

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

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

КВАЛІФІКАЦІЙНА РОБОТА

магістра

на тему

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ МЕХАНІЗ-
МОМ ПЕРЕМІЩЕННЯ ВІЗКА ЗЛИВКОВОЗА КІЛЬЦЕВОЇ
ЗЛИВКОПОДАЧІ СТАНА ГАРЯЧОЇ ПРОКАТКИ

(тема кваліфікаційної роботи)

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

(код і найменування спеціальності)

_____ / Владислав КУЛЬБАКА
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

Харків – 2024

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

Навчально-науковий інститут «Українська 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) Владислав КУЛЬБАКА
(ім'я, прізвище)

1. Тема роботи Синтез нейромережевої системи управління механізмом переміщення візка зливковоза кільцевої зливкоподачі стана гарячої прокатки
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 Маса візка зливковоза $m_v=38 \cdot 10^3$ кг; маса зливка $m_{зл}=2 \cdot 12,5 \cdot 10^3$ кг; діаметр ходових коліс $D_k=1,1$ м; середній діаметр цапф підшипників коліс $D_{ц}=0,2$ м; ККД редуктора, $\eta=0,92$; момент інерції частин зливковоза, що обертаються $J_{об}=970$ кг·м²; шлях руху зливковоза $S=280$ м; відстань від найбільш видаленого колодязя до рольгангу $S_{пр}=120$ м; робоча швидкість $V_{роб}=4$ м/с; максимальна швидкість на холостій гілці шляху, $V_{max}=V_{роб}=4$ м/с; швидкість першого ступеня при гальмуванні, $V_1=3,3$ м/с; швидкість на закругленні перед зіштовхувачем $V_3=2,5$ м/с. Система управління – астатична. Перерегулювання не більше 10%, час регулювання до 10с.
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму переміщення візка зливковоза стана гарячої прокатки. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити динамічні характеристики системи. Розробити математичну модель двомасової системи. Виконати моделювання на ЕОМ. Розробити структурну схему нейромережевої системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання системи з нейрорегулятором.
5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістрська кваліфікаційна робота. Формальні ознаки. 2. Візок зливковоза. Схема кінематична 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Системи нейрорегулювання. Структурні схеми. 6. Управління з прогнозом. Математична модель. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. 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)

Тетяна ВАСИЛЕЦЬ
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Владислав КУЛЬБАКА
(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 зливковоза кількової зливкоподачі стана гарячої прокатки.	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 пружних елементів. Розробка математичної моделі двомасової системи. Розрахунок перехідних процесів двомасової систем на ЕОМ	16.10.2023-05.11.2023	
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)

Владислав КУЛЬБАКА
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 214 сторінок, 60 рисунків, 3 таблиці, 92 використаних джерела, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом візка зливковогозана гарячої прокатки 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 зливковогозана гарячої прокатки. 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 ННІ «УІПА» (Харків, 2024 р.).

Ключові слова: ШТУЧНИЙ НЕРОН, НЕЙРОННІ МЕРЕЖІ, ПРОКАТНИЙ СТАН, ЗЛИВКОВОЗ, ЕЛЕКТРОПРИВОД ВІЗКА ЗЛИВКОВОЗА, СИСТЕМА ПІДЛЕГЛОГО УПРАВЛІННЯ, КОНТУР СТРУМУ, КОНТУР ЕРС, ОДНОМАСОВА І ДВОМАСОВА СИСТЕМИ, НЕЙРОМЕРЕЖЕВА СИСТЕМА, НЕЙРОРЕГУЛЯТОР

ABSTRACT

An explanatory message consists of: 214 pages, 60 figures, 3 tables, 92 used sources, 5 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the hot rolling mill ingot carrier 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 hot rolling mill ingot carrier. The control system of coordinates of mechanism electric drive is worked out. The parameters of automatic control system are calculated. The simulation of the system is executed on a computer.

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

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

The main provisions of the master's thesis are used in the educational process of the Department of AMET of the Educational and Scientific 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 ESI "UIPA" (Kharkov, 2024 y.).

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, ROLLING MILL, INGOT CARRIER, ELECTRIC DRIVE OF INGOT CARRIER, SUBORDINATE CONTROL SYSTEM, CURRENT CIRCUIT, EMF CIRCUIT, ONE-MASS AND TWO-MASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. ОГЛЯД СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Розвиток теорії нейронних мереж	14
1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж	20
1.3 Нейрокерування динамічними об'єктами	24
1.4 Огляд структур нейромережових систем управління. Вибір структури системи	27
1.5 Призначення і основні характеристики зливковозів	34
1.6 Постановка задачі дослідження	36
ВИСНОВКИ ДО РОЗДІЛУ 1	39
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА ЗЛИВКОВОЗА СТАНА ГАРЯЧОЇ ПРОКАТКИ	
2.1 Вибір системи електроприводу зливковоза	43
2.3 Розрахунок і побудова швидкісної діаграми електроприводу зливковоза	44
2.4 Вибір електродвигуна по потужності	52
2.5 Побудова діаграми навантаження електроприводу	54
2.6 Перевірка двигуна по нагріву і перевантажувальній здатності	57
2.7 Вибір основного електроустаткування силового кола 2.7 Вибір основного електроустаткування	58
2.8 Розрахунок параметрів силового кола електроприводу	59
2.9 Вибір структури системи автоматичного регулювання координат електроприводу	63
2.10 Синтез послідовного корегуючого пристрою контура струму	65
2.10 Синтез послідовного корегуючого пристрою контура ЕРС	69

2.11 Дослідження динамічних характеристик системи	76
ВИСНОВКИ ДО РОЗДІЛУ 2	81
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ.....	83
3.1 Математична модель двомасової системи	83
3.2 Розрахунок перехідних процесів двомасової системи.....	92
ВИСНОВКИ ДО РОЗДІЛУ 3	98
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	99
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	99
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	101
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	104
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	147
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	149
ВИСНОВКИ ДО РОЗДІЛУ 4	1565
ВИСНОВКИ	157
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	159
ДОДАТОК А. МОДЕЛІ НЕЙРОНА.....	167
А.1 Структурна схема нейрона.....	167
А.2 Функції активації.....	168
А.3 Нейрон з векторним входом.....	178
ДОДАТОК Б. АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ.....	173
Б.1 Одношарові мережі.....	173
Б.2 Багатошарові мережі	175
Б.3 Мережі з прямою передачею сигналу	186
ДОДАТОК В. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ	180
В.1 Сутність процесу навчання	180

В.2 Алгоритми навчання нейронних мереж.....	182
В.3 Метод зворотного розповсюдження помилки.....	186
В.4 Перенавчання і узагальнення нейронних мереж.....	197
ДОДАТОК Г. УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ	
З ПРОГНОЗОМ	200
Г.1 Алгоритм нейроуправління з прогнозом.....	200
Г.2 Структура нейронної мережі для моделювання об'єкту.....	207
Г.3 Прогноз з використанням нейронної мережі.....	211
Г.4 Похідні рівнянь нейронної мережі.....	212
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	214

ВСТУП

Актуальність теми. Рівень розвитку народного господарства значною мірою залежить від обсягу виробництва металу. Чавун, сталь та прокат є основними видами продукції металургійної промисловості. Чавун і сталь використовуються для виготовлення верстатів, виробничих і сільськогосподарських машин. Прокатні вироби, такі як залізничні рейки, будівельні балки, різні труби, дріт, сортова та листова сталь, мають широке застосування у всіх галузях економіки. Відсутність продукції металургійного виробництва унеможлиблює існування та розвиток сучасної незалежної держави, а високорозвинута металургія є необхідною умовою для швидкого розвитку народного господарства. Ефективність роботи металургійних агрегатів та якість продукції значною мірою залежать від вдосконаленості електроприводів як основних, так і допоміжних механізмів. Тому особлива увага приділяється систематичному покращенню силової частини приводу, а також засобам і методам управління та автоматизації.

Високі вимоги до систем управління електроприводами зумовлюють необхідність розробки нових принципів їх побудови, розрахунків, досліджень і налаштування. Ускладнення технічних процесів, збільшення їх швидкості та необхідність точного виконання виключають можливість ручного управління. У таких умовах оператор не здатен ефективно контролювати і управляти великою кількістю взаємозв'язаних параметрів без використання комп'ютерних систем. Розвиток промислової електроніки та обчислювальної техніки створив нові елементи для систем автоматичного управління, що відповідають підвищеним вимогам.

Одним із найбільш перспективних напрямків є розробка систем управління на основі адаптивного підходу в поєднанні з методами теорії штучних нейронних мереж (ШНМ) [1-12].

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 штучних нейронних мереж (ШНМ) пов'язаний із розумінням того, що функціонування мозку живого організму відрізняється від роботи комп'ютера. Людський мозок є складною нелінійною паралельною системою, здатною до інформаційного управління, мислення, накопичення і відновлення інформації, а також до розв'язання проблем. Він складається з однотипних структурних одиниць – нейронів, які є базовими елементами обробки сигналів, отримуючи та комбінуючи інформацію від інших нейронів через дендрити. З інженерної точки зору, ШНМ – це система паралельної обробки інформації, побудована з взаємопов'язаних простих обчислювальних вузлів (однорідних або різних), яка має здатність накопичувати досвід, узагальнювати його та представляти у вигляді, зручному для інтерпретації та ухвалення рішень.

На сьогодні є всі підстави говорити про досягнення певних успіхів нейромережевих технологій в рішенні складних задач як чисто наукових, так і у

сфері техніки, бізнесу, фінансів, медичної діагностики і інших областях, пов'язаних з інтелектуальною діяльністю.

Тому тема роботи, яка присвячена розробці нейромережевої системи управління електроприводом візка зливкового кільцевої зливкоподачі стана гарячої прокатки з урахуванням пружних елементів є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління механізмом переміщення візка зливкового кільцевої зливкоподачі стана гарячої прокатки для підвищення показників якості функціонування системи на основі нейромережевих регуляторів. Досягнення поставленої мети здійснюється рішенням наступних основних задач.

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

Об'єктом дослідження є динамічні процеси в системі управління механізмом переміщення візка зливкового кільцевої зливкоподачі стана гарячої прокатки з урахуванням пружних елементів.

Предметом дослідження є моделі і методи синтезу нейромережевої системи управління двомасовою електромеханічною системою механізму переміщення візка зливкового кільцевої зливкоподачі стана гарячої прокатки з урахуванням податливості механічних зв'язків.

Методи дослідження. Всі теоретичні положення роботи базуються на фундаментальних положеннях теорії автоматичного управління. Методи математичного моделювання використані для побудови математичних моделей роботи системи. Методи теорії штучних нейронних мереж дозволили синтезувати

нейромережеву модель об'єкту управління і вибрати процедуру її навчання. Методи нейроуправління 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 зливковоза кільцевої зливкоподачі стана гарячої прокатки.

2. Вперше запропоновано і синтезовано нейромережеву систему управління механізмом переміщення візка зливковому кільцевої зливкоподачі стана гарячої прокатки з урахуванням пружних елементів, що має високі показники якості функціонування.

3. Вперше запропонована математична модель системи управління механізмом переміщення візка зливковоза кільцевої зливкоподачі стана гарячої прокатки з урахуванням пружних елементів 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 зливковозакільцевої зливкоподачі стана гарячої прокатки, що забезпечує високоякісне регулювання.

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

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

Серед них: синтезована система регулювання координат електроприводу; розроблена модель системи з урахуванням пружних елементів, розроблена структурна схема нейромережевої системи; синтезований нейрорегулятор NN Predictive Controller з використанням пакету прикладних програм Neural Network Toolbox системи MATLAB; виконано моделювання нейромережевої систем і проведено аналіз отриманих результатів.

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 214 сторінок, серед них: 37 рисунків за текстом, 23 рисунки на окремих сторінках, 3 таблиці, 5 додатків на 48 сторінках, список літературних джерел з 92 найменування на 8 сторінках.

РОЗДІЛ 1

ОГЛЯД СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Розвиток теорії нейронних мереж

1.1.1 Історичний аспект

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

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

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

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

У 1950–1960-х роках група дослідників, об'єднавши біологічні та фізіологічні підходи, створила перші штучні нейронні мережі. Спочатку вони були реалізовані у вигляді електронних схем, але згодом перенесені у середовище комп'ютерного моделювання, яке використовується й нині. Перші успіхи викликали хвилю активності та оптимізму. Такі вчені, як Мінський, Розенблатт, Уїдроу та інші, розробили одношарові нейронні мережі, відомі як персептрони. Вони застосовувалися для розв'язання широкого кола завдань, таких як прогнозування погоди, аналіз електрокардіограм і моделювання штучного зору. У той час здавалося, що ключ до створення інтелекту знайдено, і побудова мозкоподібної системи - лише питання створення достатньо великої мережі.

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

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

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

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

У 1980-х роках теорія нейронних мереж почала активно застосовуватися в прикладних сферах, і з'явилися нові компанії, що займалися комерціалізацією цієї технології. Інтерес до штучних нейронних мереж зростав вибуховими темпами. У 1987 році було проведено чотири великих конференції, присвячені нейронним мережам, і опубліковано понад 500 наукових праць, що свідчило про феноменальний розвиток галузі.

1.1.2 Сучасний стан штучних нейронних мереж

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

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

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

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

Ця галузь вже довела свою ефективність і демонструє унікальні можливості, проте залишається багато обмежень і нерозв'язаних питань. Для

покращення існуючих мереж потрібні серйозні зусилля. Необхідно розробляти нові технології, вдосконалювати методи навчання та зміцнювати теоретичну базу, щоб нейронні мережі змогли повністю реалізувати свій потенціал.

1.1.3 Перспективи розвитку

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

Штучні нейронні мережі та експертні системи

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

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

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

Можна уявити штучну систему, що наслідуює такий принцип поділу праці. Штучна нейронна мережа може забезпечувати швидкі реакції в більшості стандартних ситуацій. Завдяки здатності вказувати рівень впевненості у своїх рішеннях, така мережа може «усвідомлювати», коли їй бракує інформації, і передавати складніші завдання експертній системі.

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

Питання надійності

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

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

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

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

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

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

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

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

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

1.2 Ідентифікація динамічних об'єктів з застосуванням нейронних мереж

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

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

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

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

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

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

ξ – завада вимірювання вихідного сигналу.

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

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

Тоді рівняння (1.1) може бути записане таким чином:

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

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

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

Найбільшого застосування для вирішення завдання ідентифікації нелінійних динамічних об'єктів (1.1) 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

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

де q – кількість куль в мережі;

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

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

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

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

де c_i – вагові коефіцієнти, які підлягають визначенню.

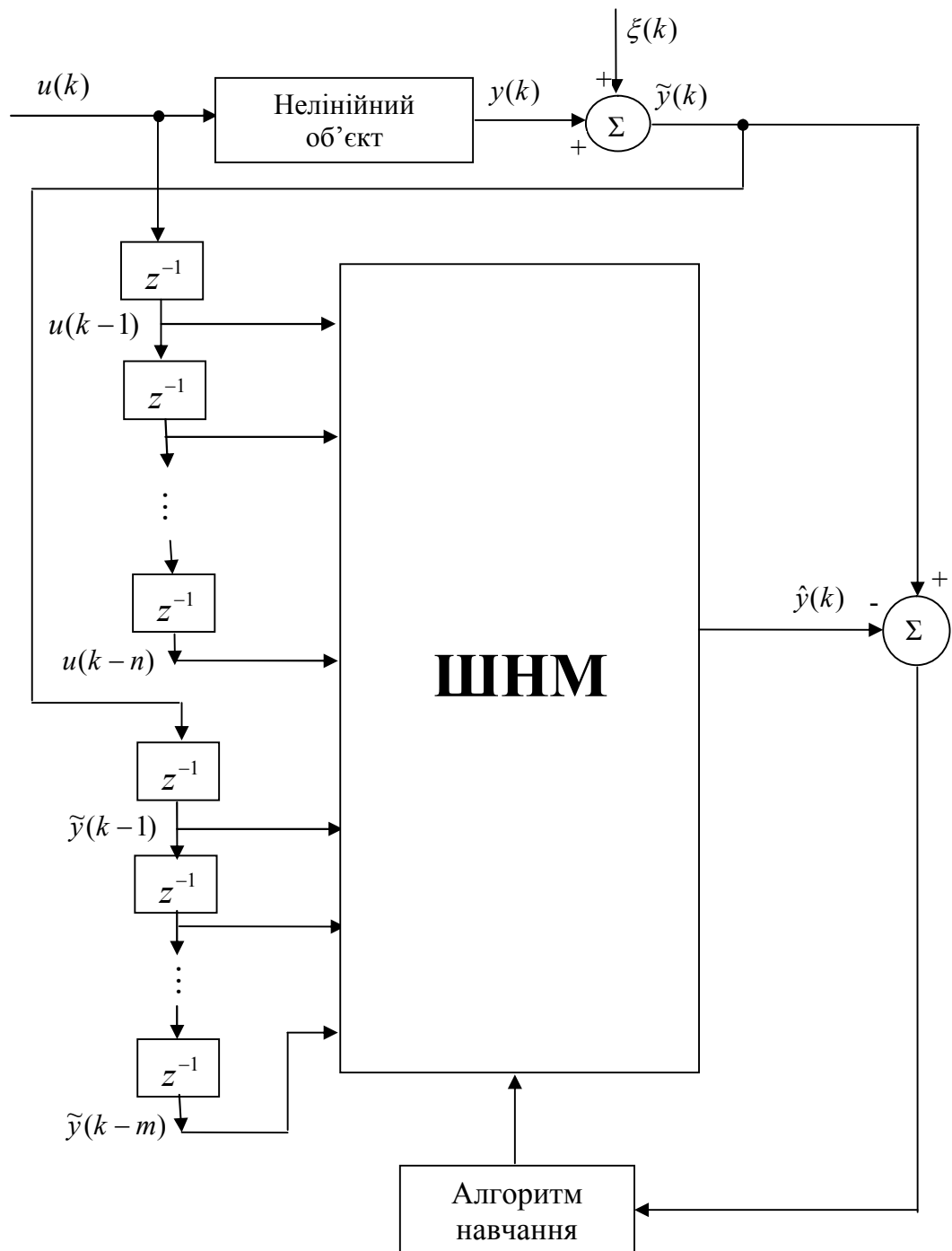


Рисунок 1.1 – Схема ідентифікації динамічного об'єкту

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

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

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 $\Phi_0(x) = 1$).

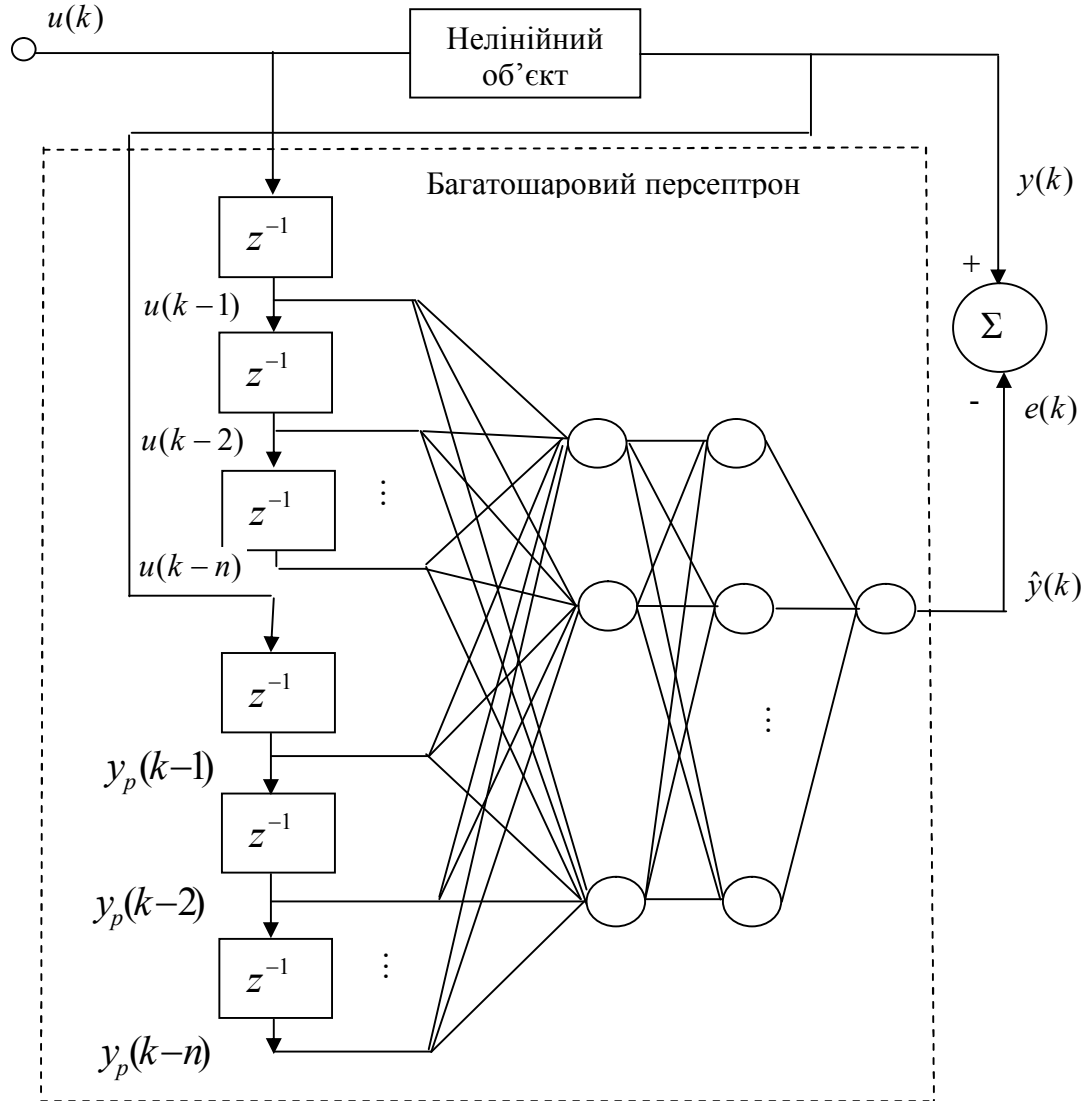


Рис 1.2 – Ідентифікація динамічного об'єкту за допомогою багатошарового персептрона

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

$$J = e^2(k) = M \left\{ [\tilde{y}(k) - \hat{y}(k)]^2 \right\}. \quad (1.2)$$

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

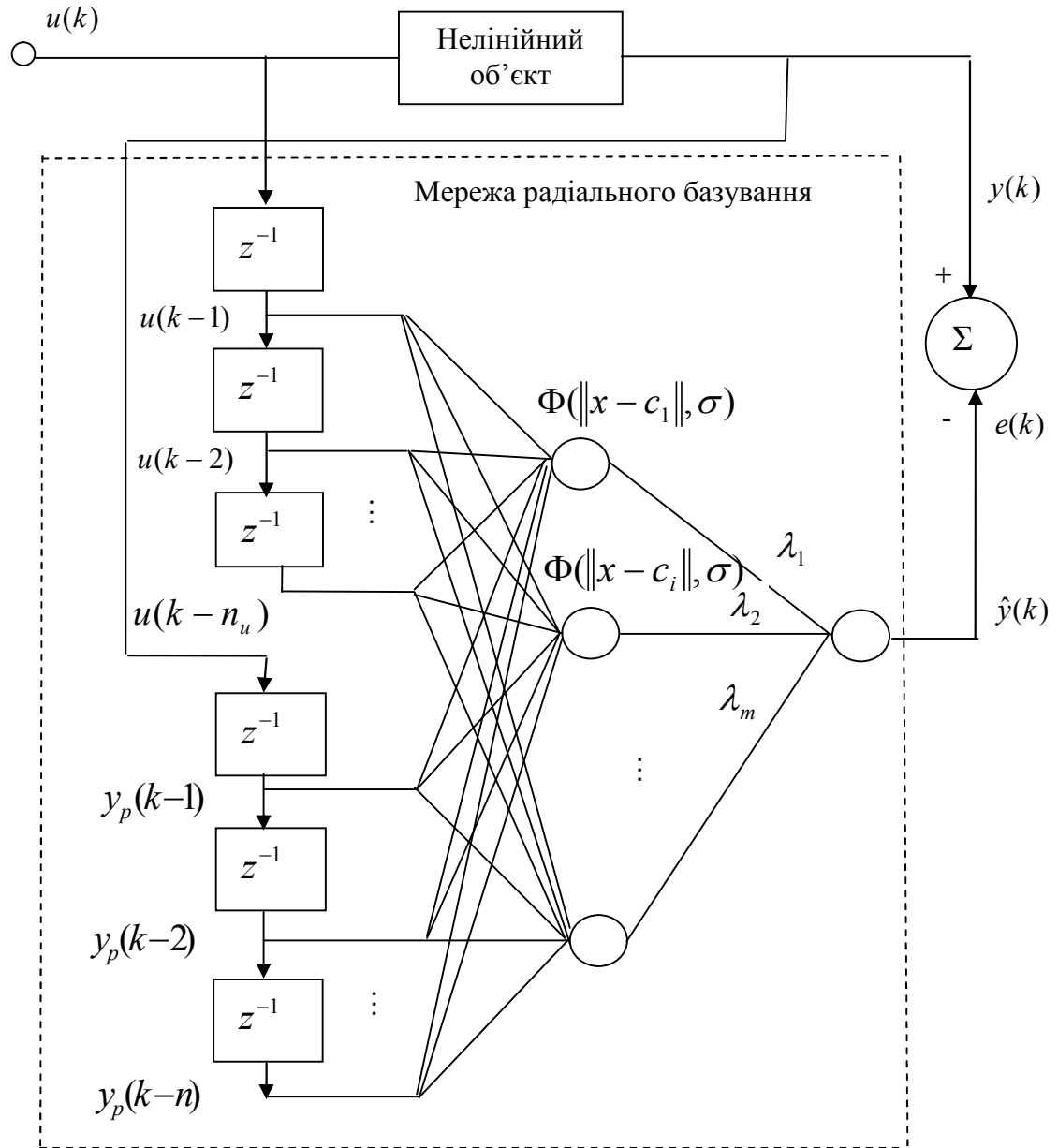


Рисунок 1.3 – Ідентифікація динамічного об'єкту за допомогою РБМ

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.

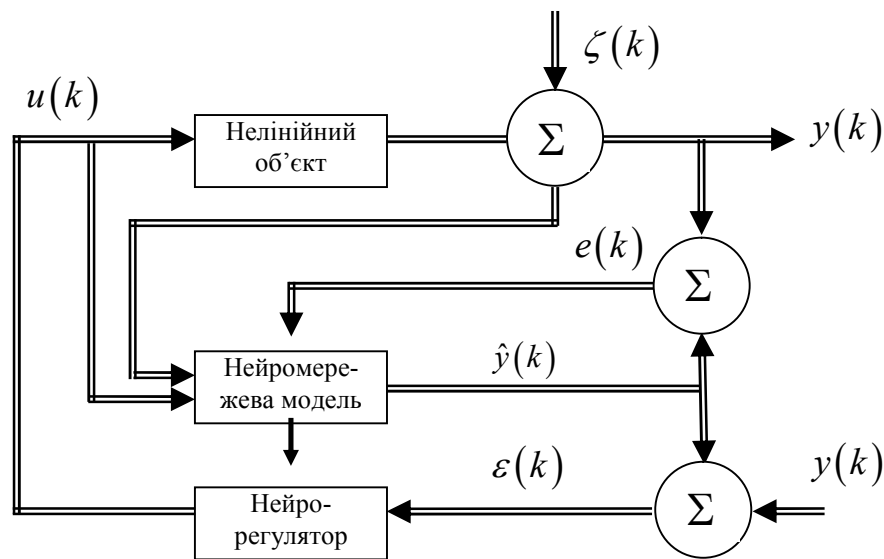


Рисунок 1.4 – Структура нейромережевої системи адаптивного управління

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

Розглянемо багатовимірний нелінійний динамічний об'єкт, описуваний рівнянням

$$y(k) = f[y(k-1), \dots, y(k-m), u(k-1), \dots, u(k-n)] + \xi(k), \quad (1.3)$$

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

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

перепишемо рівняння (1.3) у вигляді

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

Навчання мережі, до якого зводиться навчання ідентифікації, полягає в настроюванні її параметрів на основі порівняння вихідних сигналів об'єкта $y(k)$ і моделі $\hat{y}(k)$ та в мінімізації звичайно квадратичного функціонала помилки (1.2).

Після завершення процесу навчання мережа використовується для реалізації алгоритму керування.

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

$$J(k) = \|\varepsilon(k)\|^2 = \frac{1}{2} \|y^*(k) - \hat{y}(k)\|^2, \quad (1.4)$$

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

Як й у випадку ідентифікації, для знаходження мінімуму функціонала (1.4) можуть бути використані різні рекурентні алгоритми, зокрема, градієнти вигляду

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

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 нейромоделі. Нейронна мережа відновлює при цьому елементи матриць моделі об'єкту. Це так звані структуровані методи регулювання. Проте повністю використовувати весь потенціал апроксимації нейронних мереж дозволяють неструктуровані методи регулювання. Далі стисло розглядаються структури систем нейрорегулювання, актуальні на сьогоднішній день.

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

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

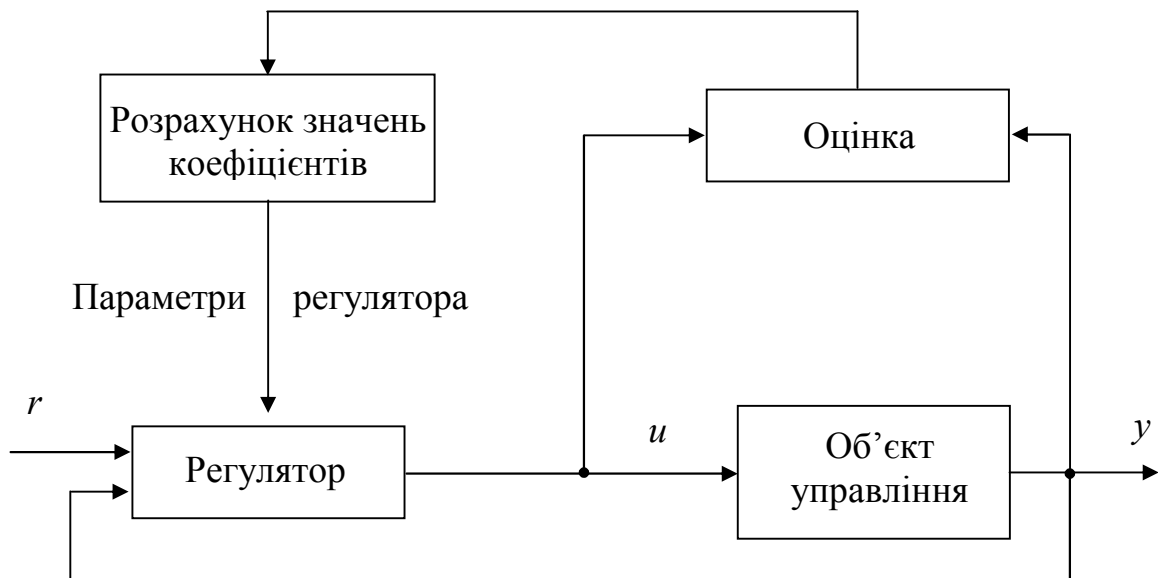


Рисунок 1.5 – Блок-схема управління із зворотним зв'язком і регульованими коефіцієнтами

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

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

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

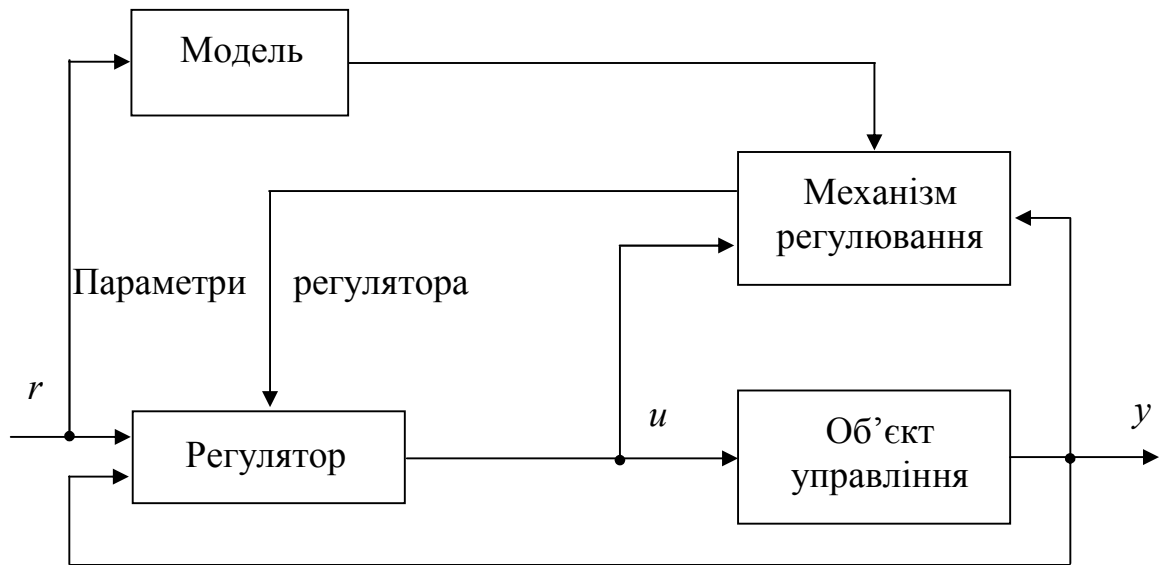


Рисунок 1.6 – Блок-схема адаптивного управління з еталонною моделлю

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

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

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

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

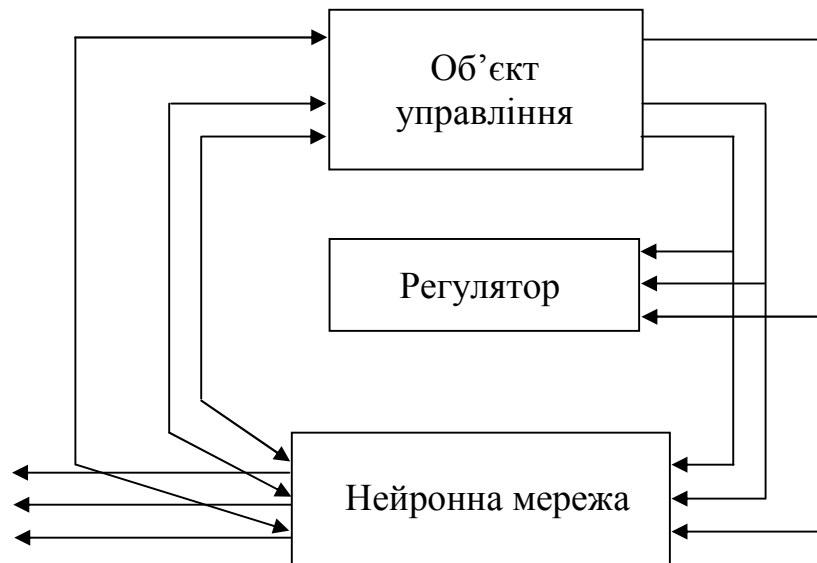


Рисунок 1.7 – Нейромережева СУ, заснована на «копіюванні» існуючого контролера

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

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

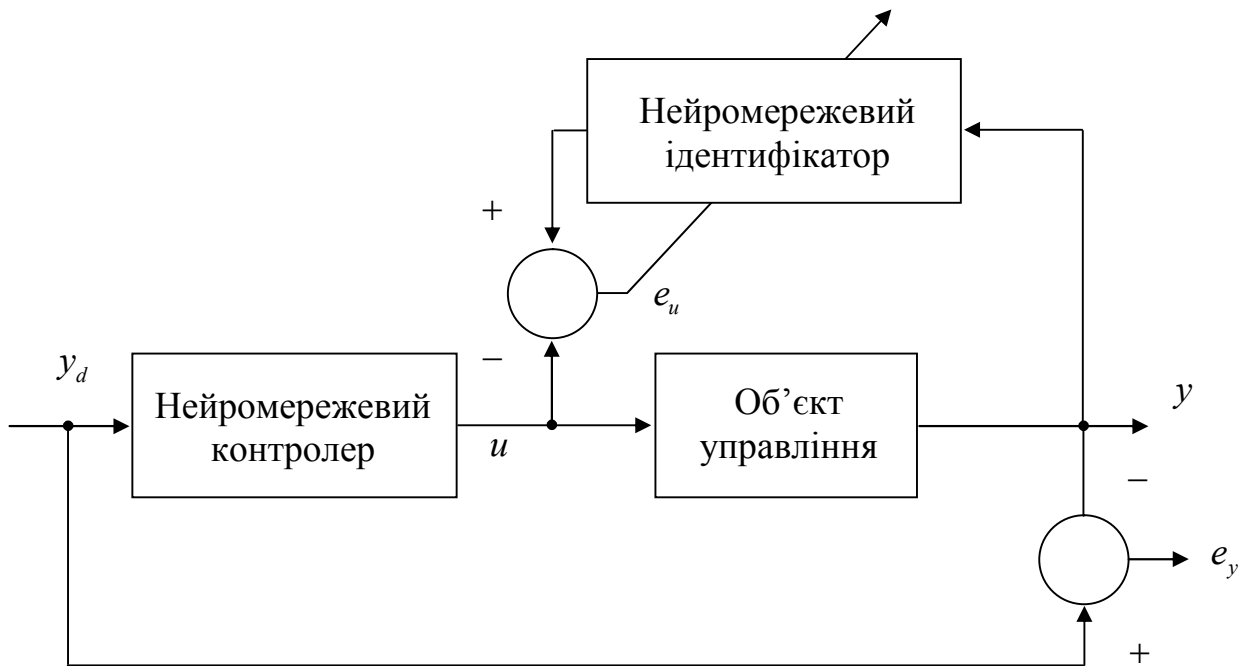


Рисунок 1.8 – Схема непрямого управління

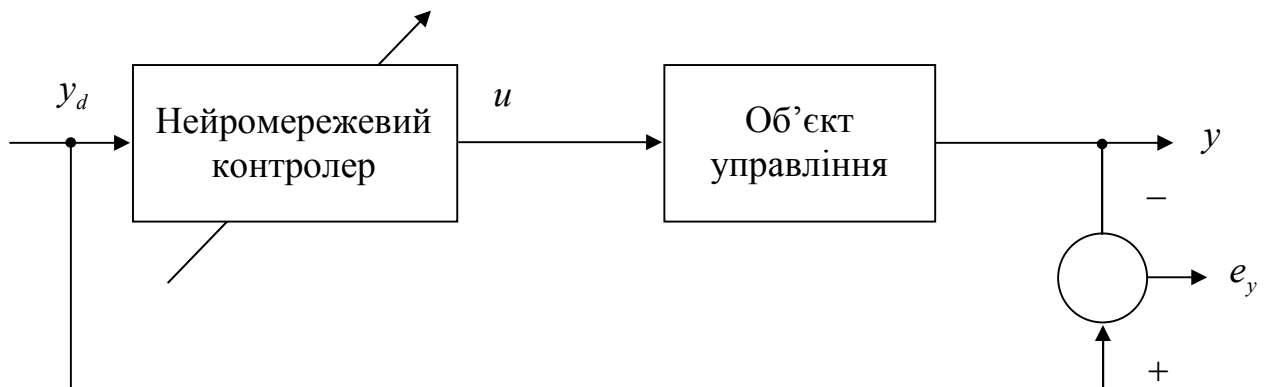


Рисунок 1.9 – Схема непрямого управління

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

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

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

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

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

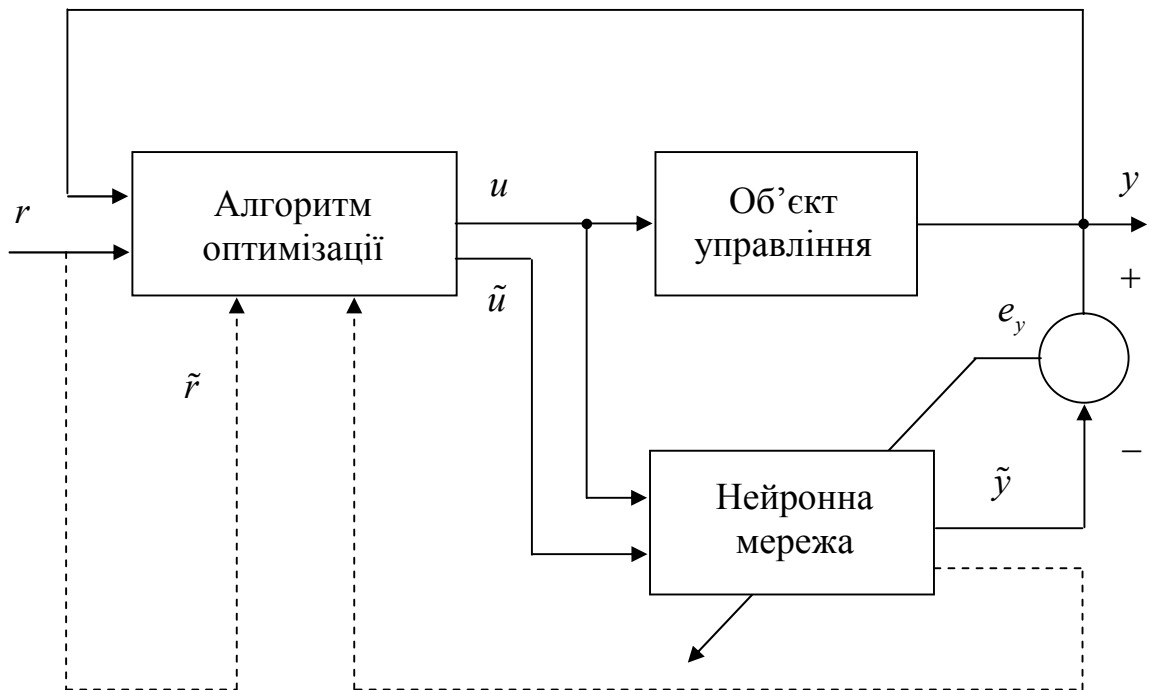


Рисунок 1.10 – Предиктивне регулювання

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

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

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

У даній магістерській роботі розглядається використання нелінійного предиктивного регулювання.

