

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

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

_____/ Станіслав ДМИТРИЄВ
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

Харків – 2025

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

ЗАВДАННЯ

на кваліфікаційну роботу магістерського рівня вищої освіти

студенту (ці) Станіслав ДМИТРИЄВ
(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 повороту стріли 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 $F_b=12\cdot10^4\text{ Н}$; 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 повороту стріли роторного екскаватора. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити статичні і динамічні характеристики системи. Розробити математичну 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. Поворот стріли екскаватора. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейронні мережі. Ідентифікація об'єктів. 6. Системи нейрорегулювання. Структурні схеми. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. Магістерська кваліфікаційна робота. Висновки.

6. Консультант:

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

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

Керівник

(підпис)

Геннадій КАНЮК.
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Станіслав ДМИТРИЄВ
(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 повороту стріли 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 ЕОМ.	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)

Станіслав ДМИТРИЄВ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 200 сторінки, 70 рисунків, 8 таблиць, 88 використаних джерел, 5 додатків.

Головною метою дослідження є створення та обґрунтування методів синтезу системи керування електроприводом поворотного механізму роторного екскаватора. Основний акцент зроблено на покращенні якісних показників функціонування системи шляхом застосування сучасних нейромережевих регуляторів.

У роботі використано комплексний підхід. Спочатку, на основі фундаментальних принципів теорії автоматичного керування, проведено визначення параметрів системи. Далі застосовано методи математичного моделювання для розроблення повномасштабної математичної моделі багатомасової електромеханічної системи. Окремий етап присвячено синтезу нейрорегулятора, який побудовано за допомогою методів теорії штучних нейронних мереж для керування двомасовою системою електропривода.

Подальше дослідження здійснювалося шляхом комп'ютерного моделювання у середовищі MATLAB, що дало змогу оцінити динамічні властивості розробленої нейромережевої системи та порівняти їх із традиційними підходами.

У процесі виконання магістерської роботи отримано наукове підтвердження доцільності впровадження нейромережевих алгоритмів у системи керування багатомасовими електромеханічними механізмами, зокрема механізмом повороту стріли роторного екскаватора. Розроблено нову систему керування координатами електропривода, визначено параметри системи автоматичного регулювання та проведено їх комп'ютерну перевірку.

Створена математична модель враховує наявність пружних елементів у механічній частині привода, що дає можливість точніше прогнозувати поведінку двомасової системи та оцінювати її реакцію на зовнішні та внутрішні впливи.

ви. На основі цієї моделі визначено динамічні параметри системи та виконано їх подальший аналіз.

Особлива увага приділена побудові структури нейромережевої системи керування. Синтезовано нейронний регулятор типу NN Predictive Controller у середовищі MATLAB, після чого проведено всебічне моделювання його роботи. За отриманими результатами встановлено, що запропонована нейромережева система відзначається підвищеною точністю, стабільністю та швидкістю порівняно з традиційними схемами керування.

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

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

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

РЕФЕРАТ

An explanatory message consists of: 200 pages, 70 figures, 8 tables, 88 used sources, 5 additions.

The main goal of the research is to create and substantiate methods for synthesizing the control system for the electric drive of the rotary mechanism of a rotary excavator. The main emphasis is on improving the qualitative indicators of the system's functioning by using modern neural network controllers.

The work uses a comprehensive approach. First, based on the fundamental principles of the theory of automatic control, the system parameters are determined. Then, mathematical modeling methods are used to develop a full-scale mathematical model of a multi-mass electromechanical system. A separate stage is devoted to the synthesis of a neuroregulator, which is built using the methods of the theory of artificial neural networks to control a two-mass electric drive system.

Further research was carried out by computer modeling in the MATLAB environment, which made it possible to evaluate the dynamic properties of the developed neural network system and compare them with traditional approaches.

In the process of completing the master's thesis, scientific confirmation of the feasibility of implementing neural network algorithms in control systems for multi-mass electromechanical mechanisms, in particular, the mechanism for turning the boom of a rotary excavator, was obtained.

A new system for controlling the coordinates of the electric drive was developed, the parameters of the automatic control system were determined and their computer verification was carried out. The created mathematical model takes into account the presence of elastic elements in the mechanical part of the drive, which makes it possible to more accurately predict the behavior of the two-mass system and evaluate its reaction to external and internal influences. Based on this model, the dynamic parameters of the system were determined and their further analysis was performed.

Special attention was paid to building the structure of the neural network con-

trol system. A neural regulator of the NN Predictive Controller type was synthesized in the MATLAB environment, after which a comprehensive modeling of its operation was carried out. According to the results obtained, it was established that the proposed neural network system is characterized by increased accuracy, stability and speed compared to traditional control schemes.

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

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

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

P_n – потужність, що відповідає номінальному режиму роботи;

U_n – напруга, встановлена як номінальна;

I_n – струм у номінальних умовах;

M_n – номінальний момент;

n_n – швидкість обертання при номінальному навантаженні;

Φ_n – магнітний потік у номінальному режимі;

R – активний опір елемента;

L – величина індуктивності;

J – момент інерції вузла;

k_d – коефіцієнт підсилення електричного двигуна;

T_e – електромагнітна стала часу електропривода;

T_m – електромеханічна стала часу електропривода;

$W(p)$ – передавальна функція;

c – коефіцієнт жорсткості;

β – коефіцієнт внутрішнього в'язкого тертя;

A – матриця, що описує стан системи;

B – матриця, яка визначає вплив керуючих дій;

C – матриця виходу;

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

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

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

САУ – система автоматичного управління;

М – електродвигун.

ЗМІСТ

ВСТУП	8	
РОЗДІЛ 1. ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ		
УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	14	
1.1 Етапи становлення та розвитку теорії штучних нейронних мереж.....	14	
1.2 Аналіз підходів до побудови систем нейрорегулювання та обґрунтування вибору структури.....	17	
1.3 Загальні відомості про роторні екскаватори	27	
1.4 Аналіз факторів, що впливають на вибір системи управління електроприводом механізму повороту стріли роторного екскаватора	31	
1.5 Постановка задачі дослідження.....	34	
ВИСНОВКИ ДО РОЗДІЛУ 1	36	
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПОВОРОТУ СТРИЛИ РОТОРНОГО ЕКСКАВАТОРА		38
2.1 Основні технічні характеристики роторного екскаватора та вимоги до си- стеми управління.....	38	
2.2 Розрахунок потужності і вибір привідних електродвигунів	40	
2.3 Розрахунок швидкісної діаграми механізму повороту стріли екскаватора	45	
2.4 Побудова діаграми навантаження та перевірка двигунів за нагрівом і перевантажувальною здатністю. Вибір основного електроустаткування	47	
2.5 Аналіз статичних характеристик системи.	57	
2.6 Аналіз динамічних характеристик системи	67	
ВИСНОВКИ ДО РОЗДІЛУ 2	85	
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЛАНОК.....		86
3.1 Математична модель двомасової системи	86	
3.2 Аналіз динамічних характеристик двомасової моделі.....	93	

ВИСНОВКИ ДО РОЗДІЛУ 3	100
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	101
4.1 Регулятори, реалізовані засобами пакета Neural Network Toolbox у середовищі MATLAB	101
4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller	103
4.3 Проєктування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB	106
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	151
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	153
ВИСНОВКИ ДО РОЗДІЛУ 4	159
ВИСНОВКИ	161
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	163
ДОДАТОК А. СТРУКТУРА ШТУЧНОГО НЕЙРОНА	171
ДОДАТОК Б. АРХІТЕКТУРА ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ.....	178
Б.1 Архітектури найбільш поширених ШНМ.....	178
Б.2 Штучні нейронні мережі прямого розповсюдження	180
ДОДАТОК В. НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ	184
В.1 Загальні питання навчання нейронних мереж.....	184
В.2 Явище перенавчання	186
В.3 Властивість узагальнення.....	188
В.4 Метод зворотного розповсюдження помилки.....	190
ДОДАТОК Г. ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ	195
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	200

ВСТУП

Актуальність теми. Підвищення ефективності виробничих процесів, зростання продуктивності праці та покращення якості промислової продукції безпосередньо пов'язані з удосконаленням автоматизованих систем електроприводу, зокрема тих, що базуються на сучасних інженерних та наукових досягненнях. Протягом останніх десятиліть сформувався значний спектр методів керування, серед яких особливе місце займають підходи адаптивного типу. Їх поширення стало можливим завдяки стрімкому розвитку теорії керування, методів аналізу стійкості та сучасних обчислювальних технологій.

Адаптивні системи автоматичного регулювання здатні коригувати свою поведінку у відповідь на зміни характеристик об'єкта керування та зовнішнього середовища, що робить їх ефективним інструментом для промислових застосувань. Проте якість функціонування таких систем значною мірою визначається точністю математичних моделей, які повинні одночасно відображати реальні властивості об'єкта та залишатися зручними для реалізації алгоритмів керування. Нестача інформації щодо умов роботи системи, параметрів об'єктів чи зовнішніх впливів потребує застосування адаптивних методів і допускає використання спрощених – зокрема, лінійних – моделей. Незважаючи на їх зручність, лінійні моделі нерідко обмежують можливість отримання бажаної якості регулювання. Тому все більшої актуальності набуває комбінування адаптивних підходів із технологіями штучних нейронних мереж.

Штучні нейронні мережі, як один із напрямів розвитку штучного інтелекту, функціонують за принципами, що частково відтворюють роботу біологічного мозку [1–12]. Їх ключовими перевагами є здатність до паралельної обробки сигналів, висока швидкодія та можливість функціонування в режимі реального часу. Окрім цього, нейронні мережі мають властивість відмовостійкості: у разі виходу з ладу частини структурних елементів інші зв'язки здатні компенсувати їх функцію.

Не менш важливою характеристикою є здатність нейронної мережі навчатися та узагальнювати набуті знання. Навіть якщо навчання проводиться на

відносно невеликій множині прикладів, мережа може демонструвати адекватні результати на даних, яких вона не «бачила» під час тренування. Технології високого ступеня інтеграції нині дозволяють реалізовувати нейромережеві структури у вигляді спеціалізованих обчислювальних модулів, що суттєво розширює сфери їх практичного застосування – від обробки аудіо- та відеосигналів до побудови складних систем керування.

У системах регулювання нейронні мережі використовуються як інструменти ідентифікації для оцінювання стану нелінійних систем, як нелінійні фільтри на зразок розширеного фільтра Калмана, а також як засоби автоматичного налаштування параметрів класичних регуляторів. Для позначення таких систем за участю нейромережевих моделей широко використовується термін *нейромережеві системи управління*. Особливо широке застосування у практиці отримали багатопарові мережі прямого поширення сигналу, зокрема багатопарові перцептрони.

Гірничодобувна галузь і сьогодні залишається стратегічно важливою складовою економіки країни. Розвиток відкритих гірничих робіт супроводжується модернізацією технічного парку підприємств та впровадженням високопродуктивного обладнання. Роторні екскаватори є ключовою ланкою технологічних комплексів такого типу, а тому вдосконалення систем керування їх електроприводами – актуальне і важливе науково-практичне завдання. Підвищення точності та швидкодії системи керування механізмом повороту стріли безпосередньо впливає на загальну продуктивність і надійність комплексу.

Мета та завдання дослідження. Метою магістерської роботи є розробка методів синтезу системи керування механізмом повороту стріли роторного екскаватора на основі нейромережевих регуляторів з метою підвищення якості роботи системи.

Для досягнення поставленої мети необхідно розв'язати такі основні завдання:

1. Провести огляд існуючих систем автоматичного керування та проаналізувати можливість застосування нейромережевих методів у задачах регулювання.

2. Побудувати розширену математичну модель динаміки об'єкта керування.
3. Розробити структурну схему системи із застосуванням нейромережевого регулятора.
4. Виконати синтез системи керування на основі нейрорегулятора з прогнозуванням і визначити параметри, що забезпечують найвищу якість регулювання.
5. Реалізувати комп'ютерне моделювання функціонування нейромережевої системи та виконати аналіз її характеристик.

Об'єктом дослідження є динамічні властивості системи керування механізмом повороту стріли роторного екскаватора з урахуванням пружності елементів приводу.

Предметом дослідження виступають моделі та методи синтезу нейромережевих систем керування двомасовими електромеханічними системами з податливими механічними зв'язками.

Методи дослідження. Теоретичні результати роботи ґрунтуються на положеннях теорії автоматичного керування. Для побудови математичної моделі застосовано методи математичного моделювання, для синтезу нейромережевих моделей – принципи теорії штучних нейронних мереж, а для побудови системи керування – методи нейроуправління. Ефективність отриманих рішень підтверджено шляхом імітаційного моделювання за допомогою сучасних програмних засобів.

Наукова новизна отриманих результатів.

1. Вперше здійснено наукове обґрунтування ефективності використання нейромережевих технологій для керування багатомасовою електромеханічною системою механізму повороту стріли роторного екскаватора.
2. Вперше запропоновано та синтезовано нейромережеву систему керування механізмом повороту стріли з урахуванням пружних властивостей передавальних елементів, яка характеризується високою якістю регулювання.
3. Запропоновано нову математичну модель системи у вигляді багато-

шарової мережі прямого поширення сигналу, що дає змогу адекватно враховувати пружні зв'язки у механізмі.

4. Отримано подальший розвиток моделей електромеханічних систем із складними кінематичними зв'язками шляхом подання їх у вигляді двомасових структур.

5. Розвинуто методи синтезу нейромережевих регуляторів із прогнозуванням для керування об'єктами з вираженою нелінійною динамікою.

Практичне значення результатів. Розроблено нейромережеву систему керування двомасовою електромеханічною системою механізму повороту стріли роторного екскаватора, що забезпечує високу точність регулювання та стійкість до зовнішніх впливів. Матеріали дослідження використовуються в навчальному процесі кафедри АМЕТ ННІ «УПА» ХНУ ім. В.Н. Каразіна при виконанні курсових проєктів, проведенні лекційних та практичних занять.

Особистий внесок здобувача. Усі основні наукові результати, представлені у магістерській кваліфікаційній роботі, отримано автором самостійно. Здобувачем синтезовано систему регулювання координат електроприводу, побудовано математичну модель з урахуванням пружних елементів, розроблено структурну схему нейромережевої системи та виконано синтез регулятора NN Predictive Controller із використанням можливостей MATLAB Neural Network Toolbox. Також виконано комплексне моделювання системи та проведено детальний аналіз отриманих характеристик.

Апробація результатів. Основні положення та висновки дослідження були представлені та обговорені на науковій конференції студентів ННІ «УПА» ХНУ ім. В.Н. Каразіна (Харків, 2025 рік).

Структура та обсяг роботи. Магістерська робота складається зі вступу, чотирьох розділів, висновків, додатків та списку використаних джерел. Загальний обсяг становить 200 сторінок. Робота містить 70 рисунків, з яких 48 наведено в основному тексті, а 22 – на 20 окремих сторінках; 8 таблиць на 5 окремих сторінках, 5 додатків обсягом 30 сторінок та список літературних джерел із 88 позицій, розміщений на 8 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ УПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Етапи становлення та розвитку теорії штучних нейронних мереж

Штучні нейронні мережі створюються за аналогією з принципами роботи біологічних нейронних структур і призначені для розв’язання широкого спектра завдань – від розпізнавання образів і класифікації до прогнозування, оптимізації та керування складними системами. Завдяки їм значною мірою зросла обчислювальна ефективність сучасних комп’ютерних систем, зокрема тих, що належать до класу нейрокомп’ютерів, основу функціонування яких становлять саме ШНМ.

Поняття «*нейронна мережа*» почало формуватися ще до 1950-х років. Подальший розвиток теорії був пов’язаний із роботами визнаних дослідників – У. Маккалоха, Д. Хебба, Ф. Розенблатта, М. Мінського та Дж. Хопфілда. У 1943 р. У. Маккаллох і У. Пітс [13] представили першу формалізовану модель штучного нейрона та заклали підґрунтя для математичного опису роботи мозку. Через кілька років, у 1949 р., Д. Хебб [14] описав механізми взаємодії нейронів і принципи формування зв’язків між ними, запропонувавши перші правила навчання нейронних структур.

Суттєвий прорив стався у 1957 р., коли Ф. Розенблатт [15,16] увів поняття *персептрона* та продемонстрував його практичну реалізацію, створивши перший нейрокомп’ютер Mark. У 1959 р. Д. Хьюбел і Т. Візел [17] експериментально довели, що в біологічних нейросистемах інформація зберігається та опрацьовується паралельно, що підтвердило перспективність подальшого розвитку штучних мереж.

У 1960–1968 рр. активно вивчалися різні типи штучних мереж: у цей час з’явилися моделі адаліна та мадаліна, створені В. Відроу в 1960–1962 рр. [18,19], а також асоціативні матриці К. Штайнбуха (1963 р.) [20]. Проте у

1969 р. після виходу книги М. Мінського та С. Пейперта *«Перцептрони»* [21], де обґрунтовувалася обмеженість можливостей однорівневих перцептронів, інтерес до штучних нейронних мереж суттєво знизився.

Відродження уваги до цієї галузі припало на кінець 1970-х років. Воно стало можливим завдяки новим науковим даним про роботу мозку, а також стрімкому розвитку мікроелектроніки та обчислювальної техніки. У 1982–1985 рр. Дж. Хопфілд [22,23] розробив новий клас нейронних мереж, орієнтованих на розв’язання оптимізаційних задач та моделювання асоціативної пам’яті. В цей період почали з’являтися перші комерційні нейрокомп’ютери (зокрема, Mark III компанії TRW у 1985 р.).

З 1987 р. у США, Японії та країнах Західної Європи розпочалися масштабні інвестиційні програми, спрямовані на розвиток ШНМ та нейрокомп’ютерних технологій (наприклад, японська програма «NITAP Frontiers» та європейська «Basic Research in Adaptive Intelligence and Neurocomputing»). Уже наприкінці 1980-х років штучні нейронні мережі досліджувалися практично всіма провідними компаніями електротехнічної галузі. Нейрокомп’ютери швидко перетворилися на одну з найбільш динамічних сфер технологічного ринку, а у США агентство DARPA ініціювало масштабні розробки високошвидкісних нейрокомп’ютерів для різних галузей застосування.

Починаючи з 1990-х років минулого сторіччя теорія штучних нейронних мереж (ШНМ) зазнала стрімкого розвитку, що визначив сучасний напрямок інтелектуальних систем керування та обробки інформації. Цей період характеризується переходом від простих архітектур типу перцептронів до багатошарових мереж з алгоритмами зворотного розповсюдження помилки (backpropagation), що дозволило ефективно навчати моделі з великою кількістю параметрів. Основоположні праці [24].Румельхарта, Гінтона та Вільямса (Rumelhart, Hinton & Williams, 1986) започаткували практичне використання методу градієнтного спуску для багатошарових перцептронів, що стало базою для подальших досліджень у 1990-х роках.

У цей час з'являються нові архітектури, орієнтовані на різні типи даних і завдань: радіально-базисні мережі (RBF), самоорганізувальні карти Кохонена (SOM), рекурентні нейронні мережі (RNN), а також мережі Хопфілда для оптимізаційних задач ([25, 26].Kohonen, 1990; Hopfield, 1991). Їхня поява стимулювала розвиток теоретичних досліджень у напрямі стійкості, збіжності та узагальнюючої здатності нейронних структур.

Починаючи з 2000-х років, основну увагу приділено глибоким нейронним мережам (Deep Neural Networks, DNN), що складаються з великої кількості прихованих шарів. Прорив у цій галузі пов'язують з роботами Лекуна, Хінтона та Бенджіо ([27].LeCun, Bengio & Hinton, 2015), де показано ефективність глибоких згорткових мереж (CNN) для розпізнавання зображень і рекурентних мереж (LSTM, GRU) для обробки часових рядів. Водночас зросла роль обчислювальної потужності (GPU) і великих наборів даних, що зробило можливим практичне навчання таких моделей.

У 2010–2020-х роках теорія ШНМ активно розвивається в напрямку обґрунтування властивостей узагальнення, стабільності, інтерпретованості та енергетичної ефективності. З'являються роботи з нейронно-символьних підходів, нейромережових керуючих систем та глибокого підкріплювального навчання (Deep Reinforcement Learning), що поєднує нейронні мережі з класичною теорією керування ([28, 29].Goodfellow, Bengio & Courville, 2016; Silver et al., 2017).

Таким чином, за три десятиліття - від 1990-х до 2020-х років - теорія штучних нейронних мереж еволюціонувала від відносно простих моделей до потужних гібридних систем, що поєднують машинне навчання, оптимізацію та методи керування складними нелінійними динамічними об'єктами.

На межі нового тисячоліття розвиток технологій перейшов до рівня субмікронних і наномасштабів. Паралельно стрімкий прогрес у галузях молекулярної та біомолекулярної інженерії відкрив можливість створення принципово нових типів нейрокомп'ютерів, які базуються на нетрадиційних архітектурах і технологічних підходах.

Комплексне вивчення штучних нейронних мереж потребує широкої між-дисциплінарної підготовки. До переліку галузей, знання з яких є необхідними, входять нейрофізіологія, когнітивні науки, психологія, фізика (особливо статистична механіка), теорія систем керування, обчислювальні науки, штучний інтелект, статистичний аналіз, розпізнавання образів, комп'ютерне бачення, теорія паралельних обчислень та апаратна інженерія як аналогових, так і цифрових пристроїв. Водночас розвиток ШНМ сам стимулює просування цих дисциплін, формуючи нові концепції та інструменти. Така взаємодія є основою продуктивного дослідження нейромережових технологій.

У додатках А–Г наведено ключові поняття, що становлять теоретичне підґрунтя для виконаних у роботі досліджень.

У межах цієї роботи для побудови нейромережової системи обрано багат шарові прямопровідні мережі, у яких взаємозв'язки між нейронами спрямовані лише у прямому напрямку, без внутрішніх зворотних зв'язків. Фактично йдеться про багат шаровий персептрон. Додаток А містить опис структури штучного нейрона, а в додатку Б подано огляд основних архітектур нейронних мереж.

Методи навчання нейронних мереж, а також обґрунтування вибору методу, використаного в дослідженні, викладено в додатку В.

Одним з ключових етапів розв'язання задач керування, прогнозування та моделювання нелінійної динаміки є побудова адекватної математичної моделі об'єкта. Особливості ідентифікації таких систем із застосуванням нейромереж розглянуто в додатку Г.

1.2 Аналіз підходів до побудови систем нейрорегулювання та обґрунтування вибору структури

У сучасній літературі, присвяченій нейромережовим методам керування, терміни «*пряме*» та «*непряме*» регулювання зустрічаються досить часто, але використовуються в різних значеннях. Тому доцільно одразу визначити спосіб

їх трактування, який надалі буде застосовуватися послідовно. Загальноприйнято розділяти методи керування за ознакою того, як саме коригуються параметри регулятора.

Пряма адаптація передбачає безпосереднє налаштування вагових коефіцієнтів регулятора відповідно до заданого критерію якості.

Непряма адаптація, навпаки, базується на попередній ідентифікації моделі об'єкта, параметри якої потім використовуються для синтезу регулятора. Саме таке трактування обрано для огляду методів нейромережевого регулювання в цьому розділі.

Ще один поширений поділ стосується способу навчання: мережу можуть навчати *offline*, тобто до увімкнення системи керування, або *online*, коли процес регулювання та навчання відбуваються одночасно. У режимі реального часу (*real-time*) усі операції – збирання, обробка та видача сигналів – виконуються в межах періоду дискретності, який визначається властивостями об'єкта, а не обчислювальними ресурсами. *Online*-навчання відповідає ситуації, коли мережа проходить хоча б одну епоху навчання в тому ж циклі, в якому формує керуючий сигнал.

Послідовність операцій у системі з *online*-навчанням зображено на рис. 1.1 [30], де позначено регульовану координату y , керуючу дію u , період дискретності T , попередню фільтрацію даних датчиків t_1 , процес тренування t_2 , обчислення керуючої дії t_3 та кількість попередніх вимірювань m, n , що залучаються в моделі.

Кількість запропонованих архітектур нейрорегулювання сьогодні надзвичайно велика. Деякі підходи поєднують аналітичний закон регулювання з нейромережею-моделлю, яка відновлює певні елементи матриць динамічної моделі – такі методи належать до структурованих. Інші повністю покладаються на апроксимаційні можливості ШНМ і не вимагають явної математичної структури – це неструктуровані моделі. У наступних підрозділах узагальнено найбільш актуальні архітектури.

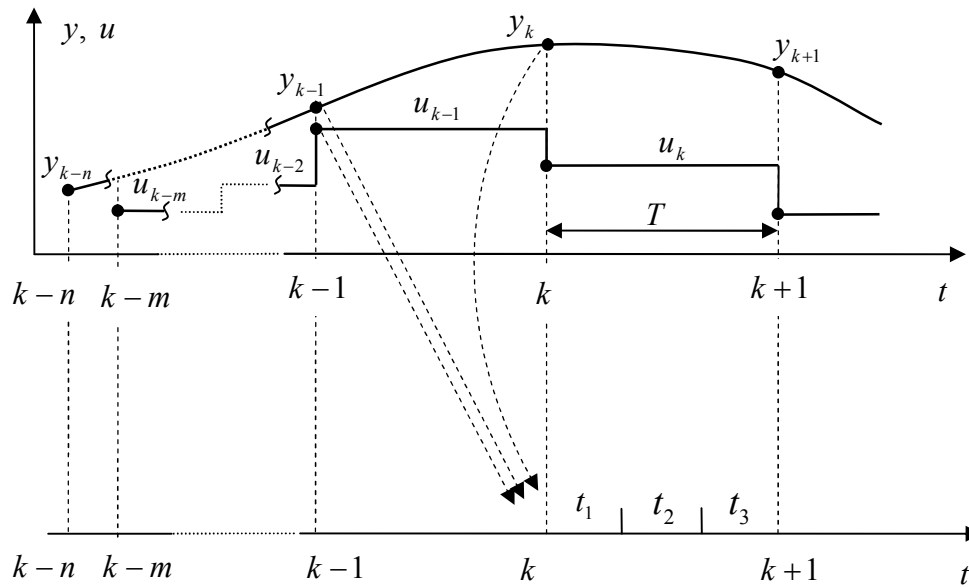


Рисунок. 1.1 – Регулювання і тренування мережі в реальному часі

1.2.1 Пряме інверсне регулювання.

Термін *Direct inverse control* був запропонований Вербосом [31] і є одним із найбільш поширених підходів регулювання (див. рис. 1.2). Навчання нейронних мереж у цій методиці може виконуватися як *offline*, так і *online*, що у класифікації Псалтиса та співавторів [32] відповідає методам *general learning* та *specialized learning*.

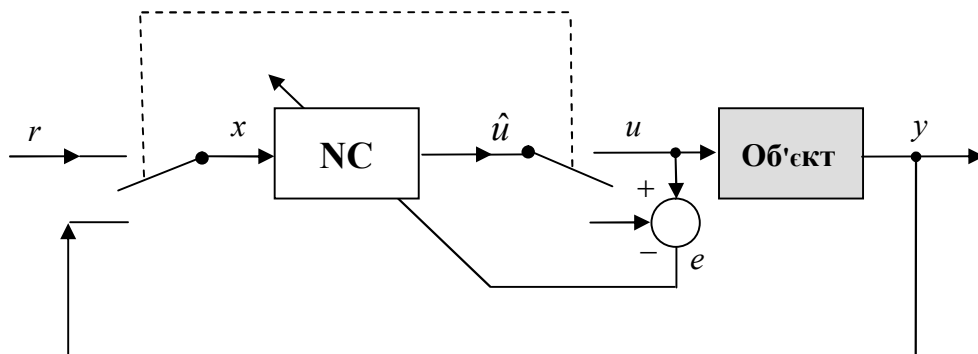


Рисунок 1.2 – Пряме інверсне регулювання з offline тренуванням

Offline-навчання. У *offline*-варіанті передбачається існування інверсного відображення між вихідними координатами та керуючими сигналами. Спершу створюється модель інверсної динаміки об'єкта (перемикач у нижньому положенні на рис. 1.2), яка потім підключається до об'єкта і виконує роль регулято-

ра (перемикач у верхньому положенні). Назва методу повністю відображає його суть: регулювання здійснюється через інверсну модель.

Основним недоліком цієї схеми є відсутність явного зворотного зв'язку, що знижує стійкість системи. Такий підхід ефективний лише для стабільних об'єктів, що далеко не завжди відповідає практичним умовам.

Online-навчання. Схема прямого інверсного регулювання в режимі *online* показана на рис. 1.3.

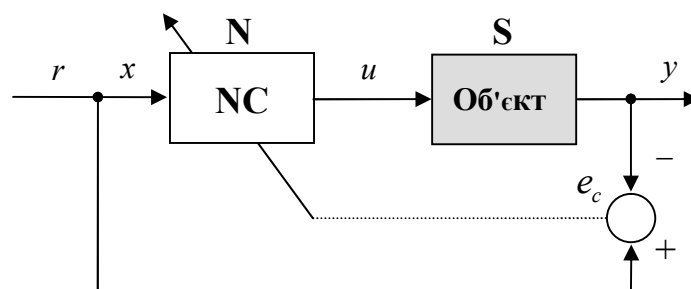


Рисунок 1.3 – Пряме інверсне регулювання з online тренуванням

Метою навчання є формування керуючого сигналу, який змушує регульовану координату максимально точно слідувати сигналу завдання.

Нейромережа навчається під час роботи системи, що робить цей метод придатним для об'єктів із змінними параметрами. Однак помилка формується на виході об'єкта, а не регулятора, тому її доводиться додатково коригувати (пунктирна лінія на рис. 1.2). Формули для обчислення градієнта наведені в табл. 1.1.

Таблиця 1.1 – Обчислення градієнта при регулюванні з online тренуванням

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

Обчислення необхідних похідних здійснюється через матрицю Якобі. Якщо точна модель об'єкта невідома, похідні можуть бути апроксимовані різницевими схемами [32] або враховувати лише знак похідної [33]. При наявності шуму обидва підходи показують низьку ефективність.

Ще один важливий аспект: якщо регулятор не пройшов попереднього грубого тренування, перші моменти роботи у розімкненому контурі можуть спричинити неконтрольовану поведінку системи.

1.2.2 Непряме інверсне нейрорегулювання

У літературі зустрічаються різні назви цього підходу: *feedforward inverse control* (Наренда [34]) або *specialized inverse learning* (Хант [31]). Схема показана на рис. 1.4.

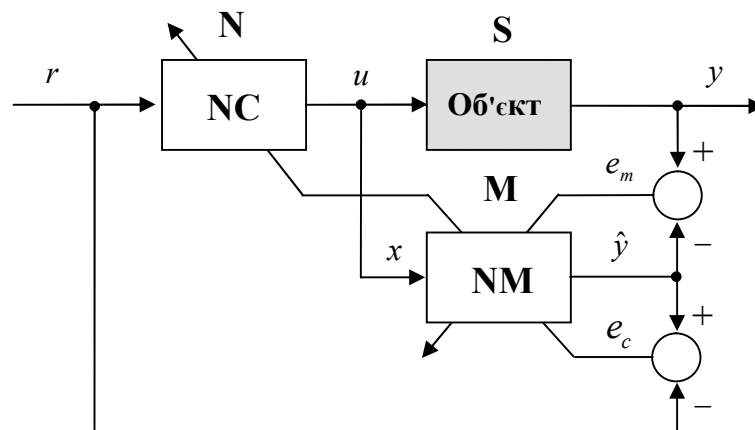


Рисунок 1.4– Непряме інверсне нейрорегулювання

Метод складається з двох етапів:

1. Offline-ідентифікація нейромоделі об'єкта.
2. Використання цієї моделі для коригування ваг нейрорегулятора під час роботи системи.

Матриця Якобі визначається через правило підстановки, використовуючи виходи шарів нейромережі. Далі вона застосовується для підстроювання регулятора.

Недоліком є те, що регулятор може генерувати дії поза діапазоном, у якому навчалася нейромодель, що призводить до нестабільності. У роботах Наренди [35] розглядався варіант, коли модель навчалася online, але це також супроводжувалося високим ризиком нестійкості.

Ішигуро [36] запропонували модифікований підхід *parallel compensator*, де мережа виробляє компенсаційний сигнал паралельно класичному регулятору.

1.2.3 Регулювання на основі внутрішньої моделі (ІМС)

Метод Internal Model Control (ІМС) запропонували Гарсія і Морарі [37]. Він передбачає використання як прямої, так і інверсної моделі об'єкта. Схема наведена на рис. 1.5.

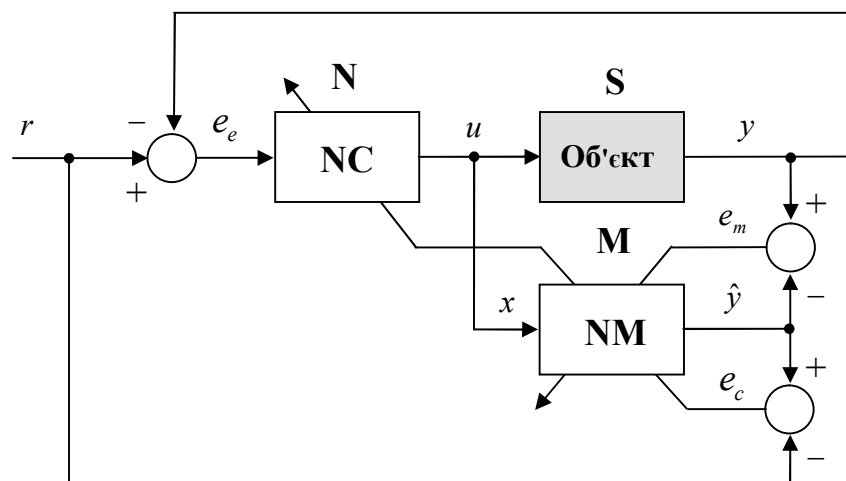


Рисунок 1.5 – Система регулювання з внутрішньою моделлю

Як і в непрямому методі, нейромодель об'єкта спочатку навчають offline. Після цього матриця Якобі використовується для тренування інверсного регулятора в режимі online. Помилка між об'єктом і його моделлю подається на вхід регулятора, що формує замкнений контур.

Властивість структури:

- якщо модель об'єкта ідеальна, ІМС повністю збігається з непрямым інверсним методом, що дозволяє довести стійкість;

– при сталих збуреннях статична помилка відсутня, якщо регулятор реалізує інверсну модель.

Однак ІМС придатний лише для стійких об'єктів; для практичного застосування потрібні фільтри низьких частот. На відміну від класичного ІМС, неймережі не дають критично нових властивостей, окрім зручності ідентифікації.

1.2.4 Feedforward-Feedback-Error регулювання

У цій структурі, зображений на рис. 1.6, до системи додається додатковий зворотний зв'язок і третя неймережа – *Feedback-Error регулятор*.

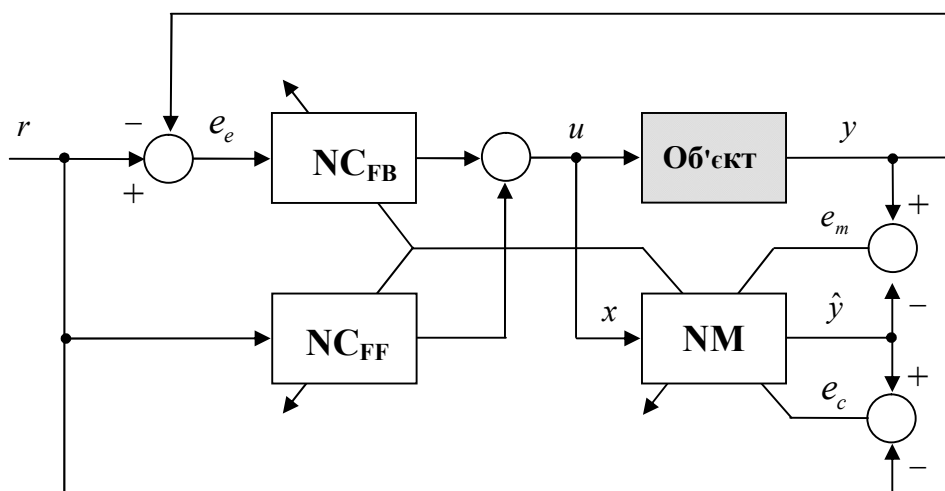


Рисунок 1.6 – Feedforward-Feedback-Error-регулятор

Дві мережі регулятора NC_{FF} і NC_{FB} та модель об'єкта попередньо навчаються offline. Попередньо також формується та навчається у відключеному контурі нейромодель об'єкта NM .

Після отримання цієї моделі запускають процедуру керування: помилка e_c , що виникає в системі, додатково обробляється прямою нейромоделлю, яка генерує коригувальний сигнал для адаптації обох регуляторів.

У сформованій структурі модель NM фактично працює як компенсуючий нейроблок у прямому каналі, подібно до реалізацій непрямого інверсного керування.

Наявність Feedback-Error-регулятора створює замкнений контур, що суттєво покращує стійкість і точність функціонування системи. Такий підхід продемонстровано, зокрема, у роботах [39, 39].

1.2.5 Регулювання за методом Feedback-Error-Learning

На рис. 1.7 показано структурну схему системи керування за підходом *Feedback-Error-Learning (FEL)*, запропонованим Кавато та співавторами [40]. По суті, ця схема нагадує класичне пряме інверсне регулювання, наприклад, із застосуванням ПД-регулятора.

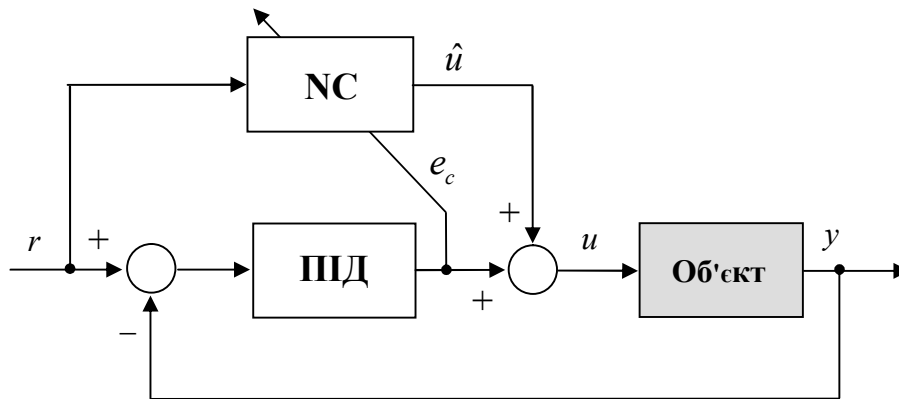


Рисунок 1.7 – Схема регулювання з Feedback-Error-Learning

Основною відмінністю FEL від інших прямих нейрорегуляцій є те, що вихід класичного ПД-регулятора використовується як сигнал помилки для налаштування вагових коефіцієнтів нейрорегулятора. Завдяки цьому відпадає потреба в корекції помилки через модель об'єкта та в обчисленні матриці Якобі. Під час online-тренування нейрорегулятор поступово починає виконувати функцію ПД-регулятора. В кінцевому результаті керування об'єктом здійснюється лише за допомогою нейрорегулятора, який виконує роль інверсної ланки об'єкта. Тим часом ПД-регулятор залишається для корекції випадкових відхилень регульованої координати та забезпечує стабільність системи на початковому етапі навчання.

Серед основних обмежень методу FEL – висока чутливість швидкості збіжності нейрорегулятора до параметрів ПІД-регулятора [41]. Також слід враховувати, що твердження про те, що нейрорегулятор виконує роль інверсної ланки об'єкта, втрачає силу, якщо тренувальні дані не охоплюють весь робочий діапазон. У таких випадках вагові коефіцієнти потрібно коригувати повторно, якщо завдання буде виконуватися в іншій області значень.

1.2.6 Регулювання за методом компенсації завдання.

Метод *Reference Compensation Technique (RCT)*, структурна схема якого наведена на рис. 1.8, був розроблений Юнгом і Хсіа [42].

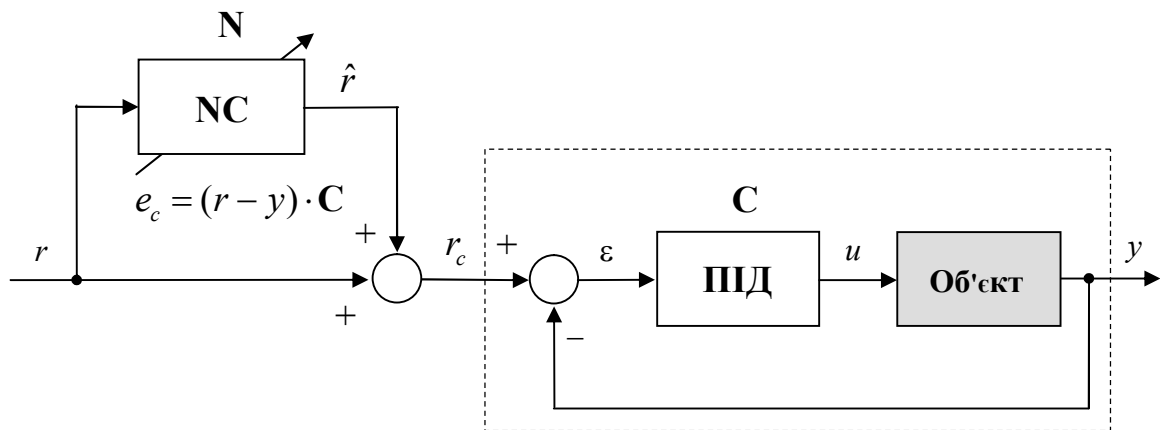


Рисунок 1.8 – Схема регулювання за методом компенсації завдання

Подібно до підходу FEL, у цьому методі немає потреби в обчисленні матриці Якобі, оскільки для навчання нейронної мережі застосовується спеціально сформований сигнал помилки ϵ (див. табл. 1.2).

Таблиця 1.2 – Формули обчислення градієнта при FEL-регулюванні

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

Головна особливість RCT полягає в тому, що корекція нелінійностей виконується безпосередньо на рівні сигналів завдання, а не на рівні керуючих впливів на об'єкт. Перевагою цього підходу є його легка інтеграція в існуючий контур регулювання. Водночас метод має складність у виборі початкових значень параметрів мережі, оскільки неправильне налаштування може призвести до формування небезпечних або некоректних сигналів завдання

1.2.7 Метод миттєвої лінеаризації (Instantaneous Linearization)

Суть методу – локальна лінеаризація нелінійної нейромоделі в робочій точці [43].

Якщо динаміка об'єкта описується NARX-моделлю, її можна проаналізувати через похідні, обчислені за правилом підстановки. Після цього синтезується лінійний регулятор (рис. 1.9).

Метод аналогічний лінійним підходам із непрямою адаптацією.

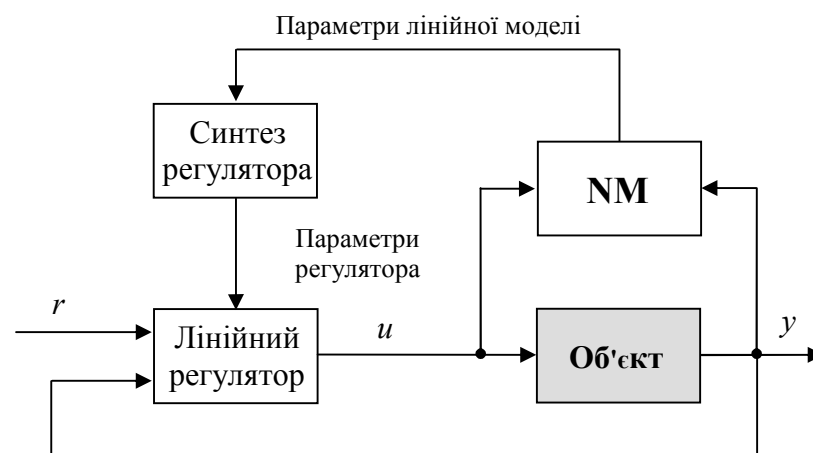


Рисунок 1.9 – Instantaneous linearization і синтез лінійного регулятора

1.2.8 Нелінійне предиктивное регулювання.

Метод MPC (Model Predictive Control) формує послідовність керуючих дій, що мінімізує відхилення виходу від завдання на певному горизонті прогнозування (рис. 1.10) [44–48].

Регулятор – це оптимізаційний алгоритм, який на кожному кроці:

1. розв'язує задачу мінімізації функціоналу якості;

2. використовує лише перший елемент оптимальної послідовності;
3. повторює обчислення на наступному кроці.

Перевага MPC – висока якість керування навіть для суттєво нелінійних систем.

Недолік – велика обчислювальна складність і вимоги до точності моделі.

У цій роботі застосовано саме цей підхід.

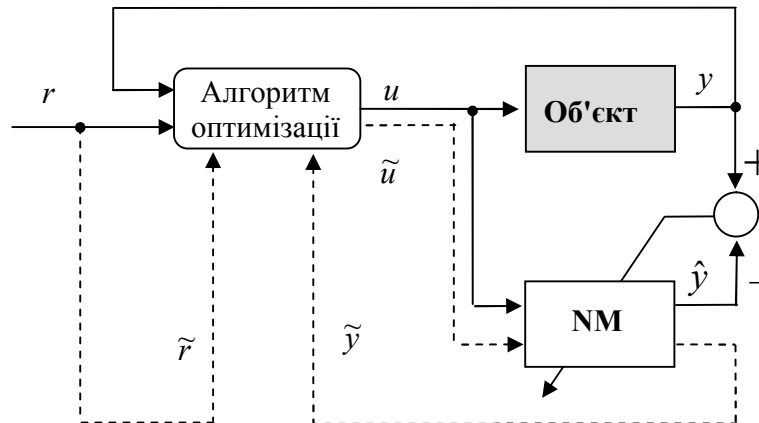


Рисунок 1.10 – Предиктивне регулювання

1.3 Загальні відомості про роторні екскаватори

Головним механізмом гірничодобувних комплексів є роторний екскаватор – складна машина, що включає понад десять взаємопов'язаних силових механізмів і містить як основне, так і допоміжне обладнання для транспортування видобутої породи до транспортних засобів комплексу. На рис. 1.11 наведена схематична компоновальна схема роторного екскаватора.

