (1.19)$$

де $\mathbf{W}^1$ – матриця ваг прихованого шару;

w^2 – вектор ваг вихідного шару;

b^1 – вектор зсувів прихованого шару;

b^2 – зсув вихідного шару.

Мінімум функціоналу J можна знайти рекурсивно за методом найшвидшого спуску:

$$u(k+1) = u(k) + \eta \frac{\partial J}{\partial u(k)}, \quad (1.20)$$

де $\eta > 0$ – норма навчання.

Точність роботи регулятора безпосередньо залежить від адекватності НММ. Для забезпечення збіжності виходу мережі $\hat{y}(k+1)$ до фактичного виходу об'єкта, мережу можна навчати у режимі on-line.

Диференціюючи функціонал J по сигналу управління $u(k)$, отримуємо:

$$\frac{\partial J}{\partial u(k)} = -e(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)}, \quad (1.21)$$

де $\frac{\partial \hat{y}(k+1)}{\partial u(k)}$ – чутливість НММ по вхідному сигналу $u(k)$.

Підставляючи (1.21) в (1.20), отримаємо

$$u(k+1) = u(k) + \eta e(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)}.$$

З урахуванням архітектури нейронної мережі (1.19), градієнт обчислюється таким чином:

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = w^{2T} [\text{sech}(\mathbf{W}^1 p + b^1)]^{-1} \mathbf{W}^1 \frac{\partial p}{\partial u(k)}$$

де $\frac{\partial p}{\partial u(k)} = [0, 0, \dots, 0, 1, 0, \dots, 0]^T$.

Оновлення керуючої дії відбувається за виразом:

$$u(k+1) = u(k) + \eta e(k+1) w^{2T} [\text{sech}^2(\mathbf{W}^1 p + b^1)]^{-1} \mathbf{W}^1 \frac{\partial p}{\partial u(k)}. \quad (1.22)$$

Таким чином, алгоритм прямого адаптивного нейроуправління складається з наступних кроків:

1. Обчислення виходу нейромережевої моделі $\hat{y}(k+1)$ з використанням (1.19).
2. Обчислення помилки керування $e(k+1)$ з використанням (1.18).

3. Налаштування ваг мережі за алгоритмом зворотного розповсюдження помилки.
4. Обчислення управляючої дії $u(k+1)$ за правилом градієнтного спуску з використанням (1.22).
5. Подача сигналу $u(k+1)$ на об'єкт управління.
6. Повторення дій 1–5 на кожному кроці моделювання.

Покращення властивостей алгоритму досягається шляхом прогнозування майбутніх значень вхідних і вихідних сигналів об'єкта та еталонної моделі. Оскільки НММ (1.28) асимптотично еквівалентна об'єкту, її виходи можуть служити для прогнозу майбутніх сигналів та забезпечення більш ефективного управління.

1.2.5 Адаптивне нейроуправління з самонастроюванням

Одним із способів застосування нейронних мереж у задачах керування є використання схем нейроуправління з можливістю самонастроювання. У таких схемах штучна нейронна мережа (ШНМ) виконує роль оператора системи, автоматично підбираючи параметри традиційного регулятора. Більшість адаптивних алгоритмів передбачає наявність параметрів, що встановлюються заздалегідь, часто методом проб і помилок. Використання ШНМ дозволяє автоматизувати процес вибору та налаштування цих параметрів, підвищуючи ефективність системи. Тому стратегія нейроуправління з самонастройкою може застосовуватися в багатьох традиційних технологічних процесах.

Розглянемо приклад схеми самонастроюваного нейроуправління для ПІД-регулятора, що представлена на рис.1.11.

У цій схемі ШНМ відповідає за підбір параметрів ПІД-регулятора, щоб мінімізувати помилку регулювання. Дискретна форма закону ПІД-регулятора описується рівнянням:

$$\begin{aligned}
 u(k) = & u(k-1) + K_p(e(k) - e(k-1)) + K_I e(k-1) + \\
 & + K_D(e(k) - 2e(k-1) + eu(k-2)),
 \end{aligned}
 \tag{1.23}$$

де K_p , K_I , K_D – пропорційний, інтегральний та диференціальний коефіцієнти відповідно;

$e(k) = d(k) - y(k)$ – помилка між виходом об'єкта та еталонною траєкторією;

$d(k)$ – необхідний вихід об'єкту управління (уставка).

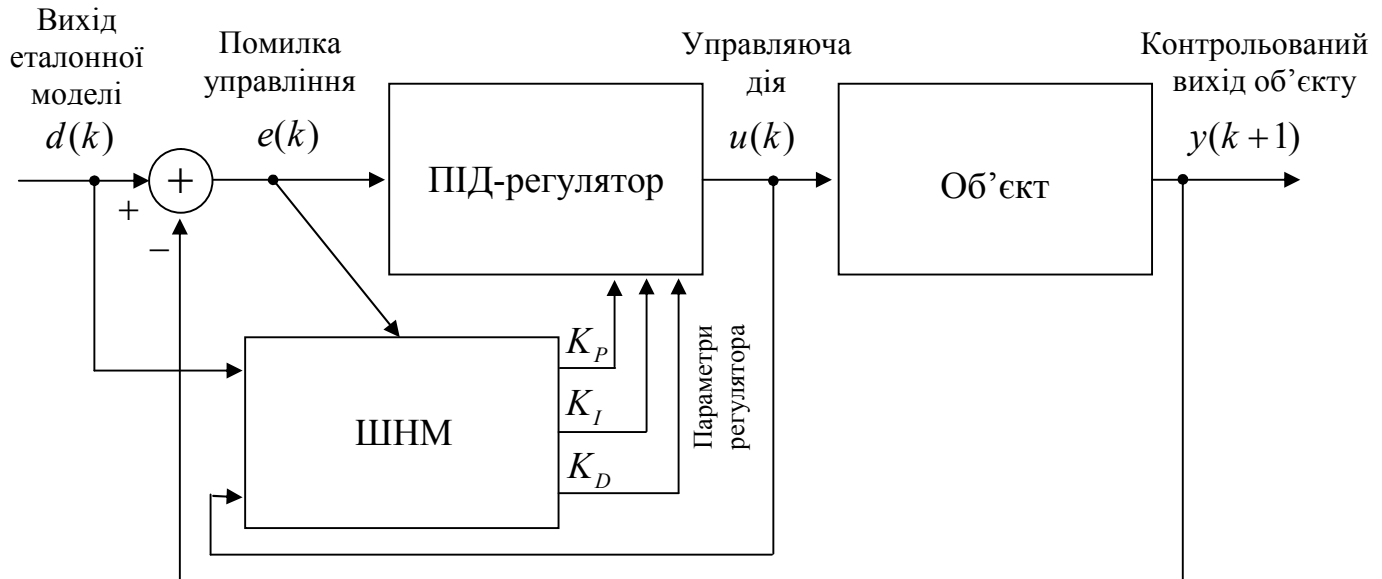


Рисунок 1.11 – Структура ПІД-регулятора з автоматичним налаштуванням через ШНМ

Розглянемо об'єкт керування, поведінка якого описується рівнянням $y(k+1) = f_0(u(k))$. Для оцінки якості регулювання приймаємо квадратичний критерій функції помилки:

$$E = \frac{1}{2} e^2(k+1). \quad (1.24)$$

Для навчання нейромережевої моделі ПІД-регулятора використовується двошаровий персептрон із трьома нейронами на виході: перший вихід формує K_p , другий – K_I , третій – K_D , тобто

$$K_p = y_1^2, \quad K_I = y_2^2, \quad K_D = y_3^2. \quad (1.25)$$

Прихований шар застосовує логістичну функцію активації, а вихідний – лінійну для забезпечення потрібного діапазону коефіцієнтів. Для корекції ваг мережі застосовується алгоритм градієнтного спуску. При використанні квадратичного критерію помилки градієнт для вихідного шару визначається як:

$$\frac{\partial E}{\partial w_{ij}^2} = \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_i^2(k)} \frac{\partial y_i^2(k)}{\partial a_i^2(k)} \frac{\partial a_i^2(k)}{\partial w_{ij}^2(k)}.$$

Оскільки $e(k+1) = d(k+1) - y(k+1)$ і з урахуванням (1.24) можемо записати

$$\frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} = -(d(k+1) - y(k+1)) = -e(k+1).$$

Правило функціонування нейронів вихідного шару має вид:

$$y_i^2 = a_i^2 = \sum_{l=1}^{N^{L-1}} w_{il}^2 y_l^1 + b_{i0}^2, \quad i = 1, 2, 3;$$

$$\frac{\partial y_i^2(k)}{\partial a_i^2(k)} = 1, \quad \frac{\partial a_i^2(k)}{\partial w_{ij}^2(k)} = y_j^1, \quad i = 1, 2, 3; \quad j = 1, 2, \dots, N^1. \quad)$$

Виходячи з дискретного закону роботи ПД-регулятора (1.23) та враховуючи співвідношення (1.25), формулюємо рівняння для часткових похідних.

$$\text{них.} \quad \frac{\partial u(k)}{\partial y_i^2(k)}$$

$$\frac{\partial u(k)}{\partial y_1^2(k)} = e(k) - e(k-1),$$

$$\frac{\partial u(k)}{\partial y_2^2(k)} = e(k-1),$$

$$\frac{\partial u(k)}{\partial y_3^2(k)} = e(k) - 2e(k-1) + e(k-2).$$

Для нейронів прихованого шару локальна помилка обчислюється як:

$$\delta_i^2 = e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_i^2(k)}. \quad (1.26)$$

Правило настройки вагів нейронів вихідного шару

$$w_{ij}^2(k+1) = w_{ij}^2(k) - \eta \delta_i^2 y_j^1, \quad i = 1, 2, 3; \quad j = 1, 2, \dots, N^1. \quad (1.27)$$

Гradient для прихованого шару визначається як

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^1} &= \sum_{n=1}^3 \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} \frac{\partial y_n^2(k)}{\partial a_n^2(k)} \frac{\partial a_n^2(k)}{\partial y_j^1(k)} \frac{\partial y_j^1(k)}{\partial a_j^1(k)} \frac{\partial a_j^1(k)}{\partial w_{ij}^1(k)} = \\ &= \sum_{n=1}^3 e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} w_{jn}^2 y_j^1(k) (1 - y_j^1(k)) d_j(k) = -\delta_i^1 d_j(k), \end{aligned}$$

де δ_i^1 – локальна помилка прихованого шару

$$\begin{aligned} \delta_i^1 &= \sum_{n=1}^3 e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} w_{jn}^2 y_j^1(k) (1 - y_j^1(k)) = \\ &= y_j^1(k) (1 - y_j^1(k)) \sum_{n=1}^3 \delta_i^1 w_{jn}^2. \end{aligned} \quad (1.28)$$

Правило корекції ваг прихованого шару:

$$w_{ij}^1(k+1) = w_{ij}^1(k) - \eta \delta_i^1 d_j, \quad i = 1, 2, \dots, N^1; \quad j = 1, 2, \dots, N^0 \quad (1.30)$$

На кожному кроці алгоритму для точності оновлення ваг необхідно обчислювати Якобіан об'єкта управління $\frac{\partial y(k+1)}{\partial u(k)}$, який можна оцінити за допомогою нейромодулятора.

Загальний алгоритм навчання ПД-нейрорегулятора з самонастроюванням:

1. Ініціалізація ваг w і зсувів b , встановлення норми навчання η та початкових параметрів ПД.
2. Подача на вхід мережі опорного сигналу $d(k)$, обчислення помилки $e(k+1)$ та локальних помилок вихідного δ_i^2 (по формулі (1.26)) і прихованого δ_i^2 шарів (по формулі (1.28)).
3. Корекція ваг вихідного шару по формулі (1.27).
4. Корекція ваг прихованого шару по формулі (1.30).
5. Повторення кроків 2–4 до досягнення необхідної точності регулювання.

1.2.6 Нелінійне предиктивне управління

Основний принцип управління за предиктивним методом, який базується на моделі (Model Predictive Control, MPC), представлений на рис. 1.12. Суть цього підходу полягає у формуванні послідовності управляючих сигналів для об'єкта таким чином, щоб мінімізувати розбіжність між заданим сигналом та виходом системи, що прогнозується моделлю на певному горизонті часу (позначеному пунктирними лініями на рис. 1.12).

Регулятор у цьому випадку працює як ітераційний оптимізаційний алгоритм [54–58], який має такі особливості:

- чисельні методи використовуються для знаходження мінімуму функціоналу;
- функціонал описує бажаний критерій якості регулювання;
- на об'єкт подається лише перший елемент оптимізованої послідовності сигналів;
- процес оптимізації повторюється на кожному дискретному кроці системи.

Головною перевагою MPC є здатність забезпечувати високу точність регулювання навіть при наявності складних нелінійностей об'єкта. Основний недолік методу полягає у високій обчислювальній складності та потребі у точних

нелінійних моделях системи. У рамках цієї магістерської роботи застосовується саме нелінійне предиктивне управління.

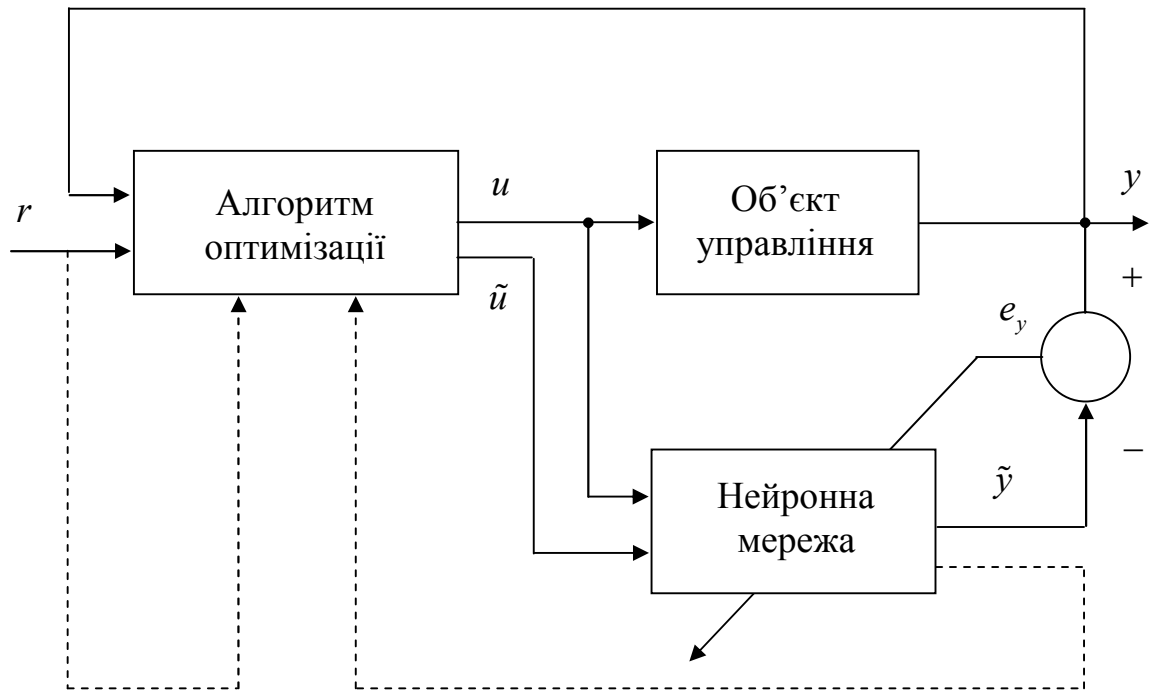


Рисунок 1.12 – Схема предиктивного управління

1.3 Загальні відомості про роторні екскаватори

Ключовим обладнанням гірничодобувних комплексів є роторний екскаватор – складна машина, що включає понад десять взаємодіючих силових механізмів та поєднує основне і допоміжне обладнання для транспортування видобутої породи до системи транспортних машин комплексу. Схематичне розташування компонентів роторного екскаватора наведено на рис. 1.13.

Конструкція машини поділяється на поворотну і неповоротну частини, що з'єднані опорно-поворотним механізмом 1, який забезпечує обертання верхньої частини екскаватора відносно вертикальної осі. До поворотної частини входить платформа, на якій змонтована стріла ротора 4 з робочим роторним колесом 3, оснащеним ковшами 2 по периферії. Під час обертання роторного колеса та переміщення стріли відбувається відділення породи, яка потрапляє на внутрішній конвеєр 16, розташований у фермовій конструкції стріли.

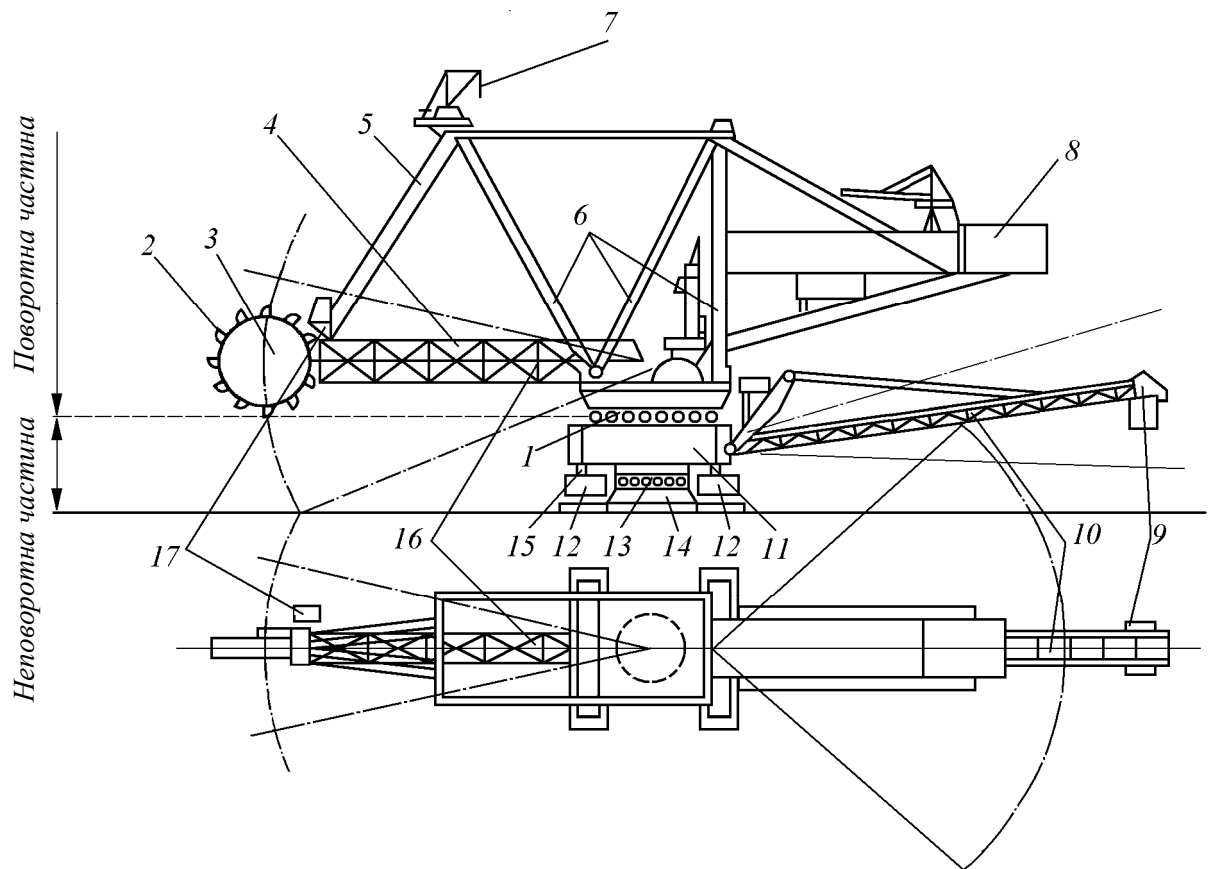


Рисунок 1.13 – Компонувальна схема роторного екскаватора

Далі порода транспортується до центральної частини екскаватора, де встановлено додроблюючий пристрій, проходить через площину опорно-поворотного механізму 1 і надходить на конвеєр нижньої частини машини, а потім – на конвеєр у розвантажувальній консолі 10.

Для регулювання положення роторного колеса у вертикальній площині стріла може займати робоче положення в межах певного сектора за допомогою канатної підвіски 5 і обертатися навколо осі поворотної частини. Верхнє положення стріли визначає максимальну висоту оброблюваного уступу, а нижнє – глибину копання нижче рівня стояння машини. Верхні блоки підвіски закріплені на порталі 6. Для балансування перекидаючого моменту стріли використовуються противаги, встановлені на консольній частині будови 8, що протилежна стрілі.

Нижче опорно-поворотного механізму розташований потужний силовий пояс 11, який з'єднується з опорною базою 14. До нерухомої частини кріплять-

ся гідродомкрати 15, здатні стабілізувати машину на ґрунті або через шагаюче-рейкове обладнання (лижі). Розвантажувальна консоль 10 з конвеєром може встановлюватися під різними кутами до напрямку руху екскаватора (до 80°), забезпечуючи автономність її просторового положення. Як і стріла ротора, консоль може змінювати кут у вертикальній площині.

Управління екскаватором здійснюється з кабін, розташованих біля роторного колеса 17, на кінці розвантажувальної консолі 10 та на нижній рамі. Для розворотів машини використовується додатковий поворотний механізм 13, що дозволяє обертати поворотну та неповоротну частини відносно бази. Для обслуговування вузлів і обладнання передбачено підйомні крани вантажопідйомністю 5–20 т у трьох точках: на поворотній платформі, верхній надбудові та похилому порталі 7.

Основні механізмами, що забезпечують робочий процес екскаватора:

- механізм приводу ротора;
- механізм підйому стріли;
- механізм повороту стріли;
- механізм розвороту консолі;
- механізм розвороту екскаватора;
- механізм ходу;
- конвеєр стріли;
- конвеєр розвантажувальної консолі.

Схема компоновки індивідуальних електроприводів роторного екскаватора показана на рис. 1.14.

У роторному колесі розташовані електродвигуни та редуктори для обертання 1, а також привід барабанного живильника 2, який направляє потік породи на конвеєр стріли. Електропривід конвеєра стріли 3 встановлений у зоні осі гойдання стріли.

На поворотній частині розташовані електроприводи 6, що забезпечують взаємний оберт верхньої та нижньої частин машини. Привід підйому стріли 5 встановлений у надбудові противаги, а в нижній частині розміщене вентиля-

ційне обладнання 4. Приводи дробильного пристрою 14 розташовані у верхньому поясі стріли.

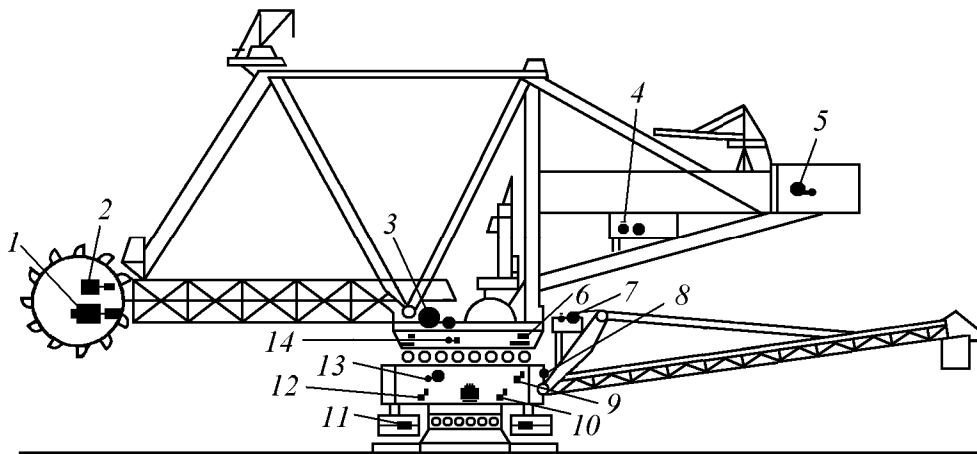


Рисунок 1.14 - Розташування електроприводів роторного екскаватора

Розвантажувальна консоль переміщується вертикально за допомогою електроприводів підйому 7 та обертання 9, а також приводів конвеєра 8. У неповоротній частині машини встановлені електроприводи механізму розвороту 12, конвеєра нижньої рами 13, а також обладнання для введення електроенергії: струмоприймачі, вимикачі, трансформатори та розподільні пристрої високої і низької напруги 10. Для руху по рейковому ходу передбачені електроприводи ходових лебідок 11.

1.4 Аналіз факторів, що визначають вибір системи управління електроприводом механізму розвороту роторного екскаватора

Ефективна робота роторного екскаватора можлива лише за умови безперебійного функціонування його ключових механізмів. Будь-яка відмова основного електроприводу призводить до порушення технологічного процесу або повного припинення розробки ґрунту, що спричиняє простій екскаватора. Збитки при цьому тим більші, чим вища продуктивність комплексу. Тому до електроприводів основних механізмів роторного екскаватора висуваються високі

вимоги щодо надійності роботи, стабільності характеристик та робочих параметрів.

З одного боку, дотримання цих вимог досягається за рахунок вибору оптимальних систем електроприводу, схем управління, електричних машин та апаратури, пристосованих до важких умов експлуатації. З іншого боку, для мінімізації простоїв електроустаткування, схеми управління не повинні вимагати частого переналадження та повинні забезпечувати швидке відновлення працездатності. Крім загальних вимог, для кожного електроприводу основних механізмів існують специфічні умови, що визначаються функціональними особливостями механізмів, конструкцією екскаватора, режимами роботи та характером навантажень.

Електропривод повороту стріли ротора забезпечує головний робочий рух – подовжню подачу для різання ґрунту – і, як і електропривод роторного колеса, працює в тривалому режимі. У той же час електроприводи повороту розвантажувальної консолі та розвороту екскаватора призначені для короткочасних робочих циклів.

Для ефективного оброблення забою необхідна зміна напрямку повороту стріли ротора на кожному проході. Розворот екскаватора забезпечує правильну орієнтацію машини відносно забою, а поворот розвантажувальної консолі підвищує гнучкість взаємодії з іншими механізмами комплексу. Це означає, що електроприводи опорно-поворотних пристроїв повинні бути реверсивними.

Однією з важливих особливостей електромеханічних систем електроприводів повороту є наявність пружно зв'язаних з платформою інерційних мас: стріли ротора з роторним колесом і консолі противаги. У таких системах перехідні процеси супроводжуються коливаннями, тому електроприводи повинні забезпечувати плавний розгін та гальмування з обмеженням ривків і прискорень до допустимого рівня.

Для цього в перехідних процесах формується оптимальна залежність швидкості від часу, що мінімізує динамічні навантаження. Це особливо важливо для електроприводу повороту стріли ротора, оскільки його допустимий ди-

намічний момент лише трохи перевищує статичний момент під час різання ґрунту.

Необхідний діапазон регулювання швидкості електроприводів опорно-поворотних механізмів становить щонайменше 3–4:1. Для електроприводів повороту розвантажувальної консолі та розвороту екскаватора це забезпечує швидке виконання поворотів при великих кутах, а для приводу повороту стріли ротора додається ще й вимога безступінчатого регулювання швидкості. Це потрібно для точного контролю товщини зрізуваної стружки в залежності від фізико-механічних властивостей ґрунту та для підтримання продуктивності машини.

Навантаження на опорно-поворотні механізми значно змінюються через можливий ухил бази екскаватора, вітрові навантаження та випадкові бічні сили різання. Тому електроприводи повинні підтримувати постійну робочу швидкість і мати високу жорсткість робочої ділянки механічної характеристики. Одночасно вони повинні ефективно гасити коливання, що виникають у багатомасових пружних системах, особливо в електромеханічній системі приводу повороту стріли, розташованої на кінці пружної стріли.

Механічна характеристика електроприводу повороту стріли повинна мати «екскаваторну форму», обмежувати момент у межах допустимого значення за умовами міцності конструкції та забезпечувати технологічну безпеку при перевантаженнях.

Складність і високі вимоги до регульованих електроприводів основних механізмів зумовили широке застосування систем постійного струму з керованим перетворювачем-двигуном. Для керованого перетворювача використовуються електромашинні перетворювачі (система Г-Д) або реверсивні тиристорні перетворювачі (система ТП-Д). Для електроприводу механізму розвороту роторного екскаватора застосовується система ТП-Д.

1.5 Постановка завдання синтезу нейромережевої системи управління

Аналіз сучасної наукової літератури показує, що існує велика кількість підходів до побудови систем управління, проте універсального або ідеального регулятора наразі не розроблено. При розробці системи управління механізмом повороту стріли роторного екскаватора необхідно враховувати особливості опорно-поворотних механізмів, які є багатомасовими пружними системами. У таких системах коливання виникають під дією як керуючих сигналів, так і зовнішніх збурень. Звичайні регулятори не здатні забезпечити прийнятний рівень якості функціонування у подібних умовах.

Застосування нейромережевих технологій дозволяє значно спростити математичне описання і синтез регулюючих систем. Це пояснюється тим, що характеристики процесів управління та досягнуті результати в нейромережевих системах визначаються насамперед фундаментальними властивостями багат шарових нелінійних нейронних мереж, а не аналітично виведеними оптимальними законами, реалізація яких зазвичай покладена на програмне забезпечення. Багат шарові нейронні мережі, що піддаються налаштуванню, мають низку переваг, що робить їх ефективними інструментами для управління динамічними об'єктами.

Універсальні апроксимаційні властивості багат шарових мереж є ключовими для побудови складних нелінійних алгоритмів управління. Виходячи з узагальненої теореми Стоуна–Вейерштрасса, багат шарова нелінійна нейронна мережа здатна з будь-якою точністю рівномірно наблизити будь-яку неперервну функцію багатьох змінних на замкненій обмеженій множині. Цей висновок забезпечує теоретичне підґрунтя для застосування таких мереж у формуванні оптимальних функцій управління у динамічних системах. Завдяки своїй структурі та алгоритмам навчання багат шарові мережі здатні ефективно розв'язувати нелінійні задачі теорії управління, які традиційні аналітичні методи не завжди дозволяють вирішити повністю. Можливість формувати керуючі

дії практично будь-якої складності робить штучні нейронні мережі природним інструментом для управління складними нелінійними системами.

Нейромережеві системи управління відносяться до класу «інтелектуальних» систем, тобто систем з неповною інформацією, складною або немодельованою динамікою і непередбачуваними змінами власних характеристик. Адаптивність багат шарових структур дозволяє в процесі функціонування коректувати закони управління в реальному часі, використовуючи поточні вимірювальні дані про стан об'єкта.

Паралельна обробка сигналів у нейронних мережах забезпечує їх природну придатність для керування багатовимірними або багатоканальними системами. Багат шарові мережі утворюють однорідне обчислювальне середовище та є універсальними паралельними обчислювальними структурами, які можна застосовувати для побудови регуляторів у системах управління нелінійними динамічними об'єктами.

Огляд сучасних досягнень та публікацій у галузі синтезу нейромережевих регуляторів демонструє, що цей напрямок має високу перспективність. Зокрема, застосування нейронних мереж у системах управління об'єктами зі складними кінематичними зв'язками є надзвичайно актуальним.

Метою цієї роботи є створення нейромережевої системи управління механізмом розвороту роторного екскаватора, яка враховує пружність механічних зв'язків платформи, а також інерційні властивості стріли ротора з роторним колесом та консолі противаги.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Виконати обґрунтований вибір системи електроприводу механізму розвороту екскаватора, підібрати електродвигун та основне силове обладнання, здійснити синтез системи управління електроприводом та провести її комп'ютерне моделювання.

2. Розробити математичну модель системи управління з урахуванням податливості механічних зв'язків, змодельовати багатомасову систему на ЕОМ та провести аналіз отриманих результатів.

3. Створити структурну схему нейромережевої системи управління, яка забезпечує високу якість регулювання з урахуванням кінцевої жорсткості елементів механічної частини електроприводу.

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

5. Використовуючи принцип предикативного управління на основі нейромережевої моделі об'єкта, розробити нейромережевий регулятор, здатний забезпечити високу швидкодію та ефективність управління.

6. Провести комп'ютерне моделювання розробленої нейромережевої системи та виконати детальний аналіз отриманих результатів.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. На основі аналізу наукових джерел за темою дослідження обґрунтовано доцільність застосування одного з класів штучних нейронних мереж – багатошарових нейронних мереж прямого розповсюдження – для задач автоматизованого управління. Сучасні методи синтезу систем автоматичного регулювання дозволяють досягати високих показників якості управління, проте їх практична реалізація пов'язана з низкою технічних труднощів. До них відносяться складності з вимірюванням деяких координат, наприклад, пружного моменту, а також обмеження, що виникають через специфіку реальних електроприводів. Тому актуальним є застосування методів, які дозволяють замінювати лінійні регулятори на нелінійні, здатні забезпечити необхідне керуюче впливання на об'єкт регулювання при меншій кількості вимірюваних сигналів. Одним із таких підходів є використання нейрорегуляторів, які функціонують за принципами біологічних нейронних мереж.

2. Самонавчання штучних нейронних мереж відкриває можливість використовувати нейрорегулятори навіть у умовах значних невизначеностей, тоді як традиційні методи адаптивного управління потребують великого обсягу апріорної інформації про об'єкт управління. Простота реалізації та здатність до навчання роблять нейронні мережі особливо ефективними для управління складними нелінійними системами в реальному часі, що є важливим для промислових об'єктів з динамічними змінами умов роботи.

3. Доведено ефективність застосування багатошарових мереж прямого розповсюдження для створення нейромережевої моделі об'єкта управління. Така мережа є універсальним аппроксиматором-класифікатором і є однією з найпоширеніших архітектур у практиці нейронних мереж. Було розглянуто методи навчання нейронних мереж, а також обрано підхід для навчання нейромережевого регулятора.

4. Для забезпечення оптимальної роботи об'єкта управління необхідно адаптивно коригувати параметри регуляторів у залежності від зміни умов робо-

ти. З цією метою доцільно розробляти системи управління на основі адаптивного підходу у поєднанні з методами штучних нейронних мереж. Такий підхід дозволяє створювати компактні нейромережеві моделі та автоматично підлаштовувати параметри моделей і регулятора під змінні умови експлуатації.

5. На основі аналізу сучасних структур нейрорегуюючих систем обрано принцип нелінійного предиктивного регулювання для побудови нейромережевого регулятора системи управління.

6. Проведено огляд існуючих методів опису нелінійних динамічних об'єктів і розглянуто принципи побудови нейромережевих моделей NARMAX та NARX. Моделі NARX відзначаються лінійною параметризацією та використанням стандартних алгоритмів навчання, що робить їх найбільш придатними для нейронної ідентифікації систем.

7. На підставі аналізу технічних вимог до електроприводів опорно-поворотних механізмів роторних екскаваторів було обрано електропривід постійного струму для механізму розвороту з реалізацією за системою ТП-Д.

8. Встановлено доцільність застосування нейромережевих технологій для синтезу системи управління механізмом повороту стріли роторного екскаватора. Це пояснюється наявністю пружно зв'язаних з платформою інерційних мас стріли ротора з роторним колесом та консолі противаги, що викликає значні коливання під час переходів. Традиційні системи управління не забезпечують необхідну якість регулювання для основних механізмів роторного екскаватора.

9. Проведено аналіз нейромережевих підходів до задач управління складними механізмами. Сформульовано дослідницьку задачу, яка полягає у розробці нейромережевої системи управління механізмом розвороту роторного екскаватора з урахуванням пружних механічних зв'язків, що дозволяє ефективно гасити коливання і підтримувати задані технологічні параметри роботи.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ РОЗВОРОТУ РОТОРНОГО ЕКСКАВАТОРА

2.1 Основні технічні характеристики роторного екскаватора та вимоги до системи управління

Для розрахунку електроприводу механізму повороту стріли роторного екскаватора та подальшого синтезу системи управління використовуються вихідні технічні дані, що наведені у таблиці 2.1. Ці параметри визначають як механічні, так і динамічні характеристики основних елементів екскаватора та дозволяють оцінити навантаження на систему управління.

Таблиця 2.1 – Основні технічні характеристики роторного екскаватора

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
1	Теоретична продуктивність за розри- хленою масою	$m^3 \cdot год$	Q	3900
2	Діаметр роторного колеса	m	D_p	12
3	Бічне зусилля різання	H	F_6	$12 \cdot 10^4$
4	Питоме зусилля копання	$кг/м^2$	k_f	$1,1 \cdot 10^6$
5	Радіус різання	m	R_0	37
6	Кутова швидкість повороту платформи	c^{-1}	$\omega_{пл}$	$0,6 \cdot 10^{-2}$
7	Момент площі поворотної платформи, що діє під впливом вітру	$m^2 \cdot m$	$M_{пв}$	2000
8	Швидкість руху транспортної стрічки	m/c	V_T	3,8
9	Маса поворотної платформи	$кг$	$m_{пл}$	$1000 \cdot 10^3$

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
10	Маса приймальної частини	кг	$m_{пч}$	$267 \cdot 10^3$
11	Маса відвальної частини	кг	$m_{вч}$	$250 \cdot 10^3$
12	Діаметр роликового круга	м	$D_{кр}$	14
13	Діаметр опорних роликів	м	$d_{тк}$	1
14	Момент інерції поворотної платфор- ми	кг · м ²	$J_{пл}$	$88 \cdot 10^7$
15	Частота обертання ротора	рад/хв	ω_p	5
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 на- вантаження	Па	$\lambda_{вт}$	300
19	Максимально подоланий ухил		γ	10°
20	Число двигунів механізму повороту		k	2
21	Максимально допустимі прискорен- ня і уповільнення платформи	с ⁻²	$\omega_{пл}$	0,005
22	ККД редуктора роторного колеса		η_p	0,8
23	ККД редуктора механізму повороту		η_p	0,45 – 0,65
24	Максимальний кут повороту		$\alpha_{п\ max}$	±75°

Вимоги до показників якості функціонування системи управління

Система управління механізмом розвороту роторного екскаватора повинна забезпечувати стабільну роботу без постійних коливань та перебоїв. Основні показники якості її функціонування визначаються наступним чином:

- Перерегулювання швидкості при подачі ступінчастого сигналу не повинно перевищувати 10%.
- Час встановлення вихідного параметра до стабільного значення (час регулювання) не повинен перевищувати 5 секунд.

Ці вимоги є критично важливими для забезпечення безпечної та ефективної роботи роторного екскаватора, збереження його механічних компонентів та досягнення заданої продуктивності при роботі в різних умовах експлуатації.

2.2 Розрахунок потужності та вибір електродвигунів механізму розвороту роторного екскаватора

Відповідно до загальних рекомендацій, потужність електроприводу механізму розвороту екскаватора рекомендується приймати в межах 0,8–1,2 від потужності приводу повороту стріли. Тому спочатку визначимо необхідну потужність електроприводу повороту стріли, а потім на її основі оберемо відповідні електродвигуни для механізму розвороту.

Момент M_{δ} , що виникає внаслідок дії зусиль на кромках ковшів при копанні ґрунту і спрямований перпендикулярно осі стріли ротора F_{δ} , розраховується за формулою:

$$M_{\delta} = F_{\delta} R_{\kappa}, \quad (2.1)$$

де R_{κ} – радіус різання в горизонтальній площині, вимірюваний по кромках ковшів

$$R_{\kappa} = R_0 \cos \beta_c, \quad (2.2)$$

β_c – кут нахилу стріли ротора відносно горизонту.

У якості розрахункового значення M_{δ} приймаємо максимальне значення моменту, що відповідає положенню стріли, при якому $R_0 = R_{\kappa}$

$$M_{\delta} = M_{\delta \max} = F_{\delta} R_0, \quad (2.3)$$

$$M_{\delta} = M_{\delta \max} = 12 \cdot 10^4 \cdot 37 = 444 \cdot 10^3 \text{ Нм}.$$

На величину моменту, який діє на поворотну частину, впливає вітер. При постійній швидкості вітру момент від аеродинамічного навантаження $M_{\text{пв}}$ визначається як:

$$M_{\text{вт}} = M_{\text{вт max}} \cos \alpha_{\text{п}} = k_{\text{ад}} \lambda_{\text{вт}} M_{\text{пв}} \cos \alpha_{\text{п}} , \quad (2.4)$$

де $M_{\text{вт max}}$ – максимальне вітрове навантаження (при $\alpha_{\text{п}} = 0$)

$k_{\text{ад}}$ – коефіцієнт аеродинамічного опору;

$\lambda_{\text{вт}}$ – максимально допустиме вітрове навантаження;

$$M_{\text{вт}} = M_{\text{вт max}} \cos \alpha_{\text{п}} = 1,4 \cdot 300 \cdot 2000 \cdot \cos \alpha_{\text{п}} = 84 \cdot 10^4 \cdot \cos \alpha_{\text{п}} \text{ Нм} .$$

Момент, обумовлений неврівноваженими силами тяжіння через відхилення осі повороту від вертикалі на кут γ та повороту стріли на кут $\alpha_{\text{п}}$, визначається формулою:

$$M_{\text{нв}} = G_{\text{нв}} R_{\text{цт}} \sin \alpha_{\text{п}} \sin \gamma , \quad (2.5)$$

де $R_{\text{цт}}$ – радіус прикладення сили тяжіння неврівноваженої частини $G_{\text{нв}}$.

Приймаємо:

$$G_{\text{нв}} R_{\text{цт}} = g[(m_{\text{пч}} + m_{\text{в пч}}) - m_{\text{вч}}] \frac{L_{\text{пч}}}{3} ,$$

$m_{\text{пч}}$, $m_{\text{вч}}$, $m_{\text{в пч}}$ – маса поворотної та відвальної частин, маса вантажу на транспортерах

$$m_{\text{в пч}} = \frac{Q \gamma_{\text{гр}} L_{\text{пч}}}{3600 \cdot V_{\text{т}} k_{\text{р}}} ;$$

Q – теоретична продуктивність по рихлій масі;

$\gamma_{\text{гр}}$ – щільність ґрунту;

$L_{\text{пч}}$ – довжина приймальної частини транспортера, приймаємо $L_{\text{пч}} = R_0$;

$V_{\text{т}}$ – швидкість руху транспортної стрічки;

k_p – коефіцієнт розпушування ґрунту, $k_p = 1,1 - 1,5$

$$m_{\text{впч}} = \frac{3900 \cdot 1,8 \cdot 10^3 \cdot 37}{3600 \cdot 3,8 \cdot 1,5} = 12,4 \cdot 10^3 \text{ кг},$$

$$G_{\text{нв}} R_{\text{цт}} = 9,8 \cdot [(267 + 12,4) - 250] \cdot 10^3 \cdot \frac{37}{3} = 361,9 \cdot 10^4 \text{ Нм},$$

$$M_{\text{нв}} = 361,9 \cdot 10^4 \cdot \sin 10^\circ \cdot \sin \alpha_{\text{п}} = 62,4 \cdot 10^4 \cdot \sin \alpha_{\text{п}} \text{ Нм}.$$

Момент опору обертання, що виникає через сили тертя, розраховується за формулою:

$$M_{\text{тр}} = g(m_{\text{пл}} + m_{\text{впч}}) \frac{D_{\text{кр}}}{2d_{\text{тк}}} f_{\text{к}} f_{\text{пч}} f_{\text{ек}},$$

де $m_{\text{пл}}$ – маса поворотної платформи;

$D_{\text{кр}}$ – діаметр роликового круга;

$d_{\text{тк}}$ – діаметр опорних роликів;

$f_{\text{к}}$ – коефіцієнт тертя кочення $f_{\text{к}} = (0,04 - 0,1) \cdot 10^{-2}$ м;

$f_{\text{пр}}$ і $f_{\text{ек}}$ коефіцієнти, що враховують прослизання роликів циліндрової форми ($f_{\text{пр}} = 1,1 - 1,25$) і ексцентриситет навантаження ($f_{\text{ек}} = 1,0 - 1,25$)

$$M_{\text{тр}} = 9,81 \cdot (1000 + 12,4) \cdot 10^3 \cdot \frac{14}{2 \cdot 1} \cdot 0,07 \cdot 1,1 \cdot 1,2 \cdot 10^{-2} = 64,3 \cdot 10^3 \text{ Н} \cdot \text{м}$$

Складові моменти $M_{\text{вт}}$ і $M_{\text{нв}}$ змінюються залежно від кута повороту та можуть бути як активними (допомагають руху), так і гальмівними. Моменти $M_{\text{б}}$ і $M_{\text{тр}}$ є пасивними і завжди чинять гальмівну дію.

Сумарне статичне навантаження на механізм визначається як:

$$M_{\text{ст}} = M_{\text{б}} + M_{\text{вт}} + M_{\text{тр}} + M_{\text{нв}}.$$

На практиці для більшості екскаваторів сумарний момент становить приблизно:

$$M_{\text{вт}} + M_{\text{тр}} + M_{\text{нв}} \cong 0,5M_{\text{б}}.$$

Потужність двигунів повороту платформи визначається за виразом:

$$P_{\text{д}} = \frac{1,5M_{\text{б}}\omega_{\text{пл}}}{\eta_{\text{п}}k} \cdot 10^{-3},$$

де $\eta_{\text{п}}$ – ККД механізму;

$\omega_{\text{пл}}$ – кутова швидкість повороту платформи;

k – число приводів, приймаємо $k = 2$;

$$P_{\text{д}} = \frac{1,5 \cdot 444 \cdot 0,6 \cdot 10^{-2}}{0,6 \cdot 2} \cdot 10^{-3} = 33,3 \text{ кВт}.$$

Враховуючи ці розрахунки, потужність двигунів механізму розвороту екскаватора приймається рівною:

$$P_{\text{д}} = (1 \div 0,8) \cdot 33,3 = 33,3 \div 26,64 \text{ кВт}.$$

Для реалізації обрано два однакових двигуни ДВ-808 з технічними характеристиками: потужність $P_{\text{н}} = 37 \text{ кВт}$, частота обертання $n_{\text{н}} = 575 \text{ об/хв}$, напруга $U_{\text{н}} = 220 \text{ В}$, струм якоря $I_{\text{н}} = 192 \text{ А}$, тривалість включення: ПВ=100%, магнітний потік полюса $\Phi_{\text{н}} = 38 \text{ мВб}$; число полюсів $2p = 4$ опір обмоток при 20°C: якоря $R_{\text{я}} = 0,034 \text{ Ом}$, додаткових полюсів $R_{\text{дп}} = 0,02 \text{ Ом}$; момент інерції якоря $J_{\text{д}} = 2 \text{ кг} \cdot \text{м}^2$; $\eta = 89\%$

Коректність вибору двигунів буде перевірятися після побудови швидкісної діаграми та діаграми навантаження механізму.

2.3 Розрахунок швидкісної діаграми механізму розвороту

Визначимо передавальне число редуктора, яке можна отримати зі співвідношення між кутовою швидкістю робочого органа та частотою обертання електродвигуна:

$$i_{\pi} = \frac{\omega_{\pi}}{\omega_{\pi\pi}},$$

де ω_{π} – кутова швидкість обертання двигуна;

$\omega_{\pi\pi}$ - кутова швидкість повороту платформи

$$\omega_{\pi} = \frac{\pi \cdot n_{\pi}}{30},$$

$$\omega_{\pi} = \frac{3,14 \cdot 575}{30} = 60,2 \text{ c}^{-1},$$

$$i_{\pi} = \frac{60,2}{0,6 \cdot 10^{-2}} = 10035.$$

Швидкісна діаграма механізму повороту стріли роторного екскаватора класично складається з трьох основних ділянок.

1. фаза інтенсивного розгону з максимально допустимим прискоренням;
2. рух із стабільною (номінальною) швидкістю, коли момент навантаження врівноважується моментом двигуна;
3. гальмування, що здійснюється з припустимим уповільненням, яке не повинно перевищувати експлуатаційні обмеження.

Оскільки рух платформи передається через редуктор, фактичні значення прискорень $\dot{\omega}_{\pi\pi}$ потрібно привести до валу двигуна:

$$\dot{\omega}_{\text{доп}} = \dot{\omega}_{\pi\pi} \cdot i_{\pi};$$

$$\dot{\omega}_{\text{доп}} = 0,005 \cdot 10035 = 50,175 \text{ c}^{-2}.$$

З використанням отриманих значень прискорення $\dot{\omega}_{\pi\pi}$ визначаємо часові

інтервали розгону та гальмування:

$$t_p = t_r = \frac{\omega_n}{\dot{\omega}_{\text{доп}}},$$

$$t_p = t_r = \frac{60,21}{50,175} = 1,2 \text{ с}.$$

Кути повороту, пройдені платформою під час розгону та гальмування:

$$\alpha_p = \alpha_r = \dot{\omega}_{\text{пл}} \frac{t_p^2}{2},$$

$$\alpha_p = \alpha_r = 0,005 \cdot \frac{1,2^2}{2} = 3,6 \cdot 10^{-3} \text{ рад},$$

$$\alpha_p = \alpha_r = 0,206^\circ.$$

Кут, що відповідає режиму сталої швидкості:

$$\alpha_y = 2 \cdot (\alpha_{\text{п max}} - \alpha_p),$$

$$\alpha_y = 2 \cdot (75 - 0,206) = 149,57^\circ,$$

$$\alpha_y = 2,61 \text{ рад}.$$

Тривалість інтервалу сталої швидкості:

$$t_y = \frac{\alpha_y \cdot i_{\text{п}}}{\omega_n},$$

$$t_y = \frac{2,61 \cdot 10035}{60,2} = 435,167 \text{ с}.$$

На підставі цих розрахунків формується повна швидкісна діаграма механізму повороту (рис. 2.1,а), яка містить залежність кутової швидкості від часу та чітко відображає всі три етапи руху.

