

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

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

(тема кваліфікаційної роботи)

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

(код і найменування спеціальності)

_____/ Наталія ЛУКА
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

Харків – 2024

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

ЗАВДАННЯ

на кваліфікаційну роботу магістерського рівня вищої освіти

студенту (ці) Наталія ЛУКА
(ім'я, прізвище)

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

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

2. Термін здачі закінченої роботи "5" грудня 2024 р.

3. Вихідні дані до роботи Теоретична продуктивність по рихлій масі $Q=1700 \text{ м}^3 \cdot \text{год}$; діаметр роторного колеса $D_p=8 \text{ м}$; бічне зусилля різання $F_6=1,2 \cdot 10^6 \text{ Н}$; питоме зусилля копання $k_f=3,8 \cdot 10^4 \text{ кг/м}^2$; радіус різання $R_0=23,3 \text{ м}$; кутова швидкість повороту платформи $\omega_{пл}=2,9 \cdot 10^{-2} \text{ с}^{-1}$; маса поворотної платформи $m_{пл}=500 \cdot 10^3 \text{ кг}$; маса приймальної частини $m_{пч}=127 \cdot 10^3 \text{ кг}$; маса відвальної частини $m_{пч}=120 \cdot 10^3 \text{ кг}$; частота обертання ротора $\omega_p=6 \text{ рад/хв}$; момент інерції поворотної платформи $J_{пл}=17 \cdot 10^7 \text{ кг} \cdot \text{м}^2$; діаметр роликового круга $D_{кр}=8,7 \text{ м}$. Перерегулювання не більше 10%, час регулювання до 5 с.

4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережових технологій для управління багатомасовою електро-механічною 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 нейромережової 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. Нейромережева ідентифікація. NARX моделі. 6. Нейромережеві системи управління. Структурні схеми. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси. 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14-15. Магістерська кваліфікаційна робота. Висновки

6. Консультант:

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

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

Керівник

(підпис)

Тетяна ВАСИЛЕЦЬ.
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Наталія ЛУКА
(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.	09.09.2024- 18.09.2024	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження статичних і динамічних характеристик системи.	19.09.2024- 15.10.2024	
3	Дослідження системи з урахуванням пружних елементів. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2024- 05.11.2024	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2024- 25.11.2024	
5	Оформлення роботи	26.11.2024- 05.12.2024	

Студент (ка)

(підпис)

Наталія ЛУКА
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 214 сторінок, 70 рисунків, 2 таблиці, 72 використаних джерел, 4 додатків.

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

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

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

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

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

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

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

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

ABSTRACT

An explanatory message consists of: 214 pages, 70 figures, 2 tables, 72 used sources, 4 additions.

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

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

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

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

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

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

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

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	14
1.1 Розвиток теорії штучних нейронних мереж.....	14
1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи.....	17
1.3 Загальні відомості про роторні екскаватори	39
1.4 Аналіз факторів, які обумовлюють вибір системи управління електроприводом механізму повороту стріли роторного екскаватора	42
1.5 Постановка задачі дослідження.....	45
ВИСНОВКИ ДО РОЗДІЛУ 1	49
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПОВОРОТУ СТРИЛИ РОТОРНОГО ЕКСКАВАТОРА	41
2.1 Основні технічні характеристики роторного екскаватора. Вимоги до системи управління.....	51
2.2 Розрахунок потужності і вибір привідних електродвигунів	53
2.3 Розрахунок швидкісної діаграми механізму повороту стріли екскаватора	54
2.4 Побудова діаграми навантаження	58
2.5 Перевірка двигунів по нагріву і перевантажувальній здатності.	64
2.6 Вибір основного електроустаткування електроприводу	66
2.7 Побудова статичних характеристик електроприводу Г-Д механізму повороту стріли екскаватора	67
2.8 Побудова статичних характеристик замкнутої системи.....	72
2.9 Дослідження динамічних характеристик системи	78
ВИСНОВКИ ДО РОЗДІЛУ 2	97

РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ	98
3.1 Математична модель двомасової системи	98
3.2 Розрахунок перехідних процесів двомасової системи	104
ВИСНОВКИ ДО РОЗДІЛУ 3	110
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	111
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	111
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	113
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	116
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	159
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	161
ВИСНОВКИ ДО РОЗДІЛУ 4	167
ВИСНОВКИ	169
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	171
ДОДАТОК А. БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ	178
А.1 Структура штучного нейрона	178
А.2 Одношарові нейронні мережі	179
А.3 Багатошарові нейронні мережі	182
А.4 Здібності апроксимації і узагальнення багатошарових мереж.....	183
ДОДАТОК Б. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ.....	186
Б.1 Завдання навчання	186
Б.2 Процедури адаптації і навчання.....	187
Б.3 Явище перенавчання	189
Б.4 Властивість узагальнення	192
Б.5 Вибір методу навчання нейронної мережі	194

ДОДАТОК В. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ.....	203
В.1 Моделі систем, використовувані при нейромережевій ідентифікації	203
В.2 Нейронні мережі як універсальні моделі.....	208
В.3 Ідентифікація динамічних систем за допомогою нейронних мереж .	211
ДОДАТОК Г. ФАЙЛ ВИХІДНИХ ДАНИХ.....	214

ВСТУП

Актуальність теми. Вдосконалення систем автоматизованого управління промисловими установками засноване на новітніх наукових та технічних досягненнях і є критично важливим для підвищення продуктивності виробництва, зростання ефективності праці та покращення якості виготовляної продукції. Еволюція автоматизованих систем управління включає в себе розширення практичного застосування адаптивного управління, вдосконалення векторного управління електроприводами із змінним струмом, широке використання управляючих обчислювальних систем на різних рівнях складності, розвиток прямого цифрового управління та інших технологічних напрямів. Одним із перспективних шляхів є використання систем управління на основі нейромереж.

Системи управління з застосуванням нейромереж належать до нелінійних динамічних систем. У цих системах штучна нейронна мережа може виконувати різноманітні функції. Це, по-перше, функція адаптивного регулятора нелінійного багатозв'язаного об'єкту. Тут можливі два варіанти функціонування нейронних мереж. По-перше – нейронна мережа навчається і одночасно формує дію, що управляє, на вході виконавчого пристрою системи управління. Мета навчання мережі і мета управління об'єктом співпадають, що відбивається в заданні єдиній цільовій функції системи.

Мережа може навчатися в реальному часі, синхронізуючись із темпом протікання процесів у системі (режим on-line). У другому варіанті її робота складається з двох етапів: попереднього навчання мережі заданої оптимальної функції управління; відтворення апроксимації цієї функції під час управління об'єктом за аналогічних або подібних умов. При цьому цільові функціонали навчання мережі та управління об'єктом можуть відрізнятися. Такий підхід, відомий як супервізорне управління, є найбільш поширеним на сьогоднішній день. Проте процес синтезу нейромережевого контролера та налаштування його параметрів у цьому випадку відбувається не в реальному часі (режим off-line).

Вибір між підходами on-line або off-line залежить від особливостей завдання і впливає на вибір алгоритму навчання мережі (наприклад, безпошукові чи пошукові схеми, глобальна чи локальна оптимізація тощо). У промислових галузях, де накопичено великі масиви даних про поведінку технічного об'єкта, доцільно застосовувати off-line навчання, використовуючи генетичні алгоритми, алгоритми з елементами випадкового пошуку або методи "статистичного" навчання. Натомість, для об'єктів, характеристики яких змінюються під час експлуатації, найбільш доцільним стає використання on-line алгоритмів адаптивного налаштування мережі.

Нейронні мережі знаходять застосування як ідентифікатори для оцінювання вектора стану нелінійних систем і як розширені фільтри Калмана.

Застосування нейронної мережі як оптимізатора для настройки параметрів регуляторів з типовими законами регулювання і для настройки параметрів алгоритмів адаптації, що реалізуються на основі відомих методів теорії адаптивних систем. Для позначення класу нелінійних систем з такого роду застосуванням штучних нейронних мереж використовується термін нейромережеві системи управління. Найбільше розповсюдження для цілей управління знайшли багаточарові нелінійні нейронні мережі [1-12].

Гірничодобувна промисловість залишається однією з ключових галузей національної економіки. Одним із пріоритетних завдань для розвитку економіки країни є підвищення продуктивності праці у цій сфері. Сучасний етап розвитку галузі відзначається активним поширенням відкритого способу видобутку корисних копалин. У зв'язку з цим на відкритих гірничих роботах реалізується масштабна програма технічного переозброєння та впровадження більш досконалого обладнання.

Роторний екскаватор є основною машиною, що входить в технологічний комплекс розробки корисних копалини відкритим способом, тому розробка систем управління електроприводами, що забезпечують високу продуктивність роторного екскаватора, є актуальним завданням.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління механізмом повороту стріли роторного екскаватора для підвищення показників якості функціонування системи на основі нейромережевих регуляторів. Досягнення поставленої мети здійснюється рішенням наступних основних задач.

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

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

Предметом дослідження є моделі і методи синтезу нейромережевої системи управління двомасовою електромеханічною системою механізму повороту стріли роторного екскаватора з урахуванням податливості механічних зв'язків.

Методи дослідження. Всі теоретичні положення роботи базуються на фундаментальних положеннях теорії автоматичного управління. Методи математичного моделювання використані для побудови математичних моделей роботи системи. Методи теорії штучних нейронних мереж дозволили синтезувати нейромережеву модель об'єкту управління і вибрати процедуру її навчання. Методи нейроуправління використані як основа для розробки нейромережевої системи. Проведене імітаційне моделювання дало змогу оцінити якість роботи системи та підтвердило ефективність отриманих результатів.

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

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

Практичне значення одержаних результатів полягає в розробці нейромережової системи управління двомасовою електромеханічною системою механізму повороту стріли роторного екскаватора, що забезпечує високоякісне регулювання.

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

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

Серед них: синтезована система регулювання координат електроприводу; розроблена модель системи з урахуванням пружних елементів, розроблена структурна схема нейромережової системи; синтезований нейрорегулятор NN

Predictive Controller з використанням пакету прикладних програм Neural Network Toolbox системи MATLAB; виконано моделювання нейромережевої систем і проведено аналіз отриманих результатів.

У роботі наведено посилання на авторів та відповідні джерела при використанні відомих залежностей, експериментальних даних і наукових положень.

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

Структура і об'єм роботи Робота складається з вступу, 4 розділів основної частини, висновків, додатків і списку використаних літературних джерел. Повний об'єм роботи складає 208 сторінок, серед них: 67 рисунків, з яких 46 рисунків за текстом, 21 рисунок на 21 окремій сторінці, 4 таблиці, з яких 1 на 2 повних сторінках, 5 додатків на 56 сторінках, список літературних джерел з 70 найменування на 6 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ УПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Розвиток теорії штучних нейронних мереж

Історія становлення штучних нейронних мереж як наукового напрямку починається з робіт Мак-Каллока, Пітерса [13] і Розенблата [14], що народили свого часу великі надії у кібернетиків, але незабаром незаслужено забутих. Перші спроби використовувати штучні нейронні мережі для управління динамічними об'єктами в «біонічних системах» були зроблені ще в початку 70-х років, наприклад [15,160]. Учені прийшли до висновку, що нейронні структури в біологічних управляючих системах працюють в n -мірному фазовому просторі, розбитому на деяке число областей по критерію вигляду

$$Q = \min \{ \max F(\Psi_i(t), \Psi_i^{(1)}(t), \Psi_i^{(2)}(t), \dots, \Psi_i^{(m)}(t)) \}, \quad i = \overline{1, n}, \quad (1.1)$$

де $\Psi_i(t) = x_i(t) - g_i(t)$ – динамічні помилки регулювання;

$\Psi_i^{(i)}(t)$ – i -а похідна від динамічної помилки;

$x_i(t)$ – поточне значення регульованої величини;

$g_i(t)$ – її задане значення;

$F(\cdot)$ – вибраний функціонал, що характеризує вимоги до системи регулювання.

Нейронна регулююча система виконує функцію навчаного або самонавчаного класифікатора в цьому просторі. Термін «класифікатор» тут означає, що мережу виконує оцінювання змінних фазового простору системи щодо околиці особливих точок (або областей) цього простору. Попадання точки системи, що зображає, що фіксується, в одну з пронумерованих областей фазового простору викликає на виході класифікатора сигнали, що коректують, $\xi_i \Psi_i^{(i)}(t)$, де ξ_i – змінні залежно від положення крапки, що зображає, усередині цієї області па-

раметри. Таким чином, нейронна регулююча структура розглядається як нейронний класифікатор n -мірного фазового простору, а сам процес регулювання зводиться до необхідної зміни вектора полягання в каналах зворотного зв'язку. Подібна ідея «біонічного» управління самими ученими називалася гіпотетичною, але з «...достатньо великим коефіцієнтом правдоподібності». Як приклад біологічних систем управління, що вписуються в запропоновану концепцію побудови нейромережевої системи з класифікатором фазового простору приводилася окодвигунна система тварин. Проте у той час ще не були відомі методи навчання багатошарових нейронних мереж.

Термін «нейроуправління» вперше з'явився в роботах Вербоса в 1974 році [17]. Проте тільки в кінці 80-х років почалося реальне застосування навчених багатошарових нейронних мереж в системах управління. Відродження інтересу до нейронних мереж в ці роки багато в чому пов'язане з ім'ям Д.Румельхарда [18,19], перевідкрившого алгоритм навчання нейронних мереж, відомий як алгоритм зворотного розповсюдження помилки. Впровадження штучних нейронних мереж у сферу управлінських завдань значною мірою було зумовлене роботами Наренди та його співавторів, зокрема [20, 21].

Останнім часом нейронні мережі широко застосовуються для управління динамічними об'єктами. У дослідженні [22] окреслено основні проблеми синтезу нейромережевих систем управління такими об'єктами, зокрема:

- розробка структур нейромережевих систем управління;
- обмеження на швидкість налаштування параметрів мережі;
- модифікація алгоритмів налаштування, що забезпечують малі траєкторні помилки за обмежень на вагові коефіцієнти синаптичних зв'язків нейронів;
- модифікація управлінських підходів, що гарантують стійкість системи за умов неконтрольованих збурень.

Підходи до вирішення цих завдань висвітлено в літературі, присвяченій застосуванню нейронних мереж у задачах управління. У 1992 році [23] було запропоновано процедуру синтезу системи управління нелінійними об'єктами із

використанням RBF-мережі, детально розглянуто питання створення її архітектури та алгоритмів налаштування. Проте питання, пов'язані з якістю процесів і нечутливістю системи до невимірюваних збурень, залишилися невирішеними.

У низці пізніших досліджень [24–31] запропоновано методи синтезу нейромережевих систем управління складними технічними об'єктами. З середини 90-х років у Національному технічному університеті «ХПІ» активно проводяться дослідження з використання методів нейронних мереж для вирішення завдань управління електромеханічними системами. [32, 33]. У роботах продемонстрований виключно високий потенціал поєднання двох обчислювальних технологій – штучних нейронних мереж (ШНМ) і генетичних алгоритмів (ГА) для вирішення завдань синтезу інтелектуальних систем управління. Поєднання ГА і ШНМ дозволило синтезувати нейромережеву модель системи управління, яка не тільки уміє будувати внутрішні уявлення про зовнішній світ, прогнозувати поведінку об'єкту управління, але так само володіє здатністю генерувати оптимальне управління відповідно до вимог еталонної моделі.

У численних публікаціях, наприклад, [34, 35], показана ефективність використання нейромережевих структур для регулювання координат одномасових і двомасових електромеханічних систем з негативним в'язким тертям.

В результаті досліджень розроблена методика синтезу гібридної нейронної мережі, що реалізовує принцип дії фаззи регулятора, дозволила автоматично за допомогою комп'ютера визначити необхідні параметри регулятора [36, 37]. Застосування гібридних нейронних мереж дозволяє добитися зменшення значень пікових моментів, збільшити швидкодію приводу і підвищити точність управління, що у поєднанні з відомими перевагами мікропроцесорних засобів управління, такими як перешкодостійкість, малі габарити, низька вартість, здібність до виконання логічних операцій, можливість поєднання з вищим ієрархічним рівнем системи управління, простотою програмованого перенастроювання і ін. дозволяє розглядати гібридну нейронну мережу як перспективний засіб управління електромеханічними системами, у тому числі і нелінійними.

У Харківському національному університеті радіоелектроніки активно проводяться дослідження, присвячені управлінню нелінійними динамічними об'єктами із застосуванням штучних нейронних мереж (зокрема, [5, 38]). Значна увага приділяється вдосконаленню методів навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів за допомогою нейромереж, а також синтезу нейромережових систем управління такими об'єктами.

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

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

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

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

1.2 Аналіз структур систем нейрорегулювання. Вибір структури системи

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

У наступних нижче підрозділах будуть стисло розглянуті структури систем нейроуправління, актуальні на сьогоднішній день.

1.2.1 Послідовна схема нейроуправління

Для управління об'єктом нейронна мережа реалізує відображення, зворотне закону функціонування об'єкту. Тому дана структура також називається «інверсно-прямим управлінням». Метою навчання мережі є мінімізація деякого функціонала помилки. Для навчання мережі використовується Якоб'іан об'єкту. Навчання мережі припиняється, якщо помилка стає близькою до нуля, що дає підставу вважати, що ШНМ реалізує інверсну динаміку об'єкту управління. Даний метод називається спеціалізованим нейроуправлінням із зворотним відображенням (рис.1.1).

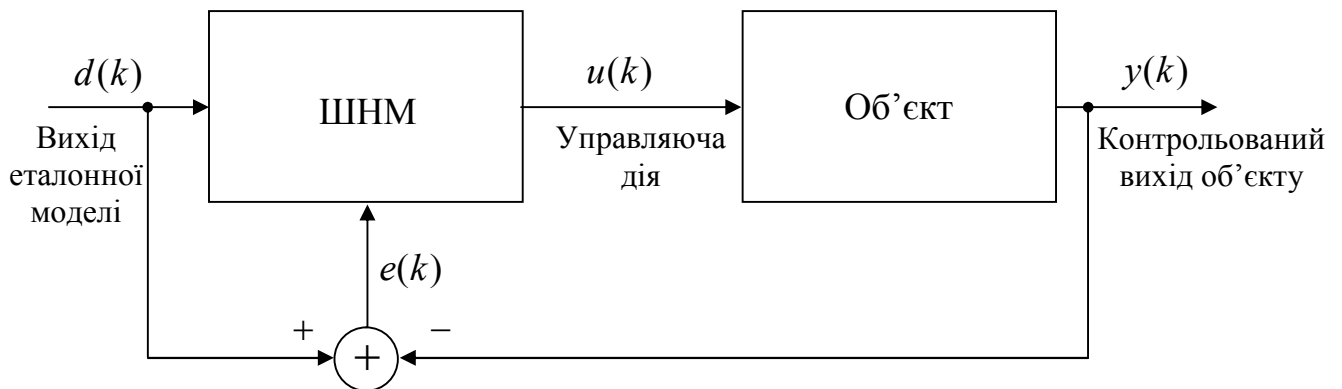


Рисунок 1.1 – Спеціалізоване нейроуправління

Розглянемо одновимірний об'єкт управління, із заданим співвідношенням вхід-вихід. Як нейрорегулятор виберемо двошаровий персептрон з одним нейроном у вихідному шарі і логістичними функціями активації у прихованому шарі

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

похідна яких обчислюється за виразом

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

Схема нейрорегулятора на основі двошарового персептрона показана на рис. 1.2.

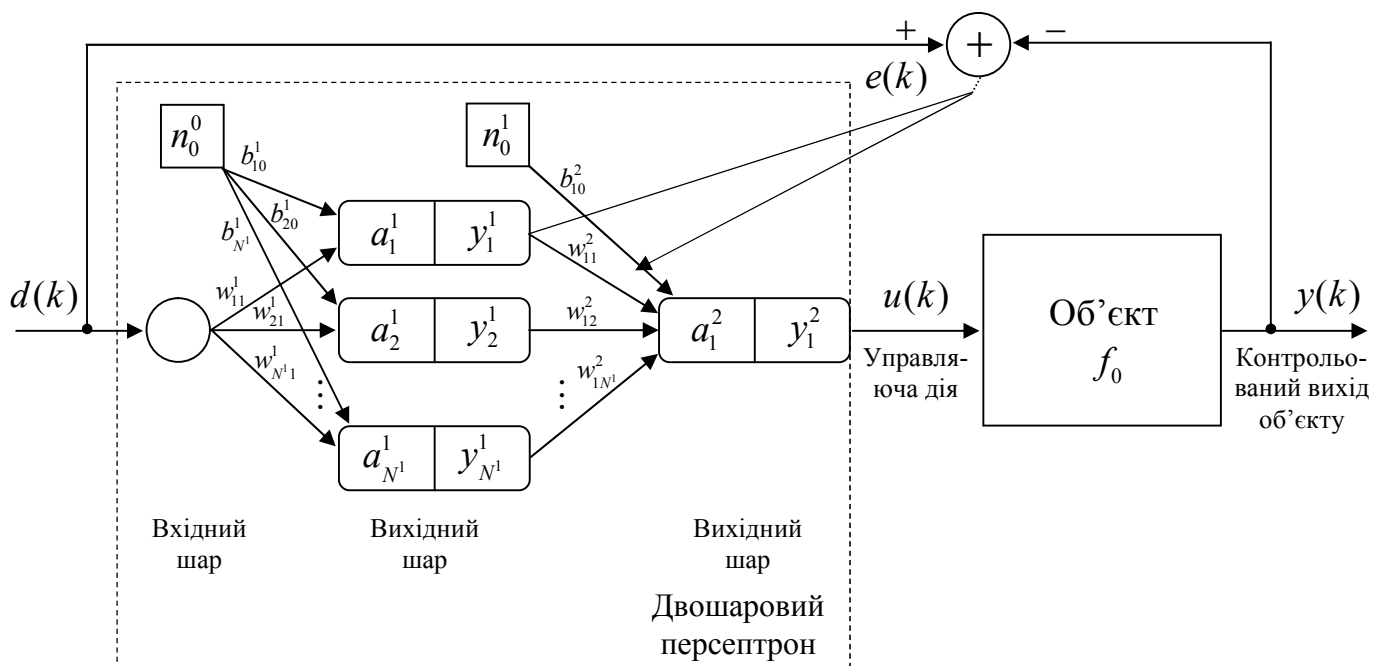


Рисунок 1.2 – Структура нейрорегулятора на основі двошарового персептрону

Мета управління полягає в мінімізації квадратичної функції помилки

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

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

Вихідний сигнал мережі може бути обчислений таким чином:

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

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

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

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

Для настройки вагових коефіцієнтів мережі скористаємося алгоритмом найшвидшого спуску. Застосовуючи правило обчислення похідних складних функцій, отримаємо наступну формулу для корекції вагів нейрона вихідного шару:

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

З (1.5) отримуємо

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

Приймаючи до уваги (1.3), (1.1), (1.2) і, ввівши позначення для локальної помилки вихідного шару $\delta^2 = \frac{\partial E}{\partial a_1^2}$, отримаємо

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

Тепер можемо записати загальну формулу настройки вагів вихідного шару

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

Аналогічно для нейронів прихованого шару, можемо записати

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

де

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

Таким чином, алгоритм настройки параметрів нейрорегулятора має вигляд:

1. Встановлюються початкові значення вагів w і зсувів b , задається норма навчання η .

2. Обчислюється помилка управління e і локальні помилки вихідного δ^2 і приховано δ_j^1 шарів (по формулах (1.8) і (1.11) відповідно), на підставі значень сигналу на виході еталонної моделі $d(k)$ розрахованих значень управляючої дії $u(k)$ і сигналу на виході об'єкту управління $y(k)$.
3. Коректуються ваги вихідного шару по формулі (1.9).
4. Коректуються ваги прихованого шару по формулі (1.10).
5. Повторюються дії пп.2-5 при $k = k + 1$.

Розглянута схема управління здатна реалізувати тільки статичне управління об'єктом. Для управління динамічними об'єктами необхідно на вхід регулятора подавати окрім поточних значень сигналів з виходу еталонної моделі і виходу об'єкту управління, значення цих сигналів, отримані на попередніх тактах. В цьому випадку як нейрорегулятор можна використовувати мережу типу багатошарового персептрона, а для її навчання використовувати розглянуту вище методику. Труднощі, що виникають при реалізації даного алгоритму, пов'язані з необхідністю обчислення Якобіана системи $f'_0(u(k))$ на другому кроці, оскільки зазвичай вираз $f_0(u(k))$ невідомий, то необхідно використовувати його апроксимацію. Для знаходження $f'_0(u(k))$ можна застосувати наближену формулу

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

де

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

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

1.2.2 Схема зворотного розповсюдження в часі

Інший підхід полягає в побудові НММ об'єкту (нейроемулятора), що є ідентифікатором системи. Функціонування системи управління складається з двох етапів: на першому етапі нейрорегулятор навчається інверсній динаміці

об'єкту, а надалі використовується для зворотного розповсюдження помилки на другому етапі функціонування системи – управлінні об'єктом. Використання нейроемулятора дозволяє обчислити Якобіан системи, замінюючи об'єкт його моделлю. В цьому випадку отримаємо:

$$f'_0(u(k)) \approx \frac{\partial \hat{f}_c(u(k))}{\partial u(k)}, \quad (1.16)$$

де $\hat{f}_c$ – відображення вхід-вихід, що реалізується НММ.

Добитися підвищення швидкості навчання нейроемулятора можна, якщо в якості і виходу об'єкту $\hat{y}$ використовувати суму сигналів деякої моделі $\hat{y}_1$ і сигналу з виходу ШНМ $\hat{y}_2$. При розузгодженні між прогнозованим і контрольованим виходом об'єкту виконується навчання мережі, направлене на мінімізацію помилки ідентифікації. В цьому випадку емулятор розглядається як сукупність деякої моделі і нейронної мережі (рис.1.3), при цьому настраюються тільки параметри нейронної мережі. Даний метод забезпечує хороші результати при рішенні практичних задач управління. Кращі результати при цьому дає використання моделі авторегресії (AR-модель), авторегресії з додатковими вхідними сигналами (ARX-модель), моделі авторегресії ковзаючого середнього з додатковими вхідними сигналами (ARMAX- модель), моделі вихідної помилки (ОЕ-модель), моделі Бокса-Дженкінса (BJ-модель) і інших, що забезпечують довгострокове прогнозування, при цьому нейронна мережа забезпечує реалізацію нелінійних залежностей.

Таким чином, загальна структура системи управління нелінійним динамічним об'єктом показана на рис.1.4. Тут ШНМ2 є ідентифікатором об'єкту (рис.1.5) і використовується для отримання оцінки його Якобіана в цілях навчання нейрорегулятора, що побудованого на ШНМ1 і реалізовує інверсно-динамічне відображення закону функціонування об'єкту.

Навчання нейрорегулятора виконується на підставі зворотного розповсюдження помилки через нейроемулятор. Дана схема управління також називається «Схема зворотного розповсюдження в часі» [39-43].

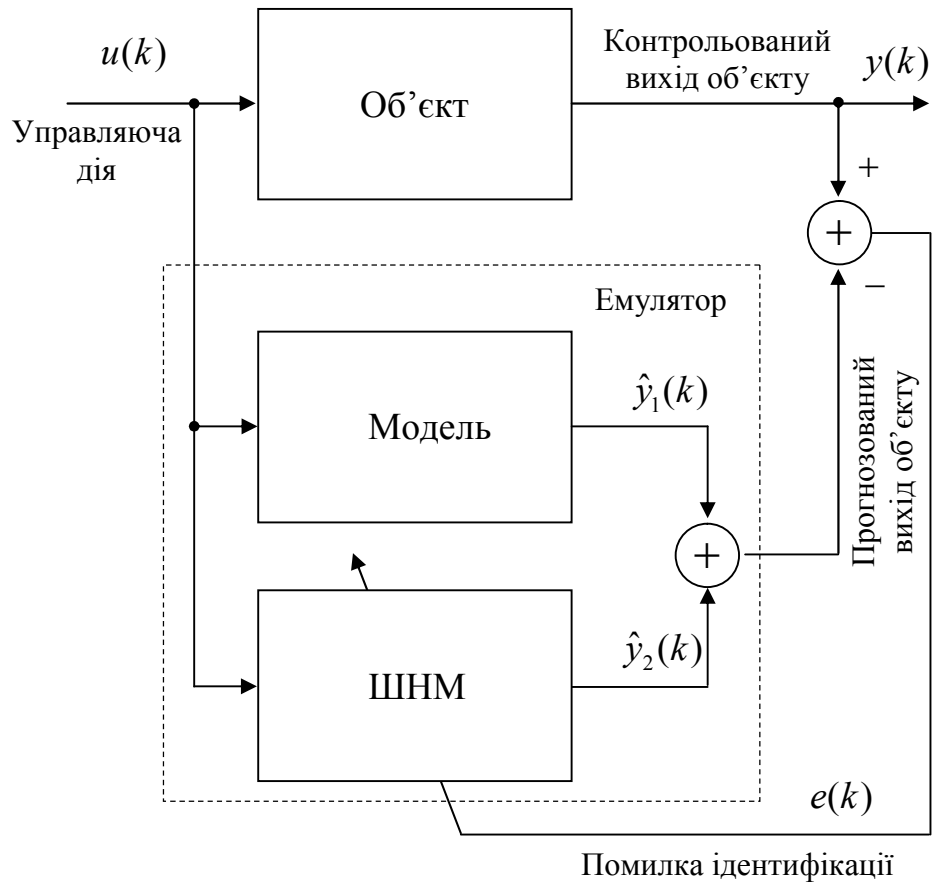


Рисунок 1.3 – Структура нейроемулятора з умовною моделлю

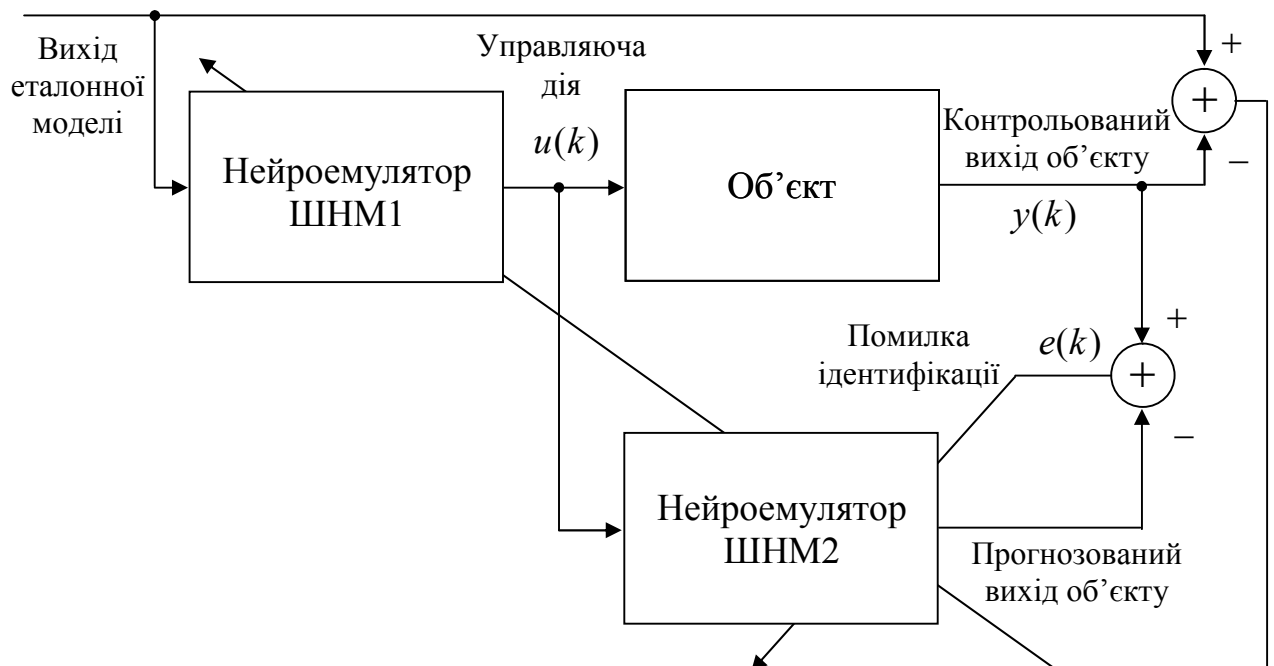


Рисунок 1.4 – Загальна схема інверсно-динамічного нейроуправління

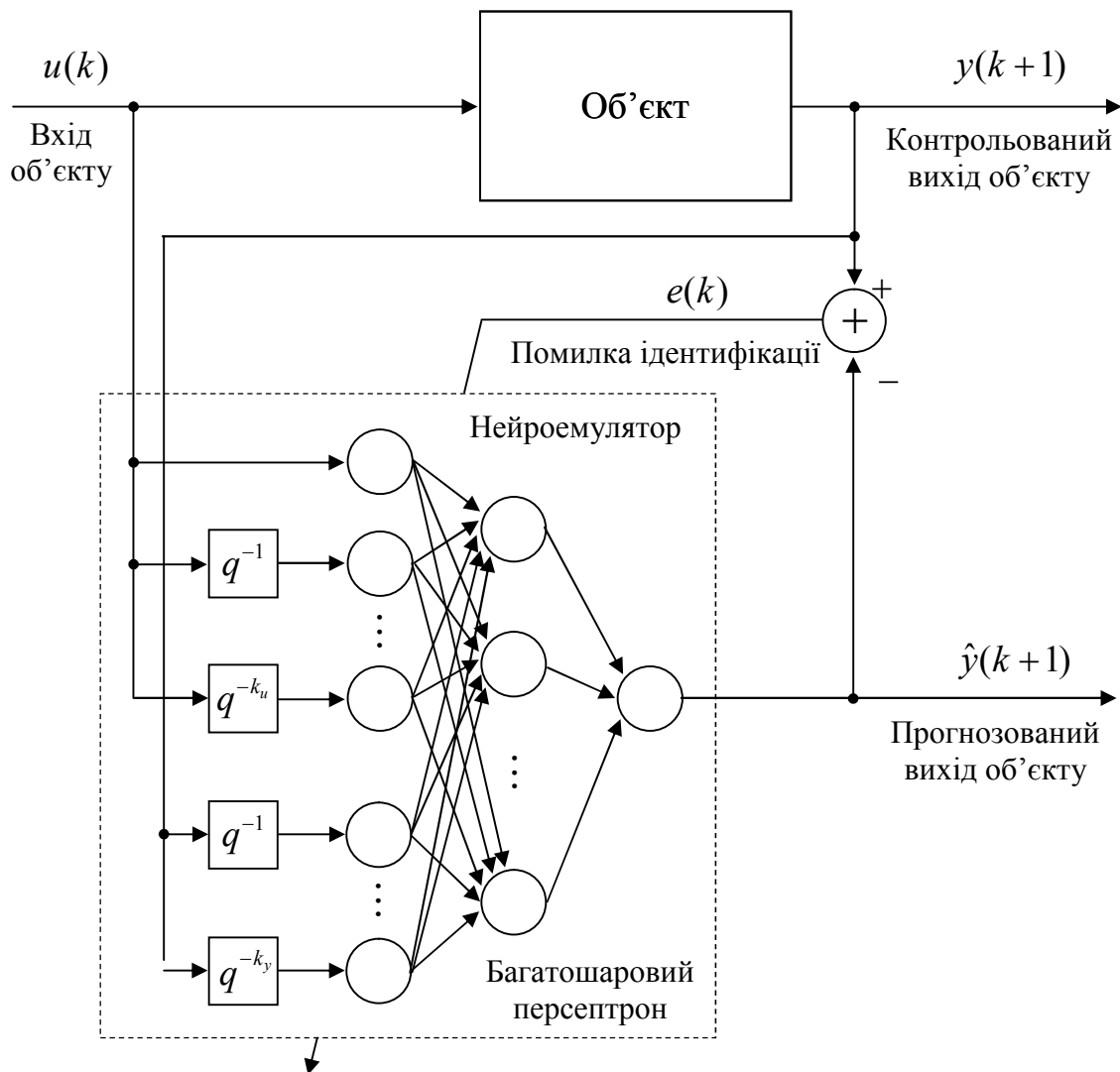


Рисунок 1.5 – Структура нейроемулятора

У роботі [44] була запропонована інша схема для реалізації інверсної динаміки об'єкту (рис.1.6). Тут нейроемулятор ШНМ2 навчається інверсній динаміці об'єкту управління, а нейрорегулятор ШНМ1 є копією мережі ШНМ2. Процедура копіювання моделі ШНМ2 відбувається після завершення її навчання. Навчання ШНМ2 інверсному відображенню виконується в автономному режимі або при подачі на вхід об'єкту управління випадкових вхідних сигналів. Як структура нейроемулятора можна використовувати модель рис.1.7.

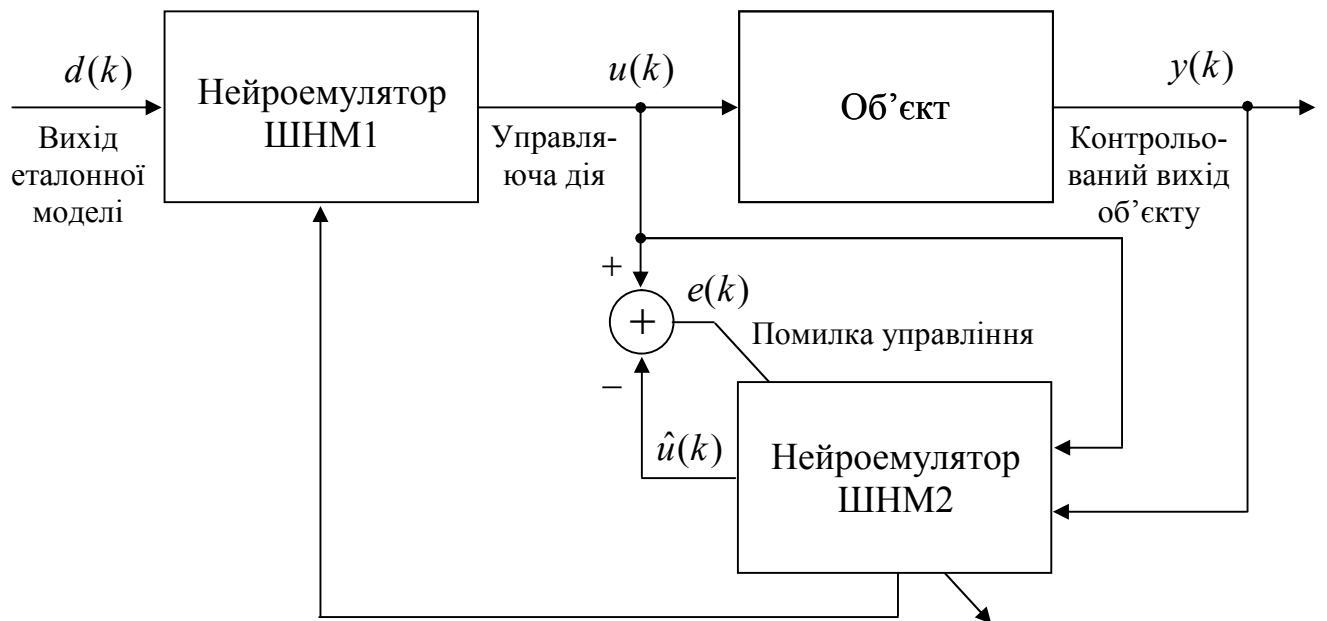


Рисунок 1.6 – Схема зворотного відображення

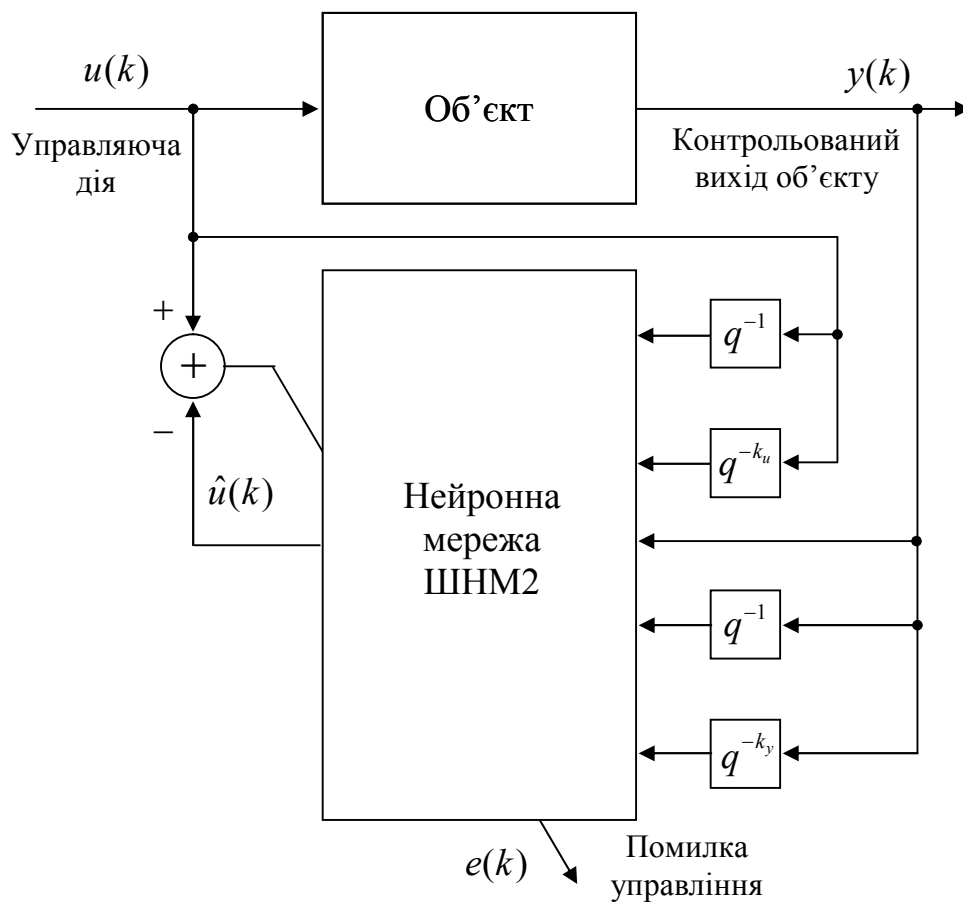


Рисунок 1.7 – Структура нейроемулятора для схеми зворотного відображення

1.2.3 Паралельне нейроуправління

Широкого поширення набули схеми паралельного нейроуправління. Управляюча дія в цьому випадку формується як сума сигналів, що формуються традиційним регулятором (наприклад, ПІД-регулятором) і нейрорегулятором (рис.1.8).

Завдання нейрорегулятора полягає в тому, щоб скоректувати вихідний сигнал звичайного регулятора, якщо той не забезпечує необхідної якості управління.

Один з варіантів паралельної схеми нейроуправління, заснований на інверсній і прямій моделі, був запропонований в роботах [45,46] (рис.1.9).

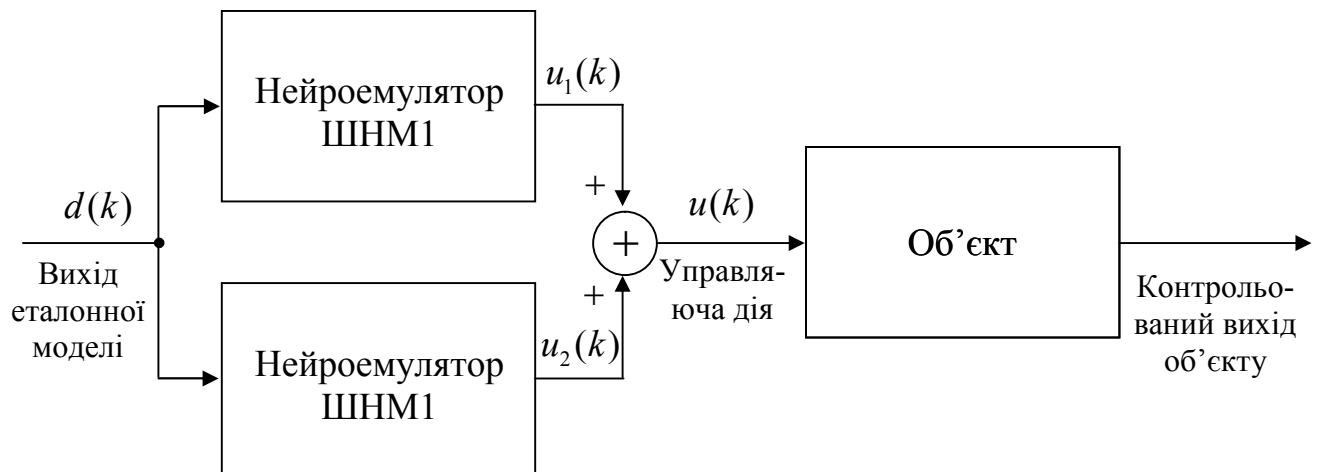


Рисунок 1.8 – Паралельна схема нейроуправління

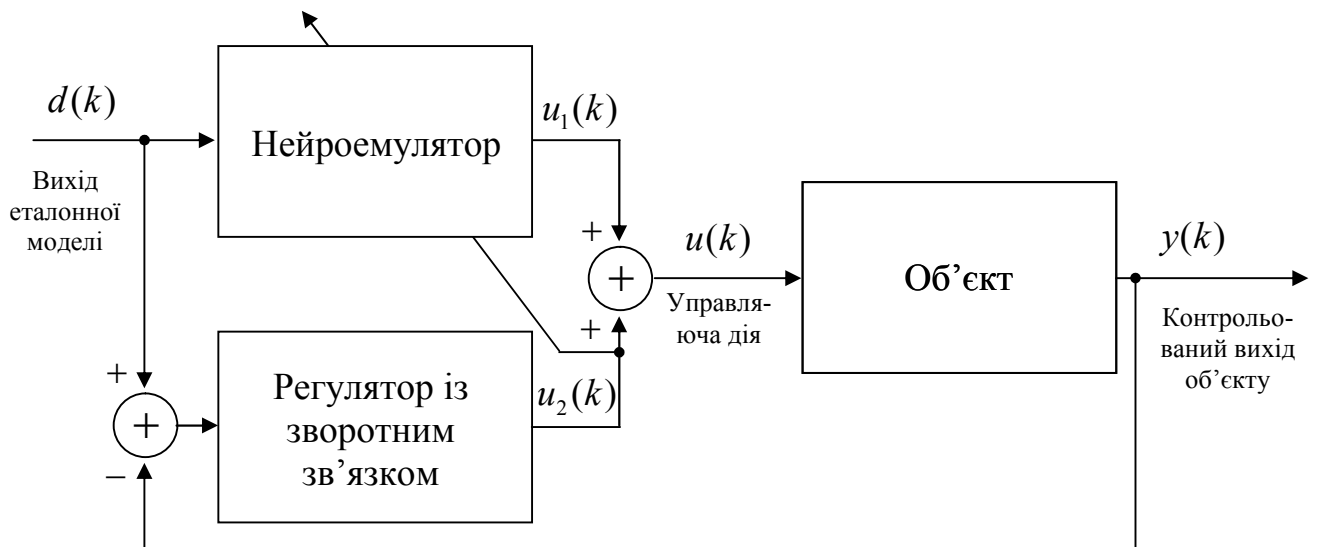


Рисунок 1.9 – Нейроуправління по помилці зворотного зв'язку

Для настройки параметрів нейрорегулятора використовується алгоритм навчання по помилці зворотного зв'язку. Навчання мережі здійснюється шляхом багатократного повторення циклів бажаної траєкторії, поки не буде досягнута необхідна точність управління, при цьому помилка зворотного зв'язку є сигналом з виходу регулятора, включеного паралельно нейромережі. Після закінчення стадії навчання нейрорегулятор приймає на себе управління об'єктом, виключаючи дію регулятора із зворотним зв'язком.