Конструкція машини складається з поворотної та неповоротної частин, які розділені опорно-поворотним механізмом 1. Він забезпечує обертання верхньої частини екскаватора навколо вертикальної осі. До поворотної частини входить платформа, на якій змонтована стріла ротора 4 із робочим роторним колесом 3 та ковшами 2, розташованими по його периферії. Під час обертання ротора та повороту стріли відбувається відділення породи, яка надходить на внутрішній конвейєр 16, розташований всередині ферменної конструкції стріли.

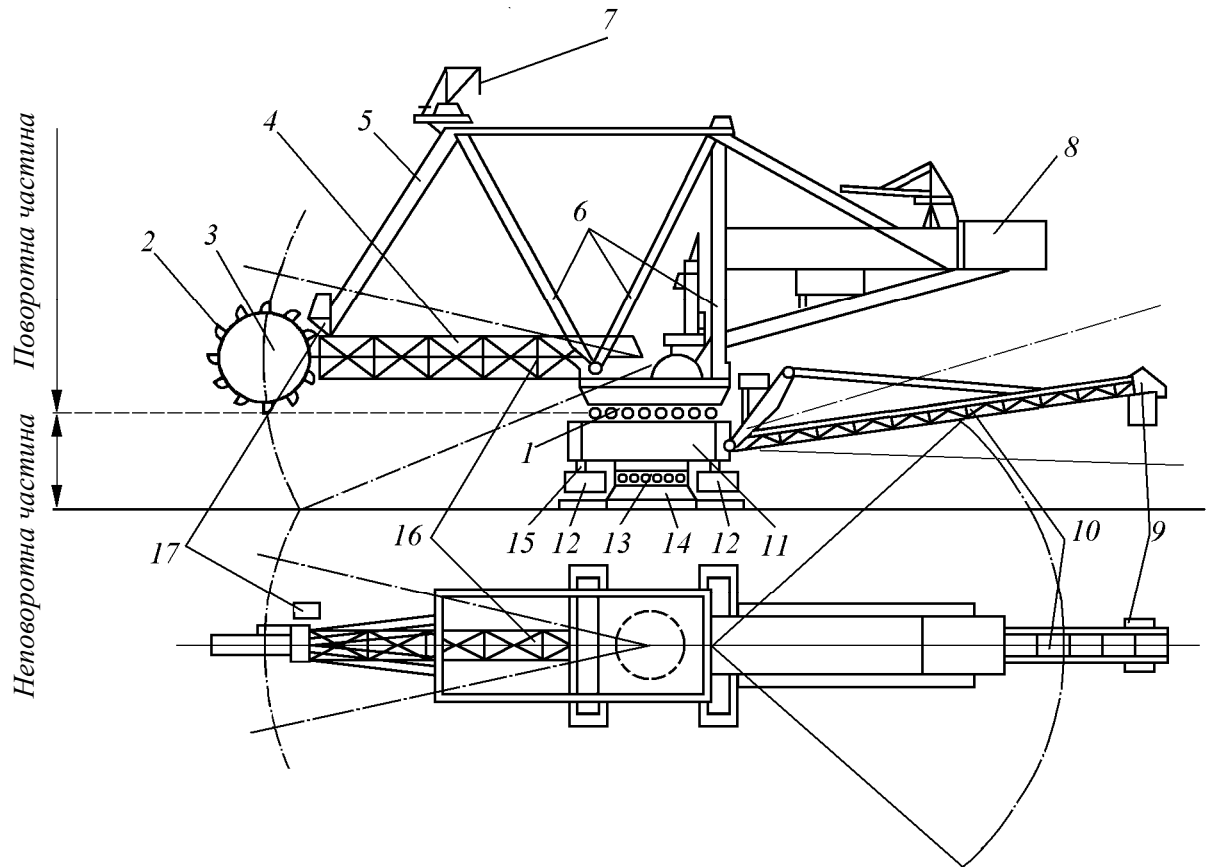


Рисунок 1.11 – Компонувальна схема роторного екскаватора

Далі матеріал переміщується у центральну частину екскаватора, де проходить через дробильний пристрій і, перетинаючи площину опорно-поворотного механізму 1, надходить на конвейер нижньої частини машини, а потім – на конвейер розвантажувальної консолі 10.

Для регулювання положення роторного колеса у вертикальній площині стріла екскаватора може змінювати робоче положення в межах певного сектора за допомогою канатної підвіски 5 і обертатися навколо осі, що проходить через поворотну частину. Верхнє положення стріли визначає максимальну висоту оброблюваного уступу, а нижнє – глибину копання нижче рівня стояння машини. Верхні блоки підвіски закріплені на одному з порталів 6. Для противаги використовується конструкція підкошувань 8, винесена протилежно ротора, що компенсує перекидаючий момент стріли.

Під опорно-поворотним механізмом розташований потужний силовий

пояс 11, з'єднаний з опорною базою 14. До нерухомої частини з обох боків прикріплені потужні гідроциліндри 15, які дозволяють встановлювати конструкцію на ґрунт через базу або шагаюче-рейкове устаткування (лижі). Розвантажувальна консоль 10 разом із конвейєром може змінювати кут установки до 80° відносно напрямку переміщення екскаватора, забезпечуючи автономність її положення під час роботи. Консоль, подібно до стріли ротора, може змінювати вертикальний кут нахилу.

Керування робочими процесами виконується з кабін, розташованих біля ротора 17, на кінці розвантажувальної консолі 10 та на нижній рамі. Для зміни напрямку руху екскаватора нерухома частина разом із поворотною може обертатися щодо бази за допомогою додаткового поворотного механізму 13.

Зручність обслуговування забезпечується наявністю підйомних кранів вантажопідйомністю від 5 до 20 т у трьох точках: на поворотній платформі, на верхній надбудові та на похилому порталі 7.

Основні механізми, що забезпечують роботу екскаватора, включають:

- привід ротора;
- механізм повороту стріли;
- механізм розвороту консолі;
- механізм розвороту екскаватора;
- підйомний механізм;
- ходовий механізм;
- конвейєр стріли;
- конвейєр розвантажувальної консолі.

Компоновка індивідуальних електроприводів роторного екскаватора наведена на рис. 1.12.

У самому роторному колесі встановлюються електродвигуни та редуктори приводу обертання 1, а також електропривод барабанного живильника 2, який направляє породу, що відділяється від ковшів, на конвейєр стріли ротора. Електропривод конвейєра 3 розташований у зоні осі гойдання стріли.

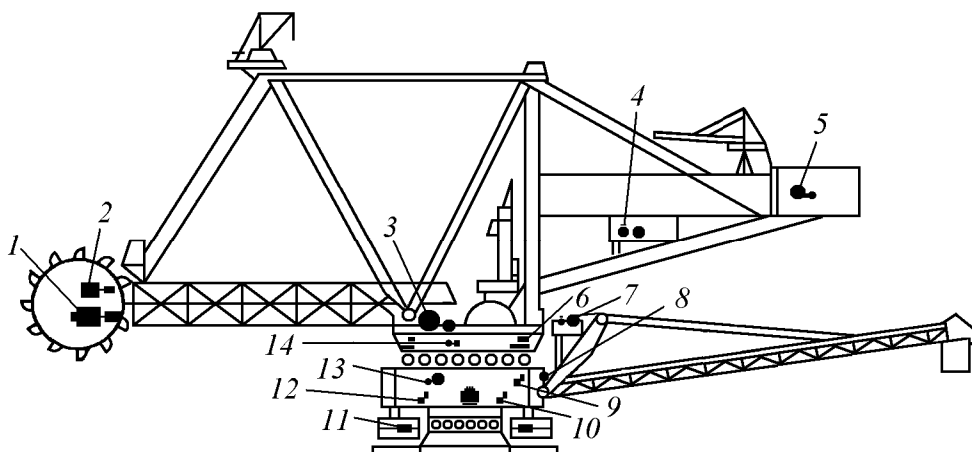


Рисунок 1.12 - Розташування індивідуальних електроприводів роторного екскаватора

На поворотній частині опорно-поворотного механізму встановлені електроприводи 6, що забезпечують обертання верхньої частини машини відносно нижньої. Привід підйому стріли ротора 5 змонтований у надбудові протипаги, де в нижній частині розташоване вентиляційне обладнання 4.

У верхньому поясі, в зоні установки обертових воронок або додроблюючого пристрою, розміщуються відповідні приводи 14. Для вертикального переміщення розвантажувальної консолі використовуються електроприводи підйому 7, змонтовані на спеціальному майданчику. Нижче горизонтальної площини встановлено привід повороту консолі 9, який схожий на привід повороту верхньої будови, але має меншу потужність, а також привід конвеєра розвантажувальної консолі 8.

У неповоротній частині машини знаходяться електроприводи механізму розвороту екскаватора 12, конвеєра нижньої рами 13, а також електрообладнання для введення електроенергії: струмоприймачі, вимикачі, трансформатори та розподільні пристрої високої і низької напруги 10. Електроприводи ходових лебідок 11 встановлені безпосередньо в металоконструкціях лиж рейково-крокуючого ходу.

1.4 Аналіз факторів, що впливають на вибір системи управління електроприводом механізму повороту стріли роторного екскаватора

Ефективна експлуатація роторних екскаваторів можлива лише за умови безперебійної роботи всіх основних механізмів. Відмова будь-якого з ключових електроприводів призводить до порушення технологічного процесу або повної зупинки видобутку ґрунту, що спричиняє простої машини. Збитки при цьому зростають пропорційно продуктивності комплексу, тому до електроприводів основних механізмів висуваються вимоги високої надійності, стабільності характеристик та робочих параметрів.

З одного боку, дотримання цих вимог досягається за рахунок правильного підбору систем електроприводу, схем управління, електричних машин, апаратури та інших елементів, максимально пристосованих до складних умов експлуатації. З іншого боку, для зменшення простоїв машини через відмови електрообладнання, системи управління електроприводами повинні забезпечувати швидке відновлення працездатності без потреби у складному переналагодженні.

Окрім загальних вимог до електроприводів основних механізмів, існують специфічні вимоги, що впливають із функцій, які виконують механізми у технологічному процесі, їх конструктивних особливостей, режимів роботи та характеру навантажень.

Електропривод повороту стріли ротора виконує головний робочий рух – подовжню подачу, що забезпечує різання ґрунту, і, подібно до електроприводу ротора, повинен працювати у тривалому режимі. Натомість електроприводи повороту розвантажувальної консолі та розвороту екскаватора призначені для короткочасних циклів роботи.

Для відробітку кожного забою необхідна зміна напрямку повороту стріли ротора. Розворот екскаватора, що забезпечує правильне орієнтування машини відносно забою, та поворот розвантажувальної консолі, який дозволяє гнучко інтегрувати роботу екскаватора з іншими механізмами комплексу, потребують швидкої зміни напрямку обертання електроприводу. Отже, електроприводи опорно-поворотних механізмів повинні мати реверсивну роботу.

Однією з ключових особливостей електромеханічних систем електроприводів, що забезпечують поворот стріли ротора та розворот екскаватора, є наявність пружно зв'язаних з платформою інерційних мас: стріли ротора з роторним колесом та консолі противаги. Перехідні процеси в таких системах супроводжуються коливаннями, тому до електроприводів висуваються підвищені вимоги щодо плавності розгону та гальмування, обмежуючи ривки та прискорення у допустимих межах.

Для задоволення цих вимог необхідно формувати оптимальну залежність швидкості від часу під час перехідних процесів, що дозволяє мінімізувати динамічні навантаження. Це особливо важливо для електроприводу повороту стріли ротора, де допустимий динамічний момент лише трохи перевищує статичний момент у режимі різання.

Діапазон регулювання швидкості електроприводів опорно-поворотних механізмів має становити не менше $(3-4):1(3-4):1(3-4):1$. Це обумовлено, з одного боку, потребою електроприводів повороту розвантажувальної консолі та розвороту екскаватора забезпечувати достатні швидкості для скорочення часу виконання відповідних операцій при значних кутах повороту. З іншого боку, для електроприводу повороту стріли ротора до цієї вимоги додається необхідність безступінчастого регулювання швидкості.

Обидві умови зумовлені:

- потребою регулювати товщину зрізаної стружки при зміні фізико-механічних властивостей ґрунту;
- відхиленням середнього навантаження приводу роторного колеса від норми;
- конструктивними особливостями екскаватора, що вимагають підтримки постійної продуктивності за рахунок регулювання швидкості повороту по заданому закону під час заходки.

Навантаження на опорно-поворотні механізми змінюються у широких межах через ухил бази екскаватора та вплив вітрового навантаження. Для електроприводу повороту стріли ротора додатково враховуються випадкові зміни

бічної сили різання, що виникають через неоднорідність ґрунту. Незважаючи на це, необхідно підтримувати постійну задану робочу швидкість опорно-поворотних механізмів, особливо електроприводу повороту стріли, оскільки допустиме відхилення параметрів зрізаної стружки від середнього значення є обмеженим. Тому жорсткість робочої ділянки механічних характеристик електроприводів зазначених механізмів повинна бути достатньо високою.

Опорно-поворотні механізми роторного екскаватора є багатомасовими пружно-зв'язаними системами, в яких коливання виникають як у відповідь на керуючі, так і на зовнішні збурюючі впливи. Найбільш помітно це проявляється в електромеханічній системі приводу повороту стріли ротора через розташування робочого органу на кінці довгої та пружної стріли. Через зміну сили різання швидкість подовжньої подачі може зазнавати значних коливань навіть при жорсткій механічній характеристиці електроприводу. При цьому електропривод не завжди ефективно гасить такі коливання, оскільки підвищення жорсткості та стабілізація його швидкості зменшують демпфуючий ефект.

Тому важливим є вибір оптимальної динамічної жорсткості робочої ділянки механічних характеристик електроприводів опорно-поворотних механізмів. Це дозволяє одночасно підтримувати технологічні параметри в заданих межах та ефективно використовувати демпфуючі властивості електроприводу. Крім того, механічна характеристика приводу повороту стріли ротора, враховуючи високий ризик технологічних перевантажень і стопоріння, повинна мати екскаваторну форму та обмежувати розвиваний момент значенням, допустимим за міцністю механізму та конструкцією машини.

Особливістю електроприводу повороту стріли є тісний зв'язок його навантажень із навантаженнями електроприводу роторного колеса. Постійна складова сили різання прямо пропорційна швидкості подовжньої подачі, що обумовлює доцільність введення в систему керування пристрою, який автоматично зменшує швидкість стріли до зупинки при технологічних перевантаженнях електроприводу ротора.

Складність вимог до регульованих електроприводів основних механізмів роторних екскаваторів зумовила використання для більшості з них систем пос-

тійного струму з керованим перетворювачем-двигуном. В якості керованого перетворювача застосовують або електромашинний перетворювач (система Г-Д), або реверсивний тиристорний перетворювач (система ТП-Д). Для електроприводу механізму повороту стріли використовується система Г-Д.

1.5 Постановка задачі дослідження

Сфера застосування нейронних мереж у системах управління активно розвивається. Нейроуправління об'єднує підходи автоматичного керування, штучного інтелекту та робототехніки. Нейронні мережі відзначаються унікальними властивостями: вони здатні навчатися на наданих даних та моделювати складні процеси й взаємозв'язки. Крім того, нейромережі адаптуються до змін у характеристиках об'єктів управління та умов навколишнього середовища і демонструють високу стійкість до локальних пошкоджень завдяки одночасній обробці інформації всіма нейронами. Це робить їх ефективним інструментом для управління складними динамічними системами, де традиційні методи можуть бути недостатніми.

У науковій літературі, як вітчизняній, так і зарубіжній, детально досліджуються теоретичні основи та методи побудови систем управління багатовимірними динамічними об'єктами на базі багат шарових нейронних мереж. Особлива увага приділяється нелінійним характеристикам таких мереж, алгоритмам навчання в реальному часі та прикладам комп'ютерного моделювання систем управління складними об'єктами з багатими взаємозв'язками.

На сьогоднішній день існує багато прикладів успішного впровадження нейромережових систем управління. У літературі описані численні випадки використання нейронних мереж для керування конкретними об'єктами, серед яких: літак [49-50], вертоліт [51], гірничозбагачувальний процес [52], автомобіль-робот [53], гібридний двигун автомобіля [54], пневмоциліндр [55], спеціалізовані об'єкти [56], модель перевернутого маятника [57], електропід [58], турбогенератор [59], зварювальні апарати [60], системи теплового комфорту та енергозбереження [61], кондиціонування повітря [62], електроприводи [63], ро-

ботизовані маніпулятори [64-68] тощо. Ці приклади демонструють ефективність нейромережевих підходів у керуванні нелінійними динамічними об'єктами в складних умовах.

Отже, аналіз сучасних досягнень у сфері синтезу регуляторів із використанням нейронних мереж показує, що цей напрямок є перспективним, а розробка нейромережевих систем для керування об'єктами зі складними кінематичними зв'язками залишається актуальною.

Метою роботи є створення нейромережевої системи управління механізмом повороту стріли роторного екскаватора з урахуванням пружності механічних зв'язків платформи та інерційних мас стріли ротора з роторним колесом і консолі противаги.

Для досягнення цієї мети ставляться наступні завдання:

1. Вибрати систему електроприводу для механізму повороту стріли роторного екскаватора, визначити електродвигун та основне силове електроустаткування, виконати синтез системи управління електроприводом та провести її комп'ютерне моделювання.
2. Розробити математичну модель системи управління з урахуванням податливості механічних зв'язків, змоделювати багатомасову систему та проаналізувати результати.
3. Розробити структурну схему нейромережевої системи управління, що забезпечує високоякісне регулювання з урахуванням кінцевої жорсткості елементів механічної частини електроприводу.
4. Створити модель багатомасової системи на основі багатопарвової прямонаправленої нейронної мережі з високою точністю ідентифікації.
5. Синтезувати нейромережевий регулятор на основі предикативного методу управління, що забезпечує високу швидкодію та якість керування системою.
6. Провести моделювання нейромережевої системи та здійснити аналіз отриманих результатів.

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Аналіз застосування методів нейронних мереж у системах управління електромеханічними об'єктами показав, що штучні нейронні мережі є перспективною альтернативою традиційним підходам для керування нелінійними системами. Основна перевага таких мереж полягає в їх здатності до навчання, що дозволяє знаходити ефективні рішення для складних завдань управління. Завдяки наявності нейронів з нелінійними функціями активації нейромережі ефективно застосовуються для регулювання нелінійних систем, де традиційні методи часто виявляються недостатніми.

2. Показано ефективність використання багат шарового персептрона для побудови нейромережевої моделі об'єкта управління. Ця архітектура є універсальним апроксиматором та широко застосовується на практиці. Розглянуто питання навчання нейромереж та обрано оптимальний метод навчання для нейромережевого регулятора.

3. Для вирішення завдань управління, прогнозування поведінки та моделювання динамічних систем критично важливим є створення адекватних математичних моделей. Показано, що ідентифікацію нелінійних динамічних об'єктів успішно реалізують за допомогою нейронних мереж, зокрема багат шарового персептрона та радіально-базисних мереж.

4. Для оптимізації роботи об'єкта необхідно коригувати параметри регуляторів відповідно до змін робочих умов. Тому застосування адаптивного підходу, що поєднує методи теорії ШНМ, є доцільним, оскільки дозволяє створювати прості нейромережеві моделі та адаптивно підлаштовувати їх параметри разом із параметрами регулятора.

5. Аналіз сучасних структур нейрорегулювання дозволив обрати принцип нелінійного предиктивного регулювання для побудови нейромережевого регулятора системи.

6. Виходячи з технічних вимог до електроприводів опорно-поворотних механізмів роторних екскаваторів, для приводу механізму повороту стріли обрано електропривод постійного струму, виконаний за системою Г–Д.

7. Обґрунтовано доцільність застосування нейромережевих технологій для синтезу системи управління механізмом повороту стріли ротора. Це зумовлено наявністю пружно зв'язаних з платформою інерційних мас стріли ротора з роторним колесом та консолі противаги, що призводить до значних коливань у перехідних процесах. Традиційні системи управління не забезпечують необхідної якості регулювання основних механізмів екскаватора.

8. Проведено аналіз нейромережевих підходів до розв'язання задачі управління. Сформульовано задачу дослідження, яка полягає у створенні нейромережевої системи управління механізмом повороту стріли роторного екскаватора з урахуванням пружних механічних зв'язків.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПОВОРОТУ СТІЛИ РОТОРНОГО ЕКСКАВАТОРА

2.1 Основні технічні характеристики роторного екскаватора та вимоги до системи управління

Вихідні дані, необхідні для розрахунку електроприводу механізму повороту стріли роторного екскаватора та синтезу системи управління, наведені у таблиці 2.1.

Таблиця 2.1 – Технічні дані роторного екскаватора

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
1	Теоретична продуктивність по рихлій масі	$m^3 \cdot год$	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 площ поворотної платформи, що піддається дії вітру щодо осі обертання	$m^2 \cdot m$	$M_{пв}$	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 платформи	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 навантаження	Па	$\lambda_{вт}$	300
19	Максимально подоланий ухил		γ	10°
20	Число двигунів механізму повороту		k	2
21	Максимально допустимі прискорення і уповільнення платформи	с ⁻²	$\omega_{пл}$	0,005
22	ККД редуктора роторного колеса		η_p	0,8
23	ККД редуктора механізму повороту		$\eta_{п}$	0,45 – 0,65
24	Максимальний кут повороту		$\alpha_{п \max}$	±75°

Вимоги до показників якості системи управління.

Система управління повинна забезпечувати статичну точність. При ступінчастому змінінні вхідного сигналу перерегулювання швидкості не повинно перевищувати 10%, а час регулювання повинен бути не більше 5 секунд.

2.2 Розрахунок потужності і вибір привідних електродвигунів

При роботі на кромках ковшів виникають сили F_{δ} , спрямовані перпендикулярно осі стріли ротора, які створюють момент відносно осі обертання:

$$M_{\delta} = F_{\delta} R_k,$$

де R_k – радіус різання в горизонтальній площині, вимірюваний по краях ковшів.

Радіус різання залежить від кута нахилу стріли ротора β_c відносно горизонту і змінюється залежно від оброблюваного шару: $R_k = R_0 \cos \beta_c$, де $R_0 = R_k$ максимальне значення радіусу при горизонтальному положенні стріли. Даному положенню відповідає максимальна величина M_{δ} , яка що приймається як розрахункова

$$M_{\delta} = M_{\delta \max} = F_{\delta} R_0,$$

$$M_{\delta} = M_{\delta \max} = F_{\delta} R_0 = 12 \cdot 10^4 \cdot 37 = 444 \cdot 10^4 \text{ Нм}.$$

Через різне розташування поверхонь конструкції машини з обох боків осі обертання поворотної частини виникає момент, спричинений вітровим навантаженням. При сталій силі вітру момент $M_{\text{пв}}$ визначається формулою:

$$M_{\text{вт}} = M_{\text{вт max}} \cos \alpha_{\text{п}} = k_{\text{ад}} \lambda_{\text{вт}} M_{\text{пв}} \cos \alpha_{\text{п}}, \quad (2.1)$$

де $M_{\text{вт max}}$ – значення максимального моменту, спричиненого вітровим навантаженням (при $\alpha_{\text{п}} = 0$);

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

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

$M_{\text{пв}}$ – момент площ поворотної платформи, що піддається дії вітру щодо осі обертання

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

При відхиленні осі обертання від вертикалі на кут γ та повороті стріли на кут $\alpha_{\text{п}}$ виникає момент від дії невідновжених сил тяжіння поворотної частини. Якщо площина кута γ співпадає з площиною подовжнього перерізу екскаватора, момент від дії цих сил $M_{\text{нв}}$ можна розрахувати як:

$$M_{\text{нв}} = G_{\text{нв}} R_{\text{цт}} \sin \alpha_{\text{п}} \sin \gamma, \quad (2.2)$$

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

Розрахунках виконується за формулами:

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

$m_{\text{пч}}$, $m_{\text{вч}}$, $m_{\text{в пч}}$ – маси поворотної, відвальної частин і вантажу, що знаходиться на транспортері приймальної частини

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

Q – теоретична продуктивність по рихлій масі;

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

$L_{\text{пч}}$ – довжина транспортера приймальної частини; в розрахунках приймається $L_{\text{пч}} = R_0$.

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

$k_{\text{р}}$ – коефіцієнт розпушування ґрунту, $k_{\text{р}} = 1,1 - 1,5$

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

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

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

Як видно з формул (2.1) і (2.2), складові статичного навантаження $M_{\text{вт}}$ та $M_{\text{нв}}$ змінюються залежно від кутового положення платформи та мають активний характер – можуть сприяти або протидіяти руху.

Момент опору обертанню, спричинений тертям $M_{\text{тр}}$ обчислюється так:

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

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

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

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

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

$f_{\text{пр}}$ і $f_{\text{ек}}$ коефіцієнти, що враховують прослизання роликів $f_{\text{пр}} = 1,1 - 1,25$ та ексцентриситет прикладення навантаження $f_{\text{ек}} = 1,0 - 1,25$

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

Моменти $M_{\text{б}}$ і $M_{\text{тр}}$ мають пасивний характер і завжди гальмують рух.

Сумарне статичне навантаження на осі поворотної частини визначається як:

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

Дослідження показали, що для практичних розрахунків можна приймати:

$$M_{\text{вт}} + M_{\text{тр}} + M_{\text{нв}} \cong 0,5M_6.$$

Виходячи з цього, потужність двигунів для повороту стріли обчислюється за формулою:

$$P_{\text{д}} = \frac{1,5M_6\omega_{\text{пл}}}{\eta_{\text{п}}k} \cdot 10^{-3},$$

де $\eta_{\text{п}}$ – ККД механізму повороту;

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

k – кількість приводів механізму (рекомендується $k = 2$);

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

З каталогу двигунів, що застосовуються на роторних екскаваторах, обирають два однакових двигуни, технічні характеристики яких занесено у таблицю 2.1. Коректність вибору перевіряють після побудови навантажувальної діаграми.

Таблиця 2.1 – Технічні дані двигунів

Тип електродвигуна	Номинальні дані							Опір обмоток, Ом			Число активних провідників якоря	Число паралельних гілок якоря	Число полюсів	Момент інерції якоря, кг м^2
	ПВ, %	Потужність кВт	ККД, %	Частота обертання, об/хв.	Напруга, В.	Струм якоря, А.	Магнітний потік полюса мВб	якоря, Ом	Незалежного збудження	додаткових полюсів				
ДВ-808	Часовий	37	89	575	220	192	38,1	0,034	44,4	0,02	186	2	4	2

2.3 Розрахунок швидкісної діаграми механізму повороту стріли екскаватора

Допустимі значення прискорення та уповільнення платформи $\dot{\omega}_{\text{пл}}$ перераховуються на вал двигуна за формулою:

$$\dot{\omega}_{\text{доп}} = \dot{\omega}_{\text{пл}} \cdot i_{\text{п}},$$

$i_{\text{п}}$ — передавальне число механізму повороту:

$$i_{\text{п}} = \frac{\omega_{\text{н}}}{\omega_{\text{пл}}},$$

$\omega_{\text{н}}$ — номінальна кутова швидкість обертання двигуна;

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

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

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

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

З використанням цих даних визначається час розгону та гальмування двигуна до номінальної швидкості:

$$t_{\text{р}} = t_{\text{г}} = \frac{\omega_{\text{н}}}{\dot{\omega}_{\text{доп}}},$$

$$t_{\text{р}} = t_{\text{г}} = \frac{60,21}{50,175} = 1,2 \text{ c}.$$

Кути повороту платформи під час розгону та гальмування обчислюються за формулами:

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

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

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

Кутові переміщення платформи під час руху з постійною швидкістю визначаються як:

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

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

$$\alpha_y = 2,61 \text{ рад}.$$

Час роботи механізму повороту на ділянці з постійною швидкістю визначається формулою:

$$t_y = \frac{\alpha_y \cdot i_{п}}{\omega_{н}},$$

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

Швидкісна діаграма роботи електроприводу механізму повороту стріли роторного екскаватора наведена на рис. 2.1.

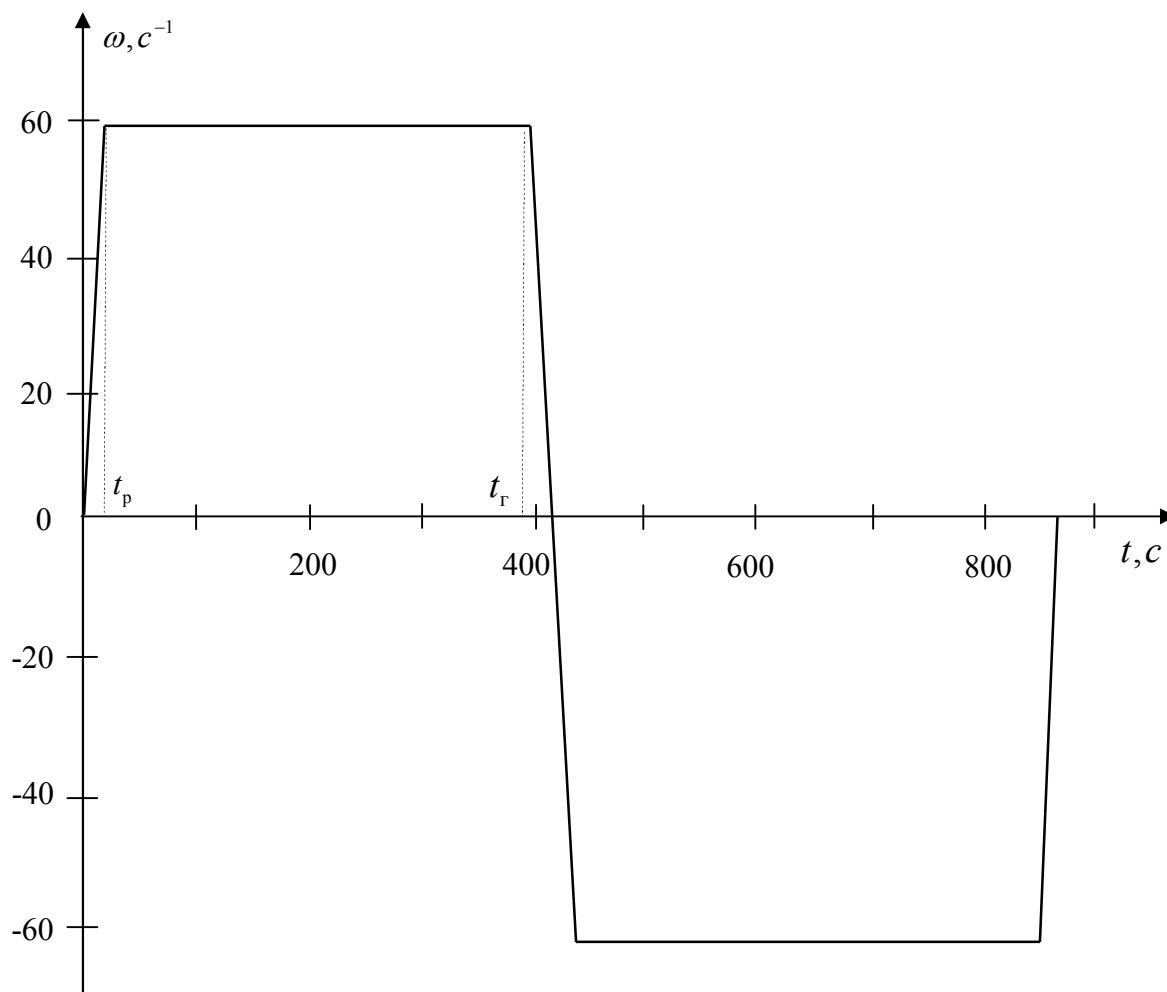


Рисунок 2.1 – Діаграма зміни швидкості електроприводу механізму повороту стріли роторного екскаватора

2.4 Побудова діаграми навантаження та перевірка двигунів за нагрівом і перевантажувальною здатністю. Вибір основного електроустаткування

При складанні навантажувальної діаграми електроприводу механізму повороту стріли екскаватора необхідно враховувати, що величини статичних моментів $M_{вт}$ та $M_{нв}$ змінюються в залежності від кута повороту платформи, кута нахилу стріли та напрямку вітрового навантаження.

Для різних ділянок робочого циклу, тобто при зміні кута α_n , сумарний статичний момент розраховується наступним чином:

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

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

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

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

де $\|$ – модуль відповідного моменту)

На ділянках розгону та гальмування необхідно враховувати динамічний момент

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

$\dot{\omega}_{\text{пл}}$ – допустиме прискорення і уповільнення платформи;

$J_{\text{м}\Sigma}$ – приведений момент інерції, що включає інерцію поворотної платформи та двигунів.

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

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

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

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

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

Динамічний момент $M_{\text{дин}}$ додається до статичного моменту $M_{\text{ст}}$ при розгоні і віднімається при гальмуванні, тобто результуючий момент визначається як:

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

Розрахунок для конкретного кута, наприклад $\alpha_n = -75^\circ$, проводиться наступним чином:

$$M_{\sigma} = 444 \cdot 10^4 \text{ Нм} ; \quad M_{\text{тр}} = 64,275 \cdot 10^3 \approx 6,43 \cdot 10^4 \text{ Нм}.$$

$$M_{\text{вт}} = 84 \cdot 10^4 \cdot \cos(-75^\circ) = 21,74 \cdot 10^4 \text{ Нм}.$$

$$M_{\text{нв}} = 62,4 \cdot 10^4 \cdot \sin(-75^\circ) = -60,84 \cdot 10^4 \text{ Нм}.$$

$$M_{\text{ст}} = 444 \cdot 10^4 + 6,43 \cdot 10^4 + |21,74 \cdot 10^4| - |-60,84 \cdot 10^4| = 411,33 \cdot 10^4 \cdot \text{Нм};$$

$$M_{\text{рез}} = M_{\text{ст}} + M_{\text{дин}} = 411,33 \cdot 10^4 + 641,4 \cdot 10^4 = 1052,73 \cdot 10^4 \text{ Нм}.$$

Отримані результати заносяться до табл. 2.2. На основі даних таблиці будується навантажувальна діаграма, представлена на рис. 2.2.

Перевірка двигунів за нагрівом здійснюється методом еквівалентного моменту. Для визначення середньоквадратичного моменту навантаження результуючу діаграму розбивають на часові інтервали Δt_i , що відповідають цикловій тривалості $\Delta \alpha_n$, і в кожному інтервалі обчислюють середнє значення моменту $M_{\text{рез } i}$ (табл. 2.3).

Таблиця 2.2 – Визначення значень результуючого моменту для побудови діаграми навантаження

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

Продовження таблиці 2.2

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

Таблица 2.3 – Значення $M_{\text{рез } i}$ та Δt_i , що застосовуються для розрахунку середньоквадратичного моменту

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$\Delta t_i, c$	1,2	13,385	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585
$M_{\text{рез } i}, Hm$	1052,8	415,83	424,41	433,46	442,45	451,59	460,73	469,79	478,7	487,395	495,81	503,88	511,55	518,74	525,42

i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33
$\Delta t_i, c$	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	13,385	1,2	1,2
$M_{\text{рез } i}, Hm$	531,02	537,02	541,85	545,98	549,39	552,04	553,98	555,03	555,32	554,82	553,52	551,44	548,6	545,6	540,69	215,1	-108,27	-1009,1

i	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
$\Delta t_i, c$	13,385	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585
$M_{\text{рез } i}, Hm$	-365,06	-360,18	-355,86	-352,27	-349,42	-347,34	-346,04	-345,54	-345,84	-346,98	-348,82	-351,48	-354,88	-359,02	-363,85	-369,34	-375,45	-382,12

i	52	53	54	55	56	57	58	59	60	61	62	63	64	
$\Delta t_i, c$	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	14,585	13,385	1,2	
$M_{\text{рез } i}, Hm$	-389,32	-396,98	-405,05	-413,47	-422,16	-431,07	-440,13	-449,27	-458,4?	-467,5	-476,46	-164,33	152,05	

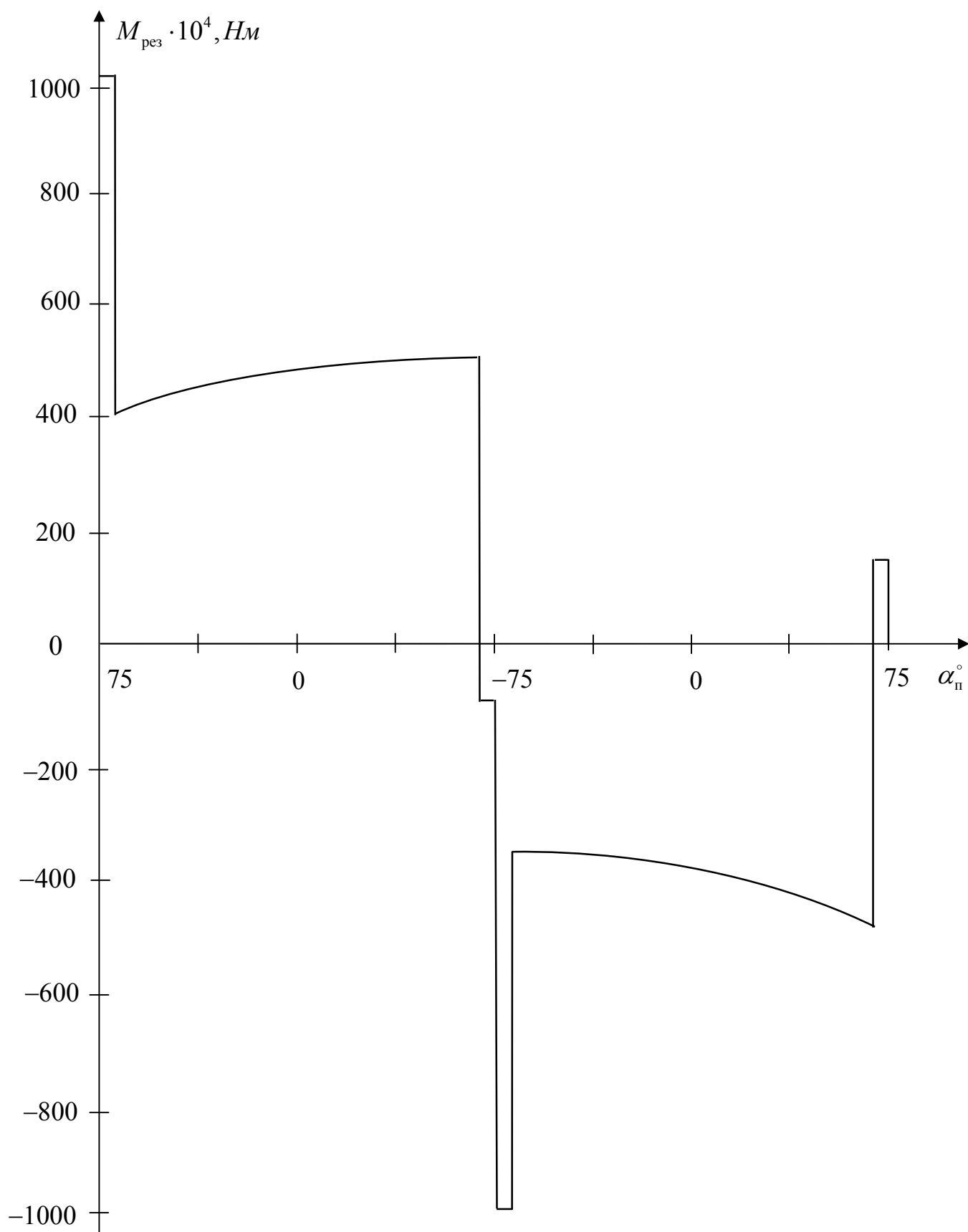


Рисунок 2.2 – Навантажувальна діаграма роботи механізму повороту стріли роторного екскаватора

Подальше визначення середньоквадратичного моменту двигуна виконується за формулою:

$$M_{\text{д экв}} = \frac{1}{\eta_{\text{п}} i_{\text{п}} k} \cdot \sqrt{\frac{\sum_{i=1}^n M_{\text{рез},i}^2 \Delta t_i}{T_{\text{ц}}}},$$

де $T_{\text{ц}}$ – тривалість повного циклу, тобто час руху механізму у прямому та зворотному напрямках

$$T_{\text{ц}} = 2 \cdot (t_{\text{п}} + t_{\text{р}} - t_{\text{у}}) = 2 \cdot (1,2 + 1,2 + 435,167) = 875,134 \text{ с}$$

$$M_{\text{д экв}} = \frac{1}{0,6 \cdot 10035 \cdot 2} \sqrt{\frac{(1052,8^2 \cdot 10^8 + (-106,27)^2 \cdot 10^8 + (-1009,1)^2 \cdot 10^8 + 152,05^2 \cdot 10^8) \cdot 1,2 + (415,83^2 + 215,1)^2 + (-365,06)^2 + (-164,33)^2 \cdot 13,385 \cdot 10^8 + (424,41^2 + 433,36^2 + 44245^2 + 451,59^2 + 460,73^2 + 469,79^2 + 478,7^2 + 487,395^2 + 495,81^2 + 503,88^2 + 511,55^2 + 51874 + 525,42^2 + 531,53^2 + 537,02^2 + 541,85^2 + 545,98^2 + 549,39^2 + 552,04^2 + 553,98^2 + 555,03^2 + 55532^2 + 554,82^2 + 553,52^2 + 551,44^2 + 548,6^2 + 545,0^2 + 540,69^2 + (-360,18)^2 + (-355,86)^2 + (-352,27)^2 + (-349,42)^2 + (-347,34)^2 + (-346,04)^2 + (-345,54)^2 + (-345,84)^2 + (-346,98)^2 + (-348,82)^2 + (-351,48)^2 + (-354,88)^2 + (-359,02)^2 + (-363,85)^2 + (-369,34)^2 + (-375,45)^2 + (-382,12)^2 + (-389,32)^2 + (-396,98)^2 + (-405,05)^2 + (-413,47)^2 + (-422,16)^2 + (-431,07)^2 + (-440,13)^2 + (-449,27)^2 + (-458,42)^2 + (-446,5)^2 + (-476,46)^2 \cdot 14,585 \cdot 10^8}{875,134}} = 412,42 \text{ Нм/}$$

Номінальний момент заздалегідь обраного електродвигуна повинен перевищувати обчислений еквівалентний момент $M_{\text{д экв}}$:

$$M_{\text{н}} \geq M_{\text{д экв}} . \quad (2.3)$$

Номінальне значення моменту вибраного двигуна:

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

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

Оскільки нерівність (2.3) виконується ($614,6 \geq 412,42$), можна зробити висновок, що двигун відповідає вимогам щодо нагрівостійкості.

Далі двигун підлягає перевірці на перевантажувальну здатність. Максимальний момент, знайдений за діаграмою навантаження та приведений до валу двигуна, повинен бути меншим за допустиме значення $2,5M_{\text{н}}$:

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

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

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

Таким чином, за показником перевантажувальної здатності вибраний двигун також вибраний правильно.

Для живлення якірних кіл двох послідовно з'єднаних двигунів постійного струму з незалежним збудженням, що керуються зміною напруги у колі якоря в системі «керований перетворювач – двигун», застосовується система Г–Д. Живлення обмотки збудження генератора забезпечує тиристорний збудник.

Беручи до уваги необхідні показники потужності, струму, напруги та кількості двигунів, за каталогом підбираємо відповідні перетворювальні агрегати. Їх технічні характеристики подані у таблицях 2.4 і 2.5.

Таблиця 2.4 – Характеристики генератора

Тип генератора	$P_{нг},$ <i>кВт</i>	$n_{нг},$ <i>об/хв</i>	$U_{нг},$ <i>В</i>	$I_{нг},$ <i>А</i>	$\Phi_{нг},$ <i>мВб</i>	$R_{яг},$ <i>Ом</i>	$R_{дпг},$ <i>Ом</i>	$R_{озг},$ <i>Ом</i>	$R_{ког},$ <i>Ом</i>	$2p$	$W_{зб},$
П 102	110	1450	460	230	27	0,0132	0,00683	3,29	-	4	850

Таблиця 2.5 – Характеристики тиристорного збудника генератора

Тип перетворювача	Схема випрямлення	Спосіб управління групами тиристорів	Номінальні дані			Перевантажувальна здібність	
			P_n <i>кВт</i>	U_n <i>В</i>	I_n <i>А</i>	тривала	Коротко-часна
БУВ-3603-12	Реверсивна трьохфазна нульова	Роздільне управління	11,5	230	50	1,0	1,1

2.5 Аналіз статичних характеристик системи

На рисунку 2.3 подано структурну схему системи типу Г–Д.

