

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 РОЗВОРОТУ
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)

_____/ Іван ЯКУБОВ
(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) Іван ЯКУБОВ
(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 розвороту 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 $F_{\text{б}}=1,2 \cdot 10^6 \text{ Н}$; 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 $\omega_{\text{пл}}=2,9 \cdot 10^2 \text{ с}^{-1}$; маса поворотної платформи $m_{\text{пл}}=500 \cdot 10^3 \text{ кг}$; маса приймальної частини $m_{\text{пч}}=127 \cdot 10^3 \text{ кг}$; маса відвальної частини $m_{\text{пч}}=120 \cdot 10^3 \text{ кг}$; частота обертання ротора $\omega_p=6 \text{ рад/хв}$; момент інерції поворотної платформи $J_{\text{пл}}=17 \cdot 10^7 \text{ кг} \cdot \text{м}^2$; діаметр роликів $D_{\text{кр}}=8,7 \text{ м}$. Перерегулювання не більше 10%, час регулювання до 5 с.

4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму розвороту роторного екскаватора. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити статичні і динамічні характеристики системи. Розробити математичну модель двомасової системи. Виконати моделювання на ЕОМ та провести аналіз показників якості перехідних процесів змінних стану двомасової системи. Розробити структурну схему нейромережевої системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання нейромережевої 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. Системи нейрорегулювання. Структурні схеми. 6. Управління з прогнозом. Математична модель. 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)

Іван ЯКУБОВ
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про вико- нання за- вдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму розвороту роторного екскаватора.	09.09.2024- 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)

Іван ЯКУБОВ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 216 сторінок, 68 рисунків, 3 таблиці, 78 використаних джерел, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму розвороту 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 розвороту 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 проекту, 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 РОЗВОРОТУ, СИСТЕМА УПРАВЛІННЯ З ПІДСУМОВУЮЧИМ ПІДСИЛЮВАЧЕМ, ОДНОМАСОВА І ДВОМАСОВА СИСТЕМИ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОРЕГУЛЯТОР

ABSTRACT

An explanatory message consists of: 216 pages, 68 figures, 3 tables, 78 used sources, 5 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the 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 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 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 СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Розвиток теорії нейронних мереж	14
1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж	20
1.3 Нейрокерування динамічними об'єктами	24
1.4 Огляд структур нейромережових систем управління.	
Вибір структури системи	27
1.5 Загальні відомості про роторні екскаватори	34
1.6 Аналіз факторів, які обумовлюють вибір системи управління електроприводом механізму розвороту роторного екскаватора.....	38
Нейромережовий підхід до управління.	
1.7 Постановка задачі синтезу нейромережової системи управління	41
ВИСНОВКИ ДО РОЗДІЛУ 1	45
РОЗДІЛ 2. РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ РОЗВОРОТУ РОТОРНОГО ЕКСКАВАТОРА	
2.1 Основні технічні характеристики роторного екскаватора.	
Вимоги до системи управління.....	47
2.2 Розрахунок потужності і вибір електродвигунів механізму розвороту екскаватора	49
2.3 Розрахунок швидкісної діаграми механізму розвороту	50
2.4 Побудова діаграми навантаження	54
2.5 Перевірка двигунів по нагріву і перевантажувальній здатності.	60
2.6 Вибір основного електроустаткування електроприводу	61
2.7 Розрахунок статичних характеристик розімкненої і замкненої систем	64
2.10 Дослідження динамічних характеристик системи	74

ВИСНОВКИ ДО РОЗДІЛУ 2	88
РОЗДІЛ 3 ХАРАКТЕРИСТИКИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ.....	89
3.1 Математична модель двомасової системи	89
3.2 Розрахунок перехідних процесів двомасової системи.....	95
ВИСНОВКИ ДО РОЗДІЛУ 3	101
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	102
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	102
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	104
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	107
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	150
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	152
ВИСНОВКИ ДО РОЗДІЛУ 4	158
ВИСНОВКИ	160
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	162
ДОДАТОК А. БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ.....	169
А.1 Класифікація штучних нейронних мереж	169
А.2 Базовий процесорний елемент (штучний нейрон).....	171
А.3 Статичні багатошарові нейронні мережі	173
ДОДАТОК Б. ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	178
Б.1 Здібність до навчання і накопичення інформації.....	178
Б.2 Апроксимаційні властивості нейронних мереж	178
Б.3 Апроксимація і пам'ять в багатошарових нейронних мереж	181
ДОДАТОК В. НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	185

В.1 Загальна характеристика алгоритмів навчання.....	185
В.2 Метод і алгоритм зворотного розповсюдження помилки.....	188
В.3 Перенавчання і узагальнення нейронних мереж.....	199
ДОДАТОК Г. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ	
З ПРОГНОЗОМ	202
Г.1 Алгоритм нейроуправління з прогнозомі	202
Г.2 Структура нейронної мережі для моделювання об'єкту.....	209
Г.3 Прогноз з використанням нейронної мережі.....	213
Г.4 Похідні рівнянь нейронної мережі.....	214
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	216

ВСТУП

Актуальність теми. На сучасному етапі гірничодобувна промисловість активно використовує відкритий метод видобутку корисних копалин, який стрімко розвивається. Ключовим елементом технологічного процесу цього методу є роторний екскаватор. Ці машини відзначаються великою продуктивністю. Рух роторного екскаватора та транспортування видобутої породи чи корисних копалин забезпечуються рядом механізмів: багатоковшовим роторним колесом, що видаляє ґрунт; механізмом підйому стріли з роторним колесом; поворотним пристроєм, який дозволяє повертати стрілу з роторним колесом під час роботи; системою ходу, що переміщує екскаватор вздовж робочого місця; та транспортною системою - стрічковими конвеєрами, які перекидають породу до головного конвеєра або на звалище та інші. Індивідуальний електропривід цих механізмів дозволяє автоматизувати операції, забезпечити автоматичний контроль над процесом, уникнути заторів, полегшити робочий процес і підвищити його продуктивність.

Підвищення продуктивності праці на відкритих розробках ґрунтується на механізації та автоматизації всіх основних і допоміжних процесів, збільшенні потужності гірничих машин та агрегатів, а також переході від розробки й впровадження окремих механізмів до створення комплексів машин, які охоплюють весь технологічний процес.

Комплексна механізація забезпечує необхідні умови для переходу до більш сучасної форми організації виробництва — автоматизації.. У цьому варіанті робота з керуванням машинами передається управляючим системам, тоді як людина залишається у творчому плані діяльності, відповідаючи за планування та контроль роботи цих систем.

Протягом останніх десятиліть було висунуто багато методів управління, зокрема, техніки адаптивного керування. Це стало можливим завдяки прогресу у сучасній теорії управління та стійкості. Здатність автоматично адаптуватися до змін у динаміці управляемого об'єкту та оточуючого середовища зробила

адаптивні системи керування все більш привабливими для різноманітних практичних застосувань. Однак ефективність таких систем керування реальними об'єктами в значній мірі залежить від якості математичних моделей, що використовуються у процесі. Ці моделі мають якнайточніше відображати характеристики вивченого об'єкту, але водночас бути зручними для використання алгоритмів управління. Неясність у відомостях про умови функціонування та характеристики об'єктів і завад створює необхідність застосування при управлінні адаптивного підходу, який може використовувати спрощені моделі, у тому числі лінійні. Хоча цей підхід і може в певних випадках значно знизити початкову неоднозначність і забезпечити ефективне керування, обмеження, пов'язані з використанням лінійних моделей, не завжди забезпечують потрібний результат. Отже, більш ефективним може бути розвиток систем управління, які базуються на адаптивному підході, сполученому з методами теорії штучних нейронних мереж.

Нейромережеве управління засноване на застосуванні повністю визначених штучних нейронних мереж для вироблення необхідних сигналів управління. Високий інтерес до даного напрямку в теорії управління визначається наступними чинниками [1-12].

Штучні нейронні мережі є адаптивними системами, здатними навчатися рішенням складних задач. Необхідною умовою якісного навчання є правильний вибір структури нейронної мережі і достатній об'єм інформації, використовуваний при навчанні.

Нейромережеві технології управління дають змогу ефективно вирішувати проблеми, які виникають при роботі з нелінійними об'єктами або об'єктами з невідомою структурою, що є складними або неможливими для розв'язання традиційними методами адаптивного управління. Можливість штучних нейронних мереж реалізовувати складні нелінійні залежності зумовлена використанням сигмоїдальних функцій активації (або інших нелінійних функцій загального типу) у нейронах прихованих шарів.

Здатність штучних нейронних мереж до самонавчання дозволяє застосовувати нейрорегулятори навіть за умов значної невизначеності, тоді як традиційні методи адаптивного управління вимагають великого обсягу апіорної інформації про об'єкт керування.

Висока швидкодія та надійність нейрорегуляторів пояснюються значним ступенем паралельності обчислень у штучних нейронних мережах. Їхня простота реалізації та здатність до навчання роблять нейронні мережі особливо привабливими для управління складними нелінійними об'єктами в режимі реального часу.

Використання нейромережевих технологій надає величезні можливості для пошуку ефективних рішень задач, багато з яких раніше не мали рішення із задовільною точністю. Тому тема роботи, яка присвячена розробці нейромережевої системи управління електроприводом механізму повороту стріли роторного екскаватора з урахуванням пружності механічних зв'язків платформи і інерційних мас стріли ротора з роторним колесом і консолі противаги, є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму розвороту 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 електроприводом механізму розвороту роторного екскаватора з урахуванням пружних елементів.

Предметом дослідження є моделі і методи синтезу нейромережевої системи управління двомасовою електромеханічною системою механізму розвороту роторного екскаватора.

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 стали основою синтезу нейромережевої системи. Імітаційне моделювання дозволило проаналізувати якість функціонування систем і 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 нейромережевих технологій для управління багатомасовою електромеханічною системою механізму розвороту роторного екскаватора.

2. Вперше запропоновано і синтезовано нейромережеву систему управління електроприводом механізму розвороту роторного екскаватора з урахуванням пружних елементів, що має високі показники якості функціонування.

3. Вперше запропонована математична модель системи управління електроприводом механізму розвороту 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 нейромережевих структур з прогнозом для управління складними динамічними об'єктами з математичними моделями у вигляді двомасових електромеханічних систем.

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

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.

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 нейромережевої 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; виконано економічне обґрунтування автоматизованої системи управління; розроблено заходи забезпечення безпеки праці при виконанні робіт з обслуговування електромеханічної системи управління механізмом розвороту роторного екскаватора.

В роботі наведені посилання на авторів і 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 216 сторінок, серед них: 68 рисунків, з яких 47 рисунків за текстом, 21 рисунок на 21 окремій сторінці, 3 таблиці, з яких 1 на 2 повних сторінках, 5 додатків на 48 сторінках, список літературних джерел з 78 найменування на 7 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Розвиток теорії нейронних мереж

1.1.1 Історичний аспект

Нейробіологи та нейроанатоми досягли значного прогресу у вивченні роботи головного мозку. Ретельно досліджуючи структуру та функції нервової системи людини, вони багато дізналися про "електропроводку" мозку, однак його повне функціонування залишається маловивченим. У процесі накопичення знань стало зрозуміло, що мозок має неймовірну складність: сотні мільярдів нейронів, кожен із яких з'єднаний із сотнями чи тисячами інших, формують систему, яка значно перевершує найсміливіші мрії про суперкомп'ютери.

Поглиблене розуміння роботи нейронів і їхніх зв'язків дозволило дослідникам розробити математичні моделі для перевірки своїх теорій. Завдяки цьому експерименти тепер можна проводити на цифрових комп'ютерах, що допомагає вирішити багато практичних проблем. Було виявлено, що ці моделі не лише імітують функції мозку, але й здатні виконувати завдання, які мають самостійну цінність.

Таким чином, у сфері нейронного моделювання сформувалися дві взаємодоповнюючі цілі: перша - зрозуміти роботу нервової системи людини на рівні фізіології та психології, і друга - створити обчислювальні системи (штучні нейронні мережі), які виконують функції, подібні до функцій мозку.

Паралельно з розвитком нейроанатомії та нейрофізіології психологи розробляли моделі людського навчання. Однією з найуспішніших стала модель Д. Хебба, який у 1949 році запропонував закон навчання, що став основою для створення алгоритмів навчання штучних нейронних мереж. Хоча сьогодні цей підхід значно доповнений іншими методами, він наочно продемонстрував, як нейронна мережа може навчатися.

У 1950–1960-х роках група дослідників, об'єднавши біологічні та

фізіологічні підходи, створила перші штучні нейронні мережі. Спочатку вони були реалізовані у вигляді електронних схем, але згодом перенесені у середовище комп'ютерного моделювання, яке використовується й нині. Перші успіхи викликали хвилю активності та оптимізму. Такі вчені, як Мінський, Розенблатт, Уїдроу та інші, розробили одношарові нейронні мережі, відомі як персептрони. Вони застосовувалися для розв'язання широкого кола завдань, таких як прогнозування погоди, аналіз електрокардіограм і моделювання штучного зору. У той час здавалося, що ключ до створення інтелекту знайдено, і побудова мозкоподібної системи - лише питання створення достатньо великої мережі.

Однак цей оптимізм швидко згас. Мережі не могли вирішувати завдання, схожі за складністю з тими, які вони вже успішно виконували. Ці невдачі спричинили період інтенсивного аналізу. Мінський, застосовуючи точні математичні методи, строго довів низку теорем, що стосувалися роботи нейронних мереж, і пояснив обмеження їх можливостей.

Дослідження Мінського разом із Папертом [24] привели до висновку, що одношарові нейронні мережі, які використовувалися на той час, теоретично не здатні вирішувати низку простих задач, зокрема реалізовувати функцію «Виключне АБО». Мінський також скептично оцінював можливість суттєвого прогресу в цій галузі.

Чіткість та строгість аргументації Мінського здобули широку довіру, і його висновки сприймалися як незаперечні. Це викликало розчарування серед дослідників, багато з яких залишили дослідження штучних нейронних мереж заради перспективніших напрямків. На майже два десятиліття ця галузь науки була фактично забута.

Однак кілька наполегливих науковців, таких як Кохонен [25], Гроссберг [26,27] і Андерсон [29], продовжували свої дослідження. Поступово було створено теоретичний базис, який став основою для сучасних потужних багатошарових мереж. Оцінка Мінського виявилася надмірно песимістичною: багато задач, які вважалися нерозв'язними на той час, тепер вирішуються за допомогою багатошарових мереж і стандартних алгоритмів.

У 1980-х роках теорія нейронних мереж почала активно застосовуватися

в прикладних сферах, і з'явилися нові компанії, що займалися комерціалізацією цієї технології. Інтерес до штучних нейронних мереж зростає вибуховими темпами. У 1987 році було проведено чотири великих конференції, присвячені нейронним мережам, і опубліковано понад 500 наукових праць, що свідчило про феноменальний розвиток галузі.

1.1.2 Сучасний стан штучних нейронних мереж

Сьогодні існує багато вражаючих прикладів можливостей штучних нейронних мереж. Наприклад, мережу навчили перетворювати текст у фонетичне представлення, яке потім за допомогою інших технологій конвертується в мову [30]. Інша нейронна мережа успішно розпізнає рукописні літери [31]. Також розроблена система стиснення зображень, яка базується на використанні нейронних мереж [32]. Усі ці досягнення реалізовані за допомогою мереж зі зворотним розповсюдженням помилки — одного з найуспішніших сучасних алгоритмів навчання.

Метод зворотного розповсюдження, запропонований незалежно в трьох різних дослідженнях [33–35], є систематичним підходом до навчання багатошарових нейронних мереж і долає обмеження, на які вказував Мінський.

Проте метод зворотного розповсюдження має свої недоліки. Зокрема, немає гарантії, що навчання мережі завершиться за кінцевий час. Часто значні обчислювальні ресурси витрачаються безрезультатно, і спроби навчання доводиться повторювати, без жодної впевненості в покращенні результату. Крім того, алгоритм може потрапити в так звані локальні мінімуми, через що отримане рішення може бути далеким від оптимального.

Розроблено багато інших алгоритмів навчання для нейронних мереж, кожен з яких має свої унікальні переваги. Проте жодна із сучасних мереж не є універсальним рішенням: усі вони мають обмеження у здатності до навчання та відтворення.

Ця галузь вже довела свою ефективність і демонструє унікальні можливості, проте залишається багато обмежень і нерозв'язаних питань. Для покращення існуючих мереж потрібні серйозні зусилля. Необхідно розробляти

нові технології, вдосконалювати методи навчання та зміцнювати теоретичну базу, щоб нейронні мережі змогли повністю реалізувати свій потенціал.

1.1.3 Перспективи розвитку

Штучні нейронні мережі пропонуються для вирішення широкого спектра завдань. Вони особливо перспективні у тих областях, де людський інтелект малоефективний, а традиційні обчислювальні методи є занадто складними або неадекватними. Такий клас застосувань не поступається за обсягом і важливістю задачам, які вирішуються звичайними обчисленнями. Можна прогнозувати, що в майбутньому штучні нейронні мережі займатимуть своє місце поряд із традиційними методами, доповнюючи їх у рівноцінному обсязі й значущості.

Штучні нейронні мережі та експертні системи

Упродовж останніх років у сфері штучного інтелекту домінували логічні та символічно-операційні підходи. Наприклад, експертні системи, які здобули помітні успіхи, проте не позбавлені недоліків, отримали значну популярність. Існує думка, що штучні нейронні мережі можуть витіснити традиційний штучний інтелект. Проте ймовірніше, що ці підходи будуть співіснувати, доповнюючи один одного в межах інтегрованих систем, де кожен з них вирішуватиме ті завдання, для яких він найбільш ефективний.

Ця точка зору знаходить підтвердження в тому, як функціонує людський мозок. Розпізнавання образів відповідає за швидку реакцію, необхідну в умовах, що вимагають миттєвих дій. Оскільки такі реакції здійснюються швидко й автоматично, вони є критично важливими для виживання у складних або ворожих умовах.

Коли система розпізнавання образів не може дати адекватної відповіді, завдання передається вищим відділам мозку. Ці відділи можуть запитати додаткову інформацію, витратити більше часу, але забезпечити більш точне та обґрунтоване рішення.

Можна уявити штучну систему, що наслідує такий принцип поділу праці. Штучна нейронна мережа може забезпечувати швидкі реакції в більшості стан-

дартних ситуацій. Завдяки здатності вказувати рівень впевненості у своїх рішеннях, така мережа може «усвідомлювати», коли їй бракує інформації, і передавати складніші завдання експертній системі.

На цьому рівні рішення будуть конкретними та логічними, хоча можуть вимагати додаткових даних для формулювання остаточного висновку. Комбінація нейронних мереж і експертних систем створить більш потужний інструмент, ніж кожна з цих технологій окремо, наслідуючи високоефективний принцип, закладений у процесі біологічної еволюції.

Питання надійності

Перш ніж використовувати штучні нейронні мережі в критичних сферах людської діяльності, необхідно вирішити ключові питання, пов'язані з їхньою надійністю.

Як і людський мозок, структурою якого вони надихаються, штучні нейронні мережі мають певний рівень непередбачуваності. Єдиний спосіб точно визначити їхній вихід полягає у тестуванні всіх можливих вхідних сигналів. Однак для великих мереж така повна перевірка практично неможлива, тому використовуються статистичні методи для оцінки їхнього функціонування. У певних сферах це неприйнятно. Наприклад, який допустимий рівень помилок для мережі, що управляє системою космічної оборони? Жодна помилка не може бути дозволена, адже вона може спричинити катастрофічні наслідки, що включають величезну кількість жертв і руйнувань. Ця проблема не стає менш актуальною від усвідомлення, що людина в подібних ситуаціях також може припускатися помилок.

Корінь проблеми лежить у загальному уявленні про безпомилковість комп'ютерів. Оскільки штучні нейронні мережі іноді допускають помилки навіть при коректному функціонуванні, це породжує сумніви щодо їхньої надійності. Для багатьох така особливість є неприйнятною, адже ми звикли вимагати абсолютної точності від наших машин.

Схожа проблема полягає у тому, що традиційні штучні нейронні мережі не можуть "пояснити" свій процес прийняття рішень. Внутрішні уявлення, що

формується в результаті навчання, часто настільки складні, що їхній аналіз можливий лише в простих випадках. Це нагадує нашу нездатність описати, як ми впізнаємо людину, незважаючи на зміну відстані, освітлення, обстановки та часу. Експертні системи можуть відслідковувати процес своїх міркувань у зворотному порядку, що дозволяє людині перевірити їхню логічність. Зазначалося, що інтеграція такої здатності в штучні нейронні мережі [36] може значно підвищити прийнятність цих систем.

Отже, штучні нейронні мережі розширюють поняття обчислювальних систем і призводять до створення автоматів, які виконують завдання, раніше доступні лише людині. Вони можуть обробляти монотонні, небезпечні та рутинні завдання, і з розвитком технологій з'являться нові області застосування. Теорія штучних нейронних мереж розвивається дуже швидко, що веде до створення більш досконалих мережевих парадигм і численних прикладних можливостей.

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

У даній роботі для синтезу нейромережевої 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, а в додатку Б - структуру багат шарових мереж.

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

В додатку Г розглянуто ефективну реалізацію узагальненого управління з прогнозом з використанням багат шарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління. Даний алгоритм є основою побудови і функціонування нейрорегулятора прогнозом NN Predictive Controller, який застосовано в магістерській роботі при синтезі нейромережевої системи управління.

1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж

Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання нелінійних динамічних систем є отримання їх адекватних математичних моделей, який базується, як правило, на теоретичному і експериментальному аналізі властивостей цих систем. Теоретичний аналіз, заснований на вивченні фізичних і хімічних процесів, які відбуваються в системі, дозволяє отримати математичний опис, наприклад, диференціальних рівнянь. При експериментальному аналізі на основі спостережень вхідних і початкових сигналів системи отримують або її параметричну, або непараметричну модель. Найбільшого розповсюдження придбали параметричні моделі, які вимагають рішення задач структурної і параметричної ідентифікації і що використовують обмежену кількість параметрів. Незважаючи на велику кількість досліджень, різноманіття видів нелінійностей ускладнює створення універсальної теорії ідентифікації нелінійних систем. Зазвичай використовується класичний підхід, що ґрунтується на апроксимації нелінійностей, наприклад, за допомогою рядів Вольтерра, Гаммерштейна, Вінера, поліномів Колмогорова-Габора та інших, проте сфера застосування таких моделей обмежена. Крім того, додаткові труднощі отримання адекватного математичного опису зумовлює наявність в реальних сигналах завад, яка вимагає попередньої фільтрації сигналів.

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

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

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

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

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

ξ – завада вимірювання вихідного сигналу.

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

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

Тоді рівняння (1.1) може бути записане таким чином:

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

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

Схема ідентифікації нелінійного об'єкту (1.1) на підставі ШНМ приведена на рис. 1.1.

Схема ідентифікації нелінійного об'єкту (1.1) на підставі ШНМ приведена на рис. 1.1.

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (1.1) сьогодні отримали багатошаровий персептрон (рис. 1.2) і радіально-базисні мережі (РБМ) (рис. 1.3), що використовують апроксимацію нелінійного оператора $f(\cdot)$ нейронною мережею.

Рівняння БП має вигляд

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

де q – кількість куль в мережі;

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

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

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

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

де c_i – вагові коефіцієнти, які підлягають визначенню.

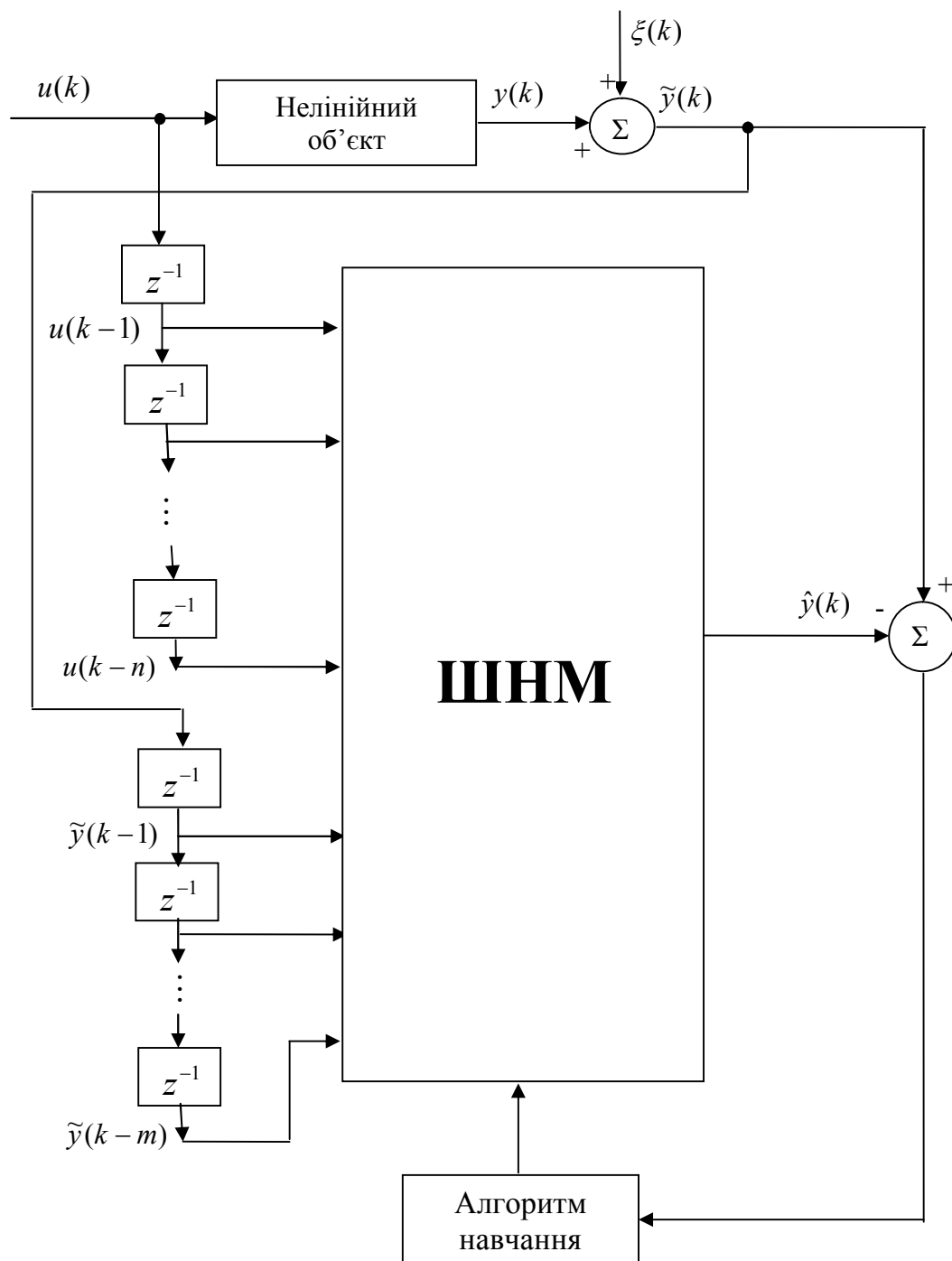


Рисунок 1.1 – Схема ідентифікації динамічного об'єкту

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

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

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

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

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

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

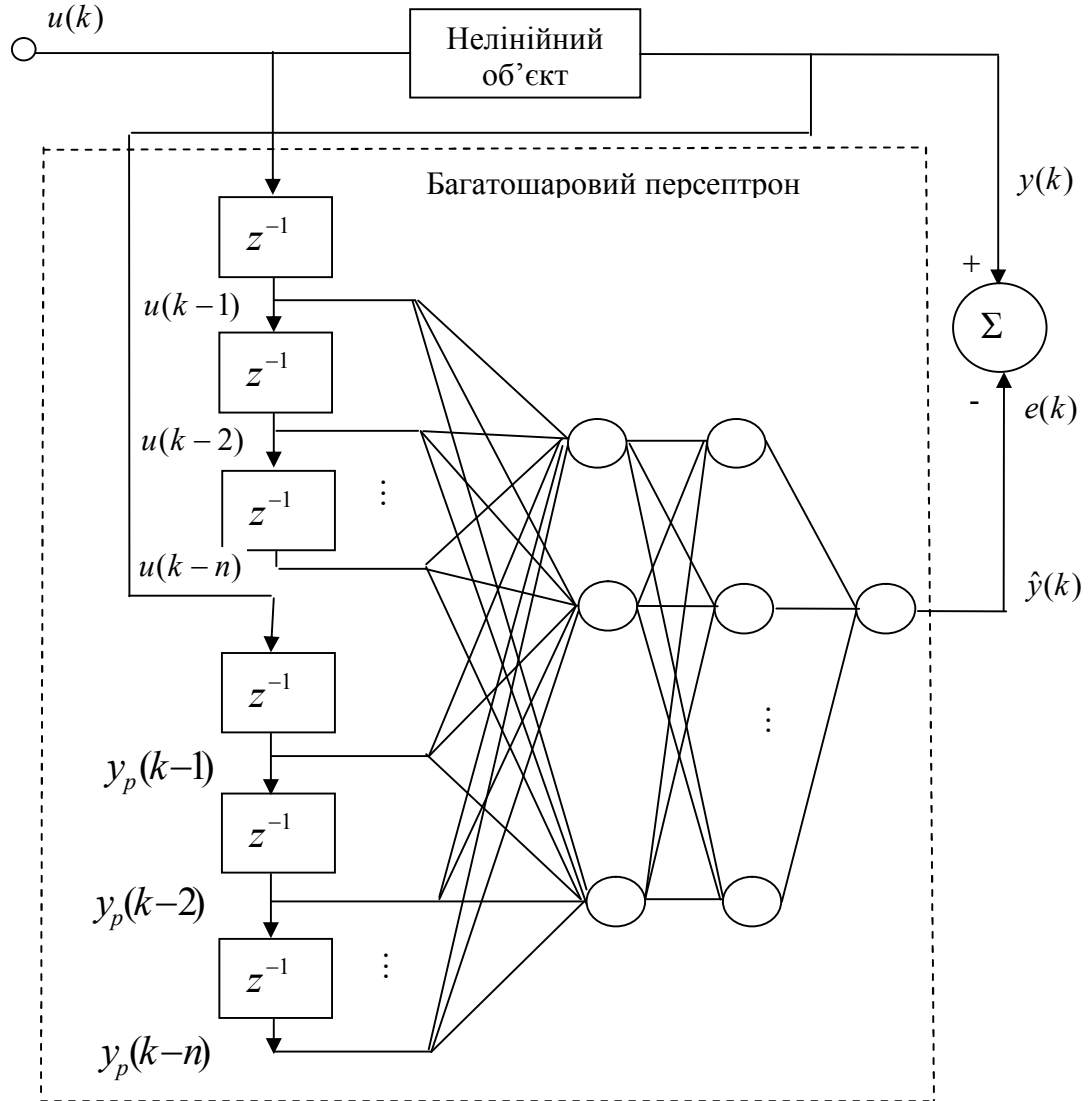


Рис 1.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

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

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

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

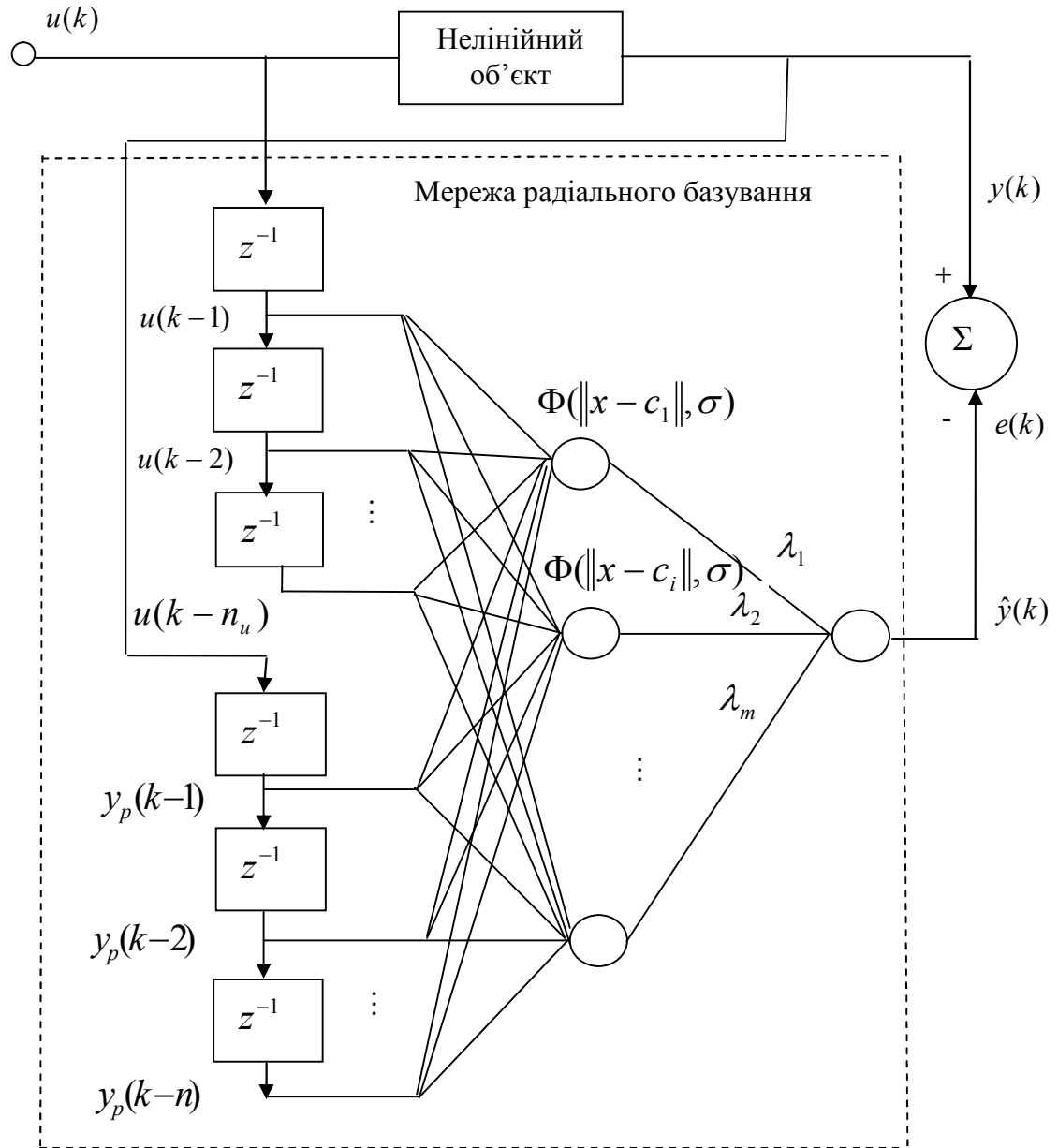


Рисунок 1.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

1.3 Нейрокерування динамічними об'єктами

Розв'язання задачі керування реальними об'єктами засновано на використанні його математичної моделі, вид і складність якої визначають складність алгоритму керування. У ряді випадків виправдовує себе заміна нелінійної моделі лінійною й знаходження параметрів регулятора, які б забезпечували най-

краще керування в деякій компромісній точці. Для оптимізації роботи об'єкта в декількох точках необхідна корекція параметрів регуляторів відповідно до зміни робочих умов. У зв'язку з цим доволіною є розробка систем керування на основі адаптивного підходу в поєднанні з методами теорії ШНМ, 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.

Структуру побудованої в такий спосіб нейромережевої системи адаптивного керування наведена на рис. 1.4.

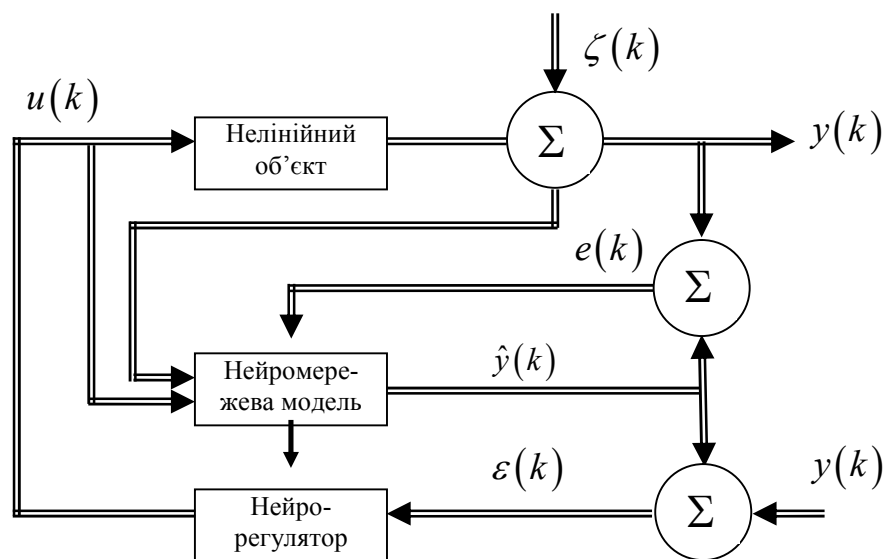


Рисунок 1.4 – Структура нейромережевої системи адаптивного управління

Найчастіше для розв'язання даних задач використовується багатошаровий персептрон і радіально-базисні мережі.

Розглянемо багатовимірний нелінійний динамічний об'єкт, описуваний рівнянням

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

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

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

$\zeta(k) = [\zeta_1(k), \dots, \zeta_{N_y}(k)]^T$ – вектор завад.

Позначивши вектор узагальненого сигналу, що надходить на вхід моделі, як

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

перепишемо рівняння (1.8) у вигляді

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

Навчання мережі, до якого зводиться навчання ідентифікації, полягає в настроюванні її параметрів на основі порівняння вихідних сигналів об'єкта $y(k)$ і моделі $\hat{y}(k)$ та в мінімізації звичайно квадратичного функціонала помилки (1.7).

Після завершення процесу навчання мережа використовується для реалізації алгоритму керування.

Задача керування полягає у визначення значень $u(k)$, що забезпечують мінімум функціоналу

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

де $y^*(k)$ – необхідні значення вектора вихідних сигналів.

Як й у випадку ідентифікації, для знаходження мінімуму функціонала (1.11) можуть бути використані різні рекурентні алгоритми, зокрема, градієнти вигляду

$$u(k) = u(k-1) + \gamma(k)(\nabla_u \varepsilon(k)), \quad (1.12)$$

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

1.4 Огляд структур нейромережевих систем управління. Вибір структури системи

Історично можна стверджувати, що розробка систем управління (СУ) завжди відбувається поетапно. До таких етапів можна віднести:

- етап розробки концепції побудови СУ;
- етап моделювання СУ, відповідно до запропонованої концепції побудови;
- етап аналізу отримуваних результатів;
- етап доопрацювання (модернізації) концепції побудови СУ.

Протягом усіх цих етапів тривають теоретичні дослідження, що сприяють визначенню основних напрямків удосконалення спочатку запропонованої концепції побудови СУ та розширенню її ідей на суміжні сфери.

Подібним чином відбувається розвиток СУ, що застосовують штучні нейронні мережі (ШНМ). Проте слід зазначити, що багато розроблених апаратних і програмних моделей нейромережевих СУ часто випереджають теоретичне розуміння процесів, що відбуваються, та існуючих проблем.

Кількість структур систем нейрорегулювання, запропонованих і випробуваних на сьогоднішній день, дуже велика. Одні системи нейрорегулювання використовують аналітичний закон регулювання і нейромодель. 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. Далі стисло розглядаються структури систем нейрорегулювання, актуальні на сьогоднішній день.

Однією з класичних моделей СУ є модель із зворотним зв'язком з регульованими в реальному масштабі часу коефіцієнтами, наприклад самоналагоджувальний регулятор Астрома [47]. Коефіцієнти такого контролера регулюються впродовж кожного циклу управління відповідно до оцінки параметрів

системи. Блок-схема управління із зворотним зв'язком і регульованими в реальному масштабі часу коефіцієнтами приведена на рис. 1.5.

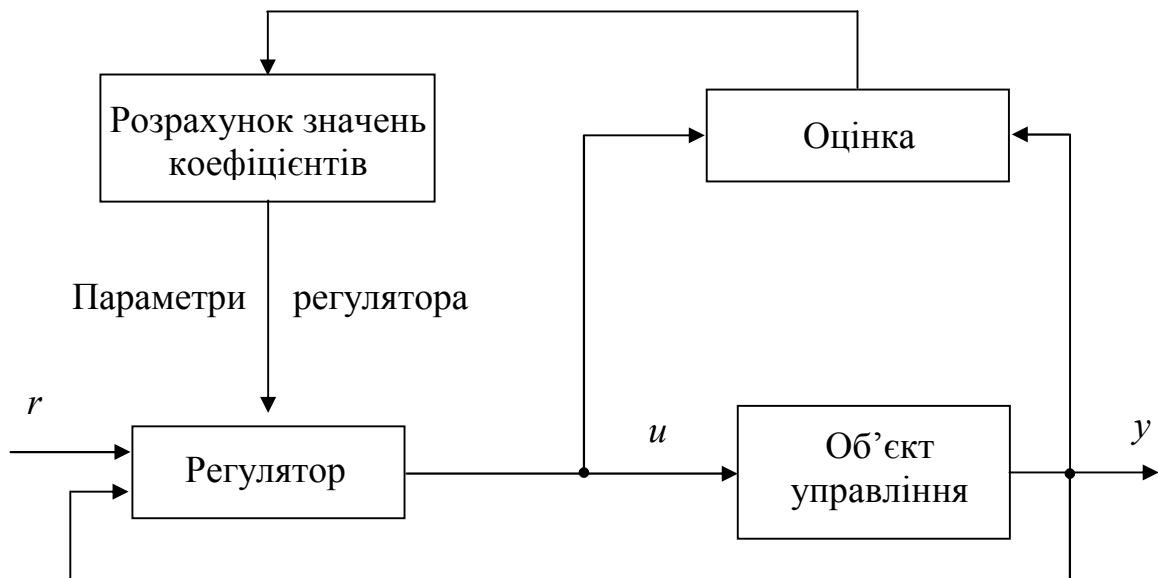


Рисунок 1.5 – Блок-схема управління із зворотним зв'язком і регульованими коефіцієнтами

Іншою добре відомою моделлю СУ є модель Ляпунова. Системи адаптивного управління, що використовують еталонну модель Ляпунова проектуються так, щоб вихідний сигнал керованої моделі врешті-решт відповідав вихідному сигналу заздалегідь визначеної моделі, яка має бажані характеристики [48]. Така система повинна бути асимптотично стійкою, тобто керована система у результаті відстежує еталонну модель з нульовою помилкою. Більш того, перехідні процеси на етапі адаптивного управління або навчального управління мають гарантовані межі. Блок-схема адаптивного управління з еталонною моделлю представлена на рис. 1.6.

Системи управління, які так чи інакше застосовують штучні нейронні мережі (ШНМ), є однією з альтернатив класичним методам управління. Здатність ШНМ ефективно розв'язувати управлінські завдання багато в чому зумовлена тим, що мережа з двома шарами та великою кількістю вузлів у прихованому шарі може апроксимувати будь-яку функцію дійсних чисел з заданою точністю. Це твердження, яке ґрунтується на відомій теоремі Вейєрштраса, детально ви-

кладено у [49]. Тому ШНМ, навіть з одним прихованим шаром, можуть використовуватись для задач ідентифікації та управління.

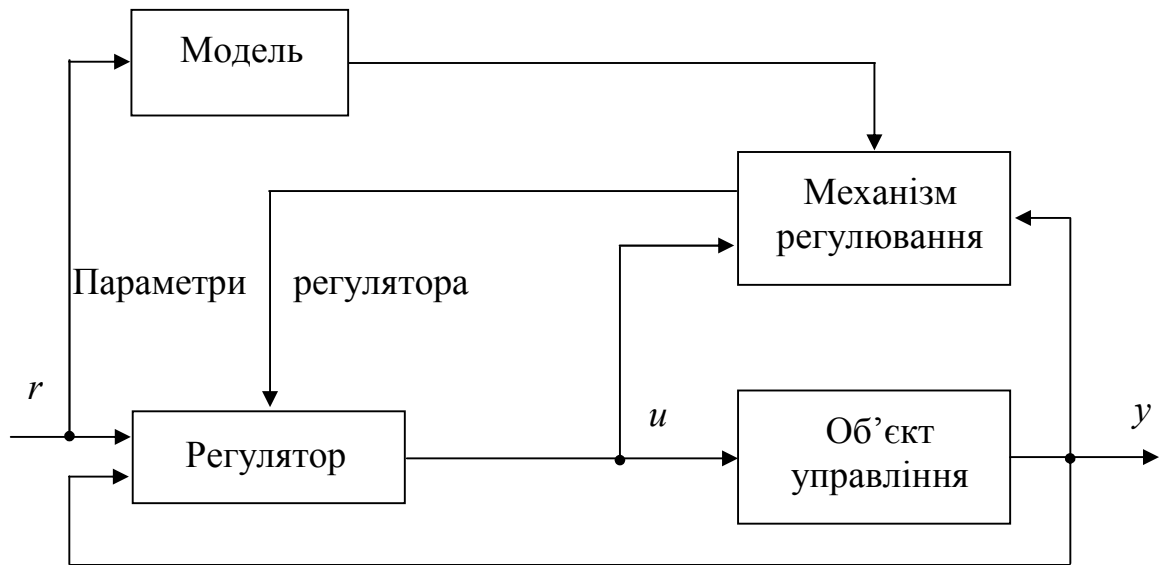


Рисунок 1.6 – Блок-схема адаптивного управління з еталонною моделлю

Одним з перших використовуваних методів побудови нейромережевих СУ був метод, заснований на "копіюванні" існуючого контролера. Застосувавши цей метод в 1964 Уїдроу назвав його методом побудови експертної системи за рахунок отримання знань від вже існуючого експерта. Архітектура такої СУ представлена на рис. 1.7.

Розглядаючи цей рисунок, можна засумніватися у доцільності застосування такого підходу. Чому потрібно додатково використовувати управляючий контролер у вигляді ШНМ, якщо вже існує один? Однак, по-перше, існуючий контролер може бути незручним у використанні (наприклад, у ролі контролера може виступати людина). По-друге, для ефективного управління ШНМ може використовувати іншу форму представлення інформації про стан об'єкта управління, що зручніше для вимірювання, формалізації тощо.

Зараз існує багато різних архітектур нейромережевих систем управління, які добре розроблені і широко використовуються [50,51]. У всіх випадках нейромережевий контролер призначений для створення відповідного управляючого сигналу, що забезпечує зміну стану об'єкта управління від початкового до

бажаного підсумкового стану, причому зміна повинна відбуватися за оптимальною траєкторією. Організація контролю за станом об'єкта і реалізація нейромережевого контролера значною мірою залежать від вибраного алгоритму навчання та структури управління. Найпоширенішими є схема прямого (безпосереднього) управління та схема непрямого управління. Зазвичай для навчання застосовується алгоритм зворотного розповсюдження помилки.

