

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

НЕЙРОМЕРЕЖЕВЕ УПРАВЛІННЯ ДВОМАСОВОЮ
ЕЛЕКТРОМЕХАНІЧНОЮ СИСТЕМОЮ МЕХАНІЗМУ ПІДЙОМУ
ШАХТНОЇ ПІДЙІМАЛЬНОЇ УСТАНОВКИ
(тема кваліфікаційної роботи)

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

_____ / Олександр БЛИЗНИЧЕНКО
(підпис) (ім'я та прізвище)

Керівник _____ / Тетяна ВАСИЛЕЦЬ
(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)

Харків – 2024

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

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

ЗАТВЕРДЖУЮ
Завідувач кафедри

(підпис)

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

« » 2024р.

ЗАВДАННЯ

на кваліфікаційну роботу магістрського рівня вищої освіти

студенту (ці) Олександр БЛИЗНИЧЕНКО
(ім'я, прізвище)

1. Тема роботи Нейромережеве управління двомасовою електромеханічною системою механізму підйому шахтної підйомальної установки
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 Корисний вантаж $m_b=11,5 \cdot 10^3$ кг; висота підйому $H=711$ м; маса порожнього скипа $m_{ск}=8,2 \cdot 10^3$ кг; діаметр барабана $D_b=5$ м; максимальна швидкість підйому $V_{max}=6,7$ м/с; число підйомів в годину $n=26$; шлях розвантаження $h_p=2,5$ м; нормальне прискорення $a=0,65$ м/с²; нормальне уповільнення $b=0,65$ м/с²; швидкість скипу при закритті затвора $V_z=0,5$ м/с; прискорення при закритті затвора $a_z=0,2167$ м/с²; маса погонного метра каната $m_{1к}=9,39$ кг. Система управління – астатична. Перерегулювання не більше 10%, час регулювання до 5 с.
4. Зміст роботи (перелік питань, що їх належить розробити). Обґрунтувати перспективність застосування нейромережевих технологій для управління багатомасовою електромеханічною системою механізму підйому шахтної підйомальної установки. Розрахувати систему автоматичного регулювання координат електроприводу. Дослідити динамічні характеристики системи. Розробити математична 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 помилок. 5. Системи нейрорегулювання. Структурні схеми. 6. Управління з прогнозом. Математична модель. 7. Система управління. Перехідні процеси. 8. Двомасова система. Математична модель. 9. Двомасова система. Перехідні процеси. 10. Регулятор з прогнозом. Структурна схема. 11. Система з нейрорегулятором. Вікна завдання параметрів. 12. Нейромережева модель об'єкту. Структурна схема. 13. Нейромережева система. Перехідні процеси. 14,15. Магістерська кваліфікаційна робота. Висновки

6. Консультант:

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

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

Керівник

(підпис)

Тетяна ВАСИЛЕЦЬ

(ім'я, прізвище)

Завдання прийняв до виконання

(підпис)

Олександр БЛИЗНИЧЕНКО

(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 шахтної підйимальної установки.	09.09.2024-18.09.2024	
2	Розрахунок системи підлеглого регулювання координат електроприводу з оптимізацією контурів за модульним ти симетричним критеріями. Моделювання системи на ЕОМ.	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.	16.10.2023-05.11.2023	
4	Синтез нейромережевої системи управління на ЕОМ з застосуванням пакету Neural Network Toolbox системи MATLAB. Моделювання нейромережевої системи.	06.11.2024-25.11.2024	
7	Оформлення роботи	26.11.2024-05.12.2024	

Студент (ка)

(підпис)

Олександр БЛИЗНИЧЕНКО

(ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка складається з: 223 сторінки, 71 рисунків, 1 таблиця, 47 використаних джерел, 5 додатків.

Мета роботи – розробка методів синтезу системи управління електроприводом механізму підйому шахтної підйимальної установки 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 шахтної підйимальної установки. 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 УІПА при читанні лекцій, виконанні курсових проєктів, проведенні практичних та лабораторних занять.

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

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

ABSTRACT

An explanatory message consists of: 223 pages, 71 figures, 1 table, 47 used sources, 5 additions.

The purpose of work is the development of methods of synthesis of electric drive control system of the mine lifting setting lifting mechanism 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 twomass electromechanics system. Computer simulations using a package of applied programs MATLAB explores the dynamic characteristics of neural network system.

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 mine lifting setting lifting mechanism. The control system of coordinates of mechanism electric drive is worked out. The parameters of automatic control system is 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 twomass system are calculated. The structure of the network control system is considered and the synthesis of neuroregulator NN Predictive Controller is executed in the system of MATLAB. The simulation of neural network control system with neuroregulation is conducted on a computer. It is set that synthesized the neural network system has high indexes of quality of functioning.

The main provisions of the master's work used in the educational process of the department AMandET of UEPA, such as lectures, the course projects, conducting practical and laboratory 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, MINE LIFTING SETTING, ELECTRIC DRIVE OF THE LIFTING MECHANISM, SYSTEM OF SLAVE CONTROL, ONEMASS AND TWOMASS SYSTEM, NEURAL NETWORK SYSTEM, NEUROREGULATOR

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

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

ЗМІСТ

ВСТУП	9
РОЗДІЛ 1. АНАЛІЗ СТАНУ ПРОБЛЕМИ І ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ	14
1.1 Перспективи розвитку теорії і застосування нейронних мереж	14
1.2 Огляд структур систем нейроуправління. Вибір структури системи	18
1.3 Вимоги до систем автоматизації і електроприводів шахтних підйомних машин	39
1.4 Нейромережевий підхід до управління. Постановка задачі дослідження	41
ВИСНОВКИ ДО РОЗДІЛУ 1	45
РОЗДІЛ 2 РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ ШАХТНОЇ ПІДЙІМАЛЬНОЇ УСТАНОВКИ ...	47
2.1 Технічні характеристики шахтної підйомальної установки. Вимоги до системи управління	47
2.2 Вибір системи електроприводу шахтної підйомальної установки	48
2.3 Вибір електродвигуна по потужності	51
2.3 Побудова швидкісної діаграми шахтного підйомника	52
2.5 Побудова діаграми навантаження	56
2.6 Перевірка двигуна по нагріву і перевантажувальній здатності	59
2.7 Вибір основного електроустаткування силового кола	61
2.8 Розрахунок параметрів силового кола електроприводу	62
2.9 Вибір структури системи автоматичного управління	67
2.10 Алгоритмічна схема системи	71
2.11 Синтез послідовного коректуючого пристрою контура напруги	73
2.12 Синтез послідовного коректуючого пристрою контура струму	77
2.13 Синтез послідовного коректуючого пристрою контура швидкості ..	81
2.14 Дослідження динамічних характеристик системи	88

ВИСНОВКИ ДО РОЗДІЛУ 2	92
РОЗДІЛ 3 ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ ПІДЙОМНОГО КАНАТА	94
3.1 Математическая модель двомасової системи	94
3.2 Розрахунок перехідних процесів двомасової системи	103
ВИСНОВКИ ДО РОЗДІЛУ 3	112
РОЗДІЛ 4 СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ.....	113
4.1 Регулятори, реалізовані в пакеті прикладних програм Neural Network Toolbox системи MATLAB	113
4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller	115
4.3 Синтез регулятора з прогнозом NN Predictive Controller з використанням системи MATLAB	118
4.4 Вибір параметрів нейрорегулятора NN Predictive Controller	162
4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller	164
ВИСНОВКИ ДО РОЗДІЛУ 4	169
ВИСНОВКИ	171
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	173
ДОДАТОК А. БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ.....	178
А.1 Класифікація штучних нейронних мереж	178
А.2 Базовий процесорний елемент (штучний нейрон).....	180
А.3 Статичні багатошарові нейронні мережі	182
ДОДАТОК Б. ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	187
Б.1 Здібність до навчання і накопичення інформації	187
Б.2 Апроксимаційні властивості нейронних мереж	187
Б.3 Апроксимація і пам'ять в багатошарових нейронних мереж	190

ДОДАТОК В. НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ.....	194
В.1 Загальна характеристика алгоритмів навчання.....	194
В.2 Метод і алгоритм зворотного розповсюдження помилки.....	197
В.3 Перенавчання і узагальнення нейронних мереж.....	208
ДОДАТОК Г. ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ.....	211
Г.1 Моделі систем, використовувані при нейромережевій ідентифікації	216
Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж	219
ДОДАТОК Д. ФАЙЛ ВИХІДНИХ ДАНИХ.....	222

ВСТУП

Актуальність теми. Для досягнення подальшого підвищення ефективності роботи підприємств гірничодобувної промисловості необхідна автоматизація виробничих процесів. Внаслідок цього підвищуються вимоги до систем управління і електроприводів підйомних установок, що є однією з основних частин технологічного комплексу по добутку корисних копалин підземним способом.

У зв'язку із зростанням виробничих потужностей гірських підприємств, збільшуванні глибини підйому, основною тенденцією розвитку шахтного підйому є підвищення вантажопідйомності і швидкості руху підйомних судин. Роботи із створення надійних підйомних установок, що забезпечують їх високопродуктивну і безпечну експлуатацію, направлені на вдосконалення як підйомних машин, так і систем управління електроприводами основних механізмів.

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

Мета і завдання дослідження. Метою роботи є розробка методів синтезу системи управління електроприводом механізму підйому шахтної підіймальної установки для підвищення показників якості функціонування системи на основі

нейромережових регуляторів. Досягнення поставленої мети здійснюється рішенням наступних основних задач.

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

Об'єктом дослідження є динамічні процеси в системі управління електроприводом механізму підйому шахтної підйимальної установки з урахуванням пружних властивостей підйомного канату.

Предметом дослідження є моделі і методи синтезу нейромережової системи управління двомасовою електромеханічною системою механізму підйому шахтної підйимальної установки.

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 нейромережової системи. Імітаційне моделювання дозволило проаналізувати якість функціонування систем і підтвердило ефективність отриманих результатів.

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

1. Вперше отримано наукове обґрунтування доцільності застосування нейромережових технологій для управління багатомасовою електромеханічною системою механізму підйому шахтної підйимальної установки.

2. Вперше запропоновано і синтезовано нейромережеву систему управління електроприводом механізму підйому шахтної підйимальної установки з урахуванням пружних властивостей підйомного канату, що має високі показники якості функціонування.

3. Вперше запропонована математична модель системи управління електроприводом механізму підйому шахтної підйимальної установки з урахуванням пружних властивостей підйомного канату у вигляді багат шарової прямонаправленої мережі – багат шарового персептрона.

4. Здійснений подальший розвиток математичних моделей об'єктів управління систем з 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 шахтної підйимальної установки, що забезпечує високоякісне регулювання.

Основні положення роботи використовуються у навчальному процесі кафедри автоматизації, метрології та енергоефективних технологій Української інженерно-педагогічної академії при виконанні магістерських кваліфікаційних робіт, читанні лекцій, виконанні курсових проектів, 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 властивостей підйимального канату, 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.

В роботі наведені посилання на авторів і відповідні джерела при використанні відомих залежностей, експериментальних даних і наукових положень.

Апробація результатів роботи. Основні положення та наукові результати були апробовані на науковій конференції студентів УПА (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 223 сторінки, серед них: 44 рисунків за текстом, 27 рисунків на 25 окремих сторінках, 1 таблиця, 5 додатків на 46 сторінках, список літературних джерел з 47 найменування на 5 сторінках.

РОЗДІЛ 1

АНАЛІЗ СТАНУ ПРОБЛЕМИ І ПОСТАНОВКА ЗАДАЧІ

ДОСЛІДЖЕННЯ

1.1 Перспективи розвитку теорії і застосування нейронних мереж

Штучні нейронні мережі - це системи паралельних обчислень, які складаються з багатьох простих процесорів, які взаємодіють між собою. Ці процесори, як правило, мають просту будову, особливо у порівнянні з тими, що використовуються у персональних комп'ютерах. Кожен процесор у такій мережі опрацьовує тільки сигнали, які він отримує періодично, та сигнали, які він періодично висилає іншим процесорам. Проте, через їх взаємодію у великій мережі, навіть такі прості процесори можуть виконувати складні завдання..

Розробка штучних нейронних мереж почалася ще на зорі ХХ сторіччя, але тільки в 90-х роках, коли були подолані деякі теоретичні бар'єри, а обчислювальні системи стали достатньо потужними, нейронні мережі отримали широке визнання. Слово "штучні" в даному контексті іноді використовується для того, щоб підкреслити, що йдеться про штучний пристрій, а не про реальні біологічні нейронні системи типу тієї, яку має людина. Створення штучних нейронних мереж було обумовлено спробами зрозуміти принципи роботи людського мозку і, без сумніву, це впливатиме і на подальший їх розвиток. Проте, порівняно з людським мозком, штучні нейронні мережі сьогодні є вельми спрощеними абстракціями. Коли ясно, в якому контексті обговорюються ці мережі, слово "штучний" зазвичай опускають. Крім того, коли потрібно підкреслити обчислювальні можливості, а не біологічну відповідність, штучні нейронні мережі називають конекціями. При цьому метою "конекціоністів" є наділ нейронної мережі можливістю вирішувати конкретні задачі, а не імітувати з максимальною точністю біологічний процес.

Хоча нейронні мережі можуть мати апаратне втілення для швидкості обчислень (такі втілення існують), більшість досліджень проводиться через про-

грамне моделювання на звичайних комп'ютерах. Це моделювання дозволяє відносно дешево і гнучко перевіряти нові ідеї, і для багатьох випадків воно є достатньо адекватним.

Рішення на основі нейронних мереж відрізняються від звичайного програмного забезпечення: вони зазвичай "навчаються", а не програмуються на пряму. Фактично, в більшості випадків нейронні мережі використовуються у тих ситуаціях, коли складно написати програму, або коли рішення, яке знайде нейронна мережа, виявиться більш вдалим.

Наприклад, в експертів з продажу нерухомості може бути великий досвід у визначенні чинників, що впливають на ціни на будинки. Проте інколи є складні чинники, які важко пояснити програмістам. Тут нейронні мережі стають потрібними, оскільки вони можуть уявляти ситуації гнучкіше і поступово покращувати точність своїх прогнозів, пристосовуючись до ринкових змін.

Розвиток рішень, що базуються на нейронних мережах, набуває зростаючої актуальності, а безперечно, у майбутньому, з розширеним розумінням їх фундаментальних принципів, можливості для створення відповідних пристроїв будуть зростати. Вже зараз ми спостерігаємо вражаючі досягнення. Розмаїття застосувань нейронних мереж безмежне: виявлення фальшивих кредитних карток, передбачення коливань на фондовій біржі, складання кредитних планів, оптичне розпізнавання символів, профілактика і діагностика хвороб, технічний стан машин, автоматизоване управління автомобілем, прийняття рішень під час посадки пошкодженого літального апарату та інше.

Майбутні досягнення у розробці штучних нейронних мереж будуть залежати від глибшого розуміння принципів функціонування людського мозку. Проте тут є важлива взаємодія: штучні нейронні мережі допомагають покращити наше розуміння процесів у нервовій системі, виступаючи як моделі цих процесів. Нейронні мережі використовуються в різноманітних галузях техніки, де вони здійснюють рішення різноманітних прикладних завдань [1-12]. Вони застосовуються:

1. У космонавтиці і аеронавтиці для створення автоматичних систем пілотування літаків, моделювання траєкторій польоту, розробки перспективних конструкцій автопілотів, виявлення несправностей літальних апаратів, керування повітряним рухом та обробки аерокосмічної інформації.
2. У автомобілебудуванні для розробки автоматичних систем управління.
3. У банківській справі для автоматичного читання документів, контролю, оцінки ефективності кредитних вкладень.
4. У військовій справі для управління зброєю, стеження за цілями, розпізнавання об'єктів, обробки звукових, радіолокаційних і телевізійних зображень, постановки і придушення перешкод, ідентифікації сигналів і зображень.
5. У електроніці для управління розміщенням мікросхем на платі, моделювання відмов мікросхем, систем машинного зору і синтезу мови.
6. У фінансовій справі для аналізу кредитних потоків, оцінки нерухомості, загального фінансового аналізу, прогнозування вартості валюти.
7. У страховій справі для визначення оптимальної стратегії страхування.
8. У промисловому виробництві для управління виробничими процесами, аналізу якості, контролю, тестування продукції, аналізу якості зварювальних робіт, оптимізації виробничих операцій, аналізу хімічних речовин та прогнозування ціни продукту.
9. У медицині для аналізу ракових клітин, діагностики захворювань, конструювання протезів, оптимізації часу трансплантації, планування витрат лікарні та консультацій у відсутність фахівців.
10. У нафтогазовій промисловості для розвідки родовищ.
11. У робототехніці для управління роботами, побудови контролерів роботів та систем технічного зору.
12. При передачі даних для стиснення і відображення даних, розпізнавання мови, класифікації звуків, перетворення тексту в мову і синхронного перекладу.

Існує безліч інших областей, де використання нейронних мереж може знайти застосування. Широкий спектр використання цих мереж, великі інвестиції в розробку програмного забезпечення та апаратури для їх реалізації свідчать про великий інтерес до розвитку штучних нейронних мереж.

Останнім часом штучні нейронні мережі активно використовуються для синтезу систем управління. Хоча традиційні методи синтезу автоматичних систем управління електроприводами, такі як модальне управління, забезпечують високі показники регулювання, їх практична реалізація стикається з певними технічними проблемами. Серед них – потреба в вимірюванні важковимірюваних координат, таких як пружний момент, а також обмеження, які не дозволяють реалізувати синтезовані параметри в реальних електроприводах через наявні технічні обмеження. Ідея заміни лінійних регуляторів на нелінійні, які при меншій кількості сигналів від вимірюваних координат забезпечували б потрібну вихідну управляючу дію, є цілком логічною. Такий підхід реалізується за допомогою нелінійного регулятора у вигляді нейрорегулятора, який є нейронною мережею, що працює за принципами біологічних нейронних мереж.

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

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

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

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 Г.

1.2 Огляд структур систем нейроуправління. 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

$$E = \frac{1}{2} e^2 = \frac{1}{2} (d(k) - y(k))^2 \quad (1.2)$$

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

Вихідний сигнал мережі може бути обчислений таким чином:

$$u = y_1^2 = f_{log}(a_1^2), \quad (1.3)$$

$$a_1^2 = \sum_{j=1}^{N^1} w_{1j}^2 y_j^1 + b_{10}^2 n_0^1, \quad (1.4)$$

$$y_j^1 = f_{log}(a_j^1), \quad (1.5)$$

$$a_j^1 = w_{j1}^1 d + b_{j0}^1 n_0^0. \quad (1.6)$$

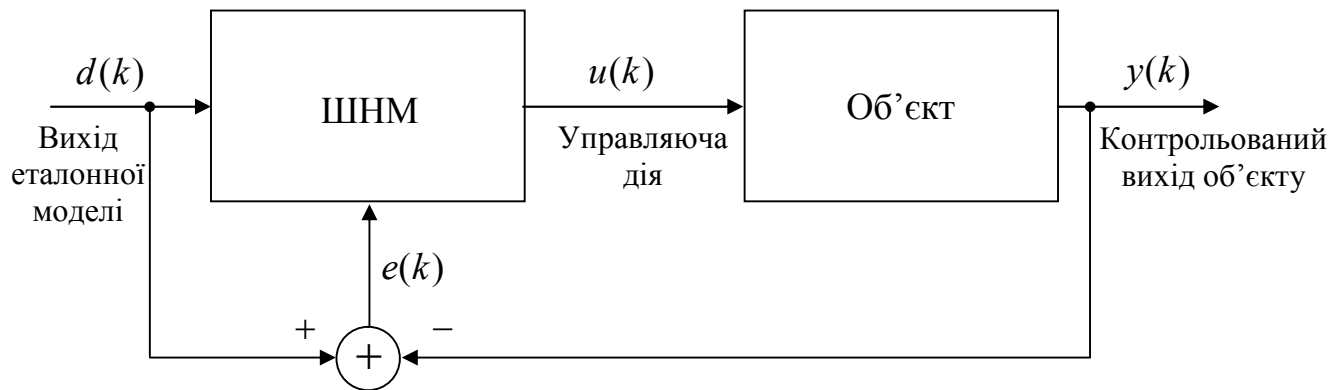


Рисунок 1.1 – Спеціалізоване нейроуправління

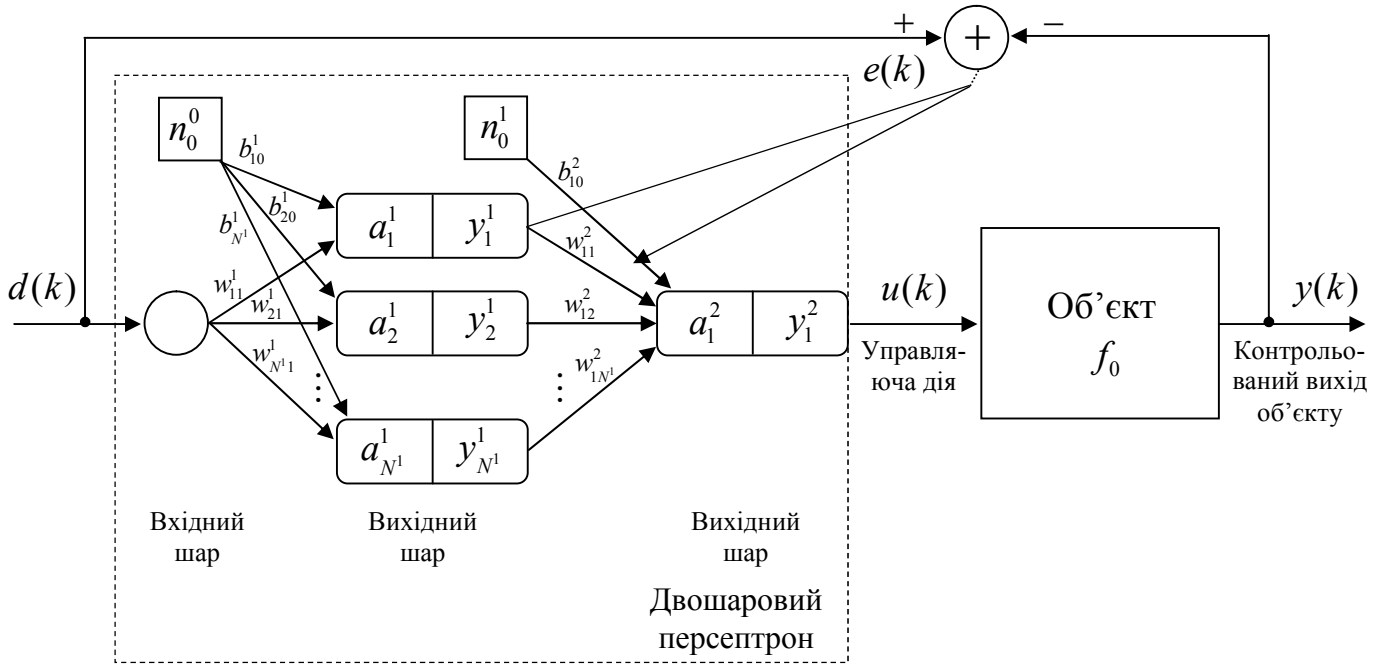


Рисунок 1.2 – Структура нейрорегулятора на основі двошарового персептрону

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

$$\Delta w_{1j}^2 = -\eta \frac{\partial E}{\partial w_{1j}^2} = -\eta \frac{\partial E}{\partial a_1^2} \frac{\partial a_1^2}{\partial w_{1j}^2}. \quad (1.7)$$

З (1.4) отримуємо

$$\frac{\partial a_1^2}{\partial w_{1j}^2} = y_j^1. \quad (1.8)$$

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

$$\delta^2 = \frac{\partial E}{\partial e} \frac{\partial e}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial a_1^2} = e \frac{\partial y_0}{\partial u} \frac{\partial u}{\partial a_1^2} = e f'_0(u) u (1 - u). \quad (1.9)$$

Тепер можемо записати загальну формулу настройки вагів вихідного шару:

$$\Delta w_{1j}^2 = -\eta \delta^2 y_j^1. \quad (1.10)$$

Аналогічно для нейронів прихованого шару, можемо записати

$$\Delta w_{j1}^1 = -\eta \delta_j^1 d, \quad (1.11)$$

де

$$\delta_j^1 = \frac{\partial E}{\partial a_1^2} \frac{\partial a_1^2}{\partial y_j^1} \frac{\partial y_j^1}{\partial a_j^1} = \delta^2 w_{1j}^2 y_j^1 (1 - y_j^1). \quad (1.12)$$

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 (1.9) і (1.12) 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 (1.10).
4. Коректуються ваги прихованого шару по формулі (1.11).
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

го апроксимацію. Для знаходження $f'_0(u(k))$ можна застосувати наближену формулу

$$f'_0(u(k)) \approx \frac{f_0(u(k))}{\Delta u(k)}, \quad (1.13)$$

де

$$\Delta f_0(u(k)) = f_0(u(k-1)) - f_0(u(k-2)), \quad (1.14)$$

$$\Delta u(k) = u(k-1) - u(k-2). \quad (1.15)$$

1.2.2 Схема зворотного розповсюдження в часі.