З метою узагальнення розглянемо модифікований варіант схеми нейроуправління по помилці зворотного зв'язку, представлений на рис.1.10 [47].

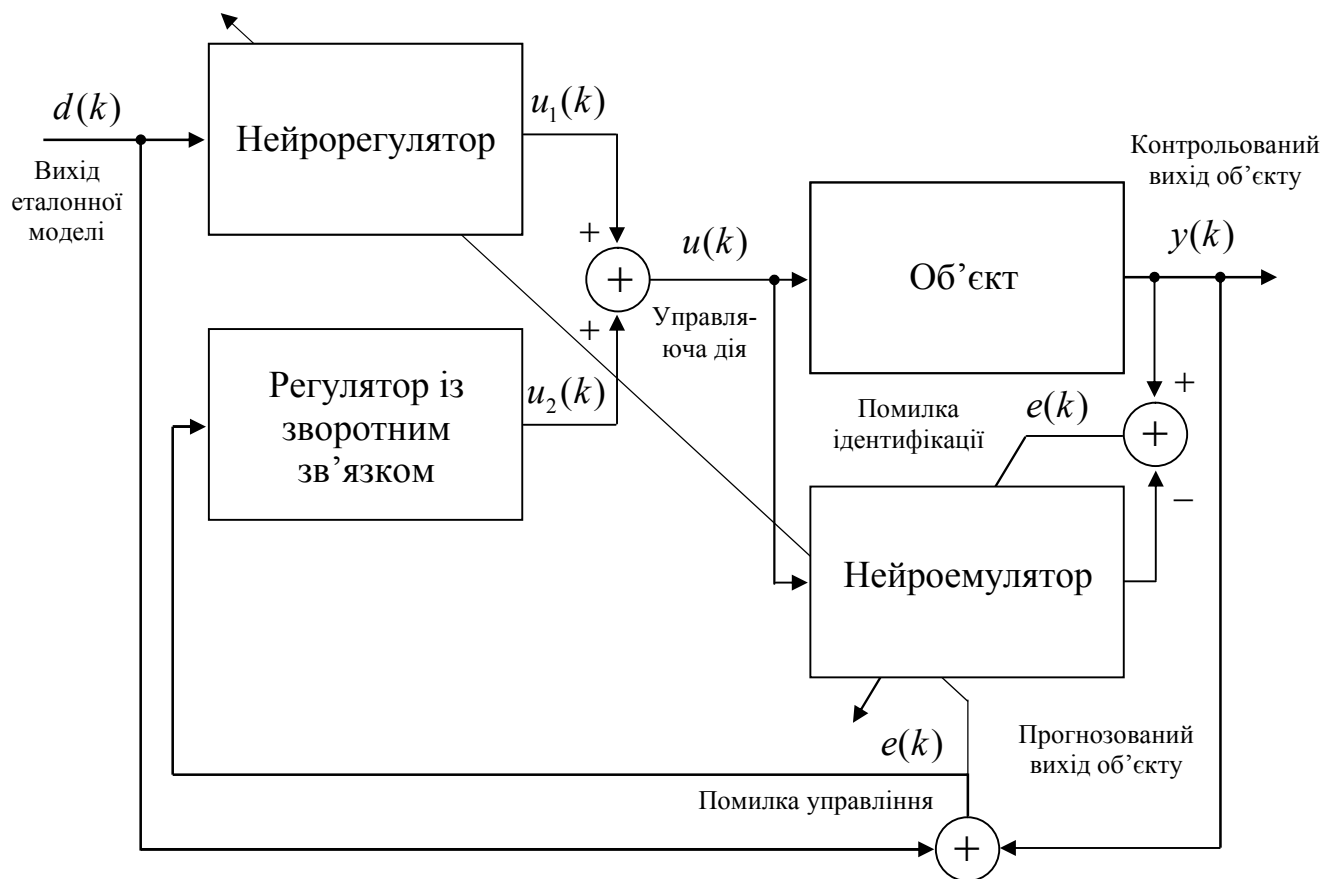


Рисунок 1.10 – Узагальнена схема нейроуправління по помилці зворотного зв'язку

Систему управління можна розділити на три частини. Перша частина показана жирними лініями і є традиційним контуром зворотного зв'язку, званим також зовнішнім зворотним зв'язком [48]. Управління по зворотному зв'язку засноване на помилці між виходом еталонної моделі і виходом об'єкту управління

$$e(k) = d(k) - y(k).$$

Як регулятор із зворотним зв'язком зазвичай використовується ПІД-регулятор. Друга частина включає нейроемулятор, що реалізовує динаміку об'єкту. Навчання нейроемулятора засноване на помилці ідентифікації $e(k) = y(k) - \hat{y}(k)$. Таким чином, нейроемулятор утворює внутрішній контур зворотного зв'язку, що забезпечує вищу швидкість реакції. Третя частина включає нейрорегулятор, що реалізовує інверсну динаміку об'єкту управління. Функціонування системи полягає в наступному: датчики зворотного зв'язку використовуються тільки на стадії навчання і оскільки вони характеризуються кінцевою величиною часової затримки, то накладаються обмеження на максимальний коефіцієнт передачі для регулятора із зворотним зв'язком. Це у свою чергу уповільнює швидкодію системи і обмежує швидкість навчання. На етапі навчання нейрорегулятор настраюється для отримання динамічної моделі об'єкту, а нейрорегулятор – інверсній динамічній моделі.

В ході навчання головну роль грає внутрішній контур зворотного зв'язку, і поступово нейрорегулятор стає ведучим. Після закінчення навчання дія регулятора із зворотним зв'язком повністю виключається, і управління здійснюється нейрорегулятором зворотного розповсюдження, що забезпечує високу швидкість управління навіть в умовах дії завад.

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

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

Як критерій якості використовуємо функцію помилки (1.3), а для настройки вагів нейрорегулятора застосуємо метод найшвидшого спуску. Виберемо структуру нейрорегулятора як показано на рис.1.11.

Таким чином, корекція вагів нейронів вихідного шару виконується у відповідно з наступним правилом:

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

а корекція вагів нейронів прихованого шару відповідно до правила:

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

Використовуючи позначення для локальної помилки вихідного шару $\delta^2 = \frac{\partial E}{\partial a_1^2}$ із співвідношення вхід-вихід для об'єкту управління (1.12), отримаємо

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

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

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

Тепер можемо записати $\Delta w_{1j}^2 = -\eta \delta^2 y_j^1$. Приймаючи до уваги, що функціонування j -го нейрона прихованого шару визначається співвідношеннями (1.5), (1.6) і, використовуючи позначення для локальної помилки прихованого шару δ_j^1 (1.13), можемо записати правило настройки вагів прихованого шару таким чином:

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

де

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

Алгоритм настройки вагів нейрорегулятора для паралельної схеми управління має вигляд:

1. Встановлюються початкові значення вагів w і зміщень b , задається норма навчання η .
2. На вхід мережі подається опорний сигнал $d(k)$ і обчислюються локальні помилки вихідного δ^2 і прихованого δ_j^1 шарів по формулах (1.13) і (1.14) відповідно.
3. Настроюються ваги вихідного шару: $\Delta w_{ij}^2(k+1) = \Delta w_{ij}^2(k) - \eta \delta^2 y_j^1$.
4. Настроюються ваги прихованого шару: $\Delta w_{j1}^1(k+1) = \Delta w_{j1}^1(k) - \eta^2 \delta_j^1 d$.
5. Повторюються дії пп.2-5 при $k = k + 1$.

Як і у разі послідовної схеми нейроуправління для навчання нейрорегулятора на кожній ітерації потрібне обчислення Якобіана об'єкту управління. Вирішити задачу можна шляхом його апроксимації або використовувати оцінку, що отримується за допомогою нейромодельованого об'єкту на етапі ідентифікації, як це показано на рис.1.11.

1.2.4 Пряме адаптивне нейроуправління

Розглянемо нелінійний об'єкт, описуваний NARX моделлю

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

де $y(k)$ – вихід об'єкту управління;

$u(k)$ – управляюча дія;

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

k_u, k_y – порядки запізнювання по каналу управління і вихідному каналу відповідно.

Для побудови нейромережевої моделі об'єкту використовуваний двошаровий персептрон, ваги якого настроєні відповідно до алгоритму зворотного розповсюдження помилки. Як функції активації нейронів прихованого шару вибирається функція гіперболічного тангенса з функцією активації

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

похідна якої обчислюється за виразом

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

Для нейрона вихідного шару вибирається лінійна функція активації. Введемо позначення для вектора вхідних сигналів нейромодулятора:

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

Нейромережева модель об'єкту (НММ) має вид:

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

де $\hat{y}(k+1)$ – вихід ШНМ

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

Оскільки алгоритм навчання нейрмережі забезпечує мінімізацію квадрата помилки ідентифікації $e^2(k+1) = [y(k+1) - \hat{y}(k+1)]^2$, то вихід нейромережевої моделі об'єкту $\hat{y}(k+1)$ може використовуватися як прогнозований вихід об'єкту управління (1.15). Внаслідок чого метою управління може служити мінімізація розузгодження між вихідними сигналами нейромережевої і еталонної моделей. Таким чином, критерій якості управління має вид:

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

де

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

Тепер управляюча дія $u(k)$ може бути вибрана так, щоб доставити мінімум функціоналу J .

Використовуючи нейромережеву структуру, рівняння (1.16) можемо записати таким чином

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

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

w^2 – вектор вагових коефіцієнтів вихідного шару;

b^1 – вектор вагових коефіцієнтів зсувів прихованого шару;

b^2 – зсув вихідного шару.

Мінімум J може бути знайдений рекурсивно за допомогою методу найшвидшого спуску

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

де $\eta > 0$ – норма навчання.

Легко бачити, що функціонування регулятора залежить від точності НММ, тому необхідно, щоб вихід ШНМ $\hat{y}(k+1)$ асимптотично прагнув до $y(k+1)$. Для цього можна використовувати навчання мережі в режимі on-line.

Диференціюючи (1.17) по $u(k)$ отримаємо

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

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

Підставляючи (1.21) в (1.20), отримаємо

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

На підставі відомої архітектури мережі (1.19) можна аналітично обчислити градієнт таким чином:

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

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

Тепер остаточно можна записати:

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

Таким чином, алгоритм прямого адаптивного управління має вигляд:

1. Обчислюється $\hat{y}(k+1)$ відповідно до (1.19).
2. Обчислюється $e(k+1)$ відповідно до (1.18).
3. Виконується настройка вагів мережі відповідно до алгоритму зворотного розповсюдження помилки.
4. Обчислюється управляюча дія $u(k+1)$ відповідно до (1.22).
5. Подається сигнал $u(k+1)$ на вхід об'єкту управління.
6. Повторюються дії пп. 1-5.

Поліпшення властивостей отриманого алгоритму може бути досягнуте з використанням методів теорії управління по прогнозуванню, шляхом урахування не тільки поточних значень вхідних і вихідних сигналів об'єкту, але і їх передбачених значень. Таким чином, для проектування закону управління знадобляться майбутні значення сигналів на виході еталонної моделі і об'єкту. Оскільки нейромережева модель (1.28) в асимптотиці еквівалентна об'єкту, то вона може використовуватися для прогнозу його вихідних сигналів.

1.2.5 Управління з самонастройкою

Одним з варіантів застосування нейронних мереж в завданнях управління є використання схем нейроуправління з самонастройкою. В цьому випадку ШНМ виконує ті функції, що і оператор системи, тобто здійснює настройку параметрів звичайного регулятора. Як відомо, багато методів адаптивного управ-

ління включають ряд параметрів, які необхідно вибирати або налаштувати заздалегідь. Зазвичай початкові значення таких параметрів налаштовуються методом проб і помилок. Використання ШНМ дозволить автоматизувати процес вибору і настройки параметрів традиційних регуляторів. Таким чином, стратегія нейроуправління з само настройкою може застосовуватися в багатьох традиційних технологіях управління.

Розглянемо схему нейроуправління з самонастройкою для ПІД-регулятора, показану на рис.1.11.

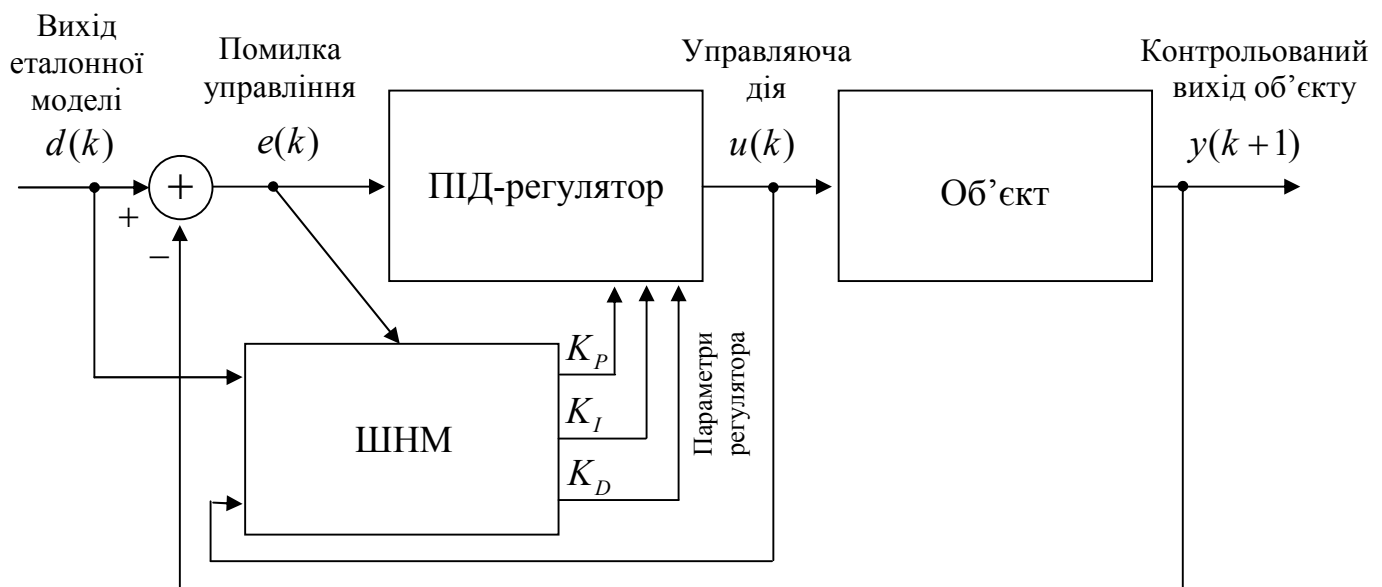


Рисунок 1.11 – Структура навчання ПІД-регулятора з само настройкою

ШНМ використовується для настройки параметрів ПІД-регулятора для того, щоб забезпечити мінімізацію помилки управління. Закон функціонування ПІД-регулятора в дискретному часі описується різницеvim рівнянням

$$u(k) = u(k-1) + K_p(e(k) - e(k-1)) + K_i e(k-1) + K_d(e(k) - 2e(k-1) + e(k-2)), \quad (1.23)$$

де K_p , K_i , K_d – пропорційний, інтегральний і диференціальний коефіцієнти передачі відповідно;

$e(k) = d(k) - y(k)$ – помилка управління;

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

Розглянемо об'єкт управління, закон функціонування якого задається рівнянням $y(k+1) = f_0(u(k))$. Як критерій якості використовується квадратичну функцію

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

Для настройки параметрів ПД-регулятора побудуємо двошаровий персеptron з трьома нейронами у вихідному шарі. Алгоритм навчання нейронної мережі побудуємо так, щоб на першому виході мережа формувала значення пропорційного коефіцієнта передачі K_p , на другому виході – інтегрального K_I , а на третьому – диференціального K_D , тобто

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

Для реалізації нелінійних відображень використовуємо логістичні функції актинії (1.1) для нейронів прихованого шару, а для нейронів вихідного шару використовуємо лінійні функції для забезпечення необхідного діапазону значень коефіцієнтів ПД-регулятора. Для настройки вагових коефіцієнтів ШНМ скористаємося алгоритмом найшвидшого спуску.

При використанні квадратичної функції помилки вираз для градієнта критерію якості функціонування для нейронів вихідного шару має вигляд

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

Зважаючи на те, що $e(k+1) = d(k+1) - y(k+1)$ і враховуючи (1.24) можемо записати

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

Оскільки правило функціонування нейронів вихідного шару має вигляд:

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

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

На підставі закону функціонування ПД-регулятора (1.23) і, враховуючи (1.25), записуємо рівняння для приватних похідних $\frac{\partial u(k)}{\partial y_i^2(k)}$

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

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

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

Тепер, ввівши позначення для локальної помилки вихідного шару

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

можемо записати правило настройки вагів нейронів вихідного шару

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

Перейшовши до прихованого шару, отримаємо

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

де δ_i^1 – локальна помилка прихованого шару

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

Таким чином, правило настройки вагів нейронів прихованого шару має вигляд

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

Для реалізації отриманого алгоритму необхідно на кожному такті обчислювати значення Якобіана об'єкту управління $\frac{\partial y(k+1)}{\partial u(k)}$, для чого можна використовувати нейроемулятор.

Загальний алгоритм настройки вагів ПД-нейрорегулятора з самонастройкою має вигляд.

1. Встановлюються початкові значення вагів w і зсувів b , задається норма навчання η . Встановлюється $k = 0$.
2. На вхід мережі подається опорний сигнал $d(k)$ і обчислюється помилка управління $e(k+1)$ і локальні помилки вихідного δ_i^2 (по формулі (1.26)) і прихованого δ_i^1 шарів по формулі (1.28)).
3. Коректуються ваги вихідного шару по формулі (1.27).
4. Коректуються ваги прихованого шару по формулі (1.30).
5. Повторюються дії пп.2-6 при $k = k + 1$.

1.2.6 Нелінійне предиктивное управління

Принцип предиктивного методу регулювання на основі моделі (*model predictive control*, MPC) (рис.1.12) полягає у формуванні такої послідовності сигналів дії на об'єкт, яка мінімізує різницю між завданням і що передбачається

моделлю процесу вихідним сигналом в майбутньому (на «горизонті»), (див. пунктирні лінії на рис. 1.12).

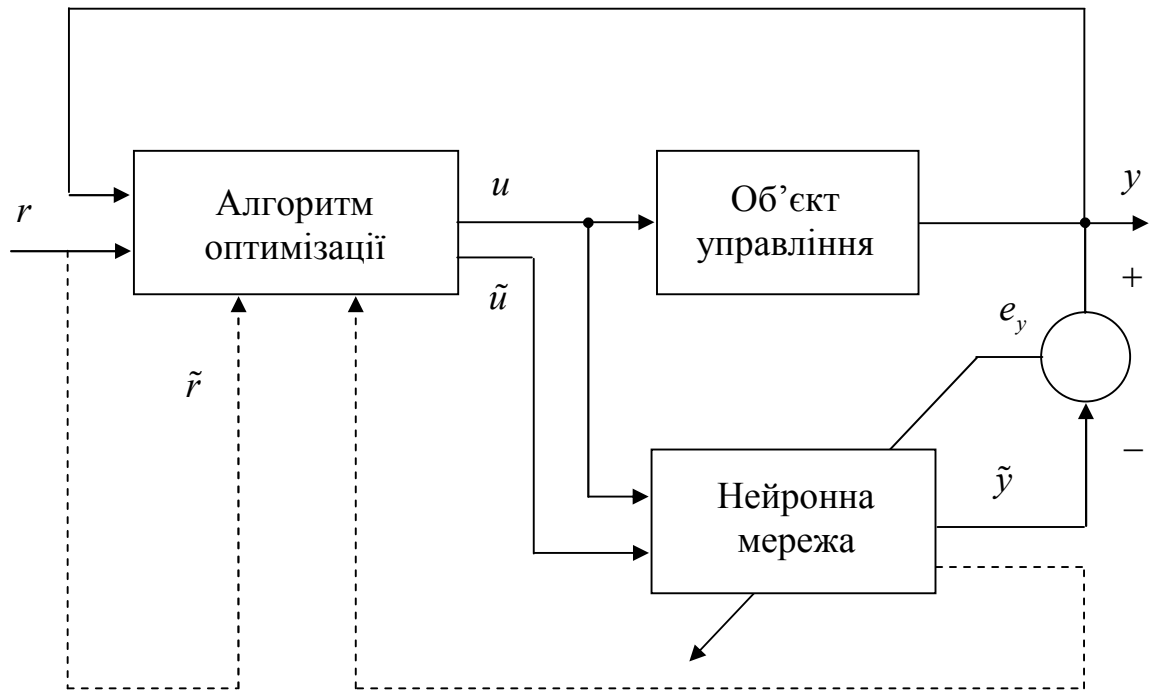


Рисунок 1.12 – Предиктивне регулювання

У цьому випадку регулятор являє собою ітераційний алгоритм оптимізації [49], який реалізується наступним чином:

- мінімізація проводиться за допомогою чисельних методів оптимізації;
- мінімізується функціонал якості, що відображає заданий критерій ефективності;
- для впливу на об'єкт використовується лише перший елемент оптимізованої послідовності;
- процедура оптимізації повторюється на кожному дискретному кроці.

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

У магістерській роботі використовується нелінійне предиктивне регулювання.

1.3 Загальні відомості про роторні екскаватори

Основною машиною гірничодобувних комплексів є роторний екскаватор - складний агрегат, який включає понад десять взаємопов'язаних силових механізмів і містить як робоче, так і допоміжне обладнання, що забезпечує транспортування видобутої породи до транспортних засобів комплексу. На рис.1.11 приведена компоновальна схема роторного екскаватора.

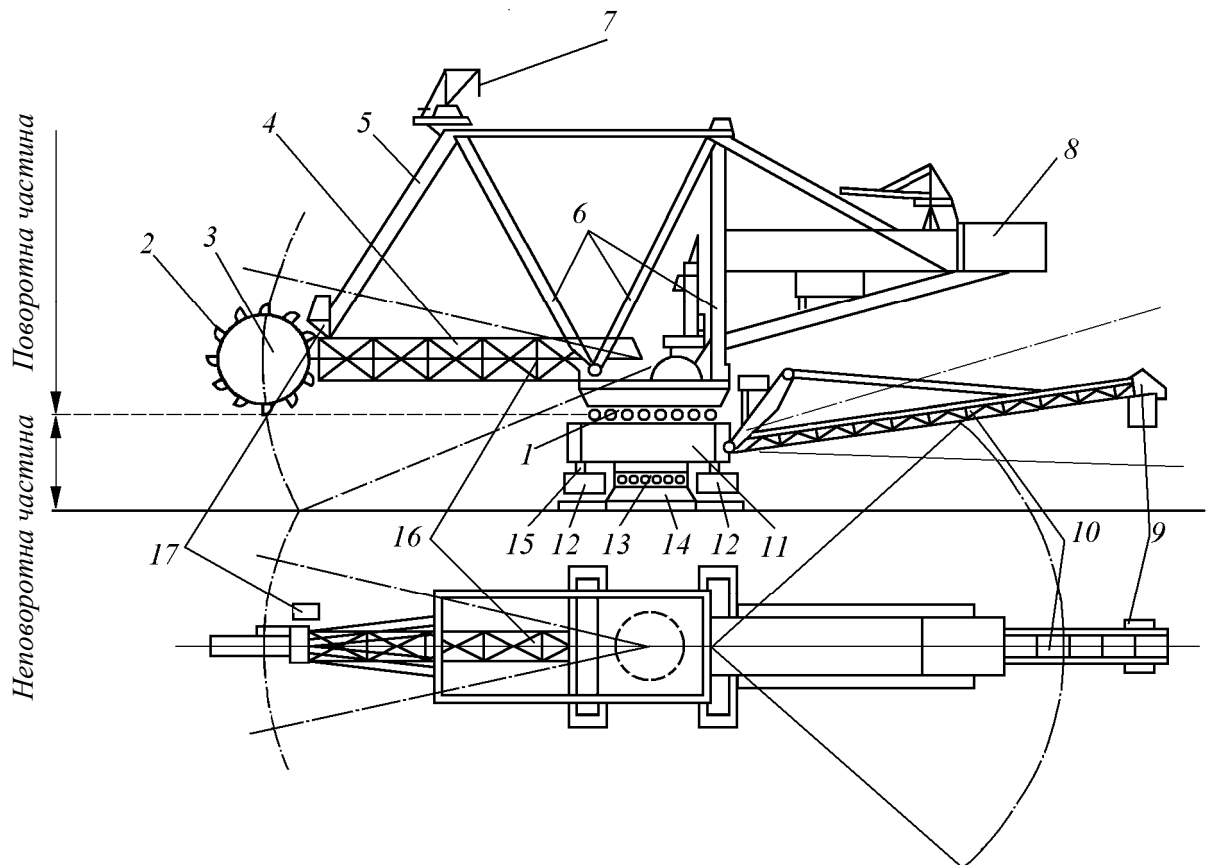


Рисунок 1.11 – Компоновальна схема роторного екскаватора

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

відділення природи, що поступає на розташований усередині фермової конструкції стріли конвейєр 16.

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

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

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

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

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

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

Основними механізмами, що забезпечують робочий процес екскаватора, є наступні: механізм приводу ротора; механізм повороту стріли екскаватора; механізм розвороту консолі; механізм розвороту екскаватора; механізм підйому; механізм ходу; механізм конвейєра стріли; механізм конвейєра розвантажувальної консолі.

Компоновка індивідуальних електроприводів роторного екскаватора показана на рис. 1.12.

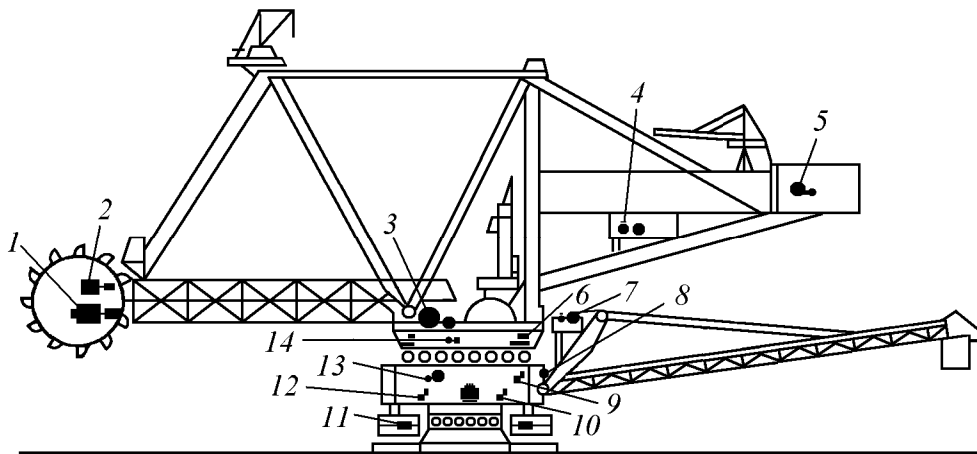


Рисунок 1.12 - Компоновка індивідуальних електроприводів роторного екскаватора

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

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

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

Переміщення розвантажувальної консолі у вертикальній площині забезпечується електроприводами підйому 7, встановленими на спеціальному майданчику, нижче за горизонтальну площину коштує привід повороту консолі 9, аналогічний приводу повороту верхньої будови, але меншій потужності, і привід конвеєра розвантажувальної консолі 8.

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

1.4 Аналіз факторів, які обумовлюють вибір системи управління електроприводом механізму повороту стріли роторного екскаватора

Ефективне використання роторних екскаваторів можливе лише при безперебійному функціональному його механізмів. Відмова в будь-якому з основних електроприводів зв'язана з порушенням технології або повним припиненням розробки ґрунту і веде до простою екскаватора, збитки від якого тим більше, чим більше продуктивність комплексу, тому до електроприводів основних механізмів роторного екскаватора пред'являються загальні вимоги високої надійності роботи, стабільності характеристик і робочих параметрів.

З одного боку, виконання вказівок вимог можливо про виборі систем електроприводу, схем управління, електричних машин, апаратів і іншої елементної бази, максимально пристосованих до важких умов експлуатації електроустаткування роторних екскаваторів. З іншого боку, для скорочення простоїв екскаватор з вини електроустаткування схеми управління електропроводами не повинні вимагати переналагоджень, повинні забезпечувати можливість швид-

кого відновлення працездатності окрім загальних вимог до кожного з електроприводів основних механізмів пред'являються специфічні вимоги, які витікають з проведеного вище аналізу функцій механізмів, що виконуються в технологічному процесі, конструктивних особливостей, режимів роботи і характеру навантажень.

Електропривод повороту стріли ротора виконує головний робочий рух, що забезпечує різання ґрунту, – подовжню подачу – і, так само як і електропривод роторного колеса, повинен працювати в тривалому режимі. Електроприводи повороту розвантажувальної консолі і розвороту екскаватора призначені для короткочасних режимів роботи.

Для відробітку забою при кожній заходці необхідна зміна напрямку повороту стріли ротора. Розворот екскаватора, що вимагає для правильної орієнтації машини по відношенню до забою, і поворот розвантажувальної консолі, що забезпечує гнучкість технологічного зв'язку між екскаватором і іншими механізмами комплексу, також вимагають оперативної зміни напрямку обертання електроприводу. Отже, електроприводи опорно-поворотних пристроїв повинні бути реверсивними.

Однією з найважливіших особливостей електромеханічних систем електроприводів повороту стріли ротора і розвороту екскаватора є наявність пружно пов'язаних з платформою інерційних мас стріли ротора з роторним колесом і консолі противаги. Перехідні процеси в таких системах супроводжуються коливаннями, тому від вказаних електроприводів потрібна підвищена плавність розгону і гальмування з обмеженими на допустимому рівні ривком і прискоренням.

Для виконання цієї вимоги необхідне формування в перехідних процесах оптимальної залежності швидкості від часу $\omega = f(t)$, яка забезпечує мінімізацію динамічних навантажень. Останнє особливо важливе для електроприводу повороту стріли ротора, оскільки його допустимий динамічний момент трохи перевищує статичний момент в режимі різання.

Необхідний діапазон регулювання швидкості електроприводів опорно-поворотних механізмів складає не менше (3- 4) : 1. Він обумовлений для елект-

роприводів повороту розвантажувальної консолі і розвороту екскаватора, з одного боку, з іншого боку, необхідністю отримання великих швидкостей для скорочення часу на виконання вказаних операцій при значних кутах поворотних переміщень.

Для електроприводу повороту стріли ротора вказана вимога до діапазону регулювання доповнюється ще і необхідністю безступінчатого регулювання швидкості. Обидві вимоги обумовлено як необхідністю регулювання товщини стружки, що зрізається, при змінах фізико-механічних властивостей ґрунту, що розробляється, і відхиленням від норми середнього завантаження приводу роторного колеса, так і конструктивними особливостями екскаватора, що вимагають для того, що потримало постійності продуктивності екскаватора регулювати швидкість повороту при заходке по заданому закону.

Навантаження опорно-поворотних механізмів змінюються в досить широкі межі. Це обумовлено можливим ухилом бази екскаватора і вітрового навантаження, а для електроприводу повороту стріли ротора ще і випадковими змінами бічної сили різання, що становить, що виникають із-за неоднорідності ґрунту, що розробляється. Разом з тим потрібна підтримка постійності заданих робочих швидкостей опорно-поворотних механізмів, особливо для електроприводу повороту стріли ротора, оскільки допустиме відхилення параметрів стружок, що зрізаються, від середнього значення складає $\pm 15\%$. Отже, жорсткість робочої ділянки механічних характеристик електроприводів вказаних механізмів повинна бути достатньо високою.

Разом з тим опорно-поворотні механізми є багатомасовими пружними системами, коливання в яких виникають при змінах як управляючих, так і збудюючих дій. Особливо сильно це виявляється в електромеханічній системі приводу повороту стріли ротора унаслідок розташування робочого органу на кінці протяжної і пружної стріли. Із-за зміни сили різання швидкість подовжньої подачі може випробовувати значні коливання навіть при абсолютному жорсткій механічній характеристиці електроприводу. Причому електропривод ефективно не сприятиме загасанню виниклих коливань, оскільки підвищення жорсткості

механічної характеристики і стабілізація швидкості електроприводу вимикають його з цього процесу.

Потрібний вибір раціональної динамічної жорсткості робочої ділянки механічних характеристик електроприводів опорно-поворотних механізмів, що забезпечує як підтримку технологічних параметрів в заданих межах, так і ефективне використання демпфуючих властивостей електроприводу. Крім того, механічна характеристика електроприводу повороту стріли ротора у зв'язку з достатньо високою вірогідністю технологічних перевантажень і стопорення повинна бути екскаваторної форми і повинна обмежувати момент, що розвивається, допустимим по умові міцності механізму і конструкції екскаватора значенням.

Особливістю електроприводу повороту стріли ротора є взаємозв'язок його навантажень з навантаженнями електроприводу роторного колеса. Постійна складова сили різання прямо пропорційна швидкості подовжньої подачі. Цим визначається доцільність введення в схему управління електроприводом повороту стріли ротора пристрою, що автоматично знижує його швидкість аж до зупинки при технологічних перевантаженнях електроприводу роторного колеса.

Складність вимог, що пред'являються до регульованих електроприводів основних механізмів роторних екскаваторів, зумовила для більшості з них застосування системи електроприводу постійного струму керований перетворювач-двигун. Як керований перетворювач використовується або електромашинний перетворювач (система Г-Д), або реверсивний тиристорний перетворювач (система ТП-Д). Для електроприводу механізму повороту стріли використовуємо систему Г-Д.

1.5 Постановка задачі дослідження

Проведений аналіз літератури показав, що існує велика кількість підходів до проблеми синтезу систем управління, при цьому універсального і ідеального регулятора поки не існує. При синтезі систем управління механізмом повороту стріли роторного екскаватора, крім вже згаданих вимог, необхідно враховувати той факт, що опорно-поворотні механізми роторного екскаватора є багатомасовими пружними системами, коливання в яких виникають при змінах як управляючих, так і збурюючих дій. Традиційні регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

Використання нейромережевих технологій управління значно спрощує вирішення математичних проблем, пов'язаних із аналітичним синтезом і аналізом властивостей проєктованих систем. Це зумовлено тим, що якість процесів управління та досягнуті характеристики в нейромережевих системах переважно визначаються фундаментальними властивостями багат шарових нелінійних нейронних мереж, а не аналітично виведеними оптимальними законами, які зазвичай реалізуються у вигляді програмного забезпечення. Налаштовувані багат шарові нейронні мережі мають низку переваг, які роблять їх ефективними для вирішення завдань управління динамічними об'єктами.

Універсальні властивості апроксимацій багат шарових мереж, грають ключову роль у формуванні нелінійних алгоритмів управління. На основі узагальненої теореми апроксимації Стоуна-Вейерштрасса зроблений висновок про те, що за допомогою нелінійних нейронних мереж можна скільки завгодно точно рівномірно наблизити будь-яку безперервну функцію багатьох змінних на будь-якій замкнутій обмеженій множині. Висновок послужив теоретичним обґрунтуванням можливого застосування багат шарових нейронних мереж для формування оптимальних функцій, що управляли, в динамічних системах. Через свою структуру і алгоритми навчання багат шарові мережі дозволяють вирішувати такі нелінійні задачі теорії управління, які традиційними аналітичними методами не завжди можна вирішити в повному об'ємі. Можливість форму-

вати дії практично будь-якої форми, що управляють, робить штучні нейронні мережі засобом, адекватним рівню складності нелінійного завдання. Тому нейромережеві системи управління відносять до так званих «інтелектуальних» систем, тобто до систем, за визначенням, з неповною інформацією, з складною або немодельованою динамікою і неконтрольованою зміною власних властивостей.

Адаптивність нейромережевих структур унаслідок їх навчання в процесі функціонування дозволяє коректувати в реальному часі функцію управління при неконтрольованих змінах динамічних і статичних характеристик об'єкту, використовуючи для цього поточну вимірювальну інформацію в системі.

Здібність нейронних мереж до паралельної обробки сигналів робить природним їх застосування для управління багатовимірними (або багатоканальними) об'єктами. Багатошарові нейронні мережі є однорідним обчислювальним середовищем. Це універсальні паралельні обчислювальні структури, призначені для вирішення найрізноманітніших класів завдань, зокрема для побудови регуляторів в системах управління нелінійними динамічними об'єктами.

Таким чином, аналіз останніх досягнень і публікацій по синтезу сучасних регуляторів систем управління з використанням нейронних мереж показує, що даний напрямок є перспективним, а робота по застосуванню нейронних мереж для управління системами із складними кінематичними зв'язками є актуальною.

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

Для досягнення поставленої мети необхідно вирішити наступні завдання.

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

2. Розробити математичну моделі системи управління з урахуванням податливості механічних зв'язків, провести моделювання багатомасової системи на ЕОМ і виконати аналіз отриманих результатів.

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

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Аналіз літературних джерел за темою роботи підтвердив доцільність використання нейронних мереж у завданнях управління. Сучасні системи управління складними динамічними об'єктами повинні бути здатними ефективно адаптуватися до змін умов роботи шляхом оперативного коригування параметрів і структури застосовуваних законів управління. Ці вимоги можна задовольнити за допомогою апарату теорії адаптивного управління. Одним із дієвих підходів до реалізації адаптивного управління є застосування методів і засобів нейромережевого моделювання та управління. Сучасні нейромережеві технології дозволяють вирішувати завдання ідентифікації та управління як на етапі проектування системи (з фіксацією отриманих алгоритмів управління), так і безпосередньо під час її функціонування.

2. Показано, що для завдань управління доцільно застосування багатошарові мережі персептронного типу, включаючи їх варіанти з зворотними зв'язками і з лініями затримки на входах за вхідними і вихідними сигналами. Застосування динамічних варіантів методів навчання мереж розглянутого класу дає можливість створювати адаптивні системи управління, які дозволяють забезпечити ефективну експлуатацію складних систем в умовах різноманітних невизначеностей.

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

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

5. Проведений аналіз існуючих методів опису нелінійних динамічних об'єктів і розглянуті принципи побудови нейромережевих NARMAX і NARX

моделей. NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при нейронній ідентифікації систем.

6. На підставі аналізу технічних вимоги, що пред'являється до електроприводів опорно-поворотних механізмів роторних екскаваторів для електроприводу механізму повороту стріли вибраний електропривод постійного струму, виконаний за системою Г–Д.

7. Показана доцільність використання нейромережевих технологій для синтезу нейромережевої системи управління механізмом повороту стріли роторного екскаватора, що зумовлено наявністю пружно пов'язаних з платформою інерційних мас стріли ротора з роторним колесом і консолі противаги. Це призводить до того, що перехідні процеси мають значні коливання. Традиційні системи управління не можуть забезпечити необхідну якість функціонування систем управління основними механізмами роторного екскаватора.

8. Проаналізовані нейромережеві підходи до рішення задачі управління. Сформульовано задачу дослідження, що складається в розробці 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. Вимоги до системи управління

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

Таблиця 2.1 – Технічні дані роторного екскаватора

	Параметри	Одиниці вимірю- вання	Позна- чення	Величина
1	Теоретична продуктивність по рих- лій масі	$m^3 \cdot год$	Q	1700
2	Діаметр роторного колеса	m	D_p	8
3	Бічне зусилля різання	H	F_6	$1,2 \cdot 10^6$
4	Питоме зусилля копання	$кг/м^2$	k_f	$3,8 \cdot 10^4$
5	Радіус різання	m	R_0	23,3
6	Кутова швидкість повороту платфо- рми	c^{-1}	$\omega_{пл}$	$2,9 \cdot 10^{-2}$
7	Момент площ поворотної платфор- ми, що піддається дії вітру щодо осі обертання	$м^2 \cdot м$	$M_{пв}$	430
8	Швидкість руху транспортної стріч- ки	$м/с$	V_t	3,5
9	Маса поворотної платформи	$кг$	$m_{пл}$	$500 \cdot 10^3$

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

Вимоги до показників якості функціонування системи управління.

Система управління повинна бути статичною.

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

2.2 Розрахунок потужності і вибір привідних електродвигунів

Потужність двигунів повороту стріли визначається по формулі:

$$P_d = \frac{1,5M_{\delta}\omega_{пл}}{\eta_{п}k} 10^{-3}, \quad (2.1)$$

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

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

k – число приводів механізму повороту, рекомендується застосовувати $k = 2$

M_{δ} – момент, обумовлений зусиллям, що виникає на кромках ковшів:

$$M_{\delta} = F_{\delta} R_{\kappa}, \quad (2.2)$$

F_{δ} - зусилля, перпендикулярні осі стріли ротора, що виникають при копанні на кромках ковшів;

R_{κ} – радіус різання в горизонтальній площині, вимірюваний по кромці ковшів; радіус різання залежить від кута нахилу β_c стріли ротора по відношенню до горизонталі і міняється залежно від висоти оброблюваного шару:

$$R_{\kappa} = R_0 \cos \beta_c. \quad (2.3)$$

При горизонтальному розташуванні стріли $R_0 = R_{\kappa}$, цьому положенню відповідає максимальне значення M_{δ} , яке і приймаємо як розрахункове

$$M_{\delta} = M_{\delta_{\max}} = F_{\delta} R_0, \quad (2.4)$$

$$M_{\delta} = M_{\delta_{\max}} = 3,8 \cdot 10^4 \cdot 23,3 = 88,54 \cdot 10^4 \text{ Нм}.$$

$$P_d = \frac{1,5 \cdot 10^{-3} \cdot 88,54 \cdot 2,9 \cdot 10^{-2}}{0,5 \cdot 2} = 43 \text{ кВт}.$$

З каталогу вибираємо два однакові двигуни ДПВ-72 . Їх технічні дані: номінальна потужність $P_n = 67 \text{ кВт}$; номінальна напруга $U_n = 220 \text{ В}$; номінальна частота обертання $n_n = 560 \text{ об/хв}$; номінальний струм $I_n = 338 \text{ А}$; номінальний струм збудження $I_{zn} = 4,3 \text{ А}$, число активних провідників якоря $N = 210$; кількість паралельних гілок обмотки якоря $2a = 2$; число полюсів $2p = 4$; опір обмоток при 20°C : обмотки якоря $R_{яд} = 0,0118 \text{ Ом}$, додаткових полюсів $R_{дпд} = 0,0086 \text{ Ом}$, обмотки збудження $R_{зб} = 3,92 \text{ Ом}$; тривалість включення $ПВ = 25\%$; номінальний магнітний потік $\Phi_n = 0,057 \text{ Вб}$; момент інерції двигуна $J_d = 8,25 \text{ кг} \cdot \text{м}^2$; $\eta = 91\%$

Правильність вибору двигунів перевіряємо після побудови швидкісної діаграми і діаграми навантаження.

2.3 Розрахунок швидкісної діаграми механізму повороту стріли екскаватора

Швидкісна діаграма механізму повороту стріли має три ділянки: розгін з граничним прискоренням, рух із сталою швидкістю і гальмування з допустимим уповільненням.