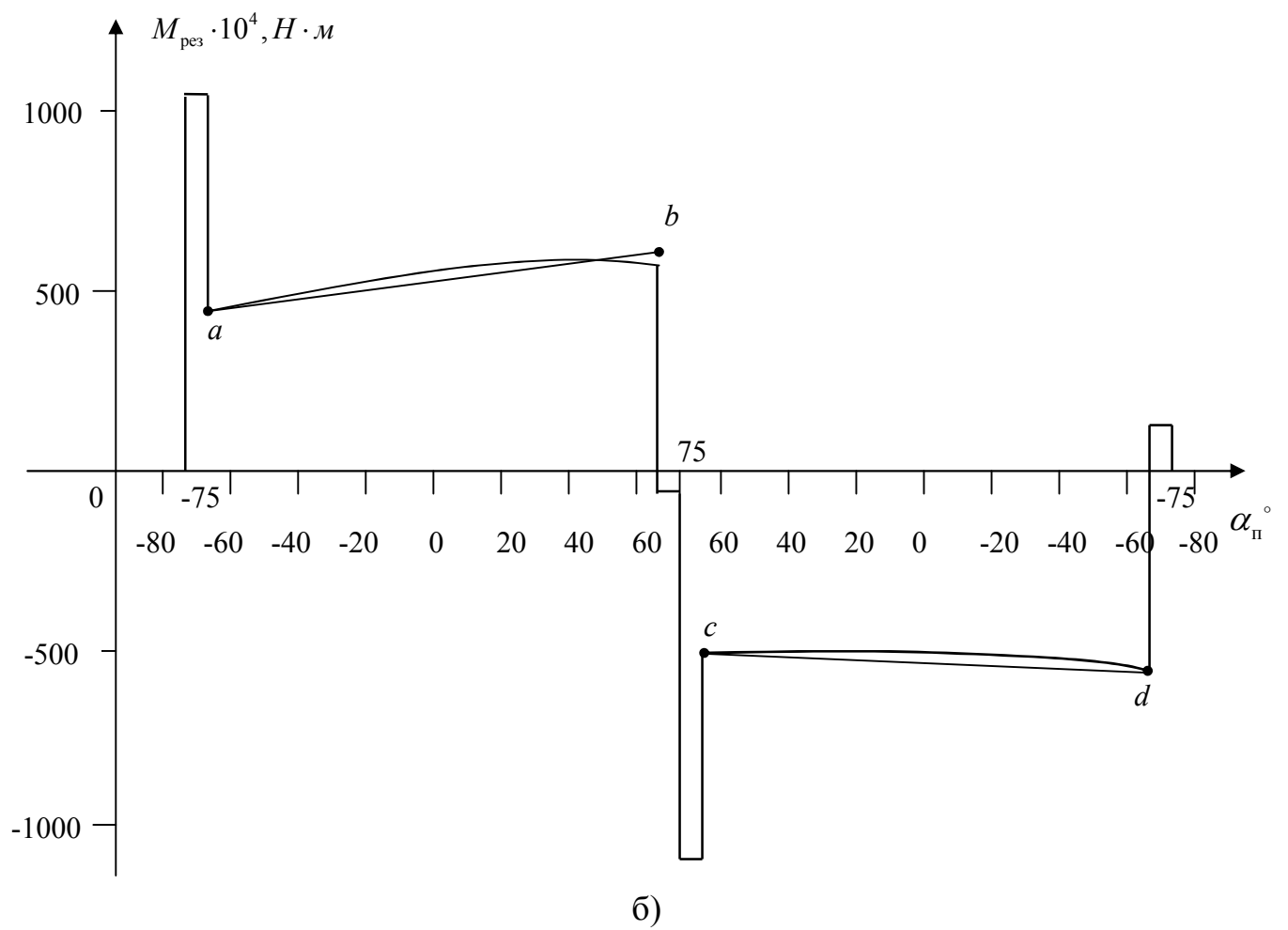
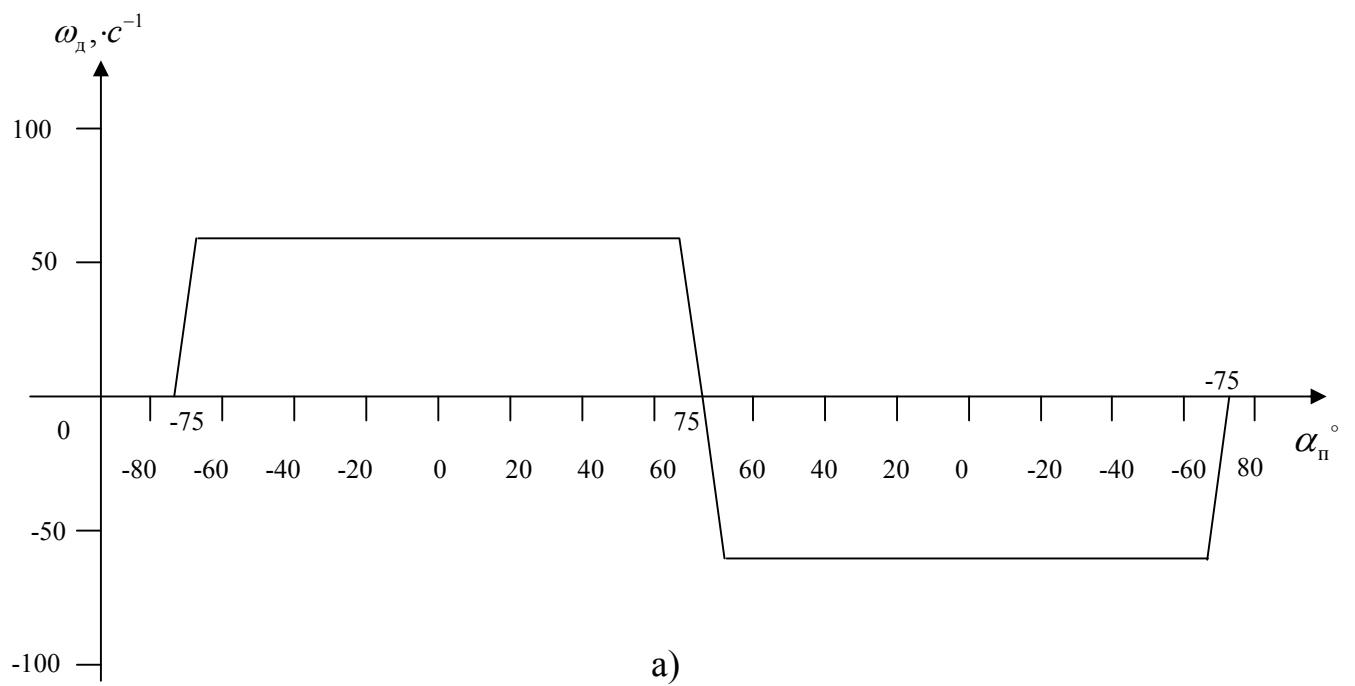


Рисунок 2.1 – Швидкісна а) та навантажувальна б) діаграми механізму повороту роторного екскаватора

2.4 Аналіз навантажувальної діаграми електроприводу та перевірка вибору двигунів

Визначаємо складові статичного моменту, що діють на механізм розвороту. Розрахунок виконується для обох напрямків обертання – як при зміні кута від -75° до $+75^\circ$, так і в протилежному напрямку.

У фазах розгону та гальмування до статичного моменту додається або віднімається динамічна складова, що враховує інерційність системи:

$$M_{\text{дин}} = J_{\text{м}\Sigma} \cdot \dot{\omega}_{\text{пл}},$$

де $J_{\text{м}\Sigma}$ – приведений момент інерції всієї системи.

$$J_{\text{м}\Sigma} = J_{\text{пл}} + kJ_{\text{д}}i_{\text{п}}^2,$$

$J_{\text{пл}}$ – момент інерції поворотної платформи;

$J_{\text{д}}$ – момент інерції якоря одного двигуна

$$J_{\text{м}\Sigma} = 88 \cdot 10^7 + 2 \cdot 2 \cdot 10035^2 = 12,83 \cdot 10^8 \text{ кгм}^2,$$

$$M_{\text{дин}} = 12,83 \cdot 10^8 \cdot 0,005 = 641,4 \cdot 10^4 \text{ Нм}.$$

Результуючий момент визначається за правилом:

$$M_{\text{рез}} = M_{\text{ст}} \pm M_{\text{дин}}.$$

Знак «+» відповідає розгону, знак «-» - гальмуванню приводу.

Для кожного діапазону кутів статичний момент визначається наступним чином:

$$\alpha_{\text{п}} = (-75^\circ \div 0) \quad M_{\text{ст}} = |M_{\text{б}}| + |M_{\text{тр}}| + |M_{\text{вт}}| - |M_{\text{нв}}|;$$

$$\alpha_{\text{п}} = (0 \div +75^\circ) \quad M_{\text{ст}} = |M_{\text{б}}| + |M_{\text{тр}}| + |M_{\text{вт}}| + |M_{\text{нв}}|;$$

$$\alpha_{\Pi} = (+75^{\circ} \div 0) \quad M_{\text{ст}} = -(|M_{\text{б}}| + |M_{\text{тр}}| - |M_{\text{вт}}| - |M_{\text{нв}}|);$$

$$\alpha_{\Pi} = (0 \div -75^{\circ}) \quad M_{\text{ст}} = -(|M_{\text{б}}| + |M_{\text{тр}}| - |M_{\text{вт}}| + |M_{\text{нв}}|).$$

Всі результати зведені в таблицю 2.2, після чого побудована навантажувальна діаграма (рис. 2.1,б).

Перевірка двигунів за нагрівостійкістю.

Щоб переконатися, що обрані двигуни працюватимуть без перегріву, використовується метод еквівалентного моменту:

$$M_{\text{д екв}} = \frac{1}{\eta_{\Pi} i_{\Pi} k} \cdot \sqrt{\frac{\sum_{i=1}^n M_{\text{рез},i}^2 \Delta t_{\Pi,i}}{T_{\Pi}}},$$

Для визначення $M_{\text{д екв}}$ необхідно визначити час циклу T_{Π} , а також середні значення моменту на кожній з ділянок Δt_i , що позначені як $M_{\text{рез},i}$. В межах одного циклу враховуються відрізки розгону, уповільнення та руху з постійною кутовою швидкістю як у прямому, так і в зворотному напрямках. Таким чином, загальна кількість ділянок, для яких обчислюється середній момент, становить шість.

$$T_{\Pi} = 2(t_{\text{р}} + t_{\text{г}} + t_{\text{у}}),$$

$$T_{\Pi} = 2 \cdot (1 + 1 + 502) = 1008 \text{ с}.$$

На етапах розгону й гальмування результуючий момент визначається композицією статичної складової та динамічного моменту прискорення, тому для цих зон отримуємо:

$$M_{\text{р1}} = 1172 \text{ Нм} \quad M_{\text{г1}} = 104 \text{ Нм},$$

$$M_{\text{р2}} = 1098 \text{ Нм} \quad M_{\text{г2}} = 30 \text{ Нм}.$$

Таблиця 2.2 – Розрахунок діаграми навантаження

На- прям руху	Складові моменти	α_{Π}																
		-75°	-75° $-\alpha_p$	-70°	-65°	-60°	-55°	-50°	-45°	-40°	-35°	-30°	-25°	-20°	-15°	-10°	-5°	0°
	$M_6 + M_{\text{тр}} \cdot 10^4$	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43
	$M_{\text{вт}} \cdot 10^4$	21,74	22,04	28,73	35,5	42,0	48,18	53,99	59,4	64,36	68,812	72,75	76,13	78,93	81,14	82,72	83,68	84
	$M_{\text{нв}} \cdot 10^4$	-60,84	-60,78	-59,19	-57,09	-54,55	-51,6	-48,25	-44,54	-40,49	-34,13	-31,5	-26,62	-21,54	-16,3	-10,94	-5,49	0
	$M_{\text{дин}} \cdot 10^3$	641,4	641,4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
По годинниковій стрілці	$M_{\text{ст}} \cdot 10^3$	411,33	411,69	419,97	428,84	437,88	447,01	456,17	465,29	474,29	483,11	491,68	499,94	507,82	515,27	522,21	528,62	534,43
	$M_{\text{рез}} \cdot 10^3$	1052,73	1052,8	419,97	428,84	437,88	447,01	456,17	465,29	474,29	483,11	491,68	499,94	507,82	515,27	522,21	528,62	534,43
	t, c	0	1,2	14,585	29,17	43,755	58,34	72,925	87,51	102,095	116,68	131,265	145,85	160,435	175,02	189,605	204,19	218,775
Проти годинни- кової стрілки	$M_{\text{ст}} \cdot 10^4$	-489,53	-489,17	-480,89	-472,02	-462,98	-453,85	-444,69	-435,57	-426,57	-417,75	-409,18	-400,92	-393,04	-385,59	-378,65	-372,24	-366,43
	$M_{\text{рез}} \cdot 10^4$	151,87	152,23	-480,89	-472,02	-462,98	-453,85	-444,69	-435,57	-426,57	-417,75	-409,18	-400,92	-393,04	-385,59	-378,65	-372,24	-366,43
	t, c	875,1	873,9	860,515	845,93	831,345	816,76	802,175	787,59	773,005	758,42	743,835	729,25	714,665	700,08	685,495	670,91	656,325

Продовження таблиці 2.2

На- прям руху	Складові моме- нти	α_n															
		5°	10°	15°	20°	25°	30°	35°	40°	45°	50°	55°	60°	65°	70°	75° $-\alpha_p$	75°
	$M_6 + M_{тр} \cdot 10^4$	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43	450,43
	$M_{вт} \cdot 10^4$	83,68	82,72	81,14	78,93	76,13	72,75	68,81	64,35	59,4	53,99	48,18	42,0	35,5	28,73	22,04	21,74
	$M_{нв} \cdot 10^4$	5,49	10,94	16,3	21,54	26,62	31,5	36,13	40,49	44,54	48,25	51,6	54,55	57,09	59,19	60,78	60,84
	$M_{дин} \cdot 10^3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-641,4	-641,4
По годинниковій стрілці	$M_{ст} \cdot 10^3$	539,60	544,09	547,87	550,9	553,18	554,68	555,37	555,27	554,37	552,67	550,21	546,98	543,02	538,35	533,25	533,01
	$M_{рез} \cdot 10^3$	539,60	544,09	547,87	550,9	553,18	554,68	555,37	555,27	554,37	552,67	550,21	546,98	543,02	538,35	-108,15	-108,39
	t, c	233,36	247,945	262,53	277,115	291,7	306,285	320,87	335,455	350,04	364,625	379,21	393,795	408,38	422,965	436,35	437,55
Проти годиннико- вої стрілки	$M_{ст} \cdot 10^4$	-361,26	-356,77	-352,99	-349,96	-347,68	-346,18	-345,49	-345,59	-346,49	-348,19	-350,65	-353,88	-357,84	-362,51	-367,61	-367,85
	$M_{рез} \cdot 10^4$	-361,26	-356,77	-352,99	-349,96	-347,68	-346,18	-345,49	-345,59	-346,49	-348,19	-35-,65	-353,99	-357,84	-362,51	-1009,01	-1009,25
	t, c	641,74	627,155	612,57	597,985	583,4	568,815	554,23	539,645	525,06	510,475	495,89	481,305	466,72	452,135	438,75	437,55

У випадку руху з усталеною швидкістю момент навантаження змінюється в залежності від кута повороту, тому він не може бути прийнятий сталою величиною. Щоб визначити його усереднене значення, скористаємося співвідношенням:

$$M_{cp} = \sqrt{\frac{M_1^2 + M_1 M_2 + M_2^2}{3}}.$$

Розглянемо ділянку ab . Підставивши відповідні значення моментів на початку та в кінці цієї зони, дістанемо:

$$M_1 = 420 \text{ Нм}, \quad M_2 = 540 \text{ Нм},$$

$$M_c = \sqrt{\frac{(420^2 + 420 \cdot 540 + 540^2) \cdot 10^8}{3}} = 489 \cdot 10^4 \text{ Нм}.$$

Аналогічно для проміжку cd :

$$M_1 = 355 \text{ Нм}, \quad M_2 = 480 \text{ Нм},$$

$$M_c = \sqrt{\frac{(355^2 + 355 \cdot 480 + 480^2) \cdot 10^8}{3}} = 431 \cdot 10^4 \text{ Нм},$$

$$M_{д\text{ екв}} = \frac{1}{0,5 \cdot 12040 \cdot 2} \cdot \sqrt{\frac{1053^2 \cdot 1 + 489^2 \cdot 502 + 108^2 \cdot 1 + 1009^2 \cdot 1 + 431^2 \cdot 502 + 152^2 \cdot 1}{1008}} = 418,73 \text{ Нм}.$$

Надалі порівняємо отримані розрахункові величини із номінальними параметрами електродвигуна. Його паспортний момент визначається співвідношенням:

$$M_{н} = \frac{P_{н}}{\omega_{н}}.$$

Після підстановки числових даних одержуємо:

$$M_n = \frac{3700}{60,2} = 614,6 \text{ Нм}.$$

Отже, порівняння показує, що еквівалентний момент, знайдений на основі навантажувальної діаграми, менший за номінальний. Це означає, що двигун працюватиме в безпечному тепловому режимі, і перегрівання в реальних умовах експлуатації не виникатиме.

Далі необхідно перевірити здатність приводу витримувати короточасні пікові перевантаження. Максимальне значення моменту, приведене до валу двигуна і визначене з діаграми навантаження, має бути меншим від максимального допустимого значення $2,5M_n$, тобто повинна виконуватися нерівність:

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} \leq 2,5M_n.$$

Підставивши конкретні величини моментів, переконуємось, що дане обмеження також виконується:

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} = \frac{1052,8 \cdot 10^4}{0,6 \cdot 10035 \cdot 2} = 874,2 \text{ Нм},$$

$$2,5M_n = 2,5 \cdot 614,6 = 1525 \text{ Нм}.$$

Таким чином, попередньо підібраний електродвигун повністю відповідає вимогам щодо теплової стійкості та перевантажувальної здатності, а отже може бути використаний як виконавчий елемент приводу механізму розвороту екскаватора.

2.5 Вибір силового електроустаткування електроприводу

У сучасних роторних екскаваторах найбільше поширення одержали тиристорні перетворювачі серій БУВ та ШУВ, які традиційно використовують як основні станції керування електроприводами. Їх типові функціональні схеми наведено на рис. 2.2.

Конструктивно такі тиристорні станції керування формуються з ряду взаємопов'язаних модулів і вузлів. До їх складу входять:

- ВК – випрямний комплект (у реверсивних системах використовується два комплекти – прямого ходу *ВКВ* та зворотного ходу *ВКН*);
- ТОР – реактор, що вводиться у коло для обмеження швидкості наростання струму;
- А – автоматичний вимикач, який забезпечує захист від перевантажень та коротких замикань;
- ДТ – датчик струму для локального вимірювання миттєвих значень навантаження;
- БЗ – блок захисту, який контролює аварійні режими;
- СІФУ – система імпульсно-фазового керування, що задає кути відкриття тиристорів;
- РП – регулюючий підсилювач, який формує сигнал керування необхідної величини;
- ЛП – логічний модуль (наявний лише у реверсивних перетворювачах), що керує перемиканням випрямних комплектів;
- БП – джерело живлення допоміжних кіл;
- В – вентилятор примусового охолодження (за потреби, залежно від температурного режиму тиристорів).

У реверсивних модифікаціях перетворювачів БУВ та ШУВ застосовано принцип *роздільного керування* групами тиристорів. Обидва комплекти вентилів керуються однією спільною СІФУ, що значно спрощує синхронізацію і підвищує точність регулювання.

Такі системи керування орієнтовані на роботу у замкнутих контурах автоматичного регулювання, де підтримується певна координата приводу – напруга, кутова швидкість, струм тощо. На вхід станції подається сигнал помилки $\Delta U = U_3 - U_{ззш}$ де U_3 – задавальна напруга, що відповідає потрібному значенню регульованого параметра, а $U_{ззш}$ – сигнал зворотного зв'язку, пропорційний фактичному значенню цієї координати.

У нереверсивних схемах цей сигнал подають безпосередньо на РП, основним призначенням якого є підсилення напруги помилки та передача її на СІФУ для формування керуючих імпульсів.

У разі введення зворотного зв'язку за швидкістю або напругою істотно зростає жорсткість механічної характеристики приводу. Крім того, зворотний зв'язок необхідний:

- для лінеаризації характеристик у зоні переривчастої струмової провідності;
- для забезпечення стійкої роботи реверсивного приводу, в якому комплект тиристорів вмикається з роздільним керуванням.

Тиристорні перетворювачі серій БУВ і ШУВ побудовані за схемою трифазного реверсивного мосту. Анодний трансформатор у них не застосовується, а для покращення електромагнітних параметрів у коло увімкнено токообмежувальний реактор.

Для приводу механізму розвороту було обрано тиристорний перетворювач ШУВ 3601-44, технічні характеристики якого наведені нижче:

- номінальна потужність – 147 кВт;
- випрямлена напруга – 460 В;
- номінальний випрямлений струм – 320 А.

Значення індуктивності токообмежувального реактора становить $L_{\text{тор}} = 0,174 \cdot 10^{-3} \text{ Гн}$.

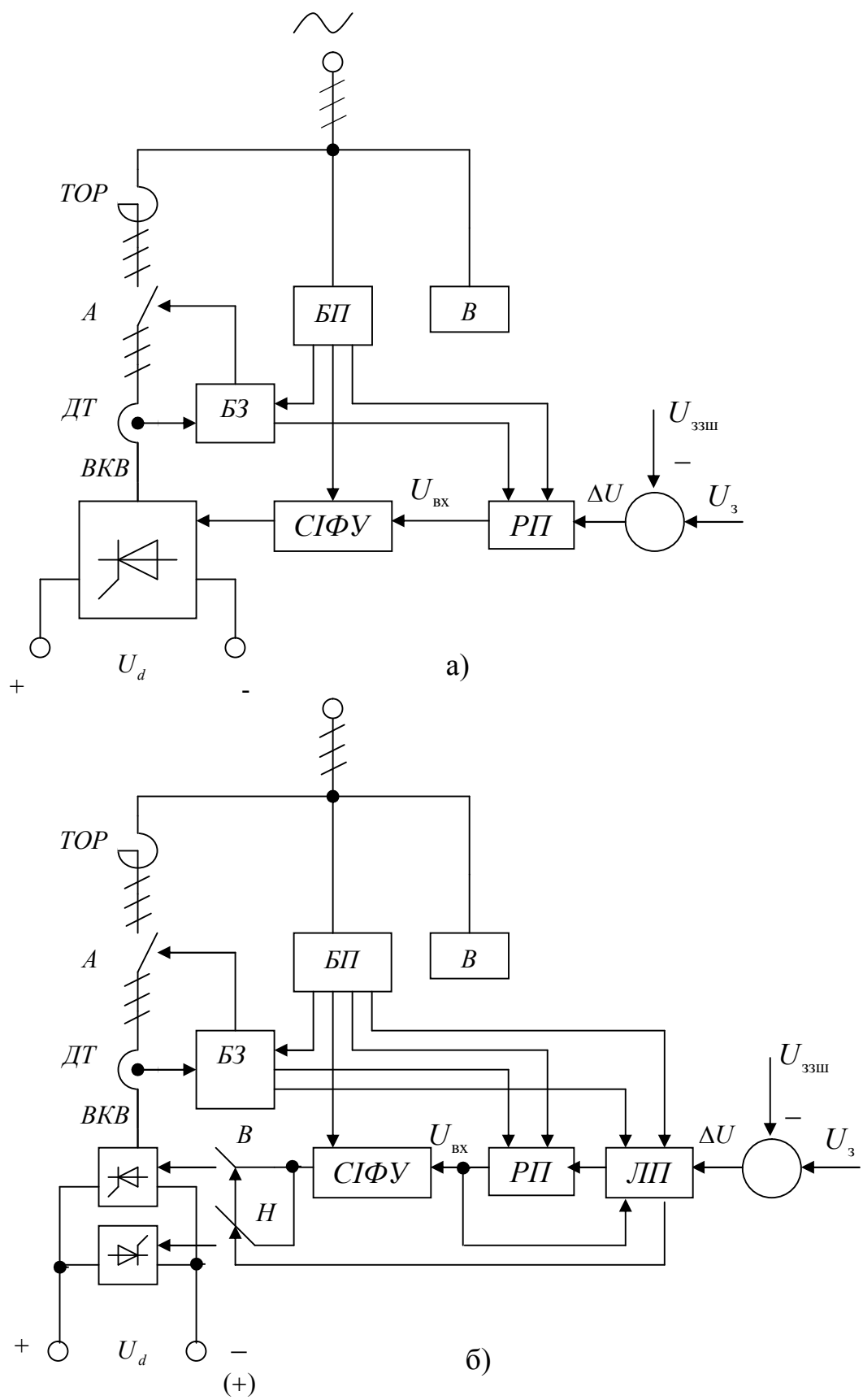


Рисунок 2.2 – Тиристорні станції управління типів БУВ (а) та ШУВ (б)

2.6 Аналіз і розрахунок статичних характеристик розімкненої та замкненої системи електроприводу

На рис. 2.3 подано структурну схему силового кола системи ТП–Д..

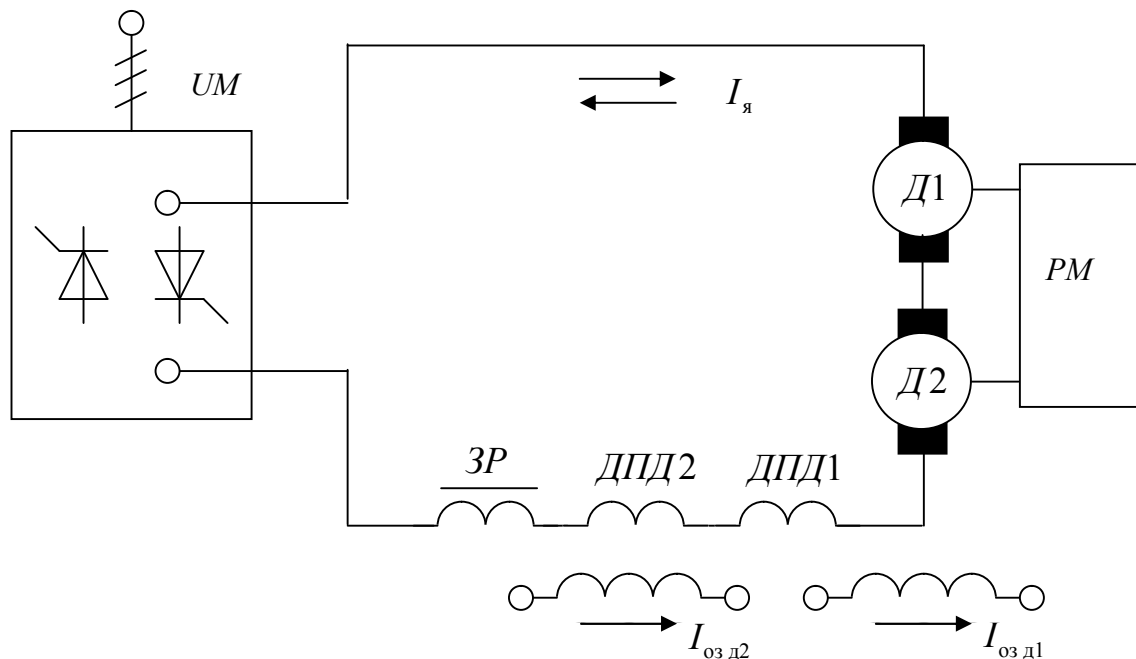


Рисунок 2.3 – Схема силового кола випрямленого струму приводу ТП–Д

На схемі позначено такі елементи: UM – силовий підсилювач, що реалізований на тиристорному перетворювачі; CP – згладжуючий реактор; PM – механізм, на який передається момент; $Д1$, $Д2$ – електродвигуни приводу розвороту; $ДПД1$, $ДПД2$ – обмотки додаткових полюсів двигунів; $I_я$ – струм якорного кола; $I_{зб\ д1}$, $I_{зб\ д2}$ – струми обмоток збудження кожного двигуна.

Як видно з схеми, якорні кола двигунів утворюють **єдиний послідовний ланцюг** разом із тиристорним перетворювачем. Якщо розглядати ЕДС перетворювача $E_{тп}$ як еквівалентне джерело живлення, тоді рівняння, що описують електромеханічну та механічну характеристики одного двигуна, набувають вигляду:

$$\omega_d = \frac{E_{тп}}{k k_d \Phi_{нд}} - I_я \frac{R_{я \Sigma}}{k k_d \Phi_{нд}} = \omega_0 - \Delta \omega_c ; \quad (2.6)$$

$$\omega_d = \frac{E_{\text{тп}}}{k k_d \Phi_{\text{нд}}} - M_d \frac{R_{\Sigma}}{k (k_d \Phi_{\text{нд}})^2} = \omega_0 - \Delta\omega_c, \quad (2.7)$$

де M_d – момент одного двигуна;

k_d – конструктивний коефіцієнт двигуна;

$\Phi_{\text{нд}}$ – номінальний магнітний потік двигуна.

Добуток k_d на $\Phi_{\text{нд}}$ можна знайти, використовуючи співвідношення для природної характеристики двигуна:

$$k_d \Phi_{\text{нд}} = \frac{U_{\text{нд}} - I_{\text{нд}} R_{\Sigma d}}{\omega_{\text{нд}}};$$

де $U_{\text{нд}}$, $I_{\text{нд}}$, $\omega_{\text{нд}}$ – відповідно номінальні значення напруги, струму та частоти обертання;

$R_{\Sigma d}$ – активний опір якірного кола двигуна

$$R_{\Sigma d} = k_t \cdot (R_{\text{яд}} + R_{\text{кд}} + R_{\text{дпд}}),$$

$R_{\text{яд}}$, $R_{\text{кд}}$, $R_{\text{дпд}}$ – опір обмотки якоря, компенсаційної і додаткових полюсів;

k_t – температурний коефіцієнт

$$R_{\Sigma d} = 1,36 \cdot (0,034 + 0 + 0,02) = 0,07344 \text{ Ом}$$

$$k_d \Phi_{\text{нд}} = \frac{220 - 192 \cdot 0,07344}{60,18} = 3,42$$

Сумарний активний опір якірного кола електроприводу:

$$R_{\Sigma} = k \cdot R_{\Sigma d} + R_{\text{етп}} + R_{\text{зр}},$$

де k – кількість двигунів у приводі;

$R_{\text{етп}}$ – еквівалентний внутрішній опір тиристорного перетворювача.

У перетворювачах серії ШУВ застосована трифазна реверсивна мостова схема, у якій анодний трансформатор відсутній. Для обмеження зміни струму використовується токообмежуючий реактор з індуктивністю $L_{\text{тор}}$.

Еквівалентний внутрішній опір перетворювача $R_{\text{е тп}}$ складається з приведенного активного опору реактора $R_{\text{тор}}$ та опору, що описує явище анодного перекриття:

$$R_{\text{е}} = 2mf_1L_{\text{тор}},$$

де m – число імпульсів схеми (для трифазного мосту $m=6$);

f_1 – частота мережі.

Хоча опір формально має індуктивний характер, зменшення середнього значення випрямленої напруги через комутаційне перекриття еквівалентне введенню активного опору $R_{\text{е}}$. Оскільки зазвичай справджується умова $R_{\text{е}} \gg R_{\text{тор}}$, то приймають $R_{\text{е тп}} = R_{\text{е}}$. Відтак :

$$R_{\text{е}} = 2 \cdot 6 \cdot 50 \cdot 0.174 \cdot 10^{-3} = 0,1044 \text{ Ом}.$$

Активний опір згладжуючого реактора позначено через $R_{\text{зр}}$. Необхідність застосування опір згладжуючого реактора визначається далі.

Загальне значення індуктивності силового кола

$$L_{\Sigma} = k L_{\text{я д}} + L_{\text{т}} + L_{\text{зр}},$$

де $L_{\text{я д}}$ – індуктивність якоря двигуна;

$$L_{\text{я д}} = \gamma \frac{U_{\text{н}}}{I_{\text{н}} \omega_{\text{н}} p},$$

γ – коефіцієнт: для компенсованих машин $\gamma = 0,25$, для некомпенсованих $\gamma = 0,6$;

p_d – число пар полюсів

$$L_{\text{яд}} = 0,6 \cdot \frac{220}{192 \cdot 60,2 \cdot 2} = 0,0057 \text{ Гн}.$$

L_T – еквівалентна індуктивність токообмежуючого реактора

$$L_T = \frac{m}{3} \cdot L_{\text{тор}},$$

$$L_T = \frac{6}{3} \cdot 0,174 \cdot 10^{-3} = 0,348 \cdot 10^{-3} \text{ Гн}.$$

$L_{\text{сп}}$ – індуктивність згладжуючого реактора.

Індуктивність згладжуючого реактора визначають із вимоги забезпечення допустимого рівня пульсацій. Його розрахункове значення

$$L_{\text{зр}} \geq \frac{\sqrt{2U_d}}{m\omega_1(0,02 - 0,15)I_H} - kL_{\text{яд}} - L_T,$$

де U_d – діюче значення основної гармоніки пульсацій, яке при $m = 6$ приймають рівним

$$U_d = 0,36 \cdot E_{d0},$$

E_{d0} – середнє значення випрямленої напруги при куті керування $\alpha = 0$; для мостової схеми випрямлення $E_{d0} = 2,34 U_\phi$,

U_ϕ – діюче значення фазної напруги

$$U_d = 0,36 \cdot 514 = 185 \text{ В} \quad E_{d0} = 2,34 \cdot 220 = 514 \text{ В}$$

ω_1 – кутова частота мережі

$$\omega_1 = 2\pi f,$$

$$\omega_1 = 2 \cdot 3,14 \cdot 50 = 314 \text{ c}^{-1}.$$

$$L_{\text{зр}} = \frac{\sqrt{2 \cdot 185}}{6 \cdot 314 \cdot 0,15 \cdot 192} - 2 \cdot 0,0057 - 0,000348 = -0,0015 \text{ Гн}.$$

Підставивши значення, отримали від'ємний результат, що означає відсутність потреби у встановленні згладжуючого реактора.

Розраховуємо результуючий опір і індуктивність силового кола:

$$R_{\Sigma} = 2 \cdot 0,07344 + 0,1044 + 0 = 0,25128 \text{ Ом},$$

$$L_{\Sigma} = 2 \cdot 0,0057 + 0,348 \cdot 10^{-3} + 0 = 0,01177 \text{ Гн}.$$

Статичні характеристики розімкненої системи.

Після підстановки чисельних значень рівняння (2.6)–(2.7) набувають вигляду:

$$\omega_d = \frac{E_{\text{тп}}}{2 \cdot 3,42} - I_{\text{я}} \frac{0,2512}{2 \cdot 3,42} = E_{\text{тп}} \cdot 0,146 - I_{\text{я}} \cdot 0,0367,$$

$$\omega_d = \frac{E_{\text{тп}}}{2 \cdot 3,42} - M_d \cdot \frac{0,2512}{2 \cdot 3,42^2} = E_{\text{тп}} \cdot 0,146 - M_d \cdot 0,01073.$$

Оскільки ці залежності є лінійними, для побудови характеристики достатньо знайти точки: холостого ходу та номінального навантаження. У якості максимального значення E_r прийнемо подвоєну номінальну напругу електродвигуна, тобто $E_{\text{тп1}} = E_{\text{тп max}} = k U_{\text{нд}} = 2 \cdot 220 = 440 \text{ В}$. Будуємо характеристики також при $E_{\text{тп2}} = 0,5 \cdot E_{\text{тп max}} = 0,5 \cdot 440 = 220 \text{ В}$ і $E_{\text{тп3}} = 0$ (режим динамічного гальмування).

1. Точка ідеального холостого ходу

$$I_{\text{я}} = 0, \quad M_d = 0,$$

$$\omega_{д01} = E_{тп1} \cdot 0,146 = 440 \cdot 0,146 = 64,24 \text{ } c^{-1},$$

$$\omega_{д02} = E_{тп2} \cdot 0,146 = 220 \cdot 0,146 = 32,12 \text{ } c^{-1},$$

$$\omega_{03} = 0.$$

2. Точка номінального навантаження

$$I_{я} = I_{нд} = 192 \text{ } A,$$

$$M = M_{н} = k_{д} \Phi_{нд} \cdot I_{нд} = 3,42 \cdot 192 = 656,6 \text{ } Hм,$$

$$\omega_{д1} = E_{г1} \cdot 0,146 - I_{я} \cdot 0,0367 = 220 \cdot 0,146 - 192 \cdot 0,0367 = 64,24 - 7,05 = 57,2 \text{ } c^{-1},$$

$$\omega_{д2} = E_{тп2} \cdot 0,146 - I_{я} \cdot 0,0367 = 220 \cdot 0,146 - 192 \cdot 0,0367 = 32,12 - 7,05 = 25,07 \text{ } c^{-1},$$

$$\omega_3 = E_{тп3} \cdot 0,146 - I_{я} \cdot 0,0367 = 0 \cdot 0,146 - 192 \cdot 0,0367 = 0 - 7,05 = -7,05 \text{ } c^{-1}.$$

Згідно розрахунків, на рис. 2.4 наведено статичні електромеханічні та механічні характеристики розімкненої системи (пунктиром). При $E'_{тп1} = -E_{тп \text{ max}}$ і $E'_{тп2} = -0,5 \cdot E_{тп \text{ max}}$ графіки симетричні відносно початку координат.

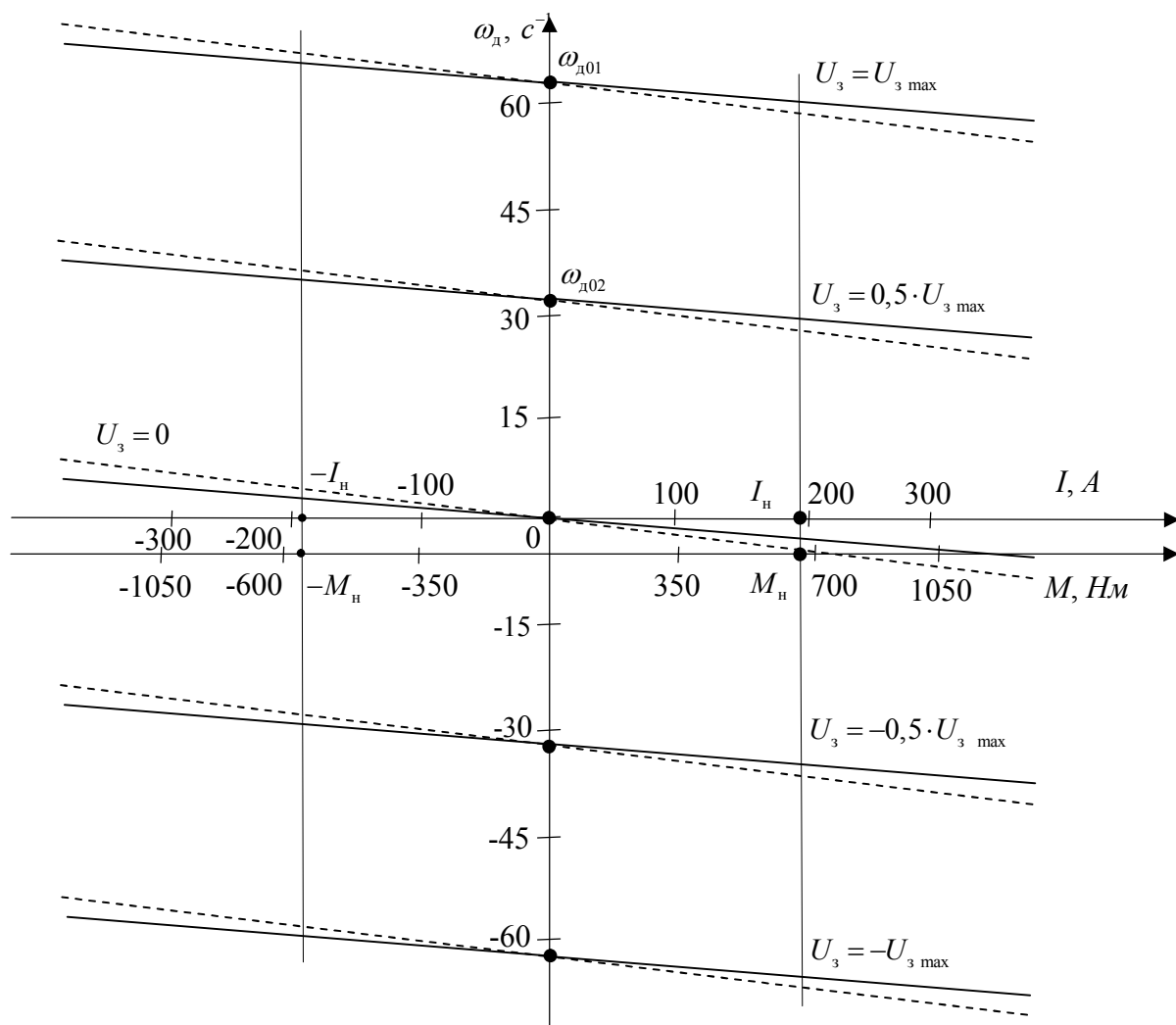


Рисунок 2.4 – Механічні і електромеханічні характеристики розімкненої і замкнутої системи

Формування характеристик у замкненій системі.

Під час роботи за схемою ТП–Д загальний опір кола збільшується, що приводить до зменшення жорсткості механічної характеристики. Тому для більшості роторних екскаваторів застосування замкнутого контуру регулювання є обов'язковим. Це дозволяє підвищити жорсткість характеристики та обмежити максимальний момент.

На рис. 2.5 наведено функціональну схему САУ з одним зворотним зв'язком по напрузі тиристорного перетворювача. Для наочності двигуни замінені одним еквівалентним.

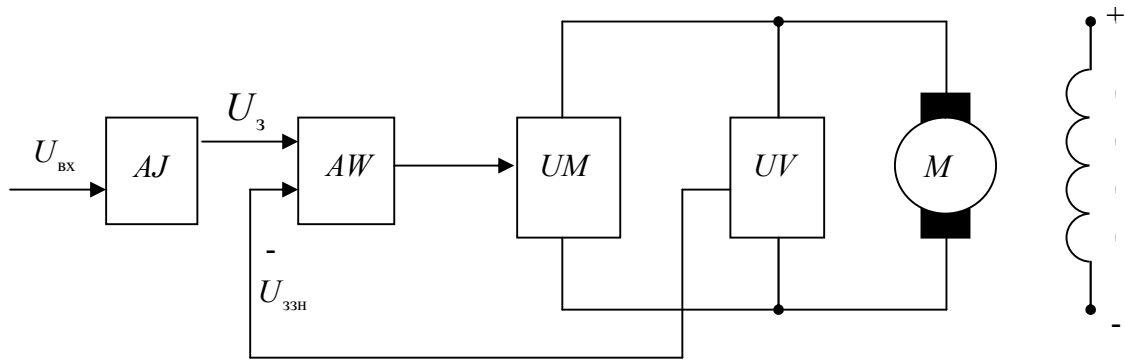


Рисунок 2.5 – Функціональна схема системи автоматичного керування

Рівняння статичної електромеханічної характеристики замкненої системи:

$$\omega_d = U_3 \frac{k_\Sigma / (k c)}{1 + k_\Sigma k_{33H}} - I_{\text{я}} \left(\frac{R_{\text{я}\Sigma}}{k c} - \frac{R_{\text{е пр}} k_\Sigma k_{33H} / (k c)}{1 + k_\Sigma k_{33H}} \right) = \omega_{d0} - \Delta \omega_{\text{ст1}},$$

де $k_\Sigma = k_{AW} k_{\text{ТП}}$ – сумарний коефіцієнт підсилення системи;

k_{AW} – коефіцієнт посилення суматора AW ;

$k_{\text{ТП}}$ – коефіцієнт підсилення ТП.

Коефіцієнт посилення суматора приймаємо $k_{AW} = 0,17$.

Для синусоїдального сигналу порівняння з амплітудою U_m коефіцієнт підсилення ТП $k_{\text{ТП}}$ визначається як

$$k_{\text{ТП}} = \frac{E_{d0}}{U_m}.$$

Приймаємо $U_m = 10 \text{ В}$, тоді

$$k_{\text{ТП}} = \frac{515}{10} = 51,4,$$

$$k_\Sigma = 0,17 \cdot 51,4 = 8,892.$$

c – коефіцієнт, визначається як

$$c = k_{\text{д}} \Phi_{\text{н,д}} = 3,42$$

$k_{\text{ззн}}$ – коефіцієнт зворотного зв'язку по напрузі підбирається виходячи з допустимого статичного перепаду швидкості $\Delta\omega_{\text{ст1}}$ при номінальному струмі:

$$\Delta\omega_{\text{ст1}} = I_{\text{н}} \left(\frac{R_{\text{я}\Sigma}}{k \ c} - \frac{R_{\text{е пр}} k_{\Sigma} k_{\text{ззн}} / (k \ c)}{1 + k_{\Sigma} k_{\text{ззн}}} \right),$$

звідки

$$k_{\text{ззн}} = \left(\frac{I_{\text{н}} R_{\text{я}\Sigma} / (k \ c) - \Delta\omega_{\text{ст1}}}{k_{\Sigma} \Delta\omega_{\text{ст1}} + I_{\text{н}} k_{\Sigma} \cdot R_{\text{е пр}} / (k \ c)} \right).$$

Величину $\Delta\omega_{\text{ст1}}$ приймаємо рівною половині статичного перепаду розімкненої системи.

Після розрахунку величини $k_{\text{ззн}}$ перевіряємо можливість реалізації такого значення фізичними елементами схеми.

$$\Delta\omega_{\text{ст}} = I_{\text{н}} \frac{R_{\text{я}\Sigma}}{k \ k_{\text{д}} \Phi_{\text{н}}},$$

$$\Delta\omega_{\text{ст}} = 192 \cdot \frac{0,25128}{2 \cdot 3,42} = 7,05,$$

$$\Delta\omega_{\text{ст1}} = \frac{1}{2} \cdot \Delta\omega_{\text{ст}} = \frac{1}{2} \cdot 7,05 = 3,53,$$

$$k_{\text{ззн}} = \frac{192 \cdot 0,2512 / (2 \cdot 3,42) - 3,53}{8,892 \cdot 3,53 + (192 \cdot 8,892 \cdot 0,1044) / (2 \cdot 3,42)} = 0,061.$$

Максимальне значення напруги завдання $U_{\text{з max}}$ визначаємо з умови отримання при $U_{\text{з max}}$ швидкості ідеального холостого ходу розімкненої системи при $E_{\text{тп}} = k \cdot U_{\text{н}}$

$$\omega_{д0} = U_{з \max} \frac{k_{\Sigma} / (k c)}{1 + k_{\Sigma} k_{33H}} = \frac{U_{н}}{c},$$

$$U_{з \max} = \frac{U_{н} (1 + k_{\Sigma} k_{33H})}{k_{\Sigma} / k},$$

$$U_{з \max} = \frac{220 \cdot (1 + 8,892 \cdot 0,061)}{8,892 / 2} = 76,44 \text{ В}.$$

Рівняння електромеханічної характеристики замкненої системи:

$$\omega_{д} = U_{з} \frac{8,892 / (2 \cdot 3,42)}{1 + 8,892 \cdot 0,061} - I_{я} \left(\frac{0,25128}{2 \cdot 3,42} - \frac{0,1044 \cdot 8,892 \cdot 0,061 / (2 \cdot 3,42)}{1 + 8,892 \cdot 0,061} \right) =$$

$$= U_{з} \cdot 0,84 - I_{я} \cdot 0,018.$$

Графіки будуються при $U_{з1} = U_{з \max} = 76,44 \text{ В}$,
 $U_{з2} = 0,5 \cdot U_{з \max} = 0,5 \cdot 76,44 = 38,22 \text{ В}$, а також при режимі гальмування $U_{з3} = 0$

Для побудови достатньо точок холостого ходу та номінального режиму.

1. Ідеальний холостий хід

$$I_{я} = 0, \quad M_{д} = 0.$$

$$\omega_{д01} = 76,44 \cdot 0,84 = 64,24 \text{ с}^{-1},$$

$$\omega_{д02} = 38,22 \cdot 0,84 = 32,12 \text{ с}^{-1},$$

2. Номінальний режим

$$I_{я} = 192 \text{ А}, \quad M_{д} = M_{н} = 656,6 \text{ Нм},$$

$$\omega_{д1} = \omega_{д01} = 76,44 \cdot 0,84 - 192 \cdot 0,018 = 61,21 \text{ с}^{-1},$$

$$\omega_{д2} = 38,22 \cdot 0,84 - 192 \cdot 0,018 = 29,07 \text{ с}^{-1}.$$

Результати наведені на рис. 2.4 суцільними лініями.

2.7 Дослідження динамічних характеристик системи

Для оцінювання динамічних властивостей розглянутої системи сформуємо алгоритмічну схему, зображену на рис. 2.6. Ця схема є еквівалентною функціональній структурі, наведеній на рис. 2.5.

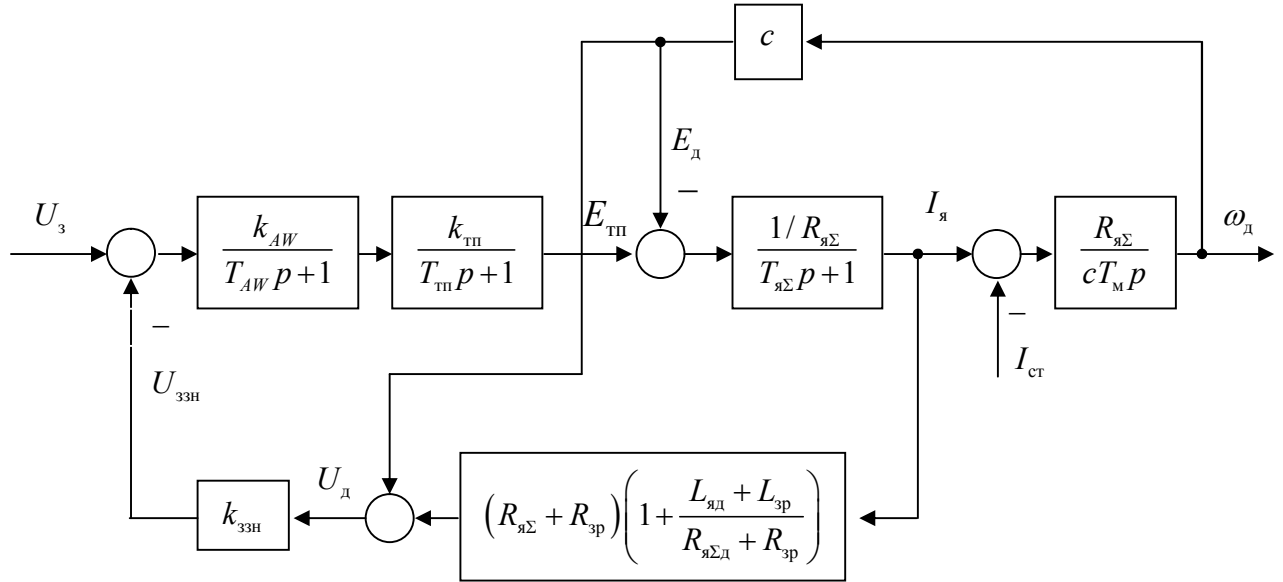


Рисунок 2.6 – Алгоритмічна схема системи керування електроприводом ТП-Д

За допомогою ряду еквівалентних перетворень зазначену схему можна подати у вигляді алгоритмічної структури, наведеної на рис. 2.7, яка відповідає системі типу Г-Д. На схемі використано такі умовні позначення:

$k_{33ш} = k_{33н} k c$ – коефіцієнт підсилення контуру зворотного зв'язку за швидкістю;

$T_{33г} = k_{33н} (k L_{яд} + L_{зр})$ – постійна часу гнучкого зворотного зв'язку по струму.

Вплив жорсткого струмового зворотного зв'язку, який виникає через присутність у сигналі $U_{33н}$ величини $k_{33н} (k R_{яΣд} + R_{зр}) I_{я}$, що дещо зменшує жорсткість механічної характеристики та слабо впливає на динаміку, допустимо не враховувати.