Інший підхід полягає в побудові НММ об'єкту (нейроеммулятора), 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 нейроеммулятора дозволяє обчислити Якобіан системи, замінюючи об'єкт його моделлю. В цьому випадку отримаємо:

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

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

Добитися підвищення швидкості навчання нейроеммулятора 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 ви-

користання 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» [15-19].

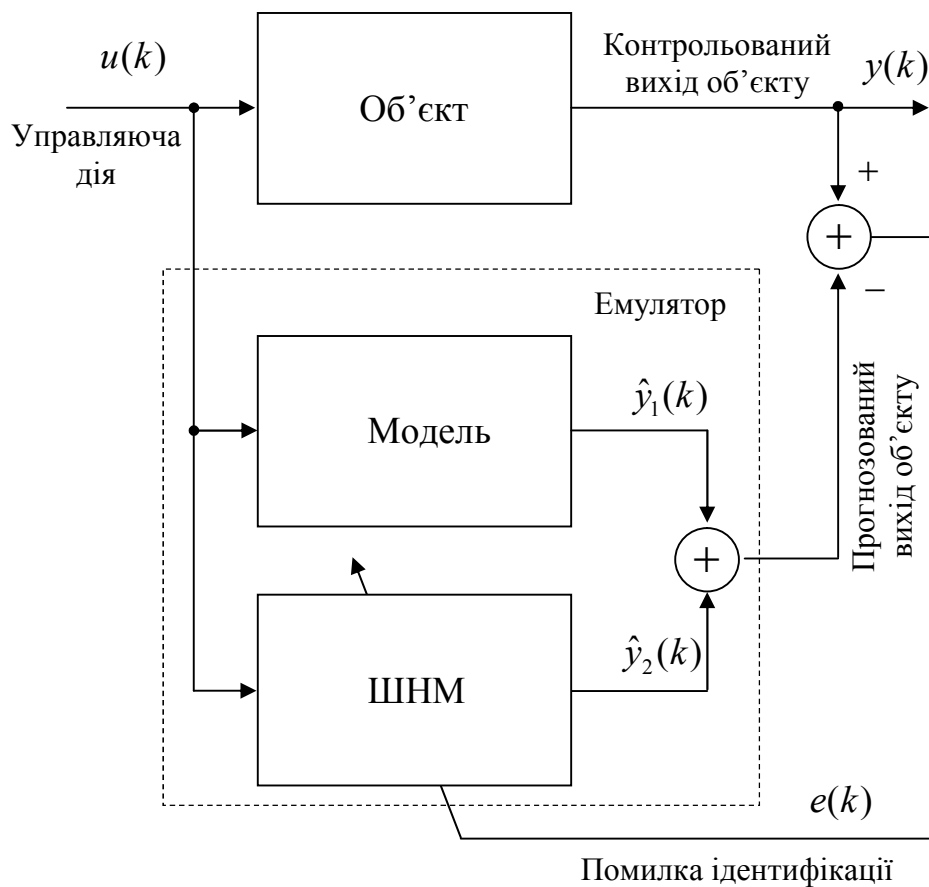


Рисунок 1.3 – Структура нейроемулятора з умовною моделлю

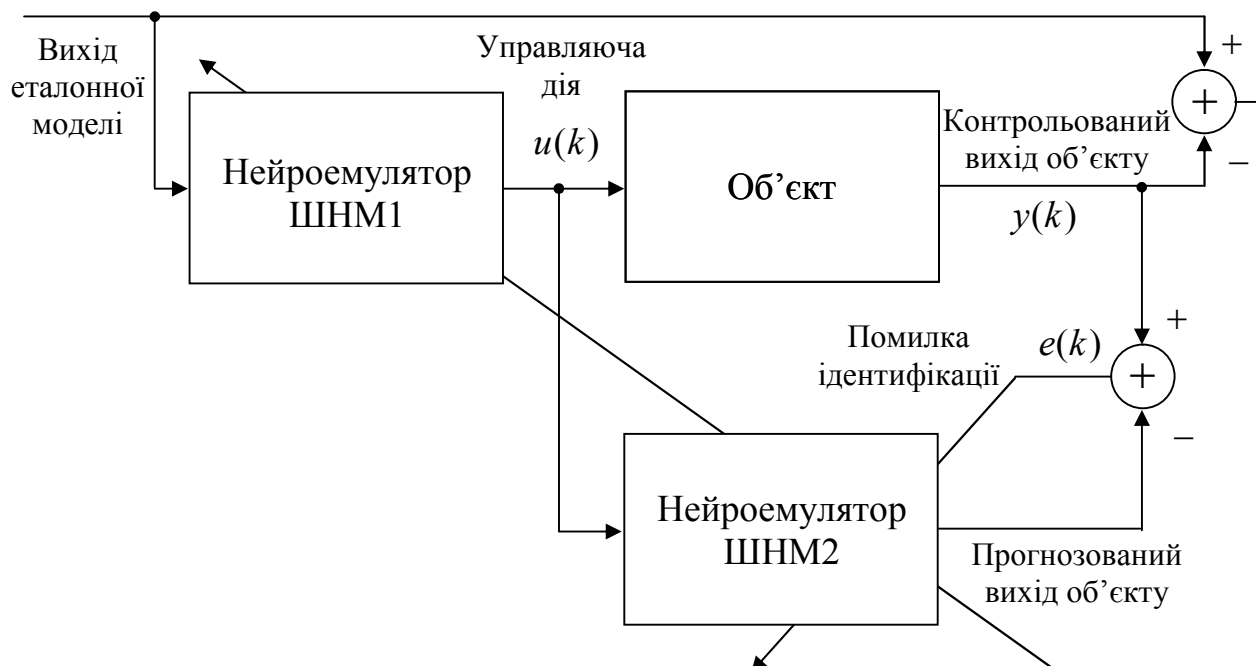


Рисунок 1.4 – Загальна схема інверсно-динамічного нейроуправління

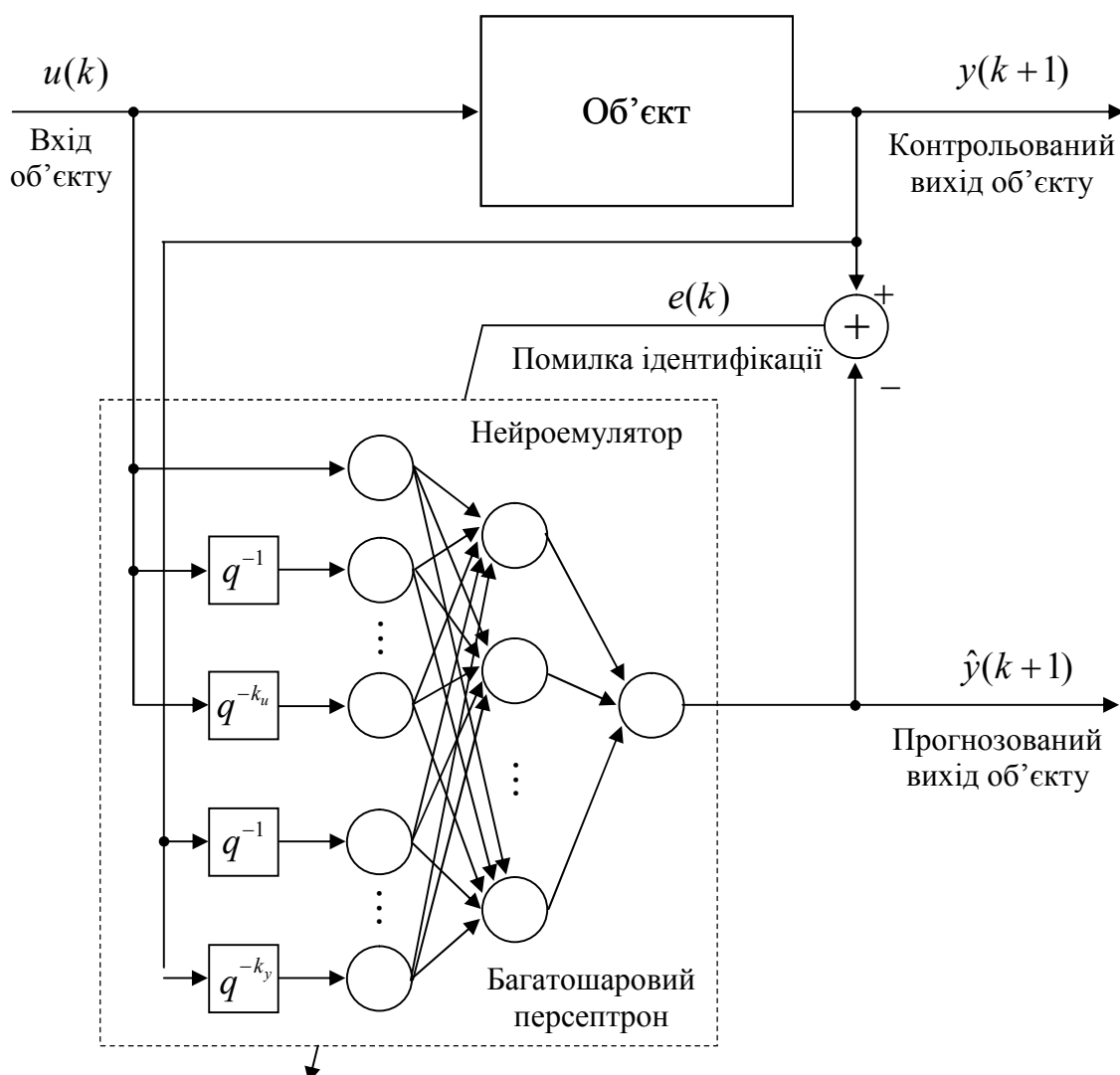


Рисунок 1.5 – Структура нейроемулятора

У роботі [20] 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.

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 [21,22] (рис.1.9).

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 [23].

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 [24]. Управління по зворотному зв'язку засноване на помилці між виходом еталонної моделі і виходом об'єкту управління $e(k) = d(k) - y(k)$.

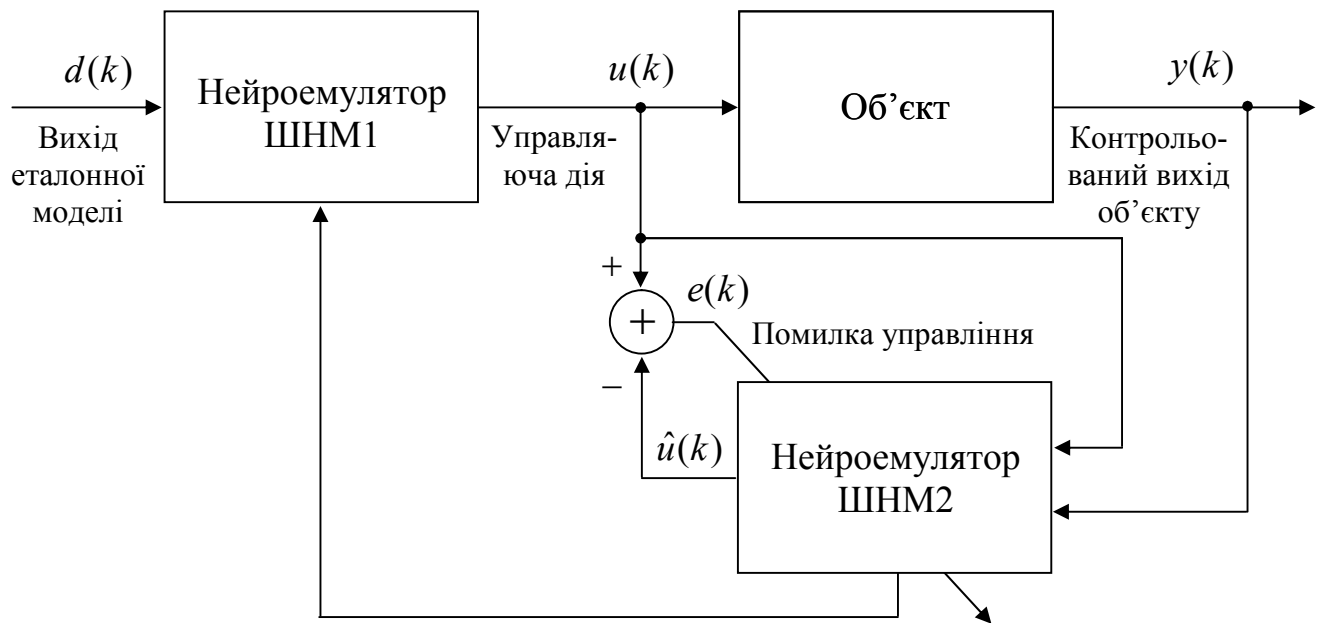


Рисунок 1.6 – Схема зворотного відображення

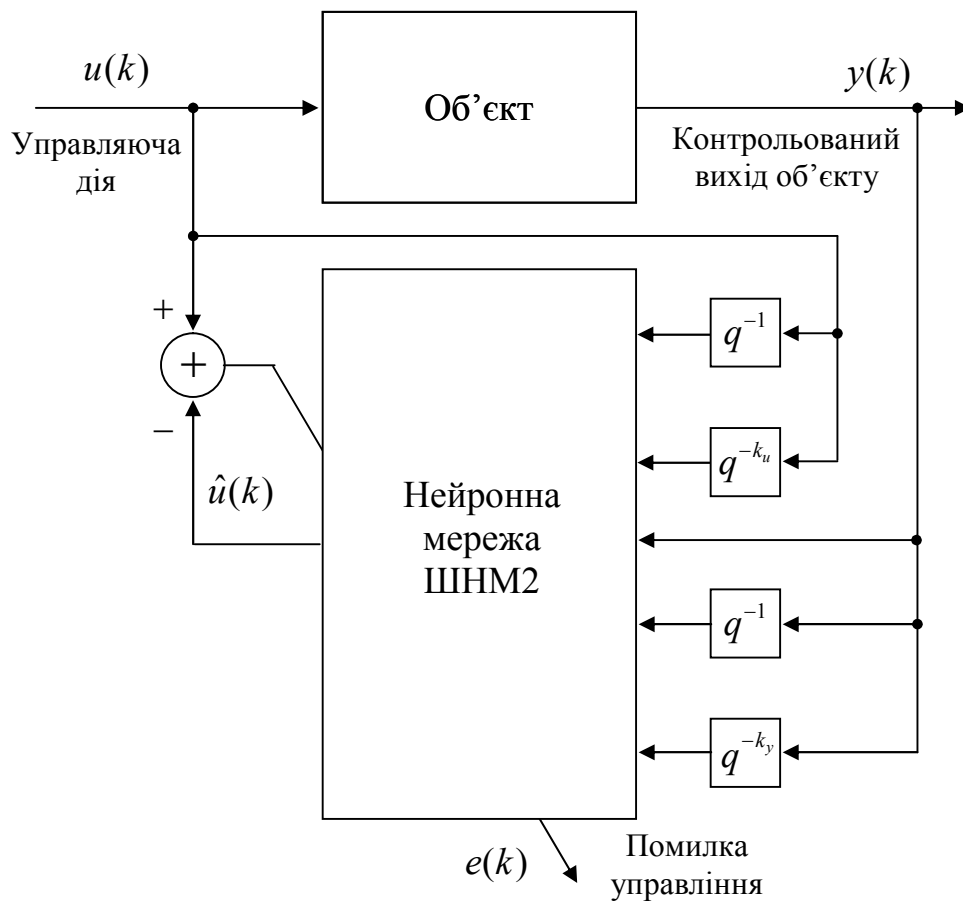


Рисунок 1.7 – Структура нейроемулятора для схеми зворотного відображення

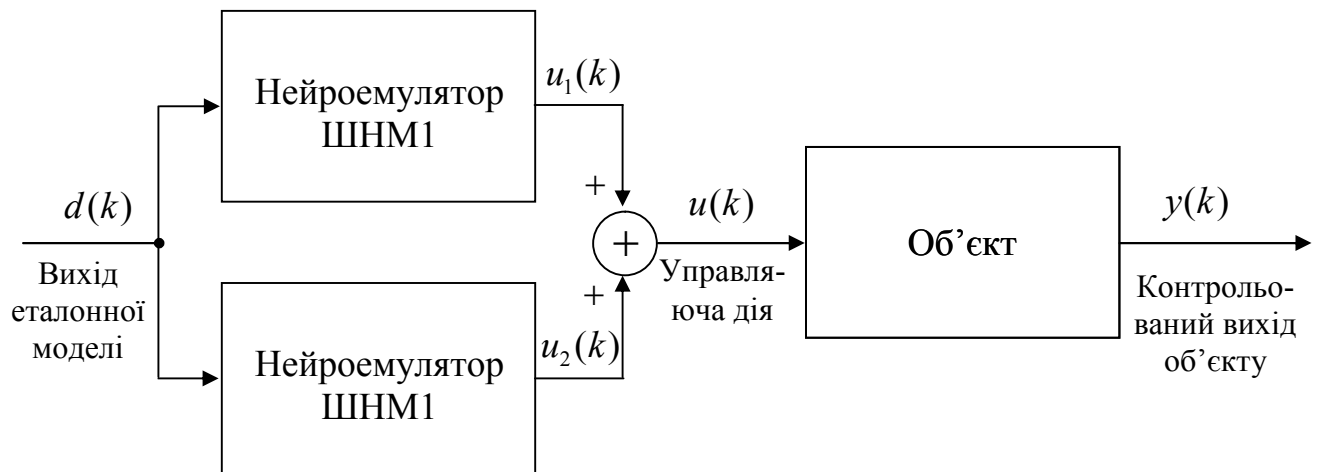


Рисунок 1.8 – Паралельна схема нейроуправління

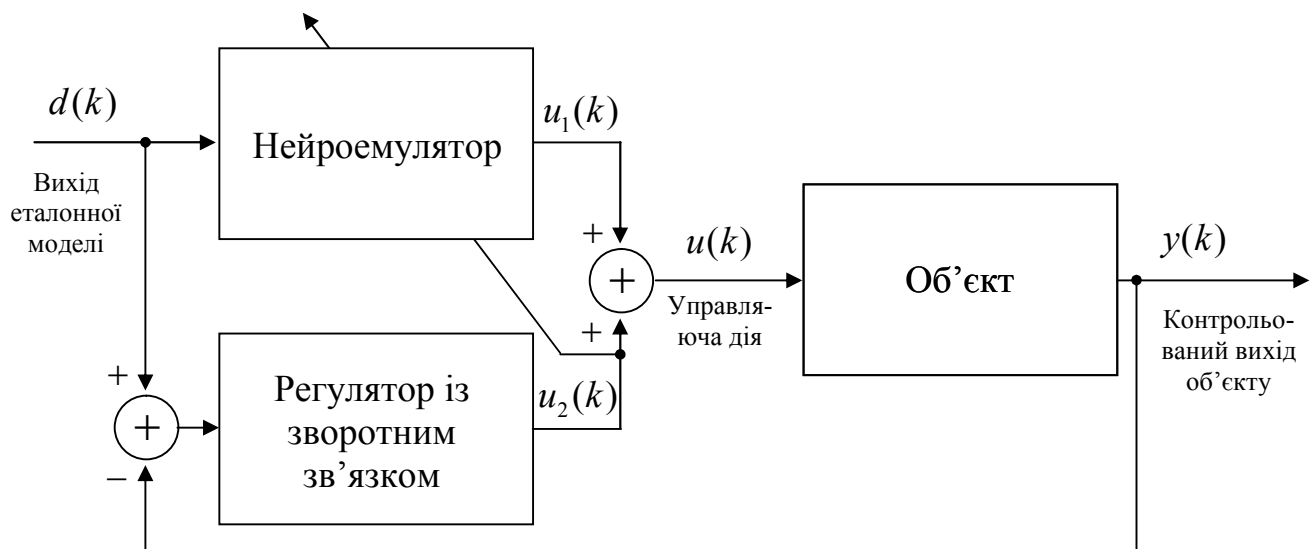


Рисунок 1.9 – Нейроуправління по помилці зворотного зв'язку

Як регулятор із зворотним зв'язком зазвичай використовується ПІД-регулятор. Друга частина включає нейроемулятор, що реалізовує динаміку об'єкту. Навчання нейроемулатора 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 викори-

стовуються 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.