Допустимі величини прискорення і уповільнення платформи $\dot{\omega}_{i\bar{e}}$ приводимо до валу двигуна

$$\dot{\omega}_{\text{доп}} = \dot{\omega}_{\text{пл}} \cdot i_n ; \quad (2.5)$$

де i_n – передатне число механізму

$$i_n = \frac{\omega_n}{\omega_{\text{пл}}} \quad (2.6)$$

ω_i - кутова швидкість обертання двигуна повороту стріли;

$$\omega_{\text{н}} = \frac{\pi \cdot n_{\text{н}}}{30}, \quad (2.7)$$

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

$$i_{\text{п}} = \frac{58,64}{2,9 \cdot 10^{-2}} = 2022,1.$$

$$\dot{\omega}_{\text{доп}} = 0,005 \cdot 2022,1 = 10,1 \text{ c}^{-2}.$$

Використовуючи значення $\dot{\omega}_{\text{доп}}$, підраховуємо час розгону і уповільнення (гальмування) двигуна

$$t_{\text{р}} = t_{\text{г}} = \frac{\omega_{\text{н}}}{\dot{\omega}_{\text{доп}}}, \quad (2.8)$$

$$t_{\text{р}} = t_{\text{г}} = \frac{58,6}{10,1} = 5,8 \text{ c}.$$

Кути повороту, прохідні платформою під час розгону гальмування

$$\alpha_{\text{р}} = \alpha_{\text{г}} = \dot{\omega}_{\text{пл}} \frac{t_{\text{р}}^2}{2}; \quad (2.9)$$

$$\alpha_{\text{р}} = \alpha_{\text{г}} = 0,005 \cdot \frac{5,8^2}{2} = 84 \cdot 10^{-3} \text{ рад}.$$

$$\alpha_{\text{р}} = \alpha_{\text{г}} = 4,8^{\circ}.$$

Кути повороту платформи за час сталої швидкості

$$\alpha_{\text{о}} = 2 \cdot (\alpha_{\text{і max}} - \alpha_{\text{р}}); \quad (2.10)$$

$$\alpha_{\text{у}} = 2 \cdot (75 - 4,8) = 140,4^{\circ}.$$

$$\alpha_{\text{у}} = 2,5 \text{ рад}.$$

Час роботи поворотного механізму із сталою швидкістю

$$t_y = \frac{\alpha_y \cdot i_{\Pi}}{\omega_H}; \quad (2.11)$$

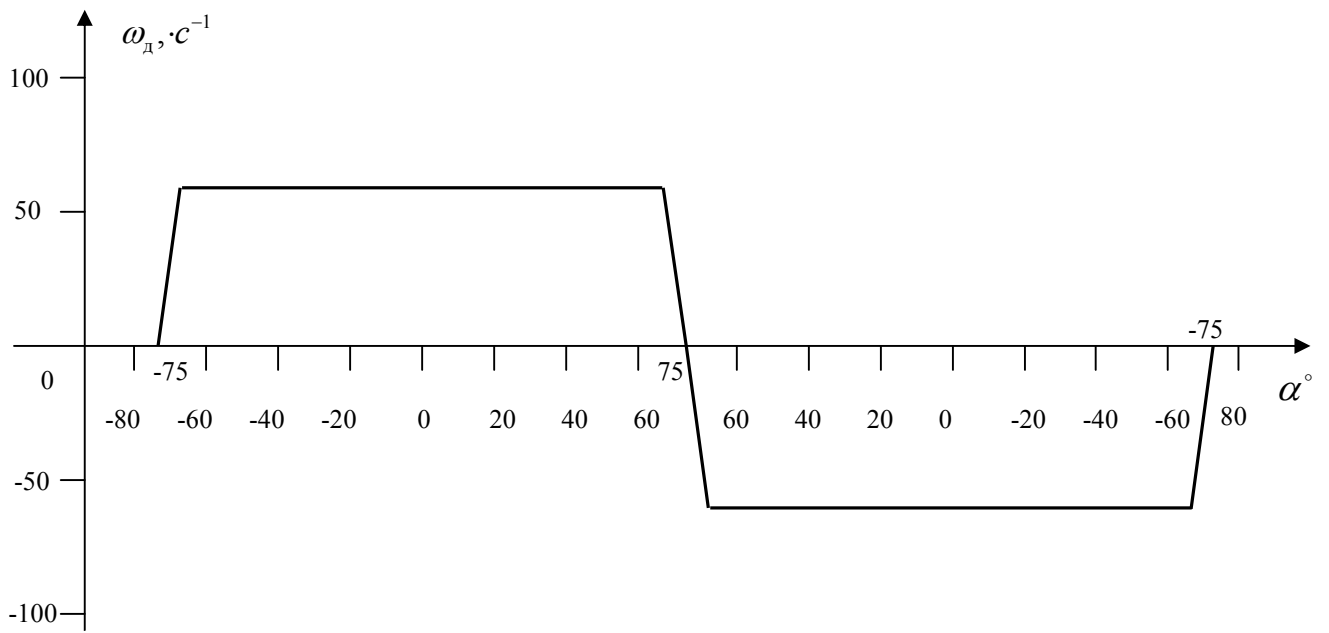
$$t_y = \frac{2,5 \cdot 2022,1}{58,6} = 86,21 \text{ с}.$$

Час циклу, тобто час руху в прямому і зворотному напрямі

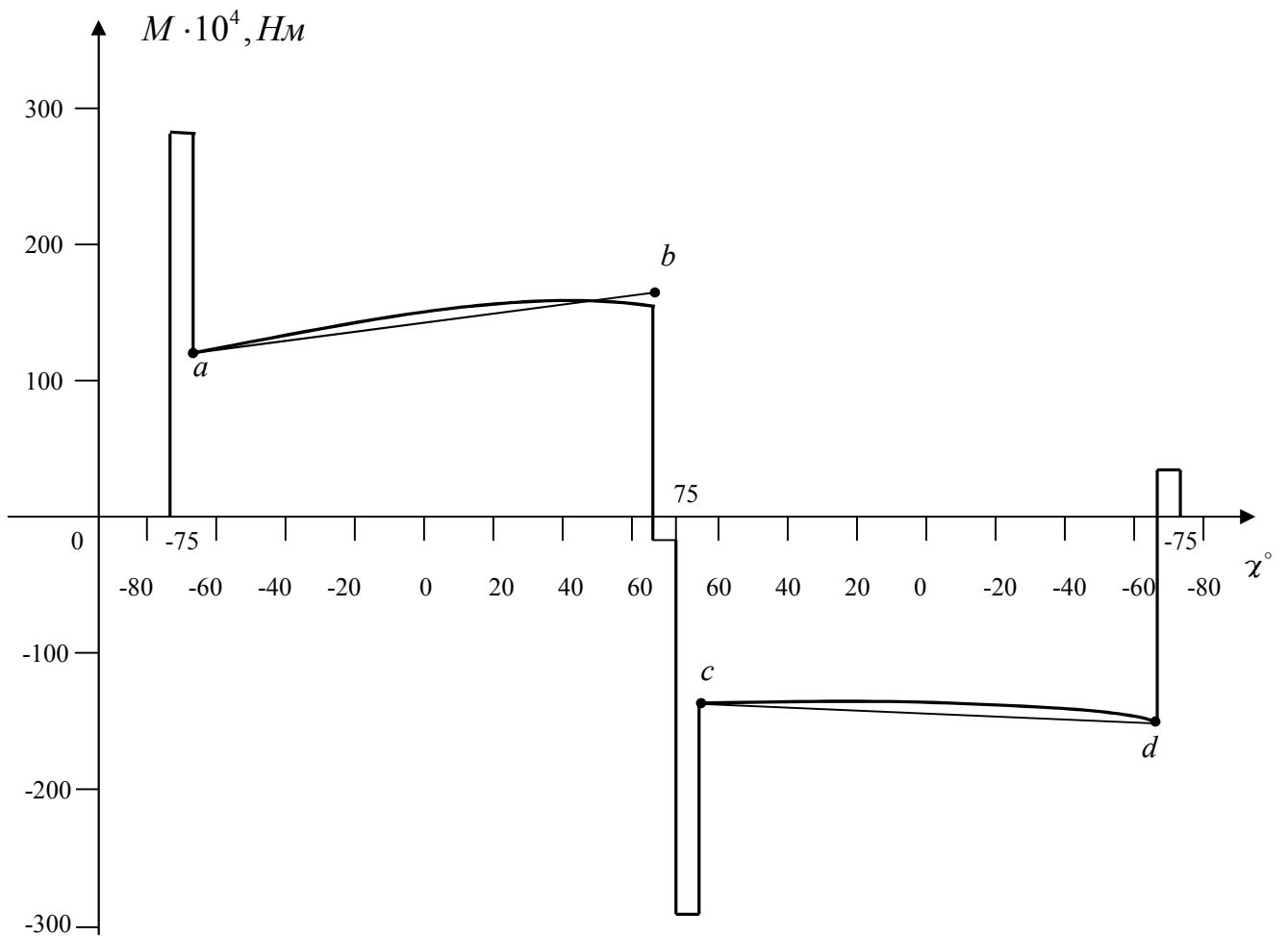
$$T_{\text{ц}} = 2 \cdot (t_p + t_r + t_y), \quad (2.12)$$

$$T_{\text{ц}} = 2 \cdot (5,8 + 5,8 + 86,21) = 195,62 \text{ с}.$$

Швидкісна діаграма механізму повороту стріли роторного екскаватора показана на рис. 2.1 а.



a)



б)

Рисунок 2.1 - Швидкісна а) і навантажувальна б) діаграми роботи механізму повороту стріли екскаватора

2.4 Побудова діаграми навантаження

Параметри швидкісної діаграми використовуємо при побудові діаграми навантаження і перевірки двигунів по нагріву і перевантажувальній здатності.

Для обчислення статичних навантажень, що визначають сумарний момент опору щодо осі обертання поворотної частини, розглянемо схему; приведено на рис.2.2.

Як було зазначено вище, при копанні на кромках ковша виникає зусилля, перпендикулярні осі стріли ротора F_6 і які створюють щодо осі момент $M_6 = F_6 R_k = 680 \cdot 10^4 \text{ Нм}$.

Із-за різних площ конструкції машини, розташованих по різні сторони щодо осі обертання поворотної частини, виникає момент, обумовлений вітровим натиском.

При постійності вітрового натиску момент від вітрового навантаження $M_{\text{вт}}$, визначається таким чином

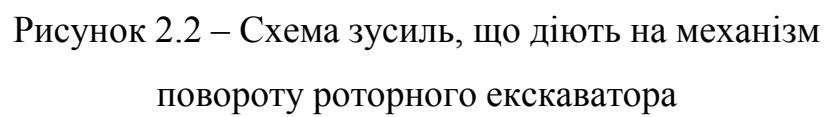
$$M_{\text{вт}} = M_{\text{вт max}} \cos \alpha_{\Pi} = k_{\text{ад}} \lambda_{\text{вт}} M_{\text{пв}} \cos \alpha_{\Pi}, \quad (2.13)$$

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

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

$M_{\text{вт max}}$ – максимальний момент від вітрового навантаження (при куті $\alpha_{\Pi} = 0$).

$$M_{\text{вт}} = 1,4 \cdot 300 \cdot 430 \cdot \cos \alpha_{\Pi} = 180,6 \cdot 10^3 \cdot \cos \alpha_{\Pi} \text{ Нм}$$



При відхиленні осі обертання від вертикалі на кут γ і повороті стріли на кут α_{Π} виникає момент від дії невідновжених сил тяжіння поворотної частини (див. рис. 2.2). Вважаючи, що площина кута γ співпадає з площиною подовжнього перетину екскаватора, момент від дії невідновжених сил визначається виразом

$$M_{\text{нв}} = G_{\text{нв}} R_{\text{цт}} \sin \alpha_{\Pi} \sin \gamma, \quad (2.14)$$

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

У розрахунках можна прийняти

$$G_{\text{нв}} R_{\text{цт}} = g[(m_{\text{пч}} + m_{\text{в пч}}) - m_{\text{вч}}] \frac{L_{\text{пч}}}{3}, \quad (2.15)$$

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

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

$$m_{\text{в пч}} = \frac{Q \gamma_{\text{гр}} L_{\text{пч}}}{3600 \cdot V_{\text{т}} k_{\text{р}}}, \quad (2.16)$$

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

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

$$m_{\text{в пч}} = \frac{1700 \cdot 1,8 \cdot 10^3 \cdot 23,3}{3600 \cdot 3,5 \cdot 1,5} = 2,7738 \cdot 10^3 \text{ кг},$$

$$G_{\text{нв}} R_{\text{цт}} = 9,8 \cdot [(127 + 2,7738) - 120] \cdot 10^3 \cdot \frac{23,3}{3} = 744,24 \cdot 10^3 \text{ Нм}$$

$$M_{\text{нв}} = 744,24 \cdot 10^3 \cdot \sin 10^\circ \cdot \sin \alpha_{\Pi} = 129,236 \cdot 10^3 \cdot \sin \alpha_{\Pi} \text{ Нм}$$

Складові статичного навантаження $M_{ет}$ і $M_{ну}$ залежать від кутового положення і носять активний характер (тобто вони можуть сприяти або перешкоджати руху).

Момент опору обертанню від сил тертя $M_{тр}$ визначається як:

$$M_{тр} = g(m_{пл} + m_{впч}) \frac{D_{кр}}{2d_{тк}} f_k f_{пч} f_{ек}, \quad (2.17)$$

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

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

$$M_{тр} = 9,8 \cdot (500 + 2,7738) \cdot 10^3 \cdot \frac{8,7}{2 \cdot 0,8} \cdot 0,09 \cdot 1,2 \cdot 1,2 \cdot 10^{-2} = 3,543 \cdot 10^3 \text{ Нм}$$

Особливістю побудови діаграми навантаження електроприводу механізму повороту є те, що моменти M_6 і $M_{тр}$ є пасивними, вони завжди перешкоджають руху незалежно від напрямку руху, а складові статичного моменту $M_{вт}$ і $M_{нв}$ залежать від кута повороту, кута нахилу і напрямку вітрового навантаження. Приймаємо, що напрям вітру і ухил екскаватора такі, що сумарне статичне навантаження щодо осі поворотної частини для повороту стріли $M_{ст}$ по і проти годинникової стрілки, тобто для зміни $\alpha_{п}$ від -75° до $+75^\circ$ і від $+75^\circ$ до -75° визначається наступним чином:

$$\alpha_{п} = (-75^\circ \div 0) \quad M_{ст} = |M_6| + |M_{тр}| + |M_{вт}| - |M_{нв}|; \quad (2.18)$$

$$\alpha_{п} = (0 \div +75^\circ) \quad M_{ст} = |M_6| + |M_{тр}| + |M_{вт}| + |M_{нв}|; \quad (2.19)$$

$$\alpha_{п} = (+75^\circ \div 0) \quad M_{ст} = -(|M_6| + |M_{тр}| - |M_{вт}| - |M_{нв}|); \quad (2.20)$$

$$\alpha_{\Pi} = (0 \div -75^\circ) \quad M_{\text{ст}} = -(|M_{\text{б}}| + |M_{\text{тр}}| - |M_{\text{вт}}| + |M_{\text{нв}}|). \quad (2.21)$$

На ділянках розгону і гальмування враховуємо динамічний момент

$$M_{\text{дин}} = J_{\text{м}\Sigma} \cdot \dot{\omega}_{\text{пл}}, \quad (2.22)$$

де $J_{\text{м}\Sigma}$ – сумарний момент інерції механізму поворотної платформи і приведені до швидкості обертання платформи моменти інерції двигунів.

$$J_{\text{м}\Sigma} = J_{\text{пл}} + kJ_{\text{д}} i_{\text{п}}^2, \quad (2.23)$$

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

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

$$J_{\text{м}\Sigma} = 17 \cdot 10^7 + 2 \cdot 8,25 \cdot 2022,1^2 = 23,7 \cdot 10^7 \text{ кг} \cdot \text{м}^2,$$

$$M_{\text{дин}} = 23,7 \cdot 10^7 \cdot 0,005 = 20,4 \cdot 10^5 \text{ Нм}.$$

Динамічний момент $M_{\text{дин}}$ складається із статичним моментом $M_{\text{ст}}$ при розгоні і віднімається при гальмуванні, тобто результуючий момент визначається наступним чином:

$$M_{\text{рез}} = M_{\text{ст}} \pm M_{\text{дин}}. \quad (2.24)$$

Всі результати зводимо в таблицю 2.1. Діаграма навантаження показана на рис.2.1 б.

Таблиця 2.1 – Розрахунок навантажувальної діаграми

Напрямок руху	Складові моменти	$\alpha_{\text{п}}$																	
		75°	$-(75^\circ - \alpha_{\text{п}})$	-65°	-55°	-45°	-35°	-25°	-15°	-5°	0	5°	15°	25°	35°	45°	55°	65°	75°
По годинниковій стрілці	$M_{\text{б}} \cdot 10^4$	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54
	$M_{\text{вт}}$	46742,7	61176,1	7632,9	103587,9	127703,5	147938,9	163679,2	174446,2	179912,8	180600	179912,8	174446,2	163679,2	147938,9	127703,5	103587,9	76324,9	124832,3
	$M_{\text{нв}}$	-124832	-121595,6	-117127,5	-105863,9	-91383,6	-74126,7	-54617,5	-33448,7	-11263,7	0	11263,7	33448,7	54617,5	74126,7	91383,6	105863,9	117127,5	124832,3
	$M_{\text{тр}}$	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1
	$M_{\text{ст}} \cdot 10^4$	81,1	82,9	84,8	88,7	92,5	96,3	99,8	103	105,8	107	108	109,7	110,7	111,1	110,8	109,8	108,2	106,1
	$M_{\text{дин}} \cdot 10^4$	204	204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	204
	$M_{\text{рез}} \cdot 10^4$	285,1	286,9	84,8	88,7	92,5	96,3	99,8	103	105,8	107	108	109,7	110,7	111,1	110,8	109,8	108,2	-97,9
Проти годинникової стрілки	$M_{\text{б}} \cdot 10^4$	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54	88,54
	$M_{\text{вт}}$	46742,7	61176,1	76324,9	103587,9	127703,5	147938,9	163679,2	174446,2	179912,8	180600	179912,8	174446,2	163679,2	147938,9	127703,5	103587,9	76324,9	46742,7
	$M_{\text{нв}}$	-124832	-121595,6	-117127,5	-105863,9	-91383,6	-74126,7	-54617,5	-33448,7	-11263,7	0	11263,7	33448,7	54617,5	74126,7	91383,6	105863,9	117127,5	124832,3
	$M_{\text{тр}}$	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1	3543,1
	$M_{\text{ст}} \cdot 10^4$	-96,7	-94,9	-93	-89,1	-85,3	-81,5	-78	-74,8	-72	-70,8	-69,1	-68,1	-67,1	-66,7	-67	-68	-69,6	-71,7
	$M_{\text{дин}} \cdot 10^4$	-204	-204	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-204
	$M_{\text{рез}} \cdot 10^4$	107,3	109,1	-93	-89,1	-85,3	-81,5	-78	-74,8	-72	-70,8	-69,1	-68,1	-67,1	-66,7	-67	-68	-69,6	-275,7

2.5 Перевірка двигунів по нагріву і перевантажувальній здатності

Для обчислення середньоквадратичного моменту діаграму навантаження розбиваємо на ділянки з інтервалом часу Δt_i (що відповідає $\Delta \alpha_n$) і в межах кожної ділянки визначаємо середнє значення результуючого моменту $M_{\text{рез } i}$.

Середньоквадратичний момент двигуна

$$M_{\text{д экв}} = \frac{1}{\eta_{\text{п}} i_{\text{п}} k} \cdot \sqrt{\frac{\sum_{i=1}^n M_{\text{рез},i}^2 \Delta t_{\text{п},i}}{T_{\text{ц}}}}. \quad (2.25)$$

Заміну ділянок, де графік має форму кривої лінії на прямолінійні ділянки здійснюємо, використовуючи наступну формулу

$$M_{\text{рез } i} = \sqrt{\frac{M_1^2 + M_1 M_2 + M_2^2}{3}}; \quad (2.26)$$

для ділянки ab :

$$M_{\text{рез } 1} = \sqrt{\frac{(88,8^2 + 88,8 \cdot 108,2 + 108,2^2) \cdot 10^8}{3}} = 97,4 \cdot 10^4 \text{ Нм};$$

для ділянки cd :

$$M_{\text{рез } 2} = \sqrt{\frac{(93^2 + 93 \cdot 69,6 + 69,6^2) \cdot 10^8}{3}} = 82,4 \cdot 10^4 \text{ Нм};$$

$$M_{\text{д экв}} = \frac{1}{0,45 \cdot 2022,1 \cdot 2} \cdot \sqrt{\frac{285,1^2 \cdot 5,8 + 97,4^2 \cdot 86,21 + 97,9^2 \cdot 5,8 + 107,3^2 \cdot 5,8 + 82,4^2 \cdot 86,21 + 275,7^2 \cdot 5,8}{195,62}} = 495,9 \text{ Нм}.$$

Номінальний момент двигуна

$$M_{\text{н}} = \frac{P_{\text{н}}}{\omega_{\text{н}}}, \quad (2.27)$$

$$M_{\text{н}} = \frac{67000}{58,6} = 1142 \text{ Нм}.$$

Оскільки виконується умова: $M_{\text{н}} \geq M_{\text{д екв}}$, то вибраний двигун за умовами нагріву відповідає вимогам.

Перевіряємо двигуни по перевантажувальній здатності. Максимальний момент, визначений по діаграмі навантаження і приведений до валу двигуна, повинен бути менше $2,5I_{\text{н}}$, тобто повинно виконуються співвідношення

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} \leq M_{\text{д max}}, \quad (2.28)$$

де $M_{\text{рез max}}$ – максимальний момент, визначений по діаграмі навантаження;

$$M_{\text{рез max}} = 285,1 \cdot 10^4 \text{ Нм};$$

$M_{\text{д max}}$ – максимально допустимий момент двигуна

$$M_{\text{д max}} = \lambda M_{\text{н}}, \quad (2.29)$$

λ – перевантажувальна здатність двигуна; як відомо, для двигунів постійного струму $\lambda = 2 \div 2,5$

$$M_{\text{д max}} = 2,5 \cdot 1142 = 2855 \text{ Нм},$$

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} = \frac{285,1}{0,45 \cdot 2022,1 \cdot 2} = 1576 \text{ Нм}.$$

Отже, вибраний двигун по перевантажувальній здатності задовольняє вимогам.

2.6 Вибір основного електроустаткування електроприводу

Відповідно до рекомендацій [1] і оглядом вживаних систем електроприводу основних механізмів роторного екскаватора, для електроприводу повороту стріли приймаємо систему Г-Д. Вибираємо генератор П-111, номінальна потужність $P_{нг} = 145 \text{ кВт}$, частота обертання $n_{нг} = 1450 \text{ об/хв}$, напруга $U_{нг} = 460 \text{ В}$, струм $I_{нг} = 315 \text{ А}$, ККД $\eta_a = 0,928$, напруга незалежної обмотки збудження $U_{зб\text{ нг}} = 220 \text{ В}$, струм незалежної обмотки збудження $I_{зб\text{ нг}} = 5,8 \text{ А}$, магнітний потік полюса $\Phi_{нг} = 0,037 \text{ Вб}$, опір обмотки при 20°C : $R_{яг} = 0,0257 \text{ Ом}$, $R_{днг} = 0,0105 \text{ Ом}$, число полюсів $2p = 4$, число витків обмотки збудження $W_{aa} = 780$, опір незалежної обмотки $R_{збг} = 21,5 \text{ Ом}$.

Для генератора вибираємо тиристорний збудник БУВ-3603-12, схема випрямлення – реверсивна трифазна нульова, потужність $P_n = 9,2 \text{ кВт}$, випрямлена напруга $U_{дн} = 230 \text{ В}$, випрямлений струм $I_{дн} = 40 \text{ А}$.

Анодний трансформатор типа ТСЗП-25/0,7-77, втрати холостого ходу $P_{xx} = 170 \text{ кВт}$, втрати короткого замикання $P_{кз} = 1300 \text{ Вт}$.

2.7 Побудова статичних характеристик електроприводу Г-Д механізму повороту стріли екскаватора

На рис.2.3 зображена принципова схема системи Г-Д.

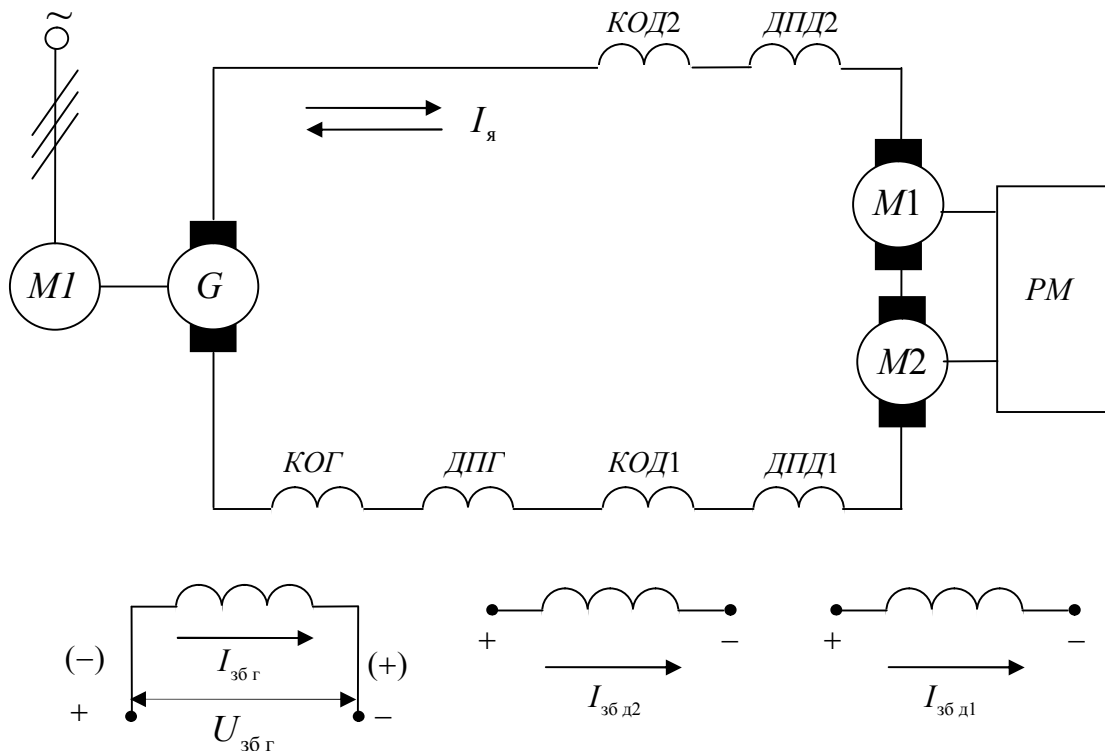


Рисунок 2.3 – Принципова схема розімкненої системи Г-Д

На схемі прийняті наступні позначення:

PM – робочий механізм;

$M1, M2, G, MG$ – двигуни даного механізму, генератор і приводний двигун генератора;

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

$I_{зб\ д1}, I_{зб\ д2}, I_{зб\ г}$ – струми в обмотках збудження двигунів $M1$ і $M2$, а також обмотки збудження генератора відповідно.

Як випливає з схеми, двигуни $M1$ і $M2$ сполучені послідовно і їх якірні обмотки утворюють якірний ланцюг з обмотками якоря генератора.

Якірні обмотки двигунів і генератора сполучені послідовно і утворюють якірний ланцюг. Сумарний опір якірного ланцюга

$$R_{\Sigma} = k \cdot R_{\Sigma \text{д}} + R_{\Sigma \text{Г}} \quad (2.30)$$

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

$$R_{\Sigma \text{д}} = k_t (R_{\text{я д}} + R_{\text{дп д}} + R_{\text{ко д}}) ; \quad (2.31)$$

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

k_t – температурний коефіцієнт $k_t = 1,36$

$$R_{\Sigma \text{д}} = 1,36 \cdot (0,0118 + 0,0086) = 0,03 \text{ Ом}$$

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

$$R_{\Sigma \text{Г}} = k_t (R_{\text{я Г}} + R_{\text{дп Г}} + R_{\text{ко Г}}) ; \quad (2.32)$$

$R_{\text{я Г}}$, $R_{\text{дп Г}}$, $R_{\text{ко Г}}$ – опір обмотки якоря, додаткових полюсів і компенсаційної обмотки генератора, Ом ;

$$R_{\Sigma \text{Г}} = 1,36 \cdot (0,0257 + 0,0105) = 0,05 \text{ Ом}$$

$$R_{\Sigma} = 2 \cdot 0,03 + 0,05 = 0,11 \text{ Ом}$$

Якщо розглядати ЕДС генератора $E_{\text{Г}}$ як джерело живлення двигунів, то рівняння електромеханічної і механічної характеристик електроприводу матимуть вигляд:

$$\omega_{\text{д}} = \frac{E_{\text{Г}}}{k_{\text{д}} k_{\text{д}} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\Sigma}}{k_{\text{д}} k_{\text{д}} \Phi_{\text{нд}}} = \omega_0 - \Delta \omega_{\text{ст}} ; \quad (2.33)$$

$$\omega_{\text{д}} = \frac{E_{\text{г}}}{k k_{\text{д}} \Phi_{\text{нд}}} - M_{\text{д}} \frac{R_{\text{я}\Sigma}}{k (k_{\text{д}} \Phi_{\text{нд}})^2} = \omega_0 - \Delta \omega_{\text{ст}} \quad (2.34)$$

де $I_{\text{я}}$ – струм якiрного кола, A ;

$M_{\text{д}}$ – момент, що розвивається одним двигуном, Hm ;

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

$\Phi_{\text{нд}}$ – номiнальний магнiтний потiк двигуна, $Bб$.

Добуток $k_{\text{д}}$ на $\Phi_{\text{нд}}$ (позначимо надалi як c) можна визначити з виразу для природної електромеханiчної характеристики двигуна:

$$c = k_{\text{д}} \Phi_{\text{нд}} = \frac{U_{\text{нд}} - I_{\text{нд}} R_{\text{я}\Sigma}}{\omega_{\text{нд}}} ; \quad (2.35)$$

$U_{\text{нд}}$, $I_{\text{нд}}$, $\omega_{\text{нд}}$ – номiнальнi напруга, B , струм, A , i швидкiсть c^{-1} двигуна;

$$c = k_{\text{д}} \Phi_{\text{нд}} = \frac{220 - 338 \cdot 0,03}{58,6} = 3,58$$

При побудовi природних електромеханiчної i механiчної характеристик по рiвняннях (2.33) i (2.34) в якостi $E_{\text{г}}$ пiдставляємо величину $k U_{\text{нд}}$.

На рис. 2.4 показано електромеханiчнi i механiчнi характеристики розiмкненої системи Г-Д (пунктирнi прямi), побудованi для номiнального значення напруги двигуна $U = U_{\text{нд}}$ i для значень напруги $U = 0,75 U_{\text{нд}}$, $U = 0,5 U_{\text{нд}}$, $U = 0,25 U_{\text{нд}}$ i $U = 0$ (характеристика динамiчного гальмування), а також для вiд'ємних значень напруги.

Кожна характеристика побудована по двох точках: точцi, вiдповiднiй режиму iдеального холостого ходу i точцi, вiдповiднiй номiнальному навантаженню. Для вiд'ємних значень напруги характеристики будуються аналогiчно i знаходяться в 3 i 4 квадрантах.

1 точка – відповідає ідеальному холостому ходу

$$I_{\text{я}} = 0, \quad M = 0, \quad \omega_0 = \frac{k U_{\text{д}}}{k k_{\text{д}} \Phi_{\text{нд}}} = \frac{U_{\text{д}}}{k_{\text{д}} \Phi_{\text{нд}}}.$$

$$\omega_{01} = \frac{220}{3,58} = 61,45 \text{ c}^{-1} \quad \omega_{02} = 0,75 \cdot 61,45 = 46,1 \text{ c}^{-1}.$$

$$\omega_{03} = 0,5 \cdot 61,45 = 30,73 \text{ c}^{-1} \quad \omega_{04} = 0,25 \cdot 61,45 = 15,36 \text{ c}^{-1}.$$

$$\omega_{05} = 0.$$

2 точка – відповідає номінальному навантаженню

$$I_{\text{я}} = I_{\text{нд}} = 338 \text{ A}. \quad M = M_{\text{н}} = k_{\text{д}} \Phi_{\text{нд}} I_{\text{нд}} = 3,58 \cdot 338 = 1210 \text{ Нм}.$$

Швидкість для кожного значення напруги визначаємо згідно (2.33)

$$\begin{aligned} \omega &= \frac{k U_{\text{д}}}{k k_{\text{д}} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\Sigma}}{k k_{\text{д}} \Phi_{\text{нд}}} = \\ &= \frac{2}{2 \cdot 3,58} \cdot U_{\text{д}} - 338 \cdot \frac{0,11}{2 \cdot 3,58} = 0,279 \cdot U_{\text{д}} - 5,19. \end{aligned}$$

$$\omega_1 = 0,279 \cdot 220 - 5,19 = 56,26 \text{ c}^{-1}.$$

$$\omega_2 = 0,279 \cdot 0,75 \cdot 220 - 5,19 = 40,91 \text{ c}^{-1}.$$

$$\omega_3 = 0,279 \cdot 0,5 \cdot 220 - 5,19 = 25,54 \text{ c}^{-1}.$$

$$\omega_4 = 0,279 \cdot 0,25 \cdot 220 - 5,19 = 10,17 \text{ c}^{-1}.$$

$$\omega_5 = 0 - 5,45 = -5,45 \text{ c}^{-1}.$$

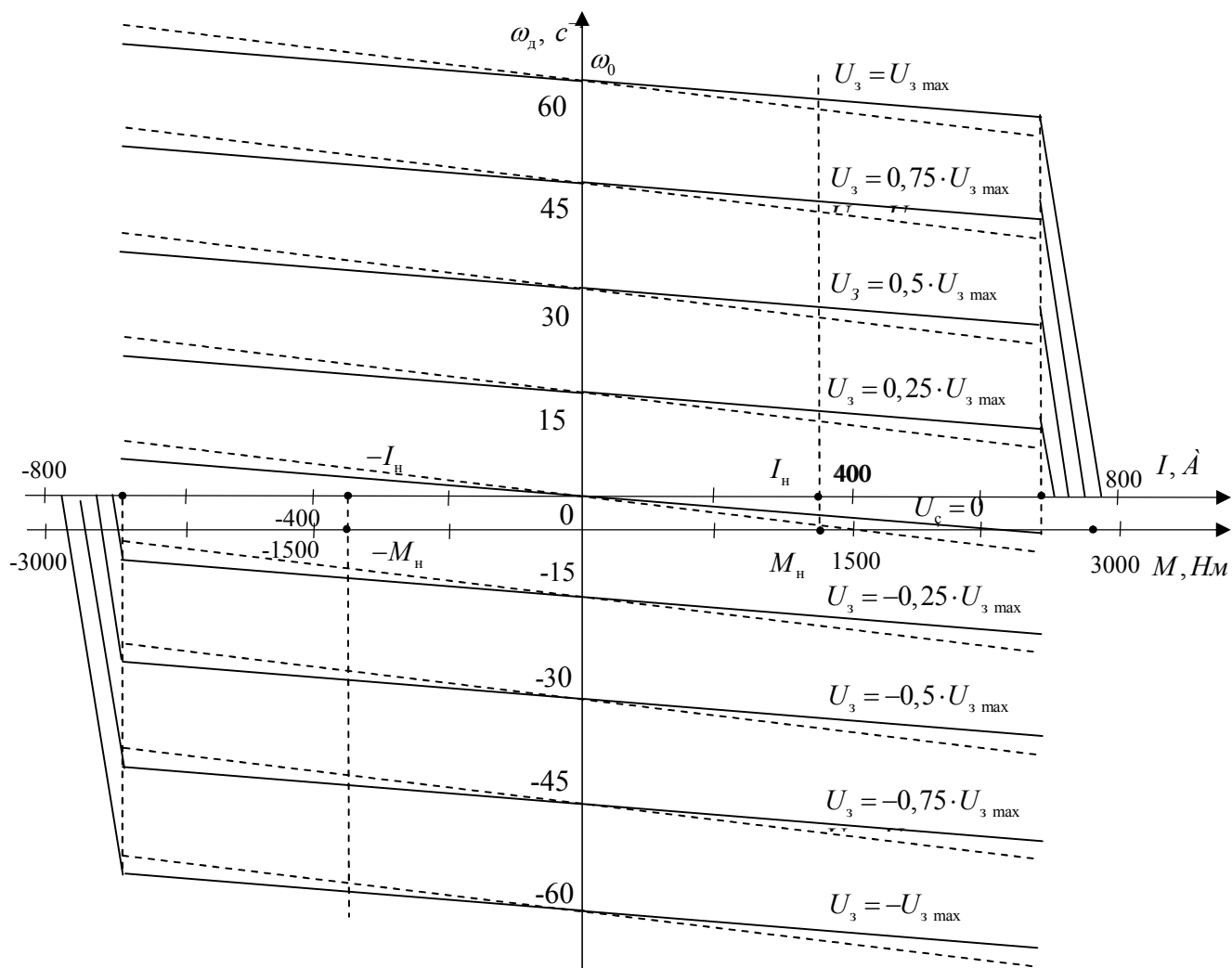


Рисунок 2.4 – Статичні електромеханічні та механічні характеристики розімкненої і замкнутої САУ

Враховуючи збільшення сумарного опору якірному ланцюгу нескладно встановити, що жорсткість механічної характеристики $\beta = \frac{k (k_d \Phi_{нд})^2}{R_{я\Sigma}}$ у системі Г-Д нижче за жорсткість природної характеристики двигуна.

2.8 Побудова статичних характеристик замкнутої системи

До електроприводів більшості основних механізмів роторних екскаваторів пред'являються підвищені вимоги відносно жорсткості робочої частини механічної характеристики і обмеження максимального значення моменту, що розвивається, тобто необхідності формування “екскаваторної” характеристики, показаної на рис. 2.5.

На робочій ділянці 1 у міру зростання навантаження потрібне аж до струму відсічення $I_{\text{я}} = I_{\text{відс}}$ (відповідає моменту відсічення $M_{\text{відс}}$) пропорційне

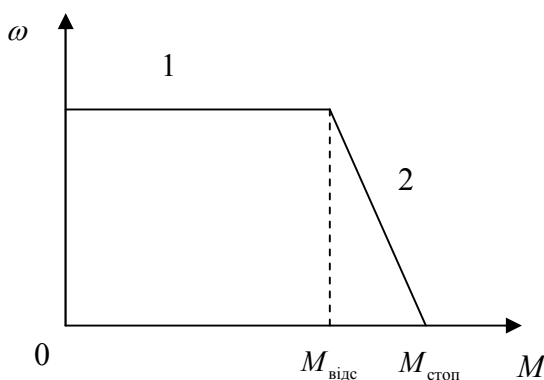


Рисунок 2.5 – Екскаваторна механічна характеристика

струму $I_{\text{я}}$ збільшення ЕДС генератора, $E_{\text{г}}$ чим компенсується зростаюче падіння напруги на опорі $R_{\text{я}\Sigma}$ і забезпечується підвищення жорсткості робочої ділянки характеристики. Подальше збільшення навантаження повинне супроводжуватися пропорційним різниці $\Delta I_{\text{я}} = I_{\text{я}} - I_{\text{відс}}$ зниженням $E_{\text{г}}$ і, отже, швидкості приводу, яка до-

сягає нуля $\omega = 0$ при струмі $I_{\text{я}} = I_{\text{стоп}}$ (відповідає моменту стопоріння $M_{\text{д}} = M_{\text{стоп}}$) і ЕДС генератора $E_{\text{г}} = I_{\text{стоп}} R_{\text{я}\Sigma}$ (ділянка 2 характеристики).

Забезпечити формування такої характеристики можна тільки в замкнутих САУ електроприводом. Як указувалося вище, на роторних екскаваторах використовуються САУ, що мають структуру з підсумовуючим підсилювачем, на вході якого підсумовуються сигнали завдання і сигнали зворотних зв'язків. Зворотні зв'язки можуть бути різних типів: жорсткі і гнучкі ЗЗ по напрузі генератора, ЗЗ за швидкістю двигуна, ЗЗ по струму з відсіченням. Для електроприводу повороту стріли використовуємо САУ із ЗЗ за швидкістю, гнучким ЗЗ по струму і ЗЗ по струму з відсіченням.

Функціональна схема показана на рис. 2.6. На схемі, для спрощення подальших розрахунків, показаний один еквівалентний двигун.

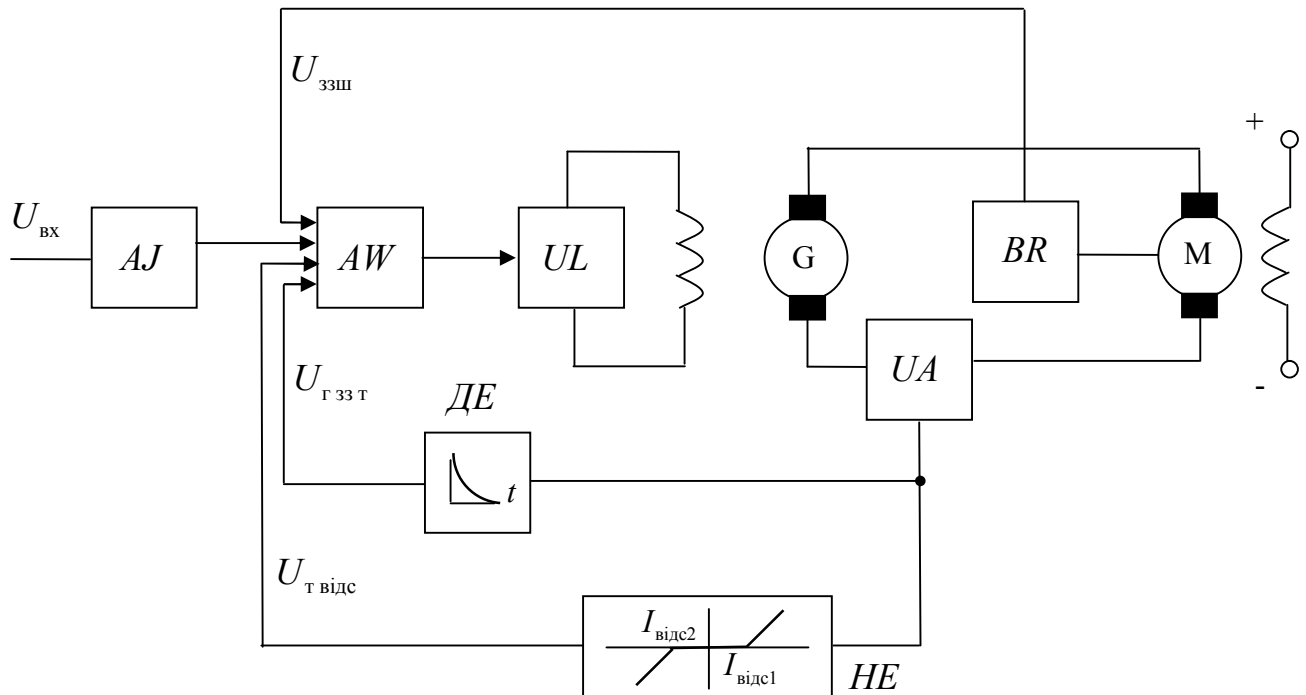


Рисунок 2.6 – Функціональна схема замкненої системи автоматичного управління електроприводу Г-Д

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

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

$$I_{\text{відс2}} \leq I_{\text{я}} \leq I_{\text{відс1}} . \quad (2.36)$$

У даному приводі зазвичай витримується рівність

$$I_{\text{відс1}} = |I_{\text{відс2}}| = I_{\text{відс}} = 2 I_{\text{нд}} . \quad (2.37)$$

Рівняння електромеханічної характеристики має вид

$$\begin{aligned} \omega_{\text{д}} &= U_3 \frac{k_{\Sigma}}{k_{\text{с}} + k_{\Sigma} k_{\text{ззш}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{k_{\text{с}} + k_{\Sigma} k_{\text{ззш}}} - 1(\Delta I) (I_{\text{я}} - I_{\text{відс}}) \frac{k_{\text{т відс}} k_{\Sigma}}{k_{\text{с}} + k_{\Sigma} k_{\text{ззш}}} = \\ &= \omega_0 - \Delta \omega_{\text{ст1}} - 1(\Delta I) \Delta \omega_{\text{ст2}} , \end{aligned} \quad (2.38)$$

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

$$k_{\Sigma} = k_{AW} k_{UL} k_{\Gamma} ; \quad (2.39)$$

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

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

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

Коефіцієнт посилення генератора

$$k_{\Gamma} = \frac{E_{\text{нг}}}{U_{\text{нзбг}}} = \frac{U_{\text{нг}} + I_{\text{нг}} R_{\text{я}\Sigma\Gamma}}{U_{\text{нзбг}}} , \quad (2.40)$$

де $U_{\text{нг}}$, $U_{\text{нзбг}}$ – номінальна напруга на якорі і номінальна напруга збудження генератора, В;

$I_{\text{нг}}$ – номінальний струм генератора, А.;

$E_{\text{нг}}$ – номінальна ЕДС генератора

$$E_{\text{нГ}} = U_{\text{нГ}} + I_{\text{нГ}} R_{\text{я}\Sigma\Gamma} . \quad (2.41)$$

$$k_{\Gamma} = \frac{460 + 315 \cdot 0,05}{220} = 2,13 .$$

Коефіцієнт посилення тиристорного збудника приблизно можна визначити як

$$k_{UL} = \frac{U_{\text{днзб}}}{U_y} , \quad (2.42)$$

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

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

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

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

$$k_{\Sigma} = 0,1 \cdot 23 \cdot 2,13 = 4,9 .$$

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

$$\Delta\omega_{\text{ст1}} = I_{\text{нд}} \frac{R_{\text{я}\Sigma}}{k c + k_{\Sigma} k_{\text{ззш}}} , \quad (2.43)$$

звідки

$$k_{\text{ззш}} = I_{\text{нд}} \frac{R_{\text{я}\Sigma}}{k_{\Sigma} \Delta\omega_{\text{ст1}}} - \frac{k c}{k_{\Sigma}} . \quad (2.44)$$

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

$$\Delta \omega_{\text{ст1}} = 0,5 \Delta \omega_{\text{ст}} = 0,5 \frac{I_{\text{н}} R_{\Sigma}}{k \cdot k_{\Sigma} \Phi_{\text{нд}}}, \quad (2.45)$$

ω_0 – швидкість ідеального холостого ходу

$$\Delta \omega_{\text{ст1}} = 0,5 \cdot \frac{338 \cdot 0,11}{2 \cdot 3,58} = 2,3 \text{ c}^{-1},$$

$$k_{\text{зш}} = 338 \cdot \frac{0,11}{4,9 \cdot 2,3} - \frac{2 \cdot 3,58}{4,9} = 3,3 - 1,46 = 1,84.$$

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

$$k_{\text{т відс}} = \frac{U_{\text{з max}} - I_{\text{стоп}} R_{\Sigma} k_{\Sigma}}{I_{\text{стоп}} - I_{\text{відс}}}. \quad (2.46)$$

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

$$I_{\text{відс}} = 2 I_{\text{нд}}; \quad I_{\text{стоп}} = 2,3 I_{\text{нд}}. \quad (2.47)$$

$$I_{\text{відс}} = 2 \cdot 338 = 676 \text{ A},$$

$$I_{\text{стоп}} = 2,3 \cdot 338 = 780,8 \text{ A}.$$

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

$$\omega_0 = U_{\text{з max}} \frac{k_{\Sigma}}{k c + k_{\Sigma} k_{\text{зш}}} = \frac{U_{\text{нд}}}{c}, \quad (2.48)$$

звідки

$$U_{\text{з max}} = \frac{U_{\text{нд}} (k c + k_{\Sigma} k_{\text{зш}})}{c k_{\Sigma}}. \quad (2.49)$$

$$U_{\text{з max}} = \frac{440 \cdot (2 \cdot 3,58 + 4,9 \cdot 1,84)}{2 \cdot 3,58 \cdot 4,9} = 202,3 \text{ В}$$

$$k_{\text{т відс}} = \frac{101,4 - 780,8 \cdot \frac{0,11}{9,8}}{780,8 - 676} = 0,91.$$

Як видно з виразу (2.46), забезпечення абсолютно м'якої характеристики на ділянці відсічення (при $I_{\text{відс}} = I_{\text{стоп}}$) зажадало б нескінченно великого коефіцієнта посилення $k_{\text{т відс}}$. Практично максимум $k_{\text{т відс}}$ обмежений фізичною реалізовуємою величиною коефіцієнта посилення $k_{\text{т відс}}$ і необхідністю збереження стійкості САУ в режимі стопоріння.

Остаточно по (2.38) отримаємо

$$\begin{aligned} \omega_d &= \frac{9,8}{2 \cdot 3,58 + 9,8 \cdot 0,92} \cdot U_3 - \frac{0,11}{2 \cdot 3,58 + 9,8 \cdot 0,92} \cdot I_{\text{я}} - \\ &- \frac{0,91 \cdot 9,8}{2 \cdot 3,58 + 9,8 \cdot 0,92} - (\Delta I) \cdot (I_{\text{я}} - I_{\text{відс}}) = \\ &= 6,92 \cdot U_3 - 5,92 \cdot 10^{-3} I_{\text{я}} - 0,56 \cdot \Delta I \cdot (I_{\text{я}} - I_{\text{відс}}). \end{aligned} \quad (2.50)$$

За наслідками розрахунків будуюмо статичні характеристики замкнутої системи (рис. 2.4, суцільні прямі).

2.9 Дослідження динамічних характеристик системи

Для дослідження динамічних характеристик розробляємо алгоритмічну схему системи. Алгоритмічна схема системи, побудована відповідно до функціональної схеми рисунка 2.6, показана на рис.2.7.