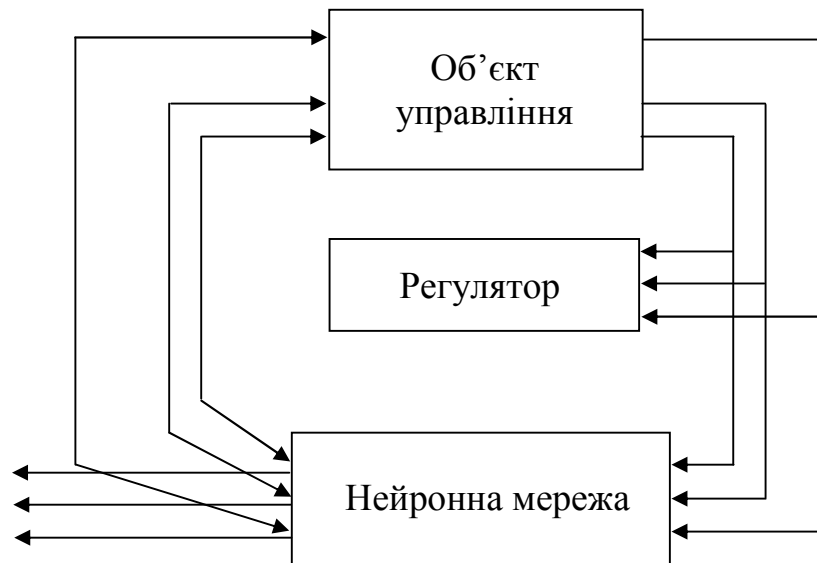


Рисунок 1.7 – Нейромережева СУ, заснована на «копіюванні» існуючого контролера

У схемі непрямого управління параметри об'єкта управління визначаються в кожний момент часу і використовуються для оцінки параметрів контролера (див. рис. 1.8). Таким чином, процес ідентифікації є чітко вираженим.

Недоліком такої схеми є те, що ідентифікація і управління ґрунтуються виключно на помилці e_u і, отже, мінімізацію помилки на виході системи e_y гарантувати не можна. У схемі прямого управління параметри нейромережевого контролера регулюються так, щоб зменшити безпосередньо помилку виходу e_y (див. рис. 1.9).

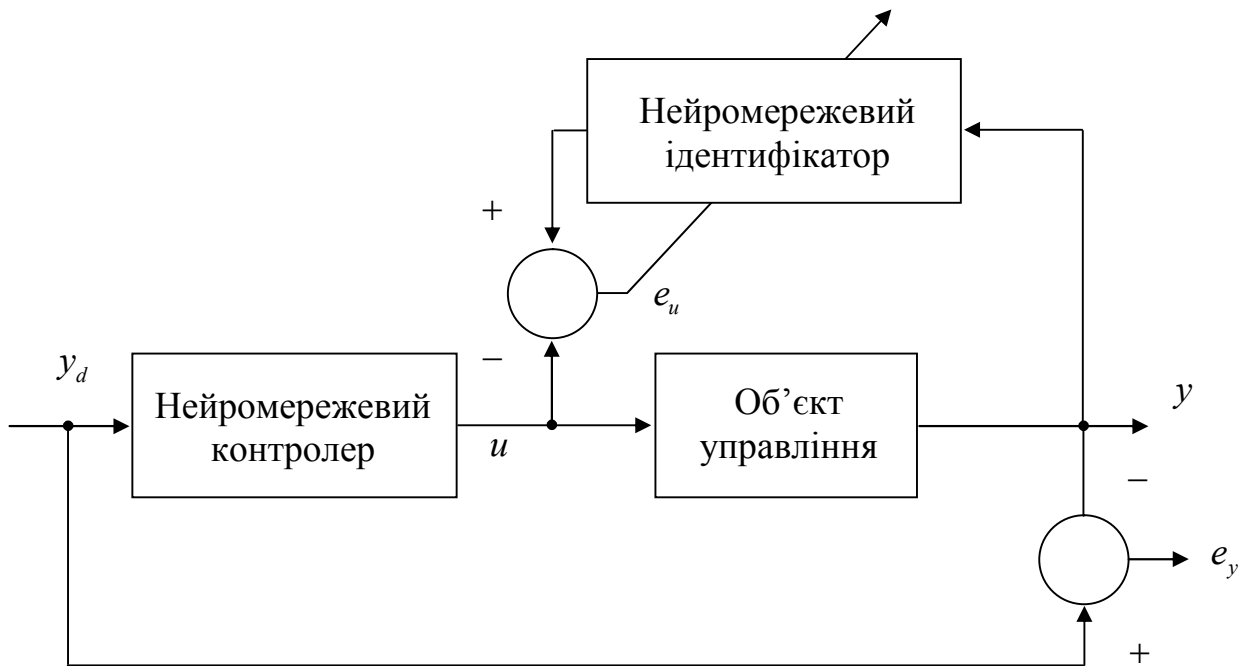


Рисунок 1.8 – Схема непрямого управління

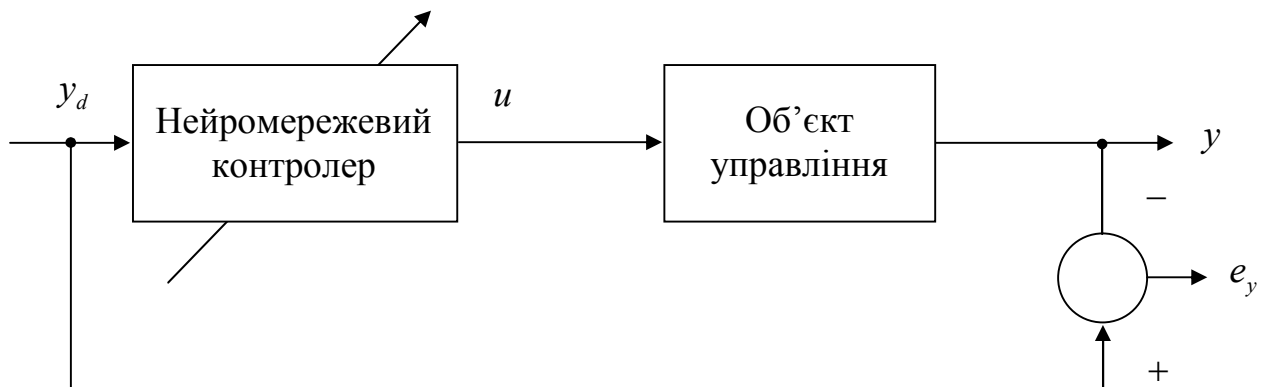


Рисунок 1.9 – Схема непрямого управління

Як цільова функція, яка повинна бути мінімізована контролером використовується середньоквадратична помилка на виході об'єкту управління

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

Питання стійкості і керованості таких СУ детально розглядаються в роботах [52-56].

Принцип предиктивного регулювання на основі моделі (model predictive control, MPC) [57] (рис. 1.6) полягає у створенні послідовності сигналів впливу

на об'єкт, яка мінімізує відхилення між заданим значенням і вихідним сигналом, що передбачає модель процесу в майбутньому (на «горизонті»), (див. пунктирні лінії на рис. 1.10).

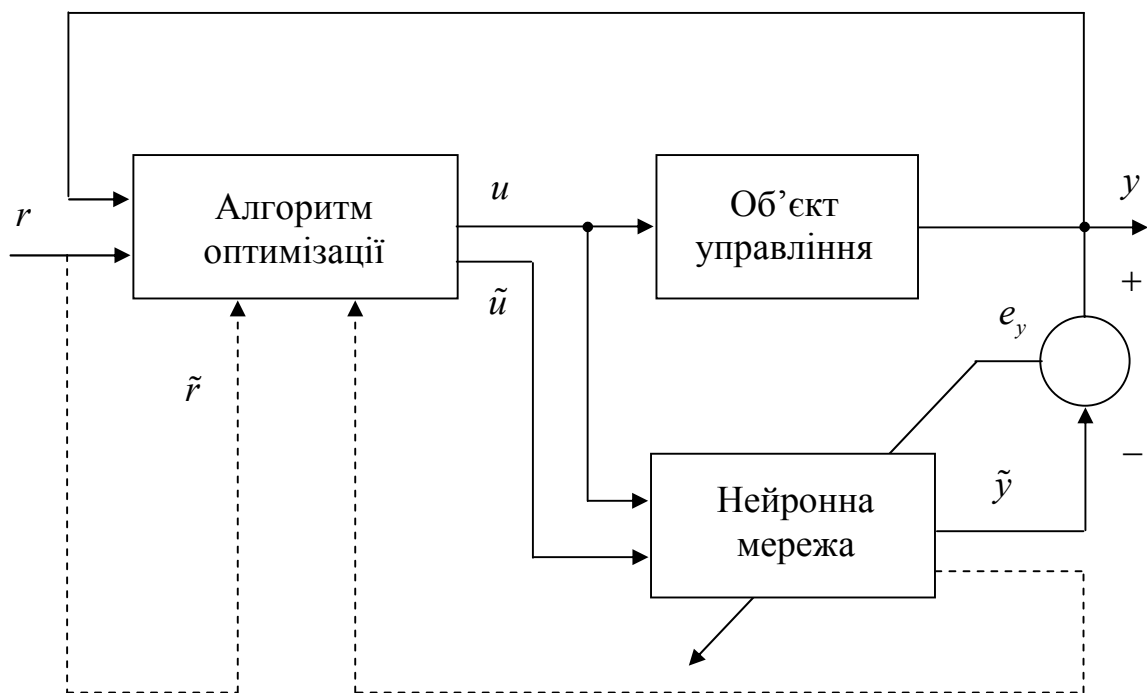


Рисунок 1.10 – Предиктивне регулювання

У цьому випадку регулятор є ітераційним алгоритмом оптимізації [57]:

- оптимізація виконується за допомогою чисельних методів;
- мінімізується функціонал якості, який описує бажані характеристики регулювання;
- для впливу на об'єкт використовується лише перший елемент оптимізованої послідовності;
- процес оптимізації повторюється на кожному кроці дискретизації.

Перевагою методу МРС є висока якість регулювання, навіть при наявності складних нелінійностей в об'єкті. Проте, недоліком є висока обчислювальна складність і значні вимоги до точності нелінійної моделі об'єкта.

У даній магістерській роботі розглядається використання нелінійного предиктивного регулювання.

Однією з напрямків теоретичних досліджень, що аналізують застосування ШНМ у системах управління, є порівняння таких методів з іншими відомими

типами СУ, виявлення їх специфічних характеристик і проведення аналізу. У роботі [58] представлений огляд характеристик зазначених СУ. Порівнювались стійкість, швидкість збіжності, ефективність у шумових умовах, обсяг необхідної пам'яті тощо. Хоча кожен з розглянутих методів має як переваги, так і недоліки, слід зазначити, що метод нейромережевого управління володіє корисними властивостями, які складно реалізувати за допомогою інших методів.

Основні результати, отримані при порівнянні, приведені в табл. 1.1.

Таблиця 1.1 – Огляд характеристик методів управління

Критерій	Управління із зворотним зв'язком і регульованими коефіцієнтами	Адаптивне управління з еталонною моделлю Ляпунова	Нейромережеве управління
Стійкість зворотного зв'язку	Якнайгірша	Якнайкраща	Середня
Швидкість збіжності	Якнайкраща	Середня	Якнайгірша
Робота в реальному часі	Середня	Середня	Якнайкраща
Складність програми управління	Якнайгірша	Середня	Середня
Помилка стеження	Середня	Якнайкраща	Середня
Зменшення перешкод	Якнайкраща	Якнайгірша	Середня
Робасність Розузгодження моделі	Якнайгірша	Середня	Якнайкраща

Метод, що використовує ШНМ, не має обмежень щодо лінійності системи, ефективно працює в умовах шумів і забезпечує управління в реальному часі після завершення навчання. Нейромережеві СУ мають більшу гнучкість у налаштуванні під реальні умови, створюючи моделі, які повністю відповідають поставленим завданням і не обмежені структурою формальних систем. Окрім того, ці системи не тільки реалізують стандартні адаптивні методи управління, але й пропонують нові алгоритмічні підходи для вирішення завдань, які важко формалізувати. Наприклад, вони можуть обробляти дані одного типу в межах однієї моделі, при цьому для ШНМ важлива лише кореляція цих даних.

Отже, можна зробити висновок, що майбутнє інтелектуального управління полягає у поєднанні традиційних методів управління з можливостями та перспективами використання систем, заснованих на штучних ШНМ.

1.5 Загальні відомості про роторні екскаватори

Головним обладнанням гірничодобувних комплексів є роторний екскаватор, який представляє собою складну конструкцію, що складається з понад десяти взаємодіючих силових механізмів і включає як основне, так і допоміжне обладнання для транспортування видобутої породи до системи транспортних машин комплексу. Схему розташування компонентів роторного екскаватора можна побачити на рис. 1.11.

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.

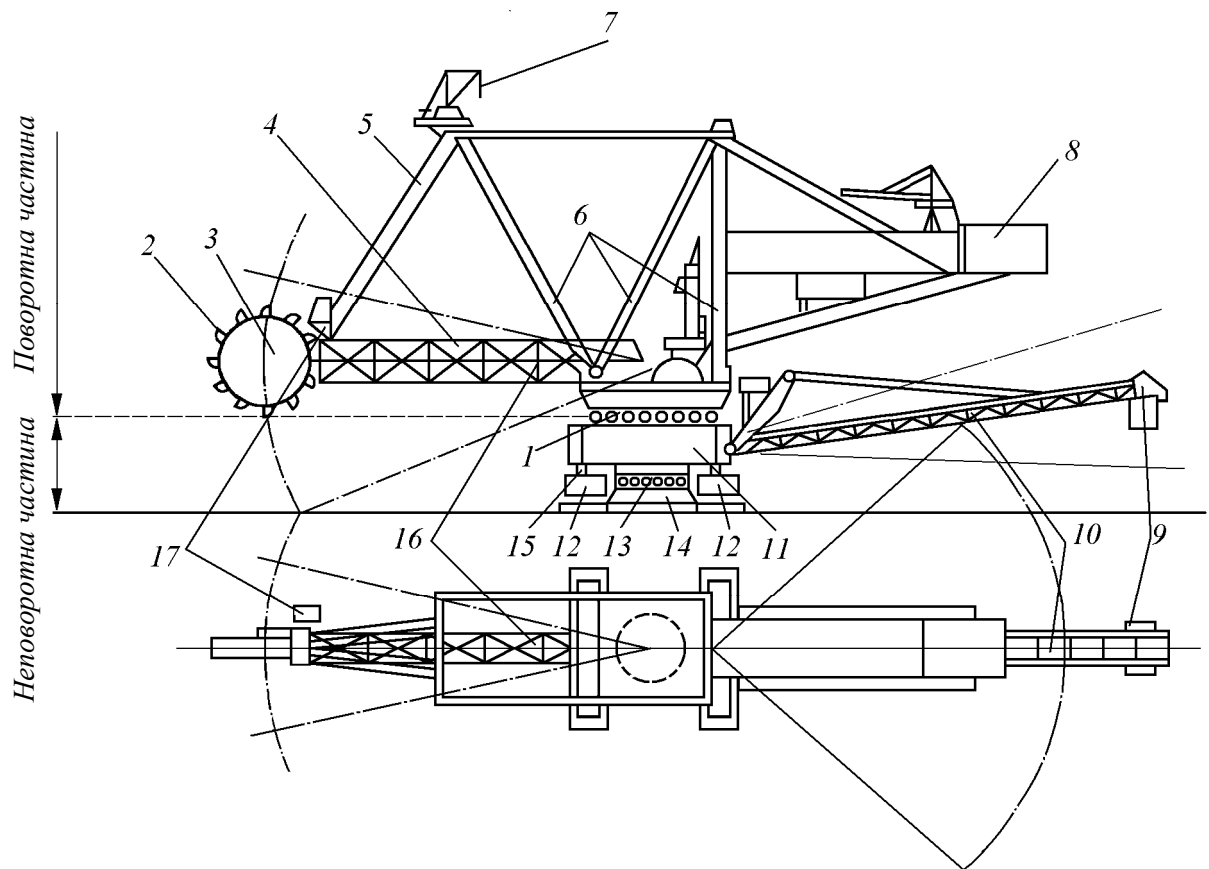


Рисунок 1.11 – Компонувальна схема роторного екскаватора

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, нижнєе- 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 протидія, 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 опорной 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.

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.

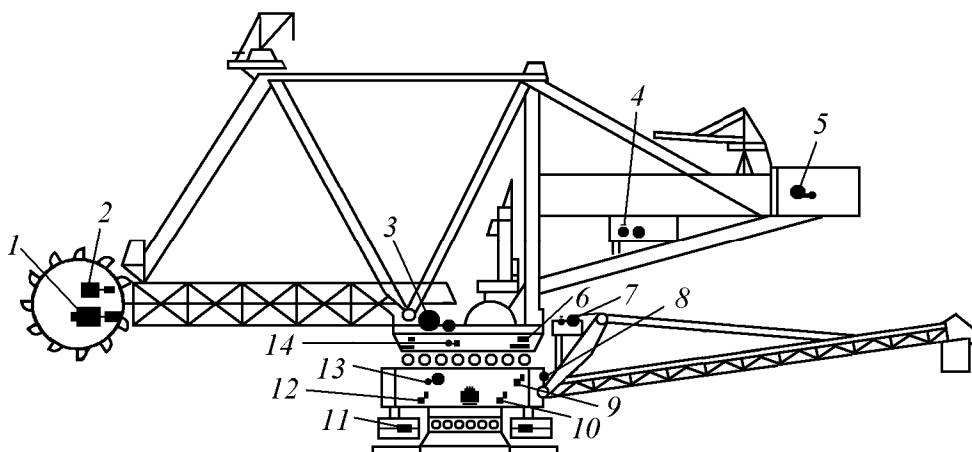


Рисунок 1.12 - Компонівка індивідуальних електроприводів
ротаторного екскаватора

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 додблюючого 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 лиж рейковий-крокуючого ходу коштують електроприводи ходових лебідок 11.

1.6 Аналіз факторів, які обумовлюють вибір системи управління електроприводом механізму розвороту роторного екскаватора

Ефективне використання екскаваторів рот 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 електропривод 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 переналаджень, повинні забезпечувати можливість швидкого 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 механізм-

мами 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 електроприводів 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.

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 розвороту використовуємо систему ТП-Д.

1.7 Постановка задачі синтезу нейромережевої системи управління

В результаті проведеного аналізу літератури можна зробити висновок, що існує велика кількість підходів до проблеми синтезу систем управління, при цьому універсального і ідеального регулятора поки не існує. При синтезі системи управління треба враховувати той фактор, що для електроприводу механізму розвороту екскаватора характерно 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. Традиційні регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

Теорія адаптивного та оптимального управління дозволяє створювати системи, що функціонують із метою мінімізації або максимізації заздалегідь обраного критерію якості. Сучасна теорія управління, поряд із методами синтезу управлінських пристроїв і систем, включає також методи ідентифікації та оцінки станів процесів. Різні алгоритми ідентифікації допомагають визначити структуру моделі об'єкта управління та відновити її параметри як у процесі управління, так і поза ним. Ці методи використовуються при створенні як звичайних, так і адаптивних систем управління.

Основою класичної теорії управління є розвинена теорія лінійних систем. Елементи цієї теорії часто використовуються при дослідженні та синтезі управлінських систем для нелінійних об'єктів, однак оптимальні результати

можна отримати лише тоді, коли нелінійність об'єкта незначна або коли об'єкт характеризується великими постійними часами та стійкістю в розімкненому стані. Управління динамічними об'єктами ускладнюється необхідністю постійного моніторингу не тільки значень параметрів моделі, але й іноді її структури. Найпоширенішим підходом є апроксимація нелінійностей за допомогою рядів Вольтерра, Гаммерштейна, Вінера, поліномів Колмогорова-Габора тощо. Однак використання таких моделей має обмеження, оскільки для їх застосування потрібно попередньо вирішити задачу структурної ідентифікації, щоб визначити порядок моделі. Зовнішні збурення та похибки вимірювань вимагають додаткової обробки та ускладнюють рішення задач управління, а потреба в управлінні динамічними об'єктами в реальному часі ставить перед розробниками завдання створення швидкодіючих алгоритмів управління.

Незважаючи на велику кількість досліджень, різноманіття типів нелінійностей, динамічні властивості об'єктів, їх нестаціонарність, а також наявність зовнішніх збурень і похибок вимірювань не дозволяє створити універсальний підхід до побудови систем управління. Крім того, як у класичній, так і в сучасній теорії управління основою є ідея лінеаризації систем. Однак при розв'язанні практичних задач цей підхід може не забезпечити необхідний рівень якості управління, оскільки модель, побудована на припущенні лінійності, може не відображати реальні властивості системи.

Штучні нейронні мережі є перспективною альтернативою класичним методам створення систем управління нелінійними об'єктами. Нейронні мережі – це галузь штучного інтелекту, де для обробки сигналів використовуються процеси, схожі на ті, що відбуваються в нейронах живих організмів. Основною особливістю нейронної мережі, яка підкреслює її потужність і великий потенціал, є паралельна обробка інформації всіма елементами. Завдяки значній кількості міжнейронних зв'язків процес обробки даних значно прискорюється, що дозволяє здійснювати обробку сигналів у реальному часі. Крім того, велика кількість з'єднань робить мережу стійкою до помилок на окремих лініях, оскільки функції пошкоджених зв'язків можуть виконувати інші, працездатні

лінії, завдяки чому діяльність мережі залишається стабільною. Іншою важливою властивістю є здатність до навчання та узагальнення накопичених знань. Нейронна мережа має властивості штучного інтелекту: навчившись на обмеженій кількості даних, вона може узагальнювати отриману інформацію та показувати хороші результати на нових даних, які не використовувались під час навчання.

Окрім того, наявність у структурі штучних нейронних мереж нейронів з нелінійними функціями активації дає змогу застосовувати їх для управління нелінійними динамічними об'єктами, тоді як традиційні методи не здатні ефективно розв'язувати такі завдання.

В даний час, при побудові систем управління нелінійними динамічними об'єктами, найбільшу популярність отримали штучні нейронні мережі типу багат шаровий персептрон, радіально-базисні мережі, а також нейро-фаззи мережі.

Всі ці структури здатні навчатися будь-яким функціям, для чого в процесі навчання необхідно надати чималий об'єм інформації у форматі «вхід-вихід» (об'єм інформації повинен на порядок перевищувати кількість зв'язків в мережі) і забезпечити правильний вибір самої нейромоделі (це завдання включає вибір кількості нейронів і шарів в мережі, а також вибір виду функцій активації нейронів). Для реалізації нелінійних залежностей між входом і виходом в ШНМ використовуються сигмоїдальні функції активації або деякі нелінійні функції загального вигляду, наприклад, радіально-базисні функції. Ця властивість ШНМ використовується при рішенні задач управління з істотними нелінійностями, для яких традиційні підходи не дають рішень, що практично реалізуються.

Необхідною умовою застосування традиційних методів оптимального і адаптивного управління є наявність великого об'єму апріорної інформації. Завдяки здібності штучних нейронних мереж до самонавчання, для систем управління, побудованих на основі нейрорегуляторів, такий об'єм інформації не потрібний. Це дозволяє використовувати нейрорегулятори в умовах істотних невідомостей.

Побудова системи управління на основі нейрорегулятора зводиться до рішення послідовно задачі ідентифікації, а потім до побудови алгоритму управління відповідно до вибраних цілей управління і критерію якості управління. Процес ідентифікації об'єкту зводиться до побудови його нейромоделі, що реалізовується в ході навчання мережі, на основі пред'явлення навчальних пар, якими служать вимірювані значення вхідних і відповідних вихідних змінних.

Метою роботи є розробка нейромережевої системи управління механізмом розвороту 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 задачі.

1. Виконати вибір системи електроприводу механізму розвороту 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. В результаті аналізу літератури по темі роботи обґрунтовано доцільність використання нейронних мереж у задачах управління. Сучасні системи управління складними динамічними об'єктами повинні мати здатність швидко адаптуватися до змінюваних умов функціонування, здійснюючи оперативне коригування параметрів і структури управлінських законів. Цим вимогам відповідає теорія адаптивного управління. Одним із ефективних способів реалізації концепцій адаптивного управління є використання методів та засобів нейронного моделювання та управління. Сучасні нейромережеві інструменти забезпечують можливість розв'язання задач ідентифікації та управління як під час проектування систем (зі збереженням фіксованих алгоритмів управління), так і під час безпосереднього функціонування системи.

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. Необхідним етапом рішення задачі управління, прогнозування поведінки, моделювання динамічних систем є отримання їх адекватних математичних моделей. Показано, що завдання ідентифікації нелінійних динамічних об'єктів успішно вирішуються за допомогою штучних нейронних мереж. Найбільше застосування для ідентифікації отримали багатошаровий персептрон і радіально-базисні мережі.

4. 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 до-

силь прості нейромережеві моделі, а з іншого – адаптивно корегувати параметри як цих моделей, так і регулятора відповідно до умов, що змінюються.

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

6. Розглянута ефективна реалізація узагальненого управління з прогнозом з використанням багат шарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління. Показано, що завдяки використанню оптимізаційного алгоритму Ньютона-Рафсона, число ітерацій, необхідних для збіжності, значно менше, ніж при використанні інших методів, унаслідок чого алгоритм Ньютона-Рафсона є швидшим, ніж інші методи, і може бути використовуватися для управління в режимі реального часу.

5. На підставі аналізу технічних вимоги, що пред'являється до електроприводів опорно-поворотних механізмів роторних екскаваторів для електроприводу механізму розвороту вибраний електропривод постійного струму, виконаний за системою ТП-Д.

6. 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.

7. Проаналізовані нейромережеві підходи до рішення задачі управління. Сформульовано задачу дослідження, що складається в розробці нейромережевої системи управління механізмом розвороту роторного екскаватора з урахуванням пружних механічних зв'язків.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ РОЗВОРОТУ 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 електродвигунів механізму розвороту екскаватора

Згідно рекомендаціям, потужність електроприводу розвороту екскаватора складає (0.8-1) потужності електроприводу повороту стріли. Тому визначимо потужність електроприводу повороту і по ній виберемо електродвигуни для розвороту екскаватора.

Потужність двигуна повороту стріли:

$$P_{\text{д}} = \frac{1,5M_{\text{б}}\omega_{\text{пл}}}{\eta_{\text{п}}k} 10^{-3}, \quad (2.1)$$

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

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

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

$M_{\text{б}}$ – момент, обумовлений зусиллям, що виникає на кромках ковшів:

$$M_{\text{б}} = F_{\text{б}}R_{\text{к}}, \quad (2.2)$$

$F_{\text{б}}$ – бічне зусилля різання;

$R_{\text{к}}$ – радіус різання в горизонтальній площині, вимірюваний по кромці ковшів; радіус різання залежить від кута нахилу $\beta_{\text{с}}$ стріли ротора по відношенню до горизонталі і міняється залежно від висоти оброблюваного шару:

$$R_{\text{к}} = R_0 \cos \beta_{\text{с}}. \quad (2.3)$$

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

$$M_{\text{б}} = M_{\text{бmax}} = F_{\text{б}}R_0, \quad (2.4)$$

$$M_{\bar{\sigma}} = M_{\bar{\sigma}_{\max}} = 3,8 \cdot 10^4 \cdot 23,3 = 88,54 \cdot 10^4 \text{ Нм},$$

$$P_{\text{д}} = \frac{1,5 \cdot 10^{-3} \cdot 885400 \cdot 2,9 \cdot 10^{-2}}{0,65 \cdot 2} = 29,6 \text{ кВт}.$$

Потужність кожного з двох двигунів механізму повороту стріли екскаватора рівна 29,6 кВт

Оскільки потужність двигунів механізму розвороту екскаватора прийнята 0,8 потужності двигунів механізму повороту стріли, то попередня потужність таких двигунів рівна:

$$P_{\text{н}} = 0,8 \cdot 29,6 = 23,68 \text{ кВт}$$

По каталогу вибираємо два двигуни типу ДВ808. Потужність $P_{\text{н}} = 37 \text{ кВт}$, частота обертання $n_{\text{н}} = 575 \text{ об/хв}$; напруга $U_{\text{н}} = 220 \text{ В}$, струм якоря $I_{\text{н}} = 192 \text{ А}$, магнітний потік полюса $\Phi_{\text{н}} = 0,0381 \text{ Вб}$, число пар полюсів $2p = 4$, опір обмоток при 20°C $R_{\text{яд}} = 0,034 \text{ Ом}$, $R_{\text{дп д}} = 0,02 \text{ Ом}$, момент інерції якоря $J_{\text{д}} = 2 \text{ кг} \cdot \text{м}^2$.

2.3 Розрахунок швидкісної діаграми механізму розвороту

Перевіряємо двигуни по нагріву і перевантажувальній здатності. Для цього розраховуємо швидкісну і навантажувальну діаграми.

Номінальна швидкість обертання двигуна

$$\omega_{\text{н}} = \frac{\pi \cdot n_{\text{н}}}{30}, \quad (2.5)$$

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

Передавальне число механізму повороту:

$$i_{\pi} = \frac{\omega_{\text{н}}}{\omega_{\text{пл}}} , \quad (2.6)$$

$$i_{\pi} = \frac{60,2}{2,9 \cdot 10^{-4}} = 2075,86 .$$

Швидкісна діаграма механізму повороту стріли має три ділянки:

- розгін з боковим прискоренням;
- рух із сталою швидкістю;
- гальмування з допустимим уповільненням.

Допустимі величини прискорень і уповільнень платформи $\dot{\omega}_{\text{пл}}$ приводимо до швидкості двигуна

$$\dot{\omega}_{\text{доп}} = \dot{\omega}_{\text{пл}} \cdot i_{\pi} , \quad (2.7)$$

$$\dot{\omega}_{\text{доп}} = 0,005 \cdot 2075,86 = 10,4 \text{ c}^{-2} .$$

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

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

$$t_{\text{р}} = t_{\text{г}} = \frac{60,2}{10,4} = 5,79 \text{ c} .$$

Кути повороту, прохідні платформною під час розгону і гальмування

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

$$\alpha_{\text{р}} = \alpha_{\text{г}} = 0,005 \cdot \frac{5,79^2}{2} = 0,0838 \text{ рад} = 4,8^{\circ} ,$$

Кут повороту платформи за час сталої швидкості.

$$\alpha_y = 2 \cdot (\alpha_{n \max} - \alpha_p) , \quad (2.10)$$

$$\alpha_y = 2(75 - 4,8) = 140,4^\circ = 2,45 \text{ рад}.$$

Час роботи поворотного механізму із сталою швидкістю

$$t_y = \frac{\alpha_y \cdot i_n}{\omega_n} , \quad (2.11)$$

$$t_y = \frac{2,45 \cdot 2075,86}{60,2} = 84,48 \text{ с}.$$

Час циклу, тобто час руху в прямому і зворотному напрямі

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

$$T_{\text{ц}} = 2(5,79 + 5,79 + 84,48) = 192,12 \text{ с}.$$

Швидкісна діаграма приведена на рис. 2.1 а.

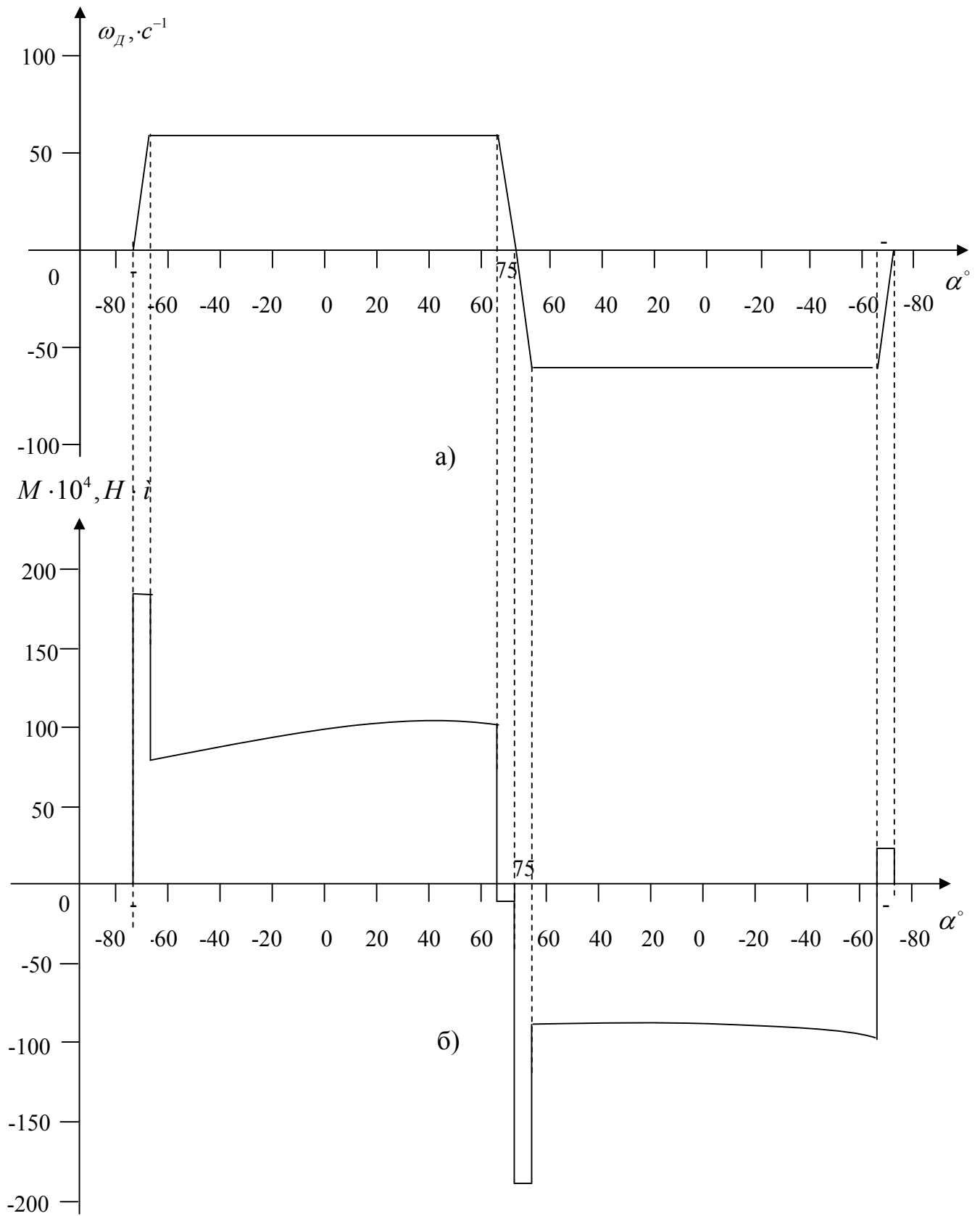


Рисунок 2.1 – Швидкісна а) і навантажувальна б) діаграми електроприводу

2.4 Побудова діаграми навантаження електроприводу

Обчислимо статичні навантаження, що визначають сумарний момент опору щодо осі обертання поворотної частини екскаватора.

Момент, обумовлений зусиллям, що виникає на кромках ковшів, розрахований в попередньому розділі і дорівнює $M_{\delta} = M_{\delta \max} = 88,54 \cdot 10^4 \text{ Нм}$

Із-за різних площ конструкції машини, розташовані по різні сторони щодо осі обертання поворотної частини, виникає момент, обумовлений вітровим натиском. При постійності вітрового натиску момент від вітрового навантаження $M_{\text{вт}}, \text{ Нм}$

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

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

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

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

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

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

При відхиленні осі обертання від вертикалі на кут γ і повороті стріли на кут α_{Π} виникає момент від дії невідноважених сил тяжіння поворотної частини. Якщо вважати, що площа кута γ співпадає з площиною подовжнього перетину екскаватора, момент від дії невідноважених сил $M_{\text{нв}}, \text{ Н} \cdot \text{м}$, можна визначити:

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

де $R_{\text{цт}}$ – радіус прикладення невідноваженої сили тяжіння поворотної частини $G_{\text{нв}}$; в розрахунок приймаємо:

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

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

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

Q – теоретична продуктивність по рихлій масі, $\text{м}^3 / \text{год}$;

$\gamma_{\text{гр}}$ – щільність ґрунту, $\text{кг} / \text{м}^3$;

$L_{\text{пч}}$ – довжина транспортера приймальної частини, м , в розрахунку можна прийняти $L_{\text{пч}} = R_0$.

$V_{\text{т}}$ – швидкість руху транспортної стрічки, $\text{м} / \text{с}$;

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

$$m_{\text{в пч}} = \frac{1250 \cdot 1,8 \cdot 10^3 \cdot 23,3}{3600 \cdot 3,5 \cdot 1,5} = 2773,8 \text{ кг}.$$

$$G_{\text{нв}} R_{\text{цт}} = 9,8 \left[(127 \cdot 10^3 + 27773,8) - 120 \cdot 10^3 \right] \cdot \frac{23,3}{3} = 743917,2.$$

Максимальний подоланий ухил складає $\gamma = 10^\circ$, $\sin 10^\circ = 0,1736$.

$$M_{\text{нв}} = 743917,2 \cdot \sin 10^\circ \cdot \sin \alpha_{\text{п}} = 128985,6 \cdot \sin \alpha_{\text{п}}.$$

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

Момент опору обертання від сил тертя $M_{\text{тр}}$, $H \cdot \text{м}$, визначається по формулі:

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

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

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

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

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

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

$$M_{\text{тр}} = 9,8(500 \cdot 10^3 + 2773,8) \cdot \frac{8,7}{2 \cdot 0,8} \cdot 0,1 \cdot 10^{-2} \cdot 1,25 \cdot 1,25 = 31276 \text{ Нм}.$$

Момент $M_{\text{б}}$ і $M_{\text{тр}}$ є пасивними, вони завжди гальмівні незалежно від напрямку руху.

Статичний момент визначимо для повороту стріли по і проти годинникової стрілки, тобто для зміни $\alpha_{\text{п}}$ від -75° до $+75^\circ$ і від $+75^\circ$ до -75° згідно виразам:

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

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

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

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

($||$ – означає модуль відповідного моменту)

На ділянках розгону і гальмування слід врахувати динамічний момент

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

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

$J_{\text{м}\Sigma}$ – 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

$$J_{\text{м}\Sigma} = J_{\text{пл}} + kJ_{\text{д}}i_{\text{п}}^2 = 17 \cdot 10^7 + 2 \cdot 2 \cdot 43094,74 = 187236779 \text{ кг} \cdot \text{м}^2 ,$$

$$M_{\text{дин}} = 187236779 \cdot 0,005 = 936183,89 \text{ Нм} .$$

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

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

Результати розрахунків занесені в таблицю 2.1.

За наслідками розрахунків на рис. 2.1,б побудована діаграма навантаження електроприводу повороту.

Таблиця 2.1 – Розрахунок діаграми навантаження

Напрямок руху	Складові моменти	α_{Π}								
		-75°	$-(75^{\circ} - \alpha_p)$	-65°	-55°	-45°	-35°	-25°	-15°	0°
По часовій стрілці	$M_{\text{б}}$	885400	885400	885400	885400	885400	885400	885400	885400	885400
	$M_{\text{вт}}$	46742,7	61176	61176	103587,9	127703,5	147938,9	163679	174446,2	180600
	$M_{\text{нв}}$	-122156,7	-118989,4	-114617	-103594,8	-89424,9	-72537,87	-53446,8	-32731,7	0
	$M_{\text{тр}}$	31276	31276	31276	31276	31276	31276	31276	31276	31276
	$M_{\text{ст}}$	841262	858862,6	878383,9	916669,1	954954,6	992077	1026908,2	1058390,5	1097276
	$M_{\text{дин}}$	936183,9	936183,9							
	$M_{\text{рез}}$	1777445,9	1795046,5	878383,9	91669,1	954954,6	992077	1026908,2	105839,05	1097276
Проти часової стрілки	$M_{\text{б}}$	885400	885400	885400	885400	885400	885400	886400	885400	885400
	$M_{\text{вт}}$	46742,7	61176	76324,9	103587,9	127703,5	147938,92	163679	174446,3	180600
	$M_{\text{нв}}$	-122156,7	-118989,4	-114617	-103594,8	-89424,9	-72537,87	-53446,8	-32731,7	0
	$M_{\text{тр}}$	31276	31276	31276	31276	31276	31276	31276	31276	31276
	$M_{\text{ст}}$	-992090	-974489,4	-954968	-916682,9	-878397,4	-841274,9	-806443,8	-774961,5	-736076
	$M_{\text{дин}}$	-936183,9	-936183,9							
	$M_{\text{рез}}$	-55906,1	-38305,5	-954968	-916682,9	-87897,4	-841274,9	-806443,8	-774961,5	-736076

Продовження таблиці 2.1

Напрям руху	Складові моменти	α_1							
		5°	25°	35°	45°	55°	65°	75° – α_p	75°
По часовій стрілці	$M_{\dot{a}}$	885400	885400	885400	885400	885400	885400	885400	885400
	$\dot{I}_{\dot{a}o}$	179912	163679	147938,9	127703,5	103587,9	76324,9	61176	46742,7
	$\dot{I}_{\dot{i}\dot{a}}$	11022,2	53446,8	72537,87	89424,9	103587,9	114617	11898,4	122156,7
	$\dot{I}_{\dot{o}o}$	31276	31276	31276	31276	31276	31276	31276	31276
	$\dot{I}_{\dot{n}o}$	1107610,2	1133801,8	1137152,7	1133804,4	1123858,7	1107617,9	1096841,4	1085575,4
	$\dot{I}_{\dot{a}\dot{e}\dot{i}}$							936183,9	936183,9
	$\dot{I}_{\dot{o}\dot{a}\dot{c}}$	1107610,2	1133801,8	1137152,7	1133804,4	1123858,7	1107617,9	160657,5	149391,5
Проти часової стрілки	$M_{\dot{a}}$	885400	885400	885400	885400	885400	885400	885400	885400
	$\dot{I}_{\dot{a}o}$	179912	163679	147938,9	127703,5	103587,9	76324,9	61176	46742,7
	$\dot{I}_{\dot{i}\dot{a}}$	11022,2	53446,8	72537,87	89424,9	103594,8	114617	118989,4	122156,7
	$\dot{I}_{\dot{o}o}$	31276	31276	31276	31276	31276	31276	31276	31276
	$\dot{I}_{\dot{n}o}$	-725741,8	-699550,2	-696199,2	-699547,6	-709493,3	-725734	-736510,6	-747776,6
	$\dot{I}_{\dot{a}\dot{e}\dot{i}}$							-936123,9	-936183,9
	$\dot{I}_{\dot{o}\dot{a}\dot{c}}$	-725741,8	-699550,2	-696199,2	-699547,6	-709493,3	-725734	-1672694,5	-1683960,5

2.5 Перевірка двигунів по нагріву і перевантажувальній здатності

Перевірку двигунів по нагріву виконуємо методом еквівалентного моменту. Для обчислення середньоквадратичного моменту результуючу 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 $M_{\text{рез},i}$. Розрахунок середньоквадратичного моменту двигуна проводиться по формулі:

$$M_{\text{д экв}} = \frac{1}{\eta_n i_n k} \cdot \sqrt{\frac{\sum_{i=1}^n M_{\text{рез},i}^2 \Delta t_i}{T_{\text{ц}}}}, \quad (2.25)$$

$$\Delta t_i = \frac{T_{\text{ц}}}{30} = \frac{192,12}{30} = 6,4 \text{ с} \quad \Delta t_i - t_{\text{п(г)}} = 6,4 - 5,79 = 0,61 \text{ с}$$

$$\begin{aligned} M_{\text{д экв}} &= \frac{1}{0,65 \cdot 2075,86 \cdot 2} \cdot \sqrt{\frac{(17,7 \cdot 10^5)^2 \cdot 5,79 + (8,5 \cdot 10^5)^2 \cdot 0,61 + (8,78 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(9,2 \cdot 10^5)^2 \cdot 6,4 + (9,5 \cdot 10^5)^2 \cdot 6,4 + (9,9 \cdot 10^5)^2 \cdot 6,4 + (10,2 \cdot 10^5)^2 \cdot 6,4 + (10,5 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(10,8 \cdot 10^5)^2 \cdot 6,4 + (11 \cdot 10^5)^2 \cdot 6,4 + (11,2 \cdot 10^5)^2 \cdot 6,4 + (11,3 \cdot 10^5)^2 \cdot 6,4 + (9,2 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(11,37 \cdot 10^5)^2 \cdot 6,4 + (11,33 \cdot 10^5)^2 \cdot 6,4 + (11,2 \cdot 10^5)^2 \cdot 6,4 + (11 \cdot 10^5)^2 \cdot 6,4 + (10,9 \cdot 10^5)^2 \cdot 0,61 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(1,4 \cdot 10^5)^2 \cdot 5,79 + (-6,8 \cdot 10^5)^2 \cdot 5,79 + (-7,2 \cdot 10^5)^2 \cdot 0,61 + (-7 \cdot 10^5)^2 \cdot 6,4 + (-6,99 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(-6,96 \cdot 10^5)^2 \cdot 6,4 + (-6,99 \cdot 10^5)^2 \cdot 6,4 + (-7 \cdot 10^5)^2 \cdot 6,4 + (-7,2 \cdot 10^5)^2 \cdot 6,4 + (-7,3 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(-7,4 \cdot 10^5)^2 \cdot 6,4 + (-7,7 \cdot 10^5)^2 \cdot 6,4 + (-8 \cdot 10^5)^2 \cdot 6,4 + (-8,4 \cdot 10^5)^2 \cdot 6,4 + (-8,7 \cdot 10^5)^2 \cdot 6,4 +}{T_{\text{ц}}}} \\ &\sqrt{\frac{(-9,1 \cdot 10^5)^2 \cdot 6,4 + (-9,5 \cdot 10^5)^2 \cdot 6,4 + (-9,7 \cdot 10^5)^2 \cdot 0,61 + (-0,5 \cdot 10^5)^2 \cdot 5,79}{T_{\text{ц}}}} = \\ &= 3,7 \cdot 10^{-4} \cdot 996127,96 = 368,567 \text{ Нм} \end{aligned}$$

Номінальний момент вибраного двигуна:

$$M_{\text{н}} = \frac{P_{\text{н}}}{\omega_{\text{н}}} , \quad (2.26)$$

$$M_{\text{н}} = \frac{37000}{60,2} = 614,6 \text{ Нм} .$$

Оскільки виконується умова: $M_{\text{н}} \leq M_{\text{д екв}}$ ($368,6 < 614,6$), 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 $2,5 \cdot M_{\text{н}}$, тобто повинно виконуються співвідношення

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} \leq 2,5 M_{\text{н}} , \quad (2.27)$$

$$\frac{M_{\text{рез max}}}{\eta_{\text{п}} i_{\text{п}} k} = \frac{1777445,9}{0,65 \cdot 2075,86 \cdot 2} = 658,6 \text{ Нм} ,$$

$$2,5 \cdot M_{\text{н}} = 2,5 \cdot 614,6 = 1536,5 \text{ Нм} .$$

Отже, вибраний двигун по перевантажувальній здатності задовольняє вимогам.

2.6 Вибір основного електроустаткування електроприводу

На роторних екскаваторах найбільш широкое використання мають тиристорні перетворювачі типа БУВ і ШУВ, Функціональні схеми цих перетворювачів, які називаються так само станціями управління, показані на рис.2.2.

Тиристорні перетворювачі складаються з наступних блоків: ВК – випрямний комплект (для реверсивної схеми їх два: ВКВ і ВКН; ТОР – токообмежувачий реактор; А – автоматичний вимикач; ДТ – датчик току; БЗ – блок захисту; СІФУ – система імпульсний-фазового управління; РП – регулюючий пристрій;

ЛП – логічний пристрій (є тільки в реверсивних ТП); БП – блок живлення; В – (вентилятор входить в комплект, якщо охолодження тиристорів примусове).

