

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

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

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

_____/ Нестор СУЛАДЗЕ
(підпис) (ім'я та прізвище)

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

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

«До захисту допущено»

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

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

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

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

« » 2024р.

ЗАВДАННЯ

на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Нестор СУЛАДЗЕ
(ім'я, прізвище)

1. Тема роботи Нейромережеві системи управління основними механізмами мостового крана. Тема комплексна. Спеціальна частина. Синтез і дослідження системи управління механізмом підйому з нейрорегулятором NN Predictive Controller
затверджена наказом по університету № 4801-5/3704 від “21” листопада 2024 р.
2. Термін здачі закінченої роботи “5” грудня 2024 р.
3. Вихідні дані до роботи Вантажопідйомність крана $Q=41 \cdot 10^3$ кг; продуктивність крана $C=240$ кг/с; маса моста без візка $G_m=28,5 \cdot 10^3$ кг; маса візка $G_v=8,5 \cdot 10^3$ кг; діаметр коліс моста $D_m=0,56$ м; діаметр коліс візка $D_v=0,25$ м; швидкість пересування моста $V_m=0,85$ м/с; швидкість пересування візка $V_v=0,55$ м/с; швидкість підйому $V_n=0,12$ м/с; маса вантажозахватного пристосування $q=1 \cdot 10^3$ кг; середня висота підйому і спуску $H_n=8$ м; діаметр барабана механізму підйому $D=0,75$ м. Перерегулювання швидкості по задаючій дії більше 10%, час регулювання до 5 с.
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електро-механічною системою механізму підйому мостового крана. Розрахувати систему підлеглого регулювання з внутрішнім контуром струму і зовнішнім контуром ЕРС. Розробити математичну модель двомасової системи. Виконати моделювання на ЕОМ та провести аналіз показників якості перехідних процесів змінних стану двомасової системи. Розробити структурну схему нейромережевої системи управління. Виконати синтез нейрорегулятора на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Провести аналіз результатів моделювання системи з нейрорегулятором.
5. Перелік графічного матеріалу (презентаційний матеріал) 1. Магістерська кваліфікаційна робота. Формальні ознаки. 2. Механізм підйому. Схема кінематична. 3. Нейронні мережі. Структурні схеми. 4. Навчання нейронних мереж. Метод зворотного розповсюдження помилки. 5. Нейромережеві системи управління. Структурні схеми. 6. Нейромережева ідентифікація. NARX моделі. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. Магістерська кваліфікаційна робота. Висновки.

6. Консультант:

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

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

Керівник

(підпис)

Геннадій КАНЮК
(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Нестор СУЛАДЗЕ
(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 підйому мостового крана.	09.09.2024-18.09.2024	
2	Розрахунок системи автоматичного регулювання координат електроприводу. Дослідження динамічних характеристик системи.	19.09.2024-15.10.2024	
3	Дослідження системи з урахуванням пружних властивостей підйимального канату. Розробка математичної моделі двомасової системи та моделювання двомасової системи на ЕОМ.	16.10.2024-05.11.2024	
4	Синтез нейромережової системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережової системи.	06.11.2024-25.11.2024	
5	Оформлення роботи	26.11.2024-05.12.2024	

Студент (ка)

(підпис)

Нестор СУЛАДЗЕ
(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 206 сторінки, 64 рисунків, 2 таблиць, 58 використаних джерел, 4 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму підйому мостового крана для підвищення показників якості функціонування системи на основі нейрорегуляторів.

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

В результаті виконання магістерської кваліфікаційної роботи науково обґрунтовано перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму підйому мостового крана. Розроблена система управління координат електроприводу механізму. Розраховані параметри системи автоматичного управління. Проведено моделювання системи на ЕОМ. Розроблена математична модель системи з урахуванням пружних властивостей підйимального канату. Розраховані динамічні характеристики двомасової системи. Розглянута структура нейромережевої системи управління і виконаний синтез нейрорегулятора NN Predictive Controller в системі MATLAB. Проведено моделювання нейромережевої системи управління з нейрорегулятором на ЕОМ. Встановлено, що синтезована нейромережева система має високі показники якості функціонування.

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

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

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

ABSTRACT

An explanatory message consists of: 206 pages, 64 figures, 2 tables, 58 used sources, 4 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the lifting mechanism of the bridge crane to improve the quality indicators of the system functioning on the basis of neuroregulators.

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

As a result of the master's qualification work scientifically grounded perspective the use of neural network technology to control multimass electromechanical system of the lifting mechanism of the bridge crane. The control system of coordinates of mechanism electric drive is worked out. The parameters of automatic control system are calculated. The simulation of the system is executed on a computer. Worked out the mathematical model of the system which takes into account the elastic properties of the lifting rope is developed. Dynamic descriptions of the two-mass system are calculated.. The structure of neural network control system are considered and made a synthesis of neuroregulator NN Predictive Controller in MATLAB. The simulation of neural network control system with neuroregulation is conducted on a computer. It is set that synthesized the neural network system has high indexes of quality of functioning.

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

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

Keywords: ARTIFICIAL NERO, NEURAL NETWORKS, BRIDGE CRANE, ELECTRIC DRIVE OF LIFTING MECHANISM, INFERIOR CONTROL SYSTEM, 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	8
РОЗДІЛ 1. АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.	
ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	13
1.1 Галузі застосування штучних нейронних мереж	13
1.2 1.2 Огляд структур систем нейрорегулювання.	
Вибір структури системи	18
1.3 Вимоги до систем управління електроприводами основних механізмів кранів	39
1.4 Обґрунтування вибору системи електроприводу механізму підйому мостового крана	42
1.5 Постановка задачі дослідження і шляхи її вирішення	44
ВИСНОВКИ ДО РОЗДІЛУ 1	49
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ МОСТОВОГО КРАНА	51
2.1 Технічні характеристики мостового крана. Вимоги до системи управління	51
2.2 Розрахунок потужності електроприводу механізму підйому. Вибір приводного двигуна	52
2.3 Вибір основного електроустаткування електроприводу і розрахунок параметрів силового кола	56
2.4 Вибір структури системи автоматичного управління	61
2.5 Синтез послідовного корегуючого пристрою контура струму	62
2.6 Синтез послідовного корегуючого пристрою контура ЕРС	71
2.7 Дослідження динамічних характеристик системи	78
ВИСНОВКИ ДО РОЗДІЛУ 2	82
РОЗДІЛ 3 ДОСЛІДЖЕННЯ СИСТЕМИ З УРАХУВАННЯМ КІНЦЕВОЇ ЖОРСТКОСТІ ПІДЙОМНОГО КАНАТА	84
3.1 Розрахункові схеми механічної частини електроприводу	84
3.2 Математична модель двомасової системи	90
3.3 Розрахунок перехідних процесів двомасової систем на ЕОМ	97

ВИСНОВКИ ДО РОЗДІЛУ 3	103
РОЗДІЛ 4. СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	104
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	104
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	106
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	109
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	152
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	154
ВИСНОВКИ ДО РОЗДІЛУ 4	160
ВИСНОВКИ	162
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	164
ДОДАТОК А. БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ	170
А.1 Структура штучного нейрона	170
А.2 Одношарові нейронні мережі	171
А.3 Багатошарові нейронні мережі	174
А.4 Здібності апроксимації і узагальнення багатошарових мереж.....	175
ДОДАТОК Б. НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ.....	178
Б.1 Завдання навчання	178
Б.2 Процедури адаптації і навчання.....	179
Б.3 Явище перенавчання	181
Б.4 Властивість узагальнення	184
Б.5 Вибір методу навчання нейронної мережі	186
ДОДАТОК В. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ.....	195
В.1 Моделі систем, використовувані при нейромережевій ідентифікації	195
В.2 Нейронні мережі як універсальні моделі.....	200
В.3 Ідентифікація динамічних систем за допомогою нейронних мереж	203
ДОДАТОК Г. ФАЙЛ ВИХІДНИХ ДАНИХ.....	206

ВСТУП

Актуальність теми. Технічний прогрес в різних галузях промисловості пов'язаний з технологією виробництва, що безперервно ускладнюється, з підвищенням вимог до точності виконання різних операцій, передбачених технологічним процесом виготовлення виробів. При цьому зростають вимоги до якості продукції, що випускається, при все більш складному процесі її виробництва. Разом з тим росте об'єм виробництва, що висуває вимоги підвищення продуктивності машин за рахунок збільшення як їх потужності, так і швидкості виконання різних дослідів технологічного процесу. Оскільки переважна більшість виробничих машин оснащуються електричними приводами, зростання вимог до цих машин веде до посилювання вимог до електроприводу, на який покладається завдання здійснення складних переміщень робочих органів механізму. В процесі реалізацій цих переміщень виникає необхідність розгону, гальмування, реверсу електроприводу, підтримки постійності регульованої величини (координати), зміна її по певному закону і т.д.

Вантажопідйомні крани є механізмами, до яких повною мірою відносяться всі перераховані вимоги. Більшість вантажопідйомних кранів характеризуються постійно змінними умовами використання при переробці вантажів, і тому механізми кранів, електроприводи, що мають в своєму складі, повинні бути в максимальному ступені пристосовані до роботи з вантажами, що постійно змінюються, різноманітними по масі, розмірам, формі, і в умовах виробничих приміщень або на відкритих вантажних майданчиках.

Дуже високий діапазон зміни навантажень практично будь-якого з електроприводів кранів є одним з головних чинників, що вимагають особливого підходу до вибору розрахункових параметрів приводних електродвигунів, систем управління і захисту.

Запити на вдосконалення систем управління вимагають нових шляхів розвитку автоматичних систем, що призводить до розгляду нових методів управління. Останнім часом, завдяки інтенсивним дослідженням, методи адап-

тивного і робастного регулювання здобули широке визнання. Вони забезпечують стійкість систем змінної динаміки, не потребуючи точного знання параметрів моделі об'єкта. Адаптивні методи намагаються відповідно налаштувати регулятор шляхом оцінки параметрів невідомого об'єкта, тоді як робастні регулятори залишаються сталими, забезпечуючи стійкість системи у межах допустимих змін параметрів об'єкта. Однак проблема синтезу адаптивних систем управління для невідомих, нелінійних та змінюючихся об'єктів залишається нерозв'язаною як з теоретичного, так і з практичного погляду. Методи робастного регулювання обмежені, переважно, лінійними системами і деякими спеціальними класами нелінійних систем. Отже, незважаючи на досягнутий прогрес, важливо визнати, що універсального та ідеального регулятора наразі не існує.

Останнім часом штучні нейронні мережі (ШНМ) стали привабливою альтернативою традиційних методів управління нелінійними об'єктами [1-12]. Одна з головних переваг ШНМ - їх здатність до самонавчання, що дозволяє отримувати простіші рішення для складних завдань управління. Також, наявність у ШНМ нейронів з нелінійними функціями активації дозволяє використовувати їх для управління нелінійними об'єктами, що залишаються непідходящими для традиційних методів..

Нейронні мережі відкривають нові можливості для створення гнучких, адаптивних систем регулювання, оскільки вони вивчають динамічні зв'язки на основі тренувальних даних. Ці мережі мають різні підходи - від структурованих, де елементи матриць аналітичних моделей формують нейронні мережі, до неструктурованих, які безпосередньо відтворюють нелінійні зв'язки між вхідними та вихідними сигналами. Прості прямонаправлені нейронні мережі можуть апроксимувати будь-які функції, що є шматково-безперервними, що робить їх універсальними для багатьох систем. Головні типи таких мереж - багаточаровий персептрон і мережі з радіально-симетричними функціями.

Тренування мережі може відбуватися онлайн під час регулювання або офлайн на основі передбачуваних тренувальних даних. Для цього існують

різноманітні методи, основні з яких - градієнтні методи, відомі в різних варіаціях у спеціальній літературі. Однак ця сфера досліджень залишається складною та абстрактною для неспеціалістів, що ускладнює їх загальне використання, особливо в промисловості.

Таким чином, використання нейромережевих технологій в системах управління, надає величезні можливості для пошуку ефективних рішень самих різних задач, багато з яких раніше не мали рішення із задовільною точністю. Тому тема роботи є актуальною.

Таким чином, використання нейромережевих технологій в системах управління, надає величезні можливості для пошуку ефективних рішень самих різних задач, багато з яких раніше не мали рішення із задовільною точністю. Тому тема роботи є актуальною.

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму підйому мостового крана для підвищення показників якості функціонування системи на основі нейромережевих регуляторів. Досягнення поставленої мети здійснюється рішенням наступних основних задач.

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

Об'єктом дослідження є динамічні процеси в системі управління електроприводом механізму підйому мостового крана з урахуванням пружних властивостей підйимального канату.

Предметом дослідження є моделі і методи синтезу нейромережевих систем управління механізмом підйому мостового крана з урахуванням пружних властивостей підйимального канату.

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 дало змогу оцінити ефективність роботи систем і підтвердити результативність отриманих даних.

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 підйому мостового крана.
2. Вперше запропоновано і синтезовано нейромережеву систему управління електроприводом механізму підйому мостового крана з урахуванням пружних властивостей підйимального канату, що має високі показники якості функціонування.
3. Вперше запропонована математична модель системи управління електроприводом механізму підйому мостового крана з урахуванням пружних властивостей підйимального канату 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 підйому мостового крана, 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 206 сторінки, серед них: 64 рисунки, з яких 40 рисунків за текстом, 24 рисунки на 22 окремих сторінках, 2 таблиці, 4 додатки на 43 сторінках, список літературних джерел з 58 найменування на 6 сторінках.

РОЗДІЛ 1

АНАЛІЗ СТРУКТУР СИСТЕМ НЕЙРОУПРАВЛІННЯ.

ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Галузі застосування штучних нейронних мереж

Термін "нейронні мережі" виник у 40-х роках минулого століття серед науковців, які досліджували структуру і функціонування біологічних нейронних мереж. Видатні досягнення в цій області пов'язані зі іменами американських вчених, таких як У. Маккалох, Д. Хебб, Ф. Розенблатт, М. Мінські, Дж. Хопфілд та інших [13-16].

Нейронні мережі - це обчислювальні структури, які моделюють прості біологічні процеси, що зазвичай асоціюються з функціонуванням людського мозку. Це системи, які можуть навчатися, аналізуючи позитивні та негативні зв'язки. Основною складовою таких мереж є штучний нейрон, або просто "нейрон", що отримав свою назву від біологічного прототипу

Сьогодні штучні нейронні мережі (ШНМ) активно застосовуються в різних сферах, від задач апроксимації функцій до створення нейрокомп'ютерів. Через різноманітність можливих застосувань нейронних мереж описати їх усі досить складно, тому ми зупинимося на окремих із них.

Апроксимація функцій.

Встановлення універсальних апроксимуючих властивостей ШНМ стало важливим етапом у формуванні загальної теорії нейронних мереж і стимулювало подальші дослідження в цій галузі. Різні дослідники довели, що нейронна мережа з одним прихованим і одним вихідним шаром здатна з довільною заданою точністю апроксимувати будь-яку неперервну функцію на компактній множині.

Зокрема, Р. Хехт-Нільсен [17,18], використовуючи теорему Колмогорова-Арнольда [19-21], що описує подання неперервних функцій декількох змінних у вигляді суперпозиції неперервних функцій однієї змінної, довів можливість

апроксимації функцій багатьох змінних загального вигляду за допомогою двох шарової нейронної мережі з прямими повними зв'язками, фіксованою кількістю нейронів і заданими обмеженими функціями активації.

Дж. Цибенкот і К. Хорник [22-24], спираючись на теорему Хана-Банаха, показали, що кінцева лінійна комбінація фіксованих одновимірних функцій може точно апроксимувати будь-яку неперервну функцію кількох дійсних змінних на заданому гіперкубі за умови достатньо м'яких обмежень на функції однієї змінної.

Асоціативна пам'ять.

Штучні нейронні мережі мають архітектури, які зберігають та асоціюють образи, що подаються на них, та в змозі викликати ці образи з пам'яті при введенні певного "асоціативного" образу. Ця властивість дозволяє здійснювати пошук інформації не за адресою, а за змістом. Навіть якщо зображення, що відтворюється, спотворене або перешкоджене, мережа може забезпечити правильний результат. В залежності від того, чи збігається шуканий образ з тим, що збережено в пам'яті, розрізняють авто- та гетероасоціативну пам'ять..

Стиснення пам'яті.

Деякі типи ШНМ мають унікальні властивості, які дозволяють використовувати їх для стиснення даних, наприклад, перед передачею, що зменшує обсяг переданої інформації. Подібні підходи застосовуються також у кластерному аналізі, де схожі за певними характеристиками об'єкти групуються в кластери. Це означає перехід від початкового n -вимірного простору ознак до m -вимірного простору кластерів, де $m < n$. Робота в просторі з меншою розмірністю значно економить обчислювальні ресурси та знижує потребу в обсязі пам'яті.

Використання ШНМ для стиснення даних у системах розпізнавання мовлення та зображень забезпечує коефіцієнт стиснення у 100 і більше разів.

Розпізнавання і класифікація/

Розпізнавання образів, таких як зображення, текст (друкований або рукописний), звук або мова, є однією з найбільш успішних сфер застосування ШНМ. Задачі розпізнавання ускладнені через високу розмірність вхідних да-

них, але використання ШНМ для стиснення інформації дозволяє знизити розмірність, зберігаючи при цьому можливість чіткого розділення різних класів.

Серед важливих завдань розпізнавання є класифікація кривих і геометричних фігур, наприклад, для діагностики чи виявлення несправностей. ШНМ демонструють високу ефективність у вирішенні таких завдань, незалежно від того, чи є наявна навчальна вибірка з класифікованими об'єктами. Однак, доступність навчальної вибірки значно прискорює процес пошуку оптимального рішення.

Оптимізаційні завдання/

Більшість практично важливих завдань можуть бути сформульовані як оптимізаційні, які доставляють екстремум деякому наперед вибраному критерію. Тут, проте, йтиметься тільки про одну таку задачу, задачу про комівояжера, яке має величезне практичне значення і рішення якої вимагає значних обчислювальних витрат. Це завдання полягає в тому, що комівояжер повинен відвідати задану кількість місць, вибравши для цього найкоротший маршрут, причому в кожному місці він повинен побувати не більше одного разу. Таке завдання виникає, наприклад, під час створення автоматичних технологічних ліній (свердлення отворів в друкованих платах, трасування друкованих плат і ін.), визначення оптимальних маршрутів перевезення, тощо.

Доведено, що це завдання відноситься до класу «NP-повних» (недетерміністськи поліноміальних), кращим методом рішення яких є повний перебір можливих варіантів. Оскільки у разі n місць існує $n!$ варіантів обходу, очевидно, що із збільшенням кількості місць складність рішення завдання різко зростає. Одне з можливих ефективних рішень цієї задачі за допомогою ШНМ, що вимагає значно менших обчислювальних витрат і при цьому сходиться з рішенням, отриманим повним перебором, було запропоновано в роботі [25].

Управління складними процесами/

Проблема синтезу ефективної системи управління є достатньо складною, оскільки реальні процеси характеризуються, як правило, нелінійними залежностями, високим рівнем шумів та їх корельованістю, які змінюються умовами

функціонування, що обумовлюють зміну характеристик досліджуваних об'єктів. Необхідність рішення задач в реальному часі висуває певні умови як до самих алгоритмів управління, які входять до складу математичного забезпечення проектованої системи, так і до технічних засобів, які їх реалізують. У цих умовах найбільш ефективними є методи і алгоритми, які базуються на теорії адаптації. Проте, як і при будь-якому підході, ці методи вимагають розробки математичних моделей досліджуваних об'єктів.

Слід зазначити, що отримані математичні моделі використовуються не тільки з метою безпосереднього управління, але і для попередження поведінки об'єкту, який дозволяє підвищити ефективність управління шляхом своєчасної корекції керованих параметрів. Якщо отримання математичної моделі ускладнене або вимагає істотних зусиль, доцільне застосування ШНМ.

Прогнозування/

Прогнозування (екстраполяція) базується на формалізованому уявленні про існуючий зв'язок між причинами та наслідками. Багато процесів формуються під впливом численних факторів, які часто мають різноспрямовану дію та нерідко залишаються невідомими. Статистичний аналіз таких процесів включає дослідження взаємозв'язків між факторами як у статичному стані, так і в динаміці. Для аналізу цих взаємозв'язків використовують часові ряди показників, що відображають розвиток об'єктів.

Найпоширенішим підходом до прогнозування є екстраполяція поточних зв'язків і закономірностей на майбутнє. Моделі прогнозування, побудовані за цим принципом, розрізняються переважно гіпотезами щодо конкретних типів збережених зв'язків. Чим загальніші припущення закладені в структуру моделі, і чим більше процесів вона здатна описати, тим ширшими є її можливості при дослідженні конкретних реалізацій.

Отже, вибір і обґрунтування математичної моделі є ключовим етапом у процесі прогнозування. На практиці нерідко виявляється, що унаслідок тих або інших причин отримати математичну модель, яка адекватно відбиває можливості досліджуваного об'єкту, незвичайно складно. І в цих випадках ефективним є

використання ШНМ. У ряді робіт приведені чисельні приклади успішного застосування ШНМ для вирішення завдань економічного, особливо, фінансового прогнозування.

Нейрокомп'ютери.

Нейрокомп'ютери є обчислювачами нового класу, які забезпечують швидше, дешевше і якісніше рішення конкретних завдань. Нейрокомп'ютер реалізує ідею створення аналого-цифрової ЕОМ, в якій аналогова частина виконує багатовимірні операції в граничному базисі, а цифрова реалізує алгоритми настройки параметрів нейронних мереж. Деякі порівняльні характеристики комп'ютера і ШНМ приведені в таблиці 1.1.

Таблиця 1.1 – Порівняльні характеристики комп'ютера і ШНМ

Параметр	Комп'ютер	ШНМ
Підхід до рішення задачі	Дедуктивний (використовує відомі правила перетворення вхідних даних у вихідні)	Індуктивний (задані вхідні і вихідні дані використовуються для конструювання правил)
Обчислення	Централізовані, синхронні, послідовні	Розподілені, асинхронні паралельні
Пам'ять	Що пакується, зосереджена і адресована по розташуванню	Розподілена, адресована за змістом
Стійкість до перебоїв	Нестійкий (вихід з ладу одного елементу приводить до виходу з ладу всього пристрою)	Стійка за рахунок надмірності і ділення функцій
Швидкодія	Високе (млн. оп./с)	Низька (тис. оп./с)
Точність	Висока	Невисока
Архітектура	Фіксована	Змінюється

Висока швидкодія нейрокомп'ютерів досягається за рахунок розпаралелювання обчислень. Кардинальним напрямом розвитку нейрокомп'ютерів, як загального призначення, так і проблемно-орієнтованих є розробка нейрочипів. Досягнення в мікроелектроніці вселяють упевненість в тому, що незабаром будуть створені сильні і в один час дешевші нейронні ЕОМ.

У додатках А-В 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 і структуру багат шарових нейронних мереж, а також властивості зазначених мереж

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

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

1.2 Огляд структур систем нейрорегулювання. Вибір структури системи

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

1.2.1 Послідовна схема нейроуправління

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

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

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

похідна яких обчислюється за виразом

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

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

Мета управління полягає в мінімізації квадратичної функції помилки

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

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

Вихідний сигнал мережі може бути обчислений таким чином:

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

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

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

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

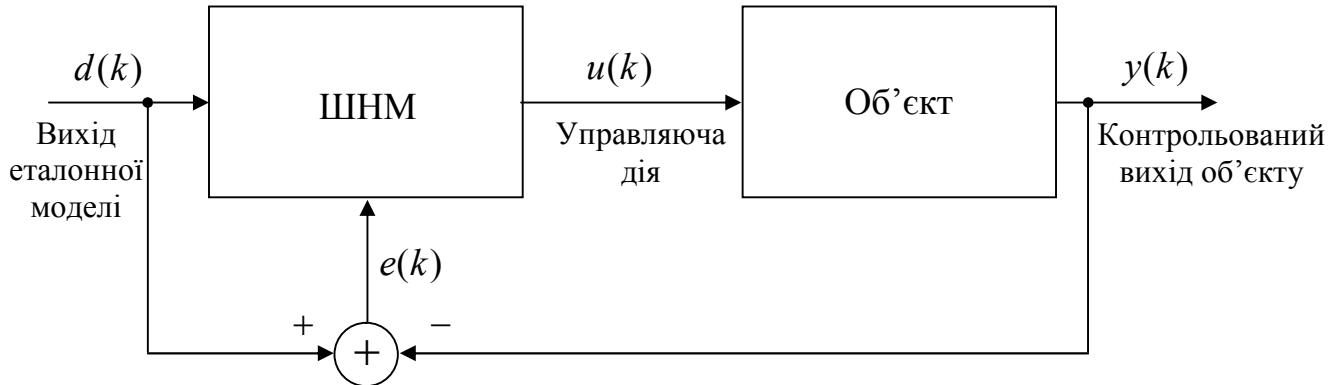


Рисунок 1.1 – Спеціалізоване нейроуправління

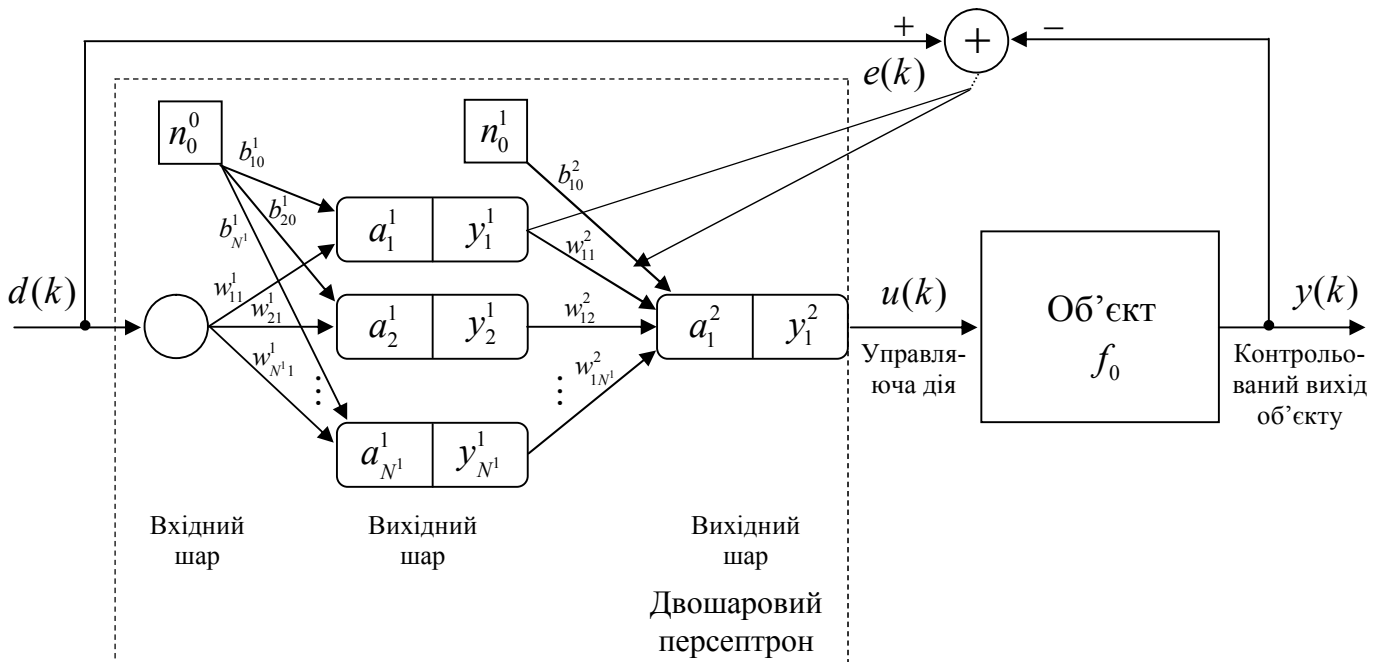


Рисунок 1.2 – Структура нейрорегулятора на основі двошарового персептрону

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

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

З (1.5) отримуємо

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

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

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

Тепер можемо записати загальну формулу настройки вагів вихідного шару

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

Аналогічно для нейронів прихованого шару, можемо записати

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

де

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

Таким чином, алгоритм настройки параметрів нейрорегулятора має вигляд:

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

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

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

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

де

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

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

1.2.2 Схема зворотного розповсюдження в часі

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

другому етапі функціонування системи – управлінні об'єктом. Використання нейроемулятора дозволяє обчислити Якобіан системи, замінюючи об'єкт його моделлю. В цьому випадку отримаємо:

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

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

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

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

Навчання нейрорегулятора виконується на підставі зворотного розповсюдження помилки через нейроемулятор. Дана схема управління також називається «Схема зворотного розповсюдження в часі» [26-30].

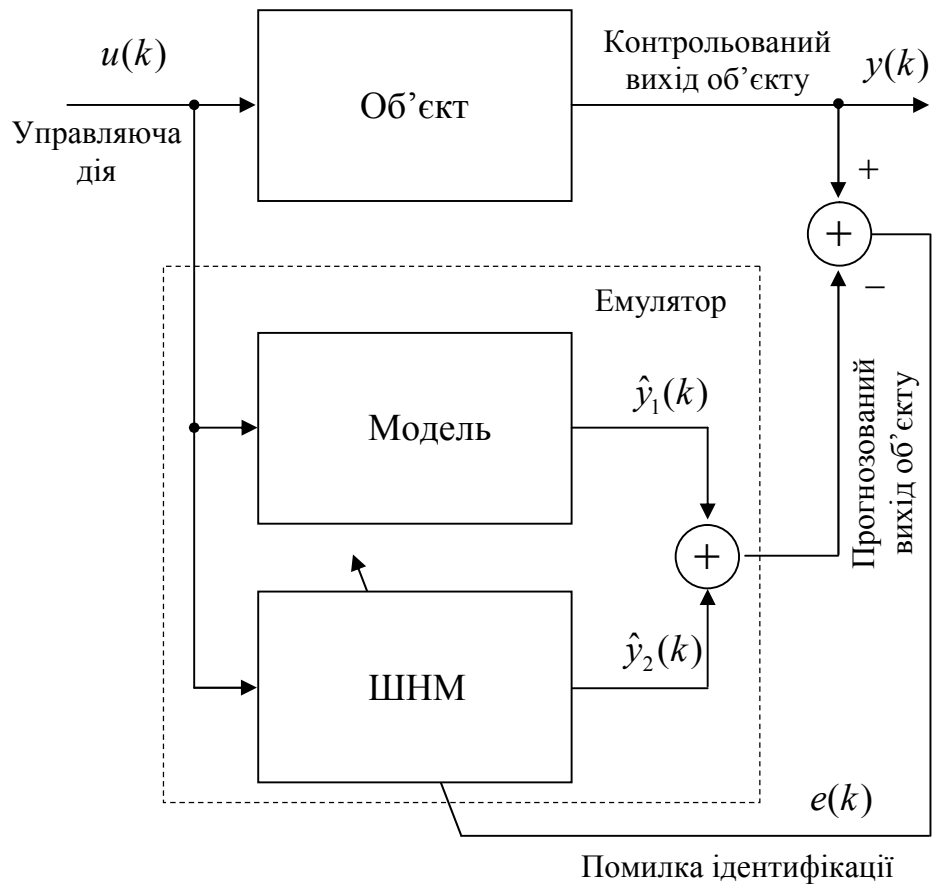


Рисунок 1.3 – Структура нейроемулятора з умовною моделлю

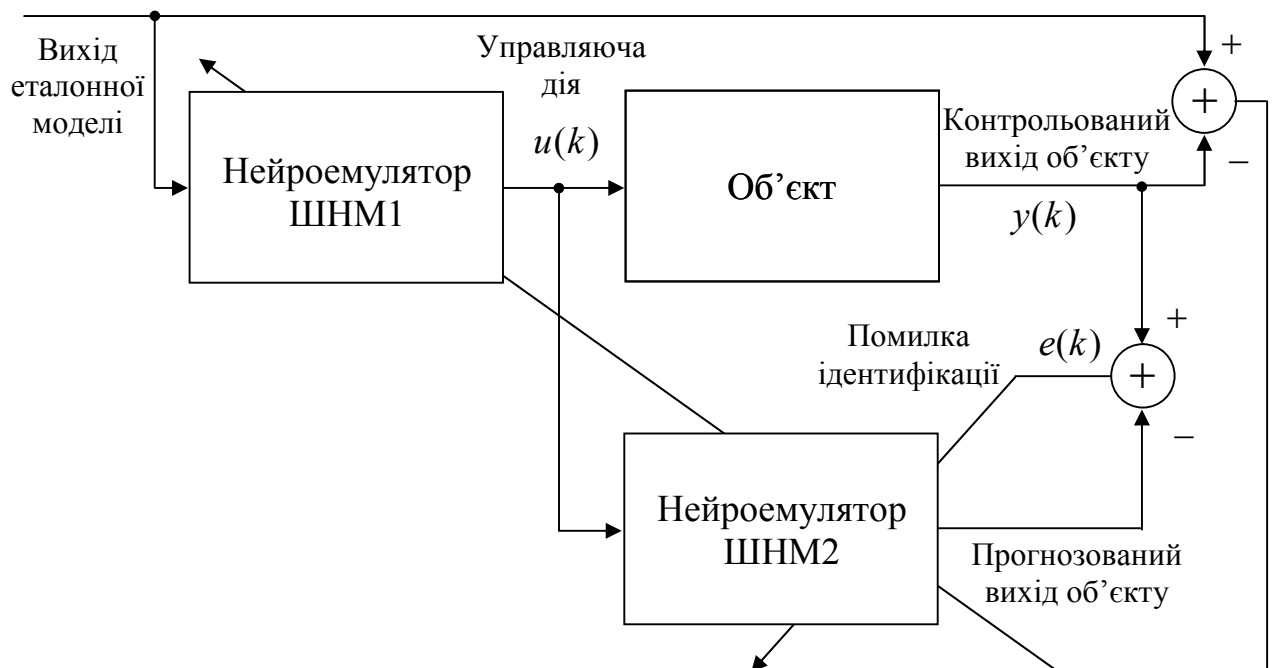


Рисунок 1.4 – Загальна схема інверсно-динамічного нейроуправління

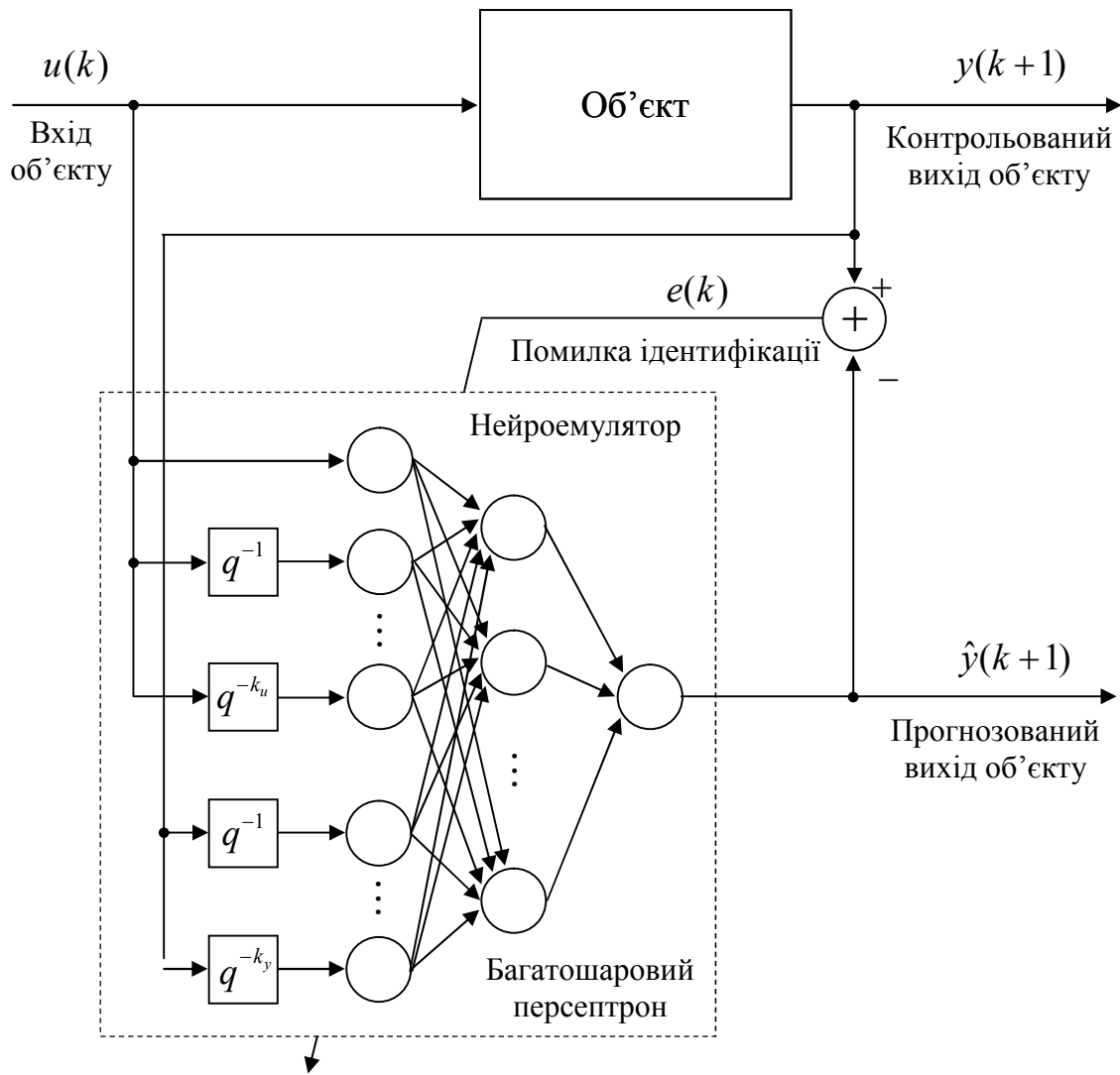


Рисунок 1.5 – Структура нейроемулятора

У роботі [31] була запропонована інша схема для реалізації інверсної динаміки об'єкту (рис.1.6). Тут нейроемулятор ШНМ2 навчається інверсній динаміці об'єкту управління, а нейрорегулятор ШНМ1 є копією мережі ШНМ2. Процедура копіювання моделі ШНМ2 відбувається після завершення її навчання. Навчання ШНМ2 інверсному відображенню виконується в автономному режимі або при подачі на вхід об'єкту управління випадкових вхідних сигналів. Як структура нейроемулятора можна використовувати модель рис.1.7.

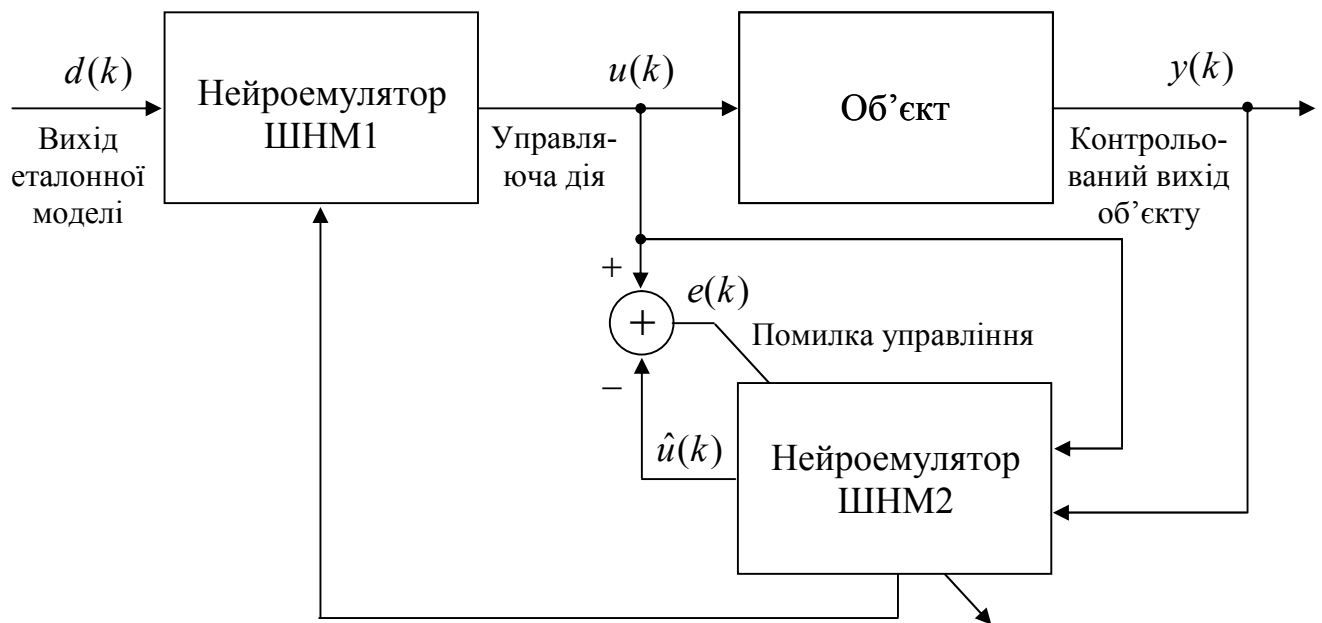


Рисунок 1.6 – Схема зворотного відображення

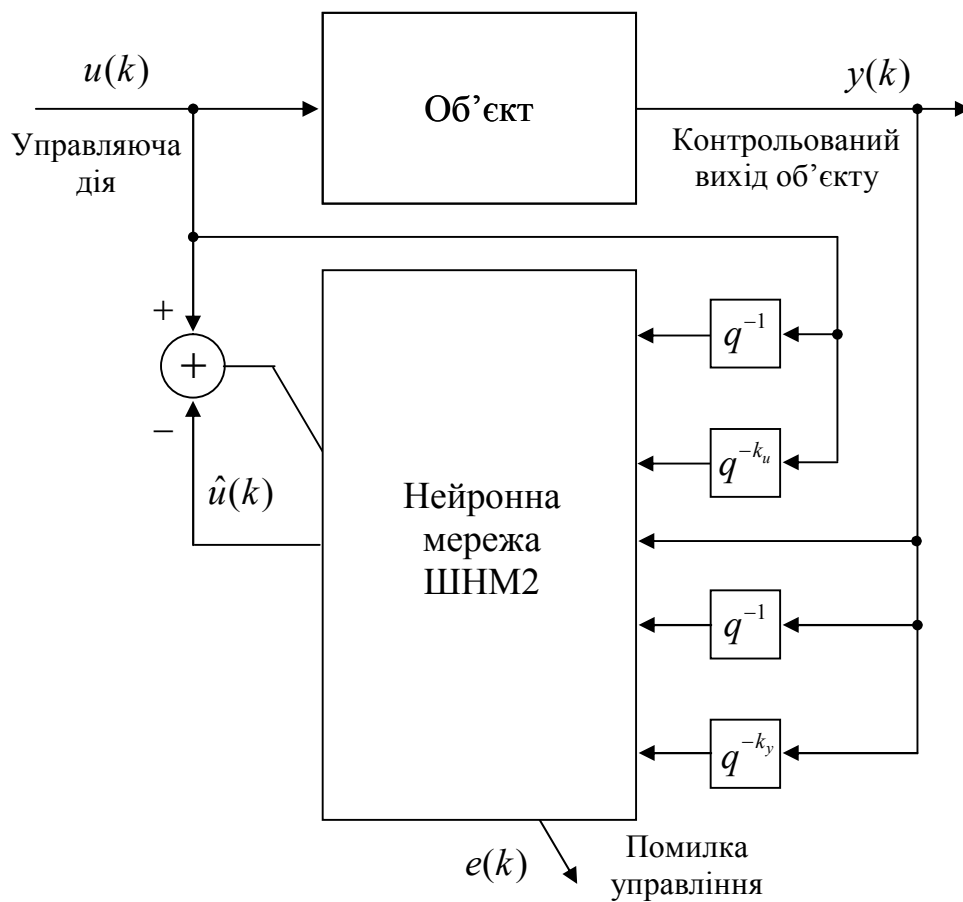


Рисунок 1.7 – Структура нейроемулятора для схеми зворотного відображення

1.2.3 Паралельне нейроуправління

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

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

Один з варіантів паралельної схеми нейроуправління, заснований на інверсній і прямій моделі, був запропонований в роботах [32,33] (рис.1.9).

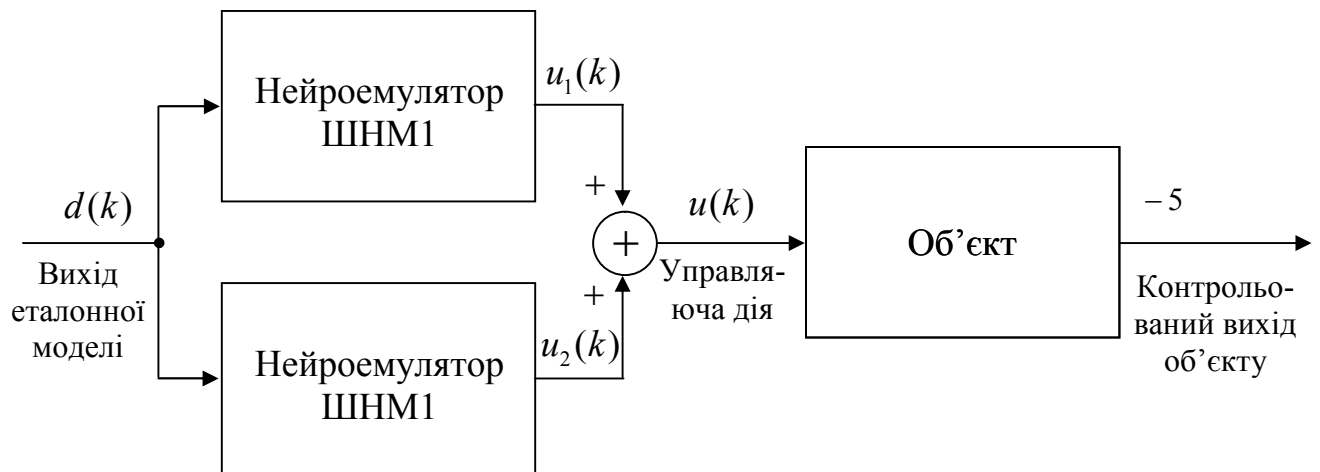


Рисунок 1.8 – Паралельна схема нейроуправління

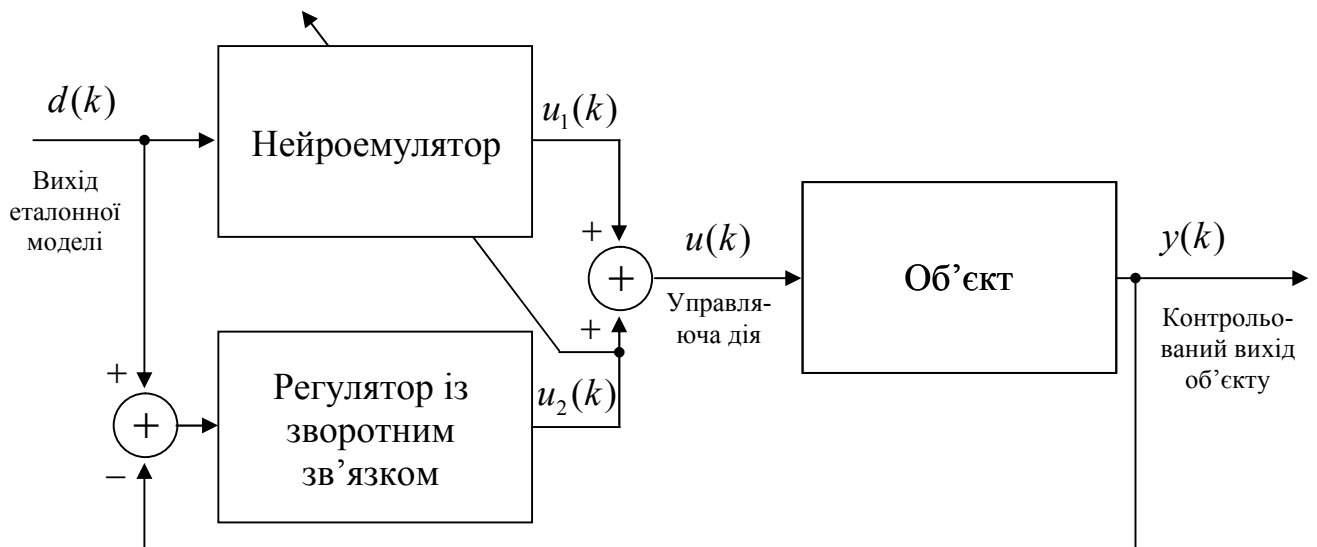


Рисунок 1.9 – Нейроуправління по помилці зворотного зв'язку

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

З метою узагальнення розглянемо модифікований варіант схеми нейроуправління по помилці зворотного зв'язку, представлений на рис.1.10 [34].