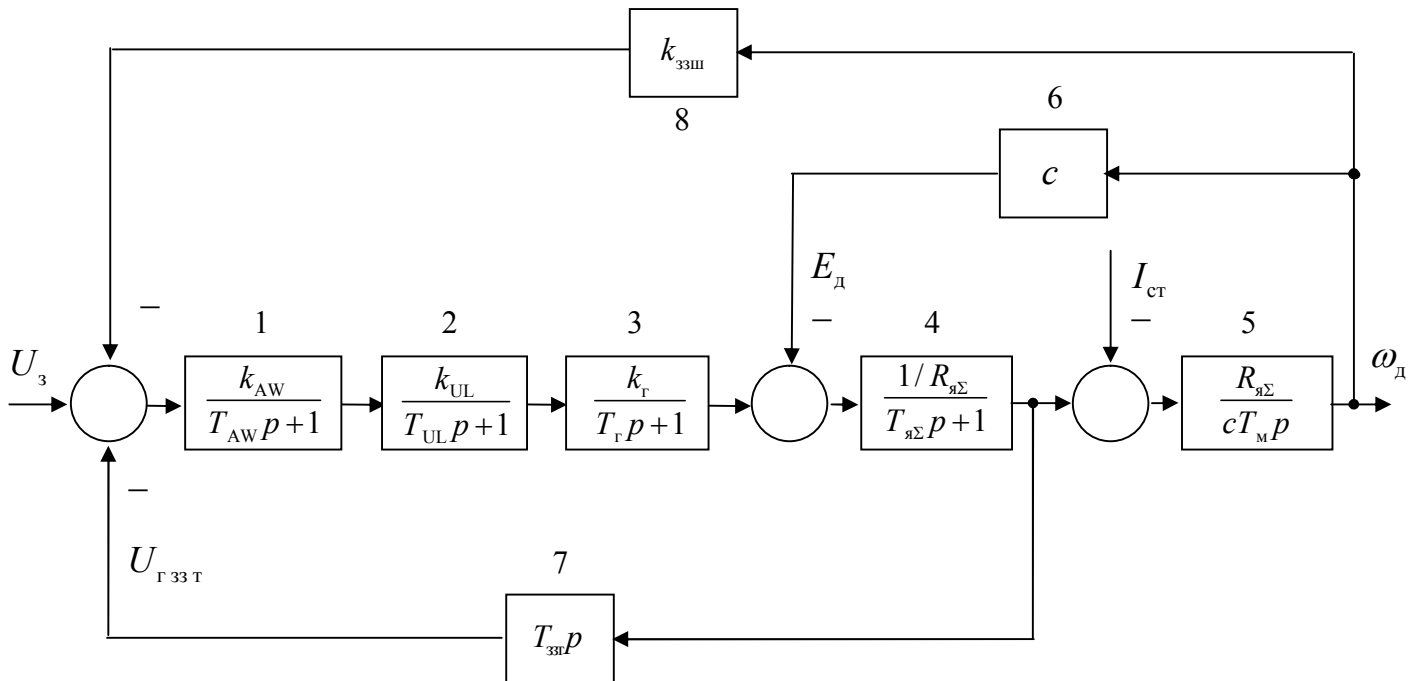


Рисунок 2.7 – Алгоритмічна схема замкненої системи автоматичного управління електроприводу Г-Д

У алгоритмічній схемі не приведений зворотний зв'язок по струму з відсіченням, оскільки він не бере участь в робочих режимах, і задатчик інтенсивності, оскільки його вплив можна врахувати шляхом подачі завдання $U_3(t)$, що лінійно змінюється в часі.

Проводимо виведення передавальних функцій системи по задаючій і збурюючій діях. Для спрощення розрахунків без істотної погрішності можна виключити з розгляду постійні часу напівпровідникових суматорів AW і тиристорних збудників UL , постійні часу яких більш ніж в 50 разів менші за T_{Γ} . Отримаємо алгоритмічну схему рис. 2.8.

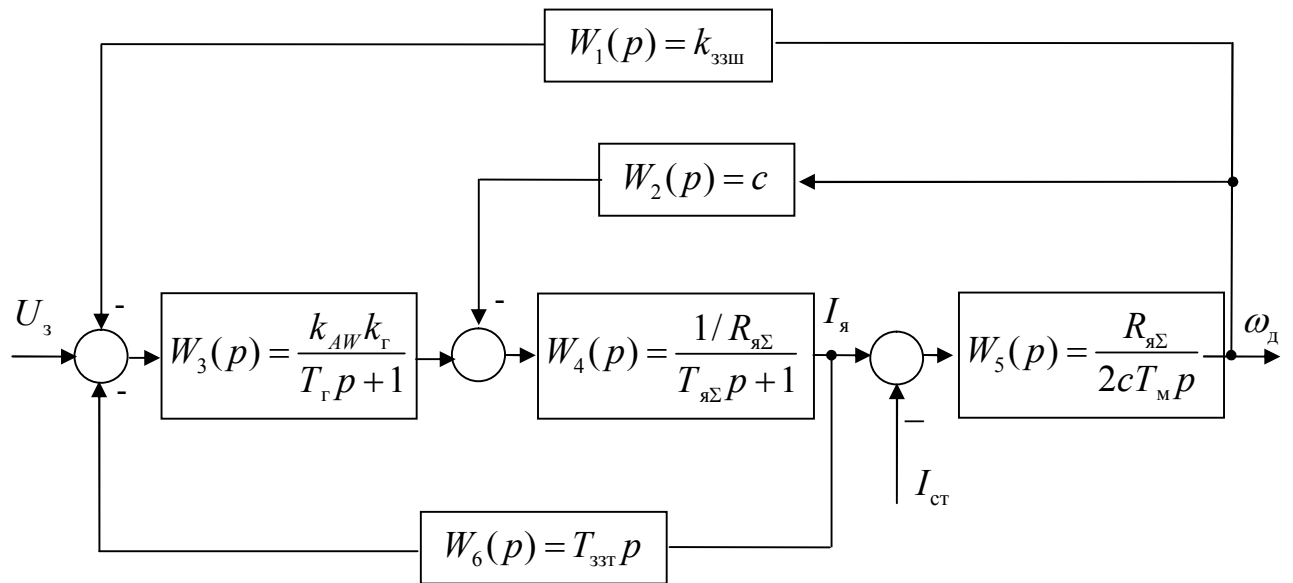
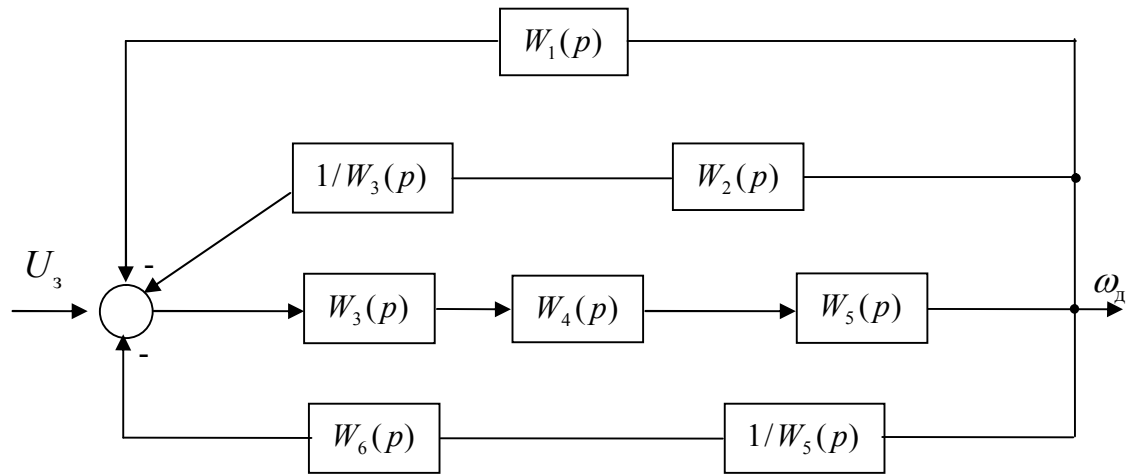


Рисунок 2.8 – Спрощена алгоритмічна схема системи

Користуючись алгоритмічною схемою рис. 2.8, можна отримати передавальні функції по задаючій і збурюючій діях.

Знайдемо передавальну функцію швидкості по задаючій дії $W^1(p) = \frac{\omega_d(p)}{U_3(p)}$. При виведенні передавальної функції по задаючій дії вважаємо

$I_{ст} = 0$. Перетворимо схему рис. 2.8 до вигляду, показаному на рис.2.9 а.



а)

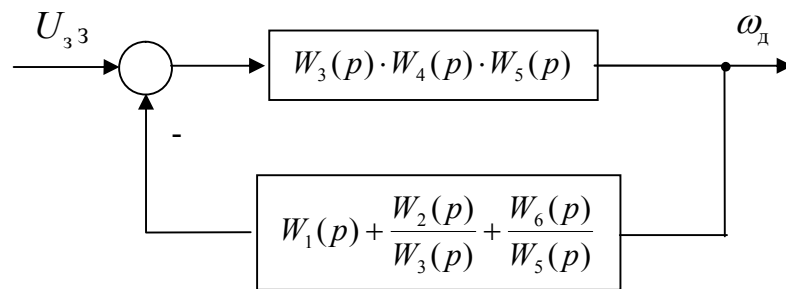


Рисунок 2.9 – Перетворення алгоритмічної схеми рис. 2.8

для отримання передавальної функції $W^1(p) = \frac{\omega_d(p)}{U_3(p)}$:

а) перший етап перетворення схеми;

б) другий етап перетворення схеми

Перетворюючи поралеельно-зустрічне з'єднання ланок рис. 2.9 а, отримаємо схему, показану на рис 2.9 б. Передавальна функція швидкості по задаючій дії визначається наступним чином:

$$\begin{aligned}
 W^1(p) &= \frac{\omega_d(p)}{U_3(p)} = \frac{W_3(p)W_4(p)W_5(p)}{1 + \left(W_1(p) + \frac{W_2(p)}{W_3(p)} + \frac{W_6(p)}{W_5(p)} \right) \cdot W_3(p)W_4(p)W_5(p)} = \\
 &= \frac{W_3(p)W_4(p)W_5(p)}{1 + W_1(p) \cdot W_3(p)W_4(p)W_5(p) + W_2(p)W_4(p)W_5(p) + W_3(p)W_4(p)W_6(p)}. \quad (2.51)
 \end{aligned}$$

Запишемо значення передавальних функцій

$$W_1(p) = k_{33\text{ш}} , \quad (2.52)$$

$$W_2(p) = c , \quad (2.53)$$

$$W_3(p) = \frac{k_{AW} \cdot k_{\Gamma}}{T_{\Gamma} p + 1} , \quad (2.54)$$

$$W_4(p) = \frac{1/R_{\Sigma}}{T_{\Sigma} p + 1} , \quad (2.55)$$

$$W_5(p) = \frac{R_{\Sigma}}{c T_{\text{м}} p} , \quad (2.56)$$

$$W_6(p) = T_{33\text{т}} p . \quad (2.57)$$

Підставляючи передавальні функції (2.52) – (2.57) в (2.51), отримаємо

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

$$= \frac{k_{AW} \cdot k_{\Gamma} / c}{T_{\Gamma} T_{\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\Sigma} + k_{AW} k_{\Gamma} \frac{T_{33\Gamma}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{AW} k_{\Gamma} \cdot k_{33\Pi} / c + 1} \cdot (2.58)$$

З урахуванням вищенаведених позначень для $k_{\text{д}}$ і k_{Σ} отримаємо

$$\begin{aligned} W^1(p) &= \frac{\omega_{\text{д}}(p)}{U_3(p)} = \\ &= \frac{k_{\Sigma} k_{\text{д}}}{T_{\Gamma} T_{\Gamma} T_{\Sigma} p^3 + \left(T_{\Gamma} T_{\Gamma} + T_{\Gamma} T_{\Sigma} + \frac{k_{\Sigma} T_{\Gamma} T_{33\Gamma}}{R_{\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\Gamma}) p + k_{\Sigma} k_{33\Pi} k_{\text{д}} + 1}. \end{aligned} \quad (2.59)$$

Визначимо передавальну функцію струму по задаючій дії $W^2(p) = \frac{I_{\text{я}}(p)}{U_3(p)}$.

Для цього перетворимо алгоритмічну схему так, як показано на рис.2.10.

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

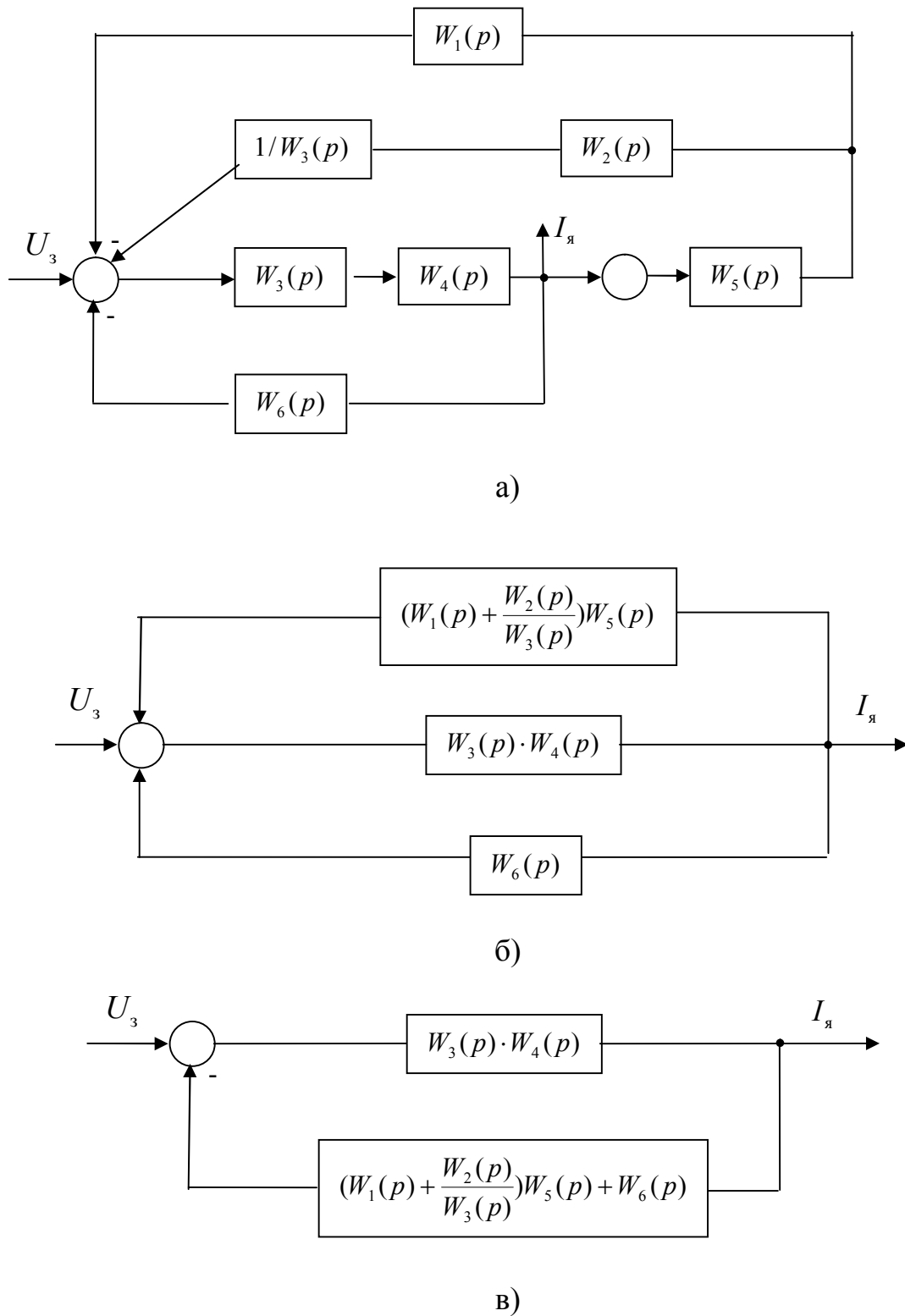


Рисунок 2.10 – Перетворення алгоритмічної схеми рис. 2.9,а

для отримання передавальної функції $W^2(p) = \frac{I_{\text{я}}(p)}{U_3(p)}$.

а) перший етап перетворення схеми;

б) другий етап перетворення схеми

б) третій етап перетворення схеми

$$\begin{aligned}
&= \frac{k_{\Gamma} k_{AW} T_{\text{м}} p / R_{\text{я}\Sigma}}{T_{\Gamma} T_{\text{я}\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Gamma} k_{AW} T_{33\text{T}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{\Gamma} k_{AW} k_{33\text{ш}} k_{\text{д}} + 1} = \\
&= \frac{k_{\Sigma} T_{\text{м}} p / R_{\text{я}\Sigma}}{T_{\Gamma} T_{\text{я}\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Gamma} k_{AW} T_{33\text{T}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{\Sigma} k_{33\text{ш}} k_{\text{д}} + 1}. \quad (2.60)
\end{aligned}$$

Знайдемо передавальну функцію струму по задаючій дії

$$\begin{aligned}
W^2(p) &= \frac{I_{\text{я}}(p)}{U_3(p)} = \frac{W_3(p)W_4(p)}{1 + \left[\left(W_1(p) + \frac{W_2(p)}{W_3(p)} \right) \cdot W_5(p) + W_6(p) \right] \cdot W_3(p)W_4(p)} = \\
&= \frac{\left(k_{AW} k_{\Gamma} \frac{1}{R_{\text{я}\Sigma}} \right) / [(T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1)]}{1 + \left[\left(k_{33\text{ш}} + \frac{c(T_{\Gamma} p + 1)}{k_{AW} k_{\Gamma}} \right) \frac{R_{\text{я}\Sigma}}{c T_{\text{м}} p} + T_{33\text{T}} p \right] \cdot \frac{k_{AW} k_{\Gamma} / R_{\text{я}\Sigma}}{(T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1)}} = \\
&= \frac{\left(k_{AW} k_{\Gamma} \frac{1}{R_{\text{я}\Sigma}} \right) / [(T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1)]}{(T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1) \left\{ (T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1) + \left[\left(k_{33\text{ш}} + \frac{c(T_{\Gamma} p + 1)}{k_{AW} k_{\Gamma}} \right) \frac{R_{\text{я}\Sigma}}{c T_{\text{м}} p} + T_{33\text{T}} p \right] \cdot \frac{k_{AW} k_{\Gamma}}{R_{\text{я}\Sigma}} \right\}} = \\
&= \frac{k_{AW} k_{\Gamma} / R_{\text{я}\Sigma}}{(k_{AW} k_{\Gamma} / R_{\text{я}\Sigma}) \left[\frac{(T_{\Gamma} p + 1)(T_{\text{я}\Sigma} p + 1) c R_{\text{я}\Sigma} T_{\text{м}} p + [k_{AW} k_{\Gamma} k_{33\text{ш}} + c(T_{\Gamma} p + 1)] R_{\text{я}\Sigma} + c T_{33\text{T}} p k_{AW} k_{\Gamma} T_{\text{м}} p}{c k_{AW} \cdot k_{\Gamma} T_{\text{м}} p} \right]} = \\
&= \frac{k_{\Gamma} k_{AW} T_{\text{м}} p / R_{\text{я}\Sigma}}{T_{\Gamma} T_{\text{я}\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Gamma} k_{AW} T_{33\text{T}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{\Gamma} k_{AW} k_{33\text{ш}} k_{\text{д}} + 1} = \\
&= \frac{k_{\Sigma} T_{\text{м}} p / R_{\text{я}\Sigma}}{T_{\Gamma} T_{\text{я}\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Gamma} k_{AW} T_{33\text{T}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{\Sigma} k_{33\text{ш}} k_{\text{д}} + 1}. \quad (2.61)
\end{aligned}$$

Отримаємо передавальну функцію для швидкості по збурюючій дії $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$, для чого проведемо перетворення алгоритмічної схеми рис.2.8 так, як показано на рис. 2.11 і рис.2.12, вважаючи $U_3 = 0$.

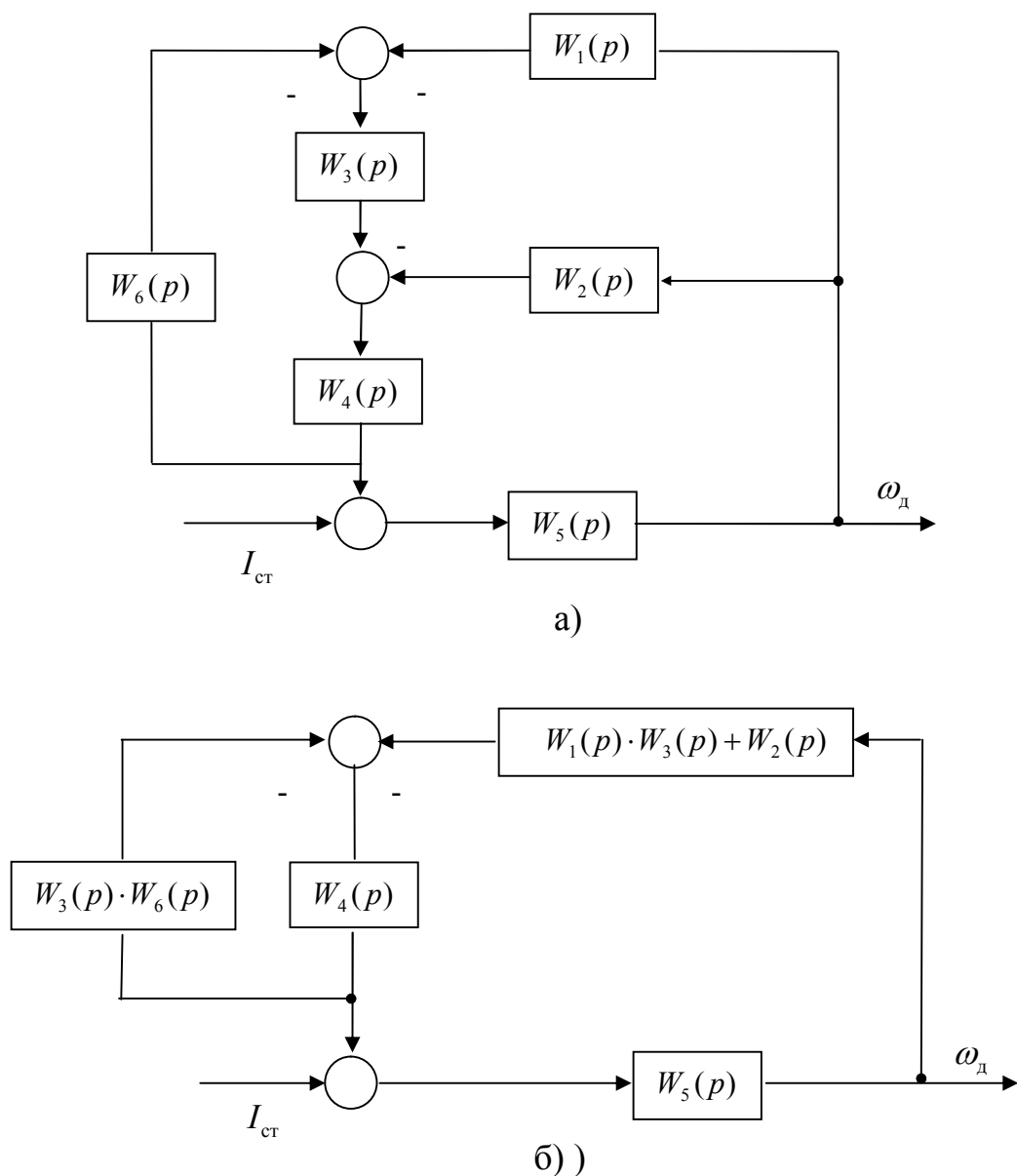
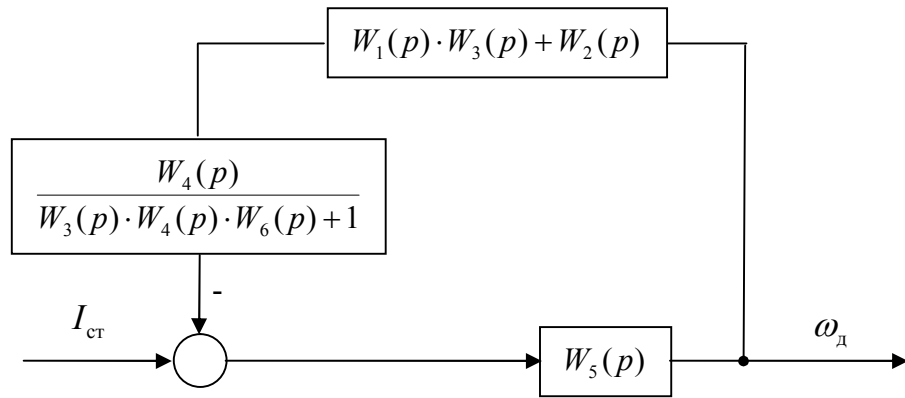


Рисунок 2.11 – Перетворення алгоритмічної схеми рис. 2.9 а

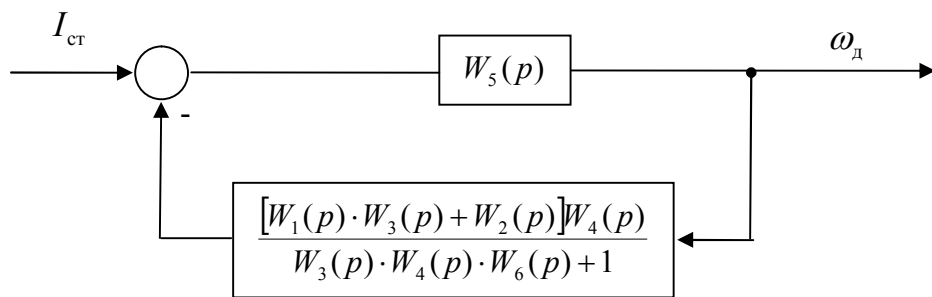
для отримання передавальної функції $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$:

а) перший етап перетворення схеми;

б) другий етап перетворення схеми



а)



б)

Рисунок 2.12 – Перетворення алгоритмічної схеми рис. 2.11,б

для отримання передавальної функції $W^3(p) = \frac{\omega_d(p)}{I_{ст}(p)}$:

а) перший етап перетворення схеми;

б) другий етап перетворення схеми

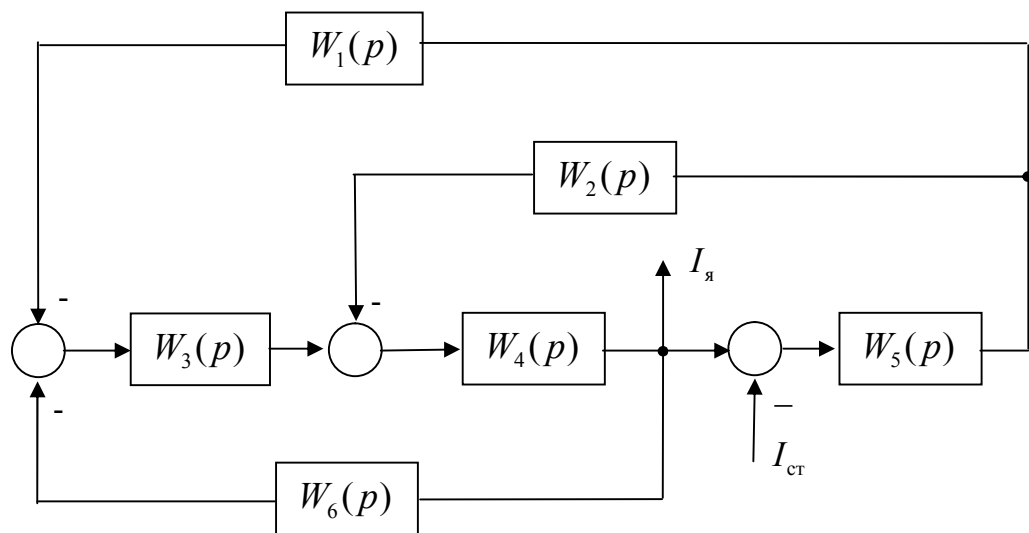
$$W_5(p) = \frac{R_{я\Sigma}}{cT_{мп}}; \quad (2.62)$$

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

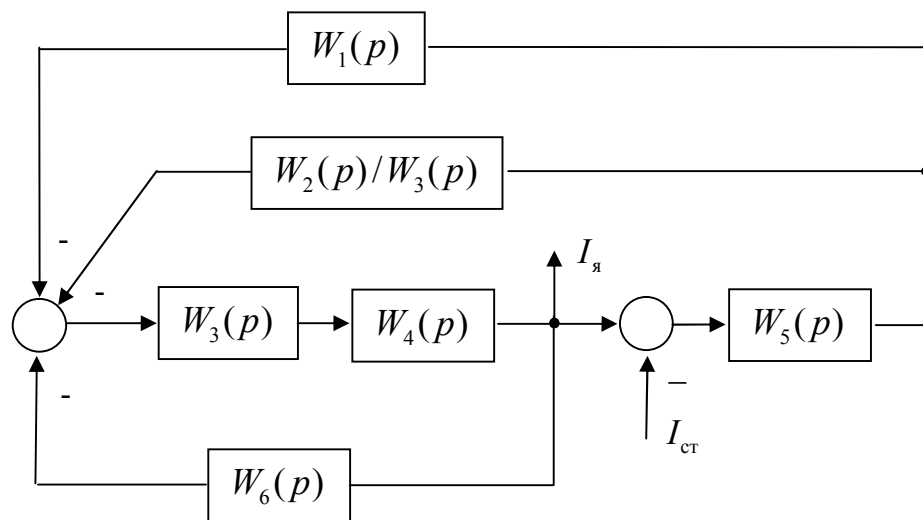
$$\begin{aligned}
&= \frac{1 + W_3(p)W_4(p)W_6(p)}{[W_1(p)W_3(p) + W_2(p)]W_4(p) + \frac{1 + W_3(p)W_4(p)W_6(p)}{W_5(p)}} = \\
&= \frac{1 + \frac{k_\Sigma}{R_{я\Sigma}} \cdot T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1)} = \\
&= \frac{\left(\frac{k_{33Ш} k_\Sigma}{T_\Gamma p + 1} + c \right) \frac{1}{T_{я\Sigma} p + 1} + \frac{\left(1 + \frac{k_\Sigma}{R_{я\Sigma}} \frac{T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1)} \right) c T_\Gamma p}{R_{я\Sigma}}}{\frac{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1) + k_\Sigma T_{33Т} p}{(T_\Gamma p + 1)(T_{я\Sigma} p + 1) R_{я\Sigma}}} = \\
&= \frac{k_\Sigma k_{33Ш} + c(T_\Gamma p + 1) + \left((T_\Gamma p + 1)(T_{я\Sigma} p + 1) + \frac{k_\Sigma}{R_{я\Sigma}} \cdot T_{33Т} p \right) c T_\Gamma p}{R_{я\Sigma} (T_\Gamma p + 1)(T_{я\Sigma} p + 1)} = \\
&= \frac{R_{я\Sigma} k_\Gamma \left[T_\Gamma T_{я\Sigma} p^2 + \left(T_\Gamma + T_{я\Sigma} + \frac{k_\Sigma T_{33Т}}{R_{я\Sigma}} \right) p + 1 \right]}{T_\Gamma T_{я\Sigma} T_\Gamma p^3 + T_\Gamma \left(T_\Gamma + T_{я\Sigma} + \frac{k_\Gamma k_{AW} T_{33Т}}{R_{я\Sigma}} \right) p^2 + (T_\Gamma + T_\Gamma) p + k_\Sigma k_{33Ш} k_\Gamma + 1}. \quad (2.63)
\end{aligned}$$

Визначимо передавальну функцію для струму по збурюючій дії

$W^4(p) = \frac{I_\Gamma(p)}{I_{ст}(p)}$. Порядок перетворення схеми показаний на рис.2.13 і рис.2.14.



а)



б)

Рисунок 2.13 – Перетворення алгоритмічної схеми рис. 2.8

для отримання передавальної функції $W^4(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$:

а) перший етап перетворення схеми;

б) другий етап перетворення схеми.

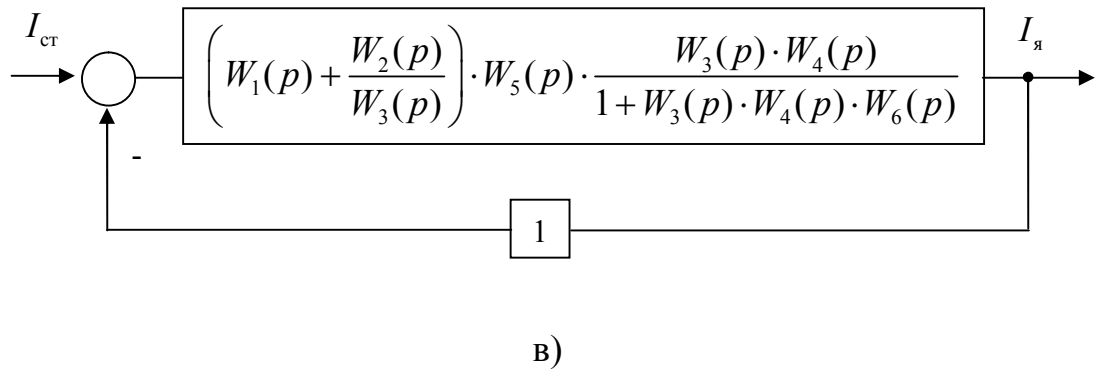
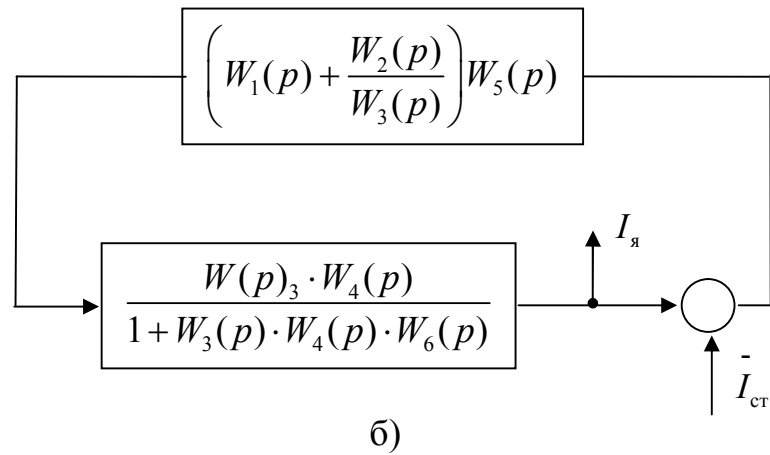
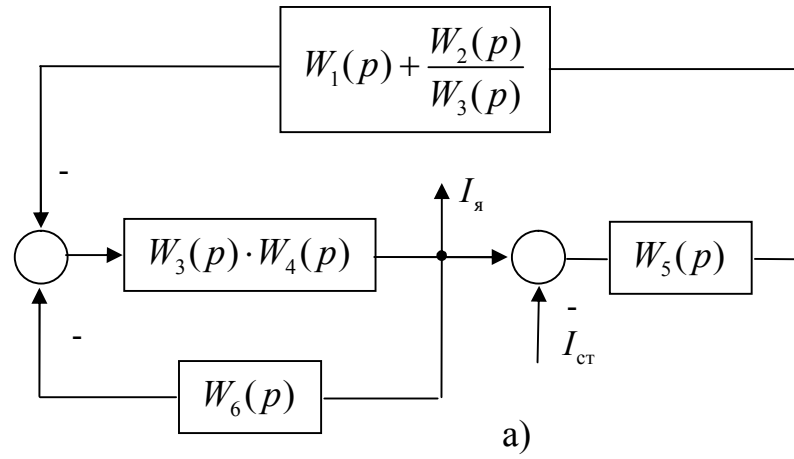


Рисунок 2.14 – Перетворення алгоритмічної схеми рис. 2.13,б

для отримання передавальної функції $W^4(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$:

а) перший етап перетворення схеми;

б) другий етап перетворення схеми.

в) третій етап преобразования схемы;

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

$$\frac{T_{\Gamma} p + k_{\Sigma} k_{\text{ззш}} k_{\text{д}} + 1}{T_{\Gamma} T_{\text{я}\Sigma} T_{\text{м}} p^3 + T_{\text{м}} \left(T_{\Gamma} + T_{\text{я}\Sigma} + \frac{k_{\Gamma} k_{\text{AW}} T_{\text{ззт}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\Gamma} + T_{\text{м}}) p + k_{\Sigma} k_{\text{ззш}} k_{\text{д}} + 1}. \quad (2.64)$$

Визначимо значення параметрів системи.

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

$$T_{\Gamma} = T_{\text{зб}\Gamma} = \frac{4 \cdot 1140 \cdot 0,29 \cdot 10^{-2}}{21,5} = 0,5 \text{ с}.$$

Електромагнітна постійна часу якірного кола, с

$$T_{\text{я}\Sigma} = \frac{L_{\text{я}\Sigma}}{R_{\text{я}\Sigma}}, \quad (2.64)$$

де $L_{\text{я}\Sigma}$ – сумарна індуктивність якірного кола

$$L_{\text{я}\Sigma} = L_{\text{яд}} + L_{\text{я}\Gamma}, \quad (2.65)$$

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

$$L_{\text{яд}(\Gamma)} = k_{\text{к}} \frac{U_{\text{нд}(\Gamma)}}{p_{\text{д}(\Gamma)} \omega_{\text{нд}(\Gamma)} I_{\text{нд}(\Gamma)}}, \quad (2.66)$$

$k_{\text{к}}$ – коефіцієнт, рівний $k_{\text{к}} = 0,25$ для компенсованої машини і $k_{\text{к}} = 0,6$ для некомпенсованої;

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

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

Номінальна швидкість генератора:

$$\omega_{\text{н}\Gamma} = \frac{\pi n_{\text{н}\Gamma}}{30}, \quad (2.67)$$

$$\omega_{\text{н}\Gamma} = \frac{1450 \cdot 3,14}{30} = 151,77 \text{ c}^{-1}$$

$$L_{\text{яд}} = 0,6 \frac{220}{2 \cdot 58,64 \cdot 338} = 3,33 \cdot 10^{-3} \text{ Гн}$$

$$L_{\text{я}\Gamma} = 0,1 \cdot \frac{0,6 \cdot 460}{2 \cdot 151,77 \cdot 315} = 2,9 \cdot 10^{-3} \text{ Гн}$$

$$L_{\text{я}\Sigma} = (2 \cdot 3,33 + 2,9) \cdot 10^{-3} = 9,16 \cdot 10^{-3} \text{ Гн}$$

$$T_{\text{я}\Sigma} = \frac{9,16 \cdot 10^{-3}}{0,11} = 0,084 \text{ с}$$

Електромеханічна постійна часу електроприводу

$$T_{\text{м}} = \frac{J_{\Sigma} R_{\text{я}\Sigma}}{(2 \cdot c)^2}, \quad (2.68)$$

де J_{Σ} – сумарний момент інерції двигуна і приведений до його валу момент інерції механізму.

$$J_{\Sigma} = 2J_{\text{д}} + J_{\text{пл}} / i_{\text{п}}^2; \quad (2.69)$$

$$J_{\Sigma} = 2J_{\text{д}} + J_{\text{пл}} / i_{\text{п}}^2 = 2 \cdot 8,25 + 17 \cdot 10^7 / 2022,1^2 = 16,5 + 41,5 = 58 \text{ кг} \cdot \text{м}^2$$

$$T_{\text{м}} = \frac{58 \cdot 0,11}{6,82^2} = 0,102 \text{ с}$$

Постійну часу зворотного зв'язку по току $T_{\text{ззт}}$ можна визначити таким чином: використовуючи критерій Гурвіца, знайти мінімальне значення $T_{\text{ззт min}}$, при якому система стійка. Потім провести розрахунок перехідних характеристик для значень $T_{\text{ззт}} \geq T_{\text{ззт min}}$ і вибрати те значення, при якому перехідний процес оптимальний. Якщо система стійка при будь-якому значенні $T_{\text{ззт}}$, то слід задатися рядом значень $T_{\text{ззт}}$ і вибрати оптимальне значення. У роботі величину $T_{\text{ззт}}$ визначимо шляхом моделювання системи на ЕОМ.

По передавальних функціях $W^1(p) \div W^4(p)$ розраховуємо перехідні характеристики системи. Розрахунок перехідних процесів виконуємо з використанням системи MATLAB. Графіки перехідних процесів представлені на рис.2.16 і рис.2.17. Ті ж результати були отримані шляхом дослідження перехідних процесів за допомогою моделі рис.2.15. Розрахунок проводився для різних значень постійної часу $T_{\text{ззт}}$. На рис.2.16 і рис. 2.17 представлені графіки перехідних процесів для трьох значень $T_{\text{ззт}}$, а саме: 1 – $T_{\text{ззт}} = 0,01 \text{ с}$; 2 – $T_{\text{ззт}} = 0,05 \text{ с}$; 3 – $T_{\text{ззт}} = 0,1 \text{ с}$. Як видно з рисунка, оптимальних значень показників якості перехідних процесів набути при $T_{\text{ззт}} = 0,01 \text{ с}$.

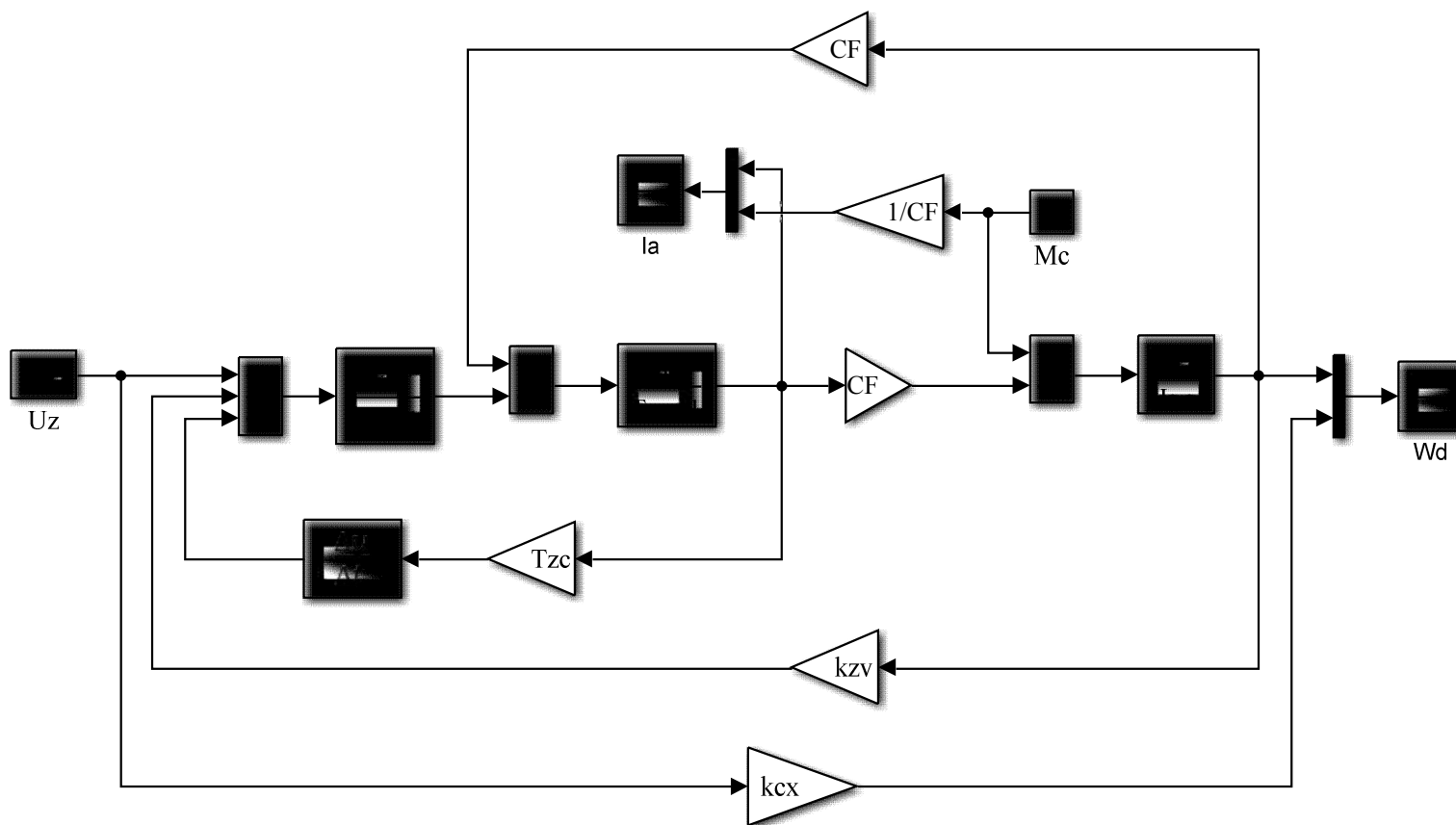


Рисунок 2.15 – Схема моделі системи, розроблена на ЕОМ

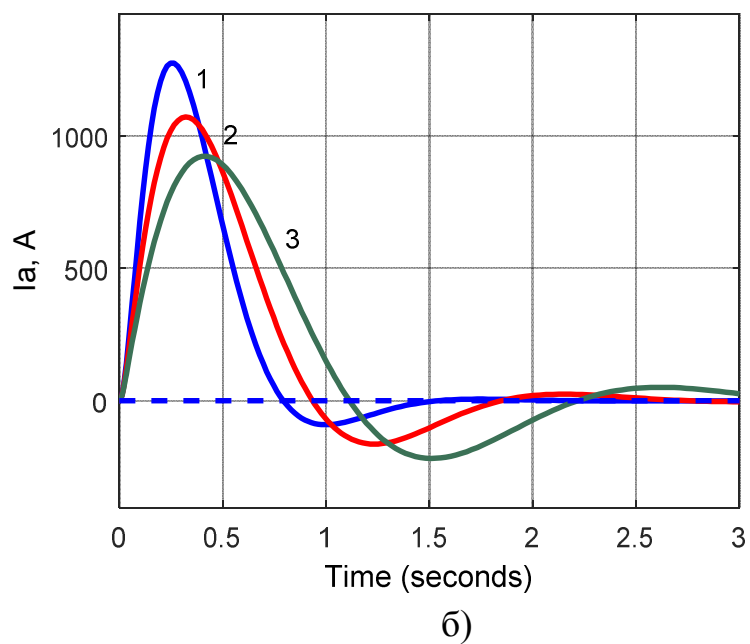
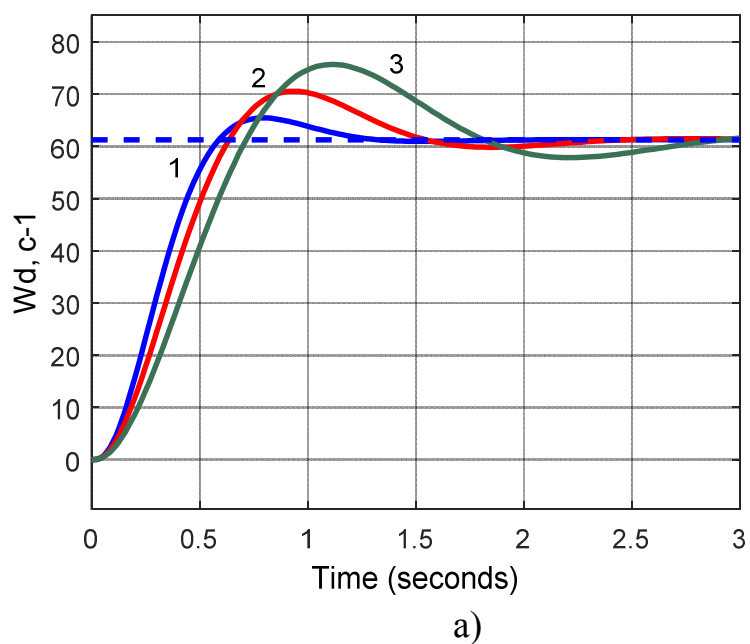


Рисунок 2.16 – Графіки перехідних процесів швидкості а) і струму б)
по залаючій дії

$$1 - T_{33T} = 0,01 \text{ c}; \quad 2 - T_{33T} = 0,05 \text{ c}; \quad 3 - T_{33T} = 0,1 \text{ c}$$

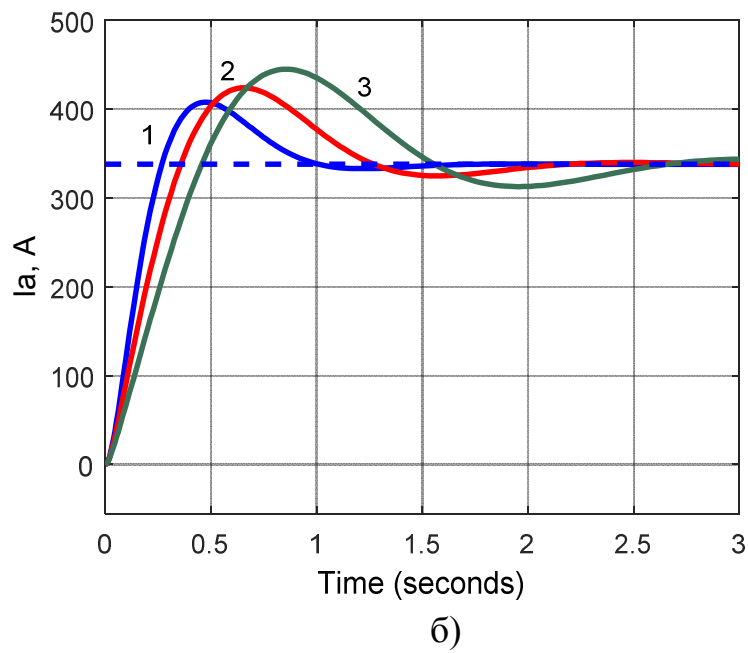
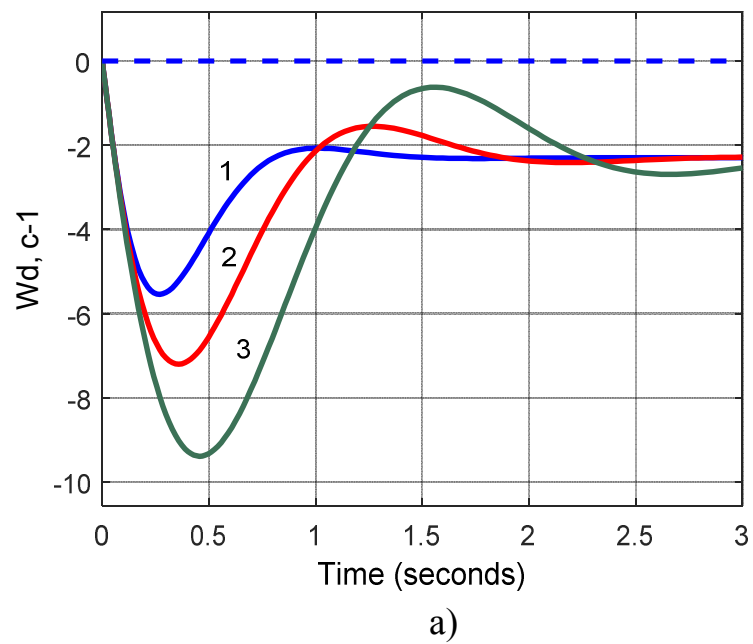


Рисунок 2.17– Графіки перехідних процесів швидкості а) і струму б)
по збурюючій дії

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

ВИСНОВКИ ДО РОЗДІЛУ 2

1. Виконано вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу, проведена перевірка двигуна за умовами нагріву і перевантажувальної здатності, для чого розраховано швидкісну і навантажувальні діаграми. Отримане при розрахунку значення еквівалентного моменту показує, що по нагріву двигун відповідає умовам експлуатації. По перевантажувальній здатності двигун також задовольняє вимогам.