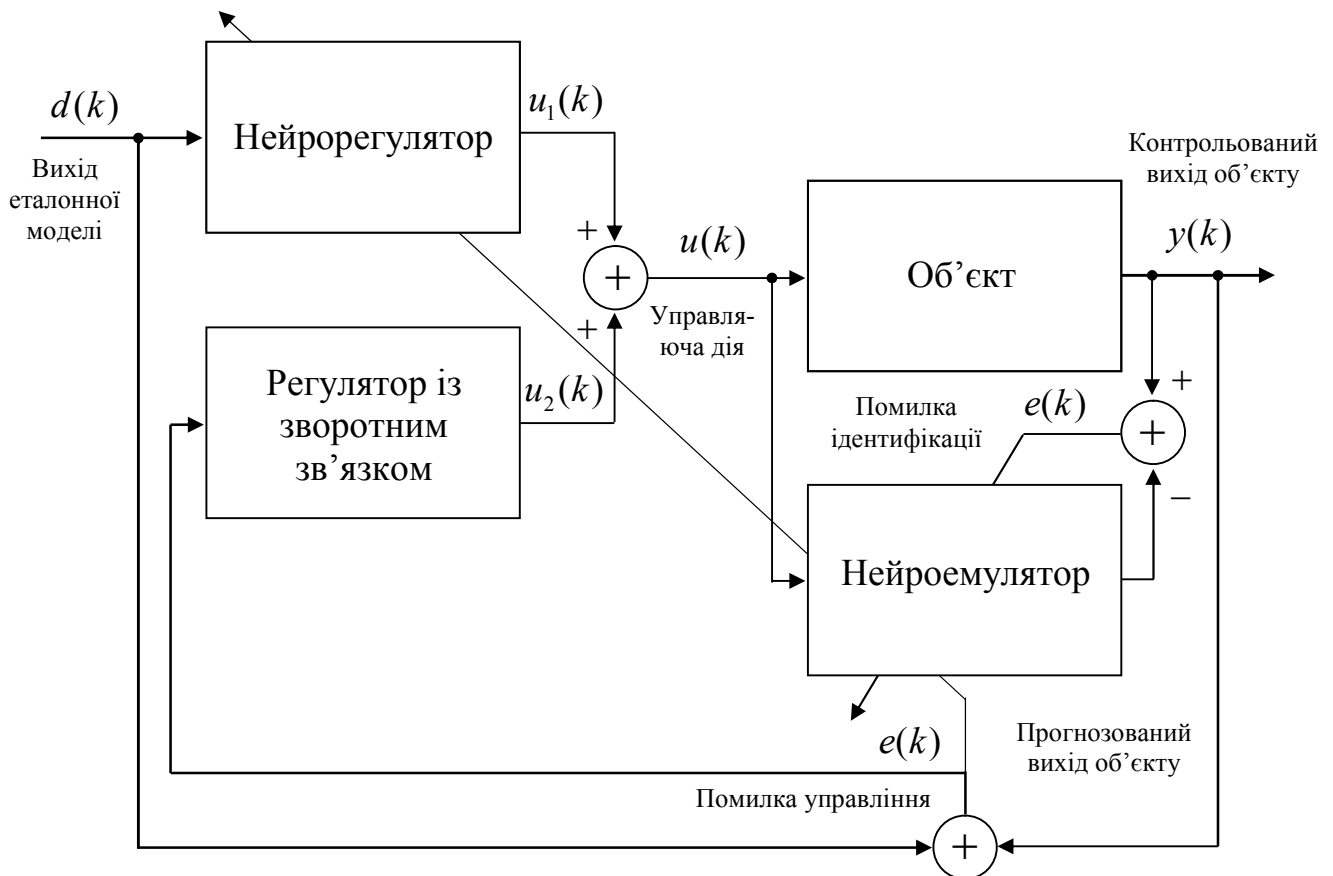


Рисунок 1.10 – Узагальнена схема нейроуправління по помилці зворотного зв'язку

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

$$\begin{aligned} y(k+1) &= f_0(u(k)) = f_0(u_1(k)) - u_2(k); \\ u_2(k) &= \Phi(k). \end{aligned} \quad (1.17)$$

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:

$$\Delta w_{1j}^2(k+1) = -\eta \frac{\partial E}{\partial w_{1j}^2(k)}, \quad (1.18)$$

а корекція вагів нейронів прихованого шару відповідно до правила:

$$\Delta w_{j1}^1(k+1) = -\eta \frac{\partial E}{\partial w_{j1}^1(k)}. \quad (1.19)$$

Використовуючи позначення для локальної помилки вихідного шару $\delta^2 = \frac{\partial E}{\partial a_1^2}$ із співвідношення вхід-вихід для об'єкту управління (1.17), отримаємо

$$\begin{aligned} \delta^2 &= \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial a_1^2} = \\ &= e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \left(\frac{\partial u_1(k)}{\partial a_1^2} + \frac{\partial u_2(k)}{\partial a_1^2} \right). \end{aligned} \quad (1.20)$$

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

$$\delta^2 = (d(k+1) - y(k+1)) \frac{\partial y(k+1)}{\partial u(k)} u_2(k)(1 - u_2(k)). \quad (1.21)$$

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 (1.21), можемо записати правило настройки вагів прихованого шару таким чином:

$$\Delta w_{j1}^1 = -\eta \delta_j^1 d, \quad (1.22)$$

де

$$\delta_j^1 = \delta^2 w_{1j}^2 y_j^1 (1 - y_j^1). \quad (1.23)$$

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 (1.21) і (1.23) 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 нейроемулятора на етапі ідентифікації, як це показано на рис.1.11.

1.2.4 Пряме адаптивне нейроуправління.

Розглянемо нелінійний об'єкт, описуваний NARX моделлю

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)] \quad (1.24)$$

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

$$f_{th}(u) = \frac{e^u - e^{-u}}{e^u + e^{-u}}, \quad (1.25)$$

похідна якої обчислюється за виразом

$$f'_{th}(u) = 1 - f_{th}^2(u). \quad (1.26)$$

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

$$p = [y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)]. \quad (1.27)$$

Нейромережева модель об'єкту (НММ) має вигляд:

$$\hat{y}(k+1) = \hat{f}[y(k), y(k-1), \dots, y(k-k_y), u(k), u(k-1), \dots, u(k-k_u)], \quad (1.28)$$

де $\hat{y}(k+1)$ – вихід ШНМ

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

Оскільки алгоритм навчання неймережі 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'єк-

ту управління (1.24). Внаслідок чого метою управління може служити мінімізація розузгодження між вихідними сигналами нейромережевої і еталонної моделей. Таким чином, критерій якості управління має вигляд:

$$J = \frac{1}{2} e^2(k+1), \quad (1.29)$$

де

$$e(k+1) = d(k+1) - \hat{y}(k+1). \quad (1.30)$$

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

Використовуючи нейромережеву структуру, рівняння (1.28) можемо записати таким чином

$$\hat{y}(k+1) = w^{2T} [\tanh(\mathbf{W}^1 p + b^1)] + b^2, \quad (1.31)$$

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

w^2 – вектор вагових коефіцієнтів вихідного шаруючи;

b^1 – вектор вагових коефіцієнтів зсувів прихованого шару;

b^2 – зсув вихідного шару.

Мінімум J може бути знайдений рекурсивно за допомогою методу найшвидшого спуску

$$u(k+1) = u(k) + \eta \frac{\partial J}{\partial u(k)}, \quad (1.32)$$

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 (1.29) по $u(k)$ отримаємо

$$\frac{\partial J}{\partial u(k)} = -e(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)}, \quad (1.33)$$

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

Підставляючи (1.33) в (1.32), отримаємо

$$u(k+1) = u(k) + \eta e(k+1) \frac{\partial \hat{y}(k+1)}{\partial u(k)}. \quad (1.34)$$

На підставі відомої архітектури мережі (1.31) можна аналітично обчислити градієнт таким чином:

$$\frac{\partial \hat{y}(k+1)}{\partial u(k)} = w^{2T} [\text{sech}(\mathbf{W}^1 p + b^1)]^{-1} \mathbf{W}^1 \frac{\partial p}{\partial u(k)} \quad (1.35)$$

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

Тепер остаточно можна записати:

$$u(k+1) = u(k) + \eta e(k+1) w^{2T} [\text{sech}^2(\mathbf{W}^1 p + b^1)]^{-1} \mathbf{W}^1 \frac{\partial p}{\partial u(k)}. \quad (1.36)$$

Таким чином, алгоритм прямого адаптивного управління має вигляд:

1. Обчислюється $\hat{y}(k+1)$ відповідно до (1.31).
2. Обчислюється $e(k+1)$ відповідно до (1.30).
3. Виконується настройка вагів мережі відповідно до алгоритму зворотного розповсюдження помилки.
4. Обчислюється управляюча дія $u(k+1)$ відповідно до (1.36).
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 знадо-

бляться 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 само настройкою.

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 управління включають ряд параметрів, які необхідно вибирати або налаштовувати 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.

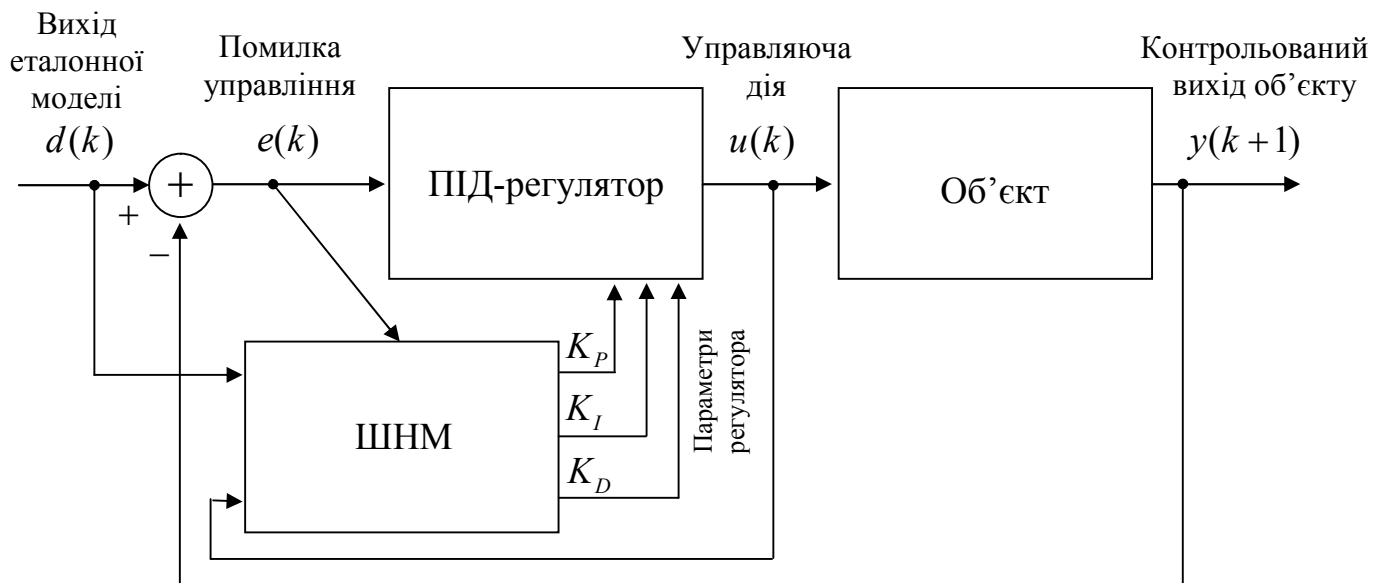


Рисунок 1.11 – Структура навчання ПІД-регулятора з само настройкою

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

$$u(k) = u(k-1) + K_p(e(k) - e(k-1)) + K_I e(k-1) + K_D(e(k) - 2e(k-1) + e(k-2)), \quad (1.37)$$

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

$$E = \frac{1}{2} e^2(k+1). \quad (1.38)$$

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

$$K_p = y_1^2, K_I = y_2^2, K_D = y_3^2. \quad (1.39)$$

Для реалізації нелінійних відображень використовуємо логістичні функції актинії (1.37) 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

$$\frac{\partial E}{\partial w_{ij}^2} = \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_i^2(k)} \frac{\partial y_i^2(k)}{\partial a_i^2(k)} \frac{\partial a_i^2(k)}{\partial w_{ij}^2(k)}. \quad (1.40)$$

Зважаючи на те, що $e(k+1) = d(k+1) - y(k+1)$ і враховуючи (1.38) можемо записати

$$\frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} = -(d(k+1) - y(k+1)) = -e(k+1). \quad (1.41)$$

Оскільки правило функціонування нейронів вихідного шару має вигляд:

$$y_i^2 = a_i^2 = \sum_{l=1}^{N^{L-1}} w_{il}^2 y_l^1 + b_{i0}^2, \quad i = 1, 2, 3; \quad (1.42)$$

$$\frac{\partial y_i^2(k)}{\partial a_i^2(k)} = 1, \quad \frac{\partial a_i^2(k)}{\partial w_{ij}^2(k)} = y_j^1, \quad i = 1, 2, 3; \quad j = 1, 2, \dots, N^1. \quad (1.43)$$

На підставі закону функціонування ПД-регулятора (1.37) і, враховуючи (1.39), запишемо рівняння для приватних похідних $\frac{\partial u(k)}{\partial y_i^2(k)}$

$$\frac{\partial u(k)}{\partial y_1^2(k)} = e(k) - e(k-1), \quad (1.44)$$

$$\frac{\partial u(k)}{\partial y_2^2(k)} = e(k-1), \quad (1.45)$$

$$\frac{\partial u(k)}{\partial y_3^2(k)} = e(k) - 2e(k-1) + e(k-2). \quad (1.46)$$

Тепер, ввівши позначення для локальної помилки вихідного шару

$$\delta_i^2 = e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_i^2(k)}. \quad (1.47)$$

можемо записати правило настройки вагів нейронів вихідного шару

$$w_{ij}^2(k+1) = w_{ij}^2(k) - \eta \delta_i^2 y_j^1, \quad i = 1, 2, 3; \quad j = 1, 2, \dots, N^1. \quad (1.48)$$

Перейшовши до прихованого шару, отримаємо

$$\begin{aligned} \frac{\partial E}{\partial w_{ij}^1} &= \sum_{n=1}^3 \frac{\partial E}{\partial e(k+1)} \frac{\partial e(k+1)}{\partial y(k+1)} \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} \frac{\partial y_n^2(k)}{\partial a_n^2(k)} \frac{\partial a_n^2(k)}{\partial y_j^1(k)} \frac{\partial y_j^1(k)}{\partial a_j^1(k)} \frac{\partial a_j^1(k)}{\partial w_{ij}^1(k)} = \\ &= \sum_{n=1}^3 e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} w_{jn}^2 y_j^1(k) (1 - y_j^1(k)) d_j(k) = -\delta_i^1 d_j(k), \end{aligned} \quad (1.49)$$

де δ_i^1 – локальна помилка прихованого шару

$$\begin{aligned} \delta_i^1 &= \sum_{n=1}^3 e(k+1) \frac{\partial y(k+1)}{\partial u(k)} \frac{\partial u(k)}{\partial y_n^2(k)} w_{jn}^2 y_j^1(k) (1 - y_j^1(k)) = \\ &= y_j^1(k) (1 - y_j^1(k)) \sum_{n=1}^3 \delta_i^1 w_{jn}^2. \end{aligned} \quad (1.50)$$

Таким чином, правило настройки вагів нейронів прихованого шару має вигляд

$$w_{ij}^1(k+1) = w_{ij}^1(k) - \eta \delta_i^1 d_j, \quad i = 1, 2, \dots, N^1; \quad j = 1, 2, \dots, N^0 \quad (1.51)$$

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 нейромодель.

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 (1.47)) і прихованого δ_i^2 шарів по формулі (1.50)).
3. Коректуються ваги вихідного шару по формулі (1.48).
4. Коректуються ваги прихованого шару по формулі (1.51).
5. Повторюються дії пп.2-6 при $k = k + 1$.

1.2.6 Нелінійне предиктивне 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).

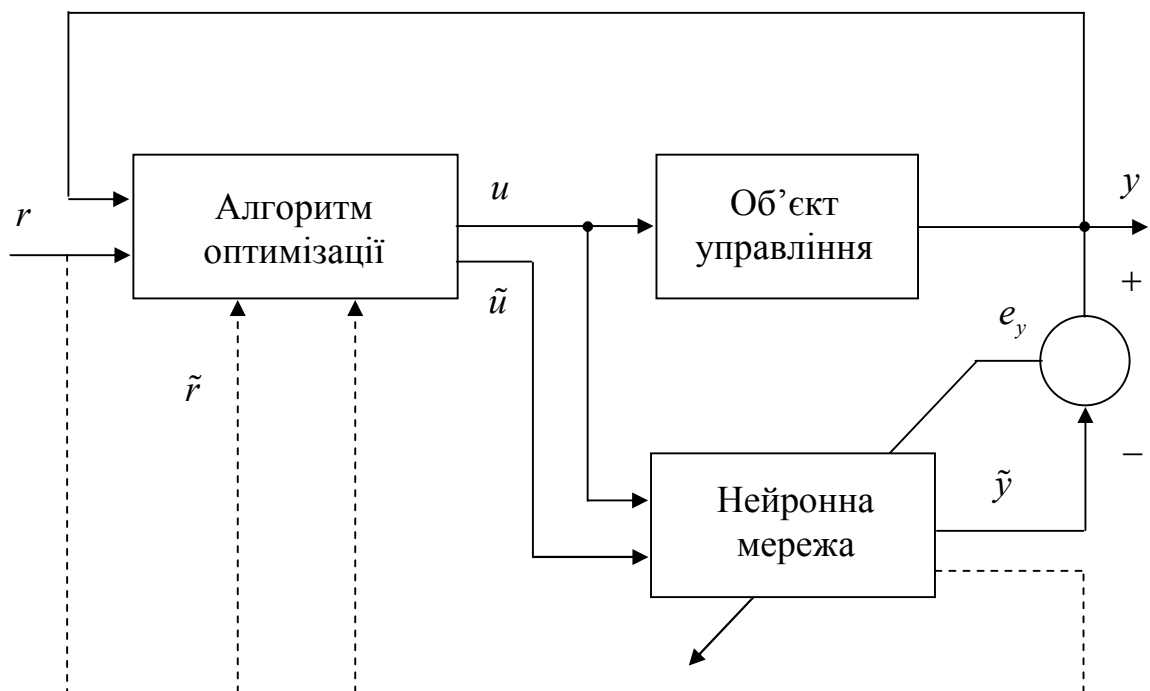


Рисунок 1.12 – Предиктивне регулювання

Регулятор в даному випадку є не що інше, як ітераційний алгоритм оптимізації [25]:

- 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 автоматизації і електроприводів шахтних підйомних машин

Вимоги до системи автоматизації. Система автоматизації шахтного підйому повинна гарантувати безперервну роботу всього обладнання комплексу та забезпечувати злагоджену взаємодію основних технологічних вузлів. У режимі автоматичного управління підйомна установка повинна функціонувати за заданою програмою без необхідності втручання обслуговуючого персоналу.

Автоматизована підйомна установка повинна працювати по шестиперіодній діаграмі швидкості.. Перший період – розгін до швидкості $1,0 \div 1,5 \text{ м/с}$, з якою скипи виходять із завантажувального і розвантажувального пристрою; другий період – розгін із заданим прискоренням до максимальної (номінальної) швидкості $v_{\max}$; третій період – рівномірний хід з $v_{\max}$; четвертий період – уповільнення від $v_{\max}$ до швидкості $v_{\text{дот}} = 0,3 \div 0,5 \text{ м/с}$; п'ятий період – дотягування з швидкістю $v_{\text{дот}} = 0,3 \div 0,5 \text{ м/с}$; шостий період – стопоріння підйомної машини.

Стопоріння машини в кінці циклу під автоматичним управлінням повинно здійснюватися робочим гальмом за сигналом датчиків кінцевого положення підйомних судин у стовбурі. Точність зупинки машини повинна забезпечувати нормальну роботу завантажувальних і розвантажувальних пристроїв.

Перехід установки в автоматичний режим управління має відбуватися тільки при досягненні кінцевих положень підйомних судин. Перехід з автоматичного управління на ручне повинен здійснюватися незалежно від положення судин і супроводжуватися включенням запобіжного гальма.

При порушенні нормального режиму роботи підйомної машини, при подача аварійних сигналів з горизонтів і з місця завантаження-розвантаження повинні здійснюватися автоматичне відключення підйомного двигуна від мережі і накладення запобіжного гальма. Після цього підйомні судини повинні перегалятися в початкове положення в режимі ручного управління або з пульта оператора розвантаження вагонеток.

Вимоги до приводу. Електропривод повинен забезпечувати:

1. Роботу в режимах руху та гальмування при зміні величини і знаку навантаження.

2. Зміни напрямку обертання двигуна з положення спокою при попередньому стопорінні машини гальмом. У деяких підйомних установках при регулюванні положення кліті на рівні завантаження-розвантаження реверс може відбуватися кілька разів під час утримування клітки на заданому рівні.

3. Регулювання швидкості в межах, що визначаються максимальною швидкістю та нормованою швидкістю ревізії.

4. Побудову розрахункової діаграми швидкості з заданою точністю, незалежно від величини і знаку навантаження, і підтримання номінальної швидкості з помилкою не більше визначеної межі. Відхилення швидкості ревізії і дотягування не повинні перевищувати $\pm 25\%$, що є достатнім для забезпечення точності стопоріння в межах ± 5 см. Цю точність можна досягти за допомогою електроприводів, що працюють у системах Г-Д і ТП-Д. При використанні приводів з більшою погрешністю регулювання швидкості слід збільшувати шлях дотягування, що призводить до зниження продуктивності установки.

5. Обмеження динамічних навантажень у компонентах підйомної установки. Рекомендується, щоб система автоматичного регулювання електроприводу постійного струму включала пристрої обмеження швидкості зміни при-

скорення (ривка), що сприятиме оптимізації динамічних режимів роботи підйомних установок.

6. Можливість місцевого (з машинного залу) і дистанційного (з верхнього приймального майданчика) управління підйомними машинами клітей та автоматичного управління шахтними підйомними установками.

7. Максимальну надійність та безпеку експлуатації підйомної установки, використання електричних машин і комплектних установок, що відповідають спеціальним технічним вимогам, а також забезпечення контролю швидкості для шахтних підйомних машин.

1.4 Нейромережевий підхід до управління. Постановка задачі дослідження

Однією з основних проблем у алгоритмічному та структурному синтезі систем управління на базі нейромереж є розумний вибір необхідної інформації для тренування багат шарових нейронних мереж, які формують оптимальні закони управління та встановлюють умови для досягнення цілей управління для класу нелінійних об'єктів. Алгоритмічний синтез охоплює вибір оптимального алгоритму для навчання нейронної мережі в реальному часі та аналіз можливості досягнення поставлених цілей управління, що має важливе практичне значення.

Для ефективного використання нейромережевого підходу необхідно, передусім, розробити математичну модель системи. Однак, математичне моделювання, яке базується на припущеннях про лінійність системи, не завжди відображає її реальні фізичні характеристики. Навіть якщо вдається створити складні математичні моделі, які точно відображають фізичні взаємозв'язки між входом і виходом системи, вони можуть виявитися непридатними для управлінських цілей. Для нелінійних систем особливо важко забезпечити низьку чутливість моделі до параметрів, що є критично важливим.