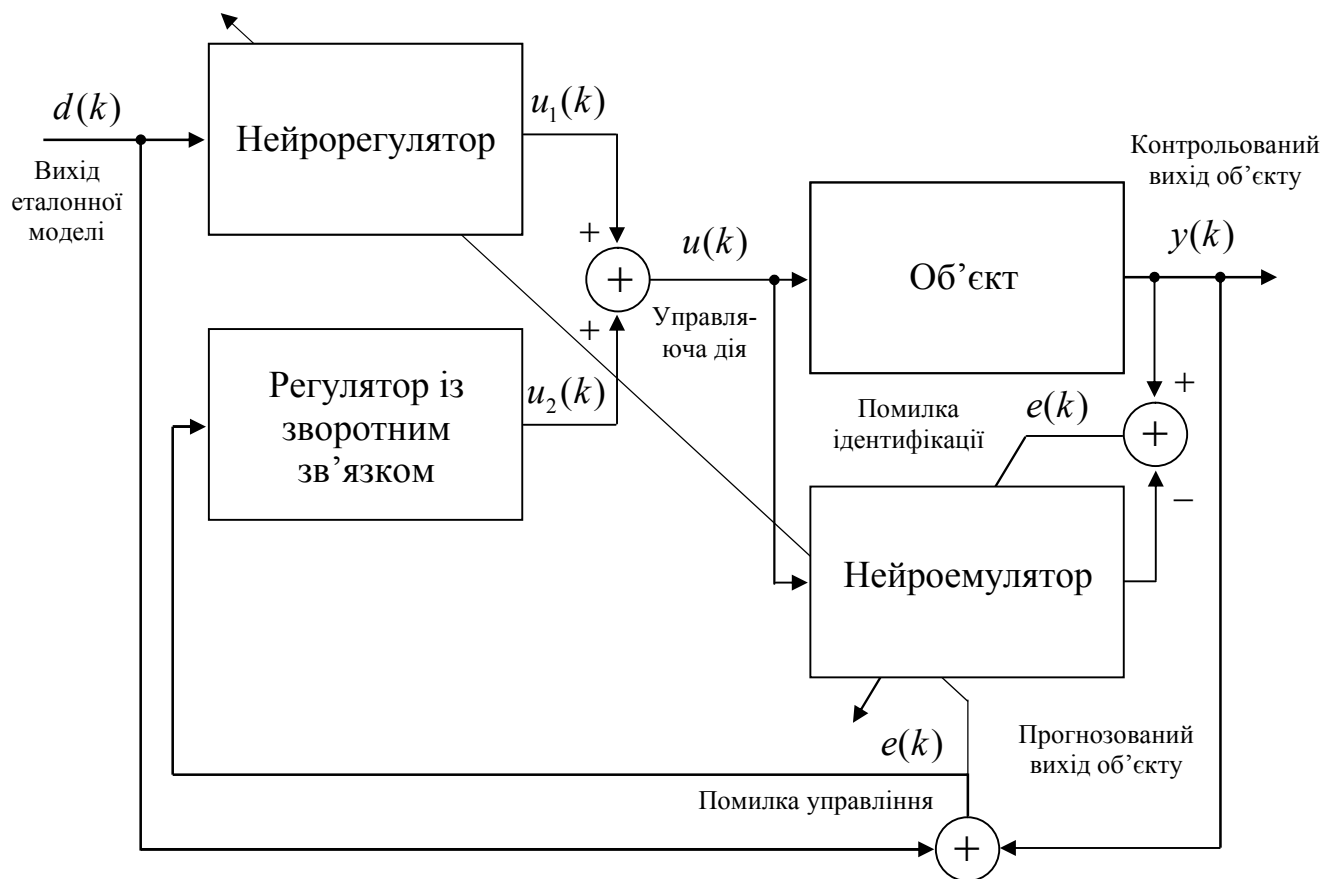


Рисунок 1.10 – Узагальнена схема нейроуправління по помилці зворотного зв'язку

Систему управління можна розділити на три частини. Перша частина показана жирними лініями і є традиційним контуром зворотного зв'язку, званим також зовнішнім зворотним зв'язком [35]. Управління по зворотному зв'язку засноване на помилці між виходом еталонної моделі і виходом об'єкту управління

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

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

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

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

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

Як критерій якості використовуємо функцію помилки (1.2), а для настройки вагів нейрорегулятора застосуємо метод найшвидшого спуску. Виберемо структуру нейрорегулятора як показано на рис.1.11.

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

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

а корекція вагів нейронів прихованого шару відповідно до правила:

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

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

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

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

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

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

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

де

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

Алгоритм настройки вагів нейрорегулятора для паралельної схеми управління має вигляд:

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

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

1.2.4 Пряме адаптивне нейроуправління

Розглянемо нелінійний об'єкт, описуваний NARX моделлю

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

де $y(k)$ – вихід об'єкту управління;

$u(k)$ – управляюча дія;

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

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

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

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

похідна якої обчислюється за виразом

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

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

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

Нейромережева модель об'єкту (НММ) має вид:

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

де $\hat{y}(k+1)$ – вихід ШНМ

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

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

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

де

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

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

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

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

де $\mathbf{W}^1$ – матриця вагових коефіцієнтів прихованого шару;
 w^2 – вектор вагових коефіцієнтів вихідного шару;
 b^1 – вектор вагових коефіцієнтів зсувів прихованого шару;
 b^2 – зсув вихідного шару.

Мінімум J може бути знайдений рекурсивно за допомогою методу найшвидшого спуску

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

де $\eta > 0$ – норма навчання.

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

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

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

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

Підставляючи (1.21) в (1.20), отримаємо

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

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

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

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

Тепер остаточно можна записати:

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

Таким чином, алгоритм прямого адаптивного управління має вигляд:

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

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

1.2.5 Управління з самонастройкою

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

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

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

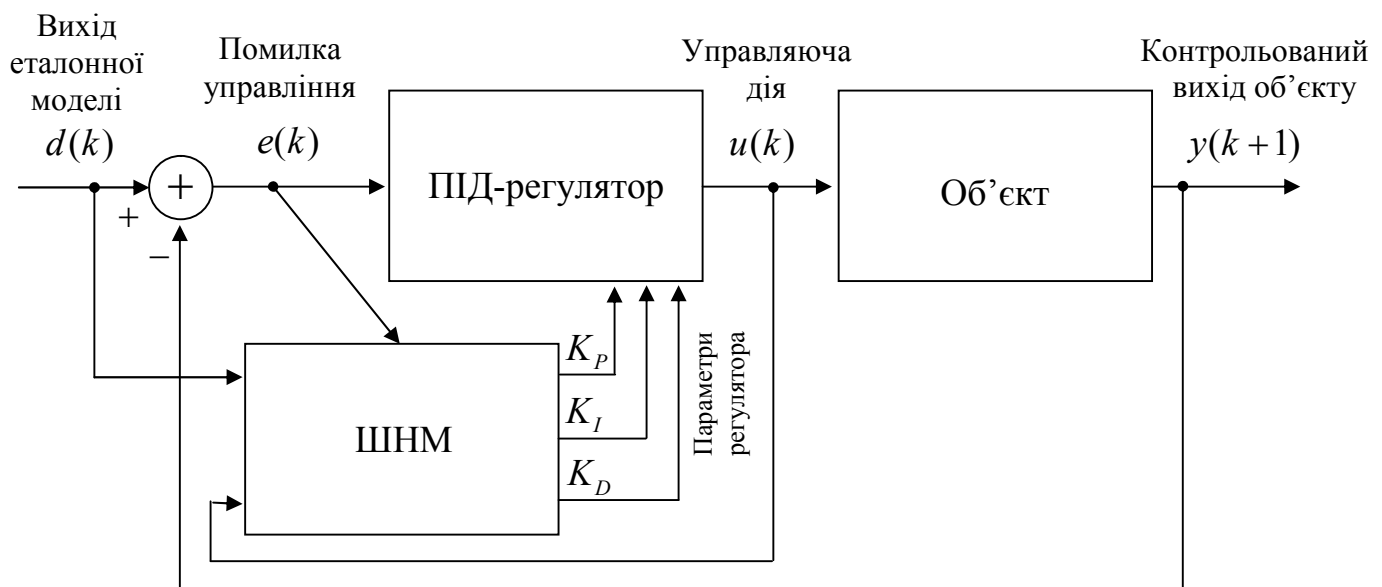


Рисунок 1.11 – Структура навчання ПІД-регулятора з само настройкою

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Оскільки правило функціонування нейронів вихідного шару має вигляд:

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

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

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

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

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

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

Тепер, ввівши позначення для локальної помилки вихідного шару

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

можемо записати правило настройки вагів нейронів вихідного шару

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

Перейшовши до прихованого шару, отримаємо

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

де δ_i^1 – локальна помилка прихованого шару

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

Таким чином, правило настройки вагів нейронів прихованого шару має вигляд

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

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

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

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

1.2.6 Нелінійне предиктивное управління

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

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

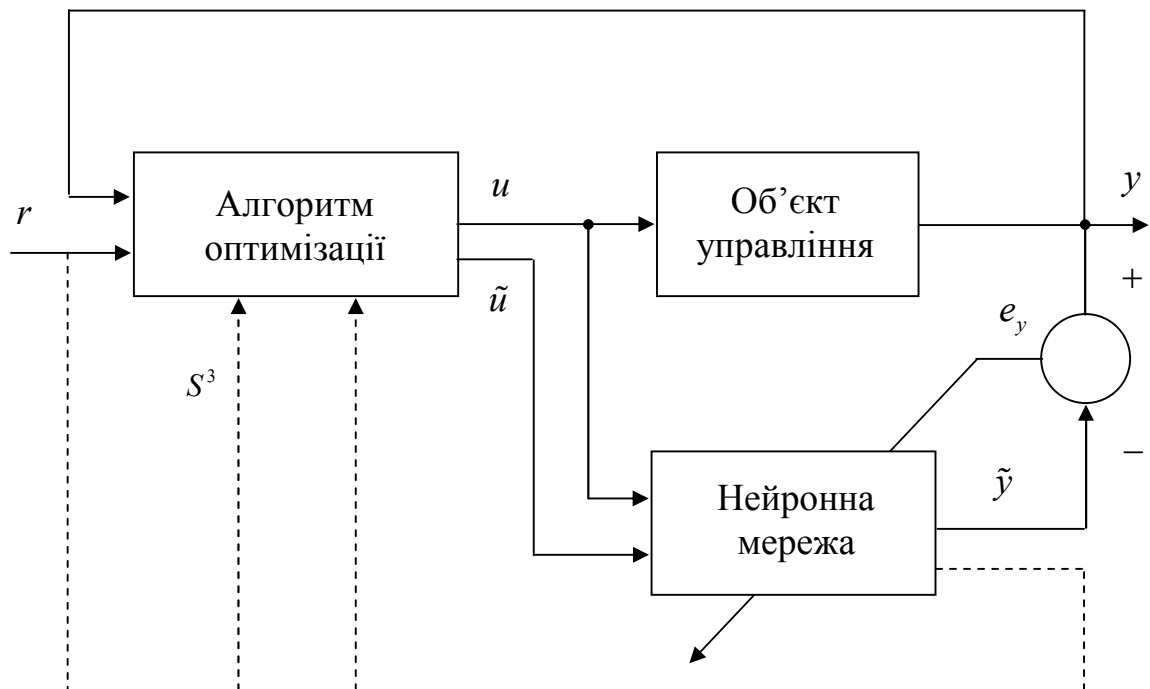


Рисунок 1.12 – Предиктивне регулювання

Регулятор в даному випадку є не що інше, як ітераційний алгоритм оптимізації [36]:

- мінімізація здійснюється чисельними методами оптимізації;
- мінімізується функціонал якості, що описує бажаний критерій якості;
- для дії на об'єкт використовується тільки перший елемент оптимізованої послідовності;
- вся процедура оптимізації повторюється на кожному кроці дискретності.

Перевага методу МРС – висока якість регулювання навіть у разі складних нелінійностей в об'єкті. Недоліком методу є дуже трудомісткі обчислення і високі вимоги до якості нелінійної моделі об'єкту. У даній магістерській роботі використовується нелінійне предиктивне регулювання.

1.3 Вимоги до систем управління електроприводами основних механізмів кранів

Мостові крани широко використовуються на підприємствах машинобудівної галузі та в інших секторах промисловості. Вони застосовуються для підйому, опускання і переміщення важких заготовок, деталей і вузлів машин по цеху в горизонтальному і вертикальному напрямках. Тип мостового крана зазвичай обирається залежно від особливостей цеху та його технологічних процесів, але багато конструктивних елементів крана, таких як механізми підйому і пересування, є стандартними для різних типів кранів. В залежності від типу вантажів, що транспортуються, використовуються різні вантажозахватні пристрої: крюки, магніти, грейфери, кліщі тощо. Найбільш поширеними є крани з підвіскою крюка. За чинними стандартами крани класифікуються за режимами роботи механічного і електричного обладнання на чотири категорії, які визначають інтенсивність використання, навантаження та умови експлуатації: Л – легкий режим, З – середній, Т – важкий, ВТ – дуже важкий режим. Основні показники, що характеризують режим роботи, включають тривалість роботи двигуна механізму ПВ, кількість його включень за годину та коефіцієнти використання механізмів за вантажопідйомністю.

Механізми кранів функціонують у режимі повторного короткочасного циклу з частими включеннями протягом години та в складних атмосферних умовах, тому для них розроблені спеціальні краневі та металургійні електродвигуни з підвищеною перевантажувальною здатністю. Основні механізми для переміщення вантажів у всіх типах кранів включають підйомні лебідки та системи пересування крана й візка. Загальний вигляд мостового крана зображено на рис. 1.13.

Несуча зварна конструкція крана складається з моста, який має дві головні балки коробчатого перетину (або гратчасті ферми), що розташовані через проліт цеху та кінцеві балки з ходовими колесами. Колеса пересуваються по рейках підкранового шляху, закріпленим на опорних балках у верхній частині

цеху. Привід колес здійснюється за допомогою електродвигунів, які працюють через редуктори. Уздовж моста розташовані рейки, по яких переміщується візок з підйомною лебідкою, що обертається завдяки електродвигуну через редуктор. Лебідка оснащена барабаном, на який намотується підйомний трос, прикріплений до крюка, що підвішений на блоках для захоплення вантажу. Барабан приводиться в обертання електродвигуном через редуктор.

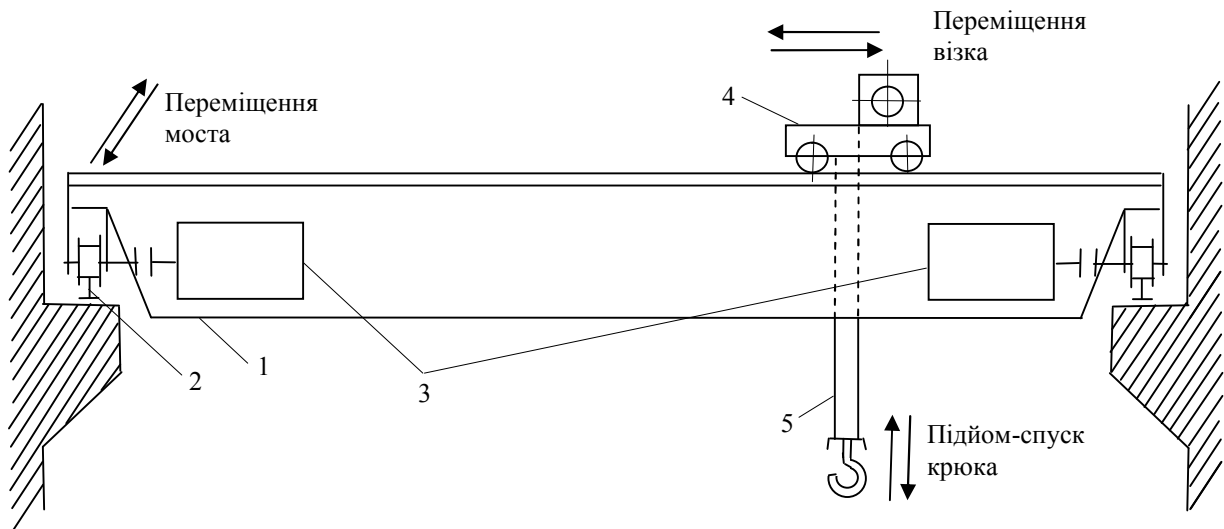


Рисунок 1.13 – Загальний вид мостового крана

1 – несуча ферма; 2 – рухомі опори; 3 – механізм пересування моста; 4 – механізм пересування візка; 5 – механізм підйому

Широке використання та складні умови роботи електроприводів кранів вимагають дотримання високих стандартів простоти експлуатації та надійності. Тому при проектуванні електроприводів кранів конструкторам важливо застосовувати максимально прості системи електроприводу та їх схеми управління, навіть якщо це передбачає деяке зниження вимог до статичних характеристик і динамічних властивостей приводів.

Оскільки підйомні крани управляються оператором на всіх етапах роботи, основною технологічною вимогою, що впливає на вибір системи електроприводу, є регулювання швидкості. Діапазон регулювання швидкості D визначається відношенням робочої швидкості до мінімальної, яка необхідна для норма-

льної роботи. Мінімальна швидкість для підйомних механізмів обмежується умовами м'якого встановлення вантажу без небажаних поштовхів. При управлінні механізмами пересування регулювання швидкості спрощує оператору точне позиціонування, зменшуючи необхідність у повторних активаціях приводу.

Умови установки вантажів та необхідна точність їх позиціонування залежать від призначення крана. Для монтажних кранів, які виконують точні операції, для кранів, що транспортують крихкі деталі, та в подібних випадках, при інших рівних умовах, потрібна нижча швидкість у порівнянні з кранами універсального призначення. Тому діапазон регулювання швидкості в спеціальних кранах має бути більшим і зростає з підвищенням робочої швидкості механізму.

Управління швидкістю електроприводу повинно здійснюватися з урахуванням обмежень на прискорення, струм та момент. Робочі швидкості основних механізмів більшості кранів є такими, що при нормованих допустимих середніх прискореннях загальний час перехідних процесів складає незначну частину загального циклу. У таких умовах регулювання моменту та відповідні перехідні процеси не мають суттєвого впливу на продуктивність крана, і точне дотримання оптимальних законів руху зазвичай не є обов'язковим. Проте під час проектування слід враховувати вплив розгойдування вантажів, кінематичних зазорів та інших факторів і за необхідності вживати додаткові заходи для обмеження динамічних навантажень.

1.4 Обґрунтування вибору системи електроприводу механізму підйому мостового крана

Система управління електроприводом є складним комплексом, що включає різноманітні компоненти. Вона охоплює перетворювач електричної енергії, пристрої управління для комутації струму в колі електродвигуна, засоби для ручного або автоматичного управління, а також елементи захисту електрообла-

днання та механізм, що впливає на пристрої відключення електроприводу. Усі електричні ланцюги можна класифікувати на наступні категорії:

1. Головні ланцюги, які забезпечують основний потік енергії електроприводу та живлення вантажопідйомних магнітів.

2. Ланцюги збудження, через які проходить струм збудження для електричних машин постійного струму, синхронних електричних машин змінного струму або електромагнітів гальмівних систем, а також струм для двигунів електрогідравлічних товкачів.

3. Ланцюги управління, що передають команди до комутаційних пристроїв головних ланцюгів і ланцюгів збудження від органів управління. В цих ланцюгах також здійснюється виконання певних послідовностей команд і перемикань згідно з заздалегідь заданою програмою.

4. Ланцюги сигналізації, які передають інформацію про стан комутуючих елементів головних ланцюгів і ланцюгів управління або про значення певних параметрів електроприводу та механізму оператора або контролюючому пристрою.

У системах електроприводу кранів використовуються два основні типи перетворювачів електричної енергії: електромашинні та статичні. Електромашинні перетворювачі включають дві або більше електричних машин, які перетворюють електроенергію від джерела живлення в енергію з регульованими параметрами (напруга, частота, струм). Статичні перетворювачі, в свою чергу, здійснюють перетворення електричної енергії за допомогою безконтактної комутації ланцюгів постійного або змінного струму за допомогою керованих і некерованих напівпровідникових пристроїв.

Система управління електроприводом включає контактні та безконтактні пристрої комутації в електродвигуні, перетворювач енергії та елементи управління, а також елементи захисту електричних ланцюгів. Контактні пристрої можна поділити на дві групи:

1. Апаратура, яка управляється оператором або виконавчим механізмом (наприклад, контролери та кінцеві вимикачі).

2. Апаратура, де комутація здійснюється за допомогою електромагнітних пристроїв (наприклад, контактори та реле).

Якщо контактні елементи апаратури, що приводиться в дію вручну, призначені для перемикання основних електричних ланцюгів, така апаратура називається силовим кулачковим контролером. Якщо ж ці елементи застосовуються для перемикання управляючих ланцюгів, вона є командоконтролером. Коли контактні елементи активуються через механізм, апаратура отримує назву кінцевих або путніх вимикачів. Послідовність відкриття та закриття контактів, що здійснюється за допомогою вала з кулачковими шайбами в залежності від його обертання, називається діаграмою включень. Ця діаграма, представлена у вигляді таблиці, відома як таблиця включень. Кілька контактів і реле, а також різні пристрої захисту, об'єднані в одному комплектному пристрої для управління електроприводом, називаються магнітними контролерами. Якщо в магнітному контролері процес перемикання відбувається без розриву електричного кола під напругою, він називається магнітним контролером з бездуговою комутацією.

Оцінка ефективності систем управління повинна включати мінімізацію витрат, до яких відносяться початкові інвестиції, витрати на обслуговування та ремонт, а також енергоспоживання під час запуску та зупинки кранів протягом їх експлуатації до капітального ремонту (приблизно 10 років). Вибір системи проводиться на основі найкращих економічних показників.

Якщо економічні показники для порівнюваних систем є подібними, додатково оцінюються габаритні характеристики та умови розташування електрообладнання. Оскільки розташування перетворювальних агрегатів або напівпровідникових перетворювачів на кранах може становити технічні труднощі, може виникнути потреба у використанні системи, яка не має оптимальних економічних показників.

Проаналізувавши основні характеристики роботи всіх наявних електроприводів для механізмів кранів, вибираємо систему ТП-Д.

1.5 Постановка задачі дослідження і шляхи її вирішення

Аналіз наявної літератури показує, що існує багато різних підходів до вирішення проблеми синтезу систем управління, але універсального та ідеального регулятора на сьогоднішній день не створено. При розробці систем управління для механізму підйому мостового крана необхідно враховувати пружність підйомного канату. Як зазначено в главі 3, цей фактор викликає коливальний характер перехідних процесів, що негативно впливає на роботу механізму. Звичайні регулятори не забезпечують прийнятних показників якості роботи системи

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

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

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

Штучні нейронні мережі є перспективною альтернативою класичним методам створення систем управління нелінійними об'єктами. Нейронні мережі – це галузь штучного інтелекту, де для обробки сигналів використовуються процеси, схожі на ті, що відбуваються в нейронах живих організмів. Основною особливістю нейронної мережі, яка підкреслює її потужність і великий потенціал, є паралельна обробка інформації всіма елементами. Завдяки значній кількості міжнейронних зв'язків процес обробки даних значно прискорюється, що дозволяє здійснювати обробку сигналів у реальному часі. Крім того, велика кількість з'єднань робить мережу стійкою до помилок на окремих лініях, оскільки функції пошкоджених зв'язків можуть виконувати інші, працездатні лінії, завдяки чому діяльність мережі залишається стабільною. Іншою важливою властивістю є здатність до навчання та узагальнення накопичених знань. Нейронна мережа має властивості штучного інтелекту: навчившись на обмеженій кількості даних, вона може узагальнювати отриману інформацію та показувати хороші результати на нових даних, які не використовувались під час навчання.

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

Сучасні системи управління нелінійними динамічними об'єктами все частіше використовують штучні нейронні мережі, такі як багатошаровий персептрон, радіально-базисні мережі та нейро-фаззі мережі. Ці структури мають здат-

ність навчатися будь-яким функціям, проте для навчання потрібен значний обсяг інформації у форматі «вхід-вихід» (обсяг інформації має бути значно більшим за кількість зв'язків у мережі), а також правильний вибір нейромоделі, що включає визначення кількості нейронів і шарів у мережі та типу функцій активації нейронів. Для моделювання нелінійних залежностей між входом і виходом у нейронних мережах використовуються сигмоїдальні функції активації або інші нелінійні функції загального вигляду, зокрема радіально-базисні функції. Ці властивості нейронних мереж використовуються для вирішення задач управління з помітними нелінійностями, для яких традиційні методи не забезпечують практичних рішень.

Необхідною Традиційні методи оптимального та адаптивного управління потребують великого обсягу апіорної інформації для їхнього застосування. У свою чергу, штучні нейронні мережі мають здатність до самонавчання, що дозволяє використовувати нейрорегулятори для систем управління без необхідності такого обсягу інформації. Це забезпечує ефективне застосування нейрорегуляторів навіть у умовах значних невизначеностей.

Створення системи управління на основі нейрорегулятора включає в себе послідовне вирішення задачі ідентифікації об'єкта, а потім розробку алгоритму управління відповідно до поставлених цілей і критеріїв якості управління. Процес ідентифікації об'єкта полягає у створенні його нейромоделі, що здійснюється під час навчання мережі за допомогою навчальних пар, які формуються на основі виміряних значень вхідних та відповідних вихідних змінних.

Метою роботи є розробка нейромережевої системи управління механізмом підйому мостового крана з урахуванням пружних властивостей підймального канату.

Для досягнення поставленої мети необхідно вирішити наступні завдання.
задач.

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

2. Розробити математичну моделі системи управління з урахуванням пружних властивостей підйимального канату, провести моделювання багатомасової системи на ЕОМ і виконати аналіз отриманих результатів.

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

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

1. У результаті аналізу літературних джерел за темою дослідження обґрунтовано використання одного з класів штучних нейронних мереж - багатошарових нейронних мереж прямого поширення - для задач управління. Існуючі методи синтезу систем автоматичного управління дозволяють створювати системи з високими показниками регулювання, однак їх практична реалізація пов'язана з рядом технічних складнощів. Серед них - необхідність вимірювання координат, які важко визначити, наприклад, пружного моменту, а також обмеження, що заважають реалізації синтезованих параметрів через специфічні особливості реальних електроприводів. У зв'язку з цим розглядаються методи, які дозволяють замінити лінійні регулятори на нелінійні, що за меншої кількості вимірюваних сигналів забезпечують потрібну вихідну дію управління для об'єкта регулювання. Одним із таких рішень є нелінійний регулятор, реалізований у вигляді нейрорегулятора, що функціонує за принципами біологічних нейронних мереж.

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

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

4. Для оптимізації роботи об'єкта необхідно коригувати параметри регуляторів відповідно до зміни робочих умов. У зв'язку з цим доцільною є розроб-

ка систем управління, що базуються на адаптивному підході, у поєднанні з методами теорії ШНМ. Це дозволяє створювати прості нейромережеві моделі та адаптивно коригувати параметри моделей і регулятора відповідно до змінюваних умов.

5. У результаті аналізу сучасних структур систем нейрорегулювання обрано принцип нелінійного предиктивного регулювання для побудови нейромережевого регулятора системи.

6. Проведений аналіз існуючих методів опису нелінійних динамічних об'єктів та розглянуті принципи побудови нейромережевих NARMAX і NARX моделей. Моделі NARX мають лінійну параметризацію і використовують стандартні методи навчання, що робить їх найбільш застосовуваними при нейронній ідентифікації систем.

7. Проведено аналіз вимог до систем управління електроприводами основних механізмів кранів. Оскільки електроприводи кранів працюють у складних умовах і мають широке застосування, до них висуваються високі вимоги щодо простоти експлуатації та надійності. Основною технологічною вимогою, що впливає на вибір системи електроприводу, є здатність до регулювання швидкості, оскільки підйомні крани управляються оператором на всіх етапах роботи. Управління швидкістю повинно здійснюватися при дотриманні обмежень за прискоренням, струмом і моментом. При проектуванні важливо враховувати вплив розгойдування вантажів, кінематичних зазорів тощо, а за потреби — вживати додаткові заходи для обмеження динамічних навантажень.

8. Сформульовано завдання дослідження, яке полягає у розробці нейромережевої системи управління багатомасовою електромеханічною системою підйомного механізму мостового крана, що забезпечує високу якість регулювання.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ МОСТОВОГО КРАНА

2.1 Технічні характеристики мостового крана. Вимоги до системи управління

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

Таблиця 2.1 – Технічні дані мостового крана

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

Вимоги до показників якості функціонування системи управління.

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

2.2 Розрахунок потужності електроприводу механізму підйому. Вибір приводного двигуна

Для вибору двигуна по потужності скористаємося методикою [40]. Дана методика вибору потужності двигунів кранів, враховує як параметри режиму роботи механізмів, так і енергетичні властивості конкретних систем регулювання, яка застосовна для всіх використовуваних в даний час типів електроприводів.

Розраховуємо статичну потужність на його валу при підйомі номінального вантажу:

$$P_p = P_{ст.н} = \frac{9,81(Q + m_0)V_n}{10^3 m_n \eta}, \quad (2.1)$$

де Q – маса вантажу, що піднімається, кг;

m_0 – маса вантажозахватного пристосування, кг;

V_n – номінальна швидкість підйому вантажу, м/с;

η – ККД канатної системи і механізму при підйомі номінального вантажу;

m_n – коефіцієнт числа механізмів, що піднімають грузнув: для кранів крюків з однією лебідкою підйому $m_n = 1$.

Значення η відповідно до [40] приймаємо рівним 0,82.

$$P_p = P_{ст.н} = \frac{9,81 \cdot (41 \cdot 10^3 + 1 \cdot 10^3) \cdot 0,12}{10^3 \cdot 1 \cdot 0,82} = 60,29 \text{ кВт}$$

Для правильного вибору двигуна згідно рекомендаціям [40] визначаємо розрахункову потужність по тепловому режиму. По таблиці 6.5 [40] знаходимо значення коефіцієнта $k_t = 1,3$ (значення коефіцієнта k_t визначається виходячи з режиму роботи і прийнятої системи електроприводу). Відповідно до 6.44 [40] потужність двигуна по тепловому режиму рівна

$$P_{н.т} = P_{ст.н} / k_{т}, \quad (2.2)$$

$$P_{н.т} = 60,29 / 1,3 = 46,38 \text{ кВт}$$

Вибір двигуна здійснюємо по більшій з потужностей, тобто по величині P_p (оскільки $P_{н.т} < P_p$).

По каталогу вибираємо двигун Д814 з наступними технічними характеристиками. номінальна потужність $P_n = 66 \text{ кВт}$; номінальна напруга $U_n = 440 \text{ В}$; номінальний струм $I_n = 168 \text{ А}$; номінальна частота обертання $n_n = 560 \text{ об / хв}$; тривалість включення ПВ=40%; момент інерції $J_{дв} = 10,2 \text{ кг} \cdot \text{м}^2$; $N = 310$; $2a = 8$; $2p = 4$; $R_{я} + R_{дп} = 0,052 \text{ Ом}$; $R_{ко} = 0$; $\Phi_n = 8,33 \cdot 10^{-2} \text{ Вб}$.

Після попереднього вибору двигуна, необхідно здійснити перевірку по нагріву і перевантажувальній здатності. Найбільшу складність представляє розрахунок теплового режиму двигуна. Загальноприйняті методи теплового розрахунку по еквівалентних параметрах навантаження (струму, моменту, потужності) або середніх втратах дають достовірні результати тільки тоді, коли достатньо точно відома діаграма навантаження роботи електродвигуна. Для електроприводів кранів в більшості випадків характерною є невизначеність режиму роботи, що при специфічних особливостях машин кранів закритого виконання з підвищеними постійними втратами і погіршеними умовами тепловіддачі приводить до великих погрешностей при традиційних методах розрахунку.

У [40] розроблений метод вибору двигунів кранів, що враховує приведені вище особливості роботи електроприводів кранів. Цей метод, доповнений традиційними методами вибору по еквівалентних величинах при заданій діаграмі вантажного циклу, рекомендований до застосування при розрахунках і виборі електроустаткування крана.

Сумарний момент інерції механізму

$$J_{\Sigma} = 1,2J_{дв} + \frac{91(Q + m_0) \cdot V_{п}^2}{n_n^2}, \quad (2.3)$$

$$J_{\Sigma} = 1,2 \cdot 10,2 + \frac{91 \cdot (41 \cdot 10^3 + 1 \cdot 10^3) \cdot 0,12^2}{560^2} = 12,24 + 0,18 = 12,42 \text{ кг} \cdot \text{м}^2$$

Енергетичні властивості електроприводу крана можуть бути охарактеризовані еквівалентним ККД $\eta_{\text{екв}}$, визначеним як відношення корисної роботи переміщення вантажу до споживаної енергії за цикл

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

де $\eta_{\text{екв.б}}$ – еквівалентний ККД, визначається залежно від числа пусків $N_{\text{а}}$. Значення $N_{\text{в}} = 120$ (визначається з таблиці 1.1 [40] для режиму роботи 4 [40]), тоді $\eta_{\text{екв.б}} = 0,93$

$\eta_{\text{екв.Н}}$ – базовий ККД, визначається по таблиці 6.5 [40]; $\eta_{\text{екв.б}} = 0,91$ для тиристорного електроприводу постійного струму.

n_{max} – максимальна частота обертання, приймаємо $n_{\text{max}} = 2 \cdot n_{\text{н}}$

$$\eta_{\text{екв}} = \frac{0,93}{1 + \frac{0,93 - 0,91}{0,91} \cdot \frac{8,51}{1,2 \cdot 7} \left(\frac{1120}{1000} \right)^2} = 0,931.$$

Перевіряємо двигун по тепловому режиму по (6.43) [40]

$$P_{\text{н.т}} = \frac{k_3 k_{\text{екв}} \sqrt{\varepsilon_{\text{р}} / \varepsilon_{\text{ном}}} \cdot k_{\text{н}}}{k_0 k_{\text{р}}} \cdot \frac{\eta_{\text{екв.б}}}{\eta_{\text{екв.0}} - k_{\text{дин}} (\eta_{\text{екв.0}} - \eta_{\text{екв}})} \cdot P_{\text{ст.н}}, \quad (2.5)$$

Тут: k_3 – коефіцієнт, що враховує запас потужності, що докладається при проектуванні електроприводу, виходячи з вимог його надійності: $k_3 = 1$ з таблиці 6.4 [40];

$k_{\text{екв}}$ – коефіцієнт еквівалентності моменту статичного навантаження $k_{\text{екв}} = 0,77$ з таблиці 6.4 [40];

$k_p = 1$ і $k_0 = 1$ згідно рис. 6.12 [40];

$k_{\text{дин}} = 1,25$ – динамічний коефіцієнт; по таблиці 6.5 [40];

$k_n = 1$ відповідно до (6.41) [40];

ε_p – реальна тривалість включення.

Для визначення ε_δ обчислюємо час циклу

$$T_{\text{ц}} = \frac{Q}{C}, \quad (2.6)$$

$$T_{\text{ц}} = \frac{Q}{C} = \frac{41 \cdot 3600}{240} = 615 \text{ с}.$$

Час підйому:

$$t_{\text{п}} = \frac{H_{\text{п}}}{V_{\text{п}}}, \quad (2.7)$$

де $H_{\text{п}}$ – середня висота підйому, м;

$$t_{\text{п}} = \frac{8}{0,12} = 66 \text{ с},$$

$$\varepsilon_p = \frac{4t_{\text{п}}}{T_{\text{ц}}}, \quad (2.8)$$

$$\varepsilon_p = \frac{4 \cdot 66}{615} = 0,43,$$

$$P_{\text{н.т}} = \frac{1 \cdot 0,7 \sqrt{0,41 / 0,4} \cdot 1}{1 \cdot 1} \cdot \frac{0,94}{0,94 - 1,25(0,94 - 0,93)} \cdot 67,29 = 55,7 \text{ кВт}.$$

Оскільки виконується співвідношення $P_{\text{н}} > P_{\text{н.т}}$, то двигун задовольняє умовам нагріву, тобто вибраний правильно.

2.3 Вибір основного електроустаткування електроприводу і розрахунок параметрів силового кола

Для електроприводу механізму підйому вибираємо тиристорний перетворювач ТПЕ-200/400-460. Номінальний випрямлений струм $I_{dн} = 200 \text{ А}$, номінальна випрямлена напруга $U_{dн} = 460 \text{ В}$.

Перетворювач отримує живлення від мережі 380 В, 50 Гц через токообмежуючий реактор. Схема випрямлення трифазна мостова з двома стрічно-паралельними групами. Управління групами роздільне. До складу ТП входять також регульований випрямляч для ланцюгів збудження, зібраний по однофазній мостовій схемі.

Вибираємо токообмежуючий реактор типу РТСТ – 420 – 0,084 УЗ з наступними технічними характеристиками: номінальна лінійна напруга живлячої мережі 410 В, номінальна сила фазного струму $I_{\text{торн}} = 420 \text{ А}$ (для трифазної мостової схеми випрямлення діюче значення струму реактора рівне 0,85 випрямленого струму); номінальна індуктивність фази $L_{\text{тор}} = 0,064 \cdot 10^{-3} \text{ Гн}$, активний опір обмоток $R_{\text{тор}} = 0,0021 \text{ Ом}$. Індуктивний опір реактора відповідає напрузі короткого замикання U_k (для номінального струму), рівній 5,5%.

Вибір згладжуючого реактора здійснюється виходячи з двох умов: обмеження до допустимого для двигуна рівня пульсацій струму і обмеження зони переривистих струмів.

Необхідне значення індуктивності якірного кола для обмеження рівня пульсацій $i_e = 5 \div 10\%$ визначається по формулі:

$$L_d = \frac{I_n E_{d0}}{i_e \omega_1 I_n}, \quad (2.9)$$

де I_n - струм навантаження перетворювача, рівний номінальному струму двигуна;

ω_1 - кутова частота першої гармоніки випрямленої напруги

$$\omega = 2\pi m f, \quad (2.10)$$

m - число фаз перетворювача, для трьохфазної мостової схеми $m = 6$

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

E_{d0} – максимальне значення випрямленої ЕРС при куті управління тиристорами $\alpha = 0$.

$$E_{d0} = \frac{m}{\pi} \sqrt{2} U_{\text{л}} \sin \frac{\pi}{m}, \quad (2.11)$$

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

$U_{\text{л}}$ – діюче значення лінійної напруги живлення $U_{\text{л}} = 380 \text{ В}$

$$E_{d0} = \frac{6}{\pi} \sqrt{2} \cdot 380 \sin \frac{3,14}{6} = 514 \text{ В}$$

$l_{\text{п}}$ - відносна величина першої гармоніки випрямленої напруги, значення $l_{\text{п}}$ для граничного кута регулювання α_{max} можна визначати з виразу для електромеханічної характеристики електроприводу

$$\omega = \frac{E_{d0} \cos \alpha - \Delta U_{\text{в}} - R_{\Sigma} I_{\text{д}}}{c} \quad (2.12)$$

Для мінімальної швидкості обертання робочого механізму ω_{min} . Приймаємо $\omega_{\text{min}} = 0,1 \omega_{\text{н}}$, де $\omega_{\text{н}}$ - номінальна кутова частота обертання двигуна

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

$$\omega_{\text{н}} = \frac{3,14 \cdot 560}{30} = 58,6 \text{ c}^{-1}$$

$$\omega_{\min} = 0,1 \cdot 58,6 = 5,86 \text{ c}^{-1}$$

$\Delta U_{\text{в}}$ - падіння напруги в провідящий період на тиристорах одного плеча моста, приймаємо $\Delta U_{\text{в}} = 1,5 \text{ В}$.

$I_{\text{д}}$ - струм електродвигуна, приймаємо $I_{\text{д}} = I_{\text{н}} = 168 \text{ А}$;

R_{Σ} - сумарний активний опір якірного ланцюгу (визначаємо R_{Σ} без урахування активного опору згладжуючого реактора, оскільки питання про доцільність його застосування в приводі, що розробляється, ще не вирішене)

$$R_{\Sigma} = R_{\text{як}} + R_{\text{е.пр}}, \quad (2.14)$$

$R_{\Sigma\text{д}}$ - активний опір якірного кола двигуна:

$$R_{\text{як}} = k_t (R_{\text{я}} + R_{\text{дп}} + R_{\text{ко}}), \quad (2.15)$$

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

$R_{\text{я}}$, $R_{\text{дп}}$, $R_{\text{ко}}$ - активні опори якоря, додаткових полюсів і компенсаційної обмотки відповідно

$$R_{\text{як}} = 1,36 \cdot 0,052 = 0,0707 \text{ Ом};$$

$R_{\text{е.пр}}$ - еквівалентний опір тиристорного перетворювача

$$R_{\text{е.пр}} = R_{\text{тор}} + R_{\text{е}}, \quad (2.16)$$

$R_{\text{тор}}$ - активний опір токообмежуючого реактора;

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

$$R_{\text{е}} = m f L_{\text{тор}}, \quad (2.17)$$

f – частота живлячої мережі;

$$R_e = 6 \cdot 50 \cdot 0,064 \cdot 10^{-3} = 0,0846 \text{ Ом},$$

$$R_{e.\text{пр}} = 0,0021 + 0,084 = 0,0861 \text{ Ом},$$

$$R_{\Sigma} = 0,0707 + 0,0861 = 0,1568 \text{ Ом}.$$

c - постійна проти ЕРС

$$c = \frac{pN}{2\pi a} \Phi_{\text{н}}, \quad (2.18)$$

де p – число пар полюсів;

N – число активних провідників обмотки якоря;

a – число пар паралельних гілок;

$\Phi_{\text{н}}$ - номінальний магнітний потік на полюс

$$c = \frac{2 \cdot 310}{2 \cdot 3,14 \cdot 1} \cdot 8,33 \cdot 10^{-2} = 7,3.$$

З виразу (2.12) визначаємо α_{max}

$$\alpha_{\text{max}} = \arccos \left(\frac{\omega_{\text{min}} \cdot C + \Delta U_{\text{в}} + R_{\Sigma} \cdot I_{\text{н}}}{E_{\text{д0}}} \right).$$

$$\alpha_{\text{max}} = \arccos \left(\frac{5,86 \cdot 7,3 + 1,5 + 0,1568 \cdot 168}{513} \right) = 82,1^\circ.$$

Значення $l_{\text{п}}$, відповідне $\alpha_{\text{max}} = 81,1^\circ$ рівне 0,227.

Необхідна індуктивність силового кола для згладжування пульсацій

$$L_{\text{д}} = \frac{0,227 \cdot 476}{5 \cdot 10^{-2} \cdot 1884 \cdot 168} = 0,0046 \text{ Гн}$$

Індуктивність обмотки якоря двигуна:

$$L_{\text{як}} = k \cdot \frac{U_{\text{н}}}{I_{\text{н}}} \cdot \frac{1}{\omega_{\text{н}} p}, \quad (2.19)$$

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

$$L_{\text{як}} = 0,6 \cdot \frac{440}{168} \cdot \frac{1}{58,6 \cdot 2} = 0,00234 \text{ Гн}$$

Сумарна індуктивність силового кола електроприводу:

$$L_{\Sigma} = L_{\text{як}} + 2L_{\text{тор}}, \quad (2.20)$$

$$L_{\Sigma} = 0,00224 + 2 \cdot 0,064 \cdot 10^{-3} = 0,00524 \text{ Гн}$$

Оскільки виконується співвідношення $L_{\Sigma} > L_{\text{д}}$, то згладжуючий реактор не потрібний.

Коефіцієнт посилення тиристорного перетворювача:

$$k_{\text{тп}} = \frac{E_{\text{д0}}}{U_{\text{м}}}, \quad (2.21)$$

де $U_{\text{м}}$ – амплітудне значення напруги порівняння; приймаємо $U_{\text{м}} = 10 \text{ В}$;

$$k_{\text{тп}} = \frac{513}{10} = 51,3.$$

Коефіцієнт посилення двигуна:

$$k_{\text{д}} = \frac{1}{c}, \quad (2.22)$$

$$k_{\text{д}} = \frac{1}{7,3} = 0,137.$$

Електромагнітна постійна часу якінного кола двигуна

$$T_a = \frac{L_{\text{як}}}{R_{\text{як}}}, \quad (2.23)$$

$$T_a = \frac{0,00234}{0,0707} = 0,033 \text{ с}.$$

Електромагнітна постійна часу головного кола

$$T_e = \frac{L_{\Sigma}}{R_{\Sigma}}, \quad (2.24)$$

$$T_e = \frac{0,00524}{0,1568} = 0,068 \text{ с}.$$

Електромеханічна постійна двигуна електроприводу

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

$$T_m = 12,42 \cdot \frac{0,1568}{7,3^2} = 0,068 \text{ с}.$$

2.4 Вибір структури системи автоматичного управління

Згідно з рекомендаціями [40], для досягнення високих регульовальних характеристик у статичних і динамічних умовах використовується двоконтурна автоматична система управління, побудована за принципом підлеглого регулювання, з внутрішнім контуром струму та зовнішнім контуром швидкості, замкнутим по ЕРС двигуна.

