

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

НЕЙРОМЕРЕЖЕВІ СИСТЕМИ УПРАВЛІННЯ ОСНОВНИМИ
МЕХАНІЗМАМИ МОСТОВОГО КРАНА. ТЕМА КОМПЛЕКСНА.
СПЕЦІАЛЬНА ЧАСТИНА. СИНТЕЗ І ДОСЛІДЖЕННЯ СИСТЕМИ
УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА
З НЕЙРОРЕГУЛЯТОРОМ NN PREDICTIVE CONTROLLER
(тема кваліфікаційної роботи)

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

(код і найменування спеціальності)

_____ / Максим САМОЙЛЕНКО
(підпис) (ім'я та прізвище)

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

Рецензент _____ / Костянтин БРОВКО
(підпис) (ім'я та прізвище)

«До захисту допущено»

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

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

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

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

« » 2024р.

ЗАВДАННЯ

на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Максим САМОЙЛЕНКО
(ім'я, прізвище)

1. Тема роботи Нейромережеві системи управління основними механізмами мостового крана. Тема комплексна. Спеціальна частина. Синтез і дослідження системи управління механізмом переміщення візка з нейрорегулятором NN Predictive Controller
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 Вантажопідйомність крана $Q=41 \cdot 10^3$ кг; продуктивність крана $C=240$ кг/с; маса моста без візка $G_M=28,5 \cdot 10^3$ кг; маса візка $G_B=8,5 \cdot 10^3$ кг; діаметр коліс моста $D_M=0,56$ м; діаметр коліс візка $D_B=0,25$ м; швидкість пересування моста $V_M=0,85$ м/с; швидкість пересування візка $V_B=0,55$ м/с; швидкість підйому $V_{\Pi}=0,12$ м/с; маса вантажозахватного пристосування $q=1 \cdot 10^3$ кг; середня висота підйому і спуску $H_{\Pi}=8$ м; діаметр барабана механізму підйому $D=0,75$ м. Перерегулювання швидкості по задаючій дії більше 10%, час регулювання до 5 с.
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму підйому мостового крана. Розрахувати систему підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. Розробити математичну модель двомасової системи. Виконати моделювання на ЕОМ та провести аналіз показників якості перехідних процесів змінних стану двомасової системи. Розробити структурну схему нейромережевої системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання системи з нейрорегулятором.
5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістрська кваліфікаційна робота. Формальні ознаки. 2. Механізм переміщення візка. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 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. Магістрська 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)

Геннадій КАНЮК
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Максим САМОЙЛЕНКО
(ім'я, прізвище)

**КАЛЕНДАРНИЙ ПЛАН-ГРАФІК
виконання кваліфікаційної роботи**

№ з/п	Назва етапів роботи та питань, які мають бути розроблені відповідно до завдання	Термін виконання	Позначки керівника про виконання завдань
1	Обґрунтування застосування нейромережових технологій для управління багатомасовою електро-механічною системою механізму переміщення візка мостового крана.	09.09.2024-18.09.2024	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження динамічних характеристик системи.	19.09.2024-15.10.2024	
3	Дослідження системи з урахуванням поперечних розгойдувань вантажу. 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)

Максим САМОЙЛЕНКО
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 205 сторінок, 61 рисунків, 2 таблиці, 79 використаних джерел, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму переміщення візка мостового крана 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 переміщення візка мостового крана. 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 «УІПА» ХНУ ім. В.Н. Каразіна при читанні лекцій, виконанні курсового проєкту, проведенні практичних занять.

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

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

ABSTRACT

An explanatory message consists of: 205 pages, 61 figures, 2 tables, 79 used sources, 5 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the moving truck mechanism of the bridge crane 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 moving truck mechanism of the bridge crane. 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 neural network control system are considered and made a synthesis of neuroregulator NN Predictive Controller in 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.). According to research published two works printed in professional journals.

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

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ	
УПРАВЛІННЯ. ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ.....	14
1.1 Розвиток теорії штучних нейронних мереж.....	14
1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж	17
1.3 Нейрокерування динамічними об'єктами	22
1.4 Огляд структур нейромережевих систем управління. Вибір структури системи	25
1.5 Основні характеристики мостових кранів.....	32
1.6 Обґрунтування вибору системи електроприводу механізму переміщення моста мостового крана.....	34
1.7 Постановка задачі синтезу нейромережевої системи управління	36
ВИСНОВКИ ДО РОЗДІЛУ 1	39
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ	
МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА МОСТОВОГО КРАНА	41
2.1 Технічні характеристики мостового крана. Вимоги до системи управління.....	41
2.2 Розрахунок потужності електроприводу механізму пересування візка. Вибір приводного двигуна.....	42
2.3 Вибір основного електроустаткування і розрахунок параметрів силового кола електроприводу	46
2.4 Вибір структури системи автоматичного управління.....	51
2.5 Синтез послідовного корегуючого пристрою контура струму	52
2.6 Синтез послідовного корегуючого пристрою контура ЕРС.....	59
2.7 Дослідження динамічних характеристик системи	66
ВИСНОВКИ ДО РОЗДІЛУ 2	71

РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ	
ПОПЕРЕЧНИХ РОЗГОЙДУВАНЬ ВАНТАЖУ	73
3.1 Розрахункові схеми механічної частини електроприводу.....	73
3.2 Математична модель двомасової системи	75
3.3 Розрахунок перехідних процесів двомасової систем на ЕОМ	83
ВИСНОВКИ ДО РОЗДІЛУ 3	89
РОЗДІЛ 4. СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	90
4.1 Регулятори, реалізовані в пакеті прикладних програм	
Neural Network Toolbox системи MATLAB	90
4.2 Принцип побудови регулятора з прогнозом	
NN Predictive Controller	92
4.3 Синтез регулятора з прогнозом NN Predictive Controller	
з використанням системи MATLAB	95
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	138
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором	
NN Predictive Controller	140
ВИСНОВКИ ДО РОЗДІЛУ 4	146
ВИСНОВКИ	148
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	150
ДОДАТОК А. БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ.....	158
А.1 Класифікація штучних нейронних мереж	158
А.2 Базовий процесорний елемент (штучний нейрон).....	160
А.3 Статичні багатошарові нейронні мережі	162
ДОДАТОК Б. ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ	
МЕРЕЖ.....	167
Б.1 Здібність до навчання і накопичення інформації.....	167
Б.2 Апроксимаційні властивості нейронних мереж	167
Б.3 Апроксимація і пам'ять в багатошарових нейронних мереж	170
ДОДАТОК В. НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ	
НЕЙРОННИХ МЕРЕЖ.....	174

В.1 Загальна характеристика алгоритмів навчання.....	174
В.2 Метод і алгоритм зворотного розповсюдження помилки.....	177
В.3 Перенавчання і узагальнення нейронних мереж.....	188
ДОДАТОК Г. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ	
З ПРОГНОЗОМ	191
Г.1 Алгоритм нейроуправління з прогнозомі	191
Г.2 Структура нейронної мережі для моделювання об'єкту.....	198
Г.3 Прогноз з використанням нейронної мережі.....	202
Г.4 Похідні рівнянь нейронної мережі.....	203
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	205

ВСТУП

Актуальність теми. В нашій країні проводиться велика робота по розробці нових конструкцій підйомно-транспортних комплексів, машин і механізмів з автоматичним управлінням, машин циклічної дії і автоматизованих конвеєрних ліній, значно збільшується випуск засобів механізації підйомний-транспортних, завантажувальних для навантаження і складських робіт. Обладнання кранів є ключовим елементом комплексної механізації у всіх галузях народного господарства.

Більшість вантажопідйомних машин, вироблених вітчизняною промисловістю, оснащені електричним приводом для основних робочих механізмів, тому їх ефективність значною мірою залежить від якості електроустаткування крана. Електроприводи більшості вантажопідйомних машин працюють у режимі короточасної роботи з високою частотою включень, забезпечують широкий діапазон регулювання швидкості і стикаються з постійними значними перевантаженнями під час розгону та гальмування механізмів

В даний час накопичений великий досвід проектування, виробництва, експлуатації і модернізації електроприводів мостових кранів. Проте комплексне рішення задачі підвищення ефективності мостових кранів неможливе без вдосконалення систем управління їх електроприводами. Саме якість систем управління електроприводами визначає можливість реалізації оптимальних режимів роботи механічного устаткування, забезпечення максимальної продуктивності і безперебійної роботи кранів. Розробка систем управління електроприводами, що забезпечують високу продуктивність мостових кранів, є актуальним завданням.

Перспективним напрямком є створення систем управління на основі адаптивного підходу, поєднаного з методами теорії штучних нейронних мереж (ШНМ).

В даний час в області ШНМ працюють учені різних напрямів таких, як комп'ютерні науки і інженерія, математика, фізика, біологія, психологія, ней-

рофізіологія, що використовують різні підходи і методи і різні завдання, що ставлять перед собою, у зв'язку з чим досить важко міркувати про теорію штучних нейронних мереж як про щось монолітно ціле. Сьогодні можна вже говорити про проектування штучних нейронних мереж, що займається архітектурою мереж, нейроінформатиці, що досліджує процеси навчання штучних нейронних мереж, нейрокомп'ютингу, пов'язаному з розпаралелюванням обчислень, нейроморфній інженерії, що займається вирішенням прикладних проблем.

Розглядаючи ШНМ з вищого рівня, можна говорити про них як про самостійний напрям в теорії інтелектуальних систем, з якими їх ріднить здібність до накопичення і узагальнення інформації, використання її для вирішення проблем і накопичення нових знань в процесі функціонування. Відмінності ж нейронних мереж від інших інтелектуальних систем визначаються їх здібністю до навчання по набору прикладів, їх запам'ятовування, формування асоціацій, відновлення і відтворення даних і знань, стійкістю до збоїв, підвищеною надійністю. Як визначальну відмітну ознаку ШНМ слід виділити їх здібність до навчання, запам'ятовування і відновлення спотвореною або частково зруйнованою інформації.

Розвиток ШНМ пов'язаний із розумінням того, що мозок живого організму функціонує інакше, ніж комп'ютер. Людський мозок – це складна нелінійна паралельна система, що відповідає за інформаційне управління, мислення, збереження і відновлення інформації, а також за вирішення проблем. Він складається з численних однотипних структурних одиниць – нервових клітин або нейронів, які є елементами обробки сигналів і отримують і комбінують інформацію через дендрити – свої вхідні зв'язки.

З інженерної точки зору, штучні нейронні мережі – це система паралельної обробки інформації, яка складається з простих обчислювальних вузлів (однакових або різних), що тісно взаємопов'язані. Ці мережі мають властивість накопичувати досвід, узагальнювати знання і надавати їх у формі, зручній для розуміння та ухвалення рішень.

Сьогодні можна говорити про значні досягнення нейромережевих технологій у вирішенні складних завдань, як у науковій сфері, так і в техніці, бізнесі, фінансах, медичній діагностиці та інших областях інтелектуальної діяльності.

Створення нових і вдосконалення існуючих систем управління основними механізмами мостового крана є важливим завданням для народного господарства, що підкреслює актуальність цієї теми.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму переміщення візка мостового крана 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 переміщення візка мостового крана з урахуванням поперечних розгойдувань вантажу.

Предметом дослідження є моделі і методи синтезу нейромережевої системи управління двомасовою електромеханічною системою механізму переміщення візкамостового крана з урахуванням поперечних розгойдувань вантажу.

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 переміщення візка мостового крана.

2. Вперше запропоновано і синтезовано нейромережеву систему управління електроприводом механізму переміщення візка мостового крана з урахуванням пружних елементів, що має високі показники якості функціонування.

3. Вперше запропонована математична модель системи управління електроприводом механізму переміщення візка мостового крана з урахуванням поперечних розгойдувань вантажу у вигляді багат шарової прямонаправленої мережі – багат шарового 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 переміщення візка мостового крана, що забезпечує високоякісне регулювання.

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

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

Серед них: синтезована система регулювання координат електроприводу; розроблена модель системи з урахуванням пружних елементів, розроблена структурна схема 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.

У роботі наведено посилання на авторів і відповідні джерела при використанні відомих залежностей, експериментальних даних та 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 205 сторінок, серед них: 61 рисунок, з яких 38 рисунків за текстом, 23 рисунків на 23 окремих сторінках, 2 таблиці, 5 додатків на 48 сторінках, список літературних джерел з 79 найменування на 8 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР НЕЙРОМЕРЕЖЕВИХ СИСТЕМ УПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Розвиток теорії штучних нейронних мереж

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

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

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

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

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

$g_i(t)$ – її задане значення;

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

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

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

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

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

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

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

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

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

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

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

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

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

У даній роботі для синтезу нейромережової системи використовуються багатошарові прямонаправлені мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямку без зворотних зв'язків всередині мережі – багатошаровий персептрон. В додатку А розглянуто структуру штучного нейрона і структуру багатошарових нейронних мереж, а в додатку Б - властивості зазначених 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 багатошарової 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 нейромережової 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.

$$\tilde{y}(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n), k] + \xi(k), \quad (1.2)$$

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:

$$x(k) = [\tilde{y}(k-1), \tilde{y}(k-2), \dots, \tilde{y}(k-m), u(k-1), u(k-2), \dots, u(k-n)]. \quad (1.3)$$

Тоді рівняння (1.2) може бути записане таким чином:

$$\tilde{y}(k) = f[x(k), k] + \xi(k). \quad (1.4)$$

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.

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (1.2) сьогодні отримали багат шаровий 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

$$\hat{y}(k) = f^q \left[(\omega^q)^T f^{q-1} \left[(\omega^{q-1})^T f^{q-2} \left[\dots f^1 \left[(\omega^1)^T \right] x(k) \right] \right] \right], \quad (1.5)$$

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:

$$\hat{y}(k) = \sum_{i=1}^N c_i \Phi_i(x), \quad (1.6)$$

де c_i – вагові коефіцієнти, які підлягають визначенню.

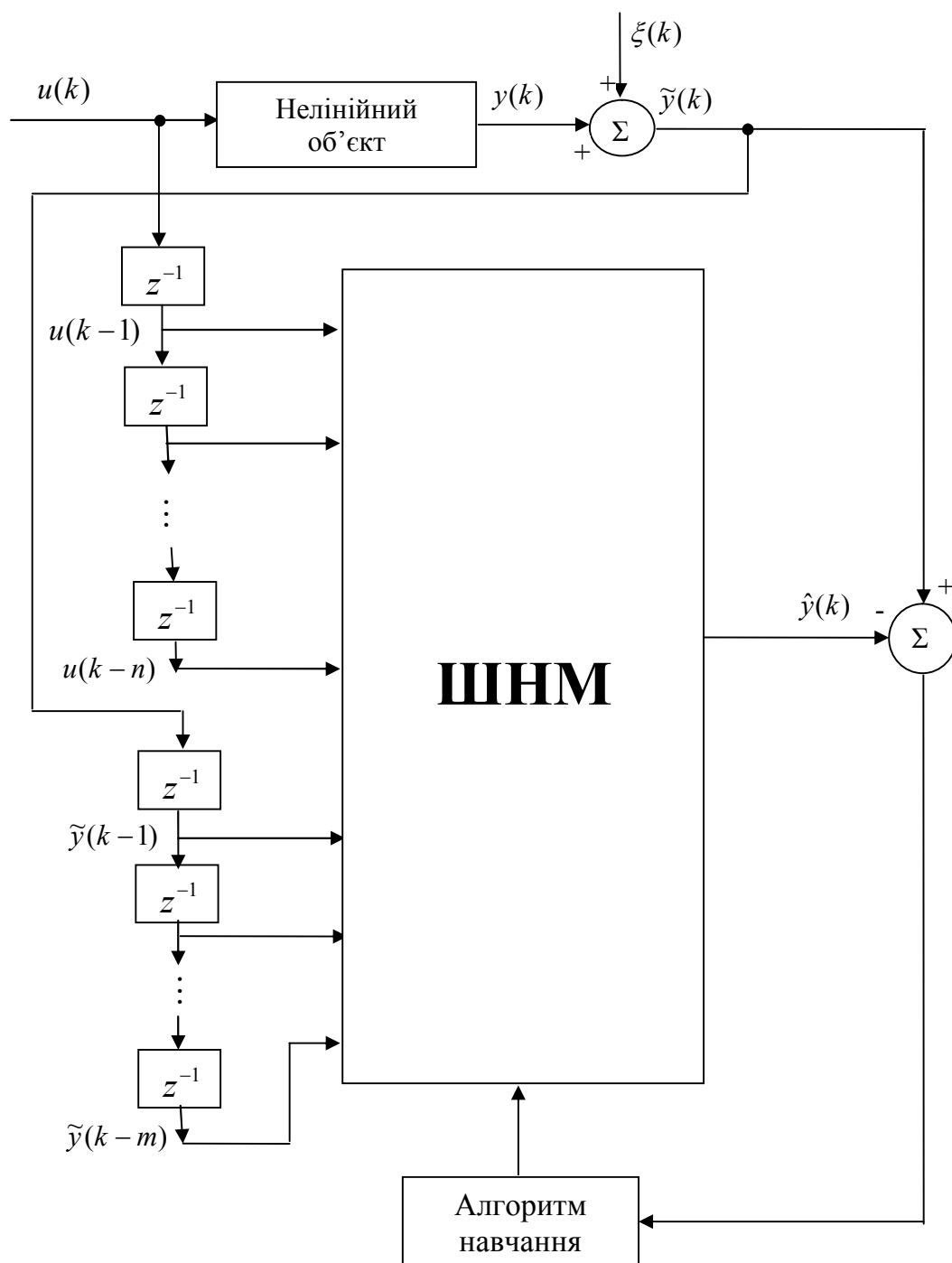


Рисунок 1.1 – Схема ідентифікації динамічного об'єкту

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

$$\Phi_i(x) = \exp \left\{ -\frac{\|x - \mu_i\|^2}{\sigma_i^2} \right\}, \quad (1.7)$$

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 $\Phi_0(x) = 1$).

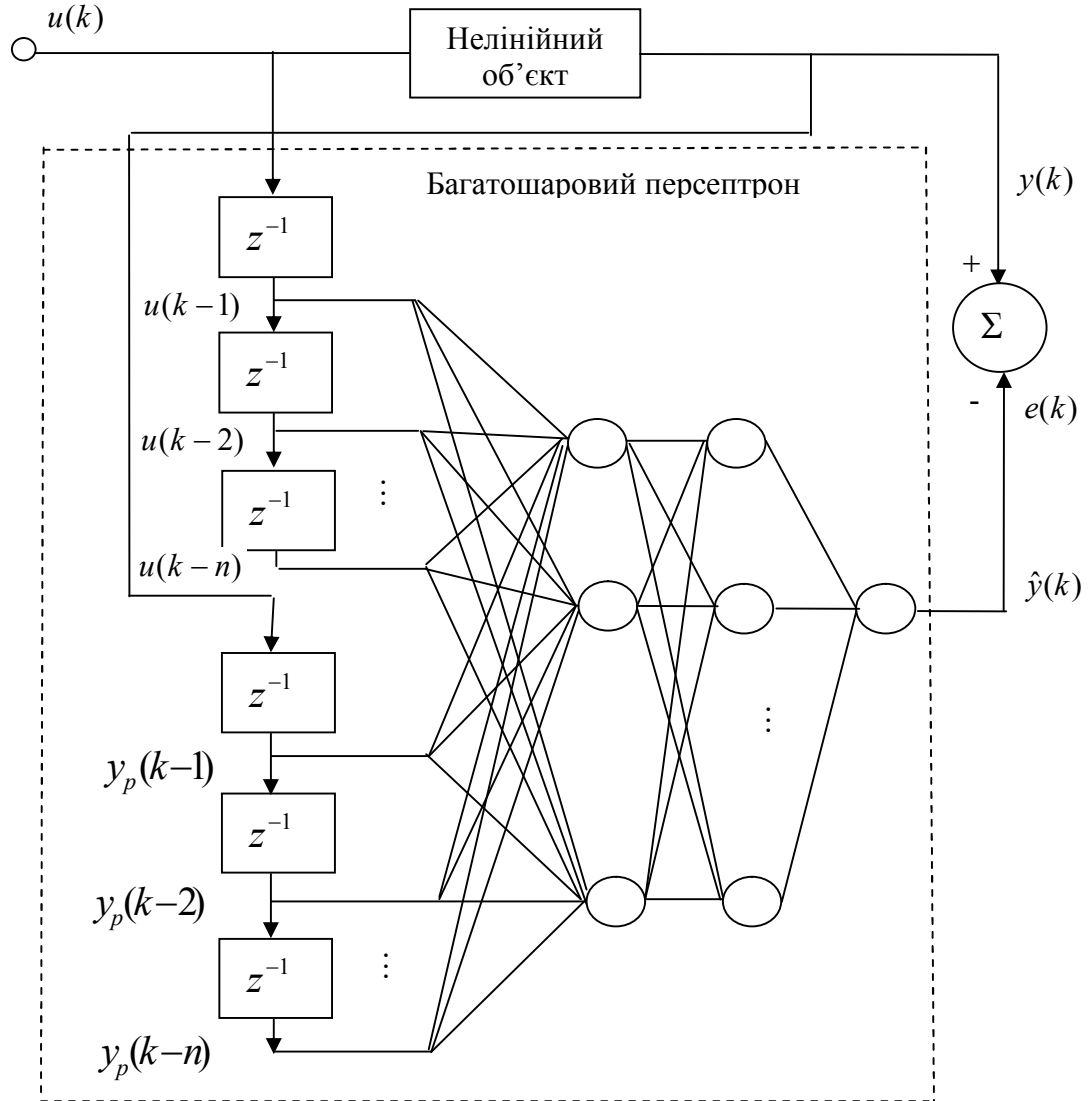


Рис 1.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

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

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (1.8)$$

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

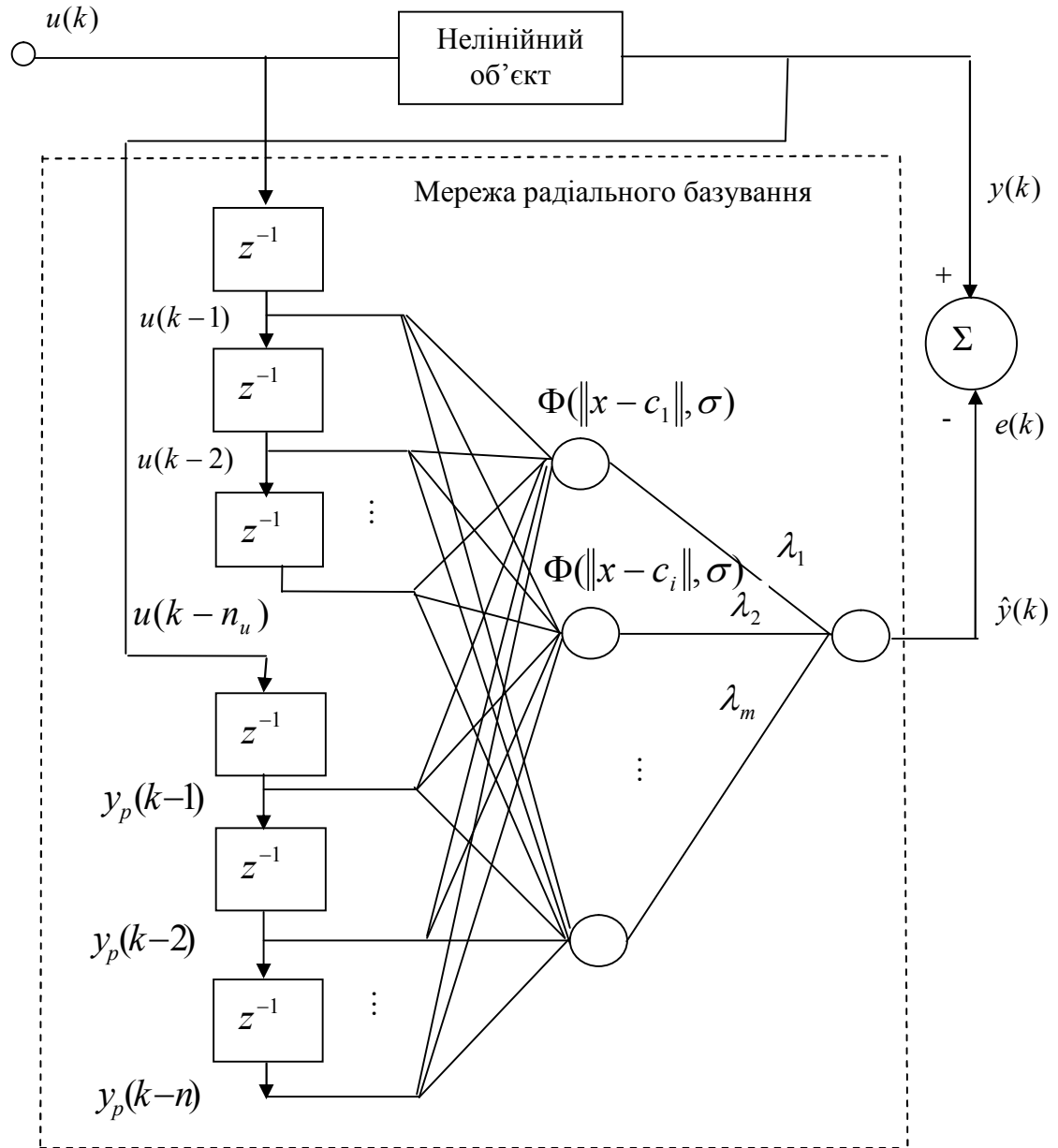


Рисунок 1.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

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 доцільною 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.

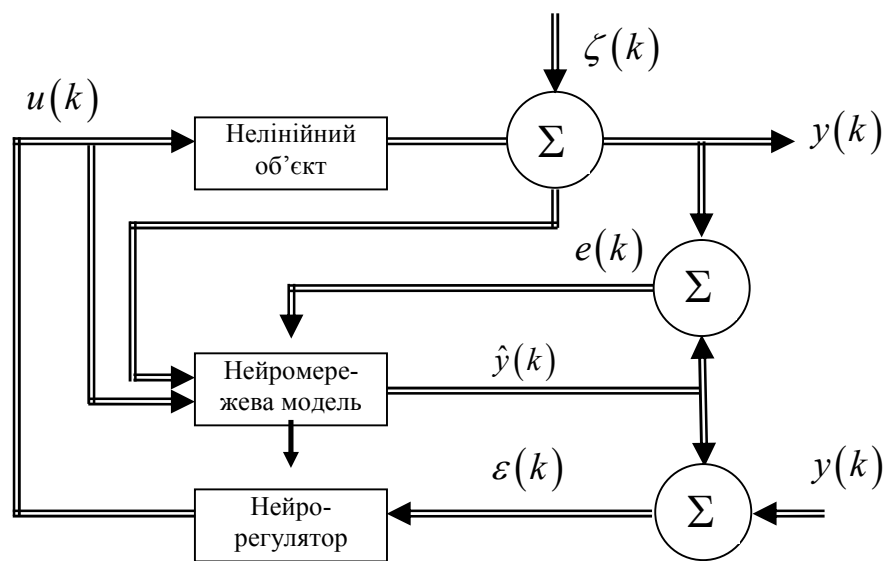


Рисунок 1.4 – Структура нейромережевої системи адаптивного управління

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

Розглянемо багатовимірний нелінійний динамічний об'єкт, описуваний рівнянням

$$y(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)] + \xi(k), \quad (1.9)$$

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

$$x(k) = [y^T(k-1), \dots, y^T(k-m), u^T(k-1), \dots, u^T(k-n)]^T, \quad (1.10)$$

перепишемо рівняння (1.9) у вигляді

$$y(k) = f(x(k) + \zeta(k)). \quad (1.11)$$

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.

Після завершення процесу навчання мережа використовується для реалізації алгоритму керування.

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

$$J(k) = \|\varepsilon(k)\|^2 = \frac{1}{2} \|y^*(k) - \hat{y}(k)\|^2, \quad (1.12)$$

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

Як й у випадку ідентифікації, для знаходження мінімуму функціонала (1.12) можуть бути використані різні рекурентні алгоритми, зокрема, градієнти вигляду

$$u(k) = u(k-1) + \gamma(k)(\nabla_u \varepsilon(k)), \quad (1.13)$$

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 [39]. 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.

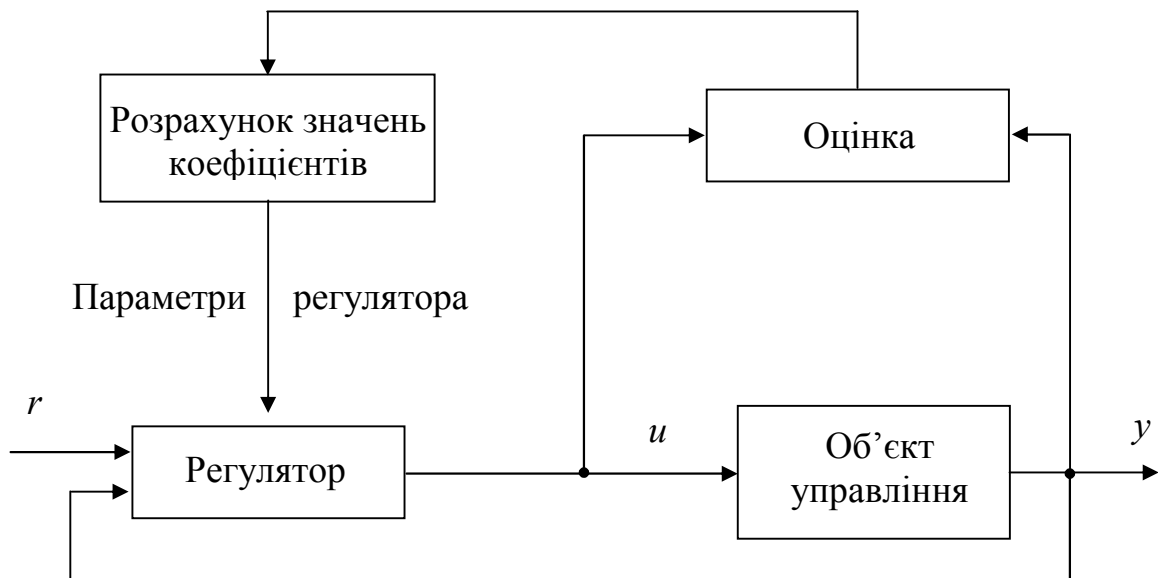


Рисунок 1.5 – Блок-схема управління із зворотним зв'язком і регульованими коефіцієнтами

Іншою добре відомою моделлю СУ є модель Ляпунова. Системи адаптивного управління, що використовують еталонну модель Ляпунова проектується так, щоб вихідний сигнал керованої моделі врешті-решт відповідав вихідному сигналу заздалегідь визначеної моделі, яка має бажані характеристики [40]. 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 [41]. Тому ШНМ, навіть з одним прихованим шаром, можуть використовуватись для задач ідентифікації та управління.

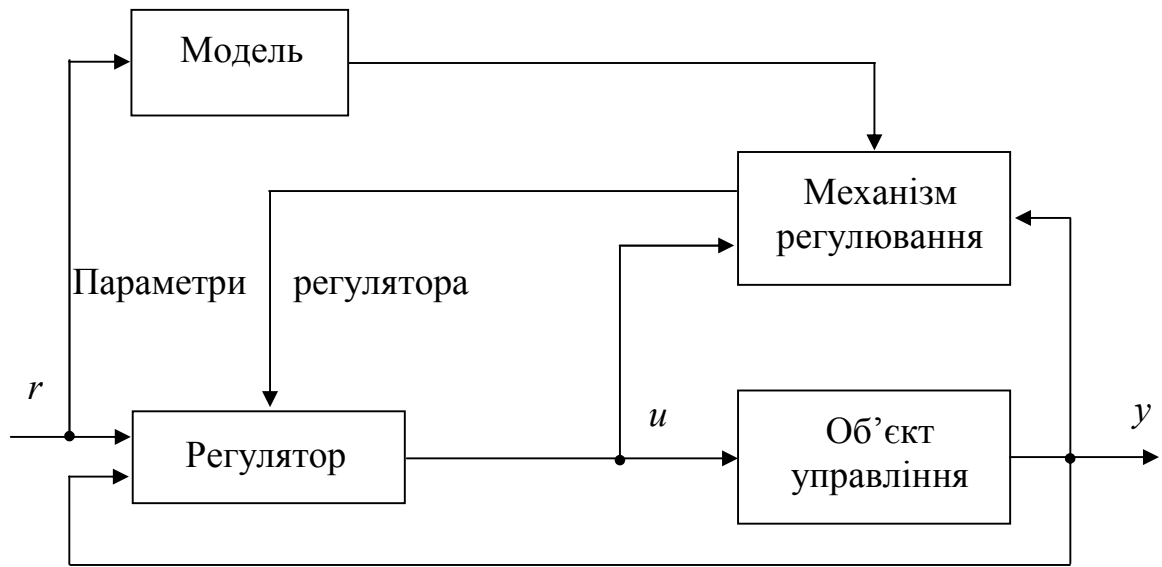


Рисунок 1.6 – Блок-схема адаптивного управління з еталонною моделлю

Одним з перших використовуваних методів побудови нейромережових 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 нейромережових систем управління, які добре розроблені і широко використовуються [42?43]. У всіх випадках нейромережовий 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.

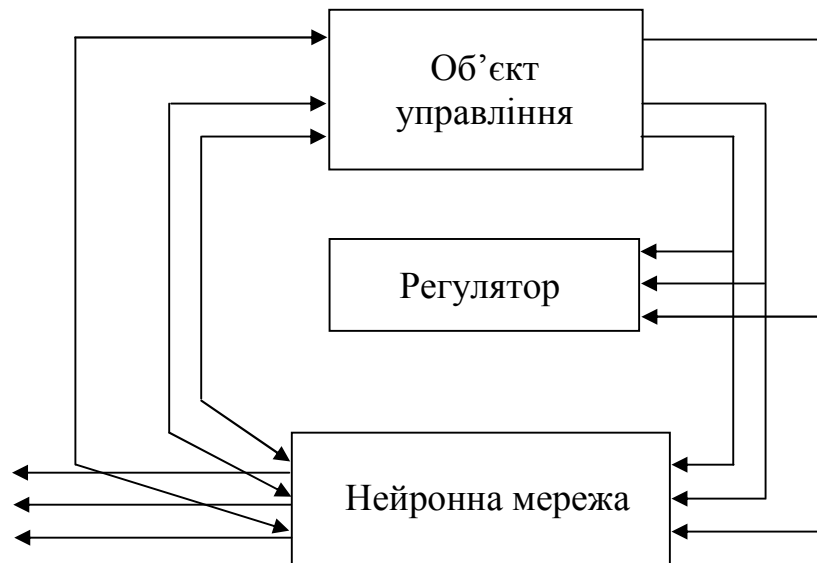


Рисунок 1.7 – Нейромережева СУ, заснована на «копіюванні» існуючого контролера

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).