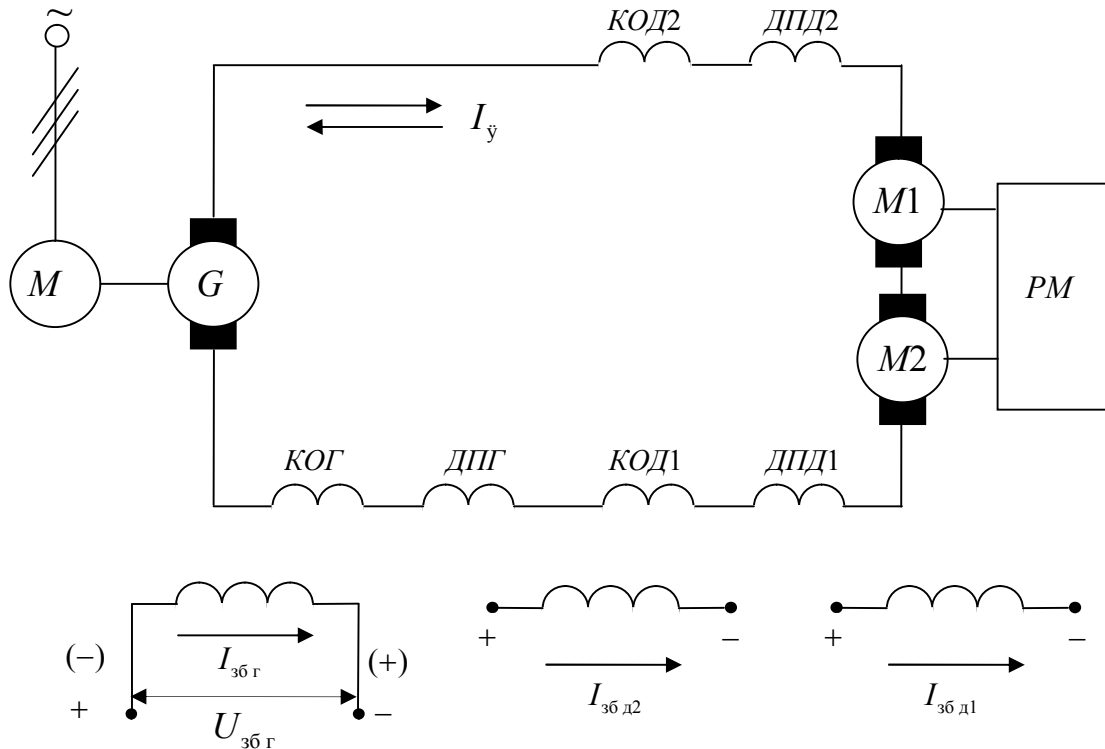


Рисунок 2.3 – Схема якірного кола перетворювального агрегату

У схемі використані такі позначення:

$M1, M2$ – двигуни механізму;

G - генератор і приводний двигун генератора;

M - приводний двигун генератора;

$КОД1, КОД2, КОГ, ДПД1, ДПД2, ДПГ$ – компенсаційні обмотки і обмотки додаткових полюсів двигунів $M1, M2$ і генератора G ;

$I_{я}$ – струм якірного кола;

$I_{зб\ Д1}, I_{зб\ Д2}$ – струми збудження двигунів $M1, M2$;

$I_{зб\ Г}$ - струми збудження генератора G ;

PM – робочий механізм.

Як видно зі схеми, якірні кола двигунів і генератора з'єднані послідовно, утворюючи єдине якірне коло. Якщо розглядати ЕРС генератора E_r як напругу

живлення двигунів, електромеханічна та механічна характеристики двигуна описуються рівняннями:

$$\omega_{\text{д}} = \frac{E_{\Gamma}}{k k_{\text{д}} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{k k_{\text{д}} \Phi_{\text{нд}}} = \omega_0 - \Delta\omega_{\text{ст}} ; \quad (2.4)$$

$$\omega_{\text{д}} = \frac{E_{\Gamma}}{k k_{\text{д}} \Phi_{\text{нд}}} - M_{\text{д}} \frac{R_{\text{я}\Sigma}}{k (k_{\text{д}} \Phi_{\text{нд}})^2} = \omega_0 - \Delta\omega_{\text{ст}} , \quad (2.5)$$

де $I_{\text{я}}$ – струм в якірному колі;

$M_{\text{д}}$ – момент, створюваний двигуном;

$k_{\text{д}}$ – конструктивний коефіцієнт двигуна;

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

Добуток $k_{\text{д}} \Phi_{\text{нд}}$ (позначимо як c) можна знайти з природної характеристики двигуна:

$$c = k_{\text{д}} \Phi_{\text{нд}} = \frac{U_{\text{нд}} - I_{\text{нд}} R_{\text{я}\Sigma}}{\omega_{\text{нд}}} ;$$

$U_{\text{нд}}$, $I_{\text{нд}}$, $\omega_{\text{нд}}$ – номінальні напруга, струм та швидкість обертання;

$R_{\text{я}\Sigma}$ – сумарний опір якірного кола системи Г–Д.

Сумарний опір визначається як:

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

$R_{\text{я}\Sigma\text{д}}$ – опір якірного кола двигуна, приведений до робочої температури,

$$R_{\text{я}\Sigma\text{д}} = k_t (R_{\text{яд}} + R_{\text{дпд}} + R_{\text{код}}) ;$$

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

k_t – температурний коефіцієнт, для машин з класом ізоляції H $k_t = 1,36$

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

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

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

R_{Γ} , $R_{\text{дп } \Gamma}$, $R_{\text{ко } \Gamma}$ – опір обмотки якоря, компенсаційної обмотки і обмотки додаткових полюсів генератора

$$R_{\Sigma \Gamma} = 1,36 \cdot (0,0132 + 0,00683 + 0) = 0,0272 \text{ Ом},$$

$$R_{\Sigma} = 2 \cdot 0,7344 + 0,0272 = 0,17408 \text{ Ом},$$

$$c = k_d \Phi_{\text{нд}} = \frac{220 - 192 \cdot 0,07344}{60,2} = 3,42.$$

Подальші обчислення виконуються, виходячи з моделі одного еквівалентного двигуна. Для нього отримаємо:

$$U_{\text{не}} = 2U_{\text{н}} = 2 \cdot 220 = 440 \text{ В},$$

$$I_{\text{не}} = I_{\text{н}} = 192 \text{ А},$$

$$R_{\Sigma \text{е}} = 2R_{\Sigma \Gamma} = 2 \cdot 0,07344 = 0,1469 \text{ Ом},$$

$$\omega_{\text{не}} = \omega_{\text{н}} = 60,21 \text{ с}^{-1}.$$

Швидкість ідеального холостого ходу

$$\omega_0 = \frac{2U_{\text{н}}}{2 \cdot k_d \Phi_{\text{нд}}} = \frac{2 \cdot 220}{2 \cdot 3,42} = 64,32 \text{ с}^{-1}.$$

Номінальний момент одного двигуна

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

Оскільки механічні та електромеханічні характеристики є лінійними, для побудови достатньо визначити дві характерні точки: режим холостого ходу й точку при номінальному навантаженні. Під час побудови необхідно враховувати залежність між струмом і моментом двигуна $M_{\text{н}} = k_{\text{д}} \Phi_{\text{нд}} \cdot I_{\text{н}}$.

Виконуємо побудову характеристик розімкненої системи Г–Д для наступних значень $E_{\text{г}}$:

$$1) E_{\text{г}} = U_{\text{не}} = 2 \cdot U_{\text{н}} = 2 \cdot 220 = 440 \text{ В}$$

перша точка – ідеальний холостий хід

$$I_{\text{я}} = 0; \quad M_{\text{д}} = 0; \quad \omega_{01} = \omega_0 = 64,32 \text{ с}^{-1};$$

друга точка – номінальне навантаження

$$I_{\text{я}} = I_{\text{н}} = 192 \text{ А}; \quad M_{\text{д}} = M_{\text{н}} = 656,6 \text{ Н} \cdot \text{м}; \quad \omega_1 = \omega_{\text{н}} = 60,2 \text{ с}^{-1};$$

$$2) E_{\text{г}} = 0,5 \cdot U_{\text{не}} = 0,5 \cdot 440 = 220 \text{ В}$$

перша точка – ідеальний холостий хід

$$I_{\text{я}} = 0; \quad M_{\text{д}} = 0; \quad \omega_{02} = 0,5 \cdot \omega_0 = 32,16 \text{ с}^{-1};$$

друга точка – номінальне навантаження

$$I_{\text{я}} = I_{\text{н}} = 192 \text{ А}; \quad M_{\text{д}} = M_{\text{н}} = 656,6 \text{ Н} \cdot \text{м};$$

$$\begin{aligned} \omega_2 &= \frac{E_{\text{г}}}{k k_{\text{д}} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\Sigma}}{k k_{\text{д}} \Phi_{\text{нд}}} = \omega_0 - \Delta \omega_{\text{ст}} = \\ &= \frac{2 \cdot 220}{2 \cdot 3,42} - 192 \cdot \frac{0,17408}{2 \cdot 3,42} = 32,16 - 4,12 = 28,04 \text{ с}^{-1}. \end{aligned}$$

Побудовані електромеханічні та механічні характеристики розімкненої системи Г–Д зображені на рисунку 2.4 (пунктирні лінії).

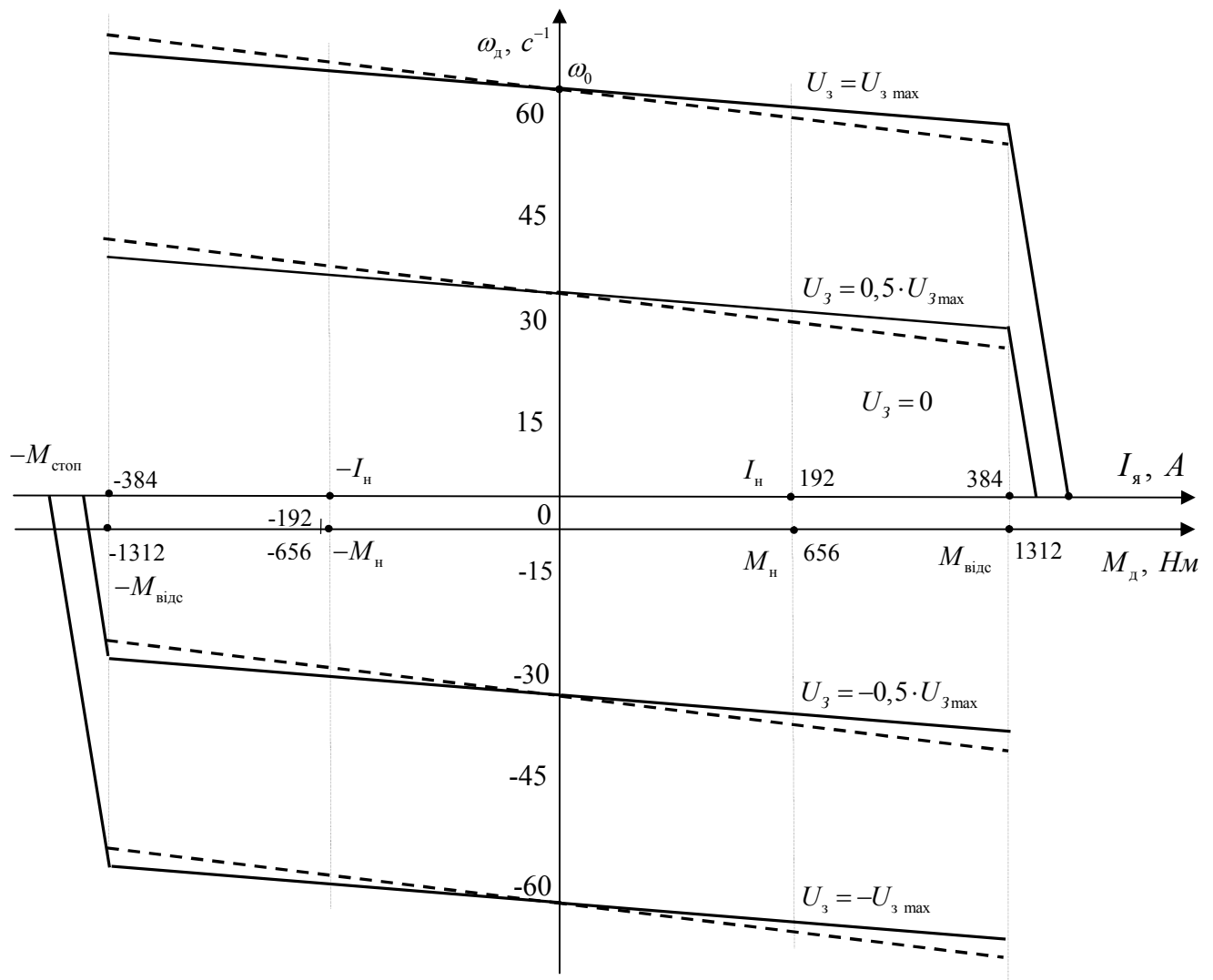


Рисунок 2.4 – Статичні характеристики системи в розімкненому та замкненому станах

Жорсткість механічної характеристики в системі Г–Д є нижчою, ніж у природної характеристики двигуна. Проте для електроприводу механізму повороту стріли встановлюються підвищені вимоги щодо збільшеної жорсткості робочої ділянки характеристики та забезпечення обмеження максимального моменту, тобто формування так званої «екскаваторної» характеристики.

Досягти таких властивостей можливо лише в замкненій системі автоматичного керування електроприводом. На роторних екскаваторах застосовують-

ся САУ, в яких на вхід підсумовуючого підсилювача подаються сигнали завдання, а також сигнали зворотних зв'язків: швидкісного, гнучкого струмового та струмового ЗЗ з відсіченням.

Функціональна структурна схема цієї системи наведена на рис. 2.5.

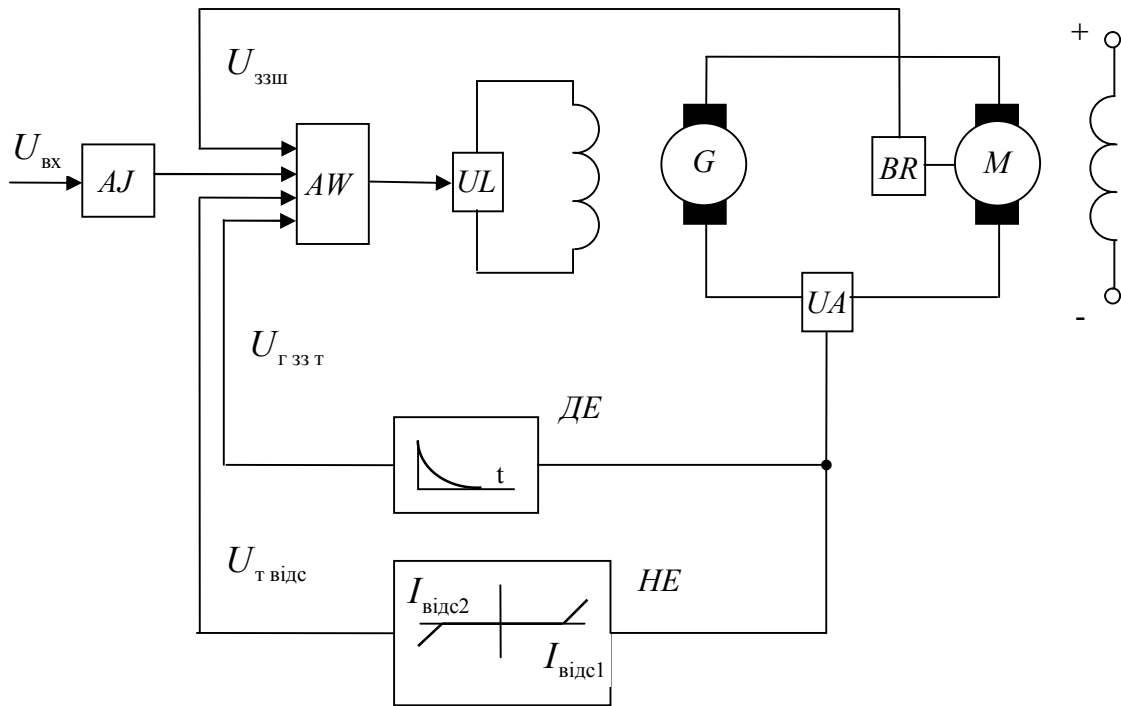


Рисунок 2.5 – Функціональна схема замкнутої системи

Позначення на схемі: AJ – задатчик інтенсивності; AW – підсумовуючий підсилювач; UL – збудник генератора; UA – датчик струму; BR – датчик швидкості (тахогенератор); HE – нелінійний елемент; DE – диференціатор.

Задатчик інтенсивності AJ забезпечує формування сигнального впливу U_3 , що лінійно змінюється у часі, при довільному вході U_{BX} , аж до виконання умови $U_3 = k' U_{BX}$, де k' – коефіцієнт пропорційності. Диференціюючий елемент (DE) створює сигнал $U_{ГЗЗТ}$, пропорційний похідній струму, який реалізує гнучкий зворотний зв'язок за струмом.

Оскільки задатчик інтенсивності та гнучкі ЗЗ не впливають на статичні характеристики, основним елементом, що визначає їх форму, є нелінійний елемент HE , розташований у каналі ЗЗ по струму з відсіченням. Його характерис-

тика має зону нечутливості, яка затримує появу коригувального сигналу до тих пір, поки струм якоря перебуває всередині меж:

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

У таких приводах зазвичай виконується умова:

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

Електромеханічна характеристика замкнутої САУ описується рівнянням

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

де k_{Σ} – загальний коефіцієнт підсилення системи (добуток коефіцієнтів підсилення елементів AW , UL і G у лінійній зоні);

$$k_{\Sigma} = k_{AW} k_{UL} k_{\Gamma} ;$$

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

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

Теоретично при необмеженому зростанні k_{Σ} або $k_{\text{ззш}}$ до ∞ жорсткість характеристики може прагнути до нескінченності. На практиці значення цих коефіцієнтів обмежені вимогами стійкості та фізичними можливостями апаратури.

Коефіцієнт підсилення генератора:

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

де $U_{нг}$, $U_{нзбг}$ – номінальні напруги якоря та збудження генератора;

$I_{нг}$ – номінальний струм генератора;

$E_{нг}$ – номінальна ЕРС генератора

$$E_{нг} = U_{нг} + I_{нг} R_{я\Sigmaг} ,$$

$$E_{г} = 460 + 230 \cdot 0,0272 = 473,4 \text{ В} ,$$

$$k_{г} = \frac{473,4}{220} = 2,09 .$$

Коефіцієнт підсилення тиристорного збудника:

$$k_{UL} = \frac{U_{днзб}}{U_y} ,$$

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

$$U_{днзб} = U_{нзбг} ;$$

U_y – напруга управління на вході тиристорного збудника; приймаємо

$$U_y = 10 \text{ В} .$$

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

Коефіцієнт підсилення підсумовуючого підсилювача $k_{AW} = 0,1$.

$$k_{\Sigma} = 2,09 \cdot 23 \cdot 0,1 = 4,8 .$$

Коефіцієнт ЗЗ за швидкістю $k_{ззш}$ визначаємо із заданого статичного перепаду швидкості $\Delta\omega_{ст1}$ при номінальному струмі:

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

звідки

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

Величину $\Delta \omega_{\text{ст1}}$ приймаємо рівним половині перепаду в розімкненій системі $\Delta \omega_{\text{ст}}$ при номінальному навантаженні

$$\Delta \omega_{\text{ст1}} = 0,5 \Delta \omega_{\text{ст}} = 0,5(\omega_0 - \omega_{\text{н}}),$$

$$\Delta \omega_{\text{ст1}} = 0,5 \cdot (64,32 - 60,2) = 2,06 \text{ c}^{-1},$$

$$k_{\text{ззш}} = 192 \cdot \frac{0,17408}{4,8 \cdot 2,06} - \frac{2 \cdot 3,42}{4,8} = 1,97.$$

Коефіцієнт струмового зворотного зв'язку з відсіченням $k_{\text{т відс}}$ обирається так, щоб забезпечити потрібний граничний струм стопоріння:

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

Задаємося значеннями струмів $I_{\text{відс}}$ та $I_{\text{стоп}}$:

$$I_{\text{відс}} = 2 I_{\text{нд}}; \quad I_{\text{стоп}} = 2,3 I_{\text{нд}},$$

$$I_{\text{відс}} = 2 \cdot 192 = 384 \text{ A},$$

$$I_{\text{стоп}} = 2,3 \cdot 192 = 441,6 \text{ A}.$$

Після визначення всіх коефіцієнтів напруга завдання $U_{\text{з max}}$ вибирається з умови, що при $E_{\text{г}} = k U_{\text{нд}}$ система відтворює швидкість ідеального холостого ходу розімкненої системи.

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

звідки

$$U_{3\max} = \frac{U_{\text{нд}} (k_c + k_{\Sigma} k_{\text{ззш}})}{c k_{\Sigma}},$$

$$U_{3\max} = \frac{440 \cdot (2 \cdot 3,42 + 4,8 \cdot 1,97)}{2 \cdot 3,42 \cdot 4,8} = 218,4B,$$

$$k_{\text{т відс}} = \frac{218,4 - 441,6 \cdot 0,17408 / 4,8}{441,6 - 384} = 3,51.$$

Підставивши знайдені параметри у рівняння (2.4), отримуємо залежність електромеханічної характеристики замкнутого приводу.

$$\begin{aligned} \omega_{\text{д}} &= U_3 \frac{4,8}{2 \cdot 3,42 + 4,8 \cdot 1,97} - I_{\text{я}} \frac{0,17408}{2 \cdot 3,42 + 4,8 \cdot 1,97} - \\ &- 1(\Delta I)(I_{\text{я}} - 384) \cdot \frac{2,1 \cdot 4,8}{2 \cdot 3,42 + 4,8 \cdot 1,97} = \\ &= 0,294 \cdot U_3 - 0,01073 \cdot I_{\text{я}} - 1(\Delta I)(I_{\text{я}} - 384) \cdot 0,62. \end{aligned}$$

На основі отриманої формули будуються електромеханічні та механічні характеристики системи для таких режимів:

$$U_{31} = U_{3\max} = 218,4B,$$

$$U_{32} = 0,5U_{3\max} = 0,5 \cdot 218,4B = 109,2B.$$

Кожну характеристику визначають за трьома обчисленими точками.

Перша точка – холостий хід

$$I_{\text{я}} = 0; \quad M_{\text{д}} = 0;$$

$$\omega_{01} = 0,294 \cdot 218,4 = 64,32 \text{ c}^{-1},$$

$$\omega_{02} = 0,294 \cdot 109,2 = 32,15 \text{ c}^{-1}.$$

Друга точка – номінальне навантаження

$$I_{\text{я}} = I_{\text{відс}} = 384 \text{ А}; \quad M_{\text{д}} = M_{\text{відс}} = 2 \cdot M_{\text{н}} = 2 \cdot 656,6 = 1313,2 \text{ Нм},$$

$$\omega_1 = 0,294 \cdot 218,4 - 0,01073 \cdot 384 = 64,32 - 4,12 = 60,2 \text{ с}^{-1},$$

$$\omega_2 = 0,294 \cdot 109,2 - 0,01073 \cdot 384 = 32,16 - 4,12 = 28,04 \text{ с}^{-1}.$$

Третя точка – стопоріння.

$$\omega_1 = \omega_2 = 0; \quad I_{\text{я1}} = I_{\text{стоп1}} = I_{\text{стоп}} = 441,6 \text{ А},$$

$$M_{\text{д1}} = M_{\text{стоп1}} = k_{\text{д}} \Phi_{\text{нд}} I_{\text{стоп1}} = 3,42 \cdot 441,6 = 1510,27 \text{ Нм}.$$

Струмові значення в режимі стопоріння для другої характеристики знаходимо, підставляючи відповідні дані у рівняння електромеханічної характеристики:

$$\omega = 0; \quad I_{\text{я}} = I_{\text{стоп2}}$$

$$0 = 0,294 \cdot 109,2 - 0,017 I_{\text{стоп2}} - (I_{\text{стоп2}} - 384) \cdot 0,62;$$

$$I_{\text{стоп2}} = 405,08 \text{ А}; \quad M_{\text{стоп2}} = 3,42 \cdot 405,08 = 1385,37 \text{ Нм}.$$

Побудовані електромеханічні та механічні характеристики замкненої системи Г–Д для позитивних і негативних значень напруги завдання наведено на рис. 2.4 суцільними кривими.

2.6 Аналіз динамічних характеристик системи

Для вивчення динаміки роботи приводу розглянемо алгоритмічну структуру, подану на рис. 2.6. Вона побудована відповідно до функціональної схеми, наведеної на рисунку 2.5.

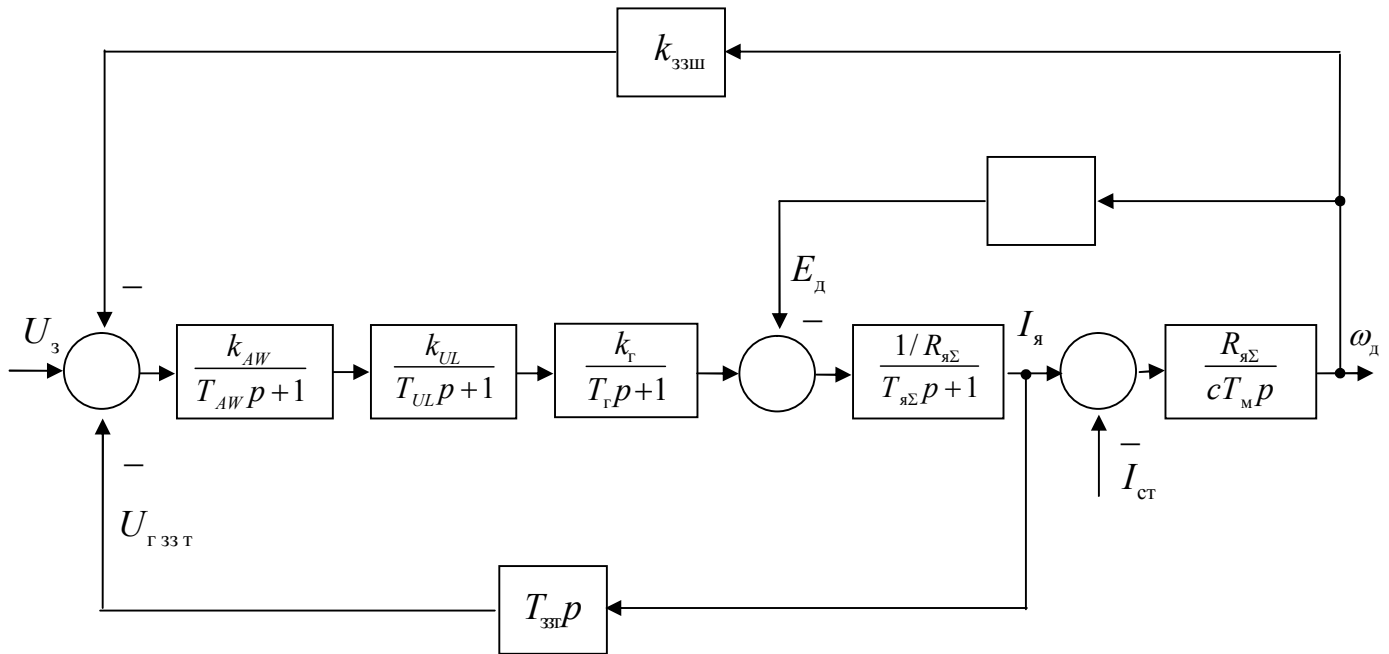


Рисунок 2.6 – Алгоритмічна схема САУ

У цій схемі не показано зворотний зв'язок за струмом із відсіченням, оскільки він не впливає на поведінку системи в робочих режимах, а також не показано задатчик інтенсивності.

Щоб спростити подальші обчислення, ігноруємо постійні часу AW та тиристорного збудника, адже вони приблизно у 50 разів менші від сталої часу генератора. У результаті отримуємо приведену алгоритмічну схему (рис. 2.7). Для зручності введемо позначення передавальних функцій окремих елементів схеми: $W_1(p)$, $W_2(p)$ і т.д.

Для спрощення розрахунків виключаємо з розгляду постійні часу і тиристорного збудника, постійні часу яких в 50 разів менші постійної часу генератора T_r . Отримаємо алгоритмічну схему, зображену на рис.2.7. Для зручності перетворення введемо позначення передавальних функцій ланок схеми $W_1(p)$, $W_2(p)$ і т.д. З цієї структури можна отримати передавальні функції для швидкості та струму як за задавальним сигналаом, так і за збуренням.

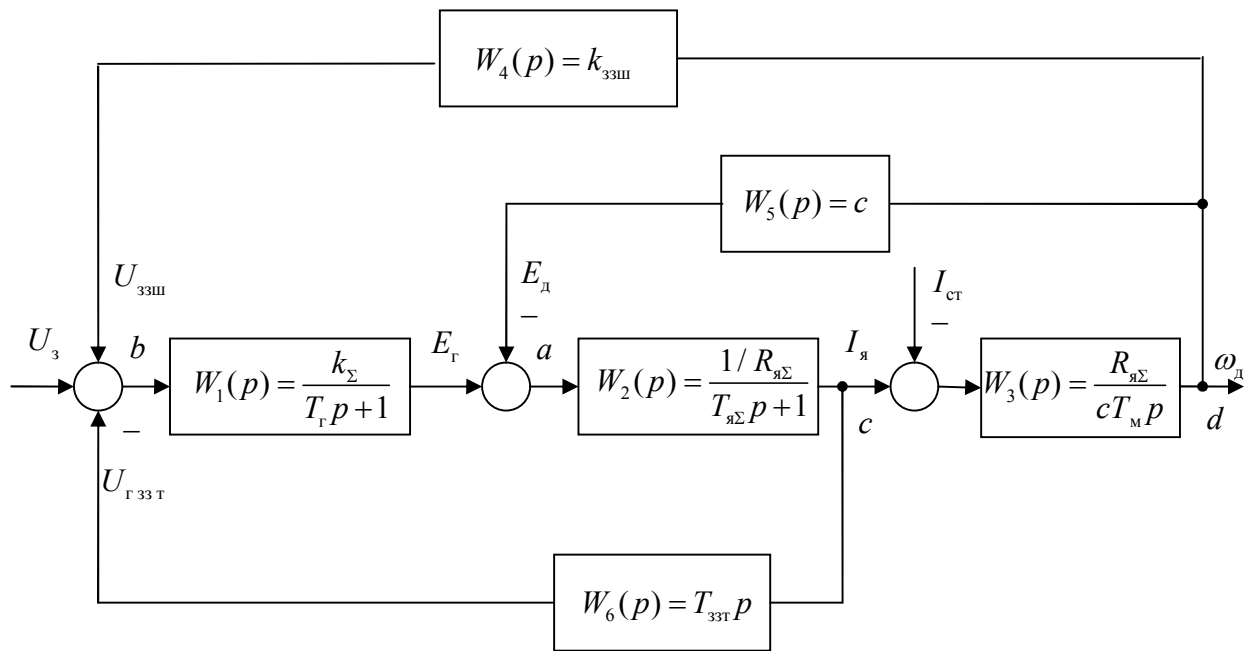


Рисунок 2.7 – Спрощена алгоритмічна схема САУ.

Щоб позбутися перехресних зворотних зв'язків, перенесемо точку підключення зворотного зв'язку по ЕРС двигуна з точки a у точку b . Отримана схема показана на рис. 2.8. На ній присутня передавальна функція $W_8(p)$, яка визначається так:

$$W_7(p) = \frac{W_5(p)}{W_1(p)}.$$

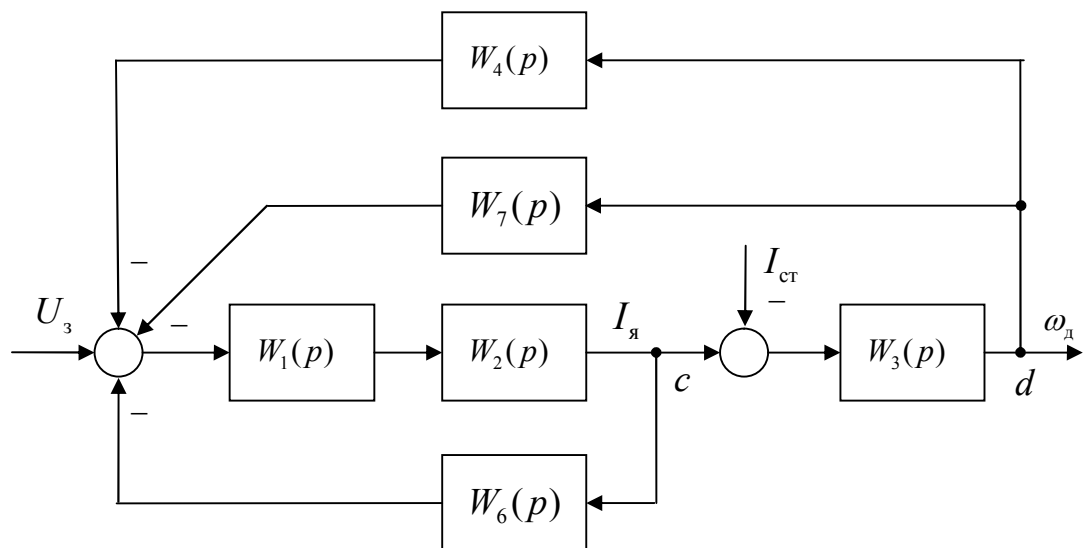


Рисунок 2.8 – Перетворена алгоритмічна схема

Для отримання передавальної функції $W_I(p) = \frac{\omega_d(p)}{U_3(p)}$ переносимо точку c у точку d , покладаючи при цьому $I_{ст} = 0$. У результаті матимемо структуру, подану на рис. 2.9.

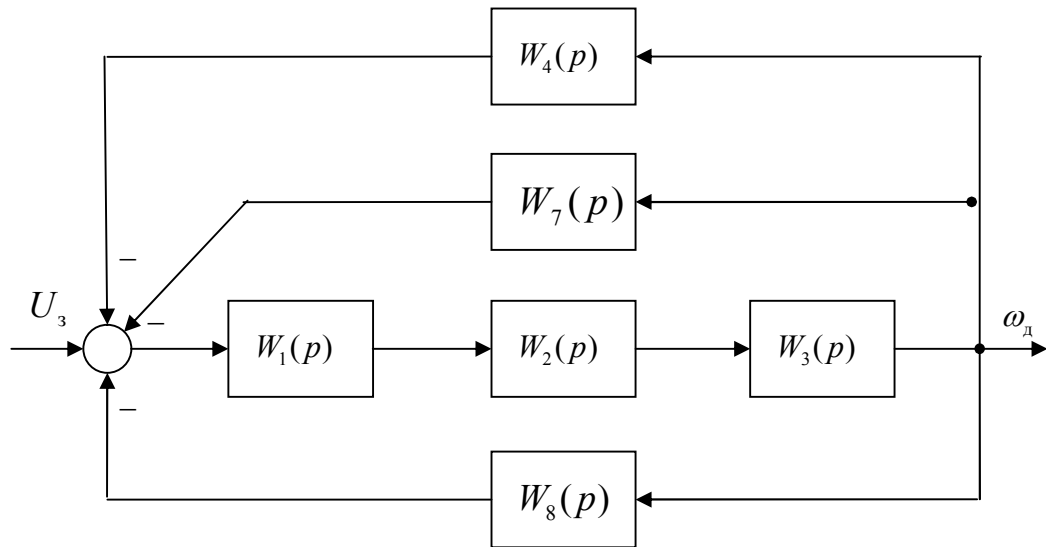


Рисунок 2.9– Схема для отримання передавальної функції $W_I(p) = \frac{\omega_d(p)}{U_3(p)}$

Передавальна функція $W_8(p)$ обчислюється за формулою:

$$W_8(p) = \frac{W_6(p)}{W_3(p)}.$$

Оскільки елементи з передавальними функціями W_4 , W_8 , W_7 , та працюють паралельно, замінюємо їх однією еквівалентною ланкою з передавальною функцією:

$$W_9(p) = W_4(p) + W_7(p) + W_8(p) = W_4(p) + \frac{W_5(p)}{W_1(p)} + \frac{W_6(p)}{W_3(p)}.$$

У результаті дістаємо розрахункову схему (рис. 2.10).

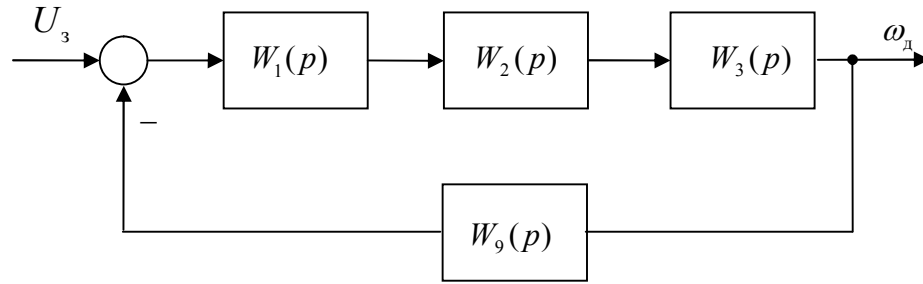


Рисунок 2.10– Розрахункова схема для визначення $W_I(p) = \frac{\omega_d(p)}{U_3(p)}$

Згідно зі схемою обчислюємо передавальну функцію швидкості за задавальним впливом:

$$\begin{aligned}
 W_I(p) &= \frac{\omega_d(p)}{U_3(p)} = \frac{W_1(p) \cdot W_2(p) \cdot W_3(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_3(p) \cdot W_9(p)} = \\
 &= \frac{\frac{k_\Sigma}{T_\Gamma p + 1} \cdot \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot \frac{R_{я\Sigma}}{c T_M p}}{1 + \frac{k_\Sigma}{T_\Gamma p + 1} \cdot \frac{1/R_{я\Sigma}}{T_{я\Sigma} p + 1} \cdot \frac{R_{я\Sigma}}{c T_M p} \cdot (k_{33\Pi} + \frac{T_{33T} T_M c p^2}{R_{я\Sigma}} + \frac{c(T_\Gamma p + 1)}{k_\Sigma})} = \\
 &= \frac{\frac{k_\Sigma}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1)c T_M p}}{1 + \frac{k_{33\Pi} R_{я\Sigma} k_\Sigma + k_\Sigma T_{33T} T_M c p^2 + R_{я\Sigma} c(T_\Gamma p + 1)}{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1)c T_M p}} = \\
 &= \frac{k_\Sigma R_{я\Sigma}}{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1)c T_M p + k_{33\Pi} R_{я\Sigma} k_\Sigma + k_\Sigma T_{33T} T_M c p^2 + R_{я\Sigma} c(T_\Gamma p + 1)} \cdot)
 \end{aligned}$$

Після поділу чисельника і знаменника на c та $R_{я\Sigma}$ і введення позначення $k_d = 1/c$ маємо:

$$W_I(p) = \frac{\omega_d(p)}{U_3(p)} = \frac{k_\Sigma k_d}{(T_\Gamma p + 1) \cdot (T_{\Sigma} p + 1) T_M p + k_{33\text{ш}} k_\Sigma k_d + \frac{k_\Sigma T_{33\Gamma} T_M}{R_{\Sigma}} p^2 + T_\Gamma p + 1} =$$

$$= \frac{k_\Sigma k_d}{T_\Gamma T_{\Sigma} T_M p^3 + \left(T_\Gamma T_M + T_{\Sigma} T_M + \frac{k_\Sigma T_{33\Gamma} T_M}{R_{\Sigma}} \right) \cdot p^2 + (T_\Gamma + T_M) p + k_\Sigma k_d k_{33\text{ш}} + 1}.$$

Для визначення передавальної функції $W_2(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)}$ знову повертаємося до схеми рис. 2.8, приймаючи $U_3 = 0$. Виконуємо необхідні перетворення передавальних функцій, що дає змогу отримати структуру, показану на рис. 2.11.

$$W_{10}(p) = W_4(p) + W_7(p),$$

$$W_{11}(p) = \frac{W_1(p) \cdot W_2(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_6(p)},$$

$$W_{12}(p) = W_{10}(p) \cdot W_{11}(p).$$

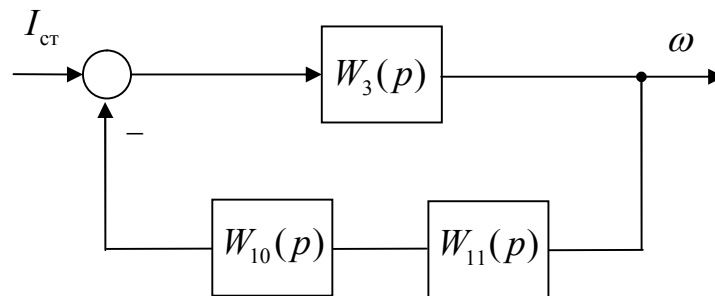


Рисунок 211 – Схема для визначення передавальної функції $W_{II}(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)}$

За цією структурою знаходимо передавальну функцію швидкості за дією збурення:

$$W_{II}(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)} = \frac{W_3(p)}{1 + W_3(p) \cdot W_{10}(p) \cdot W_{11}(p)} =$$

$$\begin{aligned}
&= \frac{W_3(p)}{1 + W_3(p) \cdot [W_4(p) + W_8(p)] \cdot \frac{W_1(p) \cdot W_2(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_6(p)}} = \\
&= \frac{\frac{R_{я\Sigma}}{cT_{\text{м}}p}}{1 + \frac{R_{я\Sigma}}{cT_{\text{м}}p} \cdot \left[k_{33\text{ш}} + \frac{c(T_{\Gamma}p + 1)}{k_{\Sigma}} \right] \cdot \frac{\frac{k_{\Sigma}}{(T_{\Gamma}p + 1)} \cdot \frac{1/R_{я\Sigma}}{(T_{я\Sigma}p + 1)}}{1 + \frac{k_{\Sigma}}{(T_{\Gamma}p + 1)} \cdot \frac{1/R_{я\Sigma}}{(T_{я\Sigma}p + 1)}} \cdot T_{33\Gamma}p} = \\
&= \frac{\frac{R_{я\Sigma}}{cT_{\text{м}}p}}{1 + \frac{\left[k_{33\text{ш}}k_{\Sigma} + c(T_{\Gamma}p + 1) \right]}{cT_{\text{м}}p \left[(T_{\Gamma}p + 1)(T_{я\Sigma}p + 1) + pT_{33\Gamma}k_{\Sigma}/R_{я\Sigma} \right]}} = \\
&= \frac{R_{я\Sigma} \left[(T_{\Gamma}p + 1)(T_{я\Sigma}p + 1) + pT_{33\Gamma}k_{\Sigma}/R_{я\Sigma} \right]}{cT_{\text{м}}p \left[(T_{\Gamma}p + 1)(T_{я\Sigma}p + 1) + pT_{33\Gamma}k_{\Sigma}/R_{я\Sigma} \right] + k_{33\text{ш}}k_{\Sigma} + c(T_{\Gamma}p + 1)} = \\
&= \frac{R_{я\Sigma}^2 \left[(T_{\Gamma}p + 1)(T_{я\Sigma}p + 1) + pT_{33\Gamma}k_{\Sigma}/R_{я\Sigma} \right]}{R_{я\Sigma}(T_{\Gamma}p + 1)(T_{я\Sigma}p + 1)cT_{\text{м}}p + k_{33\text{ш}}R_{я\Sigma}k_{\Sigma} + k_{\Sigma}T_{33\Gamma}T_{\text{м}}cp^2 + R_{я\Sigma}c(T_{\Gamma}p + 1)}.
\end{aligned}$$

Після ділення чисельника і знаменника на $R_{я\Sigma}$ та c , отримаємо:

$$\begin{aligned}
W_{\text{II}}(p) &= \frac{\omega_{\text{д}}(p)}{I_{\text{ст}}(p)} = \\
&= \frac{R_{я\Sigma}k_{\text{д}}T_{\Gamma}T_{я\Sigma}p^2 + R_{я\Sigma}k_{\text{д}} \left(T_{\Gamma} + T_{я\Sigma} + \frac{k_{\Sigma}T_{33\Gamma}}{R_{я\Sigma}} \right) p + R_{я\Sigma}k_{\text{д}}}{T_{\Gamma}T_{я\Sigma}T_{\text{м}}p^3 + \left(T_{\Gamma}T_{\text{м}} + T_{я\Sigma}T_{\text{м}} + \frac{k_{\Sigma}T_{33\Gamma}T_{\text{м}}}{R_{я\Sigma}} \right) \cdot p^2 + (T_{\Gamma} + T_{\text{м}})p + k_{\Sigma}k_{\text{д}}k_{33\text{ш}} + 1}.
\end{aligned}$$

Щоб визначити передавальну функцію $W_{\text{III}}(p) = \frac{I_{\text{я}}(p)}{U_3(p)}$, перетворимо схе-

му на рис. 2.8 у вигляд, наведений на рис. 2.12.