2. Виконано розрахунок параметрів силового ланцюга електроприводу, що використовуються при розрахунку системи автоматичного управління.

3. Проведено вибір структури системи автоматичного управління. До електроприводів більшості основних механізмів роторних екскаваторів пред'являються підвищені вимоги відносно жорсткості робочої частини механічної характеристики і обмеження максимального значення моменту, що розвивається.. Згідно рекомендацій [51] вибрано систему із зворотним зв'язком за швидкістю, гнучким зворотним зв'язком по струму і зворотним зв'язком по струму з відсіченням.

4. Проведений розрахунок статичних характеристик розімкненої і замкненої системи. Складено алгоритмічну схему системи та визначені передавальні функції системи по задаючій і збурюючій діях.

5. Проведено моделювання синтезованої системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблена схема моделі системи, що дозволяє виконати дослідження динамічних характеристик системи. Шляхом моделювання визначено значення постійної часу зворотного зв'язку по струму, при якій перехідні процеси змінних стану системи мають задовільні показники якості.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

Для електроприводу механізму повороту стріли характерною є наявність пружно пов'язаних з платформою інерційних мас стріли ротора з роторним колесом і консолі противаги. Це призводить до того, що перехідні процеси мають значні коливання.

У попередніх розділах було проведено дослідження динамічних характеристик системи електроприводу механізму повороту стріли в припущенні нескінченної жорсткості всіх зв'язків між елементами механічної частини системи. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер руху елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Із-за кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, прикладення до якої управляючої дії (моменту двигуна), або збурюючої дії (моменту статичного навантаження) викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і відробітки необхідних рухів і переміщень, що ускладнюють точність.

Кінематична схема механізму повороту стріли є складною розгалуженою механічною системою, що складається з великого числа зосереджених і розподілених мас, сполучених пружними зв'язками. При інженерних розрахунках такі системи зазвичай зводяться до трьохмасових або двомасових систем із зосередженими масами і гнучкими невагомими зв'язками між ними.

Для електроприводу механізму повороту стріли використовуються два двигуни. Така схема багатодвигунного електроприводу в загальному випадку приводиться до розгалуженої розрахункової схеми, показаної на рис.3.1. На схемі позначено: $J_{д1}$ і $J_{д2}$ – моменти інерції двигунів; c_1 , c_2 – приведені жорст-

кості кожного приводу; $J_{\text{м}}$ – приведений момент інерції механізму; тобто приведений до валу двигуна момент інерції поворотної платформи $J_{\text{м}} = J'_{\text{пл}} = 20 \text{ кг} \cdot \text{м}^2$.

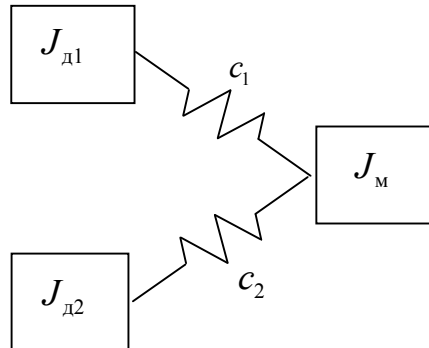


Рисунок 3.1 – Розгалужена розрахункова схема механічної частини електроприводу

Оскільки двигуни приводу рухаються синфазно, розгалужену схему рис.3.1 можна звести до двомасової розрахункової механічної схеми рис.3.2. На схемі позначено: $J_{\text{д}\Sigma}$ – сума моментів інерції двигунів; оскільки використовуються два однакові двигуни, то $J_{\text{д}\Sigma} = 2J_{\text{д}}$; c_{12} – приведена жорсткість зв'язку між масами; $c_{12} = c_1 + c_2$.

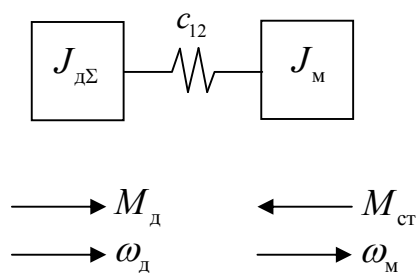


Рисунок 3.2 – Двомасова розрахункова схема

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

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

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

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

β – коефіцієнт внутрішнього в'язкого тертя каната між масами.

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

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

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

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

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

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

Запишемо систему (3.4) у формі Коші, враховуючи, що $I_{\text{ст}} = M_{\text{ст}}/c$,

$T_{\text{м}} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2} = J_{\text{д}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}$. Об'єднаємо системи (3.3) і (3.4) і проведемо деякі нескладні перетворення. Отримаємо наступну систему рівнянь стану двомасової системи:

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

Введемо вектор стану системи

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

і вектор зовнішніх дій

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

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

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

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

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

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

$$\mathbf{A} = \begin{array}{c} \begin{array}{ccccc} \omega_{\text{м}} & M_{\text{пр}} & \omega_{\text{д}} & I_{\text{я}} & E_{\text{г}} \end{array} \\ \left| \begin{array}{ccccc} -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 \\ -c_{12} & 0 & c_{12} & 0 & 0 \\ \frac{\beta}{J_{\text{м}}} & -\frac{1}{J_{\text{м}}} & -\frac{\beta}{J_{\text{м}}} & \frac{c}{J_{\text{м}}} & 0 \\ 0 & 0 & -\frac{c}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}} & -\frac{1}{T_{\text{я}\Sigma}} & \frac{1}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}} \\ 0 & 0 & \frac{ck_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\Gamma}} - \frac{k_{33\Pi}k_{\Sigma}}{T_{\Gamma}} & \frac{k_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}T_{\Gamma}} & \frac{-k_{\Sigma}T_{33\Gamma}}{T_{\text{я}\Sigma}R_{\text{я}\Sigma}T_{\Gamma}} - \frac{1}{T_{\Gamma}} \end{array} \right| \end{array}$$

$$\mathbf{B} = \begin{array}{c} \begin{array}{cc} U_3 & M_{\text{ст}} \end{array} \\ \left| \begin{array}{cc} 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{k_{\Sigma}}{T_{\Gamma}} & 0 \end{array} \right| \end{array}$$

Алгоритмічна схема двомасової системи показана на рис.3.3.

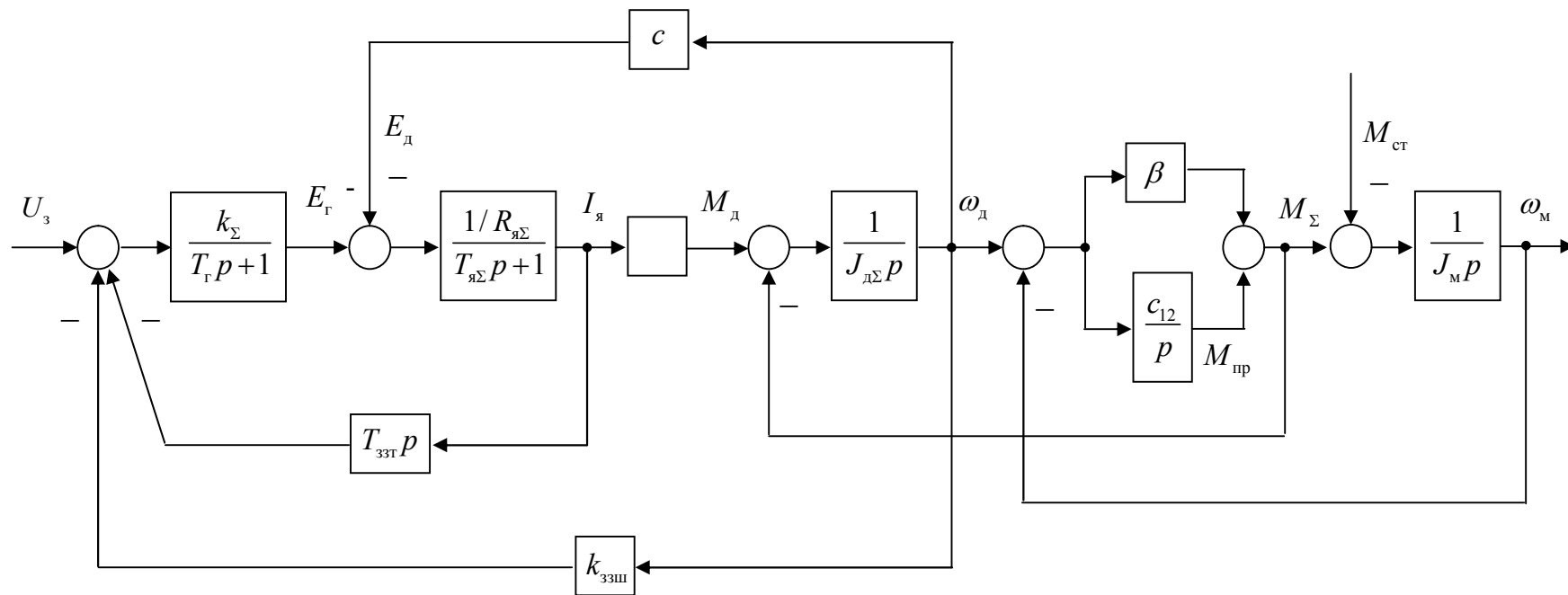


Рисунок 3.3- Алгоритмічна схема двомасової системи

3.2 Розрахунок перехідних процесів двомасової системи

Використовуючи математичну модель системи у вигляді рівнянь стану були проведені розрахунки перехідних процесів по задаючій і збурюючій діях. Текст файлу, що містить початкові дані, матриці **A**, **B**, **C** приведений в додатку Г. Дослідження проводилися також за допомогою схеми моделі системи, приведеної на рис. 3.4. Всі розрахунки виконувалися на ЕОМ за допомогою пакету прикладних програм MATLAB.

Графіки перехідних процесів показані на рисунках 3.5 – 3.8. На рисунках приведені перші чотири змінні стани системи. Як видно з графіків, мають місце значні коливання основних координат системи, що несприятливо позначається на роботі механізму, збільшуючи навантаження на елементи системи в перехідних режимах.

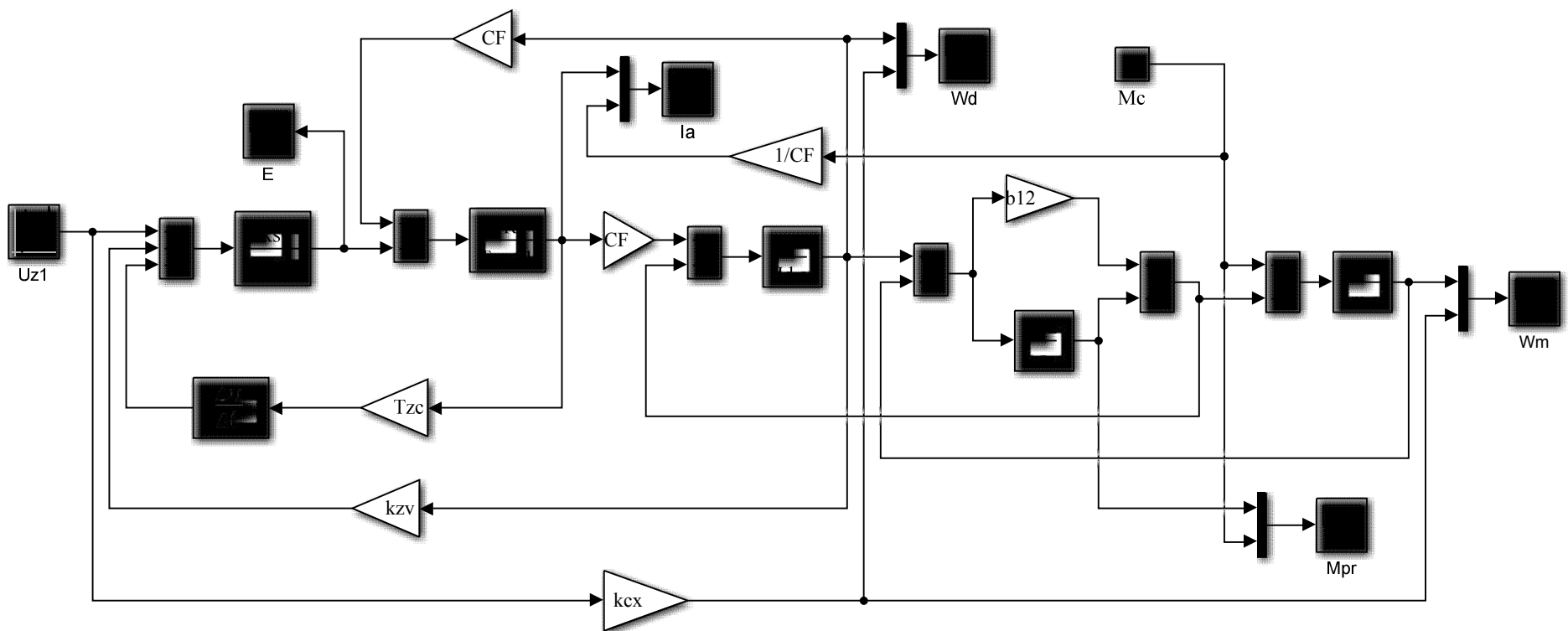
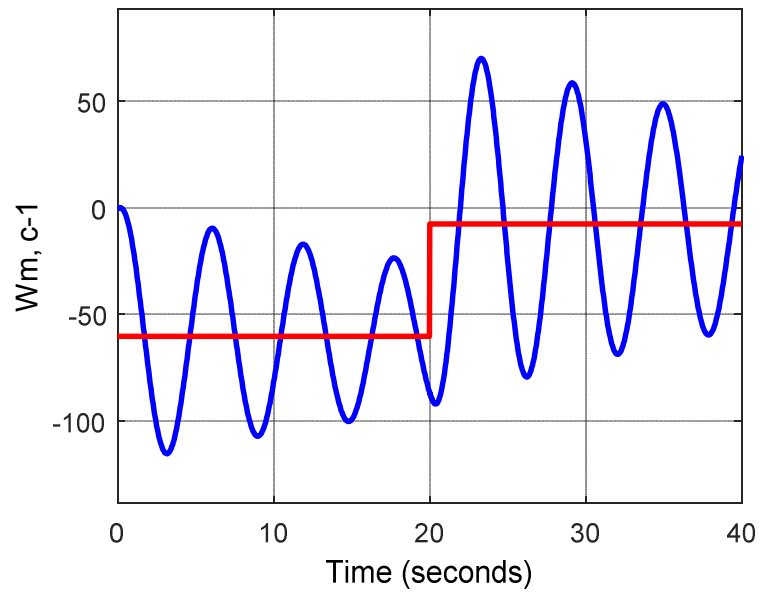
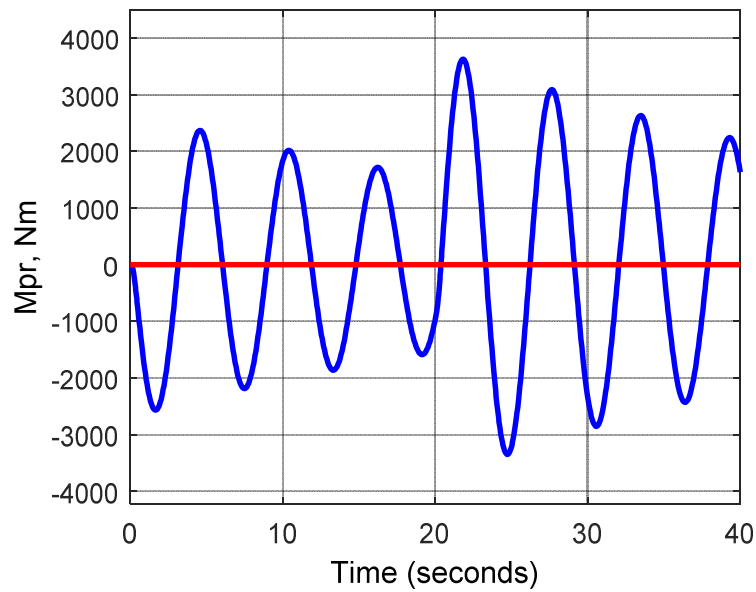


Рисунок 3.4 - Схема моделі двомасової системи, розроблена на ЕОМ

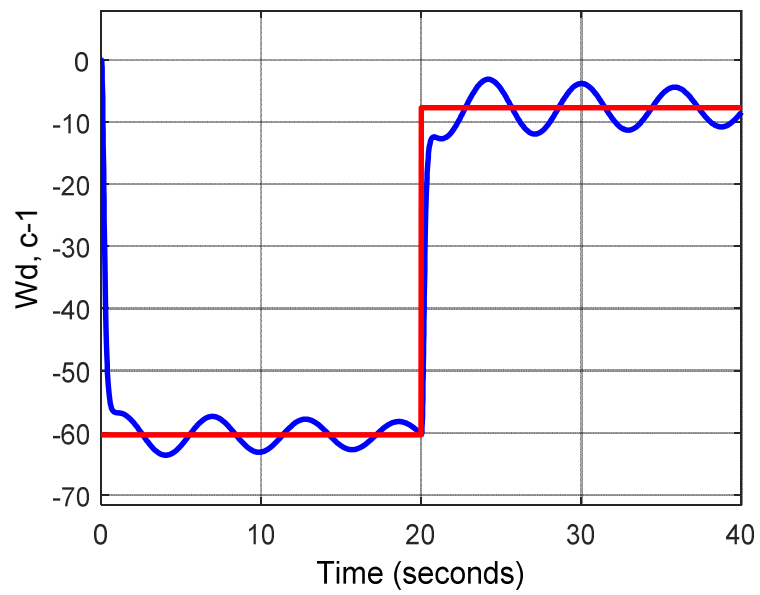


a)

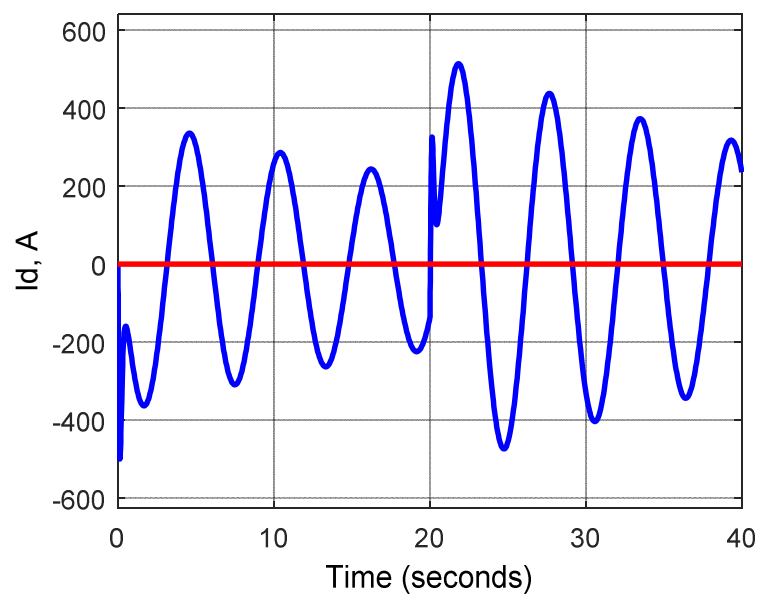


б)

Рисунок 3.5 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії



a)



б)

Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а)

і струму двигуна $I_d = f(t)$ (б) двомасової системи по задаючій дії

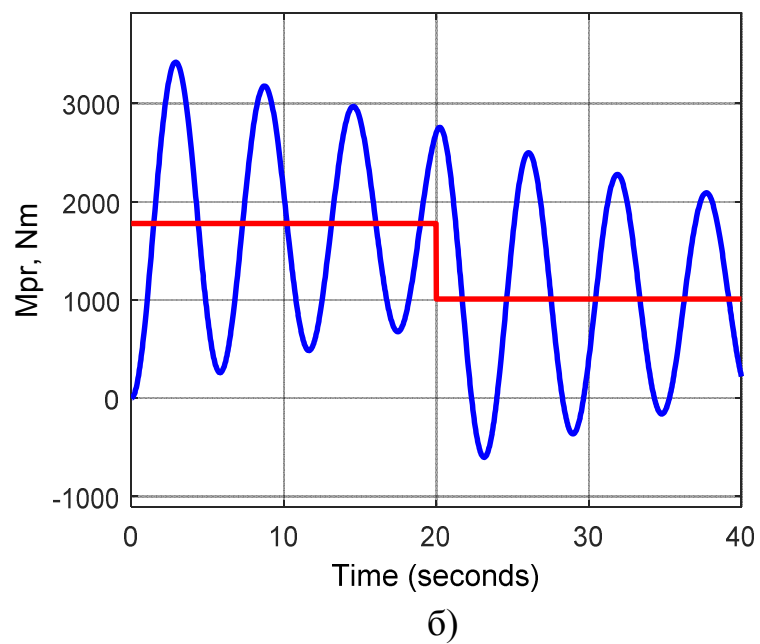
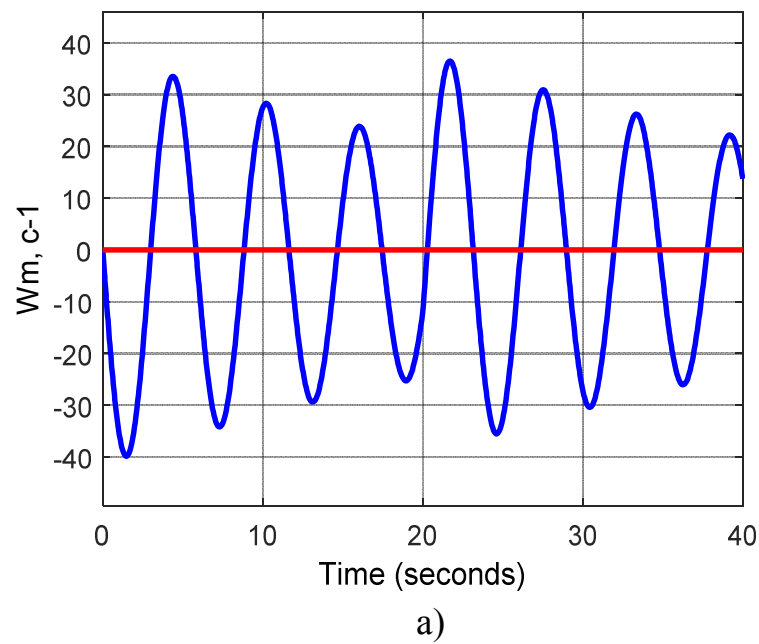
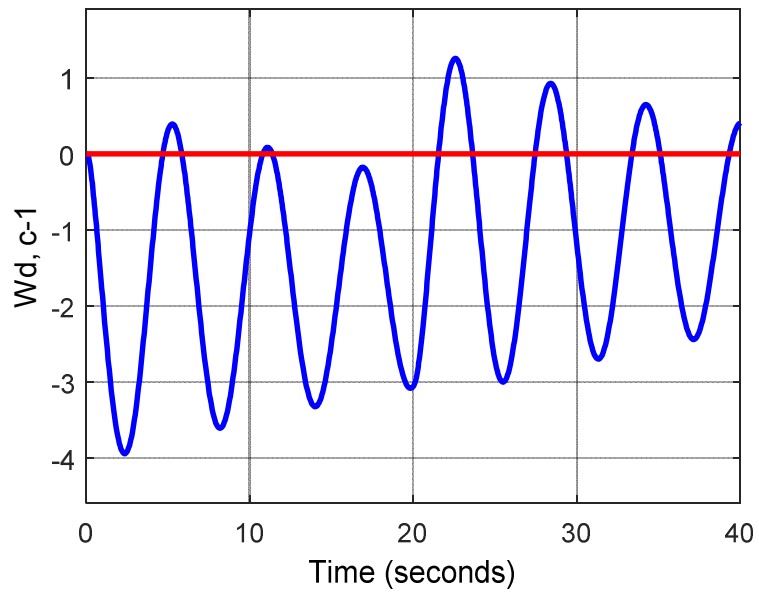
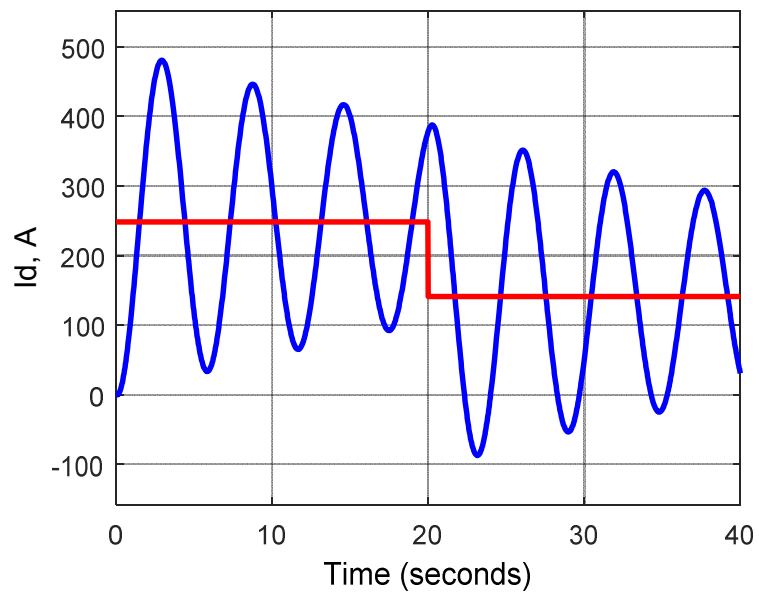


Рисунок 3.7 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по збурюючій дії



a)



б)

Рисунок 3.8 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму повороту стріли роторного екскаватора за наявності пружно пов'язаних з платформою інерційних мас стріли ротора з роторним колесом і консолі противаги. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер рухів елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Із-за кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, додаток до якої управляючої дії або збурюючої дії (моменту статичного навантаження), викликає коливання зв'язаних мас, що збільшують максимальні навантаження зв'язків і ускладнюють точність обробки необхідних рухів і переміщень. Показано, що кінематична схема механічної частини системи управління механізмом повороту стріли екскаватора з урахуванням податливості елементів механічної частини електроприводу може розглядатися як двомасова.

2. Виконано розробку математичної моделі. Отримано рівняння стану двомасової системи і рівняння стану двомасової системи у векторному – матричній формі, визначені елементи вектора стану, тобто змінні стану системи; визначений вектор зовнішніх дій, а також матриці стану **A** і управління **B**. Розроблена алгоритмічна схема двомасової системи і спрощена алгоритмічна схема двомасової системи з оптимізованим по модульному критерію контуром струму.

3. Проведено моделювання двомасової системи на ЕОМ з використанням пакету прокладних програм MATLAB. В режимі SIMULINK системи MATLAB розроблено схему моделі двомасової системи. Як тестовий сигнал на вхід системи подавався ступінчастий сигнал з випадковою амплітудою, який знаходиться в межах від +24В до -24В, що відповідає зміні швидкості в межах номінальних значень. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи мають незадовільний (коливальний) характер.

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

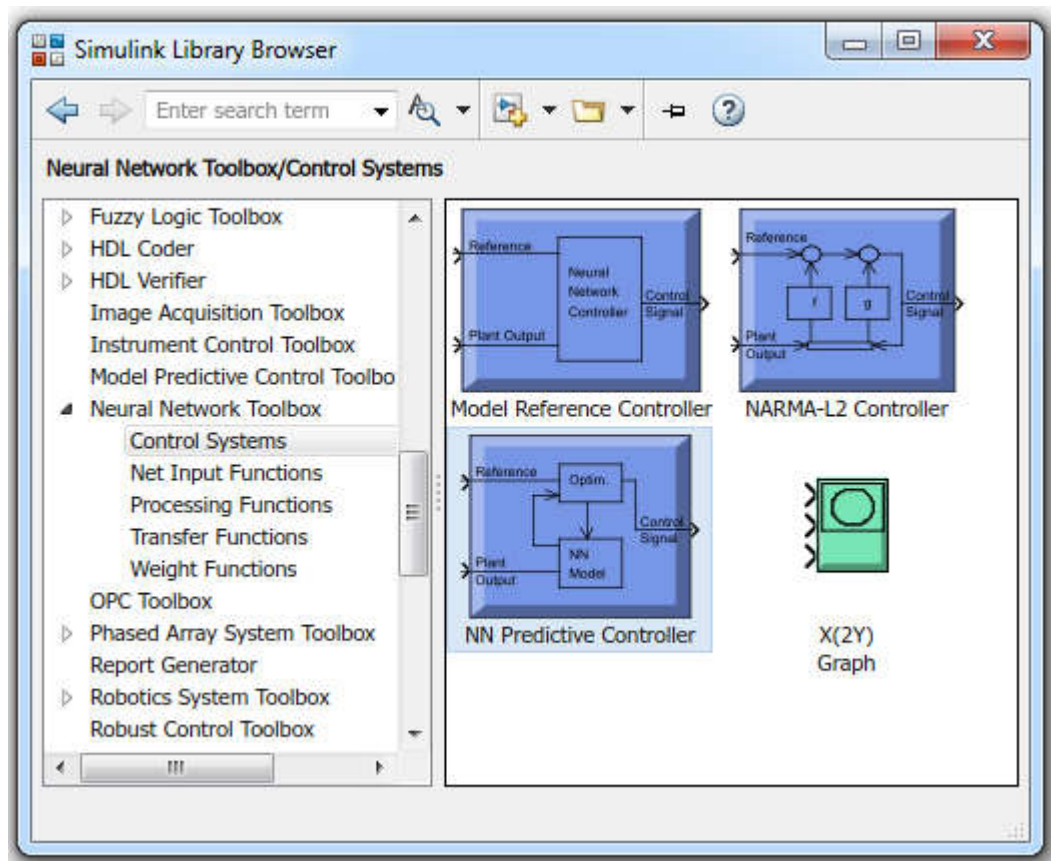


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

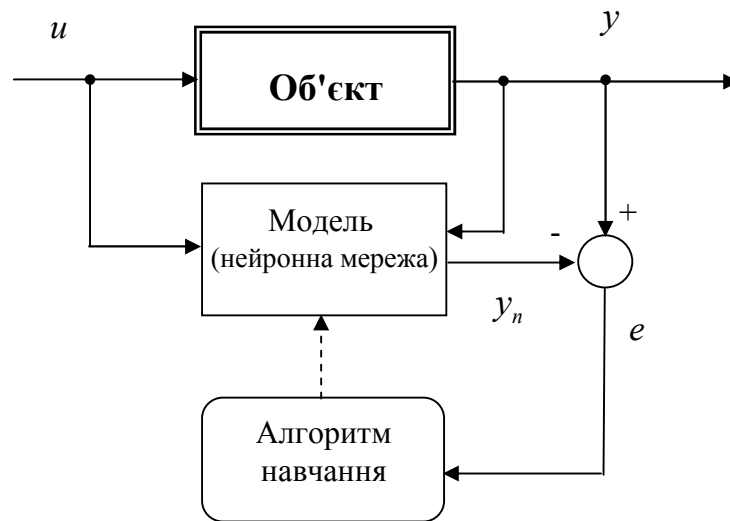


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

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

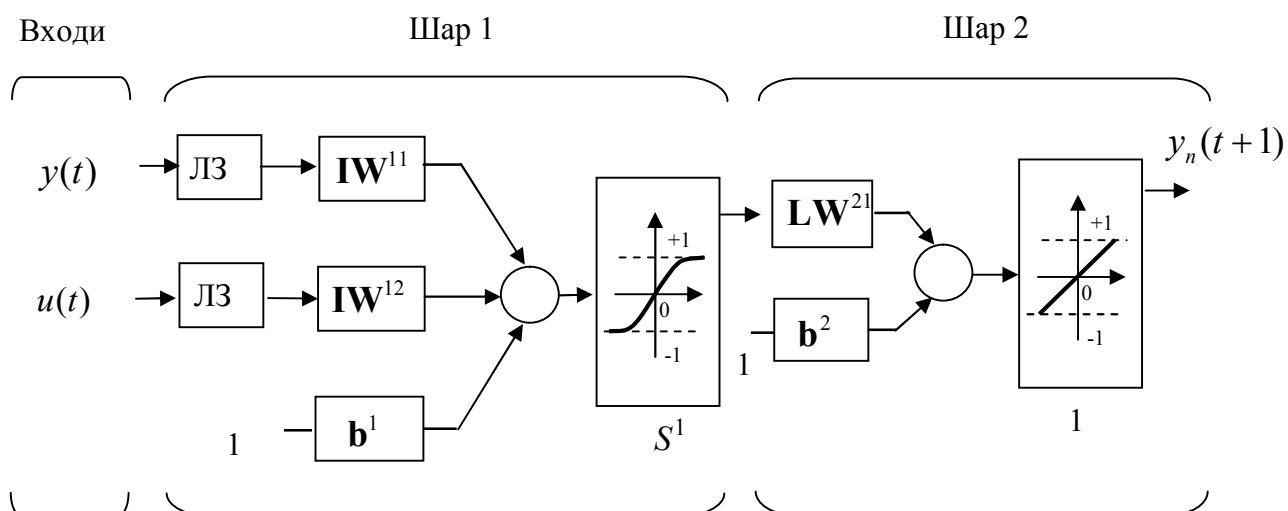


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

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

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

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

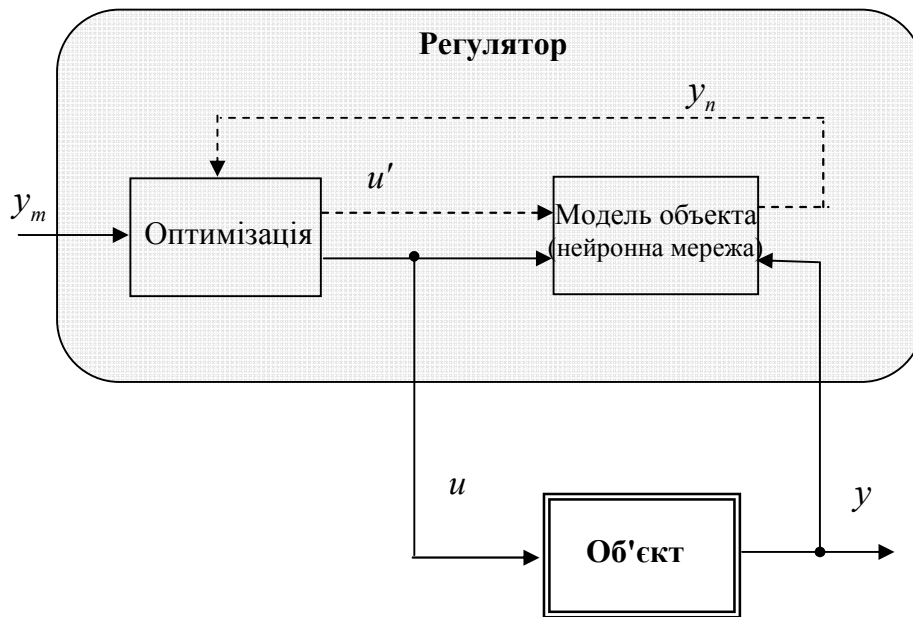


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

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

Допоміжні функції:

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

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

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

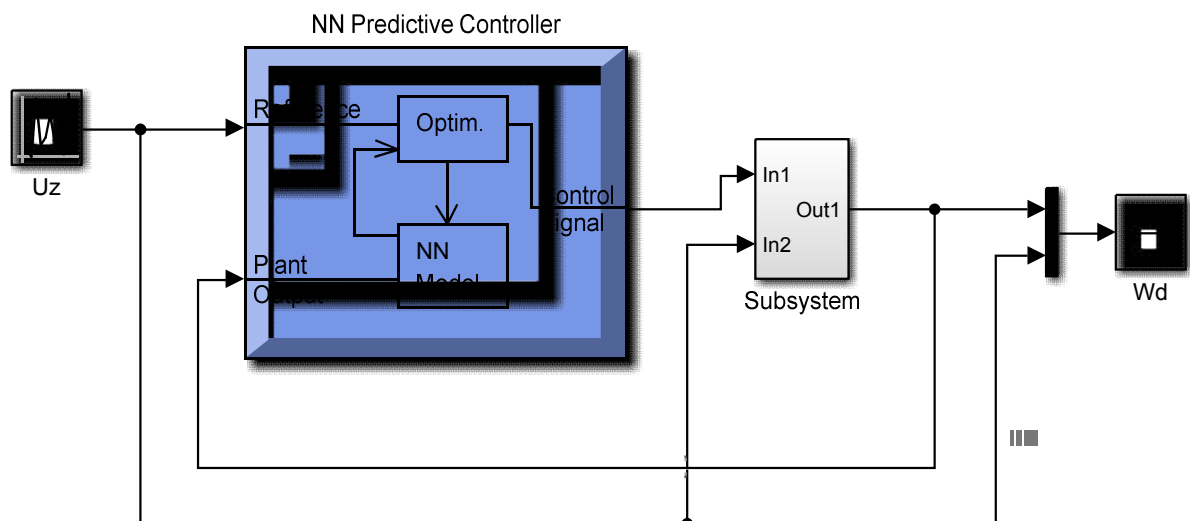
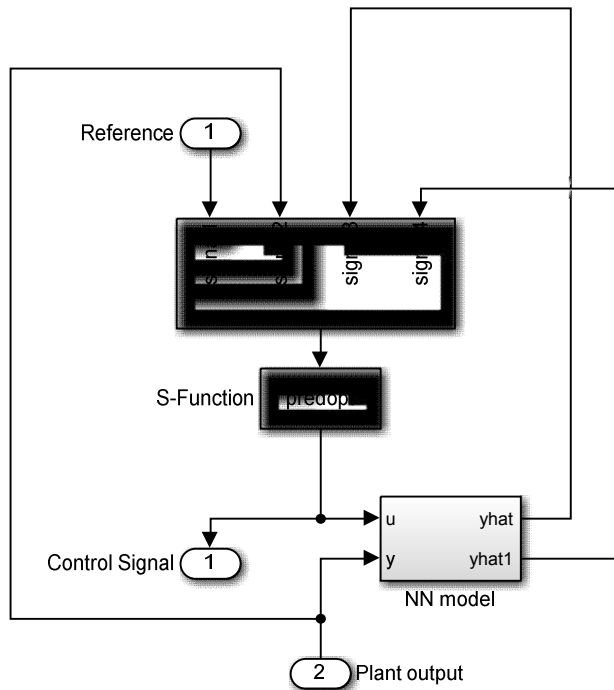


Рисунок 4.5 – Схема системи управління з нейрорегулятором

NN Predictive Controller

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

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



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

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

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

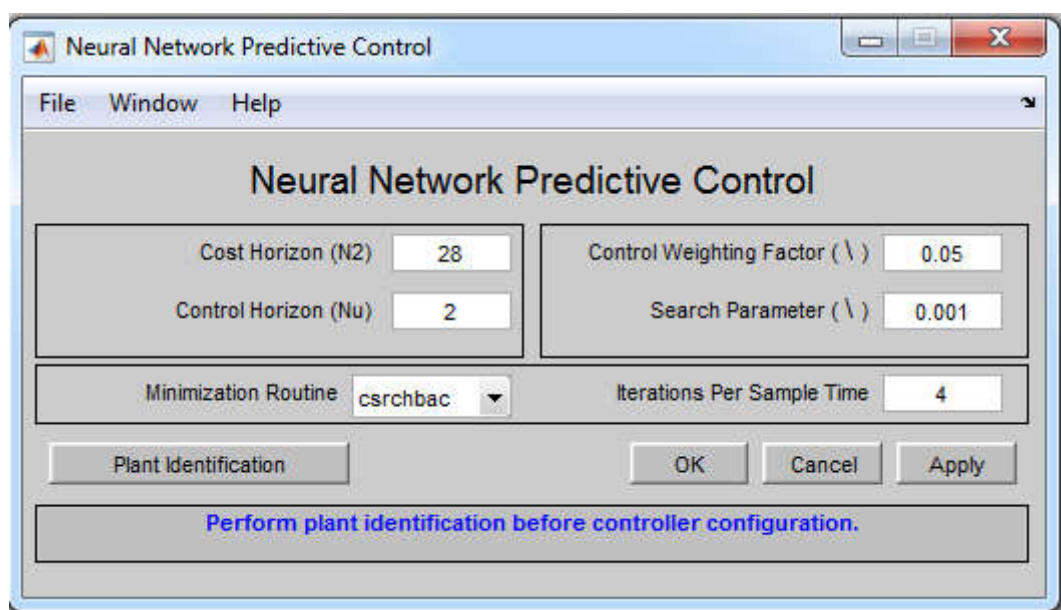


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

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

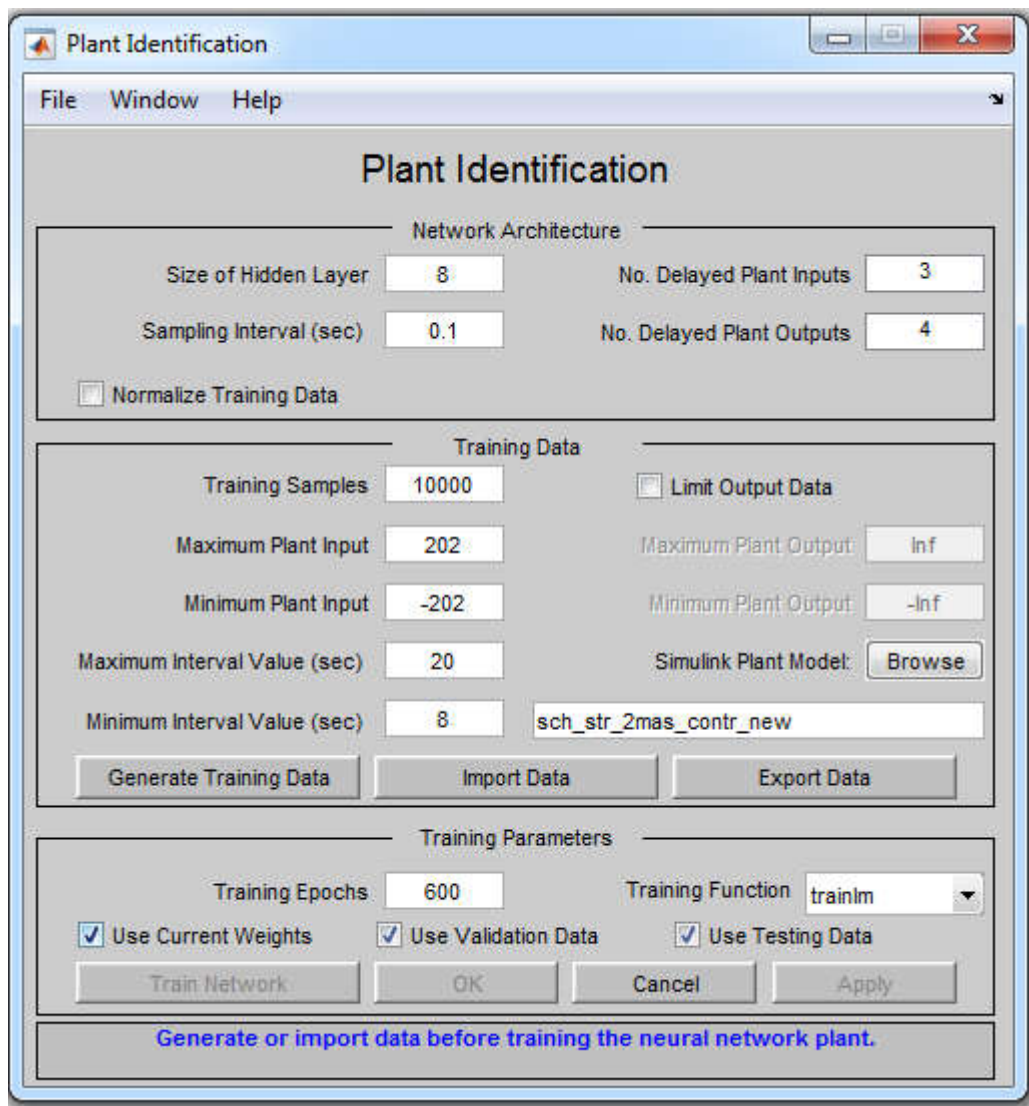


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

Опис вікна Plant Identification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

