

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

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

(тема кваліфікаційної роботи)

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

(код і найменування спеціальності)

_____/ Євгеній ТИХИЙ
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

Харків – 2025

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

д.т.н., проф. Геннадій КАНЮК
(науковий ступінь, вчене звання, ім'я, прізвище)
« » 2025р.

ЗАВДАННЯ
на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Євгеній ТИХИЙ
(ім'я, прізвище)

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

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 Теоретична продуктивність по рихлій масі $Q=3900\text{м}^3\cdot\text{год}$; діаметр роторного колеса $D_p=12\text{ м}$; бічне зусилля різання $F_6=12\cdot10^4\text{ Н}$; питоме зусилля копання $k_f=1,1\cdot10^6\text{ кг/м}^2$; радіус різання $R_0=37\text{ м}$; кутова швидкість повороту платформи $\omega_{пл}=0,6\cdot10^{-2}\text{ с}^{-1}$; маса поворотної платформи $m_{пл}=1000\cdot10^3\text{ кг}$; частота обертання ротора $\omega_p=5\text{ рад/хв}$; момент інерції поворотної платформи $J_{пл}=88\cdot10^7\text{ кг}\cdot\text{м}^2$; діаметр роликового круга $D_{кр}=14\text{ м}$. Перерегулювання не більше 10%, час регулювання до 5 с.

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

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

6. Консультант:

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

7. Дата видачі завдання “ 9 ” вересня 2025 р.

Керівник

(підпис)

Геннадій КАНЮК.
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Євгеній ТИХИЙ
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про вико- нання за- вдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму обертання колеса роторного екскаватора.	09.09.2025- 18.09.2025	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження статичних і динамічних характеристик системи.	19.09.2025- 15.10.2025	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ПЕОМ.	16.10.2025- 05.11.2025	
4	Синтез нейромережової системи управління на ПЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2025- 25.11.2025	
5	Оформлення роботи	26.11.2025- 05.12.2025	

Студент (ка)

(підпис)

Євгеній ТИХИЙ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 205 сторінок, 67 рисунок, 2 таблиці, 92 використаних джерела, 4 додатки.

Метою магістерської роботи є розробка методів синтезу системи керування електроприводом механізму обертання робочого колеса роторного екскаватора з метою підвищення якості його функціонування шляхом упровадження нейромережевих регуляторів.

Параметри системи автоматичного керування визначено шляхом розрахунків на основі фундаментальних положень теорії автоматичного керування. Методи математичного моделювання застосовано для формування моделей динаміки багатомасової електромеханічної системи. Використання апарату штучних нейронних мереж дало можливість виконати синтез нейромережевого регулятора для керування двомасовим електроприводом. За допомогою інструментів комп'ютерного моделювання, реалізованих у середовищі MATLAB, досліджено динамічні властивості розробленої нейромережевої системи керування.

У ході роботи науково підтверджено доцільність використання нейромережевих технологій для підвищення ефективності керування багатомасовими електромеханічними системами механізму обертання колеса роторного екскаватора. Розроблено структуру системи регулювання координат електроприводу та виконано розрахунок її основних параметрів. Проведено комп'ютерне моделювання функціонування системи.

Сформовано математичну модель двомасової системи з урахуванням пружних властивостей механічних передач, на основі якої визначено динамічні характеристики об'єкта керування.

Запропоновано архітектуру нейромережевої системи керування та виконано синтез нейрорегулятора типу NN Predictive Controller у середовищі MATLAB. Проведене моделювання підтвердило, що розроблена нейромереже-

ва система забезпечує високий рівень якості динамічних процесів і відповідає вимогам до сучасних електроприводів.

Результати магістерської роботи впроваджені в освітній процес кафедри АМЕТ ННІ «УПА» ХНУ імені В.Н. Каразіна та використовуються при викладанні дисциплін, виконанні курсових проєктів і проведенні практичних занять.

Основні положення та наукові результати роботи були представлені й обговорені на науково-практичній конференції студентів ННІ «УПА» ХНУ імені В.Н. Каразіна (Харків, 2025 р.).

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

ABSTRACT

An explanatory message consists of: 205 pages, 67 figures, 4 tables, 92 used sources, 4 additions.

The purpose of the master's thesis is to develop methods for synthesizing the control system for the electric drive of the rotor wheel rotation mechanism of a rotary excavator in order to improve the quality of its functioning by introducing neural network controllers.

The parameters of the automatic control system were determined by calculations based on the fundamental principles of the theory of automatic control. Mathematical modeling methods were used to form models of the dynamics of a multi-mass electromechanical system. The use of artificial neural networks made it possible to synthesize a neural network controller for controlling a two-mass electric drive. Using computer modeling tools implemented in the MATLAB environment, the dynamic properties of the developed neural network control system were investigated.

During the work, the feasibility of using neural network technologies to improve the efficiency of controlling multi-mass electromechanical systems of the rotor wheel rotation mechanism of a rotary excavator was scientifically confirmed. The structure of the electric drive coordinate control system was developed and its main parameters were calculated. Computer modeling of the system's functioning was carried out. A mathematical model of a two-mass system was formed taking into account the elastic properties of mechanical transmissions, on the basis of which the dynamic characteristics of the control object were determined.

The architecture of a neural network control system was proposed and a neural regulator of the NN Predictive Controller type was synthesized in the MATLAB environment. The modeling confirmed that the developed neural network system provides a high level of quality of dynamic processes and meets the requirements for modern electric drives.

The results of the master's thesis were implemented in the educational process of the AMET Department of the Scientific Research Institute "UIPA" of the V.N.

Karazin KhNU and are used in teaching disciplines, performing course projects and conducting practical classes.

The main provisions and scientific results of the work were presented and discussed 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, ROTARY EXCAVATOR, ELECTRIC DRIVE OF THE MECHANISM OF ROTATION OF THE WHEEL, CONTROL SYSTEM WITH SUMMARIZING AMPLIFIER, ONE-MASS AND TWO-MASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

P_n – потужність, що відповідає номінальному режиму роботи;

U_n – напруга, встановлена як номінальна;

I_n – струм у номінальних умовах;

M_n – номінальний момент;

n_n – швидкість обертання при номінальному навантаженні;

Φ_n – магнітний потік у номінальному режимі;

R – активний опір елемента;

L – величина індуктивності;

J – момент інерції вузла;

k_d – коефіцієнт підсилення електричного двигуна;

T_e – електромагнітна стала часу електропривода;

T_m – електромеханічна стала часу електропривода;

$W(p)$ – передавальна функція;

c – коефіцієнт жорсткості;

β – коефіцієнт внутрішнього в'язкого тертя;

A – матриця, що описує стан системи;

B – матриця, яка визначає вплив керуючих дій;

C – матриця виходу;

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

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

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

САУ – система автоматичного управління;

М – електродвигун.

ЗМІСТ

ВСТУП	10
РОЗДІЛ 1. ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ В СИСТЕМАХ УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Тенденції розвитку теорії та практичного застосування нейронних мереж	14
1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж	17
1.3 Нейрокерування динамічними об'єктами	23
1.4 Огляд структур нейромережевих систем управління та вибір архітектури.....	25
1.5 Загальні відомості про роторні екскаватори	32
1.6 Обґрунтування вибору системи електроприводу роторного колеса екскаватора	36
1.7 Формулювання наукової проблеми, обґрунтування актуальності дослі- дження та визначення шляхів її вирішення.....	38
ВИСНОВКИ ДО РОЗДІЛУ 1	42
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ОБЕРТАННЯ КОЛЕСА РОТОРНОГО ЕКСКАВАТОРА ...	45
2.1 Основні технічні характеристики роторного екскаватора та вимоги до системи управління.....	45
2.2 Розрахунок потужності та обґрунтування вибору електродвигуна приводу обертання роторного колеса	47
2.3 Вибір основного електротехнічного обладнання системи електроприводу	49
2.4 Розрахунок і побудова статичних характеристик розімкненої та замкненої системи	50
2.5 Аналіз динамічних характеристик системи	61
ВИСНОВКИ ДО РОЗДІЛУ 2	80
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ.....	81

3.1 Математична модель двомасової системи	81
3.2 Розрахунок перехідних процесів у двомасовій системі.....	86
ВИСНОВКИ ДО РОЗДІЛУ 3	93
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	94
4.1 Регулятори, реалізовані засобами пакета Neural Network Toolbox у середовищі MATLAB	94
4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller	96
4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB	99
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	144
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	146
ВИСНОВКИ ДО РОЗДІЛУ 4	152
ВИСНОВКИ	154
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	157
ДОДАТОК А. БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ.....	166
А.1 Структура штучного нейрона	166
А.2 Одношарові нейронні мережі	167
А.3 Багатошарові нейронні мережі	170
А.4 Здібності апроксимації і узагальнення багатошарових мереж.....	171
ДОДАТОК Б. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ.....	174
Б.1 Завдання навчання	174
Б.2 Процедури адаптації і навчання.....	175
Б.3 Явище перенавчання	177
Б.4 Властивість узагальнення	180
Б.5 Вибір методу навчання нейронної мережі	182
ДОДАТОК В. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ	191
В.1 Алгоритм нейроуправління з прогнозом	191
В.2 Структура нейронної мережі для моделювання об'єкту	198
В.3 Прогноз з використанням нейронної мережі	202
В.4 Похідні рівнянь нейронної мережі	203
ДОДАТОК Г. ФАЙЛ ВИХІДНИХ ДАНИХ	205

ВСТУП

Актуальність теми. Гірничодобувна промисловість традиційно посідає ключове місце в структурі національної економіки, забезпечуючи стратегічні потреби країни у сировинних ресурсах. Підвищення продуктивності цієї галузі є одним з найважливіших чинників зміцнення економічного потенціалу держави. Саме тому на підприємствах відкритих гірничих робіт нині відбувається активне оновлення машинно-технічного парку, спрямоване на впровадження сучасних високоефективних технологій.

Однією з базових машин технологічного комплексу відкритого видобутку корисних копалин є роторний екскаватор. Останні десятиліття позначились значним розвитком конструкцій роторних екскаваторів, їх електроприводів та систем керування. Накопичений досвід проектування, випуску та модернізації електроприводів основних і допоміжних механізмів дозволяє підвищувати продуктивність і надійність цих машин. Проте вирішальним чинником ефективного використання екскаватора залишається рівень розвитку систем управління електроприводами. Саме система керування визначає здатність механізмів працювати в оптимальних режимах, забезпечувати необхідну швидкодію, точність, стійкість та надійність функціонування в умовах інтенсивних навантажень.

Постійне зростання вимог до техніко-економічних показників електроприводів обумовлює потребу у використанні удосконалених підходів до побудови автоматичних систем керування. Значного поширення набули сучасні методи робастного та адаптивного регулювання, здатні забезпечувати стійкість системи при зміні параметрів об'єкта та впливі зовнішніх збурень. Проте задачі синтезу адаптивних регуляторів для складних, нелінійних і змінних у часі об'єктів і досі залишаються повністю не вирішеними. Методи робастного управління також мають обмеження – вони ефективні переважно для лінійних або спеціально структурованих нелінійних систем. Тому універсальних рішень, здатних забезпечити високу якість керування у всіх режимах, досі не існує.

У цьому контексті все більшого значення набувають штучні нейронні ме-

режі (ШНМ), які зарекомендували себе як ефективний інструмент управління складними динамічними об'єктами. Їх ключові переваги – здатність до навчання, швидке узагальнення інформації, апарат нелінійної апроксимації та можливість адаптування до змін умов роботи. Нейронні мережі можуть відтворювати складні нелінійні залежності між входами та виходами, що робить їх надзвичайно ефективними для управління об'єктами зі змінною структурою та параметрами. Розвиток алгоритмів навчання – як градієнтних, так і їх модифікованих варіантів – значно розширив можливості побудови нейромережових регуляторів.

Таким чином, впровадження нейромережових технологій у системи керування електроприводами роторних екскаваторів відкриває широкі перспективи для створення швидкодіючих, адаптивних та високоточних систем регулювання. Це робить обрану тему дослідження актуальною та важливою як у теоретичному, так і в практичному аспектах.

Мета та завдання дослідження. Метою магістерської роботи є розроблення методів синтезу системи керування електроприводом механізму обертання колеса роторного екскаватора на основі нейромережового регулятора, що забезпечує підвищені показники якості та ефективності функціонування.

Для досягнення поставленої мети необхідно вирішити такі науково-технічні задачі:

1. Проаналізувати сучасні системи керування та оцінити перспективи застосування нейромережових підходів у задачах регулювання.
2. Розробити математичну модель динаміки електромеханічної системи з урахуванням пружних зв'язків її механічних елементів.
3. Побудувати структурну схему системи управління з використанням нейромережового регулятора.
4. Виконати синтез нейромережової системи управління на основі регулятора прогнозного типу (NN Predictive Controller), який забезпечує оптимальні динамічні показники.
5. Провести комп'ютерне моделювання синтезованої системи та здійснити аналіз отриманих результатів.

Об’єкт дослідження – динамічні режими системи керування механізмом обертання колеса роторного екскаватора з урахуванням пружних зв’язків у механічній частині.

Предмет дослідження – математичні моделі та методи синтезу нейромережевої системи керування двомасовою електромеханічною системою механізму обертання.

Методи дослідження. У роботі використано комплекс сучасних наукових підходів і методик:

- положення класичної та сучасної теорії автоматичного керування;
- методи математичного моделювання для побудови динамічних моделей системи;
- теоретичний апарат штучних нейронних мереж для створення моделі об’єкта управління та розроблення алгоритму її навчання;
- методи нейроуправління – для синтезу регулятора на основі прогнозної моделі;
- імітаційне моделювання у середовищі MATLAB/Simulink для оцінки ефективності функціонування розробленої системи.

Наукова новизна. У процесі виконання магістерської кваліфікаційної роботи отримано такі нові наукові результати:

1. Вперше обґрунтовано доцільність застосування нейромережевих підходів для керування багатомасовою електромеханічною системою механізму обертання роторного екскаватора.
2. Вперше розроблено та синтезовано нейромережеву систему управління механізмом обертання з урахуванням пружних елементів, яка забезпечує високий рівень якості регулювання.
3. Запропоновано математичну модель об’єкта управління у вигляді багат шарового персептрона, що дозволяє описувати нелінійні та пружні властивості системи.
4. Розширено наукові підходи до моделювання об’єктів керування зі складними кінематичними зв’язками у формі двомасових електромеханічних систем.

5. Удосконалено методи синтезу нейромережевих структур прогнозного типу для управління складними динамічними об'єктами.

Практичне значення результатів. Практична цінність роботи полягає у створенні нейромережевої системи керування двомасовою електромеханічною системою механізму обертання колеса роторного екскаватора, здатної забезпечити високу точність регулювання та покращені динамічні характеристики.

Матеріали роботи застосовуються в навчальному процесі кафедри АМЕТ ННІ «УІПА» ХНУ ім. В.Н. Каразіна та використовуються під час викладання профільних дисциплін, проведення практичних занять і виконання курсових проєктів.

Особистий внесок здобувача. Усі результати, висунуті на захист, отримані автором особисто, зокрема:

- синтез системи підлеглого керування координатами електроприводу;
- побудова моделі двомасової системи з урахуванням пружних властивостей;
- розроблення структурної схеми нейромережевої системи управління;
- синтез нейрорегулятора NN Predictive Controller у середовищі MATLAB;
- моделювання системи та аналіз результатів;

Апробація результатів. Основні наукові положення та отримані результати були представлені на науково-практичній конференції студентів ННІ «УІПА» ХНУ ім. В.Н. Каразіна (Харків, 2025 р.).

Структура та обсяг роботи. Робота включає: вступ, чотири розділи основної частини, висновки, список використаних джерел та додатки. Загальний обсяг – 205 сторінок. Матеріал містить 40 рисунків у тексті, 27 рисунків на 25 окремих сторінках, 2 таблиці та 4 додатки обсягом 40 сторінок. Список літератури складається з 92 найменувань на 9 сторінках.

РОЗДІЛ 1

ЗАСТОСУВАННЯ НЕЙРОМЕРЕЖЕВИХ ТЕХНОЛОГІЙ В СИСТЕМАХ УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Тенденції розвитку теорії та практичного застосування нейронних мереж

Штучні нейронні мережі (ШНМ) являють собою особливий клас систем обробки інформації, які працюють за принципами колективної взаємодії великої кількості простих обчислювальних елементів – нейронів. Ці елементи мають відносно примітивну внутрішню структуру, однак завдяки численним зв'язкам між собою вони здатні реалізовувати досить складні функції – розпізнавання образів, прогнозування, апроксимацію нелінійних залежностей тощо. Кожен нейрон приймає на вхід певний набір сигналів, виконує над ними елементарні перетворення та передає результат іншим нейронам. Взаємодія великої кількості таких елементів формує складну нелінійну систему, яка здатна навчатися та адаптуватися до зовнішніх умов.

Історія формування цього наукового напрямку бере початок на початку ХХ ст., однак лише після подолання ряду фундаментальних теоретичних обмежень та появи високопродуктивних електронно-обчислювальних машин у 90-х роках нейронні мережі отримали активне промислове застосування. Термін «штучні» використовується для відмежування обчислювальної моделі від реальних біологічних нейронних структур, хоча натхненням для створення таких систем послужило саме прагнення зрозуміти механізми функціонування людського мозку. Допоки контекст є зрозумілим, слово «штучний» зазвичай опускається, а у випадках, коли акцент робиться саме на обчислювальних можливостях мереж, використовується термін «конекціоністські моделі» або «конекціоністи».

Попри те, що існують апаратні нейропроцесори та спеціалізовані мікросхеми для реалізації ШНМ, більшість сучасних досліджень здійснюється шля-

хом програмної симуляції. Цей підхід дозволяє значно дешевше, швидше та гнучкіше експериментувати з новими архітектурами і методами навчання. Головна відмінність нейронних мереж від класичних програм полягає в тому, що їх поведінка задається не жорстко прописаними інструкціями, а результатом процесу навчання на певній вибірці даних. Фактично вони не програмуються у традиційному сенсі – мережу навчають, а вона самостійно формує внутрішні правила перетворення інформації.

У реальних задачах нейронні мережі особливо корисні там, де неможливо або надто складно побудувати точну математичну модель. Наприклад, аналітики на ринку нерухомості можуть інтуїтивно оцінювати фактори, які впливають на вартість об'єктів, але ці фактори важко формалізувати математично. У таких випадках ШНМ легко засвоюють приховані залежності із прикладів і поступово підвищують точність роботи в умовах, що змінюються.

На сучасному етапі розвитку науки обсяг практичних застосувань штучних нейронних мереж стрімко розширюється [13-42]. Вони використовуються у таких сферах:

1. Космічна та авіаційна техніка. Реалізація інтелектуальних автопілотів, виявлення несправностей, оптимізація траєкторій руху, системи керування повітряним трафіком.
2. Автомобілебудування. Розробка адаптивних систем керування транспортом, інтелектуальних помічників водія.
3. Фінансово-банківська сфера. Автоматизація аналізу документів, оцінювання кредитних ризиків і платоспроможності клієнтів.
4. Військові системи. Обробка радіолокаційних і оптичних сигналів, класифікація об'єктів, управління озброєнням.
5. Мікроелектроніка та ІТ. Тестування мікросхем, синтез мовлення, реалізація систем технічного зору.
6. Фінансові ринки. Прогнозування зміни вартості активів, аналіз поведінки валют, оцінювання інвестиційних ризиків.

7. Страхування. Формування стратегій страхування з мінімальним ризиком.
8. Промислове виробництво. Контроль якості продукції, діагностика стану устаткування, оптимізація технологічних процесів.
9. Медицина. Ідентифікація клітин, діагностика захворювань, автоматизація аналізу медичних зображень, планування операцій.
10. Нафтогазова галузь. Передбачення можливих родовищ, аналіз геологічних даних.
11. Робототехніка. Побудова інтелектуальних контролерів і систем технічного зору.
12. Системи передавання інформації. Розпізнавання голосу, синхронний переклад, стискання даних.

Дуже важливо, що останніми роками нейронні мережі активно впроваджуються у автоматизовані системи керування. Традиційні лінійні методи синтезу регуляторів, наприклад модальне управління, теоретично дозволяють досягти високих показників якості, але їх практична реалізація часто ускладнена технічними обмеженнями. Зокрема:

- необхідність вимірювати координати, які або важко, або неможливо виміряти безпосередньо (наприклад, пружні моменти);
- ситуації, коли параметри регулятора, визначені теоретично, неможливо реалізувати через обмеження силової частини електроприводу.

У зв'язку з цим природною стає ідея замінити лінійні регулятори на нелінійні нейромережеві структури, які можуть формувати необхідні керуючі дії навіть за умов неповного вимірювання координат та наявності суттєвих нелінійностей у динаміці приводу. Нейрорегулятор, реалізований у вигляді прямонаправленої нейронної мережі, здатен забезпечити адаптивність і підвищену точність регулювання.

У додатках А–В наведено теоретичні основи, які необхідні для виконання синтезу нейромережевої системи управління. У додатку А описано структуру штучного нейрона, принципи побудови багатошарових мереж та їх основні вла-

стивості. Додаток Б присвячений огляду методів навчання ШНМ і обґрунтуванню вибору алгоритму, що застосовано в роботі. У додатку В розглянуто ефективну реалізацію узагальненого методу управління з прогнозуванням на основі нейромережевої моделі об'єкта, що стало підґрунтям для побудови нейрорегулятора типу NN Predictive Controller, який використано при створенні нейромережевої системи управління у даній роботі.

1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж

Одним із ключових етапів під час розв'язання завдань автоматичного керування, прогнозування властивостей та аналізу поведінки нелінійних динамічних систем є побудова їх адекватних математичних моделей. Формування таких моделей зазвичай ґрунтується на результатах теоретичного опрацювання та експериментального дослідження об'єкта.

Теоретичне моделювання, яке спирається на детальне вивчення фізичних, механічних, електричних чи хімічних процесів, що відбуваються в системі, дає можливість сформулювати її формальний опис у вигляді диференціальних рівнянь. Експериментальні методики передбачають отримання моделі за даними спостереження входних та вихідних сигналів, що дозволяє сформулювати параметричну або непараметричну модель об'єкта.

Найбільш широке застосування отримали саме параметричні моделі, які базуються на визначенні певної структури та обмеженої кількості параметрів, однак вимагають розв'язання задач структурної та параметричної ідентифікації. Попри значний обсяг наукових результатів у цій галузі, різноманітність проявів нелінійності ускладнює створення універсального підходу до ідентифікації складних систем.

Класичні підходи традиційно спираються на апроксимацію нелінійних залежностей різного типу – наприклад, за допомогою рядів Вольтерри, моделей Гаммерштейна–Вінера, поліномів Колмогорова–Габора та багатьох інших методів.

Однак сфера їх застосування зазвичай обмежена конкретним видом нелінійності й умовами роботи об'єкта. Додатковою складністю є наявність шумів у реальних вимірювальних сигналах, що потребує попередньої їх фільтрації.

У подальшому розглянемо задачу ідентифікації нелінійного динамічного об'єкта, який описується NARMAX-моделлю:

$$\tilde{y}(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n), k] + \xi(k), \quad (1.1)$$

де $\tilde{y}(i)$, $u(i)$ – вихідний і вхідний сигнали в момент часу i ;

m , n – порядки запізнювання по вихіді і вхіді;

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

ξ – завада вимірювання.

Введемо вектор узагальненого входу розмірності $(n+m) \times 1$:

$$x(k) = [\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-m), u(k-1), u(k-2), \dots, u(k-n)]. \quad (1.2)$$

Тоді рівняння (1.1) можна подати у вигляді

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (1.3)$$

Таким чином, ідентифікація об'єкта зводиться до наближення невідомої функції $f(\cdot)$ за вимірними вибірками вхідних $u(k)$ і вихідних $\tilde{y}(k)$ змінних.

Структурна схема ідентифікації нелінійного об'єкта (1.1) на основі штучної нейронної мережі наведена на рис. 1.1.

Серед сучасних моделей нейромереж найбільш поширеними у задачах ідентифікації нелінійних динамічних систем є багатoshаровий персептрон (БП) (рис. 1.2) та радіально-базисні мережі (РБМ) (рис. 1.3). Вони забезпечують апроксимацію операторної функції $f(\cdot)$ із високою точністю навіть за складних нелінійних залежностей.

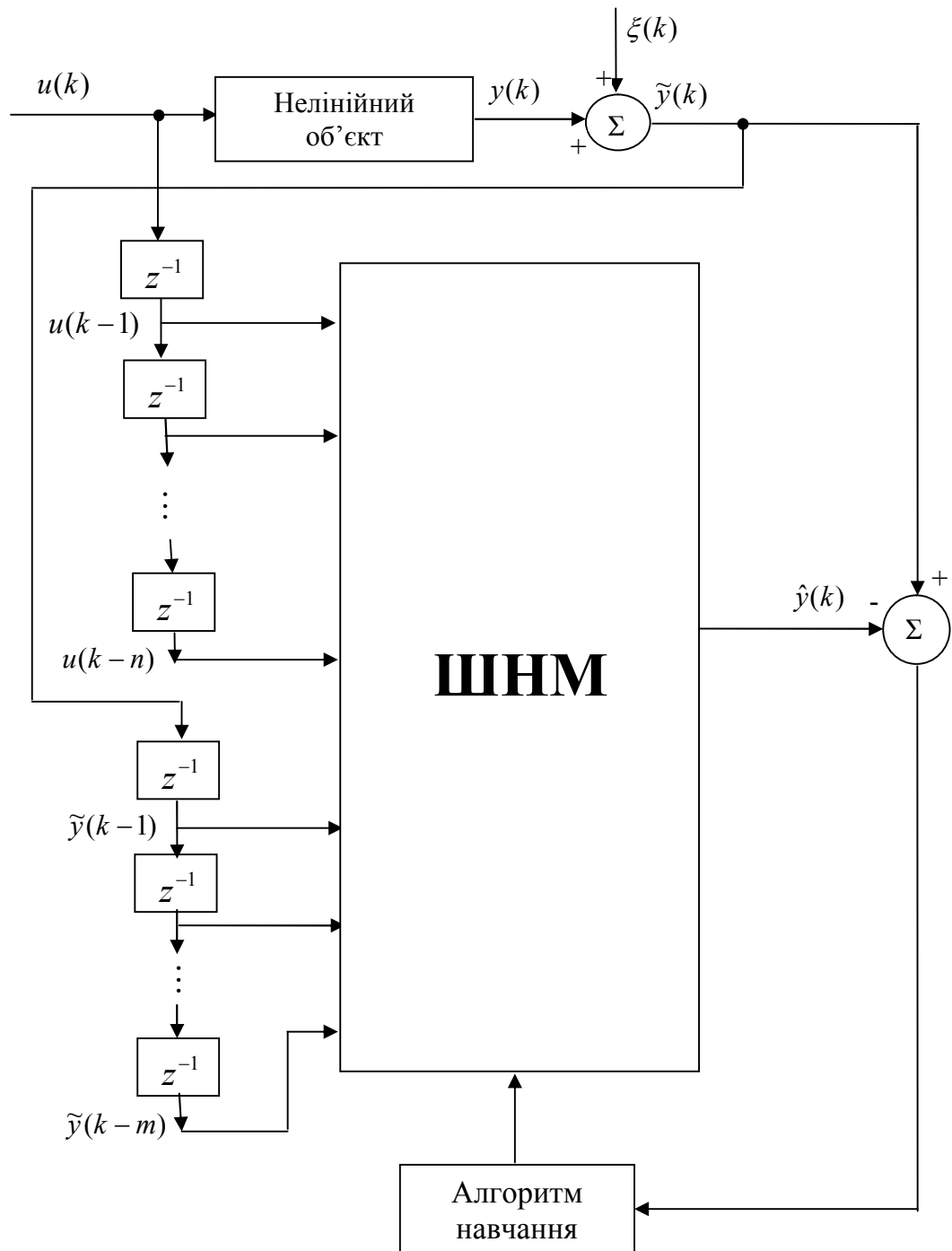


Рисунок 1.1 – Схема ідентифікації динамічного об'єкту

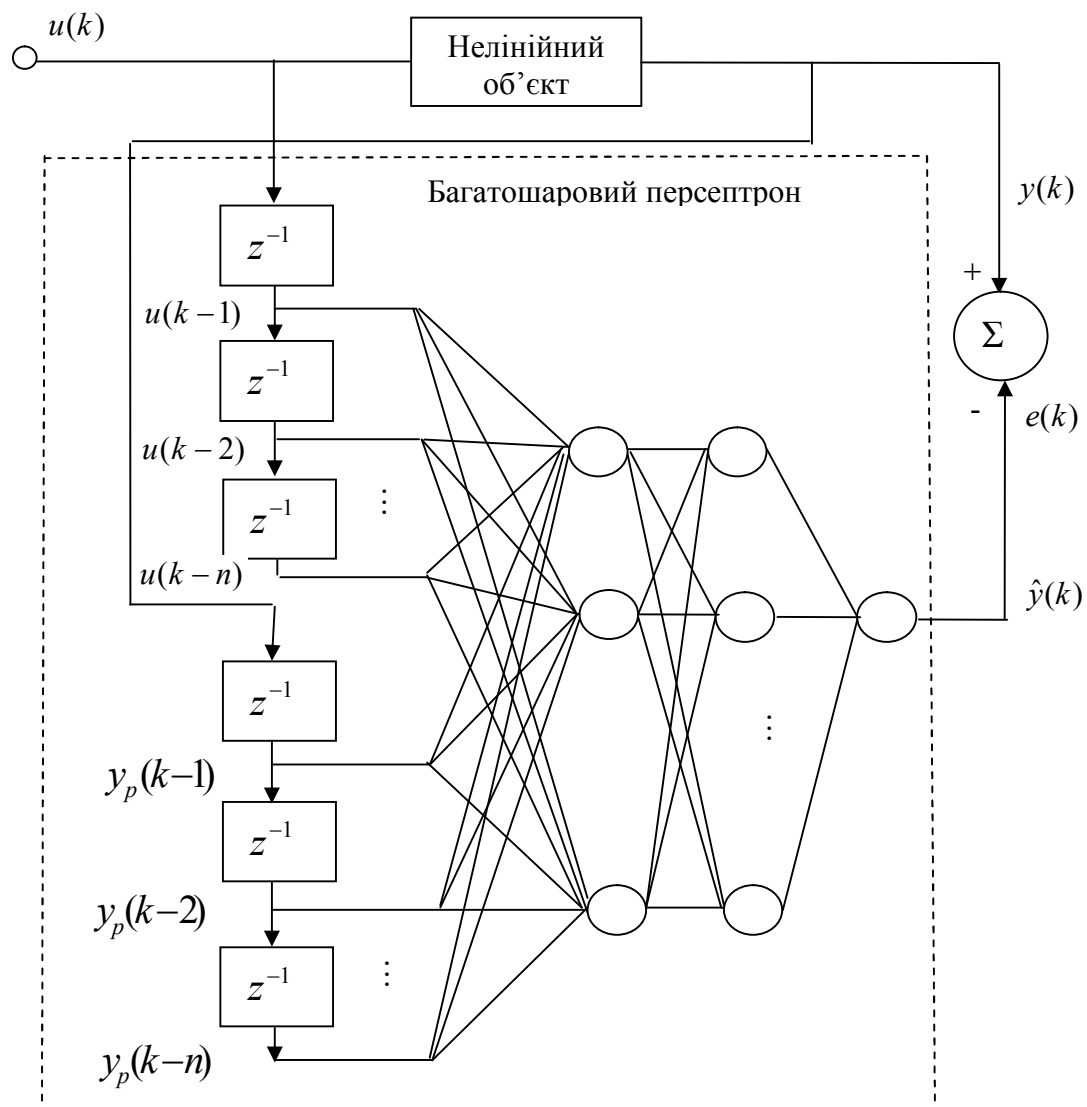


Рис 1.2 – Ідентифікація динамічного об'єкту з використанням багатошарового персептрона

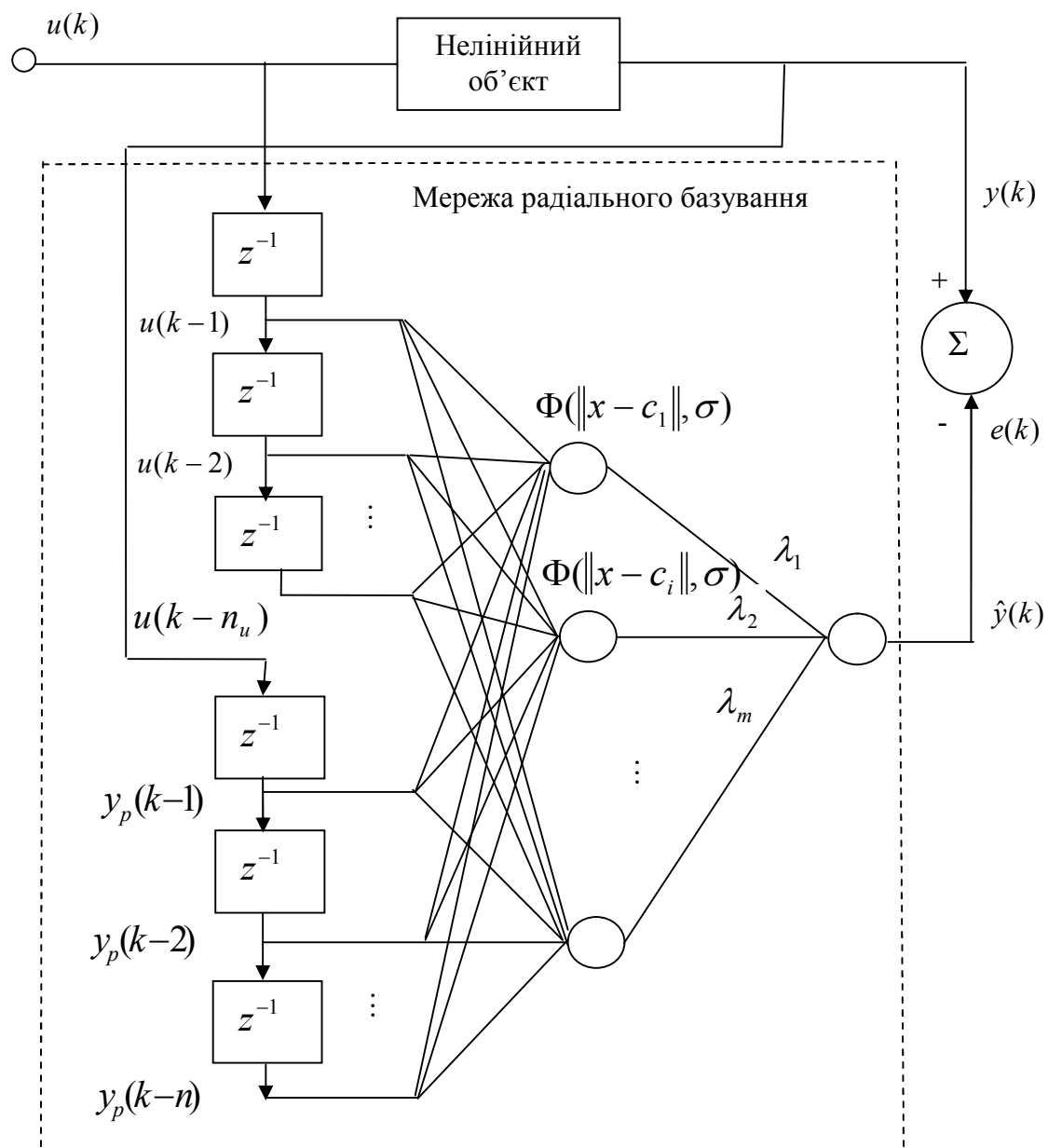


Рисунок 1.3 – Ідентифікація динамічного об'єкту з використанням РБМ

Математичний опис БП має вигляд:

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (1.4)$$

де q – число шарів в мережі;

ω^i – вектор ваг нейронів i -го шару;

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

РБМ описується моделлю

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

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (1.5)$$

де $\Phi_i(x)$ - вибрані базисні функції;

c_i – вагові коефіцієнти, що визначаються при ідентифікації.

При використанні гаусівських базисних функцій

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (1.6)$$

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

$\|\cdot\|$ – евклідова норма, визначається вектор

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

Цей вектор включає всі невідомі параметри мережі (зазвичай приймають $\Phi_0(x) = 1$).

Процес ідентифікації полягає в навчанні мережі шляхом мінімізації функціоналу помилки

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (1.7)$$

1.3 Нейрокерування динамічними об'єктами

Проектування ефективних систем автоматичного керування реальними технічними об'єктами ґрунтується на використанні їх математичних моделей. Вибір типу моделі та рівня її деталізації визначає структуру і складність алгоритмів керування.

У багатьох практичних задачах виправданим є спрощення складної нелінійної моделі до лінійної, що дозволяє отримати прийнятні параметри регулятора в певній робочій точці. Проте при суттєвих змінах умов функціонування об'єкта виникає необхідність адаптації регулятора або заміни моделі більш гнучкою.

У зв'язку з цим особливої ваги набувають системи керування, які поєднують адаптивні методи та нейромережеву апроксимацію, що забезпечує:

- можливість побудови компактної моделі об'єкта;
- автоматичну адаптацію параметрів моделі та регулятора під змінні умови роботи;
- підвищення точності та стійкості керування.

Загальна структура такої нейромережевої адаптивної системи зображена на рис. 1.4.

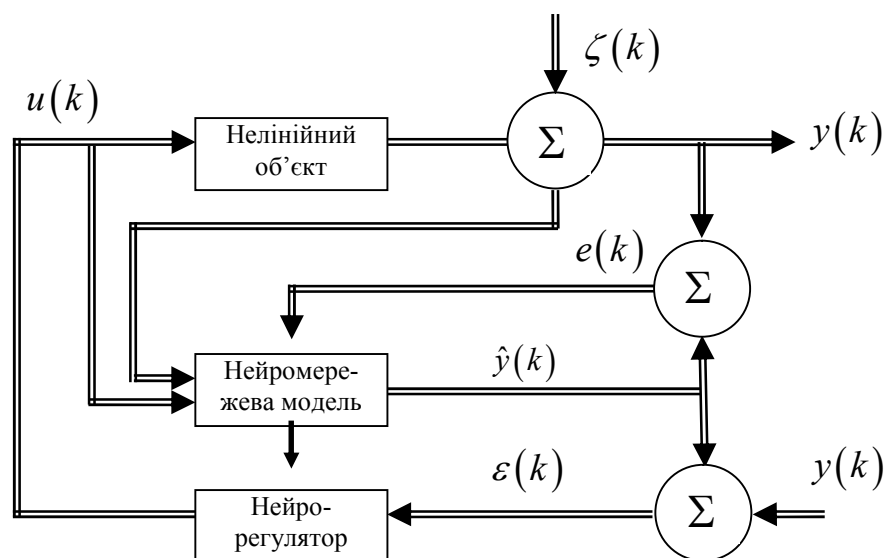


Рисунок 1.4 – Структура нейромережевої системи адаптивного управління

Розглянемо нелінійний багатовимірний динамічний об'єкт, що описується рівнянням

$$y(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)] + \xi(k), \quad (1.8)$$

де $y = (y_1(i), \dots, y_{N_y}(i))^T$; $u(i) = (u_1(i), \dots, u_{N_u}(i))^T$ – вектори вихідних і керуючих впливів розмірностей $N_y \times 1$ і $N_u \times 1$;

$f(\cdot) = (f_1(\cdot), f_2(\cdot), \dots, f_{N_y}(\cdot))^T$ – невідома нелінійна векторна функція;

$\xi(k) = [\xi_1(k), \dots, \xi_{N_y}(k)]^T$ – вектор завад.

Вектор узагальненого сигналу, що подається на вхід нейромережевої моделі, визначимо як

$$x(k) = [y^T(k-1), \dots, y^T(k-m), u^T(k-1), \dots, u^T(k-n)]^T, \quad (1.9)$$

Тоді рівняння моделі набуває вигляду

$$y(k) = f(x(k) + \xi(k)). \quad (1.10)$$

Навчання моделі здійснюється за критерієм мінімізації середньоквадратичної похибки.

Після завершення ідентифікації нейромережеву модель залучають до формування стратегії керування.

Завдання керування полягає у визначенні керуючого впливу $u(k)$, який мінімізує функціонал якості

$$J(k) = \|\varepsilon(k)\|^2 = \frac{1}{2} \|y^*(k) - \hat{y}(k)\|^2, \quad (1.11)$$

де $y^*(k)$ – бажані значення вихідних координат.

Для пошуку мінімуму функціонала (1.11) використовують різні рекурентні оптимізаційні алгоритми, у тому числі градієнтні алгоритми:

$$u(k) = u(k-1) + \gamma(k)(\nabla_u \mathcal{E}(k)), \quad (1.12)$$

де $\nabla_u \mathcal{E}(k) = \frac{\partial \mathcal{E}(k)}{\partial u(k)}$; $\gamma(k) > 0$.

1.4 Огляд структур нейромережевих систем управління та вибір архітектури

Проектування будь-якої системи автоматичного керування (СУ) традиційно здійснюється як послідовність логічно пов'язаних етапів. У загальному вигляді цей підхід формується наступними стадіями:

- формування концептуальних засад побудови системи керування;
- створення та математичне/імітаційне моделювання системи відповідно до запропонованої концепції;
- аналіз отриманих результатів, у тому числі оцінка якості регулювання;
- удосконалення первинної концепції, корекція структурних рішень і параметрів системи.

Цей цикл повторюється багаторазово, а теоретичні дослідження супроводжують процес на всіх етапах, визначаючи шляхи вдосконалення вибраної концепції та розширюючи її застосування на інші типи об'єктів і задач.

Подібним чином еволюціонував і напрям систем керування на основі штучних нейронних мереж (ШНМ). Водночас важливо зазначити, що розвиток апаратних і програмних засобів для нейромереж багато в чому випереджає теоретичне обґрунтування їх поведінки у складних умовах. Іншими словами, високий рівень інженерних рішень інколи “забігає вперед” порівняно з фундаментальним описом процесів, які відбуваються у таких системах.

Сьогодні запропоновано надзвичайно широку номенклатуру структур нейромережевих систем регулювання. Частина цих систем поєднує аналітичні

закони управління з використанням нейромоделей – у такому випадку ШНМ виконує реконструкцію або оцінку параметрів моделі об'єкта, фактично доповнюючи класичні методи. Це так звана група структурованих методів нейрорегулювання.

Протилежний підхід – неструктуровані методи, де нейронна мережа є центральним елементом регулювання і повністю реалізує закон управління, використовуючи свої потужні апроксимаційні властивості. Саме такі системи дають змогу максимально розкрити потенціал нейромереж.

Нижче стисло, але розширено проаналізовано найактуальніші архітектури та концепції нейромережових систем керування.

Однією з класичних схем є система зворотного зв'язку, де коефіцієнти регулятора змінюються в реальному часі – так звані **самоналагоджувальні регулятори**, серед яких відомий регулятор Астрома [43]. У таких системах параметри контролера адаптуються на кожному циклі дискретного управління на основі поточних оцінок характеристик об'єкта (рис. 1.5).

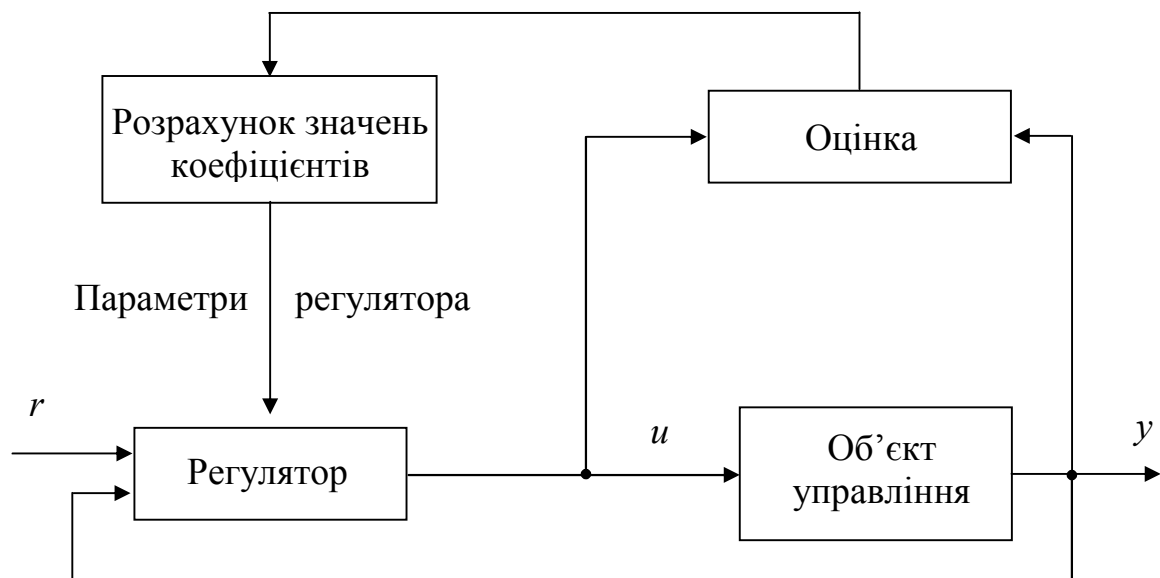


Рисунок 1.5 – Структурна схема замкненої системи із регулятором, параметри якого налаштовуються в режимі реального часу

Ще одним добре розробленим класом адаптивних систем є системи, побудовані на основі еталонної моделі Ляпунова. В адаптивних регуляторах цього типу керована система повинна прагнути до відтворення поведінки заздалегідь визначеної моделі, що має бажані динамічні властивості [44].

Забезпечується не лише асимптотичне наближення вихідного сигналу до еталонної траєкторії, а й гарантовані характеристики перехідних процесів у період адаптації (рис. 1.6).

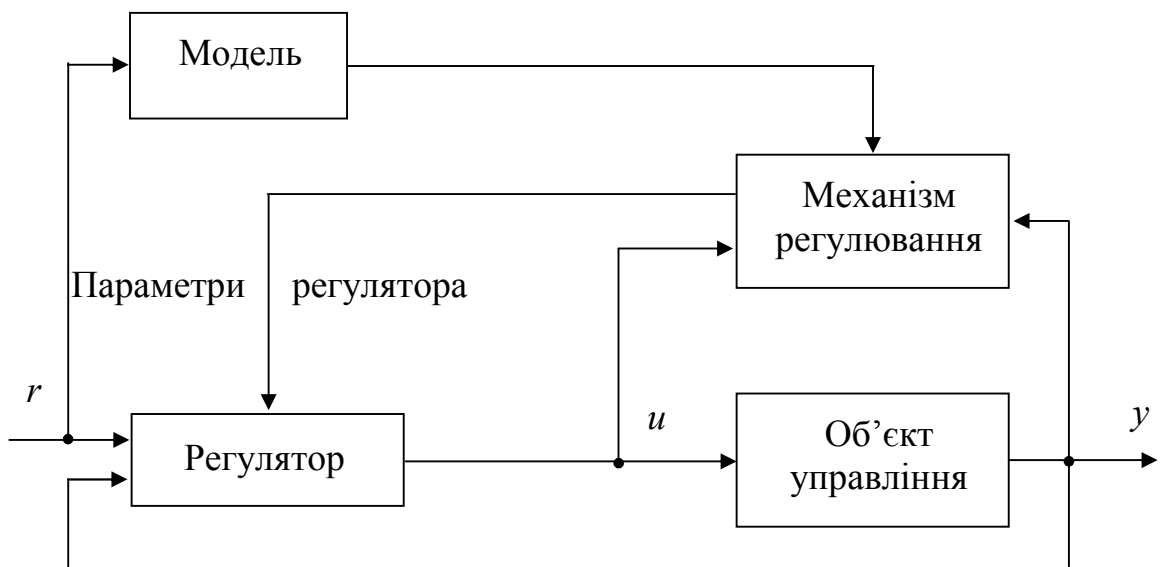


Рисунок 1.6 – Схема адаптивного регулювання, в якій реальний об'єкт повторює динаміку еталонної моделі.

Системи керування, у яких в тій чи іншій формі використовуються нейронні мережі, розглядаються як вагома альтернатива класичним методам. Їх ефективність пояснюється тим, що навіть відносно проста мережа з одним прихованим шаром та достатньою кількістю нейронів може апроксимувати будь-яку неперервну функцію з необхідною точністю. Це твердження ґрунтується на теоремі Вейерштрасса, яка детально аналізується в [45].

Саме тому ШНМ широко застосовуються як для ідентифікації, так і для синтезу закону керування.

Однією з перших нейромережових СУ була система, сформована за принципом "копіювання" вже наявного контролера. Ще у 1964 році Уїдроу (Widrow)

запропонував метод, у якому нейронна мережа набуває “експертних” властивостей шляхом навчання на вихідних даних існуючого керуючого пристрою. Архітектура такого підходу представлена на рис. 1.7.

