

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

СИСТЕМИ УПРАВЛІННЯ ОСНОВНИМИ МЕХАНІЗМАМИ
МОСТОВОГО КРАНА З НЕЙРОРЕГУЛЯТОРОМ NN PREDICTIVE
CONTROLLER. ТЕМА КОМПЛЕКСНА. СПЕЦІАЛЬНА ЧАСТИНА.
СИНТЕЗ І ДОСЛІДЖЕННЯ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ
УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА

(тема кваліфікаційної роботи)

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

(код і найменування спеціальності)

_____ / Владислав П'ЯТЕНКО
(підпис) (ім'я та прізвище)

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

Рецензент _____ / Костянтин БРОВКО
(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)

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

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

студенту (ці) Владислав П'ЯТЕНКО
(ім'я, прізвище)

1. Тема роботи Системи управління основними механізмами мостового крана з нейрорегулятором NN Predictive Controller. Тема комплексна. Спеціальна частина. Синтез і дослідження нейромережевої системи управління механізмом переміщення візка
затверджена наказом по університету № 4801-5/3664 від “06” жовтня 2025 р.
2. Термін здачі закінченої роботи “5” грудня 2025 р.
3. Вихідні дані до роботи Вихідні дані до проекту (роботи) Вантажопідйомність крану $G_{вп}=37 \cdot 10^3$ кг; продуктивність крану $Q=55$ кг/с; маса моста без візка $m_m=28 \cdot 10^3$ кг; маса візка $m_v=8 \cdot 10^3$ кг; діаметр коліс моста $D_m=0,71$ м; діаметр коліс візка $D_v=0,4$ м; швидкість пересування моста $V_m=1,25$ м/с; швидкість пересування візка $V_v=0,65$ м/с; швидкість підйому $V_p=0,1$ м/с; маса вантажозахватного пристосування $m_0=1 \cdot 10^3$ кг; середня висота підйому і спуску $L_p=6$ м; середній шлях моста в одну сторону $L_m=45$ м; середній шлях візка в одну сторону $L_v=24$ м; діаметр барабана механізму підйому $D=0,65$ м. Перерегулювання швидкості по задаючій дії 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 підйому мостового крана. Розрахувати систему підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. 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 системи з нейрорегулятором.
5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістрська кваліфікаційна робота. Формальні ознаки. 2. Механізм переміщення візка. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 5. Нейронні мережі. Ідентифікація об'єктів. 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. Магістрська 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)

Тетяна ВАСИЛЕЦЬ
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Владислав П'ЯТЕНКО
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму переміщення візка мостового крана.	09.09.2025-18.09.2025	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження динамічних характеристик системи.	19.09.2025-15.10.2025	
3	Дослідження системи з урахуванням поперечних розгойдувань вантажу. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2025-05.11.2025	
4	Синтез нейромережової системи управління на ЕОМ 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)

Владислав П'ЯТЕНКО
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 213 сторінок, 67 рисунків, 3 таблиці, 83 використаних джерел, 5 додатків.

Метою роботи є розробка методів синтезу системи керування електроприводом механізму переміщення візка мостового крана, спрямованих на підвищення показників якості функціонування за рахунок використання нейромережевих регуляторів.

Параметри системи автоматичного керування визначено на основі класичних принципів теорії автоматичного регулювання. Методи математичного моделювання застосовано для побудови моделей багатомасової електромеханічної системи. На основі положень теорії штучних нейронних мереж виконано синтез нейрорегулятора, призначеного для керування двомасовою системою. За допомогою комп'ютерного моделювання в середовищі MATLAB проведено дослідження динамічних властивостей нейромережевої системи.

У ході виконання магістерської роботи доведено доцільність використання нейромережевих технологій для керування багатомасовим електроприводом механізму переміщення візка мостового крана. Розроблено систему керування координатами електроприводу та виконано розрахунок її параметрів. Проведено моделювання роботи системи на ЕОМ.

Створено математичну модель, яка враховує пружні властивості механічних елементів електроприводу. Отримано динамічні характеристики двомасової системи. Досліджено структуру нейромережевої системи керування та здійснено синтез регулятора NN Predictive Controller у середовищі MATLAB. Здійснено моделювання роботи системи з нейрорегулятором на ЕОМ, за результатами якого встановлено, що синтезована нейромережева система забезпечує високі показники якості регулювання.

Напрацювання магістерської роботи активно використовуються у навчальному процесі кафедри АМЕТ ННІ «УІПА» ХНУ ім. В.Н. Каразіна – під час читання лекцій, виконання курсових проєктів та проведення практичних занять студентами.

Основні результати дослідження пройшли апробацію на науково-практичній конференції студентів ННІ «УІПА» ХНУ ім. В.Н. Каразіна (Харків, 2025 р.).

Ключові слова: ШТУЧНИЙ НЕРОН, НЕЙРОННІ МЕРЕЖІ, МОСТОВИЙ КРАН, ЕЛЕКТРОПРИВОД МЕХАНІЗМУ ПЕРЕМІЩЕННЯ ВІЗКА, СИСТЕМА ПІДЛЕГЛОГО УПРАВЛІННЯ, ОДНОМАСОВА І ДВОМАСОВА СИСТЕМИ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОРЕГУЛЯТОР

ABSTRACT

An explanatory message consists of: 213 pages, 67 figures, 2 tables, 83 used sources, 5 additions.

The aim of the work is to develop methods for synthesizing the control system for the electric drive of the overhead crane trolley movement mechanism, aimed at improving the performance quality indicators through the use of neural network controllers.

The parameters of the automatic control system are determined based on the classical principles of the theory of automatic control. Mathematical modeling methods are used to build models of a multi-mass electromechanical system. Based on the principles of the theory of artificial neural networks, a neuroregulator designed to control a two-mass system is synthesized. Using computer modeling in the MATLAB environment, the dynamic properties of the neural network system are studied.

During the master's thesis, the feasibility of using neural network technologies to control a multi-mass electric drive of the overhead crane trolley movement mechanism is proven. A coordinate control system for the electric drive has been developed and its parameters have been calculated. The system operation has been simulated on a computer.

A mathematical model has been created that takes into account the elastic properties of the mechanical elements of the electric drive. The dynamic characteristics of the two-mass system have been obtained. The structure of the neural network control system was studied and the NN Predictive Controller was synthesized in the MATLAB environment. The system operation with the neuroregulator was simulated on a computer, the results of which showed that the synthesized neural network system provides high control quality indicators.

The results of the master's thesis are actively used in the educational process of the Department of AMET of the Scientific Research Institute "UIPA" of the V.N. Karazin KhNU - during lectures, course projects and practical classes by students.

The main results of the study were tested at the scientific and practical conference of students of the Scientific Research Institute "UIPA" of the V.N. Karazin KhNU (Kharkiv, 2025).

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, BRIDGE CRANE, ELECTRIC DRIVE OF THE MOVING TRUCK MECHANISM, SYSTEM OF SLAVE CONTROL, ONE-MASS AND TWO-MASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

P_n – номінальна потужність;

U_n – номінальна напруга;

I_n – номінальний струм;

M_n – номінальний момент;

n_n – номінальна швидкість обертання;

Φ_n – номінальний магнітний потік;

R – активний опір;

L – індуктивність;

J – момент інерції;

k_d – коефіцієнт посилення двигуна;

T_e – електромагнітна постійна часу електроприводу;

T_m – електромеханічна постійна часу електроприводу;

$W(p)$ – передатна функція;

c – коефіцієнт жорсткості;

β – коефіцієнт внутрішнього в'язкого тертя;

A – матриця стану;

B – матриця управління;

C – матриця виходу;

$\vec{X}(t)$ – вектор стану системи;

$\vec{U}(t)$ – вектор управління;

$\vec{Y}(t)$ – вектор виходу;

САУ – система автоматичного управління;

М – двигун.

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД НЕЙРОМЕРЕЖЕВИХ СИСТЕМ КЕРУВАННЯ ТА ФОРМУЛЮВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1. Сучасний стан і перспективи розвитку штучних нейронних мереж...	14
1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи	17
1.3 Основні характеристики мостових кранів.....	34
1.4 Обґрунтування вибору системи електроприводу механізму переміщення моста мостового крана	36
1.5 Постановка задачі синтезу нейромережевої системи управління	39
ВИСНОВКИ ДО РОЗДІЛУ 1	43
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА МОСТОВОГО КРАНА	45
2.1 Технічні характеристики мостового крана та вимоги до системи управління.....	45
2.2 Розрахунок потужності електроприводу і вибір електродвигуна.....	46
2.3 Розрахунок швидкісної діаграми електроприводу	48
2.4 Розрахунок діаграми навантаження електроприводу	50
2.5 Перевірка правильності попереднього вибору електродвигуна	53
2.6 Вибір тиристорного перетворювача і розрахунок параметрів силового кола електроприводу	54
2.7 Вибір структури системи управління електроприводом	60
2.8 Синтез послідовного коректуючого пристрою контуру струму.....	61
2.9 Синтез послідовного коректуючого пристрою контура ЕРС.....	65
2.10 Дослідження динамічних характеристик системи	73
ВИСНОВКИ ДО РОЗДІЛУ 2	78
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПОПЕРЕЧНИХ РОЗГОЙДУВАНЬ ВАНТАЖУ	80

3.1 Механічна частина електроприводу	80
3.2 Рівняння стану одномасової системи.....	82
3.3 Математична модель двомасової системи	86
3.4 Розрахунок перехідних процесів двомасової системи.....	94
ВИСНОВКИ ДО РОЗДІЛУ 3	104
РОЗДІЛ 4. СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	106
4.1 Регулятори, реалізовані засобами пакета Neural Network Toolbox у середовищі MATLAB	106
4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller	108
4.3 Проєктування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB	111
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	158
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller.....	160
ВИСНОВКИ ДО РОЗДІЛУ 4	166
ВИСНОВКИ	168
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	170
ДОДАТОК ДОДАТОК А. МОДЕЛЬ НЕЙРОНА	178
А.1 Структурна схема нейрона	178
А.2 Функції активації нейронів	179
А.3 Нейрон з векторним входом.....	182
ДОДАТОК Б. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ.....	184
Б.1 Одношарові мережі.....	184
Б.2 Багатошарові мережі	186
Б.3 Мережі з прямою передачею сигналу	189
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	191
В.1 Завдання навчання.....	191
В.2 Процедури адаптації і навчання	192
В.3 Явище перенавчання	194

В.4 Властивість узагальнення.....	197
В.5 Методи навчання нейронних мереж.....	199
ДОДАТОК Г. ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУ- ВАННЯМ НЕЙРОННИХ МЕРЕЖ.....	204
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	213

ВСТУП

Актуальність теми. Підвищення ефективності автоматизованих систем керування промисловими установками є одним з ключових чинників розвитку сучасного виробництва. Удосконалення методів керування забезпечує зростання продуктивності, покращення якості технологічних процесів та зменшення енерговитрат. Еволюція систем автоматичного керування охоплює розширення можливостей адаптивних структур, розвиток векторних методів керування електроприводами змінного струму, активне впровадження багаторівневих обчислювальних систем, а також широке використання цифрових методів регулювання. Одним із найбільш перспективних напрямів сьогодення є впровадження систем керування на основі штучних нейронних мереж.

Нейромережеві методи належать до класу нелінійних динамічних систем, у яких штучна нейронна мережа може виконувати різні функції: від адаптивного регулятора до ідентифікатора або оптимізатора параметрів. При цьому можливі два основні підходи до роботи нейромережі.

Перший передбачає її паралельне навчання та керування об'єктом у режимі реального часу (on-line), коли мережа одночасно удосконалює власні параметри та формує керуючі впливи.

Другий – офлайнний (off-line) підхід, при якому спочатку здійснюється навчання мережі на апіорних даних, а вже після цього вона використовується як регулятор, що відтворює оптимальну поведінку за близьких до навчання умов. Останній спосіб, відомий як супервізорне управління, на сьогодні є найбільш поширеним, хоча його недоліком є відсутність адаптації в реальному часі.

Вибір стратегії навчання (on-line/off-line) визначається характером технічного об'єкта та доступністю даних. Для систем, про які накопичено великі масиви інформації, доцільним є off-line навчання із застосуванням алгоритмів глобальної оптимізації (генетичних, статистичних, методів випадкового пошуку тощо). Якщо ж параметри об'єкта змінюються під час роботи, оптимальнішим є on-line налаштування.

Нейронні мережі також використовуються для ідентифікації динамічних систем, оцінювання внутрішніх змінних, розширеної фільтрації Калмана, автоматичного підстроювання параметрів класичних регуляторів та для синтезу адаптивних законів керування. Усі системи такого типу об'єднують під поняттям нейромережевих систем управління. Найбільш поширеними серед них є багаточарові персептрони та інші нелінійні прямонаправлені мережі [1-12].

У промисловості України сьогодні активно розробляються нові та вдосконалюються існуючі підйомно-транспортні механізми, зокрема автоматизовані комплекси та конвеєрні лінії. Мостові крани залишаються одними з ключових елементів систем механізації, а якість їх роботи багато в чому визначається ефективністю електроприводів та систем керування. Робота електроприводів у режимах частих пусків, змінних навантажень і високих динамічних навантажень потребує удосконалених систем регулювання.

Незважаючи на великий досвід проектування та модернізації кранових електроприводів, питання підвищення їх ефективності неможливо розв'язати без удосконалення систем автоматичного керування. Висока якість регулювання забезпечує раціональні режими роботи механізму, довговічність обладнання та стабільність виробничого процесу. У зв'язку з цим розроблення інтелектуальних систем керування електроприводами мостових кранів є актуальним та перспективним напрямом розвитку галузі.

Особливої значущості набувають системи, побудовані на основі адаптивних і нейромережевих технологій, що здатні забезпечити високу точність керування у складних багатомасових механічних структурах з пружними зв'язками.

Мета і завдання дослідження. Метою магістерської роботи є розроблення методів синтезу системи керування електроприводом переміщення візка мостового крана на основі нейромережевого регулятора, що забезпечує підвищення якості роботи електромеханічної системи.

Для досягнення поставленої мети необхідно розв'язати такі завдання:

1. Провести аналіз сучасних систем керування та оцінити можливості застосування нейромережевих методів.

2. Розробити математичну модель динаміки об'єкта керування.
3. Створити структурну схему системи керування з нейромережовим регулятором.
4. Виконати синтез нейромережевої системи на основі прогнозуючого регулятора та забезпечити високу якість регулювання.
5. Провести моделювання роботи системи на ЕОМ і виконати аналіз отриманих результатів.

Об'єкт і предмет дослідження.

Об'єкт дослідження – динамічні процеси в системі керування електроприводом механізму переміщення візка мостового крана з урахуванням поперечних коливань вантажу.

Предмет дослідження – математичні моделі та методи синтезу нейромережевої системи керування двомасовою електромеханічною системою з урахуванням поперечних коливань вантажу.

Методи дослідження. Теоретична частина роботи ґрунтується на фундаментальних положеннях теорії автоматичного керування. Для побудови математичних моделей використовуються методи математичного моделювання. Створення нейромережових моделей базується на методах теорії штучних нейронних мереж, а синтез нейромережевого регулятора реалізується за допомогою методів нейроуправління. Якість роботи системи оцінюється на основі імітаційного моделювання.

Наукова новизна. Наукова новизна отриманих результатів полягає в такому:

1. Уперше надано наукове обґрунтування ефективності застосування нейромережових технологій у системах керування багатомасовим електроприводом механізму переміщення візка мостового крана.
2. Запропоновано нову нейромережеву систему керування електроприводом з урахуванням пружних властивостей механічних елементів, що забезпечує підвищення якості функціонування.

3. Уперше побудовано математичну модель системи у вигляді багатошарового персептрона, яка враховує поперечні коливання вантажу.

4. Розвинуто математичні моделі об'єктів керування із складними динамічними зв'язками шляхом подання їх у формі двомасових електромеханічних систем.

5. Удосконалено методи синтезу нейромережевих структур з прогнозом для керування об'єктами зі складною динамікою.

Практичне значення. Практична цінність роботи полягає у створенні нейромережевої системи керування двомасовою електромеханічною системою приводу переміщення візка мостового крана, яка забезпечує підвищені показники точності та якості регулювання.

Матеріали роботи впроваджені в освітній процес кафедри АМЕТ ННІ «УІПА» ХНУ ім. В. Н. Каразіна та використовуються під час лекцій, практичних занять і виконання курсових проєктів.

Особистий внесок здобувача. Усі основні результати роботи, винесені на захист, отримані магістром особисто. До них належать:

- синтезована система регулювання координат електроприводу;
- розроблена модель системи з урахуванням пружних елементів;
- розроблена структурна схема нейромережевої системи управління;
- синтезований нейрорегулятор NN Predictive Controller з використанням пакету прикладних програм Neural Network Toolbox системи MATLAB;
- виконане моделювання нейромережевої системи та проведено аналіз отриманих результатів.

У роботі наведені посилання на авторів і джерела під час використання відомих залежностей, експериментальних даних та наукових положень.

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

Структура і об'єм роботи. Робота складається зі вступу, чотирьох розділів основної частини, висновків, додатків та списку використаних літературних джерел.

Повний обсяг роботи становить 213 сторінок, серед яких: 67 рисунків, у тому числі 39 рисунків у тексті та 28 рисунки на 26 окремих сторінках; 3 таблиці; 5 додатків обсягом 36 сторінок; список використаних літературних джерел, що містить 83 найменування на 8 сторінках.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД НЕЙРОМЕРЕЖЕВИХ СИСТЕМ КЕРУВАННЯ ТА ФОРМУЛЮВАННЯ ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1. Сучасний стан і перспективи розвитку штучних нейронних мереж

Штучні нейронні мережі (ШНМ) нині посідають провідне місце серед технологій штучного інтелекту та є одним із найбільш інтенсивно досліджуваних напрямів комп'ютерних наук і теорії керування. Вони демонструють високі результати у розпізнаванні зображень, комп'ютерному зорі, обробленні природної мови, технічній діагностиці, прогнозуванні, оптимізації та автоматичному керуванні складними динамічними об'єктами [13–44].

Разом із тим розвиток нейронних мереж продовжується надзвичайно швидкими темпами, що зумовлює постійне розширення теоретичної бази та практичних галузей застосування.

Одним із ключових трендів сучасних досліджень є підвищення інтерпретованості та пояснюваності нейронних моделей. Глибинні мережі, що складаються з десятків чи сотень шарів, нерідко працюють як «чорний ящик», і користувачі не завжди можуть зрозуміти принцип формування рішень. Саме тому активно розвиваються підходи Explainable AI (XAI), спрямовані на покращення прозорості функціонування моделей, що є критично важливим для промислових систем, медицини, транспорту та інших сфер із підвищеними вимогами до безпеки.

Ще одним актуальним напрямом є оптимізація обчислювальної ефективності нейромереж. Навчання масштабних моделей вимагає великих обчислювальних потужностей, що ускладнює їх використання у вбудованих мікропроцесорних системах та реальних промислових контролерах.

Тому широкого поширення набули методи: quantization – зменшення точності представлення вагових коефіцієнтів; knowledge distillation – передавання

знань від більшої моделі до меншої; sparse training – формування розріджених моделей; edge-AI – перенесення інтелектуальних обчислень на крайові пристрої.

На особливу увагу заслуговує напрям нейроморфних обчислень, у межах якого створюються апаратні платформи, що імітують принципи функціонування біологічних нейронних мереж.

Технології spiking neural networks (SNN), апаратні рішення Intel Loihi, IBM TrueNorth та інші здатні забезпечити надзвичайно високу енергоефективність і швидкість оброблення інформації, що відкриває нові можливості для автономних керуючих систем.

У галузі керування складними технічними об'єктами спостерігається чітка тенденція до інтеграції нейромереж із класичними методами теорії автоматичного регулювання, такими як оптимальне, адаптивне, робастне та предиктивне керування.

Таке поєднання дозволяє отримати системи, що одночасно володіють гнучкістю нейронних моделей та математично обґрунтованою стійкістю.

Значні перспективи має також нейросимволічний підхід, у межах якого нейромережі поєднуються з логічними та символічними структурами, що дозволяє системам не лише розпізнавати інформацію, але й здійснювати логічні міркування, планування та робити висновки.

Важливою складовою розвитку є нейроеволюція – використання генетичних алгоритмів для автоматичного проєктування архітектур нейронних мереж, що значно спрощує процес пошуку оптимальних структур і параметрів для конкретних задач керування.

Узагальнюючи, можна зробити висновок, що розвиток ШНМ відбувається в напрямку підвищення їх адаптивності, прозорості, продуктивності та надійності, що забезпечує їх широке впровадження в автоматизовані системи управління промисловими об'єктами, включно з електроприводами, робототехнічними комплексами, мехатронними системами та інтелектуальними транспортними засобами.

Перспективи розвитку штучних нейронних мереж у системах автоматизованого керування.

Застосування ШНМ у задачах керування набуває все більшого поширення завдяки їхній здатності ефективно апроксимувати складні нелінійні залежності, адаптуватися до змінних умов експлуатації та працювати в умовах невизначеності.

Одним із найперспективніших напрямів є нейромережеве адаптивне керування, у межах якого ШНМ використовуються як адаптивні регулятори або як моделі об'єктів, що дозволяє системі самостійно підлаштовуватися під зовнішні впливи та зміну параметрів механізмів.

Особливий інтерес викликають методи Neural Model Predictive Control (NMPC), що поєднують предиктивну оптимізацію з нейромережевими моделями.

У таких системах ШНМ здійснює прогноз поведінки об'єкта, що дає змогу визначати оптимальні дії в реальному часі з урахуванням обмежень, нелінійностей та зовнішніх збурень.

У сучасних дослідженнях активно розвиваються комбіновані архітектури, у яких нейронні мережі інтегруються із структурами традиційних регуляторів (PID, LQR, MPC, Sliding Mode та ін.), що дозволяє поєднати легкість реалізації класичних алгоритмів і гнучкість нейронних моделей.

Серед інших актуальних напрямів:

- розробка стійких нейромережевих контролерів з математично гарантованими властивостями стабільності на основі функцій Ляпунова;
- створення онлайнових алгоритмів навчання для роботи в режимах реального часу;
- застосування еволюційних підходів для синтезу структури ШНМ;
- використання ШНМ у мікроконтролерних системах для автономних інтелектуальних механізмів.

Враховуючи це, у майбутньому очікується створення повністю автономних інтелектуальних систем керування, що здатні до самонавчання, адаптації та роботи в складних умовах невизначеності.

Матеріали, наведені у додатках.

У додатках А–Г наведено теоретичний базис, необхідний для виконання досліджень:

- Додаток А – структура штучного нейрона, його математичні моделі та функції активації.
- Додаток Б – основні архітектури штучних нейронних мереж, включаючи багатoshарові прямонаправлені мережі (MLP), рекурентні мережі, радіально-базисні мережі та інші.
- Додаток В – методи навчання ШНМ і обґрунтування вибору конкретного алгоритму навчання для синтезу нейромережевої системи керування у цій роботі.
- Додаток Г – питання ідентифікації динамічних систем із використанням нейронних мереж, що є критично важливими для створення адекватних моделей складних електромеханічних об'єктів.

1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи

У сучасній спеціальній літературі з нейрорегулювання досить часто використовуються терміни «пряме» та «непряме» регулювання, при цьому їх значення можуть трактуватися по-різному. Тому доцільно відразу уточнити ці поняття та послідовно застосовувати їх у подальшому викладі.

Загальноприйнято розрізняти методи регулювання з прямою та непрямою адаптацією залежно від способу підстроювання параметрів регулятора:

- Пряма адаптація полягає у безпосередній зміні параметрів регулятора з метою забезпечення виконання певного критерію якості.
- Непряма адаптація передбачає попередню ідентифікацію параметрів моделі об'єкта керування. Уже на основі цих параметрів здійснюється синтез регулятора.

Саме цей підхід покладено в основу поданого нижче огляду структур систем нейрорегулювання.

Ще одним важливим поняттям, що часто зустрічається у літературі, є поділ процесів навчання нейронної мережі на online та offline режими.

- Offline-тренування виконується до запуску системи в реальному часі. Усі дані для навчання мають бути зібрані заздалегідь.
- Online-тренування безпосередньо пов'язане з функціонуванням системи у режимі реального часу (*real-time control*). У цьому випадку введення даних, їх обробка та формування керуючих дій відбуваються в межах одного періоду дискретності. Його значення визначається, перш за все, характеристиками об'єкта керування, а не доступною обчислювальною потужністю.

Відповідно, online-навчання означає, що хоча б одна епоха тренування мережі виконується одночасно з регулюванням, і весь процес вкладається в один період дискретності.

Схема взаємодії процесів регулювання та online-тренування нейронної мережі наведена на рис. 1.1 [45].

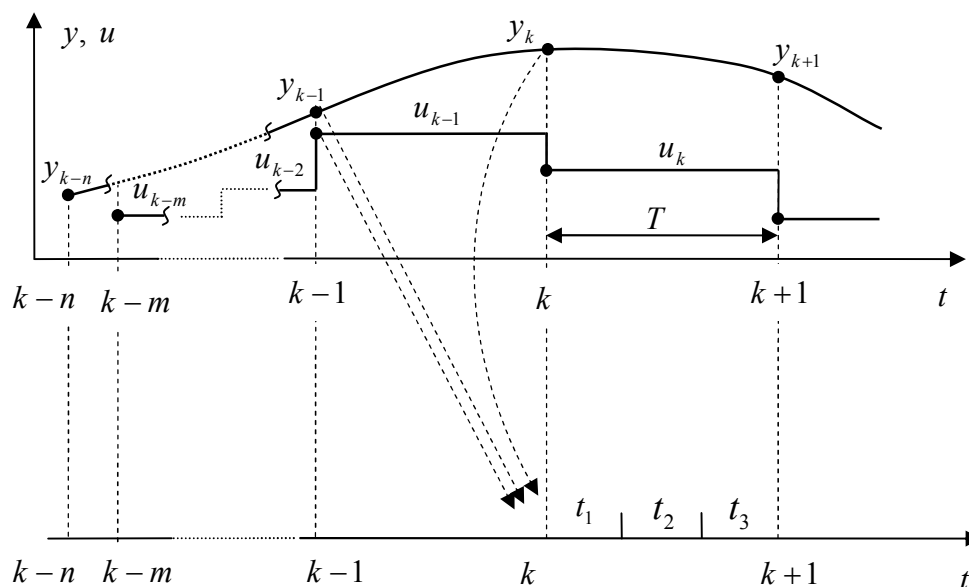


Рисунок. 1.1 – Регулювання і тренування мережі в реальному часі

На рисунку використані такі позначення: y – регульована координата; u – дія на об'єкт; T – період дискретності; t_1 – попередня обробка сигналів датчи-

ків; t_2 – тренування нейронної мережі; t_3 – обчислення та видача керуючої дії; m, n – кількість попередніх значень змінних.

На сьогоднішній день запропоновано та експериментально перевірено велику кількість структур систем нейрорегулювання. Частина з них базується на використанні аналітичного закону регулювання у поєднанні з нейромоделлю об'єкта. У таких випадках нейронна мережа використовується для відновлення (апроксимації) елементів матриць моделі, що належить до структурованих методів регулювання.

Проте найбільший потенціал штучних нейронних мереж, зокрема їх здатність до універсальної апроксимації, розкривається у неструктурованих методах нейрорегулювання, де структура регулятора не задається аналітично, а формується самою нейронною мережею під час навчання.

У подальших підрозділах стисло розглянуто найбільш актуальні структури систем нейрорегулювання, що знайшли застосування у сучасних задачах керування.

1.2.1 Пряме інверсне регулювання

Термін Direct Inverse Control був введений Вербосом [46]. Пряме інверсне нейрорегулювання є одним із найпоширеніших методів у сучасних системах керування (рис. 1.2). Тренування нейронної мережі при використанні цього підходу може здійснюватися як offline, так і online. Згідно з класифікацією Псалтіса та ін. [47], ці варіанти відповідно називають general learning та specialized learning. Відмінності структур прямого інверсного регулювання для offline та online тренування розглянуто нижче.

Offline-тренування. Пряме інверсне регулювання з offline тренуванням ґрунтується на припущенні, що існує інверсне відображення простору виходів об'єкта у простір його входів. Спочатку в режимі offline формують інверсну модель динаміки об'єкта **NC** (перемикачі на рис. 1.2 у нижньому положенні). Після завершення тренування отримана інверсна модель підключається послі-

довно з об'єктом регулювання (перемикачі у верхньому положенні) і виконує функції регулятора.

Таким чином, назва методу «*пряме інверсне регулювання*» повністю відповідає його сутності: регулятор реалізує апроксимовану інверсну модель об'єкта.

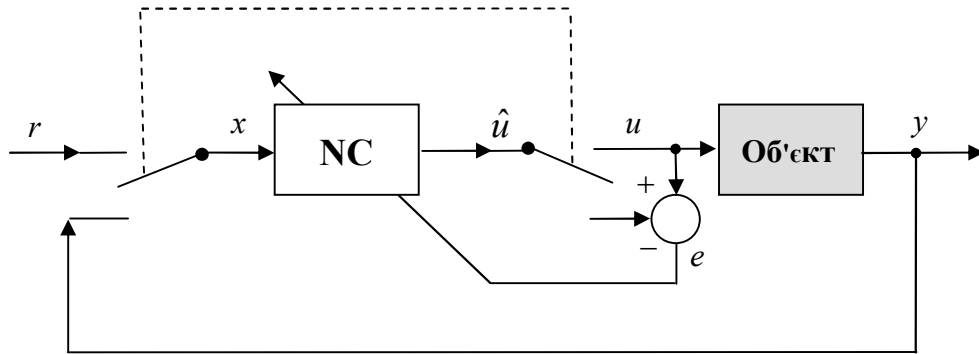


Рисунок 1.2 – Пряме інверсне регулювання з offline тренуванням

Недоліком такої структури є низька робастність, обумовлена відсутністю явного зворотного зв'язку. Отже, для успішного застосування цього підходу необхідною умовою є стійкість самого об'єкта регулювання. На практиці виконання цієї умови не завжди гарантується.

Online-тренування. Структура системи прямого інверсного регулювання з тренуванням у режимі online наведена на рис. 1.3.

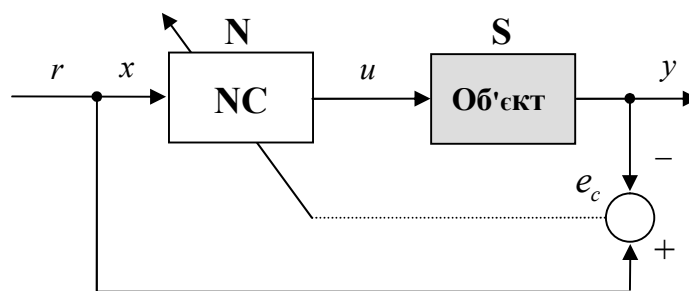


Рисунок 1.3 – Пряме інверсне регулювання з online тренуванням

Метою тренування нейрорегулятора **NC** є генерація такого керуючого сигналу u , який забезпечує відстеження регульованою координатою y сигналу

завдання r . Регулятор навчається безперервно та одночасно з процесом керування, що робить цей підхід придатним для об'єктів зі змінними параметрами.

Проблеми online-тренування. Головним недоліком online-навчання є необхідність коректного обчислення помилки регулювання e_c , оскільки ця помилка формується на виході об'єкта, а не на виході нейромережевого регулятора. Пунктирна лінія на рис. 1.3 відображає виникнення відповідного «пропуску» у зворотному поширенні помилки. Формули для обчислення градієнта наведені в таблиці 1.1.

Таблиця 1.1 – Визначення градієнта при регулюванні з online тренуванням

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

Похідна $\frac{\partial u}{\partial \mathbf{w}}$ може бути обчислена завжди. Компонента $\frac{\partial y}{\partial u}$ є матрицею Якобі. Для відомих об'єктів керування матриця Якобі може бути отримана аналітично, і її елементи легко обчислюються.

Однак у більшості практичних ситуацій об'єкт регулювання точно невідомий. У цьому випадку похідну можна апроксимувати за допомогою різницевого рівняння [47]:

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

Аналогічно до методів лінійного адаптивного керування запропоновано також використовувати знак похідної [48], якщо про об'єкт відома хоча б якісна інформація (наприклад, монотонність).

Проте обидва методи є малоефективними при роботі в умовах сильного шуму.

Важливо зазначити, що застосування нетренованого регулятора в розімкненому контурі на початковому етапі може призвести до неконтрольованої або аварійної поведінки системи. Щоб уникнути цього, необхідно виконати хоча б грубе попереднє (пре-) тренування нейрорегулятора перед переходом до online-адаптації.

1.2.2 Непряме інверсне нейрорегулювання

У спеціальній літературі цей підхід має кілька термінологічних інтерпретацій. Так, Наренда у своїх роботах позначає його як feedforward inverse control [49], тоді як Хант і співавтори застосовують назву specialized inverse learning [50]. Незважаючи на варіації назв, концептуальна суть методу залишається незмінною. Узагальнена структурна схема системи непрямого інверсного нейрорегулювання наведена на рис. 1.4.

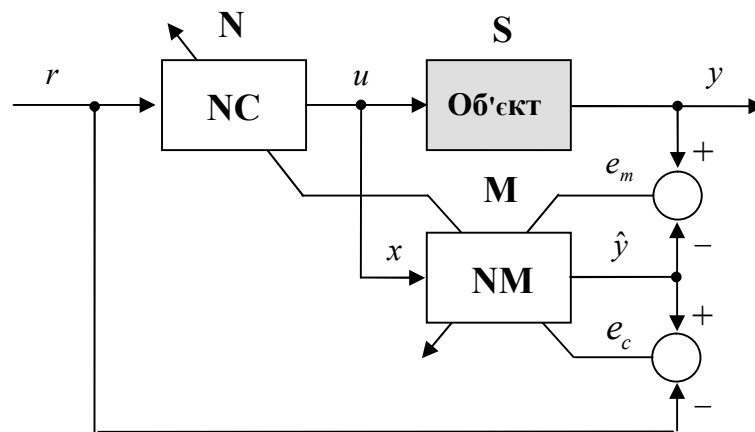


Рисунок 1.4 – Структура системи непрямого інверсного нейрорегулювання

Непряме інверсне нейрорегулювання реалізує двохетапний принцип, який передбачає розділення процесів моделювання динаміки об'єкта та синтезу керуючої дії.

Перший етап – побудова нейромоделі об'єкта. На початковій стадії здійснюється тренування нейромоделі об'єкта **NM** у режимі offline. Метою є отри-

мання адекватної апроксимації внутрішньої динаміки об'єкта на основі доступних експериментальних або синтетичних даних. Після успішного навчання нейромодель здатна відтворювати поведінку об'єкта з необхідною точністю в досліджуваному діапазоні входів.

Для подальшого синтезу керування важливо, що у момент часу τ на основі натренованої моделі можна обчислити матрицю Якобі, використовуючи правило підстановки:

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

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

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

$x_0 = u$ – один із вхідних сигналів мережі.

Отже, нейромодель надає необхідну інформацію про чутливість виходу об'єкта до зміни керуючого сигналу. Саме ця чутливість є основою для застосування алгоритмів градієнтного підстроювання ваг у наступній фазі.

Другий етап – синтез і підстроювання нейрорегулятора.

Завдяки наявності матриці Якобі помилка регулювання e_c може бути використана для адаптивної корекції вагових коефіцієнтів нейрорегулятора. Таким чином формування оптимальної дії на об'єкт відбувається з урахуванням локальних властивостей моделі, отриманих у попередній фазі.

Однак цей підхід має суттєві обмеження. Якщо нейрорегулятор починає генерувати керуючий сигнал поза діапазоном, у якому здійснювалося тренування нейромоделі, то остання стає некоректною апроксимацією реального об'єкта. У такому випадку обчислена матриця Якобі може бути неточною, що призводить до помилок у підстроюванні регулятора і потенційної нестабільності всієї системи.

Проблеми та модифікації методу. У роботах Наренди та ін. [49] запропонована модифікація, у якій нейромодель **NM** підлягає додатковому оновленню вже в режимі online. Такий підхід покликаний компенсувати обмеження offline-моделі та забезпечити її актуальність під час роботи в умовах змінної динаміки об'єкта.

Проте реалізація online-тренування нейромоделі разом із адаптацією нейрорегулятора створює низку складнощів:

1. Проблеми стійкості на початкових етапах. У перші моменти роботи система може знаходитися далеко від робочої точки, а обидві мережі – модель та регулятор – є ще ненавченими. Це може призвести до нестійкої або хаотичної поведінки об'єкта.

2. Відсутність явного зворотного зв'язку. Структура класичного непрямого інверсного регулювання не передбачає чіткого стабілізуючого каналу зворотного зв'язку, що обмежує робастність системи.

Удосконалені структури: підхід “parallel compensator”. Для подолання зазначених недоліків Ішигуро [51] запропонували вдосконалену архітектуру, відому як parallel compensator. У цій структурі:

- нейронна мережа формує паралельний компенсуючий сигнал,
- базовий класичний регулятор забезпечує первинну стабільність,
- сигнал для тренування нейромережі формується опосередковано як різниця між фактичною дією на об'єкт та вихідним сигналом аналітичної моделі об'єкта.

Таким чином, у системі з'являється непрямий, але стабілізуючий канал зворотного зв'язку, який значно покращує її робастність і запобігає дестабілізації на ранніх етапах тренування.

1.2.3 Регулювання з внутрішньою моделлю.

Метод регулювання з внутрішньою моделлю (Internal Model Control, IMC) належить до класичних підходів сучасної теорії автоматичного управління. Концепція була запропонована Гарсія та Морарі [52] для систем з аналітични-

ми, зокрема нелінійними, моделями. Сутність підходу полягає у тому, що для підвищення якості керування використовуються **дві моделі об'єкта** – пряма модель і модель-інверсія, що дозволяє компенсувати динаміку об'єкта та передбачати його реакцію на керуючі дії.

У контексті нейромережових технологій метод ІМС є яскравим прикладом інтеграції класичних регуляторних структур із можливостями штучних нейронних мереж. Використання нейромоделей відкриває можливість апроксимації складних, нелінійних та частково невизначених динамічних систем, що суттєво розширює область застосування концепції ІМС. Узагальнена структурна схема нейромережової системи регулювання з внутрішньою моделлю наведена на рис. 1.5.

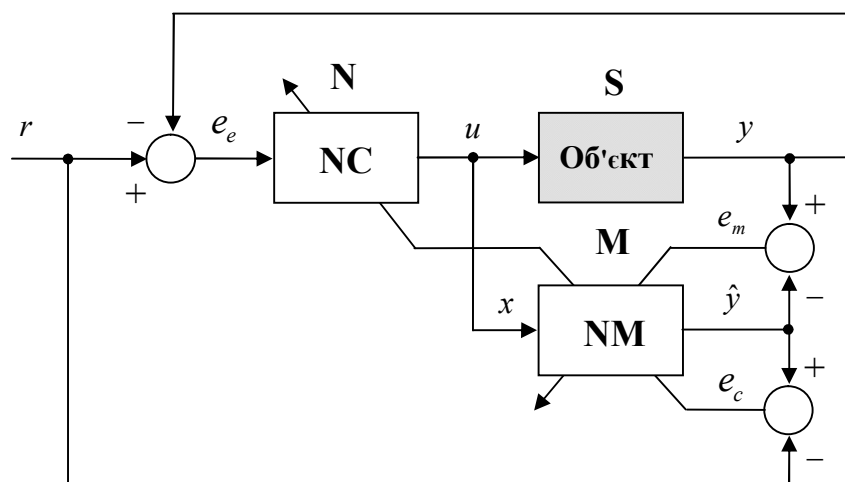


Рисунок 1.5 – Структурна схема нейромережової системи регулювання на основі внутрішньої моделі

Принцип функціонування нейромережевого ІМС. На першому етапі, аналогічно до непрямого інверсного нейрорегулювання, виконується offline-тренування прямої нейромоделі об'єкта **NM**. Метою цього етапу є відтворення динаміки об'єкта у всьому робочому діапазоні сигналів, що дозволяє забезпечити коректну роботу регулятора.

Після успішного навчання прямої моделі на її основі формується матриця Якобі, необхідна для обчислення градієнтів і підстроювання ваг інверсної нейромережі-регулятора **NC**.

Корекція параметрів регулятора виконується в онлайн-режимі, що забезпечує адаптивність системи до змінних умов та параметричних збурень.

На відміну від непрямих схем, в ІМС обидві нейромережеві моделі входять до контуру зворотного зв'язку: помилка, що виникає між виходом об'єкта і виходом його прямої моделі e_m , безпосередньо подається на вхід нейрорегулятора. Це створює додатковий компенсуючий канал, який підвищує точність відстежування та дозволяє частково компенсувати модельні похибки.

Особливі властивості та теоретичні результати. Нейромережевий ІМС має низку важливих теоретичних переваг. Якщо модель об'єкта є ідеальною, тобто $\mathbf{M} = \mathbf{S}$, тоді схема ІМС еквівалентна структурі непрямого інверсного регулювання із розімкненим контуром ($e_m = 0$). У цьому випадку можливим є формальне доведення стійкості системи на основі методів класичної теорії керування.

Ще одна суттєва властивість полягає у тому, що при асимптотично сталих збуреннях і сталому сигналі завдання статична похибка регулювання дорівнює нулю ($y = r$), якщо нейрорегулятор є точною інверсною ланкою до моделей об'єкта $\mathbf{N} = \mathbf{M}^{-1}$. Це впливає з фундаментального принципу внутрішньої моделі:

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

Таким чином, якість прямої моделі не є визначальною для забезпечення нульової статичної помилки – ключовим залишається коректність побудови інверсної моделі.

Обмеження та проблеми практичної реалізації. Попри привабливі теоретичні властивості, використання нейромережевої структури ІМС має низку суттєвих обмежень:

1. Застосовність лише до стійких об'єктів. Оскільки модель «віднімається» з реального об'єкта у структурі зворотного зв'язку, навіть незначні похибки можуть спричинити розбалансування системи та посилення нестійких мод.

2. Необхідність фільтра низьких частот. У реальних умовах апроксимація точної моделі недосяжна, тому на вході компенсатора доцільно встановлювати

фільтр, що згладжує високочастотні компоненти та дозволяє запобігти «розгойдуванню» регулятора.

3. Проблеми, аналогічні непрямому інверсному нейрорегулюванню. Зокрема: чутливість до похибок моделювання; ризик виходу роботи регулятора за межі області навчання; складність гарантування глобальної стійкості адаптаційного процесу.

Слід зазначити, що у порівнянні з класичними реалізаціями ІМС нейронні мережі не надають фундаментально нових властивостей стабілізації чи робастності. Основним їхнім внеском є спрощення задачі нелінійної ідентифікації, що дозволяє ефективно застосовувати ІМС до об'єктів зі складною та невизначеною динамікою.

1.2.4 Feedforward-Feedback-Error регулювання.

Концепція Feedforward–Feedback–Error регулювання є подальшим розвитком ідей непрямого інверсного нейрорегулювання. Основна відмінність полягає у розширеній структурі системи та наявності додаткового каналу зворотного зв'язку, у якому працює третя нейронна мережа – Feedback-Error-регулятор. Структурна схема представлена на рис. 1.6.

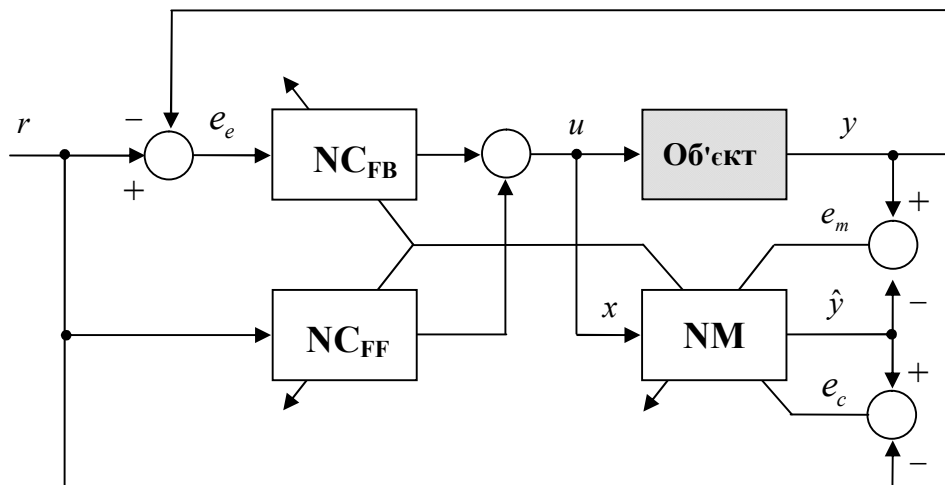


Рисунок 1.6 – Структура Feedforward–Feedback–Error-регулятора

У цій архітектурі використовується три нейронні мережі:

- Feedforward-нейрорегулятор $\mathbf{NC}_{FF}$. Виконує функцію інверсного компенсатора в прямому каналі, як це властиво непрямому інверсному підходу.
- Feedback-Error-регулятор $\mathbf{NC}_{FB}$. створює додатковий замкнений контур, що підвищує робастність та точність регулювання.
- Нейромодель об'єкта $\mathbf{NM}$. використовується для корекції помилки та обчислення градієнтів при налаштуванні регуляторів.

На першому етапі всі три нейронні мережі навчаються offline, причому для кожної задається власний функціонал якості, оскільки їх ролі в системі відмінні. Після завершення ідентифікації моделі об'єкта $\mathbf{NM}$ розпочинається етап регулювання. Помилка e_c , модифікована за допомогою прямої нейромоделі, використовується як сигнал підстроювання ваг обох регуляторів.

Завдяки Feedback-Error-регулятору в системі формується замкнений контур, що:

- підвищує стійкість,
- покращує перехідні характеристики,
- забезпечує компенсацію некоректностей моделі.

Таким чином, загальна структура поєднує переваги інверсного feedforward-регулювання та класичного feedback-регулювання. Приклад застосування цього методу можна знайти в роботах [53, 54].

1.2.5 Регулювання з Feedback-Error-Learning

Структура регулювання з *Feedback-Error-Learning* (FEL), запропонована Кавато і співавт. [55], представлена на рис. 1.7. Ідея цього підходу полягає у поєднанні безпосереднього інверсного регулювання з класичним методом – зазвичай ПД-регулятором.

На відміну від інших методів прямих інверсних нейрорегуляторів, FEL використовує вихід ПД-регулятора як сигнал помилки, що підлягає мінімізації e_c при online-навчанні нейромережі:

Завдяки цьому:

- відпадає потреба у корекції помилки за допомогою прямої моделі об'єкта;
- немає необхідності обчислювати матрицю Якобі;
- нейрорегулятор навчається відслідковувати саме ту дію, яку вважає оптимальною ПІД-регулятор.

У процесі online-тренування нейрорегулятор **НС** поступово переймає функції ПІД-регулятора. У міру наближення до інверсної динаміки об'єкта справджується співвідношення $e_c = 0$, що означає, що основне регулювання здійснюється нейромережею, а ПІД-регулятор:

- забезпечує стабільність у початковій фазі навчання;
- коригує випадкові відхилення та шумові компоненти сигналів;
- виконує роль «страхувального» регулятора.

Таким чином, структура FEL поєднує переваги класичного та інверсного регулювання, зберігаючи стійкість при переході до нейромережевого управління.

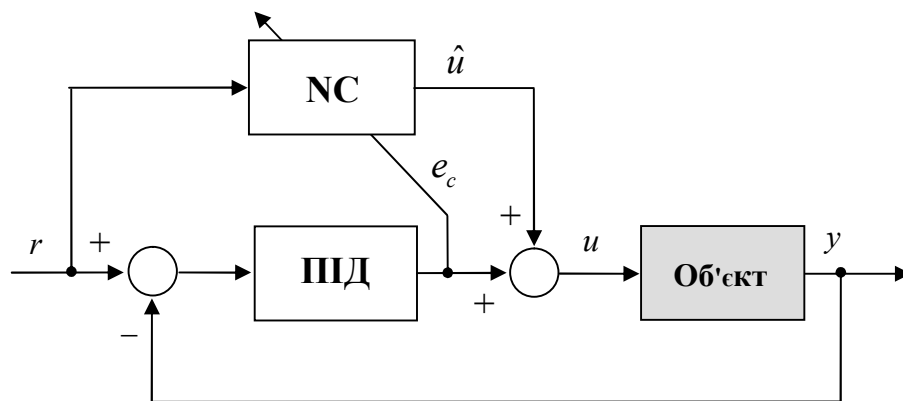


Рисунок 1.7 – Регулювання за методом Feedback-Error-Learning

Переваги та недоліки методу FEL

До переваг належать:

- уникнення необхідності обчислення матриці Якобі;
- природне забезпечення стійкості на ранніх етапах завдяки ПІД-регулятору;

- адаптивність та здатність системи «навчатися від власних помилок».