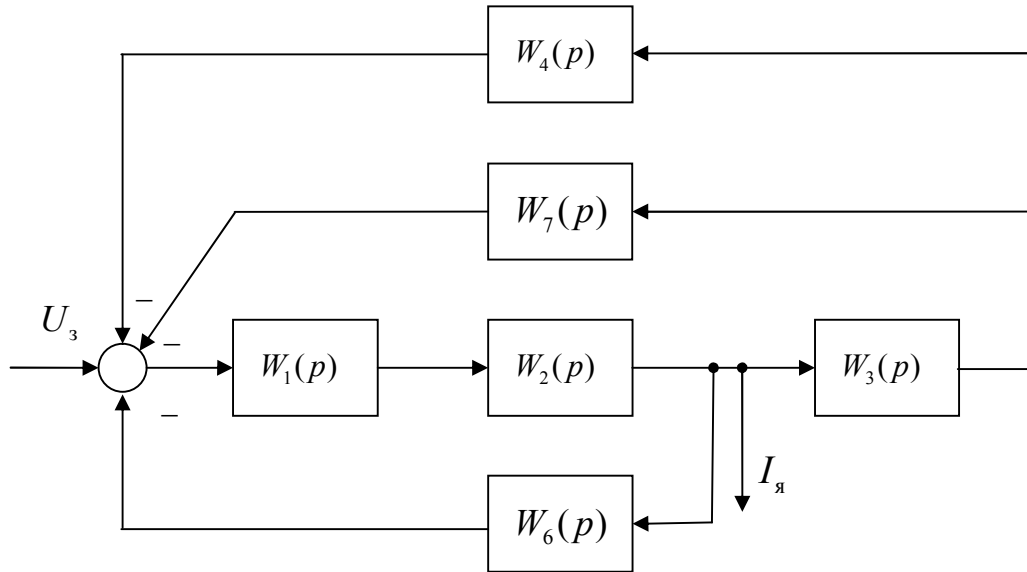


Рисунок 2.12 – Схема для визначення передавальної функції $W_{III}(p) = \frac{I_{я}(p)}{U_3(p)}$

Елементи з передавальними функціями $W_3(p)$, $W_4(p)$, $W_7(p)$ об'єднуємо в одну ланку з передавальною функцією:

$$W_{13}(p) = W_3(p) \cdot [W_4(p) + W_7(p)].$$

Використовуючи отриману раніше передавальну функцію $W_{11}(p)$, формуємо розрахункову структуру, подану на рис. 2.13

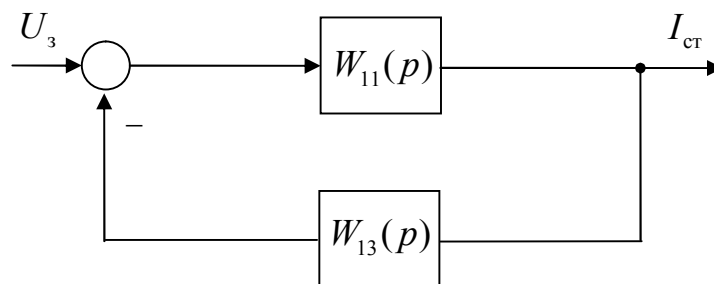


Рисунок 2.13 – Розрахункова схема для визначення $W_{III}(p) = \frac{I_{я}(p)}{U_3(p)}$

За нею отримуємо шукану передавальну функцію струму за задавальним сигналом:

$$\begin{aligned}
W_{III}(p) &= \frac{I_{\text{я}}(p)}{U_3(p)} = \frac{W_{11}(p)}{1 + W_{11}(p) \cdot W_{13}(p)} = \\
&= \frac{\frac{W_1(p) \cdot W_2(p)}{1 + W_2(p) \cdot W_6(p)}}{1 + \frac{W_1(p) \cdot W_2(p)}{1 + W_2(p) \cdot W_6(p)} \cdot W_3(p) \cdot [W_4(p) + W_7(p)]} = \\
&= \frac{W_1(p) \cdot W_2(p)}{1 + W_2(p) \cdot W_6(p) + W_1(p) \cdot W_2(p) \cdot W_3(p) \cdot [W_4(p) + W_7(p)]} = \\
&= \frac{\frac{k_{\Sigma}}{(T_{\Gamma}p + 1)} \cdot \frac{1/R_{\text{я}\Sigma}}{(T_{\text{я}\Sigma}p + 1)}}{1 + \frac{1/R_{\text{я}\Sigma}}{(T_{\text{я}\Sigma}p + 1)} \cdot \frac{k_{\Sigma}T_{33\Gamma}p}{(T_{\Gamma}p + 1)} + \frac{k_{\Sigma}}{(T_{\Gamma}p + 1)} \cdot \frac{1/R_{\text{я}\Sigma}}{(T_{\text{я}\Sigma}p + 1)} \cdot \frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot (k_{33\text{ш}} + \frac{c \cdot (T_{\Gamma}p + 1)}{k_{\Sigma}})} \\
&= \frac{\frac{k_{\Sigma}}{R_{\text{я}\Sigma}(T_{\text{я}\Sigma}p + 1)(T_{\Gamma}p + 1)}}{1 + \frac{k_{\Sigma}T_{33\Gamma}p}{R_{\text{я}\Sigma} \cdot (T_{\Gamma}p + 1) \cdot (T_{\text{я}\Sigma}p + 1)} + \frac{k_{\Sigma}}{(T_{\Gamma}p + 1)(T_{\text{я}\Sigma}p + 1)cT_{\text{м}}p} \cdot \frac{k_{33\text{ш}}k_{\Sigma} + c(T_{\Gamma}p + 1)}{k_{\Sigma}}} \\
&= \frac{k_{\Sigma}cT_{\text{м}}p}{R_{\text{я}\Sigma}(T_{\Gamma}p + 1) \cdot (T_{\text{я}\Sigma}p + 1) \cdot cT_{\text{м}}p + k_{\Sigma}T_{33\Gamma}cT_{\text{м}}p^2 + k_{33\text{ш}}k_{\Sigma}R_{\text{я}\Sigma} + R_{\text{я}\Sigma}c(T_{\Gamma}p + 1)}.
\end{aligned}$$

Поділивши чисельник і знаменник на c і $R_{\text{я}\Sigma}$, одержимо остаточний вираз:

$$W_{III}(p) = \frac{I_{\text{я}}(p)}{U_3(p)} = \frac{\frac{k_{\Sigma}T_{\text{м}}p/R_{\text{я}\Sigma}}{(T_{\Gamma}p + 1) \cdot (T_{\text{я}\Sigma}p + 1)T_{\text{м}}p + \frac{k_{\Sigma}T_{OT}T_{\text{м}}}{R_{\text{я}\Sigma}}p^2 + k_{33\text{ш}}k_{\Sigma}k_{\text{д}} + T_{\Gamma}p + 1}}{1} =$$

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

Для визначення $W_{IV}(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$ представимо схему рис. 2.8 у вигляді по-

даному на рис. 2.14.

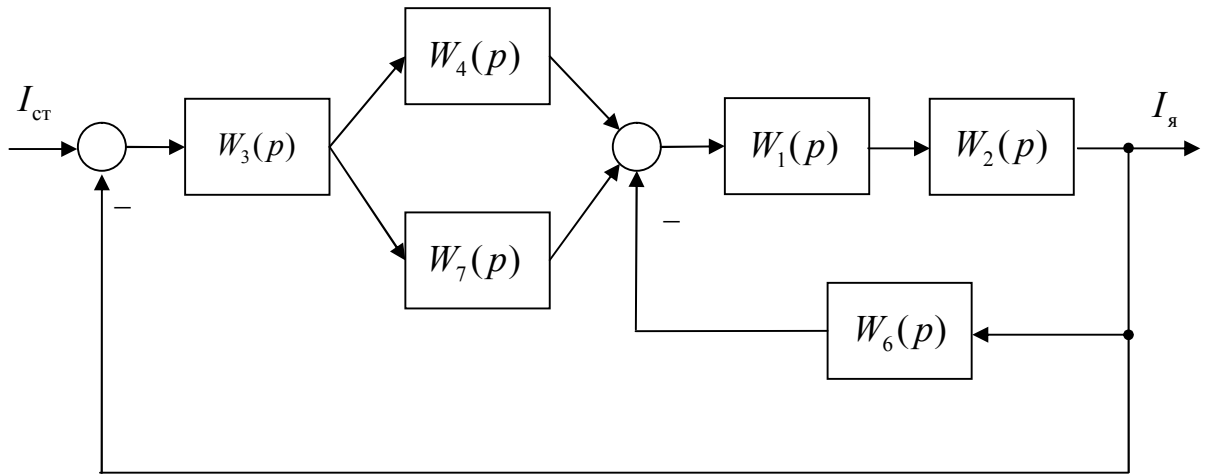


Рисунок 2.14 – Схема для визначення передавальної функції $W_{IV}(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$.

Користуючись схемою рис. 2.14, отримуємо передавальну функцію струму якоря за збурюючою дією:

$$\begin{aligned} W_{IV}(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)} &= \frac{W_3(p) \cdot [W_4(p) + W_7(p)] \cdot \frac{W_1(p) \cdot W_2(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_6(p)}}{1 + W_3(p) \cdot [W_4(p) + W_7(p)] \cdot \frac{W_1(p) \cdot W_2(p)}{1 + W_1(p) \cdot W_2(p) \cdot W_6(p)}} = \\ &= \frac{W_1(p) \cdot W_2(p) \cdot W_3(p) \cdot (W_4(p) + W_7(p))}{1 + W_1(p) \cdot W_2(p) \cdot W_6(p) + W_1(p) \cdot W_2(p) \cdot W_3(p) \cdot [W_4(p) + W_7(p)]} = \end{aligned}$$

$$\begin{aligned}
&= \frac{\frac{k_{\Sigma}}{(T_{\Gamma}p+1)} \cdot \frac{1/R_{\Sigma}}{(T_{\Sigma}p+1)} \cdot \frac{R_{\Sigma}}{cT_{\Sigma}p} \cdot \left[k_{33\text{ш}} + \frac{c(T_{\Gamma}p+1)}{k_{\Sigma}} \right]}{1 + \frac{1/R_{\Sigma}}{(T_{\Sigma}p+1)} \cdot \frac{k_{\Sigma}}{(T_{\Gamma}p+1)} \cdot T_{33\Gamma}p + \frac{k_{\Sigma}}{(T_{\Gamma}p+1)} \cdot \frac{1/R_{\Sigma}}{(T_{\Sigma}p+1)} \cdot \frac{R_{\Sigma}}{cT_{\Sigma}p} \cdot (k_{33\text{ш}} + \frac{c(T_{\Gamma}p+1)}{k_{\Sigma}})} \\
&= \frac{\frac{k_{\Sigma}}{(T_{\Gamma}p+1)(T_{\Sigma}p+1)} \cdot \frac{k_{33\text{ш}}k_{\Sigma} + c(T_{\Gamma}p+1)}{k_{\Sigma}}}{1 + \frac{k_{\Sigma}T_{33\Gamma}p}{R_{\Sigma} \cdot (T_{\Gamma}p+1)(T_{\Sigma}p+1)} + \frac{k_{33\text{ш}}k_{\Sigma} + c(T_{\Gamma}p+1)}{(T_{\Gamma}p+1) \cdot (T_{\Sigma}p+1)cT_{\Sigma}p}} = \\
&= \frac{k_{33\text{ш}}k_{\Sigma}R_{\Sigma} + cR_{\Sigma}(T_{\Gamma}p+1)}{R_{\Sigma}cT_{\Sigma}p(T_{\Gamma}p+1)(T_{\Sigma}p+1) + k_{\Sigma}T_{33\Gamma}cT_{\Sigma}p^2 + k_{33\text{ш}}k_{\Sigma}R_{\Sigma} + R_{\Sigma}c(T_{\Gamma}p+1)}.
\end{aligned}$$

Після ділення чисельника та знаменника на c і R_{Σ} маємо :

$$\begin{aligned}
W_{IV}(p) &= \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)} = \frac{k_{33\text{ш}}k_{\Sigma}k_{\text{д}} + T_{\Gamma}p + 1}{T_{\Sigma}p(T_{\Gamma}p+1) \cdot (T_{\Sigma} \cdot p + 1) + \frac{k_{\Sigma}T_{33\Gamma}T_{\Sigma}p^2}{R_{\Sigma}} + k_{\text{oc}}k_{\Sigma}k_{\text{д}} + T_{\Gamma}p + 1} = \\
&= \frac{k_{33\text{ш}}k_{\Sigma}k_{\text{д}} + T_{\Gamma}p + 1}{T_{\Gamma}T_{\Sigma}T_{\Sigma}p^3 + \left(T_{\Sigma}T_{\Sigma} + T_{\Gamma}T_{\Sigma} + \frac{k_{\Sigma}T_{\Sigma}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\Sigma})p + k_{\Sigma}k_{33\text{ш}}k_{\text{д}}}.
\end{aligned}$$

Визначимо числові значення параметрів, що входять до складу передавальних функцій.

Постійна часу генератора.

Під час розрахунку постійної часу генератора T_{Γ} будемо вважати, що вона практично збігається з постійною часу його незалежної обмотки збудження. Це припущення коректне за умови відсутності паралельної та послідовної обмоток і незначного впливу вихрових струмів. Таким чином,

$$T_{\Gamma} = T_{36\Gamma} = \frac{2p_{\Gamma} W_{36} k_{\Phi}}{R_{36}},$$

де W_{36} , R_{36} – кількість витків на полюс та активний опір незалежної обмотки збудження;

p_{Γ} – число пар полюсів генератора;

k_{Φ} – коефіцієнт нахилу лінійної ділянки кривої намагнічування, який для спрощення приймаємо $k_{\Phi} = \Delta\Phi / \Delta I_{36} = 0,29 \cdot 10^{-2} \text{ Ом} \cdot \text{с}$.

$$T_{\Gamma} = T_{36\Gamma} = \frac{4 \cdot 850 \cdot 0,49 \cdot 10^{-2}}{1,36 \cdot 3,29} = 0,37 \text{ с}.$$

Сумарна індуктивність якірного кола системи Г–Д

$$L_{\Sigma} = L_{\text{яд}} + L_{\text{яГ}},$$

$L_{\text{яд}}$, $L_{\text{яГ}}$ – індуктивності якорів двигуна та генератора.

Ці величини можна оцінити за наближеною формулою:

$$L_{\text{яд}(\Gamma)} = \gamma \frac{U_{\text{нд}(\Gamma)}}{p_{\text{д}(\Gamma)} \omega_{\text{нд}(\Gamma)} I_{\text{нд}(\Gamma)}},$$

γ – коефіцієнт, що береться рівним $\gamma = 0,25$ для компенсованої машини та $\gamma = 0,6$ для некомпенсованої

$U_{\text{нд}(\Gamma)}$, $I_{\text{нд}(\Gamma)}$, $\omega_{\text{нд}(\Gamma)}$ – номінальні значення напруги, струму та частоти обертання двигуна або генератора відповідно;

$p_{\text{д}(\Gamma)}$ – кількість пар полюсів відповідної машини.

Номінальна частота обертання генератора:

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

$$\omega_{\text{нГ}} = \frac{3,14 \cdot 1450}{30} = 151,84 \text{ с}^{-1}.$$

$$L_{яд} = 0,6 \cdot \frac{220}{2 \cdot 60,2 \cdot 192} = 5,7 \cdot 10^{-3} \text{ Гн},$$

$$L_{яг} = 0,6 \cdot \frac{460}{2 \cdot 151,84 \cdot 230} = 3,95 \cdot 10^{-3} \text{ Гн},$$

$$L_{я\Sigma} = (2 \cdot 5,7 + 3,95) \cdot 10^{-3} = 13,06 \cdot 10^{-3} \text{ Гн}.$$

Електромагнітна стала часу якірного кола системи Г–Д

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

$$T_{я\Sigma} = \frac{13,06 \cdot 10^{-3}}{0,17408} = 0,075 \text{ с}.$$

Електромеханічна постійна часу приводу T_m визначається наступним виразом:

$$T_m = \frac{J_{\Sigma} R_{я\Sigma}}{c^2},$$

де J_{Σ} – сумарний момент інерції якорів двигунів та приведений до їх швидкості момент інерції механізму

$$J_{\Sigma} = J'_{пл} + k \cdot J_d,$$

$J'_{пл}$ – момент інерції поворотної платформи, приведений до валу двигуна

$$J'_{пл} = \frac{J_{пл}}{i_{\Pi}^2},$$

$$J'_{пл} = \frac{88 \cdot 10^7}{10035^2} = 8,74 \text{ кг} \cdot \text{м}^2,$$

$$J_{\Sigma} = 2 \cdot 2 + 8,74 = 12,74 \text{ кг} \cdot \text{м}^2,$$

$$T_{\text{м}} = \frac{12,74 \cdot 0,17408}{(2 \cdot 3,42)^2} = 0,16 \text{ с}.$$

Далі обчислимо постійну часу $T_{\text{зт}}$, підставивши числові значення у передавальну функцію $W_I(p) = \frac{\omega_d(p)}{U_3(p)}$. У результаті дістанемо:

$$W_{\omega_d U_3}(p) = \frac{0,3501}{1,166 \cdot 10^{-4} p^3 + (3,249 \cdot 10^{-3} + 0,262 T_{\text{зт}}) p^2 + 0,1082 p + 2,028}.$$

Для визначення мінімально допустимого значення $T_{\text{зт min}}$, за якої система залишається стійкою, застосуємо критерій Гурвіца. Система описується рівнянням третього порядку, яке у загальному вигляді має форму:

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

$$\text{де } a_0 = 1,166 \cdot 10^{-4}; \quad a_1 = 3,294 \cdot 10^{-3} + 0,262 T_{\text{зт}}; \quad a_2 = 0,1082; \quad a_3 = 2,028$$

Відповідно до критерію Гурвіца, система третього порядку буде стійкою за умови, що всі коефіцієнти додатні і виконується умова:

$$a_1 \cdot a_2 - a_0 \cdot a_3 > 0.$$

$$(3,294 \cdot 10^{-3} + 0,262 T_{\text{зт}}) \cdot 0,1082 - 1,166 \cdot 10^{-4} \cdot 2,028 > 0,$$

$$1,199 \cdot 10^{-4} + 2,83 \cdot 10^{-2} T_{\text{зт}} > 0.$$

Оскільки отриманий вираз є додатним для будь-яких значень $T_{\text{зт}}$ та $T_{\text{зт}}$, то оптимальне обирається шляхом чисельного експерименту.

Розрахунок перехідних процесів проводився у MATLAB. Цей програмний комплекс дозволяє аналізувати динаміку систем за передавальними функціями, матричними рівняннями стану, а також за допомогою моделей, реалізованих у SIMULINK.

Для моделювання побудовано структурну модель системи, показану на рис. 2.15. Вона відповідає алгоритмічній схемі з рисунка 2.7.

На рис. 2.16 та 2.17 наведені графіки перехідних процесів швидкості ω_d та струму I_y як для задавального, так і для збурюючого впливів. Криві 1, 2, 3 відповідають значенням $T_{33T} = 0,005 \text{ с}$, $T_{33T} = 0,02 \text{ с}$, $T_{33T} = 0,05 \text{ с}$ відповідно. Аналіз показує, що при значенні $T_{33T} = 0,005 \text{ с}$ система демонструє найкращі показники якості перехідного процесу.

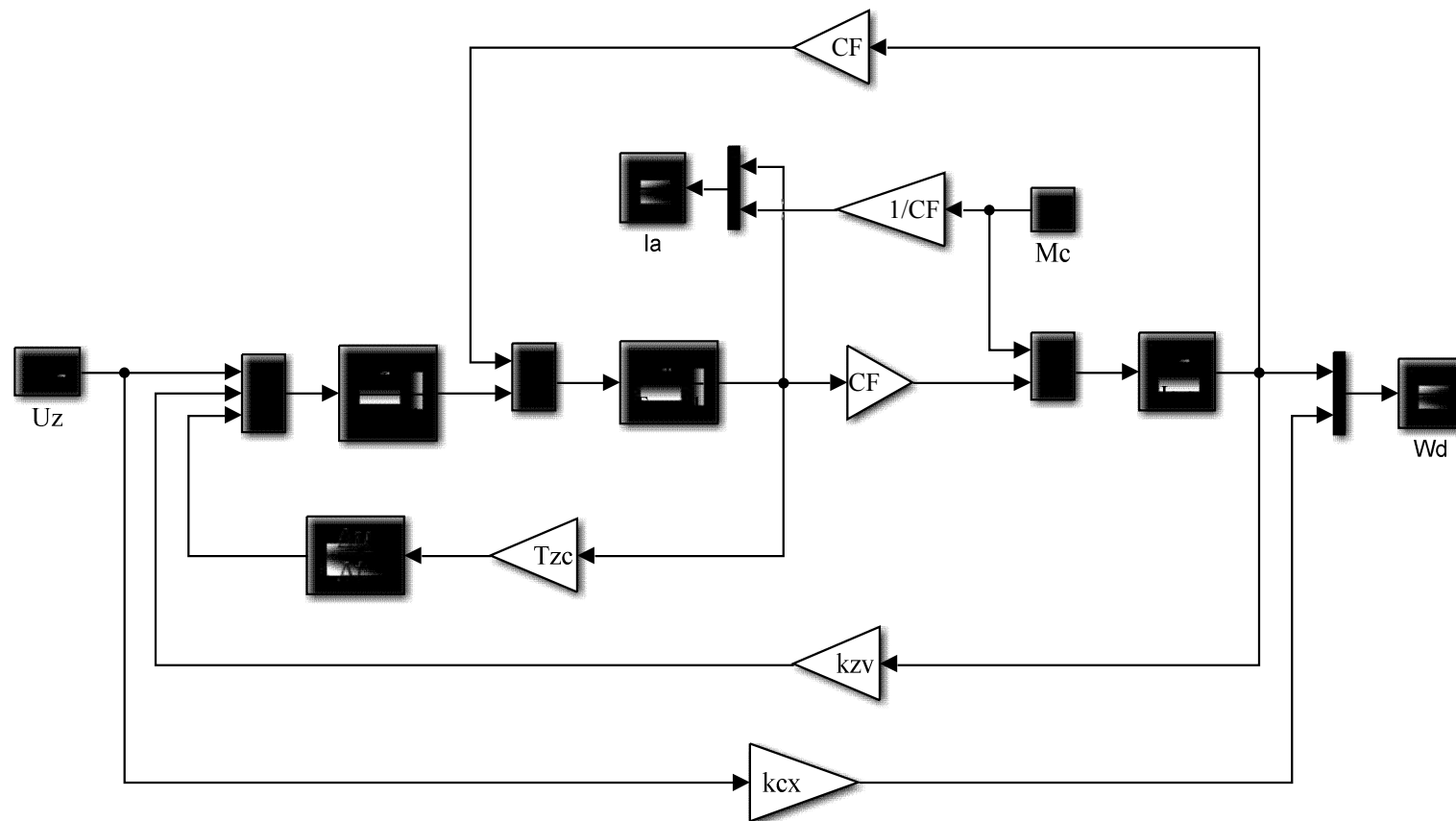


Рисунок 2.15 – Модель системи, реалізована у MATLAB

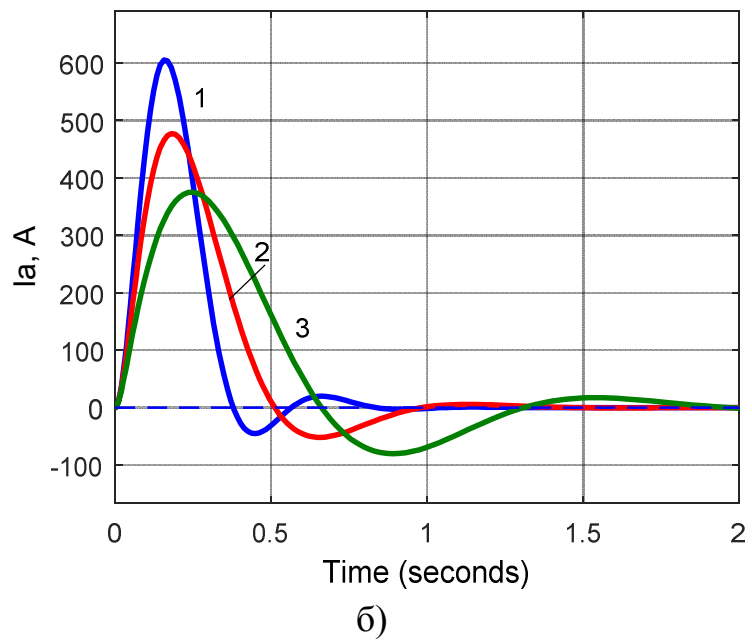
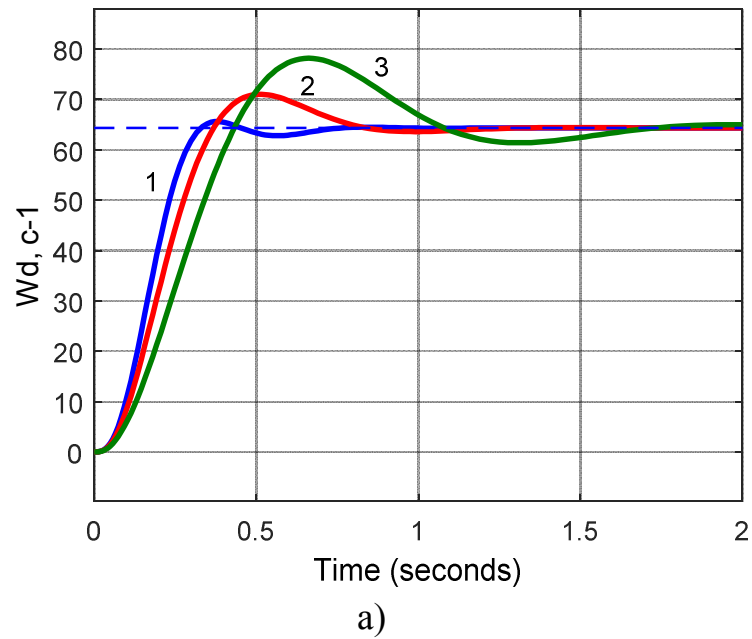


Рисунок 2.16 – Перехідні процеси швидкості $\omega_d = f(t)$ (а)
та струму $I_a = f(t)$ (б) при дії задавального сигналу
1 – $T_{33T} = 0,005 \text{ c}$; 2 – $T_{33T} = 0,02 \text{ c}$; 3 – $T_{33T} = 0,05 \text{ c}$

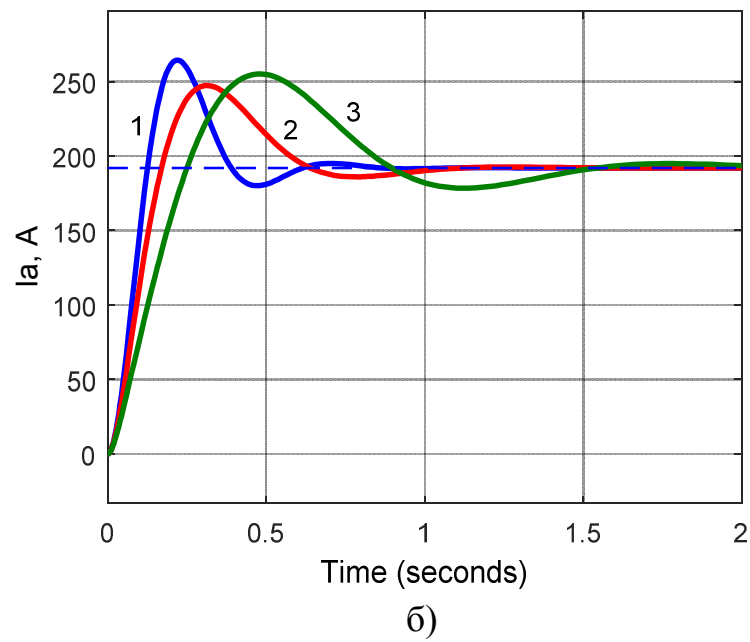
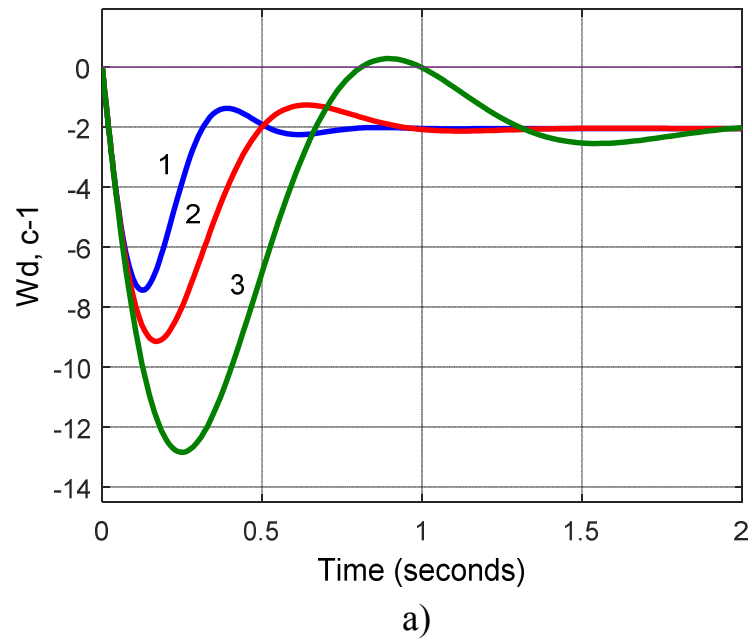


Рисунок 2.17 – Перехідні процеси швидкості $\omega_d = f(t)$ (а)

та струму $I_a = f(t)$ (б) при дії збурення

1 – $T_{33T} = 0,005 \text{ c}$; 2 – $T_{33T} = 0,02 \text{ c}$; 3 – $T_{33T} = 0,05 \text{ c}$

ВИСНОВКИ ДО РОЗДІЛУ 2

1. У ході роботи здійснено підбір електродвигуна та основних елементів силового електрообладнання приводу. Проведено перевірку відповідності вибраного двигуна тепловим режимам і вимогам до перевантажувальної здатності, для чого побудовано швидкісну та навантажувальну діаграми. Розрахований еквівалентний момент підтвердив, що двигун працездатний у допустимих умовах за нагрівом, а також відповідає нормативам щодо допустимого перевантаження.

2. Розраховано параметри силової частини електроприводу, необхідні для подальшого синтезу системи автоматичного регулювання.

3. Обґрунтовано та вибрано структуру системи керування. Оскільки для електроприводів головних вузлів роторних екскаваторів характерна підвищена вимога до жорсткості механічної характеристики та обмеження максимального значення розвиваного моменту, рішенням стало застосування системи з зворотним зв'язком за швидкістю, гнучким струмовим зворотним зв'язком, а також контуром обмеження струму (з відсіченням).

4. Виконано аналіз і розрахунок статичних характеристик як розімкненої, так і замкненої системи керування. Побудовано алгоритмічну модель, за якою визначені передавальні функції системи для задавальних і збурюючих впливів.

5. Проведено імітаційне моделювання синтезованої САУ в середовищі MATLAB із застосуванням інструментів SIMULINK. Побудована модель дала можливість дослідити динамічні властивості системи. У процесі моделювання підібрано таке значення постійної часу струмового зворотного зв'язку, за якого перехідні процеси системи відповідають вимогам щодо якості.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЛАНОК

3.1 Математична модель двомасової системи

Для електроприводу, який забезпечує поворот стріли екскаватора, характерною є наявність декількох масивних елементів, інерція яких передається до поворотної платформи через систему пружних з'єднань. До таких елементів належать маса стріли, ротор разом з робочим колесом, а також противага, що розміщена на консолі. Усі ці складові не є жорстко інтегрованими у конструкцію, а з'єднані з платформою через еластичні механізми або гнучкі ланки. Через це реальний рух системи супроводжується значною кількістю пружних деформацій, що, у свою чергу, породжує коливання під час будь-яких перехідних режимів електроприводу.

У попередніх розділах аналіз динамічних характеристик механізму повороту стріли виконувався на основі припущення, що всі механічні зв'язки між його елементами є абсолютно жорсткими. Такий підхід дозволяє суттєво спростити модель, зводячи всю механічну частину до однієї приведенної маси. Однак подібне спрощення відображає реальну поведінку системи лише наближено та не дозволяє оцінити вплив еластичності окремих ланок. У реальних умовах жорсткість конструктивних елементів завжди є скінченною, тому механічна частина електроприводу функціонує як пружна система. Унаслідок цього прикладання керуючого моменту двигуна або поява зовнішнього моменту навантаження викликають пружні коливання, що призводять до збільшення динамічних навантажень у з'єднаннях і можуть істотно впливати на точність позиціонування та стабільність руху.

Кінематична структура механізму повороту стріли є складною просторовою системою, яка включає значну кількість зосереджених і розподілених мас, з'єднаних між собою гнучкими або пружними елементами різного типу. У практиці інженерного моделювання подібні об'єкти зазвичай спрощують.

Найчастіше використовують тримасові або двомасові еквівалентні схеми, де маси зосереджуються в окремих точках, а зв'язки між ними розглядаються як пружні та невагомі. Такий підхід дозволяє зберегти ключові властивості системи, але при цьому значно полегшує математичний опис та подальші динамічні розрахунки.

Електропривід механізму повороту стріли, що розглядається в даній роботі, приводиться в рух двома синхронно працюючими двигунами. Наявність двох приводних машин формує так звану багаторухову структуру електроприводу. У загальному випадку така система має розгалужену механічну модель, яка враховує індивідуальні параметри кожного двигуна та характер їх взаємодії через еластичні елементи. Розрахункова схема цього типу електроприводу наведена на рис. 3.1 і використовується як початкова для подальшого спрощення та зведення до двомасової моделі.

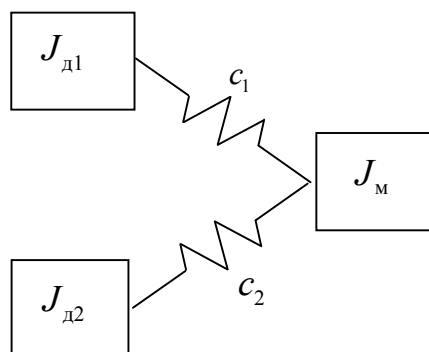


Рисунок 3.1 – Розгалужена модель механічної частини електроприводу

Позначення на схемі:

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

c_1 , c_2 – приведені коефіцієнти жорсткості ланок кожного приводу;

$J_{м}$ – приведений до валу двигуна момент інерції поворотної платформи

$$J_{м} = J'_{пл} = 20 \text{ кг} \cdot \text{м}^2 ;$$

Оскільки обидва двигуни працюють синхронно, то схему, наведену на рис. 3.1, можна спростити, зводячи її до двомасової моделі, зображеної на рис. 3.2.

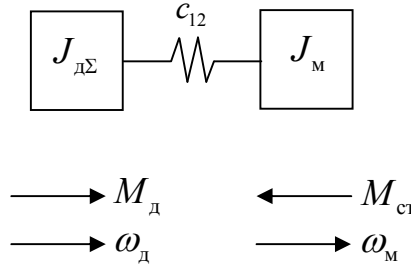


Рисунок 3.2 – Двомасова спрощена модель

У цій моделі $J_{дΣ}$ є сумарним моментом інерції обох двигунів. Оскільки привід використовує два ідентичні електродвигуни, то їх моменти інерції рівні, а отже $J_{дΣ} = 2J_{д}$; c_{12} представляє собою приведену жорсткість зв'язку між масами, яка визначається наступним чином: $c_{12} = c_1 + c_2$.

Якщо перенести всі рухомі маси та жорсткі елементи електроприводу механізму повороту стріли до валу двигуна, то для двомасової системи можна записати наступну систему диференціальних рівнянь:

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

де сумарний момент, що передаються пружною передачею M_{Σ} , дорівнює сумі пружного моменту $M_{пр}$ і моменту в'язкого тертя $M_{вт} = \beta(\omega_{д} - \omega_{м})$:

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

β – коефіцієнт внутрішнього в'язкого тертя між масами.

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

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

Систему (3.1) із урахуванням (3.2) можна подати у вигляді задачі Коші.

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

Диференціальні рівняння, що описують електромеханічні процеси в одномасовій системі, розробленій у попередніх розділах, мають наступний вигляд:

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

Систему (3.4) також запишемо у формі Коші, беручи до уваги, що $I_{\text{ст}} = M_{\text{ст}} / c$, $T_{\text{м}} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}$. Об'єднавши системи (3.3) і (3.4) та виконавши необхідні перетворення, отримаємо кінцеву систему рівнянь стану двомасової системи.

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

Введемо вектор стану системи

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

і вектор зовнішніх впливів

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

Система у векторно-матричній формі, доповнена рівнянням виходу, набуває вигляду:

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

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

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

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

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

Алгоритмічна схема двомасової системи наведена на рис. 3.3.

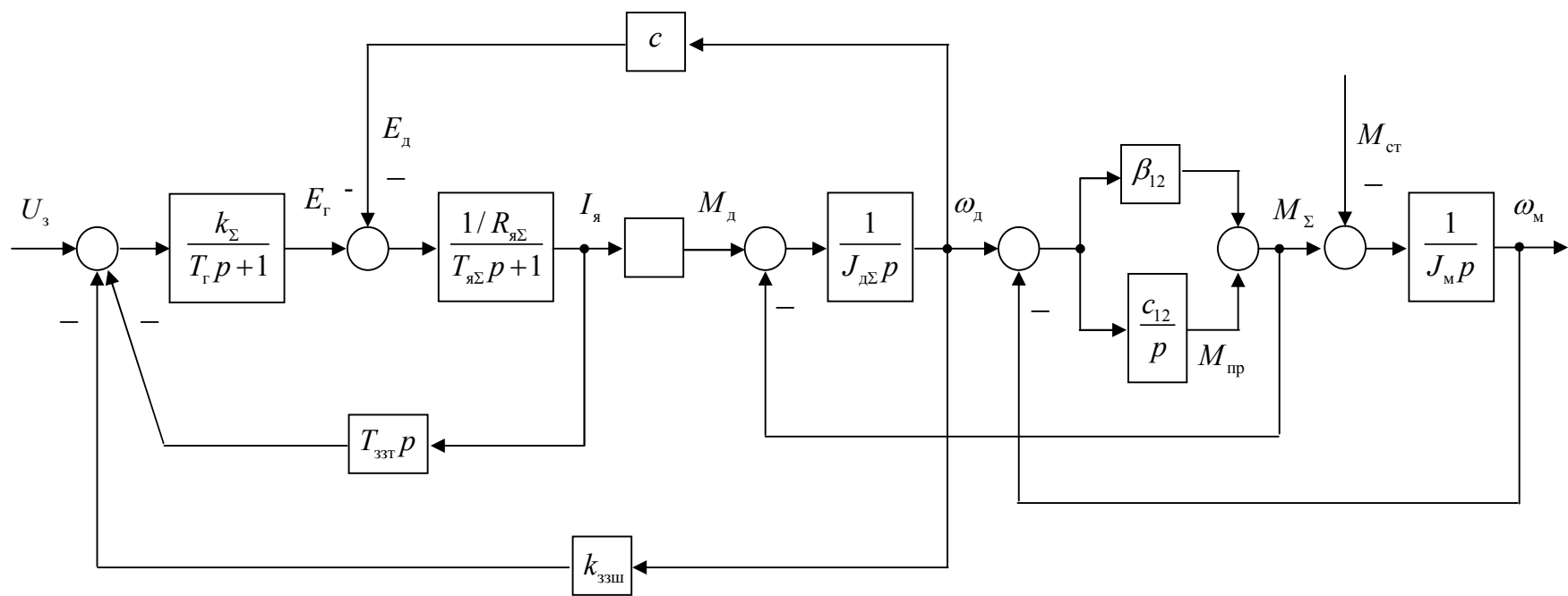


Рисунок 3.3 – Алгоритмічна схема двомасової системи

3.2 Аналіз динамічних характеристик двомасової моделі

Для оцінювання поведінки двомасової системи було використано її математичний опис у формі рівнянь стану, на основі яких сформовано залежності перехідних процесів у відповідь як на задавальний вплив, так і на дію збурень. Усі обчислювальні операції виконувалися на персональній ЕОМ із застосуванням пакета технічних обчислень MATLAB.

У додатку Д наведено файл, що містить початкові параметри системи, включаючи матриці **A**, **B**, **C** і **D**. Для моделювання часових перехідних процесів використовувалась схема на базі середовища SIMULINK, яка містить блок State-Space (див. рис. 3.4). У цьому блоці задаються імена матриць, що відповідають математичному опису двомасової моделі.

Кількість виведених координат, тобто розмірність вектора виходу $\vec{Y}(t)$, а відповідно й число вікон Score, де відображаються результати моделювання, визначається безпосередньо під час проведення експерименту в SIMULINK. При подвійній активації блоку State-Space відкривається вікно для задання матриць **A**, **B**, **C** і **D**., зображене на рис. 3.5.

Матриця виходу CCC приймає вигляд одиничної матриці у випадках, коли необхідно отримати графіки для всіх компонент вектора станів. Якщо ж аналізують лише частину змінних, одиничні елементи на головній діагоналі цієї матриці розташовують вибірково – у рядках, що відповідають потрібним координатам. Матриця **D** при цьому залишається нульовою.

Окрім моделі на базі блоку State-Space, дослідження системи проводили й за допомогою іншої структури Simulink-моделі, наведеної на рис. 3.6.

Графічні результати імітації перехідних процесів представлені на рис. 3.7–3.10. На цих рисунках подано часові залежності перших чотирьох змінних стану. Аналіз отриманих кривих демонструє наявність істотних коливальних компонент у характеристиках основних координат системи. Подібні коливання небажані, оскільки вони збільшують динамічні навантаження на ме-

ханічні елементи привода під час перехідних режимів, негативно впливаючи на точність роботи механізму та його довговічність.