Характерною рисою електроприводів з реверсивним тиристорним перетворювачем із роздільним управлінням вентилями є зона нечутливості під час роботи груп тиристорів і виникнення переривистого струму при низьких наван-

таженнях. Зона нечутливості призводить до уповільнення перехідних процесів і виникнення імпульсних змін динамічного струму у системі із задатчиком інтенсивності через розмикання контуру регулювання струму. Переривистий струм супроводжується різким зростанням еквівалентного опору якірного кола, що наближається до нескінченності при зниженні середнього струму до нуля. Це призводить до зміни параметрів об'єктів регулювання і погіршення динамічних властивостей системи.

Щоб усунути негативні ефекти, що виникають через особливості тиристорних перетворювачів, у контур регулювання струму додається підлеглий контур регулювання напруги тиристорного перетворювача.

2.5 Синтез послідовного корегуючого пристрою контура струму

Характер перехідного процесу визначається співвідношеннями постійних часу елементів системи, тому для забезпечення високої якості перехідних процесів необхідно створювати потрібні значення цих співвідношень за допомогою коригувальних регуляторів. У системах підлеглого регулювання кількість послідовно включених регуляторів відповідає кількості контурів. Згідно з рекомендаціями [40], у контур регулювання напруги тиристорного перетворювача вводиться П-регулятор, а в контур струму – ІІІ-регулятор.

Схема контура струму з внутрішнім контуром напруги показана на рис. 2.1. Параметри елементів контурів напруги та струму вибираються так, щоб передавальна функція контуру струму при безперервному струмі збігалася з передавальною функцією контуру струму без додаткового контуру напруги.

У режимі переривистого струму середнє значення ЕРС тиристорного перетворювача дорівнює ЕРС двигуна. Тому, якщо припустити, що ця величина залишається сталою, як це зазвичай робиться при розгляді контуру струму, то можна вважати, що зворотний зв'язок по напрузі відсутній, а коефіцієнт посилення регулятора збільшується, що сприяє покращенню динамічних характеристик системи в режимі переривистих струмів.

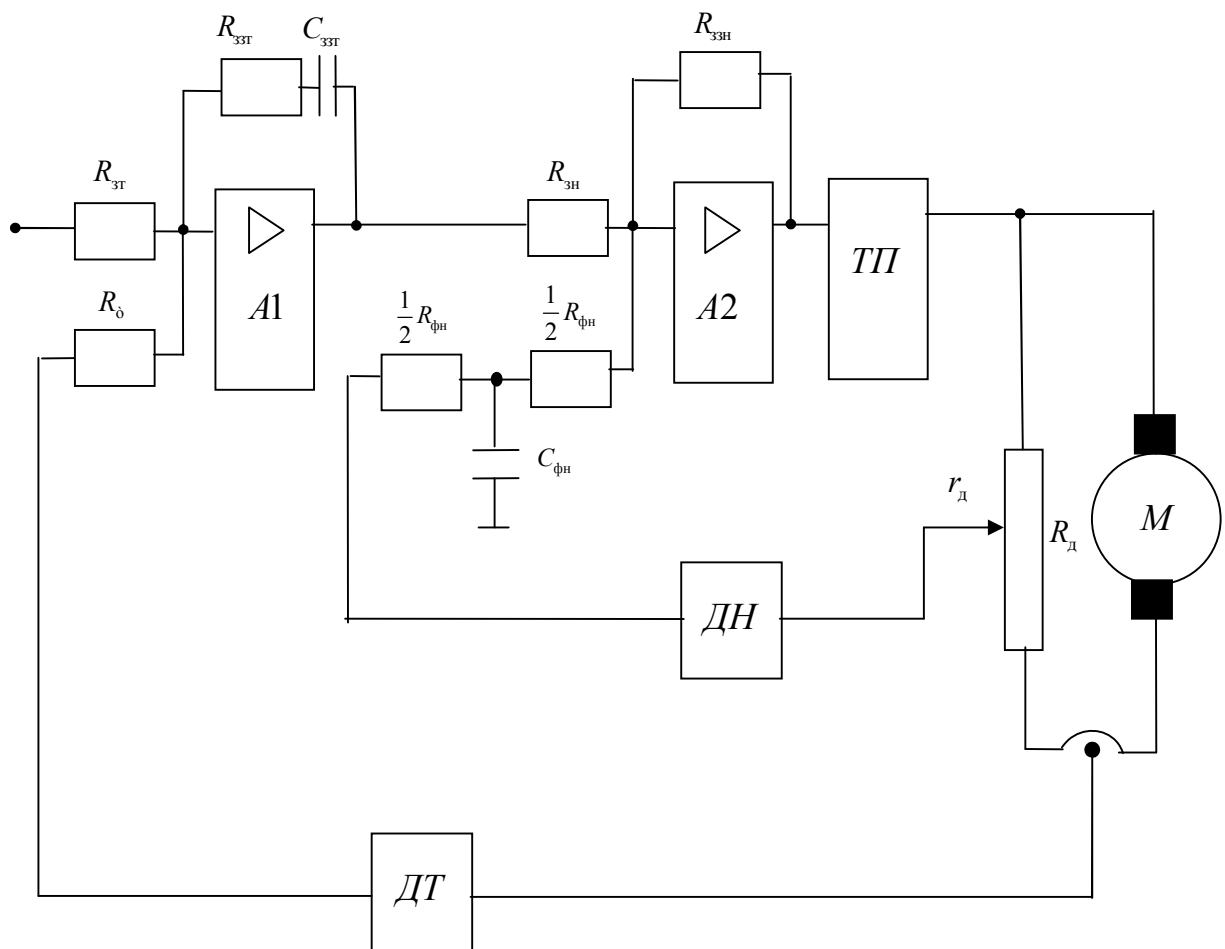


Рисунок 2.1 – Спрощена принципова схема контура струму

На практиці зворотний зв'язок реалізується не по ЕРС, а по напрузі тиристорного перетворювача. Відповідна алгоритмічна схема зображена на рис.2.2.

Схема складається з наступних елементів.

1. Вентильний перетворювач, що є інерційною ланкою з коефіцієнтом посилення $k_{\text{тп}}$ і малою постійною часу $T_{\text{тп}}$, яка складається з середньостатистичного запізнювання перетворювача і постійної часу система імпульсно-фазового управління.

2. Якірний ланцюг електродвигуна. У контурі струму системи підлеглого регулювання якірний ланцюг двигуна є інерційною ланкою з коефіцієнтом посилення $1/R_{\text{я}\Sigma}$ і «великою» постійною часу T_e .

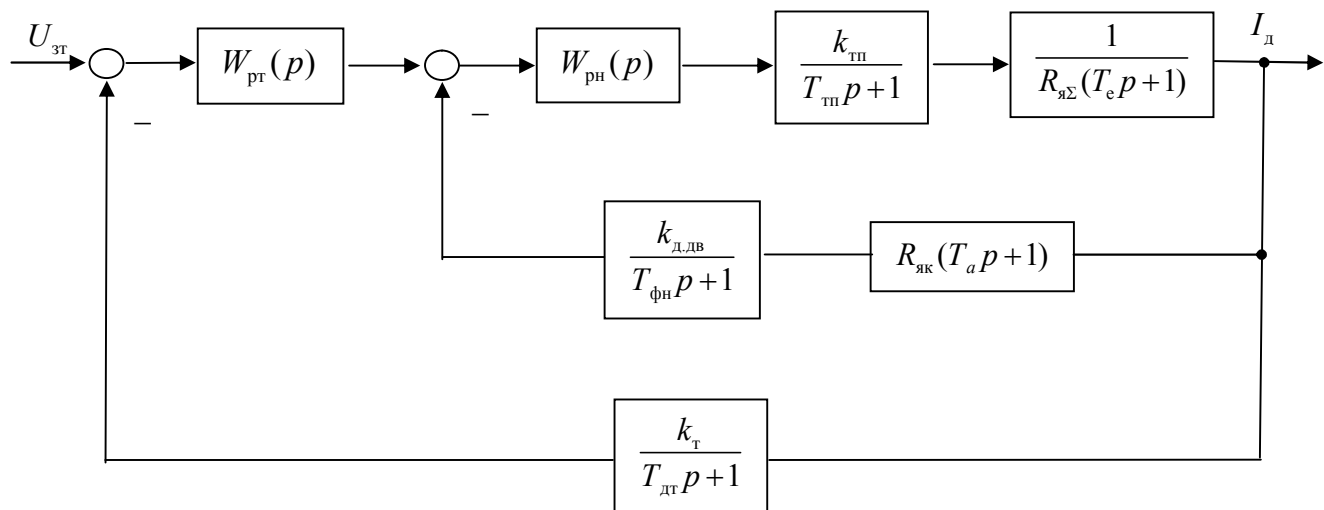


Рисунок 2.2 – Алгоритмічна схема контура струму

3. Датчик струму, що є інерційною ланкою з коефіцієнтом посилення $k_{\text{дт}}$ і «малою» постійною часу $T_{\text{дт}}$, які складаються з постійної часу власне датчика струму і постійної часу фільтру на його вході.

Для отримання сигналу зворотного зв'язку по струму використовується приладовий шунт, тобто шунт, встановлюється для амперметра. Номінальний струм шунта приймаємо рівним максимальному значення струму робочого перевантаження двигуна I_{max} , приймаємо $I_{\text{шн}} = I_{\text{max}} = 2,5 I_{\text{н}} = 2,5 \cdot 168 = 420 \text{ A}$, падіння напруги на шунті при постійному струмі $U_{\text{шн}} = 75 \cdot 10^{-3} \text{ B}$.

Коефіцієнт посилення шунта

$$k_{\text{ш}} = \frac{U_{\text{шн}}}{I_{\text{шн}}}, \quad (2.26)$$

$$k_{\text{ш}} = \frac{75 \cdot 10^{-3}}{420} = 0,1786 \cdot 10^{-3} \text{ B / A}$$

Коефіцієнт посилення датчика струму визначуваний таким чином:

$$k_{\text{дт}} \leq \frac{24}{k_{\text{ш}} I_{\text{max}}}, \quad (2.27)$$

$$k_{\text{дт}} \leq \frac{24}{0,1786 \cdot 10^{-3} \cdot 420} = 320$$

Вживані як датчик струму операційні підсилювачі мають два фіксовані значення коефіцієнта посилення 130 і 320. Друге значення використовується у разі, коли для отримання зворотного зв'язку по струму використовується «приладовий» шунт, що має місце в проектуваному електроприводі. Приймаємо $k_{\text{дт}} = 320$.

Передавальна функція датчика струму спільно з шунтом

$$W_{\text{дт}}(p) = \frac{k_{\text{т}}}{T_{\text{дт}} p + 1}, \quad (2.28)$$

де $k_{\text{т}}$ – коефіцієнт посилення ланцюга зворотного зв'язку по струму

$$k_{\text{т}} = k_{\text{дт}} k_{\text{ш}}, \quad (2.29)$$

$$k_{\text{т}} = 320 \cdot 0,1786 \cdot 10^{-3} = 0,057 \text{ В} / \text{А}$$

4. Дільник напруги. Сигнал, пропорційний напрузі двигуна, знімається з дільника напруги $R_{\text{д}}$, коефіцієнт посилення якого визначається з умови отримання сигналу 24 В на вході датчика напруги при максимальному напругою двигуна

$$k_{\text{д дв}} = \frac{24}{U_{\text{дв max}}}. \quad (2.30)$$

Приймаючи $U_{\text{дв max}} = U_{\text{н}} = 440 \text{ В}$, отримуємо

$$k_{\text{д дв}} = \frac{24}{440} = 0,055.$$

Коефіцієнт посилення датчика напруги $k_{\text{д н}} = 1$.

На виході датчика напруги встановлений фільтр з постійною часу $T_{\text{фн}}$.

5. Регулятор напруги. Згідно рекомендаціям [40], вибираємо типовий П-регулятор з передавальною функцією

$$W_{\text{рн}}(p) = k_{\text{рн}} = \frac{R_{\text{ззн}}}{R_{\text{зн}}}. \quad (2.31)$$

Визначимо передавальну функцію замкнутого контура напруги

$$W_{\text{н}}(p) = \frac{\frac{k_{\text{рн}} k_{\text{тп}}}{(T_{\text{тп}} p + 1) R_{\text{я}\Sigma} (T_{\text{е}} p + 1)}}{1 + \frac{k_{\text{рн}} k_{\text{тп}}}{(T_{\text{тп}} p + 1) R_{\text{я}\Sigma} (T_{\text{е}} p + 1)} \cdot \frac{k_{\text{д дв}} R_{\text{а}} (T_{\text{а}} p + 1)}{(T_{\text{фн}} p + 1)}} = \frac{k_{\text{рн}} k_{\text{тп}} (T_{\text{фн}} p + 1)}{[k_{\text{д дв}} k_{\text{рн}} k_{\text{тп}} R_{\text{а}} (T_{\text{а}} p + 1) + R_{\text{я}\Sigma} (T_{\text{фн}} p + 1) (T_{\text{е}} p + 1)] (T_{\text{тп}} p + 1)}. \quad (2.32)$$

Якщо прийняти $T_{\text{фн}} = T_{\text{а}}$, то передавальна функція замкнутого контура напруги прийме вигляд:

$$W_{\text{н}}(p) = \frac{k_{\text{рн}} k_{\text{тп}}}{[k_{\text{рн}} k_{\text{тп}} k_{\text{д дв}} R_{\text{а}} + R_{\text{я}\Sigma} (T_{\text{е}} p + 1)] (T_{\text{тп}} p + 1)}. \quad (2.33)$$

Введемо позначення для коефіцієнта посилення контура напруги

$$k_{\text{пн}} = k_{\text{рн}} k_{\text{тп}} k_{\text{д дв}} \frac{R_{\text{як}}}{R_{\text{я}\Sigma}}. \quad (2.34)$$

Після нескладних перетворень отримуємо передатну функцію контура напруги у вигляді

$$W_{\text{н}}(p) = \frac{k_{\text{рн}} k_{\text{тп}}}{R_{\text{я}\Sigma} (1 + k_{\text{пн}})} \cdot \frac{1}{\left(\frac{T_{\text{е}}}{1 + k_{\text{пн}}} p + 1 \right) \cdot (T_{\text{тп}} p + 1)}. \quad (2.35)$$

Алгоритмічна схема контура струму з урахуванням (2.35) показана на рис.2.3. На рис.2.4 показана перетворена алгоритмічна схема з одиничним зворотним зв'язком. Форсуюча ланка $(T_{\text{дт}} p + 1)$ на вході контура не показана, оскі-

льки її вплив на динамічні характеристики контура струму у край не значно зважаючи на мале значення постійної часу $T_{\text{дт}}$.

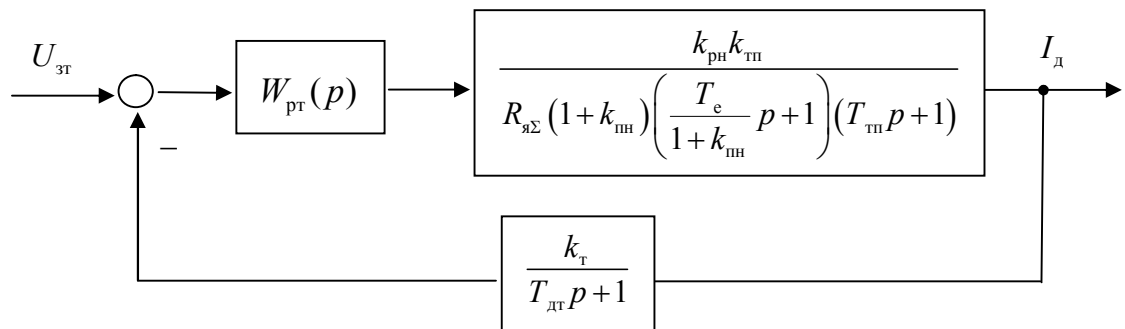


Рисунок 2.3 – алгоритмічна схема контура струму з перетвореним контуром напруги.

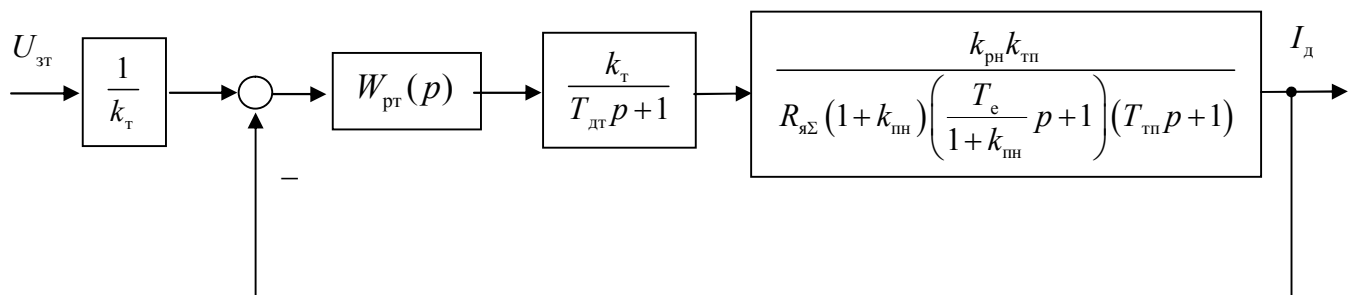


Рисунок 2.4 – Алгоритмічна схема контура струму з одиничним зворотним зв'язком

Мала некомпенсуєма постійна часу контура струму рівна сумі малих постійних часу тиристорного перетворювача ч датчика струму

$$T_{\text{вт}} = T_{\text{тп}} + T_{\text{дт}}. \quad (2.36)$$

Згідно рекомендаціям [40] приймаємо $T_{\text{вт}} = 0,005 \text{ с}$

Контур струму оптимізує на основі використання модульного критерію, тому бажана передавальна функція розімкненого контура струму повинна мати вигляд

$$W_{\text{от}}(p) = \frac{1}{2T_{\text{от}}p(1 + T_{\text{от}}p)}. \quad (2.37)$$

Дана передавальна функція може бути реалізована, якщо як регулятор струму прийняти типовий ПІ-регулятор з передавальною функцією

$$W_{\text{рт}}(p) = \frac{T_{\text{рт}}p + 1}{T'_{\text{рт}}p}. \quad (2.38)$$

Прийmemo $T_{\text{рт}} = \frac{T_{\text{е}}}{1 + R_{\text{н}}}$, тоді передавальна функція розімкненого контура струму матиме вигляд

$$W_{\text{т}}(p) = \frac{k_{\text{рн}}k_{\text{тп}}k_{\text{т}}}{T'_{\text{рт}}pR_{\Sigma}(1 + k_{\text{пн}})(T_{\text{от}}p + 1)}, \quad (2.39)$$

При рівності

$$\frac{T'_{\text{рт}}R_{\Sigma}(1 + k_{\text{пн}})}{k_{\text{рн}}k_{\text{тп}}k_{\text{т}}} = 2T_{\text{от}} \quad (2.40)$$

отримуємо бажану передавальну функцію розімкненого контура струму.

Передавальна функція замкнутого контура струму, оптимізованого згідно модульного критерію має вигляд:

$$W_{\text{т}}(p) = \frac{1/k_{\text{т}}}{2T_{\text{от}}^2p^2 + 2T_{\text{от}}p + 1}. \quad (2.41)$$

Дана передавальна функція з точністю до $1/k_{\text{т}}$ співпадає з бажаною передавальною функцією.

Визначаємо параметри регуляторів контурів напруги і струму.

Для визначення коефіцієнта підсилювача ПІ-регулятора необхідно визначити критичний коефіцієнт посилення контура напруги $k_{\text{пнкp}}$, відповідний межі стійкої роботи перетворювача тиристора

$$k_{\text{пнкр}} = \frac{\pi^2 T_e T_r}{2T_0^2 \left(1 - \frac{T_r}{T_e}\right)}, \quad (2.42)$$

де $T_r = 2T_{\text{мт}} = 2 \cdot 0,005 = 0,01 \text{ с}$ при настройці контура струму по модульному критерію.

$$T_0 \equiv \frac{2\pi}{m\omega_0}, \quad (2.43)$$

ω_0 – кутова частота живлячої мережі; $\omega_0 \equiv 2\pi f$;

$$T_0 = \frac{2\pi}{m2\pi f} = \frac{1}{mf} = \frac{1}{6 \cdot 50} = \frac{1}{300} \text{ с}.$$

$$k_{\text{пн.кр}} = \frac{3,14^2 \cdot 0,068 \cdot 0,01}{2 \cdot \left(\frac{1}{300}\right)^2 \cdot \left(1 - \frac{0,01}{0,068}\right)} = 78.$$

Прийmemo $k_{\text{пн}} = 50$.

Коефіцієнт посилення П-регулятора визначуваний із співвідношення

$$k_{\text{рн}} = \frac{k_{\text{пн}} R_{\text{я}\Sigma}}{k_{\text{тп}} k_{\text{д дв}} R_{\text{як}}}, \quad (2.44)$$

$$k_{\text{рн}} = \frac{50 \cdot 0,1568}{51,3 \cdot 0,055 \cdot 0,0707} = 39,3.$$

Приймаємо $R_{\text{зн}} = 4 \cdot 10^3 \text{ Ом}$, тоді

$$R_{\text{зн}} = R_{\text{зн}} k_{\text{рн}}, \quad (2.45)$$

$$R_{\text{зн}} = 4 \cdot 10^3 \cdot 39,3 = 156,2 \cdot 10^3 \text{ Ом/}$$

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

$$\frac{R_{\text{фн}}}{4} C_{\text{фн}} = T_{\text{фн}} = T_a. \quad (2.46)$$

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

$$C_{\text{фн}} = \frac{4T_a}{R_{\text{зн}}}, \quad (2.47)$$

$$C_{\text{фн}} = \frac{4 \cdot 0,033}{4 \cdot 10^3} = 3,3 \cdot 10^{-6} \Phi$$

Параметри регулятора струму визначаємо із співвідношень

$$T_{\text{рт}} = C_{\text{ззт}} R_{\text{ззт}} = \frac{T_e}{1 + k_{\text{пн}}}, \quad (2.48)$$

$$T'_{\text{рт}} = C_{\text{ззт}} R_{\text{зт}} = \frac{2T_{\text{мт}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}}}{R_{\text{я}\Sigma} (1 + k_{\text{пн}})}. \quad (2.49)$$

Приймаємо $C_{\text{ззт}} = 1 \cdot 10^{-6} \Phi$, тоді

$$R_{\text{ззт}} = \frac{T_e}{(1 + k_{\text{пн}}) \cdot C_{\text{ззт}}}, \quad (2.50)$$

$$R_{\text{ззт}} = \frac{0,068}{(1 + 50) \cdot 1 \cdot 10^{-6}} = 1,3 \cdot 10^3 \text{ Ом}$$

$$R_{\text{зт}} = \frac{2T_{\text{мт}} k_{\text{рн}} k_{\text{тп}} k_{\text{т}}}{R_{\text{я}\Sigma} (1 + k_{\text{пн}}) C_{\text{ззт}}}, \quad (2.51)$$

$$R_{\text{зт}} = \frac{2 \cdot 0,005 \cdot 39,3 \cdot 51,3 \cdot 0,057}{0,1568 \cdot (1 + 50) \cdot 10^{-6}} = 143,7 \cdot 10^3 \text{ Ом}$$

$$R_{\text{т}} = R_{\text{зт}} = 143,7 \cdot 10^3 \text{ Ом}$$

2.6 Синтез послідовного коректуючого пристрою контура ЕРС

Система регулювання швидкості обертання із зворотним зв'язком по ЕРС двигуна застосовуються для електроприводів з нерегульованим потоком збудження в тих випадках, коли вимоги до точності регулювання швидкості відносно невеликі. Тахогенератори на таких приводах зазвичай не встановлюються. Серед електроприводів загальнопромислових механізмів електроприводу вказаного класу в кількісному відношенні є переважаючими. Тому в електроприводі, що розробляється, для регулювання швидкості обертання використовується зворотний зв'язок по ЕРС двигуна.

Вимірювання ЕРС проводиться шляхом підсумовування сигналу, пропорційного напрузі двигуна з сигналом струмової компенсації, пропорційним падінню напруги в активному опорі якоря двигуна.

На рис.2.5 показана принципова схема системи регулювання швидкістю із зворотним зв'язком по ЕРС двигуна. Сигнал, пропорційний напрузі двигуна, знімається з ділянки R_d і подається на один з входів датчика ЕРС ДЕ через фільтр $R_n C_n$, що виконує функцію аперіодичної ланки з постійною часу T_a в ланцюзі зворотного зв'язку по напрузі. На другий вхід ДЕ подається сигнал струмової компенсації.

Як датчик ЕРС застосовується операційний підсилювач з гальванічно розділеним входом і виходом. Цей підсилювач має два входи, один з яких служить для обхвату підсилювача зворотним зв'язком через вбудований опір $R_{де}$. Другий вихід підсилювача, гальванічно відокремлений 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 збуренню за завданням.

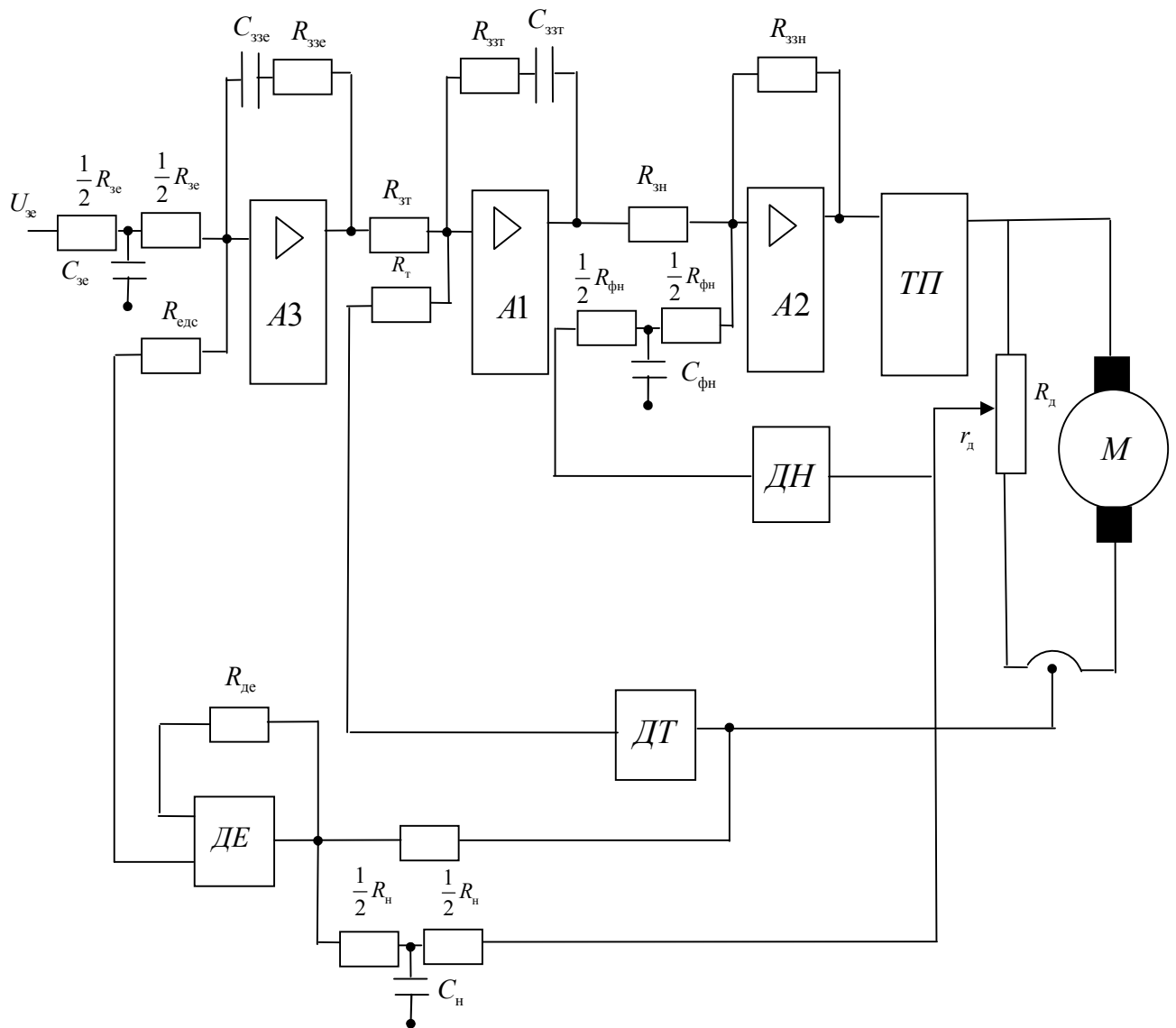


Рисунок 2.5 – Принципова схема системи управління із зворотним зв'язком по ЕРС

На рис.2.6 показана алгоритмічна схема системи.

Оптимізований по модульному критерію контур струму представлений аперіодичною ланкою

$$W_{\tau}(p) = \frac{1/k_{\tau}}{2T_{\mu\tau}p + 1}, \quad (2.52)$$

яка отримується при зневазі доданком $2T_{\mu\tau}^2 p^2$ в знаменнику передавальної функції контура струму.

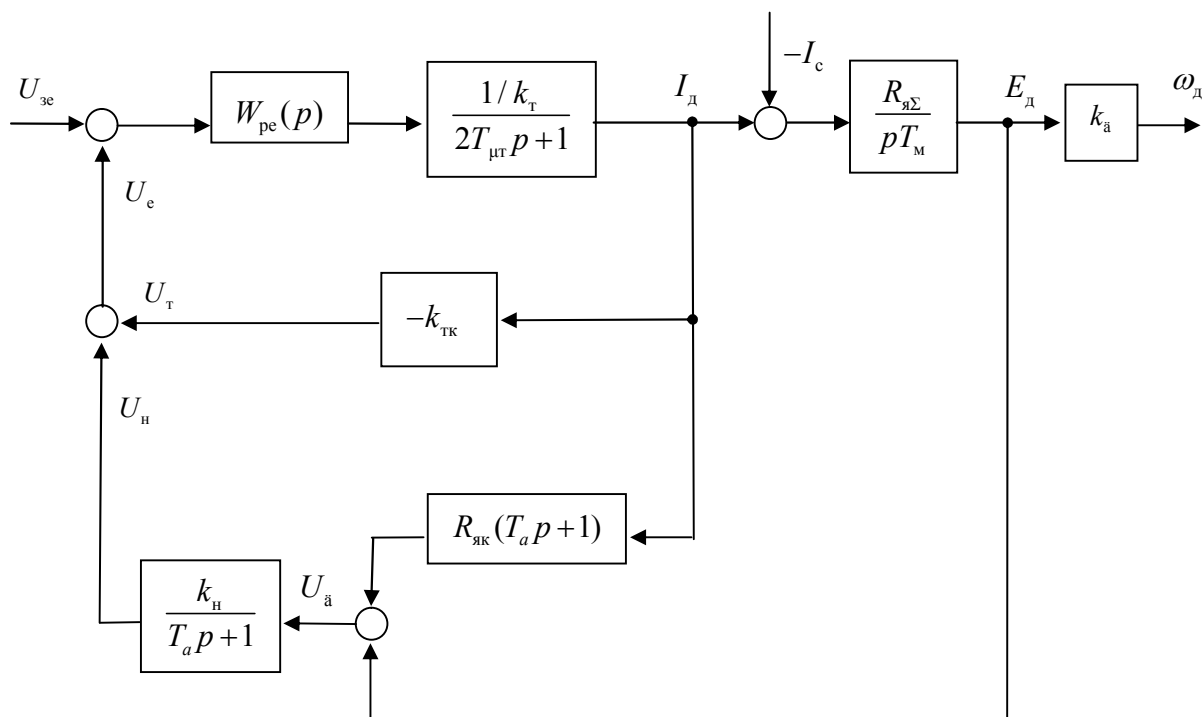


Рисунок 2.6 – Алгоритмічна схема системи регулювання швидкості із зворотним зв'язком по ЕРС

Коефіцієнт посилення ланцюга струмової компенсації вибирається таким чином:

$$k_{TK} = k_H R_{як}, \quad (2.53)$$

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

Сигнал на виході датчика ЕРС, тобто сигнал зворотного зв'язку по ЕРС:

$$U_e = U_H - U_T = \frac{k_H}{T_a p + 1} E_{дв}. \quad (2.54)$$

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

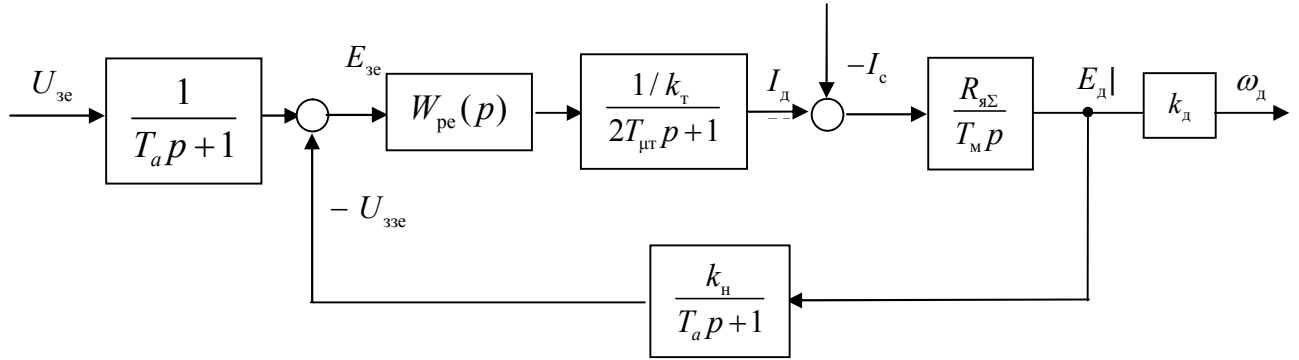


Рисунок 2.7 – Перетворена алгоритмічна схема система

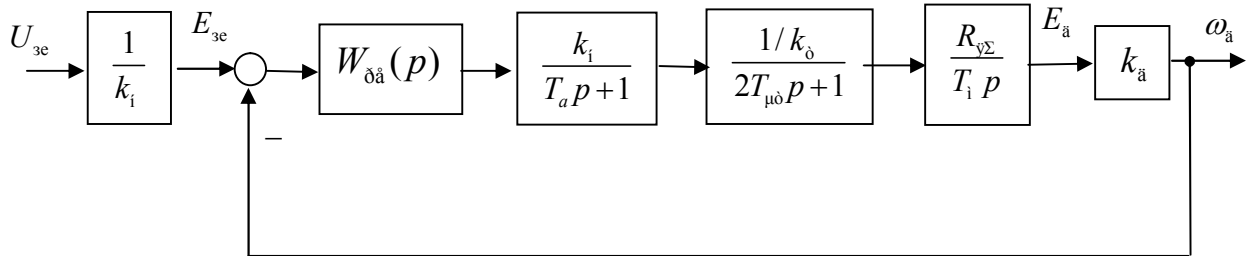


Рисунок 2.8 – Розрахункова алгоритмічна схема системи
з одиничним зворотним зв'язком

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

$$W_{pe}(p) = \frac{T_M k_T}{2k_H T_{\mu e} R_{\Sigma}} \cdot \frac{4T_{\mu e} p + 1}{4T_{\mu e} p}, \quad (2.55)$$

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

$$T_{\mu e} = 2T_{\mu t} + T_a, \quad (2.56)$$

$$T_{\mu e} = 2 \cdot 0,005 + 0,035 = 0,043 \text{ с}$$

Оскільки в склад $T_{\mu e}$ входить T_a , то швидкодія системи із зворотним зв'язком по ЕРС нижче, ніж в системі з використанням тахогенератора.

Коефіцієнт посилення k_n рівний

$$k_n = \frac{U_{ze \max}}{E_{дв \max}}, \quad (2.57)$$

де $E_{дв \max}$ – ЕРС двигуна;

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

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

$$k_n = \frac{24}{440} = 0,055.$$

Зіставляючи передавальну функцію (2.55) з передавальною функцією типового ПІ-регулятора, отримуємо наступні рівняння

$$R_{ззе} C_{ззе} = 4T_{\mu e}, \quad (2.58)$$

$$\frac{R_{ззе}}{R_{зе}} = \frac{T_M k_T}{2T_{\mu e} k_n k_{\Sigma}}. \quad (2.59)$$

Задаємося величиною місткості $C_{ззе} = 5 \cdot 10^{-6} \text{ Ф}$, тоді з (2.58)

$$R_{ззе} = \frac{4T_{\mu e}}{C_{ззе}}, \quad (2.60)$$

$$R_{ззе} = \frac{4 \cdot 0,043}{5 \cdot 10^{-6}} = 34,3 \cdot 10^3 \text{ Ом}$$

Із співвідношення (2.59) знаходимо

$$R_{зе} = \frac{R_{ззе} 2T_{\mu e} k_n R_{я\Sigma}}{T_M k_T}, \quad (2.61)$$

$$R_{зе} = \frac{34,3 \cdot 10^3 \cdot 2 \cdot 0,043 \cdot 0,055 \cdot 0,1568}{0,068 \cdot 0,057} = 65,6 \cdot 10^3 \text{ Ом}$$

Приймаємо $R_{\text{едс}} = R_{\text{зе}} = 65,6 \cdot 10^3 \text{ Ом}$

З виразу для коефіцієнта посилення ланцюга зворотного зв'язку по напру-
зі

$$k_{\text{н}} = k_{\text{д дв}} \cdot \frac{R_{\text{зе}}}{R_{\text{н}}} \cdot \frac{R_{\text{де}}}{R_{\text{едс}}} \quad (2.62)$$

визначаємо

$$R_{\text{н}} = \frac{k_{\text{д дв}}}{k_{\text{н}}} \cdot \frac{R_{\text{зе}} \cdot R_{\text{де}}}{R_{\text{едс}}}, \quad (2.63)$$

$$R_{\text{н}} = \frac{0,1091}{0,1091} \cdot \frac{65,6 \cdot 10^3 \cdot 50 \cdot 10^3}{65,6 \cdot 10^3} = 50 \cdot 10^3 \text{ Ом}$$

Тут $R_{\text{де}}$ – вбудований опір; $R_{\text{де}} = 50 \text{ кОм}$

Передавальний коефіцієнт струмової компенсації

$$k_{\text{тк}} = k_{\text{н}} \cdot R_{\text{як}}, \quad (2.64)$$

$$k_{\text{тк}} = 0,055 \cdot 0,0707 = 3,9 \cdot 10^{-3}$$

Передавальний коефіцієнт струмової компенсації може бути визначений
також таким чином

$$k_{\text{тк}} = k_{\text{ш}} \cdot \frac{R_{\text{зе}}}{R_{\text{тк}}} \cdot \frac{R_{\text{де}}}{R_{\text{зе}}}, \quad (2.65)$$

звідки

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

$$R_{\text{тк}} = \frac{0,1786 \cdot 10^{-3}}{3,9 \cdot 10^{-3}} \cdot \frac{65,6 \cdot 10^{-3}}{65,6 \cdot 10^{-3}} \cdot 50 \cdot 10^3 = 2,3 \cdot 10^3 \text{ Ом}.$$

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

$$\frac{R_{3e}}{4} \cdot C_{3e} = \frac{R_H}{4} \cdot C_H = T_a, \quad (2.66)$$

$$C_{3e} = \frac{4 \cdot T_a}{R_{3e}}. \quad (2.67)$$

$$C_{3e} = \frac{4 \cdot 0,033}{65,6 \cdot 10^3} = 2 \cdot 10^{-6} \Phi$$

$$C_H = \frac{4 \cdot T_a}{R_H}, \quad (2.68)$$

$$C_H = \frac{4 \cdot 0,033}{65,6 \cdot 10^3} = 2 \cdot 10^{-6} \Phi.$$

Передавальні функції системи оптимізованою по симетричному критерію, мають вигляд:

$$W_{\omega_d u_{3e}}(p) = \frac{\omega_d(p)}{U_{3e}(p)} = \frac{k_d}{k_H} \cdot \frac{1 + 4T_{\mu e}p}{8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e}p + 1}; \quad (2.69)$$

$$W_{\omega_d I_{ст}}(p) = \frac{\omega_d(p)}{I_{ст}(p)} = -\frac{8T_{\mu e}^2 k_d R_{\Sigma}}{T_M} \cdot \frac{p(1 + T_{\mu e}p)}{8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e}p + 1}; \quad (2.70)$$

$$W_{I_d U_{3e}}(p) = \frac{I_d(p)}{U_{3e}(p)} = \frac{T_M}{k_H R_{\Sigma}} \cdot \frac{p(1 + 4T_{\mu e}p)}{8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e}p + 1}; \quad (2.71)$$

$$W_{I_d I_{ст}}(p) = \frac{I_d(p)}{I_{ст}(p)} = \frac{1 + 4T_{\mu e}p}{8T_{\mu e}^3 p^3 + 8T_{\mu e}^2 p^2 + 4T_{\mu e}p + 1}, \quad (2.72)$$

Алгоритмічна схема системи показана на рис.2.9.

2.7 Дослідження динамічних характеристик системи

По приведених передавальних функціях в системі MATLAB на ЕОМ розраховані перехідні процеси системи.

Графіки перехідних процесів вказані на рис. 2.10 і рис. 2.11 криві 1. Як видно з графіків, при настроюванні системи на симетричний оптимум, має місце значне перерегулювання швидкості по заданій дії рівне 43%. Для зменшення перерегулювання на вхід системи ставимо фільтр з постійною часу $4T_{\mu c}$. Графіки перехідних процесів системи з фільтром на вході вказаному на рис. 2.10 криві 1. Як видно з графіка перерегулювання зменшилося до 8,1%, проте час першого погоджено і час регулювання при цьому збільшився.

Розрахунок перехідних процесів проводився також за допомогою моделі системи, показаної на рис. 2.9, побудованої на ЕОМ в пакеті MATLAB. Графіки перехідних процесів, отримані за допомогою моделі, повністю співпадає з графіками отриманими по передавальних функціях.

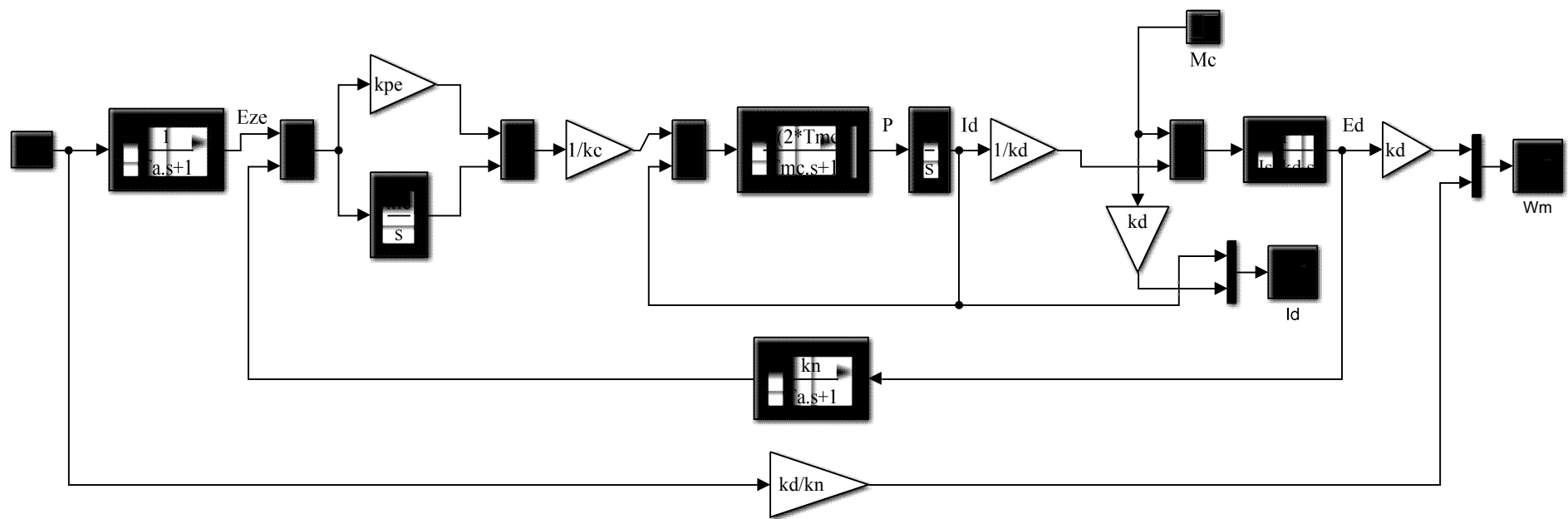


Рисунок 2.9 – Схема моделі системи, розроблена на ЕОМ

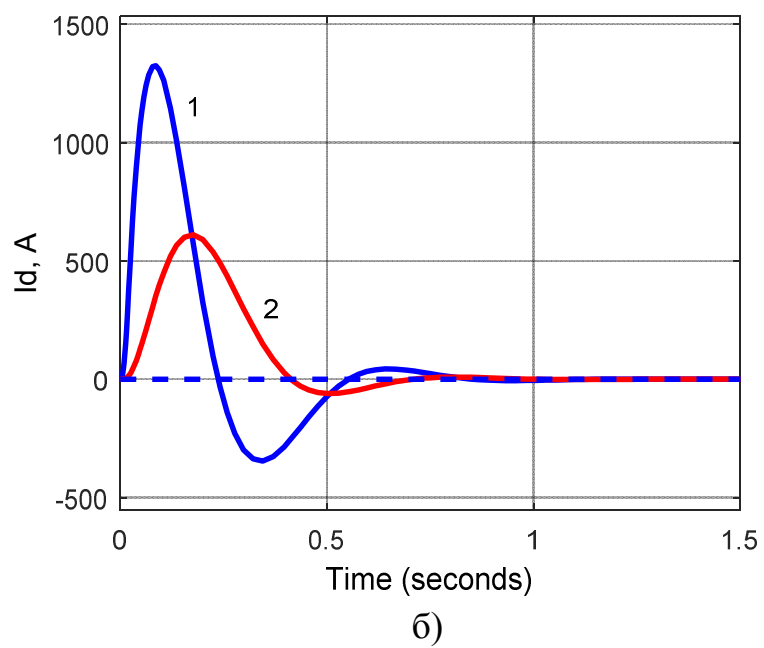
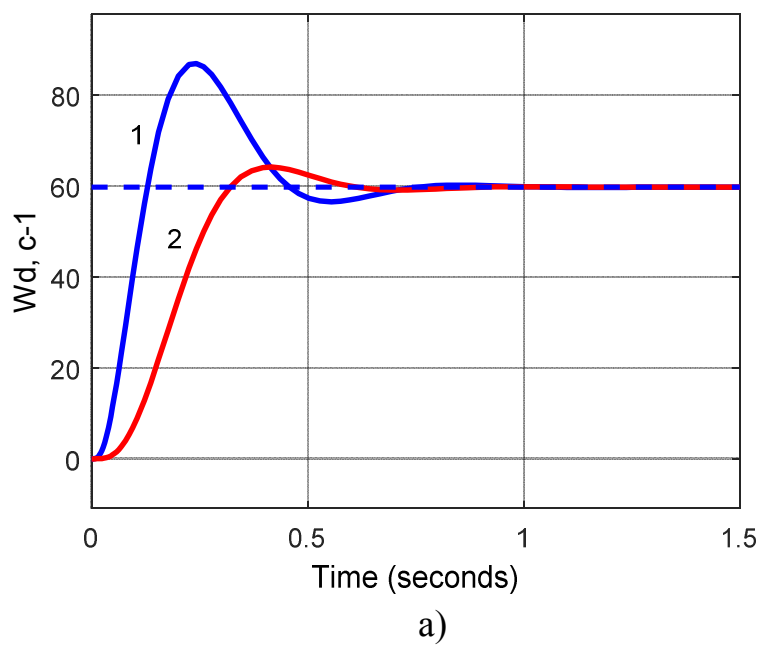


Рисунок 2.10 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ (а) і струму $I_d = f(t)$ (б) по задаючій дії

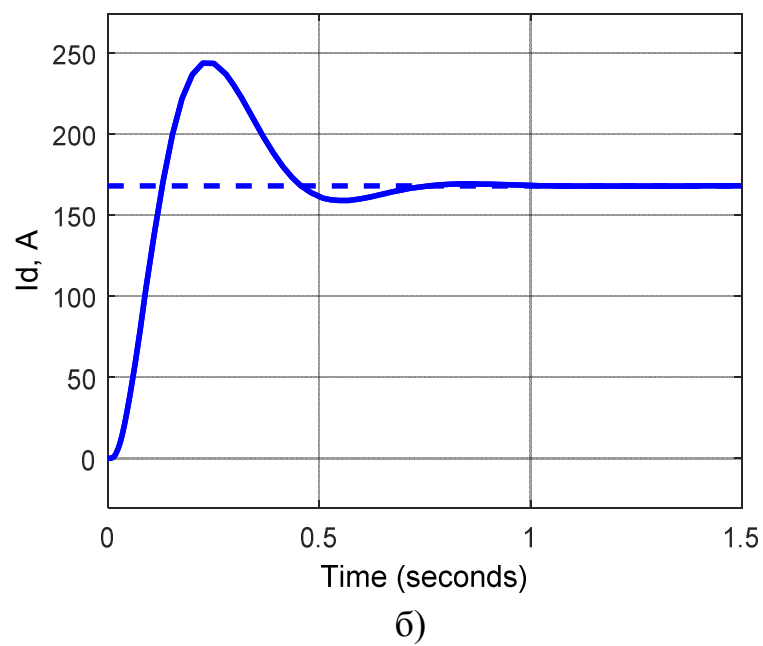
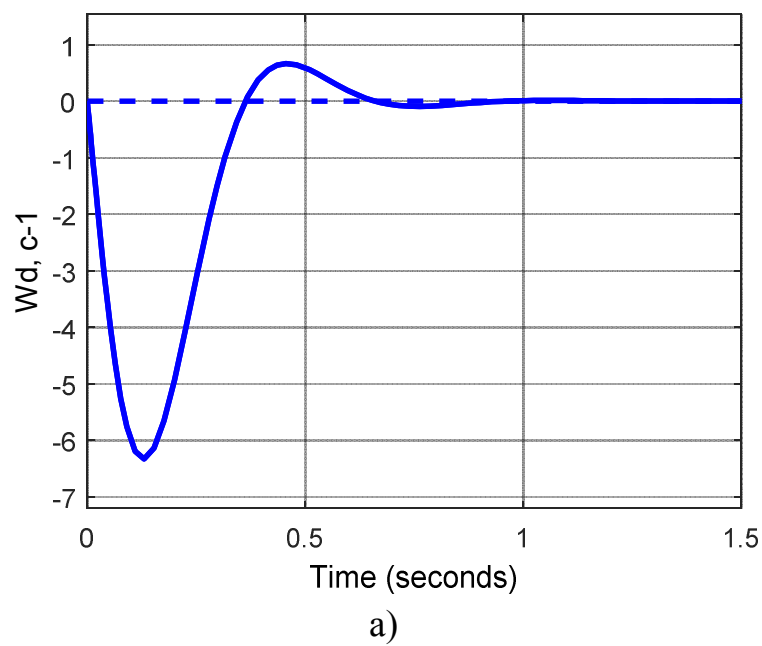


Рисунок 2.11 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ (а)
і струму $I_d = f(t)$ (б) по збурюючій дії:

ВИСНОВКИ ДО РОЗДІЛУ 2

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

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

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

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

5. Проведений синтез послідовних коригуючих пристроїв для окремих контурів. Оскільки характер перехідних процесів залежить від відношення постійних часу елементів системи, для досягнення високоякісних перехідних процесів використовувалися регулятори, що коригують ці відношення. Для контура напруги синтезований П-регулятор, 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 ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ КІНЦЕВОЇ ЖОРСТКОСТІ ПІДЙОМНОГО КАНАТА

3.1 Розрахункові схеми механічної частини електроприводу

Розрахунок системи регулювання проводився з припущенням, що всі елементи механічної частини системи, включаючи підйомний канат, мають нескінченну жорсткість, тобто розглядалася одномасова система. В реальних системах зв'язки між елементами мають кінцеву жорсткість, а підйомний канат є пружним елементом з розподіленими параметрами. Тому механізм підйому мостового крана є багатомасовою системою, яка включає ряд мас, що рухаються поступально і обертаються.