Основні недоліки:

1. Висока залежність швидкості збіжності від параметрів ПД-регулятора [56]. Надто агресивні чи занадто повільні налаштування можуть сповільнювати або навіть зривати процес навчання.

2. Локальність навчання. Якщо тренувальні дані не охоплюють усю робочу область, нейрорегулятор перестав бути точною інверсною ланкою.

У разі зміни робочого діапазону ваги мережі потребують повторної адаптації.

Попри це, метод FEL залишається одним із найефективніших і практично стабільних підходів у нейромережевому регулюванні для реальних об'єктів із невизначеностями.

1.2.6 Регулювання методом компенсації завдання.

Метод компенсації завдання (*Reference Compensation Technique, RCT*), структурна схема якого наведена на рис. 1.8, був запропонований Юнгом і Хсіа [57]. За своїми принципами RCT є близьким до методу *Feedback-Error Learning* (FEL), однак має принципову відмінність у способі формування компенсуючого сигналу.

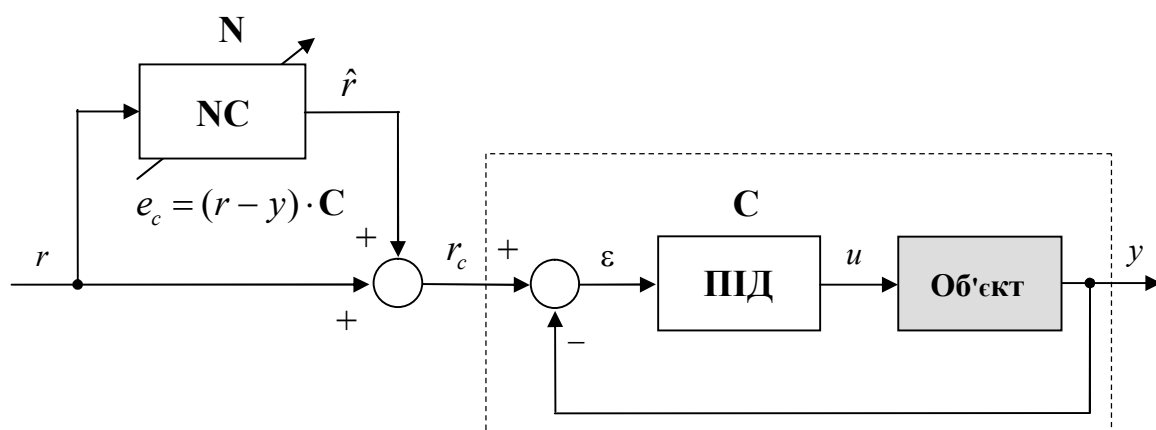


Рисунок 1.8 – Регулювання методом компенсації завдання

На відміну від інших методів прямих нейромережевих регуляторів, у RCT не виконується обчислення матриці Якобі, оскільки для навчання нейромережі

використовується спеціально побудований сигнал помилки e_c (табл. 1.2). Це дозволяє уникнути необхідності аналітичної чи чисельної апроксимації похідних об'єкта.

Ключова особливість техніки RCT полягає в тому, що компенсація нелінійностей здійснюється на рівні сигналів завдання, а не сигналів керування. Таким чином, нейронна мережа модифікує не дію на об'єкт, а саме траєкторію, яку повинен відстежувати класичний регулятор.

Переваги методу:

- можливість простого включення у вже наявний контур керування;
- відсутність необхідності моделі об'єкта для корекції помилки;
- стійкість системи в початковій фазі роботи забезпечується традиційним регулятором.

Складність методу полягає у виборі початкових вагових коефіцієнтів нейромережі. У разі їх невдалого вибору можливе формування небезпечних або некоректних сигналів завдання, що обмежує область практичного застосування.

Таблиця 1.2 – Обчислення градієнта при FEL регулюванні

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

1.2.7 Метод миттєвої лінеаризації (Instantaneous Linearization)

Метод *Instantaneous Linearization* базується на ідеї лінеаризації нелінійної нейромоделі об'єкта в поточній робочій точці [58]. Це дозволяє застосовувати класичні лінійні регулятори, але з урахуванням локальних нелінійних властивостей моделі.

Нехай нелінійний динамічний об'єкт описується рівнянням

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

а його нейронна модель реалізована у вигляді NARX-мережі:

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

де $\mathbf{x}(k) = [y(k-1) \dots y(k-n_y), u(k-n_k) \dots u(k-n_u-n_k+1)]^T$ – вектор стану (попередні входи та виходи),;

$\mathbf{w}$ – вагові коефіцієнти мережі.

Лінеаризація моделі у момент часу $k = \kappa$ здійснюється шляхом розкладання функції у ряд Тейлора першого порядку:

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

де

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

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

Приватні похідні розраховуються безпосередньо зі структури нейромережі за правилом підстановки (аналогічно обчисленню матриці Якобі у методах непрямого інверсного регулювання).

Таким чином формується лінійна модель, локально еквівалентна нелінійному об'єкту, і до неї може бути синтезований будь-який лінійний регулятор (LQR, PID, MRAC тощо). Загальна структура регулювання наведена на рис. 1.9.

Метод Instantaneous Linearization є аналогом класичних методів регулювання з непрямою адаптацією, проте з тією відмінністю, що модель об'єкта формується нейромережею і може описувати складні нелінійності.

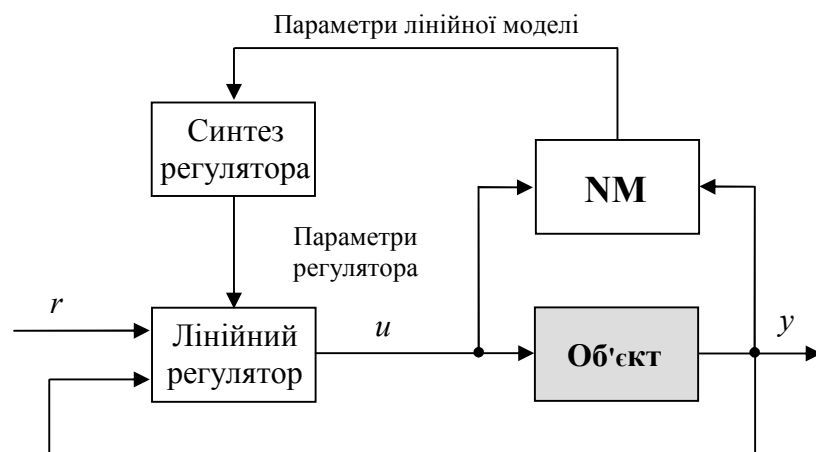


Рисунок 1.9 – Instantaneous linearization і синтез лінійного регулятора

1.2.8 Нелінійне предиктивне регулювання

Метод *Model Predictive Control* (MPC) базується на передбаченні майбутньої поведінки об'єкта за допомогою моделі та формуванні оптимальної послідовності керуючих впливів. На рис. 1.10 пунктирними лініями показано прогнозовану траєкторію вихідної величини на майбутньому горизонті.

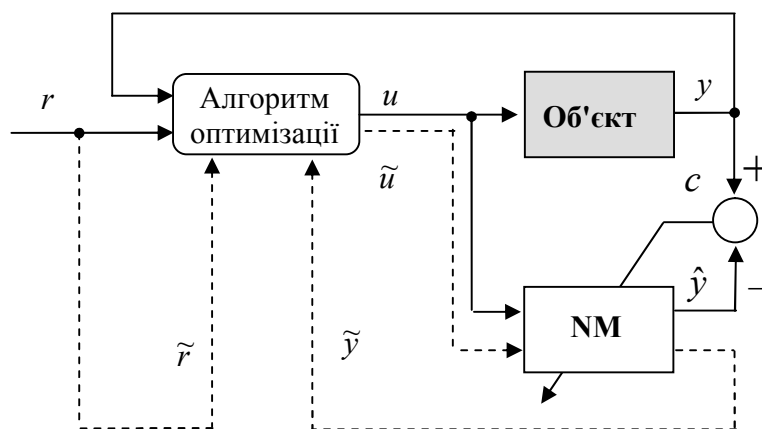


Рисунок 1.10 – Предиктивне регулювання

Суть MPC полягає в наступному:

1. На кожному кроці дискретності формується прогноз виходу системи на певному горизонті часу за допомогою моделі об'єкта.
2. Генерується послідовність дій на об'єкт, яка мінімізує вибраний функціонал якості, що зазвичай містить:

- відхилення від сигналу завдання,
- згладжування керуючих впливів,
- обмеження по входу та виходу.

3. Оптимізація здійснюється чисельними методами (градієнтні методи, SQP, метод Ньютона тощо).

4. На об'єкт подається лише перший елемент оптимізованої послідовності, після чого прогноз і оптимізація виконуються знову.

Переваги MPC:

- висока якість керування навіть за сильно нелінійних об'єктів;
- можливість врахування обмежень;
- природна компенсація запізнень та взаємозв'язків між змінними.

Недоліки:

- велика обчислювальна складність;
- висока чутливість до якості моделі об'єкта;
- необхідність швидких алгоритмів оптимізації для реального часу.

У даній роботі застосовується нелінійне предиктивне регулювання, де роль моделі об'єкта виконує нейронна мережа.

1.3 Основні характеристики мостових кранів

У промисловості України широко використовуються різні типи електричних кранів, що відрізняються конструкцією та призначенням. Так, у металургійних та машинобудівних цехах переважно застосовуються мостові крани, у дворах копалень та на складах вугільних електростанцій – порталні та козлові крани, а на будівельних майданчиках – баштові крани.

Серед усіх типів підйомно-транспортного обладнання найпоширенішим і найбільш універсальним є мостовий кран, який дозволяє ефективно організувати переміщення вантажів над великою площею виробничих або складських приміщень. Металева конструкція моста крана спирається на ходові візки, що рухаються по підкранових шляхах, встановлених на стаціонарних опорах над

робочою зоною. Уздовж моста прокладені рейки, по яких переміщується візок із підйомним механізмом та лебідкою, що здійснює підйом і спуск вантажів.

Таким чином, основними механізмами мостового крана є:

1. Механізм пересування моста – забезпечує переміщення всієї ферми вздовж підкранових шляхів.
2. Механізм пересування візка – переміщує вантажний візок уздовж мосту.
3. Механізм підйому лебідки – піднімає та опускає вантаж.

Кожен механізм оснащений окремим електродвигуном, що дозволяє незалежно керувати рухами крана та забезпечувати точне позиціонування вантажу.

Електроустаткування мостових кранів

Електричне обладнання мостових кранів повинно забезпечувати:

- надійну роботу у повторно-короткочасному режимі та при високій частоті включень;
- стабільну експлуатацію за змінної вологості та різких перепадів температури;
- безперебійну роботу, високу продуктивність і безпеку для обслуговуючого персоналу;
- простоту в експлуатації та технічному обслуговуванні.

Номінальна вантажопідйомність мостових кранів регламентується стандартами і варіюється в діапазоні від 1 до 1000 *t*. Середня маса переміщуваних вантажів зазвичай складає 40–60 % номінальної вантажопідйомності. Для мостових кранів з лебідковими візками $G_{\text{вп}} = 5 \div 100 \text{ t}$

Режими роботи та швидкості

Режими роботи вантажопідйомних кранів визначені стандартами та поділяються на вісім режимних груп залежно від призначення і частоти експлуатації механізмів. Для мостових кранів з лебідковими візками, що працюють у цехах та на складах, встановлені відповідні групи і режими роботи, які регламентують:

- номінальні швидкості руху механізмів: механізм підйому $V_{\text{п}} = 0,1 \div 1,2 \text{ м / с}$; пересування моста $V_{\text{м}} = 0,8 \div 1,5 \text{ м / с}$; пересування візка $V_{\text{в}} = 0,5 \div 0,7 \text{ м / с}$;

- електроустаткування повинно витримувати напругу 220/380 В змінного струму;

- обмеження ходу механізмів: підйому – вгору; візка – в обидві сторони.

Для захисту від короткого замикання та перевантаження понад 200 % номіналу застосовуються струмові реле. Крім того, у схемі управління крана встановлюється нульовий захист, який запобігає самозапуску електродвигунів після відновлення електроживлення.

Гальмівні та захисні пристрої. Всі механізми мостового крана обладнані гальмами. Підйомні механізми обов'язково оснащені автоматичними гальмами закритого типу, які спрацьовують при відключенні електроживлення.

Для кранів, що працюють на відкритому повітрі, передбачені протиугінні пристрої. На потужних мостових кранах встановлюється апаратура, яка запобігає виходу ферми моста за допустимі межі, що підвищує безпеку та надійність роботи.

1.4 Обґрунтування вибору системи електроприводу механізму переміщення моста мостового крана

Системи управління механізмами мостових кранів відносяться до категорії пристроїв, що працюють під постійним контролем оператора. Вони забезпечують точне керування рухами крана у всіх напрямках: початок операції, зміна швидкості, розгін та гальмування, а також завершення переміщення вантажу. Основним завданням системи управління є забезпечення безпечної та надійної роботи крана, попередження перевантажень, контроль швидкісних параметрів та автоматичне перемикання режимів роботи електроприводів відповідно до заданих алгоритмів.

Класифікація систем управління кранами за способом керування

На сьогоднішній день системи управління мостовими кранами класифікуються за способом керування на такі групи:

1. Пряме керування з силовими кулачковими регуляторами. У даному випадку всі етапи керування – від прискорення до гальмування – виконуються виключно оператором. Ця система характеризується високою надійністю, але обмеженою точністю при виконанні складних траєкторій.

2. Системи з кнопковими постами. Управління здійснюється за допомогою набору кнопок, що відповідають певним швидкісним режимам. Функціональні можливості таких систем обмежуються конструктивними особливостями поста та заздалегідь заданою програмою запуску та гальмування.

3. Системи з комплексними пристроями керування. Використовуються магнітні або силові регулятори з енергетичними перетворювачами. В цьому випадку оператор задає лише бажану швидкість або положення вантажу, а розгін, уповільнення та проміжні перемикання виконуються автоматично. Такі системи забезпечують високу точність позиціонування, плавність руху та безпеку експлуатації.

Типи регулювання електроприводів.

В залежності від умов експлуатації виділяють такі типи систем регулювання:

1. Регулювання швидкості нижче номінальної – використовується для виконання точних операцій, коли швидкість руху повинна бути меншою за максимальну.

2. Регулювання швидкості вище і нижче номінальної – дозволяє реалізувати як швидкі переміщення, так і точне позиціонування вантажу.

3. Регулювання прискорення та уповільнення – забезпечує плавний розгін і гальмування, що зменшує динамічні навантаження на механізм крана та вантаж.

Види електроприводів, що використовуються в мостових кранах

На основі типу двигуна та способу регулювання виділяють такі системи електроприводу:

- Постійного струму:
 - К-ДП – керування через силовий регулятор;

- МК-ДП – керування через магнітний регулятор;
- ТП-Д – тиристорний перетворювач–двигун;
- Г-Д – схема генератор–двигун.
- Змінного струму (асинхронні двигуни):
 - МП-АДК, К-АДК – з короткозамкненим ротором, керовані магнітним або силовим регулятором;
 - МК-АДД – двошвидкісний короткозамкнений двигун з магнітним регулятором;
 - К-АДФ, КД-АДФ, КІ-АДФ – двигуни з фазним ротором з різними типами регуляторів та динамічного гальмування;
 - МКП-АДФ, МКД-АДФ, МКБ-АДФ, МКІ-АДФ – магнітне регулювання з різними способами гальмування та комутації;
 - НПЧ-АДД – двошвидкісний асинхронний двигун з частотним перетворювачем.

Кожен тип електроприводу має свої переваги та недоліки щодо точності позиціонування, плавності ходу, енергоспоживання та вартості системи.

Сучасні дослідження також показують перспективність використання регульованих частотних перетворювачів змінного струму для асинхронних двигунів, що дозволяє зменшити масу обладнання, спростити обслуговування і забезпечити більш високий ресурс роботи механізмів.

Обґрунтування вибору системи електроприводу для механізму переміщення моста

Вибір системи електроприводу здійснюється з урахуванням:

- експлуатаційних умов (середовище, температурні перепади, вологість, запиленість);
- режимів роботи (частота включень, середні навантаження, пікові перевантаження);
- вимог до плавності та точності переміщень, що критично для мостових кранів з великими вантажопідйомностями;

- потреб у захисті від перевантажень, коротких замикань та небезпечних режимів роботи.

На основі аналізу сучасних методів управління та технічних характеристик кранових механізмів, оптимальним є вибір електроприводу постійного струму з тиристорним перетворювачем (ТП-Д).

Переваги цього вибору:

- можливість регулювання швидкості як вище, так і нижче номінальної;
- плавне прискорення та уповільнення без ривків, що зменшує динамічні навантаження на механіку крана;
- високий ККД системи та економія енергії;
- можливість інтеграції з сучасними автоматизованими системами керування;
- забезпечення стійкої роботи при змінних навантаженнях та швидких циклах включення/вимкнення.

1.5 Постановка задачі синтезу нейромережевої системи управління

В результаті проведеного ґрунтовного аналізу наукової літератури можна зробити висновок, що на сьогодні існує велика кількість різноманітних підходів до синтезу систем автоматичного управління. Проте, незважаючи на суттєвий прогрес у галузі теорії керування, універсального та абсолютно ідеального регулятора, здатного забезпечити оптимальну роботу в усіх можливих умовах, досі не створено. Це особливо актуально при побудові систем керування електроприводами складних об'єктів, до яких належить і електропривод механізму розвороту або переміщення робочих органів вантажопідіймальних машин та екскаваторів. У таких системах наявні пружно пов'язані інерційні маси, зокрема маса стріли, ротора з робочим колесом, а також маса консолі противаги, які жорстко або пружно взаємодіють із платформою. Саме ця особливість призводить до виникнення значних коливальних процесів під час перехідних режимів роботи. У результаті традиційні лінійні регулятори, незалежно від їх типу, час-

то не здатні забезпечити потрібний рівень якості керування, оскільки не враховують складну динаміку багатомасових механічних систем.

Теорія адаптивного та оптимального управління відкриває можливості побудови систем, які працюють відповідно до визначених критеріїв оптимальності, мінімізуючи помилки або максимізуючи певні функціональні показники. Сучасні підходи до керування охоплюють не лише методи синтезу регуляторів, але й широкий спектр методів ідентифікації динамічних процесів, що дає змогу отримувати моделі об'єкта управління, навіть коли їхня структура або параметри не є відомими заздалегідь. Процедури параметричної та структурної ідентифікації забезпечують формування адекватної моделі об'єкта як у реальному часі, так і під час попередніх експериментальних досліджень. Такі алгоритми лежать в основі як класичних систем керування, так і адаптивних та інтелектуальних контролерів.

Разом із тим, фундаментом класичної теорії керування залишається теорія лінійних систем. Хоча її інструменти широко застосовуються також для нелінійних об'єктів, найбільш ефективні результати досягаються лише в тих випадках, коли нелінійність незначна або об'єкт має великі сталі часу та достатню стійкість у розімкненому режимі. У задачах керування складними динамічними об'єктами додаткові труднощі спричиняє необхідність постійного відстеження змін параметрів моделі, а інколи й переоцінки самої її структури. Для апроксимації нелінійних залежностей використовуються моделі у вигляді рядів Вольтерра, Гаммерштейна, Вінера, поліномів Колмогорова–Габора та інших функціональних представлень. Проте застосування таких моделей обмежується тим, що попередньо необхідно виконати структурну ідентифікацію – визначити порядок моделі та відповідні параметри, що є складним завданням, особливо за наявності зовнішніх збурень і шумів вимірювання. Це ускладнює побудову систем, що мають працювати в режимі реального часу, та ставить вимогу створення високошвидкісних алгоритмів керування.

З огляду на велику різноманітність типів нелінійностей, складність динаміки об'єктів, їх нестаціонарність та вплив зовнішніх випадкових факторів, досі

неможливо розробити універсальний підхід до побудови систем управління, придатний для будь-якого об'єкта. Більше того, навіть сучасні теорії часто базуються на ідеї лінеаризації, яка у практичних умовах не завжди забезпечує необхідну точність, оскільки модель, побудована на припущенні лінійності, може значно відхилятися від реальної поведінки об'єкта.

Альтернативою класичним підходам, що демонструє високу ефективність при роботі з істотно нелінійними та складними динамічними об'єктами, є штучні нейронні мережі. Нейронні мережі – це інтелектуальні системи, у яких обробка інформації здійснюється подібно до роботи біологічних нейронних структур. Їхньою ключовою перевагою є здатність до паралельної обробки інформації, що забезпечує високу швидкодію та можливість роботи в режимі реального часу. Завдяки великій кількості міжнейронних зв'язків нейронна мережа має високу відмовостійкість: помилки на окремих лініях або вузлах не призводять до критичних збоїв, оскільки мережа здатна компенсувати пошкоджені зв'язки шляхом перерозподілу інформаційних потоків.

Ще однією важливою особливістю нейронних мереж є здатність до навчання та узагальнення накопичених знань. Навчившись на обмеженому наборі даних, мережа здатна коректно реагувати на нові, раніше не зустрічані вхідні сигнали. Завдяки нелінійним функціям активації нейронні мережі здатні відтворювати складні нелінійні залежності між входами та виходами системи, що робить їх перспективними для задач керування динамічними і нелінійними об'єктами, для яких традиційні методи часто не дають прийнятних результатів. Сучасні системи керування широко використовують такі типи мереж як багатшарові персептрони, радіально-базисні та нейро-фаззі структури.

Разом з тим, ефективність нейронних мереж визначається правильним вибором структури моделі – кількості шарів, числа нейронів, типів функцій активації – а також достатнім обсягом навчальних даних. Для побудови якісної нейромоделі кількість навчальних прикладів повинна значно перевищувати кількість вагових коефіцієнтів. Управління складними динамічними об'єктами за допомогою нейронних мереж базується на двох основних етапах: ідентифікації

об'єкта (побудові нейромоделі) та синтезі нейрорегулятора, який використовує прогнозовану поведінку системи для формування керуючого впливу.

Метою цієї роботи є розробка нейромережевої системи управління механізмом переміщення візка мостового крана з урахуванням поперечних розгойдувань вантажу, що забезпечує підвищену точність, стійкість та якість регулювання в умовах пружних зв'язків механічної частини.

Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Виконати вибір системи електроприводу механізму переміщення візка мостового крана, вибір електродвигуна та силового електроустаткування. Провести синтез системи підлеглого керування електроприводом і виконати її моделювання на ЕОМ.

2. Розробити математичну модель системи управління з урахуванням поперечних розгойдувань вантажу, провести комп'ютерне моделювання багатомасової системи та виконати аналіз перехідних процесів.

3. Розробити структурну схему нейромережевої системи управління, здатної забезпечити високоякісне регулювання з урахуванням пружності механічних зв'язків.

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

5. Використовуючи принцип предиктивного управління, синтезувати нейромережевий регулятор на основі отриманої моделі об'єкта, який забезпечує високу швидкодію та покращені показники якості регулювання.

6. Провести моделювання нейромережевої системи управління на ЕОМ і виконати детальний аналіз отриманих результатів.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Проведений аналіз методів застосування штучних нейронних мереж для управління електромеханічними системами показав, що нейромережеві моделі здатні формувати внутрішнє уявлення про стан об'єкта управління, прогнозувати його поведінку та генерувати оптимальні управляючі дії відповідно до еталонної моделі. Особлива увага приділяється методам навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів та синтезу адаптивних систем управління.

2. Здатність штучних нейронних мереж до самонавчання дозволяє ефективно використовувати нейрорегулятори навіть за умов значних невизначеностей та зовнішніх збурень, тоді як класичні методи адаптивного управління потребують великого обсягу попередньої інформації про об'єкт. Простота реалізації нейронних мереж та їх здатність до навчання роблять їх привабливими для керування складними нелінійними об'єктами в реальному часі.

3. Дослідження показали ефективність використання багат шарового персептрону як універсального апроксиматора для створення нейромережевої моделі об'єкта управління. Багат шаровий персептрон забезпечує точне моделювання нелінійних залежностей та використовується як для ідентифікації об'єкта, так і для синтезу регулятора.

4. Адекватне моделювання динаміки системи є необхідною умовою прогнозування поведінки та синтезу ефективного регулятора. Завдання ідентифікації нелінійних динамічних об'єктів успішно вирішуються за допомогою багат шарового персептрону та радіально-базисних мереж, що дозволяє відтворювати як короткочасні, так і коливальні ефекти, пов'язані з пружними зв'язками механізму.

5. Для оптимізації роботи системи управління необхідно адаптивно підстроювати параметри регуляторів залежно від умов роботи. Інтеграція методів адаптивного управління з нейронними мережами дозволяє створити ефективні, прості у реалізації нейромережеві моделі та регулятори, здатні підтримувати

стабільність і якість управління навіть при зміні навантажень та зовнішніх умов.

6. Аналіз вимог до електроприводів механізмів мостових кранів показав, що ключовими критеріями є простота експлуатації, надійність та можливість регулювання швидкості, з урахуванням обмежень по прискоренню, струму та моменту. Робочі швидкості механізмів дозволяють забезпечити короткі перехідні процеси у порівнянні з повним робочим циклом. Водночас необхідно враховувати динамічні фактори, такі як розгойдування вантажу та кінематичні зазори, і застосовувати заходи для обмеження небажаних навантажень.

7. Сформульовано завдання дослідження: розробка нейромережевої системи управління багатомасовою електромеханічною системою механізму переміщення візка мостового крана, яка забезпечує високу точність позиціонування, ефективне гасіння коливань вантажу та високу швидкодію в умовах нелінійної динаміки.

8. На основі аналізу сучасних методів нейрорегулювання та вимог до електроприводів обрано принцип нелінійного предиктивного регулювання як основу для побудови нейромережевого регулятора, що забезпечує передбачувану поведінку системи та ефективне управління у реальному часі.

9. Загальний висновок: використання штучних нейронних мереж у системах управління мостовими кранами дозволяє поєднати адаптивність, точність і здатність до прогнозування, що є критично важливим для управління складними електромеханічними об'єктами з багатомасовою структурою та пружними зв'язками.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА МОСТОВОГО КРАНА

2.1 Технічні характеристики мостового крана та вимоги до системи управління

Для розрахунку електроприводу механізму переміщення візка мостового крана та синтезу системи автоматичного управління використані вихідні дані, наведені у таблиці 2.1.

Таблиця 2.1 – Основні технічні характеристики мостового крана

Параметри	Величина
Вантажопідйомність $G_{\text{вп}}$, кг	37000
Продуктивність крана Q , кг/с	55
Маса моста(без візка) $m_{\text{м}}$, кг	28000
Маса візка $m_{\text{т}}$, кг	8000
Діаметр коліс моста $D_{\text{м}}$, м	0,71
Діаметр коліс візка $D_{\text{т}}$, м	0,4
Діаметр цапф коліс моста $d_{\text{м}}$, м	0,145
Діаметр цапф коліс візка $d_{\text{т}}$, м	0,095
Швидкість пересування моста $V_{\text{м}}$, м/с	1,25
Швидкість пересування візка $V_{\text{т}}$, м/с	0,65
Швидкість підйому $V_{\text{п}}$, м/с	0,1
Маса вантажозахватного пристрою m_0 , кг	1000
Середня висота підйому і спуску $L_{\text{п}}$, м	6
Середній шлях моста в одну сторону $L_{\text{м}}$, м	45
Середній шлях візка в одну сторону $L_{\text{в}}$, м	24
Діаметр барабана механізму підйому D , м	0,65

Вимоги до показників якості роботи системи управління.

Система управління повинна бути статичною. Перерегулювання швидкості при ступінчастій вхідній дії не повинно перевищувати 10 %. Час регулювання системи не має перевищувати 5 секунд.

2.2 Розрахунок потужності електроприводу і вибір електродвигуна

1. Статична потужність на валу двигуна механізму пересування візка

Статична потужність визначається за формулою:

$$P_{\text{ст в}} = \frac{k_p \cdot q \cdot (G_{\text{вп}} + m_{\text{в}} + m_0) \cdot V_{\text{в}}}{\eta_{\text{н в}} \cdot R_{\text{в}}} \cdot \left(\mu \cdot \frac{d_{\text{в}}}{2} + f \right),$$

де q – прискорення вільного падіння;

$k_p \approx 1,35$ – коефіцієнт тертя реборд коліс об рейки;

$R_{\text{в}} = 0,2 \text{ м}$ – радіус коліс візка;

$\mu = 0,015$ – коефіцієнт тертя в підшипниках;

$f = 0,5 \cdot 10^{-3} \text{ м}$ – коефіцієнт тертя кочення коліс об рейки;

$\eta_{\text{н}} = 0,88$ – номінальний ККД завантаженого механізму.

Після підстановки числових значень отримуємо:

$$P_{\text{ст в}} = \frac{1,35 \cdot 9,81 \cdot (37 + 8 + 1) \cdot 10^3 \cdot 0,65}{0,88 \cdot 0,2} \cdot \left(0,015 \cdot \frac{0,095}{2} + 0,5 \cdot 10^{-3} \right) \cdot 10^{-3} = 2,5 \text{ кВт}.$$

2. Розрахунок циклів роботи і часу включення

Число циклів пересування в секунду:

$$Z = \frac{Q}{G_{\text{вп}}},$$

$$Z = \frac{55}{37000} = 1,486 \cdot 10^{-3},$$

Час циклу пересування візка:

$$T = \frac{1}{Z},$$

$$T = \frac{1}{1,486 \cdot 10^{-3}} = 670 \text{ с}.$$

Усереднений час пересування візка

$$t_B = \frac{2L_B}{V_B},$$

$$t_B = \frac{2 \cdot 24}{0,65} = 73,8 \text{ с}.$$

Тривалість включення електроприводу:

$$ПВ_B = \frac{t_B}{T} \cdot 100,$$

$$ПВ_B = \frac{73,8}{670} \cdot 100 = 11\%.$$

3. Розрахункова потужність електродвигуна

Розрахункова потужність враховує пускові умови та додаткові динамічні навантаження:

$$P_{pB} = \frac{0,66 \cdot (G_{вп} + m_B + m_0) \cdot a_{мех} \cdot V_B}{\eta_H} + \frac{P_{стB}}{1,75},$$

де $a_{мех} = 0,65 \text{ м/с}^2$ – допустиме прискорення механізму пересування візка

$$P_{pB} = \frac{0,66 \cdot (36000 + 8000 + 1000) \cdot 0,65 \cdot 0,65}{0,88} \cdot 10^{-3} + \frac{2,5}{1,75} = 15,7 \text{ Вт}.$$

Потужність електродвигуна приводимо до стандартної тривалості включення $ПВ_{ст} = 40\%$

$$P_{\epsilon} = P_{pB} \cdot \sqrt{\frac{ПВ_B}{ПВ_{ст}}},$$

$$P_T = 15,7 \cdot \sqrt{\frac{11}{40}} = 8,2 \text{ кВт}.$$

4. Вибір електродвигуна.

На основі розрахованої потужності вибираємо електродвигун типу Д- 32 з технічними характеристиками : $P_{\text{н}} = 9,5 \text{ кВт}$; $U_{\text{н}} = 220 \text{ В}$; $I_{\text{н}} = 57 \text{ А}$; $n_{\text{н}} = 800 \text{ об/хв}$; $R_{\text{я}} + R_{\text{дп}} = 0,28 \text{ Ом}$; $2a = 2$; $2p = 4$; $\Phi_{\text{н}} = 0,0132 \text{ Вб}$; ПВ=40%; $J_{\text{дв}} = 0,425 \text{ кг} \cdot \text{м}^2$; $N = 538..$

5. Пояснення вибору.

- Вибір двигуна здійснюється на основі забезпечення необхідної потужності для стабільного розгону та гальмування візка, із врахуванням інерційних мас та можливих коливань вантажу.
- Стандартизація двигуна спрощує експлуатацію, ремонт і запасні частини.
- Врахування тривалості включення (ПВ) забезпечує термічну стійкість двигуна і надійну роботу в циклічному режимі.

2.3 Розрахунок швидкісної діаграми електроприводу

1. Передавальне число механізму

Передавальне число механізму визначає зв'язок між кутовою швидкістю електродвигуна та лінійною швидкістю пересування візка:

$$i_{\text{в}} = \frac{\omega_{\text{н}} \cdot D_{\text{в}}}{2V_{\text{в}}},$$

де $\omega_{\text{н}}$ – номінальна кутова швидкість обертання електродвигуна

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

$$\omega_{\text{н}} = \frac{3,14 \cdot 800}{30} = 83,7 \text{ с}^{-1}.$$

$$i_{\text{в}} = \frac{83,7 \cdot 0,4}{2 \cdot 0,65} = 25,8$$

2. Статичні моменти на механізмі пересування візка

2.1 Пересування візка з вантажем

Статичний момент для переміщення візка з вантажем визначається за формулою:

$$M_{\text{ст в}} = \frac{k_p \cdot q \cdot (G_{\text{вп}} + m_{\text{в}} + m_0)}{\eta_{\text{н в}} \cdot i_{\text{в}}} \cdot \left(\mu \cdot \frac{d_{\text{в}}}{2} + f \right),$$

Після підстановки чисельних значень отримуємо:

$$M_{\text{ст в}} = \frac{1,35 \cdot 9,81 \cdot (36000 + 8000 + 1000)}{0,88 \cdot 25,8} \cdot \left(0,015 \cdot \frac{0,095}{2} + 0,5 \cdot 10^{-3} \right) = 31,12 \text{ Нм}.$$

2.2 Пересування візка без вантажу

Для порожнього візка статичний момент зменшується:

$$M_{\text{ст в 0}} = \frac{k_p \cdot q \cdot (m_{\text{в}} + m_0)}{\eta_{\text{в 0}} \cdot i_{\text{в}}} \cdot \left(\mu \cdot \frac{d_{\text{в}}}{2} + f \right),$$

де $\eta_{\text{в 0}}$ – ККД візка при переміщенні без вантажу, визначається залежно від співвідношення:

$$\gamma = \frac{m_{\text{в}}}{G_{\text{вп}} + m_{\text{в}}},$$

$$\gamma = \frac{8000}{36000 + 8000} = 0,18.$$

Приймаємо $\eta_{\text{в 0}} = 0,68$.

$$M_{\text{ст в 0}} = \frac{1,35 \cdot 9,81 \cdot (8000 + 1000)}{0,68 \cdot 25,8} \cdot \left(0,015 \cdot \frac{0,095}{2} + 0,5 \cdot 10^{-3} \right) = 7,2 \text{ Нм}.$$

3. Час розгону і гальмування електроприводу

Час пуску та гальмування визначається з урахуванням допустимих прискорень $a_{\text{в}} \leq 0,7 \text{ м/с}^2$.

При невеликому перепаді швидкостей на природній характеристиці:

$$t_{\text{пв}} = t_{\text{гв}} = \frac{V_{\text{в max}}}{a_{\text{в}}},$$

$$t_{\text{пв}} = t_{\text{гв}} = \frac{0,65}{0,7} = 0,93 \text{ с}.$$

4. Шлях, пройдений візком під час розгону та гальмування:

$$l_{\text{пв}} = l_{\text{гв}} = \frac{V_{\text{в}}}{2} \cdot t_{\text{пв}},$$

$$l_{\text{пв}} = l_{\text{гв}} = \frac{0,65}{2} \cdot 0,93 = 0,3 \text{ м}.$$

5. Час роботи механізму з сталою швидкістю

Час переміщення візка на сталому режимі (коли досягнута номінальна швидкість) визначається за формулою:

$$t_{\text{ст в 1}} = t_{\text{ст в 2}} = \frac{L_{\text{в}} - (l_{\text{пв}} + l_{\text{гв}})}{V_{\text{в}}},$$

$$t_{\text{ст в 1}} = t_{\text{ст в 2}} = \frac{24 - (0,3 + 0,3)}{0,65} = 33,7 \text{ с}.$$

Ці параметри дозволяють побудувати швидкісну діаграму електроприводу, де чітко виділяються фази розгону, руху з номінальною швидкістю та гальмування.

2.4 Розрахунок діаграми навантаження електроприводу

Для побудови діаграми навантаження електроприводу необхідно визначити значення моментів на всіх ділянках швидкісної діаграми механізму пересування візка.

1. Розгін завантаженого візка

$$M_{рв} = M_{ств} + M_{дин},$$

де $M_{дин}$ – динамічний момент електроприводу

$$M_{дин} = J_{пр} \cdot \frac{\omega_n}{t_{пв}},$$

$J_{пр}$ – момент інерції приводу, приведений до валу двигуна

$$J_{пр} = \delta \cdot J_{дв} + \frac{(G_{вп} + m_{в} + m_0) \cdot V_{в}^2}{\omega_n^2},$$

де $\delta = 1,3$ – коефіцієнт, що враховує момент інерції передавального механізму.

$$J_{пр} = 1,3 \cdot 0,425 + \frac{(37000 + 8000 + 1000) \cdot 0,65^2}{83,4^2} = 0,5525 + 1,71 = 2,27 \text{ кг} \cdot \text{м}^2,$$

$$M_{дин} = 2,27 \cdot \frac{83,7}{0,93} = 203 \text{ Нм},$$

$$M_{рв} = 31,12 + 203 = 233,12 \text{ Нм}.$$

2. Рух завантаженого візка із сталою швидкістю.

На ділянці руху з постійною швидкістю:

$$M_{ствл} = M_{ств} = 31,12 \text{ Нм}.$$

3. Гальмування завантаженого візка

При гальмуванні динамічний момент змінює знак:

$$M_{гв} = M_{ств} - M_{дин},$$

$$M_{\Gamma B} = 31,12 - 203 = 171,88 \text{ Нм}.$$

4. Розгін порожнього візка

Аналогічно розгону завантаженого візка:

$$-M_{pB0} = M_{стB0} + M_{дин0},$$

де $M_{дин0}$ – динамічний момент електроприводу при розгоні порожнього візка

$$M_{дин0} = J_{пр0} \cdot \frac{\omega_H}{t_{пв}}$$

$J_{пр0}$ – момент інерції приводу для розгону і гальмування для порожнього візка, приведений до валу двигуна

$$J_{пр0} = \delta \cdot J_{дв} + \frac{(m_B + m_0) \cdot V_B^2}{\omega_H^2},$$

$$J_{пр0} = 1,2 \cdot 0,425 + \frac{(8000 + 1000) \cdot 0,65^2}{83,7^2} = 0,5525 + 0,34 = 0,89 \text{ кг} \cdot \text{м}^2,$$

$$M_{дин0} = 0,89 \cdot \frac{83,7}{0,93} = 78 \text{ Нм},$$

$$-M_{pB0} = 7,2 + 78 = 85 \text{ Нм}.$$

5. Рух порожнього візка із сталою швидкістю

$$-M_{стB2} = M_{стB0} = 7,2 \text{ Нм}.$$

6. Гальмування порожнього візка

$$-M_{\Gamma B0} = M_{стB0} - M_{дин0},$$

$$-M_{\Gamma B0} = 7,2 - 85 = -77,8 \text{ Нм}.$$

7. Уточнення тривалості включення механізму

Тривалість включення (ПВ) механізму визначається як:

$$ПВ_{\text{в}} = \frac{2 \cdot t_{\text{пв}} + 2 \cdot t_{\text{гв}} + t_{\text{ств1}} + t_{\text{ств2}}}{T},$$

$$ПВ_{\text{в}} = \frac{2 \cdot 0,93 + 2 \cdot 0,93 + 2 \cdot 33,7}{460} = 0,149.$$

2.5 Перевірка правильності попереднього вибору електродвигуна

Для оцінки правильності вибору потужності електродвигуна використовується метод еквівалентного моменту:

$$M_{\text{екв в}} = \sqrt{\frac{\sum M_{\text{п}_i}^2 t_{\text{п}_i} + \sum M_{\text{г}_i}^2 t_{\text{г}_i} + \sum M_{\text{ст}_i}^2 t_{\text{ст}_i}}{\beta (\sum t_{\text{п}_i} + \sum t_{\text{г}_i}) + \sum t_{\text{ст}_i}}} \quad (2.1)$$

де $M_{\text{п}_i}$, $M_{\text{г}_i}$, $M_{\text{ст}_i}$ – середнє значення моменту на i тій ділянці циклу;

$t_{\text{п}_i}$, $t_{\text{г}_i}$, $t_{\text{ст}_i}$ – час роботи механізму на кожній ділянці;

$$\beta = \frac{1 + \beta_o}{2} = \frac{1 + 0,35}{2} = 0,675 \text{ – коефіцієнт, що враховує погіршення охоло-$$

дження під час пауз.

Умова відповідності двигуна за потужністю:

$$M_{\text{н}} \geq M_{\text{екв пр}}. \quad (2.2)$$

де

$$M_{\text{екв пр}} = \frac{M_{\text{екв в}}}{2} \cdot \sqrt{\frac{ПВ_{\text{в}}}{ПВ_{\text{ст}}}}.$$

Після підстановки моментів і тривалостей роботи на різних ділянках в (2.1) отримуємо $M_{\text{екв м}} = 67,2 \text{ Нм}$

$$M_{\text{екв пр}} = 67,2 \cdot \sqrt{\frac{0,149}{0,4}} = 41,15 \text{ Нм}.$$

Номінальний момент обраного двигуна:

$$M_{\text{н}} = \frac{P_{\text{н}}}{\omega_{\text{н}}},$$

$$M_{\text{н}} = \frac{9500}{83,7} = 113,5 \text{ Нм},$$

Висновок: умова (2.2) виконується, тобто двигун відповідає вимогам по нагріву.

2. Перевірка за перевантажувальною здатністю

Максимальний момент при пуску приводу складає $M_{\text{max}} = 233,12 \text{ Нм}$.

Перевантажувальна здатність двигунів постійного струму $\lambda = 2 \div 2,5$.

Допустимий момент двигуна становить:

$$M_{\text{дв max}} = 2,5 \cdot 113,5 = 283,75 \text{ Нм}.$$

Оскільки $M_{\text{дв max}} > M_{\text{max}}$, то двигун задовольняє вимогам по перевантажувальній здатності.

Таким чином, діаграма навантаження електроприводу враховує: розгін і гальмування завантаженого та порожнього візка; рух з постійною швидкістю і дозволяє виконати перевірку двигуна за нагрівом і перевантаженням.

2.6 Вибір тиристорного перетворювача і розрахунок параметрів силового кола електроприводу

Для живлення електроприводів вантажопідіймальних машин, зокрема кранів, традиційно застосовують тиристорні перетворювачі типів АТРК та ТПЕ. Хоча перетворювачі серії ТПЕ вирізняються кращими масо-габаритними харак-

теристиками та підвищеною енергоефективністю, їхня номінальна випрямлена напруга становить 460 В. Це робить їх менш доцільними для даного об'єкта, оскільки у вибраному для механізму переміщення візка електродвигуні передбачено номінальну напругу 220 В, і потреби у додатковому запасі по напрузі немає.

З огляду на це, за каталогом обрано тиристорний перетворювач серії АТРК, модель АТРК-100/230, технічні параметри якого повністю відповідають вимогам до роботи електроприводу: $U_{дн} = 230 \text{ В}$; $I_{дн} = 100 \text{ А}$. Даний перетворювач забезпечує надійну роботу в режимах, характерних для механізмів пересування, та має достатній запас по струму для компенсації можливих переважань під час пуску та гальмування.

Живлення перетворювача здійснюється від силового трансформатора типу ТЗЗР-40/0,5. Основні технічні характеристики трансформатора такі: $U_1 = 380 \text{ В}$; $U_2 = 205 \text{ В}$; $I_1 = 48,6 \text{ А}$; $I_2 = 92 \text{ А}$; $U_{кз} = 0,08$. Застосування трансформатора ТЗЗР-40/0,5 забезпечує необхідний рівень електричної ізоляції, компенсацію можливих коливань напруги в мережі та стабільність живлення тиристорного перетворювача.

1. Сумарна індуктивність силового ланцюга електроприводу.

Сумарна індуктивність визначається як:

$$L_{\Sigma} = L_{\text{як}} + 2L_{\text{тр}} ,$$

де $L_{\text{як}}$ – індуктивність якірного ланцюга електродвигуна;

$L_{\text{тр}}$ – приведена до випрямленого струму індуктивність трансформатора.

1.1 Індуктивність якірного ланцюга

$$L_{\text{як}} = k_1 \frac{U_{\text{н}}}{I_{\text{н}} \omega_{\text{н}} p} ,$$

де k_1 – постійна, що враховує тип двигуна: 0,1–0,25 для компенсованих, ~0,6 для некомпенсованих.

p – число пар полюсів.

$$L_{\text{як}} = 0,6 \cdot \frac{220}{51 \cdot 83,7 \cdot 2} = 0,0157 \text{ Гн} .$$

1.2 Приведена індуктивність трансформатора

Для трифазного мостового випрямляча:

$$L_{\text{тр}} = U_{\text{кз}} \cdot \frac{k_{\text{в}} \cdot U_2}{2\pi \cdot m \cdot f \cdot I_{\text{н}}} ,$$

де $k_{\text{в}}$ – коефіцієнт випрямлення (для трифазної мостової схеми $k_{\text{в}} = 1,35$);

f – частота мережі,

m – еквівалентне число фаз (для мостового випрямляча $m = 6$)

$$L_{\text{тр}} = 0,055 \cdot \frac{1,35 \cdot 205}{2\pi \cdot 6 \cdot 50 \cdot 51} = 0,159 \cdot 10^{-3} \text{ Гн} ,$$

$$L_{\Sigma} = 15,7 \cdot 10^{-3} + 2 \cdot 0,159 \cdot 10^{-3} = 15,84 \cdot 10^{-3} \text{ Гн} .$$

2. Сумарний активний опір силового ланцюга

$$R_{\Sigma} = R_{\text{як}} + R_{\text{епр}} + 2R_{\text{тр}} ,$$

де $R_{\text{як}}$ – опір якірного ланцюга, враховуючи нагрів

$$R_{\text{як}} = \alpha_t \cdot (R_{\text{я}} + R_{\text{дп}} + R_{\text{к}}) ,$$

$R_{\text{я}}$, $R_{\text{дп}}$, $R_{\text{к}}$ – опори обмотки якоря, обмотки додаткових полюсів і компенсаційної обмотки;

α_t – коефіцієнт, який враховує зміну електричного опору обмоток у процесі нагрівання. Його значення дозволяє перейти від опору, виміряного за температури навколишнього середовища $Q_1 = 15^0$, до опору при робочій температурі $Q_2 = 110^0$, що відповідає усталеному тепловому режиму електродвигуна.

$$\alpha_t = 1 + 0,04 \cdot (Q_2 - Q_1),$$

$$\alpha_t = 1 + 0,004(110 - 15) = 1,36,$$

$$R_{\text{як}} = 1,36 \cdot 0,28 = 0,38 \text{ Ом}.$$

$R_{\text{епр}}$ – еквівалентний опір тиристорного перетворювача

$$R_{\text{епр}} = 0,5 \cdot U_{\text{кз}} \cdot \frac{k_{\text{г}} \cdot U_2}{I_{\text{н}}^2},$$

$$R_{\text{епр}} = 0,5 \cdot 0,055 \cdot \frac{1,35 \cdot 205}{51^2} = 2,9 \cdot 10^{-3} \text{ Ом}.$$

$R_{\text{тр}}$ – приведений до випрямленого струму опір трансформатора

$$R_{\text{тр}} = 0,16 \cdot \frac{\pi}{m} \cdot \frac{k_{\text{в}} \cdot U_2}{I_{\text{н}}},$$

$$R_{\text{тр}} = 0,16 \cdot \frac{\pi}{6} \cdot \frac{1,35 \cdot 205}{51} = 0,45 \text{ Ом},$$

$$R_{\Sigma} = 0,38 + 0,0029 + 0,25 = 0,633 \text{ Ом}.$$

3. Проти ЕРС двигуна:

$$E_{\text{д}} = U_{\text{н}} - I_{\text{н}} \cdot R_{\text{як}} - \Delta U_{\text{щ}},$$

де $U_{\text{н}}$ і $I_{\text{н}}$ – номінальна напруга і номінальний струм двигуна;

$\Delta U_{\text{щ}}$ – падіння напруги на щітках ($\approx 2 \text{ В}$).;

$$E_{\text{д}} = 220 - 51 \cdot 0,38 - 2 = 200 \text{ В}.$$

4. Постійні часу силового кола

4.1 Електромагнітна постійна часу

$$T_e = \frac{L_{\Sigma}}{R_{\Sigma}},$$

$$T_e = \frac{0,0158}{0,633} = 0,025 \text{ с}.$$

4.2 Постійна часу якірного кола двигуна

$$T_a = \frac{L_{\text{як}}}{R_{\text{як}}},$$

$$T_a = \frac{0,0157}{0,38} = 0,04 \text{ с}.$$

4.3 Електромеханічна постійна часу

$$T_m = J_{\text{пр}} \cdot \frac{R_{\Sigma}}{(k\Phi_n)^2} = J_{\text{пр}} \cdot \frac{R_{\Sigma}}{c^2};$$

де $c = k\Phi_n$; k – конструктивний коефіцієнт двигуна

$$k = \frac{pN}{2\pi a},$$

$$k = \frac{2 \cdot 558}{\pi \cdot 1} = 182,$$

$$c = 182 \cdot 0,0132 = 2,4,$$

$$T_m = 2,27 \cdot \frac{0,633}{2,4^2} = 0,25 \text{ с}.$$

Електромеханічна постійна часу електроприводу

$$T_{\text{м}} = J_{\text{пр}} \cdot \frac{R_{\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{пр}} \cdot \frac{R_{\Sigma}}{c^2};$$

де $c = k\Phi_{\text{н}}$;

k – конструктивний коефіцієнт двигуна

$$k = \frac{pN}{2\pi a},$$

$$k = \frac{2 \cdot 558}{\pi \cdot 1} = 182,$$

$$c = 182 \cdot 0,0132 = 2,4,$$

$$T_{\text{м}} = 2,27 \cdot \frac{0,633}{2,4^2} = 0,25 \text{ с}.$$

5. Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c},$$

$$k_{\text{д}} = \frac{1}{2,4} = 0,42.$$

6 Коефіцієнт посилення тиристорного перетворювача

Середнє значення коефіцієнта посилення визначається як:

$$k_{\text{тп}} = \frac{1,3 \cdot U_{\text{дн}}}{U_{\text{у}}},$$

де $U_{\text{у}}$ – напруга керування на вході СІФУ (приймаємо за умовами експлуатації $U_{\text{у}} = 10 \text{ В}$)

$$k_{\text{тп}} = \frac{1,3 \cdot 230}{10} = 29,9.$$

2.7 Вибір структури системи управління електроприводом

Система автоматичного управління електроприводом механізму переміщення візка мостового крана повинна гарантувати формування необхідних механічних характеристик як у статичних, так і в динамічних режимах роботи. До її ключових вимог відносять забезпечення високої якості перехідних процесів – тобто мінімального часу досягнення заданого значення регульованої координати при допустимому рівні перерегулювання, а також здатність оптимізувати режим роботи електроприводу відповідно до встановлених технічних і експлуатаційних критеріїв.