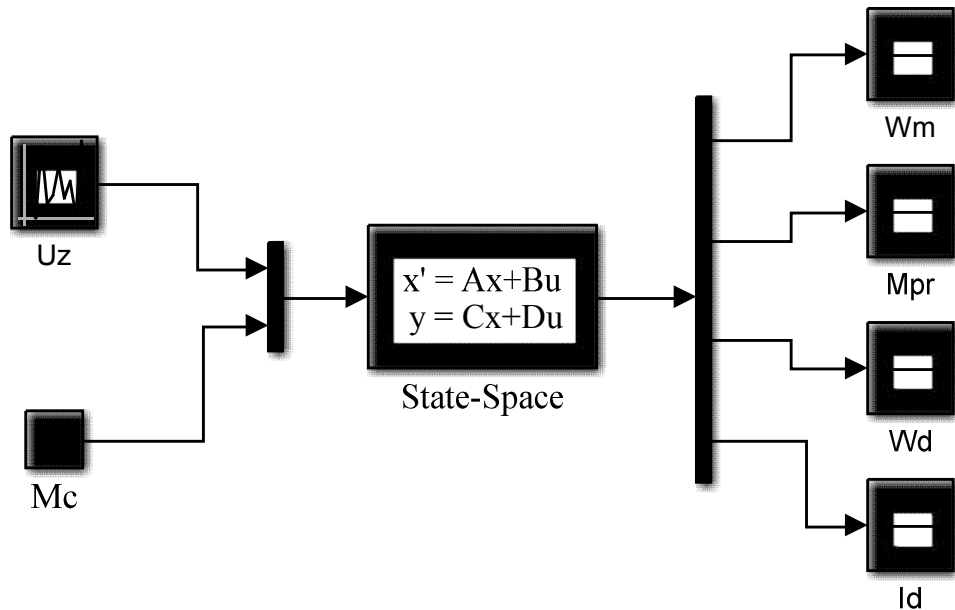


Рисунок 3.4 – Модель системи, реалізована з використанням блоку State-Space

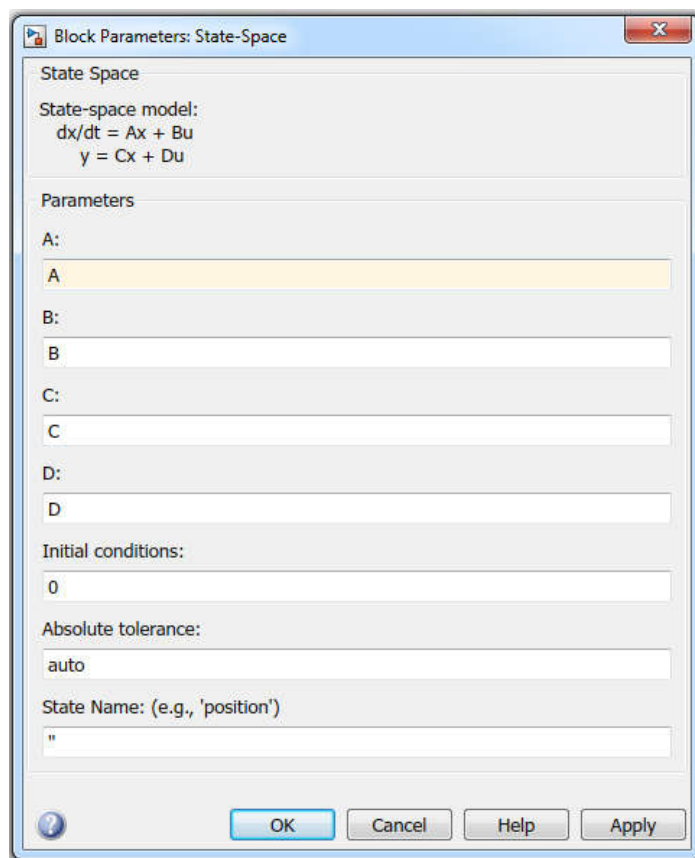


Рисунок 3.5 Вікно налаштування матриць **A**, **B**, **C**, **D** у блоці State-Space

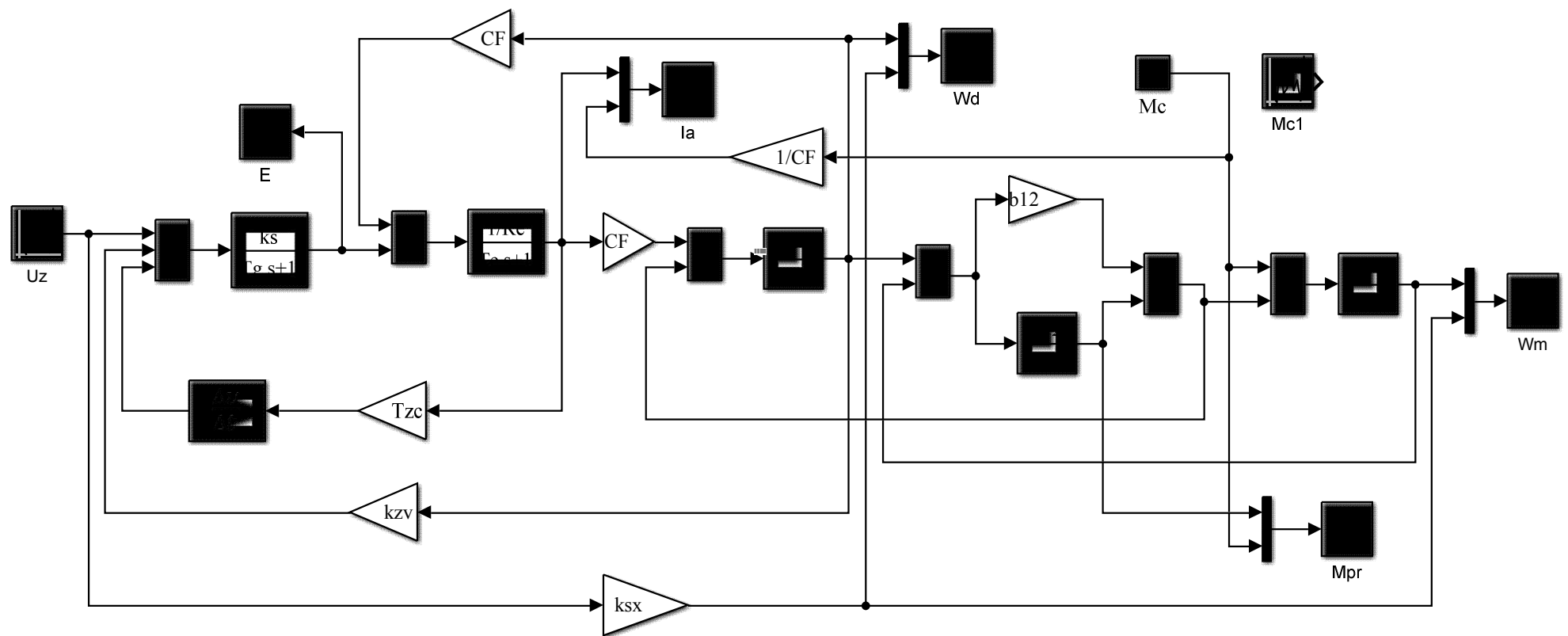


Рисунок 3.6 – Модель двомасової системи, створена в SIMULINK

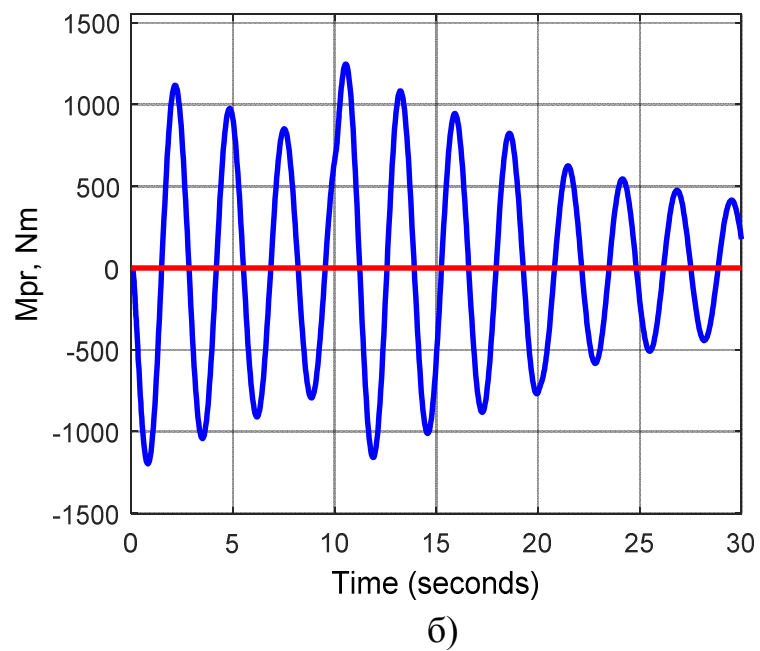
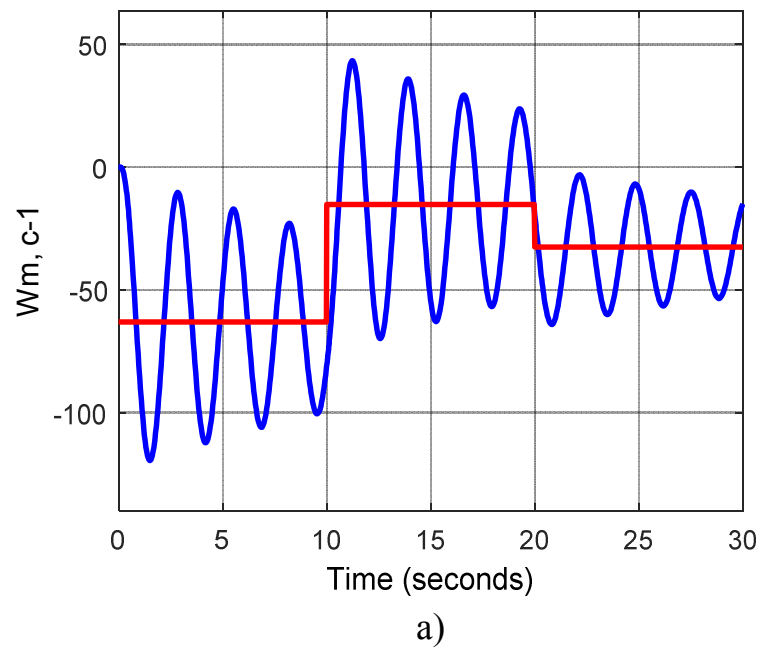


Рисунок 3.7 – Перехідні процеси механічних параметрів двомасової системи під дією задаючої величини:

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

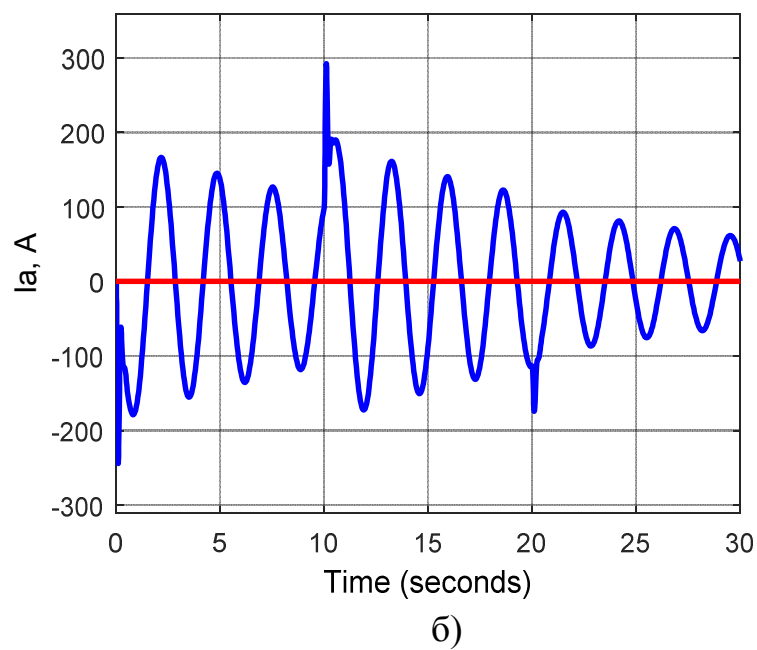
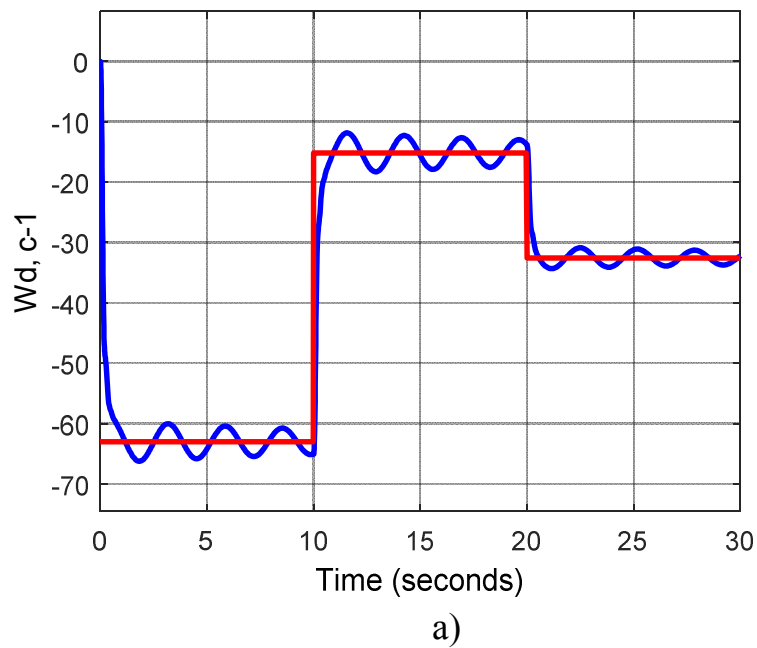
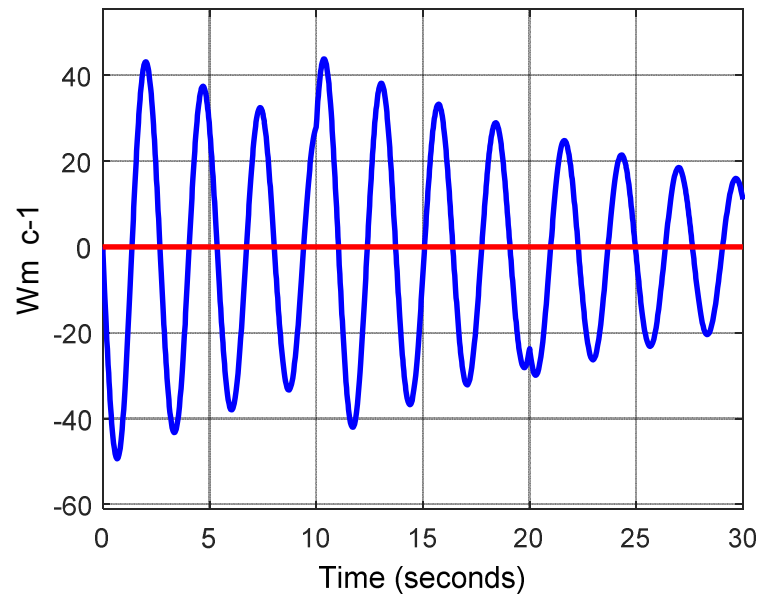
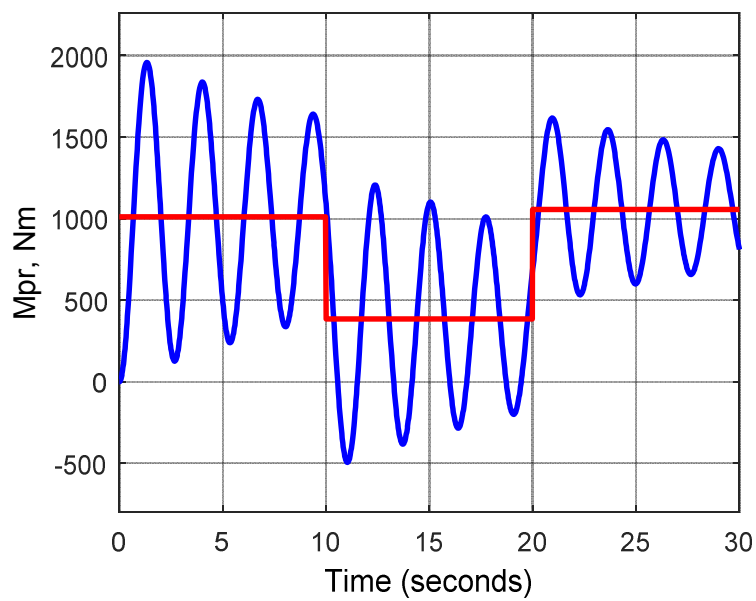


Рисунок 3.8 – Динаміка двигуна двомасової системи
під впливом задаючої величини:

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



a)



б)

Рисунок 3.9 – Перехідні процеси механічних параметрів двомасової системи під дією зовнішнього збурення:

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

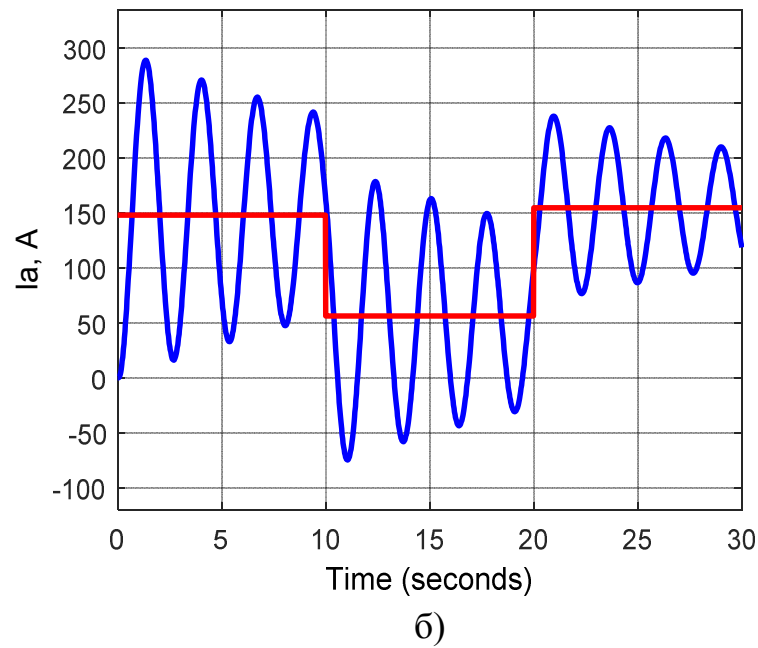
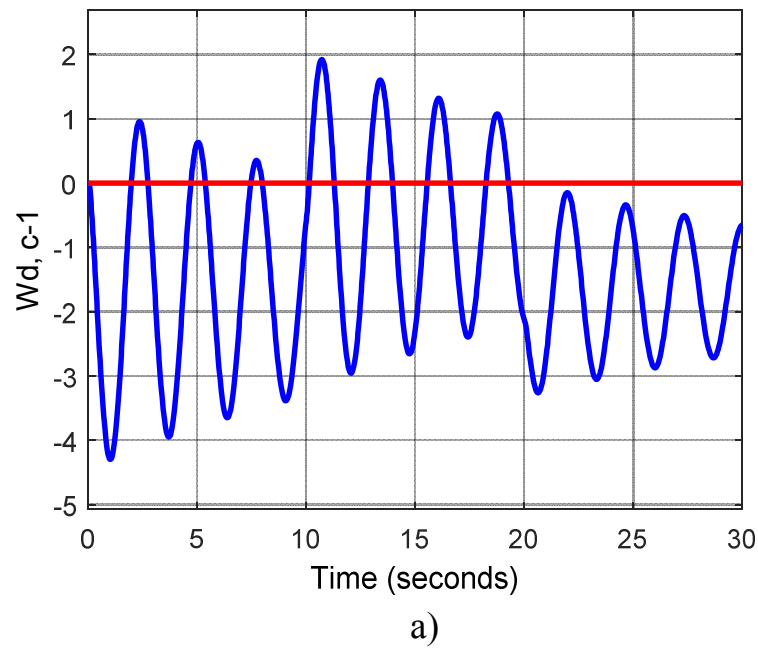


Рисунок 3.10 – Динаміка двигуна двомасової системи
під дією зовнішнього збурення:

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

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Виконано дослідження роботи механізму повороту стріли роторного екскаватора з урахуванням наявності пружно зв'язаних з платформою інерційних мас стріли ротора, роторного колеса та консолі противаги. Моделювання механічної частини як жорсткої приведенної ланки дозволяє відтворити лише середнє положення руху елементів, оскільки реальна пружність механічних зв'язків при цьому ігнорується. Через кінцеву жорсткість цих зв'язків електропривод утворює пружну систему, до якої прикладення управляючого моменту або зовнішнього обурення (моменту статичного навантаження) призводить до коливань пов'язаних мас. Це, в свою чергу, збільшує пікові навантаження на зв'язки та ускладнює точність виконання необхідних переміщень і рухів. Показано, що кінематична структура механічної частини системи управління при врахуванні гнучкості елементів може бути адекватно представлена двомасовою моделлю.

2. Розроблено математичну модель двомасової системи. Отримано рівняння стану в класичній та векторно-матричній формах, визначено елементи вектора стану, що відповідають змінним стану системи, сформований вектор зовнішніх впливів, а також матриці стану **A** і управління **B**. Створено алгоритмічну схему моделі двомасової системи.

3. Проведено комп'ютерне моделювання двомасової системи з використанням програмного пакету MATLAB. У середовищі SIMULINK розроблено модель системи, на вхід якої подавався ступінчастий сигнал із випадковою амплітудою. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи мають коливальний характер, що є ознакою незадовільної стабільності у динаміці.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

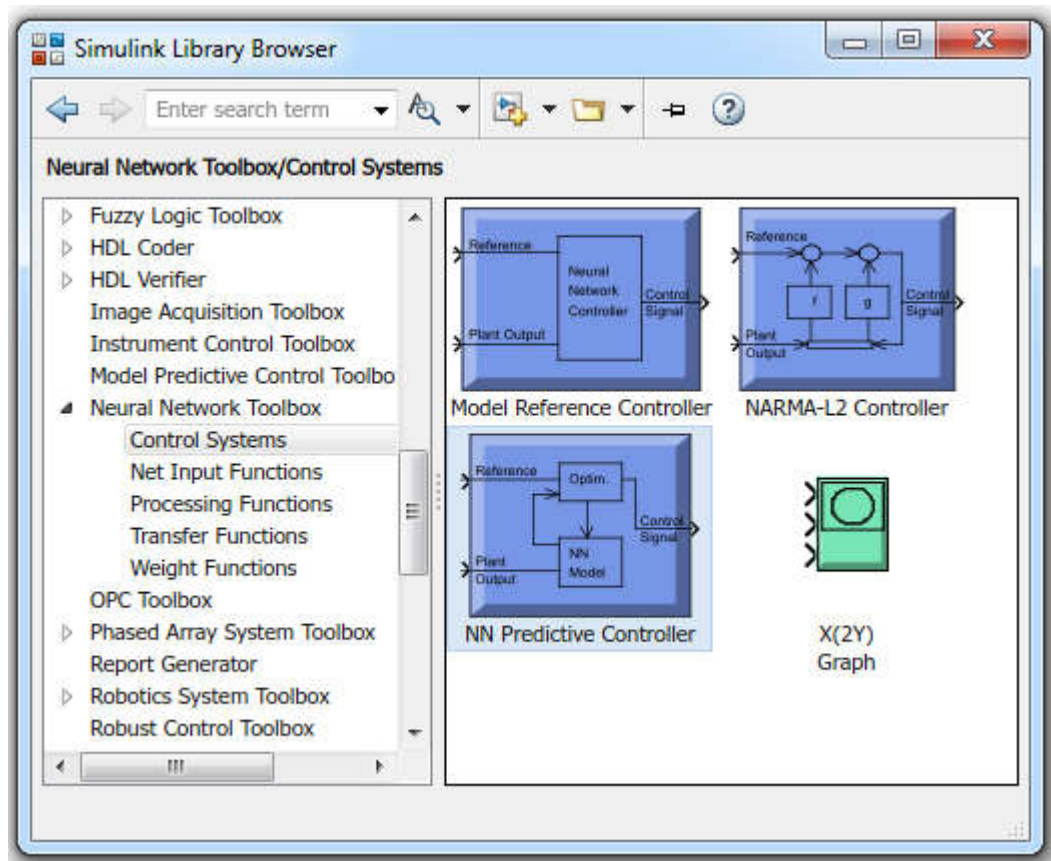


Рисунок 4.1 – Набір блоків Neural Network Blocksets підсистеми Control Systems середовища SIMULINK

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

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

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

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

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

4.2 Принцип формування прогнозуючого регулятора NN Predictive Controller

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

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

4.2.1 Процедура ідентифікації об'єкта керування

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

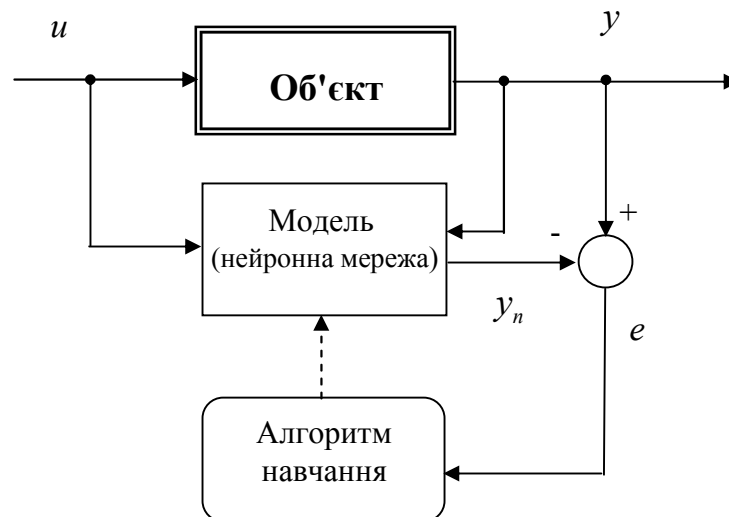


Рисунок 4.2 – Структурна схема підсистеми ідентифікації

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

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

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

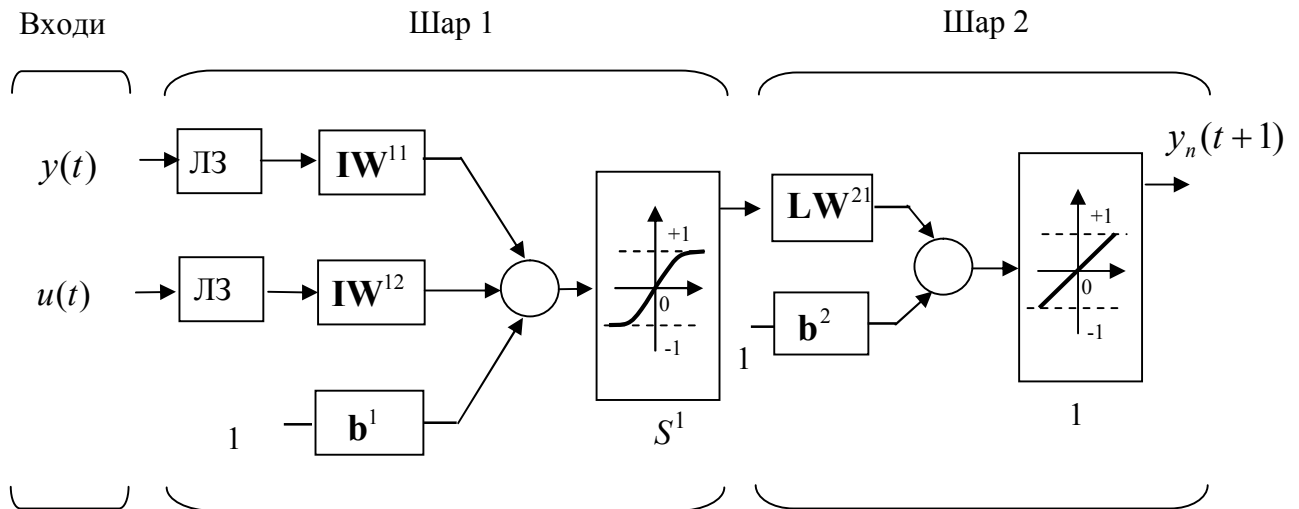


Рисунок 4.3 – Структура нейронної мережі, що моделює об’єкт керування

4.2.2 Принцип функціонування прогнозуючої системи керування

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

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

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

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

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

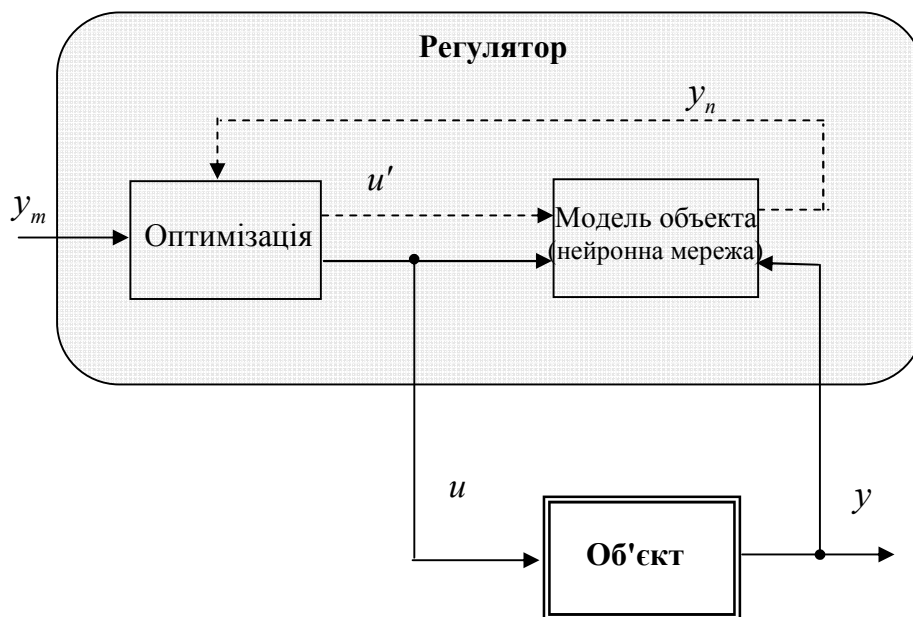


Рисунок 4.4 – Узагальнена схема керування з використанням прогнозуючого регулятора

4.3 Проектування прогнозуючого регулятора NN Predictive Controller у середовищі MATLAB

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

1. Функції одновимірної оптимізації

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

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

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

2. Функції, що реалізують механізм керування з прогнозуванням

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

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

3. Моделі та графічні додатки Simulink

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

4. Допоміжні програмні модулі

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

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

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

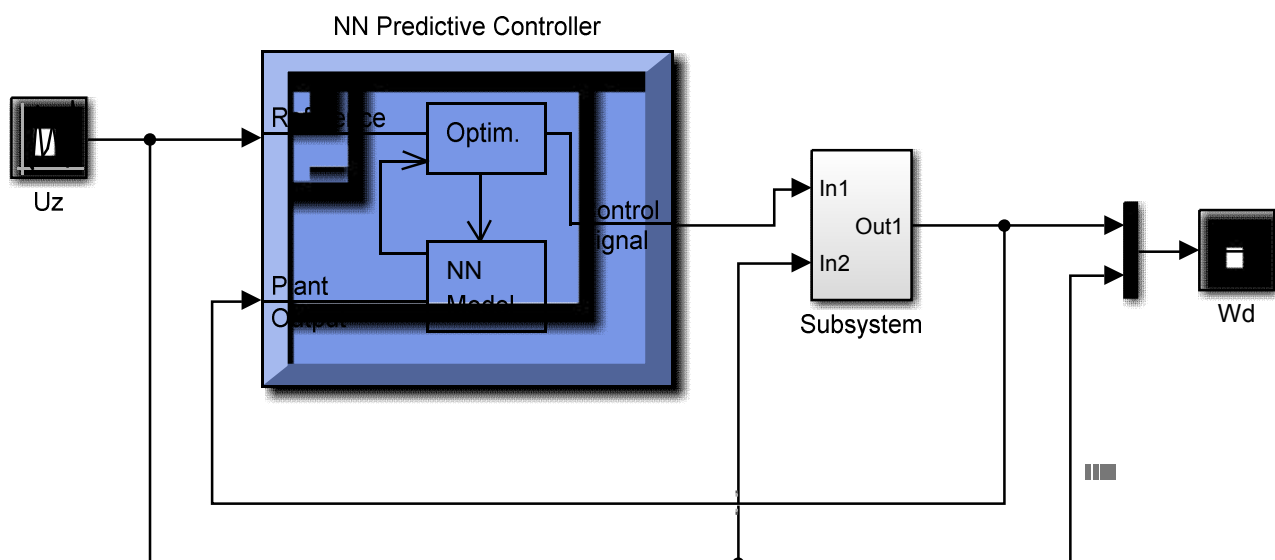


Рисунок 4.5 – Структурна модель системи керування з нейрорегулятором NN Predictive Controller

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

Структура внутрішньої організації нейромережевого регулятора NN

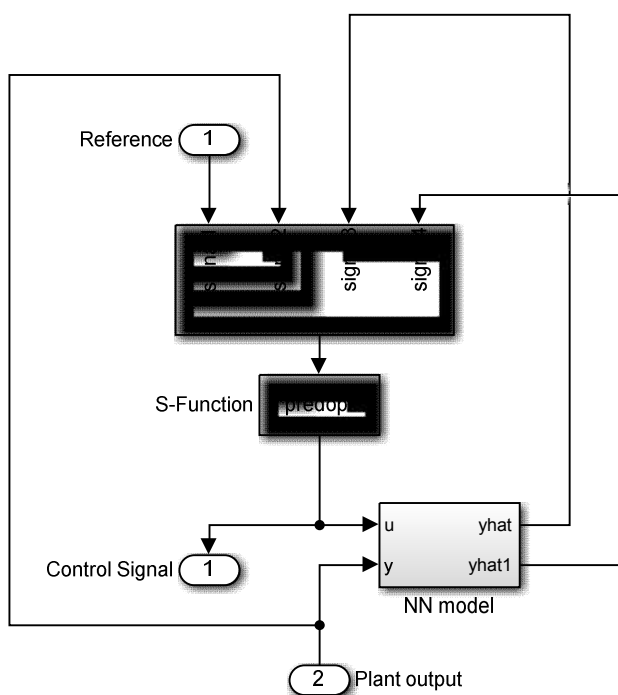


Рисунок 4.6 – Структурна схема нейрорегулятора

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

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

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

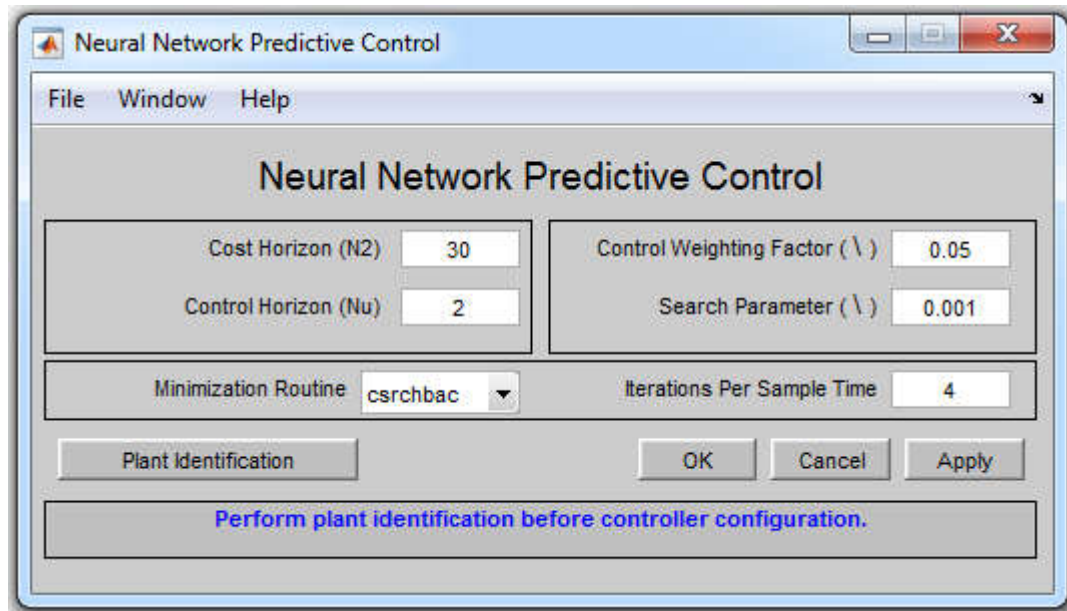


Рисунок 4.7 – Інтерфейс налаштування параметрів регулятора

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.

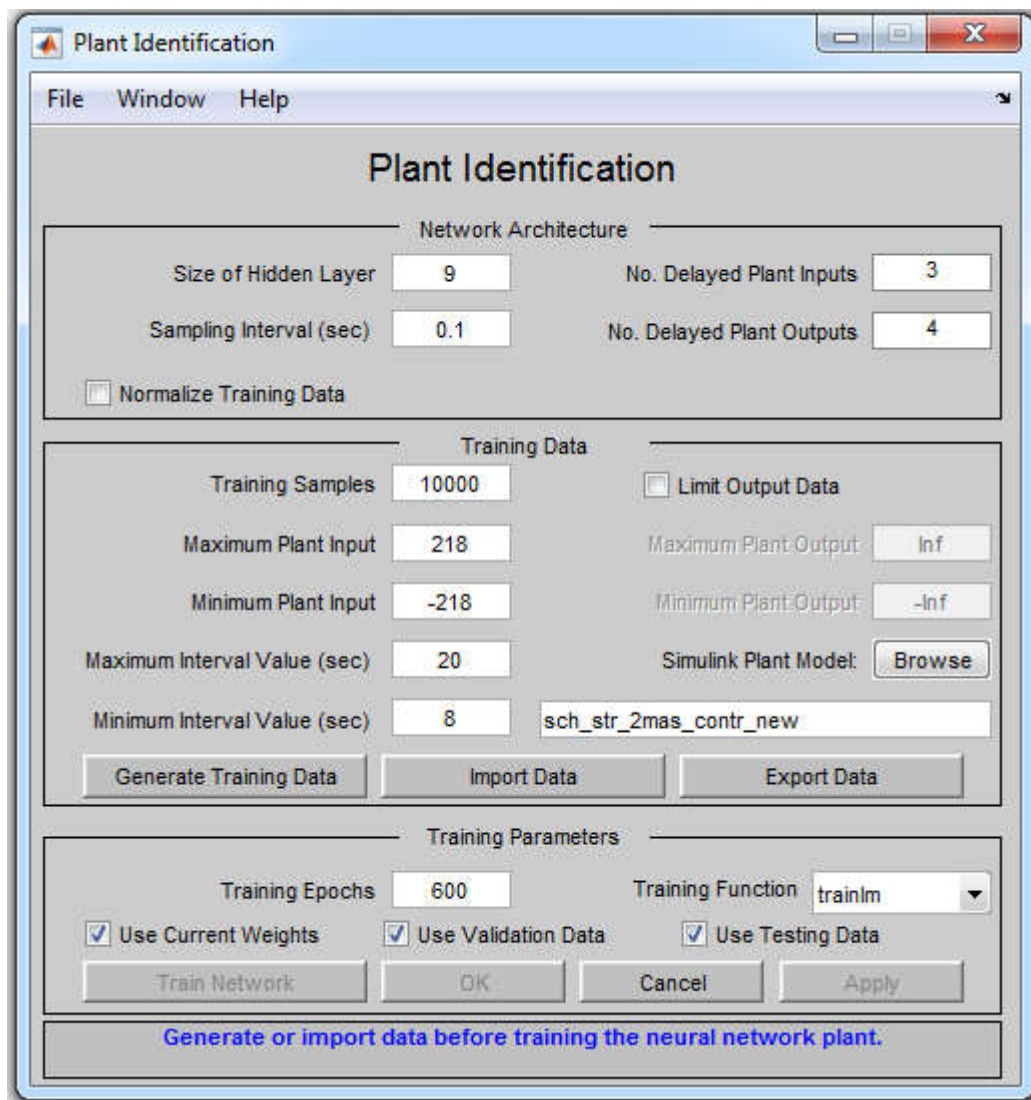


Рисунок 4.8 – Вікно процедури ідентифікації об'єкта керування

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.



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

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

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

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

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

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

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

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

- `traingd` – градієнтний спуск;
- `traingdm` – градієнтний спуск з моментом;
- `trainlm` – метод найменших квадратів Левенберга-Марквардта;
- `trainbr` – байєсівський регулятор.

Кожна з цих функцій має свої особливості та параметри, які можна налаштовувати для досягнення максимальної точності моделі та здатності адекватно відтворювати динаміку системи. Training Function визначає, як нейромережа коригує внутрішні ваги та параметри під час навчання для підвищення якості прогнозу та стабільності моделі.

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

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

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

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

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

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

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

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

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

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

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

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

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

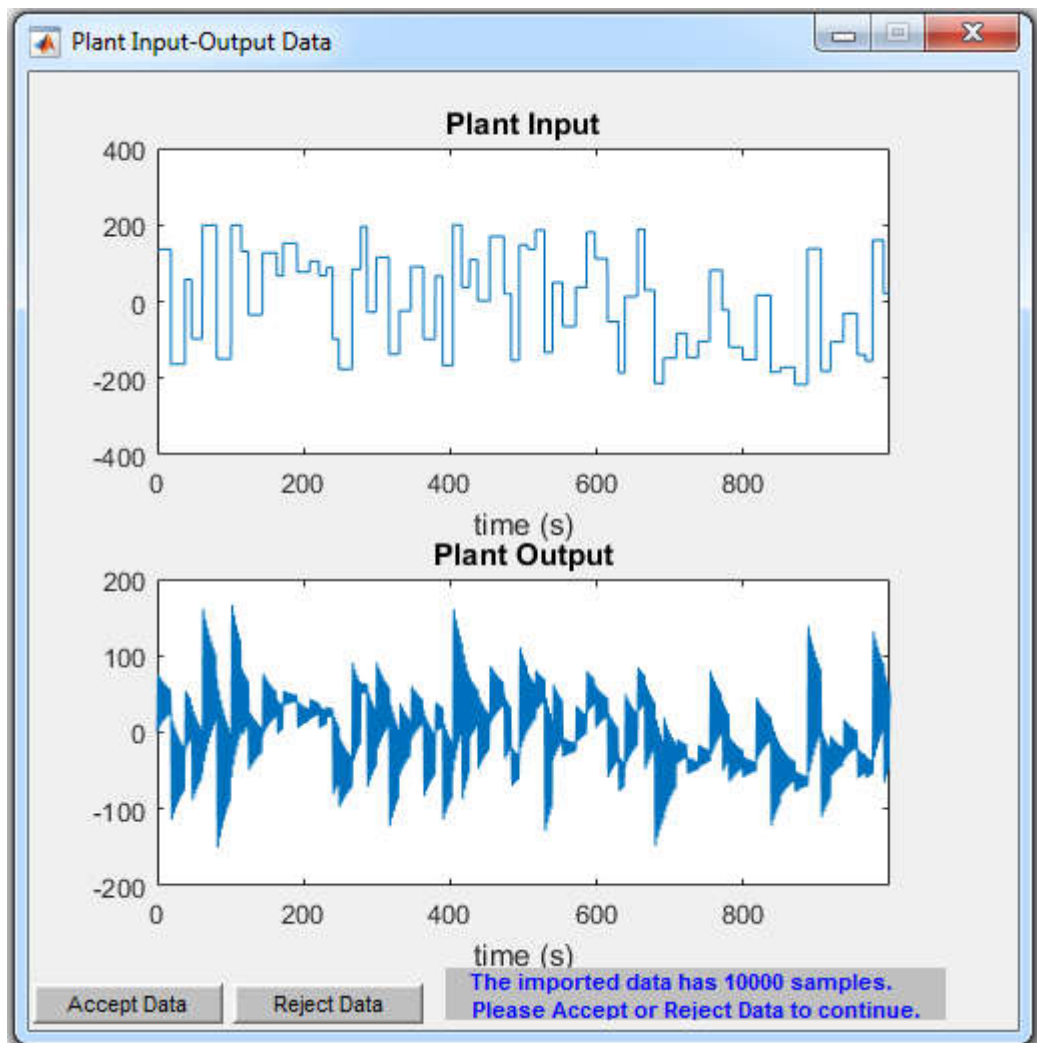


Рисунок 4.10 – Графіки вхідного та вихідного сигналів під час генерації навчальної послідовності

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

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

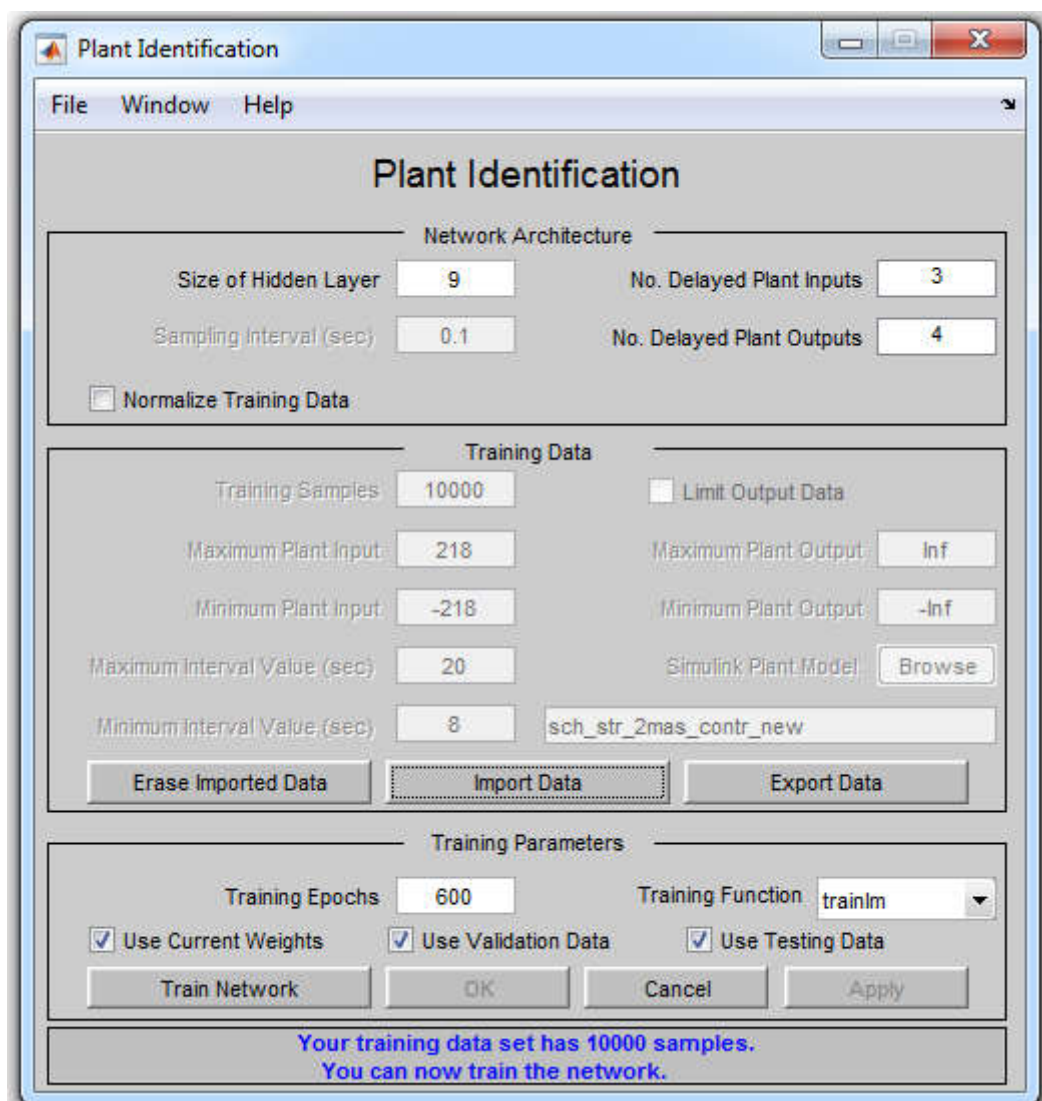


Рисунок 4.11 – Вікно Plant Identification після генерації навчальної послідовності