У реверсивних ТП БУВ і ШУВ використаний принцип роздільного управління комплектами тиристорів, причому для управління ними використовується одна і та ж СІФУ. Дані станції управління призначені для роботи в замкнутах системах управління, що використовують принцип регулювання по відхиленню координат приводу-напруги, швидкості, струму і т.д. від заданого значення. Відповідно на вхід станції подається сигнал, пропорційний відхиленню, $\Delta U = U_3 - U_{ззш}$ де U_3 – задаюча напруга, пропорційна бажаному значенню регульованої координати, а $U_{ззш}$ – напруга зворотного зв'язку, пропорційна її реальному значенню.

У неререверсивній схемі ΔU подається безпосередньо на вхід РП, основним значенням якого є посилення сигналу ΔU і подача його на вхід СІФУ. При замиканні системи зворотним зв'язком за швидкістю або напрузі підвищується жорсткість робочої частини механічної характеристики. Крім того, замикання САУ електроприводу зворотним зв'язком необхідне для лінеаризації характеристик в зоні переривистих струмів, а також для нормальної роботи реверсивного приводу що має ТП з роздільним способом управління комплектами вентилів.

У даних ТП використовується трифазна реверсивна мостова схема випрямлення, відсутній анодний трансформатор і використовується токообмежувачий реактор.

Вибираємо тиристорний перетворювач ШУВ 3601-44. Номінальні дані перетворювача: потужність: 147 кВт; випрямлена напруга: 460 В; випрямлений струм: 320 А.

Індуктивність токообмежувачого реактора: $L_{\text{тор}} = 0,174 \cdot 10^{-3} \text{ Гн}$.

2.7 Розрахунок статичних характеристик розімкненої і замкненої систем

Схема силового ланцюга випрямленого струму системи ТП – Д показана на рис. 2.3.

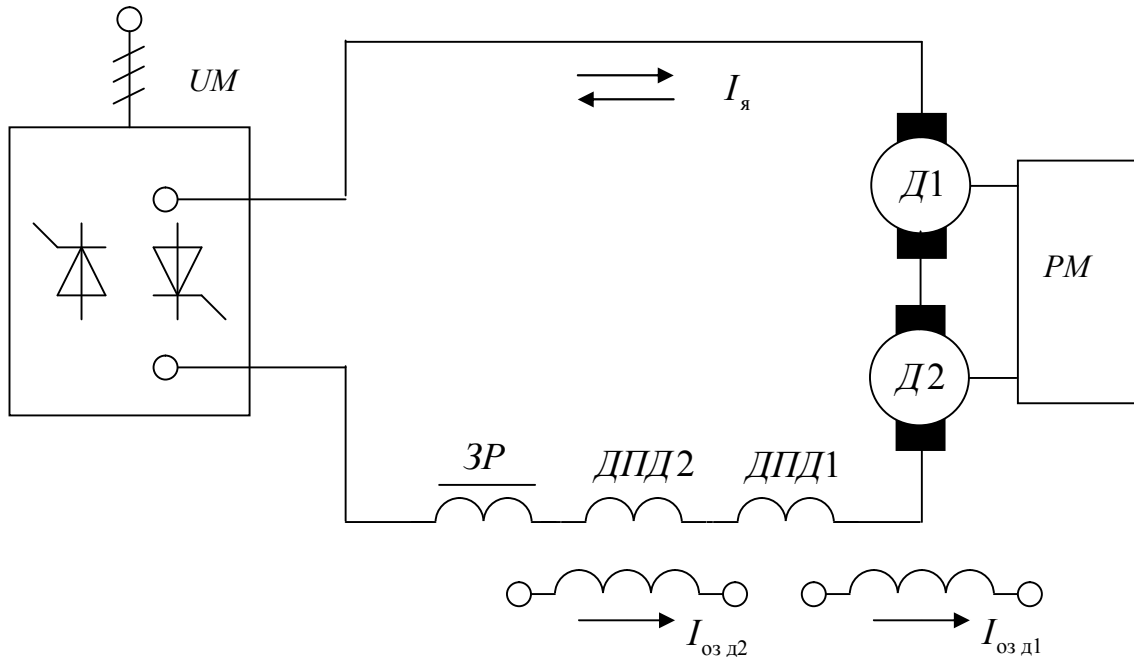


Рисунок 2.3 – Схема силового ланцюга випрямленого струму системи ТП-Д

На схемі позначено: UM – підсилювач потужності – тиристорний перетворювач; CP – згладжуючий реактор; PM – робочий механізм; $M1$, $M2$ – привідні двигуни механізму розвороту екскаватора; $ДПД1$, $ДПД2$ – обмотки додаткових полюсів двигунів; $I_я$ – струм якірного ланцюга; $I_{зб д1}$, $I_{зб д2}$ – струми в обмотках збудження двигунів $M1$, $M2$.

Як впливає з схеми, якірні обмотки двигунів і тиристорного перетворювача сполучені послідовно і утворюють якірний ланцюг.

Обчислимо сумарний опір якірного ланцюга $R_{я\Sigma}$:

$$R_{я\Sigma} = k \cdot R_{я\Sigma д} + R_{е\text{ тп}} + R_{зр} , \quad (2.28)$$

$R_{я\Sigma д}$ – сумарний опір якірного ланцюга двигуна:

$$R_{я\Sigma д} = k_t \left(R_{я д} + R_{дп д} + R_{ко д} \right) ; \quad (2.29)$$

$R_{я д}$, $R_{ко д}$, $R_{дп д}$ – опір обмотки якоря, компенсаційної і додаткових полюсів двигуна, вибрані двигуни не компенсовані, тобто $R_{ко д} = 0$;

k_t – температурний коефіцієнт $k_t = 1,36$

$$R_{я\Sigma д} = 1,36 \cdot (0,034 + 0,02) = 0,073 \text{ Ом}$$

$R_{е\text{ тп}}$ - еквівалентний внутрішній опір тиристорного перетворювача, для мостової схеми визначається виразом:

$$R_{е\text{ тп}} = 2 \cdot \left(R_t + \frac{3X_t}{2\pi} \right), \quad (2.30)$$

де $R_t = R_{\text{тор}}$ - активний опір токообмежуючого реактора (активним опором токообмежуючого реактора можна знехтувати).

Індуктивний опір реактора $X_t = X_{\text{тор}}$:

$$X_t = m \cdot \omega_1 \cdot L_{\text{тор}}, \quad (2.31)$$

$m = 6$ - пульсність мостової схеми випрямлення.

ω_1 – кутова частота живлячої мережі.

$$\omega_1 = 2\pi \cdot f_1 = 2 \cdot 3,14 \cdot 50 = 314.$$

$$X_t = 6 \cdot 314 \cdot 0,174 \cdot 10^{-3} = 0,33 \text{ Ом}.$$

$$R_{\text{еп}} = 2 \cdot \frac{3 \cdot 0,33}{2 \cdot 3,14} = 0,16 \text{ Ом}.$$

$R_{\text{зр}}$ - активний опір включеного в якірний ланцюг згладжуючого реактора.

Визначимо необхідність включення в якірний ланцюг двигунів згладжуючого реактора для обмеження рівня пульсацій ланцюга.

Для забезпечення заданого рівня пульсацій, імпульсна згладжуючого реактора повинна бути:

$$L_{\text{зр}} \geq \frac{\sqrt{2U_{\text{д}}}}{m\omega_1(0,02-0,15)I_{\text{н}}} - kL_{\text{яд}} - L_{\text{т}}, \quad (2.32)$$

де $U_{\text{д}}$ – дієве значення змінної складової випрямленої напруги, В, яке при $m = 6$ можна прийняти

$$U_{\text{д}} = 0,36 \cdot E_{\text{д0}}, \quad (2.33)$$

$E_{\text{д0}}$ – середнє значення випрямленої ЕДС тиристорного перетворювача при куті управління $\alpha = 0$; для мостової схеми випрямлення

$$E_{\text{д0}} = 2,34 U_{\text{ф}}, \quad (2.34)$$

$U_{\text{ф}}$ – дієве значення фазної напруги живлення тиристорного перетворювача, В

$$E_{\text{д0}} = 2,34 \cdot 220 = 514,8 \text{ В}.$$

$$U_{\text{д}} = 0,36 \cdot 514 = 185,33 \text{ В}.$$

$L_{\text{яд}}$ - індуктивність якірного ланцюга (оскільки привід двохдвигунний, то беремо подвоєне значення індуктивності якірного ланцюга одного двигуна)

$$L_{\text{яд}} = \gamma \frac{U_{\text{н}}}{I_{\text{н}} \omega_{\text{н}} p}, \quad (2.35)$$

γ – коефіцієнт, рівний $\gamma = 0,25$ для компенсованої машини і $\gamma = 0,6$ для некомпенсованої машини;

U_n , I_n , ω_n – номінальні величини напруги, струму і швидкості двигуна відповідно;

p – число пар полюсів двигуна

$$L_{яд} = 0,6 \cdot \frac{220}{192} \cdot \frac{1}{4 \cdot 60,2} = 0,003 Гн.$$

L_T - еквівалентна індуктивність тиристорного перетворювача

$$L_T = \frac{m}{3} \cdot L_{top}, \quad (2.36)$$

$$L_T = \frac{6}{3} \cdot \frac{0,33}{314} = 0,0021 Гн.$$

$$L_{зр} \geq \frac{\sqrt{2} \cdot 186,33}{6 \cdot 314 \cdot 0,15 \cdot 192} - 2 \cdot 0,003 - 0,0021 = -0,00027 Гн.$$

Отримано негативне число, тому згладжуючий реактор не потрібний, оскільки індуктивності двигунів і тиристорного перетворювача досить для обмеження пульсації струму якірного ланцюга.

Отже, активний опір якірного ланцюга обчислюється за виразом:

$$R_{я\Sigma} = k \cdot R_{я\Sigmaд} + R_{e\piп} + R_{зр}, \quad (2.37)$$

$$R_{я\Sigma} = 2 \cdot 0,073 + 0,16 = 0,307 Ом.$$

Сумарна індуктивність силового ланцюга електроприводу

$$L_{я\Sigma} = k L_{яд} + L_T, \quad (2.38)$$

$$L_{я\Sigma} = 2 \cdot 0,003 + 0,0021 = 0,0081 Гн.$$

Якщо розглядати ЕДС перетворювача тиристора $E_{\text{тп}}$ як джерело живлення двигунів, то рівняння електромеханічної і механічної характеристик двигуна матимуть вигляд:

$$\omega_{\text{д}} = \frac{E_{\text{тп}}}{k k_{\text{д}} \Phi_{\text{нд}}} - I_{\text{я}} \frac{R_{\text{я}\Sigma}}{k k_{\text{д}} \Phi_{\text{нд}}} = \omega_0 - \Delta\omega_{\text{ст}} ; \quad (2.39)$$

$$\omega_{\text{д}} = \frac{E_{\text{тп}}}{k k_{\text{д}} \Phi_{\text{нд}}} - M_{\text{д}} \frac{R_{\text{я}\Sigma}}{k (k_{\text{д}} \Phi_{\text{нд}})^2} = \omega_0 - \Delta\omega_{\text{ст}} \quad (2.40)$$

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

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

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

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

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

$$c = k_{\text{д}} \Phi_{\text{нд}} = \frac{220 - 192 \cdot 0,074}{60,2} = 3,42 . \quad (2.41)$$

Оскільки характеристики мають вид прямих ліній, то для їх побудови досить визначити координати двох точок, наприклад точки холостого ходу і точки, відповідній номінальному навантаженню. У якості максимального значення $E_{\text{өі}}$ приймемо подвоєну номінальну напругу електродвигуна, тобто $E_{\text{тп1}} = E_{\text{тп max}} = k U_{\text{н}} = 2 \cdot 220 = 440 \text{ В}$. Будуємо характеристики також при $E_{\text{тп2}} = 0,75 \cdot E_{\text{тп max}} = 0,75 \cdot 440 = 330 \text{ В}$ і $E_{\text{тп2}} = 0,5 \cdot E_{\text{тп max}} = 0,5 \cdot 440 = 220 \text{ В}$.

Швидкість ідеального холостого ходу визначається виразом:

$$\omega_0 = \frac{E_{\text{тп}}}{k k_{\text{д}} \Phi_{\text{нд}}}, \quad (2.42)$$

Визначимо ω_0 при $E_{\text{тп1}} = 440 \text{ В}$, $E_{\text{тп2}} = 330 \text{ В}$ і $E_{\text{тп3}} = 220 \text{ В}$

$$\omega_{01} = \frac{440}{2 \cdot 3,42} = 64,33 \text{ с}^{-1}.$$

$$\omega_{02} = \frac{330}{2 \cdot 3,42} = 48,24 \text{ с}^{-1}.$$

$$\omega_{03} = \frac{220}{2 \cdot 3,42} = 32,16 \text{ с}^{-1}.$$

Точка 1 -

$$I = 0, \quad M = 0$$

При номінальному навантаженні двигунів

$$I_{\text{я}} = I_{\text{н}} = 192 \text{ А}, \quad M_{\text{д}} = k M_{\text{н}} = k k_{\text{д}} \Phi_{\text{н}} I_{\text{н}} = 2 \cdot 3,42 \cdot 192 = 1313,3 \text{ Нм}$$

Визначимо швидкість при номінальному навантаженні і $E_{\text{тп1}} = 440 \text{ В}$,
 $E_{\text{тп2}} = 330 \text{ В}$, $E_{\text{тп3}} = 220 \text{ В}$

$$\omega_1 = \frac{440}{2 \cdot 3,42} - 192 \cdot \frac{0,307}{2 \cdot 3,42} = 55,7 \text{ с}^{-1}.$$

$$\omega_2 = \frac{330}{2 \cdot 3,42} - 192 \cdot \frac{0,307}{2 \cdot 3,42} = 39,64 \text{ с}^{-1}.$$

$$\omega_3 = \frac{220}{2 \cdot 3,42} - 192 \cdot \frac{0,307}{2 \cdot 3,42} = 23,56 \text{ с}^{-1}.$$

Графіки електромеханічних і механічних характеристик розімкненої системи ТП-Д приведені на рис. 2.4.(пунктирні прямі).

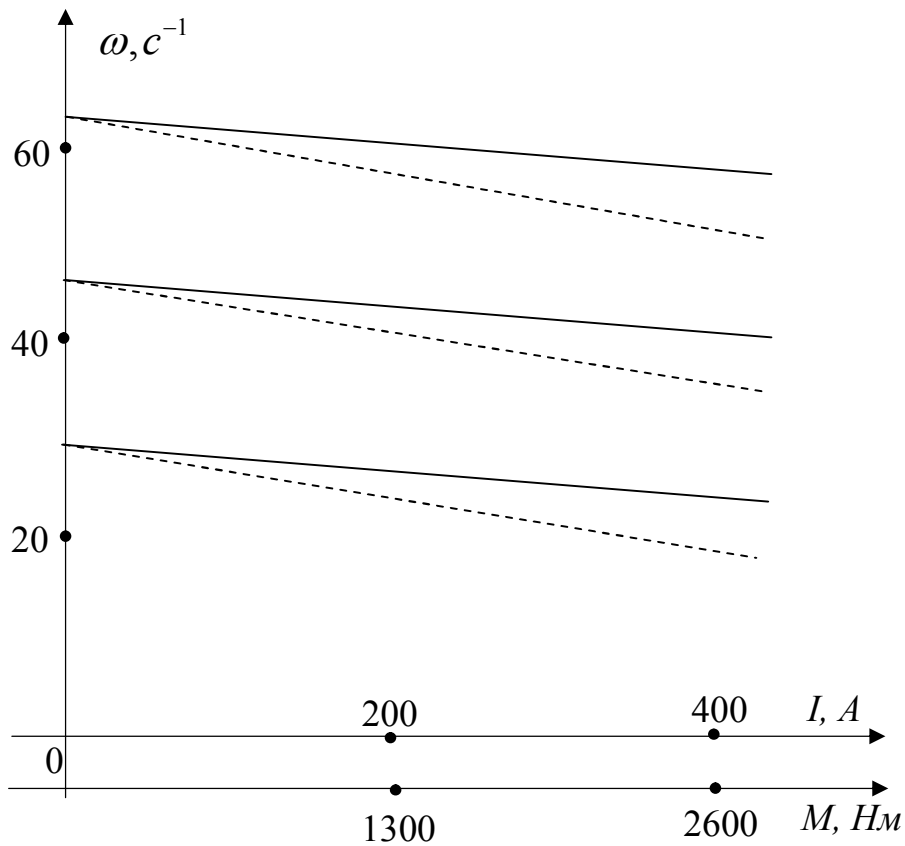


Рисунок 2.4 - Механічні і електромеханічні характеристики розімкненої і замкненої системи

До електроприводів більшості основних механізмів роторних екскаваторів пред'являються підвищені вимоги відносно жорсткості робочої частини механічної характеристики і обмеження максимального значення моменту, що розвивається. Забезпечити вказані вимоги можна тільки в замкнутих системах автоматичного управління електроприводом. На рис. 2.5 зображена функціональна схема вживаної на роторних екскаваторах системи, яка має тільки один зворотний зв'язок – по напрузі тиристорного перетворювача. На схемі для спрощення подальших досліджень показаний один еквівалентний двигун.

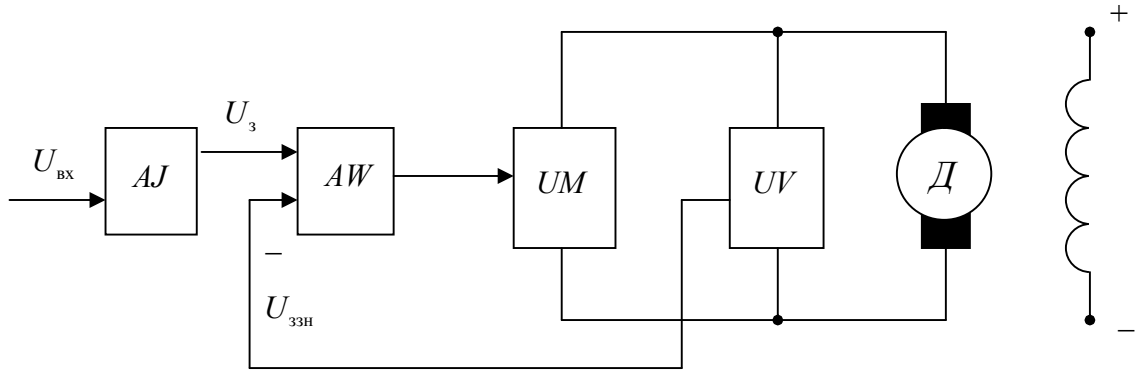


Рисунок 2.5 – Функціональна схема замкненої системи ТП-Д із зворотним зв'язком по напрузі

Позначення, що прийняті на схемі: AJ – задатчик інтенсивності; AW – підсилювач, що підсумовує; UI – тиристорний перетворювач; UV – датчик напруги спільно з ділянкою напруги.

Як показано в [59], рівняння статичної електромеханічної характеристики цієї системи має вид :

$$\omega_d = U_3 \frac{k_\Sigma / (k c)}{1 + k_\Sigma k_{ззп}} - I_\text{я} \left(\frac{R_{\text{я}\Sigma}}{k c} - \frac{R_{\text{е пр}} k_\Sigma k_{ззп} / (k c)}{1 + k_\Sigma k_{ззп}} \right) = \omega_{\text{д}0} - \Delta\omega_{\text{ст}1}, \quad (2.43)$$

де $k_\Sigma = k_{AW} k_{\text{тп}}$ – сумарний коефіцієнт посилення, рівний добутку коефіцієнтів посилення k_{AW} суматора AW і тиристорного перетворювача $k_{\text{тп}}$.

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

Коефіцієнт посилення тиристорного перетворювача для синусоїдальної напруги порівняння з амплітудою $U_\text{м}$ приблизно можна визначити як

$$k_{\text{тп}} = \frac{E_{d0}}{U_\text{м}}. \quad (2.44)$$

Приймаємо $U_\text{м} = 15 \text{ В}$.

$$k_{\text{тп}} = \frac{514,8}{15} = 34,32,$$

$$k_{\Sigma} = 0,1 \cdot 34,32 = 3,43,$$

$k_{\text{ззн}}$ – коефіцієнт зворотного зв'язку по напрузі, величину $k_{\text{ззн}}$ слід визначити виходячи з необхідного статичного перепаду швидкості $\Delta\omega_{\text{ст1}}$ при номінальному струмі двигуна:

$$\Delta\omega_{\text{ст1}} = I_{\text{н}} \left(\frac{R_{\text{я}\Sigma}}{k \ c} - \frac{R_{\text{епр}} k_{\Sigma} k_{\text{ззн}} / (k \ c)}{1 + k_{\Sigma} k_{\text{ззн}}} \right), \quad (2.45)$$

звідки

$$k_{\text{ззн}} = \left(\frac{I_{\text{н}} R_{\text{я}\Sigma} / (k \ c) - \Delta\omega_{\text{ст1}}}{k_{\Sigma} \Delta\omega_{\text{ст1}} + I_{\text{н}} k_{\Sigma} \cdot R_{\text{епр}} / (k \ c)} \right). \quad (2.46)$$

Значення $\Delta\omega_{\text{ст1}}$ приймаємо рівним половині $\Delta\omega_{\text{ст}}$ в розімкненій системі. Після визначення $k_{\text{ззн}}$ слід проаналізувати можливість його фізичної реалізації в розробленій схемі.

$$\Delta\omega_{\text{ст1}} = \frac{1}{2} \cdot \Delta\omega_{\text{ст}} = \frac{1}{2} \cdot I_{\text{н}} \frac{R_{\text{я}\Sigma}}{k \ k_{\text{д}} \Phi_{\text{нд}}}, \quad (2.47)$$

$$\Delta\omega_{\text{ст1}} = \frac{1}{2} \cdot \Delta\omega_{\text{ст}} = \frac{1}{2} \cdot 192 \cdot \frac{0,307}{2 \cdot 3,42} = 4,3$$

$$k_{\text{ззн}} = \frac{192 \cdot 0,307 / 2 \cdot 3,42 - 4,3}{3,43 \cdot 4,3 + (192 \cdot 3,43 \cdot 0,16) / 2 \cdot 3,42} = 0,71$$

Величину $U_{\text{з max}}$ необхідно визначити з умови отримання при $U_{\text{з max}}$ швидкості ідеального холостого ходу розімкненої системи при $E_{\text{тп}} = k \cdot U_{\text{н}}$

$$\omega_{\text{д0}} = U_{\text{з max}} \frac{k_{\Sigma} / (k \ c)}{1 + k_{\Sigma} k_{\text{ззн}}} = \frac{U_{\text{н}}}{c}. \quad (2.48)$$

$$U_{\text{з max}} = \frac{U_{\text{н}} (1 + k_{\Sigma} k_{\text{ззн}})}{k_{\Sigma} / k} \quad (2.49)$$

$$U_{3 \max} = \frac{2 \cdot 220 \cdot (1 + 3,43 \cdot 0,71)}{3,43} = 371 \text{ В}$$

Рівняння електромеханічної характеристики замкнутої системи

$$\omega = U_3 \frac{3,43 / (2 \cdot 3,42)}{1 + 3,43 \cdot 0,71} - I \left(\frac{0,307}{2 \cdot 3,42} - \frac{0,146 \cdot 3,43 \cdot 0,71 / (2 \cdot 3,42)}{1 + 3,43 \cdot 0,71} \right) =$$

$$= U_3 \cdot 0,1733 - I_{\text{я}} \cdot 0,018.$$

Будуємо статичні електромеханічні і механічні характеристики при $U_{31} = U_{3 \max} = 371 \text{ В}$, а також при $U_{32} = 0,75 U_{3 \max} = 0,55 \cdot 242,7 = 278 \text{ В}$ і $U_{33} = 0,5 U_{3 \max} = 0,5 \cdot 371 = 185,5 \text{ В}$.

Швидкості ідеального холостого ходу замкнутої системи такі ж, як і для розімкнутої системи, тобто $\omega_{01} = 64,33 \text{ с}^{-1}$, $\omega_{02} = 48,24 \text{ с}^{-1}$ і $\omega_{03} = 32,16 \text{ с}^{-1}$

Визначимо швидкість, що відповідають номінальному навантаженню $U_{31} = U_{3 \max} = 371 \text{ В}$, а також при $U_{32} = 0,75 U_{3 \max} = 0,55 \cdot 242,7 = 278 \text{ В}$ і $U_{33} = 0,5 U_{3 \max} = 0,5 \cdot 371 = 185,5 \text{ В}$.

$$\omega_1 = 371 \cdot 0,1733 - 192 \cdot 0,018 = 60,03 \text{ с}^{-1}.$$

$$\omega_2 = 278 \cdot 0,1733 - 192 \cdot 0,018 = 43,9 \text{ с}^{-1}.$$

$$\omega_3 = 185,5 \cdot 0,1733 - 192 \cdot 0,018 = 17,86 \text{ с}^{-1}.$$

Характеристики замкнутої системи показані на рис.2.2 суцільними лініями.

2.10 Дослідження динамічних характеристик системи

Для отримання рівнянь динаміки і передавальних функцій системи складемо алгоритмічну схему системи, представлену на рис.2.6.

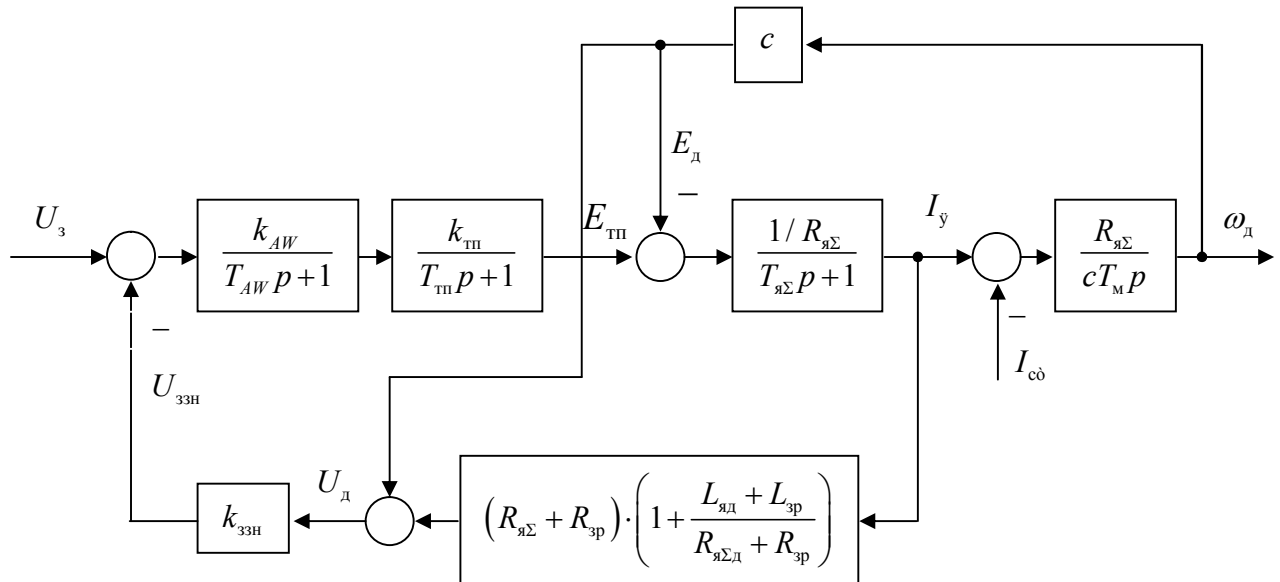


Рисунок 2.6 – Алгоритмічна схема системи автоматичного управління електроприводом ТП-Д

Дана схема може бути приведена до алгоритмічної схеми рис.2.7, яка відповідає системі Г-Д, детально описаній в [59]. Для цього потрібно виконати наступне:

- замість постійної часу генератора T_r підставити постійну часу тиристорного перетворювача $T_{тп}$;
- замість коефіцієнта посилення ланцюга зворотного зв'язку за швидкістю $k_{ззн}$ підставити $k_{ззн} \approx k_{ззн} c$;
- замість постійної часу ланцюга гнучкого зворотного зв'язку по струму $T_{зт}$ підставити $T_{зт} = k_{ззн} (2 L_{яд} + L_{зр})$;

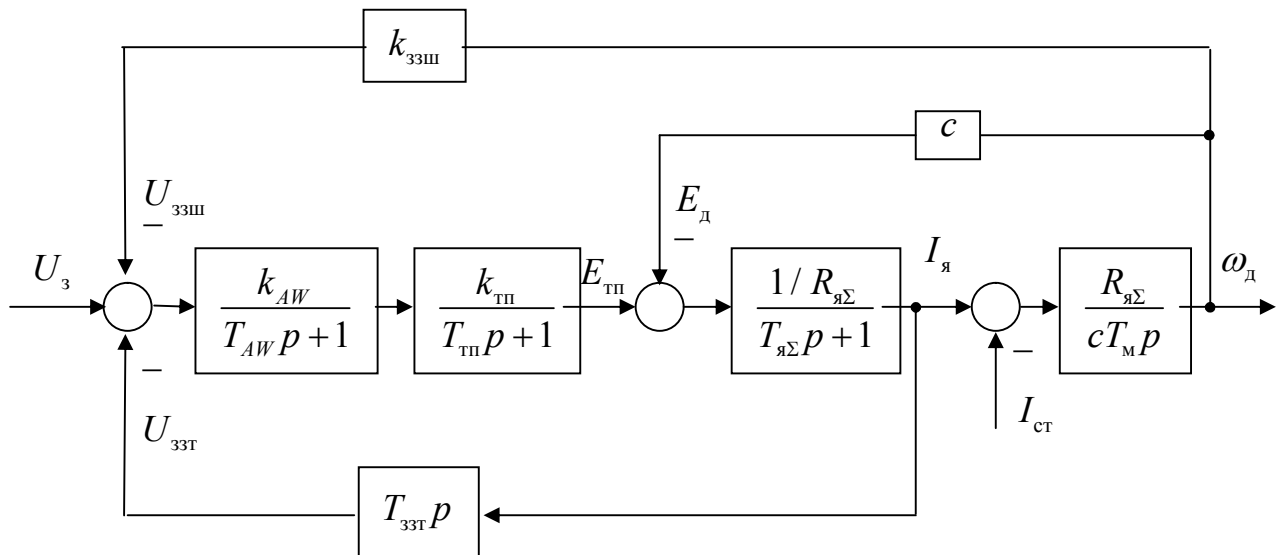


Рисунок 2.7 – Перетворена алгоритмічна схема системи автоматичного управління електроприводом ТП-Д

Впливом жорсткого ЗЗ по струму, що виникає за рахунок присутності в сигналі $U_{33П}$ величини $k_{33П} (k R_{яΣД} + R_{зр}) I_я$, яка викликає деяке зменшення жорсткості механічної характеристики і трохи позначається на динаміці, також можна знехтувати.

По відношенню до системи Г-Д динаміка системи ТП-Д має істотні відмінності, пов'язані з незрівнянно вищою швидкодією ТП. З метою обмеження динамічних навантажень в конструкціях і механізмах екскаваторів управління проводиться через задатчик інтенсивності AJ . Можливості настройки якості перехідних процесів в заданій структурі САУ обмежені, оскільки можна варіювати тільки один ЗЗ по напрузі ТП, яка одночасно впливає на статичні і динамічні характеристики приводу. Зазвичай приймають невеликі значення $k_{33П}$, що декілька знижує жорсткість статичної характеристики, проте сприятливо впливає на динаміку системи. У попередньому розділі розрахунок статичних характеристик був проведений для величини $k_{33П} = 0,71$. Надалі проведемо розрахунок динамічних характеристик системи для декількох значень $k_{33П}$ і виберемо значення, при якому показники якості перехідних процесів мають найкращі значення.

Визначимо значення параметрів схеми:

$T_{\text{тп}}$ – постійна часу тиристорного перетворювача; приймаємо $T_{\text{тп}} = 0,005 \text{ с}$.

$k_{\text{ззш}}$ – коефіцієнта посилення ланцюга зворотного зв'язку за швидкістю

$$k_{\text{ззш}} = k_{\text{ззн}} k c, \quad (2.50)$$

$$k_{\text{ззш}} = k_{\text{ззн}} k c = 0,71 \cdot 2 \cdot 3,42 = 4,86.$$

$T_{\text{ззт}}$ – постійна часу ланцюга гнучкого зворотного зв'язку по струму

$$T_{\text{ззт}} = k_{\text{ззн}} (k L_{\text{яд}} + L_{\text{зр}}), \quad (2.51)$$

$$T_{\text{ззт}} = 0,71 \cdot (2 \cdot 0,003 + 0,0021) = 0,006 \text{ с}. \quad (2.52)$$

$T_{\text{м}}$ - електромеханічна постійна часу електроприводу

$$T_{\text{м}} = \frac{J_{\Sigma} R_{\text{я}\Sigma}}{(k c)^2}, \quad (2.53)$$

де J_{Σ} – сумарний момент інерції якоря двигуна і приведений до його валу момент інерції механізмів

$$J_{\Sigma} = 2J_{\text{д}} + \frac{J_{\text{пл}}}{i_{\text{п}}^2}; \quad (2.54)$$

$$J_{\Sigma} = 2 \cdot 2 + \frac{17 \cdot 10^7}{2075,86^2} = 4 + 39,4 = 43,4 \text{ кг} \cdot \text{м}^2.$$

$$T_{\text{м}} = \frac{43,4 \cdot 0,307}{(2 \cdot 3,42)^2} = 0,28 \text{ с}.$$

$T_{\text{я}\Sigma}$ – електромагнітна постійна часу електроприводу

$$T_{\text{я}} = \frac{L_{\text{я}\Sigma}}{R_{\text{я}\Sigma}} \quad (2.55)$$

$$T_{\text{я}} = \frac{0,0081}{0,307} = 0,026 \text{ с.}$$

Для отримання передавальних функцій системи по задаючій і збурюючій діях проведемо перетворення алгоритмічної схеми рис. 2.7. З огляду на те, що AW виконаний на ОУ, його постійною часу T_{AW} , вимірюваною мікросекундами нехтуємо. Тоді ланки, що відповідають підсумовуючому підсилювачу і тиристорному перетворювачу можна об'єднати в одну еквівалентну ланку з передавальною функцією

$$W(p) = \frac{k_{\Sigma}}{T_{\text{тп}}p + 1}. \quad (2.56)$$

Отримаємо спрощену алгоритмічну схему, показану на рис. 2.8. Всі ланки для зручності подальших перетворень пронумеровані від 1 до 6.

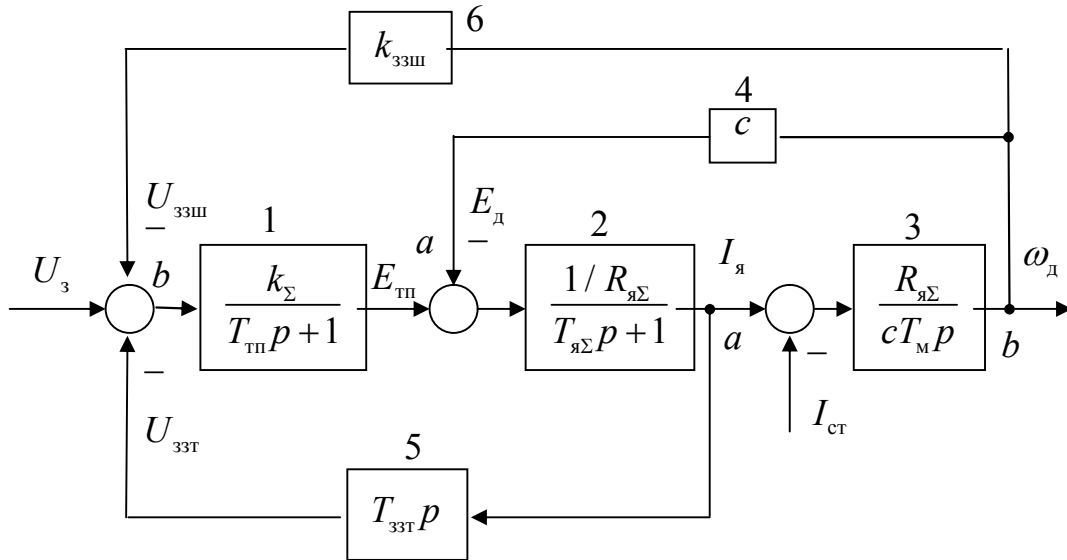


Рисунок 2.8 – Спрощена алгоритмічна схема системи

Перемикаємо зворотний зв'язок по ЕДС двигуна з точки *a* в точку *b*, включаючи при цьому в зворотний зв'язок ланку з передатною функцією

$$W_7(p) = \frac{1}{W_1(p)} = \frac{1}{\frac{k_\Sigma}{T_{\text{тп}}p + 1}} = \frac{T_{\text{тп}}p + 1}{k_\Sigma}. \quad (2.57)$$

Отримаємо алгоритмічну схему, показану на рис.2.9 із зворотними зв'язками, що не перехрещуються.

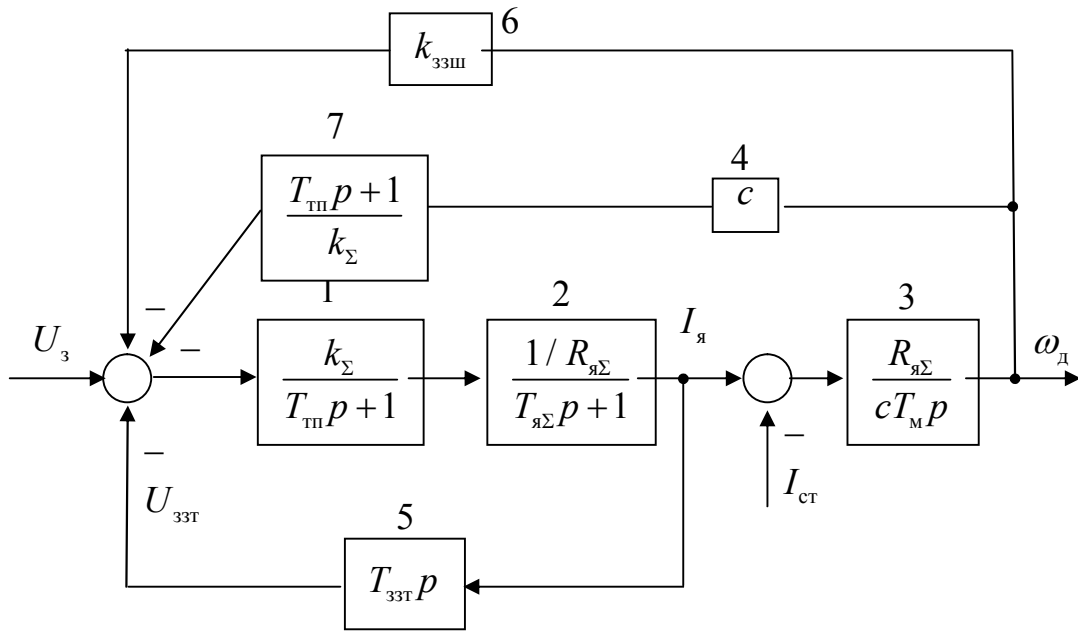


Рисунок 2.9 – Схема, отримана перетворенням алгоритмічної схеми, зображеної на рис. 2.8

Об'єднаємо ланки 1 і 2, а також 4 і 7

$$W_{1,2}(p) = W_1(p) \cdot W_2(p) = \frac{k_\Sigma}{T_{\text{тп}}p + 1} \cdot \frac{1/R_{\text{я}\Sigma}}{T_{\text{я}\Sigma}p + 1} = \frac{k_\Sigma / R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1)}, \quad (2.58)$$

$$W_{4,7}(p) = W_4(p) \cdot W_7(p) = c \cdot \frac{T_{\text{тп}}p + 1}{k_\Sigma} = \frac{c(T_{\text{тп}}p + 1)}{k_\Sigma}. \quad (2.59)$$

Отримаємо схему, показану на рис. 2.10.

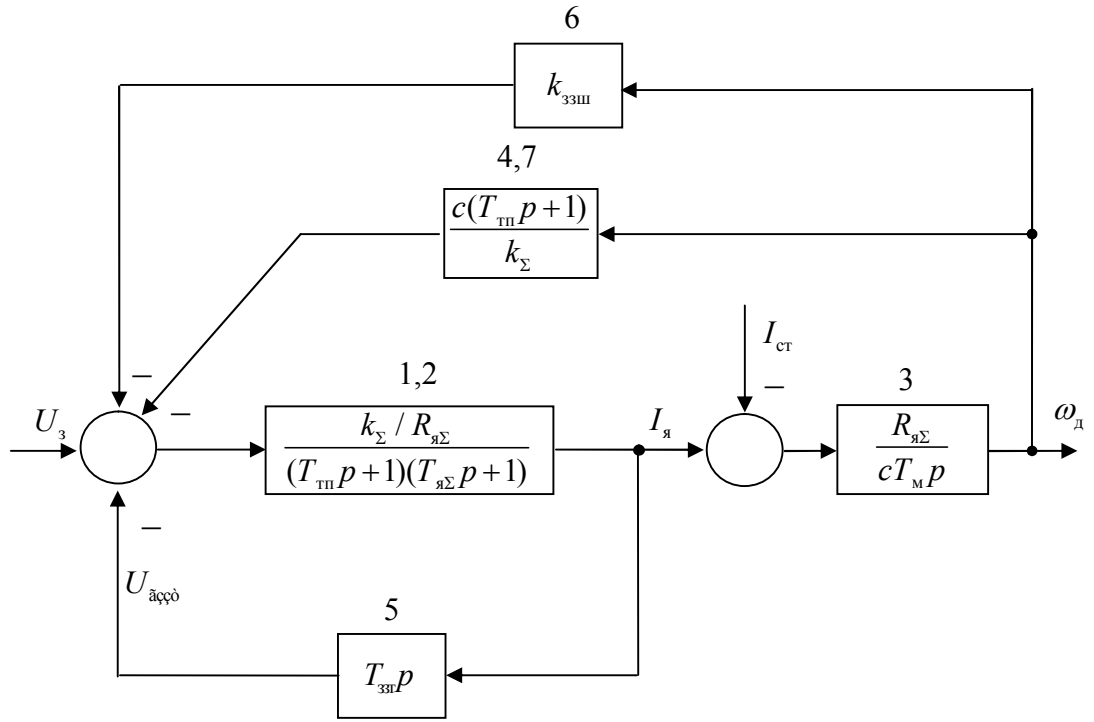


Рисунок 2.10 – Перетворена алгоритмічна схема системи,
показаної на рис. 2.9

Об'єднаємо ланки 4,7 і 6 , а також виконаємо з'єднання ланок 1,2 і 5

$$W_{4,7,6}(p) = W_{4,7}(p) + W_6(p) = \frac{c(T_{тп}p + 1)}{k_{\Sigma}} + k_{33ш} . \quad (2.60)$$

$$\begin{aligned} W_{1,2,5}(p) &= \frac{W_{1,2}(p)}{1 + W_{1,2}(p) \cdot W_5(p)} = \frac{\frac{k_{\Sigma} / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1)}}{1 + \frac{k_{\Sigma} / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1)} \cdot T_{33т}p} = \\ &= \frac{\frac{k_{\Sigma} / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1)}}{\frac{(T_{тп}p + 1)(T_{я\Sigma}p + 1) + T_{33т}p k_{\Sigma} / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1)}} = \frac{k_{\Sigma} / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1) + T_{33т}p k_{\Sigma} / R_{я\Sigma}} . \quad (2.61) \end{aligned}$$

Одержимо схему, показану на рис.2.11.

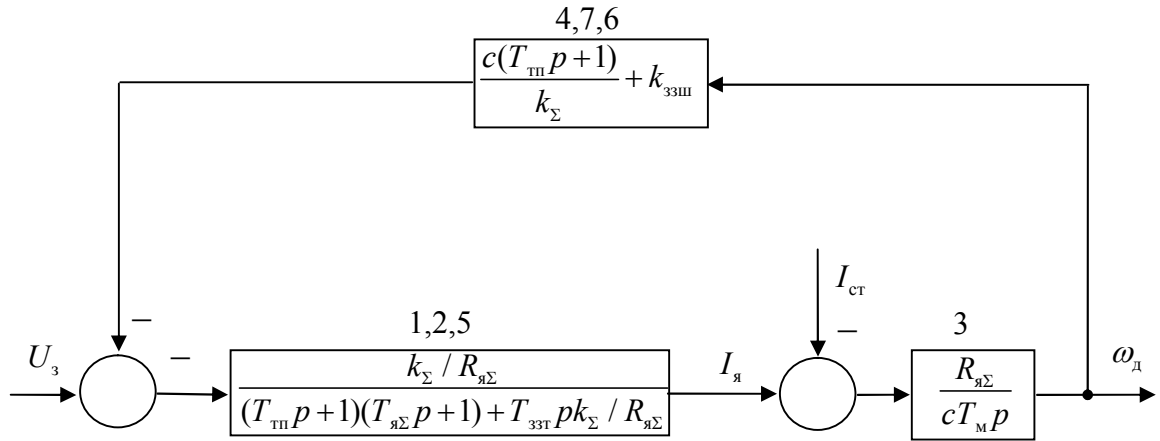


Рисунок 2.11- Перетворена алгоритмічна схема системи,
показаної на рис. 2.10

Використовуючи схему рис. 2.11 знайдемо передатні функції системи. При виведенні передавальних функцій по задаючій дії вважаємо $I_{ст} = 0$, а при виведенні передавальних функцій по збурюючій дії вважаємо $U_3 = 0$.

Передавальна функція для швидкості по задаючій дії.

$$\begin{aligned}
 W_{\omega_d, U_3}(p) &= \frac{\omega_d(p)}{U_3(p)} = \\
 &= \frac{\frac{k_\Sigma / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1) + T_{зт}pk_\Sigma / R_{я\Sigma}} \cdot \frac{R_{я\Sigma}}{cT_мp}}{1 + \frac{k_\Sigma / R_{я\Sigma}}{(T_{тп}p + 1)(T_{я\Sigma}p + 1) + T_{зт}pk_\Sigma / R_{я\Sigma}} \cdot \frac{R_{я\Sigma}}{cT_мp} \cdot \left[\frac{c(T_{тп}p + 1)}{k_\Sigma} + k_{зтш} \right]} = \\
 &= \frac{\frac{k_\Sigma}{[(T_{тп}p + 1)(T_{я\Sigma}p + 1) + T_{зт}pk_\Sigma / R_{я\Sigma}] \cdot cT_мp}}{cT_мp \cdot k_\Sigma \cdot (T_{тп}p + 1)(T_{я\Sigma}p + 1) + cT_мp \cdot k_\Sigma \cdot T_{зт}pk_\Sigma / R_{я\Sigma} + c(T_{тп}p + 1) + k_\Sigma^2 \cdot k_{зтш}} = \\
 &= \frac{R_{я\Sigma} \cdot k_\Sigma}{cT_мp \cdot R_{я\Sigma} (T_{тп}p + 1)(T_{я\Sigma}p + 1) + cT_мp \cdot p \cdot k_\Sigma + R_{я\Sigma} \cdot k_\Sigma \cdot k_{зтш} + R_{я\Sigma} \cdot cT_{тп}p + R_{я\Sigma} \cdot c}
 \end{aligned}$$

$$\begin{aligned}
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p \cdot R_{я\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p \cdot p \cdot k_{\Sigma} + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} \cdot cT_{\Sigma} p + R_{я\Sigma} \cdot c} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p \cdot R_{я\Sigma} (T_{\Sigma} p \cdot T_{я\Sigma} p + 1) + cT_{\Sigma} p \cdot T_{33\text{т}} p \cdot k_{\Sigma} + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} \cdot cT_{\Sigma} p + R_{я\Sigma} \cdot c} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cT_{\Sigma} p R_{я\Sigma} T_{\Sigma} p T_{я\Sigma} p + cT_{\Sigma} p R_{я\Sigma} T_{\Sigma} p + cT_{\Sigma} p R_{я\Sigma} T_{я\Sigma} p + cT_{\Sigma} p \cdot R_{я\Sigma} + cT_{\Sigma} p T_{33\text{т}} p k_{\Sigma} +} \\
&\quad + R_{я\Sigma} \cdot k_{\Sigma} \cdot k_{33\text{ш}} + R_{я\Sigma} cT_{\Sigma} p + R_{я\Sigma}} = \\
&= \frac{R_{я\Sigma} \cdot k_{\Sigma}}{cR_{я\Sigma} T_{\Sigma} T_{\Sigma} T_{я\Sigma} p^3 + c(R_{я\Sigma} T_{\Sigma} T_{\Sigma} + R_{я\Sigma} T_{\Sigma} T_{я\Sigma} + k_{\Sigma} T_{\Sigma} T_{\Sigma}) p^2 + cR_{я\Sigma} (T_{\Sigma} + T_{\Sigma}) p + R_{я\Sigma} (k_{\Sigma} k_{33\text{ш}} + c)} = \\
&= \frac{k_{\Sigma} k_{\Sigma}}{T_{\Sigma} T_{\Sigma} T_{я\Sigma} p^3 + \left(T_{\Sigma} T_{\Sigma} + T_{я\Sigma} T_{\Sigma} + \frac{k_{\Sigma} T_{\Sigma} T_{33\text{т}}}{R_{я\Sigma}} \right) p^2 + (T_{\Sigma} + T_{\Sigma}) p + 1 + k_{\Sigma} k_{\Sigma} k_{33\text{ш}}} \cdot (2.62)
\end{aligned}$$

При виведенні передавальної функції прийнято позначення $k_{\Sigma} = \frac{1}{c}$ – коефіцієнт посилення еквівалентного двигуна.