1. Замкнуті системи автоматичного управління.

Досягти таких показників можливо лише за використання замкнутих систем автоматичного управління. Саме у замкненому контурі вплив збурень компенсується зворотним зв'язком, а характер перехідних процесів визначається взаємодією постійних часу окремих елементів системи. Для покращення цих властивостей у структуру САУ вводять регулятори корекції, що дозволяють компенсувати інерційність, нелінійність і неоднорідність характеристик об'єкта керування, завдяки чому забезпечуються необхідні часові та частотні характеристики системи.

2. Загальний підхід до побудови САУ

Проектування будь-якої системи керування передбачає послідовну реалізацію декількох етапів. Спочатку формується структурна схема, яка відображає взаємозв'язок усіх складових елементів системи. Після цього в систему вводяться коригувальні регулятори, що забезпечують задані співвідношення постійних часу та необхідні запаси стійкості. Для електроприводів ефективним є принцип підлеглого регулювання: кожен регульований параметр має власний контур, який містить об'єкт керування (електродвигун або механізм) та відповідний регулятор. Кількість послідовно включених регуляторів, як правило, відповідає кількості параметрів, що підлягають стабілізації.

3. Вибір структури системи керування електроприводом візка

Для електроприводу механізму переміщення візка мостового крана найбільш доцільною є двоконтурна система керування. У ній внутрішній контур формується регулятором струму, який забезпечує швидкодію електроприводу, стабілізує струм якоря та запобігає перевантаженням двигуна під час пуску чи гальмування. Зовнішній контур швидкості, у свою чергу, замкнутий по проти-ЕРС двигуна, що дозволяє регулювати швидкість руху візка без застосування додаткових датчиків швидкості.

Зворотний зв'язок по ЕДС використовується там, де вимоги до точності стабілізації швидкості не є надмірно високими. У такому випадку замість тахогенератора застосовується вимірювання проти-ЕРС за допомогою сумування двох сигналів: одного, пропорційного напрузі на якорі двигуна, та іншого – пропорційного падінню напруги на активному опорі якоря (сигнал струмової компенсації).

4. Переваги обраної структури

Використання двоконтурної структури з зворотним зв'язком по ЕДС має низку істотних переваг. По-перше, внутрішній контур струму забезпечує високу швидкодію та стійкість системи, дозволяючи оперативно реагувати на зміни навантаження. По-друге, така система є конструктивно простішою, оскільки не потребує додаткових сенсорів швидкості. По-третє, подібна структура відзначається економічністю та високою надійністю, що робить її оптимальною для більшості промислових електроприводів кранів. Крім того, внутрішній струмовий контур забезпечує ефективний захист двигуна від перевантажень у динамічних режимах.

2.8 Синтез послідовного коректуючого пристрою контуру струму

Контур струму виконує роль внутрішнього контуру в системі підлеглого регулювання і є ключовою складовою всієї структури керування. Саме від швидкодії та якості роботи цього контуру значною мірою залежать динамічні влас-

тивості електроприводу загалом, а також точність регулювання швидкості обертання двигуна. Для проведення оптимізації параметрів контуру робляться кілька припущень: вплив проти-ЕРС двигуна у процесі розрахунків не враховується, на виході датчика струму встановлюється згладжувальний фільтр у вигляді аперіодичної ланки, що дозволяє зменшити пульсації напруги до рівня, за якого система імпульсно-фазового управління (СІФУ) функціонує без збоїв; додатковий фільтр передбачено і на вході СІФУ для стабілізації сигналу керування.

Структурна алгоритмічна схема контуру струму з урахуванням зазначених припущень наведена на рис. 2.1.

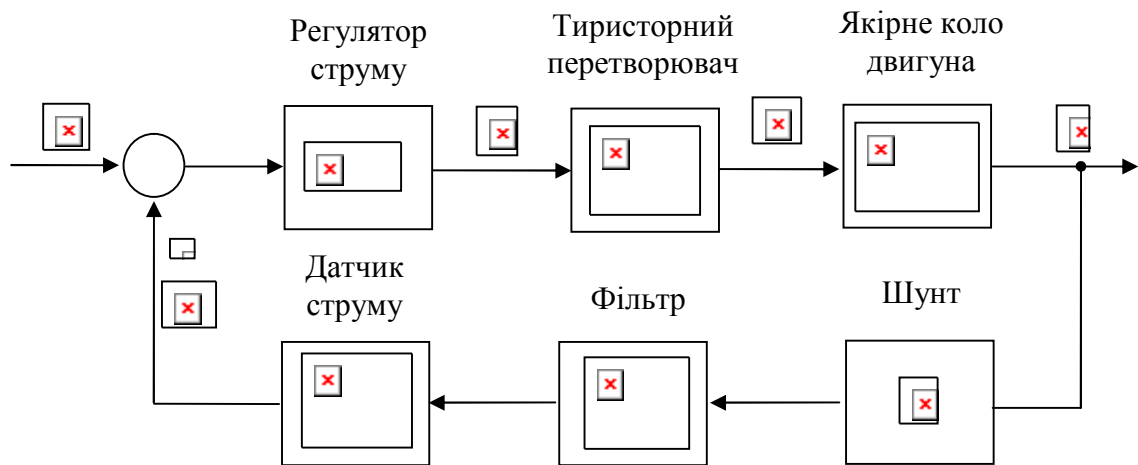


Рисунок 2.1 – Алгоритмічна схема внутрішнього контуру регулювання струму

Для реалізації зворотного зв'язку за струмом обирається вимірювальний шунт. Його номінальний струм визначається максимальним значенням струму робочого перевантаження двигуна $I_{нш} = I_{max} = 2I_n = 2 \cdot 57 = 114 A$, а номінальна напруга – величиною $U_{шн} = 0,075 B$. Коефіцієнт підсилення шунта при цьому становить:

$$k_{ш} = \frac{U_{шн}}{I_{шн}},$$

$$k_{\text{ш}} = \frac{75 \cdot 10^{-3}}{114} = 0,658 \cdot 10^{-3} \text{ B / A}$$

Для подальшої обробки сигналу струму та забезпечення гальванічної розв'язки між силовою та керувальною частинами схеми застосовується датчик струму. Його коефіцієнт підсилення визначається з виразу:

$$k_{\text{дт}} \leq \frac{24}{k_{\text{ш}} I_{\text{max}}},$$

$$k_{\text{дт}} \leq \frac{24}{0,658 \cdot 10^{-3} \cdot 114} = 319,9 \text{ B / A}.$$

У подальших розрахунках коефіцієнт підсилення датчика струму приймається рівним $k_{\text{дт}} = 320$.

Аналіз схеми на рис. 2.1 показує, що об'єкт регулювання включає одну аперіодичну ланку з великою постійною часу та чотири аперіодичні ланки з малими постійними часу. Для можливості застосування класичної методики синтезу послідовних коригуючих пристроїв необхідно перетворити схему до вигляду з одиничним зворотним зв'язком. Перетворена алгоритмічна схема представлена на рис. 2.2.

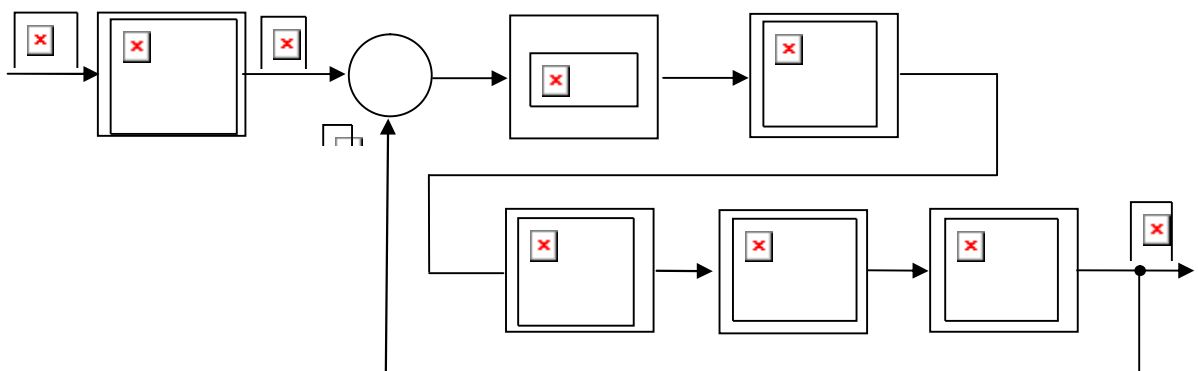


Рисунок 2.2 – Перетворена структурна схема контуру струму з одиничним зворотним зв'язком

Аперіодичні ланки з малими постійними часу $T_{\text{дт}}$, $T_{\text{ф}}$ і $T_{\text{тп}}$ можуть бути об'єднані в еквівалентну одну ланку з малою некомпенсованою постійною часу:

$$T_{\text{мт}} = T_{\text{тп}} + T_{\text{ф}} + T_{\text{дт}}.$$

Під час розрахунку параметрів регулятора струму у реальних системах значення $T_{\text{мт}}$ задається у межах, рекомендованих практикою $(4 \div 10) \cdot 10^{-3} \text{ с}$. У подальшому приймаємо $T_{\text{мт}} = 0,005 \text{ с}$. Форсувальна ланка $(1 + T_{\text{дт}}p)$ при розрахунках не враховується, оскільки її внесок у динаміку струмового контуру є незначним. Розрахункова алгоритмічна модель контуру подана на рис. 2.3.

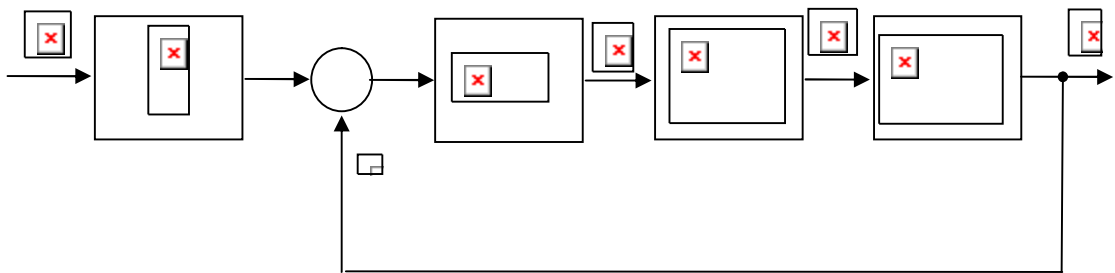


Рисунок 2.3 – Розрахункова алгоритмічна схема струмового контуру

Коефіцієнт підсилення зворотного зв'язку за струмом визначається виразом:

$$k_{\text{т}} = k_{\text{дт}} k_{\text{ш}},$$

$$k_{\text{т}} = 320 \cdot 0,658 \cdot 10^{-3} = 0,21 \text{ В / А}.$$

Оптимізація параметрів контуру струму здійснюється на основі модульного критерію оптимізації. Бажана передавальна функція розімкненого контуру має вигляд:

$$W_{\text{от}}(p) = \frac{1}{2T_{\text{мт}}p(T_{\text{мт}}p + 1)}.$$

Передавальна функція об'єкта керування:

$$W_{\text{обт}}(p) = \frac{k_{\text{тп}}k_{\text{т}}/R_{\Sigma}}{(1+T_{\text{мт}}p)(1+T_{\text{еп}}p)}.$$

Для того щоб забезпечити відповідність реальної системи бажаним динамічним характеристикам, регулятор струму повинен мати передавальну функцію виду:

$$W_{\text{рт}}(p) = \frac{W_{\text{бт}}(p)}{W_{\text{обт}}(p)} = \frac{\frac{1}{2T_{\text{мт}}p(T_{\text{мт}}p+1)}}{\frac{k_{\text{тп}}k_{\text{т}}/R_{\Sigma}}{(1+T_{\text{мт}}p)(1+T_{\text{еп}}p)}} = \frac{R_{\Sigma}}{2T_{\text{мт}}k_{\text{тп}}k_{\text{т}}} \cdot \frac{1+T_{\text{еп}}p}{p} = k_{\text{рт}} \cdot \frac{1+T_{\text{еп}}p}{p}, \quad (2.3)$$

$k_{\text{рт}} = \frac{R_{\Sigma}}{2T_{\text{мт}}k_{\text{тп}}k_{\text{т}}}$ – коефіцієнт підсилення ПІ-регулятора струмового контуру.

Отже, отримана функції (2.3) відповідає пропорційно-інтегральній (ПІ) ланці, що є типовим рішенням для внутрішніх контурів струму в системах тиристорного електроприводу. З урахуванням цього передавальна функція замкнутого оптимізованого струмового контуру набуває вигляду:

$$W_{\text{зт}}(p) = \frac{I_{\text{д}}(p)}{U_{\text{зт}}(p)} = \frac{1/k_{\text{т}}}{2T_{\text{мт}}^2p + 2T_{\text{мт}} + 1}.$$

2.9 Синтез послідовного коректуючого пристрою контура ЕРС

Синтез послідовного коригувального пристрою для контуру ЕРС здійснюється на основі алгоритмічної схеми системи керування, наведеної на рис. 2.4. У цій схемі оптимізований за модульним критерієм контур струму представлено у вигляді передавальної функції:

$$W_{\text{т}}(p) = \frac{1/k_{\text{т}}}{2T_{\text{мт}}p + 1},$$

(де доданок $2T_{\text{мт}}^2p^2$ у знаменнику, як малий за величиною, не враховується).

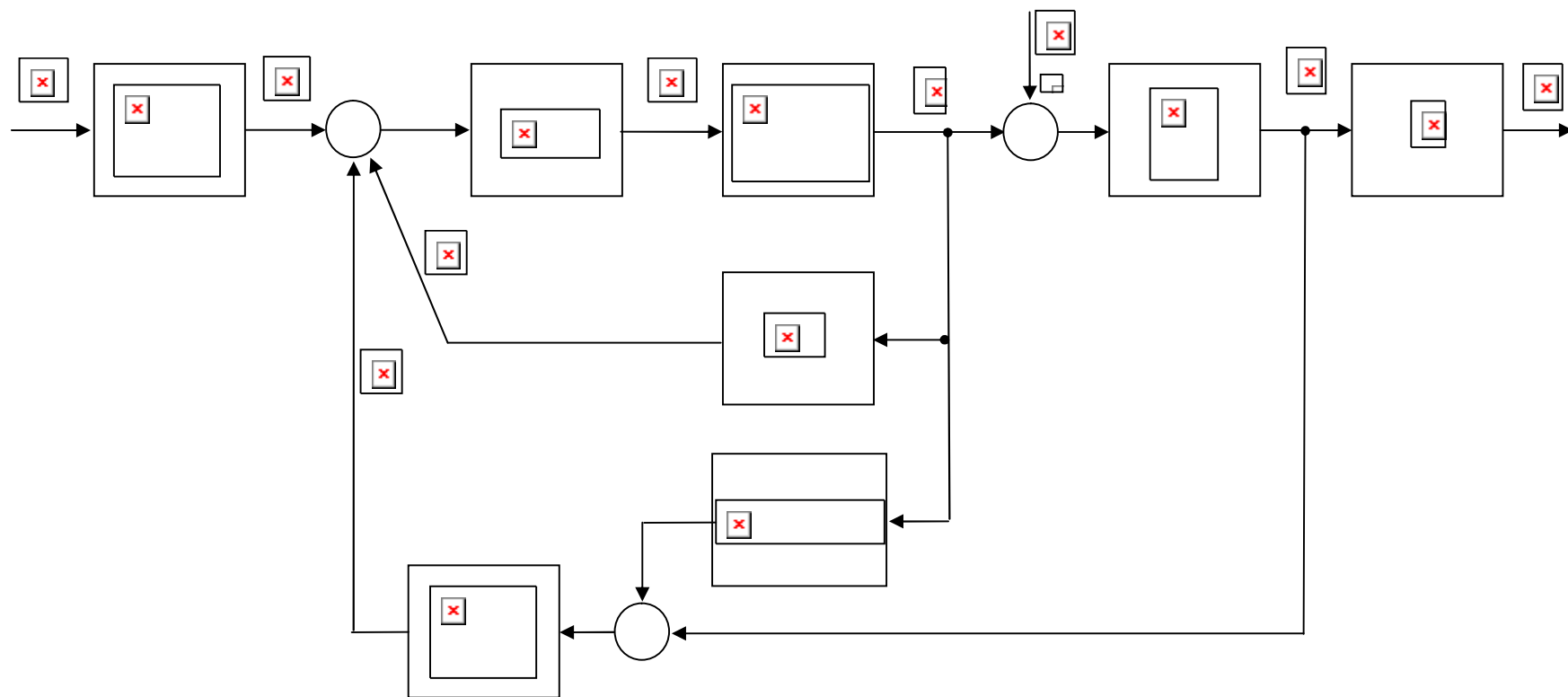


Рисунок 2.4 – Алгоритмічна схема регулювання швидкості з ЕРС-зв'язком

Для реалізації струмової компенсації задається коефіцієнт підсилення

$$k_{\text{TK}} = k_{\text{H}} R_{\text{ЯК}},$$

де k_n – коефіцієнт підсилення в контурі зворотного зв'язку за напругою.

За такого вибору $k_{\text{тк}}$ на вході регулятора ЕРС формується сигнал, пропорційний фактичній електрорушійній силі двигуна:

$$U_{33\text{e}} = U_{\text{H}} - U_{\text{T}} = \frac{k_{\text{H}}}{T_{\text{a}} p + 1} E_{\text{ДВ}} \text{ .}$$

Алгоритмічна схема, подана на рис. 2.4, може бути приведена до більш компактного вигляду, поданого на рис. 2.5.

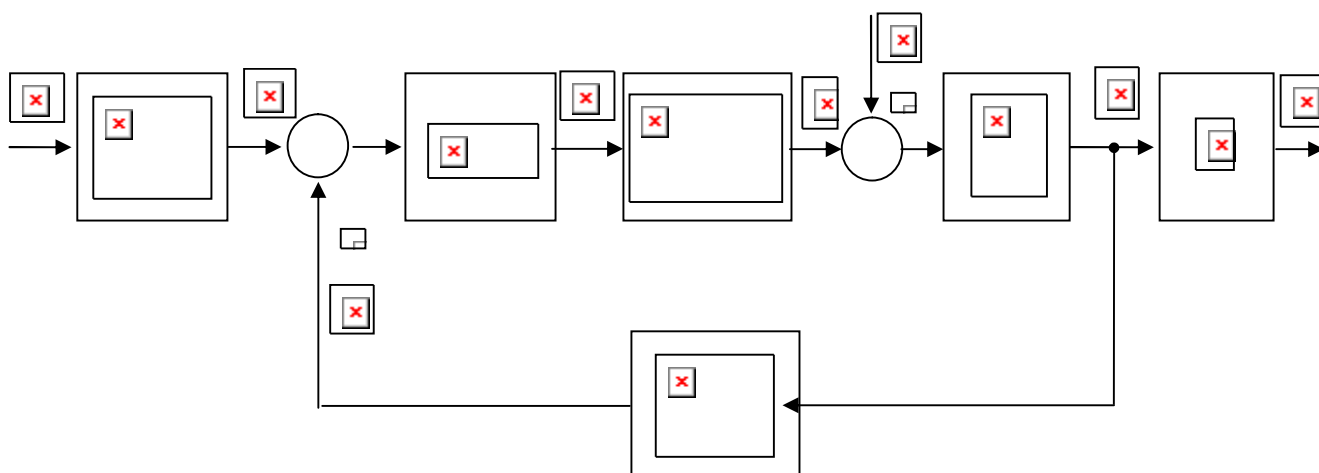


Рисунок 2.5 – Трансформована алгоритмічна схема системи

У зворотному зв'язку за ЕРС знаходиться аперіодична ланка $\frac{1}{T_a p + 1}$ з постійною часу T_a . Це може зумовити короткочасні перевищення струму при зміні завдання. Щоб усунути цей ефект, на вході системи встановлюється ідентична аперіодична ланка $\frac{1}{T_a p + 1}$.

Контур ЕРС налаштовується за модульним критерієм. Для синтезу послідовного коригувального пристрою схема з рис. 2.5 перетворюється до структури з одиничним зворотним зв'язком (рис. 2.6).

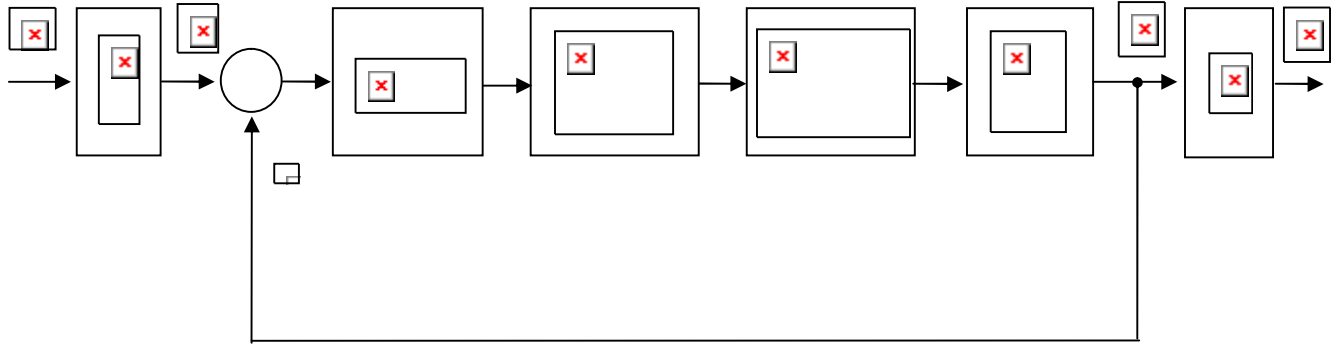


Рисунок 2.6 – Розрахункова схема системи з одиничним зворотним зв'язком

Передавальна функція об'єкта регулювання в контурі ЕРС має вигляд:

$$W_{\text{обє}}(p) = \frac{k_n R_{\Sigma} / k_T}{p T_M (T_{\mu e} p + 1)},$$

де $T_{\mu e}$ – мала некомпенсована постійна часу контуру ЕРС

$$T_{\mu e} = T_a + 2T_{\mu T},$$

$$T_{\mu e} = 0,04 + 2 \cdot 0,005 = 0,05 \text{ с}.$$

Через наявність у складі $T_{\mu e}$ постійної часу T_a швидкодія контуру ЕРС виявляється в $\frac{T_{\mu e}}{T_{\mu T}}$ раз нижчою, а статична точність гіршою порівняно з системою, де використовують зворотний зв'язок безпосередньо за швидкістю.

Бажана розімкнена передавальна функція для контуру ЕРС при оптимізації за симетричним критерієм має вигляд

$$W_{\text{бє}}(p) = \frac{1}{2T_{\mu e} p (1 + T_{\mu e} p)}.$$

Передавальна функція регулятора ЕДС

$$W_{pe}(p) = \frac{W_{\text{бе}}(p)}{W_{\text{обе}}(p)} = \frac{1}{\frac{k_{\text{н}} R_{\Sigma}}{k_{\text{т}} T_{\text{м}}} \cdot \frac{2T_{\text{м}} p(1 + T_{\text{м}} p)}{p(1 + T_{\text{м}} p)}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\text{м}} k_{\text{н}} R_{\Sigma}} = k_{\text{пе}}. \quad (2.4)$$

Ця передавальна функція відповідає типовому пропорційному регулятору.

Коефіцієнт підсилення зворотного зв'язку за напругою визначається виразом:

$$k_{\text{н}} = \frac{U_{\text{зе max}}}{E_{\text{дв max}}},$$

де $E_{\text{дв max}}$ – максимальна ЕРС двигуна;

$U_{\text{зе max}}$ – напруга завдання, яка відповідає $E_{\text{дв max}}$ (приймаємо

$$U_{\text{зе max}} = 24 \text{ В}).$$

Якщо прийняти $E_{\text{дв max}}$ рівною ЕРС у режимі холостого ходу, тобто $E_{\text{дв max}} = U_{\text{н}} = 220 \text{ В}$, отримуємо значення коефіцієнта зворотного зв'язку

$$k_{\text{н}} = \frac{24}{220} = 0,1091.$$

Після цього визначається коефіцієнт підсилення струмової компенсації:

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}};$$

$$k_{\text{тк}} = 0,1091 \cdot 0,38 = 0,04.$$

Система, оптимізована за модульним критерієм, описується такими передавальними функціями для швидкості та струму щодо задавальних і збурювальних сигналів:

$$W_{\omega_d, U_{3e}}(p) = \frac{\omega_d(p)}{U_{3e}(p)} = \frac{k_d / k_H}{2T_{\mu e}^2 p^2 + 2T_{\mu e} p + 1};$$

$$W_{\omega_d, I_{ст}} = \frac{\omega_d(p)}{I_{ст}(p)} = \frac{-2T_{\mu e} R_{\Sigma} k_d / T_M (1 + T_{\mu e} p)}{2T_{\mu e}^2 p^2 + 2T_{\mu e} p + 1};$$

$$W_{I_d, U_{3e}} = \frac{I_d(p)}{U_{3e}(p)} = \frac{T_M}{R_{\Sigma} k_H} \cdot \frac{p}{2T_{\mu e}^2 p^2 + 2T_{\mu e} p + 1};$$

$$W_{I_d, I_{ст}} = \frac{I_d(p)}{I_{ст}(p)} = \frac{1}{2T_{\mu e}^2 p^2 + 2T_{\mu e} p + 1}.$$

Структура системи представлена на рис. 2.7.

Подальше перетворення схеми на рис. 2.7 дає модель, подану на рис. 2.8. Воно виконується за умови, що коефіцієнт підсилення струмової компенсації вибрано як $k_{тк} = k_H R_{як}$.

На схемі використовуються такі позначення:

U_{ic} – вихід інтегральної частини струмового регулятора;

$k_{пс}$, $k_{іс}$ – коефіцієнти пропорційної та інтегральної складових ПІ-регулятора струму, які визначаються з передавальної функції регулятора:

$$W_{пс}(p) = \frac{R_{\Sigma}}{2T_{\mu c} k_{тп} k_c} \cdot \frac{1 + T_e p}{p} = \frac{R_{\Sigma} T_e}{2T_{\mu c} k_{тп} k_c} + \frac{R_{\Sigma}}{2T_{\mu c} k_{тп} k_c} \frac{1}{p} = k_{пс} + \frac{k_{іс}}{p},$$

$$k_{пс} = \frac{R_{\Sigma} T_e}{2T_{\mu c} k_{тп} k_c}, \quad k_{іс} = \frac{R_{\Sigma}}{2T_{\mu c} k_{тп} k_c}.$$

У передавальних функціях двигуна замість T_M використано співвідношення

$$T_M = \frac{R_{\Sigma} \cdot J_{д\Sigma}}{(k\Phi_H)^2} = R_{\Sigma} J_{д\Sigma} \cdot k_d^2; \quad k_d = 1 / k\Phi_H.$$

Також враховано внутрішній зворотний зв'язок за ЕРС двигуна.

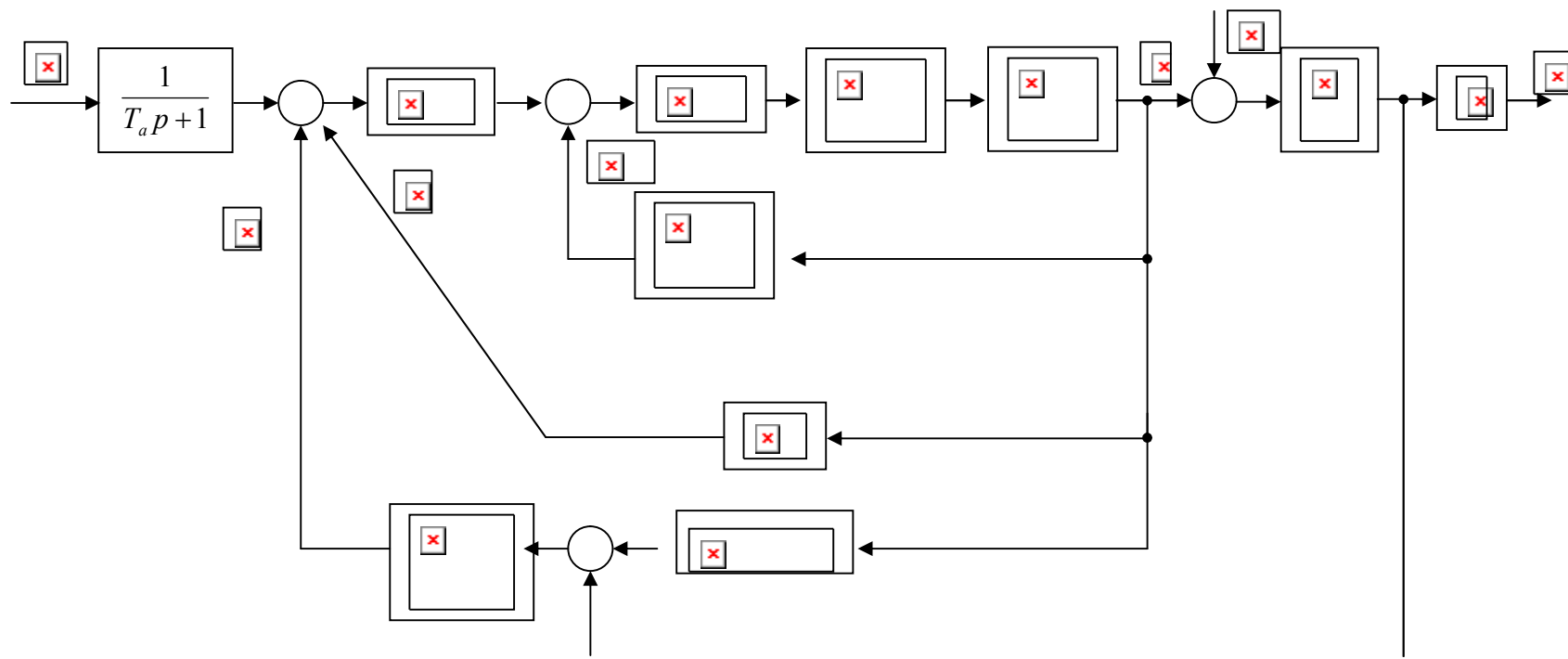


Рисунок 2.7 – Структурна схема системи регулювання швидкості із зворотним зв'язком за ЕРС

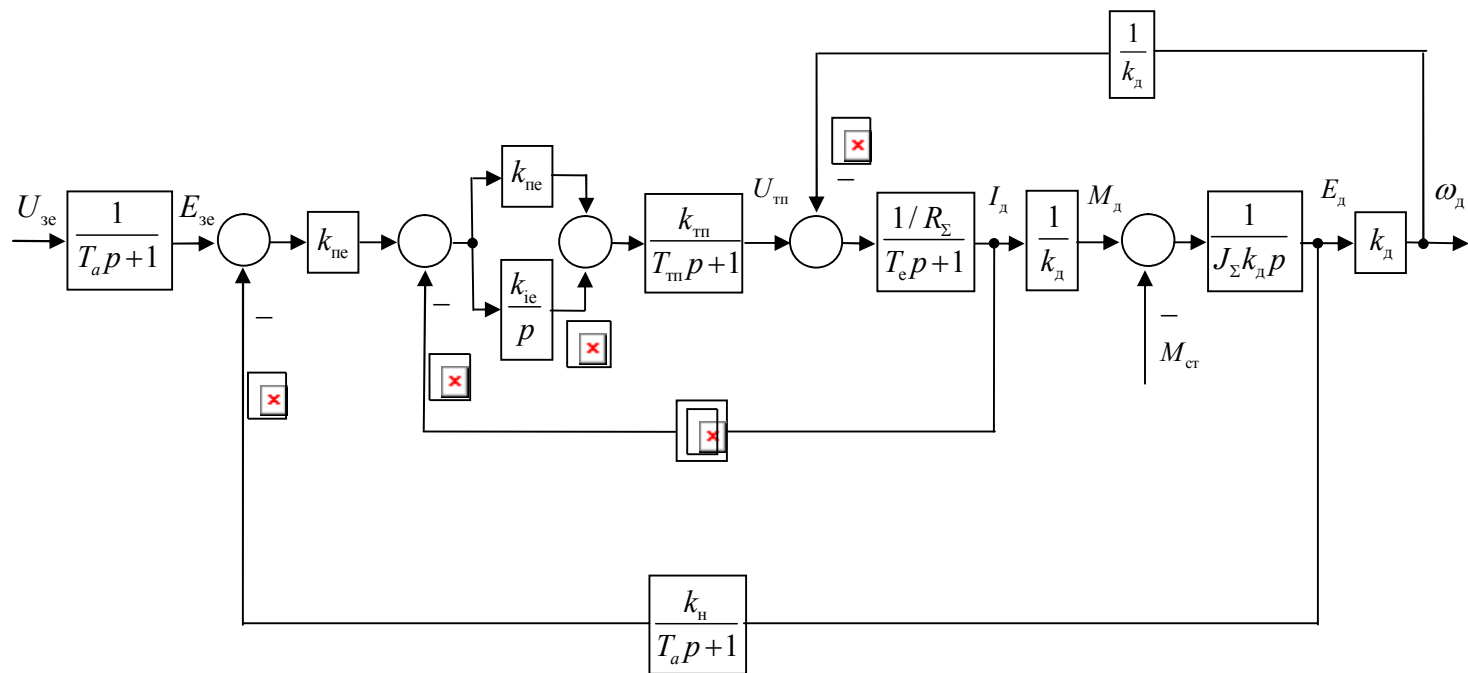


Рисунок 2.8 – Модифікована алгоритмічна схема системи

Систему можна подати у ще більш компактному вигляді (рис. 2.9). Тут оптимізований контур струму еквівалентний послідовному з'єднанню аперіодичної та інтегруючої ланок з одиничним зворотним зв'язком, а вихід першої ланки відповідає рівню електроприводу P .

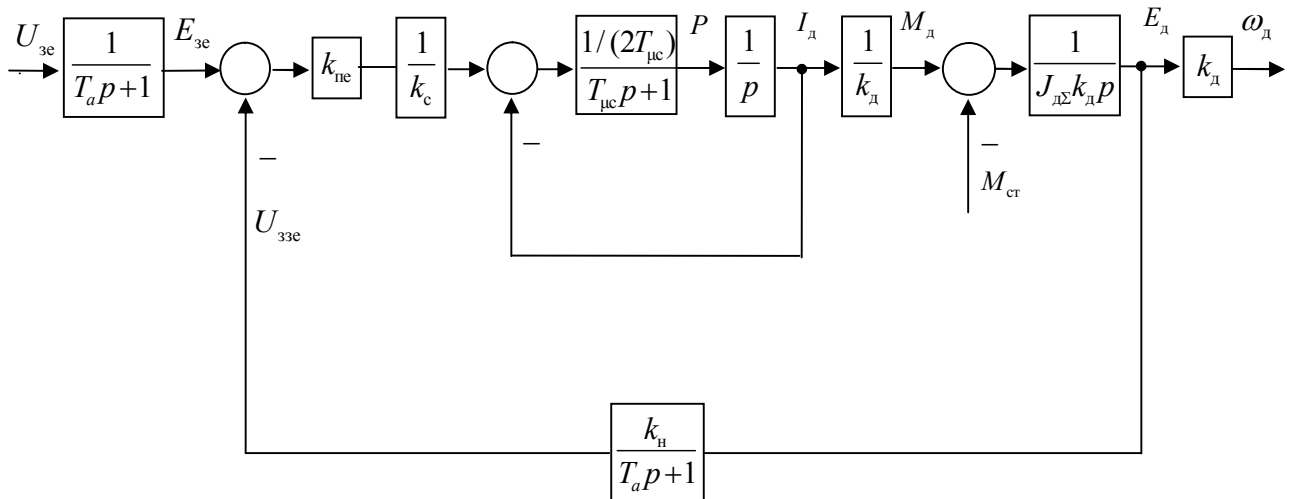


Рисунок 2.9 – Спрощена алгоритмічна структура системи

2.10 Дослідження динамічних характеристик системи

Розрахунок перехідних процесів виконувався у MATLAB. Цей програмний комплекс дозволяє моделювати динаміку систем як засобами передавальних функцій, так і у векторно-матричній формі, зокрема через моделі SIMULINK.

Для аналізу динаміки була створена модель системи відповідно до схеми на рис. 2.8. Вона показана на рис. 2.10. Спрощений варіант, що відповідає рис. 2.9, подано на рис. 2.11.

На рис. 2.12 та 2.13 наведено графіки перехідних процесів швидкості ω_d і струму I_d при дії задавальних та збурювальних впливів.

Налаштування системи за симетричним оптимумом призводить до суттєвого перерегулювання швидкості (43,3%). Для його зменшення на вхід системи введено фільтр із постійною часу $4T_{\mu c}$, що знижує перерегулювання до 8,1%, хоча й дещо подовжує час першого узгодження. На графіках криві $\omega_d = f(t)$ і $I_d = f(t)$ з фільтром і без нього позначені як варіанти 1 та 2.

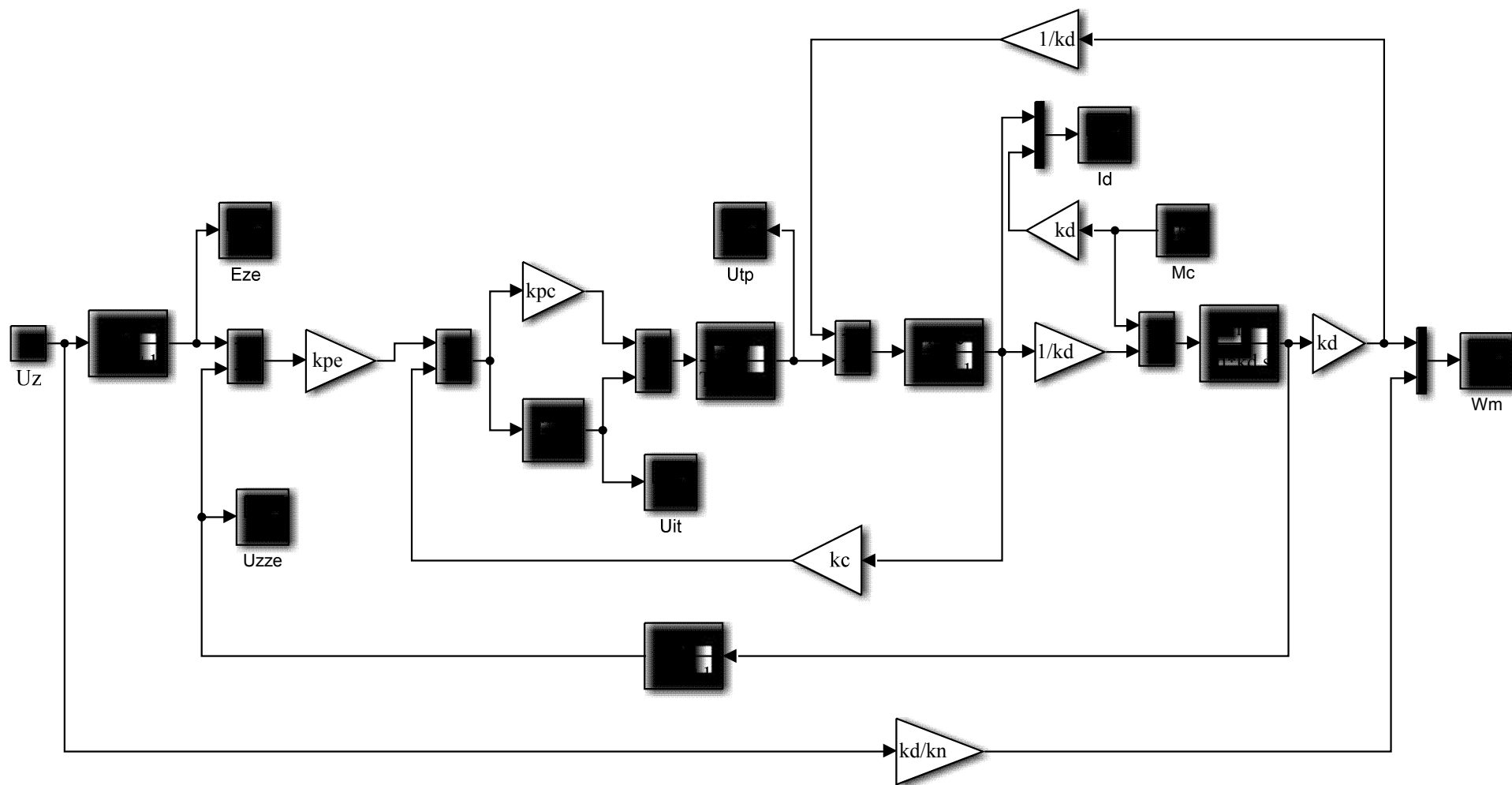


Рисунок 2.10 – Імітаційна модель системи, розроблена в SIMULINK/MATLAB

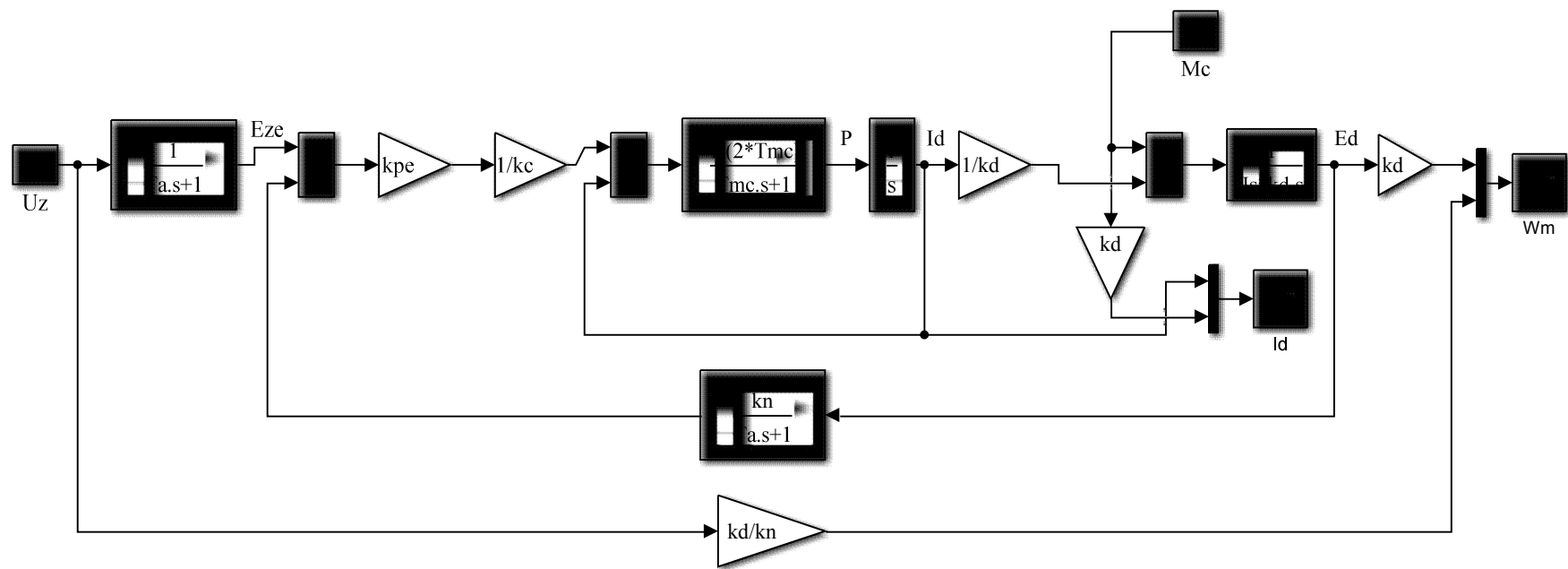


Рисунок 2.11 – Спрощена імітаційна модель системи, розроблена в SIMULINK/MATLAB

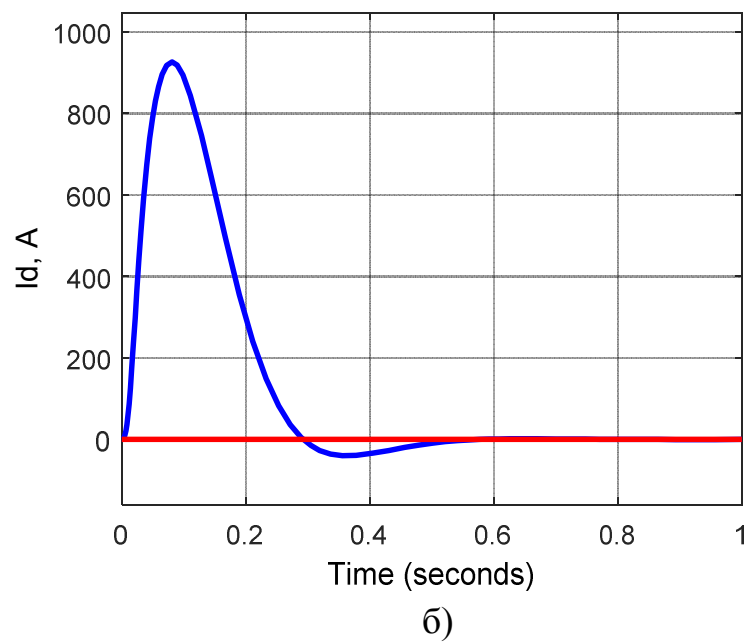
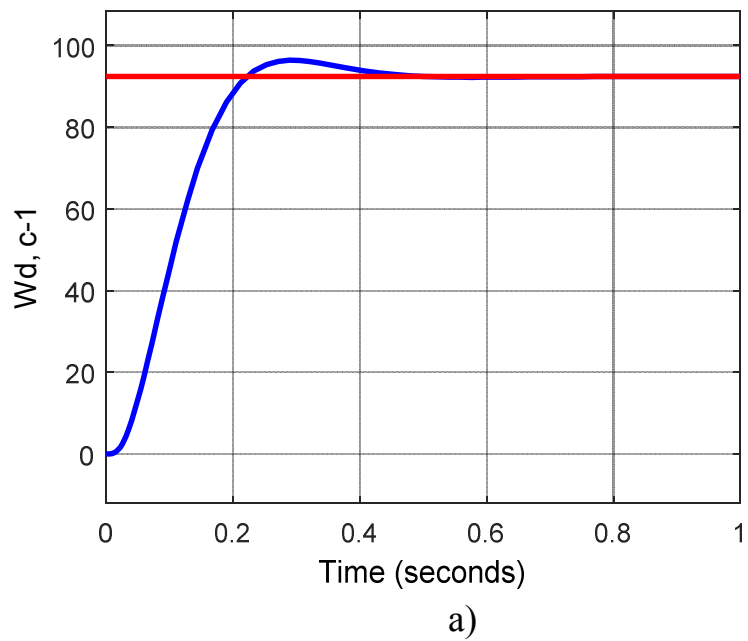


Рисунок 2.12 – Перехідні процеси швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) при дії задавального сигналу

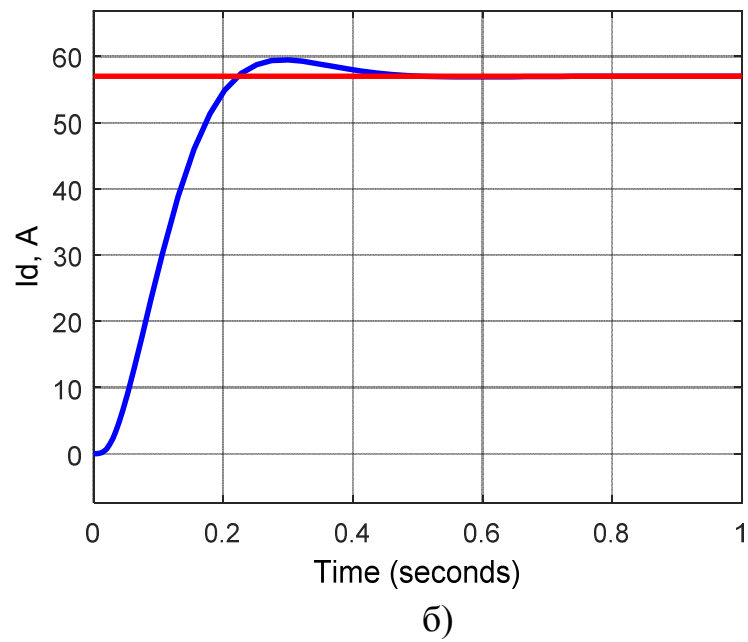
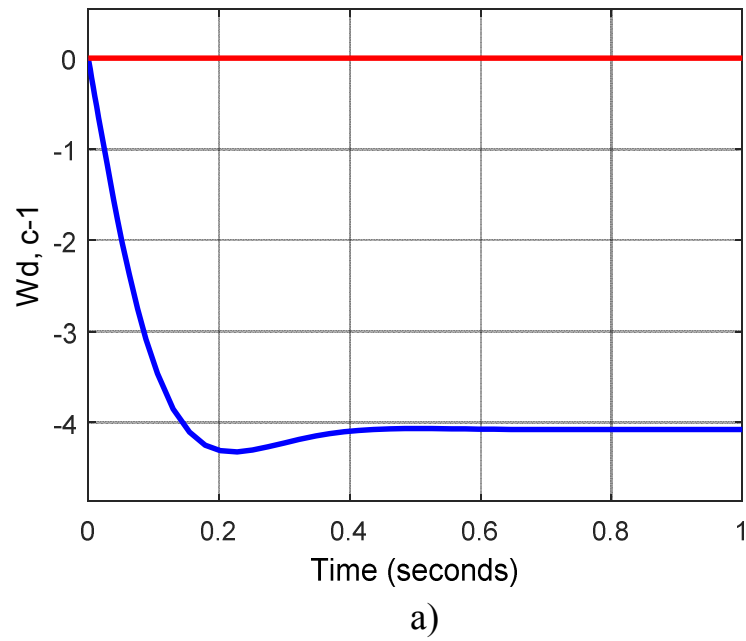


Рисунок 2.13 – Перехідні процеси швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) при збурюючому впливі

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На основі аналізу вимог до електроприводів мостових кранів встановлено, що для механізму пересування візка найбільш доцільним є використання електроприводу постійного струму за схемою ТП–Д. Такий вибір зумовлений необхідністю забезпечення високої точності регулювання швидкості та надійності при змінних навантаженнях, характерних для кранових механізмів. Для керування координатами електроприводу застосовано систему підлеглого регулювання, що дозволяє розділити функції окремих контурів та підвищити якість керування у широкому діапазоні режимів.