Застосування статистичних моделей дозволяє оцінювати параметри процесу через аналіз частотних характеристик за допомогою методів класичної теорії управління. Проте ефективність цього підходу може бути недостатньою для досягнення необхідної точності в управлінні. Основною причиною цього є спроба адаптувати лінійні моделі до складних нелінійних систем. Окрім того, ці методи базуються на моделях «чорного ящика», де параметри часто не мають фізичного значення. Через це їх застосування в управлінських практиках є обмеженим.

Використання адаптивних методів управління припускає наявність математичної моделі, заснованої на фізичних явищах, і методів оцінки параметрів цієї моделі. Потім будується закон управління, направлений на досягнення деякої мети, а математична модель, розглядається як реальна система. Даний підхід також ґрунтується на теорії лінійних систем. При виникненні змін в об'єкті управління або в зовнішніх умовах необхідно перебудовувати параметри моделі і задавати новий закон управління. Це вимагає втручання оператора для перевірки адекватності моделі і реальної системи.

Отже, можна зробити наступний висновок: для ефективного застосування алгоритмів управління вони повинні бути адаптивними, стійкими, нелінійними, а також легкими у реалізації та розумінні. Саме тому штучні нейронні мережі (ШНМ), які володіють усіма зазначеними властивостями, набули широкого застосування в управлінських задачах.

Управління на основі нейронних мереж базується на використанні повністю визначених ШНМ для генерації необхідних управлінських сигналів. Високий інтерес до цього напрямку в теорії управління обумовлений наступними факторами:

- ШНМ є адаптивними системами, які здатні навчатися розв'язувати складні завдання. Для ефективного навчання необхідно правильно обрати структуру мережі та забезпечити достатній об'єм навчальної інформації.
- Нейромережеві технології управління допомагають подолати багато труднощів, які виникають при роботі з нелінійними об'єктами або об'єктами невідомої струк-

тури, які не можна розв'язати за допомогою традиційних методів адаптивного управління. Здатність ШНМ реалізовувати складні нелінійні відображення забезпечується використанням сигмоїдальних функцій активації (або інших нелінійних функцій загального вигляду) у нейронах прихованих шарів. – Здібність ШНМ до самонавчання дозволяє застосовувати нейрорегулятори навіть за значних невизначеностей, тоді як для традиційних методів адаптивного управління необхідно мати великий обсяг апріорної інформації про об'єкт управління. – Висока швидкодія та надійність нейрорегуляторів забезпечуються високим ступенем паралельності ШНМ. Простота реалізації нейронних мереж та їх здатність до навчання роблять їх особливо привабливими для управління складними нелінійними об'єктами в реальному часі.

Аналіз наукової літератури показав, що існує багато підходів до проблеми синтезу систем управління, однак універсальний ідеальний регулятор досі не створений. При розробці систем управління для механізму підйому шахтної підйомної установки, окрім зазначених вимог, необхідно враховувати пружні властивості підйомного канату. Як буде показано в розділі 3, кінцева жорсткість підйомного канату впливає на коливальний характер перехідних процесів у системі, що негативно позначається на роботі механізму. Традиційні регулятори не можуть забезпечити належні показники якості функціонування системи.

Метою роботи є розробка нейромережевої системи управління механізмом підйому шахтної підйомної установки з урахуванням пружних властивостей підйомного канату.

Для досягнення поставленої мети необхідно вирішити наступні завдання.

1. Виконати вибір системи електроприводу механізмом підйому шахтної підйомної установки, 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 кінцевої жорсткості підйимального канату.

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

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

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

ВИСНОВКИ ДО РОЗДІЛУ 1

1. Аналіз досліджень щодо застосування методів нейронних мереж для управління електромеханічними системами показав, що використання штучних нейронних мереж дозволяє створювати нейромережеві моделі систем управління, які не тільки формують внутрішні уявлення про зовнішні процеси та прогнозують поведінку об'єкта управління, але й здатні генерувати оптимальні управляючі впливи відповідно до еталонної моделі. Зокрема, значну увагу приділяють покращенню методів навчання нейронних мереж, ідентифікації нелінійних нестационарних об'єктів за допомогою нейронних мереж та створенню нейромережевих систем управління для нелінійних об'єктів.

2. Завдяки здатності штучних нейронних мереж до самонавчання, нейрорегулятори можна використовувати навіть у випадках значних невизначеностей, тоді як для традиційних методів адаптивного управління необхідна велика кількість попередньої інформації про об'єкт управління. Простота реалізації нейронних мереж і їх здатність до навчання роблять їх особливо ефективними при управлінні складними нелінійними об'єктами в реальному часі.

3. Досліджено ефективність використання мережі типу багатошаровий персептрон для створення нейромережевої моделі об'єкта управління. Багатошаровий персептрон є універсальним апроксиматором і класифікатором, що в сучасній практиці є однією з найзатребуваніших нейронних мереж. Розглянуті аспекти навчання нейронних мереж і вибрано метод навчання нейронної мережі для регулятора.

4. Проведений аналіз існуючих методів опису нелінійних динамічних об'єктів і розглянуті принципи побудови нейромережевих NARMAX і NARX моделей. NARX мережеві моделі параметризуються лінійно і тому використовують стандартні методи тренування мережі. Внаслідок цього саме даний вид моделі найчастіше використовується при ~~нейронній~~ ідентифікації систем.

5. Для оптимізації роботи об'єкта в необхідна корекція параметрів регуляторів відповідно до зміни робочих умов. У зв'язку з цим доцільною є розроб-

ка систем керування на основі адаптивного підходу в поєднанні з методами теорії ШНМ, що дозволить, з одного боку, побудувати досить прості нейромережеві моделі, а з іншого – адаптивно корегувати параметри як цих моделей, так і регулятора відповідно до умов, що змінюються. В результаті аналізу структур систем нейрорегулювання, актуальних в даний час, вибраний принцип нелінійного предиктивного регулювання при побудові нейромережевого регулятора системи.

6. Виконаний аналіз основних вимог, що пред'являються до систем автоматизації і електроприводів шахтних підйомних машин. Показано, що внаслідок кінцевої жорсткості підйомного канату загальнозживані регулятори не в змозі забезпечити задовільні показники якості функціонування системи, тому недоліки традиційного підходу до синтезу таких систем в значній мірі можуть бути подолані за рахунок застосування нейромережевих технологій.

7. Сформульовано задачу дослідження, що складається в розробці нейромережевої системи управління механізмом підйому шахтної підйомальної установки з урахуванням пружних властивостей підйомального канату.

РОЗДІЛ 2

РОЗРАХУНОК СИСТЕМИ АВТОМАТИЧНОГО УПРАВЛІННЯ МЕХАНІЗМОМ ПІДЙОМУ ШАХТНОЇ ПІДЙІМАЛЬНОЇ УСТАНОВКИ

2.1 Технічні характеристики шахтної підйимальної установки. Вимоги до системи управління

Вихідні дані для розрахунку електроприводу механізму підйому шахтної-підйимальної установки і синтезу системи управління наведені в табл. 2.1.

Таблиця 2.1 – Технічні характеристики копальневої підйомної установки

Найменування величини	Розмірність	Значення
Корисний вантаж m_b	кг	$11,5 \cdot 10^3$
Висота підйому, H	м	711
Маса порожнього скипа $m_{ск}$	кг	$8,2 \cdot 10^3$
Діаметр барабана $D_б$	м	5
Максимальна швидкість підйому V_{max}	м/с	6,7
Число підйомів в годину n		26
Шлях розвантаження h_p	м	2,5
Уповільнення в кінці шляху b_p	м/с ²	0,5
Швидкість входу в розвантажувальну криву V_p	м/с	0,5
Нормальне прискорення a	м/с ²	0,65
Нормальне уповільнення b	м/с ²	0,65
Швидкість скипа при закритті затвора $V_з$	м/с	0,5
Прискорення при закритті затвора $a_з$	м/с ²	0,167
Маса погонного метра каната m_{lk}	кг/м	9,39

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 5 с.

2.2 Вибір системи електроприводу механізму підйому шахтної підйомної установки

На сьогоднішній день не існує єдиного рішення щодо вибору системи електроприводу для кожного типу підйомної машини. Зменшити кількість можливих варіантів електроприводів можна, використовуючи критерії, такі як відносна вартість приводів, необхідна потужність і максимальний пусковий момент, а також рівень керованості, що залежить від призначення та технологічних умов роботи підйомної установки. При техніко-економічному порівнянні приводів важливим показником є капітальні витрати на електрообладнання, що включає вартість самого обладнання, будівельні витрати та інші супутні капітальні вкладення.

Потужність обмежує область застосування асинхронного електроприводу. Існуюча апаратура управління пуском двигуна (роторні контакторні станції управління) дозволяє застосувати асинхронний привід при потужності не вище 1250 *кВт* при однодвигунному приводі і 2500 *кВт* при двохдвигунному.

При потужності до 2500 *кВт* для скіпових і підйомів клітей руднічних шахт можливий вибір між приводом змінного і постійного струму. При потужності понад 2500 *кВт* повинен розглядатися тільки один з варіантів приводу постійного струму по системі ТП-Д або Г-Д.

На вибір системи приводу впливають умови роботи підйомної установки. На малозавантажених підйомах клітей і допоміжних стовбурах виявляються переваги асинхронного приводу за експлуатаційними показниками, перш за все у вигляді меншої витрати електроенергії. Для підйомних машин клітей головних стовбурів при роботі з декількох горизонтів і необхідності тривалої роботи на зниженій швидкості перевага віддається приводу постійного струму; з його допомогою можна без додаткових втрат регулювати швидкість в необхідних межах.

При великих швидкостях підйому (понад 8-10 м/с) небажано застосування асинхронного приводу з роторним реостатом по безпеці із-за його низької

керованості. При цьому слід віддавати перевагу більш здійсненій система приводу. При виборі між однодвигунним і двохдвигунним приводом переважний однодвигунний привід, що має простішу схему управління і можливість установки на фундаменті резервного двигуна. При виборі між двома варіантами приводу редуктора, наприклад: $u_p = 20$, $n_d = 600 \text{ об/хв}$ або $u_p = 10,54$, $n_d = 300 \text{ об/хв}$, перевага віддається приводу з одноступінчатим двохприводним редуктором, що має в порівнянні з двухступінчатим редуктором більший ККД, менше число деталей (підшипників, шестерень), що зношуються, порівняно менший шум в роботі, можливість 100%-го резервування при установці резервного двигуна або можливість підвищення потужності приводу установкою другого робочого двигуна. Двуступінчаті редуктори ЦД-20, $u_p = 20$ рекомендуються при малих швидкостях підйому.

Область застосування однодвигунного приводу редуктора обмежується допустимим крутильним моментом на тихохідному валу редуктора.

При виборі між редуктором і безредукторним приводом постійного струму враховується деяке збільшення капітальних витрат при безредукторному приводі і значні переваги за експлуатаційними показниками. Це дозволяє рекомендувати безредукторний привід для більшості установок при потужності вище 1000 кВт .

В більшості випадків підвищені капітальні витрати при приводі постійного струму не можуть бути компенсовані в нормовані терміни зменшенням витрати електроенергії. Тому важливого значення набуває якість управління, забезпечення необхідних меж регулювання, умов автоматизації, зниженні навантажень на механічні вузли машини. З урахуванням даних чинників, а також враховуючи можливість вдосконалення і зниження вартості приводу за системою ТП-Д, даному виду приводу слід віддавати перевагу для скіпових установок головних стовбурів.

У різниці експлуатаційних витрат по приводах ТП-Д і Г-Д найбільш істотним пунктом є оплата за електроенергію, менш істотним – витрати на обслуговування електроустаткування і його ремонт.

Якщо для приводів по системах ТП-Д і Г-Д економічні показники виявляються порівнянними, то найбільш переважний привід за системою ТП-Д, оскільки він має переваги, що підвищують потенційну надійність унаслідок застосування статичного перетворювача.

Привід системи Г-Д може бути економічно вигіднішим при модернізації існуючих підйомних установок, де зміна потужності підстанції або установка фільтрокомпенсуючих пристроїв є неможливими через обмеженість площі промислового майданчика. Використання приводу Г-Д замість ТП-Д може бути доцільним у таких випадках:

- потужність шахтної підстанції є відносно невеликою і не здатна забезпечити пускові навантаження для ТП;
- при запуску приводу ТП-Д підйомних машин виникають неприпустимі падіння і коливання напруги через створення значної реактивної потужності;
- приводи підйомних машин розташовані далеко від живлячих і розподільних підстанцій, де лінії мають значну протяжність і опір, що призводить до недопустимих падінь і коливань напруги при використанні ТП;
- до живлячої підстанції підключено багато приводів, що спричиняє погіршення гармонійного складу напруги в електромережі до неприйняттого рівня.
- при необхідності застосування додаткового устаткування для задовільної сумісності приводів ТП-Д з електромережею і поліпшення їх енергетичних показників.

На основі техніко-економічного аналізу вибираємо безредукторний електропривод постійного струму за системою Г-Д.

2.3 Вибір електродвигуна по потужності

Попередній вибір електродвигуна шахтної підйомної установки виконується по розрахунковій потужності $P_{\text{дв розр}}$ і швидкості обертання $\omega_{\text{д}}$. Потім виконується перевірка двигуна за умовами нагріву і перевантажувальній спроможності.

Розрахункова потужність електродвигуна визначається по формулі:

$$P_{\text{дв розр}} \geq \frac{\xi \cdot k_{\text{ш}} \cdot G_{\text{в}} \cdot V_{\text{max}}}{\eta} \cdot 10^{-3},$$

де $\xi = 1,15 - 1,35$ – коефіцієнт ефективного зусилля підйому;

$k_{\text{ш}} = 1,12$ – коефіцієнт шахтних опорів для навантаженої гілки;

$G_{\text{в}} = m_{\text{в}} \cdot g$ – сила, необхідна для підйому корисного вантажу, H ;

V_{max} – максимальна швидкість підйому, m/c

$\eta = 0,97$ – ККД механізму

$$P_{\text{дв розр}} \geq \frac{1,35 \cdot 1,14 \cdot 11,5 \cdot 10^3 \cdot 9,8 \cdot 6,7}{0,85} \cdot 10^{-3} = 1272 \text{ кВт}$$

Номінальна швидкість обертання двигуна:

$$\omega_{\text{д}} = \frac{2V_{\text{max}}}{D_{\text{б}}},$$

де $D_{\text{б}}$ – діаметр барабана, m

$$\omega_{\text{д}} = \frac{2 \cdot 6,7}{5} = 2,7 \text{ c}^{-1}.$$

По каталогу вибираємо двигун П2-26/41-2,25УХЛ4. Двигуни серії П2 призначені для живлення від тиристорних перетворювачів. В таких двигунах ізоляція обмоток силового кола від корпусу розрахована на напругу 1200-1500

B , оскільки ці обмотки знаходяться під впливом амплітудного значення лінійної напруги вентиляційної обмотки силового живлячого трансформатора. Ізоляція обмоток збудження виконується на напругу 600-1000 B , оскільки живлення здійснюється від тиристорного збудника напругою 460 B для забезпечення форсування напруги збудження двигуна в приводах з реверсом поля. Щоб уникнути підвищених втрат в двигуні і звуженні безіскрової зони комутації за наявності змінній складової струму якоря, ефективне значення її обмежується 2% номінального струму двигуна при номінальній частоті обертання. Станина двигунів серії П2 вибирається шихтованою для зменшення втрат і збільшення швидкодії наростання струму якоря і струму збудження в приводах з реверсом поля. Двигуни виконуються з перевантаженням по струму якоря 200%, і відключаючому струму 225-250% з примусовою вентиляцією по замкнутому і по розімкненому циклу.

Технічні дані двигуна: номінальна потужність $P_n = 1800 \text{ кВт}$, номінальна частота обертання $n_n = 40 \text{ об/хв}$, номінальна напруга $U_n = 750 \text{ В}$, номінальний струм $I_n = 2625 \text{ А}$, опір обмотки якоря $R_{\text{я}} = 0,0081 \text{ Ом}$, опір обмотки додаткових полюсів $R_{\text{дп}} = 0,0013 \text{ Ом}$, опір компенсаційної обмотки $R_{\text{ко}} = 0,00041 \text{ Ом}$, число пар полюсів $p = 5$.

2.4 Побудова швидкісної діаграми шахтного підйомника

Для перевірки правильності попередньо вибраного двигуна проводимо розрахунок і побудову швидкісної і навантажувальної діаграм електроприводу. Робочий цикл підйомника складається з періодів розгону, рівномірного руху і уповільнення. Скипові підйомники мають шестиперіодну швидкісну діаграму. Побудову швидкісної діаграми слід проводити для однієї активної ділянки підйому (підйому корисного вантажу).

Розрахунок ділянок швидкісної діаграми виконуємо в наступній послідовності.

Ділянка I – період закривання затвора.

$$t_1 = \frac{V_3}{a_3},$$

$$h_1 = \frac{V_3}{2} \cdot t_1,$$

де t_1 , h_1 – час роботи, c , і шлях, m , що проходить скіп на 1 ділянці роботи

$$t_1 = \frac{1,5}{0,167} = 9 \text{ c};$$

$$h_1 = \frac{1,5}{2} \cdot 9 = 6,8 \text{ м}.$$

Ділянка II – період нормального прискорення:

$$t_2 = \frac{V_{\max} - V_3}{a};$$

$$h_2 = \frac{V_{\max} + V_3}{2} \cdot t_2,$$

$$t_2 = \frac{6,7 - 1,5}{0,65} = 8 \text{ c};$$

$$h_2 = \frac{6,7 + 1,5}{2} \cdot 8 = 33 \text{ м}.$$

Ділянка IV – період нормального уповільнення:

$$t_4 = \frac{V_{\max} - V_p}{b};$$

$$h_4 = \frac{V_{\max} + V_p}{2} \cdot t_4,$$

$$t_4 = \frac{8 - 0,5}{0,8} = 9,4 \text{ с} ,$$

$$h_4 = \frac{8 + 0,5}{2} \cdot 9,4 = 40 \text{ м} .$$

Ділянка VI – період стопоріння скипа:

$$t_6 = \frac{V_p}{b_p} ;$$

$$h_6 = \frac{V_p}{2} \cdot t_6 ,$$

$$t_6 = \frac{0,5}{0,5} = 1 \text{ с} ;$$

$$h_6 = \frac{0,5}{2} \cdot 1 = 0,25 \text{ м} .$$

Ділянка V – період руху скипа в розвантажувальних кривих;

$$t_5 = \frac{h_5}{V_p} ;$$

$$h_5 = h_p - h_6 ,$$

$$h_5 = 2,75 - 0,25 = 2,5 \text{ м} ;$$

$$t_5 = \frac{2,5}{0,5} = 5 \text{ с}$$

Ділянка III – період руху скипа з максимальною сталою швидкістю:

$$t_3 = \frac{h_3}{V_{\max}} ;$$

$$h_3 = H - (h_1 + h_2 + h_4 + h_p),$$

$$h_3 = 711 - (6,8 + 33 + 40 + 2,75) = 628,5 \text{ м},$$

$$t_3 = \frac{628,5}{6,7} = 94 \text{ с}.$$

Тривалість роботи підйомної машини (повний час підйому), с

$$T = \sum_1^n t_i = \sum_1^6 (t_1 + t_2 + t_3 + t_4 + t_5 + t_6) ;$$

$$T = 9 + 8 + 94 + 9,4 + 5 + 1 = 126,4 \text{ с}.$$

Час паузи

$$\theta = T_{\text{ц}} - T ,$$

де $T_{\text{ц}}$ – час циклу підйому, с

$$T_{\text{ц}} = \frac{3600}{n},$$

n – число підйомів в годину

$$T_{\text{ц}} = \frac{3600}{26} = 139 \text{ с},$$

$$\theta = 139 - 126,4 = 12,6 \text{ с}.$$

Швидкісна діаграма шахтного підйомника приведена на рис.2.1, а.

2.5 Побудова діаграми навантаження

Діаграма навантаження шахтного підйомника будується як залежність підйомного зусилля на ободі барабана F_6 від часу. Побудова діаграми навантаження ведеться спрощеним методом в припущенні постійності динамічного моменту. На підставі побудованої діаграми навантаження перевіряється відповідність заздалегідь вибраного двигуна заданому режиму роботи виконавчого механізму за умовами нагріву, перевантажувальної здатності і заданої продуктивності.

Величина зусилля знаходиться по формулі:

$$F_6 = [(k_{\text{ш}} - \beta_{\kappa})m_{\text{в}} + (m_{\text{лк}} - q)(H - 2x)] \pm ma,$$

де β_{κ} – коефіцієнт, що враховує зменшення навантаження при русі скипа в розвантажувальних кривих, $\beta_{\kappa} = 0,6$;

$m_{\text{лк}}$ – маса погонного метра робочого каната, кг/м ;

q – маса погонного метра хвостового каната, кг/м ; $q = 0$;

x – шлях, пройдений скипом від початку підйому, м ;

m – сумарна приведена маса підйомної установки до обода барабана, кг ;

a, b – величини прискорень (уповільнень) на ділянках, м/с^2 .