Передавальна функція для швидкості по збурюючій дії.

$$\begin{aligned}
W_{\omega_{\Sigma}, I_{\Sigma}}(p) &= \frac{\omega_{\Sigma}(p)}{I_{\Sigma}(p)} = \frac{\frac{R_{я\Sigma}}{cT_{\Sigma} p}}{1 + \frac{R_{я\Sigma}}{cT_{\Sigma} p} \cdot \frac{k_{\Sigma} / R_{я\Sigma}}{(T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + T_{33\text{т}} p k_{\Sigma} / R_{я\Sigma}} \cdot \left[\frac{c(T_{\Sigma} p + 1)}{k_{\Sigma}} + k_{33\text{ш}} \right]} = \\
&= \frac{\frac{R_{я\Sigma}}{cT_{\Sigma} p}}{\frac{cT_{\Sigma} p R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p + R_{я\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{я\Sigma} k_{\Sigma} cT_{\Sigma} p + R_{я\Sigma} k_{\Sigma} c}{cT_{\Sigma} p \cdot R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p}} = \\
&= \frac{R_{я\Sigma}}{cT_{\Sigma} p} \cdot \frac{cT_{\Sigma} p \cdot R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p}{cT_{\Sigma} p R_{я\Sigma} k_{\Sigma} (T_{\Sigma} p + 1)(T_{я\Sigma} p + 1) + cT_{\Sigma} p k_{\Sigma}^2 T_{33\text{т}} p + R_{я\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{я\Sigma} k_{\Sigma} cT_{\Sigma} p + R_{я\Sigma} k_{\Sigma} c} =
\end{aligned}$$

$$\begin{aligned}
&= \frac{R_{\Sigma}^2 k_{\Sigma} (T_{\text{тп}} p R_{\Sigma} p + T_{\text{тп}} p + T_{\Sigma} p + 1) + k_{\Sigma}^2 T_{33\text{т}} p}{c T_{\text{м}} p R_{\Sigma} k_{\Sigma} (T_{\text{тп}} p T_{\Sigma} p + T_{\text{тп}} p + T_{\Sigma} p + 1) + c T_{\text{м}} p k_{\Sigma}^2 T_{33\text{т}} p + R_{\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{\Sigma} k_{\Sigma} c T_{\text{тп}} p + R_{\Sigma} k_{\Sigma} c} = \\
&= \frac{R_{\Sigma}^2 k_{\Sigma} T_{\text{тп}} p + T_{\Sigma} p + R_{\Sigma}^2 k_{\Sigma} T_{\text{тп}} p + R_{\Sigma}^2 k_{\Sigma} T_{\Sigma} p + R_{\Sigma}^2 k_{\Sigma} + k_{\Sigma}^2 T_{33\text{т}} p}{c T_{\text{м}} p R_{\Sigma} k_{\Sigma} T_{\text{тп}} p T_{\Sigma} p^3 + c k_{\Sigma} (R_{\Sigma} T_{\text{м}} T_{\text{тп}} + T_{\text{м}} T_{\Sigma} R_{\Sigma} + k_{\Sigma} T_{\text{м}} T_{\text{тп}}) p^2 + c R_{\Sigma} k_{\Sigma} (T_{\text{м}} + T_{\text{тп}}) p +} \\
&\quad + R_{\Sigma} k_{\Sigma} (k_{\Sigma} k_{33\text{ш}} + c) \\
&= \frac{R_{\Sigma}^2 T_{\text{тп}} T_{\Sigma} p^3 + (R_{\Sigma}^2 T_{\text{тп}} + R_{\Sigma}^2 T_{\Sigma} + k_{\Sigma} T_{33\text{т}}) p + R_{\Sigma}^2}{c R_{\Sigma} T_{\text{м}} T_{\text{тп}} T_{\Sigma} p^3 + c (R_{\Sigma} T_{\text{м}} T_{\text{тп}} + T_{\Sigma} R_{\Sigma} + k_{\Sigma} T_{\text{м}} T_{33\text{т}}) p^2 + c R_{\Sigma} (T_{\text{м}} + T_{\text{тп}}) p + R_{\Sigma} (k_{\Sigma} k_{33\text{ш}} + c)} = \\
&= \frac{(R_{\Sigma} T_{\text{тп}} T_{\Sigma} p^2 + (R_{\Sigma} T_{\text{тп}} + R_{\Sigma} T_{\Sigma} + k_{\Sigma} T_{33\text{т}}) p + R_{\Sigma}) k_{\Sigma}}{T_{\text{тп}} T_{\text{м}} T_{\Sigma} p^3 + \left(T_{\text{тп}} T_{\text{м}} + T_{\Sigma} T_{\text{м}} + \frac{k_{\Sigma} T_{\text{м}} T_{33\text{т}}}{R_{\Sigma}} \right) p^2 + (T_{\text{тп}} + T_{\text{м}}) p + 1 + k_{\Sigma} k_{\Sigma} k_{33\text{ш}}}. \quad (2.63)
\end{aligned}$$

Передавальна функція для струму по задаючій дії.

$$\begin{aligned}
W_{I_{\Sigma}, U_{\Sigma}}(p) &= \frac{I_{\Sigma}(p)}{U_{\Sigma}} = \frac{\frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + T_{33\text{т}} p k_{\Sigma} / R_{\Sigma}}}{1 + \frac{k_{\Sigma} / R_{\Sigma}}{(T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + T_{33\text{т}} p k_{\Sigma} / R_{\Sigma}} \cdot \left[\frac{c(T_{\text{тп}} p + 1)}{k_{\Sigma}} + k_{33\text{ш}} \right] \cdot \frac{R_{\Sigma}}{c T_{\text{м}} p}} = \\
&= \frac{\frac{k_{\Sigma}}{(T_{\text{тп}} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{33\text{т}} p k_{\Sigma}}}{\frac{R_{\Sigma} k_{\Sigma} c T_{\text{м}} p (T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\text{м}} p T_{33\text{т}} p + R_{\Sigma} k_{\Sigma}^2 k_{33\text{ш}} + R_{\Sigma} k_{\Sigma} c T_{\text{тп}} p + R_{\Sigma} k_{\Sigma} c}{R_{\Sigma} k_{\Sigma} c T_{\text{м}} p (T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\text{м}} p T_{\text{от}} p}} = \\
&= \frac{k_{\Sigma}}{(T_{\text{тп}} p + 1)(R_{\Sigma} p + 1) R_{\Sigma} + T_{33\text{т}} p k_{\Sigma}} \cdot \frac{R_{\Sigma} k_{\Sigma} c T_{\text{м}} p (T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma}^2 c T_{\text{м}} p T_{33\text{т}} p}{k_{\Sigma} \left[R_{\Sigma} c T_{\text{м}} p (T_{\text{тп}} p + 1)(T_{\Sigma} p + 1) + k_{\Sigma} c T_{\text{м}} p T_{33\text{т}} p + \right.} \\
&\quad \left. + R_{\Sigma} k_{\Sigma} k_{33\text{ш}} + R_{\Sigma} c T_{\text{м}} p + R_{\Sigma} c \right]} = \\
&= \frac{k_{\Sigma} T_{\text{м}} p / R_{\Sigma}}{T_{\text{тп}} T_{\text{м}} T_{\Sigma} p^3 + \left(T_{\text{тп}} T_{\text{м}} + T_{\Sigma} T_{\text{м}} + \frac{k_{\Sigma} T_{\text{м}} T_{33\text{т}}}{R_{\Sigma}} \right) p^2 + (T_{\text{тп}} + T_{\text{м}}) p + 1 + k_{\Sigma} k_{\Sigma} k_{33\text{ш}}}. \quad (2.64)
\end{aligned}$$

Запишемо передавальну функцію для струму по збурюючій дії, заздалегідь перетворивши алгоритмічну схему рис.2.11 до вигляду, показаному на рис. 2.12.

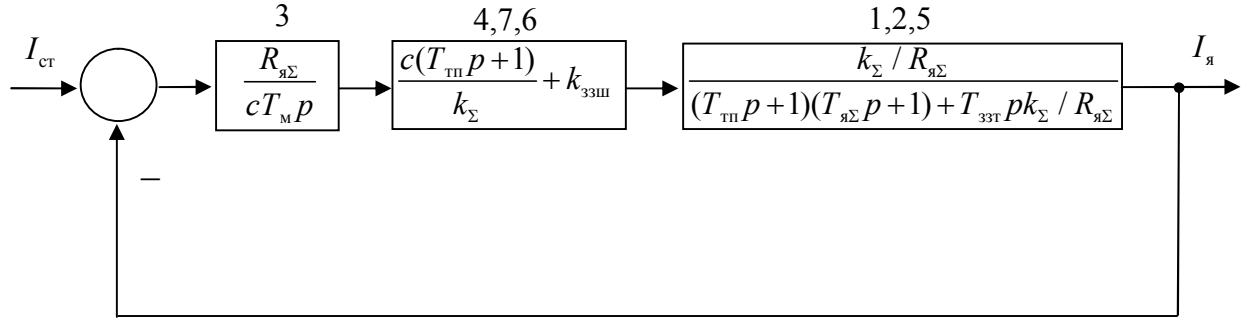


Рисунок 2.12- Перетворена алгоритмічна схема рис. 2.11 для визначення

$$\text{передавальної функції } W_4(p) = \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)}$$

$$\begin{aligned} W_{I_{\text{я}}, I_{\text{ст}}}(p) &= \frac{I_{\text{я}}(p)}{I_{\text{ст}}(p)} = \frac{\frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot \left[\frac{c(T_{\text{тп}}p + 1)}{k_{\Sigma}} + k_{33\text{ш}} \right] \cdot \frac{k_{\Sigma} / R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1) + T_{33\text{т}}pk_{\Sigma} / R_{\text{я}\Sigma}}}{1 + \frac{R_{\text{я}\Sigma}}{cT_{\text{м}}p} \cdot \left[\frac{c(T_{\text{тп}}p + 1)}{k_{\Sigma}} + k_{33\text{ш}} \right] \cdot \frac{k_{\Sigma} / R_{\text{я}\Sigma}}{(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1) + T_{33\text{т}}pk_{\Sigma} / R_{\text{я}\Sigma}}} = \\ &= \frac{R_{\text{я}\Sigma}k_{\Sigma}k_{33\text{ш}} + R_{\text{я}\Sigma}cT_{\text{тп}}p + R_{\text{я}\Sigma}c}{cT_{\text{м}}p(T_{\text{тп}}p + 1)(T_{\text{я}\Sigma}p + 1)R_{\text{я}\Sigma} + cT_{\text{м}}T_{33\text{т}}p^2 \cdot k_{\Sigma} + R_{\text{я}\Sigma}k_{\Sigma}k_{33\text{ш}} + R_{\text{я}\Sigma}cT_{\text{тп}}p + R_{\text{я}\Sigma}c} = \\ &= \frac{R_{\text{я}\Sigma}(k_{\Sigma}k_{33\text{ш}} + cT_{\text{тп}}p + c)}{cT_{\text{м}}p(T_{\text{тп}}T_{\text{я}\Sigma}p^2 + T_{\text{я}\Sigma}p + T_{\text{тп}}p + 1)R_{\text{я}\Sigma} + cT_{\text{м}}T_{33\text{т}}p^2 \cdot k_{\Sigma} + R_{\text{я}\Sigma}cT_{\text{тп}}p + R_{\text{я}\Sigma}(k_{\Sigma}k_{33\text{ш}} + c)} = \\ &= \frac{R_{\text{я}\Sigma}(k_{\Sigma}k_{33\text{ш}} + cT_{\text{тп}}p + c)}{R_{\text{я}\Sigma}cT_{\text{м}}T_{\text{тп}}T_{\text{я}\Sigma}p^3 + c(R_{\text{я}\Sigma}T_{\text{м}}T_{\text{я}\Sigma} + R_{\text{я}\Sigma}T_{\text{м}}T_{\text{тп}} + k_{\Sigma}T_{\text{м}}T_{33\text{т}})p^2 + R_{\text{я}\Sigma}c(T_{\text{м}} + T_{\text{тп}})p + R_{\text{я}\Sigma}(k_{\Sigma}k_{33\text{ш}} + c)} = \\ &= \frac{T_{\text{тп}}p + 1 + k_{\Sigma}k_{33\text{ш}}k_{\text{д}}}{T_{\text{тп}}T_{\text{м}}T_{\text{я}\Sigma}p^3 + \left(T_{\text{тп}}T_{\text{м}} + T_{\text{я}\Sigma}T_{\text{м}} + \frac{k_{\Sigma}T_{\text{м}}T_{33\text{т}}}{R_{\text{я}\Sigma}} \right) p^2 + (T_{\text{тп}} + T_{\text{м}})p + 1 + k_{\Sigma}k_{\text{д}}k_{33\text{ш}}} \cdot (2.65) \end{aligned}$$

Графіки перехідних процесів, побудовані по отриманих передавальних функціях, показані на рис.2.14 і 2.15. Графіки отримані за допомогою ЕОМ з використанням пакету прикладних програм MATLAB. Криві 1 відповідають $k_{\text{зпн}} = 0,26$, криві 2 - $k_{\text{зпн}} = 0,52$, криві 3 - $k_{\text{зпн}} = 1,56$. Перехідні процеси, що мають найкращі показники якості, отримані при $k_{\text{зпн}} = 0,04$. Побудова перехідних процесів була виконана також за допомогою схеми моделі, показаної на рис.2.13. Графіки, отримані різними методами, повністю співпадають, що свідчить про правильність розробленої математичної моделі системи.

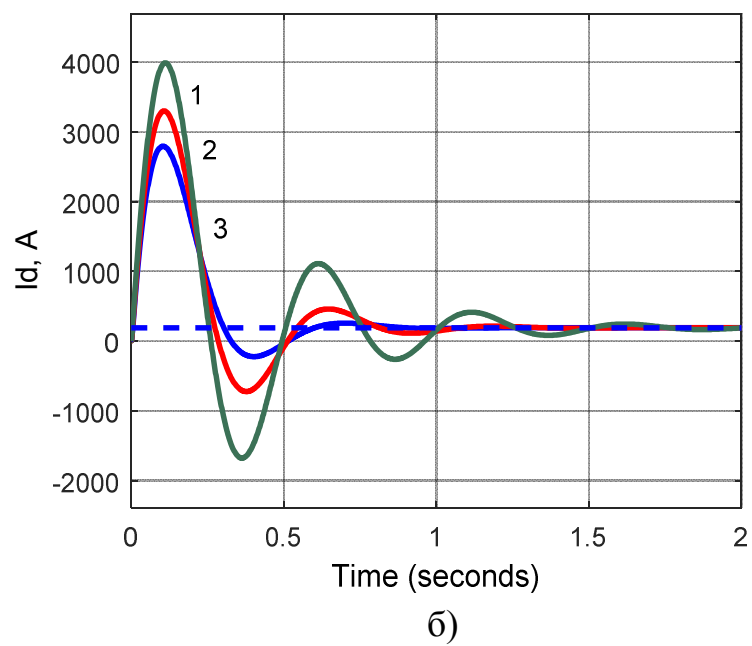
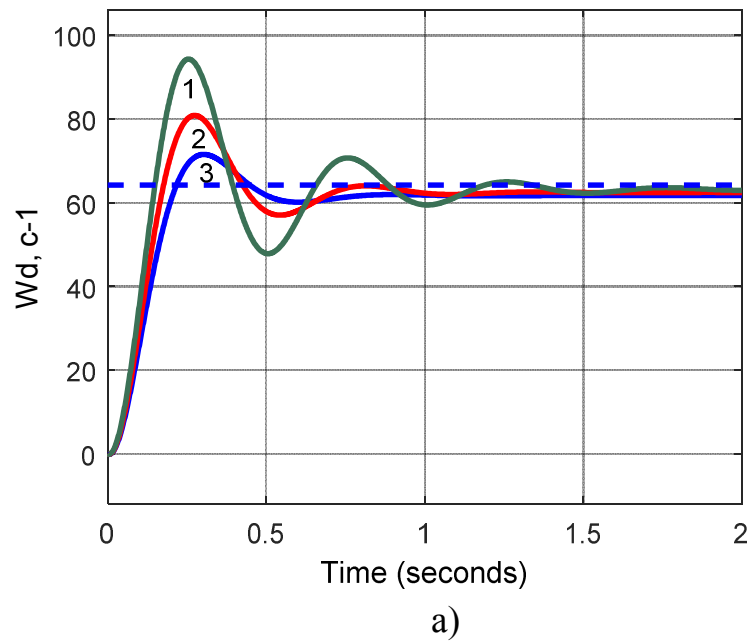


Рисунок 2.14 – Графіки перехідних процесів швидкості а)
і струму б) по задаючій дії:

$$1 - k_{33H} = 0,71, \quad 2 - k_{33H} = 1,136; \quad 3 - k_{33H} = 2,13$$

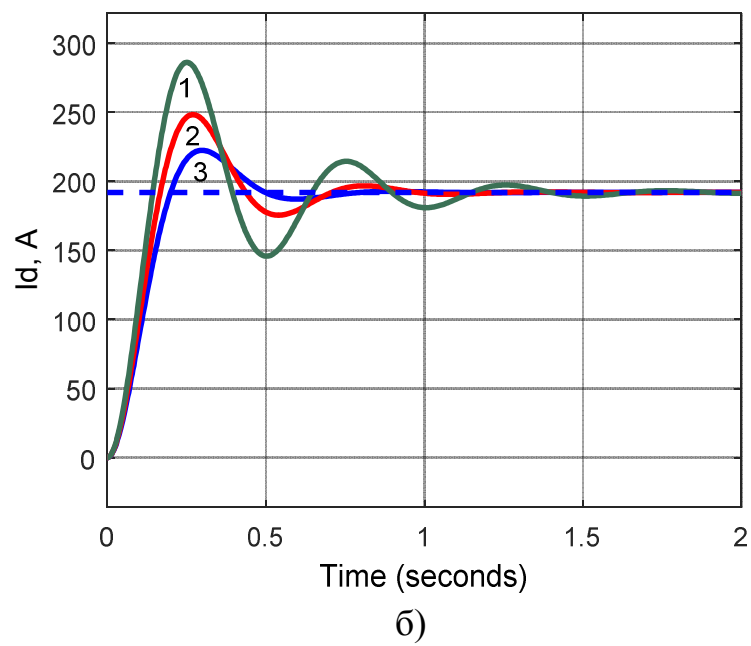
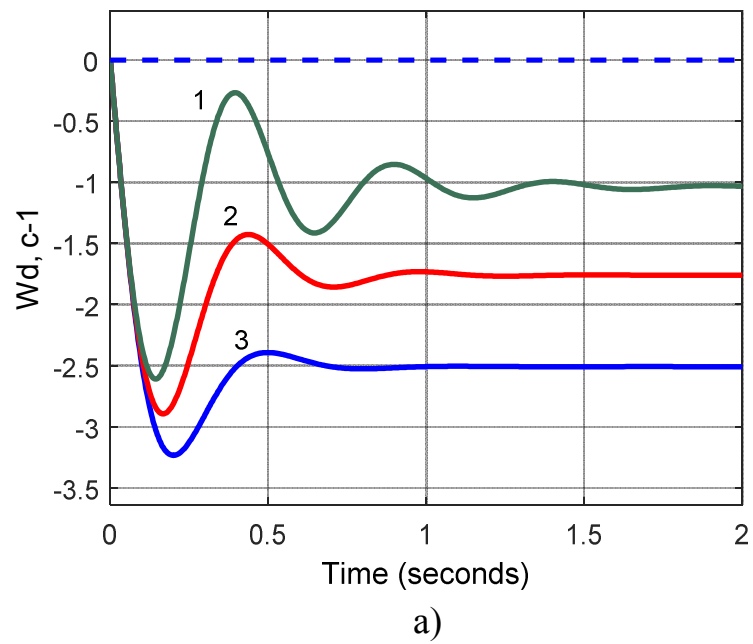


Рисунок 2.15 – Графіки перехідних процесів швидкості а)
і струму б) по збурюючій дії:

$$1 - k_{33H} = 0,71, \quad 2 - k_{33H} = 1,136; \quad 3 - k_{33H} = 2,13$$

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 [59] 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

ХАРАКТЕРИСТИКИ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

Для електроприводу механізму розвороту екскаватора характерно 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 попередньому розділі проводилося дослідження динамічних характеристик системи електроприводу механізму розвороту за умови нескінченної жорсткості усіх зв'язків між елементами механічної частини. Представлення механічної частини як жорсткої приведеної ланки відображає лише середній рух елементів, оскільки не враховує вплив пружності реальних механічних зв'язків. Оскільки ці зв'язки мають обмежену жорсткість, механічна частина електроприводу стає пружною системою, на яку навантаження у вигляді управляючого моменту (від двигуна) або збурюючого моменту (від статичного навантаження) призводить до коливань зв'язаних мас, що збільшує максимальне навантаження зв'язків і ускладнює виконання необхідних рухів і переміщень, що впливає на точність.

Кінематична схема механізму обертання стріли представляє собою складну механічну систему з великою кількістю зосереджених і розподілених мас, які з'єднані пружними зв'язками. При проектуванні таких систем інженери зазвичай спрощують їх до трьох- або двомасових моделей з централізованими масами і гнучкими невагомими зв'язками між ними.

Для електроприводу механізму розвороту 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 жорсткості кожно-

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

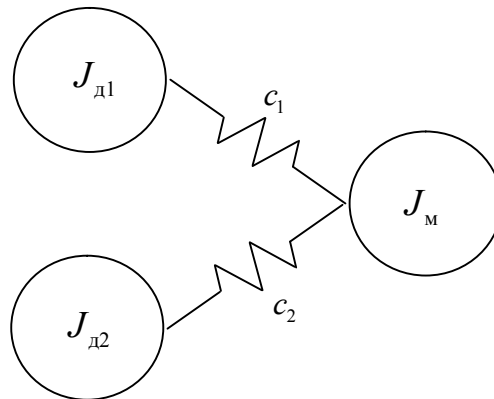


Рисунок 3.1 – Розгалужена розрахункова схема механічної частини електроприводу

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; $c_{12} = c_1 + c_2$.

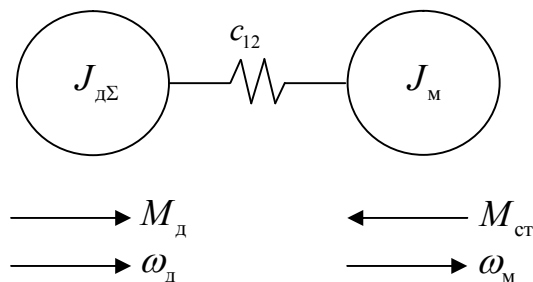


Рисунок 3.2 – Двомасова розрахункова схема

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

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

де M_{Σ} сумарний момент, 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

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

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

$$\left\{ \begin{array}{l} T_{\text{м}} \frac{d\omega_{\text{д}}}{dt} = \frac{R_{\text{я}\Sigma}}{c} (I_{\text{я}} - I_{\text{ст}}); \\ T_{\text{я}\Sigma} \frac{dI_{\text{я}}}{dt} + I_{\text{я}} = \frac{1}{R_{\text{я}\Sigma}} (E_{\text{тп}} - c\omega_{\text{д}}); \\ T_{\text{тп}} \frac{dE_{\text{тп}}}{dt} + E_{\text{тп}} = k_{\Sigma} U_3 - k_{33\text{ш}} k_{\Sigma} \omega_{\text{д}} - k_{\Sigma} U_{\Gamma 33\text{т}}; \\ T_{33\text{т}} \frac{dI_{\text{я}}}{dt} = U_{\Gamma 33\text{т}}. \end{array} \right. \quad (3.4)$$

Запишемо систему (3.4) у формі Коші, враховуючи, що $I_{\text{ст}} = M_{\text{ст}} / c$, $T_{\text{і}} = J_{\text{а}\Sigma} \frac{R_{\text{я}\Sigma}}{(k\hat{O}_{\text{і}})^2} = J_{\text{а}\Sigma} \frac{R_{\text{я}\Sigma}}{c^2}$. Об'єднаємо системи (3.3) і (3.4) і проведемо деякі нескладні перетворення. Отримаємо наступну систему рівнянь стану двомасової системи:

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

Введемо вектор стану системи

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

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:

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

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:

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

Алгоритмічна схема двомасової системи показана на рис.3.3.

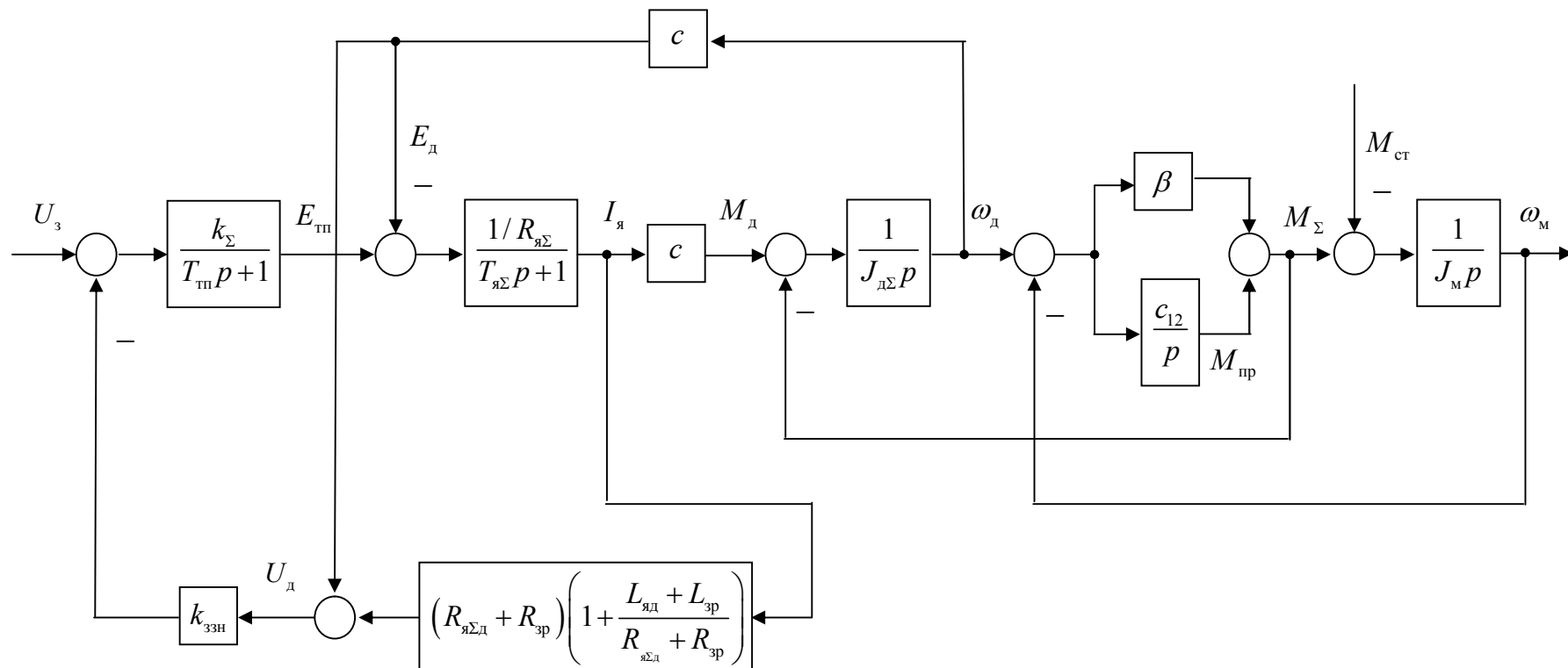


Рисунок 3.3 - Алгоритмічна схема двомасової системи.

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 Д. 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 рис. 3.5 - рис. 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.

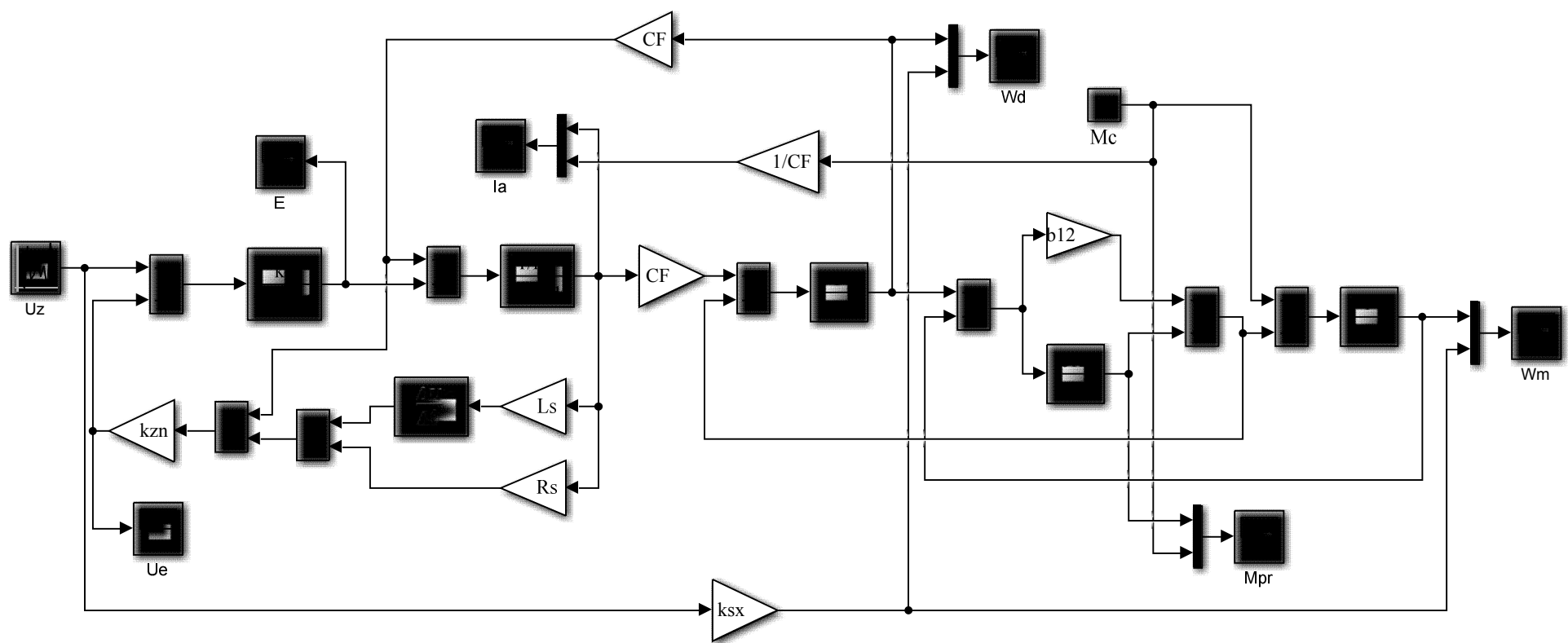


Рисунок 3.4 - Схема моделі двомасової системи, розроблена на ЕОМ

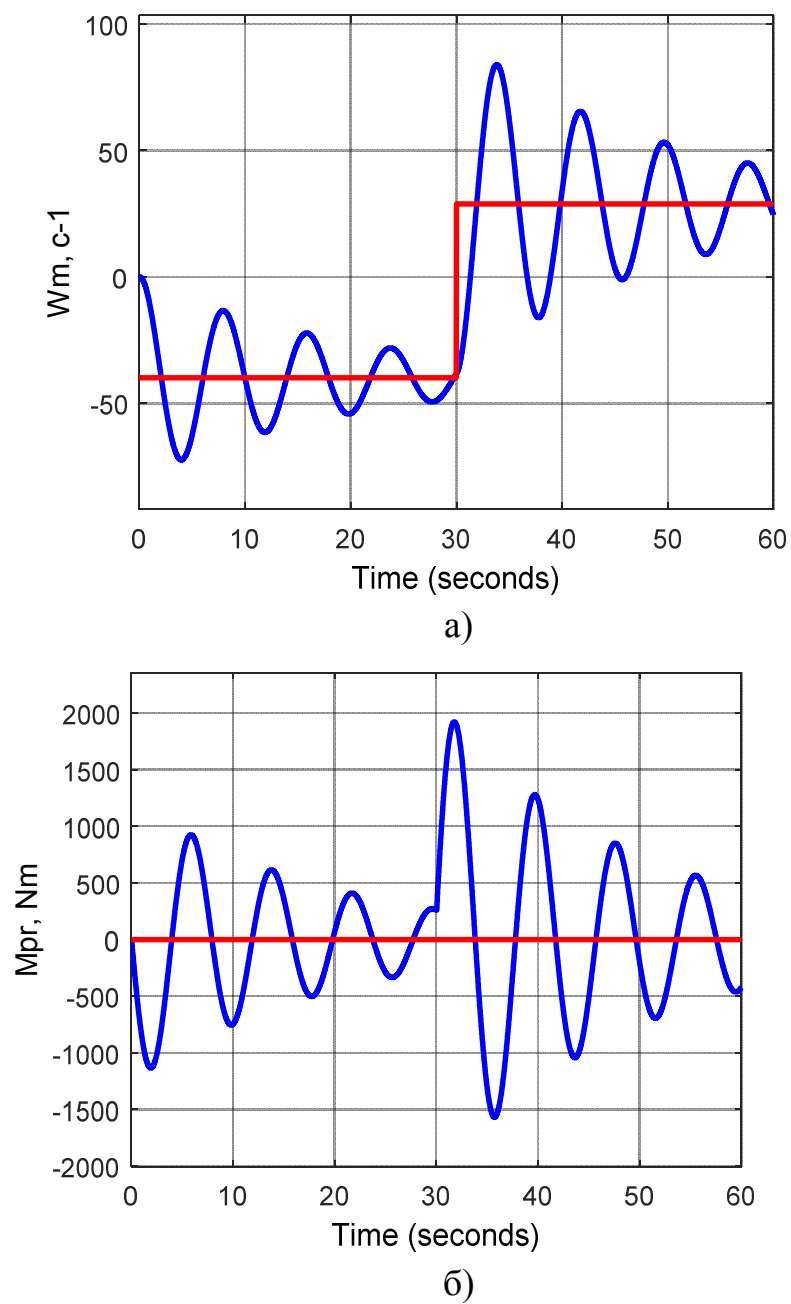
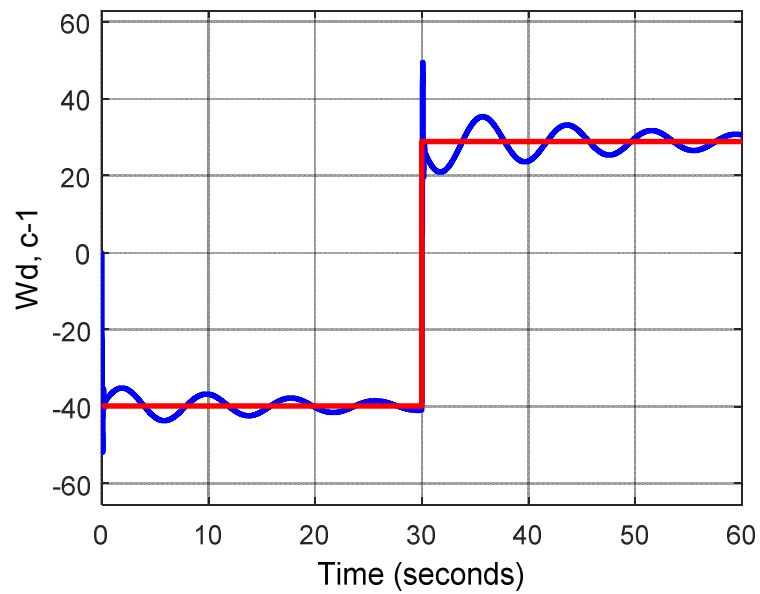
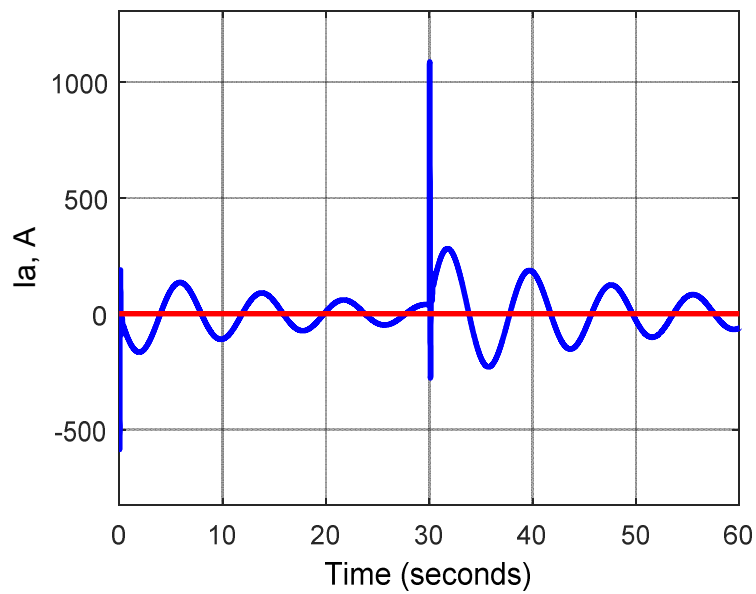


Рисунок 3.5 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

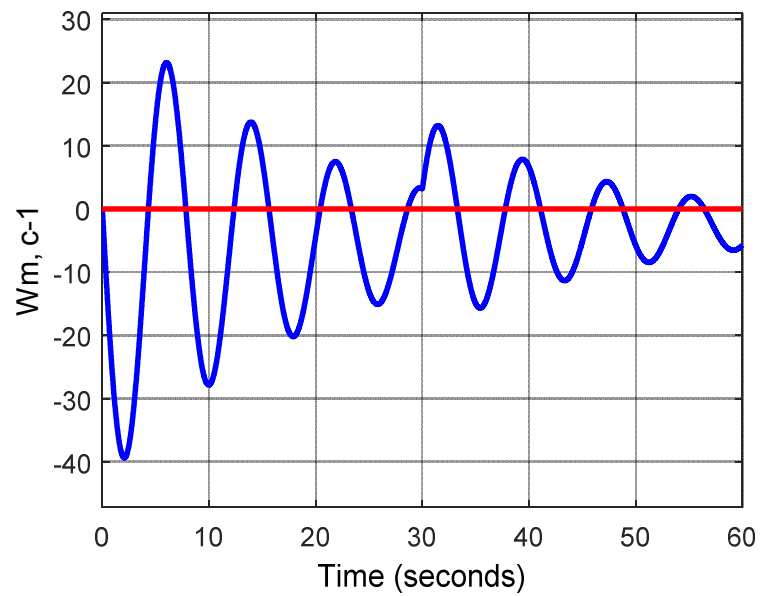


a)

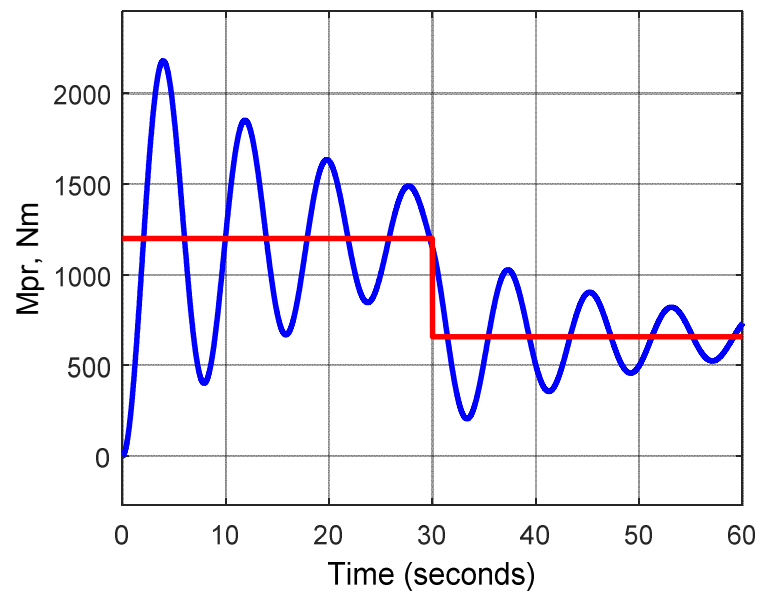


б)

Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму якірного кола двигуна $I_a = f(t)$ (б) двомасової системи по задаючій дії

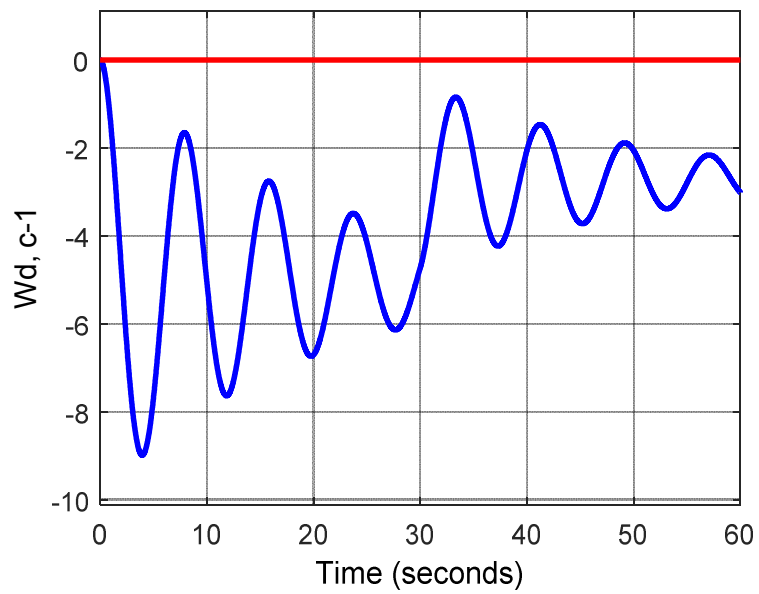


a)

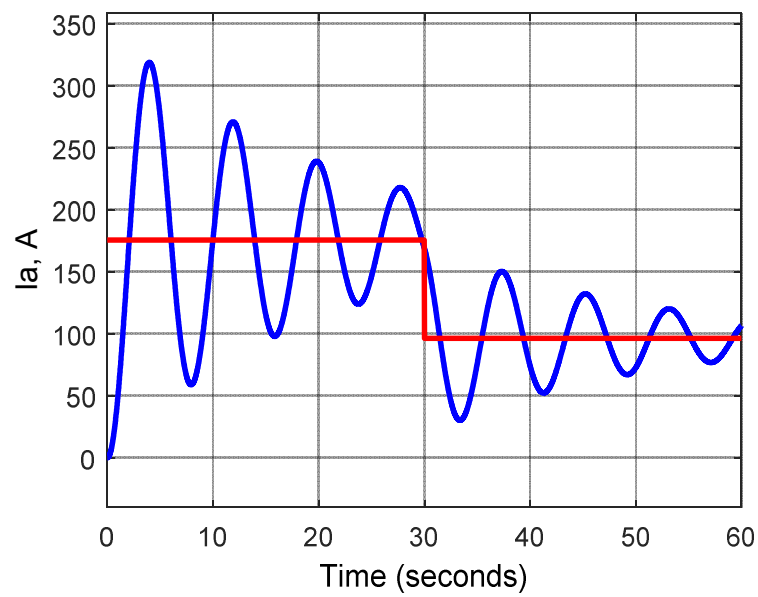


б)

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 якорного кола двигуна $I_a = f(t)$ (б) двомасової системи
по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму розвороту 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 механізму розвороту 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.

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

Синтез системи управління проводився за допомогою **Neural Network Toolbox** в системі MATLAB. Далі 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'єкта 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'єкта 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'єкта.

При управлінні на основі еталонної моделі, регулятор - це навчена нейронна мережа, яка управляє об'єктом так, щоб він відтворював поведінку ета-

лонної 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).

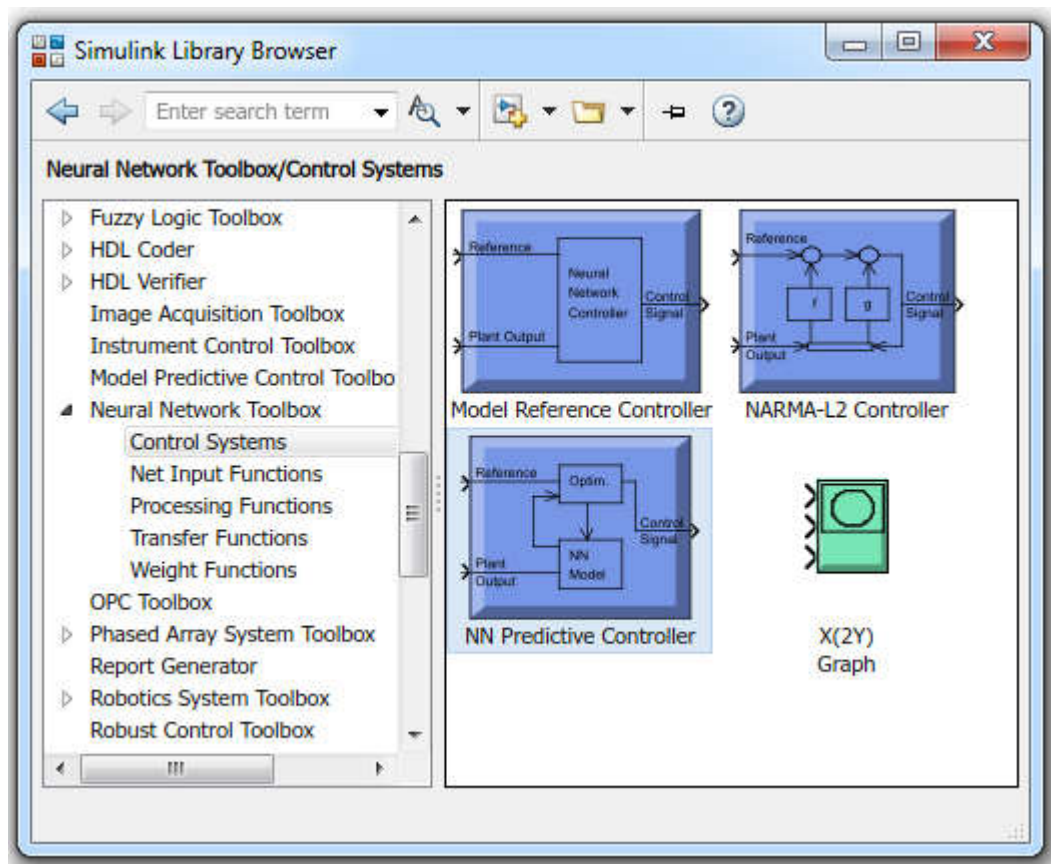


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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 в груповому режимі за одним з алгоритмів навчання. Цей тип регулятора по-

требує 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.