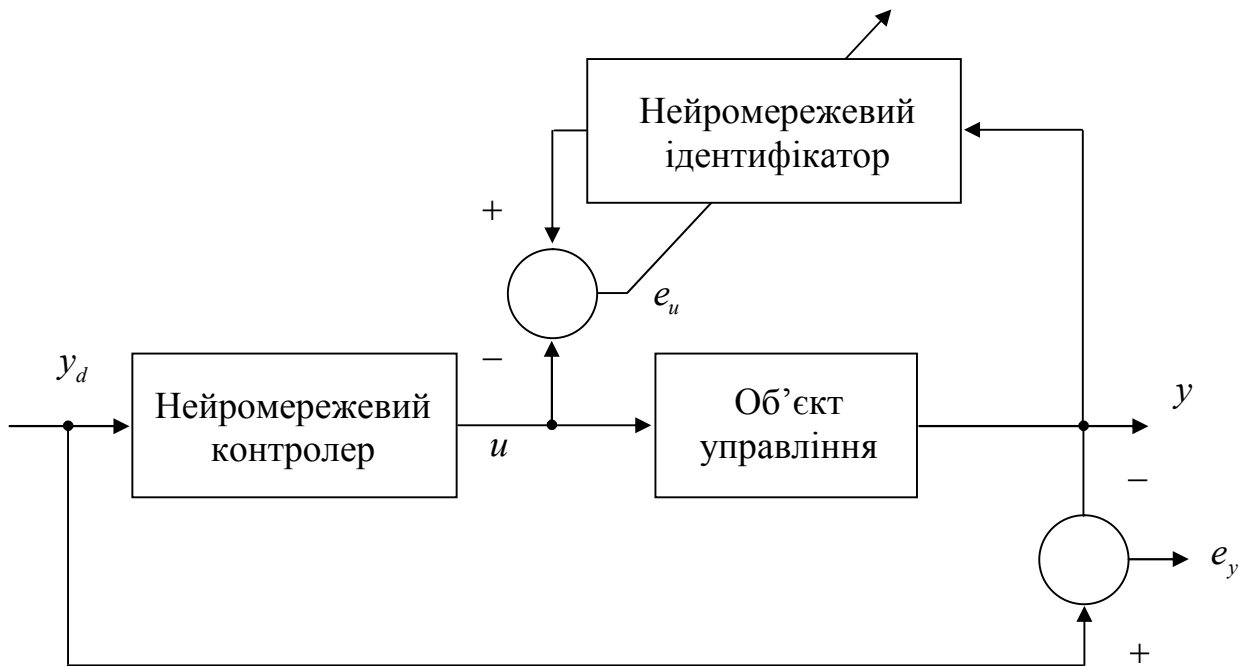


Рисунок 1.8 – Схема непрямого управління

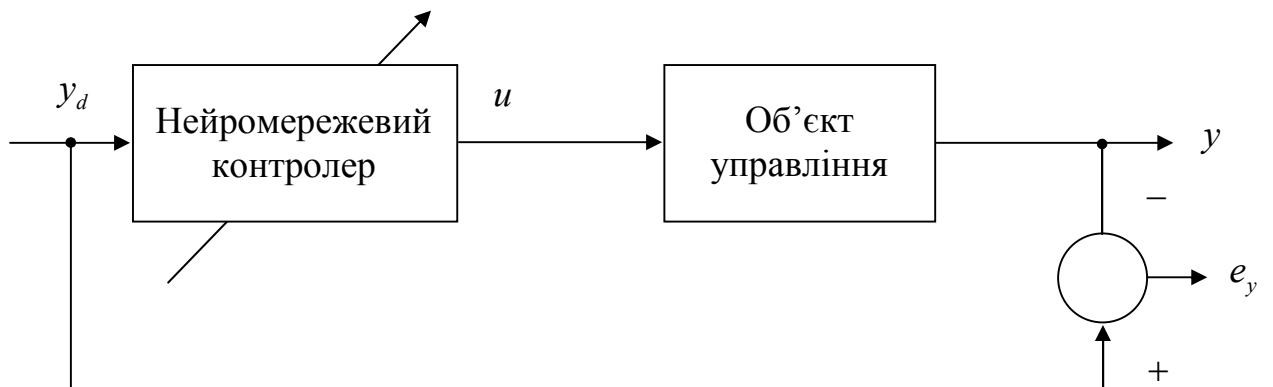


Рисунок 1.9 – Схема непрямого управління

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 [44-48].

Принцип предиктивного регулювання на основі моделі (model predictive control, MPC) [49] (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).

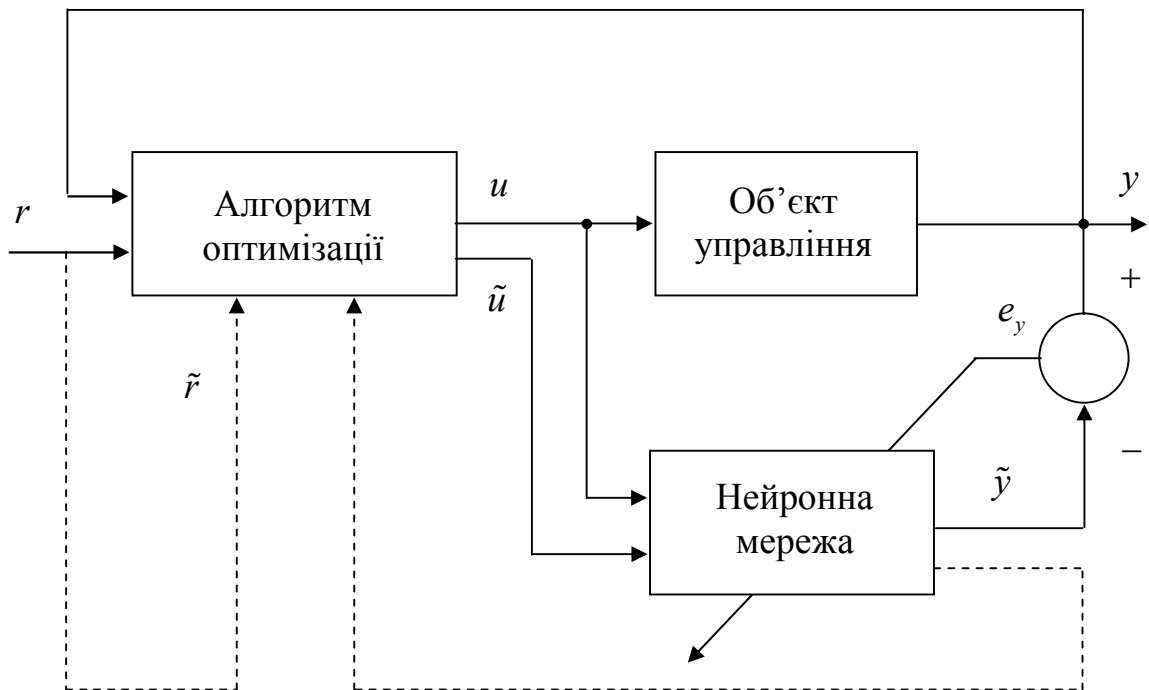


Рисунок 1.10 – Предиктивне регулювання

У цьому випадку регулятор є ітераційним алгоритмом оптимізації [49]:

- 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 [50] 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 Основні характеристики мостових кранів

У народному господарстві України використовуються електричні крани різних типів і конструкцій. У металургійних і машинобудівних цехах активно експлуатуються мостові крани, у дворах копалень і на вугільних складах електростанцій – порталні та козлові перевантажувальні крани, а на будівельних майданчиках – баштові крани тощо.

Серед промислових підприємств найбільш універсальним і поширеним підйомно-транспортним пристроєм є мостовий кран. Його металева конструкція моста спирається на ходові візки, які за допомогою механізму пересування переміщуються по підкранових шляхах, встановлених на стаціонарних опорах над обслуговуваною площею. Уздовж моста прокладені рейки, по яких пересувається візок із встановленим на ньому механізмом пересування та підйомною лебідкою, що виконує підйом і спуск вантажів.

Таким чином, основними механізмами мостового крана є: механізм пересування моста, механізм пересування візка та механізм підйому лебідки, кожен із яких оснащений окремим електродвигуном.

Електричне устаткування кранів повинне забезпечувати надійну роботу при повторнокороткочасному режимі і великій частоті включень, вологості повітря, різких змін температури. До електроустаткування пред'являється жорсткі вимоги в сенсі безперебійності в роботі, високій продуктивності, безпеці обслуговування і простоти експлуатації.

Номінальна вантажопідйомність кранів регламентовані стандартом і є рядом дискретних значень в діапазоні від 1 до 1000 t . Середні маси переміщуваних вантажів складають $30 \div 50\%$ номінальної вантажопідйомності. Для мостових кранів з лебідковими візками $G_{\text{вп}} = 5 \div 100 t$

Режими роботи вантажопідйомних кранів всіх типів і призначень встановлені стандартом. Все різноманіття вантажопідйомних кранів охоплене вісьма режимними групами $1K - 8K$. Відповідно до стандарту встановлено шість режимів експлуатації механізмів кранів $1M - 6M$. Мостові крани з лебідковими вантажопідйомними візками для роботи в цехах і на складах промислових підприємств відносяться до груп, режим роботи механізму яких $4M, 5M$

Номінальні швидкості руху робочих органів регламентовані стандартами і для режимних груп $5K, 7K$ складають: механізм підйому $v_{\text{п}} = 0.1 \div 1.2 \text{ м / с}$; пересування моста $v_{\text{м}} = 0,8 \div 1,5 \text{ м / с}$; пересування візка $v_{\text{в}} = 0,5 \div 0,7 \text{ м / с}$.

Вимоги до електроустаткування кранів: максимальна напруга 500 В змінного струму.

На кранах передбачено обмеження ходу вгору на механізмі підйому і обмеження ходу в обидві сторони візка. Захист електричних ланцюгів і двигунів від короткого замикання і перевантаження вище 200 % виконується струмовим реле.

У схемі управління краном повинен бути встановлений нульовий захист, що оберігає двигуни від самозапуску при подачі напруги після перерви в електропостачанні.

Всі механізми кранів повинні бути забезпечені гальмами, підйомні механізми обов'язково автоматичними гальмами закритого типу живлення, що діють

при відключенні. Крани, що працюють поза приміщеннями, повинні бути забезпечені протиугінними засобами. На потужних мостових кранах повинна бути встановлена апаратура, що запобігає перенесенню ферми моста понад допустимі межі.

1.6 Обґрунтування вибору системи електроприводу механізму переміщення моста мостового крана

Системи управління механізмами кранів належать до пристроїв, що працюють під постійним контролем оператора. У таких системах вибір моменту початку операції, її швидкісних параметрів та завершення здійснюється людиною, яка керує механізмом. Завдання системи управління полягає в забезпеченні необхідної послідовності перемикачів для реалізації заданих швидкісних параметрів, запобіганні недопустимим перевантаженням і гарантуванні відповідного захисту.

Різноманітність систем управління можна класифікувати за способом керування на такі групи:

1. Системи з прямим керуванням за допомогою силових кулачкових регуляторів: У цьому випадку всі етапи управління, включаючи вибір прискорень, здійснює виключно оператор.
2. Системи з кнопковими постами: Можливості управління обмежуються конструктивними особливостями поста та наперед заданою програмою запуску і гальмування.
3. Системи з комплексними пристроями керування: Наприклад, магнітними регуляторами із використанням енергетичних перетворювачів або без них. Оператор задає лише необхідну швидкість, тоді як процеси розгону, гальмування та всі необхідні проміжні перемикачів виконуються автоматично.

За умовами регулювання можна виділити такі типи:

1. Системи з регулюванням швидкості нижче номінальної.

2. Системи з регулюванням швидкості як вище, так і нижче номінальної.
3. Системи з регулюванням прискорення і уповільнення.

Відповідно до цієї класифікації, у кранах застосовуються наступні види електроприводів:

- Електропривод постійного струму з керуванням через силовий регулятор (К-ДП).
- Електропривод постійного струму з керуванням через магнітний регулятор (МК-ДП).
- Електропривод постійного струму з живленням і керуванням через тиристорний електропривод (ТП-Д).
- Електропривод постійного струму за схемою генератор–двигун (Г-Д).
- Електропривод змінного струму з асинхронним короткозамкненим двигуном, керованим магнітним пускатчем (МП-АДК).
- Електропривод змінного струму з асинхронним короткозамкненим двигуном, керованим силовим регулятором (К-АДК).
- Електропривод змінного струму з двошвидкісним асинхронним двигуном, керованим магнітним регулятором (МК-АДД).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим силовим регулятором (К-АДФ).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим силовим регулятором із динамічним гальмуванням через самозбудження (КД-АДФ).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим силовим регулятором із тиристорно-імпульсним ключовим регулюванням (КІ-АДФ).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим магнітним регулятором із гальмуванням методом противключення (МКП-АДФ).

- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим магнітним регулятором із динамічним гальмуванням через самозбудження (МКД-АДФ).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим магнітним регулятором із бездуговою комутацією (МКБ-АДФ).
- Електропривод змінного струму з асинхронним двигуном з фазним ротором, керованим магнітним регулятором із імпульсно-ключовим тиристорним регулюванням (МКІ-АДФ).
- Електропривод змінного струму з двошвидкісним асинхронним короткозамкненим двигуном, керованим безпосереднім перетворювачем частоти (НПЧ-АДД).

На основі аналізу умов експлуатації, режимів роботи та відповідно до рекомендацій [61] обрано систему електропривода тиристорний перетворювач–двигун (ТП-Д).

1.7 Постановка задачі синтезу нейромережевої системи управління

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

Використання нейромережевих технологій управління значно спрощує вирішення математичних проблем, пов'язаних із аналітичним синтезом і аналізом властивостей проєктованих систем. Це зумовлено тим, що якість процесів управління та досягнуті характеристики в нейромережевих системах переважно

визначаються фундаментальними властивостями багат шарових нелінійних нейронних мереж, а не аналітично виведеними оптимальними законами, які зазвичай реалізуються у вигляді програмного забезпечення. Налаштовувані багат шарові нейронні мережі мають низку переваг, які роблять їх ефективними для вирішення завдань управління динамічними об'єктами.

Універсальні властивості апроксимацій багат шарових мереж, грають ключову роль у формуванні нелінійних алгоритмів управління. На основі узагальненої теореми апроксимації Стоуна-Вейерштрасса зроблений висновок про те, що за допомогою нелінійних нейронних мереж можна скільки завгодно точно рівномірно наблизити будь-яку безперервну функцію багатьох змінних на будь-якій замкнутій обмеженій множині. Висновок послужив теоретичним обґрунтуванням можливого застосування багат шарових нейронних мереж для формування оптимальних функцій, що управляли, в динамічних системах. Через свою структуру і алгоритми навчання багат шарові мережі дозволяють вирішувати такі нелінійні задачі теорії управління, які традиційними аналітичними методами не завжди можна вирішити в повному об'ємі. Можливість формувати дії практично будь-якої форми, що управляють, робить штучні нейронні мережі засобом, адекватним рівню складності нелінійного завдання. Тому нейромережеві системи управління відносять до так званих «інтелектуальних» систем, тобто до систем, за визначенням, з неповною інформацією, з складною або немодельованою динамікою і неконтрольованою зміною власних властивостей.

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

Здібність нейронних мереж до паралельної обробки сигналів робить природним їх застосування для управління багатовимірними (або багатоканальними) об'єктами. Багат шарові нейронні мережі є однорідним обчислювальним середовищем. Це універсальні паралельні обчислювальні структури, призначені

для вирішення найрізноманітніших класів завдань, зокрема для побудови регуляторів в системах управління нелінійними динамічними об'єктами.

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

Метою роботи є розробка нейромережевої системи управління механізмом переміщення візка мостового крана з урахуванням поперечних розгойдувань вантажу.

Для досягнення поставленої мети необхідно вирішити наступні завдання. задач.

1. Виконати вибір системи електроприводу механізму переміщення візка мостового крана, вибір електродвигуна і основного електроустаткування силового кола електроприводу. Виконати синтез системи управління електроприводом і провести моделювання системи на ЕОМ.

2. Розробити математичну моделі системи управління з урахуванням поперечних розгойдувань вантажу, провести моделювання багатомасової системи на ЕОМ і виконати аналіз отриманих результатів.

3. Розробити структурну схему нейромережевої системи управління, що реалізує високоякісне регулювання з врахуванням пружності механічних зв'язків.

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Аналіз досліджень, присвячених застосуванню методів нейронних мереж для вирішення задач управління електромеханічними системами, виявив, що використання штучних нейронних мереж дає змогу створювати нейромережеву модель системи управління. Така модель здатна не лише формувати внутрішні уявлення про зовнішнє середовище та прогнозувати поведінку об'єкта управління, а й генерувати оптимальні керуючі дії відповідно до вимог еталонної моделі. Особливий акцент робиться на вдосконаленні методів навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів і розробці нейромережевих систем управління для нелінійних об'єктів.

2. Самонавчання штучних нейронних мереж дозволяє використовувати нейрорегулятори навіть за умов значних невизначеностей, тоді як для традиційних методів адаптивного управління необхідна велика кількість апріорної інформації про об'єкт управління. Простота реалізації нейронних мереж та їх здатність до навчання роблять їх особливо ефективними при управлінні складними нелінійними системами в реальному часі.

3. Ефективність використання багат шарових мереж прямого розповсюдження для створення нейромережевої моделі об'єкта управління була доведена. Багат шарова мережа прямого розповсюдження є універсальним аппроксиматором-класифікатором і є однією з найпоширеніших у практиці нейронних мереж. Описані методи навчання нейронних мереж, а також обраний метод навчання нейронної мережі регулятора.

4. Для забезпечення оптимальної роботи об'єкта необхідно коригувати параметри регуляторів залежно від змін робочих умов. У зв'язку з цим доцільно розробити системи управління, засновані на адаптивному підході в поєднанні з методами теорії штучних нейронних мереж (ШНМ). Це дозволить, з одного боку, створювати прості нейромережеві моделі, а з іншого – адаптивно налаштовувати параметри як цих моделей, так і регуляторів відповідно до змінних умов. На основі аналізу актуальних структур нейрорегулювання було обрано принцип

нелінійного предиктивного регулювання для побудови нейромережевого регулятора системи.

5. Розглянуто ефективну реалізацію узагальненого управління з прогнозуванням із використанням багат шарової прямонаправленої нейронної мережі як нелінійної моделі об'єкта управління. Продемонстровано, що застосування оптимізаційного алгоритму Ньютона-Рафсона значно зменшує кількість ітерацій, необхідних для досягнення збіжності, порівняно з іншими методами. Завдяки цьому алгоритм Ньютона-Рафсона є швидшим і придатним для управління в режимі реального часу.

6. Проведено аналіз вимог до систем управління електроприводами основних механізмів кранів. Масове використання і складні умови експлуатації електроприводів визначають жорсткі вимоги щодо їхньої простоти у використанні та високої надійності. Оскільки управління підйомними кранами на всіх етапах роботи здійснюється оператором, головною технологічною вимогою до вибору системи електроприводу є забезпечення можливості регулювання швидкості. Регулювання швидкості повинно виконуватись із врахуванням накладених обмежень на прискорення, струм і момент. Робочі швидкості основних механізмів більшості кранів забезпечують, що при допустимих середніх прискореннях час перехідних процесів займає лише незначну частку загального робочого циклу. Водночас умови регулювання моменту і характер перехідних процесів зазвичай не мають значного впливу на продуктивність крана і не потребують точного відтворення оптимальних законів руху. Однак під час проектування необхідно враховувати вплив таких факторів, як розгойдування вантажів і кінематичні зазори, а за потреби вживати заходів для обмеження динамічних навантажень.

7. Сформульовано завдання дослідження, яке полягає в розробці нейромережевої системи управління багатомасовою електромеханічною системою механізму переміщення візка мостового крана, що забезпечує високу якість регулювання.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА МОСТОВОГО КРАНА

2.1 Технічні характеристики мостового крана. Вимоги до системи управління

Вихідні дані для розрахунку електроприводу механізму підйому мостового крана і синтезу системи управління наведені в табл. 2.1.

Таблиця 2.1 – Технічні дані мостового крана

Вантажопідйомність Q , кг	41000
Продуктивність крана C , кг/с	240
Маса моста(без візка) G_m , кг	28500
Маса візка G_b , кг	8500
Діаметр коліс моста D_m , м	0,56
Діаметр коліс візка D_b , м	0,25
Швидкість пересування моста V_m , м/с	0,85
Швидкість пересування візка V_b , м/с	0,55
Швидкість підйому V_n , м/с	0,12
Маса вантажозахватного пристосування m_0 , кг	1000
Середня висота підйому і спуску H_n , м	8
Діаметр барабана механізму підйому D , м	0,75
Режими роботи електроприводів	4 М

Вимоги до показників якості функціонування системи управління.

Система управління повинна бути астатичною. Перерегулювання швидкості при ступінчастій вхідній дії не більше 10%, час регулювання не більше 5с.

2.2 Розрахунок потужності електроприводу механізму пересування візка. Вибір приводного двигуна

Вибір двигуна механізму пересування візка виконано по методиці, приведений в [61]. Дана методика враховує особливості роботи електроприводів кранів і є в даний час найбільш точною і зручною у використанні.

Для попереднього вибору електродвигуна визначаємо розрахункову потужність, тобто потужність, на яку повинен розраховуватися двигун крана за умовами забезпечення пускового режиму:

$$P_p = \frac{0,66(G + Q_{\pi})\alpha_b V_b}{10^3 \eta_b} + \frac{P_{ст}}{1,75}, \quad (2.1)$$

де G - сумарна маса переміщуваного вантажу, тобто маса візка;

$$G = G_b = 8,5 \cdot 10^3 \text{ кг},$$

Q_{π} – маса вантажу з підвіскою

$$Q_{\pi} = Q + q,$$

$$Q_{\pi} = (41 + 0,85) \cdot 10^3 = 41,85 \cdot 10^3 \text{ кг}.$$

α_b – прискорення візка, приймаємо $\alpha_m = 0,3$ згідно табл. 1.4 [61]

η_b – КПД механізму, приймаємо $\eta_b = 0,9$ згідно табл. 1.6 [61].

$P_{ст}$ – потужність статичного навантаження; для механізмів горизонтального переміщення, що працюють в приміщенні за відсутності вітрового навантаження $P_{ст}$ визначається по наступній формулі:

$$P_{ст} = \frac{(G + Q)V_b}{10^3 m_b \eta_b} \left(\frac{\varphi_{\pi} d_{ст} + 2\mu}{D_b} k_{pb} + \beta \right), \quad (2.2)$$

де φ_{π} – коефіцієнт тертя в підшипниках маточин коліс, для підшипників качення $\varphi_{\pi} = 0,015$;

D_B – діаметр ходового колеса;

$d_{ст}$ – діаметр ступиць ходового колеса, зазвичай для розрахунків застосовується $d_{ст} / D_M = 0,25$;

μ – коефіцієнт тертя кочення; $\mu = 0,5 \cdot 10^{-3}$ м;

$k_{рб}$ – коефіцієнт форми ходового колеса, що враховує тертя реборд ходового колеса $k_{рб} = 1,3 \div 1,4$; приймаємо $k_{рб} = 1,4$;

m_B – число механізмів пересування, для візка $m_B = 1$;

β – ухил рейкового шляху візка; при розрахунку мостових кранів приймається $\beta = 0,003$;

V_M – швидкість пересування візка

$$P_{ст} = \frac{9,8 \cdot (8,5 + 41,85) \cdot 10^3 \cdot 0,55}{10^3 \cdot 1 \cdot 0,9} \left(\frac{0,015 \cdot 0,25 \cdot 0,25 + 2 \cdot 0,5 \cdot 10^3}{0,25} \cdot 1,4 + 0,003 \right) = 4,46 \text{ кВт}$$

Розрахункова потужність

$$P_p = \frac{0,66 \cdot (8,5 + 41,85) \cdot 10^3 \cdot 0,3 \cdot 0,55}{10^3 \cdot 0,9} + \frac{4,3}{1,75} = 8,16 \text{ кВт}.$$

Розрахункова потужність двигуна по тепловому режиму визначається як

$$P_{ном.т} = P_{ст} / k_t, \quad (2.3)$$

де k_t – коефіцієнт, залежний від режиму роботи і вибраної системи електроприводу; $k_t = 1,3$ по табл. 6.5 [61]

$$P_{ном.т} = 4,46 / 1,3 = 3,43 \text{ кВт}.$$

Оскільки $P_p > P_{ном.т}$ то вибір двигуна здійснюємо по P_p . По каталогу вибираємо двигун Д806 з наступними технічними характеристиками:

– номінальна потужність при ПВ=40% $P_n = 9,5 \text{ кВт}$;

- номінальна напруга $U_H = 220B$;
- номінальний струм $I_H = 57 A$;
- частота обертання $n_H = 800 об/хв$;
- момент інерції $J_{дв} = 0,425 кг \cdot м^2$;
- $N = 558$, $2a = 2$; $2p = 4$; $R_{я} + R_{дп} = 0,28 Ом$, $\Phi_H = 0,0132 Вб$; $\eta = 0,77$.

Для перевірки правильності вибору двигуна по тепловому режиму визначимо еквівалентний ККД

$$\eta_{екв} = \frac{\eta_{екв.б}}{1 + \frac{(\eta_{екв.б} - \eta_{екв.N})}{\eta_{екв.N}} \cdot \frac{J_{\Sigma}}{1,2 \cdot J_{дв}} \cdot \left(\frac{n_{max}}{1000}\right)^2}, \quad (2.4)$$

де J_{Σ} - сумарний момент інерції, приведений до валу двигуна.

$$J_{\Sigma} = 1,2J_{дв} + \frac{91 \cdot (G + Q_{п}) \cdot V_B^2}{m_B \cdot n_H^2 \cdot \eta_B}, \quad (2.5)$$

$$J_{\Sigma} = 1,2 \cdot 0,425 + \frac{91 \cdot (8,5 + 41,95) \cdot 10^3 \cdot 0,55^2}{1 \cdot 850^2 \cdot 0,9} = 0,51 + 2,95 = 3,46 кг \cdot м^2.$$

$\eta_{екв.б}$ – базовий ККД, відповідний числу виключень $N_B = 0$; для тиристорного електроприводу постійного струму $\eta_{екв.б} = 0,94$ по табл. 6.5 [61].

$\eta_{екв.N}$ – еквівалентний ККД при числі включень $N_B = 120$; $\eta_{екв.N} = 0,92$ по рис.6.5 [61].

n_{max} – максимальна частота обертання, приймаємо $n_{max} = n_H = 800 об/хв$

$$\eta_{екв} = \frac{0,94}{1 + \frac{0,94 - 0,92}{0,92} \cdot \frac{3,46}{1,2 \cdot 0,425} \cdot \left(\frac{800}{1000}\right)^2} = 0,827.$$

Розрахункова номінальна потужність приводу по тепловому режиму визначається по формулі

$$P_{\text{розр}} = \frac{k_3 k_{\text{екв}} k_{\text{н}} \sqrt{\varepsilon_p / \varepsilon_{\text{ном}}}}{k_0 k_p} \cdot \frac{\eta_{\text{екв.б}}}{\eta_{\text{екв.б}} - k_{\text{дн}} (\eta_{\text{екв.б}} - \eta_{\text{екв}})} \cdot P_{\text{ст}}, \quad (2.6)$$

де k_3 – коефіцієнт, що враховує запас потужності, що докладається при проектуванні приводу, для механізмів групи 4 м $k_3 = 1$ згідно табл.6.4 [61];

$k_{\text{екв}}$ – коефіцієнт, що визначається по табл. 6.4[61], для групи 4м $k_{\text{екв}} = 0,63 \div 0,75$; приймаємо $k_{\text{екв}} = 0,68$;

ε_p – фактична тривалість включення ; $\varepsilon_p = 0,4$ з табл.6.4[61];

$k_{\text{дн}}$ – динамічний коефіцієнт; $k_{\text{дн}} = 1,25$ згідно табл.6.5 [61];

$k_0, k_p, k_0, k_{\text{н}}$ – коефіцієнти, що визначаються по рис. 6.12 [61]: $k_0 = 1, k_p = 1, k_0 = 1, k_{\text{н}} = 1,$

$$P_{\text{розр}} = \frac{1 \cdot 0,68 \sqrt{0,4 / 0,4} \cdot 1}{1 \cdot 1} \cdot \frac{0,94}{0,94 - 1,25 \cdot (0,94 - 0,827)} \cdot 4,46 = 4,41 \text{ кВт}.$$

Оскільки номінальна потужність двигуна $P_{\text{н}}$ більша $P_{\text{ном.т}}$, то вибраний двигун за умовами нагріву відповідає вимогам.

Перевіримо електропривод по запасу зчеплення при пуску і гальмуванні для найменше сприятливих умов роботи. Для цього обчислюємо прискорення, яке забезпечується проектованим електроприводом

$$a_{\text{дв}} = \frac{10^3 [k_{\text{п}} \gamma_{\text{п}} \eta_{\text{м}} P_{\text{п}} - P_{\text{ст}}]}{(G + Q_{\text{п}}) V_{\text{в}} \alpha'}, \quad (2.7)$$

де $k_{\text{п}}, \gamma_{\text{п}}$ – коефіцієнти використання двигуна по пусковому моменту і стислість відношення максимального пускового моменту до номінального моменту двигуна; $k_{\text{п}} = 0,9 \quad \gamma_{\text{п}} = 2$ з табл. 6.2 [61]

α' – коефіцієнт, визначуваний співвідношенням

$$\alpha' = \frac{J_{\Sigma}}{J_{\text{пост}}} = \frac{J_{\Sigma}}{J_{\Sigma} - 1,2J_{\text{дв}}} , \quad (2.8)$$

$$\alpha' = \frac{3,46}{3,46 - 1,2 \cdot 0,425} = 1,18 ,$$

$$a_{\text{дв}} = \frac{10^3 \cdot [0,9 \cdot 2 \cdot 0,9 \cdot 8,16 - 4,46]}{(8,5 + 41,85) \cdot 10^3 \cdot 0,55 \cdot 1,18} = 0,24 .$$

Максимальне прискорення приводу визначається згідно (6.51) [61].

$$\alpha_{\text{max}} = 1,4\alpha_{\text{дв}} - 0,09 = 1,4 \cdot 0,24 - 0,09 = 0,246 .$$

Оскільки виконується умова $\alpha_{\text{дв}} < \alpha_{\text{max}}$, то двигун задовольняє умовам зчеплення і за умовами зчеплення може бути використаний для електроприводу механізму переміщення візка.

2.3 Вибір основного електроустаткування і розрахунок параметрів силового кола електроприводу

Відповідно до рекомендацій [61], для електроприводу механізму пересування техніки вибираємо тиристорний перетворювач типу АТРК-100/230. Номінальна випрямлена напруга $U_{\text{дн}} = 230 \text{ В}$; номінальний випрямлений струм $I_{\text{дн}} = 100 \text{ А}$; номінальна випрямлена напруга збудження $U_{\text{в.н}} = 230 \text{ В}$; номінальний випрямлений струм збудження $I_{\text{в.н}} = 15 \text{ А}$.

Перетворювач підключається до мережі через силовий трансформатор типа ТСЗР-40/0,5. Технічні дані трансформатора: номінальна напруга мережевої обмотки 380 В, вентиляної обмотки 200 В, номінальна сила струму мережевої обмотки 48,6 А, вентиляної обмотки 92 А. Напруга короткого замикання $U_{\text{кз}}$ трансформаторів серії ТСЗР складає 8%.

Для електродвигунів серії Д допустимий рівень пульсацій першої гармонійної випрямленого струму не повинен перевищувати 5-10%, тому необхідно розглянути питання про необхідність включення в силове коло системи ТП-Д згладжуючого реактора.

Сумарна індуктивність силового кола електроприводу, необхідна для згладжування пульсацій до заданого рівня, визначається по формулі

$$L_d = \frac{e_n E_0}{i_e \omega_1 I_{нп}}, \quad (2.9)$$

де e_n - відносна величина першої гармоніки випрямленої напруги; для трифазної мостової схеми випрямлення при граничних кутах регулювання $\alpha_{\max}$ величина e_n знаходяться в межах 0,22-0,24. Приймаємо $e_n = 0,23$.

E_0 – максимальне значення випрямленою ЕРС при вугіллі управління тиристорами $\alpha = 0$.

$$E_0 = \frac{m}{\pi} \sqrt{2} U_2 \sin \frac{\pi}{m}, \quad (2.10)$$

U_2 - напруга вентильної обмотки трансформатора;

m – число фаз перетворювача, для трифазної мостової схеми $m = 6$;

$$E_0 = \frac{6}{3,14} \sqrt{2} \cdot 200 \cdot \sin \frac{3,14}{6} = 270 \text{ В},$$

ω_1 – кутова частота першої гармонійної випрямленої напруги

$$\omega_1 = 2\pi m f, \quad (2.11)$$

$$\omega_1 = 2 \cdot 3,14 \cdot 6 \cdot 50 = 1884 \text{ с}^{-1}$$

$I_{нп}$ - номінальна величина сили струму навантаження перетворювача, приймаємо рівною номінальному струму двигуна.

$$L_d = \frac{0,23 \cdot 270}{0,1 \cdot 2 \cdot 3,14 \cdot 6 \cdot 50 \cdot 57} = 0,0058 \text{ Гн}.$$

Індуктивність якірної обмотки двигуна:

$$L_{\text{як}} = \gamma \cdot \frac{U_{\text{н}}}{I_{\text{н}}} \cdot \frac{1}{\omega_{\text{н}} p}, \quad (2.12)$$

де γ – коефіцієнт, рівний $0,5 \div 0,6$ для некомпенсованих двигунів і $k = 0,25$ для компенсованих машин; використовуваний двигун є некомпенсованим, тому приймаємо $k = 0,6$.

p – число пар полюсів, для двигуна;

$\omega_{\text{н}}$ - номінальна кутова частота обертання двигуна

$$\omega_{\text{н}} = \frac{\pi n_{\text{н}}}{30}, \quad (2.13)$$

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

$$L_{\text{як}} = 0,6 \cdot \frac{220}{57} \cdot \frac{1}{83,7 \cdot 2} = 0,0138 \text{ Гн}.$$

Оскільки індуктивність обмотки якоря двигуна більше індуктивності L_d , то згладжуючий реактор не потрібний.

Приведена до ланцюга випрямленого струму індуктивність трансформатора:

$$L_{\text{т}} = \frac{U_{\text{кз}}}{100} \cdot \frac{1}{2mf} \cdot \frac{E_0}{I_{\text{н}}}, \quad (2.14)$$

$$L_{\text{т}} = \frac{8}{100} \cdot \frac{1}{2 \cdot 6 \cdot 50} \cdot \frac{270}{57} = 0,63 \cdot 10^{-3} \text{ Гн}.$$

Сумарна індуктивність силового кола електроприводу:

$$L_{\Sigma} = L_{\text{як}} + L_{\text{т}}, \quad (2.15)$$

$$L_{\Sigma} = 0,0138 + 0,63 \cdot 10^{-3} = 0,0144 \text{ Гн} .$$

Активний опір якірного кола електродвигуна, приведений до робочої температури:

$$R_{\text{як}} = k_t (R_{\text{я}} + R_{\text{дп}}) , \quad (2.16)$$

k_t - температурний коефіцієнт, що враховує зміну опору при нагріві, для двигунів серії Д, що має клас ізоляції Н $k_t = 1,36$;

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

Еквівалентний активний опір вентильного перетворювача визначається по формулі:

$$R_{\text{е.пр}} = R_{\text{е}} + R_{\text{т}} , \quad (2.17)$$

$R_{\text{е}}$ - еквівалентний опір анодного перекриття

$$R_{\text{е}} = m f L_{\text{т}} , \quad (2.18)$$

$$R_{\text{е}} = 6 \cdot 50 \cdot 0,63 \cdot 10^{-3} = 0,189 \text{ Ом} .$$

$R_{\text{е}} = m f L_{\text{р}}$ - приведенне до ланцюга випрямленого струму активний опір обмотки трансформатора

$$R_{\text{е}} = (0,1 \div 0,2) \frac{U_{\text{кз}} \%}{100} \cdot \frac{\pi}{m} \cdot \frac{E_0}{I_{\text{н}}} \quad (2.19)$$

$$R_{\text{т}} = 0,2 \cdot \frac{8}{100} \cdot \frac{3,14}{6} \cdot \frac{270}{57} = 0,0396 \text{ Ом} ,$$

$$R_{\text{е.пр}} = 0,189 + 0,0396 = 0,2286 \text{ Ом} .$$

Сумарний активний опір силового кола електроприводу

$$R_{\Sigma} = R_{\text{як}} + R_{\text{е.пр}} , \quad (2.20)$$

$$R_{\Sigma} = 0,38 + 0,2286 = 0,61 \text{ Ом} .$$

Електромагнітна постійна часу якорного кола електродвигуна:

$$T_a = \frac{L_{\text{як}}}{R_{\text{як}}}, \quad (2.21)$$

$$T_a = \frac{0,0138}{0,38} = 0,036 \text{ с}.$$

Електромагнітна постійна часу силового кола електроприводу:

$$T_e = \frac{L_{\Sigma}}{R_{\Sigma}}, \quad (2.22)$$

$$T_e = \frac{0,0144}{0,61} = 0,024 \text{ с}.$$

Конструктивна постійна двигуна

$$k_{\text{дв}} = \frac{pN}{2\pi a}, \quad (2.23)$$

де p – число пар полюсів;

N – число активних провідників обмотки якоря;

a – число пар паралельних гілок;

$$k_{\text{дв}} = \frac{2 \cdot 558}{2 \cdot 3,14 \cdot 1} = 177,7.$$

Постійна проти - ЕРС двигуна

$$c = k_{\text{дв}} \Phi_{\text{н}}, \quad (2.24)$$

$$c = k_{\text{дв}} \Phi_{\text{н}} = 177,7 \cdot 0,01326 = 2,37.$$

Електромеханічна постійна двигуна електроприводу

$$T_{\text{м}} = J_{\Sigma} \frac{R_{\Sigma}}{c^2}, \quad (2.25)$$

$$T_m = 3,46 \cdot \frac{0,561}{2,37^2} = 0,34 \text{ с.}$$

Коефіцієнт посилення двигуна

$$k_d = \frac{1}{c}, \quad (2.26)$$

$$k_d = \frac{1}{2,37} = 0,42.$$

Коефіцієнт посилення перетворювача тиристора:

$$k_{тп} = \frac{E_0}{U_m}, \quad (2.27)$$

де U_m - максимальне значення напруги порівняння, приймаємо $U_m = 10 \text{ В}$

$$k_{тп} = \frac{270}{10} = 27.$$

2.4 Вибір структури системи автоматичного управління

Система регулювання повинна забезпечувати необхідні механічні характеристики електроприводу як у статиці, так і в динаміці, гарантувати високу якість перехідних процесів, а також оптимізувати роботу електроприводу відповідно до заданих показників у різних режимах роботи механізму.