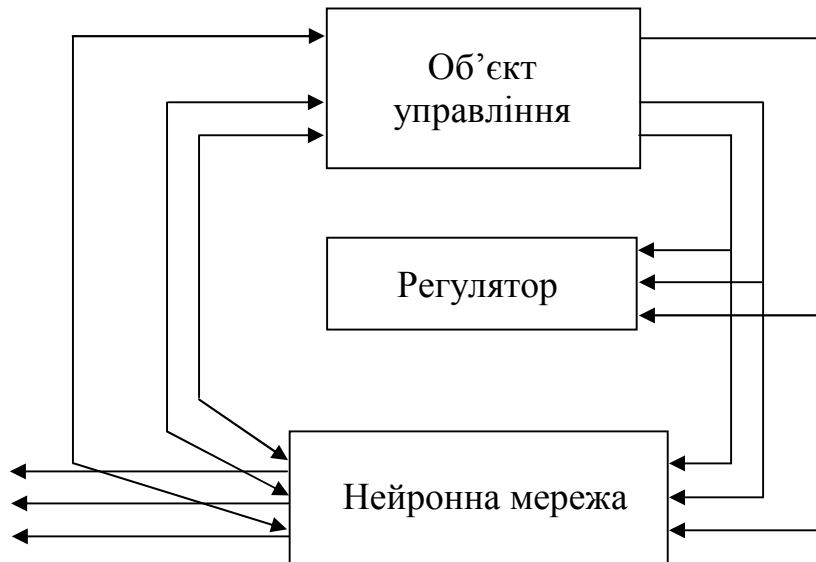


Рисунок 1.7 – Архітектура нейромережевої системи управління, що навчається за виходами вже існуючого контролера (“експертне копіювання”).

Хоча на перший погляд може здатися нелогічним створювати додатковий контролер, якщо робочий контрольний пристрій вже існує, це має ряд аргументів:

- первинний контролер може бути незручним для експлуатації (наприклад, людина-оператор);
- нейронна мережа може працювати з іншим набором змінних, зручнішим для сенсорного вимірювання або цифрової обробки;
- ШНМ може узагальнити поведінку “експерта” і працювати стабільніше в нестандартних ситуаціях.

На сьогодні розроблено багато структур нейромережевих регуляторів, однак їхнє функціональне призначення однакове – формування такого керуючого впливу, який забезпечить перехід системи з початкового стану до необхідного кінцевого за оптимальною траєкторією.

Якість керування та спосіб навчання суттєво залежать від обраної схеми. Найбільш вживаними є:

1) Непряме нейромережеве управління.

У цьому випадку в режимі реального часу спочатку визначаються параметри моделі об'єкта, а вже на їх основі – параметри регулятора (рис. 1.8).

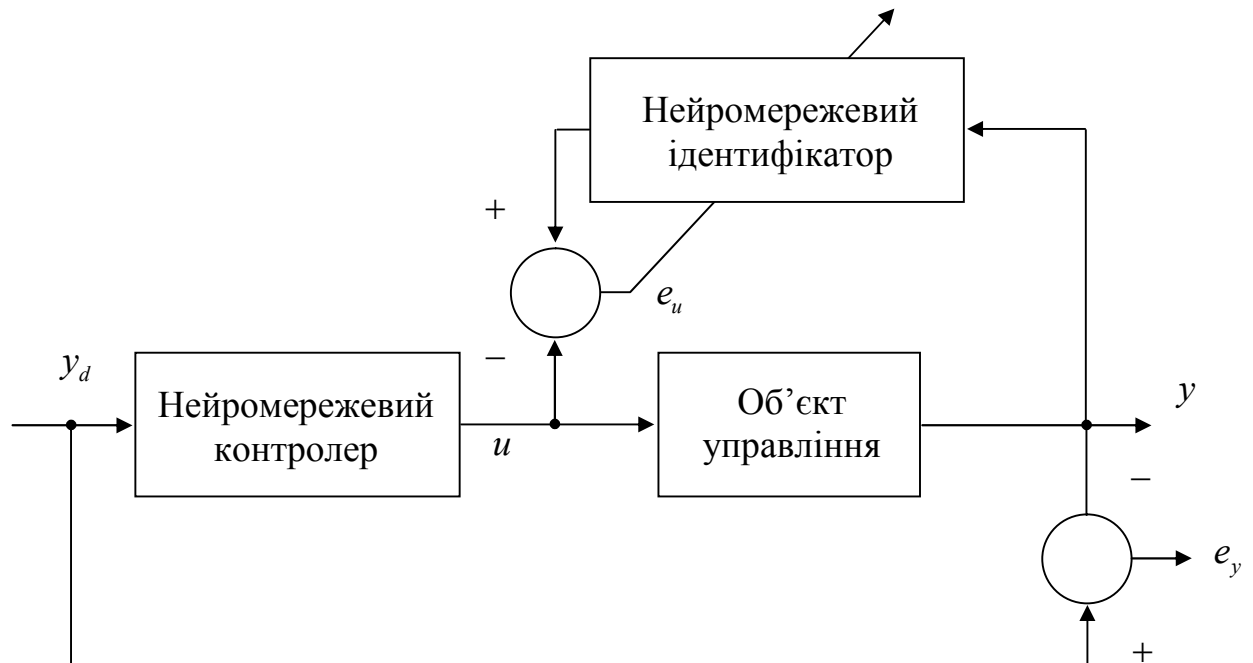


Рисунок 1.8 – Архітектура системи непрямого керування, де нейромережа виконує роль ідентифікатора об'єкта

Недолік полягає в тому, що мінімізується лише помилка моделі e_u , а не безпосередньо помилка вихідного сигналу e_y . Відповідно, немає гарантій, що похибка є мінімальною.

2) Пряме нейромережеве управління

У схемах прямого регулювання мережа налаштовується таким чином, щоб мінімізувати саме помилку виходу e_y без проміжної реконструкції моделі (рис. 1.9). Це підвищує точність, але ускладнює гарантування стабільності.

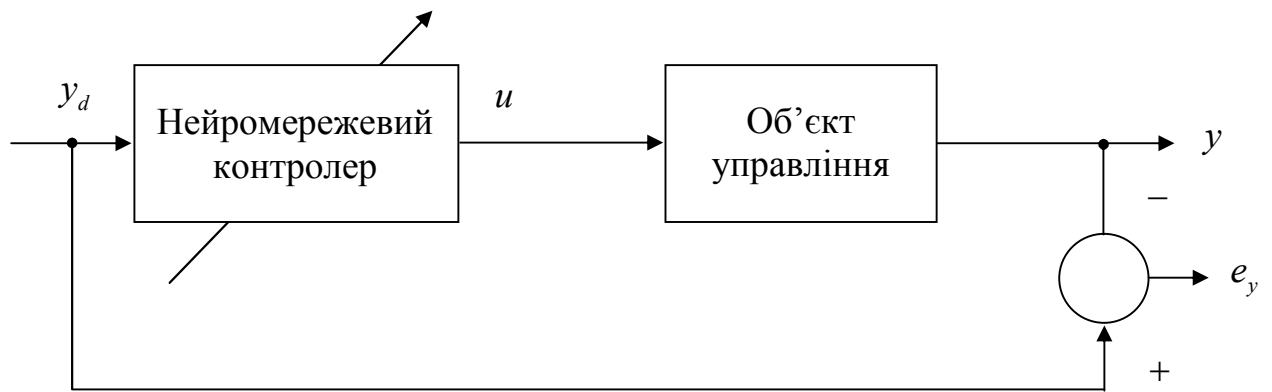


Рисунок 1.9 – Структура прямого нейромережевого управління, де адаптація відбувається за вихідною похибкою об'єкта.

У більшості варіантів цільовою функцією є середньоквадратична похибка:

$$E_y = \frac{1}{2}(y_d - y)^2. \quad (1.13)$$

Питання стабільності та керованості таких систем детально проаналізовано у значній кількості робіт [46–56].

Особливу увагу в сучасній теорії привертає метод предиктивного керування (Model Predictive Control, MPC) [57–61]. Він базується на тому, що регулятор формує оптимальну послідовність керуючих впливів на основі прогнозу майбутніх значень вихідного сигналу (рис. 1.10).

Система обчислює оптимальну траєкторію на горизонті прогнозування, а до об'єкта подає тільки перший елемент знайденої послідовності. У наступний момент часу оптимізація повторюється.

Переваги MPC:

- висока якість регулювання;
- можливість врахування складних нелінійностей і обмежень;
- природна здатність працювати з прогнозними моделями.

Недоліки:

- значні обчислювальні витрати;
- необхідність точної моделі об'єкта.

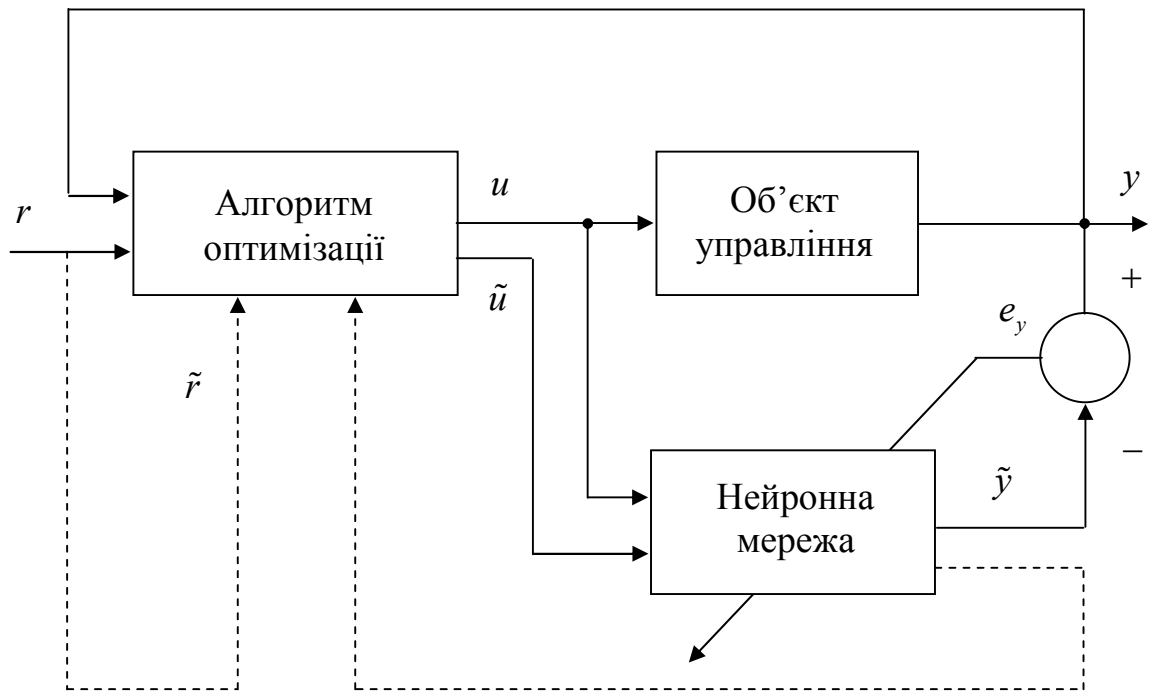


Рисунок 1.10 – Принцип дії предиктивного регулятора, який формує оптимальну траєкторію керуючого сигналу на основі прогнозної моделі

У магістерській роботі увага приділяється нелінійному MPC із застосуванням нейромережевої моделі.

Одним із важливих напрямів сучасних досліджень є порівняння нейромережевих методів керування з іншими типами регуляторів. Такий огляд представлено у роботі [62], де проаналізовано показники стійкості, швидкість збіжності, робастність до збурень, складність реалізації та інші характеристики.

Результати узагальнено в таблиці 1.1.

Таблиця 1.1 – Порівняльні характеристики різних методів систем управління

Критерій	Зворотний зв'язок з адаптивними коефіцієнтами	Адаптивне управління з еталонною моделлю Ляпунова	Нейромережеве управління
Стійкість	Найнижча	Найвища	Середня
Швидкість збіжності	Дуже висока	Помірна	Найнижча
Робота в реальному часі	Середня	Середня	Найкраща
Складність реалізації	Найвища	Середня	Середня
Помилка стеження	Середня	Найменша	Середня
Придушення збурень	Найкраще	Найгірше	Середнє
Робастність до розузгодження моделі	Найгірша	Середня	Найвища

1.5 Загальні відомості про роторні екскаватори

Роторні екскаватори становлять ключову ланку сучасних видобувних технологічних комплексів. Це надзвичайно складні багатофункціональні машини, до складу яких входить більше десяти взаємопов'язаних виконавчих механізмів, а також велика кількість допоміжного обладнання, головним завданням якого є безперебійне транспортування добутої маси до наступних елементів технологічного ланцюга. Узагальнена структурна схема компонування такої машини наведена на рис. 1.13.

Конструктивно роторний екскаватор складається з двох основних частин – поворотної та нерухомої, які з'єднані між собою опорно-поворотним пристроєм 1. Цей вузол забезпечує обертання верхньої будови відносно вертикаль-

ної осі, що дозволяє здійснювати забір породи у широкому секторі. На поворотній платформі кріпиться роторна стріла 4 з робочим колесом 3, по периферії якого встановлено ковші 2. Під час обертання ротора та поворотного руху стріли ковші зрізають шар породи, а відібраний матеріал надходить на стрічковий конвеєр 16, змонтований всередині фермової конструкції стріли.

Подрібнена маса надходить у центральну частину екскаватора, де розташований додатковий дробильний вузол. Після проходження через площину опорно-поворотного пристрою 1 порода переміщується на конвеєр нижньої платформи, а далі – на стрічковий конвеєр, встановлений у розвантажувальній консолі 10.

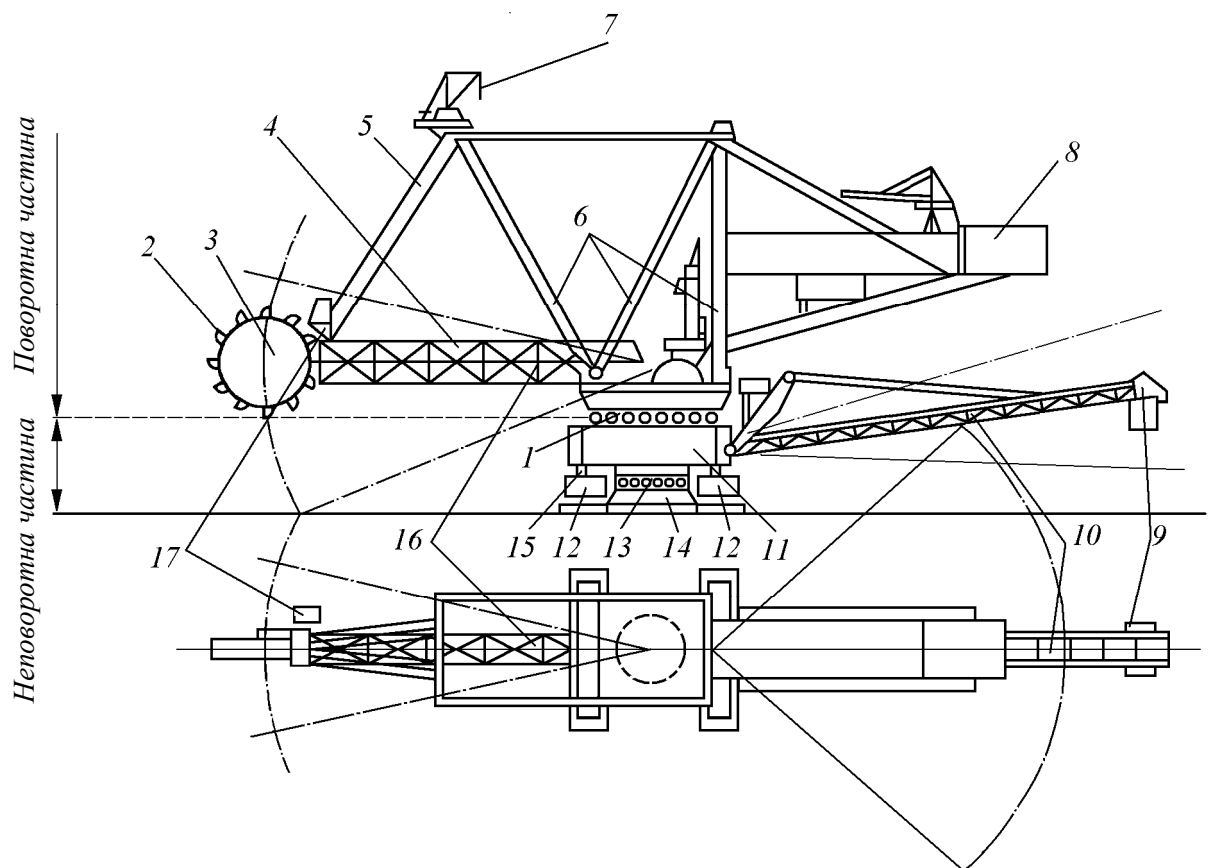


Рисунок 1.13 – Структурно-компонувальна схема роторного екскаватора

Для коригування просторового положення робочого колеса у вертикальній площині стріла може змінювати кут нахилу в певних межах завдяки канатній підвісці 5. Верхнє положення визначає максимально можливу висоту уступу, нижнє – граничну глибину копання. Верхні блоки підвіски закріплено на

порталі 6. Система підкосів 8, винесених у бік протилежний роторній стрілі, виконує функцію противаги та компенсує момент, що виникає від маси стріли.

Безпосередньо під опорно-поворотним вузлом розміщений масивний силовий пояс 11, який з'єднаний з опорною рамою екскаватора 14. З двох сторін до нерухомої частини кріпляться потужні гідроциліндри 15, призначені для встановлення машини на ґрунт через опорну базу або через рейково-крокуючі «лижі».

Розвантажувальна консоль 10 може змінювати своє положення як у горизонтальній, так і у вертикальній площинах, забезпечуючи гнучкість і автономність під час подачі породи на транспортні засоби або стаціонарні конвеєрні лінії. Аналогічно роторній стрілі, вона обладнана механізмами регулювання кута нахилу.

Управління всіма технологічними процесами здійснюється з декількох операторських кабін: біля роторного колеса 17, на кінці розвантажувальної консолі 10, а також на нижній рамі, що забезпечує максимальну зручність та безпеку роботи персоналу.

Для виконання розворотів екскаватора при зміні напрямку його переміщення застосовується додатковий поворотний вузол 13, який дозволяє обертати всю машину разом із верхньою будовою відносно бази.

Обслуговування конструктивних елементів і технологічного обладнання полегшується завдяки встановленим у трьох зонах 7 підйомним кранам вантажопідйомністю 5–20 т: на поворотній платформі, у верхній надбудові та на похилому порталі.

Основними механізмами, відповідальними за виконання робочого циклу роторного екскаватора, є: привід ротора, привід повороту стріли, механізм розвороту розвантажувальної консолі, механізм розвороту екскаватора, підйомний привід, механізм ходу, приводи стрічкових конвеєрів стріли та консолі.

Компоновка електроприводів екскаватора показана на рисунку 1.14. Безпосередньо на роторному колесі встановлено електродвигуни та редуктори приводу обертання 1, а також електропривод барабанного живильника 2, який

скеровує потік добутої породи на конвеєр стріли. Привід стрілового конвеєра 3 розміщений поблизу осі гойдання стріли.

На поворотній частині опорно-поворотного механізму встановлені електроприводи 6, що забезпечують поворот верхньої будови відносно нижньої. Підйомний привід стріли 5 розташований у надбудові протипаги, де також змонтовано вентиляційне обладнання 4.

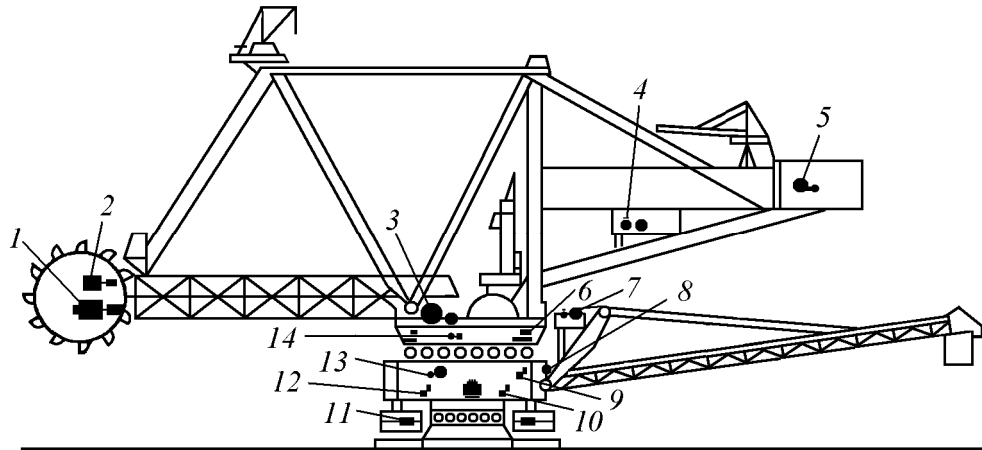


Рисунок 1.14 – Розташування індивідуальних електроприводів роторного екскаватора

У верхньому поясі конструкції знаходяться приводи 14 дробильного або завантажувального обладнання. Підйом розвантажувальної консолі у вертикальній площині здійснюється за допомогою електроприводів 7, встановлених на спеціальному майданчику. Нижче цієї зони розміщується привід повороту консолі 9 та привід її стрічкового конвеєра 8.

У нерухомій частині екскаватора монтуються електроприводи механізму розвороту машини 12, привід конвеєра нижньої рами 13, а також система електроживлення, до складу якої входять струмоприймачі, вимикачі, трансформатори та розподільні пристрої як високої, так і низької напруги 10. У конструкції рейково-крокуючого ходу розташовано електроприводи ходових лебідок 11.

1.6 Обґрунтування вибору системи електроприводу роторного колеса екскаватора

Роторне колесо екскаватора виконує основний робочий рух, забезпечуючи процес різання ґрунту. Тому електропривод цього механізму повинен функціонувати в режимі тривалого навантаження. Оскільки за технологічними умовами немає потреби регулярно змінювати напрямок обертання ротора, привід може бути виконаний у нереверсивному варіанті.

Для того щоб забезпечити раціональні умови різання порід різної міцності та максимально використати потенціал машини, електропривод має давати можливість зменшувати кутову швидкість роторного колеса на 30–60% відносно номінальної величини. Звідси випливає вимога до діапазону регулювання швидкості D , що визначається відношенням середнього значення максимальної швидкості до мінімальної:

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

Типові величини такого діапазону лежать у межах $(1,4 - 1,7) : 1$.

Для запобігання можливим резонансним коливанням металоконструкцій, що можуть виникати внаслідок несприятливих поєднань робочої швидкості ротора, пружності системи «ротор – стріла – підвіска», особливостей породи тощо, доцільним є використання плавного, безступінчатого регулювання швидкості. Перехідні процеси повинні мати оптимальний характер і не допускати надмірних динамічних навантажень на механізми та конструктивні елементи екскаватора.

Специфіка роботи механізму полягає в тому, що регулювання швидкості привода відбувається переважно під час різання, тобто під навантаженням. Через обмеження щодо динамічних зусиль, а відповідно й допустимих прискорень, електропривод у режимах пуску та зниження швидкості працює фактично лише в руховому режимі. Тому реалізація рекуперативного гальмування не є обов'язковою вимогою.

Механічна характеристика електроприводу, що встановлює залежність між електромагнітним моментом та кутовою швидкістю електродвигуна, повинна мати «екскаваторний» тип. Для привода роторного колеса бажано забезпечити підвищену жорсткість робочої ділянки характеристики. Якщо жорсткість недостатня, коливання сили різання спричинятимуть відповідні зміни швидкості двигуна і роторного колеса. Чим м'якша характеристика, тим помітнішими будуть ці коливання. Додатково амплітуда може збільшитися через обернено пропорційну залежність між лінійною швидкістю різання та площею зрізу стружки, а отже, моментом навантаження. Подібні коливання створюють загрозу резонансних явищ у металоконструкціях екскаватора.

Оскільки можлива поява короточасних перевантажень або навіть заклинювання ковшів при зустрічі з твердою перешкодою, електропривод повинен мати обмеження максимального моменту. Ділянка обмеження моменту характеризується коефіцієнтом відсічення:

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

де $M_{\text{відс}}$ – момент відсічення, перевищення якого переводить систему з робочої частини механічної характеристики на ділянку обмеження;

$M_{\text{стоп}}$ – момент стопоріння, що відповідає максимально допустимому для механізму моменту з урахуванням його міцності.

Таке співвідношення $k_{\text{відс}}$ дозволяє машиністу, навіть за відсутності автоматичного коректора швидкості повороту роторної стріли від навантаження привода роторного колеса, орієнтуватися на зміну обертів і вчасно зменшувати навантаження, запобігаючи перевантаженню та стопорінню механізму.

Незважаючи на використання екскаваторної характеристики, у режимі стопоріння можливі надмірні динамічні навантаження. Тому в кінематичну схему додають елемент із обмеженим передаваним моментом, який запобігає пошкодженню обладнання.

Електродвигуни привода роторного колеса встановлюються на оголовку стріли і працюють у складних умовах: постійна вібрація, струс, пил, змінні кути нахилу конструкції. Тому до них висуваються вимоги щодо мінімальної маси й габаритів, закритого виконання, здатності працювати при вібраціях і нахилах, а також оснащення системою очищення повітря для охолодження.

Сукупність цих складних вимог визначила широке застосування в екскаваторних приводах систем «керований перетворювач – двигун» постійного струму. Як перетворювачі використовують електромашинні схеми (Г–Д) або реверсивні тиристорні перетворювачі (ТП–Д). У межах цієї магістерської роботи для електроприводу роторного колеса обрана система типу «генератор – двигун», яка забезпечує необхідні характеристики та режимні можливості.

1.7 Формулювання наукової проблеми, обґрунтування актуальності дослідження та визначення шляхів її вирішення

Проведений у попередніх розділах аналіз технічних, динамічних і експлуатаційних особливостей механізму обертання роторного колеса екскаватора свідчить про доцільність інтеграції нейромережевих підходів у процес синтезу системи керування. Такий вибір обумовлений, насамперед, складною і суттєво нелінійною взаємодією між елементами системи «ротор – стріла – підвіска стріли – порода». Саме комбінація кутових швидкостей ротора, пружних властивостей кінематичного ланцюга та параметрів породи може викликати резонансні явища, що негативно впливають на працездатність механізму, викликаючи небажані коливання металоконструкцій та підвищені динамічні навантаження. Відповідно, система керування повинна формувати перехідні процеси оптимальної форми, обмежувати амплітуди коливань та забезпечувати стабільність у широкому діапазоні режимів.

Як показує практика експлуатації роторних екскаваторів, традиційні регулятори, що використовуються в електроприводах, не дозволяють гарантувати необхідні показники якості управління при наявності значних нелінійностей,

пружних зв'язків і змінних навантажень. Тому актуальним є пошук альтернативних підходів, здатних гнучко адаптуватися до умов роботи механізму.

Теоретичні основи адаптивного та оптимального управління надають можливість будувати системи, які мінімізують або максимізують заданий критерій якості. У сучасну теорію управління також включені методи ідентифікації моделей та оцінювання внутрішніх станів динамічних об'єктів, що дозволяє більш точно відтворювати їх поведінку в режимі реального часу. Процедури ідентифікації використовуються як для статичних, так і для адаптивних систем керування, даючи змогу визначати структуру моделі, оцінювати параметри та формувати математичну основу для синтезу регуляторів.

Однак класичні підходи ґрунтуються переважно на лінеаризації, яка є прийнятною лише у випадку слабо виражених нелінійних ефектів або значних сталих часових параметрів. Для складних механічних систем з множинними ступенями свободи, такими як привід роторного колеса, цей підхід не забезпечує необхідної точності моделі, що в свою чергу обмежує ефективність синтезованих регуляторів. Поширені методи апроксимації нелінійностей – ряди Вольтерра, моделі Гаммерштейна–Вінера, поліноми Колмогорова–Габора – вимагають попереднього визначення структури моделі та її порядку, що значно ускладнює застосування цих методів у системах зі змінною структурою та у присутності значних зовнішніх збурень.

Додаткову складність створює необхідність обробки шумів вимірювання, а також вимога до виконання алгоритмів у реальному часі, що потребує підвищеної швидкодії й обчислювальної ефективності.

З огляду на вказані обмеження, класичні методи управління не завжди здатні забезпечити прийнятні результати. У той же час штучні нейронні мережі, як представники методів штучного інтелекту, демонструють високу ефективність у задачах моделювання та керування нелінійними динамічними системами. Нейромережеві структури здійснюють паралельну обробку сигналів, що дозволяє значно збільшити швидкість розрахунків та забезпечити стійкість роботи навіть при відмові частини елементів. Властивості навчання й узагальнення

дають можливість будувати моделі складних об'єктів на основі обмеженого набору вхідно-вихідних даних.

Особливу цінність для задач управління становить нелінійність активаційних функцій нейронів, яка дає можливість моделювати та відтворювати властивості реальних об'єктів без необхідності лінеаризації. Саме тому останніми роками нейромережеві системи все ширше застосовуються для керування процесами з вираженими нелінійними характеристиками.

На відміну від традиційних адаптивних та оптимальних регуляторів, які вимагають детального набору апіорної інформації, нейромережеві системи здатні до самонавчання, тобто формування моделі об'єкта лише на підставі експериментальних даних. Це значно розширює можливості їх практичного застосування в умовах невизначеності характеристик механізму.

Побудова системи управління на основі нейрорегулятора передбачає два основних етапи:

1. ідентифікацію об'єкта, тобто створення його нейромережевої моделі, що досягається шляхом навчання на вибірках вхідних і вихідних даних;
2. синтез закону управління, який забезпечує досягнення заданих показників якості з урахуванням отриманої моделі.

З огляду на викладене, формулюється мета магістерської роботи.

Мета дослідження. Створення нейромережевої системи управління електроприводом механізму обертання роторного колеса екскаватора, що враховує пружні властивості структурних елементів системи «ротор – стріла – підвіска» та забезпечує поліпшення динамічних характеристик привода.

Для досягнення мети поставлено такі *завдання*:

1. Обрати та обґрунтувати тип системи електроприводу механізму обертання роторного колеса, виконати вибір електродвигуна та силового обладнання, провести синтез системи керування електроприводом і змоделювати роботу системи на ЕОМ.

2. Розробити математичну модель системи управління, яка враховує пружні властивості ланок механізму; виконати моделювання багатомасової структури та провести аналіз отриманих результатів.
3. Сформувати структурну схему нейромережевої системи управління, здатної забезпечувати високоякісне регулювання при обмеженій жорсткості механічної частини привода.
4. Побудувати та навчити багат шарову пряму нейронну мережу, що виконує високоточну ідентифікацію багатомасової системи.
5. На основі предиктивного принципу керування та нейромережевої моделі об'єкта синтезувати нейромережевий регулятор, який забезпечує підвищену швидкодію та якість управління.
6. Здійснити чисельне моделювання роботи нейромережевої системи та виконати детальний аналіз отриманих характеристик.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. У ході аналізу літературних джерел, присвячених сучасним методам управління, обґрунтовано доцільність застосування штучних нейронних мереж, зокрема багатошарових мереж прямого поширення, у задачах автоматичного регулювання. Хоча класичні принципи синтезу систем керування дозволяють формувати високоякісні регулятори, їх практичне впровадження ускладнюється технічними обмеженнями: необхідністю вимірювання трудноспостережуваних станів (наприклад, пружного моменту), а також неможливістю реалізації оптимальних параметрів через особливості функціонування реальних електроприводів. Тому актуальним є використання нелінійних регуляторів, які, спираючись на меншу кількість вимірюваних сигналів, здатні забезпечувати необхідні керувальні впливи. Одним із таких рішень виступає нейрорегулятор, що реалізує принципи нейронних мереж, подібні до біологічних систем.

2. Самонавчальна здатність штучних нейронних мереж відкриває можливість ефективного використання нейрорегуляторів в умовах значних параметричних і структурних невизначеностей. На відміну від класичних адаптивних систем, що вимагають детальної апріорної інформації про об'єкт керування, нейромережеві системи дозволяють мінімізувати обсяг необхідних даних. Поєднання простоти апаратної та програмної реалізації з високими адаптаційними властивостями робить такі регулятори придатними для роботи зі складними нелінійними процесами у режимі реального часу.

3. У процесі дослідження підтверджено ефективність використання багатошарових мереж прямого розповсюдження для побудови нейромережевих моделей об'єктів керування. Такі мережі є універсальними апроксиматорами та широко застосовуються для класифікації, прогнозування та моделювання нелінійних систем. Окремо були розглянуті методи навчання та обґрунтовано вибір алгоритму навчання для нейронної мережі-регулятора.

4. Забезпечення оптимальної роботи об'єкта неможливе без коригування параметрів регуляторів при зміні зовнішніх умов та характеристики наванта-

ження. Це визначає доцільність побудови адаптивних систем керування на основі нейромережових методів, які забезпечують оновлення параметрів моделей і регулятора у процесі експлуатації, що дозволяє підтримувати якість управління на високому рівні.

5. Створення адекватної математичної моделі є ключовим етапом під час побудови систем регулювання та прогнозування динамічних процесів. Доведено, що для ідентифікації нелінійних динамічних об'єктів особливо ефективним є використання штучних нейронних мереж. Найчастіше для цього застосовуються багат шарові персептрони та мережі радіально-базисних функцій, які забезпечують високу точність відтворення властивостей об'єкта.

6. На основі аналізу існуючих структур нейрорегуляторів обрано концепцію нелінійного предиктивного керування як найбільш ефективну для синтезу нейромережового регулятора. Такий підхід дозволяє враховувати нелінійності, багатозв'язність та динаміку об'єкта при формуванні оптимального керувального впливу.

7. Розглянуто практичну реалізацію узагальненого прогнозуючого керування, у якому багат шарова нейронна мережа прямого поширення використовується як нелінійна модель об'єкта. Підкреслено, що застосування оптимізаційного алгоритму Ньютона–Рафсона забезпечує швидку збіжність завдяки високій швидкості обчислень, що робить його придатним для використання в режими реального часу.

8. Встановлено, що нейромережові технології є перспективними для синтезу систем керування механізмом обертання роторного колеса екскаватора. У конструкціях роторних машин можливе виникнення резонансних явищ через поєднання параметрів швидкості, жорсткості та властивостей породи. Це негативно впливає на роботу механізмів та їх ресурс. Традиційні схеми регулювання не забезпечують необхідного рівня демпфування та не гарантують обмеження динамічних навантажень, тоді як нейромережові системи дозволяють значно покращити якість перехідних процесів та уникнути перевантажень конструкції.

9. На підставі аналізу вимог до електроприводів основних механізмів роторних екскаваторів обґрунтовано вибір системи електропривода постійного струму за схемою «генератор–двигун» (Г–Д) для механізму обертання роторного колеса. Такий електропривод забезпечує необхідний діапазон регулювання, стійку роботу під навантаженням, а також задовольняє вимоги щодо надійності в умовах вібрацій та запиленості.

10. Проведений огляд нейромережевих методів управління дав змогу сформулювати наукову задачу дослідження, що полягає у розробленні нейромережевої системи керування механізмом обертання роторного колеса екскаватора з урахуванням пружних зв'язків у механічній системі та потреби оптимізації перехідних процесів.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ОБЕРТАННЯ КОЛЕСА РОТОРНОГО ЕКСКАВАТОРА

2.1 Основні технічні характеристики роторного екскаватора та вимоги до системи управління

Для розрахунку електроприводу механізму обертання колеса роторного екскаватора та подальшого синтезу системи управління використовуються вихідні технічні дані, що наведені у таблиці 2.1.

Таблиця 2.1 – Основні технічні характеристики роторного екскаватора

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

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
10	Маса приймальної частини	кг	$m_{пч}$	$267 \cdot 10^3$
11	Маса відвальної частини	кг	$m_{вч}$	$250 \cdot 10^3$
12	Діаметр роликового круга	м	$D_{кр}$	14
13	Діаметр опорних роликів	м	$d_{тк}$	1
14	Момент інерції поворотної платфор- ми	кг · м ²	$J_{пл}$	$88 \cdot 10^7$
15	Частота обертання ротора	рад/хв	ω_p	5
16	Щільність ґрунту	кг/м ³	γ_p	$1,8 \cdot 10^3$
17	Коефіцієнт аеродинамічного опору		$k_{ад}$	1,4
18	Максимально допустиме вітрове на- вантаження	Па	$\lambda_{вт}$	300
19	Максимально подоланий ухил		γ	10°
20	Число двигунів механізму повороту		k	2
21	Максимально допустимі прискорен- ня і уповільнення платформи	с ⁻²	$\omega_{пл}$	0,005
22	ККД редуктора роторного колеса		η_p	0,8
23	ККД редуктора механізму повороту		η_p	0,45 – 0,65
24	Максимальний кут повороту		$\alpha_{п\ max}$	±75°

Вимоги до показників якості функціонування системи управління

Система управління механізмом обертання колеса екскаватора повинна забезпечувати стабільну роботу без постійних коливань та перебоїв. Основні показники якості її функціонування визначаються наступним чином: перерегулювання швидкості при подачі ступінчастого сигналу не повинно перевищува-

ти 10%, час встановлення вихідного параметра до стабільного значення (час регулювання) не повинен перевищувати 5 секунд.

2.2 Розрахунок потужності та обґрунтування вибору електродвигуна приводу обертання роторного колеса

Електропривод роторного колеса функціонує у режимі тривалої сталості, що визначає характер вибору потужності двигуна та його параметрів. Потужність, яку повинен розвивати електродвигун на своєму валу, визначається співвідношенням:

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

де η_p – ККД редуктора роторного колеса;

$P_{p\Sigma}$ – загальна потужність, необхідна для забезпечення роботи роторного колеса. Остання величина формується з урахуванням декількох складових і визначається виразом:

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

де $k_t = 1,03 \div 1,05$ – коефіцієнт, що описує сукупність навантажень на робочому органі, зумовлених тертям ковзів об масив породи та енерговитратами на надання ґрунтовим частинкам кінетичної енергії після їх відокремлення;

P_p – потужність, необхідна для здійснення процесу копання

P_{Π} – складова потужності, що витрачається на підйом відокремленої маси породи.

Значення потужності копання визначається залежністю:

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

де k_p – коефіцієнт розпушування, для даного типу породи приймається рівним $1,1 \div 1,5$;

$$P_p = \frac{3900 \cdot 1,1 \cdot 10^6}{3600 \cdot 1,4} \cdot 10^{-3} = 851 \text{ кВт}.$$

Потужність, що витрачається на подолання сили тяжіння при підйомі частини породи після її відриву, визначається за формулою:

$$P_n = Q \cdot \frac{\gamma_n \cdot g \cdot D_p}{k_p \cdot 7200} \cdot 10^{-3},$$

$$P_n = 3900 \cdot \frac{1,8 \cdot 10^3 \cdot 9,81 \cdot 12}{1,4 \cdot 7200} \cdot 10^{-3} = 82 \text{ кВт}.$$

Підставивши отримані значення P_p і P_n у формулу (2.2), отримуємо необхідну повну потужність приводу роторного органа:

$$P_{p\Sigma} = 1,04 \cdot (851 + 82) = 970 \text{ кВт}.$$

З огляду на це, потужність, яку повинен розвивати електродвигун, становить:

$$P_d = \frac{1,04 \cdot (851 + 82)}{0,8} = 1213 \text{ кВт}.$$

Відповідно до отриманого значення P_d з каталогу вибирається двигун постійного струму. Для забезпечення регламентованих режимів роботи доцільним є застосування компенсованого електродвигуна серії П-18-33-9к.

Основні паспортні дані вибраного електродвигуна: номінальна потужність $P_n = 1350 \text{ кВт}$, номінальна напруга $U_n = 630 \text{ В}$, частота обертання $n_n = 550 \text{ об/хв}$, номінальна сила струму $I_n = 2280 \text{ А}$, опір обмотки якоря $R_{яд} = 0,00462 \text{ Ом}$, опір обмотки додаткових полюсів $R_{дпд} = 0,00053 \text{ Ом}$, опір

компенсаційної обмотки $R_{\text{ко д}} = 0,00303 \text{ Ом}$, кількість пар полюсів $2p = 6$, ККД $\eta_{\text{д}} = 0,94$, момент інерції ротора $J_{\text{д}} = 775 \text{ кг} \cdot \text{м}^2$

Оскільки частка часу, що припадає на пускові та гальмівні режими, є незначною, перевірка двигуна на теплову перевантаженість не є необхідною.

2.3 Вибір основного електротехнічного обладнання системи електроприводу

Для побудови регульованої системи електроприводу роторного колеса прийнято використання схеми «генератор–двигун» (Г–Д), у якій функцію керованого силового перетворювача виконує генератор постійного струму.

Параметри генератора повинні задовольняти умовам:

$$I_{\text{нг}} \geq I_{\text{нд}}; \quad U_{\text{нг}} \geq U_{\text{нд}}; \quad P_{\text{нг}} \geq \frac{P_{\text{нд}}}{\eta_{\text{д}}}.$$

У відповідності до цих вимог вибирається генератор типу П-18-33-9к, який узгоджується з параметрами раніше обраного двигуна. Основні технічні характеристики генератора: номінальна потужність $P_{\text{н}} = 1750 \text{ кВт}$, номінальна напруга $U_{\text{н}} = 750 \text{ В}$, частота обертання $n_{\text{н}} = 750 \text{ об/хв}$, номінальний струм $I_{\text{н}} = 2535 \text{ А}$, опір обмотки якоря $R_{\text{я г}} = 0,00443 \text{ Ом}$, опір обмотки додаткових полюсів $R_{\text{дп г}} = 0,00049 \text{ Ом}$, опір компенсаційної обмотки $R_{\text{ко г}} = 0,00314 \text{ Ом}$, кількість пар полюсів $2p = 6$, ККД $\eta_{\text{г}} = 0,94$, напруга збудження $U_{\text{н зб}} = 750 \text{ В}$, опір обмотки збудження $R_{\text{зб г}} = 5,73 \text{ Ом}$, кількість витків обмотки збудження $W_{\text{зб}} = 330$

У якості привідного агрегату для генератора застосовується синхронний електродвигун СДЭ8-15-76-6, який має такі експлуатаційні параметри: номінальна потужність $P_{\text{н}} = 2200 \text{ кВт}$, номінальна напруга $U_{\text{н}} = 10 \text{ кВ}$, швидкість обертання $n_{\text{н}} = 1000 \text{ об/хв}$.

Для живлення обмотки збудження генератора обрано тиристорний збудник БУВ-3603-12, який забезпечує стабільну та плавно регульовану подачу струму збудження. Характеристики збудника: номінальна потужність $P_n = 11,5 \text{ кВт}$, випрямлена напруга $U_{дн} = 230 \text{ В}$, випрямлений струм $I_{дн} = 50 \text{ А}$, принцип керування – незалежне регулювання груп тиристорів.

2.4 Розрахунок і побудова статичних характеристик розімкненої та замкненої системи

На рисунку 2.1 наведена принципова схема розімкненого електроприводу типу Г–Д.

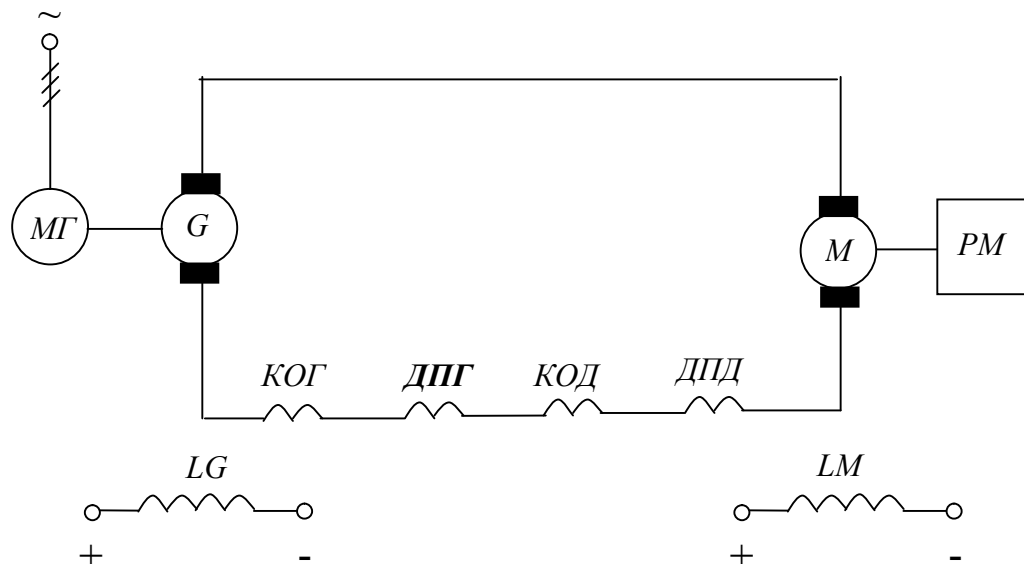


Рисунок 2.1 – Структурне зображення розімкненої системи типу Г–Д, що використовується в електроприводі повороту роторного колеса екскаватора

У схемі використано такі умовні позначення:

$РМ$ – виконавчий (робочий) механізм;

$М$, $Г$, $МГ$ – відповідно: робочий двигун, генератор та приводний (синхронний) двигун генератора;

$КОД$, $КОГ$, $ДПД$, $ДПГ$ – системи компенсаційних обмоток і додаткових полюсів робочого двигуна та генератора;

LM , LG – обмотки збудження двигуна і генератора.

Якірні кола генератора й двигуна ввімкнені послідовно, формуючи єдине якірне коло приводу.

Якщо ЕРС генератора розглядати як вхідну електричну напругу живлення двигуна, то загальний вираз, що описує електромеханічну характеристику приводу, можна записати у вигляді:

$$\omega = \frac{E_{\Gamma}}{k_{\Gamma} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\Sigma}}{k_{\Gamma} \Phi_{\text{нд}}} = \omega_0 - \Delta\omega_{\text{ст}}, \quad (2.1)$$

R_{Σ} – повний опір якірного кола, який дорівнює:

$$R_{\Sigma} = R_{\Sigma\text{д}} + R_{\Sigma\Gamma};$$

$R_{\Sigma\text{д}}$ – повний опір якірного кола двигуна:

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

$R_{\text{я д}}$, $R_{\text{дп д}}$, $R_{\text{ко д}}$ – опори відповідних обмоток двигуна;

$R_{\Sigma\Gamma}$ – повний опір якірного кола генератора:

$$R_{\Sigma\Gamma} = k_t (R_{\text{я Г}} + R_{\text{дп Г}} + R_{\text{ко Г}});$$

$R_{\text{я Г}}$, $R_{\text{дп Г}}$, $R_{\text{ко Г}}$ – опори відповідних обмоток генератора, позначено індексом «Г»;

k_t – температурний множник, який переводить номінальні (каталожні) опори, виміряні при 20°C, до робочої температури. Для ізоляції класу Н приймають $k_t = 1,36$.

Відомо, що в системі типу Г–Д загальний опір якірного кола суттєво більший за опір одного лише двигуна, тому характеристика робочого приводу є менш жорсткою, ніж природна характеристика двигуна постійного струму:

$$R_{\Sigma\text{д}} = 1,36 \cdot (0,00462 + 0,00303 + 0,00053) = 0,011 \text{ Ом},$$

$$R_{\Sigma\Gamma} = 1,36 \cdot (0,00433 + 0,00314 + 0,00049) = 0,01 \text{ Ом},$$

$$R_{\Sigma} = (0,011 + 0,01) = 0,021 \text{ Ом}.$$

k_d – конструктивний параметр двигуна;

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

Добуток величин k_d та $\Phi_{нд}$ (надалі позначатимемо як c) можна отримати, скориставшись співвідношенням, що описує природну електромеханічну характеристику електродвигуна:

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

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

$$\omega_n = \frac{\pi n_n}{30},$$

$$\omega_n = \frac{\pi \cdot 550}{30} = 57,6 \text{ с}^{-1},$$

$$k_d \Phi_{нд} = \frac{630 - 2280 \cdot 0,011}{57,6} = 10,5 \text{ В} \cdot \text{с} / \text{рад}.$$

Побудова електромеханічних характеристик розімкненої системи Г–Д.

Для розрахунку залежностей приймаємо максимальне значення ЕРС генератора E_r рівним номінальній напрузі двигуна: $E_r = U_n = 630 \text{ В}$.

Електромеханічні характеристики будуються для таких значень ЕРС: $E_r = U_n = 630 \text{ В}$, $E_r = 0,5U_n = 630/2 = 315 \text{ В}$, $E_r = 0$, $E_r = -0,5U_n = -315 \text{ В}$ і $E_r = -U_n = -630 \text{ В}$. Оскільки характеристики є прямими лініями, то для їх побудови достатньо двох точок.