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

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

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

Таблиця 1.1 – Огляд характеристик методів управління

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

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

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

1.5 Призначення і основні характеристики зливкозовів

Зливковоз призначається для транспортування злитків від нагрівальних колодязів до приймального рольгангу. Злиток, нагрітий до $1200 \div 1250^{\circ}\text{C}$ виймають кліщовим краном з нагрівального колодязя і ставлять в люльку зливковоза. Після цього зливковоз переміщається до приймального рольгангу, на який і перекидається поданий злиток. Потім зливковоз знов повертається до нагрівальних колодязів за наступним злитком. Режим роботи зливковоза – повторно – короткочасний, при якому час роботи чергується з паузою, необхідною для завантаження зливковоза злитком і для укладання злитка на приймальний рольганг.

Зливковози, що існують в даний час, ділять на самохідні (тролейні) і канатні. Самохідний зливковоз є двовісним візком, що переміщається по рейках за допомогою одного або двох двигунів, одержуючих живлення через тролей. Люлька зливковоза, в яку укладається листок, має механізм повороту для плавного опускання злитка на приймальний рольганг.

Продуктивність зливкового заздості від швидкості його переміщення, величин прискорення і уповільнення, кількості злитків, що перевозяться за один раз. Швидкість пересування самохідних зливковозів складає $6 \div 7 \text{ м/с}$. Подальше збільшення швидкості недоцільне, бо шлях, прохідний при сталій (максимальною) швидкості, виходить вже відносно невеликим, а ефект від підвищення швидкості – незначним. Величини прискорення і уповільнення залежать від коефіцієнта зчеплення коліс з рейками. Якщо при русі зливкового сила інерції $T = G_0 a / g$ більше сил зчеплення $G_0 \varphi$ між ходовими колесами і рейками, то відбудеться ковзання коліс об рейки (буксирування). Тут: G_0 - сила тяжіння зливкового із злитком, H ; a - прискорення м/с^2 ; $g = 9,81 \text{ м/с}^2$; $\varphi = 0,18 \div 0,22$ - коефіцієнт зчеплення. За умови $T \leq G_0 \varphi$ прослизання буде відсутнє.

З рівності $T = G_0 a / g = G_0 \varphi$ можна визначити допустиме прискорення, зване критичним, вище за яке починається буксирування, тобто

$$a_{\text{ед}} = \varphi g$$

Таким чином, критичне прискорення зливкового

$$a_{\text{кр}} = (0,18 \div 0,22) 9,81 = 1,75 \div 2,16 \text{ м/с}^2$$

Зливковоз блюмінга, розрахований на перевезення одного злитка, може забезпечити продуктивність стану $2 \div 2,5$ млн. тонн в рік. При більшій продуктивності блюмінга ($3 \div 5$ млн. тонн в рік) застосовують двомісні зливковоз, за допомогою якого проводиться одночасне транспортування двох злитків.

1.6 Постановка задачі дослідження

Синтез і дослідження динамічних характеристик системи електроприводу механізму переміщення візка зливкового стана гарячої прокатки зазвичай виконується в припущенні нескінченної жорсткості всіх зв'язків між елементами механічної частини системи. Представлення механічної частини жорсткою приведеною ланкою відображає дійсний характер руху елементів тільки в середньому, оскільки не враховує вплив пружності реальних механічних зв'язків. Внаслідок кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, прикладення до якої управляючої дії (моменту двигуна), або збурюючої дії (моменту статичного навантаження) викликає коливання зв'язаних мас, які збільшують максимальні навантаження зв'язків і ускладнюють точність відробітку необхідних рухів і переміщень. Загальновживані регулятори не в змозі забезпечити задовільні показники якості функціонування системи.

Використання нейронних мереж у системах управління - це нова сфера, що поступово розвивається. Нейроуправління поєднує аспекти автоматичного керування, штучного інтелекту та робототехніки. Нейронні мережі мають унікальні характеристики, які роблять їх важливими для управлінських систем: вони вивчають дані, що представлені, і є потужним інструментом для імітації складних процесів і зв'язків. Також вони можуть адаптуватися до змін у властивостях об'єктів управління та оточуючого середовища, і володіють високою стійкістю до пошкоджень завдяки одночасній обробці інформації всіма нейронами. Використання штучних нейронних мереж дозволяє ефективно вирішувати завдання управління складними динамічними об'єктами, які традиційні методи не завжди можуть вирішити.

У науковій літературі, як у вітчизняній, так і за кордоном, ретельно досліджуються теоретичні аспекти і методи створення систем управління складними багатовимірними динамічними об'єктами на основі багат шарових нейронних мереж. У працях, присвячених цим мережам, велика увага приділяється основ-

ним характеристикам нелінійних багатошарових нейронних мереж та алгоритмам їх навчання в реальному часі, а також приводяться приклади комп'ютерного моделювання систем управління, заснованих на таких нейромережах, для складних нелінійних об'єктів з багатими взаємозв'язками.

В даний час отримано багато прикладів успішно працюючих нейромережових систем управління. У наукових джерелах представлено численні випадки використання нейронних мереж для керування різними об'єктами: літаком [59-61], вертольотом [62], гірничозбагачувальним процесом [63], автомобілем-роботом [64], гібридним двигуном автомобіля [65], пневмоциліндром [66], об'єктом спеціального призначення [67], моделлю перевернутого маятника [68], електропечю [69], турбогенератором [70], зварювальним апаратом [71] і інших. Ці приклади демонструють ефективність використання нейромережових методів для управління нелінійними динамічними системами в складних умовах експлуатації.

Таким чином, огляд останніх досягнень і публікацій щодо синтезу сучасних регуляторів систем управління на основі нейронних мереж свідчить про перспективність цього напрямку. Робота, спрямована на застосування нейронних мереж для управління системами із складними кінематичними зв'язками, є актуальною.

Метою роботи є синтез нейромережевої системи управління механізму переміщення візка зливковому кільцевій зливкоподачі стана гарячої прокатки, що має високі динамічні характеристики. При цьому об'єкт управління характеризується наявністю в ній пружних елементів, що призводить до необхідності розглядати його як багатомасову систему. Система повинна забезпечити високоякісне регулювання по задаючій і збурюючій діях при ступінчастому вхідному сигналі з випадковою амплітудою.

Для досягнення поставленої мети необхідно вирішити наступні задачі.

1. Виконати вибір системи електроприводу механізму переміщення візка зливковоза кільцевій зливкоподачі стана гарячої прокатки, вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу. Виконати

нати синтез системи підлеглого регулювання параметрами електроприводу, розглядаючи об'єкт управління як одномасову систему.

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

3. Синтезувати нейрорегулятор, що забезпечує високу швидкодію і якість управління двомасовою електромеханічною системою.

4. Виконати моделювання нейромережевої системи і провести аналіз отриманих результатів.

ВИСНОВКИ ДО РОЗДІЛУ 1

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

2. Дослідження показали ефективність застосування багат шарових мереж прямого розповсюдження для створення нейромережевих систем управління. Двошарова мережа з прямою передачею сигналу здатна досить точно апроксимувати будь-яку функцію з обмеженою кількістю точок розриву, якщо забезпечити достатню кількість нейронів у прихованому шарі. Обговорені питання навчання нейронних мереж та вибрано метод навчання для нейронної мережі регулятора.

3. Один із ключових етапів у розв'язанні задач управління, прогнозування поведінки та моделювання динамічних систем – це створення їх адекватних математичних моделей. Дослідження показують, що ідентифікацію нелінійних динамічних об'єктів ефективно можна здійснити за допомогою штучних нейронних мереж. Для ідентифікації найчастіше застосовуються багат шаровий перцептрон і радіально-базисні мережі.

4. На основі аналізу сучасних структур систем нейрорегулювання було обрано принцип нелінійного предиктивного регулювання для створення нейромережевого регулятора.

5. У результаті аналізу вимог до систем управління електроприводами основних і допоміжних механізмів прокатних станів, з урахуванням пружності механічних зв'язків, показана ефективність використання нейромережевих технологій для синтезу системи управління механізмом переміщення візка зливкової кільцевої зливкоподачі на стані гарячої прокатки.

6. Проведено аналіз нейромережевих підходів до розв'язання задач управління. Сформульовано задачу дослідження, що полягає в розробці нейромережевої системи управління механізмом переміщення візка зливкової кільцевої зливкоподачі на стані гарячої прокатки, з урахуванням податливості механічних зв'язків.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПЕРЕМІЩЕННЯ ВІЗКА ЗЛИВКОВОЗА СТАНА ГАРЯЧОЇ ПРОКАТКИ

2.1 Основні технічні характеристики зливковоза. Вимоги до системи управління

Вихідні дані для розрахунку електроприводу механізму переміщення візка зливковоза кільцевої зливкоподачі і синтезу системи управління наведені в табл. 2.1.

Таблиця 2.1 – Технічні дані зливковоза

Найменування величини	Розмірність	Значення
1	2	3
Маса зливковоза, $m_{\text{в}}$	кг	$38 \cdot 10^3$
Маса зливка, $m_{\text{зл}}$	кг	$2 \cdot 12,5 \cdot 10^3$
Діаметр ходових коліс	м	1,1
Середній діаметр цапф підшипників коліс, $D_{\text{ц}}$	м	0,2
ККД редуктора, η	$\eta = 0,92$	0,92
Момент інерції частин зливковоза, що обертаються, $J_{\text{об}}$	кг · м ²	970
Шлях руху зливковоза, S	м	280
Відстань від найбільш видаленого колодязя до рольгангу, $S_{\text{пр}}$	м	120м
Швидкість руху на прямолінійних ділянках шляху (робоча швидкість), $V_{\text{роб}}$	м/с	4
Максимальна швидкість на холостій гілці шляху, $V_{\text{max}} = V_{\text{роб}}$	м/с	4
Швидкість першого ступеня при гальмуванні, V_1	м/с	3,3
Швидкість на закругленні перед зіштовхувачем, V_3	м/с	2,5

Швидкість другого ступеня при гальмуванні, V_2	$м/с$	1,5
Швидкість на поворотах і перед зіштовхуванням, V_{π}	$м/с$	1,5
Мінімальна швидкість на холостій гілці шляху, $V_{\min}$	$м/с$	0,7
Швидкість підходу до зіштовхувала, V_c	$м/с$	0,7
Шляхи прохідні зливковозом після завантаження на ділянці 14 (див. діаграму швидкості) з швидкостями:		
V_1	$м$	4
V_3	$м$	4,5
V_{π}	$м$	6
V_c	$м$	1
Час стоянки у колодязів, $t_{п.1}$	$с$	7
Час стоянки у рольганга, $t_{п.2}$	$с$	7

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 10 с.

2.2 Вибір системи електроприводу зливковоза

У зв'язку з широкою механізацією і автоматизацією технологічних процесів прокатного виробництва високі вимоги пред'являють до різного роду допоміжним механізмам прокатних станів і їх електроприводам.

Для класифікації допоміжних електроприводів необхідно розглянути основні види допоміжних механізмів і особливості їх роботи.

Кінематика і конструкція допоміжних механізмів вельми різноманітні. Слід розрізняти два основні класи допоміжних механізмів:

1. Оброблювальні механізми, робота яких пов'язана із зміною форми прокатуваного металу - механізми різкі, правки, змотування, гнучкі, обробки і обробки поверхні, обв'язування і упаковки.

2. Транспортні механізми, робота яких пов'язана з переміщенням металу - подачею до валянь і транспортуванням прокатуваного металу, переміщенням, кантівкою.

По вигляду регулятора основного технологічного параметра і іншим вимогам електропривод допоміжних механізмів можна розділити на декілька груп. Електроприводи правильних машин, дискових ножиць і інших аналогічних механізмів, електроприводи намотувальних-розмотувальних пристроїв (моталок і розмотувачів), електроприводи побічних роликів, електроприводи летючих ножиць, електроприводи натискних пристроїв, лінійок маніпуляторів і інших механізмів, що вимагають позиційного регулювання, електроприводи робочих і гуркотах рольгангов, кантують і маніпуляторів, слитковозов, нажимних пристроїв, обтискових, сортових і листових станів гарячого плющення.

Не дивлячись на велику різноманітність механізмів прокатних станів, системи регулювання електроприводів цих механізмів можуть бути істотно уніфіковані незалежно від конкретного застосування. Основні структури системи регулювання наступні:

1 Пропорційний або інтегро-пропорційний регулятор частоти обертання (або ЕРС) з підлеглим регулятором струму якоря.

2 Пропорційний або інтегро-пропорційний регулятор частоти з підлеглим регулятором струму якоря і залежним управлінням потоком збудження двигуна.

3 Регулятори для дводвигунних приводів при живленні кожного якоря від окремого перетворювача.

4 Регулятори структур 1 і 2 для двохдвигунних приводів при паралельному включенні якоря і з вирівнюванням навантажень дією на полі двигуна.

5 Регулятор положення з підлеглими регуляторами швидкості і струму.

6 Регулятори частоти обертання (або ЕРС) з підлеглими регуляторами струму якоря для двох технологічно зв'язаних механізмів, з регулятором синхронізації переміщення виконавських органів цих механізмів.

7 Регулятор потужності для намотувальних механізмів.

8 Регулятор струму, уставка якого регулюється відповідно до заданого натягнення, із зовнішнім контуром швидкості, що насичається, як обмежувачем.

Всі необхідні модифікації структур систем регулювання можуть бути реалізовані на основі цих восьми структур шляхом додавання деяких допоміжних пристроїв (завдання частоти обертання, пом'якшення механічних характеристик і т.д.).

Зливковоз відноситься до допоміжних механізмів реверсивних обтискових прокатних станів, безпосередньо обслуговуючих плющення металу в кліті. До електроприводу зливковоза пред'являється основна вимога – забезпечення максимальної продуктивності всього агрегату.

Електроприводи зливковозів працюють в повторно-короткочасному режимі з великою частотою включень. Велика частота включень пред'являє високі вимоги до динаміки електроприводу при розгоні і гальмуванні, оскільки час роботи допоміжних механізмів – істотний чинник, що визначає роботу стану.

Електропривод повинен мати достатньо широкий діапазон регулювання. Швидкості – не меншого 10:1. Задовольнити всім перерахованим вимогам може електропривод постійного струму при живленні двигуна від перетворювача тиристора – система ТП-Д.

2.3 Розрахунок і побудова швидкісної діаграми електроприводу зливковоза

Швидкісну діаграму розраховуємо для перевезення злитків від найбільш видаленого нагрівального колодязя до рольганга.

Допустиме лінійне прискорення або уповільнення приймається з умови відсутності тієї, що пробуксувала (прослизання) коліс об рейки

$$a_1 < a_{\text{кр}} = \mu_6 \cdot g,$$

де $g = 9,81 \text{ м/с}^2$

$\mu_6 = 0,15 \div 0,2$ – коефіцієнт тертя холодного металу об метал, приймаємо $\mu_6 = 0,2$.

$$a_{\text{кр}} = 0,2 \cdot 9,81 = 1,72 \text{ м/с}^2.$$

Приймаємо $a = 0,6 \text{ м/с}^2$.

Розраховуємо час і шлях, прохідний зливковозом на різних ділянках руху.

Ділянка 1 – період нормального прискорення зливковоза із злитком

$$t_1 = \frac{V_{\text{роб}}}{a},$$

$$t_1 = \frac{4}{0,6} = 6,7 \text{ с},$$

$$S_1 = \frac{V_{\text{роб}}}{2} \cdot t_1,$$

$$S_1 = \frac{4}{2} \cdot 6,7 = 13,4 \text{ м}.$$

Ділянка 3 – гальмування від швидкості $V_{\text{роб}}$ до V_1

$$t_3 = \frac{V_{\text{роб}} - V_1}{a},$$

$$t_3 = \frac{4 - 3,2}{0,6} = 1,34 \text{ с},$$

$$S_3 = \frac{V_{\text{роб}} + V_1}{2} \cdot t_3,$$

$$S_3 = \frac{4 + 3,2}{2} \cdot 1,34 = 5,26 \text{ м}.$$

Ділянка 5 – гальмування від швидкості V_1 до V_3

$$t_5 = \frac{V_1 - V_3}{a},$$

$$t_5 = \frac{3,2 - 2,5}{0,6} = 1,29 \text{ с},$$

$$S_5 = \frac{V_1 + V_3}{2} \cdot t_5,$$

$$S_5 = \frac{3,2 + 2,5}{2} \cdot 1,29 = 3,8 \text{ м}.$$

Ділянка 7 – гальмування від швидкості V_3 до $V_{\text{п}}$

$$t_7 = \frac{V_3 - V_{\text{п}}}{a},$$

$$t_7 = \frac{2,5 - 1,5}{0,6} = 1,67 \text{ с},$$

$$S_7 = \frac{V_3 + V_{\text{п}}}{2} \cdot t_7,$$

$$S_7 = \frac{2,5 + 1,5}{2} \cdot 1,67 = 3,92 \text{ м}.$$

Ділянка 9 – гальмування від швидкості V_n до V_c

$$t_9 = \frac{V_n - V_c}{a},$$

$$t_9 = \frac{1,5 - 0,7}{0,6} = 1,47 \text{ с},$$

$$S_9 = \frac{V_n + V_c}{2} \cdot t_9,$$

$$S_9 = \frac{1,5 + 0,7}{2} \cdot 1,47 = 1,69 \text{ м}.$$

Ділянка 11 – період зупинки зливковогоза перед зіштовхувачем

$$t_{11} = \frac{V_c}{a},$$

$$t_{11} = \frac{0,7}{0,6} = 1,16 \text{ с},$$

$$S_{11} = \frac{V_c}{2} \cdot t_{11},$$

$$S_{11} = \frac{0,7}{2} \cdot 1,16 = 0,43 \text{ м}.$$

Час руху на ділянках 4, 6, 8 і 10

$$t_4 = \frac{S_4}{V_1},$$

$$t_4 = \frac{4}{3} = 1,25 \text{ с},$$

$$t_6 = \frac{S_6}{V_3},$$

$$t_6 = \frac{4,5}{2,5} = 1,8 \text{ с}.$$

$$t_8 = \frac{S_8}{V_{\pi}},$$

$$t_8 = \frac{6}{1,5} = 4 \text{ с},$$

$$t_{10} = \frac{S_{10}}{V_c},$$

$$t_{10} = \frac{1}{0,7} = 1,43 \text{ с}.$$

Ділянки 13 і 17 – рух з швидкістю V_{π} на поворотах

$$t_{13} = t_{17} = \frac{S_{\pi}}{V_{\pi}},$$

$$t_{13} = t_{17} = \frac{20}{1,5} = 13,3 \text{ с}.$$

Ділянки 14 і 16 – розгін від швидкості V_{π} до $V_{\max} = V_{\text{роб}}$ і гальмування від $V_{\text{роб}}$ до V_{π} відповідно

$$t_{14} = t_{16} = \frac{V_{\text{роб}} - V_{\pi}}{a},$$

$$t_{14} = t_{16} = \frac{4 - 1,5}{0,6} = 4,17 \text{ с},$$

$$S_{14} = S_{16} = \frac{V_{\text{роб}} + V_{\text{п}}}{2} \cdot t_{13},$$

$$S_{14} = S_{16} = \frac{4 + 1,5}{2} \cdot 4,74 = 15,88 \text{ м}.$$

Ділянка 2 – рух зливковоза із злитком із сталою швидкістю $V_{\text{роб}}$

$$S_2 = S_{\text{пр}} - (S_1 + S_2 + S_3 + S_4 + S_5 + S_6 + S_7 + S_8 + S_9 + S_{10} + S_{11} + S_{12} + S_{18}) =$$

$$= 120 - (13,4 + 5,26 + 3,8 + 4 + 4,5 + 3,92 + 6 + 1,69 + 1 + 0,43 + 2,25 + 2,25) = 63,9 \text{ м}.$$

$$t_2 = \frac{S_2}{V_{\text{роб}}},$$

$$t_2 = \frac{63,9}{4} = 16,76 \text{ с}.$$

Ділянка 15 – рух зливковоза без злитка із сталою швидкістю $V_{\text{роб}}$

$$S_{15} = S_{\text{пр}} - (S_{13} + S_{15}),$$

$$S_{15} = 120 - 2 \cdot 16,25 = 87,5 \text{ м},$$

$$t_{15} = \frac{S_{14}}{V_{\text{роб}}},$$

$$t_{15} = \frac{87,5}{4} = 21,9 \text{ с}.$$

Ділянки 12 і 18 – розгін до швидкості $V_{\text{п}}$ після зіштовхувача і гальмувань від швидкості $V_{\text{п}}$ до нуля перед нагрівальним колодязем

$$t_{12} = t_{18} = \frac{V_{\text{п}}}{a},$$

$$t_{12} = t_{18} = \frac{1,5}{0,6} = 2,7 \text{ с},$$

$$S_{12} = S_{18} = \frac{V_{\Pi}}{2} \cdot t_{12},$$

$$S_{12} = S_{18} = \frac{1,5}{2} \cdot 2,7 = 2,1 \text{ м}.$$

Сумарний час роботи зливковогоза

$$t_{\text{роб}} = \sum_{i=1}^{18} t_i,$$

$$t_{\text{роб}} = 6,7 + 16,76 + 1,34 + 1,25 + 1,29 + 1,8 + 1,67 + 4 + 1,47 + 1,43 + \\ + 1,16 + 3 + 13,3 + 4,74 + 22 + 5 + 13,3 + 3 = 104,2 \text{ с}.$$

Час циклу

$$T_{\text{ц}} = t_{\text{роб}} + t_{\text{п}},$$

$$T_{\text{ц}} = 104,2 + 2 \cdot 7 = 118,2 \text{ с}.$$

Еквівалентна тривалість включення

$$PB_{\text{е}} = \frac{t_{\text{роб}}}{T_{\text{ц}}},$$

$$PB_{\text{е}} = \frac{104,2}{118,2} \cdot 100\% = 86,2\%.$$

Швидкісна діаграма показана на рис.2.1,а.

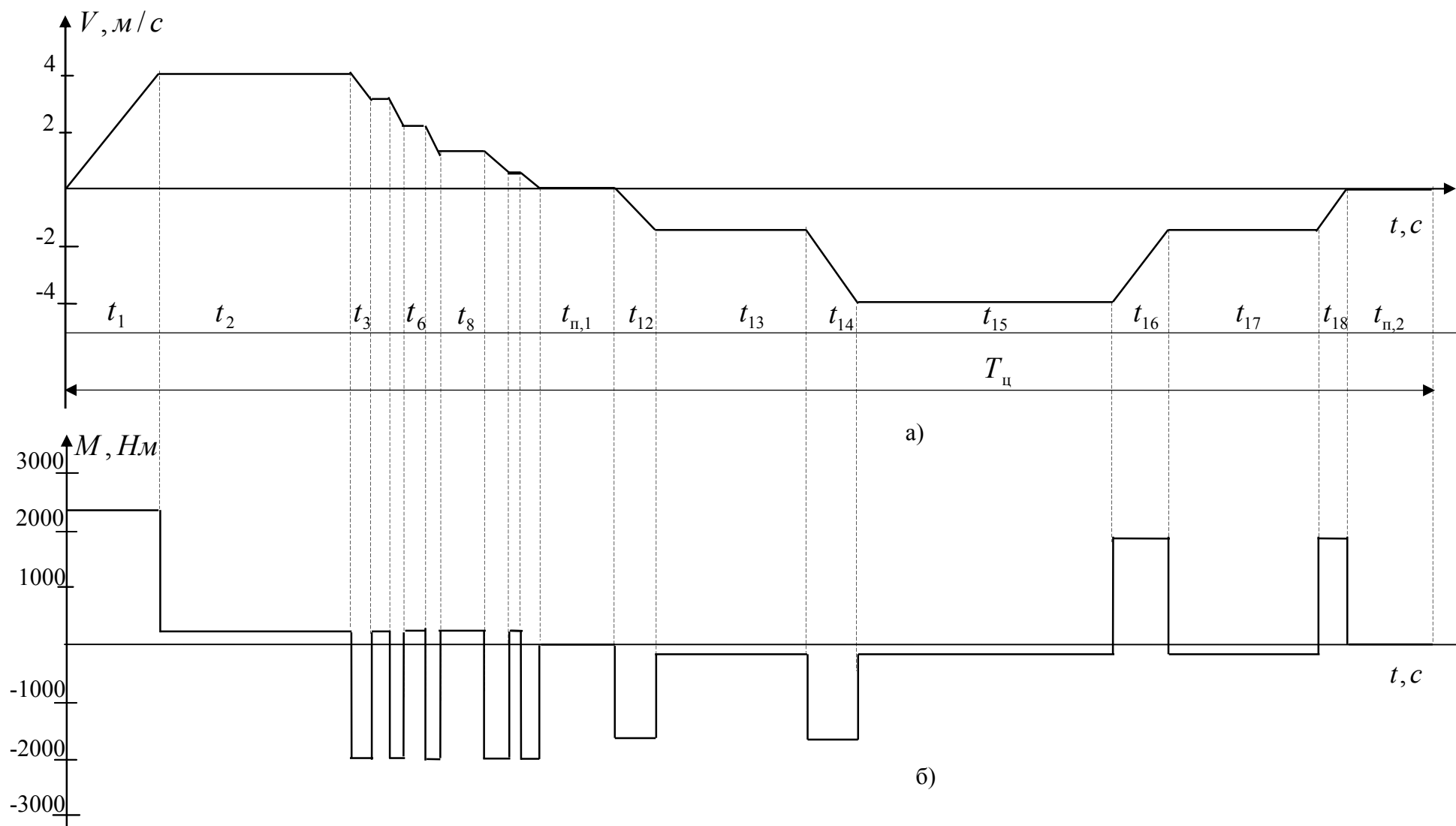


Рисунок 2.1 – Швидкісна а) і навантажувальна б) діаграми роботи електроприводу зливковоза

2.4 Вибір електродвигуна по потужності

Використовуючи дані, отримані при розрахунку швидкісної діаграми, виконуємо вибір електродвигуна по потужності. Визначаємо перевантажувальну здатність двигуна, обчисливши орієнтовну величину пускового моменту по заданих величинах граничних прискорень і уповільнень.

Статичний момент при русі зливковогоз із злитком

$$M_c = \frac{m_{\text{зл}} + m_{\text{в}}}{\eta} \left(\mu \frac{D_{\text{ц}}}{2} + \varphi \right) \beta g,$$

де η – ККД редуктора;

$\mu = 0,008 \div 0,01$ – коефіцієнт тертя в підшипниках кочення;

$\varphi = (0,5 \div 0,8) \cdot 10^{-3}$ – коефіцієнт тертя кочення ходового колеса;

$\beta = 1,5 \div 2,0$ – коефіцієнт, що враховує тертя реборд об рейки

$$M_c = \frac{10^3 \cdot (2 \cdot 12,5 + 38)}{0,92} \cdot \left(0,009 \cdot \frac{0,2}{2} + 0,5 \cdot 10^{-3} \right) \cdot 1,5 \cdot 9,8 = 1138,7 \text{ Нм}.$$

Динамічний момент, необхідний для подолання сил інерції поступально рухомих мас

$$M_{\text{дин.1}} = \frac{(m_{\text{зл}} + m_{\text{в}}) \cdot a \cdot D_{\text{к}}}{2\eta},$$

$$M_{\text{дин.1}} = \frac{10^3 (2 \cdot 12,5 + 38) \cdot 0,6 \cdot 1,1}{2 \cdot 0,92} = 13821 \text{ Нм}.$$

Динамічний момент, необхідний для подолання сил інерції частин зливковогоз, що обертаються (без урахування моменту інерції двигуна)

$$M_{\text{дин.2}} = \frac{2J_{\text{об}}a}{D_{\text{к}}},$$

$$M_{\text{дин.2}} = \frac{2 \cdot 970 \cdot 0,5}{1,1} = 1104 \text{ Нм}.$$

Потужність попередньо вибраного двигуна

$$P_{\text{дв}} \geq \frac{M_c + M_{\text{дин.1}} + M_{\text{дин.2}}}{\lambda} \cdot \frac{2V_{\text{пр}}}{D_{\text{к}}},$$

де λ – допустимий коефіцієнт перевантаження двигуна; для двигунів постійного струму $\lambda = 2 \div 2,5$.

$$P_{\text{дв}} \geq \frac{11387 + 13821 + 1104}{2} \cdot \frac{2 \cdot 4}{1,1} = 75218 \text{ Вт}.$$

По каталогу вибираємо двигун Д814 з наступними технічними даними: Номінальна потужність $P_{\text{н}} = 83 \text{ кВт}$. Номінальна напруга $U_{\text{н}} = 220 \text{ В}$. Номінальна частота обертання $n_{\text{н}} = 470 \text{ об/хв}$. Номінальний струм $I_{\text{н}} = 415 \text{ А}$. Напруга незалежної обмотки збудження $U_{\text{зн}} = 220 \text{ В}$. Номінальний струм збудження $I_{\text{зн}} = 10,2 \text{ А}$. Число активних провідників обмотки якоря $N = 504$. Число паралельних гілок якоря $2a = 4$. Число полюсів $2p = 4$. Магнітний потік полюса $\Phi_{\text{н}} = 111 \cdot 10^{-3} \text{ Вб}$. Опір якоря і додаткових полюсів $R_{\text{я}} + R_{\text{дп}} = 0,0066 \text{ Ом}$. Момент інерції якоря $J_{\text{дв}} = 16,25 \text{ кг} \cdot \text{м}^2$. Тривалість включення $ПВ = 40\%$

2.5 Побудова діаграми навантаження електроприводу

Виконаємо розрахунок навантажувальної діаграми і перевірку двигуна по нагріву і перевантажувальній здатності.

Статичний момент при пересуванні зливковоза із злитками із сталою швидкістю, приведений до валу двигуна

$$M_c^{np} = \frac{M_c}{i},$$

де i – передавальне число редуктора

$$i = \frac{\omega_{дв}}{\omega_k} = \frac{\omega_n}{\omega_k},$$

ω_k – кутова швидкість коліс зливковоза

$$\omega_k = \frac{2V_{роб}}{D_k},$$

$$\omega_k = \frac{2 \cdot 4}{1,1} = 7,27 \text{ c}^{-1}.$$

ω_n – номінальна кутова швидкість обертання двигуна

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

$$\omega_n = \frac{3,14 \cdot 470}{30} = 49,2 \text{ c}^{-1},$$

$$i = \frac{49,2}{7,27} = 7,78,$$

$$M_c^{np} = \frac{1098}{6,3} = 174 \text{ Нм}.$$

Статичний момент при русі порожнього зливковоза із сталою швидкістю,

приведений до валу двигуна

$$M_{с.хх}^{пр} = \frac{m_{в}}{\eta \cdot i} \left(\mu \frac{D_{ц}}{2} + \varphi \right) \beta g ,$$

$$M_{с.хх}^{пр} = \frac{38 \cdot 10^3}{7,78 \cdot 0,92} \cdot \left(0,01 \frac{0,2}{2} + 0,7 \cdot 10^3 \right) \cdot 1,6 \cdot 9,8 = 92,7 \text{ Нм} .$$

Динамічний момент, необхідний для подолання сил інерції поступально рухомих мас при русі зливковоза із злитками, приведений до валу двигуна

$$M_{дин.1}^{пр} = \frac{M_{дин.1}}{i} ,$$

$$M_{дин.1}^{пр} = \frac{1138,7}{7,76} = 156,4 \text{ Нм} .$$

Динамічний момент, необхідний для подолання сил інерції поступально рухомих мас при русі зливковоза без зливків, приведений до валу двигуна

$$M_{дин.1.хх}^{пр} = \frac{m_{в} a D_{к}}{2 \eta i} ,$$

$$M_{дин.1.хх}^{пр} = \frac{38 \cdot 10^3 \cdot 0,5 \cdot 1,1}{2 \cdot 0,92 \cdot 7,78} = 1138 \text{ Нм} .$$

Динамічний момент, необхідний для подолання сил інерції частин зливковоза, що обертаються, приведений до валу двигуна з урахуванням моменту інерції двигуна

$$M_{дин.2}^{пр} = \frac{2 J_{\Sigma об} a}{D_{к} i} ,$$

де

$$J_{\Sigma об} = J_{об} + J_{дв} \cdot i^2 ,$$

$$J_{\Sigma \text{об}} = 970 + 16,25 \cdot 7,78^2 = 1649 \text{ кг} \cdot \text{м}^2,$$

$$M_{\text{дин.2}}^{\text{пр}} = \frac{2 \cdot 1649 \cdot 0,6}{0,9 \cdot 7,78} = 271 \text{ Нм}.$$

Результуючий момент на валу двигуна для різних ділянок руху визначається таким чином.

Ділянки 2,4,6,8,10

$$M_I = M_c^{\text{пр}} = 146,3 \text{ Нм}.$$

Ділянка 1

$$M_{II} = M_c^{\text{пр}} + M_{\text{дин.1}}^{\text{пр}} + M_{\text{дин.2}}^{\text{пр}},$$

$$M_{II} = M_I = 146,3 + 1564 + 271 = 1981 \text{ Нм}.$$

Ділянки 3,5,7,9,11

$$M_{III} = M_c^{\text{пр}} - M_{\text{дин.1}}^{\text{пр}} - M_{\text{дин.2}}^{\text{пр}},$$

$$M_{III} = 146,3 - 1564 - 271 = -1687 \text{ Нм}$$

Ділянки 13,15,17

$$M_{IV} = M_{\text{с.хх}}^{\text{пр}} = 92,7 \text{ Нм}.$$

Ділянки 12,14

$$M_V = M_{\text{с.хх}}^{\text{пр}} + M_{\text{дин.1.хх}}^{\text{пр}} + M_{\text{дин.2}}^{\text{пр}},$$

$$M_V = 92,7 + 1138 + 271 = 1372 \text{ Нм}.$$

Ділянки 16,18

$$M_{VI} = M_{\text{с.хх}}^{\text{пр}} - M_{\text{дин.1.хх}}^{\text{пр}} - M_{\text{дин.2}}^{\text{пр}},$$

$$M_{VI} = 92,7 - 1138 - 271 = -1684 \text{ Нм}.$$

Діаграма навантаження, побудована на підставі вищенаведених розрахунків, показана на рис.2.1 б

2.6 Перевірка двигуна по нагріву і перевантажувальній здатності

Еквівалентний середньоквадратичний момент електроприводу зливковогоза визначається по формулі:

$$M_e = \sqrt{\frac{\sum_{i=1}^{18} M_i^2 t_i}{\sum_{i=1}^{18} t_i}}. \quad (2.1)$$

Після підстановки числових значень, отриманих при побудові швидкісної діаграми і діаграми навантаження в (2.1), отримуємо $M_e = 653 \text{ Нм}$

Номінальний момент двигуна

$$M_n = \frac{P_n}{\omega_n},$$

$$M_n = \frac{83 \cdot 10^3}{49,2} = 1687 \text{ Нм}.$$

Номінальний момент двигуна повинен задовольняти умові

$$M_n \geq M_e^{\text{пр}},$$

де $M_e^{\text{пр}}$ – еквівалентний приведений момент

$$M_e^{\text{пр}} = M_e \sqrt{\frac{PB_e}{PB_{\text{ст}}}},$$

$$M_e^{\text{пр}} = 683 \cdot \sqrt{\frac{86,2}{40}} = 1001 \text{ Нм}.$$

Оскільки умова $M_n \geq M_e^{\text{пр}}$ виконується, то двигун за умовами нагріву відповідає вимогам.

Перевіряємо двигун по перевантажувальній здатності. Для цього повинна виконуватись умова:

$$M_{\max} \leq M_{\text{дв max}},$$

де $M_{\max} = M_{II} = 2494 \text{ Нм}$ – максимальний момент електроприводу, визначений в процесі розрахунку діаграми навантаження;

$M_{\text{дв max}}$ – максимальний момент, що розвивається двигуном

$$M_{\text{дв max}} = \lambda \cdot M_{II},$$

λ – перевантажувальна здатність двигуна; $\lambda = 2 \div 2,5$, отже

$$M_{\text{дв max}} = 2 \cdot 1687 = 3374 \text{ Нм}.$$

Оскільки виконується співвідношення $M_{\max} \leq M_{\text{дв max}}$, то по перевантажувальній здатності двигун теж задовольняє вимогам.

2.7 Вибір основного електроустаткування

Згідно рекомендаціям [72] вибираємо по каталогу комплектний тиристорний перетворювач типу КТЕУ 1000/220–13222, що має наступні технічні характеристики.

Номинальна випрямлена напруга $U_{\text{дн}} = 230 \text{ В}$;

Номинальний випрямлений струм $I_{\text{дн}} = 1000 \text{ А}$.

По каталогу вибираємо трансформатор типа ТСП – 400 / 0,7 У4. Характеристики трансформатора наступні.

Номинальна напруга мережевої обмотки: $U_{1л} = 6 \text{ кВ}$

Номинальна фазна напруга вентильної обмотки $U_{2ф} = 115 \text{ В}$

Номинальна випрямлена напруга перетворювача $U_{\text{дн}} = 230 \text{ В}$.

Номинальна сила випрямленого струму $I_{\text{дн}} = 1250 \text{ А}$.

Втрати холостого ходу $P_{\text{хх}} = 1500 \text{ Вт}$.

Втрати короткого замикання $P_{\text{кз}} = 5545 \text{ Вт}$.

Напруга короткого замикання $U_{\text{кз}} = 6,4\%$

Згладжуючий реактор виберемо виходячи з величини номінального струму перетворювача.

Тип СРОСЗ – 800 МУХЛЧ. Номінальний струм реактора $I_{\text{рн}} = 1250 \text{ А}$; індуктивність $L_{\text{р}} = 0,8 \cdot 10^{-3} \text{ Гн}$; активний опір $R_{\text{р}} = 0,002 \text{ Ом}$.

2.8 Розрахунок параметрів силового кола електроприводу

Розраховуємо параметри силового ланцюга електроприводу, які використовуються для розрахунку системи регулювання електроприводом.