2. Виконано повний комплекс розрахунків щодо вибору електродвигуна та силового обладнання, включаючи аналіз теплового режиму, допустимих перевантажень та розрахунок швидкісної й навантажувальної діаграм. Розраховане значення еквівалентного моменту підтвердило відповідність двигуна вимогам за нагрівостійкістю, а оцінка перевантажувальної здатності показала, що двигун здатний працювати у реальних умовах експлуатації без ризику перевищення допустимих струмових та теплових меж.

3. Визначено параметри силового кола електроприводу, необхідні для подальшого математичного опису електроприводу та розрахунку системи автоматичного керування. Отримані значення були використані при побудові алгоритмічних моделей та при синтезі регуляторів у структурі системи підлеглого регулювання.

4. Обґрунтовано вибір структури системи автоматичного управління, яка повинна забезпечувати стабільні механічні характеристики електроприводу у статичних і динамічних режимах. З метою забезпечення високої точності та оптимальної динаміки обрано двоконтурну систему підлеглого регулювання: внутрішній контур струму та зовнішній контур швидкості. Система замкнута по ЕРС, що дозволяє покращити відтворення швидкісної координати та підвищити стійкість при змінних робочих навантаженнях. Така структура є найбільш ефективною для механізмів мостових кранів та аналогічних пристроїв.

5. Здійснено синтез послідовних коригуючих пристроїв для кожного контуру регулювання. Ураховуючи, що форму перехідних процесів визначають співвідношення часових сталих системи, було застосовано спеціальні коректори, які дозволили оптимізувати ці співвідношення. Для контуру напруги використано П-регулятор. Для контуру струму, де потрібне швидке та точне відпрацювання завдання при мінімальних коливаннях, синтезовано ПІ-регулятор. Для контуру ЕРС обрано П-регулятор, завдяки чому контури струму та ЕРС були оптимізовані за модульним критерієм, забезпечивши високу якість регулювання й відсутність значних перерегулювань у робочому діапазоні.

6. Проведено всебічне моделювання синтезованої системи за допомогою прикладного пакету MATLAB. У середовищі SIMULINK було розроблено детальну модель електроприводу, що відповідає алгоритмічній схемі підлеглого регулювання. Моделювання виконано як за передавальними функціями, так і у векторно-матричній формі, що дозволило всебічно дослідити поведінку системи за задавальних та збурювальних впливів. Отримані результати підтвердили правильність вибраної структури, доцільність використання модульного критерію оптимізації та адекватність синтезованих регуляторів.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПОПЕРЕЧНИХ РОЗГОЙДУВАНЬ ВАНТАЖУ

3.1 Механічна частина електроприводу

У традиційних моделях електроприводу переміщення візка мостового крана механічна частина системи часто подається у вигляді приведеної жорсткої ланки. Такий підхід значно спрощує математичний опис, проте не дозволяє врахувати реальні деформаційні властивості механічних елементів. Насправді механічні зв'язки крана не є абсолютно жорсткими, а податливість каната та коливання підвішеного вантажу суттєво впливають на динамічну поведінку електроприводу, особливо у перехідних режимах пуску та гальмування.

Одним із характерних проявів цієї податливості є поперечне розгойдування вантажу на канаті, яке виникає щоразу при зміні швидкості руху візка. У механізмі присутні дві маси, що рухаються поступально: маса візка m_1 та маса вантажу m_2 . Вони з'єднані гнучкою підвіскою – канатом, який у рамках дослідження розглядається як нерозтяжна нитка. Через це підвішений вантаж у перехідних режимах поводить себе як маятник, що змінює характер силової взаємодії між масами.

Під час рівноприскореного руху під дією сили F рівноважне положення вантажу зміщується, і сила тяжіння $G = m_2 g$ розкладається на дві компоненти: складову F_k , що компенсується натягом каната і складову $F_{12} = a_2 m_2$, яка створює поперечну силу між масами та визначає їх взаємодію під час руху.

Схему сил показано на рис. 3.1.

На рис. 3.1 прийняті наступні позначення : a_1 , a_2 – прискорення мас системи; g – прискорення вільного падіння.

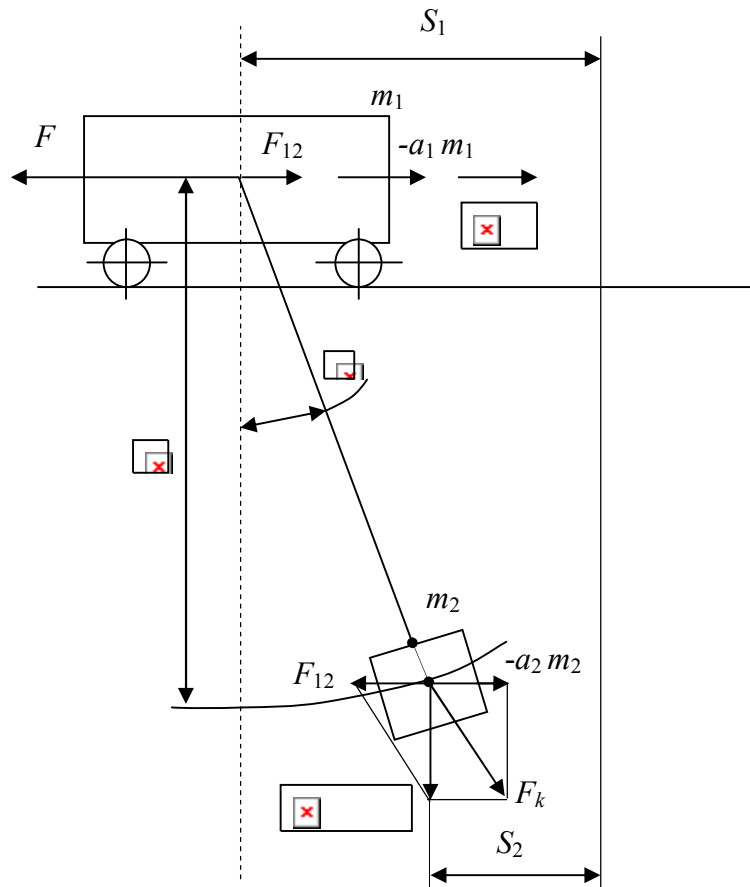


Рисунок 3.1 – Схема дії сил на підвішений вантаж під час прискореного руху візка

Сила горизонтальної взаємодії між візком і вантажем визначається виразом:

$$F_{12} = G \tan \alpha = \frac{G}{l \cos \alpha} (S_1 - S_2). \quad (3.1)$$

Після приводу (3.1) до валу електродвигуна, отримуємо вираз для приведенного моменту взаємодії між масами:

$$M_{12} = \frac{G \rho^2}{l \cos \alpha} (\varphi_1 - \varphi_2) = c_{12} (\varphi_1 - \varphi_2),$$

де ρ – радіус приведення;

c_{12} – приведена жорсткість зв'язку між масами m_1 і m_2 ;

M_{12} – приведений момент сили пружної взаємодії між масами m_1, m_2 ;

φ_1, φ_2 – приведені координати відповідно маси візка та вантажу

$$\varphi_1 = S_1 / \rho, \quad \varphi_2 = S_2 / \rho;$$

l – довжина каната.

Жорсткість зв'язку між масою візка та вантажем c_{12} , підвішеним на канаті, є величиною, що залежить від кута відхилення підвіски α . Через це така жорсткість є нелінійною. Проте при невеликих кутах відхилення, які характерні для більшості технологічних процесів переміщення візка, можна прийняти наближення $l \cos \alpha \approx l$.

За цих умов жорсткість підвіски набуває практично сталої величини:

$$c_{12} \approx G \rho^2 / l = \text{const}$$

Отриманий вираз було виведено зі статичної рівноваги, але його можна застосовувати і в динамічних режимах, якщо знехтувати вертикальним переміщенням вантажу при малих кутах відхилення. Така модель адекватно описує реальні процеси та дозволяє побудувати математичну модель двомасової системи, придатну для подальшого аналізу її динамічних властивостей.

3.2 Рівняння стану одномасової системи

Для переходу до аналізу повної двомасової моделі електроприводу доцільно спершу розглянути одномасову систему, що відповідає спрощеному випадку, коли механічна частина приводного механізму замінюється приведеною масою та одним моментом інерції. Така модель є базовою та дозволяє наочно простежити вплив окремих ланок системи керування на динаміку електроприводу.

Рівняння стану одномасової системи, побудовані згідно з алгоритмічною схемою на рис. 2.8, мають вигляд:

$$\left\{ \begin{aligned} \frac{d\omega_{\text{д}}}{dt} &= \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dI_{\text{д}}}{dt} &= -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} &= -\frac{k_{\text{т}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} I_{\text{д}} - \frac{1}{T_{\mu\text{с}}} U_{\text{тп}} - \\ &\quad + \frac{k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{іс}} + \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} E_{\text{зе}} - \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іс}}}{dt} &= -k_{\text{іс}}k_{\text{с}} I_{\text{д}} + k_{\text{пе}}k_{\text{іс}} E_{\text{зе}} - k_{\text{пе}}k_{\text{іс}} U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} &= -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{ззе}}}{dt} &= \frac{k_{\text{н}}}{k_{\text{д}}T_{\text{а}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{aligned} \right. \quad (3.2)$$

Для компактного та уніфікованого запису система диференціальних рівнянь подається у векторно-матричній формі. Введемо вектор стану $\vec{X}(t)$ розмірності n

$$\vec{X}(t) = [\omega_{\text{д}}(t), I_{\text{д}}(t), U_{\text{тп}}(t), U_{\text{іс}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T$$

який описує внутрішні змінні системи, та вектор керування $\vec{U}(t)$ розмірності m

$$\vec{U}(t) = [U_{\text{зе}}(t), M_{\text{ст}}(t)]^T,$$

де T – символ транспонування.

Тоді рівняння стану та рівняння виходу набувають вигляду:

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

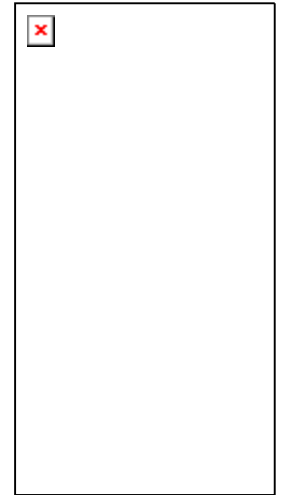
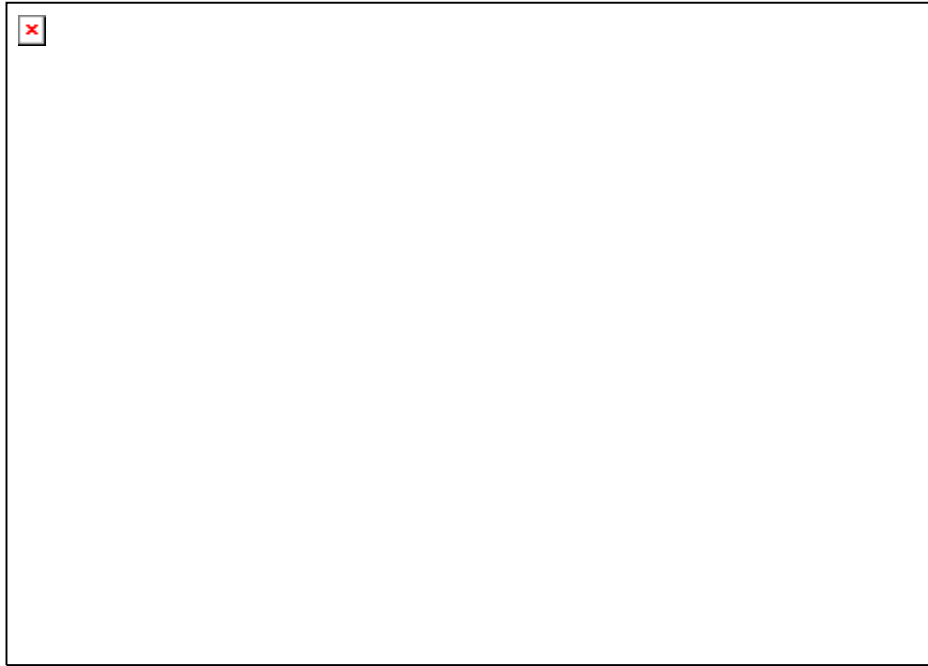
де $\mathbf{A}$ – матриця стану системи розмірності $n \times n$;

$\mathbf{B}$ – матриця управління розмірності $n \times m$;

$\mathbf{C}$ - матриця виходу розмірності $r \times n$;

$\vec{Y}(t)$ - вектор виходу розмірності r .

Матриці $\mathbf{A}$ і $\mathbf{B}$ одномасової системи:



Як було зазначено раніше, алгоритмічна схема одномасової системи може бути подана також у спрощеному вигляді (рис. 2.9), де оптимізований за модульним критерієм контур струму представлений як послідовне з'єднання аперіодичної та інтегруючої ланок з одиничним зворотним зв'язком. У такій структурі вихід першої ланки є величиною, пропорційною ривку електроприводу P , що дозволяє спростити аналіз перехідних процесів.

Рівняння стану цієї спрощеної моделі:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{л}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{л}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = -\frac{k_{\text{д}}}{2T_{\text{мс}}^2} M_{\text{д}} - \frac{1}{T_{\text{мс}}} P + \frac{k_{\text{пе}}}{2T_{\text{мс}}^2 k_{\text{с}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\text{мс}}^2 k_{\text{с}}} U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.3)$$

Вектор стану та вектор керування мають вигляд:

$$\vec{X}(t) = [\omega_{\text{д}}(t), M_{\text{д}}(t), P(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T,$$

$$\vec{U}(t) = [U_{\text{зе}}(t), M_{\text{ст}}(t)]^T,$$

Відповідні матриці стану **A** та управління **B** спрощеної одномасової системи:

✖

✖

Таким чином, одномасова модель є вихідною базою для подальшого формування двомасової моделі системи.

3.3 Математична модель двомасової системи

Кінематична схема механізму переміщення візка крана з урахуванням податливості канату може бути адекватно описана двомасовою моделлю, де маси візка та вантажу взаємодіють через пружний та в'язкий зв'язок. Така модель точніше відображає реальні перехідні процеси, зокрема коливання вантажу, що виникають при пуску та гальмуванні.

Загальний випадок математичного опису включає момент внутрішнього в'язкого тертя та записується рівняннями:

$$\begin{cases} J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma} - M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_{\text{д}} - \omega_{\text{м}}); \\ J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = M_{\text{д}} - M_{\Sigma}. \end{cases} \quad (3.4)$$

Сумарний момент M_{Σ} , що передається через пружний зв'язок, визначається як сума пружної складової $M_{\text{пр}}$ та моменту внутрішнього в'язкого тертя $M_{\text{вт}} = \beta_{12}(\omega_{\text{д}} - \omega_{\text{м}})$:

$$M_{\Sigma} = M_{\text{пр}} + \beta_{12}(\omega_{\text{д}} - \omega_{\text{м}}), \quad (3.5)$$

β_{12} – коефіцієнт внутрішнього в'язкого тертя. У подальших розрахунках приймаємо значення $\beta_{12} = 0$.

З урахуванням (3.5) система (3.4) у формі Коші набуває вигляду:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta_{12}}{J_{\text{д}\Sigma}} \omega_{\text{м}}. \end{array} \right. \quad (3.6)$$

Подальше моделювання виконується для системи підлеглого регулювання швидкості, що має: контур струму, оптимізований за модульним критерієм; контур ЕРС, налаштований за симетричним критерієм.

Алгоритмічну схему двомасової системи наведено на рис. 3.2, а еквівалентне подання – на рис. 3.3.

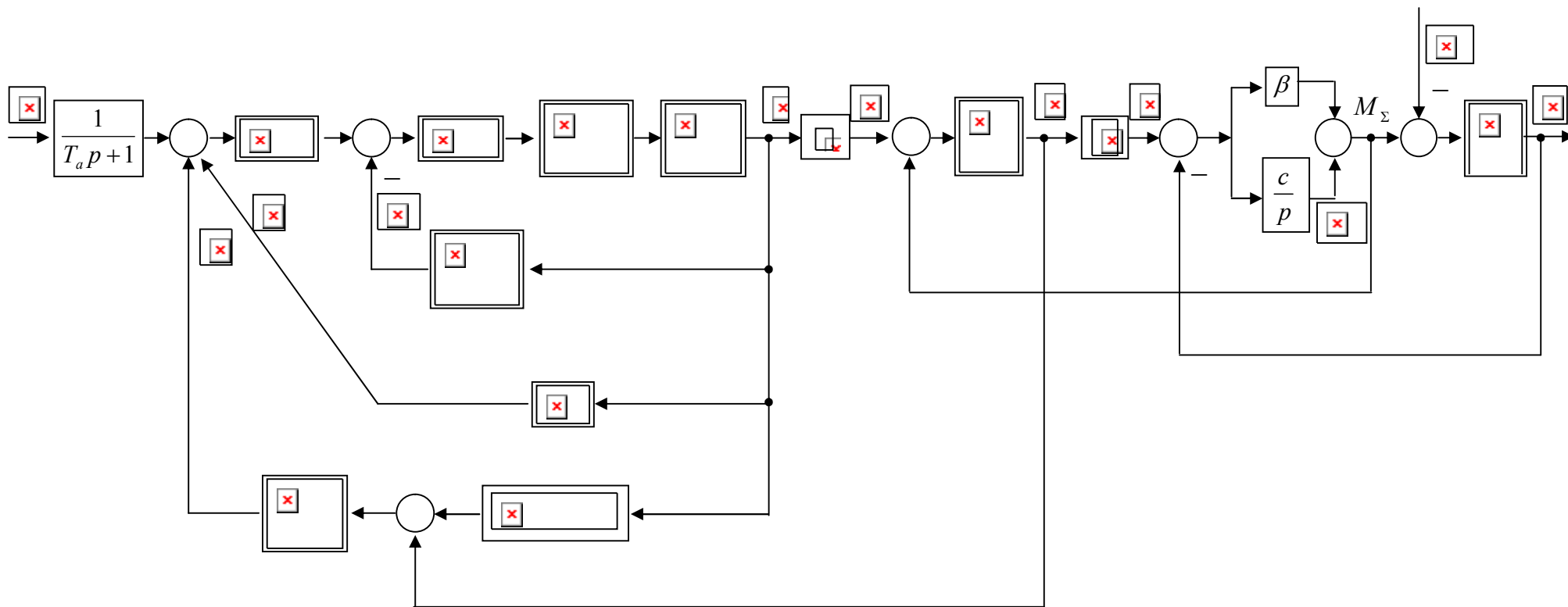


Рисунок 3.2 – Алгоритмічна схема двомасової системи

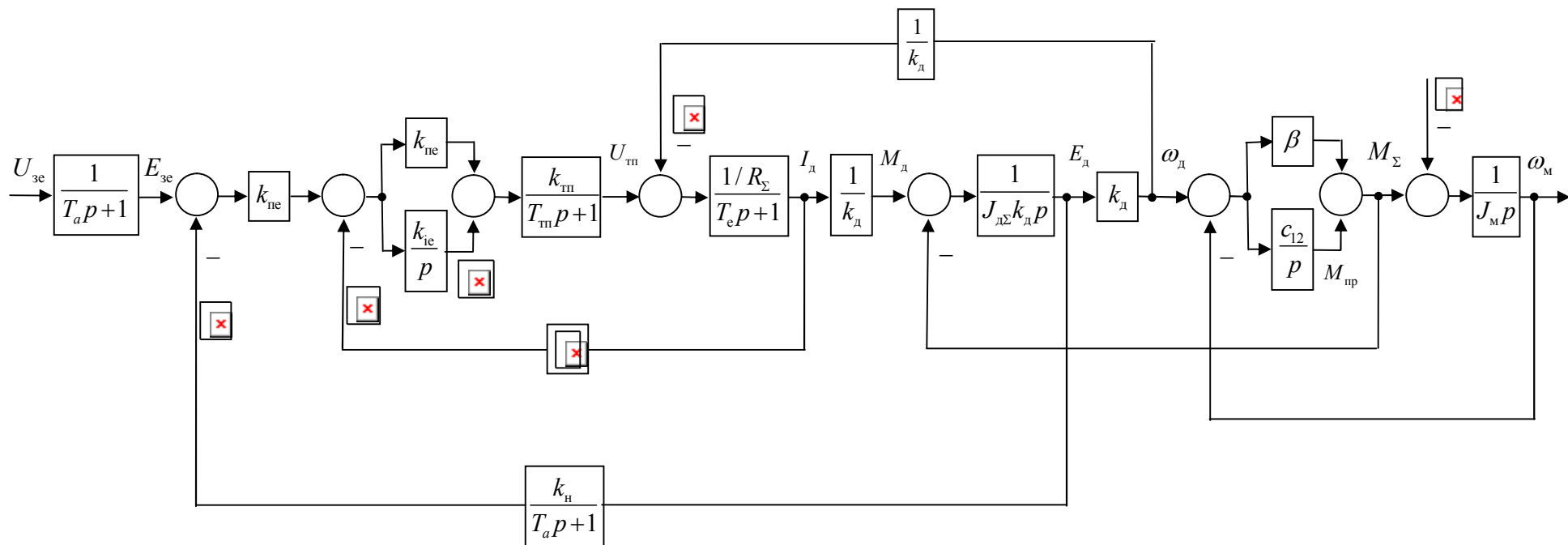


Рисунок 3.3 – Модифікована алгоритмічна схема двомасової системи

Об'єднання рівнянь електромеханічної частини (3.2) та двомасової моделі системи (3.6) дає загальну систему рівнянь стану електроприводу:

$$\left\{ \begin{aligned} \frac{d\omega_{\text{м}}}{dt} &= -\frac{\beta}{J_{\text{м}}} \omega_{\text{м}} + \frac{1}{J_{\text{м}}} M_{\text{пп}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пп}}}{dt} &= c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} &= \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пп}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dI_{\text{д}}}{dt} &= -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} &= -\frac{k_{\text{с}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} I_{\text{д}} - \frac{1}{T_{\mu\text{с}}} U_{\text{тп}} - \\ &\quad + \frac{k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{іс}} + \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} E_{\text{зе}} - \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іс}}}{dt} &= -k_{\text{іс}}k_{\text{с}}I_{\text{д}} + k_{\text{пе}}k_{\text{іс}}E_{\text{зе}} - k_{\text{пе}}k_{\text{іс}}U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} &= -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} &= \frac{k_{\text{н}}}{k_{\text{д}}T_{\text{а}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{aligned} \right. \quad (3.7)$$

Запишемо систему рівнянь стану двомасової системи у векторно-матричній формі, доповнивши її рівнянням виходу системи.

Вектор стану $\vec{X}(t)$ і вектор управління $\vec{U}(t)$ двомасової моделі системи

$$\vec{X}(t) = [\omega_{\text{м}}(t), M_{\text{пп}}(t), \omega_{\text{д}}(t), I_{\text{д}}(t), U_{\text{тп}}(t), U_{\text{іт}}(t), E_{\text{зе}}(t), U_{\text{ззе}}(t)]^T,$$

$$\vec{U}(t) = [U_{\text{зе}}(t), M_{\text{ст}}(t)]^T,$$

Структура матриць **A** і **B** двомасової моделі системи

$$\mathbf{A} = \begin{vmatrix} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & I_{\text{д}} & U_{\text{тп}} & U_{\text{іс}} & E_{\text{зе}} & U_{\text{ззе}} \\ -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ -c_{12} & 0 & c_{12} & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta}{J_{\text{д}\Sigma}} & -\frac{1}{J_{\text{д}\Sigma}} & -\frac{\beta}{J_{\text{д}\Sigma}} & \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} & 0 & 0 & 0 & 0 \\ 0 & 0 & -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} & -\frac{1}{T_{\text{е}}} & \frac{1}{R_{\Sigma}T_{\text{е}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{\text{с}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{1}{T_{\mu\text{с}}} & \frac{k_{\text{тп}}}{T_{\mu\text{с}}} & \frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} & -\frac{k_{\text{пе}}k_{\text{пс}}k_{\text{тп}}}{T_{\mu\text{с}}} \\ 0 & 0 & 0 & -k_{\text{іс}}k_{\text{с}} & 0 & 0 & k_{\text{пе}}k_{\text{іс}} & -k_{\text{пе}}k_{\text{іс}} \\ 0 & 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & k_{\text{іе}} \\ 0 & 0 & \frac{k_{\text{н}}}{k_{\text{д}}T_a} & 0 & 0 & 0 & 0 & -\frac{1}{T_a} \end{vmatrix}$$

$$\mathbf{B} = \begin{vmatrix} U_{\text{зе}} & M_{\text{ст}} \\ 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_a} & 0 \\ 0 & 0 \end{vmatrix}$$

Аналогічно одномасовій системі, двомасову модель можна спростити, подаючи її як послідовне з'єднання аперіодичної та інтегруючої ланок із замкненим контуром зворотного зв'язку (рис. 3.4). Це дозволяє значно полегшити подальший синтез та аналіз системи керування, зберігаючи при цьому вплив податливості механічних зв'язків.

Рівняння стану спрощеної моделі системи мають вигляд:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta_{12}}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = -\frac{k_{\text{д}}}{2T_{\mu\text{с}}^2} M_{\text{д}} - \frac{1}{T_{\mu\text{с}}} P + \frac{1}{2T_{\mu\text{с}}^2 k_{\text{с}}} U_{\text{іе}} + \frac{k_{\text{ре}}}{2T_{\mu\text{с}}^2 k_{\text{с}}} E_{\text{зе}} - \frac{k_{\text{ре}}}{2T_{\mu\text{с}}^2 k_{\text{с}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{array} \right. \quad (3.8)$$

Вектор стану та вектор управління :спрощеної двомасової моделі системи

$$\vec{X}(t) = \left\{ \omega_{\text{м}}(t), M_{\text{пр}}(t), \omega_{\text{д}}(t), M_{\text{д}}(t), P(t), E_{\text{зе}}(t), U_{\text{ззе}}(t), U_{\text{іе}}(t) \right\}^T,$$

$$\vec{U}(t) = [U_{\text{зе}}(t), M_{\text{ст}}(t)]^T,$$

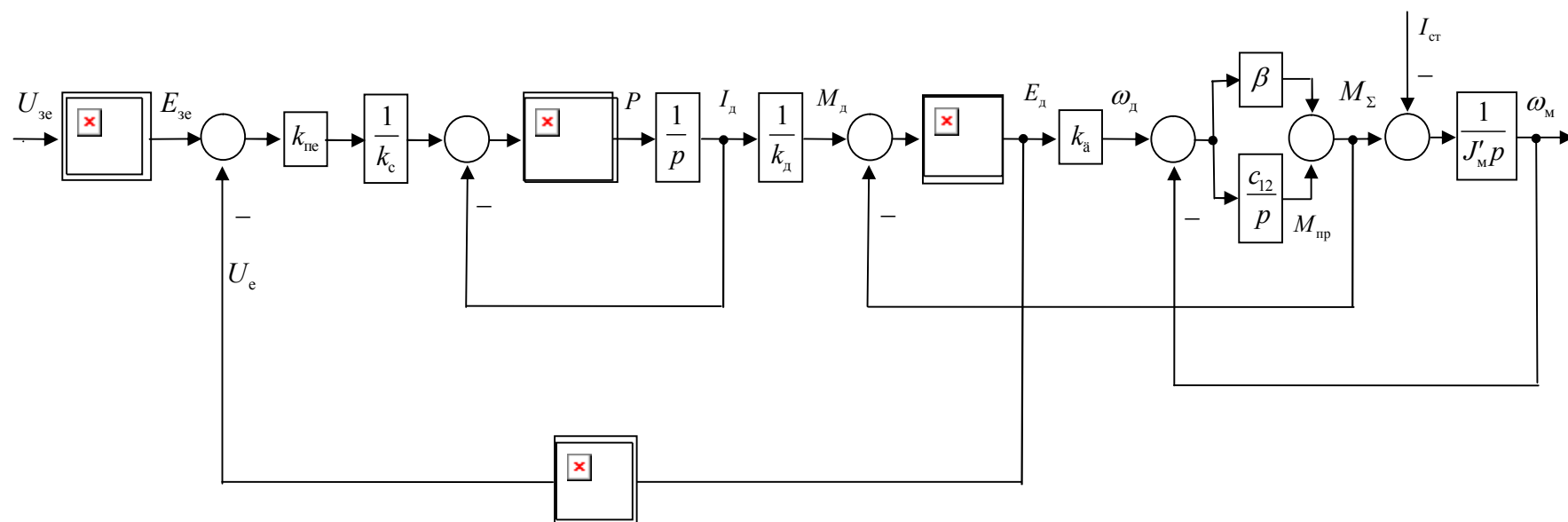


Рисунок 3.4 – Спрощена модель двомасової системи

Матриці **A** і **B** спрощеної двомасової системи:

$$\mathbf{A} = \begin{vmatrix} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & M_{\text{д}} & P & U_{\text{іе}} & E_{\text{зе}} & U_{\text{ззе}} \\ -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ -c & & c & 0 & 0 & 0 & 0 & 0 \\ \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & -\frac{k_{\text{д}}}{2T_{\text{мс}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{1}{2T_{\text{мс}}^2 k_{\text{с}}} & \frac{k_{\text{пе}}}{2T_{\text{мт}}^2 k_{\text{с}}} & +\frac{k_{\text{пе}}}{2T_{\text{мт}}^2 k_{\text{с}}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 & 0 \\ 0 & 0 & \frac{k_{\text{н}}}{T_{\text{а}} K_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_{\text{а}}} & 0 \\ 0 & & 0 & 0 & 0 & k_{\text{іс}} & -k_{\text{іс}} & 0 \end{vmatrix} \quad \mathbf{B} = \begin{vmatrix} U_{\text{зе}} & M_{\text{ст}} \\ 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_{\text{а}}} & 0 \\ 0 & 0 \\ 0 & 0 \end{vmatrix}$$

3.4 Розрахунок перехідних процесів двомасової системи

Для аналізу динамічної поведінки електроприводу з податливою механічною частиною необхідно визначити параметри, що входять до математичної моделі двомасової системи. У першу чергу обчислюються моменти інерції елементів приводу, приведені до валу електродвигуна.

Момент інерції якоря двигуна разом з жорстко з'єднаними частинами і приведеним моментом інерції візка:

$$J_{\text{д}\Sigma} = \delta \cdot J_{\text{дв}} + \frac{m_{\text{в}} \cdot V_{\text{в}}^2}{\omega_{\text{н}}^2} ;$$

$$J_{\text{д}\Sigma} = 1,3 \cdot 0,425 + \frac{8 \cdot 10^3 \cdot 0,65^2}{83,4^2} = 0,5525 + 0,4859 = 1,038 \text{ кг} \cdot \text{м}^2.$$

Аналогічно визначаються момент інерції вантажу та вантажозахопного пристрою, які утворюють інерційну масу механізму:

$$J_{\text{м}} = \frac{(G_{\text{вп}} + m_0) \cdot V_{\text{в}}^2}{\omega_{\text{н}}^2};$$

$$J_{\text{м}} = \frac{(37 + 1) \cdot 10^3 \cdot 0,65^2}{83,7^2} = 2,29 \text{ кг} \cdot \text{м}^2.$$

Усі інші параметри, що входять до рівнянь двомасової моделі, були встановлені та уточнені у попередніх підрозділах.

Розрахунок перехідних процесів двомасової системи здійснювався із використанням пакету прикладних програм MATLAB, що дозволяє реалізувати високоточне моделювання динаміки електроприводу. Для цього були розроблені математичні моделі системи у середовищі SIMULINK, що забезпечує наочне представлення алгоритмічної структури системи та взаємодії її елементів. Схема моделі двомасової системи, побудована відповідно до алгоритмічної схеми, наведеної на рис. 3.3, та рівнянь стану (3.17), представлена на рис. 3.5. Для спрощеного аналізу була розроблена спрощена схема моделі, що відповідає алгоритмічній схемі рис. 3.4 та рівнянням стану (3.8), яка наведена на рис. 3.6.

Крім того, для моделювання можна використовувати блок State-Space у SIMULINK, що дозволяє задавати імена матриць двомасової системи, визначати вектор стану та вектор управління. Розмір вектора вихідних координат $\vec{Y}(t)$ визначає кількість вікон Scope, у яких відображаються графіки перехідних процесів. При цьому вікно завдання матриць **A**, **B**, **C** і **D** показане на рис. 3.8 і з'являється при подвійному клацанні мишею на блоці State-Space. Якщо досліджуються всі змінні стану, матриця виходу **C** є одиничною. У випадку часткового аналізу, одиниці встановлюються лише у тих рядках, які відповідають обраним змінним стану, а решта елементів матриці **D** дорівнює нулю.

У додатку Д наведено файл, що містить початкові параметри системи, включаючи матриці **A**, **B**, **C** і **D**.

Графічне відображення перехідних процесів двомасової системи наведено на рис. 3.9–3.12. Для зручності аналізу та зменшення обсягу інформації на екрані монітора та друкованих матеріалах були відображені лише перші чотири змінні стану. Як показують отримані графіки, перехідні процеси швидкості механізму (вантажу) та моменту пружності по задаючій дії, а також динаміка всіх змінних стану по збурюючій дії характеризуються слабо затухаючими коливаннями, що свідчить про реалістичне відтворення впливу податливості механічних зв'язків на динамічні характеристики системи.

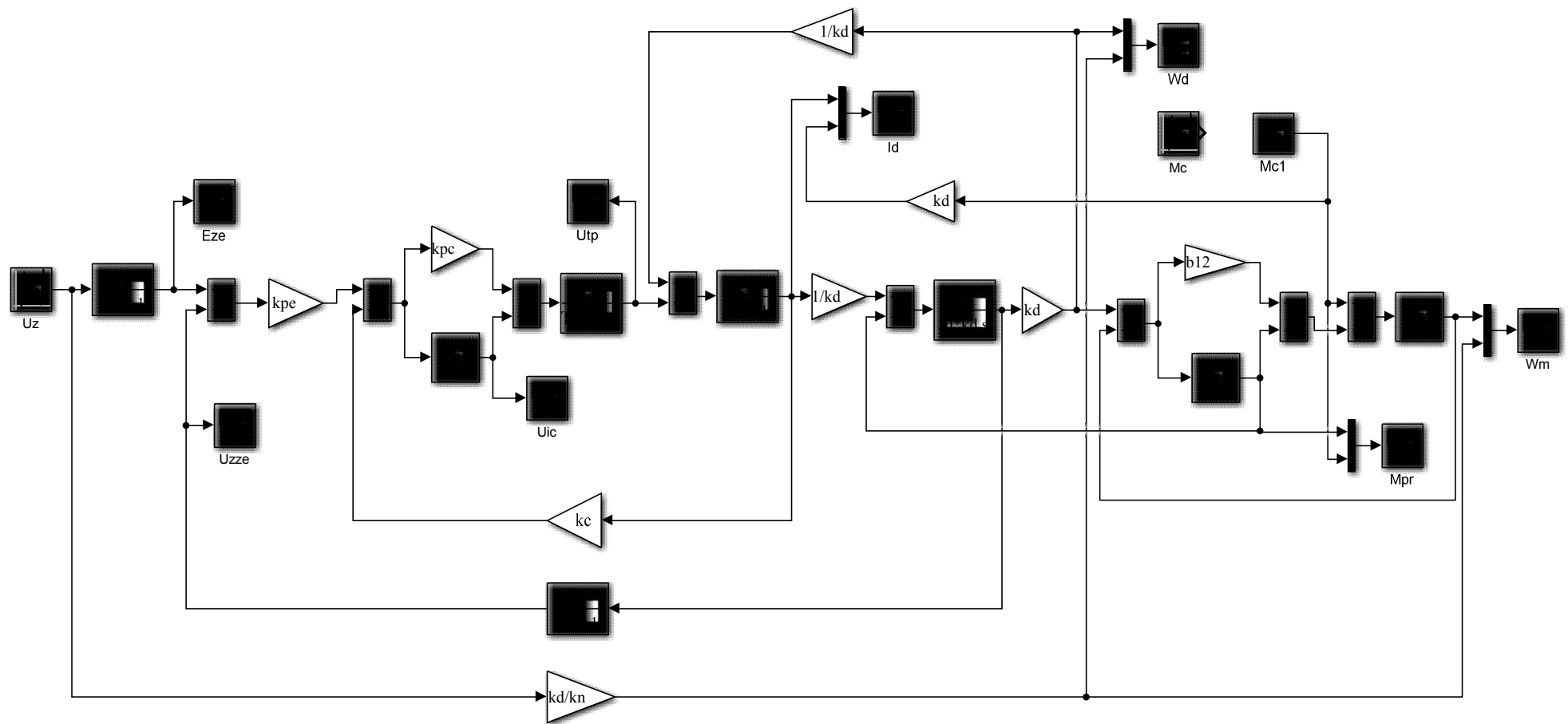


Рисунок 3.5 – – Модель двомасової системи, реалізована в середовищі SIMULINK MATLAB

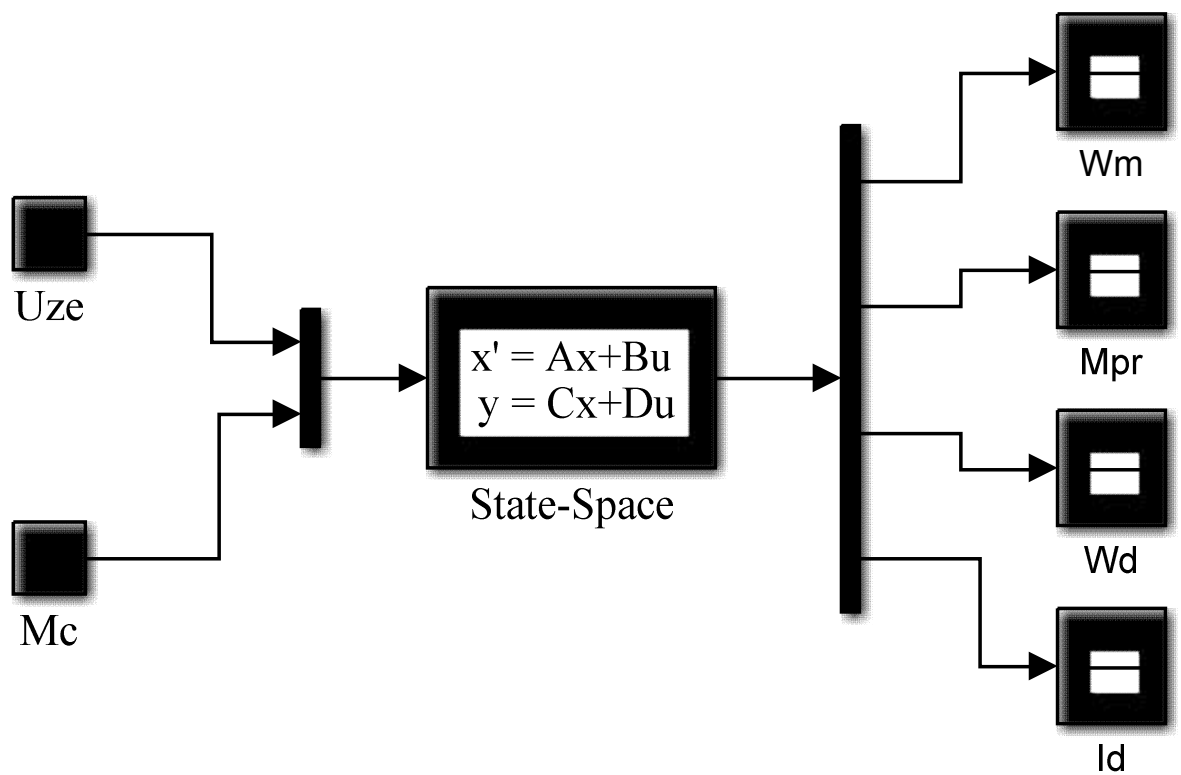


Рисунок 3.7 – Модель системи з використанням блоку State-Space

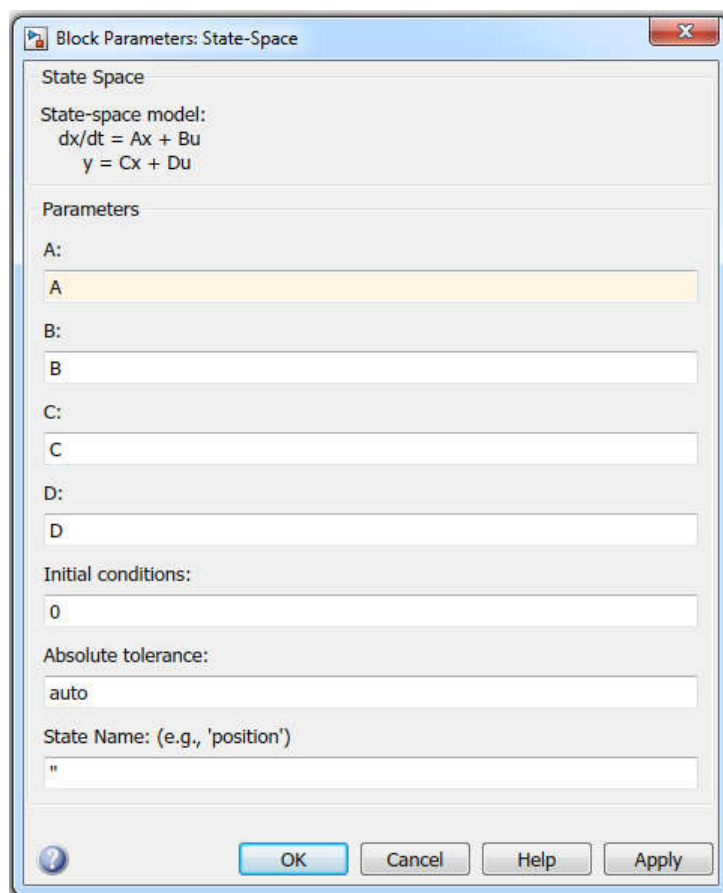


Рисунок 3.8 – Вікно завдання матриць **A**, **B**, **C**, **D** для блоку State-Space