1) Точка ідеального холостого ходу.

$$I = 0, \quad \omega = \omega_0 = \frac{E_r}{k_d \Phi_{нд}},$$

$$E_r = U_n = 630 \text{ В}, \quad \omega_{01} = \frac{630}{10,5} = 60 \text{ с}^{-1},$$

$$E_r = 0,5U_n = 315 \text{ В}, \quad \omega_{02} = \frac{315}{10,5} = 30 \text{ с}^{-1},$$

$$E_r = 0, \quad \omega_{03} = 0.$$

2) Номінальна робоча точка.

$$I = I_n = 2280 \text{ А}, \quad \omega_n = \frac{E_r}{k_d \Phi_{нд}} - I_n \frac{R_{я\Sigma}}{k_d \Phi_{нд}},$$

$$E_r = U_n = 630 \text{ В}, \quad \omega_{н1} = \frac{630}{10,5} - 2280 \cdot \frac{0,021}{10,5} = 60 - 4,56 = 55,44 \text{ с}^{-1},$$

$$E_r = 0,5U_n = 315 \text{ В}, \quad \omega_{н1} = \frac{315}{10,5} - 2280 \cdot \frac{0,021}{10,5} = 30 - 4,56 = 25,44 \text{ с}^{-1}.$$

Отримані залежності наведено пунктирними лініями на рисунку 2.2. Характеристики для негативних значень ЕРС мають дзеркальну симетрію відносно осі координат, що відповідає реверсному режиму приводу.

Оскільки $M = k_d \Phi_{нд} \cdot I$, механічні характеристики повністю повторюють електромеханічні.

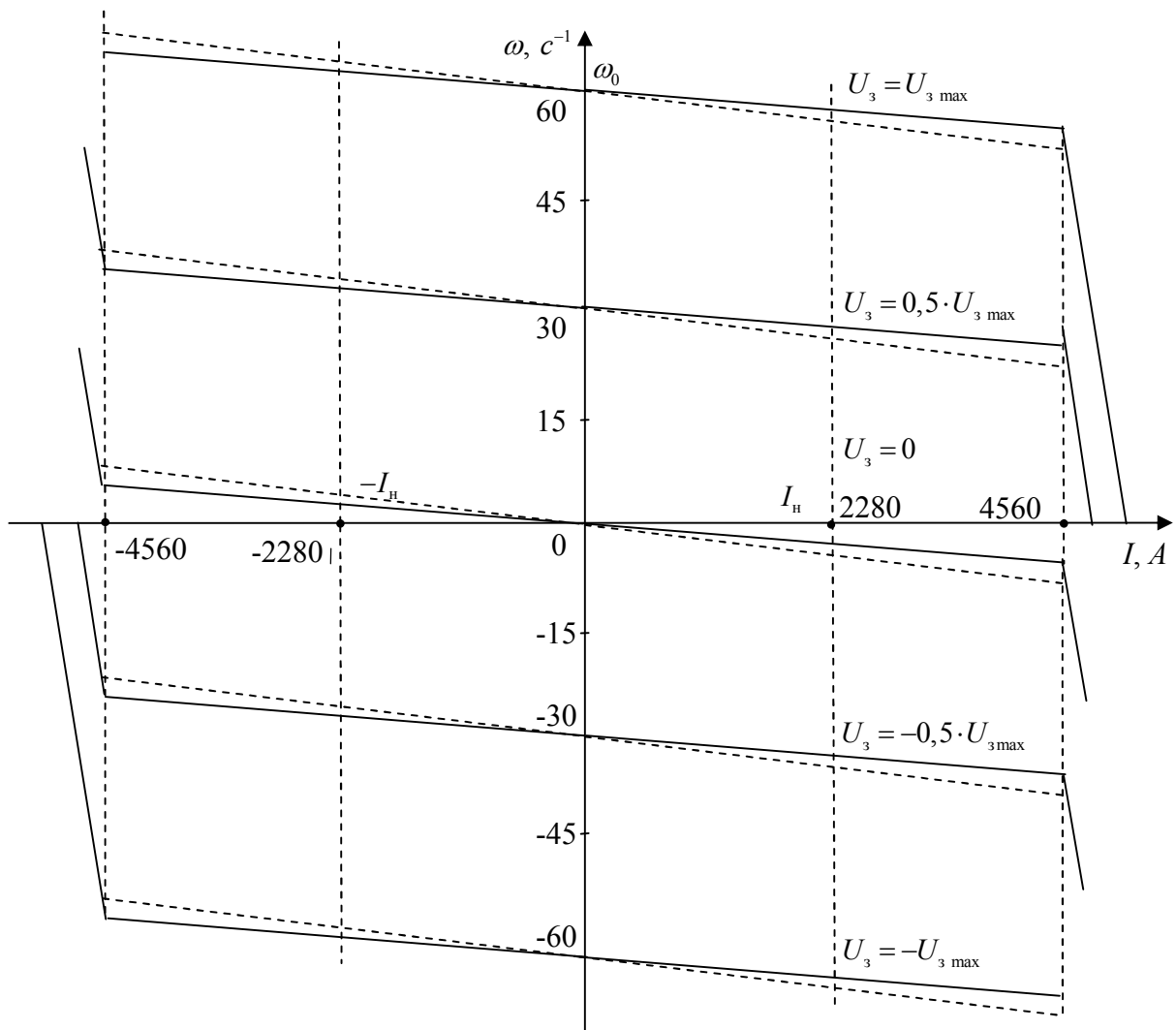


Рисунок 2.2 – Графічне представлення електромеханічних характеристик системи Г–Д у замкненому режимі роботи

Для електроприводів, що застосовуються в механізмах роторних екскаваторів, висуваються посилені вимоги щодо підвищеної жорсткості робочої частини електромеханічних (а відповідно й механічних) характеристик, а також щодо обмеження максимального струму та розвиваного моменту. Це зумовлює необхідність формування так званої «екскаваторної» характеристики, зображеної на рисунку 2.3.

На першій робочій ділянці характеристики, зі зростанням навантаження, до моменту досягнення струму відсічення $I_{я} = I_{\text{відс}}$ (що відповідає моменту відсічення $M_{\text{відс}}$), потрібно забезпечити пропорційне струму підвищення ЕРС ге-

нератора E_r . Такий підхід компенсує збільшення падіння напруги на опорі R_{Σ} і дозволяє суттєво збільшити жорсткість робочої частини характеристики. Після досягнення струму відсічення подальше зростання навантаження має приводити до пропорційного різниці $\Delta I_{\Sigma} = I_{\Sigma} - I_{\text{відс}}$ зменшення ЕРС E_r , а відповідно – і швидкості приводу. У результаті швидкість електроприводу знижується до нульового значення при струмі $I_{\Sigma} = I_{\text{стоп}}$ (що відповідає моменту стопоріння $M_{\text{д}} = M_{\text{стоп}}$) та при ЕРС генератора $E_r = I_{\text{стоп}} R_{\Sigma}$, що відповідає ділянці 2 характеристики.

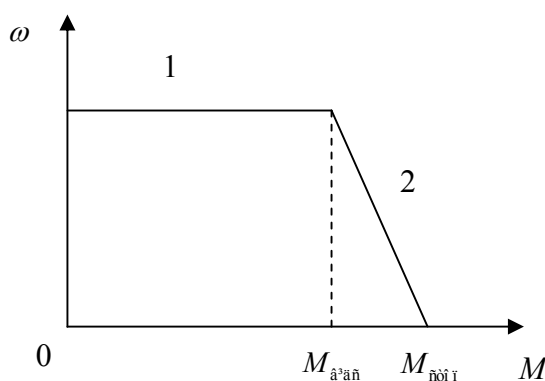


Рисунок 2.3 - Типова форма робочої (екскаваторної) механічної характеристики приводу, що застосовується для роторних екскаваторів

Сформулювати описану вище характеристику можливо лише при використанні замкненої системи автоматичного керування електроприводом. Як зазначалося раніше, у роторних екскаваторах застосовуються САУ, структурна організація яких включає підсумовуючий підсилювач, на вході якого об'єднуються сигнали завдання та сигнали різних каналів зворотного зв'язку. Зворотні зв'язки можуть мати різну природу: жорсткий або гнучкий ЗЗ за напругою генератора, ЗЗ за швидкістю обертання двигуна, а також зворотний зв'язок за струмом із відсіченням. У даному електроприводі використовується система, що поєднує ЗЗ за швидкістю, гнучкий ЗЗ за струмом та канал відсічення струму.

Функціональну структуру такого електроприводу подано на рис. 2.4.

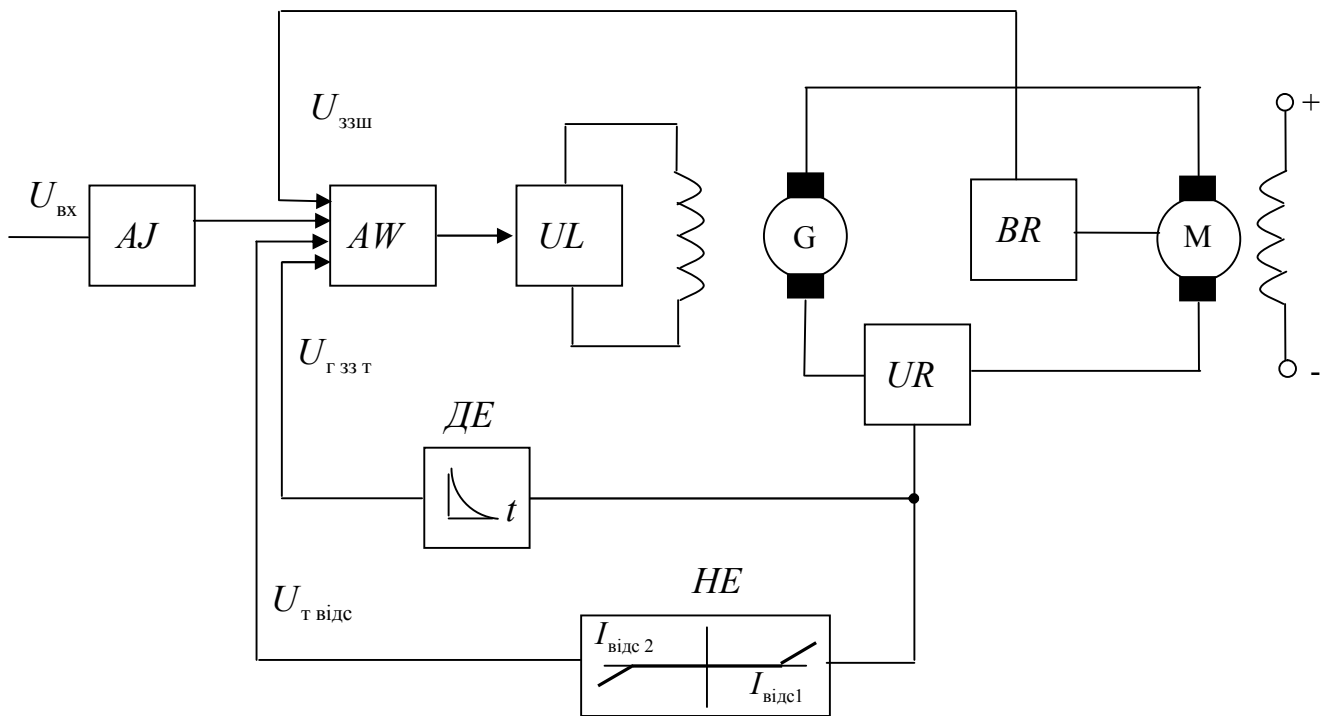


Рисунок 2.4 – Функціональна структурна схема системи автоматичного керування електроприводом типу Г–Д.

На схемі застосовано такі позначення: AJ – блок задання інтенсивності; AW – підсумовуючий підсилювач; UL – збудник генератора; UA – датчик струму; BR – датчик швидкості; HE – нелінійний елемент; DE – диференціюючий елемент

Задавач інтенсивності та гнучкі канали зворотного зв'язку не впливають на формування статичних властивостей системи. Основну нелінійність, що визначає характер статичних характеристик, вносить нелінійний елемент, який розміщений у контурі струмового ЗЗ з відсіченням. Його передаточна функція включає зону нечутливості, завдяки якій зворотний зв'язок не активується, поки струм якоря знаходиться у діапазоні

$$I_{\text{відс}2} \leq I_{\text{я}} \leq I_{\text{відс}1}.$$

Для даної системи приймається співвідношення:

$$I_{\text{відс}1} = |I_{\text{відс}2}| = I_{\text{відс}} = 2 I_{\text{нд}}.$$

Характеристика замкненої системи записується як:

$$\omega = U_3 \frac{k_\Sigma}{c + k_\Sigma k_{\text{ззш}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{c + k_\Sigma k_{\text{ззш}}} - 1(\Delta I)(I_{\text{я}} - I_{\text{відс}}) \frac{k_{\text{т відс}} k_\Sigma}{c + k_\Sigma k_{\text{ззш}}} =$$

$$= \omega_0 - \Delta\omega_{\text{ст 1}} - 1(\Delta I)\Delta\omega_{\text{ст 2}},$$

де $k_\Sigma = k_{AW} k_{UL} k_\Gamma$ – коефіцієнт підсилення прямого каналу, який обчислюється як добуток коефіцієнтів підсилення складових системи AW , UL і G на лінійній частині їх характеристик вхід-вихід;

$k_{\text{ззш}}$, $k_{\text{т відс}}$ – коефіцієнти підсилення зворотного зв'язку за швидкістю і струму з відсіченням;

$$c = k_{\text{д}} \Phi_{\text{нд}};$$

$1 \cdot (\Delta I)$ – функція Хевісайда, яка дорівнює 0 при $|I_{\text{я}}| \leq I_{\text{відс}}$ і дорівнює 1 при $|I_{\text{я}}| > I_{\text{відс}}$.

Коефіцієнт посилення генератора

$$k_\Gamma = \frac{E_{\text{нг}}}{U_{\text{нзб}\Gamma}} = \frac{U_{\text{нг}} + I_{\text{нг}} R_{\text{я}\Sigma\Gamma}}{U_{\text{нзб}\Gamma}},$$

де $U_{\text{нг}}$, $U_{\text{нзб}\Gamma}$ – напруга на якорі і напруга збудження генератора;

$I_{\text{нг}}$, $E_{\text{нг}}$ – струм та ЕРС генератора;

$$k_\Gamma = \frac{750 + 2535 \cdot 0,01}{220} = 3,4.$$

Коефіцієнт підсилення тиристорного збудника:

$$k_{UL} = \frac{U_{\text{днзб}}}{U_y},$$

де $U_{\text{днзб}}$ – номінальна випрямлена напруга збудника, приймаємо $U_{\text{днзб}} = U_{\text{нзб}\Gamma}$;

U_y – напруга управління на вході тиристорного збудника.

Оскільки каталожні дані по U_y відсутні, приймаємо $U_y = 10 \text{ В}$.

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

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

Розраховуємо коефіцієнт підсилення прямого каналу:

$$k_{\Sigma} = 0,1 \cdot 23 \cdot 3,4 = 7,82.$$

При визначенні коефіцієнта зворотного зв'язку за швидкістю $k_{ззш}$ виходимо з бажаного статичного перепаду швидкості $\Delta\omega_{ст1}$ при номінальному струмі двигуна:

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

звідки

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

Величину $\Delta\omega_{ст1}$ прийmemo такою, що дорівнює половині статичного падіння швидкості в розімкненій системі $\Delta\omega_{ст}$ при номінальному навантаженні, тобто

$$\Delta\omega_{ст1} = 0,5 \Delta\omega_{ст} = 0,5(\omega_{01} - \omega_{н1}),$$

де ω_{01} , $\omega_{н1}$ – раніше визначені значення швидкості ідеального холостого ходу та швидкості під номінальним навантаженням для розімкненого контуру.

$$\Delta\omega_{с01} = 0,5 \cdot (60 - 55,44) = 2,28 \text{ с}^{-1},$$

$$k_{ззш} = 2280 \cdot \frac{0,021}{8,1 \cdot 2,28} - \frac{10,5}{8,1} = 1,3.$$

Визначення коефіцієнта струмового відсічення $k_{т відс}$ здійснюють з метою

забезпечення необхідного значення максимального струму стопоріння. Для цього використовується співвідношення:

$$k_{\text{т відс}} = \frac{U_{\text{з max}} - I_{\text{стоп}} R_{\Sigma} k_{\Sigma}}{I_{\text{стоп}} - I_{\text{відс}}}$$

У подальших розрахунках приймаємо такі параметри:

$$I_{\text{відс}} = 2I_{\text{нд}} = 2 \cdot 2280 = 4560 \text{ A}; \quad I_{\text{стоп}} = 2,3I_{\text{нд}} = 2,3 \cdot 2280 = 5244 \text{ A}.$$

Значення напруги завдання $U_{\text{з max}}$ обирається виходячи з умови, що при привід повинен забезпечувати швидкість, яка відповідає ідеальному режиму холостого ходу розімкненої електромеханічної системи за умови $E_{\Gamma} = U_{\text{нд}}$.

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

звідки

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

$$U_{\text{з max}} = \frac{630 \cdot (10,5 + 7,82 \cdot 1,3)}{10,5 \cdot 7,82} = 158,6 \text{ B},$$

$$k_{\text{т відс}} = \frac{158,6 - 5244 \cdot 0,021 / 7,82}{5244 - 4560} = 0,21.$$

Таким чином, після підстановки числових значень рівняння електромеханічної характеристики замкнутої системи (Г-Д) для управління електроприводом роторного колеса екскаватора набуває вигляду:

$$\begin{aligned} \omega &= U_{\text{з}} \cdot \frac{7,82}{10,5 + 7,82 \cdot 1,3} - I_{\text{я}} \frac{0,021}{10,5 + 7,82 \cdot 1,3} - 1(\Delta I)(I_{\text{я}} - 4560) \frac{0,21 \cdot 7,82}{10,5 + 7,82 \cdot 1,3} = \\ &= U_{\text{з}} \cdot 0,378 - I_{\text{я}} \cdot 0,001016 - 1 \cdot (\Delta I) \cdot (I_{\text{я}} - 4560) \cdot 0,0795. \end{aligned}$$

Для побудови характеристик розрахунок виконуємо для трьох значень напруги завдання U_3 , а саме: $U_3 = U_{3\max} = 158,6 \text{ В}$, $U_3 = 0,5U_{3\max} = 0,5 \cdot 158,6 = 79,3 \text{ В}$, $U_3 = 0$.

1 Точка ідеального холостого ходу

$$I_{\text{я}} = 0, \quad M = 0.$$

$$\omega_{01} = 158,6 \cdot 0,378 = 60 \text{ с}^{-1},$$

$$\omega_{02} = 79,3 \cdot 0,378 = 30 \text{ с}^{-1},$$

$$\omega_{03} = 0 \cdot 0,378 = 0.$$

2 Точка, що відповідає подвійному номінальному навантаженню

$$I_{\text{я}} = I_{\text{відс}} = 4560 \text{ А},$$

$$\omega_1 = 60 - 4560 \cdot 0,001016 = 55,44 \text{ с}^{-1},$$

$$\omega_2 = 30 - 4560 \cdot 0,001016 = 25,44 \text{ с}^{-1},$$

$$\omega_3 = 0 - 4560 \cdot 0,001016 = -4,56 \text{ с}^{-1}.$$

3 Точка, що відповідає струму стопоріння

$$\omega_{1,2} = 0;$$

$$I_1 = I_{\text{стоп1}} = 5244 \text{ А}.$$

Струми стопоріння для характеристики при $U_3 = 79,3 \text{ В}$ визначаються виразу для електромеханічної характеристики. Якщо підставити $\omega = 0$, отримаємо $I_{\text{стоп2}} = 4902 \text{ А}$.

Електромеханічні характеристики замкнутої системи, побудовані для трьох значень напруги завдання та для обох напрямків обертання ротора, показані на рисунку 2.4 суцільними лініями.

2.5 Аналіз динамічних характеристик системи

Для вивчення динамічних властивостей електроприводу розглянемо алгоритмічну схему, наведеною на рис. 2.5, яка побудована на основі функціональної схеми, показаної на рис. 2.4.

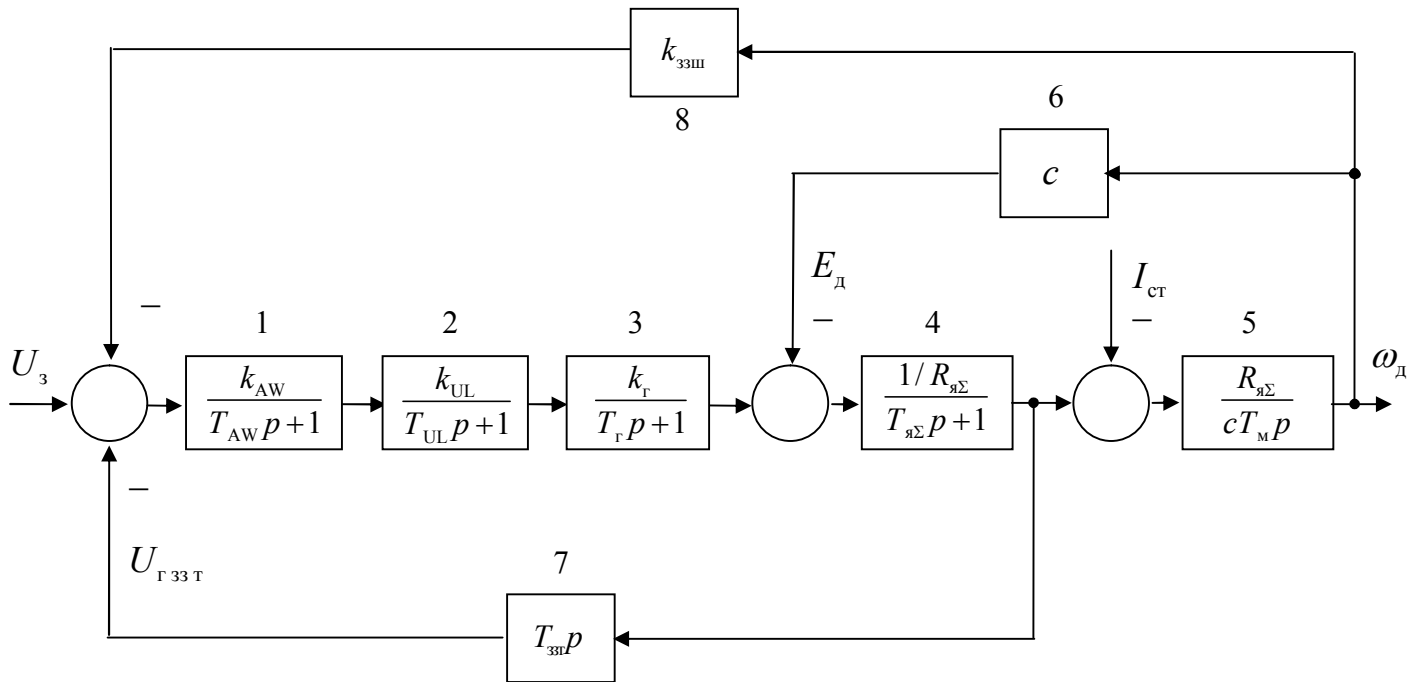


Рисунок 2.5 – Алгоритмічна схема системи автоматичного управління

У цій схемі відсутній зворотний зв'язок по струму з відсіченням, оскільки він не впливає на робочі режими, а також не включено задатчик інтенсивності, оскільки його ефект можна врахувати через подачу завдання, яке змінюється лінійно у часі.

Постійна часу генератора приблизно дорівнює постійній часу його незалежної обмотки збудження (за умови відсутності паралельної обмотки та нехтування вихровими струмами):

$$T_{\Gamma} = T_{\text{зб } \Gamma} = \frac{2 p_{\Gamma} W_{\text{зб}} k_{\Phi}}{R_{\text{зб}}},$$

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

p_{Γ} – кількість пар полюсів генератора;

k_{Φ} – коефіцієнт нахилу кривої намагнічення в лінійній ділянці, приймається приблизно $k_{\Phi} = \Delta\Phi / \Delta I_{3\phi} = 0,29 \cdot 10^{-2} \text{ Ом} \cdot \text{с}$.

$$k_{\Phi} = \frac{\Delta\Phi}{\Delta I_{3\phi}} = 0,29 \cdot 10^{-2} \text{ Ом} \cdot \text{с};$$

$$T_{\Gamma} = T_{3\phi \Gamma} = \frac{6 \cdot 330 \cdot 0,29 \cdot 10^{-2}}{5,73} = 1,002 \text{ с}.$$

Електромагнітна постійна часу якорного кола визначається як:

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

де $L_{я\Sigma}$ – сумарна індуктивність якорного кола, що враховує індуктивності двигуна $L_{яд}$ та $L_{яг}$ генератора:

$$L_{я\Sigma} = k L_{яд} + L_{яг},$$

Індуктивності $L_{яд}$ та $L_{яг}$ можна оцінити за наближеними формулами:

$$L_{яд(\Gamma)} = k_k \frac{U_{нд(\Gamma)}}{p_{д(\Gamma)} \omega_{нд(\Gamma)} I_{нд(\Gamma)}}, \quad \dots$$

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

$U_{нд(\Gamma)}$, $I_{нд(\Gamma)}$, $\omega_{нд(\Gamma)}$ – номінальні значення напруги, струму, і швидкості, двигуна або генератора;

$p_{д(\Gamma)}$ – кількість пар полюсів двигуна або генератора.

Номінальна швидкість генератора визначається як:

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

$$\omega_{\text{нГ}} = \frac{3,14 \cdot 750}{30} = 78,4 \text{ с}^{-1},$$

Підставляємо числові значення параметрів та визначаємо $L_{\text{я}\Sigma}$ і $T_{\text{я}\Sigma}$:

$$L_{\text{яД}} = 0,25 \cdot \frac{630}{3 \cdot 57,6 \cdot 2280} = 0,0004 \text{ Гн},$$

$$L_{\text{яГ}} = 0,25 \cdot \frac{750}{3 \cdot 78,4 \cdot 2535} = 0,00031 \text{ Гн},$$

$$L_{\text{я}\Sigma} = 0,0004 + 0,00031 = 0,00071 \text{ Гн.},$$

$$T_{\text{я}\Sigma} = \frac{0,00071}{0,021} = 0,034 \text{ с}.$$

Електромеханічна постійна часу системи обчислюється за формулою:

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

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

$$T_{\text{м}} = \frac{1,3 \cdot 775 \cdot 0,021}{10,5^2} = 0,19 \text{ с}.$$

Постійна часу $T_{\text{згт}}$ визначається за критерієм Гурвіца. Характеристичне рівняння замкнутої системи Г-Д має вигляд:

$$T_{\Gamma} T_M T_{\Sigma} \cdot p^3 + \left(T_M T_{\Sigma} + T_{\Gamma} T_M + \frac{k_{\Sigma} T_M}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_M) p + k_{\Sigma} k_{\Sigma\text{ш}} k_{\text{д}} = 0,$$

або

$$a_0 p^3 + a_1 p^2 + a_2 p + a_3 = 0.$$

Умова стійкості системи формулюється як:

$$a_1 a_2 - a_0 a_3 > 0,$$

$$a_0 = 0,0065, \quad a_1 = 0,2 + 43,2 T_{\Sigma\text{ш}}, \quad a_2 = 1,192, \quad a_3 = 0,88.$$

Після проведення розрахунків отримуємо умову що задовольняє умові стійкості:

$$73,2 \cdot T_{\Sigma\text{т}} > -0,1952.$$

Умова стійкості виконується для будь-якого обраного значення $T_{\Sigma\text{т}}$. Далі, шляхом комп'ютерного моделювання системи визначаються оптимальні значення параметру $T_{\Sigma\text{т}}$ для забезпечення необхідних динамічних характеристик.

Для отримання передавальних функцій системи по задаючій та збурюючій діях можна спростити розрахунки, виключивши з розгляду постійну часу напівпровідникового суматора AW та тиристорного збудника UL , оскільки вони більш ніж у 50 разів менші за основну постійну часу системи T_{Γ} . На основі цього спрощення формується алгоритмічна схема, наведена на рис. 2.6.

За допомогою цієї схеми можливо вивести передавальні функції системи як по задаючій, так і по збурюючій дії.

Передавальна функція швидкості по задаючій дії $W^1(p) = \frac{\omega_d(p)}{U_3(p)}$ визнача-

ється після спрощення схеми ($I_{\text{ст}} = 0$) та перетворення її до вигляду, показаного на рис. 2.7 а.

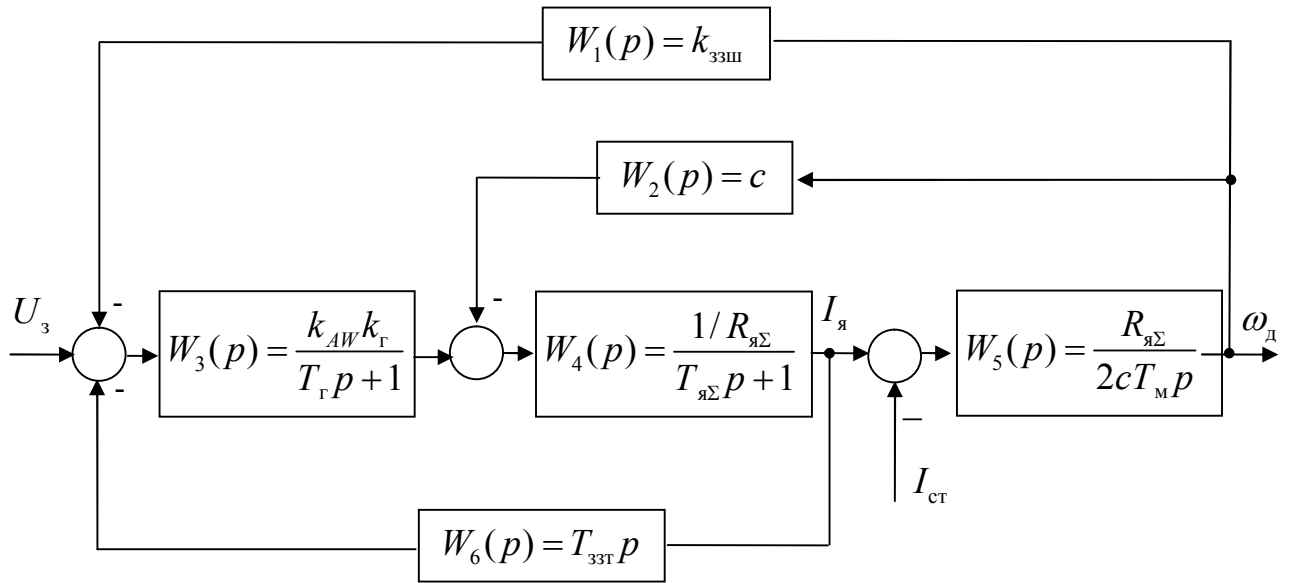


Рисунок 2.6 – Спрощена алгоритмічна схема системи, отримана при усуненні малих постійних постійних суматора і збудника

Паралельно-зустрічне з'єднання ланок схеми трансформується у форму, представлену на рис. 2.7 б, що дозволяє визначити передавальну функцію швидкості по задаючій дії за формулою:

$$\begin{aligned}
 W^1(p) &= \frac{\omega_d(p)}{U_3(p)} = \frac{W_3(p)W_4(p)W_5(p)}{1 + \left(W_1(p) + \frac{W_2(p)}{W_3(p)} + \frac{W_6(p)}{W_5(p)} \right) \cdot W_3(p)W_4(p)W_5(p)} = \\
 &= \frac{W_3(p)W_4(p)W_5(p)}{1 + W_1(p) \cdot W_3(p)W_4(p)W_5(p) + W_2(p)W_4(p)W_5(p) + W_3(p)W_4(p)W_6(p)}. \quad (2.2)
 \end{aligned}$$

Передавальні функції, що входять у (2.2):

$$W_1(p) = k_{33ш}, \quad (2.3)$$

$$W_2(p) = c, \quad (2.4)$$

$$W_3(p) = \frac{k_{AW} \cdot k_{\Gamma}}{T_{\Gamma} p + 1}, \quad (2.5)$$

$$W_4(p) = \frac{1/R_{\Sigma}}{T_{\Sigma} p + 1}, \quad (2.6)$$

$$W_5(p) = \frac{R_{\Sigma}}{c T_M p}, \quad (2.7)$$

$$W_6(p) = T_{3\Gamma} p. \quad (2.8)$$

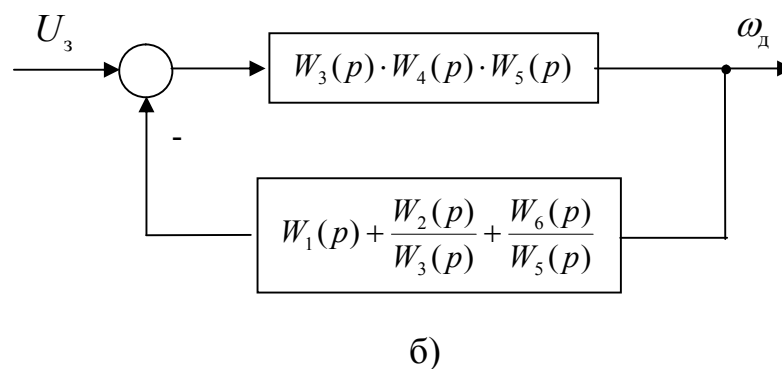
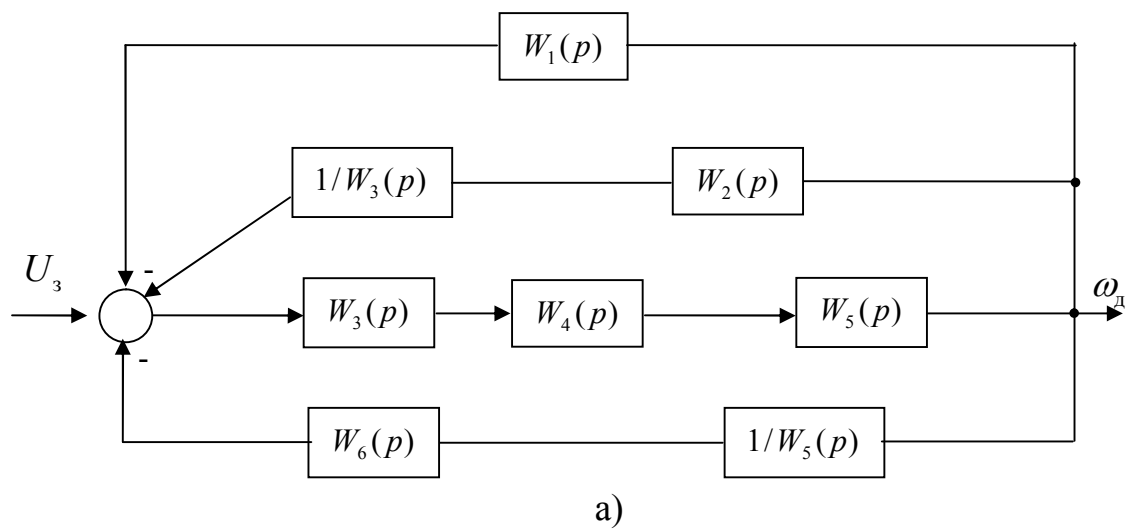


Рисунок 2.7 – Етапи перетворення алгоритмічної схеми рис. 2.6 для визначення

передавальної функції швидкості по задаючій дії $W^1(p) = \frac{\omega_d(p)}{U_3(p)}$:

а) перший етап перетворення ;

б) другий етап перетворення

Підставляємо передавальні функції (2.3) – (2.8) в (2.2) і отримаємо

$$\begin{aligned}
 W^1(p) &= \frac{\omega_d(p)}{U_3(p)} = \\
 &= \frac{\frac{k_{AW} \cdot k_\Gamma}{T_\Gamma p + 1} \cdot \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot \frac{R_{я\Sigma}}{c T_M p}}{1 + k_{33\Pi} \cdot \frac{k_{AW} \cdot k_\Gamma}{T_\Gamma p + 1} \cdot \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot \frac{R_{я\Sigma}}{c T_M p} + c \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot \frac{R_{я\Sigma}}{c T_M p} + \frac{k_{AW} \cdot k_\Gamma}{T_\Gamma p + 1} \cdot \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot T_{33T} p} = \\
 &= \frac{k_{AW} k_\Gamma (T_\Gamma p + 1)(T_{я\Sigma} p + 1) T_M p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1) T_M p \cdot c \left[(T_\Gamma T_{я\Sigma} p^2 + (T_\Gamma + T_{я\Sigma}) p + 1) T_M p + \frac{k_{AW} k_\Gamma k_{33\Pi}}{c} + T_\Gamma p + 1 + \frac{k_{AW} k_\Gamma T_{OT} T_M p^2}{R_{я\Sigma}} \right]} = \\
 &= \frac{k_{AW} \cdot k_\Gamma / c}{T_\Gamma T_{я\Sigma} T_M p^3 + (T_\Gamma + T_{я\Sigma}) T_M p^2 + T_M p + \frac{k_{AW} k_\Gamma k_{33\Pi}}{c} + T_\Gamma p + 1 + \frac{k_{AW} k_\Gamma T_{33T} T_M p^2}{R_{я\Sigma}}} = \\
 &= \frac{k_{AW} \cdot k_\Gamma / c}{T_\Gamma T_{я\Sigma} T_M p^3 + T_M \left(T_\Gamma + T_{я\Sigma} + k_{AW} k_\Gamma \frac{T_{33T}}{R_{я\Sigma}} \right) p^2 + (T_\Gamma + T_M) p + k_{AW} k_\Gamma \cdot \frac{k_{33\Pi}}{c} + 1}.
 \end{aligned}$$

Враховуючи позначення для k_d і k_Σ , маємо:

$$\begin{aligned}
 W^1(p) &= \frac{\omega_d(p)}{U_3(p)} = \\
 &= \frac{k_\Sigma k_d}{T_\Gamma T_{я\Sigma} T_M p^3 + \left(T_M T_\Gamma + T_M T_{я\Sigma} + \frac{k_\Sigma T_M T_{33T}}{R_{я\Sigma}} \right) p^2 + (T_\Gamma + T_M) p + k_\Sigma k_{33\Pi} k_d + 1}.
 \end{aligned}$$

Аналогічно проводиться визначення передавальної функції струму по задаючій дії $W^2(p) = \frac{I_\Sigma(p)}{U_3(p)}$, для чого алгоритмічна схема трансформується відповідно до етапів, показаних на рис. 2.8.

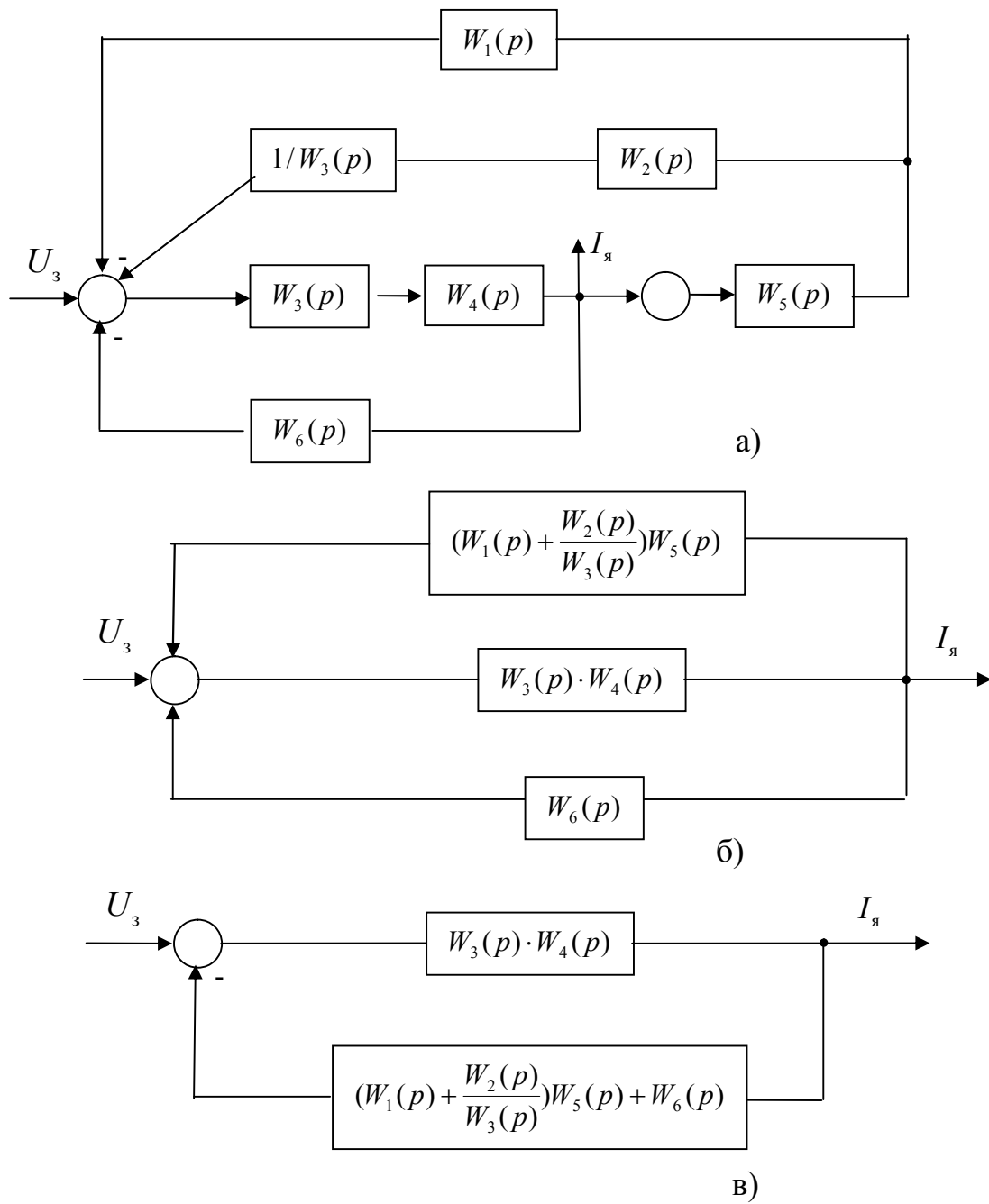


Рисунок 2.8 – Перетворення схеми рис. 2.7,б для визначення передавальної функції струму по задаючій дії $W^2(p) = \frac{I_я(p)}{U_3(p)}$.

- а) перший етап перетворення;
- б) другий етап перетворення
- в) третій етап перетворення

$$W_3(p) \cdot W_4(p) = \frac{k_{AW} \cdot k_{\Gamma}}{T_{\Gamma} p + 1} \cdot \frac{1/R_{\Sigma}}{T_{\Sigma} p + 1} = \frac{k_{AW} \cdot \frac{k_{\Gamma}}{R_{\Sigma}}}{T_{\Gamma} T_{\Sigma} p^2 + (T_{\Gamma} + T_{\Sigma}) p + 1}.$$

Визначаємо передавальну функцію струму по задаючій дії

$$\begin{aligned} W^2(p) &= \frac{I_{\Sigma}(p)}{U_3(p)} = \frac{W_3(p)W_4(p)}{1 + \left[\left(W_1(p) + \frac{W_2(p)}{W_3(p)} \right) \cdot W_5(p) + W_6(p) \right] \cdot W_3(p)W_4(p)} = \\ &= \frac{\left(k_{AW} k_{\Gamma} \frac{1}{R_{\Sigma}} \right) / [(T_{\Gamma} p + 1)(T_{\Sigma} p + 1)]}{1 + \left[\left(k_{33\Pi} + \frac{c(T_{\Gamma} p + 1)}{k_{AW} k_{\Gamma}} \right) \frac{R_{\Sigma}}{c T_{\Sigma} p} + T_{33\Gamma} p \right] \cdot \frac{k_{AW} k_{\Gamma} / R_{\Sigma}}{(T_{\Gamma} p + 1)(T_{\Sigma} p + 1)}} = \\ &= \frac{\left(k_{AW} k_{\Gamma} \frac{1}{R_{\Sigma}} \right) / [(T_{\Gamma} p + 1)(T_{\Sigma} p + 1)]}{(T_{\Gamma} p + 1)(T_{\Sigma} p + 1) \left\{ (T_{\Gamma} p + 1)(T_{\Sigma} p + 1) + \left[\left(k_{33\Pi} + \frac{c(T_{\Gamma} p + 1)}{k_{AW} k_{\Gamma}} \right) \frac{R_{\Sigma}}{c T_{\Sigma} p} + T_{33\Gamma} p \right] \cdot \frac{k_{AW} k_{\Gamma}}{R_{\Sigma}} \right\}} = \\ &= \frac{k_{AW} k_{\Gamma} / R_{\Sigma}}{(k_{AW} k_{\Gamma} / R_{\Sigma}) \left[\frac{(T_{\Gamma} p + 1)(T_{\Sigma} p + 1) c R_{\Sigma} T_{\Sigma} p + [k_{AW} k_{\Gamma} k_{33\Pi} + c(T_{\Gamma} p + 1)] R_{\Sigma} + c T_{33\Gamma} p k_{AW} k_{\Gamma} T_{\Sigma} p}{c k_{AW} \cdot k_{\Gamma} T_{\Sigma} p} \right]} = \\ &= \frac{k_{\Gamma} k_{AW} T_{\Sigma} p / R_{\Sigma}}{T_{\Gamma} T_{\Sigma} T_{\Sigma} p^3 + T_{\Sigma} \left(T_{\Gamma} + T_{\Sigma} + \frac{k_{\Gamma} k_{AW} T_{33\Gamma}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\Sigma}) p + k_{\Gamma} k_{AW} k_{33\Pi} k_{\Sigma} + 1} = \\ &= \frac{k_{\Sigma} T_{\Sigma} p / R_{\Sigma}}{T_{\Gamma} T_{\Sigma} T_{\Sigma} p^3 + \left(T_{\Sigma} T_{\Gamma} + T_{\Sigma} T_{\Sigma} + \frac{k_{\Sigma} T_{\Sigma} T_{33\Gamma}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\Sigma}) p + k_{\Sigma} k_{33\Pi} k_{\Sigma} + 1}. \end{aligned}$$

Для отримання передавальної функції швидкості по збурюючій дії $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$ застосовуються перетворення схеми з рис. 2.6 у вигляді, представленою на рис. 2.9 та рис. 2.10, з урахуванням $U_3 = 0$.