Досягнення високих показників регулювання можливе лише за умови використання замкнутих систем. Відповідно до рекомендацій [61], обрано двокоптурну систему підлеглого регулювання, яка включає внутрішній контур регулювання струму та зовнішній контур регулювання швидкості, замкнутий за ЕРС двигуна. Для покращення динамічних характеристик системи в контур регулювання струму додано підлеглий контур регулювання напруги тиристорного перетворювача.

2.5 Синтез послідовного коректуючого пристрою контуру струму

Відомо, що характер перехідного процесу визначається співвідношенням постійних часу елементів системи. Тому забезпечення високоякісних перехідних процесів пов'язане із досягненням необхідних значень цих співвідношень шляхом використання коригувальних регуляторів. У системах підлеглого регулювання кількість послідовно включених регуляторів відповідає кількості контурів.

Згідно з рекомендаціями [61], у контурі регулювання напруги тиристорного перетворювача використовується П-регулятор, а в контурі струму — ПІ-регулятор. На рис. 2.1 зображена спрощена принципова схема контуру струму з внутрішнім контуром напруги, а відповідна алгоритмічна схема наведена на рис. 2.2.

На рисунках прийняті такі позначення: ТП – тиристорний перетворювач; М – двигун постійного струму; ДТ, ДН – датчики струму і напруги відповідно; R_d – ділник напруги; $W_{pn}(p)$ – передавальні функції регуляторів напруги і струму відповідно.

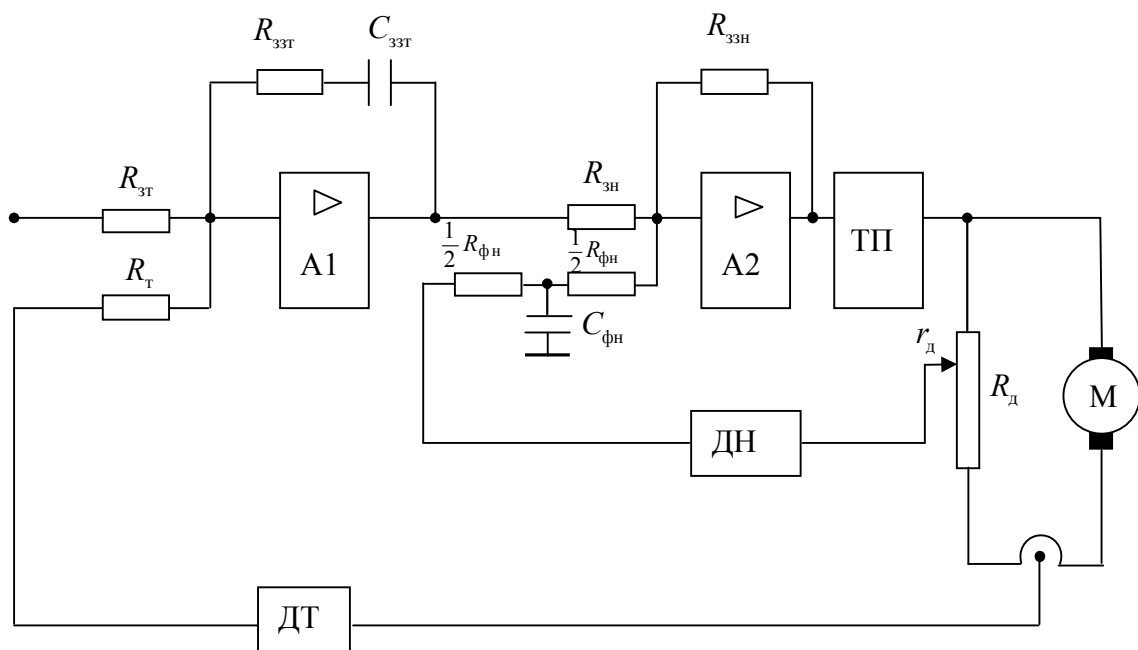


Рисунок 2.1 – Спрощена принципова схема контуру струму

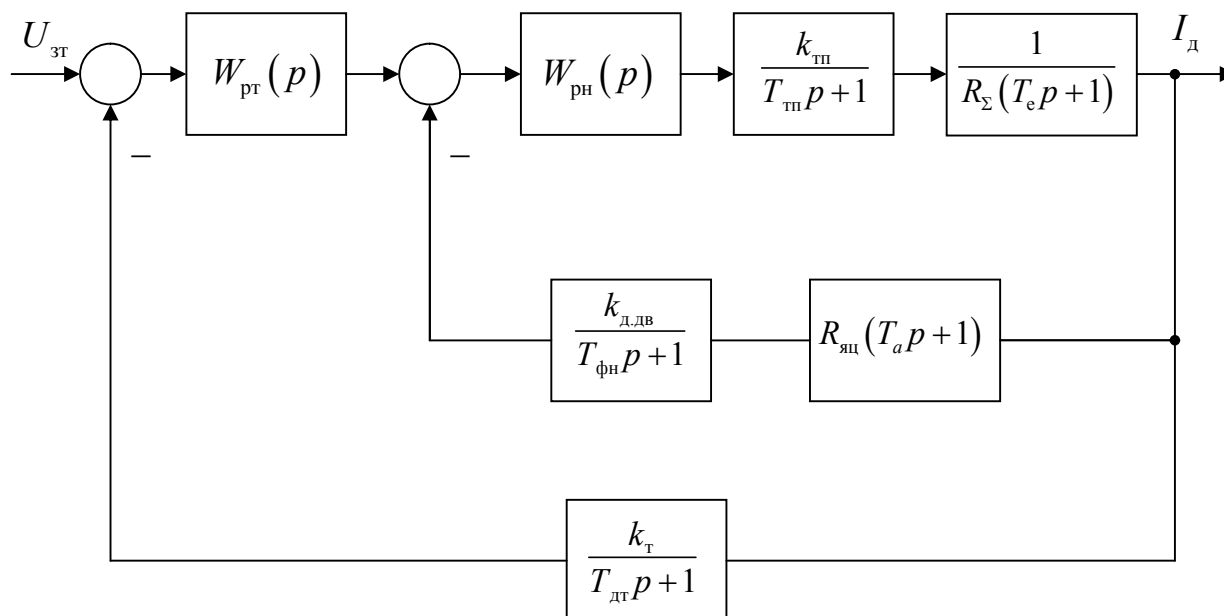


Рисунок 2.2 – Алгоритмічна схема контура струму

Для компенсації форсуючої ланки в ланцюзі зворотного зв'язку контура напруги використовується фільтр з постійною часу, яка вибирається рівною електромагнітною постійною часу якірного кола електродвигуна T_a .

Сигнал, пропорційний напрузі двигуна знімається з дільника напруги R_d . Коефіцієнт посилення дільника напруги визначається з умови отримання 24В на вході датчика напруги при максимальній напрузі двигуна, тобто

$$k_{\text{д.дв}} = \frac{24}{U_{\text{д.мак}}} . \quad (2.28)$$

Приймаючи в якості $U_{\text{д.мак}}$ номінальну напругу двигуна, отримаємо

$$k_{\text{д.дв}} = \frac{24}{220} = 0,109 .$$

Критичний коефіцієнт посилення контура напруги визначаються із співвідношення

$$k_{\text{н кр}} = \frac{\pi^2 T_{\text{е}} T_{\text{т}}}{2T_0^2 \left(1 - \frac{T_{\text{т}}}{T_{\text{е}}}\right)}, \quad (2.29)$$

де $T_{\text{т}} = 2T_{\mu\text{т}}$;

$T_{\mu\text{т}}$ – мала постійна, що не компенсується, контура струму

$$T_{\mu\text{т}} = T_{\text{дт}} + T_{\text{тп}}; \quad (2.30)$$

$T_{\text{дт}}$ – постійна часу датчика струму з фільтром на виході;

$T_{\text{тп}}$ – постійна часу тиристорного перетворювача.

Оскільки точне значення $T_{\mu\text{т}}$ визначити складно, то на практиці величиною $T_{\mu\text{т}}$ задаються в межах 0,004-0,01с. Приймаємо $T_{\mu\text{т}} = 0,005$ с ;

$$T_0 = \frac{2\pi}{m\omega_0}; \quad (2.31)$$

ω_0 – кутова частота живлячої мережі $\omega_0 = 2\pi f$.

$$T_0 = \frac{2\pi}{2 \cdot 2\pi \cdot 50} = \frac{1}{300} \text{ с},$$

$$k_{\text{н кр}} = \frac{3,14^2 \cdot 0,024 \cdot 0,01}{2 \cdot \left(\frac{1}{300}\right)^2 \cdot \left(1 - \frac{0,01}{0,024}\right)} = 76.$$

Приймаємо $k_{\text{н}} = 40$.

Сумарний коефіцієнт посилення контура напруги рівний

$$k_{\text{н}} = k_{\text{рп}} k_{\text{тп}} k_{\text{д дв}} \frac{R_{\text{як}}}{R_{\Sigma}}. \quad (2.32)$$

Коефіцієнт посилення П–регулятора контура напруги визначається як

$$k_{\text{рн}} = \frac{k_{\text{н}} R_{\Sigma}}{k_{\text{тп}} k_{\text{д дв}} R_{\text{як}}} , \quad (2.33)$$

$$k_{\text{рн}} = \frac{40 \cdot 0,061}{27 \cdot 0,1091 \cdot 0,38} = 21,6 .$$

Коефіцієнт посилення П-регулятора визначається відношенням опорів

$$k_{\text{рн}} = \frac{R_{\text{ззн}}}{R_{\text{зн}}} , \quad (2.34)$$

звідки

$$R_{\text{ззн}} = R_{\text{зн}} \cdot k_{\text{рн}} . \quad (2.35)$$

Приймаємо $R_{\text{зн}} = 2 \text{ кОм}$, тоді

$$R_{\text{ззн}} = 2 \cdot 21,6 = 43,2 \text{ кОм} .$$

Параметри фільтру в ланцюзі зворотного зв'язку по напрузі визначаються із співвідношення

$$\frac{R_{\text{фн}}}{4} C_{\text{фн}} = T_{\text{фн}} = T_a = 0,036 \text{ с}$$

Приймаємо $R_{\text{фн}} = R_{\text{зн}}$, тоді значення $C_{\text{фн}}$:

$$C_{\text{фн}} = \frac{4T_a}{R_{\text{фн}}} \quad (2.36)$$

$$C_{\text{фн}} = \frac{4 \cdot 0,036}{2 \cdot 10^3} = 7,2 \cdot 10^{-6} \text{ Ф} .$$

Передавальна функція замкнутого контура напруги визначається виразом:

$$W_{\text{н}}(p) = \frac{k_{\text{рн}} k_{\text{тп}}}{R_{\Sigma} (1 + k_{\text{н}}) \left(\frac{T_{\text{е}}}{1 + k_{\text{н}}} p + 1 \right) (T_{\text{тп}} p + 1)} . \quad (2.37)$$

Алгоритмічна схема контура струму з урахуванням (2.37) показана на рис. 2.3. На рис. 2.4 показана перетворена алгоритмічна схема з одиничним зворотним зв'язком. Форсуюча ланка $(T_{\text{дт}} p + 1)$ на вході контура не показана, оскільки його вплив на динамічні характеристики контура струму у край тріхи зважаючи на тріхи постійної часу $T_{\text{дт}}$.

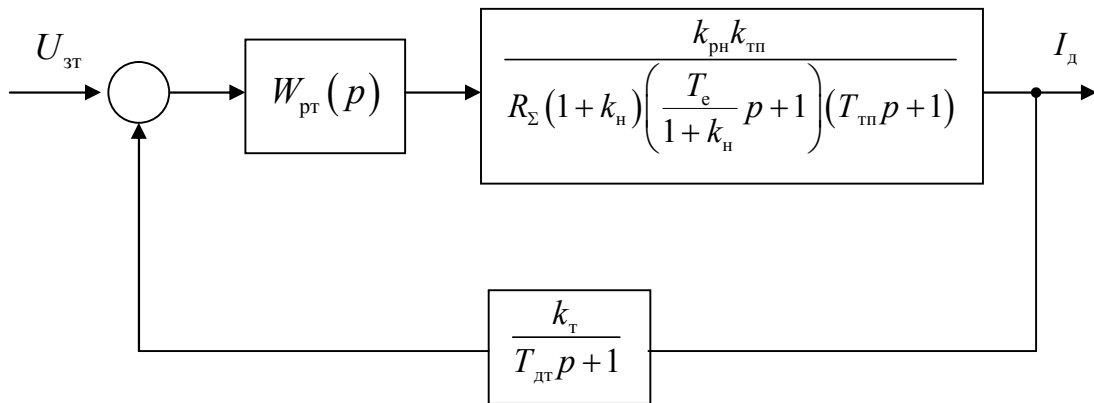


Рисунок 2.3 – Алгоритмічна схема контура струму
з перетвореним контуром напруги

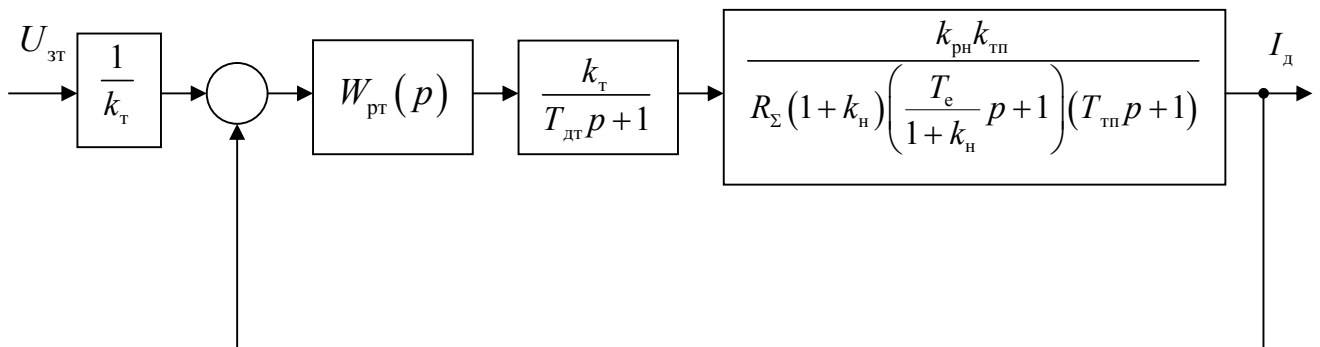


Рисунок 2.4 – Алгоритмічна схема контура струму
з одиничним зворотним зв'язком

Оптимізація контура струму по модульному критерію досягається шляхом включення в контур послідовного пристрою, що коректує, з передавальною функцією

$$W_{\text{рт}}(p) = \frac{T_{\text{рт}}p + 1}{T'_{\text{рт}}p} = \frac{\left(\frac{T_{\text{е}}}{1 + k_{\text{н}}} p + 1 \right) R_{\Sigma} (1 + k_{\text{н}})}{2T_{\mu\text{т}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}} p}. \quad (2.38)$$

Із співвідношення (2.38) визначаємо параметри регулятора.

$$T_{\text{рт}} = C_{\text{ззт}} R_{\text{ззт}} = \frac{T_{\text{е}}}{1 + k_{\text{н}}}. \quad (2.39)$$

Вибираємо величину ємності $C_{\text{ззт}} = 0,5 \cdot 10^{-6} \text{ Ф}$, тоді

$$R_{\text{ззт}} = \frac{T_{\text{е}}}{C_{\text{ззт}} (1 + k_{\text{н}})}, \quad (2.40)$$

$$R_{\text{ззт}} = \frac{0,024}{(1 + 40) \cdot 0,5 \cdot 10^{-6}} = 11,7 \cdot 10^3 \text{ Ом}.$$

У виразі (2.38) $k_{\text{т}}$ – коефіцієнт посилення ланцюга зворотного зв'язку по струму

$$k_{\text{т}} = k_{\text{ш}} k_{\text{дт}} \frac{R_{\text{зт}}}{R_{\text{т}}}, \quad (2.41)$$

де $k_{\text{ш}} = \frac{U_{\text{шн}}}{I_{\text{шн}}}$ – коефіцієнт посилення шунта;

$U_{\text{шн}}, I_{\text{шн}}$ – номінальна напруга і струм шунта;

$k_{\text{дт}} = \frac{U_{\text{шн}}}{U_{\text{дтн}}}$ – коефіцієнт посилення датчика струму;

$U_{\text{дтн}}$ – номінальна напруга на виході датчика струму.

Коефіцієнт посилення датчика струму вибираємо так, щоб максимальному робочому перевантаженню двигуна виконувалась нерівність:

$$U_{\text{дт max}} = k_{\text{дт}} k_{\text{ш}} I_{\text{max}} \frac{R_{\text{зт}}}{R_{\text{т}}} \leq 24 \text{ В} . \quad (2.42)$$

При рівності опорів $R_{\text{зт}}$ і $R_{\text{т}}$ вибране значення $k_{\text{дт}}$ не повинне перевищувати

$$k_{\text{дт}} = \frac{24}{k_{\text{ш}} I_{\text{max}}} = \frac{24}{k_{\text{ш}} k_{\text{рп max}} I_{\text{н}}} , \quad (2.43)$$

де $k_{\text{рп max}}$ – коефіцієнт максимального робочого перевантаження двигуна, $k_{\text{рп max}} = 2,5$.

Приймаємо шунт з номінальною напругою $U_{\text{шн}} = 0,075 \text{ В}$. В цьому випадку шунт має максимально можливий для даного двигуна передавальний коефіцієнт, завдяки чому забезпечується якнайкраща перешкодозахисна каналу вимірювання струму. Як датчик струму приймаємо операційний підсилювач з фіксованим значенням коефіцієнта посилення $k_{\text{дт}} = 320$

$$k_{\text{ш}} = \frac{75 \cdot 10^{-3}}{2 \cdot 57} = 0,658 \cdot 10^{-3} \text{ В / А} .$$

$$k_{\text{т}} = 0,658 \cdot 10^{-3} \cdot 320 = 0,231 \text{ В / А} .$$

Величину опору $R_{\text{зт}}$ визначаємо з умови

$$T'_{\text{рт}} = C_{\text{ззт}} R_{\text{зт}} = \frac{2T_{\mu\text{т}} k_{\mu\text{т}} k_{\text{тп}} k_{\text{т}}}{R_{\Sigma} (1 + k_{\text{н}})} , \quad (2.44)$$

(дане співвідношення виходить з (2.38)).

Приймаємо $C_{\text{ззт}} = 0,5 \cdot 10^{-6} \text{ Ф}$. З (2.16) отримуємо

$$R_{\text{зт}} = \frac{2T_{\mu\text{т}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}}}{C_{\text{ззт}} R_{\Sigma} (1 + k_{\text{н}})} , \quad (2.45)$$

$$R_{\text{зт}} = \frac{2 \cdot 0,005 \cdot 21,6 \cdot 27 \cdot 0,0231}{0,61 \cdot (1 + 40) \cdot 0,5 \cdot 10^{-6}} = 10,8 \cdot 10^3 \text{ Ом},$$

$$R_{\text{т}} = R_{\text{зт}} = 10,8 \cdot 10^3 \text{ Ом}.$$

2.6 Синтез послідовного корегуючого пристрою контура ЕРС

Відповідно до рекомендацій [61] приймаємо двоконтурну однократно інтегруючу систему. Контур швидкості замикаємо по ЕРС двигуна. Принципова схема системи управління показана на рис. 2.5. Підсумовування сигналів пропорційних напрузі і падінню напруги в якірному ланцюзі двигуна, здійснюється на вході регулятора ЕРС РЕ. Перший з вказаних сигналів поступає від датчика напруги ДН, призначеної для гальванічного розділення ланцюгів регулювання і головного ланцюга. Сигнал струмової компенсації подається від датчика струму ДТ контуру регулювання струму.

Оскільки в ланцюзі зворотного зв'язку по ЕРС є аперіодична ланка $\frac{1}{T_a p + 1}$, то це може викликати перерегулювання струму при зміні задаючої дії. Для усунення вказаного явища на вхід системи необхідно включити таку ж ланку.

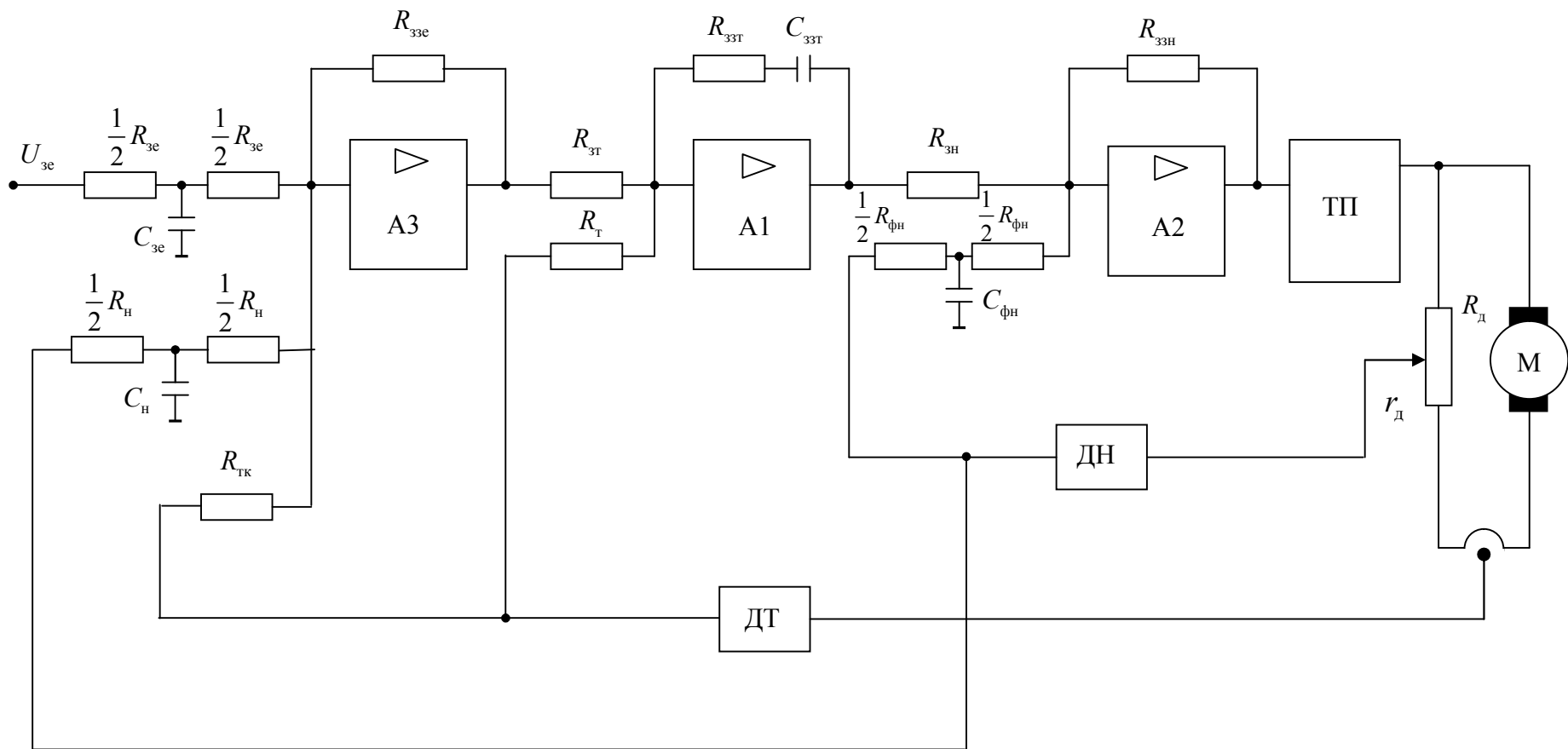


Рисунок 2.5 – Принципова схема системи управління із зворотним зв'язком по ЕРС

На рис. 2.6 показана алгоритмічна схема системи.

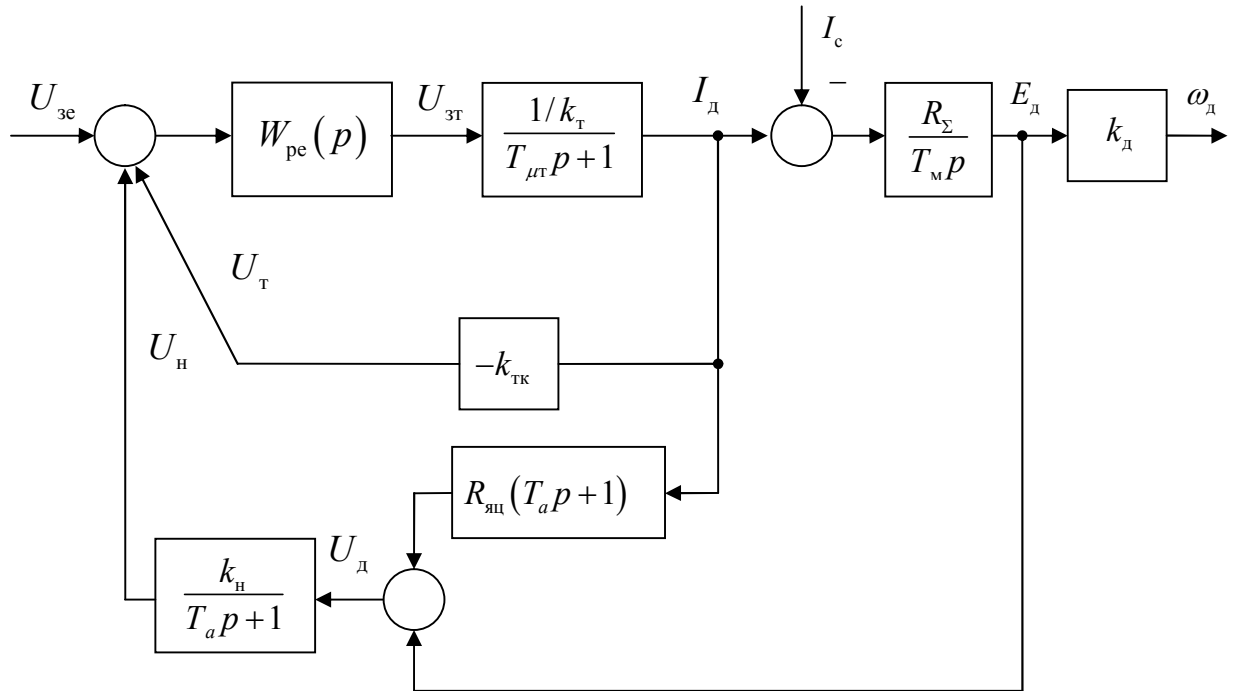


Рисунок 2.6– Алгоритмічна схема системи управління із зворотним зв'язком по ЕРС

Оптимізований по модульному критерію контур струму наданий аперіодичною ланкою

$$W_T(p) = \frac{1/k_T}{2T_{\mu T}^2 p^2 + 1}, \quad (2.46)$$

яке виходить при зневазі доданком $2T_{\mu T}^2 p^2$ в знаменнику передавальної функції контура струму.

Коефіцієнт посилення ланцюга струмової компенсації вибирається таким чином:

$$k_{TK} = k_H R_{ЯК}, \quad (2.47)$$

де k_H – коефіцієнт посилення ланцюга зворотного зв'язку по напрузі.

При вказаному вище виборі коефіцієнта k_{TK} на вході регулятора ЕРС виходить сигнал, пропорційний ЕРС.

$$U_{\text{н}} - U_{\text{т}} = \frac{k_{\text{н}}}{T_a p + 1} E_{\text{д}}. \quad (2.48)$$

Алгоритмічна схема, показана на рис. 2.6 шляхом еквівалентних перетворень перетвориться в схему, показану на рис. 2.7. На рис.2.8 показана розрахункова схема з одиничним зворотним зв'язком.

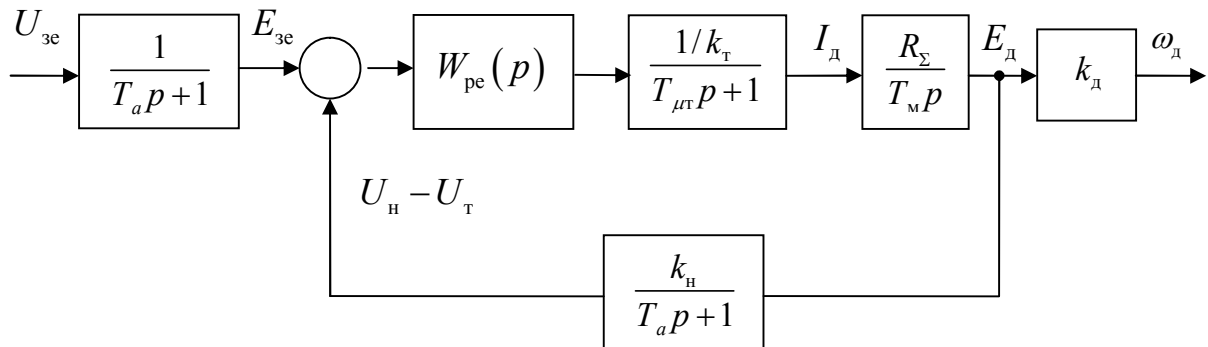


Рисунок 2.7 – Перетворена алгоритмічна схема системи регулювання

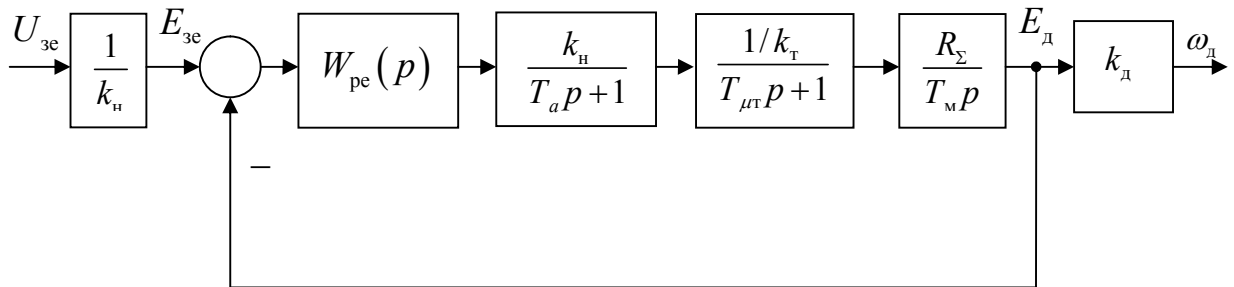


Рисунок 2.8 – Розрахункова алгоритмічна схема системи з одиничним зворотним зв'язком

Для оптимізації контура ЕРС по модульному критерію необхідно використовувати П-регулятор з передавальною функцією

$$W_{\text{ре}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{2 T_{\mu\epsilon} k_{\text{н}} R_{\Sigma}}, \quad (2.49)$$

де $T_{\mu\epsilon}$ – мала постійна, що не компенсується, контура ЕРС.

$$T_{\mu e} = 2T_{\mu t} + T_a, \quad (2.50)$$

$$T_{\mu e} = 2 \cdot 0,005 + 0,036 = 0,046 \text{ с}.$$

Оскільки до складу малої некомпенсованою постійною контура ЕРС входить T_a , то швидкодія і статична точність регулювання проектованої системи в $\frac{T_{\mu e}}{T_{\mu t}}$ раз менше, ніж у системи із зворотним зв'язком за швидкістю.

Зіставляючи передавальну функцію (2.49) з передавальною функцією типового П-регулятора, отримуємо рівняння.

$$\frac{R_{зе}}{R_{зе}} = \frac{T_M k_T}{2T_{\mu e} k_H R_{\Sigma}}. \quad (2.51)$$

Передавальний коефіцієнт ланцюга зворотного зв'язку по напрузі рівний

$$k_H = \frac{U_{зе \max}}{E_{дв \max}}, \quad (2.52)$$

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

$U_{зе \max}$ – напруга завдання відповідна $E_{дв \max}$; приймаємо $U_{зе \max} = 24 \text{ В}$.