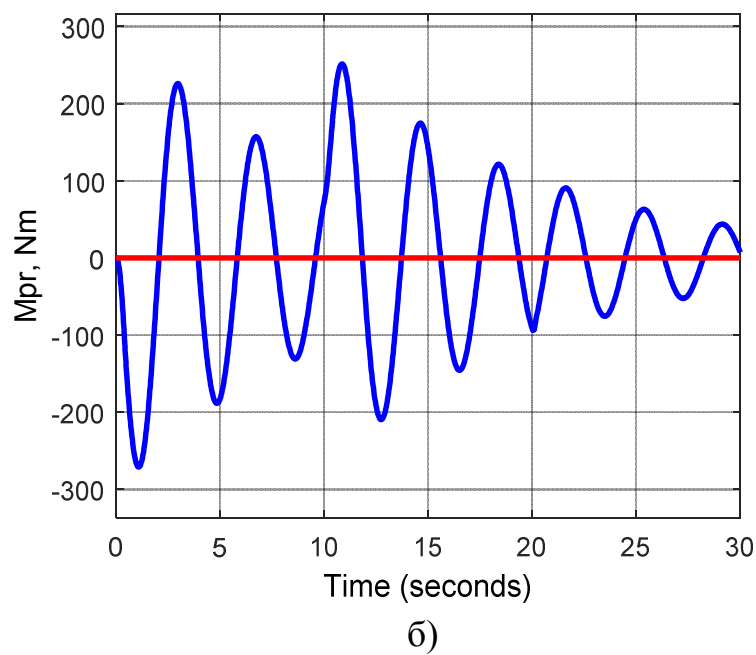
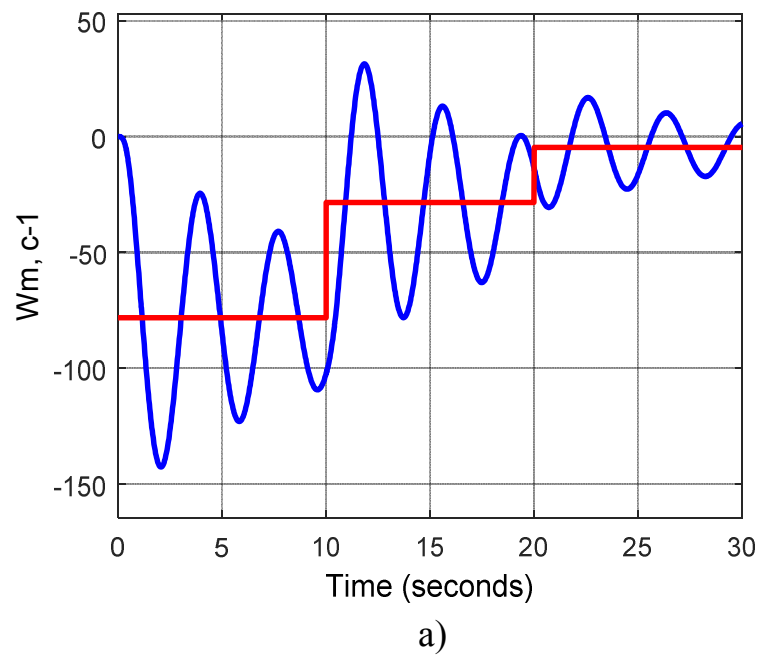


Рисунок 3.9 – Пперехідні процеси швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи за задаючим впливом

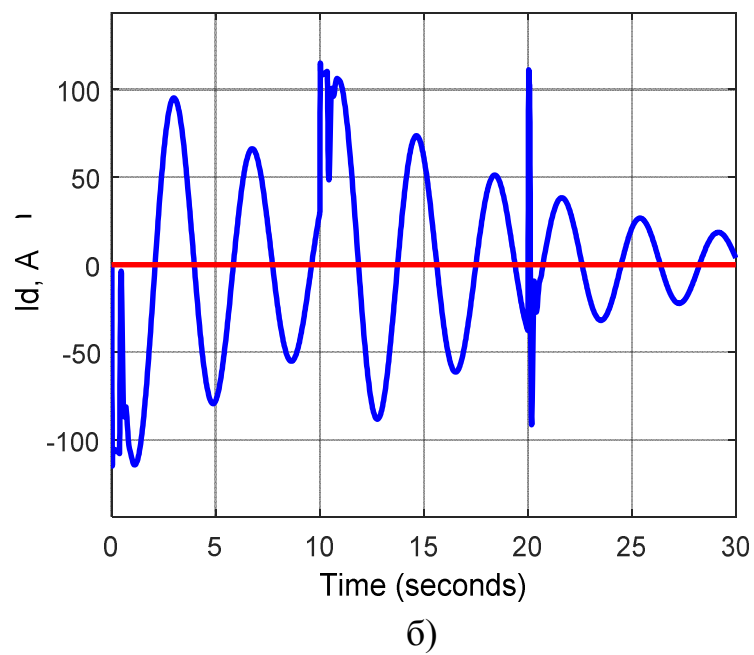
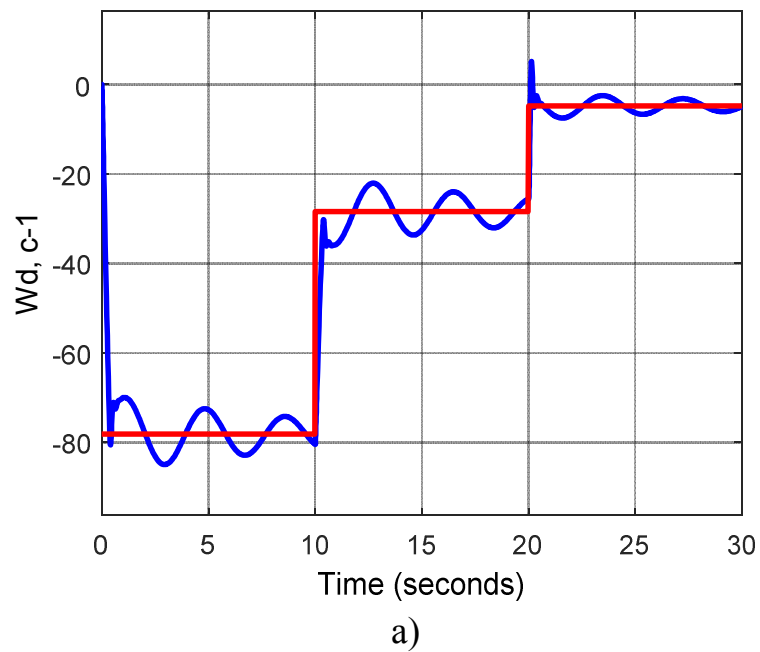
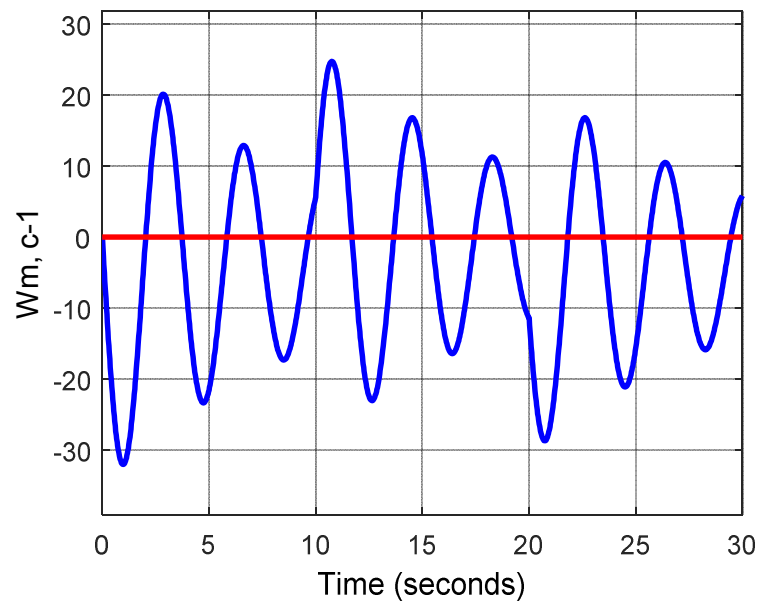
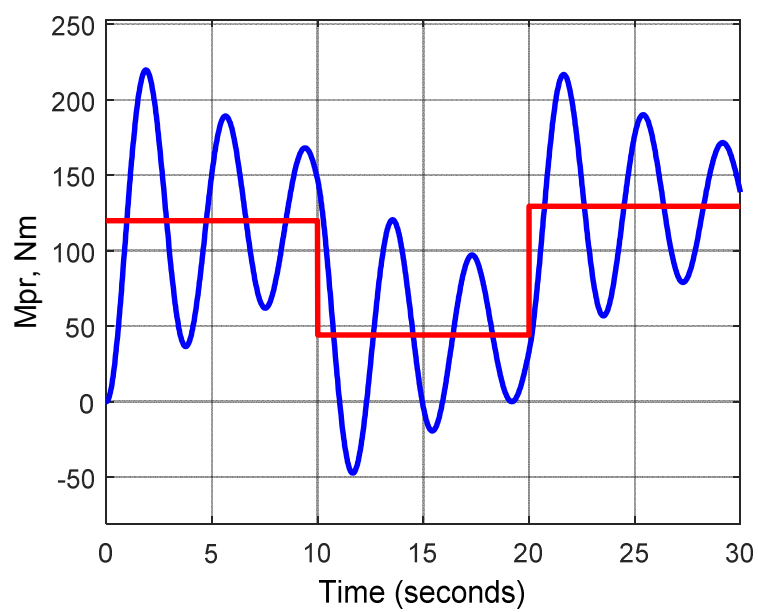


Рисунок 3.10 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи за задаючим впливом



a)



б)

Рисунок 3.11 – Перехідні процеси швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи за збурюючим впливом

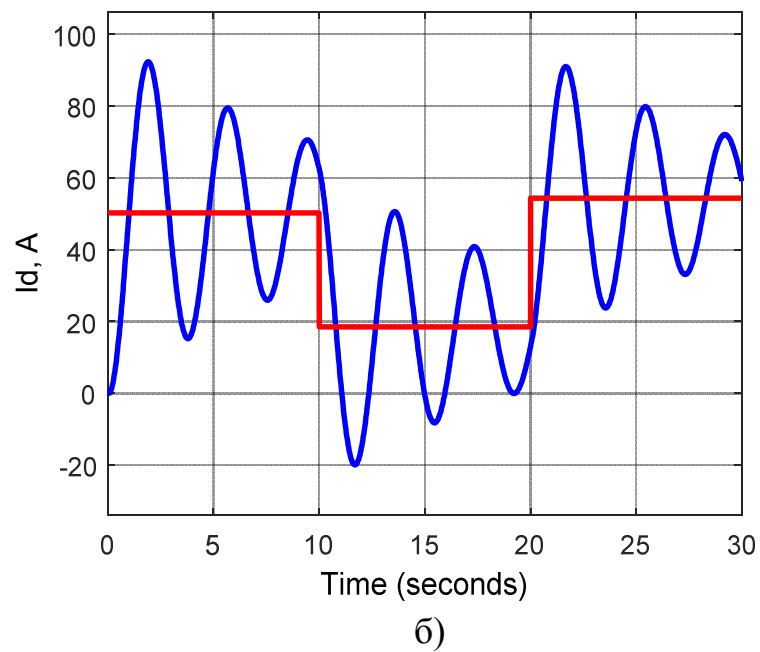
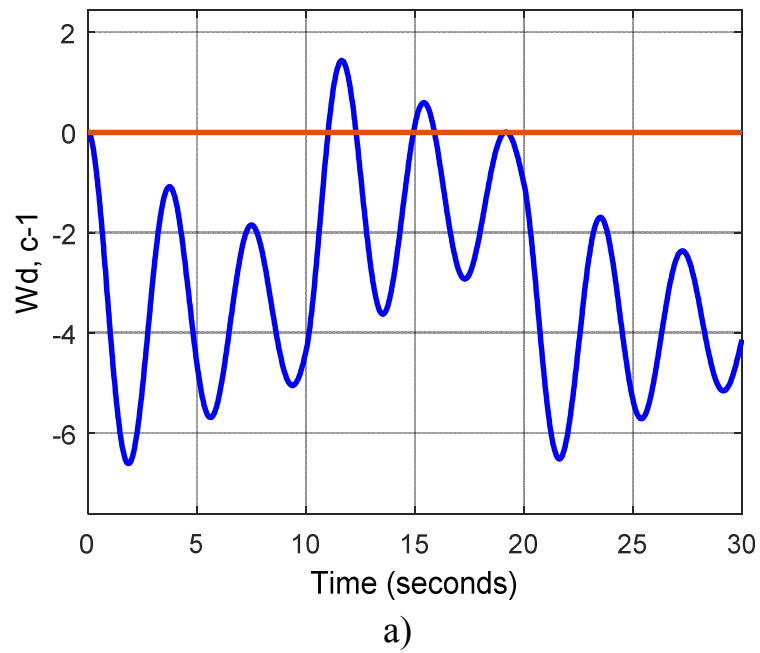


Рисунок 3.12 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи за збурюючим впливом

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено детальний аналіз роботи механізму переміщення візка з урахуванням кінцевої жорсткості механічних зв'язків. Було показано, що представлення механічної частини жорсткою приведеною ланкою дозволяє оцінити середні характеристики руху елементів системи, проте не враховує вплив пружності реальних механічних зв'язків. Через обмежену жорсткість цих зв'язків механічна частина електроприводу виявляє пружні властивості: будь-яка управляюча або збурююча дія, наприклад, момент статичного навантаження, викликає коливання зв'язаних мас. Це призводить до збільшення максимальних навантажень на механічні зв'язки та ускладнює забезпечення точності необхідних переміщень і рухів. Дослідження показало, що кінематична схема механічної частини електроприводу, з урахуванням взаємодії вантажу та візка через підвіс, може розглядатися як двомасова система, що дозволяє більш точно моделювати динаміку процесу та оцінювати вплив коливань на стабільність роботи механізму.

2. Виконано розробку математичної моделі двомасової системи. Було отримано рівняння стану системи та їх представлення у векторно-матричній формі, визначено елементи вектора стану, тобто змінні стану системи, а також вектор зовнішніх впливів. Крім того, визначені матриці стану AAA та управління BBB , що забезпечують повний опис динаміки системи. Розроблено алгоритмічну схему двомасової системи та її спрощену форму, у якій контур струму оптимізований за модульним критерієм. Такий підхід дозволяє спростити подальший аналіз системи та забезпечує основу для синтезу коригуючих пристроїв для регулювання струму та швидкості, а також для подальшого моделювання перехідних процесів з урахуванням пружності механічних зв'язків.

3. Проведено моделювання динаміки двомасової системи на ЕОМ з використанням пакету прикладних програм MATLAB. У середовищі SIMULINK зроблено детальну схему моделі системи, відповідну алгоритмічній схемі та рівнянням стану. Для тестування на вхід системи подавався ступінчастий сигнал

із випадковою амплітудою в межах від +24 В до –24 В, що відповідає зміні швидкості в межах номінальних значень електроприводу. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи мають коливальний характер, що відображає слабо затухаючі коливання механізму. Це свідчить про необхідність врахування пружності механічних зв'язків при проектуванні системи керування та можливу потребу введення додаткових демпфуючих елементів або коригувальних пристроїв для підвищення стабільності та точності роботи електроприводу.

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).

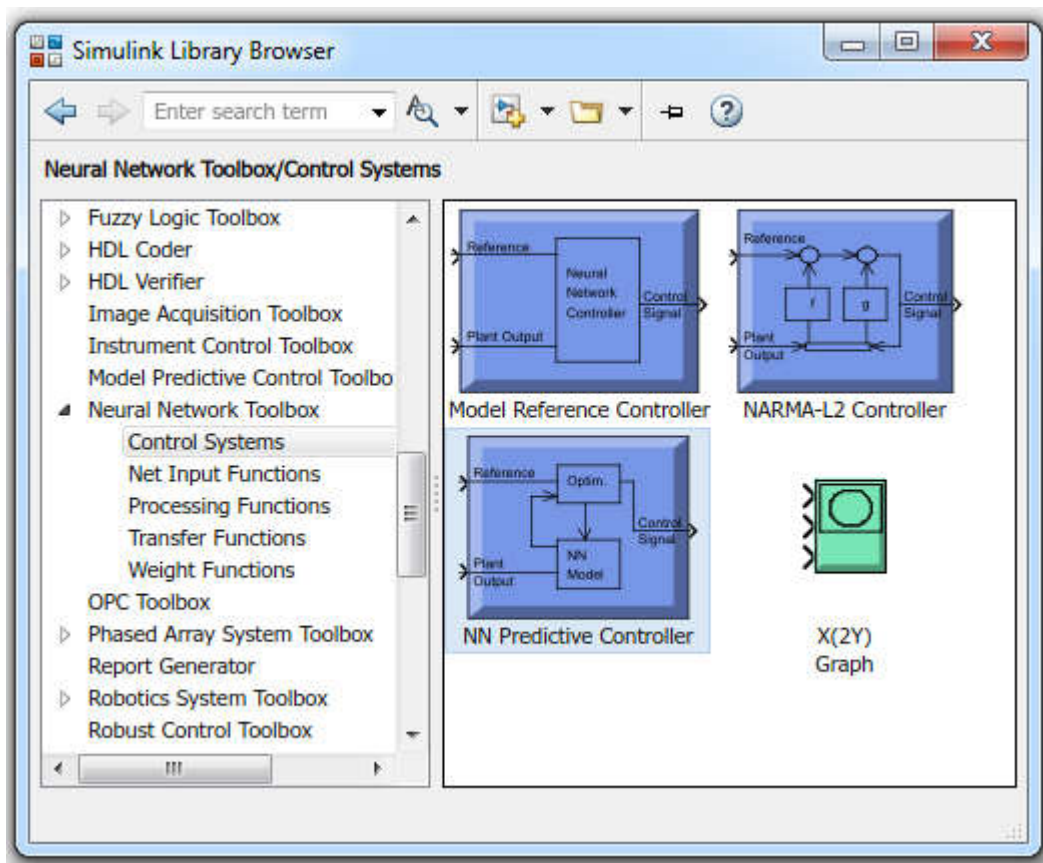


Рисунок 4.1 – Набір блоків Neural Network Blocksets підсистеми Control Systems середовища SIMULINK

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 керува

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.

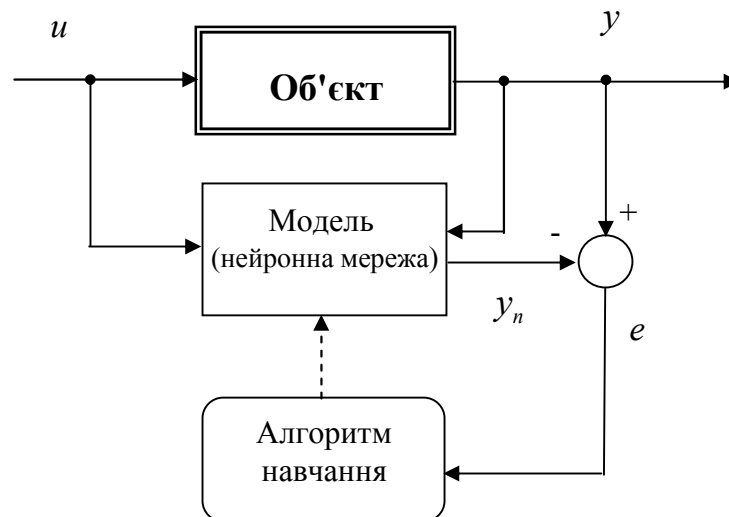


Рисунок 4.2 – Структурна схема підсистеми ідентифікації

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**.

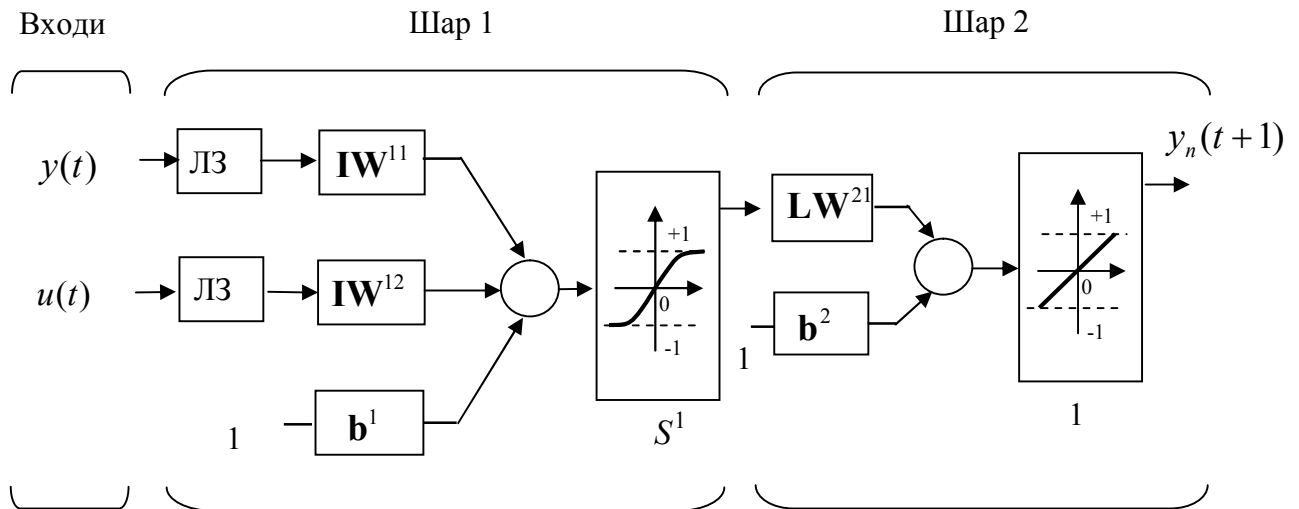


Рисунок 4.3 – Структура нейронної мережі, що моделює об’єкт керування

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.

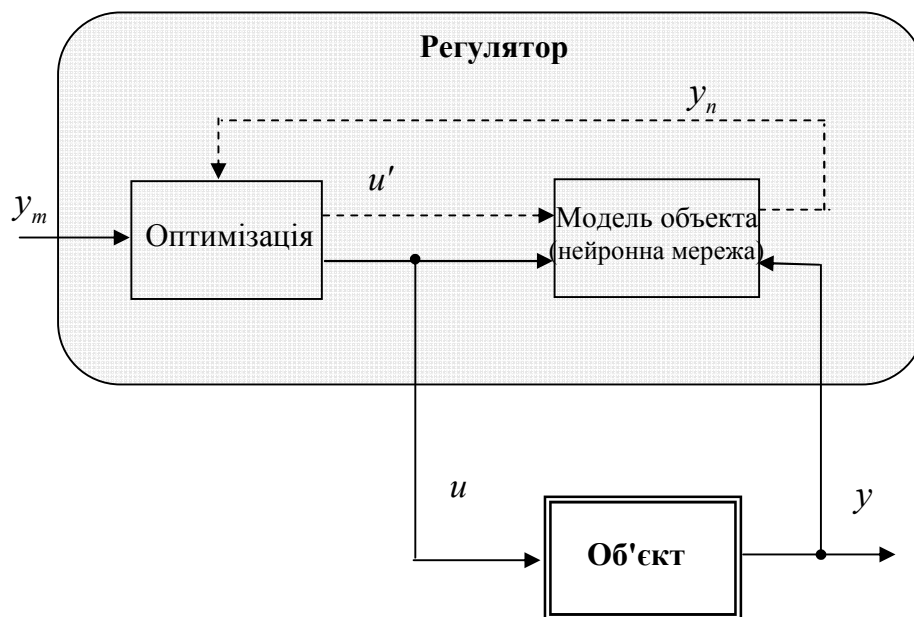


Рисунок 4.4 – Узагальнена схема керування з використанням прогнозуючого регулятора

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:

5. 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.

6. 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.

7. 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.

8. 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.

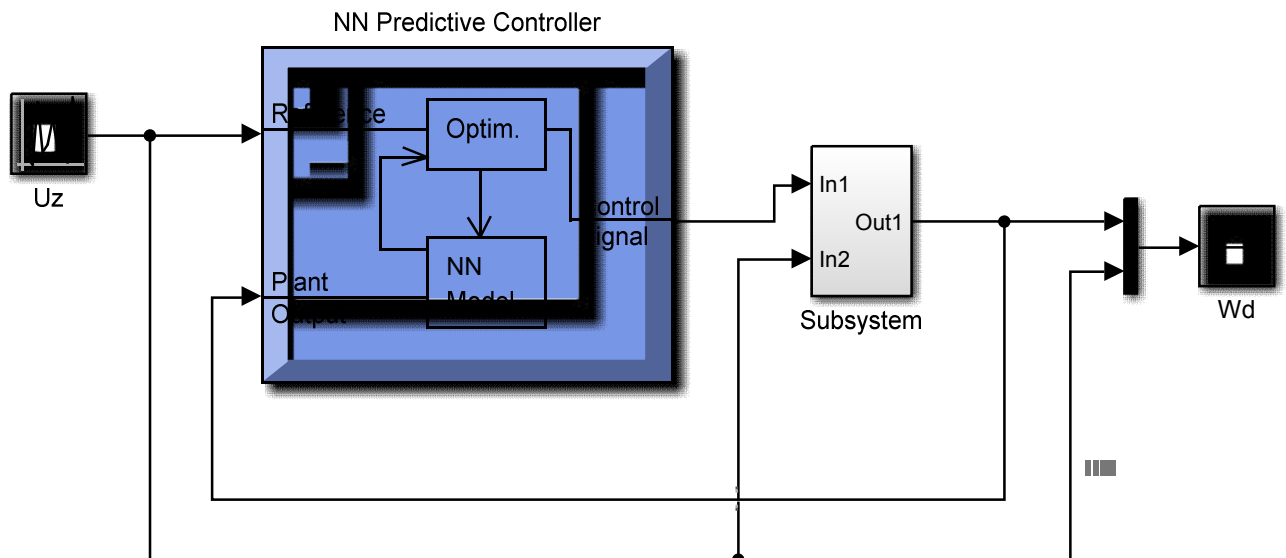


Рисунок 4.5 – Структурна модель системи керування з нейрорегулятором NN Predictive Controller

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

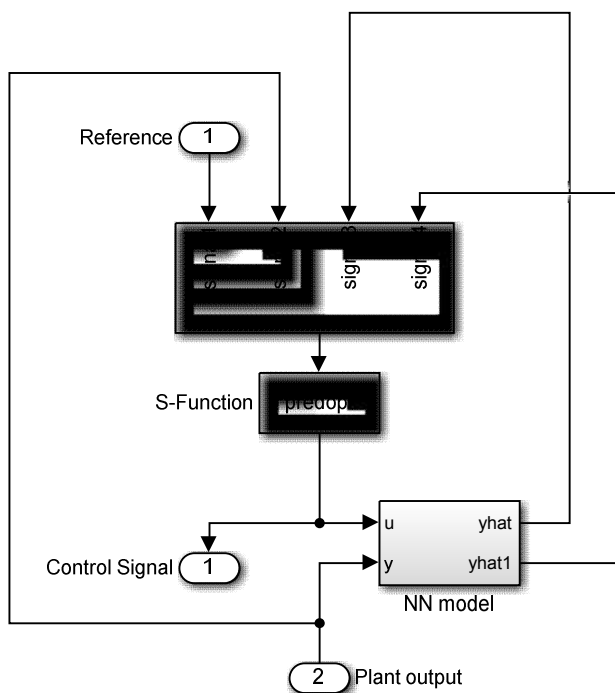


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

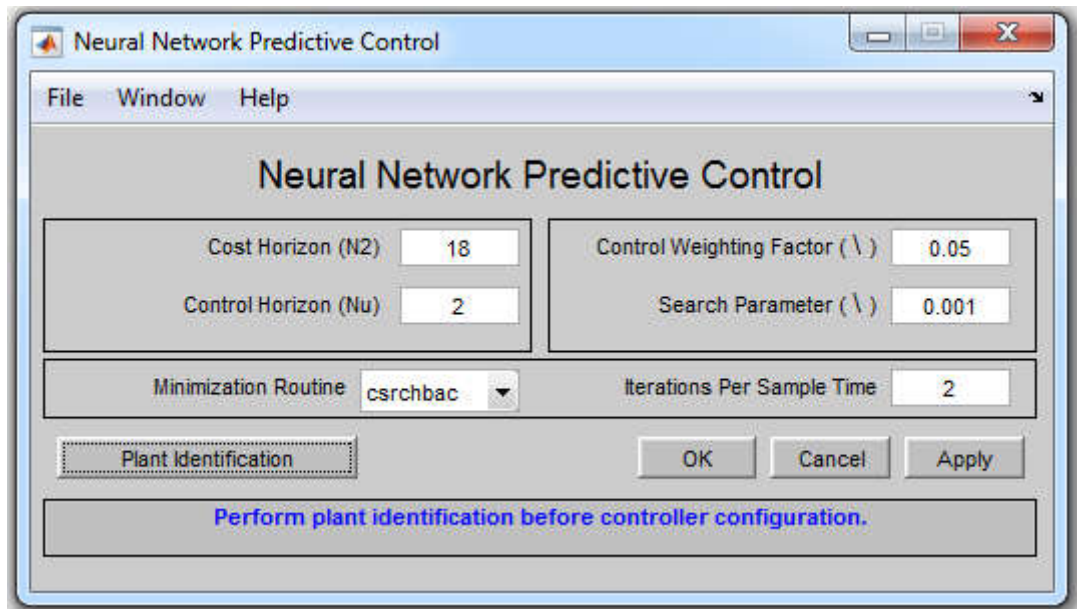


Рисунок 4.7 – Інтерфейс налаштування параметрів регулятора

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.

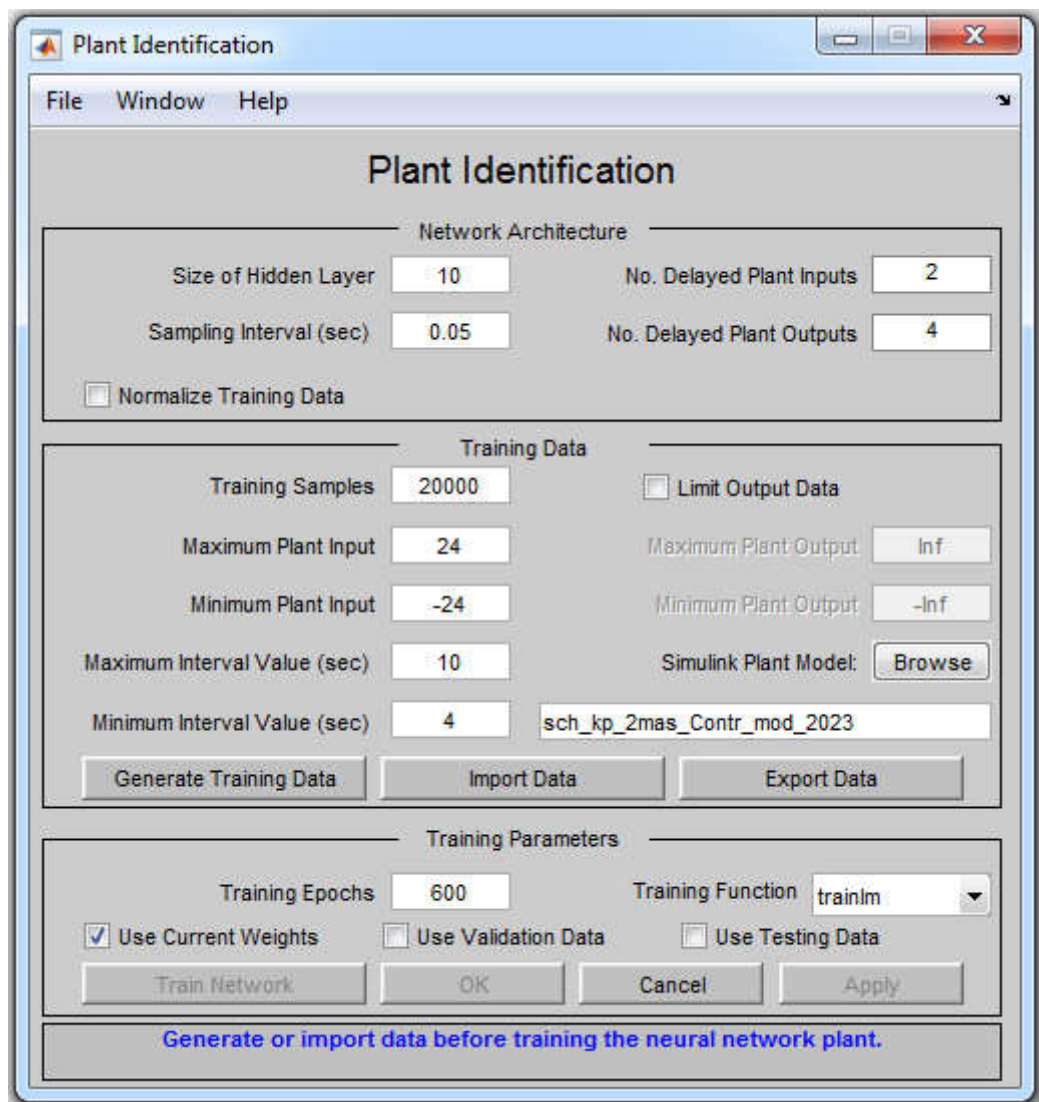


Рисунок 4.8 – Вікно процедури ідентифікації об’єкта керування

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 нейронмережі:

- 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 вказує 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.

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.

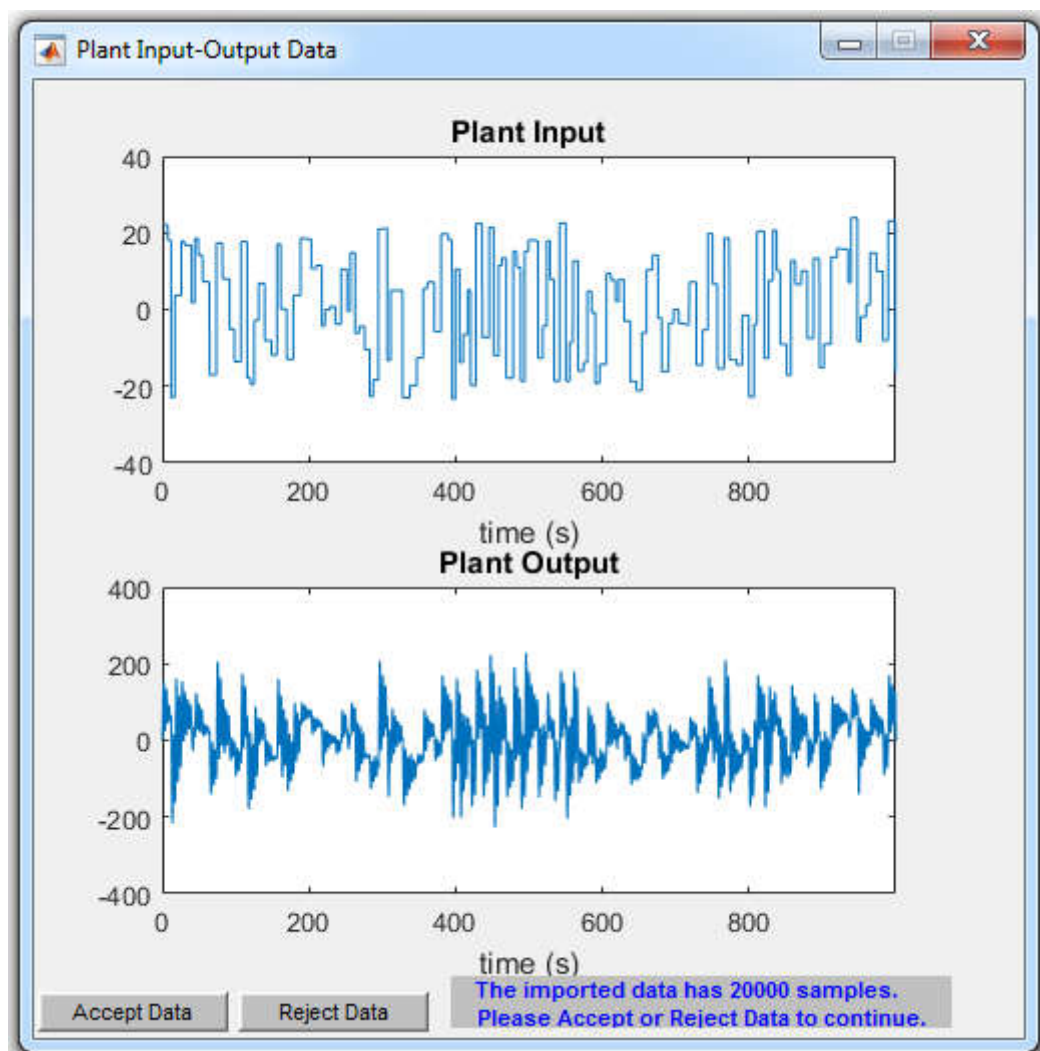


Рисунок 4.10 – Графіки вхідного та вихідного сигналів під час генерації навчальної послідовності

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 Generate Data**, що надає можливість видалити раніше згенеровані навчальні дані.

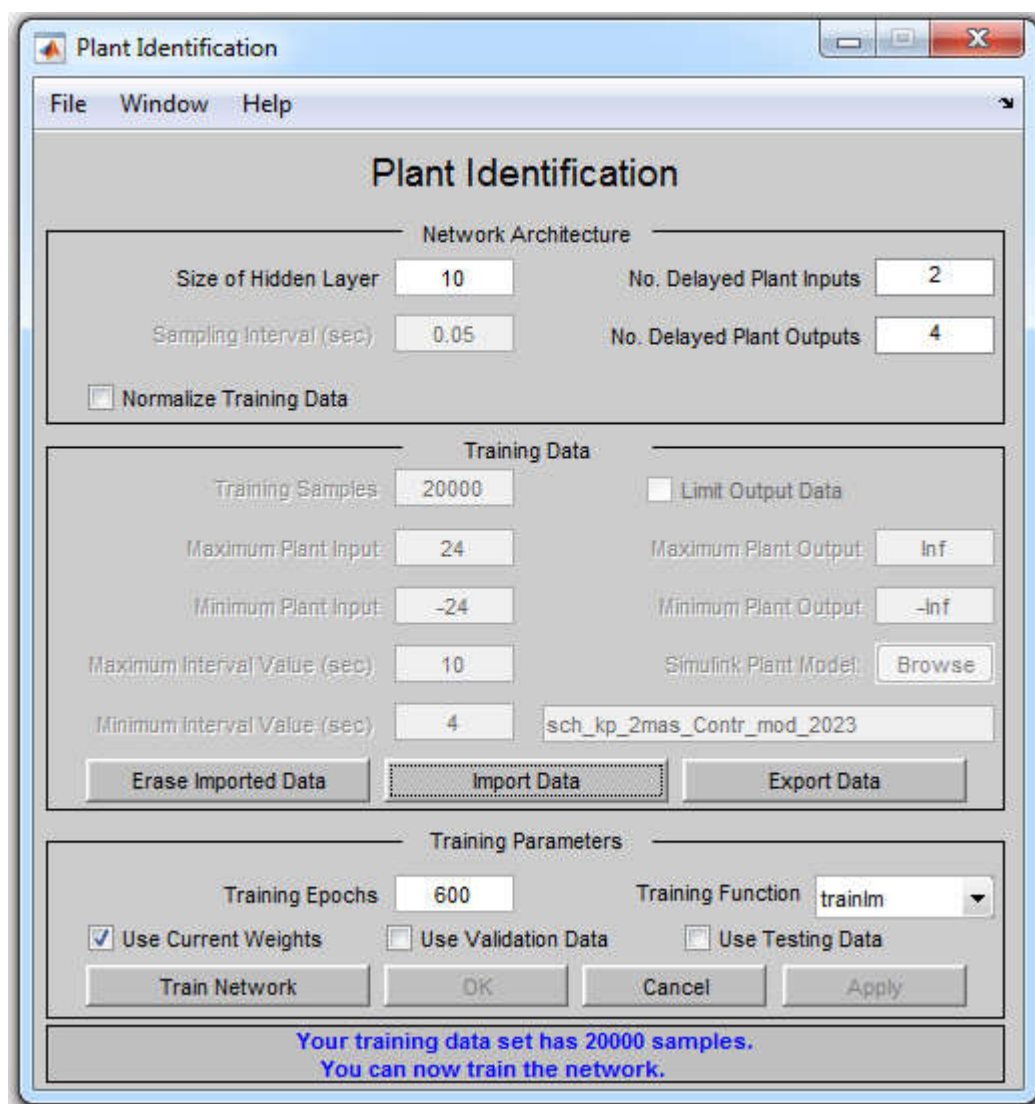


Рисунок 4.11 – Вікно Plant Identification після генерації навчальної послідовності

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).

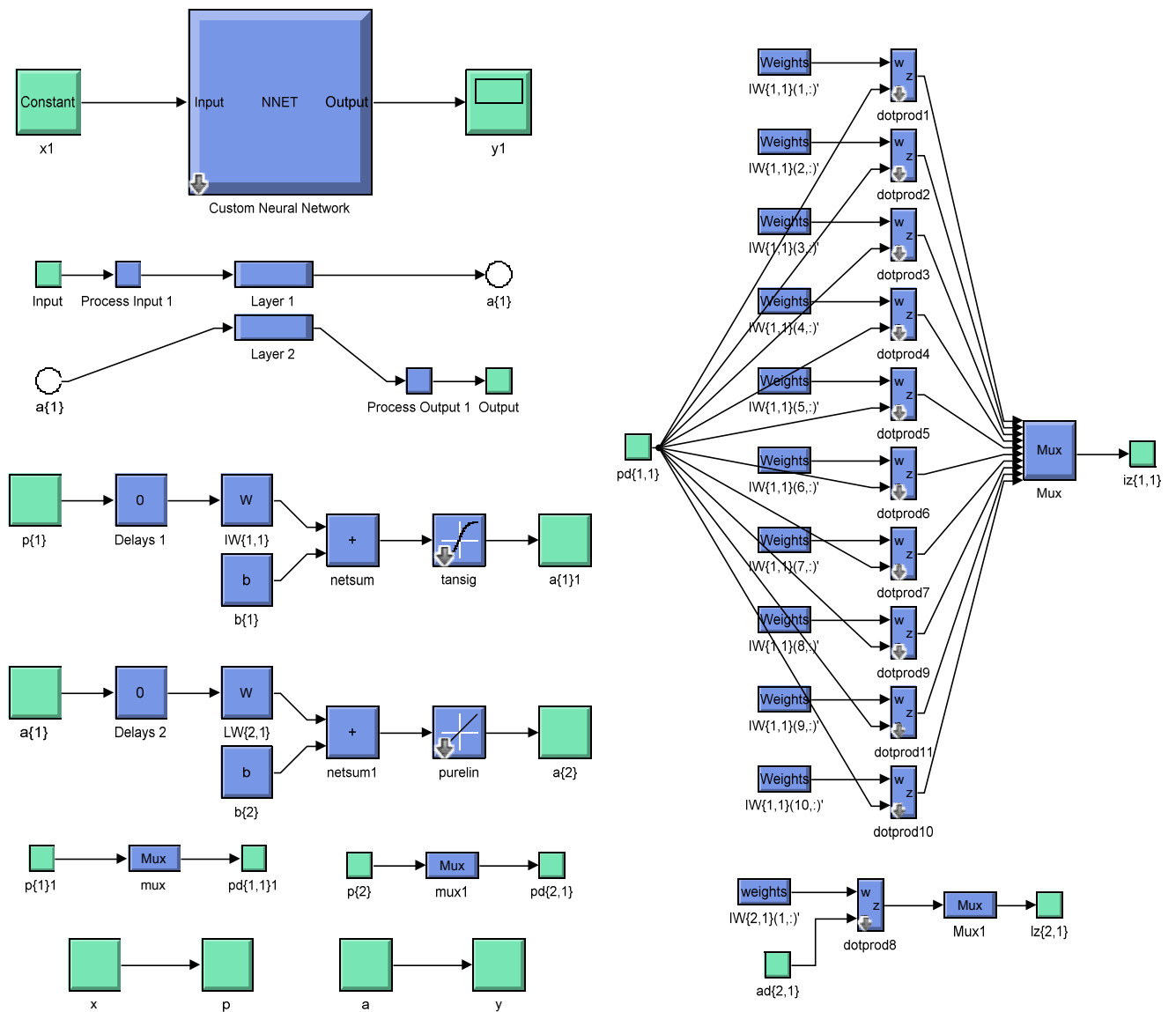


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

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.

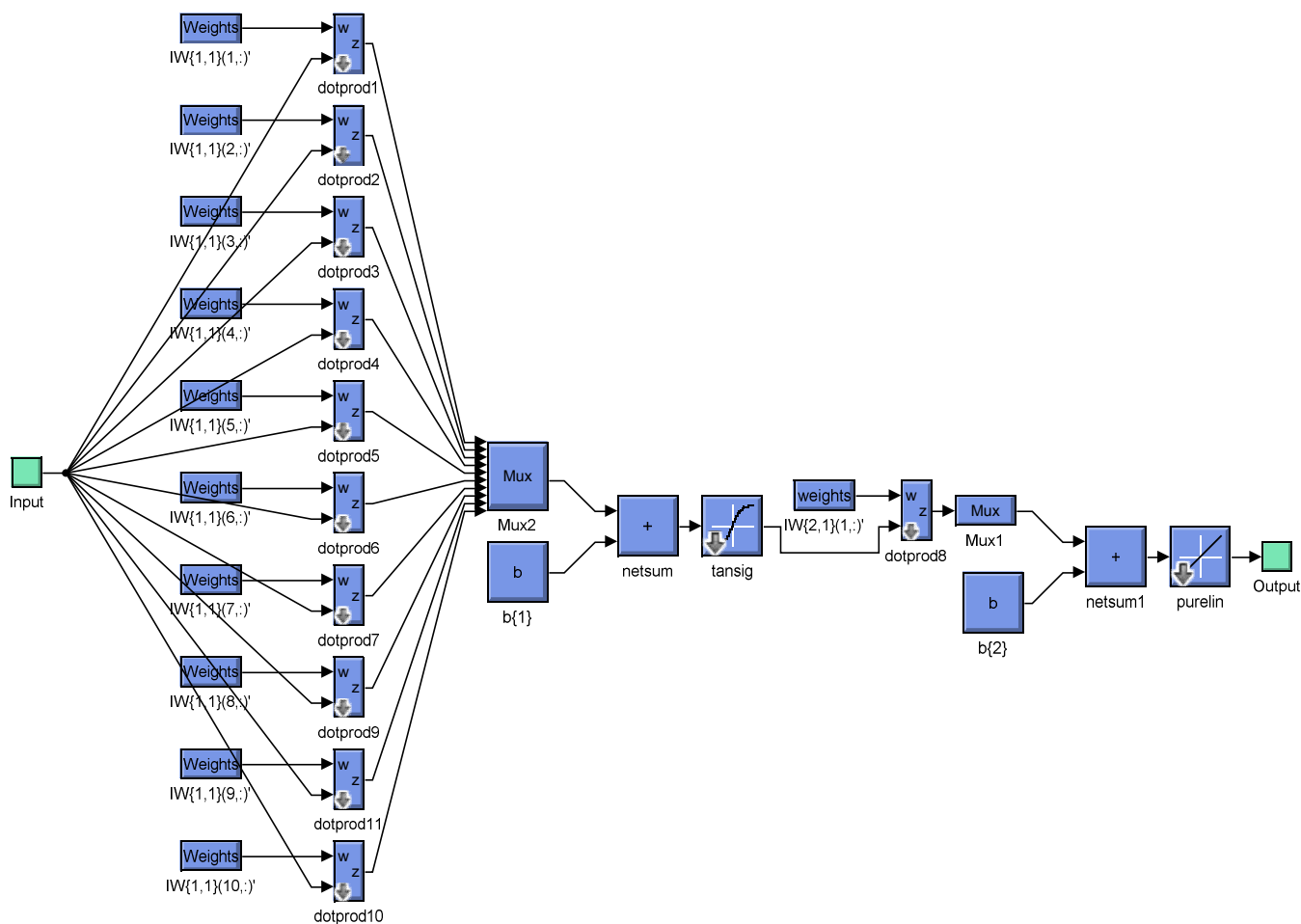


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

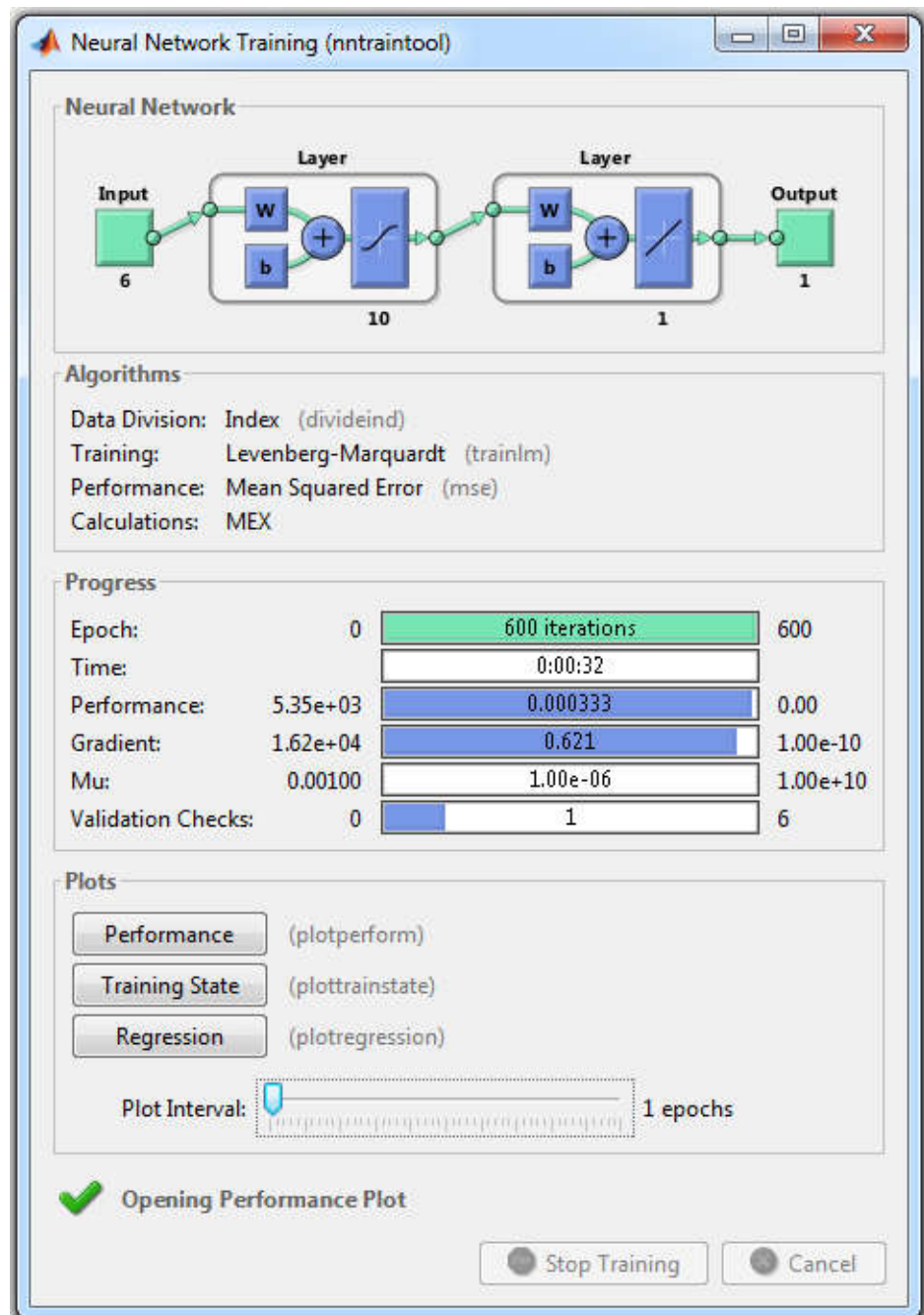


Рисунок 4.14 - Головне вікно для відображення процесу та результатів навчання нейронної мережі

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.