Проти-ЕРС, що наводиться в якірному колі двигуна

$$E_{\text{д}} = U_{\text{н}} - I_{\text{н}} \cdot R_{\text{яц}},$$

$$E_{\text{д}} = 220 - 415 \cdot 0,009 = 216 \text{ В}.$$

Постійна проти-ЕРС двигуна

$$c = k \cdot \Phi_{\text{н}} = \frac{E_{\text{д}}}{\omega_{\text{н}}},$$

де k – конструктивна постійна двигуна;

$\Phi_{\text{н}}$ – номінальний магнітний потік двигуна

$$c = k \cdot \Phi_{\text{н}} = \frac{216}{49,2} = 4,3.$$

Сумарний опір якірного ланцюга електродвигуна

$$R_{\text{як}} = \alpha_t \cdot (R_{\text{я}} + R_{\text{дп}} + R_{\text{ко}}),$$

де $R_{\text{як}} = \alpha_t \cdot (R_{\text{я}} + R_{\text{дп}} + R_{\text{ко}})$ – активний опір обмотки якоря, додаткових полюсів і компенсаційної обмотки відповідно, згідно паспортних даних двигуна $R_{\text{я}} + R_{\text{дп}} = 0,023 \text{ Ом}$; $R_{\text{ко}} = 0$;

α_t – температурний коефіцієнт; для класу ізоляції Н $\alpha_t = 1,36$.

$$R_{\text{як}} = 1,36 \cdot 0,0066 = 0,009 \text{ Ом}.$$

Еквівалентний опір тиристорного перетворювача

$$R_{\text{екв}} = 0,5 \cdot U_{\text{к}} \cdot \frac{E_{\text{д0}}}{I_{\text{дн}}}$$

де $I_{\text{дн}}$ – сила номінального випрямленого струму;

$U_{\text{к}}$ – напруга короткого замикання трансформатора;

$E_{\text{д0}}$ – ЕРС холостого ходу перетворювача, для трьохфазної мостової схеми випрямлення

$$E_{\text{д0}} = 2,34 \cdot U_{2\phi},$$

де $U_{2\phi}$ – фазна напруга вторинної обмотки трансформатора

$$E_{\text{д0}} = 2,34 \cdot 115 = 269 \text{ В},$$

$$R_{\text{екв}} = 0,5 \cdot 0,064 \cdot \frac{269}{415} = 0,0177 \text{ Ом}.$$

Активний опір трансформатора

$$R_{\text{т}} = (0,1 \div 0,2) \cdot U_{\text{к}} \cdot \frac{\pi}{m} \cdot \frac{E_{\text{д0}}}{I_{\text{дн}}},$$

$$R_{\text{т}} = R_{\text{екв}} = 0,2 \cdot \frac{269}{415} = 0,058 \text{ Ом}.$$

Активний опір силового кола

$$R_{\Sigma} = R_{\text{як}} + R_{\text{т}} + R_{\text{екв}} + R_{\text{р}},$$

$R_{\text{р}}$ – опір згладжуючого реактора; $R_{\text{р}} = 0,002 \text{ Ом}$

$$R_{\Sigma} = 0,009 + 0,0177 + 0,058 + 0,002 = 0,093 \text{ Ом}.$$

Індуктивність якірного ланцюга електродвигуна

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

де γ – коефіцієнт, рівний $\gamma = 0,6$ для некомпенсованих машин;

p – число пар головних полюсів

$$L_{\text{як}} = 0,6 \cdot \frac{220}{415} \cdot \frac{1}{49,2 \cdot 2} = 3,2 \cdot 10^{-3} \text{ Гн}.$$

Індуктивність розсіювання трансформатора

$$L_{\text{т}} = U_{\text{к}} \cdot \frac{1}{2mf} \cdot \frac{E_{\text{до}}}{I_{\text{дн}}},$$

де f – частота живлячої мережі;

m – фазність пульсації випрямленої напруги перетворювача

$$L_{\text{т}} = 0,064 \cdot \frac{1}{2 \cdot 6 \cdot 50} \cdot \frac{269}{1000} = 0,02 \cdot 10^{-3} \text{ Гн}.$$

Індуктивність силового кола

$$L_{\Sigma} = L_{\text{я}} + L_{\text{т}} + L_{\text{р}},$$

де $L_{\text{р}}$ – індуктивність згладжуючого реактора $L_{\text{р}} = 0,6 \cdot 10^{-3} \text{ Гн}$

$$L_{я\Sigma} = 3,2 \cdot 10^{-3} + 0,02 \cdot 10^{-3} + 0,8 \cdot 10^{-3} = 4,02 \cdot 10^{-3} \text{ Гн}.$$

Електромагнітна постійна часу електроприводу

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

$$T_e = \frac{4,02 \cdot 10^{-3}}{0,093} = 0,043 \text{ с}.$$

Електромагнітна постійна якірного ланцюга електродвигуна

$$T_a = \frac{L_{як}}{R_{як}},$$

$$T_a = \frac{0,32 \cdot 10^{-3}}{0,009} = 0,035 \text{ с}.$$

Електромеханічна постійна часу електроприводу

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

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

$$J_{\Sigma} = J_d + \frac{(m_{зл} + m_b) \cdot D_k^2}{2i^2} + \frac{J_{об}}{i^2},$$

J_d – момент інерції двигуна;

$J_{пр}$ – приведений момент інерції поступально і обертальний рухомих мас зливковоза при русі зливковоза із злитками

$$J_{пр} = \frac{(m_{зл} + m_b) \cdot D_k^2}{2i^2} + \frac{J_{об}}{i^2},$$

$$J_{\Sigma} = 16,25 + \frac{970}{7,78^2} + \frac{(2 \cdot 12,5 + 38) \cdot 1,1 \cdot 10^3}{7,78^2} = 16,25 + 16,02 + 546 = 578 \text{ кг} \cdot \text{м}^2.$$

$$T_{\text{м}} = 578 \cdot \frac{0,093}{4,3^2} = 1,2 \text{ с}$$

Коефіцієнт посилення тиристорного перетворювача

$$k_{\text{тп}} = \frac{E_{\text{д0}}}{U_{\text{у max}}}, \quad \dots$$

де $U_{\text{у max}}$ – амплітудне значення напруги порівняння, приймаємо $U_{\text{у max}} = 14 \text{ В}$.

Отже:

$$k_{\text{тп}} = \frac{269}{14} = 20.$$

Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c},$$

$$k_{\text{д}} = \frac{1}{4,3} = 0,233.$$

2.9 Вибір структури системи автоматичного регулювання координат електроприводу

Система автоматичного управління повинна забезпечити необхідні механічні характеристики електроприводу як в статичному, так і в динамічному режимах, високу якість перехідних процесів і оптимізацію роботи електроприводу за заданими параметрами у різних умовах роботи механізму. Досягнення високих показників регулювання можливе тільки при створенні замкнених систем автоматичного управління. Завдання забезпечення якісних перехідних процесів

полягає в мінімізації часу перехідного процесу t_n для будь-якого вихідного параметра при заданому допустимому пере регулюванні $\Delta u_{\max}$.

Відомо, що характер перехідного процесу визначається співвідношенням постійних часу елементів системи. Тому створення високоякісних перехідних процесів пов'язане із забезпеченням потрібних значень зазначених співвідношень шляхом введення коригуючих регуляторів. Загальний підхід до побудови систем автоматичного управління, незалежно від типу електроприводу, включає складання структурних схем, виходячи з виразів для окремих ланок, та впровадження відповідних регуляторів у систему.

У практиці створення систем автоматичного управління електроприводами найбільше застосування знайшли системи підлеглого регулювання параметрів. У таких системах використовується принцип компенсації основних інерційних властивостей об'єктів управління шляхом послідовного включення регуляторів. Кількість послідовно включених регуляторів відповідає кількості регульованих параметрів, і кожен контур системи автоматичного управління структурно складається з двох основних ланок: об'єкта регулювання та регулятора.

Згідно з рекомендаціями [85], обирається двоконтурна система автоматичного управління з дворазовою інтеграцією, яка включає внутрішній контур струму і зовнішній контур, замкнутий по електрорушійній силі (ЕРС) двигуна. Системи регулювання швидкості з зворотним зв'язком за ЕРС використовуються для електроприводів у випадках, коли вимоги до точності регулювання швидкості не є надто високими. Серед електроприводів загальнопромислових механізмів електроприводи цього класу становлять переважну більшість.

2.10 Синтез послідовного коректуючого пристрою контура струму

Контур струму представлений алгоритмічною схемою, зображеною на рис. 2.2.

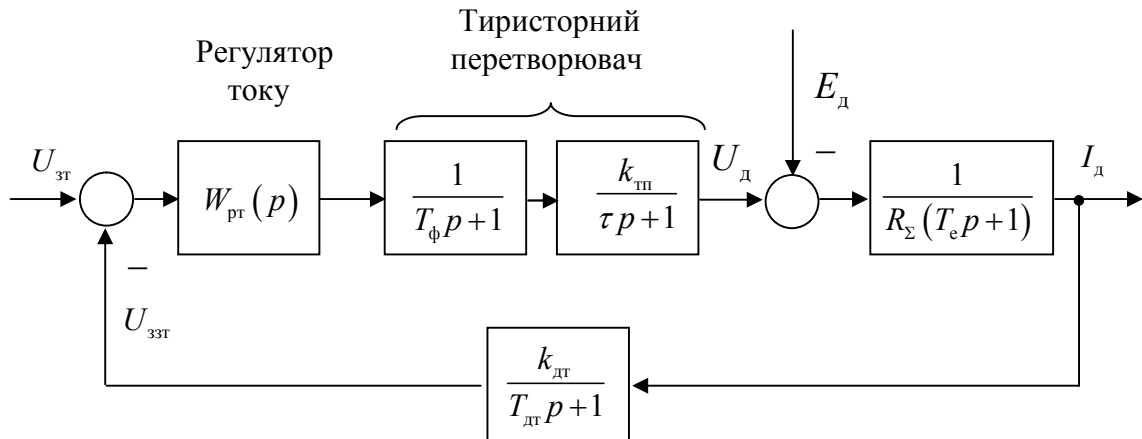


Рисунок 2.2 – Алгоритмічна схема контура струму

Середньостатистичне запізнювання тиристорного перетворювача

$$\tau = \frac{1}{2mf} ,$$

де $m = 6$ – число пульсацій випрямленої ЕРС за період;

f – частота живлячої мережі,

$$\tau = \frac{1}{2 \cdot 6 \cdot 50} 0.0016 \text{ с} .$$

Мала постійна, що не компенсується, контура струму

$$T_{\mu T} = \tau + T_{dT} + T_{\phi} ,$$

T_{dT} , T_{ϕ} – постійні часу датчика струму і фільтру на вході тиристорного перетворювача; приймаємо $T_{dT} + T_{\phi} = 0,001 \text{ с}$, тоді

$$T_{\mu T} = 0,0016 + 0,001 = 0,0026 \text{ с} .$$

Отримаємо розрахункову схему контура струму, показану на рис. 2.3.

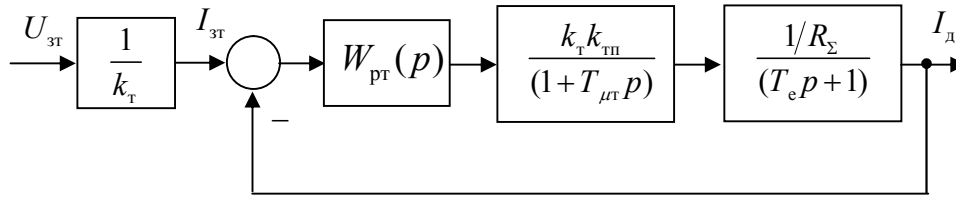


Рисунок 2.3 – Розрахункова алгоритмічна схема контура струму

Відповідно до даної схеми

$$W_{\text{бт}}(p) = W_{\text{рт}}(p) \cdot W_{\text{об}}(p),$$

де $W_{\text{бт}}(p)$ – бажана передавальна функція контура струму; при оптимізації по модульному критерію

$$W_{\text{бт}}(p) = \frac{1}{2T_{\mu\Gamma}p(T_{\mu\Gamma}p + 1)}.$$

$W_{\text{об}}(p)$ – передавальна функція об'єкту регулювання:

$$W_{\text{об}}(p) = \frac{k_{\Gamma}k_{\text{тп}} \cdot 1/R_{\Sigma}}{(T_{\mu\Gamma}p + 1)(T_{\epsilon}p + 1)}.$$

На підставі цих виразів знайдемо передавальну функцію регулятора контура струму $W_{\text{рт}}(p)$

$$W_{\text{рт}}(p) = \frac{W_{\text{бт}}(p)}{W_{\text{об}}(p)} = \frac{\frac{1}{2T_{\mu\Gamma}p(T_{\mu\Gamma}p + 1)}}{\frac{k_{\Gamma}k_{\text{тп}}/R_{\Sigma}}{(T_{\mu\Gamma}p + 1)(T_{\epsilon}p + 1)}} = \frac{T_{\epsilon}p + 1}{2T_{\mu\Gamma}pk_{\Gamma}k_{\text{тп}}/R_{\Sigma}} = k_{\text{рт}} \frac{T_{\epsilon}p + 1}{p}, \quad (2.2)$$

$k_{\text{рт}} = R_{\Sigma} / 2T_{\mu\Gamma}k_{\text{тп}}$ – коефіцієнт посилення регулятора току.

Передавальній функції (2.2) відповідає типовий ПІ-регулятор.

Розрахунок параметрів регулятора і параметрів ланцюга зворотного зв'язку контура току проводиться таким чином.

Передавальна функція типового ПІ-регулятора

$$W_{\text{пр}}(p) = \frac{Z_{33\Gamma}(p)}{Z_{3\Gamma}(p)} = \frac{R_{33\Gamma} + 1/(C_{33\Gamma}p)}{R_{3\Gamma}} = \frac{R_{33\Gamma}C_{33\Gamma}p + 1}{R_{3\Gamma}C_{33\Gamma}p} = \frac{1}{R_{3\Gamma}C_{33\Gamma}} \cdot \frac{R_{33\Gamma}C_{33\Gamma}p + 1}{p}, \quad (2.3)$$

де $Z_{3\Gamma}(p)$, $Z_{33\Gamma}(p)$ – вхідний опір і опір ланцюгу зворотного зв'язку регулятора.

Із зіставлення коефіцієнтів у виразах (2.2) і (2.3) отримуємо

$$C_{33\Gamma}R_{33\Gamma} = T_{\epsilon}; \quad (2.4)$$

$$C_{33\Gamma}R_{3\Gamma} = 2T_{\mu\Gamma}k_{\Gamma}k_{\text{ТП}} / R_{\Sigma}. \quad (2.5)$$

З виразу (2.4) визначається $R_{33\Gamma}$. Приймаємо $C_{33\Gamma} = 2\text{мкФ}$, тоді

$$R_{33\Gamma} = \frac{T_{\epsilon}}{C_{33\Gamma}},$$

$$R_{33\Gamma} = \frac{0,043}{2 \cdot 10^{-6}} = 21,5 \cdot 10^3 \text{ Ом}.$$

З виразу (2.5) визначається $R_{3\Gamma}$

$$R_{3\Gamma} = \frac{2T_{\mu\Gamma}k_{\Gamma}k_{\text{ТП}}}{R_{\Sigma}C_{33\Gamma}},$$

де k_{Γ} – коефіцієнт посилення ланцюга зворотного зв'язку по току:

$$k_{\Gamma} = k_{\text{ш}}k_{\text{дт}} \frac{R_{3\Gamma}}{R_{\Gamma}},$$

$\frac{R_{3\Gamma}}{R_{\Gamma}}$ – коефіцієнт приведення напруги зворотного зв'язку по струму до ла-

нцюга задаючого сигналу, приймаємо $R_{3\Gamma} = R_{\Gamma}$;

$k_{\text{ш}}$ – коефіцієнт посилення шунта

$$k_{\text{ш}} = \frac{U_{\text{ш}}}{I_{\text{ш}}},$$

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

$k_{\text{дт}}$ – коефіцієнт посилення датчика струму; коефіцієнт посилення датчика струму вибираємо так, щоб максимальному робочому перевантаженню двигуна відповідало

$$U_{\text{дт max}} = k_{\text{дт}} k_{\text{ш}} I_{\text{max}} \frac{R_{\text{зт}}}{R_{\text{т}}} \leq 24 \text{ В},$$

звідки

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

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

Приймаємо: $U_{\text{ш}} = 75 \text{ мВ}$ $k_{\text{рп max}} = 2,5$.

$$k_{\text{ш}} = \frac{0,075}{2,5 \cdot 415} = 0,072 \cdot 10^{-3} \text{ В / А},$$

$$k_{\text{дт}} = \frac{24}{0,072 \cdot 10^{-3} \cdot 2,5 \cdot 415} = 321,$$

$$k_{\text{т}} = 0,072 \cdot 10^{-3} \cdot 321 = 0,023 \text{ В / А},$$

$$R_{\text{зт}} = \frac{2T_{\mu\text{т}} k_{\text{т}} k_{\text{тп}}}{R_{\Sigma} C_{\text{зт}}},$$

$$R_{\text{зт}} = \frac{2 \cdot 0,0026 \cdot 0,023 \cdot 20}{0,093 \cdot 2 \cdot 10^{-6}} = 12,8 \cdot 10^3 \text{ Ом},$$

$$R_{\text{т}} = R_{\text{зт}} = 12,3 \cdot 10^3 \text{ Ом}.$$

Розрахуємо максимальну величину вихідної напруги контура швидкості (вхідний сигнал замкнутого контура струму) по формулі:

$$U_{\text{зт max}} = k_{\text{ш}} k_{\text{дт}} I_{\text{дв max}} = k_{\text{ш}} k_{\text{дт}} 2,5 I_{\text{н}},$$

$$U_{\text{зт max}} = 0,072 \cdot 10^{-3} \cdot 321 \cdot 2,5 \cdot 415 = 24 \text{ В}.$$

Передавальна функція замкнутого контура струму, що оптимізується:

$$W_{\tau}(p) = \frac{1/k_{\tau}}{2T_{\mu\tau}^2 p^2 + 2T_{\mu\tau} p + 1}.$$

2.11 Синтез послідовного коректуючого пристрою контура ЕРС

Для механізму підйому застосовуємо двоконтурну, дворазовоінтегруючу систему. Цей вибір обумовлений тим, що механізм підйому є одним з основних механізмів мостового крана і від якісних показників роботи системи управління залежить якість і продуктивність роботи всього мостового крана. У контурі швидкості тахогенератор не використовують, а контур замикають по ЕРС двигуна. Для вимірювання ЕРС використовується датчик ЕРС. На вході датчика відбувається підсумовування сигналів, пропорційних напрузі двигуна і падінню напруги в активному опорі якірного ланцюга, – сигналу струмової компенсації.

На рис. 2.4 показана принципова схема системи регулювання швидкості із зворотним зв'язком по ЕРС двигуна.

Сигнал, пропорційний напрузі двигуна, знімається з опору R_d і подається на один з входів датчиків ЕРС ДЕ через фільтр $R_n C_n$, що виконує функцію аперіодичної ланки з постійною часу T_a у ланцюзі зворотного зв'язку по напрузі. На другий вхід ДЕ подається сигнал струмової компенсації.

Як датчик ЕРС застосовується операційний підсилювач з гальванічно розділеними входом і виходом. Цей підсилювач має два виходи, один з яких служить для обхвату підсилювача зворотним зв'язком через вбудований опір $R_{де}$. Другий вихід підсилювача, гальванічно окремий від першого, призначений для зв'язку з системою регулювання.

На вході системи встановлений фільтр $R_{зе} C_{зе}$, що виконує функцію аперіодичної ланки з постійною часу T_a . Цей фільтр необхідний для усунення підвищеного перерегулювання струму при обуренні за завданням.

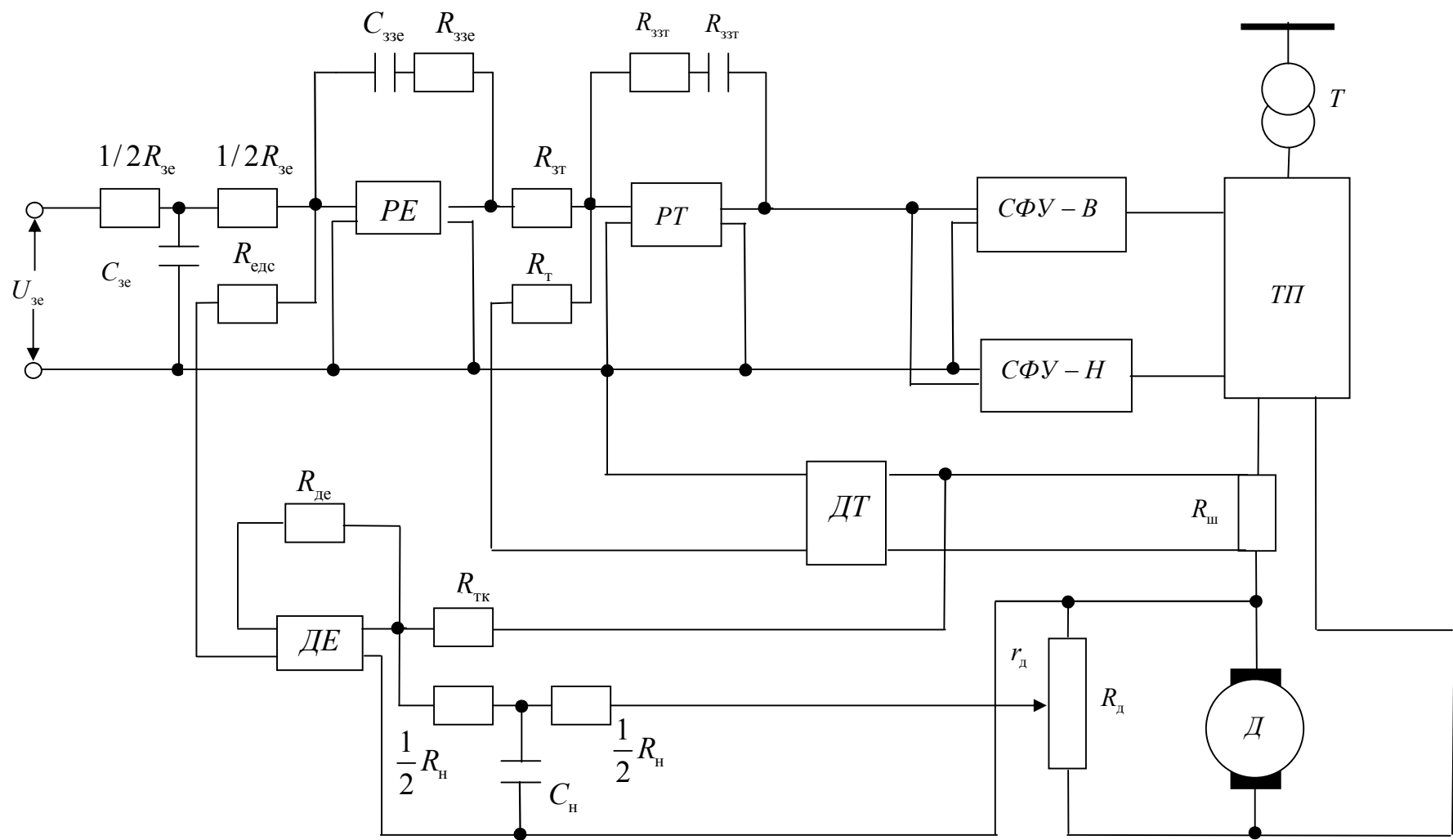


Рисунок 2.4 – Принципова схема системи

На рис. 2.5. показана алгоритмічна схема системи.

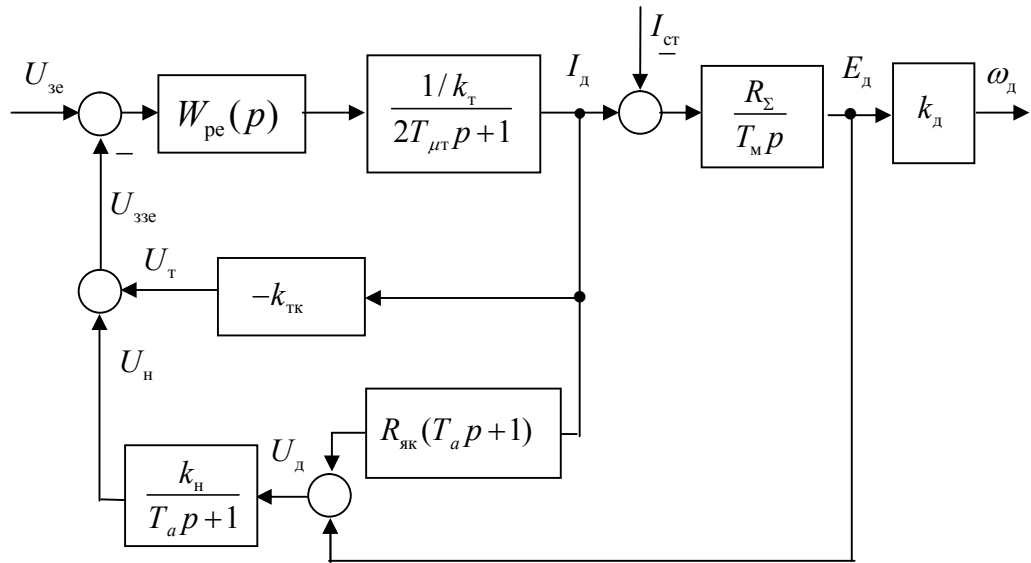


Рисунок 2.5 – Алгоритмічна схема системи з представленням оптимізованого контуру струму у вигляді аперіодичної ланки

Оптимізований по модульному критерію контур струму представлений аперіодичною ланкою

$$W_{\tau}(p) = \frac{1/k_{\tau}}{2T_{\mu\tau}p + 1}.$$

Таке перетворення виходить при зневазі доданком $2T_{\mu\tau}^2 p^2$ у знаменнику передавальної функції оптимізованого за модульним критерієм контуру струму.

Коефіцієнти посилення ланцюга струмової компенсації вибирається таким чином:

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}},$$

де $k_{\text{н}}$ – коефіцієнт посилення ланцюга зворотного зв'язку по напрузі.

Сигнал на вході датчика ЕРС (сигнал зворотного зв'язку по ЕРС):

$$U_{\text{ззе}} = U_{\text{н}} - U_{\text{т}} = \frac{k_{\text{н}}}{T_a p + 1} E_{\text{д}}.$$

Алгоритмічна схема рис.2.5 шляхом еквівалентних перетворювань перетвориться в схему рис. 2.6. На рис 2.7. показана розрахункова схема з одиничним зворотним зв'язком.

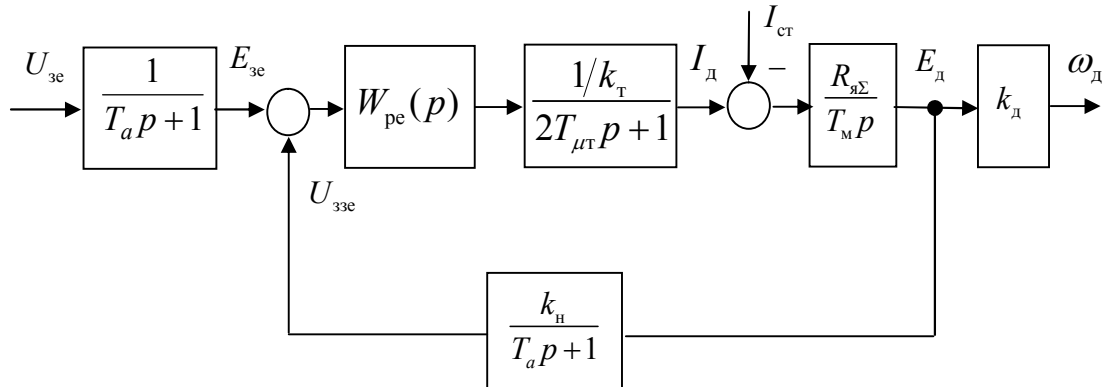


Рисунок 2.6 – Перетворена алгоритмічна схема системи

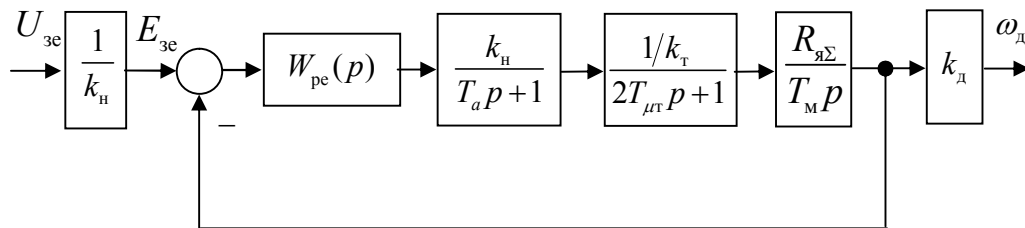


Рисунок 2.7 – Розрахункова алгоритмічна схема системи з одиничним зворотним зв'язком

Для оптимізації контура ЕРС по симетричному критерію необхідно використовувати ПІ-регулятор з передавальною функцією:

$$W_{pe}(p) = \frac{T_M \cdot k_T}{2 \cdot T_{\mu\epsilon} \cdot k_H \cdot R_\Sigma} \cdot \frac{4T_{\mu\epsilon}p + 1}{4T_{\mu\epsilon}p}, \quad (2.6)$$

де $T_{\mu\epsilon} = 2T_{\mu\Gamma} + T_a$ – мала некомпенсована постійна контура ЕРС

$$T_{\mu\epsilon} = 2 \cdot 0,0026 + 0,035 = 0,0375 \text{ с.}$$

Оскільки до складу $T_{\mu\epsilon}$ входить T_a , то швидкодія системи із зворотним зв'язком по ЕРС нижче, ніж системи з використанням тахогенератора.

Коефіцієнт посилення по напрузі рівний:

$$k_H = \frac{U_{ze\max}}{E_{d\max}},$$

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

$U_{ze\max}$ – напруга завдання відповідна $E_{d\max}$ приймаємо $U_{ze\max} = 24 \text{ В}$.

У якості $E_{d\max}$ приймемо ЕРС в режимі ідеального холостого ходу, тобто $E_{d\max} = U_H = 440 \text{ В}$ тоді

$$k_H = \frac{24}{220} = 0,109.$$

Зіставляючи передавальну функцію (2.6) з передавальною функцією типового ПІ-регулятора, отримуємо наступні рівняння:

$$R_{33\epsilon} C_{33\epsilon} = 4T_{\mu\epsilon}, \quad (2.7)$$

$$\frac{R_{33\epsilon}}{R_{3\epsilon}} = \frac{T_M k_T}{2T_{\mu\epsilon} k_H R_\Sigma}. \quad (2.8)$$

Задамося ємністю $C_{33\epsilon} = 6 \text{ мкФ}$, тоді з (2.7) визначаємо

$$R_{33\epsilon} = \frac{4T_{\mu\epsilon}}{C_{33\epsilon}},$$

$$R_{33\epsilon} = \frac{4 \cdot 0,0376}{6 \cdot 10^{-6}} = 25 \cdot 10^3 \text{ Ом}.$$

Із співвідношення (2.8) знаходимо

$$R_{3\epsilon} = \frac{R_{33\epsilon} 2T_{\mu\epsilon} k_H R_\Sigma}{T_M k_T},$$

$$R_{зе} = \frac{25 \cdot 10^3 \cdot 2 \cdot 0,0376 \cdot 0,109 \cdot 0,093}{1,2 \cdot 0,023} = 6,9 \cdot 10^3 \text{ Ом}.$$

Приймаємо $R_{едс} = R_{зе} = 6,9 \cdot 10^3 \text{ Ом}.$

З виразу для коефіцієнта посилення зворотного зв'язку по напрузі

$$k_{\text{н}} = k_{\text{д.дв}} \frac{R_{зе}}{R_{\text{н}}} \frac{R_{\text{де}}}{R_{\text{едс}}}.$$

Визначимо

$$R_{\text{н}} = k_{\text{д.дв}} \frac{R_{зе} R_{\text{де}}}{R_{\text{едс}} k_{\text{н}}},$$

Тут $R_{\text{де}} = 50 \text{ кОм}$ – вбудований опір датчика ЕРС.

$k_{\text{д.дв}}$ – коефіцієнт посилення діляника напруги; значення $k_{\text{д.дв}}$ визначається з умови отримання сигналу 24В на вході датчика ЕРС при максимальній напрузі двигуна $U_{\text{д max}}$. Приймаємо в якості $U_{\text{д max}}$ номінальна напруга двигуна, отримуємо

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

$$R_{\text{н}} = 0,109 \cdot \frac{6,9 \cdot 10^3 \cdot 50 \cdot 10^3}{6,9 \cdot 10^3 \cdot 0,109} = 50 \cdot 10^3 \text{ Ом}.$$

Передавальний коефіцієнт струмової компенсації:

$$k_{\text{тк}} = k_{\text{н}} R_{\text{як}},$$

$$k_{\text{тк}} = 0,109 \cdot 0,009 = 8,3 \cdot 10^{-3}.$$

З іншого боку

$$k_{\text{тк}} = k_{\text{ш}} \frac{R_{зе} R_{\text{де}}}{R_{\text{едс}} R_{\text{тк}}},$$

звідки

$$R_{\text{TK}} = \frac{k_{\text{ш}}}{k_{\text{TK}}} \cdot \frac{R_{\text{зе}} R_{\text{де}}}{R_{\text{едс}}},$$

$$R_{\text{TK}} = \frac{0,072 \cdot 10^{-3}}{8,3 \cdot 10^{-3}} \cdot \frac{6,9 \cdot 10^3 \cdot 50 \cdot 10^3}{6,9 \cdot 10^3} = 4,3 \cdot 10^3 \text{ Ом}.$$

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

$$\frac{R_{\text{зе}}}{4} C_{\text{зе}} = \frac{R_{\text{н}}}{4} C_{\text{н}} = T_a,$$

$$C_{\text{зе}} = \frac{4T_a}{R_{\text{зе}}},$$

$$C_{\text{зе}} = \frac{4 \cdot 0,035}{6,9 \cdot 10^3} = 20,3 \cdot 10^{-6} \text{ Ф},$$

$$C_{\text{н}} = \frac{4T_a}{R_{\text{н}}},$$

$$C_{\text{н}} = \frac{4 \cdot 0,035}{50 \cdot 10^3} = 2,8 \cdot 10^{-6} \text{ Ф}.$$

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

$$W_{\omega_d U_{\text{зе}}}(p) = \frac{\omega_d(p)}{U_{\text{зе}}(p)} = \frac{k_d}{k_{\text{н}}} \cdot \frac{B(p)}{A(p)};$$

$$W_{\omega_d I_{\text{ст}}}(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)} = \frac{8T_{\mu\epsilon}^2 k_d R_{\Sigma}}{T_{\text{м}}} \cdot \frac{p(1 + T_{\mu\epsilon} p)}{A(p)};$$

$$W_{I_d U_{\text{зе}}}(p) = \frac{I_d(p)}{U_{\text{зе}}(p)} = \frac{T_{\text{м}}}{k_{\text{н}} R_{\Sigma}} \cdot \frac{pB(p)}{A(p)};$$

$$W_{I_d I_{ct}}(p) = \frac{I_d(p)}{I_{ct}(p)} = \frac{B(p)}{A(p)},$$

де $A(p)$, $B(p)$ – поліноми, що мають наступний вигляд:

$$A(p) = 8T_{\mu\epsilon}^3 p^3 + 8T_{\mu\epsilon}^2 p^2 + 4T_{\mu\epsilon} p + 1;$$

$$B(p) = 1 + 4T_{\mu\epsilon} p$$

Алгоритмічна схема системи показана на рис. 2.8. На схемі тиристорний перетворювач представлений у вигляді аперіодичної ланки з постійною часу $T_{\text{тп}} = T_{\text{ф}} + \tau$.

2.11 Дослідження динамічних характеристик системи

Для побудови перехідних процесів системи була розроблена схема моделі системи, показана на рис.2.9. Схема розроблена на ЕОМ за допомогою пакету MATLAB в режимі SIMULINK. Перехідні процеси досліджувалися також по векторний-матричному рівнянню системи, порядок складання якого дається в подальших розділах. На рис. 2.10-2.11 приведені графіки перехідних процесів швидкості ω_d і струму I_d двигуна по задаючій дії. При настроюванні системи на симетричний оптимум є значне (43,3%) перерегулювання швидкості. Для зменшення перерегулювання на вихід системи доцільно включити фільтр з постійною часу, рівною $4T_{\mu\epsilon}$, унаслідок чого перерегулювання знижується до 8,1%, проте при цьому збільшується час першого узгодження. Графіки перехідних процесів системи з фільтром на вході показані на рис 2.10, криві 2.

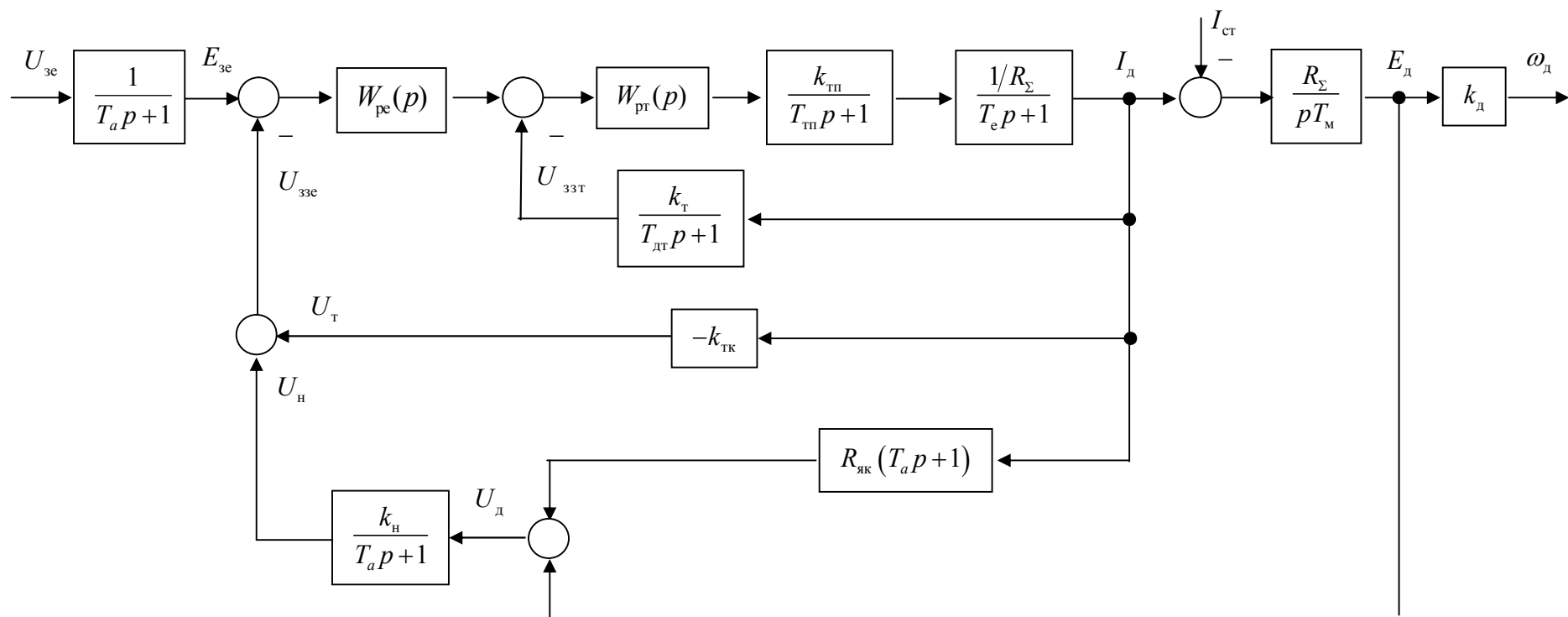


Рисунок 2.9 – Алгоритмічна схема системи



Рисунок 2.10 – Схема моделі системи, розроблена в MATLAB

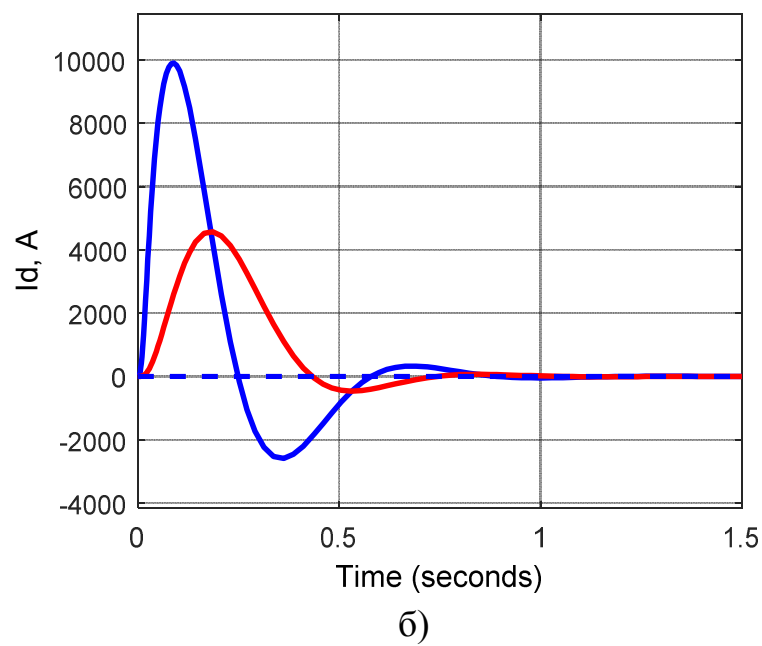
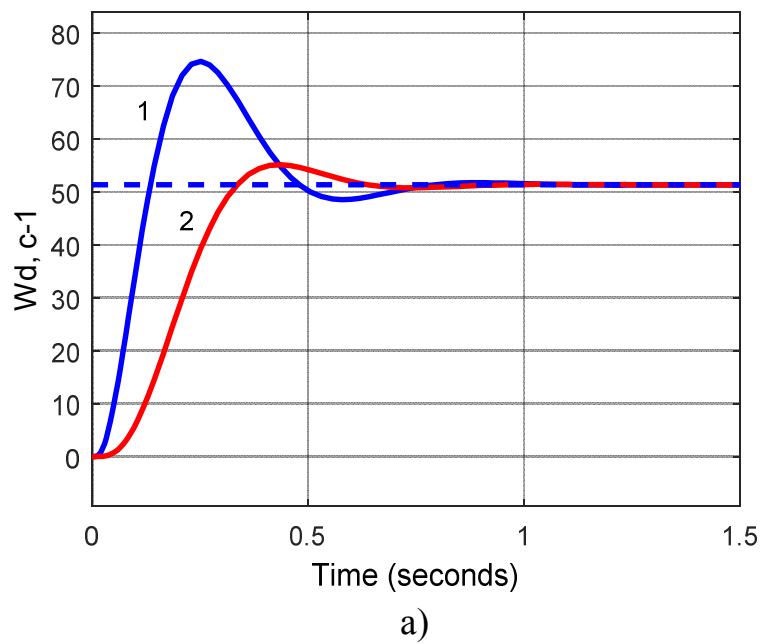


Рисунок 2.11 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) по задаючій дії

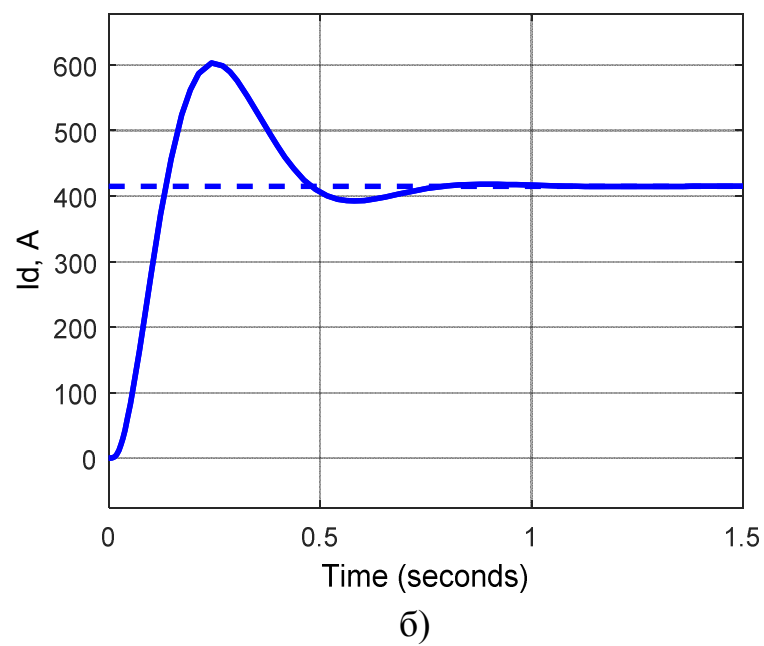
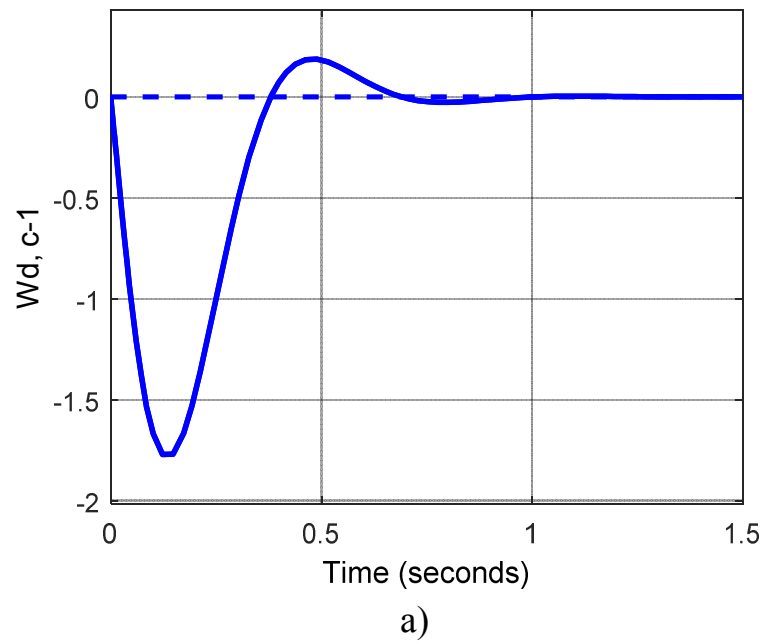


Рисунок 2.12 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ а)
і струму $I_d = f(t)$ б) по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На підставі аналізу технічних вимоги, що пред'являється до електроприводів допоміжних механізмів прокатних станів для електроприводу механізму переміщення візка зливковогоз кільцевої зливкоподачі вибраний електропривод постійного струму, виконаний за системою ТП–Д.