У якості $E_{дв \max}$ приймемо ЕРС в режимі ідеального холостого ходу, тобто

$$E_{дв \max} = U_H = 220 \text{ В} \text{ тоді}$$

$$k_H = \frac{24}{220} = 0,1091.$$

Коефіцієнт посилення ланцюга зворотного зв'язку по напрузі і ланцюга токової компенсації можна визначити із співвідношень:

$$k_H = k_{д.дв} \cdot \frac{R_{зе}}{R_H}; \quad (2.53)$$

$$k_{ТК} = k_{ш} k_{дт} \frac{R_{зе}}{R_{ТК}}. \quad (2.54)$$

де $k_{\text{д.дв}}$ – коефіцієнт дільника напруги визначається з умови отримання сигналу 24В на вході датчика напруги ДН при максимальній напрузі двигуна

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

Керуючись рекомендаціями [73], величиною $R_{\text{зе}} = R_{\text{едс}}$ задаються в межах $(50 \div 100) \text{ кОм}$. Приймаємо $R_{\text{зе}} = R_{\text{едс}} = 30 \cdot 10^3 \text{ Ом}$, тоді з (2.53) визначаємо $R_{\text{н}}$.

$$R_{\text{н}} = \frac{k_{\text{д.дв}} \cdot R_{\text{зе}}}{k_{\text{н}}}; \quad (2.55)$$

$$R_{\text{н}} = \frac{0,1091 \cdot 30}{0,1091} = 30 \text{ кОм}.$$

Згідно (2.47)

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}}; \quad (2.56)$$

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

З рівняння (2.54) отримуємо

$$R_{\text{тк}} = \frac{k_{\text{ш}} k_{\text{дт}} R_{\text{зе}}}{k_{\text{тк}}}; \quad (2.57)$$

$$R_{\text{тк}} = \frac{0,658 \cdot 10^{-3} \cdot 320 \cdot 30 \cdot 10^3}{0,04} = 157,9 \cdot 10^3 \text{ Ом}.$$

З виразу (2.51) визначаємо величину $R_{\text{ззе}}$.

$$R_{\text{ззе}} = \frac{T_{\text{м}} k_{\text{т}} R_{\text{зе}}}{2T_{\text{мє}} k_{\text{н}} R_{\Sigma}}; \quad (2.58)$$

$$R_{\text{ззе}} = \frac{0,34 \cdot 0,231 \cdot 30 \cdot 10^3}{2 \cdot 0,046 \cdot 0,1091 \cdot 0,61} = 38,1 \cdot 10^3 \text{ Ом}.$$

Параметри фільтрів на вході системи і в ланцюзі зворотного зв'язку по напрузі визначаються з рівнянь

$$\frac{R_{зе}}{4} \cdot C_{зе} = \frac{R_{н}}{4} \cdot C_{н} = T_a; \quad (2.59)$$

$$C_{н} = \frac{4 \cdot T_a}{R_{н}}, \quad (2.60)$$

$$C_{зе} = \frac{4 \cdot T_a}{R_{зе}}, \quad (2.61)$$

$$C_{зе} = \frac{4 \cdot 0,036}{30 \cdot 10^3} = 4,8 \cdot 10^{-6} \text{ Ф}.$$

$$C_{н} = \frac{4 \cdot 0,036}{30 \cdot 10^3} = 4,8 \cdot 10^{-6} \text{ Ф}.$$

Запишемо передавальні функції системи, оптимізованої по модульному критерію, для швидкості і току по задавальній і збурюючій діях:

$$W_{\omega_d, U_{зе}}(p) = \frac{\omega_d(p)}{U_{зе}(p)} = \frac{k_d / k_{н}}{2T_{\mu\epsilon}^2 p^2 + 2T_{\mu\epsilon} p + 1}; \quad (2.62)$$

$$W_{\omega_d, I_{ст}} = \frac{\omega_d(p)}{I_{ст}(p)} = \frac{-2T_{\mu\epsilon} R_{\Sigma} k_d / T_{м}(1 + T_{\mu\epsilon} p)}{2T_{\mu\epsilon}^2 p^2 + 2T_{\mu\epsilon} p + 1}; \quad (2.63)$$

$$W_{I_d, U_{зе}} = \frac{I_d(p)}{U_{зе}(p)} = \frac{T_{м}}{R_{\Sigma} k_{н}} \cdot \frac{p}{2T_{\mu\epsilon}^2 p^2 + 2T_{\mu\epsilon} p + 1}; \quad (2.64)$$

$$W_{I_d, I_{ст}} = \frac{I_d(p)}{I_{ст}(p)} = \frac{1}{2T_{\mu\epsilon}^2 p^2 + 2T_{\mu\epsilon} p + 1}. \quad (2.65)$$

Алгоритмічна схема системи показана на рис. 2.9.

2.4. Дослідження динамічних характеристик системи

Для побудови перехідних процесів системи на ЕОМ розроблена схема моделі системи (рис.2.10). Схема побудована з використанням пакету MATLAB в режимі SIMULINK. Перехідні процеси досліджувалися також по векторно-матричному рівнянню системи, порядок складання якого дається в подальших розділах і по передатних функціях системи. На рисунку 2.11 приведені графіки перехідних процесів швидкості ω_d і струму I_d електроприводу по задаючій дії, а на рисунку 2.12 показані графіки перехідних процесів швидкості і струму електроприводу по збурюючій дії.

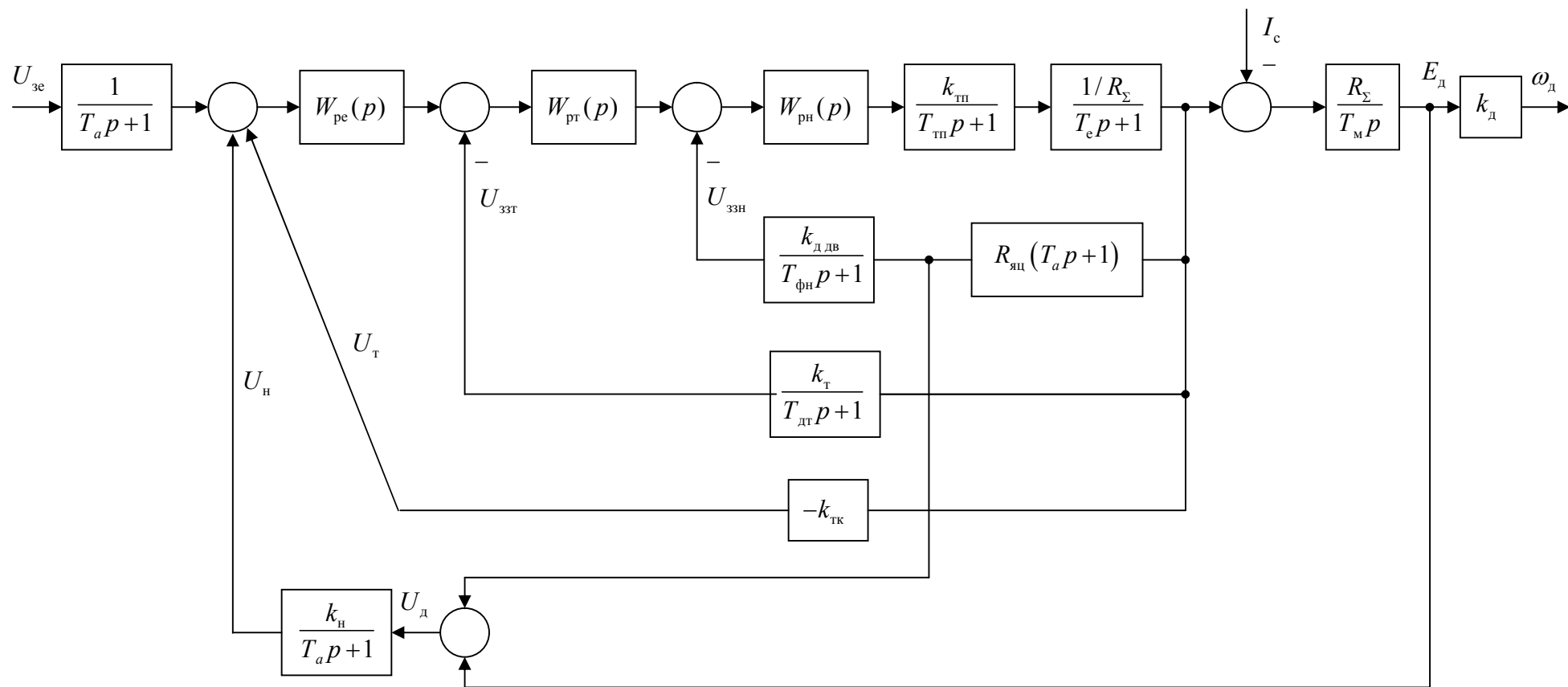


Рисунок 2.9 – Алгоритмічна схема системи

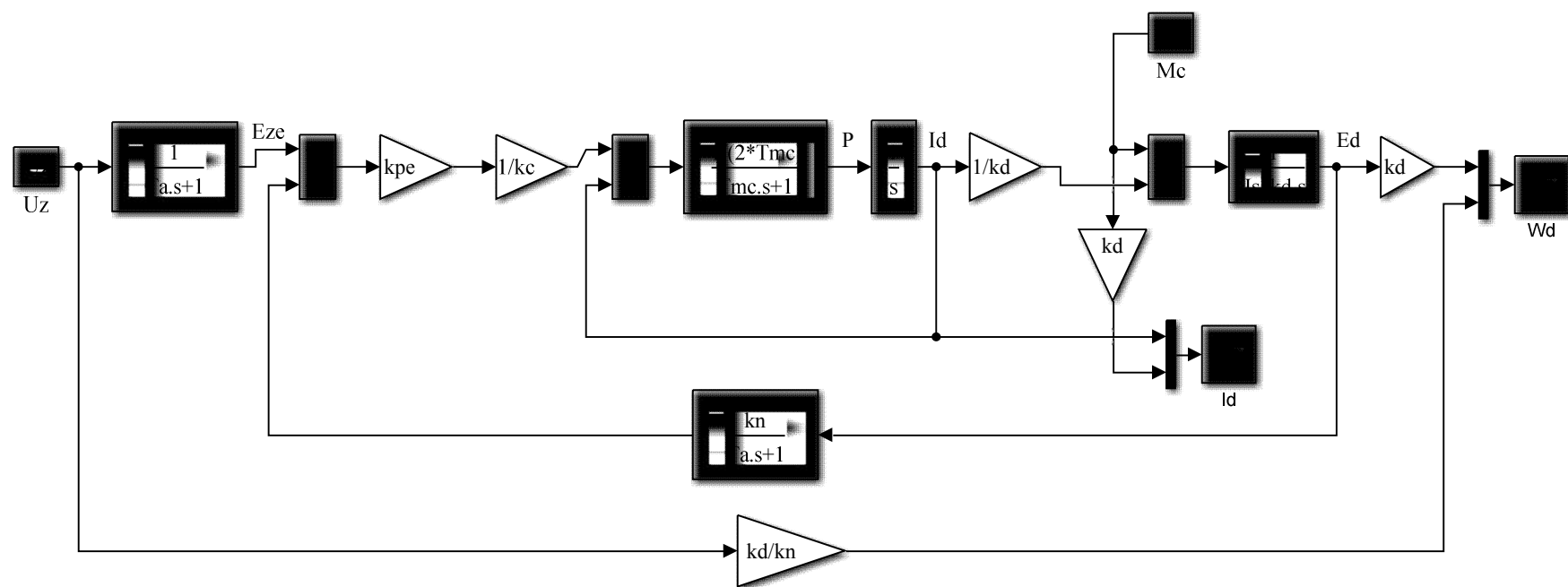


Рисунок 2.10 – Схема моделі системи, розроблена на ЕОМ

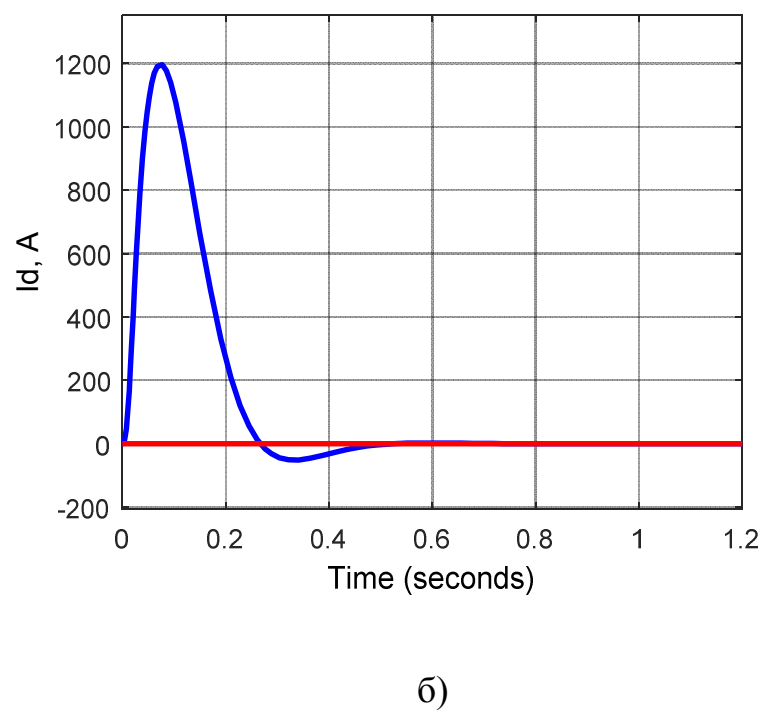
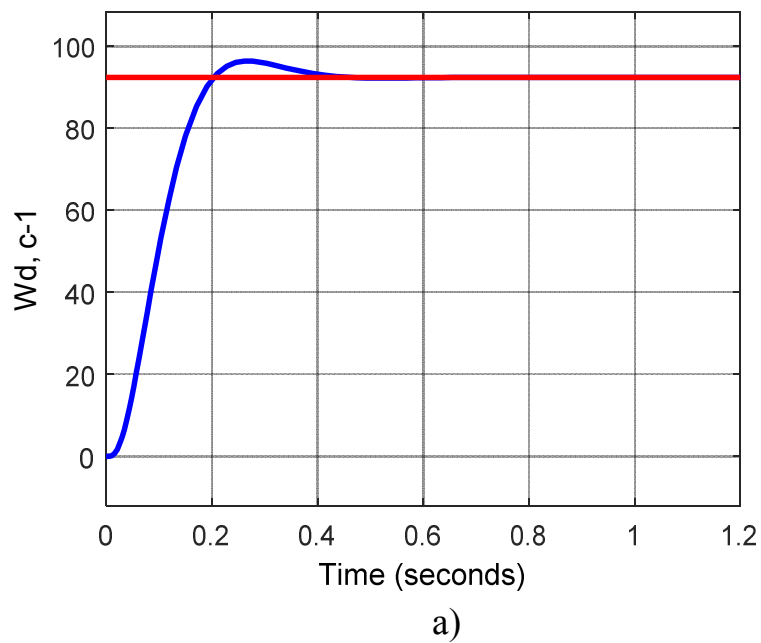


Рисунок 2.11 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ (а)
і струму $I_d = f(t)$ (б) по задаючій дії

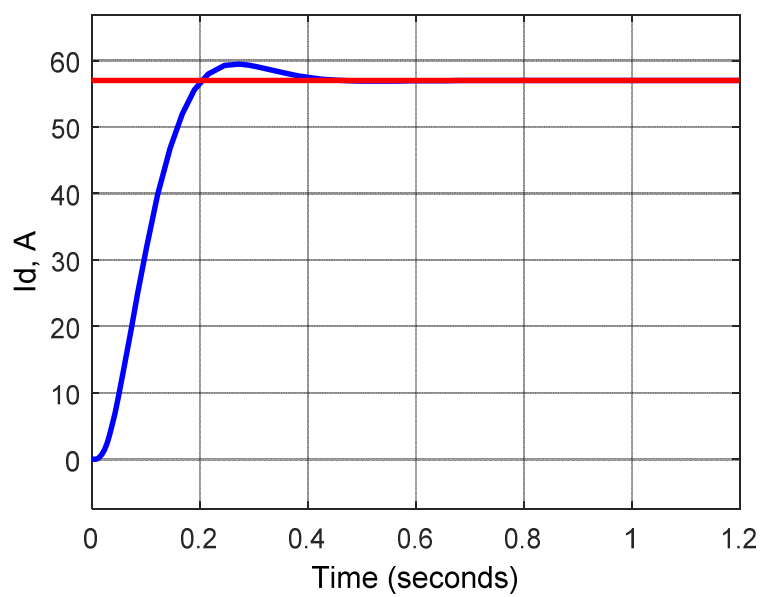
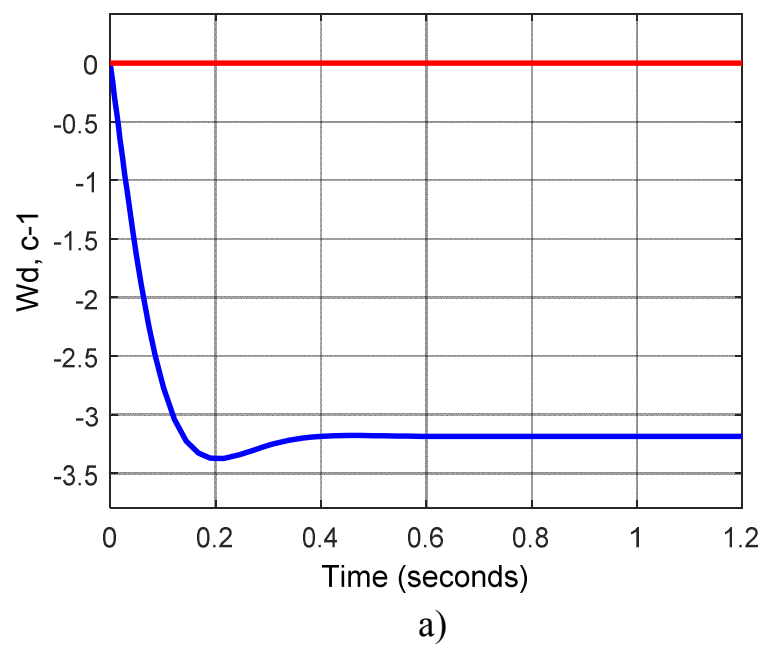


Рисунок 2.12 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ (а)
і струму $I_d = f(t)$ (б) по збурюючій дії:

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На підставі аналізу вимог до систем електроприводів основних механізмів мостових кранів для електроприводу механізму пересування візка вибраний електропривод постійного струму, виконаний за системою ТП-Д. Для управління координатами електроприводу застосована система підлеглого регулювання.

2. Виконано вибір електродвигуна і основного електроустаткування силового кола 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.

3. Виконано розрахунок параметрів силового кола електроприводу, що використовуються при розрахунку системи автоматичного управління.

4. Обрана конфігурація системи автоматичного управління, яка має забезпечувати потрібні механічні властивості електроприводу як у статичному, так і в динамічному режимах, включаючи високу якість у перехідних процесах і оптимізацію роботи електроприводу в різних режимах роботи механізму. Тому для електроприводу механізму рольганга була обрана двоконтурна система підлеглого регулювання, де один контур відповідає за струм, а інший - за швидкість, при цьому система замкнена по ЕРС електродвигуна.

5. Проведений синтез послідовних коригуючих пристроїв для окремих контурів. Оскільки характер перехідних процесів залежить від відношення постійних часу елементів системи, для досягнення високоякісних перехідних процесів використовувалися регулятори, що коригують ці відношення. Для контура напруги синтезований П-регулятор. Для контуру струму синтезовано ПІ-регулятор, а для контура ЕРС - П-регулятор, унаслідок чого контур струму і ЕРС оптимізовані по модульному критерію..

6. 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 перехідних процесів в системі по задаючій і збурюючій діях. Перехідні процеси досліджувалися також по векторно-матричному рівнянню системи і з використанням передатних функцій системи.

РОЗДІЛ 3

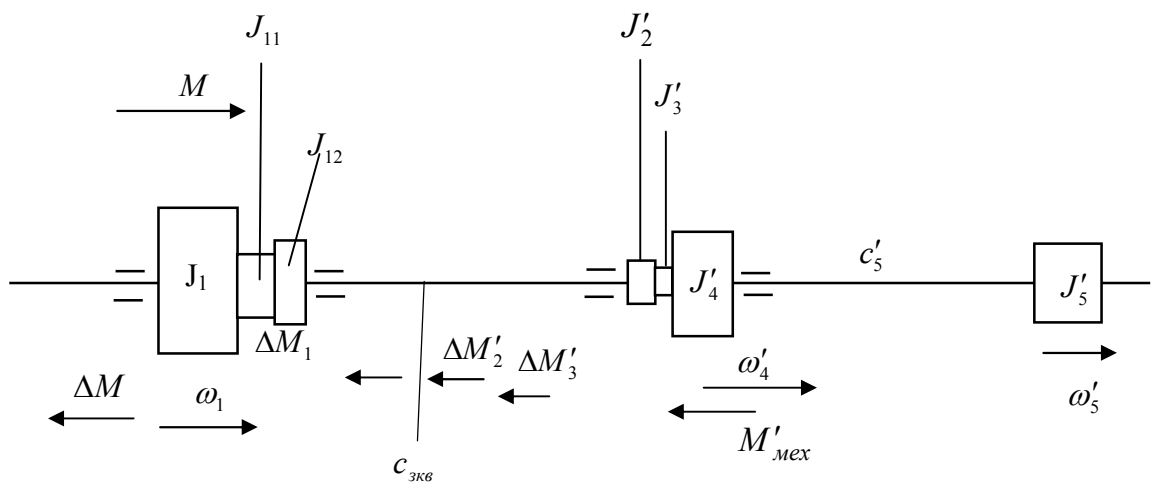
ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПОПЕРЕЧНИХ РОЗГОЙДУВАНЬ ВАНТАЖУ

3.1 Розрахункові схеми механічної частини електроприводу

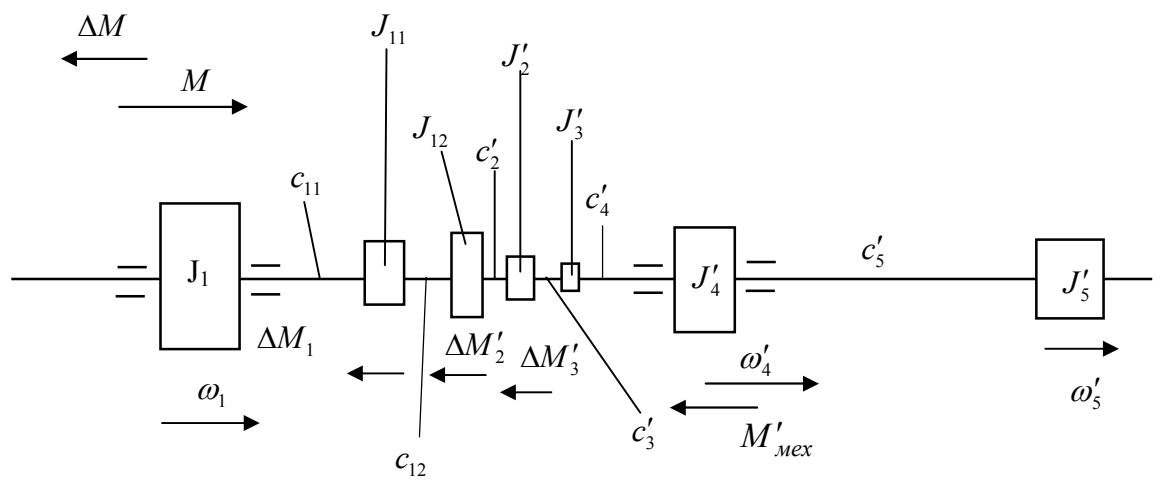
Механічна частина системи, що розглядається вище, розглядалася як жорстка приведена ланка, при цьому нехтували пружні зв'язки в системі, що, як показано в [73], дає достатньо далекі від істини уявлення про динаміку системи.

Механізм пересування візка підйомного крана є класичним прикладом, коли механічна частина електроприводу приводяться до трьохмасової пружної механічної системи.

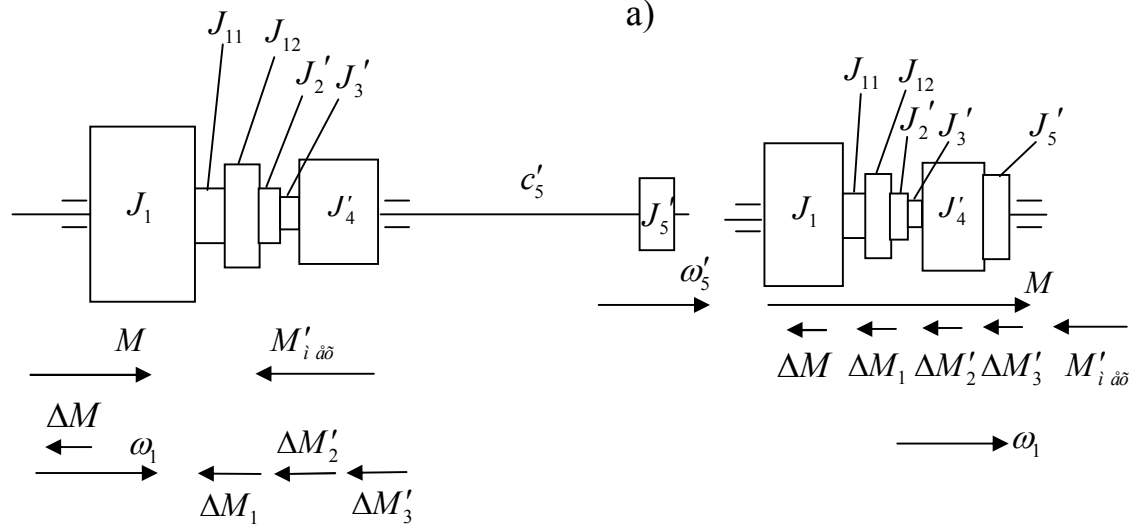
Приведена до швидкості двигуна розрахункова схема цього механізму представлена на рис.3.1 а. Тут J_1, J_{11}, J_{12} - моменти інерції відповідно ротора двигуна, що веде напівмуфти, відомої напівмуфти з вхідним валом редуктора; c_{11}, c_{12} - жорсткості зв'язків між цими масами; J'_2 - приведений момент інерції проміжного валу редуктора; c'_2, c'_3 - приведені жорсткості його зв'язків з вхідним і вихідним валом; J'_3, J'_4 - приведені моменти інерції відповідно вихідного валу редуктора і поступально рухомих мас візка m_1 ; c'_4 - приведена жорсткість зв'язку між ними; J'_5 - приведений момент інерції поступально рухомого вантажу m_2 ; c'_5 - приведена жорсткість зв'язку між масами m_1, m_2 ; $\Delta M'_2, \Delta M'_3$ - приведені моменти в редукторі; $M'_{\text{мех}}$ - приведений момент сили опору руху візка $F_{\text{мех}}$.



б)



а)



в)

Рисунок 3.1 – Розрахункові механічні схеми електроприводу пересування візка

Розглядаючи розрахункову схему на рис.3.1 а, можна бачити в ній три істотну масу: ротора двигуна, візка J'_4 і вантажу J'_5 . Решта мас незначна і без істотної погрішності врахована збільшенням моментів інерції J_1, J'_4 , як показано на рис.3.1 б, а жорсткості зв'язків між ними замінені еквівалентним зв'язком $c_{\text{екв}}$. Отримана трьохмасова система відображає основні особливості руху даного механізму і дозволяє отримати уявлення про закони зміни швидкостей трьох істотних мас системи $\omega_1(t)$, $\omega_4(t)$, $\omega_5(t)$, достатньо близьких до істинних.

Залежно від конкретних цілей дослідження схема може бути без великої погрішності додатково спрощена. Жорсткість зв'язку вантажу, що розгойдується, з візком $c'_5 = c_{\text{пр}}$ зазвичай на порядок або на два порядки менше, ніж еквівалентна жорсткість передач $c_{\text{екв}}$, тому при розгляді коливальних процесів в передачах можна звести схему до двомасової, прийнявши $c'_5 \approx 0$, а при дослідженні процесів розгойдування вантажу прийняти $c_{\text{екв}} = \infty$ і використовувати двохмасову схему, показану на рис.3.1 в. Нарешті, за умов, які розглядаються нижчим, можна нехтувати пружними зв'язками в системі і звести її до жорсткої приведенної ланки, представленої на рис.3.1 г. Цей варіант розрахункової механічної схеми для механізмів з гнучкими зв'язками дає найбільш далекі від істини уявлення про динамічну систему, але при рішенні ряду завдань його використання завдяки простоті виявляється найбільш переважним.

3.2 Математична модель двомасової системи

Розглянемо двомасову систему підлеглого регулювання швидкості з контуром струму, настроєним по модульному критерію і контуро ЕРС, настроєним по симетричному критерію. Алгоритмічна схема двомасової системи показана на рис. 3.2.

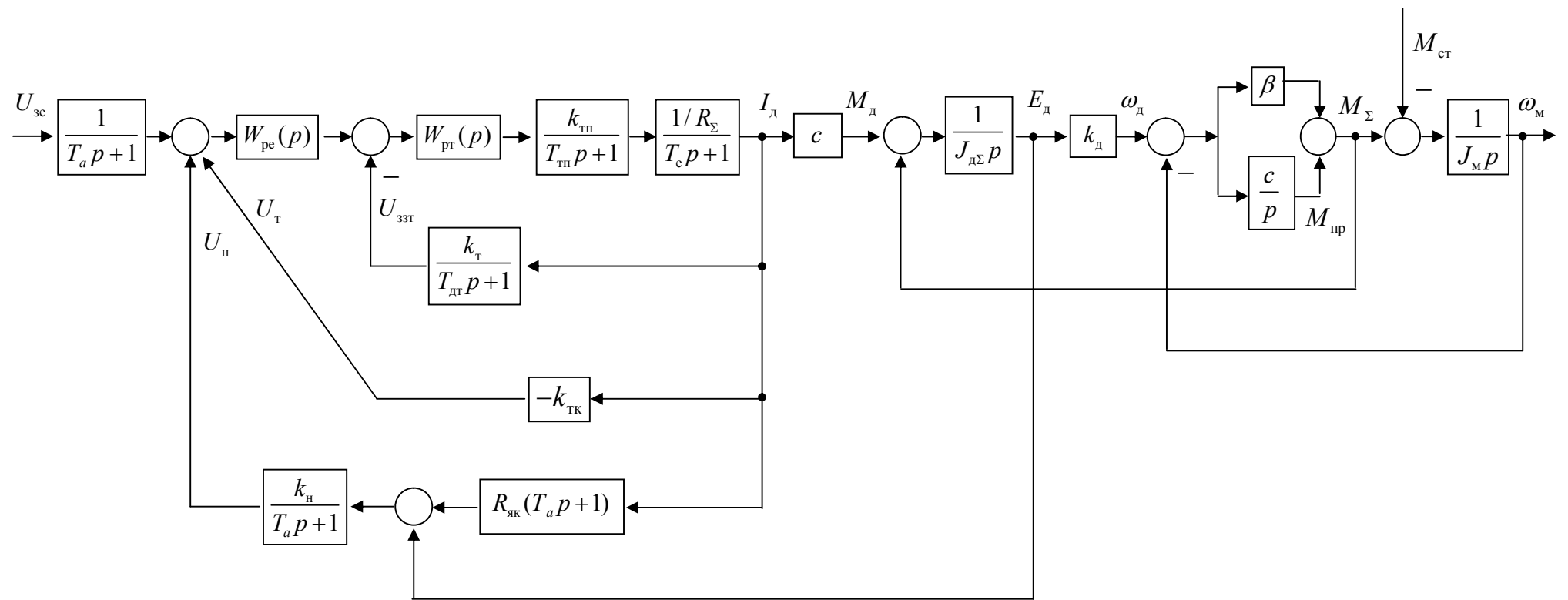


Рисунок 3.2 - Алгоритмічна схема двомасової системи

Динаміка механічної частини двомасової системи описується рівняннями

$$\begin{cases} J_{\Sigma 1} \frac{d\omega_d}{dt} = M_d - M_{\Sigma}; \\ M_{\Sigma} = M_{\text{пр}} + \beta(\omega_d - \omega_m); \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_d - \omega_m); \\ J_m \frac{d\omega_m}{dt} = M_{\Sigma} - M_c. \end{cases} \quad (3.1)$$

де $J_{\Sigma 1}$ - сумарний момент інерції провідної маси.

$$J_{\Sigma 1} = 1,2J_{\text{дв}} + J'_4 = 1,2J_{\text{дв}} + \frac{m_B V_B^2}{\omega_H^2}; \quad (3.2)$$

J_m - приведений момент інерції візка з вантажем (момент інерції механізму);

$$J_m = J'_5 = \frac{(G + m_0)V_B^2}{\omega_H^2}; \quad (3.3)$$

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

$$M_{\Sigma} = M_{\text{пр}} + \beta(\omega_d - \omega_m), \quad (3.4)$$

β – коефіцієнт внутрішнього в'язкого тертя каната між масами; у розрахунках приймаємо $\beta = 0$;

ω_d, ω_m - кутова швидкість двигуна і механізму відповідно;

c_{12} - коефіцієнт жорсткості, в даній системі

$$c_{12} = c_{\text{пр}} = \frac{(G + m_0) \cdot g \cdot V_{\text{в}}^2}{H_{\text{п}} \omega_{\text{н}}^2}; \quad (3.5)$$

$M_{\text{с}}$ - момент статичного навантаження, прикладений до другої (відо-
мою) маси, в даному завданні розгойдування вантажу

$M_{\text{д}}$ - момент, що розвивається двигуном.

Запишемо систему рівнянь (3.1) в наступному вигляді:

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

Одномасова система підлеглого регулювання швидкості з контуром струму, настроєним по модульному критерію і контуро ЕРС, настроєним по симетричному критерію описується наступною системою диференціальних рівнянь, записаних у формі Коші:

$$\begin{cases} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\Gamma}^2} M_{\text{д}} - \frac{1}{T_{\mu\Gamma}} P; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}. \end{cases} \quad (3.7)$$

При записі системи (3.7) прийняті наступні позначення.

P - ривок електроприводу;

$k_{\text{пе}}$ - коефіцієнти підсилення регулятора ЕРС; у попередніх розділах була отримана передатна функція П-регулятора ЕРС

$$W_{\text{пе}}(p) = k_{\text{пе}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu E} k_{\text{н}} R_{\Sigma}}. \quad (3.8)$$

При записі системи (3.7) замість $T_{\text{м}}$ записано відповідний вираз

$$T_{\text{м}} = \frac{R_{\Sigma} \cdot J_{\text{д}\Sigma}}{c^2} = R_{\Sigma} J_{\text{д}\Sigma} \cdot k_{\text{д}}^2; \quad k_{\text{д}} = 1/c.$$

Об'єднаємо системи (3.6 і (3.7), одержимо наступну систему рівнянь стану двомасової системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = -\frac{\beta}{J_{\text{м}}} \omega_{\text{м}} + \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = -c_{12} \omega_{\text{м}} - c_{12} \omega_{\text{д}}; \\ \frac{d\omega_{\text{д}}}{dt} = -\frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}} + \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = -\frac{k_{\text{д}}}{2T_{\mu T}^2} M_{\text{д}} - \frac{1}{T_{\mu T}} P + \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_{\text{т}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu T}^2 k_{\text{т}}} U_{\text{е}}; \\ \frac{dE_{\text{зе}}}{dt} = -\frac{1}{T_{\text{а}}} E_{\text{зе}} + \frac{1}{T_{\text{а}}} U_{\text{зе}}; \\ \frac{dU_{\text{е}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{е}}; \end{array} \right. \quad (3.9)$$

Алгоритмічна схема двомасової системи, що відповідає рівнянням (3.9), показана на рис. (3.3).

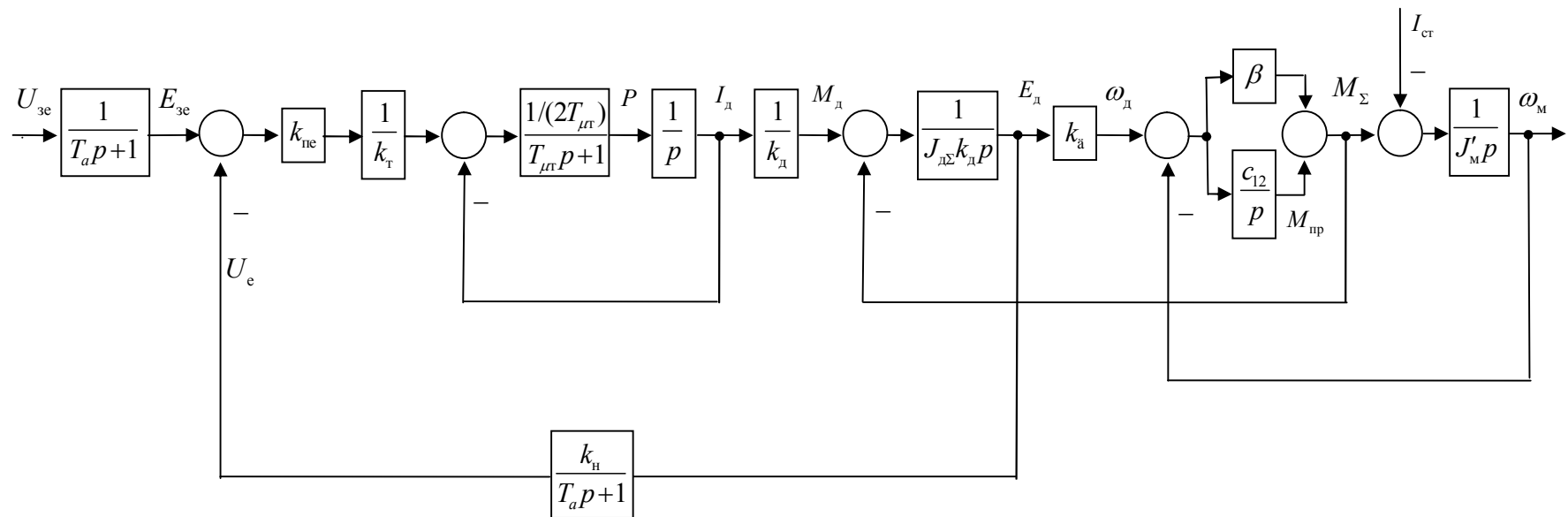


Рисунок 3.3 – Спрощена алгоритмічна схема двомасової системи

На схемі оптимізований по модульному критерію контур струму представлений у вигляді послідовного з'єднання аперіодичної і інтегруючої ланки, охоплених одиничним зворотним зв'язком. Сигнал на виході першої ланки пропорційний ривку електроприводу P .