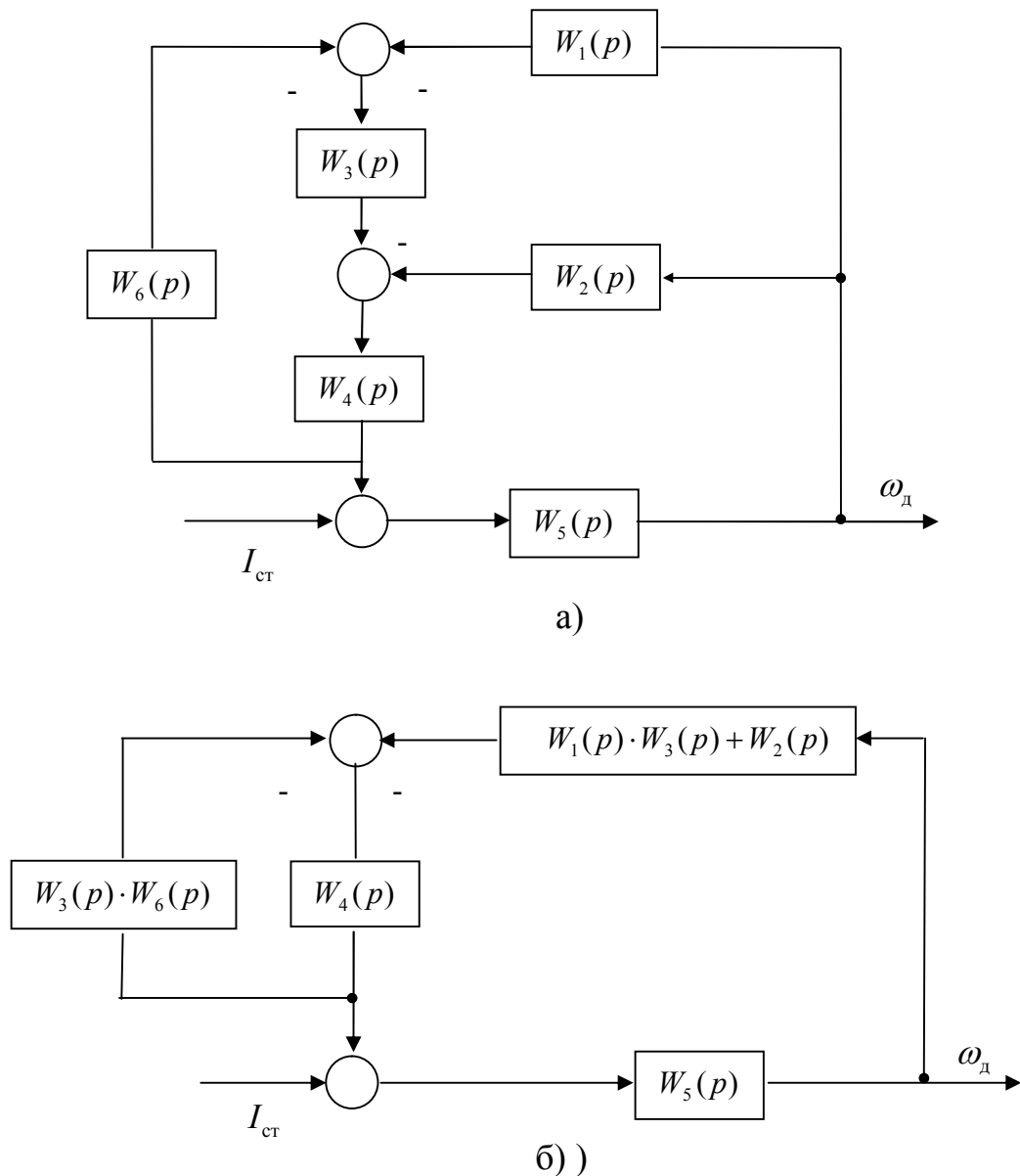
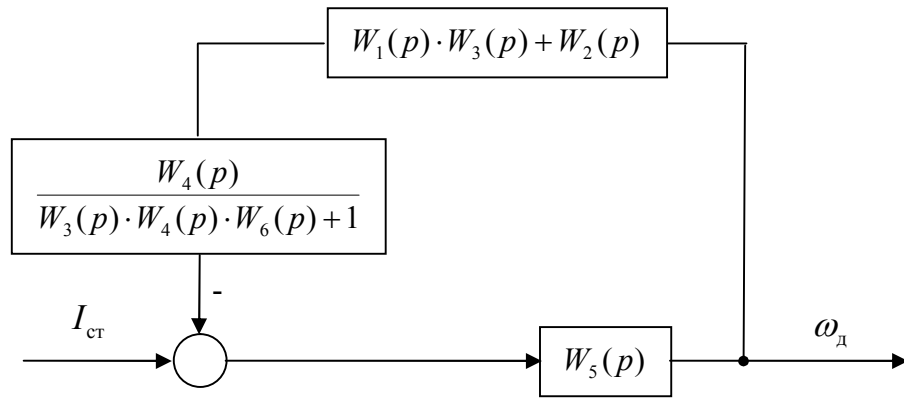


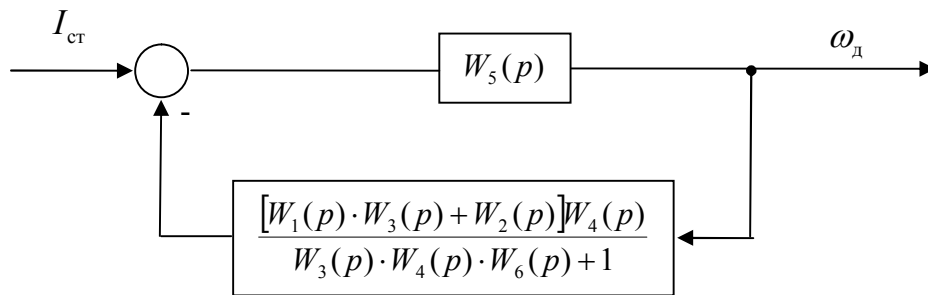
Рисунок 2.9 – Перетворення схеми рис. 2.6 для визначення передавальної функції швидкості по збурюючій дії $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$:

а) перший етап перетворення;

б) другий етап перетворення



а)



б)

Рисунок 2.10 – Подальші етапи перетворення схеми рис. 2.9,б для визначення

передавальної функції швидкості по збудуючій дії $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$:

а) перший етап перетворення;

б) другий етап перетворення

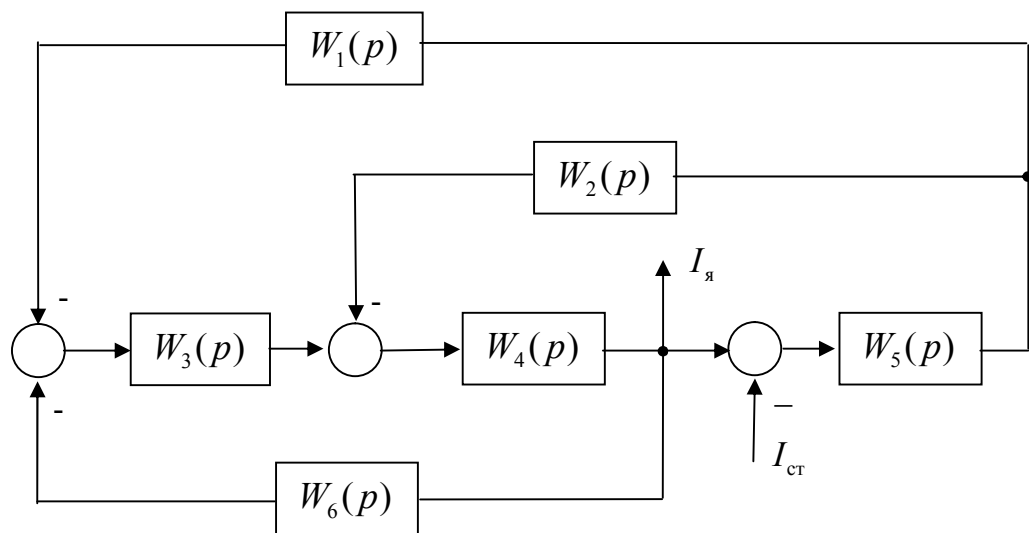
$$W_5(p) = \frac{R_{я\Sigma}}{cT_m p}$$

$$\begin{aligned}
 W^3(p) &= \frac{\omega_d(p)}{I_{ст}(p)} = \frac{W_5(p)}{\frac{[W_1(p)W_3(p) + W_2(p)] \cdot W_4(p)W_5(p)}{1 + W_3(p)W_4(p)W_6(p)} + 1} = \\
 &= \frac{[1 + W_3(p)W_4(p)W_6(p)]W_5(p)}{W_5(p) \left[[W_1(p)W_3(p) + W_2(p)]W_4(p) + \frac{1 + W_3(p)W_4(p)W_6(p)}{W_5(p)} \right]} =
 \end{aligned}$$

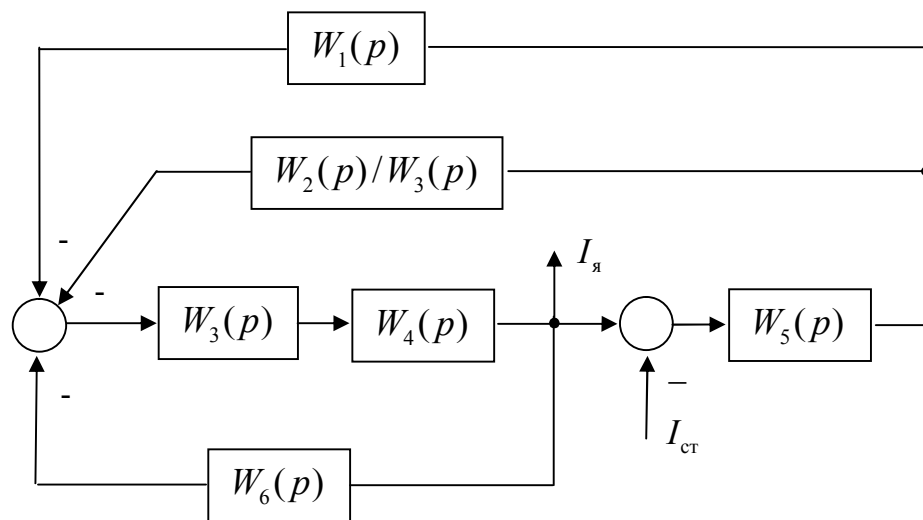
$$\begin{aligned}
&= \frac{1 + W_3(p)W_4(p)W_6(p)}{[W_1(p)W_3(p) + W_2(p)]W_4(p) + \frac{1 + W_3(p)W_4(p)W_6(p)}{W_5(p)}} = \\
&= \frac{1 + \frac{k_\Sigma / R_{я\Sigma} \cdot T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1)}}{\left(\frac{k_{33Ш} k_\Sigma}{T_\Gamma p + 1} + c \right) \frac{1 / R_{я\Sigma}}{T_{я\Sigma} p + 1} + \frac{\left(1 + \frac{k_\Sigma / R_{я\Sigma} T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1)} \right) c T_\Gamma p}{R_{я\Sigma}}} = \\
&= \frac{\frac{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1) + k_\Sigma T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1) R_{я\Sigma}}}{\frac{k_\Sigma k_{33Ш} + c(T_\Gamma p + 1) + \left((T_\Gamma p + 1)(T_{я\Sigma} p + 1) + \frac{k_\Sigma}{R_{я\Sigma}} \cdot T_{33Т} p \right) c T_\Gamma p}{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1)}} = \\
&= \frac{R_{я\Sigma} k_\Sigma \left[T_\Gamma T_{я\Sigma} p^2 + \left(T_\Gamma + T_{я\Sigma} + \frac{k_\Sigma T_{33Т}}{R_{я\Sigma}} \right) p + 1 \right]}{T_\Gamma T_{я\Sigma} T_\Gamma p^3 + \left(T_\Gamma T_\Gamma + T_\Gamma T_{я\Sigma} + \frac{k_\Sigma T_\Gamma T_{33Т}}{R_{я\Sigma}} \right) p^2 + (T_\Gamma + T_\Gamma) p + k_\Sigma k_{33Ш} k_\Sigma + 1}.
\end{aligned}$$

Передавальна функція струму по збурюючій дії $W^4(p) = \frac{I_\Gamma(p)}{I_{ст}(p)}$ визнача-

ється за схемою, послідовно перетвореною на рис. 2.11 та рис. 2.12.



а)



б)

Рисунок 2.11 – Перетворення схеми рис. 2.8 а для визначення передавальної

функції струму по збурюючій дії $W^4(p) = \frac{I_{я}(p)}{I_{ст}(p)}$:

а) перший етап перетворення;

б) другий етап перетворення

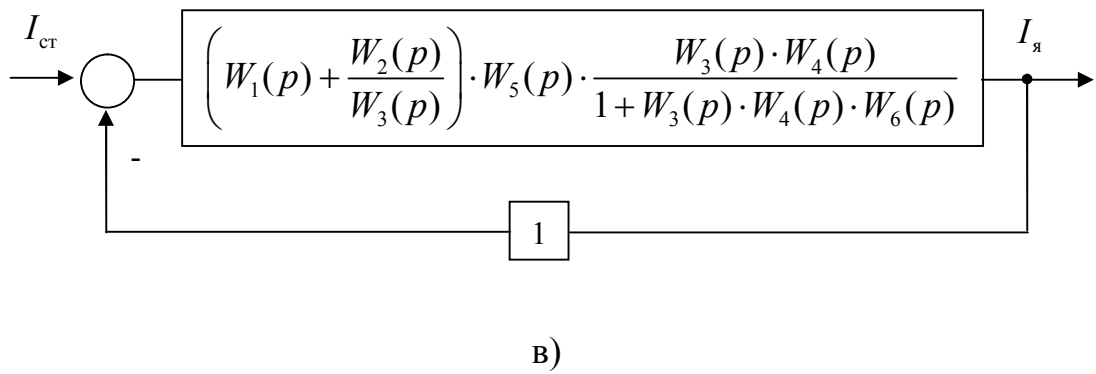
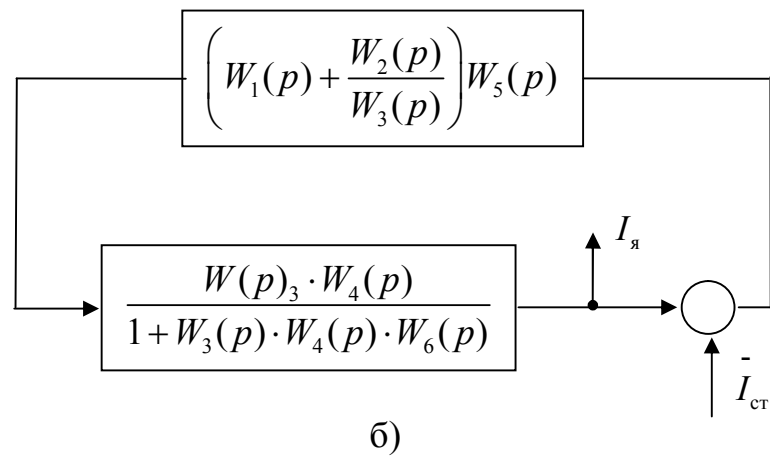
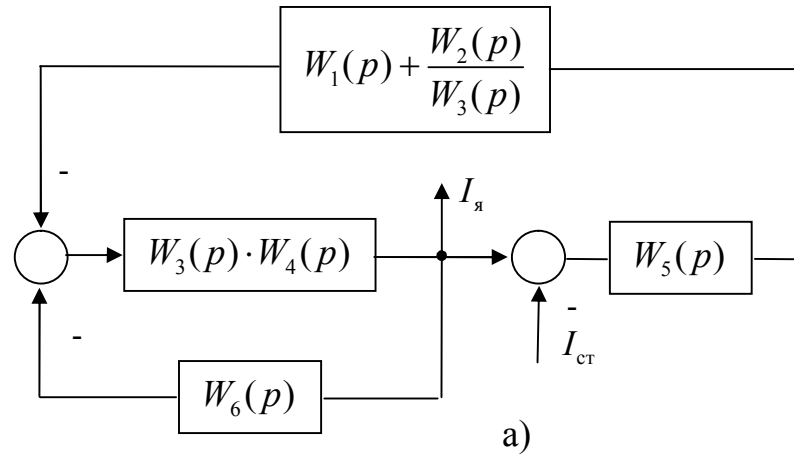


Рисунок 2.12 – Подальші етапи перетворення схеми рис. 2.11,б для визначення

передавальної функції струму по збурюючій дії $W^4(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$:

а) перший етап перетворення;

б) другий етап перетворення;

в) третій етап перетворення

$$\begin{aligned}
W(p) &= \left(W_1(p) + \frac{W_2(p)}{W_3(p)} \right) \cdot W_5(p) \cdot \frac{W_3(p)W_4(p)}{1 + W_3(p)W_4(p)W_5(p)} = \\
&= \left(k_{33\Pi} + \frac{c(T_\Gamma p + 1)}{k_\Sigma} \right) \cdot \frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot \frac{\frac{k_\Sigma}{T_\Gamma p + 1} \cdot \frac{1/R_{\text{я}\Sigma}}{T_{\text{я}\Sigma} p + 1}}{1 + \frac{k_\Sigma}{T_\Gamma p + 1} \cdot \frac{1/R_{\text{я}\Sigma}}{T_{\text{я}\Sigma} p + 1} \cdot T_{33\Gamma} p} = \\
&= \left(\frac{k_{33\Pi} k_\Sigma + c(T_\Gamma p + 1)}{k_\Sigma} \right) \cdot \frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot \frac{\frac{k_\Sigma}{T_\Gamma p + 1} \cdot \frac{1/R_{\text{я}\Sigma}}{T_{\text{я}\Sigma} p + 1}}{\frac{(T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p}{(T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1)}} = \\
&= \frac{k_{33\Pi} k_\Sigma + c(T_\Gamma p + 1)}{cT_{\text{м}}p \left((T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p \right)} = \\
&= \frac{T_\Gamma p + k_{33\Pi} k_\Sigma k_{\text{д}} + 1}{T_{\text{м}}p \left((T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p \right)} \\
W^4(p) &= \frac{I_{\text{я}}(p)}{I_{\text{сг}}(p)} = \frac{\frac{T_\Gamma p + k_{33\Pi} k_\Sigma k_{\text{д}} + 1}{T_{\text{м}}p \left((T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p \right)}}{1 + \frac{T_\Gamma p + k_{33\Pi} k_\Sigma k_{\text{д}} + 1}{T_{\text{м}}p \left((T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p \right)}} = \\
&= \frac{T_\Gamma p + k_{33\Pi} k_\Sigma k_{\text{д}} + 1}{T_{\text{м}}p \left[(T_\Gamma p + 1)(T_{\text{я}\Sigma} p + 1) + k_\Sigma T_{33\Gamma} / R_{\text{я}\Sigma} p \right] + T_\Gamma p + k_{33\Pi} k_\Sigma k_{\text{д}} + 1} =
\end{aligned}$$

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

На основі отриманих передавальних функцій $W^1(p) \div W^4(p)$ здійснюється розрахунок перехідних характеристик системи. Розрахунки перехідних процесів проводяться у MATLAB та перевіряються моделлю в SIMULINK (рис. 2.13).

Графіки перехідних процесів швидкості $\omega_{\text{д}}$ та струму $I_{\text{я}}$ по задаючій та збурюючій діях наведені на рис. 2.13 та рис. 2.14 відповідно. Розрахунки виконані для трьох значень постійної часу зворотного зв'язку за струмом $T_{\text{ззт}}$: 1 – $T_{\text{ззт}} = 0,005\text{с}$; 2 – $T_{\text{ззт}} = 0,015\text{с}$; 3 – $T_{\text{ззт}} = 0,03\text{с}$. Найкращі показники якості перехідних процесів спостерігаються при $T_{\text{ззт}} = 0,005\text{с}$.

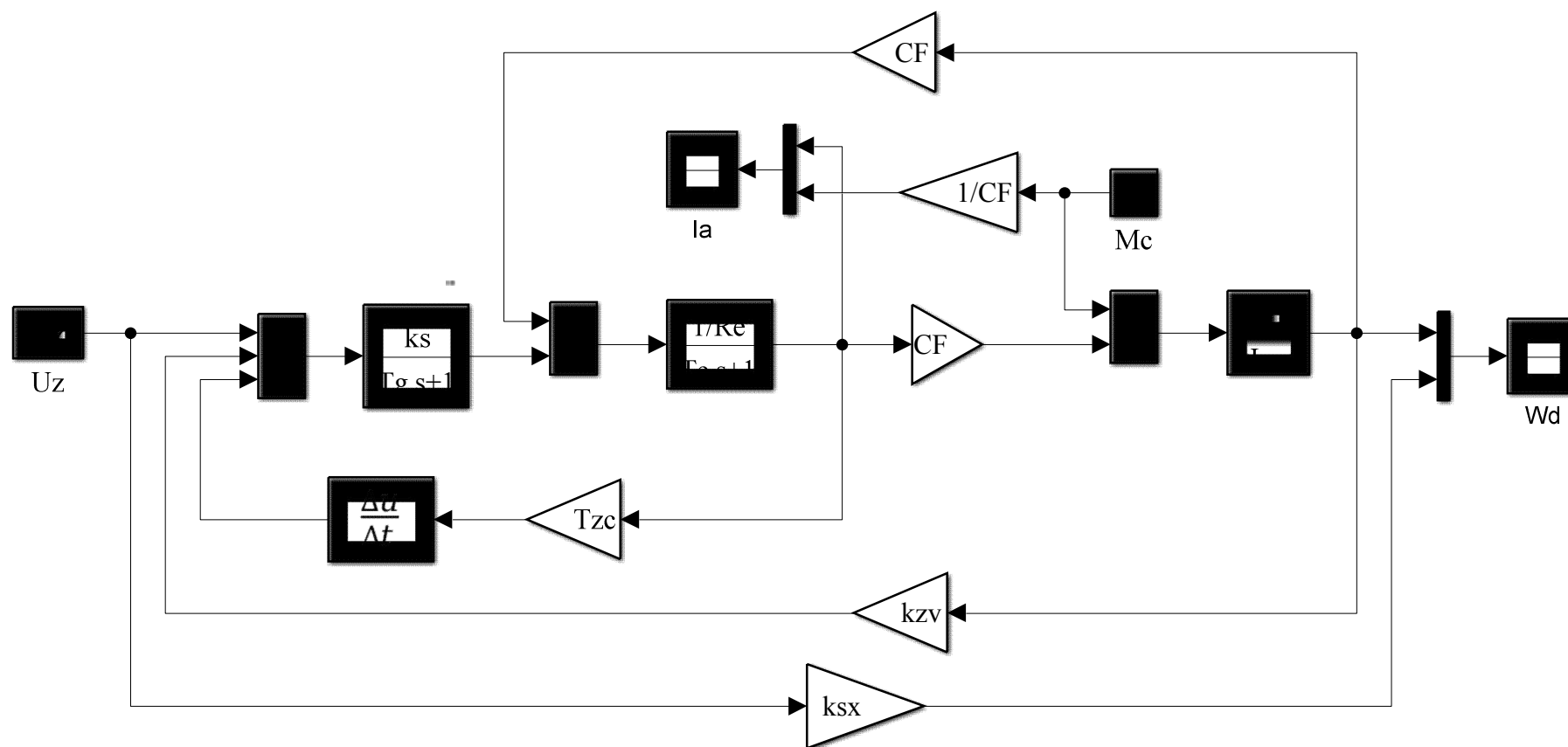


Рисунок 2.12 – Імітаційна модель системи, реалізована у MATLAB/SIMULINK

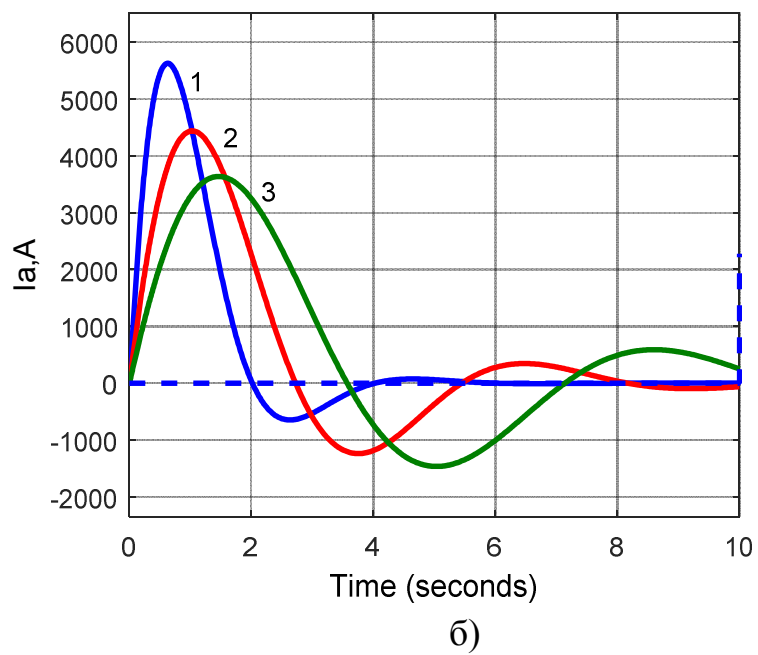
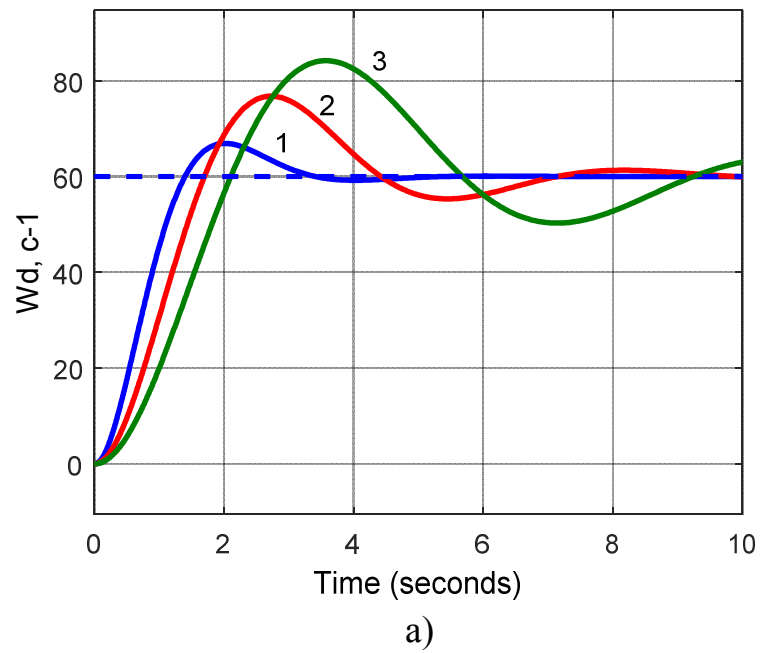
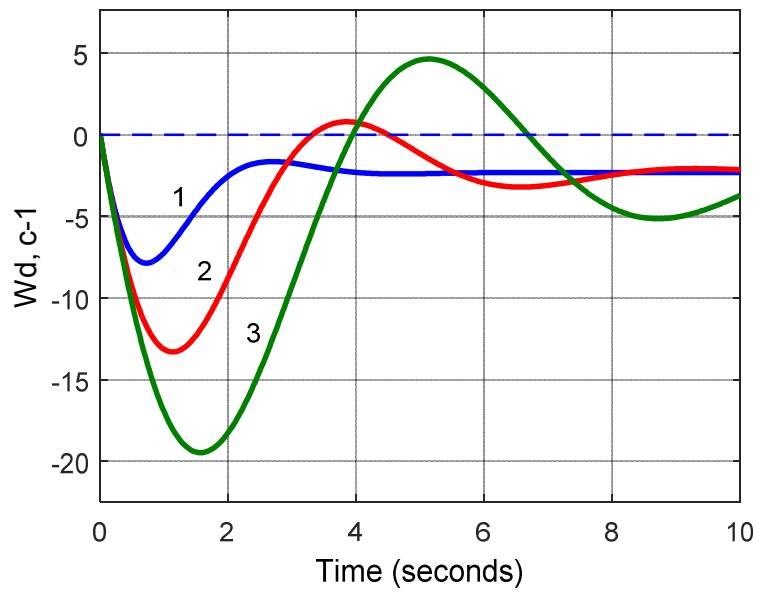
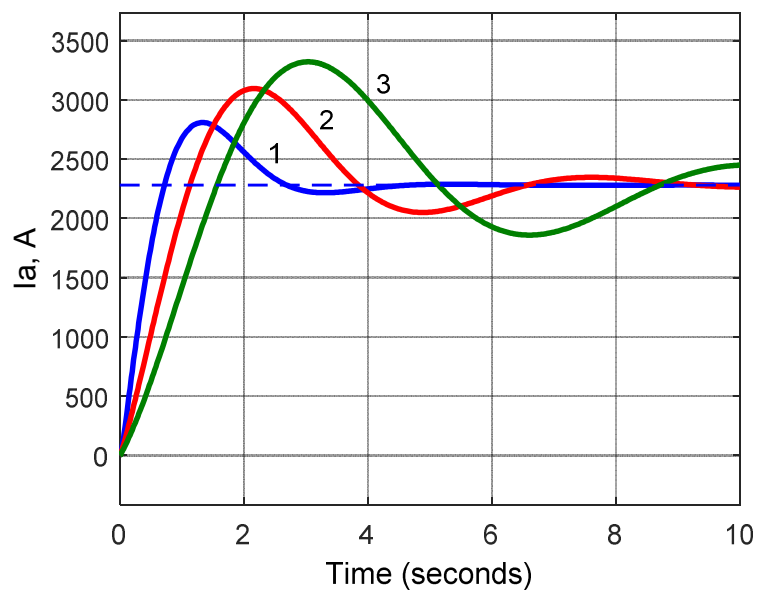


Рисунок 2.13 – Перехідні процеси швидкості (а) і струму (б)
при прикладенні задаючого сигналу:

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



a)



б)

Рисунок 2.14 – Перехідні процеси швидкості (а) і струму (б)
при прикладенні збурюючого впливу:

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

ВИСНОВКИ ДО РОЗДІЛУ 2

1. У ході виконання розділу здійснено обґрунтований вибір типу електродвигуна та основних елементів силового кола електроприводу. Підібране обладнання забезпечує необхідні енергетичні та експлуатаційні характеристики для роботи приводу в умовах роторного екскаватора.

2. Визначено та розраховано параметри силового ланцюга, що є базовими для подальшого синтезу системи автоматичного управління. Отримані величини використовуються як вхідні дані при формуванні математичної моделі та під час налаштування контурів регулювання.

3. Проведено аналіз можливих варіантів структури САУ та обґрунтовано вибір оптимальної конфігурації. З урахуванням високих вимог до жорсткості робочої ділянки механічної характеристики та необхідності обмеження максимального моменту, який може створювати електропривод механізмів роторного екскаватора, обрано систему керування зі зрозумілою та ефективною структурою. Вона включає контур зворотного зв'язку за швидкістю, гнучкий зворотний зв'язок за струмом якоря, а також контур обмеження струму (з відсіченням), що дозволяє формувати так звану "екскаваторну" механічну характеристику.

4. Виконано аналітичний розрахунок статичних характеристик як розімкненої, так і замкненої системи керування. На основі отриманих рівнянь побудовано алгоритмічну модель приводу, визначено передавальні функції по задаючих і збурюючих діях, що дозволило дослідити вплив параметрів керування на поведінку системи.

5. Проведено моделювання синтезованої системи автоматичного управління засобами MATLAB та її середовища SIMULINK. Створена імітаційна модель дала змогу отримати перехідні процеси для різних варіантів параметрів і проаналізувати динамічні властивості системи. За результатами моделювання визначено раціональне значення постійної часу контуру зворотного зв'язку по струму, яке забезпечує належну якість перехідних процесів та стабільність роботи приводу.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

У попередніх розділах механічна частина електроприводу розглядалася у спрощеному вигляді – як абсолютно жорстка приведена ланка. Такий підхід не враховує реальний вплив пружних властивостей механічних елементів, з яких складається привід (валів, муфт, зубчастих передач тощо). Оскільки жорсткість цих складових є кінцевою, реальна механічна система має пружний характер. Тому прикладання до неї керуючого моменту двигуна або дії зовнішнього статичного навантаження супроводжується виникненням коливань взаємопов'язаних мас. Це призводить до збільшення пікових динамічних навантажень та може ускладнювати точне відтворення необхідного закону руху робочого органа машини.

Якщо виокремити основні інерційні маси, що визначають динаміку механізму, а також елементи з найменшою жорсткістю у кінематичному ланцюзі, то конструкцію механізму обертання роторного колеса доцільно звести до математичної моделі двомасової механічної системи, поданої на рис. 3.1.

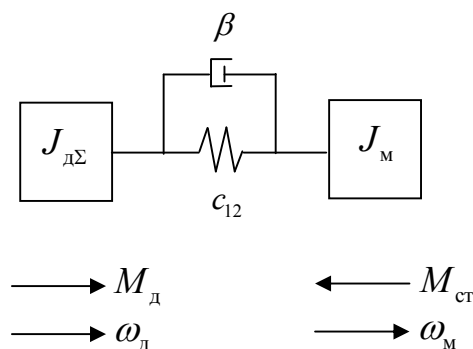


Рисунок 3.1 – Двомасова розрахункова схема

Параметрами такої системи є: сумарний момент інерції якоря електродвигуна разом із жорстко зв'язаними з ним деталями - $J_{дΣ}$; приведений до швид-

кості двигуна момент інерції роторного колеса (механізму) – $J_{\text{м}}$; пружний безінерційний зв'язок між масами, що описується еквівалентною жорсткістю c_{12} та коефіцієнтом внутрішнього в'язкого тертя каната β .

Рух двомасової моделі описується системою диференціальних рівнянь:

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

де M_{Σ} - сумарний момент у пружному елементі, що дорівнює сумі пружної складової $M_{\text{пр}}$ і моменту в'язкого тертя $M_{\text{вт}} = \beta(\omega_{\text{д}} - \omega_{\text{м}})$:

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

$\omega_{\text{д}}$, $\omega_{\text{м}}$ – кутова швидкість двигуна та механізму відповідно;

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

Перетворимо систему (3.1) з урахуванням (3.2) до стандартного вигляду

Коші:

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

У попередніх розділах розроблено одномасову модель електромеханічної частини приводу, яка може бути описана диференціальними рівняннями:

$$\left\{ \begin{array}{l} T_{\text{м}} \frac{d\omega_{\text{д}}}{dt} = \frac{R_{\text{я}\Sigma}}{c} (I_{\text{я}} - I_{\text{ст}}); \\ T_{\text{я}\Sigma} \frac{dI_{\text{я}}}{dt} + I_{\text{я}} = \frac{1}{R_{\text{я}\Sigma}} (E - c\omega_{\text{д}}); \\ T_{\text{г}} \frac{dE_{\text{г}}}{dt} + E_{\text{г}} = k_{\Sigma} U_{\text{з}} - k_{\text{ззш}} k_{\Sigma} \omega_{\text{д}} - k_{\Sigma} U_{\text{г зз т}}; \\ T_{\text{ззт}} \frac{dI_{\text{я}}}{dt} = U_{\text{г зз т}}. \end{array} \right. \quad (3.4)$$

Записавши систему (3.4) у формі Коші та врахувавши співвідношення $I_{\text{ст}} = M_{\text{ст}}/c$, $T_{\text{м}} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}$, об'єднаємо її з (3.3). Після нескладних математичних перетворень отримаємо розширені рівняння стану для повної двомасової системи:

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

Введемо вектор стану

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

та вектор зовнішніх дій

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

У векторно-матричній формі система записується так:

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

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

$\mathbf{B}$ – матриця керування;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор вихідних величин.

Структура матриць $\mathbf{A}$ та $\mathbf{B}$ має вигляд:

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

Алгоритмічна схема двомасової системи, представлена на рис.3.2, наочно демонструє взаємозв'язки між електромагнітними та механічними процесами, а також враховує пружну взаємодію між двома інерційними ланками приводу.

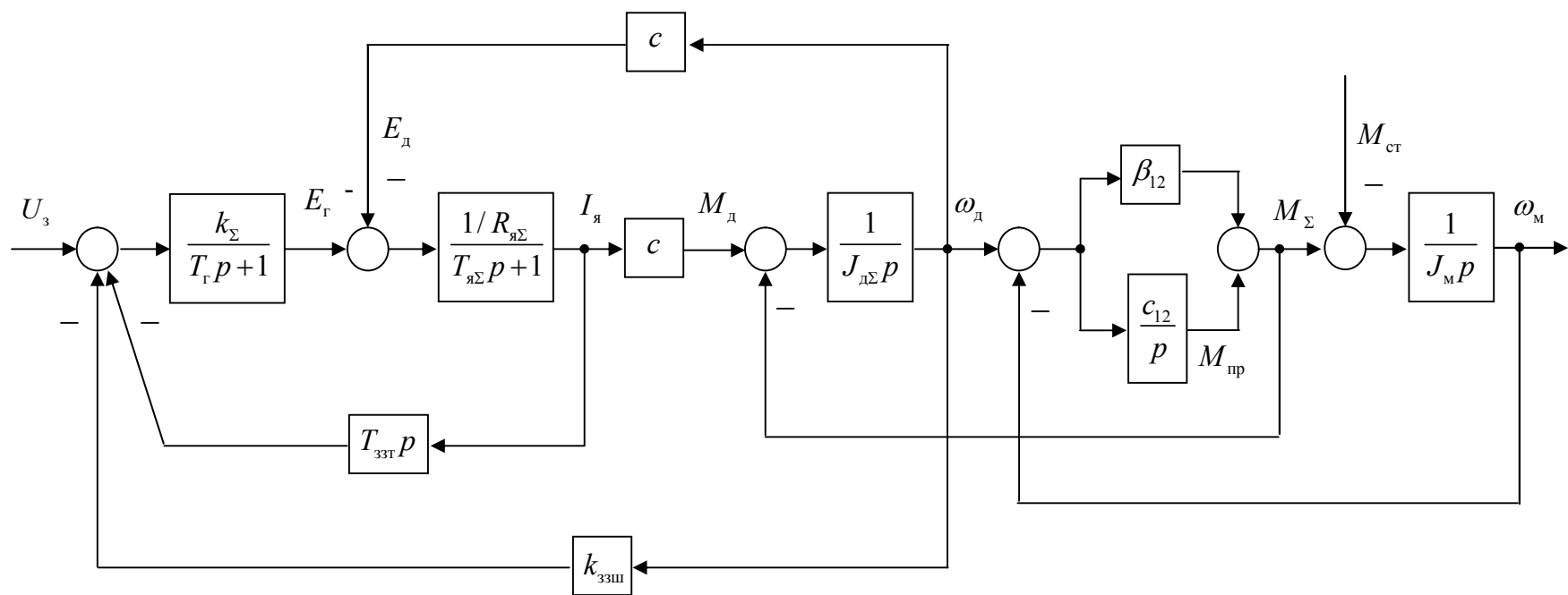


Рисунок 3.2– Алгоритмічна схема двомасової моделі системи

3.2 Розрахунок перехідних процесів у двомасовій системі

На основі математичного опису двомасового електромеханічного об'єкта у формі рівнянь стану були проведені обчислення перехідних процесів для випадків дії як задаючих, так і збурюючих впливів. Усі числові експерименти виконувались на персональній ЕОМ із застосуванням пакету прикладних програм MATLAB.

Початкові дані та матриці **A**, **B**, **C** наведені у файлі, розміщеному в додатку Г. Для моделювання динаміки використовувалася модель SIMULINK із базовим блоком *State-Space*, структура якої подана на рис. 3.3. У параметрах цього блоку задаються імена матриць, що описують двомасову систему.

Кількість вихідних координат (розмір вектора $\vec{Y}(t)$) та, відповідно, число вікон *Scope* для відображення графіків перехідних процесів, визначається на етапі проведення моделювання. Інтерфейс введення матриць **A**, **B**, **C** та **D** подано на рис. 3.4. Це вікно відкривається під час подвійного натискання на блок *State-Space*.

Матриця вихідних величин приймається одиничною, якщо аналіз проводиться для всіх змінних стану. У разі дослідження лише окремих координат одиниці залишаються тільки в тих рядках головної діагоналі, які відповідають необхідному набору сигналів виходу $\vec{Y}(t)$. Матриця **D**, як і в стандартному випадку, дорівнює нульовій.

Для перевірки коректності результатів та порівняння динамічних режимів дослідження виконувалися також за допомогою альтернативної структурної моделі системи, наведеної на рис. 3.5.

Графічні результати моделювання представлені на рис. 3.6–3.9. На діаграмах показано часові залежності перших чотирьох змінних стану. Із наведених графіків видно наявність суттєвих коливальних компонент у динаміці координат двомасового механізму. Це свідчить про підвищене динамічне навантаження на елементи приводу та може негативно впливати на довговічність і стабільність роботи механізму в перехідних режимах.

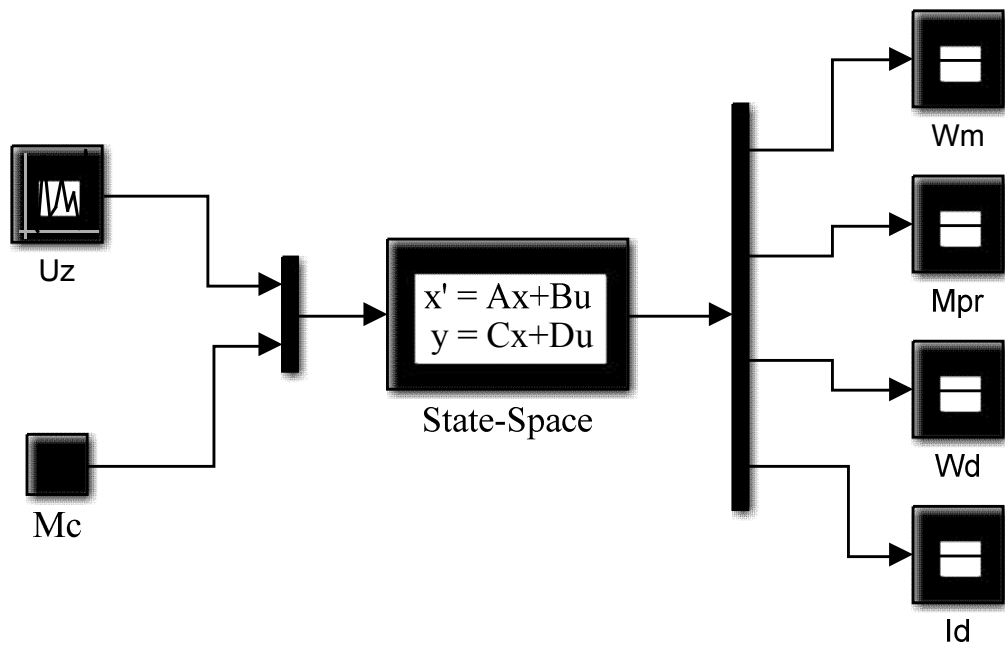


Рисунок 3.3 – Модель системи в середовищі SIMULINK із використанням блоку *State-Space*

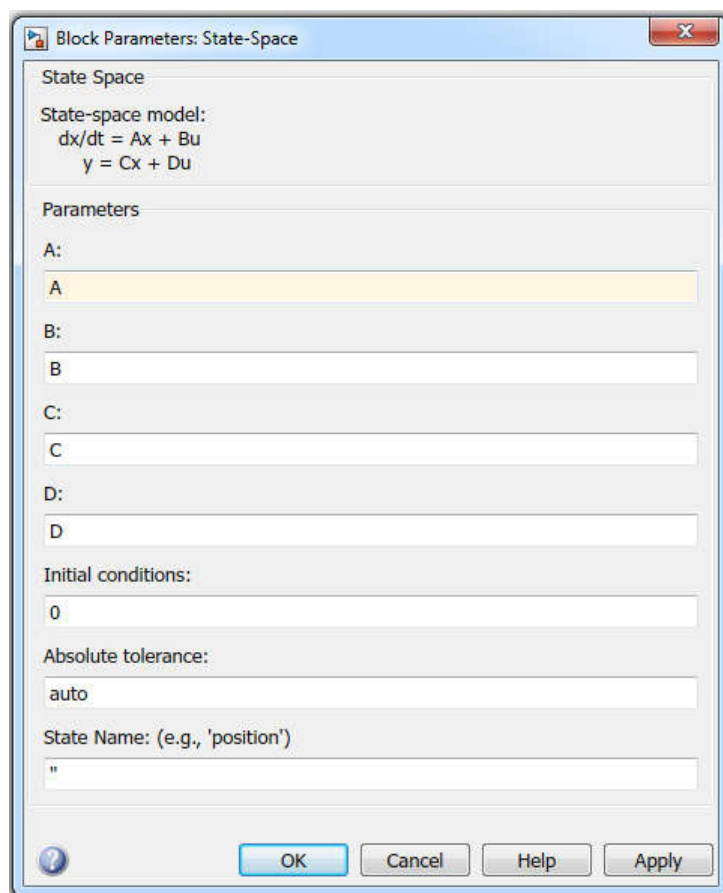


Рисунок 3.4 – Вікно введення матриць **A**, **B**, **C**, **D** для блоку State-Space

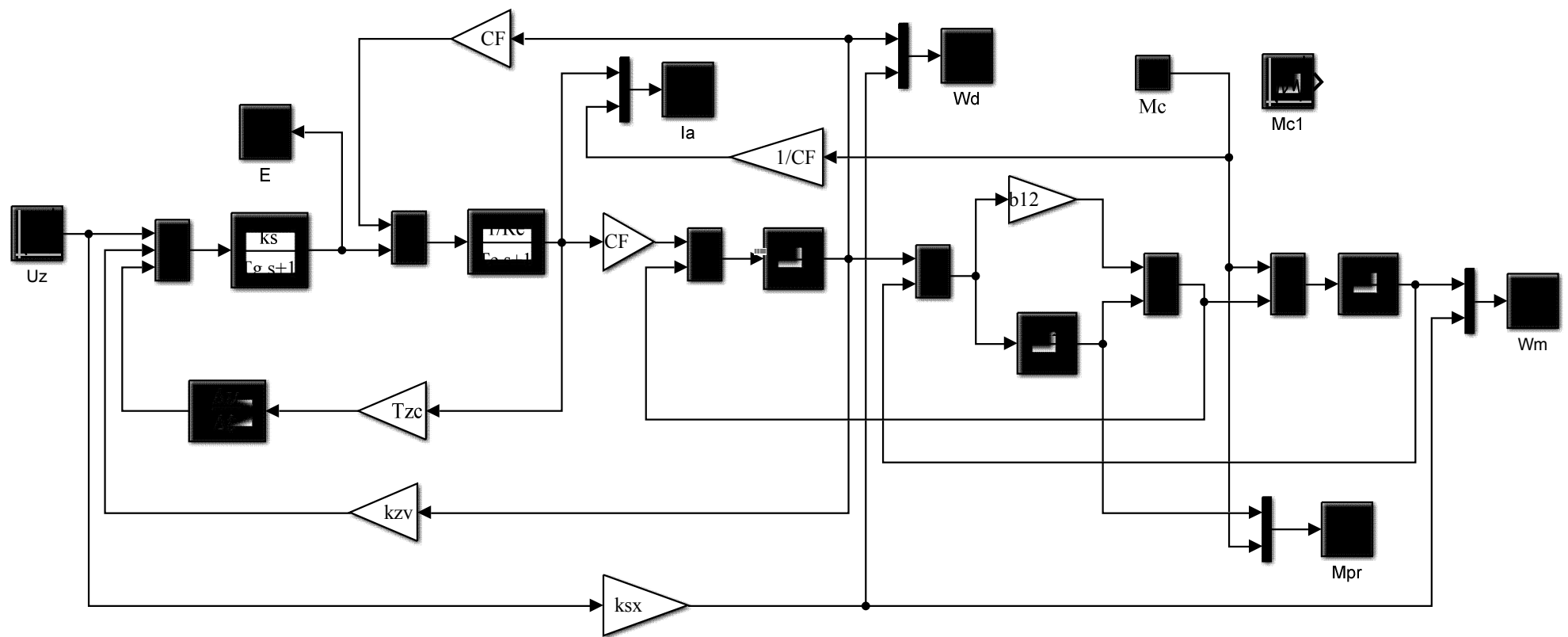


Рисунок 3.5 – Імітаційна модель двомасової системи, реалізована у середовищі MATLAB/SIMULINK

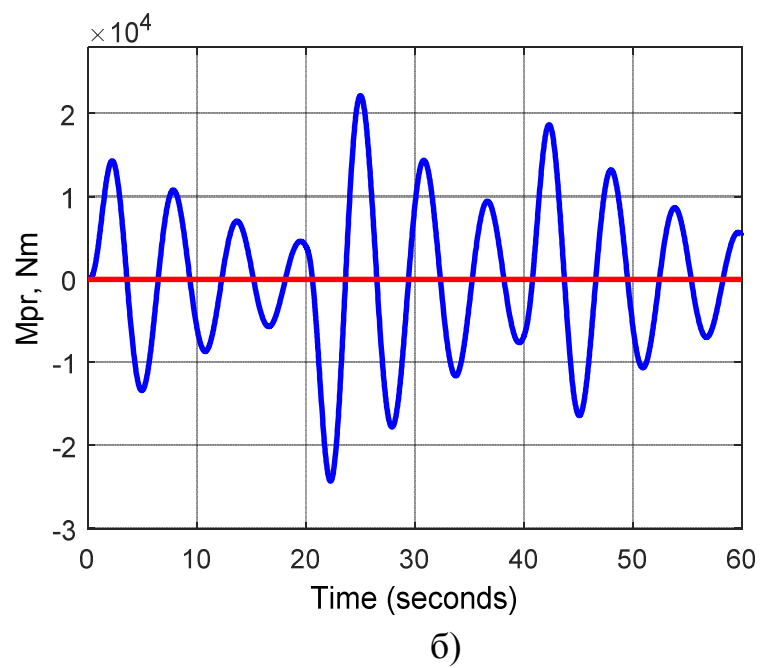
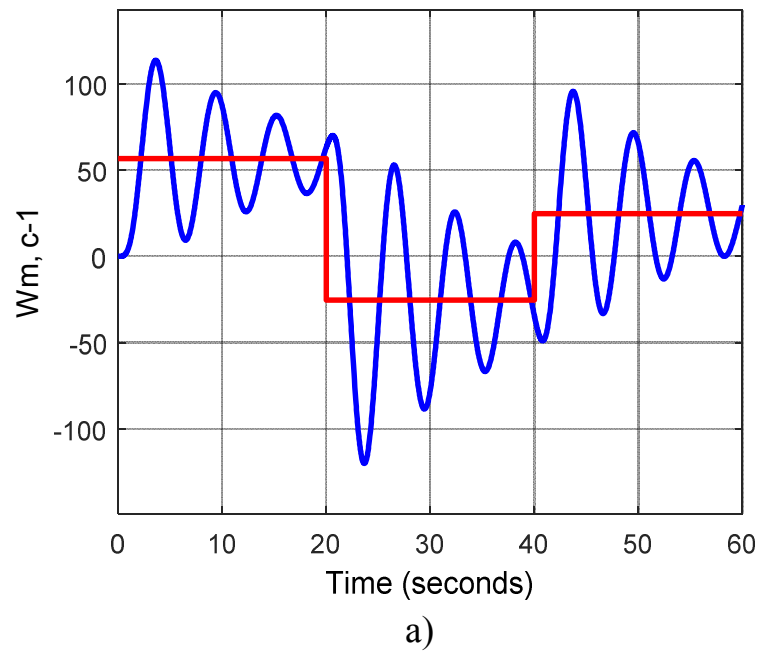


Рисунок 3.6 – Перехідні процеси швидкості механізму $\omega_m = f(t)$ (а) та моменту пружності $M_{пр} = f(t)$ (б) двомасової системи за задаючим впливом