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

3. Виконано розрахунок параметрів силового ланцюга 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 переходових 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, а для контурів струму і ЕРС – ПІ-регулятори, унаслідок чого контур струму оптимізований по модульному критерію, а контур ЕРС – по симетричному 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. Під час налаштування системи на симетричний оптимум спостерігається значне (43,3) перерегулювання швидкості. Щоб зменшити перерегулювання, до виходу системи додано фільтр з постійною часу, рівною $4T_{\mu e}$, унаслідок чого перерегулювання знижене до 8,1%.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ЕЛЕМЕНТІВ

3.1 Математична модель двомасової системи

У попередніх розділах механічна частина електроприводу представлялася жорсткою приведеною ланкою, що не враховує впливу пружності реальних механічних зв'язків (вали, сполучні муфти, зубчаті зачеплення). Із-за кінцевої жорсткості цих зв'язків механічна частина електроприводу є пружною системою, прикладення до якої управляючої дії (моменту двигуна) або збурюючої (статичного навантаження), викликають коливання зв'язаних мас, що збільшує максимальні навантаження зв'язків і ускладнює точність відпрацювання необхідних рухів робочого органу машини.

Якщо виділити основні маси, які визначають рух системи, і найменші жорсткості зв'язків між ними, то кінематична схема механізму може бути приведена до двомасової системи (рис.3.1).

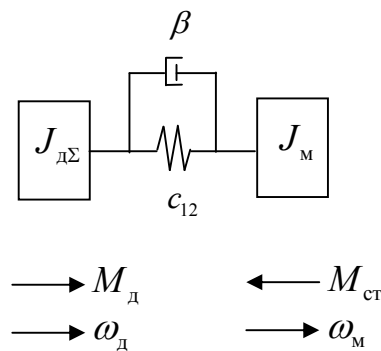


Рисунок 3.1 – Двомасова розрахункова схема

Параметрами двомасової системи є сумарний момент інерції якоря двигуна і жорстко пов'язаних з ним елементів $J_{дΣ}$, сумарний приведений до швидкості двигуна момент інерції механізму $J_{м}$ і безінерційний пружний зв'язок між

ними, що характеризується приведеною еквівалентною жорсткістю c_{12} і коефіцієнтом внутрішнього в'язкого тертя каната між масами β .

Рівняння динаміки, що описують процеси в двомасовій системі при приведенні всіх елементів до швидкості валу двигуна, мають вид:

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

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

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

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

$M_{\text{ст}}$ – момент статичного навантаження.

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

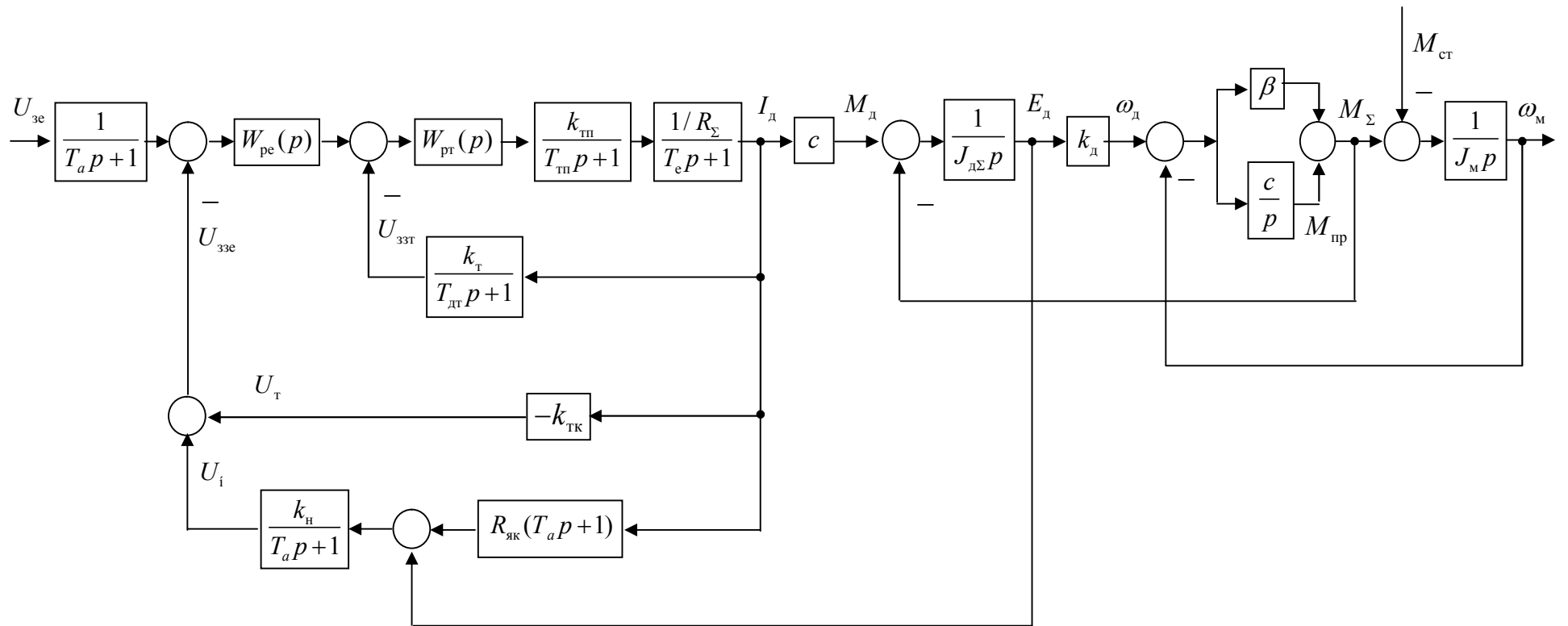


Рисунок 3.2 – Алгоритмічна схема двомасової системи

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

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\text{т}}^2 k_{\text{т}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\text{т}}^2 k_{\text{т}}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\text{т}}^2} M_{\text{д}} - \frac{1}{T_{\mu\text{т}}} P + \frac{1}{T_{\mu\text{т}}^2 k_{\text{т}}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (3.3)$$

При записі системи (3.3) прийняті наступні позначення.

$U_{\text{іе}}$ – напруга на виході інтегральної частини регулятора ЕРС;

P – ривок.

$k_{\text{пе}}, k_{\text{іе}}$ – коефіцієнти підсилення пропорційної і інтегральної частин регулятора ЕРС; значення указаних параметрів можуть бути отримані з виразу передатної функції регулятора ЕРС:

$$W_{\text{пе}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} \cdot \frac{4T_{\mu\text{е}} p + 1}{4T_{\mu\text{е}} p} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} + \frac{T_{\text{м}} k_{\text{т}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}} \frac{1}{p} = k_{\text{пе}} + \frac{k_{\text{іе}}}{p},$$

$$k_{\text{пе}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}}, \quad k_{\text{іе}} = \frac{T_{\text{м}} k_{\text{т}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}}.$$

При записі системи (3.3) замість T_m записано відповідний вираз

$$T_m = \frac{R_\Sigma \cdot J_{d\Sigma}}{c^2} = R_\Sigma J_{d\Sigma} \cdot k_d^2;$$

$$k_d = 1/c.$$

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

$$\begin{cases} \frac{d\omega_m}{dt} = \frac{1}{J_m} M_{\text{пр}} + \frac{\beta}{J_m} \omega_d - \frac{\beta}{J_m} \omega_m - \frac{1}{J_m} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_d - c_{12} \omega_m; \\ \frac{d\omega_d}{dt} = \frac{1}{J_{d\Sigma}} M_d - \frac{1}{J_{d\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{d\Sigma}} \omega_d + \frac{\beta}{J_{d\Sigma}} \omega_m. \end{cases} \quad (3.4)$$

Об'єднаємо системи (3.3) і (3.4), одержимо наступну систему рівнянь стану двомасової системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна:

$$\left\{ \begin{aligned}
\frac{d\omega_{\text{м}}}{dt} &= \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \\
\frac{dM_{\text{пр}}}{dt} &= c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\
\frac{d\omega_{\text{д}}}{dt} &= \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\
\frac{dM_{\text{д}}}{dt} &= \frac{1}{k_{\text{д}}} P; \\
\frac{dP}{dt} &= \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\Gamma}^2} M_{\text{д}} - \frac{1}{T_{\mu\Gamma}} P + \frac{1}{T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{іе}}; \\
\frac{dE_{\text{зе}}}{dt} &= \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\
\frac{dU_{\text{ззе}}}{dt} &= \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\
\frac{dU_{\text{іе}}}{dt} &= k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}.
\end{aligned} \right. \quad (3.5)$$

Алгоритмічна схема двомасової системи, що відповідає рівнянням (3.5), показана на рис. (3.3). На схемі оптимізований контур струму представлений у вигляді послідовного з'єднання аперіодичної і інтегруючої ланки, охоплених одиничним зворотним зв'язком. Вихідний сигнал першої ланки пропорційний ривку електроприводу. P .

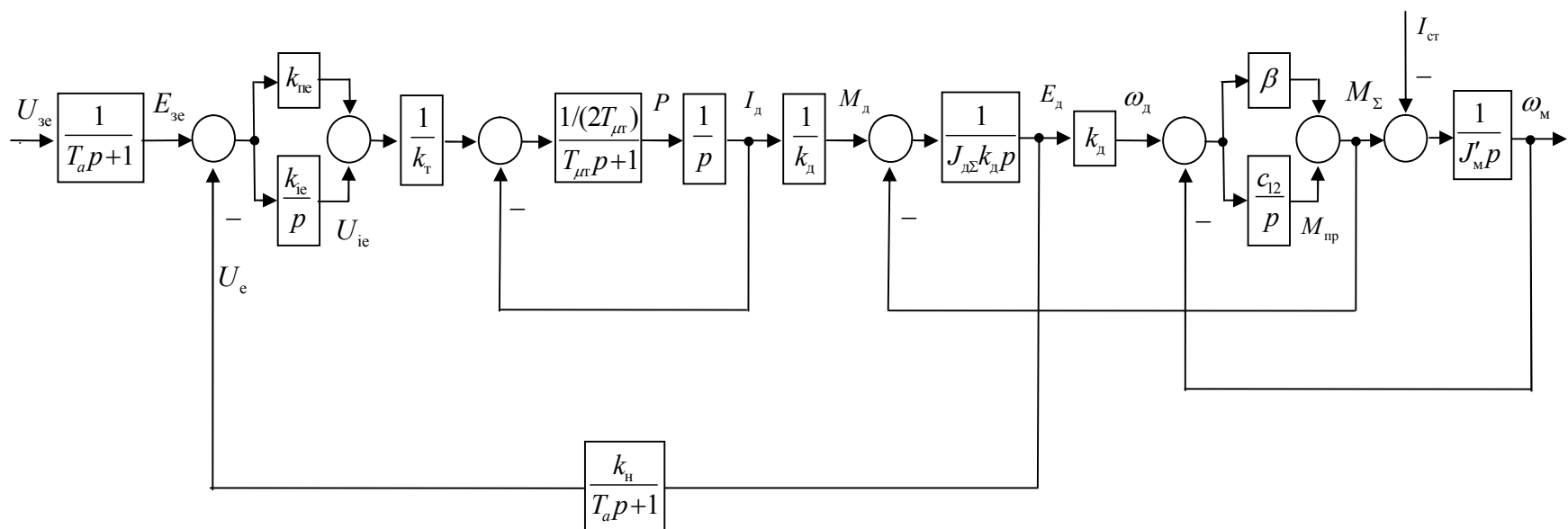


Рисунок 3.3 – Спрощена алгоритмічна схема двомасової системи

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

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

$$\vec{X}(t) = \left\{ \omega_m(t), M_{\text{пр}}(t), \omega_d(t), M_d(t), P(t), E_{\text{зе}}(t), U_{\text{зе}}(t), U_{\text{іе}}(t) \right\}^T,$$

$$\vec{U}(t) = \left\{ U_{\text{зе}}(t), M_{\text{ст}}(t) \right\}^T,$$

де T – знак транспонування.

Рівняння стану системи і рівняння виходу має вид

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.6)$$

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:

$$\begin{array}{cccccccc}
& \omega_{\text{м}} & M_{\text{пп}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{зс}} & U_{\text{зс}} & U_{\text{иэ}} \\
A=0 & \left| \begin{array}{cccccccc}
-\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{д}}} & 0 & 0 & 0 & 0 & 0 \\
-C_{12} & 0 & C_{12} & 0 & 0 & 0 & 0 & 0 \\
\frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\
0 & 0 & 0 & -\frac{k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{пе}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{пе}}}{2T_{\text{мт}}^2 k_{\text{т}}} & \frac{1}{2T_{\text{мт}}^2 k_{\text{т}}} \\
0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 & 0 \\
0 & & \frac{k_{\text{н}}}{T_a k_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\
0 & 0 & 0 & 0 & 0 & k_{\text{иэ}} & -k_{\text{иэ}} & 0
\end{array} \right|
\end{array}$$

$$U_{\text{зс}} \quad M_{\text{сг}}$$

$$B = \left| \begin{array}{cc}
0 & -\frac{1}{J_{\text{м}}} \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\frac{1}{T_a} & 0 \\
0 & 0 \\
0 & 0
\end{array} \right|$$

3.2 Розрахунок перехідних процесів двомасової системи

Момент інерції двигуна і жорстко пов'язаних з ним елементів системи приймаємо рівним $J_{д\Sigma} = 1,2 \cdot J_{дв} = 1,2 \cdot 16,25 = 19,5 \text{ кг} \cdot \text{м}^2$

Приведений момент інерції механізму (завантаженого зливковоза)

$$J_{пр} = \frac{(m_{зл} + m_{в}) \cdot D_{к}^2}{2i^2} + \frac{J_{об}}{i^2},$$

$$J_{м} = J_{пр} = \frac{970}{7,78^2} + \frac{(2 \cdot 12,5 + 38) \cdot 1,1^2 \cdot 10^3}{4 \cdot 7,78^2} = 16 + 191 = 207 \text{ кг} \cdot \text{м}^2$$

Решта параметрів визначені у попередніх розділах.

Розрахунок перехідних процесів проводився з використанням пакету прикладних програм MATLAB. Розрахунок перехідних процесів проводився на ЕОМ з використанням пакету прикладних програм MATLAB. Для дослідження використана модель системи, приведена на рис. 3.4, а також векторно-матричне рівняння (3.6).

У додатку Д приведений текст програми розрахунку, в якому містяться початкові дані, матриці **A**, **B** і **C** одномасової і двомасової системи.

Графіки перехідних процесів двомасової системи показані на рис. 3.5-3.8. Для зменшення інформації, що виводиться, на ЕОМ були виведені на екран монітора, і відповідно роздруковані перші чотири змінні стану. Як видно з графіків, мають місце значні коливання основних координат системи, що несприятливо позначається на роботі механізму, збільшуючи навантаження на елементи системи в перехідних режимах.

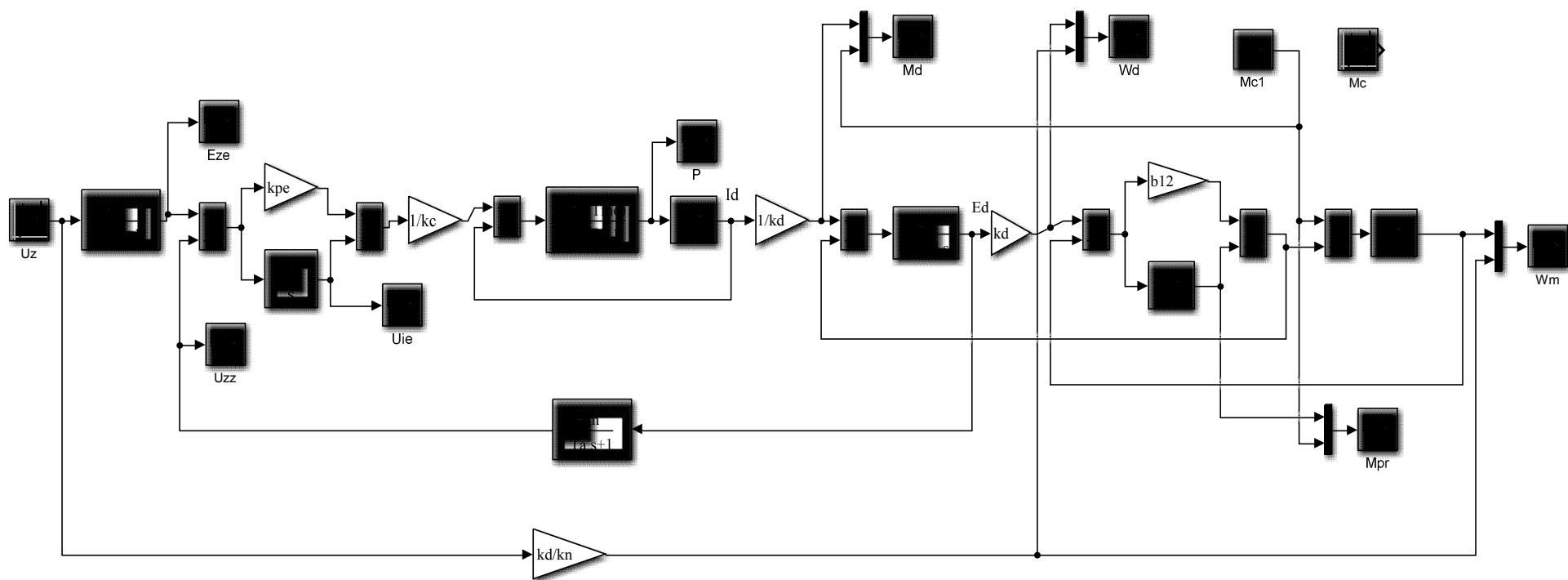
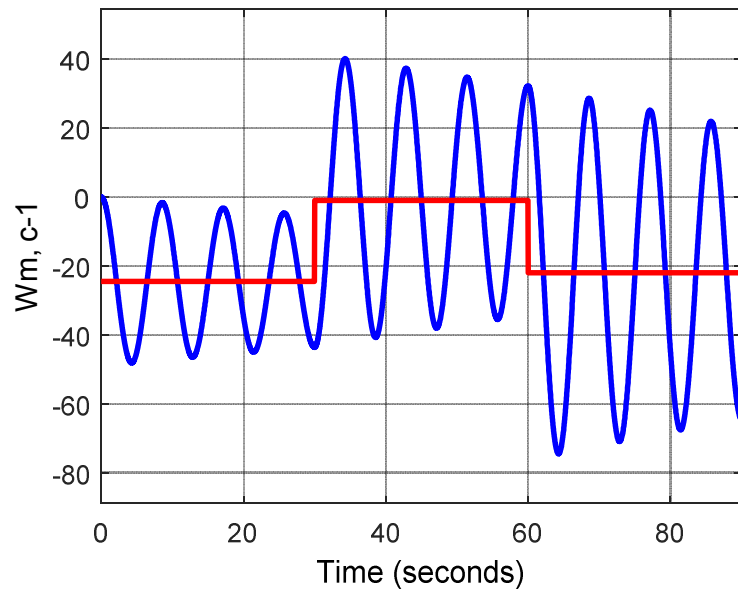
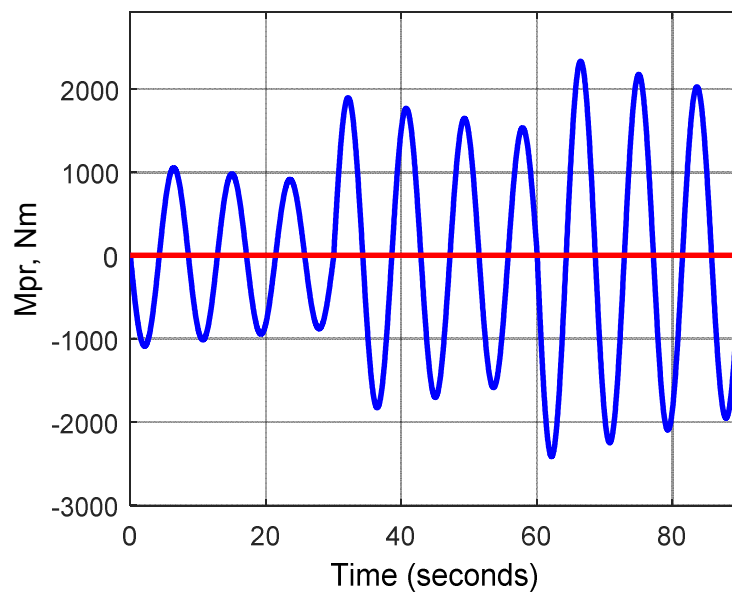


Рисунок 3.4 - Схема моделі двомасової системи, розроблена на EOM



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

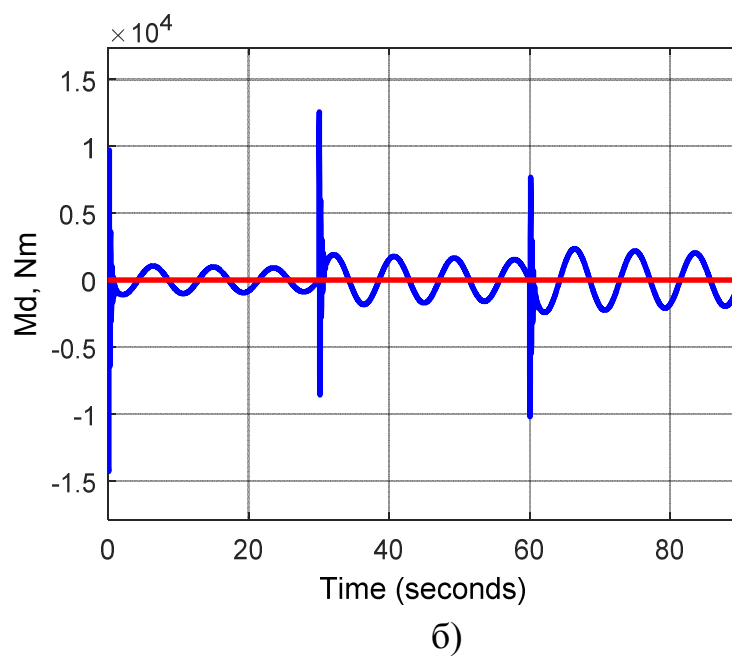
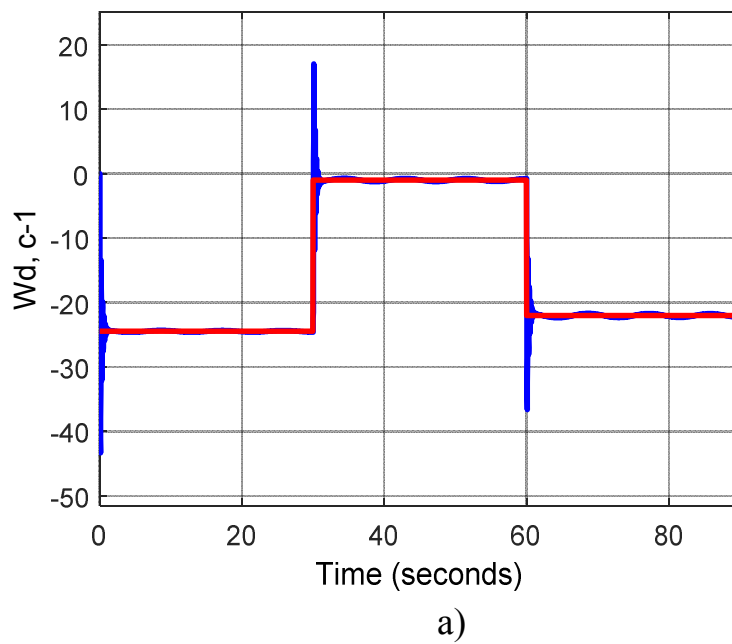
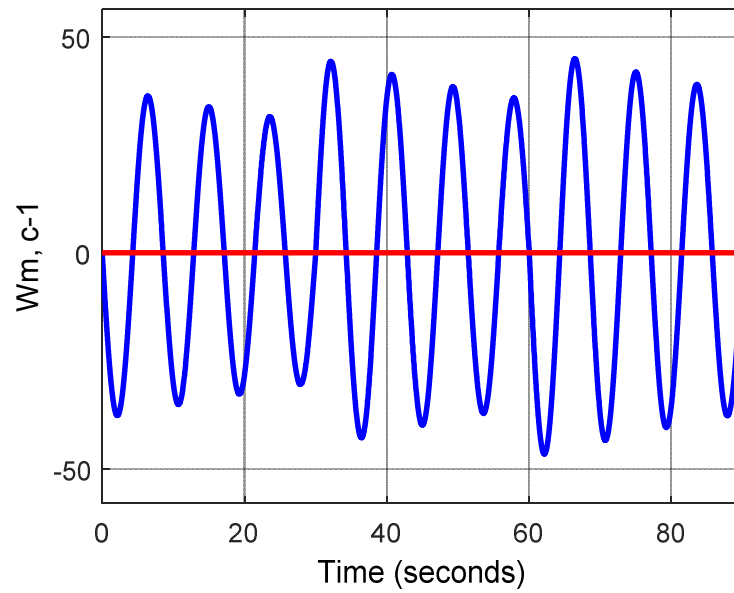
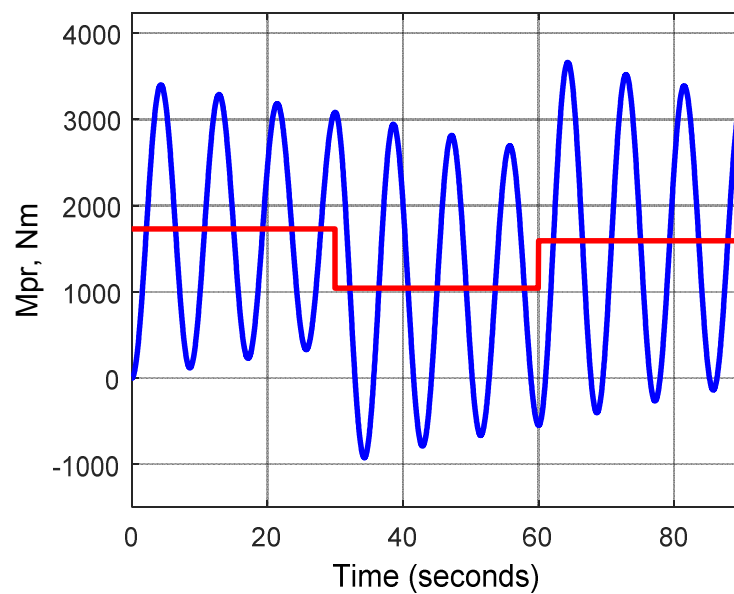


Рисунок 3.6 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по задаючій дії

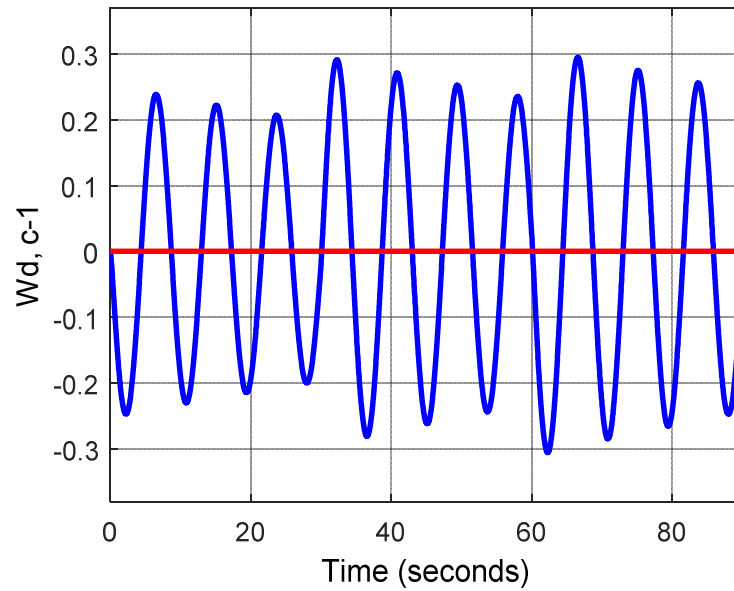


a)

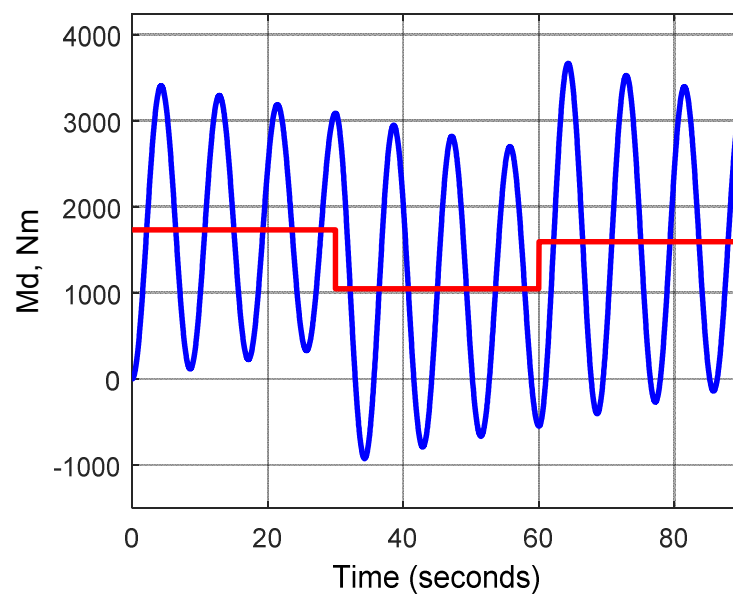


б)

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



a)



б)

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 зливковоза з урахуванням кінцевої жорсткістю зв'язку між двигунами і візком. 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 ускладнюючу точність обробки необхідних рухів і переміщень. Показано, що кінематична схема механічної частини системи управління механізмом переміщення візка з урахуванням податливості елементів механічної частини електроприводу може розглядатися як двомасова.

2. Виконано розробку математичної моделі. Отримано рівняння стану двомасової системи і рівняння стану двомасової системи у векторному – матричній формі, визначені елементи вектора стану, тобто змінні стану системи; визначений вектор зовнішніх дій, а також матриці стану **A** і управління **B**. Розроблена алгоритмічна схема двомасової системи і спрощена алгоритмічна схема 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'єкту у вигляді нейронної мережі, яка потім використовується на етапі синтезу для розробки регулятора. Для кожної з трьох архітектур використовується той самий процес ідентифікації, проте етапи синтезу суттєво відрізняються.

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

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

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

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

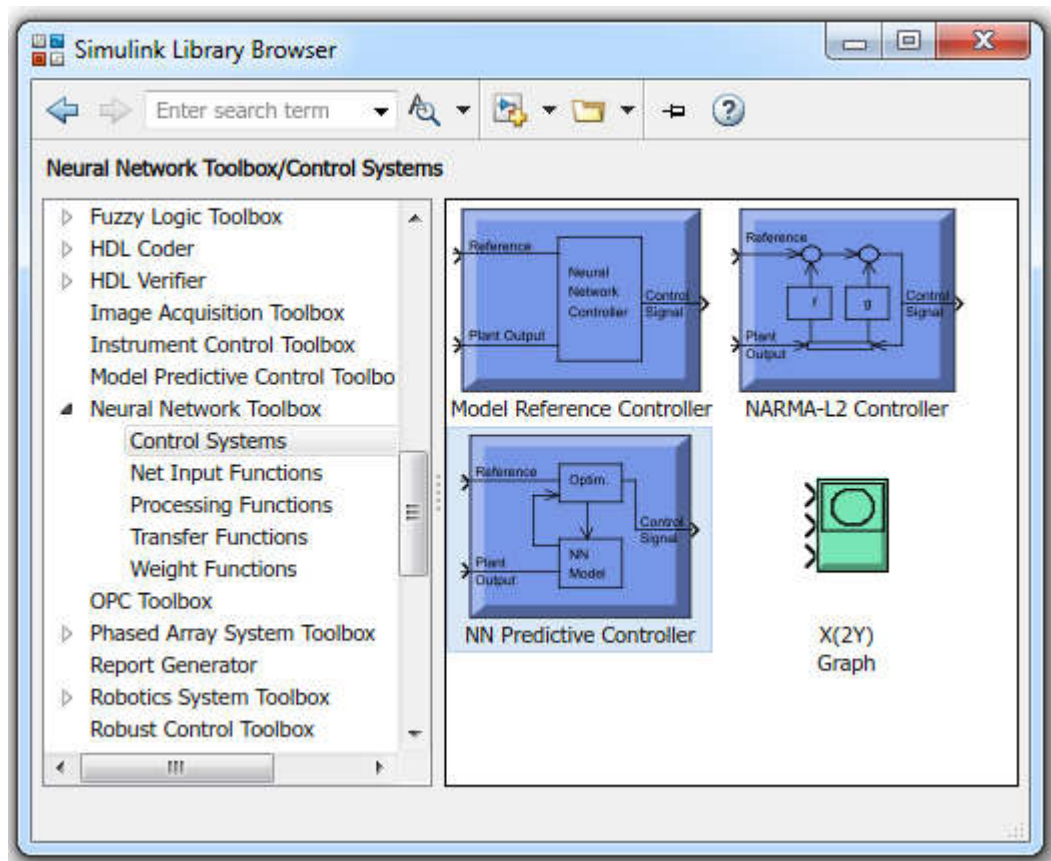


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

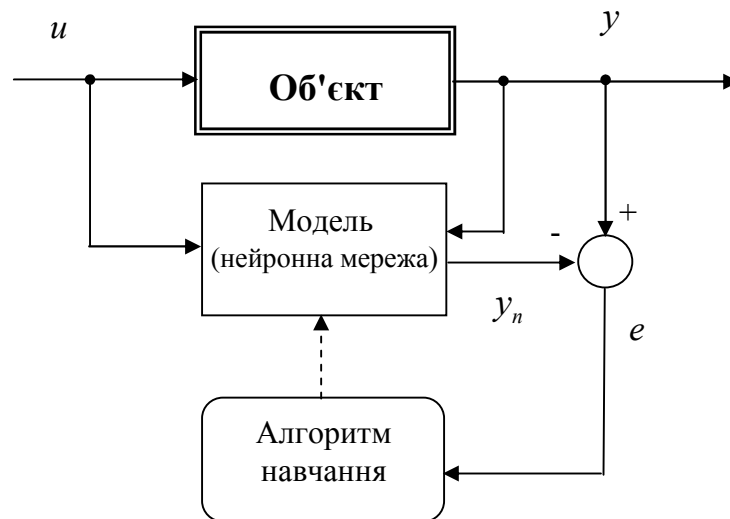


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

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

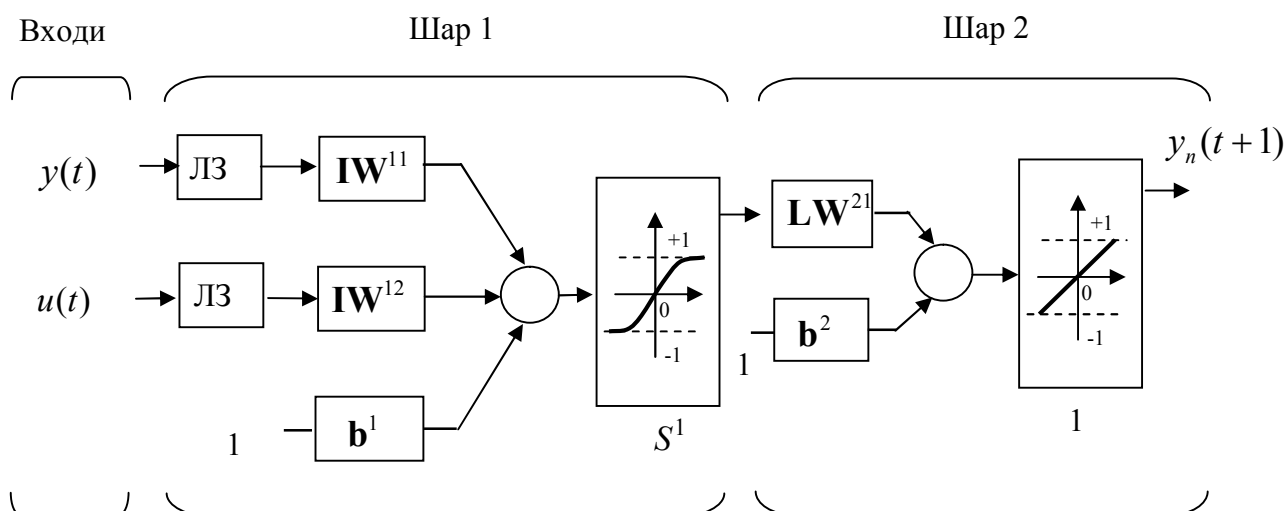


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

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

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

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

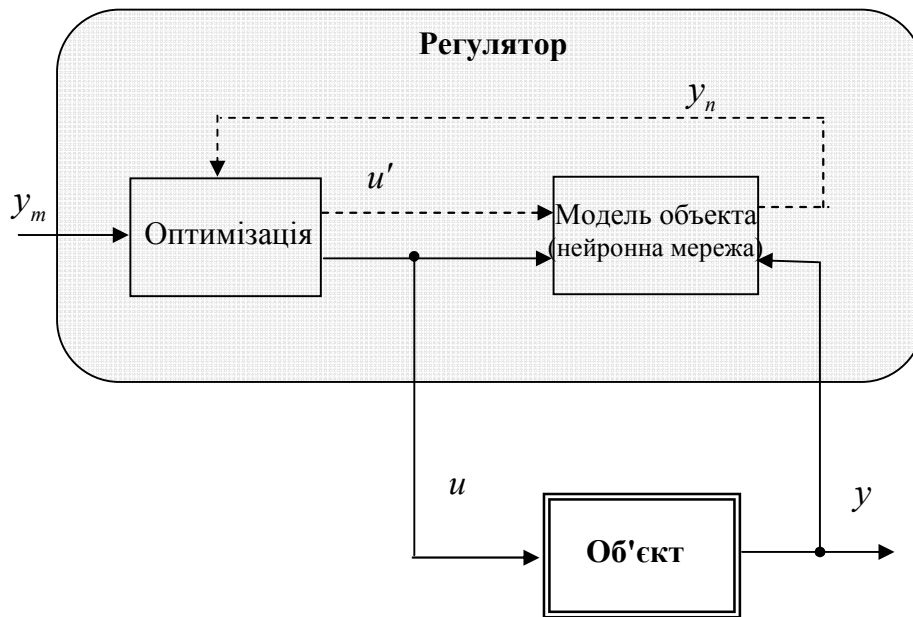


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

Prest3sim2 – нейромережева модель управління використовувана 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.