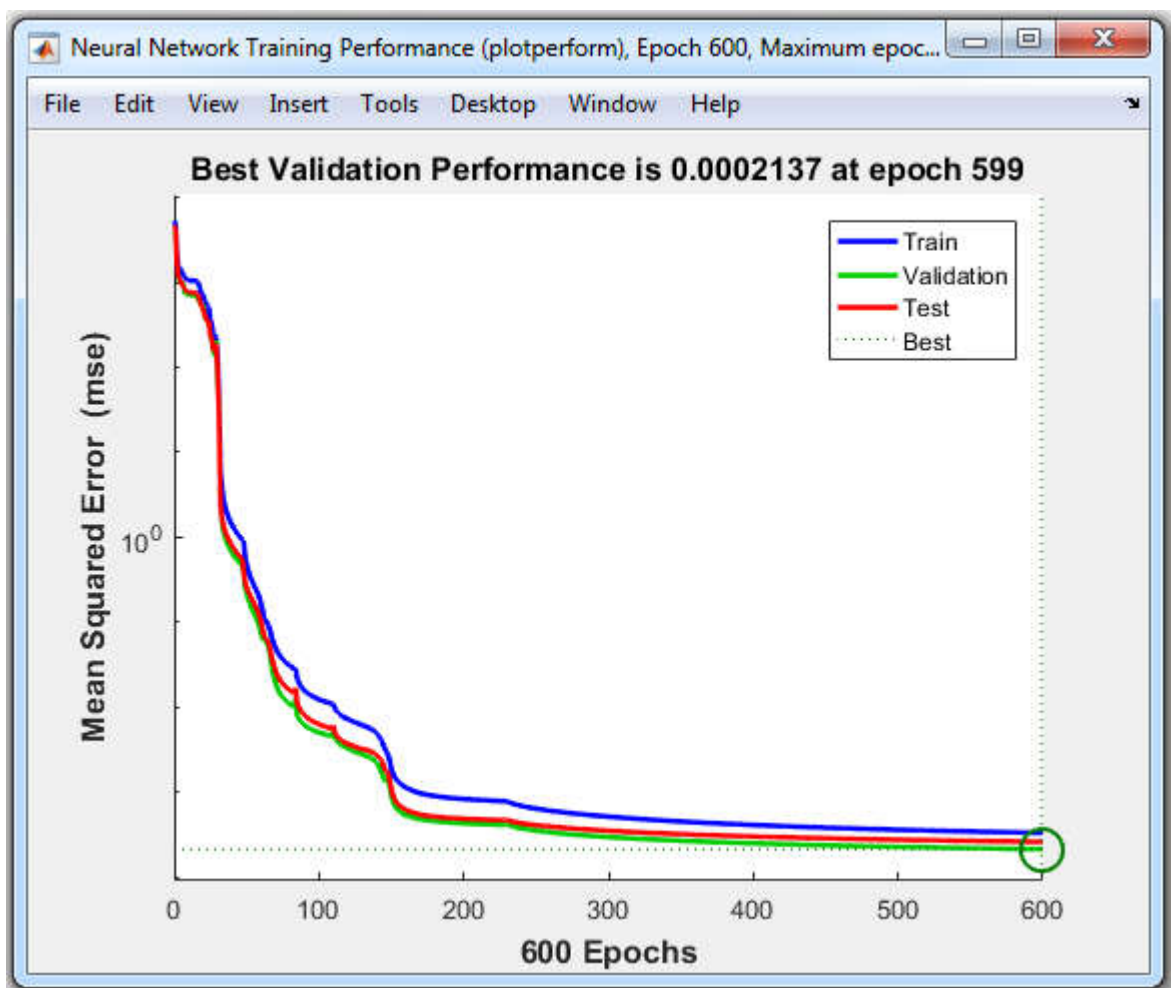


Рисунок 4.15 – Вікно контролю процесу навчання

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.

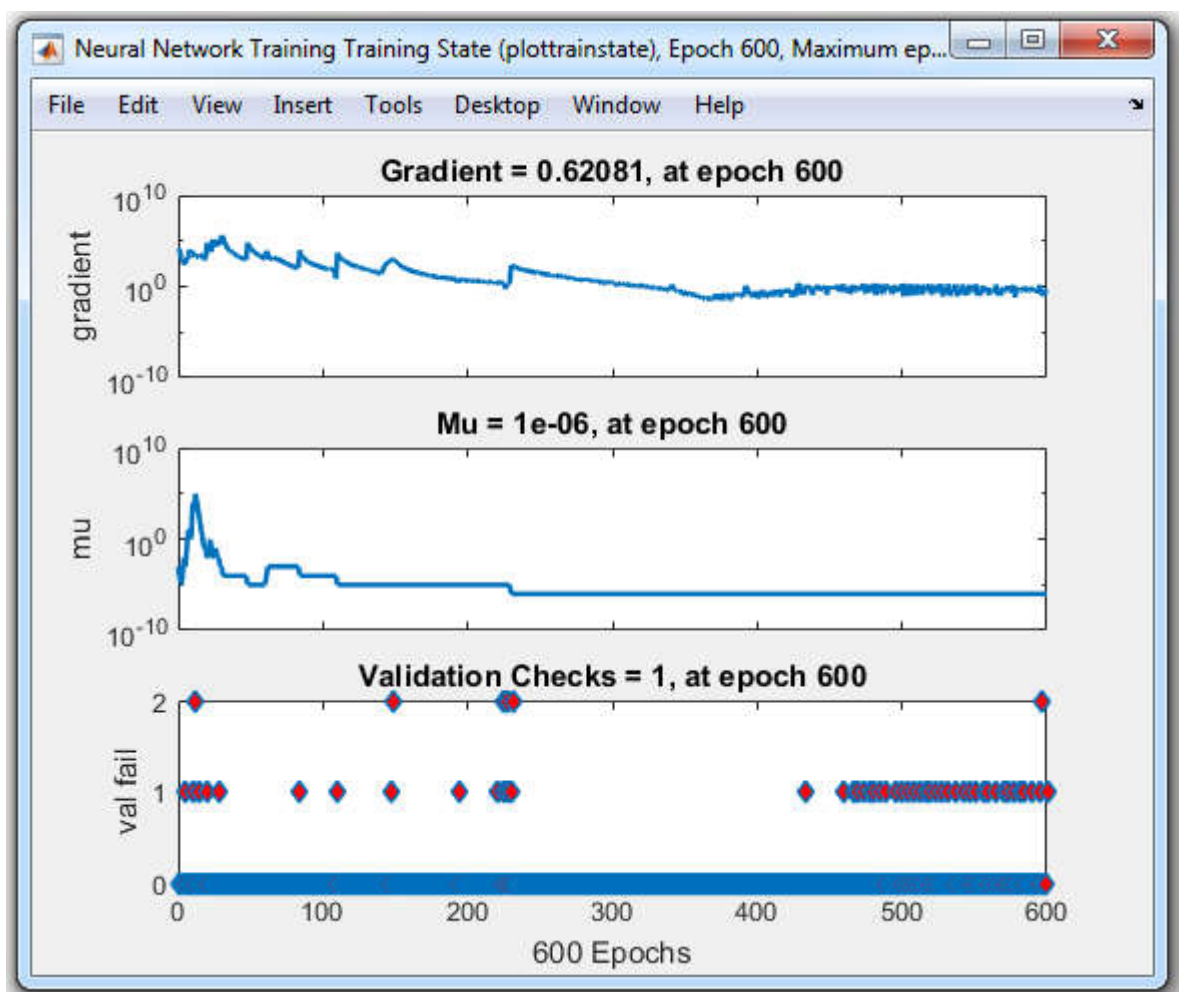


Рисунок 4.16 - Вікно Neural Network Training State

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.

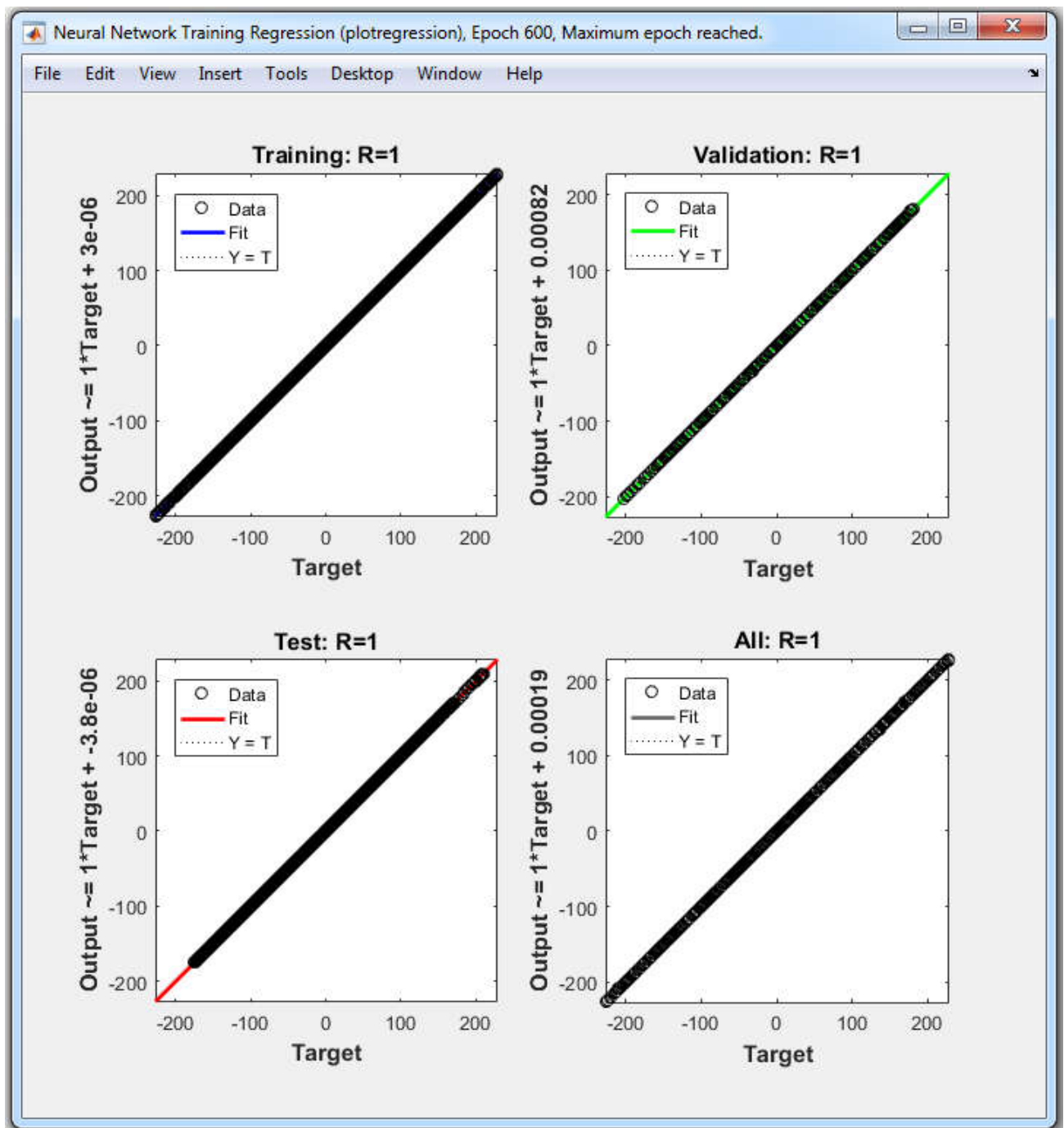


Рисунок 4.17 - Вікно Neural Network Training Regression

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, перевірочних 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).

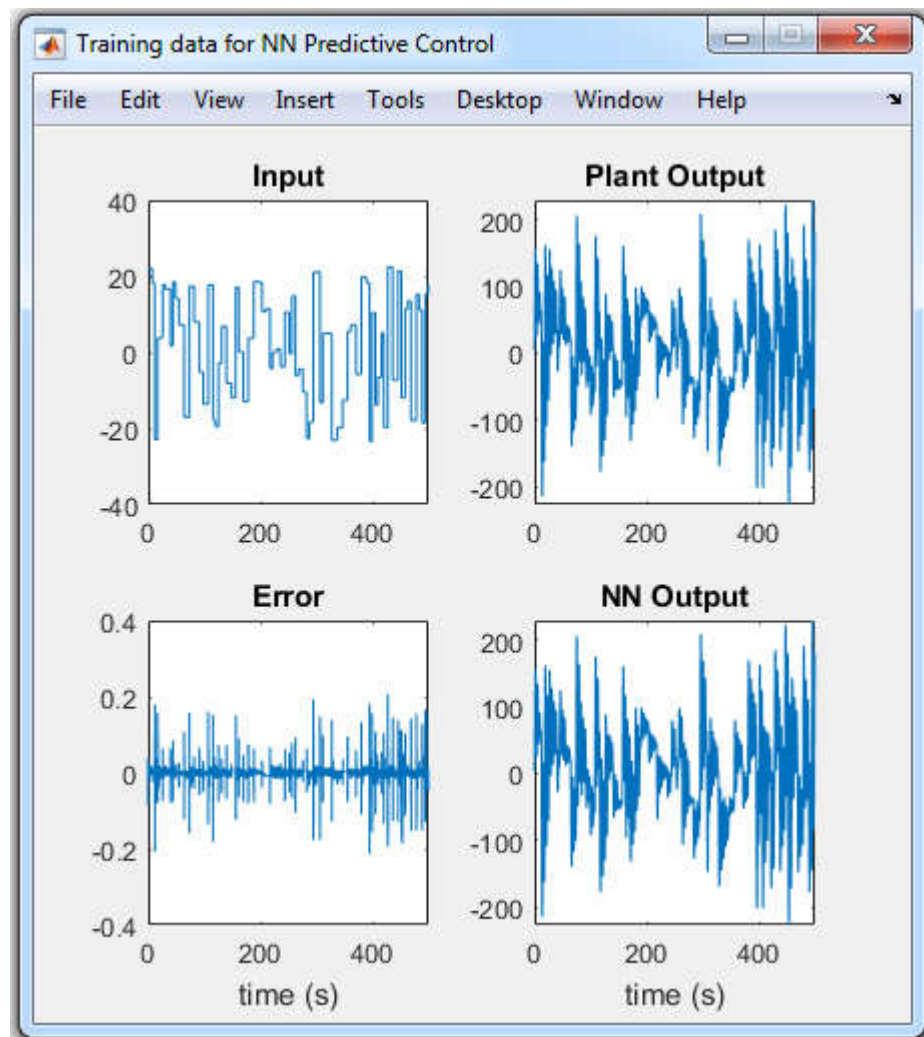


Рисунок 4.18 – Графічне представлення результатів навчання

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.

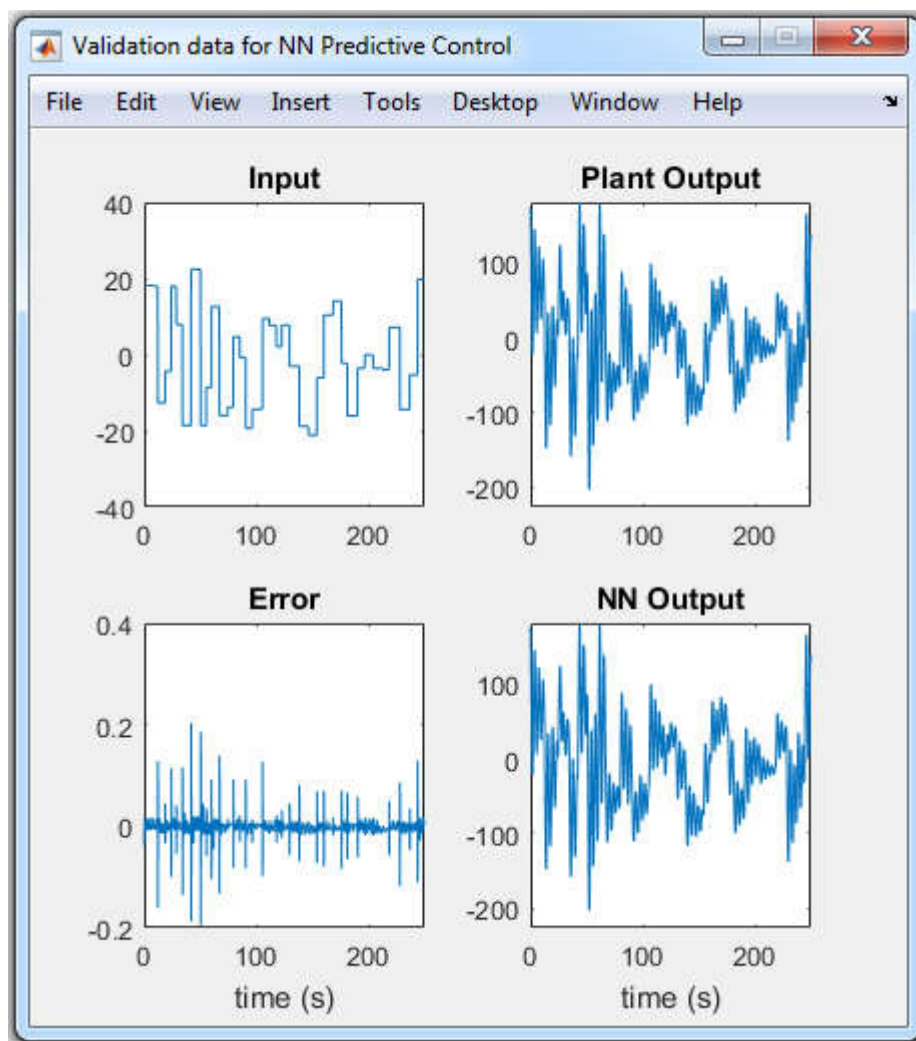


Рисунок 4.19 – Візуалізація результатів перевірки нейронної мережі на контрольній множині

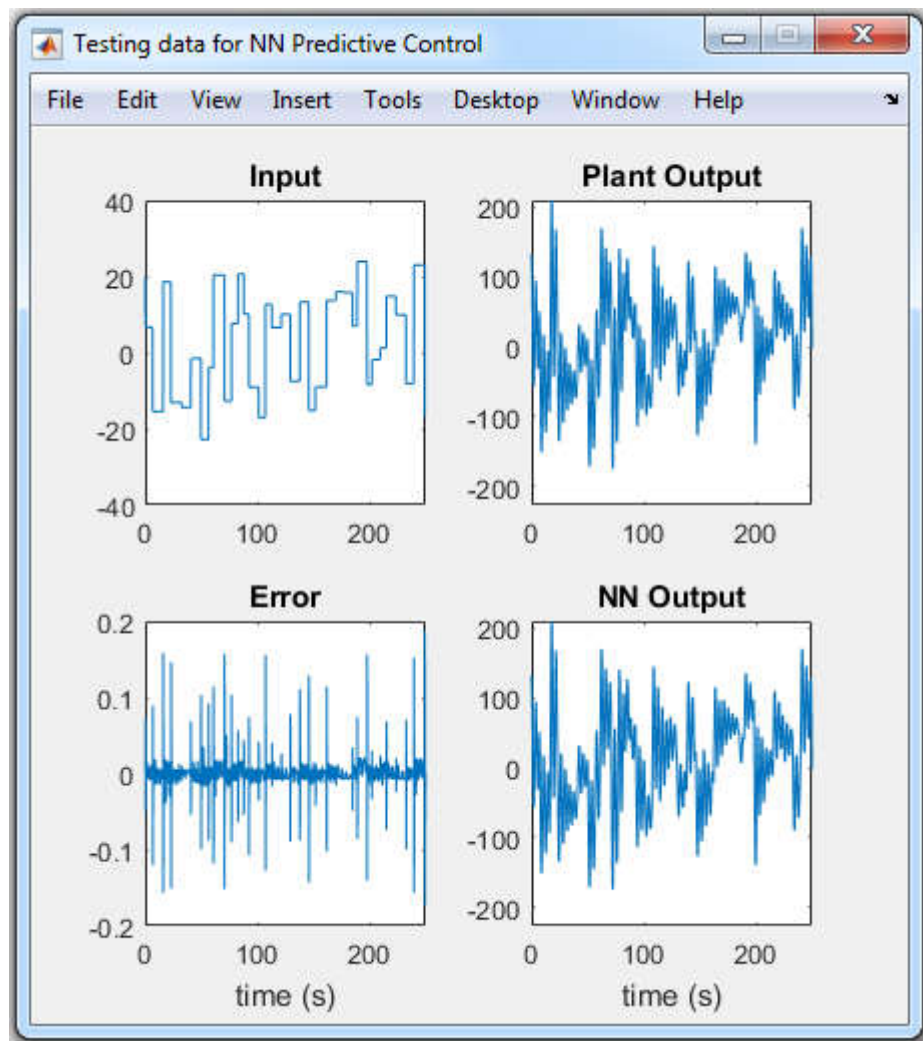


Рисунок 4.20 – Візуалізація результатів перевірки нейронної мережі на тестовій множині

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 **Nnident** 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.

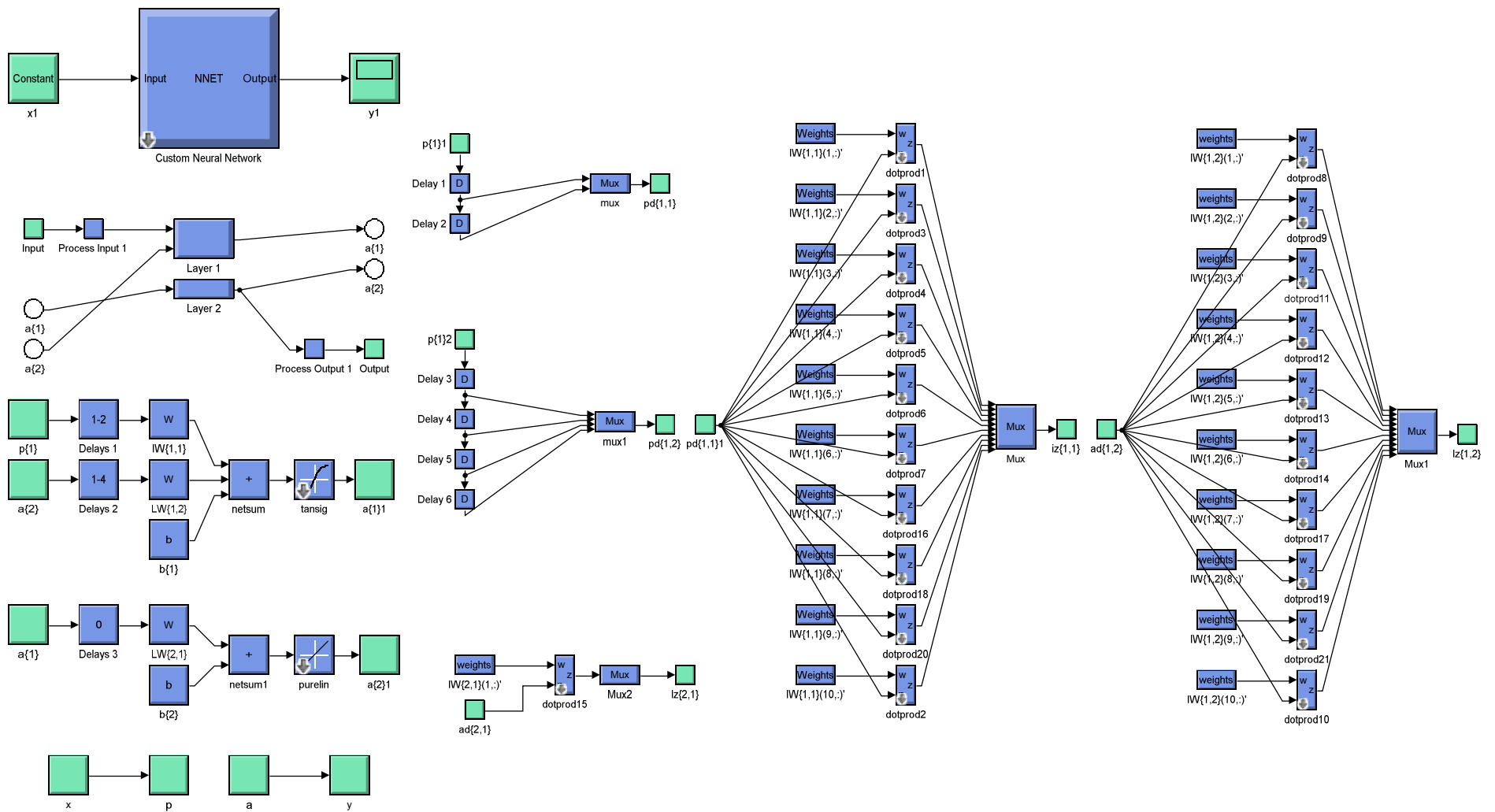


Рисунок 4.21 – Моделі елементів динамічної мережі з блоками затримок у Simulink

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.

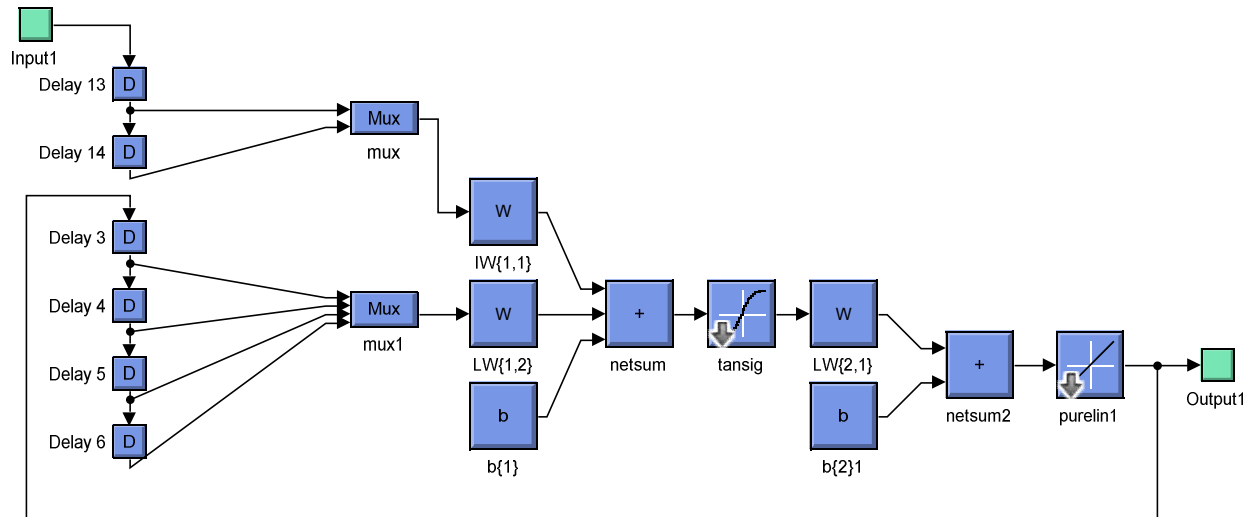


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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.

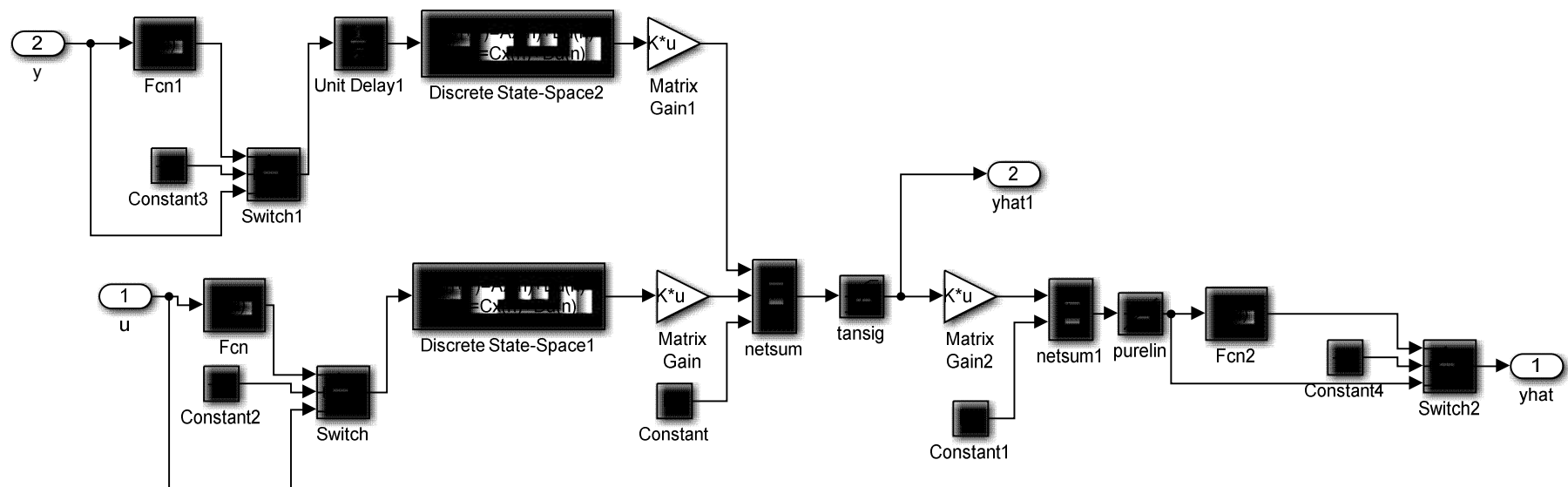


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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.

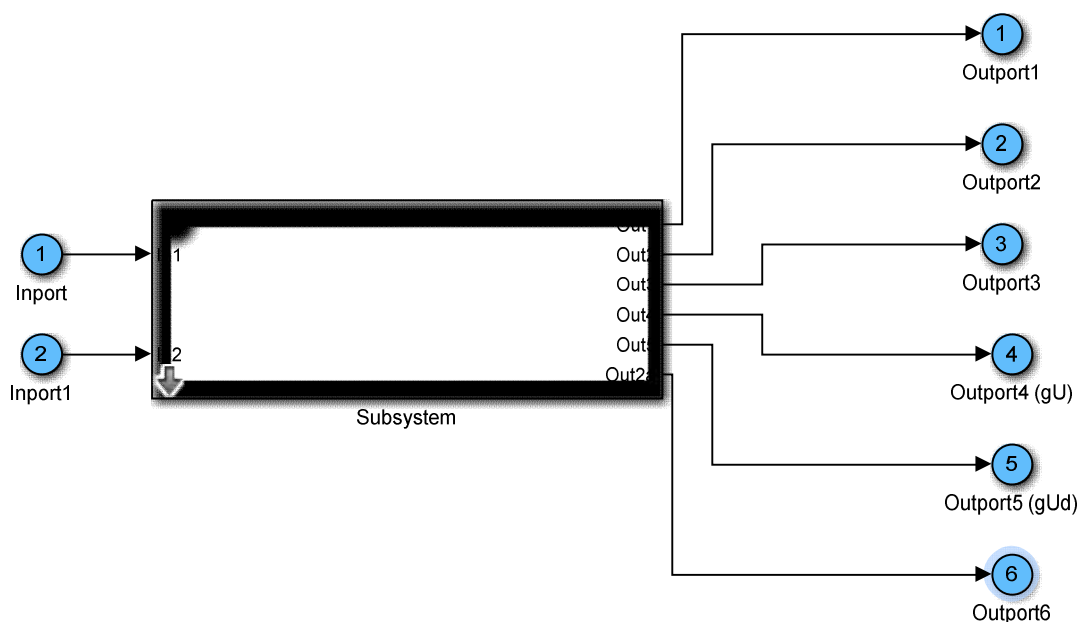


Рисунок 4.24 – Схема ptest3sim2, що використовується для прогнозування процесу функцією predopt

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.

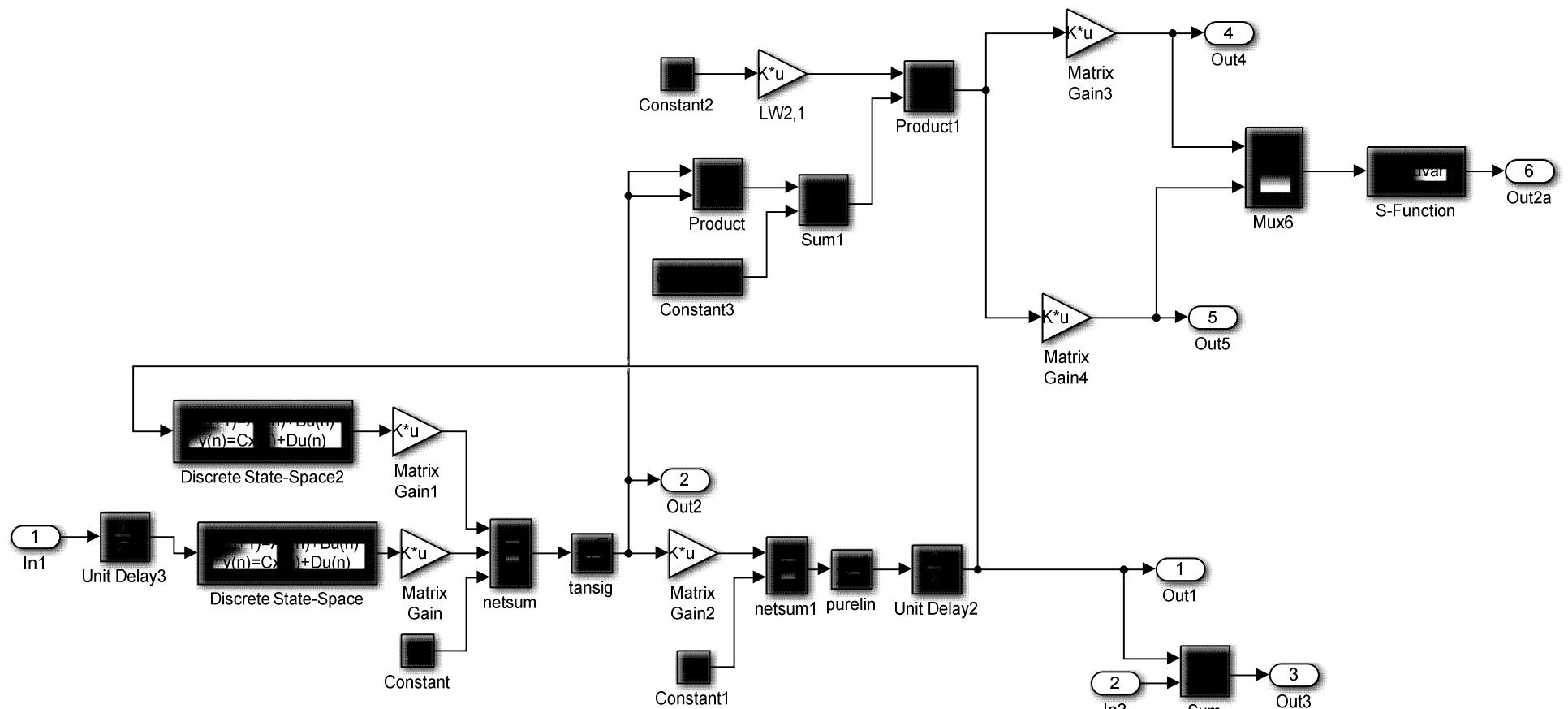


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

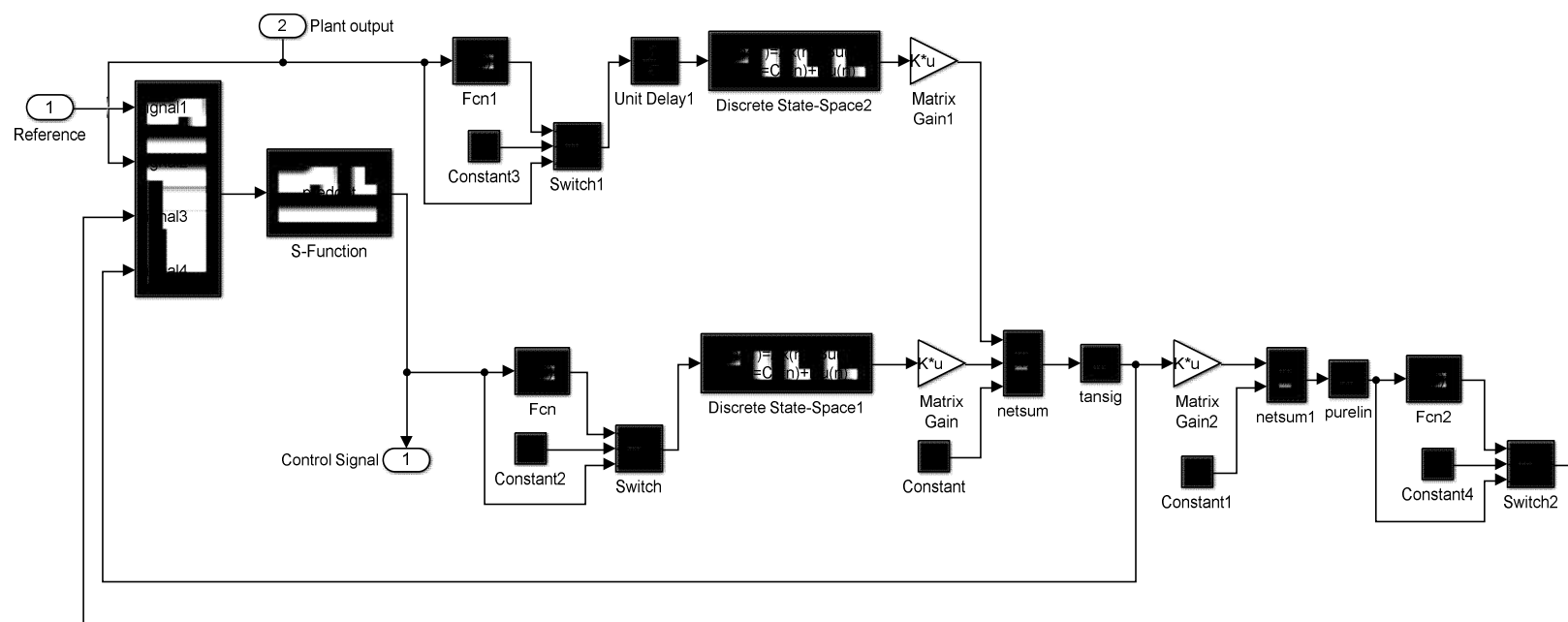


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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 одновимірного 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 проектуванні нейрорегулятора відбувається варіювання параметрів Const Horizon (N_2), Control Horizon (N_u) та Control Weighting Factor (ρ) (за умови, що нижня межа Const Horizon залишається зафіксованою і дорівнює 1), а також параметра одновимірного пошуку Search Parameter α , який визначає поріг допустимого зменшення показника якості, та кількість ітерацій оптимізації на один крок дискретності. Додатково обирається метод проведення одновимірного пошуку.

Дослідження показали, що параметри Const Horizon, Control Weighting Factor та обрана процедура пошуку мають незначний вплив на кінцеві результати синтезу регулятора. Тому для даної задачі були обрані значення: $N_u=2$, $\rho=0,05$, $\alpha=0,001$, а метод одновимірного пошуку – csgchbas.

Водночас значення Control Horizon (N_u) і кількість ітерацій на такт дискретності γ значно впливають на продуктивність контролера. Збільшення цих параметрів дозволяє підвищити точність керування, проте істотно зростає обчислювальне навантаження на кожному кроці. Для розв'язуваної задачі оптимальні значення знаходяться в межах $N_2 = 20 \div 30$, $\gamma = 2 \div 4$

При ідентифікації об'єкта керування критично важливим є вибір кількості нейронів у прихованому шарі N . Недостатня кількість нейронів унеможливило виконання поставлених завдань, тоді як надмірна – призводить до перенавчання та збільшення часу обчислень. Для розглядуваної системи оптимальне значення

$N = 7 \div 12$, при цьому похибка на навчальній, контрольній та тестовій вибірках залишалася в межах $10^{-4} \div 10^{-6}$

Якість навчання мережі безпосередньо залежить від довжини навчальної вибірки N_b та дискретного кроку часу Δt , який визначає інтервал між двома послідовними точками збору даних. Оптимальні значення для даної системи: $N_b = 10000 \div 20000$, $\Delta t = 0,05 \div 0,1$ с. Збільшення Δt зменшує точність обчислень і скорочує різницю між помилкою на навчальному наборі та помилками на контрольній і тестовій множинах. Зменшення Δt потребує подовження навчальної вибірки N_b , що суттєво збільшує час навчання без значного зниження похибки.

Для формування репрезентативної вибірки важливо правильно встановити мінімальні та максимальні межі інтервалу ідентифікації, що залежать від динамічних характеристик об'єкта Subsystem. У даній роботі вони обрані як $t_{\min} = 4 \div 8$ с, $t_{\max} = 10 \div 20$ с.

При побудові нейромережевої моделі задається кількість затримок на вході z_1 та виході z_2 моделі. Найкращі результати були досягнуті при $z_1 = 2$, $z_2 = 2 \div 5$.

Результат навчання також залежить від початкових значень вагових коефіцієнтів w_{ij} і кількості навчальних циклів $N_{\text{ц}}$. Для досягнення глобального мінімуму навчання рекомендується багаторазово повторювати процес з різними початковими значеннями ваг w_{ij} та різними конфігураціями $N_{\text{ц}}$. У даній роботі для кожної конфігурації мережі використовувалася кілька сотень початкових точок, а кількість циклів, після якої похибка переставала зменшуватись, становила $300 \div 900$.

Для навчання мережі використано алгоритм Levenberg-Marquardt (trainlm), який забезпечує ефективну оптимізацію вагових коефіцієнтів і швидке зближення до мінімуму функції втрат.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором **NN Predictive Controller**

Побудова графіків перехідних процесів двомасової системи, керованої нейрорегулятором **NN Predictive Controller**, виконувалась з використанням моделей, створених в середовищі Simulink, що представлені на рис. 4.5 та 4.27.

На рис. 4.28–4.31 наведено графіки перехідних процесів основних змінних стану синтезованої системи. Аналіз отриманих результатів показує, що система демонструє задовільну реакцію на ступінчасті впливи з випадковими амплітудами. Перехідні процеси мають коливальний характер із невеликим перерегулюванням, що свідчить про стабільність та адекватність регулятора.

Таким чином, синтезований нейромережевий регулятор з прогнозом **NN Predictive Controller** може бути ефективно використаний для керування двомасовою електромеханічною системою, розглянутою в даній роботі.