Для розрахунку m' по кривим рис. 1.2 [29] визначимо відносні приведені маси: завантаженого скипа μ_2 , каната μ_{κ} і двигуна з підйомним барабаном μ_1

$$\mu_1 = 0,73; \quad \mu_{\kappa} = 0,095; \quad \mu_2 = 0,175.$$

Сумарна приведена маса

$$m = \frac{m_{\text{в}} + m_{\text{ск}} + m_{\text{лк}} \cdot H}{\mu_{\kappa} + \mu_2},$$

$$m = \frac{11500 + 8200 + 9,39 \cdot 711}{0,095 + 0,175} = 56 \cdot 10^3 \text{ кг}.$$

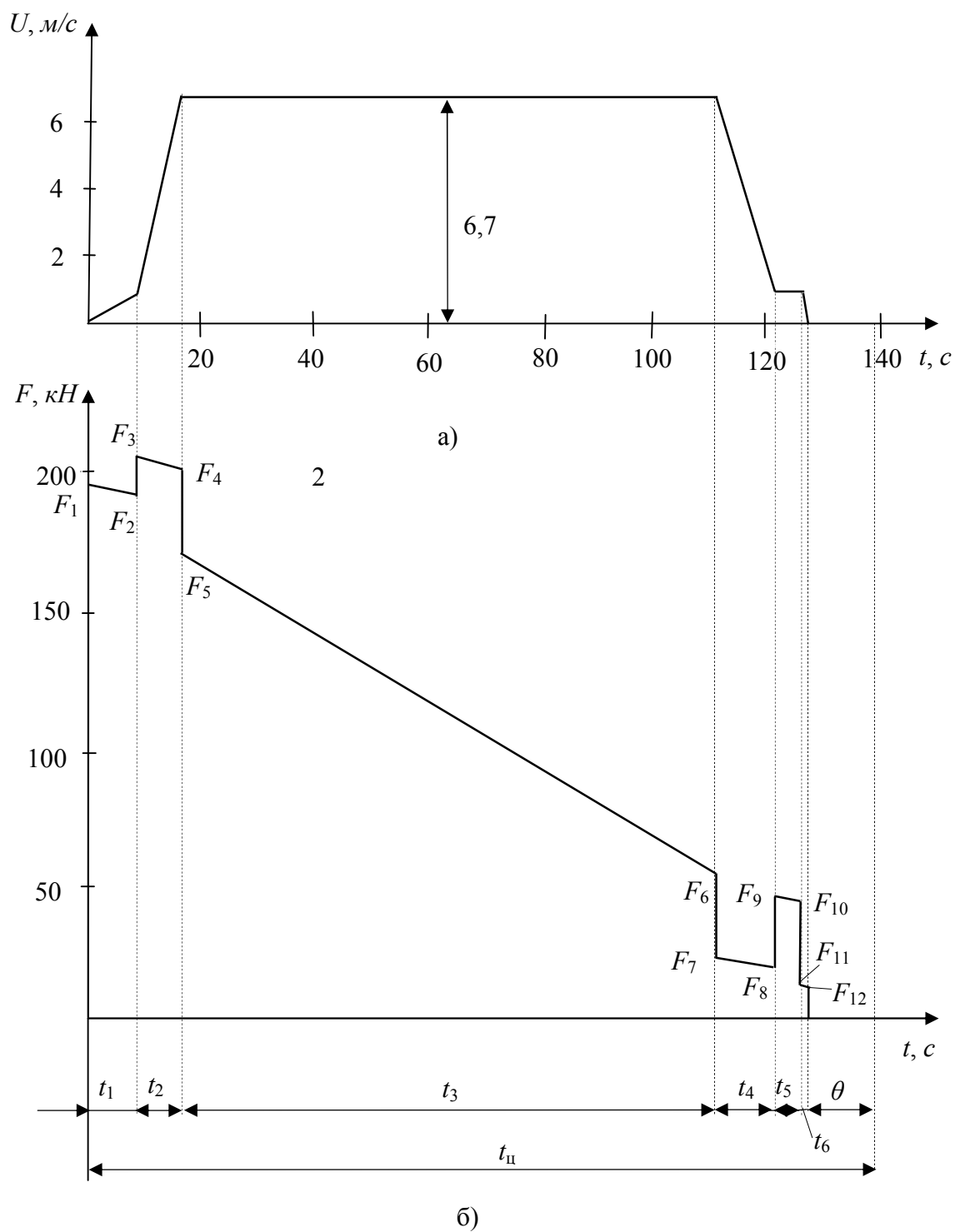


Рисунок 2.1 – Швидкісна діаграма (а) і діаграма навантаження (б)
шахтної підйомальної установки

$$1. \quad x_1 = 0 \quad a = a_3 = 0,167; \quad b = 0$$

$$F_{61} = (1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot 711) \cdot 9,81 + 56 \cdot 10^3 \cdot 0,167 = 192 \cdot 10^3 \text{ Hм.}$$

$$2. \quad x_2 = h_1 = 6,8 \text{ м} \quad a = a_3 = 0,167; \quad b = 0$$

$$F_{62} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 6,8)] \cdot 9,81 + 56 \cdot 10^3 \cdot 0,167 = 189 \cdot 10^3 \text{ H.}$$

$$3. \quad x_3 = h_1 = 6,8 \text{ м} \quad a = 0,65 \quad b = 0$$

$$F_{63} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 6,73)] \cdot 9,81 + 56 \cdot 10^3 \cdot 0,65 = 217 \cdot 10^3 \text{ H.}$$

$$4. \quad x_4 = h_1 + h_2 = 6,8 + 33 = 39,8 \text{ м} \quad a = 0,65; \quad b = 0$$

$$F_{64} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 39,8)] \cdot 9,81 + 56 \cdot 10^3 \cdot 0,65 = 207,8 \cdot 10^3 \text{ H}$$

$$5. \quad x_5 = h_1 + h_2 = 39,8 \text{ м} \quad a = 0 \quad b = 0$$

$$F_{65} = [1,09 \cdot 11,5 \cdot 10^3 + 8,39 \cdot (711 - 2 \cdot 39,8)] \cdot 9,81 = 173,7 \cdot 10^3 \text{ H.}$$

$$6. \quad x_6 = h_1 + h_2 + h_3 = 6,8 + 33 + 628,5 = 668,3 \text{ м} \quad a = 0; \quad b = 0$$

$$F_{66} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 668,3)] \cdot 9,81 = 57 \cdot 10^3 \text{ H.}$$

$$7. \quad x_7 = h_1 + h_2 + h_3 = 668,3 \text{ м} \quad a = b = 0,65; \quad b = 0$$

$$F_{67} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 668,3)] \cdot 9,81 - 56 \cdot 10^3 \cdot 0,67 = 22,6 \cdot 10^3 \text{ H}$$

$$8. \quad x_8 = x_7 + h_4 = 668,3 + 40 = 708,3 \text{ м}; \quad a = 0 \quad b = 0,65$$

$$F_{68} = [1,09 \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 708,3)] \cdot 9,81 - 56 \cdot 10^3 \cdot 0,65 = 18,7 \cdot 10^3 \text{ H}$$

$$9. \quad x_8 = x_8 = 708,3 \text{ м} \quad a = 0 \quad b = 0,3$$

$$F_{69} = [(1,09 - 0,3) \cdot 10 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 708,3)] \cdot 9,81 = 47,2 \cdot 10^3 \text{ H.}$$

$$10. \quad x_{10} = x_8 + h_5 = 707, +2,55 = 709,5 \text{ м} \quad a = 0 \quad b = 0,3$$

$$F_{610} = [(1,09 - 0,3) \cdot 11,5 \cdot 10^3 + 9,39(711 - 2 \cdot 709,5)] \cdot 9,81 = 45,9 \cdot 10^3 \text{ Н}$$

$$11. \quad x_{11} = x_{10} \quad a = b_p = 0,5 \quad b = 0$$

$$F_{611} = [(1,09 - 0,3) \cdot 11,5 \cdot 10^3 + 9,39 \cdot (711 - 2 \cdot 709,5)] \cdot 9,81 - \\ - 56 \cdot 10^3 \cdot 0,5 = 19,2 \cdot 10^3 \text{ Н}$$

$$12. \quad x_{12} = H; \quad a = b_p = 0,5 \quad b = 0$$

$$F_{612} = [(1,09 - 0,3) \cdot 10 \cdot 10^3 + 9,39(711 - 2 \cdot 711)] \cdot 9,81 - 55 \cdot 10^3 \cdot 0,5 = 18,6 \cdot 10^3 \text{ Н}$$

Навантажувальна діаграма приведена на рис. 2.1,б.

2.6 Перевірка двигуна по нагріву і перевантажувальній здатності

Для того, щоб двигун задовольняв умовам нагріву, повинно виконуватись

$$P_{\text{еф}} \leq P_{\text{н}},$$

де $P_{\text{н}}$ – номінальна потужності заздалегідь вибраного двигуна;

$P_{\text{еф}}$ – ефективна потужність

$$P_{\text{еф}} = \frac{F_{\text{еф}} \cdot V_{\text{max}}}{\eta};$$

$F_{\text{еф}}$ – середньоквадратичне зусилля

$$F_{\text{еф}} = \sqrt{\frac{\sum_{i=1}^n F_i^2 t_i}{\alpha \sum_{j=1}^k t_j + \sum_{i=1}^m t_i + \beta \theta}}, \quad (2.1)$$

де t_i – тривалість роботи на ділянці, де $\omega = \omega_n$ (третя ділянка);

t_j – тривалість роботи на ділянках, де $\omega < \omega_n$

β , α – коефіцієнти, що залежать від системи вентиляції двигунів; для двигунів з незалежною вентиляцією $\beta = 1$, $\alpha = 1$.

Підставляючи відповідні значення в (2.1), отримаємо

$$F_{\text{еф}} = 158,8 \cdot 10^3 \text{ Н},$$

$$P_{\text{еф}} = \frac{158,8 \cdot 10^3 \cdot 6,7}{0,85} = 1252 \cdot 10^3 \text{ Вт}.$$

Умова $P_{\text{еф}} \leq P_n$ виконується, отже за умовами нагріву двигун відповідає умовам експлуатації.

Перевіряємо двигун по перевантажувальній здатності. Дана перевірка проводиться для того, щоб переконатися, що вибраний електропривод зможе забезпечити задані величини прискорення.

По перевантажувальній здатності двигун задовольняє вимогам, якщо виконується умова

$$\lambda_{\phi} < \lambda,$$

λ_{ϕ} – фактичне перевантаження двигуна

$$\lambda_{\phi} = \frac{F_{\text{max}}}{F_n},$$

F_{max} – максимальне зусилля, яке визначається по діаграмі навантаження:

$$F_{\text{max}} = F_{63} = 217 \cdot 10^3 \text{ Н}$$

F_n – номінальне зусилля двигуна;

$$F_n = \frac{P_n \eta}{V_{\text{max}}},$$

$$F_{\text{н}} = \frac{1800 \cdot 10^3 \cdot 0,85}{6,7} = 163 \cdot 10^3 \text{ Н},$$

$$\lambda_{\phi} = \frac{217}{163} = 1,33.$$

де λ – кратність перевантаження по моменту двигуна

$$\lambda = \frac{M_{\text{max}}}{M_{\text{н}}}.$$

Для двигунів постійного струму $\lambda = 2 \div 2,5$;

Умова $\lambda_{\phi} < \lambda$ виконується, отже по перевантажувальній здатності двигун задовольняє вимогам.

2.7 Вибір основного електроустаткування силового кола

Відповідно до [41], потужність генератора вибирається виходячи із співвідношення

$$P_{\text{г}} \geq k_3 P_{\text{нд}},$$

де $k_3 = 1,1 \div 1,3$ – коефіцієнт запасу

$P_{\text{нд}}$ – номінальна потужність двигуна

$$P_{\text{г}} \geq 1,1 \cdot 1800 = 1980 \text{ кВт}.$$

По каталогу вибираємо генератор П21-35-17К. Технічні дані генератора: номінальна потужність $P_{\text{н}} = 3000 \text{ кВт}$; номінальна напруга $U_{\text{н}} = 750 \text{ В}$; номінальний струм $I_{\text{н}} = 4000 \text{ А}$.

По таблиці 4.2 [28] вибираємо приводний синхронний двигун СДНЗ-17-59-12У34. Потужність 3200/3710 кВт/кВА, частота обертання $n_{\text{н}} = 500 \text{ об/хв}$; номінальна напруга $U_{\text{н}} = 750 \text{ В}$, номінальний струм $I_{\text{н}} = 4000 \text{ А}$, опір обмотки

якоря $R_{яГ} = 0,00341 \text{ Ом}$, опір обмотки додаткових полюсів $R_{дпГ} = 0,00043 \text{ Ом}$, опір компенсаційної обмотки, $R_{коГ} = 0,00187 \text{ Ом}$, число пар полюсів $p = 4$.

Вибираємо тиристорні збудники електричних машин. У приводі Г-Д підйомних машин застосовуються: реверсивний тиристорний пристрій КТУЕР1-63/430Р для збудження генератора; нереверсивний тиристорний пристрій КТУП2 – 250/460Р, КТУЕ-63/460Р для збудження двигуна.

Пристрої забезпечують перетворення трифазної змінної напруги 380 В в регульовану напругу постійного струму.

2.8 Розрахунок параметрів силового кола електроприводу

Визначаємо параметри силового кола, необхідні для подальших розрахунків.

Сумарна індуктивність силового кола

$$L_{\Sigma} = L_{д} + L_{Г},$$

де $L_{д}$, $L_{Г}$ – індуктивність якірного кола двигуна і генератора відповідно

$$L_{д(Г)} = \gamma_{д(Г)} \frac{U_{н д(Г)}}{I_{н д(Г)} \omega_{н д(Г)} p_{д(Г)}},$$

$\gamma_{д(Г)}$ – коефіцієнт, рівний $0,1 \div 0,25$ для компенсованих машин.

$U_{н д(Г)}$, $I_{н д(Г)}$, $\omega_{н д(Г)}$, $p_{д(Г)}$ – номінальні напруга, струм, швидкість обертання і число пар полюсів двигуна і генератора відповідно.

$$\omega_{н д} = \frac{\pi n_{н д}}{30},$$

$$\omega_{н д} = \frac{3,14 \cdot 40}{30} = 4,2 \text{ с}^{-1};$$

$$\omega_{\text{нГ}} = \frac{\pi n_{\text{нГ}}}{30},$$

$$\omega_{\text{нГ}} = \frac{3,14 \cdot 500}{30} = 52,3 \text{ с}^{-1},$$

$$L_{\text{д}} = 0,1 \cdot \frac{750}{2625 \cdot 4,19 \cdot 8} = 0,8523 \cdot 10^{-3} \text{ Гн},$$

$$L_{\text{Г}} = 0,1 \cdot \frac{750}{2870 \cdot 52,3 \cdot 8} = 0,0619 \cdot 10^{-3} \text{ Гн},$$

$$L_{\Sigma} = 0,8523 \cdot 10^{-3} + 0,0619 \cdot 10^{-3} = 0,9142 \cdot 10^{-3} \text{ Гн}.$$

Сумарний активний опір якірного кола електроприводу:

$$R_{\Sigma} = R_{\text{як д}} + R_{\text{як Г}},$$

де $R_{\text{як д}}$, $R_{\text{як Г}}$ – сумарні опори якірного кола двигуна і генератора відповідно

$$R_{\text{як д}} = \alpha_t (R_{\text{я д}} + R_{\text{дп д}} + R_{\text{ко д}});$$

$$R_{\text{як Г}} = \alpha_t (R_{\text{я Г}} + R_{\text{дп Г}} + R_{\text{ко Г}});$$

α_t – температурний коефіцієнт, що враховує зміну опору при нагріві машини; для машин з класом ізоляції Н значення $\alpha_t = 1,36$.

$R_{\text{я д(Г)}}$, $R_{\text{дп д(Г)}}$, $R_{\text{ко д(Г)}}$ – активні опори обмоток якоря, додаткових потоків і компенсаційної обмотки відповідно двигуна і генератора.

Згідно з паспортними даними:

двигуна – $R_{\text{я д}} = 0,0081 \text{ Ом}$, $R_{\text{дп д}} = 0,0013 \text{ Ом}$, $R_{\text{ко д}} = 0,0041 \text{ Ом}$

генератора – $R_{\text{я Г}} = 0,0032 \text{ Ом}$, $R_{\text{дп Г}} = 0,00039 \text{ Ом}$, $R_{\text{ко Г}} = 0,0016 \text{ Ом}$.

$$R_{\text{як д}} = 1,36 \cdot (0,0081 + 0,0013 + 0,0041) = 0,01836 \text{ Ом}$$

$$R_{\text{як Г}} = 1,36 \cdot (0,0032 + 0,00039 + 0,0016) = 0,00701 \text{ Ом}$$

$$R_{\Sigma} = 0,01836 + 0,00701 = 0,0254 \text{ Ом}$$

Постійна проти-ЕДС двигуна

$$c = k\Phi_{\text{н}} = \frac{U_{\text{н}} - I_{\text{н}} R_{\text{як д}}}{\omega_{\text{н}}},$$

де k – конструктивна постійна двигуна;

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

$$c = k\Phi_{\text{н}} = \frac{750 - 2625 \cdot 0,0135}{4,19} = 170.$$

Коефіцієнт посилення двигуна

$$k_{\text{д}} = \frac{1}{c},$$

$$k_{\text{д}} = \frac{1}{170} = 0,0059.$$

Коефіцієнт посилення генератора

$$k_{\text{Г}} = \frac{E_{\text{н Г}}}{U_{\text{н зб Г}}} = \frac{U_{\text{н Г}} + I_{\text{н Г}} R_{\Sigma \text{Г}}}{U_{\text{н зб Г}}},$$

де $U_{\text{н Г}}$, $U_{\text{н зб Г}}$ – номінальна напруга на якорі і напруга збудження генератора;

для генератора ГП 21/22-315 $U_{\text{н зб Г}} = 220 \text{ В}$;

$I_{\text{н Г}}$ – номінальний струм генератора;

$E_{\text{н Г}}$ – номінальна ЕДС генератора

$$k_{\text{Г}} = \frac{750 + 2870 \cdot 0,0052}{220} = 3,34.$$

Коефіцієнт посилення тиристорного збудника генератора визначається для синусоїдальної напруги порівняння і амплітудного значення напруги порівняння $U_{\Pi} = 10 \text{ В}$

$$k_{\text{тп}} = \frac{U_{\text{нзбг}}}{U_{\Pi}},$$

$$k_{\text{тп}} = \frac{220}{10} = 22.$$

Електромагнітна постійна якірного ланцюгу

$$T_{\text{е}} = \frac{L_{\Sigma}}{R_{\Sigma}},$$

$$T_{\text{е}} = \frac{0,9142 \cdot 10^{-3}}{0,0254} = 0,036 \text{ с}.$$

Електромеханічна постійна часу електроприводу

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

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

$$J_{\Sigma} = J_{\text{в}} + J_{\text{к}} + J_{\Sigma\text{д}},$$

$J_{\text{в}}$ – приведений до валу двигуна момент інерції скипа з вантажем

$$J_{\text{в}} = (Q_{\Pi} + m_{\text{ск}}) \rho^2,$$

ρ – радіус приведення для поступально рухомих елементів установки.

$$\rho = \frac{V_{\text{max}}}{\omega_{\text{к}}} = \frac{D}{2};$$

$$J_{\text{в}} = (11500 + 8200) \cdot 5/2 = 49250 \text{ кг} \cdot \text{м}^2.$$

J_k – приведений до валу двигуна момент інерції підйомного каната.

J_{Σ_d} – момент інерції двигуна і інших елементів механічної частини електроприводу, що обертаються із швидкістю двигуна, а також приведений момент інерції порожнього скипа.

Оскільки моменти інерції всіх елементів не відомі, то J_{Σ_d} визначуваний непрямым шляхом. Відносні маси трьох масової системи, що визначаються по кривим рис. 1.2 [29], рівні: $\mu_1 = 0,73$, $\mu_2 = 0,175$, $\mu_k = 0,095$. Звідси

$$J_{\Sigma_d} = \frac{J_b \mu_1}{\mu_2},$$

$$J_{\Sigma_d} = \frac{49250 \cdot 0,73}{0,175} = 161584 \text{ кг} \cdot \text{м}^2,$$

$$J_k = \frac{J_b \mu_k}{\mu_2},$$

$$J_k = \frac{49250 \cdot 0,095}{0,175} = 21028 \text{ кг} \cdot \text{м}^2,$$

$$J_{\Sigma} = 161584 + 49250 + 21028 = 231862 \text{ кг} \cdot \text{м}^2,$$

$$T_m = 231862 \cdot \frac{0,0254}{170^2} = 0,204 \text{ с}.$$

Постійну часу генератора T_r визначуваний в припущенні, що вона приблизно рівна постійною часу незалежної обмотки збудження (вихровими струмами нехтуємо).

$$T_r = T_{36} = \frac{2 p w_{36} k_{\phi}}{R_{36}},$$

де w_{36} , R_{36} – число витків на полюсі і активний опір обмотки збудження;

k_{ϕ} – коефіцієнт нахилу кривої намагнічення.

Згідно з каталожними даними

$$w_{36} = 224, R_{36} = 3,4 \text{ Ом}, k_{\phi} = 0,745 \cdot 10^{-2},$$

$$T_r = \frac{2 \cdot 4 \cdot 212 \cdot 0,745 \cdot 10^{-2}}{3,1} = 4,08 \text{ с}.$$

2.9 Вибір структури системи автоматичного управління

Система регулювання повинна забезпечувати відповідні механічні характеристики електроприводу як у статичних, так і в динамічних умовах, гарантувати високу якість перехідних процесів і оптимізувати роботу електроприводу за заданими параметрами в різних режимах функціонування механізму.

Для досягнення високих показників регулювання необхідне створення замкнутих систем, тому згідно з рекомендаціями [28], для управління приводом Г-Д використовується система підлеглого регулювання. Функціональна схема приводу наведена на рис. 2.2. Система управління швидкістю є трьохконтурною: регулятор швидкості (РШ) підпорядковується регулятору струму (РТ), який у свою чергу підпорядковується регулятору напруги генератора (ДН). Датчики швидкості, струму та напруги позначені як ДС, ДТ та ДН відповідно. Регулювання швидкості здійснюється за допомогою розузгодження ω_3 і ω_d .

Збудження генератора забезпечується реверсивним тиристорним збудником (ТЗБГ), а генератор – нереверсивним тиристорним збудником (ТЗБД).

Завдання швидкості виконується через командоапарат ручного управління (КАР) та блок завдання швидкості (ЗШ) за допомогою задатчика інтенсивності (ЗІ).

Збудженням двигуна управляє замкнута система. Завдання номінального струму збудження $I_{збд}$ здійснюється подачею напруги $U_{ззб}$ від незалежного стабілізованого джерела напруги на вхід регулятора збудження (РТЗБ). Зворот-

ний зв'язок по $I_{зб\ д}$ знімається з шунта і через датчик струму (ДТЗБ) вводиться на вхід РТЗБ ($I_{зб\ д} k_{зб} = U_{о\ зб}$).

На схемі прийняті наступні позначення: R_d – дільний напруги; Ш – шунт; ТГ – тахогенератор, ДС, ДТ, ДН – датчики швидкості, струму і напруги відповідно.

Принципова схема трьохконтурної системи підлеглого управління приводом Г-Д з ТЗБ показана на рис. 2.3. Синтез регуляторів контурів наведено далі.