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

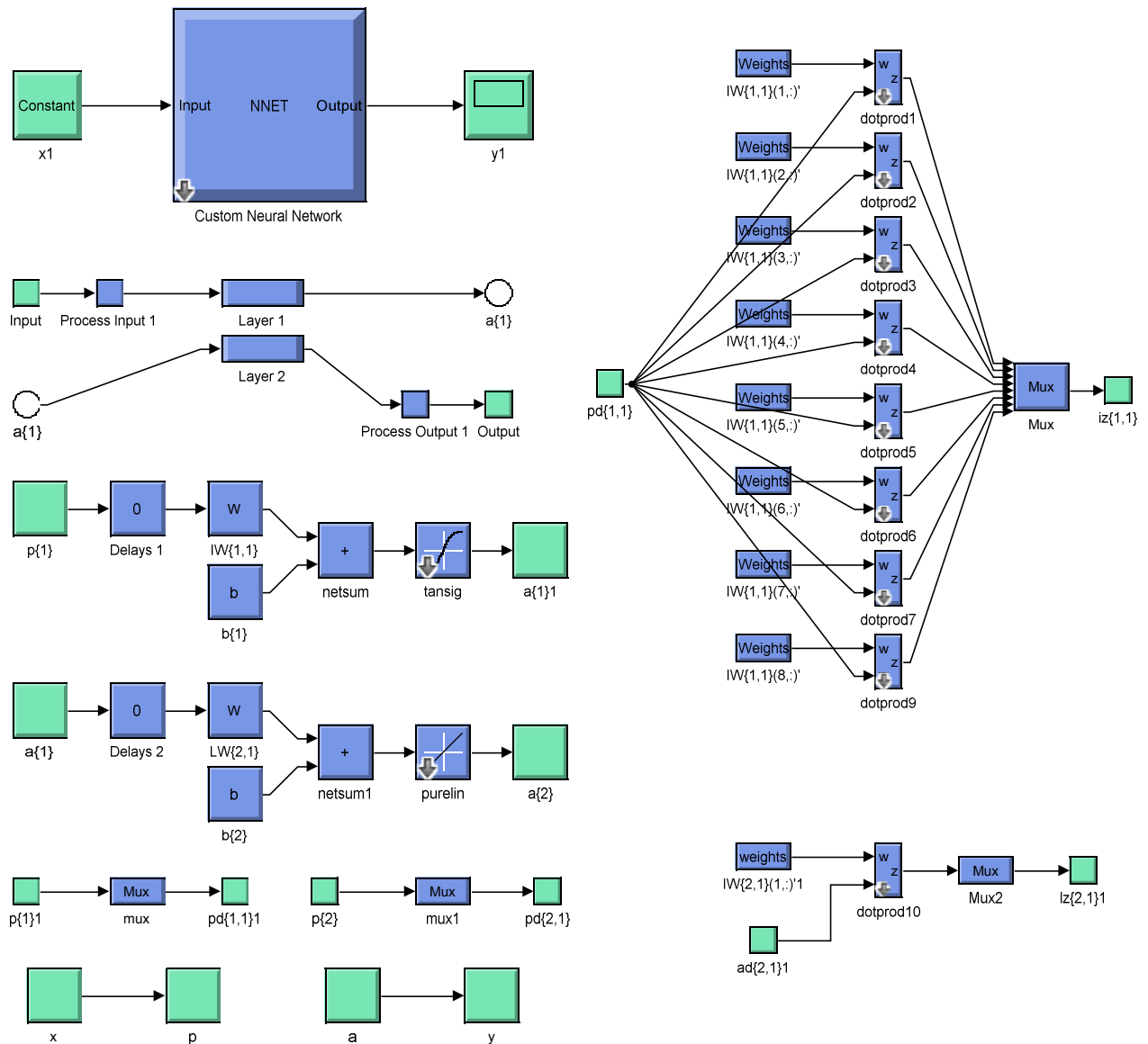


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

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

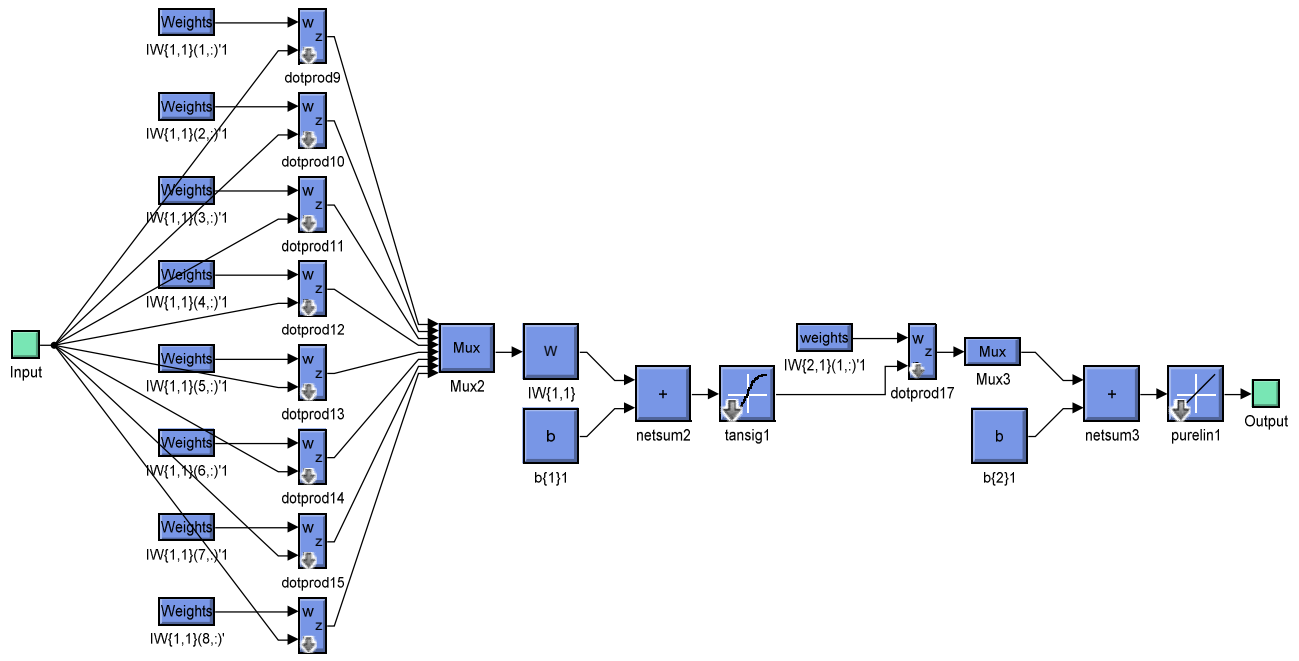


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

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

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

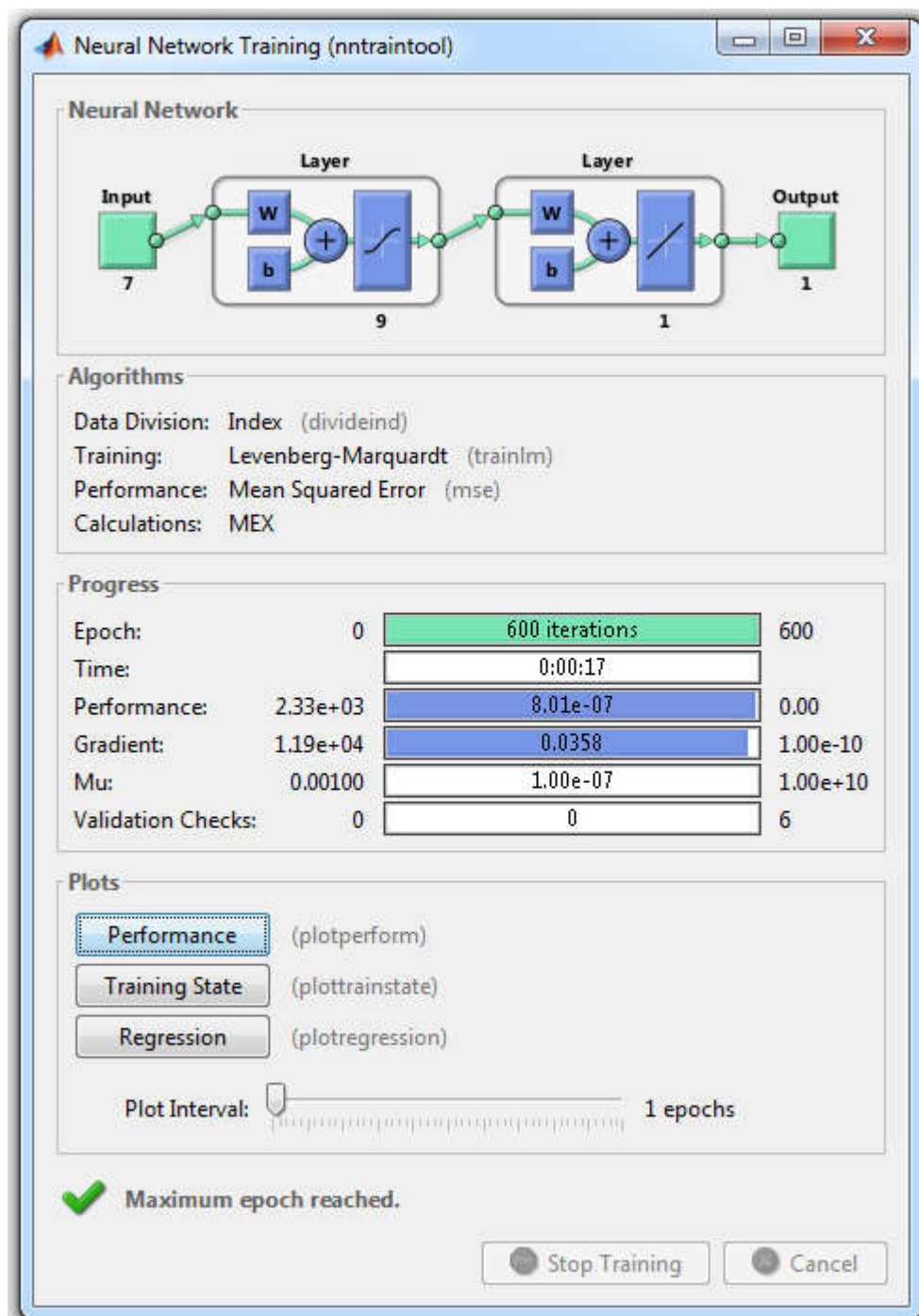


Рисунок 4.14 - Головне вікно для відображення процесу та результатів навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

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

Математично MSE можна записати так:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

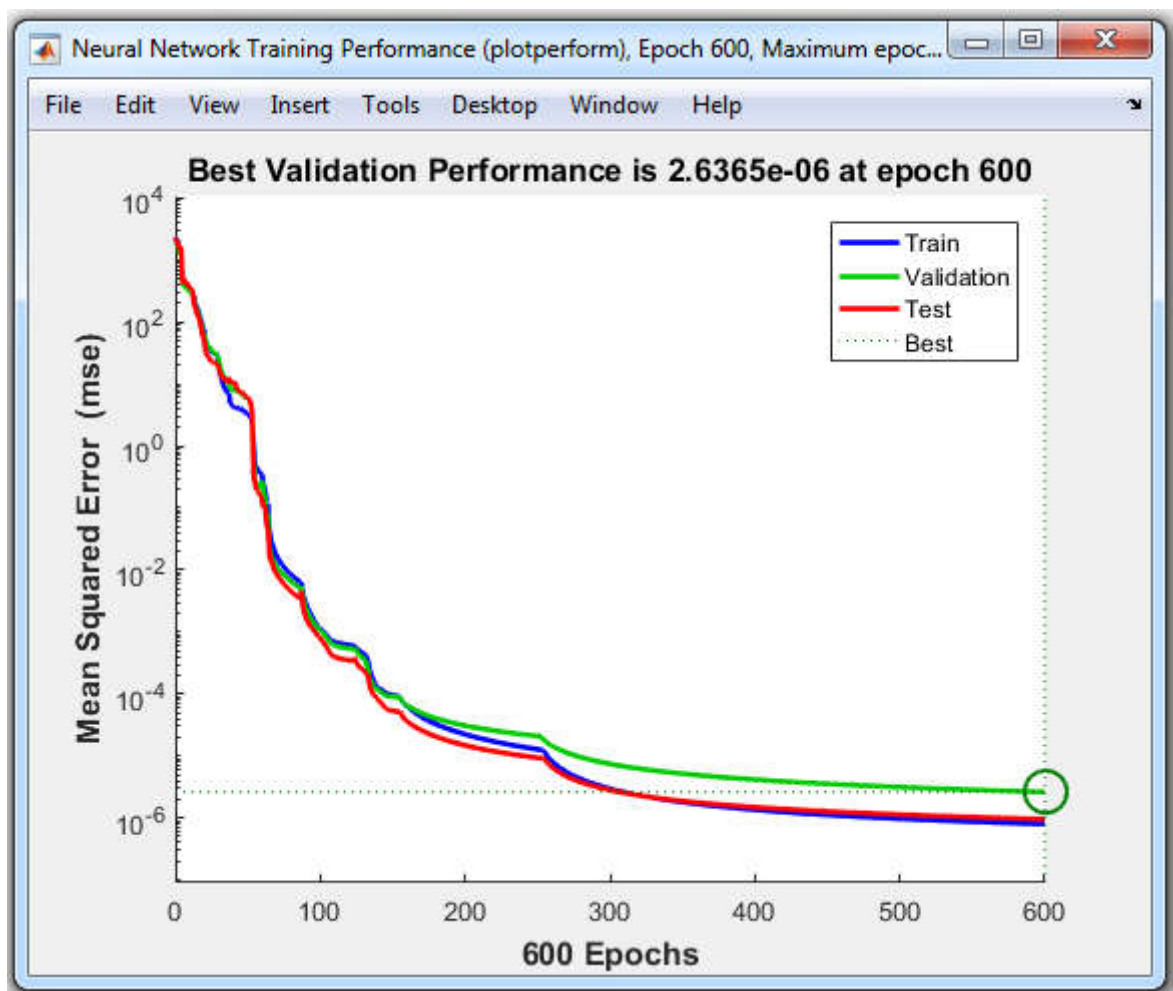


Рисунок 4.15 – Вікно контролю процесу навчання

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

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

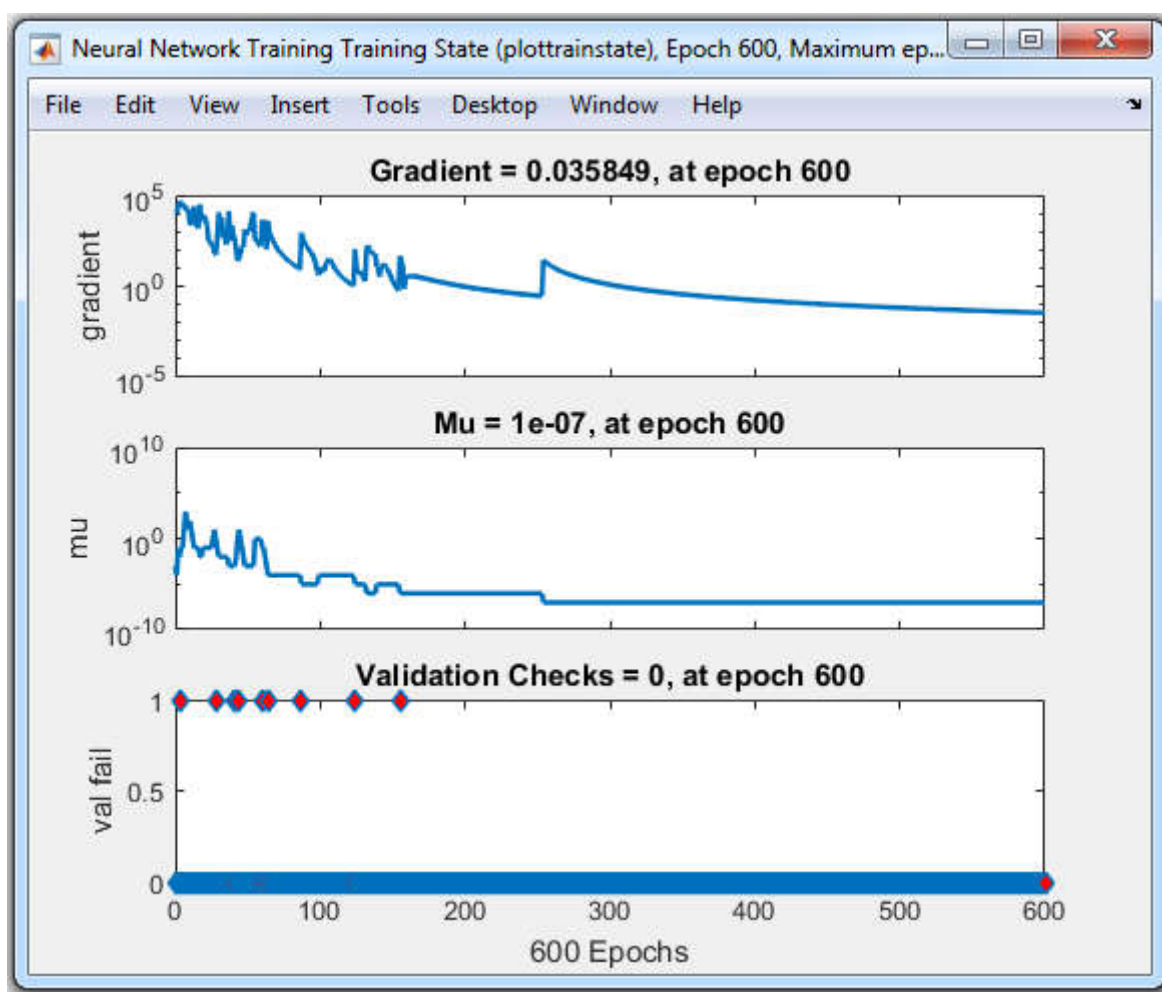


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

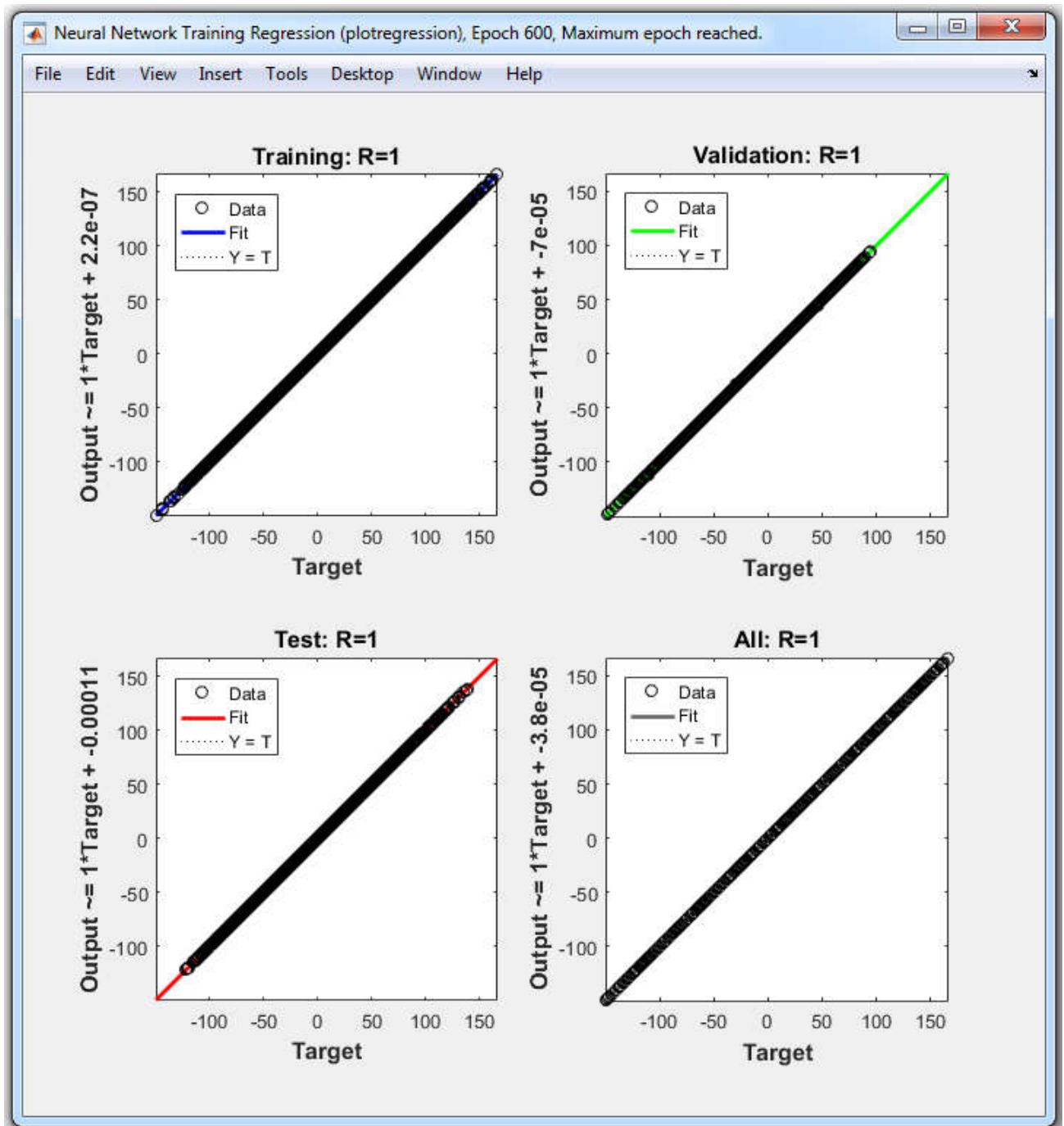


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис секцій вікна Neural Neural Network Training Regression

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

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

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

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

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

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

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

Якщо всі точки розташовані близько до діагональної лінії $Y = T$, це свідчить про високу точність моделі. Велика розсіюваність точок може вказувати на наявність шуму в даних або недостатню адекватність моделі. Графіки Data дозволяють наочно оцінити відповідність моделі даним на різних наборах і виявити можливі аномалії чи області для поліпшення моделі.

Графіки **Fit** представляють результати лінійної регресії, що демонструють прогнозовані значення, які генерує нейронна мережа на основі вхідних даних. Вони відображають, наскільки модель узгоджується з фактичними значеннями, і служать для візуальної оцінки точності прогнозів. Якщо лінія Fit проходить близько до більшості точок, це вказує на високу продуктивність моделі. Відхилення лінії від фактичних значень може сигналізувати про потребу в додатковій оптимізації або корекції гіперпараметрів нейронної мережі.

У вікні Neural Network Training Regression також відображаються коефіцієнти кореляції та рівняння лінійної регресії для кожного набору даних (Training, Validation, Test та All). Вони використовуються для кількісної оцінки точності моделі.

Коефіцієнт кореляції R визначає ступінь лінійної залежності між прогнозованими та фактичними значеннями:

- $R = 1$ – ідеальна лінійна залежність, модель точно передбачає всі значення;
- $R = 0$ – відсутність лінійної залежності, модель не здатна передбачити дані;

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

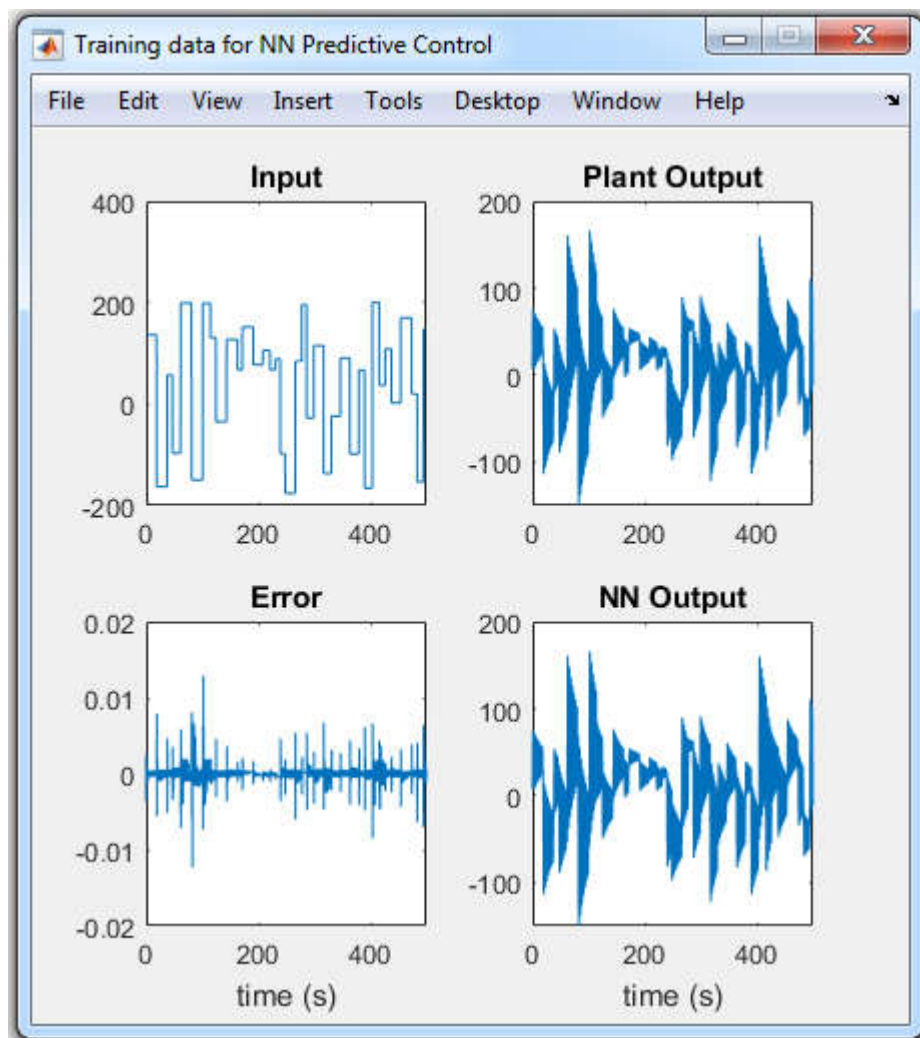


Рисунок 4.18 – Графічне представлення результатів навчання нейронної мережі

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.

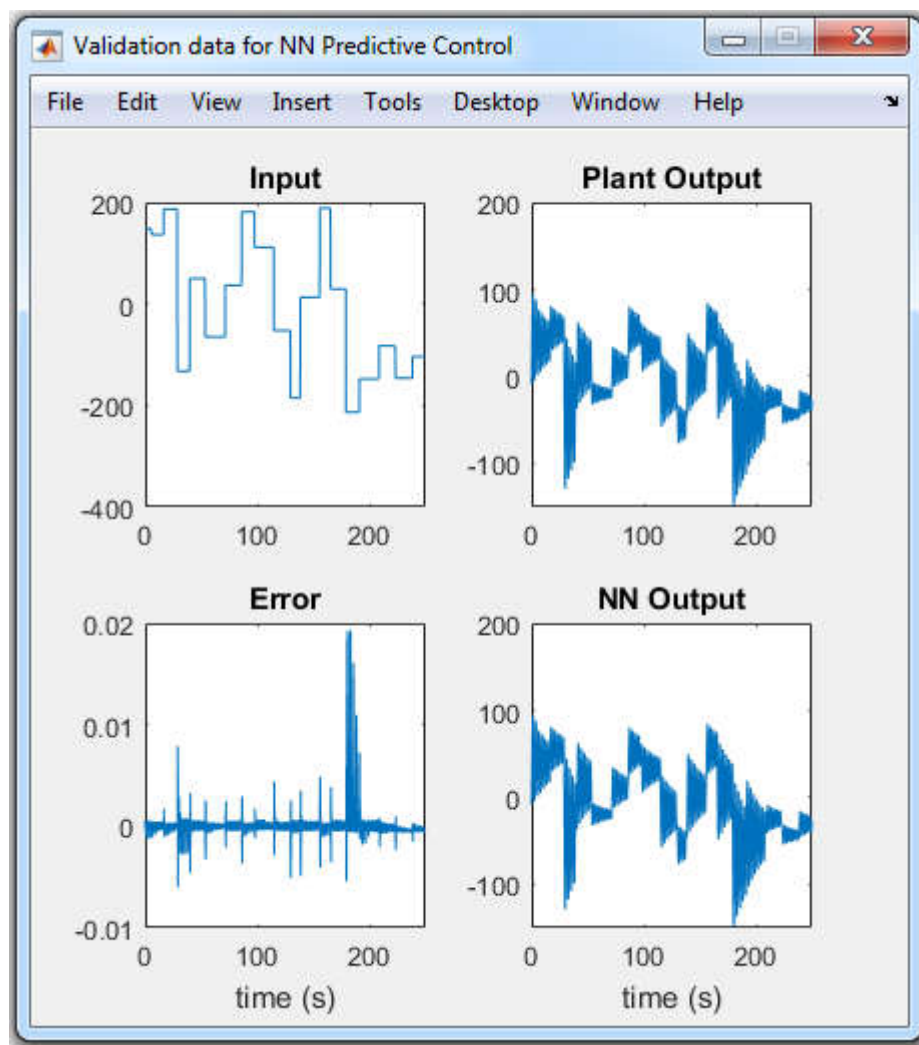


Рисунок 4.19 – Візуалізація результатів перевірки нейронної мережі на контрольній множині

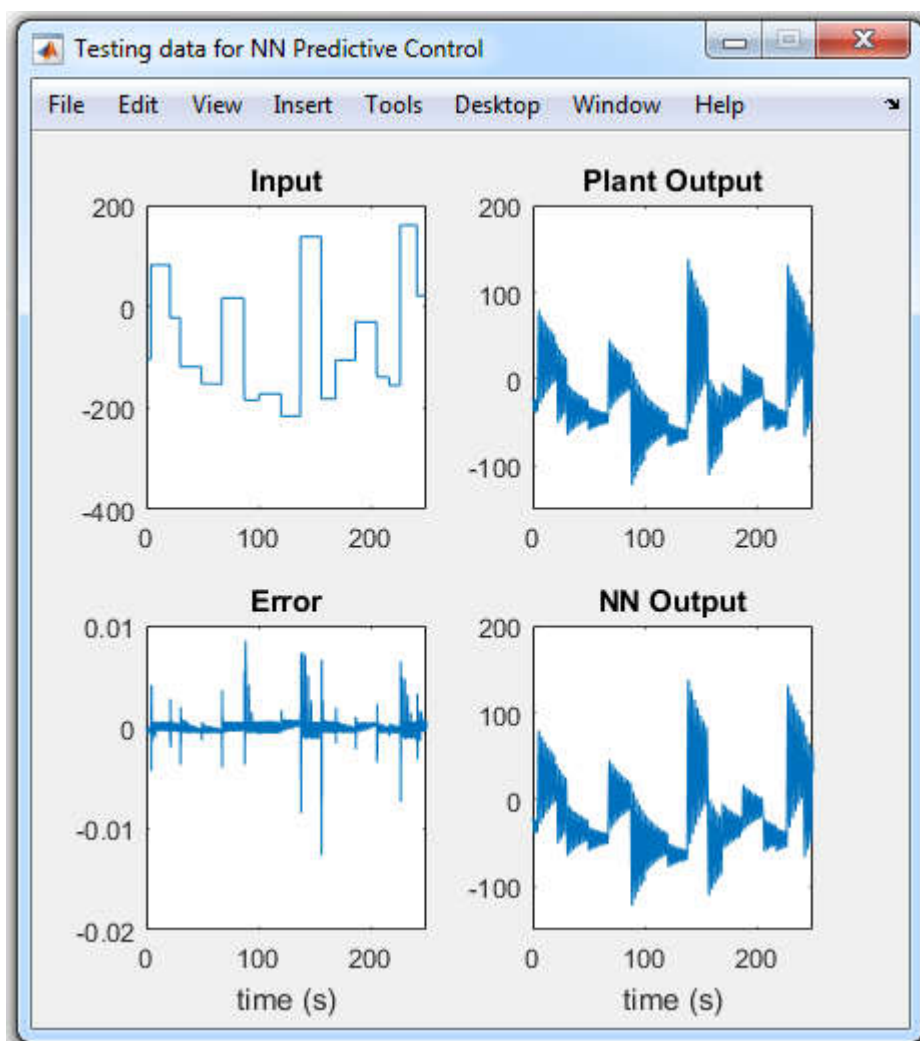


Рисунок 4.20 – Візуалізація результатів перевірки нейронної мережі на тестовій множині

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.

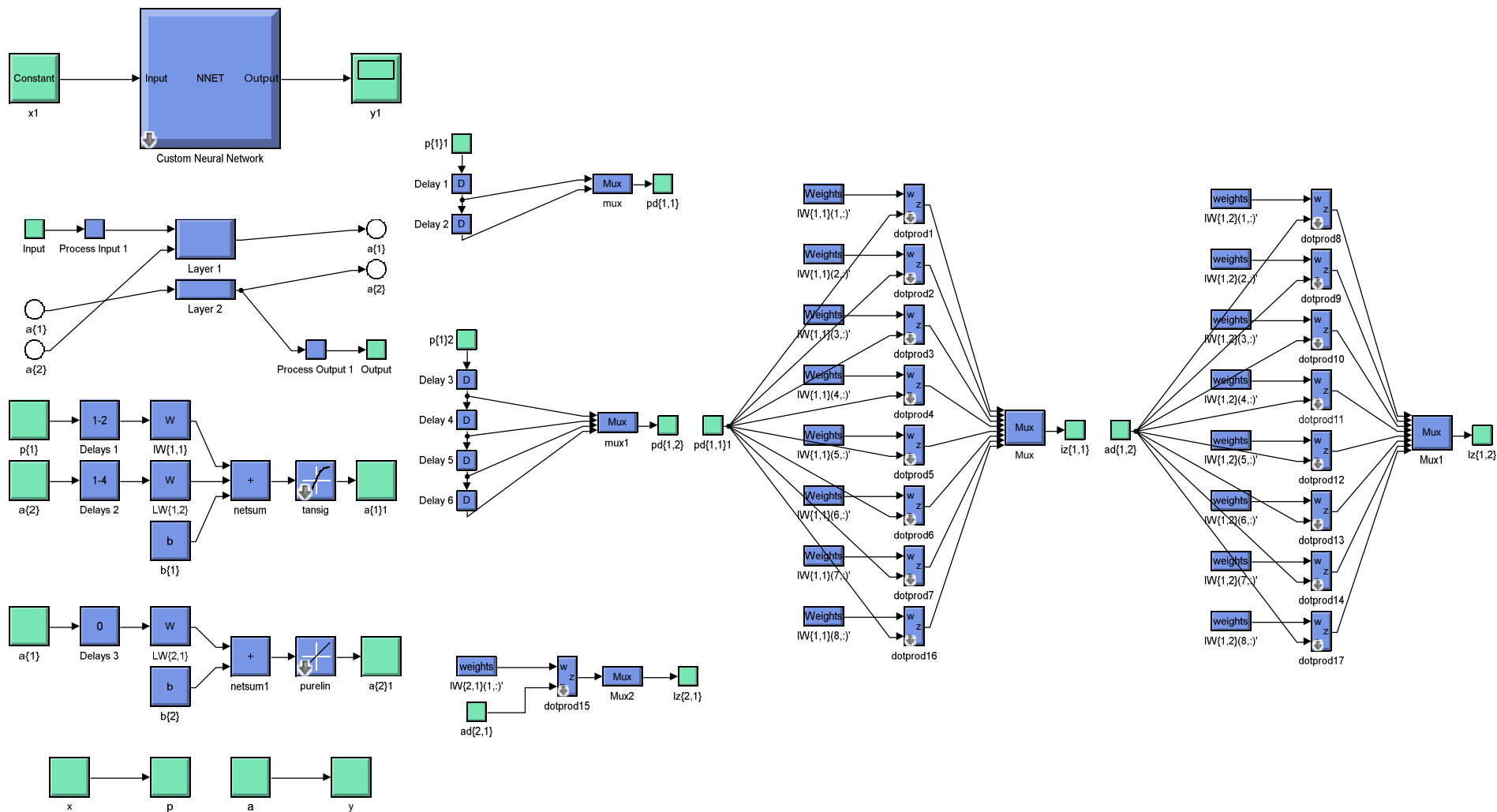


Рисунок 4.21 – Моделі елементів динамічної мережі з блоками затримок у Simulink

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.

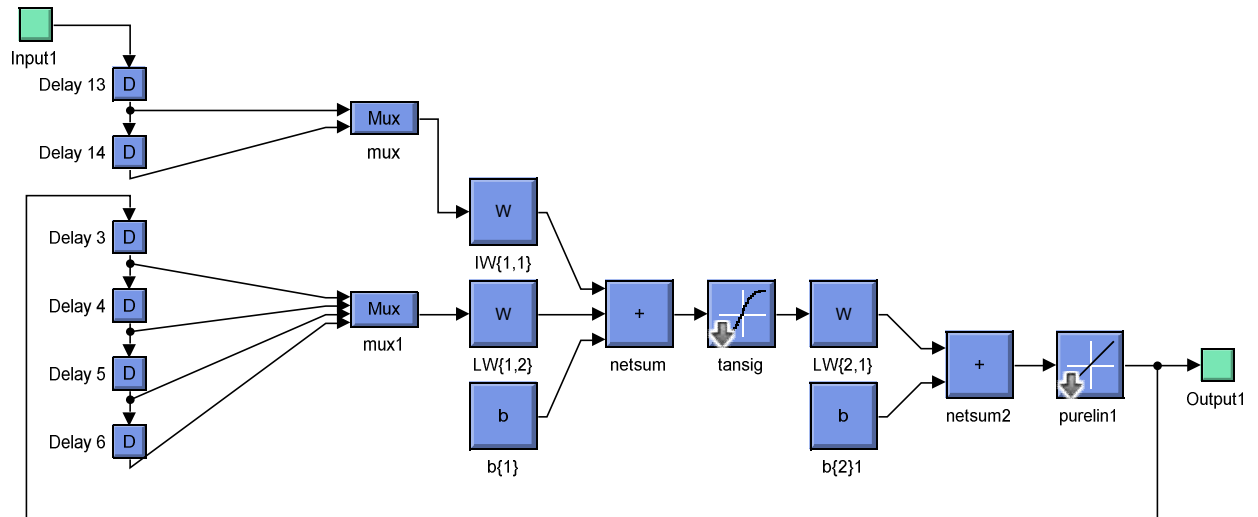


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

Для успішної побудови моделей **netn** і **netn2** необхідно встановити **Breakpoint** у відповідному місці M-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.

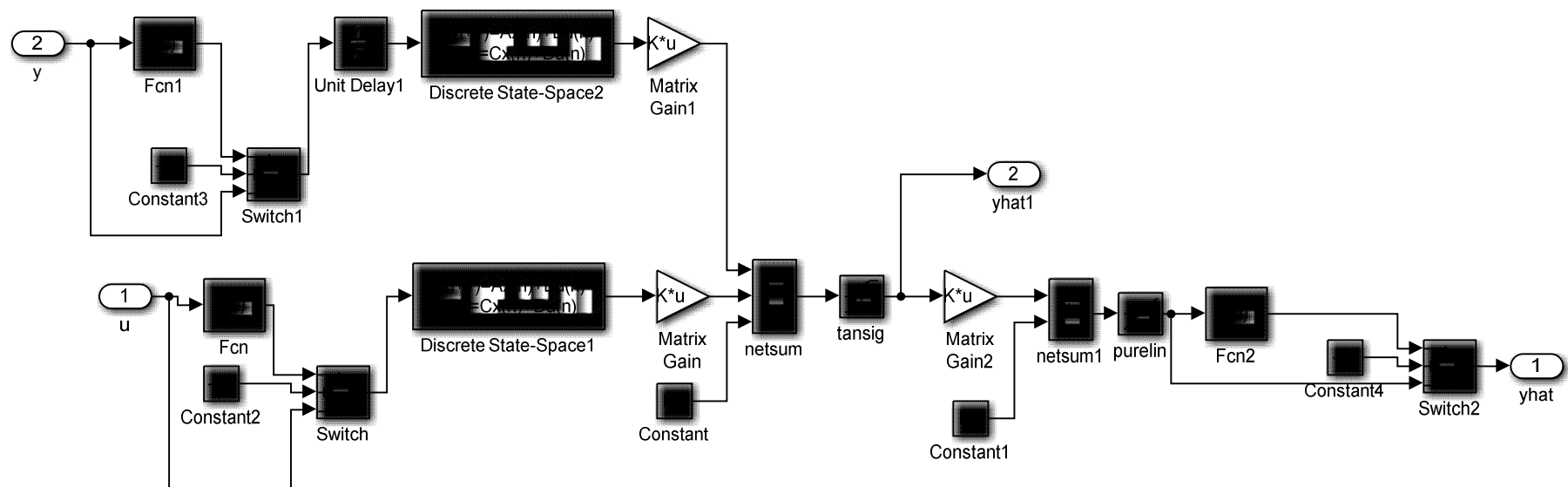


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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.

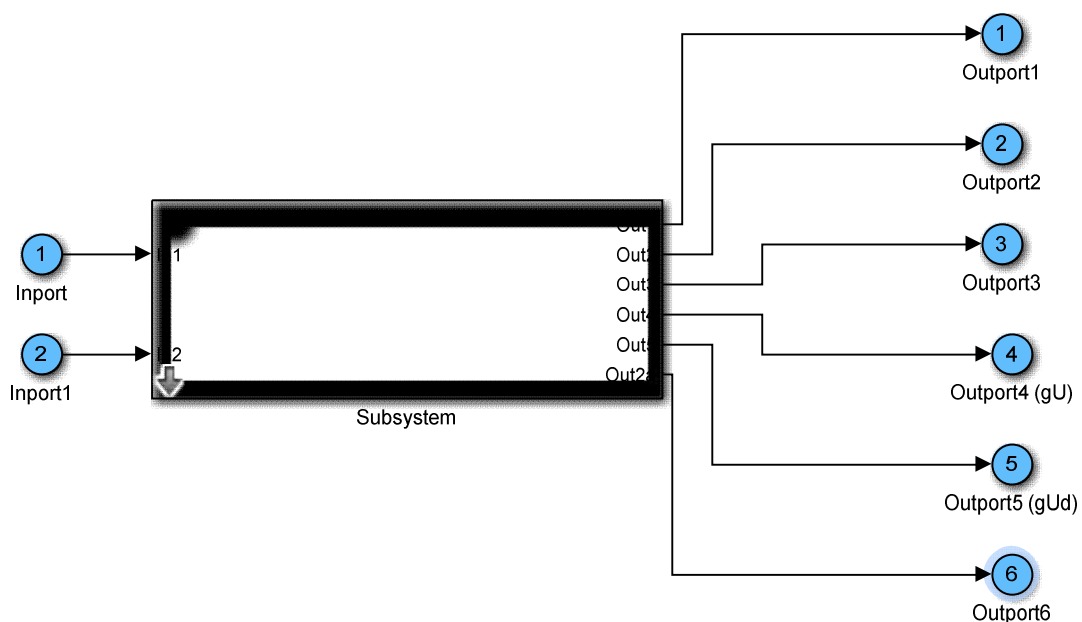


Рисунок 4.24 – Схема ptest3sim2, що використовується для прогнозування процесу функцією predopt

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.

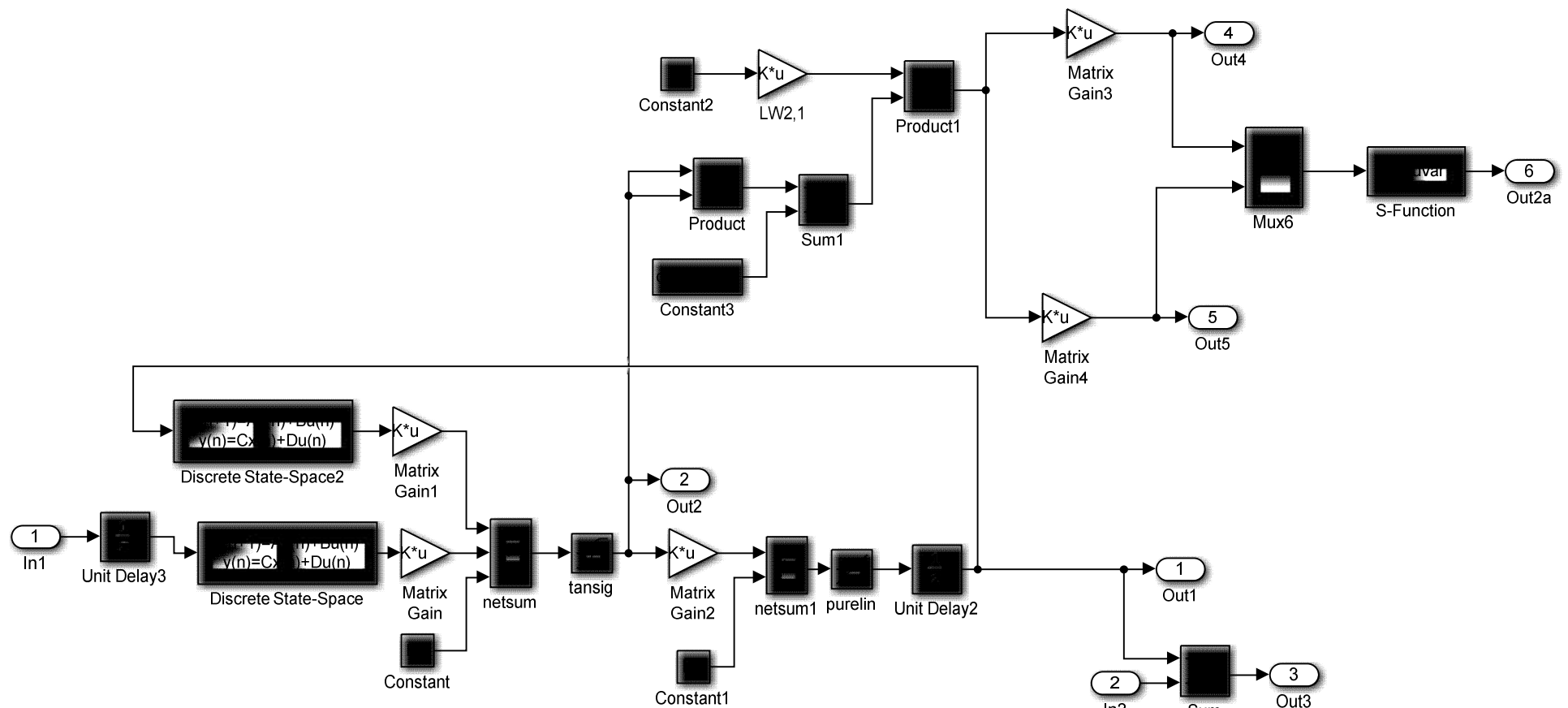


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

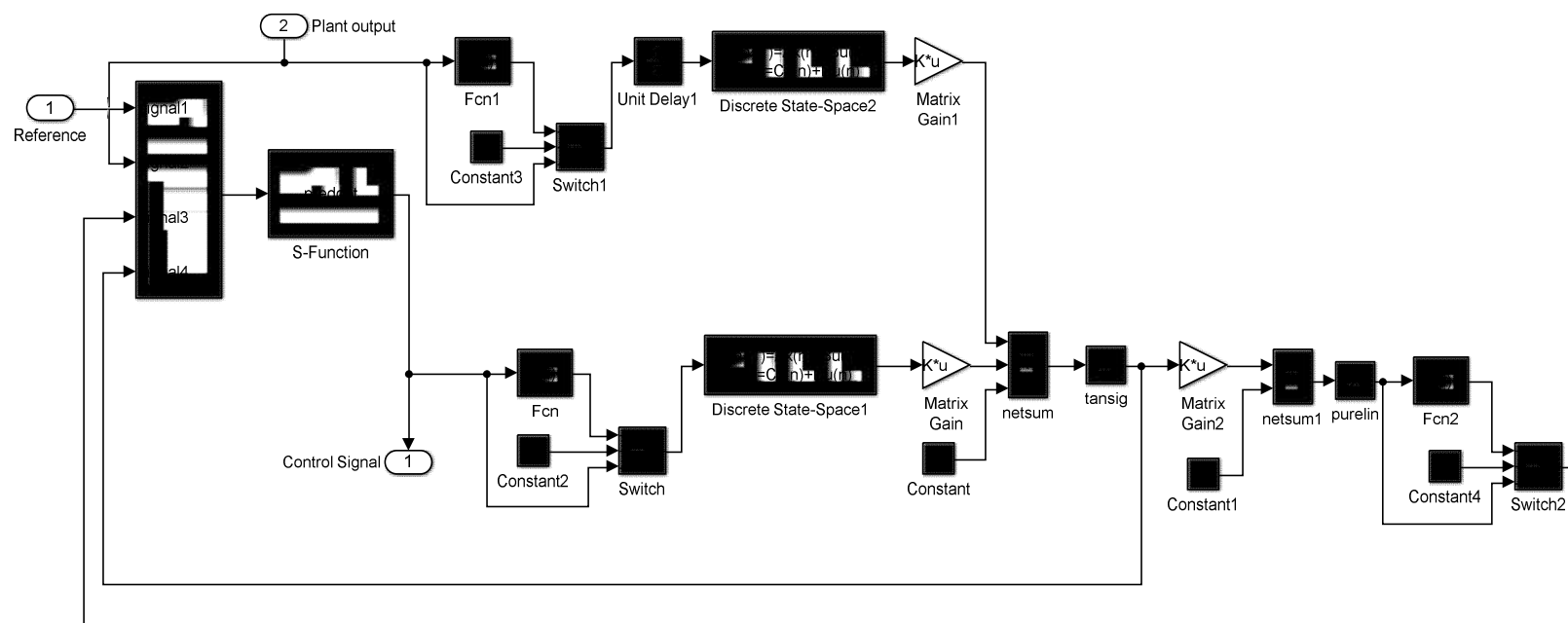


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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.