Кінематичної схеми електроприводу механізму підйому мостового крана представлена на рис. 3.1,а. Тут двигун через сполучну муфту $СМ1$, клиноремену передачу $КРП$, ряд зубчатих передач $ЗП_1 \dots, ЗП_i$ і сполучну муфту $СМ2$ приводить в обертання барабан $Б$, що перетворює обертальний рух в поступальний переміщення ряду зв'язаних мас. У даній схемі передбачається, що робочим органом механізму є вантажозахватний пристрій, що переміщає вантаж $В$, що має масу m_B , яка рухається зі швидкістю V_B і на яку діє сила тяжіння F_B .

Розглянута схема демонструє, що механічна частина електроприводу в загальному випадку є системою зв'язаних мас, які рухаються з різними швидкостями, обертаючись або переміщуючись поступально. Під навантаженням елементи системи (вали, опори, клиноременні передачі, зубчаті зачеплення, канати тощо) зазнають деформацій, оскільки механічні зв'язки не є абсолютно жорсткими. Зміни навантаження дозволяють масам переміщуватись одна відносно одної, при цьому рівень цього переміщення визначається жорсткістю зв'язків.

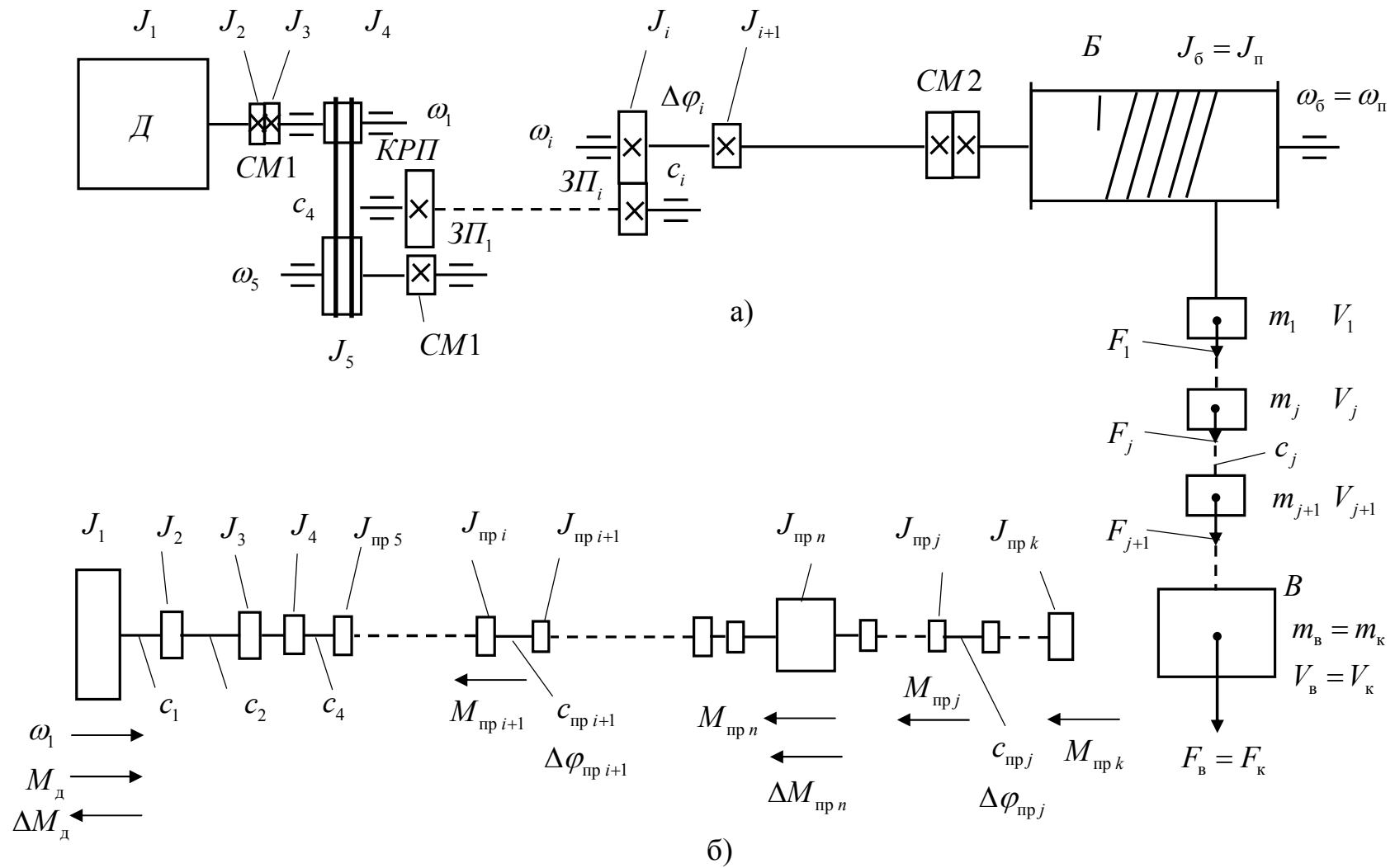


Рисунок 3.1 – Кінематична (а) і розрахункова (б) схеми механічної частини електроприводу

При складанні цієї кінематичної схеми припущено, що механічна частина приводу включає n оберально рухомих зосереджених мас та k поступально рухомих мас, при цьому інерція зв'язуючих елементів не враховується. Кожен оберально рухомий елемент має момент інерції J_i і з'єднаний з $(i + 1)$ -м елементом за допомогою механічного зв'язку з жорсткістю c_i . Кожен поступально рухомий елемент має масу m_j і з'єднаний з наступним елементом за допомогою механічного зв'язку з жорсткістю c_j . Жорсткість пружних механічних зв'язків, які підлягають закону Гука, можна визначити за допомогою відповідних співвідношень.

$$c_i = M_{\text{пр}i} \Delta \varphi_i, \quad c_j = F_{\text{пр}j} \Delta S_j, \quad (3.1)$$

де $M_{\text{пр}i}$ і $F_{\text{пр}j}$ – навантаження пружного механічного зв'язку;

$\Delta \varphi_i$; і ΔS_j – деформація пружного елемента при оберальній і поступальній ході.

Маси елементів і жорсткості окремих зв'язків у кінематичному ланцюзі приводу можуть відрізнятися. Основний вплив на рух системи мають найбільші маси та найменші жорсткості зв'язків. Тому одним із перших кроків при проектуванні та дослідженні електроприводів є створення спрощених розрахункових схем механічної частини, що дозволяють ігнорувати пружність досить жорстких зв'язків і приблизно враховувати вплив малих мас. Важливо розуміти, що через наявність передач різні елементи системи можуть мати різні швидкості, тому їхні моменти інерції J_i , маси m_j , жорсткості зв'язків c_i і c_j , переміщення $\Delta \varphi_i$; і ΔS_j та інші параметри не можна безпосередньо порівнювати. Тому при складанні розрахункових схем механічної частини електроприводу необхідно привести всі параметри елементів до однієї розрахункової швидкості. Зазвичай зручніше привести їх до швидкості двигуна, але можливо також привести до швидкості будь-якого іншого елемента. Наприклад, при розв'язанні певних завдань корисно приводити до швидкості механізму, особливо під час його поступального руху.

Для того щоб розрахункова схема відповідала реальній механічній системі, необхідно дотримуватися закону збереження енергії. При приведенні потрібно забезпечити збереження кінетичної та потенційної енергії системи, а також елементарної роботи всіх сил і моментів, які діють у системі, під час можливих переміщень.

Формули приведення моменту інерції елемента системи, що обертається зі швидкістю ω_i , або маси, що переміщується поступально зі швидкістю V_j , до розрахункової швидкості ω_1 мають наступний вид:

$$J_{\text{пр}i} = J_i / i_{1i}^2, \quad J_{\text{пр}j} = m_j / \rho_{1j}^2, \quad (3.2)$$

де $i_{1i} = \omega_1 / \omega_i$ – передавальне число від валу приведення до i -го валу;
 $\rho_{1j} = V_j / \omega_1$ – радіус приведення до валу з швидкістю ω_1 .

При приведенні обертальних φ_i і поступальних S_j переміщень потрібно враховувати, що передавальне число та радіус приведення визначаються за відношенням швидкостей. Формули приведення переміщень виглядають так:

$$\Delta\varphi_{\text{пр}i} = \varphi_i i_{1i}, \quad \varphi_{\text{пр}j} = S_j / \rho_{1j}. \quad (3.3)$$

При приведенні жорсткостей механічних зв'язків слід дотримуватись умови збереження потенційної енергії. Формули приведення мають вигляд:

$$c_{\text{пр}i} = c_i / i_{1i}^2, \quad c_{\text{пр}j} = c_j / \rho_{1j}^2. \quad (3.4)$$

Приведення моментів і сил навантаження елементів кінематичного кола повинно базуватись на умові рівності елементарної роботи під час можливих переміщень:

$$M_{\text{пр}i} = M_i / i_{1i}, \quad M_{\text{пр}j} = F_j \rho_{1j}. \quad (3.5)$$

У процесі проектування та дослідження електроприводів моменти інерції, маси та жорсткості зв'язків реальних елементів зазвичай відомі. Сили, що діють

у системі, можуть бути або задані, або розраховані за початковими даними механізму та умовами технологічного процесу. Після приведення цих значень до розрахункової швидкості можна зіставити приведені значення моментів інерції та жорсткостей, що дозволить вибрати головні маси та основні пружні зв'язки і на їхній основі створити наближену розрахункову схему механічної частини. Для наочності, після приведення, можна побудувати початкову розрахункову схему, де маси представлені у вигляді прямокутників, площа яких пропорційна приведеним моментам інерції, а жорсткості зв'язків зображуються у вигляді з'єднань, довжина яких обернено пропорційна жорсткості (прямо пропорційна податливості зв'язків).

Для кінематичної схеми, представленої на рис. 1.1,а, розрахункова схема може виглядати так, як показано на рис. 1.1,б. У цьому прикладі виділено три основні маси - ротор двигуна з моментом інерції J_1 , барабан з приведеним моментом інерції $J_{пр n}$ і вантаж $J_{пр k}$. Аналізуючи цю схему, видно, що через незначні значення інших моментів інерції її можна спростити. Для цього необхідно додати малі маси до найближчих великих і визначити еквівалентні жорсткості зв'язків між цими масами за загальною формулою:

$$1 / c_{екв} = 1 / c_1 + 1 / c_2 + 1 / c_3 + \dots \quad (3.6)$$

На початковій розрахунковій схемі (рис. 1.1,б) стрілками позначені прикладені до окремих мас системи приведені моменти зовнішніх сил, що діють у системі, включаючи зовнішні сили $M_{пр i}$ і $M_{пр j}$. До ротора двигуна J_1 прикладені електромагнітний момент M_d та момент механічних втрат ΔM_d . Для правильного врахування напрямку моментів у схемі вказано позитивний напрямок швидкості ω_1 . При переході до спрощеної розрахункової схеми слід підсумувати всі зовнішні сили, прикладені до мас, і припустити, що зв'язки між ними є жорсткими.

Дослідження динаміки електроприводів показують, що в більшості практичних випадків розрахункові механічні схеми, що не мають розгалужень, після

виділення основних мас і жорсткостей зводяться до трьохмасової (рис. 3.2, а) і двомасової (рис. 3.2,б) розрахункових схем.

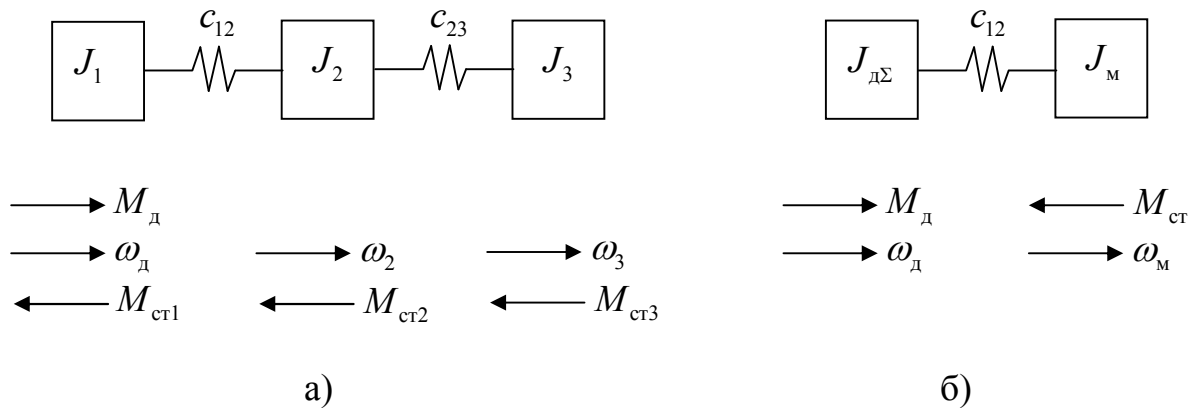


Рисунок 3.2 – Розрахункові схеми механічної частини електроприводу

а) – схема трьохмасової системи; б) – схема двомасової системи

Параметрами узагальненої трьохмасової механічної системи (розрахункова схема на рис. 3.2, а) є сумарні приведені моменти інерції мас J_1 , J_2 і J_3 , утворені приведеними масами, зв'язки між якими вважаються жорсткими, а також еквівалентні приведені жорсткості механічних пружних зв'язків між масами J_1 і J_2 – c_1 і між масами J_2 і J_3 – c_2 . Перша маса є ротором двигуна і жорстко з'єднана з ним; до цієї маси прикладені електромагнітний момент двигуна M і момент статичного навантаження $M_{ст1}$, який зазвичай є сумарним моментом втрат на валу двигуна і в жорстко з ним з'єднаних елементах.

До проміжної маси механізму (J_2) прикладений момент опору $M_{ст2}$, а до третьої маси (J_3) – момент зовнішнього навантаження $M_{ст3}$.

Трьохмасова пружна система використовується при дослідженні електромеханічних систем автоматизованого електроприводу в ситуаціях, коли необхідно провести детальний аналіз умов руху мас механізму. Для розв'язання таких завдань зазвичай застосовується математичне моделювання на обчислювальних системах. У разі дослідження окремих фізичних аспектів трьохмасова розрахункова схема може бути спрощена до двомасової.

У спрощеній двомасовій пружній системі (рис. 3.2, б) сумарний приведений момент інерції елементів, що жорстко з'єднані з двигуном, позначається як $J_{\text{д}\Sigma}$. Сумарний приведений момент інерції елементів, що жорстко з'єднані з робочим органом механізму, позначається як $J_{\text{м}}$. Швидкість механізму позначена як $\omega_{\text{м}}$. Безінерційний пружний зв'язок між цими масами характеризується еквівалентною приведеною жорсткістю c_{12} . Сумарний момент навантажень на валу двигуна $M_{\text{ст1}}$, який включає втрати на валу і в жорстко з ним з'єднаних елементах, у подальших розрахунках не враховується. Сумарний момент навантажень на валу механізму позначається як $M_{\text{ст}}$.

Електромеханічна система з двомасовою пружною механічною частиною є спрощеною моделлю електроприводу, яка найбільше підходить для вивчення впливу пружних механічних зв'язків, тому в цьому розділі вона є основним об'єктом дослідження.

3.2 Математична модель двомасової системи

Кінематична схема механічної частини електроприводу механізму підйому, приведена до двомасової системи, в загальному випадку з урахуванням моменту внутрішнього в'язкого тертя, описується рівняннями:

$$\begin{cases} J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma} - M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12}(\omega_{\text{д}} - \omega_{\text{м}}); \\ J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = M_{\text{д}} - M_{\Sigma}. \end{cases} \quad (3.7)$$

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

$$M_{\Sigma} = M_{\text{пр}} + \beta(\omega_{\text{д}} - \omega_{\text{м}}), \quad (3.8)$$

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

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. 3.3.

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

$$\begin{cases} \frac{d\omega_m}{dt} = \frac{1}{J_m} M_{\text{пр}} + \frac{\beta}{J_m} \omega_d - \frac{\beta}{J_m} \omega_m - \frac{1}{J_m} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_d - c_{12} \omega_m; \\ \frac{d\omega_d}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_d - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_d + \frac{\beta}{J_{\text{д}\Sigma}} \omega_m. \end{cases} \quad (3.9)$$

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

$$\begin{cases} \frac{d\omega_d}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_d - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dM_d}{dt} = \frac{1}{k_d} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{зе}} - \frac{k_d}{2T_{\mu\Gamma}^2} M_d - \frac{1}{T_{\mu\Gamma}} P + \frac{1}{T_{\mu\Gamma}^2 k_{\Gamma}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_a} U_{\text{зе}} - \frac{1}{T_a} E_{\text{зе}}; \\ \frac{dU_{\text{зе}}}{dt} = \frac{k_{\text{н}}}{T_a k_d} \omega_d - \frac{1}{T_a} U_{\text{зе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{зе}}. \end{cases} \quad (3.10)$$

При записі системи (3.10) прийняті наступні позначення.

U_{ie} – напруга на виході інтегральної частини регулятора ;

P – ривок.

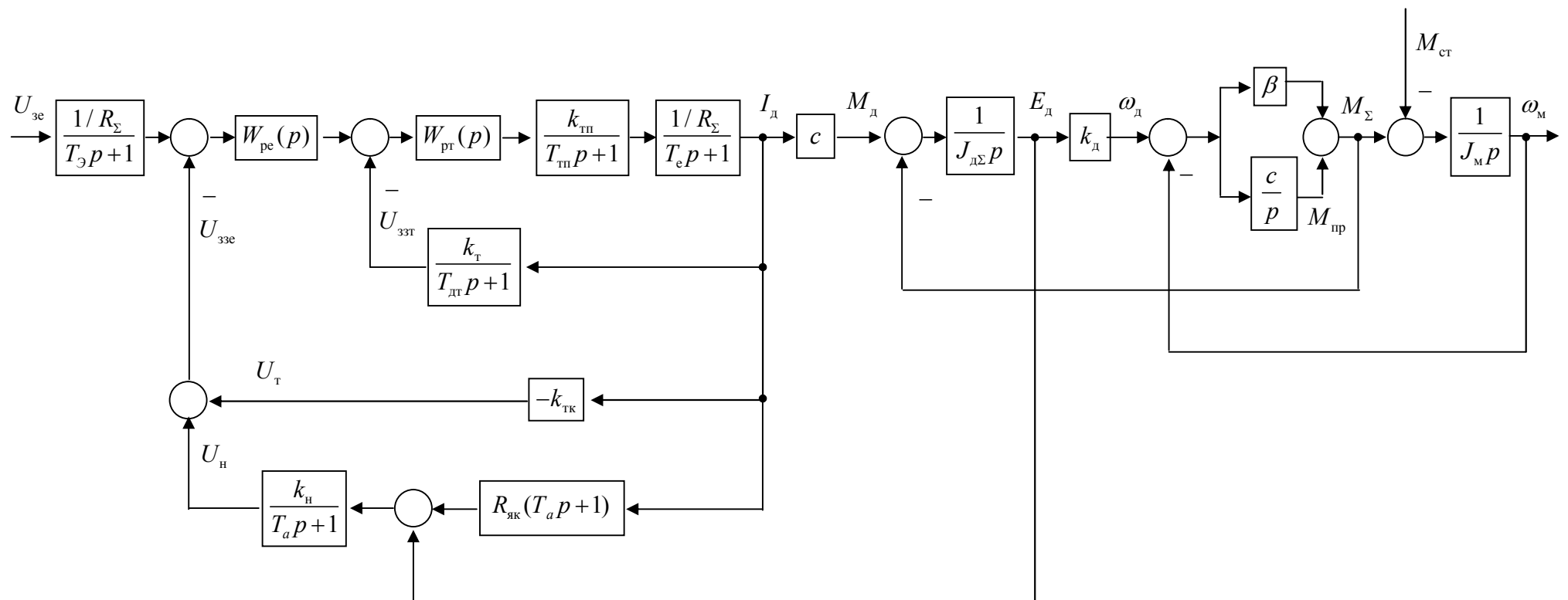


Рисунок 3.3 – Алгоритмічна схема двомасової системи

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

$$W_{\text{пе}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} \cdot \frac{4T_{\mu\text{е}} p + 1}{4T_{\mu\text{е}} p} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}} + \frac{T_{\text{м}} k_{\text{т}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}} \frac{1}{p} = k_{\text{пе}} + \frac{k_{\text{іе}}}{p}, \quad (3.11)$$

$$k_{\text{пе}} = \frac{T_{\text{м}} k_{\text{т}}}{2T_{\mu\text{е}} k_{\text{н}} R_{\Sigma}}, \quad k_{\text{іе}} = \frac{T_{\text{м}} k_{\text{т}}}{8T_{\mu\text{е}}^2 k_{\text{н}} R_{\Sigma}}. \quad (3.12)$$

При записі системи (3.4) замість $T_{\text{м}}$ записано відповідний вираз

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

Об'єднаємо системи (3.8) і (3.10), одержимо наступну систему рівнянь стану двомасової системи підлеглого регулювання швидкості із зворотним зв'язком по ЕРС двигуна:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J_{\text{м}}} \omega_{\text{м}} - \frac{1}{J_{\text{м}}} M_{\text{ст}}; \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{1}{J_{\text{д}\Sigma}} M_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{д}} - \frac{\beta}{J_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dM_{\text{д}}}{dt} = \frac{1}{k_{\text{д}}} P; \\ \frac{dP}{dt} = \frac{k_{\text{пе}}}{2T_{\mu\text{т}}^2 k_{\text{т}}} E_{\text{зе}} - \frac{k_{\text{пе}}}{2T_{\mu\text{т}}^2 k_{\text{т}}} U_{\text{ззе}} - \frac{k_{\text{д}}}{2T_{\mu\text{т}}^2} M_{\text{д}} - \frac{1}{T_{\mu\text{т}}} P + \frac{1}{T_{\mu\text{т}}^2 k_{\text{т}}} U_{\text{іе}}; \\ \frac{dE_{\text{зе}}}{dt} = \frac{1}{T_{\text{а}}} U_{\text{зе}} - \frac{1}{T_{\text{а}}} E_{\text{зе}}; \\ \frac{dU_{\text{ззе}}}{dt} = \frac{k_{\text{н}}}{T_{\text{а}} k_{\text{д}}} \omega_{\text{д}} - \frac{1}{T_{\text{а}}} U_{\text{ззе}}; \\ \frac{dU_{\text{іе}}}{dt} = k_{\text{іе}} E_{\text{зе}} - k_{\text{іе}} U_{\text{ззе}}. \end{array} \right. \quad (3.13)$$

Алгоритмічна схема двомасової системи, що відповідає рівнянням (3.13), показана на рис. (3.4). На схемі оптимізований контур струму представлений у вигляді послідовного з'єднання аперіодичної і інтегруючої ланки, охоплених одиничним зворотним зв'язком. Сигнал на виході першої ланки пропорційний ривку електроприводу P .

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

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

$$\vec{X}(t) = \{ \omega_m(t), M_{\text{пр}}(t), \omega_d(t), M_d(t), P(t), E_{\text{зе}}(t), U_{\text{зе}}(t), U_{\text{іе}}(t) \}^T, \quad (3.14)$$

$$\vec{U}(t) = \{ U_{\text{зе}}(t), M_{\text{ст}}(t) \}^T, \quad (3.15)$$

де T – знак транспонування.

Рівняння стану системи і рівняння виходу має вигляд

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.16)$$

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 вид, показаний нижче.

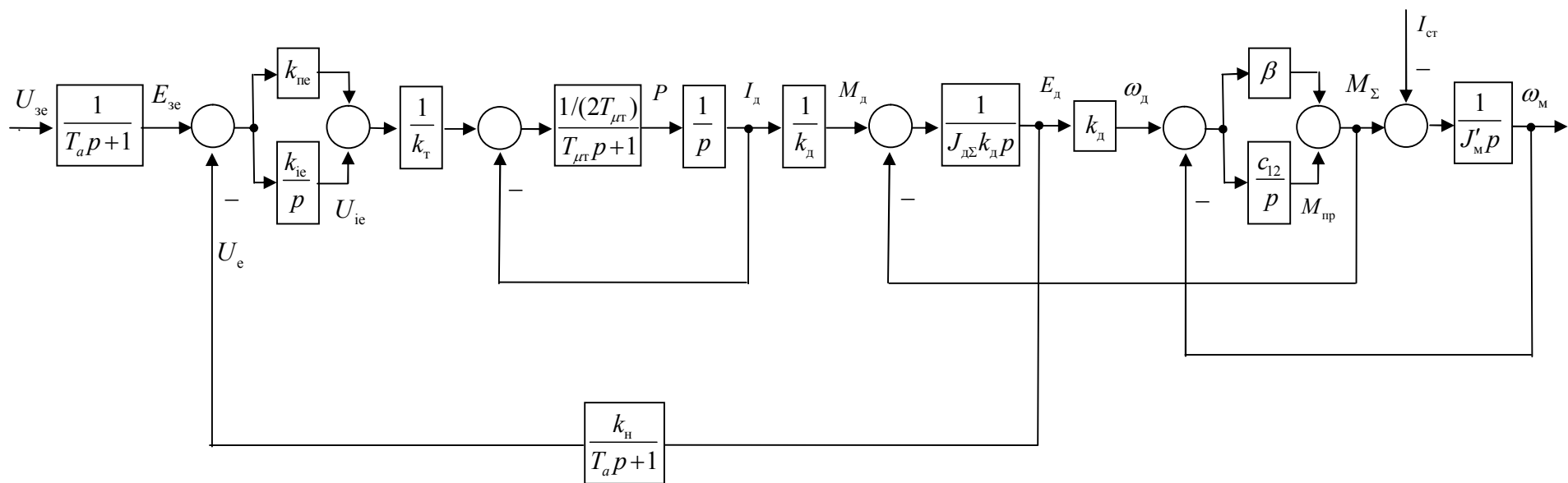


Рисунок 3.4 – Спрощена алгоритмічна схема двомасової системи

$$\mathbf{A} = \begin{array}{c} \begin{array}{cccccccc} \omega_{\text{м}} & M_{\text{пп}} & \omega_{\text{д}} & M_{\text{д}} & P & E_{\text{з\text{е}}} & U_{\text{е}} & U_{\text{и\text{е}}} \end{array} \\ \begin{array}{|c|c|c|c|c|c|c|c|} \hline -\frac{\beta}{J_{\text{м}}} & \frac{1}{J_{\text{м}}} & \frac{\beta}{J_{\text{м}}} & 0 & 0 & 0 & 0 & 0 \\ \hline -c & 0 & c & 0 & 0 & 0 & 0 & 0 \\ \hline \frac{\beta}{J_{\text{д}}} & -\frac{1}{J_{\text{д}}} & -\frac{\beta}{J_{\text{д}}} & \frac{1}{J_{\text{д}}} & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & \frac{1}{k_{\text{д}}} & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & \frac{-k_{\text{д}}}{2T_{\text{мт}}^2} & -\frac{1}{T_{\text{мт}}} & \frac{k_{\text{п\text{е}}}}{2T_{\text{мт}}^2 k_{\text{т}}} & -\frac{k_{\text{п\text{е}}}}{2T_{\text{мт}}^2 k_{\text{т}}} & \frac{1}{2T_{\text{мт}}^2 k_{\text{т}}} \\ \hline 0 & 0 & 0 & 0 & 0 & -\frac{1}{T_a} & 0 & 0 \\ \hline 0 & 0 & \frac{k_{\text{н}}}{T_a k_{\text{д}}} & 0 & 0 & 0 & -\frac{1}{T_a} & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & k_{\text{и\text{е}}} & -k_{\text{и\text{е}}} & 0 \\ \hline \end{array} \end{array}$$

$$\mathbf{B} = \begin{array}{|c|c|} \hline 0 & -\frac{1}{J_{\text{м}}} \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \frac{1}{T_a} & 0 \\ \hline 0 & 0 \\ \hline 0 & 0 \\ \hline \end{array}$$

3.3 Розрахунок перехідних процесів двомасової систем на ЕОМ

Визначимо параметри, необхідні для розрахунку перехідних процесів двомасової системи,

Момент інерції двигуна і жорстко пов'язаних з ним елементів системи визначений раніше $J_{\text{д}\Sigma} = 1,1 \cdot J_{\text{дв}} = 1,1 \cdot 12,42 = 14,9 \text{ кг} \cdot \text{м}^2$

Момент інерції вантажу визначасмо як

$$J_{\text{м}} = \frac{(G_{\text{вп}} + m_0) \cdot V_{\text{п}}^2}{\omega_{\text{н}}^2}, \quad (7.17)$$

$$J_{\text{м}} = \frac{(41000 + 1000) \cdot 0,1^2}{58,6^2} = 0,18 \text{ кг} \cdot \text{м}^2.$$

Значення інших параметрів системи визначені в попередніх розділах.

Розрахунок перехідних процесів проводився з використанням пакету прикладних програм MATLAB. Дана система дозволяє проводити розрахунок перехідних процесів по передавальних функціях системи, по векторно-матричному рівнянню, з використанням схеми моделі системи, показаної на рис. 3.5.

У додатку Г 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. 3.6 - рис. 3.9. Для зменшення інформації, що виводиться, на ЕОМ були виведені на екран монітора, і відповідно роздруковані перші чотири змінні стану. Як видно з графіків характер перехідних процесів швидкості механізму (вантаж) і моменту пружності по задаючій дії і перехідні процеси всіх змінних стану по збурюючій дії мають характер слабо затухаючих коливань.

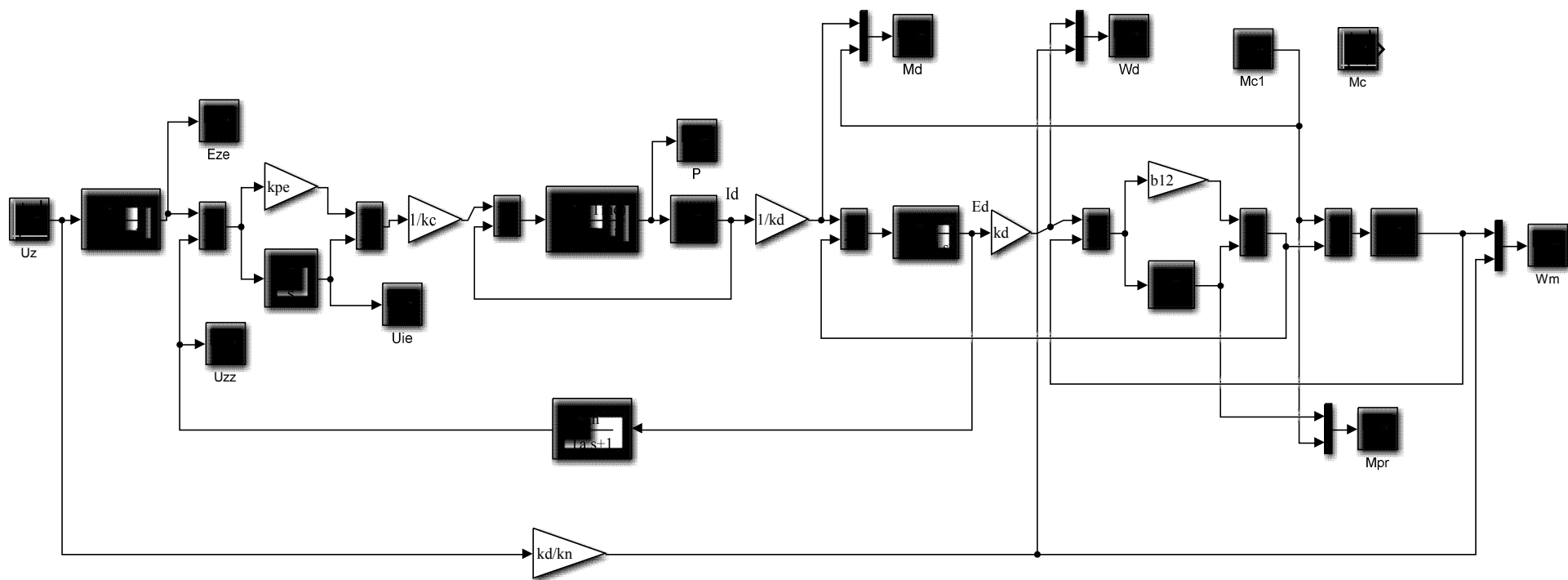


Рисунок 3.5 – Схема моделі двомасової системи, розроблена на EOM

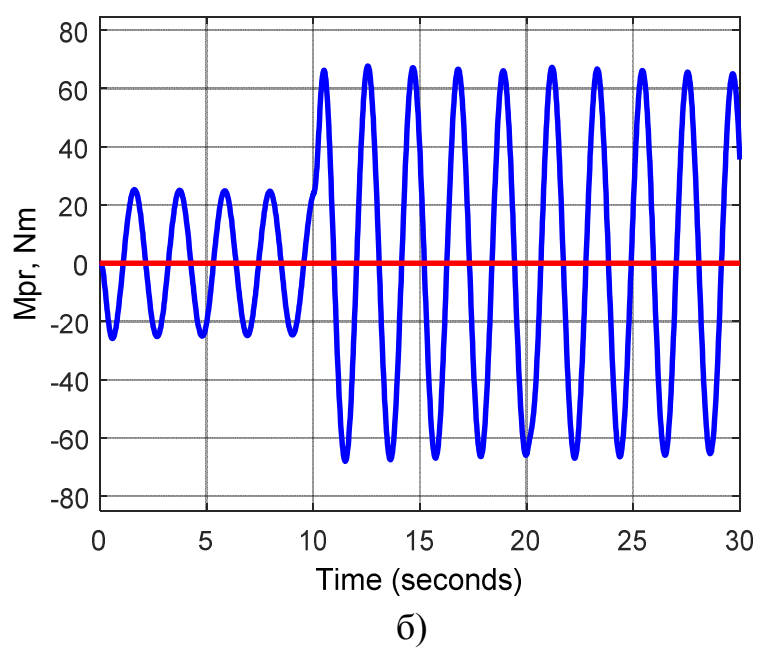
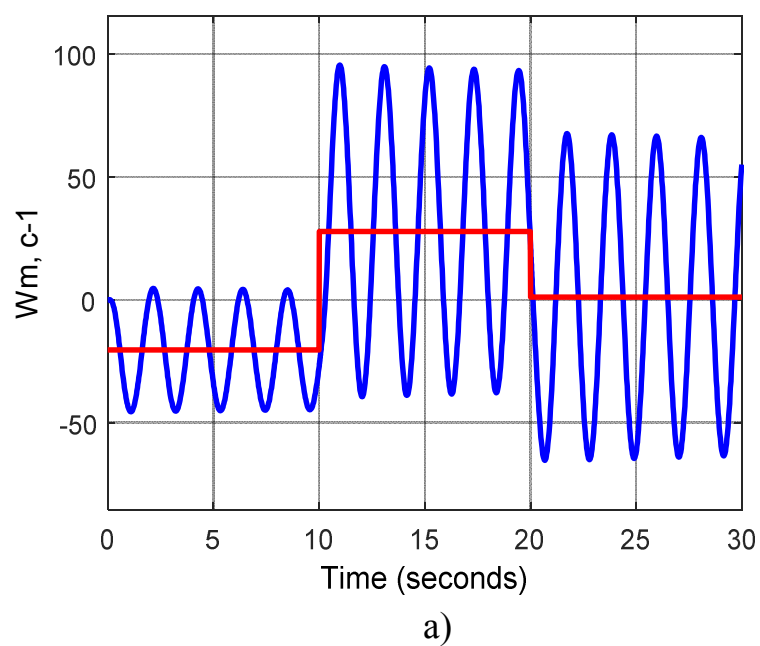


Рисунок 3.6 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

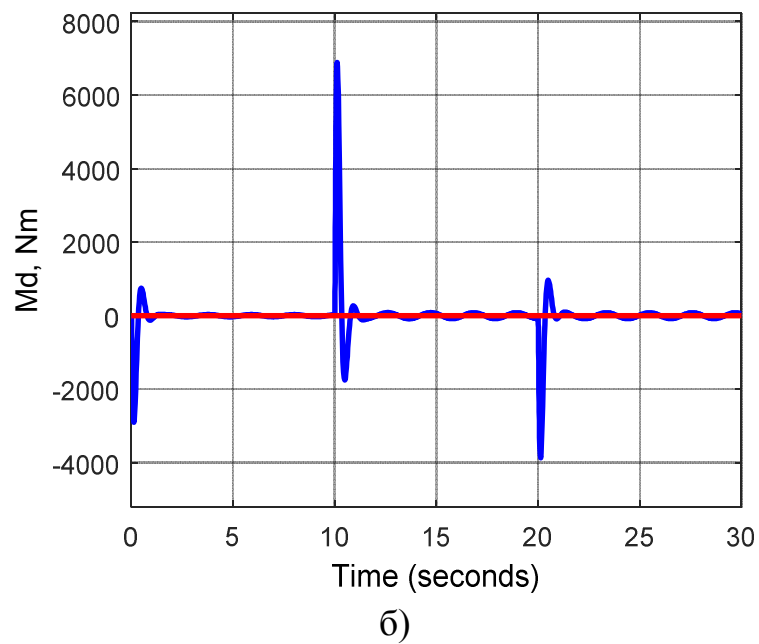
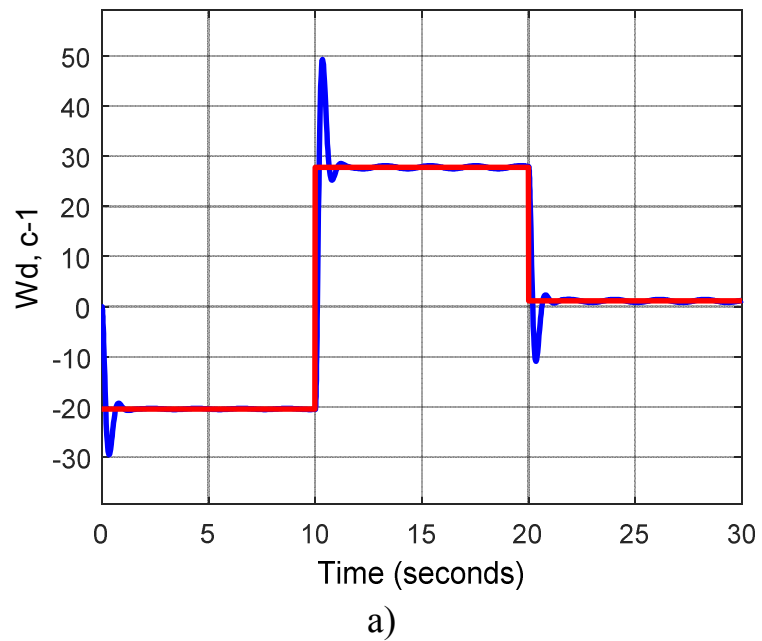
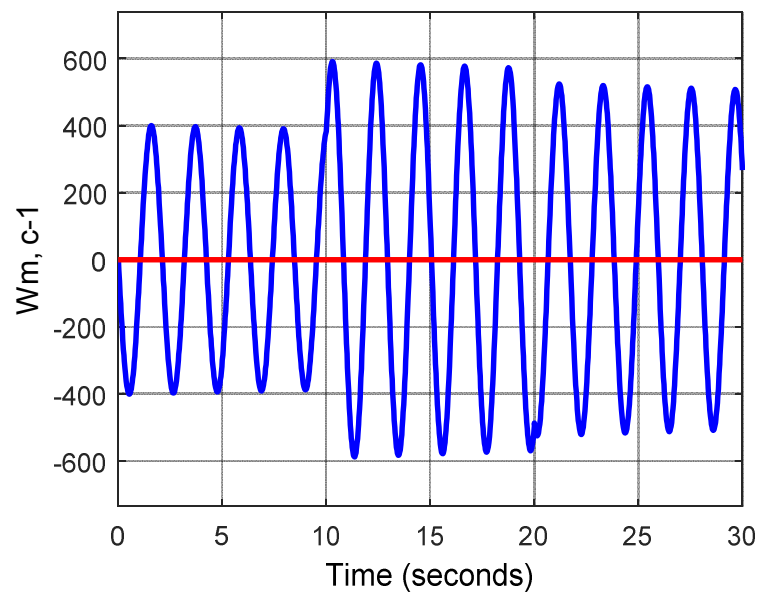
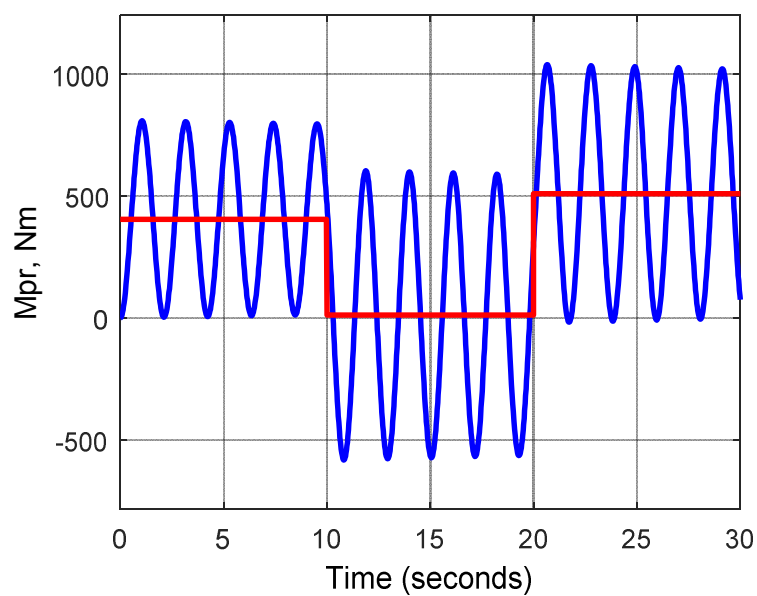


Рисунок 3.7 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по задаючій дії



a)



б)

Рисунок 3.8 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по збурюючій дії

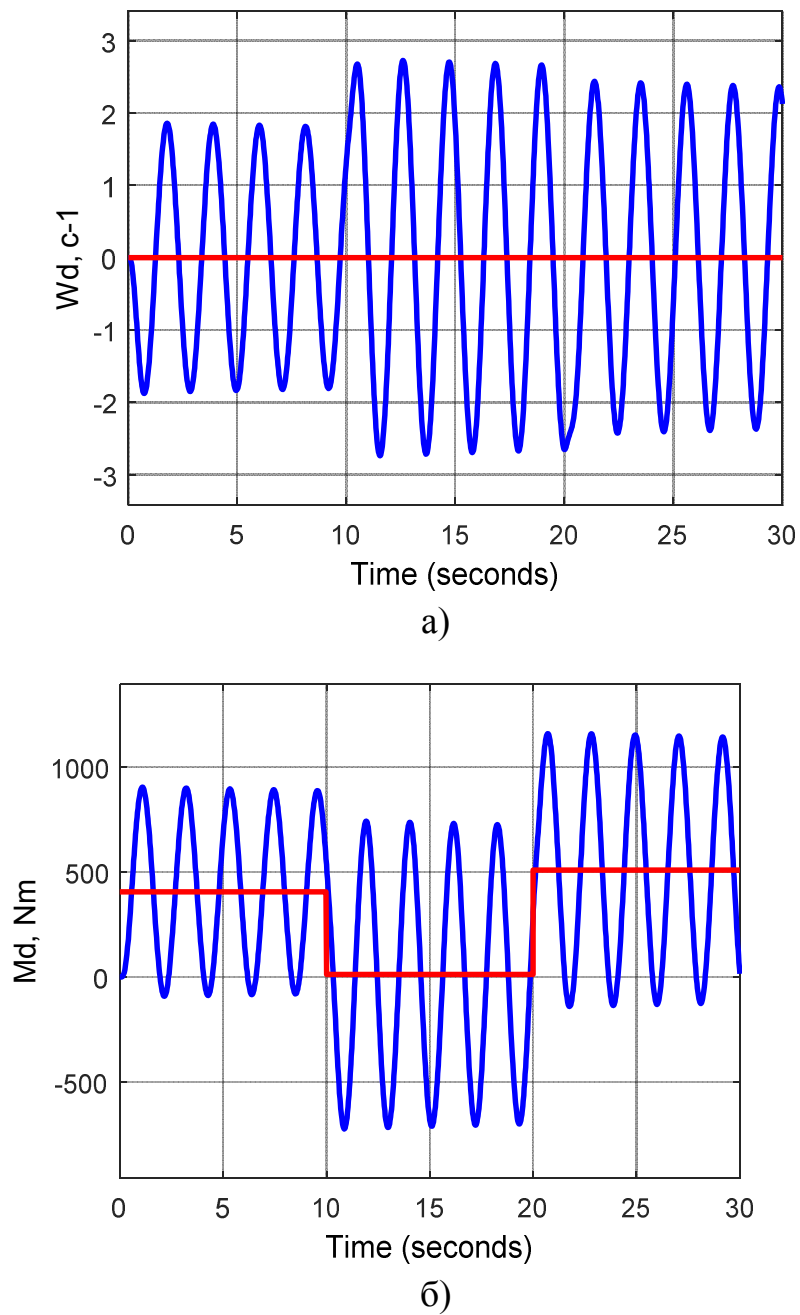


Рисунок 3.9 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено дослідження роботи підйомного механізму мостового крана з урахуванням кінцевої жорсткості підйомного канату. Аналізувалася кінематична схема електроприводу підйомного механізму. Виявлено, що механічна частина електроприводу в загальному випадку являє собою систему зв'язаних мас, які рухаються з різними обертальними або поступальними швидкостями. У разі навантаження елементи системи зазнають деформацій, оскільки механічні зв'язки не є абсолютно жорсткими. При зміні навантаження маси можуть переміщуватися одна відносно одної, при цьому переміщення визначається жорсткістю зв'язку. Аналіз динаміки електроприводу показав, що розрахункові механічні схеми, після виділення основних мас і жорсткостей, зводяться до трьохмасових або двомасових моделей. Електромеханічна система з двомасовою пружною механічною частиною є спрощеною моделлю електроприводу, яка найбільш зручна для дослідження впливу пружних механічних зв'язків і тому є основним об'єктом даного дослідження.