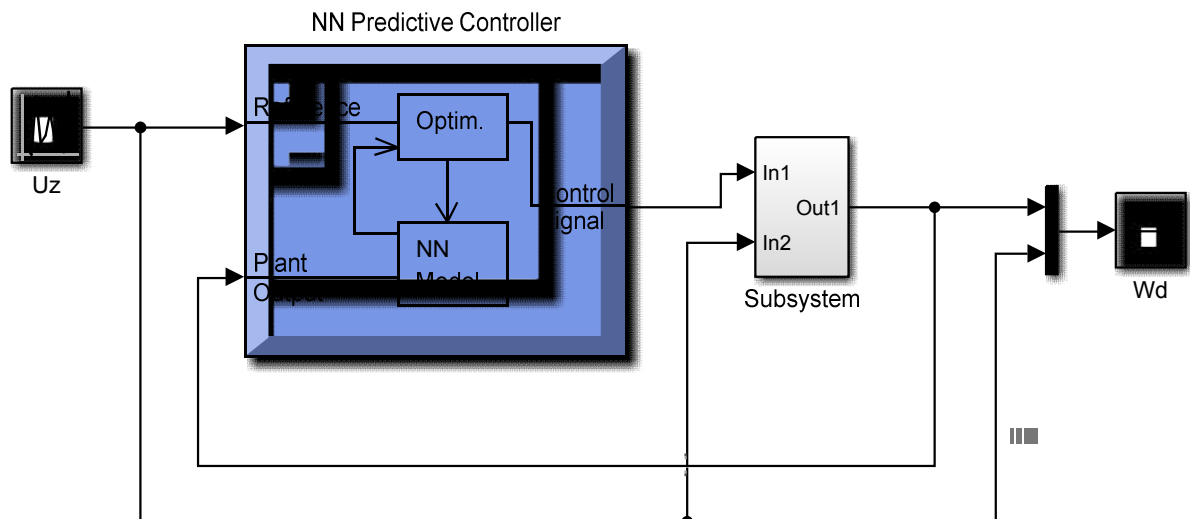


Рисунок 4.5 – Схема системи управління з нейрорегулятором

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-

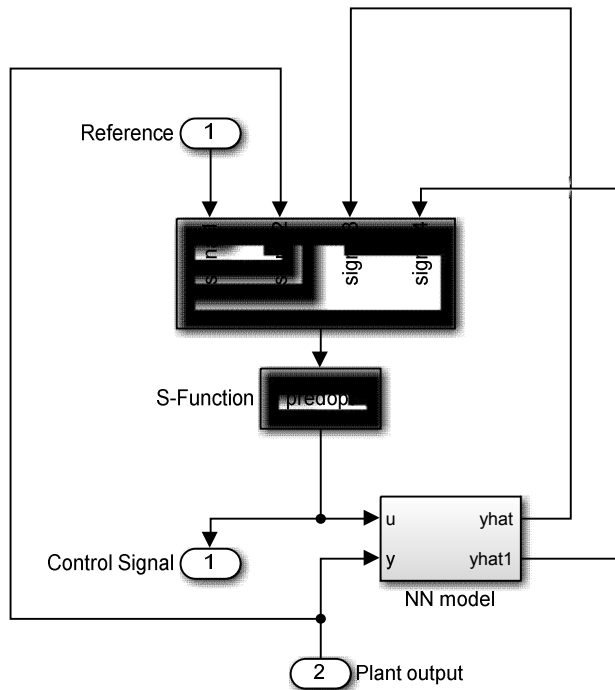


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

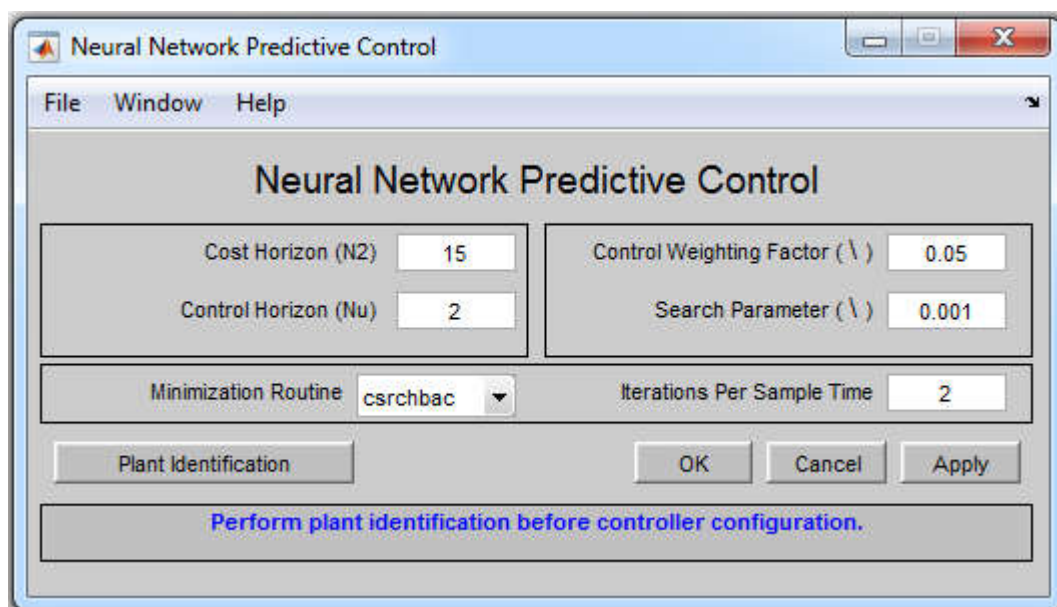


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

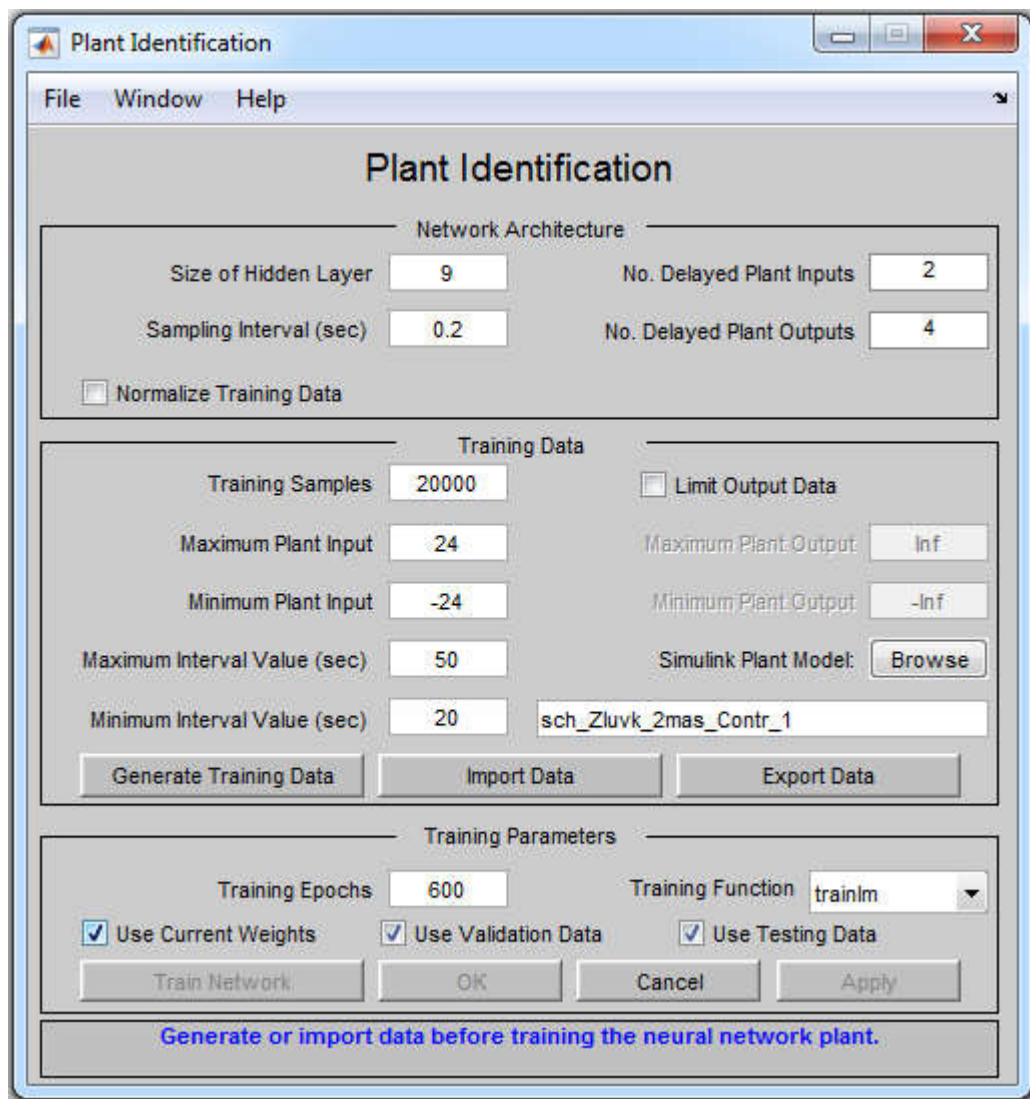


Рисунок 4.8 – Вікно ідентифікації керованої системи

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 або рис.4.10

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.

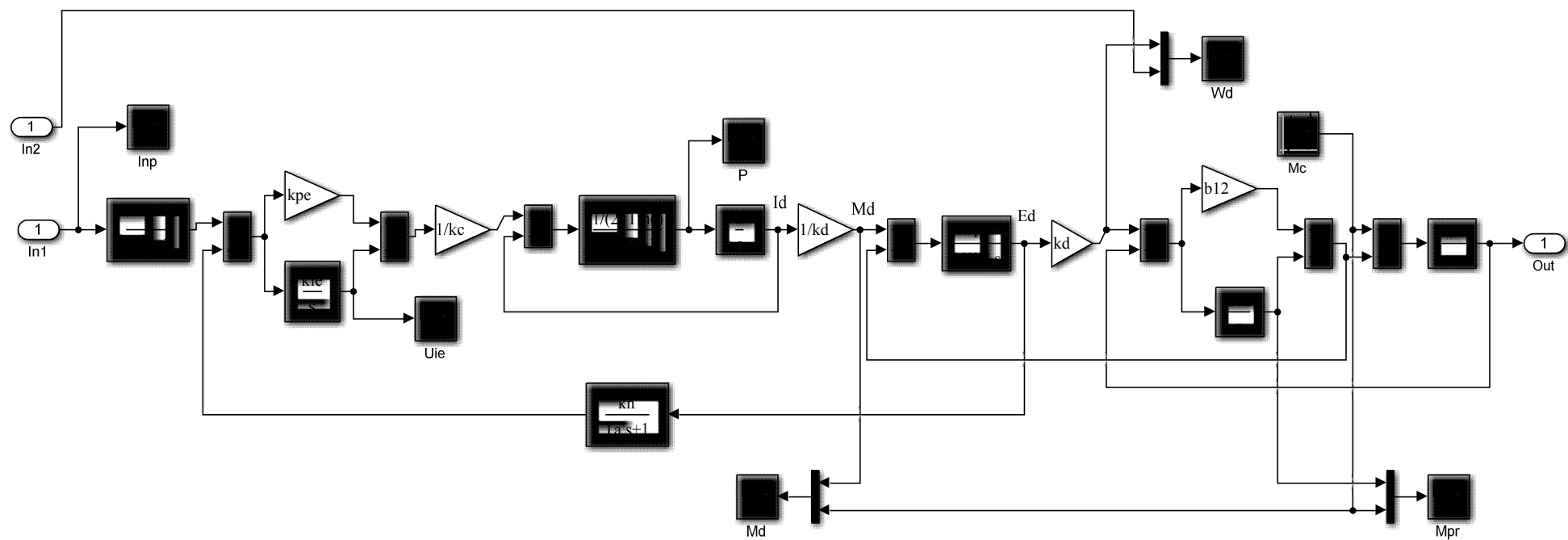


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

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.

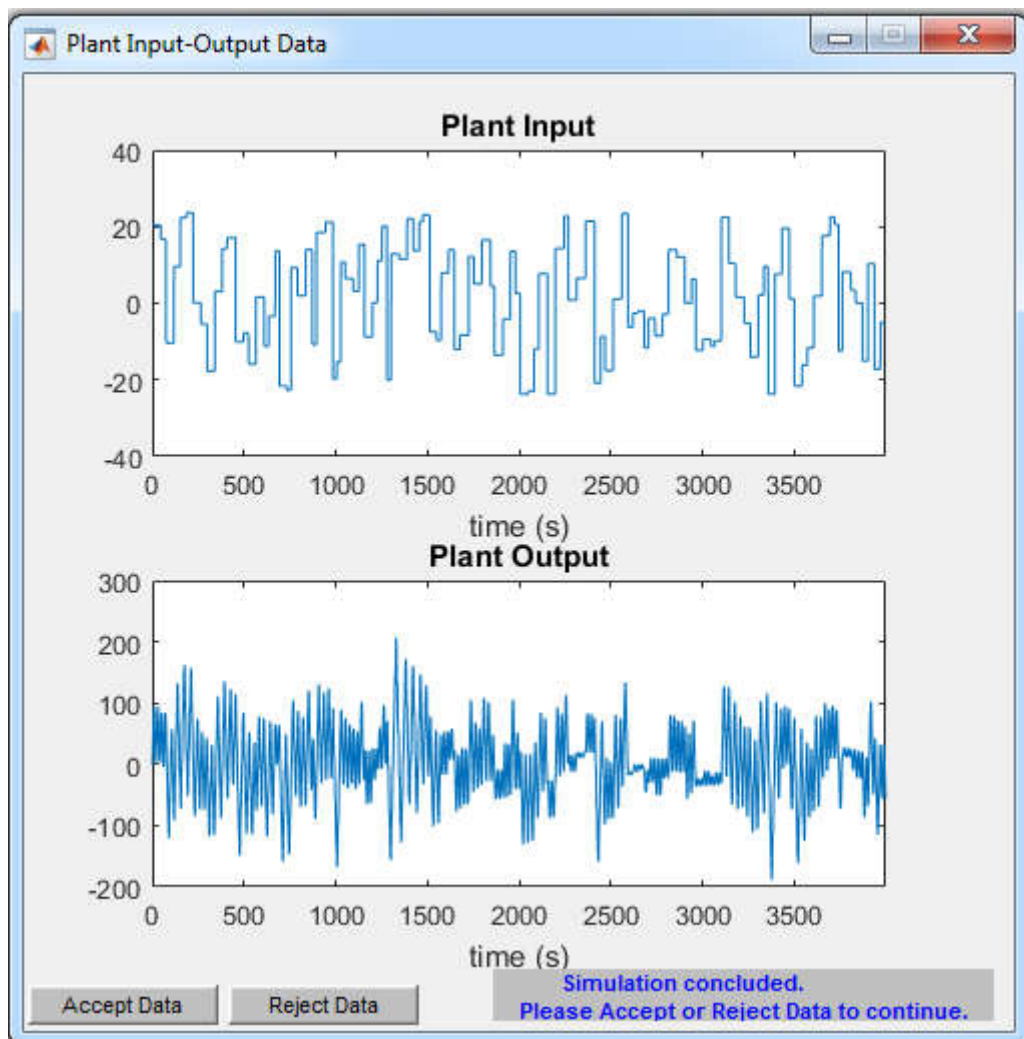


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

Якщо дані прийняті, програма відкриває 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).

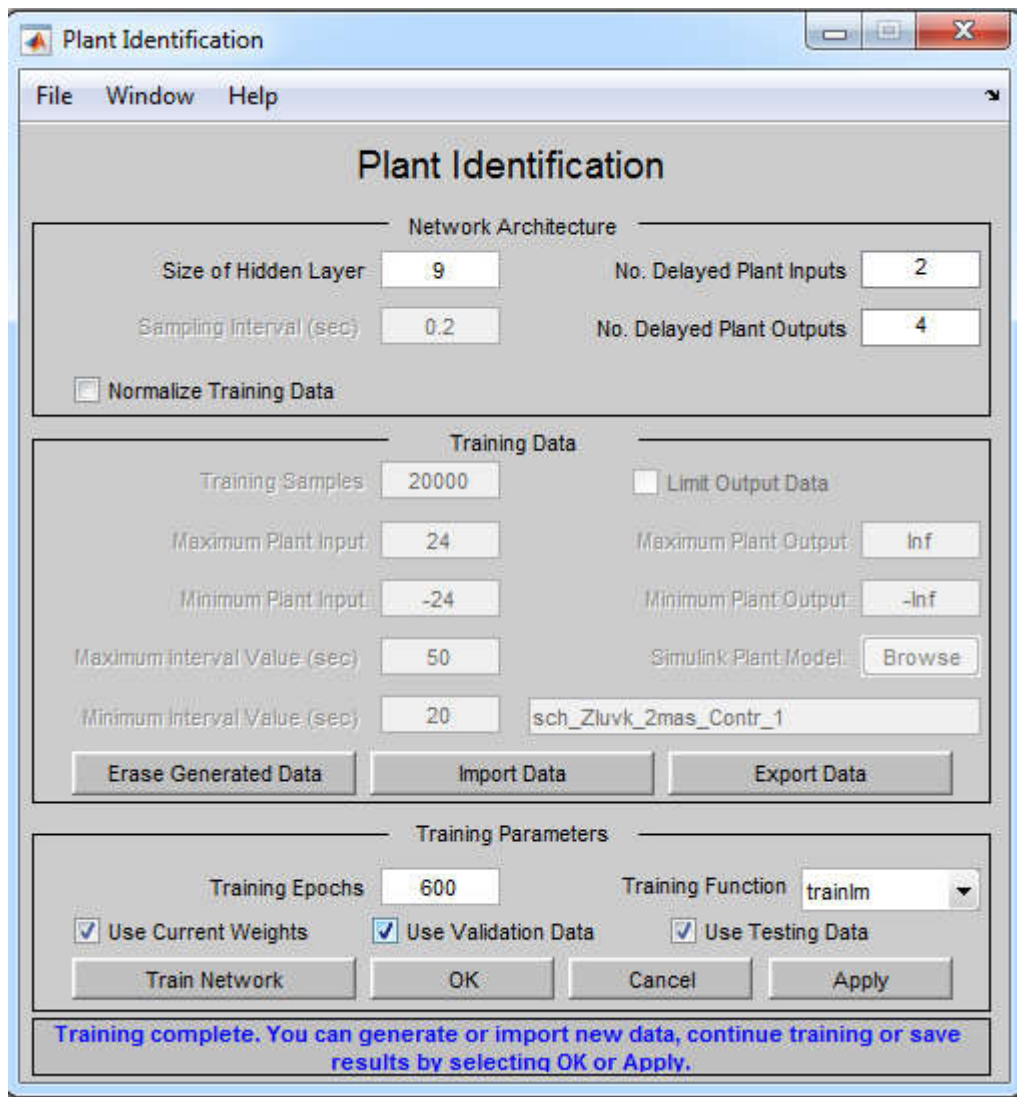


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

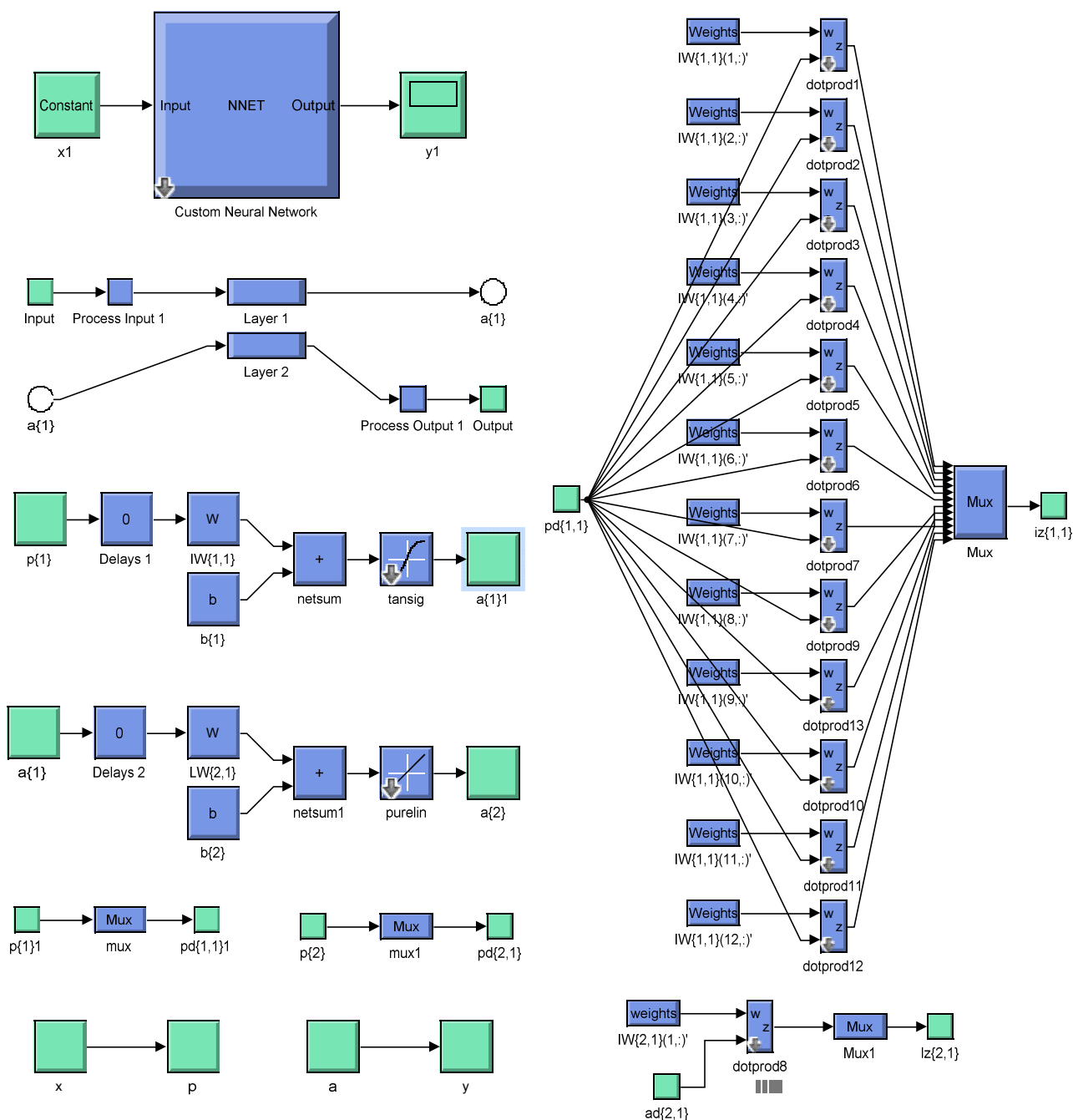


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

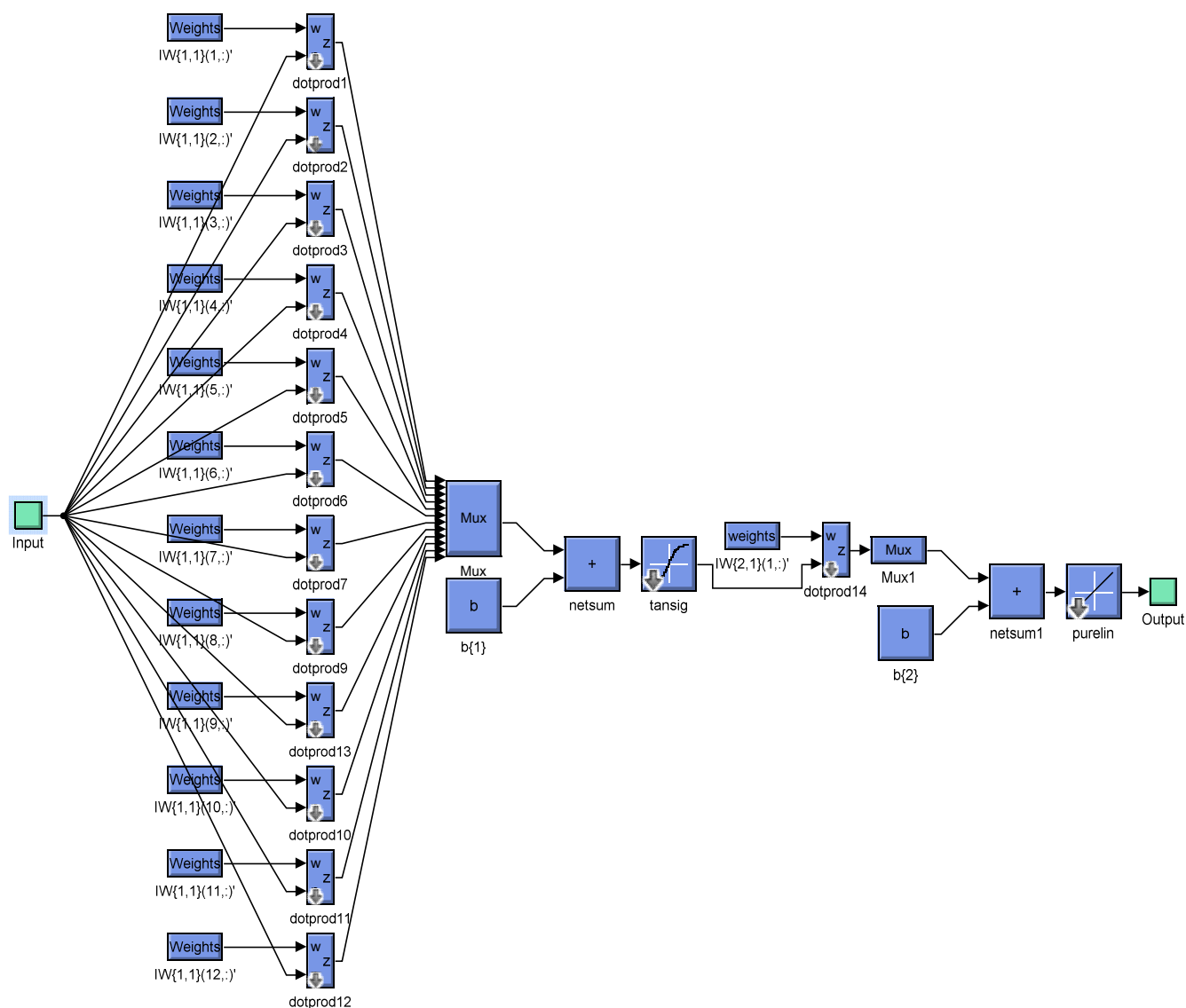


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

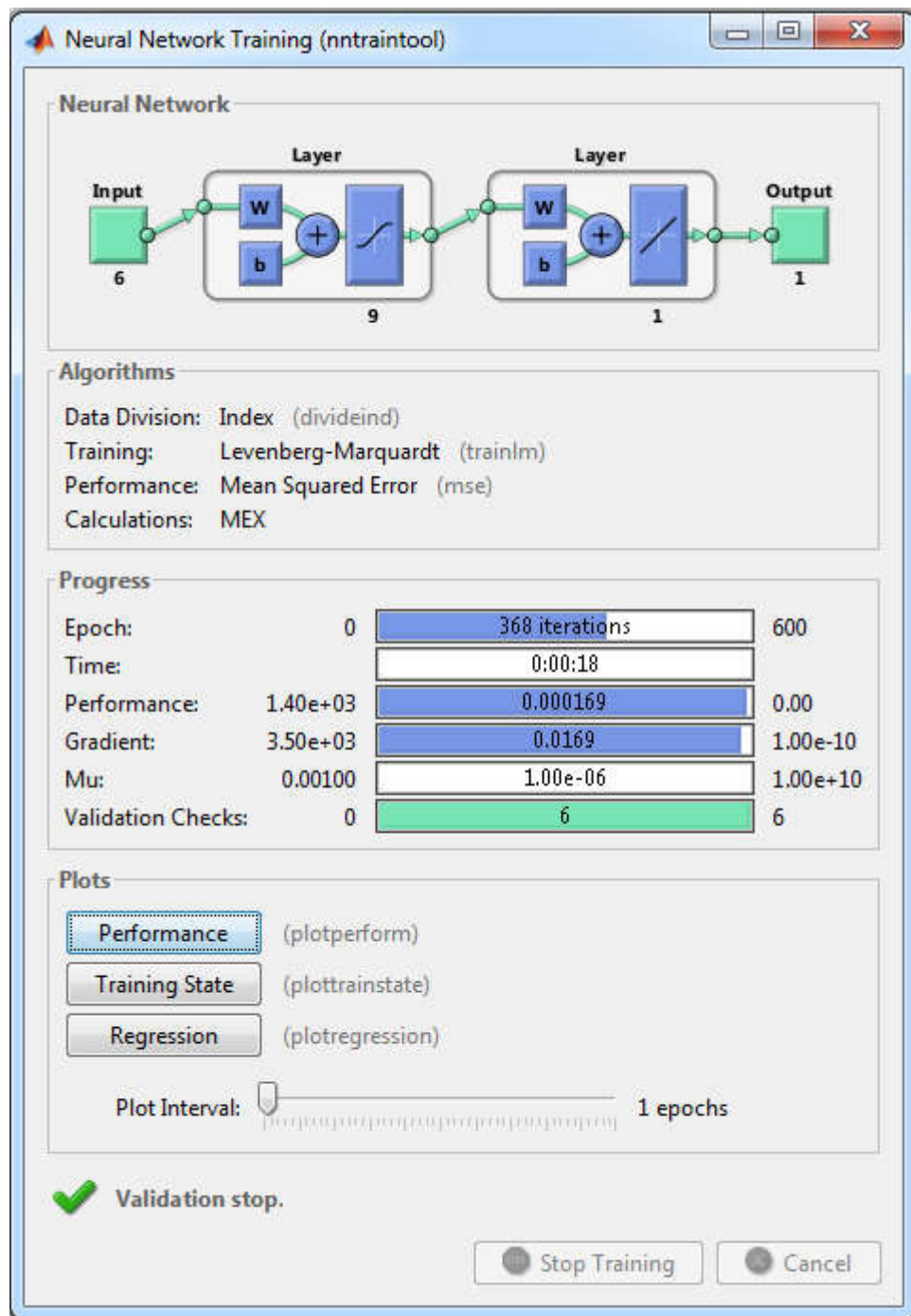


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

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.

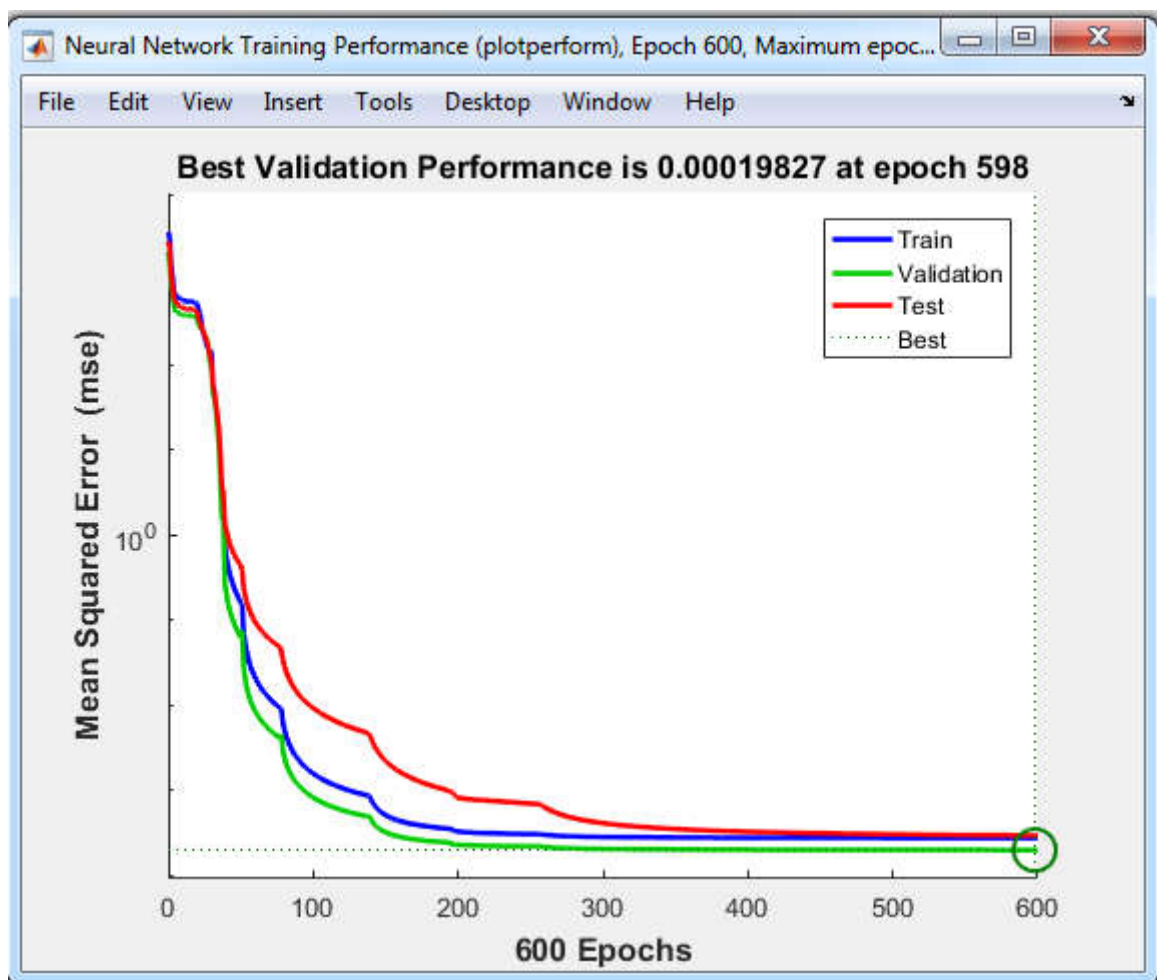


Рисунок 4.15 – Вікно контролю процесу навчання

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 точ-

ність 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.

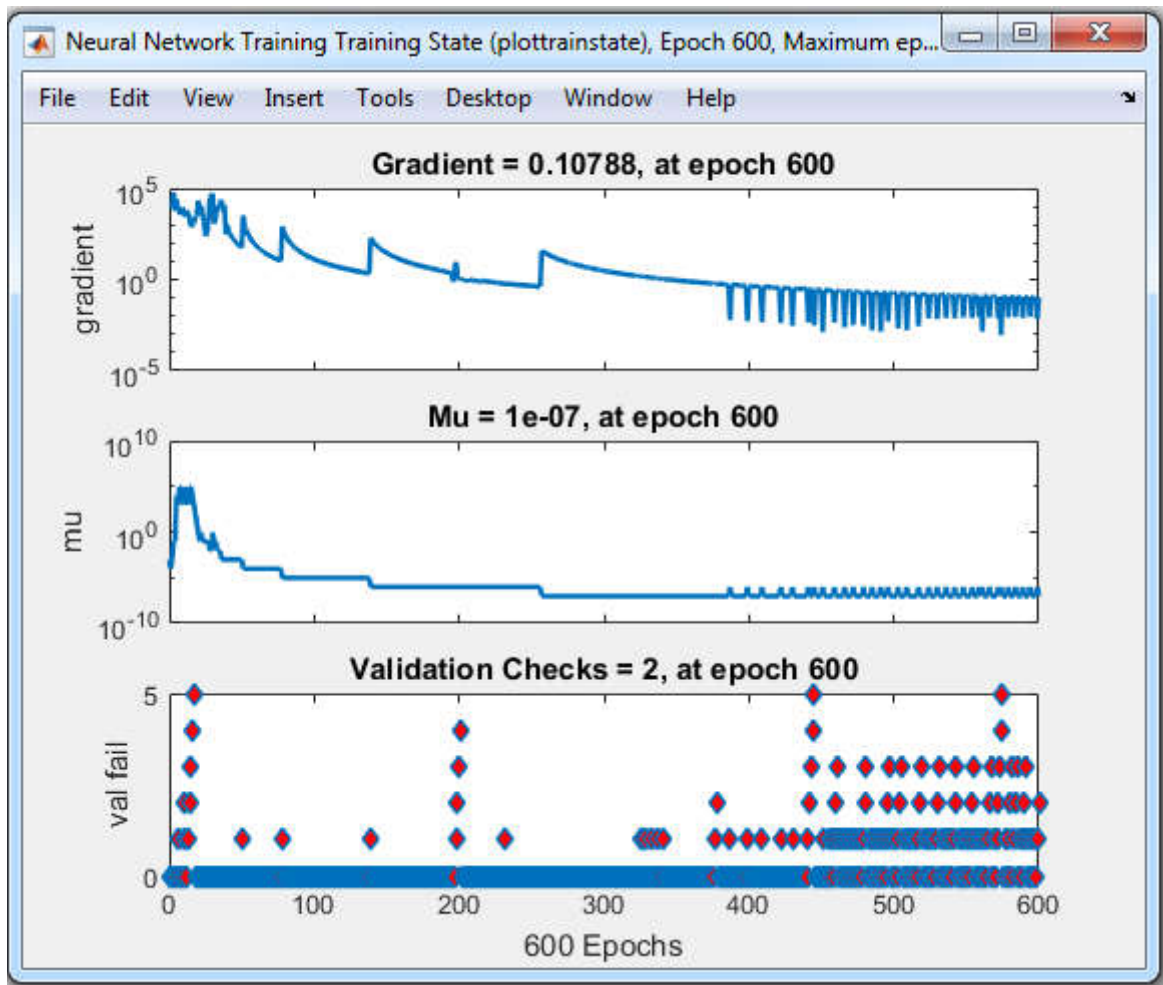


Рисунок 4.16 - Вікно Neural Network Training State

Onuc 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 по-

милки) 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 μ 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.