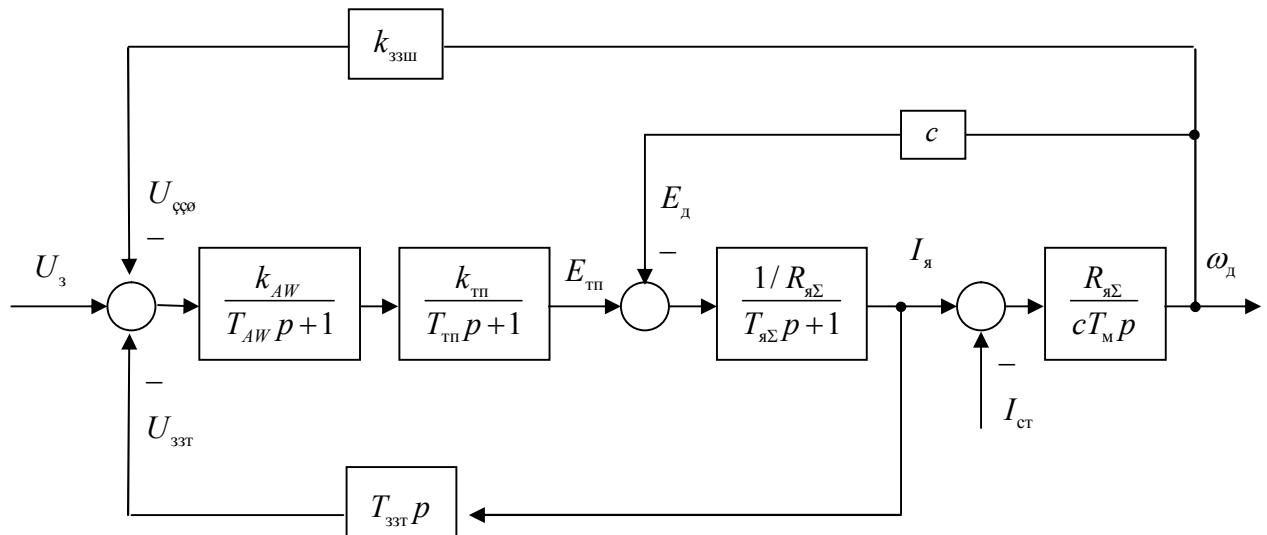


Рисунок 2.7 – Модифікована алгоритмічна схема системи керування

У порівнянні з системою Г-Д, динамічна поведінка приводу ТП-Д суттєво відрізняється, що обумовлено значно вищою швидкістю тиристорного перетворювача. Для зменшення динамічних навантажень у механічних вузлах екскаватора регулювання здійснюється через задатчик інтенсивності AJ .

У межах вибраної структури САУ можливості впливу на якість перехідних процесів обмежені, оскільки регулювати можна лише один зворотний зв'язок за напругою ТП, який одночасно змінює статичні та динамічні властивості приводу. Зазвичай приймають невеликі значення коефіцієнта $k_{33ш}$, що дещо зменшує жорсткість статичної характеристики, але покращує динаміку. У попередньому розділі статичні характеристики були визначені для $k_{33ш} = 0,061$.

Надалі визначимо динамічні характеристики системи для кількох значень $k_{33ш}$ і виберемо те, що забезпечує найкращі показники якості перехідних процесів.

Оскільки суматор AW реалізовано на операційних підсилювачах, його постійна часу T_{AW} , яка становить мікросекунди, може бути знехтувана. Тоді передавальні функції напівпровідникових суматорів AW та тиристорного перетворювача можна об'єднати в одну еквівалентну ланку з передавальною функцією

$$W_1(p) = \frac{k_{\Sigma}}{T_{\Sigma}p + 1},$$

де $k_{\Sigma} = k_{AW} \cdot k_{\Sigma}$

У результаті отримаємо алгоритмічну схему, представлену на рис. 2.8.

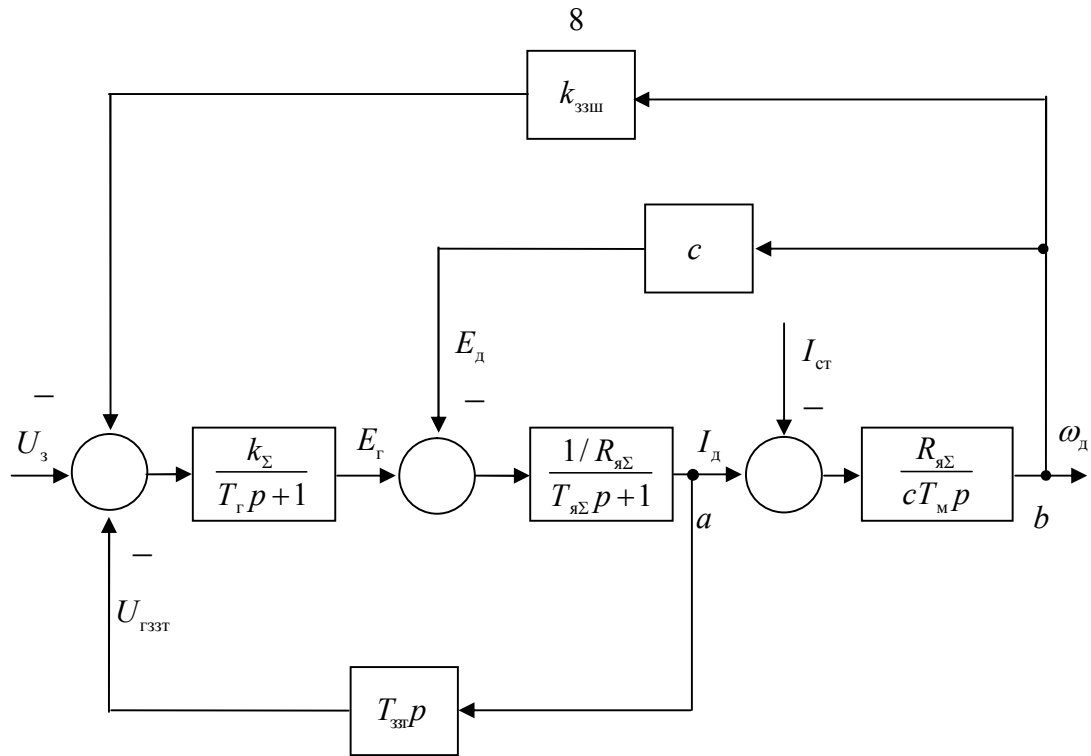


Рисунок 2.8 – Спрощена алгоритмічна структура САУ електроприводу Г-Д

Для подальшого аналізу визначимо передавальні функції системи за каналом задавання та збурення. Алгоритмічна схема для визначення передавальної функції швидкості по задаючому впливу $W_{\omega_d U_3}(p) = \frac{\omega_d(p)}{U_3(p)}$ ($I_{ст} = 0$) подана на рис. 2.9. На ній усі функціональні ланки пронумеровано від 1 до 6.

$$W_1(p) = \frac{k_{\Sigma}}{T_r p + 1}; \quad W_2(p) = \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1}; \quad W_3(p) = \frac{R_{я\Sigma}}{c T_m p};$$

$$W_4(p) = c = \frac{1}{k_d}; \quad W_5(p) = T_{3\Gamma} p; \quad W_6(p) = k_{33\Gamma}.$$

У передавальній функції $W_4(p)$ використано позначення $k_d = \frac{1}{c}$ – коефіцієнт підсилення двигуна.

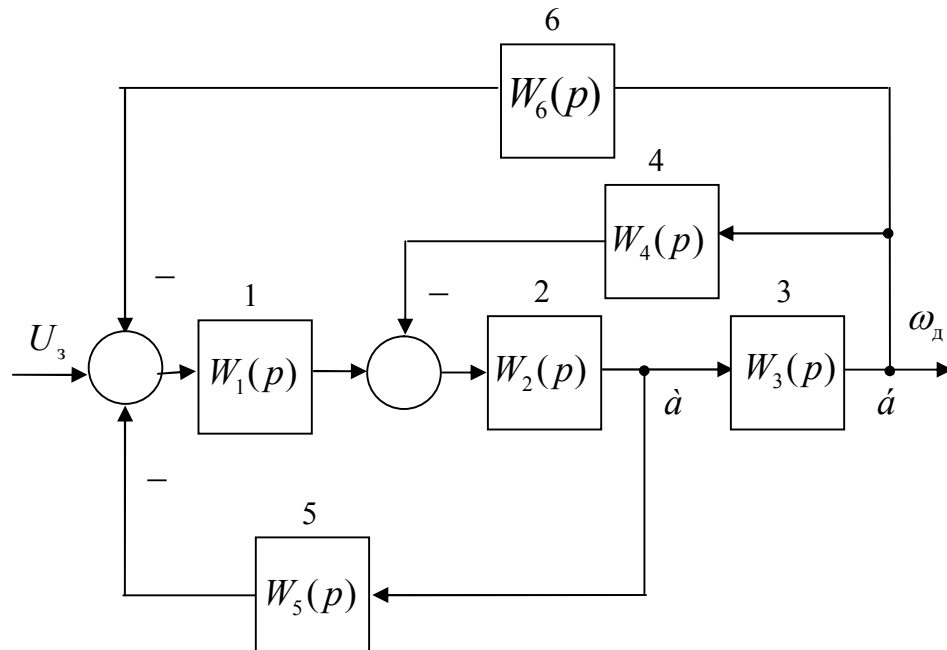


Рисунок 2.9 –Алгоритмічна схема для отримання передавальної функції швид-

$$\text{кості за задаючою дією } W_{\omega_d U_3}(p) = \frac{\omega_d(p)}{U_3(p)}$$

Перемістимо струмовий зворотний зв'язок із точки «а» у точку «б», увімкнувши в зворотний зв'язок ланку з передавальною функцією .

$$W_9(p) = \frac{1}{W_5(p)} = \frac{cT_{\Sigma}p}{R_{\Sigma}}.$$

Після цього схема набуває вигляду рис. 2.10, де відсутні перехрещення зворотних зв'язків.

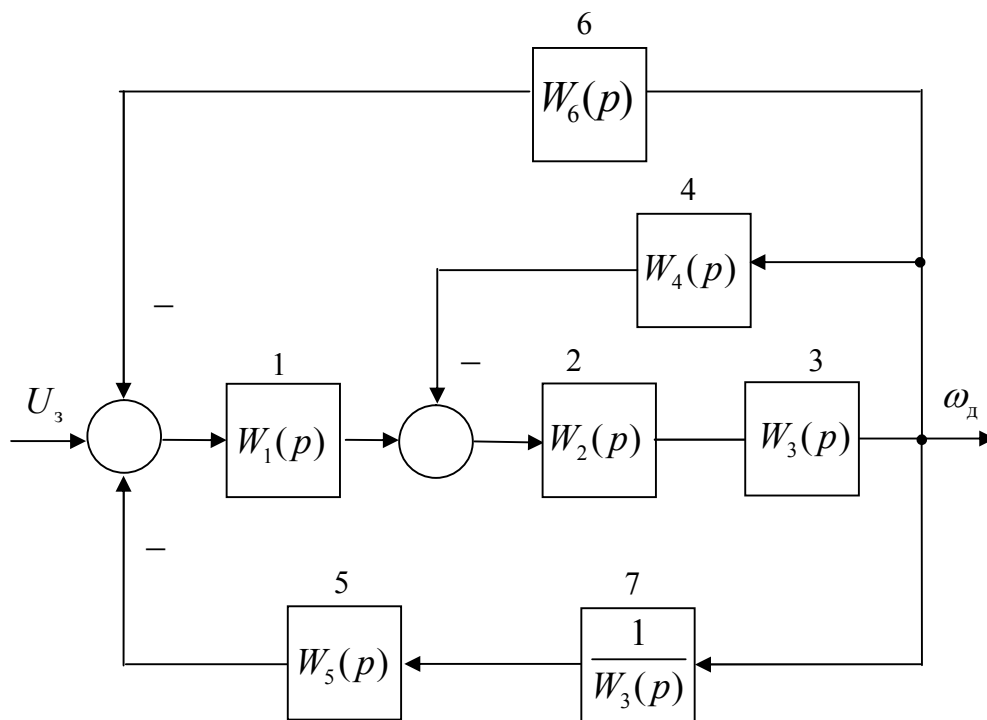


Рисунок 2.10 – Розрахункова схема для формування передавальної функції

швидкості за задаванням $W_{\omega U_3}(p) = \frac{\omega(p)}{U_3(p)}$

Об'єднаємо ланки 2, 3 і 4. Еквівалентна передавальна функція визначається за формулою:

$$W_{2,3,4}(p) = \frac{W_2(p)W_3(p)}{1 + W_2(p)W_3(p)W_4(p)}. \quad (2.8)$$

Підставивши в (2.8) відповідні передавальні функції, отримаємо:

$$W_{2,3,4}(p) = \frac{\frac{1/R_{\Sigma}}{T_{\Sigma}p + 1} \cdot \frac{R_{\Sigma}}{cT_{\Sigma}p}}{1 + \frac{c}{(T_{\Sigma}p + 1)cT_{\Sigma}p}} = \frac{k_d}{T_{\Sigma}T_{\Sigma}p^2 + T_{\Sigma}p + 1}.$$

Сумарна передавальна функція для ланок 5 і 6 дорівнює їх добутку:

$$W_{5,7}(p) = W_5(p) \cdot \frac{1}{W_3(p)} = T_{33T} p \cdot \frac{cT_M p}{R_{\Sigma}} = \frac{cT_M T_{33T} p^2}{R_{\Sigma}}.$$

Оскільки ланки $W_{5,7}(p)$ та $W_6(p)$ з'єднані паралельно, еквівалентна передавальна функція визначається сумою:

$$W_{5,6,7}(p) = W_{5,7}(p) + W_6(p) = \frac{cT_M T_{33T} p^2}{R_{\Sigma}} + k_{33\text{ш}}.$$

Ланки $W_1(p)$ та $W_{2,3,4}(p)$ увімкнені послідовно, тому:

$$\begin{aligned} W_{1,2,3,4}(p) &= W_1(p) \cdot W_{2,3,4}(p) = \\ &= \frac{k_{\Sigma}}{T_{\Gamma} p + 1} \cdot \frac{k_{\Delta}}{T_{\Sigma} T_M p^2 + T_M p + 1} = \frac{k_{\Sigma} k_{\Delta}}{(T_{\Gamma} p + 1)(T_{\Sigma} T_M p^2 + T_M p + 1)}. \end{aligned}$$

Передавальна функція швидкості за заданням:

$$\begin{aligned} W_{\omega_d U_3}(p) &= \frac{\omega_d(p)}{U_3(p)} = \frac{W_{1,2,3,4}(p)}{1 + W_{1,2,3,4}(p) \cdot W_{5,6,7}(p)} = \\ &= \frac{\frac{k_{\Sigma} k_{\Delta}}{(T_{\Gamma} p + 1)(T_{\Sigma} T_M p^2 + T_M p + 1)}}{1 + \frac{k_{\Sigma} k_{\Delta}}{(T_{\Gamma} p + 1)(T_{\Sigma} T_M p^2 + T_M p + 1)} \left(\frac{cT_M T_{33T} p^2}{R_{\Sigma}} + k_{33\text{ш}} \right)} = \\ &= \frac{k_{\Sigma} k_{\Delta}}{T_{\Gamma} T_M T_{\Sigma} p^3 + \left(T_{\Gamma} T_M + T_{\Sigma} T_M + \frac{k_{\Sigma} T_M T_{33T}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_M) p + 1 + k_{\Sigma} k_{\Delta} k_{33\text{ш}}}. \end{aligned}$$

Передавальну функцію струму по каналу задання $W(p) = \frac{I(p)}{U_3(p)}$ ($I_c = 0$)

отримаємо з перетвореної алгоритмічної схеми, наведеної на рис. 2.11.

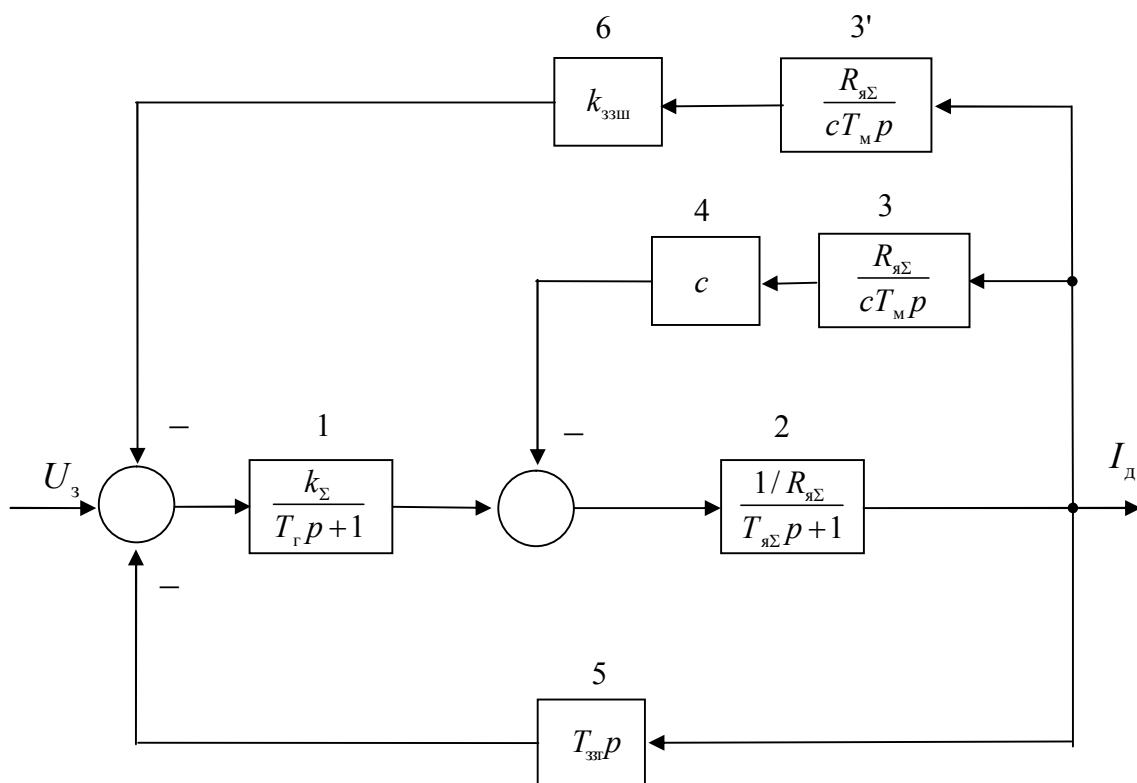


Рисунок 2.11 – Алгоритмічна схема для отримання передавальної функції

струму за задаючим сигналом $W_{I_d U_3}(p) = \frac{I_d(p)}{U_3(p)}$

Об'єднання ланок проводиться аналогічно попередньому випадку.

$$W_{2,3,4}(p) = \frac{W_2(p)}{1 + W_2(p) \cdot W_3(p) \cdot W_4(p)} =$$

$$\frac{\frac{1/R_{я\Sigma}}{T_{я\Sigma}p + 1}}{1 + \frac{1/R_{я\Sigma}}{T_{я\Sigma}p + 1} \cdot \frac{R_{я\Sigma}}{cT_{М}p} \cdot c} = \frac{T_{М}p / R_{я\Sigma}}{T_{я\Sigma}T_{М}p^2 + T_{М}p + 1}.$$

$$W_{3',6}(p) = W_3(p) \cdot W_6(p) = \frac{k_{33ш} R_{я\Sigma}}{cT_{М}p}.$$

$$W_{3',6,5}(p) = W_{3',6}(p) + W_5(p) = \frac{k_{33ш} R_{я\Sigma}}{cT_{М}p} + T_{3т}p.$$

$$\begin{aligned}
W_{1,2,3,4}(p) &= W_1(p) \cdot W_{2,3,4}(p) = \frac{k_{\Sigma}}{T_{\Gamma}p + 1} \cdot \frac{T_{\text{м}}p / R_{\text{я}\Sigma}}{T_{\text{я}\Sigma}T_{\text{м}}p^2 + T_{\text{м}}p + 1} = \\
&= \frac{k_{\Sigma}T_{\text{м}}p}{R_{\text{я}\Sigma}(T_{\Gamma}p + 1)(T_{\text{я}\Sigma}T_{\text{м}}p^2 + T_{\text{м}}p + 1)} \\
W_{I_{\text{д}}U_3}(p) &= \frac{I_{\text{д}}(p)}{U_3(p)} = \frac{W_{1,2,3,4}(p)}{1 + W_{1,2,3,4}(p) \cdot W_{3',6,5}(p)} = \\
&= \frac{\frac{k_{\Sigma}T_{\text{м}}p}{R_{\text{я}\Sigma}(T_{\Gamma}p + 1)(T_{\text{я}\Sigma}T_{\text{м}}p^2 + T_{\text{м}}p + 1)}}{1 + \frac{k_{\Sigma}T_{\text{м}}p}{R_{\text{я}\Sigma}(T_{\Gamma}p + 1)(T_{\text{я}\Sigma}T_{\text{м}}p^2 + T_{\text{м}}p + 1)} \cdot \left(\frac{k_{33\text{ш}}R_{\text{я}\Sigma}}{cT_{\text{м}}p} + T_{33\text{т}}p \right)} = \\
&= \frac{k_{\Sigma}T_{\text{м}}p / R_{\text{я}\Sigma}}{T_{\Gamma}T_{\text{м}}T_{\text{я}\Sigma}p^3 + \left(T_{\Gamma}T_{\text{м}} + T_{\text{я}\Sigma}T_{\text{м}} + \frac{k_{\Sigma}T_{\text{м}}T_{33\text{т}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}})p + 1 + k_{\Sigma}k_{\text{д}}k_{33\text{ш}}}.
\end{aligned}$$

Передавальну функцію швидкості за збурювальною дією

$W_{\omega_{\text{д}}I_{\text{ст}}}(p) = \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)}$ ($U_3 = 0$) знайдемо за схемою, поданою на рис. 2.12.

$$W_{4,7}(p) = \frac{c \cdot (T_{\Gamma}p + 1)}{k_{\Sigma}} = \frac{(T_{\Gamma}p + 1)}{k_{\Sigma}k_{\text{д}}}.$$

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

$$W_{1,2,5}(p) = \frac{W_1(p) \cdot W_2(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_5(p)} =$$

$$= \frac{\frac{k_{\Sigma}}{T_{\Gamma}p+1} \cdot \frac{1/R_{\Sigma}}{T_{\Sigma}p+1}}{1 + \frac{k_{\Sigma}}{T_{\Gamma}p+1} \cdot \frac{1/R_{\Sigma}}{T_{\Sigma}p+1} \cdot T_{33\Gamma}p} = \frac{k_{\Sigma}/R_{\Sigma}}{(T_{\Gamma}p+1)(T_{\Sigma}p+1) + \frac{k_{\Sigma}T_{33\Gamma}}{R_{\Sigma}}p}.$$

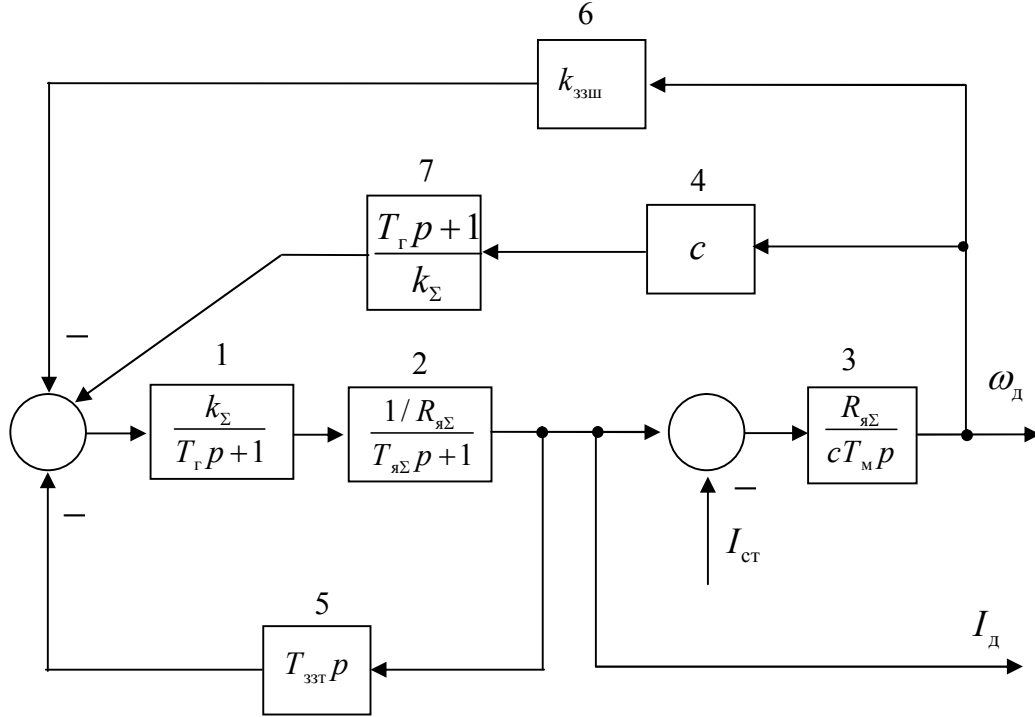


Рисунок 2.12 – Розрахункова схема для отримання передавальної функції швидкості за збуренням $W_{\omega_{\text{д}} I_{\text{ст}}}(p) = \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)}$

$$W_{\omega_{\text{д}} I_{\text{ст}}}(p) = \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)}$$

З урахуванням перетворень одержуємо схему, наведену на рис. 2.13.

$$W_{1,2,5,4,7,6}(p) = \frac{k_{\Sigma}/R_{\Sigma}}{(T_{\Gamma}p+1)(T_{\Sigma}p+1) + \frac{k_{\Sigma}T_{33\Gamma}}{R_{\Sigma}}p} \cdot \left(\frac{(T_{\Gamma}p+1)}{k_{\Sigma}k_{\text{д}}} + k_{33\Pi} \right) =$$

$$= \frac{T_{\Gamma}p+1 + k_{\text{д}}k_{\Sigma}k_{33\Pi}}{k_{\text{д}}R_{\Sigma} \left[T_{\Gamma}T_{\Sigma}p^2 + \left(T_{\Gamma} + T_{\Sigma} + \frac{k_{\Sigma}T_{33\Gamma}}{R_{\Sigma}} \right) p + 1 \right]}.$$

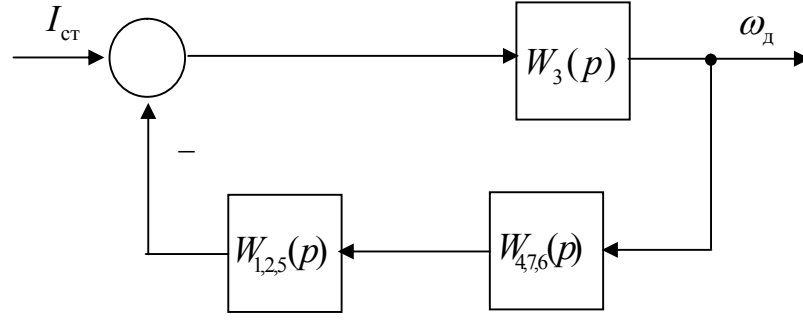


Рисунок 2.13 – Перетворена схема для визначення функції швидкості

$$\text{за збуренням } W_{\omega_d I_{\text{ст}}}(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)}.$$

Шукана передавальна функція:

$$\begin{aligned} W_{\omega_d I_{\text{ст}}}(p) &= \frac{\omega_d(p)}{I_{\text{ст}}(p)} = \frac{W_3(p)}{1 + W_3(p) \cdot W_{1,2,5,4,7,6}(p)} = \\ &= \frac{\frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p}}{1 + \frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot \frac{T_{\Gamma}p + 1 + k_{\text{д}}k_{\Sigma}k_{33\text{ш}}}{k_{\text{д}}R_{\text{я}\Sigma} \left[T_{\Gamma}T_{\text{я}\Sigma}p^2 + \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Sigma}T_{33\text{т}}}{R_{\text{я}\Sigma}} \right) p + 1 \right]}} = \\ &= \frac{\frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p}}{T_{\text{м}}T_{\Gamma}T_{\text{я}\Sigma}p^2 + \left(T_{\text{м}}T_{\Gamma} + T_{\text{м}}T_{\text{я}\Sigma} + \frac{k_{\Sigma}T_{33\text{т}}T_{\text{м}}}{R_{\text{я}\Sigma}} \right) \cdot p^2 + (T_{\text{м}} + T_{\Gamma})p + 1 + k_{\text{д}}k_{\Sigma}k_{33\text{ш}}} =; \\ &= \frac{[R_{\text{я}\Sigma}T_{\Gamma}T_{\text{я}\Sigma}p^2 + (R_{\text{я}\Sigma}T_{\Gamma} + R_{\text{я}\Sigma}T_{\text{я}\Sigma} + k_{\Sigma}T_{33\text{т}})p + R_{\text{я}\Sigma}]k_{\text{д}}}{T_{\Gamma}T_{\text{м}}T_{\text{я}\Sigma}p^3 + \left(T_{\Gamma}T_{\text{м}} + T_{\text{я}\Sigma}T_{\text{м}} + \frac{k_{\Sigma}T_{\text{м}}T_{33\text{т}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}})p + 1 + k_{\Sigma}k_{\text{д}}k_{33\text{ш}}}. \end{aligned}$$

Для знаходження передавальної функції струму щодо збурення

$W_{I_d I_{ct}}(p) = \frac{I_d(p)}{I_{ct}(p)}$ ($U_3 = 0$) скористаємося схемою на рис. 2.13, перетворивши її у вигляд рис. 2.14.

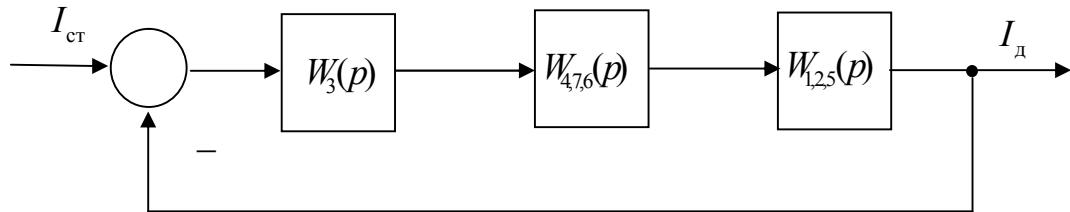


Рисунок 2.14 – Розрахункова схема для визначення передавальної

функції струму за збуренням $W_{I_d I_{ct}}(p) = \frac{I_d(p)}{I_{ct}(p)}$.

$$W_{5,4,7,6,1,2,5}(p) = W_3(p) \cdot W_{4,7,6}(p) \cdot W_{1,2,5}(p) =$$

$$\frac{R_{я\Sigma}}{cT_m p} \cdot \left(\frac{(T_\Gamma p + 1)}{k_\Sigma k_d} + k_{33ш} \right) \cdot \frac{k_\Sigma / R_{я\Sigma}}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1) + \frac{k_\Sigma T_{33т}}{R_{я\Sigma}} p} =$$

$$= \frac{T_\Gamma p + 1 + k_{33ш} k_\Sigma k_d}{T_m p (T_\Gamma p + 1) \cdot (T_{я\Sigma} \cdot p + 1) + \frac{k_\Sigma T_{33т} T_m}{R_{я\Sigma}} p^2}.$$

У результаті:

$$\begin{aligned} W_{I_d I_{ct}}(p) &= \frac{I_d(p)}{I_{ct}(p)} = \frac{W_{3,4,7,6,1,2,5}(p)}{1 + W_{3,4,7,6,1,2,5}(p)} = \\ &= \frac{\frac{T_\Gamma p + 1 + k_{33ш} k_\Sigma k_d}{T_m p (T_\Gamma p + 1) \cdot (T_{я\Sigma} \cdot p + 1) + \frac{k_\Sigma T_{33т} T_m}{R_{я\Sigma}} p^2}}{1 + \frac{T_\Gamma p + 1 + k_{33ш} k_\Sigma k_d}{T_m p (T_\Gamma p + 1) \cdot (T_{я\Sigma} \cdot p + 1) + \frac{k_\Sigma T_{33т} T_m}{R_{я\Sigma}} p^2}} = \end{aligned}$$

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

Далі знайдемо числові значення величин, що входять до передавальних функцій системи.

Електромагнітна постійна часу якоря:

$$T_{\text{я}} = \frac{L_{\text{я}\Sigma}}{R_{\text{я}\Sigma}},$$

$$T_{\text{я}} = \frac{0,01177}{0,25128} = 0,047 \text{ с}$$

Коефіцієнта зворотного зв'язку за швидкістю

$$k_{\text{ззш}} = k_{\text{ззн}} k_{\text{с}},$$

$$k_{\text{ззш}} = 0,061 \cdot 2 \cdot 3,42 = 0,417$$

Постійна часу гнучкого струмового зворотного зв'язку:

$$T_{\text{ззт}} = k_{\text{ззн}} (k \cdot L_{\text{яд}} + L_{\text{зр}}),$$

$$T_{\text{ззт}} = 0,061 \cdot (2 \cdot 5,7 \cdot 10^{-3} + 0) = 0,66 \cdot 10^{-3} \text{ с}.$$

Електромеханічна постійна часу

$$T_{\text{м}} = \frac{J_{\Sigma} R_{\text{я}\Sigma}}{(2c)^2},$$

де J_{Σ} – сумарний момент інерції якоря двигуна та приведений до вала момент інерції механізмів

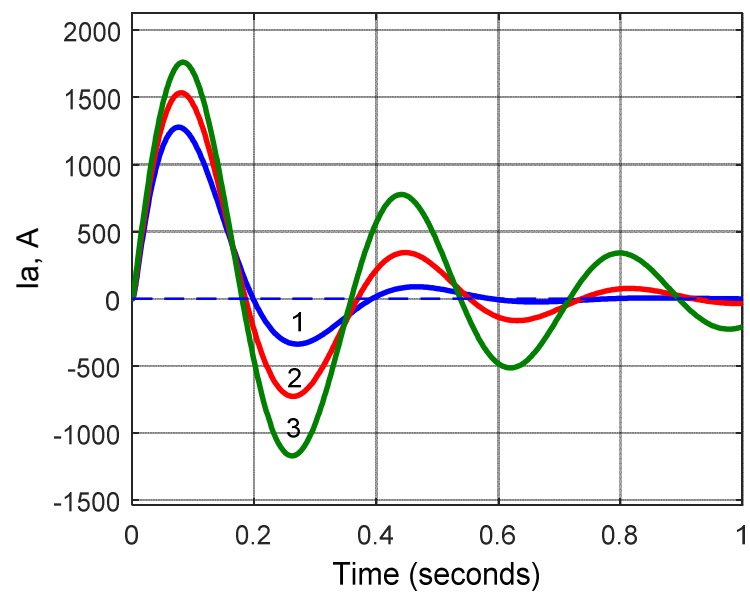
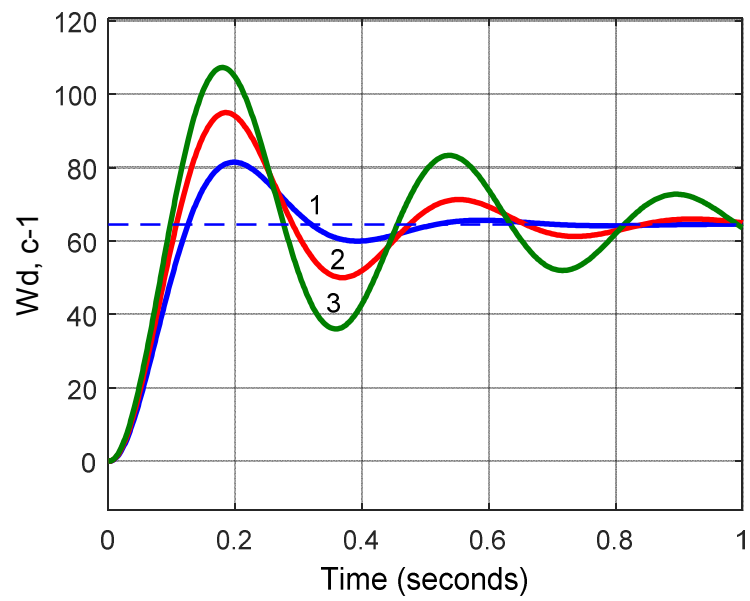
$$J_{\Sigma} = 2J_{\text{д}} + J_{\text{пл}} / i_{\text{п}}^2;$$

$$J_{\Sigma} = 2 \cdot 2 + 88 \cdot 10^7 / 10035^2 = 4 + 8,74 = 12,74 \text{ кг} \cdot \text{м}^2,$$

$$T_{\text{м}} = 12,74 \cdot \frac{0,25128}{(2 \cdot 3,42)^2} = 0,21 \text{ с}.$$

За отриманими передавальними функціями $W_1(p) \div W_4(p)$ розраховуються перехідні процеси системи. Обчислення виконано засобами MATLAB; графіки наведені на рис. 2.16–2.17. Додатково результати підтверджені моделюванням у середовищі SIMULINK (рис. 2.15).

Розрахунок проведено для трьох значень коефіцієнта зворотного зв'язку $k_{\text{ззп}}$. На рис. 2.16–2.17 показано перехідні процеси відповідно при: 1 – $k_{\text{ззп}} = 0,061$; 2 – $k_{\text{ззп}} = 0,183$; 3 – $k_{\text{ззп}} = 0,427$. Як видно з графіків, найкраща якість перехідних процесів досягається при $k_{\text{ззп}} = 0,061$, хоча при цьому статична похибка становить $4,5 \text{ с}^{-1}$.



а)

б)

Рисунок 2.16 – Перехідні характеристики швидкості (а) та струму (б)
при дії задавального сигналу

1 – $k_{3ш} = 0,061$; 2 – $k_{3ш} = 0,183$; 3 – $k_{3ш} = 0,427$.

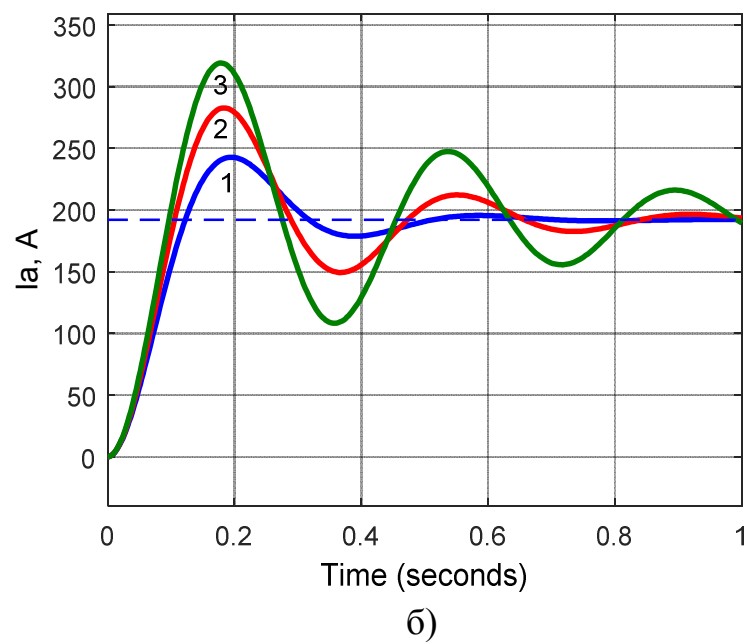
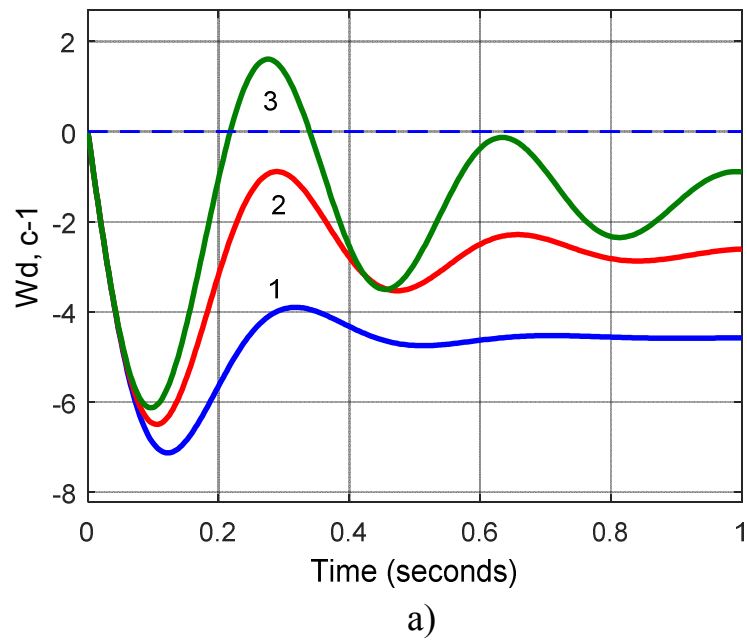


Рисунок 2.17 – Перехідні характеристики швидкості (а) та струму (б)
по збурюючій дії

$$1 - k_{33H} = 0,061; \quad 2 - k_{33H} = 0,183; \quad 3 - k_{33H} = 0,427.$$

ВИСНОВКИ ДО РОЗДІЛУ 2

1. Проведено вибір електродвигуна та основного силового обладнання електроприводу. Виконано перевірку двигуна за критеріями допустимого нагріву та перевантажувальної здатності шляхом побудови швидкісної та навантажувальної діаграм. Отримане значення еквівалентного моменту підтвердило, що двигун відповідає умовам експлуатації як за нагрівом, так і за допустимими перевантаженнями.

2. Розраховано параметри силового кола електроприводу, які необхідні для подальшого проектування та аналізу системи автоматичного керування.

3. Обґрунтовано та вибрано структуру системи автоматичного керування. Оскільки електроприводи основних механізмів роторних екскаваторів повинні забезпечувати високу жорсткість механічної характеристики та обмеження максимального моменту, прийнята система включає зворотний зв'язок за швидкістю, гнучкий струмовий зворотний зв'язок та струмовий зворотний зв'язок з відсіченням.

4. Виконано розрахунок статичних характеристик як розімкненої, так і замкненої системи. Складено алгоритмічну схему САУ та визначено передавальні функції системи для задавальних і збурюючих впливів.

5. Проведено моделювання синтезованої системи на ЕОМ із застосуванням пакету MATLAB. На основі моделі, реалізованої у середовищі SIMULINK, виконано дослідження динамічних характеристик. За результатами моделювання визначено оптимальне значення постійної часу струмового зворотного зв'язку, при якому перехідні процеси відповідають вимогам до якості регулювання.

РОЗДІЛ 3

ХАРАКТЕРИСТИКИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

Для електроприводу, який забезпечує розворот екскаватора, типовою є наявність декількох масивних елементів, пов'язаних із поворотною платформою через пружні механічні з'єднання. До таких елементів належать маса стріли з установленим на ній ротором і робочим колесом, а також маса протиаги, розміщена на консолі. Усі ці частини системи мають власні моменти інерції та взаємодіють між собою не жорстко, а через еластичні ланки. Саме тому при зміні режимів роботи електроприводу виникають суттєві коливальні процеси, що помітно впливають на якість перехідних режимів.

У попередніх розділах аналіз функціонування електроприводу поворотного механізму здійснювався на основі припущення, що всі кінематичні зв'язки між елементами механізму характеризуються нескінченною жорсткістю. Тобто механічну частину системи було замінено спрощеною моделлю – єдиною приведеною масою, яка в середньому відображає рух всієї сукупності елементів. Однак така ідеалізація ігнорує реальну еластичність конструкції. На практиці жорсткість з'єднувальних елементів завжди є кінцевою, що неминуче робить механічну частину електроприводу пружною системою.

Унаслідок цього будь-яка зміна прикладеного моменту – як керуючої дії від двигуна, так і зовнішньої дії у вигляді статичного моменту навантаження – викликає взаємні коливання мас, пов'язаних пружними ланками. Такі коливання здатні суттєво збільшувати пікові навантаження у вузлах механізму, обмежувати точність реалізації необхідного руху та ускладнювати роботу системи в перехідних режимах.

Кінематична структура механізму розвороту має складний, розгалужений характер: вона містить багато зосереджених та розподілених мас, які взаємоді-

ють через пружні та часто невагомі з'єднувальні елементи. Для полегшення інженерних розрахунків такі багатоланкові системи зазвичай приводять до еквівалентних моделей у вигляді тримасових або двомасових структур. У таких моделях кожна маса розглядається як зосереджена, а зв'язки між ними – як пружні безмасові елементи з певними характеристиками жорсткості та демпфування.

Механізм розвороту екскаватора приводиться в дію двома електродвигунами, які працюють синхронно, формуючи багаторухову систему електроприводу. Повна розрахункова схема такої механічної структури, що враховує внесок кожного двигуна та особливості їх з'єднання з механізмом, наведена на рис. 3.1. Саме ця схема використовується як вихідна для подальшого приведення моделі до двомасового еквіваленту, що дозволяє точніше відобразити реальні коливальні властивості системи.

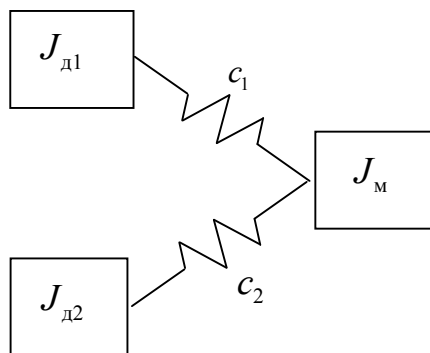


Рисунок 3.1 – Розрахункова розгалужена модель механічної частини електроприводу

На схемі використано такі позначення:

$J_{д1}$ та $J_{д2}$ – моменти інерції окремих двигунів;

c_1 та c_2 – приведені жорсткості пружних зв'язків кожного приводу;

$J_{м}$ – загальний приведений момент інерції механізму, тобто інерція поворотної платформи, зведена до вала двигуна: $J_{м} = J'_{пл}$;

Оскільки обидва двигуни функціонують синфазно, розгалужену систему (рис. 3.1) можна еквівалентно замінити на простішу двомасову модель, наведено на рис. 3.2.

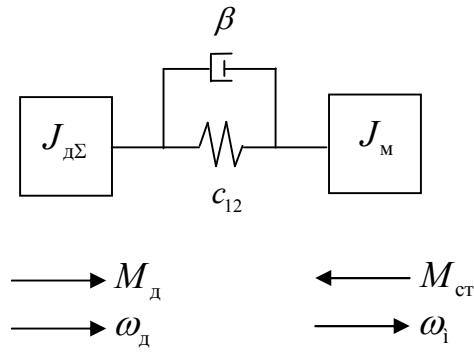


Рисунок 3.2 – Еквівалентна двомасова модель
механічної частини приводу

$J_{дΣ}$ – сумарний момент інерції двох двигунів; при використанні двох однакових машин: $J_{дΣ} = k \cdot J_{д} = 2 \cdot 7 = 14 \text{ кг} \cdot \text{м}^2$;

c_{12} – еквівалентна жорсткість зв'язку між масами, яка визначається як $c_{12} = c_1 + c_2$;

β – коефіцієнт в'язкого внутрішнього тертя вгнучкій ланці.

Після приведення інерцій та жорсткостей до кутової швидкості валу двигуна отримуємо систему диференціальних рівнянь руху двомасової моделі:

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

де M_{Σ} - сумарний момент у пружному зв'язку, який визначається як сума пружного моменту $M_{пр}$ та моменту, обумовленого в'язким тертям $M_{вт} = \beta(\omega_{д} - \omega_{м})$:

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

$\omega_{\text{д}}$, $\omega_{\text{м}}$ – кутові швидкості двигуна та механізму відповідно;

$M_{\text{ст}}$ – момент статичного навантаження на другій масі.

Перепишемо систему (3.1) з урахуванням виразу (3.2) у формі Коші:

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

Диференціальні рівняння електромеханічної частини одномасові системи, що розглядалась у попередньому розділі, записуються у вигляді:

$$\begin{cases} T_{\text{м}} \frac{d\omega_{\text{д}}}{dt} = \frac{R_{\text{я}\Sigma}}{c} (I_{\text{я}} - I_{\text{ст}}); \\ T_{\text{я}\Sigma} \frac{dI_{\text{я}}}{dt} + I_{\text{я}} = \frac{1}{R_{\text{я}\Sigma}} (E_{\text{тп}} - c\omega_{\text{д}}); \\ T_{\text{тп}} \frac{dE_{\text{тп}}}{dt} + E_{\text{тп}} = k_{\Sigma} U_{\text{з}} - k_{\text{ззш}} k_{\Sigma} \omega_{\text{д}} - k_{\Sigma} U_{\text{гззт}}; \\ T_{\text{ззт}} \frac{dI_{\text{я}}}{dt} = U_{\text{гззт}}. \end{cases} \quad (3.4)$$

Рівняння (3.4) записуємо у формі Коші, враховуючи співвідношення

$$I_{\text{ст}} = M_{\text{ст}} / c, \quad T_{\text{м}} = J_{\text{л}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{л}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}.$$

Після об'єднання систем (3.3) і (3.4) і проведення деяких перетворень, отримуємо повну систему рівнянь стану двомасового приводу двомасової системи:

$$\left\{ \begin{aligned} \frac{d\omega_{\text{д}}}{dt} &= +\frac{\beta}{J_{\text{д}\Sigma}}\omega_{\text{м}} - \frac{1}{J_{\text{д}\Sigma}}M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}}\omega_{\text{д}} + \frac{c}{J_{\text{д}\Sigma}}I_{\text{я}}; \\ \frac{dM_{\text{пр}}}{dt} &= -c_{12}\omega_{\text{м}} + c_{12}\omega_{\text{д}}; \\ \frac{d\omega_{\text{м}}}{dt} &= -\frac{\beta}{J_{\text{м}}}\omega_{\text{м}} + \frac{1}{J_{\text{м}}}M_{\text{пр}} + \frac{\beta}{J_{\text{м}}}\omega_{\text{д}} - \frac{1}{J_{\text{м}}}M_{\text{ст}}; \\ \frac{dI_{\text{я}}}{dt} &= -\frac{c}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}}\omega_{\text{д}} - \frac{1}{T_{\text{я}\Sigma}}I_{\text{я}} + \frac{1}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}}E_{\text{тп}}; \\ \frac{dE_{\text{тп}}}{dt} &= \left(\frac{ck_{\Sigma}T_{\text{ззт}}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\text{тп}}} - \frac{k_{\text{ззш}}k_{\Sigma}}{T_{\text{тп}}} \right) \omega_{\text{д}} + \frac{k_{\Sigma}T_{\text{ззт}}}{T_{\text{я}\Sigma}T_{\text{тп}}}I_{\text{я}} + \left(-\frac{k_{\Sigma}T_{\text{ззт}}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\text{тп}}} - \frac{1}{T_{\text{тп}}} \right) E_{\text{т}} + \frac{k_{\Sigma}}{T_{\text{тп}}}U_{\text{з}}. \end{aligned} \right. \quad (3.5)$$

У векторно-матричному вигляді система описується як:

$$\left\{ \begin{aligned} \dot{\vec{X}}(t) &= \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) &= \mathbf{C}\vec{X}(t), \end{aligned} \right.$$

При запису системи у векторно-матричному вигляді прийнято наступні позначення:

вектор стану системи:

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

вектор зовнішніх дій:

$$\vec{U}(t) = [U_{\text{з}}(t); M_{\text{ст}}(t)]^T.$$

A – матриця стану; **B** – матриця керування;

C – матриця виходу; $\vec{Y}(t)$ – вектор виходу.

Структура матриць **A** та **B** має вигляд:

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

Алгоритмічна блок-схема двомасової моделі подана на рис. 3.3.