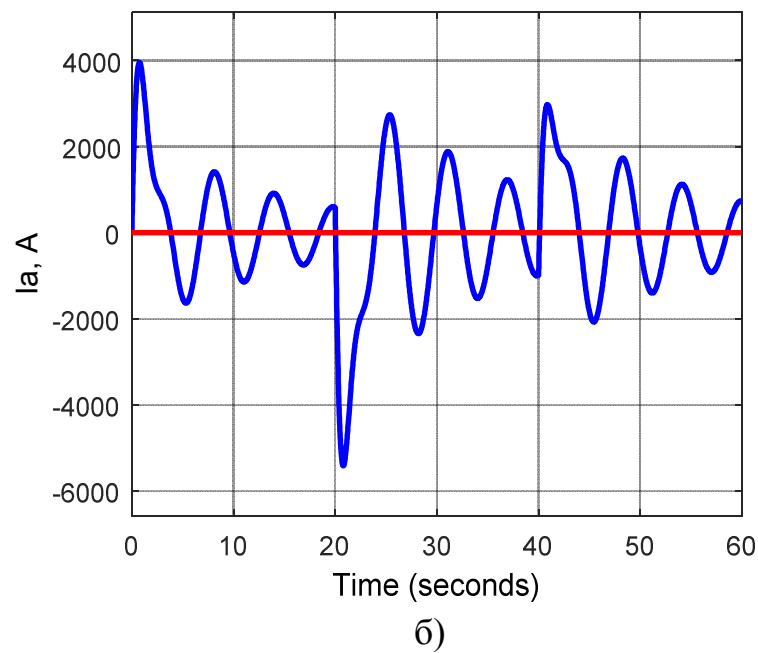
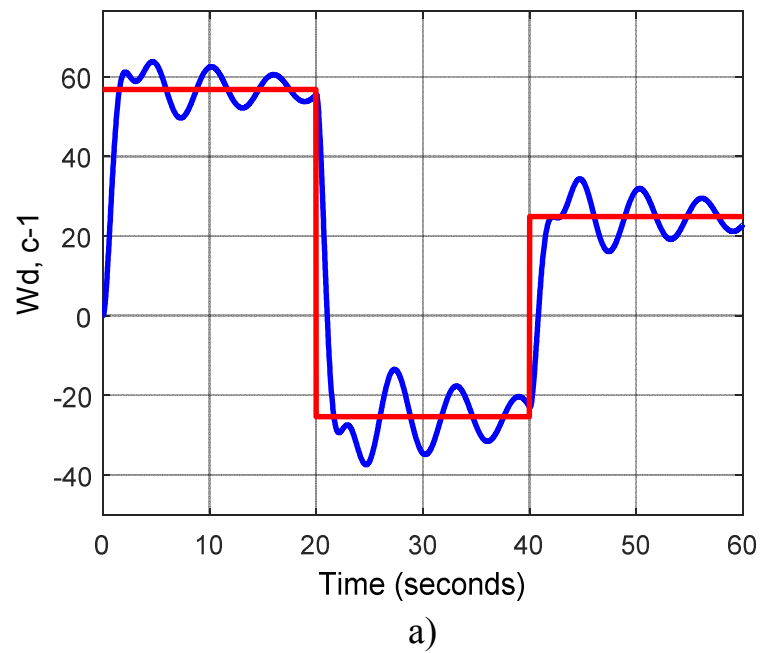


Рисунок 3.7 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) та струму двигуна $I_a = f(t)$ двомасової системи за задаючим впливом

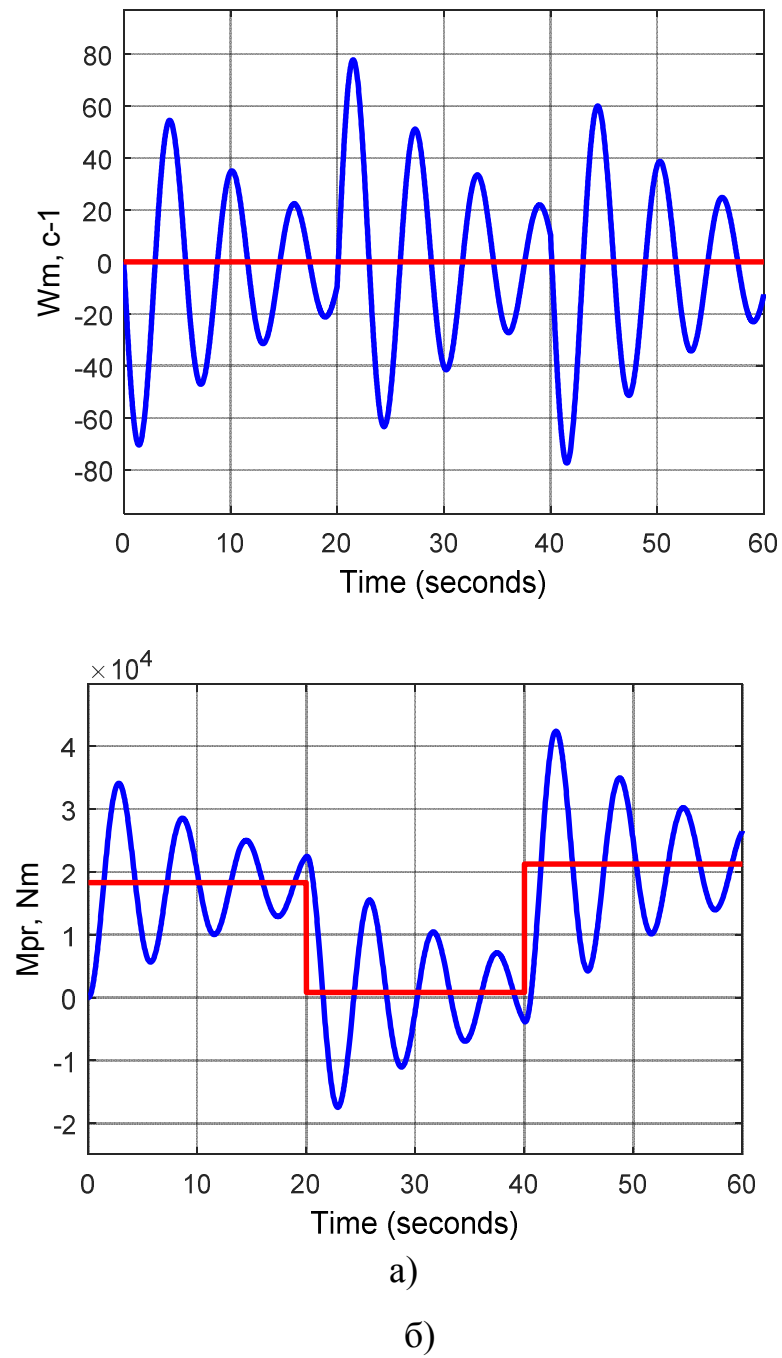


Рисунок 3.8 – Перехідні процеси швидкості механізму $\omega_m = f(t)$ (а) та моменту пружності $M_{пр} = f(t)$ (б) двомасової системи за збурюючим впливом

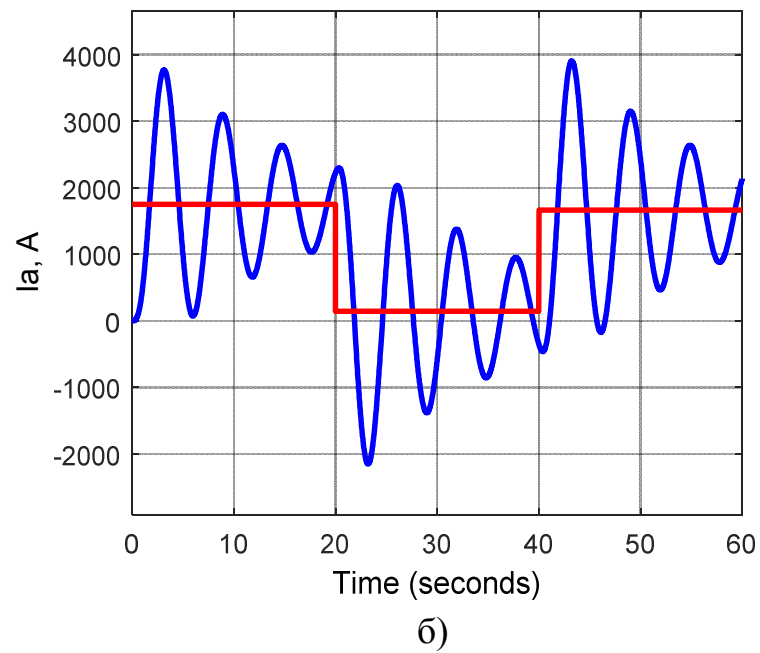
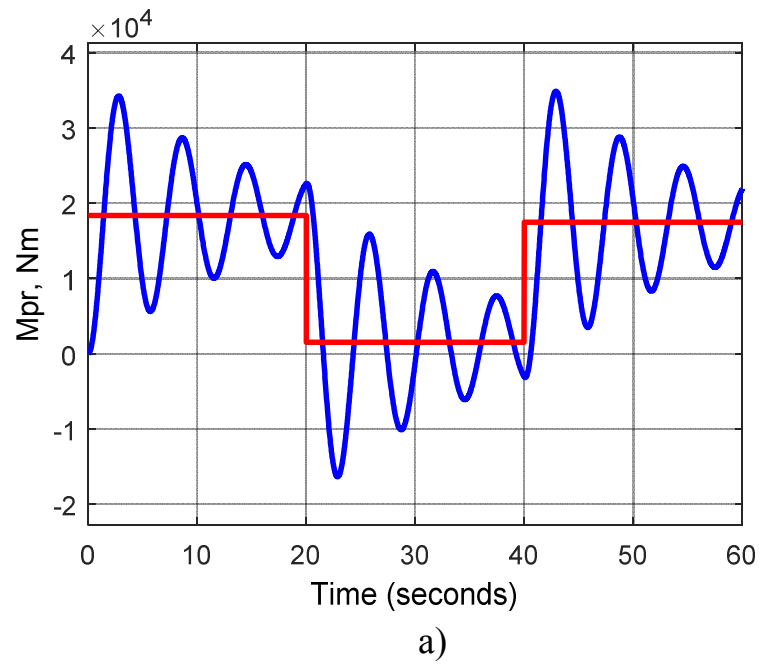


Рисунок 3.9 – Перехідні процеси швидкості двигуна $\omega_d = f(t)$ (а) та струму двигуна $I_a = f(t)$ двомасової системи за збурюючим впливом

ВИСНОВКИ ДО РОЗДІЛУ 3

1. У розділі виконано детальний аналіз роботи приводу механізму повороту роторного колеса з урахуванням пружних характеристик елементів конструкції: системи «ротор – стріла – підвіска», а також опору ґрунту або гірської породи. Встановлено, що за певних комбінацій робочих швидкостей та динамічних збурень можливе виникнення резонансних коливань, які призводять до підвищеного навантаження на металоконструкцію та негативно впливають на надійність і плавність роботи механізму. Показано, що механічну частину приводу з урахуванням пружних зв'язків доцільно розглядати як двомасову систему, що дозволяє адекватніше описати динаміку та коливальні процеси.

2. Побудовано математичну модель двомасового приводу. Отримано рівняння стану системи як у диференціальній формі, так і у вигляді векторно-матричного представлення. Визначено склад вектора стану, тобто перелік змінних, що описують динаміку механічної та електромеханічної частин, а також сформовано вектор зовнішніх впливів. Синтезовано матрицю стану **A** та матрицю керування **B**, які визначають структуру моделі. Розроблено детальну алгоритмічну схему двомасової системи, а також спрощений її варіант із включеним модульним оптимізованим контуром струмового регулювання, що дає змогу дослідити вплив покращеного струмового регулятора на загальну поведінку приводу.

3. За допомогою засобів MATLAB проведено моделювання роботи двомасової системи, використовуючи середовище SIMULINK. На основі отриманої математичної моделі побудовано функціональну схему, яка дозволяє відтворити реакцію системи на різні типи впливів. На вхід моделі подавався ступінчастий сигнал випадкової амплітуди в межах від $+U_{z \max}$ до $-U_{z \max}$, що відповідає зміні швидкості в діапазоні нормальної експлуатації. Аналіз отриманих графіків перехідних процесів показав, що динамічні характеристики системи є недостатньо задовільними: основні координати виявляють суттєву коливальність, що підтверджує чутливість механізму до пружних властивостей та необхідність удосконалення структури системи керування для зменшення коливань та підвищення стійкості приводу.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

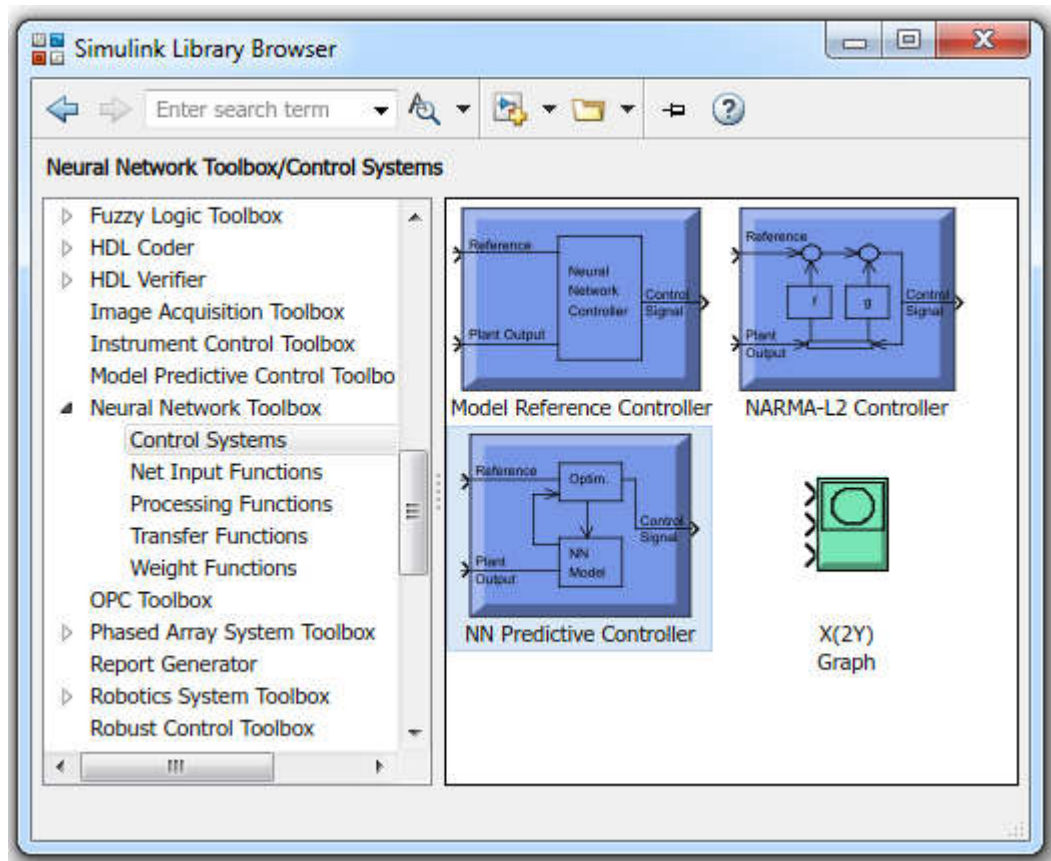


Рисунок 4.1 – Набір блоків Neural Network Blocksets підсистеми Control Systems середовища SIMULINK

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

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

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

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

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

4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller

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

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

4.2.1 Процедура ідентифікації об'єкта керування

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

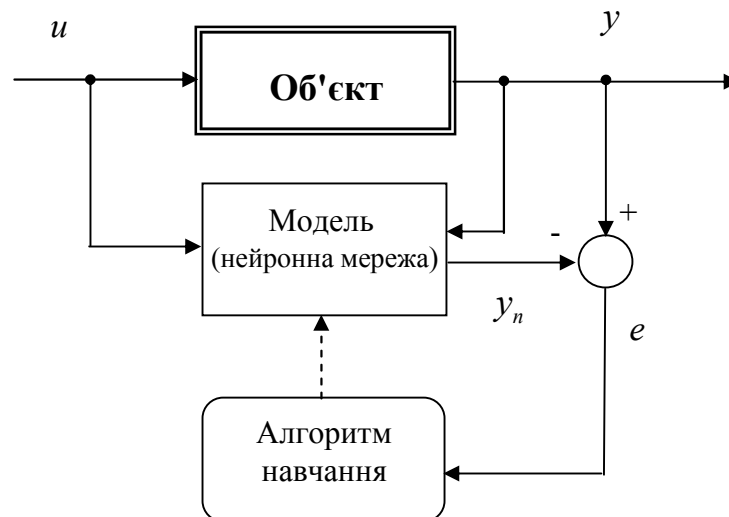


Рисунок 4.2 – Структурна схема підсистеми ідентифікації

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

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

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

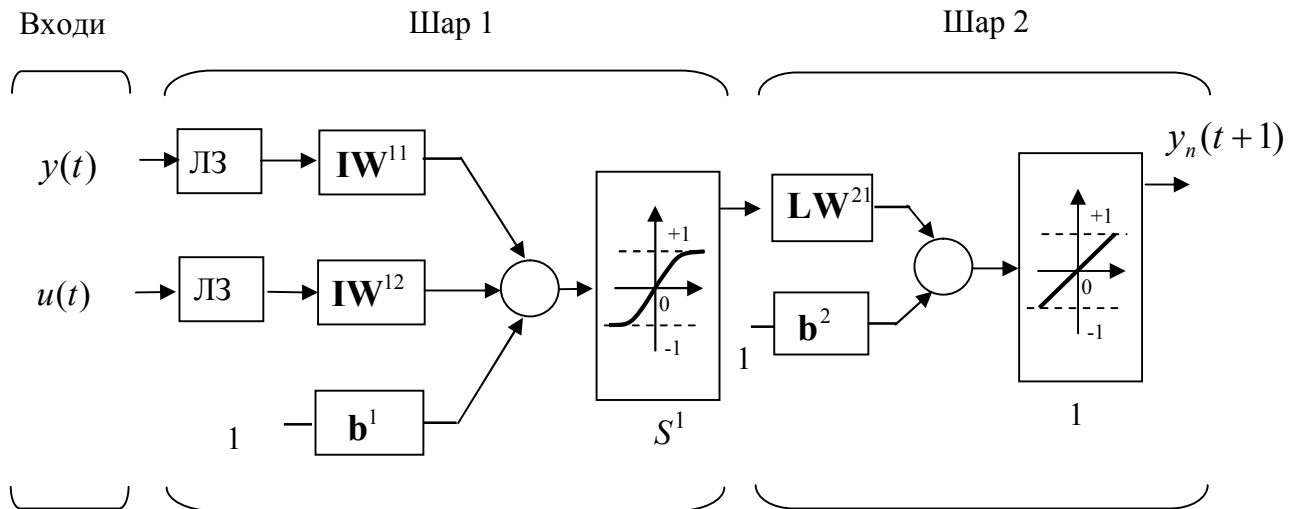


Рисунок 4.3 – Структура нейронної мережі, що моделює об’єкт керування

4.2.2 Принцип функціонування прогнозуючої системи керування

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

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

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

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

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

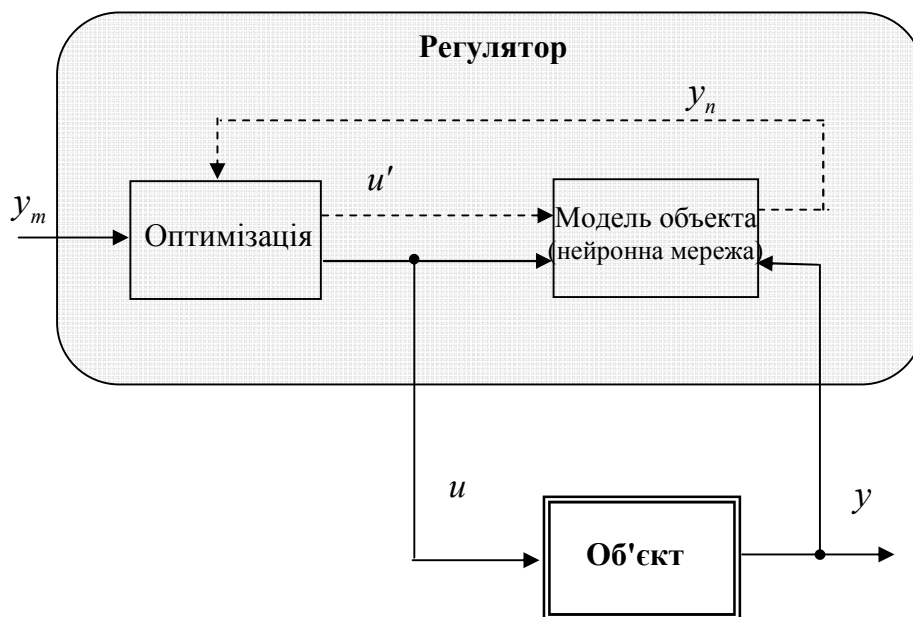


Рисунок 4.4 – Узагальнена схема керування з використанням прогнозуючого регулятора

4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB

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

1. Функції одновимірної оптимізації

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

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

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

2. Функції, що реалізують механізм керування з прогнозуванням

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

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

3. Моделі та графічні додатки Simulink

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

4. Допоміжні програмні модулі

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

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

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

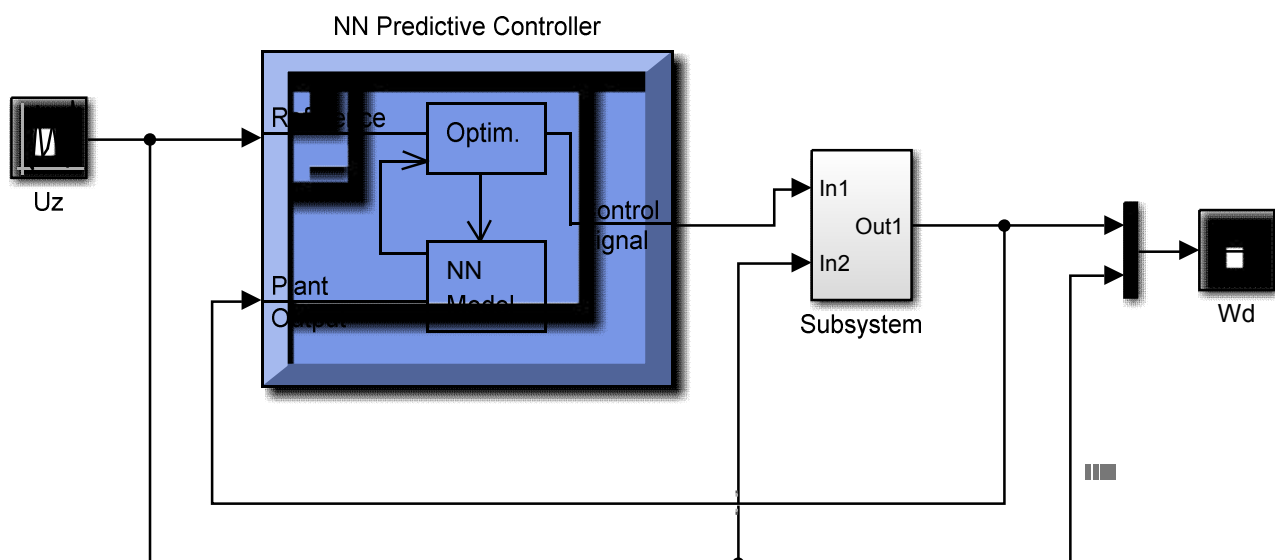


Рисунок 4.5 – Структурна модель системи керування з нейрорегулятором NN Predictive Controller

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

Структура внутрішньої організації нейромережевого регулятора NN

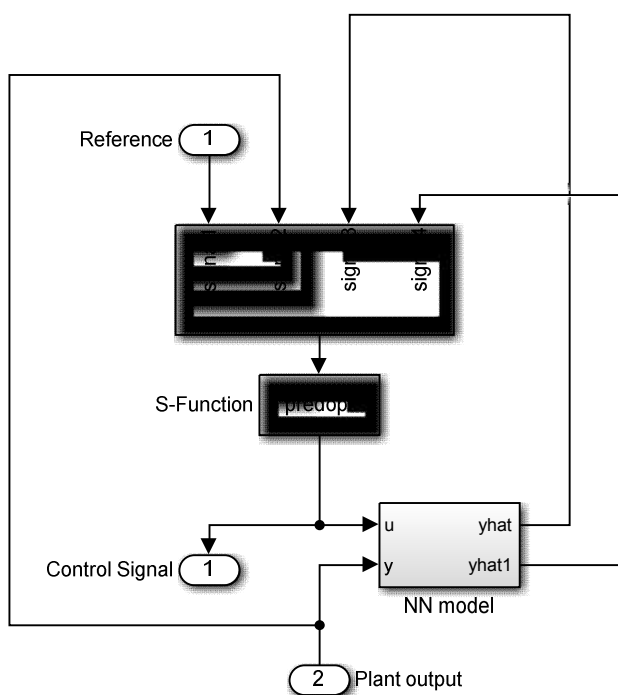


Рисунок 4.6 – Структурна схема нейрорегулятора

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

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

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

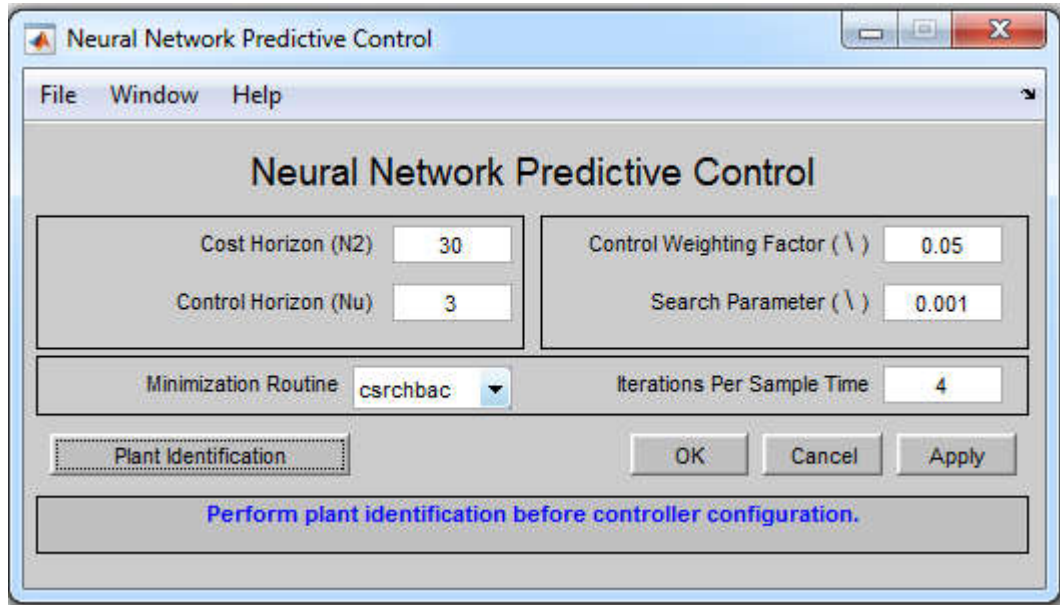


Рисунок 4.7 – Інтерфейс налаштування параметрів регулятора

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

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

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

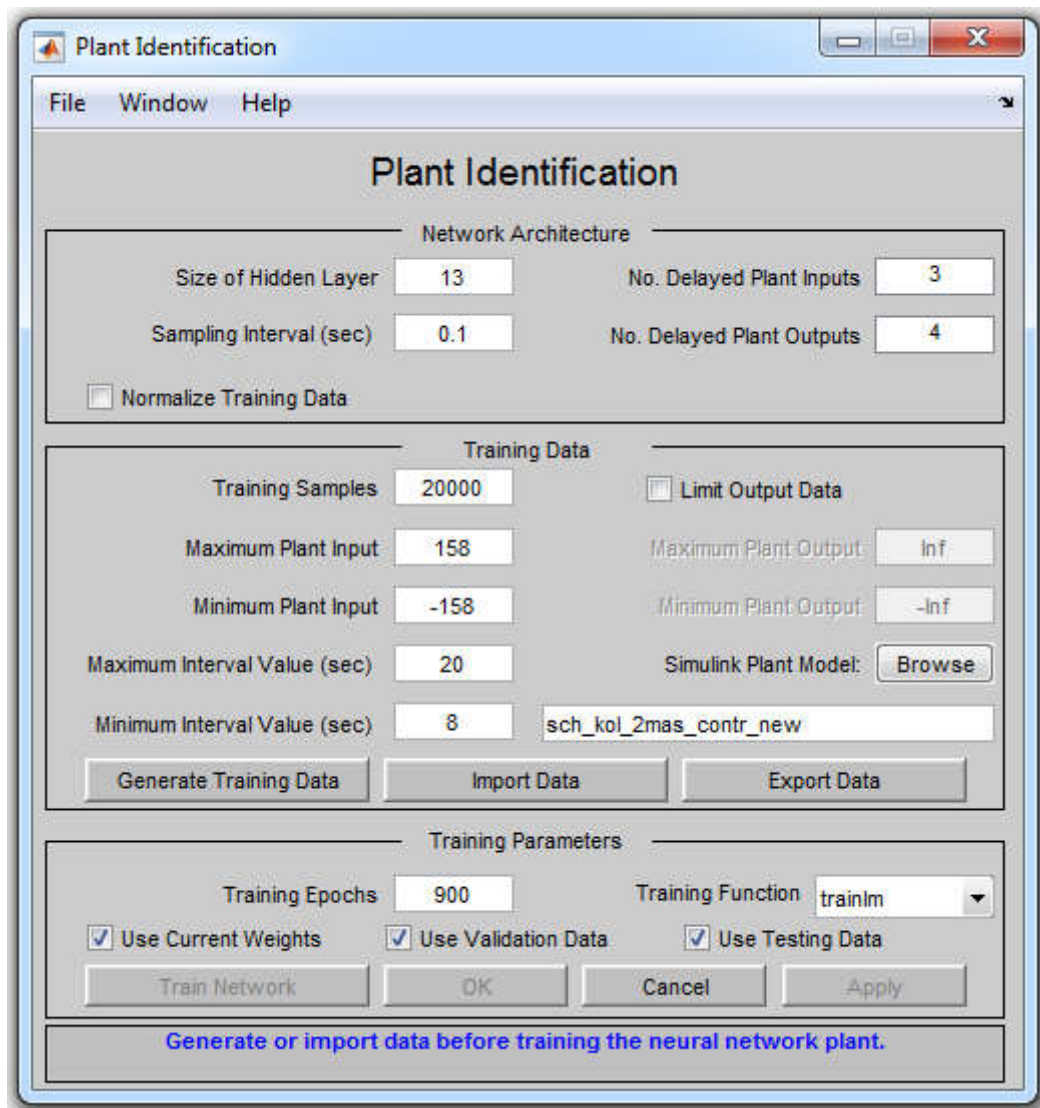


Рисунок 4.8 – Вікно процедури ідентифікації об'єкта керування

Опис вікна Plant Identification

Секція Network Architecture (Архітектура мережі).

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

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

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

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

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

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

Вибір інтервалу дискретизації має суттєвий вплив на:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

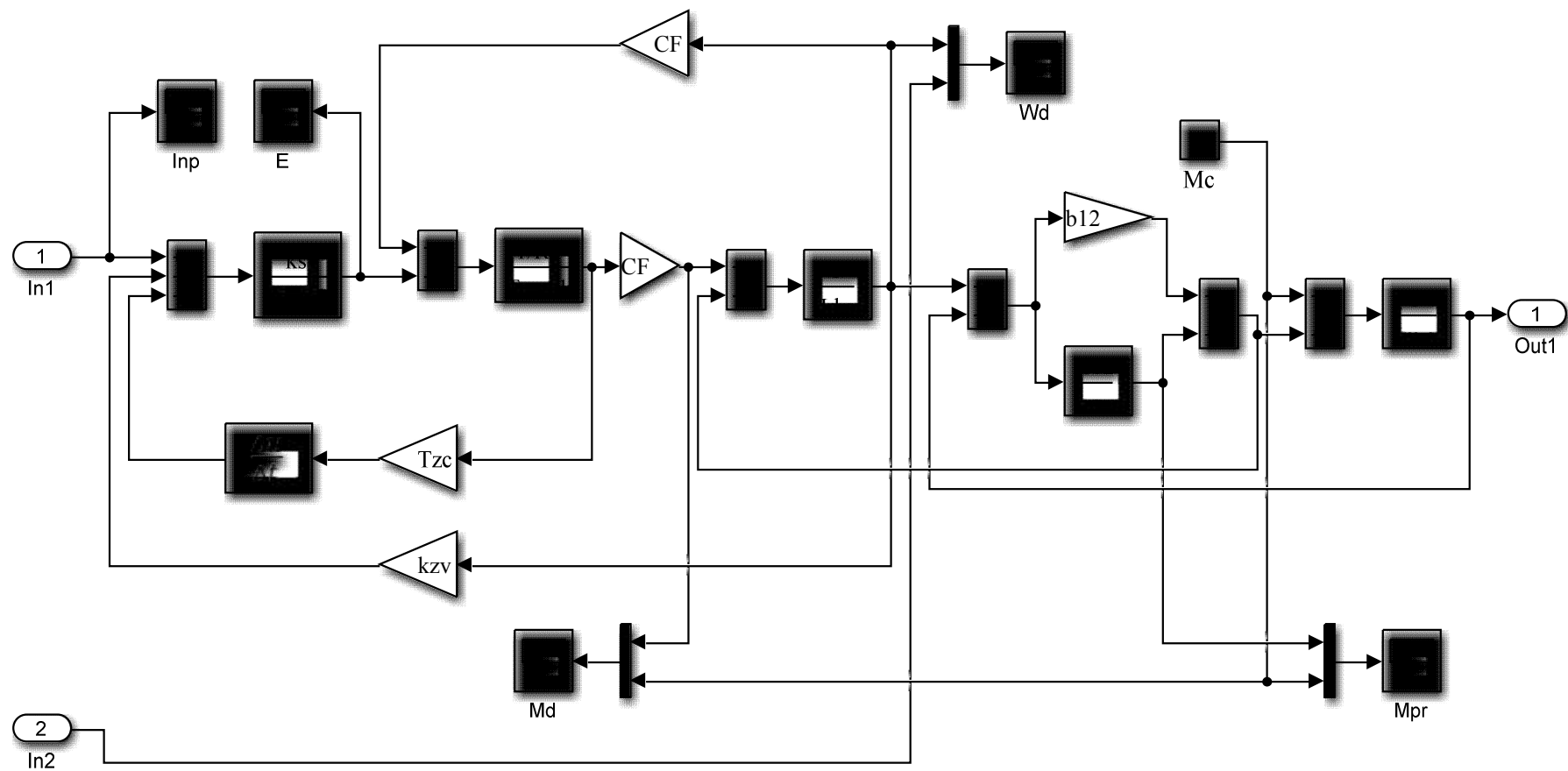


Рисунок 4.9 – Модель керованого об'єкта нейрорегулятора

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

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

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

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

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

Правильне визначення кількості епох є критично важливим: якщо їх замало, 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.

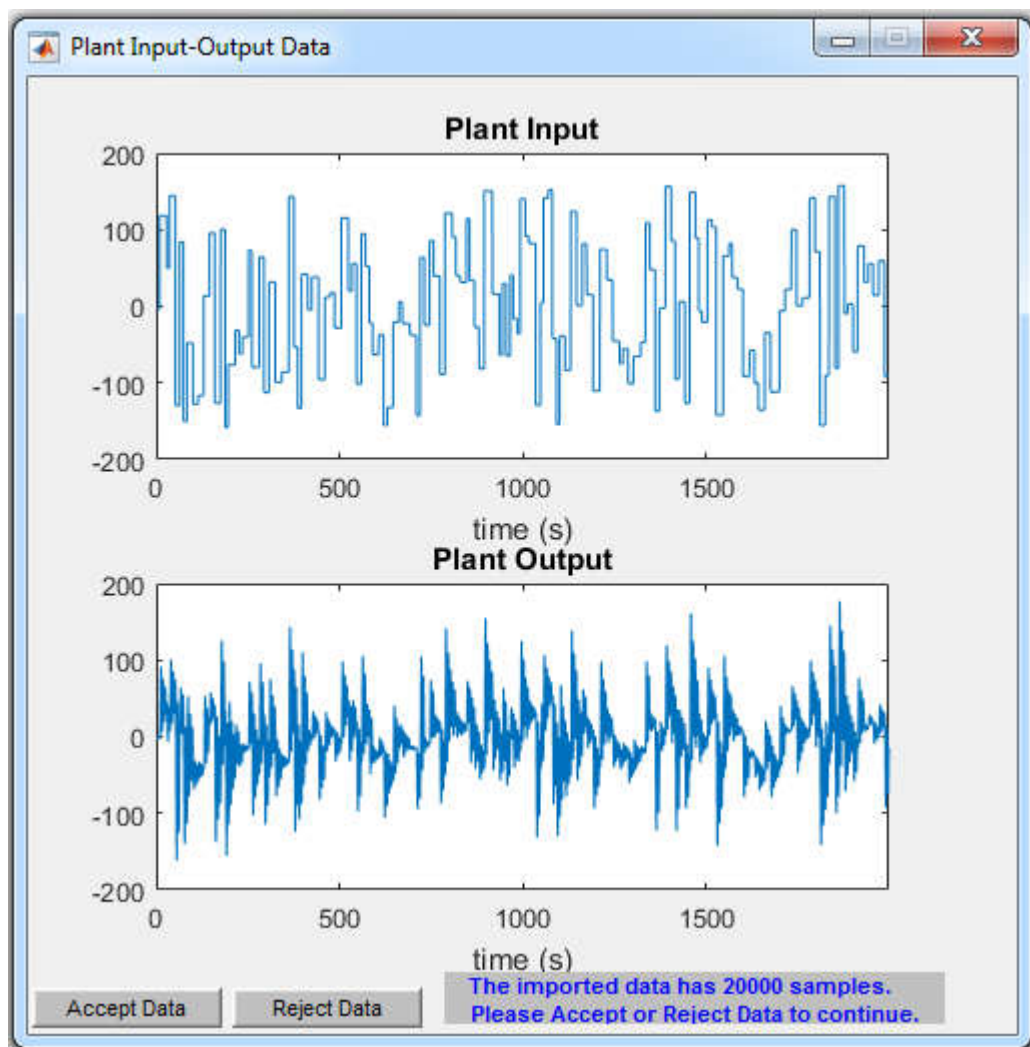


Рисунок 4.10 – Графіки вхідного та вихідного сигналів під час генерації навчальної послідовності

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.

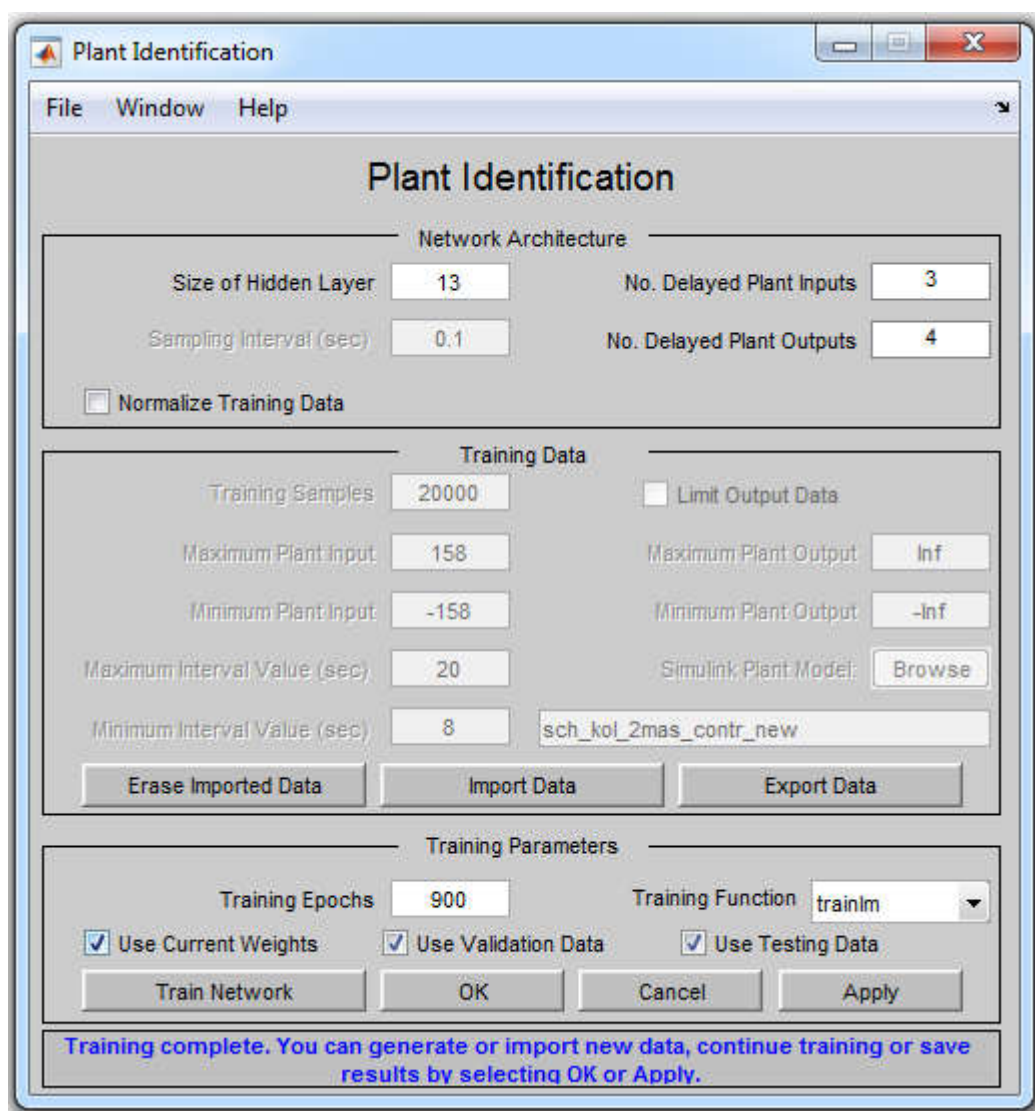


Рисунок 4.11 – Вікно Plant Identification після генерації навчальної послідовності

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).

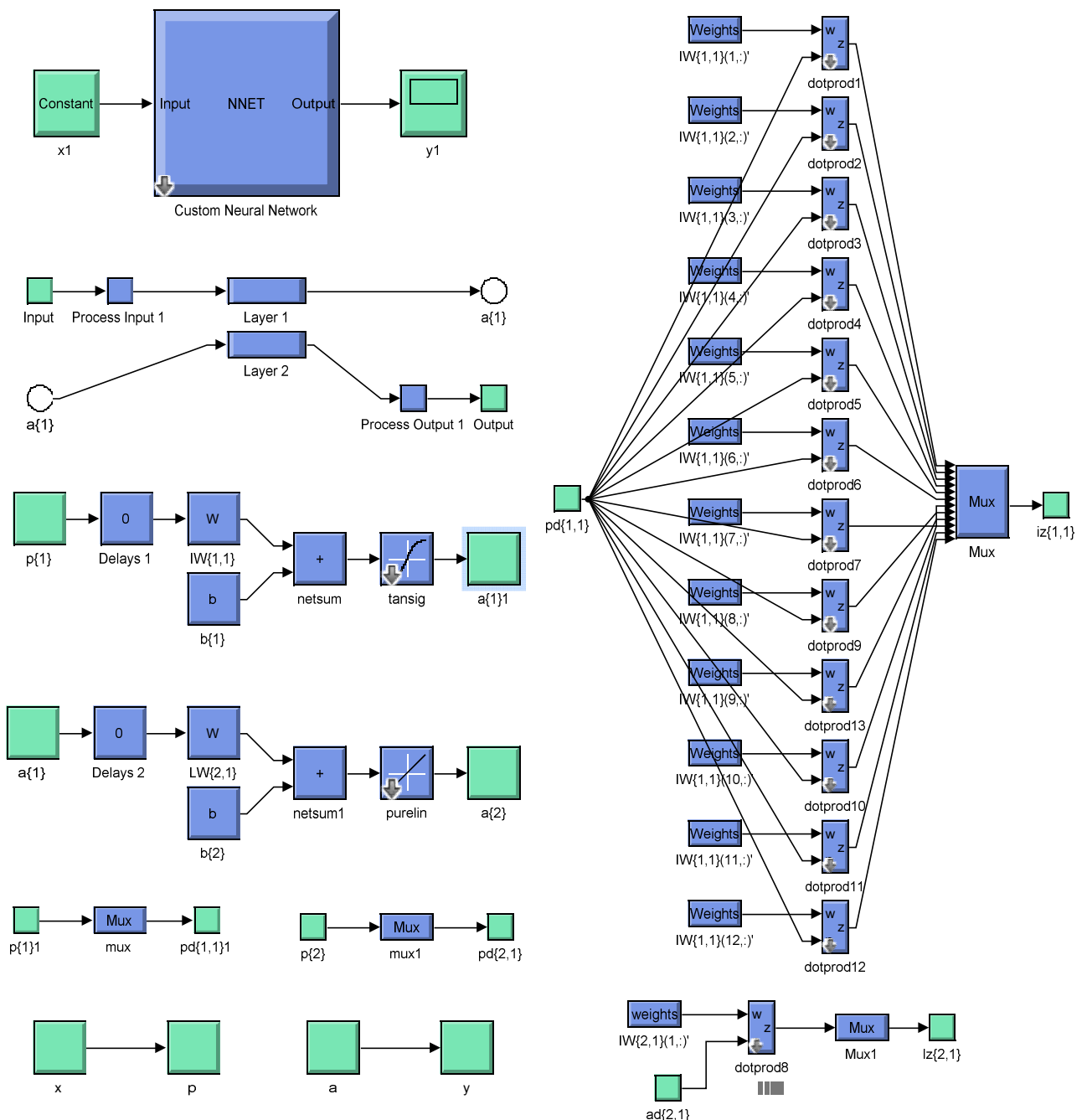


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

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.