Запишемо результуючу систему у векторному – матричній формі, доповнивши її рівнянням виходу системи.

Введемо вектор стану системи $\vec{X}(t)$ і вектор управління $\vec{U}(t)$

$$\vec{X}(t) = \{ \omega_m(t), M_{пр}(t), \omega_d(t), M_d(t), P(t), E_{зе}(t), U_e(t) \}^T \quad (3.10)$$

$$\vec{U}(t) = \{ U_{зе}(t), M_{ст}(t) \}^T \quad (3.11)$$

де T - знак транспонування.

Рівняння стану системи і рівняння виходу має вид

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.12)$$

де $\mathbf{A}$ – матриця стану;

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ - вектор виходу.

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

$$\mathbf{A} = \begin{vmatrix} \omega_{\text{м}} & M_{\text{пп}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{3e}} & U_{\text{e}} \\ -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 \\ -c & 0 & c & 0 & 0 & 0 & 0 \\ \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 \\ 0 & 0 & 0 & \frac{-k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{pe}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{pe}}}{2T_{\text{мт}}^2 k_{\text{т}}} \\ 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ 0 & 0 & \frac{k_{\text{н}}}{T_a k_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_a} \end{vmatrix}$$

$$\mathbf{B} = \begin{vmatrix} U_{\text{3e}} & M_{\text{сг}} \\ 0 & -\frac{1}{J_{\text{м}}} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \frac{1}{T_a} & 0 \\ 0 & 0 \end{vmatrix}$$

3.3 Розрахунок перехідних процесів двомасової системи на ЕОМ

Визначимо параметри, необхідні для розрахунку перехідних процесів двомасової системи,

Момент інерції двигуна і жорстко пов'язаних з ним елементів системи, а також приведений момент інерції візка

$$J_{\Sigma} = \delta \cdot J_{\text{дв}} + \frac{m_{\text{в}} \cdot V_{\text{в}}^2}{\omega_{\text{н}}^2} ; \quad (7.5)$$

$$J_{\Sigma 1} = 1,2 \cdot 0,425 + \frac{8,5 \cdot 10^3 \cdot 0,55^2}{83,7^2} = 0,51 + 0,46 = 0,97 \text{ кг} \cdot \text{м}^2$$

Приведений момент інерції вантажу і вантажозахватного пристрою (момент інерції механізму):

$$J_{\text{м}} = \frac{(G_{\text{вп}} + m_0) \cdot V_{\text{в}}^2}{\omega_{\text{н}}^2} ; \quad (7.6)$$

$$J_{\text{м}} = \frac{(41 + 1) \cdot 10^3 \cdot 0,55^2}{83,7^2} = 2,95 \text{ кг} \cdot \text{м}^2 .$$

Решта параметрів визначені у попередніх розділах.

Розрахунок перехідних процесів проводиться на ЕОМ з використанням пакету прикладних програм MATLAB. Для розрахунку використовувалася модель системи, показана на рис.3.4. Крім того, розрахунок проводився по векторний-матричному рівнянню, отриманому в попередньому розділі.

У додатку Д приведений текст файлу, що містить початкові дані, матриці **A**, **B** і **C**, і команди розрахунку.

На рисунках 3.5 - 3.8 приведені графіки перших чотирьох змінних стану системи. Як видно з графіків, мають місце значні коливання основних координат системи. Це несприятливо позначається на роботі механізму, збільшуючи навантаження на елементах системи в перехідних режимах

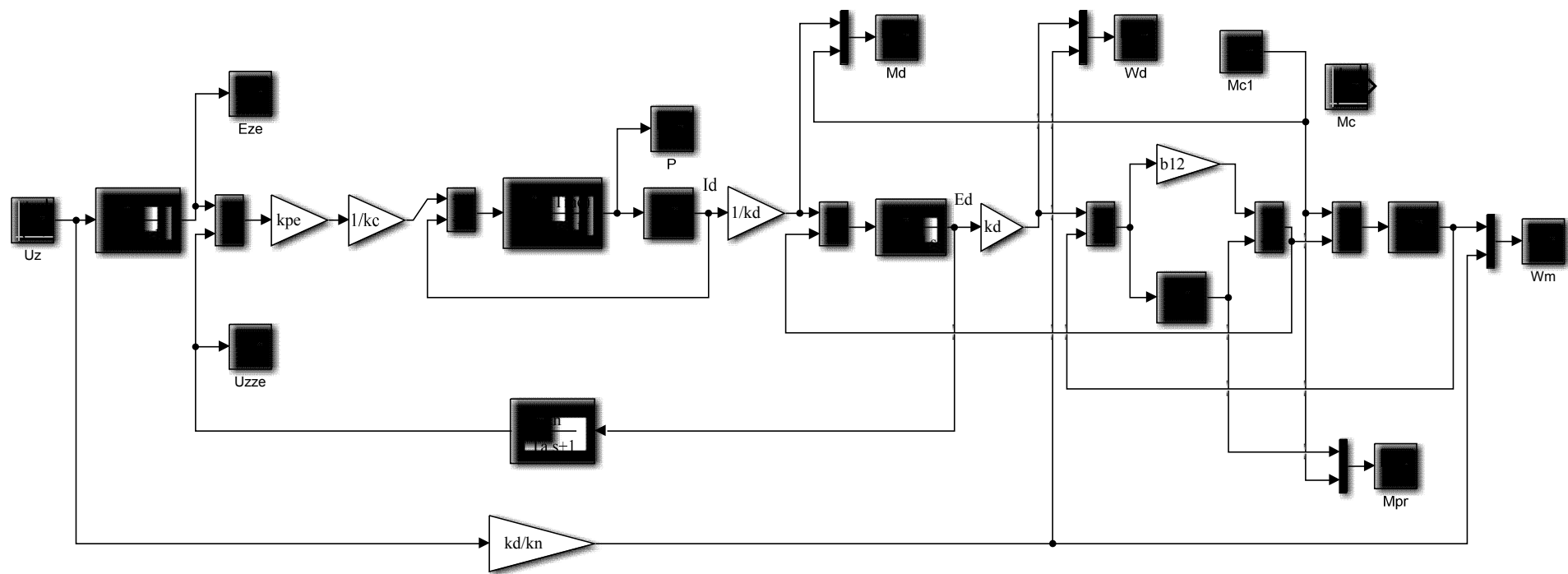
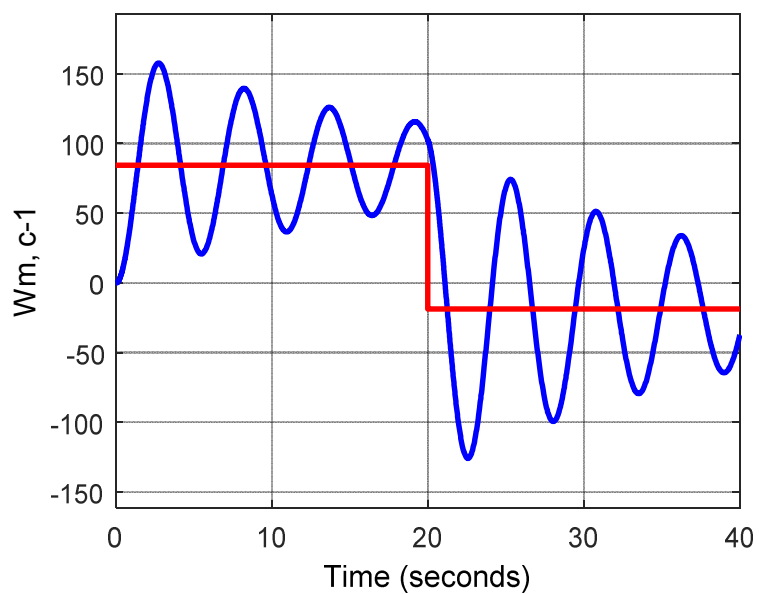
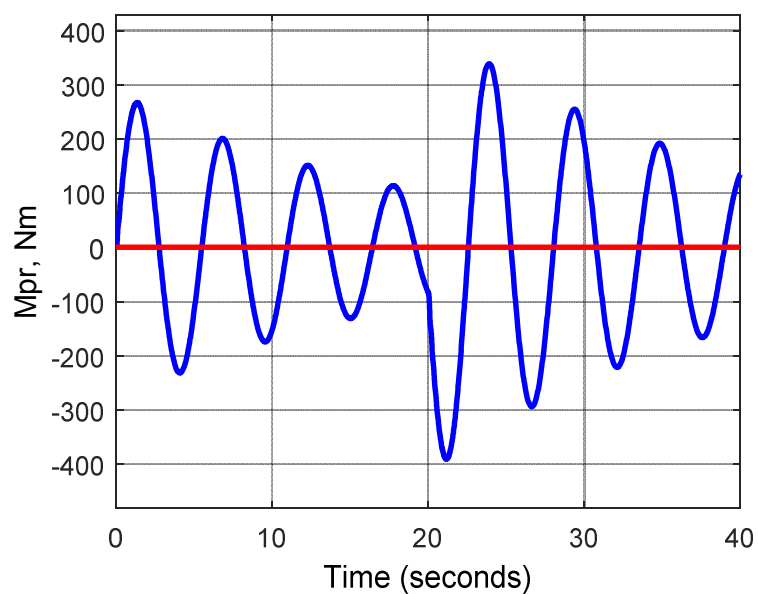


Рисунок 3.4 – Схема моделі двомасової системи, розроблена на ЕОМ



a)



б)

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

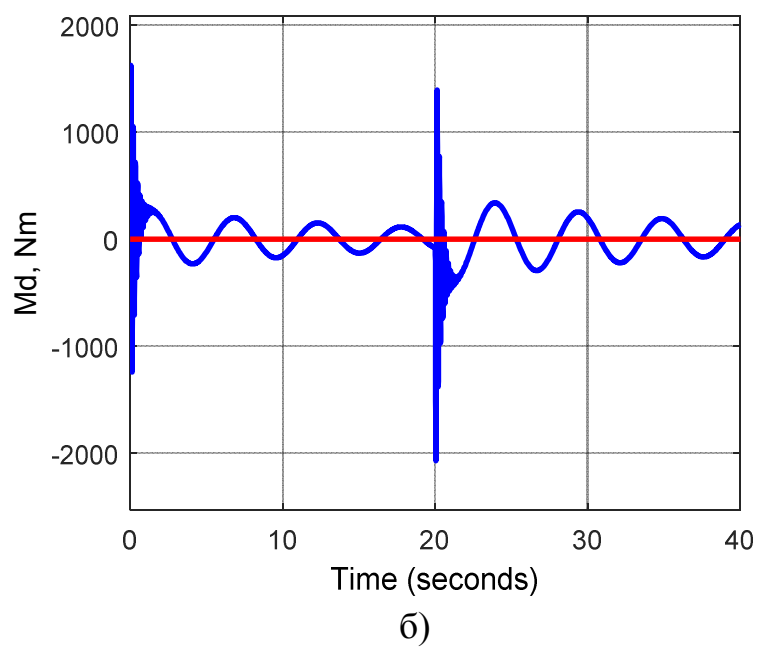
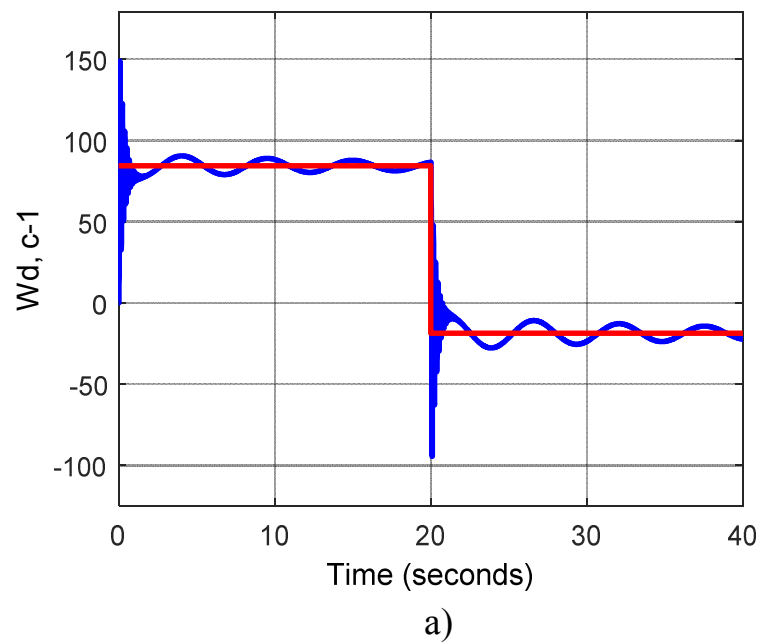
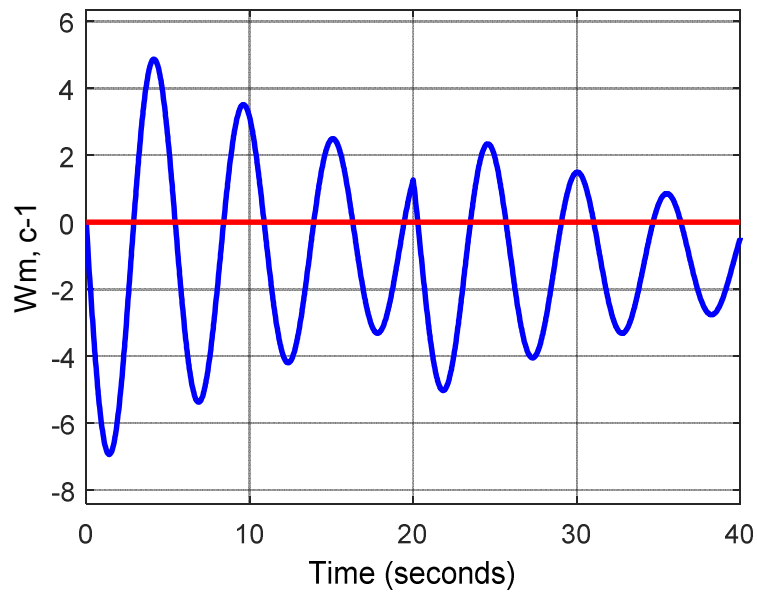
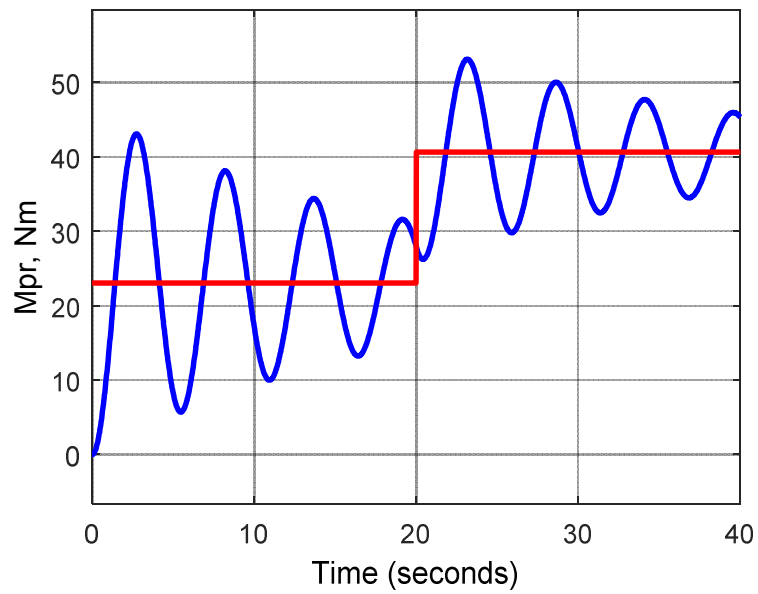


Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по задаючій дії



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

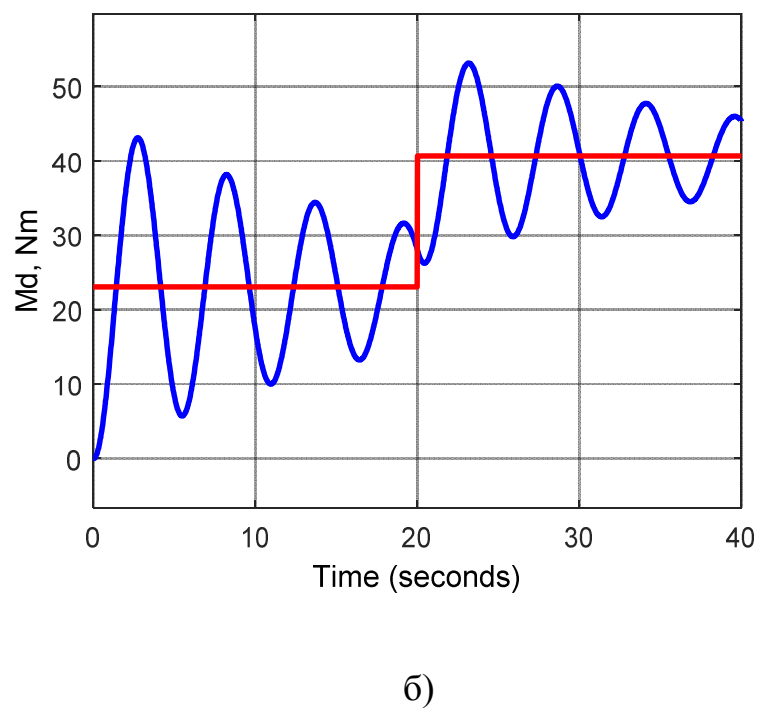
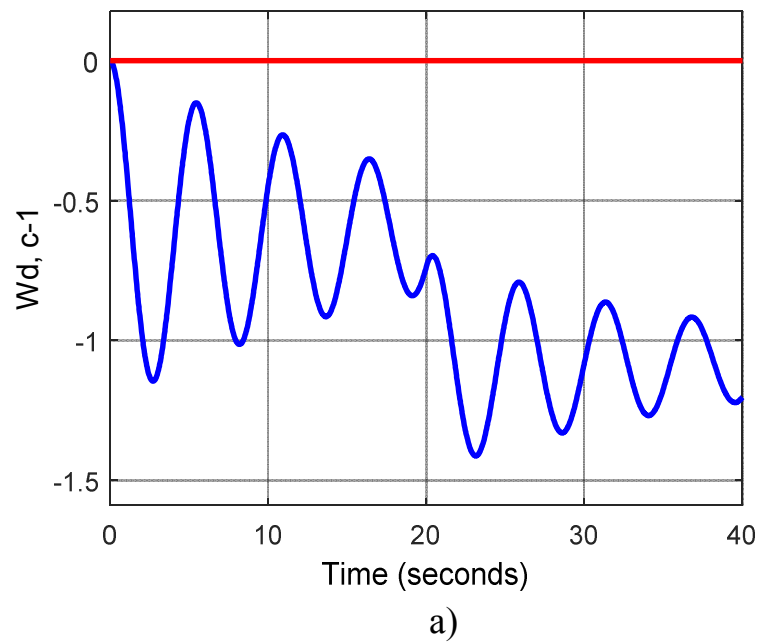


Рисунок 3.8 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму переміщення візка з урахуванням кінцевої жорсткості механічних зв'язків. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер рухів елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Внаслідок 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 переміщення візка з урахуванням кінцевої жорсткості зв'язку вантажу, що розгойдується, з візком може розглядатися як двомасова.

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

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

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

На етапі ідентифікації створюється модель керованого об'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 еталонної моделі. У цьому випадку модель керованого об'єкту активно використовується для налаштування параметрів самого регулятора.

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

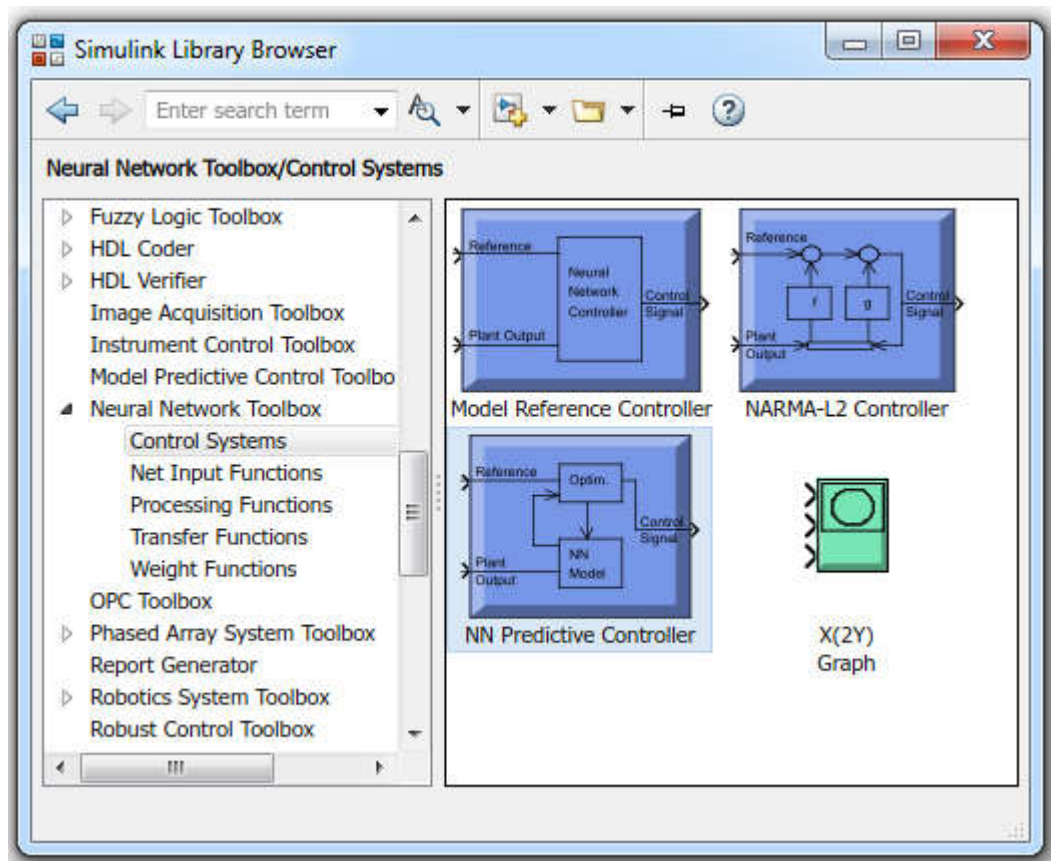


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

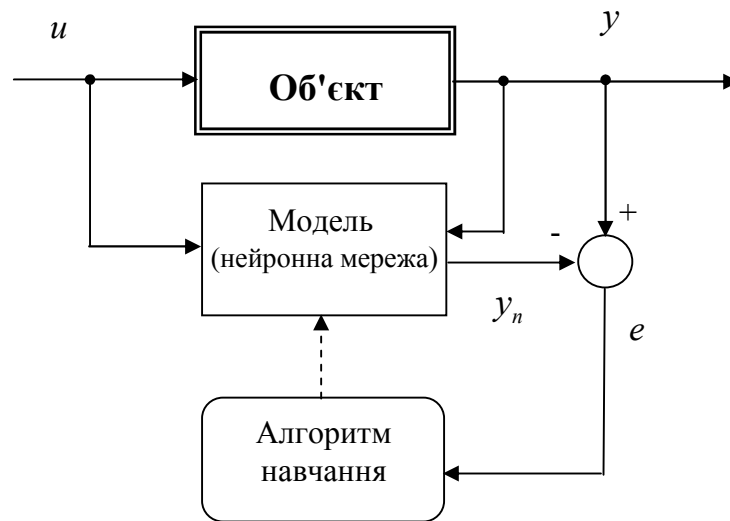


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

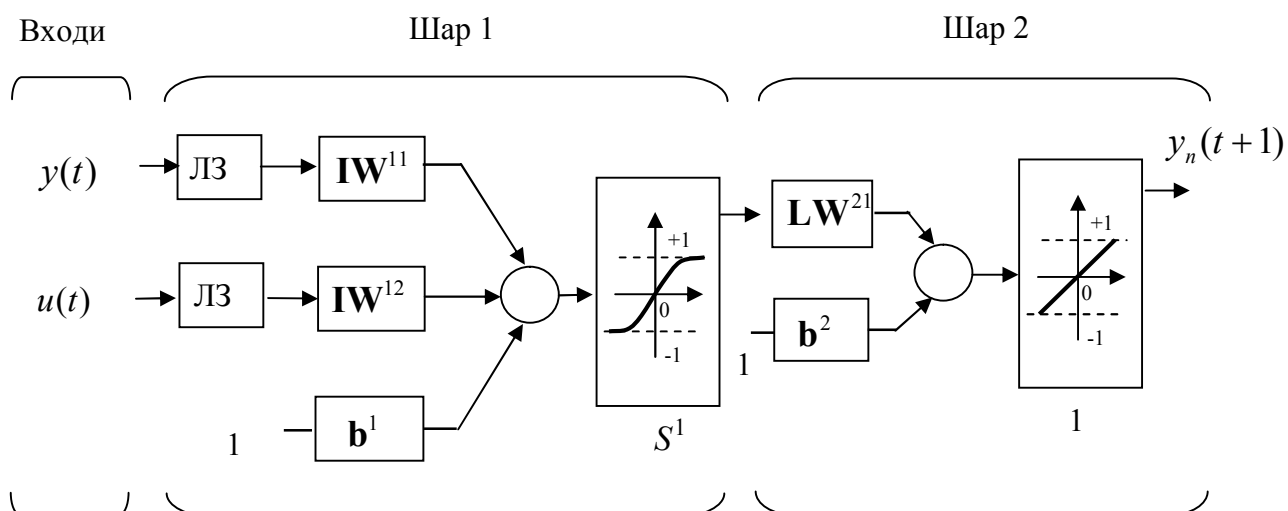


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

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.

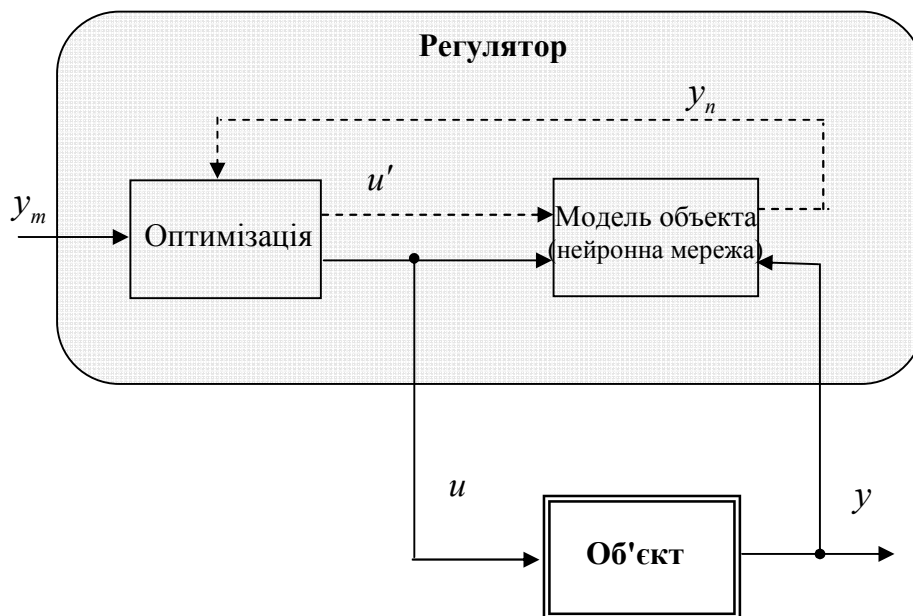


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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.

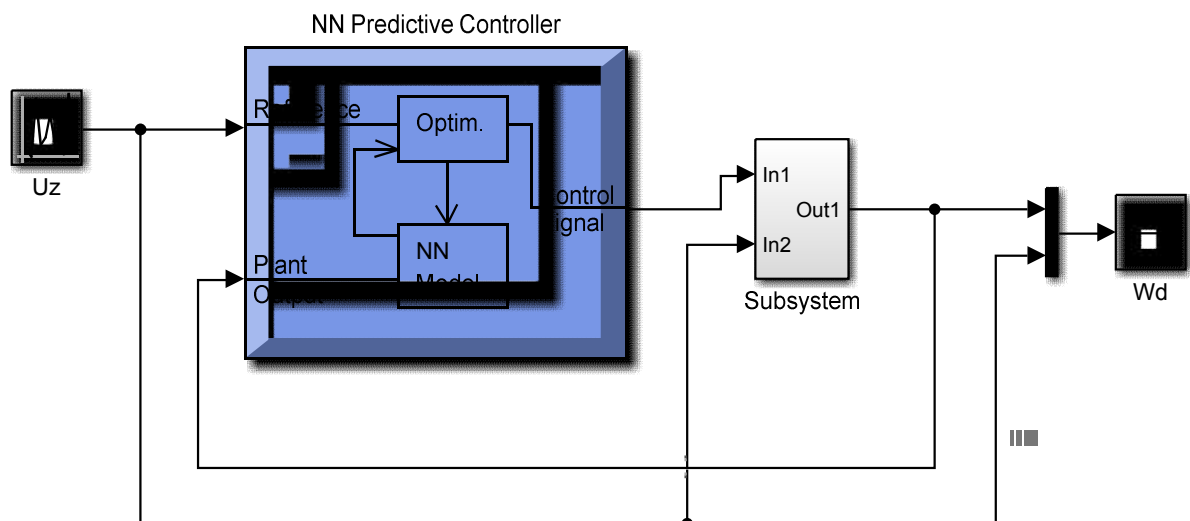


Рисунок 4.5 – Схема системи управління з нейрорегулятором

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-

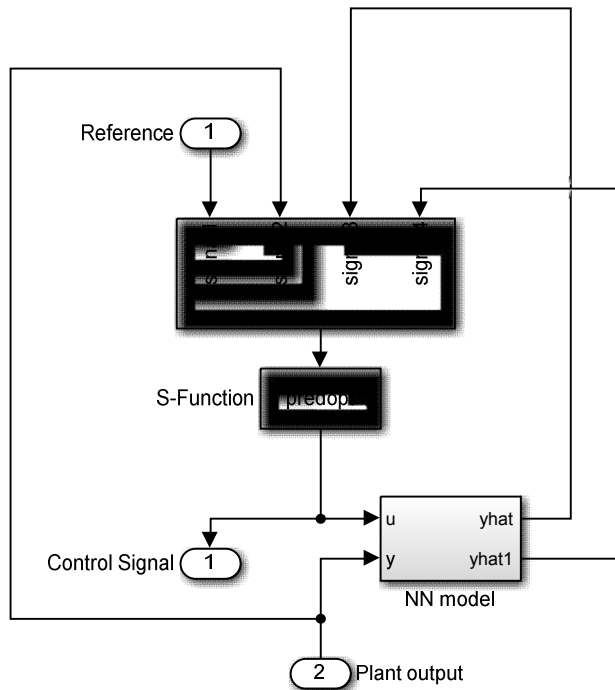


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

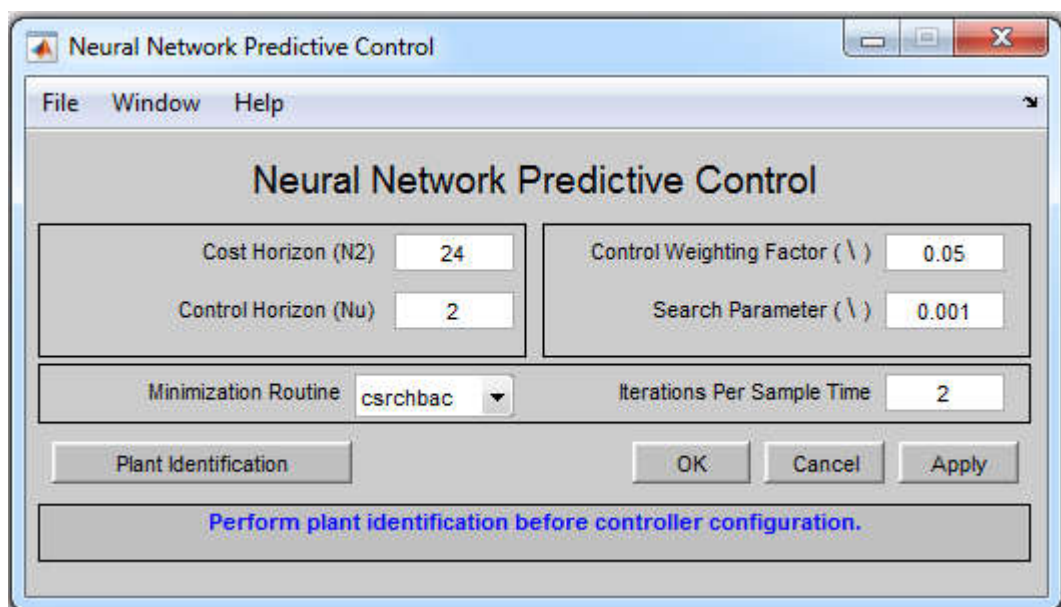


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

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

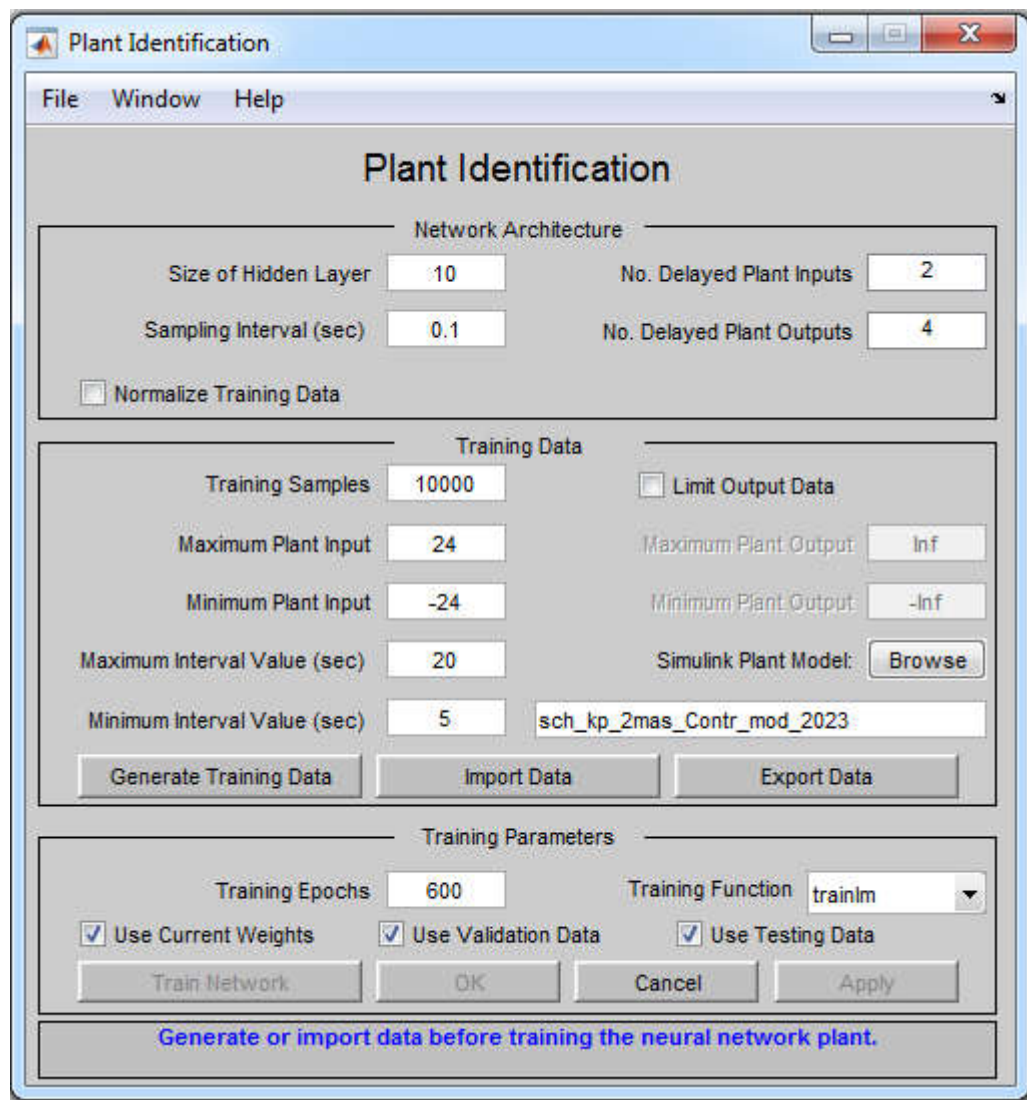


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

Опис вікна Plant Identification

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

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

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

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

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

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

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

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

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

No. Delayed Plant Outputs (кількість затриманих виходів системи). Параметр No. Delayed Plant Outputs вказує на кількість запізнених (затриманих) виходів системи, які враховуються при ідентифікації динаміки системи. Запізнені виходи 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 виходи 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.