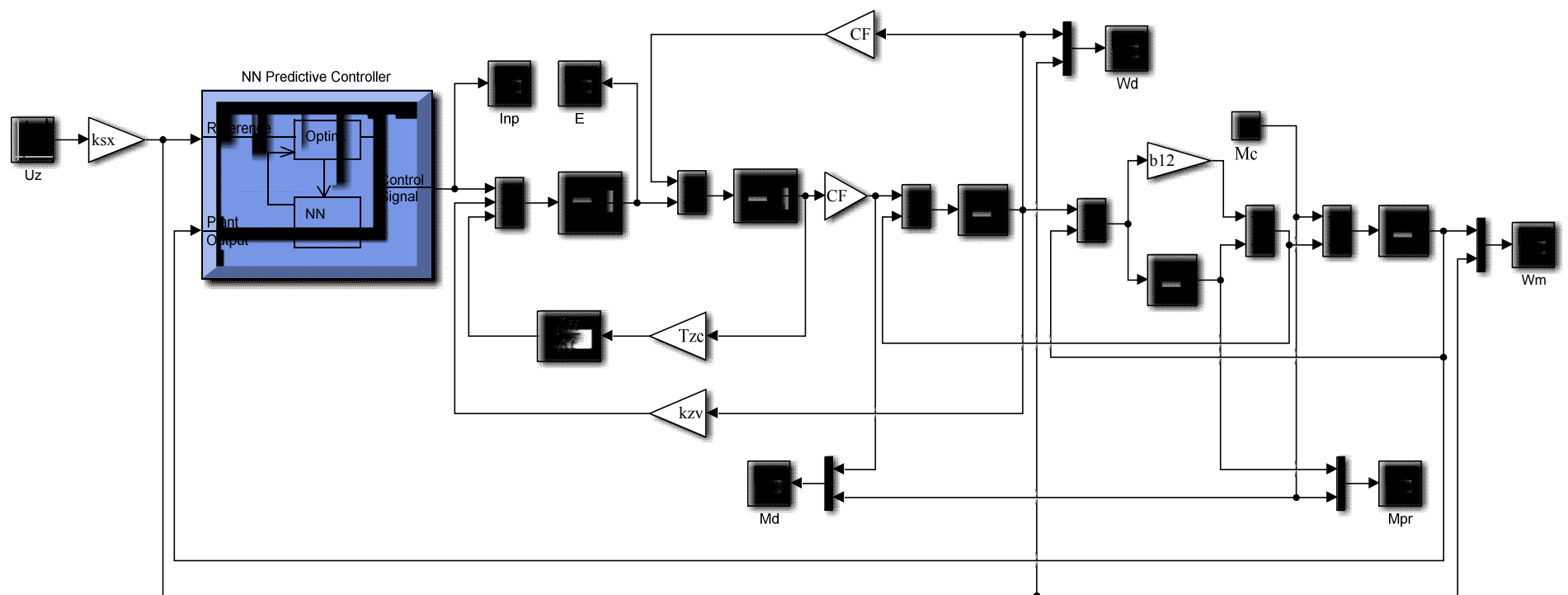


Рисунок 4.27 – Блок-схема моделі двомасової системи із застосуванням нейрорегулятора

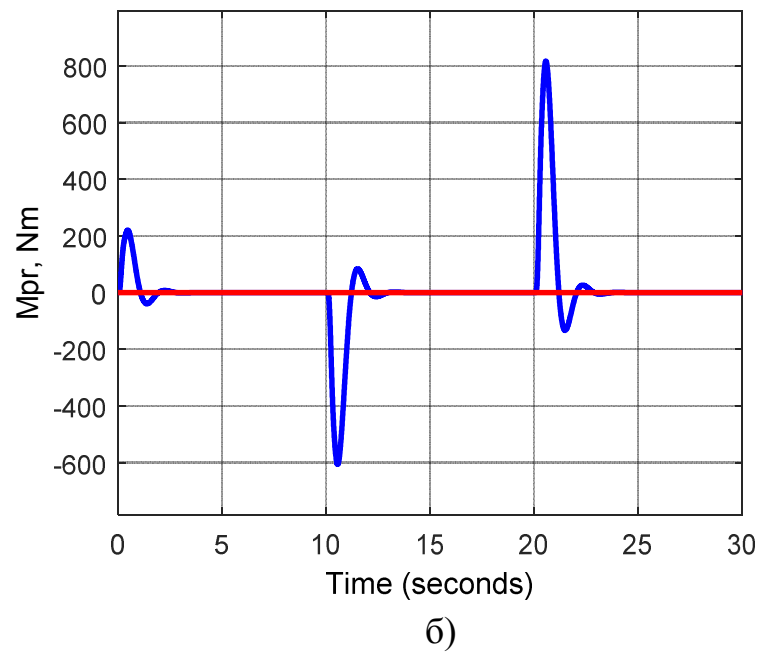
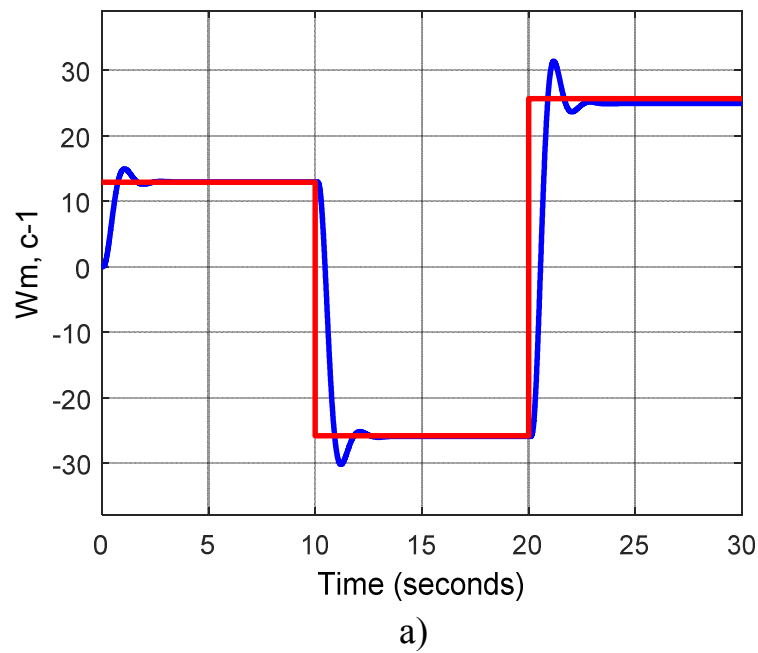


Рисунок 4.28 – Перехідні процеси механічних параметрів двомасової системи з нейрорегулятором під дією задаючої величини:

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

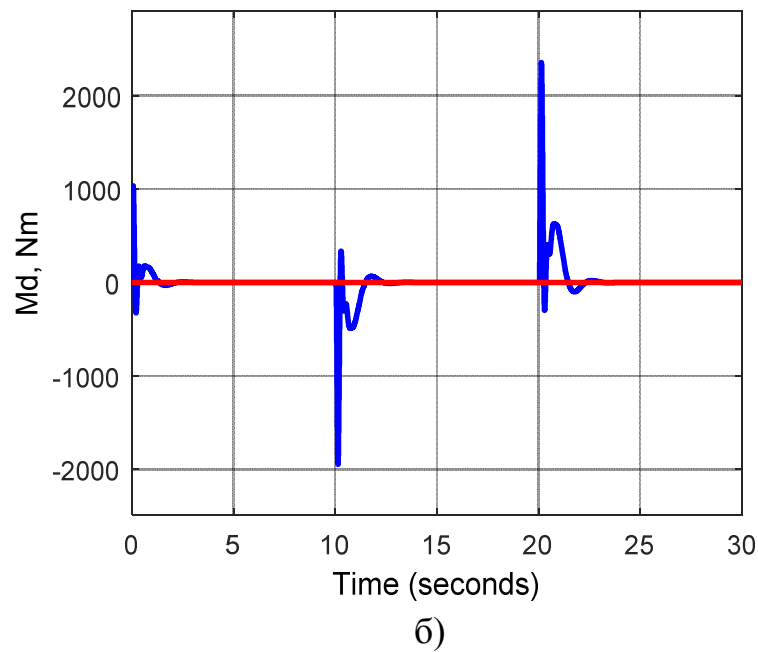
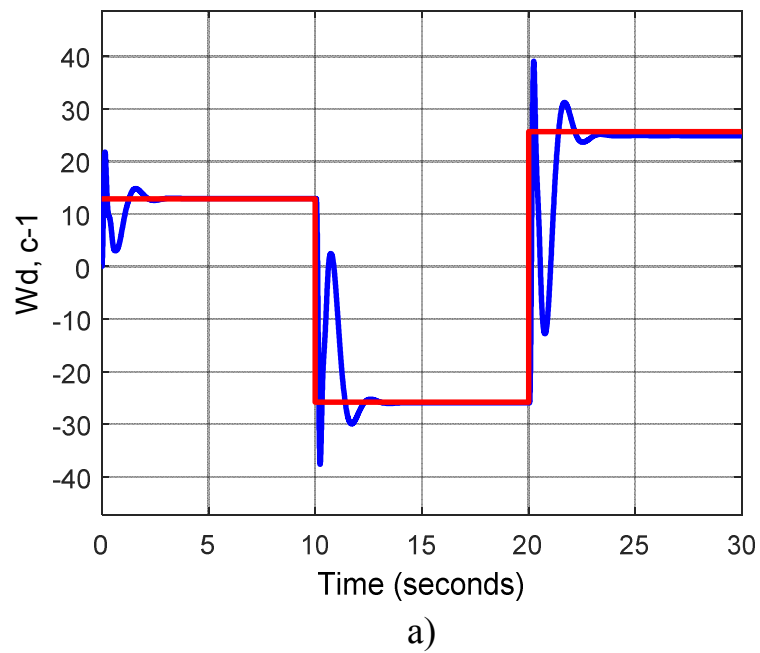
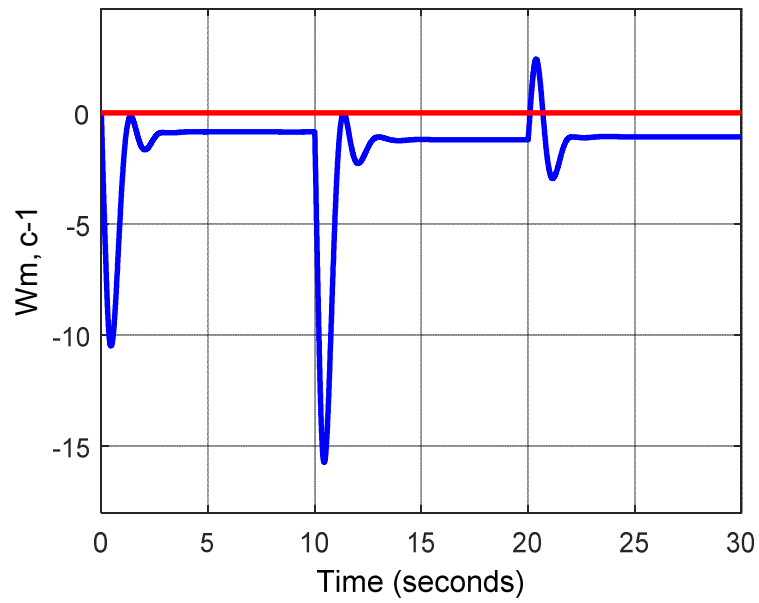
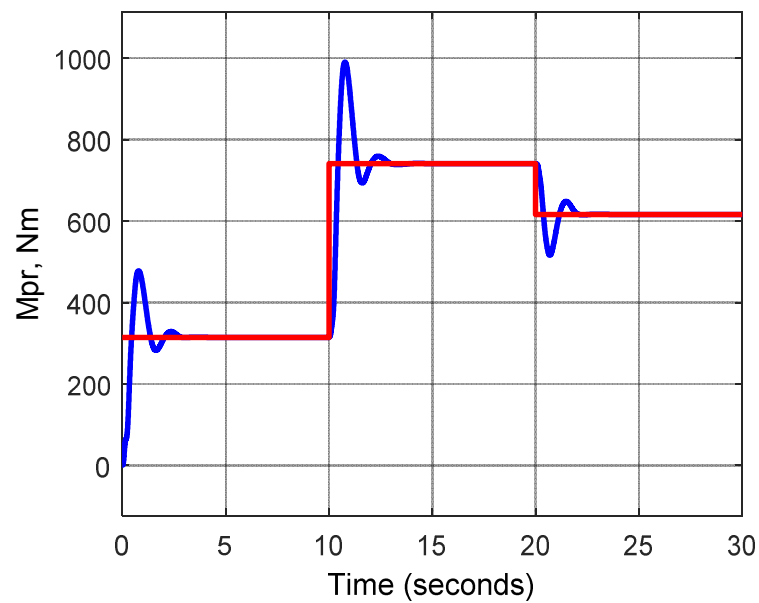


Рисунок 4.29 – Динаміка двигуна двомасової системи з нейрорегулятором під впливом задаючої величини:

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



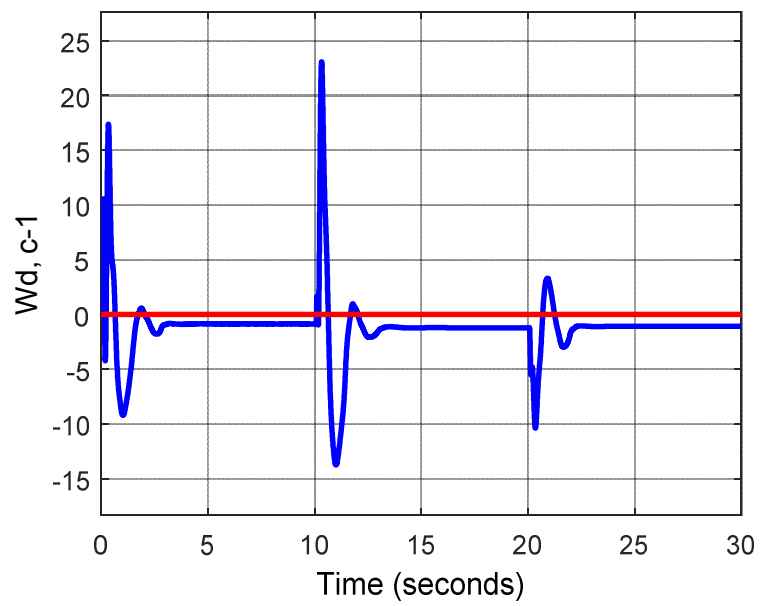
a)



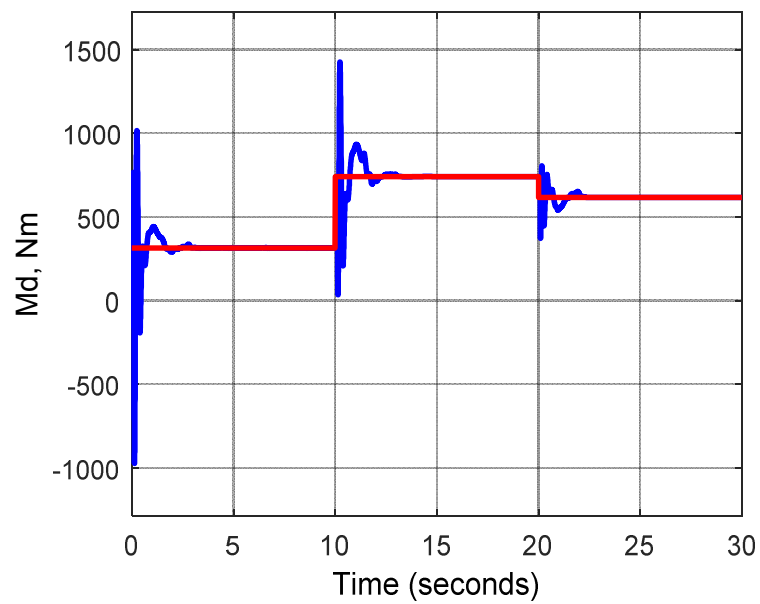
б)

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. Проаналізовано принципи побудови нейрорегуляторів, реалізованих у пакеті MATLAB Neural Network Toolbox: прогнозний регулятор (NN Predictive Controller), регулятор на основі авторегресивної моделі з ковзним середнім (NARMA-L2 Controller) та регулятор із використанням еталонної моделі (Model Reference Controller). Для системи керування механізмом повороту стріли роторного екскаватора, що розглядається в роботі, обрано прогнозний нейрорегулятор NN Predictive Controller. Розглянуто його структуру та алгоритм синтезу.

2. У середовищі SIMULINK створено систему керування на базі нейрорегулятора NN Predictive Controller, яка включає блок керованого об'єкта у вигляді моделі двомасової системи механізму повороту стріли та блок прогнозного регулятора.

3. Виконано генерацію навчальної послідовності шляхом варіювання параметрів тренування з використанням моделі виконавчого механізму. Це дозволило навчити мережу з високою точністю. Оптимальні параметри навчальної послідовності визначені наступним чином: довжина навчальної вибірки $N_b = 10000$, інтервал між зніманнями даних $\Delta t = 0,05 \div 0,1$ с, мінімальний та максимальний інтервали ідентифікації $t_{\min} = 4 \div 8$ с та $t_{\max} = 10 \div 20$ с відповідно.

4. Створено двошарову нейронну мережу з прямим поширенням сигналу та виконано її навчання за допомогою алгоритму Левенберга-Марквардта. Встановлено оптимальну кількість нейронів у прихованому шарі. Помилка навчання знаходиться в межах $10^{-4} \div 10^{-6}$, а миттєві помилки на навчальній, контрольній та тестовій множинах не перевищують 0,02.

5. Розроблено нейрорегулятор з прогнозуванням NN Predictive Controller. Під час варіювання параметрів нейрорегулятора визначено, які з них істотно впливають на ефективність керування, та обрано оптимальні значення. Використання нейромережевої моделі об'єкта, що забезпечує високий рівень

ідентифікації, дозволило створити регулятор із високими показниками якості системи.

6. Проведено моделювання роботи системи з синтезованим нейрорегулятором. Аналіз отриманих перехідних процесів показав, що зміни стану системи під впливом запуску та навантажень мають задовільні характеристики: перерегулювання не перевищує 10%, а час регулювання складає 3 с, що відповідає поставленим технічним вимогам.

ВИСНОВКИ

У магістерській кваліфікаційній роботі розв'язано актуальну науково-практичну задачу створення нейромережевої системи управління багатомасовою електромеханічною системою механізму повороту стріли роторного екскаватора, що забезпечує високу точність та стабільність регулювання. Основні результати дослідження полягають у наступному:

1. Аналіз систем управління та обґрунтування вибору нейромережевого регулятора. Проведено детальний аналіз існуючих систем управління механізмами роторного екскаватора, що забезпечують високу якість регулювання. Враховано вимоги сучасних систем керування, такі як точність, швидкодія та стійкість. На основі цього обґрунтовано доцільність використання нейромережевих технологій, зокрема нейромережевого регулятора з прогнозом (NN Predictive Controller), для управління багатомасовою електромеханічною системою механізму повороту стріли.

2. Вибір електроприводу та розрахунок основного обладнання. Для механізму повороту стріли обрано електропривод постійного струму за схемою Г–Д. Виконано вибір електродвигуна та основного силового обладнання електроприводу, що дозволяє забезпечити необхідну потужність, стабільність роботи та дотримання технічних вимог до сучасних систем керування.

3. Розробка системи автоматичного управління. Виконано вибір і розрахунок системи автоматичного регулювання параметрів електроприводу. Застосовано зворотний зв'язок за швидкістю, гнучкий зворотний зв'язок по струму та струмовий зворотний зв'язок з відсіченням. Проведено розрахунок статичних характеристик замкненої та розімкненої системи, складено алгоритмічну схему управління та визначено передавальні функції по задаючій та збурюючій діях. Проведено моделювання системи в середовищі MATLAB, що дозволило візуалізувати поведінку електроприводу та перевірити працездатність розробленої системи.

4. Математичне моделювання двомасової системи. Розроблено математичну модель двомасової механічної системи, яка враховує пружність механічних з'єднань. Моделювання показало, що перехідні процеси основних координат супроводжуються коливаннями, що підкреслює необхідність застосування ефективного регулятора для стабілізації системи та зменшення перерегулювань.

5. Синтез нейромережевого регулятора NN Predictive Controller. Імплементовано синтез нейромережевого регулятора з прогнозом. Проведено широке варіювання параметрів регулятора для визначення їх впливу на якість регулювання. Встановлено оптимальні значення параметрів, які забезпечують ефективну роботу системи. Використання нейромережевої моделі об'єкта керування дозволило досягти високої точності ідентифікації та створити регулятор, що забезпечує стабільну та прогнозовану реакцію системи.

6. Моделювання та оцінка роботи нейромережевої системи. Проведено моделювання синтезованої нейромережевої системи. Аналіз перехідних процесів показав, що система демонструє задовільний характер реакції на стартові та збурюючі дії. Система є статичною, перерегулювання не перевищує 10%, а час регулювання становить 3 с. Отримані результати підтверджують ефективність розробленого нейромережевого регулятора та його відповідність вимогам до сучасних систем управління.

Таким чином, у роботі створено повністю функціонуючу нейромережеву систему управління, яка забезпечує високий рівень точності та стабільності керування багатомасовою електромеханічною системою механізму повороту стріли роторного екскаватора. Розроблений підхід може бути використаний для подальшої оптимізації систем керування складними механічними об'єктами та розширення на багатомасові динамічні системи в промисловому застосуванні.

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 the Ideas Immanent in Nervous Activity // Bulletin of Mathematical Biophysics. – 1943. – № 5. – P. 115-133
14. Hebb D. The Organization of Behavior. – New York: Wiley Publications, 1949.
15. Rosenblatt F. The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain // Psychological Review. – 1958. – № 65. – P. 386-408.
16. Rosenblatt, F. (1962). *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, D.C.: Spartan Books.
17. Hubel, D. H., & Wiesel, T N. (1959). Brain mechanisms of vision. Scientific American, 241, 150-162.
18. Widrow B., Hoff M. E. Adaptive Switching Circuits / IRE WESCON Convention Record – New York, IRE, 1960. – P. 96-104.
19. Widrow B., Winter R. Neural Nets for Adaptive Filtering and Adaptive Pattern Recognition // IEEE Computer. – 1988. – № 21. – № 3. – P. 25-39.
20. Steinbuch, K., & Piske, U. A. W. (1963). Learning matrices and their applications. IRE Transactions on Electronic Computers, EC-12(6), 846–862. .
21. Minsky, M., & Papert, S. (1969). Perceptrons: An Introduction to Computational Geometry. Cambridge, MA: MIT Press.
22. Hopfield J.J. Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons // Proc. of the National Academy of Science. – 1982. – 81. – P. 3088-3092.
23. Hopfield J. J., Tank D. W. Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1985. – 52. – P. 141-152.
24. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors*. Nature, 323(6088), 533–536.
25. Kohonen, T. (1990). *The self-organizing map*. Proceedings of the IEEE, 78(9), 1464–1480

26. Hopfield, J. J. (1991). *Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences*, 79(8), 2554–2558
27. LeCun, Y., Bengio, Y., & Hinton, G. (2015). *Deep learning. Nature*, 521(7553), 436–444.
28. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
29. Silver, D., Schrittwieser, J., Simonyan, K., et al. (2017). *Mastering the game of Go without human knowledge. Nature*, 550, 354–359.
30. Sen P., Hearn G.E., Zhang Y. Adaptive Neural Controller // In Neural Network Systems Techniques and Applications. Hrsg. Leondes C.T. – 1998. – Vol.4. – P. 274–343.
31. Werbos P.J. Backpropagation and neurocontrol: A review and prospects // Proc. of International Joint Conf. On Neural Networks. – Washington DC. – 1989. – Vol. 1. – P. 209–216.
32. Psaltis A.; Sideris A., Yamamura A. A multi-layered neural network controller // IEEE Control Systems Mag. – 1988. – №8. – P. 17–21.
33. Saerens M.; Soquet A. A neural controller // Proceedings of the 1st IEE International Conference on Artificial Neural Networks, London. – 1989. – P. 211–215.
34. Narendra K.S. Neural networks for control: theory and practice // Proc. IEEE. – 1996. – №84. – P. 1385–1406.
35. Neural networks for control systems: A survey K. J. Hunt, D. Sbarbaro, R. Zbikowski, P. J. Gawthrop // Automatica. – 1992. – Vol. 28. – № 6. P. 1083–1112.
36. Ishiguro A. et al. A neural network compensator for uncertainties of robotic manipulators // IEEE Trans, on Industrial Electronics. – 1992. – 339. – P. 565–569.
37. Garcia C.E.; Morari M. Internal Model Control: Unifying Review and Some New Results // Ind. Eng. Chem. Process Des. Dev. – 1982. – №21. – 308–323.
38. Park Y.-M., Choi M.S.; Lee Y. An optimal tracking neuro-controller for nonlinear dynamic systems // IEEE Trans. Neural Networks. – 1996. – №7. – P. 1099–1110.
39. Mistry, S.I.; Nair S.S. Indirect control of a class of nonlinear dynamic systems.

// IEEE Trans. Neural Networks. – 1996. – №7. – P. 1015-1023.

40.Miyamoto H., Kawato M., Setoyama, T., Suzuki R. Feedback error learning neural network for trajectory control of robotic manipulator //IEEE Trans, on Neural Networks. – 1988. – Vol. 1. – P. 251-265.

41.Jung S. Neural Network Controllers for Robot Manipulators. PhD thesis, University of California. – Davis. – 1996. – 132 p.

42.Jung S.; Hsia T.C. A new neural network control technique for robot manipulators // Robotica. – 1995. – Vol. 13. – P. 477-484.

43.Sorensen O. Neural Networks in Control Applications. PhD thesis. Aalborg University. Department of Control Engineering. – 1994. – 187 p.

44.Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.

45.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.

46.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.

47.Review on model predictive control: an engineering perspective – *The International Journal of Advanced Manufacturing Technology*, 2021, Vol. 117, pp. 1327-1349.

48.Lan J. *Efficient model predictive control for nonlinear systems modelled by deep neural networks* : e-print (arXiv) / J. Lan. - 2024.

49.Li Y., Sundararajan N., Saratchandran P. (2001), Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks, *Automatica*, Vol. 37, No 8, pp. 1293-1301.

50.Gundy-Burlet K., Krishnakumar K., Limes G., Bryant D. (2004), Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft, *J. of Aerospace Computing, Information, and Communication*, Vol. 1, No.12, pp. 526-542.

51. Nikiforova L. N., Petrosyan E. A., Yakemenko G. V. (2000), "Neyrokompyuteryi v upravlenii vertoletami", [Neurocomputers in helicopter control], Artificial Intelligence, No. 3, pp. 290-298.
52. Kupin A. I. (2008), "Intelektualna Identifikatsiya ta keruvannya v umovah protsesiv zbagachuvalnoyi tehnologiyi", [Intelligent identification and control in the processes of enrichment technology], KTU, Krivoy Rog, 204 p.
53. D. Gu and H. Hu. (2002), Neural Predictive Control for a Car-like Mobile Robot, International Journal of Robotics and Autonomous Systems, Vol. 39 (2), pp. 73-86.
54. Danil V. Prokhorov. (2008), Toyota Prius HEV Neurocontrol and Diagnostics, Neural Networks, No. 21, pp. 458-465.
55. Zmeu K. V., Markov N. A., Shipitko I. A., Notkin B. S. (2009), "Bezmodelnoe prognoziruyshee inversnoe neyro-upravlenie s regeneriruemym etalonnym perehodnym protsessom", [A modelless predictive inverse neuropravlenie with a regenerated reference transient process], Intelligent Systems, No. 3, pp. 109-117.
56. Kuznetsov B. I. , Vasilets T. Yu., Varfolomiev O. O. (2015), "Neyromerezheva sistema navedennya i stabilizatsiyi z regulyatorom na osnovi etalonnoyi modeli", [Neural network aiming and stabilization system with the reference model controller], Electrical engineering and electromechanics, No. 4, pp.35-39.
57. Dzyuba D. A., Chernodub A. N. (2010), "Primenenie metoda kontroliruemogo vozmuscheniya dlya modifikatsii neyrokontrollerov v realnom vremeni", [The application of the controlled disturbance method for the modification of real time neural controllers], Mathematical machines and systems, No. 4, pp. 20-28.
58. Dias F.M., Mota A.M. (2001), Comparison between Different Control Strategies using Neural Networks, 9th Mediterranean Conference on Control and Automation, Croatia, Dubrovnik.
59. Venayagamoorthy G.K., Harley R.G., Wunsch D.C. (2003), Implementation of Adaptive Criticbased Neurocontrollers for Turbogenerators in a Multimachine Power System", IEEE Transactions on Neural Networks, Vol. 14, Issue 5, pp. 1047–1064.
60. D’Emilia G., Marrab A., Natalea E. (2007), Use of neural networks for quick

and accurate autotuning of PID controller, *Robotics and Computer-Integrated Manufacturing*, Vol. 23, pp. 170 – 179.

61.Ferreira,P.M., Ruano,A.E., Silva,S., Conceição,E.Z.E. (2012). Прогнозоване керування на основі нейронних мереж для теплового комфорту та енергозбереження у громадських будівлях. *Energy & Buildings*,55, 238-251.

62.Yadav,M.K., Chandra,D. (2015). Керування системами кондиціонування повітря за допомогою нейронної мережі. *Міжнародний журнал електроніки та електротехніки*, 3(5), 406-410.

63.Камінські, М. та ін. (2023). Застосування нейронних мереж в електроприладах - тенденції та перспективи. *Energies*, 16 (11).

64.Jin,L., Li,S., Yu,J., He,J. (2018). Керування роботом-маніпулятором за допомогою нейронних мереж: Огляд. *Neurocomputing*,285, 23-34.

65.Лі, Т., Чжан, Г., Чжан, Т., Пан, Дж. (2024). Адаптивне нейромережеве відстеження керування роботизованими маніпуляторами на основі спостерігача збурень. *Processes*, 12(3), 499.

66.Liu,Z., Peng,K., Han,L., Guan,S.C. (2023). Моделювання та керування роботизованими маніпуляторами на основі штучних нейронних мереж: Огляд. *Іранський журнал науки і техніки, Transactions of Mechanical Engineering*, 47, 1307-1347.

67.Лі, Дж., Су, Дж., Ю, В., Мао, Х., Лю, З., Фу, Х. (2024).Рекурентна нейронна мережа для керування траєкторією маніпулятора з невідомою матрицею мас. *Frontiers in Neurorobotics*.

68.Гупта, П., Сінха, Н. К. (2017). Керування роботизованими маніпуляторами за допомогою нейронних мереж: Огляд. У *Методах та застосуваннях інтелектуального керування*, 103-136. Springer.

69.Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.

70.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.

71. 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.

72. Теорія автоматичного керування : навчальний посібник / П. В. Леонтьєв та ін. ; за заг. ред. П. В. Леонтьєва. – Суми : Сумський державний університет, 2024. – 296 с.

73. Теорія автоматичного керування : навчальний посібник / О. К. Аблесімов – К. : «Освіта України», 2019. – 270 с.

74. 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.

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

76. 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

77. 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.

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

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

80. А. А. Видмиш, Л. В. Ярошенко. Основи електропривода. Теорія та прак-

тика. Частина 1. Навчальний посібник. – Вінниця: ВНАУ, 2020. – 387 с.

81.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.

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

83.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.

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

85.Шефер О.В. Автоматизований електропривод загальнопромислових ме-
ханізмів: конспект лекцій. – Полтава: ПолтНТУ, 2020. – 154 с.

86.Електропривод виробничих машин і механізмів: Навчальний посібник /
О.Ю. Синявський, В.В. Савченко, В.Я. Бунько, В.Ю. Рамш; За ред. О.Ю. Си-
нявського. – К.: ФОП Ямчинський О.В. , 2020. – 444 с. ISBN 978-617-7890-88-0.

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

88.Модельювання систем управління в SIMULINK : навч. посібник / В. О. Бо-
гомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтєв, О. М. Красюк. Харків
ХНАДУ, 2018. – 220 с.

ДОДАТОК А

СТРУКТУРА ШТУЧНОГО НЕЙРОНА

Актуальність досліджень штучних нейронних мереж підтверджується різноманіттям їх можливих застосувань. Теорія штучних нейронних мереж є алгоритмічним базисом розвитку нейрокомп'ютерів, подібно до тому, яка булева алгебра більше 50 років була алгоритмічним базисом однопроцесорних і багатопроцесорних ЕОМ.

Сучасні цифрові обчислювальні машини перевершують людину по здатності проводити числові і символічні обчислення. Проте людина може без зусиль вирішувати складні задачі сприйняття зовнішніх даних (наприклад, дізнатися в натовпі знайомого тільки по його обличчю, що промайнуло) з такою швидкістю і точністю, що найпотужніший в світі комп'ютер в порівнянні з ним виявиться безнадійним тугодумом. Причина такої значної відмінності в їх продуктивності заключається в тому, що архітектура біологічної нейронної системи абсолютно не схожа на архітектуру машини фон Неймана (табл. А.1), а це істотно впливає на типи функцій, які ефективніше використовуються кожною з моделей.

Нейронна мережа – це мережа з кінцевим числом шарів з однотипних елементів – аналогів нейронів з різними типами зв'язків між шарами. При цьому число нейронів в шарах вибирається виходячи з необхідності забезпечення заданої якості рішення задачі, а число шарів нейронів – як можна менше для скорочення часу рішення задачі.

Розвиток штучних нейронних мереж надихався і надихається біологією. Проте наші знання про роботу мозку обмежені, і розробникам штучних ШНМ доводиться виходити за межі сучасних біологічних знань у пошуках структур, що дозволяють виконувати корисні функції. Так часто створюються мережі, що неможливі в живій матерії або такі, що вимагають неправдоподібно великих допущень про анатомію і функціонування мозку.

Нервова система людини, побудована з елементів, званих нейронами, дуже складна. Близько 10^{11} нейрона беруть участь в приблизно 10^{15} передаючих

зв'язках. Унікальними здібностями нейронів є прийом, обробка і передача електрохімічних сигналів по нервових шляхах, які утворюють комунікаційну систему мозку.

Таблиця А.1 – Машина фон Неймана в порівнянні з біологічною нейронною системою

Параметри порівняння	Машина фон Неймана	Біологічна нейронна система
	1	2
Процесор	Складний	Простий
	Високошвидкісний	Низькошвидкісний
	Один або кілька	Велика кількість
Пам'ять	Відокремлена від процесора	Інтегрована в процесор
	Локалізована	Розподілена
	Адресація не за змістом	Адресація за змістом
Надійність	Висока уразливість	Живучість
Обчислення	Централізовані	Розподілені
	Послідовні	Паралельні
	Програми, що зберігаються	Самонавчання
Спеціалізація	Чисельні і символні операції	Проблеми сприйняття
Середовище функціонування	Строго визначена	Погано визначена
	Строго обмежена	Без обмежень

Не дивлячись на величезну кількість нейронів, їх тіла займають всього декілька відсотків загального об'єму мозку. Решта майже всього простору зайнята міжнейронними зв'язками. Число зв'язків кожного нейрона не має аналогів в сучасній техніці. Розуміння того, що міжнейронні зв'язки відносяться до основних структурних компонентів мозку, що в першу чергу визначає його функціональні характеристики, є одним з найбільш істотних висновків, зроблених нейрофізіологами. В підтвердження можна привести вислів відомого нейрофізіолога Э. Кендела: «На переконання багатьох нейробіологів врешті-решт буде

доведено, що унікальні властивості кожної людини – здатність відчувати, думати, навчатися і пам'ятати – поміщені в строго організованих мережах синаптичних взаємозв'язків між нейронами головного мозку».

На рис. А.1 приведена структура типового біологічного нейрона. Дендрити йдуть від тіла нервової клітини до інших нейронів, де вони приймають сигнали в точках з'єднання, званих синапсами. Прийняті синапсом вхідні сигнали підводяться до тіла нейрона. Тут вони підсумовуються, причому одні входи прагнуть збудити нейрон, а інші перешкодити його збудженню. Відповідно до цього розрізняють процеси синаптичного збудження і гальмування нейрона.

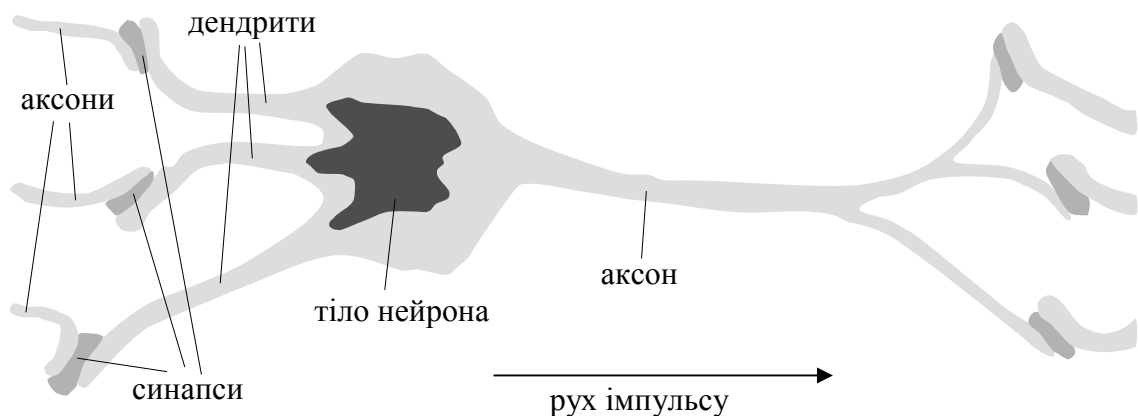


Рисунок А.1 – Біологічний нейрон

Коли сумарне збудження в тілі нейрона перевищує деякий поріг, нейрон збуджується, посылаючи по аксону, який на кінці сильно гілкується, сигнал іншим нейронам. Нервовий імпульс володіє здатністю розповсюджуватися по нервових волокнах без загасання, аж до синаптичних закінчень із швидкістю розповсюдження біля 10 м/с .

Синаптичне гальмування знижує збудження нейрона і тим самим утрудняє перехід збудження нейрона в нервовий імпульс.

Після генерації імпульсу настає так званий рефрактерний період, під час якого нейрон відновлює здібність до генерації наступного імпульсу. Тривалість рефрактерного періоду менш 1 мс . Тому максимальна частота генерації імпульсів нейронів біля 1000 с^{-1} .

Штучний нейрон імітує в першому наближенні властивості біологічного нейрона. Він володіє групою синапсів – однонаправлених вхідних зв'язків, сполучених з виходами інших нейронів, а також має аксон – вихідний зв'язок даного нейрона, з яким сигнал (збудження або гальмування) поступає на синапси наступних нейронів. Кожен вхід множиться на відповідну вагу, аналогічну синаптичній силі, і всі добутки підсумовуються, визначаючи рівень активації нейрона. Загальний вид нейрона, що реалізовує ці властивості, приведений на рис. А.2.

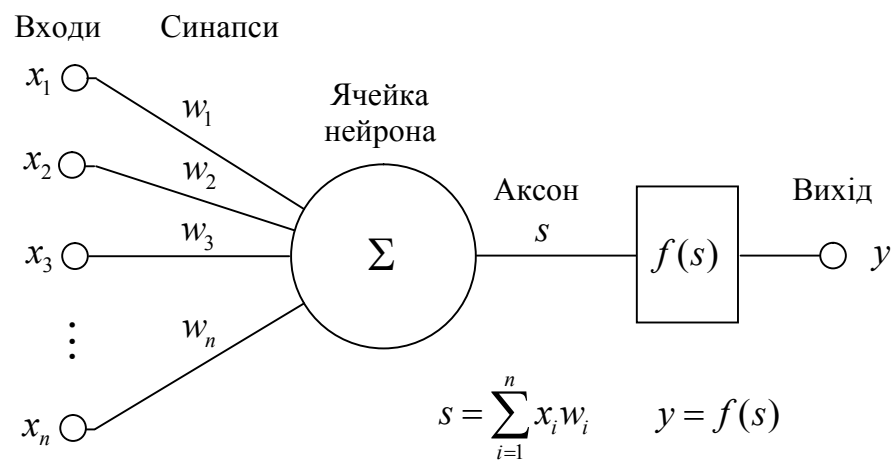


Рисунок А.2 – Штучний нейрон

Множина вхідних сигналів позначається вектором $\mathbf{X}$. Кожна вага w_i відповідає "силі" одного біологічного синаптичного зв'язку. Множина вагів в сукупності позначається вектором $\mathbf{W}$.

Блок, що підсумовує, відповідний тілу біологічного елементу, алгебраїчно складає зважені входи

$$s = \sum_{i=1}^n x_i w_i . \quad (3.1)$$

Вихід нейрона є функцією його стану:

$$y = f(s) \quad (3.2)$$

Нелінійна функція f називається активаційною і може мати різний вигляд, що, зокрема, показано на рис. А.3.