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

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

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

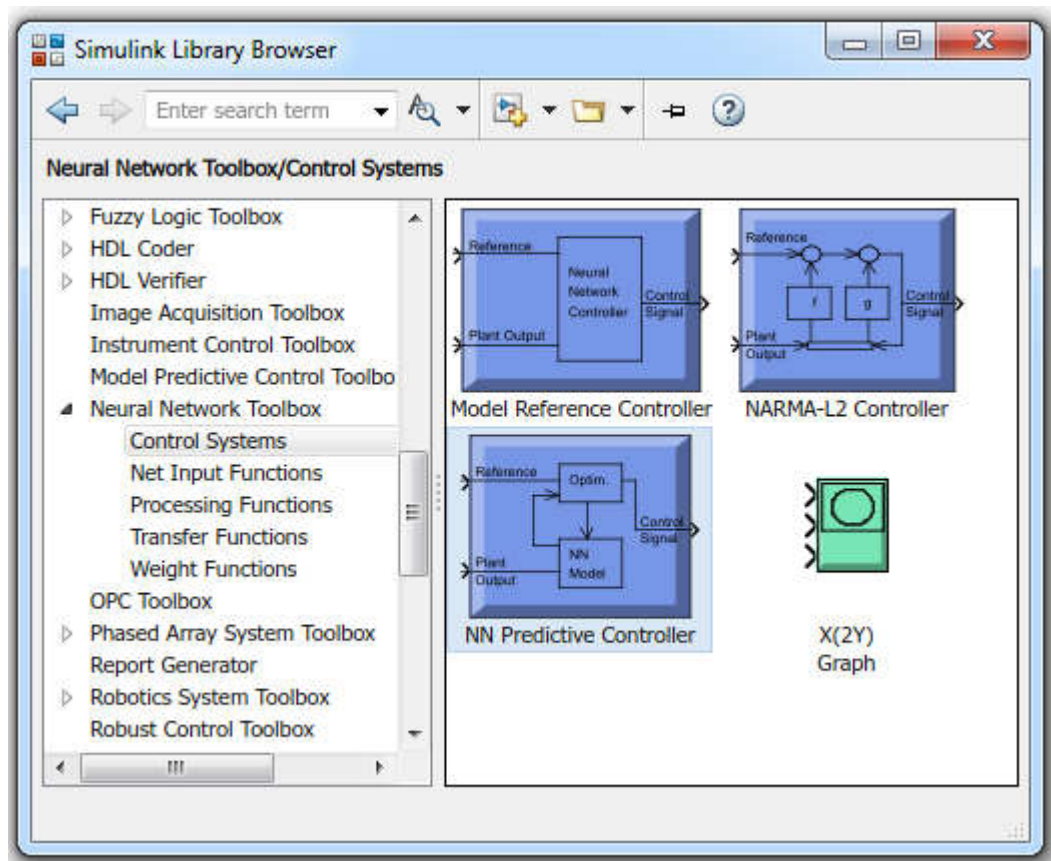


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

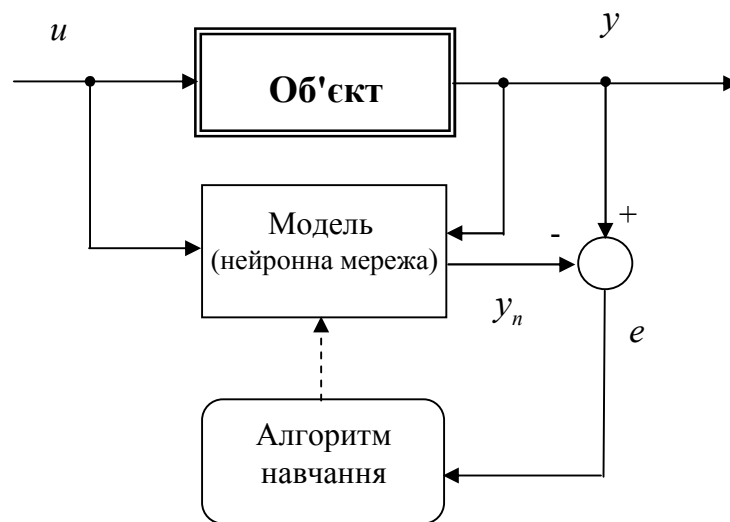


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

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

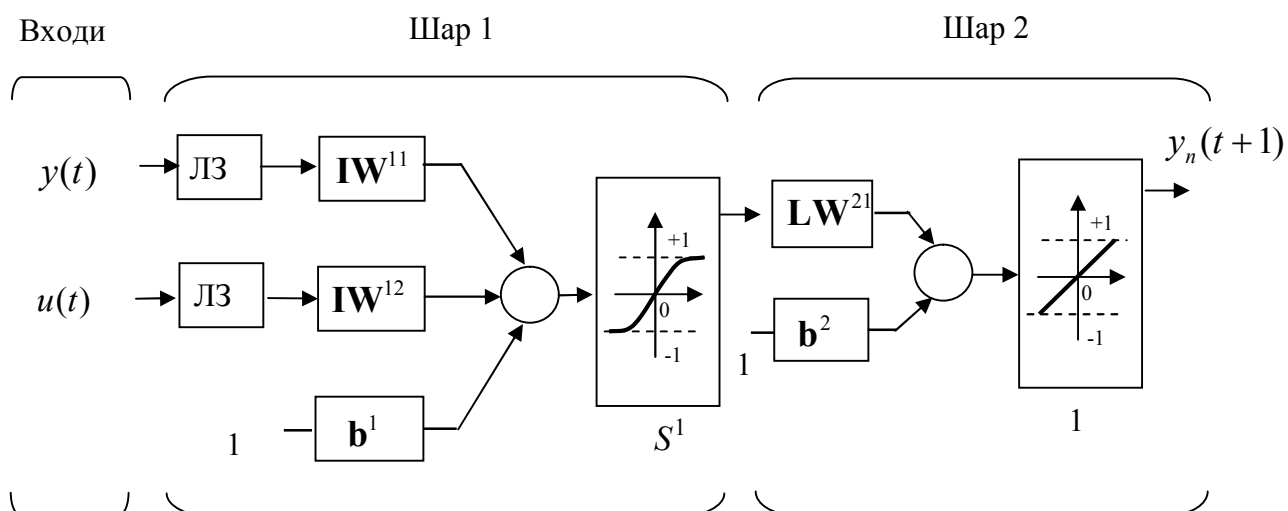


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

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

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

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

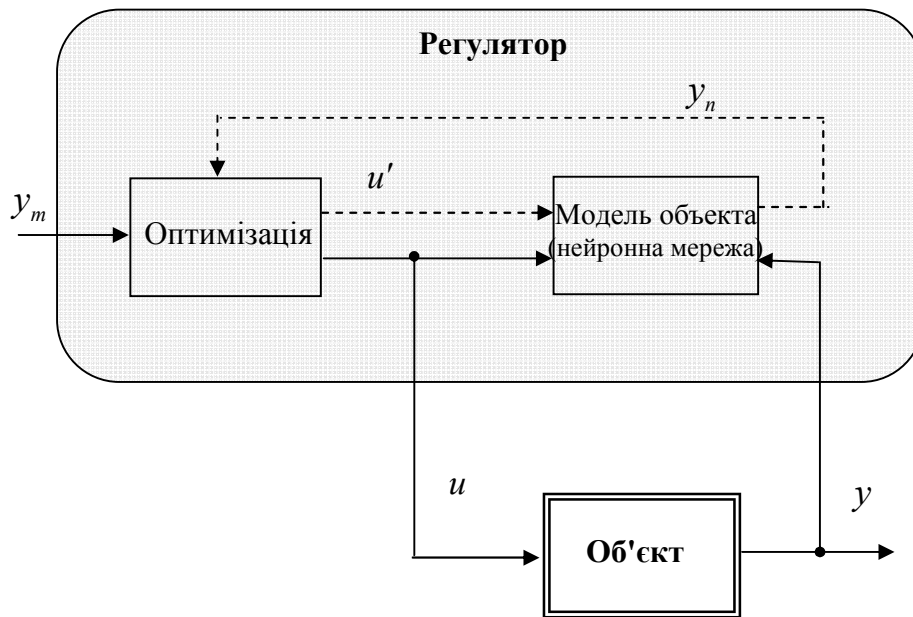


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

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

Допоміжні функції:

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

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

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

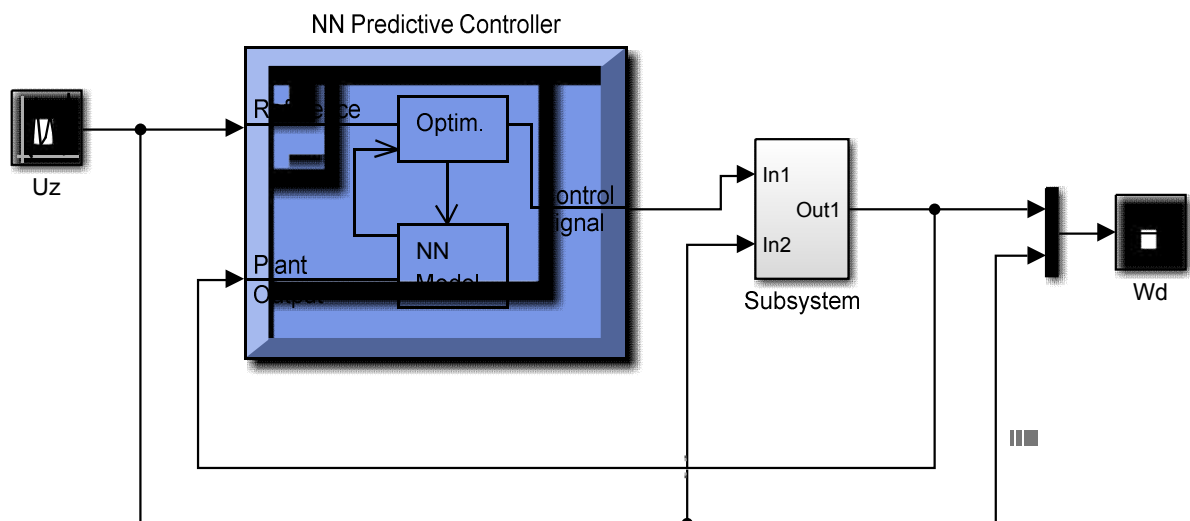


Рисунок 4.5 – Схема системи управління з нейрорегулятором

NN Predictive Controller

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

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

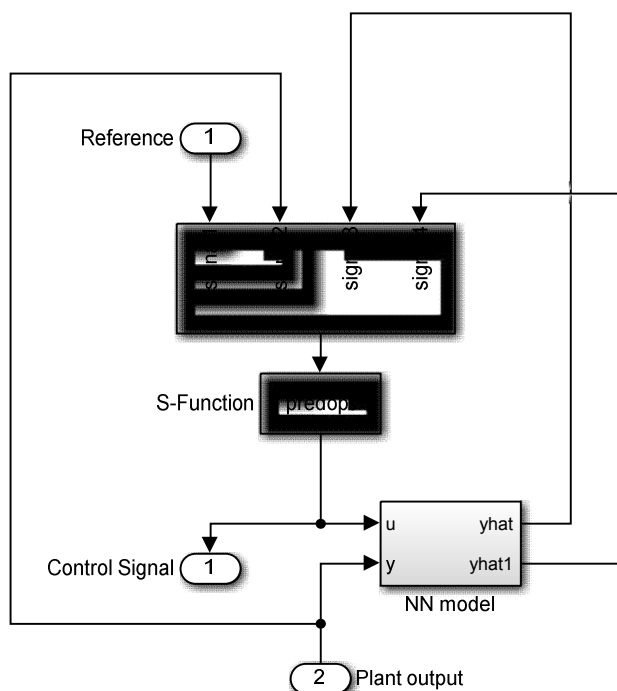


Рисунок 4.6 – Структурна схема нейрорегулятора

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**.

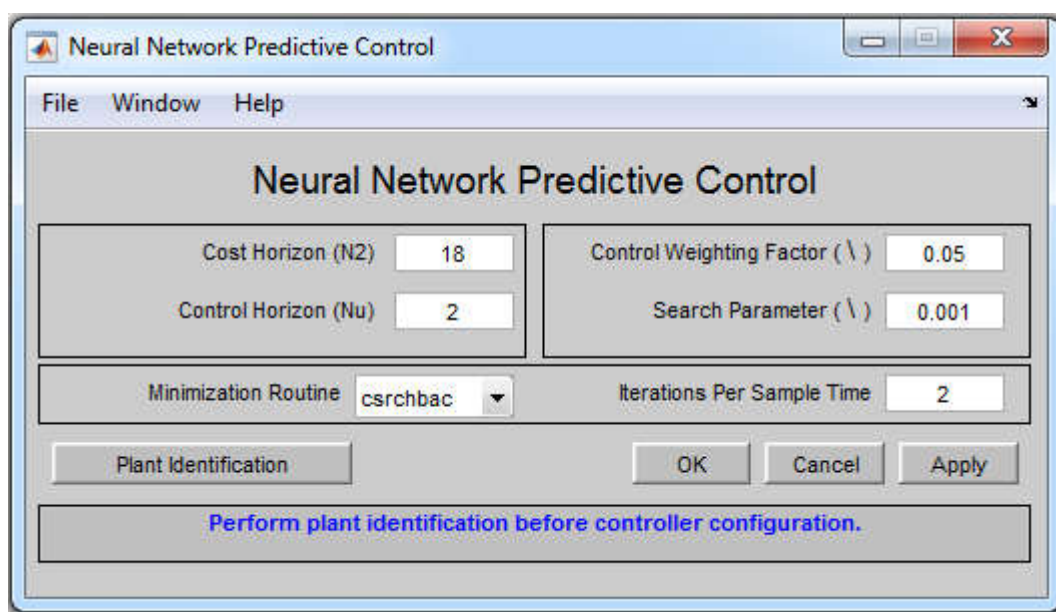


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

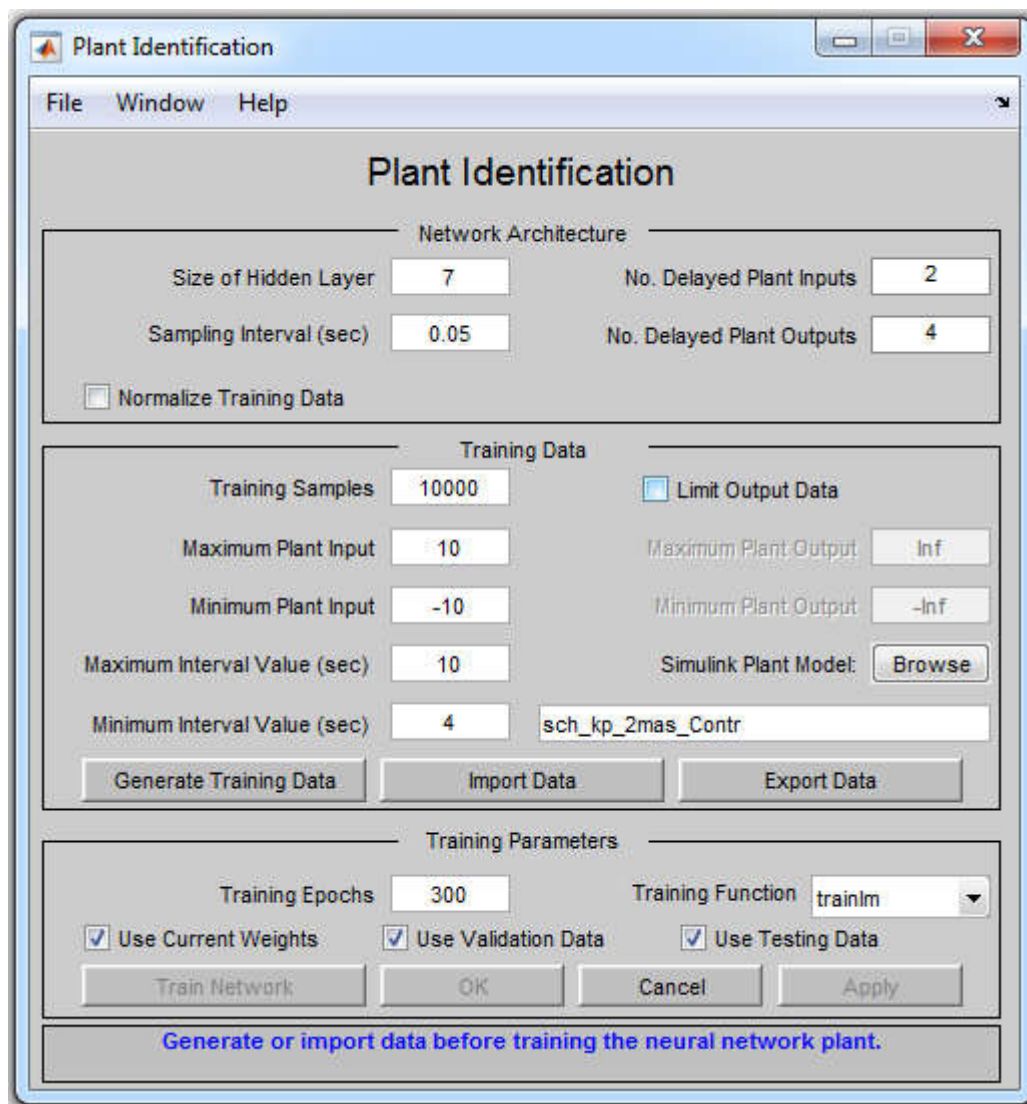


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

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

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

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

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

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

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

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

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

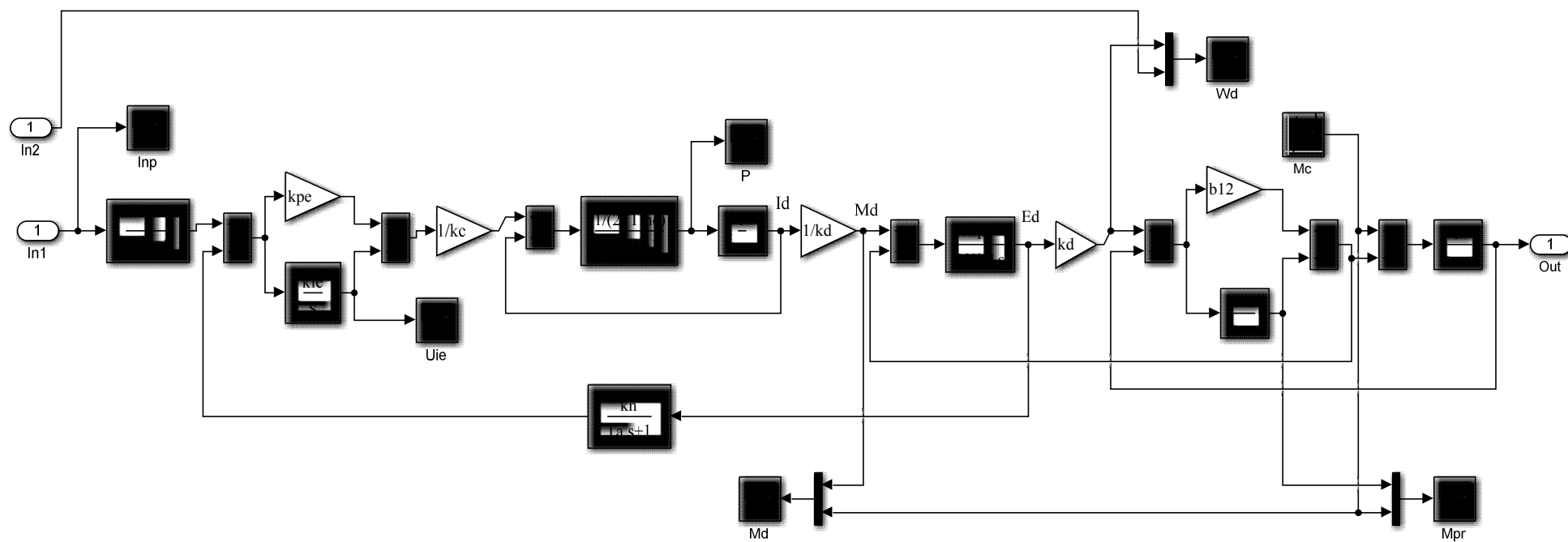


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

менших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

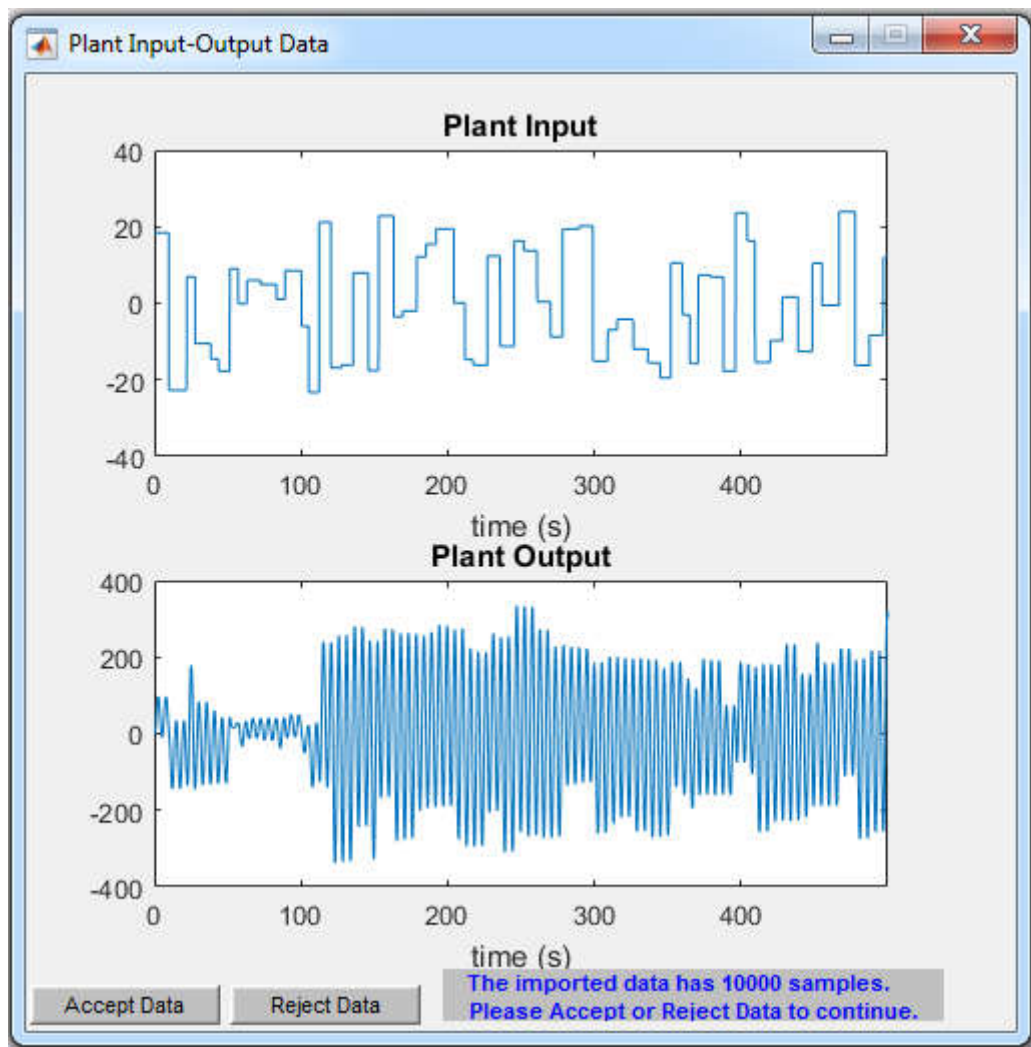


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

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

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

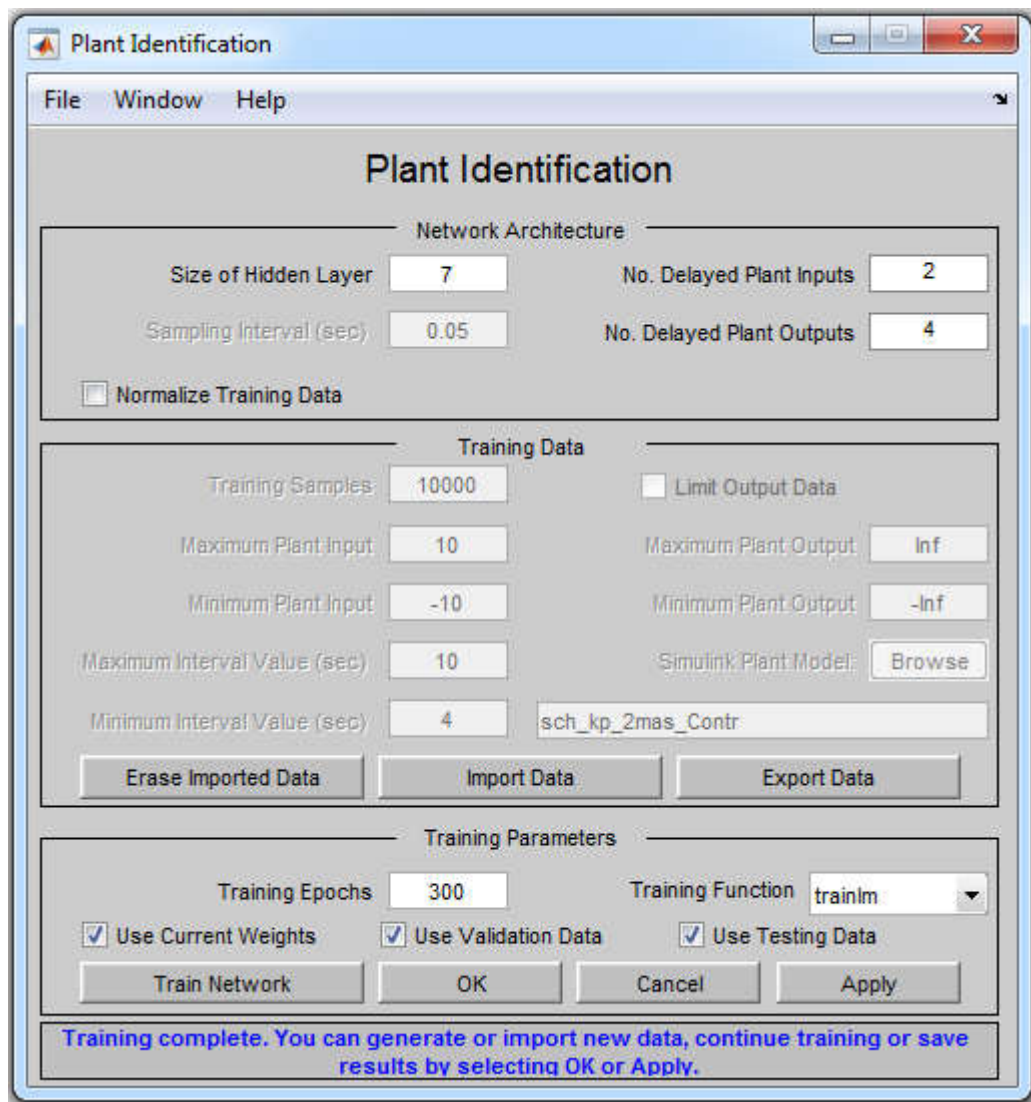


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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

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

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

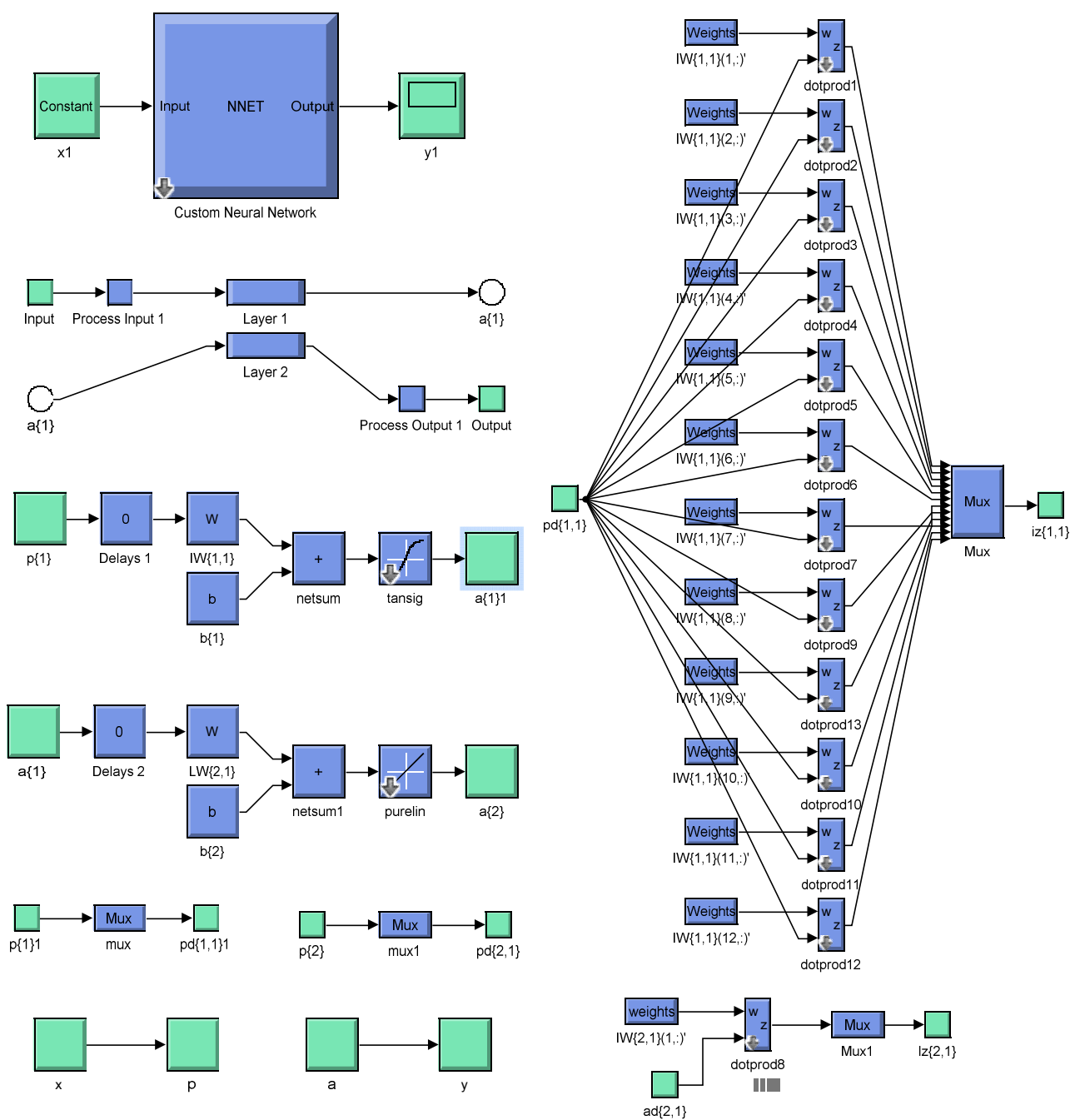


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

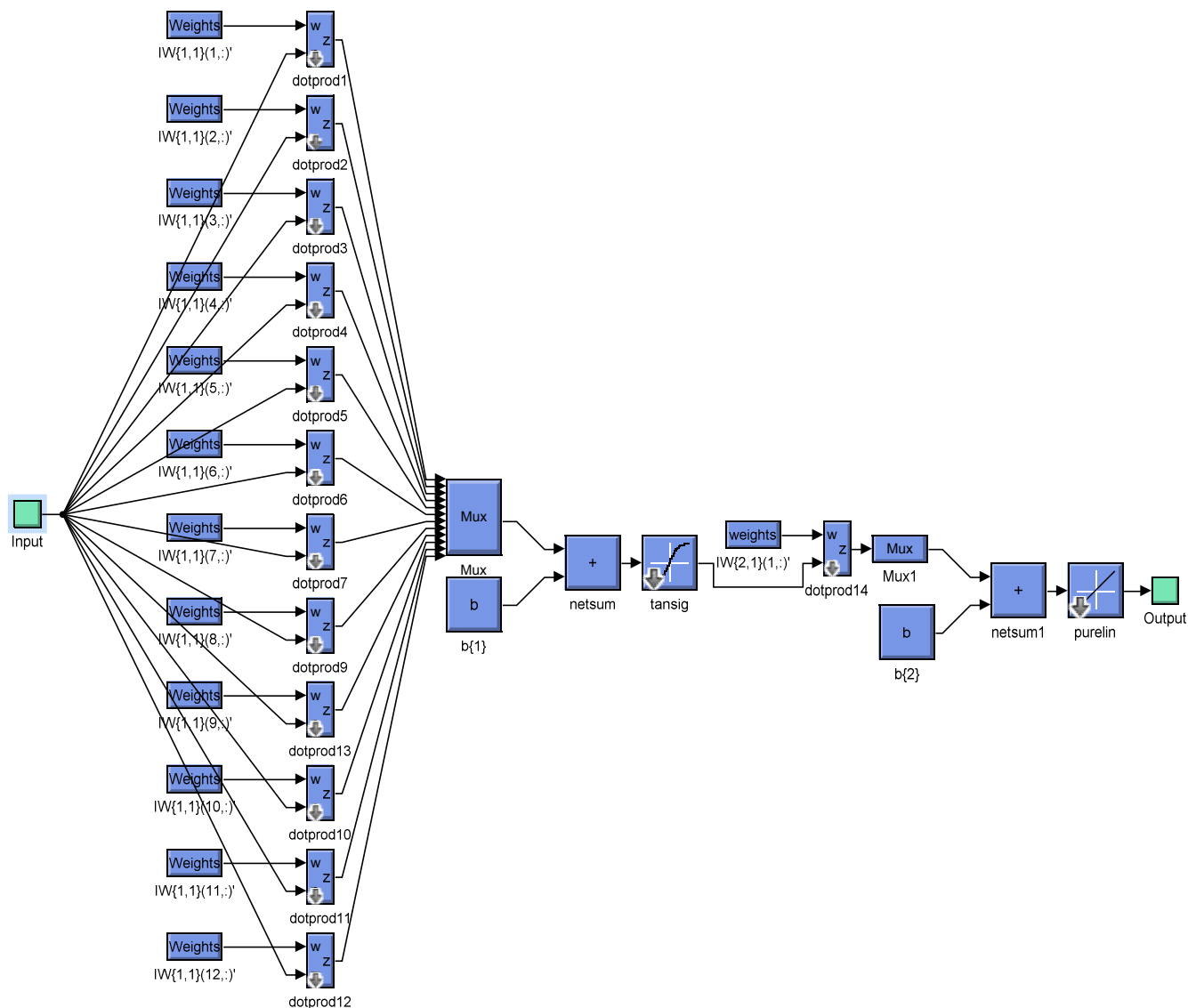


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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

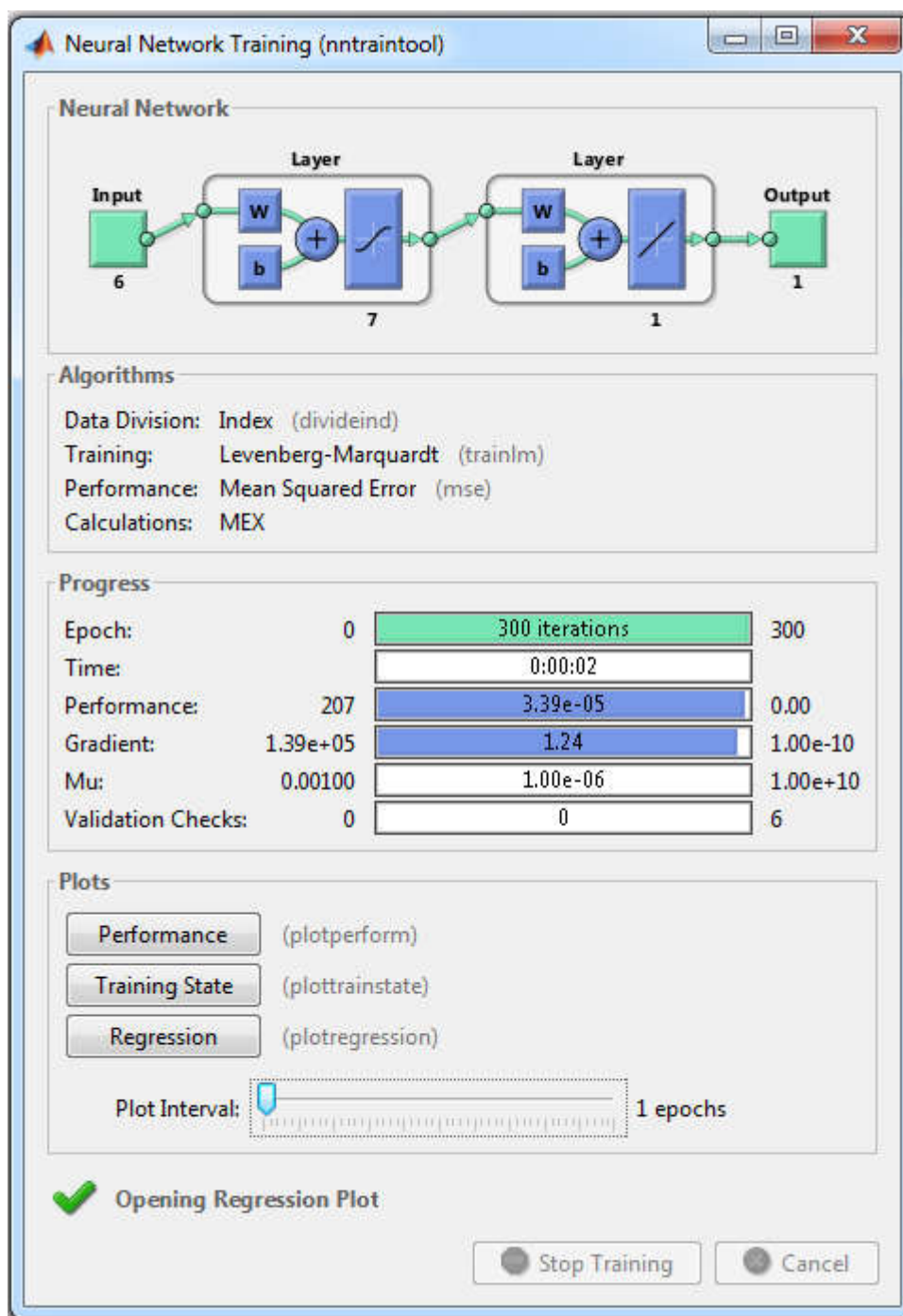


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

Опис вікна Neural Network Training (nntraintool)

Вікно Neural Network Training (nntraintool) має наступні секції.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

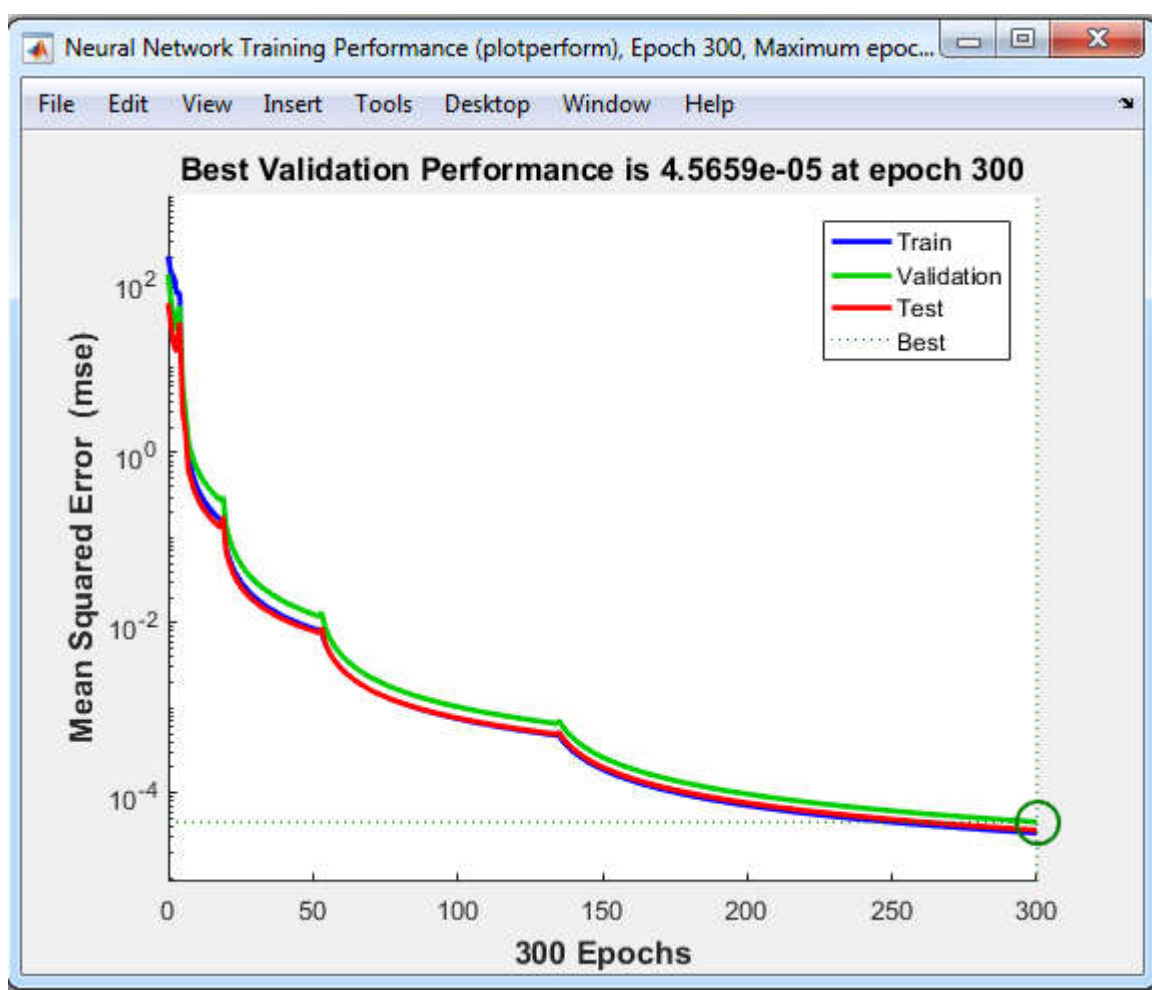


Рисунок 4.15 – Вікно контролю процесу навчання

У вікні Neural Network Training Performance відображаються графіки **Mean Squared Error (mse)**, які показують, як змінюється середньоквадратична помилка на навчальних, валідаційних (перевірочних) та тестових наборах даних

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

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

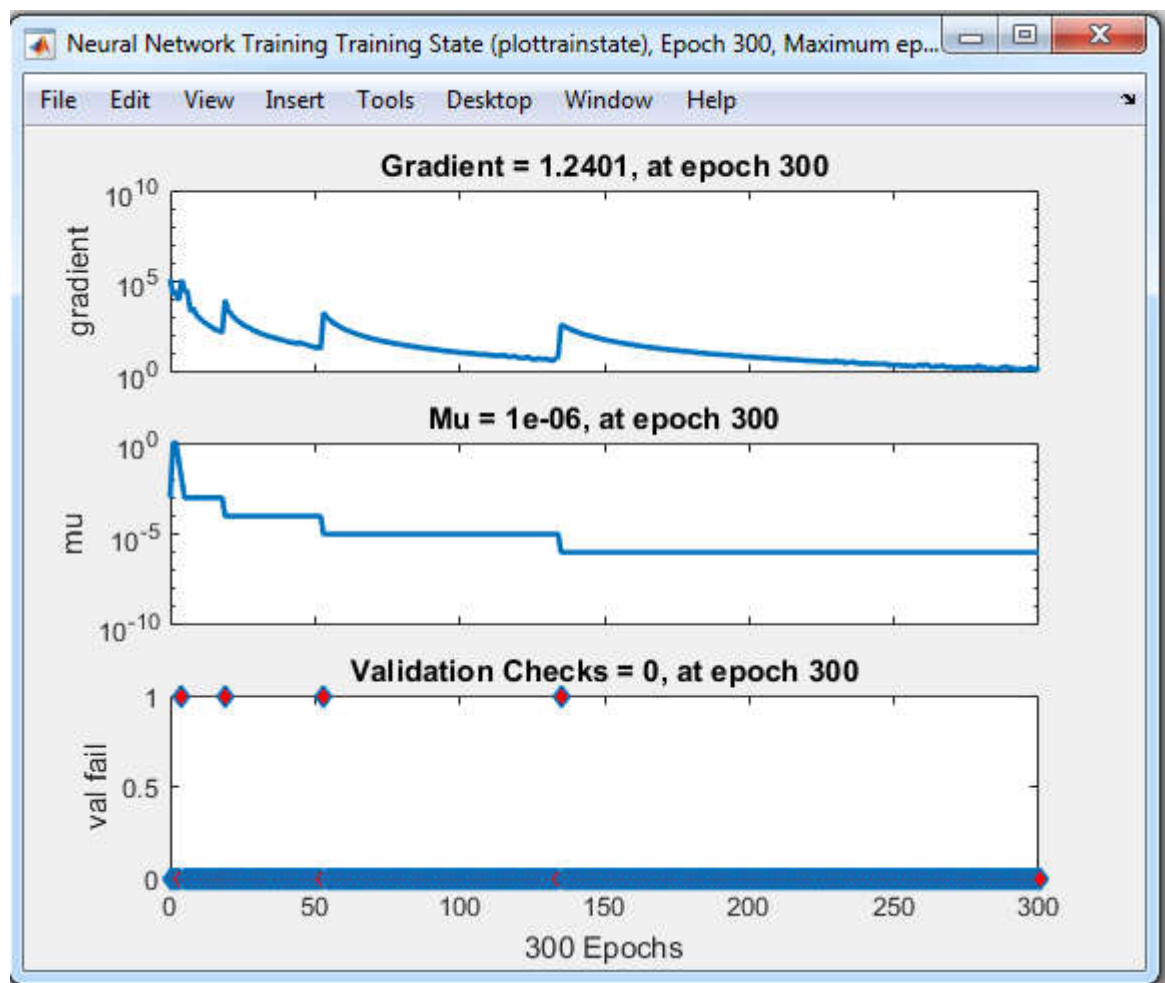


Рисунок 4.16 - Вікно Neural Network Training State

Опис вікна Neural Network Training State

У вікні Neural Network Training State відображаються наступні графіки.