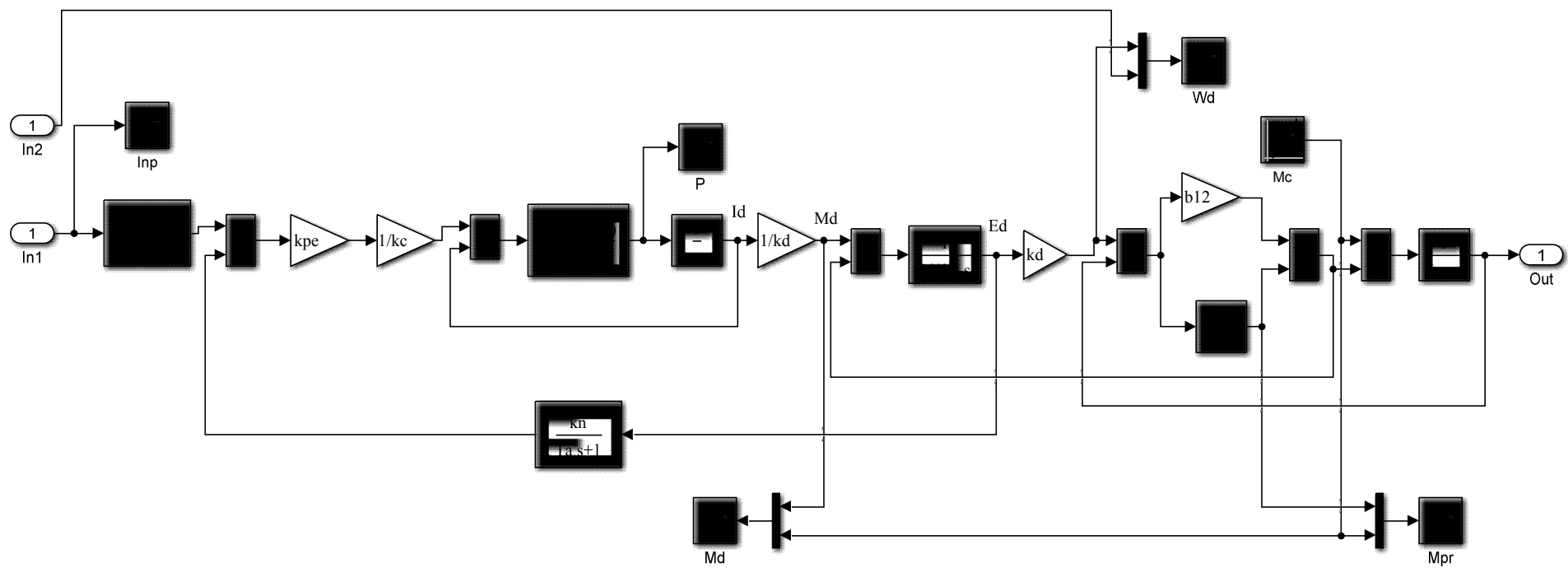


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

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

найменших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

Кожна з цих функцій має свої особливості та параметри, які можна налаштовувати 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. Ви-

користання 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.

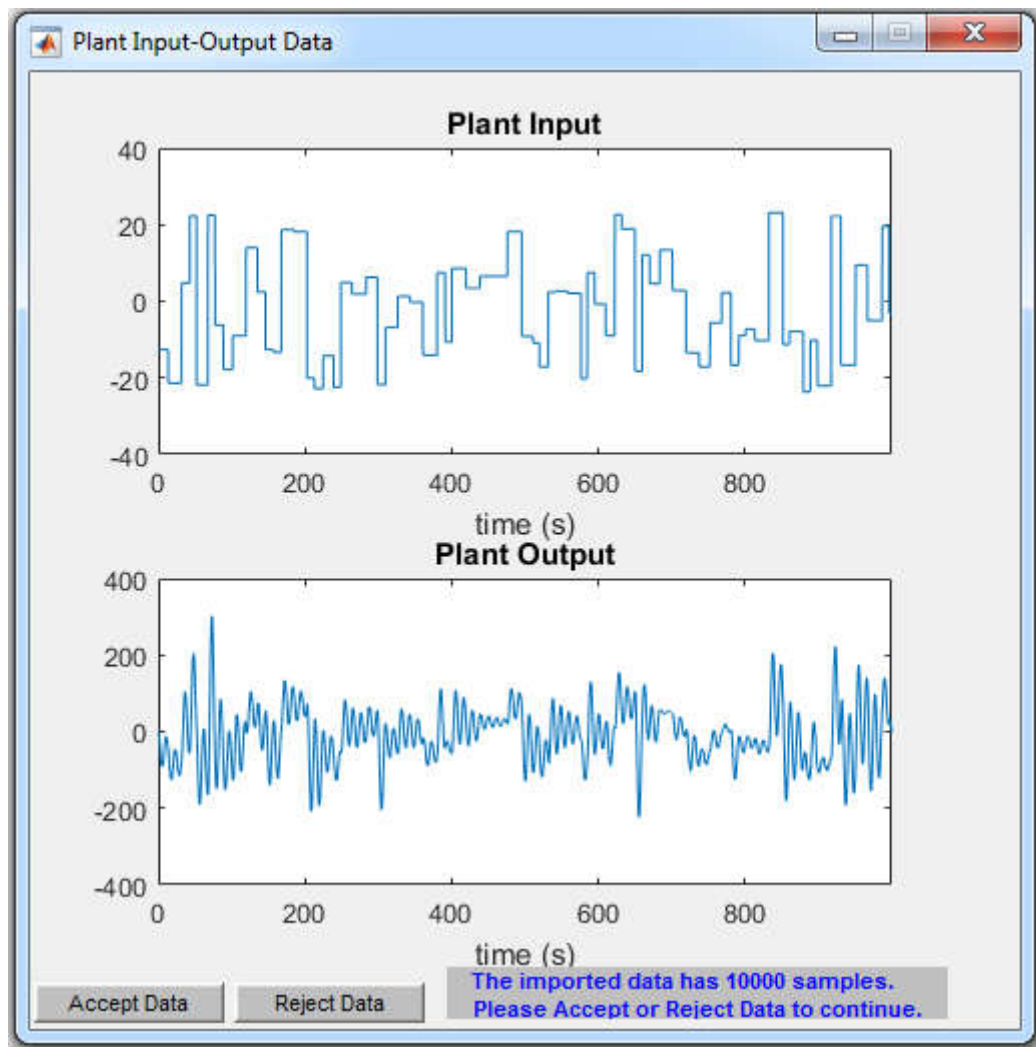


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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).

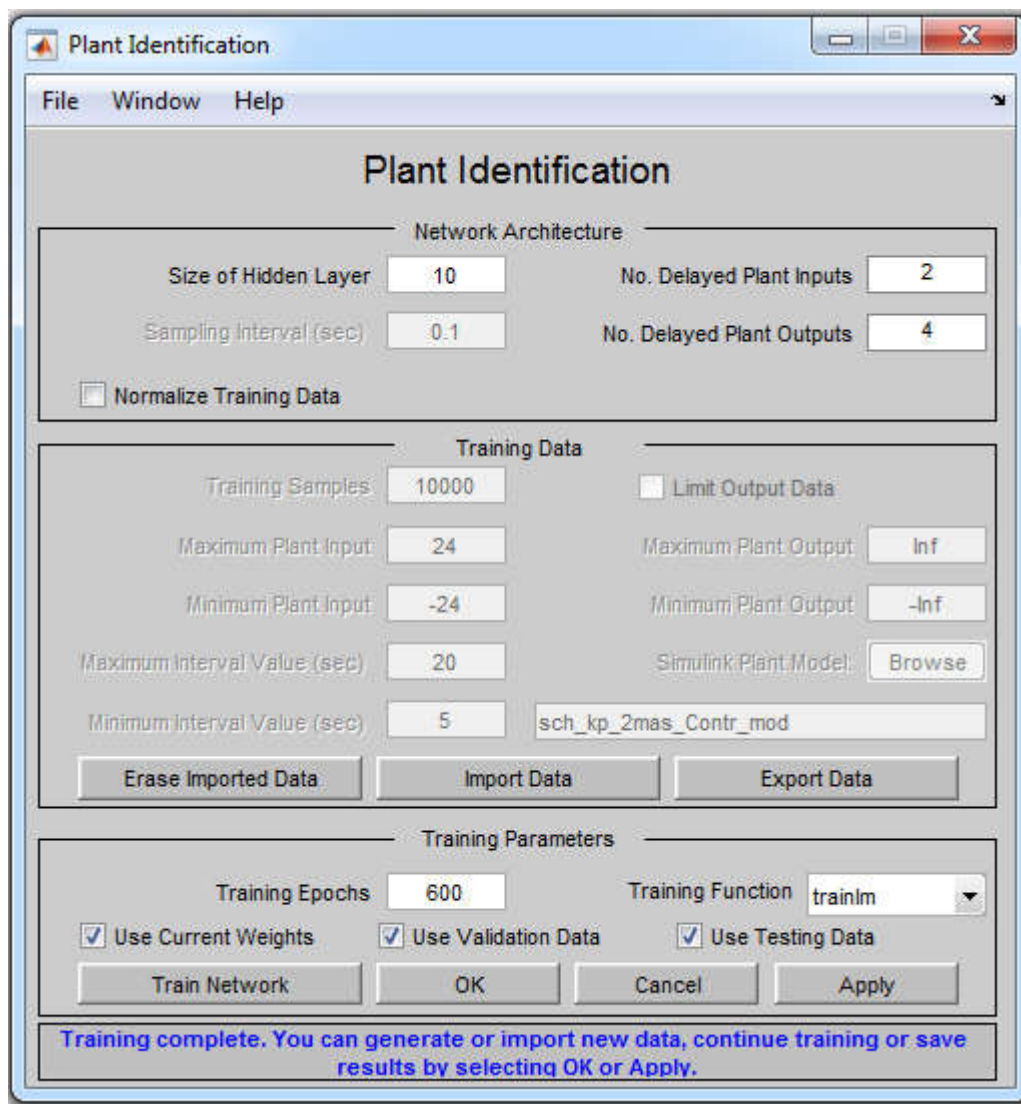


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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.

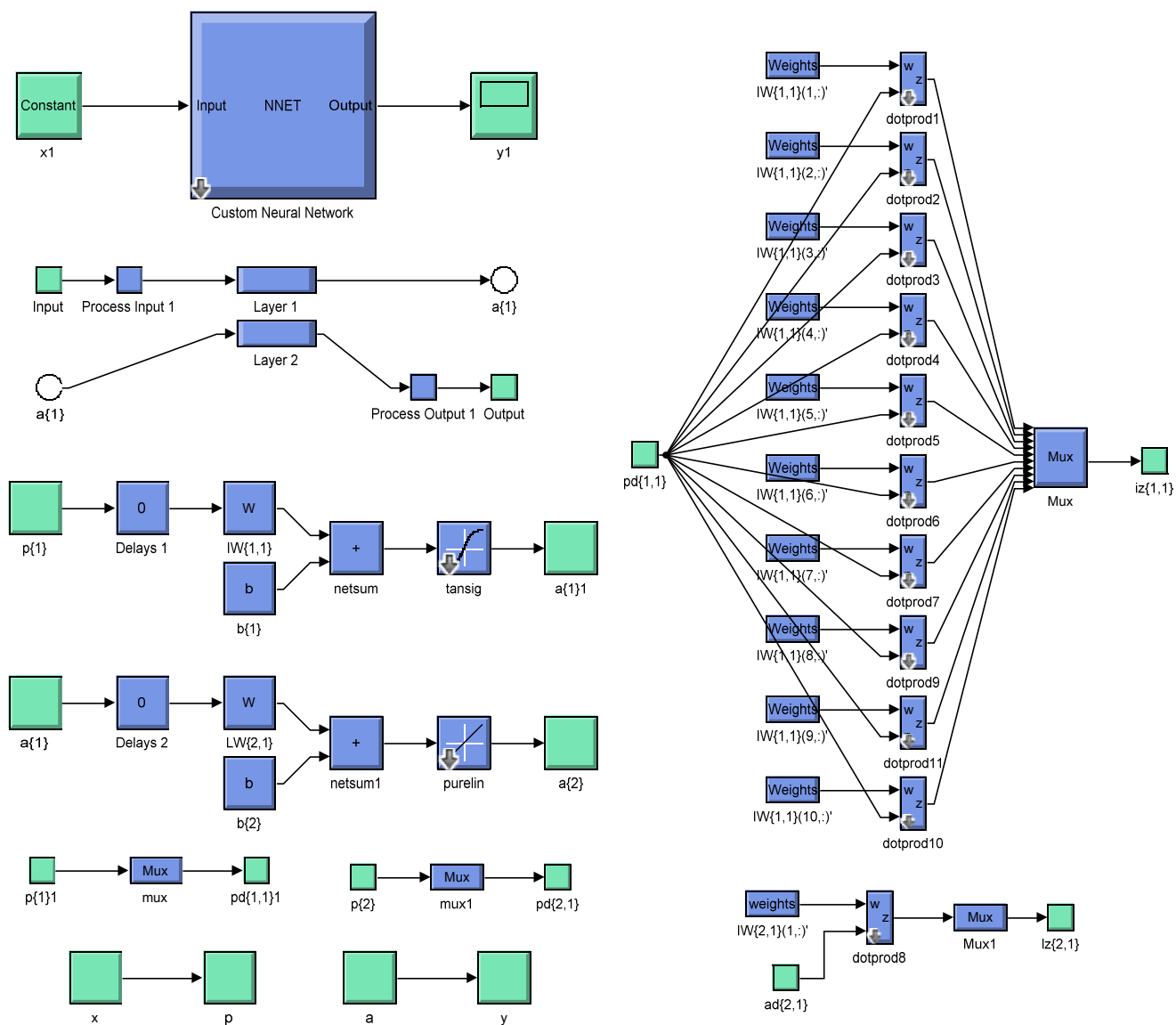


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

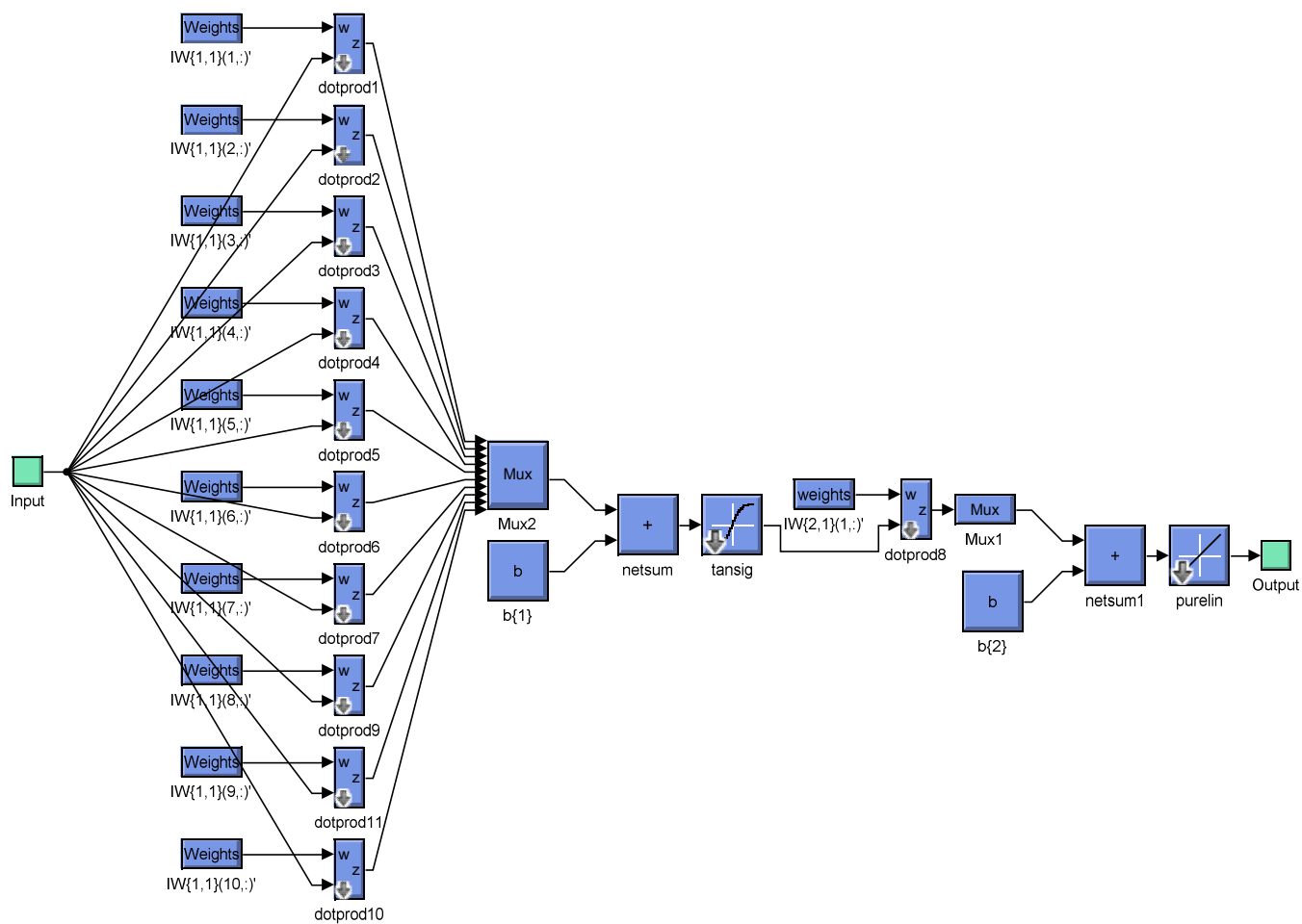


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

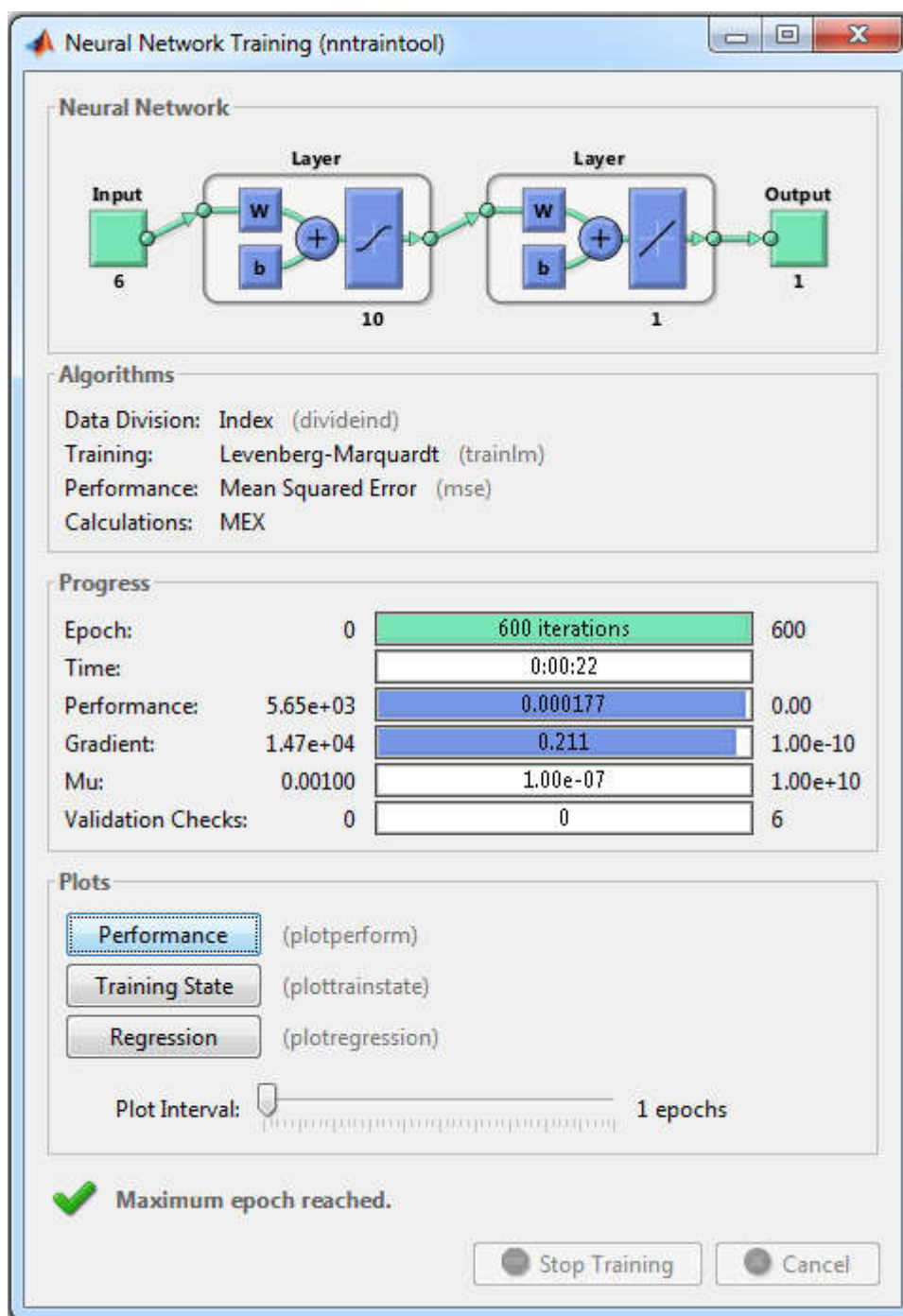


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

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.

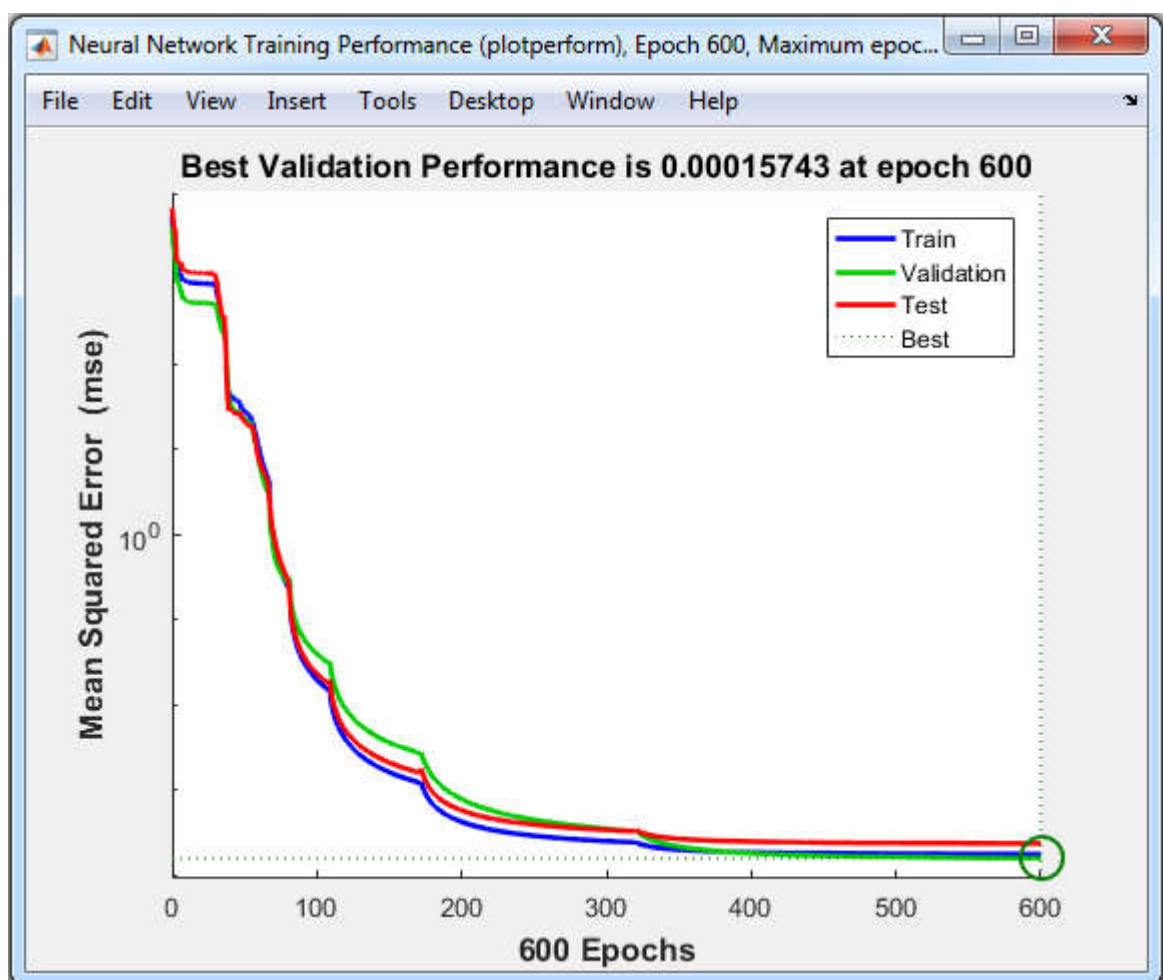


Рисунок 4.15 – Вікно контролю процесу навчання

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

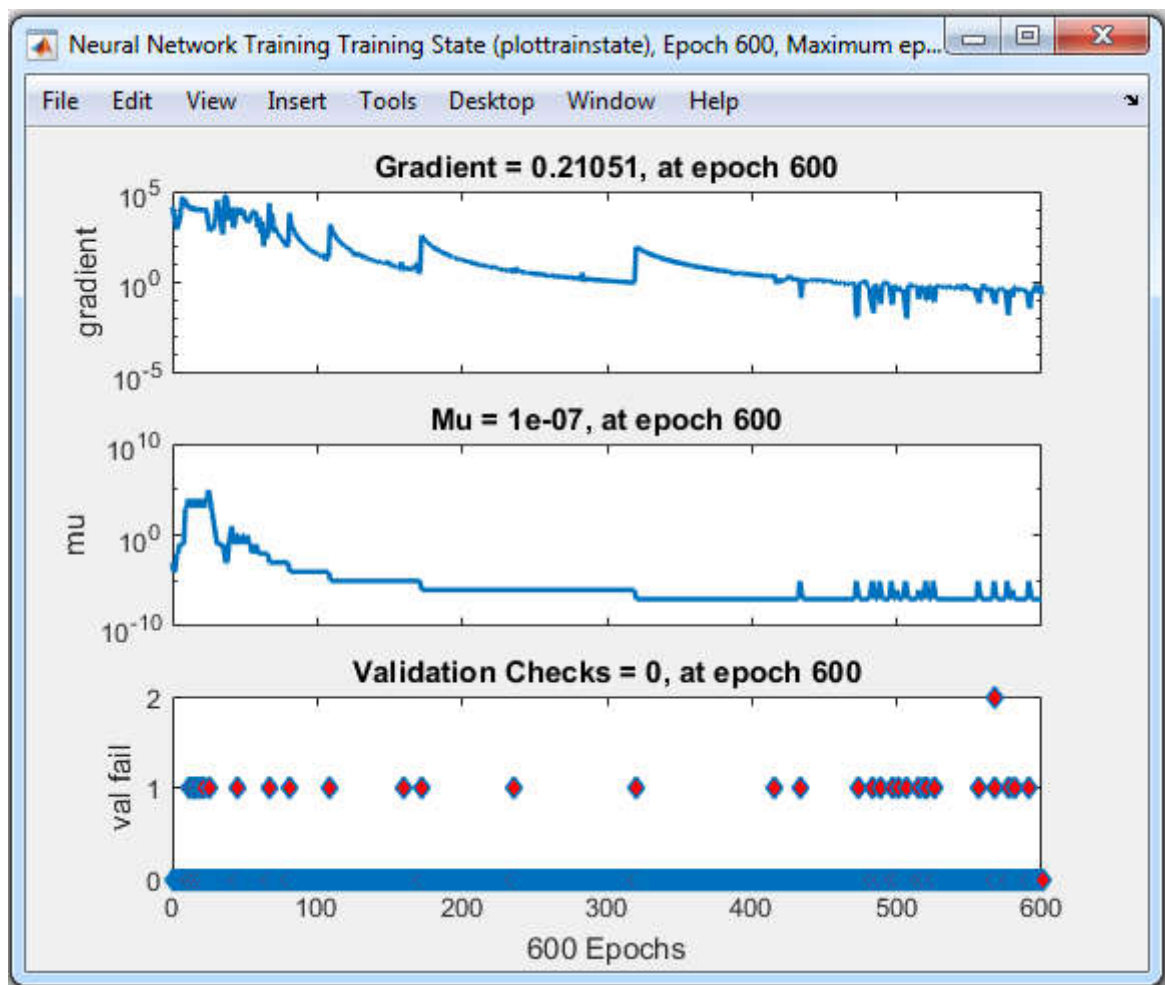


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

Графік Gradient показує зміну значення градієнта під час навчання нейронної мережі. Градієнт в контексті навчання нейронних мереж є вектором,

який вказує напрямок та міру найшвидшого зростання функції втрати (або помилки) мережі в точці навчання. Градієнт використовується для оновлення вагових коефіцієнтів мережі під час оптимізації.

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

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

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

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

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

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.

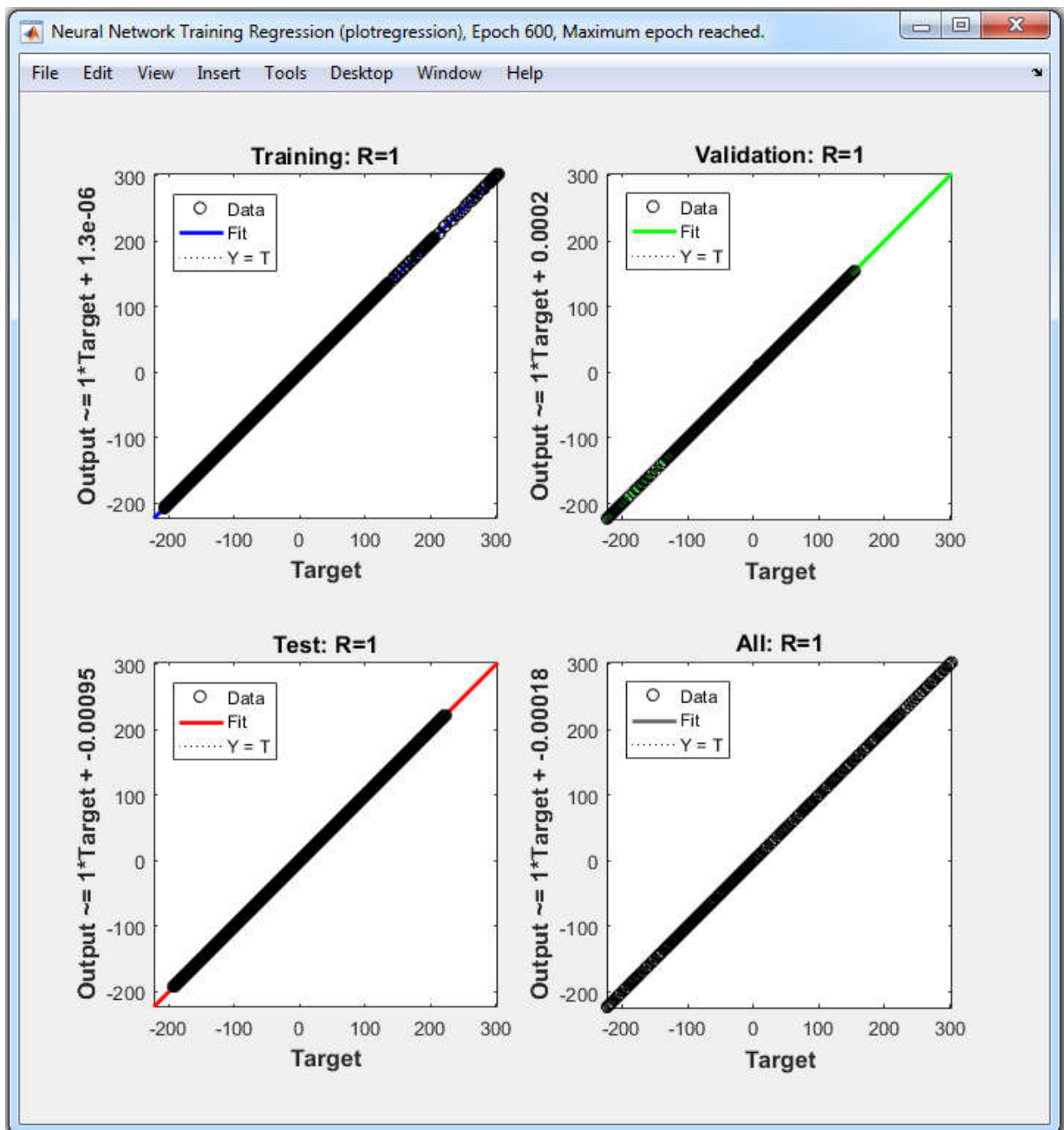


Рисунок 4.17 - Вікно Neural Network Training Regression

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

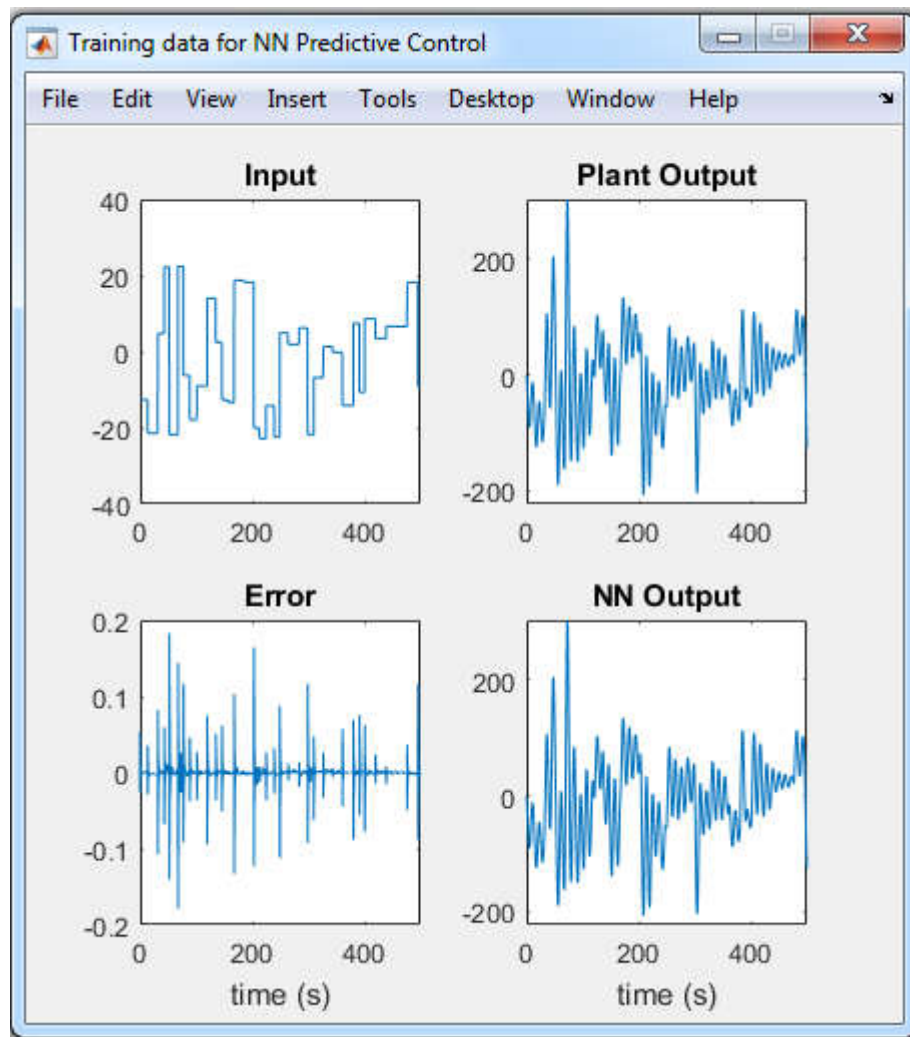


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

динаміку системи під впливом вхідних сигналів.