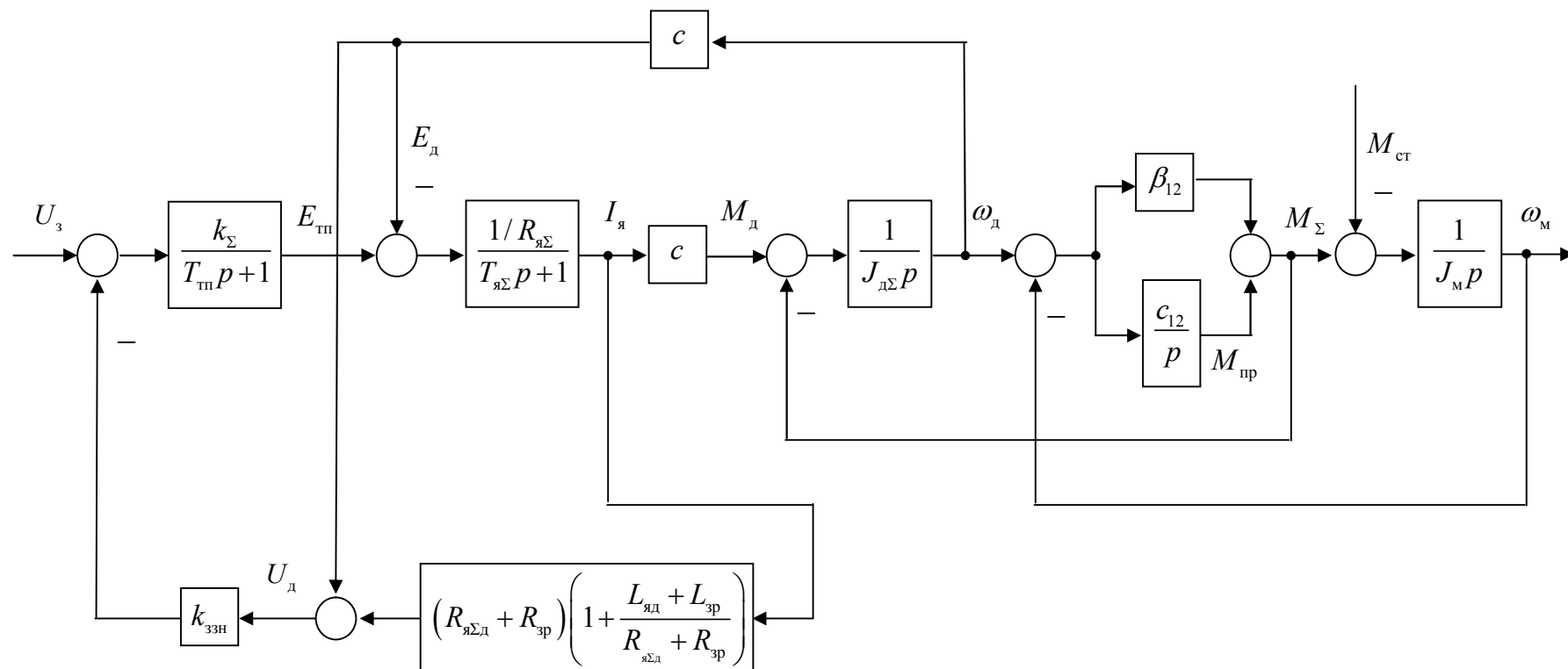


Рисунок 3.3 – Алгоритмічна блок-схема двомасової системи.

3.2 Розрахунок перехідних процесів двомасової системи

3.2 Розрахунок перехідних процесів двомасової системи

Для аналізу динаміки системи застосовувалась математична модель у формі рівнянь стану. На основі цієї моделі були виконані обчислення перехідних процесів як для задаючих, так і для збурюючих впливів. Усі розрахунки здійснено на ПЕОМ із використанням середовища MATLAB.

Початкові дані, а також матриці **A**, **B**, **C** наведено у вигляді текстового файлу (див. додаток Д). Для моделювання застосовувалась схема SIMULINK із блоком State-Space, яка показана на рис. 3.4. У вікні блоку задаються назви матриць двомасової моделі.

Кількість вихідних змінних (тобто розмірність вектора $\vec{Y}(t)$) задає число вікон **Scope**, у яких відображаються перехідні процеси. Вікно введення матриць **A**, **B**, **C** і **D** показано на рис. 3.5 – воно відкривається подвійним клацанням по блоку State-Space.

Якщо необхідно побудувати графіки всіх координат стану, матриця **C** береться одиничною. Якщо ж досліджуються лише окремі змінні, одиниці розміщують лише в потрібних рядках її головної діагоналі. Матриця **D** у даній моделі є нульовою.

Альтернативний варіант моделювання виконано за допомогою схеми, зображеної на рис. 3.6.

Графіки перехідних процесів наведено на рис. 3.7–3.10. На них показано поведінку перших чотирьох координат стану. Із аналізу видно, що система демонструє виражені коливання, які негативно впливають на функціонування механізму, підвищуючи навантаження на механічні елементи під час перехідних режимів.

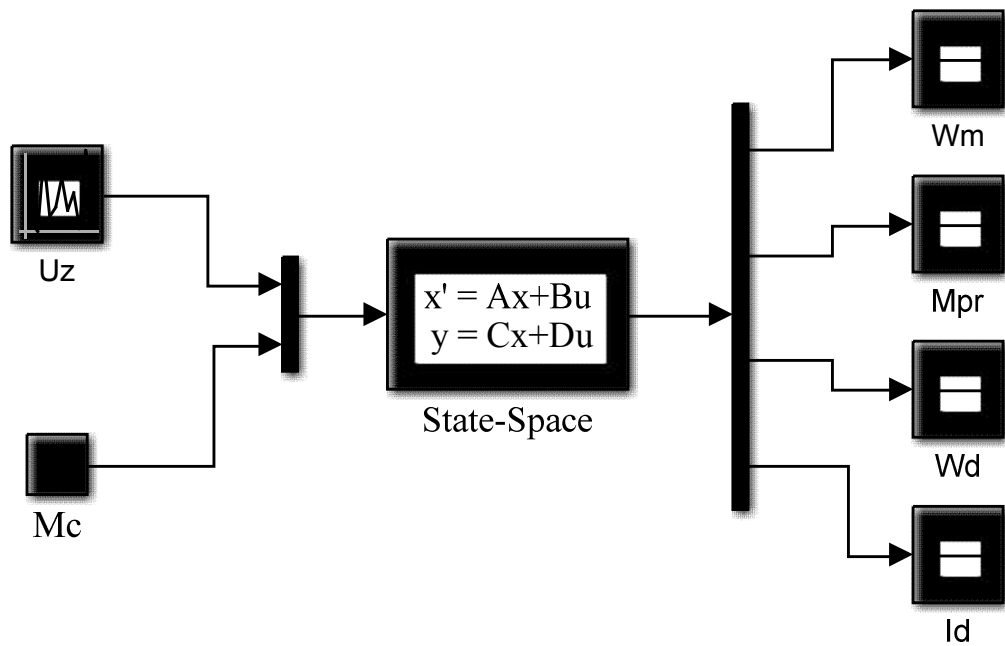


Рисунок 3.4 – Модель системи в середовищі SIMULINK з використанням State-Space-блоку

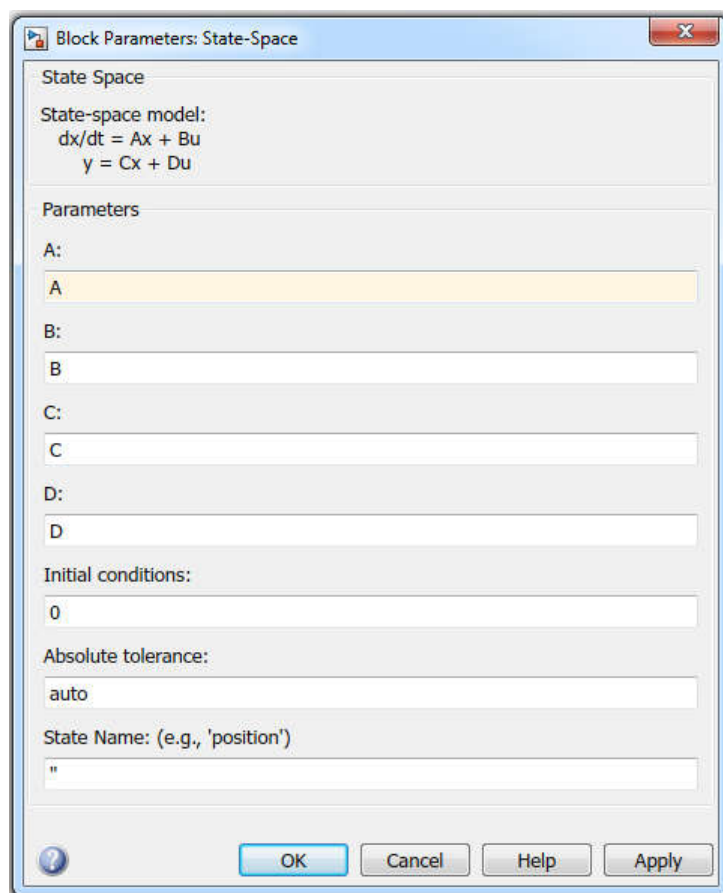


Рисунок 3.5 – Діалогове вікно введення матриць **A**, **B**, **C**, **D** у блоці State-Space

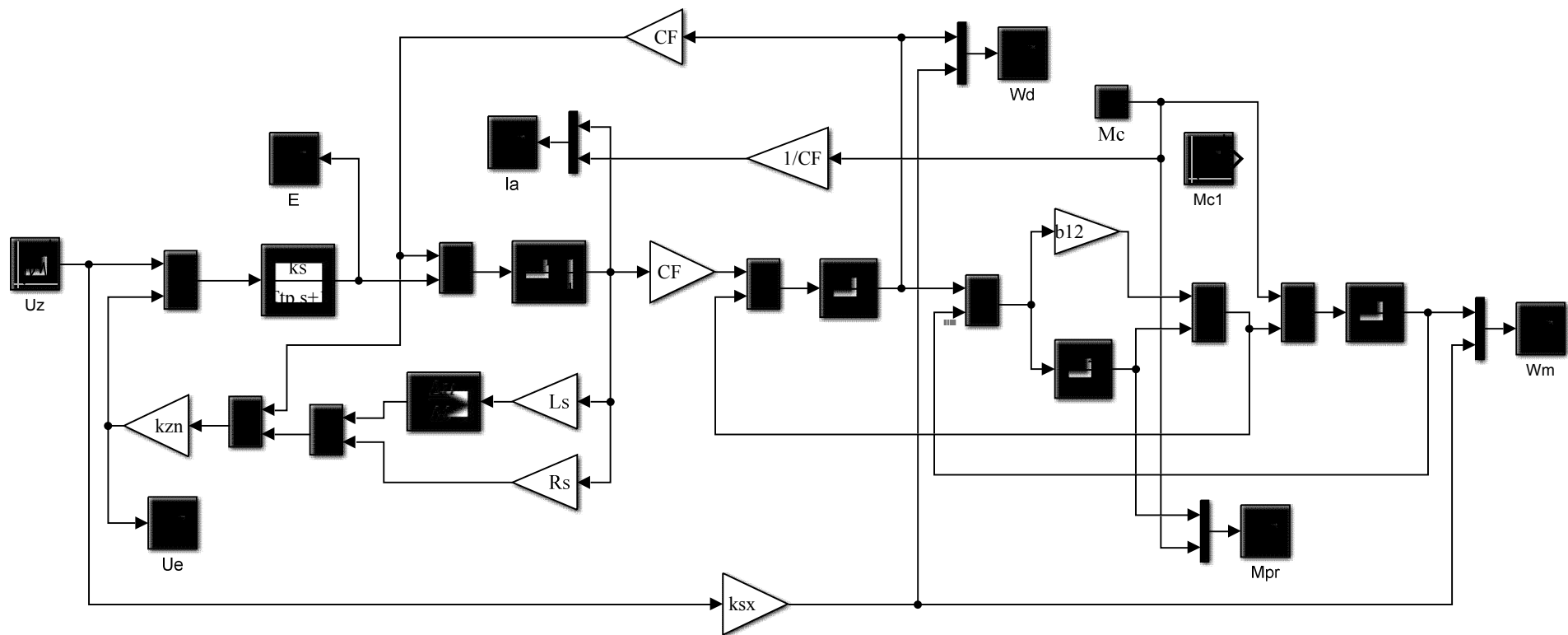


Рисунок 3.6 – Комп'ютерна модель двомасової системи, реалізована на ПЕОМ

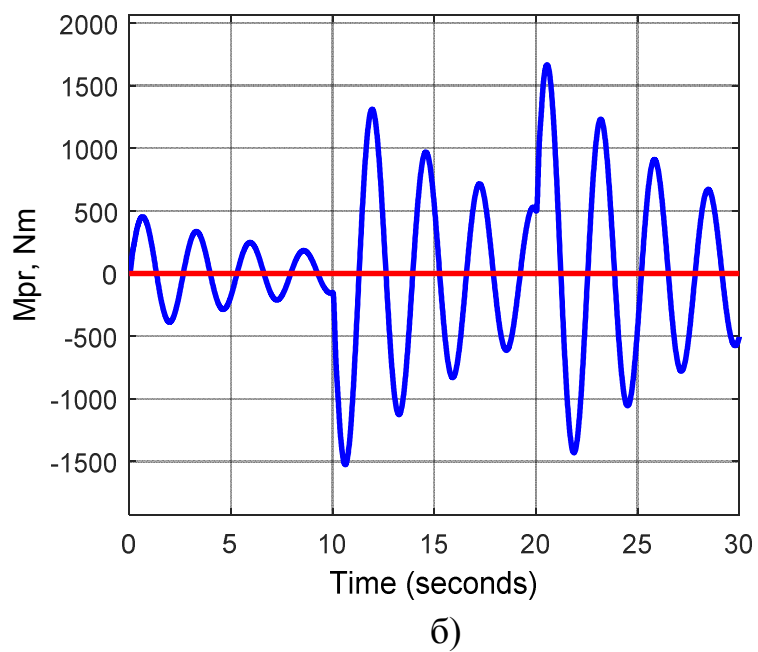
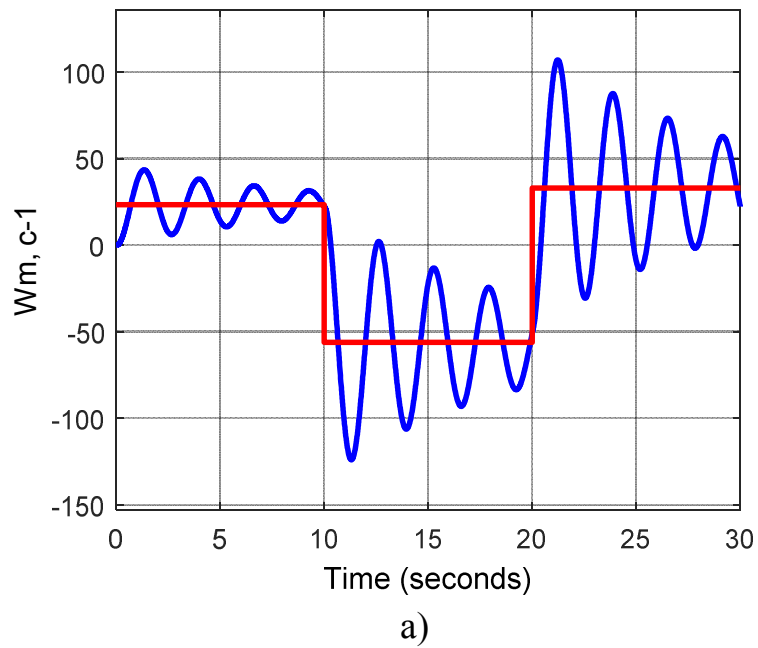


Рисунок 3.7 – Реакція двомасової системи на задаючий вплив: графік швидкості механізму $\omega_m = f(t)$ (а) та пружного моменту $M_{np} = f(t)$ (б)

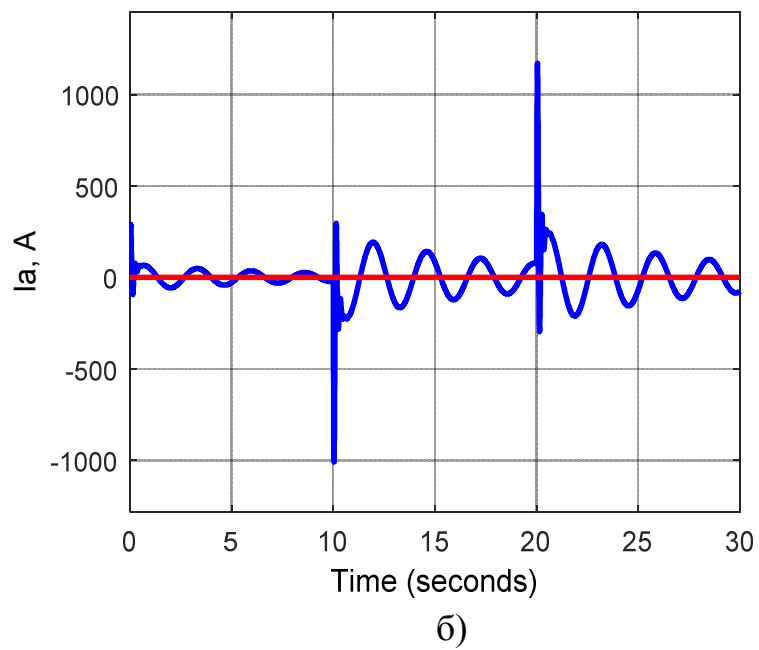
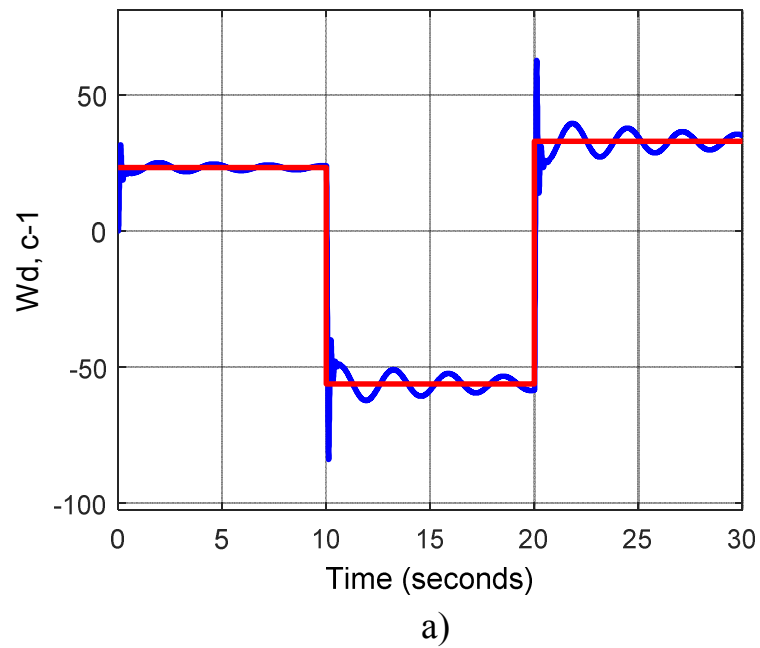


Рисунок 3.8 – Реакція двомасової системи на задаючий вплив: графік швидкості двигуна $\omega_d = f(t)$ (а) та струму двигуна $I_d = f(t)$ (б) двомасової системи по задаючій дії

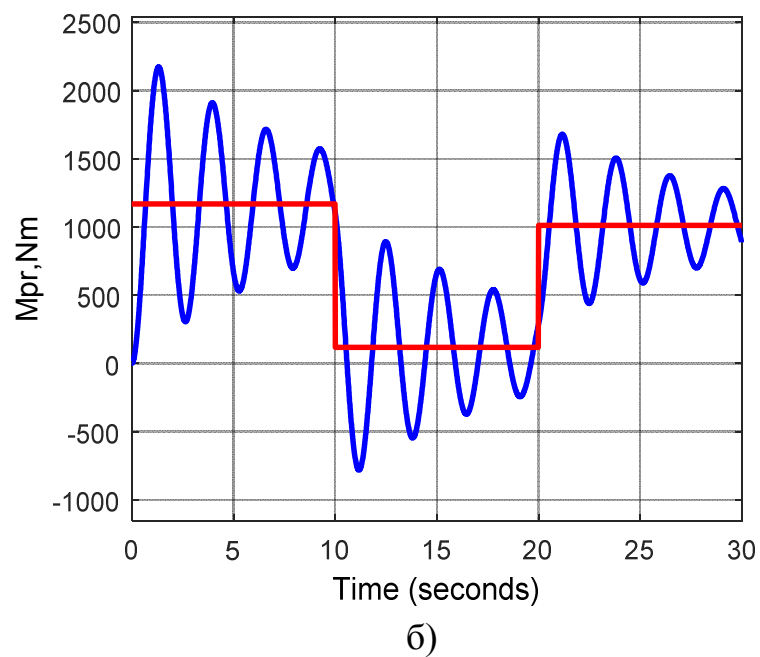
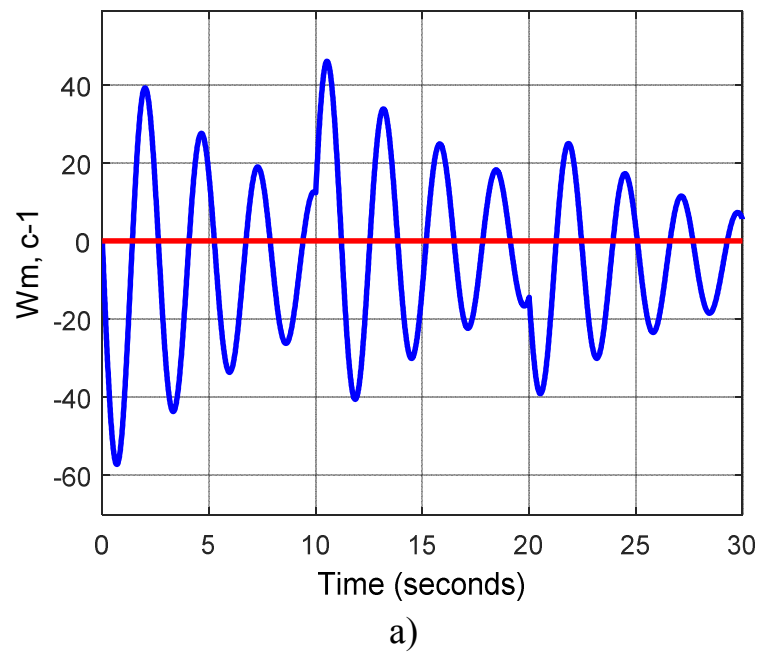
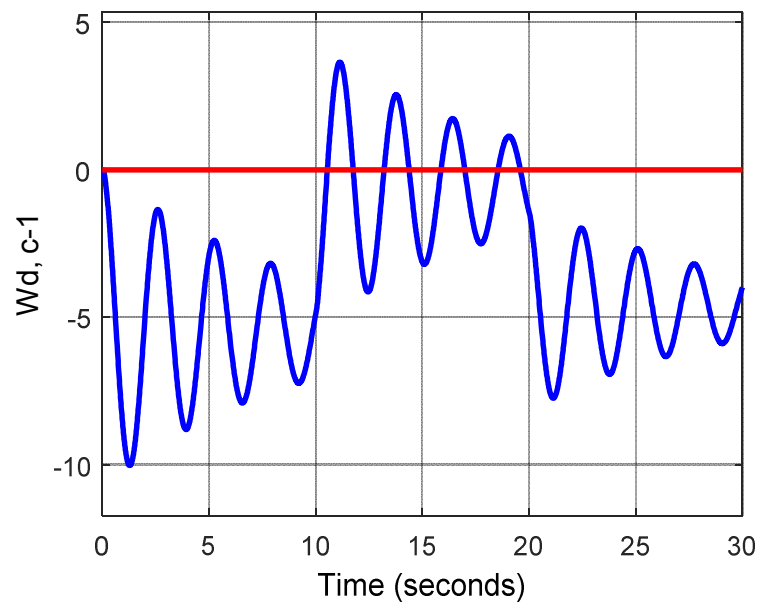
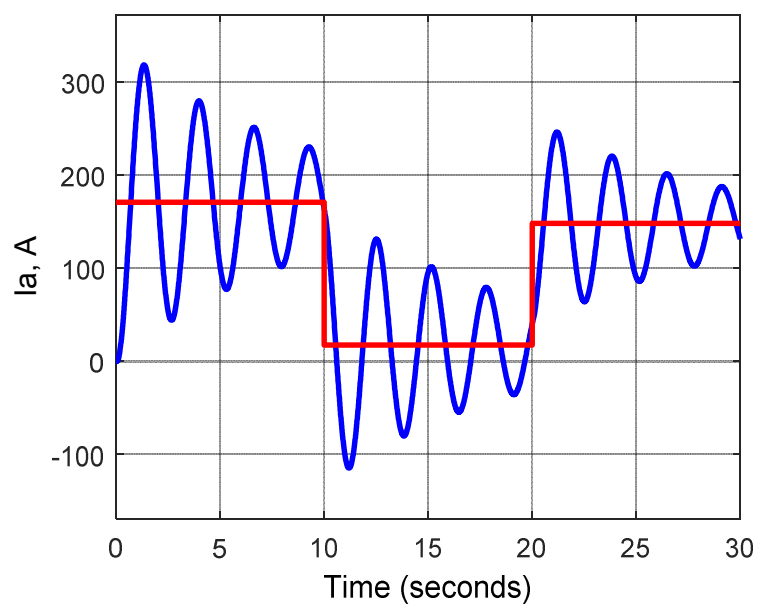


Рисунок 3.9 – Реакція двомасової системи на збурюючий вплив: графік швидкості механізму $\omega_m = f(t)$ (а) та пружного моменту $M_{пр} = f(t)$ (б)



a)



б)

Рисунок 3.10 – Реакція двомасової системи на збурюючий вплив: графік швидкості двигуна $\omega_d = f(t)$ (а) та струму двигуна $I_d = f(t)$ (б)

ВИСНОВКИ ДО РОЗДІЛУ 3

1. У розділі здійснено детальний аналіз роботи механізму розвороту роторного екскаватора з урахуванням того, що до платформи пружно приєднані інерційні маси – стріла з роторним колесом та консоль противаги. Показано, що подання механічної частини у вигляді жорсткої приведенної ланки дає лише усереднене відображення динаміки, оскільки не враховує вплив пружності реальних кінематичних з'єднань. Через скінченну жорсткість механічних зв'язків електропривід фактично працює як пружна система, у якій як керуюча дія двигуна, так і збурення у вигляді статичного моменту викликають коливання взаємопов'язаних мас. Ці коливання збільшують пікові навантаження в з'єднаннях і ускладнюють забезпечення необхідної точності рухів та позиціонування. Обґрунтовано, що з урахуванням податливості елементів механічну частину системи керування механізмом розвороту доцільно розглядати у вигляді двомасової моделі.

2. Побудовано математичну модель механізму у формі двомасової системи. Складено рівняння стану у класичному та матрично-векторному поданні, визначено склад вектора стану (змінні, що описують динаміку системи), сформовано вектор зовнішніх впливів, а також матриці стану **A** та керування **B**. Розроблено повну алгоритмічну схему моделі та її спрощений варіант – схему двомасової системи з оптимізованим за модульним критерієм контуром регулювання струму.

3. Виконано моделювання побудованої двомасової моделі на EOM із застосуванням програмного середовища MATLAB. У середовищі SIMULINK створено функціональну схему моделі системи. Для дослідження реакції моделі як тестовий вплив використовувався ступінчастий сигнал зі змінною амплітудою. За результатами моделювання встановлено, що перехідні процеси описуваних змінних стану мають виражений коливальний характер, що свідчить про недостатню якість динаміки системи у наявній конфігурації.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

4.1 Регулятори, реалізовані засобами пакета **Neural Network Toolbox** у середовищі **MATLAB**

Процес побудови нейромережевої системи управління виконувався за допомогою інструментів пакета **Neural Network Toolbox**, що входить до складу **MATLAB**. Далі наведено короткий огляд можливостей цього пакета, описано послідовність створення нейронного регулятора та подано результати моделювання роботи нейромережевої системи керування.

У **MATLAB** розглянуто три основні архітектури нейронних мереж, реалізованих у зазначеному пакеті прикладних програм, які використовуються при побудові таких типів регуляторів:

- регулятор із прогнозуванням майбутніх станів системи (**NN Predictive Controller**);
- регулятор, що ґрунтується на моделі авторегресії з ковзним середнім (**NARMA-L2 Controller**);
- регулятор на основі еталонної моделі (**Model Reference Controller**).

Використання нейронних мереж у задачах автоматичного керування передбачає поділ процесу проектування на два основних етапи:

- етап ідентифікації моделі об'єкта;
- етап синтезу закону керування.

Під час ідентифікації будується нейромережева модель об'єкта керування, яка надалі використовується для розробки потрібного регулятора. Для всіх трьох згаданих архітектур застосовується однакова процедура побудови моделі, однак принципи синтезу регуляторів між собою істотно відрізняються.

У випадку регулятора, що працює з прогнозом, нейронна модель об'єкта призначена для оцінювання його майбутніх станів, а оптимізаційний алгоритм підбирає таке керуюче діяння, яке мінімізує відхилення між бажаною та реальною траєкторіями вихідної величини.

Для регулятора типу NARMA-L2 структура керування являє собою модифіковану версію моделі самого об'єкта і будується на основі авторегресії з ковзним середнім.

У випадку використання еталонної моделі регулятор являє собою нейронну мережу, що навчається таким чином, щоб змусити реальний об'єкт повторювати поведінку вибраної еталонної системи. При цьому модель об'єкта задіюється безпосередньо в процесі налаштування параметрів регулятора.

Динамічні моделі систем керування з нейромережевими регуляторами зібрані в окремому підрозділі *Control Systems* бібліотеки Neural Network Toolbox (див. рис. 4.1).

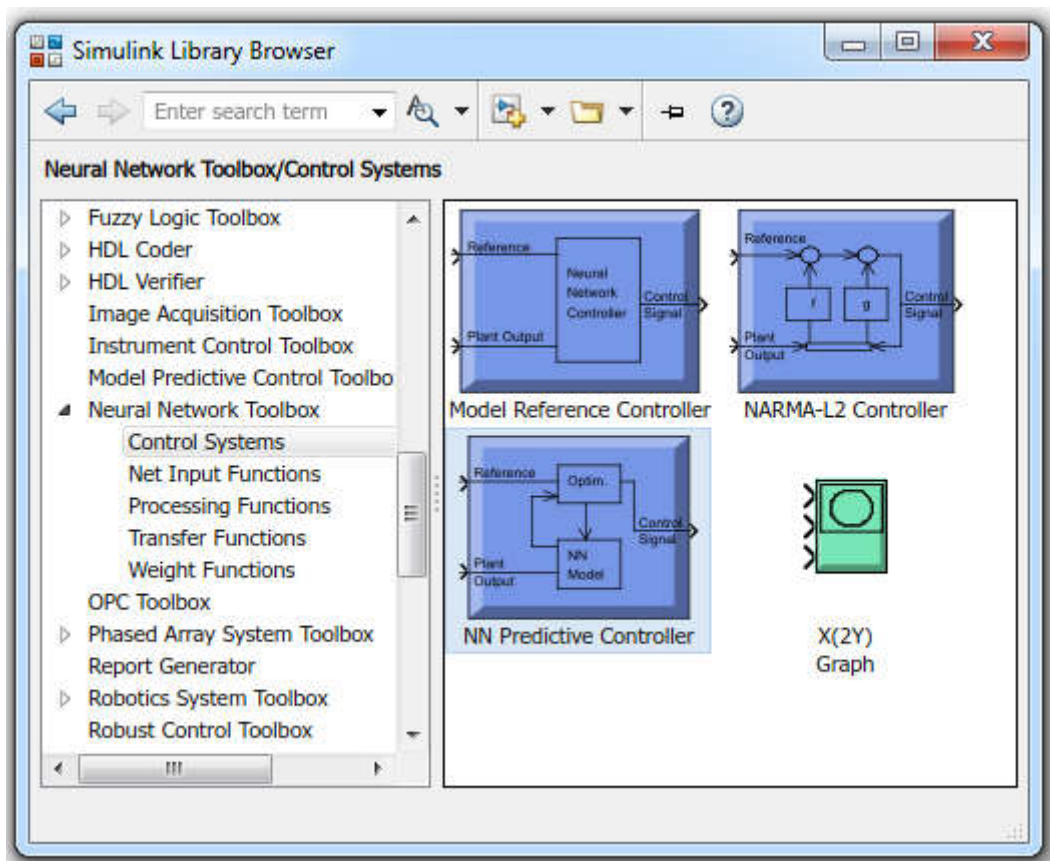


Рисунок 4.1 – Набір блоків Neural Network Blocksets підсистеми Control Systems середовища SIMULINK

Регулятор із прогнозуванням. У цьому типі регулятора нейромережева модель об'єкта використовується для прогнозу того, як система поводитиметься при різних можливих керуючих впливах. На основі цих прогнозів оптиміза-

ційний алгоритм вибирає такі сигнали керування, які забезпечують мінімальну похибку між заданою та фактичною траєкторією вихідного сигналу моделі. Формування нейромережевої моделі здійснюється окремо: мережу навчають пакетно, застосовуючи один із доступних навчальних алгоритмів. Недоліком цього підходу є значні обчислювальні витрати, оскільки оптимізація проводиться на кожному кроці формування керуючої дії.

Регулятор NARMA-L2 (на основі авторегресійної моделі з ковзним середнім). Серед трьох архітектур цей варіант є найбільш економним з точки зору обчислювальних ресурсів. По суті, регулятор є модифікацією отриманої на попередньому етапі нейромережевої моделі об'єкта. Основним недоліком методу є необхідність подавати модель у формі простору станів у канонічному вигляді зі строго визначеною структурою супровідної матриці. Це може спричинювати додаткові неточності в обчисленнях.

Регулятор з еталонною моделлю. Обсяг обчислень, необхідний для такого регулятора, приблизно відповідає обсягу, потрібному для NARMA-L2. Однак структура цього регулятора складніша: потрібно одночасно навчити нейромережу, що відтворює поведінку об'єкта, і нейромережу, яка виконує функції регулятора. Процес навчання складніший, оскільки використовує динамічний варіант алгоритму зворотного поширення помилки. Головною перевагою підходу є його універсальність – регулятор на базі еталонної моделі можна адаптувати до багатьох типів об'єктів керування.

Для електроприводу, що аналізується у цій роботі, було обрано саме регулятор із прогнозуванням. Варіанти на основі NARMA-L2 та еталонної моделі не забезпечили достатнього рівня якості керування в процесі моделювання.

4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller

Регулятор типу NN Predictive Controller, реалізований у складі пакету Neural Network Toolbox системи MATLAB, ґрунтується на використанні ней-

ромережевої моделі нелінійного об'єкта керування. Така модель забезпечує можливість передбачити подальший розвиток процесів у системі, а також дозволяє сформувати оптимальний керуючий вплив на визначеному інтервалі прогнозування.

4.2.1 Процедура ідентифікації об'єкта керування

Схему модуля ідентифікації показано на рис. 4.2. До складу цього модуля входить нейромережева модель об'єкта, яка навчається у відокремленому (автономному) режимі. Метою навчання є мінімізація розбіжності між реакцією реального об'єкта та відповіддю моделі на подану послідовність тестових впливів.

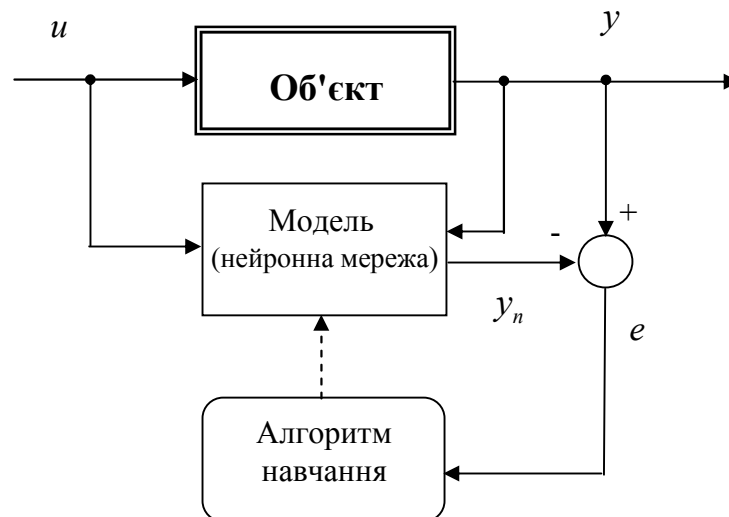


Рисунок 4.2 – Структурна схема підсистеми ідентифікації

Архітектуру нейромережі, що використовується для моделювання об'єкта, наведено на рис. 4.3. Вона є дворівневою (двошаровою) і включає лінії затримки, які дозволяють зберігати попередні значення сигналів на вході та виході об'єкта. Це забезпечує можливість прогнозувати майбутні значення вихідної координати.

Параметри мережі налаштовуються автономно, на основі набору експериментальних даних, отриманих у процесі тестування системи. Для навчання може застосовуватися будь-який зі стандартних алгоритмів нейромережевого

навчання. У даному дослідженні використано метод Левенберга–Марквардта, який в Neural Network Toolbox реалізований за допомогою М-функції **trainlm**.

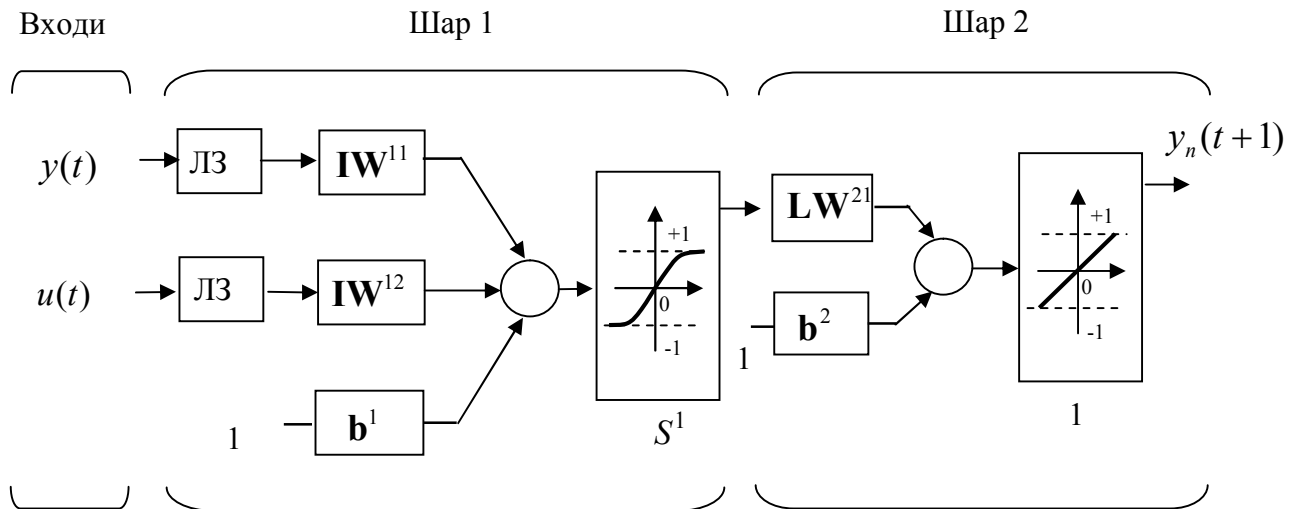


Рисунок 4.3 – Структура нейронної мережі, що моделює об’єкт керування

4.2.2 Принцип функціонування прогнозуючої системи керування

Алгоритм управління на основі прогнозу працює за ідеологією «ковзного горизонту». Це означає, що нейромережева модель об’єкта безперервно обчислює ймовірну реакцію системи на деякому майбутньому відрізку часу. Отриманий прогноз передається в оптимізаційний модуль, який обирає таку керуючу дію, що мінімізує критерій якості:

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

Тут коефіцієнти N_1 , N_2 і N_4 визначають межі інтервалу, на якому аналізуються відхилення вихідного сигналу та енергетичні витрати на формування управління. Змінна u' – пробний керуючий вплив, y_r – бажана траєкторія, а y_m – прогнозована реакція моделі. Параметр ρ регулює вагу енергетичної складової у функції якості.

Узагальнену блок-схему системи із застосуванням прогнозуючого принципу наведено на рис. 4.4. Вона складається з нейронної моделі об’єкта та оп-

тимізаційного модуля. Останній знаходить значення u' , які мінімізують критерій якості, а сформований сигнал керування надходить безпосередньо на об'єкт.

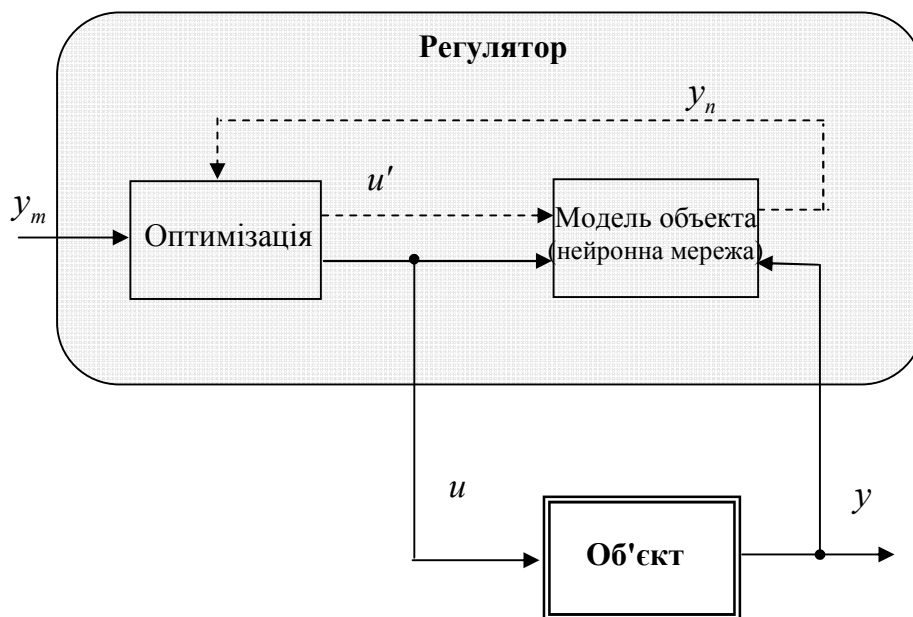


Рисунок 4.4 – Узагальнена схема керування з використанням прогнозуючого регулятора

4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB

Під час побудови нейромережевого регулятора типу NN Predictive Controller використовується набір спеціалізованих функцій і моделей, що входять до складу каталогу *toolbox/nnet/nncntrl* пакету Simulink. До цього набору належать такі групи програмних засобів:

1. Функції одновимірної оптимізації

Ці алгоритми застосовуються під час пошуку оптимального керуючого впливу в процесі роботи регулятора:

- **Csrchbac** – алгоритм пошуку за схемою «зворотного проходу» (backtracking).
- **Csrchbre** – модифікований метод Брента, який поєднує техніку золотого перетину та квадратичну інтерполяцію.

- **Csrchcha** – метод кубічної інтерполяції Чараламбуса.
- **Csrchgol** – класичний метод золотого перетину.
- **Csrchhyb** – комбінований алгоритм, що використовує бісекцію та кубічну інтерполяцію.

2. Функції, що реалізують механізм керування з прогнозуванням

Ці М-функції забезпечують обчислення критеріїв якості та роботу оптимізаційних алгоритмів:

- **Calcjddjj** – оцінювання критерію якості та його градієнта;
- **Predopt** – основний модуль оптимізації прогнозуючого регулятора;
- **Dyduvar** – розрахунок часткових похідних вихідних сигналів за входами.

3. Моделі та графічні додатки Simulink

- **Predcstr** – графічний інтерфейс користувача (GUI) для роботи з регулятором типу Predictive Controller;
- **Prest3sim2** – нейромережева модель об'єкта керування, що застосовується функцією *deport* для формування прогнозу поведінки системи.

4. Допоміжні програмні модулі

- **Nncontrolutil** – набір службових функцій, що забезпечують взаємодію регулятора з внутрішніми компонентами Simulink;
- **Nnident.m** – ключова функція, яка використовується під час етапу ідентифікації об'єкта. Розташована у підкаталозі *private*, забезпечує створення навчальної вибірки, формування нейронної мережі та процес її навчання через зручний GUI.

Після підготовки необхідних даних у середовищі Simulink формується модель системи керування, структура якої наведена на рисунку 4.5.

У представленій конфігурації системи міститься підсистема, що моделює об'єкт керування (блок *Subsystem*), модуль нейромережевого регулятора (*NN Predictive Controller*), генератор випадкового еталонного ступінчастого сигналу (*Random Reference*), а також елементи для візуалізації результатів.

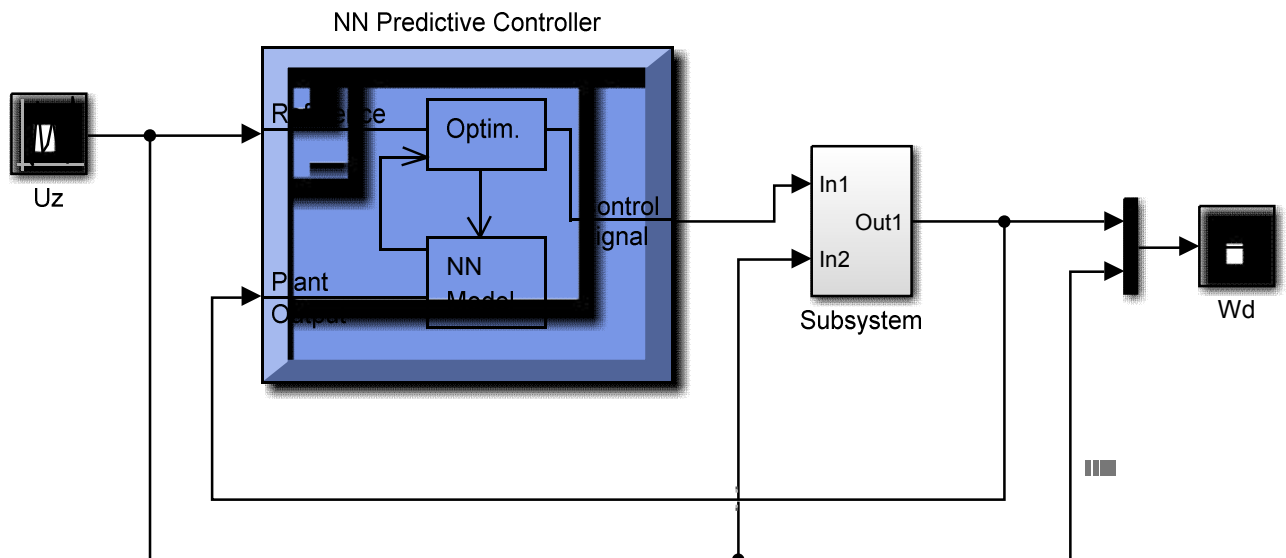


Рисунок 4.5 – Структурна модель системи керування з нейрорегулятором NN Predictive Controller

Важливо зазначити, що дана схема одночасно виконує роль стандартної моделі середовища SIMULINK і працює як графічний інтерфейс користувача (GUI), забезпечуючи безпосередню взаємодію з налаштуваннями системи.

Структура внутрішньої організації нейромережевого регулятора NN

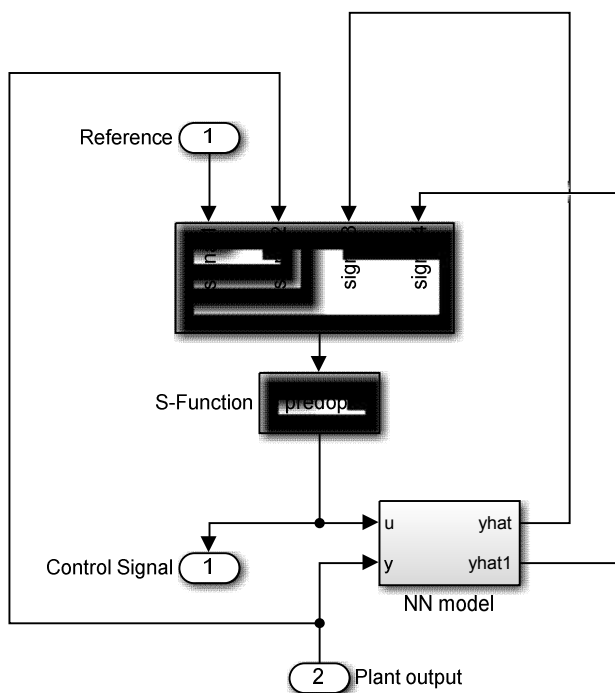


Рисунок 4.6 – Структурна схема нейрорегулятора

Predictive Controller подана на рис. 4.6. Вона відкривається у спеціальному вікні після вибору команди **Look Under Mask** для відповідного блоку.

Розробка регулятора розпочинається з активації блоку NN Predictive Controller. Після цього відкривається інтерфейс, зображений на рис. 4.7, який слугує панеллю налаштування параметрів. У верхній частині вікна відображається інформаційне повідомлення *«Perform plant*

identification before controller configuration», яке підказує, що перед подальшими налаштуваннями обов'язково потрібно провести етап ідентифікації – сформувати нейромережеву модель керованого об'єкта за допомогою процедури **Plant Identification**.

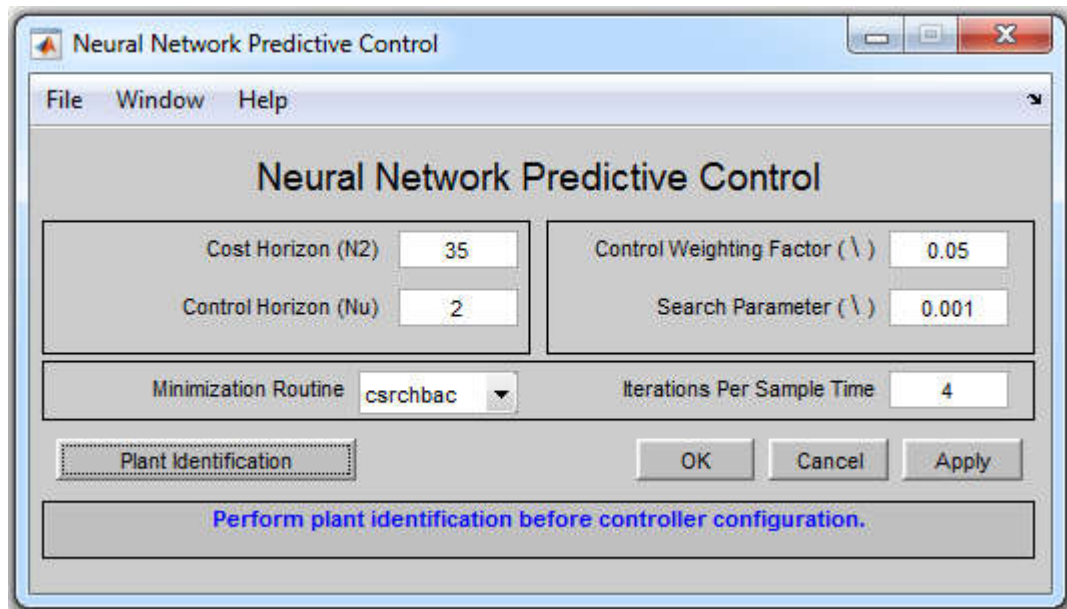


Рисунок 4.7 – Інтерфейс налаштування параметрів регулятора

На рис. 4.8 показано вікно модуля Plant Identification. Воно є універсальним інструментом, що дозволяє створювати нейромережеві моделі для будь-яких динамічних систем, описаних у SIMULINK. Для даної роботи таким об'єктом виступає модель двомасового механізму.

Під час ідентифікації формується нейронна мережа, яка повинна відтворювати поведінку реального об'єкта, оскільки саме ця модель буде використана в подальшому для налаштування та оптимізації регулятора. Тому етап створення моделі має виконуватися до розрахунків самого регулятора.