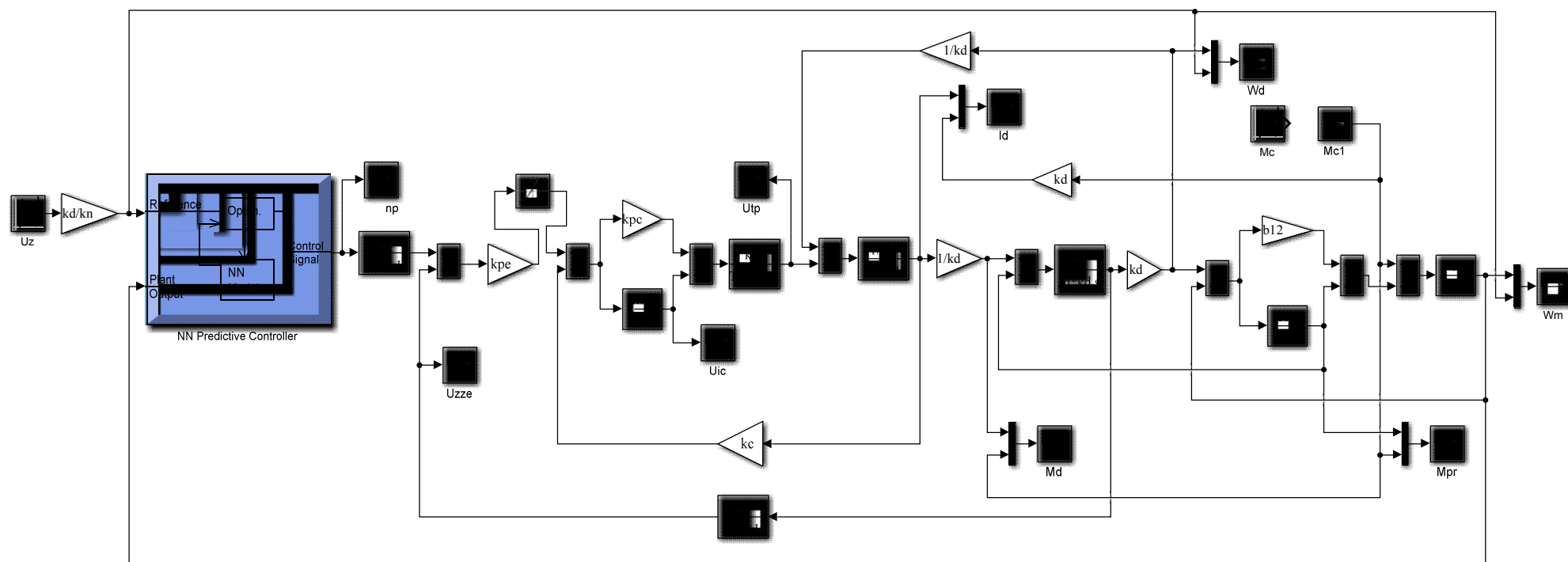
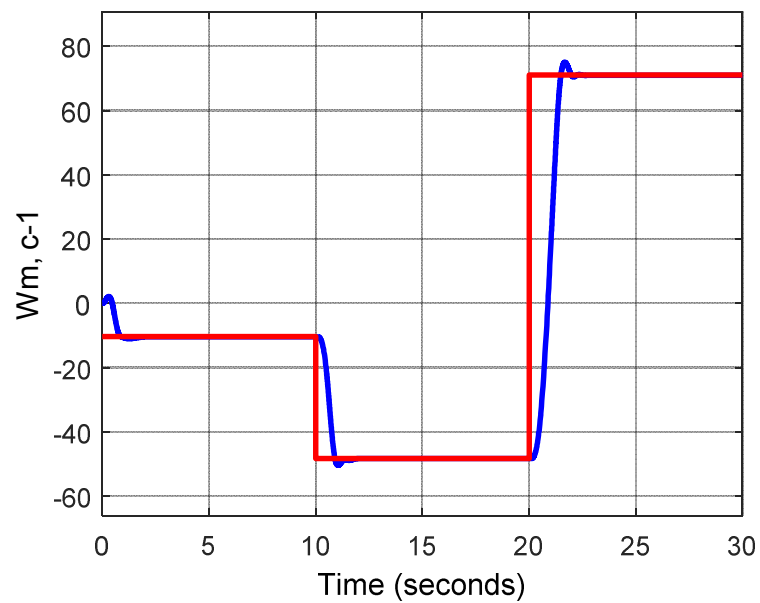
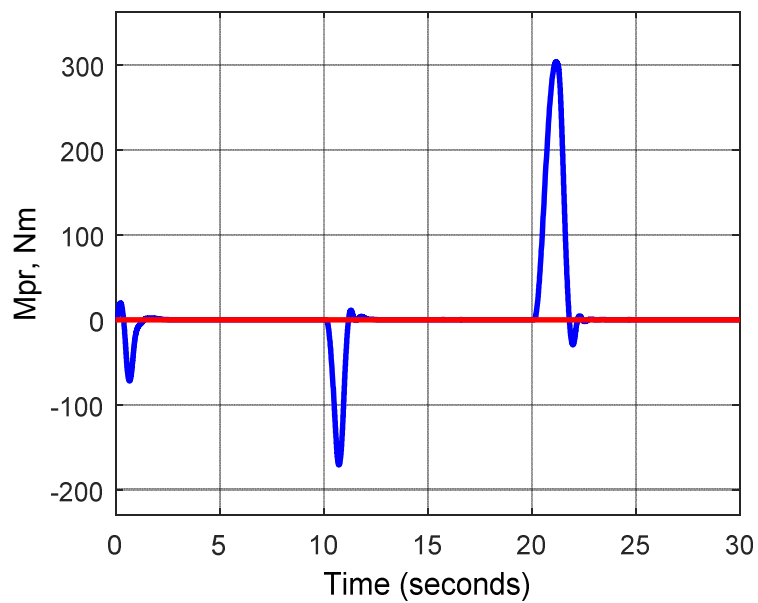


Рисунок 4.27 – Імітаційна модель двомасової системи з нейрорегулятором



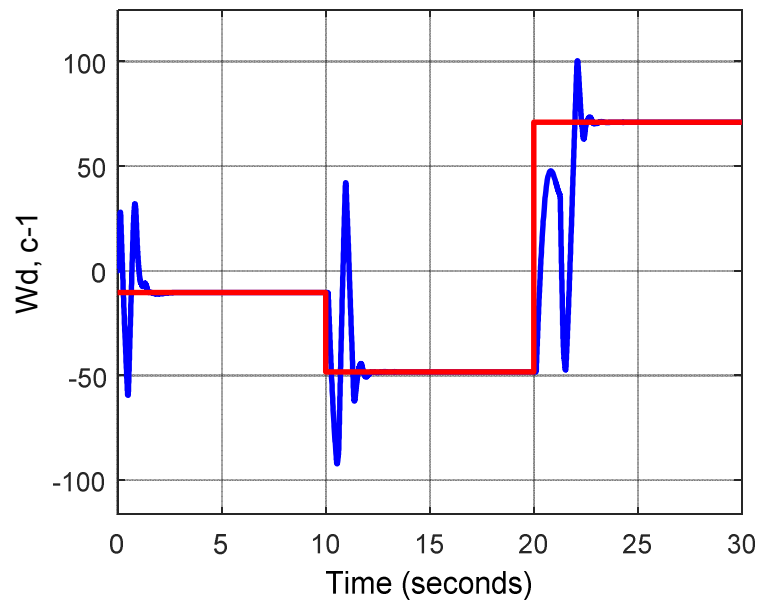
а)



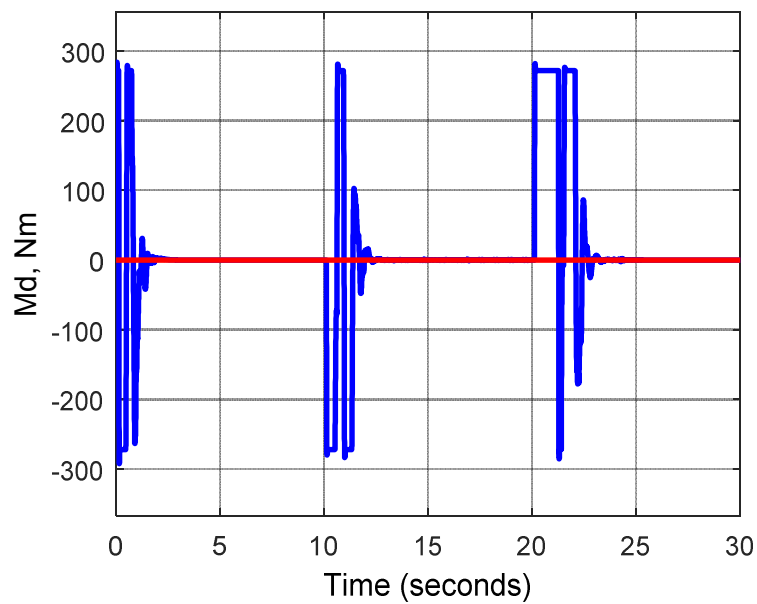
б)

Рисунок 4.28 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією задаючої величини:

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



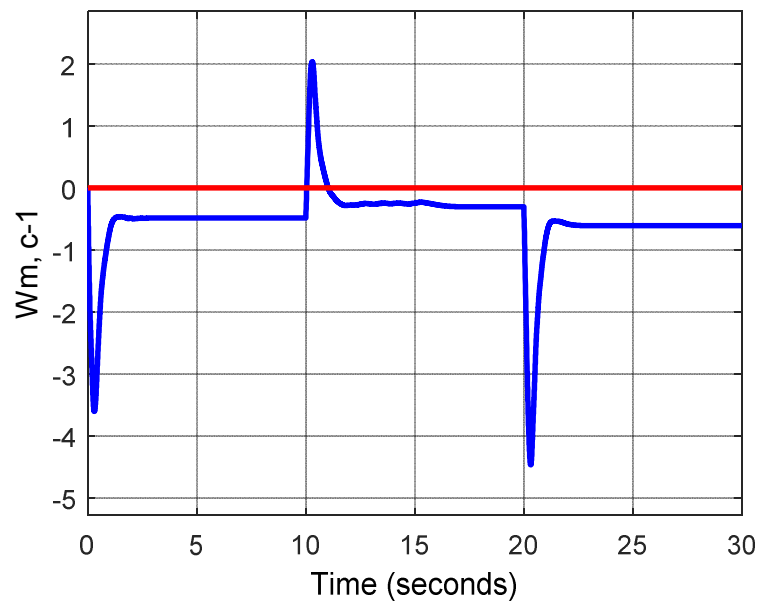
а)



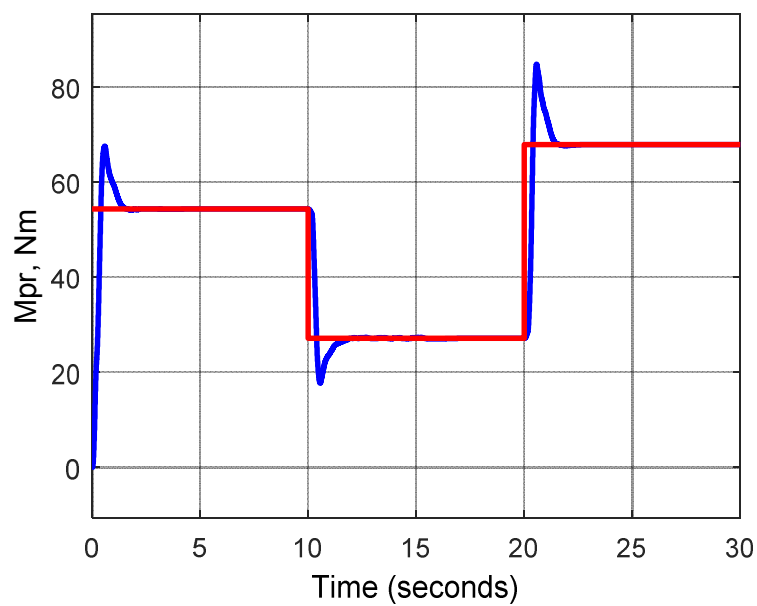
б)

Рисунок 4.29 – Динаміка двигуна двомасової системи з нейрорегулятором під впливом задаючої величини:

а) швидкість обертання двигуна $\omega_d = f(t)$, б) момент двигуна $M_d = f(t)$



a)



б)

Рисунок 4.30 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією зовнішнього збурення:

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

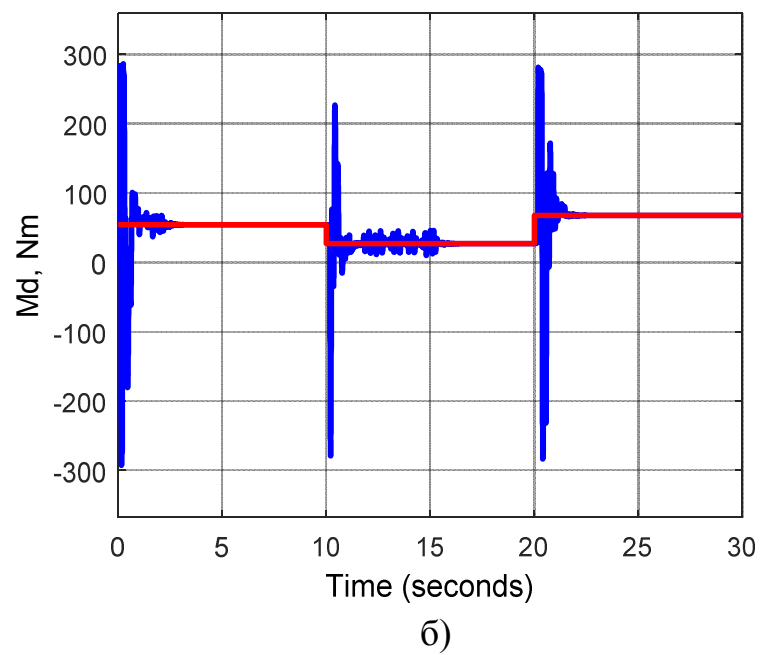
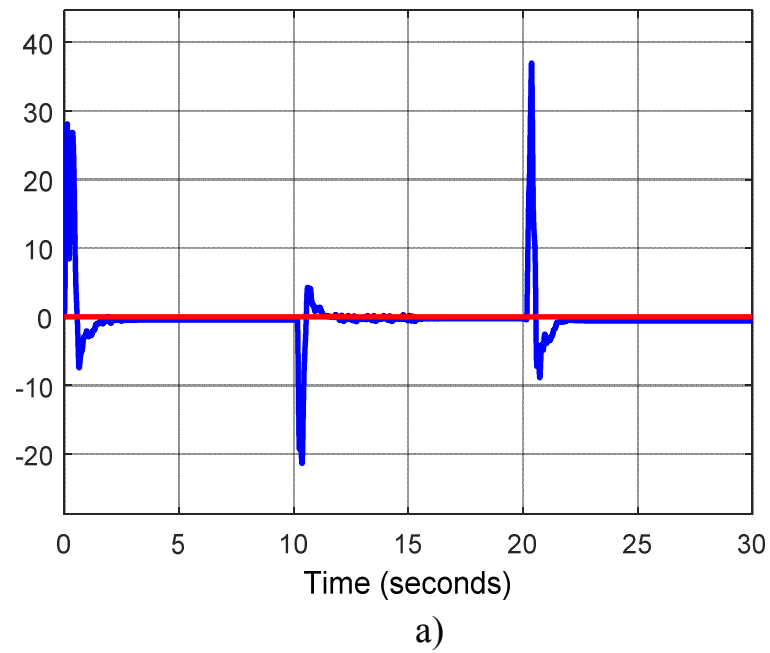


Рисунок 4.31 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією зовнішнього збурення:

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

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проведено аналіз принципів побудови нейрорегуляторів, реалізованих у пакеті прикладних програм MATLAB Neural Network Toolbox. Розглянуто три основні підходи: регулятор з прогнозом (NN Predictive Controller), регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) та регулятор на основі еталонної моделі (Model Reference Controller). Для системи управління механізмом підйому мостового крана, що досліджується в роботі, обрано регулятор з прогнозом NN Predictive Controller. Було детально розглянуто його структуру, принцип роботи та порядок синтезу, що дозволяє реалізувати передбачення поведінки об'єкта управління на основі навченої моделі.

2. У середовищі SIMULINK системи MATLAB розроблено модель системи управління з використанням нейрорегулятора NN Predictive Controller. Модель включає блок керованого об'єкта, який відтворює динаміку двомасової системи механізму підйому мостового крана, та блок нейрорегулятора. Така конфігурація дозволяє перевірити ефективність регулювання та проводити чисельне моделювання перехідних процесів системи з урахуванням реальних динамічних властивостей механізму.

3. Виконано генерацію навчальної послідовності для тренування нейромережі шляхом варіювання параметрів моделі виконавчого пристрою. Це забезпечило високий ступінь точності навчання мережі. Оптимальні параметри навчальної послідовності встановлено наступні: довжина вибірки – 20 000 значень; інтервал між двома послідовними моментами знімання даних – 0,05 с; мінімальний інтервал ідентифікації – 4 с; максимальний інтервал ідентифікації – 10 с. Такі параметри дозволили створити репрезентативну навчальну вибірку для формування високоякісної моделі об'єкта управління.

4. Створено двошарову нейронну мережу з прямою передачею сигналу та виконано її навчання за алгоритмом Левенберга-Марквардта. Визначено оптимальну кількість нейронів у прихованому шарі для досягнення балансу між точністю моделі та складністю мережі. Помилка навчання мережі мала порядок

10^{-4} , а миттєві помилки на навчальній, контрольній та тестовій множинах також не перевищували 0,4. Коефіцієнт кореляції R дорівнював 1, що свідчить про високу точність ідентифікації моделі об'єкта управління нейромережею.

5. Проведено синтез нейрорегулятора з прогнозом NN Predictive Controller. Шляхом варіювання параметрів нейрорегулятора визначено, які з них істотно впливають на якість регулювання, і встановлено оптимальні значення цих параметрів. Використання нейромережевої моделі об'єкта управління забезпечило високий рівень ідентифікації та дало змогу синтезувати регулятор, який забезпечує високі показники якості системи: точність регулювання, швидкість реакції та стабільність перехідних процесів.

6. Проведено моделювання роботи системи з синтезованим нейрорегулятором. Аналіз результатів показав, що перехідні процеси змінних стану системи в режимі пуску та при різкому наборі навантаження мають задовільний характер. Система виявилася астатичною, з невеликим перерегулюванням (не більше 5 %) та часом регулювання близько 2 с, що повністю відповідає поставленим вимогам до динаміки системи та забезпечує стабільну та точну роботу механізму підйому мостового крана.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розв'язана актуальна науково-практична задача розробки нейромережевої системи управління багатомасовою електромеханічною системою механізму переміщення візка мостового крана, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному:

1. Проведено аналіз існуючих систем управління основних механізмів мостових кранів з урахуванням вимог сучасних систем автоматизації та високих стандартів якості регулювання. Науково обґрунтовано доцільність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму переміщення візка мостового крана. Встановлено, що використання нейромережевих регуляторів дозволяє підвищити точність і швидкодію системи, забезпечити адаптивність до змінних навантажень і компенсувати нелінійності об'єкта управління.

2. На підставі аналізу технічних вимог до систем електроприводів основних механізмів мостових кранів обрано електропривод постійного струму, виконаний за системою ТП–Д, для механізму переміщення візка. Виконано вибір електродвигуна та основного електроустаткування силового кола електроприводу, проведено перевірку двигуна на відповідність умовам нагріву та перевантажувальної здатності.

3. Проведено розрахунок і вибір системи автоматичного управління параметрами електроприводу. Для управління координатами електроприводу застосовано систему підлеглого регулювання з внутрішнім контуром струму та зовнішнім контуром ЕРС. Виконано розрахунок параметрів силового кола, що використовуються при побудові системи автоматичного управління. Синтезовано послідовні коректуючі пристрої для контурів струму та ЕРС з оптимізацією за модульним критерієм. Проведено чисельне моделювання системи у MATLAB, що дозволило оцінити перехідні процеси та підтвердити адекватність розрахункових моделей.

4. Розроблено математичну модель двомасової системи, яка враховує динамічні особливості системи при поперечних розгойдуваннях вантажу в перехідних режимах. Моделювання показало, що в двомасовій системі перехідні процеси основних координат мають значні коливання, що обумовлено кінцевою жорсткістю механічних зв'язків і взаємодією мас візка та вантажу. Це підкреслює необхідність використання адаптивних або прогнозних регуляторів для стабілізації системи та покращення точності переміщень.

5. Для забезпечення бажаних динамічних характеристик виконано синтез системи управління з нейрорегулятором NN Predictive Controller, реалізованим у MATLAB Neural Network Toolbox. Розглянуто принцип побудови та функціонування нейромережевого регулятора, наведено порядок його синтезу та навчання. Проведено моделювання роботи двомасової системи з нейрорегулятором, у якому як вхідна дія застосовувався ступінчастий сигнал з випадковою амплітудою в межах номінальних значень. Аналіз результатів показав, що система забезпечує задовільний перехідний процес із швидким загасанням коливань. Система є астатичною, перерегулювання не перевищує 5%, а час регулювання складає 2 с, що повністю відповідає поставленим технічним вимогам.

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. Тимошук 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. 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
14. Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu, *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*, IEEE/NLN (2021) (Accepted Manuscript)
15. 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
16. 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.
17. 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.
18. 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.
19. Cybenko G. Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314.
20. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // Neural Networks. – 1989. – 2. – P. 359-366
21. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks / Neural Networks. – 1991. – 4. – P. 251-257.
22. Pinkus, A. (1999). *Approximation theory of the MLP model in neural networks*. Acta Numerica, 8, 143–195.
23. Hopfield J.J., Tank D. W. Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1985. – 52. – P. 141-152.

24. 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.
25. 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.
26. Rawat W., Wang Z. Deep convolutional neural networks for image classification: A comprehensive review // *Neural Computation*. – 2017. – Vol. 29, No. 9. – P. 2352–2449.
27. 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.
28. 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.
29. 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
30. 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.
31. 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.
32. Esteva A., Robicquet A., Ramsundar B., et al. A guide to deep learning in healthcare // *Nature Medicine*. – 2019. – Vol. 25. – P. 24–29.
33. 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.

34. 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.
35. Runge, J., et al. Renewable energy forecasting: A review // Renewable and Sustainable Energy Reviews. – 2019. – Vol. 120. – 109642.
36. 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.
37. 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.
38. 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.
39. 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.
40. 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.
41. B. T. Sanghai. *Advances in Transformers for Robotic Applications: A Review* : e-print (arXiv) / N. Sanghai. – 2024.
42. Pyzer-Knapp, E.O., Manica, M., Staar, P. *et al.* Foundation models for materials discovery – current state and future directions. *npj Comput Mater* 11, 61 (2025).
43. *A Survey on State-of-the-art Deep Learning Applications* : e-print (arXiv) / Mohd Halim Mohd Noor, Ayokunle Olalekan Ige. - 2024.
44. *Neuromorphic Computing: Innovations and Future Prospects* / Panja Naga Laxmi // International Journal of Advanced Research in Science Communication and Technology - 2024.

45. Sen P., Hearn G.E., Zhang Y. Adaptive Neural Controller // In Neural Network Systems Techniques and Applications. Hrsg. Leondes C.T. – 1998. – Vol.4. – P. 274-343.
46. Werbos P.J. Backpropagation and neurocontrol: A review and prospects // Proc. of International Joint Conf. On Neural Networks. – Washington DC. – 1989. – Vol. 1. – P. 209-216.
47. Psaltis A.; Sideris A., Yamamura A. A multi-layered neural network controller // IEEE Control Systems Mag. – 1988. – №8. – P. 17-21.
48. Saerens M.; Soquet A. A neural controller // Proceedings of the 1st IEE International Conference on Artificial Neural Networks, London. – 1989. – P. 211-215.
49. Narendra K.S. Neural networks for control: theory and practice // Proc. IEEE. – 1996. – №84. – P. 1385-1406.
50. Neural networks for control systems: A survey K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop // Automatica. – 1992. – Vol. 28. – № 6. P. 1083-1112.
51. Ishiguro A. et al. A neural network compensator for uncertainties of robotic manipulators // IEEE Trans, on Industrial Electronics. – 1992. – 339. – P. 565-569.
52. Garcia C.E.; Morari M. Internal Model Control: Unifying Review and Some New Results // Ind. Eng. Chem. Process Des. Dev. – 1982. – №21. – 308-323.
53. Park Y.-M., Choi M.S.; Lee Y. An optimal tracking neuro-controller for nonlinear dynamic systems // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1099-1110.
54. Mistry, S.I.; Nair S.S. Indirect control of a class of nonlinear dynamic systems. // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1015-1023.
55. Miyamoto H., Kawato M., Setoyama, T., Suzuki R. Feedback error learning neural network for trajectory control of robotic manipulator // IEEE Trans, on Neural Networks. – 1988. – Vol. 1. – P. 251-265.
56. Jung S. Neural Network Controllers for Robot Manipulators. PhD thesis, University of California. – Davis. – 1996. – 132 p.
57. Jung S.; Hsia T.C. A new neural network control technique for robot manipulators // Robotica. – 1995. – Vol. 13. – P. 477-484.

58. Sorensen O. Neural Networks in Control Applications. PhD thesis. Aalborg University. Department of Control Engineering. – 1994. – 187 p.
59. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // *Automatica*. – 1989. – Vol.25. – P.335-348.
60. 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.
61. 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.
62. Review on model predictive control: an engineering perspective – *The International Journal of Advanced Manufacturing Technology*, 2021, Vol. 117, pp. 1327-1349.
63. Lan J. *Efficient model predictive control for nonlinear systems modelled by deep neural networks* : e-print (arXiv) / J. Lan. - 2024.
64. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
65. 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.
66. 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.
67. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.
68. Теорія автоматичного керування : навчальний посібник / О. К. Аблесімов – К. : «Освіта України», 2019. – 270 с.

69. 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.
70. Теорія автоматичного управління : курс лекцій / Т. М. Боровська. – Вінниця : ВНТУ, 2018. – 256 с.
71. 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
72. 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.
73. Автоматизований електропривод. Розділ 1. Теорія електропривода. Навчальний посібник. / Н.Д. Красношاپка – К.: КПІ ім. Ігоря Сікорського. - 2023. – 101 с.
74. Возняк, О.М., Штуць. А.А., Колісник М.А. Сучасні системи електроприводів. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ТВОРИ, 2021. – 280 с.
75. А. А. Видмиш, Л. В. Ярошенко. Основи електропривода. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ВНАУ, 2020. – 387 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. Колб Ант. А, Колб А. А. Теорія електроприводу: Навчальний посібник. – 2-е вид. перероб. і доп. – Д., Національний гірничий університет, 2011. – 540 с.
78. 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.

79. Василега П. О. Електропривод робочих машин : підручник / П. О. Василега. – Суми: Сумський державний університет, 2022. – 290 с. ISBN 978-966-657-900-6.

80. Шефер О.В. Автоматизований електропривод загальнопромислових механізмів: конспект лекцій. – Полтава: ПолтНТУ, 2020. – 154 с.

81. 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.

82. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоменко. Харків, ХНАДУ, 2012. – 284 с.

83. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

ДОДАТОК А

МОДЕЛЬ НЕЙРОНА

А.1 Структурна схема нейрона

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

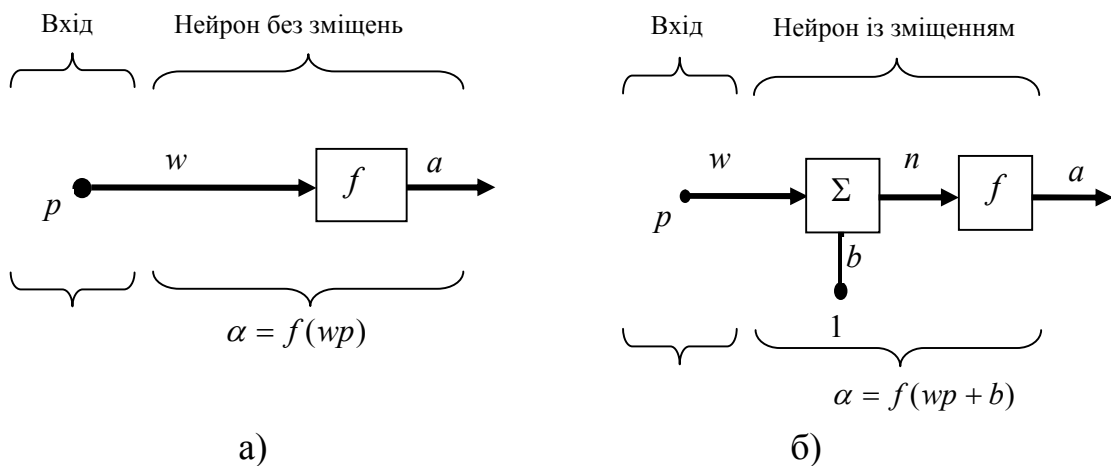


Рисунок А.1 – Структура простого нейрона

а) нейрон з одним скалярним входом;

б) нейрон з скалярним зсувом

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

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

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

Рівняння нейрона із зсувом має вигляд

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

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

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

A.2 Функції активації нейронів

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

Розглянемо три найбільш поширені форми функції активації.

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

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

Лінійна функція активації `purelin`. Ця функція описується співвідношення $a = \text{purelin}(n) = n$ і показана на рис. A1.3.

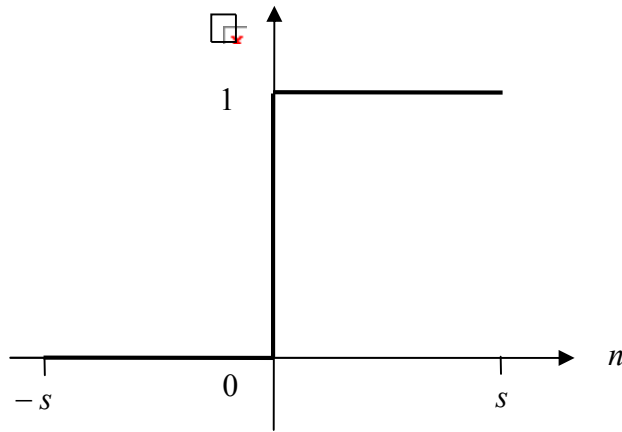


Рисунок А.2 – Одинична функція активації

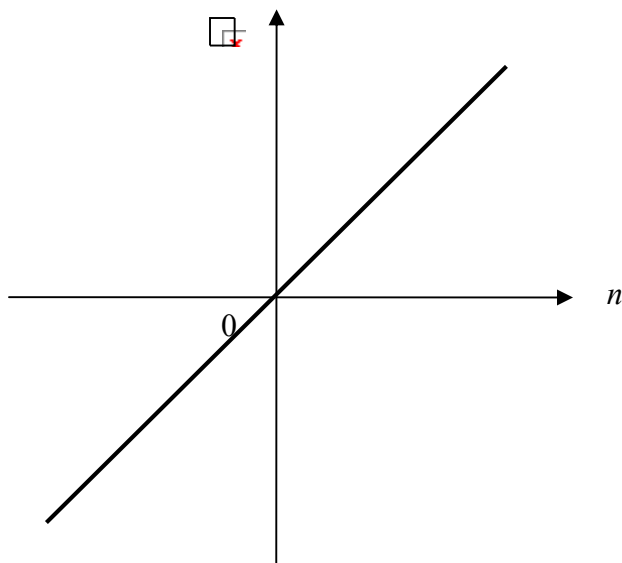


Рисунок А.3 – Лінійна функція активації

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

У багатошарових мережах часто застосовуються нелінійні сигмоїдальні функції активації типу гіперболічного тангенса (рис. А.5) і функції типу ReLU (рис. А.6)

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

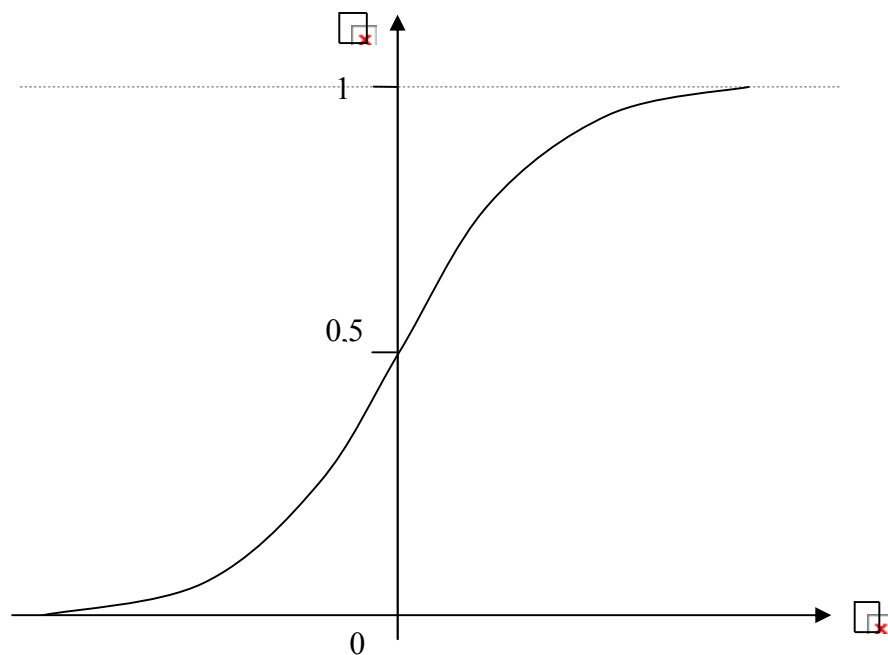


Рисунок А.4 – Логістична функція активації

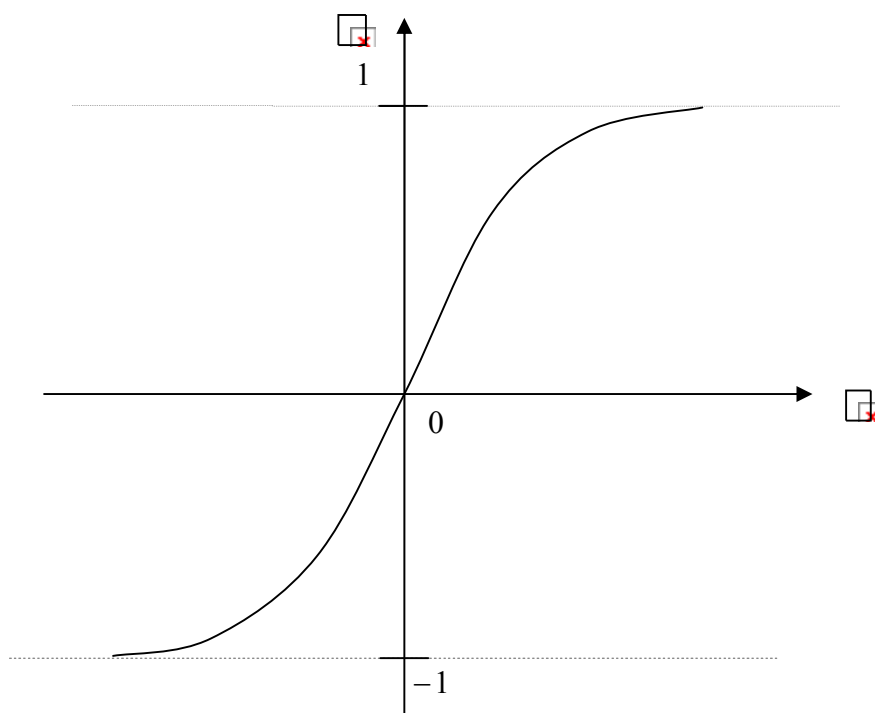


Рисунок А.5 – Функція активації типу гіперболічного тангенса

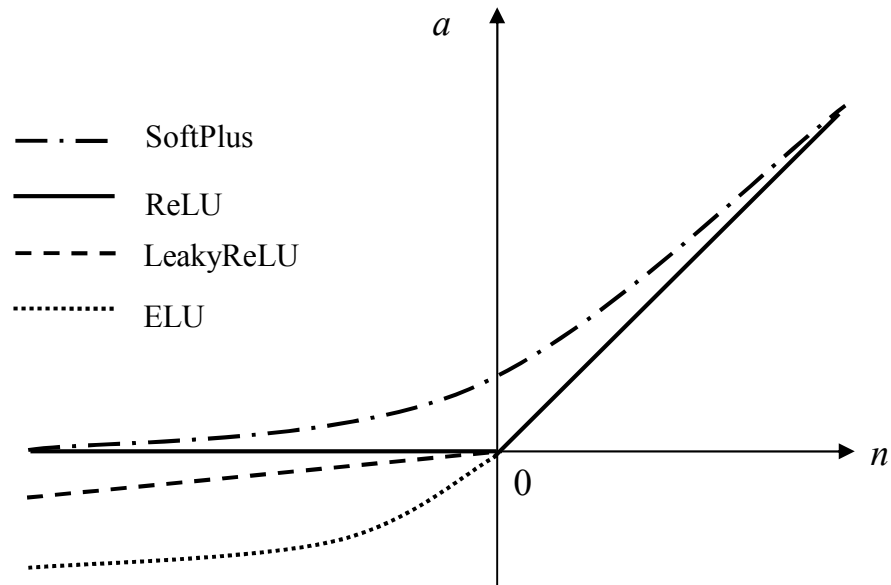


Рисунок А.6 – Функція активації типу ReLU

А.3 Нейрон з векторним входом

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

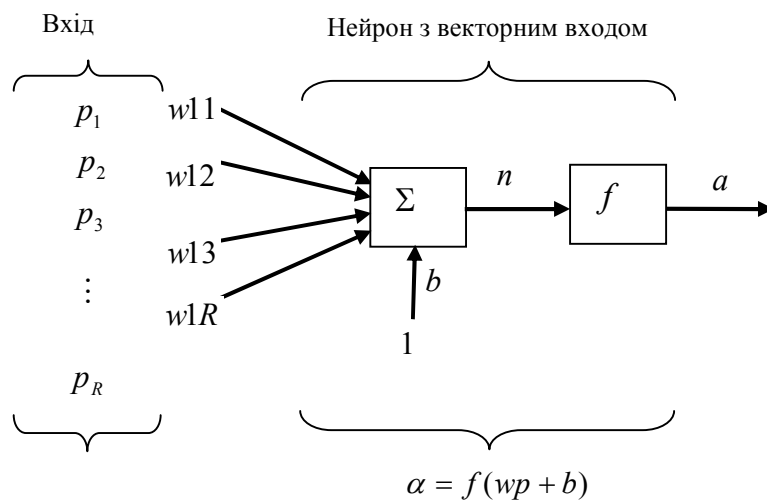


Рисунок А.6 – Схема нейрона з векторним входом

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

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

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

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

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

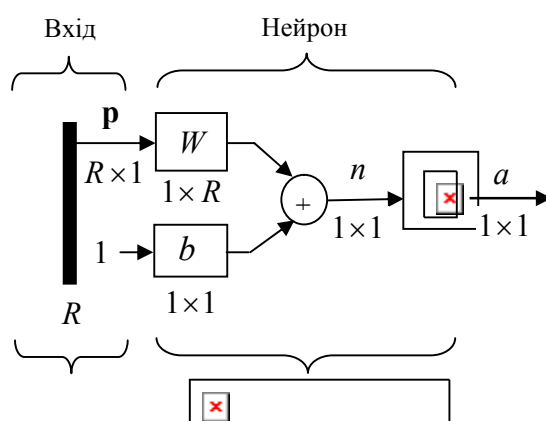


Рисунок А.7 – Укрупнена структурна схема нейрона

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

ДОДАТОК Б

АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

Б.1 Одношарові мережі

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

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

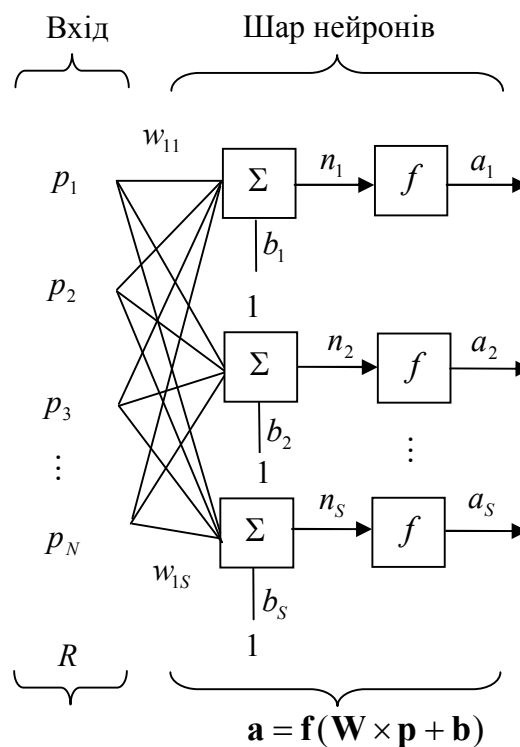


Рисунок Б.1 – Схема одношарової мережі

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

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

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

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

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

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

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

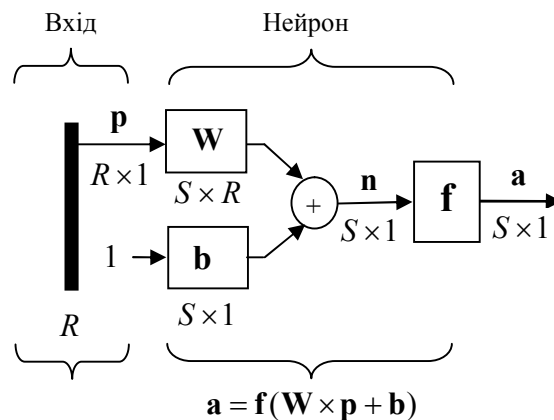


Рисунок Б.2 – Укрупнена структурна схема одношарової мережі

Б.2. Багатошарові мережі

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

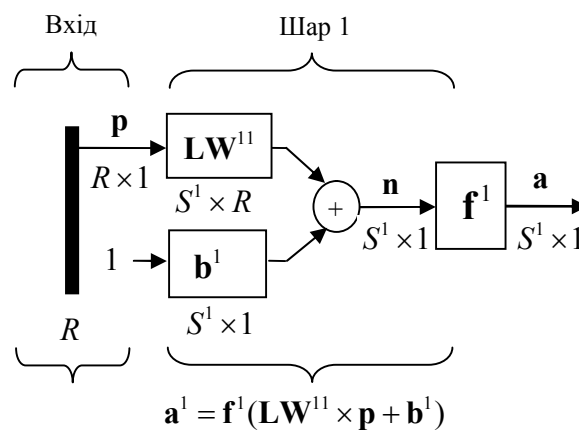


Рисунок Б.3 – Структурна схема першого шару багатошарової мережі

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

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

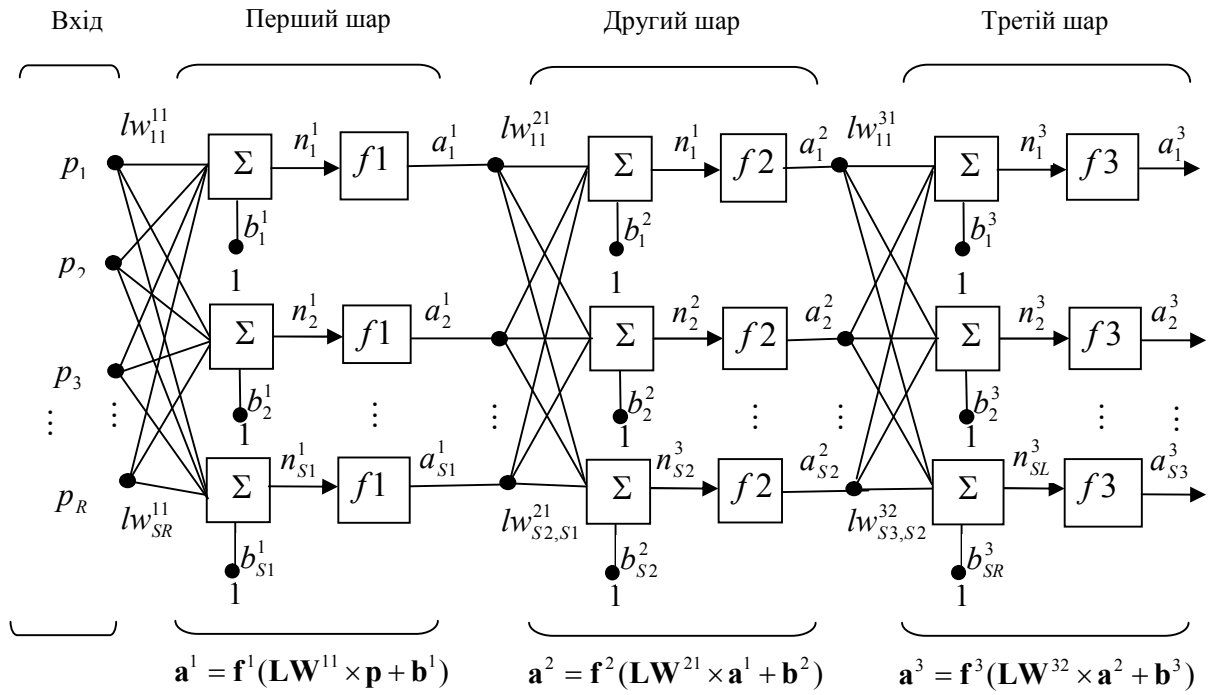


Рисунок Б.4 – Структурна схема багатошарової мережі

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

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

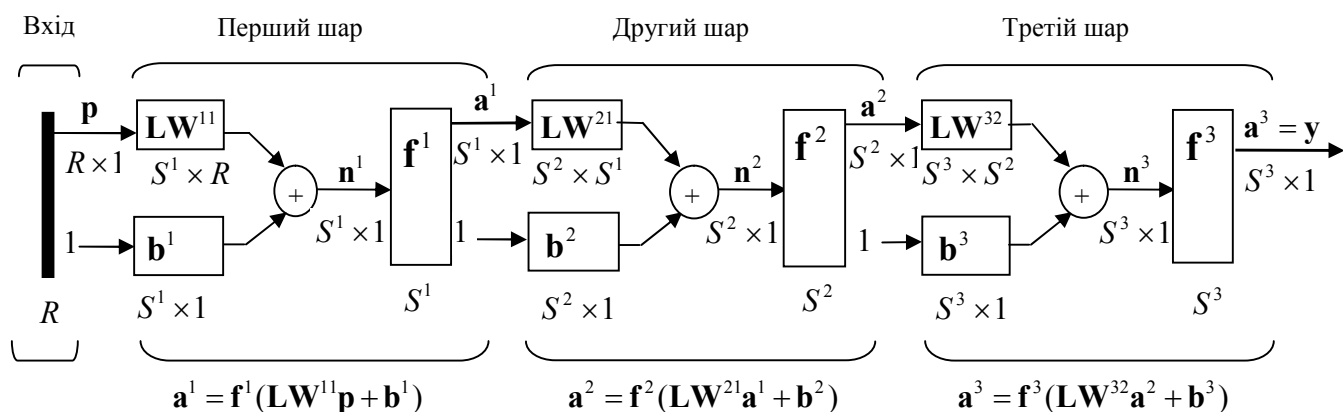


Рисунок Б.5 – Укрупнена структурна схема багатошарової мережі

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

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

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

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

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

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

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

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

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

Б.3 Мережі з прямою передачею сигналу

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

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

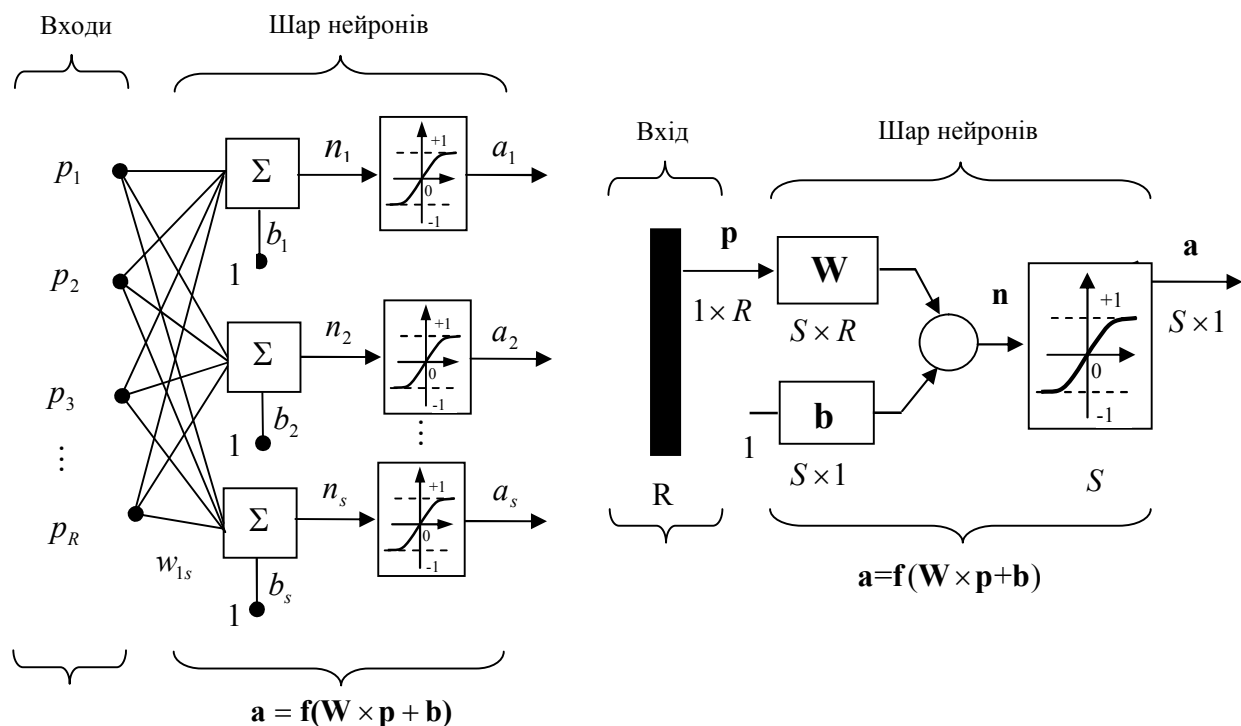


Рисунок Б.6 – Структурна схема одношарової мережі з R входами

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

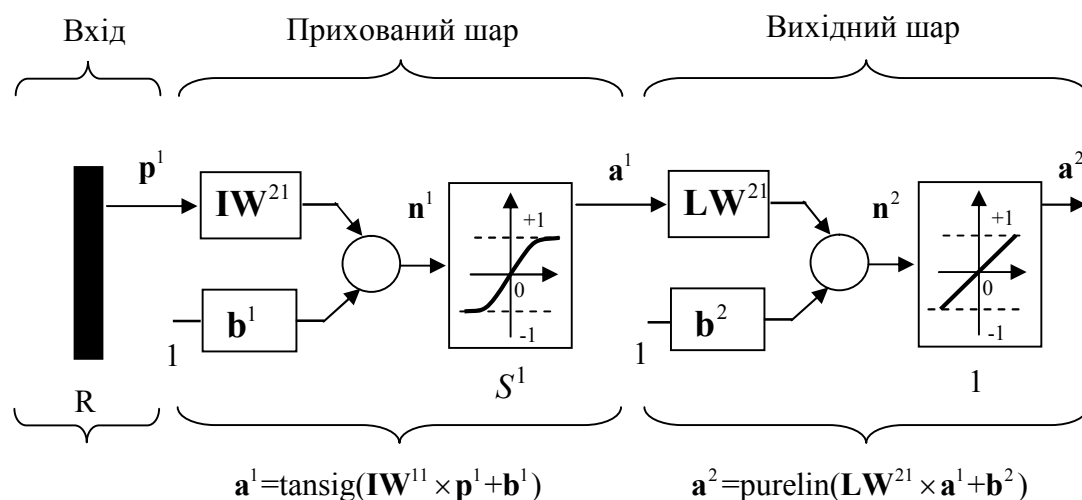


Рисунок Б.7 – Структурна схема багатошарової мережі з R входам

ДОДАТОК В

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

В.1 Завдання навчання

При рішенні за допомогою нейронних мереж прикладних завдань необхідно зібрати точний і представницький об'єм даних для того, щоб навчити нейронну мережу рішенню таких задач. Навчальний набір даних - це набір спостережень, що містять ознаки об'єкту, що вивчається. Перше питання, які ознаки використовувати і скільки і які спостереження треба провести.