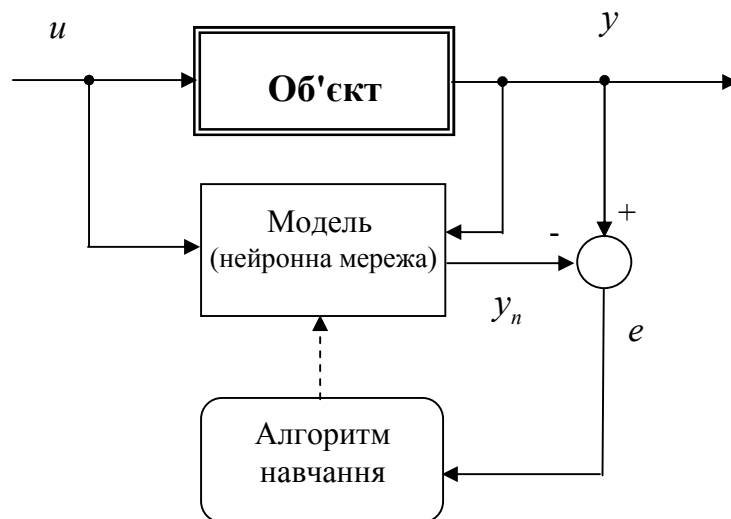


Рисунок 4.2 – Схема підсистеми ідентифікації.

Нейронна мережа, що представлена на рисунку 4.3 і використовується у регуляторі об'єкту управління, має два шари нейронів та використовує лінії за-тримки (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**.

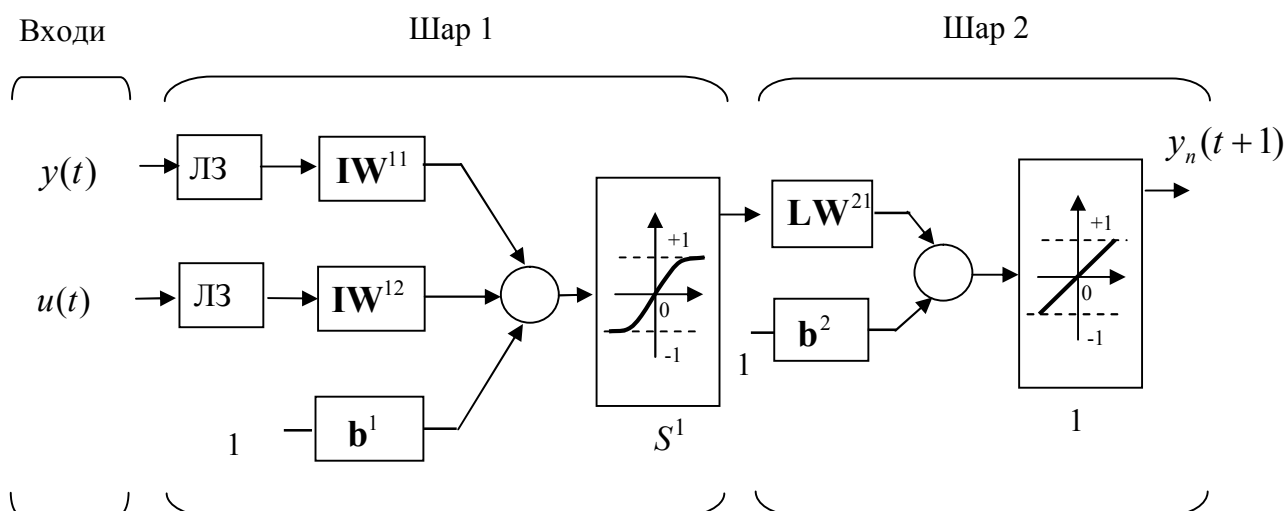


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

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.

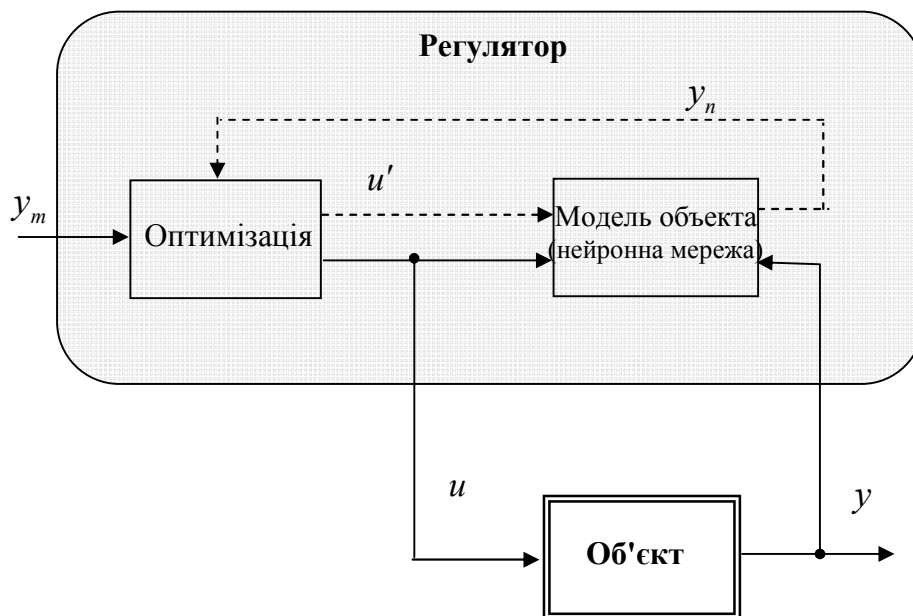


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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.

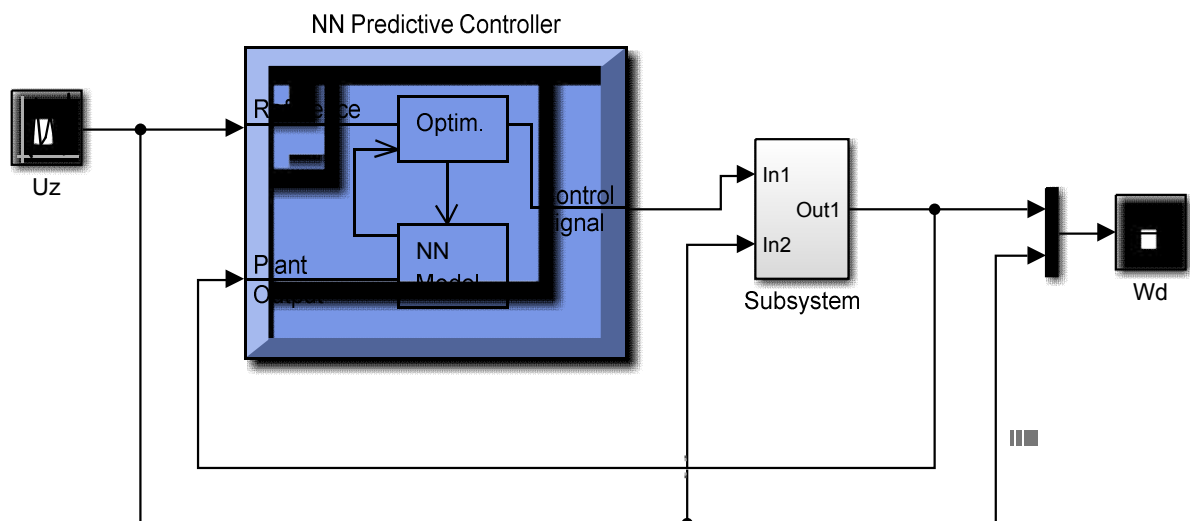


Рисунок 4.5 – Схема системи управління з нейрорегулятором

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-

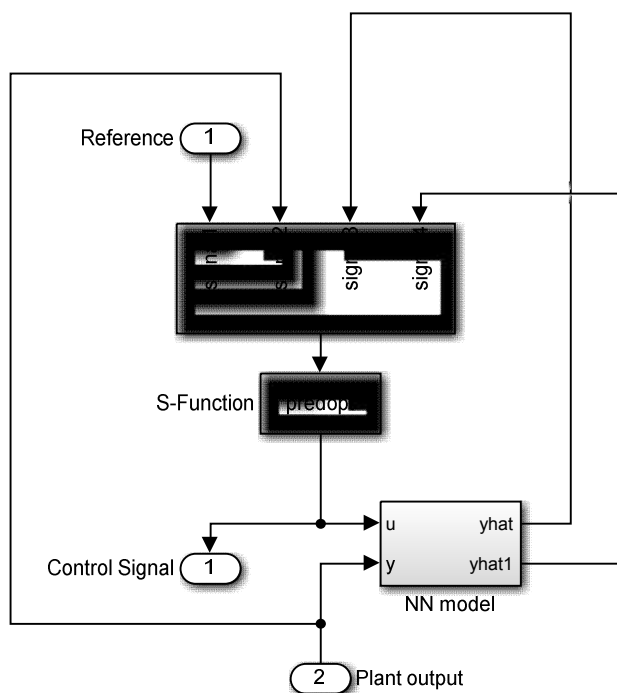


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

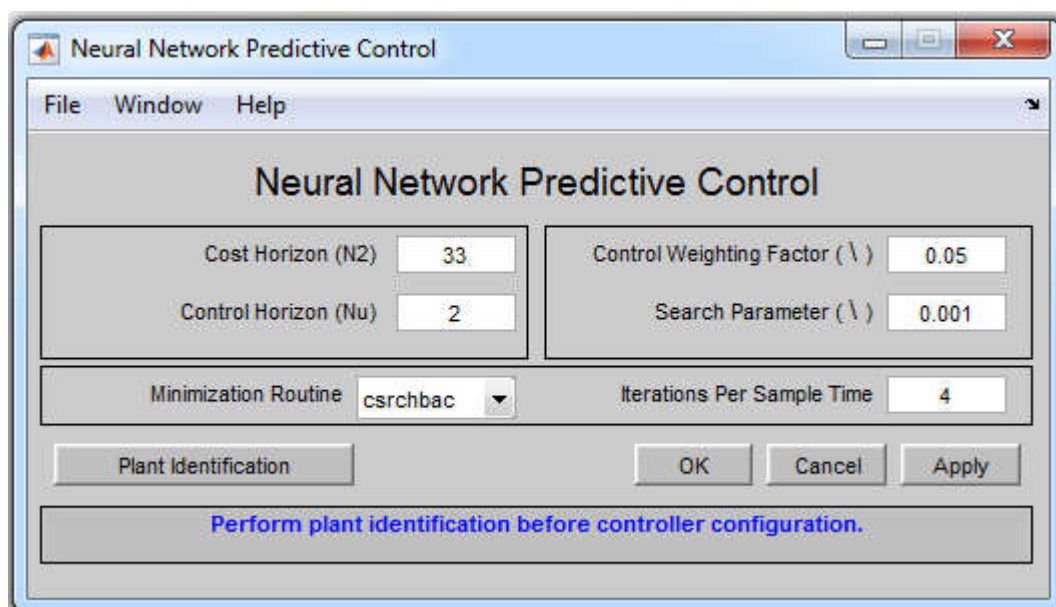


Рисунок 4.7 – Вікно завдання параметрів регулятора

Вікно **Plant Identification** зображено на рисунку 4.8. Ця структура є універсальною і може бути використана для розробки нейромережових 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.

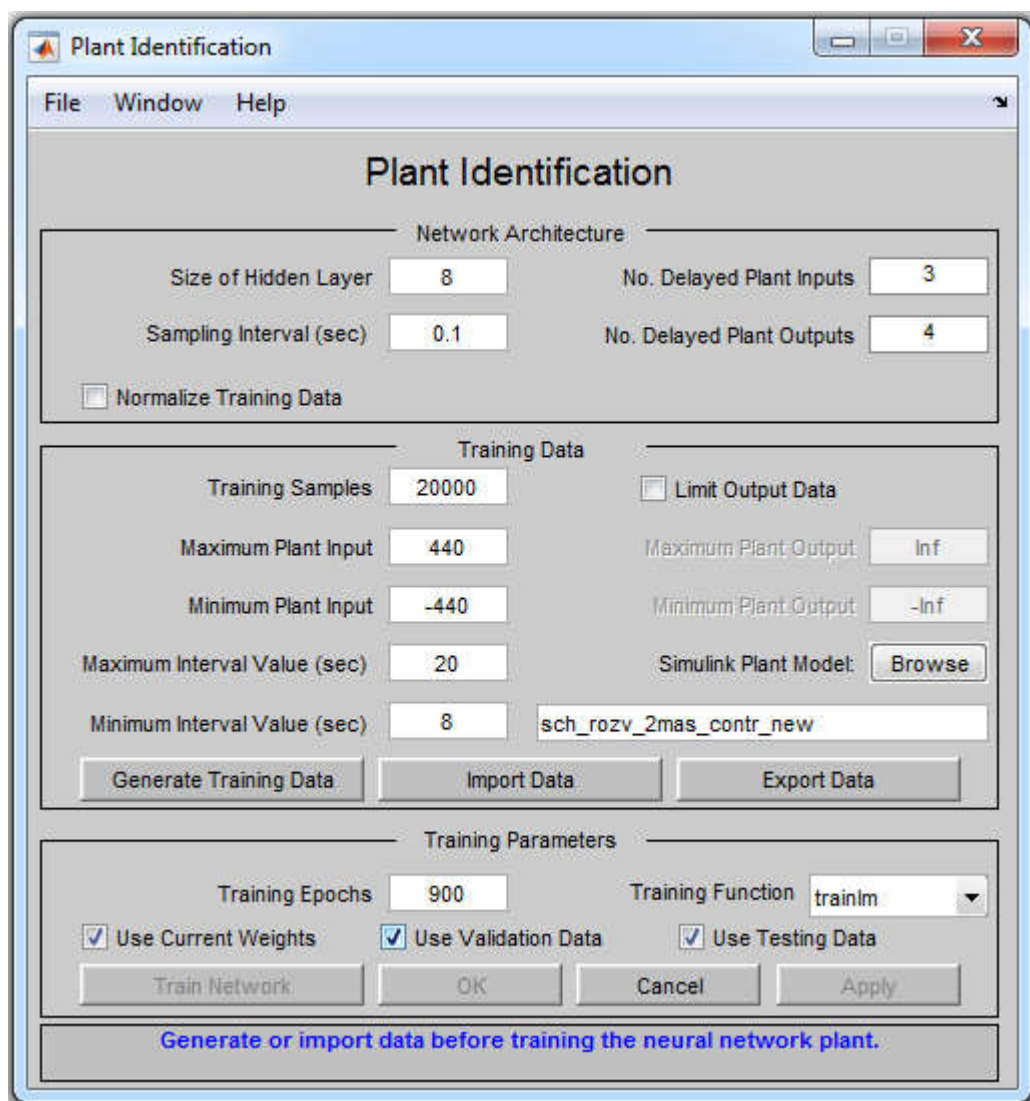


Рисунок 4.8 – Вікно ідентифікації керованої системи

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.

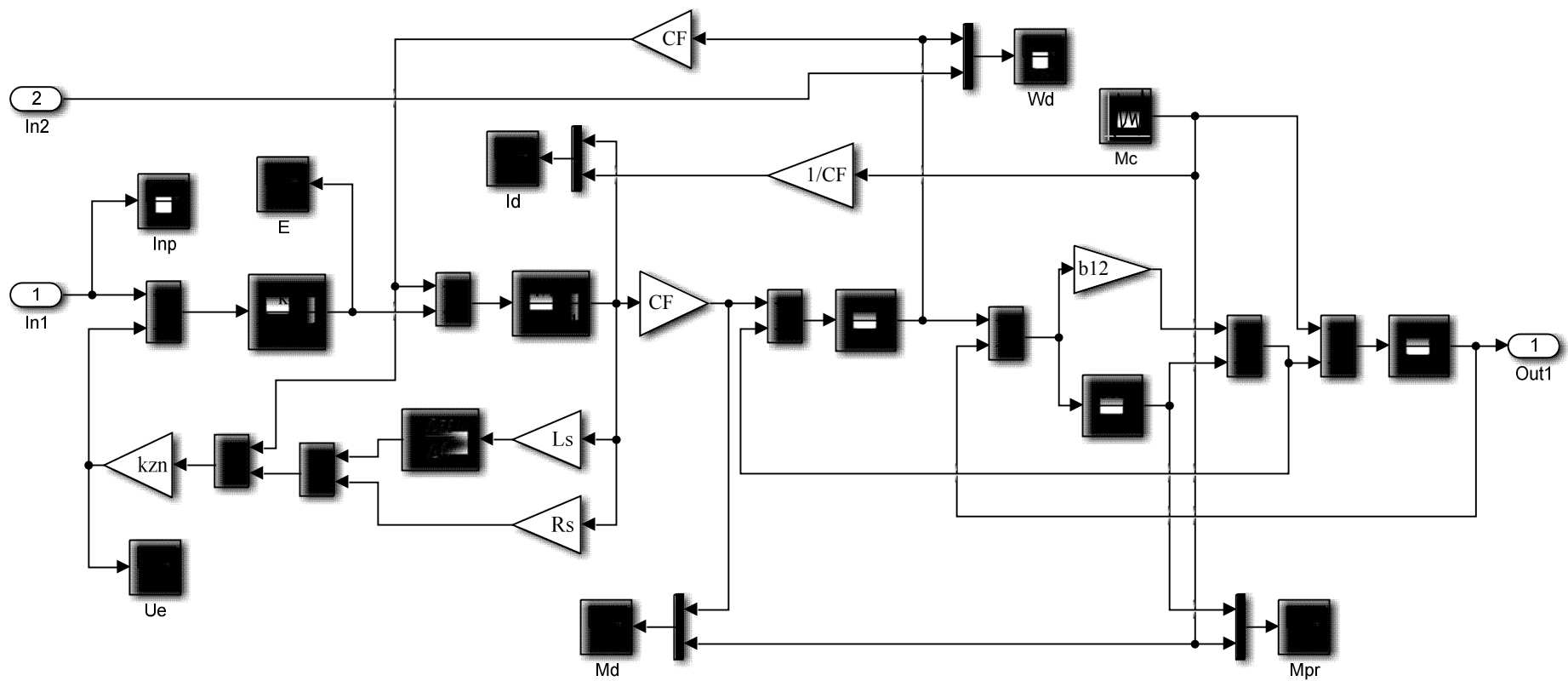


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

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.

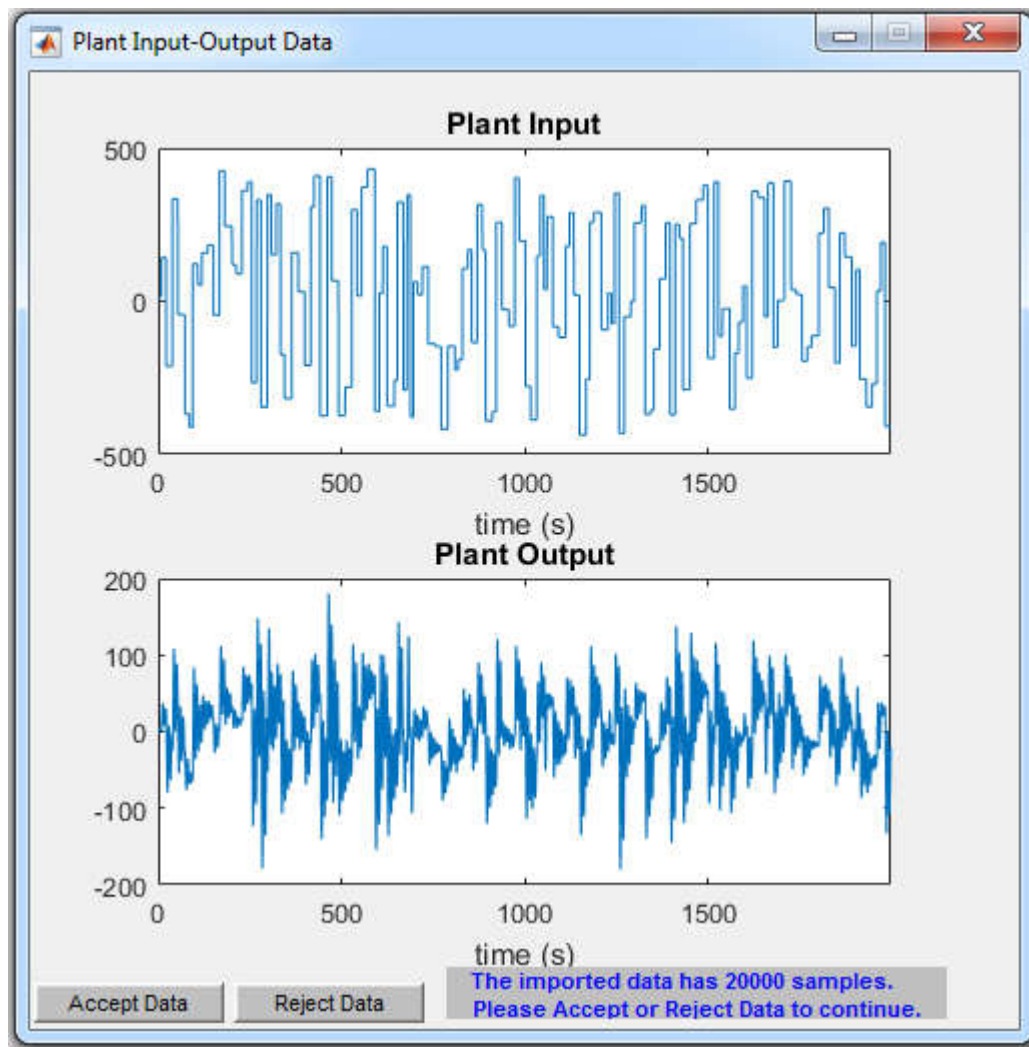


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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)

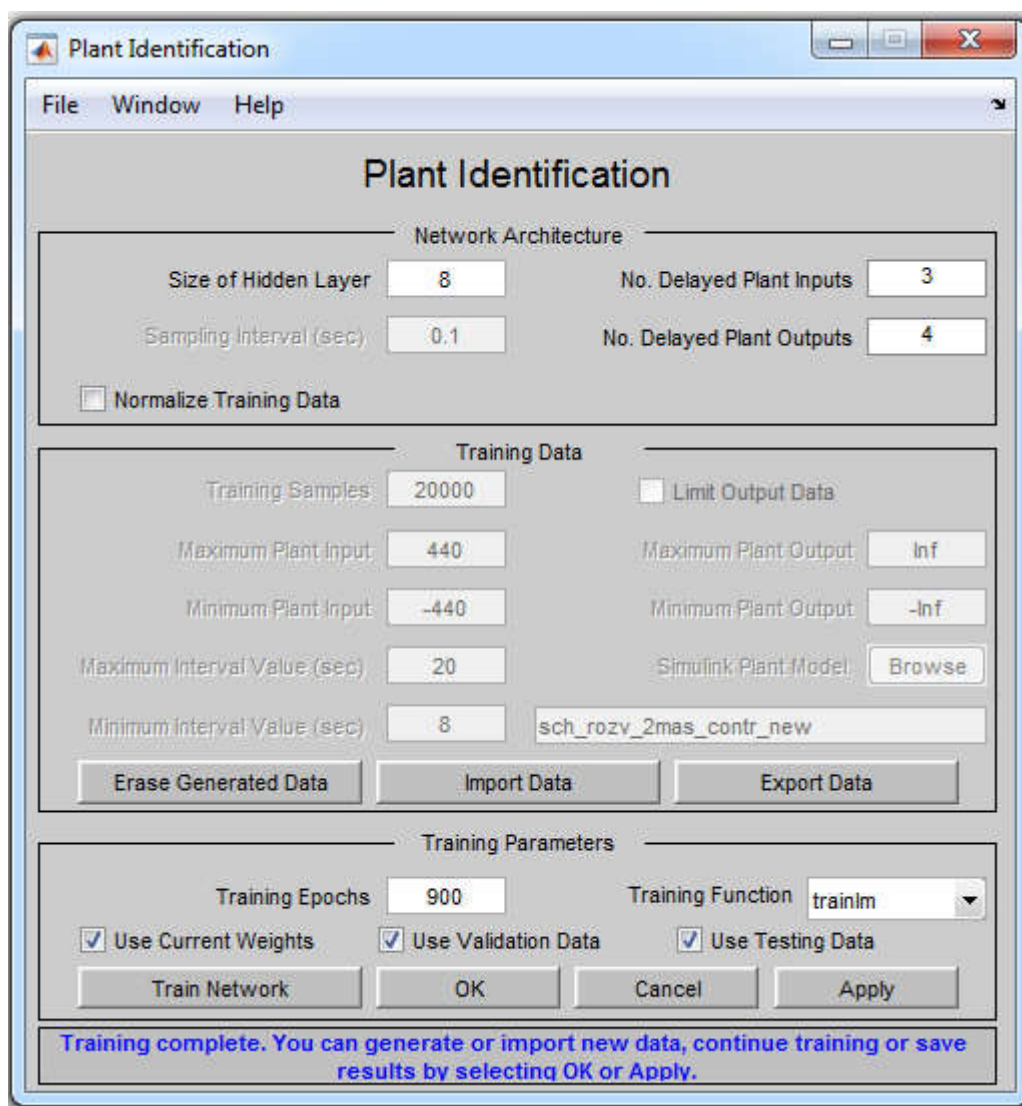


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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.

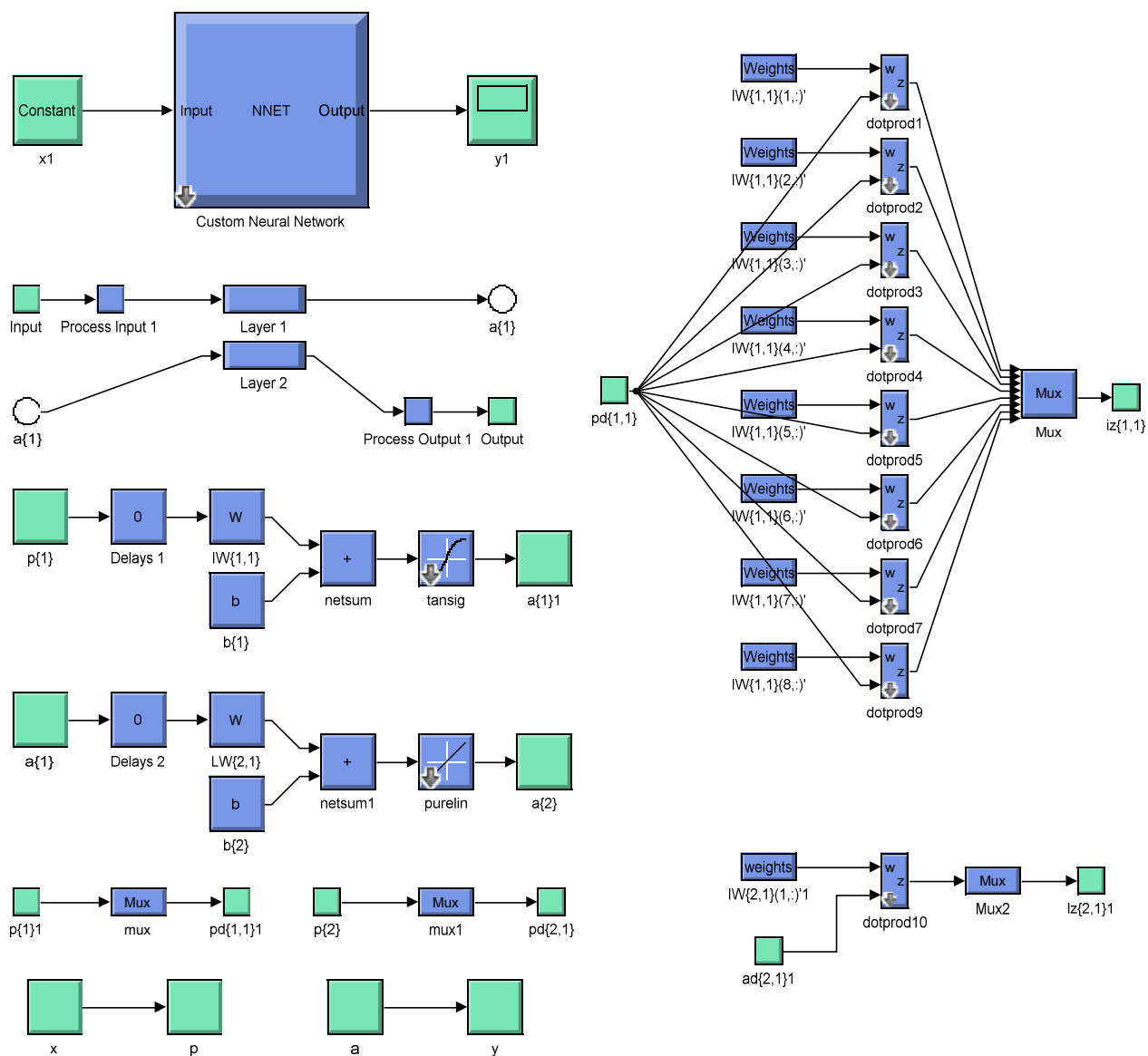


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

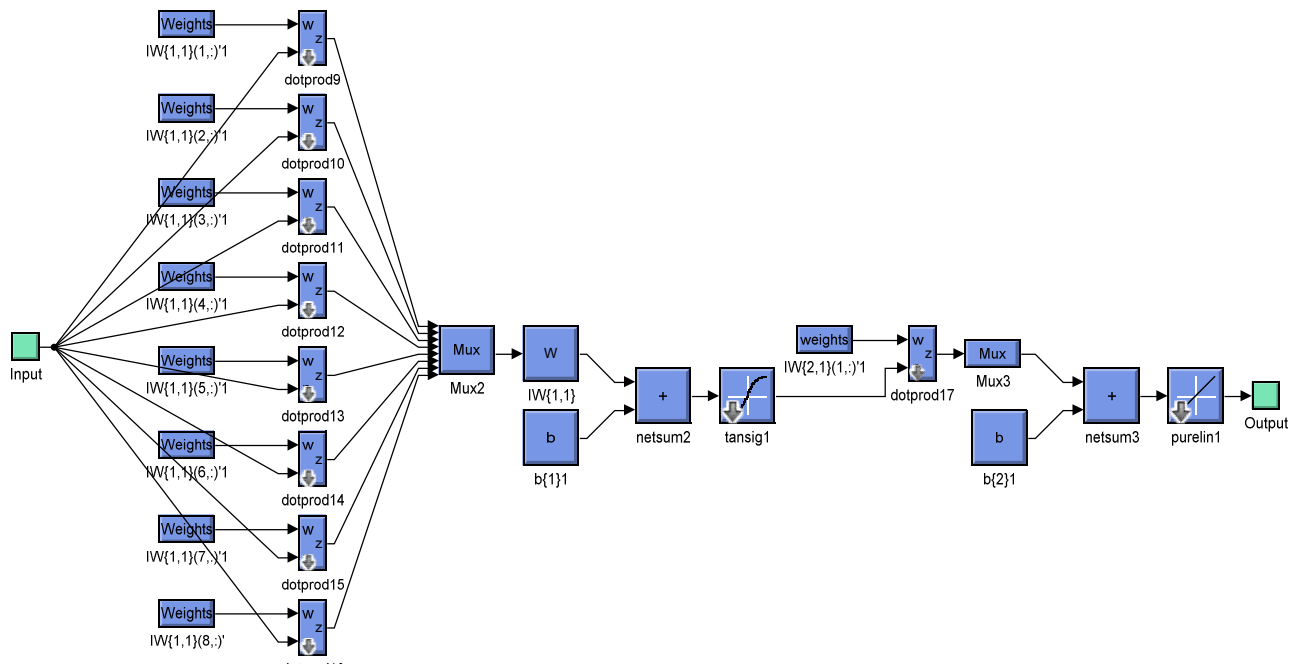


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

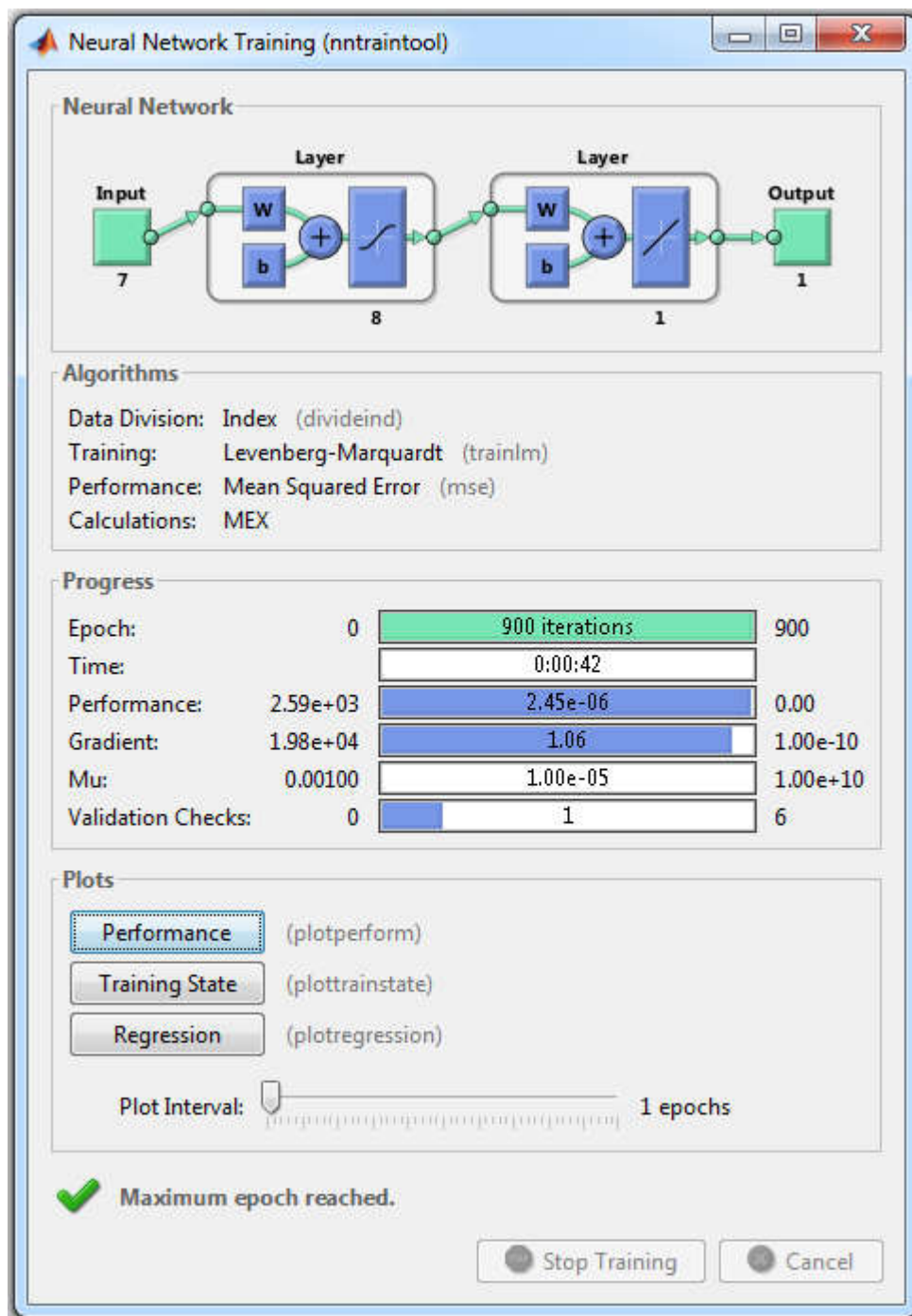


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

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 (Mu) 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 Mu 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 перестає 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.

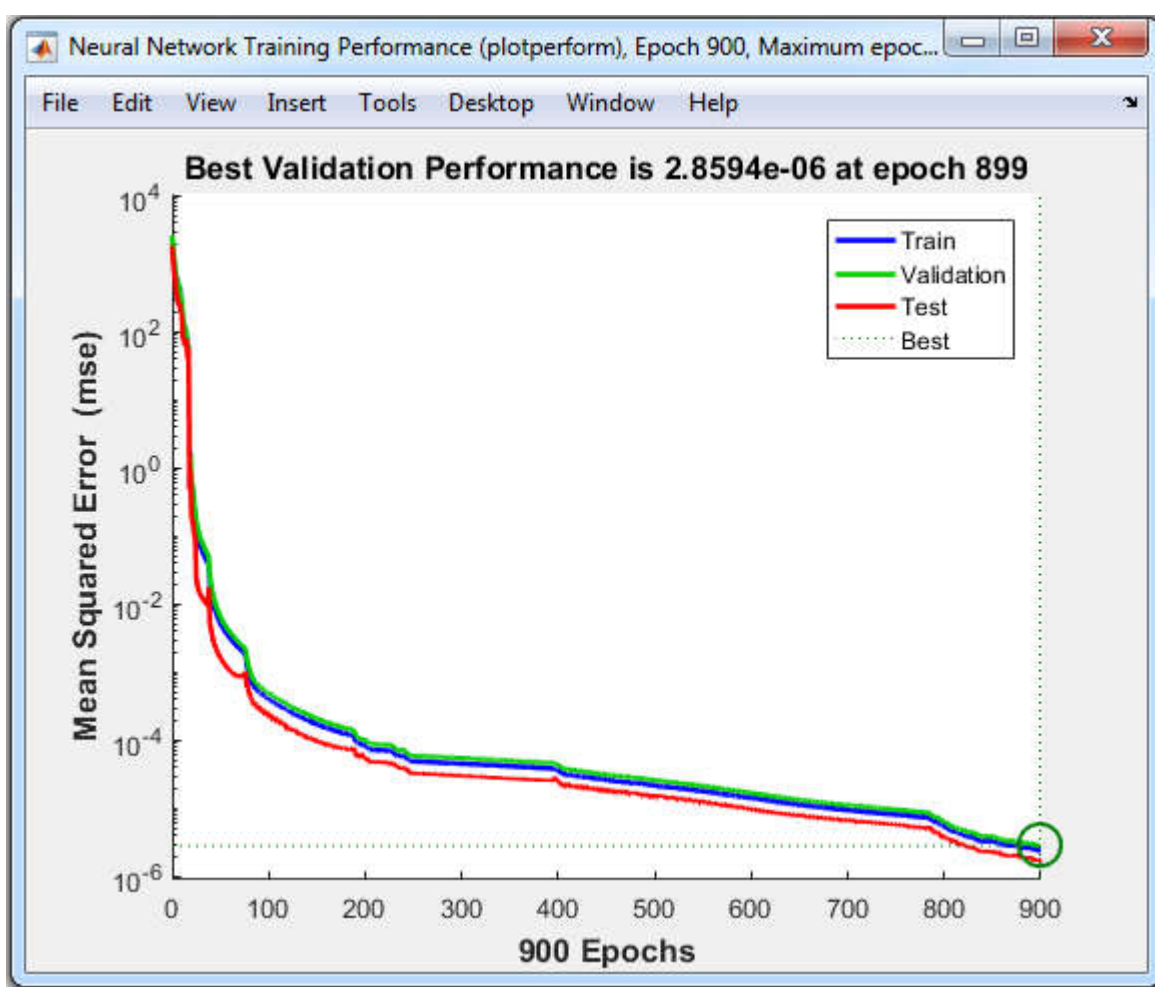


Рисунок 4.15 – Вікно контролю процесу навчання

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 інформація про стан навчання нейронної мережі на певний момент часу під час процесу навчання.

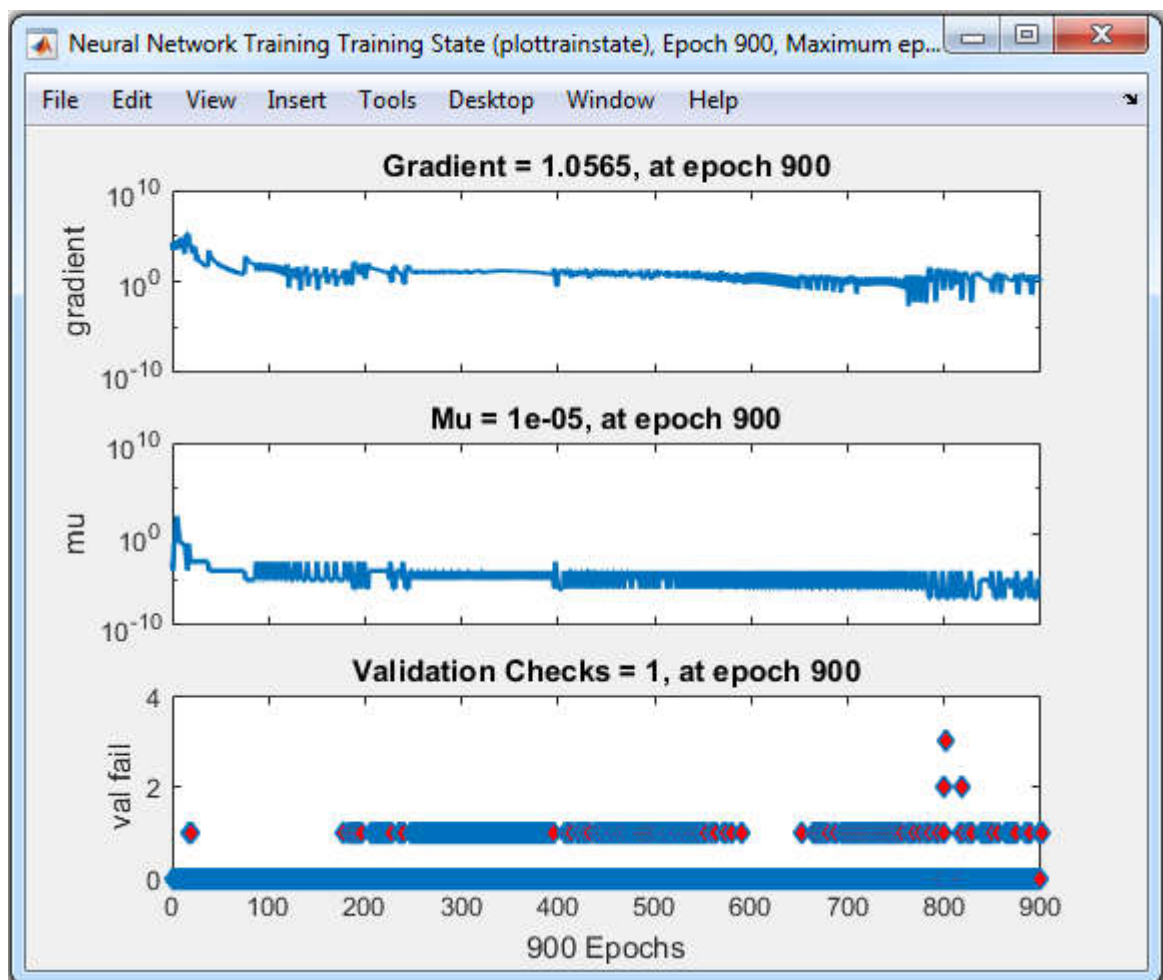


Рисунок 4.16 - Вікно Neural Network Training State

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 вказувати 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 вхідних даних. У регресійних завданнях модель намагається знайти функціональну залежність між вхідними 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.

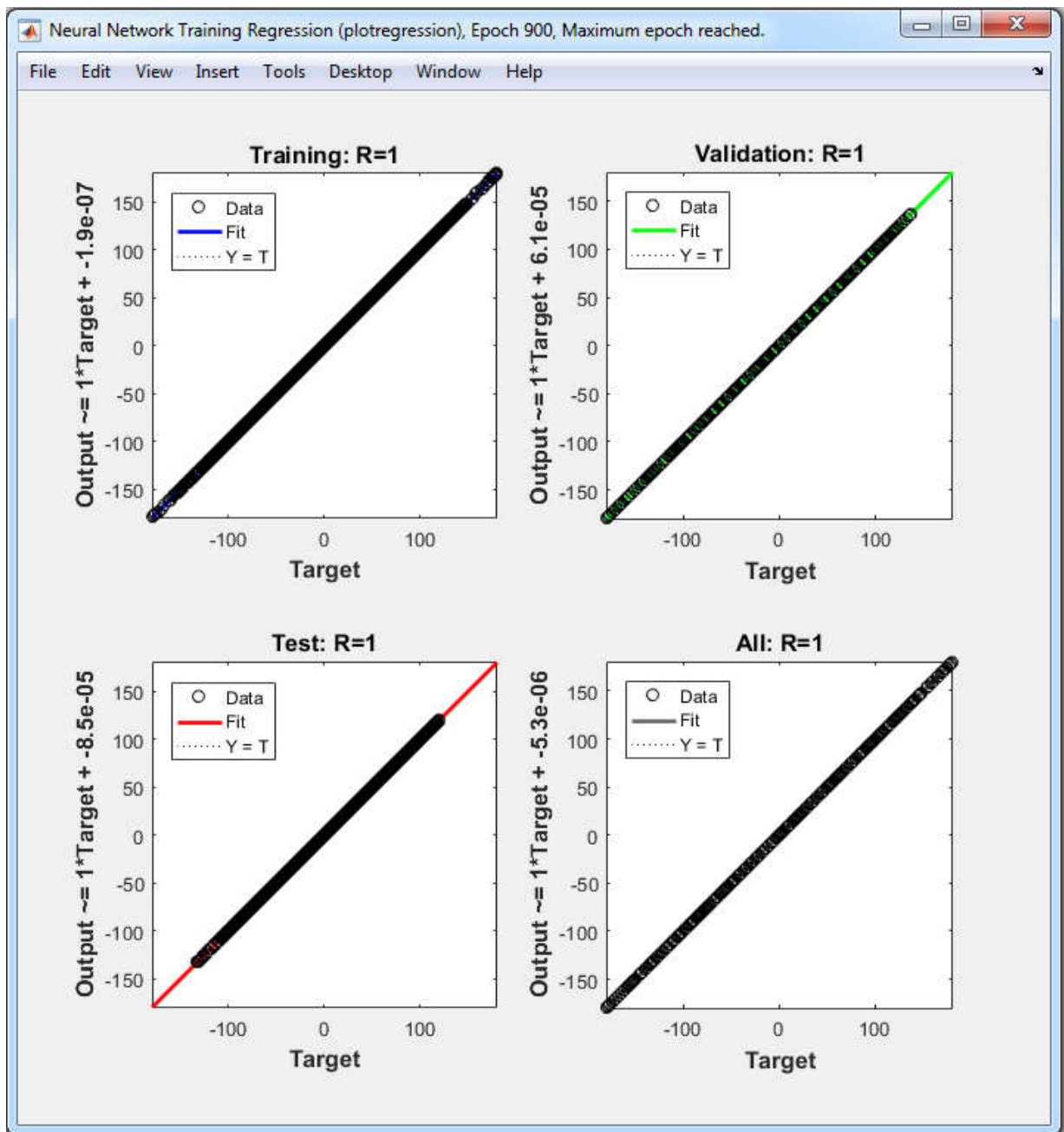


Рисунок 4.17 - Вікно Neural Network Training Regression

Onuc вікна Neural 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. Гра-

фіки 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 точ-

ність. 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.