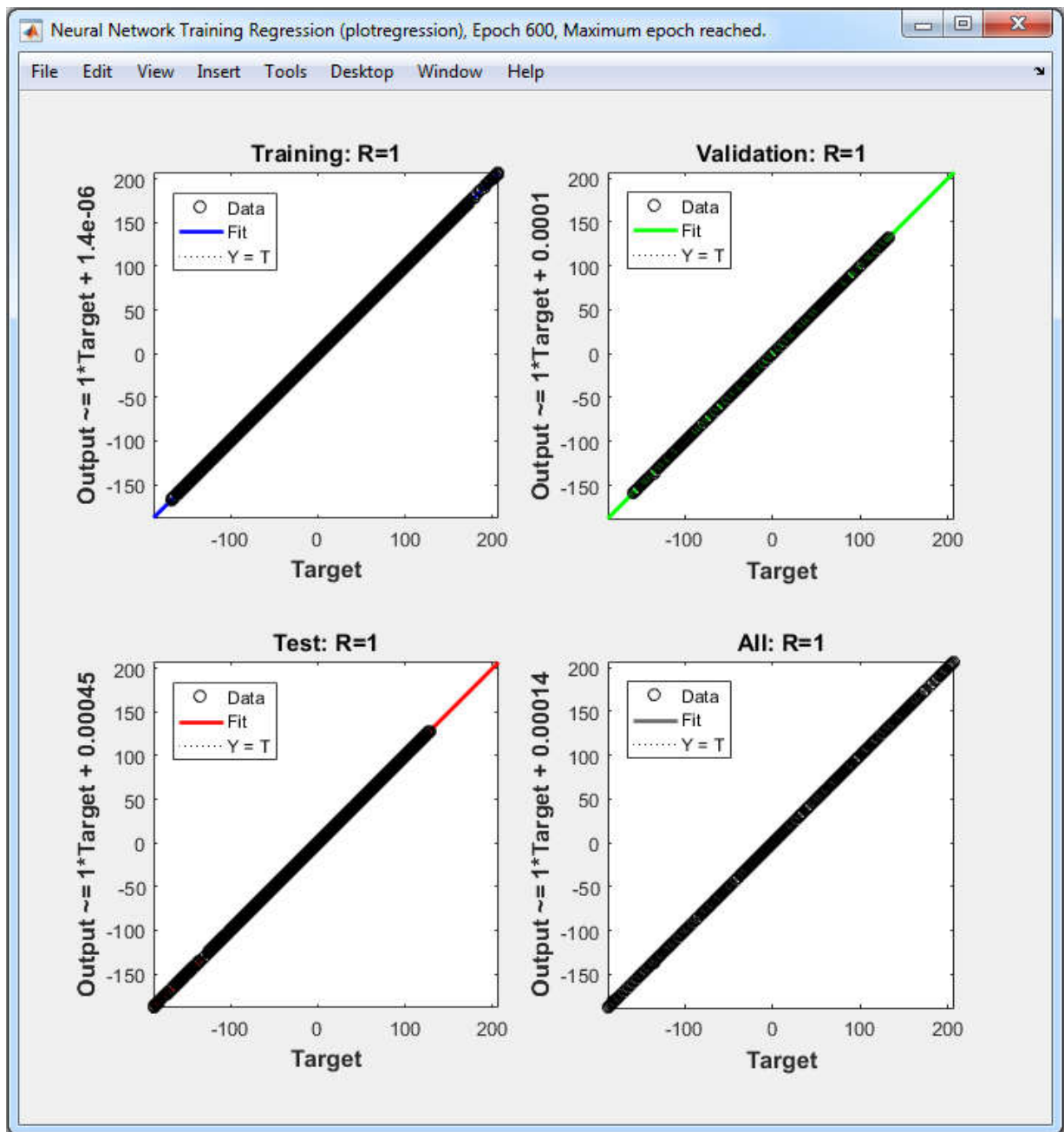


Рисунок 4.17 - Вікно Neural Network Training Regression

Опис вікна Neural Network Training Regression

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

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

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

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

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

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

Графіки **Fit (прогноз)** – це графіки лінійної регресії, які відображають прогнозовані значення, які надає нейронна мережа на основі вхідних даних.

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

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

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

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

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

Кореляційний коефіцієнт R близький до 1 вказує на високу точність моделі регресії, тоді як значення близьке до 0 або від'ємне вказує на низьку

точність. Цей коефіцієнт дуже корисний для візуальної оцінки точності моделі та порівняння прогнозованих та фактичних значень у завданнях регресії.

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

Кнопка **Cancel (скасувати)** використовується для припинення процесу навчання, але зазвичай вона надає можливість підтвердження та збереження поточного стану моделі. Вона дозволяє вам контролювати ітерації навчання,

перевіряти результати, змінювати параметри та вирішувати, коли припинити навчання з можливістю збереження результатів навчання.

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

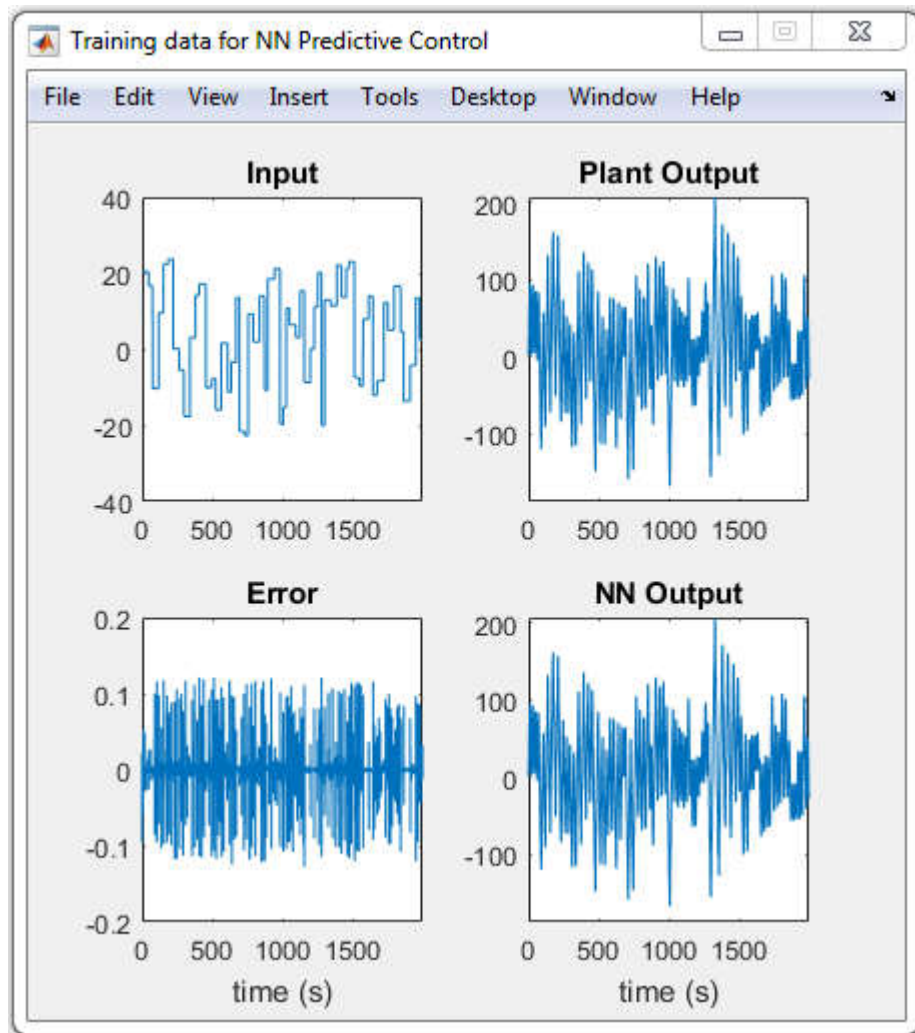


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

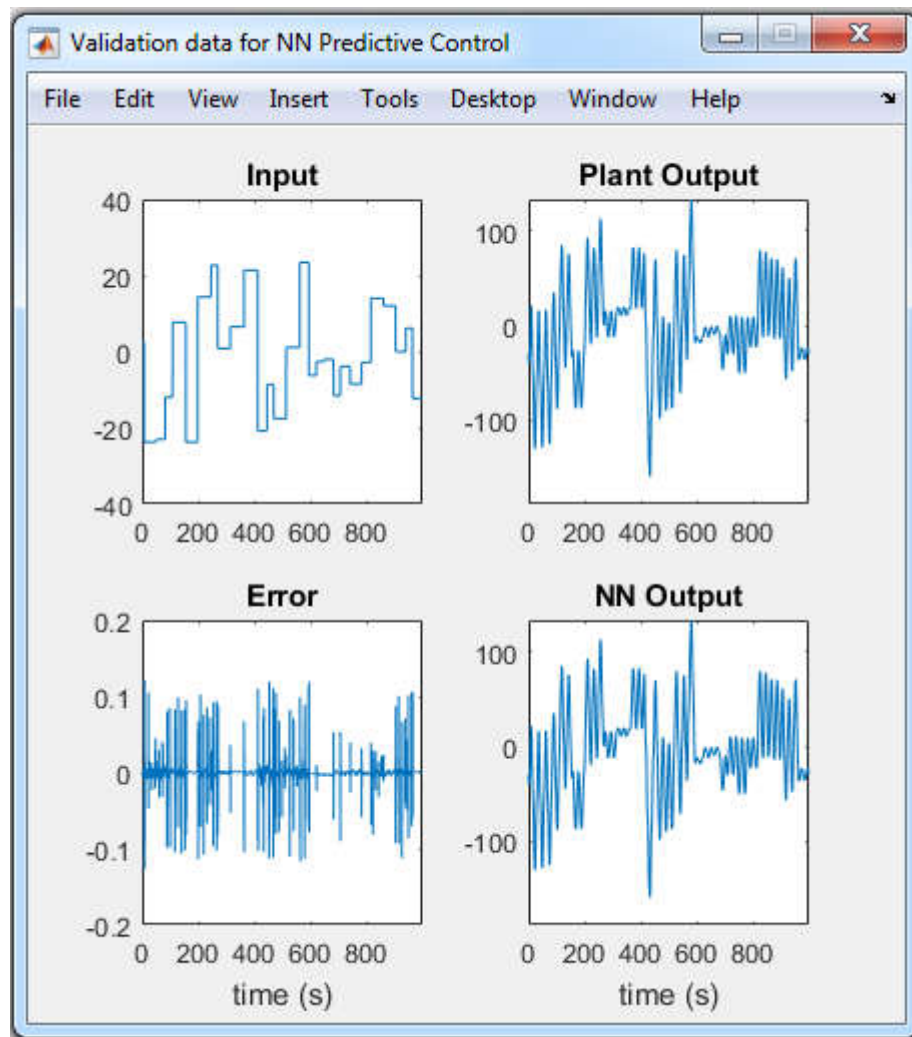


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

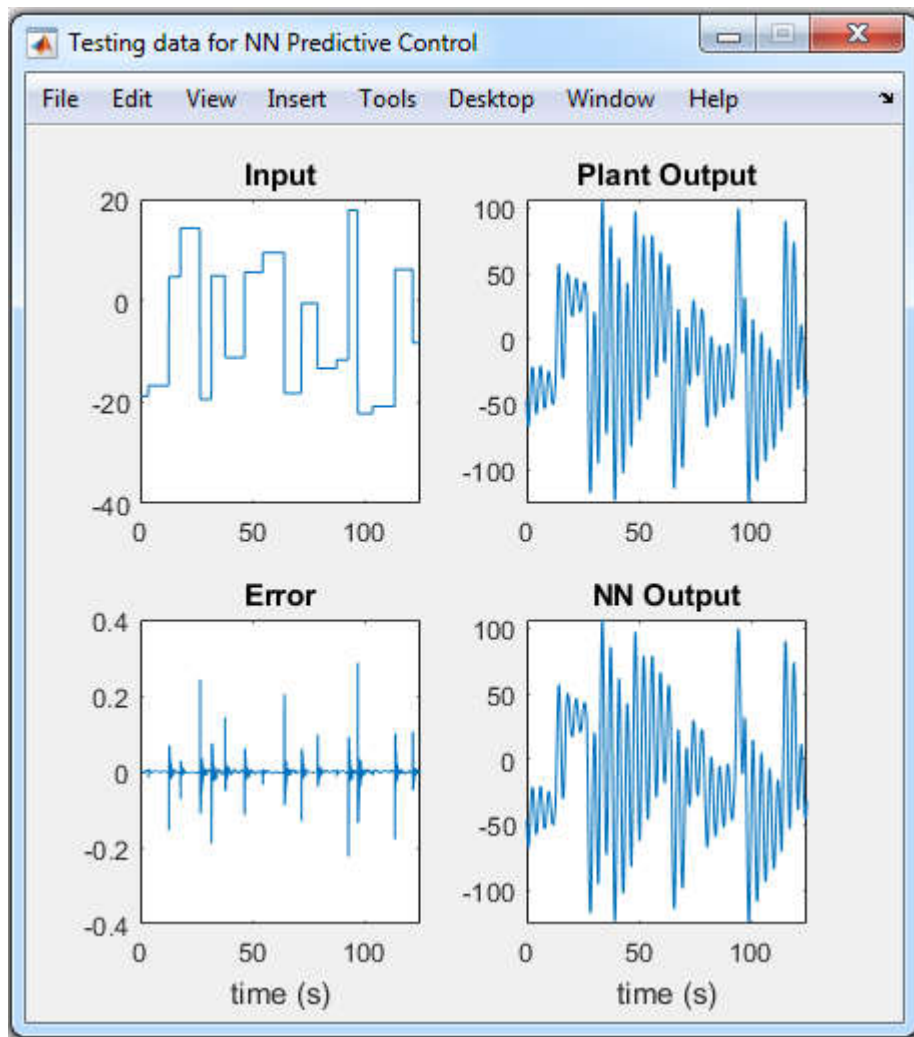


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

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.

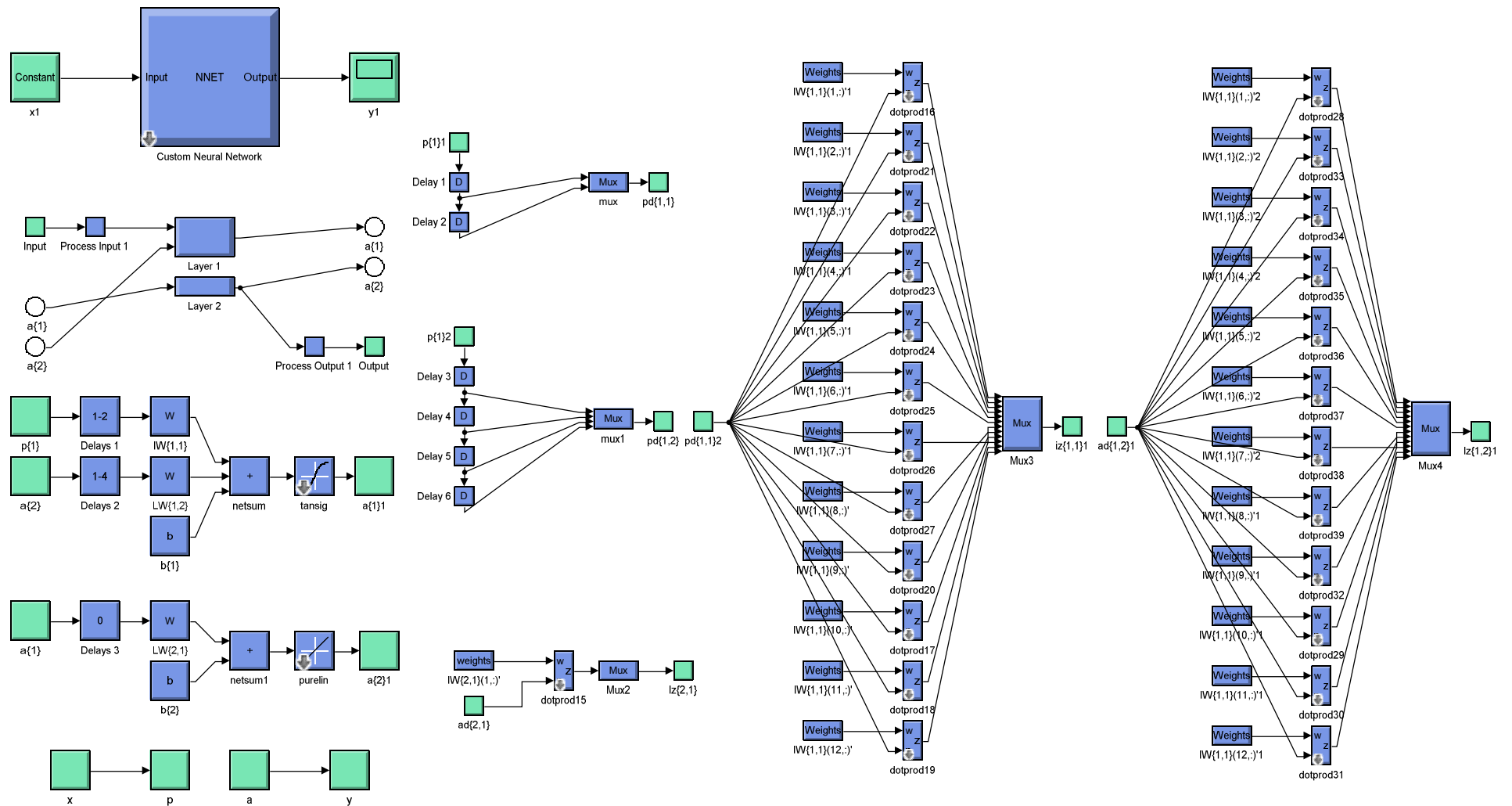


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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.

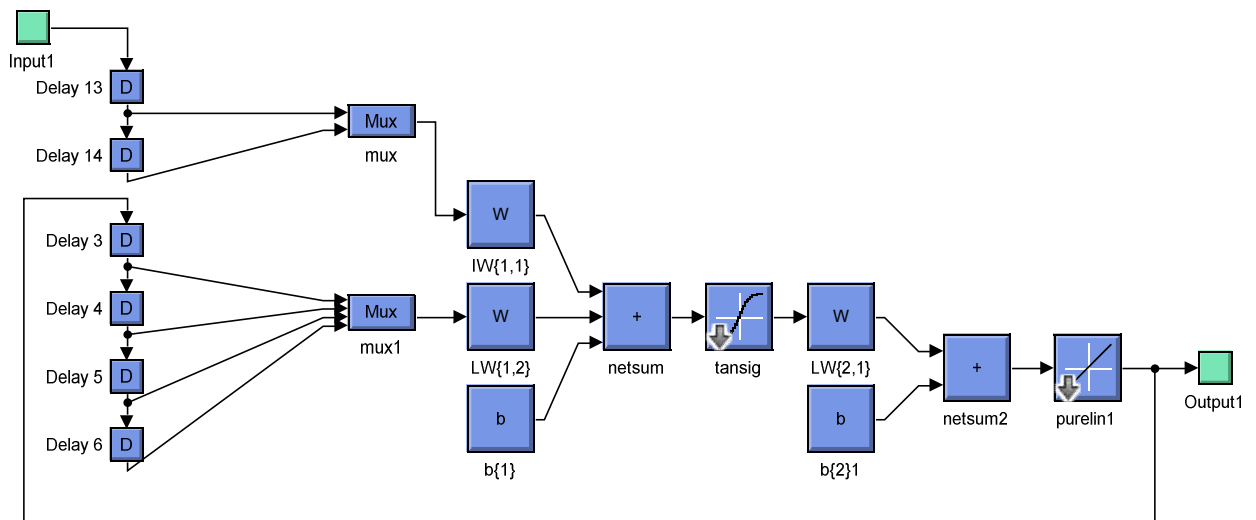


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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.

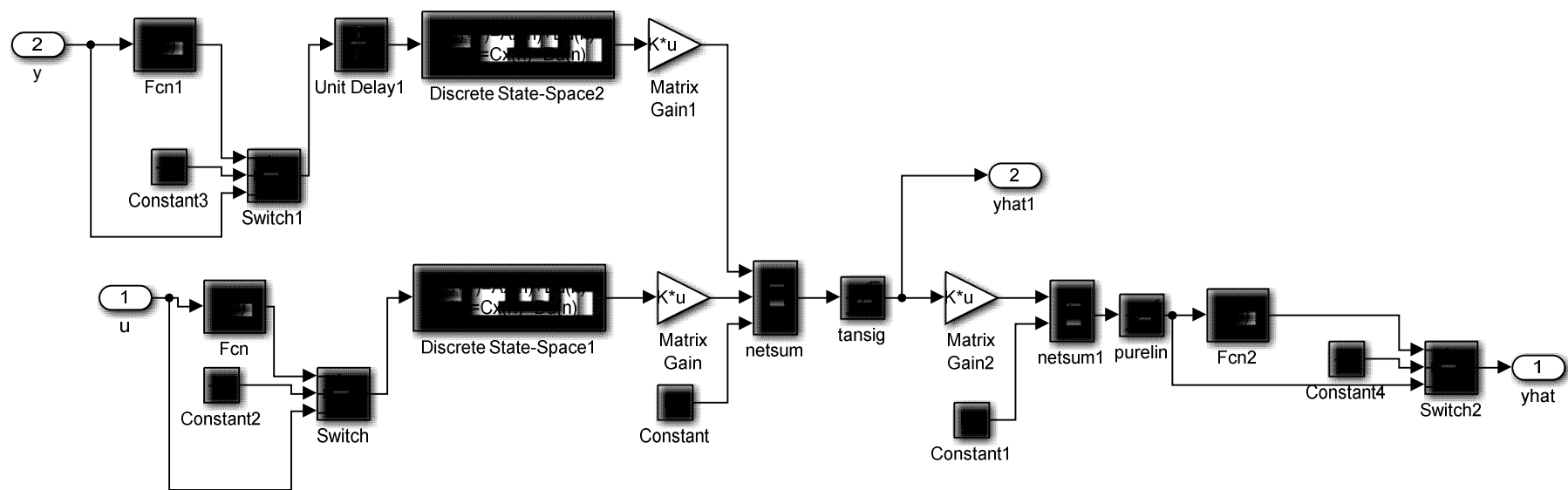


Рисунок 4.23 – Структурна схема нейромережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

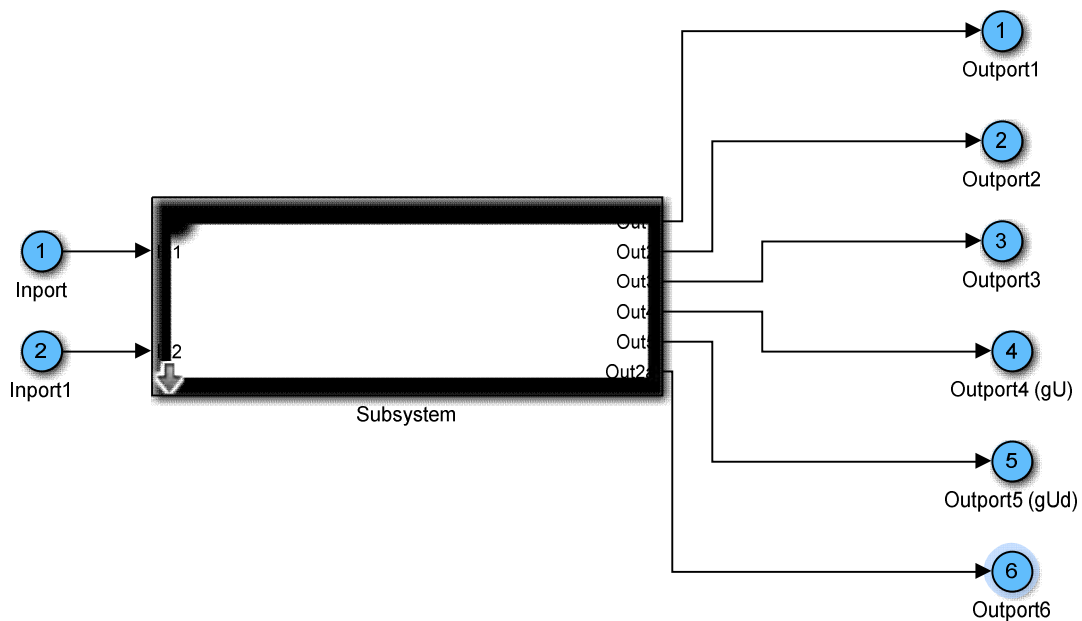


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

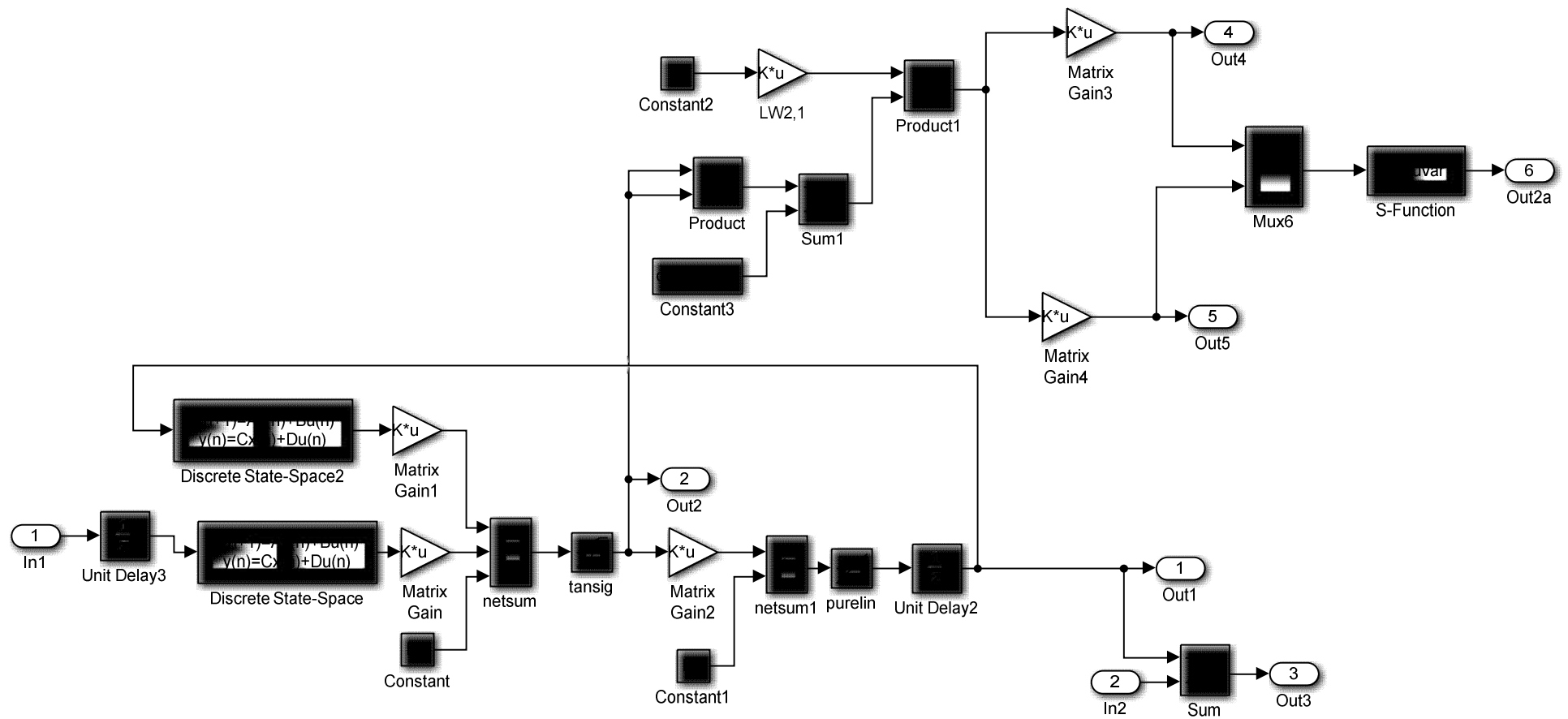


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

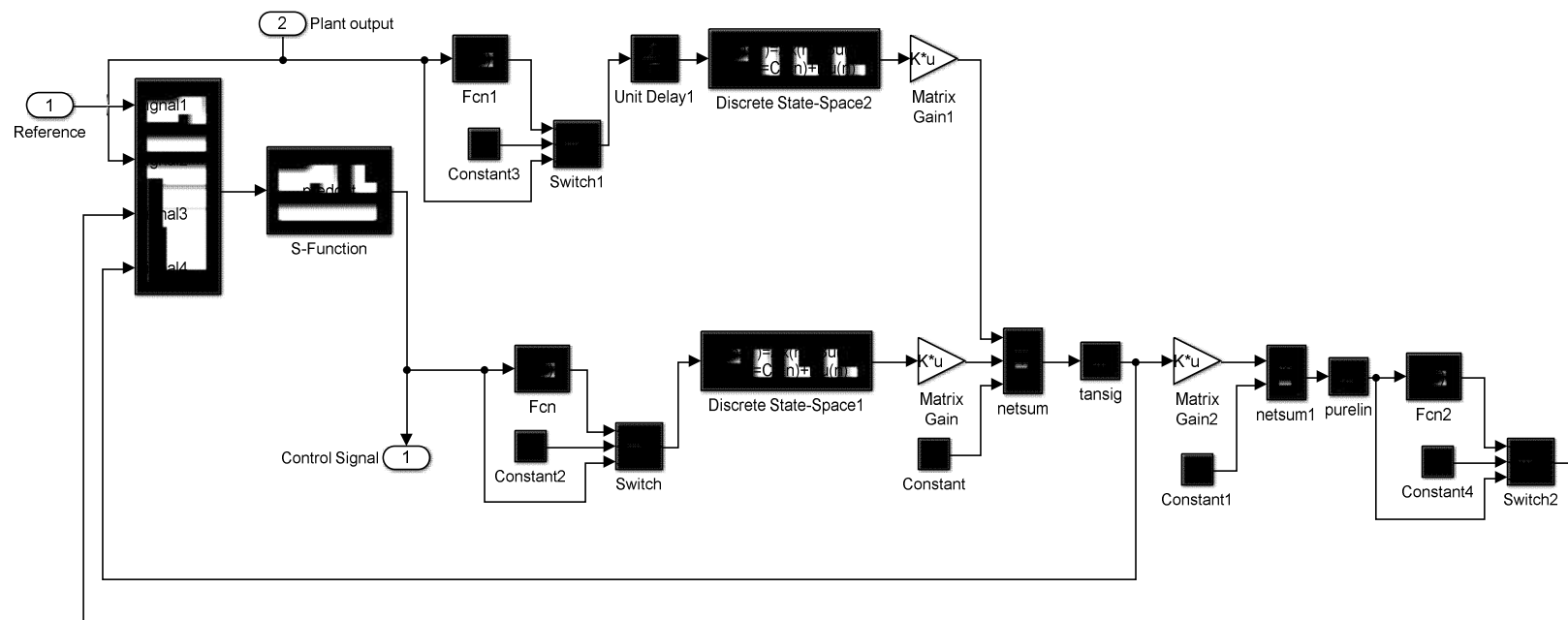


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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.

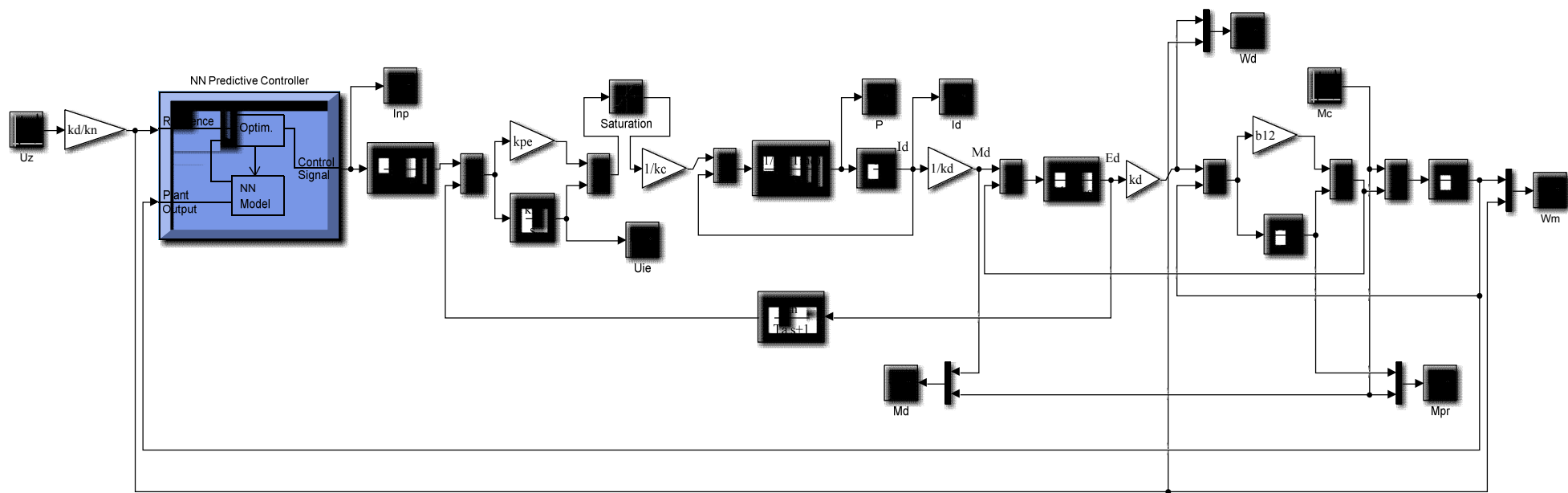
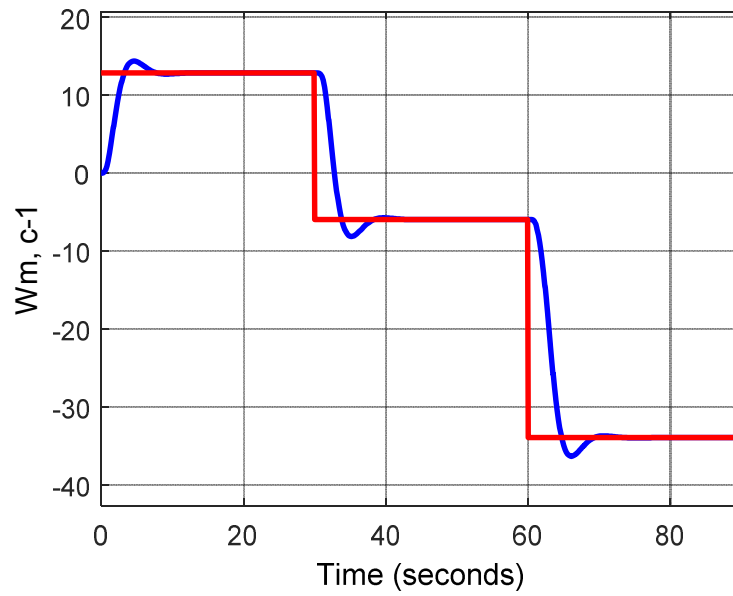
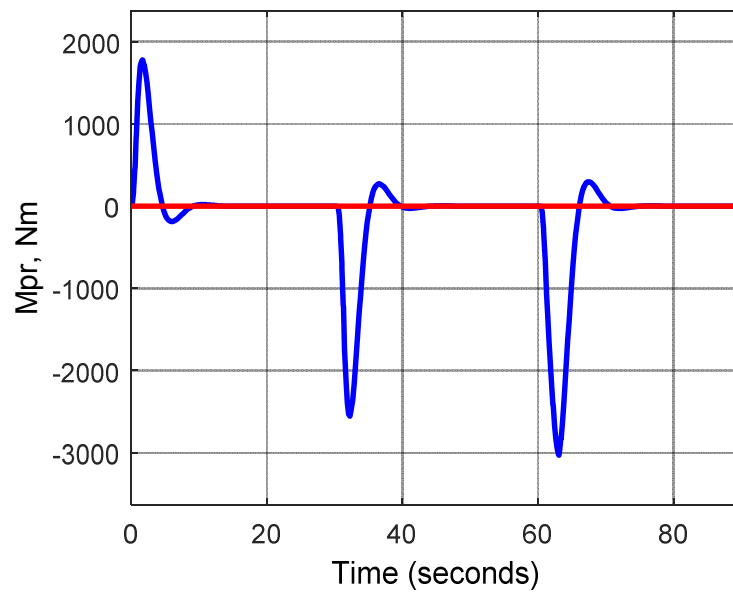


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

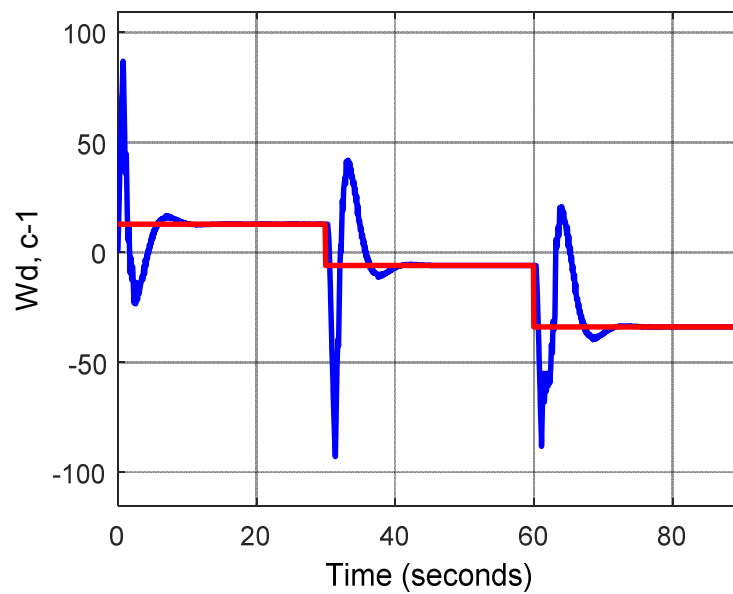


a)

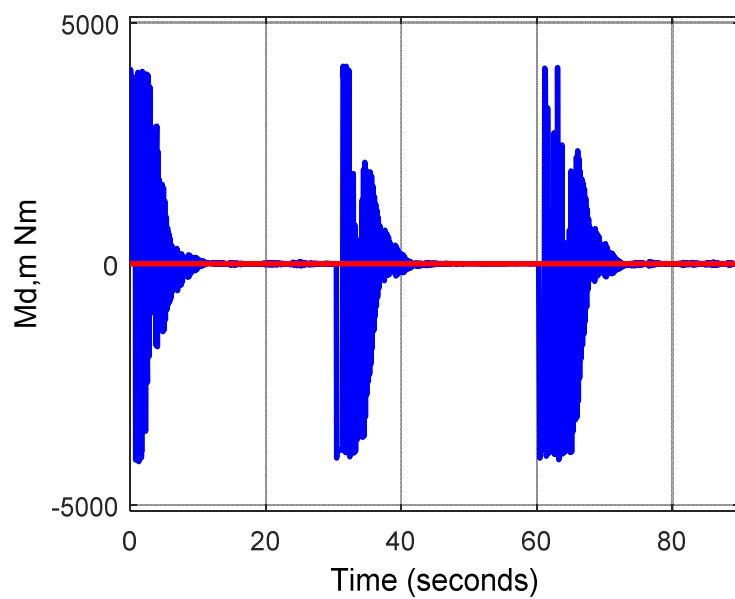


б)

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

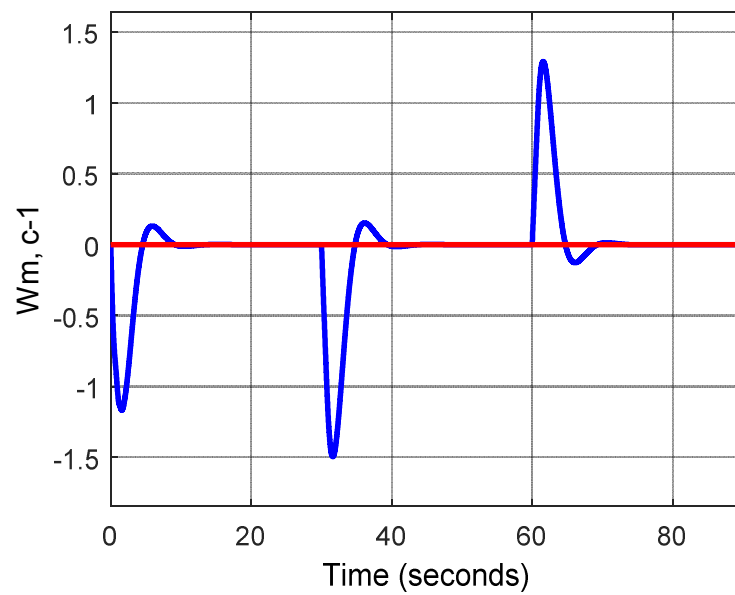


a)

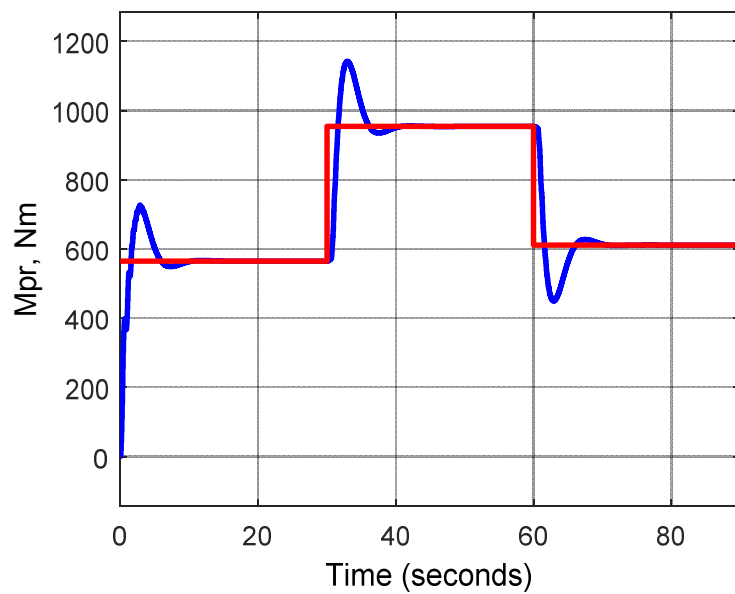


б)

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



a)



б)

Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_1 = f(t)$ (а) і моменту пружності $M_6 = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

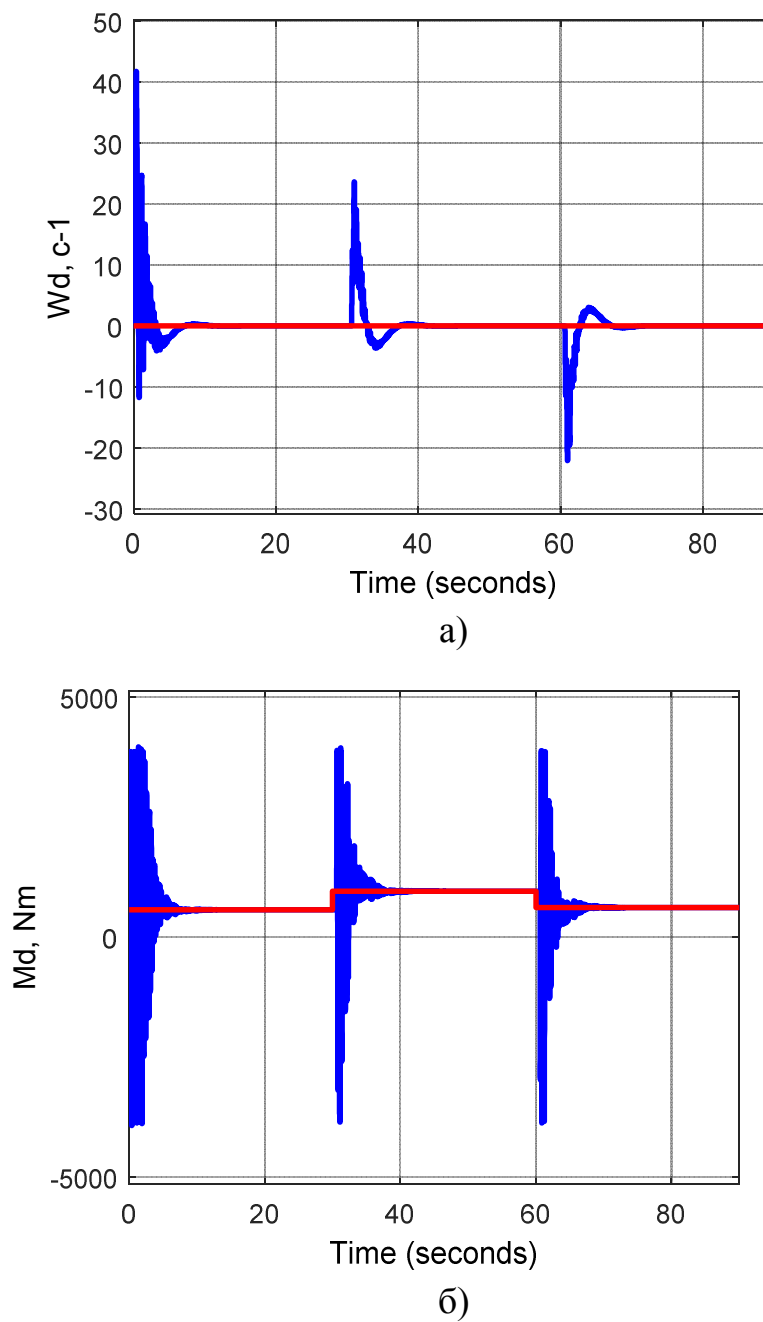


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

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 стабілізації, що розглядається в даній роботі, обраний регулятор з прогнозом NN Predictive Controller. Розглянуто структуру та порядок синтезу даних нейро регуляторів.