Вибір ознак, принаймні первинний, здійснюється евристично на основі наявного досвіду, який може підказати, які ознаки є найбільш важливими. Спочатку слід включити всі ознаки, які, на думку аналітиків або експертів, є істотними, на подальших етапах ця множина буде скорочена.

Нейронні мережі працюють з числовими даними, узятими, як правило, з деякого обмеженого діапазону. Це може створити проблеми, якщо значення спостережень виходять за межі цього діапазону або пропущені.

Питання про те, скільки потрібно мати спостережень для навчання мережі, часто виявляється непростим. Відомий ряд евристичних правил, які встановлюють зв'язок між кількістю необхідних спостережень і розмірами мережі. Просте з них свідчить, що кількість спостережень повинна бути в 10 разів більше числа зв'язків в мережі. Насправді це число залежить від складності того відображення, яке повинна відтворювати нейронна мережа. Із зростанням числа використовуваних ознак кількість спостережень зростає по нелінійному закону, так що вже при досить невеликому числі ознак, скажемо 50, може потрібно величезне число спостережень. Ця проблема носить назву "Прокляття розмірності".

Для більшості реальних завдань буває достатнім декількох сотень або тисяч спостережень. Для складних завдань може потрібно більша кількість, проте дуже рідко зустрічаються завдання, де потрібно менше 100 спостережень. Якщо даних мало, то мережа не має достатньої інформації для навчання, і краще, що

можна в цьому випадку зробити, - це спробувати підігнати до даним деяку лінійну модель.

В.2 Процедури адаптації і навчання

Після того, як визначена кількість шарів мережі і число нейронів в кожному з них, потрібно призначити значення вагів і зсувів, які мінімізують помилку рішення. Це досягається за допомогою процедур навчання. Шляхом аналізу тих, що є у розпорядженні аналітика вхідних і вихідних даних ваги і зсуву мережі автоматично настроюються так, щоб мінімізувати різницю між бажаним сигналом і отриманим на виході в результаті моделювання. Ця різниця носить назву помилки навчання. Таким чином, процес навчання - це процес підгонки параметрів тієї моделі процесу або явища, яка реалізується нейронною мережею. Помилка навчання для конкретної конфігурації нейронної мережі визначається шляхом прогону через мережу всіх наявних спостережень і порівняння вихідних значень з бажаними, цільовими значеннями. Ці різниці дозволяють сформувати так звану функцію помилок (критерій якості навчання). Як така функція найчастіше береться сума квадратів помилок. При моделюванні нейронних мереж з лінійними функціями активації нейронів можна побудувати алгоритм, що гарантує досягнення абсолютного мінімум помилки навчання. Для нейронних мереж з нелінійними функціями активації в загальному випадку не можна гарантувати досягнення глобального мінімуму функції помилки.

При такому підході до процедури навчання може виявитися корисним геометричний аналіз поверхні функції помилок. Визначимо ваги і зсуву як вільні параметри моделі і їх загальне число позначимо через N ; кожному набору таких параметрів поставимо у відповідність одне вимірювання у вигляді помилки мережі. Тоді для всіляких поєднань вагів і зсувів відповідну помилку мережі можна зобразити крапкою в $N+1$ - мірному просторі, а всі такі крапки утворюють деяку поверхню, звану поверхнею функції помилок. При такому підході мета

навчання нейронної мережі полягає в тому, щоб знайти на цій багатовимірній поверхні глобальний мінімум.

У разі лінійної моделі мережі і функції помилок у вигляді суми квадратів така поверхня буде параболоїдом, який має єдиний мінімум і це дозволяє відшукати такий мінімум досить просто.

У разі нелінійної моделі поверхню помилок має набагато складніша будова і володіє поряд несприятливих властивостей, зокрема може мати локальні мінімуми, плоскі ділянки, сідлові крапки і довгі вузькі яри.

Визначити глобальний мінімум багатовимірної функції аналітично неможливо, і тому навчання нейронної мережі, по суті справи, є процедурою вивчення поверхні функції помилок. Відштовхуючись від випадково вибраної крапки на поверхні функції помилок, алгоритм навчання поступово відшукує глобальний мінімум. Як правило, для цього обчислюється градієнт (нахил) функції помилок в даній крапці, а потім ця інформація використовується для просування вниз по схилу. Врешті-решт алгоритм зупиняється в деякому мінімумі, який може опинитися лише локальним мінімумом, а якщо повезе, то і глобальним.

Таким чином, по суті алгоритми навчання нейронних мереж аналогічні алгоритмам пошуку глобального екстремуму функції багатьох змінних. Серед останніх слід виділити алгоритми зв'язаних градієнтів і Левенберга - Марквардта (Levenberg - Marquardt).

Проте з урахуванням специфіки нейронних мереж для них розроблені спеціальні алгоритми навчання, серед яких слід виділити алгоритм зворотного розповсюдження помилки.

При використанні алгоритму зворотного розповсюдження помилки мережа розраховує помилку, що виникає у вихідному шарі, і обчислює вектор градієнта як функцію вагів і зсувів. Цей вектор указує напрям найкоротшого спуску по поверхні для даної крапки, тому якщо просунутися в цьому напрямі, то помилка зменшиться. Послідовність таких кроків врешті-решт приведе до міні-

муму того або іншого типу. Певну трудність тут викликає вибір величини кроку.

При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути через рішення або піти в неправильному напрямі. Класичним прикладом такого явища при навчанні нейронної мережі є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного схилу на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. На практиці величина кроку вибирається пропорційній крутизні схилу (градієнту функції помилок); такий коефіцієнт пропорційності називається параметром швидкості настройки. Правильний вибір параметра швидкості настройки залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; цей параметр може також залежати від часу, зменшуючись у міру виконання алгоритму.

Алгоритм діє ітеративно, і його кроки прийнято називати епохами або циклами. На кожному циклі на вхід мережі послідовно подаються всі повчальні спостереження, вихідні значення порівнюються з цільовими значеннями і обчислюється функція помилки. Значення функції помилки, а також її градієнта використовуються для коректування вагів і зсувів, після чого всі дії повторюються. Початкові значення вагів і зсувів мережі вибираються випадковим чином, і процес навчання припиняється або коли реалізована певна кількість циклів, або коли помилка досягне деякого малого значення або перестане зменшуватися.

В.3 Явище перенавчання

Одна з найбільш серйозних труднощів при навчанні мережі полягає в тому, що у ряді випадків ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати; потрібно мінімізувати помилку, яка з'являється в мережі, коли на неї подаються абсолютно нові спостереження. Вельми важливо, щоб нейронна мережа володіла здатністю пристосовуватися до цих нових спостережень. Що ж відбувається насправді? Мережа навчається мінімізувати помилку на деякій

обмеженій повчальній множині. Це не відповідає вимогам теорії про наявність ідеальної і нескінченно великої повчальної множини. І це не відповідає тій реальній ситуації, коли треба мінімізувати конкретну функцію помилок для наперед невідомої моделі.

Це породжує проблему, яка відома як явище перенавчання. Звернемося до завдання апроксимації деякої функції многочленом. Графіки многочленів часто мають вельми хитромудрі форми, і чим вище ступінь многочлена, тим складніше їх форма. Якщо є деякий набір даних, то можна поставити мету підібрати для нього апроксимуючий многочлен і таким чином отримати відповідну математичну модель для цього набору даних. Оскільки початкові дані, як правило, задані з погрішностями, то не можна вважати, що краща модель задається кривою, яка проходить точно через задані точки. Многочлен низького порядку може виявитися достатньо грубим для апроксимації даних, тоді як многочлен високого порядку може точно слідувати даним, приймаючи при цьому вельми хитромудру форму, що не має ніякого відношення до форми дійсної залежності. Остання ситуація і демонструє те, що називається явищем перенавчання.

При роботі з нейронними мережами користувач стикається з тією ж проблемою. Мережі з великою кількістю вагів дозволяють відтворювати дуже складні функції, і в цьому сенсі вони схильні до перенавчання. Мережа ж з невеликою кількістю вагів може опинитися недостатньо гнучкою, щоб змоделювати наявну залежність. Наприклад, одношарова лінійна мережа здатна відтворювати тільки лінійні функції. Якщо використовувати багатшарові лінійні мережі, то помилка завжди буде менша, але може свідчити не про хорошу якість моделі, а про те, що виявляється явище перенавчання.

Для того, щоб виявити ефект перенавчання, використовується механізм контрольної перевірки. Частина повчальних спостережень резервується як контрольні спостереження і не використовується при навчанні мережі. Натомість у міру роботи алгоритму ці спостереження застосовуються для незалежного контролю результату. Спочатку помилка мережі на повчальне і контрольному множині буде однаковою; якщо вони істотно відрізняються, то, мабуть, це

означає, що розбиття спостережень на 2 множини не забезпечило їх однорідність. У міру навчання мережі помилка убиває, і, поки навчання зменшує функцію помилок помилка на контрольній множині також убиватиме. Якщо ж контрольна помилка перестала убивати або стала рости, це указує на те, що мережа почала дуже близько слідувати початковим даним і навчання слід зупинити. В цьому випадку слід зменшити кількість нейронів або шарів, бо мережа є дуже могутнього для вирішення даного завдання. Якщо ж, навпаки, мережа має недостатню потужність, щоб відтворити наявну залежність, те явище перенавчання швидше за все спостерігатися не буде і обидві помилки - навчання і перевірки - не досягнуть достатньо малого рівня.

Що виникають при роботі з нейронними мережами проблеми відшукування глобального мінімуму або вибору розміру мережі призводять до того, що при практичній роботі доводиться експериментувати з великим числом мереж різних конфігурацій, деколи навчаючи кожен з них кілька разів і порівнюючи отримані результати. Головним критерієм вибору в цих випадках є контрольна погрішність. При цьому застосовується правило, згідно якому з двох нейронних мереж з приблизно рівними контрольними погрішностями слід вибрати ту, яка простіше.

Необхідність багатократних експериментів веде до того, що контрольна множина починає грати ключову роль у виборі моделі нейронної мережі, тобто стає частиною процесу навчання. Тим самим його роль як незалежного критерію якості моделі ослабляється, оскільки при великому числі експериментів виникає ризик перенавчання нейронної мережі на контрольній множині. Для того, щоб гарантувати надійність вибраної моделі мережі, резервують ще одне - тестова безліч спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на повчальній і контрольній множинах реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки 1 раз: якщо його використовувати повторно для коректування процесу навчання, то воно фактично перетвориться на контрольну множину.

Отже, процедура побудови нейронної мережі складається з наступних кроків:

вибору початкової конфігурації мережі; наприклад, у вигляді одного шару з числом нейронів, рівним $1/2$ загальної кількості входів і виходів;

моделювання і навчання мережі з оцінкою контрольної помилки і використанням додаткових нейронів або проміжних шарів;

виявлення ефекту перенавчання і коректування конфігурації мережі.

В.4 Властивість узагальнення

При описі процедури навчання нейронних мереж неявно використовувалося припущення, що повчальна, контрольна і тестова множини є представницькими для вирішуваної задачі. Зазвичай як повчальні беруться дані, випробувань на ряду прикладів. Якщо обставини змінилися, то закономірності, що мали місце у минулому, можуть більше не діяти.

Крім того, нейронна мережа може навчатися тільки на тих даних, які вона має в своєму розпорядженні. Припустимо, що відоме повчальна множина для системи стабілізації літака при польоті в спокійній атмосфері, а потрібно спроектувати систему стабілізації на основі нейронної мережі для умов польоту при сильних обуреннях. Тоді навряд чи можна чекати від мережі правильного рішення в абсолютно новій для неї ситуації.

Класичним прикладом непредставницької моделі нейронної мережі є наступна ситуація. При проектуванні системи машинного зору, призначеної для автоматичного розпізнавання цілей, мережа навчалася на 100 картинках, що містять зображення танків, і на 100 інших картинках, де танків не було. Після навчання мережі був досягнутий стовідсотково "правильний" результат. Але коли на вхід мережі були подані нові дані, вона безнадійно провалилася. У чому ж була причина? З'ясувалося, що фотографії з танками були зроблені в похмурий, дощовий день, а фотографії без танків - в сонячний день. Мережа навчилася уловлювати різницю в загальній освітленості. Щоб мережа могла результатив-

но працювати, її слід було навчати на даних, де б були присутні всі погодні умови і типи освітлення, при яких мережу передбачається використовувати, і це не кажучи ще про рельєф місцевості, вугілля і дистанцію зйомки і т.д.

Якщо мережа мінімізує загальну погрішність, великого значення набувають пропорції, в яких представлені дані різних типів. Мережа, навчена на 900 "хороших" і 100 "поганих" спостереженнях, спотворюватиме результат на користь хороших спостережень, оскільки це дозволить алгоритму зменшити загальну погрішність. Якщо в реальній ситуації "хороші" і "погані" об'єкти представлені в іншій пропорції, то результати, що видаються мережею, можуть виявитися невірними. Прикладом цього може бути завдання виявлення захворювань. Хай, наприклад, при звичайних обстеженнях в середньому 90% людей виявляються здоровими і мережа, таким чином, навчається на даних, в яких пропорція здорові/хворі рівна 90/10. Потім ця ж мережа застосовується для діагностики пацієнтів з певними скаргами, серед яких співвідношення здорові/хворі вже 50/50. В цьому випадку мережа ставитиме діагноз занадто обережно і не розпізнаватиме захворювання у деяких хворих. Якщо ж, навпаки, мережу навчити на даних "з скаргами", а потім протестувати на "звичайних" даних, то вона видаватиме підвищене число неправильних діагнозів про наявність захворювання. У таких ситуаціях почальні дані потрібно скоректувати так, щоб були враховані відмінності в розподілі даних (наприклад, можна повторити рідкісні спостереження або видалити що часто зустрічаються). Як правило, краще всього постаратися зробити так, щоб спостереження різних типів були представлені рівномірно, і відповідно цьому інтерпретувати результати, які видає мережа.

Здатність мережі, навченої на деякій безлічі даних, видавати правильні результати для достатньо широкого класу нових даних, у тому числі і не представлених при навчанні, називається властивістю узагальнення нейронної мережі.

Інший підхід до процедури навчання мережі можна сформулювати, якщо розглядати її як процедуру, зворотну моделюванню. В цьому випадку потрібно підібрати такі значення вагів і зсувів, які забезпечували б потрібну відповід-

ність між входами і бажаними значеннями на виході. Така процедура навчання носить назву адаптації і достатньо широко застосовується для настройки параметрів нейронних мереж.

В.5 Методи навчання нейронних мереж

Як тільки початкові ваги і зсуву нейронів встановлені користувачем або за допомогою датчика випадкових чисел, мережа готова для того, щоб почати процедуру її навчання. Мережа може бути навчена рішенню різних прикладних задач - апроксимації функцій, ідентифікації і управління об'єктами, розпізнавання образів, класифікації об'єктів і т.п. Процес навчання вимагає набору прикладів її бажаної поведінки - входів p і бажаних (цільових) виходів t ; під час цього процесу ваги і зсуву настраюються так, щоб мінімізувати деякий функціонал помилки. За умовчанням як такий функціонал для мереж з прямою передачею сигналів приймається середньоквадратична помилка між векторами виходу $\mathbf{a}$ і $\mathbf{t}$. Розглянемо декілька методів навчання для мереж з прямою передачею сигналів.

При навчанні мережі розраховується деякий функціонал, що характеризує якість навчання:

$$J = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^M} (t_i^q - a_i^{qS^M})^2, \quad (\text{В.1})$$

де J - функціонал;

Q - об'єм вибірки;

M - число шарів мережі;

q - номер вибірки;

S^M - число нейронів вихідного шаруючи;

$\mathbf{a}^q = [a_i^{qM}]$ - вектор сигналу на виході мережі;

$\mathbf{t}^q = [t_i^q]$ - вектор бажаних (цільових) значень сигналу на виході мережі

для вибірки з номером q .

Потім за допомогою того або іншого методу навчання визначаються значення параметрів (вагів і зсувів), що настраюються, мережі, які забезпечують мінімальне значення функціонала помилки. Більшість методів навчання заснована на обчисленні градієнта функціонала помилки по параметрах, що настраюються.

В.5.1 Навчання одношарової мережі.

Найпростіше градієнт функціонала обчислюється для одношарових нейронних мереж. В цьому випадку $M=1$ і вираз для функціонала приймає вигляд:

$$J = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^1} (t_i^q - a_i^{qS^1})^2 = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^M} (t_i^q - f(n_i^q))^2 \quad i = \overline{1, S}, \quad (B.2)$$

де $f(n_i^q)$ - функція активації;

$n_i^q = \sum_{j=0}^R w_{ij} p_j^q$ - сигнал на вході функції активації для i -го нейрона;

$\mathbf{p}^q = [p_i^q]$ - вектор вхідного сигналу;

R - число елементів вектора входу;

S - число нейронів в шарі;

w_{ij} - вагові коефіцієнти мережі.

Включимо вектор зсуву до складу матриці вагів $\mathbf{W} = [w_{ij}]$, $i = \overline{1, S}$, $j = \overline{1, R}$, а вектор входу доповнимо елементом, рівним 1.

Застосовуючи правило диференціювання складної функції, обчислимо градієнт функціонала помилки, припускаючи при цьому, що функція активації диференцируема.

$$\frac{\partial J}{\partial w_{ij}} = - \sum_{q=1}^Q (t_i^q - f(n_i^q)) \frac{\partial(f(n_i^q))}{\partial w_{ij}} = - \sum_{q=1}^Q (t_i^q - f(n_i^q)) f'(n_i^q) p_j^q, \quad (B.3)$$

Введемо позначення

$$\Delta_i^q = (t_i^q - f(n_i^q))f'(n_i^q)p_j^q = (t_i^q - a_i^q)f'(n_i^q)p_j^q, \quad i = \overline{1, S} \quad (\text{B.4})$$

і перетворимо вираз (B.3) таким чином

$$\frac{\partial J}{\partial w_{ij}} = - \sum_{q=1}^Q \Delta_i^q f'(n_i^q) p_j^q, \quad i = \overline{1, S} \quad (\text{B.5})$$

Отримані вирази спрощуються, якщо мережа лінійна. Оскільки $f'(n_i^q) = 1$. В цьому випадку вираз (B.3) приймає вигляд:

$$\frac{\partial J}{\partial w_{ij}} = (t_i^q - a_i^q) p_j^q, \quad i = \overline{1, S}, j = \overline{0, R}. \quad (\text{B.6})$$

Вираз (B.6) покладений в основу алгоритму WH, вживаного для навчання лінійних нейронних мереж.

Лінійні мережі можуть бути навчені і без використання ітераційних методів, а шляхом рішення наступної системи лінійних рівнянь:

$$\sum_{j=0}^R w_{ij} p_j^q = t_i^q, \quad i = \overline{1, S}, \quad q = \overline{1, Q} \quad (\text{B.7})$$

або у векторній формі запису:

$$\mathbf{Wp} = \mathbf{t}, \quad \mathbf{W} = [w_{ij}], \quad \mathbf{p} = [p_j^q], \quad \mathbf{t} = [t_i^q], \quad i = \overline{1, S}, \quad j = \overline{0, R}, \quad q = \overline{1, Q} \quad (\text{B.8})$$

Якщо число невідомих системи (B.7) рівно числу рівнянь, то така система може бути вирішена, наприклад, методом виключення Гауса з вибором головного елемента. Якщо ж число рівнянь перевищує число невідомих, то рішення шукається з використанням методу найменших квадратів.

В.5.2. Навчання багатошарової мережі.

Архітектура багатошарової мережі істотно залежить від вирішуваної задачі. Для лінійних нейронних мереж може бути встановлений зв'язок між сумарною кількістю вагів і зсувів з довжиною повчальної послідовності. Для інших типів мереж число шарів і нейронів в шарі часто визначається досвідом, інтуїцією проектувальника і евристичними правилами.

Навчання мережі включає декілька кроків:

- вибір початкової конфігурації мережі з використанням, наприклад, наступного евристичного правила: кількість нейронів проміжного шару визначається половиною сумарної кількості входів і виходів;
- проведення ряду експериментів з різними конфігураціями мережі і вибір тій, яка дає мінімальне значення функціонала помилки;
- якщо якість навчання недостатньо, слід збільшити число нейронів шаруючи або кількість шарів;
- якщо спостерігається явище перенавчання, слід зменшити число нейронів в шарі або видалити один або декілька шарів.

Нейронні мережі, призначені для вирішення практичних завдань, можуть містити до декількох тисяч параметрів, що настроюються, тому обчислення градієнта може зажадати вельми великих витрат обчислювальних ресурсів. З урахуванням специфіки багатошарових нейронних мереж для них розроблені спеціальні методи розрахунку градієнта, серед яких слід виділити метод зворотного розповсюдження помилки.

В.5.3 Метод зворотного розповсюдження помилки.

Термін "зворотне розповсюдження" відноситься до процесу, за допомогою якого можуть бути обчислені похідні функціонала помилки по параметрах мережі. Цей процес може використовуватися у поєднанні з різними стратегіями оптимізації. Існує багато варіантів і самого алгоритму зворотного розповсюдження. Звернемося до одного з них.

Розглянемо вираз для градієнта критерію якості по вагових коефіцієнтах для вихідного шару M :

$$\frac{dJ}{dw_{ij}^M} = \frac{\partial}{\partial w_{ij}^M} \left(\frac{1}{2} \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM})^2 \right) = \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) \frac{\partial a_k^{qM}}{\partial w_{ij}^M} ; \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}}, \quad (\text{B.9})$$

де S^M - число нейронів в шарі;

a_k^{qM} - k -й елемент вектора виходу шару M для елемента вибірки з номером q .

Правило функціонування шару M :

$$a_k^{qM} = f_M \left(\sum_{i=0}^{S^{M-1}} w_{ij}^M a_i^{q(M-1)} \right), \quad m = \overline{1, S^M} . \quad (\text{B.10})$$

З рівняння (B.10) виходить

$$\frac{\partial a_k^{qM}}{\partial w_{ij}^M} = \begin{cases} 0, & k \neq i \\ f'(n_i^{qM}) a_j^{q(M-1)}, & k = i \end{cases} \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}} \quad (\text{B.11})$$

Після підстановки (B.11) в (B.9) маємо:

$$\frac{dJ}{dw_{ij}^M} = - \sum_{q=1}^Q (t_i^q - a_i^{qM}) f'(n_i^{qM}) a_i^{q(M-1)},$$

Якщо позначити

$$\Delta_i^{qM} = (t_i^q - a_i^{qM}) f'(n_i^{qM}), \quad i = 1, \dots, S^M \quad (\text{B.12})$$

то отримаємо

$$\frac{dJ}{dw_{ij}^M} = - \sum_{q=1}^Q \Delta_i^{qM} a_i^{q(M-1)} ; \quad i = \overline{1, S^M} ; \quad j = \overline{0, S^{M-1}} \quad (\text{B.13})$$

Перейдемо до виведення співвідношень для настройки вагів w_{ij}^{M-1} шару $M-1$

$$\begin{aligned}
\frac{dJ}{dw_{ij}^{M-1}} &= - \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) \frac{\partial n_k^{qM}}{\partial a_i^{q(M-1)}} \frac{\partial a_k^{q(M-1)}}{\partial w_{ij}^{M-1}} a_i^{q(M-1)} = \\
&= - \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) w_k^M \frac{\partial n_k^{qM}}{\partial a_i^{q(M-1)}} f'_{M-1}(n_i^{q(M-1)}) a_j^{q(M-2)} = \\
&= - \sum_{q=1}^Q \Delta_i^{q(M-1)} a_j^{q(M-2)}, \tag{B.14}
\end{aligned}$$

де

$$\Delta_i^{qM} = \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) w_k^M f'_{M-1}(n_i^{q(M-1)}) = \left(\sum_{k=1}^{S^M} \Delta_k^{qM} \right) f'_{M-1}(n_i^{q(M-1)}), \quad i = \overline{1, S^{M-1}},$$

Для шарів $M-2, M-3, \dots, 1$ обчислення часних похідних критерію J по елементах матриць вагових коефіцієнтів виконується аналогічно. У результаті отримуємо наступну загальну формулу:

$$\frac{dJ}{dw_{ij}^{M-1}} = \sum_{q=1}^Q \Delta_i^{q(r-1)} a_j^{q(r-1)}, \quad r = \overline{1, M}, \quad i = \overline{1, S^r}, \quad i = \overline{0, S^{r-1}}, \tag{B.15}$$

де r - номер шару.

$$\Delta_i^{qr} = \left(\sum_{k=1}^{S^{r-1}} \Delta_k^{q(r-1)} w_{ki}^{r+1} \right) f'_r(n_i^{qr}), \quad r = \overline{1, M-1}$$

$$\Delta_i^{qM} = (t_i^q - a_i^{qM}) f'_M(n_i^{qM}), \quad i = \overline{1, S^{M-1}}$$

На рис. В.1 представлена схема обчислень, відповідна виразу (B.15). На схемі символом $*$ позначена операція поелементного множення векторів, а символом $**$ - множення вектора Δ на $\mathbf{a}^T$; символ, що позначає номер елементу вибірки, скорочено опущений.

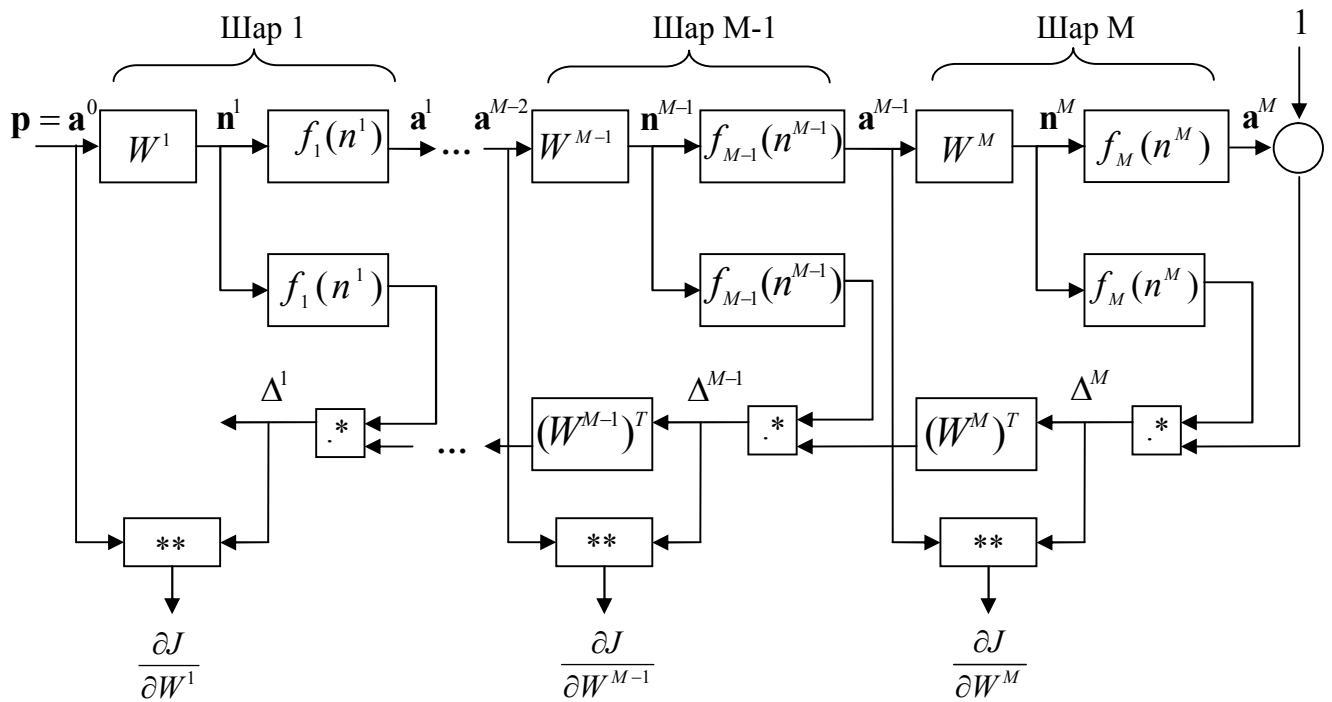


Рисунок В.1 – Схема обчислень часних похідних критерію J по елементах матриць вагових коефіцієнтів

В.5.4 Характеристика методів навчання.

Методи, використовувані при навчанні нейронних мереж, багато в чому аналогічні методам визначення екстремуму функції декілька змінних. У свою чергу, останні діляться на 3 категорії - методи нульового, першого і другого порядку.

У методах нульового порядку для знаходження екстремуму використовується тільки інформація про значення функції в заданих точках.

У методах першого порядку використовується градієнт функціонала помилки по параметрах, що настраюються

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \mathbf{g}_k \quad (\text{B.16})$$

де $\mathbf{x}_k$ - вектор параметрів;

a_k - параметр швидкості навчання;

$\mathbf{g}_k$ - градієнт функціонала, відповідні ітерації з номером k .

Вектор в напрямі, протилежному градієнту, указує напрям найкоротшого спуску по поверхні функціонала помилки. Якщо реалізується рух в цьому напрямі, то помилка зменшуватиметься. Послідовність таких кроків врешті-решт приведе до значень параметрів, що настроюються, що забезпечують мінімум функціонала. Певну трудність тут викликає вибір параметра швидкості навчання a_k . При великому значенні параметра a_k збіжність буде швидкою, але існує небезпека пропустити рішення або піти в неправильному напрямі. Класичним прикладом є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. Залежно від прийнятого алгоритму параметр швидкості навчання може бути постійним або змінним. Правильний вибір цього параметра залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; у разі змінного параметра його значення зменшується у міру наближення до мінімуму функціонала.

У алгоритмах зв'язаного градієнта пошук мінімуму виконується уздовж зв'язаних напрямів, що забезпечує зазвичай швидшу збіжність, ніж при найскорішому спуску. Всі алгоритми зв'язаних градієнтів на першій ітерації починають рух у напрямі антиградієнта

$$\mathbf{p}_0 = -\mathbf{g}_0 \quad (\text{B.17})$$

Тоді напрям наступного руху визначається так, щоб воно було зв'язане з попереднім. Відповідний вираз для нового напрямку руху є комбінацією нового напрямку найскорейшого спуску і попереднього напрямку:

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \mathbf{p}_{k-1}, \quad (\text{B.18})$$

де $\mathbf{p}_k$ - напрям руху

$\mathbf{g}_k$ - градієнт функціонала помилки

β_k - коефіцієнт відповідають ітерації з номером k . Коли напрям спуску визначений, то нове значення вектора параметрів $\mathbf{x}_{k+1}$, що настроюються, обчислюється за формулою

$$\mathbf{x}_{k+1} = \mathbf{x}_k - a_k \mathbf{p}_k \quad (\text{B.19})$$

Методи другого порядку вимагають знання других похідних функціонала помилки. До методів другого порядку відноситься метод Ньютона. Основний крок методу Ньютона визначається по формулі

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \quad (\text{B.20})$$

де $\mathbf{x}_k$ - вектор значень параметрів на k -й ітерації;

$\mathbf{H}$ - матриця других приватних похідних цільової функції, або матриця Гессе;

$\mathbf{g}_k$ - вектор градієнта на k -й ітерації.

У багатьох випадках метод Ньютона сходиться швидше, ніж методи зв'язаного градієнта, але вимагає великих витрат із-за обчислення гессиана. Для того, щоб уникнути обчислення матриці Гессе, пропонуються різні способи її заміни наближеними виразами, що породжує так звані квазиньютонові алгоритми (алгоритм методу січних площин OSS, алгоритм LM Левенберга-Марквардта).

ДОДАТОК Г

ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання нелінійних динамічних систем є отримання їх адекватних математичних моделей, який базується, як правило, на теоретичному і експериментальному аналізі властивостей цих систем. Теоретичний аналіз, заснований на вивченні фізичних і хімічних процесів, які відбуваються в системі, дозволяє отримати математичний опис, наприклад, диференціальних рівнянь. При експериментальному аналізі на основі спостережень вхідних і початкових сигналів системи отримують або її параметричну, або непараметричну модель. Найбільшого розповсюдження придбали параметричні моделі, які вимагають рішення задач структурної і параметричної ідентифікації і що використовують обмежену кількість параметрів. Не дивлячись на величезну кількість робіт, різноманіття видів нелінійностей не дозволяє створити єдину теорію ідентифікації нелінійних систем. Вживаний найчастіше класичний підхід заснований на апроксимації нелінійностей, наприклад рядами Вольтерра, Гаммерштейна, Вінера, поліномами Колмогорова-Габора і ін., проте сфера застосування таких моделей обмежена. Крім того, додаткові труднощі отримання адекватного математичного опису зумовлює наявність в реальних сигналах завад, яка вимагає попередньої фільтрації сигналів.

Розглянемо завдання ідентифікації нелінійного динамічного об'єкту, що описується NARMAX-моделлю.

$$\tilde{y}(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n), k] + \xi(k), \quad (\text{Г.1})$$

де $\tilde{y}(i)$, $u(i)$ – вихідний і вхідний сигнали об'єкту у момент часу i відповідно;

m , n – порядки запізнювання по вихідним і вхідним каналами відповідно;

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

ξ – завада вимірювання вихідного сигналу.

Введемо вектор узагальненого вхідного сигналу розмірністю $(n + m) \times 1$ так, що:

$$x(k) = [\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-m), u(k-1), u(k-2), \dots, u(k-n)]. \quad (\Gamma.2)$$

Тоді рівняння (Г.1) може бути записане таким чином:

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (\Gamma.3)$$

Завдання ідентифікації полягає в оцінюванні функції $f(\cdot)$ по вимірюваннях вхідних $u(k)$ і вихідних $\tilde{y}(k)$ змінних.

Схема ідентифікації нелінійного об'єкту (Г.1) на підставі ШНМ приведена на рис. Г.1.

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (Г.1) сьогодні отримали багат шаровий персептрон (рис. Г.2) і радіально-базисні мережі (РБМ) (рис. Г.3), що використовують апроксимацію нелінійного оператора $f(\cdot)$ нейронною мережею.

Рівняння БП має вигляд

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (\Gamma.4)$$

де q – кількість куль в мережі;

ω^i – вектор вагових параметрів нейронів i -го куль мережі;

$f(\cdot)$ – функція активації i -го шару.

Модель у вигляді радіально базової мережі РБМ, яка використовує для апроксимації $f(\cdot)$ деякі базисні функції $\Phi_i(x)$ може бути представлена таким чином:

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (\Gamma.5)$$

де c_i – вагові коефіцієнти, які підлягають визначенню.

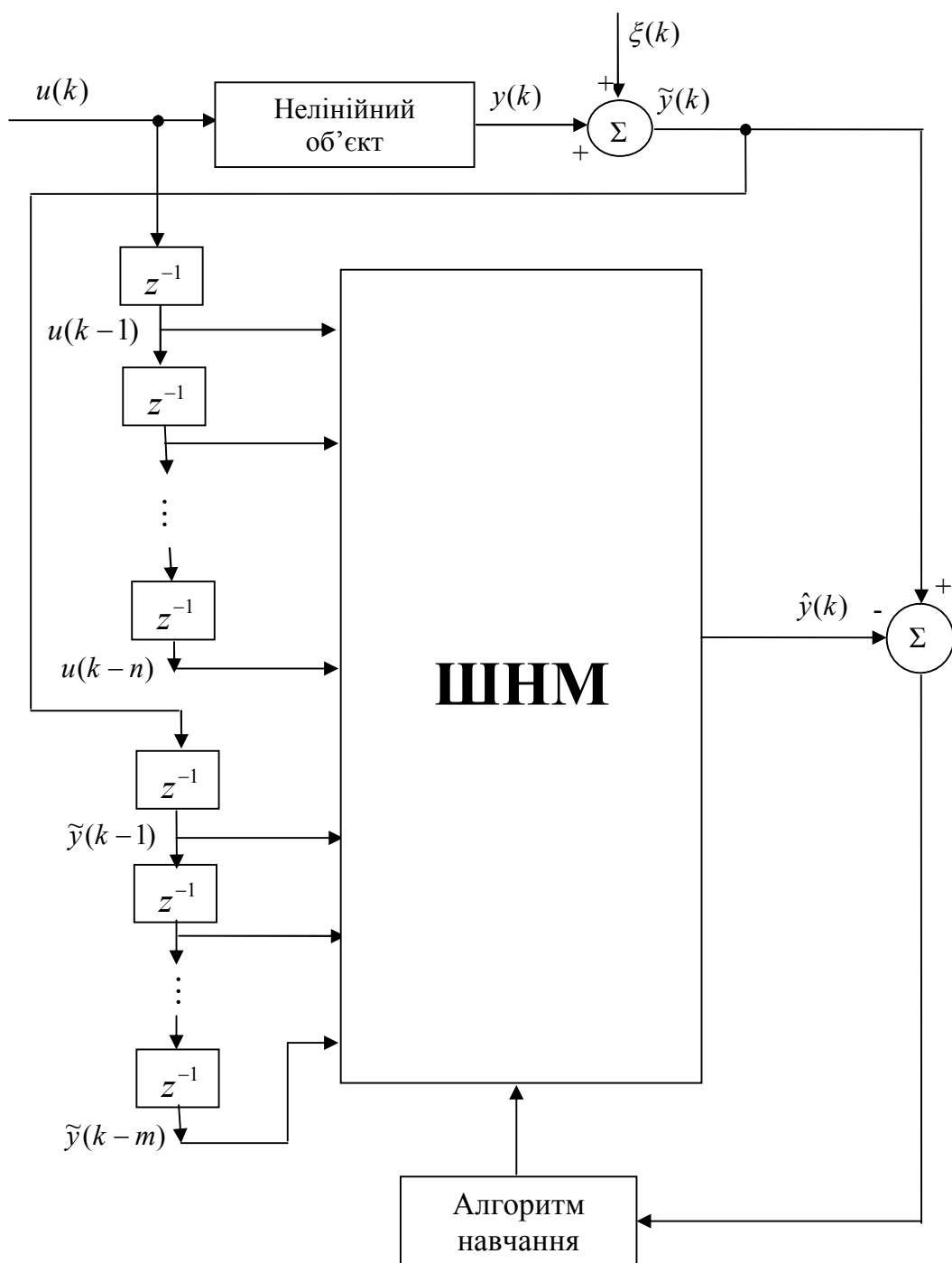


Рисунок Г.1 – Схема ідентифікації динамічного об'єкту

Якщо як базисну функцію $\Phi_i(x)$ вибрати

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (\text{Г.6})$$

де μ_i, σ_i – центри і радіуси базисних функцій відповідно;

$\|\cdot\|$ – евклідова норма, то визначенню підлягає вектор

$$\omega(k) = (c_0(k), c_1(k), \mu_1^T(k), \sigma_1(k), \dots, c_N(k), \mu_N^T(k), \sigma_N(k))^T,$$

який містить всі невідомі параметри мережі (зазвичай приймають $\Phi_0(x) = 1$).

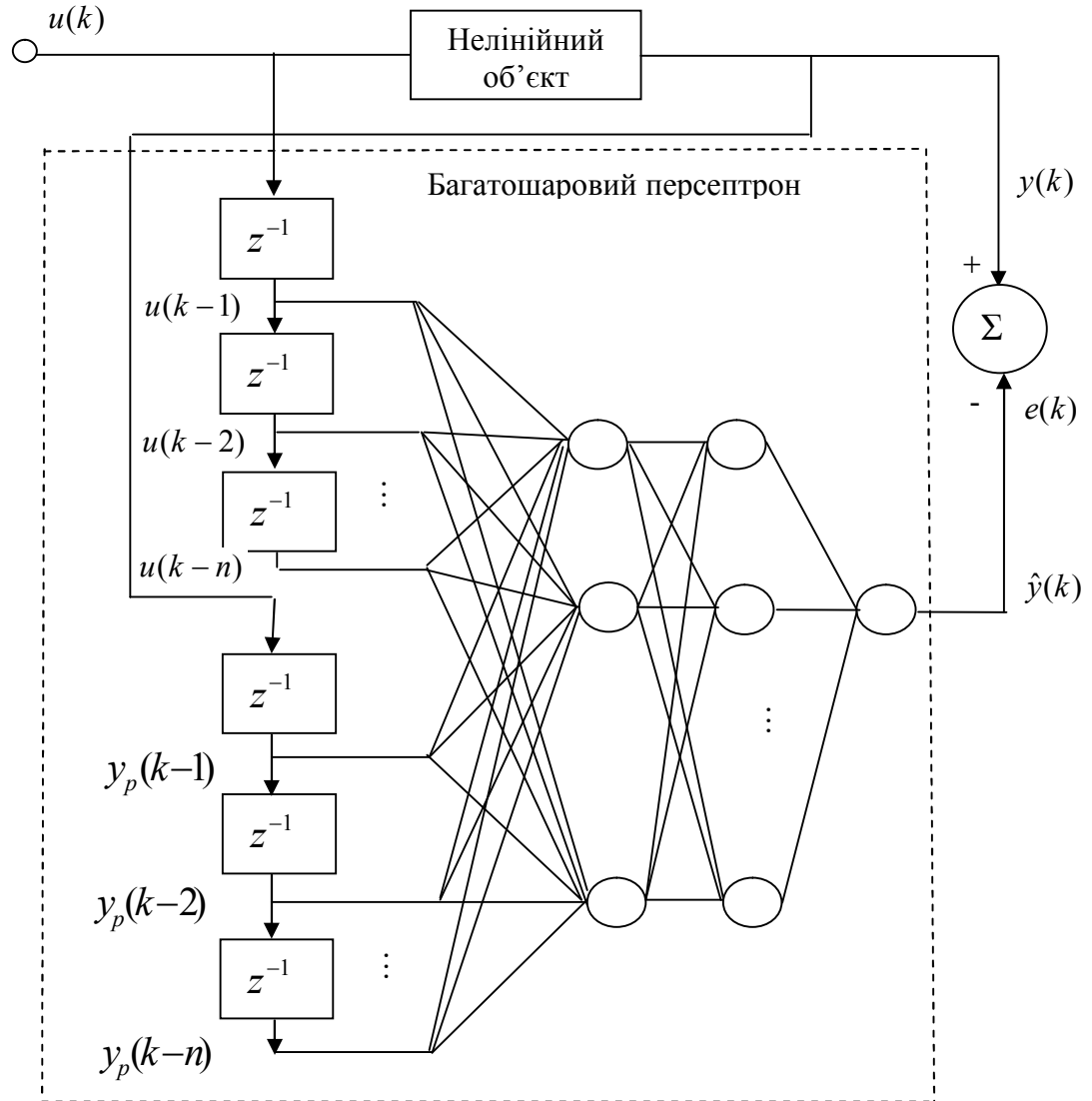


Рис Г.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового перцептрона

Завдання ідентифікації полягає в навчанні мережі, тобто в такому настроюванні її параметрів, які б забезпечували мінімум функціоналу

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (\Gamma.7)$$

Навчання мережі забезпечується за допомогою якого-небудь методу мінімізації функціонала якості.

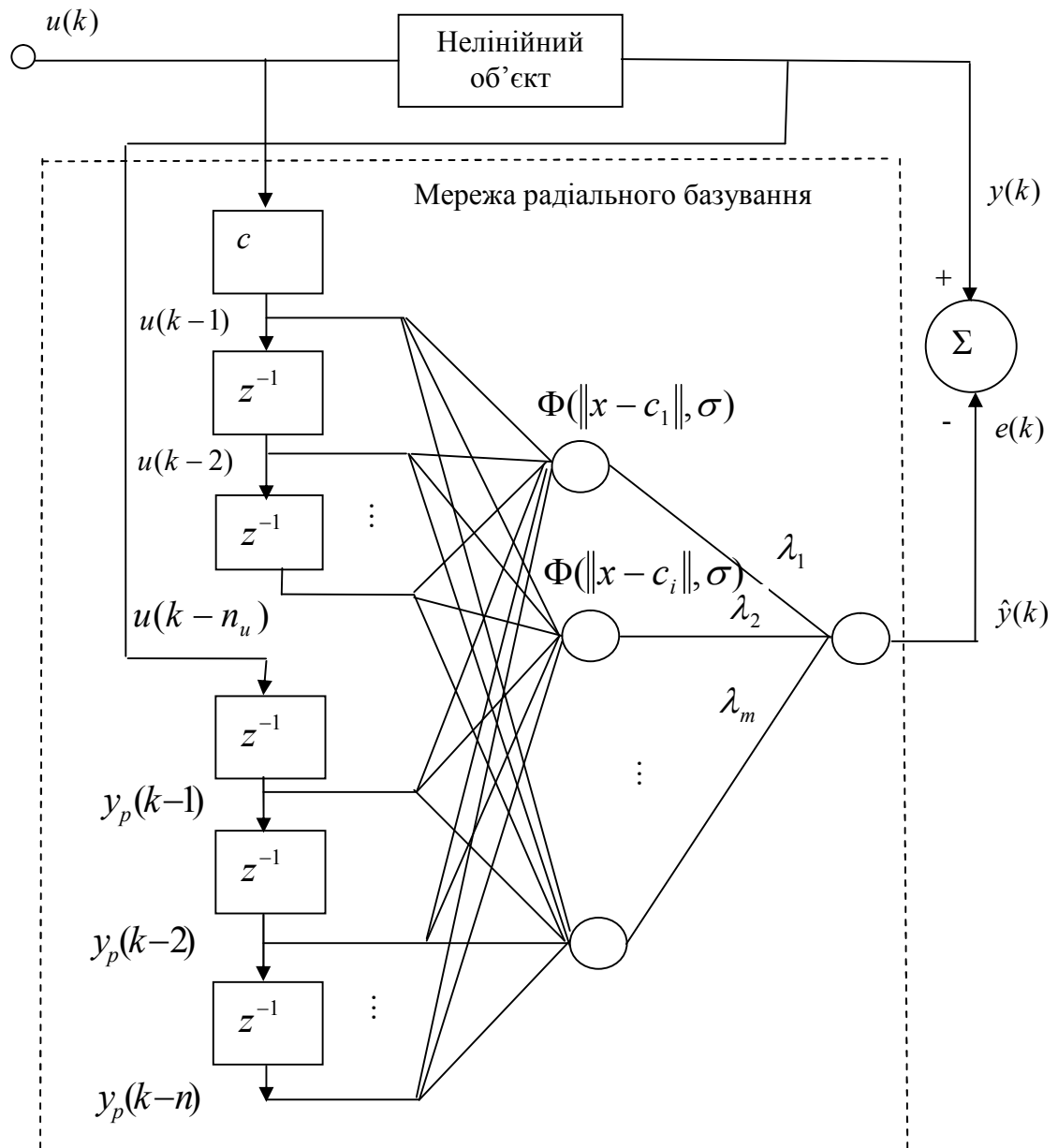


Рисунок Г.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% механізму переміщення візка мостового крану
% 2025 рік

% Вихідні дані
kd=0.42;
CF=1/kd;
In=57;
Mn=In/kd;
ktp=29.9;
kn=0.1091;
kc=0.21;
Tmc=5e-3;
Ta=0.04;
Te=0.025;
Re=0.633;
b12=0;
Jd=1.038;
Jm=2.29;
Js=Jd+Jm;
w0=3;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);

Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);
Uz=24;

% Матриці
A=[-b12/Jm 1/Jm b12/Jm zeros(1,5);
-c12 0 c12 zeros(1,5);
b12/Jd -1/Jd -b12/Jd CF/Jd zeros(1,4);
0 0 -CF/(Re*Te) -1/Te 1/(Re*Te) 0 0 0;
0 0 0 (-kc*kpc*ktp)/Tmc -1/Tmc ktp/Tmc (kpe*kpc*ktp)/Tmc -
(kpe*kpc*ktp)/Tmc;
0 0 0 -kic*kc 0 0 kpe*kic -kpe*kic;
zeros(1,6) -1/Ta 0;
0 0 kn/(kd*Ta) zeros(1,4) -1/Ta ];
B=[0 -1/Jm;
zeros(5,2);
1/Ta 0;
0 0];
C=zeros(4,8);
C(1,1)=1; C(2,2)=1;C(3,3)=1;C(4,4)=1;%C(5,5)=1; C(6,6)=1;
C(7,7)=1; C(8,8)=1;
D=zeros(4,2);
```