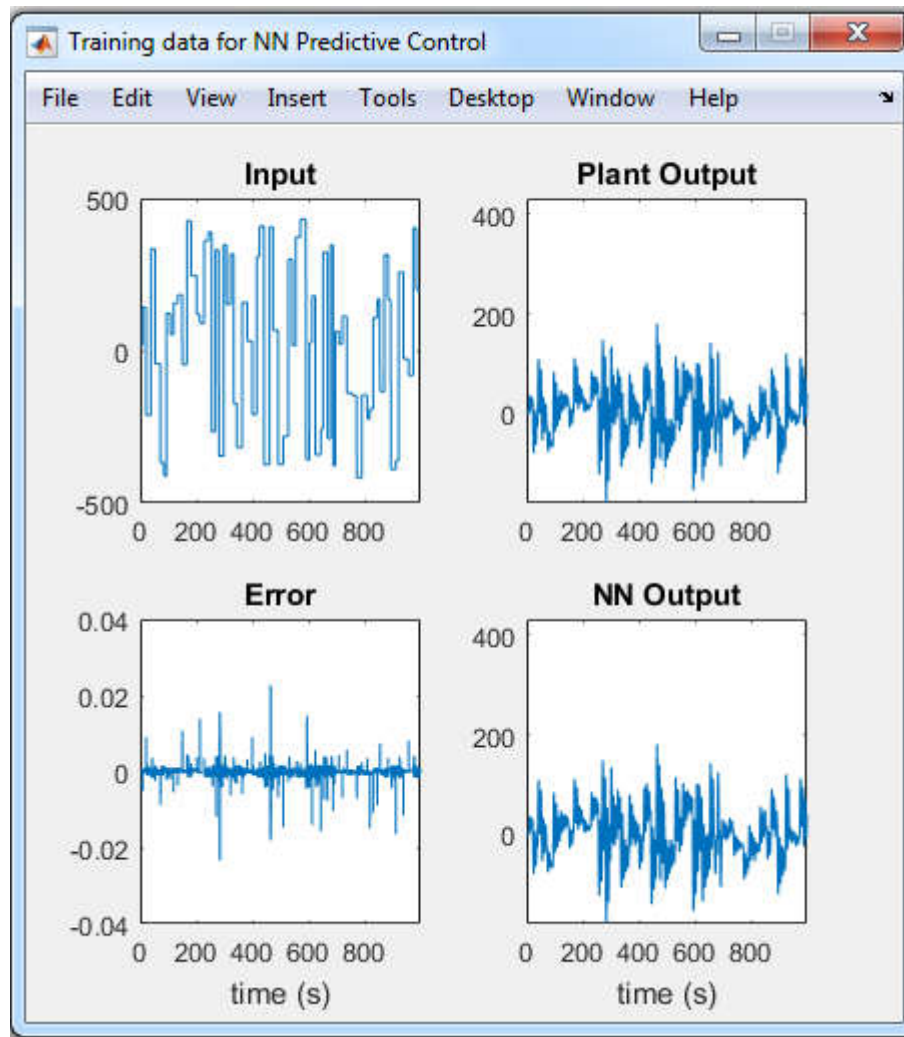


Рисунок 4.18 – Результати тренування мережі

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.

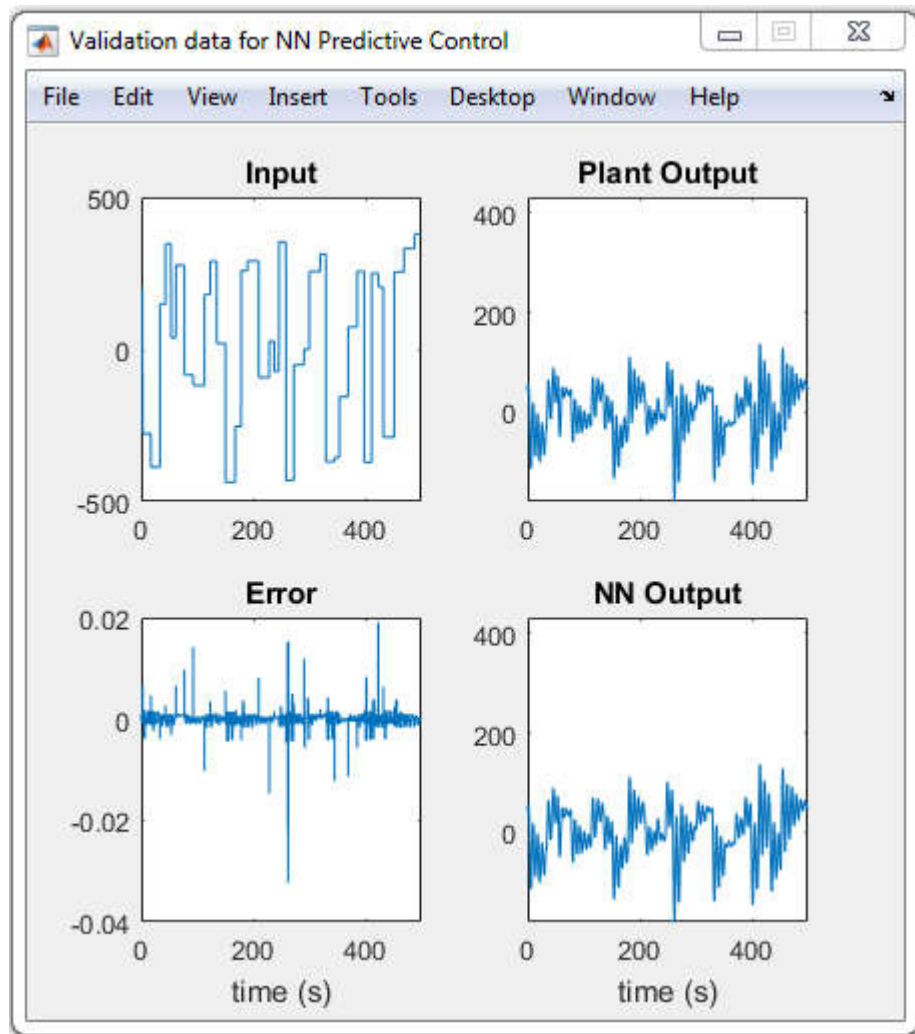


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

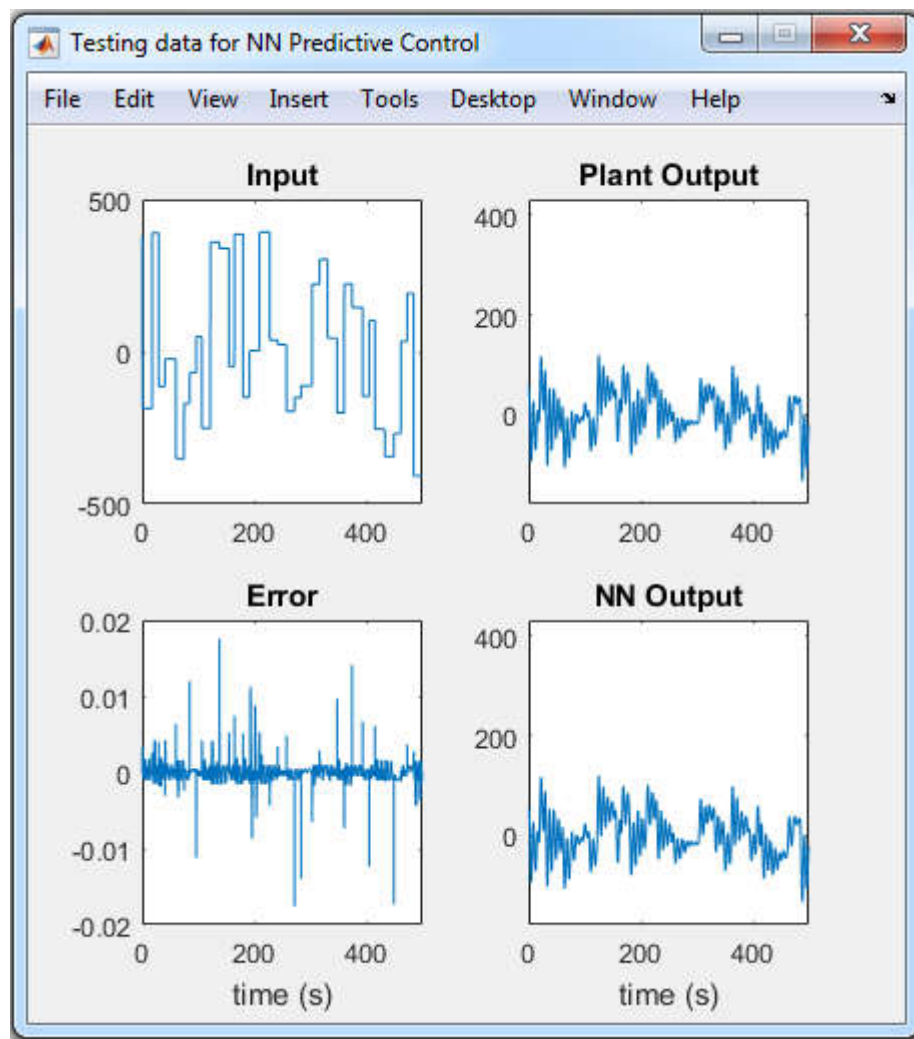


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

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*).

Після завершення навчання 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.

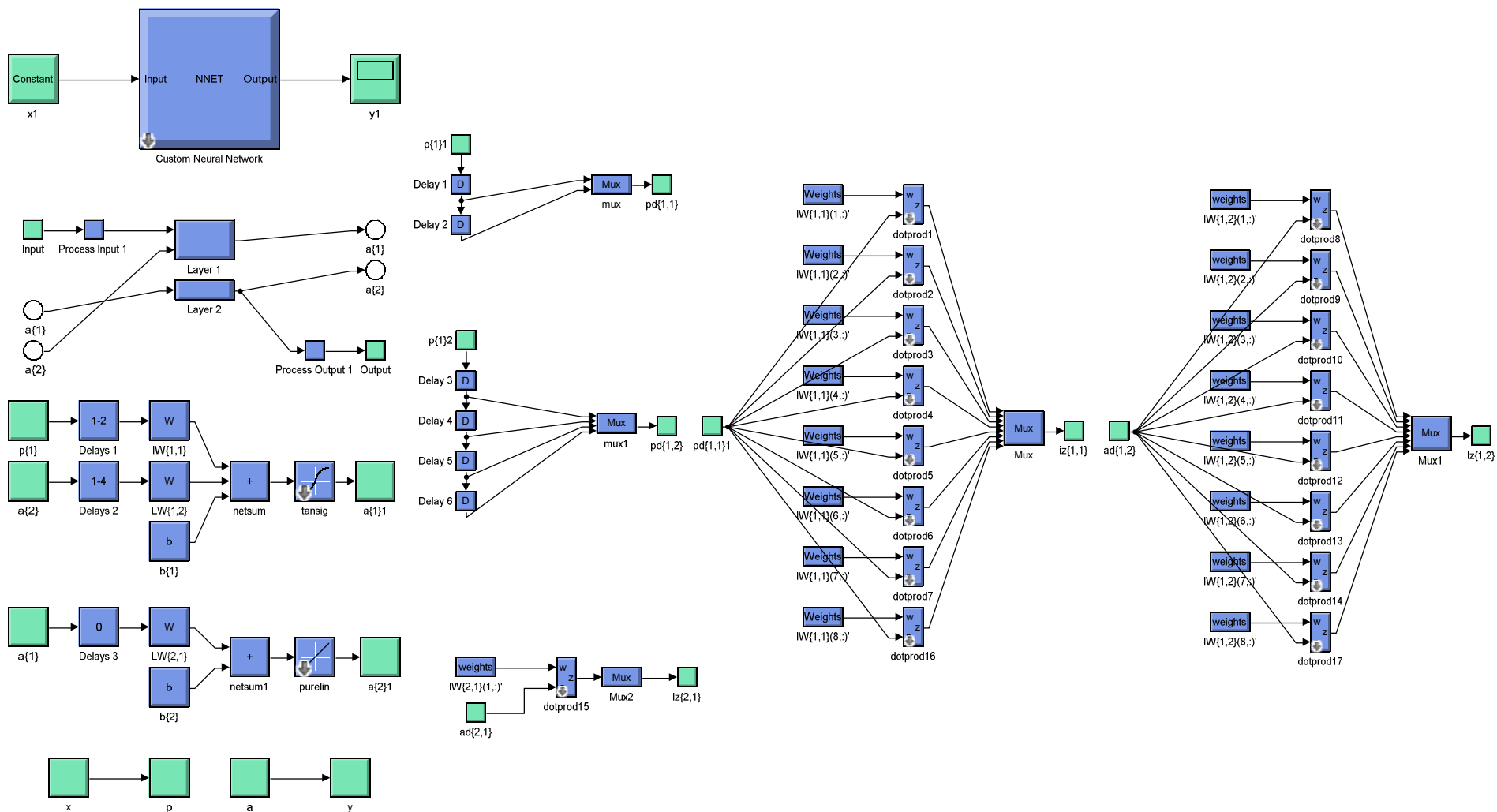


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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.

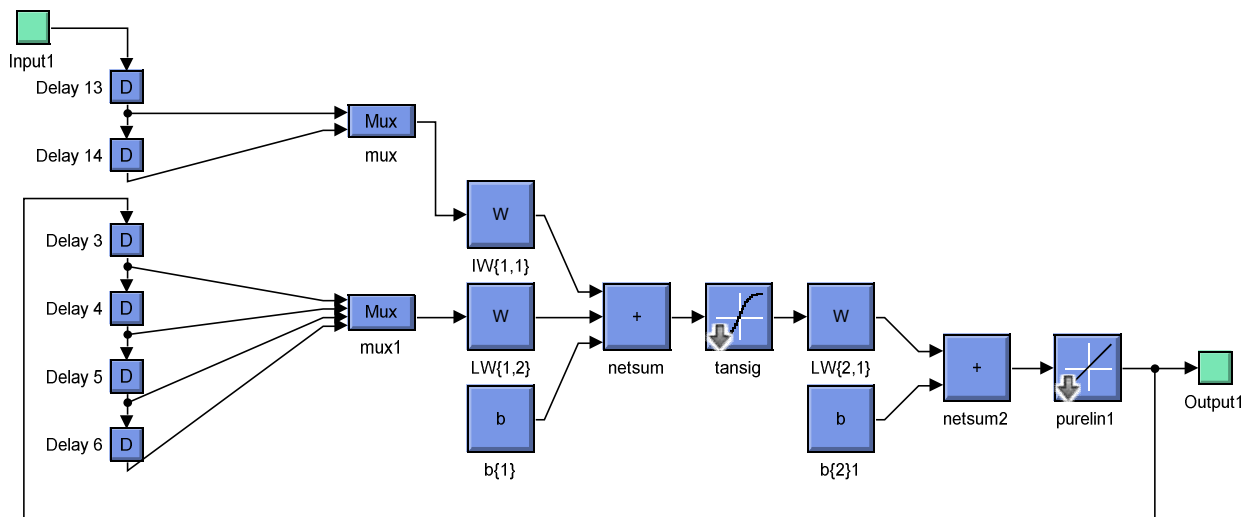


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

В результаті параметри нейромережевої моделі керованого об'єкту вводяться в блок NN Predictive Controller системи Simulink. У системі Simulink дана мережа представляється у вигляді структурної схеми, показаної на рис.4.23.

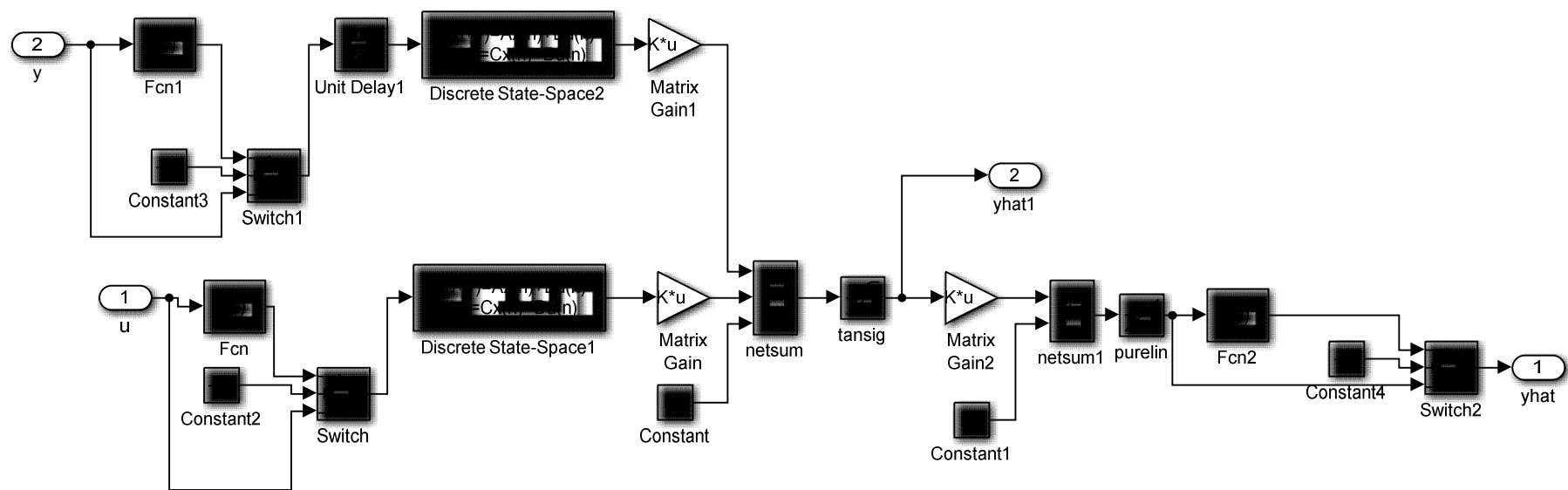


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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.

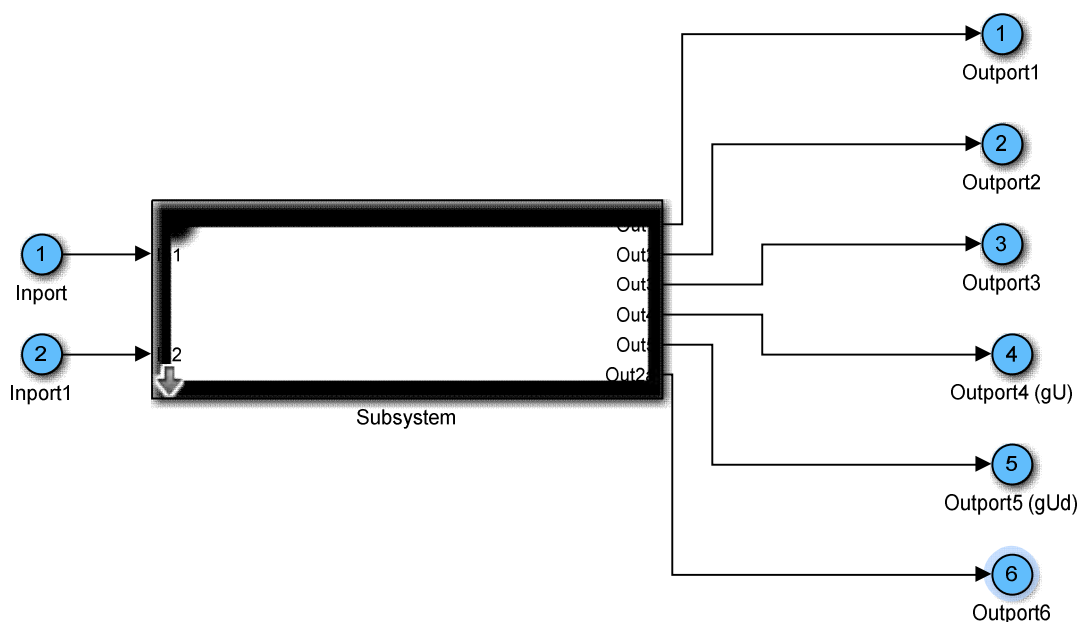


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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.

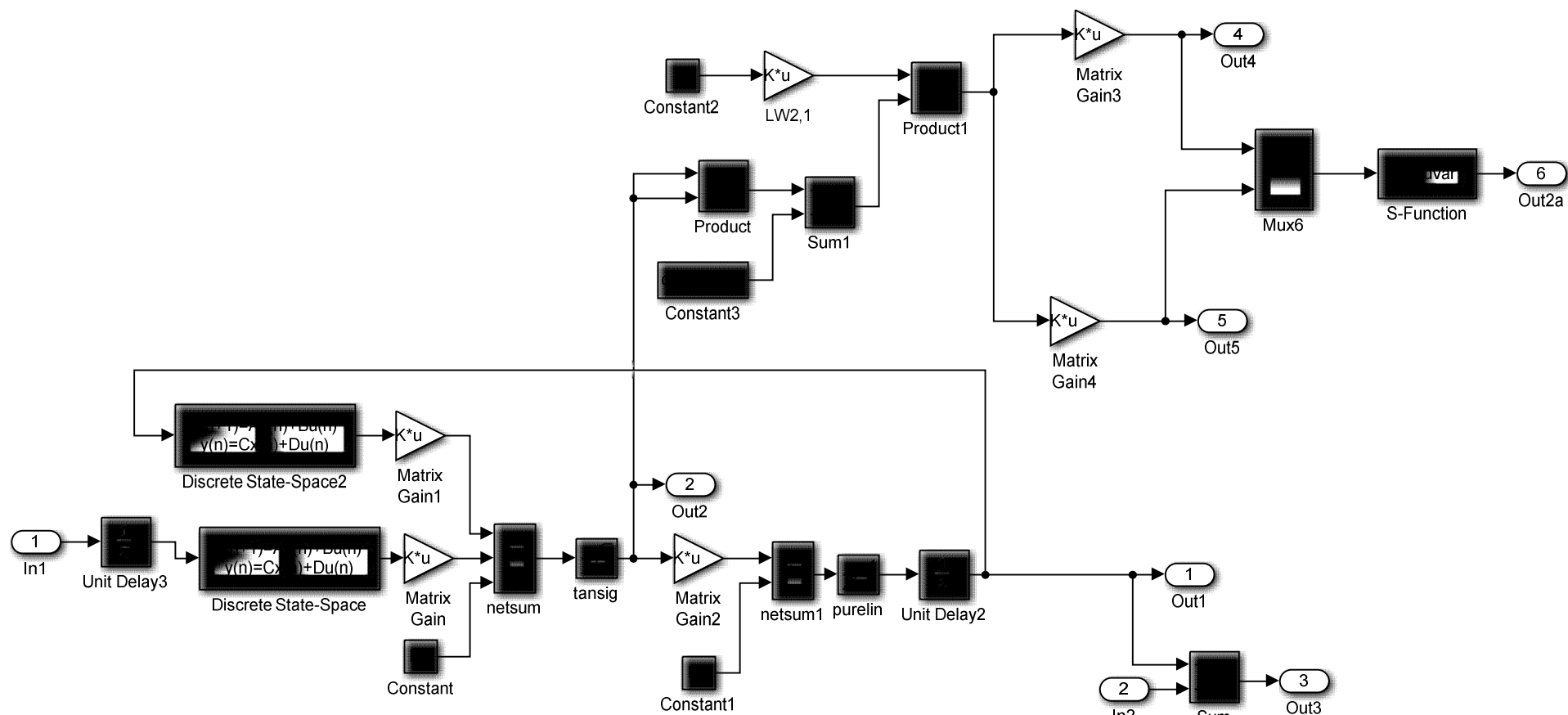


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

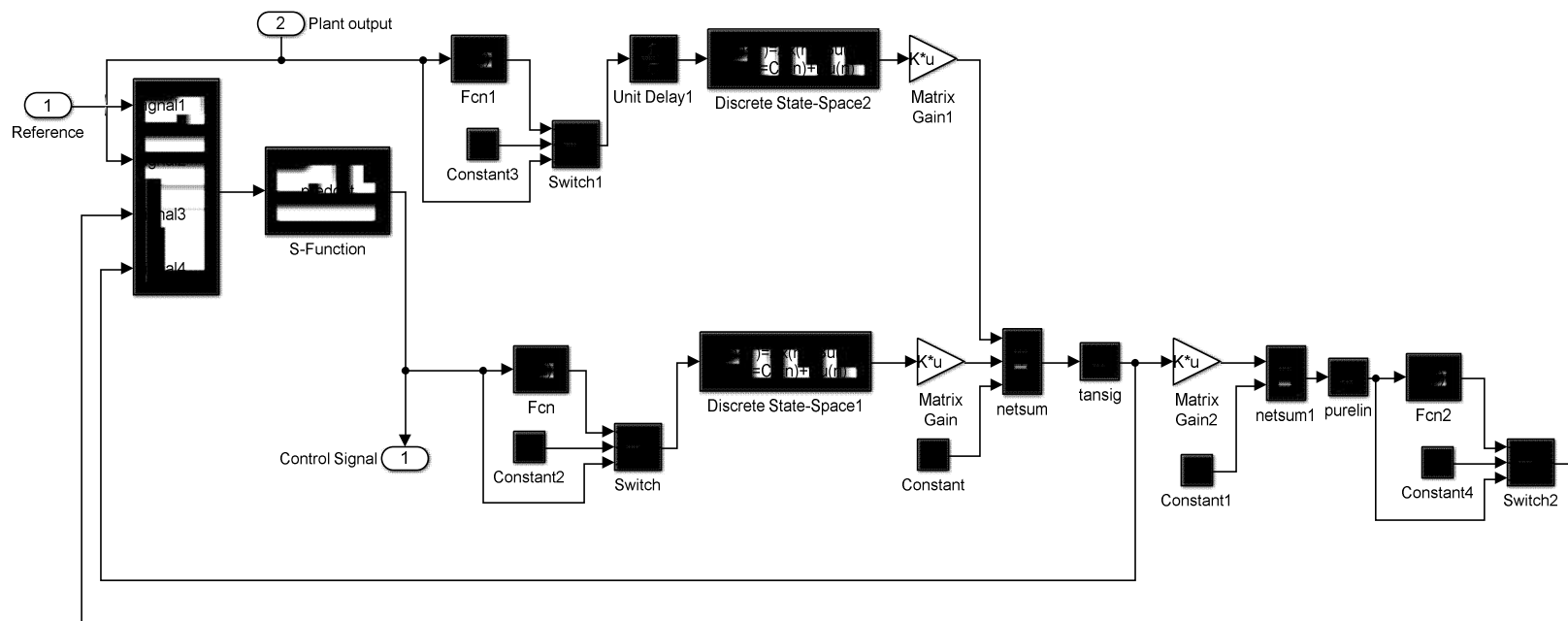


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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 кі-

льність 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.

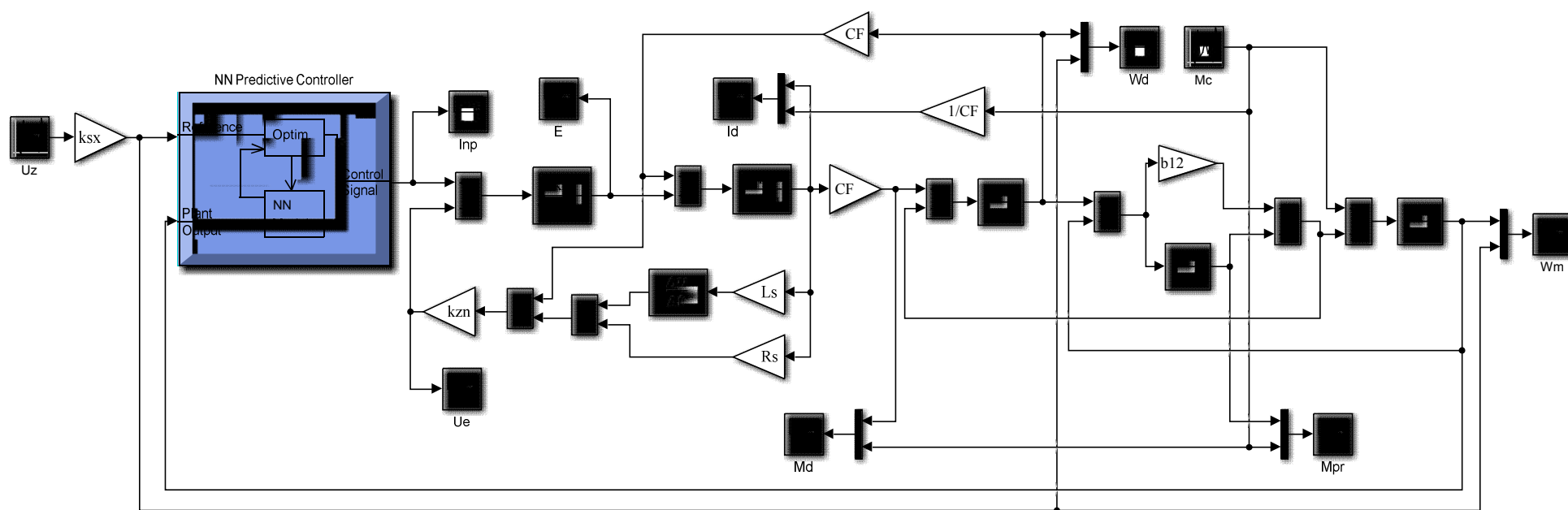
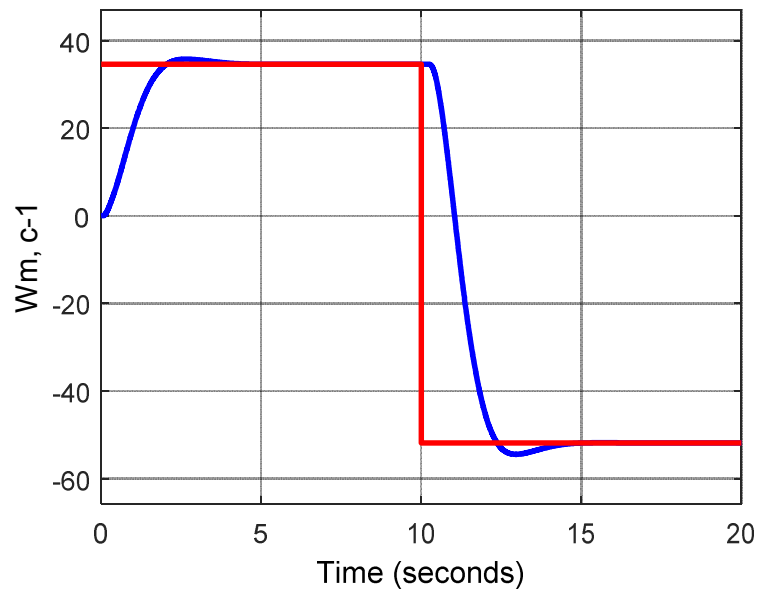
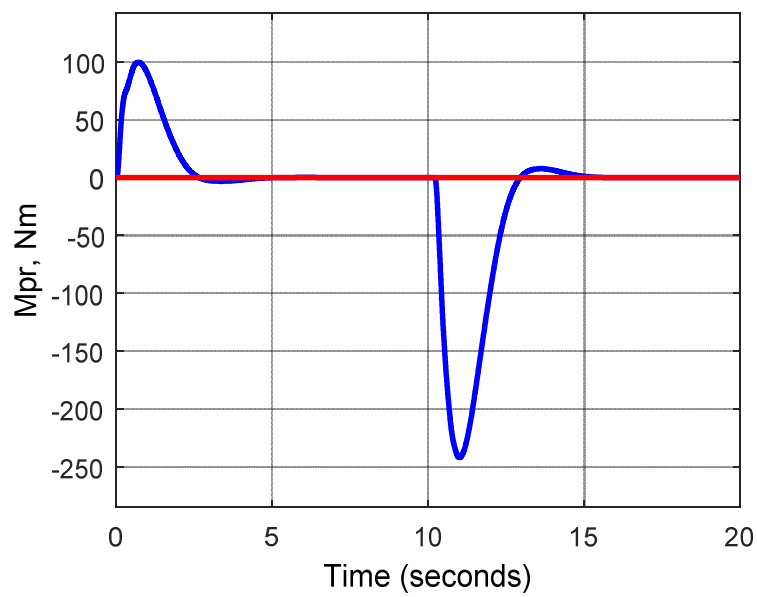


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

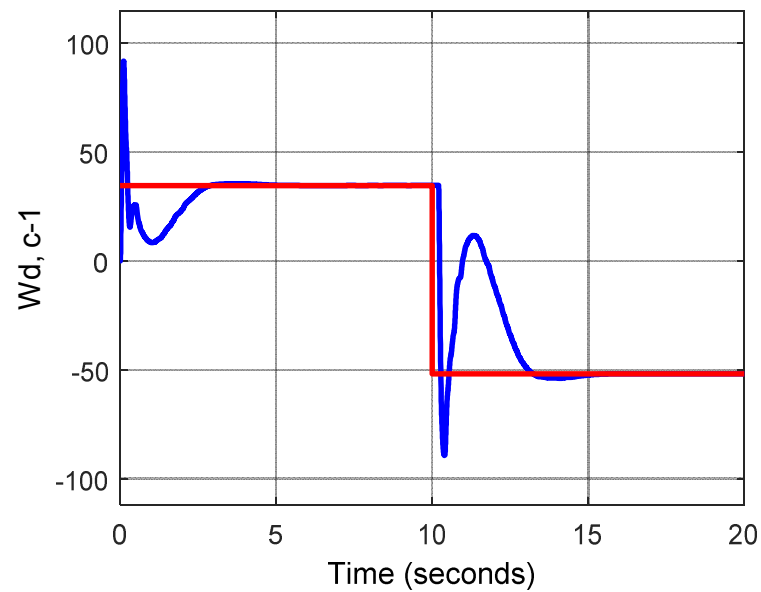


a)

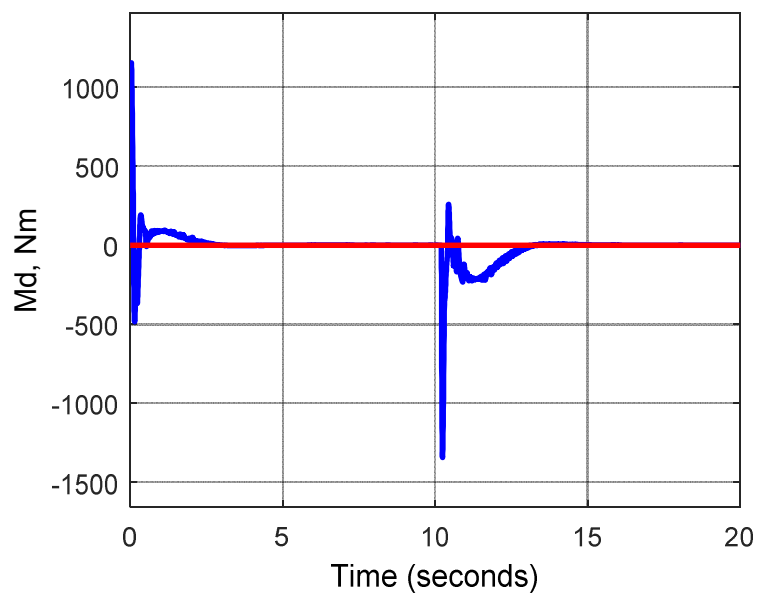


б)

Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором п по задаючій дії

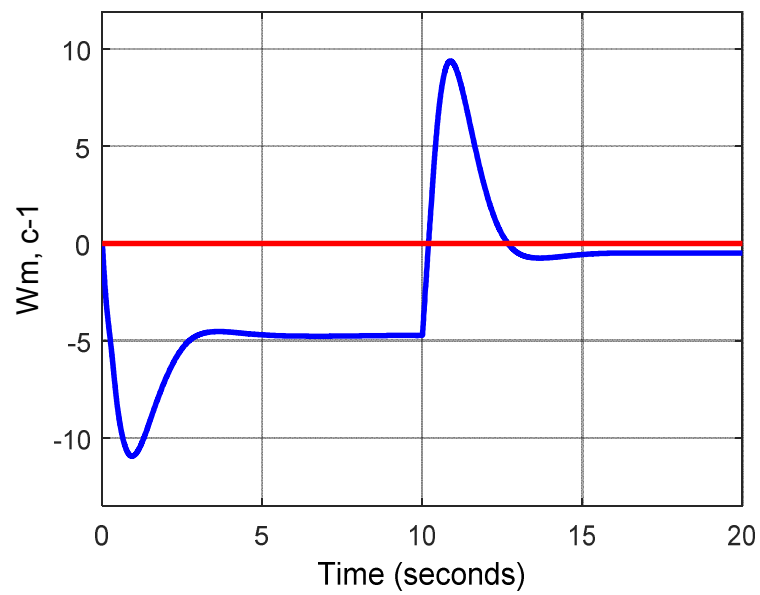


a)

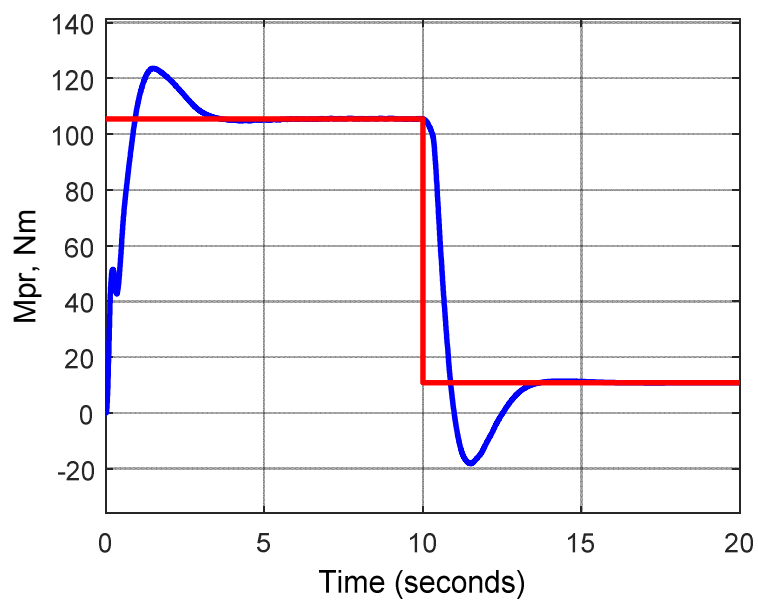


б)

Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії



a)



б)

Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

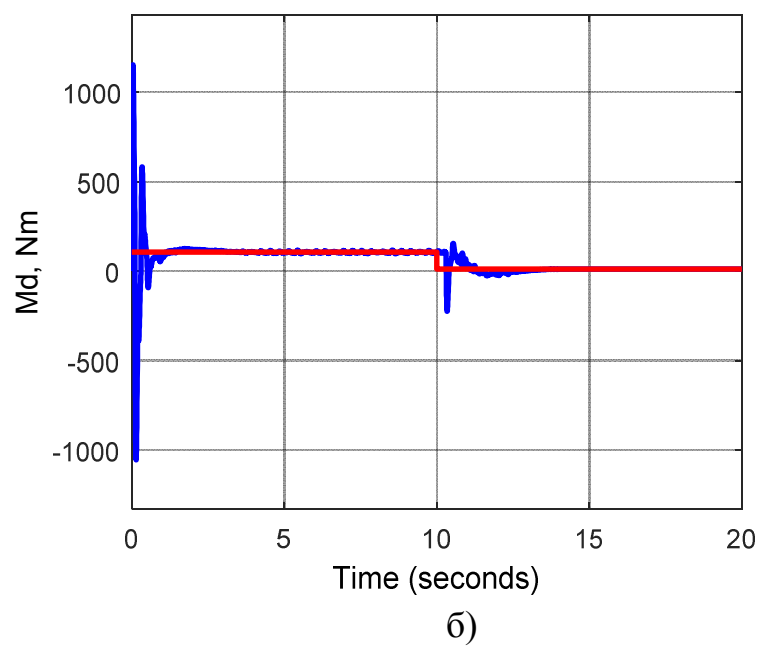
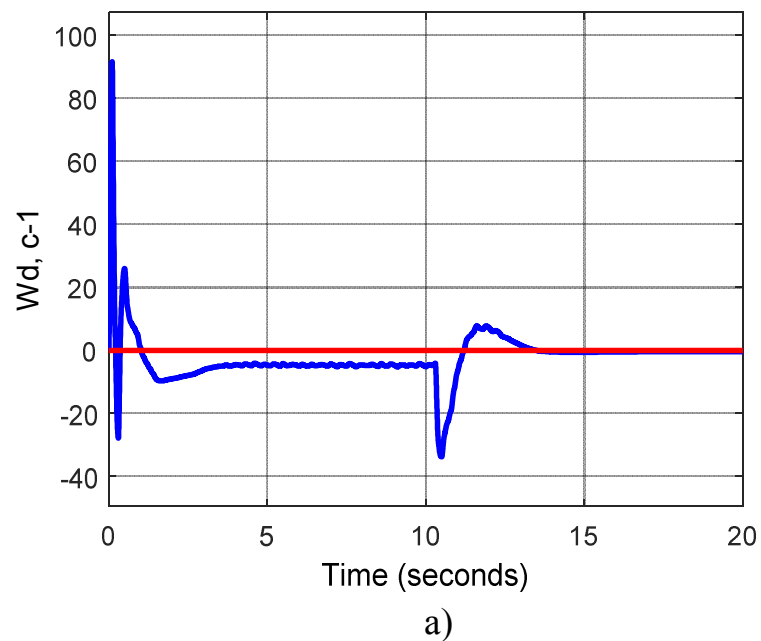


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проаналізовано принципи побудови нейрорегуляторів, 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 розвороту 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 розвороту 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 $N_b = 10000 \div 20000$; інтервал між двома послідовними моментами знімання даних $\Delta t = 0,05 \div 0,1 \text{ c}$, мінімальний інтервал ідентифікації $t_{\min} = 2 \div 8 \text{ c}$; максимальний інтервал ідентифікації $t_{\max} = 10 \div 20 \text{ c}$.

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^{-6} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 0,02.

5. Виконано синтез нейрорегулятора з передбаченням NN Predictive Controller. Шляхом варіювання параметрів нейрорегулятора в широких межах, встановлено, які параметри істотно впливають на якість регулювання та визна-

чено оптимальні значення параметрів. Використання нейромережевої моделі об'єкта управління, що забезпечує високу ступінь ідентифікації, дозволило синтезувати нейрорегулятор, що забезпечує високі показники якості системи.

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи в системі з нейрорегулятором NN Predictive Controller в режимі пуску і набросу навантаження мають задовільний характер. Система є астатичною. Перерегулювання не перевищує 3%, час регулювання складає 4 с, що відповідає завданню.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розв'язана актуальна науково-практична задача розробки нейромережевої системи управління багатомасовою електромеханічною системою механізму розвороту 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 розвороту роторного екскаватора.

2. На підставі аналізу вимог до систем електроприводів основних механізмів роторних екскаваторів для електроприводу механізму розвороту 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. Здійснено 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 з нейрорегулятором NN Predictive Controller при запуску та накиді навантаження мають задовільні характеристики. Перерегулювання не перевищує 3%, час регулювання складає 4 с, що відповідає завданню

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. D.O. Hebb, The Organization of Behavior, John Wiley & Sons, New York, 1949
14. Rosenblatt F. 1962. Principles of neurodynamics. New York: Spartan Books.
(Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир., 1965.)
15. Widrow B. 1959. Adaptive sampled-data systems, a statistical theory of adaptation. 1959 IRE WESCON Convention Record, part 4, pp. 88-91. New York: Institute of Radio Engineers.
16. Widrow B., Hoff M. 1960. Adaptive switching circuits. 1960 IRE WESCON Convention Record, pp. 96-104. New York: Institute of Radio Engineers.
17. McCulloch W. W., Pitts W. 1943. A logical calculus of the ideas imminent in nervous activity. Bulletin of Mathematical Biophysics 5:115-33. (Русский перевод: Маккаллох У. С., Питтс У. Логическое исчисление идей, относящихся к нервной деятельности. Автоматы. – М.: ИЛ. – 1956.
18. Pitts W. McCulloch W. W. 1947. How we know universals. Bulletin of Mathematical Biophysics 9:127-47.
19. Rosenblatt F. 1962. Principles of Neurodynamics. New York: Spartan Books.
(Русский перевод: Розенблатт Ф. Принципы нейродинамики. – М: Мир. – 1965.)
20. Widrow B. 1961. The speed of adaptation in adaptive control system, paper 193-317
21. Widrow B. 1963. A statistical theory of adaptation. Adaptive control systems. New York: Pergamon Press.
22. Widrow B., Angell J. B. 1962. Reliable, trainable networks for computing and control. Aerospace Engineering 21:78-123.
23. Widrow B., Hoff M. E. 1960. Adaptive switching circuits. 1960 IRE WESCON Convention Record, part 4, pp. 96-104. New York: Institute of Radio Engineers
24. M. Minsky and S. Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, Cambridge, Mass., 1969
25. T. Kohonen, Self-Organization and Associative Memory, Third Edition, Springer-Verlag, New York, 1989.
26. Kohonen, T. Self-Organizing Maps / Kohonen T. – Berlin : Springer-Verlag. – 1995. – 362 p

27. Carpenter, G.A. & Grossberg, S. (1987), ART 2: Self-organization of stable category recognition codes for analog input patterns, *Applied Optics*, 26(23), P. 4919-4930.
28. Carpenter, G.A. & Grossberg, S. (1990), ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures, *Neural Networks (Publication)*, 3, P.129-152
29. J.A. Anderson and E. Rosenfeld, "Neurocomputing: Foundation of Research", MIT Press, Cambridge, Mass., 1988
30. Sejnowski T. J., and Rosenberg C. R. 1987. Parallel Networks that learn to pronounce English text. *Complex Systems* 3:145–68.7
31. Burr, D. J. 1987. Experiments with a connectionist text reader. In *Proceedings of the First International on Neural Networks*, eds. M. Caudill and C. Butler, vol. 4, pp. 717–24. San Diego, CA: SOS Printing.1
32. Cottrell, G. W., Munro P., and Zipser D., 1987. Image compressions by backpropagation: An example of extensional programming. *Advances in cognitive science (vol.3)*. Norwood, NJ: Ablex.2
33. Werbos P. J. 1974. Beyond regression: New tools for prediction and analysis in the behavioral sciences. Masters thesis. Harvard University 8
34. Parker, D. B. 1982. Learning-logic. Invention Report, s. 81–64, File 1. Office of Technology Licensing, Stanford University.5
35. Rumelhart D. E., Hinton G. E., and Williams R. J. 1986. Learning internal representations by error propagation. In *Parallel distributed processing*, vol. 1, pp. 318–362. Cambridge, MA: MIT Press.6
36. Gallant S. I., 1988. Connectionist expert system. *Communications of the ACM* 31:152–69.
37. Soloway D. Neural Generalized Predictive Control / Soloway D., Haley P.J //Proceedingof the 1996 IEEE Symposium on Intelligent Control. – 1996. – P. 277–282.
38. Clarke D.W., Mohtadi C., Tuffs C. Generalized Predictive Control - Part 1: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 137-148.