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

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

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

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

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

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

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

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

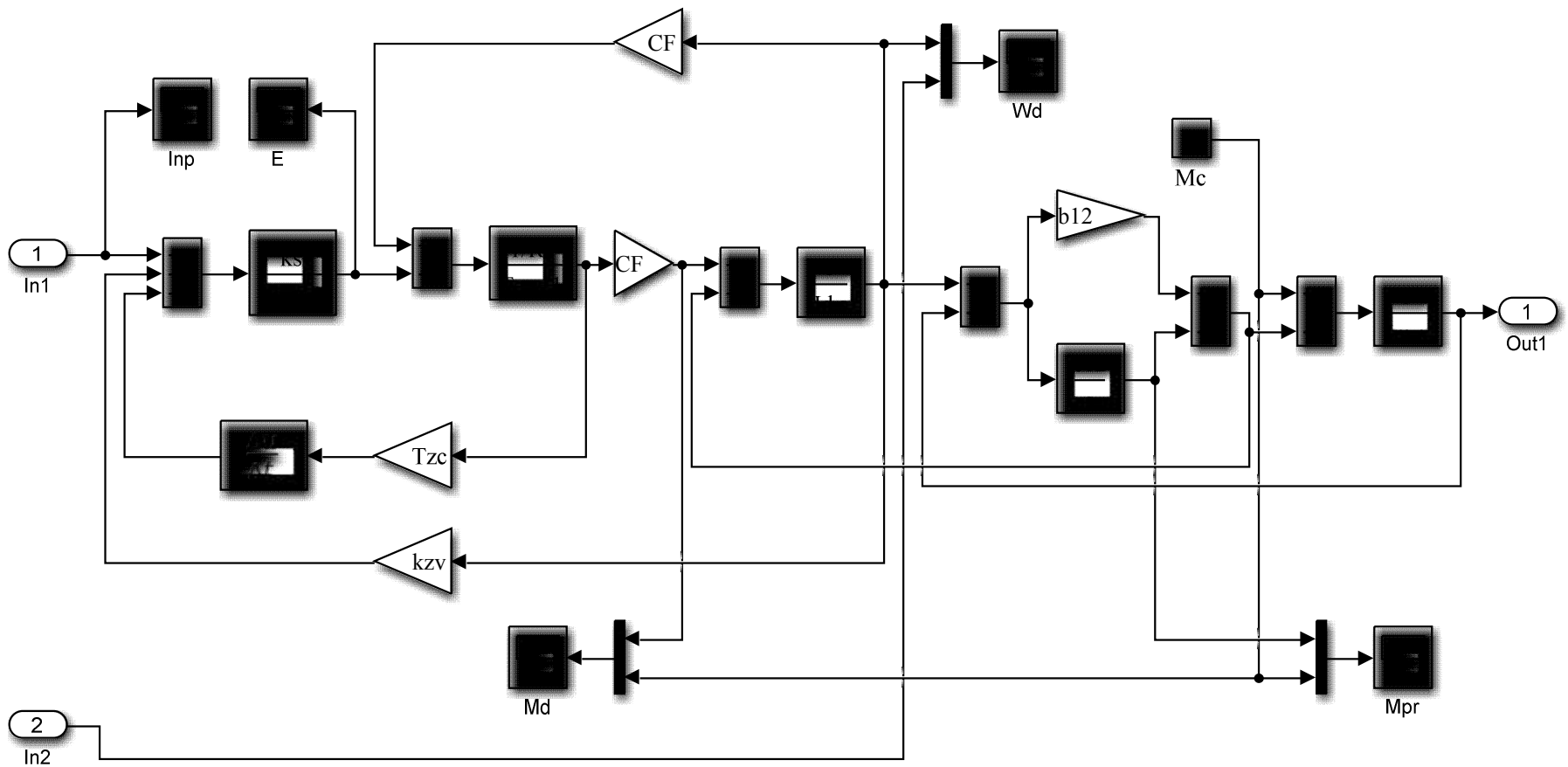


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

менших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

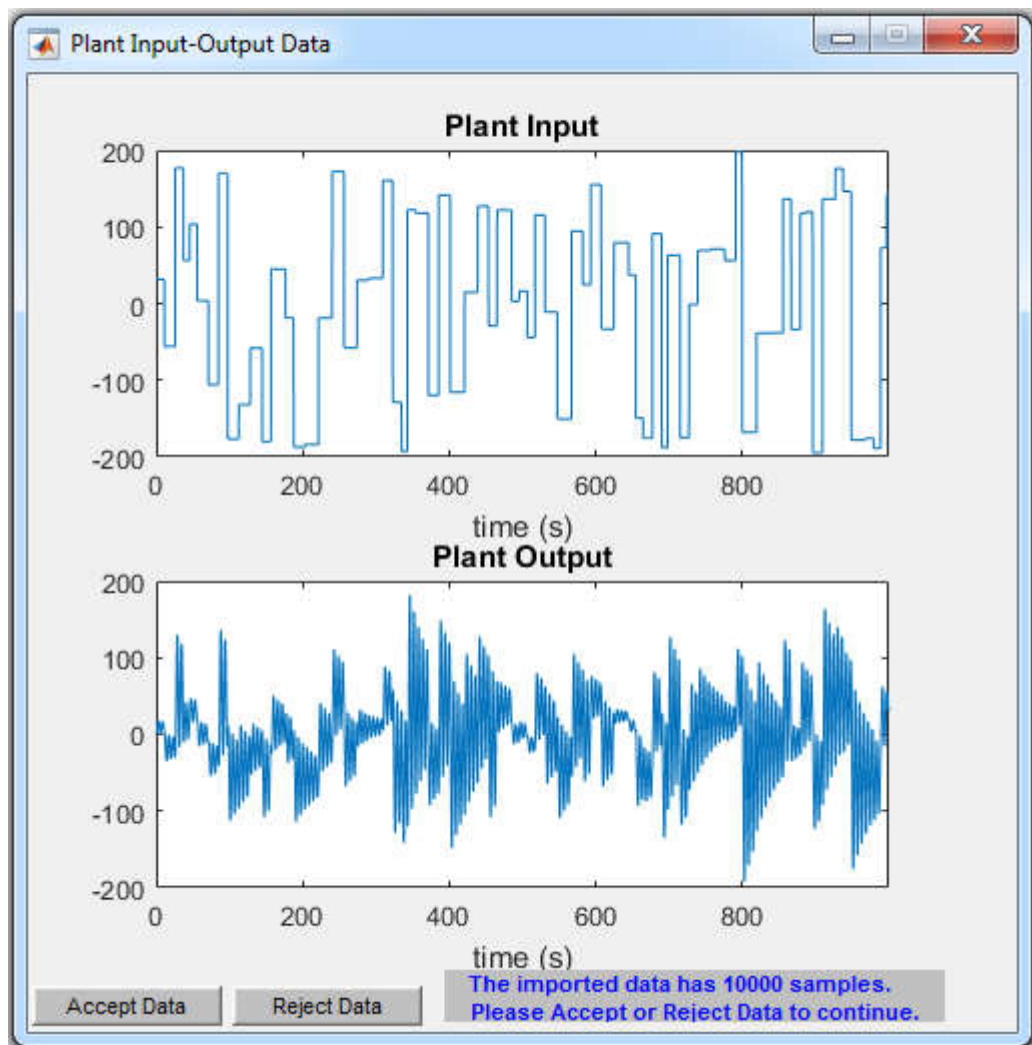


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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

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

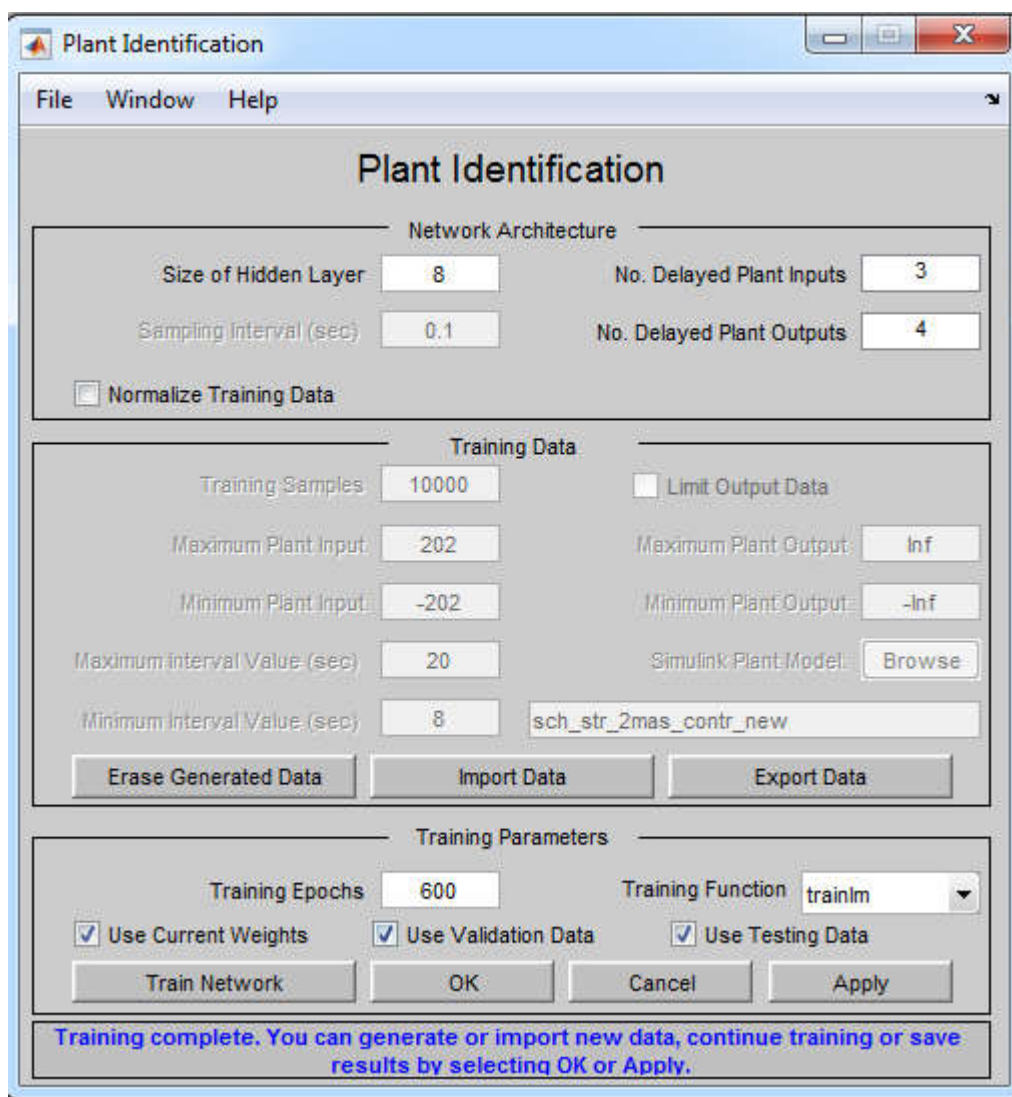


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

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

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

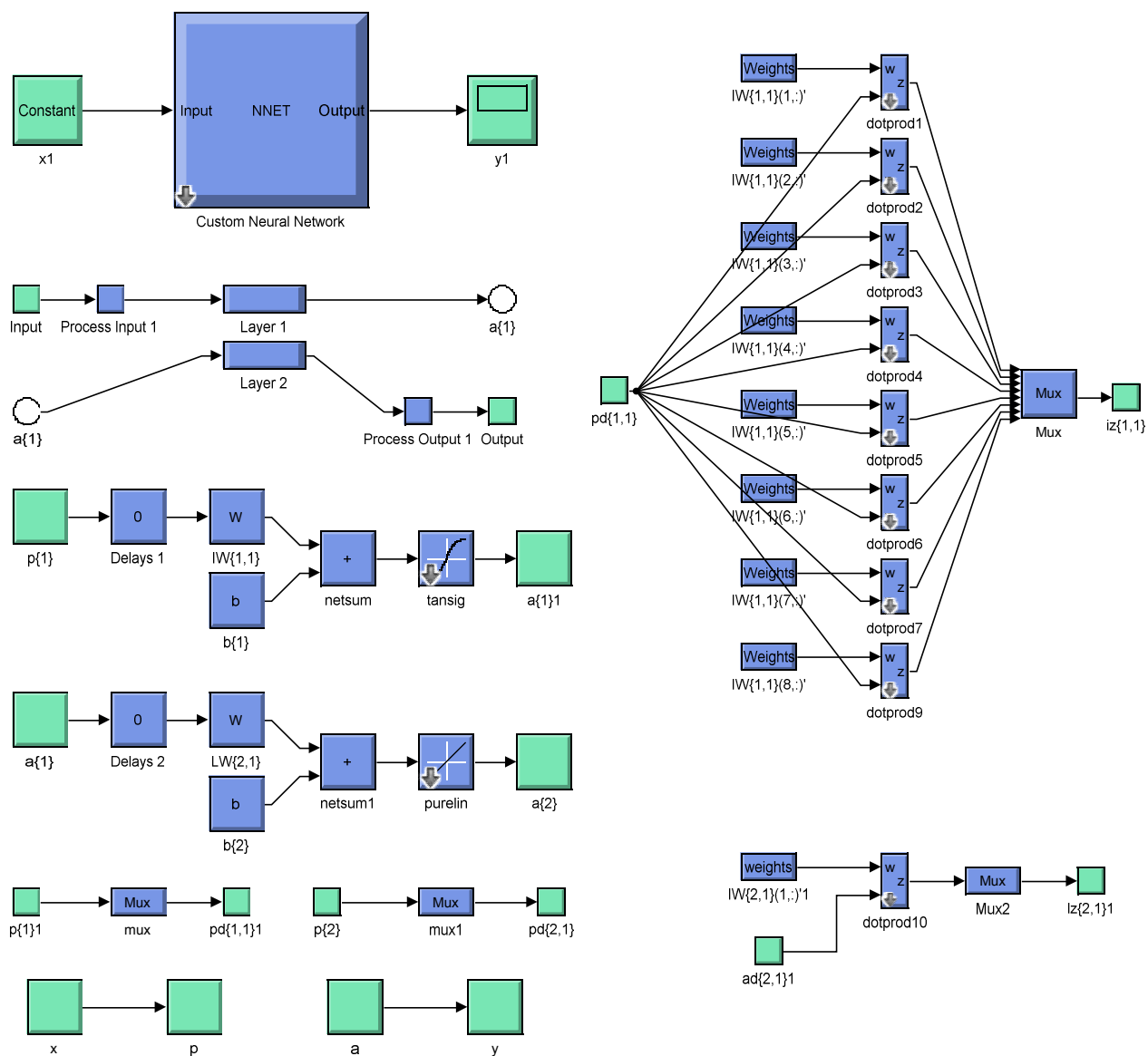


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

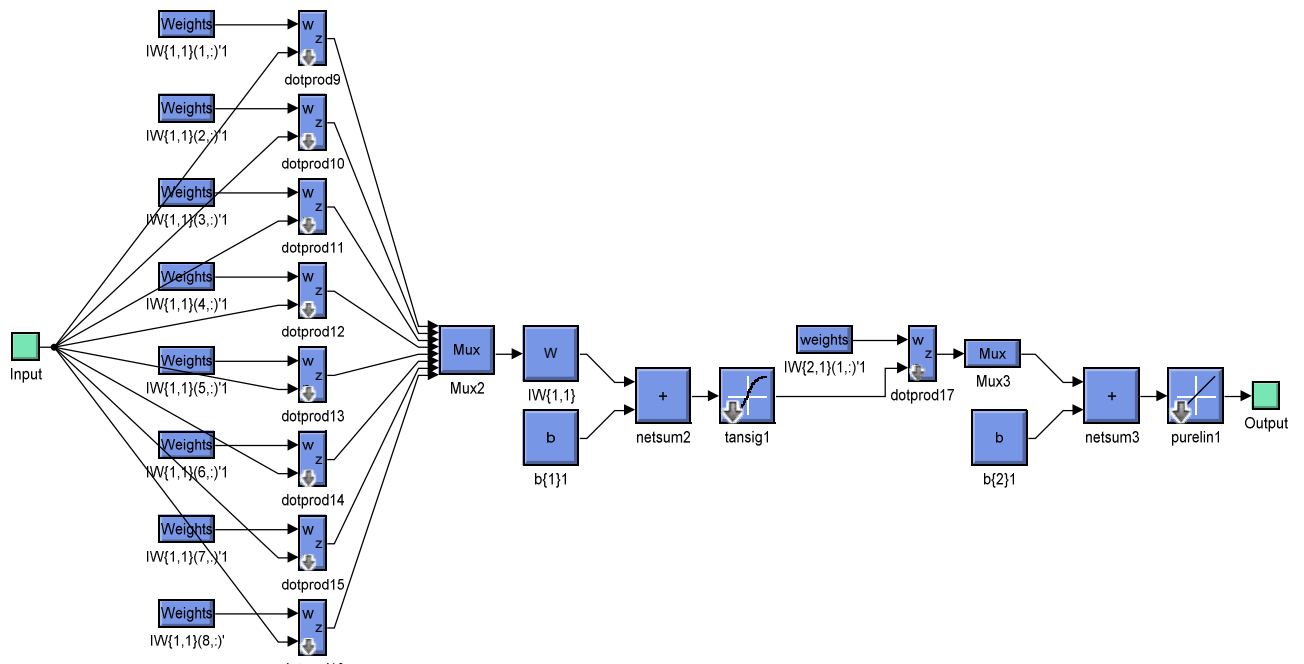


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

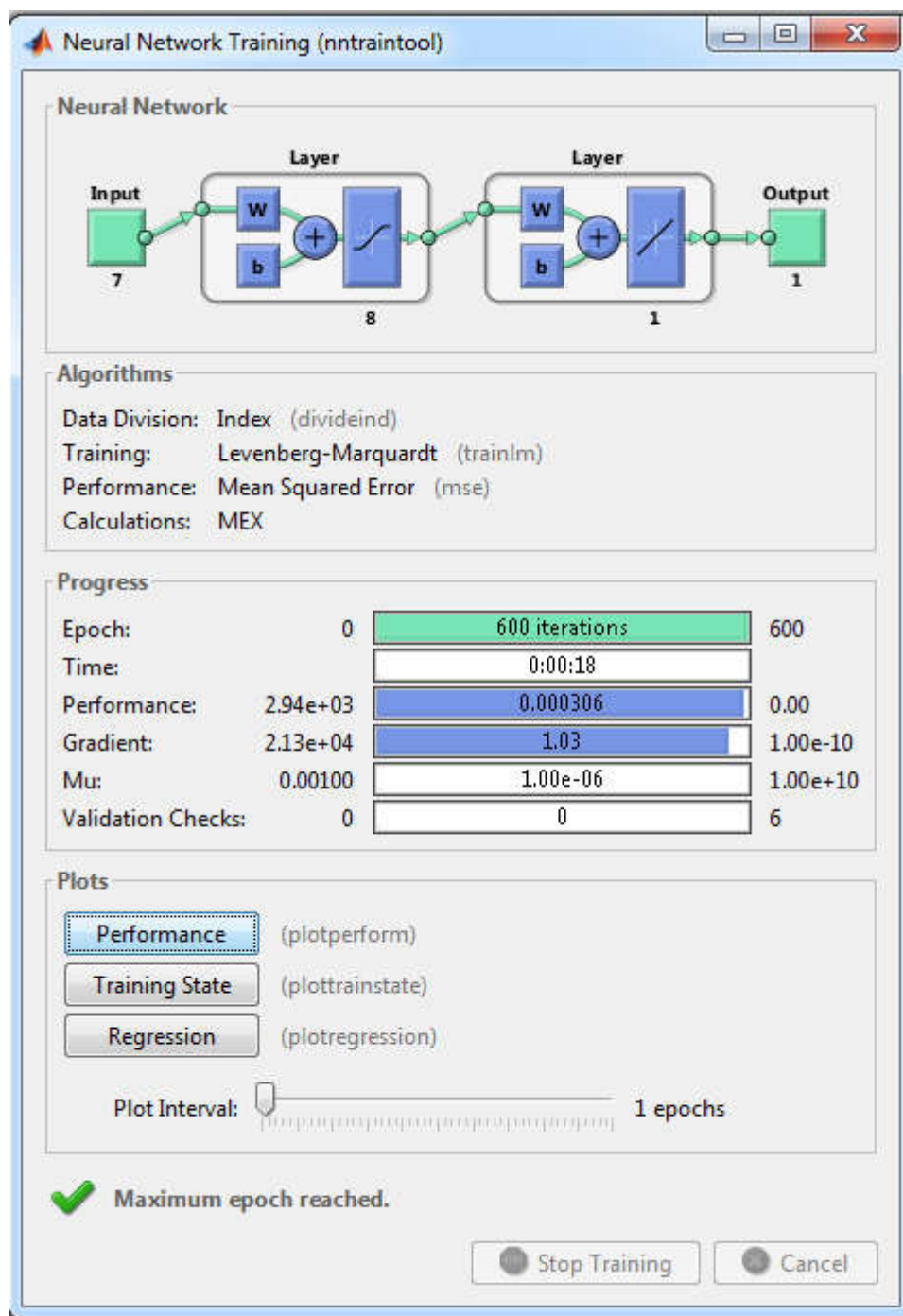


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

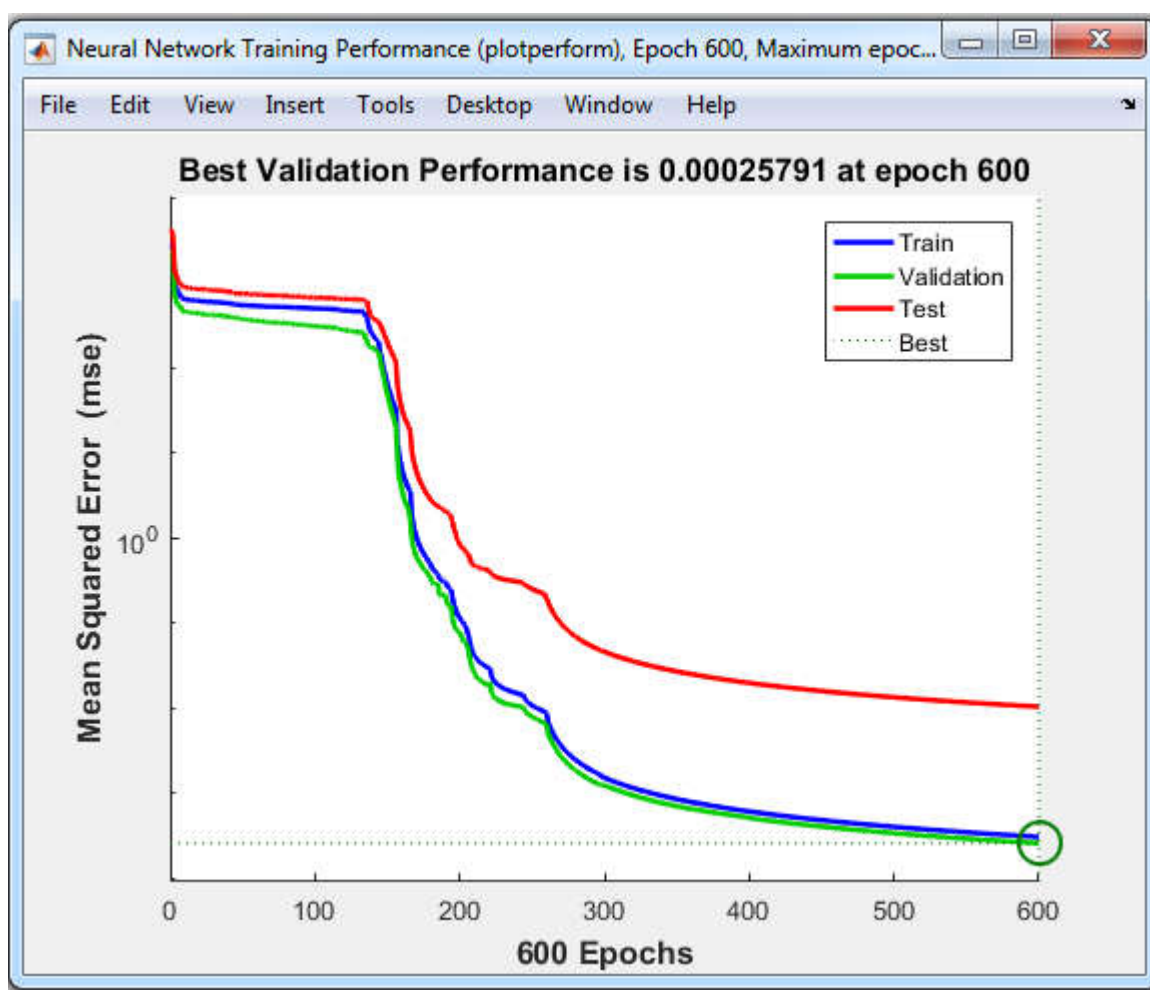


Рисунок 4.15 – Вікно контролю процесу навчання

У вікні Neural Network Training Performance відображаються графіки **Mean Squared Error (mse)**, які показують, як змінюється середньоквадратична

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

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

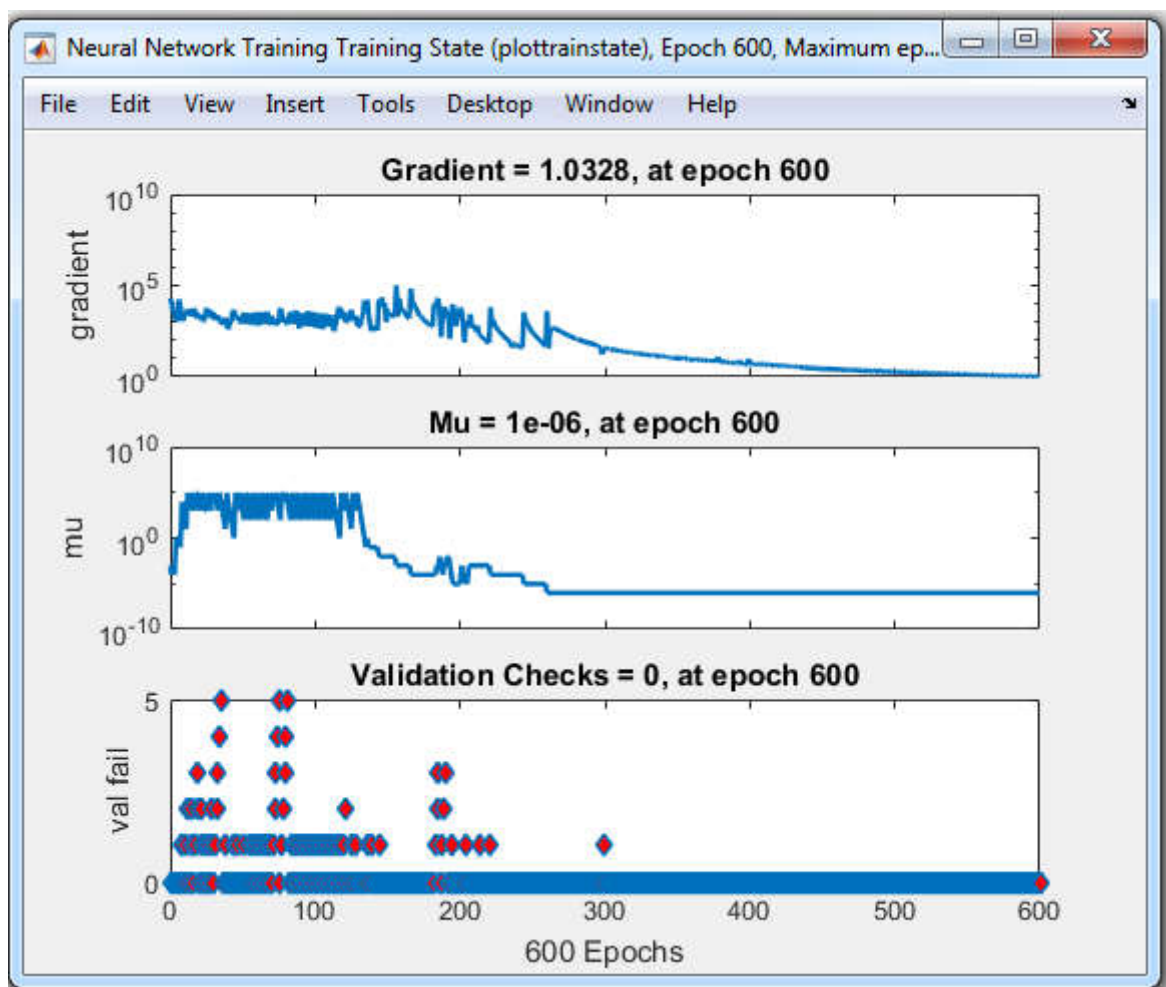


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

Графік Gradient показує зміну значення градієнта під час навчання нейронної мережі. Градієнт в контексті навчання нейронних мереж є вектором, який вказує напрямок та міру найшвидшого зростання функції втрати (або помилки) мережі в точці навчання. Градієнт використовується для оновлення вагових коефіцієнтів мережі під час оптимізації.

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

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

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

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

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

Зміна μ показує, як модифікується швидкість навчання під час процесу навчання. Зазвичай, μ може збільшуватися або зменшуватися в залежності від реакції нейронної мережі на дані. Наприклад, велике значення " μ " може вка-

зувати на те, що швидкість навчання збільшилася для прискорення збіжності, але при цьому може виникнути ризик нестабільності. Зменшення ти може вказувати на те, що навчання сталося занадто швидко або навіть зупинилося.

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

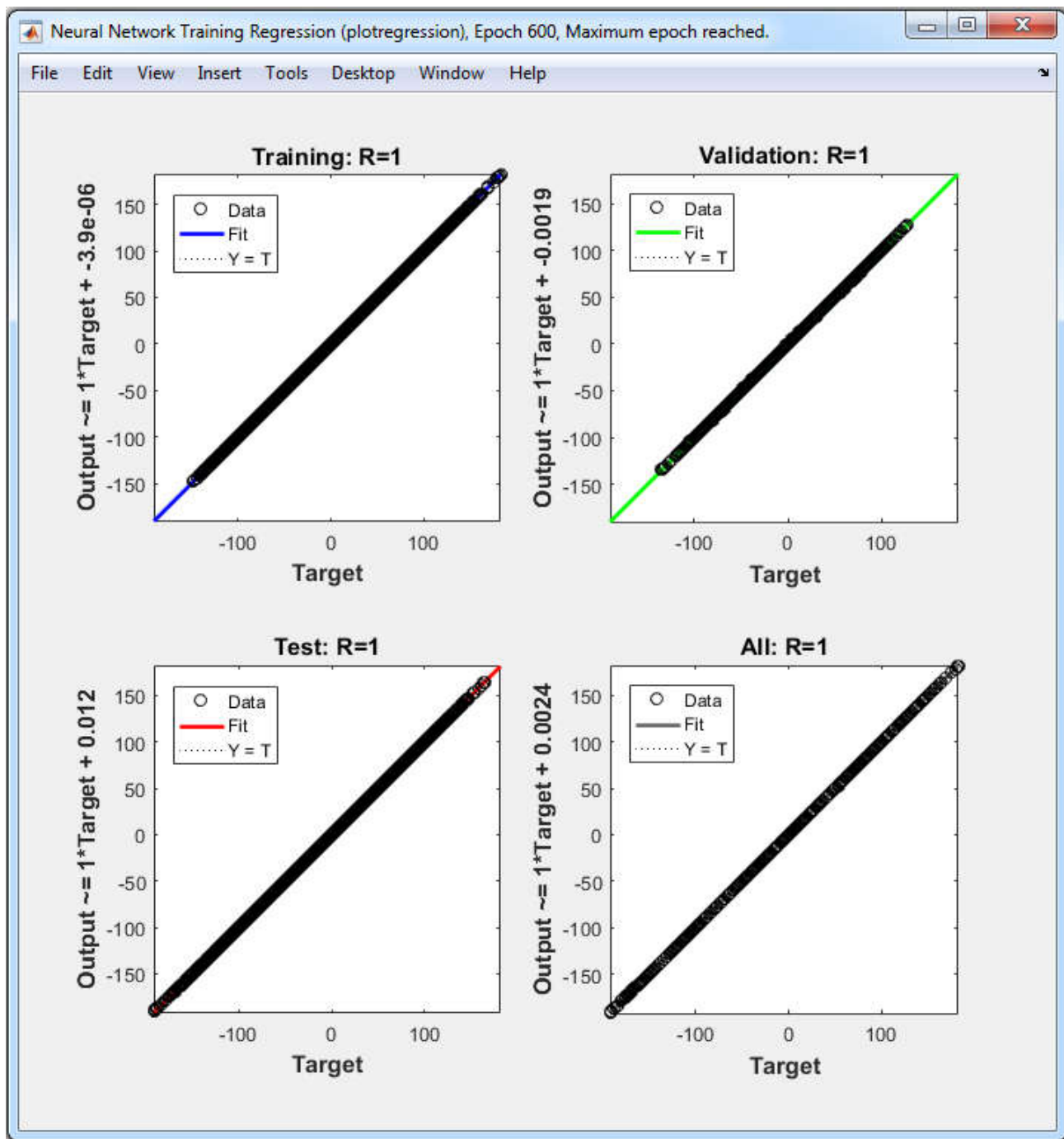


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис вікна Neural Network Training Regression

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

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

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

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

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

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

Графіки **Fit (прогноз)** – це графіки лінійної регресії, які відображають прогнозовані значення, які надає нейронна мережа на основі вхідних даних. Графіки Fit демонструють відповідність прогнозованих значень, отриманих від моделі, фактичним значенням з наборів даних. Вони допомагають візуально оцінити, наскільки точно модель регресії передбачає числові значення на основі вхідних даних. Якщо прогнозована лінія добре відповідає фактичним даним і проходить через багато точок, це свідчить про високу точність моделі. Напротив, якщо лінія регресії відхиляється від фактичних значень, це може вказувати на неадекватність моделі і необхідність налаштування параметрів нейронної мережі.

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

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

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

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

Кореляційний коефіцієнт R близький до 1 вказує на високу точність моделі регресії, тоді як значення близьке до 0 або від'ємне вказує на низьку точність. Цей коефіцієнт дуже корисний для візуальної оцінки точності моделі та порівняння прогнозованих та фактичних значень у завданнях регресії.

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

Кнопка **Cancel (скасувати)** використовується для припинення процесу навчання, але зазвичай вона надає можливість підтвердження та збереження

поточного стану моделі. Вона дозволяє вам контролювати ітерації навчання, перевіряти результати, змінювати параметри та вирішувати, коли припинити навчання з можливістю збереження результатів навчання.

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

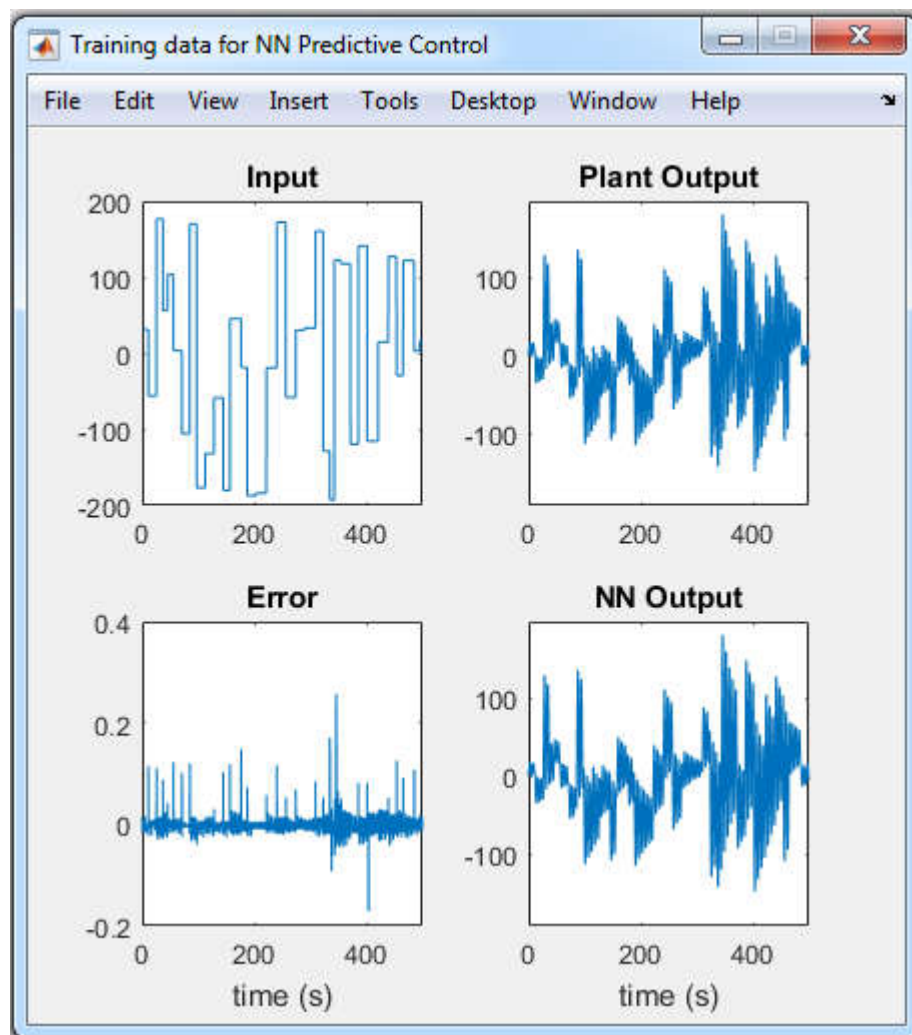


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

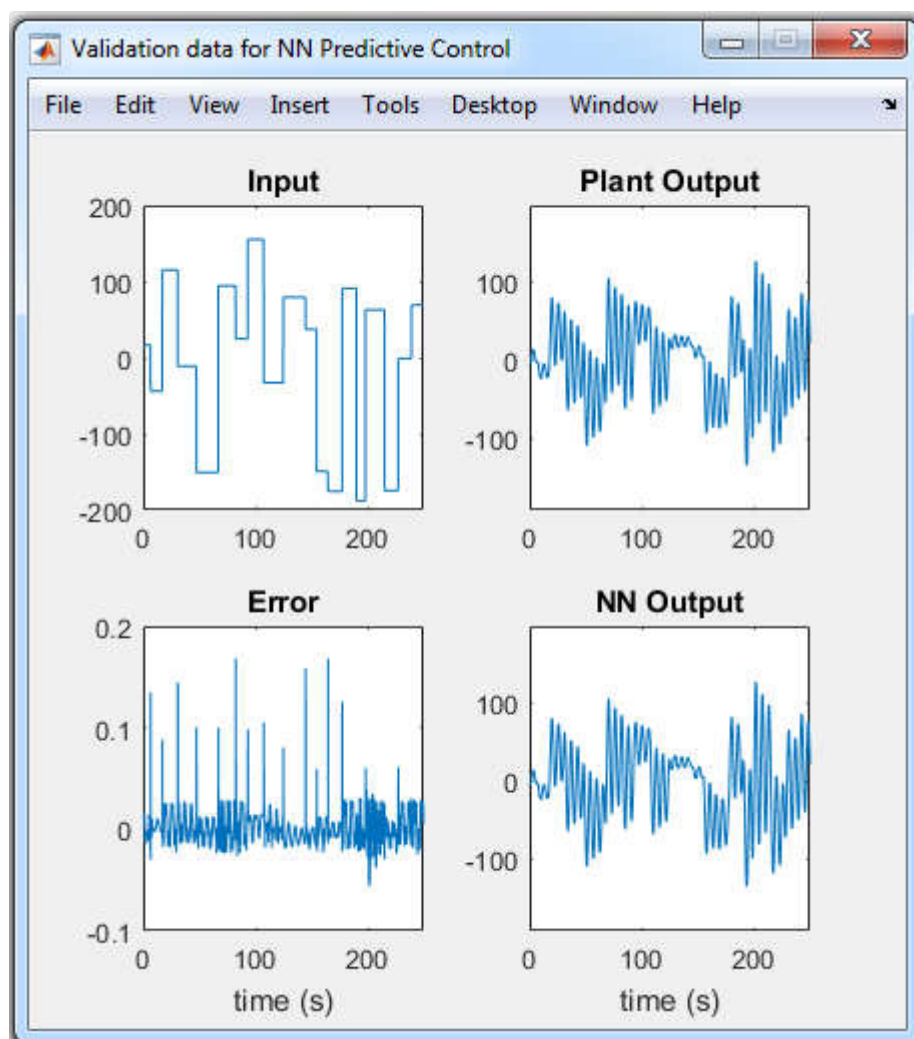


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

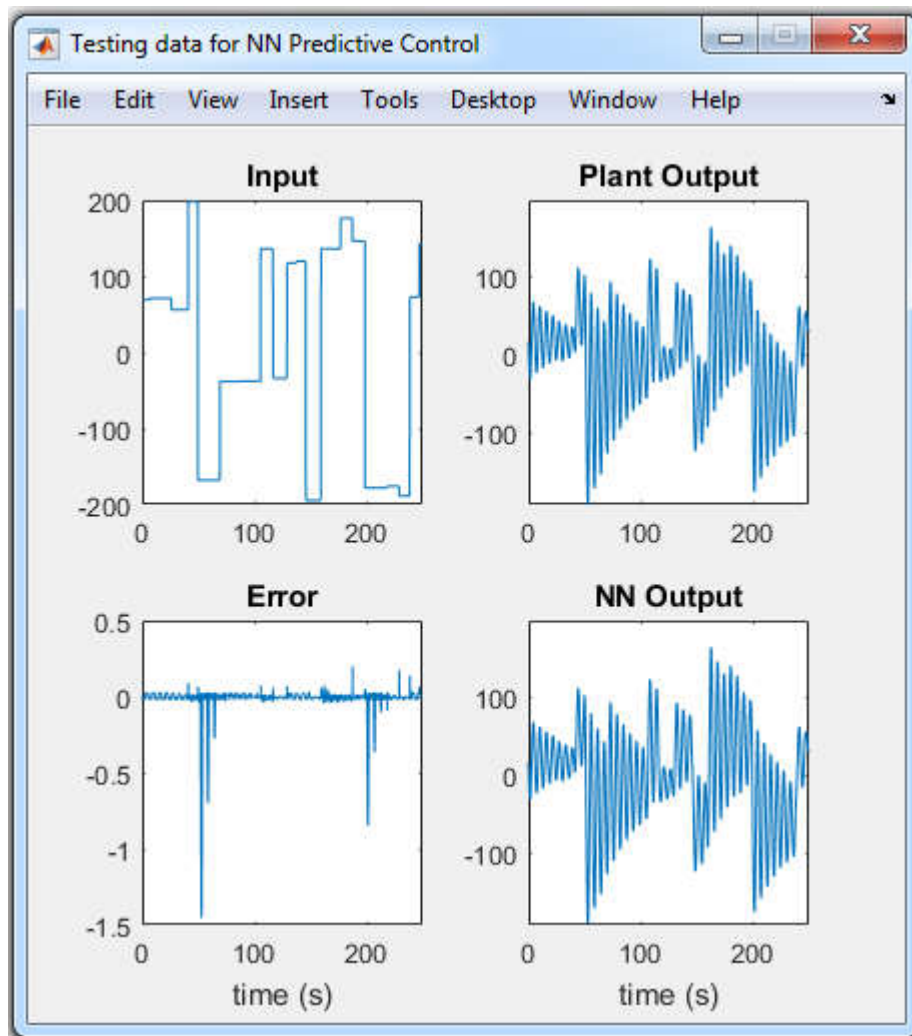


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

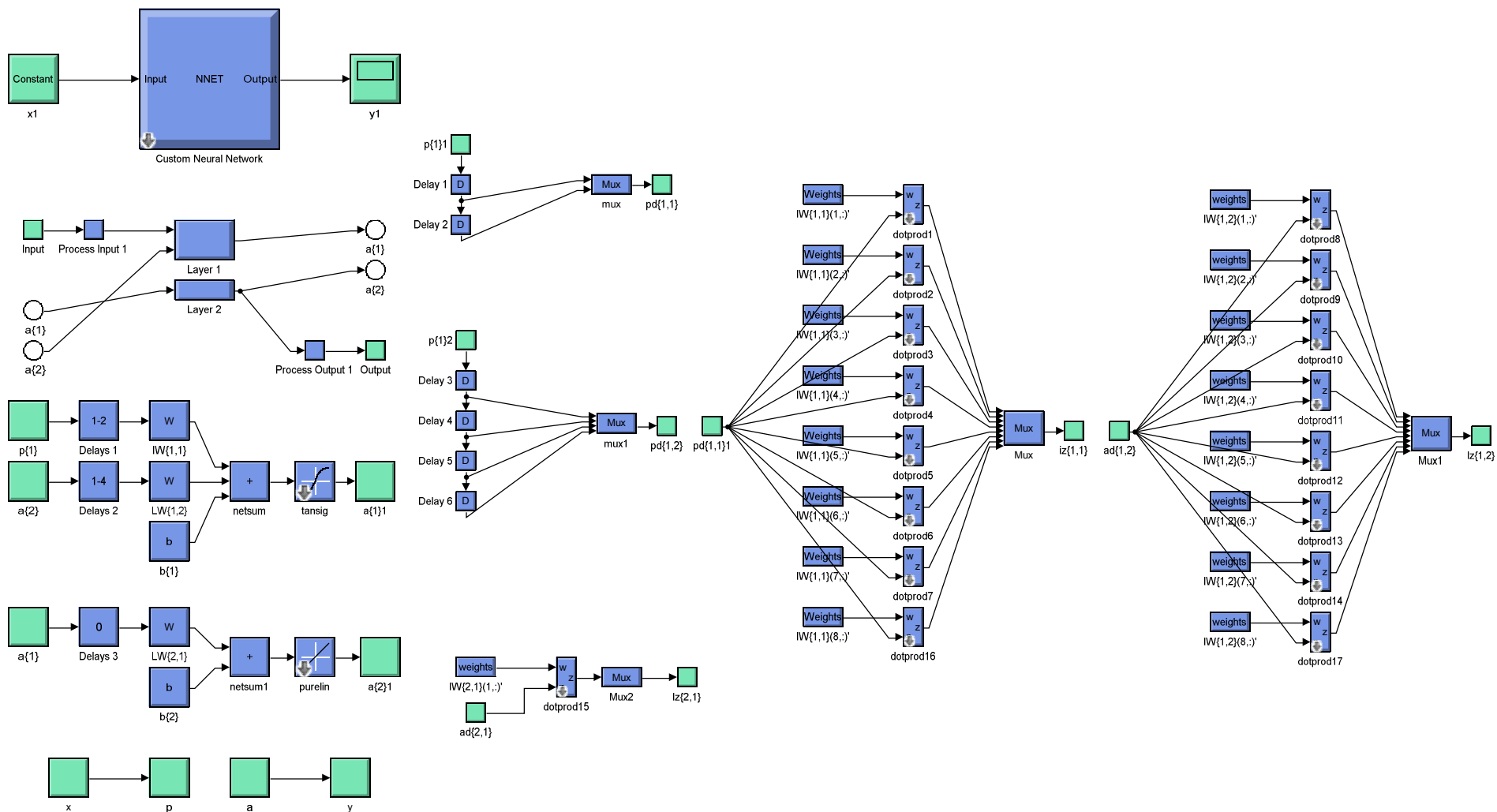


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

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

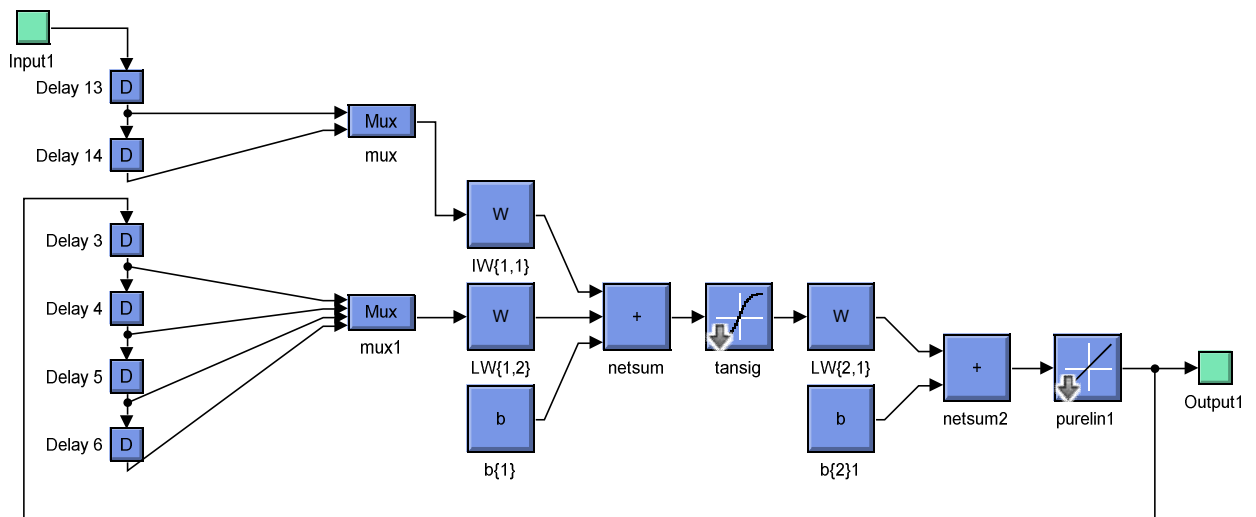


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

У результаті параметри нейромережевої моделі керованого об'єкта передаються до блоку **NN Predictive Controller** у системі Simulink. У цій системі мережа відображається у вигляді структурної схеми, наведеної на рис. 4.23