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 переміщення візка зливковому кільцевої зливкоподачі стану гарячої прокатки 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 0,2с; мінімальний інтервал ідентифікації 20 с; максимальний інтервал ідентифікації 50с.

4. Створено модель двомасової системи у вигляді двошарової нейронної мережі 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 0,4 .

5. Виконано синтез нейрорегулятора з використанням предиктивного контролера на основі нейронних мереж (NN Predictive Controller). Завдяки варіації параметрів нейрорегулятора в широкому діапазоні вдалося визначити параметри, які суттєво впливають на якість регулювання, та знайти оптимальні значен-

ня цих параметрів. Використання нейромережевої моделі об'єкта управління з високою точністю ідентифікації дозволило створити нейрорегулятор, що забезпечує відмінні показники якості системи.

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних станів у системі з нейрорегулятором NN Predictive Controller під час запуску та при навантаженні мають задовільний характер. Перерегулювання не перевищує 10%, час регулювання складає 8 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 зливковоза кільцевої зливкоподачі стану гарячої прокатки, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління основних і допоміжних механізмів прокатних станів, 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 зливковоза..

2. На підставі аналізу технічних характеристик зливковозів і вимог до електроприводів допоміжних механізмів станів гарячої прокатки вибраний електропривод постійного струму, виконаний за системою ТП-Д. Виконано вибір електродвигуна і основного електроустаткування силового ланцюга електроприводу.

3. Проведений вибір структури і розрахунок системи автоматичного управління параметрами електроприводу. Для управління координатами електроприводу застосована система підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. Виконано розрахунок параметрів силового ланцюга електроприводу, що використовуються при розрахунку системи автоматичного управління. Проведений синтез послідовних коректуючи пристроїв контурів, з оптимізацією контура струму по модульному критерію, а контура ЕРС – по симетричному критерію. Проведено моделювання системи на ЕОМ з використанням пакету прокладних програм MATLAB.

4. Розроблена математична модель двомасової системи, що враховує пружність механічних зв'язків, і після моделювання на ЕОМ виявлено, що ос-

новні координати у двомасовій системі проявляють значні коливання під час перехідних процесів.

5. Для додання системі бажаних динамічних характеристик в роботі виконаний синтез нейромережевої 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 10%, час регулювання складає 8 с, що відповідає завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Технології нейронних мереж і нечіткого моделювання в системах управління: підруч. для здобувачів вищої освіти спец. 151 Автоматизація та комп'ютерно-інтегровані технології / Г.І. Канюк, Б.І. Кузнецов, Т.Ю. Василець, А.Ю. Мезеря, О.О. Варфоломійєв. – Харків : Друкарня Мадрид, 2020. – 306 с.
2. Нейромережеві технології в системах управління: Підручник для візів./ Б. І. Кузнецов, Т.Ю. Василець, Т.Б. Нікітіна, В. В. Коломиєць, О.О. Варфоломійєв; Укр. інж.-пед. акад.. - Харків: УПА, 2014. - 232 с.
3. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник.. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 224 с.
4. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. – 353 с.
5. Штучні нейронні мережі: навчальний посібник для студентів вищих навчальних закладів / О.Г.Руденко, Є.В. Бодянський. – К: Компанія СМІТ, 2006, 404 с.
6. Субботін С. О. Нейронні мережі : теорія та практика: навч. посіб. / С. О. Субботін. – Житомир : Вид. О. О. Євенок, 2020. – 184 с.
7. Тимощук П.В. Штучні нейронні мережі; Навч. посібн. - Львів: Львівська політехніка, 2011. - 444 с.
8. Кононюк, А. Ю. .Нейронні мережі і генетичні алгоритми. - Київ: Корнійчук, 2008. - 468 с.
9. Кирик В. В. Математичний апарат штучного інтелекту в електроенергетичних системах: підручник.. – Київ: КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2019. – 224 с.
10. Зайченко Ю.П. Основи проектування інтелектуальних систем.. – К.: Слово, 2004. – 353 с.
11. Осовский С. Нейронные сети для обработки информации. – М.: Финансы и статистика, 2002. – 344с.
12. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления. – М.: ИПРЖР, 2002. – 480 с.

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

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

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

52. Haykin S , Neural Networks A comprehensive foundation, MacMillan College Publishing Co , New York, 1994.
53. Heermann P D Neural network techniques for stable learning control of nonlinear systems Dissertation D S University of Texas at Austin, 1992.
54. Narendra K S , Parthasarathy K Neural networks and dynamical systems Part 1 A gradient approach to Hopfield networks TR 8820, Center for systems science, Department of Electrical Engineering, Yale University, New Haven, CT, 1988.
55. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
56. Werbos P J , Beyond Regression New Tools for Prediction and Analysis in the Behavioral Sciences, Phd Thesis, Dept of Applied Mathematics, Harvard University, Cambridge, Mass , 1974.
57. Garcia C.E. Model predictive control: theory and practice a survey / Garcia C.E., Prett D.M., Morari M. // Automatica. – 1989. – Vol.25. – P.335-348.
58. Neural networks for control / Miller W T , Sutton R S , Werbos P J , Eds , The MIT Press, 1990.
59. Li Y., Sundararajan N., Saratchandran P. (2001), Neuro-controller design for nonlinear fighter aircraft maneuver using fully tuned RBF networks, Automatica, Vol. 37, No 8, pp. 1293-1301.
60. Gundy-Burlet K., Krishnakumar K., Limes G., Bryant D. (2004), Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft, J. of Aerospace Computing, Information, and Communication, Vol. 1, No.12, pp. 526-542.
61. Kondratev A. I., Tyumentsev Yu. V. (2010), “Neyrosetevoe adaptivnoe otkazoustoychivoe upravlenie dvizheniem manevrennogo samoleta”, [Neural network adaptive fault-tolerant control of the movement of a maneuverable aircraft], XII All-Russian Scientific and Technical Conference "Neuroinformatics – 2010", Part 2, NIYaU MIFI, Moscow, pp. 262-273.
62. Nikiforova L. N., Petrosyan E. A., Yakemenko G. V. (2000), “Neyrokompyuteryi v upravlenii vertoletami”, [Neurocomputers in helicopter control], Artificial Intelligence, No. 3, pp. 290-298.

63. Kupin A. I. (2008), "Intelektualna Identifikatsiya ta keruvannya v umovah protsesiv zbagachuvalnoyi tehnologiyi", [Intelligent identification and control in the processes of enrichment technology], KTU, Krivoy Rog, 204 p.
64. D. Gu and H. Hu. (2002), Neural Predictive Control for a Car-like Mobile Robot, International Journal of Robotics and Autonomous Systems, Vol. 39 (2), pp. 73-86.
65. Danil V. Prokhorov. (2008), Toyota Prius HEV Neurocontrol and Diagnostics, Neural Networks, No. 21, pp. 458-465.
66. Zmeu K. V., Markov N. A., Shipitko I. A., Notkin B. S. (2009), "Bezmodelnoe prognoziruyushee inversnoe neyro-upravlenie s regeneriruemym etalonnym perehodnym protsessom", [A modelless predictive inverse neuropravlenie with a regenerated reference transient process], Intelligent Systems, No. 3, pp. 109-117.
67. Kuznetsov B. I. , Vasilets T. Yu., Varfolomiev O. O. (2015), "Neyromerezheva sistema navedennya i stabilizatsiyi z regulyatorom na osnovi etalonnoyi modeli", [Neural network aiming and stabilization system with the reference model controller], Electrical engineering and electromechanics, No. 4, pp.35-39.
68. Dzyuba D. A., Chernodub A. N. (2010), "Primenenie metoda kontroliruemogo vozmuscheniya dlya modifikatsii neyrokontrollerov v realnom vremeni", [The application of the controlled disturbance method for the modification of real time neural controllers], Mathematical machines and systems, No. 4, pp. 20-28.
69. Dias F.M., Mota A.M. (2001), Comparison between Different Control Strategies using Neural Networks, 9th Mediterranean Conference on Control and Automation, Croatia, Dubrovnik.
70. Venayagamoorthy G.K., Harley R.G., Wunsch D.C. (2003), Implementation of Adaptive Criticbased Neurocontrollers for Turbogenerators in a Multimachine Power System", IEEE Transactions on Neural Networks, Vol. 14, Issue 5, pp. 1047–1064.
71. D’Emilia G., Marrab A., Natalea E. (2007), Use of neural networks for quick and accurate autotuning of PID controller, Robotics and Computer-Integrated Manufacturing, Vol. 23, pp. 170 – 179.

72. Бычков В. П. Электропривод и автоматизация металлургического производства. Учеб. Пособие для вузов. Изд. 2-е. перераб. и доп. – М.: Высшая школа, 1977. – 391 с.
73. Тиристорные электроприводы прокатных станов. Перельмутер В. М. и др. – М., Металлургия, 1978. – 152 с.
74. Бешта О.С., Балахонцев О.В., Бородай В.А. Автоматизований електропривод у прокатному виробництві. – Національний гірничий університет, Дніпропетровськ, 2010. – 224 с.
75. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
76. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
77. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
78. 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.
79. 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.
80. 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.
81. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
82. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.

83. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
84. 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
85. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч1.
86. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інш. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інш. – К.: Либідь, 2005. – 680 с. Ч2.
87. 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.
- 88.Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоєнко. Харків, ХНАДУ, 2012. – 284 с.
89. 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.
90. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
91. 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.
92. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

ДОДАТОК А

МОДЕЛІ НЕЙРОНА

А.1 Структурна схема нейрона

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

Елементарним осередком нейронної мережі є нейрон. Структура нейрона з єдиним скалярним входом показана на рис. А.1, а

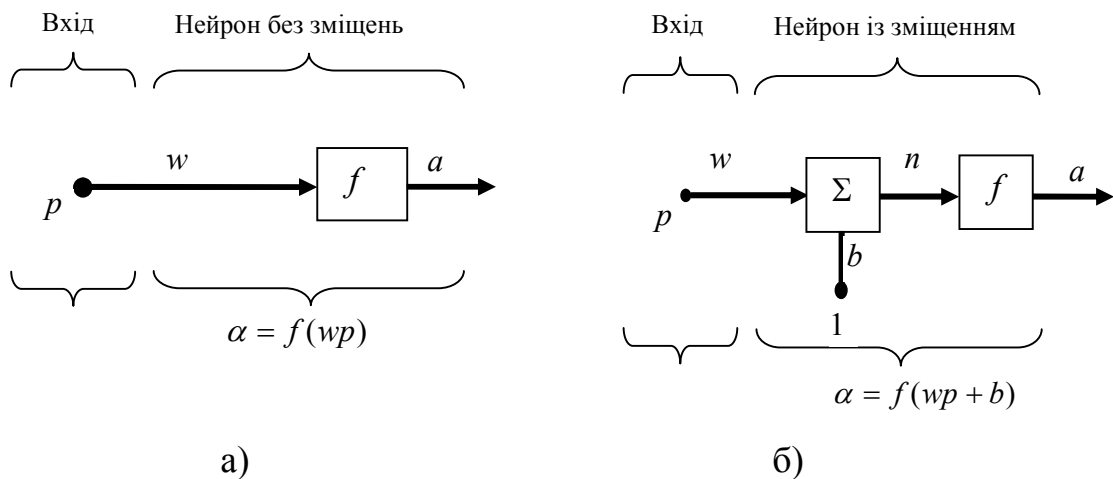


Рисунок А.1 – Структура простого нейрона:

а) нейрон з одним скалярним входом; б) нейрон з скалярним зсувом

Скалярний вхідний сигнал p множиться на скалярний ваговий коефіцієнт w і результуючий зважений вхід $w \cdot p$ є аргументом функції активації нейрона, яка породжує скалярний вихід a .

Нейрон, показаний на рис. А.1 б, доповнений скалярним зсувом b . Зсув підсумовується із зваженим входом $w \cdot p$ і приводить до зрушення аргументу функції f на величину b . Дію зсуву можна звести до схеми зважування, якщо уявити, що нейрон має другий вхідний сигнал із значенням, рівним 1. Вхід n функції активації нейрона як і раніше залишається скалярним і рівним сумі зваженого входу і зсуву b . Ця сума є аргументом функції активації f ; виходом

функції активації є сигнал a . Константи w і b є скалярними параметрами нейрона. Основний принцип роботи нейронної мережі полягає в настройці параметрів нейрона так, щоб поведінка мережі відповідала деякій бажаній поведінці. Регулюючи ваги або параметри зсуву, можна навчити мережу виконувати конкретну роботу; можливо також, що мережа сама коректуватиме свої параметри, щоб досягти необхідного результату.

Рівняння нейрона із зсувом має вигляд

$$a = f(w \cdot p + b \cdot 1). \quad (\text{A.1})$$

Як вже наголошувалося, зсув b – скалярний параметр нейрона, який не є входом, що настроюється, а константа 1, яка управляє зсувом, розглядається, як вхід і може бути врахована у вигляді лінійної комбінації векторів входу

$$a = [w \ p] \begin{bmatrix} b \\ 1 \end{bmatrix}. \quad (\text{A.2})$$

A.2 Функції активації

Функції активації (передавальні функції) нейрона можуть мати самий різний вигляд. Функція активації f , як правило, належить до класу сигмоїдальних функцій (сигмоїдальною (S-подібною) функцією називається безперервна функція, що має дві горизонтальні асимптоти і одну точку перегину) з аргументом n і виходом a .

Розглянемо три найбільш поширені форми функції активації. Ці функції входять до складу пакету прикладних програм (ППП) Neural Network Toolbox системи MATLAB у вигляді М-функція `hardlim`, `purelin` і `logsig`.

Одинична функція активації з жорстким обмеженням hardlim . Ця функція описується співвідношенням $a = \text{hardlim}(n) = 1(n)$ і показана на рис. А.2. Вона рівна 0, якщо $n < 0$, і 1, якщо $n \geq 0$.

Лінійна функція активації purelin . Ця функція описується співвідношенням $a = \text{purelin}(n) = n$ і показана на рис. А.3.

Логістична функція активації logsig . Ця функція описується співвідношенням

$$a = \text{logsig}(n) = 1 / (1 + \exp(-n))$$
 і

показана на рис. А.4. Вона належить до класу сигмоїдальних функцій, і її аргумент може приймати будь-яке значення в діапазоні від $-\infty$ до $+\infty$ а вихід змінюється в діапазоні від 0 до 1. Завдяки властивості диференцируемости ця функція часто використовується в мережах з навчанням на основі методу зворотного розповсюдження помилки.

У багатошарових мережах часто застосовуються нелінійні сигмоїдальні функції активації типу гіперболічного тангенса (рис. А.5).

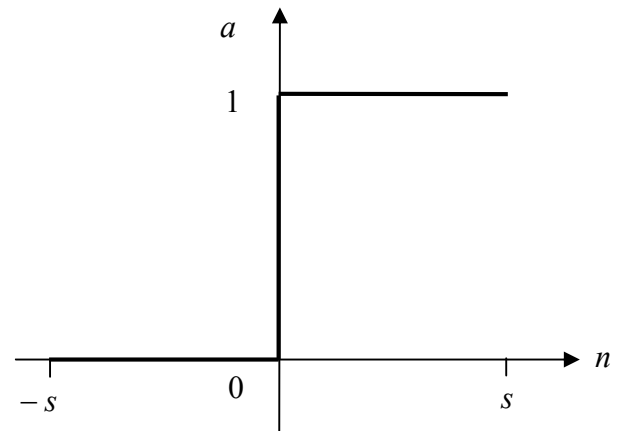


Рисунок А.2 – Одинична
Функція активації

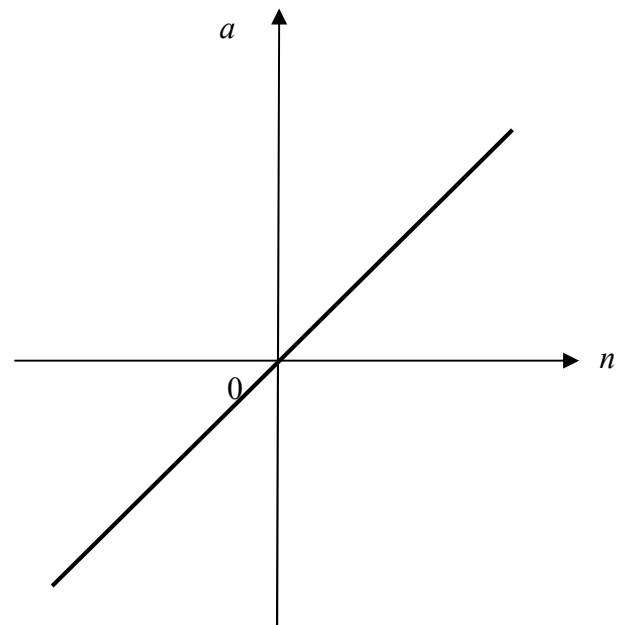


Рисунок А.3 – Лінійна
Функція активації

У ППП Neural Network Toolbox включені і інші функції активації. Використовуючи мову MATLAB, користувач може створювати і свої власні унікальні функції.

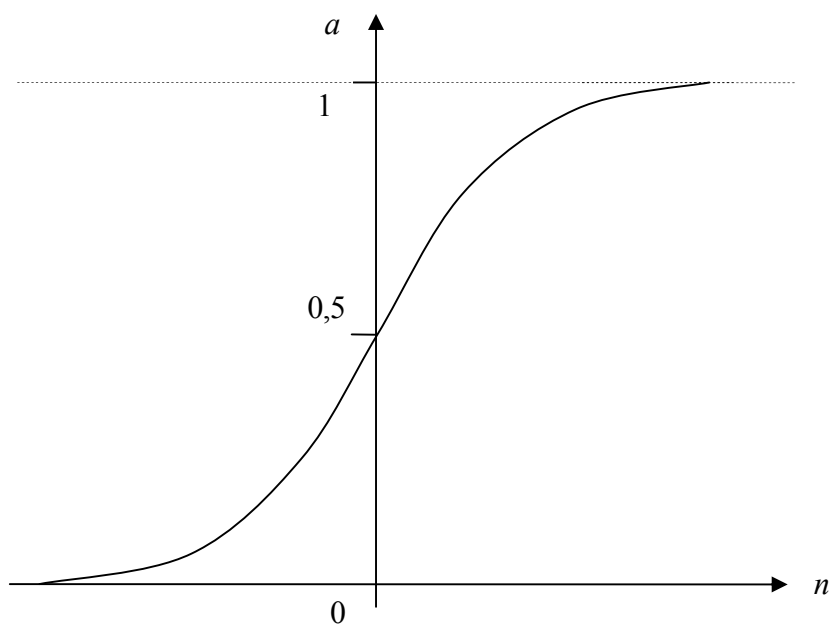


Рисунок А.4 – Логістична функція активації

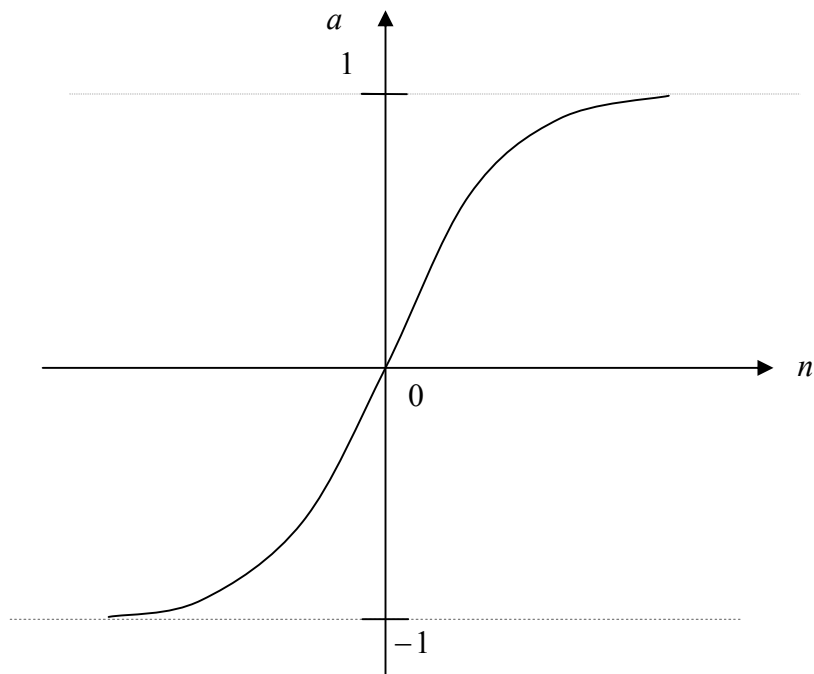


Рисунок А.5 – Функція активації типу гіперболічного тангенса

А.3 Нейрон з векторним входом

Нейрон з одним вектором входу $\mathbf{p}$ з R елементами $p_1, p_2, \dots, p_R$ показаний на рис. А.6. Тут кожен елемент входу умножається на ваги $w_{11}, w_{12}, \dots, w_{1R}$ відповідно і зважені значення передаються на суматор. Їх сума рівна скалярному добутку вектора-рядка $\mathbf{W}$ на вектор входу $\mathbf{p}$.

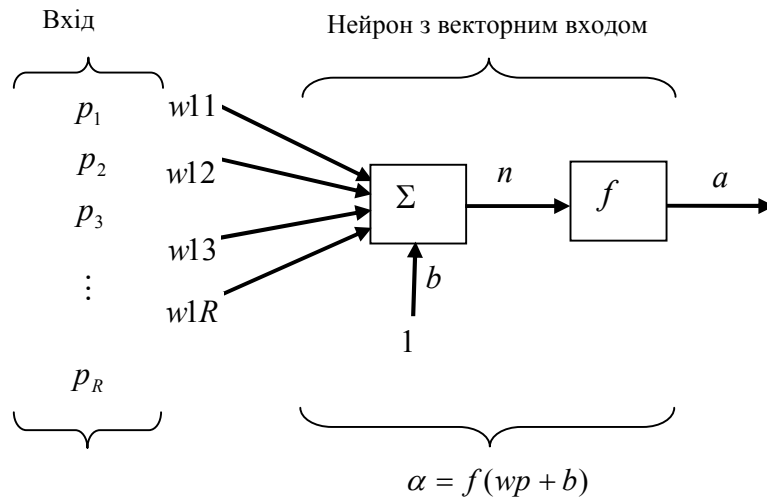


Рисунок А.6 – Схема нейрона з векторним входом

Нейрон має зсув b , який підсумовується із зваженою сумою входів. Результуюча сума n рівна

$$n = w_{11} \cdot p_1 + w_{12} \cdot p_2 + \dots + w_{1R} \cdot p_R + b \quad (\text{A.3})$$

і служить аргументом функції активації f . У нотації мови MATLAB цей вираз записується так:

$$n = \mathbf{W} \cdot \mathbf{p} + b . \quad (\text{A.4})$$

Структура нейрона, показана вище, містить багато зайвих деталей. При розгляді мереж, що складаються з великого числа нейронів, в ППП Neural Network Toolbox використовується укрупнена структурна схема нейрона (рис.А.7).

Вхід нейрона зображується у вигляді темної вертикальної межі, під якою вказується кількість елементів входу R . Розмір вектора входу $\mathbf{p}$ вказується нижче за символ $\mathbf{p}$ і рівний R . Вектор входу множиться на вектор-рядок $\mathbf{W}$ довжини R . Як і раніше, константа 1 розглядається як вхід, який множиться на скалярний зсув b . Входом n функції активації нейрона служить сума зсуву b і добутки $\mathbf{W} \cdot \mathbf{p}$. Ця сума перетвориться функцією активації f на виході якої отримуємо вихід нейрона a , який в даному випадку є скалярною величиною. Структурна схема, приведена на рис. А.7, називається шаром

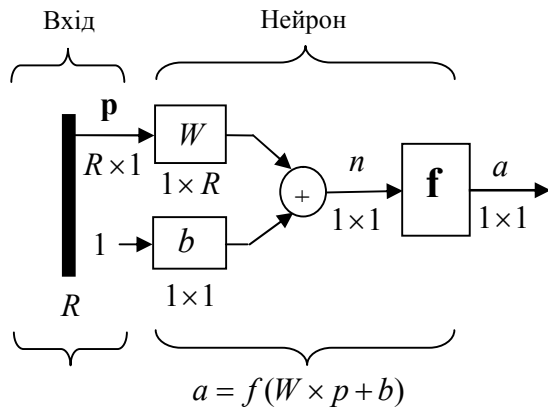


Рисунок А.7 – Укрупнена структурна схема нейрона

мережі. Шар характеризується матрицею вагів $\mathbf{W}$, зсувом b , операціями множення $\mathbf{W} \cdot \mathbf{p}$, підсумовування і функцією активації f . Вектор входів $\mathbf{p}$ зазвичай не включається в характеристики шару.

ДОДАТОК Б

АРХІТЕКТУРА НЕЙРОННИХ МЕРЕЖ

Б.1 Одношарові мережі

Реальна нейронна мережа може містити один або більша кількість шарів і відповідно характеризуватися як одношарова або як багатшарова.

Розгорнута схема мережі з одного шару з R вхідними елементами і S нейронами показана на рис. Б.1. У цій мережі кожен елемент вектора входу сполучений зі всіма входами нейрона і це з'єднання задається матрицею вагів $\mathbf{W}$; при цьому кожен i -й нейрон включає елемент, що підсумовує, який формує скалярний вихід $n(i)$. Сукупність скалярних функцій $n(i)$ об'єднується в S -елементний вектор входу $\mathbf{n}$ функції активації шаруючи. Виходи шару нейронів формують вектор-стовпець $\mathbf{a}$, і, таким чином, опис шару нейронів має вигляд

$$\mathbf{a} = f(\mathbf{W} \cdot \mathbf{p} + \mathbf{b}). \quad (\text{Б.1})$$

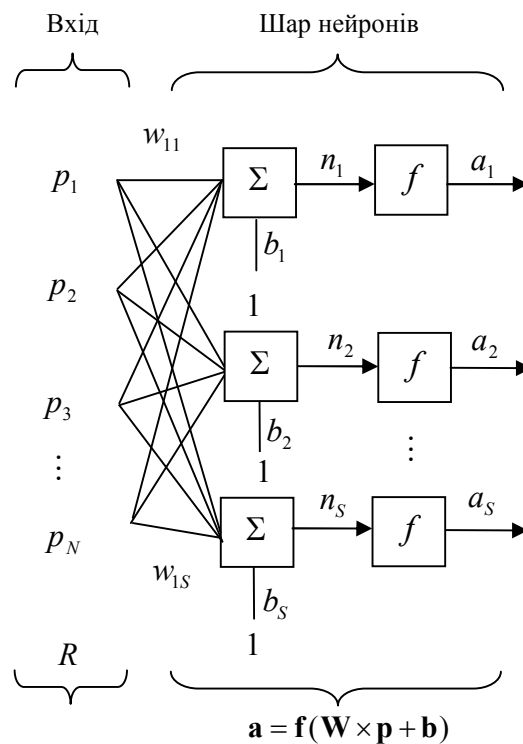


Рисунок Б.1 – Схема одношарової мережі

Кількість входів R в шарі може не співпадати з кількістю нейронів S . У кожному шарі, як правило, використовується одна і та ж функція активації. Проте можна створювати складені шари нейронів з використанням різних функцій активації, сполучаючи мережі, подібні зображеній на рис. Б.1, паралельно. Обидві мережі матимуть ті ж самі входи, і кожна мережа генеруватиме певну частину виходів. Елементи вектора входу передаються в мережу через матрицю вагів $\mathbf{W}$, що має вигляд

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1R} \\ w_{21} & w_{22} & \dots & w_{2R} \\ \dots & \dots & \dots & \dots \\ w_{S1} & w_{S2} & \dots & w_{SR} \end{bmatrix}. \quad (\text{Б.2})$$

Відмітимо, що індекси рядків матриці $\mathbf{W}$ указують адресатів (пункти призначення) вагів нейронів, а індекси стовпців – яке джерело є входом для цієї ваги. Таким чином, елемент матриці вагів $w_{12} = \mathbf{W}(1,2)$ визначає коефіцієнт, на який множиться другий елемент входу при передачі його на перший нейрон.

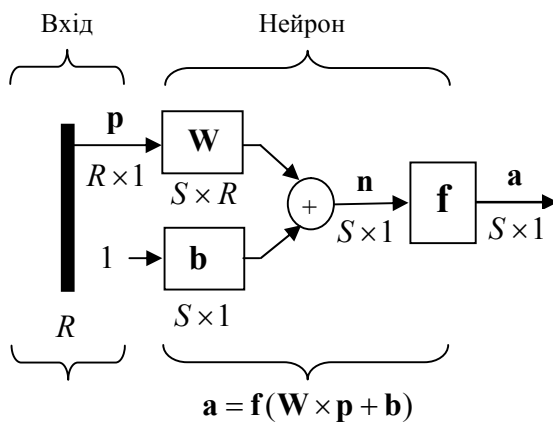


Рисунок Б.2 – Укрупнена структурна схема одношарової мережі

Для одношарової мережі з S нейронами укрупнена структурна схема показана рис. Б.2.

На рисунку позначено: $\mathbf{p}$ – вектор входу розміру $R \times 1$; $\mathbf{W}$ – вагова матриця розміру $S \times R$; $\mathbf{a}$, $\mathbf{b}$, $\mathbf{n}$ – вектори розміру $S \times 1$.

Б.2 Багат шарові мережі

Розглянемо мережі, що мають декілька шарів. Вагові матриці, сполучені з входами, називаються вагами вхідного шару, а вагові матриці для сигналів, ви-

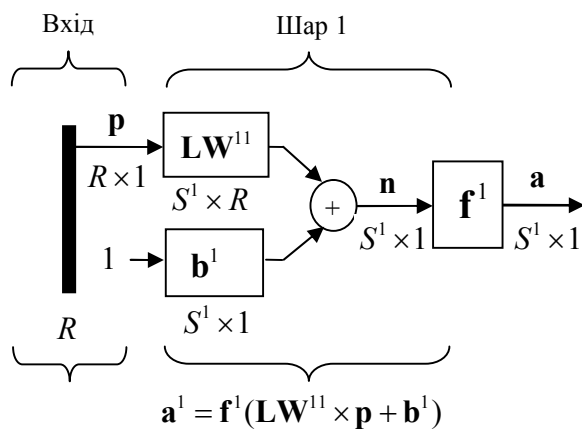


Рисунок Б.3 – Структурна схема першого шару багат шарової мережі

тікаючі з шару, називаються вагами вихідного шару. Далі, будемо використовувати верхні індекси, щоб вказати джерело і адресат для різних вагів і інших елементів нейронної мережі. Щоб пояснити це, розглянемо спочатку тільки один, перший шар багат шарової мережі (рис. Б.3)

Позначимо вагову матрицю, пов'язану з входами, через IW^{11} , верхні індекси якої указують, що джерелом входів є перший шар (другий індекс) і адреси том є також перший шар (перший індекс). Елементи цього шару, такі, як зсув b^1 , вхід функції активації n^1 і вихід шару a^1 мають верхній індекс 1, щоб позначити, що вони пов'язані з першим шаром. Надалі для матриць вагів входу і виходу шару будуть використані позначення IW (Input Weight) і LW (Layer Weight) відповідно.

Коли мережа має декілька шарів, то кожен шар має свою матрицю вагів W , вектор зсуву b і вектор виходу a . Щоб розрізняти вагові матриці, вектори виходу і т.д. для кожного з цих шарів, введемо номер шару як верхній індекс для тієї, що представляє інтерес, змінної. Використання цієї системи позначень для мережі з трьох шарів можна бачити на показаній нижче структурній схемі і з рівнянь, приведених в нижній частині рис. Б.4.

Мережа, показана вище, має R входів, S^1 нейронів в першому шарі, S^2 нейронів в другому шарі і т.д. Для спільності вважатимемо, що різні шари мають різне число нейронів. На зсуви для кожного нейрона поданий постійний вхідний сигнал 1. Відмітимо, що виходи кожного проміжного шару служать

входами для наступного шару. Таким чином, шар 2 може бути розглянутий як один шар мережі з S^1 входами S^2 нейронами і $S^1 \times S^2$ матрицею вагів $\mathbf{W}_2$. Вхід до шару 2 є 1, а вихід – 2. Тепер, коли позначені всі вектори і матриці шару 2, можна трактувати його як самостійну одношарову мережу. Такий підхід може бути використаний до будь-якого шару мережі.

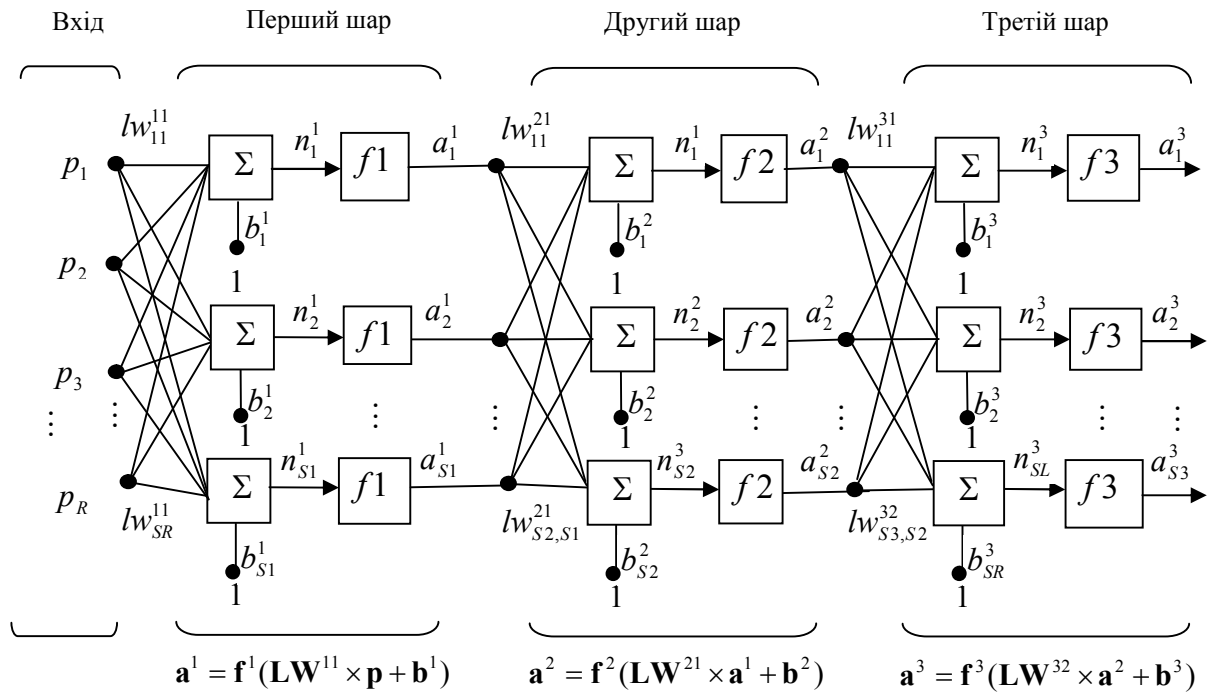


Рисунок Б.4 – Структурна схема багатошарової мережі

Шари багатошарової мережі мають різні призначення. Шар, який утворює вихід мережі, називається шаром виходу. Всі інші шари називаються прихованими шарами. Тришарова мережа, показана вище, має вихідний шар (шар 3) і 2 прихованих шари (шар 1 і шар 2). Ця ж тришарова мережа може бути представлена у вигляді укрупненої структурної схеми (рис. Б.5).

Відмітимо, що вихід третього шару $\mathbf{a}^3$ позначений через $\mathbf{y}$. Ця зроблено для того, щоб підкреслити, що вихід останнього шару є виходом мережі.

Багатошарові мережі володіють вельми потужними можливостями. Наприклад, двошарова мережа, в якій перший шар містить сигмоїдальну, а другий шар – лінійну функцію активації, може бути навчена апроксимувати з довільною точністю будь-яку функцію з кінцевим числом точок розриву.

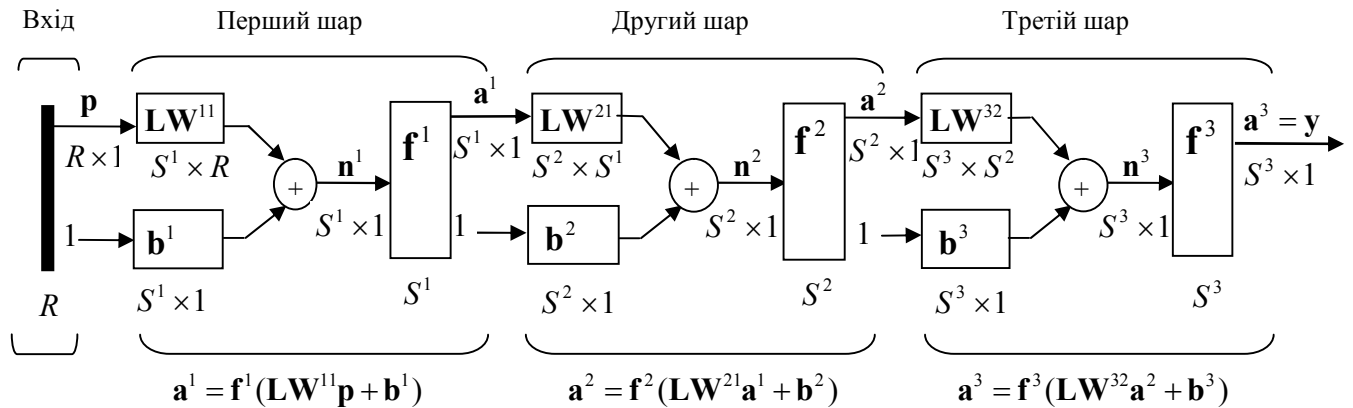


Рисунок Б.5 – Укрупнена структурна схема багатошарової мережі

На закінчення можна сформулювати наступні висновки. Вхід функції активації нейрона визначається зсувом і сумою зважених входів. Вихід нейрона залежить як від входів нейрона, так і від виду функції активації. Один нейрон не може вирішувати складні завдання, проте декілька нейронів, об'єднаних в один або декілька шарів, володіють великими можливостями.

Архітектура мережі складається з опису того, скільки шарів має мережу, кількості нейронів в кожному шарі, виду функції активації кожного шару і інформації про з'єднання шарів. Архітектура мережі залежить від того конкретного завдання, яке повинна вирішувати мережа.