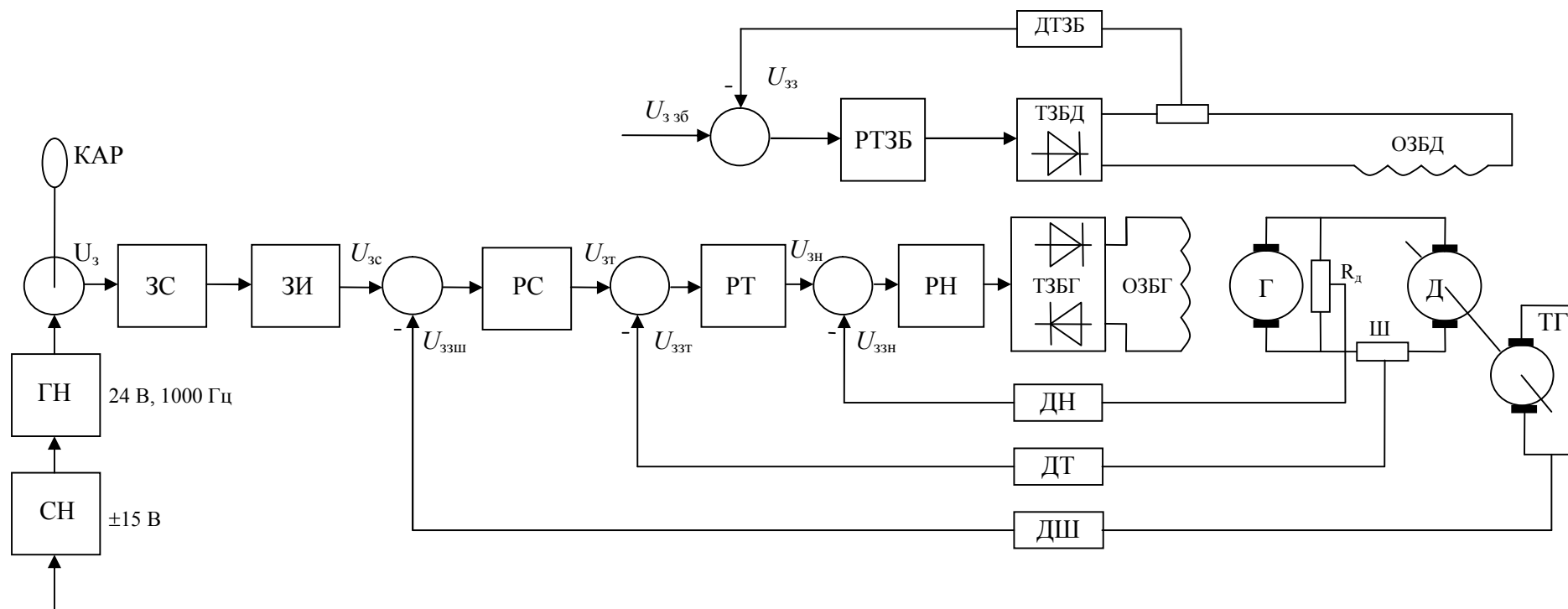


Рисунок 2.2 – Функціональна схема трьохконтурної системи підлеглого управління приводом Г-Д з ТЗБ

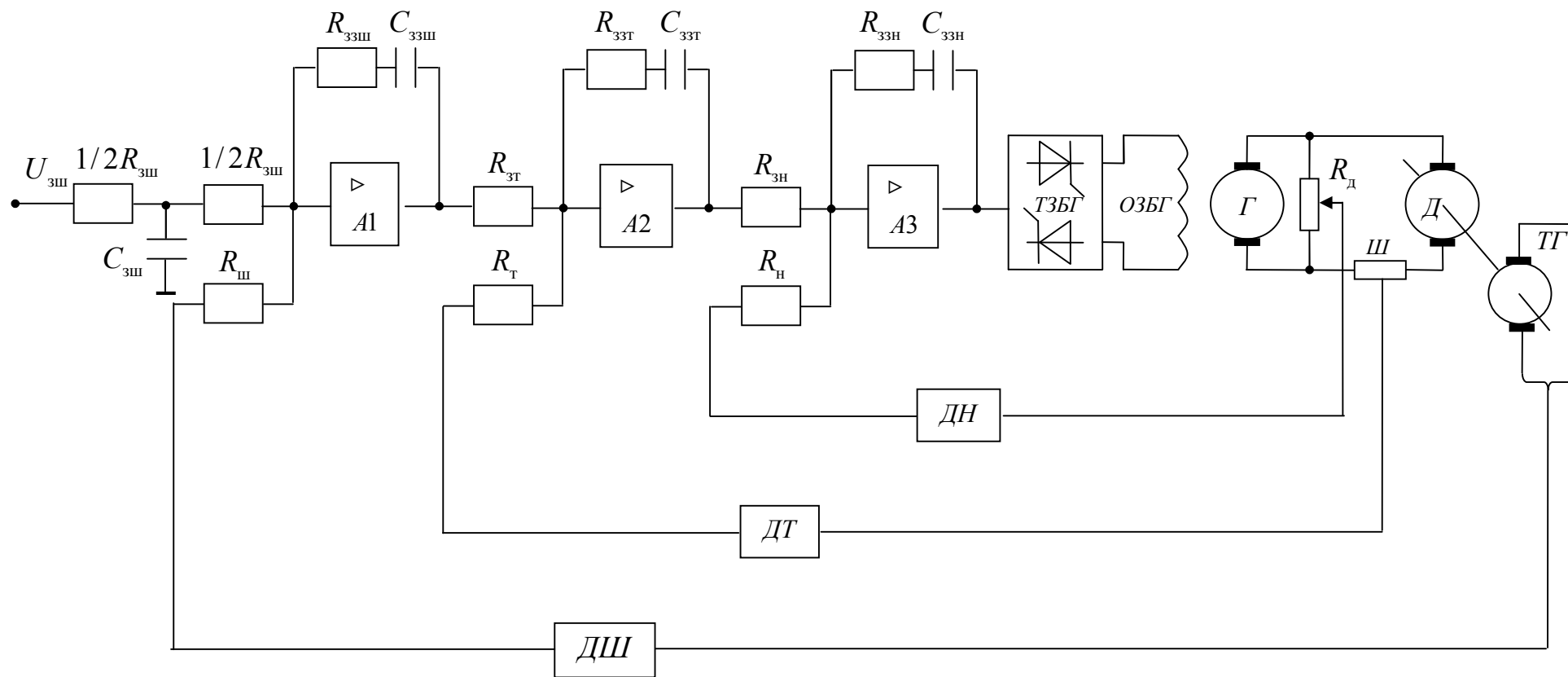


Рисунок 2.3 – Принципова схема трьохконтурної системи підлеглого управління приводом Г-Д з ТЗБ

2.10 Алгоритмічна схема системи

Функціональний і принциповий схемам рис. 2.2 і рис.2.3 відповідає алгоритмічна схема системи, показана на рис.2.4 (система управління збудженням не показана, оскільки в подальших дослідженнях магнітний потік двигуна залишається незмінним).

Алгоритмічна схема складається з наступних передавальних функцій.

Передавальна функція якірного кола електроприводу

$$W_{\text{як}}(p) = \frac{1/R_{\Sigma}}{T_e p + 1}.$$

Передавальна функція механічної частини електроприводу

$$W_{\text{м}}(p) = \frac{k_{\text{д}} R_{\Sigma}}{T_{\text{м}}}.$$

Передавальна функція зворотного зв'язку по ЕДС двигуна

$$W_{\text{зз}}(p) = \frac{1}{k_{\text{д}}}.$$

Передавальна функція генератора

$$W_{\text{г}}(p) = \frac{k_{\text{г}}}{T_{\text{г}} p + 1}.$$

Передавальна функція тиристорного перетворювача (збудника генератора)

$$W_{\text{тп}}(p) = \frac{k_{\text{тп}}}{T_{\text{тп}} p + 1}.$$

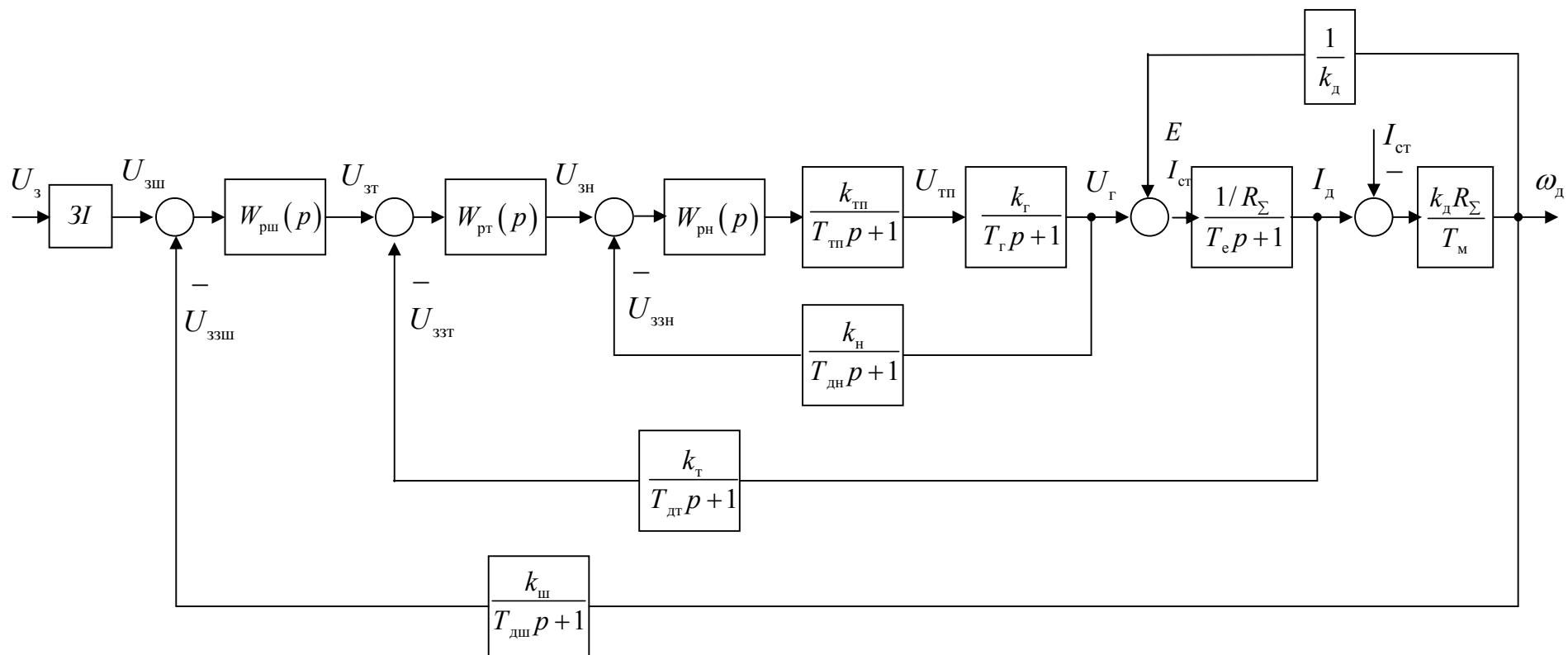


Рисунок 2.4 – Алгоритмічна схема трьохконтурної системи підлеглого управління

Передавальна функція ланцюги зворотного зв'язку по напрузі

$$W_{\text{зн}}(p) = \frac{k_{\text{н}}}{T_{\text{дн}}p + 1},$$

де $T_{\text{дн}}$ – постійна часу датчика напруги.

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

Передавальна функція ланцюги зворотного зв'язку струму

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

де $T_{\text{дт}}$ – постійна часу датчика струму.

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

Передавальна функція ланцюги зворотного зв'язку швидкості

$$W_{\text{зш}}(p) = \frac{k_{\text{ш}}}{T_{\text{дш}}p + 1},$$

де $T_{\text{дш}}$ – постійна часу датчика швидкості.

$k_{\text{ш}}$ – передавальний коефіцієнт ланцюга зворотного зв'язку по швидкості.

$W_{\text{рш}}(p)$, $W_{\text{рт}}(p)$, $W_{\text{рн}}(p)$ – передавальні функції регуляторів швидкості, струму і напруги відповідно, які визначається далі.

Задатчик інтенсивності ЗІ в подальших розрахунках не враховується.

2.11 Синтез послідовного коректуючого пристрою контура напруги

Контур напруги є внутрішнім контуром. Оскільки швидкодія системи в цілому визначається внутрішнім контуром, то, отже, правильний розрахунок параметрів контуру напруги є найбільш відповідальним етапом проектування системи підлеглого регулювання. Алгоритмічна схема контура напруги приведена на рис.2.5.

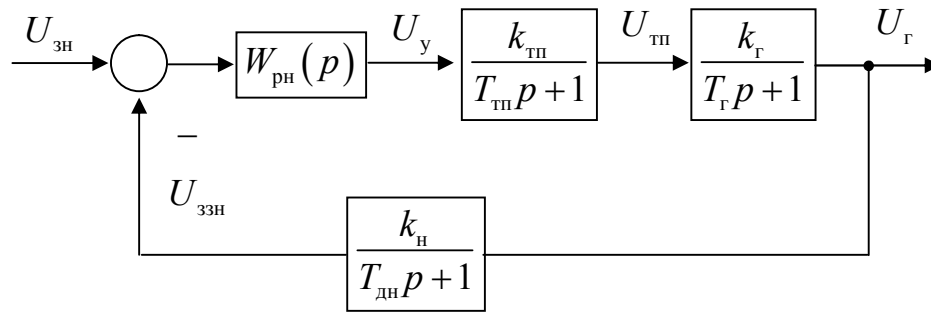


Рисунок 2.5 – Алгоритмічна схема контура напруги

Для того, щоб можна було скористатися загальноприйнятою методикою синтезу послідовних пристроїв, що коректують, необхідно схему рис.2.5, перетворити до алгоритмічної схеми з одиничним зворотним зв'язком (рис.2.6).

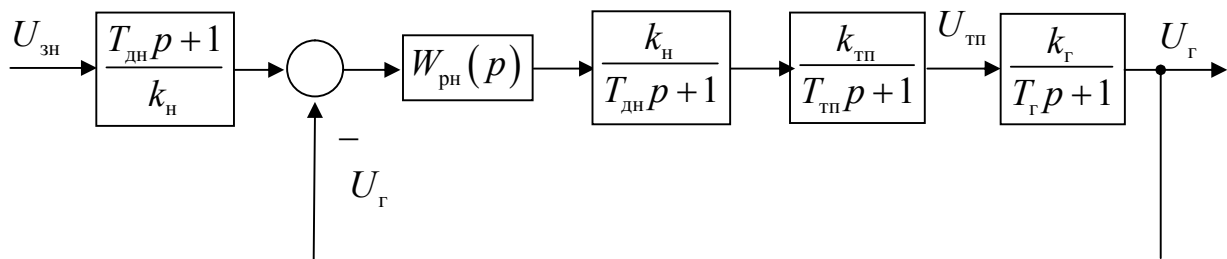


Рисунок 2.6 – Алгоритмічна схема контура напруги з одиничним зворотним зв'язком

Ланки з малими (в порівнянні з постійною часу генератора $T_г$) постійними часу $T_дн$ і $T_тп$ можна об'єднати в одну аперіодичну ланку з малою постійною часу, що не компенсується, контура напруги

$$T_{\mu н} = T_{дн} + T_{тп}.$$

Згідно з рекомендаціям [29] приймаємо $T_{\mu н} = 0,005 \text{ с}$.

Форсуюча ланка $(1 + T_{дн} p)$ у блоці що стоїть на вході замкнутого контура напруги з одиничним зворотним зв'язком (рис. 2.6) при розрахунку контура напруги згідно загальноприйнятій методиці не враховується. З урахуванням

сказаного отримуємо розрахункову алгоритмічну схему контура напруги, показану на рис.2.7.

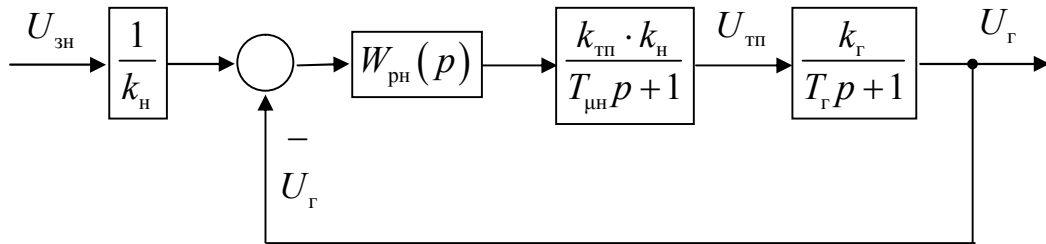


Рисунок 2.7 – Розрахункова алгоритмічна схема контура напруги

Згідно рекомендаціям [28], контур напруги оптимізуємо по модульному критерію. Передавальна функція оптимізованого розімкненого контура напруги (бажана передавальна функція) має вид

$$W_{\text{бн}}(p) = \frac{1}{2T_{\text{мн}} p (T_{\text{мн}} p + 1)}.$$

Передавальна функція об'єкту регулювання контура напруги визначається як

$$W_{\text{он}}(p) = \frac{k_{\text{г}} k_{\text{тп}} k_{\text{н}}}{(T_{\text{мн}} p + 1)(T_{\text{г}} p + 1)},$$

Із співвідношення

$$W_{\text{бн}}(p) = W_{\text{рн}}(p) \cdot W_{\text{он}}(p),$$

визначимо передавальну функцію регулятора напруги

$$W_{\text{рн}}(p) = \frac{W_{\text{бн}}(p)}{W_{\text{он}}(p)} = \frac{\frac{1}{2T_{\text{мн}} p (T_{\text{мн}} p + 1)}}{\frac{k_{\text{г}} k_{\text{тп}} k_{\text{н}}}{(T_{\text{мн}} p + 1)(T_{\text{г}} p + 1)}} = \frac{1}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}} \cdot \frac{T_{\text{г}} p + 1}{p} = k_{\text{рн}} \frac{T_{\text{г}} p + 1}{p}, \quad (2.2)$$

де $k_{pn} = \frac{1}{2T_{\mu n} k_r k_{tp} k_n}$ – коефіцієнт підсилення регулятора напруги.

Передавальній функції (2.2) відповідає ПІ-регулятор, принципова схема якого показана на рис. 2.3.

Передавальна функція типового регулятора напруги визначається наступним чином:

$$W_{\text{ПІ-рег н}}(p) = \frac{Z_{\text{зз н}}(p)}{Z_{\text{вх н}}(p)} = \frac{R_{\text{зз н}} + \frac{1}{C_{\text{зз н}} p}}{R_{\text{з н}}} = \frac{R_{\text{зз н}} C_{\text{зз н}} p + 1}{C_{\text{зз н}} R_{\text{з н}} p}. \quad (2.3)$$

Згідно рис.2.3 передавальний коефіцієнт ланцюга зворотного зв'язку по напрузі:

$$k_n = k_{дг} \frac{R_{\text{з н}}}{R_n},$$

де $k_{дг}$ – коефіцієнт дільника напруги генератора визначається з умови отримання сигналу 24 В на вході ДН при максимальній напрузі генератора (коефіцієнт посилення датчика напруги рівний 1), приймаємо значення напруги $U_{г \text{ max}}$ рівним номінальній напрузі двигуна $U_{г \text{ max}} = 600 \text{ В}$

$$k_{дг} = \frac{24}{U_{г \text{ max}}} = \frac{24}{600} = 0,04.$$

Приймаємо $R_{\text{з н}} = R_n$, отримаємо

$$k_n = k_{дг} = 0,04.$$

Зіставлення передавальної функції (2.2) з передавальною функцією типового регулятора (2.3), отримуємо наступні рівняння

$$R_{\text{зз н}} C_{\text{зз н}} = T_{г}; \quad (2.4)$$

$$R_{3H} C_{33H} = 2T_{\mu H} k_{\Gamma} k_{TP} k_H, \quad (2.5)$$

Використовуючи (2.4) і (2.5) розраховуємо параметри типового регулятора: C_{33H} , R_{33H} , R_{3H} , R_H . Приймаємо $C_{33H} = 10 \cdot 10^{-6} \Phi$, тоді

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

$$R_{33H} = \frac{4,08}{8 \cdot 10^6} = 510 \cdot 10^3 \text{ Ом}.$$

$$R_{3H} = R_H = \frac{2T_{\mu H} k_{\Gamma} k_{TP} k_H}{C_{33H}},$$

$$R_{3H} = R_H = \frac{2 \cdot 0,005 \cdot 3,34 \cdot 22 \cdot 0,032}{8 \cdot 10^6} = 9,6 \cdot 10^3 \text{ Ом}.$$

Передавальна функція замкнутого контура напруги, оптимізованої згідно модульного критерія, має вид

$$W_{3H}(p) = \frac{1/k_H}{2T_{\mu H}^2 p^2 + 2T_{\mu H} p + 1}. \quad (2.6)$$

2.12 Синтез послідовного коректуючого пристрою контура струму

Об'єкт регулювання контура струму складається з якірного кола двигуна, що є аперіодичною ланкою з коефіцієнтом посилення $1/R_{\Sigma}$ і великою постійною часу T_e , оптимізованого контура напруги, властивості якого визначаються передавальною функцією (2.6) і датчика струму.

Алгоритмічна схема контура струму зображена на рис. 2.8.

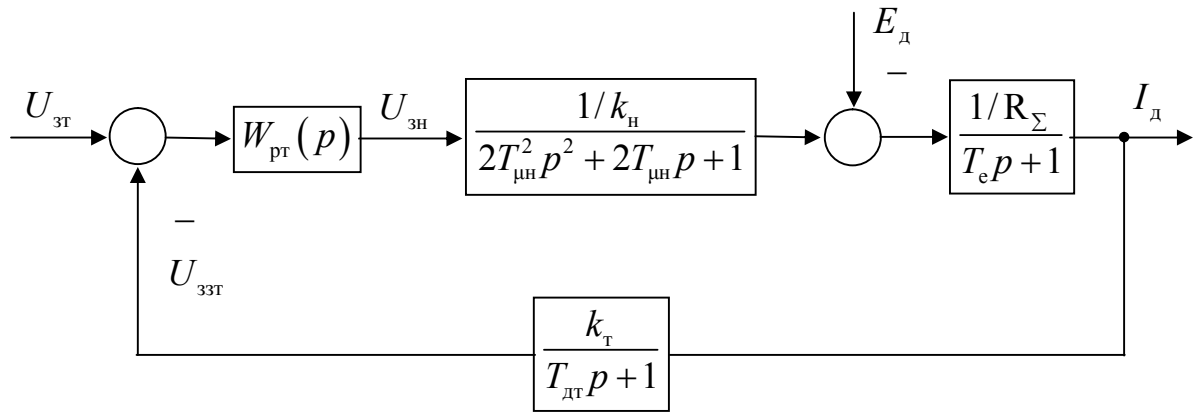


Рисунок 2.8 – Алгоритмічна схема контура струму

Ця схема, аналогічно контуру напруги, може бути перетворена до алгоритмічної схеми з одиничним зворотним зв'язком, як показано на рис. 2.9.

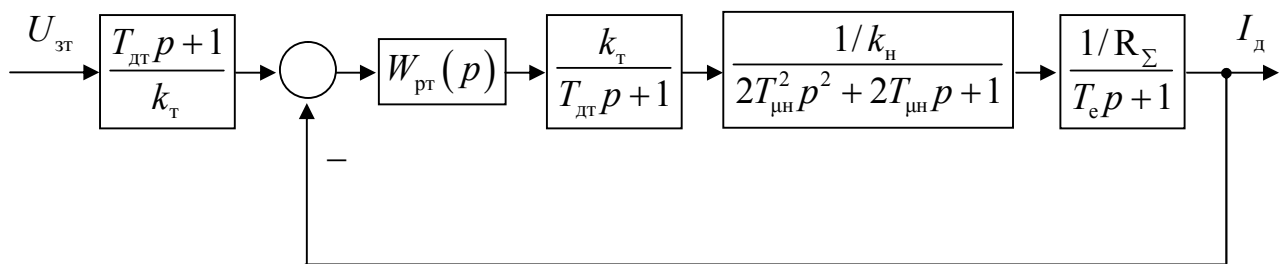


Рисунок 2.9 – Алгоритмічна схема контура струму з одиничним зворотним зв'язком

Мала некомпенсуєма постійна часу контура струму $T_{\mu\tau}$ з достатньою для практики точністю визначаються по формулі

$$T_{\mu\tau} = 2T_{\mu\text{н}} + T_{\text{дт}} .$$

Вважаючи $T_{\text{дт}} = 0$, отримуємо $T_{\mu\tau} = 2T_{\mu\text{н}} = 0,01 \text{ с}$.