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

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

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

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

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

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

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

Зміна μ показує, як модифікується швидкість навчання під час процесу навчання. Зазвичай, μ може збільшуватися або зменшуватися в залежності від реакції нейронної мережі на дані. Наприклад, велике значення " μ " може вказувати на те, що швидкість навчання збільшилася для прискорення збіжності,

але при цьому може виникнути ризик нестабільності. Зменшення ти може вказувати на те, що навчання сталося занадто швидко або навіть зупинилося.

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

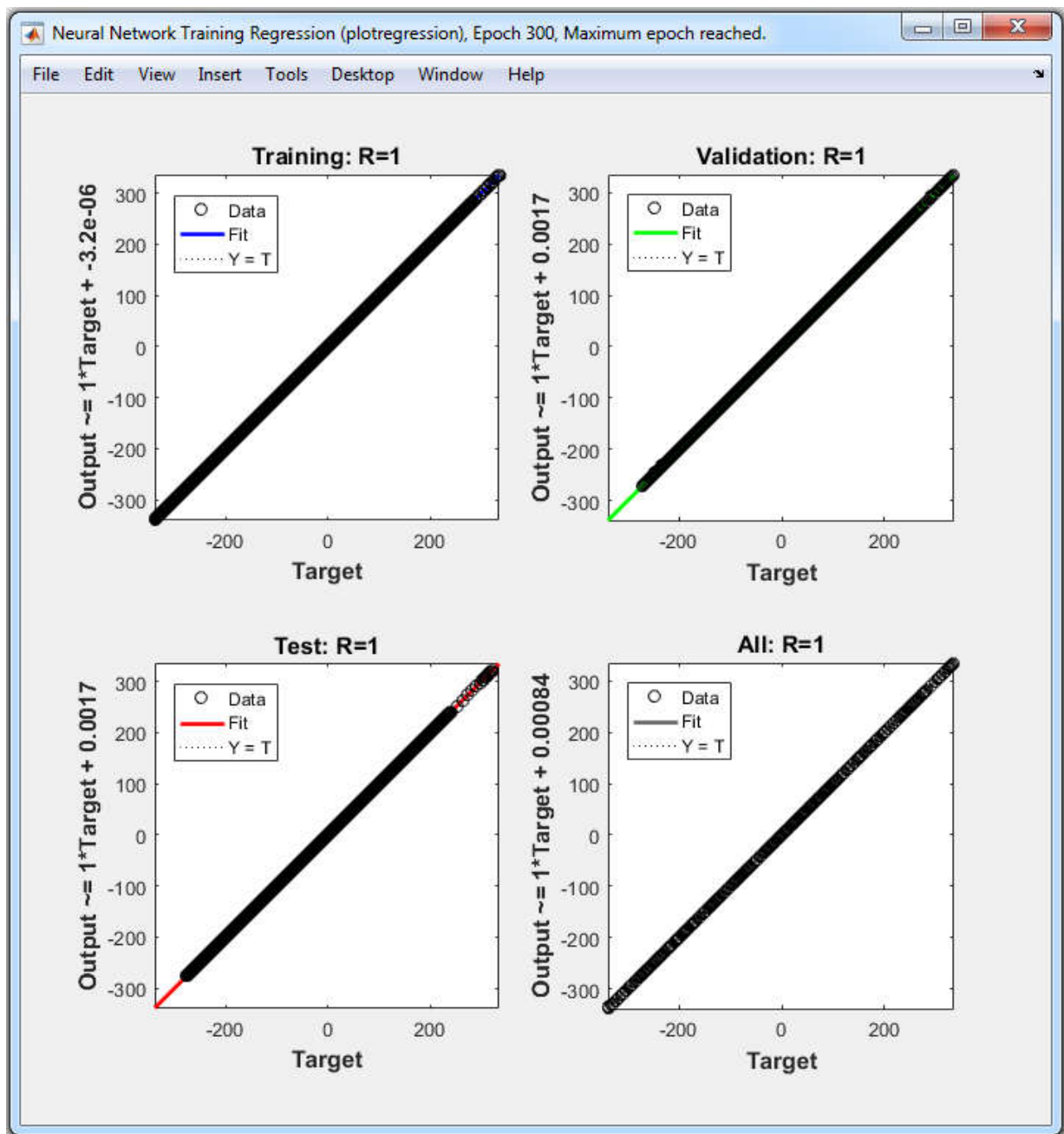


Рисунок 4.17 - Вікно Neural Network Training Regression

Onuc вікна Neural Neural Network Training Regression

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

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

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

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

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

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

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

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

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

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

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

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

Продовження опису вікна Neural Network Training (nntraintool)

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

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

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

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

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

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

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

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

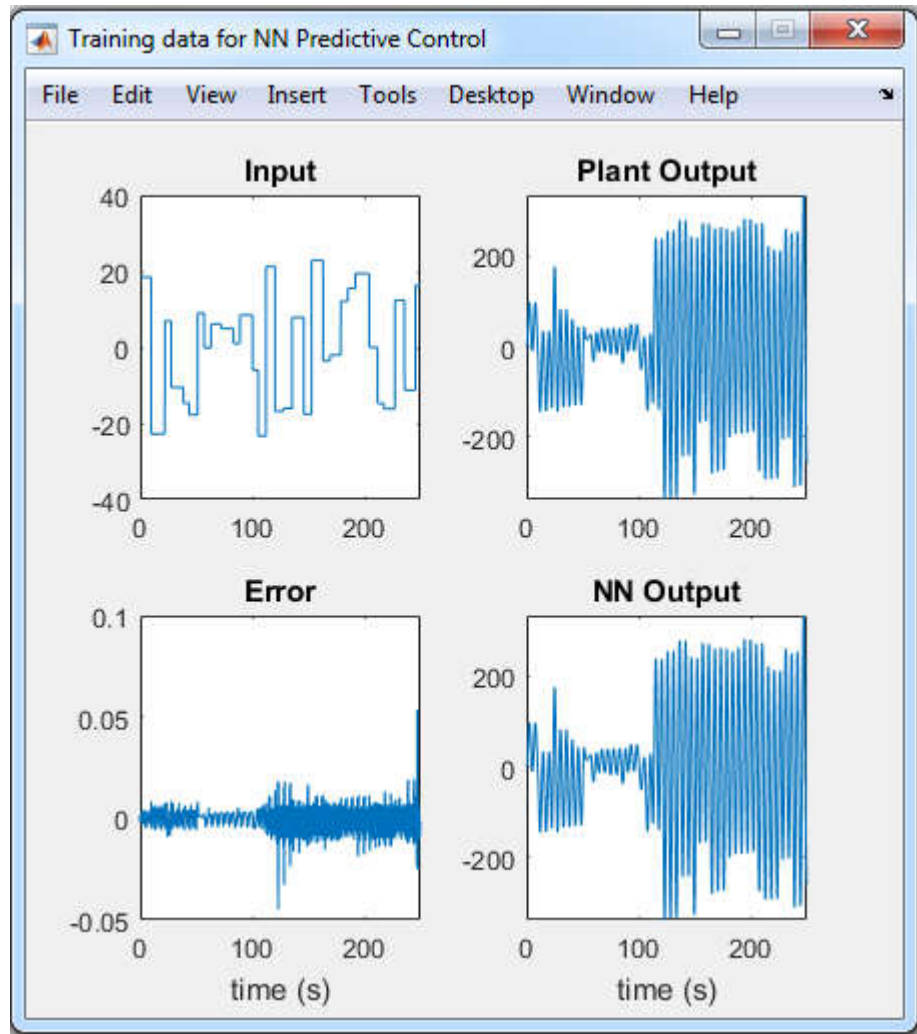


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

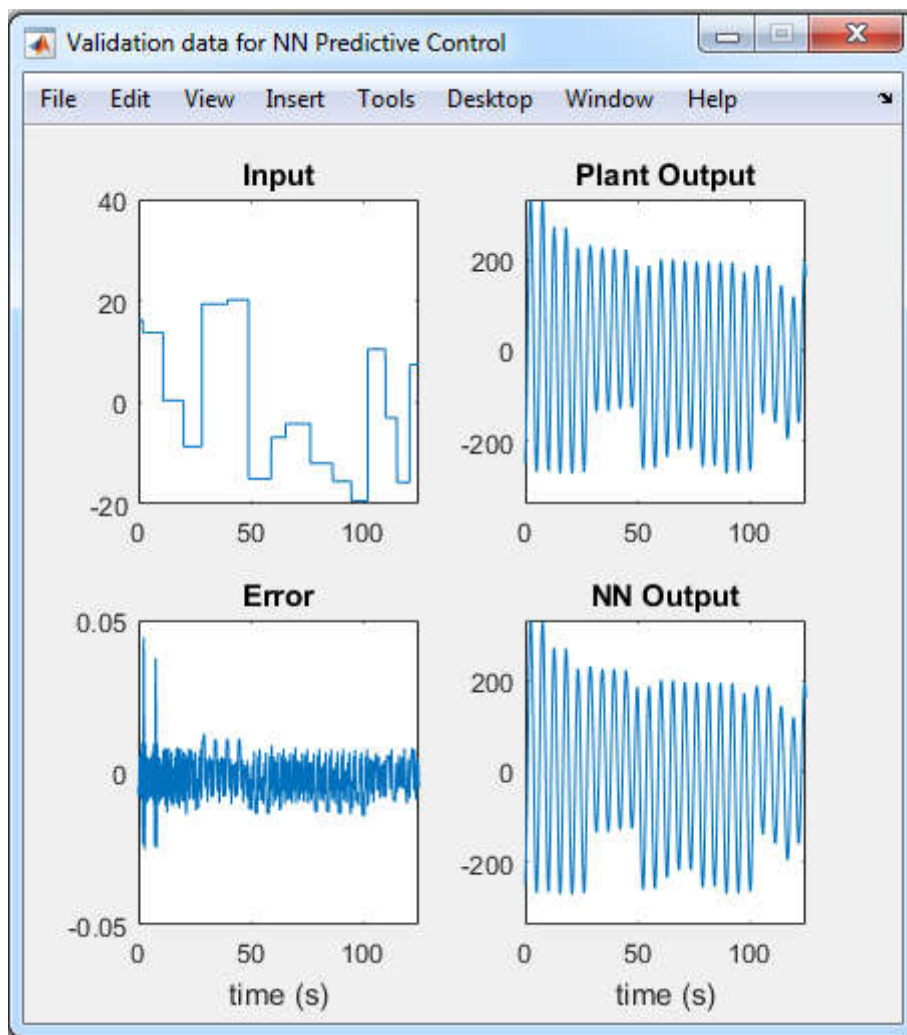


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

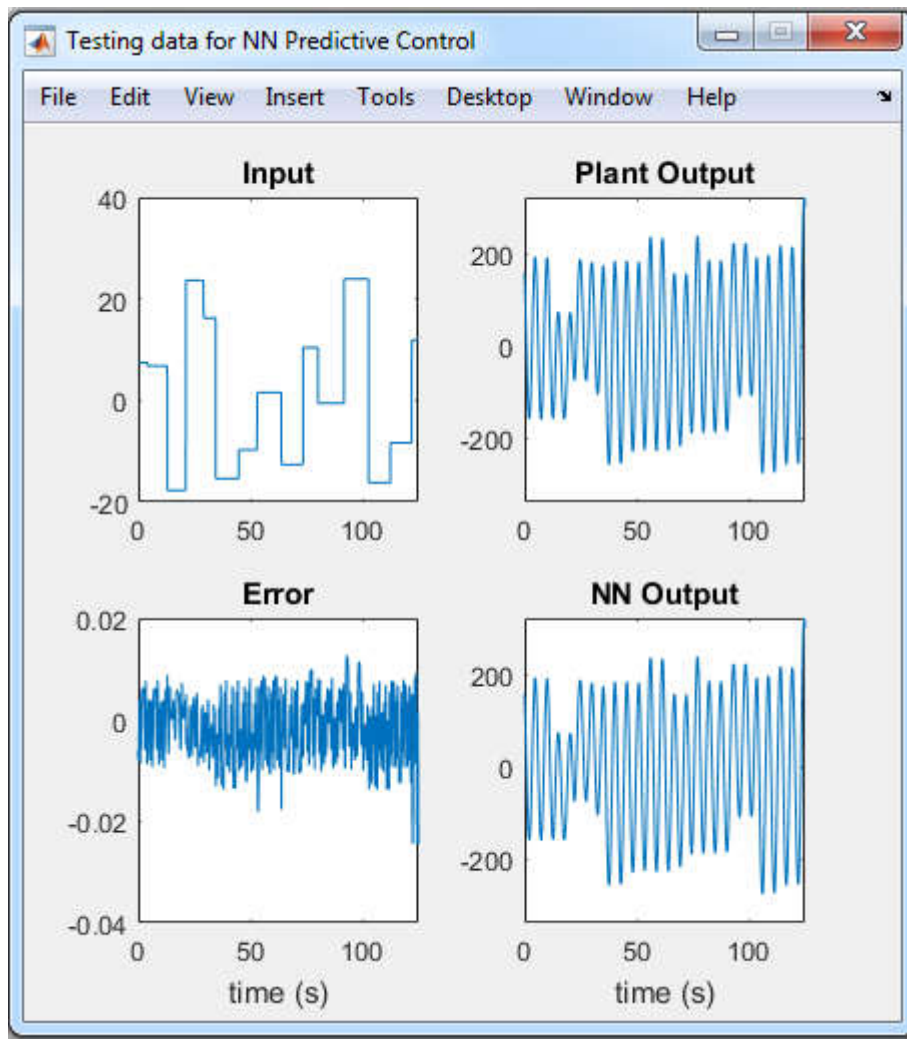


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

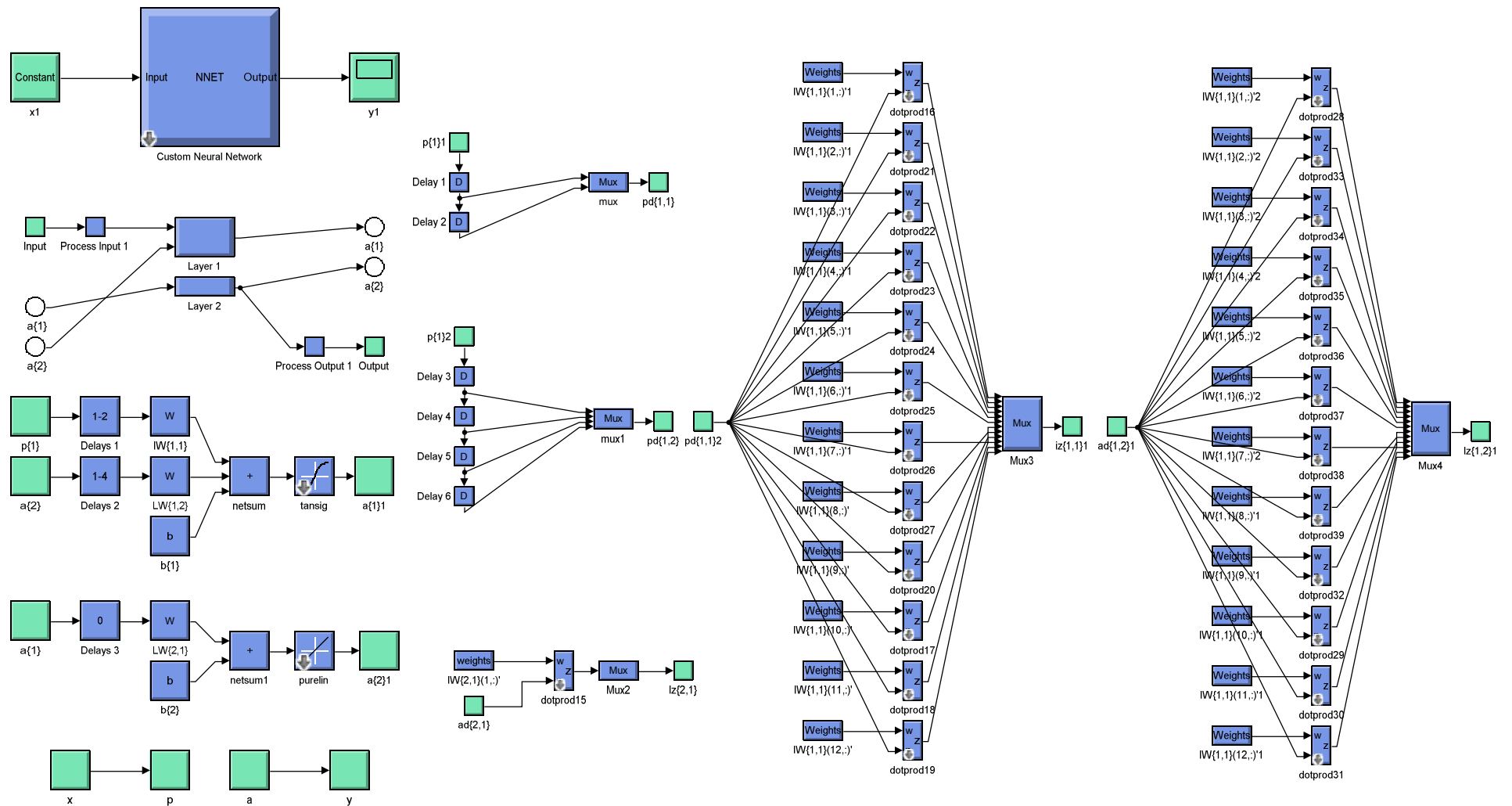


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

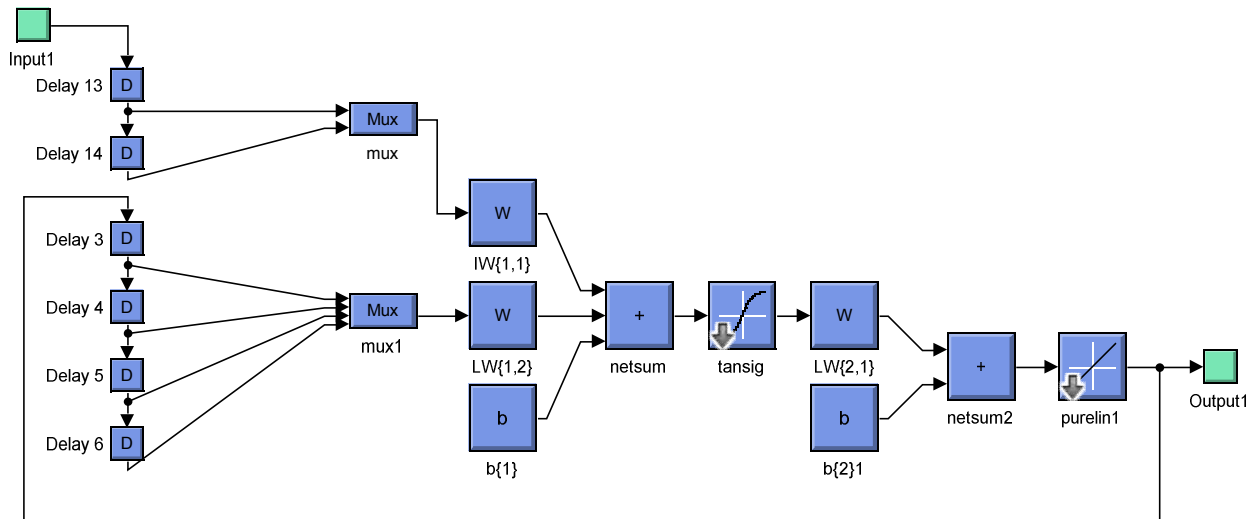


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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.

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

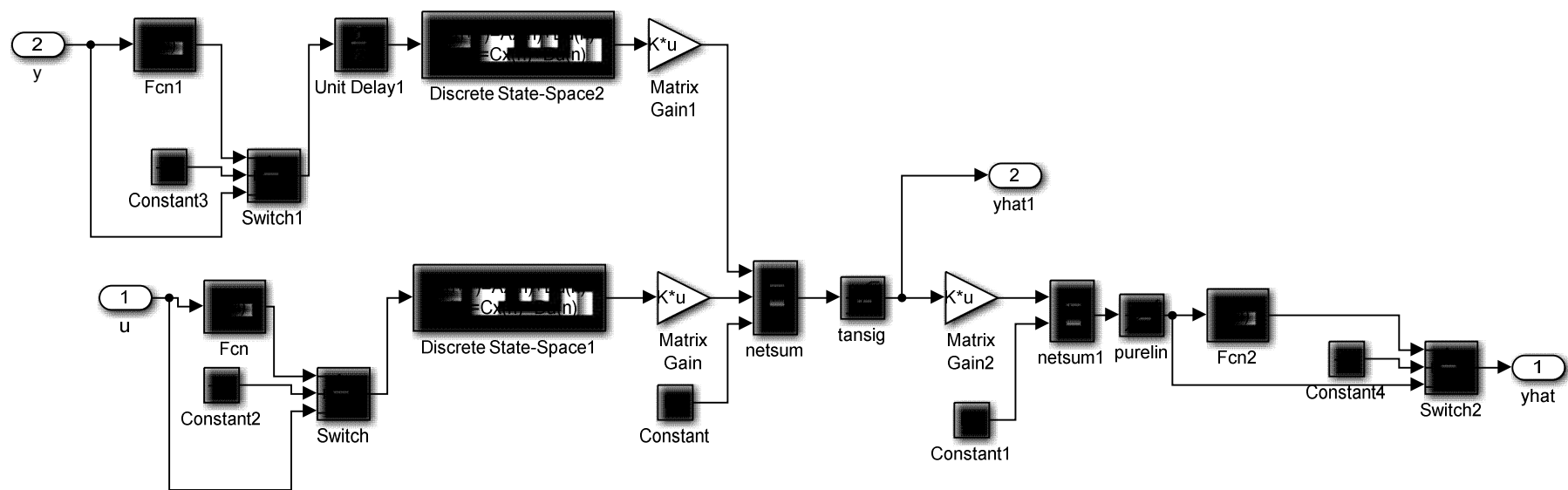


Рисунок 4.23 – Структурна схема нейронмережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

Перетворена структурна схема нейрорегулятора, отримана в результаті об'єднання схем рис.4.6 і рис.4.23 показана на рис. 4.27.

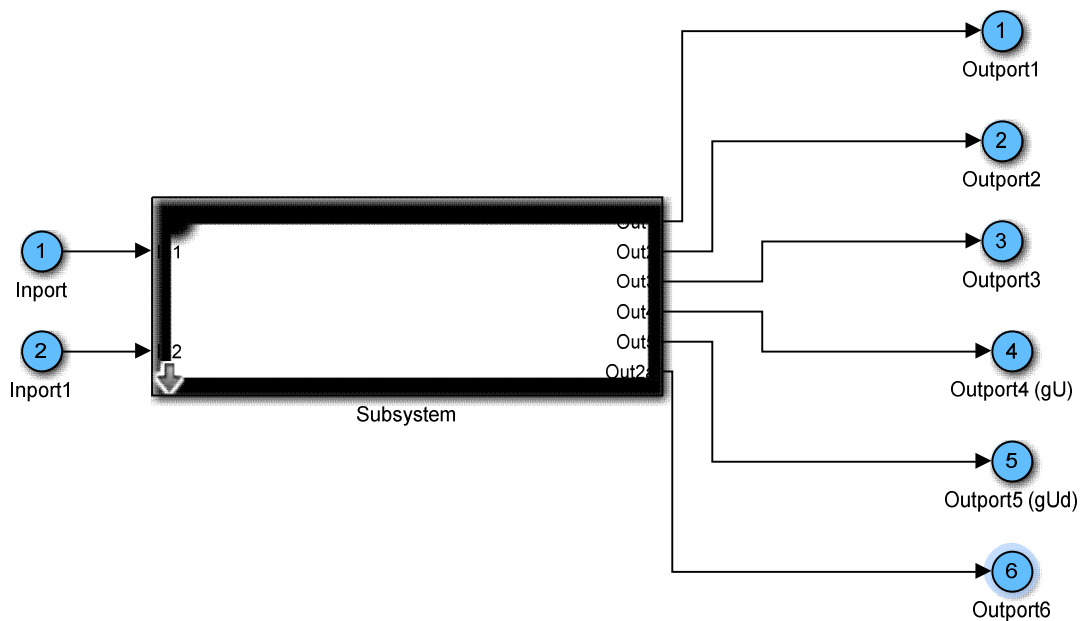


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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.

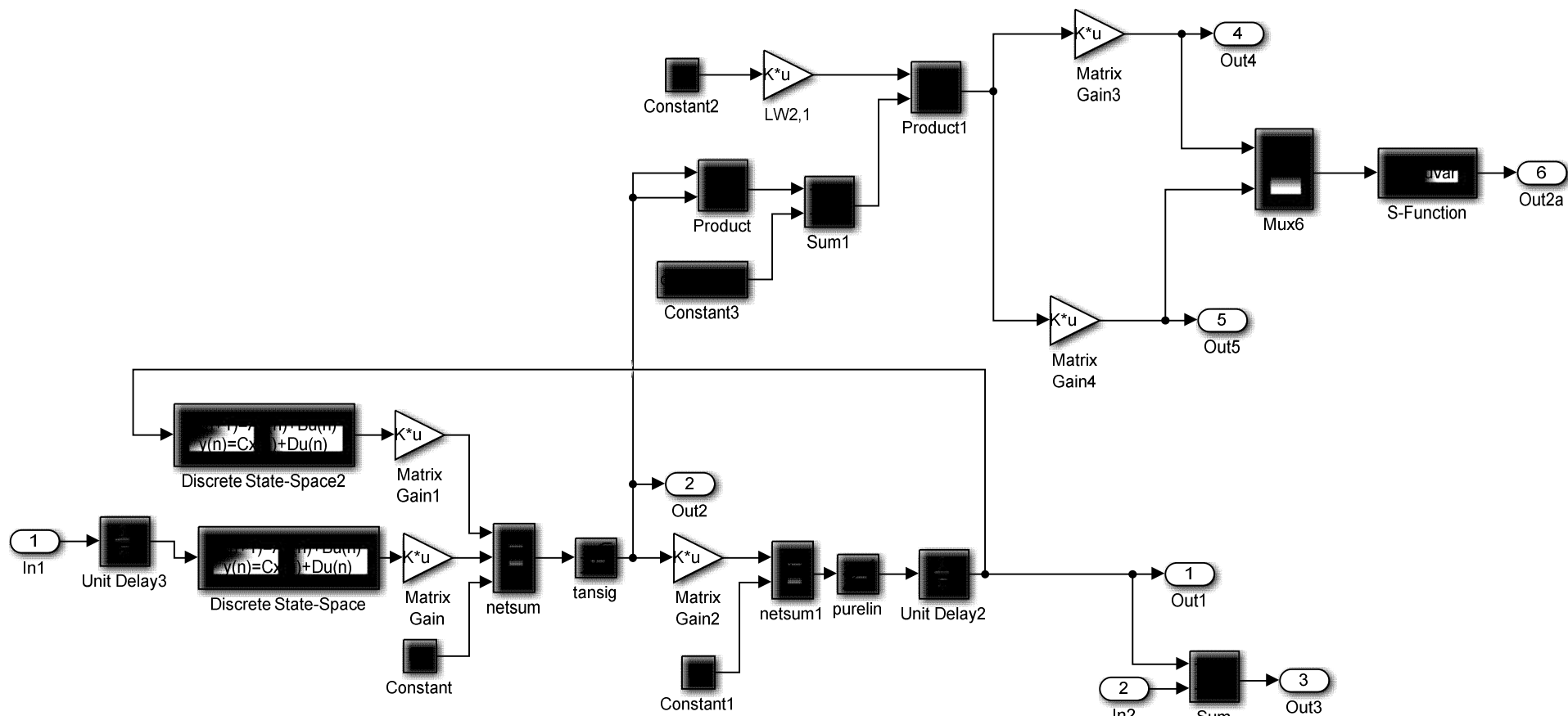


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

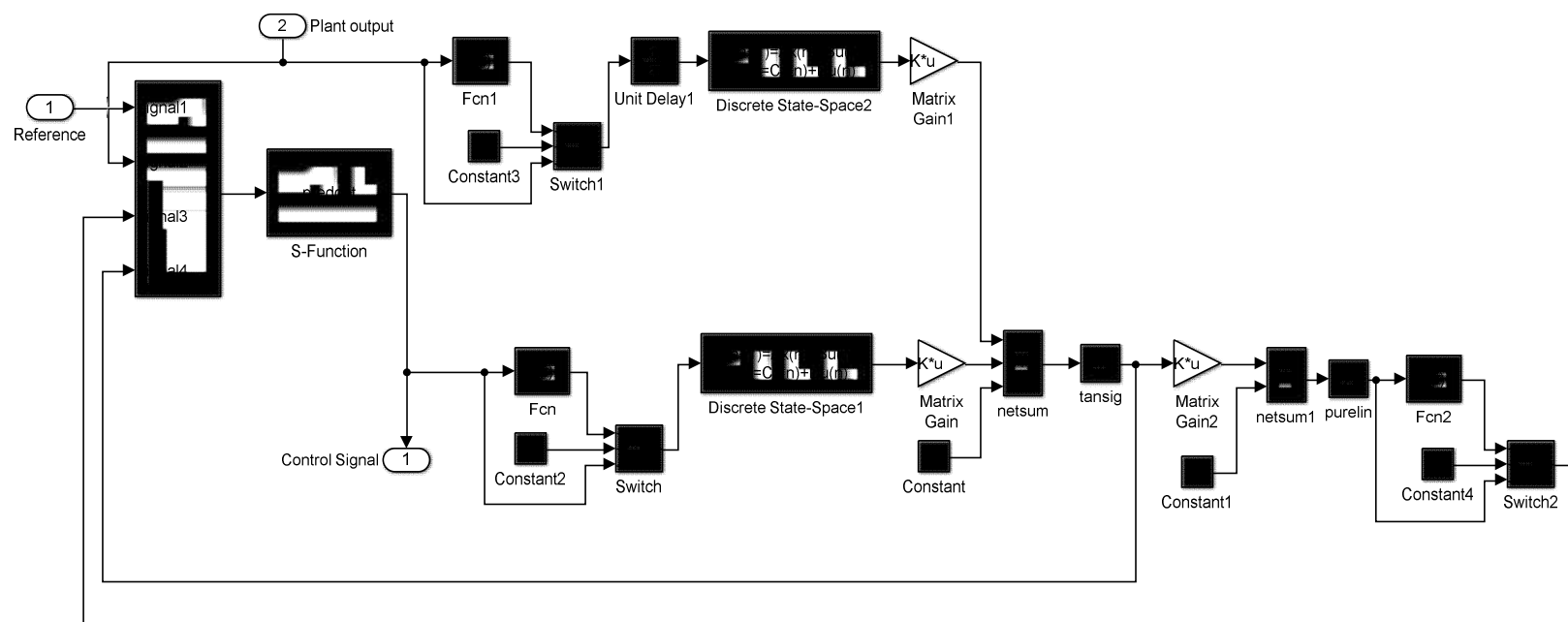


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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 кі-

льність ітерацій дозволяє отримати більш точну оптимальну стратегію керування, але водночас може збільшити обчислювальну складність та вимоги до ресурсів.

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

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

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

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

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

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

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

При ідентифікації об'єкта керування ключовим питанням є вибір кількості нейронів у прихованому шарі N . Недостатня кількість нейронів унеможливилює 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.

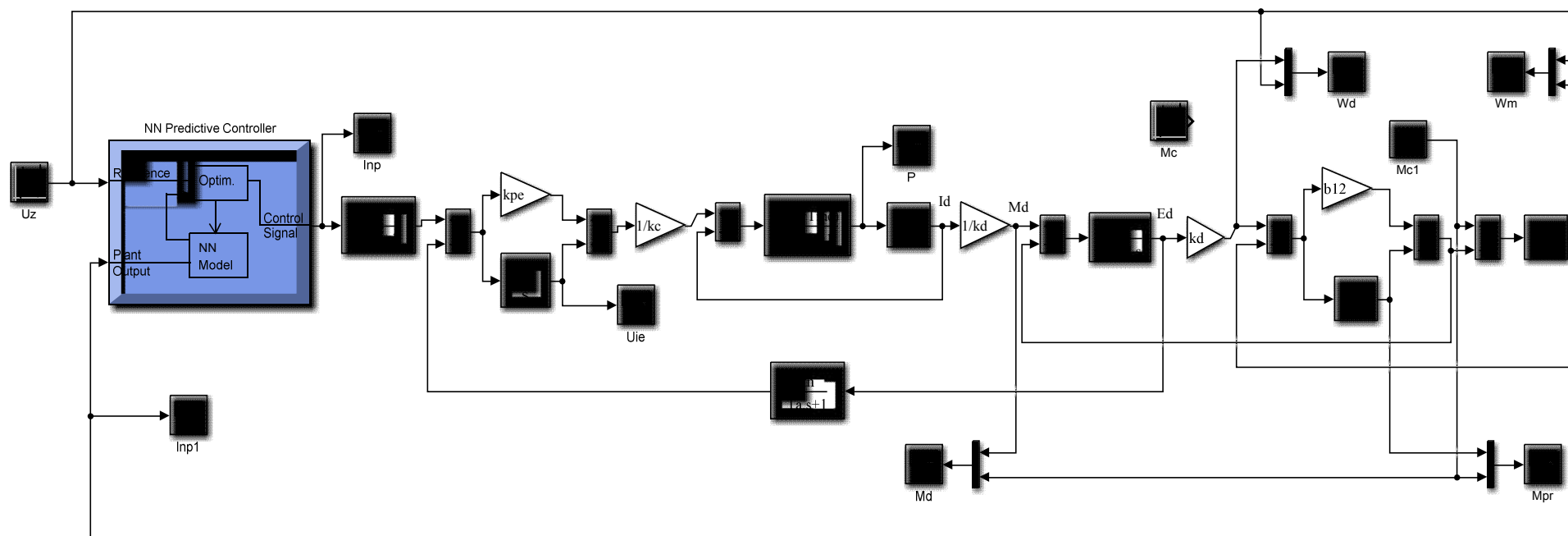


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

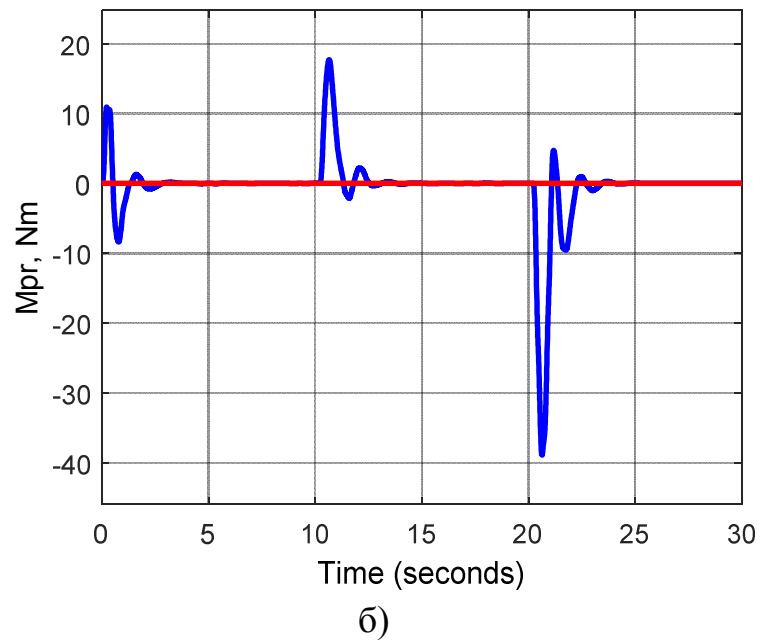
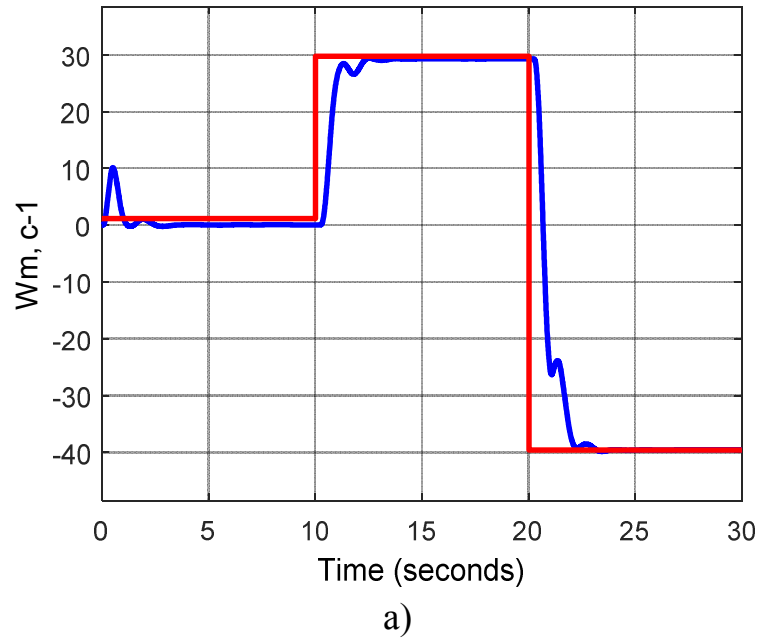


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

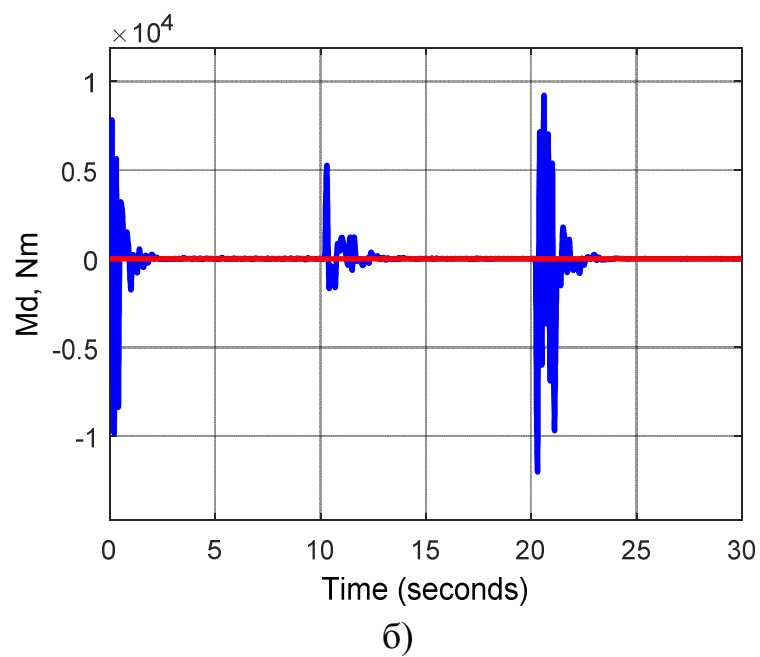
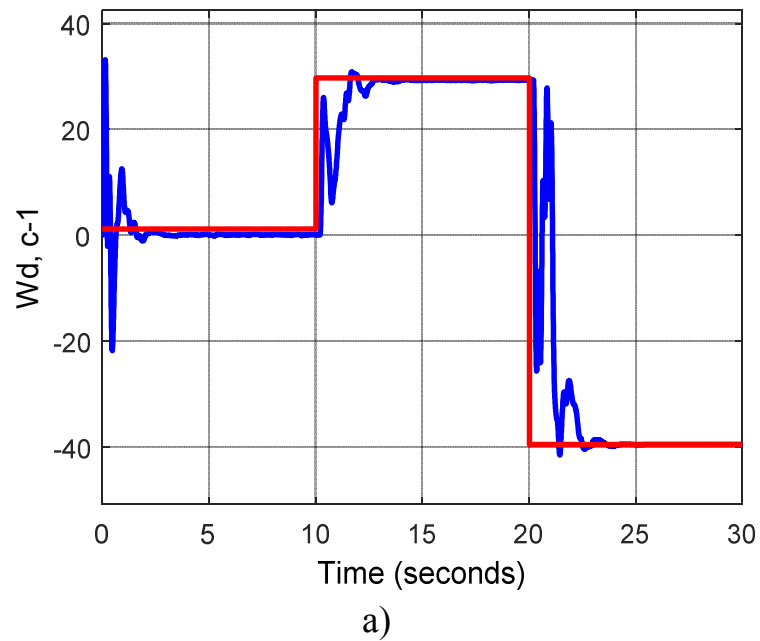


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

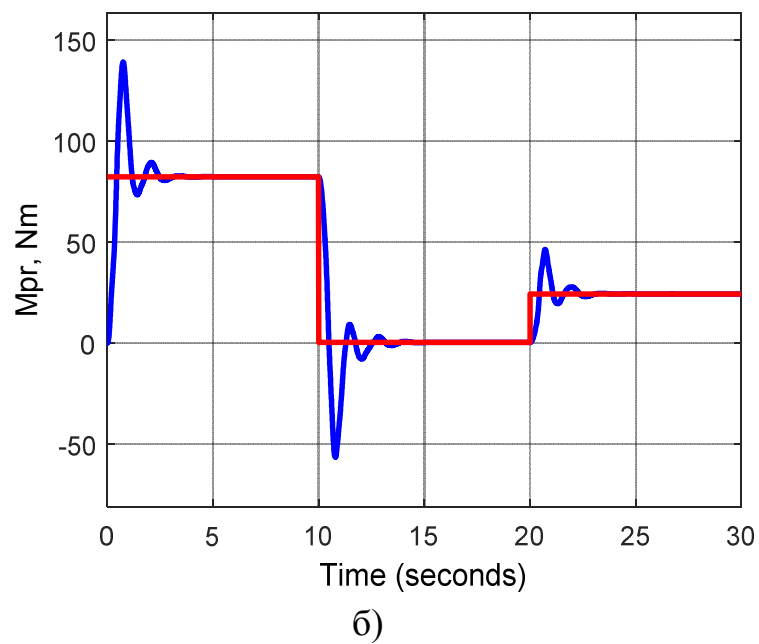
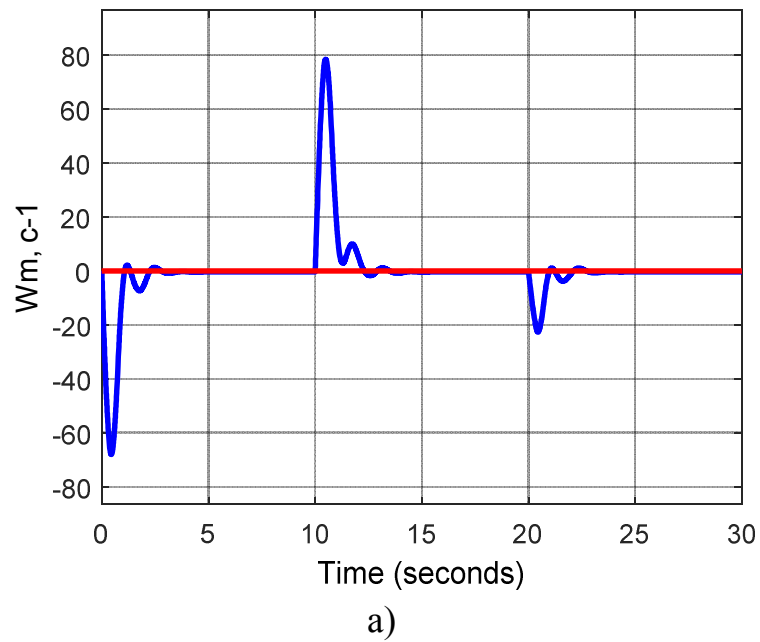


Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

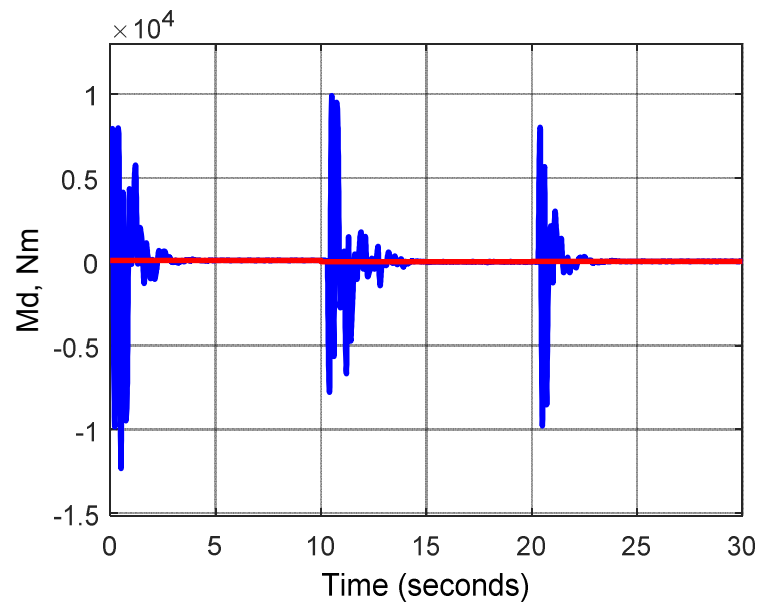
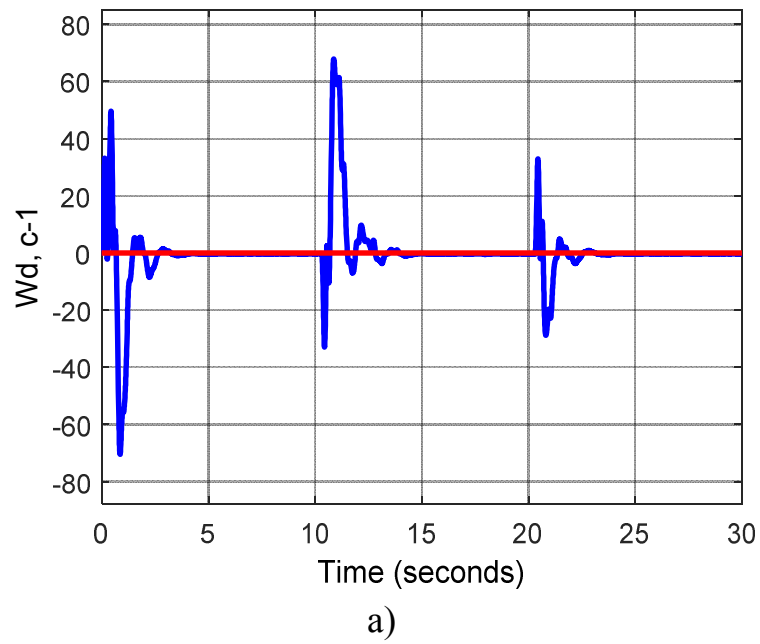


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і моменту двигуна $M_d = f(t)$ (б) двомасової системи з нейрорегулятором по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

1. Проаналізовано принципи побудови нейрорегулятора, реалізованих в пакеті прикладних програм (ППП) Neural Network Toolbox системи MATLAB: регулятор з прогнозом (NN Predictive Controller); регулятор на основі моделі авторегресії з ковзним середнім (NARMA-L2 Controller) і регулятор на основі еталонної моделі (Model Reference Controller). Для системи управління механізмом підйому мостового крана, що розглядається в даній роботі, обраний регулятор з прогнозом NN Predictive Controller. Розглянуто структуру та порядок синтезу нейрорегулятора.