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

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

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

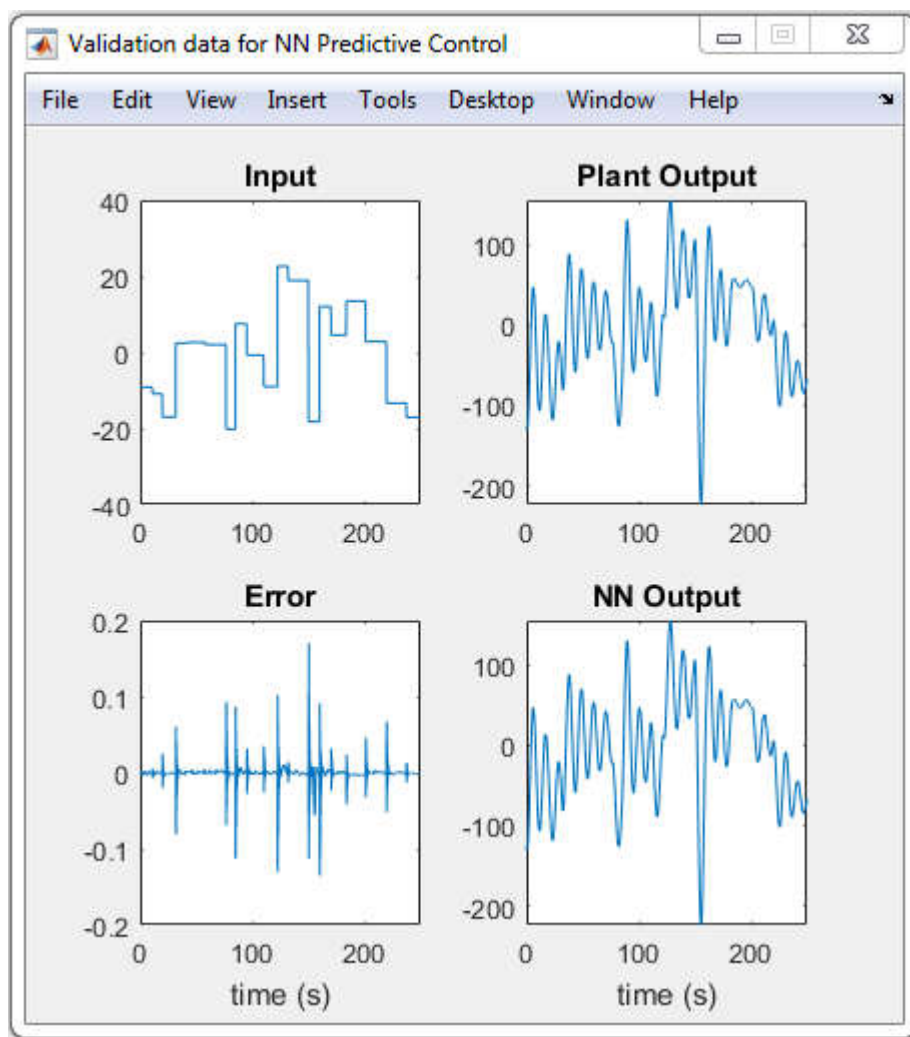


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

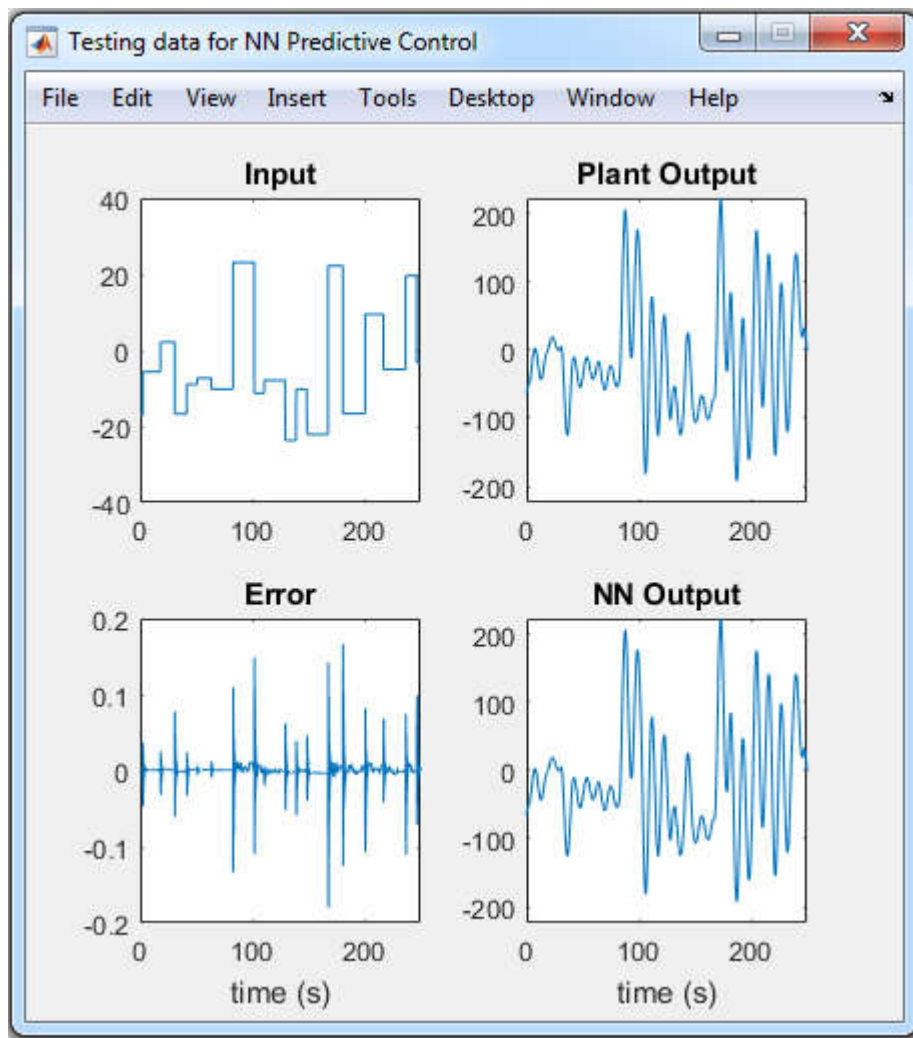


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

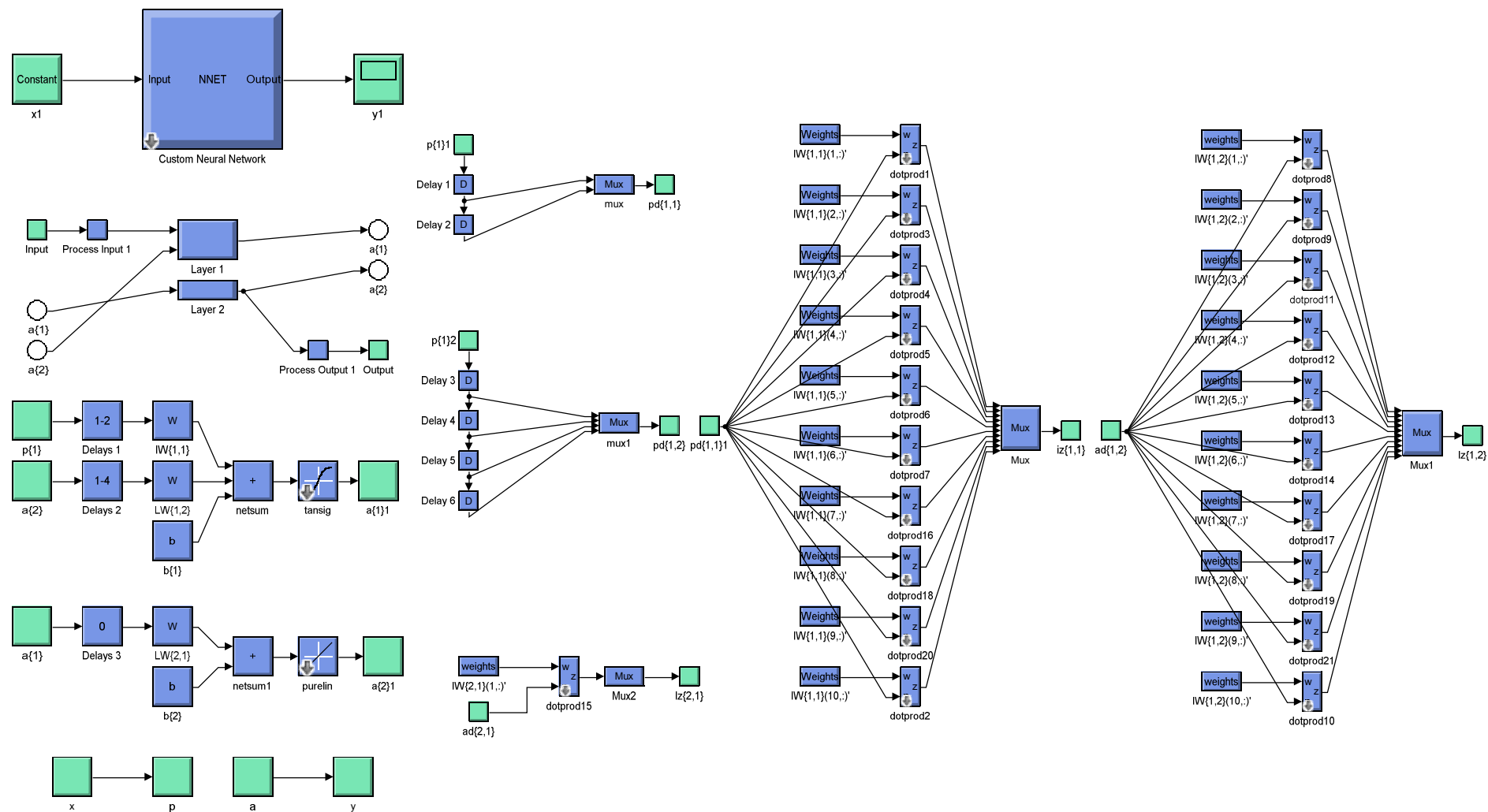


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

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

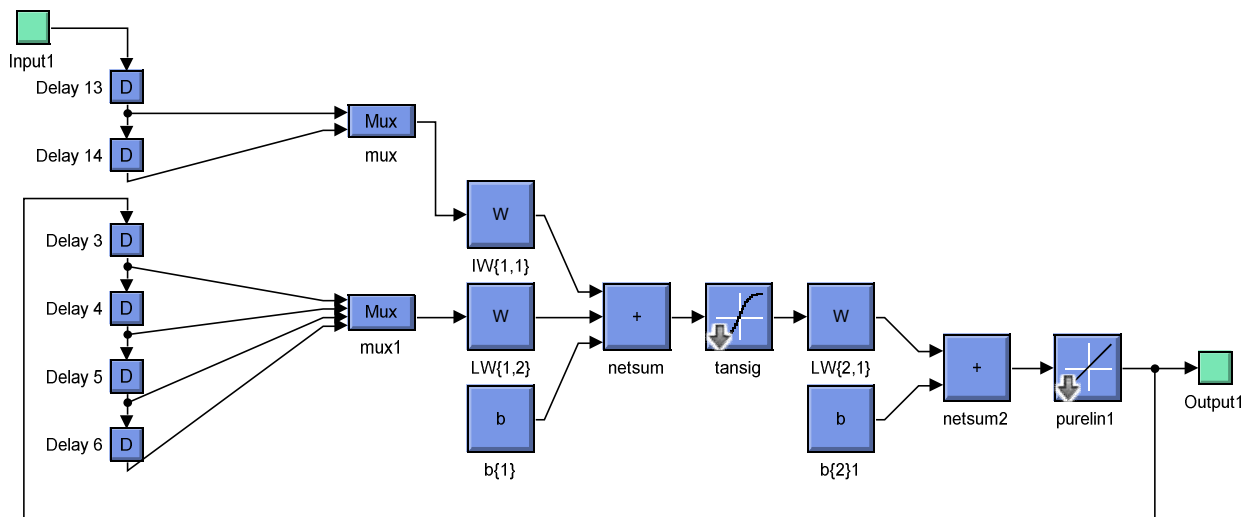


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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.

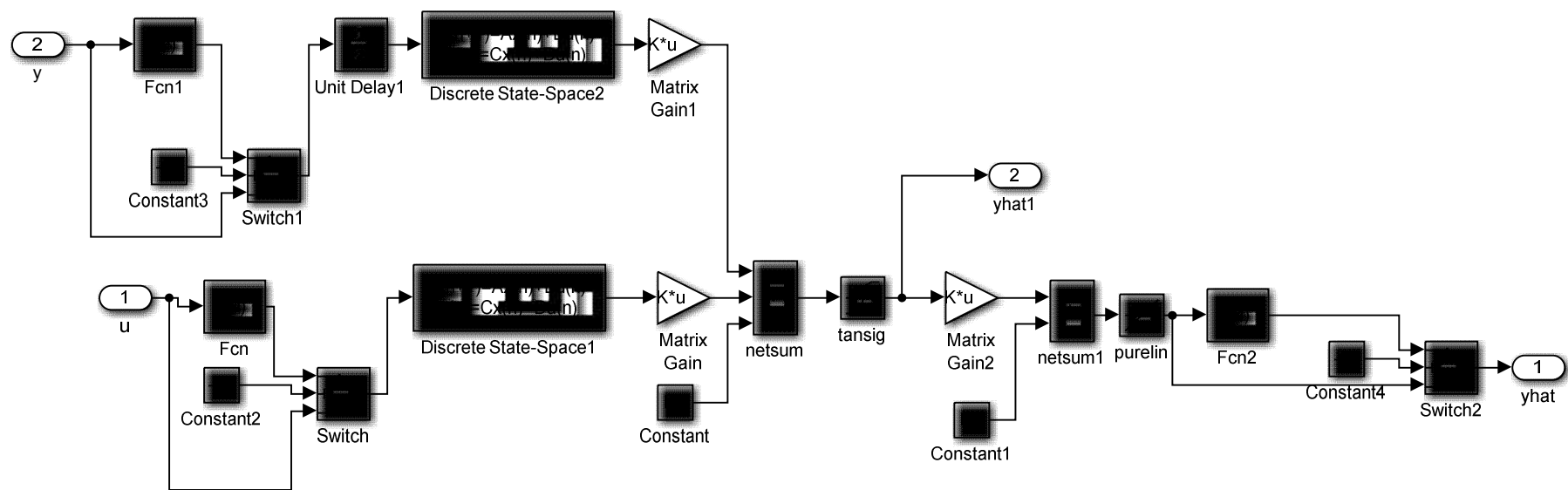


Рисунок 4.23 – Структурна схема нейронмережевої моделі об'єкту регулювання

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.

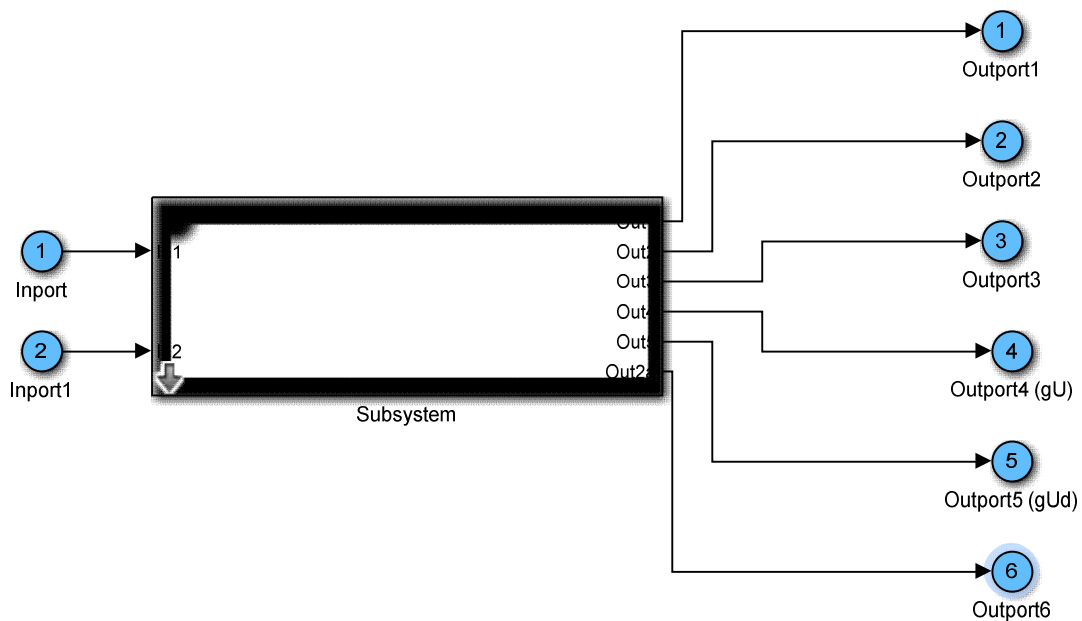


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

Після завершення побудови нейромережевої моделі керованого об'єкту відбувається повернення до вікна Neural Network Predictive Controller (рис.4.7) і встановлюються параметри оптимізації.

Продовження опису вікна Neural Network Predictive Controller

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

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

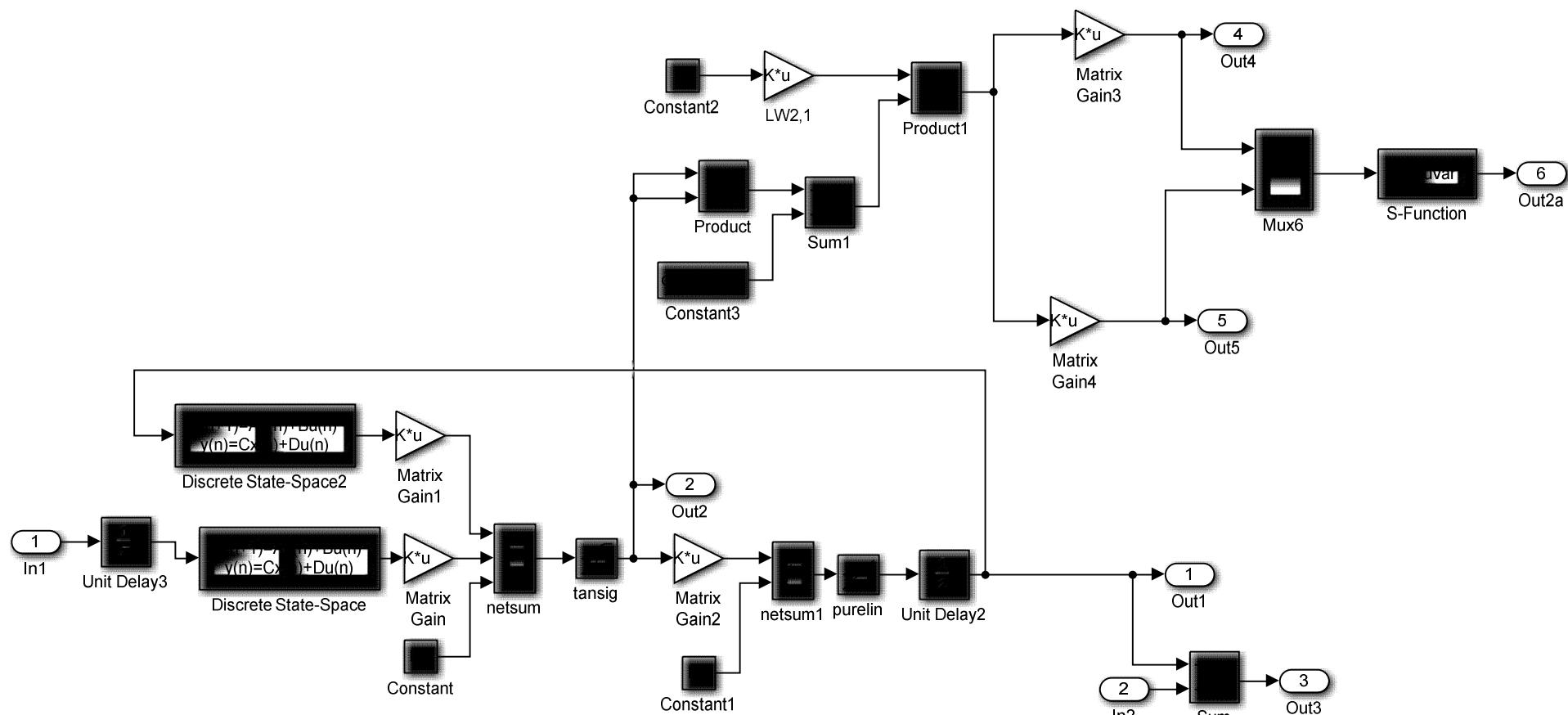


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

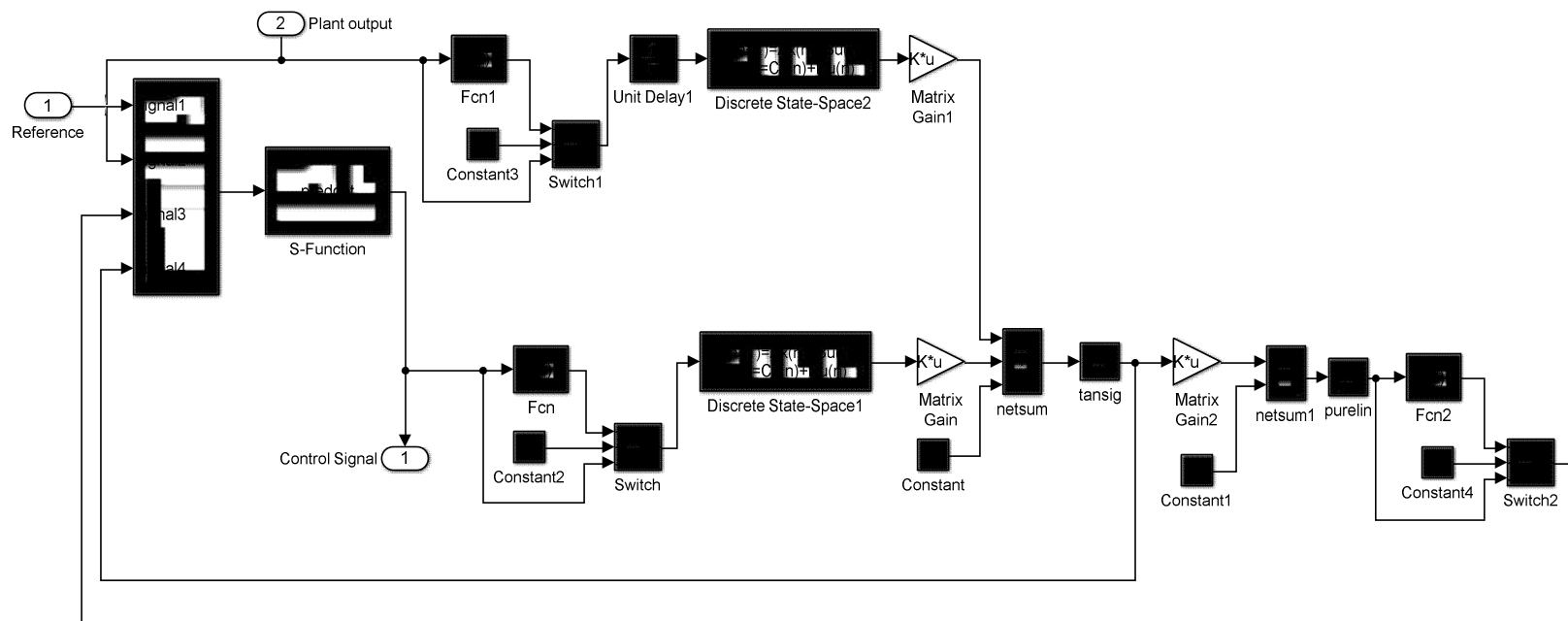


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

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

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

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

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

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

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

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

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

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

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

Успіх навчання мережі суттєво залежить від довжини навчальної вибірки N_b та такту дискретності Δt , який визначає інтервал між двома послідовними моментами збору даних. Оптимальні значення: $N_b = 10000$, $\Delta t = 0,05 \div 0,1$ с. Збільшення Δt знижує точність обчислень і зменшує різницю між помилкою на навчальній множині та помилками на контрольній і тестовій множинах. Зменшення Δt вимагає збільшення N_b , що призводить до значного зростання часу навчання без суттєвого зменшення помилки.

Для отримання репрезентативної вибірки важливо правильно встановити мінімальне та максимальне значення інтервалу ідентифікації. Ці параметри залежать від характеристик об'єкта **Subsystem**. У даній роботі обрані межі: $t_{\min} = 4 \div 5$ с, $t_{\max} = 10 \div 20$ с.

Під час синтезу нейромережевої моделі системи задається кількість елементів затримки на вході z_1 та виході z_2 моделі. Найкращі результати досягнуті при $z_1 = 2$, $z_2 = 2 \div 5$.

Результат навчання мережі також залежить від початкових значень ваг w_{ij} та кількості навчальних циклів $N_{\text{ц}}$. Для досягнення глобального мінімуму процес навчання слід повторювати багаторазово з різними початковими значеннями w_{ij} та різними значеннями $N_{\text{ц}}$. У даній роботі для кожної конфігурації мережі було обрано кілька сотень початкових точок, а кількість навчальних циклів, після яких помилка переставала зменшуватись, становила $300 \div 600$.

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

Для побудови графіків перехідних процесів двомасової системи з нейрорегулятором NN Predictive Controller використовуємо моделі Simulink, приведені на рис. 4.5 або 4.27.

На рис. 4.28 – 4.31 приведені графіки перехідних процесів основних змінних стану синтезованої системи. З аналізу отриманих характеристик виходить, що реакція системи на ступінчасті дії з випадковою амплітудою цілком задовільна, має коливальний характер з невеликим перерегулюванням

Отже, синтезований нейромережевий регулятор з прогнозом NN Predictive Controller може бути використаний для управління двомасовою електромеханічною системою, що розглядається в роботі.

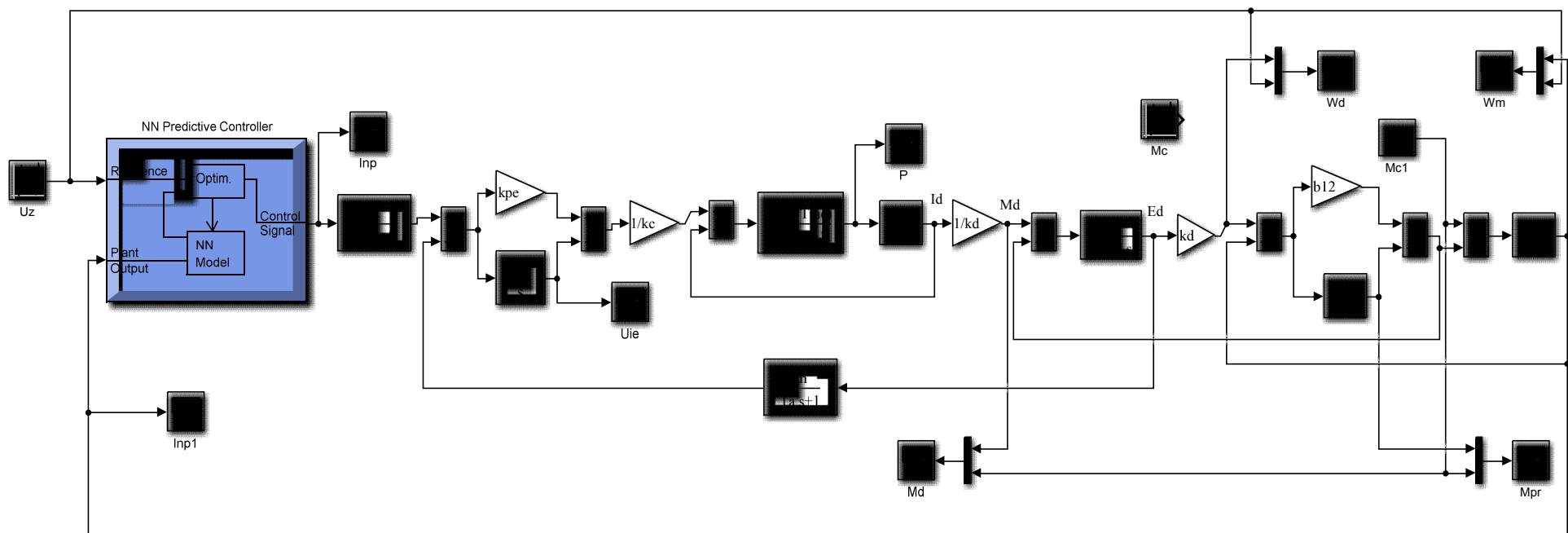
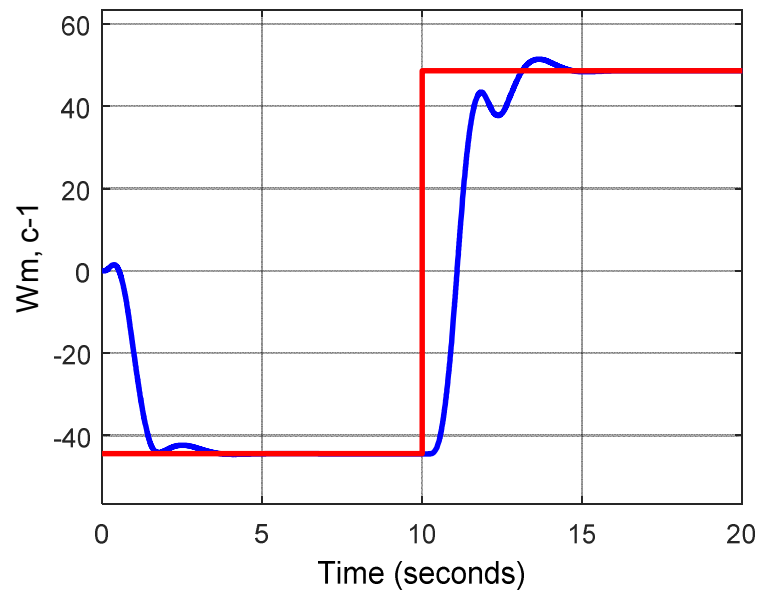
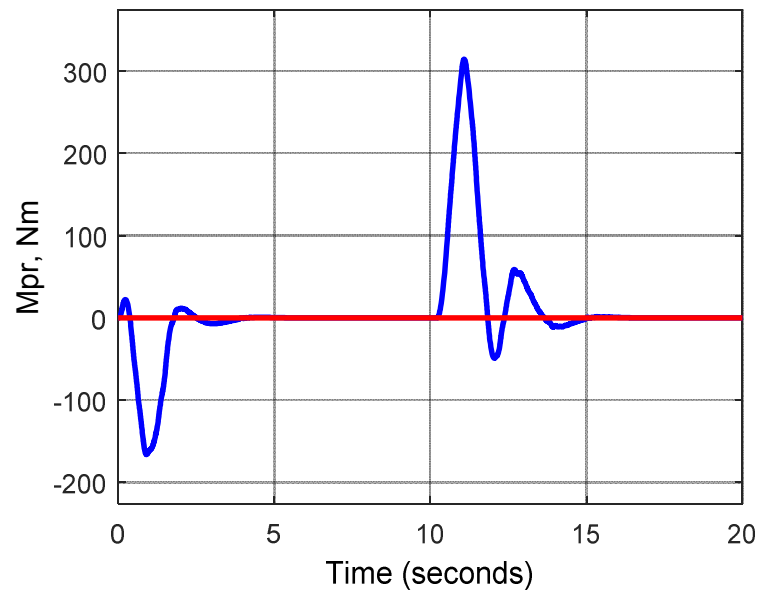


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором



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 по задаючій дії

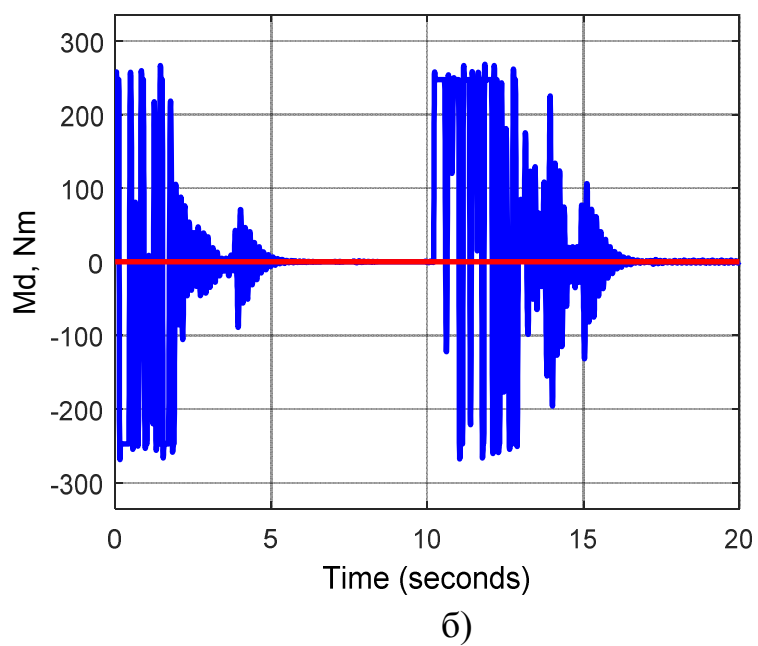
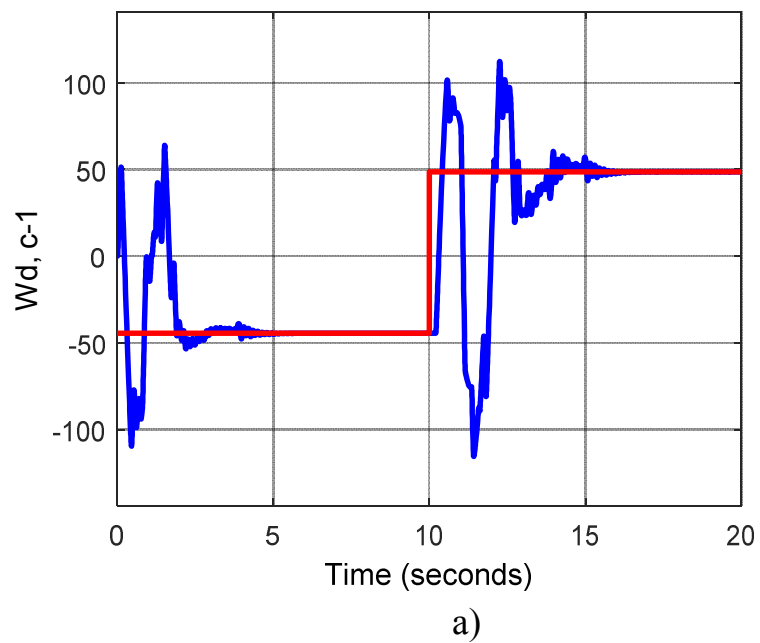