39. Clarke D. W., Mohtadi C., Tuffs C. Generalized Predictive Control. Part 2: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 149-163.
40. Clarke D. W. Advances in model-based predictive control.- In *Advances in Model-Based Predictive Control* / D. W. Clarke: Oxford University Press. – 1994. – 274
41. Montague G. A., Willis M. J., Tham M. T., Morris A. J. Artificial Neural Network Based Control // *Int. Conference on Control*. – 1991. – Vol.1. – P. 266-271.
42. Takahashi Y. Adaptive Predictive Control of Nonlinear Time-Varying System using Neural Network // *IEEE International Conference on Neural Networks*, 1993. – Vol.3. – P.1464-1468
43. Jeong Jun Song, Sunwon Park. Neural Model-Predictive Control for Nonlinear Chemical Processes // *Journal of Chemical Engineering of Japan*. – 1993. – Vol.26. – №4. –P.347-354.
44. Psychogios D. C., Ungar L. H. Nonlinear Internal Model Control and Model Predictive Control using Neural Networks // *5th IEEE Int. Symposium on Intelligent Control*. – 1990. – P.1082-1087.
45. Press W. H., Flannery B. P., Teukolsky S. A., Vetterling V. T. *Numerical Recipes.- The Art of Scientific Computing*. Cambridge University Press, 1988. – 452.
46. Rumelhart D. E., McClelland J. L. and the PDP Research Group. *Parallel Distributed Processing*. The MIT Press. – 1986. – Vol.1. – Chapter 8. – 337 p.
47. Astrom K J , Wittenmark B *Adaptive control*, Reading MA Addison Wesley, 1989.
48. Parks P C Lyapunov redesign of model reference adaptive control systems, *IEEE Journal Transactions of Automatic Control*, 1966, vol 11, pp 362-367.
49. Hornik K , Stinchcombe M , White H Mylti-layer feed-forward networks are universal approximators Discussion paper, Department of Economics, University of California, San Diego, La Jolla, CA, 1988.
50. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.

51. Уоссерман Ф Нейрокомпьютерная техника Теория и практика - М Мир, 1990.
52. Haykin S , Neural Networks A comprehensive foundation, MacMillan College Publishing Co , New York, 1994.
53. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
54. Narendra K S , Parthasarathy K Neural networks and dynamical systems Part 1 A gradient approach to Hopfield networks TR 8820, Center for systems science, Department of Electrical Engineering, Yale University, New Haven, CT, 1988.
55. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
56. Werbos P J , Beyond Regression New Tools for Prediction and Analysis in the Behavioral Sciences, Phd Thesis, Dept of Applied Mathematics, Harvard University, Cambridge, Mass , 1974.
57. Garcia C.E. Model predictive control: theory and practice a survey / Garcia C.E., Prett D.M., Morari M. // Automatica. – 1989. – Vol.25. – P.335-348.
58. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
59. Системы электропривода и электрооборудования роторных экскаваторов. / Ю.Т. Калашников, А.О. Горнов, В.Н. Остриров и др. – М.; Энергоатомиздат, 1988 - 312 с.
60. Подерни Р.Ю. Горные машины и комплексы для открытых работ в 2 томах - М.; МГГУ, Издание 4, 2001 – 754 с.
61. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
62. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
63. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
64. 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.

65. 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.

66. 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.

67. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.

68. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.

69. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.

70. 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

71. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч1.

72. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч2.

73. 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.
74. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоєнко. Харків, ХНАДУ, 2012. – 284 с.
75. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
76. 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.
77. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. – СПб: Питер, 2001. –488 с.
78. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтьєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

ДОДАТОК А

БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ

А.1 Класифікація штучних нейронних мереж

Розглянемо основні положення теорії нейронних, які служать базою для розуміння виконаних в роботі досліджень.

Як було зазначено вище, під нейронними мережами розуміється обчислювальні структури, які моделюють прості біологічні процеси, що зазвичай асоціюються з процесами людського мозку. Вони є системами, здібними до навчання шляхом аналізу позитивних і негативних дій. Елементарним перетворювачем в даних мережах є штучний нейрон або просто нейрон, названий так по аналогії з біологічним прототипом.

Штучні нейронні мережі (ШНМ), що утворюються асоціацією штучних нейронів, є вельми спрощеною аналогією біологічних нервових мереж. При цьому ступінь спрощення обумовлений незрівняними рівнями зв'язності нейронів і їх числом. Нервова система людини має близько 10 нейронів, об'єднаних в мережу з приблизно 1015 зв'язками, що передають. У біологічних нервових системах кожен нейрон володіє безліччю якостей і функцій, серед яких унікальною функцією є прийом, обробка і передача електрохімічних сигналів по нервових волокнах, створюючих комунікаційну систему мозку.

Штучні нейронні мережі відрізняються своєю архітектурою: структурою зв'язків між нейронами (рис. А.1), числом шарів, функцією активації нейронів, алгоритмом навчання. З цієї точки зору серед відомих ШНМ можна виділити статичні, динамічні мережі і fuzzy-структури; одно- або багатошарові мережі. Відмінності обчислювальних процесів в мережах частково обумовлені способом взаємозв'язку нейронів, тому виділяють наступні види мереж:

- мережі прямого розповсюдження (feedforward) – сигнал по мережі проходить тільки в одному напрямі: від входу до виходу;
- мережі із зворотними зв'язками (feedforward/feedback);

- мережі з бічними зворотними зв'язками (laterally connected);
- гібридні мережі.

В цілому по структурі зв'язків ШНМ можуть бути згруповані в два класи:

- 1) мережі прямого розповсюдження – без зворотних зв'язків в структурі;
- 2) рекурентні мережі – із зворотними зв'язками.

У першому класі мереж найбільш відомими і використовуваними є багат шарові нейронні мережі (багат шарові перцептрони по Ф. Розенблатту), де штучні нейрони розташовані шарами.

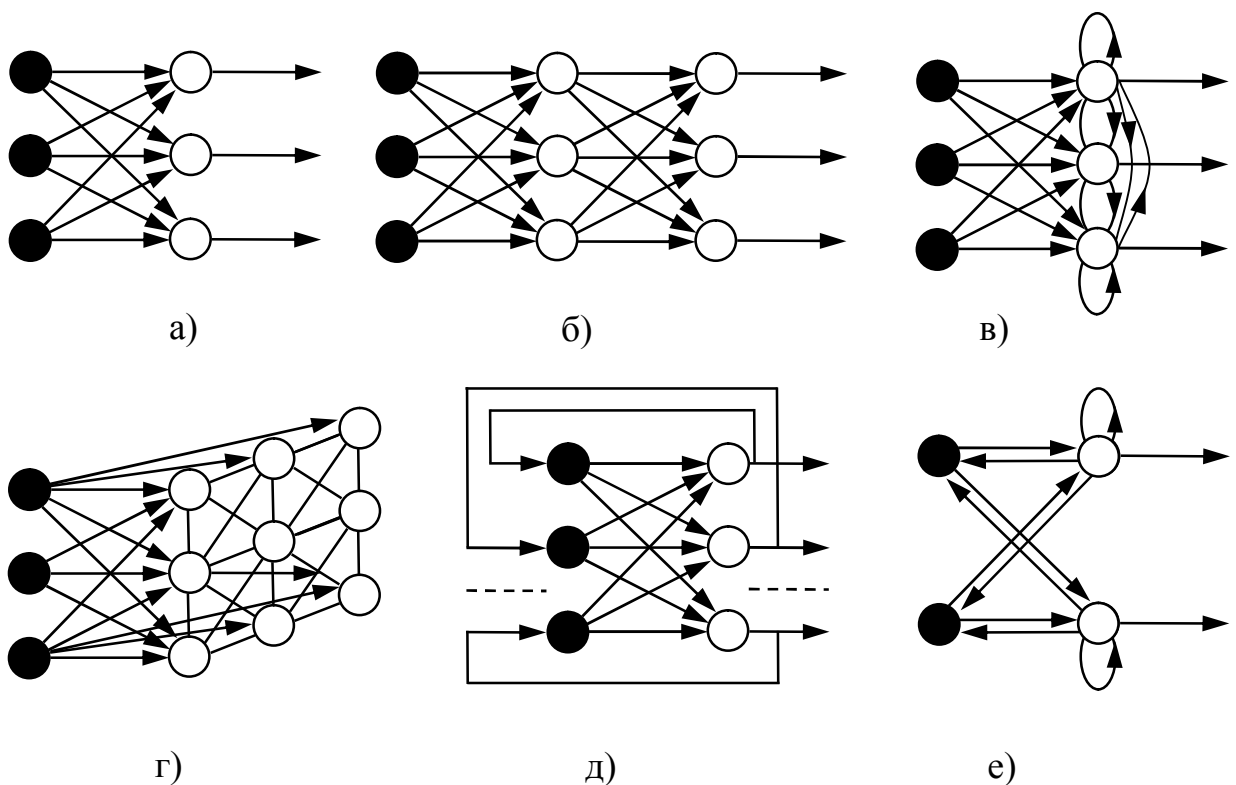


Рисунок А.1 – Структури найбільш поширених ШНМ

а – одношарова мережа; б – багат шарова мережа (перцептрон); в – мережі з «бічними з'єднаннями»; г – мережі Кохонена; д – мережі Хопфілда; е – АРТ-мережі

Зв'язок між шарами – однонаправлена і в загальному випадку вихід кожного нейрона пов'язаний зі всіма входами нейронів подальшого шару. Такі мережі є статичними, оскільки не мають в своїй структурі ні зворотних зв'язків, ні

динамічних елементів, а вихід залежить від заданої множини на вході і не залежить від попередніх станів мережі. На відміну від статичних мереж прямого розповсюдження мережі другого класу (рекурентні) є динамічними, оскільки із-за зворотних зв'язків стан мережі в кожен момент часу залежить від попереднього стану.

У подальшому викладі для синтезу нейромережових системи управління розглядатимемо мережі прямого розповсюдження, тобто мережі без зворотних зв'язків в структурі;

А.2 Базовий процесорний елемент (штучний нейрон)

Базовий процесорний елемент (БПЕ) здійснює відображення $R^n \rightarrow R^1$ відповідно до виразу для виходу q (рис. А.2):

$$q = f\left(\sum_{j=1}^n w_j r_j + w_0 r_0\right) = f\left(\sum_{j=0}^n w_j r_j\right) \quad (\text{A.1})$$

де $r_0, r_1, \dots, r_n$ – входи БПЕ;

$w_0, \dots, w_n$ – вагові коефіцієнти синаптичних зв'язків БПЕ.

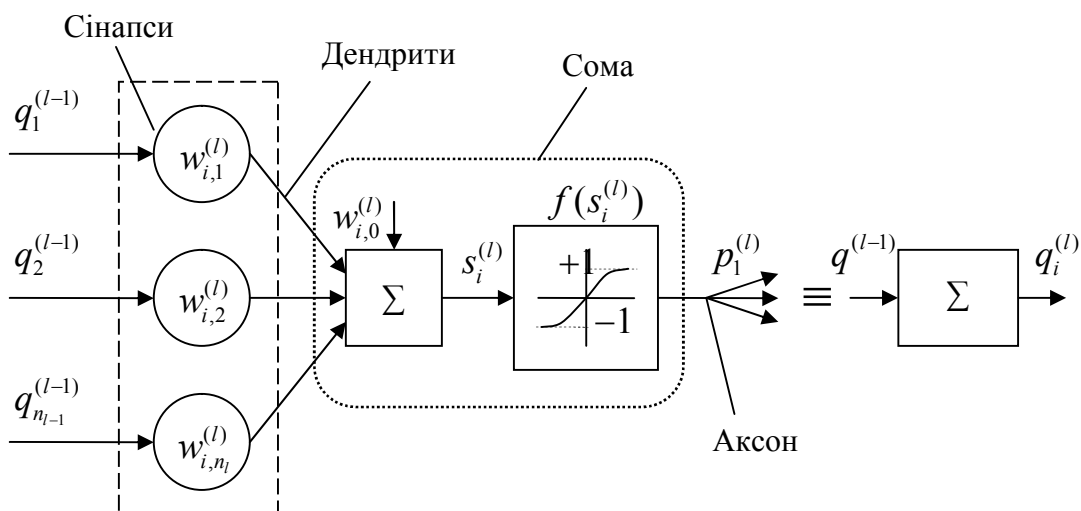


Рисунок А.2 – Схема базового процесорного елементу (штучного нейрона)

Вхід r_0 і коефіцієнт зв'язку w_0 в кожному БПЕ введені спеціально для ініціалізації мережі; зазвичай $r_0 = +1$; всі коефіцієнти w_0 , як і інші $w_j (j \in \overline{1, n})$ настроюються в процесі навчання;

$f(\cdot)$ – «функція активації» – монотонна, така, що безперервно диференціюється на інтервалі або $(-1, +1)$, або $(0, +1)$.

У штучних нейронах найбільшого поширення (і по аналогії з біологічними нейронами) набули наступні функції:

- експоненціальні або радіально-базисні (функції Гауса):

$$f(x) = \exp(-x^2 / 2\sigma^2), \sigma = \text{const};$$

- гіперболічного тангенса:

$$f(x) = \text{th}(x) = (e^x - e^{-x}) / (e^x + e^{-x}); f'(x) = (1 - f(x))(1 + f(x))$$

- сигмоїдні:

$$f(x) = 1 / (1 + e^{-x}); f'(x) = f(x)(1 - f(x));$$

- бінарні (різного визначення).

Серед простих типів штучних нейронів найбільш відома адаліна Б. Уїдроу, для якої функція $f(x)$ – лінійна, а вихід визначається співвідношенням

$$q = \sum_{j=1}^n w_j r_j + w_0 \cdot 1; w_j = \text{var}; j = \overline{0, n}.$$

2.3 Статичні лінійні одношарові мережі.

Проста одношарова лінійна ШНМ складається з K адалін (рис. А.3), називається мадаліна і має K виходів $q_1, \dots, q_i, \dots, q_K$ і $n+1$ входів $r_0, \dots, r_i, \dots, r_j, \dots, r_n$. Вона може бути описана векторний-матричним співвідношенням:

$$q = W^{(k)} r_1 + w_0 r_0 = W^{(k)} r, \quad (\text{A.2})$$

де $\mathbf{q} = col(q_1, \dots, q_i, \dots, q_K)$ – вектор виходу;

$\mathbf{r}_1 = col(r_0, \dots, r_j, \dots, r_n)$ – вектор входу; з урахуванням входу ініціалізації r_0 ,

однакового для всіх адалин,

вектор входу $\mathbf{r}^T = \mathbf{r}_1^T : r_0$;

$\mathbf{W}^{(K)} = \|w_{ij}\|$, $i = \overline{1, K}$, $j = \overline{0, n}$ – матриця вагових коефіцієнтів;

$\mathbf{W}_1^{(K)} = \|w_{ij}\|$ $i = \overline{1, K}$, $j = \overline{0, n}$ – матриця, що отримується з матриці $\mathbf{W}^{(k)}$

видаленням останнього стовпця – вектора коефіцієнтів ініціалізуючих зв'язків

$\mathbf{w}_0 = col(w_{1,0}, \dots, w_{K,0})$.

Відповідно до (A.2) лінійна багатошарова мережа типу мадаліни здійснює відображення $\mathbf{r}_1 \in R^n$ в $\mathbf{q} \in R$.

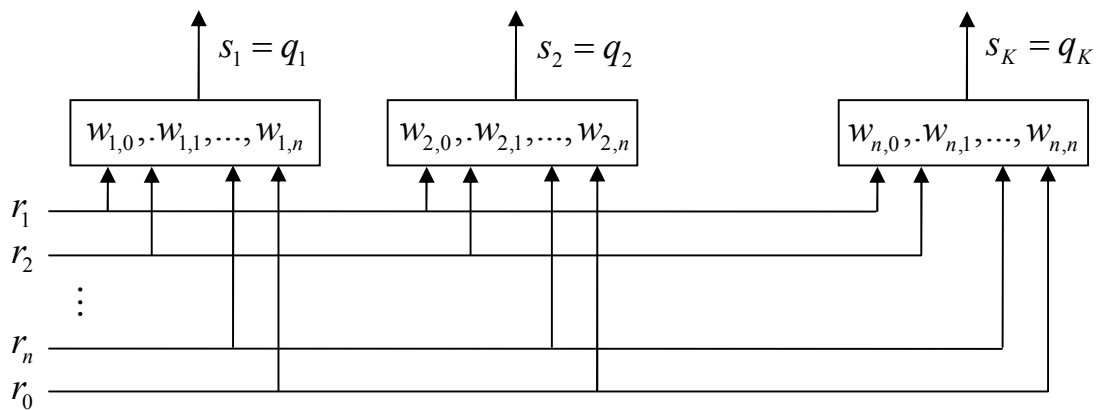


Рисунок А.3—схематичне представлення одношарової лінійної штучної нейронної мережі

А.3 Статичні багатошарові нейронні мережі

Структура статичних багатошарових нейронних мереж (БНМ) зображена на рис. А.4.

У повнозв'язних гомогенних багатошарових нейронних мережах виходи базових елементів кожного шару поступають на входи всіх базових елементів наступного шару, а функція активації $f(s)$ приймається однаковою для всіх БПЕ. Вихідний шар багатошарової мережі виконується у вигляді мадаліни.

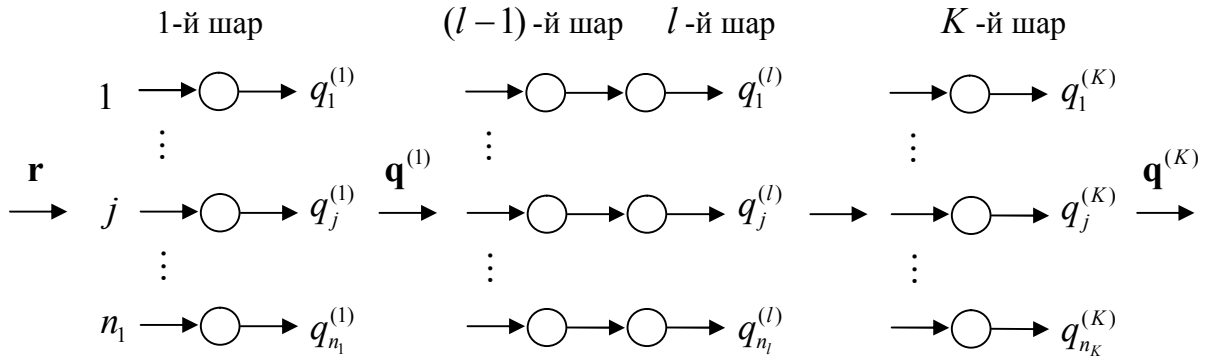


Рисунок А.4 – Структурна схема статичних БНМ

Вхідний шар є буферним для сигналу $\mathbf{r}$. Багатошарова нейронна мережа може бути визначена в символічній формі як $N_{n_0, n_1, \dots, n_K}^K$, де K – число шарів в мережі; n_0 – число входів мережі; $n_i (i = \overline{1, K-1})$ – число базових елементів в i -х «прихованих» або проміжних шарах; n_K – число базових елементів у вихідному K -му шарі і одночасне число виходів $q_1, \dots, q_{n_K}$ багатошарової нейронної мережі. Проміжний l -й шар має n_l базових процесорних елементів. Зв'язки між базовими елементами в шарі відсутні. Виходи базових елементів l -го шару поступають на входи базових елементів тільки наступного $(l+1)$ -го шару. Вихід i -го елемента в l -му шарі багатошарової нейронної мережі може бути визначений так само, як і для будь-якого базового елемента, у вигляді.

$$q_i^{(l)} = f \left(\sum_{j=1}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)} + w_{i,0}^{(l)} q_0^{(l-1)} \right) = f(s_i^{(l)}). \quad (\text{A.3})$$

У векторній формі вихід l -го шару мережі рівний

$$\mathbf{q}_1^{(l)} = \mathbf{f}(\mathbf{W}_1^{(l)} \mathbf{q}_1^{(l-1)} + \mathbf{w}_0^{(l)} q_0^{(l-1)}), \quad (\text{A.4})$$

де $\mathbf{w}_0^{(l)} = \text{col}(w_{1,0}^{(l)}, \dots, w_{n_l,0}^{(l)})$ – вектор вагових коефіцієнтів сигналу ініціалізації

$q_0^{(l-1)}$ в шарі l , який для всіх $l = \overline{1, K}$ приймається рівним $+1$;

$\mathbf{q}_1^{(l)} = \text{col}(q_1^{(l)}, \dots, q_{n_l}^{(l)})$, $\mathbf{q}_1^{(l-1)} = \text{col}(q_1^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$ вектори виходів базових

процесорних елементів шару l і виходів попереднього $(l-1)$ -го шару, базових елементів, що поступають на входи, шару l . Для вихідного K -го шару маємо:

$$\begin{aligned} \mathbf{q}_l^{(k)} &= \mathbf{W}_1^{(K)} \mathbf{q}_1^{(K-1)} + \mathbf{w}_0^{(K)}, \\ \mathbf{q}_l^{(K-1)} &= \mathbf{f}(\mathbf{W}_1^{(K-1)} \mathbf{q}_2^{(K-1)} + \mathbf{w}_0^{(K-1)}). \end{aligned} \quad (\text{A.5})$$

Співвідношення (A.3) – (A.5) описують статичні багат шарові нейронної мережі з нелінійними функціями активації $f(s)$ пошарово. Функцію s_i^l на i -х виходах суматорів базових елементів шару l називають дискримінантною. У загальному випадку – це відрізок багатовимірного ряду Тейлора, і його найбільший ступінь визначає порядок базових елементів. Так, співвідношення (A.3) відповідає виходу елементу першого порядку, а вектор вагових коефіцієнтів відповідного ряду Тейлора $\mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, \dots, w_{i,n_{l-1}}^{(l)})$ є «пам'ять» базового процесорного елементу. Дискримінантна функція є зважена сума входів БПЕ з коефіцієнтами, рівними значенням «вагів», що настроюються $w_{i,j}^{(l)}$. Дискримінантна функція розділяє вхідний вектор (патерн вхідної активності) на класи – мінімум на два класи у разі лінійної мережі з дискримінантною функцією першого порядку. В цьому випадку функція $s_j = 0$ – дискримінантна лінія, яка розділяє вхідну множину $q_1^{(l-1)}, q_2^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}$ на дві підмножини:

$$q_j^{(l)} = \begin{cases} 1, & \text{якщо } s_j > 0 \\ 0, & \text{якщо } s_j < 0 \end{cases}.$$

Для елементів другого порядку дискримінантна функція визначається таким чином:

$$s_i^{(l)} = w_{i,0}^{(l)} + \sum_{j=1}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)} + \sum_{j=1}^{n_{l-1}} \sum_{k=1}^j w_{j,k}^{(l)} q_j^{(l-1)} q_k^{(l-1)} = (\mathbf{w}_i^{(l)})^T \mathbf{q}_1^{(l-1)}. \quad (\text{A.6})$$

де $w_{i,0}^{(l)}, w_{i,j}^{(l)}, w_{j,k}^{(l)}$ ($j = \overline{1, n_{l-1}}, k \leq j$) – компоненти вектора вагових коефіцієнтів

i -го базового елементу в l -му шарі мережі, а вектор входу $q^{(l-1)}$ має вигляд:

$$\mathbf{q}^{(l-1)} = \text{col}(1, q_1^{(l-1)}, q_2^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}, (q_1^{(l-1)})^2, q_1^{(l-1)}, q_2^{(l-1)}, \dots, (q_{n_{l-1}}^{(l-1)})^2). \quad (\text{A.7})$$

З (A.6) і (A.7) витікає, що в кожному l -му шарі нейронної мережі з елементами другого порядку разом з лінійним перетворенням виходів попереднього шару $\mathbf{q}^{(l-1)}$ здійснюється білінійне перетворення цих же сигналів.

Нейронні мережі з дискримінантними функціями високого порядку дозволяють розділяти (класифікувати) вхідну множину на класи за допомогою нелінійних розділяючих поверхонь. Зокрема, розділяюча поверхня $S_i^{(l)} = 0$, де $S_i^{(l)}$ визначається співвідношенням (A.6), є розділяючою поверхнею другого порядку і отримала назву гіперквадрика.

БПЕ з дискримінантною функцією високого порядку має істотно більше число «синаптичних» зв'язків і коефіцієнтів, що настраюються, у складі вектора $\mathbf{w}_i^{(l)}$ при однаковому числі шарів і елементів в структурах багат шарових нейронних мереж з базовими елементами як першого, так і другого порядків. У подальшому будемо розглядати багат шарові нейронні мережі з базовими процесорними елементами першого порядку.

Визначимо зв'язок між входом $\mathbf{r}$ і виходом $\mathbf{u} = \mathbf{q}^{(K)}$ нейронної мережі з елементами першого порядку. Використовуючи (A.4) і $(n_1 \times (n_{l-1} + 1))$ матрицю вагових коефіцієнтів базових елементів l -го шару

$$\mathbf{W}_1^{(l)} = \left[w_0^{(l)} : w_1^{(l)} : w_2^{(l)} : \dots : w_{n_l}^{(l)} \right]^T = \left[(w_0^{(l)}) : (\mathbf{W}_1^{(l)})^T \right], \quad (\text{A.8})$$

запишемо у векторний-матричній формі вихід l -го шару:

$$\mathbf{q}_1^{(l)} = \mathbf{f}(w_0^{(l)} q_0^{(l-1)} + \mathbf{W}_1^{(l)} \mathbf{q}_1^{(l-1)}) = \mathbf{f}^{(l)}(\mathbf{q}^{(l-1)}), \quad q_0^{(l)} = +1; \quad l = \overline{1, K}. \quad (\text{A.9})$$

У (A.9) $\mathbf{f}^{(l)}$ визначає нелінійне перетворення кожній компоненти вектора $\mathbf{q}^{(l-1)}$ так, як це встановлено співвідношенням (A.3).

Нелінійне рекурентне рівняння, застосоване послідовно для всіх $l = K, K-1, K-2, \dots, 2, 1$ дозволяє встановити зв'язок між входом $\mathbf{r}$ і виходом $\mathbf{u} = \mathbf{q}^{(K)}$ мережі у вигляді:

$$\begin{aligned} \mathbf{u} = \mathbf{q}^{(K)} = & \mathbf{f}^{(K)}(\mathbf{w}_0^{(K)} + \mathbf{W}_1^{(K)} \mathbf{f}^{(K-1)}(\mathbf{w}_0^{(K-1)} + \mathbf{W}_1^{(K-1)} \mathbf{f}^{(K-2)} \times \\ & \times (\dots \mathbf{w}_0^{(l)} + \mathbf{W}_1^{(l)} \mathbf{f}^{(l-1)}(\mathbf{w}_0^{(l-1)} + \mathbf{W}_1^{(l-1)} \mathbf{f}^{(l-2)} \times \\ & \times (\dots \mathbf{w}_0^{(2)} + \mathbf{W}_1^{(2)} \mathbf{f}_1(\mathbf{w}_0^{(1)} + \mathbf{w}_1^{(1)} \mathbf{q}^{(0)}) \dots)) \dots)) = \mathbf{F}(\mathbf{r}). \end{aligned} \quad (\text{A.10})$$

З (A.10) витікає, що багатошарова нейронна мережа виконує через свою структуру складне нелінійне перетворення $\mathbf{F}(\mathbf{r})$ вхідного вектора $\mathbf{r}$ залежно від векторів вагових коефіцієнтів $\mathbf{w}^{(l)}$ нелінійних функцій активації $\mathbf{f}^{(l)}$.

ДОДАТОК Б

ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

Б. 1 Здібність до навчання і накопичення інформації

Класичне завдання навчання багатошарової нейронної мережі зводиться до виконання умов, при яких по заданій послідовності $\mathbf{u}^*$ з вказівкою їх приналежності до одного з класів ситуацій, виробляється рішення, що дозволяє класифікувати «незнайомі» ситуації, що знов пред'являються. Зокрема, для статичних багатошарових нейронних мереж таке рішення $\mathbf{u}^*$ формується шляхом настройки вектора вагових коефіцієнтів $\mathbf{w}_i^l$. Якість рішення визначається критерієм $Q(\mathbf{u}^*, \mathbf{u})$, а множинність рішень $\mathbf{u}$ – «конструкцією» алгоритму настройки вагових коефіцієнтів $\mathbf{w}_i^l$. Процес навчання в цьому випадку направлений на досягнення якнайкращого рішення з множини можливих і розуміється при цьому як процес накопичення інформації $\mathbf{u}^*$ і одночасно процес вибору рішення $\mathbf{u}$.

Б.2 Апроксимаційні властивості нейронних мереж

Ці властивості грають вирішальну роль для використання багатошарових нейронних мереж в задачах управління. Доведений ряд теорем що математично обґрунтовують властивості апроксимацій мереж з різними функціями активації. З схем, представлених в попередніх параграфах, витікає, що нейронні мережі обчислюють лінійні або нелінійні функції однієї змінної, а також всілякі суперпозиції функції від функцій. Що можна отримати, використовуючи тільки такі операції? Які функції вдасться обчислити точно, а які можна скільки завгодно точно апроксимувати за допомогою нейронних мереж? Щоб вивчити їх можливості, потрібно відповісти на ці питання. Виникає і більш загальне питання: чи можна довільну безперервну функцію n змінних представити за допомогою операцій складання, множення і суперпозиції з безперервних функцій однієї

або двох змінних? Таке наближення функцій багатьох змінних за допомогою лінійних операцій і суперпозицій функцій одного змінного і здійснюється нейронними мережами. Кожна мережа складається з штучних нейронів – базових процесорних елементів. Нейрон отримує на вході вектор сигналів $\mathbf{q}$, обчислює його скалярний добуток на вектор вагів $\mathbf{w}$ і деяку функцію одного змінного $f(s)$. Результат поступає на вході інших базових елементів або на вихід. Таким чином, нейронні мережі обчислюють суперпозиції простих функцій одного змінного і їх лінійних комбінацій.

У серії робіт А. Н. Колмогоров і В. И. Арнольд вирішили наступну математичну проблему (складову сутність тринадцятої проблеми Гільберта): будь-яку безперервну функцію n змінних можна точно представити за допомогою операцій складання, множення і суперпозиції з безперервних функцій одного змінного. У теоремі Колмогорова про представлення будь-якої безперервної функції n змінних у вигляді суперпозиції функцій однієї змінної доведено наступне співвідношення:

$$f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \chi_q \left[\sum_{p=1}^n \psi^{pq}(x_p) \right]. \quad (\text{Б.1})$$

де χ_q, ψ^{pq} – безперервні функції однієї змінної, причому χ_q – спеціальні функції для кожної функції $f(x_1, \dots, x_n)$;

ψ^{pq} – незалежно від $f(x_1, \dots, x_n)$ вибрані функції, стандартні для свого класу. Доведено, що функції χ_q можна замінити єдиною функцією χ , а функції ψ^{pq} – на $\alpha^{pq} \psi_p$, де множники $\alpha^{pq} = \text{const}$ і ψ_p – монотонно зростаючі функції.

Пізніше R. Hecht-Nielsen застосував теорему Колмогорова для пояснення апроксимаційних властивостей багатошарових нейронних мереж і показав, що можливе представлення будь-якого безперервного відображення $f: x \in R^n \rightarrow (f_1(x), \dots, f_n(x)) \in R^m$ в мережі з двома прихованими шарами з n БПЕ в першому шарі і $(2n+1)$ БПЕ – в другому. Проте існує, по-перше, необ-

хідність, витікаюча з теореми Колмогорова, в різних функціях активації для кожного БПЕ в першому прихованому шарі i , по-друге, залежність функцій другого шару від форми функції, що апроксимується. Доведені універсальні апроксимаційні властивості багатошарових нейронних мереж з сигмоїдними функціями активації.

У подальшому доведений ряд теорем про апроксимацію безперервних функцій багатьох змінних нейронними мережами з використанням практично довільної безперервної функції одного змінного. Ця функція дійсно може бути довільною, див., наприклад, узагальнену теорему апроксимації Стоуна. У свою чергу теорема Стоуна є узагальненням відомої теореми Вейерштрасса про рівномірне наближення безперервних функцій поліномами. Тому теорема Стоуна-Вейерштрасса служить теоретичним обґрунтуванням застосовності штучних нейронних мереж для вирішення різноманітних завдань управління.

Нейронні мережі дозволяють з будь-якою точністю обчислювати довільну безперервну функцію $f_1(s_1, \dots, s_n)$. Отже, з їх допомогою можна скільки завгодно точно апроксимувати функцію, породжену будь-якою безперервною системою. Тому на головне питання: що можуть нейронні мережі? відповідь отримана: нейронні мережі можуть все або майже все – залежно від числа шарів і базових елементів. Залишається відкритим інше питання – як їх цьому навчати? Який механізм апроксимації і пам'яті штучних нейронних мереж? Нижче розглядаються окремі аспекти цієї проблеми.

Багатошарова мережа з одним прихованим шаром і сигмоїдними функціями активації, навчена по градієнтному алгоритму, може апроксимувати функції із заданою точністю у разі відсутності обмеження на число базових елементів в шарі. У ряді робіт дано теоретичне обґрунтування емпіричному спостереженню, що мережі з двома прихованими шарами забезпечують велику точність при обмеженому числі базових елементів в шарах, порівняно з мережами з одним прихованим шаром.

Б.3 Апроксимація і пам'ять в багат шарових нейронних мереж

Як указувалося вище, багат шарова нейронна мережа здійснює нелінійне перетворення $\mathbf{F}$ вхідної множини $\mathbf{r} \in R^N$ у вихідну множину $\mathbf{u} \in U$. Особливістю мережі є її здатність в процесі навчання наближати $\mathbf{u}$ до заданої множини $\mathbf{u} \in U^* \subset U$ з будь-якою точністю, запам'ятовувати умови перетворення у вигляді значень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ ($l = \overline{1, K}$, $i = \overline{1, n_l}$) базових елементів і при повторних пред'явленнях на вході мережі навіть спотвореної множини $\mathbf{r}$ практично відразу ж, без настройки елементів, відновлювати множину $\mathbf{u} \approx \mathbf{u}^*$.

Найбільше застосування, у тому числі і в прикладних завданнях теорії управління, знайшли сигмоїдні, багат шарові нейронні мережі. Поширеність сигмоїдних нейронних мереж не випадкова. Через аналітичні властивості функцій сигмоїдного типу $f(x)$ (нагадаємо, це – непарні, симетричні, такі, що монотонно зростають від 0 до +1 для сигмоїдної функції і від -1 до +1 для функції гіперболічного тангенса, диференціюємі по аргументу x функції) множина всіх базових елементів багат шарової мережі при установці початкових значень коефіцієнтів $\mathbf{w}_i^{(l)}(0) = \mathbf{W}$ розбивається на підмножини слабо активованих БПЕ з $f'(x) \approx 0$ для $|x| > a$ і сильно активованих БПЕ, для яких $0 < f'(x) < \max f'(x)$ при $|x| < a$.

Під час подачі на вхід мережі сигналу $\mathbf{r}$ в процесі навчання підстроюються коефіцієнти $w_{ij}^{(l)}$ лише активованих БПЕ, оскільки в багат шаровій нейронній мережі (це можна показати), $w_{ij}^{(l)} \cong f'(x), x = s_i^{(l)}$. Значення $|x| = a$ визначають аргумент $f(x) = \text{th}(x)$, що розділяє статичну функцію на області, де $f'(x) \neq 0$ ($|x| \leq a$) і $f'(x) \approx 0$ ($|x| > a$) з центром симетрії функції на початку координат $x = 0$. Для сигмоїдної функції $f(x) = (1 + e^{-x})^{-1}$ центр симетрії визначається при $x = 0,5$ і $f'(x) \neq 0$ при $(0,5 - a) \leq x \leq (0,5 + a)$, а при $x < (0,5 - a)$ і $x > (0,5 + a)$ $f'(x) \approx 0$. Необхідно відзначити, що через довільний вибір почат-

кових коефіцієнтів $w_i^{(l)}(0)$ початкове розбиття безлічі базових процесорних елементів на підмножини слабо і сильно активованих елементів має випадковий характер. Особлива ситуація має місце, коли при «невдалій» установці початкових значень $w_i^{(l)}(0)$ надходженні на входи мережі сигналу $\mathbf{r}$ майже всі елементи виявляються слабо активованими. Такий стан називають паралічем мережі. При паралічі процес навчання йде украй поволі або взагалі припиняється. Можливість виникнення паралічу мережі як на стадії установки початкового стану, так і в процесі навчання належить до серйозних недоліків навчаних по алгоритму ВР сигмоїдних багатошарових нейронних мереж. Для боротьби з цим явищем відомі лише евристичні способи.

Для з'ясування питання про те, що відбувається в мережі в процесі апроксимації нелінійним перетворенням вхідної безлічі $\mathbf{r}$ бажаної функції $\mathbf{u}^*$ при виділенні серед всієї множини базових елементів непорожньої підмножини активованих елементів, запишемо для вихідного K -го шару співвідношення:

$$u_i = q_i^{(K-1)} = \mathbf{w}_i^{(K)} \mathbf{q}^{(K-1)} + \mathbf{w}_{i,0}^{(K)} \cdot 1 = \mathbf{w}_i^{(K)} \mathbf{f}^T(\mathbf{s}^{(K-1)}) + \mathbf{w}_{i,0}^{(K)}; \quad (Б.2)$$

$$i = \overline{1, n_K}.$$

Тут $\mathbf{f}(\cdot)$ – векторна функція, компонентами якої є сигмоїдні функції:

$$f(s_i^{(K-1)}) = f\left(\sum_{j=1}^{n_{K-2}} w_{i,j}^{(K-1)} q_j^{(K-2)} + w_{i,0}^{(K-1)}\right) = f((\mathbf{q}^{(K-2)})^T \mathbf{w}_i^{(K-1)}), \quad (Б.3)$$

$$i = \overline{1, n_{K-1}}$$

Очевидно, що для будь-якого проміжного шару $l = \overline{K-1, 1}$ можна записати подібні співвідношення:

$$f(s_i^{(l)}) = f(q^{(l-1)})^T \mathbf{w}_i^{(l)},$$

$$q^{(l-1)} = \text{col}(+1, q_1^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)}); \quad \mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, w_{i,1}^{(l)}, \dots, w_{i,n_{l-1}}^{(l)}); \quad (Б.4)$$

$$i = \overline{1, n_l}.$$

Якщо зіставити (Б.2) з (Б.3) або (Б.4), то без втрати спільності, наприклад для $s_i^{(K-1)}$, можна записати:

$$f(s_i^{(K-1)}) = f(kx), \quad x = (\mathbf{q}^{(K-2)})^T \mathbf{w}_i^{(K-1)}(0), \quad (\text{Б.5})$$

де k – скалярний множник, що в середньому враховує зміну вагових коефіцієнтів i -го елементу в шарі $K-1$ від початкового значення $\mathbf{w}_i^{(K-1)}(0)$ відповідно до алгоритму навчання. Для функції активації типу гіперболічного тангенса ця зміна повинна бути такою, щоб значення скалярної величини $s_i^{(K-1)}$ ($s_i^{(l)}$ – в загальному випадку) для активованих елементів не перевершувало по абсолютній величині заданого числа a з тим щоб $f'(s_i^{(K-1)}) \neq 0$.

Оскільки $df(kx)/dx = k(df(kx)/d(kx))$, через монотонне і обмежене зростання сигмоїдних функцій $f(kx) > f(x)$ для $k > 1$, $x \leq a$ при $f'(x) > 0 \vee x \in (-\infty, +\infty)$ із зміною k (тобто із зміною в процесі навчання мережі вагових коефіцієнтів $\mathbf{w}_i^{(K-1)}$ в даному випадку) змінюється інтервал значень $s_i^{(K-1)}$, для яких $f'(s_i^{(K-1)}) = f'(kx) \approx 0$. Зокрема, із зростанням k ($k > 1$) інтервал значень $s_i^{(K-1)}$, в якому $f'(\cdot) \approx 0$, розширюється і відповідний i -й елемент ($i = \overline{1, n_{K-1}}$) може переміститися в підмножину слабо активованих елементів. Оскільки $\max_x f(kx) > \max_x f(x)$ і ще $k > 1$ то центр симетрії функції $f(kx)$ і відповідно $f'(kx)$ зміщується по осі $s = kx$ на величину $a(1 - k^{-1})$ (щоб це показати, необхідно записати і проаналізувати другу похідну $d^2 f(kx)/dx^2$ з умови рівності її нулю при $kx = a$). Таким чином, залежно від початкових значень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ ($l = \overline{1, K-1}$) і в процесі їх настройки по алгоритму ВР безліч базових елементів мережі розшаровується на підмножини сильно і слабо активованих елементів при нелінійному перетворенні вхідної множини $\mathbf{r} \in R^{(N)}$. Настройка вагових коефіцієнтів слабо активованих елементів трохи змінює вихід мережі $\mathbf{u} = \mathbf{q}^{(K)}$ унаслідок малих значень похідних $f'(s_i^{(l)})$ функцій активації цих елементів. Настройка вагових коефіцієнтів базових елементів другої мно-

жини приводить до віртуальних зсувів центрів симетрії функцій $f(s)$ уздовж осі s і відповідно центрів симетрії парних функцій похідних $f'(s)$. Ці зсуви, як і значення відповідних коефіцієнтів активованих елементів, в результаті навчання утворюють в структурі багат шарової нейронної мережі асоціативну пам'ять про вхідну множину $\mathbf{r}$ в класі R^n в процесі його нелінійного перетворення (Б.1) що апроксимує задану множину $\mathbf{u}^*$. При повторних пред'явленнях на вході мережі нових $\mathbf{r}$ з того ж класу R^n підстроюються лише небагато з сильно активованих елементів, і через це мережу здійснює перетворення $\mathbf{r} \in R^{(N)}$ в $\mathbf{u} \in U$ практично безінерційно. Слід зазначити, що вибір кращого рішення зі всіх можливих в БНМ з сигмоїдними функціями активації здійснюється лише по одному критерію: близькістю до нуля похідної функції активації по аргументу. В цьому випадку в процесі апроксимації змінюється лише форма вибраної функції.

Статичне перетворення (Б.2) загального виду для конкретної багат шарової нейронної мережі з функціями активації $f(s)$, що мають різні, такі, що не піддаються контролю зміщення своїх центрів симетрії мають стохастичний характер і настільки складне аналітичне представлення, що робить практично неможливим аналіз нелінійного перетворення (Б.2) з метою, наприклад, обґрунтованого вибору параметрів багат шарової нейронної мережі. Навчання мережі по алгоритму (Б.2) вносить динаміку в процес перетворення, оскільки алгоритми навчання відповідають нелінійним диференціальним або різницеvim рівнянням, де змінними являються коефіцієнти синаптичних вагів $\mathbf{w}_i^{(l)}$ і виходів $\mathbf{q}^{(l)}$ елементів в l -шарах. Така точка зору дозволяє навчати нейронну мережу еквівалентно представити на множині $\{\mathbf{w}, \mathbf{q}\}$ системою нелінійних диференціальних рівнянь або різницеvim рівнянь і структурами із зворотними зв'язками. Рішення цієї системи рівняннями руху об'єкту відкриває можливість описати динаміку процесів в нейронних мережах, спираючись на методи теорії нелінійних систем. З цією метою розглянемо алгоритми навчання багат шарових нейронних мереж.

ДОДАТОК В

НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

В.1 Загальна характеристика алгоритмів навчання

Під навчанням штучної нейронної мережі розуміється процес настройки вагових коефіцієнтів $w_{i,j}$ її базових процесорних елементів, результатом чого є виконання мережею конкретних завдань – розпізнавання, оптимізації, апроксимації, управління. Досягнення подібної мети формалізується критерієм якості Q , мінімальне значення $\min_w Q = Q^*$ якого відповідає якнайкращому рішення поставленої задачі.

Різноманіття алгоритмів навчання визначається функціональним призначенням мережі, її архітектурою і вибраною стратегією навчання. Розрізняють три основні стратегії навчання: 1) «з вчителем», 2) «без вчителя», тобто з самонавчанням і 3) змішану. У першому випадку нейронна мережа настраюється по заданій навчальній вибірці відповідно до прийнятого правила або алгоритму. У другому випадку наперед не потрібно знати правильний результат навчання і в процесі настройки вагових коефіцієнтів утворюється внутрішня структура активованих базових елементів, відповідна пред'явленому вектору входу мережі. При змішаній стратегії навчання частина вагових коефіцієнтів $w_{i,j}$ настраюється по заданій навчальній вибірці, а інша – відповідно до правил навчання «без вчителя».

Перерахованим стратегіям відповідають конкретні алгоритми навчання штучних нейронних мереж. Кожен алгоритм орієнтований на конкретну архітектуру мережі і застосовується для вирішення певного типу завдань. Так, одно- і багатошарові мережі навчаються по першій стратегії, і це застосовно для вирішення завдань класифікації, апроксимації і управління. Самонавчання використовується в мережах Хопфілда, картах (мережах) Кохонена, що самоорганізо-

вуються, мережах ART (adaptive resonance theory); змішана стратегія навчання застосовується в RBF-мережах з радіально-базисними функціями активації.

Загальним для всіх модифікацій алгоритмів є їх рекурентний характер, а самі алгоритми навчання є лінійними або частіше нелінійні диференціальні (або різницеві) рівняння першого порядку, складені за певним правилом. Ці правила мають евристичний характер, не завжди витікаючи з принципів функціонування біонейронів і нервової мережі живого організму.

Найбільшого поширення набули наступні правила навчання штучних нейронних мереж: правило Хебба; δ -правило; навчання з «конкуренцією» або методом змагання; ART-правило (навчання «без вчителя»); «больцманове» навчання; генетичні алгоритми. У даному розділі стисло опишемо перші два правила. Методу ВР – як базовому методу навчання багат шарових нейронних мереж – буде присвячений окремий розділ.

Правило Хебба – історично перша запропонована концепція навчання нейронних мереж «без вчителя». Відповідно до неї вагові коефіцієнти $w_{i,j}$ синаптичних зв'язків збільшуються, якщо активовано (збуджені) обидва базові елементи – джерело збудження і приймач сигналу збудження. Тим самим регулярно використовувані входи і синаптичні зв'язки активуються за рахунок збільшення «своїх» вагових коефіцієнтів більшою мірою, ніж решта нейронів мережі. Модель навчання Хебба дає пояснення здатності нейронної мережі до звикання і навчання цієї здатності через повторення. Правило або алгоритм Хебба записується таким чином:

$$w_{i,j}(k+1) = w_{i,j}(k) + \gamma q_j q_i, \quad (B.1)$$

де $w_{i,j}(k)$ – значення вагового коефіцієнта, що змінюється від нейрона i до нейрона j до настройки;

$w_{i,j}(k+1)$ – те ж, але після настройки;

q_i, q_j – виходи нейронів i, j відповідно;

$\gamma > 0$ – коефіцієнт «посилення» алгоритму;

$$k = 0, 1, 2, \dots$$

Правило Хебба може бути застосовано для навчання простих мереж типу мадаліни, рекурентних мереж Хопфілда, воно поклало початок процесу «конструювання» безлічі подібних правил навчання нейронних мереж «без вчителя» і, зокрема, на цьому шляху був отриманий вдалий алгоритм навчання «з конкуренцією» для мережі Кохонена (правило Кохонена). Важливою особливістю правила Хебба є те, що зміни синаптичних вагів $w_{i,j}^{(K)}$ залежать тільки від активності базових елементів, зв'язаних між собою даним синапсом (δ -правило реалізує стратегію навчання «з вчителем» або супервізорного навчання. «Вчитель» – це цільовий, бажаний вихід базового елемента типу персептрона u^* , а q – реальний вихід. Для настройки вагових коефіцієнтів базових елементів використовується помилка навчання $\delta = u^* - q$ в алгоритмі градієнтного спуску:

$$w_i(k+1) = w_i(k) - \gamma \delta r_i \quad \gamma > 0. \quad (B.2)$$

де r_i – вхід i -го синаптичного зв'язку базового елемента. Узагальнення δ -правила на багатошарові нейронні мережі приводить до методу зворотного розповсюдження помилки – методу ВР.