Процедура ідентифікації вимагає попереднього задання низки параметрів, які налаштовуються через вікно **Plant Identification**. Це вікно поділене на кілька логічних секцій, кожна з яких відповідає за певний аспект формування нейромережевої моделі.

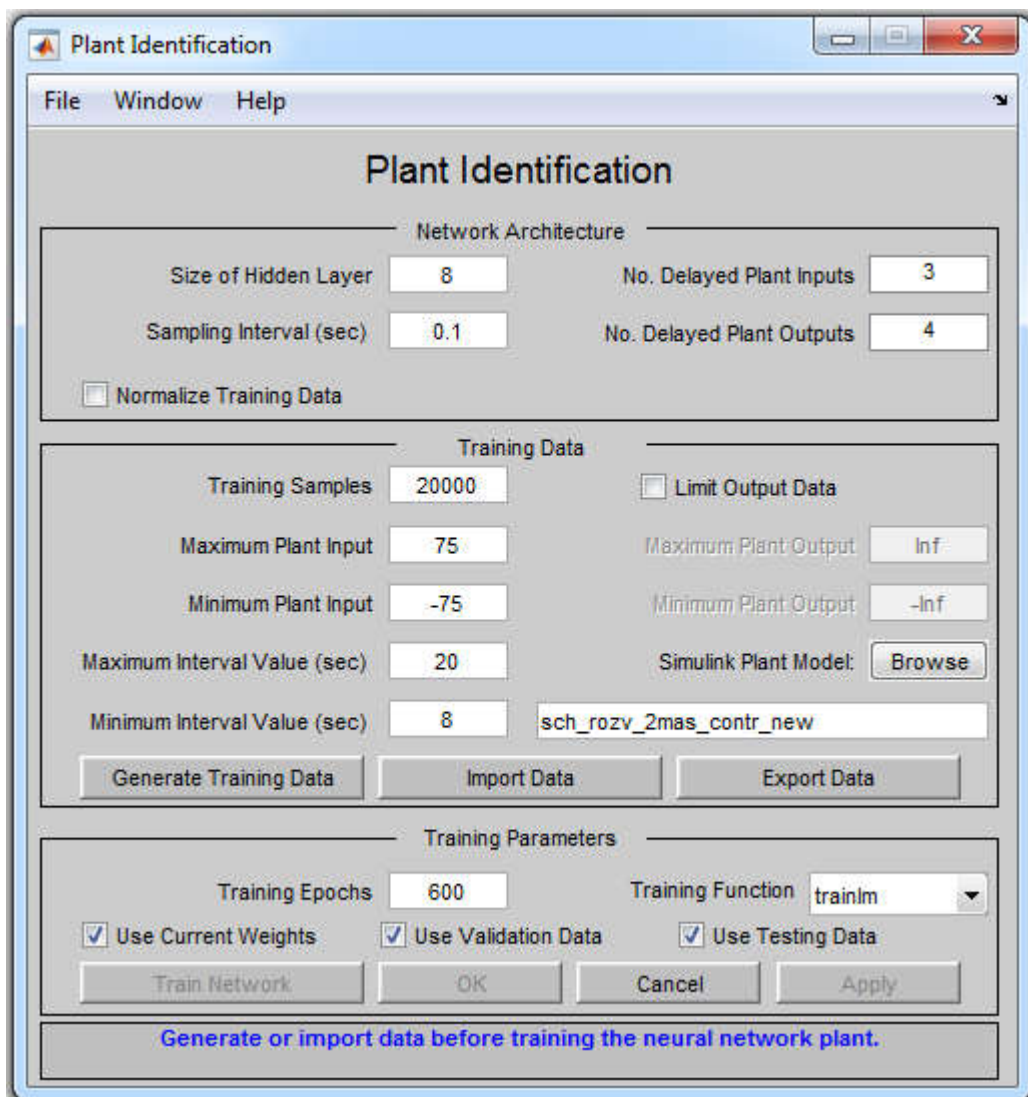


Рисунок 4.8 – Вікно процедури ідентифікації об'єкта керування

Опис вікна Plant Identification

Секція Network Architecture (Архітектура мережі).

У цій частині задаються параметри, які визначають структуру нейронної мережі, що використовується для ідентифікації та подальшого відтворення динаміки досліджуваної системи. Коректний вибір налаштувань у секції **Network Architecture** є принципово важливим, оскільки саме від них залежить, наскільки точно нейромережа здатна описати поведінку об'єкта.

Параметр Size of Hidden Layer (кількість нейронів прихованого шару). Цей параметр вказує, скільки нейронів міститиметься у прихованому шарі мережі, яка застосовується для моделювання або ідентифікації об'єкта. Вибір

розміру прихованого шару є одним із ключових рішень при проектуванні нейромережі:

- збільшення кількості нейронів, як правило, дозволяє мережі краще відтворювати складні, нелінійні залежності в даних;
- однак занадто великий прихований шар може спричинити перенавчання (overfitting), коли мережа добре відтворює навчальні дані, але погано узагальнює поведінку на нових вхідних впливах.

Тому розмір прихованого шару зазвичай розглядають як гіперпараметр, який підбирають експериментально, орієнтуючись на якість моделювання для конкретної задачі.

Параметр Sampling Interval (sec) (інтервал дискретизації, с). Цей параметр задає часовий інтервал між двома послідовними вимірюваннями сигналів, що використовуються для ідентифікації. Іншими словами, це період, з яким знімаються дані з моделі або реального об'єкта.

Вибір інтервалу дискретизації має суттєвий вплив на:

- деталізацію та роздільну здатність отриманих даних;
- точність подальшого відтворення динаміки системи.

Якщо інтервал вибірки занадто великий, модель може «не побачити» швидких змін у процесі, а надто малий інтервал призводить до надлишку даних і зростання обчислювальних витрат. Тому при заданні Sampling Interval необхідно враховувати швидкодію об'єкта і вимоги до точності результатів ідентифікації.

No. Delayed Plant Inputs (кількість затриманих входів). Цей параметр визначає, скільки попередніх значень вхідних сигналів системи беруться до уваги під час ідентифікації. Урахування запізнених входів дає змогу моделі відобразити наявні у системі затримки та більш коректно відтворити її динаміку.

Такі затримки часто зустрічаються в реальних керованих процесах, де реакція на вхід проявляється із певним часовим відставанням. Тому коректне встановлення параметра **No. Delayed Plant Inputs** є важливим для отримання

точної моделі, особливо коли відомо, що між дією на об'єкт і його реакцією існує часовий лаг.

Правильне задання цього параметра дає можливість адаптувати модель до реальних умов роботи системи та підвищити якість процесу ідентифікації.

No. Delayed Plant Outputs (кількість затриманих виходів). Параметр указує кількість попередніх значень вихідних сигналів, які залучаються до формування моделі. Використання запізнених виходів дозволяє врахувати затримки у формуванні відгуку системи, що суттєво впливає на точність моделювання її поведінки.

Цей параметр визначає, наскільки глибоко модель аналізує історію вихідних значень. Якщо система має значні внутрішні затримки, то включення більшої кількості запізнених виходів може суттєво підвищити адекватність отриманої моделі.

Оптимальний вибір No. Delayed Plant Outputs залежить від властивостей конкретного об'єкта та рівня точності, якого потрібно досягти під час ідентифікації.

Normalize Training Data (нормалізація навчальних даних). Цей пункт відповідає за попереднє масштабування вхідних і вихідних сигналів перед їх використанням у процесі навчання нейромережевої моделі. Вмикання цього параметра означає, що всі дані будуть приведені до певного стандартного діапазону чи форми – наприклад, до інтервалу $[0;1]$, або ж віднормовані за середнім значенням і стандартним відхиленням.

Якщо нормалізація активована, система автоматично перетворює дані та подає на вхід мережі більш узгоджені значення. Якщо ж цей пункт вимкнено, навчання виконується на сирих даних без попередньої обробки.

Основне призначення нормалізації – забезпечити стабільніше та швидше навчання нейронної мережі, особливо у випадках, коли окремі змінні мають різні масштаби або різко відрізняються за величинами. Коректно виконане масштабування також позитивно впливає на точність і якість побудованої моделі.

Доцільність увімкнення **Normalize Training Data** визначається особливостями конкретної системи та вимогами до процесу її ідентифікації.

Секція Training Data (навчальні дані). У розділі *Training Data* задається набір сигналів, на основі яких формується нейромережева модель досліджуваної системи. До таких даних входять часові послідовності вхідних впливів та відповідних виходів об'єкта, що дозволяє мережі відтворити закономірності та динаміку реального процесу.

Головне призначення цього набору – забезпечити мережу достатньою кількістю інформації для коректного навчання, щоб побудована модель могла описувати поведінку системи в різних умовах. Як правило, до Training Data включають результати роботи об'єкта в різних режимах, що підвищує здатність моделі враховувати широке коло можливих станів та реакцій.

Під час ідентифікації саме на цій вибірці нейронна мережа шукає відповідність між вхідними і вихідними сигналами, що визначає точність отриманої моделі. Тому якість і повнота навчальних даних безпосередньо впливають на надійність побудованої моделі.

Training samples (кількість навчальних точок). Цей параметр визначає розмір навчальної вибірки – тобто число вимірів, використаних для побудови моделі. Від його значення залежить, наскільки глибоко та точно нейронна мережа зможе відтворити характеристики керованого об'єкта: надто мала вибірка може призвести до неточностей, тоді як достатній обсяг даних забезпечує більш адекватну ідентифікацію.

Maximum Plant Input (верхня межа вхідного сигналу). Цей параметр визначає найбільше допустиме значення сигналу, що подається на вхід системи під час процедури ідентифікації. Фактично він задає верхню границю амплітуди вхідного впливу, який використовуватиметься у навчальній вибірці.

Коректне встановлення цього обмеження дає змогу забезпечити, щоб модель навчалася лише в межах реальних або фізично обґрунтованих режимів роботи об'єкта. Якщо допустимий максимум буде вибрано занадто великим, то

мережа може навчитися на значеннях, які не відповідають реальним умовам експлуатації, що негативно позначиться на точності прогнозування.

Minimum Plant Input (нижня межа вхідного сигналу). Параметр визначає мінімально можливе значення керуючого сигналу, який може подаватися під час збирання даних для ідентифікації. Це нижня межа діапазону вхідних впливів, що враховується у процесі навчання нейромережевої моделі.

Разом із максимальним значенням цей параметр формує допустимий інтервал вхідних сигналів, у межах якого система вважається працездатною. Невдало вибрані межі можуть спричинити зниження якості моделі, оскільки мережа або не отримає достатньо різноманітних даних, або навпаки – опрацьовуватиме нереалістичні значення сигналів.

Правильно заданий діапазон входів забезпечує адекватність моделі та її здатність точно передбачати реакцію системи в реальних умовах.

Maximum Interval Value (sec) – верхня межа інтервалу часу. Цей параметр задає максимальну тривалість інтервалу у секундах, протягом якого використовуються вхідні та вихідні сигнали для ідентифікації динаміки системи. По суті, він визначає максимально допустиму довжину часу, на яку поширюється навчальний набір даних під час побудови моделі.

Встановлення цього значення дозволяє обмежити часовий діапазон аналізованої інформації. Це особливо корисно, коли деякі періоди роботи системи менш значущі для процесу ідентифікації, а дані в інших проміжках є ключовими. Вибір параметра Maximum Interval Value зазвичай враховує специфіку досліджуваного об'єкта та тривалість циклів його роботи, обмежуючи аналіз лише необхідним часовим відрізком.

Minimum Interval Value (sec) (нижня межа інтервалу часу). Цей параметр визначає мінімальну тривалість інтервалу у секундах, протягом якого враховуються вхідні та вихідні дані для ідентифікації системи. Іншими словами, він задає коротший допустимий часовий проміжок для навчальної вибірки.

Налаштування Minimum Interval Value дозволяє контролювати, які частини часового ряду вважати значущими для побудови моделі. Використання цьо-

го параметра допомагає виключати надто короткі або неінформативні інтервали, забезпечуючи, щоб у навчанні брали участь тільки дані з необхідною тривалістю або значущістю. Правильне встановлення мінімального інтервалу дозволяє оптимізувати ідентифікаційний процес відповідно до особливостей та вимог конкретної системи.

Limit Output Data (обмеження вихідних даних). Цей параметр активує вікно управління, яке дозволяє регулювати обсяг вихідних даних, що використовуються під час ідентифікації. При ввімкненні цього вікна стають доступними два додаткові поля для редагування:

- Maximum Plant Output – максимальне значення вихідного сигналу.
- Minimum Plant Output – мінімальне значення вихідного сигналу.

Параметри Maximum Plant Output та Minimum Plant Output визначають обмеження вихідних даних, які беруть участь у процесі навчання моделі та аналізу динаміки системи. Це особливо корисно, коли потрібно сконцентруватися на певному діапазоні вихідних величин або дослідити специфічну поведінку системи протягом конкретного проміжку часу.

Використання цих обмежень дозволяє зменшити обсяг оброблюваних даних, спростити процес навчання та зробити ідентифікацію моделі більш ефективною.

Simulink Plant Model (модель об'єкта в Simulink). Цей параметр задає модель керованого об'єкта в середовищі Simulink, визначаючи її вхідні та вихідні порти, які використовуються для побудови нейромережевої моделі. Вибирається конкретна модель об'єкта або системи для ідентифікації. За допомогою кнопки Browser здійснюється вибір моделі Simulink; у цій роботі використовується схема моделі, наведена на рис. 4.9.

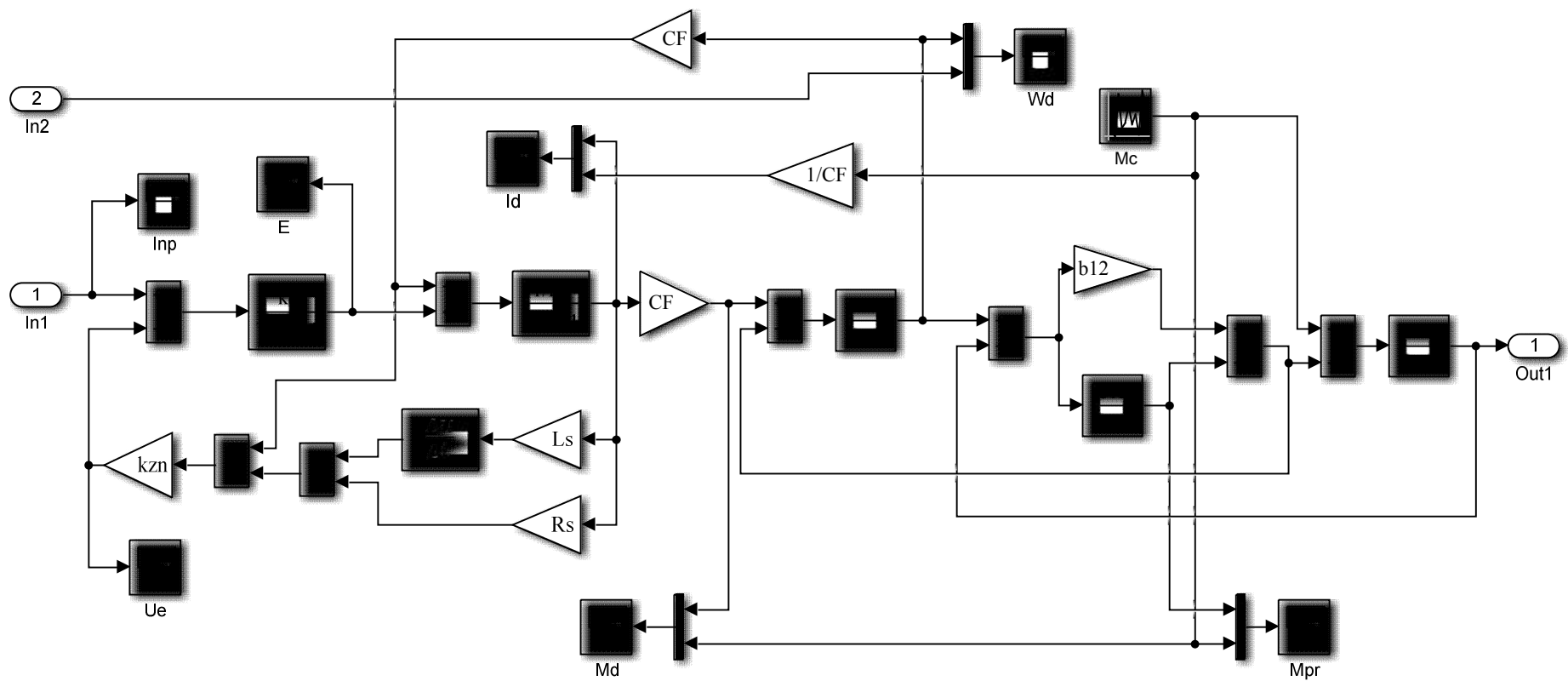


Рисунок 4.9 – Модель керованого об'єкта нейрорегулятора

Generate Training Data (генерація навчальних даних). Ця функція запускає процес створення навчальної послідовності для ідентифікації динаміки системи на основі заданих параметрів та моделі об'єкта. Вона дозволяє формувати симульовані дані для навчання, коли реальні виміряні дані недоступні.

Import Data (імпорт даних). Дозволяє завантажити навчальну послідовність із робочої області або зовнішнього файлу для використання у процесі ідентифікації.

Export Data (експорт даних). Надає можливість зберегти згенеровані дані в робочу область Simulink або у файл формату MAT для подальшого використання.

Секція Training Parameters (параметри навчання). Параметри навчання (Training Parameters) визначають, яким чином буде організовано процес навчання моделі системи на основі навчальних даних. Вони забезпечують керування процесом ідентифікації та впливають на точність і ефективність отриманої моделі.

Training Epochs (кількість епох навчання). Параметр Training Epochs задає число ітерацій або повних циклів навчання нейромережі. Кожна епоха передбачає, що модель проходить через весь набір навчальних даних, оновлюючи внутрішні параметри для поліпшення точності відтворення динаміки системи.

Правильне визначення кількості епох є критично важливим: якщо їх занадто мало, модель може залишитися недостатньо навченою і не здатною точно відтворювати поведінку системи; надмірна кількість епох може призвести до перенавчання (overfitting), коли модель запам'ятовує навчальні дані, але погано узагальнює нові випадки. Оптимальна кількість епох залежить від конкретної задачі ідентифікації та характеристик навчальних даних.

Training Function (навчальна функція). Параметр Training Function визначає алгоритм, за допомогою якого модель нейромережі навчається на підготовлених даних та оптимізує свої параметри. Від правильного вибору функції навчання залежить ефективність процесу та точність отриманої моделі.

У MATLAB і пакеті Neural Network Toolbox доступні різні функції навчання, такі як:

- `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 налаштовувати для досягнення максимальної точності моделі та здатності адекватно відтворювати динаміку системи. Training Function визначає, як нейромережа коригує внутрішні ваги та параметри під час навчання для підвищення якості прогнозу та стабільності моделі.

Use Current Weights (використовувати поточні ваги). Це елемент керування, який дозволяє підтвердити, що під час подальшої ідентифікації та навчання нейронної мережі слід використовувати вже наявні, попередньо сформовані вагові коефіцієнти.

Параметр *Use Current Weights* визначає, чи застосовуватимуться актуальні значення ваг як стартова точка для нового процесу навчання моделі. Він безпосередньо впливає на ініціалізацію мережі:

- Якщо *Use Current Weights* встановлено на «Так» (активовано), тоді мережа починає тренування зі значеннями ваг, які сформувалися під час попередніх навчальних процедур. Тобто навчання продовжує вдосконалення вже наявної моделі.
- Якщо параметр вимкнений («Ні»), навчання розпочинається знову – ваги ініціалізуються випадковим чином або відповідно до обраного алгоритму початкової ініціалізації.

Використання вже наявних ваг доцільне, коли потрібно досягти покращення або точнішої корекції існуючої моделі, а також коли попередні ваги вже певною мірою відображають динаміку об'єкта. Важливо розуміти, що цей параметр визначає стартові умови процесу навчання та може суттєво впливати на кінцеву точність ідентифікації системи.

Use Validation Data (використовувати дані валідації). Цей параметр задає, чи потрібно залучати окрему вибірку даних для етапу валідації під час навчання нейронної мережі. Валідація є важливою процедурою, що дозволяє оцінити здатність моделі узагальнювати інформацію, а не просто запам'ятовувати навчальні приклади.

- Якщо опцію *Use Validation Data* активовано («Так»), навчальний процес використовує додатковий набір даних, який не входить до навчальної вибірки. Це дає змогу оцінити продуктивність моделі на даних, які вона не бачила під час навчання. Такий підхід допомагає виявити перенавчання та визначити рівень узагальнення.

- Якщо параметр вимкнено («Ні»), валідація не проводиться, і результати оцінки моделі ґрунтуються лише на навчальній вибірці. Такий варіант може застосовуватися у випадках, коли немає можливості сформувати додатковий набір даних.

Використання валідаційної вибірки зазвичай рекомендується, оскільки вона забезпечує більш точну оцінку якості моделі та допомагає уникнути деградації її узагальнюючих властивостей.

Use Testing Data (використовувати тестові дані). Це налаштування визначає, чи буде застосовано спеціальний набір тестових даних для підсумкової оцінки роботи моделі після завершення процесу навчання та валідації. Тестові дані є незалежною вибіркою, яку модель ніколи не бачить під час навчання.

- Якщо параметр *Use Testing Data* встановлено на «Так», то після навчання на тренувальних даних і перевірки на валідаційних, модель проходить додаткову оцінку на тестовому наборі. Це дозволяє отримати об'єктивну картину того, наскільки якісно модель зможе працювати з абсолютно новими сигналами та чи здатна вона коректно відтворювати реальну динаміку системи.

- Якщо параметр вимкнено («Ні»), тестування не використовується, і оцінювання моделі обмежується навчальною та, за потреби, валідаційною вибірками.

Використання тестових даних є важливим етапом для перевірки здатності моделі до узагальнення, запобігання перенавчанню та оцінки її практичної придатності.

Процедура Generate Training Data (генерувати навчальні дані). Після вибору цієї опції запускається програмний модуль, який формує навчальну послідовність. Генерація відбувається шляхом подачі на модель керованого об'єкта в середовищі *Simulink* набору випадкових ступінчастих сигналів. У процесі формування навчальних даних на екран виводяться графіки вхідних та вихідних сигналів об'єкта (рис. 4.10), що дозволяє візуально контролювати характер реагування системи.

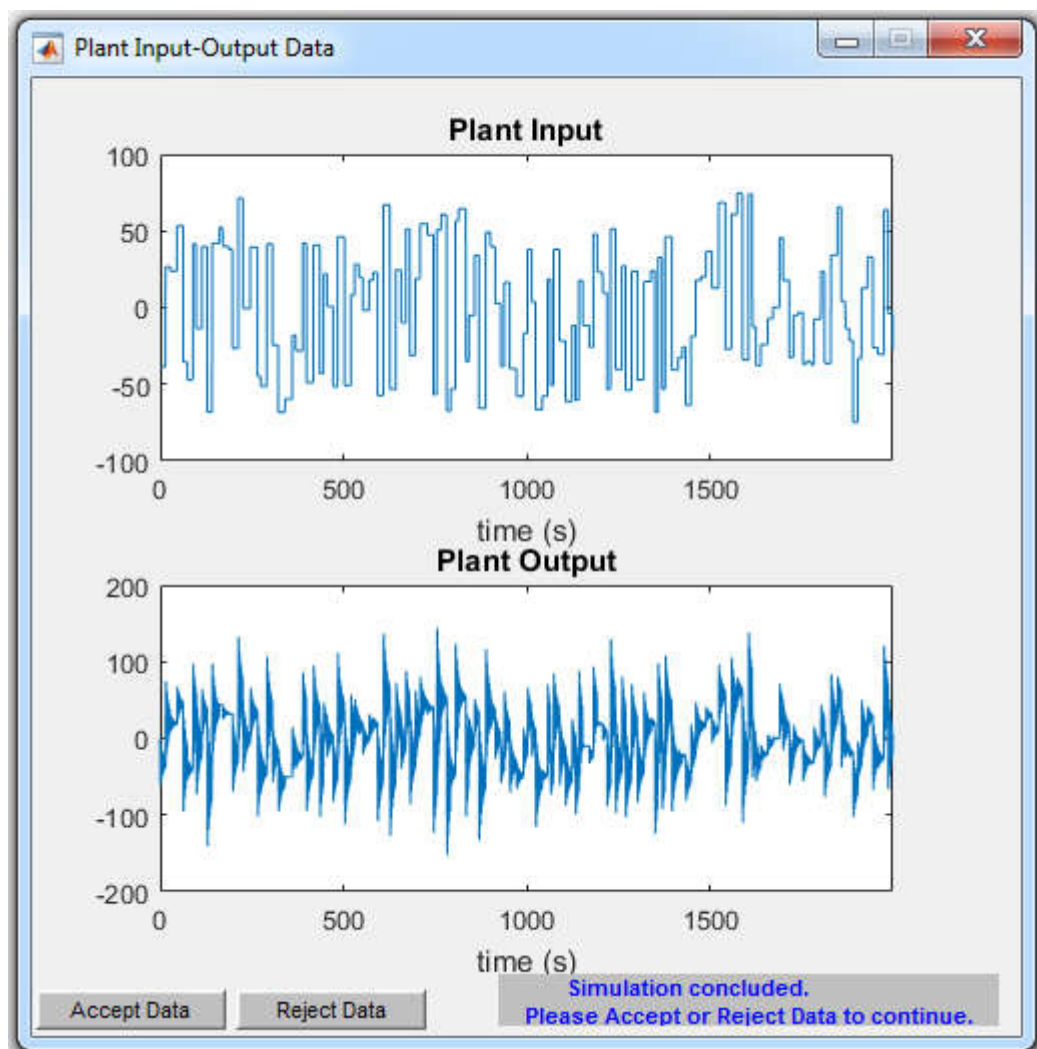


Рисунок 4.10 – Графіки вхідного та вихідного сигналів
під час генерації навчальної послідовності

Після завершення генерації користувачу пропонується прийняти отримані дані (*Accept Data*) або відхилити їх (*Reject Data*), якщо вони не відповідають вимогам щодо структури чи якості.

Якщо дані прийняті користувачем, програма автоматично відкриває модифіковане вікно **Plant Identification** (рис. 4.11). У цьому вікні деякі елементи інтерфейсу стають недоступними для редагування, а кнопка **Generate Training Data** змінюється на кнопку **Erase Generated Data**, що надає можливість видалити раніше згенеровані навчальні дані.

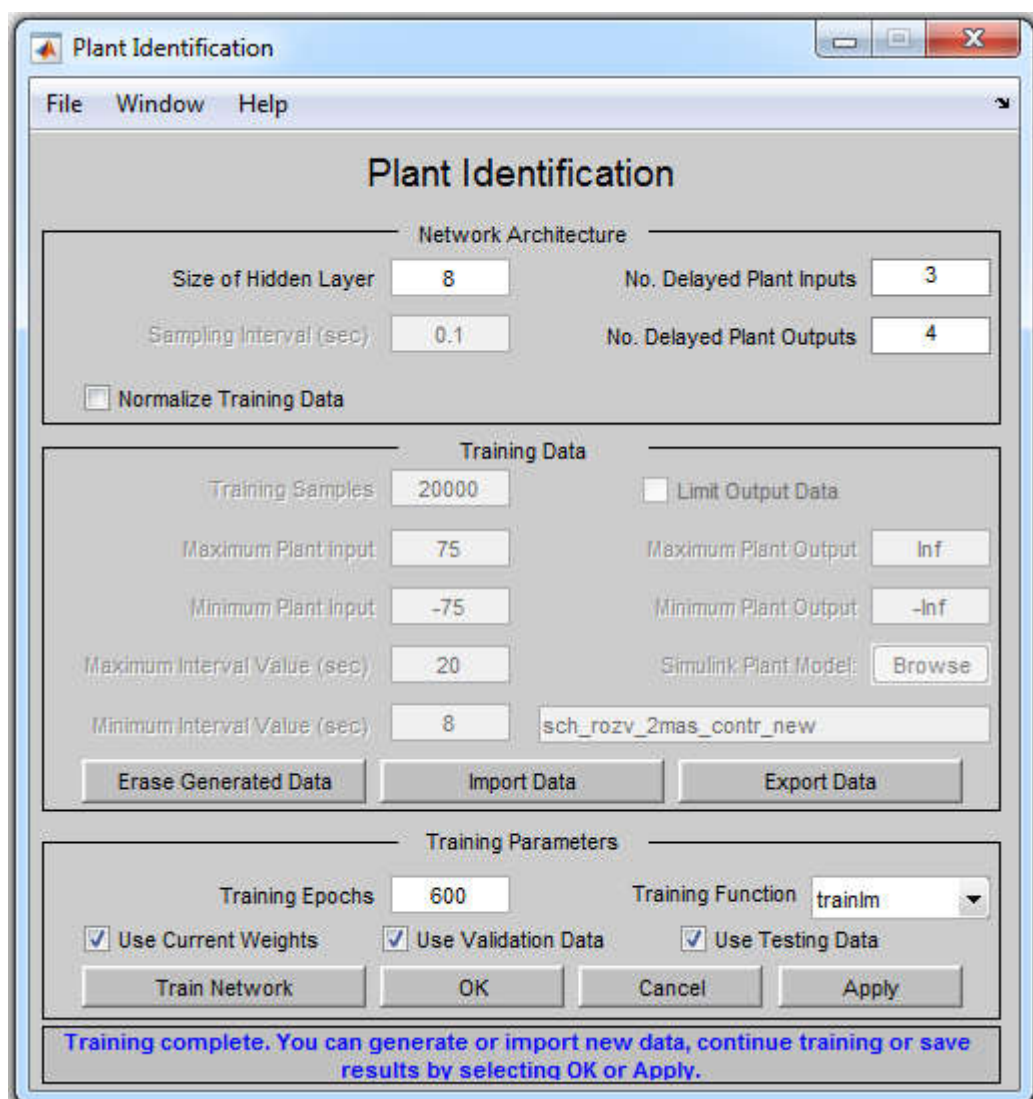


Рисунок 4.11 – Вікно Plant Identification після генерації навчальної послідовності

Натискання кнопки **Train Network** ініціює створення нейронної мережі з прямим розподілом сигналу за допомогою М-функції **newff**. Ця функція не тільки формує мережу з ім'ям **netn**, але й автоматично ініціалізує її ваги та зсуви, готуючи мережу до навчання. Створену нейронну мережу можна відобразити в середовищі Simulink за допомогою оператора **gensim(netn)** (див. рис. 4.12).

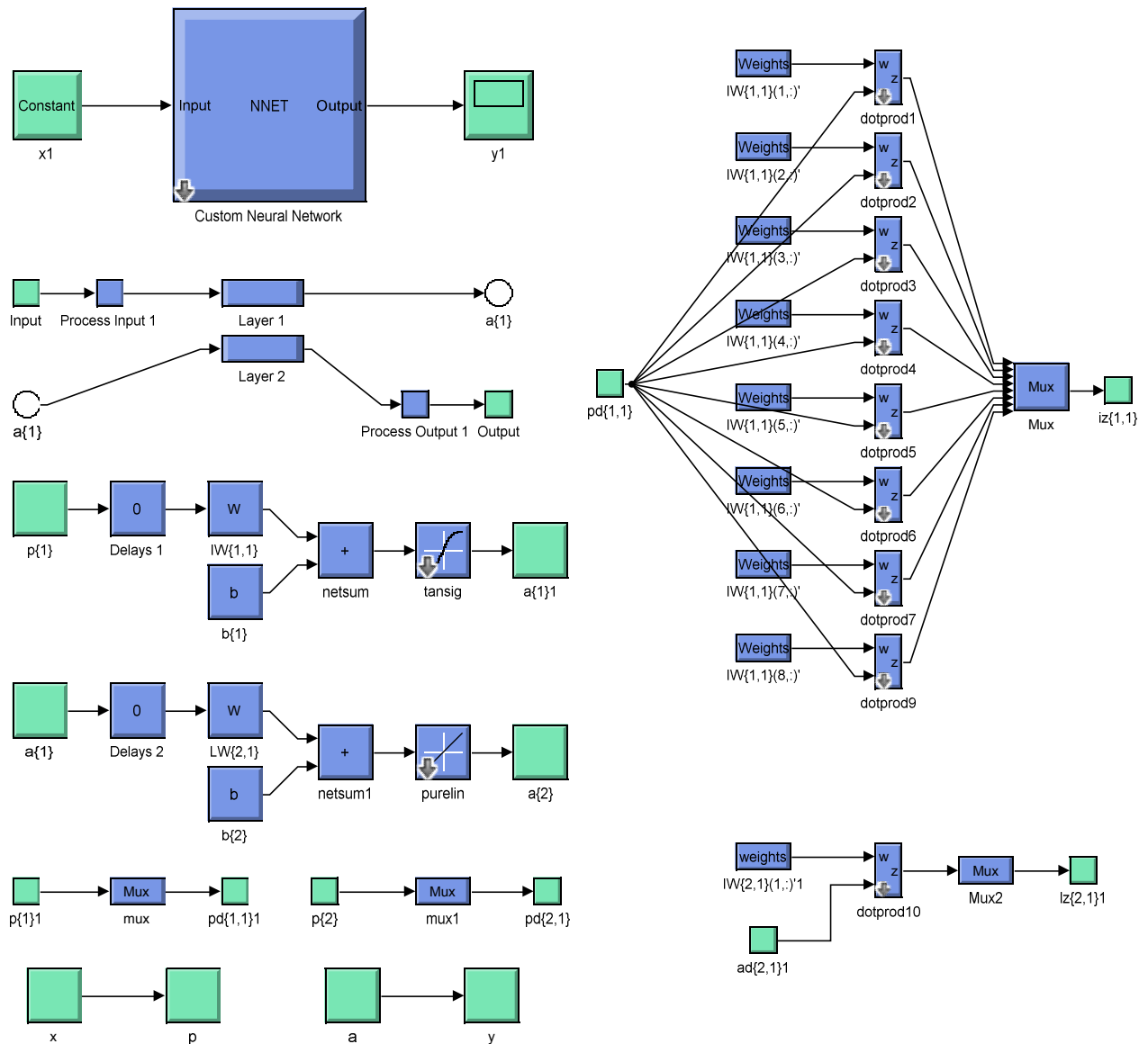


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

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

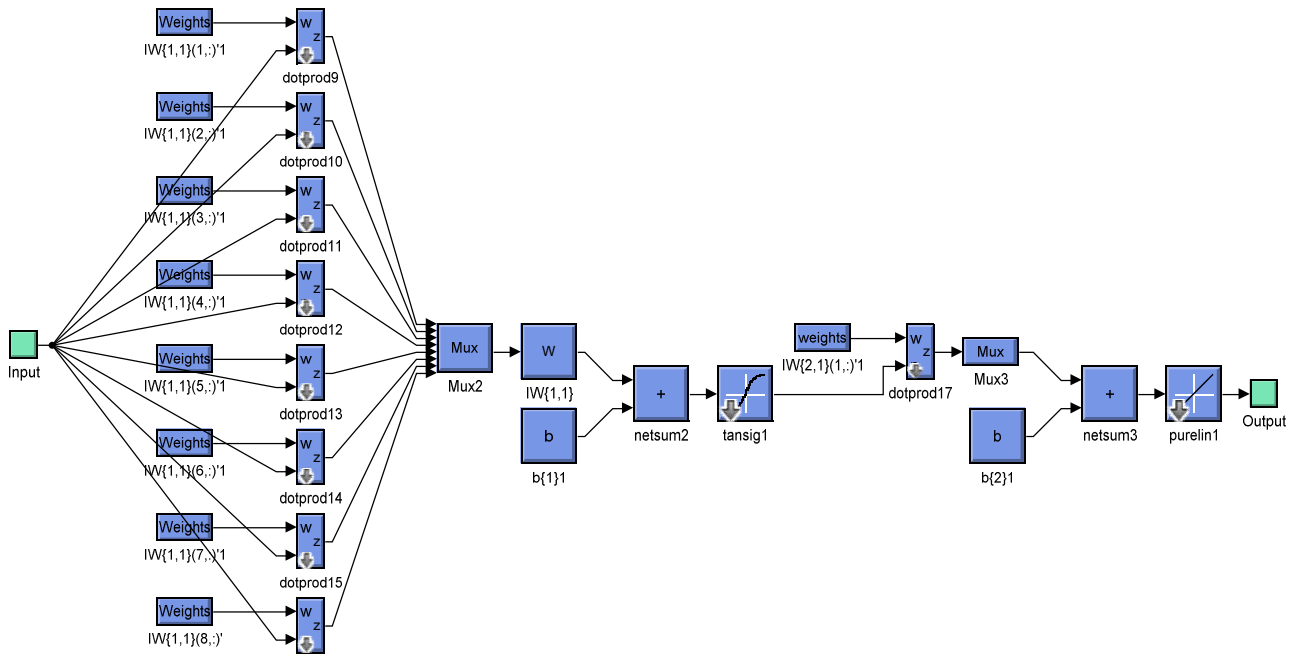


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

Створена мережа є статичною, тобто вона не містить елементів затримки або зворотних зв'язків. Мережа приймає один вектор входу з шістьма елементами, має два шари: перший – прихований, з визначеною кількістю нейронів, другий – вихідний, що складається з одного нейрона. Активаційні функції обрані такі: у прихованому шарі використовується гіперболічний тангенс (**tansig**), а у вихідному шарі – лінійна функція (**purelin**).

Після створення мережі розпочинається процес її навчання. Процес навчання та його результати наочно демонструються через діалогове вікно **Neural Network Training (nntraintool)**, яке показано на рис. 4.14. Це вікно є основним інтерфейсом для навчання нейронних мереж у **Neural Network Toolbox** систе-

ми MATLAB. Воно забезпечує потужні можливості для налаштування мережі, візуалізації процесу навчання та моніторингу динаміки зміни параметрів мережі під час тренування.

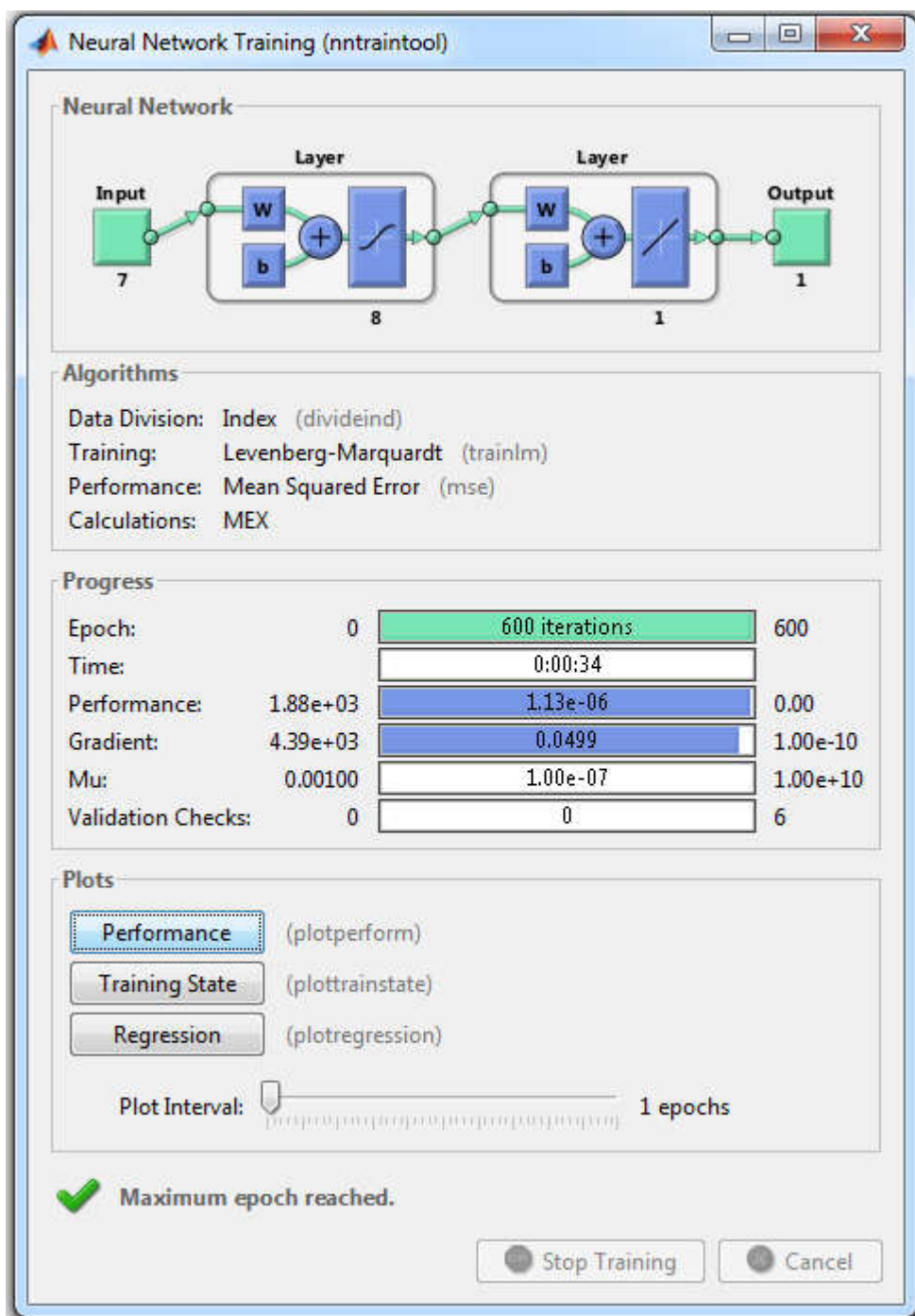


Рисунок 4.14 - Головне вікно для відображення процесу та результатів навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно **Neural Network Training (nntraintool)** у MATLAB складається з декількох функціональних секцій, кожна з яких відповідає за окремий аспект навчання нейронної мережі.

Секція Neural Network (нейронна мережа). У цій секції відображається структура згенерованої мережі, включаючи кількість шарів, нейронів у кожному шарі, а також типи активаційних функцій. Користувач може наочно бачити топологію мережі та взаємозв'язки між шарами, що допомагає оцінити її складність та потенційну здатність до моделювання системи.

Секція Algorithms (алгоритми навчання). Тут відображаються алгоритми, які використовуються для тренування мережі, а також параметри цих алгоритмів. Це дозволяє вибирати відповідний метод оптимізації для конкретного завдання та регулювати швидкість навчання, величину кроку, критерії зупинки та інші параметри, що впливають на процес адаптації ваг і зсувів мережі.

Data Division (поділ даних). Параметр Data Division: Index визначає, які приклади з набору даних будуть використані для навчання, валідації та тестування нейронної мережі. Цей механізм дозволяє контролювати, яка частина даних йде на формування навчальної моделі, а яка використовується для перевірки її здатності до узагальнення. Якщо дані не розподіляються на підмножини, рядок Data Division у вікні nntraintool не відображається.

Training (навчання). Ця секція відповідає за процес оновлення ваг і зсувів нейронної мережі під час тренування. Серед популярних алгоритмів навчання доступні Backpropagation, Levenberg-Marquardt, Bayesian Regularization та інші.

Вибір Training: Levenberg-Marquardt (trainlm) означає, що мережа буде навчатися за допомогою алгоритму Левенберга-Марквардта. Цей метод поєднує властивості методу найменших квадратів та методу Ньютона для ефективної оптимізації функції втрат. Його основна перевага полягає в швидкій мінімізації функції втрат і стабільному пошуку локального мінімуму через адаптивне онов-

влення вагових коефіцієнтів мережі. Алгоритм добре підходить для складних навчальних задач, де стандартні методи можуть показувати гірші результати.

Performance: Mean Squared Error (MSE). Параметр відображає показники точності навчання нейронної мережі на основі обраної функції втрат та даних для валідації. Цей показник дає змогу оцінити, наскільки ефективно мережа відтворює динаміку системи під час навчання, а також визначає момент, коли досягнуто оптимальної точності та процес тренування можна завершити. Моніторинг Performance допомагає уникнути перенавчання та забезпечує належну здатність моделі до узагальнення на нових даних.

Performance: Mean Squared Error (MSE) визначає використання середньоквадратичної помилки як критерію оцінки точності нейронної мережі під час її навчання. Середньоквадратична помилка є однією з найпоширеніших метрик у задачах регресії та моделювання. Вона розраховується як середнє значення квадратів різниці між прогнозованими виходами моделі (нейронної мережі) та відомими або фактичними значеннями з навчального набору даних.

Математично MSE можна записати так:

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

де N – кількість прикладів у наборі даних, y_{actual_i} – фактичне значення виходу, а $y_{predicted_i}$ – прогнозоване значення, яке видає нейронна мережа.

MSE показує середнє квадратичне відхилення між фактичними та передбаченими результатами. Чим менше значення MSE, тим точніше модель повторює поведінку системи. Під час навчання мережі оптимізаційний алгоритм прагне зменшити MSE, тобто підбирати ваги мережі так, щоб прогнозовані значення максимально відповідали реальним даним.

Calculations (обчислення). Секція Calculations відповідає за контроль кількості обчислень, виконаних під час процесу навчання. Вона важлива для оці-

нки продуктивності та часу, необхідного для завершення тренування нейронної мережі.

Параметр **Calculations: MEX** визначає, чи слід використовувати MEX-функції для прискорення обчислень у MATLAB. MEX (MATLAB Executable) дозволяє запускати оптимізований код, написаний на C, C++ або Fortran, безпосередньо у MATLAB, що забезпечує значно більшу швидкість виконання порівняно зі стандартним інтерпретованим кодом MATLAB.

У контексті навчання нейронних мереж використання MEX-функцій дозволяє швидше виконувати складні математичні обчислення, що особливо корисно при роботі з великими наборами даних або складними моделями. Завдяки MEX-файлам тренування мережі стає ефективнішим, зменшується час очікування результатів і підвищується продуктивність процесу.

Секція Progress (прогрес навчання). У цій секції відображається поточний стан навчання нейронної мережі в реальному часі. Вона надає користувачу можливість спостерігати за ходом навчання та оцінювати ефективність процесу. Інформація, що міститься у секції, допомагає контролювати навчання моделі, виявляти проблеми на ранніх стадіях та своєчасно вносити необхідні коригування для підвищення точності та стабільності моделі.

Epoch (епоха). Параметр Epoch означає повний прохід усіх прикладів навчального набору даних через нейронну мережу. Під час кожної епохи мережа отримує всі вхідні дані один раз, після чого відбувається оновлення її вагових коефіцієнтів та зсувів (biases) відповідно до обчислених помилок. Кількість епох визначає, скільки разів модель "пройде" через дані, що дозволяє поступово покращувати її здатність відтворювати динаміку системи. Відстеження епох дозволяє оцінювати, наскільки ефективно мережа навчається, і приймати рішення про зупинку або продовження тренування.

Time (час). Параметр Time показує фактичний час, витрачений на процес навчання нейронної мережі. Це допомагає контролювати тривалість навчання,

планувати ресурси та оцінювати продуктивність навчальної сесії, особливо при роботі з великими наборами даних або складними моделями.

Performance (ефективність/якість навчання). Показник Performance відображає, наскільки успішно нейронна мережа навчається на даних, використовуючи обрану функцію втрат (loss function). Для задач регресії найчастіше застосовується середньоквадратична помилка (Mean Squared Error, MSE), яка обчислює середнє значення квадратів різниці між прогнозованими виходами та реальними значеннями цільової змінної. Чим менше значення MSE, тим точніше мережа відтворює поведінку системи. Показник Performance дозволяє оцінити ефективність навчання на тренувальних та валідаційних даних і допомагає приймати рішення про необхідність коригування параметрів навчання.

Gradient (градієнт). Параметр Gradient демонструє вектор градієнта функції втрат відносно всіх параметрів нейронної мережі, тобто ваг і зсувів. Градієнт показує напрямок і величину змін, які слід застосувати до параметрів мережі, щоб мінімізувати функцію втрат. Під час навчання мережі використовується метод зворотного поширення помилки (backpropagation), і градієнт є ключовим елементом цього процесу. Він дозволяє алгоритму навчання ефективно коригувати ваги та зсуви, забезпечуючи поступове зменшення помилки і підвищення точності моделі.

Таким чином, секція Progress у nntraintool дозволяє детально контролювати процес навчання нейронної мережі, оцінювати якість навчання на кожному етапі та своєчасно реагувати на будь-які відхилення або проблеми, що виникають під час тренування.

Mu (тау-параметр). Параметр Mu відноситься до так званого тау-параметра, який застосовується в алгоритмах оптимізації для регулювання швидкості навчання нейронної мережі. Він особливо актуальний при використанні адаптивних методів оптимізації, таких як зворотне поширення помилки (backpropagation) або алгоритми на кшталт BFGS (Broyden-Fletcher-Goldfarb-Shanno). Основна функція Mu полягає у корекції величини кроку навчання відповідно до поточної динаміки процесу оптимізації. Це дозволяє алгоритму ефе-

ктивно знаходити мінімум функції втрат і забезпечує стабільну збіжність навіть при складних поверхнях втрат або великих градієнтах.

Значення μ зазвичай обчислюється автоматично алгоритмом на кожній ітерації навчання та може змінюватися в процесі, підлаштовуючись під поведінку мережі. Від правильного регулювання цього параметра залежить швидкість та стабільність навчання, а також здатність мережі до ефективного узагальнення даних.

Validation Checks (перевірки на валідаційному наборі). Параметр Validation Checks визначає кількість перевірок точності або якості нейронної мережі під час процесу навчання за допомогою окремого набору даних, який не бере участі у безпосередньому навчанні – так званого validation dataset. Основна мета цих перевірок – контроль за процесом навчання і запобігання перенавчанню (overfitting), коли модель «запам'ятовує» навчальні дані і втрачає здатність правильно реагувати на нові, невідомі приклади.

Наприклад, якщо встановлено 6 перевірок, нейронна мережа після кожних шести епох (де епоха – це повний прохід через весь навчальний набір даних) перевірятиме свою продуктивність на валідаційному наборі. Якщо точність моделі на цих даних перестає покращуватися або починає погіршуватися, це сигнал для зупинки навчання, оскільки подальші ітерації можуть призвести до перенавчання.

Таким чином, Validation Checks дозволяє автоматично відслідковувати ефективність мережі на даних, які вона раніше не бачила, і забезпечує своєчасне завершення навчання для досягнення оптимального балансу між підгонкою під тренувальні дані та здатністю узагальнювати на нові дані.

Секція Plots (графіки). Ця секція призначена для візуалізації ходу навчання нейронної мережі і дозволяє відслідковувати зміни її продуктивності у реальному часі. Візуальні графіки допомагають аналізувати ефективність навчання, виявляти проблеми, такі як перенавчання, та оцінювати динаміку зменшення помилки на різних етапах тренування.

Кнопка **Performance** відкриває вікно **Neural Network Training Performance** (рис. 4.15), в якому представлені графіки середньоквадратичної помилки (**Mean Squared Error, MSE**) для навчального, валідаційного та тестового наборів даних. Ці графіки демонструють, як змінюється точність мережі під час кожної ітерації навчання.

Графіки **MSE** дозволяють відстежувати, чи відбувається зменшення помилки на всіх наборах даних і чи немає ознак перенавчання, коли помилка на валідаційному та тестовому наборі починає збільшуватися. Основна мета – досягти мінімальної помилки на всіх наборах даних для отримання максимально точної та узагальнювальної моделі.