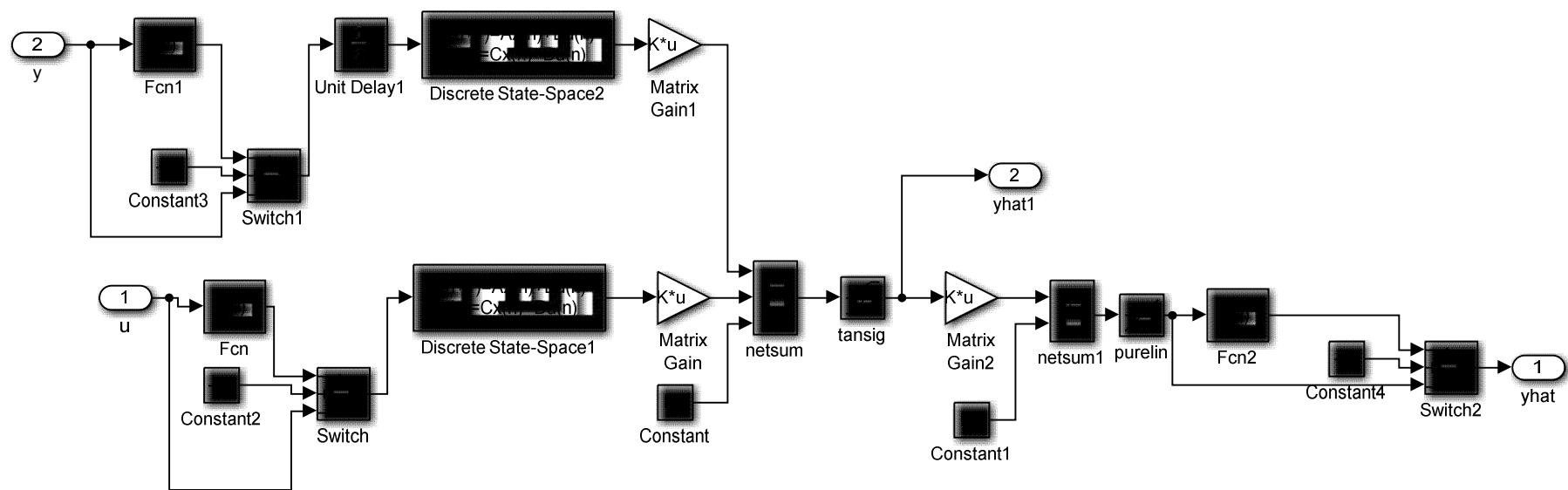


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

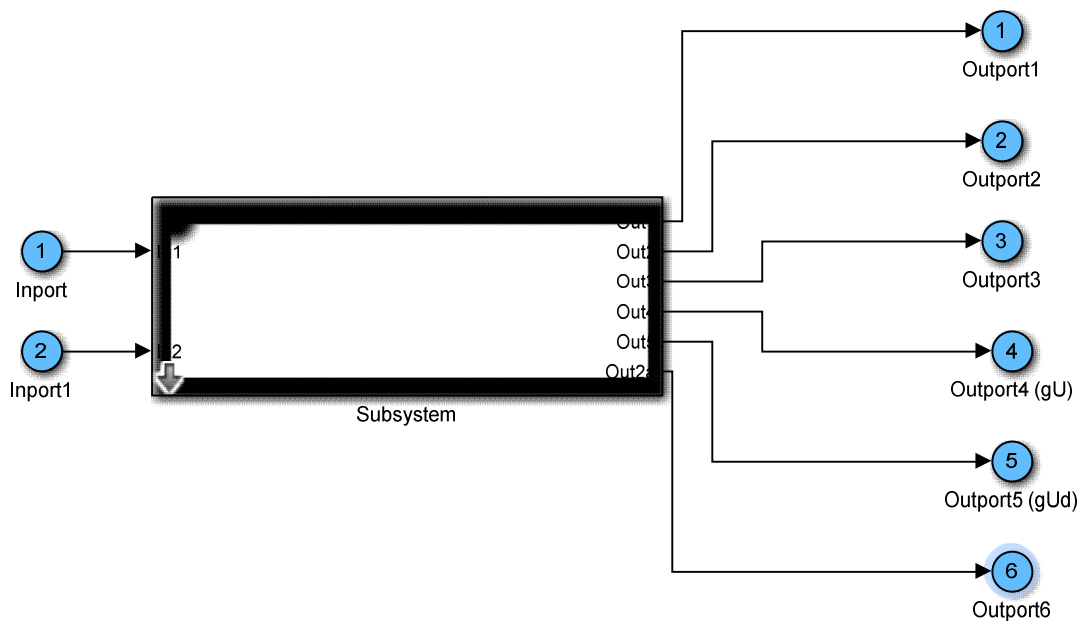


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

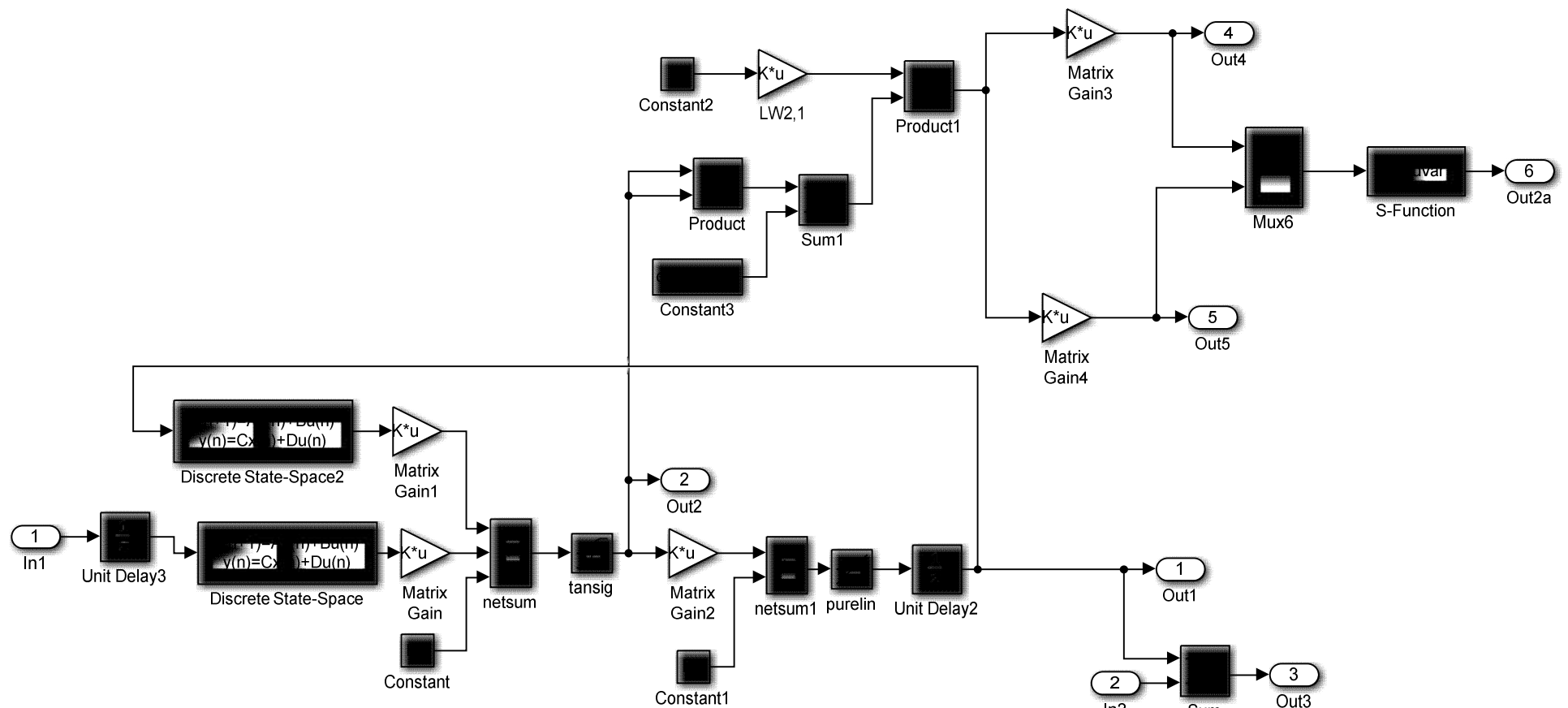


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

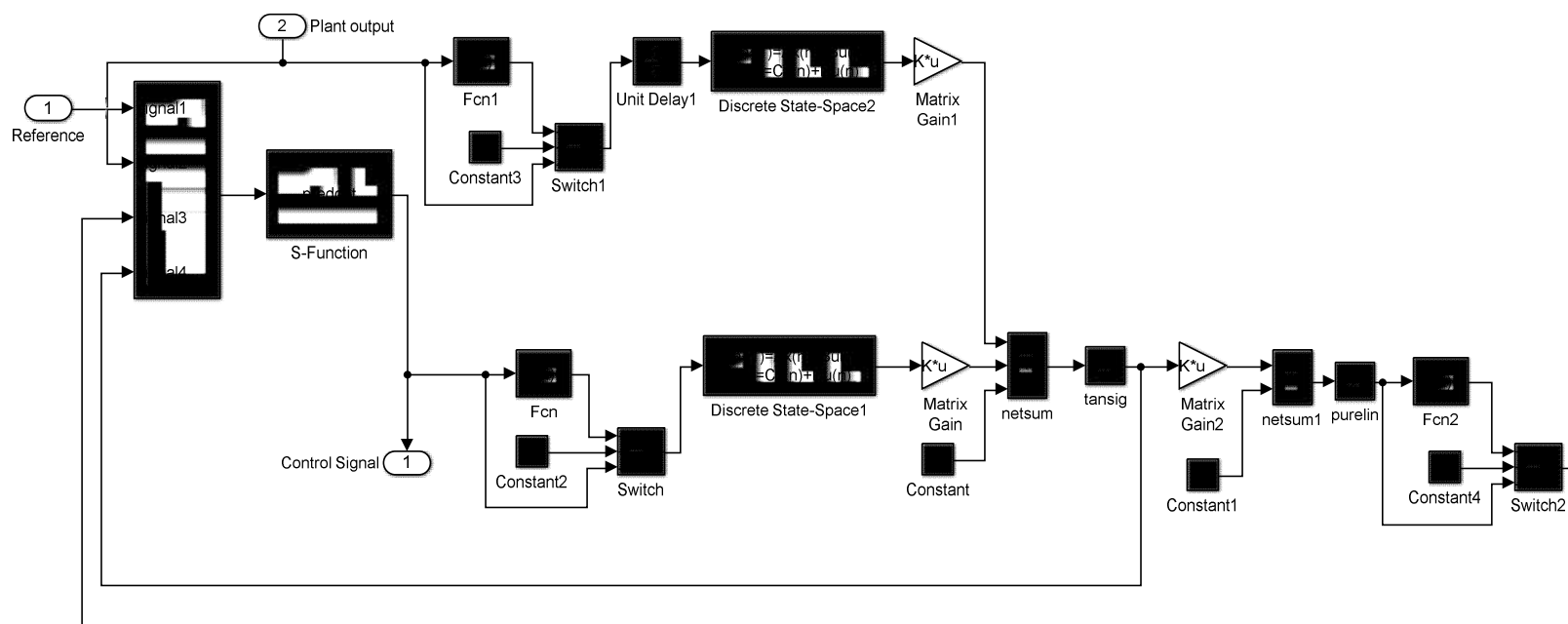


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

Iterations Per Sample Time. Параметр означає кількість ітерацій оптимізації, які виконуються протягом кожного відрізка часу (Sample Time) при розрахунку оптимального керування системою. Цей параметр дозволяє встановити, скільки ітерацій оптимізації виконується в межах кожного відрізка часу, і

впливає на швидкість та точність оптимізації контролера. Зазвичай, більша кількість ітерацій дозволяє отримати більш точну оптимальну стратегію керування, але водночас може збільшити обчислювальну складність та вимоги до ресурсів.

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

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

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

4.4 Вибір параметрів нейрорегулятора *NN Predictive Controller*

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

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

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

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

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

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

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

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

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

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

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

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

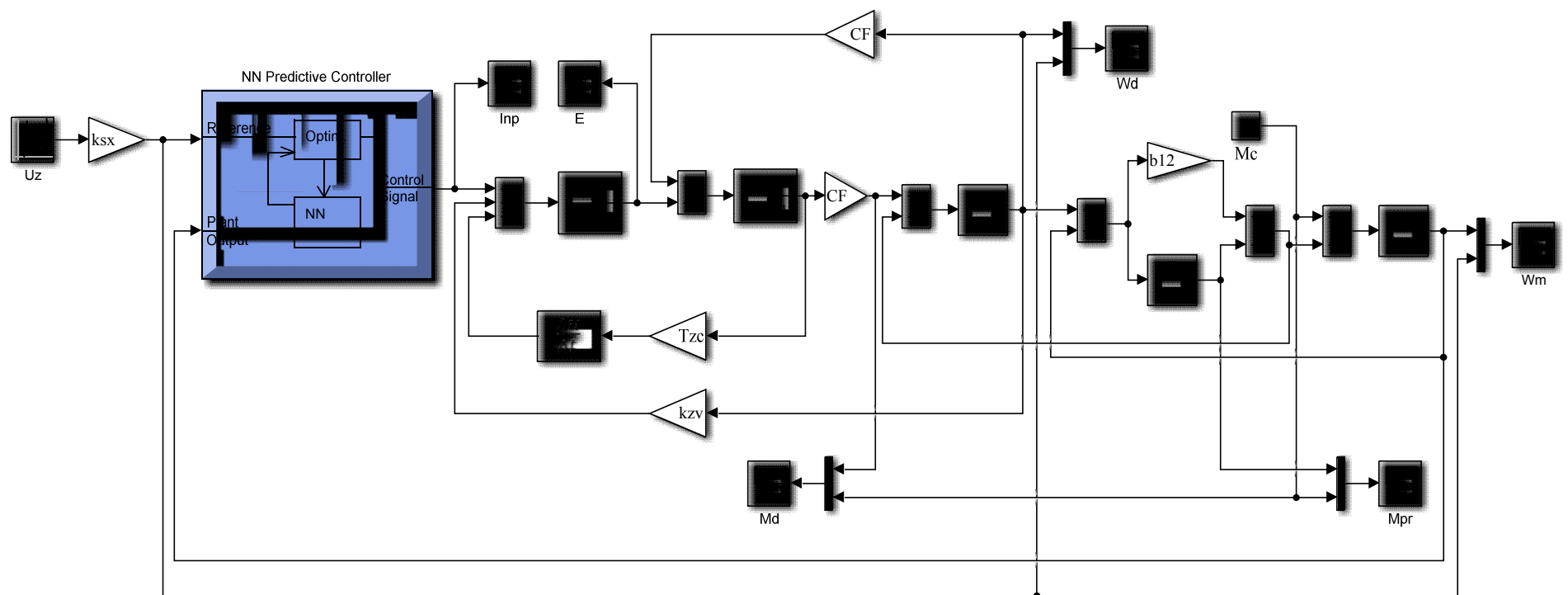


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

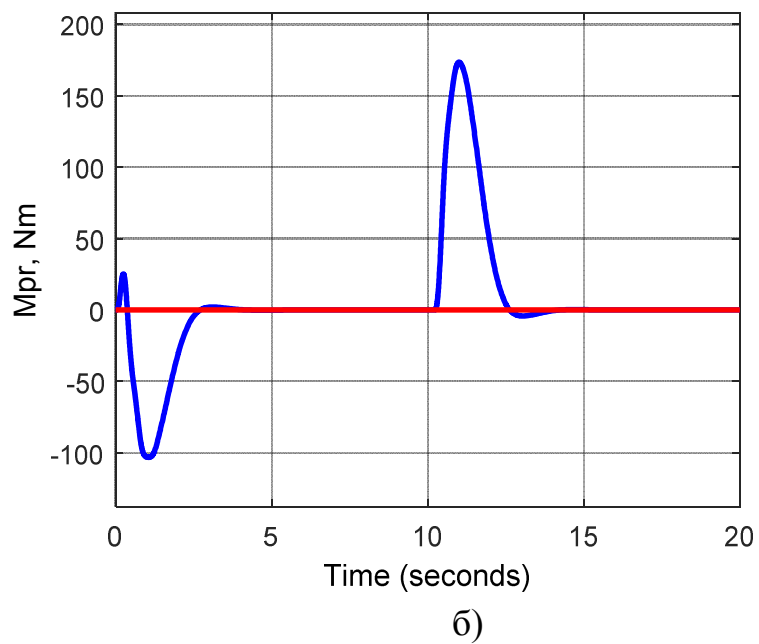
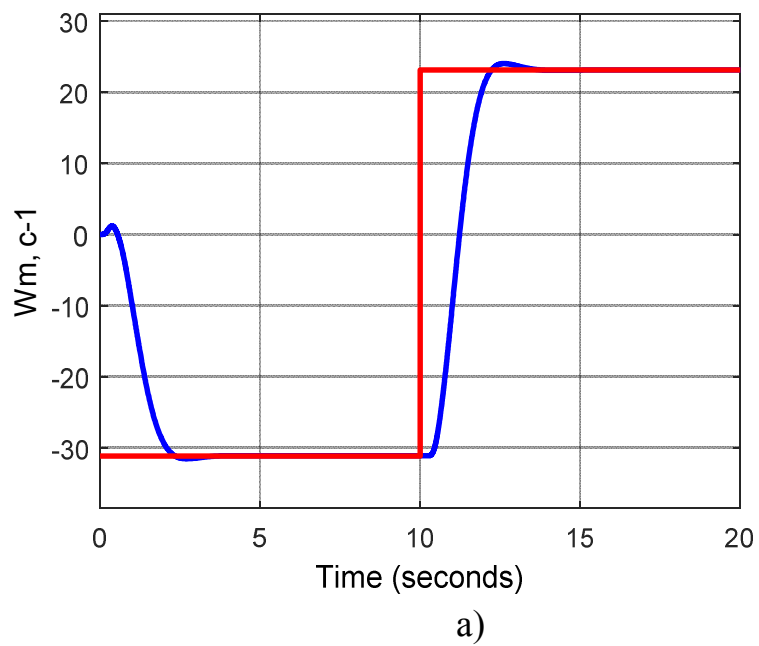


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором п по задаючій дії

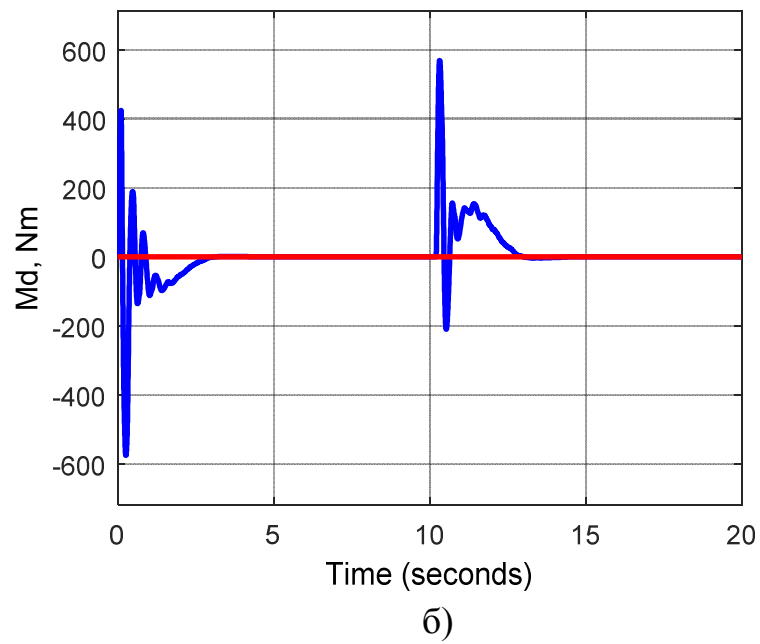
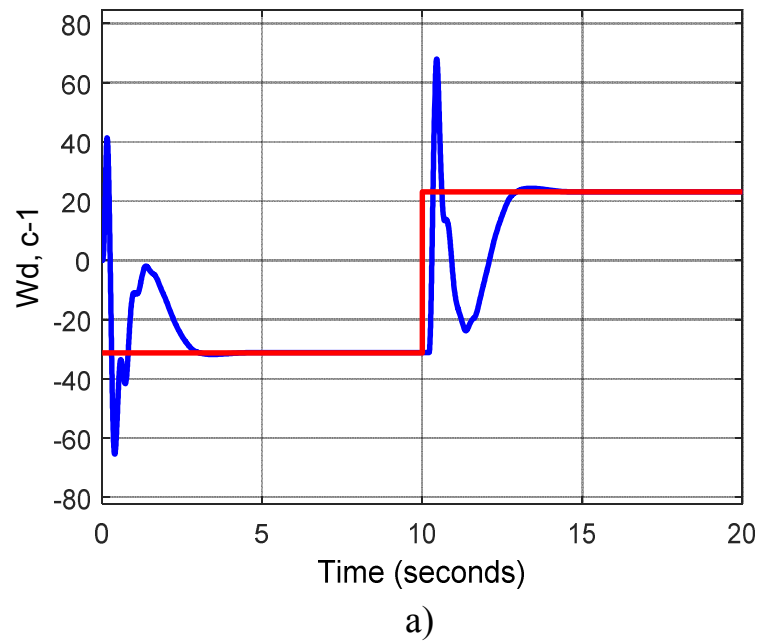


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

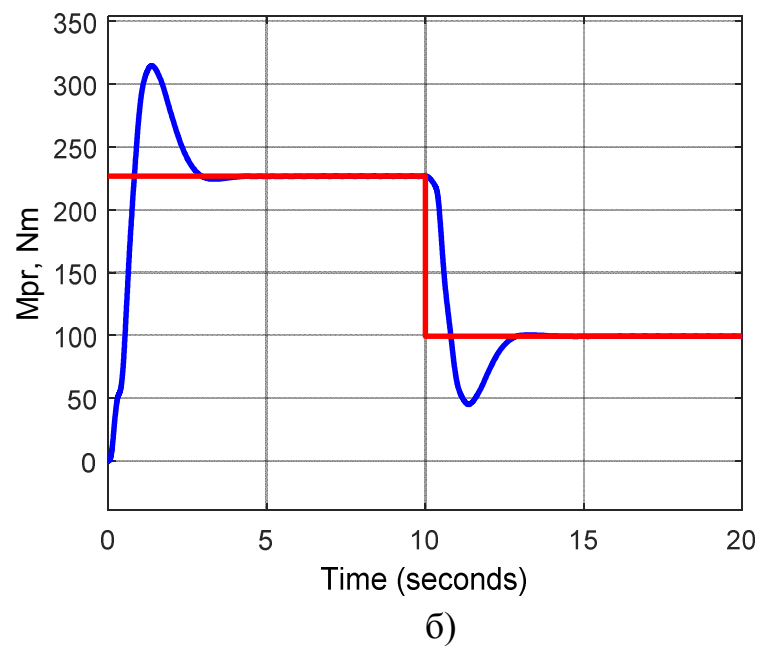
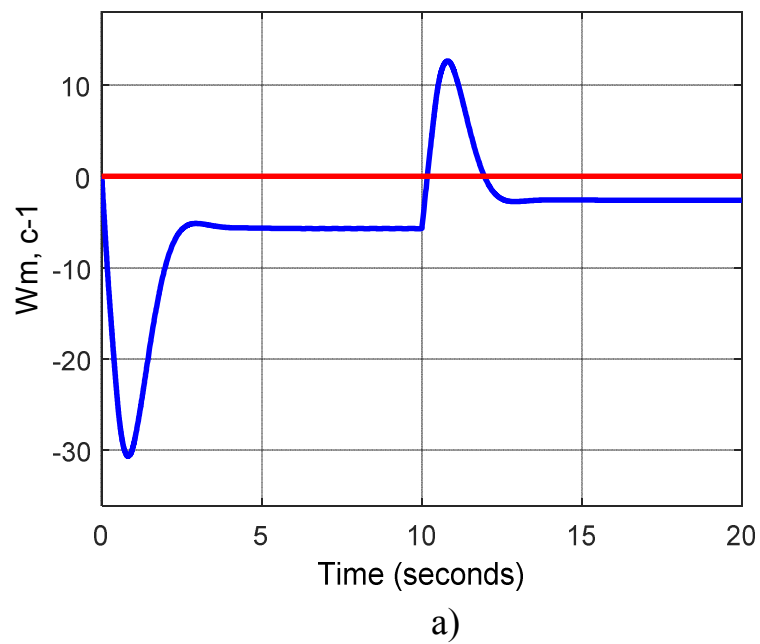


Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

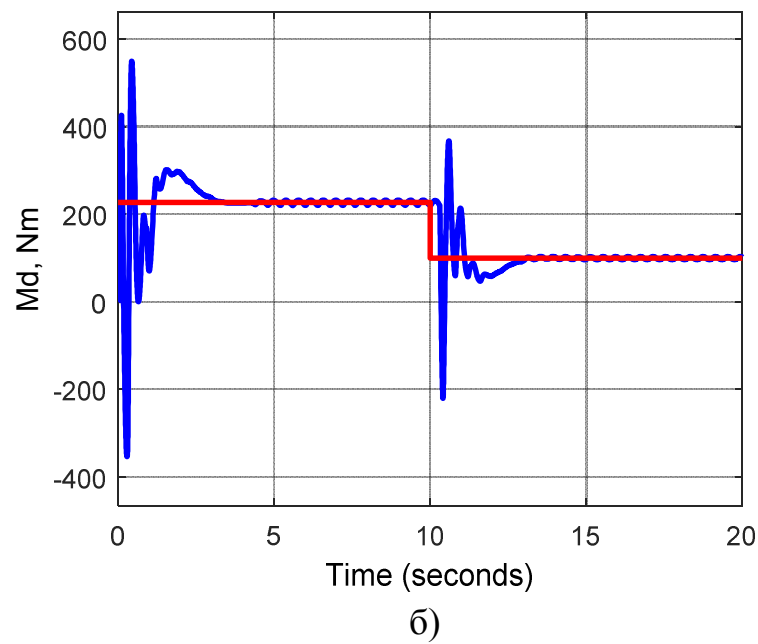
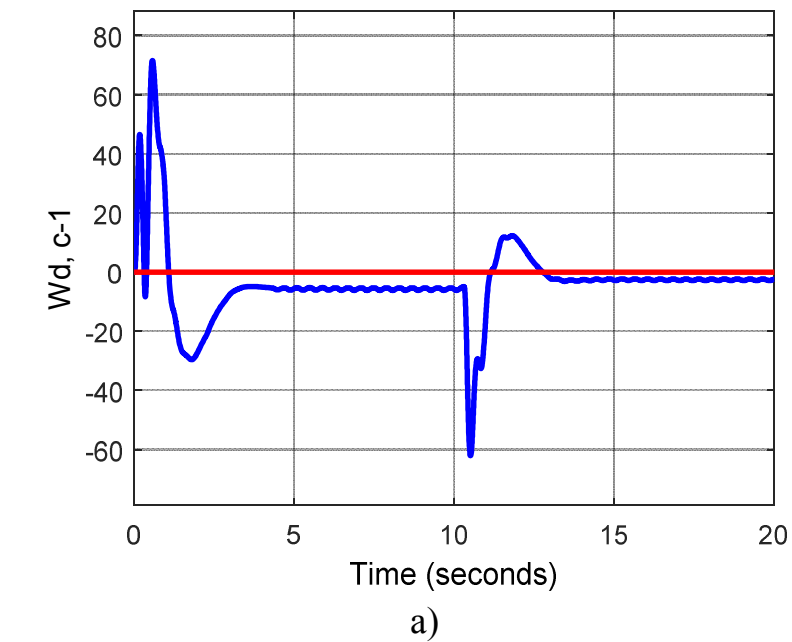


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

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

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

3. Шляхом варіювання параметрів навчальної послідовності, з використанням моделі виконавчого пристрою, виконана генерація навчальної послідовності, яка дозволила виконати тренування мережі з високим ступенем точності. Оптимальні параметри навчальної послідовності наступні: довжина навчальної вибірки $N_b = 10000$; інтервал між двома послідовними моментами знімання даних $\Delta t = 0,05 \div 0,1$ с, мінімальний інтервал ідентифікації $t_{\min} = 4 \div 5$ с; максимальний інтервал ідентифікації $t_{\max} = 10 \div 20$ с.

4. Створено двошаровий нейронна мережа з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-4} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 0,5.

5. Виконано розробку нейрорегулятора з передбаченням NN Predictive Controller. В ході варіювання параметрів нейрорегулятора в широких межах визначено, які параметри суттєво впливають на ефективність регулювання, та

встановлено їх оптимальні значення. Застосування нейромережевої моделі об'єкта управління, що забезпечує високий рівень ідентифікації, дозволило створити нейрорегулятор, який демонструє високі показники якості системи.

6. Проведено моделювання системи з синтезованим нейрорегулятором. Аналіз моделювання показав, що перехідні процеси змінних стану системи за участю нейрорегулятора NN Predictive Controller при запуску та накиді навантаження мають задовільні характеристики. Перерегулювання не перевищує 2%, а час регулювання становить 2 с, що відповідає поставленим вимогам.

ВИСНОВКИ

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

Основні результати роботи полягають у наступному.

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

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

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

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

перехідні процеси основних координат супроводжуються значними коливаннями.

5. Виконано синтез нейрорегулятора з передбаченням NN Predictive Controller. В результаті варіювання параметрів нейрорегулятора у широких межах визначено, які з них суттєво впливають на якість регулювання, та обрано їх оптимальні значення. Використання нейромережевої моделі об'єкта керування, що забезпечує високу точність ідентифікації, дозволило розробити нейрорегулятор із високими показниками якості роботи системи.

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи при запуску та прикладенні навантаження мають задовільний характер. Система є астатичною, перерегулювання не перевищує 2%, а час регулювання становить 2 с, що відповідає поставленим вимогам.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології нейронних мереж і нечіткого моделювання в системах управління: підруч. для здобувачів вищої освіти спец. 151 Автоматизація та комп'ютерно-інтегровані технології / Г.І. Канюк, Б.І. Кузнецов, Т.Ю. Василець, А.Ю. Мезеря, О.О. Варфоломійєв. – Харків : Друкарня Мадрид, 2020. – 306 с.
2. Нейромережеві технології в системах управління: Підручник для візів./ Б. І. Кузнецов, Т.Ю. Василець, Т.Б. Нікітіна, В. В. Коломиєць, О.О. Варфоломійєв; Укр. інж.-пед. акад.. - Харків: УПА, 2014. - 232 с.
3. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник.. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
4. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
5. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г.Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006, 404 с.
6. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
7. Тимощук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. -444 с.
8. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
9. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник.. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. –224 с.
10. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. –353 с.
11. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344с.
12. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления. – М.: ИПРЖР, 2002. – 480 с.

13. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review.
15. Рябинин А. Д., Шквар А. М. Некоторые принципы функционального построения инвариантных бионических систем управления // Тр. 4-го Всесоюз. совещания. "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26 - 30 апреля 1971.- Киев, 1971.- С 54-64.
16. Шквар А. М. Функциональные и структурные аспекты построения некоторых нейронных структур управления // Тр. 4-го Всесоюз. совещания "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С. 78-89.
17. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD Thesis, Harvard University, Cambridge, MA. – 1974.– P. 124-137.
18. Rumelhart D. E., Hinton G E., Williams R. J. Learning internal representation by error propagation // In: D.E. Rumelhart, J.L. McClelland (Eds.) Parallel Distributed Processing, Vol. I Foundations. – Cambridge, MA: MIT Press, 1986. – P. 318-362.
19. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representation by back-propagating errors // Nature. – 1986. – Vol.323. – P. 533-536.
20. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // IEEE Trans. Neural Networks. – 1990. – Vol.1. – №1. – P. 4-27.
21. Narendra K.S., Parthasarathy K. Gradient methods for the optimization of dynamical systems containing neural networks // IEEE Trans. Neural Networks. – 1991. – Vol.2. – № 2. – P.252-262.

22. Sunner R. M., Slotine J. E. Gaussian Networks for Direct Adaptive Control // IEEE Trans. Neural Networks. – 1992. – Vol. 3. № 6. – P. 837 - 863.
23. Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches / Ed. by David A. White & Donald A. Sofge. N.-Y. Van Nostrand Reinhold. – 1992. – 568 p.
24. 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.
25. Suykens Johan A. K., Vandewall Joos P. L., De Moor Bart L. R. Artificial Neural Networks for Modelling and Control of Non Linear Systems. Kluwer Academic Publishers. Boston, Dordrecht, London, 1997. – 235 p.
26. Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. - New Jersey: Prentice-Hall Englewood Cliffs, 1991. – 375 p.
27. Vas, P: Artificial-Intelligence-Based Electrical Machines and Drives. Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithms, Oxford University Press, Monographs in Electrical and Electronic Engineering, ISBN 0-19-859397-X. – 1999. – 626 p.
28. Efe M.O., Fiskiran A.M., Kaynak O. Derivation of a Parameter Stabilizing Training Criterion for Adaptive Neuro-Fuzzy Inference Systems in Motion Control // International Journal of Systems Science. – 2001. – Vol.32. – №4. – P. 513-521.
29. Werbos, P.J. Neurocontrol and related techniques // In Handbook of Neural Computing Applications. Hrsg. Maren,A.; Harston, C; Pap, R. Academic Press. San Diego, 1990. – P. 345-380.
30. Werbos P. J. Backpropagation and neurocontrol: A review and prospectus // Proc. of International Joint Conf. On Neural Networks. Washington, DC.–1989.–Vol. 1.– P.209-216.
31. Holland J. H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. – London: Bradford book edition, 1994. – 211 p.
32. Клепиков В.Б., Махотило К.В., Сергеев С.А., Вороновский Г.К. Искусственные нейронные сети: новая парадигма в управлении // Труды межд. конф.

- "Проблемы автоматизированного электропривода. Теория и практика". – Харьков: Основа. – 1995. – С. 111 – 115.
33. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. – Харьков: Основа. – 1997. – 112 с.
34. Клепиков В.Б., Махотило К.В., Обруч И.В. Синтез нейросетевой: системы управления одномассовой электромеханической системы с отрицательным вязким трением при ограничении координат электропривода // Труды международной конференции "Проблемы автоматизированного электропривода. Теория и практика", Харьков: Основа. – 1997. – С. 19-21.
35. Клепиков В.Б., Полянская И.С. Синтез двухмассовой электромеханической системы с отрицательным вязким трением при регулировании по полному и неполному вектору состояния // Вісник Національного технічного університету "Харківський політехнічний інститут".-Харків:НТУ "ХПИ".-2001.-Вип.10.-С.36-40.
36. Клепиков В.Б., Палис Ф., Цепковский Ю.А. Гибридные нейронные сети в управлении электромеханическими системами // Вестник НТУ "ХПИ". Проблемы автоматизированного электропривода. Теория и практика, Харьков. – 2003. – Вып.10. – Т.1. – с. 29-33.
37. Клепиков В. Б. Моисеенко П. Л. Цепковский Ю. А. Синтез гибридной нейронной сети для двухмассовой ЭМС при срывных фрикционных автоколебаниях //Вестник НТУ "ХПИ": Сб. научн. трудов "Проблемы автоматизированного электропривода. Теория и практика". – Харьков, 2003. – Вып.10.- Т.1. – С. 512-514.
38. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения. – Харьков: Телетех, 2004. – 264 с.
39. Narendra K. Gallman P.O. An iterative method for the identification of nonlinear system using a Hammerstein model // IEEE Trans. Aut. Control. - 1966. - Vol. 11. -N3. - P.546-550.
40. Psaltis, D., Sideris, A., Yamamura, A. A Multilayered neural network controller // IEEE Control Systems Magazine. - 1988. - Vol.8. - N4. -P. 17-21.

41. Joardan M.A. Generic constraints on underspecified target trajectories // Proc. Of Int. Joint Conf. On Neural Networks (IJCNN), Washington. - 1989.- Vol.1. -N3. - P.217-225.
42. Joardan M.A., Rumelhart D.E. Forward models: Supervised learning with a distal teacher // Cognitive Science.- 1990. -Vol.16. -N5.-P.313-355.
43. Kawato, M., Uno, Y., Isobe, M., Suzuki, R. Hierarchical neural network model for voluntary movements with application to robotics // IEEE Control Systems Magazine. - 1988. - Vol.8. - N7.-P.8-15
44. Saerens M, Soquet A. Neural controller based on backpropagation algorithm // IEEE Proceedings-F. - 1991. -Vol.138. -N.1. -P.55-62.
45. Miller III, W.T., Glanz, F.H., Kraft III, L.G. Application of a general learning algorithm to control of robotic manipulators // Int. J. of Robotics Research. - 1987. - Vol.6. - N2. - P.84-98.
46. Kawato, M., Furukawa, K, Suzuki, R. A hierarchical neural network model for control and learning of voluntary movement//Biological Cybernetics. - 1987. - Vol.57. - N2. -P. 169-185.
47. Pham D.T., Liu X. Neural Network for Identification, Prediction and Control. - London: Springer - Verlag, 1997. - 238p.
48. Milito R., Padilla C.S., Padilla R.A., Cadorin D. An innovations approach to dual control // IEEE Trans. Automat. Control. - 1982. -Vol.27. -N1. -P.133-137.
49. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.
50. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // IEEE Trans. Neural Networks.–1990. – Vol.1.–№1. – P.4-27.
51. Leontaritis I.J., Billings S.A. Input-output parametric model for nonlinear systems. Part 1: Deterministic non-linear systems. Part 2: Stochastic non-linear systems // Int. J. Control. – 1985. – Vol.41. – №2. –P. 303-344.
52. Bunke J. Kunstliche neuronale Netze zur Systemidentifikation aus gestorten Mefiwerten. Forsch. Ber. VDI, Reihe 8, № 667. – Dusseldorf: VDI Verlag. – 1997. – P. 37-51.

53. Системи електропривода и електрооборудования роторных экскаваторов. / Ю.Т. Калашников, А.О. Горнов, В.Н. Остриров и др. – М.; Энергоатомиздат, 1988 - 312 с.
54. Подерни Р.Ю. Горные машины и комплексы для открытых работ в 2 томах - М.; МГГУ, Издание 4, 2001 – 754 с.
55. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
56. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
57. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
58. Ладанюк А.П. Теорія автоматичного керування технологічними об'єктами : навч. посіб. / Ладанюк А.П., Архангельська К.С., Власенко Л.О. – К.: НУХТ, 2014 – 274 с.
59. Сорока К.О. Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина перша. Основи теорії систем автоматичного керування. : навч. посібник. – Харків : ФОП Тімченко, 2010. – 218 с.
60. Сорока К.О. Теорія автоматичного керування і комп'ютерне моделювання (неперервні лінійні системи). Частина друга. Аналіз систем автоматичного керування засобами комп'ютерного моделювання : навч. посібник. – Харків : ФОП Тімченко, 2010. – 156 с.
61. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
62. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
63. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.

64. Автоматизований електропривод ч. 2 [Електронний ресурс]: навчальний посібник для студентів освітньої програми «Електромеханічні системи автоматизації, електропривод та електромобільність» спеціальності 141 «Електроенергетика, електротехніка та електромеханіка» / В.І. Теряєв. – Київ: КПІ ім. Ігоря Сікорського, 2022.– 204 с
65. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч1.
66. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч2.
67. Системи керування електроприводами. Видання 2: Навч. посібник з дисципліни «Системи керування електроприводами» (для студентів спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» денної і заочної форми навчання)/ – Краматорськ: ДДМА, 2018. – 225 с.
68. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоменко. Харків, ХНАДУ, 2012. – 284 с.
69. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.
70. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
71. Дьяконов В.П., Круглов В.В. MATLAB 6.5 SP1/SP2 + Simulink 5/6 инструменты искусственного интеллекта и биоинформатики. Серия «Библиотека профессионала». – М.: СОЛОН-ПРЕСС, 2006.– 456 с.
72. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

ДОДАТОК А

БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ

А.1 Структура штучного нейрона

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

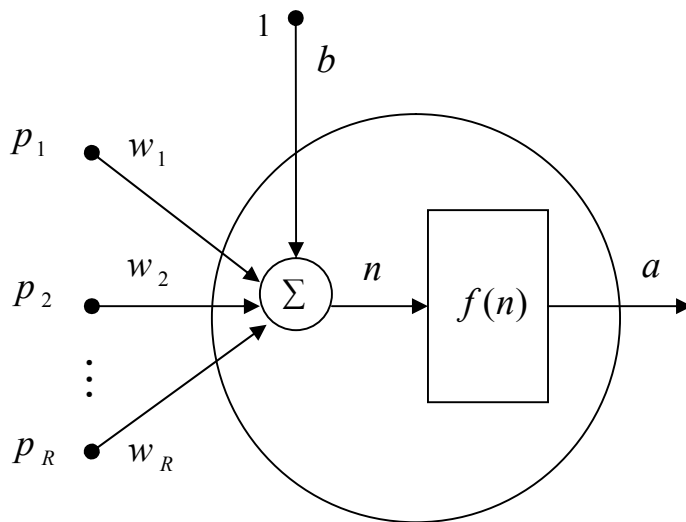


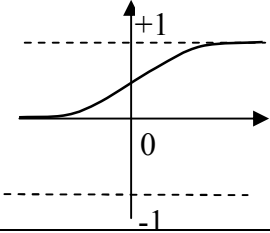
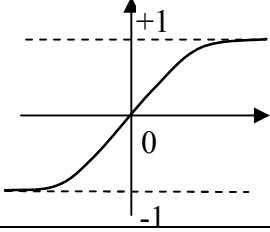
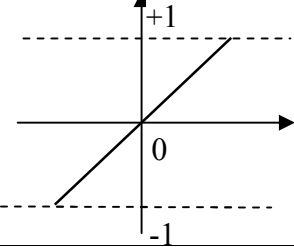
Рисунок А.1 - Схема штучного нейрона

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

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

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

Таблиця А.1 – Передавальні функції персептрона

Передавальна функція	Графічне зображення	Функція	Похідна
сигмоїдальна		$f(n) = \frac{1}{1 + e^{-n}}$	$f'(n) = f(n)(1 - f(n))$
тангенс-гіперболічна		$f(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}}$	$f'(n) = 1 - f^2(n)$
лінійна		$f(n) = n$	$f'(n) = 1$

Рівняння нейрона має вид

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

А.2 Одношарові нейронні мережі

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

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

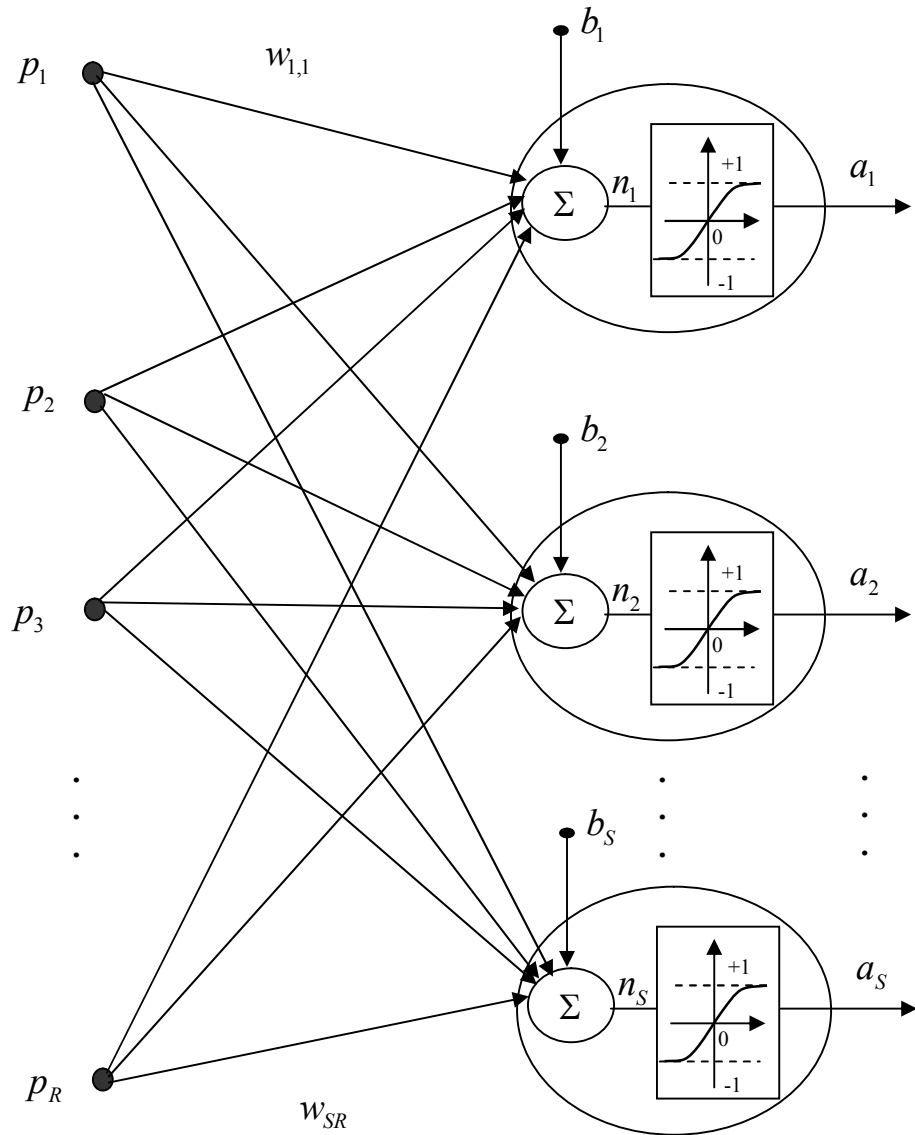


Рисунок А.2 - Структурна схема одношарової мережі
(одношаровий персептрон)

В цій мережі кожен елемент вектора входу сполучений зі всіма входами нейрона і це з'єднання задається матрицею вагів $\mathbf{W}$ при цьому кожен i -й ней-

рон включає елемент, що підсумовує, який формує скалярний вихід $n(i)$. Сукупність скалярних функцій $n(i)$ об'єднується в S -елементний вектор входу $\mathbf{n}$ функції активації шару. Виходи шару нейронів формують вектор-стовпець $\mathbf{a}$, і, таким чином, опис шару нейронів має вигляд:

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

У кожному шарі, як правило, використовується одна і та ж функція активації. Проте можна створювати складені шари нейронів з використанням різних функцій активації, сполучаючи мережі, подібні зображеною на рис.. А.2, паралельно. Обидві мережі матимуть ті ж самі входи, і кожна мережа генеруватиме певну частину виходів. Елементи вектора входу передаються в мережу через матрицю вагів $\mathbf{W}$, що має вигляд:

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

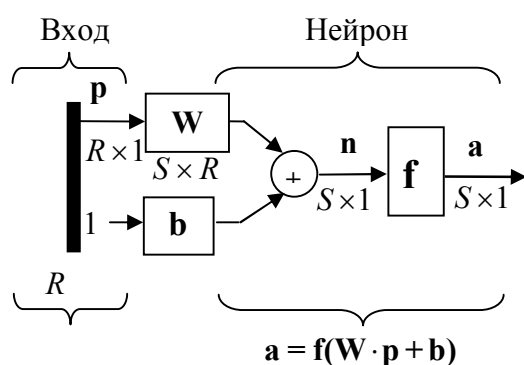


Рисунок А.3 – Укрупнена структурна схема одношарової мережі

Індекси рядків матриці $\mathbf{W}$ указують адресатів (пункти призначення) вагів нейронів, а індекси стовпців – яке джерело є входом для цієї ваги. Таким чином, елемент матриці вагів $w_{1,2} = \mathbf{W}(1,2)$ визначає коефіцієнт, на який множиться другий елемент входу при передачі його на перший нейрон. Укрупнена структурна схема одношарової мережі показана на рис. А.3.