Навчання «конкуренцією» на відміну від правила Хебба, по якому може збуджуватися одночасно безліч базових елементів вихідного шару, передбачає «змагання» цих елементів за право активації. Така стратегія отримала назву «переможець бере все». Після активації в процесі настройки змінюються елементи, що лише «перемогли». Відомо, що така ж стратегія навчання властива біологічним нейронним мережам на відміну від навчання по δ -правилу. Навчання «з конкуренцією» використовується в ART-мережах, мережах Кохонена для вирішення задач класифікації зорових, звукових образів, стиснення даних, кластеризації і категоризації усередині класу.

Розглянемо базові алгоритми навчання багатошарових нейронних мереж, що набули переважного поширення при використанні штучних нейронних мереж в задачах управління.

В.2 Метод і алгоритм зворотного розповсюдження помилки

Метод зворотного розповсюдження помилки – метод ВР розроблений для навчання статичних нелінійних багатошарових нейронних мереж і використовує технологію послідовної і пошарової настройки базових процесорних елементів, починаючи з останнього, вихідного шару і закінчуючи настройкою елементів першого шару. «Урок» навчання штучної нейронної мережі може бути повторений необхідне число разів. Для настройки вагових коефіцієнтів базових елементів застосовується δ -правило в його загальній формі: у вигляді алгоритму градієнтного методу мінімізації критерію навчання $Q(\mathbf{e}_u)$. Тут $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)}$ – вектор помилки навчання мережі щодо бажаного (еталонного) виходу $\mathbf{u}^*$ при нелінійному перетворенні ввідного вектора $\mathbf{r}$ у вихідний вектор $\mathbf{u} = \mathbf{q}^{(K)}$ останнього шару K . Помилка $\mathbf{e}_u$, явним чином залежить від коефіцієнтів $w_{i,j}^{(K)}$ і може бути використана як аргумент функціонала Q для настройки базових елементів K -го шару. Явної залежності вектора $\mathbf{e}_u$ і функції $Q(\mathbf{e}_u)$ від вагових коефіцієнтів $w_{i,j}^{(l)}$ ($l = \overline{1, K-1}$) немає. Тому у разі використання критерію навчання $Q(\mathbf{e}_u)$ для настройки коефіцієнтів $w_{i,j}^{(l)}$ помилка $\mathbf{e}_u$ в процесі навчання багатошарової нейронної мережі перераховується послідовно в узагальнені помилки $\boldsymbol{\sigma}^{(l)}$, що явно залежать від значень $w_{i,j}^{(l)}$. Настройка коефіцієнтів $w_{i,j}^{(l)}$ здійснюється за δ -правилом, де використовуються узагальнені помилки $\boldsymbol{\sigma}^{(l)}$. Очевидно, що для шару K помилка $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$. У методі ВР далі використовується локальний квадратичний функціонал $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$. Мінімізація $Q(\boldsymbol{\sigma})$ по вагових коефіцієнтах $w_{i,j}^{(l)}$ ($l = K-1, \dots, 1$), що настроюються, методом градієнтного спуску з перерахунком помилки $\boldsymbol{\sigma}^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$ для шару K в помилку $\boldsymbol{\sigma}^{(l)} = \mathbf{e}(\mathbf{w}^{(l)})$ для шарів $l = K-1, K-2, \dots, 1$ визначає сутність ВР-технології навчання багатошарових нейронних мереж.

В.2.1. Налаштування коефіцієнтів вихідного шару

Для шару K процедура налаштування вектора всіх вагових коефіцієнтів в K -м шарі $\mathbf{w}^{(k)}$ відповідає зазвичай алгоритму градієнтного спуску:

$$\mathbf{w}^{(k)}(k+1) = \mathbf{w}^{(K)} - \gamma \nabla_{\mathbf{w}} Q(\mathbf{e}_u(\mathbf{w}^{(k)}(k))) , \quad (\text{B.3})$$

де:

$$\nabla_{\mathbf{w}} Q(\cdot) = \text{col} \left(\frac{\partial Q(\cdot)}{\partial \mathbf{w}_1^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_j^{(K)}}, \dots, \frac{\partial Q(\cdot)}{\partial \mathbf{w}_{n_k}^{(K)}} \right) -$$

вектор розмірністю $(n_k \times 1)$. Вектор $\mathbf{w}^{(k)}$ – складний:

$$\mathbf{w}^{(K)} = \text{col}(\mathbf{w}_1^{(K)}, \dots, \mathbf{w}_i^{(K)}, \dots, \mathbf{w}_{n_K}^{(K)}),$$

де:

$$\mathbf{w}^{(K)} = \text{col}(w_{i,1}^{(K)}, \dots, w_{i,m}^{(K)}, \dots, w_{n_{K-1}}^{(K)})$$

$\mathbf{q}^{(K-1)}(k)$ – вектор розмірністю $((n_{K-1} + 1) \times 1)$ базових елементів в шарі K :

$$\mathbf{q}^{(K-1)}(k) = \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}).$$

Сигнал ініціалізації $q_0^{(K-1)}$ зазвичай приймається рівним +1,

Градієнт

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = \mathbf{e}_n^T \frac{\partial \mathbf{e}_u}{\partial \mathbf{w}} = -\mathbf{e}_u^T \frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}},$$

де похідна:

$$\frac{\partial \mathbf{q}^{(K)}}{\partial \mathbf{w}^{(K)}} = \frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{w}^{(K)}} = \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right)$$

є добуток матриць Якобі. перша з них, така, що розкривається в наступному вигляді:

$$\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} = \begin{bmatrix} \frac{\partial}{\partial \mathbf{w}_1^{(K)}} \\ \frac{\partial}{\partial \mathbf{w}_2^{(K)}} \\ \dots \\ \dots \\ \frac{\partial}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} \times [s_1^{(K)}, s_2^{(K)}, \dots, s_{n_K}^{(K)}] = \begin{bmatrix} \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_1^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_1^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_i^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_i^{(K)}} \\ \dots & \dots & \dots \\ \frac{\partial s_1^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} & \dots & \frac{\partial s_{n_K}^{(K)}}{\partial \mathbf{w}_{n_K}^{(K)}} \end{bmatrix} = \text{diag} \left(\frac{\partial s_i^{(K)}}{\partial \mathbf{w}_i^{(K)}} \right) -$$

квадратна матриця, кожен елемент якої на діагоналі є вектор-стовпець розмірністю $[(n_{K-1} + 1) \times 1]$, отже повна розмірність матриці рівна $(n_K(n_{K-1} + 1) \times n_K)$. В матриці $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ всі елементи, окрім діагональних, рівні нулю, а приватні похідні $\partial s_i^{(K)} / \partial \mathbf{w}_i^{(K)}$ на діагоналі – всі однакові і рівно вектору $\mathbf{q}^{(K-1)}$. Це витікає з визначення дискримінантної функції $s_i^{(K)}$

$$s_i^{(K)} = \sum_{j=0}^{n_{K-1}} w_{i,j}^{(K)} q_j^{(K-1)} = \mathbf{w}_i^{(K)} \mathbf{q}^{(K-1)}.$$

Інша матриця Якобі розкривається у вигляді діагональної матриці $(n_K \times n_K)$

$$\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} = \text{diag}(f'_s(\mathbf{s}^{(K)})) = \begin{bmatrix} f'_s(\mathbf{s}_1^{(K)}) & 0 & 0 \\ \dots & \dots & \dots \\ 0 & 0 & f'_s(\mathbf{s}_{n_K}^{(K)}) \end{bmatrix}.$$

З урахуванням приведених матриць Якобі градієнт функціонала настройки K -го шару багат шарової нейронної мережі можна записати таким чином:

$$\nabla_{\mathbf{w}} Q(\mathbf{e}_u) = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \right) \left(\frac{\partial \mathbf{f}(\mathbf{s}^{(K)})}{\partial \mathbf{s}^{(K)}} \right) \mathbf{e}_u = - \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}} \text{diag}(f'_s(\mathbf{s}^{(K)})) \right) \mathbf{e}_u,$$

а алгоритм настройки вагових коефіцієнтів K -го шару має вигляд

$$\mathbf{w}^{(K)}(k+1) = \mathbf{w}^{(K)}(k) + \gamma \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}(k)} \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) \right) \mathbf{e}_u(k+1) . \quad (\text{B.4})$$

Рекурентний алгоритм (B.4) можна розділити на n_K алгоритмів навчання i -х базових елементів, $i = \overline{1, n_K}$, оскільки діагональна матриця $\partial \mathbf{s}^{(K)} / \partial \mathbf{w}^{(K)}$ складається з однакових елементів $\mathbf{q}^{(K-1)}$.

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i(k+1) \mathbf{q}^{(K-1)}(k+1). \end{aligned} \quad (\text{B.5})$$

Узагальнена помилка σ , навчання мережі для шару K рівна $f'_s(s_i^{(K)})e_{ui}$. У багатошарових мережах останній вихідний шар виконується зазвичай у вигляді мадаліни з лінійними функціями активації і $f'_s(s) = 1$. Тоді $\sigma_i = e_{ui}$ і алгоритм настройки вагових коефіцієнтів шаруючи K приймає вигляд:

$$\mathbf{w}_i^{(K)}(k+1) = \mathbf{w}_i^{(K)}(k) + \gamma e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) \quad (\text{B.6})$$

У алгоритмах (B.4)-(B.6) $k = 1, 2, \dots$ – номер інтеграції. Початкові значення вагових коефіцієнтів $\mathbf{w}^{(K)}(0)$ задаються довільно, але бажано в околиці значень $\mathbf{w}^{*(K)}$, для яких $Q(\mathbf{w}_{(K)}^*) = \min_{\mathbf{w}} Q(\mathbf{w}^{(K)})$.

В.2.2. Налаштування вагових коефіцієнтів прихованих шарів нейронної мережі.

Для прихованих шарів $l = \overline{K-1, 1}$ помилки навчання в явному вигляді не можуть бути визначені. Проте очевидно, що якщо $\mathbf{e}_u = \mathbf{u}^* - \mathbf{u} = \mathbf{u}^* - \mathbf{q}^{(K)} \neq 0$, то через архітектуру мережі існує ненульова узагальнена помилка і для всіх попередніх шарів. Суть методу ВР якраз і полягає в послідовному визначенні «прихованих» помилок, починаючи з обчислення помилки $\sigma^{(K)}$ у вихідному шарі K , потім помилки $\sigma^{(K-1)}$ в шарі $K-1$ і так до вхідного шару 1. Всі приховані по-

милки $\sigma^{(l)} = \sigma(\mathbf{w}_i^{(l)})$ залежать явно від вагових коефіцієнтів, що настраюються $\mathbf{w}_i^{(l)}$. Розглянемо суть необхідних перетворень в δ -правилі, яке використовує ідею обчислення прихованих помилок в рекурентній процедурі настройки вагових коефіцієнтів в шарах $l = \overline{K-1, 1}$. З цією метою складемо алгоритм настройки вагових коефіцієнтів для окремих i -х базових елементів в шарі l :

$$\mathbf{w}_i^{(l)} = \text{col}(w_{i,0}^{(l)}, \dots, w_{i,j}^{(l)}, \dots, w_{i,n_l}^{(l)})$$

При виведенні алгоритму настройки вагових коефіцієнтів в шарах $K-1, K-2, \dots, 1$ вкажемо, що в загальному випадку градієнт функціонала Q щодо вектора вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ можна обчислити як:

$$\nabla \mathbf{w}_i^{(l)} Q = \frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \frac{\partial Q}{\partial \mathbf{q}^{(l)}}.$$

Тут, як і для шару K , вважаємо для спільності і для спрощення подальших перетворень вектор $\mathbf{q}^{(l)} = \text{col}(q_0^{(l)}, q_1^{(l)}, \dots, q_{n_l}^{(l)})$

У практичних застосуваннях $q_0^{(l)} = +1$. У подальшому виводі вважатимемо, що ініціалізуючі сигнали $q_0^{(l)}$ є виходами фіктивних, «нульових» базових елементів з функціями активації $f(s_0^{(l)})$ того ж типу, що і в реальних елементах. Тоді випадку $q_0^{(l)} = +1$ відповідає робоча точка $f(s_0^{(l)}) = 1$.

З урахуванням зробленого припущення розмірність матриці Якобі $\partial \mathbf{s}^{(l)} / \partial \mathbf{w}_i^{(l)}$, де $\mathbf{s}^{(l)} = \text{col}(s_0^{(l)}, s_1^{(l)}, \dots, s_{n_l}^{(l)})$, рівна $(n_{l-1} + 1) \times (n_l + 1)$, відповідно розмірність матриці Якобі $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)} = \dim \text{diag}(f'_s(s_0^{(l)})) = (n_l + 1) \times (n_l + 1)$. Якщо покласти $f'_s(s_0^{(l)}) = 1$, то верхній діагональний елемент матриці $\partial \mathbf{q}^{(l)} / \partial \mathbf{s}^{(l)}$ буде рівний нулю. Отже, матриці

$$\left(\frac{\partial \mathbf{s}^{(l)}}{\partial \mathbf{w}_i^{(l)}} \right) = \left\| \frac{\partial s_i^{(l)}}{\partial w_{i,j}^{(l)}} \right\|, \quad i = \overline{0, n_l}, \quad j = \overline{0, n_{l-1}};$$

$$\left(\frac{\partial \mathbf{q}^{(l)}}{\partial \mathbf{s}^{(l)}} \right) = \text{diag}(f'_s(s^{(l)}))$$

складені. Позначимо першу з них символом $\mathbf{S}^{(l)}$, другу $\mathbf{F}^{(l)}$. Елементами матриці служать приватні похідні $\partial s_i^{(l)} / \partial w_{i,j}^{(l)}$. Оскільки

$$s_i^{(l)} = \sum_{j=0}^{n_{l-1}} w_{i,j}^{(l)} q_j^{(l-1)},$$

матриця $\mathbf{S}^{(l)}$ має наступну структуру:

$$\mathbf{S}^{(l)} = \begin{bmatrix} q_0^{(l-1)} & q_0^{(l-1)} & \dots & q_0^{(l-1)} \\ q_1^{(l-1)} & q_1^{(l-1)} & \dots & q_1^{(l-1)} \\ \dots & \dots & \dots & \dots \\ q_{n_{j-1}}^{(l-1)} & q_{n_{j-1}}^{(l-1)} & \dots & q_{n_{j-1}}^{(l-1)} \end{bmatrix} = [\mathbf{q}^{(l-1)} : \mathbf{q}^{(l-1)} : \dots : \mathbf{q}^{(l-1)}],$$

розмірність $\mathbf{S}^{(l)}$ рівна $(n_{l-1} + 1) \times (n_{l-1} + 1)$ і $\mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, \dots, q_{n_{l-1}}^{(l-1)})$.

Елементами діагональної $\mathbf{F}^{(l)}$ матриці є приватні похідні функцій активації $f(s)$ по своєму аргументу $s_i (i = \overline{0, n_l})$. Можна показати, що добуток матриць $\mathbf{S}^{(l)}$ і $\mathbf{F}^{(l)}$ зводиться до зовнішнього добутку векторів $\mathbf{q}^{(l-1)}$ і $\mathbf{f}'_s(s^{(l)}) = (f'_s(s_0^{(l)}), \dots, f'_s(s_{n_l}^{(l)}))$.

$$\mathbf{S}^{(l)} \mathbf{F}^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(s^{(l)}))^T.$$

Похідна скалярного функціонала $Q(\boldsymbol{\sigma})$ по векторному аргументу $\mathbf{q}^{(l)} (l = \overline{K-1, 1})$ по своєму сенсу є вектор чутливості функціонала Q до зміни вектора виходу $\mathbf{q}^{(l)}$ l -го шару мережі. Для шару l введемо вектор чутливості функціонала Q :

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \text{col} \left(\frac{\partial Q}{\partial q_0^{(l)}}, \frac{\partial Q}{\partial q_1^{(l)}}, \dots, \frac{\partial Q}{\partial q_{n_l}^{(l)}} \right) = \mathbf{col}(\lambda_0^{(l)}, \lambda_1^{(l)}, \dots, \lambda_{n_l}^{(l)}).$$

Приватні функції чутливості $\lambda_i^{(l)}$ обчислюються, по-перше, відповідно до правила диференціювання складних функцій; по-друге, відповідно до структури багат шарової нейронної мережі і мають вигляд:

$$\lambda_i^{(l)} = \frac{\partial Q}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \frac{\partial Q}{\partial q_j^{(l+1)}} \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \sum_{j=1}^{n_{l+1}} \lambda_j^{(l+1)} d_{j,i}^{(l+1)} = (\Lambda^{(l+1)})^T \mathbf{d}_i^{(l+1)},$$

$$i = \overline{1, n_l}.$$

Повний вектор функцій чутливості функціонала Q до зміни вектора виходу нейронів l -го шару є

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \mathbf{D}^{(l+1)} \Lambda^{(l+1)}, \quad l = \overline{K-1, 1}. \quad (\text{B.7})$$

Матрицю $\mathbf{D}^{(l+1)}$ назовемо матрицею переходу у зворотному напрямі, оскільки для розрахунку функції $\Lambda^{(l)}$ використовуються функції чутливості $\Lambda^{(l+1)}$ і $\mathbf{D}^{(l+1)}$ подальшого шару. Матриця $\mathbf{D}^{(l+1)}$ складена з вектор-рядків $(\mathbf{d}_i^{(l+1)})^T$ де $i = \overline{0, n_l}$. Нагадаємо, що за умовчанням $q_0^{(l)} = +1$, а вагові коефіцієнти ініціалізуючих входів $w_{j,0}^{(l+1)}$ всіх базових елементів в шарі $l+1$ також настроюються. Елементами векторів за визначенням є приватні похідні:

$$\frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} = \frac{\partial}{\partial q_i^{(l)}} f \left(\sum_{j=0}^{n_l} w_{j,i}^{(l+1)} q_j^{(l)} \right) = \left(\frac{\partial f(s_j^{(l+1)})}{\partial s_j^{(l+1)}} \right) \left(\frac{\partial s_j^{(l+1)}}{\partial q_i^{(l)}} \right) = f'_s(s_j^{(l+1)}) w_{j,i}^{(l+1)}.$$

Для всієї множини індексів $i = \overline{0, n_l}, j = \overline{1, n_{l+1}}$ матриця переходу $\mathbf{D}^{(l+1)}$ має наступну структуру:

$$\mathbf{D}^{(l+1)} = \frac{\partial \mathbf{q}^{(l+1)}}{\partial \mathbf{q}^{(l)}} = \begin{bmatrix} \frac{\partial}{\partial q_0^{(l)}} \\ \frac{\partial}{\partial q_1^{(l)}} \\ \dots \\ \frac{\partial}{\partial q_{n_l}^{(l)}} \end{bmatrix} \times (q_0^{(l+1)}, q_1^{(l+1)}, \dots, q_{n_{l+1}}^{(l+1)}) = \begin{bmatrix} \frac{\partial q_0^{(l+1)}}{\partial q_0^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_0^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_0^{(l)}} \\ \frac{\partial q_0^{(l+1)}}{\partial q_1^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_1^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_1^{(l)}} \\ \dots & \dots & \dots & \dots \\ \frac{\partial q_0^{(l+1)}}{\partial q_{n_l}^{(l)}} & \frac{\partial q_1^{(l+1)}}{\partial q_{n_l}^{(l)}} & \dots & \frac{\partial q_{n_{l+1}}^{(l+1)}}{\partial q_{n_l}^{(l)}} \end{bmatrix}.$$

Відзначимо, що розмірність матриці переходу $\mathbf{D}^{(l+1)}$ рівна $[(n_{l+1} + 1) \times (n_l + 1)]$. Приватні похідні в матриці $\mathbf{D}^{(l+1)}$ визначені раніше. Тепер можна скласти вектор-рядки $(\mathbf{d}_i^{(l+1)})^T$:

$$(\mathbf{d}_i^{(l+1)})^T = (f'_s(s_0^{(l+1)})w_{0,i}^{(l+1)}, f'_s(s_1^{(l+1)})w_{1,i}^{(l+1)}, \dots, f'_s(s_{n_{l+1}}^{(l+1)})w_{n_{l+1},i}^{(l+1)})$$

розмірністю $[1 \times (n_{l+1} + 1)]$. Введемо матриці:

а)

$$\mathbf{W}^{(l+1)} = [\mathbf{w}_0^{(l+1)} : \mathbf{w}_1^{(l+1)} : \dots : \mathbf{w}_{n_{l+1}}^{(l+1)}] = \|\mathbf{w}_{i,j}^{(l)}\|, \quad i = \overline{0, n_{l+1}}; \quad j = \overline{0, n_l},$$

тобто

$$\dim \mathbf{W}^{(l+1)} = (n_l + 1) \times (n_{l+1} + 1),$$

де

$$\mathbf{w}_i^{(l+1)} = \text{col}(w_{i,0}^{(l+1)}, \dots, w_{i,j}^{(l+1)}, \dots, w_{i,n_l}^{(l+1)});$$

б)

$$\mathbf{F}^{(l+1)} = \text{diag}(f'_s(\mathbf{s}^{(l+1)})) = \begin{bmatrix} f'_s(s_0^{(l+1)}) & \dots & 0 & 0 \\ 0 & f'_s(s_1^{(l+1)}) & \dots & \dots \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & f'_s(s_{n_{j-1}}^{(l+1)}) \end{bmatrix}$$

$$\dim \mathbf{F}^{(l+1)} = (n_{l+1} + 1) \times (n_{l+1} + 1).$$

Відмітимо, що для фіктивного «нульового» базового елементу $q_0^{(l+1)} = +1$ і елемент матриці $\mathbf{F}^{(l+1)} f'_s(s_0^{(l+1)}) = 0$. З урахуванням введених матриць $\mathbf{W}^{(l+1)}$ і

$\mathbf{F}^{(l+1)}$ матрицю $\mathbf{D}^{(l+1)}$ запишемо у вигляді:

$$\mathbf{D}^{(l+1)} = \left\| \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} \right\|_{i=0, \overline{n_l}}^{j=0, \overline{n_{l+1}}} = (\mathbf{W}^{(l+1)})^T \mathbf{F}^{(l+1)} = (\mathbf{W}^{(l+1)})^T \text{diag} (f'_s(s^{(l+1)})). \quad (\text{B.8})$$

Тепер можна записати формулу обчислення градієнта функціонала навчання багатошарової нейронної мережі $Q(\boldsymbol{\sigma}) = 0,5 \boldsymbol{\sigma}^T \boldsymbol{\sigma}$ по вагових коефіцієнтах будь-якого шару $l = \overline{K-1, 1}$:

$$\nabla \mathbf{w}_i^{(l)} Q = \mathbf{S}^{(l)} \mathbf{F}^{(l)} \Delta^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T \Lambda^{(l)} \quad (\text{B.9})$$

Повний алгоритм навчання багатошарової нейронної мережі по методу ВР включає початковий алгоритм настройки вагових коефіцієнтів K -го шару (B.6) і алгоритми настройки базових елементів шарів $l = \overline{K-1, 1}$ використовують формулу (B.9).

В.2.3. Алгоритм навчання багатошарової нейронної мережі по методу зворотного розповсюдження помилки.

Приведемо алгоритм навчання нейронної мережі методом зворотного розповсюдження помилки в одній з можливих форм, витікаючої з прийнятої методики виводу алгоритму:

а)

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k+1)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i^{(K)}(k+1) \mathbf{q}^{(K-1)}(k+1); \\ \gamma &> 0; \quad k = 0, 1, 2, \dots; \quad e_{ui} = u_i^* - u_i; \quad i \in \overline{1, n_K}; \\ \mathbf{q}^{(K-1)} &= \text{col}(q_0^{(K-1)}, q_1^{(K-1)}, \dots, q_j^{(K-1)}, \dots, q_{n_{K-1}}^{(K-1)}); \end{aligned} \quad (\text{B.10})$$

б)

$$\begin{aligned}
\mathbf{w}_i^{(l)}(k+1) &= \mathbf{w}_i^{(l)}(k) - \gamma \mathbf{q}^{(l-1)}(k+1)(\mathbf{f}'_s(s_i^{(l)}(k+1)))^T \Lambda^{(l)}(k); \\
\Lambda^{(l)}(k) &= \mathbf{D}^{(l+1)} \Lambda^{(l+1)}(k); \\
\mathbf{D}^{(l+1)}(k) &= (\mathbf{W}^{(l+1)}(k))^T \text{diag}(f'_s(\mathbf{s}^{(l)}(k))); \\
\gamma > 0; \quad l &= \overline{K-1, 1}; \quad \mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, q_1^{(l-1)}, \dots, q_{l_{K-1}}^{(l-1)}); \quad i = \overline{0, n_l},
\end{aligned} \tag{B.11}$$

Зокрема, алгоритм настройки БПЕ шару $K-1$ у відповідності до пункту б) має вид

$$\begin{aligned}
\mathbf{w}_i^{(K-1)}(K+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \mathbf{q}^{(K-2)}(k+1)(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k), \\
\Lambda^{(K-1)}(k) &= \mathbf{D}^{(K)}(k) \Lambda^{(K)}(k); \\
\mathbf{D}^{(K)}(k) &= \mathbf{W}^{(K)}(k) \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) = \mathbf{W}^{(K)}(k); \\
\Lambda^{(K)}(k) &= \text{col} \left(\frac{\partial Q}{\partial \mathbf{q}_1^{(K)}}, \dots, \frac{\partial Q}{\partial \mathbf{q}_{n_K}^{(K)}} \right) = -\mathbf{e}_u(k).
\end{aligned}$$

тоді

$$\begin{aligned}
\Lambda^{(K-1)}(k) &= -\mathbf{W}^{(K)}(k) \mathbf{e}_u(k), \\
(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k) &= \\
&= -(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \mathbf{W}^{(K)}(k) \mathbf{e}_u(k) = \sigma_1^{(K-1)}(k+1)
\end{aligned}$$

– скалярна узагальнена помилка навчання нейронної мережі для i -го базового елементу в шарі $K-1$. З її використанням отримуємо алгоритм настройки, формою в точності відповідний δ -правилу:

$$\begin{aligned}
\mathbf{w}_i^{(K-1)}(k+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \sigma_i^{(K-1)}(k) \mathbf{q}^{(K-2)}(k+1); \\
\sigma_i^{(K-1)}(k+1) &= -(\mathbf{f}'_s(s_i^{(K-1)}(k)))^T \mathbf{W}^{(K)}(k) \mathbf{e}^{(K)}(k+1); \\
\mathbf{e}_u^{(K)}(k) &= \mathbf{u}^*(k) - \mathbf{q}^{(K)}(k).
\end{aligned} \tag{B.12}$$

Аргумент k позначає момент використання відповідних значень заданих множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$ на k -й ітерації обчислень вагових коефіцієнтів $\mathbf{w}_i^{(l)}$ на

«уроках» навчання, число яких залежить від числа шарів мережі і кількості нейронів в шарах.

Фаза навчання завершена, коли досягнутий для заданих в своєму класі множин $\mathbf{u}^*$ і $\mathbf{r} = \mathbf{q}^{(0)} = \mathbf{r}$. Мережа фіксує цей стан в значеннях своїх вагових коефіцієнтів і відповідних «віртуальних» зсувах центрів симетрії функцій активації і їх похідних активованих базових елементів і зберігає цей стан після зняття сигналу $\mathbf{q}^{(0)} = \mathbf{r}$ на входах мережі а виді віртуальної статичної нелінійної характеристики, зібраної з фрагментів сигмоїдних функцій. Якщо тепер па входах пред'явити ту ж множину або близьку до неї з того ж класу, то нейронна мережа практично миттєво відновить на виході ту ж множину $\mathbf{u} = \mathbf{q}^{(K)}$, що і після фази навчання. Цей етап функціонування мережі називають фазою спогаду. Мережа, таким чином, виконує функцію асоціативної пам'яті. Якщо ж множини $\mathbf{u}^*$, $\mathbf{q}^{(0)} = \mathbf{r}$ є функціями часу: $\mathbf{q}^{(0)} = \mathbf{r}(t)$, $\mathbf{u}^* = \mathbf{u}^*(t)$, то стандартний алгоритм ВР стає неавтономним і вимагає завдання всієї множини значень змінних $\mathbf{r}(t)$ і $\mathbf{u}^*(t)$. Неминуче зростання числа шарів і базових елементів в них і пов'язане з цим збільшення тривалості навчання викликає необхідність забезпечення стійкості процесу настройки мережі по алгоритму (B.10) (B.11).

Очевидно, що в завданнях управління $\mathbf{u}^*(t)$ – функція оптимального управління динамічним об'єктом. Управління за допомогою багатошарової нейронної мережі в двохетапному режимі можливо, якщо тільки функція оптимального управління наперед розрахована і може бути визначена вся множина значень помилки навчання мережі $\mathbf{e}_u(t)$. на інтервалі її визначення. Використання іншої вимірювальної інформації, наприклад помилки відтворення заданої функції $\mathbf{r}(t)$ на виході динамічного об'єкту, вносить запізнювання до алгоритму настройки і приводить до нестійкого процесу. Виникає проблема синтезу динамічних алгоритмів навчання багатошарової нейронної мережі, що враховують вплив інерційності, що вноситься до процесу настройки по алгоритму ВР.

В.3 Перенавчання і узагальнення нейронних мереж

Одна з найбільш серйозних труднощів викладеного підходу полягає в тому, що таким чином ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати, – помилку, яку можна чекати від мережі, коли їй подаватимуться абсолютно нові спостереження. Інакше кажучи, ми хотіли б, щоб нейронна мережа володіла здатністю узагальнювати результат на нові спостереження. Насправді мережа навчається мінімізувати помилку на навчальній множині, і у відсутність ідеальної і нескінченно великої навчальної множини це зовсім не те ж саме, що мінімізувати «справжню» помилку на поверхні помилок в наперед невідомій моделі явища.

Найсильніше ця відмінність виявляється в проблемі перенавчання. При дуже близькій підгонці явище простіше буде продемонструвати не для нейронної мережі, а на прикладі апроксимації за допомогою поліномів, при цьому суть явища абсолютно та ж.

Поліном (або многочлен) – це вираз, що містить тільки константи і цілі ступені незалежної змінної. Графіки поліномів можуть мати різну форму. Чим вище ступінь многочлена (і, тим самим, ніж більше членів в нього входить), тим більше складною може бути ця форма. Якщо у нас є деякі дані, ми можемо поставити мету підігнати до них поліноміальну криву (модель) і отримати, таким чином, пояснення для наявної залежності. Проте наші дані можуть бути зашумлені, тому не можна вважати, що сама краща модель задається кривою, яка в точності проходить через все наявні крапки. Поліном низького порядку може бути недостатньо гнучким засобом для апроксимації даних, тоді як поліном високого порядку може виявитися занадто гнучким і буде точно слід даним, приймаючи при цьому хитромудру форму, що не має ніякого відношення до форми справжньої залежності.

Нейронна мережа стикається з точно такою ж трудностю. Мережі з великим числом вагів моделюють складніші функції і, отже, схильні до перенавчання. Мережа ж з невеликим числом вагів може опинитися недостатньо гнучкою,

щоб змодельовати наявну залежність. Наприклад, мережа без проміжних шарів насправді моделює звичайну лінійну функцію.

Як же вибрати «правильний» ступінь складності для мережі? Складніша мережа майже завжди дає меншу помилку, але це може свідчити не про хорошу якість моделі, а про перенавчання.

Відповідь полягає в тому, щоб використовувати механізм контрольної перехресної перевірки (крос-перевірки), при якому частина навчальних спостережень резервується і не використовується в процесі навчання по алгоритму зворотного розповсюдження. Натомість у міру роботи алгоритму вона використовується для незалежної перевірки результату. На самому початку роботи помилка мережі на навчальній і перевірочній множинах буде однаковою, але у міру того як мережа навчається, помилка навчання, природно, убиває, і, поки навчання зменшує дійсну функцію помилок, помилка на контрольній (перевірочній) множині також убиватиме. Якщо ж контрольна помилка перестала убивати або навіть стала рости, це указує на те, що мережа почала дуже близько апроксимувати дані і навчання слід зупинити.

Відмічене явище занадто точної апроксимації в процесі навчання і називається перенавчанням. Якщо таке трапляється, то рекомендується зменшити число прихованих елементів і/або шарів, бо мережа є дуже могутньою для даного завдання. Якщо ж мережа, навпаки, була узята недостатньо складною для того, щоб моделювати наявну залежність, то перенавчання, швидше за все, не відбудеться, і обидві помилки – навчання і перевірки – не досягнуть достатнього рівня малості.

Описані проблеми з локальними мінімумами і вибором розміру мережі призводять до того, що при практичній роботі з нейронними мережами, як правило, доводиться експериментувати з великим числом різних мереж, деколи навчаючи кожен з них по кілька разів (щоб не бути введеним в оману локальними мінімумами) і порівнюючи отримані результати. Головним показником якості результату тут є контрольна помилка. При цьому, відповідно до загальнонаукового принципу, згідно якому при інших рівних слід віддати перевагу простішій

моделі, з двох мереж з приблизно рівними помилками контролю має сенс вибрати ту, яка менше.

Необхідність багатократних експериментів веде до того, що перевірна множина починає грати ключову роль у виборі моделі, тобто стає частиною процесу навчання. Тим самим ослабляється її роль як незалежного критерію якості моделі – при великому числі експериментів є ризик вибрати «вдалу» мережу, що дає добрий результат на перевірочній (контрольній) множині.

Для того, щоб додати остаточній моделі належну надійність, часто (принаймні, коли об'єм навчальних даних це дозволяє) поступають так: резервують ще одну – тестову множину спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на навчальній і контрольній множинах, реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки один раз: якщо її використовувати повторно для коректування процесу навчання, то вона фактично перетвориться на контрольну множину.

ДОДАТОК Г

УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ

Г.1 Алгоритм нейроуправління з прогнозом

У роботі використовується ефективна реалізація узагальненого управління з прогнозом (Generalized Predictive Control, GPC) з використанням багатошарової прямонаправленої нейронної мережі, як нелінійній моделі об'єкту управління [37]. Завдяки використанню оптимізаційного алгоритму Ньютона-Рафсона, число ітерацій, необхідних для збіжності, значно менше, ніж при використанні інших методів. Головні витрати алгоритму Ньютона-Рафсона в обчисленні Гесиана, але навіть з цими додатковими розрахунками низьке число ітерацій робить алгоритм Ньютона-Рафсона швидшим, ніж інші методи, при цьому він може використовуватися для управління в режимі реального часу (real-time control).

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

Узагальнене управління з прогнозом, введене Кларком і його співробітниками в 1987, належить до класу методів цифрового управління, званих модель-орієнтоване управління з прогнозом (Model-Based Predictive Control, MBPC) [38-40]. MBPC техніки були успішно проаналізовані і впроваджені в процес управління виробництвом в кінці 1970-х і продовжують використовуватися, оскільки вони системно враховують обмеження реальних об'єктів в режимі реального часу. GPC застосовуються в управлінні немінімально фазовими об'єктами, нестійкими об'єктами і об'єктами із змінним або невідомим часом затримки. Узагальнене управління з прогнозом також є робастним по відношенню до помилок моделювання, перевизначення і недовизначення значень параметрів і шумів датчиків [38]. GPC було спочатку розроблене для моделей лінійних об'єктів, що приводить до рівнянь, які можуть бути вирішені аналіти-

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

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

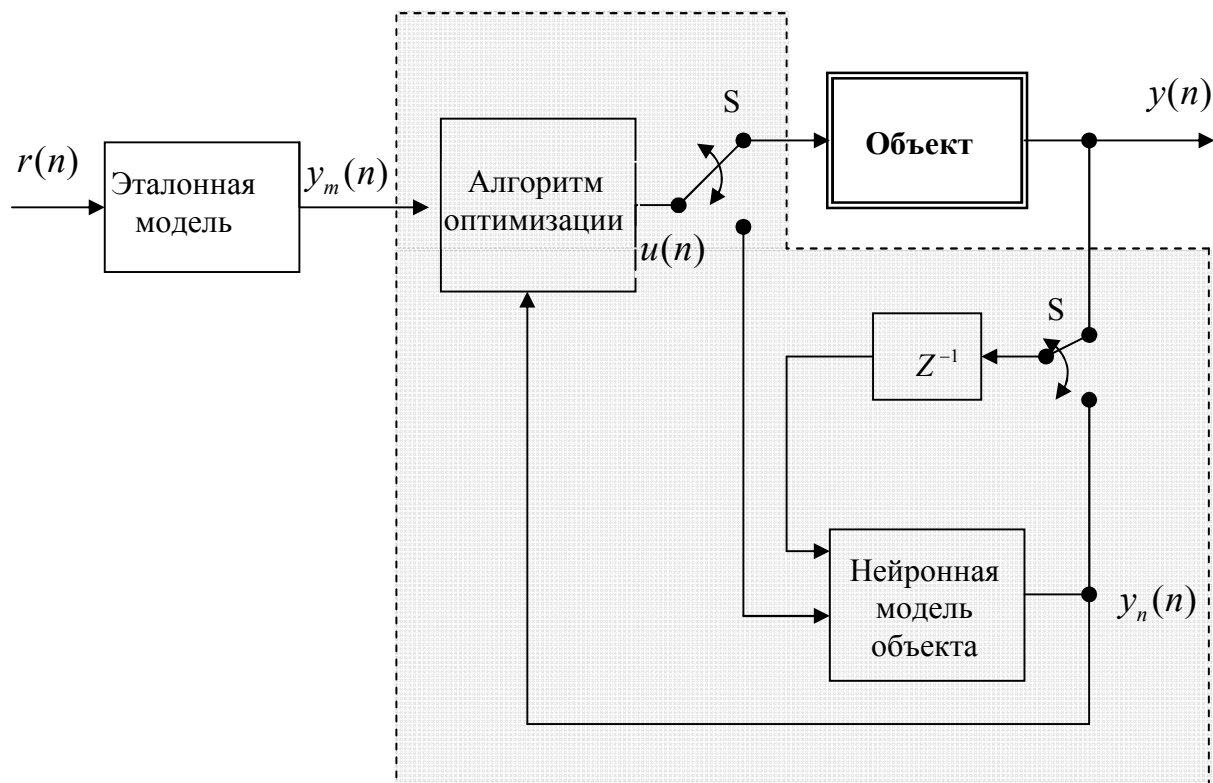


Рисунок Г.1 – Схема системи узагальненого нейроуправління з прогнозом

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

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

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

Узагальнений алгоритм нейроуправління з прогнозом складається з наступних основних кроків.

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

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

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

Якість обчислення при реалізації узагальненого управління з прогнозом основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient [41], Simplex [42] і Successive Quadratic Programming [43,44]. Вибір методу мінімізації може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу [43, 44]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

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

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

де N_1 – нижня межа прогнозу;

N_2 – верхня межа прогнозу;

N_u – діапазон управління;

y_m – бажана траєкторія;

y_n – передбачений вихід нейронної мережі;

λ – ваговий множник;

$\Delta u(n+j)$ – зміна u визначається таким чином:

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

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

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

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

кожної ітерації розраховується також значення управляючого вектора

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

де $k=1, \dots$ – номер ітерації.

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

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

де якобіан позначений, як

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

а гесіан, як

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

Рішення рівняння (8.2) вимагає явного звернення матриці Гессе. Цей процес може зажадати великих обчислювальних витрат. Одним з підходів, використовуваних, щоб уникнути звернення матриці, служить LU декомпозиція [45] для отримання вхідного вектора $\mathbf{U}(k+1)$. Це досягається шляхом запису рівняння (8.2) у вигляді системи лінійних рівнянь $\mathbf{Ax} = \mathbf{b}$, що приводить до виразу

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

де

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

У даній формі рівняння (8.3) може бути вирішене за допомогою двох процедур, описаних в [45]: процедури LU декомпозиції, ludcmp і процедури рішення системи лінійних рівнянь lubksb. Після того, як обчислений вектор $\mathbf{x}$. Процедура повторюється до тих пір, поки зміна кожній компоненти $\mathbf{U}(k+1)$ не стає меншим деякого ε . При рішенні системи відносно $\mathbf{x}$ для кожної ітерації алгоритму Ньютона-Рафсона необхідне обчислення кожного елементу якобіана і гессиана. Елемент якобіана з індексом h має наступний вигляд:

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

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

дельта функції, яка визначається як

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

отримаємо

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

Елемент з індексами m, h гессиана визначається таким чином:

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

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

Можна скористатися також визначенням дельта-функції для виразу

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

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

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

Г.2 Структура нейронної мережі для моделювання об'єкту

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

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

ким чином, що безліч входних значень продукує бажані вихідні значення. Помилка формується між відповідями нейронної мережі $y_n(n)$ і об'єкту $y(n)$. Надалі ця помилка використовується для оновлення вагів за допомогою одного з розглянутих вище методів, наприклад градієнтного спуску [46]. Цей процес повторюється до тих пір, поки помилка не досягне допустимого рівня.

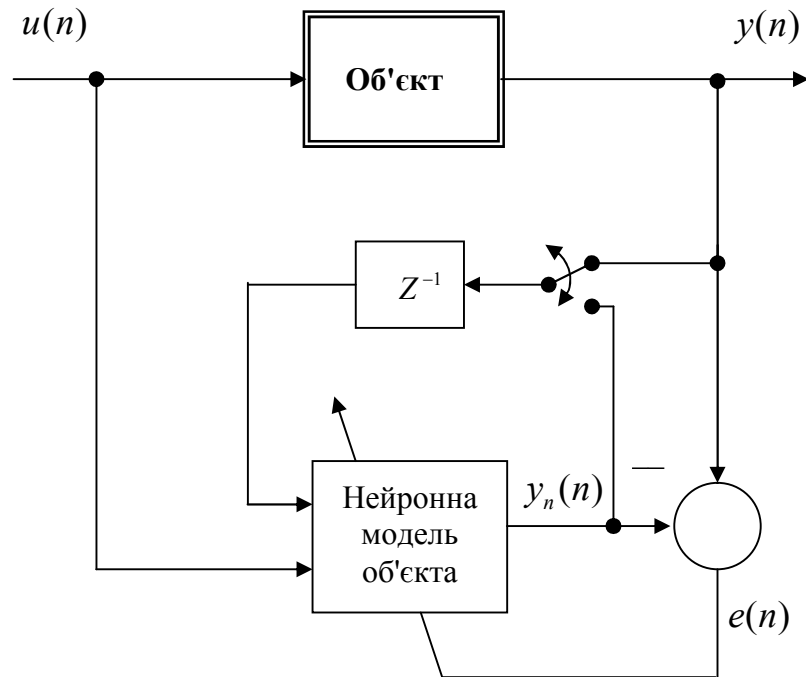


Рисунок Г.2 – Схема тренування мережі в режимі offline

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

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

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

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

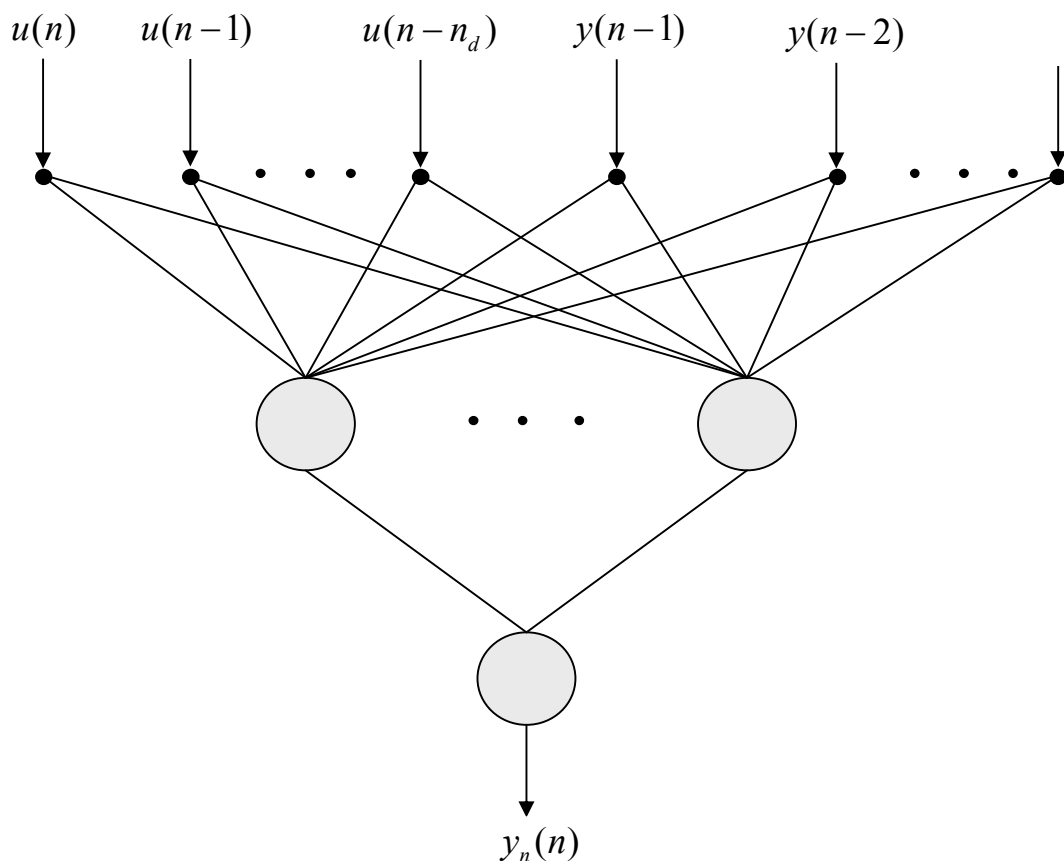


Рисунок Г.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

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

Рівняння мережі з даною архітектурою виглядають таким чином:

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

і

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

де $y_n(n)$ – вихід нейронної мережі;

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

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

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

L – число шарів;

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

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

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

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

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

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

b_j – зсув j -го прихованого нейрона;

b – зсув нейрона виходу.

Г.3 Прогноз з використанням нейронної мережі

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

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

і

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

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

Г.4 Похідні рівнянь нейронної мережі

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

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

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

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

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

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

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

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

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

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

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

і

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

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

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

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

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% розвороту роторного екскаватора
% 2024 рік

% Вихідні дані
ks=3.43;
Re=0.307;
Rd=2*0.073;
Rcp=0;
Rs=Rd+Rcp;
Ld=2*0.003;
Lcp=0;
Ls=Ld+Lcp;
kzn=0.71;
CF=2*3.42;
Tzc=kzn*Ls;
Ttp=0.005;
Tg=Ttp;
Te=0.03;
kzv=kzn*CF;
In=192;
Mn=CF*In;
Jd=2*2;
Jm=39.4;
b12=0;
c12=25;
Js=Jd+Jm;
Uz=440;
ksx=ks/(CF*(1+ks*kzn));
k1=1.0;
k2=1.6;
k3=3.0;

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