2. У режимі SIMULINK системи MATLAB розроблена системи управління з нейрорегулятора NN Predictive Controller, яка включає блок керованого об'єкта, в якості якого використовується схема моделі двомасової системи управління механізмом підйому мостового крана і блок регулятора NN Predictive Controller.

3. Шляхом варіювання параметрів навчальної послідовності, з використанням моделі виконавчого пристрою, виконана генерація навчальної послідовності, яка дозволила виконати тренування мережі з високим ступенем точності. Оптимальні параметри навчальної послідовності наступні: довжина навчальної вибірки 10000; інтервал між двома послідовними моментами знімання даних 0,05с; мінімальний інтервал ідентифікації 4 с; максимальний інтервал ідентифікації 10 с.

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

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

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

6. Проведено моделювання нейромережевої системи з синтезованим нейрорегулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних стану системи в системі з нейрорегулятором NN Predictive Controller в режимі пуску і набросу навантаження мають задовільний характер. Система є астатичною. Перерегулювання не перевищує 2%, час регулювання складає 3 с, що відповідає завданню.

ВИСНОВКИ

В магістерській кваліфікаційній роботі розв'язана актуальна науково-практична задача розробки нейромережевої системи управління багатомасовою електромеханічною системою механізму підйому мостового крана, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління основних механізмів мостового крана, 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 підйому мостового крана.

2. На підставі аналізу технічних характеристик мостових кранів і вимог до електроприводів основних механізмів вибраний електропривод постійного струму, виконаний за системою ТП-Д. Виконано вибір електродвигуна і основного електроустаткування силового кола 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. Для поліпшення динамічних характеристик системи при роботі в режимі переривистих струмів в контур струму включений внутрішній контур напруги тиристорного перетворювача. Виконано розрахунок параметрів силового кола електроприводу, що використовуються при розрахунку системи автоматичного управління. Проведений синтез послідовних 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 виконаний синтез системи управління з нейрорегулятором NN Predictive Controller, реалізованим в пакеті прикладних програм Neural Network Toolbox системи MATLAB. Розглянутий принцип побудови і функціонування нейромережевого регулятора, приведений порядок синтезу регулятора, побудовані процеси в двомасовій системі з нейрорегулятором. Аналіз результатів моделювання показав, що реакція системи на ступінчасту дію з випадковою амплітудою цілком задовільна, має коливальний характер з достатньо швидким загасанням. Система є астатичною. Перерегулювання не перевищує 2%, час регулювання складає 3 с, що відповідає завданню.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

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

12. Терехов В.А., Ефимов Д.В., Тюкин И.Ю. Нейросетевые системы управления. – М.: ИПРЖР, 2002. – 480 с.
13. McCulloch W. S., Pitts W. A logical calculus of ideas imminent in nervous activity // Bulletin Mathematical Biophysics. – 1943. – № 5. – P.115-133 (Имеется перевод: Дж. Маккаллох, У. Питтс. Логическое исчисление идей, относящихся к нервной деятельности. В кн.: Автоматы. – М.: ИЛ, 1956).
14. Rosenblatt F. The perceptron: A probabilistic model for information storage and organization in the brain // Psychological Review. – 1958. – № 65. – P.386-407 (Имеется перевод: Розенблатт Ф. Принципы нейродинамики. – М.: Мир, 1966. – 480 с.).
15. Рябинин А. Д., Шквар А. М. Некоторые принципы функционального построения инвариантных бионических систем управления // Тр. 4-го Всесоюз. совещания. "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С 54-64.
16. Шквар А. М. Функциональные и структурные аспекты построения некоторых нейронных структур управления // Тр. 4-го Всесоюз. совещания "Теория инвариантности и теория чувствительности автоматических систем". Ч. II. 26-30 апреля 1971. – Киев, 1971. – С. 78-89.
17. Hecht-Nielsen R. Kolmogorov's Mapping Neural Network Existence Theorem / Proc. of IEEE First Ann. Int. Conf. on Neural Networks. – San Diego, 1987. – 3. – P. 11-13.
18. Hecht-Nielsen R. Theory of the Backpropagation Neural Network / Proc. of Int. Joint. Conf. on Neural Networks. – Washington, D.C., 1989. – 1. – P. 593-606.
19. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных суперпозициями функций меньшего числа переменных // Докл. АН СССР. – 1956. – 108. – № 2. – С. 179-182
20. Колмогоров А. Н. О представлении непрерывных функций нескольких переменных в виде суперпозиции непрерывных функций одного переменного и сложения // Докл. АН СССР, 1957. – 114. – С. 953-956..

21. Kolmogorov, A.N. On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition // American Math. Soc. Transl. – 1963. – №28. – P. 55-63.
22. Cybenko G. Approximation by Superposition of a Sigmoidal Function // Math. Contr. Sign. Syst. – 1989. – 2. – P. 303-314.
23. Hornik K., Stinchcombe M., White H. Multilayer Feedforward Networks are Universal Approximators // Neural Networks. – 1989. – 2. – P. 359-366
24. Hornik K. Approximation Capabilities of Multilayer Feedforward Networks / Neural Networks. – 1991. – 4. – P. 251-257.
25. Hopfield J.J., Tank D. W. Neural Computation of Decisions in Optimization Problems // Biol. Cybernetics. – 1985. – 52. – P. 141-152.
26. Narendra Narendra K. Gallman P.O. An iterative method for the identification of nonlinear system using a Hammerstein model // IEEE Trans. Aut. Control. - 1966. - Vol. 11. -N3. - P.546-550.
27. Psaltis, D., Sideris, A., Yamamura, A. A Multilayered neural network controller // IEEE Control Systems Magazine. - 1988. - Vol.8. - N4. -P. 17-21.
28. Joordan M.A. Generic constraints on underspecified target trajectories // Proc. Of Int. Joint Conf. On Neural Networks (IJCNN), Washington. - 1989.- Vol.1. -N3. - P.217-225.
29. Joordan M.A., Rumelhart D.E. Forward models: Supervised learning with a distal teacher // Cognitive Science.- 1990. -Vol.16. -N5.-P.313-355.
30. Saerens M, Soquet A. Neural controller based on backpropagation algorithm // IEEE Proceedings-F. - 1991. -Vol.138. -N.I. -P.55-62.
31. Miller III, W.T., Glanz, F.H., Kraft III, L.G. Application of a general learning algorithm to control of robotic manipulators // Int. J. of Robotics Research. - 1987. - Vol.6. - N2. - P.84-98.
32. Kawato, M., Furukawa, K, Suzuki, R. A hierarchical neural network model for control and learning of voluntary movement//Biological Cybernetics. - 1987. - Vol.57. - N2. -P. 169-185.

33. Pham D.T., Liu X. Neural Network for Identification, Prediction and Control. - London: Springer - Verlag, 1997. - 238p.
34. Milito R., Padilla C.S., Padilla R.A., Cadorin D. An innovations approach to dual control // IEEE Trans. Automat. Control. - 1982. -Vol.27. -Nl. -P.133-137.
35. Cichocki A., Unbehauen R. Neural networks for optimization and signal processing. – New York: John Wiley & Sons, 1998. – 52lp.
36. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.
37. Narendra K.S., Parthasarathy K. Identification and control of dynamical system using neural networks // IEEE Trans. Neural Networks.–1990. – Vol.1.–№1. – P.4-27.
38. Leontaritis I.J., Billings S.A. Input-output parametric model for nonlinear systems. Part 1: Deterministic non-linear systems. Part 2: Stochastic non-linear systems // Int. J. Control. – 1985. – Vol.41. – №2. –P. 303-344.
39. Bunke J. Kunstliche neuronale Netze zur Systemidentifikation aus gestorten Mefiwerten. Forsch. Ber. VDI, Reihe 8, № 667. – Dusseldorf: VDI Verlag. – 1997. – P. 37-51.
40. Яуре А.Г., Певзнер Е.М. Крановый электропривод: Справочник. – М.: Энергоатомиздат, 1988. – 314 с.
41. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
42. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
43. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
44. 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.
45. 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.

46. 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.
47. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
48. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
49. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
50. 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
51. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г.Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч1.
52. Попович М.Г., Лозинський О.Ю., Клепиков В.Б. та інші. Електромеханічні системи автоматичного керування та електроприводи. Навч. посіб. за напрямом «Електромеханіка» / М.Г. Попович, О.Ю. Лозинський, В.Б. Клепиков та інші. – К.: Либідь, 2005. – 680 с. Ч2.
53. 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.
- 54.Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: На-

- вчальний посібник / О.Г. Гурко, І.Ф. Єрьоменко. Харків, ХНАДУ, 2012. – 284 с.
55. 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.
56. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
57. 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.
58. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. – СПб: Питер, 2001. –488 с.

ДОДАТОК А

БАГАТОШАРОВІ ПРЯМОНАПРАВЛЕНІ МЕРЕЖІ

А.1 Структура штучного нейрона

Основним елементом всіх мереж є нейрон. Штучний нейрон відображає структуру біологічного нейрона (див. рис.А.1). Він має декілька входів і один вихід, який може бути пов'язаний з багатьма входами інших нейронів. Кожен

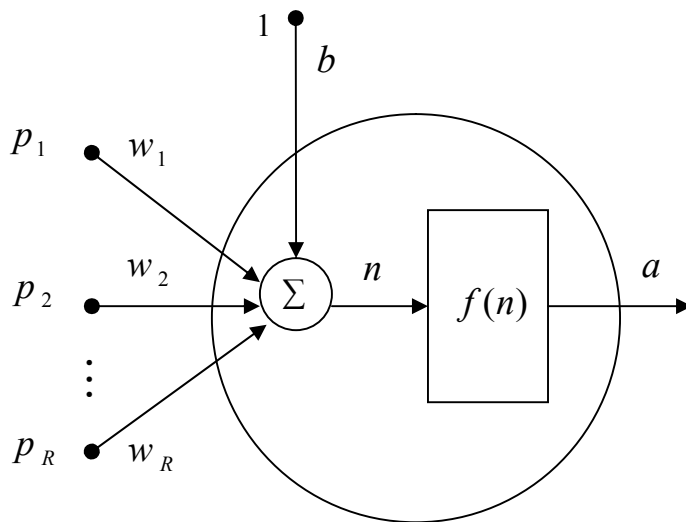


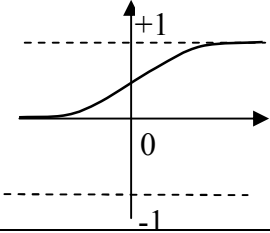
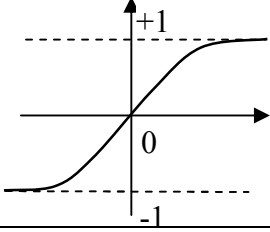
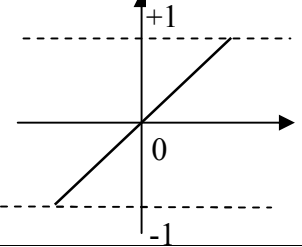
Рисунок А.1 - Схема штучного нейрона

подібний зв'язок має свій коефіцієнт посилення. Нейрон вчиться шляхом модифікації коефіцієнтів посилення відповідно до закону навчання.

Нейрон реалізує скалярну функцію векторного аргументу. Сигнали обробляються в нейроні таким чином. Кожен вхідний сигнал p_i , множиться на відповідний ваговий коефіцієнт w_i і складається з ін-

шими зваженими вхідними сигналами. Порогова величина, яка позначена як b розглядається в більшості випадків як додатковий ваговий коефіцієнт w_0 . Потім отримана сума n перетвориться за допомогою функції активації або передавальної функції f . Як передавальна функція в нейроні може використовуватися принципово будь-яка функція. Вид активаційної функції надає істотний вплив на показники якості регулювання електромеханічної системи при нейроконтрольному управлінні. Для навчання методом зворотного розповсюдження помилки необхідна передавальна функція, що диференціюється. У таблиці А.1 представлені найчастішим використовувані в персептроні передавальні функції.

Таблиця А.1 – Передавальні функції персептрона

Передавальна функція	Графічне зображення	Функція	Похідна
сигмоїдальна		$f(n) = \frac{1}{1 + e^{-n}}$	$f'(n) = f(n)(1 - f(n))$
тангенс-гіперболічна		$f(n) = \frac{1 - e^{-2n}}{1 + e^{-2n}}$	$f'(n) = 1 - f^2(n)$
лінійна		$f(n) = n$	$f'(n) = 1$

Рівняння нейрона має вид

$$a = f\left(\sum_{i=1}^R w_i \cdot p_i + b\right).$$

А.2 Одношарові нейронні мережі

Внутрішній потенціал нейронних технологій виявляється при об'єднанні окремих нейронів в нейромережеві структури. З огляду на те, що в даний час існує величезна кількість різних типів мереж, неможливо розглянути кожний з них детально. Тому далі виклад обмежується тільки фактично використовуваними в цій роботі типами мереж. Сукупність нейронів з однаковими вхідними сигналами називають шаром. Одношарова прямонаправлена мережа зв'язується зазвичай з поняттям одношаровий персептрон. Якщо нейрон має одну з сигмоїдних функцій, то значення його вихідної величини будуть обмежені в деякому діапазоні. Вихід лінійного нейрона може приймати будь-які значення.

В результаті об'єднання декілька нейронних шарів виникає типова багатошарова мережа – багатошаровий персептрон (БП). БП тренується переважно алгоритмом зворотного розповсюдження помилки. Одношарові мережі описуються однозначно вказівкою числа входів R і числа нейронів S (число виходів S), і, типом передавальної функції. Розгорнута схема мережі показана на рис.А.2.

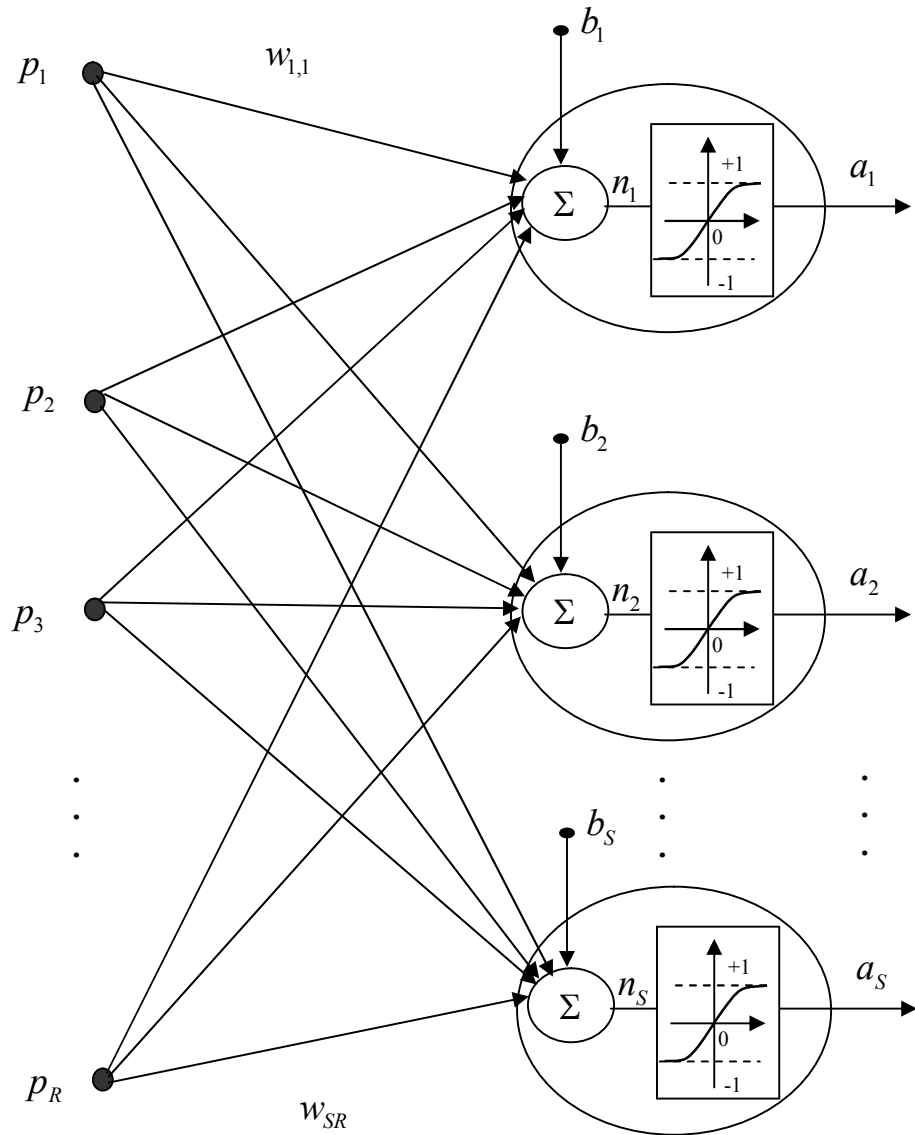


Рисунок А.2 - Структурна схема одношарової мережі
(одношаровий персептрон)

В цій мережі кожен елемент вектора входу сполучений зі всіма входами нейрона і це з'єднання задається матрицею вагів $\mathbf{W}$ при цьому кожен i -й ней-

рон включає елемент, що підсумовує, який формує скалярний вихід $n(i)$. Сукупність скалярних функцій $n(i)$ об'єднується в S -елементний вектор входу $\mathbf{n}$ функції активації шару. Виходи шару нейронів формують вектор-стовпець $\mathbf{a}$, і, таким чином, опис шару нейронів має вигляд:

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

У кожному шарі, як правило, використовується одна і та ж функція активації. Проте можна створювати складені шари нейронів з використанням різних функцій активації, сполучаючи мережі, подібні зображеною на рис.. А.2, паралельно. Обидві мережі матимуть ті ж самі входи, і кожна мережа генеруватиме певну частину виходів. Елементи вектора входу передаються в мережу через матрицю вагів $\mathbf{W}$, що має вигляд:

$$\mathbf{W} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1R} \\ w_{21} & w_{22} & \cdots & w_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ w_{S1} & w_{S2} & \cdots & w_{SR} \end{bmatrix}.$$

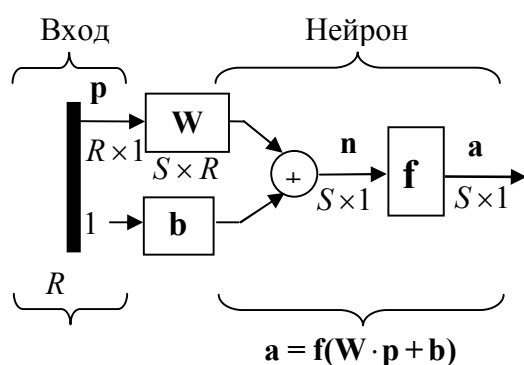


Рисунок А.3 – Укрупнена структурна схема одношарової мережі

Індекси рядків матриці $\mathbf{W}$ указують адресатів (пункти призначення) вагів нейронів, а індекси стовпців – яке джерело є входом для цієї ваги. Таким чином, елемент матриці вагів $w_{1,2} = \mathbf{W}(1,2)$ визначає коефіцієнт, на який множиться другий елемент входу при передачі його на перший нейрон. Укрупнена структурна схема одношарової мережі показана на рис. А.3.

А.3 Багат шарові нейронні мережі

Для багат шарового персептрона повинні указуватися кількість і тип нейронів всіх прихованих шарів і вихідного шаруючи. Вхідний шар, як правило, не вважається шаром, оскільки сигнали, що поступають в нього, не змінюються. На рис.А.4 представлений БП з R входами, S^1 і S^2 нейронами в прихованих шарах і S^3 нейронами у вихідному шарі. Ця ж тришарова мережа може бути представлена у вигляді укрупненої структурної схеми (рис.А.5).

Перевага цих мереж в тому, що принцип їх роботи описується простими матричними операціями. Для матриць вагів входу і виходу шару використовуються позначення **IW** (Input Weight) і **LW** (Layer Weight) відповідно (такі позначення прийняті в системі MATLAB, яка використовувалася для досліджень в даній роботі). Вихід останнього шару $\mathbf{a}^3$ позначений через $\mathbf{y}$. Це зроблено для того, щоб підкреслити, що вихід останнього шару є виходом мережі.

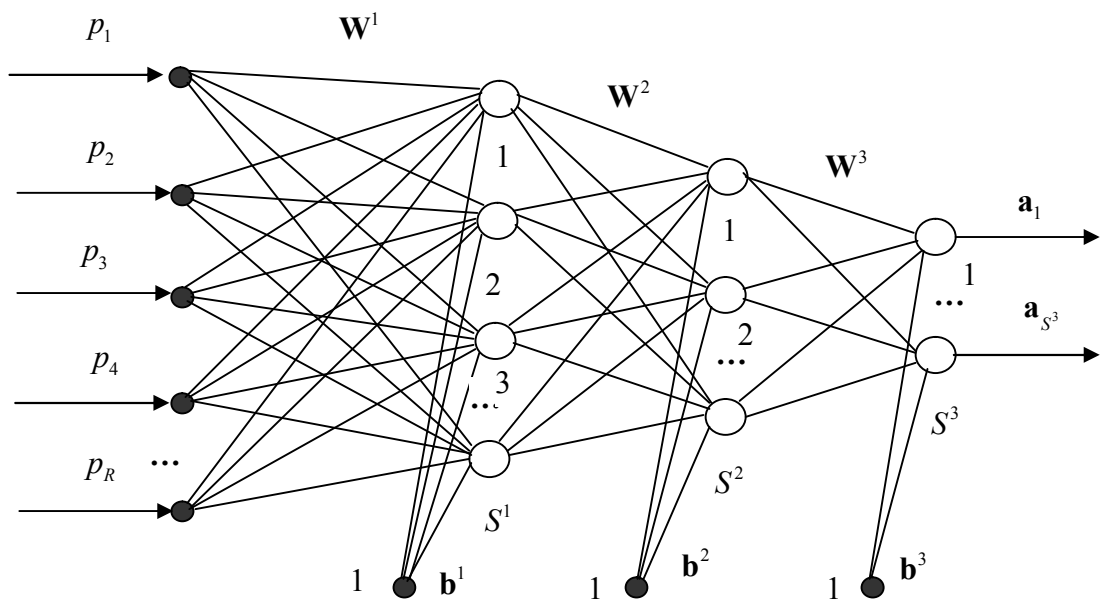


Рисунок А.4 – Структурна схема тришарової мережі
(тришаровий персептрон)

Як правило, для вихідного шару вибирається лінійна передавальна функція, щоб уникнути обмежень вихідної величини в деякому діапазоні. Застосу-

вання декількох наступних один за одним лінійних шарів було б даремним, оскільки їх можна об'єднати без втрати інформації в один шар. Про оптимальний вибір передавальної функції прихованих шарів не існує до цих пір ніяких тверджень. З прагматичних міркувань на практиці призначають в межах шару однакові передавальні функції і найчастіше це сигмоїдні функції. З погляду здібностей апроксимацій не має ніякого значення, яка з двох сигмоїдних функцій вибирається, оскільки одну можна перераховувати в іншу. МП є універсальним апроксиматором-класифікатором і належить в сьогоденній практиці до найбільш використовуваних нейронних мереж.

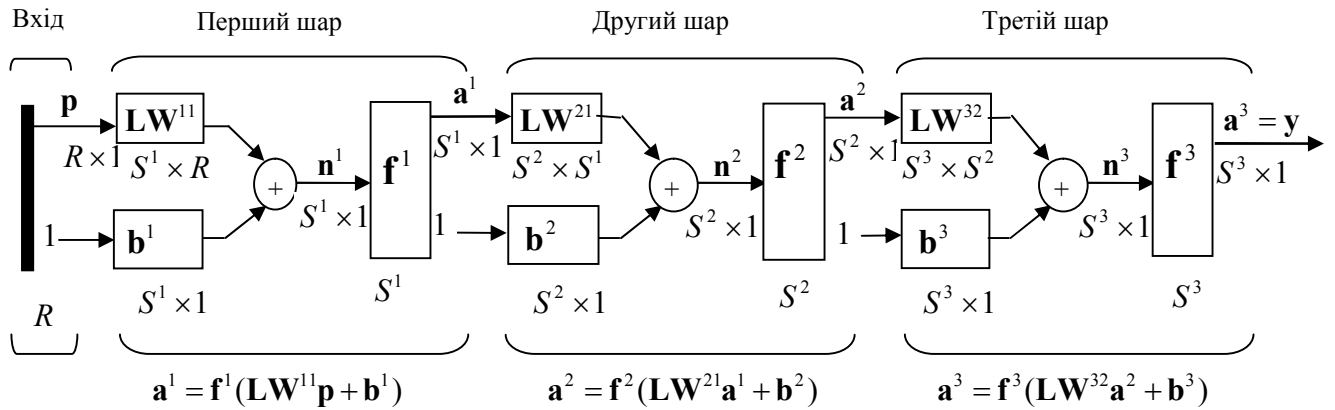


Рисунок А.5 – Укрупнена структурна схема тришарової мережі

А.4 Здібності апроксимації і узагальнення багатoshарових мереж

Принципово в більшій частині практичного застосування нейронних мереж йдеться, про те, щоб реалізувати відображення функції за допомогою кінцевого числа параметрів мережі. При цьому повинні виконуватися дві наступні вимоги: апроксимувати представлений для навчання набір даних; узагальнити отримані знання на невідомі вхідні дані.

Виконання першої вимоги означає знайти мережу, яка може з будь-якою точністю вивчити будь-яку задану функцію. Проблема відображення багатовимірної нелінійної функції за допомогою лінійної суперпозиції простих нелінійних функцій сходиться до так званої 13-ої проблеми Гілберта. Гілберт припускав,

що ця проблема не має рішення. Це припущення було спростоване теоремою Колмогорова-Арнольдса в кінці 50-х років. В кінці 80-х років ця теорема була переведена на термінологію нейронних мереж Р. Хехт-Нильсоном і Цибенко.

Публікації останніх років зачіпали питання здібностей апроксимацій прямонаправлених нейронних мереж. Для докладного розгляду доказів і деталей слід звернутися до спеціальної літератури. Далі представляються тільки підсумки математичних викладень:

- Теоретично завжди існує така прямонаправлена нейронна мережа з одним нелінійним прихованим шаром і лінійним вихідним шаром, яка здатна відображати будь-які шматково-безперервні багатовимірні функції.
- Необхідна якість апроксимації завжди досягається при кінцевому числі нейронів, якщо задана певна точність апроксимації.

Тільки для небагатьох нейронних структур в сьогодення існує строге математичне обґрунтування їх здібностей. БП в даний момент краще всього досліджений теоретично:

- Збіжність процесу навчання доведена теоремою Perceptron-Konvergenz, яка стверджує, що функція може бути апроксимована за кінцеве, але не відоме, число кроків.
- БП з одним прихованим шаром формує гіперплощині. При двох прихованих шарах утворюються замкнуті гіперпростори, які досить, щоб вирішувати які завгодно по складності завдання класифікації. Подальші шари нічого не покращують, а збільшують час навчання.

Оцінка числа нейронів в прихованих шарах мережі – все ще нетривіальне завдання. Не підлягає сумніву факт вирішального впливу на розмір мережі ступеня нелінійності об'єкту, що ідентифікується. Існують численні емпіричні формули, що дають мінімальне і максимальне число нейронів або зв'язків для забезпечення необхідної точності апроксимації. Але на сьогоднішній день вони дуже неточні. Запропоновані також методи автоматичного підбору числа нейронів в процесі тренування, які можна розділити на дві групи: група редукуючих методів і група конструюючих методів. Редукуючі алгоритми менш ефек-

тивні, оскільки вони мають, принаймні, два недоліки: надмірне число нейронів при ініціалізації мережі невідоме і тому оцінюється приблизно, із-за явно надмірного числа нейронів процес тренування протікає досить поволі. Сьогоднішня практика свідчить, що систематичний підбір оптимального числа нейронів також є ефективним. З вищесказаного виходить, що для успішної ідентифікації деякої нелінійної системи мають значення вибір типу мережі, виду функції активації, числа нейронів (а для БП ще і числа прихованих шарів).

Здібності нейронних мереж до узагальнення дозволяють проводити тренування при одних парах тренувальних даних і після цього отримувати правильний вихід при інших можливих вхідних даних. Виникаюча проблема формування інформативних тренувальних даних не вирішена в загальному вигляді до цих пір. Проте, успішно практикується наступний емпіричний метод. Узагальнювані вхідні дані не повинні сильно відрізнятися від тренувальних даних. Це вимушує обмежуватися при навчанні небагатьма крапками даних, між якими мережею виконуватиметься нелінійна інтерполяція.

ДОДАТОК Б

НАВЧАННЯ НЕЙРОННИХ МЕРЕЖ

Б.1 Завдання навчання

При рішенні за допомогою нейронних мереж прикладних завдань необхідно зібрати точний і представницький об'єм даних для того, щоб навчити нейронну мережу рішенням таких задач. Повчальний набір даних – це набір спостережень, що містять ознаки об'єкту, що вивчається. Перше питання, які ознаки використовувати і скільки і які спостереження треба провести.

Вибір ознак, принаймні первинний, здійснюється евристично на основі наявного досвіду, який може підказати, які ознаки є найбільш важливими. Спочатку слід включити всі ознаки, які, на думку аналітиків або експертів, є істотними, на подальших етапах ця множина буде скорочена.

Нейронні мережі працюють з числовими даними, узятими, як правило, з деякого обмеженого діапазону. Це може створити проблеми, якщо значення спостережень виходять за межі цього діапазону або пропущені.

Питання про те, скільки потрібно мати спостережень для навчання мережі, часто виявляється непростим. Відомий ряд евристичних правил, які встановлюють зв'язок між кількістю необхідних спостережень і розмірами мережі. Просте з них свідчить, що кількість спостережень повинна бути в 10 разів більше числа зв'язків в мережі. Насправді це число залежить від складності того відображення, яке повинна відтворювати нейронна мережа. Із зростанням числа використовуваних ознак кількість спостережень зростає по нелінійному закону, так що вже при досить невеликому числі ознак, скажемо 50, може потрібно величезне число спостережень. Ця проблема носить назву «прокляття розмірності».

Для більшості реальних завдань буває достатнім декількох сотень або тисяч спостережень. Для складних завдань може потрібно більша кількість, проте дуже рідко зустрічаються завдання, де потрібно менше 100 спостережень. Якщо

даних мало, то мережа не має достатньої інформації для навчання, і краще, що можна в цьому випадку зробити, – це спробувати підігнати до даним деяку лінійну модель.

Б.2 Процедури адаптації і навчання

Після того, як визначена кількість шарів мережі і число нейронів в кожному з них, потрібно призначити значення вагів і зсувів, які мінімізують помилку рішення. Це досягається за допомогою процедур навчання. Шляхом аналізу тих, що є у розпорядженні аналітика вхідних і вихідних даних ваги і зсуву мережі автоматично настроюються так, щоб мінімізувати різницю між бажаним сигналом і отриманим на виході в результаті моделювання. Ця різниця носить назву помилки навчання. Таким чином, процес навчання – це процес підгонки параметрів тієї моделі процесу або явища, яка реалізується нейронною мережею. Помилка навчання для конкретної конфігурації нейронної мережі визначається шляхом прогону через мережу всіх наявних спостережень і порівняння вихідних значень з бажаними, цільовими значеннями. Ці різниці дозволяють сформувати так звану функцію помилок (критерій якості навчання). Як така функція найчастіше береться сума квадратів помилок. При моделюванні нейронних мереж з лінійними функціями активації нейронів можна побудувати алгоритм, що гарантує досягнення абсолютного мінімуму помилки навчання. Для нейронних мереж з нелінійними функціями активації в загальному випадку не можна гарантувати досягнення глобального мінімуму функції помилки.

При такому підході до процедури навчання може виявитися корисним геометричний аналіз поверхні функції помилок. Визначимо ваги і зсуву як вільні параметри моделі і їх загальне число позначимо через N ; кожному набору таких параметрів поставимо у відповідність одне вимірювання у вигляді помилки мережі. Тоді для всіляких поєднань вагів і зсувів відповідну помилку мережі можна зобразити точкою в $N+1$ -вимірному просторі, а всі такі точки утворюють деяку поверхню, звану поверхнею функції помилок. При такому підході мета

навчання нейронної мережі полягає в тому, щоб знайти на цій багатовимірній поверхні глобальний мінімум.

У разі лінійної моделі мережі і функції помилок у вигляді суми квадратів така поверхня буде параболоїдом, який має єдиний мінімум і це дозволяє відшукати такий мінімум досить просто.

У разі нелінійної моделі поверхня помилок має набагато складнішу будову і має ряд несприятливих властивостей, зокрема може мати локальні мінімуми, плоскі ділянки, сідлові точки і довгі вузькі яри.

Визначити глобальний мінімум багатовимірної функції аналітично неможливо, і тому навчання нейронної мережі, по суті справи, є процедурою вивчення поверхні функції помилок. Відштовхуючись від випадково вибраної точки на поверхні функції помилок, алгоритм навчання поступово відшукує глобальний мінімум. Як правило, для цього обчислюється градієнт (нахил) функції помилок в даній точці, а потім ця інформація використовується для просування вниз по схилу. Врешті-решт алгоритм зупиняється в деякому мінімумі, який може опинитися лише локальним мінімумом, а якщо повезе, то і глобальним.

Таким чином, по суті алгоритми навчання нейронних мереж аналогічні алгоритмам пошуку глобального екстремуму функції багатьох змінних. Серед останніх слід виділити алгоритми зв'язаних градієнтів і Левенберга-Марквардта (Levenberg-Marquardt).

Проте з урахуванням специфіки нейронних мереж для них розроблені спеціальні алгоритми навчання, серед яких слід виділити алгоритм зворотного розповсюдження помилки.

При використанні алгоритму зворотного розповсюдження помилки мережа розраховує помилку, що виникає у вихідному шарі, і обчислює вектор градієнта як функцію вагів і зсувів. Цей вектор указує напрям найкоротшого спуску по поверхні для даної крапки, тому якщо просунутися в цьому напрямі, то помилка зменшиться. Послідовність таких кроків врешті-решт приведе до мінімуму того або іншого типу. Певну трудність тут викликає вибір величини кроку.

При великій довжині кроку збіжність буде швидшою, але є небезпека перестрибнути через рішення або піти в неправильному напрямі. Класичним прикладом такого явища при навчанні нейронної мережі є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного схилу на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. На практиці величина кроку вибирається пропорційній крутизні схилу (градієнту функції помилок); такий коефіцієнт пропорційності називається параметром швидкості настройки. Правильний вибір параметра швидкості настройки залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; цей параметр може також залежати від часу, зменшуючись у міру виконання алгоритму.

Алгоритм діє ітеративно і його кроки прийнято називати епохами або циклами. На кожному циклі на вхід мережі послідовно подаються всі повчальні спостереження, вихідні значення порівнюються з цільовими значеннями і обчислюється функція помилки. Значення функції помилки, а також її градієнта використовуються для коректування вагів і зсувів, після чого всі дії повторюються. Початкові значення вагів і зсувів мережі вибираються випадковим чином, і процес навчання припиняється або коли реалізована певна кількість циклів, або коли помилка досягне деякого малого значення або перестане зменшуватися.

Б.3 Явище перенавчання

Одна з найбільш серйозних труднощів при навчанні мережі полягає в тому, що у ряді випадків ми мінімізуємо не ту помилку, яку насправді потрібно мінімізувати; потрібно мінімізувати помилку, яка з'являється в мережі, коли на неї подаються абсолютно нові спостереження. Вельми важливо, щоб нейронна мережа володіла здатністю пристосовуватися до цих нових спостережень. Що ж відбувається насправді? Мережа навчається мінімізувати помилку на деякій обмеженій повчальній множині. Це не відповідає вимогам теорії про наявність ідеальної і нескінченно великої повчальної множини. І це не відповідає тій реа-

льній ситуації, коли треба мінімізувати конкретну функцію помилок для наперед невідомої моделі.

Це породжує проблему, яка відома як явище перенавчання. Звернемося до завдання апроксимації деякої функції многочленом. Графіки многочленів часто мають вельми хитромудрі форми, і чим вище ступінь многочлена, тим складніше їх форма. Якщо є деякий набір даних, то можна поставити мету підібрати для нього апроксимуючий многочлен і таким чином отримати відповідну математичну модель для цього набору даних. Оскільки початкові дані, як правило, задані з погрішностями, то не можна вважати, що краща модель задається кривою, яка проходить точно через задані точки. Многочлен низького порядку може виявитися достатньо грубим для апроксимації даних, тоді як многочлен високого порядку може точно слідувати даним, приймаючи при цьому вельми хитромудру форму, що не має ніякого відношення до форми дійсної залежності. Остання ситуація і демонструє те, що називається явищем перенавчання.

При роботі з нейронними мережами користувач стикається з тією ж проблемою. Мережі з великою кількістю вагів дозволяють відтворювати дуже складні функції, і в цьому сенсі вони схильні до перенавчання. Мережа ж з невеликою кількістю вагів може опинитися недостатньо гнучкою, щоб змоделювати наявну залежність. Наприклад, одношарова лінійна мережа здатна відтворювати тільки лінійні функції. Якщо використовувати багат шарові лінійні мережі, то помилка завжди буде менша, але може свідчити не про хорошу якість моделі, а про те, що виявляється явище перенавчання.

Для того, щоб виявити ефект перенавчання, використовується механізм контрольної перевірки. Частина повчальних спостережень резервується як контрольні спостереження і не використовується при навчанні мережі. Натомість у міру роботи алгоритму ці спостереження застосовуються для незалежного контролю результату. Спочатку помилка мережі на повчальне і контрольному множині буде однаковою; якщо вони істотно відрізняються, то, мабуть, це означає, що розбиття спостережень на 2 множини не забезпечило їх однорідність. У міру навчання мережі помилка убиває, і, поки навчання зменшує функ-

цію помилок помилка на контрольній множині також убуватиме. Якщо ж контрольна помилка перестала убувати або стала рости, це указує на те, що мережа почала дуже близько слідувати початковим даним і навчання слід зупинити. В цьому випадку слід зменшити кількість нейронів або шарів, бо мережа є дуже могутнього для вирішення даного завдання. Якщо ж, навпаки, мережа має недостатню потужність, щоб відтворити наявну залежність, те явище перенавчання швидше за все спостерігатися не буде і обидві помилки – навчання і перевірки – не досягнуть достатньо малого рівня.

Що виникають при роботі з нейронними мережами проблеми відшукування глобального мінімуму або вибору розміру мережі призводять до того, що при практичній роботі доводиться експериментувати з великим числом мереж різних конфігурацій, деколи навчаючи кожен з них кілька разів і порівнюючи отримані результати. Головним критерієм вибору в цих випадках є контрольна погрішність. При цьому застосовується правило, згідно якому з двох нейронних мереж з приблизно рівними контрольними погрішностями слід вибирати ту, яка простіше.

Необхідність багатократних експериментів веде до того, що контрольна множина починає грати ключову роль у виборі моделі нейронної мережі, тобто стає частиною процесу навчання. Тим самим його роль як незалежного критерію якості моделі ослабляється, оскільки при великому числі експериментів виникає ризик перенавчання нейронної мережі на контрольній множині. Для того, щоб гарантувати надійність вибраної моделі мережі, резервують ще одне – тестова безліч спостережень. Підсумкова модель тестується на даних з цієї множини, щоб переконатися, що результати, досягнуті на повчальній і контрольній множинах реальні. Зрозуміло, для того, щоб добре грати свою роль, тестова множина повинна бути використана тільки 1 раз: якщо його використовувати повторно для коректування процесу навчання, то воно фактично перетвориться на контрольну множину.

Отже, процедура побудови нейронної мережі складається з наступних кроків:

вибору початкової конфігурації мережі; наприклад, у вигляді одного шару з числом нейронів, рівним $1/2$ загальної кількості входів і виходів;
моделювання і навчання мережі з оцінкою контрольної помилки і використанням додаткових нейронів або проміжних шарів;
виявлення ефекту перенавчання і коректування конфігурації мережі.

Б.4 Властивість узагальнення

При описі процедури навчання нейронних мереж неявно використовувалося припущення, що повчальна, контрольна і тестова множини є представницькими для вирішуваної задачі. Зазвичай як повчальні беруться дані, випробувань на ряду прикладів. Якщо обставини змінилися, то закономірності, що мали місце у минулому, можуть більше не діяти.

Крім того, нейронна мережа може навчатися тільки на тих даних, які вона має в своєму розпорядженні. Припустимо, що відоме повчальна множина для системи стабілізації літака при польоті в спокійній атмосфері, а потрібно спроектувати систему стабілізації на основі нейронної мережі для умов польоту при сильних збуреннях. Тоді навряд чи можна чекати від мережі правильного рішення в абсолютно новій для неї ситуації.

Класичним прикладом непредставницької моделі нейронної мережі є наступна ситуація. При проектуванні системи машинного зору, призначеної для автоматичного розпізнавання цілей, мережа навчалася на 100 картинках, що містять зображення танків, і на 100 інших картинках, де танків не було. Після навчання мережі був досягнутий стовідсотково «правильний» результат. Але коли на вхід мережі були подані нові дані, вона безнадійно провалилася. У чому ж була причина? З'ясувалося, що фотографії з танками були зроблені в похмурий, дощовий день, а фотографії без танків – в сонячний день. Мережа навчилася уловлювати різницю в загальній освітленості. Щоб мережа могла результативно працювати, її слід було навчати на даних, де б були присутні всі погодні

умови і типи освітлення, при яких мережу передбачається використовувати, і це не кажучи ще про рельєф місцевості, вугілля і дистанцію зйомки і т.д.

Якщо мережа мінімізує загальну погрішність, великого значення набувають пропорції, в яких представлені дані різних типів. Мережа, навчена на 900 «хороших» і 100 «поганих» спостереженнях, спотворюватиме результат на користь хороших спостережень, оскільки це дозволить алгоритму зменшити загальну погрішність. Якщо в реальній ситуації «хороші» і «погані» об'єкти представлені в іншій пропорції, то результати, що видаються мережею, можуть виявитися невірними. Прикладом цього може бути завдання виявлення захворювань. Хай, наприклад, при звичайних обстеженнях в середньому 90% людей виявляються здоровими і мережа, таким чином, навчається на даних, в яких пропорція здорові/хворі рівна 90/10. Потім ця ж мережа застосовується для діагностики пацієнтів з певними скаргами, серед яких співвідношення здорові/хворі вже 50/50. В цьому випадку мережа ставитиме діагноз занадто обережно і не розпізнаватиме захворювання у деяких хворих. Якщо ж, навпаки, мережу навчити на даних «з скаргами», а потім протестувати на «звичайних» даних, то вона видаватиме підвищене число неправильних діагнозів про наявність захворювання. У таких ситуаціях почальні дані потрібно скоректувати так, щоб були враховані відмінності в розподілі даних (наприклад, можна повторити рідкісні спостереження або видалити що часто зустрічаються). Як правило, краще всього постаратися зробити так, щоб спостереження різних типів були представлені рівномірно, і відповідно цьому інтерпретувати результати, які видає мережа.

Здатність мережі, навченої на деякій безлічі даних, видавати правильні результати для достатньо широкого класу нових даних, у тому числі і не представлених при навчанні, називається властивістю узагальнення нейронної мережі.

Інший підхід до процедури навчання мережі можна сформулювати, якщо розглядати її як процедуру, зворотну моделюванню. В цьому випадку потрібно підібрати такі значення вагів і зсувів, які забезпечували б потрібну відповід-

ність між входами і бажаними значеннями на виході. Така процедура навчання носить назву адаптації і достатньо широко застосовується для настройки параметрів нейронних мереж.

Б.5 Вибір методу навчання нейронної мережі

Б.5.1 Критерій якості навчання. Методи навчання.

Фаза навчання є найбільш проблематичною при використанні нейронних мереж. Для БП мереж процес навчання є багатозмінним нелінійним оптимізаційним завданням. Її аналітичне рішення неможливе, що вимагає використання ітераційних методів оптимізації.