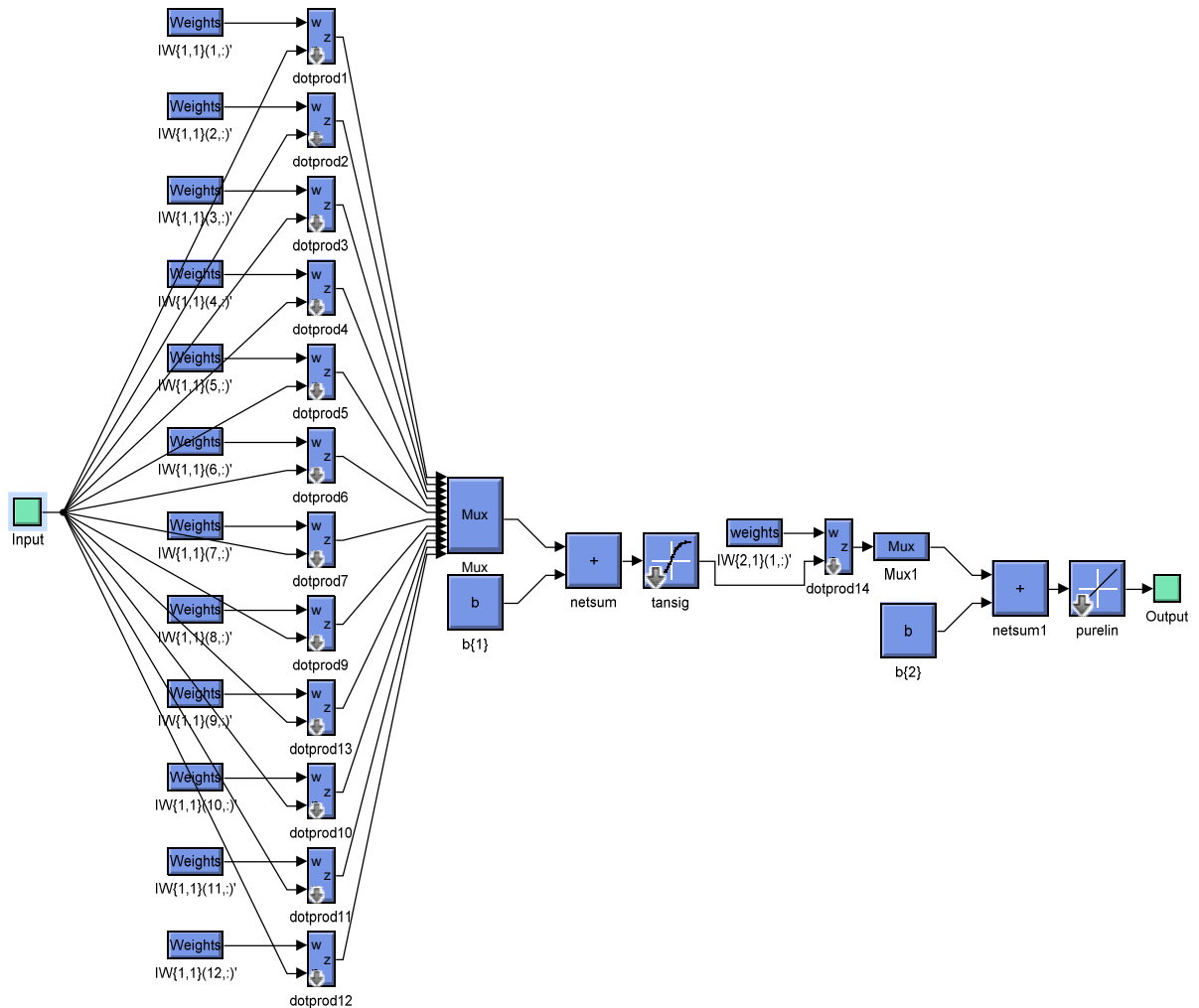


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

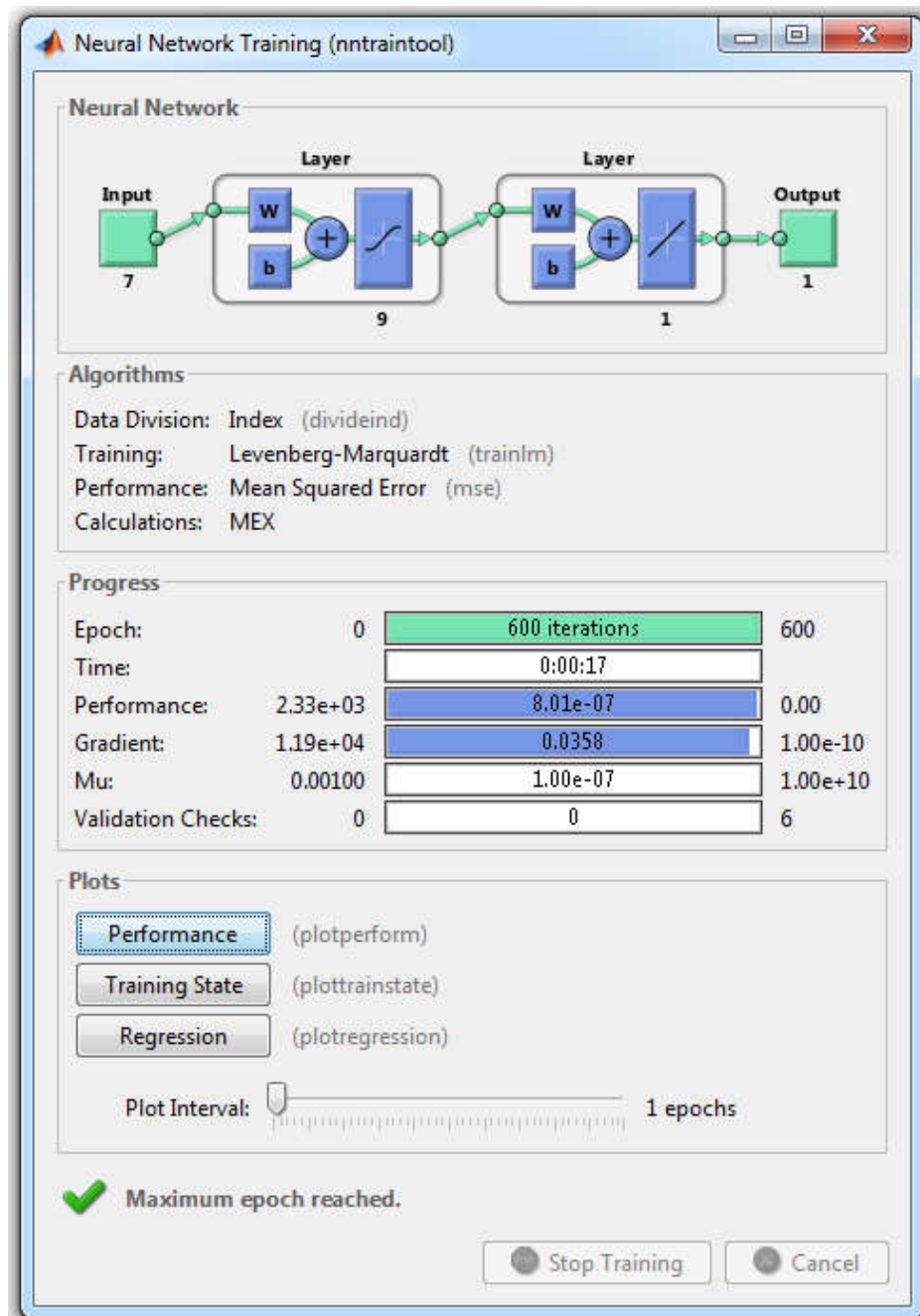


Рисунок 4.14 - Головне вікно для відображення процесу та результатів навчання нейронної мережі

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.

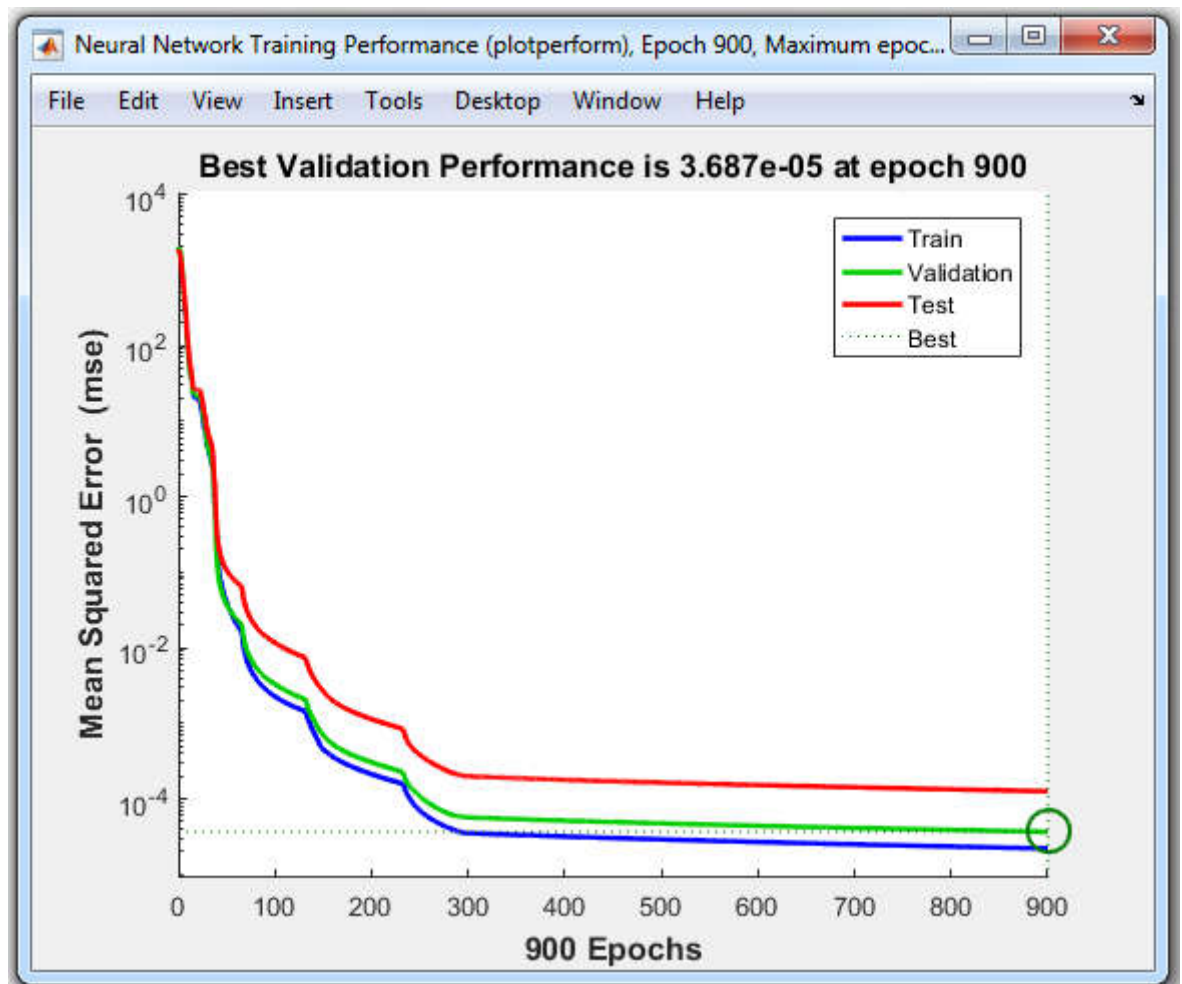


Рисунок 4.15 – Вікно контролю процесу навчання

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.

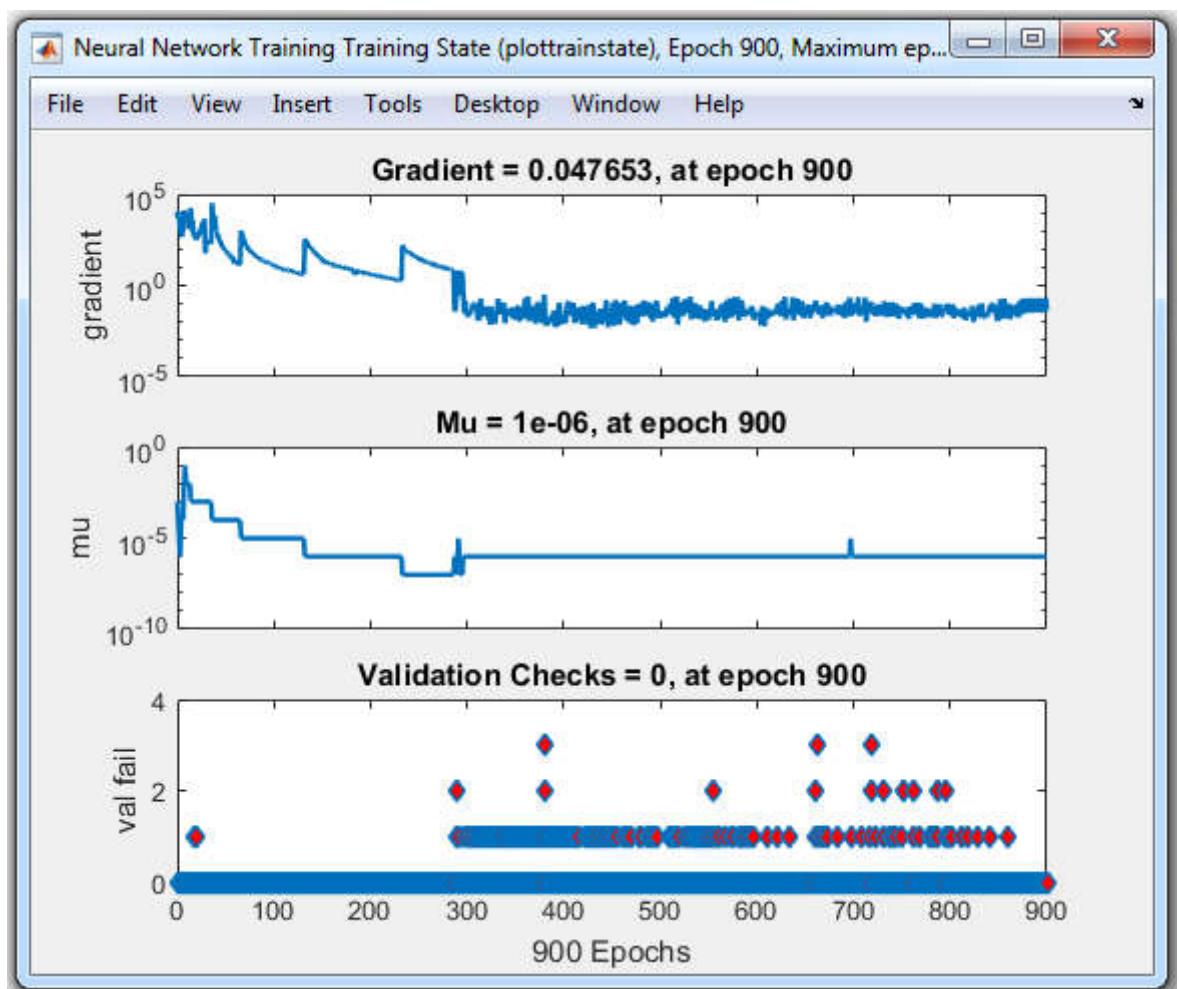


Рисунок 4.16 - Вікно Neural Network Training State

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.

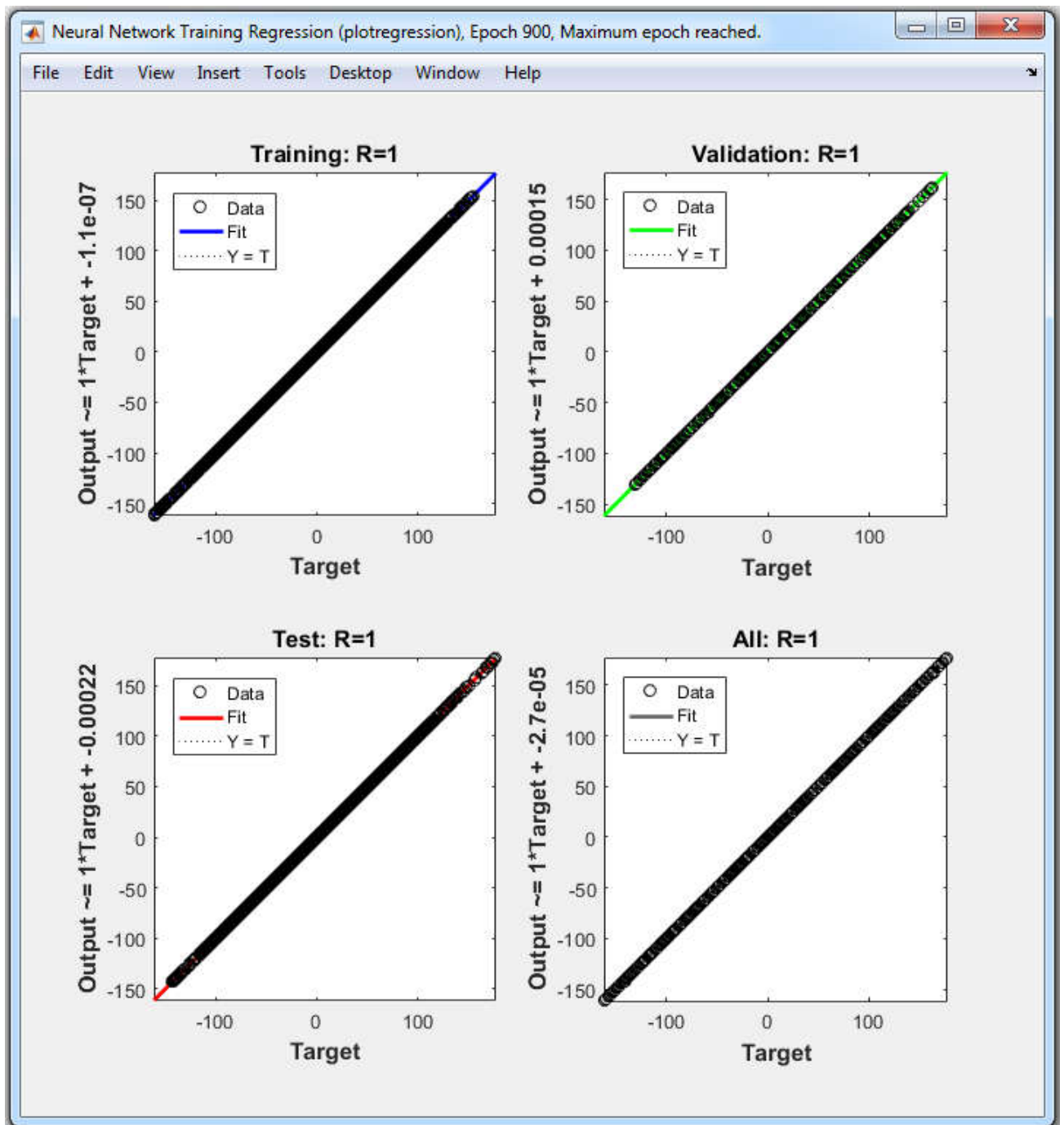


Рисунок 4.17 - Вікно Neural Network Training Regression

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 передбачати дані;

- $R = -1$ – повна зворотна лінійна залежність, модель передбачає значення у протилежному напрямку.

Коефіцієнт кореляції, близький до 1, свідчить про високу точність регресійної моделі, тоді як значення близько до 0 або від’ємне сигналізує про низьку точність прогнозування. Використання R разом із графіками Data та Fit дозволяє комплексно оцінити ефективність моделі нейронної мережі на всіх етапах навчання та перевірки, а також визначити можливі напрями для її вдосконалення.

Продовження опису вікна Neural Network Training (nntraintool)

Plot Interval (інтервал графіка). Параметр Plot Interval визначає, як часто під час навчання нейронної мережі для задач регресії оновлюються графіки та відображаються проміжні результати. Іншими словами, він встановлює кількість ітерацій або епох, після яких відбувається оновлення інформації на графіках.

Правильне налаштування Plot Interval дозволяє ефективно контролювати обсяг відображуваної інформації під час навчання, особливо коли кількість епох велика. Менше значення Plot Interval забезпечує частіше оновлення графіків, що дає змогу уважніше аналізувати процес навчання, але одночасно збільшує обсяг візуальної інформації. Більше значення, навпаки, зменшує частоту оновлення, спрощуючи відстеження загальної тенденції навчання без перевантаження графіків.

Налаштування цього параметра є важливою частиною контролю процесу навчання, оскільки дозволяє оцінювати вплив змін параметрів мережі на результати моделі в режимі реального часу.

Кнопки Stop Training і Cancel. У вікні **Neural Network Training (nntraintool)** середовища MATLAB присутні дві кнопки для управління процесом навчання, які мають різні функції:

- **Stop Training (зупинка навчання)** – негайно припиняє процес навчання нейронної мережі без можливості його продовження. Ця кнопка викори-

стовується, коли навчання виявляється занадто тривалим, не дає задовільних результатів або потрібно терміново припинити тренування.

- **Cancel (скасувати)** – дозволяє зупинити навчання, але при цьому зазвичай передбачає підтвердження дії та збереження поточного стану мережі. Використання кнопки Cancel дає змогу зупинити навчання після перегляду проміжних результатів, внести необхідні зміни до параметрів та при необхідності продовжити навчання пізніше.

Після завершення навчання результати відображаються у вікні **Training Data for NN Predictive Control**, на якому показані графіки фактичних та прогнозованих значень, що дозволяє оцінити точність моделі та якість її навчання (рис. 4.18).

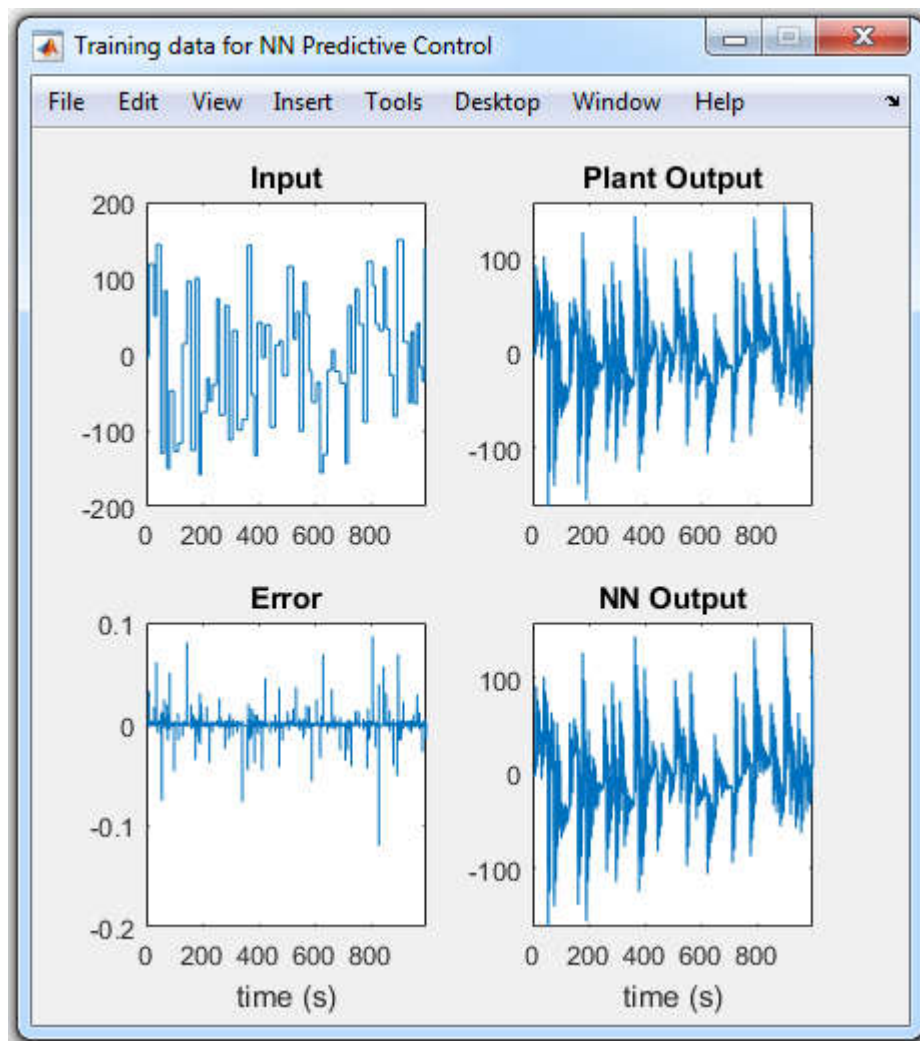


Рисунок 4.18 – Графічне представлення результатів навчання нейронної мережі

Onuc вікна Training Data for NN Predictive Control

У вікні **Training Data for NN Predictive Controller** відображаються графіки, що ілюструють процес навчання нейронної мережі та взаємодію моделі з системою:

- **Input (вхідні сигнали)** – графік Input показує вхідні сигнали, які подаються на систему та нейронну мережу під час навчання. Ці сигнали стимулюють модель для генерації відповідей і дозволяють досліджувати її реакцію на різні умови.
- **Plant Output (вихід системи)** – графік Plant Output демонструє фактичні вихідні дані системи (Plant) під час навчання. Він відображає реальну динаміку системи у відповідь на вхідні сигнали, що дозволяє порівняти її поведінку з прогнозами моделі.
- **NN Output (вихід нейромережі)** – графік NN Output відображає прогнозовані вихідні сигнали, згенеровані нейронною мережею під час навчання. Ці дані ілюструють здатність моделі відтворювати поведінку системи на основі поданих вхідних сигналів.
- **Error (помилка)** – графік Error показує різницю між прогнозованими значеннями нейронної мережі (NN Output) та фактичними виходами системи (Plant Output). Цей показник відображає точність моделі та дозволяє оцінити, наскільки адекватно мережа відтворює динаміку системи.

Аналогічні графіки відображаються у вікнах **Validation Data for NN Predictive Control** та **Testing Data for NN Predictive Control** (рис. 4.19, 4.20), які ілюструють результати перевірки моделі на контрольній та тестовій множині даних. Ці графіки дозволяють оцінити здатність нейронної мережі узагальнювати знання, отримані під час навчання, на нові дані.

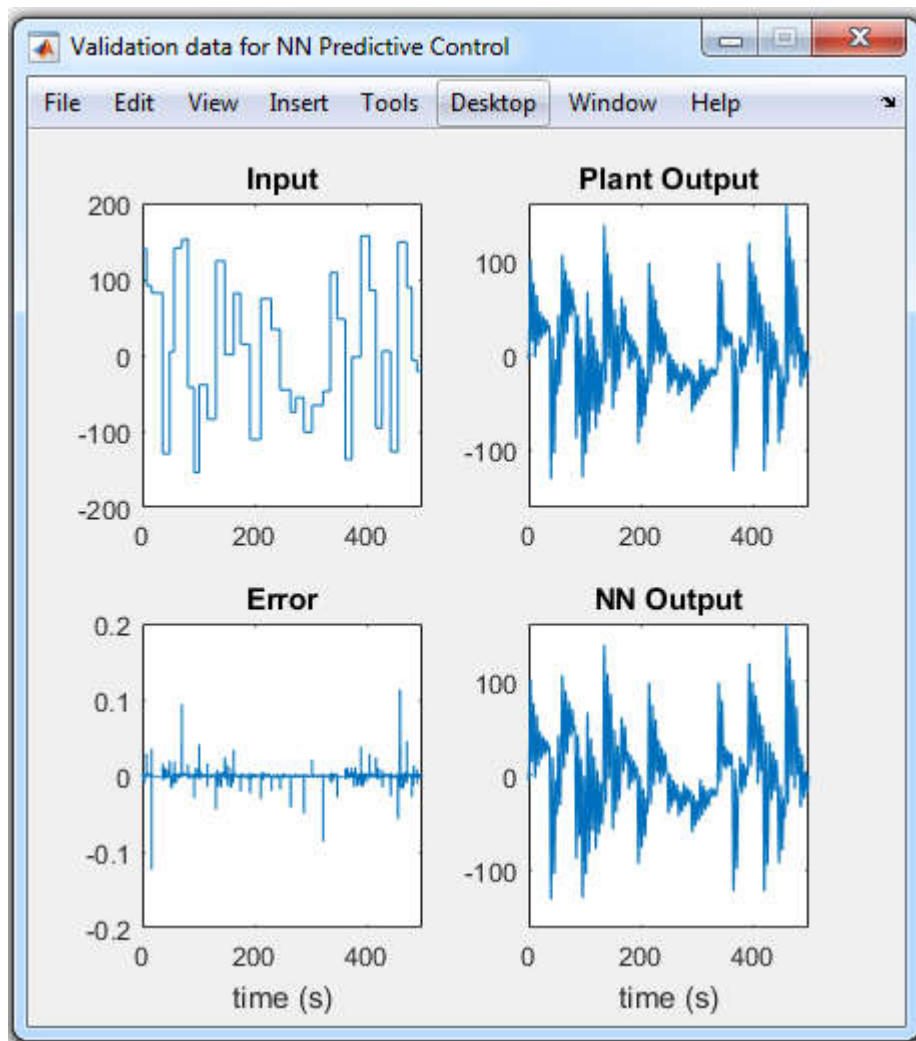


Рисунок 4.19 – Візуалізація результатів перевірки нейронної мережі на контрольній множині

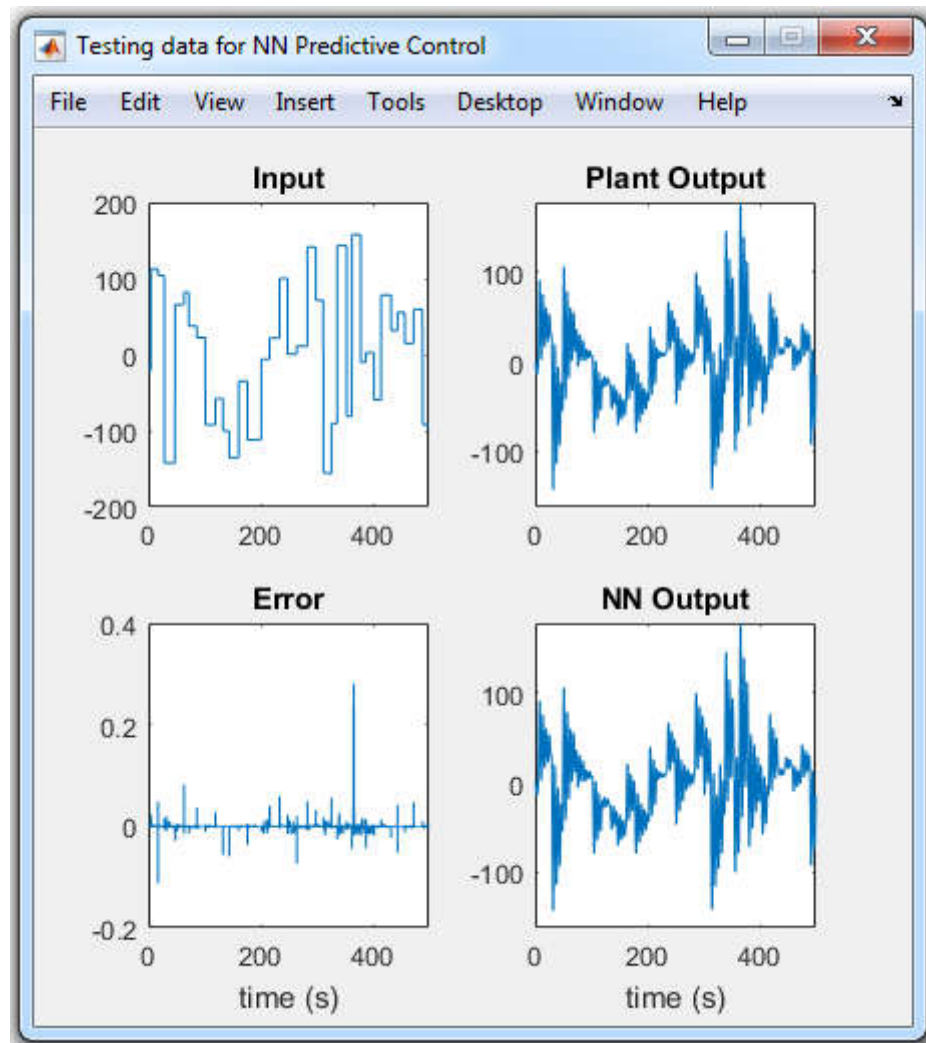


Рисунок 4.20 – Візуалізація результатів перевірки нейронної мережі на тестовій множині

Продовження опису вікна *Plant Identification*.

Після завершення процесу навчання у вікні **Plant Identification** з'являється повідомлення: *"Training complete. You can generate or import new data, continue training or save results by selecting OK or Apply"* (Навчання завершено. Можна згенерувати або імпортувати нові дані, продовжити навчання або зберегти результати, обравши кнопки **OK** або **Apply**).

На цьому етапі М-функція **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.

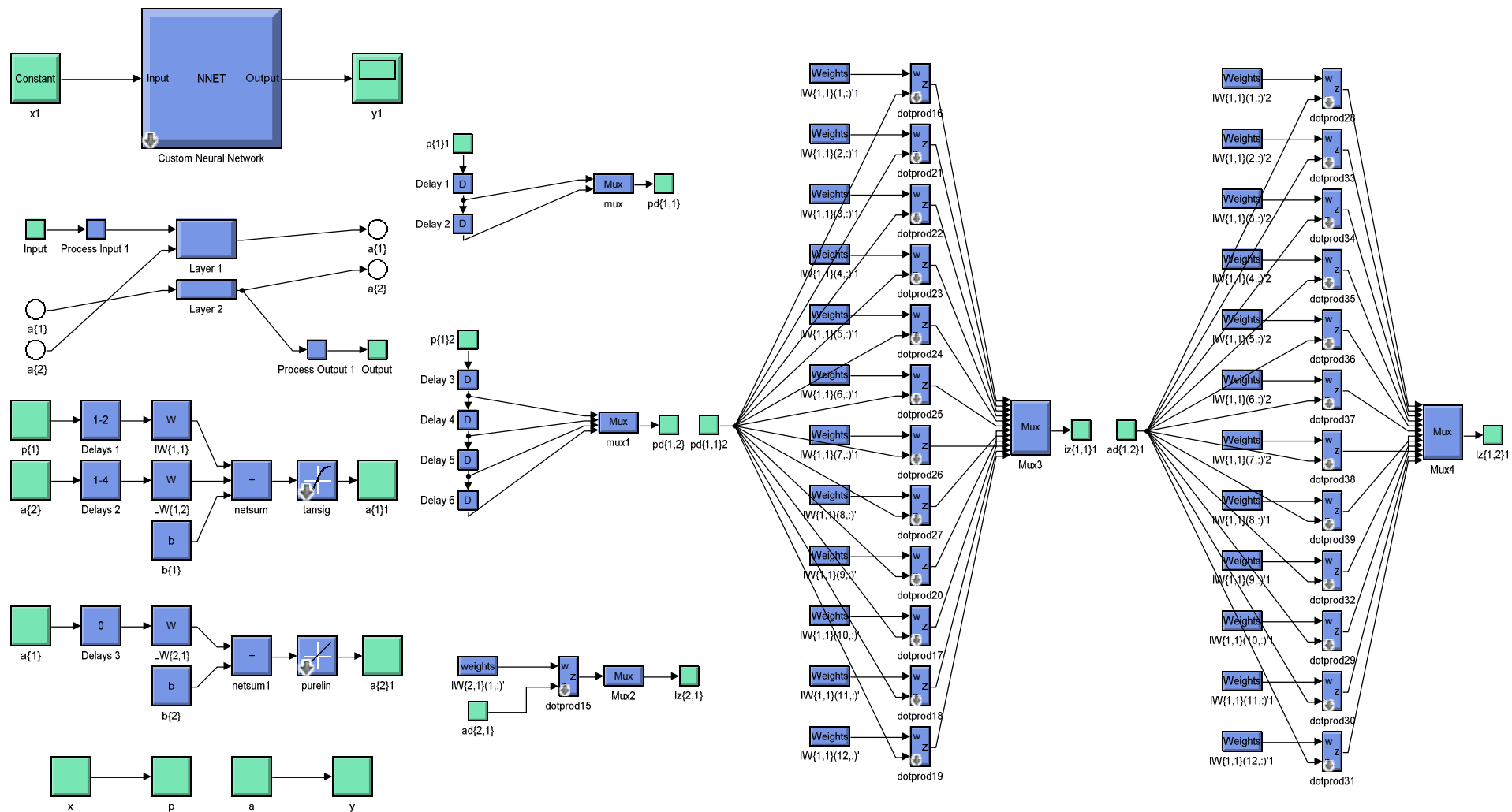


Рисунок 4.21 – Моделі елементів динамічної мережі з блоками затримок у Simulink

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.

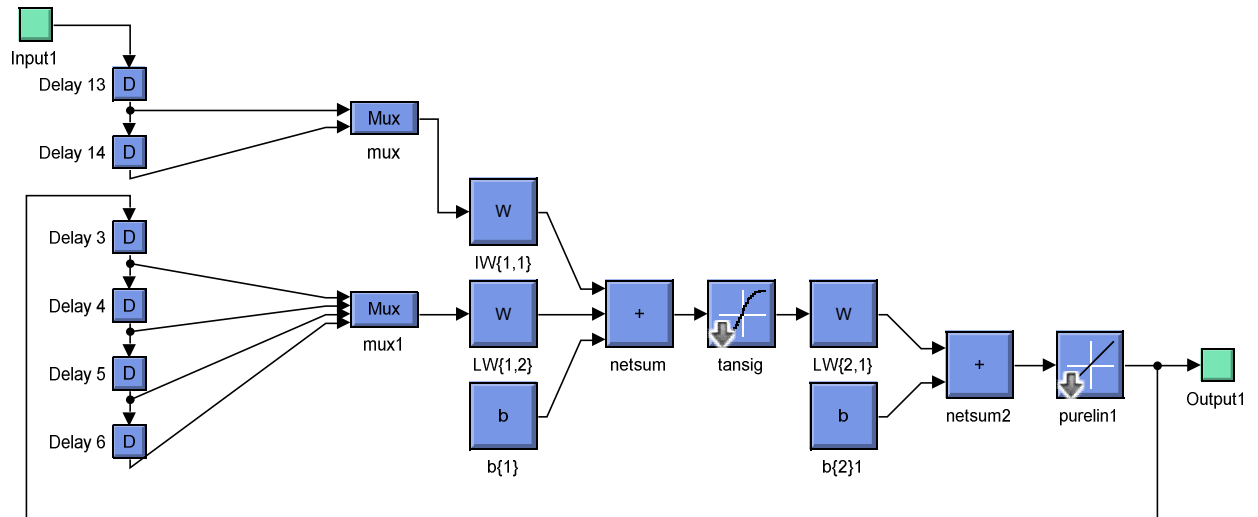


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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.

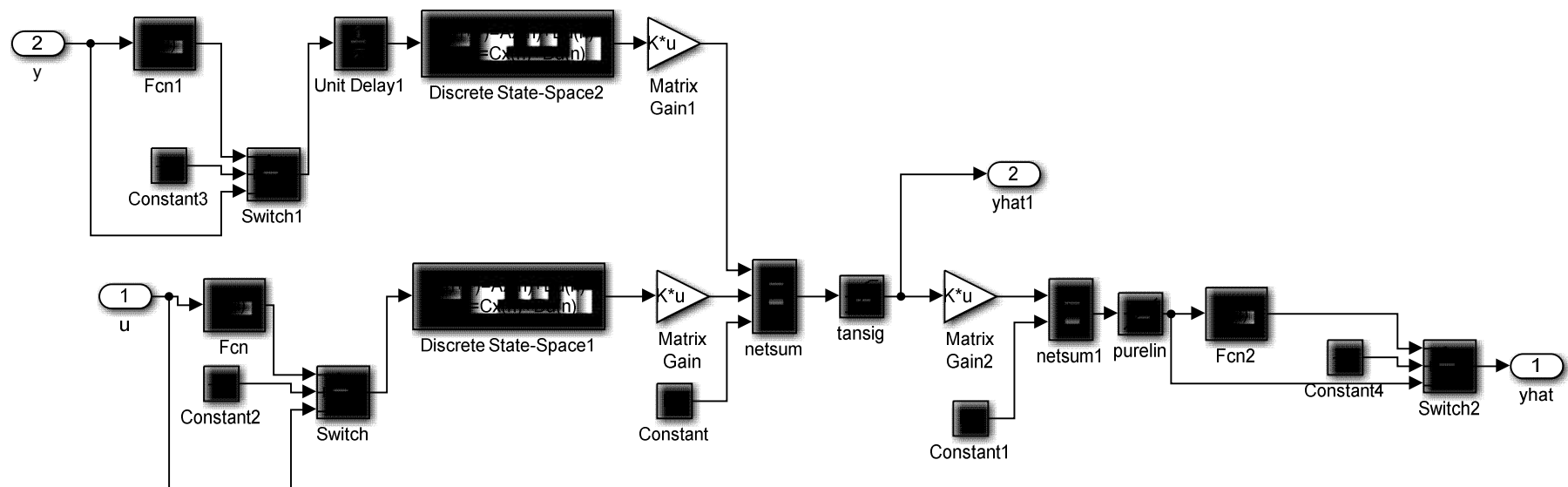


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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.

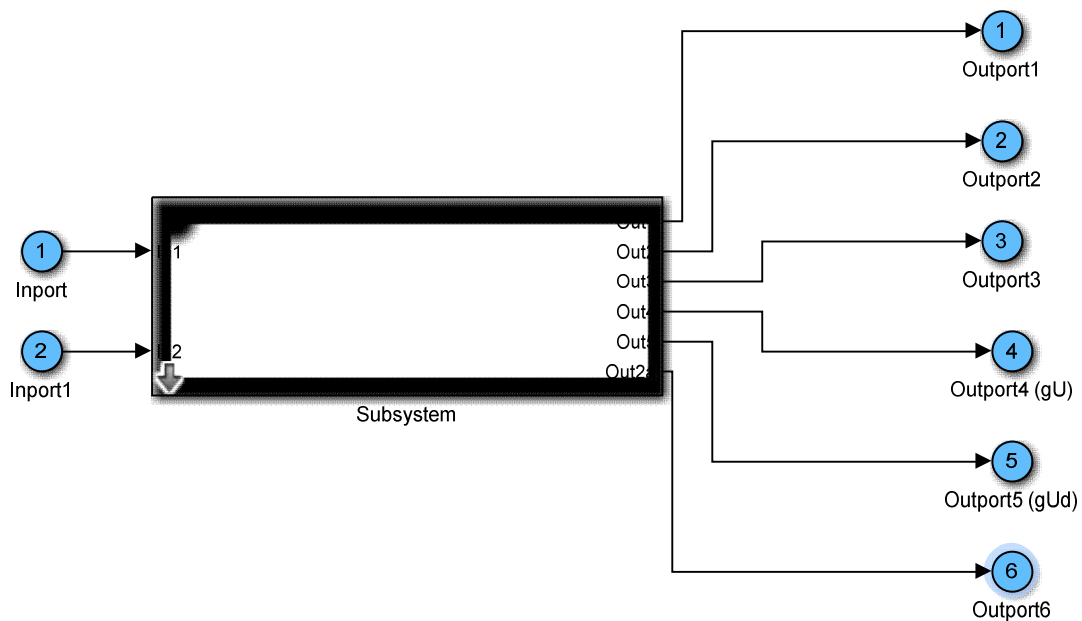


Рисунок 4.24 – Схема ptest3sim2, що використовується для прогнозування процесу функцією predopt

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.

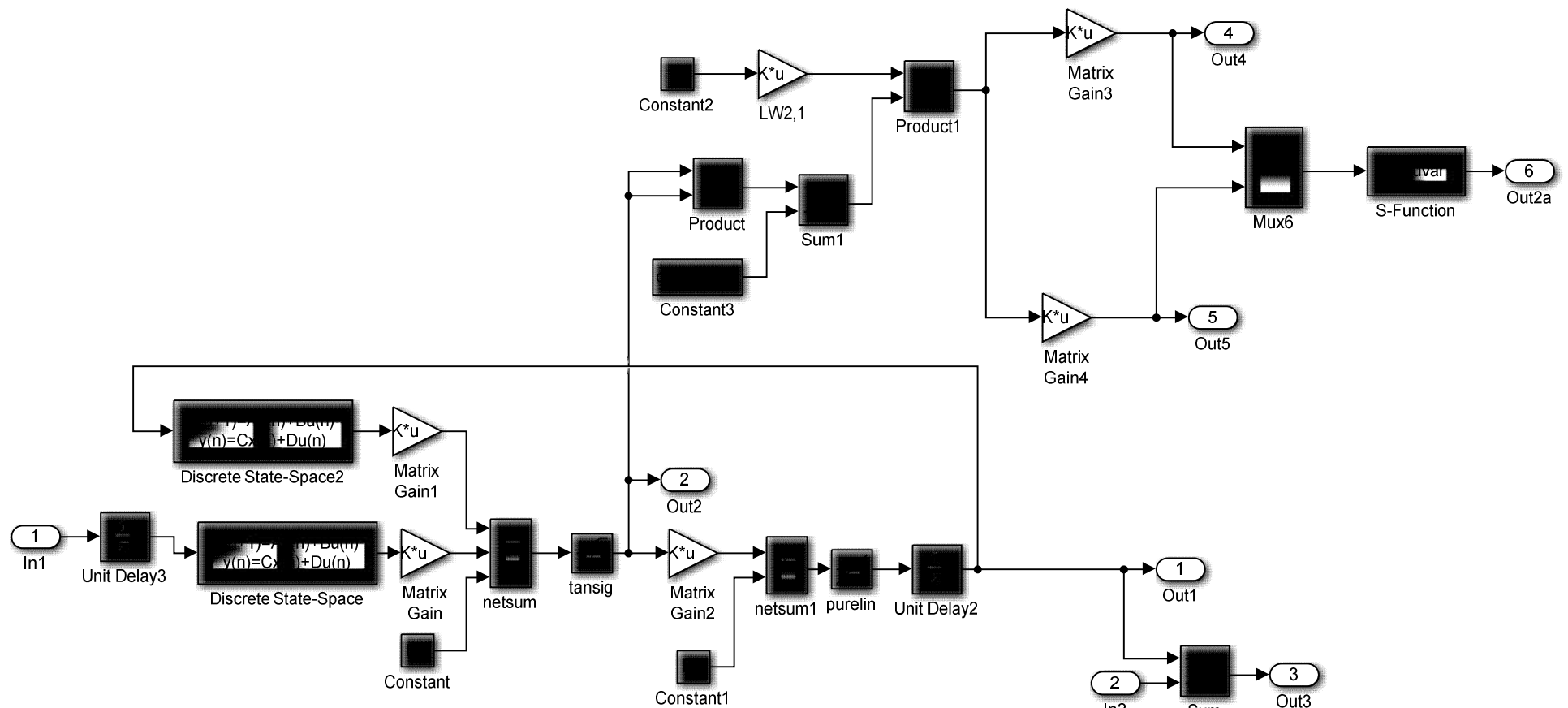


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

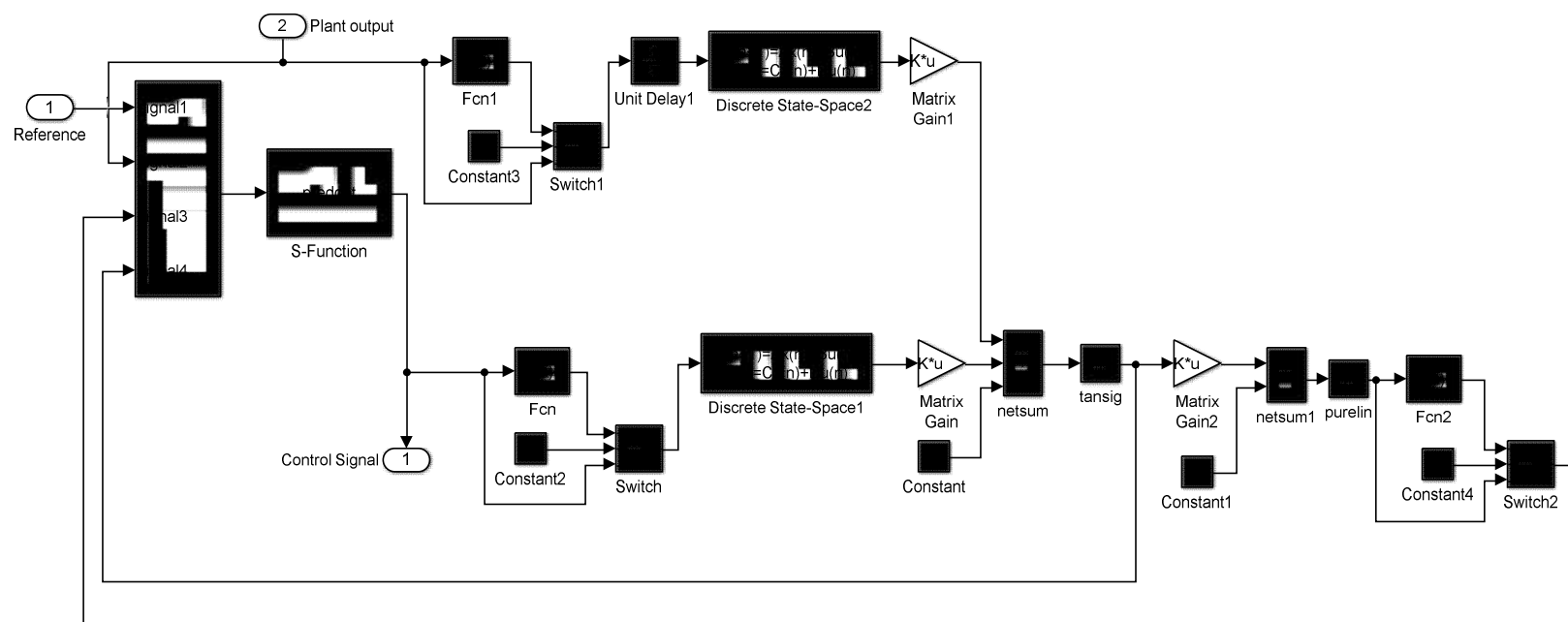


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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 одновимірної пошуку, який задає поріг допустимого зменшення показника якості. Він використовується для контролю процесу оптимізації та визначає момент завершення пошуку, коли подальше покращення якості стає незначним.

Одновимірний пошук – це метод оптимізації, при якому оптимізується лише один параметр, тоді як інші залишаються фіксованими. У контексті нейромережевого прогнозного контролера цей метод допомагає налаштувати окремі параметри контрольної стратегії, що безпосередньо впливають на ефективність керування.

Поріг зменшення показника якості визначає мінімальне значення, яке повинен мати показник покращення, щоб вважати процес оптимізації завершеним. Коли приріст або зменшення показника якості стає меншим за цей поріг, алгоритм припиняє пошук, а отримані параметри вважаються оптимальними. Такий підхід дозволяє заощадити обчислювальні ресурси та прискорити збіжність оптимізації.

Minimization Routine (Процедура мінімізації). Minimization Routine визначає алгоритм оптимізації, який використовується для налаштування параметрів контрольної стратегії. Вибір конкретного методу оптимізації має значний вплив на точність і ефективність роботи контролера.

При виборі процедури враховуються особливості завдання керування, наявні обмеження, а також доступні обчислювальні ресурси. Ретельний підбір алгоритму дозволяє досягти оптимального балансу між точністю розрахунку керуючих сигналів і швидкістю обчислень.

Iterations Per Sample Time (Ітерації на крок часу). Параметр Iterations Per Sample Time визначає кількість ітерацій оптимізації, які виконуються протягом одного кроку часу системи (Sample Time). Він впливає на точність розрахунку оптимального керування та швидкість реакції контролера на зміни стану системи.

Збільшення кількості ітерацій дозволяє отримати більш точний розрахунок керуючих дій, проте підвищує обчислювальне навантаження. Навпаки, занадто мала кількість ітерацій може призвести до недостатньої точності оптимізації та втрати ефективності керування.

Оптимальне значення цього параметра зазвичай підбирається експериментально, виходячи з характеристик конкретної системи керування та обмежень

обчислювальних ресурсів. Воно забезпечує баланс між точністю оптимізації та швидкістю реакції контролера.

Після налаштування всіх параметрів оптимізації вони передаються в блок **NN Predictive Controller** у середовищі Simulink, де використовуються для розрахунку оптимальних керуючих сигналів під час роботи системи.

4.4 Вибір параметрів нейрорегулятора NN Predictive Controller

При 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: $N_b = 10000$, $\Delta t = 0,05 \div 0,1$ 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 $300 \div 600$.

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.