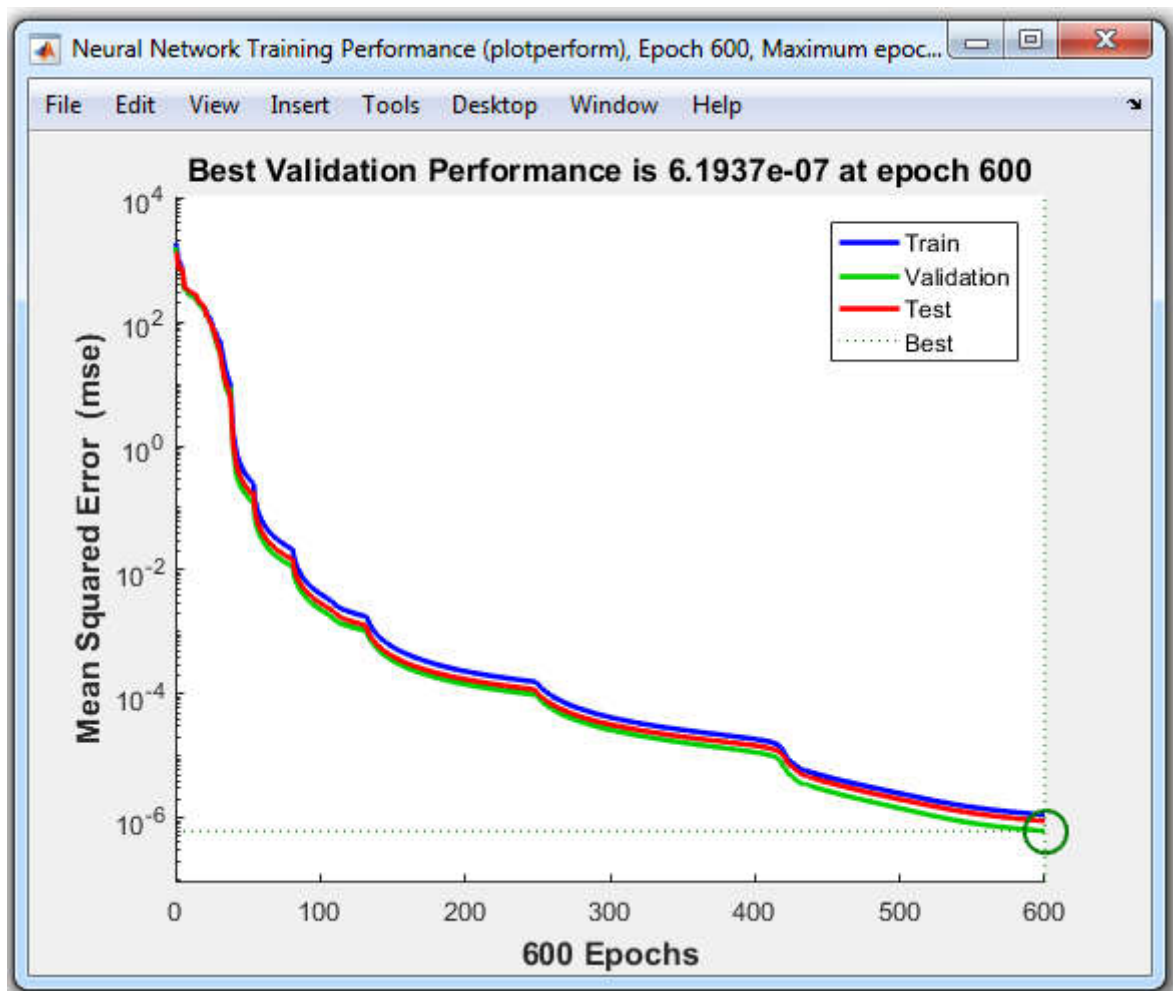


Рисунок 4.15 – Вікно контролю процесу навчання

Кнопка **Training State** відкриває вікно **Neural Network Training State** (рис. 4.16), яке відображає стан навчання нейронної мережі на конкретний момент часу. У цьому вікні можна переглядати актуальні значення таких параметрів, як кількість епох, поточна середньоквадратична помилка, градієнти, тау-параметр (μ) та інші показники. Це дозволяє детально відстежувати процес навчання та приймати рішення про його коригування, наприклад, про зупинку навчання у разі виникнення перенавчання або про необхідність зміни налаштувань алгоритму.

Таким чином, секція **Plots** є ключовим інструментом для аналізу навчального процесу, що забезпечує як візуальний контроль, так і можливість оперативно реагувати на будь-які відхилення у процесі тренування нейронної мережі.

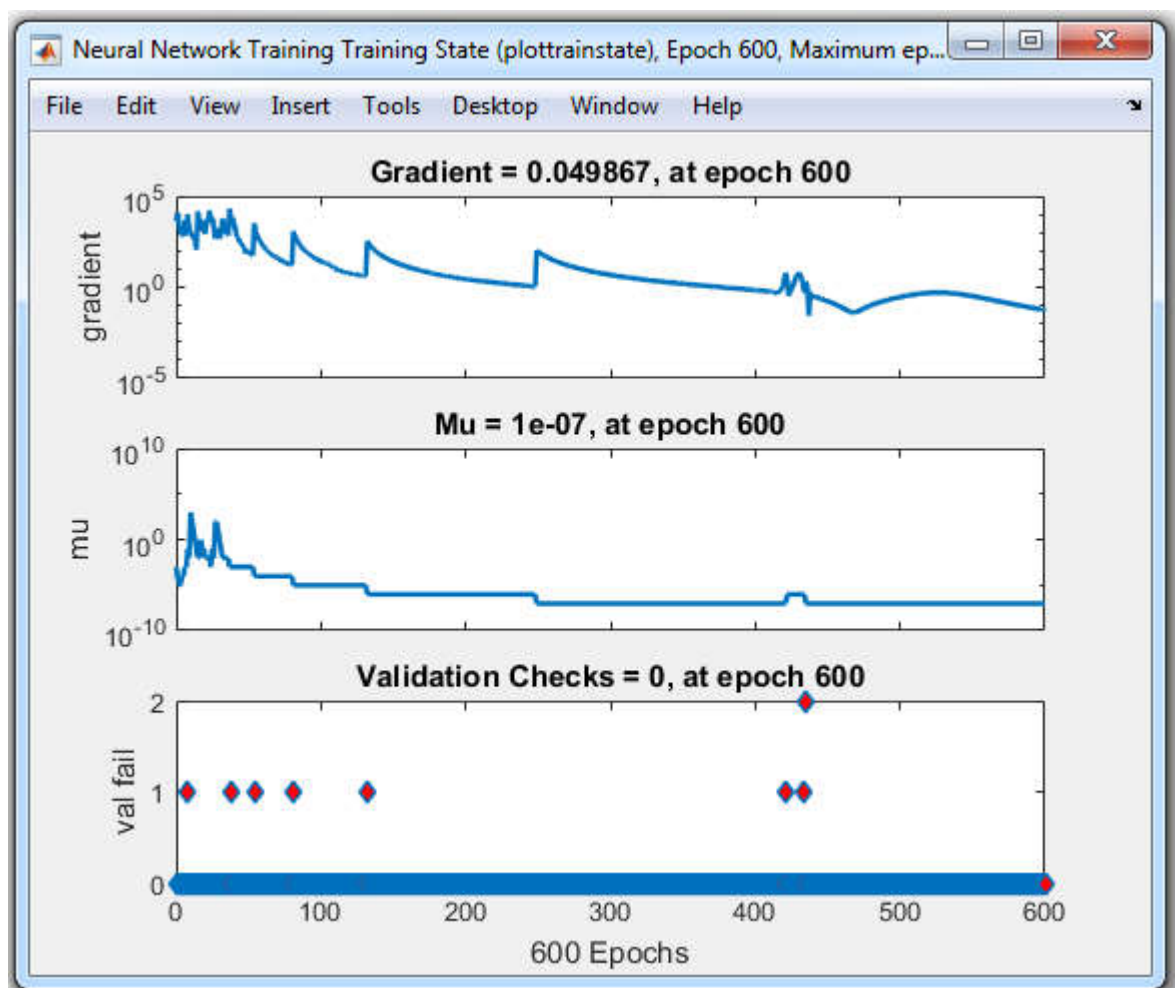


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні **Neural Network Training State** відображаються основні графіки, які дозволяють відстежувати динаміку навчання нейронної мережі в реальному часі. Це вікно дає змогу детально контролювати процес тренування та своєчасно реагувати на можливі проблеми.

Графік Gradient (градієнт). Цей графік демонструє зміну величини градієнта функції втрат під час навчання мережі. У контексті нейронних мереж градієнт є вектором, який вказує напрямок та швидкість найшвидшого зростання функції помилки. Використовуючи цю інформацію, оптимізаційний алгоритм коригує вагові коефіцієнти мережі в протилежному напрямку, щоб мінімізувати значення функції втрат.

На графіку Gradient можна побачити, як змінюється величина градієнта від початкових ітерацій до моменту, коли мережа наближається до оптимальних вагових коефіцієнтів. Спочатку градієнти можуть бути значними, але з часом вони зазвичай зменшуються, що свідчить про поступове зближення мережі до стабільного стану.

Моніторинг цього графіка допомагає виявляти проблеми gradient vanishing або gradient exploding та контролювати збіжність навчання. Надто великі або малі значення градієнта можуть сигналізувати про необхідність коригування параметрів навчання, таких як швидкість навчання або структура мережі.

Графік Mu (параметр навчання). Графік μ відображає зміну значення тау-параметра, який регулює швидкість навчання та рівень регуляризації мережі. У процесі тренування оптимізаційні алгоритми (наприклад, стохастичний градієнтний спуск або модифікації методу Бройдена-Флетчера-Гольдфарба-Шанно) використовують μ для адаптації швидкості корекції вагових коефіцієнтів.

Зміни параметра μ показують, як адаптується швидкість навчання до поточного стану мережі. Наприклад, збільшення μ може прискорити збіж-

ність, але водночас збільшити ризик нестабільності, тоді як зменшення ти може свідчити про уповільнення навчання або необхідність перегляду налаштувань оптимізатора. Регулярний контроль графіка ти дозволяє підтримувати баланс між швидкістю збіжності та стабільністю процесу навчання.

Графік Validation Failure (val fail). Цей графік демонструє зміну показника невдач на наборі валідаційних даних протягом навчання. Основне призначення валідаційного набору – оцінити здатність мережі узагальнювати знання на дані, які не використовувалися під час тренування, і запобігти перенавчанню.

Якщо кількість невдач на валідації зростає, це може вказувати на *overfitting*, тобто на надмірне підлаштування мережі під навчальні дані та втрату здатності правильно передбачати нові вхідні сигнали. У таких випадках доцільно застосувати ранню зупинку навчання, змінити гіперпараметри або додатково скоригувати структуру мережі. Мінімізація цього показника є однією з ключових цілей тренування, адже вона прямо впливає на точність та стабільність моделі в реальних умовах.

Таким чином, Neural Network Training State забезпечує комплексний моніторинг навчального процесу, дозволяючи спостерігати за градієнтами, швидкістю навчання та поведінкою мережі на валідаційних даних. Ця інформація є критично важливою для контролю якості навчання та своєчасного коригування параметрів для досягнення оптимальної моделі.

Продовження опису вікна Neural Network Training (nntraintool)

Кнопка **Regression (регресія)** призначена для налаштування параметрів навчання нейронної мережі у задачах регресії. Регресійне завдання – це один із типів задач машинного навчання, мета якого полягає у передбаченні числових значень або кількісних показників на основі вхідних даних. У таких задачах модель намагається відтворити функціональну залежність між вхідними ознаками (параметрами) та вихідними значеннями, які називаються цільовими або відгуком.

При натисканні на кнопку **Regression** відкривається вікно **Neural Network Training Regression** (рис. 4.17), яке надає графічне та числове представлення продуктивності моделі під час навчання. У цьому вікні користувач може спостерігати детальну інформацію щодо точності прогнозів мережі на різних наборах даних.

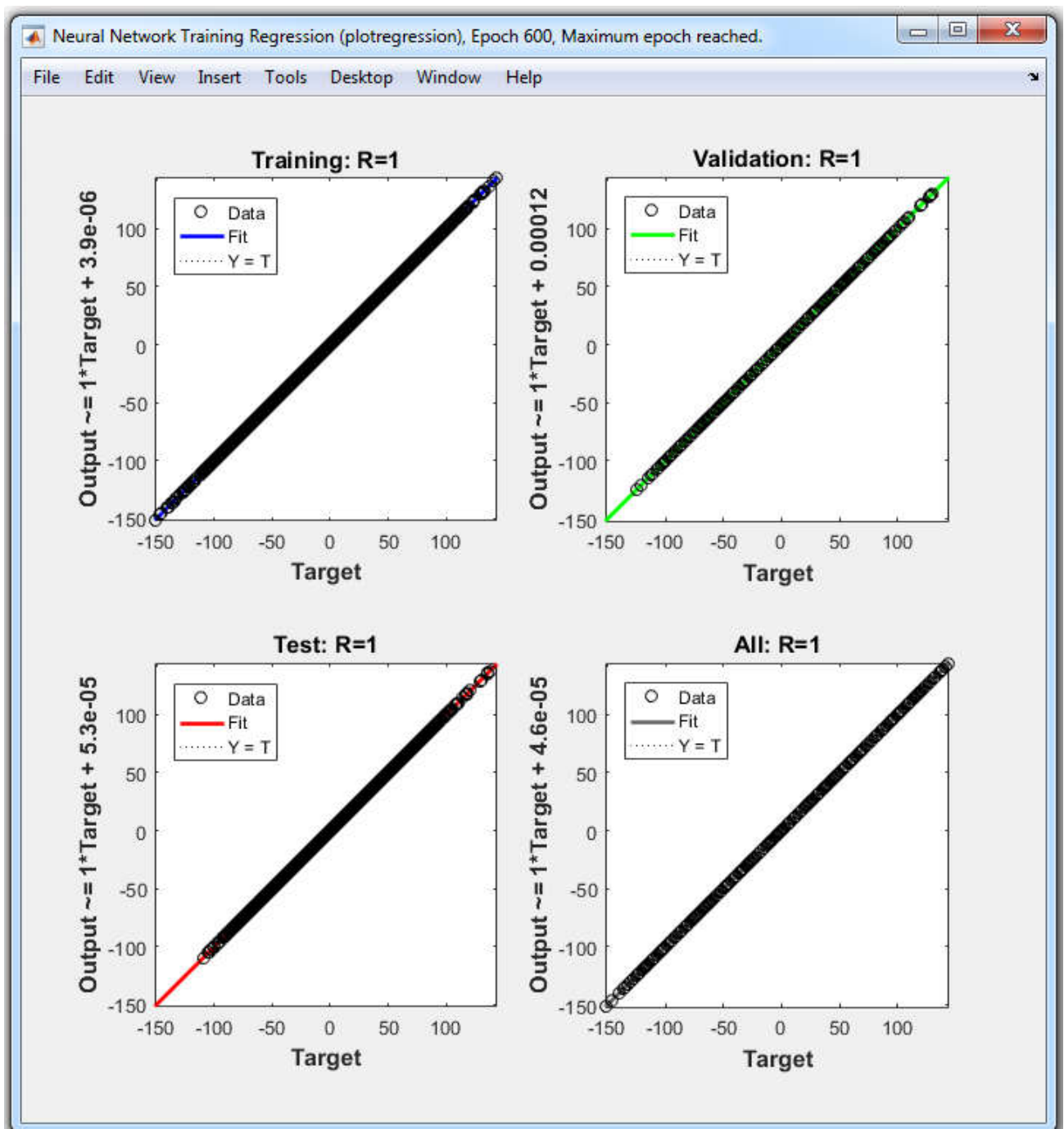


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис секцій вікна Neural Neural Network Training Regression

Training (навчання). Ця секція відображає, наскільки добре нейронна мережа відтворює дані навчального набору, тобто дані, на яких модель проходила тренування. Графіки показують відповідність прогнозованих значень фактичним даним, що дозволяє оцінити ефективність навчання та ступінь узгодження мережі з вихідними даними.

Графіки Validation (валідація). Валідаційні графіки демонструють точність роботи моделі на окремому наборі даних, який не використовувався під час навчання. Валідація дозволяє оцінити, наскільки добре модель узагальнює знання на нові дані, і служить для своєчасного виявлення ознак перенавчання (overfitting).

Графіки Test (тестування). Тестові графіки включають незалежний набір даних, який не був задіяний ні під час навчання, ні під час валідації. Вони забезпечують об'єктивну оцінку продуктивності нейронної мережі на абсолютно нових даних. Аналіз графіків Test дозволяє перевірити, наскільки модель здатна передбачати результати для непередбачених ситуацій.

Графіки All (всі дані разом). Ця секція об'єднує результати прогнозування для всіх трьох наборів даних: Training, Validation та Test. Графіки **All** показують одночасно прогнозовані та фактичні значення для кожного набору, що дозволяє наочно порівнювати продуктивність мережі на різних етапах навчання. Такий підхід допомагає швидко визначити, чи існують розбіжності між поведінкою моделі на навчальних, перевіірочних і тестових даних, та робить оцінку її стабільності більш наочною.

Таким чином, вікно **Neural Network Training Regression** забезпечує комплексний аналіз регресійних результатів моделі, дозволяє відстежувати точність прогнозів на всіх основних наборах даних та вчасно коригувати параметри нейронної мережі для досягнення оптимальної продуктивності.

Графіки Data та Fit у вікні Neural Network Training Regression. У вікні Neural Network Training Regression графіки Data для наборів Training, Validation,

Test та All демонструють, наскільки прогнозовані значення моделі нейронної мережі узгоджуються з фактичними даними для кожного з наборів. На горизонтальній осі відкладаються фактичні значення з відповідного набору даних, тоді як на вертикальній осі – прогнозовані значення, отримані від моделі. Кожна точка на графіку відображає відповідність прогнозованого і справжнього значення: координата по осі X відповідає бажаному значенню, а координата по осі Y – вихідному значенню мережі. У середовищі MATLAB точки позначаються кружечками ().

Якщо всі точки розташовані близько до діагональної лінії $Y = T$, це свідчить про високу точність моделі. Велика розсіюваність точок може вказувати на наявність шуму в даних або недостатню адекватність моделі. Графіки Data дозволяють наочно оцінити відповідність моделі даним на різних наборах і виявити можливі аномалії чи області для поліпшення моделі.

Графіки **Fit** представляють результати лінійної регресії, що демонструють прогнозовані значення, які генерує нейронна мережа на основі вхідних даних. Вони відображають, наскільки модель узгоджується з фактичними значеннями, і служать для візуальної оцінки точності прогнозів. Якщо лінія Fit проходить близько до більшості точок, це вказує на високу продуктивність моделі. Відхилення лінії від фактичних значень може сигналізувати про потребу в додатковій оптимізації або корекції гіперпараметрів нейронної мережі.

У вікні Neural Network Training Regression також відображаються коефіцієнти кореляції та рівняння лінійної регресії для кожного набору даних (Training, Validation, Test та All). Вони використовуються для кількісної оцінки точності моделі.

Коефіцієнт кореляції R визначає ступінь лінійної залежності між прогнозованими та фактичними значеннями:

- $R = 1$ – ідеальна лінійна залежність, модель точно передбачає всі значення;
- $R = 0$ – відсутність лінійної залежності, модель не здатна передбачити дані;

- $R = -1$ – повна зворотна лінійна залежність, модель передбачає значення у протилежному напрямку.

Коефіцієнт кореляції, близький до 1, свідчить про високу точність регресійної моделі, тоді як значення близько до 0 або від’ємне сигналізує про низьку точність прогнозування. Використання R разом із графіками Data та Fit дозволяє комплексно оцінити ефективність моделі нейронної мережі на всіх етапах навчання та перевірки, а також визначити можливі напрями для її вдосконалення.

Продовження опису вікна Neural Network Training (nntraintool)

Plot Interval (інтервал графіка). Параметр Plot Interval визначає, як часто під час навчання нейронної мережі для задач регресії оновлюються графіки та відображаються проміжні результати. Іншими словами, він встановлює кількість ітерацій або епох, після яких відбувається оновлення інформації на графіках.

Правильне налаштування Plot Interval дозволяє ефективно контролювати обсяг відображуваної інформації під час навчання, особливо коли кількість епох велика. Менше значення Plot Interval забезпечує частіше оновлення графіків, що дає змогу уважніше аналізувати процес навчання, але одночасно збільшує обсяг візуальної інформації. Більше значення, навпаки, зменшує частоту оновлення, спрощуючи відстеження загальної тенденції навчання без перевантаження графіків.

Налаштування цього параметра є важливою частиною контролю процесу навчання, оскільки дозволяє оцінювати вплив змін параметрів мережі на результати моделі в режимі реального часу.

Кнопки Stop Training і Cancel. У вікні **Neural Network Training (nntraintool)** середовища MATLAB присутні дві кнопки для управління процесом навчання, які мають різні функції:

- **Stop Training (зупинка навчання)** – негайно припиняє процес навчання нейронної мережі без можливості його продовження. Ця кнопка викори-

стовується, коли навчання виявляється занадто тривалим, не дає задовільних результатів або потрібно терміново припинити тренування.

- **Cancel (скасувати)** – дозволяє зупинити навчання, але при цьому зазвичай передбачає підтвердження дії та збереження поточного стану мережі. Використання кнопки Cancel дає змогу зупинити навчання після перегляду проміжних результатів, внести необхідні зміни до параметрів та при необхідності продовжити навчання пізніше.

Після завершення навчання результати відображаються у вікні **Training Data for NN Predictive Control**, на якому показані графіки фактичних та прогнозованих значень, що дозволяє оцінити точність моделі та якість її навчання (рис. 4.18).

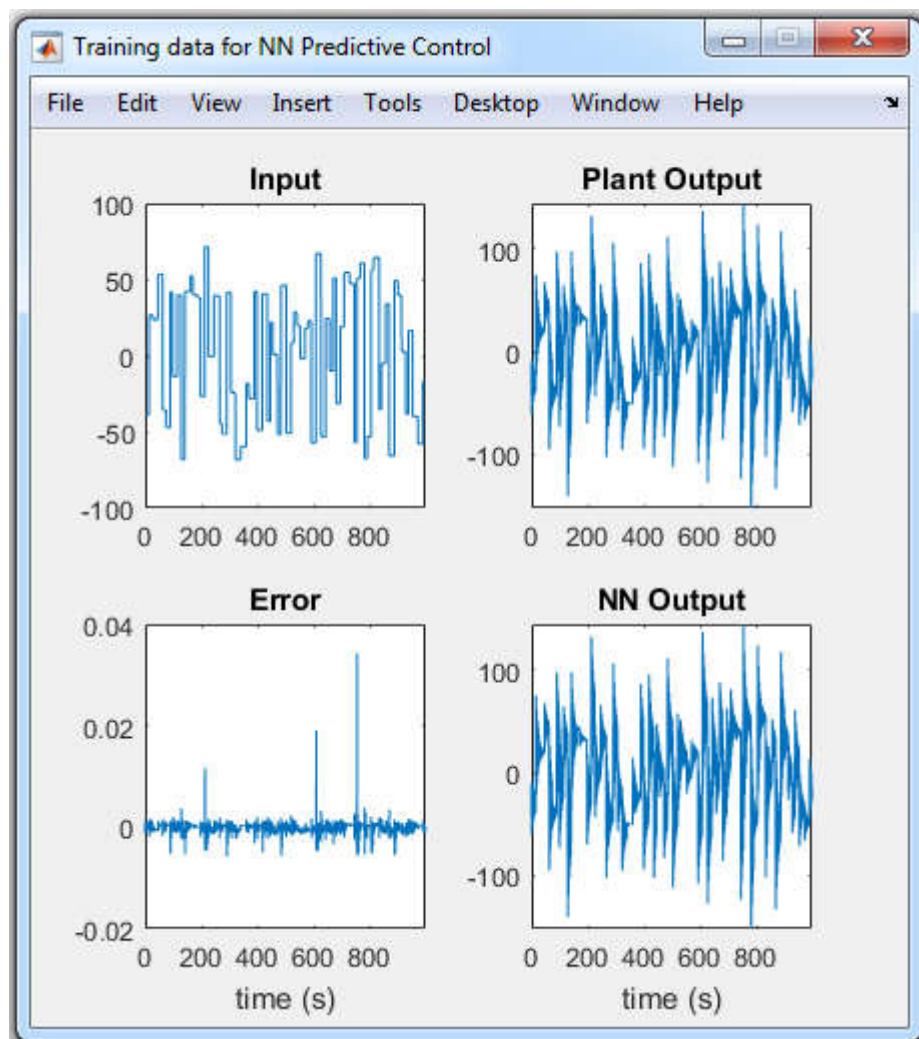


Рисунок 4.18 – Графічне представлення результатів навчання нейронної мережі

Onuc вікна Training Data for NN Predictive Control

У вікні **Training Data for NN Predictive Controller** відображаються графіки, що ілюструють процес навчання нейронної мережі та взаємодію моделі з системою:

- **Input (вхідні сигнали)** – графік Input показує вхідні сигнали, які подаються на систему та нейронну мережу під час навчання. Ці сигнали стимулюють модель для генерації відповідей і дозволяють досліджувати її реакцію на різні умови.

- **Plant Output (вихід системи)** – графік Plant Output демонструє фактичні вихідні дані системи (Plant) під час навчання. Він відображає реальну динаміку системи у відповідь на вхідні сигнали, що дозволяє порівняти її поведінку з прогнозами моделі.

- **NN Output (вихід нейромережі)** – графік NN Output відображає прогнозовані вихідні сигнали, згенеровані нейронною мережею під час навчання. Ці дані ілюструють здатність моделі відтворювати поведінку системи на основі поданих вхідних сигналів.

- **Error (помилка)** – графік Error показує різницю між прогнозованими значеннями нейронної мережі (NN Output) та фактичними виходами системи (Plant Output). Цей показник відображає точність моделі та дозволяє оцінити, наскільки адекватно мережа відтворює динаміку системи.

Аналогічні графіки відображаються у вікнах **Validation Data for NN Predictive Control** та **Testing Data for NN Predictive Control** (рис. 4.19, 4.20), які ілюструють результати перевірки моделі на контрольній та тестовій множині даних. Ці графіки дозволяють оцінити здатність нейронної мережі узагальнювати знання, отримані під час навчання, на нові дані.

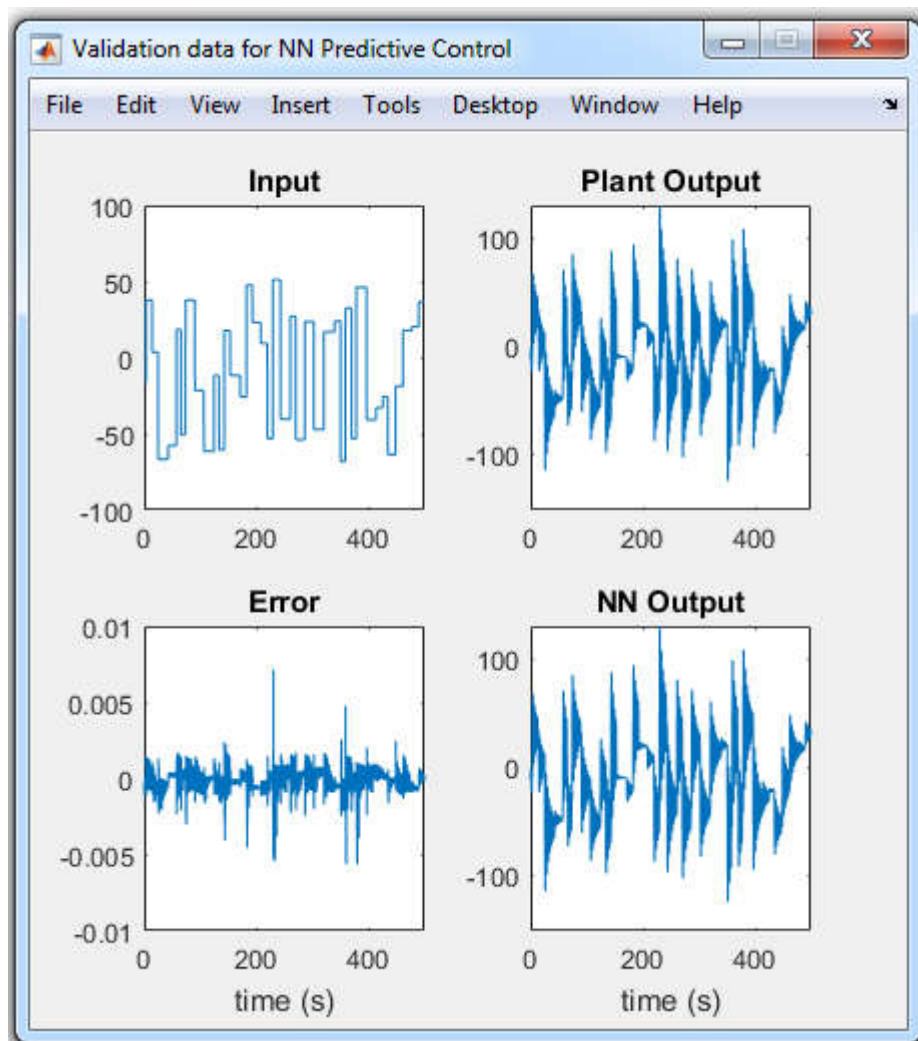


Рисунок 4.19 – Візуалізація результатів перевірки нейронної мережі на контрольній множині

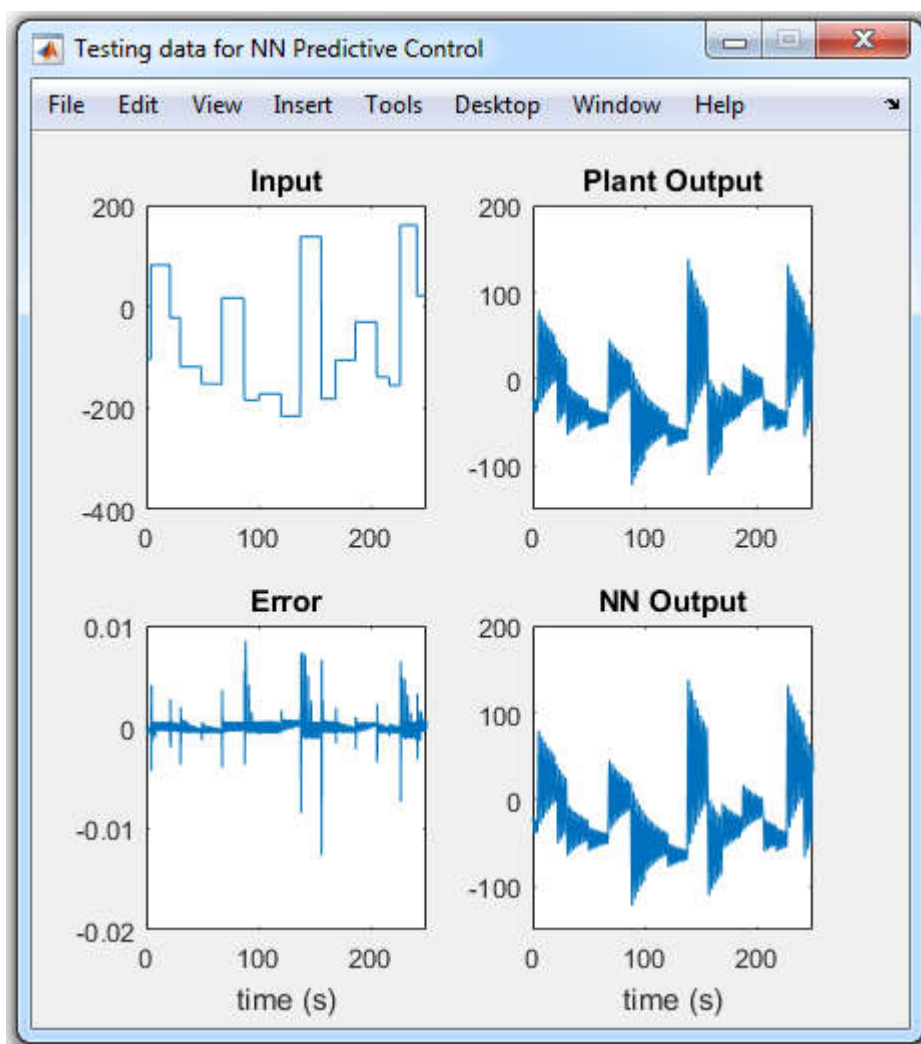


Рисунок 4.20 – Візуалізація результатів перевірки нейронної мережі на тестовій множині

Продовження опису вікна *Plant Identification*.

Після завершення процесу навчання у вікні **Plant Identification** з'являється повідомлення: *"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 результати, обравши кнопки **OK** або **Apply**).

На цьому етапі М-функція **Nident** формує динамічну нейромережу **netn2** із заданою кількістю затримок на вході та виході моделі. При цьому попередньо набуті значення вагів і зміщень нейронів зберігаються. Елементи динамічної мережі показані на рис. 4.21.

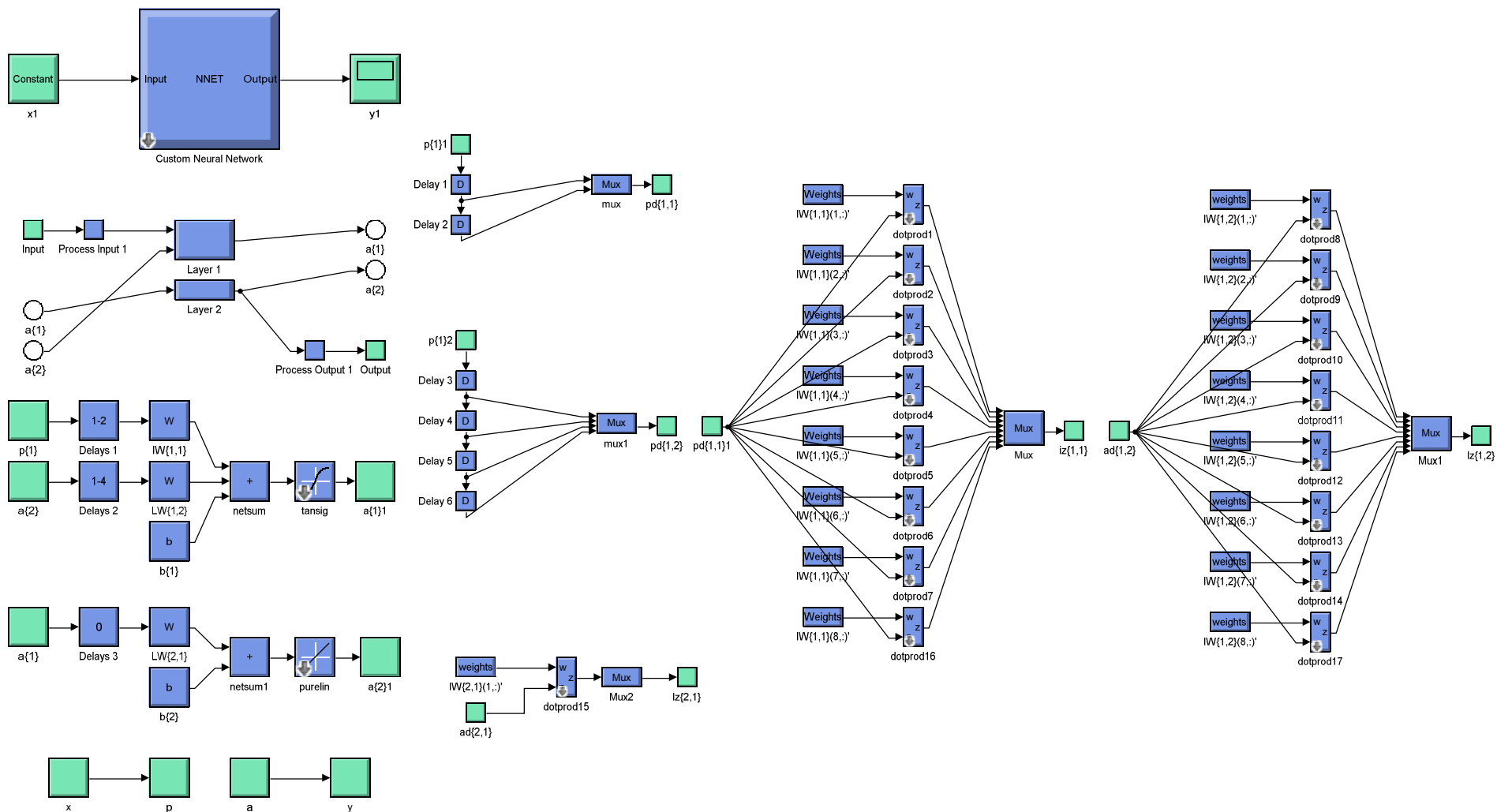


Рисунок 4.21 – Моделі елементів динамічної мережі з блоками затримок у Simulink

Модель динамічної мережі створюється у Simulink за допомогою команди **gensim(netn2)**. Кожен новий елемент стає доступним у окремому вікні після подвійного клацання на попередньому. На основі цих елементів формується повна схема динамічної мережі, показана на рис. 4.22.

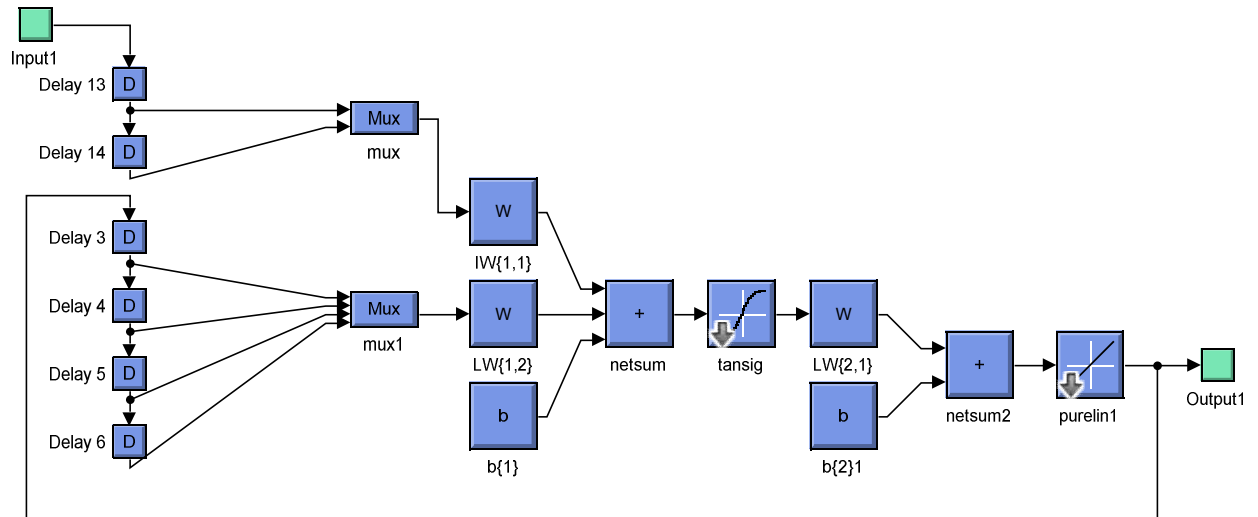


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

Для успішної побудови моделей **netn** і **netn2** необхідно встановити **Breakpoint** у відповідному місці М-функції **Nnident.m**, оскільки після завершення виконання цієї функції подальше формування мереж стає неможливим.

Параметри нейромережевої моделі керованого об'єкта передаються до блоку **NN Predictive Controller** у Simulink. Мережа відображається як структурна схема, показана на рис. 4.23.

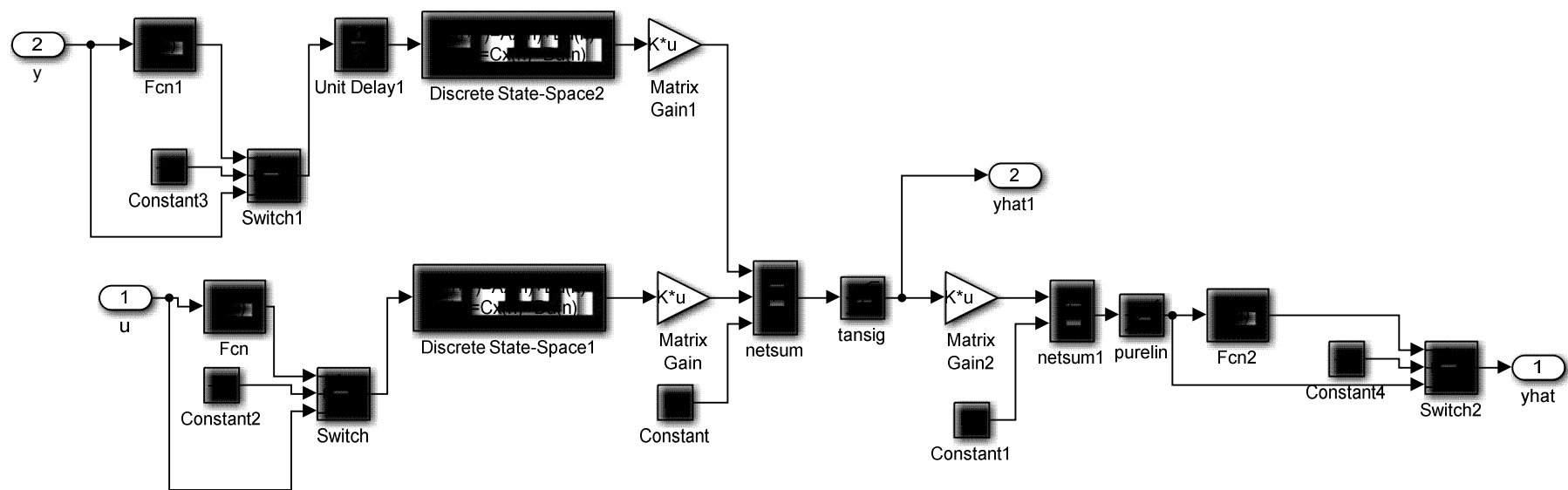


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

Блоки Matrix Gain і Matrix Gain1 відповідають матрицям $IW\{1,1\}$ та $LW\{1,2\}$; блоки Constant (B1) і Constant1 (B2) – це зсуви нейронів першого та другого шарів; елементи затримок реалізовані блоками Discrete State Space 1 та Discrete State Space 2.

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

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

Візьмемо, для прикладу, $N_i = 2$ і $N_j = 5$. МВ цьому випадку матриці $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$ і $\mathbf{D}$ мають наступний вид.

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

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

Також у Simulink формується схема **pctest3sim2** (рис. 4.24), яка містить додаткові виходи та використовується М-функцією **predopt** для прогнозування процесу у майбутньому. При активації блоку **Subsystem** відкривається детальна схема, показана на рис. 4.25.

Об'єднана структурна схема нейрорегулятора, що поєднує моделі з рис. 4.6 і рис. 4.23, представлена на рис. 4.26.

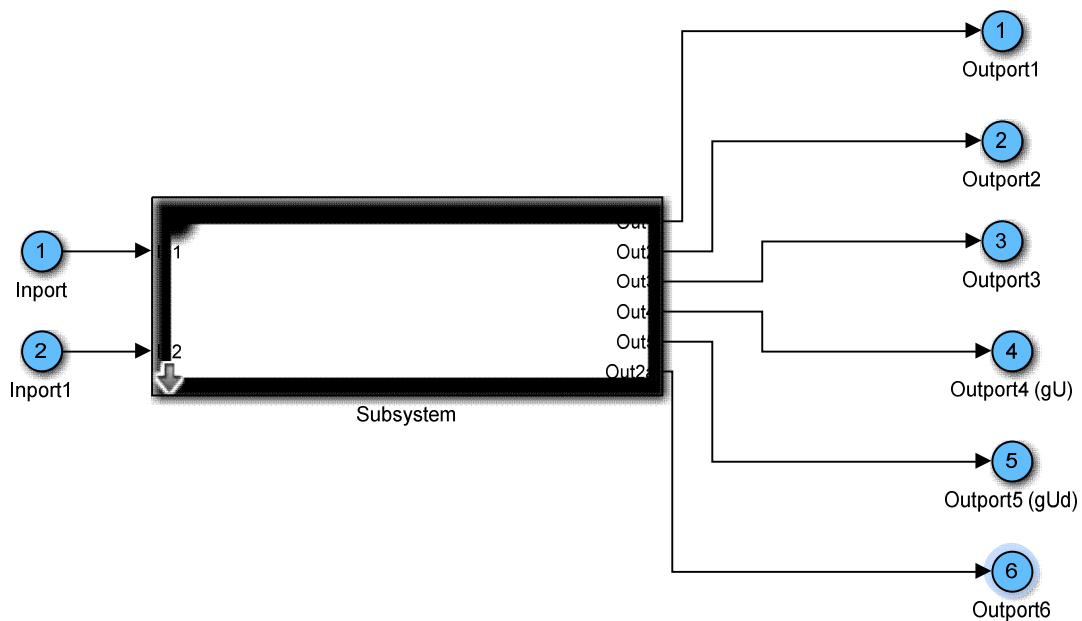


Рисунок 4.24 – Схема ptest3sim2, що використовується для прогнозування процесу функцією predopt

Після формування нейромережевої моделі керованого об'єкта користувач повертається до вікна **Neural Network Predictive Controller** (рис. 4.7), де здійснюються налаштування параметрів оптимізації.

Продовження опису вікна Neural Network Predictive Controller

Const Horizon (N2) – верхня межа підсумовування у функції якості (нижня межа фіксована і дорівнює 1). Цей параметр визначає горизонт прогнозу, тобто кількість кроків у майбутньому, на які алгоритм NNPC робить передбачення.

Збільшення Const Horizon (N2) дозволяє отримати більш далекі випереджальні прогнози, але водночас підвищує обчислювальну складність системи. Вибір оптимального значення цього параметра залежить від специфіки керованого об'єкта та вимог до точності прогнозу.

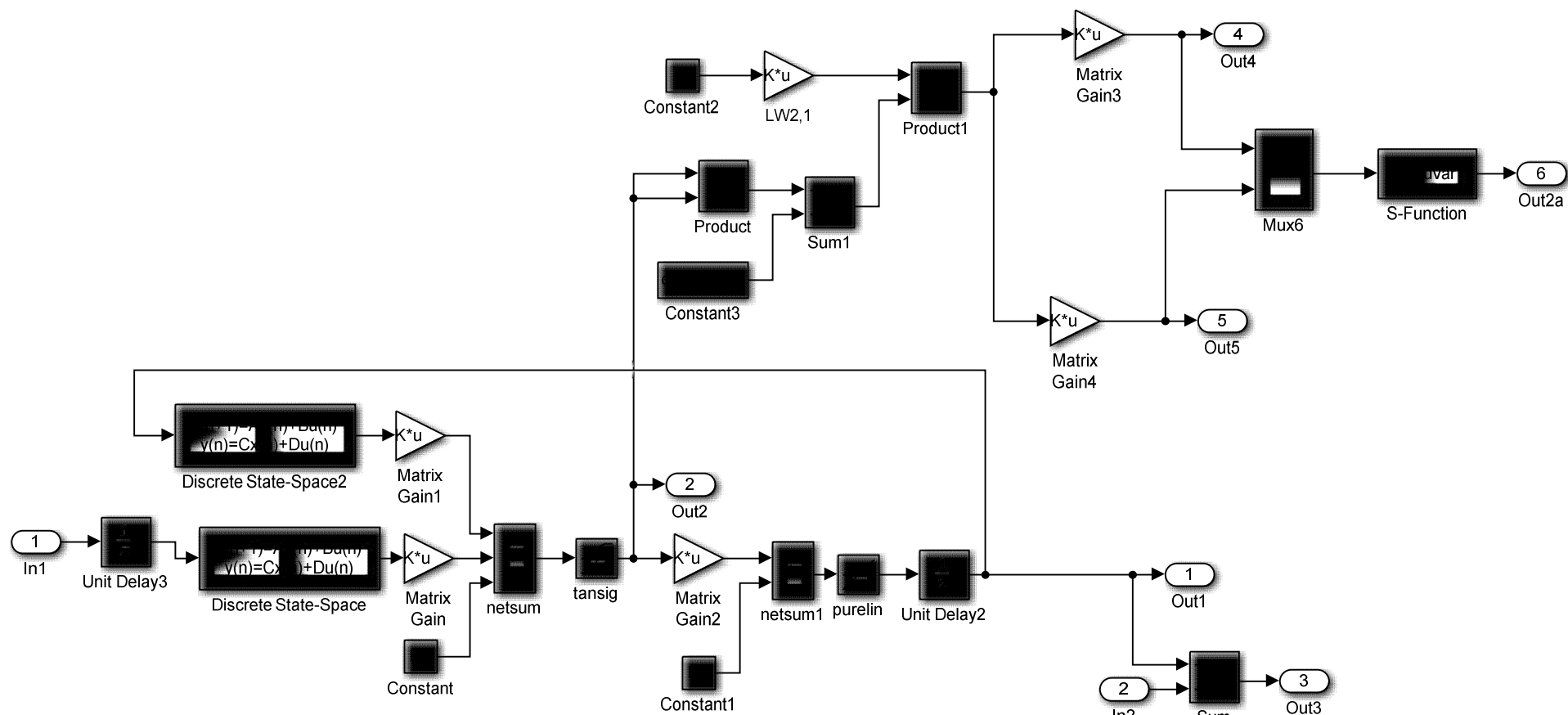


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

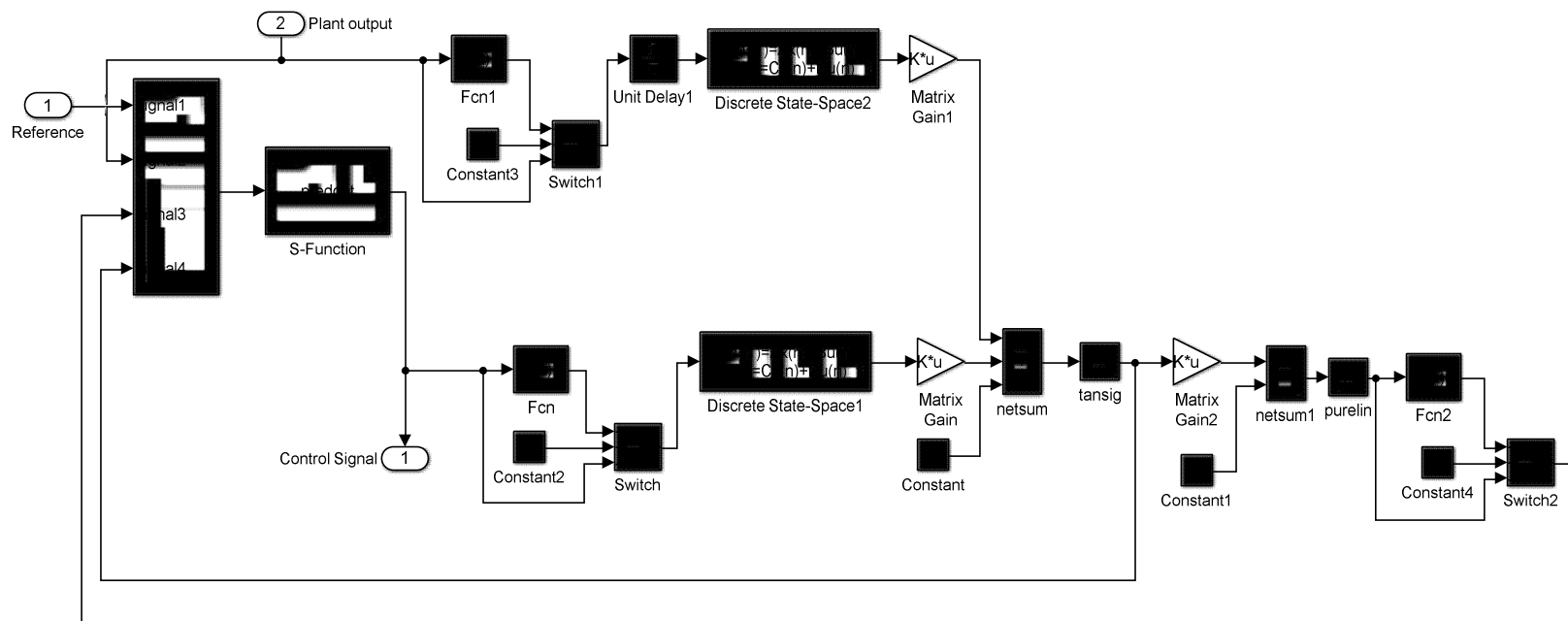


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

Control Horizon (Nu). Параметр **Control Horizon (Nu)** визначає верхню межу підсумовування при оцінці потужності керування. Він безпосередньо впливає на алгоритм прогнозного керування, визначаючи, на скільки кроків у майбутньому контролер буде розраховувати керуючі дії.