Розрахункова алгоритмічна схема контура струму показана на рис.2.10.

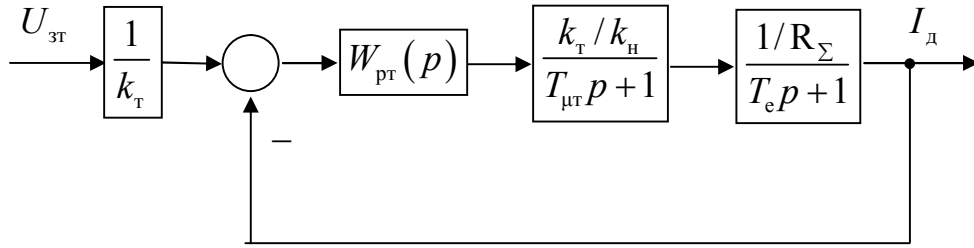


Рисунок 2.10 – Розрахункова алгоритмічні схеми контура струму.

Контур струму оптимізуємо по модульному критерію, тоді бажана передавальна функція розімкненого контура струму має вид

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

Передавальна функція об'єкту регулювання контура струму

$$W_{\text{от}}(p) = \frac{1/k_{\text{н}} \cdot 1/R_{\Sigma} \cdot k_{\text{т}}}{(T_{\text{пт}}p + 1)(T_{\text{е}}p + 1)}.$$

Передавальна функція регулятора контура струму

$$W_{\text{пт}}(p) = \frac{W_{\text{от}}(p)}{W_{\text{обт}}(p)} = \frac{\frac{1}{2T_{\text{пт}}p(T_{\text{пт}}p + 1)}}{\frac{1/k_{\text{н}} \cdot 1/R_{\Sigma} \cdot k_{\text{т}}}{(T_{\text{пт}}p + 1)(T_{\text{е}}p + 1)}} = \frac{k_{\text{н}} \cdot R_{\Sigma}}{2T_{\text{пт}}k_{\text{т}}} \cdot \frac{T_{\text{е}}p + 1}{p} = k_{\text{пт}} \cdot \frac{T_{\text{е}}p + 1}{p}. \quad (2.7)$$

Передавальна функція (2.7) відповідає типовому ПІ-регулятору, схема якого показана на рис. 2.3.

Передавальна функція типового ПІ-регулятора струму визначається наступним чином

$$W_{\text{ПІ-рег т}}(p) = \frac{Z_{\text{зз т}}(p)}{Z_{\text{вх т}}(p)} = \frac{R_{\text{ззт}} + \frac{1}{C_{\text{ззт}}p}}{R_{\text{зт}}} = \frac{R_{\text{ззт}}C_{\text{ззт}}p + 1}{C_{\text{ззт}}R_{\text{зт}}p}. \quad (2.8)$$

Коефіцієнт посилення ланцюга зворотного зв'язку по струму:

$$k_{\text{т}} = k_{\text{ш}} \cdot k_{\text{дт}} \cdot \frac{R_{\text{зт}}}{R_{\text{т}}},$$

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

$U_{\text{н ш}}$ – номінальна вихідна напруга шунта, приймаємо $U_{\text{н ш}} = 75 \cdot 10^{-3} \text{ В}$;

$I_{\text{н ш}}$ – струм шунта, приймаємо $I_{\text{н ш}} = I_{\text{д max}}$;

$I_{\text{д max}} = 2I_{\text{н}}$ – струм максимального робочого перевантаження двигуна

$$I_{\text{н ш}} = 2,5 \cdot I_{\text{н}} = 2,5 \cdot 2625 = 6562 \text{ А},$$

$$k_{\text{ш}} = \frac{0,075}{6562} = 1,14 \cdot 10^{-5} \text{ В / А}.$$

Значення $k_{\text{дт}}$ визначаємо із співвідношення

$$U_{\text{дт max}} = k_{\text{дт}} \cdot k_{\text{ш}} \cdot I_{\text{д max}} \leq 24 \text{ В}, \quad (2.9)$$

де $U_{\text{дт max}}$ – максимальна напруга на виході датчика струму.

З виразу (2.9) одержуємо:

$$k_{\text{дт}} \leq \frac{24}{k_{\text{ш}} I_{\text{д max}}},$$

$$k_{\text{дт}} = \frac{24}{1,14 \cdot 10^{-4} \cdot 2,5 \cdot 2625} = 320.$$

Приймаємо як датчик струму операційний підсилювач з коефіцієнтом підсилювача $k_{\text{дт}} = 320$, тоді при $R_{\text{зт}} = R_{\text{т}}$

$$k_{\text{т}} = 1,14 \cdot 10^{-5} \cdot 320 = 3,66 \cdot 10^{-3} \text{ В / А}.$$

Зіставлення передавальної функції (2.7) з передавальною функцією типового регулятора (2.8) дає наступні співвідношення:

$$R_{33T} C_{33T} = T_e,$$

$$R_{3T} C_{33T} = \frac{2T_{\mu T} k_T}{k_H R_{\Sigma}}.$$

Приймаємо $C_{33T} = 5 \text{ мкФ}$, тоді

$$R_{33T} = \frac{T_e}{C_{33T}},$$

$$R_{33T} = \frac{0,036}{0,5 \cdot 10^{-6}} = 72 \cdot 10^3 \text{ Ом}$$

$$R_{3T} = R_T = \frac{2T_{\mu T} k_T}{k_H R_{\Sigma} C_{33T}},$$

$$R_{3T} = R_T = \frac{2 \cdot 0,01 \cdot 3,66 \cdot 10^{-3}}{0,109 \cdot 0,0254 \cdot 0,5 \cdot 10^6} = 58,2 \cdot 10^3 \text{ Ом}.$$

Передавальна функція замкнутого контура струму, оптимізованого згідно модульному критерію, має вид

$$W_{3T}(p) = \frac{1/k_T}{2T_{\mu T}^2 p^2 + 2T_{\mu T} p + 1}.$$

2.13 Синтез послідовного коректуючого пристрою контура швидкості

Контур швидкості є зовнішнім, і тому динамічні властивості системи повністю визначаються динамічними властивостями контура швидкості.

До складу об'єкту регулювання контура швидкості входять: механічна частина електроприводу, що є ідеальною інтегруючою ланкою з коефіцієнтом по-

силення за швидкістю $\frac{R_{\Sigma}}{cT_M}$, оптимізований контур струму і датчик швидкості, технічна реалізація якого здійснюється за допомогою тахогенератора постійного струму.

Алгоритмічна схема контура швидкості представлена на рис. 2.11.

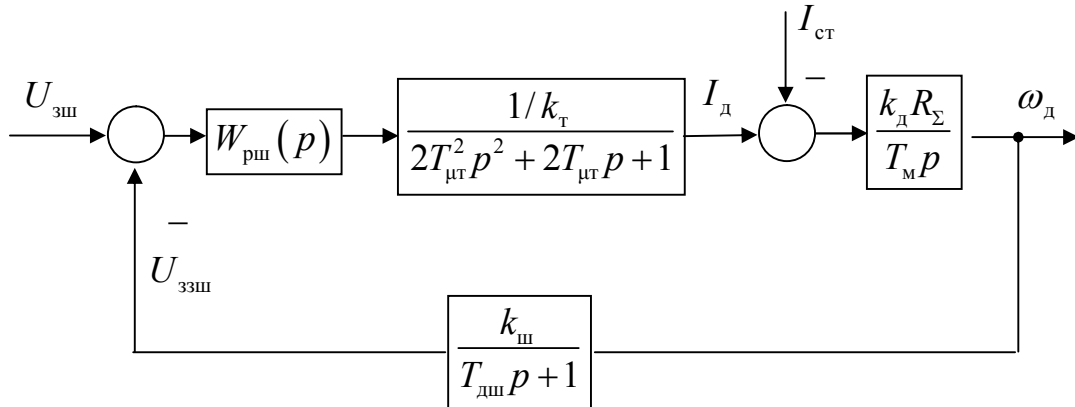


Рисунок 2.11 – Початкова алгоритмічна схема контура швидкості

Ця схема перетворюється до алгоритмічної схеми з одиничним зворотним зв'язком, показаним на рис.2.12.

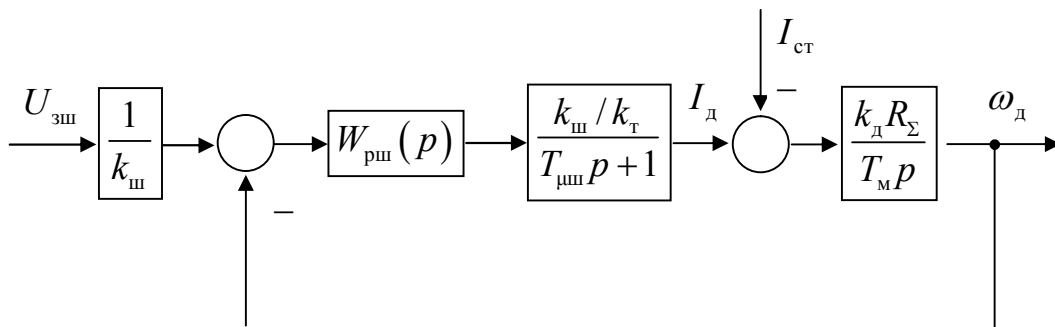


Рисунок 2.12 – Розрахункова алгоритмічна схема контура швидкості

Мала некомпенсуєма постійна часу контура швидкості рівна

$$T_{\mu\text{ш}} = 2T_{\mu\text{т}} + T_{\text{дш}} ,$$

Приймаємо $T_{\text{дс}} = 0$, тоді $T_{\mu\text{ш}} = 2T_{\mu\text{т}} = 0,02 \text{ с}$.

Контур швидкості, згідно рекомендаціям [28], оптимізуємо по симетричному критерію.

Бажана передавальна функція контура швидкості, оптимізованого по симетричному критерію, має вид

$$W_{\text{бш}}(p) = \frac{(1 + 4T_{\mu\text{ш}}p)}{8T_{\mu\text{ш}}^2 p^2 (1 + T_{\mu\text{ш}}p)}.$$

Передавальна функція об'єкту регулювання контура швидкості:

$$W_{\text{ош}}(p) = \frac{k_{\text{ш}} \cdot k_{\text{д}} \cdot R_{\Sigma}}{T_{\text{м}} \cdot k_{\text{т}}} \cdot \frac{1}{p(1 + T_{\mu\text{ш}}p)}.$$

Передавальна функція регулятора швидкості

$$\begin{aligned} W_{\text{рш}}(p) &= \frac{W_{\text{бш}}(p)}{W_{\text{ош}}(p)} = \frac{\frac{(1 + 4T_{\mu\text{ш}}p)}{8T_{\mu\text{ш}}^2 p^2 (1 + T_{\mu\text{ш}}p)}}{\frac{k_{\text{ш}} k_{\text{д}} R_{\Sigma}}{T_{\text{м}} k_{\text{т}}} \cdot \frac{1}{p(1 + T_{\mu\text{ш}}p)}} = \\ &= \frac{T_{\text{м}} k_{\text{т}}}{8T_{\mu\text{ш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}} \cdot \frac{1 + 4T_{\mu\text{ш}}p}{p} = k_{\text{рш}} \cdot \frac{1 + 4T_{\mu\text{ш}}p}{p}. \end{aligned} \quad (2.10)$$

Дана передавальна функція реалізується за допомогою типового ПІ-регулятора, схема якого показана на рис. 2.3.

Передавальна функція типового ПІ-регулятора швидкості

$$W_{\text{ПІ-рег ш}}(p) = \frac{Z_{\text{зз ш}}(p)}{Z_{\text{вх ш}}(p)} = \frac{R_{\text{ззш}} + \frac{1}{C_{\text{ззш}}p}}{R_{\text{зш}}} = \frac{R_{\text{ззш}} C_{\text{ззш}} p + 1}{C_{\text{ззш}} R_{\text{зш}} p}. \quad (2.11)$$

Розрахунок параметрів регулятора швидкості виконується, виходячи з співвідношень, отриманих шляхом зіставлення передатних функцій (2.10) і (2.11):

$$\frac{R_{33\text{ш}}}{R_{3\text{ш}}} = \frac{T_{\text{м}} \cdot k_{\text{т}}}{T_{\text{мш}} \cdot k_{\text{ш}} \cdot k_{\text{д}} \cdot R_{\Sigma}}, \quad (2.12)$$

$$k_{\text{ш}} = k_{\text{тг}} \cdot \frac{R_{3\text{ш}}}{R_{\text{ш}}}, \quad (2.13)$$

$$R_{33\text{ш}} \cdot C_{33\text{ш}} = 4T_{\text{мш}}, \quad (2.14)$$

де $k_{\text{тг}} = \frac{U_{\text{н тг}}}{\omega_{\text{н тг}}}$ – коефіцієнт посилення тахогенератора, по таблиці П9 [29] вибираємо тахогенератор ПТ – 42, частота обертання $n_{\text{н тг}} = 1000 \text{ об/хв}$, напруга $U_{\text{н тг}} = 230 \text{ В}$, струм якоря $I_{\text{н тг}} = 0,1 \text{ А}$.

$$\omega_{\text{н тг}} = \frac{\pi \cdot n_{\text{н тг}}}{30},$$

$$\omega_{\text{н тг}} = \frac{3,14 \cdot 1000}{30} = 10,5 \text{ с}^{-1}$$

$$k_{\text{тг}} = \frac{230}{10,5} = 21,9.$$

Сумісне рішення рівнянь (2.12) і (2.14) дає наступне співвідношення:

$$\frac{R_{33\text{ш}}}{R_{\text{ш}}} = \frac{T_{\text{м}} \cdot k_{\text{т}}}{8T_{\text{мш}}^2 \cdot k_{\text{тг}} \cdot k_{\text{д}} \cdot R_{\Sigma}}; \quad (2.15)$$

Відношення опорів $\frac{R_{3\text{ш}}}{R_{\text{ш}}}$ визначається з рівняння

$$U_{3\text{ш max}} = k_{\text{ш}} \cdot \omega_{\text{max}} = k_{\text{тг}} \cdot \frac{R_{3\text{ш}}}{R_{\text{ш}}} \cdot \omega_{\text{max}},$$

звідки

$$\frac{R_{3\text{ш}}}{R_{\text{ш}}} = \frac{U_{3\text{ш max}}}{k_{\text{тг}} \cdot \omega_{\text{max}}}, \quad (2.16)$$

де $\omega_{\max}$ – максимальне задане значення швидкості обертання двигуна, приймаємо $\omega_{\max} = \omega_0$, де ω_0 – швидкість ідеального холостого ходу двигуна

$$\omega_0 = \frac{U_{\text{н}}}{c} = \frac{750}{170} = 4,41 \text{ c}^{-1}.$$

$U_{\text{зш max}}$ – вихідна напруга задаючого елемента, що відповідає $\omega_{\max}$, приймаємо $U_{\text{зш max}} = 24 \text{ В}$.

Задаємося величиною $R_{\text{зш}} = 40 \text{ кОм}$, тоді з (2.16) знаходимо

$$R_{\text{ш}} = \frac{k_{\text{тг}} \cdot \omega_{\max}}{U_{\text{зш max}}} \cdot R_{\text{зш}},$$

$$R_{\text{ш}} = \frac{22 \cdot 4,41}{24} \cdot 80 \cdot 10^3 = 314 \cdot 10^3 \text{ Ом}.$$

Із співвідношення (2.15) визначаємо опір $R_{\text{ззш}}$

$$R_{\text{ззш}} = \frac{T_{\text{м}} \cdot k_{\text{т}}}{8T_{\text{мш}}^2 \cdot k_{\text{тг}} \cdot k_{\text{д}} \cdot R_{\Sigma}} \cdot R_{\text{ш}},$$

$$R_{\text{ззш}} = \frac{0,204 \cdot 170 \cdot 3,66 \cdot 10^{-3}}{2 \cdot 0,02 \cdot 22 \cdot 0,0254} \cdot 314 \cdot 10^3 = 178,7 \cdot 10^3 \text{ Ом}.$$

Передавальна функція замкнутого контура швидкості, оптимізованого по симетричному критерію, має вид:

$$W_{\text{зш}}(p) = \frac{1}{k_{\text{ш}}} \cdot \frac{1 + 4T_{\text{мш}}}{8T_{\text{мш}}^3 p^3 + 8T_{\text{мш}}^2 p^2 + 4T_{\text{мш}} p + 1}.$$

Як відомо, при оптимізації контура швидкості по симетричному критерію має місце значне (43%) перерегулювання швидкості по задаючій дії. Для зменшення перерегулювання на вхід замкнутого контура швидкості ставимо фільтр (див. рис.2.3), що є аперіодичною ланкою з постійною часу $4T_{\text{мш}}$. При включен-

ні фільтру, як показано на рис. 2.3, величина ємності $C_{зш}$ визначається з рівняння

$$\frac{R_{зс}}{4} C_{зш} = 4T_{\mu ш},$$

звідки

$$C_{зш} = \frac{4 \cdot 4 \cdot T_{\mu ш}}{R_{зш}},$$

$$C_{зш} = \frac{4 \cdot 4 \cdot 0,02}{80 \cdot 10^3} = 4 \cdot 10^{-6} \text{ Ф}.$$

Передавальна функція контура швидкості з фільтром на вході має вид:

$$W_{зш\phi}(p) = \frac{1}{k_{ш}} \cdot \frac{1}{8T_{\mu ш}^3 p^3 + 8T_{\mu ш}^2 p^2 + 4T_{\mu ш} p + 1}.$$

Перерегулювання в такій системі складає $\sigma = 8,1\%$.

Алгоритмічна схема системи з розкриттям передавальних функцій показана на рис. 2.13. З алгоритмічної схеми, шляхом її структурного перетворення, визначаємо наступні вирази для передавальних функцій по задаючій $U_{зш}(t)$ і обурюючій $I_{ст}(t)$ діям

$$W_{\omega_d U_{зш}}(p) = \frac{\omega_d(p)}{U_{зш}(p)} = \frac{1}{k_{ш}} \cdot \frac{1 + 4T_{\mu ш} p}{8T_{\mu ш}^3 p^3 + 8T_{\mu ш}^2 p^2 + 4T_{\mu ш} p + 1},$$

$$W_{\omega_d I_{ст}}(p) = \frac{\omega_d(p)}{I_{ст}(p)} = -\frac{k_d \cdot R_{\Sigma} \cdot 8T_{\mu ш}^2}{T_m} \cdot \frac{p \cdot (1 + T_{\mu ш} p)}{8T_{\mu ш}^3 p^3 + 8T_{\mu ш}^2 p^2 + 4T_{\mu ш} p + 1},$$

$$\Phi_{I_d U_{зш}}(p) = \frac{I_d(p)}{U_{зш}(p)} = \frac{T_m}{k_d R_{\Sigma} k_{ш}} \cdot \frac{p(1 + 4T_{\mu ш} p)}{8T_{\mu ш}^3 p^3 + 8T_{\mu ш}^2 p^2 + 4T_{\mu ш} p + 1},$$

$$\Phi_{I_d I_{ст}}(p) = \frac{I_d(p)}{I_{ст}(p)} = \frac{1 + 4T_{\mu ш} p}{8T_{\mu ш}^3 p^3 + 8T_{\mu ш}^2 p^2 + 4T_{\mu ш} p + 1}.$$

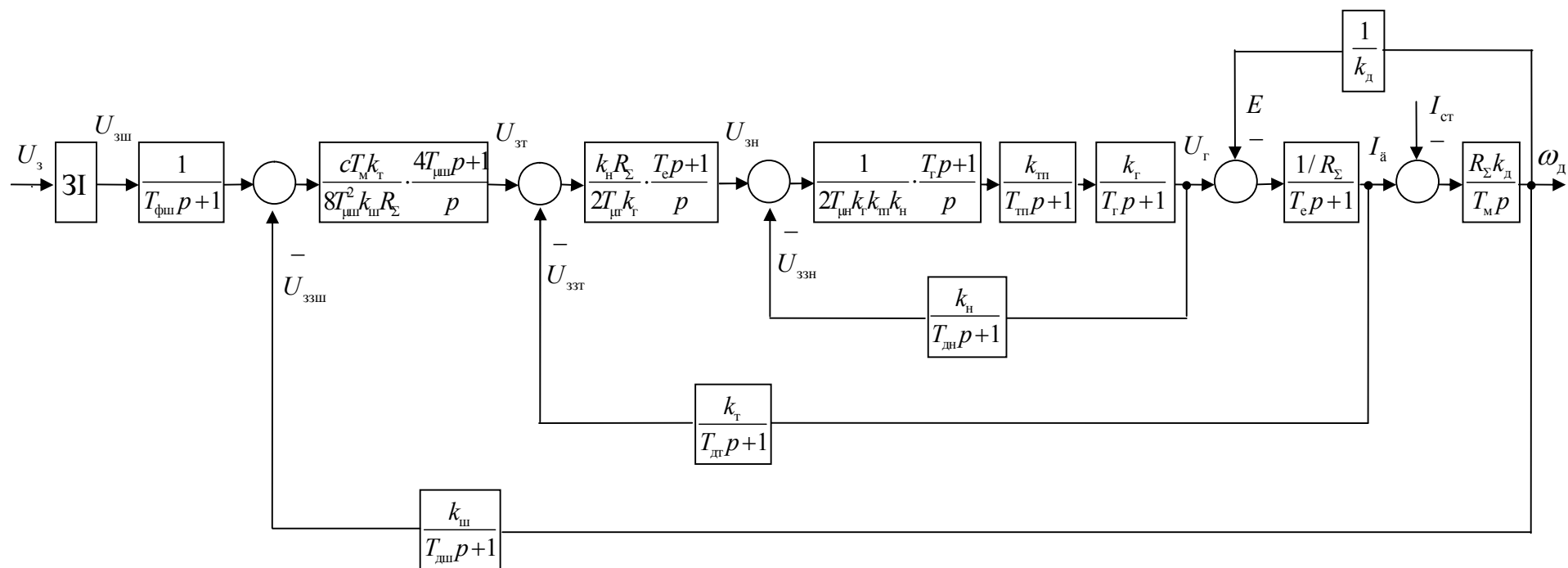


Рисунок 2.13 – Алгоритмічна схема системи підлеглого регулювання швидкості

При включенні фільтру на вхід системи передавальні функції мають вид:

$$W_{\omega_d U_{\text{зш}}}(p) = \frac{\omega_d(p)}{U_{\text{зш}}(p)} = \frac{1}{k_{\text{ш}}} \cdot \frac{1}{8T_{\text{ш}}^3 p^3 + 8T_{\text{ш}}^2 p^2 + 4T_{\text{ш}} p + 1},$$

$$W_{\omega_d I_{\text{ст}}}(p) = \frac{\omega_d(p)}{I_{\text{ст}}(p)} = -\frac{k_d R_{\Sigma} 8T_{\text{ш}}^2}{T_{\text{м}}} \cdot \frac{p}{8T_{\text{ш}}^3 p^3 + 8T_{\text{ш}}^2 p^2 + 4T_{\text{ш}} p + 1},$$

Передавальні функції по обурюючій дії залишаються незмінними.