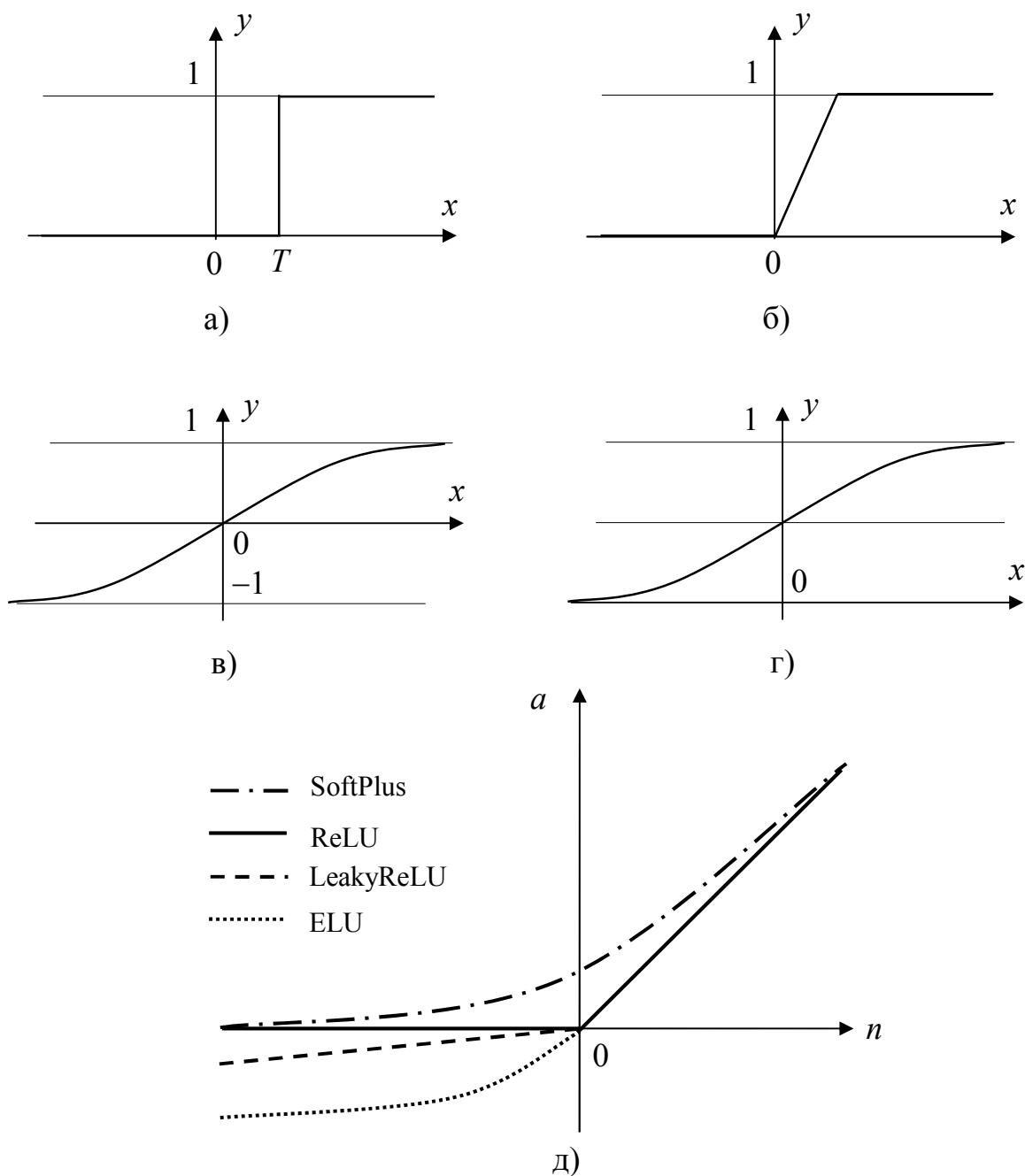


Рисунок А.3 – Типи активаційних функцій: а – функція одиничного стрибка; б – лінійний поріг (гістерезис); в – сигмоїд – гіперболічний тангенс; г – сигмоїд (формула (А.3)); д - функція активації ReLU

У разі, коли функція активації одна і та ж для всіх нейронів мережі, мережу називають однорідною (гомогенною). Якщо ж активаційна функція залежить ще від одного або декількох параметрів, значення яких міняються від нейрона до нейрона, то мережу називають неоднорідною (гетерогенною).

В магістерській роботі використовується нелінійна функція з насиченням, так звана логістична функція або сигмоїд (тобто функція S -подібного вигляду):

$$f(x) = \frac{1}{1 + e^{-\alpha x}}.$$

При зменшенні α сигмоїд стає пологішим, в межі при $\alpha = 0$ вироджується в горизонтальну лінію на рівні 0,5. При збільшенні α сигмоїд наближається на вигляд до функції одиничного стрибка з порогом T у точці $x = 0$. З виразу для сигмоїда очевидно, що вихідне значення нейрона лежить в діапазоні $[0,1]$. Одна з цінних властивостей сигмоїдної функції – простий вираз для її похідної, застосування якого буде розглянуто надалі

$$f'(x) = \alpha f(x)(1 - f(x)).$$

Слід зазначити, що сигмоїдна функція диференціюється на всій осі абсцис, що широко використовується в багатьох алгоритмах навчання. Крім того, вона володіє властивістю підсилювати слабкі сигнали краще, ніж сильні, і запобігає насиченню від сильних сигналів, оскільки вони відповідають областям аргументів, де сигмоїд має пологий нахил. Іншою широко використовуваною активаційною функцією є гіперболічний тангенс. На відміну від логістичної функції гіперболічний тангенс приймає значення різних знаків, що для ряду мереж виявляється доцільнішим.

Кажучи про можливу класифікацію ШНМ, важливо відзначити існування бінарних і аналогових мереж. Перші з них оперують з двійковими сигналами, і вихід кожного нейрона може приймати тільки два значення: логічний нуль ("загальмований" стан) і логічна одиниця ("збуджений" стан). У аналогових мережах вихідні значення нейронів здатні приймати безперервні значення. Ще одна класифікація ділить ШНМ на синхронні і асинхронні. У першому випадку в кожен момент часу свій стан міняє лише один нейрон. У другому – стан міняється відразу у цілої групи нейронів, як правило, у всього шару. Для програмних імітаторів нейронних мереж на цифрових ЕОМ, питання пов'язані з синхронізацією вирішуються комп'ютером, на якому реалізуються ШНМ.

Розглянута проста модель штучного нейрона істотно спрощує ряд властивостей свого біологічного двійника. Наприклад, вона не приймає до уваги затримки в часі, які впливають на динаміку системи. Вхідні сигнали відразу ж породжують вихідний сигнал. І, що важливіше, вона не враховує дію синхронізуючої функції біологічного нейрона, яку ряд дослідників вважає вирішальною. Не дивлячись на ці обмеження, мережі, побудовані з цих нейронів, виявляють властивості, що сильно нагадують біологічну систему.

При складанні нейронних мереж (при їх конструкції) строгих обмежень немає. Завдання полягає лише в тому, щоб на виходах формувалися необхідні сигнали. Можливості в цьому питанні безмежні, але зазвичай використовують декілька стандартних архітектур, з яких при деяких невеликих модифікаціях будують більшість використовуваних мереж.

ДОДАТОК Б

АРХІТЕКТУРА ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

Б.1 Архітектури найбільш поширених ШНМ

Хоча один нейрон і здатний виконувати прості процедури розпізнавання, сила нейронних обчислень виникає від з'єднань нейронів в мережах.

Широкий круг завдань, вирішуваний ШНМ, не дозволяє в даний час створювати універсальні, потужні мережі, вимушуючи розробляти спеціалізовані ШНМ, що функціонують по різних алгоритмах. На рис. Б.1 показані варіанти найбільш поширених класичних архітектур нейронних мереж. Глибокі нейронні мережі в роботі не розглядаються.

Вибір структури ШНМ здійснюється відповідно до особливостей і складності завдання. Якщо ж завдання не може бути зведене ні до одного з відомих типів, розробникові доводиться вирішувати складну проблему синтезу нової конфігурації. При цьому він керується декількома основоположними принципами: можливості мережі зростають із збільшенням числа осередків мережі, щільності зв'язків між ними і числом виділених шарів; введення зворотних зв'язків разом із збільшенням можливостей мережі піднімає питання про динамічну стійкість мережі; складність алгоритмів функціонування мережі (зокрема, наприклад, введення декількох типів синапсів – збудливих, гальмуючих і ін.) також сприяє посиленню потужності ШНМ. Питання про необхідні і достатні властивості мережі для вирішення того або іншого роду завдань є цілим напрямом нейрокомп'ютерної науки. Оскільки проблема синтезу ШНМ сильно залежить від вирішуваної задачі, дати загальні докладні рекомендації складно. В більшості випадків оптимальний варіант вибирається на основі інтуїтивного підбору. Єдина жорстка вимога, що пред'являється архітектурою до елементів мережі, це відповідність розмірності вектора вхідних сигналів мережі числу її входів.

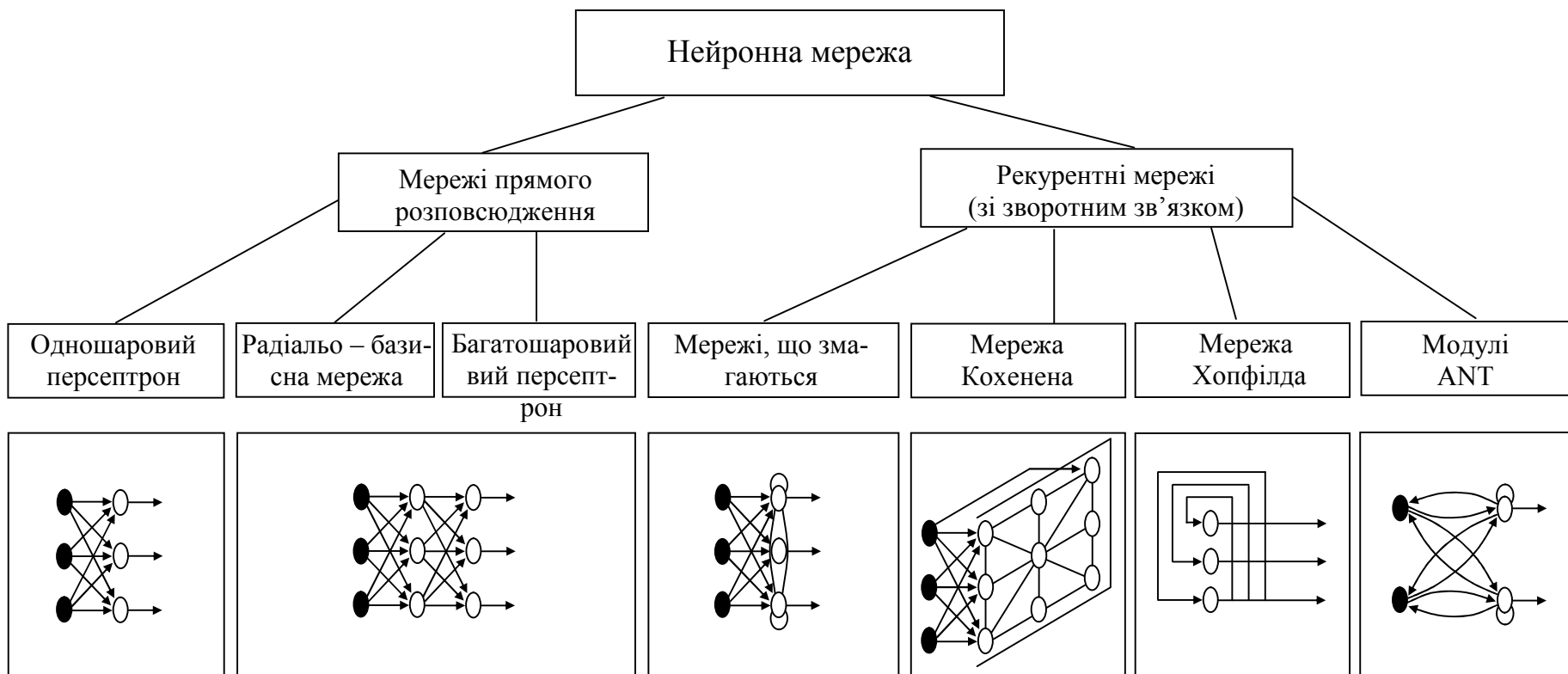


Рисунок Б.1 – Варіанти найбільш поширених класичних архітектур нейронних мереж

Б.2 Штучні нейронні мережі прямого розповсюдження

Найбільшого застосування при побудові нейрорегуляторів систем автоматичного управління знайшли мережі прямого розповсюдження без зворотних зв'язків (багатошарові персептрони). Розглянемо їх більш детально.

Проста одношарова мережа, що складається з m нейронів, показана на рис. Б.2.

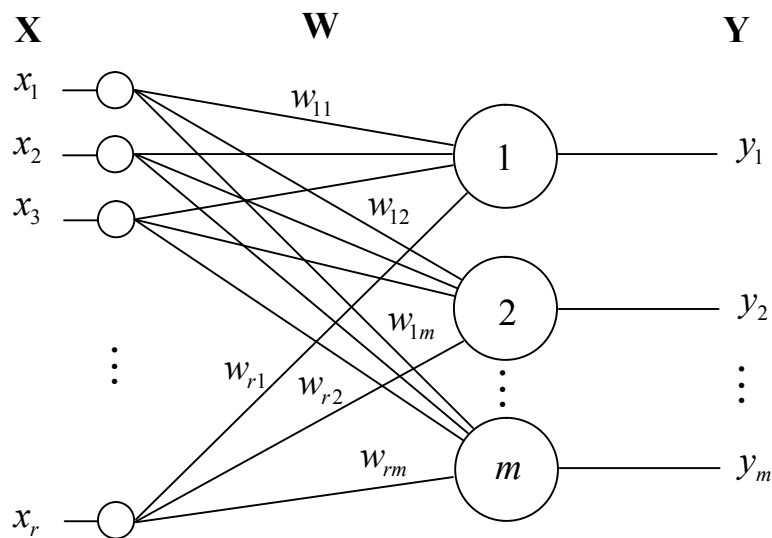


Рисунок Б.2 – Одношарова нейронна мережа

На r входів поступають сигнали, що проходять по синапсах на m нейрони, що створюють єдиний шар цієї ШНМ, які видають вихідні сигнали:

$$y_j = f \left[\sum_{i=1}^r x_i w_{ij} \right], \quad (\text{Б.1})$$

де $j = 1, 2, \dots, m$.

У штучних і природних (біологічних) мережах багато з'єднань можуть бути відсутніми, всі з'єднання показані в цілях спільності.

Очевидно, що всі вагові коефіцієнти синапсів одного шару нейронів можна звести в матрицю W , у якій кожен елемент w_{ij} задає величину i -го синапти-

чного зв'язку j -го нейрона. Таким чином процес, що відбувається в ШНМ, може бути записаний в матричній формі:

$$\mathbf{Y} = f(\mathbf{XW})$$

де $\mathbf{X}$ і $\mathbf{Y}$ – відповідно вхідний і вихідний сигнальні вектори;

$f(\mathbf{V})$ – активаційна функція, вживана поелементно до компонентів вектора $\mathbf{V} = \mathbf{XW}$.

На рис. Б.3 представлена двошарова ШНМ, отримана з одношарової (див. рис. Б.2) шляхом додавання другого шару, що складається з l нейронів. Тут доречно відзначити важливість ролі розглянутої раніше нелінійності активаційної функції, оскільки, якби вона не володіла даною властивістю або не входила в алгоритм роботи кожного нейрона, результат функціонування будь-якої k -шарової ШНМ з ваговими матрицями $\mathbf{W}^{(i)}$, де $i = 1, 2, \dots, k$ для кожного шару i зводився б до перемножування вхідного вектора сигналів $\mathbf{X}$ на матрицю

$$\mathbf{W}^{(\Sigma)} = \mathbf{W}^{(1)}\mathbf{W}^{(2)}\dots\mathbf{W}^{(k)},$$

тобто така k -шарова еквівалентна одношаровій ШНМ з ваговою матрицею єдиного шару $\mathbf{W}^{(\Sigma)}$:

$$\mathbf{X} = \mathbf{YW}^{(\Sigma)}.$$

Таким чином, для розширення обчислювальних можливостей багатошарових ШНМ в порівнянні з одношаровими ШНМ необхідно використання нелінійних активаційних функцій.

Необхідно відзначити, що нелінійність іноді може вводитися і в синаптичні зв'язки. У більшості відомих на сьогоднішній день ШНМ для знаходження зваженої суми входів нейрона використовують формулу (Б.1), проте в деяких додатках ШНМ корисно ввести інший запис, наприклад:

$$S = \sum_{i=1}^r x_i^2 w_{ij}$$

або

$$S = \sum_{i=1}^r x_i x_{(i+1) \bmod r} w_{ij}.$$

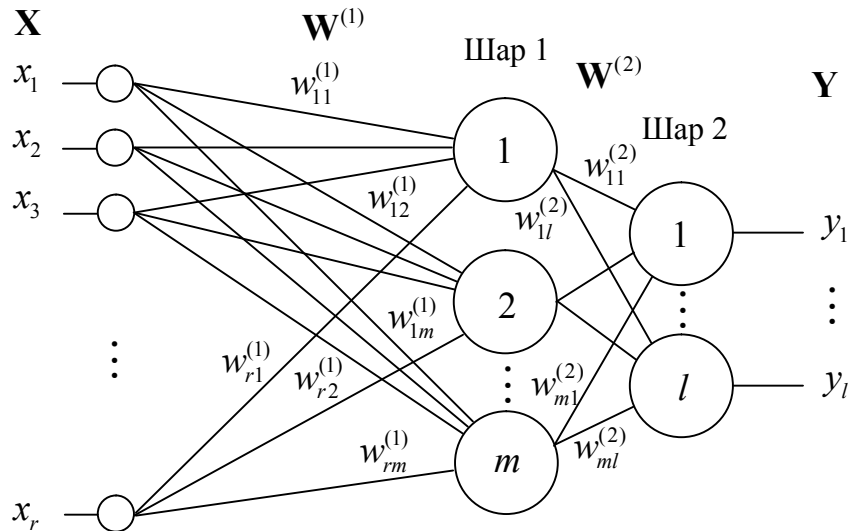


Рисунок Б.3 – Двошарова нейронна мережа

Питання в тому, щоб розробник ШНМ чітко розумів, для чого він це робить, якими цінними властивостями він тим самим додатково наділяє нейрон, і яких позбавляє. Введення такого роду нелінійностей, взагалі кажучи, збільшує обчислювальну потужність мережі, тобто дозволяє з меншого числа нейронів з "нелінійними" синапсами сконструювати ШНМ, що виконує роботу звичайної ШНМ з великим числом стандартних нейронів і складнішою конфігурацією.

У мереж, розглянутих до цих пір, не було зворотних зв'язків, тобто з'єднань, що йдуть від виходів деякого шару до входів цього ж шару або попередніх шарів. Цей спеціальний клас мереж, званих мережами без зворотних зв'язків або мережами прямого розповсюдження, представляє інтерес і широко використовується. В даній магістерській роботі використовується саме цей тип нейронних мереж.

Декілька слів необхідно сказати про необхідну потужність вихідного шару мережі, що виконує остаточну класифікацію простору станів. Річ у тому, що для розділення множини вхідних образів, наприклад, по двох класах достатньо

всього одного виходу. При цьому кожен логічний рівень – "1" і "0" – позначатиме окремий клас. На двох виходах можна закодувати вже чотири класи і т.д. Проте результати роботи мережі, організованої таким чином, недостатньо надійні. Для підвищення достовірності класифікації бажано ввести надмірність шляхом виділення кожному класу одного нейрона у вихідному шарі або, що ще краще, декілька, кожний з яких навчається визначати приналежність конкретних станів входів до певного класу з своєю мірою достовірності, наприклад високої, середньої і низької. Такі ШНМ. дозволяють проводити класифікацію входних неявно виражених станів, об'єднаних в нечіткі (розмиті або пересічні) множини. Ця властивість дозволяє широко використовувати ШНМ в практичних застосуваннях.

ДОДАТОК В

НАВЧАННЯ ШТУЧНИХ НЕЙРОННИХ МЕРЕЖ

В.1 Загальні питання навчання нейронних мереж

Коли йде розмова про використання ШНМ і нейромережових алгоритмів, майже завжди розуміються певні процедури їх навчання. ШНМ є адаптивною системою, життєвий цикл якої складається з двох незалежних фаз – навчання і роботи мережі. Навчання вважається закінченим, коли мережа правильно виконує перетворення на тестових прикладах і подальше навчання не викликає значної зміни вагових коефіцієнтів, що настроюються. Далі мережа виконує перетворення раніше невідомих їй даних на основі сформованої нею в процесі навчання нелінійної моделі процесу. Мережа успішно працює до тих пір, поки істотно не зміниться реальна модель явища (наприклад, у разі виникнення ситуації, інформація про яку ніколи не пред'являлася мережі при навчанні), що відображається. Після цього мережа може дообучатися з урахуванням нової інформації, причому при дообучанні попередня інформація не втрачається, а узагальнюється з урахуванням інформації, що знов поступила. При "пошкодженні" частини вагових коефіцієнтів ШНМ її властивості можуть бути повністю відновлені в процесі дообучання. Від того, наскільки якісно буде виконаний етап навчання ШНМ, залежить здатність мережі вирішувати поставлені перед нею проблеми під час експлуатації. Теорія навчання розглядає три фундаментальні властивості, пов'язаних з навчанням по прикладах: місткість, складність зразків і обчислювальна складність. Під місткістю розуміється, скільки зразків може запам'ятати мережу і які функції і межі ухвалення рішень можуть бути на ній сформовані. Складність зразків визначає число навчальних прикладів, необхідних для досягнення здатності мережі до узагальнення. Важливою характеристикою є час, що витрачається на навчання. Як правило, час навчання і якість навчання зв'язані зворотною залежністю і вибирати ці параметри доводиться на основі компромісу.

Існують три парадигми навчання "з вчителем", "без вчителя" (самонавчання) і змішана.

У свою чергу множина різних алгоритмів навчання діляться на два великі класи детерміністських і стохастичних алгоритмів. В першому з них підстроювання вагів є жорсткою послідовністю дій, в другому – воно проводиться на основі дій, що підкоряються деякому випадковому процесу

Навчання з вчителем припускає, що для кожного вхідного вектора існує цільовий вектор, що є необхідним виходом. Разом вони називаються навчальною парою. Зазвичай мережа навчається на деякому числі таких навчальних пар.

Коли в мережі тільки один шар, алгоритм її навчання з вчителем досить очевидний, оскільки правильні вихідні стани нейронів єдиного шару заздалегідь відомі, і підстроювання синаптичних зв'язків йде в напрямі, що мінімізує помилку на виході мережі.

У багатошарових же мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, не відомі, і двох або більш шарову ШНМ вже неможливо навчити, керуючись тільки величинами помилок на виходах цієї мережі. Один з варіантів вирішення цієї проблеми – розробка наборів вихідних сигналів, відповідних вхідним, для кожного шару ШНМ, що, звичайно, є дуже трудомісткою операцією і не завжди здійсненою. Другий варіант – динамічне підстроювання вагових коефіцієнтів синапсів, в ході якої вибираються, як правило, найбільш слабкі зв'язки, які змінюються на малу величину в ту або іншу сторону. Зберігаються ж тільки ті зміни, які спричинили зменшення помилки на виході всієї мережі. Очевидно, що даний метод "проб", не дивлячись на свою простоту, що здається, вимагає громіздких рутинних обчислень. І, нарешті, третій, більш прийнятний варіант – розповсюдження сигналів помилки від виходів ШНМ до її входів, в напрямі зворотному прямому розповсюдженню сигналів в звичайному режимі роботи. Цей алгоритм навчання ШНМ отримав назву процедури зворотного розповсюдження (back-propagation algorithm) і є найширше використовуваним. Саме він буде докладніший розглянутий надалі, оскільки використовується в магістерській роботі.

В.2 Явище перенавчання

Одна з найбільш серйозних труднощів при навчанні мережі полягає в тому, що у ряді випадків ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати; потрібно мінімізувати помилку, яка з'являється в мережі, коли на неї подаються абсолютно нові спостереження. Вельми важливо, щоб нейронна мережа володіла здатністю пристосовуватися до цих нових спостережень. Що ж відбувається насправді? Мережа навчається мінімізувати помилку на деякій обмеженій повчальній множині. Це не відповідає вимогам теорії про наявність ідеальної і нескінченно великої повчальної множини. І це не відповідає тій реальній ситуації, коли треба мінімізувати конкретну функцію помилок для наперед невідомої моделі.

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

При роботі з нейронними мережами користувач стикається з тією ж проблемою. Мережі з великою кількістю вагів дозволяють відтворювати дуже складні функції, і в цьому сенсі вони схильні до перенавчання. Мережа ж з невеликою кількістю вагів може опинитися недостатньо гнучкою, щоб змоделювати наявну залежність. Наприклад, одношарова лінійна мережа здатна відтворювати тільки лінійні функції. Якщо використовувати багат шарові лінійні ме-

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

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

никає ризик перенавчання нейронної мережі на контрольній множині. Для того, щоб гарантувати надійність вибраної моделі мережі, резервують ще одне – тестова безліч спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на повчальній і контрольній множинах реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки 1 раз: якщо його використовувати повторно для коректування процесу навчання, то воно фактично перетвориться на контрольну множину.

Отже, процедура побудови нейронної мережі складається з наступних кроків:

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

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

- виявлення ефекту перенавчання і коректування конфігурації мережі.

В.3 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 непередставницької 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 всьо-

го постаратися зробити так, щоб спостереження різних типів були представлені рівномірно, і відповідно цьому інтерпретувати результати, які видає мережа.

Здатність мережі, навченої на деякій множині 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 відповідність між входами і бажаними значеннями на виході. Така процедура навчання носить назву адаптації і достатньо широко застосовується для настройки параметрів нейронних мереж.

В.4 Метод зворотного розповсюдження помилки

Згідно методу найменших квадратів, цільовою функцією помилки ШНМ, що мінімізується, є величина

$$E(w) = \frac{1}{2} \sum_{jp} (y_{jp}^{(N)} - d_{jp})^2, \quad (\text{В.1})$$

де $y_{jp}^{(N)}$ – реальний вихідний стан нейрона j у вихідному шару N нейронної мережі при подачі на її входи p -го образу;

d_{jp} – ідеальний (бажаний) вихідний стан цього нейрона.

Підсумовування ведеться по всіх нейронах вихідного шару і по всіх оброблюваних мережею образах. Мінімізація ведеться методом градієнтного спуску, що означає підстроювання вагових коефіцієнтів таким чином

$$\Delta w_{ij}^{(n)} = -\eta \frac{\partial E}{\partial w_{ij}}. \quad (\text{В.2})$$

Тут w_{ij} – ваговий коефіцієнт синаптичного зв'язку, що сполучає i -й нейрон шару $(n-1)$ з j -м нейроном шару n

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j} \frac{\partial s_j}{\partial w_{ij}}. \quad (\text{B.3})$$

Тут під y_j як і раніше, розуміється вихід нейрона j , а під s_j – зважена сума його вхідних сигналів, тобто аргумент активаційної функції. Множник $\frac{\partial y_j}{\partial s_j}$ є похідною цієї функції по її аргументу. З цього виходить, що похідна активаційною функція повинна бути визначена на всій осі абсцис U зв'язку з цим функція одиничного стрибка і інші активаційні функції з неоднорідностями не підходять для даних ШНМ. У них застосовуються такі гладкі функції, як гіперболічний тангенс або класичний сигмоїд з експонентою. У разі гіперболічного тангенса

$$\frac{\partial y}{\partial s} = 1 - s^2. \quad (\text{B.4})$$

Третій множник $\frac{\partial s_j}{\partial w_{ij}}$, очевидно, рівний виходу нейрона попереднього шару $y_i^{(n-1)}$.

Що стосується першого множника в (B.3), він легко розкладається таким чином

$$\frac{\partial E}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} \frac{\partial s_k}{\partial y_j} = \sum_k \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial s_k} w_{jk}^{(n+1)}. \quad (\text{B.5})$$

Тут підсумовування по k виконується серед нейронів шару $(n+1)$. Ввівши нову змінну

$$\delta_j^{(n)} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial s_j}, \quad (\text{B.6})$$

отримаємо рекурсивну формулу для розрахунків величин $\delta_j^{(n)}$ шару n з величин $\delta_k^{(n+1)}$ більш старшого шару $(n+1)$

$$\delta_j^{(n)} = \left[\sum_k \delta_k^{(n+1)} w_{jk}^{(n+1)} \right] \frac{\partial y_j}{\partial s_j}. \quad (\text{B.7})$$

Для вихідного ж шару

$$\delta_j^{(N)} = (y_j^{(N)} - d_j) \frac{\partial y_j}{\partial s_j}. \quad (\text{B.8})$$

Тепер можна записати (B.2) в розкритому вигляді

$$\Delta w_{ij}^{(n)}(t) = -\eta \delta_j^{(n)} y_i^{(n-1)}. \quad (\text{B.9})$$

Іноді для додання процесу корекції вагів деякої інерційності, що згладжує різкі скачки при переміщенні по поверхні цільової функції, (B.1) доповнюється значенням зміни ваги на попередній ітерації

$$\Delta w_{ij}^{(n)}(t) = -\eta [\mu w_{ij}^{(n)}(t-1) + (1-\mu) \delta_j^{(n)} y_i^{(n-1)}] \quad (\text{B.10})$$

де μ – коефіцієнт інерційності t – номер поточної ітерації.

Таким чином, повний алгоритм навчання ШНМ за допомогою процедури зворотного розповсюдження будується в наступному порядку.

1. Подати на входи мережі один з можливих образів і в режимі звичайного функціонування ШНМ, коли сигнали розповсюджуються від входів до виходів, розрахувати значення останніх. Як було відмічено раніше

$$s_j^{(n)} = \sum_{i=1}^M y_i^{(n-1)} w_{ij}^{(n)}, \quad (\text{B.11})$$

де M – число нейронів в шарі $(n-1)$ з урахуванням нейрона з постійним вихідним станом $+1$, що задає зсув;

$$y_i^{(n-1)} = x_{ij}^{(n)} - i\text{-й вхід нейрона } j \text{ шару } n$$

$$y_j^{(n)} = f(s_j^{(n)}) \quad (\text{B.12})$$

$f()$ – сигмоїд.

$$y_q^{(0)} = I_q \quad (\text{B.13})$$

I_q – q -а компоненту вектора вхідного образу.

2. Розрахувати $\delta^{(N)}$ для вихідного шару по формулі (B.8).

Розрахувати по формулі (B.9) або (B.10) зміни вагів $\Delta w^{(N)}$ шару N .

3. Розрахувати по формулах (B.7) і (B.9) (або (B.7) і (B.10)) відповідно $\delta^{(n)}$ і $\Delta w^{(n)}$ для решти всіх шарів $n = N - 1, \dots, 1$.

4. Скоректувати всі ваги в ШНМ

$$w_{ij}^{(n)}(t) = w_{ij}^{(n)}(t-1) + \Delta w_{ij}^{(n)}(t). \quad (\text{B.14})$$

5. Якщо помилка мережі істотна, перейти на крок 1, інакше – кінець.

На кроці 1, на входи мережі поперемінно у випадковому порядку пред'являються всі тренувальні образи, щоб мережа, образно кажучи, не забувала одні у міру запам'ятовування інших.

З виразу (B.9) виходить, що, коли вихідне значення $y_i^{(n-1)}$ прагне до нуля, ефективність навчання помітно знижується. При двійкових вхідних векторах в середньому половина вагових коефіцієнтів не коректуватиметься, тому область можливих значень виходів нейронів $[0,1]$ бажано зрушити в межі $[-0.5, +0.5]$, що досягається простими модифікаціями логістичних функцій. Наприклад, сигмоїд з експонентою перетвориться до вигляду

$$f(x) = -0,5 + \frac{2}{2 + e^{-\alpha x}}. \quad (\text{B.15})$$

Даний алгоритм навчання ШНМ має декілька "вузьких місць". По-перше, в процесі навчання може виникнути ситуація, коли великі позитивні або негативні значення вагових коефіцієнтів змістять робочу точку на сигмоїдах багатьох нейронів в область насичення. Малі величини похідної від логістичної функції приведуть у відповідність з (B.14) і (B.15) до зупинки навчання, що паралізує ШНМ. По-друге, застосування методу градієнтного спуску не гарантує,

що буде знайдений глобальний, а не локальний мінімум цільової функції. Ця проблема пов'язана ще з однією, а саме – з вибором величини швидкості навчання.

Доказ збіжності навчання в процесі зворотного розповсюдження заснований на обчисленні похідних, тобто прирості вагів i , отже, швидкість навчання повинна бути нескінченно малою, проте в цьому випадку навчання відбуватиметься неприйнятно поволі. З іншого боку, дуже великі корекції вагів можуть привести до постійної нестійкості процесу навчання. Тому в якості η зазвичай вибирається число менше 1, але не дуже маленьке, наприклад 0,1, і воно, в загальному випадку, може поступово зменшуватися в процесі навчання. Крім того, для виключення випадкових попадань в локальні мінімуми іноді, після того, як значення вагових коефіцієнтів за стабілізуються, η короткочасно сильно збільшують, щоб почати градієнтний спуск з нової точки. Якщо повторення цієї процедури кілька разів приведе алгоритм в один і той же стан ШНМ, можна більш менш упевнено сказати, що знайдений глобальний мінімум.

Алгоритми, що модифікують процедуру зворотного розповсюдження, змінюють процедуру вибору функції критерію, що оптимізується, вибір оптимального кроку і правила корекції параметрів навчаної мережі детально описані в багатьох роботах.

Відзначимо, що всі ці алгоритми не узгоджуються з даними нейробіології, де показано, що сигнали в біологічних нейронних мережах можуть розповсюджуватися тільки в одному, прямому напрямі.

Запропоновані також алгоритми настройки багат шарової ШНМ без використання процедури зворотного розповсюдження помилки, які отримали назву «Автономні алгоритми». Ці алгоритми засновані на тому факті, що для багат шарової ШНМ існує деяка множина значень параметрів, що настраюються, на якій вектор корекції параметрів, що настраюються, обчислюється досить просто. Ці алгоритми за інших рівних умов гірше вирішують задачу навчання, ніж алгоритми зворотного розповсюдження помилки, проте вони є структурно незалежними і дуже простими з обчислювальної точки зору.

ДОДАТОК Г

ІДЕНТИФІКАЦІЯ ДИНАМІЧНИХ ОБ'ЄКТІВ З ЗАСТОСУВАННЯМ НЕЙРОННИХ МЕРЕЖ

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

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

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

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

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

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

ξ – завада вимірювання вихідного сигналу.

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

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

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

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

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

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

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

Рівняння БП має вигляд

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

де q – кількість куль в мережі;

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

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

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

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

де c_i – вагові коефіцієнти, які підлягають визначенню.

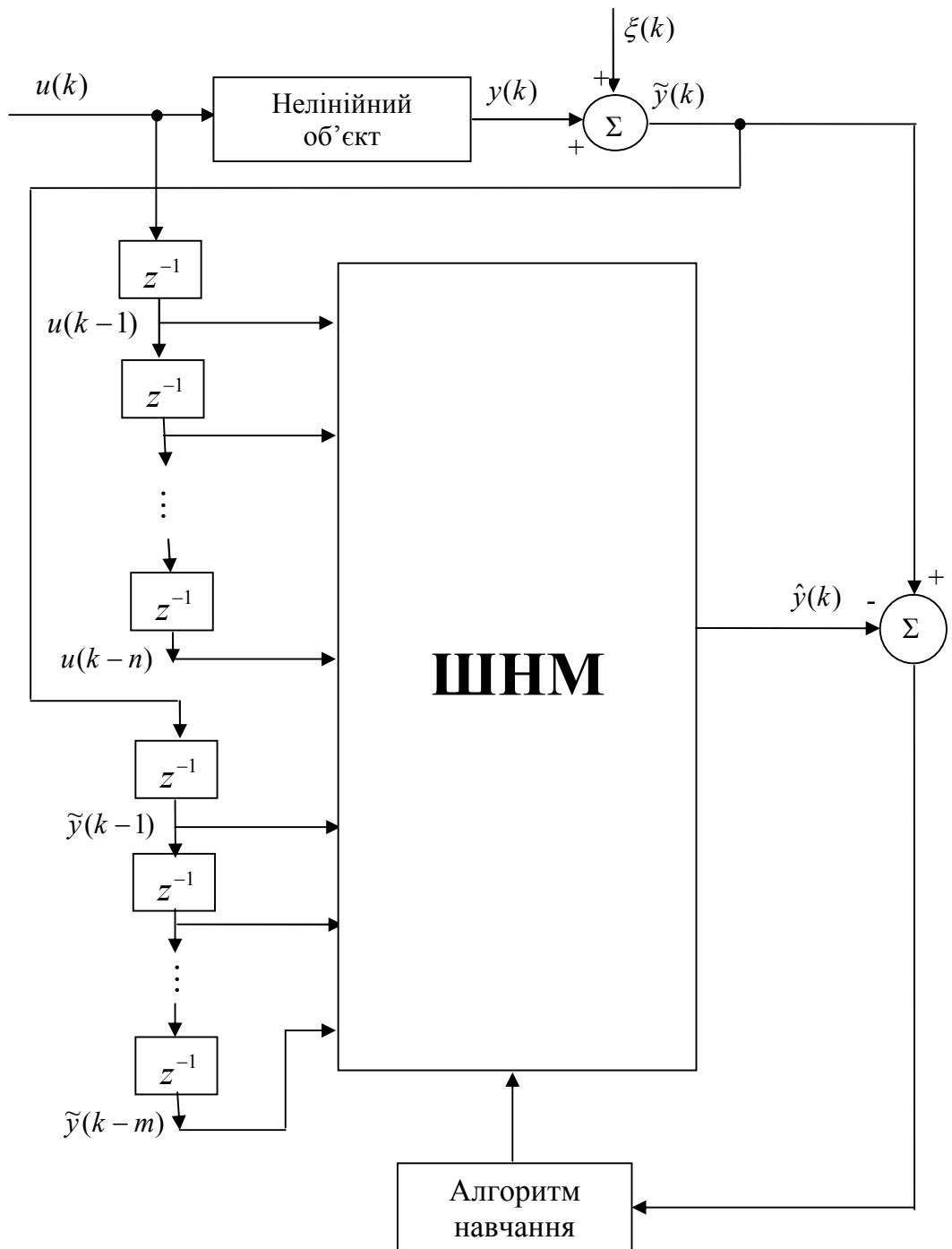


Рисунок Г.1 – Схема ідентифікації динамічного об'єкту

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

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

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

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

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

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

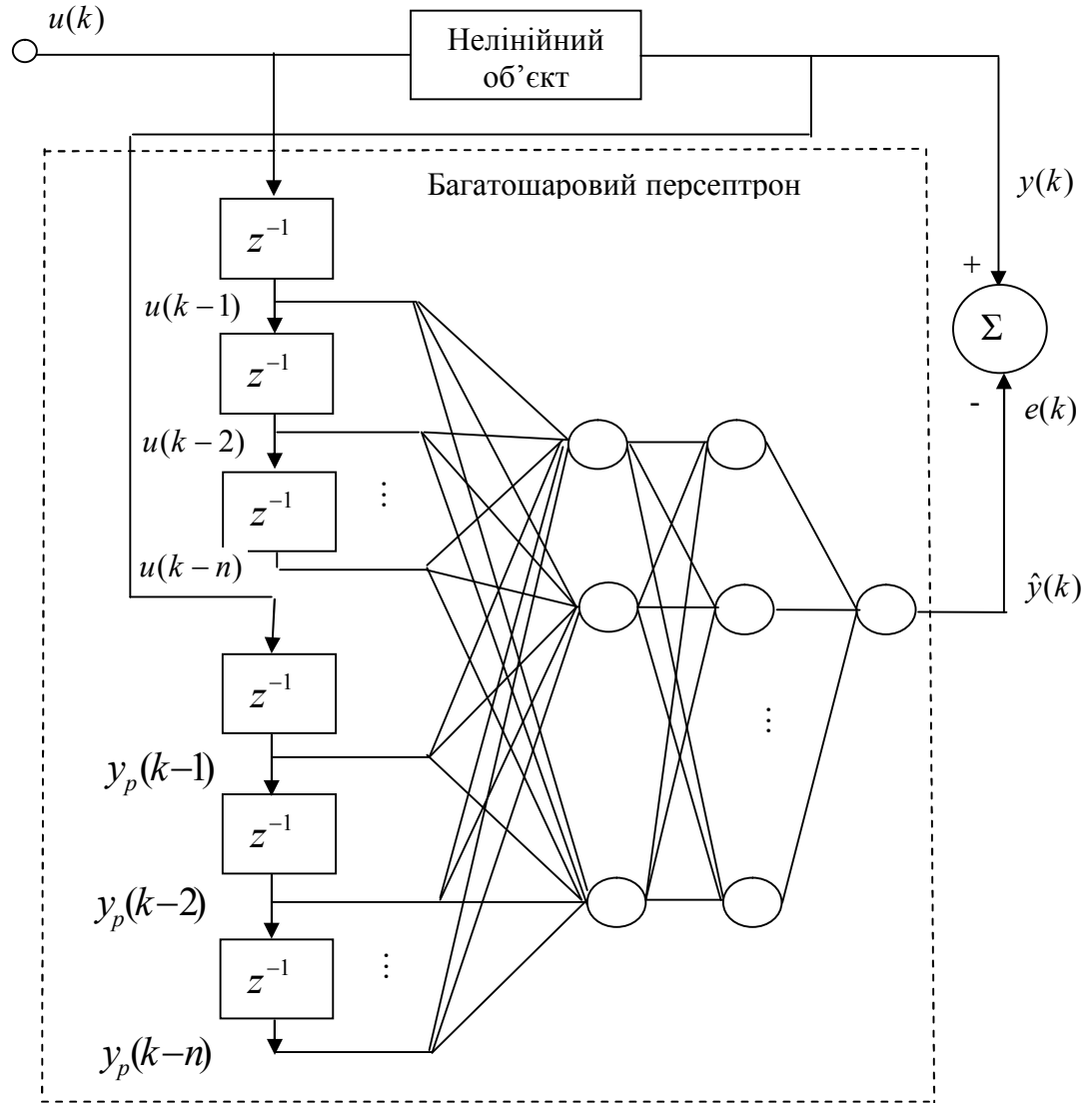


Рис Г.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

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

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

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

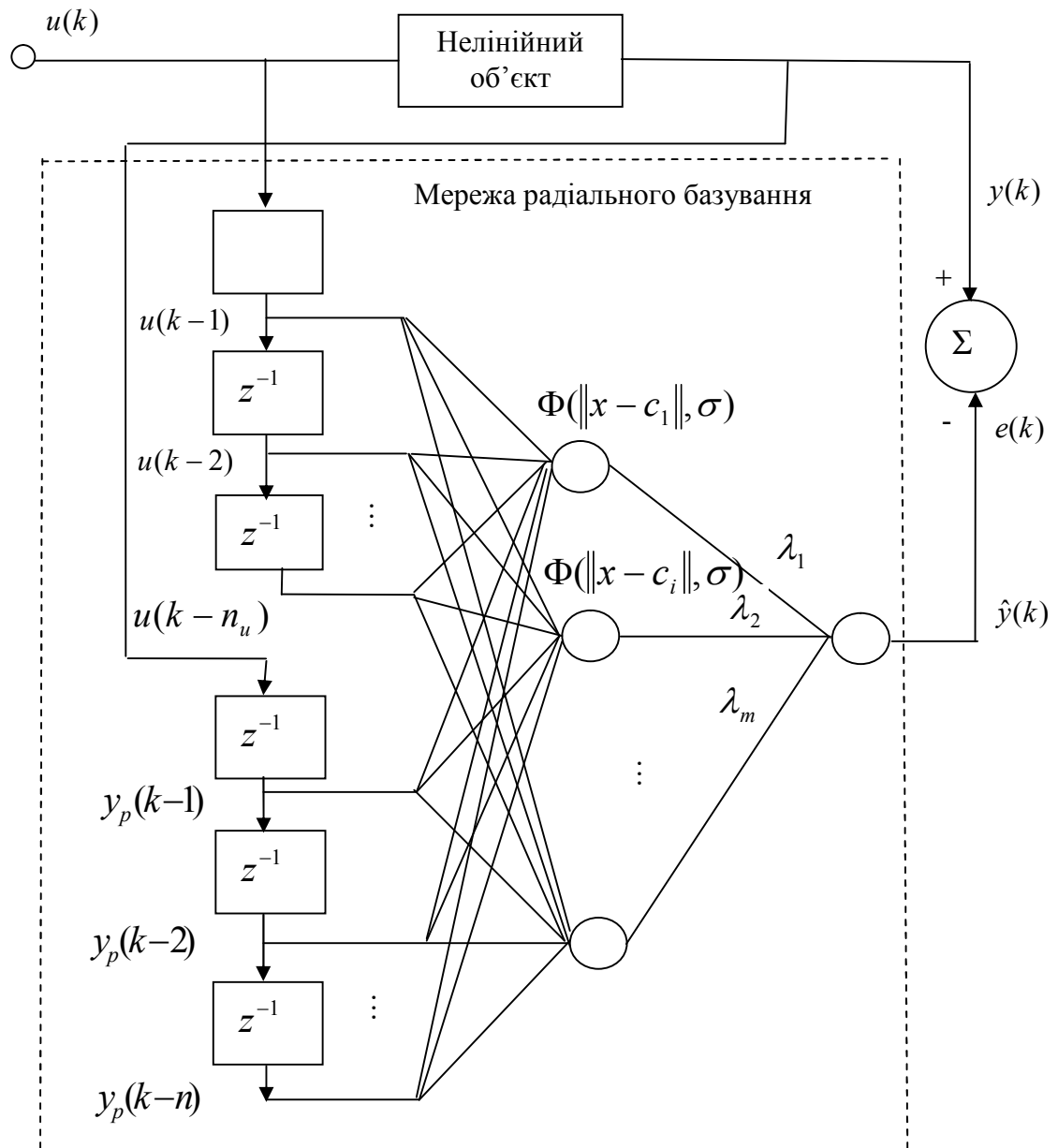


Рисунок Г.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
% Система управління електроприводом повороту  
% стріли роторного екскаватора  
% 2025 рік
```

```
% Вихідні дані
```

```
kzv=1.97;  
ks=4.8;  
Tg=0.37;  
Tzc=0.01;  
Te=0.075;  
Re=0.17408;  
CF=2*3.42;  
In=192;  
Mn=CF*In;  
Jd=2*2;  
Jm=8.74;  
b12=0.02;  
b12=0;  
c12=50;  
Js=Jd+Jm;  
Uz=218.4;  
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(4,5);  
C(1,1)=1;C(2,2)=1; C(3,3)=1;C(4,4)=1;  
D=zeros(4,2);
```