Основна ідея цього параметра полягає в тому, що збільшення **Control Horizon** дозволяє алгоритму оптимального керування враховувати більш віддалені майбутні стани системи, що сприяє точнішому досягненню цільових значень та підтриманню стабільності системи. Однак занадто велике значення **Nu** може призвести до підвищення обчислювального навантаження та збільшення вимог до ресурсів комп'ютера. Тому при виборі цього параметра слід враховувати характеристики системи, складність задачі та необхідний рівень прогнозу.

Control Weighting Factor ρ (Коефіцієнт ваги керування). **Control Weighting Factor** визначає вагу або важливість мінімізації керуючих сигналів у функції витрат контролера. Іншими словами, цей параметр дозволяє налаштувати, наскільки сильно контролер буде прагнути робити керуючі дії економічними або «дешевими» щодо витрат на управління.

Якщо коефіцієнт ρ встановлено великим, контролер надає пріоритет зменшенню амплітуди керуючих сигналів, що може бути корисно у випадках, коли управління дороге коштує або є фізичні обмеження. Зменшення значення коефіцієнта ρ призводить до того, що контролер більше концентрується на досягненні цільових значень системи, а економія керуючих сигналів стає менш пріоритетною.

Налаштування **Control Weighting Factor** дозволяє знайти оптимальний баланс між ефективністю керування та вартістю керуючих дій, що особливо важливо при застосуванні нейромережевого прогнозного контролю, де одночасно враховуються точність прогнозу і обмеження на ресурси управління.

Search Parameter α (Параметр пошуку). **Search parameter** – це параметр одновимірної пошуку, який задає поріг допустимого зменшення показника якості. Він використовується для контролю процесу оптимізації та визначає момент завершення пошуку, коли подальше покращення якості стає незначним.

Одновимірний пошук – це метод оптимізації, при якому оптимізується лише один параметр, тоді як інші залишаються фіксованими. У контексті нейромережевого прогнозного контролера цей метод допомагає налаштувати окремі параметри контрольної стратегії, що безпосередньо впливають на ефективність керування.

Поріг зменшення показника якості визначає мінімальне значення, яке повинен мати показник покращення, щоб вважати процес оптимізації завершеним. Коли приріст або зменшення показника якості стає меншим за цей поріг, алгоритм припиняє пошук, а отримані параметри вважаються оптимальними. Такий підхід дозволяє заощадити обчислювальні ресурси та прискорити збіжність оптимізації.

Minimization Routine (Процедура мінімізації). Minimization Routine визначає алгоритм оптимізації, який використовується для налаштування параметрів контрольної стратегії. Вибір конкретного методу оптимізації має значний вплив на точність і ефективність роботи контролера.

При виборі процедури враховуються особливості завдання керування, наявні обмеження, а також доступні обчислювальні ресурси. Ретельний підбір алгоритму дозволяє досягти оптимального балансу між точністю розрахунку керуючих сигналів і швидкістю обчислень.

Iterations Per Sample Time (Ітерації на крок часу). Параметр Iterations Per Sample Time визначає кількість ітерацій оптимізації, які виконуються протягом одного кроку часу системи (Sample Time). Він впливає на точність розрахунку оптимального керування та швидкість реакції контролера на зміни стану системи.

Збільшення кількості ітерацій дозволяє отримати більш точний розрахунок керуючих дій, проте підвищує обчислювальне навантаження. Навпаки, занадто мала кількість ітерацій може призвести до недостатньої точності оптимізації та втрати ефективності керування.

Оптимальне значення цього параметра зазвичай підбирається експериментально, виходячи з характеристик конкретної системи керування та обмежень

обчислювальних ресурсів. Воно забезпечує баланс між точністю оптимізації та швидкістю реакції контролера.

Після налаштування всіх параметрів оптимізації вони передаються в блок **NN Predictive Controller** у середовищі Simulink, де використовуються для розрахунку оптимальних керуючих сигналів під час роботи системи.

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 унеможливило 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 вибірки ε знаходяться 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: $t_{\min} = 4 \div 5 \text{ c}$, $t_{\max} = 10 \div 20 \text{ c}$.

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.

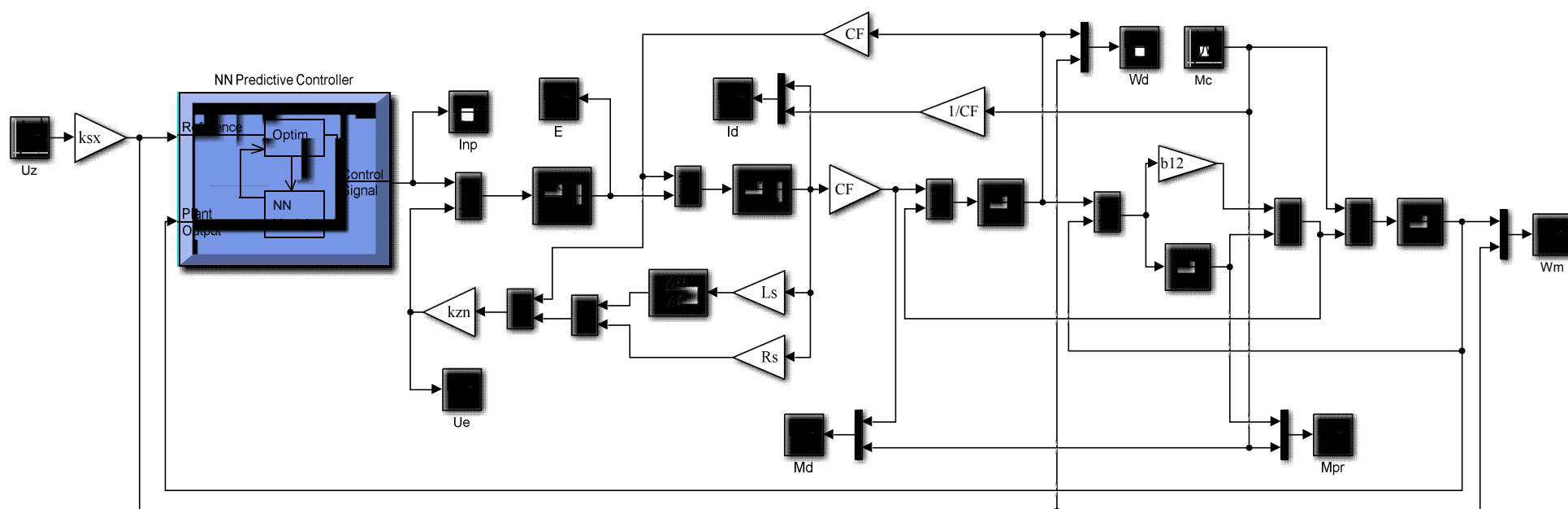


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

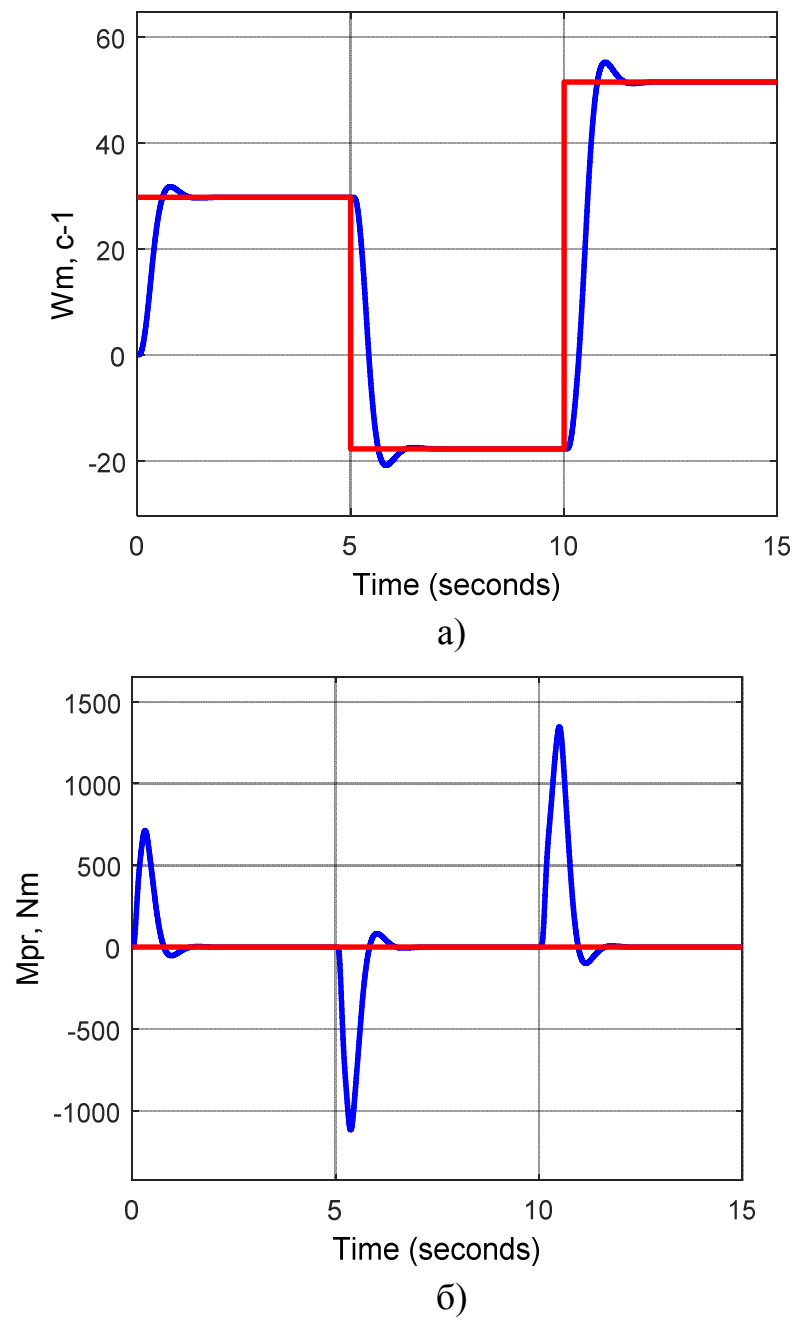
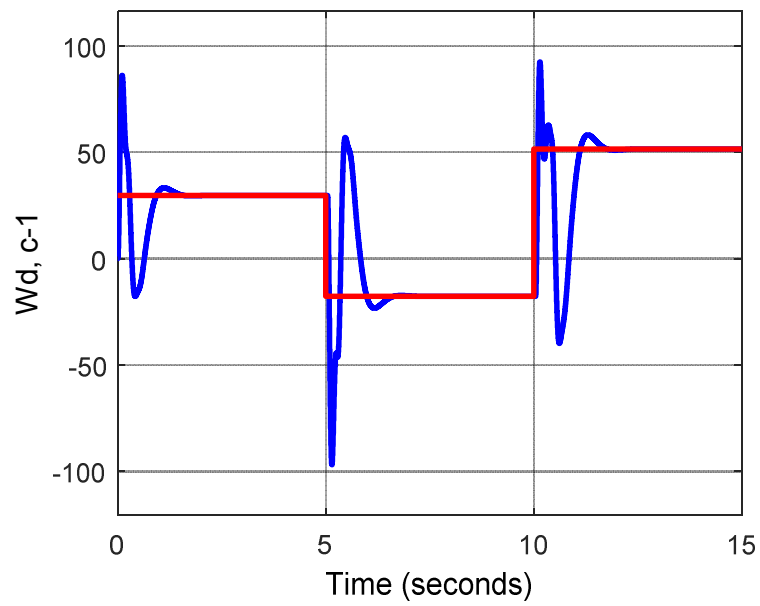
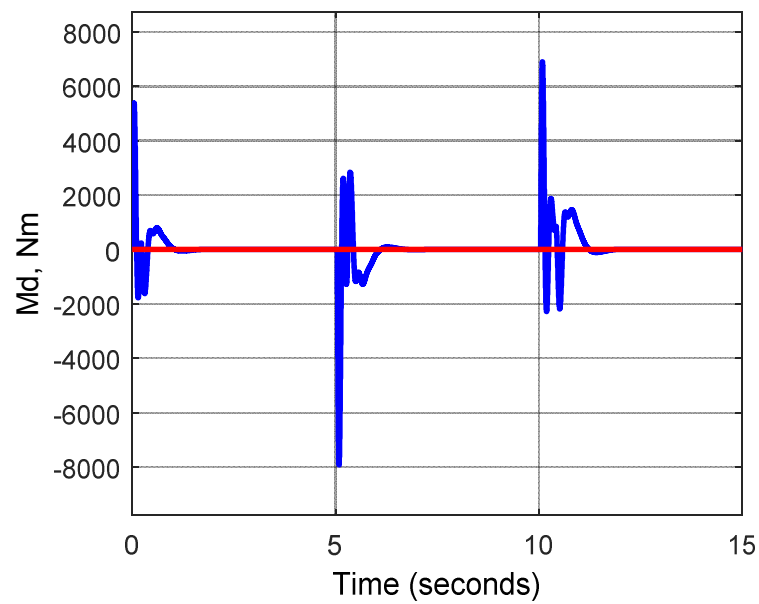


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором п по задаючій дії



a)

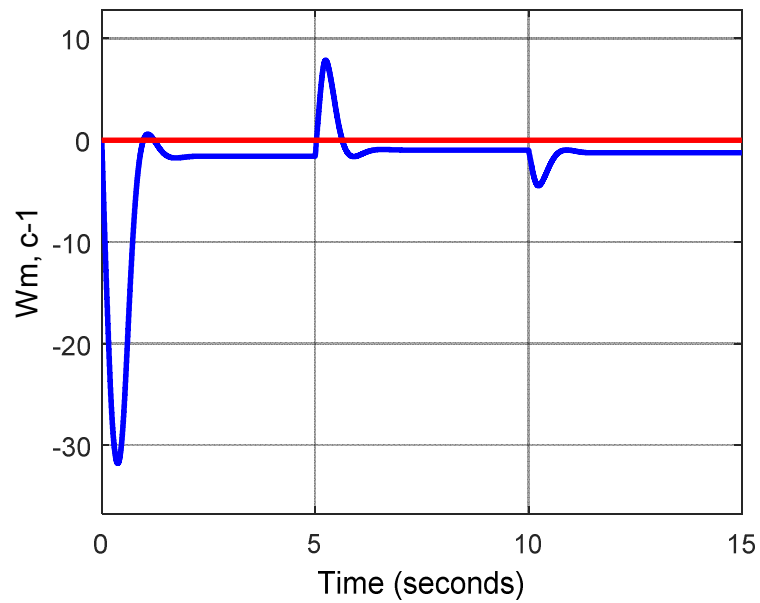


б)

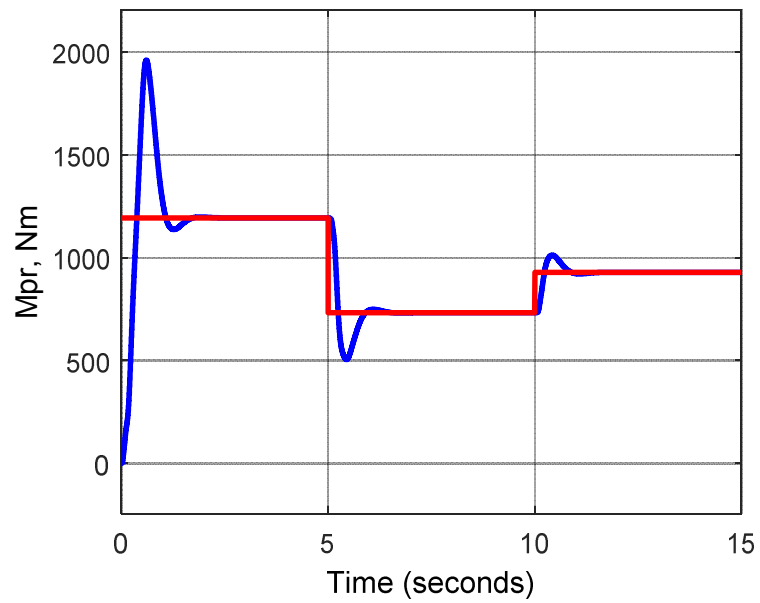
Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$

(а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи

з нейрорегулятором по задаючій дії

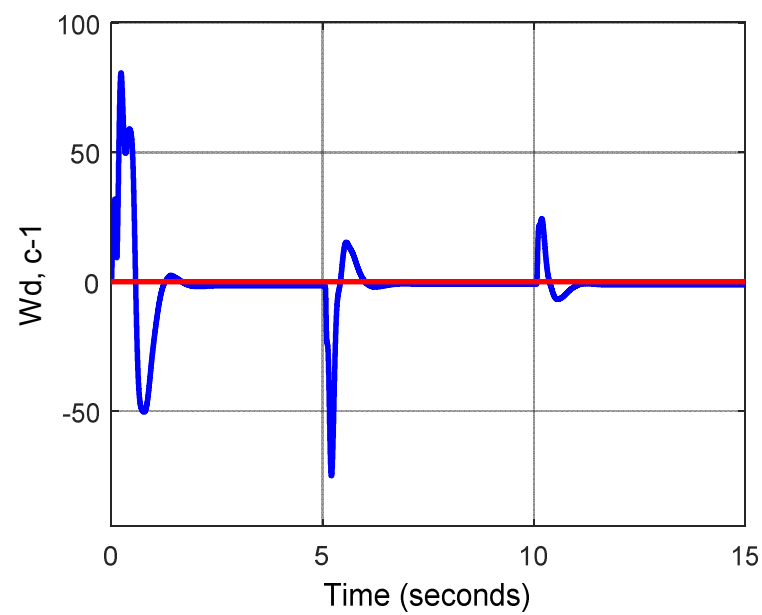


a)

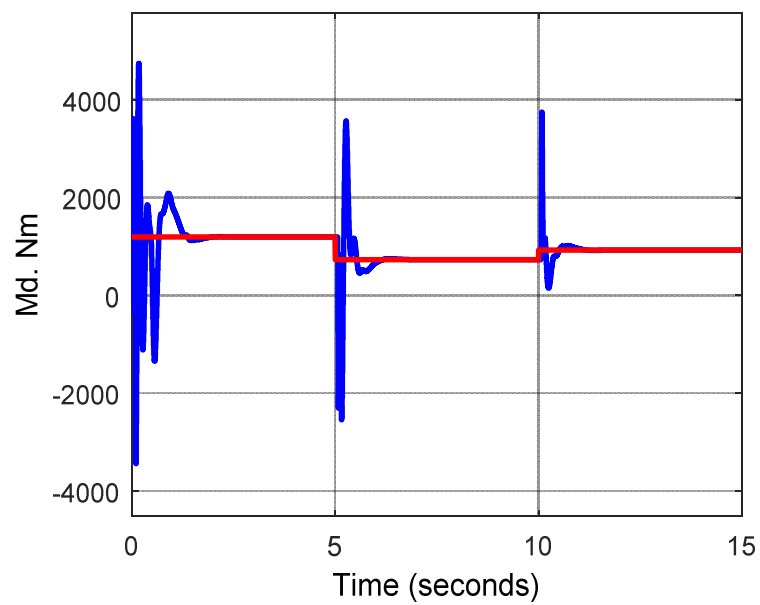


б)

Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (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

2. У четвертому розділі здійснено детальний огляд підходів до побудови нейронних регуляторів, що реалізуються засобами пакета Neural Network Toolbox у середовищі MATLAB. Розглянуто три різні варіанти нейрокерування: прогнозний регулятор (NN Predictive Controller), регулятор, побудований на основі авторегресійної моделі з ковзним середнім NARMA-L2, а також регулятор, який працює за принципом наслідування еталонної моделі (Model Reference Controller). Проведений порівняльний аналіз показав, що саме прогнозний нейрорегулятор NN Predictive Controller найбільш придатний для системи керування поворотом роторного екскаватора, що й зумовило його вибір для подальших досліджень. У роботі докладно описано архітектуру цього регулятора та послідовність дій, необхідних для його синтезу.

2. На базі середовища SIMULINK створено повноцінну модель системи автоматичного керування з використанням нейрорегулятора NN Predictive Controller. До її складу входить модель об'єкта керування – двомасової системи електроприводу механізму розвороту, а також інтегрований блок нейронного регулятора, що формує керуючі дії відповідно до прогнозованої поведінки виконавчої частини.

3. На основі численних експериментів із параметрами навчальної вибірки та за допомогою моделі виконавчого механізму сформовано навчальну послідовність, яка забезпечує максимально точне навчання нейромережі. Було визначено оптимальні значення таких параметрів, як загальна кількість елементів навчання, інтервал між сусідніми моментами зчитування даних, мінімальний та максимальний часові інтервали для ідентифікації системи. Сформована навчальна вибірка забезпечила можливість високоточної апроксимації динаміки об'єкта керування.

4. Розроблено двошарову нейронну мережу прямого поширення сигналу, що була навчена за допомогою алгоритму Левенберга–Марквардта. У процесі оптимізації встановлено кількість нейронів прихованого шару, яка забезпечує

найкращу точність моделі без надмірного збільшення її складності. Отримана модель характеризується низьким рівнем помилки: середньоквадратична помилка має порядок , а миттєві відхилення на навчальній, валідаційній та тестовій вибірках не перевищують 0,02, що свідчить про високу якість ідентифікації.

5. Виконано синтез прогностного нейрорегулятора NN Predictive Controller. Під час дослідження впливу параметрів регулятора на поведінку системи було встановлено, які з них відіграють ключову роль у формуванні якісного перехідного процесу. Після оптимізації параметрів отримано структуру регулятора, що забезпечує високу точність регулювання за рахунок використання нейромережевої моделі об'єкта з високим рівнем адекватності.

6. Проведене моделювання роботи системи з інтегрованим нейронним регулятором підтвердило ефективність синтезованого підходу. Під час перевірки в режимах запуску та при накладенні збурюючих моментів перехідні процеси продемонстрували стабільність і високу якість: система зберігає властивість астатичності, величина перерегулювання не перевищує 5 %, а тривалість процесу регулювання становить приблизно 2 секунди. Отримані характеристики повністю відповідають технічним вимогам, що висувалися до системи керування.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розв'язано важливу науково-прикладну проблему створення нейромережевої системи керування багатомасовою електромеханічною установкою механізму розвороту роторного екскаватора. Запропонований підхід дозволяє значно підвищити точність регулювання та стабільність динамічних процесів у порівнянні з традиційними методами керування.

Основні досягнуті результати можна узагальнити таким чином:

1. На основі ретельного аналізу сучасних принципів побудови систем керування основними механізмами роторних екскаваторів доведено доцільність використання нейромережевих підходів для автоматизації процесу розвороту. Показано, що саме нейромережеві регулятори здатні забезпечити високу адаптивність і точне відтворення необхідних режимів роботи при складній, багатомасовій структурі механічної частини приводу.

2. Враховуючи специфіку роботи механізму розвороту та вимоги до надійності електроприводів роторних екскаваторів, обґрунтовано вибір системи електроприводу постійного струму за схемою ТП–Д. Проведено підбір двигуна та елементів силової частини електроприводу з урахуванням необхідної перевантажувальної здатності, динамічних вимог та умов експлуатації.

3. Виконано вибір структури та проведено розрахунок системи автоматичного управління параметрами електроприводу. З метою забезпечення необхідної жорсткості робочої ділянки механічної характеристики, а також обмеження максимального значення розвиваного електроприводом моменту, обрано систему зі зворотним зв'язком за напругою тиристорного перетворювача. Для обґрунтування прийнятого рішення здійснено розрахунок та аналіз статичних характеристик як розімкненої, так і замкненої системи автоматичного управління. Побудовано алгоритмічну схему функціонування системи та визначено її передавальні функції за задавальною і збурювальною діями. З метою перевірки працездатності та оцінювання динамічних властивостей розробленої системи про-

ведено комп'ютерне моделювання із застосуванням пакету прикладних програм MATLAB.

4. Розроблено математичну модель двомасової механічної частини електроприводу, в якій ураховано пружні властивості механічних передач і зв'язків. Моделювання показало, що у такій системі перехідні процеси супроводжуються суттєвими коливаннями основних координат, що негативно впливає на точність позиціювання і навантажувальні характеристики. Це підтвердило необхідність застосування більш досконалих методів керування.

5. Реалізовано синтез нейромережевого прогнозного регулятора типу NN Predictive Controller. У процесі параметричного аналізу визначено ті налаштування регулятора, які мають найбільший вплив на динамічні властивості системи. Завдяки використанню нейромережевої моделі об'єкта, що забезпечує якісну ідентифікацію, вдалося отримати регулятор, який дозволяє істотно покращити показники якості керування за рахунок точного прогнозування поведінки об'єкта.

6. Моделювання інтегрованої системи з розробленим нейрорегулятором підтвердило ефективність запропонованого підходу. У режимах пуску та при накладенні збурюючого навантаження система демонструє стійкі та якісні перехідні процеси: перерегулювання знаходиться в межах до 5 %, а час встановлення становить приблизно 2 секунди. Такі показники повністю відповідають заданим вимогам і свідчать про досягнення поставленої у роботі мети.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Штучні нейронні мережі: навчальний посібник / С. В. Ткаліченко. – Кривий Ріг: Державний університет економіки і технологій, 2023. –150 с.
2. 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.
3. 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.
4. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
5. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
6. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г.Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006, 404 с.
7. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
8. Тимощук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. -444 с.
9. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
10. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах:підручник..– Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
11. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.

12. Штучні нейронні мережі: базові положення Навчальний посібник Електронне мережне навчальне видання / І.А. Терейковський, Д.А. Бушуєв, Л.О. Терейковська, Київ: КПІ ім. Ігоря Сікорського, 2022. - 123 с.
13. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – 1958. – № 65.– P.386-407 (Имеется перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир, 1966. – 480 с.).
15. Kaur, *A Comprehensive Survey of Deep Learning Models Across Diverse Application Domains, International Journal of Intelligent Systems and Applications in Engineering*, 12(22s) (2024) 2176-2178
16. Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu, *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*, IEEE/NLN (2021) (Accepted Manuscript)
17. Neural networks for machine learning applications”, Sanjay Lote, Praveena K B, Durugappa Patrer, *World Journal of Advanced Research and Reviews*, 2020, 06(01), 270-282
18. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem / Proc. of IEEE First Ann. Int. Conf. on Neural Networks. – San Diego, 1987. – 3. – P. 11-13.
19. Hecht-Nielsen R. Theory of the Backpropagation Neural Network / Proc. of Int. Joint. Conf. on Neural Networks. – Washington, D.C., 1989. – 1. – P. 593-606.
20. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition // American Math. Soc. Transl. – 1963. – №28. – P. 55-63.
21. Cybenko G. Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314.

22. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // *Neural Networks*. – 1989. – 2. – P. 359-366
23. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks / *Neural Networks*. – 1991. – 4. – P. 251-257.
24. Pinkus, A. (1999). *Approximation theory of the MLP model in neural networks*. *Acta Numerica*, 8, 143–195.
25. Hopfield J.J., Tank D. W. Neural Computation of Decisions in Optimization Problems // *Biol. Cybernetics*. – 1985. – 52. – P. 141-152.
26. Sun S., Cao Z., Zhu H., Zhao J. A survey of optimization methods from a machine learning perspective // *IEEE Transactions on Cybernetics*. – 2020. – Vol. 50, No. 8. – P. 3668–3681.
27. Zhao R., Yan R., Chen Z., Mao K., Wang P., Gao R. X. Deep learning and its applications to machine health monitoring // *Mechanical Systems and Signal Processing*. – 2019. – Vol. 115. – P. 213–237.
28. Rawat W., Wang Z. Deep convolutional neural networks for image classification: A comprehensive review // *Neural Computation*. – 2017. – Vol. 29, No. 9. – P. 2352–2449.
29. Lim B., Zohren S. Time-series forecasting with deep learning: A survey // *Philosophical Transactions of the Royal Society A*. – 2021. – Vol. 379, No. 2194. – Article 20200209.
30. Liu H., Lang B., Liu M., Yan H. Machine learning and deep learning methods for cybersecurity // *IEEE Access*. – 2020. – Vol. 8. – P. 139 101–139 126.
31. Liu, H., Lang, B., Liu, M., Yan, H. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey // *Applied Sciences*. – 2019. – Vol. 9, No. 20. – P. 4396. – MDPI
32. Aljawarneh, S., Aldwairi, M., Yassein, M. B. A Survey of Neural Networks Usage for Intrusion Detection Systems // *Journal of Ambient Intelligence and Humanized Computing*. – 2021. – Vol. 12, No. 3. – P. 3153–3172.
33. Kober J., Bagnell J. A., Peters J. Reinforcement learning in robotics: A survey // *The International Journal of Robotics Research*. – 2013. – Vol. 32, No. 11. – P.

1238–1274.

34. Esteva A., Robicquet A., Ramsundar B., et al. A guide to deep learning in healthcare // *Nature Medicine*. – 2019. – Vol. 25. – P. 24–29.
35. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J. T. A Review of Deep Learning Algorithms and Their Applications in Healthcare // *Journal of Biomedical Informatics*. – 2018. – Vol. 83. – P. 168–185.
36. Young, T., Hazarika, D., Poria, S., Cambria, E. Recent trends in deep learning based natural language processing // *IEEE Computational Intelligence Magazine*. – 2018. – Vol. 13, No. 3. – P. 55–75.
37. Runge, J., et al. Renewable energy forecasting: A review // *Renewable and Sustainable Energy Reviews*. – 2019. – Vol. 120. – 109642.
38. Kuster, C., Rezgui, Y., Mourshed, M. A Scoping Review of Deep Neural Networks for Electric Load Forecasting // *Energy Informatics*. – 2021. – Vol. 4, No. 1. – P. 1–26.
39. Amasyali, K., El-Gohary, N. M. A Review of Deep Learning Techniques for Forecasting Energy Use in Buildings // *Energy and Buildings*. – 2018. – Vol. 177. – P. 535–548.
40. Jain, A., Kumar, S. Deep learning in manufacturing: A review of applications, challenges, and future directions // *Journal of Manufacturing Systems*. – 2022. – Vol. 63. – P. 224–249.
41. Li, X., Zhang, W., Chen, X. Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review // *Sensors*. – 2023. – Vol. 23, No. 3. – P. 1305. – MDPI.
42. Wang, Y., Chen, Z., Li, H., et al. Data-Driven Machinery Fault Detection: A Comprehensive Review // *Mechanical Systems and Signal Processing*. – 2024. – Vol. 210. – 110013.
43. Cao, Y., Gu, Q. A survey on generative AI for art, design, and media // *ACM Computing Surveys*. – 2023. – Vol. 55, No. 12. – P. 1–35

44. Narendra Narendra K. Gallman P.O. An iterative method for the identification of nonlinear system using a Hammerstein model // IEEE Trans. Aut. Control. - 1966. - Vol. 11. -N3. - P.546-550.
45. Psaltis, D., Sideris, A., Yamamura, A. A Multilayered neural network controller // IEEE Control Systems Magazine. - 1988. - Vol.8. - N4. -P. 17-21.
46. Joardan M.A. Generic constraints on underspecified target trajectories // Proc. Of Int. Joint Conf. On Neural Networks (IJCNN), Washington. - 1989.- Vol.1. -N3. - P.217-225.
47. Joardan M.A., Rumelhart D.E. Forward models: Supervised learning with a distal teacher // Cognitive Science.- 1990. -Vol.16. -N5.-P.313-355.
48. Saerens M, Soquet A. Neural controller based on backpropagation algorithm // IEEE Proceedings-F. - 1991. -Vol.138. -N.I. -P.55-62.
49. Miller III, W.T., Glanz, F.H., Kraft III, L.G. Application of a general learning algorithm to control of robotic manipulators // Int. J. of Robotics Research. - 1987. - Vol.6. - N2. - P.84-98.
50. Kawato, M., Furukawa, K, Suzuki, R. A hierarchical neural network model for control and learning of voluntary movement//Biological Cybernetics. - 1987. - Vol.57. - N2. -P. 169-185.
51. Pham D.T., Liu X. Neural Network for Identification, Prediction and Control. - London: Springer - Verlag, 1997. - 238p.
52. Milito R., Padilla C.S., Padilla R.A., Cadorin D. An innovations approach to dual control // IEEE Trans. Automat. Control. - 1982. -Vol.27. -N1. -P.133-137.
53. Cichocki A., Unbehauen R. Neural networks for optimization and signal processing. – New York: John Wiley & Sons, 1998. – 52lp.
54. Garcia C.E. Model predictive control: theory and practice a survey / Garcia C.E., Prett D.M., Morari M. // Automatica. – 1989. – vol.25. – P.335-348.
55. Recurrent Neural Network-Based Model Predictive Control for Continuous Pharmaceutical Manufacturing (Wong W.C., Chee E., Li J., Wang X.) – *Mathematics*, 2018, 6(11):242.

56. Neural Network Based Model Predictive Control (Piche S., Keeler J.D., Martin G., Boe G., Johnson D., Gerules M.) – *Advances in Neural Information Processing Systems NIPS'99*, 1999.
57. Review on model predictive control: an engineering perspective – *The International Journal of Advanced Manufacturing Technology*, 2021, Vol. 117, pp. 1327-1349.
58. Lan J. *Efficient model predictive control for nonlinear systems modelled by deep neural networks* : e-print (arXiv) / J. Lan. - 2024.
59. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // *IEEE Trans. Neural Networks*.–1990. – Vol.1.–№1. – P.4-27.
60. Leontaritis I.J., Billings S.A. Input-output parametric model for nonlinear systems. Part 1: Deterministic non-linear systems. Part 2: Stochastic non-linear systems // *Int. J. Control*. – 1985. – Vol.41. – №2. –P. 303-344.
61. Bunke J. Kunstliche neuronale Netze zur Systemidentifikation aus gestorten Mefiwerten. *Forsch. Ber. VDI, Reihe 8, № 667*. – Dusseldorf: VDI Verlag. – 1997. – P. 37-51.
62. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
63. Теорія автоматичного управління : навчальний посібник [Електронний ресурс] : навч. посіб. / О. Й. Штіфзон, П. В. Новіков, В. П. Бунь. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 144 с.
64. Теорія автоматичного управління. Нелінійні та дискретні системи [Електронний ресурс] : навчальний посібник / О. Й. Штіфзон, П. В. Новіков. – Київ : КПІ ім. Ігоря Сікорського, 2021. – 98 с.
65. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
66. Теорія автоматичного керування : навчальний посібник / О. К. Аблесімов – К. : «Освіта України», 2019. – 270 с.

67. Навчальний посібник з дисципліни "Теорія автоматичного керування" : у 2 ч. Ч. 1 / А. П. Гуров, С. І. Ольшевський, О. О. Черно, Л. І. Бугрім. – Миколаїв : НУК, 2018. – 111 с.
68. Теорія автоматичного управління : курс лекцій / Т. М. Боровська. – Вінниця : ВНТУ, 2018. – 256 с.
69. Теорія автоматичного управління: Навчальний посібник [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», освітньо-професійна програма «Автоматизація та комп'ютерно-інтегровані технології кібер-енергетичних систем»; уклад.: О. Й. Штіфзон, П. В. Новіков, В.П. Бунь. – Електронні текстові дані (1 файл: 2,2 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2020. – 144 с
70. Лістровий С. В., Мірошник М. А., Клименко Л. А. Теорія автоматичного керування, штучний інтелект і автоматизація процесу прийняття рішення: Навч. посібник. – Харків: УкрДУЗТ, 2019. – 120 с.
71. Автоматизований електропривод. Розділ 1. Теорія електропривода. Навчальний посібник. / Н.Д. Красношопка – К.: КПІ ім. Ігоря Сікорського. - 2023. – 101 с.
72. Возняк, О.М., Штуць. А.А., Колісник М.А. Сучасні системи електроприводів. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ТВОРИ, 2021. – 280 с.
73. А. А. Видмиш, Л. В. Ярошенко. Основи електропривода. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ВНАУ, 2020. – 387 с.
74. Панкратов А.І. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерноінтегровані технології» денної і заочної форми навчання) – Краматорськ: ДДМА, 2018. – 225 с.
75. Колб Ант. А, Колб А. А. Теорія електроприводу: Навчальний посібник. – 2-е вид. перероб. і доп. – Д., Національний гірничий університет, 2011. – 540 с.
76. Панкратов А.І. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів

спеціальності 151 «Автоматизація та комп'ютерноінтегровані технології» денної і заочної форми навчання) – Краматорськ: ДДМА, 2018. – 225 с.

77. Василега П. О. Електропривод робочих машин : підручник / П. О. Василега. – Суми: Сумський державний університет, 2022. – 290 с. ISBN 978-966-657-900-6.

78. Шефер О.В. Автоматизований електропривод загальнопромислових механізмів: конспект лекцій. – Полтава: ПолтНТУ, 2020. – 154 с.

79. Електропривод виробничих машин і механізмів: Навчальний посібник / О.Ю. Синявський, В.В. Савченко, В.Я. Бунько, В.Ю. Рамш; За ред. О.Ю. Синявського. – К.: ФОП Ямчинський О.В., 2020. – 444 с. ISBN 978-617-7890-88-0.

80. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоєнко. Харків, ХНАДУ, 2012. – 284 с.
Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолов, О. Г. Гурко, В. І. Клименко, Д. М.

ДОДАТОК А

СТРУКТУРА ШТУЧНОГО НЕЙРОНА

Нервова система і мозок людини складаються з нейронів, сполучених між собою нервовими волокнами. Нервові волокна здатні передавати електричні імпульси між нейронами. Всі процеси передачі роздратувань від шкіри, вух і очей до мозку, процеси мислення і управління діями - все це реалізовано в живому організмі як передача електричних імпульсів між нейронами.

Нейрон (нервова клітина) є особливою біологічною кліткою, яка обробляє інформацію (рис. А.1). Він складається з тіла і відростків нервових волокон двох типів - дендритов, по яким приймаються імпульси, і єдиного аксона, по якому нейрон може передавати імпульс. Тіло нейрона включає ядро, яке містить інформацію про спадкові властивості, і плазму, що володіє молекулярними засобами для виробництва необхідних нейрону матеріалів. Нейрон отримує сигнали (імпульси) від аксонів інших нейронів через дендрити (приймачі) і передає сигнали, які згенеровані тілом клітки, уздовж свого аксона (передавач), який в кінці розгалужується на волокна. На закінченнях цих волокон знаходяться спеціальні утворення - синапси, які впливають на силу імпульсу.

Синапс є елементарною структурою і функціональним вузлом між двома нейронами (волокно аксона одного нейрона і дендрит іншого). Коли імпульс досягає синаптичного закінчення, вивільняються певні хімічні речовини, звані нейротрансмітерами. Нейротрансмітери дифундують через синаптичну щілину, збуджуючи або загальмовуючи, в залежності від типу синапсу, здатність нейрона-приймача генерувати електричні імпульси. Результативність синапсу може настраюватися сигналами, що проходять через нього, так що синапси можуть навчатися залежно від активності процесів, в яких вони беруть участь. Ця залежність від передісторії діє як пам'ять, яка, можливо, відповідальна за пам'ять людини. Важливо відзначити, що ваги синапсів можуть змінюватися з часом, що змінює і поведінку відповідного нейрона.

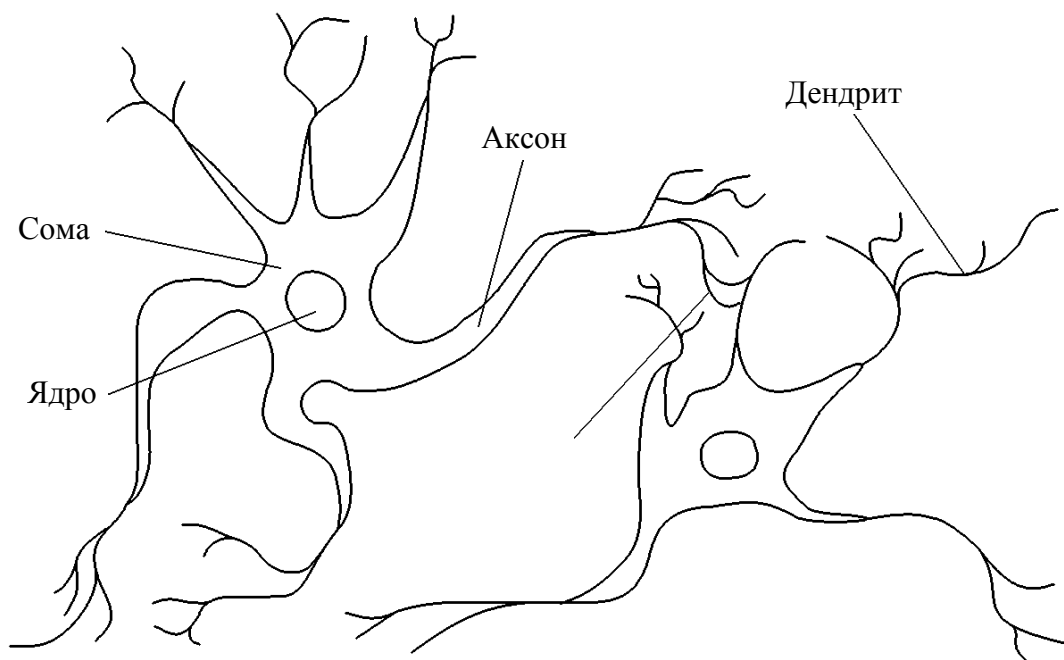


Рисунок А.1 – Взаємозв'язок біологічних нейронів

Структура штучного нейрона показана на рис. А.2.

До складу нейрона входять помножувачі (синапси), суматор і нелінійний перетворювач. Синапси здійснюють зв'язок між нейронами і умножають вхідний сигнал на число, що характеризує силу зв'язку, – вага синапсу. Суматор виконує складання сигналів, що поступають по синаптичним зв'язкам від інших нейронів, і зовнішніх вхідних сигналів. Нелінійний перетворювач реалізує нелінійну функцію одного аргументу – виходу суматора. Ця функція називається функцією активації або передавальною функцією нейрона. Нейрон в цілому реалізує скалярну функцію векторного аргументу. Математична модель нейрона описується співвідношеннями:

$$s = \sum_{i=1}^n w_i \cdot x_i + b, \quad y = f(s)$$

де w_i – вага синапсу ($i = 1 \dots n$);

b – значення зсуву;

s – результат підсумовування;

x_i – компонент вхідного вектора (вхідний сигнал) ($i = 1 \dots n$);

y – вихідний сигнал нейрона;
 n – число входів нейрона;
 f – нелінійне перетворення (функція активації або передавальна функція).

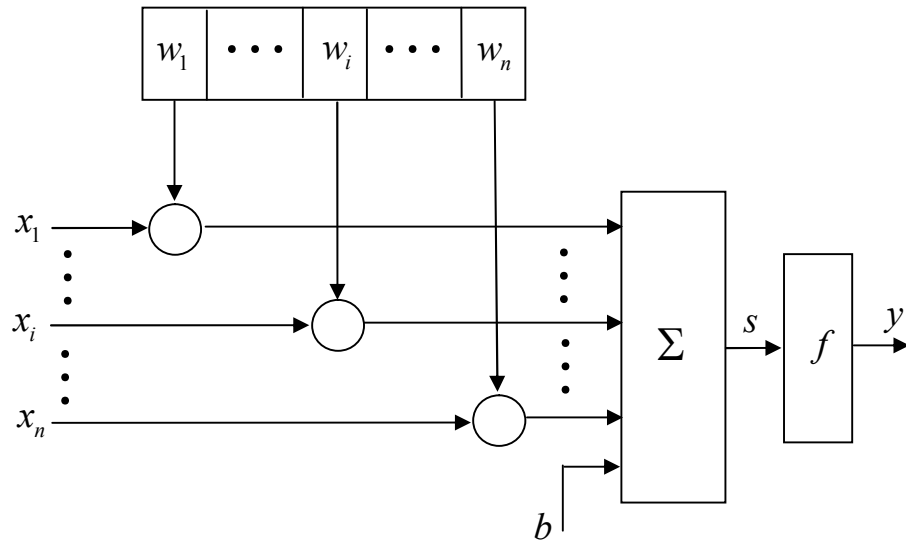


Рисунок А.2 – Структура штучного нейрона

У загальному випадку вхідний сигнал, вагові коефіцієнти і значення зсуву можуть приймати речовинні значення. Вихід (y) визначається видом функції активації і може бути як дійсним, так і цілим. У багатьох практичних завданнях входи, ваги і зсуви можуть приймати лише деякі фіксовані значення.

Синаптичні зв'язки з позитивними вагами називають збудливими, з негативними вагами – гальмуючими.

Таким чином, нейрон повністю описується своїми вагами w_i і передавальною функцією $f(s)$. Отримавши набір чисел (вектор) x_i як входи, нейрон видає деяке число y на виході.

Описаний обчислювальний елемент можна вважати спрощеною математичною моделлю біологічних нейронів – кліток, з яких складається нервова система людини і тварин.

Щоб підкреслити відмінність нейронів біологічних і штучних (математичних), другі іноді називають нейроноподібними елементами або формальними (штучними) нейронами.

На вхідний сигнал s нелінійний перетворювач відповідає вихідним сигналом $f(s)$, який є виходом нейрона y . Приклади активаційних функцій представлені на рис. А.3 і в табл. А.1.

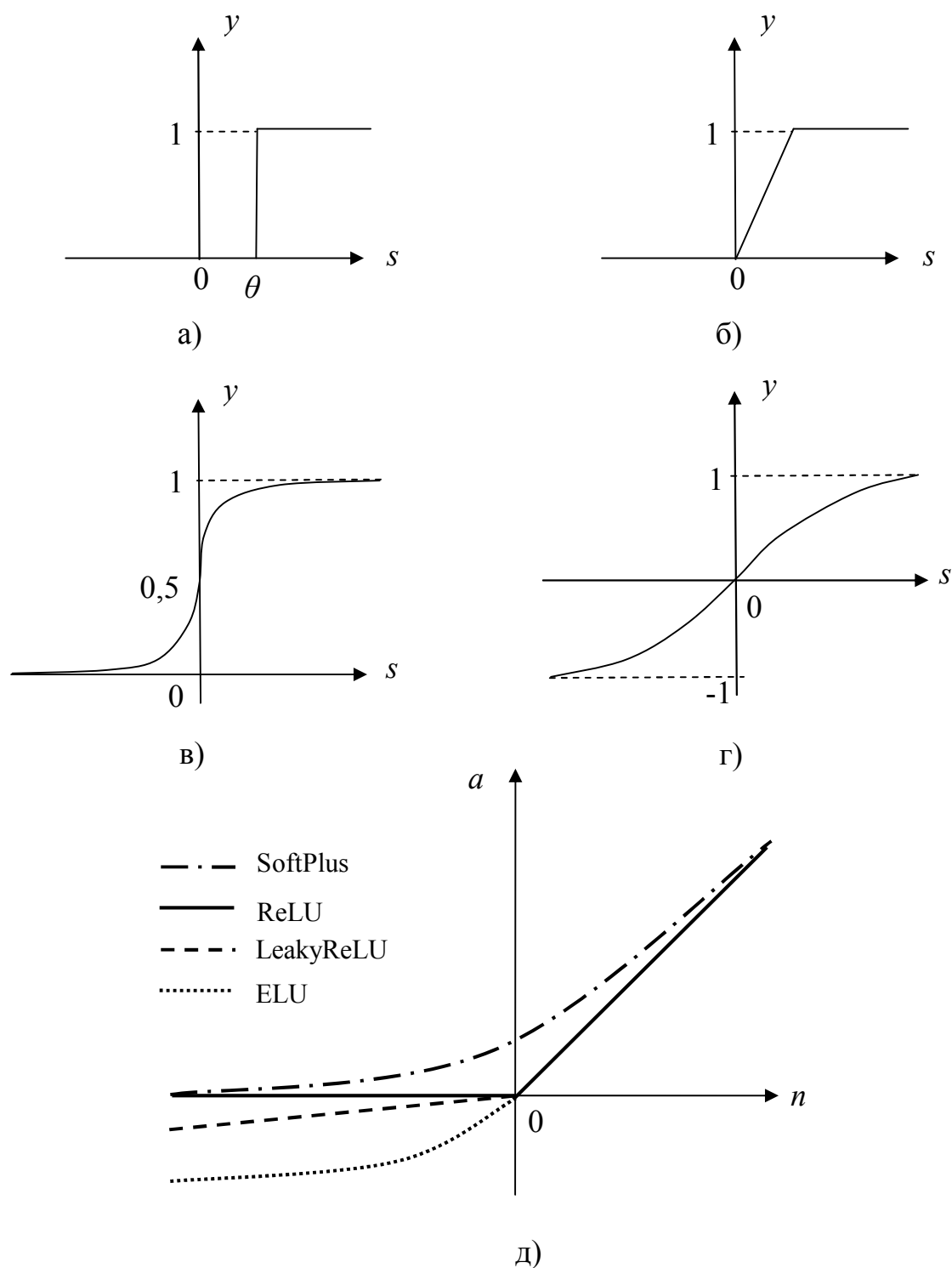


Рисунок А.3 – Приклади активаційних функцій

а – порогова; б – напівлінійна з насиченням; в – сигмоїд (логістична функція); г – сигмоїд (гіперболічний тангенс); д - функція активації ReLU

Таблиця А.1 - Перелік функцій активації нейронів

Назва	Формула	Область значень
Порогова	$f(s) = \begin{cases} 0, & s < \theta \\ 1, & s \geq \theta \end{cases}$	$(0, 1)$
Знакова (сигнатурна)	$f(s) = \begin{cases} 1, & s > 0 \\ -1, & s \leq 0 \end{cases}$	$(-1, 1)$
Сигмоїдальна (логістична)	$f(s) = \frac{1}{1 + e^{-s}}$	$(0, 1)$
Напівлінійна	$f(s) = \begin{cases} s, & s > 0 \\ 0, & s \leq 0 \end{cases}$	$(0, \infty)$
Лінійна	$f(s) = s$	$(-\infty, \infty)$
Радіальна базисна (гаусова)	$f(s) = \exp(-s^2)$	$(0, 1)$
Напівлінійна з насиченням	$f(s) = \begin{cases} 0, & s \leq 0 \\ s, & 0 < s < 1 \\ 1, & s \geq 1 \end{cases}$	$(0, 1)$
Лінійна з насиченням	$f(s) = \begin{cases} -1, & s \leq -1 \\ s, & -1 < s < 1 \\ 1, & s \geq 1 \end{cases}$	$(-1, 1)$
Гіперболічний тангенс (сигмоїдальна)	$f(s) = \frac{e^s - e^{-s}}{e^s + e^{-s}}$	$(-1, 1)$
Трикутна	$f(s) = \begin{cases} 1 - s , & s \leq 1 \\ 0, & s > 1 \end{cases}$	$(0, 1)$
ReLU	$f(n) = \begin{cases} 0, & n < 0 \\ n, & n \geq 0 \end{cases}$	$f(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$
SoftPlus	$f(n) = \log(1 + e^n)$	$f'(n) = \frac{1}{1 + e^{-n}}$
Leaky ReLU, Parameterized ReLU	$f(n) = \begin{cases} an, & n < 0 \\ n, & n \geq 0 \end{cases}$	$f'(n) = \begin{cases} a, & n < 0 \\ 1, & n \geq 0 \end{cases}$
ELU	$f(n) = \begin{cases} a(e^n - 1), & n < 0 \\ n, & n \geq 0 \end{cases}$	$f'(n) = \begin{cases} f(n) + a, & n < 0 \\ 1, & n \geq 0 \end{cases}$

В магістерській роботі використовується нелінійна функція з насиченням, так звана логістична функція або сигмоїд (тобто функція *S-подібного* вигляду)

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

З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить в діапазоні $[0, 1]$. Одна з цінних властивостей сигмоїдальної функції – простий вираз для її похідної:

$$f'(s) = f(s) \cdot [1 - f(s)].$$

Крім того, дана функція володіє властивістю «підсилювати» слабкі сигнали краще, ніж великі, і запобігає насиченню від великих сигналів, оскільки вони відповідають областям аргументів, де сигмоїд має пологий нахил.