Робота мережі полягає в обчисленні виходів мережі на основі відомих входів з метою формування бажаного відображення вхід/вихід. Конкретне завдання визначає число входів і число виходів мережі. Окрім числа нейронів у вихідному шарі мережі, для проектувальника важливе число нейронів в кожному шарі. Більша кількість нейронів в прихованих шарах забезпечує могутнішу мережу. Якщо повинне бути реалізоване лінійне відображення, то слід викори-

стовувати нейрони з лінійними функціями активації. При цьому треба пам'ятати, що лінійні нейронні мережі не можуть формувати нелінійні відображення. Використання нелінійних функцій активації дозволяє набудувати нейронну мережу на реалізацію нелінійних зв'язків між входом і виходом.

Мережі із зсувом дозволяють формувати складніші зв'язки між входами і виходами, чим мережі без зсуву. Наприклад, нейрон без зсуву, коли всі входи нульові, завжди задаватиме вхід функції активації рівним нулю, проте нейрон із зсувом може бути навчений так, щоб за тих же умов задати вхід функції активації довільної форми.

У багатошарових мережах часто застосовуються нелінійні сигмоїдальні функції активації типа логістичної (див. рис. А.4) або гіперболічного тангенса (див. рис. А.5).

Якщо останній шар багатошарової мережі використовує такі функції активації, то виходи мережі будуть обмежені. Коли у вихідному шарі використовуються лінійні нейрони, то виходи мережі можуть приймати довільні значення. У ППП NNT передбачені М-функції, що дозволяють обчислювати похідні функцій активації.

Б.3.Мережі з прямою передачею сигналу

Одношарова мережа з S нейронами з функціями активації logsig , що має R входів, показана на рис. Б.6. Ця мережа, що не має зворотних зв'язків, називається мережею з прямою передачею сигналу. Такі мережі часто мають один або більше прихованих шарів нейронів з сигмоїдальними функціями активації, тоді як вихідний шар містить нейрони з лінійними функціями активації. Мережі з такою архітектурою можуть відтворювати вельми складні нелінійні залежності між входом і виходом мережі.

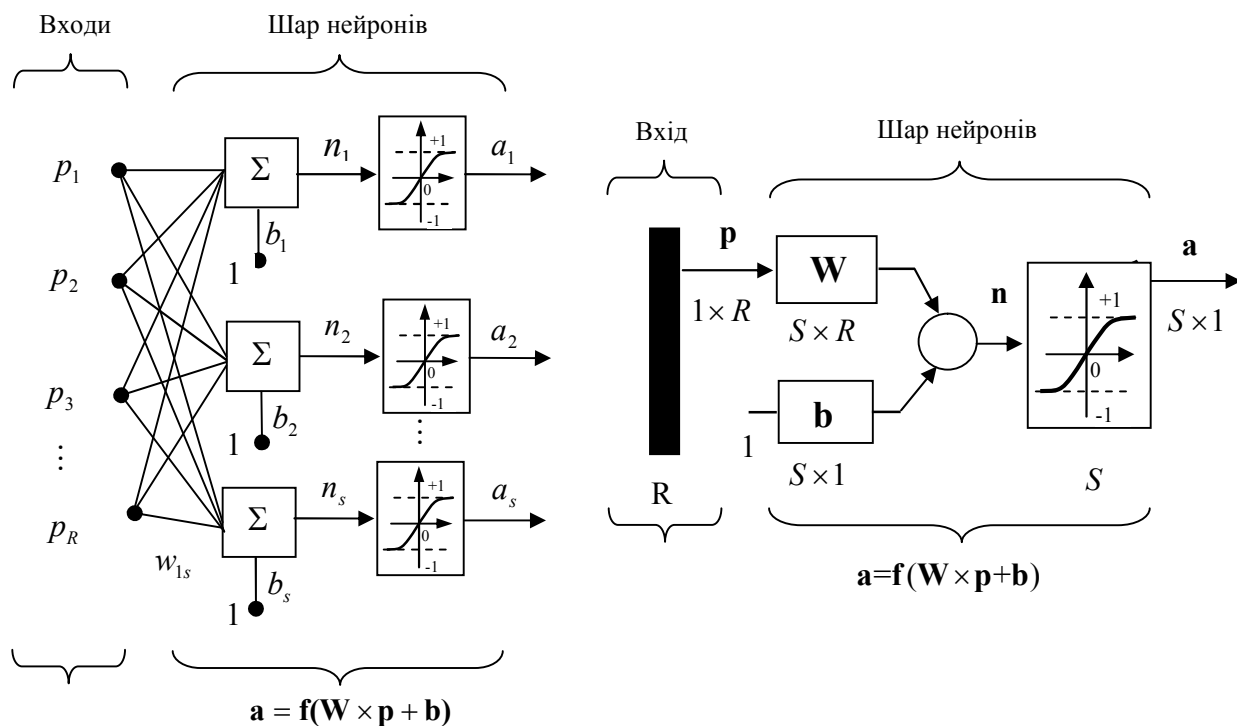


Рисунок Б.6 – Структурная схема одношаровой мережі з R входами

Двошарова мережа, показана на рис. Б.7, може бути використана для апроксимації функцій. Вона може достатньо точно відтворити будь-яку функцію з кінцевим числом точок розриву, якщо задати достатнє число нейронів прихованого шару.

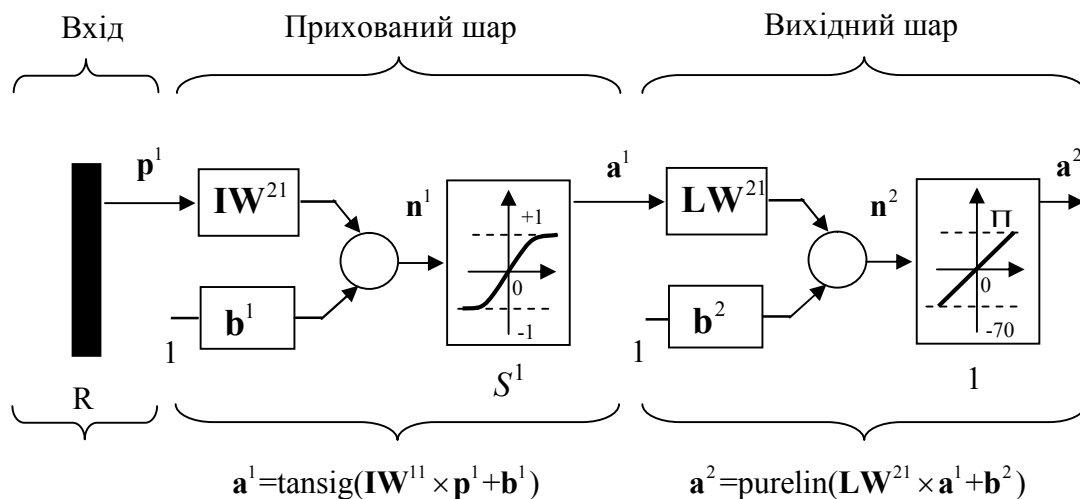


Рисунок Б.7 – Структурна схема багатошарової мережі з R входам

ДОДАТОК В

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

В.1 Сутність процесу навчання

Доведено, що будь-яка безперервна функція на замкнутій обмеженій множині може бути рівномірно наближена функціями, обчислюваними нейронними мережами, якщо функція активації нейрона двічі безперервно диференціюєма і нелінійна (існують і інші варіанти теорем про обличчя).

Таким чином, нейронні мережі є універсальними апроксимуючими системами.

Очевидно, що процес функціонування ШНМ, тобто суть дій, які вона здатна виконувати, залежить від величин синаптичних зв'язків, тому, задавшись певною структурою ШНМ, що відповідає якому-небудь завданню, розробник мережі повинен знайти оптимальні значення всіх вагових коефіцієнтів.

Цей етап називається навчанням ШНМ, і від того, наскільки якісно він буде виконаний, залежить здатність мережі вирішувати поставлені перед нею проблеми під час функціонування.

Навчання штучною нейронною полягає в настройці 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.

Навчання мережі відбувається відповідно до схеми, показаної на рис.В.1.

При навчанні мережі є деяка база даних, що містить приклади. Пред'являючи перший приклад на вхід мережі, отримуємо від неї деяку відповідь, не обов'язково вірну. Відома і вірна (бажаний) відповідь. Обчислюючи різницю між бажаною відповіддю і реальною відповіддю мережі, отримуємо вектор помилки. Алгоритм навчання – це набір формул, який дозволяє по вектору поми-

лки обчислити необхідні поправки для вагів мережі. Один і той же приклад може пред'являтися мережі багато раз.

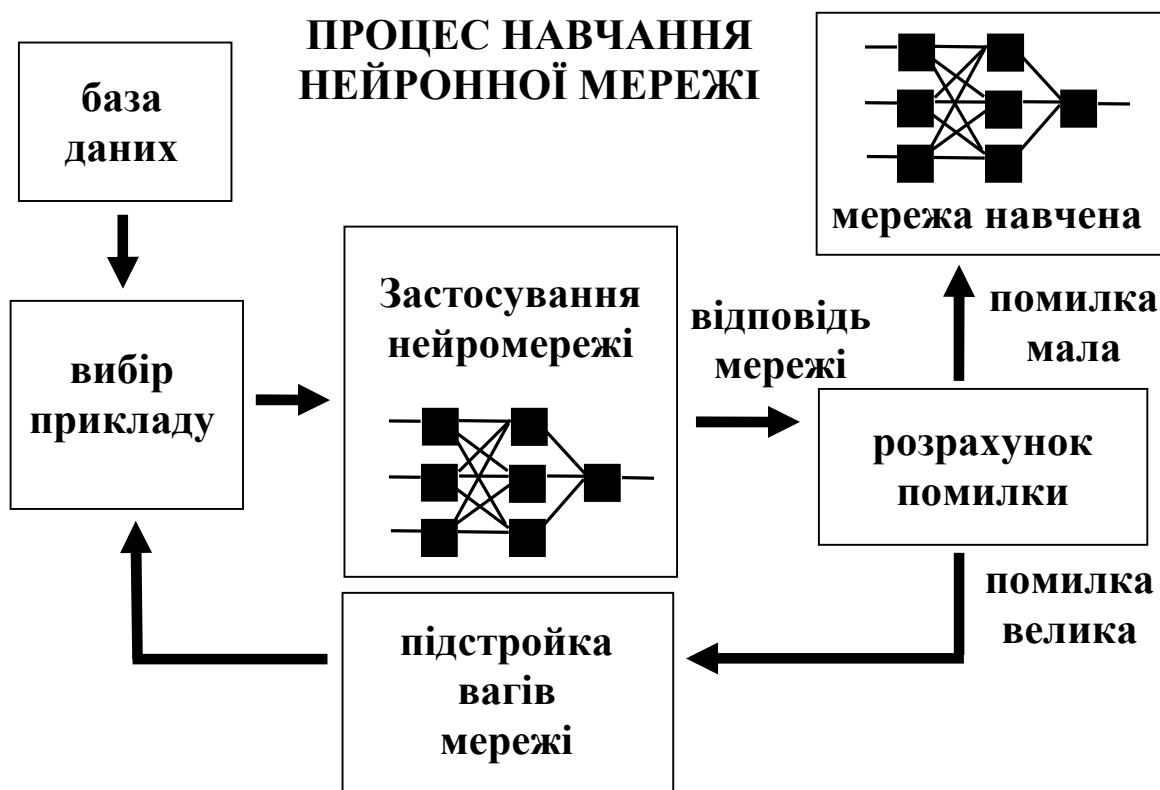


Рисунок В.1 – Ілюстрація процесу навчання НМ

Виявляється, що після багатократного пред'явлення прикладів ваги мережі стабілізуються, причому мережа дає правильні відповіді на всі (або майже все) приклади з бази даних. У такому разі говорять, що «мережа вивчила всі приклади», «мережа навчена», або «мережа натренована». У програмних реалізаціях можна бачити, що в процесі навчання функція помилки (наприклад, сума квадратів помилок по всіх виходах) поступово зменшується. Коли функція помилки досягає нуля або прийнятного малого рівня, тренування зупиняють, а отриману мережу вважають натренованою і готовою до застосування на нових даних.

Важливо відзначити, що вся інформація, яку мережа має про завдання, міститься в наборі прикладів. Тому якість навчання мережі безпосередньо залежить від кількості прикладів в навчальній вибірці, а також від того, наскільки

повно ці приклади описують дане завдання. Так, наприклад, безглуздо використовувати мережу для прогнозу фінансової кризи, якщо в навчальній вибірці криз не представлено. Вважається, що для повноцінного тренування потрібно хоч би декілька десятків (а краще за сотні) прикладів.

Математично процес навчання можна описати таким чином.

В процесі функціонування нейронна мережа формує вихідний сигнал Y відповідно до вхідного сигналу X , реалізуючи деяку функцію $Y = G(X)$. Якщо архітектура мережі задана, то вид функції G визначається значеннями синаптичних вагів і зсувів мережі.

Хай рішенням деякої задачі є функція $Y = F(X)$, задана парами вхідних, – вихідних даних $(X^1, Y^1), (X^2, Y^2), \dots, (X^N, Y^N)$, для яких $Y^k = F(X^k)$ ($k = 1 \dots N$).

Навчання полягає в пошуку (синтезі) функції G , близької до F і мінімізації деякої функції помилки E .

Якщо вибрана безліч навчальних прикладів — пара (X^k, Y^k) (де $k = 1 \dots N$) і спосіб обчислення функції помилки E , то навчання нейронної мережі перетворюється на завдання багатовимірної оптимізації, дуже велику розмірність, що має, при цьому, оскільки функція E може мати довільний вигляд, навчання в загальному випадку – багатоекстремальне неопукле завдання оптимізації.

В.2 Алгоритми навчання нейронних мереж

В.2.1 Класифікація алгоритмів навчання.

Для вирішення цього завдання можуть бути використані наступні (ітеративні) алгоритми.

1 – Алгоритми локальної оптимізації з обчисленням приватних похідних першого порядку:

градієнтний алгоритм (метод швидкого спуску);

методи з одновимірною і двовимірною оптимізацією цільової функції у напрямі антиградієнта;
метод зв'язаних градієнтів;
методи, що враховують напрям антиградієнта на декількох кроках алгоритму.

2 – Алгоритми локальної оптимізації з обчисленням приватних похідних першого і другого порядку:

метод Ньютона;
методи оптимізації з розрідженими матрицями Гессе;
квазіньютонівські методи;
метод Гаусса-Ньютона;
метод Левенберга-Марквардта і ін.

3 – Стохастичні алгоритми оптимізації:

пошук у випадковому напрямі;
імітація віджигу;
метод Монте-Карло (чисельний метод статистичних;
випробувань).

4) Алгоритми глобальної оптимізації (завдання глобальної оптимізації вирішуються за допомогою перебору значень змінних, від яких залежить цільова функція).

В.2.2 Алгоритм зворотного розповсюдження.

Розглянемо ідею одного з найпоширеніших алгоритмів навчання — алгоритму зворотного розповсюдження помилки (БР) (back propagation). Це ітеративний градієнтний алгоритм навчання, який використовується з метою мінімізації середньоквадратичного відхилення поточного виходу від бажаного виходу в багатошарових нейронних мережах.

Алгоритм зворотного розповсюдження використовується для навчання багатошарових нейронних мереж з послідовними зв'язками. Як зазначено вище, нейрони в таких мережах діляться на групи із загальним вхідним сигналом —

шари, при цьому на кожен нейрон першого шару подаються всі елементи зовнішнього вхідного сигналу, а всі виходи нейронів q -го шару подаються на кожен нейрон шаруючи $(q + 1)$. Нейрони виконують зважене (з синаптичними вагами) підсумовування елементів вхідних сигналів; до даної суми додається зсув нейрона. Над отриманим результатом потім виконується нелінійне перетворення за допомогою активаційної функції. Значення функції активації є вихід нейрона.

У багатошарових мережах оптимальні вихідні значення нейронів всіх шарів, окрім останнього, як правило, невідомі, і трьох- або більш шаровий персептрон вже неможливо навчити, керуючись тільки величинами помилок на виходах ШНМ. Найбільш прийнятним варіантом навчання в таких умовах виявився градієнтний метод пошуку мінімуму функції помилки з розглядом сигналів помилки від виходів ШНМ до її входів, тобто в напрямі, зворотному прямому розповсюдженню сигналів в звичайному режимі роботи. Цей алгоритм навчання ШНМ отримав назву процедури зворотного розповсюдження.

У даному алгоритмі функція помилки є сумою квадратів розузгодження (помилки) бажаного виходу мережі і реального. При обчисленні елементів вектора градієнта використаний своєрідний вид похідних функцій активації сигмоїдального типу. Алгоритм діє циклічно (ітеративно), і його цикли прийнято називати епохами. На кожній епосі на вхід мережі по черзі подаються всі навчальні спостереження, вихідні значення мережі порівнюються з цільовими значеннями і обчислюється помилка. Значення помилки, а також градієнта поверхні помилок використовується для коректування вагів, після чого всі дії повторюються. Початкова конфігурація мережі вибирається випадковим чином, і процес навчання припиняється або коли пройдена певна кількість епох, або коли помилка досягне деякого певного рівня малості, або коли помилка перестане зменшуватися (користувач може сам вибрати потрібну умову зупинки).

Приведемо словесний опис алгоритму.

Крок 1. Вагам мережі присвоюються невеликі початкові значення.

Крок 2. Вибирається чергова навчальна пара (X, Y) з навчальної множини; вектор X подається на вхід мережі.

Крок 3. Обчислюється вихід мережі.

Крок 4. Обчислюється різниця між необхідним (цільовим, Y) і реальним (обчисленим) виходом мережі.

Крок 5. Ваги мережі коректуються так, щоб мінімізувати помилку (спочатку ваги вихідного шаруючи, потім, з використанням правила диференціювання складної функції і відміченого своєрідного виду похідної сигмоїдальної функції, – ваги попереднього шару і т. п.).

Крок 6. Кроки з 2-го по 5-й повторюються для кожної пари навчальної множини до тих пір, поки помилка на всій множині не досягне прийнятної величини.

Кроки 2 і 3 подібні до тих, які виконуються у вже навченій мережі.

Обчислення в мережі виконуються пошарово. На кроці 3 кожний з виходів мережі віднімається з відповідного компоненту цільового вектора з метою отримання помилки. Ця помилка використовується на кроці 5 для корекції вагів мережі.

Кроки 2 і 3 можна розглядати як «прохід вперед», оскільки сигнал розповсюджується по мережі від входу до виходу. Кроки 4 і 5 складають «зворотний прохід», оскільки тут обчислюваний сигнал помилки розповсюджується назад по мережі і використовується для підстроювання вагів.

Класичний метод зворотного розповсюдження відноситься до алгоритмів з лінійною збіжністю. Його відомими недоліками є: невисока швидкість збіжності (велике число ітерацій, потрібних для досягнення мінімуму функції помилки), можливість сходитися не до глобального, а до локальних рішень (локальним мінімумам відміченої функції). Можливий також параліч мережі, при якому більшість нейронів функціонують при дуже великих значеннях аргументу функції активації, тобто на її пологій ділянці (оскільки помилка пропорційна похідною, яка на даних ділянках мала, то процес навчання практично завмирає).

Для усунення цих недоліків були запропоновані численні модифікації алгоритму зворотного розповсюдження, які зв'язані з використанням різних фун-

кцій помилки, різних процедур визначення напрямку і величини кроку і т.п. Розглянемо один з них.

В.3 Метод зворотного розповсюдження помилки

Як було зазначено вище, 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 вхідного вектора $\mathbf{r}$ у вихідний вектор $\mathbf{u} = \mathbf{q}^{(K)}$ останнього шару K . Помилка $\mathbf{e}_u$, явним чином залежить від коефіцієнтів $w_{i,j}^{(K)}$ і може бути використана як аргумент функціонала Q для настройки базових елементів K -го шару. Явної залежності вектора $\mathbf{e}_u$ і функції $Q(\mathbf{e}_u)$ від вагових коефіцієнтів $w_{i,j}^{(l)}$ ($l = \overline{1, K-1}$) немає. Тому у разі використання критерію навчання $Q(\mathbf{e}_u)$ для настройки коефіцієнтів $w_{i,j}^{(l)}$ помилка $\mathbf{e}_u$ в процесі навчання багатошарової нейронної мережі перераховується послідовно в узагальнені помилки $\sigma^{(l)}$, що явно залежать від значень $w_{i,j}^{(l)}$. Настройка коефіцієнтів $w_{i,j}^{(l)}$ здійснюється за δ -правилом, де використовуються узагальнені помилки $\sigma^{(l)}$. Очевидно, що для шару K помилка $\sigma^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$. У методі ВР далі використовується локальний квадратичний функціонал $Q(\sigma) = 0,5 \sigma^T \sigma$. Мінімізація $Q(\sigma)$ по вагових коефіцієнтах $w_{i,j}^{(l)}$ ($l = K-1, \dots, 1$), що настраюються, методом градієнтного спуску з перерахунком помилки

$\sigma^{(K)} = \mathbf{e}_u(\mathbf{w}^{(K)})$ для шару K в помилку $\sigma^{(l)} = \mathbf{e}(\mathbf{w}^{(l)})$ для шарів $l = K - 1, K - 2, \dots, 1$ визначає сутність ВР-технології навчання багат шарових нейронних мереж.

В.3.1. 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:

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

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,

Градiєнт

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

де похідна:

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

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

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

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

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

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

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

З урахуванням приведених матриць Якобі градієнт функціонала настройки K -го шару багатошарової 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

$$\mathbf{w}^{(K)}(k+1) = \mathbf{w}^{(K)}(k) + \gamma \left(\frac{\partial \mathbf{s}^{(K)}}{\partial \mathbf{w}^{(K)}(k)} \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) \right) \mathbf{e}_u(k+1). \quad (\text{B.2})$$

Рекурентний алгоритм (B.2) 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 $\mathbf{q}^{(K-1)}$.

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i(k+1) \mathbf{q}^{(K-1)}(k+1). \end{aligned} \quad (\text{B.3})$$

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 мадалини з лінійними функціями активації і $f'_s(s) = 1$. Тоді $\sigma_i = e_{ui}$ і алгоритм настройки вагових коефіцієнтів шаруючи K приймає вигляд:

$$\mathbf{w}_i^{(K)}(k+1) = \mathbf{w}_i^{(K)}(k) + \gamma e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1). \quad (\text{B.4})$$

У алгоритмах (B.2)-(B.4) $k = 1, 2, \dots$ – 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.3.2. 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 «при-

хованих» 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 помилки $\sigma^{(l)} = \sigma(\mathbf{w}_i^l)$ залежать явно від вагових коефіцієнтів, що настроюються $\mathbf{w}_i^l$. Розглянемо суть необхідних перетворень в δ -правилі (2.2), 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

$$\Lambda^{(l)} = \frac{\partial Q}{\partial \mathbf{q}^{(l)}} = \mathbf{D}^{(l+1)} \Lambda^{(l+1)}, \quad l = \overline{K-1, 1}. \quad (\text{B.5})$$

Матрицю $\mathbf{D}^{(l+1)}$ назвемо 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:

$$\mathbf{D}^{(l+1)} = \left\| \frac{\partial q_j^{(l+1)}}{\partial q_i^{(l)}} \right\|_{i=0, \overline{n_l}}^{j=0, \overline{n_{l+1}}} = (\mathbf{W}^{(l+1)})^T \mathbf{F}^{(l+1)} = (\mathbf{W}^{(l+1)})^T \text{diag} (f'_s(s^{(l+1)})). \quad (\text{B.6})$$

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 $l = \overline{K-1, 1}$:

$$\nabla \mathbf{w}_i^{(l)} Q = \mathbf{S}^{(l)} \mathbf{F}^{(l)} \Delta^{(l)} = \mathbf{q}^{(l-1)} (\mathbf{f}'_s(\mathbf{s}^{(l)}))^T \Lambda^{(l)} \quad (\text{B.7})$$

Повний алгоритм навчання багатошарової нейронної мережі по методу ВР включає початковий алгоритм настройки вагових коефіцієнтів K -го шару (B.4) і алгоритми настройки базових елементів шарів $l = \overline{K-1, 1}$ використовують формулу (B.7).

В.3.3. 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)

$$\begin{aligned} \mathbf{w}_i^{(K)}(k+1) &= \mathbf{w}_i^{(K)}(k) + \gamma f'_s(s_i^{(K)}(k+1)) e_{ui}(k+1) \mathbf{q}^{(K-1)}(k+1) = \\ &= \mathbf{w}_i^{(K)}(k) + \gamma \sigma_i^{(K)}(k+1) \mathbf{q}^{(K-1)}(k+1); \end{aligned} \quad (\text{B.8})$$

$$\gamma > 0; \quad k = 0, 1, 2, \dots; \quad e_{ui} = u_i^* - u_i; \quad i \in \overline{1, n_K};$$

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

б)

$$\mathbf{w}_i^{(l)}(k+1) = \mathbf{w}_i^{(l)}(k) - \gamma \mathbf{q}^{(l-1)}(k+1)(\mathbf{f}'_s(s_i^{(l)}(k+1)))^T \Lambda^{(l)}(k);$$

$$\Lambda^{(l)}(k) = \mathbf{D}^{(l+1)} \Lambda^{(l+1)}(k);$$

(B.9)

$$\mathbf{D}^{(l+1)}(k) = (\mathbf{W}^{(l+1)}(k))^T \text{diag}(f'_s(\mathbf{s}^{(l)}(k)));$$

$$\gamma > 0; \quad l = \overline{K-1, 1}; \quad \mathbf{q}^{(l-1)} = \text{col}(q_0^{(l-1)}, q_1^{(l-1)}, \dots, q_{l_{K-1}}^{(l-1)}); \quad i = \overline{0, n_l}$$

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

$$\mathbf{w}_i^{(K-1)}(K+1) = \mathbf{w}_i^{(K-1)}(k) - \gamma \mathbf{q}^{(K-2)}(k+1)(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k),$$

$$\Lambda^{(K-1)}(k) = \mathbf{D}^{(K)}(k) \Lambda^{(K)}(k);$$

$$\mathbf{D}^{(K)}(k) = \mathbf{W}^{(K)}(k) \text{diag}(f'_s(\mathbf{s}^{(K)}(k))) = \mathbf{W}^{(K)}(k);$$

$$\Lambda^{(K)}(k) = \text{col}\left(\frac{\partial Q}{\partial \mathbf{q}_1^{(K)}}, \dots, \frac{\partial Q}{\partial \mathbf{q}_{n_K}^{(K)}}\right) = -\mathbf{e}_u(k).$$

тоді

$$\Lambda^{(K-1)}(k) = -\mathbf{W}^{(K)}(k) \mathbf{e}_u(k),$$

$$(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \Lambda^{(K-1)}(k) =$$

$$= -(\mathbf{f}'_s(s_i^{(K-1)}(k+1)))^T \mathbf{W}^{(K)}(k) \mathbf{e}_u(k) = \sigma_1^{(K-1)}(k+1)$$

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

$$\begin{aligned}
\mathbf{w}_i^{(K-1)}(k+1) &= \mathbf{w}_i^{(K-1)}(k) - \gamma \sigma_i^{(K-1)}(k) \mathbf{q}^{(K-2)}(k+1); \\
\sigma_i^{(K-1)}(k+1) &= -(\mathbf{f}'_s(s_i^{(K-1)}(k)))^T \mathbf{W}^{(K)}(k) \mathbf{e}^{(K)}(k+1); \\
\mathbf{e}_u^{(K)}(k) &= \mathbf{u}^*(k) - \mathbf{q}^{(K)}(k).
\end{aligned} \tag{B.10}$$

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.8) (B.9).

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 $e_u(t)$. на інтервалі її визначення. Використання іншої вимірювальної інформації, наприклад помилки відтворення заданої функції $r(t)$ 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.4 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 виявитись 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 перекресної 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 трохи.

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 кожну 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 (контрольному) множині.

Для того, щоб додати остаточної 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 множині, реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки один раз: якщо її використовувати повторно для коректування процесу навчання, то вона фактично перетвориться на контрольну множину.

ДОДАТОК Г

УЗАГАЛЬНЕНЕ НЕЙРОННЕ УПРАВЛІННЯ З ПРОГНОЗОМ

Г.1 Алгоритм нейроуправління з прогнозом

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

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

Узагальнене управління з прогнозом, введене Кларком і його співробітниками в 1987, належить до класу методів цифрового управління, званих модель-орієнтоване управління з прогнозом (Model-Based Predictive Control, MBPC) [38-44]. MBPC техніки були успішно проаналізовані і впроваджені в процес управління виробництвом в кінці 1970-х і продовжують використовуватися, оскільки вони системно враховують обмеження реальних об'єктів в режимі реального часу. GPC застосовуються в управлінні немінімально фазовими об'єктами, нестійкими об'єктами і об'єктами із змінним або невідомим часом затримки. Узагальнене управління з прогнозом також є робастним по відношенню до помилок моделювання, перевизначення і недовизначення значень параметрів і шумів датчиків [38]. GPC було спочатку розроблене для моделей лінійних об'єктів, що приводить до рівнянь, які можуть бути вирішені аналіти-

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

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

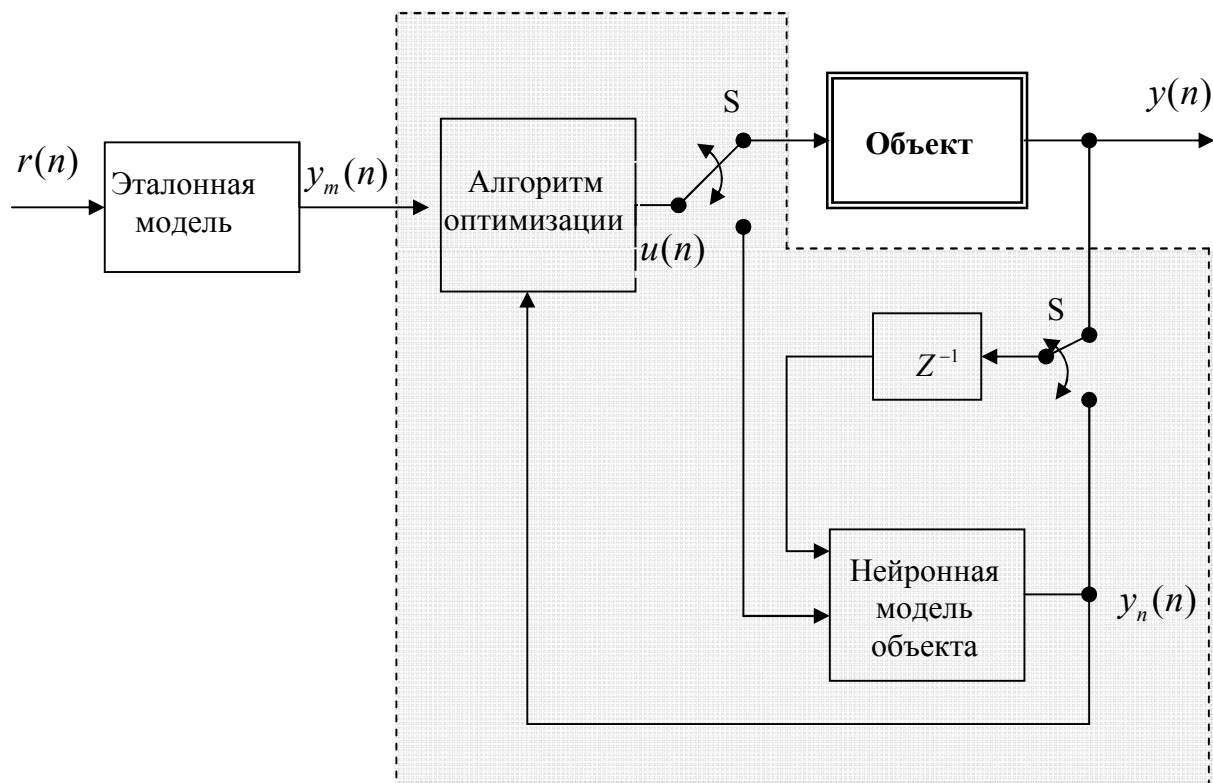


Рисунок Г.1 – Схема системи узагальненого нейроуправління з прогнозом

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

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

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

Узагальнений алгоритм нейроуправління з прогнозом складається з наступних основних кроків.

- 1) Генерується задаюча траєкторія. Якщо майбутня траєкторія $y_m(n)$ невідома, вважається $y_m(n) = const$.
- 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 [41], Simplex [43] і Successive Quadratic Programming [43,44]. Вибір методу мінімізації може бути заснований на наступних критеріях: число ітерацій, обчислювальні витрати і точність рішення. У загальному випадку, ці алгоритми вимагають великого числа ітерацій, що робить управління в режимі реального часу скрутним. Дуже невелике число публікацій присвячені реалізаціям управління в режимі реального часу для об'єктів, що мають великі постійні часу [43,44]. Для можливості використання в системах з об'єктами, що мають малі постійні часу, потрібний швидший оптимізаційний алгоритм. Алгоритм Ньютона-Рафсона володіє квадратичною збіжністю, тоді як інші алгоритми мають нижчу швидкість збіжності. Вища швидкість збіжності алгоритму Ньютона-Рафсона вимагає високих обчислювальних витрат, але виправдовується показником швидкості збіжності.

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

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

де N_1 – нижня межа прогнозу;

N_2 – верхня межа прогнозу;

N_u – діапазон управління;

y_m – бажана траєкторія;

y_n – передбачений вихід нейронної мережі;

λ – ваговий множник;

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

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

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

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

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

кожної ітерації розраховується також значення управляючого вектора

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

де $k=1, \dots$ – номер ітерації.

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

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

де якобіан позначений, як

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

а гесіан, як

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

Рішення рівняння (8.2) вимагає явного звернення матриці Гессе. Цей процес може зажадати великих обчислювальних витрат. Одним з підходів, використовуваних, щоб уникнути звернення матриці, служить LU декомпозиція [45] для отримання вхідного вектора $\mathbf{U}(k+1)$. Це досягається шляхом запису рівняння (Г.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 (Г.3) може бути вирішене за допомогою двох процедур, описаних в [45]: процедури LU декомпозиції, ludcmp і процедури рішення системи лінійних рівнянь lubksb. Після того, як обчислений вектор $\mathbf{x}$. Процедура повторюється до тих пір, поки зміна кожній компоненти $\mathbf{U}(k+1)$ не стає меншим деякого ε . При рішенні системи відносно $\mathbf{x}$ для кожної ітерації алгоритму Ньютона-Рафсона необхідне обчислення кожного елементу якобіана і гессиана. Елемент якобіана з індексом h має наступний вигляд:

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

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

дельта функції, яка визначається як

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

отримаємо

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

Елемент з індексами m, h гессиана визначається таким чином:

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

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

Можна скористатися також визначенням дельта-функції для виразу

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

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

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

Г.2 Структура нейронної мережі для моделювання об'єкту

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

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

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

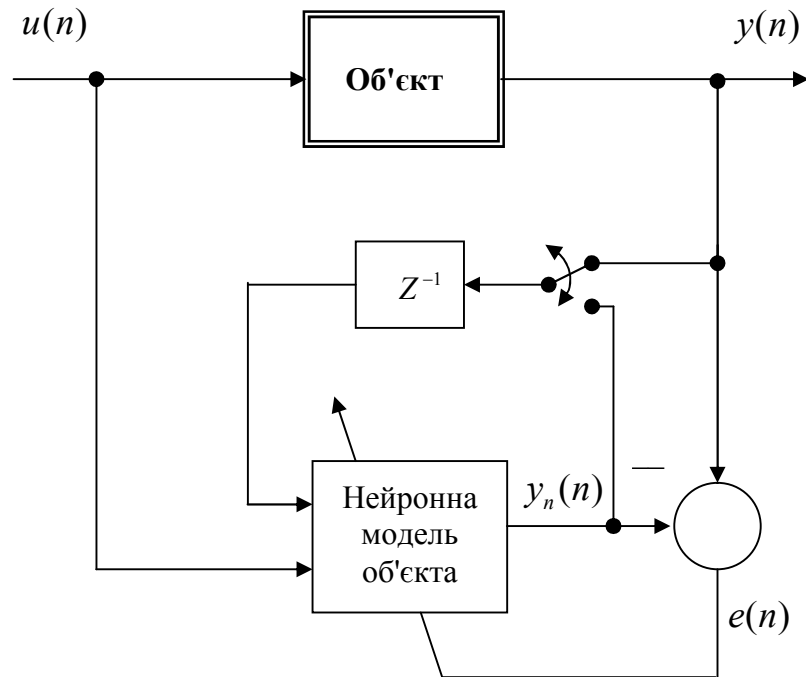


Рисунок Г.2 – Схема тренування мережі в режимі offline

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

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

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

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

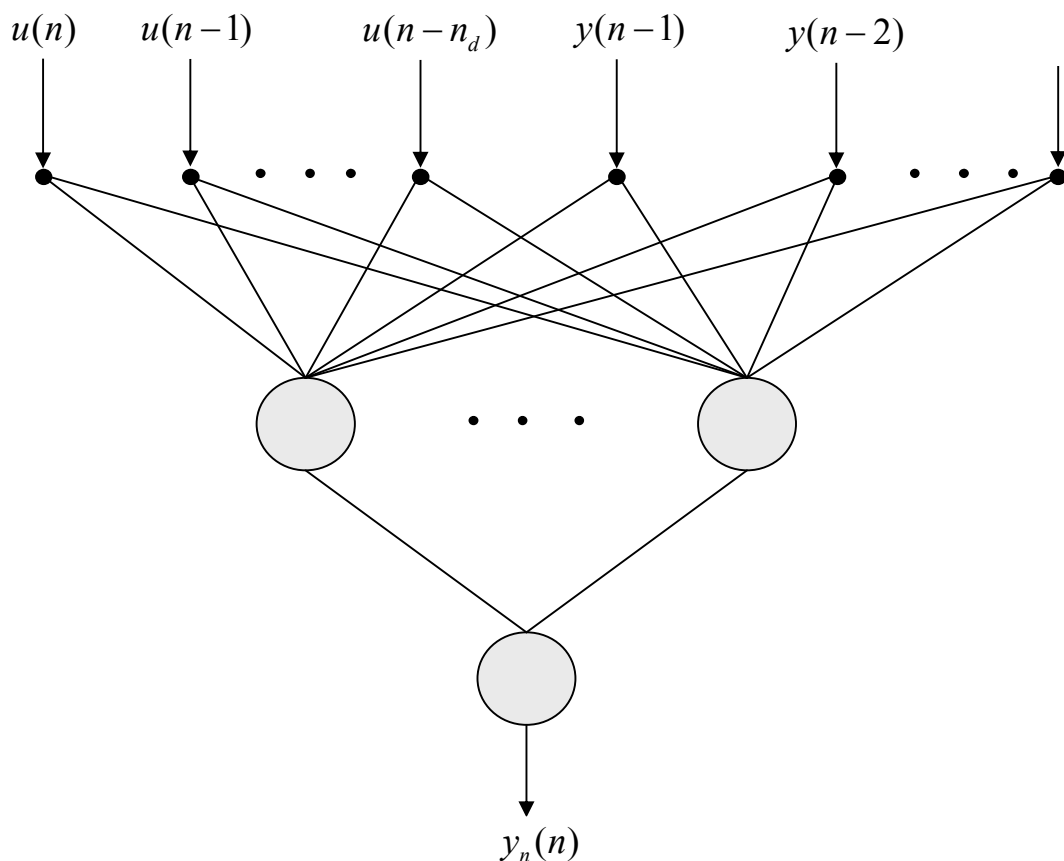


Рисунок Г.3 – Багатошарова нейронна мережа прямого розповсюдження з елементами затримок за часом

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

Рівняння мережі з даною архітектурою виглядають таким чином:

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

і

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

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

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

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

і

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

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

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

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

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% механізму пересування візка зливковогоза
%(кільцева зливкоподача)
% 2024 рік

% Вихідні дані

kd=0.233;
In=415;
Mn=In/kd;
ktp=20;
kn=0.109;
kc=0.024;
Tmc=0.005;
Ta=0.035;
Te=0.043;
Re=0.093;
Jd=19.5;
Jm=191;
w0=1.5;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);
b12=0.02*c12;
Js=Jd+Jm;
Tme=2*Tmc+Ta;
kic=Re/(2*Tmc*kc*ktp);
kpc=kic*Te;
kpe=Js*kd^2*kc/(2*Tme*kn);
kie=kpe/(4*Tme);

Uz=24;

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

```

zeros(1,2) kn/(Ta*kd) zeros(1,3) -1/Ta 0
zeros(1,5) kie -kie 0];
B=[0 -1/Jm;
zeros(4,2);
1/Ta 0;
0 0
0 0];
C=zeros(6,8);
C(1,1)=1;C(2,2)=1;C(3,3)=1;C(4,4)=1;
C(5,5)=1;C(6,8)=1;%C(6,6)=1;%C(3,7)=1;C(4,8)=1;
D=zeros(6,2);

```