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

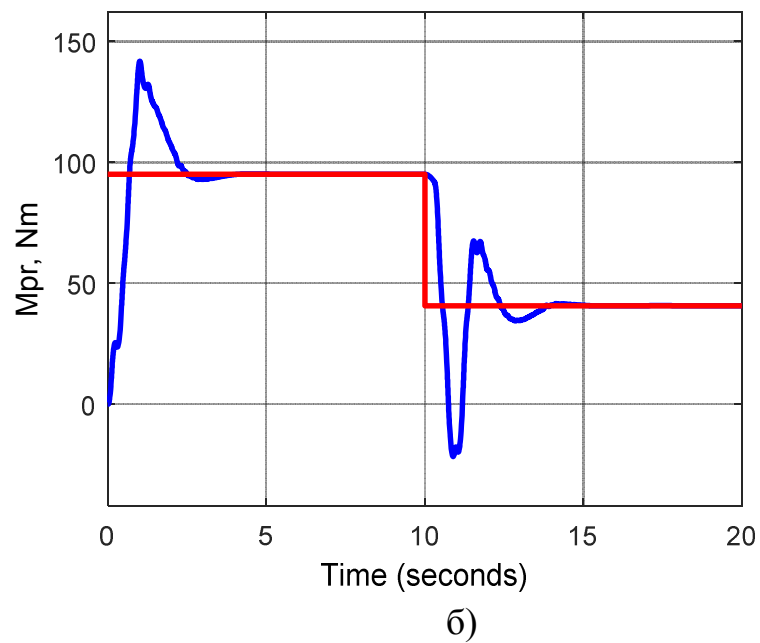
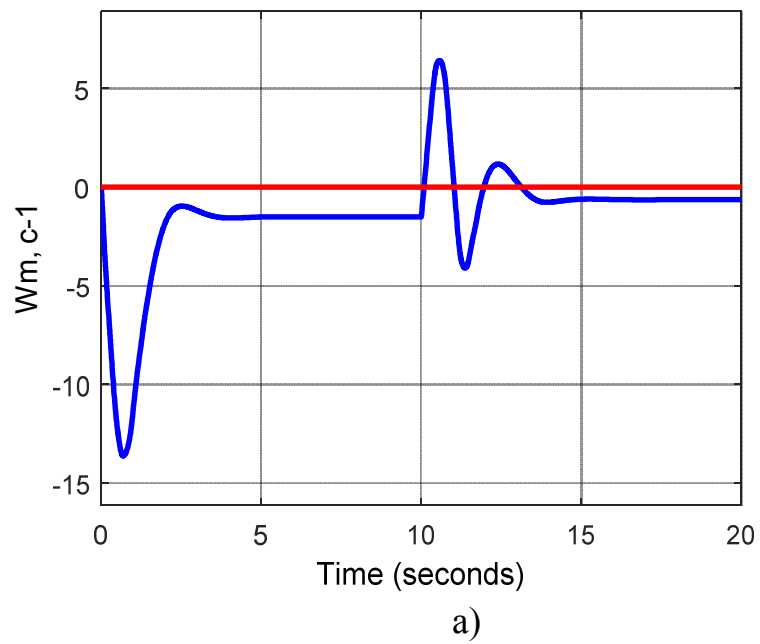


Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

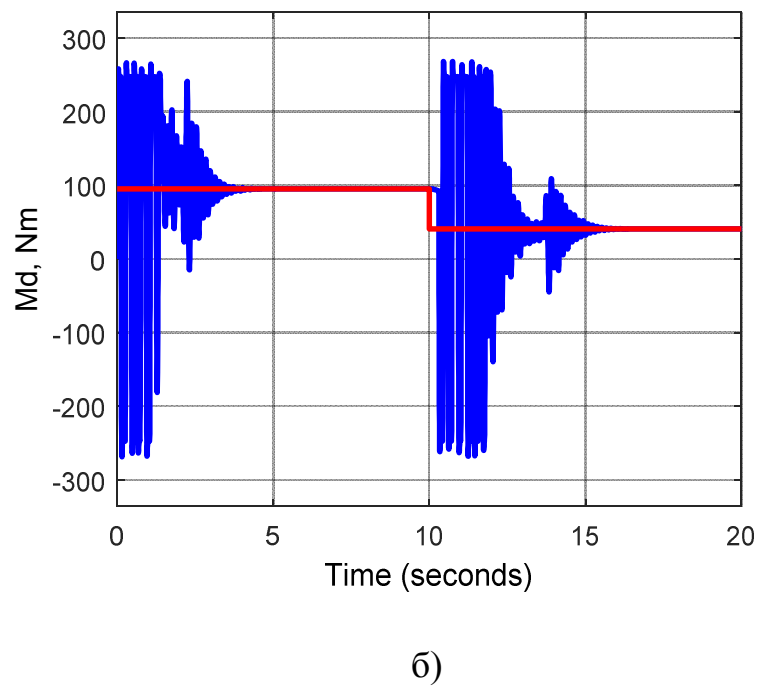
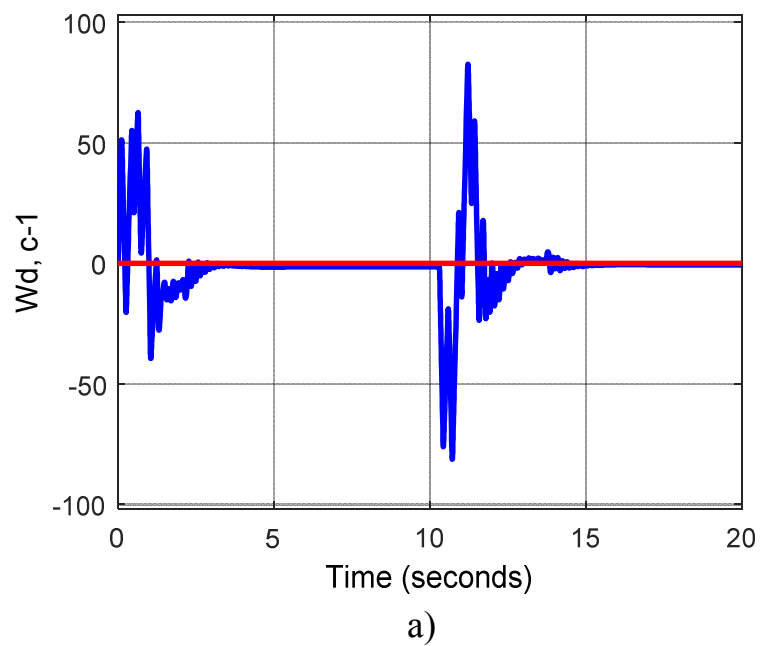


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проаналізовано принципи побудови нейрорегулятора, реалізованих в пакеті прикладних програм (ППП) Neural Network Toolbox системи MATLAB: регулятор з прогнозом (NN Predictive Controller); регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) і регулятор на основі еталонної моделі (Model Reference Controller). Для системи управління механізмом підйому мостового крана, 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 підйому мостового крана 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 10000; інтервал між двома послідовними моментами знімання даних 0,1с; мінімальний інтервал ідентифікації 5 с; максимальний інтервал ідентифікації 20 с.

4. Створено двошарову нейронну мережу з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-4} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують 0,2. Коефіцієнт кореляції R дорівнює 1:

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 переміщення візка мостового крана, що забезпечують високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління основних механізмів мостових кранів, 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 переміщення візка мостового крана.

2. На підставі аналізу вимог до систем електроприводів основних механізмів мостових кранів для електроприводу механізму переміщення візка вибраний електропривод постійного струму, виконаний за системою ТП–Д. Виконано вибір електродвигуна і основного електроустаткування силового кола електроприводу.

3. Проведений вибір і розрахунок системи автоматичного управління параметрами електроприводу. Для управління координатами електроприводу застосована система підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. Для поліпшення динамічних характеристик системи при роботі в режимі переривистих струмів в контур струму включений внутрішній контур напруги тиристорного перетворювача. Виконано розрахунок параметрів силового кола електроприводу, що використовуються при розрахунку системи автоматичного управління. Проведений синтез послідовних коректуючих пристроїв контурів, з оптимізацією контура струму і контура ЕРС – по модульному критерію. Проведено моделювання системи на ЕОМ з використанням пакету прокладних програм MATLAB.

4. Розроблена математична модель двомасової системи, що враховує особливості динаміки системи при урахуванні поперечних розгойдувань вантажу в перехідних режимах. Проведено моделювання системи на ЕОМ. Дослідження показали, що в двомасовій системі перехідні процеси основних координат мають значні коливання.

5. Для додання системі бажаних динамічних характеристик виконаний синтез системи управління з нейрорегулятором NN Predictive Controller, реалізованим в пакеті прикладних програм Neural Network Toolbox системи MATLAB. Розглянутий принцип побудови і функціонування нейромережевого регулятора, приведений порядок синтезу регулятора, побудовані процеси в двомасовій системі з нейрорегулятором. Аналіз результатів моделювання показав, що реакція системи на ступінчасту дію з випадковою амплітудою цілком задовільна, має коливальний характер з достатньо швидким загасанням. Система є астатичною. 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. Тимошук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. - 444 с.
8. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
9. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник.. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 224 с.
10. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. – 353 с.
11. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344с.

12. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления. – М.: ИПРЖР, 2002. – 480 с.
13. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review.
15. Рябинин А. Д., Шквар А. М. Некоторые принципы функционального построения инвариантных бионических систем управления // Тр. 4-го Всесоюз. совещания. "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26 - 30 апреля 1971.- Киев, 1971.- С 54-64.
16. Шквар А. М. Функциональные и структурные аспекты построения некоторых нейронных структур управления // Тр. 4-го Всесоюз. совещания "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С. 78-89.
17. Werbos P. J. Beyond regression: New tools for prediction and analysis in the behavioral sciences. PhD Thesis, Harvard University, Cambridge, MA. – 1974.– P. 124-137.
18. Rumelhart D. E., Hinton G E., Williams R. J. Learning internal representation by error propagation // In: D.E. Rumelhart, J.L. McClelland (Eds.) Parallel Distributed Processing, Vol. I Foundations. – Cambridge, MA: MIT Press, 1986. – P. 318-362.
19. Rumelhart D. E., Hinton G. E., Williams R. J. Learning representation by back-propagating errors // Nature. – 1986. – Vol.323. – P. 533-536.
20. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // IEEE Trans. Neural Networks. – 1990. – Vol.1. – №1. – P. 4-27.

21. Narendra K.S., Parthasarathy K. Gradient methods for the optimization of dynamical systems containing neural networks // IEEE Trans. Neural Networks. – 1991. – Vol.2. – № 2. – P.252-262.
22. Sunner R. M., Slotine J. E. Gaussian Networks for Direct Adaptive Control // IEEE Trans. Neural Networks. – 1992. – Vol. 3. № 6. – P. 837 - 863.
23. Handbook of Intelligent Control: Neural, Fuzzy and Adaptive Approaches / Ed. by David A. White & Donald A. Sofge. N.-Y. Van Nostrand Reinhold. – 1992. – 568 p.
24. Neural networks for control systems: A survey K. J. Hunt, D. Sbarbaro, R. Zbi-kowski, P. J. Gawthrop // Automatica. – 1992. – Vol. 28. – № 6. P. 1083-1112.
25. Suykens Johan A. K., Vandewall Joos P. L., De Moor Bart L. R. Artificial Neural Networks for Modelling and Control of Non Linear Systems. Kluwer Academic Publishers. Boston, Dordrecht, London, 1997. – 235 p.
26. Kosko B. Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence. - New Jersey: Prentice-Hall Englewood Cliffs, 1991. – 375 p.
27. Vas, P: Artificial-Intelligence-Based Electrical Machines and Drives. Application of Fuzzy, Neural, Fuzzy-Neural and Genetic Algorithms, Oxford University Press, Monographs in Electrical and Electronic Engineering, ISBN 0-19-859397-X. – 1999. – 626 p.
28. Efe M.O., Fiskiran A.M., Kaynak O. Derivation of a Parameter Stabilizing Training Criterion for Adaptive Neuro-Fuzzy Inference Systems in Motion Control // International Journal of Systems Science. – 2001. – Vol.32. – №:4. – P. 513-521.
29. Werbos, P.J. Neurocontrol and related techniques // In Handbook of Neural Computing Applications. Hrsg. Maren,A.; Harston, C; Pap, R. Academic Press. San Diego, 1990. – P. 345-380.
30. Werbos P. J. Backpropagation and neurocontrol: A review and prospectus // Proc. of International Joint Conf. On Neural Networks. Washington, DC.–1989.–Vol. 1.– P.209-216.

31. Holland J. H. Adaptation in natural and artificial systems. An introductory analysis with application to biology, control, and artificial intelligence. – London: Bradford book edition, 1994. – 211 p.
32. Клепиков В.Б., Махотило К.В., Сергеев С.А., Вороновский Г.К. Искусственные нейронные сети: новая парадигма в управлении // Труды межд. конф. "Проблемы автоматизированного электропривода. Теория и практика". – Харьков: Основа. – 1995. – С. 111 – 115.
33. Вороновский Г.К., Махотило К.В., Петрашев С.Н., Сергеев С.А. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. – Харьков: Основа. – 1997. – 112 с.
34. Клепиков В.Б., Махотило К.В., Обруч И.В. Синтез нейросетевой: системы управления одномассовой электромеханической системы с отрицательным вязким трением при ограничении координат электропривода // Труды международной конференции "Проблемы автоматизированного электропривода. Теория и практика", Харьков: Основа. – 1997. – С. 19-21.
35. Клепиков В.Б., Полянская И.С. Синтез двухмассовой электромеханической системы с отрицательным вязким трением при регулировании по полному и неполному вектору состояния // Вісник Національного технічного університету "Харківський політехнічний інститут".-Харків:НТУ "ХПІ".-2001.-Вип.10.-С.36-40.
36. Клепиков В.Б., Палис Ф., Цепковский Ю.А. Гибридные нейронные сети в управлении электромеханическими системами // Вестник НТУ "ХПИ". Проблемы автоматизированного электропривода. Теория и практика, Харьков. – 2003. – Вып.10. — Т.1. – с. 29-33.
37. Клепиков В. Б. Моисеенко П. Л. Цепковский Ю. А. Синтез гибридной нейронной сети для двухмассовой ЭМС при срывных фрикционных автоколебаниях //Вестник НТУ "ХПИ": Сб. научн. трудов "Проблемы автоматизированного электропривода. Теория и практика". – Харьков, 2003. – Вып.10.- Т.1. – С. 512-514.
38. Бодянский Е.В., Руденко О.Г. Искусственные нейронные сети: архитектуры, обучение, применения. – Харьков: Телетех, 2004. – 264 с.

39. Astrom K J , Wittenmark B Adaptive control, Reading MA Addison Wesley, 1989.
40. Parks P C Lyapunov redesign of model reference adaptive control systems, IEEE Journal Transactions of Automatic Control, 1966, vol 11, pp 362-367.
41. 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.
42. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
43. Уоссерман Ф Нейрокомпьютерная техника Теория и практика - М Мир, 1990.
44. Haykin S , Neural Networks A comprehensive foundation, MacMillan College Publishing Co , New York, 1994.
45. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
46. 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.
47. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
48. 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.
49. 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.
50. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
51. Solowey D. Neural Generalized Predictive Control / Solowey D., Haley P.J //Proceedingof the 1996 IEEE Symposium on Intelligent Control. – 1996. – P. 277–282.

52. Clarke D.W., Mohtadi C., Tuffs C. Generalized Predictive Control - Part 1: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 137-148.
53. Clarke D. W., Mohtadi C., Tuffs C. Generalized Predictive Control. Part 2: The Basic Algorithm // *Automática*. – 1987. – Vol. 23. – P. 149-163.
54. Clarke D. W. Advances in model-based predictive control.- In *Advances in Model-Based Predictive Control* / D. W. Clarke: Oxford University Press. – 1994. – 274
55. 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.
56. 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
57. 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.
58. 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.
59. Press W. H., Flannery B. P., Teukolsky S. A., Vetterling V. T. Numerical Recipes.- *The Art of Scientific Computing*. Cambridge University Press, 1988. – 452.
60. Rumelhart D. E., McClelland J. L. and the PDP Research Group. *Parallel Distributed Processing*. The MIT Press. – 1986. – Vol.1. – Chapter 8. – 337 p.
61. Яуре А.Г., Певзнер Е.М. Крановый электропривод: Справочник. – М.: Энергоатомиздат, 1988. – 314 с.
62. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
63. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
64. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.

65. 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.
66. 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.
67. 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.
68. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
69. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
70. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
71. 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
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. Ч1.
73. 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 інш. – К.: Либідь, 2005. – 680 с. Ч2.
74. 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.
75. Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрьоменко. Харків, ХНАДУ, 2012. – 284 с.
76. 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.
77. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
78. 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.
79. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 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 багатшарові нейронні мережі (багатшарові персептрони по Ф. Розенблатту), де штучні нейрони розташовані шарами.

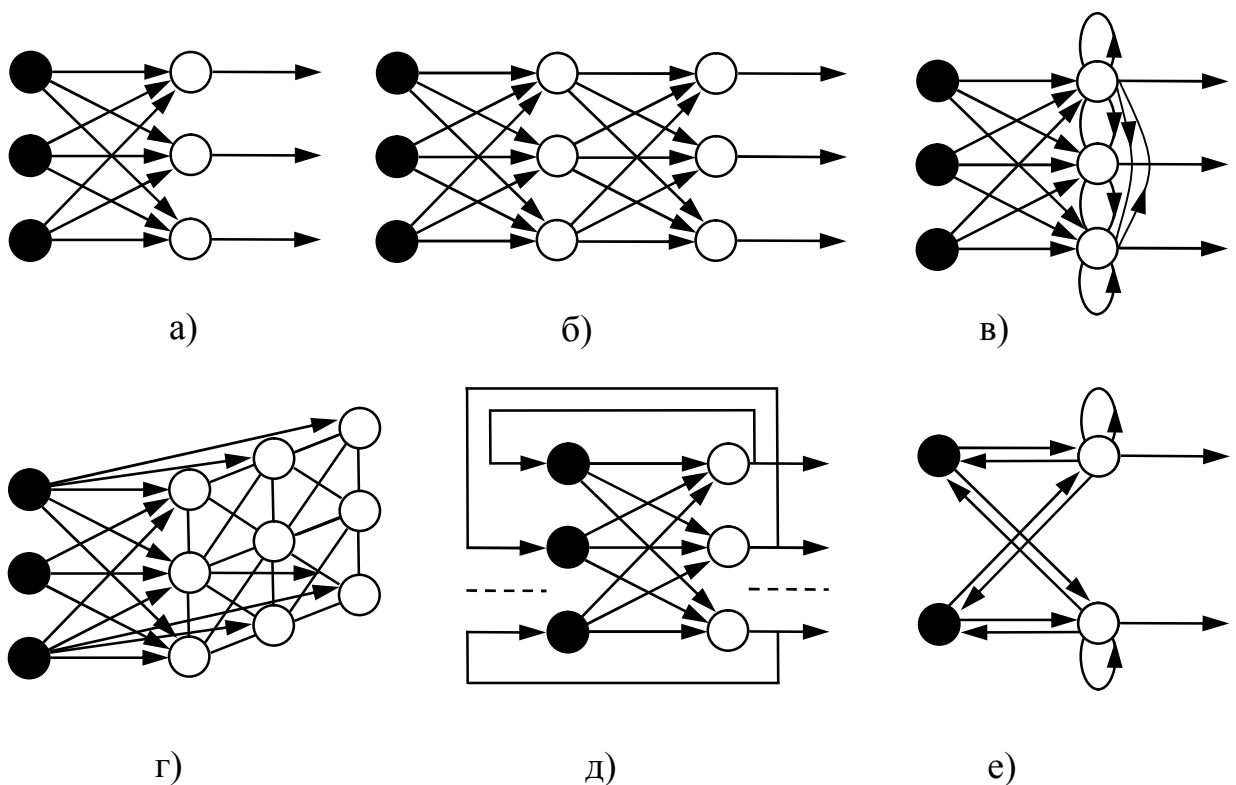


Рисунок А.1 – Структури найбільш поширених ШНМ

а – одношарова мережа; б – багатшарова мережа (персептрон); 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):

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

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

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

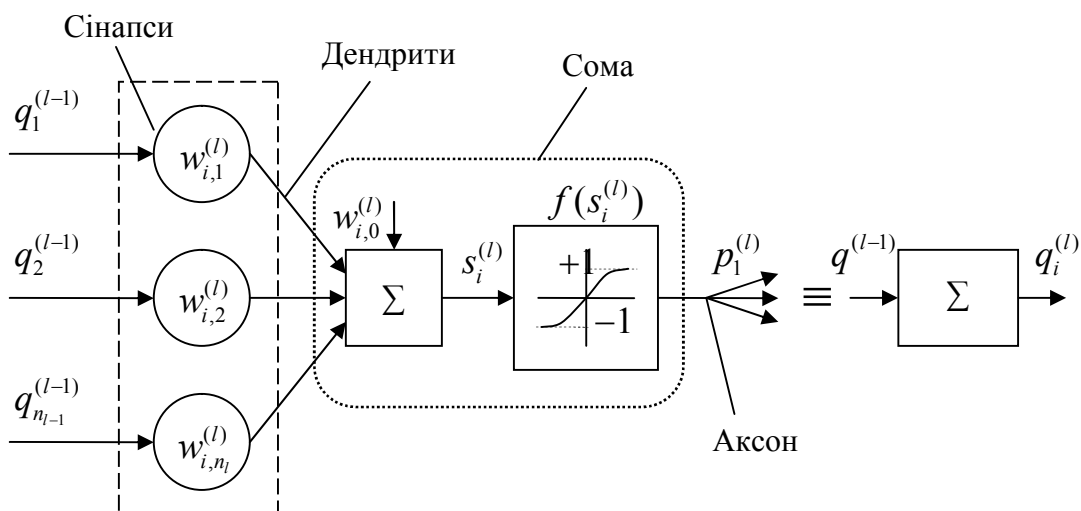


Рисунок А.2 – Схема базового процесорного елементу (штучного нейрона)

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 $\mathbf{q} \in R$.

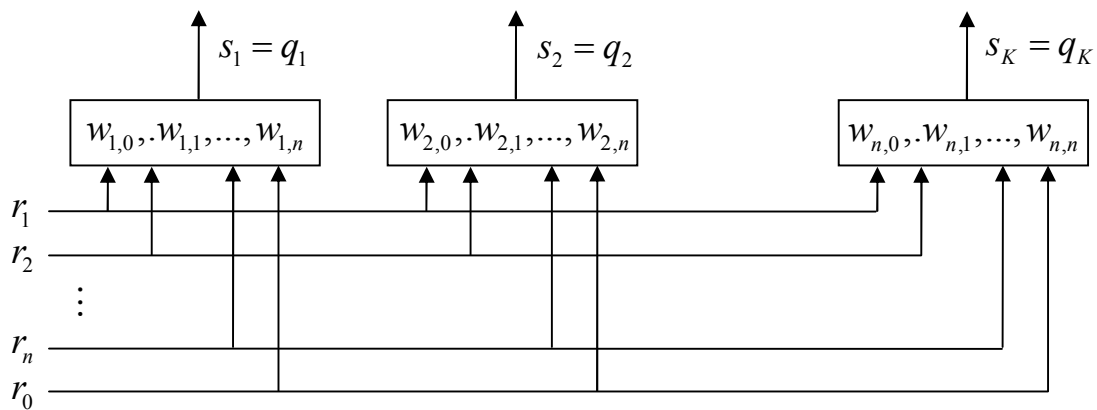


Рисунок А.3—схематичне представлення одношарової лінійної штучної нейронної мережі

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.

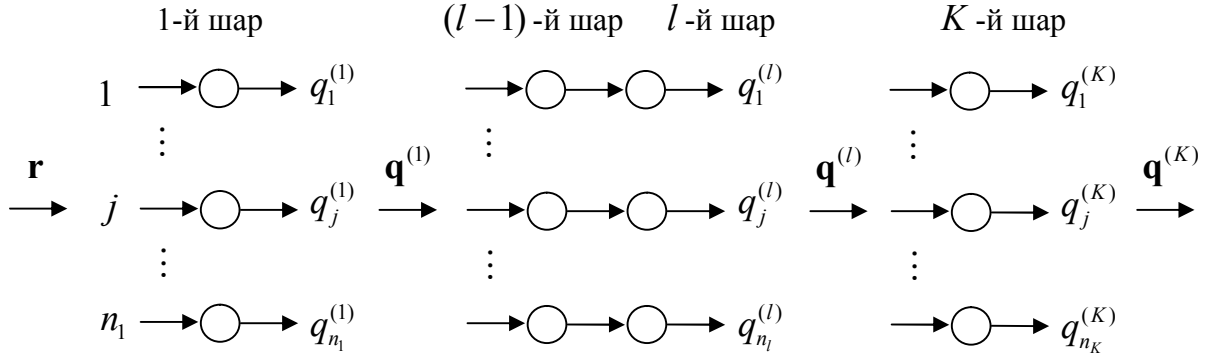


Рисунок А.4 – Структурна схема статичних БНМ

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 багатошарових нейронних мереж з базовими елементами як першого, так і другого порядків. У подальшому будемо розглядати багатошарові 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 функцій 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 входи 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 багат шарових 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 Вейєрштрасса про рівномірне наближення безперервних функцій поліномами. Тому теорема Стоуна-Вейєрштрасса 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.

Багат шарова 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:

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

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 елементу 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 елементу. 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 $w_{i,j}^{(l)}$ ($l = 1, \overline{K-1}$) 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 настраюються, 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:

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

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 (В.6) і алгоритми настройки базових елементів шарів $l = \overline{K-1, 1}$ використовують формулу (В.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 елемента 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 [51]. 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) [52-54]. 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 [52]. 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.

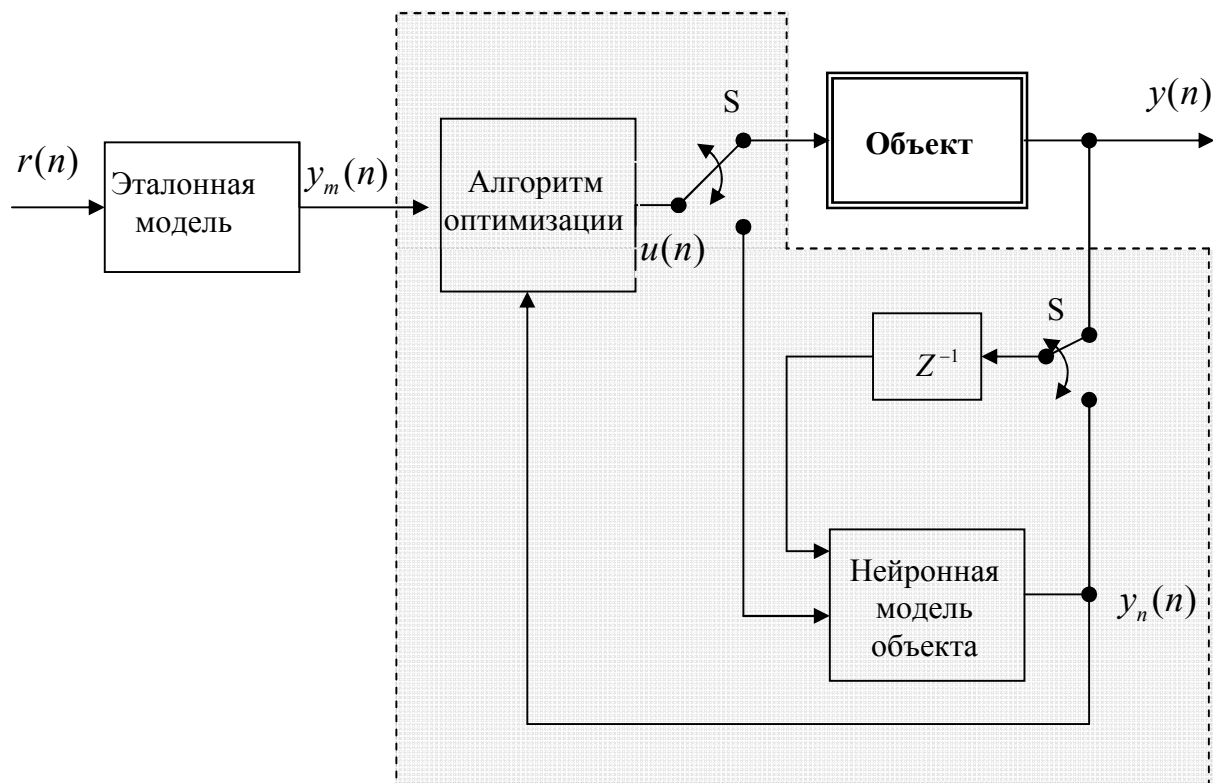


Рисунок Г.1 – Схема системи узагальненого нейроуправління з прогнозом

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 в основному залежить від вибору мінімізуючого алгоритму для CFM блоку. Існують декілька алгоритмів мінімізацій, які реалізуються в CFM, такі як: Non-gradient [55], Simplex [56] і Successive Quadratic Programming [57, 58]. 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 [57, 58]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

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

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

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 [59] 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 [59]: 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 Ньютон-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 настроюються 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 [60]. Цей процес повторюється до тих пір, поки помилка не досягне допустимого рівня.

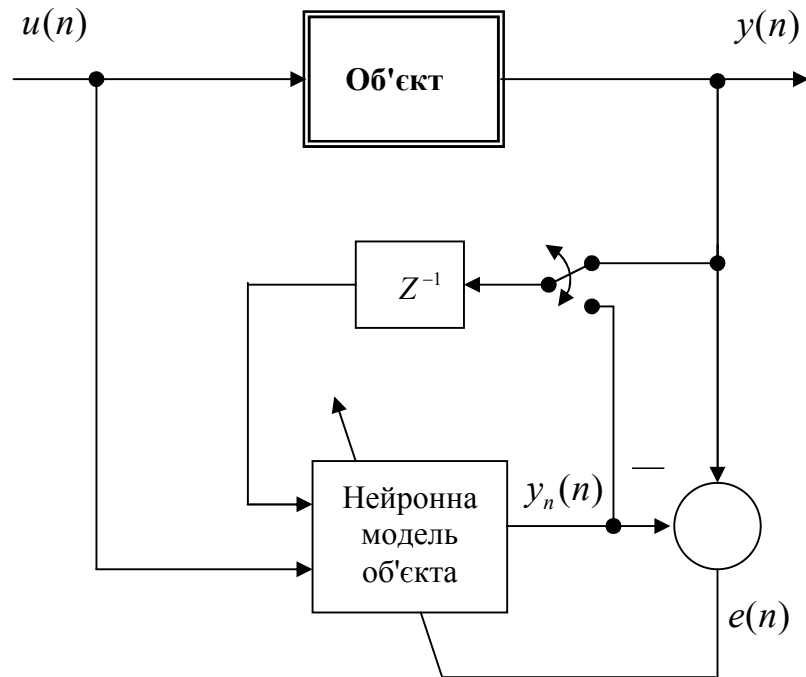


Рисунок Г.2 – Схема тренування мережі в режимі offline

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.

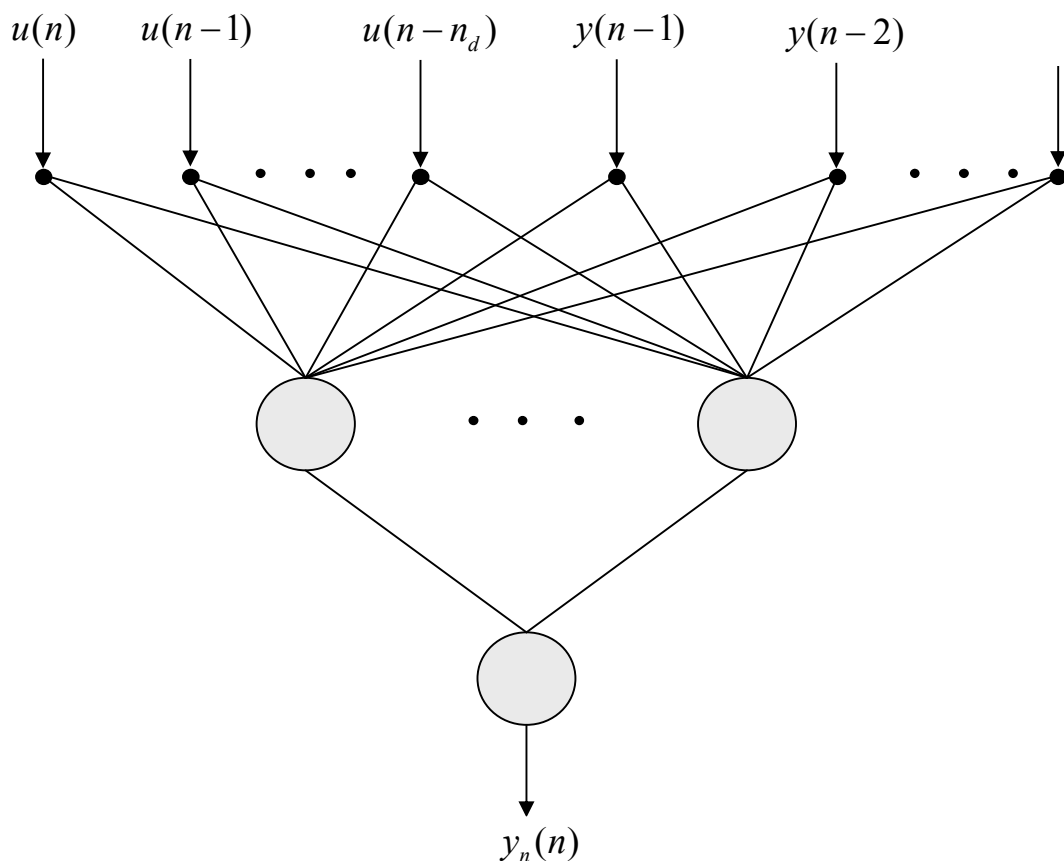


Рисунок Г.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

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

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

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

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

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

i

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

Рівняння (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
% механізму переміщення візка мостового крана
% 2024 рік

% Вихідні дані
kd=0.42;
In=57;
Mn=In/kd;
ktp=27;
kn=0.1091;
kc=0.231;
Tmc=5e-3;
Ta=0.036;
Te=0.024;
Re=0.61;
b12=0;
c12=3.29;
Jd=0.97;
Jm=2.95;
Js=Jd+Jm;
Uz=24;
Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);

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