ДОДАТОК Б

КЛАСИФІКАЦІЯ НЕЙРОННИХ МЕРЕЖ І ЇХ ВЛАСТИВОСТІ

Штучна нейронна мережа – це набір нейронів, сполучених між собою. Як правило, передавальні (активаційні) функції всіх нейронів в мережі фіксовані, а ваги є параметрами мережі і можуть змінюватися. Деякі входи нейронів помічені як зовнішні входи мережі, а деякі виходи – як зовнішні виходи мережі. Подаючи будь-які числа на входи мережі, ми отримуємо якийсь набір чисел на виходах мережі. Таким чином, робота нейронної мережі полягає в перетворенні вхідного вектора X у вихідний вектор Y , причому це перетворення задається вагами мережі.

Практично будь-яке завдання можна звести до завдання, вирішуваною нейронною мережею.

Питання про методику побудови нейронної мережі вирішується в два етапи:

- 1 – вибір типу (архітектура) мережі;
- 2 – підбір вагів (навчання) мережі.

На першому етапі слід вибрати наступне:

- 1 – які нейрони ми хочемо використовувати (число входів, передавальні функції);
- 2 – яким чином слід з'єднати їх між собою;
- 3 – що взяти за входи і виходи мережі.

Це завдання на перший погляд здається незрозумілим, але, на щастя, необов'язково придумувати нейронну мережу «з нуля» – існує декілька десятків різних нейромережових архітектур, причому ефективність багатьох з них доведена математично. Найбільш популярні і вивчені архітектури – це багатошаровий перцептрон, нейромережа із загальною регресією, мережі Кохонена та інші.

На другому етапі слід «навчити» вибрану мережу, тобто підібрати такі значення, її вагів, щоб мережа працювала потрібним чином. У використовуваних на практиці нейромережах кількість вагів може складати декілька десятків

тисяч, тому навчання – дійсно складний процес. Для багатьох архітектури розроблені спеціальні алгоритми навчання, які дозволяють набудувати ваги мережі певним чином.

Залежно від функцій, що виконуються нейронами в мережі, можна виділити три їх типу:

1 – вхідні нейрони – на які подається вхідний вектор, що кодує вхідну дію або образ зовнішнього середовища; у них зазвичай не здійснюється обчислювальних процедур, інформація передається з входу на вихід нейрона шляхом зміни його активації;

2 – вихідні нейрони – вихідні значення яких представляють вихід мережі; перетворення в таких нейронах здійснюються по приведених вище виразах;

3 – проміжні нейрони – складають основу штучних нейронних мереж, перетворення в них виконуються по цих же виразах.

У більшості нейронних моделей тип нейрона пов'язаний з його розташуванням в мережі. Якщо нейрон має тільки вихідні зв'язки, то це вхідний нейрон, якщо навпаки – вихідний нейрон. Проте може зустрітися випадок, коли вихід внутрішнього нейрона розглядається як частина виходу мережі. В процесі функціонування мережі здійснюється перетворення вхідного вектора у вихідний, тобто деяка переробка інформації. Конкретний вид виконуваного мережею перетворення інформації обумовлюється не тільки характеристиками нейроподібних елементів (функціями активації і т. п.), але і особливостями її архітектури. Зокрема, тією або іншою топологією міжнейронних зв'язків, вибором певних підмножин нейроподібних елементів для введення і виведення інформації, способами навчання мережі, наявністю або відсутністю конкуренції між нейронами, напрямом і способами передачі інформації між нейронами.

Класифікуючи нейронні мережі по топології, можна виділити три основні типи таких мереж:

- 1 – повнозв'язні мережі (рис. Б.1 а);
- 2 – багатошарові або шаруваті мережі (рис.Б.1 б);
- 3 – слабозв'язані мережі (нейронні мережі з локальними зв'язками) (рис.Б.1в).

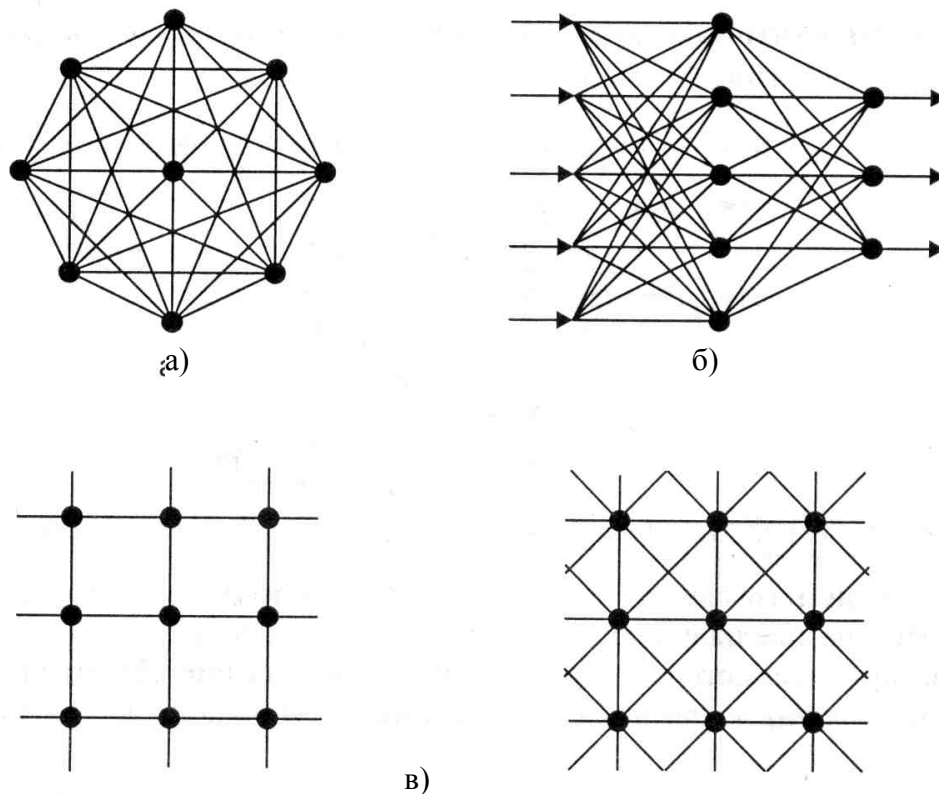


Рисунок – Б.1 Архітектура нейронних мереж

а – повнозв'язна мережа

б – багатошарова мережа з послідовними зв'язками;

в – слабозв'язні мережі

Повнозв'язна мережа – це ШНМ, кожен нейрон якої передає свій вихідний сигнал решті нейронів, у тому числі і самому собі. Всі вхідні сигнали подаються всім нейронам. Вихідними сигналами мережі можуть бути все або деякі вихідні сигнали нейронів після декількох тактів функціонування мережі.

У багатошарових мережах нейрони об'єднуються в шари. Шар містить сукупність нейронів з єдиними вхідними сигналами. Число нейронів в кожному шарі може бути будь-яким і ніяк наперед не пов'язано з кількістю нейронів в інших шарах. У загальному випадку мережа складається з Q шарів, пронумерованих зліва направо. Зовнішні вхідні сигнали подаються на входи нейронів першого шару (вхідний шар часто нумерують як нульовою), а виходами мережі є вихідні сигнали останнього шару. Вхід нейронної мережі можна розглядати як

вихід «нульового шару» вироджених нейронів, які служать лише як розподільні крапки, підсумовування і перетворення сигналів тут не проводиться. Окрім вхідного і вихідного шарів в багатошаровій нейронній мережі є один або декілька проміжних (прихованих) шарів. Зв'язки від виходів нейронів деякого шару q до входів нейронів наступного шару ($q + 1$) називаються послідовними.

У свою чергу, серед шаруватих мереж виділяють наступні типи.

Мережі без зворотних зв'язків. У таких мережах нейрони вхідного шару отримують вхідні сигнали, перетворюють їх і передають нейронам 1-го прихованого шару, далі спрацьовує 1-й прихований шар і т.д. до 0-го шару, який видає вихідні сигнали для інтерпретатора і користувача. Якщо не обумовлене осоружне, то кожен вихідний сигнал q -го шару подається на вхід всіх нейронів ($q + 1$)-го шару; проте можливий варіант з'єднання q -го шару з довільним ($q + p$)-м шаром.

Слід зазначити, що класичним варіантом шаруватих мереж є мережі прямого розповсюдження (рис. Б.2).

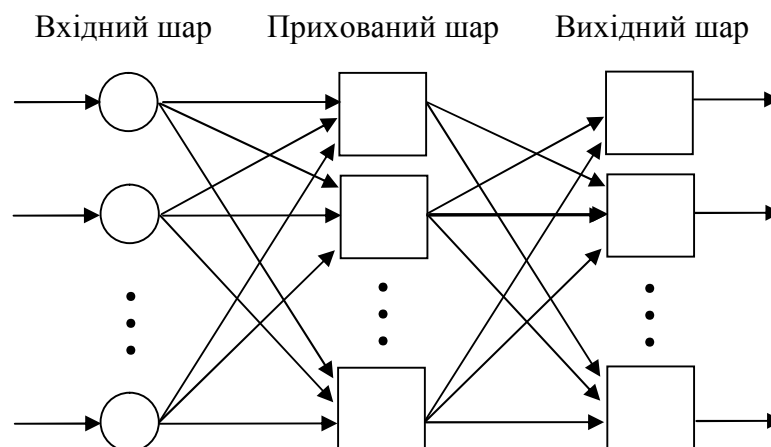


Рисунок Б.2 – Багатошарова (двошарова) мережа прямого розповсюдження

Мережі із зворотними зв'язками. Це мережі, у яких інформація з подальших шарів передається на попередні.

Як приклад мереж із зворотними зв'язками на рис. Б.3 представлені так звані частково-рекурентні мережі Елмана і Жордана.

Мережі можна класифікувати також по числу шарів.

Теоретично число шарів і число нейронів в кожному шарі може бути довільним, проте фактично воно обмежене ресурсами комп'ютера або спеціалізованої мікросхеми, на яких зазвичай реалізуються ШНМ. Чим складніше ШНМ, тим масштабніші завдання, підвладні їй.

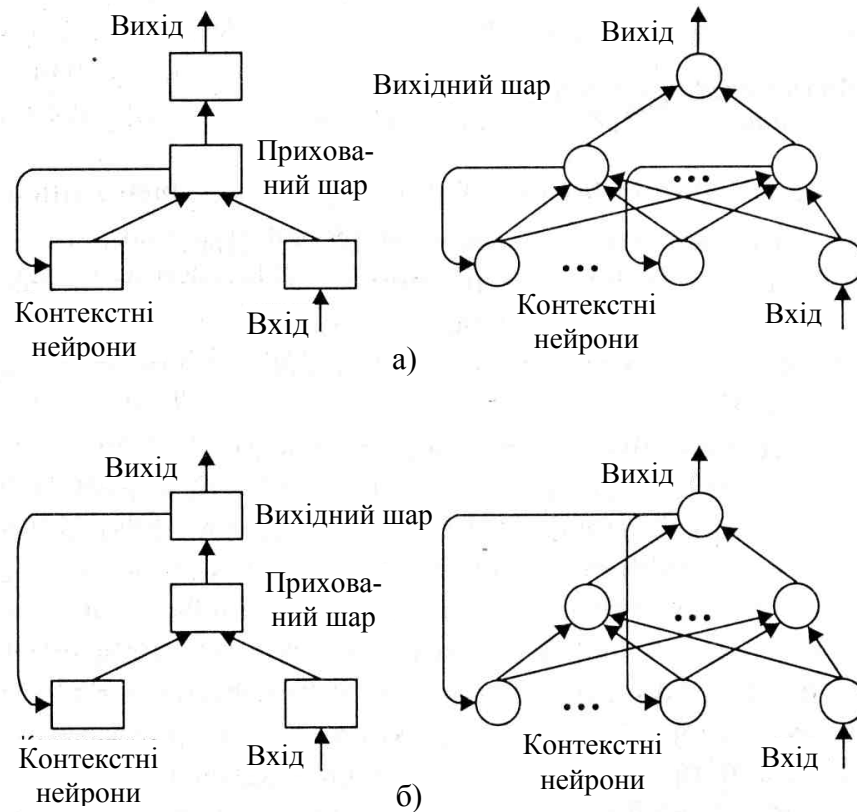


Рисунок Б.3 – Частково-рекурентні мережі
а – мережа Елмана, б – мережа Жордана

Вибір структури ШНМ здійснюється відповідно до особливостей і складності завдання. Для вирішення деяких окремих типів завдань вже існують оптимальні (на сьогоднішній день) конфігурації. Якщо ж завдання не може бути зведене ні до одного з відомих типів, розробникові доводиться вирішувати складну проблему синтезу нової конфігурації. При цьому він керується декількома основоположними принципами:

1 – можливості мережі зростають із збільшенням числа осередків мережі, щільність зв'язків між ними і числом виділених шарів;

2 – введення зворотних зв'язків разом із збільшенням можливостей мережі піднімає питання про так звану динамічну стійкість мережі;

3 – складність алгоритмів функціонування мережі (зокрема, наприклад, введення декількох типів синапсів – збудливих, гальмуючих і ін.) також сприяє посиленню можливостей ШНМ.

Питання про необхідні і достатні властивості мережі для вирішення того або іншого роду завдань є цілим напрямом нейрокомп'ютерної науки. Оскільки проблема синтезу ШНМ сильно залежить від вирішуваної задачі. В більшості випадків оптимальний варіант виходить на основі інтуїтивного підбору, хоча в літературі приведені докази того, що для будь-якого алгоритму існує нейронна мережа, яка може його реалізувати. Зупинимось на цьому докладніше.

Багато завдань: розпізнавання образів (зорових, мовних), виконання функціональних перетворень при обробці сигналів, управління, прогнозування, ідентифікації складних систем, зводяться до наступної математичної постановки. Необхідно побудувати відображення $X \rightarrow Y$ таке, щоб у відповідь на кожен можливий вхідний сигнал X формувався правильний вихідний сигнал Y . Відображення задається кінцевим набором пар ($\langle \text{вхід} \rangle$, $\langle \text{відомий вихід} \rangle$). Число таких пар (навчальних прикладів) істотно менше загального числа можливих поєднань значень вхідних і вихідних сигналів. Сукупність всіх навчальних прикладів носить назву навчальної вибірки.

У завданнях розпізнавання образів X – деяке представлення образу (зображення, вектор чисел), Y – номер класу, до якого належить вхідний образ.

У завданнях управління X – набір контрольованих параметрів керованого об'єкту, Y – код, що визначає дію, що управляє, відповідну поточним значенням контрольованих параметрів.

У завданнях прогнозування як вхідні сигнали використовуються тимчасові ряди, що представляють значення контрольованих змінних на деякому інтервалі часу. Вихідний сигнал – безліч змінних, яка є підмножиною змінних вхідного сигналу.

При ідентифікації $\mathbf{X}$ і $\mathbf{Y}$ представляють вхідні і вихідні сигнали системи відповідно.

Взагалі кажучи, велика частина прикладних завдань може бути зведена до реалізації деякого складного багатовимірного функціонального перетворення $\mathbf{X} \rightarrow \mathbf{Y}$.

В результаті побудови такого перетворення (відображення) необхідно добитися того, щоб забезпечувалося формування правильних вихідних сигналів у відповідності:

- 1 – зі всіма прикладами навчальної вибірки;
- 2 – зі всіма можливими вхідними сигналами, які не увійшли до навчальної вибірки.

Друга вимога в значній мірі ускладнює завдання формування навчальної вибірки. У загальному вигляді ця задача в даний час ще не вирішена, проте у всіх відомих випадках було знайдено приватне рішення. Подальші міркування припускають, що навчальна вибірка вже сформована.

Відзначимо, що теоретичною основою для побудови нейронних мереж є наступне твердження: для будь-якої множини пар вхідних – вихідних векторів довільної розмірності $\{(\mathbf{X}^k \rightarrow \mathbf{Y}^k), k=1...N\}$ існує двошарова однорідна нейронна мережа з послідовними зв'язками, з сигмоїдальними передавальними функціями і з кінцевим числом нейронів, яка для кожного вхідного вектора $\mathbf{X}^k$ формує відповідний йому вихідний вектор $\mathbf{Y}^k$.

Таким чином, для представлення багатовимірних функцій багатьох змінних може бути використана однорідна нейронна мережа, що має всього один прихований шар, з сигмоїдальними передавальними функціями нейронів.

Для оцінки числа нейронів в прихованих шарах однорідних нейронних мереж можна скористатися евристичною формулою для оцінки необхідного числа синаптичних вагів L_w (у багатошаровій мережі з сигмоїдальними передавальними функціями):

$$\frac{mN}{1 + \log_2 N} \leq L_w \leq \left(\frac{N}{m} + 1 \right) (n + m + 1) + m ,$$

де n – розмірність вхідного сигналу

m – розмірність вихідного сигналу

N – число елементів навчальної вибірки.

Оцінивши необхідне число вагів, можна розрахувати число нейронів в прихованих шарах. Наприклад, число нейронів в двошаровій мережі складе

$$L = \frac{L_w}{n + m} .$$

Відомі і інші подібні формули, наприклад, вигляду

$$2 \cdot (L + n + m) \leq N \leq 10 \cdot (L + n + m) ;$$

$$\frac{N}{10} - n - m \leq L \leq \frac{N}{2} - n - m .$$

Також можна розрахувати число нейронів в мережах з великим числом шарів, які іноді доцільно використовувати: такі багатошарові нейронні мережі можуть мати менші розмірності матриць вагів нейронів одного шару, ніж двошарові мережі, що реалізують те ж саме відображення.

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.

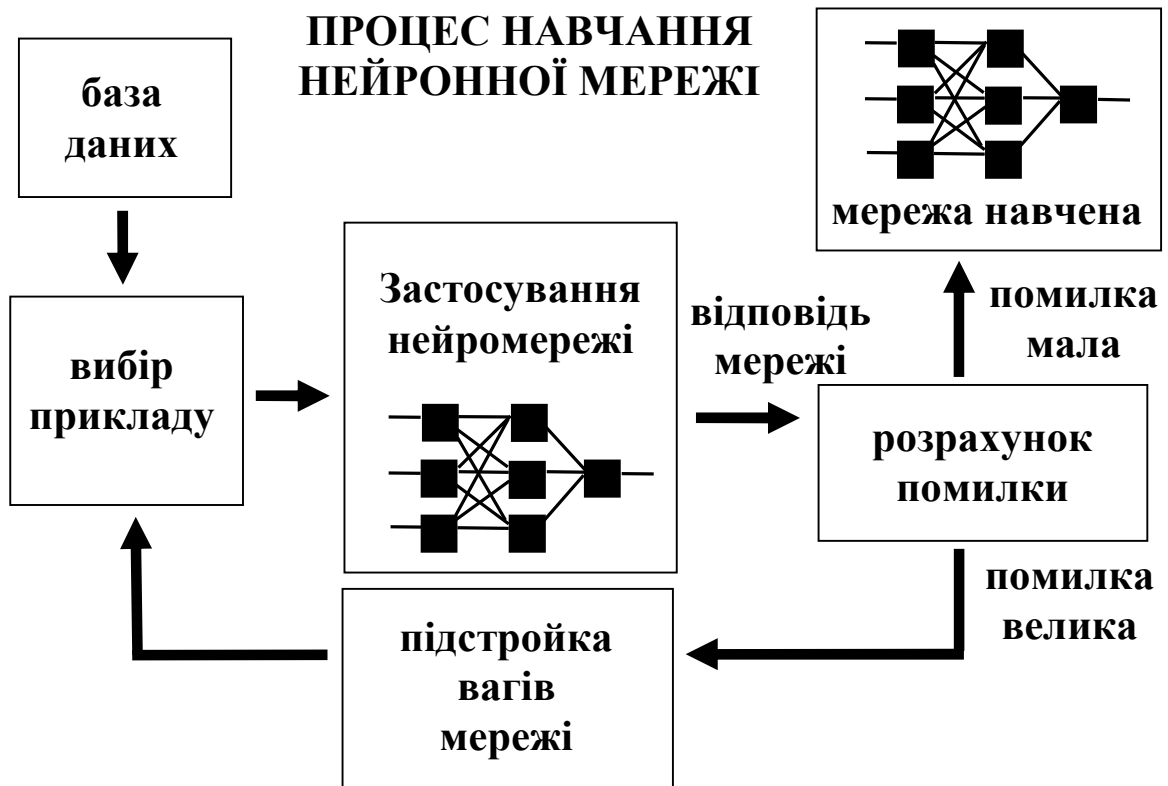


Рисунок В.1 – Ілюстрація процесу навчання ШНМ

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 [1-6]:

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 компонента 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 багат шарових 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 багат шарової 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 настраюються, 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 багат шарових 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:

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

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 багат шарової 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 багат шарової 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 кожну 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 перевірна 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 перевірочній (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 множині, 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

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ

Г.1 Моделі систем, використовувані при нейромережевій ідентифікації

Основним питанням в нейрорегулювання є отримання адекватних моделей динамічних систем. Питання про існування дискретних моделей, або іншими словами про спостереженість нелінійних систем, є ключовим. У загальному випадку завдання побудови математичної моделі об'єкту полягає у виборі структури і оцінці її параметрів так, щоб при використанні критерію мінімуму деякої функції різниці розрахункових і експериментальних даних дотримувалася умова близькості моделі досліджуваному процесу. Застосування нейронних мереж як моделі може служити альтернативою класичним методам математичного опису систем, оскільки в цьому випадку точне знання внутрішніх процесів не є необхідною умовою моделювання.

При нейронній ідентифікації нелінійних систем найчастіше знаходять застосування наступні основні нелінійні дискретні моделі.

У найбільш загальному вигляді, прийнятому в класичній теорії управління, опис стохастичних нелінійних систем в просторі станів розглядався в роботі Наренди К.С. і Партасараті К [59].

$$z(k+1) = \Psi[z(k), u(k), \phi(k)] ;$$

$$y(k) = \Phi[z(k), u(k), \phi(k)],$$

де $z(k) = [z_1(k), z_2(k), \dots, z_{k_z}(k)]^T$ – вектор стану системи;

$u(k) = [u_1(k), u_2(k), \dots, u_{k_u}(k)]^T$ – вектор вхідних сигналів;

$y(k) = [y_1(k), y_2(k), \dots, y_{k_y}(k)]^T$ – вектор вихідних сигналів;

$\Psi[\cdot]$, $\Phi[\cdot]$ – деякі статичні нелінійні функції;

$\phi(k)$, (k) – шум процесу і шум вимірювань відповідно.

Леонартіс І. Дж. і Биллінгз С.Д.[60] запропонували для опису нелінійних систем використовувати так звану нелінійну модель авторегресії ковзаючого середнього з додатковими вхідними сигналами (NARMAX-модель).

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), e(k-1), \dots, e(k-n_e)] + e(k) \quad (\Gamma.1)$$

де $y(k) \in \mathbf{R}^{M \times 1}$ – вихідний сигнал об'єкту;

$u(k) \in \mathbf{R}^{N \times 1}$ – вхідний сигнал об'єкту;

$e(k) \in \mathbf{R}^{M \times 1}$ – різниця вихідних сигналів об'єкту і моделі;

$f[\cdot]$ – нелінійна функція перетворення $f: \mathbf{R}^{(k_y M + k_u N + k_e M) \times 1} \rightarrow \mathbf{R}^{M \times 1}$;

n_y, n_u, n_e – порядки запізнювання по вихідному, вхідному сигналам об'єкту і помилці вимірювань відповідно.

Скалярна форма рівняння (Г.1) має вигляд:

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-n_y), y_2(k-1), \dots, y_2(k-n_y), y_M(k-1), \dots, y_M(k-n_y), \\ & u_1(k-1), \dots, u_1(k-n_u), u_2(k-1), \dots, u_2(k-n_u), u_N(k-1), \dots, u_N(k-n_u), \\ & e_1(k-1), \dots, e_1(k-n_e), e_2(k-1), \dots, e_2(k-n_e), e_M(k-1), \dots, e_M(k-n_e)] + e_i(k) \end{aligned} \quad (\Gamma.2)$$

Нелінійні системи можуть бути представлені нелінійною NARX-моделлю:

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)] + e(k) \quad (\Gamma.3)$$

або в скалярному вигляді

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-k_y), y_2(k-1), \dots, y_2(k-k_y), y_M(k-1), \dots, y_M(k-k_y), \\ & u_1(k-1), \dots, u_1(k-k_u), u_2(k-1), \dots, u_2(k-k_u), u_N(k-1), \dots, u_N(k-k_u)] + e_i(k). \end{aligned} \quad (\Gamma.4)$$

Представлення об'єктів у вигляді NARMAX або NARX моделей грають фундаментальну роль при дослідженні нелінійних об'єктів за допомогою штучних нейронних мереж. При цьому важливо розглянути узагальнення моделей (Г.1) - (Г.4), що враховує різні види нелінійностей по вхідних і вихідних змін-

них. Наренда виділив чотири такі моделі:

– модель 1 (рис. Г.1):

$$y(k+1) = \sum_{i=1}^{k_y-1} a_i y(k-i) + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\Gamma.5)$$

– модель 2 (рис. Г.2):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + \sum_{i=1}^{k_u-1} b_i u(k-i); \quad (\Gamma.6)$$

– модель 3 (рис. Г.3):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\Gamma.7)$$

– модель 4 (рис. Г.4):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (\Gamma.8)$$

Модель (Г.4) є найбільш загальною зі всіх моделей вхід/вихід, описаних вище. При цьому модель (Г.8) є аналітично вирішуваною і найбільш зручною для практичних застосувань. Внаслідок цього вона використовується в роботі при ідентифікації нелінійних об'єктів за допомогою штучних нейронних мереж.

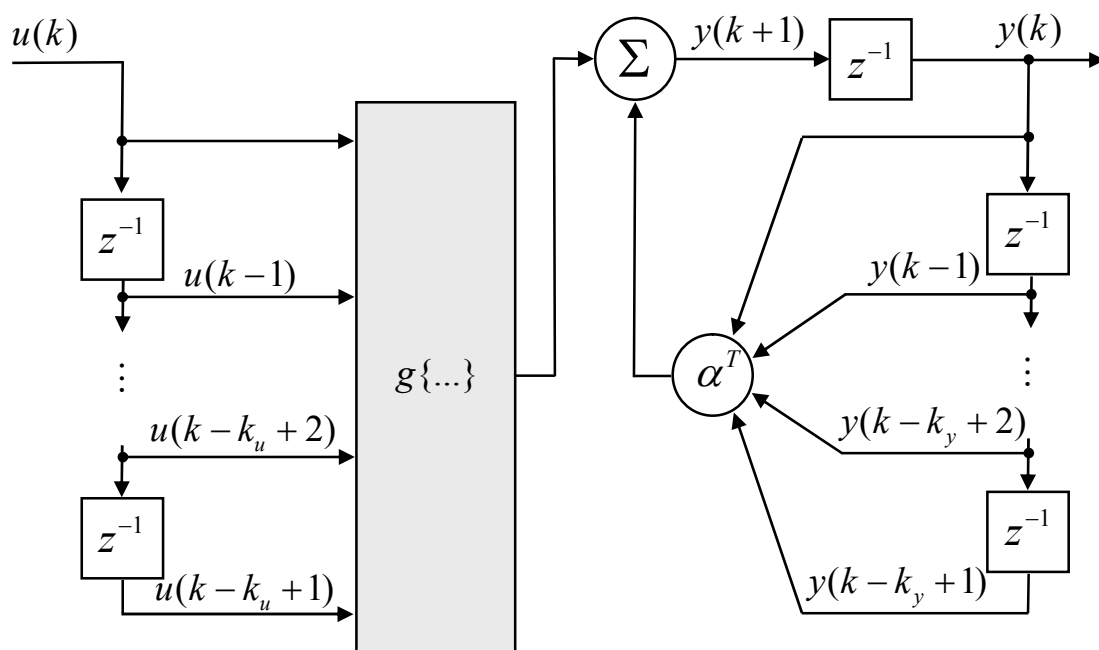


Рисунок Г.1 – Модель 1. Вихід моделі нелінійно залежить від вхідних сигналів і лінійно від вихідних

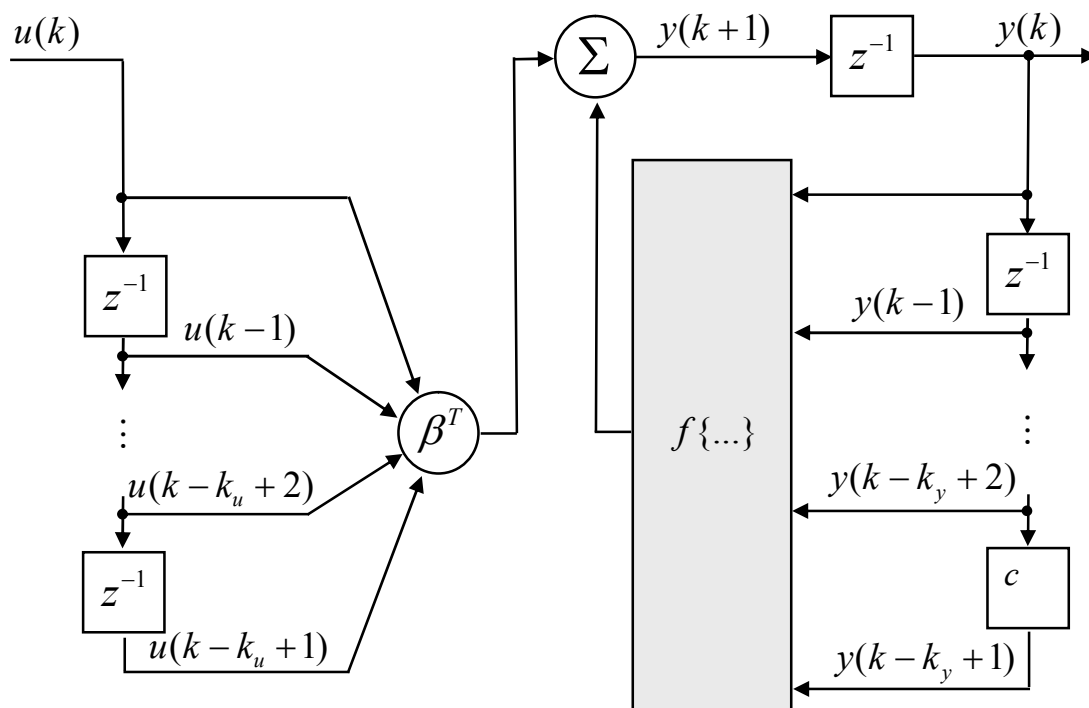


Рисунок Г.2 – Модель 2. Вихід моделі лінійно залежить від вхідних сигналів і нелінійно від вихідних

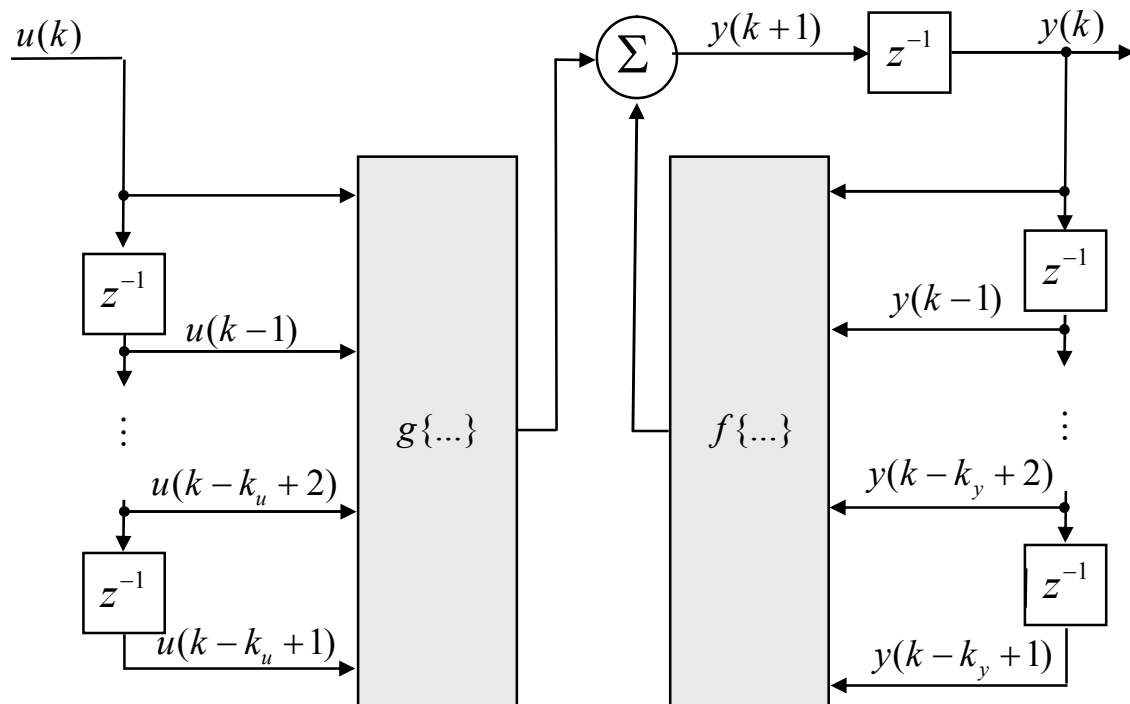


Рисунок Г.3 – Модель 3. Результовий сигнал є адитивною функцією нелінійних перетворень вхідних і вихідних сигналів

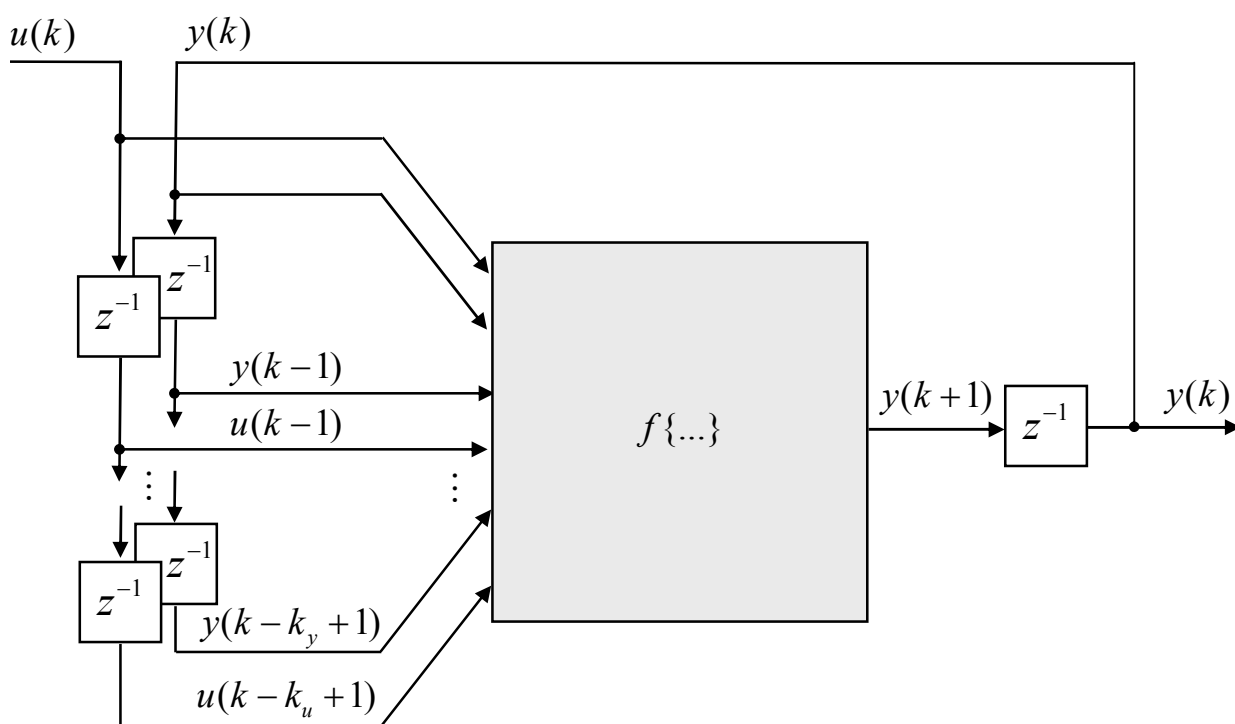


Рисунок Г.4 – Модель 4. Вихід моделі нелінійно залежить як від вхідних, так і від вихідних сигналів

Г.2 Нейронні мережі як універсальні моделі

Оскільки тип і взаємозв'язок нелінійностей в об'єкті, як правило, не відомі, для моделювання слід використовувати якомога більш універсальні структури моделей. Нейронні мережі є саме такими структурами.

Нелінійні моделі в просторі станів можуть бути реалізовані за допомогою рекурсивних нейронних мереж з внутрішніми зворотними зв'язками (мережі Елмана). Ідентифікація параметрів цих мереж вимагає спеціальної і щодо трудомісткої форми алгоритму навчання. Крім того, доказ стійкості таких нелінійних динамічних моделей практично неможливо зробити.

У роботі використовуються прямонаправленні мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямі без зворотних зв'язків усередині мережі, – багат шаровий персептрон. Найбільш часто використовується мережа, багат шаровий персептрон (БП). За допомогою БП можна апроксимувати з бажаною точністю не тільки будь-які статичні функції. Попередні значення вхідних/вихідних координат у вхідному векторі дозволяють додати прямонаправленим мережам динамічні властивості.

Володіючи схожими з радіально-базисними мережами (РБМ) властивостями, БП забезпечує компактніше представлення нелінійного об'єкту, оскільки число параметрів в мережі РБМ експоненціально зростає із збільшенням розмірності вхідного простору, тоді як для БП ця залежність лінійна. Важливою є наочність, що виражається в тому, що персептронне представлення моделі еквівалентне традиційному. Це дозволяє використовувати для визначення параметрів ШНМ добре розвинені методи оцінювання, використовувані при традиційному підході.

Модель системи в просторі станів може бути успішно реалізована за допомогою прямонаправлених мереж, якщо змінні стани сформувати на виході мережі і потім завести їх на вхід мережі із затримкою. Але для цього потрібний, щоб всі змінні стани були вимірювані, оскільки тільки в цьому випадку можна забезпечити тренування мережі.

NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при нейронній ідентифікації систем. Для простоти викладу надалі розглядаємо системи з одним входом і одним виходом SISO (singl input singl output), хоча всі висновки можуть бути поширені і на MIMO (multi input multi output) системи. Прямий нейронний NARX моделі нелінійної динамічної системи відповідає наступний вираз:

$$\hat{y}(k) = f\left(y(k-1)\dots y(k-n_y), u(k-n_k)\dots u(k-n_u-n_k+1), \mathbf{w}\right), \quad (\Gamma.9)$$

а нейронній інверсній NARX моделі:

$$\hat{u}(k) = g\left(y(k+n_k)\dots y(k+n_k-n_y), u(k-1)\dots u(k-n_u+1), \mathbf{w}\right) \quad (\Gamma.10)$$

де f, g – функції, що реалізуються нейронними мережами;

u, y – вхідна і вихідна координати відповідно;

$\hat{u}, \hat{y}$ – значення координат на виході мережі;

$\mathbf{w}$ – матриця лінійних параметрів мережі;

n_k – запізнювання в системі, найчастіше $n_k = 1$.

Вирази (Г.9) і (Г.10) можна представити графічно у вигляді структурних схем, показаних на рис. Г.5 і рис. Г.6. NARMAX моделі характеризуються вищою якістю ідентифікації при зашумлених тренувальних даних. У [61] пропонується спосіб модифікації критерію якості алгоритму тренування. Результати ідентифікації за допомогою NARX моделей наближаються в цьому випадку до результатів, отриманих за допомогою NARMAX моделей. Дані NARX нейромоделі можуть бути успішно застосовані для вирішення поставлених завдань.

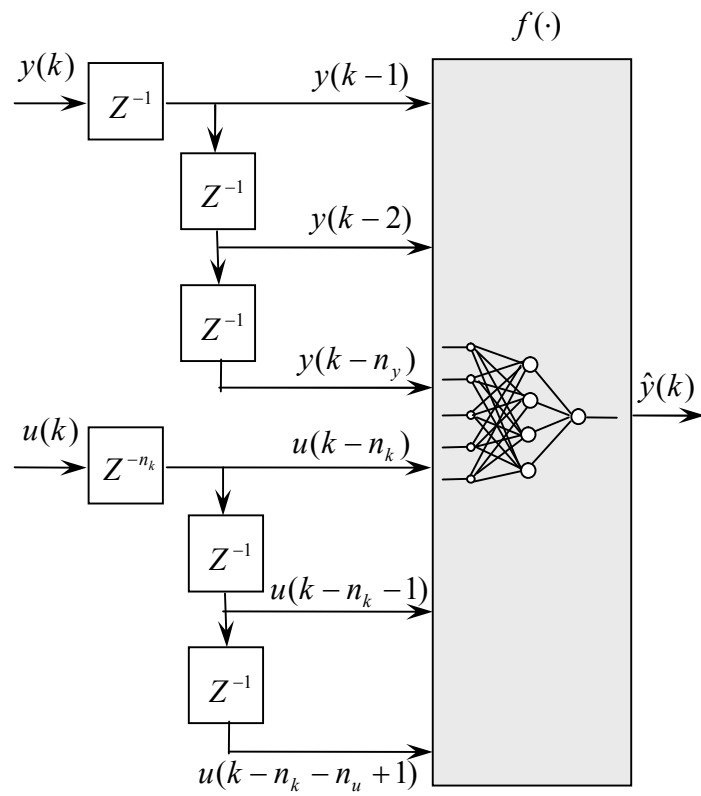


Рисунок Г.5— Пряма мережева NARX модель

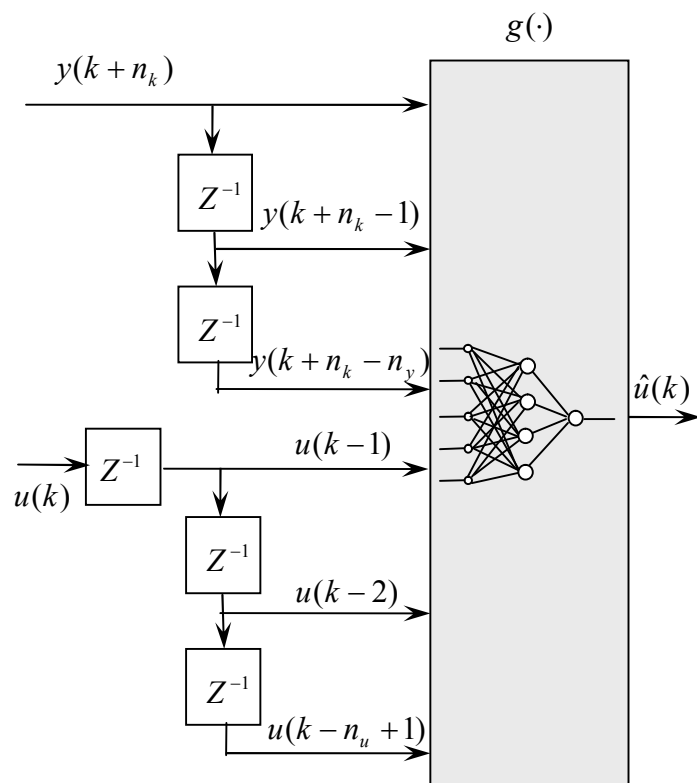


Рисунок Г.6 – Інверсна мережева NARX модель

Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж

Завдання теорії регулювання, що займається пошуком моделей динамічних систем на основі апріорної інформації і вимірюваних входних/вихідних сигналів системи, носить назву ідентифікація. Відомі класичні методи ідентифікації нелінійних систем застосовні тільки у поєднанні з певним типом моделі (наприклад, ряди Вальтера, моделі Хаммерштейна і Вінера), що описує певний клас нелінійних динамічних систем. У роботі під термінами «ідентифікація системи», «тренування» і «навчання» розуміється процес підстроювання вагових коефіцієнтів мережі за допомогою градієнтних методів. Метою цього процесу є по можливості точне наближення поведінки моделі до спостережуваної поведінки динамічної системи.

Принципово розрізняють пряму і інверсну модель об'єкту. На рис. Г.7 представлені схеми тренування цих нейронних моделей. Для спрощення на рис. Г.7 не зображені ланки затримки за часом, що формують попередні значення координат на вході нейронних мереж.

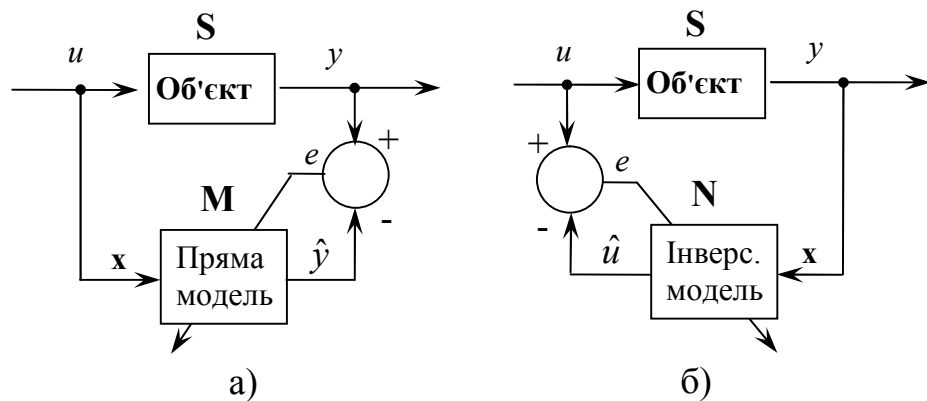


Рисунок Г.7 – Схеми тренування прямої (а) і інверсної (б) моделей

Вхідний сигнал x нейронній моделі об'єкту M на рис. Г.7,а ідентичний вхідному сигналу об'єкту S . Для інверсної нейромоделі N як вхідний сигнал використовується вихідний сигнал об'єкту S тоді як нейромодель повинна відновити вхідний сигнал об'єкту (див. рис. Г.7,б). Поки присутня помилка між виходом нейромоделі і її зразковим сигналом, ця помилка використовується для

тренування мережі. У обох випадках ідентифікації обчислюється градієнт функціонала якості (див. таблицю Г.1). Функціонал якості найчастіше при цьому квадратичний.

Таблиця Г.1 – Обчислення градієнта при тренуванні нейромоделей

	Пряма модель	Інверсна модель
Бажаний вихідний сигнал	y	u
Вихідний сигнал нейромоделі	$\hat{y} = \mathbf{M}(\mathbf{x}, \mathbf{w})$	$\hat{u} = \mathbf{N}(\mathbf{x}, \mathbf{w})$
Помилка навчання	$e = y - \hat{y}$	$e = u - \hat{u}$
Градієнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{y}}{\partial \mathbf{w}}$	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{u}}{\partial \mathbf{w}}$

Оскільки $\frac{\partial \hat{y}}{\partial \mathbf{w}}$ і $\frac{\partial \hat{u}}{\partial \mathbf{w}}$ обчислювані, то це дозволяє вести підстроювання вагових коефіцієнтів мережі згідно розглянутим градієнтним методам. Результатом цього буде мінімізація функціонала якості. Оскільки набір тренувальних даних однозначно відображає лише поведінку модельованого об'єкту $\mathbf{S}$ у робочому діапазоні, то після завершення тренування маємо $\mathbf{M} \approx \mathbf{S}$ і $\mathbf{N} \approx \mathbf{S}^{-1}$.

Типова послідовність дій, що приводить до формування нейромоделі (ідентифікації) динамічної системи, відображена в таблиці Г.2.

Таблиця Г.2 – Ідентифікація за допомогою нейронних мереж

Шаг	Дія
1	Знімання тренувальних даних
2	Вибір структури мережі
3	Вибір типу моделі
4	Тренування мережі
5	Тестування нейромоделі
6	При незадов. результаті тестування, перейти до п.3 або до п.2

Оскільки тренувальні дані є носієм інформації про систему, що ідентифікується, то наявності, кількості і якості цих даних слід приділити особливу увагу. Тренувальних дані повинні мати наступні характеристики: якісна дискретизація, рівномірність дисперсії і величини варіації, рівномірність амплітуд всіх сигналів і їх середньоквадратичних значень.

При отриманні позитивного результату апроксимації, слід також перевірити здібність нейромоделі до узагальнення, тобто її здатність повторювати істотні характеристики об'єкту. Для цього тесту необхідний ще один набір даних, що не брав участь при тренуванні мережі. Цей тест дає остаточний висновок про адекватність нейромоделі.

Представлені теоретичні положення по нейронній ідентифікації, а також характеристики прямонаправлених мереж є основою успішного регулювання нелінійних динамічних систем.

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% розвороту роторного екскаватора
% 2025 рік

% Вихідні дані
ks=8.892;
Re=0.25128;
Rd=2*0.07344;
Rcp=0;
Rs=Rd+Rcp;
Ld=2*0.0057;
Lcp=0;
Ls=Ld+Lcp;
kzn=0.061;
CF=2*3.42;
Tzc=kzn*Ls;
Ttp=0.005;
Tg=Ttp;
Te=0.047;
kzv=kzn*CF;
In=192;
Mn=CF*In;
Jd=2*2;
Jm=8.74;
b12=0;
c12=50;
Js=Jd+Jm;
Uz=76.44;
ksx=ks/(CF*(1+ks*kzn));

k1=1.0;
k2=3.0;
k3=7.0;

% 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(4,5);
```