А.3 Багат шарові нейронні мережі

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

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

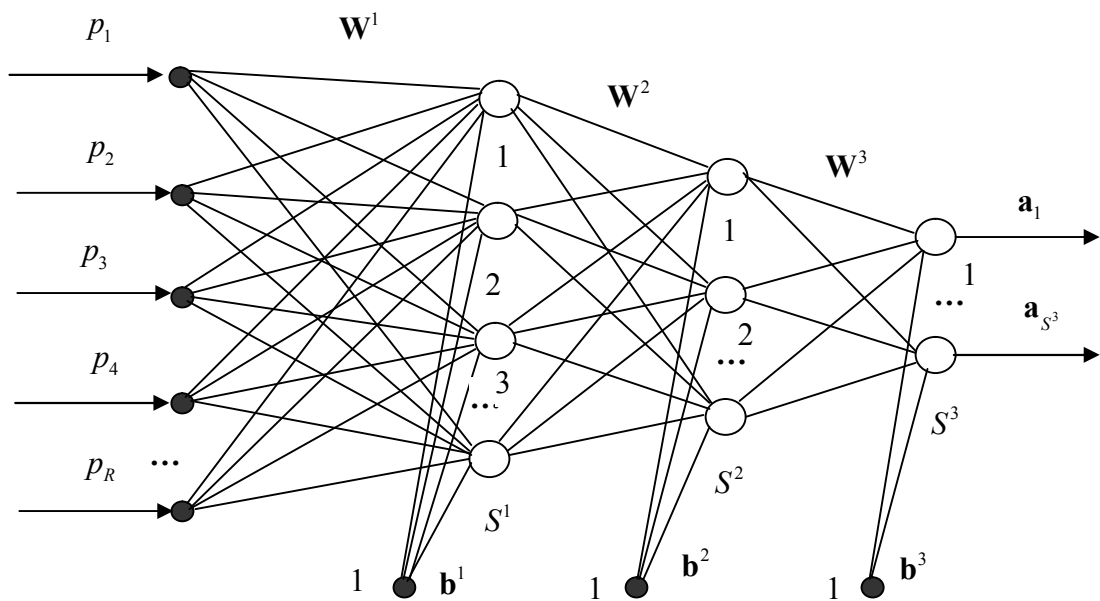


Рисунок А.4 – Структурна схема тришарової мережі
(тришаровий персептрон)

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

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

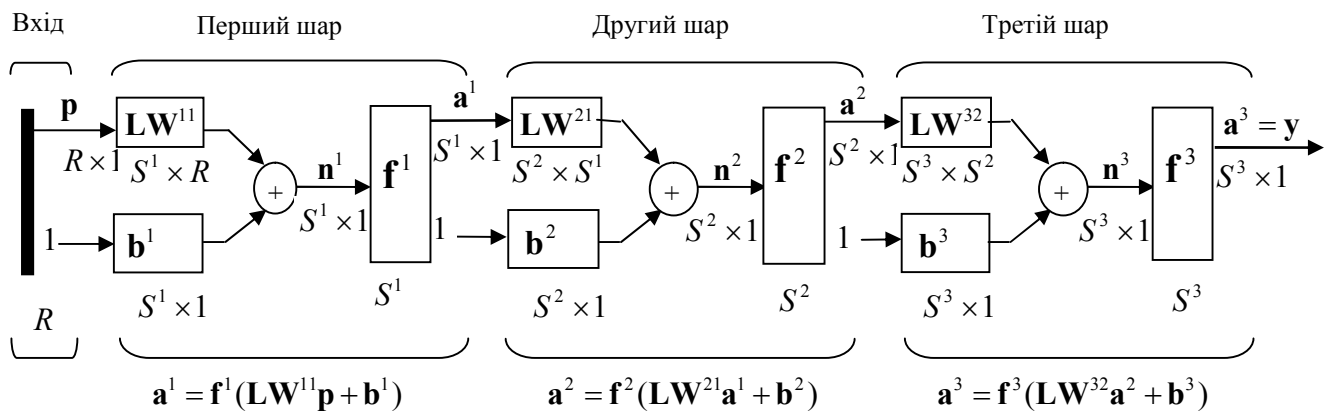


Рисунок А.5 – Укрупнена структурна схема тришарової мережі

А.4 Здібності апроксимації і узагальнення багатшарових мереж

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

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

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

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

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

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

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

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

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

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

ДОДАТОК Б

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

Б.1 Завдання навчання

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

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

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

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

Для більшості реальних завдань буває достатнім декількох сотень або тисяч спостережень. Для складних завдань може потрібно більша кількість, проте дуже рідко зустрічаються завдання, де потрібно менше 100 спостережень. Якщо

даних мало, то мережа не має достатньої інформації для навчання, і краще, що можна в цьому випадку зробити, – це спробувати підігнати до даним деяку лінійну модель.

Б.2 Процедури адаптації і навчання

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

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

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

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

У разі нелінійної моделі поверхня помилок має набагато складнішу будову і має ряд несприятливих властивостей, зокрема може мати локальні мінімуми, плоскі ділянки, сідлові точки і довгі вузькі яри.

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

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

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

При використанні алгоритму зворотного розповсюдження помилки мережа розраховує помилку, що виникає у вихідному шарі, і обчислює вектор градієнта як функцію вагів і зсувів. Цей вектор указує напрям найкоротшого спуску по поверхні для даної крапки, тому якщо просунутися в цьому напрямі, то помилка зменшиться. Послідовність таких кроків врешті-решт приведе до мінімуму того або іншого типу. Певну трудність тут викликає вибір величини кроку.

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

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

Б.3 Явище перенавчання

Одна з найбільш серйозних труднощів при навчанні мережі полягає в тому, що у ряді випадків ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати; потрібно мінімізувати помилку, яка з'являється в мережі, коли на неї подаються абсолютно нові спостереження. Вельми важливо, щоб нейронна мережа володіла здатністю пристосовуватися до цих нових спостережень. Що ж відбувається насправді? Мережа навчається мінімізувати помилку на деякій обмеженій повчальній множині. Це не відповідає вимогам теорії про наявність ідеальної і нескінченно великої повчальної множини. І це не відповідає тій реа-

льній ситуації, коли треба мінімізувати конкретну функцію помилок для наперед невідомої моделі.

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

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

Для того, щоб виявити ефект перенавчання, використовується механізм контрольної перевірки. Частина повчальних спостережень резервується як контрольні спостереження і не використовується при навчанні мережі. Натомість у міру роботи алгоритму ці спостереження застосовуються для незалежного контролю результату. Спочатку помилка мережі на повчальне і контрольному множині буде однаковою; якщо вони істотно відрізняються, то, мабуть, це означає, що розбиття спостережень на 2 множини не забезпечило їх однорідність. У міру навчання мережі помилка убиває, і, поки навчання зменшує функ-

цію помилок помилка на контрольній множині також убуватиме. Якщо ж контрольна помилка перестала убувати або стала рости, це указує на те, що мережа почала дуже близько слідувати початковим даним і навчання слід зупинити. В цьому випадку слід зменшити кількість нейронів або шарів, бо мережа є дуже могутнього для вирішення даного завдання. Якщо ж, навпаки, мережа має недостатню потужність, щоб відтворити наявну залежність, те явище перенавчання швидше за все спостерігатися не буде і обидві помилки – навчання і перевірки – не досягнуть достатньо малого рівня.

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

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

Отже, процедура побудови нейронної мережі складається з наступних кроків:

вибору початкової конфігурації мережі; наприклад, у вигляді одного шару з числом нейронів, рівним $1/2$ загальної кількості входів і виходів;
моделювання і навчання мережі з оцінкою контрольної помилки і використанням додаткових нейронів або проміжних шарів;
виявлення ефекту перенавчання і коректування конфігурації мережі.

Б.4 Властивість узагальнення

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

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

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

умови і типи освітлення, при яких мережу передбачається використовувати, і це не кажучи ще про рельєф місцевості, вугілля і дистанцію зйомки і т.д.

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

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

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

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

Б.5 Вибір методу навчання нейронної мережі

Б.5.1 Критерій якості навчання. Методи навчання.

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

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

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

де J – функціонал;

Q – об'єм вибірки;

L – число шарів мережі;

q – номер вибірки;

S^L – число нейронів вихідного шаруючи;

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

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

для вибірки з номером q .

Потім за допомогою того або іншого методу навчання визначаються значення параметрів (вагів і зсувів), що настроюються, мережі, які забезпечують мінімальне значення функціонала помилки.

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

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

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

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

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

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

Б.5.2 Метод зворотного розповсюдження помилки.

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

Навчання мережі включає декілька кроків:

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

Нейронні мережі, призначені для вирішення практичних завдань, можуть містити до декількох тисяч параметрів, що настроюються, тому обчислення градієнта може зажадати вельми великих витрат обчислювальних ресурсів. З урахуванням специфіки багатошарових нейронних мереж для них розроблені спеціальні методи розрахунку градієнта, серед яких слід виділити метод зворотного розповсюдження помилки (ЗР), який був розроблений в 1986г. Потім з'явилися численні роботи по прискоренню збіжності алгоритму. Метод звор-

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

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

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

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

де S^L – число нейронів в шарі;

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

Правило функціонування шару L записується так:

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

Приймаючи до уваги, що

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

отримаємо

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

Ввівши позначення для локальної помилки вихідного шару

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

отримаємо

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

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

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

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

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

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

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

де r – номер шару

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

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

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

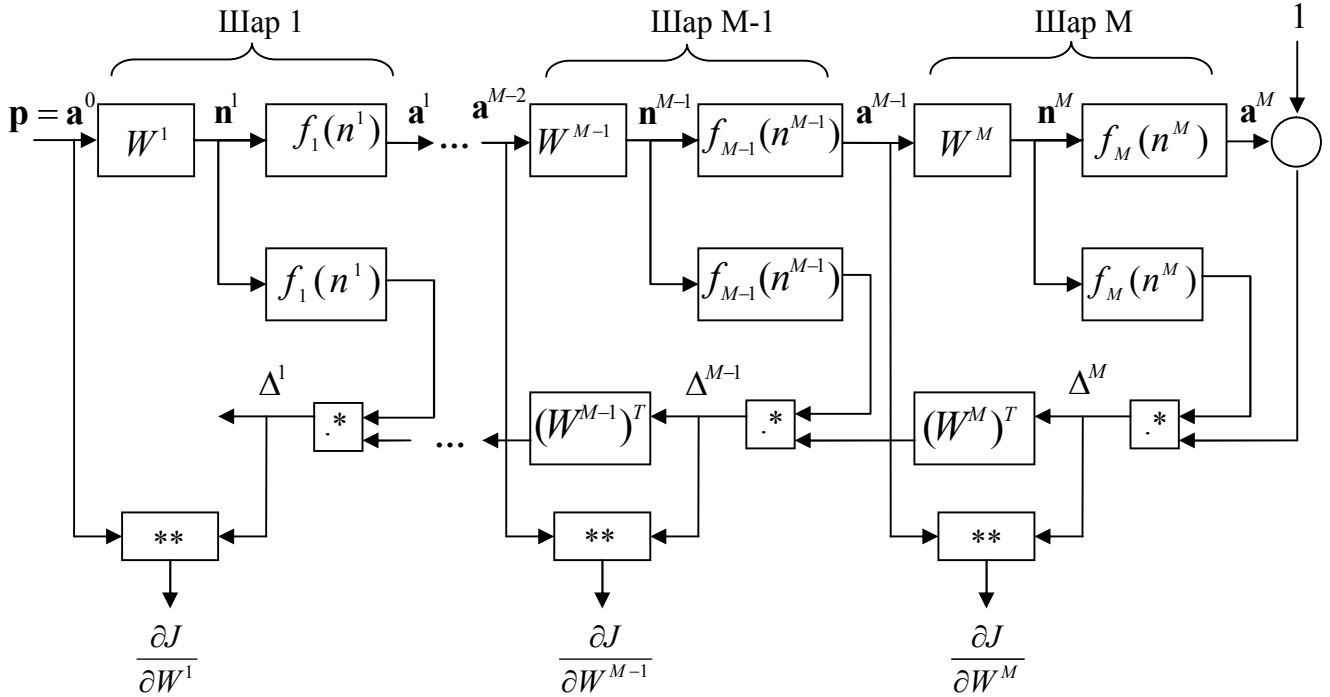


Рисунок Б.1 – Схема обчислення приватних похідних критерію J по елементах матриць вагових коефіцієнтів

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

Б.5.3 Налаштування вагових коефіцієнтів мережі.

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

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

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

де a_k – параметр швидкості навчання;

$\mathbf{g}_k$ – градієнт функціонала, відповідний ітерації з номером k .

Вектор в напрямі, протилежному градієнту, указує напрям найкоротшого спуску по поверхні функціонала помилки. Якщо реалізується рух в цьому напрямі, то помилка зменшуватиметься. Послідовність таких кроків врешті-решт приведе до значень параметрів, що настроюються, що забезпечують мінімум функціонала. Певну трудність тут викликає вибір параметра швидкості навчання a_k . При великому значенні параметра a_k збіжність буде швидкою, але існує небезпека пропустити рішення або піти в неправильному напрямі. Класичним прикладом є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. Залежно від прийнятого алгоритму параметр швидкості навчання може бути постійним або змінним. Правильний вибір цього параметра залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; у разі змінного параметра його значення зменшується у міру наближення до мінімуму функціонала. Згідно (Б.3) уточнюються значення вагів $\mathbf{w}_k$ в методі ЗР в його стандартній реалізації.

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

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

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

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

де $\mathbf{p}_k$ – напрям руху;

$\mathbf{g}_k$ – градієнт функціонала помилки;

β_k – коефіцієнт відповідають ітерації з номером k .

Коли напрям спуску визначений, то нове значення вектора параметрів, що настраюються, $\mathbf{w}_{k+1}$ обчислюється за формулою

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

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

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

Основний крок методу Ньютона визначається по формулі

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

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

У багатьох випадках метод Ньютона сходиться швидше, ніж методи зв'язаного gradientа, але вимагає великих витрат із-за обчислення гессиана. Для того, щоб уникнути обчислення матриці Гессе, пропонуються різні способи її заміни наближеними виразами, що породжує так звані квазіньютонівські алгоритми такі як алгоритм, запропонований Бройлером, Флетчером, Гольдфарбом і Шанно, однокроковий алгоритм методу січних площин OSS, алгоритм LM Левенберга-Марквардта і ін.

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

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

а градієнт розрахований по формулі

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

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

$\mathbf{e}$ – вектор помилки мережі.

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

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

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

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

ДОДАТОК В

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ

В.1 Моделі систем, використовувані при нейромережевій ідентифікації

Основним питанням в нейрорегулювання є отримання адекватних моделей динамічних систем. Питання про існування дискретних моделей, або іншими словами про спостереженість нелінійних систем, є ключовим. У загальному випадку завдання побудови математичної моделі об'єкту полягає у виборі структури і оцінці її параметрів так, щоб при використанні критерію мінімуму деякої функції різниці розрахункових і експериментальних даних дотримувалася умова близькості моделі досліджуваному процесу. Застосування нейронних мереж як моделі може служити альтернативою класичним методам математичного опису систем, оскільки в цьому випадку точне знання внутрішніх процесів не є необхідною умовою моделювання.

При нейронній ідентифікації нелінійних систем найчастіше знаходять застосування наступні основні нелінійні дискретні моделі.

У найбільш загальному вигляді, прийнятому в класичній теорії управління, опис стохастичних нелінійних систем в просторі станів розглядався в роботі Наренди К.С. і Партасараті К [50].

$$z(k+1) = \Psi[z(k), u(k), \phi(k)] ;$$

$$y(k) = \Phi[z(k), u(k), \phi(k)],$$

де $z(k) = [z_1(k), z_2(k), \dots, z_{k_z}(k)]^T$ – вектор стану системи;

$u(k) = [u_1(k), u_2(k), \dots, u_{k_u}(k)]^T$ – вектор вхідних сигналів;

$y(k) = [y_1(k), y_2(k), \dots, y_{k_y}(k)]^T$ – вектор вихідних сигналів;

$\Psi[\cdot]$, $\Phi[\cdot]$ – деякі статичні нелінійні функції;

$\phi(k)$, (k) – шум процесу і шум вимірювань відповідно.

Леонартіс І. Дж. і Биллінгз С.Д.[51] запропонували для опису нелінійних систем використовувати так звану нелінійну модель авторегресії ковзаючого середнього з додатковими вхідними сигналами (NARMAX-модель).

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), e(k-1), \dots, e(k-n_e)] + e(k) \quad (\text{B.1})$$

де $y(k) \in \mathbf{R}^{M \times 1}$ – вихідний сигнал об'єкту;

$u(k) \in \mathbf{R}^{N \times 1}$ – вхідний сигнал об'єкту;

$e(k) \in \mathbf{R}^{M \times 1}$ – різниця вихідних сигналів об'єкту і моделі;

$f[\cdot]$ – нелінійна функція перетворення $f: \mathbf{R}^{(k_y M + k_u N + k_e M) \times 1} \rightarrow \mathbf{R}^{M \times 1}$;

n_y, n_u, n_e – порядки запізнювання по вихідному, вхідному сигналам об'єкту і помилці вимірювань відповідно.

Скалярна форма рівняння (Г.1) має вигляд:

$$\begin{aligned} y_1(k) = f_1[y_1(k-1), \dots, y_1(k-n_y), y_2(k-1), \dots, y_2(k-n_y), y_M(k-1), \dots, y_M(k-n_y), \\ u_1(k-1), \dots, u_1(k-n_u), u_2(k-1), \dots, u_2(k-n_u), u_N(k-1), \dots, u_N(k-n_u), \\ e_1(k-1), \dots, e_1(k-n_e), e_2(k-1), \dots, e_2(k-n_e), e_M(k-1), \dots, e_M(k-n_e)] + e_i(k) \end{aligned} \quad (\text{B.2})$$

Нелінійні системи можуть бути представлені нелінійною NARX-моделлю:

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)] + e(k) \quad (\text{B.3})$$

або в скалярному вигляді

$$\begin{aligned} y_1(k) = f_1[y_1(k-1), \dots, y_1(k-k_y), y_2(k-1), \dots, y_2(k-k_y), y_M(k-1), \dots, y_M(k-k_y), \\ u_1(k-1), \dots, u_1(k-k_u), u_2(k-1), \dots, u_2(k-k_u), u_N(k-1), \dots, u_N(k-k_u)] + e_i(k). \end{aligned} \quad (\text{B.4})$$

Представлення об'єктів у вигляді NARMAX або NARX моделей грають фундаментальну роль при дослідженні нелінійних об'єктів за допомогою штучних нейронних мереж. При цьому важливо розглянути узагальнення моделей (B.1) - (B.4), що враховує різні види нелінійностей по вхідних і вихідних змін-

них. Наренда виділив чотири такі моделі:

– модель 1 (рис. В.1):

$$y(k+1) = \sum_{i=1}^{k_y-1} a_i y(k-i) + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\text{B.5})$$

– модель 2 (рис. В.2):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + \sum_{i=1}^{k_u-1} b_i u(k-i); \quad (\text{B.6})$$

– модель 3 (рис. В.3):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\text{B.7})$$

– модель 4 (рис. В.4):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (\text{B.8})$$

Модель (В.4) є найбільш загальною зі всіх моделей вхід/вихід, описаних вище. При цьому модель (В.8) є аналітично вирішуваною і найбільш зручною для практичних застосувань. Внаслідок цього вона використовується в роботі при ідентифікації нелінійних об'єктів за допомогою штучних нейронних мереж.

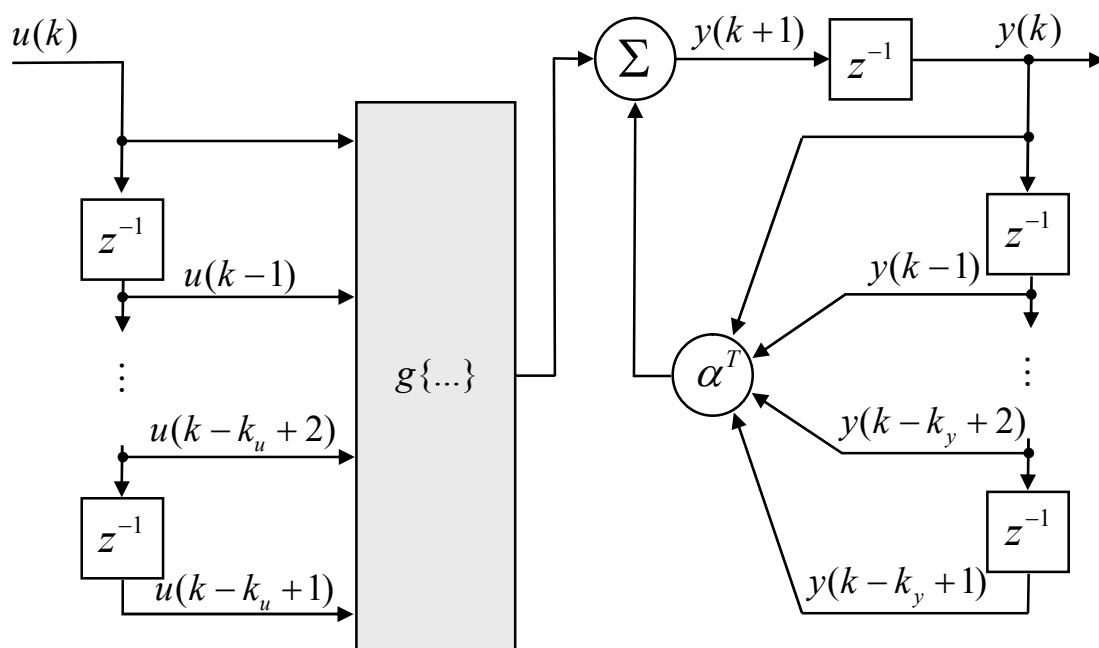


Рисунок В.1 – Модель 1. Вихід моделі нелінійно залежить від вхідних сигналів і лінійно від вихідних

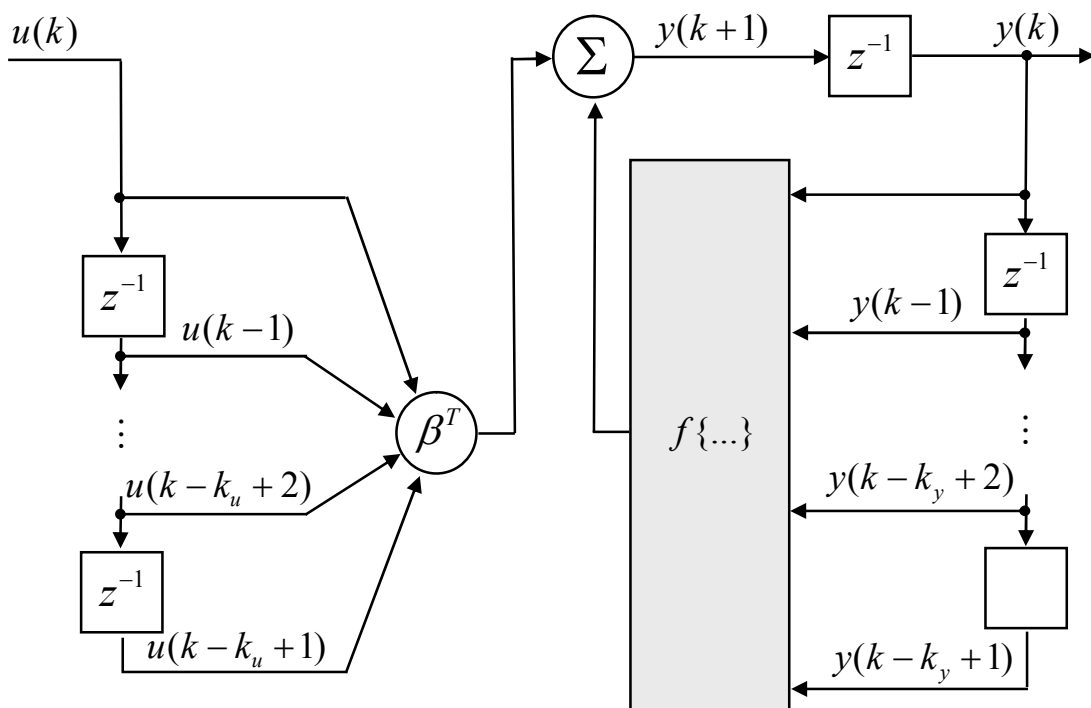


Рисунок В.2 – Модель 2. Вихід моделі лінійно залежить від вхідних сигналів і нелінійно від вихідних

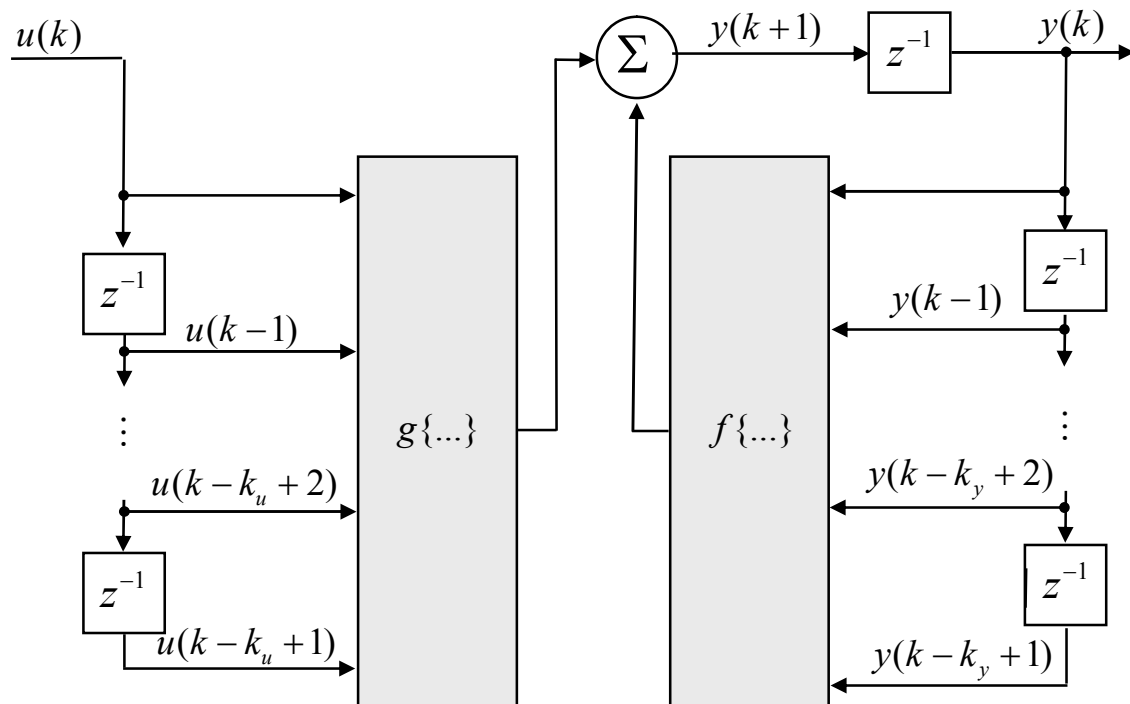


Рисунок В.3 – Модель 3. Резульгуючий сигнал є адитивною функцією нелінійних перетворень вхідних і вихідних сигналів

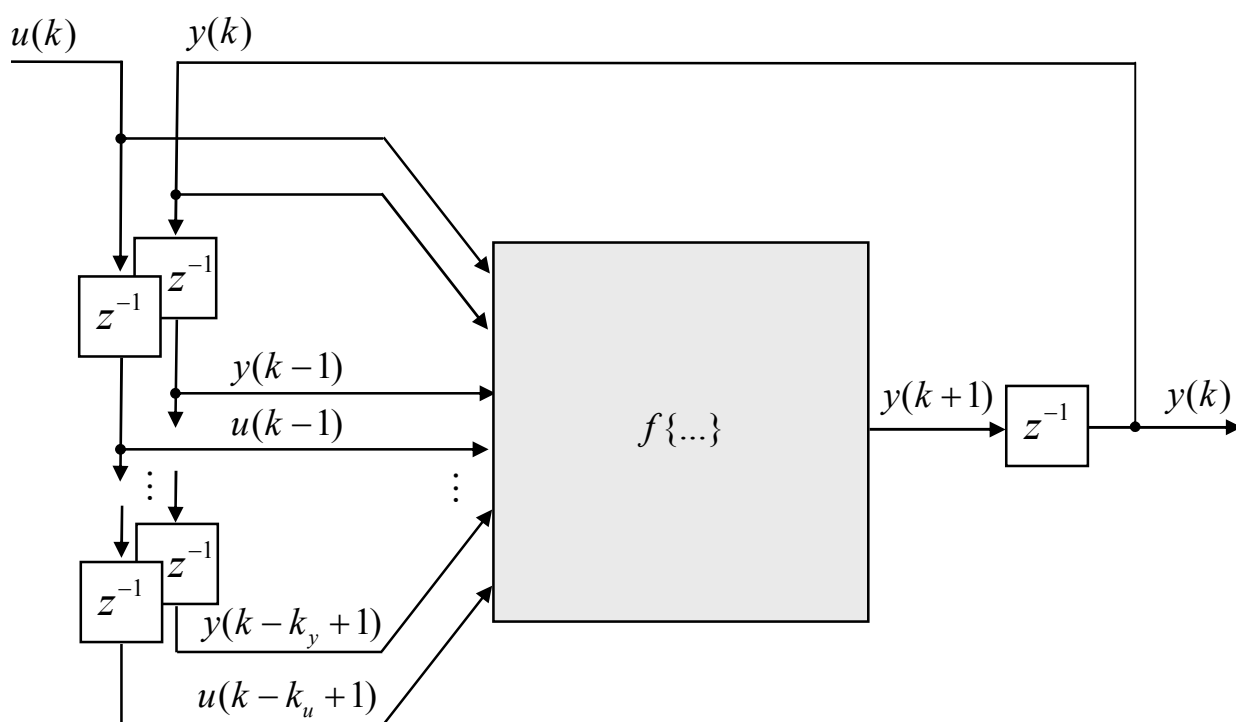


Рисунок В.4 – Модель 4. Вихід моделі нелінійно залежить як від вхідних, так і від вихідних сигналів

В.2 Нейронні мережі як універсальні моделі

Оскільки тип і взаємозв'язок нелінійностей в об'єкті, як правило, не відомі, для моделювання слід використовувати якомога більш універсальні структури моделей. Нейронні мережі є саме такими структурами.

Нелінійні моделі в просторі станів можуть бути реалізовані за допомогою рекурсивних нейронних мереж з внутрішніми зворотними зв'язками (мережі Елмана). Ідентифікація параметрів цих мереж вимагає спеціальної і щодо трудомісткої форми алгоритму навчання. Крім того, доказ стійкості таких нелінійних динамічних моделей практично неможливо зробити.

У роботі використовуються прямонаправленні мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямі без зворотних зв'язків усередині мережі, – багат шаровий персептрон. Найбільш часто використовується мережа, багат шаровий персептрон (БП). За допомогою БП можна апроксимувати з бажаною точністю не тільки будь-які статичні функції. Попередні значення вхідних/вихідних координат у вхідному векторі дозволяють додати прямонаправленим мережам динамічні властивості.

Володіючи схожими з радіально-базисними мережами (РБМ) властивостями, БП забезпечує компактніше представлення нелінійного об'єкту, оскільки число параметрів в мережі РБМ експоненціально зростає із збільшенням розмірності вхідного простору, тоді як для БП ця залежність лінійна. Важливою є наочність, що виражається в тому, що персептронне представлення моделі еквівалентне традиційному. Це дозволяє використовувати для визначення параметрів ШНМ добре розвинені методи оцінювання, використовувані при традиційному підході.

Модель системи в просторі станів може бути успішно реалізована за допомогою прямонаправлених мереж, якщо змінні стани сформувати на виході мережі і потім завести їх на вхід мережі із затримкою. Але для цього потрібний, щоб всі змінні стани були вимірювані, оскільки тільки в цьому випадку можна забезпечити тренування мережі.

NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при нейронній ідентифікації систем. Для простоти викладу надалі розглядаємо системи з одним входом і одним виходом SISO (singl input singl output), хоча всі висновки можуть бути поширені і на MIMO (multi input multi output) системи. Прямий нейронний NARX моделі нелінійної динамічної системи відповідає наступний вираз:

$$\hat{y}(k) = f\left(y(k-1)\dots y(k-n_y), u(k-n_k)\dots u(k-n_u-n_k+1), \mathbf{w}\right), \quad (\text{B.9})$$

а нейронній інверсній NARX моделі:

$$\hat{u}(k) = g\left(y(k+n_k)\dots y(k+n_k-n_y), u(k-1)\dots u(k-n_u+1), \mathbf{w}\right) \quad (\text{B.10})$$

де f, g – функції, що реалізуються нейронними мережами;

u, y – вхідна і вихідна координати відповідно;

$\hat{u}, \hat{y}$ – значення координат на виході мережі;

$\mathbf{w}$ – матриця лінійних параметрів мережі;

n_k – запізнювання в системі, найчастіше $n_k = 1$.

Вирази (B.9) і (B.10) можна представити графічно у вигляді структурних схем, показаних на рис. B.5 і рис. B.6. NARMAX моделі характеризуються вищою якістю ідентифікації при зашумлених тренувальних даних. У [52] пропонується спосіб модифікації критерію якості алгоритму тренування. Результати ідентифікації за допомогою NARX моделей наближаються в цьому випадку до результатів, отриманих за допомогою NARMAX моделей. Дані NARX нейромоделі можуть бути успішно застосовані для вирішення поставлених завдань.

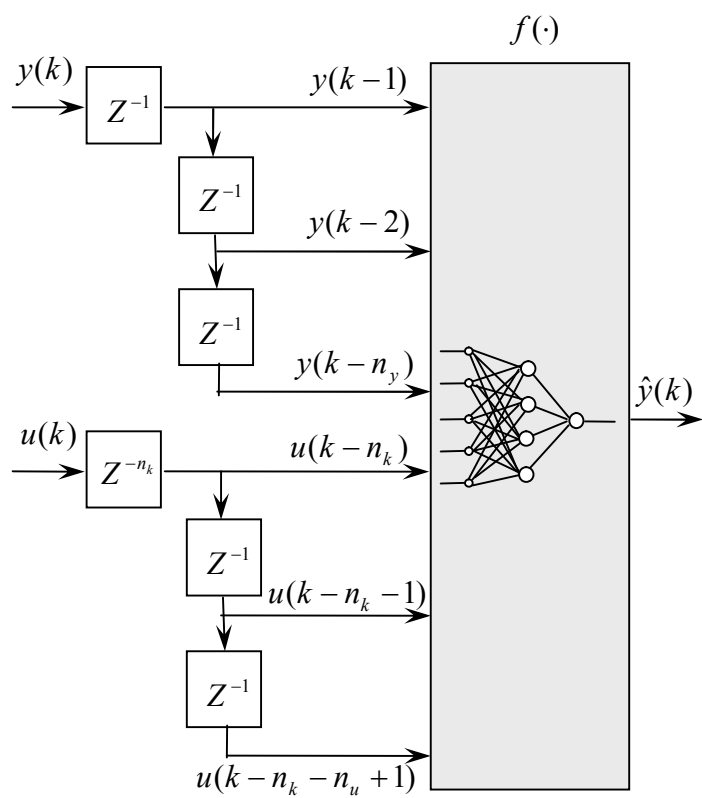


Рисунок В.5— Пряма мережева NARX модель

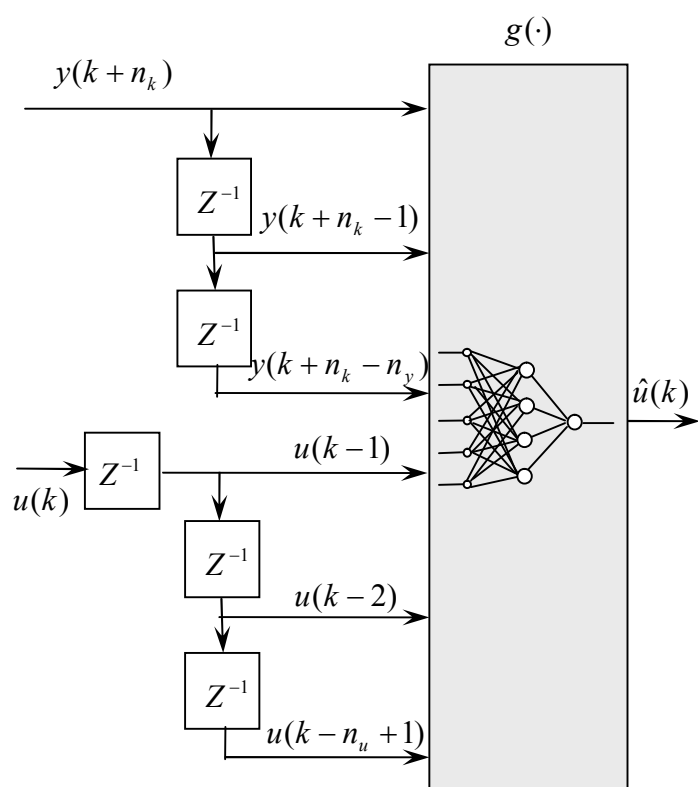


Рисунок В.6 – Інверсна мережева NARX модель

В.3 Ідентифікація динамічних систем за допомогою нейронних мереж

Завдання теорії регулювання, що займається пошуком моделей динамічних систем на основі апріорної інформації і вимірюваних входних/вихідних сигналів системи, носить назву ідентифікація. Відомі класичні методи ідентифікації нелінійних систем застосовні тільки у поєднанні з певним типом моделі (наприклад, ряди Вальтера, моделі Хаммерштейна і Вінера), що описує певний клас нелінійних динамічних систем. У роботі під термінами «ідентифікація системи», «тренування» і «навчання» розуміється процес підстроювання вагових коефіцієнтів мережі за допомогою градієнтних методів. Метою цього процесу є по можливості точне наближення поведінки моделі до спостережуваної поведінки динамічної системи.

Принципово розрізняють пряму і інверсну модель об'єкту. На рис. В.7 представлені схеми тренування цих нейронних моделей. Для спрощення на рис. В.7 не зображені ланки затримки за часом, що формують попередні значення координат на вході нейронних мереж.

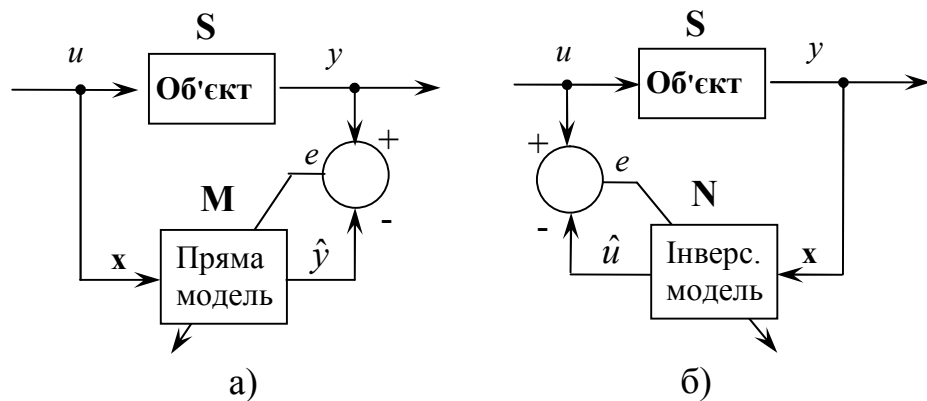


Рисунок В.7 – Схеми тренування прямої (а) і інверсної (б) моделей

Вхідний сигнал x нейронній моделі об'єкту M на рис. В.7,а ідентичний вхідному сигналу об'єкту S . Для інверсної нейромоделі N як вхідний сигнал використовується вихідний сигнал об'єкту S тоді як нейромодель повинна відновити вхідний сигнал об'єкту (див. рис. В.7,б). Поки присутня помилка між виходом нейромоделі і її зразковим сигналом, ця помилка використовується для

тренування мережі. У обох випадках ідентифікації обчислюється градієнт функціонала якості (див. таблицю В.1). Функціонал якості найчастіше при цьому квадратичний.

Таблиця В.1 – Обчислення градієнта при тренуванні нейромоделей

	Пряма модель	Інверсна модель
Бажаний вихідний сигнал	y	u
Вихідний сигнал нейромоделі	$\hat{y} = \mathbf{M}(\mathbf{x}, \mathbf{w})$	$\hat{u} = \mathbf{N}(\mathbf{x}, \mathbf{w})$
Помилка навчання	$e = y - \hat{y}$	$e = u - \hat{u}$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{y}}{\partial \mathbf{w}}$	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{u}}{\partial \mathbf{w}}$

Оскільки $\frac{\partial \hat{y}}{\partial \mathbf{w}}$ і $\frac{\partial \hat{u}}{\partial \mathbf{w}}$ обчислювані, то це дозволяє вести підстроювання вагових коефіцієнтів мережі згідно розглянутим градієнтним методам. Результатом цього буде мінімізація функціонала якості. Оскільки набір тренувальних даних однозначно відображає лише поведінку модельованого об'єкту $\mathbf{S}$ у робочому діапазоні, то після завершення тренування маємо $\mathbf{M} \approx \mathbf{S}$ і $\mathbf{N} \approx \mathbf{S}^{-1}$.

Типова послідовність дій, що приводить до формування нейромоделі (ідентифікації) динамічної системи, відображена в таблиці В.2.

Таблиця В.2 – Ідентифікація за допомогою нейронних мереж

Шаг	Дія
1	Знімання тренувальних даних
2	Вибір структури мережі
3	Вибір типу моделі
4	Тренування мережі
5	Тестування нейромоделі
6	При незадов. результаті тестування, перейти до п.3 або до п.2

Оскільки тренувальні дані є носієм інформації про систему, що ідентифікується, то наявності, кількості і якості цих даних слід приділити особливу увагу. Тренувальних дані повинні мати наступні характеристики: якісна дискретизація, рівномірність дисперсії і величини варіації, рівномірність амплітуд всіх сигналів і їх середньоквадратичних значень.

При отриманні позитивного результату апроксимації, слід також перевірити здібність нейромоделі до узагальнення, тобто її здатність повторювати істотні характеристики об'єкту. Для цього тесту необхідний ще один набір даних, що не брав участь при тренуванні мережі. Цей тест дає остаточний висновок про адекватність нейромоделі.

Представлені теоретичні положення по нейронній ідентифікації, а також характеристики прямонаправлених мереж є основою успішного регулювання нелінійних динамічних систем.

ДОДАТОК Г

ФАЙЛ ВИХІДНИХ ДАНИХ

```
% Система управління електроприводом повороту  
% стріли роторного екскаватора  
% 2024 рік
```

```
% Вихідні дані
```

```
kzv=1.84;  
ks=4.9;  
Tg=0.5;  
Tzc=0.01;  
Te=0.054;  
Re=0.11;  
CF=2*3.58;  
In=338;  
Mn=CF*In;  
Jd=2*8.25;  
Jm=41.5;  
b12=0.02;  
b12=0;  
c12=50;  
Js=Jd+Jm;  
Uz=202.3;  
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);
```