У загальному випадку початковими даними є: структура мережі, тренувальні дані і критерій оцінки якості тренування. Мета оптимізації – визначити параметри мережі, що забезпечують виконання критерію якості. Таким критерієм якості найчастіше виступає мінімум квадратичного функціонала:

$$J = \frac{1}{2} \sum_{q=1}^Q \sum_{i=1}^{S^L} (t_i^q - a_i^{qS^L})^2, \quad (\text{Б.1})$$

де J – функціонал;

Q – об'єм вибірки;

L – число шарів мережі;

q – номер вибірки;

S^L – число нейронів вихідного шаруючи;

$\mathbf{a}^q = [a_i^{qL}]$ – вектор сигналу на виході мережі;

$\mathbf{t}^q = [t_i^q]$ – вектор бажаних (цільових) значень сигналу на виході мережі

для вибірки з номером q .

Потім за допомогою того або іншого методу навчання визначаються значення параметрів (вагів і зсувів), що настраюються, мережі, які забезпечують мінімальне значення функціонала помилки.

Методика тренування нейронних мереж була в основному запозичена з відомих методів теорії оптимізації. Зважаючи на величезне число створених алгоритмів тут можна дати лише їх короткий огляд. Вживані методи навчання грубо можна розділити на наступні основні групи: методи глобальної оптимізації; градієнтні методи першого і другого порядку.

Істотною проблемою, що виникає при використанні градієнтних методів настройки параметрів ШНМ, є їх локальність. В той же час цільова функція (сумарна помилка на тренувальному наборі шаблонів або кумулятивна оцінка ефективності роботи навчаної системи) не унімодальна. Кількість локального оптимуму для більшості практичних завдань навчання обчислюється мільйонами при розмірності пошукового простору порядку $10^2 - 10^3$. Внаслідок цього результат навчання залежить від правильності вибору стартової крапки, і виникає необхідність багатократного повторення процедури настройки параметрів ШНМ. Перераховані проблеми можуть бути вирішені при використанні методів глобальної оптимізації. Найбільш ефективним з них є генетичний алгоритм (ГА). Розглядаючи ШНМ як єдиний набір параметрів, ГА здатний здійснювати її оптимальну настройку при розмірності пошукового простору достатньою для вирішення більшості практичних завдань.

У останні десятиліття розроблена безліч способів навчання ШНМ за допомогою ГА. Отримані результати доводять великі можливості такого симбіозу. Сумісне використання ШНМ і ГА алгоритмів має і ідеологічну перевагу тому, що вони відносяться до методів еволюційного моделювання і розвиваються в рамках одного підходу запозичення технікою природних методів і механізмів як найбільш оптимальних.

Слід зазначити, що завдання глобальної оптимізації вирішуються шляхом систематичного перебору всіх значень всіх аргументів, тому із-за експоненціального зростання складності завдання із зростанням її розмірності практичне застосування цих методів для великих мереж дуже утруднено.

У сучасних завданнях використовуються в основному градієнтні методи навчання. Вони відносно швидко знаходять оптимум. Проте при цьому відсут-

ний доказ того, що знайдений абсолютний екстремум. Численний досвід тренування мереж свідчить, що застосування методів локальної оптимізації сумісне з «виштовхуванням» мережі з локального мінімуму і послідовним збільшенням числа нейронів також приводить до успішного навчання мережі. У справжній роботі використовуються градієнтні методи навчання, тому розглянемо їх детальніше.

Б.5.2 Метод зворотного розповсюдження помилки.

Архітектура багатошарової мережі істотно залежить від вирішуваної задачі. Для лінійних нейронних мереж може бути встановлений зв'язок між сумарною кількістю вагів і зсувів з довжиною повчальної послідовності. Для інших типів мереж число шарів і нейронів в шарі часто визначається досвідом, інтуїцією проектувальника і евристичними правилами.

Навчання мережі включає декілька кроків:

- вибір початкової конфігурації мережі з використанням, наприклад, наступного евристичного правила: кількість нейронів проміжного шару визначається половиною сумарної кількості входів і виходів;
- проведення ряду експериментів з різними конфігураціями мережі і вибір тій, яка дає мінімальне значення функціонала помилки;
- якщо якість навчання недостатньо, слід збільшити число нейронів шаруючи або кількість шарів;
- якщо спостерігається явище перенавчання, слід зменшити число нейронів в шарі або видалити один або декілька шарів.

Нейронні мережі, призначені для вирішення практичних завдань, можуть містити до декількох тисяч параметрів, що настроюються, тому обчислення градієнта може зажадати вельми великих витрат обчислювальних ресурсів. З урахуванням специфіки багатошарових нейронних мереж для них розроблені спеціальні методи розрахунку градієнта, серед яких слід виділити метод зворотного розповсюдження помилки (ЗР), який був розроблений в 1986г. Потім з'явилися численні роботи по прискоренню збіжності алгоритму. Метод звор-

тного розповсюдження помилки є першим теоретично доведеним алгоритмом для настройки вагів при навчанні багат шарового персептрона. В даний час ЗР може розглядатися як еталонний тест, з яким порівнюються всі інші методи навчання.

Термін «зворотне розповсюдження» відноситься до процесу, за допомогою якого можуть бути обчислені похідні функціонала помилки по параметрах мережі. Цей процес може використовуватися у поєднанні з різними стратегіями оптимізації. Існує багато варіантів алгоритму зворотного розповсюдження. Звернемося до одного з них.

Розглянемо вираз для градієнта критерію якості по вагових коефіцієнтах для вихідного шару L :

$$\frac{dJ}{dw_{ij}^L} = \frac{\partial}{\partial w_{ij}^L} \left(\frac{1}{2} \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qL})^2 \right) = \sum_{q=1}^Q \sum_{k=1}^{S^M} (t_k^q - a_k^{qL}) \frac{\partial a_k^{qL}}{\partial w_{ij}^L}; \quad i = \overline{1, S^L}; \quad j = \overline{0, S^{L-1}},$$

де S^L – число нейронів в шарі;

a_k^{qL} – k -й елемент вектора виходу шаруючи L для елемента вибірки з номером q .

Правило функціонування шару L записується так:

$$a_k^{qL} = f_L \left(\sum_{i=0}^{S^{L-1}} w_{ij}^L a_i^{q(L-1)} \right), \quad k = \overline{1, S^L}.$$

Приймаючи до уваги, що

$$\frac{\partial a_k^{qL}}{\partial w_{ij}^L} = \begin{cases} 0, & k \neq i \\ f'_L(n_i^{qL}) a_j^{q(L-1)}, & k = i \end{cases} \quad i = \overline{1, S^L}; \quad j = \overline{0, S^{L-1}},$$

отримаємо

$$\frac{dJ}{dw_{ij}^L} = - \sum_{q=1}^Q (t_i^q - a_i^{qL}) f'_L(n_i^{qL}) a_i^{q(L-1)}.$$

Ввівши позначення для локальної помилки вихідного шару

$$\Delta_i^{qL} = (t_i^q - a_i^{qL}) f'_L(n_i^{qL}), \quad i = 1, \dots, S^L,$$

отримаємо

$$\frac{dJ}{dw_{ij}^L} = - \sum_{q=1}^Q \Delta_i^{qL} a_i^{q(L-1)}; \quad i = \overline{1, S^M}; \quad j = \overline{0, S^{M-1}}.$$

Перейдемо шару $L-1$ і запишемо співвідношення для настройки вагів w_{ij}^{L-1}

$$\begin{aligned} \frac{dJ}{dw_{ij}^{L-1}} &= - \sum_{q=1}^Q \sum_{k=1}^{S^L} (t_k^q - a_k^{qM}) f'_L(n_k^{qL}) \frac{\partial n_k^{qL}}{\partial a_i^{q(L-1)}} \frac{\partial a_k^{q(L-1)}}{\partial w_{ij}^{L-1}} a_i^{q(L-1)} = \\ &= - \sum_{q=1}^Q \sum_{k=1}^{S^L} (t_k^q - a_k^{qL}) f'_L(n_k^{qM}) w_k^L \frac{\partial n_k^{qL}}{\partial a_i^{q(L-1)}} f'_{L-1}(n_i^{q(L-1)}) a_j^{q(L-2)} = \\ &= - \sum_{q=1}^Q \Delta_i^{q(L-1)} a_j^{q(L-2)}, \end{aligned}$$

де Δ_i^{qL} – локальна помилка шару $L-1$

$$\Delta_i^{qL} = \sum_{k=1}^{S^M} (t_k^q - a_k^{qM}) f'_M(n_k^{qM}) w_k^M f'_{M-1}(n_i^{q(M-1)}) = \left(\sum_{k=1}^{S^L} \Delta_k^{qL} \right) f'_{L-1}(n_i^{q(L-1)}), \quad i = \overline{1, S^{L-1}}.$$

Для шарів $L-2, L-3, \dots, 1$ обчислення приватних похідних критерію J по елементах матриць вагових коефіцієнтів виконується аналогічно. У результаті отримуємо наступну загальну формулу:

$$\frac{dJ}{dw_{ij}^r} = \sum_{q=1}^Q \Delta_i^{q(r-1)} a_j^{q(r-1)}, \quad r = \overline{1, L}, \quad i = \overline{1, S^r}, \quad i = \overline{0, S^{r-1}}, \quad (\text{Б.2})$$

де r – номер шару

$$\Delta_i^{qr} = \left(\sum_{k=1}^{S^{r-1}} \Delta_k^{q(r-1)} w_{ki}^{r+1} \right) f'_r(n_i^{qr}), \quad r = \overline{1, M-1};$$

$$\Delta_i^{qL} = (t_i^q - a_i^{qL}) f'_L(n_i^{qL}), \quad i = \overline{1, S^{L-1}}.$$

На рис. Б.1 представлена схема обчислень, відповідна виразу (Б.2).

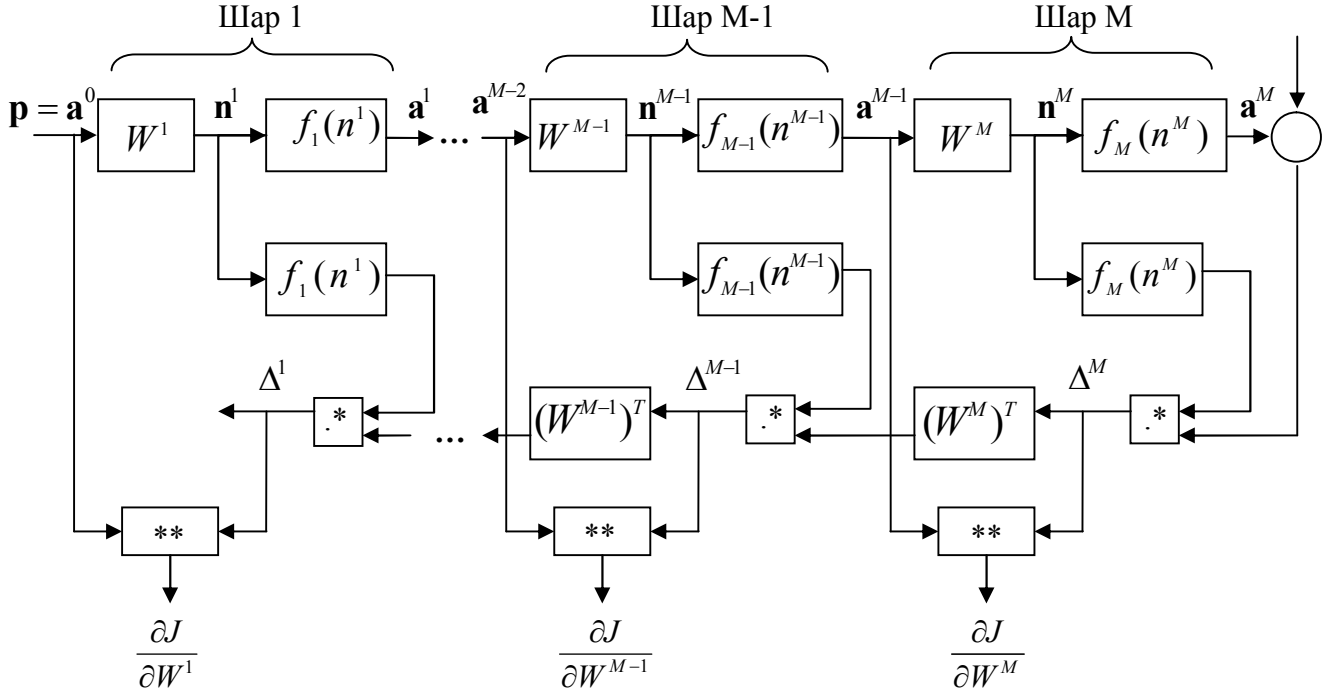


Рисунок Б.1 – Схема обчислення приватних похідних критерію J по елементах матриць вагових коефіцієнтів

На цій схемі символом $*$ позначена операція поелементного множення векторів, а символом $**$ – множення вектора Δ на $\mathbf{a}^T$; символ, що позначає номер елемента вибірки, скорочено опущений.

Б.5.3 Налаштування вагових коефіцієнтів мережі.

Методи, використовувані при навчанні нейронних мереж, багато в чому аналогічні методам визначення екстремуму функції декілька змінних. Вони діляться на дві категорії – методи першого і другого порядку.

У методах *першого* порядку використовується градієнт функціонала по параметрах, що настраюються, і на кожній ітерації k уточнення вектора вагів $\mathbf{w}_k$ проводиться по формулі

$$\mathbf{w}_{k+1} = \mathbf{w}_k - a_k \mathbf{g}_k, \quad (\text{Б.3})$$

де a_k – параметр швидкості навчання;

$\mathbf{g}_k$ – градієнт функціонала, відповідний ітерації з номером k .

Вектор в напрямі, протилежному градієнту, указує напрям найкоротшого спуску по поверхні функціонала помилки. Якщо реалізується рух в цьому напрямі, то помилка зменшуватиметься. Послідовність таких кроків врешті-решт приведе до значень параметрів, що настроюються, що забезпечують мінімум функціонала. Певну трудність тут викликає вибір параметра швидкості навчання a_k . При великому значенні параметра a_k збіжність буде швидкою, але існує небезпека пропустити рішення або піти в неправильному напрямі. Класичним прикладом є ситуація, коли алгоритм дуже поволі просувається по вузькому яру з крутими схилами, перестрибуючи з одного на іншій. Навпаки, при малому кроці, ймовірно, буде вибрано вірний напрям, проте при цьому буде потрібно дуже багато ітерацій. Залежно від прийнятого алгоритму параметр швидкості навчання може бути постійним або змінним. Правильний вибір цього параметра залежить від конкретного завдання і зазвичай здійснюється досвідченим шляхом; у разі змінного параметра його значення зменшується у міру наближення до мінімуму функціонала. Згідно (Б.3) уточнюються значення вагів $\mathbf{w}_k$ в методі ЗР в його стандартній реалізації.

У алгоритмах зв'язаного градієнта пошук мінімуму виконується уздовж зв'язаних напрямів, що забезпечує зазвичай швидшу збіжність, ніж при найшвидшому спуску. Всі алгоритми зв'язаних градієнтів на першій ітерації починають рух у напрямі антиградієнта

$$\mathbf{p}_0 = -\mathbf{g}_0.$$

Тоді напрям наступного руху визначається так, щоб він був зв'язаний з попереднім. Відповідний вираз для нового напрямку руху є комбінацією нового напрямку найшвидшого спуску і попереднього напрямку:

$$\mathbf{p}_k = -\mathbf{g}_k + \beta_k \cdot \mathbf{p}_{k-1}.$$

де $\mathbf{p}_k$ – напрям руху;

$\mathbf{g}_k$ – градієнт функціонала помилки;

β_k – коефіцієнт відповідають ітерації з номером k .

Коли напрям спуску визначений, то нове значення вектора параметрів, що настраюються, $\mathbf{w}_{k+1}$ обчислюється за формулою

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \cdot \mathbf{p}_k.$$

Гradientні методи першого порядку характеризуються малою швидкістю збіжності. Це породило помилкову думку про те, що всі нейронні мережі вимагають для тренування великих витрат часу.

Потужніші gradientні методи *другого* порядку базуються на методі локальної оптимізації Ньютона, який при пошуку оптимуму додатково враховує ще і вигин поверхні функціонала якості (другі похідні).

Основний крок методу Ньютона визначається по формулі

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{H}_k^{-1} \mathbf{g}_k,$$

де $\mathbf{H}$ – матриця Гессе других похідних функціонала якості по вагових коефіцієнтах мережі.

У багатьох випадках метод Ньютона сходиться швидше, ніж методи зв'язаного gradientа, але вимагає великих витрат із-за обчислення гессиана. Для того, щоб уникнути обчислення матриці Гессе, пропонуються різні способи її заміни наближеними виразами, що породжує так звані квазіньютонівські алгоритми такі як алгоритм, запропонований Бройлером, Флетчером, Гольдфарбом і Шанно, однокроковий алгоритм методу січних площин OSS, алгоритм LM Левенберга-Марквардта і ін.

У даній роботі використовуємо алгоритм Левенберга-Марквардта. Даний алгоритм реалізує наступну стратегію для оцінки матриці Гессе. Оскільки функціонал якості визначається як сума квадратів помилок, то гессиан може бути приблизно обчислений як

$$\mathbf{H} \cong \mathbf{J}^T \mathbf{J}, \quad (\text{Б.4})$$

а градієнт розрахований по формулі

$$\mathbf{g} \cong \mathbf{J}^T \mathbf{e},$$

де $\mathbf{J} = \frac{\partial J}{\partial \mathbf{w}}$ – матриця Якобі похідних функціонала помилки по параметрах, що настраюються;

$\mathbf{e}$ – вектор помилки мережі.

Матриця Якобі може бути обчислена на основі стандартного методу зворотного розповсюдження помилки, що істотно простіше за матрицю Гессе.

У алгоритмі Левенберга-Марквардта нове значення вектора параметрів $\mathbf{w}_{k+1}$, що настраюються, обчислюється за формулою

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}^T \mathbf{J} + \mu \mathbf{I})^{-1} \mathbf{J}^T \mathbf{e}_k.$$

Коли коефіцієнт μ рівний 0, отримуємо метод Ньютона з наближенням Гессиана у формі (Б.4), коли значення μ велике, отримуємо метод градієнтного спуску з маленьким кроком. Оскільки метод Ньютона має велику точність і швидкість збіжності поблизу мінімуму, завдання полягає в тому, щоб в процесі мінімізації щонайшвидше перейти до методу Ньютона. З цією метою параметр μ зменшують після кожної успішної ітерації і збільшують тільки тоді, коли пробний крок показує, що функціонал помилки зростає. Така стратегія забезпечує зменшення помилки після кожної ітерації алгоритму.

ДОДАТОК В

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ

В.1 Моделі систем, використовувані при нейромережевій ідентифікації

Основним питанням в нейрорегулювання є отримання адекватних моделей динамічних систем. Питання про існування дискретних моделей, або іншими словами про спостереженість нелінійних систем, є ключовим. У загальному випадку завдання побудови математичної моделі об'єкту полягає у виборі структури і оцінці її параметрів так, щоб при використанні критерію мінімуму деякої функції різниці розрахункових і експериментальних даних дотримувалася умова близькості моделі досліджуваному процесу. Застосування нейронних мереж як моделі може служити альтернативою класичним методам математичного опису систем, оскільки в цьому випадку точне знання внутрішніх процесів не є необхідною умовою моделювання.

При нейронній ідентифікації нелінійних систем найчастіше знаходять застосування наступні основні нелінійні дискретні моделі.

У найбільш загальному вигляді, прийнятому в класичній теорії управління, опис стохастичних нелінійних систем в просторі станів розглядався в роботі Наренди К.С. і Партасараті К [37].

$$z(k+1) = \Psi[z(k), u(k), \phi(k)] ;$$

$$y(k) = \Phi[z(k), u(k), \phi(k)],$$

де $z(k) = [z_1(k), z_2(k), \dots, z_{k_z}(k)]^T$ – вектор стану системи;

$u(k) = [u_1(k), u_2(k), \dots, u_{k_u}(k)]^T$ – вектор вхідних сигналів;

$y(k) = [y_1(k), y_2(k), \dots, y_{k_y}(k)]^T$ – вектор вихідних сигналів;

$\Psi[\cdot]$, $\Phi[\cdot]$ – деякі статичні нелінійні функції;

$\phi(k)$, (k) – шум процесу і шум вимірювань відповідно.

Леонартіс І. Дж. і Биллінгз С.Д.[38] запропонували для опису нелінійних систем використовувати так звану нелінійну модель авторегресії ковзаючого середнього з додатковими вхідними сигналами (NARMAX-модель).

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), e(k-1), \dots, e(k-n_e)] + e(k) \quad (\text{B.1})$$

де $y(k) \in \mathbf{R}^{M \times 1}$ – вихідний сигнал об'єкту;

$u(k) \in \mathbf{R}^{N \times 1}$ – вхідний сигнал об'єкту;

$e(k) \in \mathbf{R}^{M \times 1}$ – різниця вихідних сигналів об'єкту і моделі;

$f[\cdot]$ – нелінійна функція перетворення $f: \mathbf{R}^{(k_y M + k_u N + k_e M) \times 1} \rightarrow \mathbf{R}^{M \times 1}$;

n_y, n_u, n_e – порядки запізнювання по вихідному, вхідному сигналам об'єкту і помилці вимірювань відповідно.

Скалярна форма рівняння (Г.1) має вигляд:

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-n_y), y_2(k-1), \dots, y_2(k-n_y), y_M(k-1), \dots, y_M(k-n_y), \\ & u_1(k-1), \dots, u_1(k-n_u), u_2(k-1), \dots, u_2(k-n_u), u_N(k-1), \dots, u_N(k-n_u), \\ & e_1(k-1), \dots, e_1(k-n_e), e_2(k-1), \dots, e_2(k-n_e), e_M(k-1), \dots, e_M(k-n_e)] + e_i(k) \end{aligned} \quad (\text{B.2})$$

Нелінійні системи можуть бути представлені нелінійною NARX-моделлю:

$$y(k) = f[y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)] + e(k) \quad (\text{B.3})$$

або в скалярному вигляді

$$\begin{aligned} y_1(k) = & f_1[y_1(k-1), \dots, y_1(k-k_y), y_2(k-1), \dots, y_2(k-k_y), y_M(k-1), \dots, y_M(k-k_y), \\ & u_1(k-1), \dots, u_1(k-k_u), u_2(k-1), \dots, u_2(k-k_u), u_N(k-1), \dots, u_N(k-k_u)] + e_i(k). \end{aligned} \quad (\text{B.4})$$

Представлення об'єктів у вигляді NARMAX або NARX моделей грають фундаментальну роль при дослідженні нелінійних об'єктів за допомогою штучних нейронних мереж. При цьому важливо розглянути узагальнення моделей (B.1) - (B.4), що враховує різні види нелінійностей по вхідних і вихідних змін-

них. Наренда виділив чотири такі моделі:

– модель 1 (рис. В.1):

$$y(k+1) = \sum_{i=1}^{k_y-1} a_i y(k-i) + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\text{В.5})$$

– модель 2 (рис. В.2):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + \sum_{i=1}^{k_u-1} b_i u(k-i); \quad (\text{В.6})$$

– модель 3 (рис. В.3):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (\text{В.7})$$

– модель 4 (рис. В.4):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (\text{В.8})$$

Модель (В.4) є найбільш загальною зі всіх моделей вхід/вихід, описаних вище. При цьому модель (В.8) є аналітично вирішуваною і найбільш зручною для практичних застосувань. Внаслідок цього вона використовується в роботі при ідентифікації нелінійних об'єктів за допомогою штучних нейронних мереж.

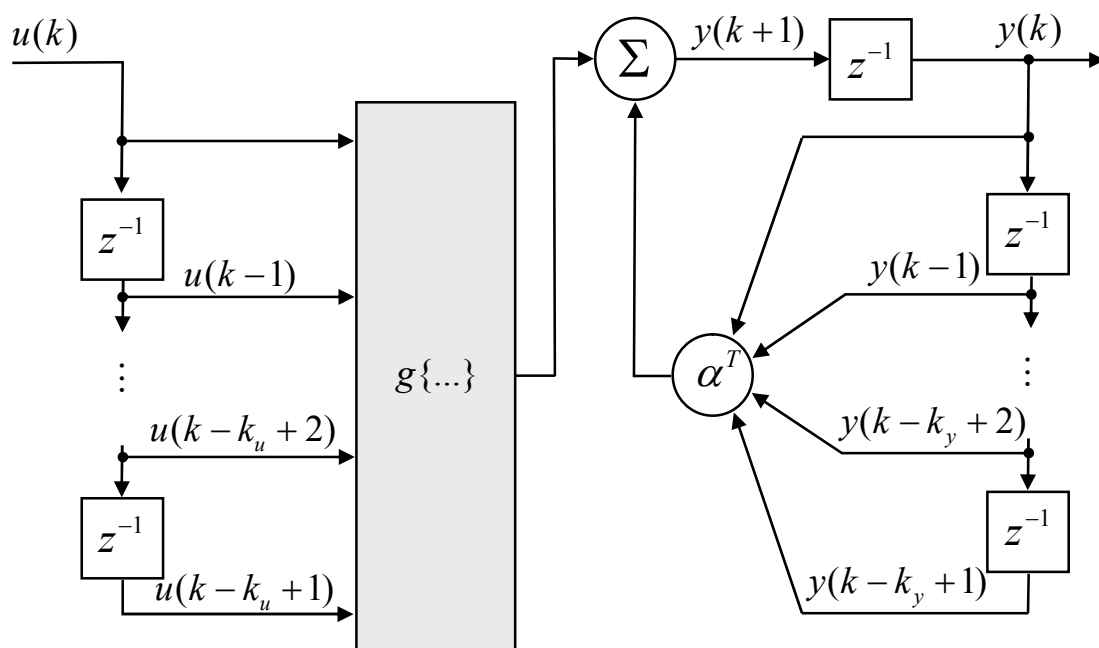


Рисунок В.1 – Модель 1. Вихід моделі нелінійно залежить від вхідних сигналів і лінійно від вихідних

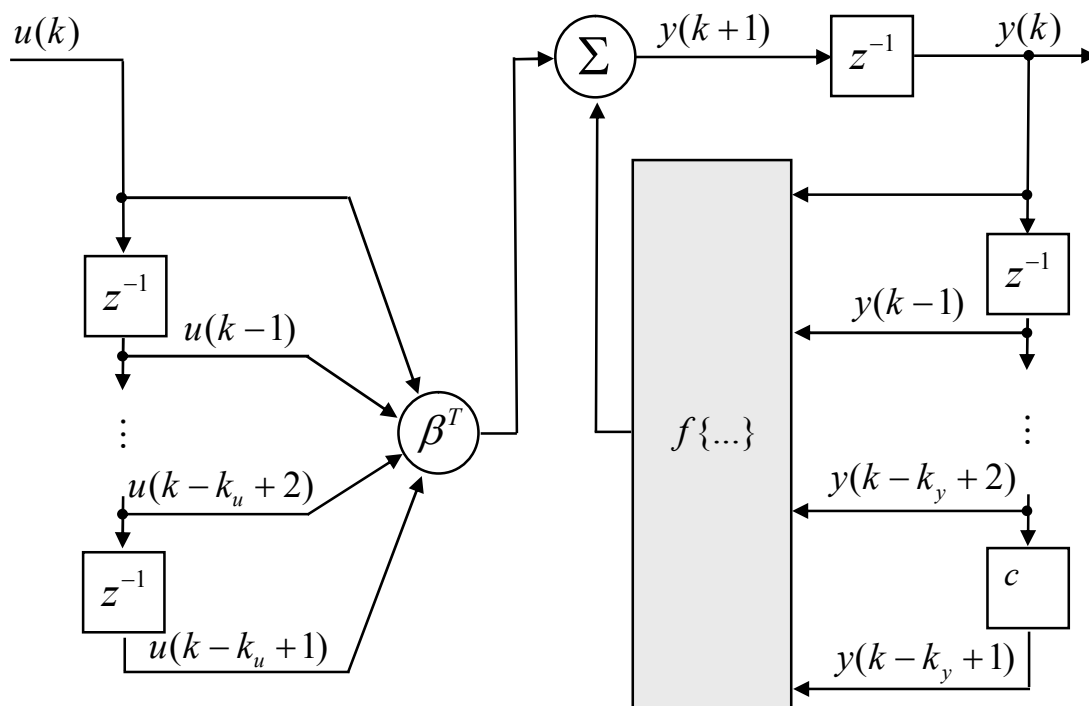


Рисунок В.2 – Модель 2. Вихід моделі лінійно залежить від вхідних сигналів і нелінійно від вихідних

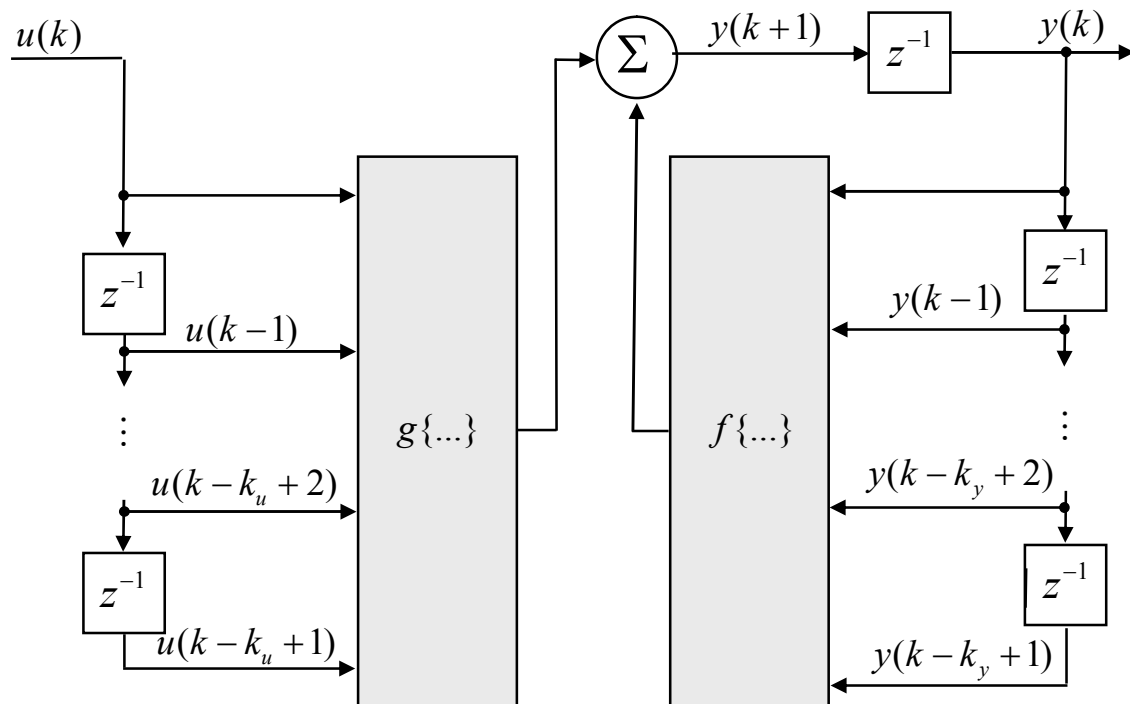


Рисунок В.3 – Модель 3. Резульгуючий сигнал є адитивною функцією нелінійних перетворень вхідних і вихідних сигналів

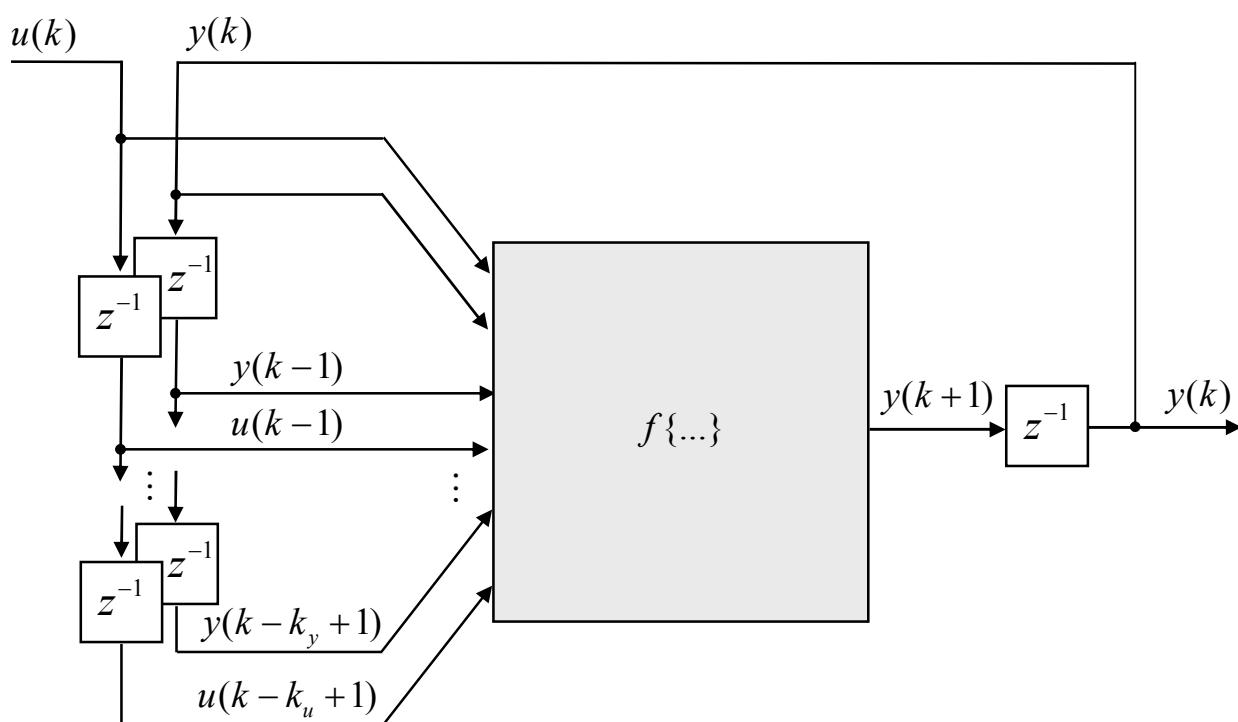


Рисунок В.4 – Модель 4. Вихід моделі нелінійно залежить як від вхідних, так і від вихідних сигналів

В.2 Нейронні мережі як універсальні моделі

Оскільки тип і взаємозв'язок нелінійностей в об'єкті, як правило, не відомі, для моделювання слід використовувати якомога більш універсальні структури моделей. Нейронні мережі є саме такими структурами.

Нелінійні моделі в просторі станів можуть бути реалізовані за допомогою рекурсивних нейронних мереж з внутрішніми зворотними зв'язками (мережі Елмана). Ідентифікація параметрів цих мереж вимагає спеціальної і щодо трудомісткої форми алгоритму навчання. Крім того, доказ стійкості таких нелінійних динамічних моделей практично неможливо зробити.

У роботі використовуються прямонаправленні мережі, що характеризуються наявністю зв'язків між нейронами тільки в прямому напрямі без зворотних зв'язків усередині мережі, – багат шаровий персептрон. Найбільш часто використовується мережа, багат шаровий персептрон (БП). За допомогою БП можна апроксимувати з бажаною точністю не тільки будь-які статичні функції. Попередні значення вхідних/вихідних координат у вхідному векторі дозволяють додати прямонаправленим мережам динамічні властивості.

Володіючи схожими з радіально-базисними мережами (РБМ) властивостями, БП забезпечує компактніше представлення нелінійного об'єкту, оскільки число параметрів в мережі РБМ експоненціально зростає із збільшенням розмірності вхідного простору, тоді як для БП ця залежність лінійна. Важливою є наочність, що виражається в тому, що персептронне представлення моделі еквівалентне традиційному. Це дозволяє використовувати для визначення параметрів ШНМ добре розвинені методи оцінювання, використовувані при традиційному підході.

Модель системи в просторі станів може бути успішно реалізована за допомогою прямонаправлених мереж, якщо змінні стани сформувати на виході мережі і потім завести їх на вхід мережі із затримкою. Але для цього потрібний, щоб всі змінні стани були вимірювані, оскільки тільки в цьому випадку можна забезпечити тренування мережі.

NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при нейронній ідентифікації систем. Для простоти викладу надалі розглядаємо системи з одним входом і одним виходом SISO (singl input singl output), хоча всі висновки можуть бути поширені і на MIMO (multi input multi output) системи. Прямий нейронний NARX моделі нелінійної динамічної системи відповідає наступний вираз:

$$\hat{y}(k) = f\left(y(k-1)\dots y(k-n_y), u(k-n_k)\dots u(k-n_u-n_k+1), \mathbf{w}\right), \quad (\text{B.9})$$

а нейронній інверсній NARX моделі:

$$\hat{u}(k) = g\left(y(k+n_k)\dots y(k+n_k-n_y), u(k-1)\dots u(k-n_u+1), \mathbf{w}\right) \quad (\text{B.10})$$

де f, g – функції, що реалізуються нейронними мережами;

u, y – вхідна і вихідна координати відповідно;

$\hat{u}, \hat{y}$ – значення координат на виході мережі;

$\mathbf{w}$ – матриця лінійних параметрів мережі;

n_k – запізнювання в системі, найчастіше $n_k = 1$.

Вирази (B.9) і (B.10) можна представити графічно у вигляді структурних схем, показаних на рис. B.5 і рис. B.6. NARMAX моделі характеризуються вищою якістю ідентифікації при зашумлених тренувальних даних. У [39] пропонується спосіб модифікації критерію якості алгоритму тренування. Результати ідентифікації за допомогою NARX моделей наближаються в цьому випадку до результатів, отриманих за допомогою NARMAX моделей. Дані NARX нейромоделі можуть бути успішно застосовані для вирішення поставлених завдань.

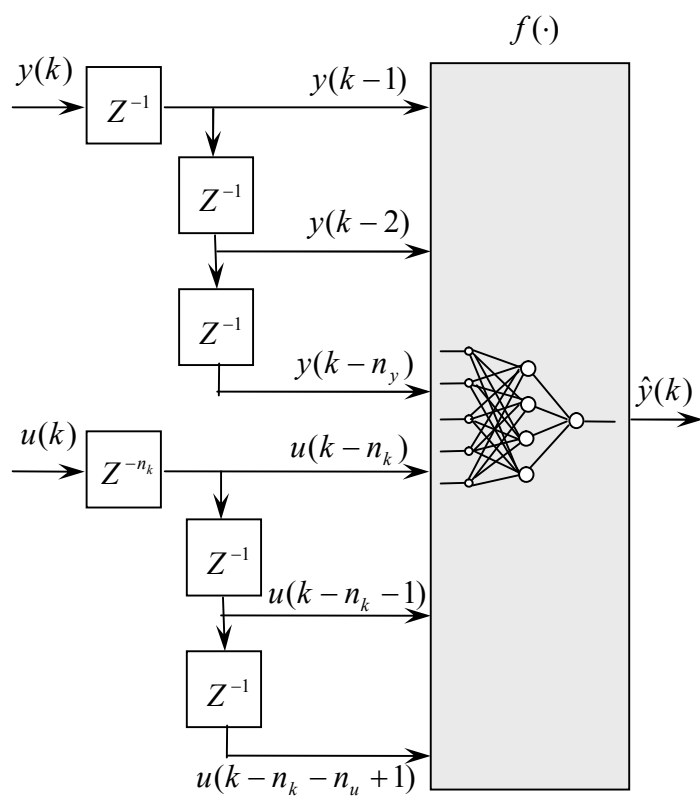


Рисунок В.5— Пряма мережева NARX модель

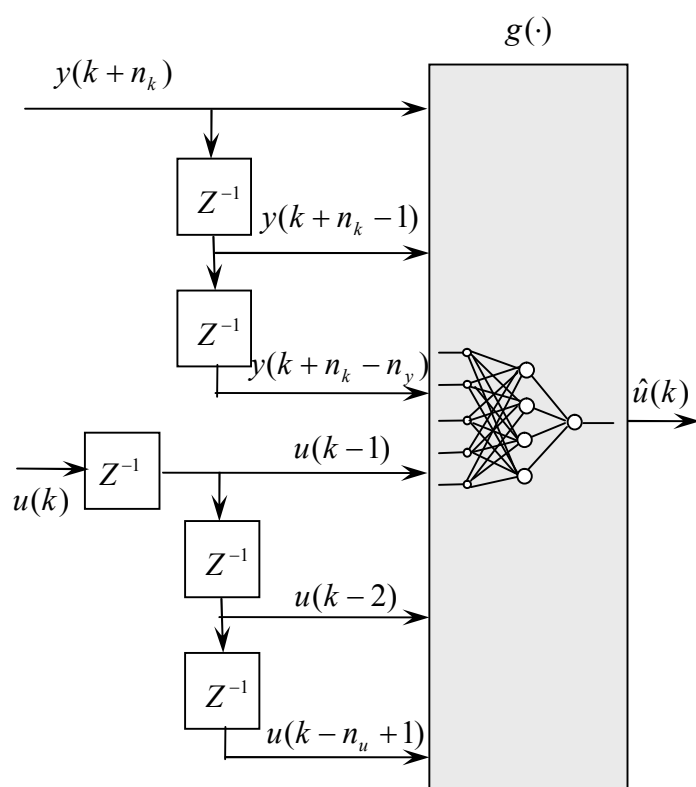


Рисунок В.6 – Інверсна мережева NARX модель

В.3 Ідентифікація динамічних систем за допомогою нейронних мереж

Завдання теорії регулювання, що займається пошуком моделей динамічних систем на основі апріорної інформації і вимірюваних вхідних/вихідних сигналів системи, носить назву ідентифікація. Відомі класичні методи ідентифікації нелінійних систем застосовні тільки у поєднанні з певним типом моделі (наприклад, ряди Вальтера, моделі Хаммерштейна і Вінера), що описує певний клас нелінійних динамічних систем. У роботі під термінами «ідентифікація системи», «тренування» і «навчання» розуміється процес підстроювання вагових коефіцієнтів мережі за допомогою градієнтних методів. Метою цього процесу є по можливості точне наближення поведінки моделі до спостережуваної поведінки динамічної системи.

Принципово розрізняють пряму і інверсну модель об'єкту. На рис. В.7 представлені схеми тренування цих нейронних моделей. Для спрощення на рис. В.7 не зображені ланки затримки за часом, що формують попередні значення координат на вході нейронних мереж.

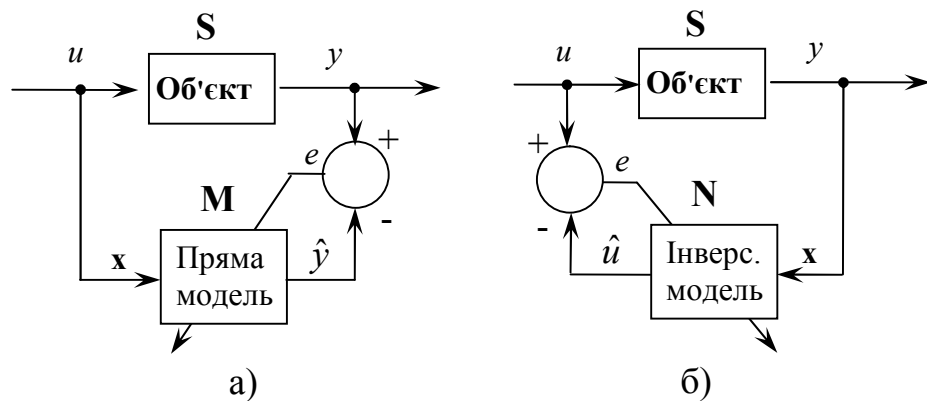


Рисунок В.7 – Схеми тренування прямої (а) і інверсної (б) моделей

Вхідний сигнал x нейронній моделі об'єкту M на рис. В.7,а ідентичний вхідному сигналу об'єкту S . Для інверсної нейромоделі N як вхідний сигнал використовується вихідний сигнал об'єкту S тоді як нейромодель повинна відновити вхідний сигнал об'єкту (див. рис. В.7,б). Поки присутня помилка між виходом нейромоделі і її зразковим сигналом, ця помилка використовується для

тренування мережі. У обох випадках ідентифікації обчислюється градієнт функціонала якості (див. таблицю В.1). Функціонал якості найчастіше при цьому квадратичний.

Таблиця В.1 – Обчислення градієнта при тренуванні нейромоделей

	Пряма модель	Інверсна модель
Бажаний вихідний сигнал	y	u
Вихідний сигнал нейромоделі	$\hat{y} = \mathbf{M}(\mathbf{x}, \mathbf{w})$	$\hat{u} = \mathbf{N}(\mathbf{x}, \mathbf{w})$
Помилка навчання	$e = y - \hat{y}$	$e = u - \hat{u}$
Градієнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{y}}{\partial \mathbf{w}}$	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{u}}{\partial \mathbf{w}}$

Оскільки $\frac{\partial \hat{y}}{\partial \mathbf{w}}$ і $\frac{\partial \hat{u}}{\partial \mathbf{w}}$ обчислювані, то це дозволяє вести підстроювання вагових коефіцієнтів мережі згідно розглянутим градієнтним методам. Результатом цього буде мінімізація функціонала якості. Оскільки набір тренувальних даних однозначно відображає лише поведінку модельованого об'єкту $\mathbf{S}$ у робочому діапазоні, то після завершення тренування маємо $\mathbf{M} \approx \mathbf{S}$ і $\mathbf{N} \approx \mathbf{S}^{-1}$.

Типова послідовність дій, що приводить до формування нейромоделі (ідентифікації) динамічної системи, відображена в таблиці В.2.

Таблиця В.2 – Ідентифікація за допомогою нейронних мереж

Шаг	Дія
1	Знімання тренувальних даних
2	Вибір структури мережі
3	Вибір типу моделі
4	Тренування мережі
5	Тестування нейромоделі
6	При незадов. результаті тестування, перейти до п.3 або до п.2

Оскільки тренувальні дані є носієм інформації про систему, що ідентифікується, то наявності, кількості і якості цих даних слід приділити особливу увагу. Тренувальних дані повинні мати наступні характеристики: якісна дискретизація, рівномірність дисперсії і величини варіації, рівномірність амплітуд всіх сигналів і їх середньоквадратичних значень.

При отриманні позитивного результату апроксимації, слід також перевірити здібність нейромоделі до узагальнення, тобто її здатність повторювати істотні характеристики об'єкту. Для цього тесту необхідний ще один набір даних, що не брав участь при тренуванні мережі. Цей тест дає остаточний висновок про адекватність нейромоделі.

Представлені теоретичні положення по нейронній ідентифікації, а також характеристики прямонаправлених мереж є основою успішного регулювання нелінійних динамічних систем.

ДОДАТОК Г

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% механізму підйому мостового крана
% 2024 рік

% Вихідні дані

kd=0.137;
CF=1/kd;
In=168;
Mn=In/kd;
ktp=51.3;
kn=0.055;
kc=0.057;
Tmc=0.005;
Ta=0.033;
Te=0.068;
Re=0.1568;
c12=5;
b12=0;
Jd=14.9;
Jm=0.18;
Js=Jd+Jm;
Uz=24;

w0=3;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);
%b12=0.005*c12;

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);

% 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) 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;
D=zeros(6,2);

```