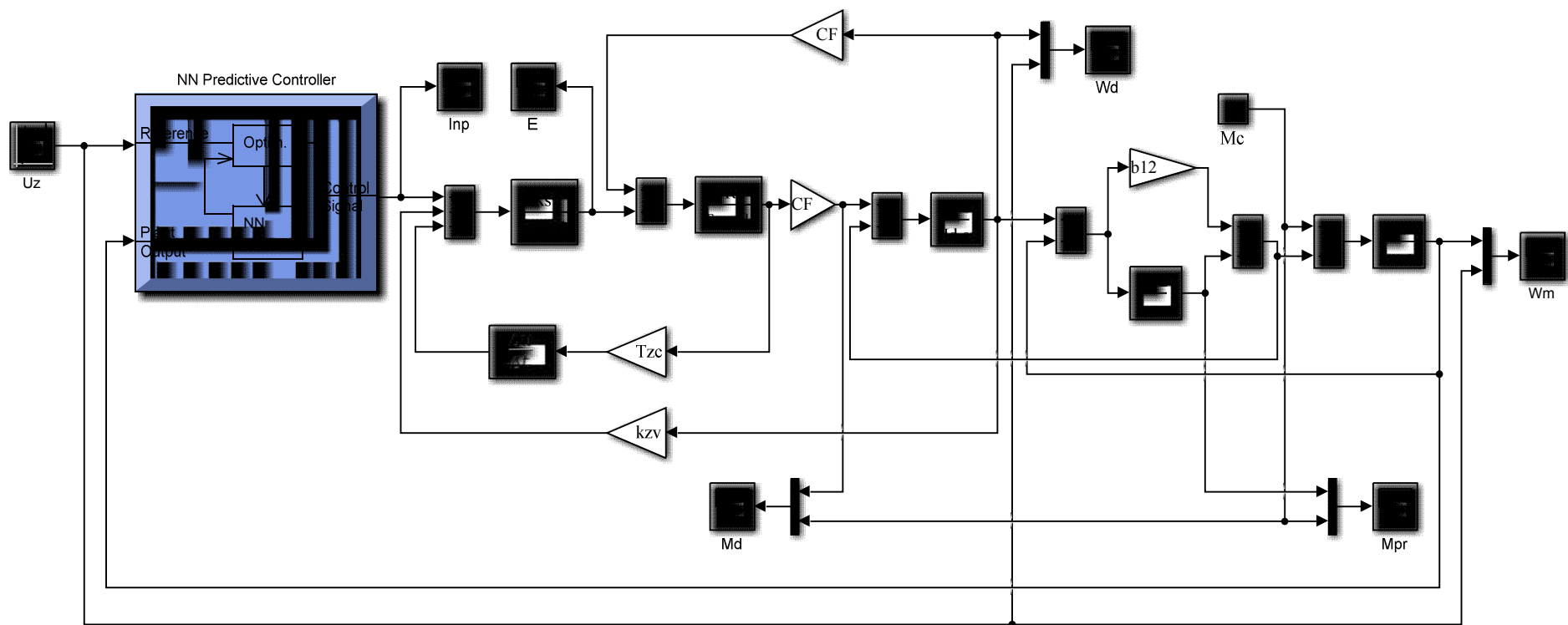


Рисунок 4.27 – Блок-схема моделі двомасової системи із застосуванням нейрорегулятора

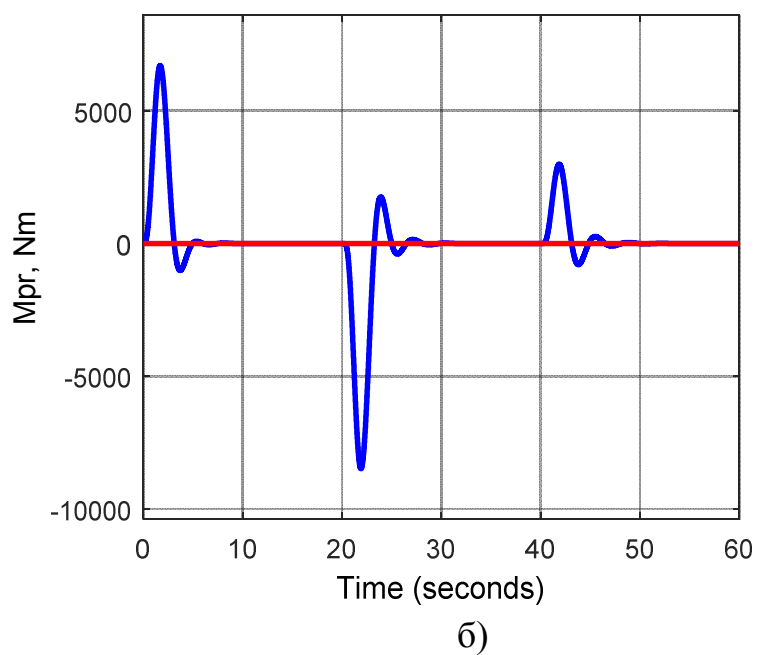
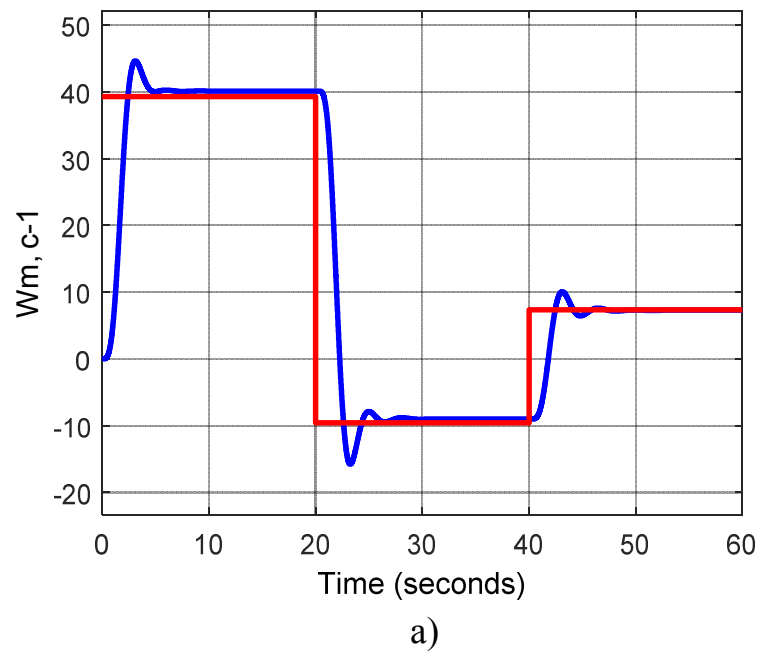
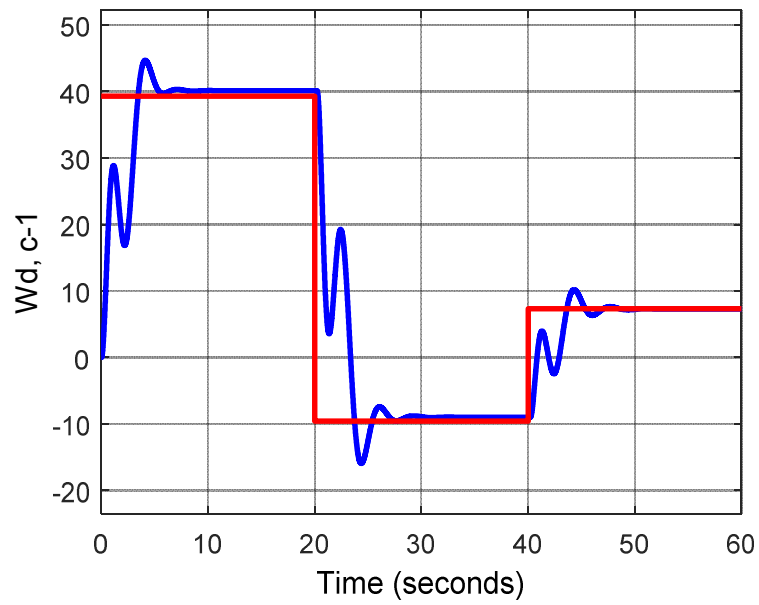
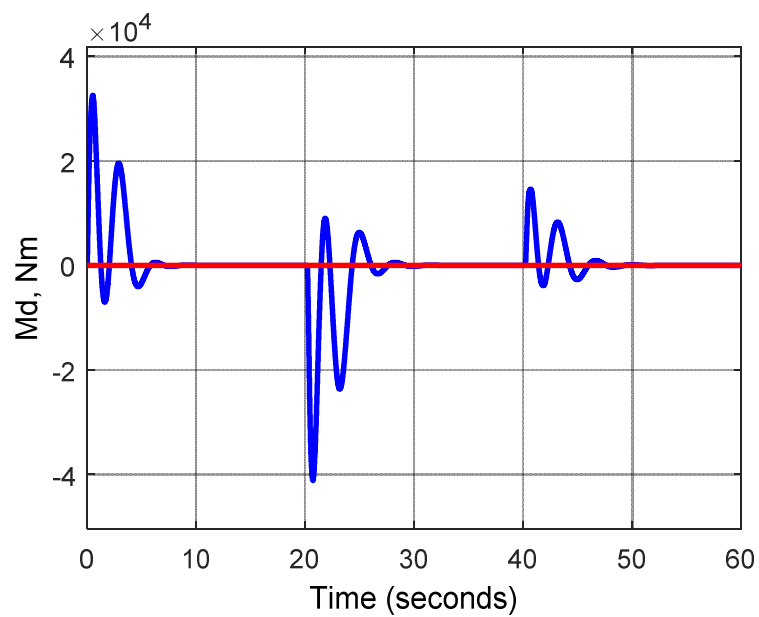


Рисунок 4.28 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією задаючої величини:

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



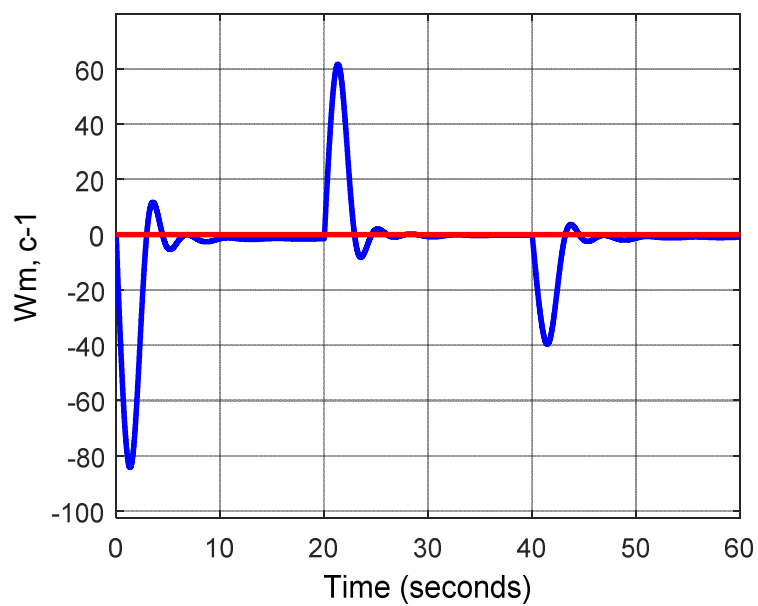
а)



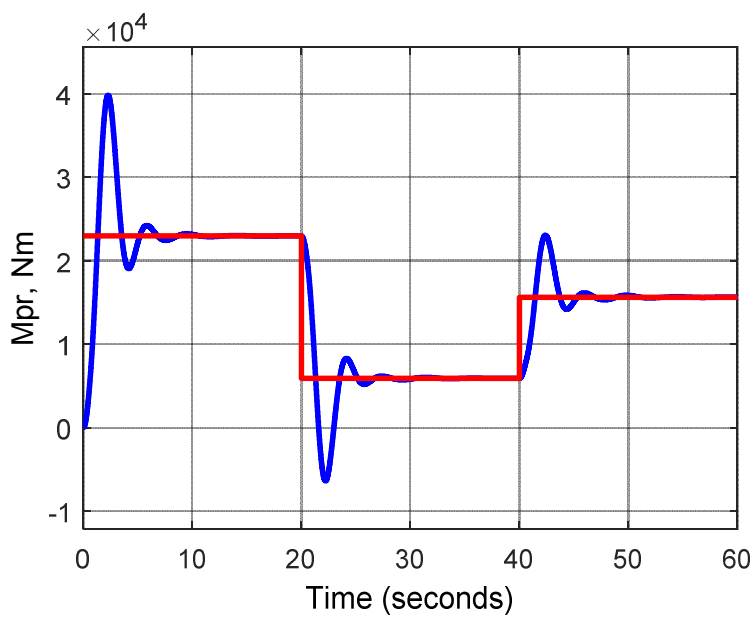
б)

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

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



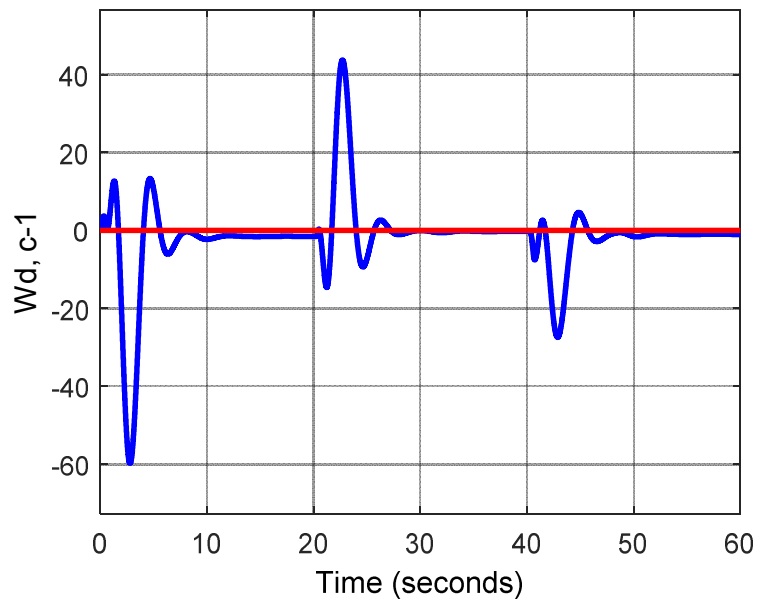
a)



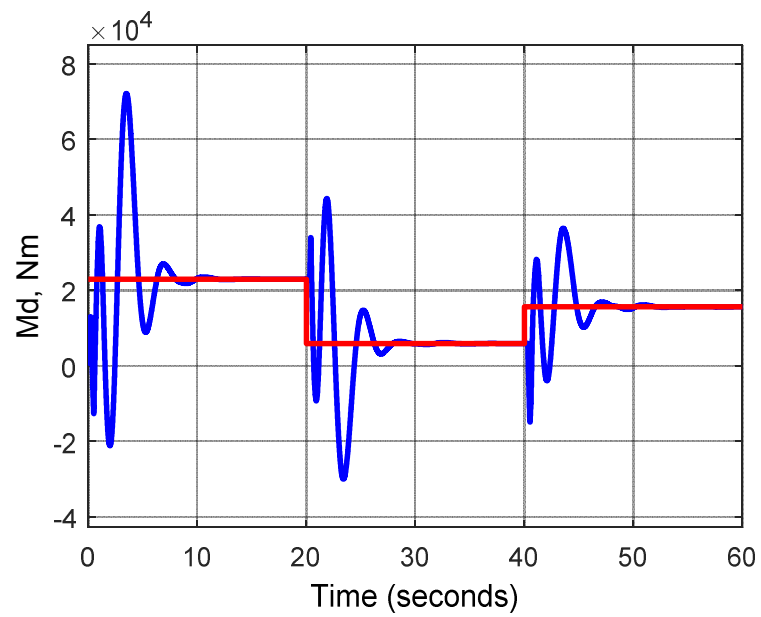
б)

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 $M_{np} = f(t)$



а)



б)

Рисунок 4.31 – Динаміка двигуна двомасової системи з нейрорегулятором під дією зовнішнього збурення:

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

ВИСНОВКИ ДО РОЗДІЛУ 4

1. У розділі проведено детальне дослідження сучасних підходів до побудови нейрорегуляторів, реалізованих у пакеті прикладних програм Neural Network Toolbox середовища MATLAB. Розглянуто три основні типи регуляторів: нейропрогнозуючий (NN Predictive Controller), регулятор, побудований на основі авторегресійної моделі з ковзним середнім (NARMA-L2 Controller), та регулятор із використанням еталонної моделі (Model Reference Controller). Порівняльний аналіз показав, що для досліджуваної системи електропривода найбільш придатним є NN Predictive Controller, оскільки він забезпечує можливість врахування майбутньої поведінки об'єкта та дозволяє компенсувати нелінійності й коливальні властивості двомасової механічної структури. Описано принципи побудови, архітектуру та послідовність синтезу вибраного регулятора.

2. У середовищі SIMULINK створено інтегровану модель системи керування, що включає двомасову модель механізму повороту роторного колеса та нейропрогнозуючий регулятор NN Predictive Controller. Структура моделі забезпечує можливість дослідження динамічних характеристик у замкненому контурі, а також дозволяє проводити моделювання перехідних режимів у реальному часі з використанням високоточної ідентифікаційної підсистеми.

3. На базі моделі виконавчого механізму сформовано навчальну вибірку для тренування нейронної мережі. Підбір параметрів навчальної послідовності здійснювався шляхом багаторазових експериментів, що дало змогу мінімізувати похибку моделі та забезпечити стійку збіжність процесу навчання. Оптимальні параметри вибірки включають: розмір 20000 точок, інтервал дискретизації 0,1 с, діапазон інтервалів ідентифікації від 8 до 20 с. Така конфігурація дозволила отримати репрезентативні дані й охопити широкий спектр режимів роботи електропривода.

4. Побудовано двошарову нейронну мережу прямого розповсюдження сигналу та здійснено її навчання за алгоритмом Левенберга–Маркуардта, який забезпечує високу швидкість збіжності та ефективну мінімізацію помилки. Ви-

значено раціональну кількість нейронів прихованого шару, що дозволяє досягти балансу між точністю моделі та складністю структури. Отримані результати тренування свідчать про високий рівень апроксимаційної здатності мережі: середня квадратична помилка становить величину порядку 10^{-4} , а максимальні миттєві відхилення у навчальній, валідаційній та тестовій підвбірках не перевищують 0,2.

5. На основі навченої нейронної моделі створено повнофункціональний NN Predictive Controller. Під час налаштування регулятора досліджено вплив його основних параметрів на якість регулювання, зокрема – горизонту прогнозування, довжини вікна оптимізації, коефіцієнтів ваги похибки та зміни управляючого впливу. Встановлено параметри, що забезпечують оптимальний компроміс між швидкодією та стійкістю системи. Показано, що використання точної нейромережевої моделі об'єкта дає змогу значно покращити реакцію системи на динамічні збурення й підвищити якість керування в умовах нелінійності та змінних навантажень.

6. Проведено всебічне моделювання роботи електропривода з упровадженим нейропрогнозуючим регулятором. Отримані графіки перехідних процесів демонструють, що система з NN Predictive Controller забезпечує високу стабільність роботи при запуску та прикладенні навантаження. Перерегулювання не перевершує 10%, час встановлення становить близько 4 с, що повністю відповідає вимогам до якості регулювання. Це підтверджує доцільність використання нейромережевих підходів у системах керування механізмами з вираженими пружними властивостями.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розв'язано важливу науково-прикладну задачу створення нейромережевої системи керування багатомасовим електромеханічним приводом механізму обертання роторного колеса екскаватора. Розроблене рішення спрямоване на підвищення якості регулювання, зменшення коливальних режимів та забезпечення стійкої роботи механізму в умовах змінного навантаження.

У процесі дослідження отримано такі основні результати:

1. *Науково обґрунтовано доцільність використання нейронних мереж у системах керування роторними екскаваторами.* На основі детального аналізу існуючих систем автоматичного регулювання, що застосовуються в механізмах великогабаритних гірничих машин, встановлено, що класичні методи керування мають обмеження при роботі з багатомасовими структурами, які характеризуються нелінійністю, наявністю пружних з'єднань та складними динамічними взаємодіями. Показано, що нейромережеві методи здатні ефективно компенсувати ці недоліки завдяки здатності до апроксимації складних динамічних залежностей і адаптації до змінних умов роботи. Це свідчить про перспективність і актуальність застосування нейронних регуляторів для задач керування механізмом повороту роторного колеса.

2. *Обґрунтовано вибір електропривода та силового обладнання.* Вивчення вимог до автоматизованих приводів роторних екскаваторів дозволило визначити, що для приводу механізму повороту найбільш раціонально застосувати систему постійного струму типу Г–Д. Виконано підбір електродвигуна, вибрано силове та комутаційне обладнання, а також визначено їх параметри відповідно до умов експлуатації та необхідної потужності. Це забезпечило формування бази для подальшого синтезу системи автоматичного керування.

3. *Розроблено і проаналізовано систему автоматичного керування електроприводом.* Сформовано структуру системи регулювання, що включає контур зворотного зв'язку за швидкістю, адаптивний (гнучкий) зворотний зв'язок за

струмом та контур захисного відсічення струму. Виконано розрахунок статичних характеристик розімкненої та замкненої системи, визначено передавальні функції за задавальною та збудуючою діями. Побудовано алгоритмічну схему та виконано комп'ютерне моделювання у MATLAB/Simulink. Отримані результати підтверджують працездатність класичного регулятора, але демонструють його обмеженість у випадку багатомасової механічної структури.

4. *Створено математичну модель двомасової механічної системи.* З урахуванням пружних властивостей валів, канатів і муфт виконано побудову двомасової моделі механізму обертання роторного колеса. Модель враховує жорсткість і внутрішнє демпфування, що дозволяє адекватно описувати реальні процеси. Моделювання в середовищі MATLAB показало, що у двомасовій структурі виникають істотні коливання швидкості та моментів, які негативно впливають на роботу екскаватора та можуть призводити до перевантаження механізмів. Це підкреслило необхідність застосування більш інтелектуальних підходів до керування.

5. *Виконано синтез нейромережевої системи керування на основі NN Predictive Controller.* У роботі розроблено нейропрогнозуючий регулятор, реалізований засобами Neural Network Toolbox. Детально розглянуто принцип його функціонування, етапи формування навчальної вибірки, навчання мережі та інтеграції в структуру привода. Проведено порівняльний аналіз впливу параметрів регулятора на якість керування та визначено їх оптимальні значення. Нейромережевий регулятор продемонстрував спроможність ефективно згладжувати коливання, зменшувати реакцію системи на збурення та стабілізувати роботу двомасової структури.

6. *Підтверджено ефективність застосування нейрорегулятора шляхом чисельного моделювання.* Результати моделювання показали, що система з NN Predictive Controller забезпечує суттєве покращення динаміки у порівнянні з класичним підходом. Перехідні процеси характеризуються швидким загасанням, перерегулювання не перевершує 10%, час встановлення близько 4 с, що повністю відповідає технічним вимогам. Система демонструє астатичні власти-

вості й здатність стабільно відпрацьовувати сигнали з випадковою амплітудою, що свідчить про її високу робастність.

Узагальнюючи, у роботі створено сучасну нейромережеву систему керування, яка враховує специфіку нелінійної багатомасової структури механізму обертання роторного екскаватора та забезпечує підвищену якість регулювання. Отримані результати підтверджують доцільність впровадження нейромережових регуляторів у практику управління складними електромеханічними системами та можуть бути використані для подальшої оптимізації приводів гірничих машин і технологічного обладнання.

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. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – 1958. – № 65.– P.386-407 (Имеется перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир, 1966. – 480 с.).
15. Kaur, *A Comprehensive Survey of Deep Learning Models Across Diverse Application Domains, International Journal of Intelligent Systems and Applications in Engineering*, 12(22s) (2024) 2176-2178
16. Zewen Li, Wenjie Yang, Shouheng Peng, Fan Liu, *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*, IEEE/NLN (2021) (Accepted Manuscript)
17. Neural networks for machine learning applications”, Sanjay Lote, Praveena K B, Durugappa Patrer, *World Journal of Advanced Research and Reviews*, 2020, 06(01), 270-282
18. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem / Proc. of IEEE First Ann. Int. Conf. on Neural Networks. – San Diego, 1987. – 3. – P. 11-13.
19. Hecht-Nielsen R. Theory of the Backpropagation Neural Network / Proc. of Int. Joint. Conf. on Neural Networks. – Washington, D.C., 1989. – 1. – P. 593-606.
20. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition // American Math. Soc. Transl. – 1963. – №28. – P. 55-63.
21. Cybenko G. Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314.

22. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // *Neural Networks*. – 1989. – 2. – P. 359-366
23. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks / *Neural Networks*. – 1991. – 4. – P. 251-257.
24. Pinkus, A. (1999). *Approximation theory of the MLP model in neural networks*. *Acta Numerica*, 8, 143–195.
25. Hopfield J.J., Tank D. W. Neural Computation of Decisions in Optimization Problems // *Biol. Cybernetics*. – 1985. – 52. – P. 141-152.
26. Sun S., Cao Z., Zhu H., Zhao J. A survey of optimization methods from a machine learning perspective // *IEEE Transactions on Cybernetics*. – 2020. – Vol. 50, No. 8. – P. 3668–3681.
27. Zhao R., Yan R., Chen Z., Mao K., Wang P., Gao R. X. Deep learning and its applications to machine health monitoring // *Mechanical Systems and Signal Processing*. – 2019. – Vol. 115. – P. 213–237.
28. Rawat W., Wang Z. Deep convolutional neural networks for image classification: A comprehensive review // *Neural Computation*. – 2017. – Vol. 29, No. 9. – P. 2352–2449.
29. Lim B., Zohren S. Time-series forecasting with deep learning: A survey // *Philosophical Transactions of the Royal Society A*. – 2021. – Vol. 379, No. 2194. – Article 20200209.
30. Liu H., Lang B., Liu M., Yan H. Machine learning and deep learning methods for cybersecurity // *IEEE Access*. – 2020. – Vol. 8. – P. 139 101–139 126.
31. Liu, H., Lang, B., Liu, M., Yan, H. Machine Learning and Deep Learning Methods for Intrusion Detection Systems: A Survey // *Applied Sciences*. – 2019. – Vol. 9, No. 20. – P. 4396. – MDPI
32. Aljawarneh, S., Aldwairi, M., Yassein, M. B. A Survey of Neural Networks Usage for Intrusion Detection Systems // *Journal of Ambient Intelligence and Humanized Computing*. – 2021. – Vol. 12, No. 3. – P. 3153–3172.
33. Kober J., Bagnell J. A., Peters J. Reinforcement learning in robotics: A survey // *The International Journal of Robotics Research*. – 2013. – Vol. 32, No. 11. – P.

1238–1274.

34. Esteva A., Robicquet A., Ramsundar B., et al. A guide to deep learning in healthcare // *Nature Medicine*. – 2019. – Vol. 25. – P. 24–29.
35. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J. T. A Review of Deep Learning Algorithms and Their Applications in Healthcare // *Journal of Biomedical Informatics*. – 2018. – Vol. 83. – P. 168–185.
36. Young, T., Hazarika, D., Poria, S., Cambria, E. Recent trends in deep learning based natural language processing // *IEEE Computational Intelligence Magazine*. – 2018. – Vol. 13, No. 3. – P. 55–75.
37. Runge, J., et al. Renewable energy forecasting: A review // *Renewable and Sustainable Energy Reviews*. – 2019. – Vol. 120. – 109642.
38. Kuster, C., Rezgui, Y., Mourshed, M. A Scoping Review of Deep Neural Networks for Electric Load Forecasting // *Energy Informatics*. – 2021. – Vol. 4, No. 1. – P. 1–26.
39. Amasyali, K., El-Gohary, N. M. A Review of Deep Learning Techniques for Forecasting Energy Use in Buildings // *Energy and Buildings*. – 2018. – Vol. 177. – P. 535–548.
40. Jain, A., Kumar, S. Deep learning in manufacturing: A review of applications, challenges, and future directions // *Journal of Manufacturing Systems*. – 2022. – Vol. 63. – P. 224–249.
41. Li, X., Zhang, W., Chen, X. Deep Learning Techniques in Intelligent Fault Diagnosis and Prognosis for Industrial Systems: A Review // *Sensors*. – 2023. – Vol. 23, No. 3. – P. 1305. – MDPI.
42. Wang, Y., Chen, Z., Li, H., et al. Data-Driven Machinery Fault Detection: A Comprehensive Review // *Mechanical Systems and Signal Processing*. – 2024. – Vol. 210. – 110013.
43. Astrom K J , Wittenmark B Adaptive control, Reading MA Addison Wesley, 1989.
44. Parks P C Lyapunov redesign of model reference adaptive control systems, *IEEE Journal Transactions of Automatic Control*, 1966, vol 11, pp 362-367.

45. Hornik K , Stinchcombe M , White H Mylti-layer feed-forward networks are universal approximators Discussion paper, Department of Economics, University of California, San Diego, La Jolla, CA, 1988.
46. Haykin S , Neural Networks A comprehensive foundation, MacMillan College Publishing Co , New York, 1994.
47. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
48. Narendra K S , Parthasarathy K Neural networks and dynamical systems Part 1 A gradient approach to Hopfield networks TR 8820, Center for systems science, Department of Electrical Engineering, Yale University, New Haven, CT, 1988.
49. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
50. Werbos P J , Beyond Regression New Tools for Prediction and Analysis in the Behavioral Sciences, Phd Thesis, Dept of Applied Mathematics, Harvard University, Cambridge, Mass , 1974.
51. Pobitxakic Г. А.: «Stable adaptive neuro-control design via Lyapunov function derivative estimation» // Automatica. – 2001. – Vol. 37, No. 8. – P. 1213–1221.
[ResearchGate+3ResearchGate+3dl.acm.org+3](#)
52. Tenavi Nakamura-Zimmerer, Qi Gong, Wei Kang: «Neural network optimal feedback control with guaranteed local stability» // arXiv preprint. – 2022. – ID arXiv:2205.00394
53. Gao, Q., Li, H., & Sun, Y. Adaptive neural network-based control for a class of nonlinear pure-feedback systems with time-varying full state constraints // Journal of Applied Science & Engineering. – 2018. – Vol. 11, No. 3. – P. 203-218.
54. Park, J. W., Harley, R. G., Venayagamoorthy, G. K. Indirect adaptive control for synchronous generator: Comparison of MLP/RBF neural networks approach with Lyapunov stability analysis // IEEE Trans. Neural Networks. – 2004. – Vol. 15, No. 2. – P. 460-464. [Yonsei University](#)

55. Xiong, Z., Li, J., & Sun, C. Fault-tolerant control for nonlinear systems using sliding mode and adaptive neural network estimator // *Soft Computing*. – 2020. – Vol. 24. – P. 11535-11544. [SpringerLink](#)
56. Fu, X., Li, S., Wunsch, D. C., Alonso, E. Local stability and convergence analysis of neural network controllers with error integral inputs // *Elec. & Comp. Engineering Res. & Creative Works*. – 2018. – P. (выпуск)-(сторінки). [Scholars' Mine](#)
57. Garcia C.E. Model predictive control: theory and practice a survey / Garcia C.E., Prett D.M., Morari M. // *Automatica*. – 1989. – Vol.25. – P.335-348.
58. 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.
59. 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.
60. Review on model predictive control: an engineering perspective – *The International Journal of Advanced Manufacturing Technology*, 2021, Vol. 117, pp. 1327-1349.
61. Lan J. *Efficient model predictive control for nonlinear systems modelled by deep neural networks* : e-print (arXiv) / J. Lan. - 2024..
62. Neural networks for control / eds. W. T. Miller, R. S. Sutton, P. J. Werbos. – Cambridge, MA : The MIT Press, 1990. – 537 p.
63. Solowey D. Neural Generalized Predictive Control / Solowey D., Haley P.J // *Proceeding of the 1996 IEEE Symposium on Intelligent Control*. – 1996. – P. 277–282.
64. Clarke D.W., Mohtadi C., Tuffs C. Generalized Predictive Control - Part 1: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 137-148.
65. Clarke D. W., Mohtadi C., Tuffs C. Generalized Predictive Control. Part 2: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 149-163.
66. Clarke D. W. Advances in model-based predictive control.- In *Advances in*

- Model-Based Predictive Control / D. W. Clarke: Oxford University Press. – 1994. – 274
67. Montague G. A., Willis M. J., Tham M. T., Morris A. J. Artificial Neural Network Based Control // Int. Conference on Control. – 1991. – Vol.1. – P. 266-271.
68. Takahashi Y. Adaptive Predictive Control of Nonlinear Time-Varying System using Neural Network //IEEE International Conference on Neural Networks, 1993. – Vol.3. – P.1464-1468
69. Jeong Jun Song, Sunwon Park. Neural Model-Predictive Control for Nonlinear Chemical Processes // Journal of Chemical Engineering of Japan. – 1993. – Vol.26. – №4. –P.347-354.
70. Psichogios D. C., Ungar L. H. Nonlinear Internal Model Control and Model Predictive Control using Neural Networks // 5th IEEE Int. Symposium on Intelligent Control. – 1990. – P.1082-1087.
71. Press W. H., Flannery B. P., Teukolsky S. A., Vetterling V. T. Numerical Recipes.- The Art of Scientific Computing. Cambridge University Press, 1988. – 452.
72. Rumelhart D. E., McClelland J. L. and the PDP Research Group. Parallel Distributed Processing. The MIT Press. – 1986. – Vol.1. – Chapter 8. – 337 p.
73. Теорія автоматичного керування : навчальний посібник / П. В. Леонт'єв та ін. ; за заг. ред. П. В. Леонт'єва. – Суми : Сумський державний університет, 2024. – 296 с.
74. 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.
75. 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.
76. Теорія автоматичного керування : навчальний посібник / П. В. Леонт'єв та ін. ; за заг. ред. П. В. Леонт'єва. – Суми : Сумський державний університет, 2024. – 296 с.
77. Теорія автоматичного керування : навчальний посібник / О. К. Аблесімов

– К. : «Освіта України», 2019. – 270 с.

78. 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.

79. Теорія автоматичного управління : курс лекцій / Т. М. Боровська. – Вінниця : ВНТУ, 2018. – 256 с.

80. 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

81. 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.

82. Автоматизований електропривод. Розділ 1. Теорія електропривода. Навчальний посібник. / Н.Д. Красношاپка – К.: КПІ ім. Ігоря Сікорського. - 2023. – 101 с.

83. Возняк, О.М., Штуць. А.А., Колісник М.А. Сучасні системи електроприводів. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ТВОРИ, 2021. – 280 с.

84. А. А. Видмиш, Л. В. Ярошенко. Основи електропривода. Теорія та практика. Частина 1. Навчальний посібник. – Вінниця: ВНАУ, 2020. – 387 с.

85. 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.

86. Колб Ант. А, Колб А. А. Теорія електроприводу: Навчальний посібник. – 2-е вид. перероб. і доп. – Д., Національний гірничий університет, 2011. – 540 с.

87. 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.

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

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

90. 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.

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

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

ДОДАТОК А

БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ

А.1 Структура штучного нейрона

Основним елементом всіх мереж є нейрон. Штучний нейрон відображає структуру біологічного нейрона (див. рис.А.1). Він має декілька входів і один вихід, який може бути пов'язаний з багатьма входами інших нейронів. Кожен

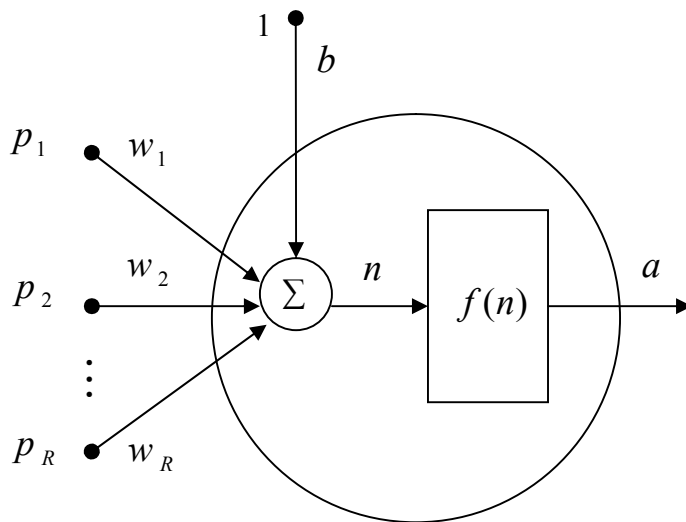


Рисунок А.1 - Схема штучного нейрона

подібний зв'язок має свій коефіцієнт посилення. Нейрон вчиться шляхом модифікації коефіцієнтів посилення відповідно до закону навчання.

Нейрон реалізує скалярну функцію векторного аргументу. Сигнали обробляються в нейроні таким чином. Кожен вхідний сигнал p_i , множиться на відповідний ваговий коефіцієнт w_i і складається з ін-

шими зваженими вхідними сигналами. Порогова величина, яка позначена як b розглядається в більшості випадків як додатковий ваговий коефіцієнт w_0 . Потім отримана сума n перетвориться за допомогою функції активації або передавальної функції f . Як передавальна функція в нейроні може використовуватися принципово будь-яка функція. Вид активаційної функції надає істотний вплив на показники якості регулювання електромеханічної системи при нейроконтрольному управлінні. Для навчання методом зворотного розповсюдження помилки необхідна передавальна функція, що диференціюється. У таблиці А.1 представлені найчастішим використовувані передавальні функції.

Таблиця А.1 – Передавальні функції

Назва	Функція	Похідна
Лінійна	$f(n) = n$	$f'(n) = 1$
Ступінчаста (порогова)	$f(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$	$f'(n) = 0$
Сигмоїдальна (логістична) sigmoid	$f(n) = \frac{1}{1 + e^{-n}}$	$f'(n) = f(n)(1 - f(n))$
Гіперболічний тангенс tang	$f(n) = \frac{e^n - e^{-n}}{e^n + e^{-n}}$	$f'(n) = 1 - f^2(n)$
ReLU	$f(n) = \begin{cases} 0, & n < 0 \\ n, & n \geq 0 \end{cases}$	$f'(n) = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$
SoftPlus	$f(n) = \log(1 + e^n)$	$f'(n) = \frac{1}{1 + e^{-n}}$
Leaky ReLU, Parameterized ReLU	$f(n) = \begin{cases} an, & n < 0 \\ n, & n \geq 0 \end{cases}$	$f'(n) = \begin{cases} a, & n < 0 \\ 1, & n \geq 0 \end{cases}$
ELU	$f(n) = \begin{cases} a(e^n - 1), & n < 0 \\ n, & n \geq 0 \end{cases}$	$f'(n) = \begin{cases} f'(n) + a, & n < 0 \\ 1, & n \geq 0 \end{cases}$

Рівняння нейрона має вид

$$a = f\left(\sum_{i=1}^R w_i \cdot p_i + b\right).$$

А.2 Одношарові нейронні мережі

Внутрішній потенціал нейронних технологій виявляється при об'єднанні окремих нейронів в нейромережеві структури. З огляду на те, що в даний час існує величезна кількість різних типів мереж, неможливо розглянути кожний з них детально. Тому далі виклад обмежується тільки фактично використовуваними в цій роботі типами мереж. Сукупність нейронів з однаковими вхідними сигналами називають шаром. Одношарова прямонаправлена мережа зв'язується зазвичай з поняттям одношаровий персептрон. Якщо нейрон має одну з сигмоїдних функцій, то значення його вихідної величини будуть обмежені в деякому діапазоні. Вихід лінійного нейрона може приймати будь-які значення.

В результаті об'єднання декілька нейронних шарів виникає типова багатошарова мережа – багатошаровий персептрон (БП). БП тренується переважно алгоритмом зворотного розповсюдження помилки. Одношарові мережі описується однозначно вказівкою числа входів R і числа нейронів S (число виходів S), і, типом передавальної функції. Розгорнута схема мережі показана на рис.А.2.

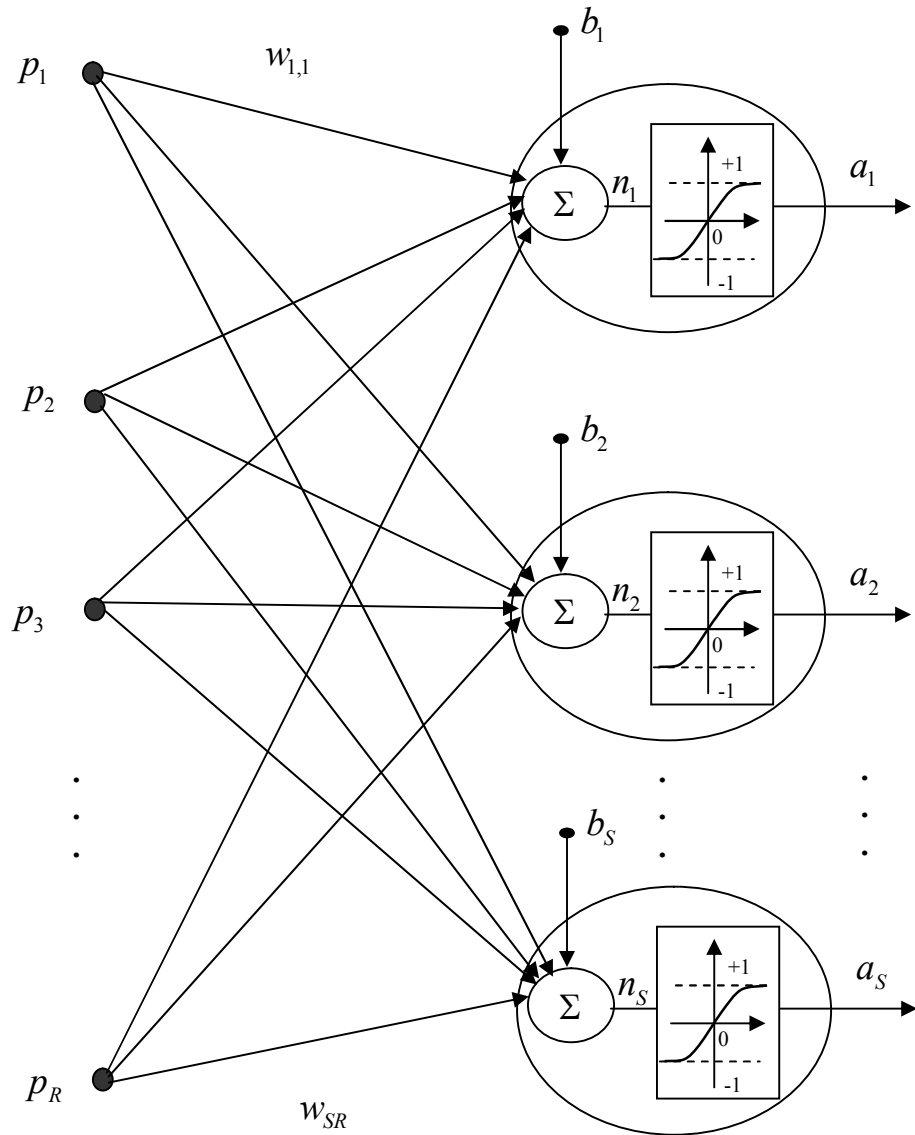


Рисунок А.2 - Структурна схема одношарової мережі
(одношаровий персептрон)

В 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 ней-

рон 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 шару. Виходи шару нейронів формують вектор-стовпець $\mathbf{a}$, і, таким чином, опис шару нейронів має вигляд:

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

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.. А.2, 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:

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

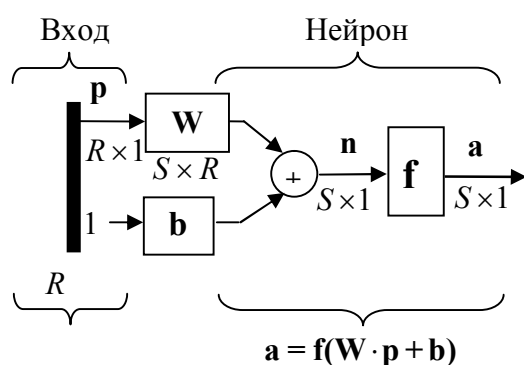


Рисунок А.3 – Укрупнена структурна схема одношарової мережі

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 $w_{1,2} = \mathbf{W}(1,2)$ визначає коефіцієнт, на який множиться другий елемент входу при передачі його на перший нейрон. Укрупнена структурна схема одношарової мережі показана на рис. А.3.

А.3 Багат шарові нейронні мережі

Для багат шарового персептрона повинні указуватися кількість і тип нейронів всіх прихованих шарів і вихідного шаруючи. Вхідний шар, як правило, не вважається шаром, оскільки сигнали, що поступають в нього, не змінюються. На рис.А.4 представлений БП з R входами, S^1 і S^2 нейронами в прихованих шарах і S^3 нейронами у вихідному шарі. Ця ж тришарова мережа може бути представлена у вигляді укрупненої структурної схеми (рис.А.5).

Перевага цих мереж в тому, що принцип їх роботи описується простими матричними операціями. Для матриць вагів входу і виходу шару використовуються позначення **IW** (Input Weight) і **LW** (Layer Weight) відповідно (такі позначення прийняті в системі MATLAB, яка використовувалася для досліджень в даній роботі). Вихід останнього шару $\mathbf{a}^3$ позначений через $\mathbf{y}$. Це зроблено для того, щоб підкреслити, що вихід останнього шару є виходом мережі.

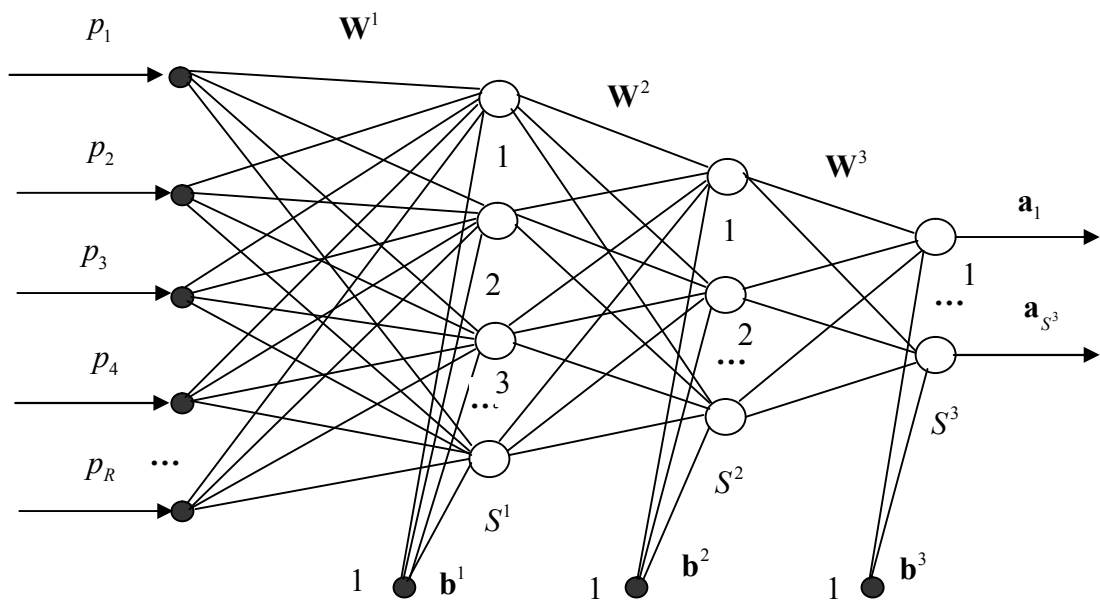


Рисунок А.4 – Структурна схема тришарової мережі
(тришаровий персептрон)

Як правило, для вихідного шару вибирається лінійна передавальна функція, щоб уникнути обмежень вихідної величини в деякому діапазоні. Застосу-

вання декількох наступних один за одним лінійних шарів було б даремним, оскільки їх можна об'єднати без втрати інформації в один шар. Про оптимальний вибір передавальної функції прихованих шарів не існує до цих пір ніяких тверджень. З прагматичних міркувань на практиці призначають в межах шару однакові передавальні функції і найчастіше це сигмоїдні функції. З погляду здібностей апроксимацій не має ніякого значення, яка з двох сигмоїдних функцій вибирається, оскільки одну можна перераховувати в іншу. МП є універсальним апроксиматором-класифікатором і належить в сьогоdnішній практиці до найбільш використовуваних нейронних мереж.

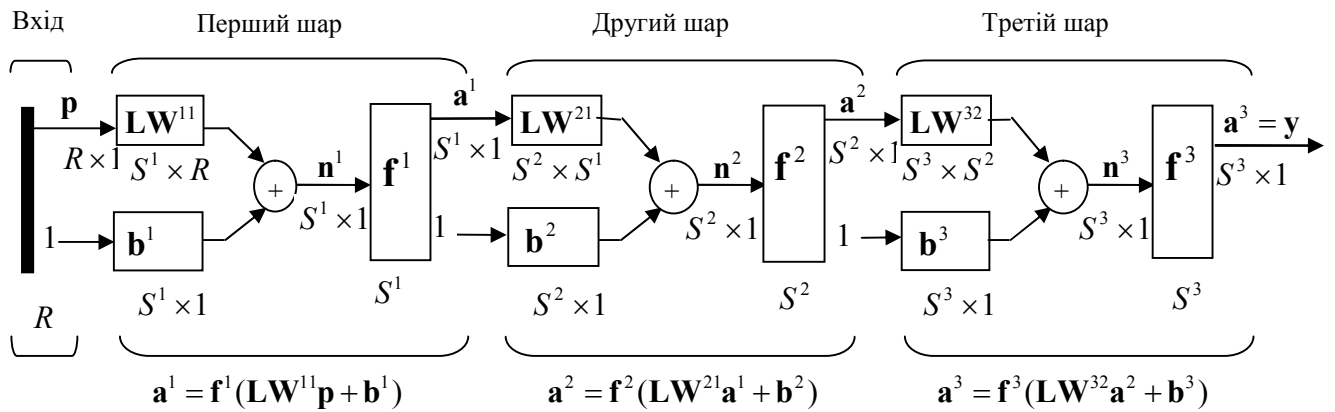


Рисунок А.5 – Укрупнена структурна схема тришарової мережі

А.4 Здібності апроксимації і узагальнення багатшарових мереж

Принципово в більшій частині практичного застосування нейронних мереж йдеться, про те, щоб реалізувати відображення функції за допомогою кінцевого числа параметрів мережі. При цьому повинні виконуватися дві наступні вимоги: апроксимувати представлений для навчання набір даних; узагальнити отримані знання на невідомі вхідні дані.

Виконання першої вимоги означає знайти мережу, яка може з будь-якою точністю вивчити будь-яку задану функцію. Проблема відображення багатовимірної нелінійної функції за допомогою лінійної суперпозиції простих нелінійних функцій сходиться до так званої 13-ої проблеми Гілберта. Гілберт припускав,

що ця проблема не має рішення. Це припущення було спростоване теоремою Колмогорова-Арнольдса в кінці 50-х років. В кінці 80-х років ця теорема була переведена на термінологію нейронних мереж Р. Хехт-Нильсоном і Цибенко.

Публікації останніх років зачіпали питання здібностей апроксимацій прямонаправлених нейронних мереж. Для докладного розгляду доказів і деталей слід звернутися до спеціальної літератури. Далі представляються тільки підсумки математичних викладень:

- Теоретично завжди існує така прямонаправлена нейронна мережа з одним нелінійним прихованим шаром і лінійним вихідним шаром, яка здатна відображати будь-які шматково-безперервні багатовимірні функції.
- Необхідна якість апроксимації завжди досягається при кінцевому числі нейронів, якщо задана певна точність апроксимації.

Тільки для небагатьох нейронних структур в сьогодення існує строге математичне обґрунтування їх здібностей. БП в даний момент краще всього досліджений теоретично:

- Збіжність процесу навчання доведена теоремою Perceptron-Konvergenz, яка стверджує, що функція може бути апроксимована за кінцеве, але не відоме, число кроків.
- БП з одним прихованим шаром формує гіперплощині. При двох прихованих шарах утворюються замкнуті гіперпростори, які досить, щоб вирішувати які завгодно по складності завдання класифікації. Подальші шари нічого не покращують, а збільшують час навчання.

Оцінка числа нейронів в прихованих шарах мережі – все ще нетривіальне завдання. Не підлягає сумніву факт вирішального впливу на розмір мережі ступеня нелінійності об'єкту, що ідентифікується. Існують численні емпіричні формули, що дають мінімальне і максимальне число нейронів або зв'язків для забезпечення необхідної точності апроксимації. Але на сьогоднішній день вони дуже неточні. Запропоновані також методи автоматичного підбору числа нейронів в процесі тренування, які можна розділити на дві групи: група редукуючих методів і група конструюючих методів. Редукуючі алгоритми менш ефек-

тивні, оскільки вони мають, принаймні, два недоліки: надмірне число нейронів при ініціалізації мережі невідоме і тому оцінюється приблизно, із-за явно надмірного числа нейронів процес тренування протікає досить поволі. Сьогоднішня практика свідчить, що систематичний підбір оптимального числа нейронів також є ефективним. З вищесказаного виходить, що для успішної ідентифікації деякої нелінійної системи мають значення вибір типу мережі, виду функції активації, числа нейронів (а для БП ще і числа прихованих шарів).

Здібності нейронних мереж до узагальнення дозволяють проводити тренування при одних парах тренувальних даних і після цього отримувати правильний вихід при інших можливих вхідних даних. Виникаюча проблема формування інформативних тренувальних даних не вирішена в загальному вигляді до цих пір. Проте, успішно практикується наступний емпіричний метод. Узагальнювані вхідні дані не повинні сильно відрізнятися від тренувальних даних. Це вимушує обмежуватися при навчанні небагатьма крапками даних, між якими мережею виконуватиметься нелінійна інтерполяція.

ДОДАТОК Б

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

Б.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. Повчальний 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.

Б.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 мінімуму 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 точкою в $N+1$ -вимірному просторі, а всі такі точки 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 поверхня помилок має набагато складнішу будову і має ряд несприятливих властивостей, зокрема може мати локальні мінімуми, плоскі ділянки, сідлові точки 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 точки на поверхні функції помилок, алгоритм навчання поступово відшукує глобальний мінімум. Як правило, для цього обчислюється градієнт (нахил) функції помилок в даній точці, 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 мінімуму 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.

Б.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 реа-

льній 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 багат шарові 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 функ-

цію помилок помилка на контрольній множині також убуватиме. Якщо ж контрольна помилка перестала убувати 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 вибирати 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.

Б.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 збуреннях. 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 результативно 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 повчальні 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.

Б.5 Вибір методу навчання нейронної мережі

Б.5.1 Критерій якості навчання. Методи навчання.

Фаза навчання є найбільш проблематичною при використанні нейронних мереж. Для БП мереж процес навчання є багатозмінним нелінійним оптимізаційним завданням. Її аналітичне рішення неможливе, що вимагає використання ітераційних методів оптимізації.

У загальному випадку початковими даними є: структура мережі, тренувальні дані і критерій оцінки якості тренування. Мета оптимізації – визначити параметри мережі, що забезпечують виконання критерію якості. Таким критерієм якості найчастіше виступає мінімум квадратичного функціонала:

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

де J – функціонал;

Q – об'єм вибірки;

L – число шарів мережі;

q – номер вибірки;

S^L – число нейронів вихідного шаруючи;

$\mathbf{a}^q = [a_i^{qL}]$ – 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 настроюються, мережі, які забезпечують мінімальне значення функціонала помилки.

Методика тренування нейронних мереж була в основному запозичена з відомих методів теорії оптимізації. Зважаючи на величезне число створених алгоритмів тут можна дати лише їх короткий огляд. Вживані методи навчання грубо можна розділити на наступні основні групи: методи глобальної оптимізації; градієнтні методи першого і другого порядку.

Істотною проблемою, що виникає при використанні градієнтних методів настройки параметрів ШНМ, є їх локальність. В той же час цільова функція (сумарна помилка на тренувальному наборі шаблонів або кумулятивна оцінка ефективності роботи навчаної системи) не унімодальна. Кількість локального оптимуму для більшості практичних завдань навчання обчислюється мільйонами при розмірності пошукового простору порядку $10^2 - 10^3$. Внаслідок цього результат навчання залежить від правильності вибору стартової крапки, і виникає необхідність багатократного повторення процедури настройки параметрів ШНМ. Перераховані проблеми можуть бути вирішені при використанні методів глобальної оптимізації. Найбільш ефективним з них є генетичний алгоритм (ГА). Розглядаючи ШНМ як єдиний набір параметрів, ГА здатний здійснювати її оптимальну настройку при розмірності пошукового простору достатньою для вирішення більшості практичних завдань.

У останні десятиліття розроблена безліч способів навчання ШНМ за допомогою ГА. Отримані результати доводять великі можливості такого симбіозу. Сумісне використання ШНМ і ГА алгоритмів має і ідеологічну перевагу тому, що вони відносяться до методів еволюційного моделювання і розвиваються в рамках одного підходу запозичення технікою природних методів і механізмів як найбільш оптимальних.

Слід зазначити, що завдання глобальної оптимізації вирішуються шляхом систематичного перебору всіх значень всіх аргументів, тому із-за експоненціального зростання складності завдання із зростанням її розмірності практичне застосування цих методів для великих мереж дуже утруднено.

У сучасних завданнях використовуються в основному градієнтні методи навчання. Вони відносно швидко знаходять оптимум. Проте при цьому відсут-

ній доказ того, що знайдений абсолютний екстремум. Численний досвід тренування мереж свідчить, що застосування методів локальної оптимізації сумісне з «виштовхуванням» мережі з локального мінімуму і послідовним збільшенням числа нейронів також приводить до успішного навчання мережі. У справжній роботі використовуються градієнтні методи навчання, тому розглянемо їх детальніше.

Б.5.2 Метод зворотного розповсюдження помилки.

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 (ЗР), який був розроблений в 1986г. Потім з'явилися численні роботи по прискоренню збіжності алгоритму. Метод звор-

тного розповсюдження помилки є першим теоретично доведеним алгоритмом для настройки вагів при навчанні багат шарового персептрона. В даний час ЗР може розглядатися як еталонний тест, з яким порівнюються всі інші методи навчання.

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 алгоритму зворотного розповсюдження. Звернемося до одного з них.

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

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

де S^L – число нейронів в шарі;

a_k^{qL} – k -й елемент вектора виходу шаруючи L для елемента вибірки з номером q .

Правило функціонування шару L записується так:

$$a_k^{qL} = f_L \left(\sum_{i=0}^{S^{L-1}} w_{ij}^L a_i^{q(L-1)} \right), \quad k = \overline{1, S^L}.$$

Приймаючи до уваги, що

$$\frac{\partial a_k^{qL}}{\partial w_{ij}^L} = \begin{cases} 0, & k \neq i \\ f'_L(n_i^{qL}) a_j^{q(L-1)}, & k = i \end{cases} \quad i = \overline{1, S^L}; \quad j = \overline{0, S^{L-1}},$$

отримаємо

$$\frac{dJ}{dw_{ij}^L} = - \sum_{q=1}^Q (t_i^q - a_i^{qL}) f'_L(n_i^{qL}) a_i^{q(L-1)}.$$

Ввівши позначення для локальної помилки вихідного шару

$$\Delta_i^{qL} = (t_i^q - a_i^{qL}) f'_L(n_i^{qL}), \quad i = 1, \dots, S^L,$$

отримаємо

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

Перейдемо шару $L-1$ і запишемо співвідношення для настройки вагів w_{ij}^{L-1}

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

де Δ_i^{qL} – локальна помилка шару $L-1$

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

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

$$\frac{dJ}{dw_{ij}^r} = \sum_{q=1}^Q \Delta_i^{q(r-1)} a_j^{q(r-1)}, \quad r = \overline{1, L}, \quad i = \overline{1, S^r}, \quad i = \overline{0, S^{r-1}}, \quad (\text{Б.2})$$

де r – номер шару

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

$$\Delta_i^{qL} = (t_i^q - a_i^{qL}) f'_L(n_i^{qL}), \quad i = \overline{1, S^{L-1}}.$$

На рис. Б.1 представлена схема обчислень, відповідна виразу (Б.2).

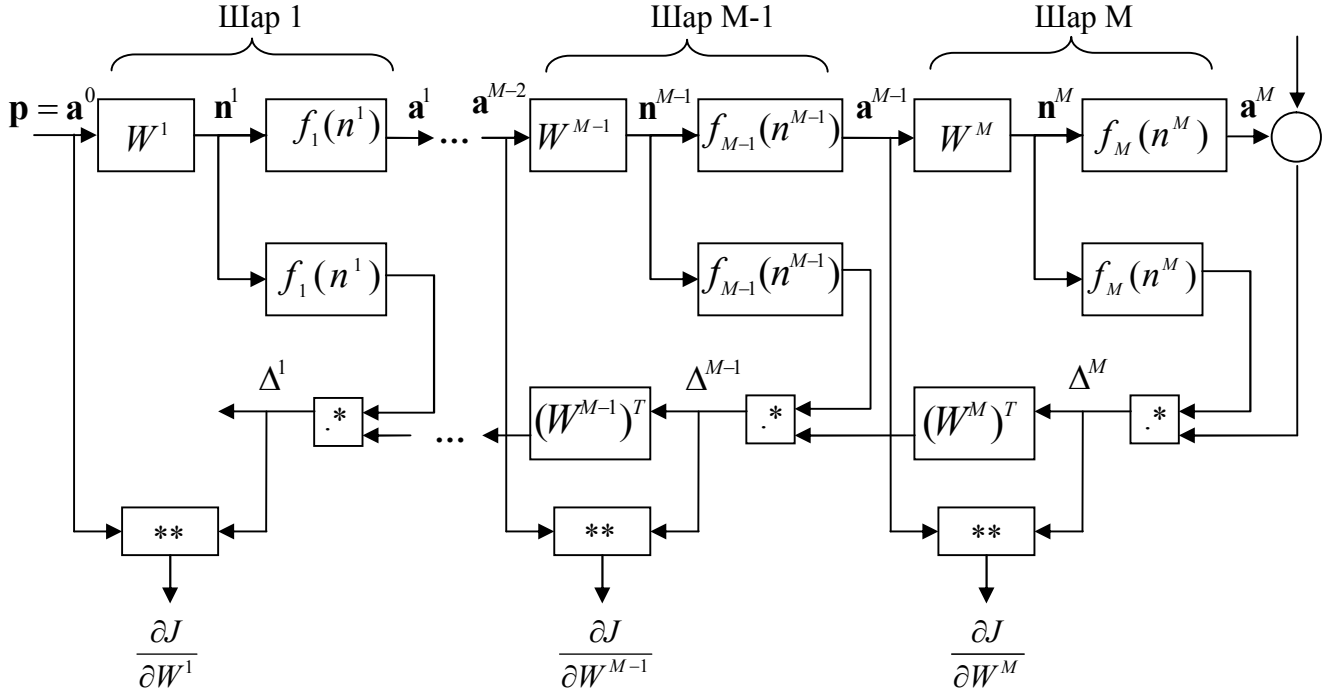


Рисунок Б.1 – Схема обчислення приватних похідних критерію J по елементах матриць вагових коефіцієнтів

На цій схемі символом $*$ позначена операція поелементного множення векторів, а символом $**$ – множення вектора Δ на $\mathbf{a}^T$; символ, що позначає номер елемента вибірки, скорочено опущений.

Б.5.3 Налаштування вагових коефіцієнтів мережі.

Методи, використовувані при навчанні нейронних мереж, багато в чому аналогічні методам визначення екстремуму функції декілька змінних. Вони діляться на дві категорії – методи першого і другого порядку.

У методах *першого* порядку використовується градієнт функціонала по параметрах, що настраюються, і на кожній ітерації k уточнення вектора вагів $\mathbf{w}_k$ проводиться по формулі

$$\mathbf{w}_{k+1} = \mathbf{w}_k - a_k \mathbf{g}_k, \quad (\text{Б.3})$$

де a_k – параметр швидкості навчання;

$\mathbf{g}_k$ – градієнт функціонала, відповідний 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 настраюються, 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. Згідно (Б.3) уточнюються значення вагів $\mathbf{w}_k$ в методі ЗР в його стандартній реалізації.

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

$$\mathbf{p}_0 = -\mathbf{g}_0.$$

Тоді напрям наступного руху визначається так, щоб він був зв'язаний з попереднім. Відповідний вираз для нового напрямку руху є комбінацією нового напрямку найшвидшого спуску і попереднього напрямку:

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \cdot \mathbf{p}_{k-1}.$$

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, що настраюються, $\mathbf{w}_{k+1}$ обчислюється за формулою

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \cdot \mathbf{p}_k.$$

Гradientні методи першого порядку характеризуються малою швидкістю збіжності. Це породило помилкову думку про те, що всі нейронні мережі вимагають для тренування великих витрат часу.

Потужніші gradientні методи *другого* порядку базуються на методі локальної оптимізації Ньютона, який при пошуку оптимуму додатково враховує ще і вигин поверхні функціонала якості (другі похідні).

Основний крок методу Ньютона визначається по формулі

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1} \mathbf{g}_k,$$

де $\mathbf{H}$ – матриця Гессе других похідних функціонала якості по вагових коефіцієнтах мережі.

У багатьох випадках метод Ньютона сходиться швидше, ніж методи зв'язаного gradientа, 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 квазіньютонівські алгоритми такі як алгоритм, запропонований Бройлером, Флетчером, Гольдфарбом і Шанно, однокроковий алгоритм методу січних площин OSS, алгоритм LM Левенберга-Марквардта і ін.

У даній роботі використовуємо алгоритм Левенберга-Марквардта. Даний алгоритм реалізує наступну стратегію для оцінки матриці Гессе. Оскільки функціонал якості визначається як сума квадратів помилок, то гессиан може бути приблизно обчислений як

$$\mathbf{H} \cong \mathbf{J}^T \mathbf{J}, \quad (\text{Б.4})$$

а градієнт розрахований по формулі

$$\mathbf{g} \cong \mathbf{J}^T \mathbf{e},$$

де $\mathbf{J} = \frac{\partial J}{\partial \mathbf{w}}$ – матриця Якобі похідних функціонала помилки по параметрах, що настраюються;

$\mathbf{e}$ – вектор помилки мережі.

Матриця Якобі може бути обчислена на основі стандартного методу зворотного розповсюдження помилки, що істотно простіше за матрицю Гессе.

У алгоритмі Левенберга-Марквардта нове значення вектора параметрів $\mathbf{w}_{k+1}$, що настраюються, обчислюється за формулою

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^T \mathbf{e}_k.$$

Коли коефіцієнт μ рівний 0, отримуємо метод Ньютона з наближенням Гессиана у формі (Б.4), коли значення μ велике, отримуємо метод градієнтного спуску з маленьким кроком. Оскільки метод Ньютона має велику точність і швидкість збіжності поблизу мінімуму, завдання полягає в тому, щоб в процесі мінімізації щонайшвидше перейти до методу Ньютона. З цією метою параметр μ зменшують після кожної успішної ітерації і збільшують тільки тоді, коли пробний крок показує, що функціонал помилки зростає. Така стратегія забезпечує зменшення помилки після кожної ітерації алгоритму.

ДОДАТОК В

УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ

В.1 Алгоритм нейроуправління з прогнозом

У роботі використовується ефективна реалізація узагальненого управління з прогнозом (Generalized Predictive Control, GPC) з використанням багатошарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління [63]. Завдяки використанню оптимізаційного алгоритму Ньютона-Рафсона, число ітерацій, необхідних для збіжності, значно менше, ніж при використанні інших методів. Головні витрати алгоритму Ньютона-Рафсона в обчисленні Гесиана, але навіть з цими додатковими розрахунками низьке число ітерацій робить алгоритм Ньютона-Рафсона швидшим, ніж інші методи, при цьому він може використовуватися для управління в режимі реального часу (real-time control).

Розглянемо стисло основні положення узагальненого алгоритму нейроуправління з прогнозом (Neural Generalized Predictive Control algorithm) із застосуванням оптимізаційного алгоритму Ньютона-Рафсона.

Узагальнене управління з прогнозом, введене Кларком і його співробітниками в 1987, належить до класу методів цифрового управління, званих модель-орієнтоване управління з прогнозом (Model-Based Predictive Control, MBPC) [64-66]. MBPC техніки були успішно проаналізовані і впроваджені в процес управління виробництвом в кінці 1970-х і продовжують використовуватися, оскільки вони системно враховують обмеження реальних об'єктів в режимі реального часу. GPC застосовуються в управлінні немінімально фазовими об'єктами, нестійкими об'єктами і об'єктами із змінним або невідомим часом затримки. Узагальнене управління з прогнозом також є робастним по відношенню до помилок моделювання, перевизначення і недовизначення значень параметрів і шумів датчиків [64]. GPC було спочатку розроблене для моделей лінійних об'єктів, що приводить до рівнянь, які можуть бути вирішені аналіти-

чно. При використанні нелінійних моделей необхідні нелінійні оптимізаційні алгоритми. Це впливає на якість і ефективність обчислень, що визначають вхідні сигнали, що управляють. Для нелінійних об'єктів здатність GPC робити точні прогнози може бути вдосконалена, якщо для вивчення динаміки об'єкту замість стандартних методів нелінійного моделювання застосовувати нейронні мережі. Вибір алгоритму мінімізації впливає на обчислювальну ефективність алгоритму. Як було вказано, при використанні для оптимізації алгоритму Ньютона-Рафсона число ітерацій, необхідних для збіжності, істотно менше, ніж при використанні інших методів.

Схема системи узагальненого нейроуправління з прогнозом (Neural Generalized Predictive Control, NGPC) показана на рис. В.1.

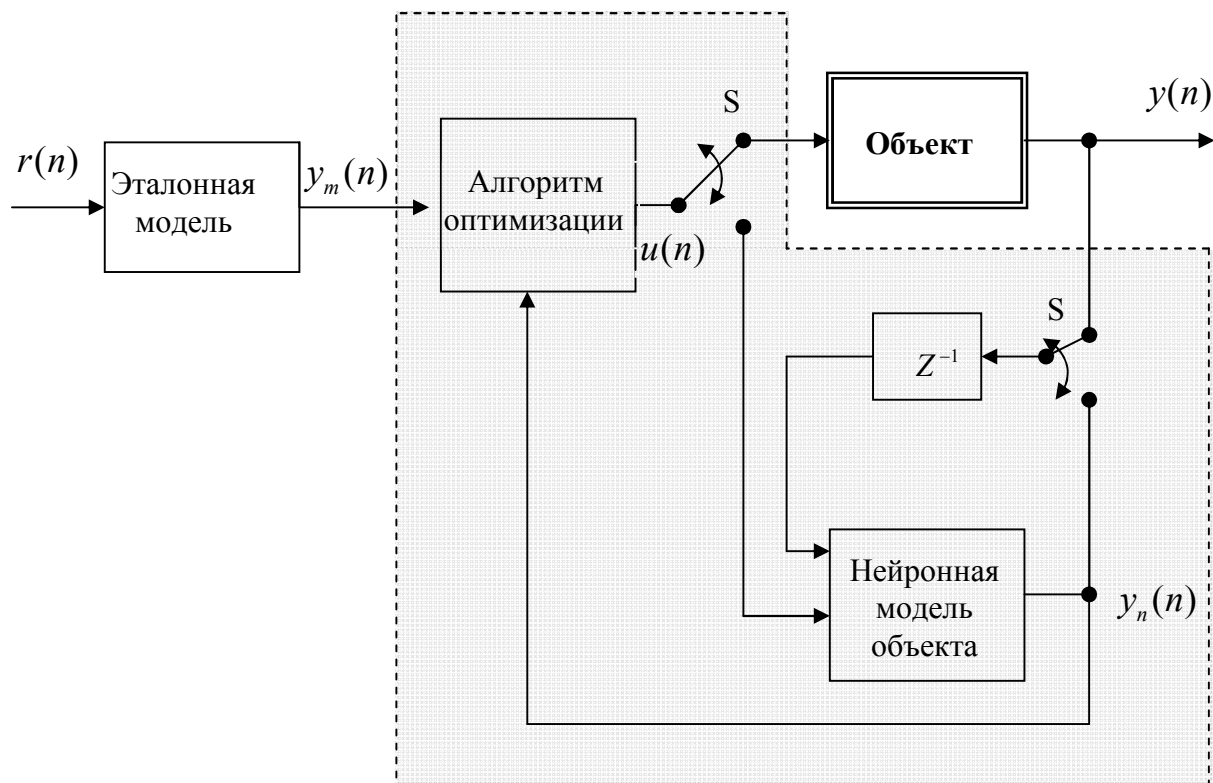


Рисунок В.1 – Схема системи узагальненого нейроуправління з прогнозом

Вона складається з чотирьох компонент: керованого об'єкту, еталонної моделі, яка описує бажану якість об'єкту, що моделює об'єкт нейронної мережі і алгоритму мінімізації функціонала якості (Cost Function Minimization, CFM), що

визначає вхідний сигнал, необхідний для досягнення бажаної поведінки об'єкту. NGPC алгоритм складається з блоку CFM і блоку нейронної мережі.

Принцип роботи системи полягає в наступному. Вхідний сигнал $r(n)$ подається еталонній моделі. Ця модель видає еталонний сигнал $y_m(n)$, який служить входом для CFM блоку. CFM алгоритм розраховує сигнал, який служить входом для об'єкту або моделі об'єкту. Двополюсною двохпозиційний перемикач S встановлюється в положення до об'єкту, коли CFM алгоритм використовується для визначення оптимального вхідного сигналу $u(n)$, який мінімізує вибраний критерій якості управління. Між тактами перемикач встановлений в положення до моделі об'єкту, де CFM алгоритм використовує цю модель для розрахунку наступного входу, що управляє $u(n+1)$ шляхом прогнозу у відповідь сигналу, отриманого від моделі об'єкту. Як тільки функціонал якості мінімізований, цей вхідний сигнал подається на об'єкт.

Узагальнений алгоритм нейроуправління з прогнозом складається з наступних основних кроків.

- 1) Генерується задаюча траєкторія. Якщо майбутня траєкторія $y_m(n)$ невідома, вважається $y_m(n) = const$.
- 2) Використовуючи заздалегідь розрахований управляючий вхідний вектор і нейронну модель об'єкту управління, виконується прогноз поведінки об'єкту.
- 3) Розраховується новий управляючий вхідний сигнал, що мінімізує функціонал якості.
- 4) Повторюються кроки 2 і 3, поки не буде досягнута необхідна мінімізація.
- 5) Посилається перший управляючий вхідний сигнал на об'єкт.
- 6) Повторюється весь процес для кожного тимчасового кроку.

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient, Simplex і Successive Quadratic Programming. Вибір методу мінімізації

може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

Якість обчислення при реалізації узагальненого управління з прогнозом в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient [67], Simplex [68] і Successive Quadratic Programming [69, 70]. Вибір методу мінімізації може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу [69, 70]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

Як указувалося вище, узагальнений алгоритм нейроуправління з прогнозом ґрунтується на мінімізації функціонала якості на кінцевому діапазоні прогнозів. Функціонал якості, використовуваний в роботі, має наступний вид:

$$J = \sum_{j=N_1}^{N_2} [y_m(n+j) - y_n(n+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(n+j)]^2, \quad (\text{B.1})$$

де N_1 – нижня межа прогнозу;
 N_2 – верхня межа прогнозу;
 N_u – діапазон управління;
 y_m – бажана траєкторія;
 y_n – передбачений вихід нейронної мережі;
 λ – ваговий множник;
 $\Delta u(n + j)$ – зміна u визначається таким чином:

$$u(n + j) - u(n + j - 1).$$

Цей функціонал мінімізує не тільки середньоквадратичну помилку між еталонним сигналом і сигналом, що видається моделлю об'єкту, але також зважену середньоквадратичну швидкість зміни сигналу, що управляє.

Коли функціонал якості мінімізований, генерується вхідний управляючий сигнал, що дозволяє об'єкту відстежувати задану траєкторію з деякою точністю. Функціонал якості містить чотири параметри, що настраюються N_1 , N_2 , N_u , λ . Параметри N_1 і N_2 задають межі, усередині яких обчислюється помилка стеження. N_u є межею діапазону управління. Єдине обмеження, що накладається на значення N_u і N_1 наступне: вони повинні бути менше або рівні N_2 . Друга сума містить ваговий множник λ , який введений для управління балансом між першою і другою сумою.

Для реалізації оптимального управління, функціонал якості мінімізується за допомогою алгоритму Ньютона-Рафсона. Мета алгоритму – мінімізувати функціонал J (В.1) по вектору $[u(n + 1), u(n + 2), u(n + N_u)]^T$. Позначимо даний вектор як $\mathbf{U}$. Використовуючи як алгоритм CFM алгоритм Ньютона-Рафсона, ітераційно мінімізується функціонал J для визначення оптимального $\mathbf{U}$. Ітераційний процес визначає значення J на кожній ітерації, яке позначимо як $J(k)$. Для кожної ітерації розраховується також значення управляючого вектора

$$\mathbf{U}(k) = \begin{bmatrix} u(n+1) \\ u(n+2) \\ \vdots \\ u(n+N_u) \end{bmatrix},$$

де $k=1, \dots$ – номер ітерації.

Правило оновлення $\mathbf{U}(k+1)$ у алгоритмі Ньютона-Рафсона виглядає таким чином:

$$\mathbf{U}(k+1) = \mathbf{U}(k) - \left(\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) \right)^{-1} \frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k) \quad (\text{B.2})$$

де якобіан позначений, як

$$\frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k) = \begin{bmatrix} \frac{\partial J}{\partial u(n+1)} \\ \vdots \\ \frac{\partial J}{\partial u(n+N_u)} \end{bmatrix},$$

а гесіан, як

$$\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) = \begin{bmatrix} \frac{\partial^2 J}{\partial u(n+1)^2} & \cdots & \frac{\partial^2 J}{\partial u(n+1)\partial u(n+N_u)} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 J}{\partial u(n+N_u)\partial u(n+1)} & \cdots & \frac{\partial^2 J}{\partial u(n+N_u)^2} \end{bmatrix}.$$

Рішення рівняння (B.2) вимагає явного звернення матриці Гессе. Цей процес може зажадати великих обчислювальних витрат. Одним з підходів, використовуваних, щоб уникнути звернення матриці, служить LU декомпозиція [71] для отримання вхідного вектора $\mathbf{U}(k+1)$. Це досягається шляхом запису рівняння (8.2) у вигляді системи лінійних рівнянь $\mathbf{Ax} = \mathbf{b}$, що приводить до виразу

$$\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k)(\mathbf{U}(k+1) - \mathbf{U}(k)) = -\frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k), \quad (\text{B.3})$$

де

$$\frac{\partial^2 \mathbf{J}}{\partial \mathbf{U}^2}(k) = \mathbf{A} ; \quad -\frac{\partial \mathbf{J}}{\partial \mathbf{U}}(k) = \mathbf{b} ; \quad \mathbf{U}(k+1) - \mathbf{U}(k) = \mathbf{x}.$$

У даній формі рівняння (8.3) може бути вирішене за допомогою двох процедур, описаних в [71]: процедури LU декомпозиції, ludcmp і процедури рішення системи лінійних рівнянь lubksb. Після того, як обчислений вектор $\mathbf{x}$. Процедура повторюється до тих пір, поки зміна кожній компоненти $\mathbf{U}(k+1)$ не стає меншим деякого ε . При рішенні системи відносно $\mathbf{x}$ для кожної ітерації алгоритму Ньютона-Рафсона необхідне обчислення кожного елементу якобіана і гессиана. Елемент якобіана з індексом h має наступний вигляд:

$$\begin{aligned} \frac{\partial J}{\partial u(n+h)} = & -2 \sum_{j=N_1}^{N_2} [y_m(n+j) - y_n(n+j)] \frac{\partial y_n(n+j)}{\partial u(n+h)} + \\ & + 2 \sum_{j=1}^{N_u} \lambda(j) [\Delta u(n+j)] \frac{\partial \Delta u(n+j)}{\partial u(n+h)}, \quad h=1, \dots, N_u. \end{aligned}$$

Якщо розписати вираз $\frac{\partial \Delta u(n+j)}{\partial u(n+h)}$ і скористатися символом Кронекера

дельта функції, яка визначається як

$$\delta(h, j) = \begin{cases} 1 & \text{если } h = j, \\ 0 & \text{если } h \neq j, \end{cases}$$

отримаємо

$$\frac{\partial u(n+j)}{\partial u(n+h)} - \frac{\partial u(n+j-1)}{\partial u(n+h)} = \delta(h, j) - \delta(h, j-1).$$

Елемент з індексами m, h гессиана визначається таким чином:

$$\begin{aligned}
& \frac{\partial^2 J}{\partial u(n+m)\partial u(n+h)} = \\
& = 2 \sum_{j=N_1}^{N_2} \left\{ \frac{\partial y_n(n+j)}{\partial u(n+m)} \frac{\partial y_n(n+j)}{\partial u(n+h)} - \frac{\partial^2 y_n(n+j)}{\partial u(n+m)\partial u(n+h)} [y_m(n+j) - y_n(n+j)] \right\} + \\
& + 2 \sum_{j=1}^{N_u} \lambda(j) \left\{ \frac{\partial \Delta u(n+j)}{\partial u(n+m)} \frac{\partial \Delta u(n+j)}{\partial u(n+h)} + \Delta u(n+j) \frac{\partial^2 \Delta u(n+j)}{\partial u(n+m)\partial u(n+h)} \right\},
\end{aligned}$$

$$h = 1, \dots, N_u, \quad m = 1, \dots, N_u.$$

Можна скористатися також визначенням дельта-функції для виразу

$$\frac{\partial \Delta u(n+j)}{\partial u(n+m)} \frac{\partial \Delta u(n+j)}{\partial u(n+h)} = (\delta(h, j) - \delta(h, j-1))(\delta(m, j) - \delta(m, j-1)).$$

Вираз $\frac{\partial^2 \Delta u(n+j)}{\partial u(n+m)\partial u(n+h)}$ завжди дорівнює нулю.

Остання компоненту, потрібна для отримання $U(k+1)$, це обчислення виходу об'єкту $y_n(n+j)$ і його похідних. Для цього використовується нейронна мережа.

В.2 Структура нейронної мережі для моделювання об'єкту

Моделлю об'єкту в алгоритмі NGPC служить нейронна мережа. Початкове тренування нейронної мережі зазвичай виконується в автономному режимі (offline), до застосування управління. Блокова схема для тренування нейронної мережі, що моделює об'єкт, приведена на рис. В.2.

Нейронна мережа і об'єкт набувають одного і того ж вхідного значення $u(n)$. Нейронна мережа також має додатковий вхід, на який подається або вихідне значення об'єкту $y(n)$, або нейронній мережі $y_n(n)$. Вибір значення залежить від кроку прогнозу. В процесі тренування мережі її ваги настраюються та-

ким чином, що безліч входних значень продукує бажані вихідні значення. Помилка формується між відповідями нейронної мережі $y_n(n)$ і об'єкту $y(n)$. Надалі ця помилка використовується для оновлення вагів за допомогою одного з розглянутих вище методів, наприклад градієнтного спуску [72]. Цей процес повторюється до тих пір, поки помилка не досягне допустимого рівня.

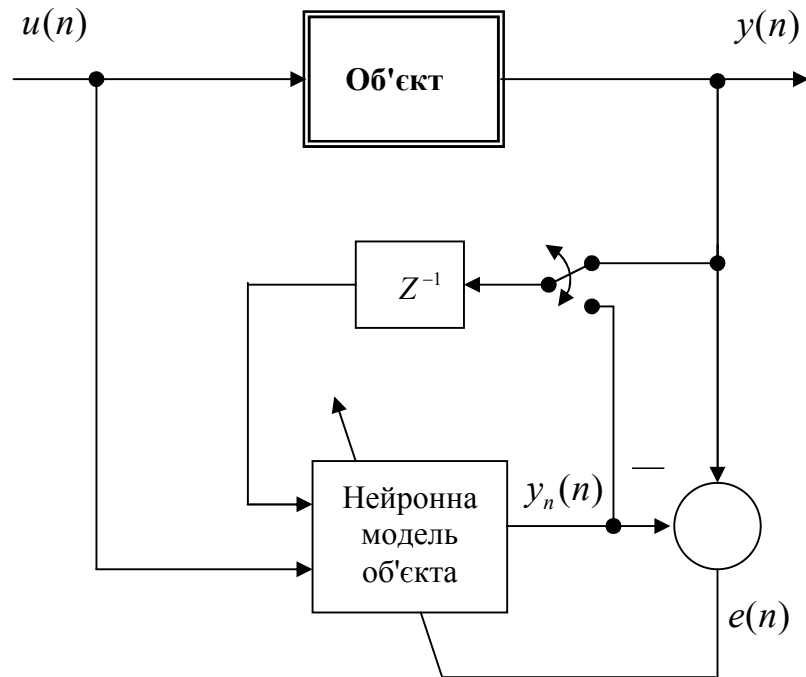


Рисунок В.2 – Схема тренування мережі в режимі offline

Оскільки для моделювання об'єкту буде використана нейронна мережа, повинна бути розглянута конфігурація архітектури нейронної мережі. Нейронна мережа може бути настроєна або для моделювання входу/виходу, або для моделювання простору станів.

У даній реалізації алгоритму NGPC використовуються моделі, що використовують тільки входні/вихідні координати, оскільки моделювання простору станів вимагає вимірювання станів об'єкту, що не завжди можливо. Згідно сказаному вище, мережі МП відрізняються універсальністю при моделюванні різних об'єктів регулювання. У зв'язку з цим подальші дослідження ведуться тільки з цим типом мережі.

МП в даний час – якнайповніше теоретично досліджена мережа. Як указувалося, існуючі теореми стверджують, що МП за умови достатньої кількості нейронів можуть відображати практично будь-які взаємозв'язки. Тому вони часто використовуються для побудови нейромережкових моделей нелінійних об'єктів. Проте перше, з чим стикаються, це питання про те, яку структуру повинна мати мережу. Знаходження оптимальної структури мережі – проблема, яка до цих пір повністю не вирішена. В цьому випадку на допомогу приходить цілеспрямоване моделювання і практичний досвід.

Визначення структури нейронної мережі полягає, по-перше, з визначення входів мережі і, по-друге, з визначення внутрішньої топології мережі (кількість шарів і нейронів). На рис. В.3 приведена багатошарова нейронна мережа прямого розповсюдження із структурою, що має вузли затримок за часом.

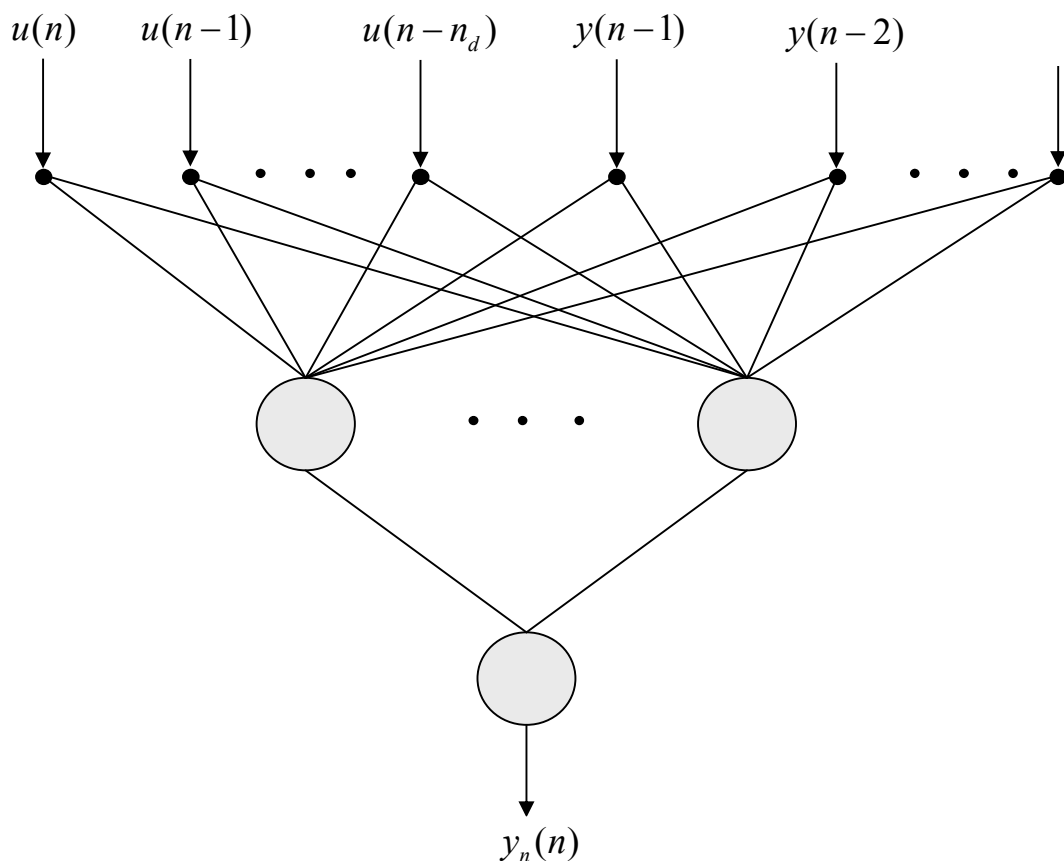


Рисунок В.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

На даному прикладі входи мережі складаються із зовнішніх входів $u(n)$ і $u(n-1)$ і відповідних ним вузлів затримок $u(n-1), \dots, u(n-n_d)$ і $u(n-2), \dots, u(n-d_d)$. Параметри n_d d_d відображають число вузлів затримки, що відносяться до відповідних вузлів входу. Другий вхід мережі може мати зовнішній вхід $y_n(n)$ і відповідні затримані значення. Мережу має один прихований шар, що складається з декількох прихованих нейронів, які використовують загальну функцію висновку (активації) $f_j(\cdot)$. Вихідний нейрон використовує лінійну функцію висновку з одиничним нахилом для масштабування вихідних значень.

Рівняння мережі з даною архітектурою виглядають таким чином:

$$y_n(n) = \sum_{j=1}^{S^{L-1}} w_j f_j(net_j(n)) + b \quad (B.4)$$

і

$$n_j(l) = \sum_{i=0}^{n_d} w_{j,i+1} u(n-i) + \sum_{i=1}^{d_d} w_{j,n_d+i+1} y(l-i) + b_j, \quad (B.5)$$

де $y_n(n)$ – вихід нейронної мережі;

$f_j(\cdot)$ – функція активації для j -го нейрона прихованого шару;

$net_j(n)$ – аргумент функції активізації j -го нейрона;

S^{L-1} – число нейронів в прихованому шарі;

L – число шарів;

n_d – число вузлів входу, що асоціюються з $u(\cdot)$ без урахування $u(n)$;

d_d – число вузлів входу, що асоціюються з $y(\cdot)$;

w_j – вага, що сполучає j -й прихований нейрон з нейроном виходу;

$w_{j,i}$ – вага, що сполучає i -й вузол входу з j -м прихованим нейроном;

$y(n-i)$ – затриманий вихід об'єкту, використовуваний як вхід для мережі;

$u(n-i)$ – вхід мережі і його затримки;

b_j – зсув j -го прихованого нейрона;

b – зсув нейрона виходу.

В.3 Прогноз з використанням нейронної мережі

NGPC алгоритм використовує вихід моделі об'єкту для прогнозу динаміки об'єкту для довільного входу від теперішнього моменту часу n до деякого майбутнього моменту $n + k$. Це досягається зрушенням рівнянь (В.4) і (В.5) на k маємо

$$y_n(n+k) = \sum_{j=1}^{S^{L-1}} \{w_j f_j(\text{net}_j(n+k))\} + b \quad (\text{В.6})$$

і

$$n_j(n+k) = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} u(n+k-i), & k - N_u < i \\ u(n+N_u), & k - N_u \geq i \end{cases} + \\ + \sum_{i=1}^{\min(k, d_d)} (w_{j, n_d+i+1} y_n(n+k-i)) + \sum_{i=k+1}^{d_d} w_{j, n_d+i+1} y(n+k-i) + b_j. \quad (\text{В.7})$$

Особливості рівняння (В.7) пов'язані з рекурсивним характером прогнозу. Перша сума в (В.7) визначається відповідно до приведених умов. Умова $k - N_u < i$ управляє попередніми майбутніми значеннями u до $u(n + N_u - 1)$. Умова $k - N_u \geq i$ встановлює входи від $u(n + N_u)$ до $u(n + k)$ рівними $u(n + N_u)$. Друга умова виникає тільки, якщо $N_2 > N_u$. Наступне підсумовування в (В.7) управляє рекурсивною частиною прогнозу. Підсумовування відповідає зворотному зв'язку по виходу нейронної мережі y_n і виконується k або d_d раз залежно від того, яке значення менше. Остання сума в (В.7) містить попередні значення y . У наступному підрозділі виводяться похідні від рівнянь (В.6) і (В.7) для входу $u(n + h)$.

В.4 Похідні рівнянь нейронної мережі

Для мінімізації функціонала якості в режимі реального часу за допомогою ітераційного рішення, заснованого на градієнті, необхідно використовувати формули для ефективного обчислення градієнта нейронної мережі.

Для обчислення якобіана і гесиана необхідна перша і друга похідні мережі по вхідному вектору. Елементи якобіана виходять диференціюванням $y_n(n+k)$ з рівняння (В.6) по $u(n+h)$

$$\frac{\partial y_n(n+k)}{\partial u(n+h)} = \sum_{j=1}^{S^{L-1}} w_j \frac{\partial f_j(net_j(n+k))}{\partial u(n+h)}. \quad (\text{В.8})$$

Застосовуючи правило диференціювання складної функції для $\frac{\partial f_j(net_j(n+k))}{\partial u(n+h)}$ маємо

$$\frac{\partial f_j(net_j(n+k))}{\partial u(n+h)} = \frac{\partial f_j(net_j(n+k))}{\partial net_j(n+k)} \frac{\partial net_j(n+k)}{\partial u(n+h)} \quad (\text{В.9})$$

де $\frac{\partial f_j(net_j(n+k))}{\partial net_j(n+k)}$ – похідна функції виходу і

$$\frac{\partial net_j(n+k)}{\partial u(n+h)} = \sum_{i=0}^{n_d} w_{j,i+1} \begin{cases} \delta(k-i, h), & k - N_u < i \\ \delta(N_u, h), & k - N_u \geq i \end{cases} + \quad (\text{В.10})$$

$$+ \sum_{i=1}^{\min(k, d_d)} w_{j, n_d+i+1} \frac{\partial y_n(n+k-i)}{\partial u(n+h)} \delta_1(k-i-1).$$

Відмітимо, що в (В.10) в останній сумі була введена ступінчаста функція δ_1 . Це було зроблено, щоб підкреслити, що сума рівна нулю для $k-i < 1$, отже доданки, що задовольняють цій умові, не вимагають обчислень. Щоб отримати елементи гесиана, необхідно продиференціювати рівняння (В.8), (В.9) і (В.10) по $u(n+t)$, в результаті маємо

$$\frac{\partial^2 y_n(n+k)}{\partial u(n+h)\partial u(n+m)} = \sum_{j=1}^{S^{L-1}} w_j \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} \quad (\text{B.11})$$

$$\begin{aligned} \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} &= \frac{\partial f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)} \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial u(n+h)\partial u(n+m)} + \\ &+ \frac{\partial^2 f_j(\text{net}_j(n+k))}{\partial \text{net}_j(n+k)^2} \frac{\partial \text{net}_j(n+k)}{\partial u(n+h)} \frac{\partial \text{net}_j(n+k)}{\partial u(n+m)}, \end{aligned} \quad (\text{B.12})$$

i

$$\frac{\partial^2 \text{net}_j(n+k)}{\partial u(n+h)\partial u(n+m)} = \sum_{i=1}^{\min(k, d_d)} w_{j, n_d+i+1} \frac{\partial^2 y_n(n+k-i)}{\partial u(n+h)\partial u(n+m)} \delta_1(k-i-1).$$

Рівняння (B.12) є результатом застосування правила диференціювання складної функції двічі.

Слід відмітити, що нелінійні оптимізації є процесами, що вимагають великих обчислювальних витрат. Алгоритм Ньютона-Рафсона використовується для створення ефективного обчислювального процесу. При написанні коду, що реалізовує даний алгоритм, всі значення, пораховані в одній процедурі, передаються іншим процедурам для уникнення повторення розрахунку. Найбільш важлива частина коду – обчислення гесиана. Змінні $\frac{\partial \text{net}_j(n+k)}{\partial u(n+h)}$ і $\frac{\partial y_n(n+k)}{\partial u(n+h)}$ обчислювані для якобіана, також використовуються для обчислення гесиана, тому вони передаються в процедуру обчислення гесиана. Оскільки гесіан симетричний, тільки верхньотрикутна частина матриці підлягає визначенню. Використання даних двох зауважень дозволяє значно понизити час завантаження процесора.

Таким чином, узагальнене нейроуправління з прогнозом з використання оптимізаційного алгоритму Ньютона-Рафсона для мінімізації функціонала якості дозволяє отримати високу якість управління у разі складних нелінійностей в об'єкті і дає можливість застосування алгоритму в режимі реального часу.

ДОДАТОК Г

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;
```

```
% Система управління електроприводом обертання  
% роторного колеса екскаватора  
% 2025 рік
```

```
% Вихідні дані
```

```
kzv=1.3;  
ks=7.82;  
Tg=1.002;  
Tzc=0.01;  
Te=0.034;  
Re=0.021;  
CF=10.5;  
In=2535;  
Mn=CF*In;  
Jd=775;  
Jm=233;  
b12=0;  
c12=300;  
Js=Jd+Jm;  
Uz=158.6;  
ksx=ks/(CF+ks*kzv);
```

```
% Матриці
```

```
A=[-b12/Jm 1/Jm b12/Jm 0 0;  
-c12 0 c12 0 0;  
b12/Jd -1/Jd -b12/Jd CF/Jd 0;  
0 0 -CF/(Te*Re) -1/Te 1/(Te*Re) ;  
0 0 (CF*ks*Tzc/(Te*Re*Tg)-kzv*ks/Tg) ks*Tzc/(Te*Tg) (-  
ks*Tzc/(Te*Re*Tg)-1/Tg)];  
B=[0 -1/Jm;  
zeros(3,2);  
ks/Tg 0];  
C=zeros(5,5);  
C(1,1)=1;C(2,2)=1; C(3,3)=1;C(4,4)=1;C(5,5)=1;  
D=zeros(5,2);
```