2.14 Дослідження динамічних характеристик системи

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

Графіки перехідних процесів змінних стану системи по задаючій дії показані на рис. 2.15, криві 1. Як видно з графіків, при настроюванні системи на симетричний оптимум, має місце значне перерегулювання швидкості по задаючій дії, рівне 43%. Для зменшення перерегулювання на вхід системи ставимо фільтр з постійною часу $4T_{\text{ш}}$. Графіки перехідних процесів системи з фільтром на вході показані на рис. 2.15, криві 2. Як видно з графіка перерегулювання зменшилося до 8,1%, проте час першого погоджено і час регулювання при цьому збільшився.

Графіки перехідних процесів змінних стану системи по збурюючій дії показані на рис. 2.16.

Розрахунок перехідних процесів проводився також за допомогою моделі системи, показаної на рис. 2.14, побудованої на ЕОМ в пакеті MATLAB. Графіки перехідних процесів, отримані за допомогою моделі, повністю співпадає з графіками отриманими по передавальних функціях.

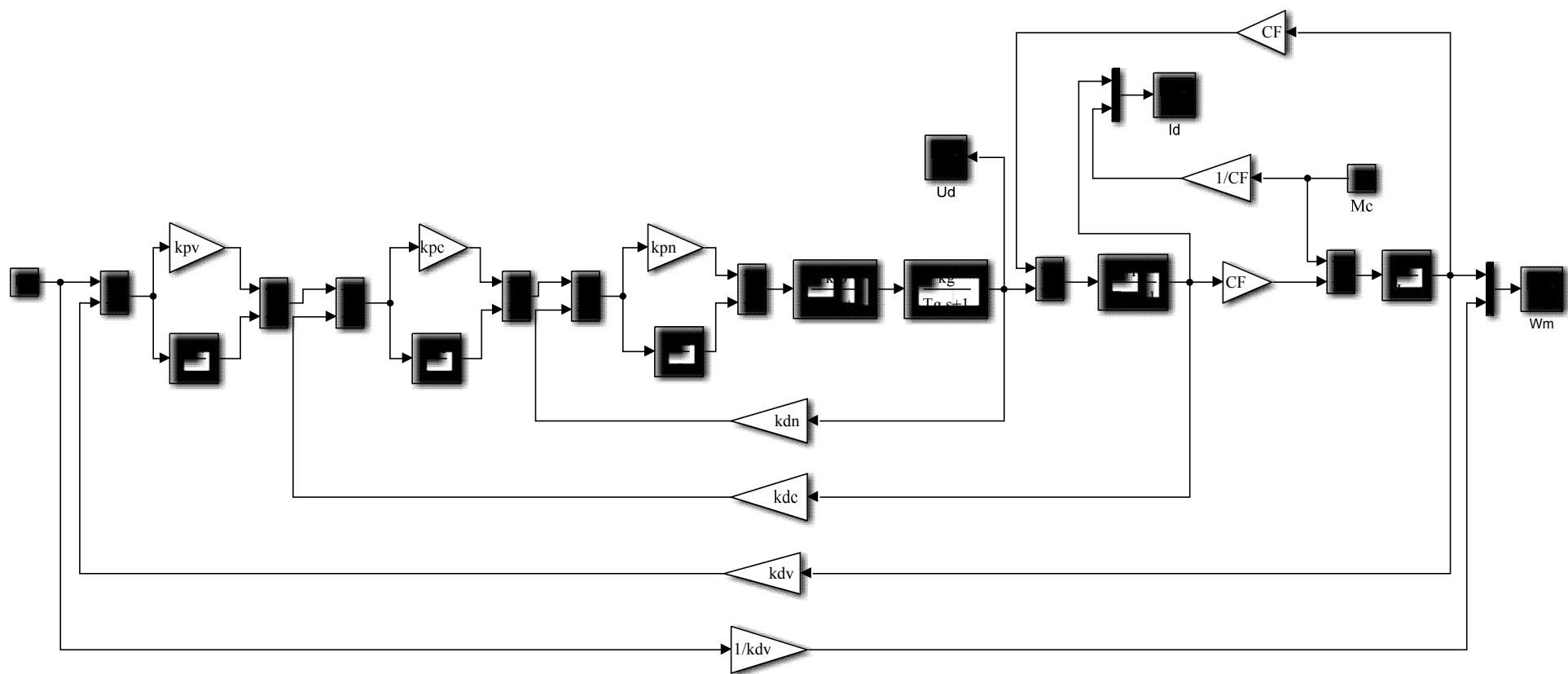


Рисунок 2.14 – Схема моделі системи, розроблена на ЕОМ

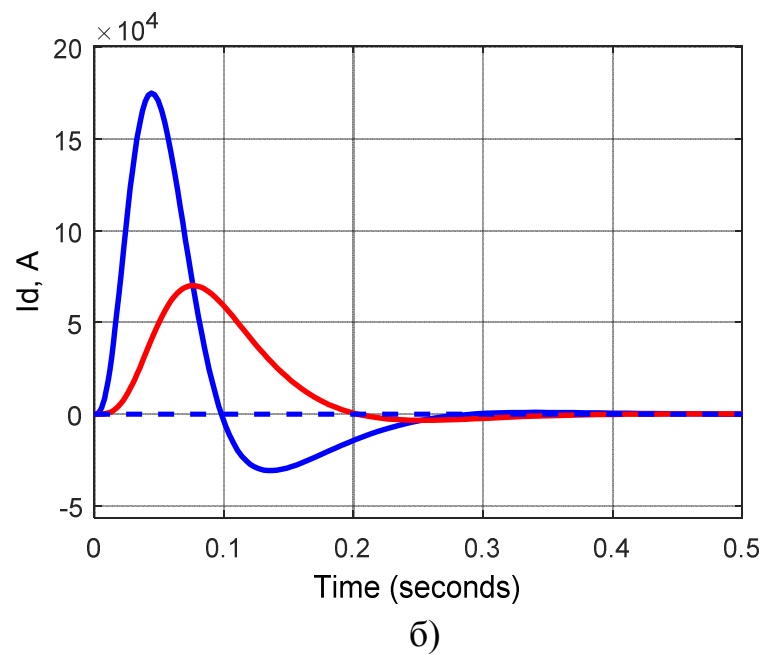
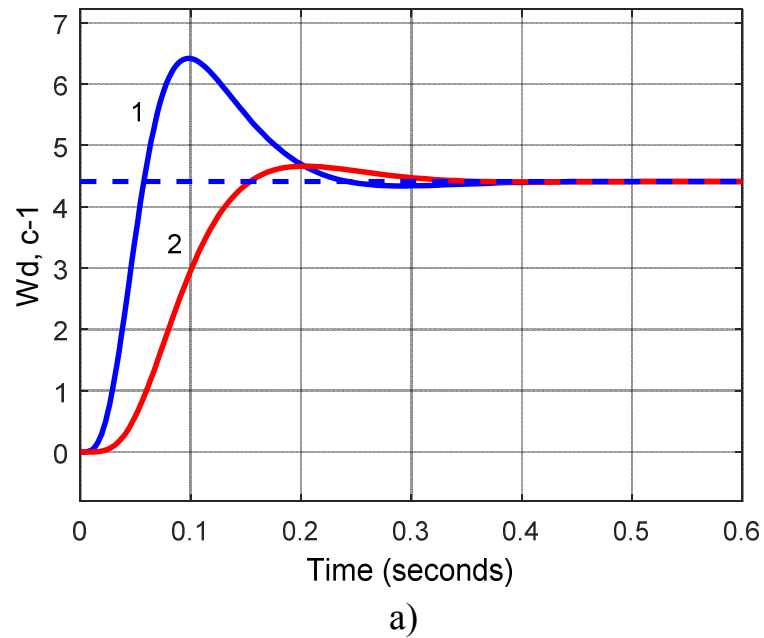
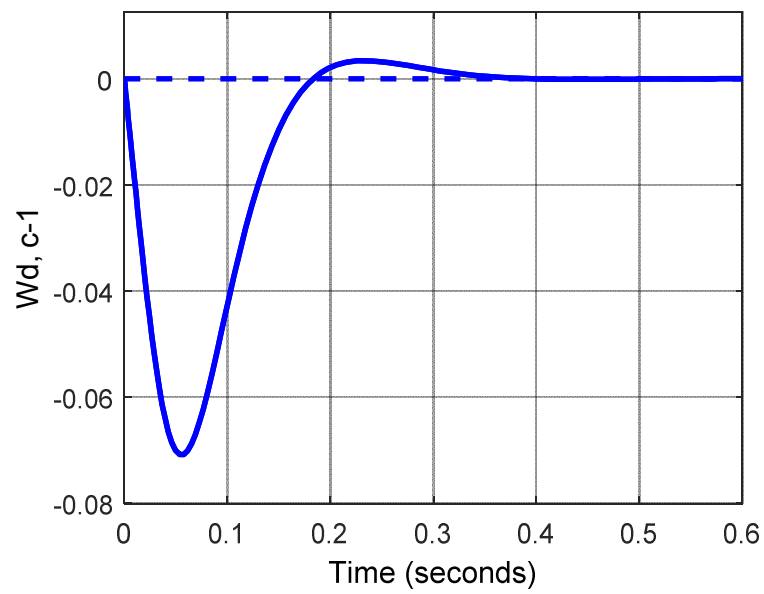
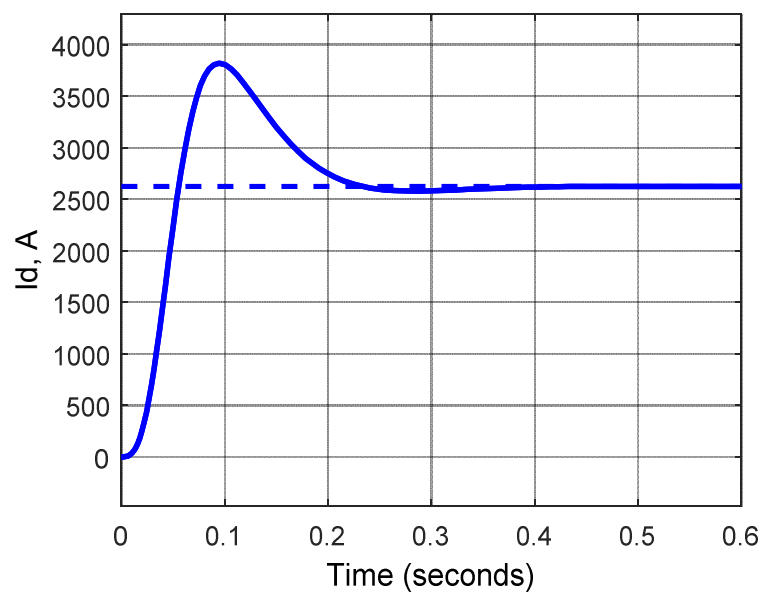


Рисунок 2.15 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ а) і струму $I_d = f(t)$ (б) по задаючій дії



a)



б)

Рисунок 2.16 – Графіки перехідних процесів швидкості $\omega_d = f(t)$ а) і струму $I_d = f(t)$ (б) по збурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 2

1. На підставі аналізу технічних характеристик і умови роботи підйомних установок для електроприводу головного підйому вибраний безредукторний електропривод постійного струму, виконаний за системою Г–Д.

2. Виконано вибір електродвигуна і основного електроустаткування силового кола електроприводу. Проведена перевірка двигуна по перевантажувальній здатності. Встановлено, що прийнятий електродвигун по перевантажувальній здатності відповідає вимогам, оскільки допустиме короткочасне перевантаження двигунів постійного струму відповідає $2 \div 2,5$. Виконано перевірку двигуна за умовами нагріву та перевантажувальній здатності.

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

4. Проведено вибір структури системи автоматичного управління. Система регулювання повинна пояснювати необхідні механічні характеристики електроприводу в статичі і динаміці, високу якість протікання перехідних процесів і оптимізацію роботи електроприводу за заданими показниками в різних режимах роботи механізму, тому для електроприводу механізму підйому шахтної підйимальної установки вибрано трьохконтурну систему підлеглого регулювання з контуром напруги, струму і зовнішнім контуром швидкості.

5. Проведений синтез послідовних коректуючи пристроїв контурів. Оскільки характер перехідного процесу визначається співвідношеннями постійних часу елементів системи, тому формування високоякісних перехідних процесів пов'язане із забезпеченням необхідних значень вказаних співвідношень шляхом введення коректуючи регуляторів.. Для контурів напруги, струму і швидкості синтезовані ПІ-регулятори, унаслідок чого контур напруги і струму оптимізовані по модульному критерію, а контур швидкості – по симетричному критерію. Таким чином, згідно завдання, система є астатичною з астатизмом другого порядку по задаючій дії і астатичною з астатизмом першого порядку по збурюючій дії.

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. При налаштуванні системи на симетричний оптимум є значне (43,3%) перерегулювання швидкості. Для зменшення перерегулювання на вихід системи включено фільтр з постійною часу, рівною $4T_{\mu E}$, унаслідок чого перерегулювання знижене до 8,1%.

РОЗДІЛ 3

ДОСЛІДЖЕННЯ ДИНАМІЧНИХ ХАРАКТЕРИСТИК СИСТЕМИ З УРАХУВАННЯМ ПРУЖНИХ ВЛАСТИВОСТЕЙ ПІДЙОМНОГО КАНАТА

3.1 Математична модель двомасової системи

Розрахунок системи регулювання, що був проведений раніше, ґрунтувався на припущенні нескінченної жорсткості всіх елементів механічної частини системи, включаючи підйомний канат, тобто для одномасової системи. У реальних системах зв'язки між елементами мають обмежену жорсткість, а підйомний канат є пружним елементом з розподіленими параметрами, отже механізм підйому копальневої підйомної установки є багатомасовою системою, яка складається з ряду поступально і обертальний рухомих мас. Ці маси зв'язані механічним зв'язком, кінцевою жорсткістю, що володіє. Особливе значення в шахтних установках має кінцева жорсткість підйомного каната. Унаслідок пружних властивостей каната мають місце подовжні коливання вантажу, що несприятливо позначається на роботі механізму підйому.

Найбільший вплив на процеси в системі надає навантажена гілка каната при розташуванні судини в крайньому нижньому положенні. Оскільки тривалість перехідного процесу при розгоні трохи, та зміна довжини схилу каната за рахунок навивки його на барабан нехтує мала по відношенню до загальної довжини каната. Внаслідок цього можна прийняти, що довжина пружного елемента, а відповідно і жорсткості його в інтервалі часу розгону незмінна.

На рис. 3.1 приведена кінематична схема двохкінцевої шахтної підйомної установки.

На рис. 3.2 площі прямокутників, що зображають приведені до валу двигуна моменти інерції елементів установки, пропорційні їх значенням, а довжини з'єднань між ними обернено пропорційні значенням приведених жорсткостей зв'язку. Прямокутник з номером 9 відповідає зосередженій масі підйомного каната.

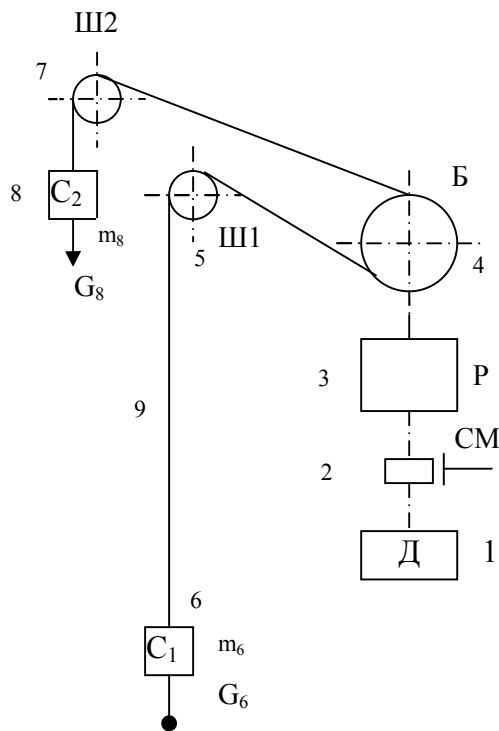


Рисунок 3.1 – Кінематична схема двохкінцевої шахтної підйомної установки.

На схемі позначено: Д – двигун; СМ – сполучна муфта; Р – редуктор; Б – барабан; Ш1, Ш2 – направляючі шківи; С1, С2 – скипи; m_6, m_8 – маси завантаженого і порожнього скипів.

Розглядаючи рис. 3.2 можна переконатися, що механічна частина електроприводу двохкінцевої підйомної установки є розгалуженою механічною системою. У положенні скипів, коли один з них з вантажем знаходиться в крайньому нижньому положенні, а другий (порожній) в крайньому верхньому положенні, схему можна перетворити таким чином: порожній скип С2 (елемент 8) пов'язаний з барабаном Б (елемент 4) через направляючий шків Ш2 (елемент 7) великою жорсткістю, яку при складанні спрощеної розрахункової схеми можна прийняти рівній нескінченності. Схема, що враховує це спрощення, приведена на рис. 3.3.

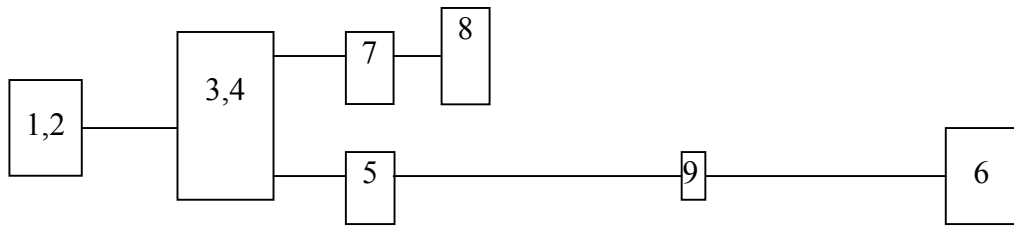


Рисунок 3.2 – Розрахункова схема механічної частини.

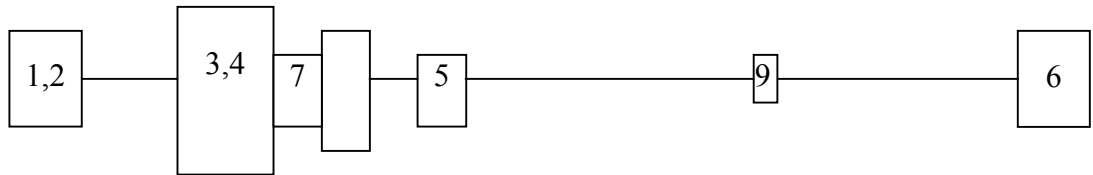


Рисунок 3.3 – Спрощена схема механічної частини

Жорсткості зв'язків між масами такі, що без великої погрішності механічну частину електроприводу можна представити у вигляді трьохмасової пружної системи, показаної на рис. 3.4.

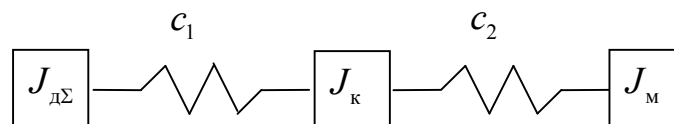


Рисунок 3.4 – Розрахункова трьохмасова схема механічної частини системи

На схемі рис. 3.4 прийняти позначення.

$J_{дΣ} = J_1 + J_2 + J_3 + J_4 + J_7 + J_8 + J_5$ – приведений момент інерції провідної маси;

$J_к = m_{к1} H \rho^2$ – приведений момент інерції каната (другої маси);

$m_{к1}$ – маса 1-го погонного метра каната;

$\rho = \frac{V_{п}}{\omega_{д}} = \frac{D_{\phi}}{2}$ – радіус приведення;

$J_{\text{м}} = m_6 \rho^2$ – приведений момент інерції навантаженого скипа (третя маса);

c_1 – приведені жорсткості ділянок каната між масами; $c_1 = c_2 = 2c_{\text{к}} \rho^2$;

$c_{\text{к}}$ – жорсткість каната, $c_{\text{к}} = \frac{c_{1\text{к}}}{H}$;

$c_{1\text{к}}$ – жорсткість одного метра підйомного каната.

У інженерних розрахунках часто замість трьохмасової системи розглядають двомасову систему (рис. 3.5).



Рисунок 3.5 – Розрахункова двомасова схема
механічної частини системи

На схемі рис. 3.5 позначено:

$$J'_{\text{д}\Sigma} = J_{\text{д}\Sigma} + \frac{J_{\text{к}}}{2}; \quad J'_\text{м} = J_\text{м} + \frac{J_{\text{к}}}{2}; \quad c_{12} = c_{\text{к}} \rho^2.$$

Динамічні процеси в механічній частині підйомної установки, представлену трьохмасовою системою, описуються рівняннями

$$\left\{ \begin{array}{l} J_{\text{д}\Sigma} \frac{d\omega_{\text{д}}}{dt} = k\Phi I_{\text{д}} - M_{\Sigma 1}; \\ J_{\text{к}} \frac{d\omega_{\text{к}}}{dt} = M_{\Sigma 1} - M_{\Sigma 2}; \\ J_{\text{м}} \frac{d\omega_{\text{м}}}{dt} = M_{\Sigma 2} - M_{\text{ст}}; \\ \frac{dM_{\text{np1}}}{dt} = c_1 (\omega_{\text{д}} - \omega_{\text{к}}); \\ \frac{dM_{\text{np2}}}{dt} = c_2 (\omega_{\text{к}} - \omega_{\text{м}}), \end{array} \right.$$

де $M_{\Sigma 1}$, $M_{\Sigma 2}$ – сумарні моменти, що передаються пружними передачами

$$M_{\Sigma 1} = M_{\text{пр}1} + M_{\text{вт}1}, \quad M_{\Sigma 2} = M_{\text{пр}2} + M_{\text{вт}2}$$

$M_{\text{пр}1}$, $M_{\text{пр}2}$ – моменти пружної взаємодії між масами трьохмасової системи;

$M_{\text{вт}1}$, $M_{\text{вт}2}$ – моменти в'язкого тертя

$$M_{\text{вт}1} = \beta_1(\omega_{\text{д}} - \omega_{\text{к}}); \quad M_{\text{вт}2} = \beta_2(\omega_{\text{к}} - \omega_{\text{м}}),$$

β_1 , β_2 – коефіцієнти внутрішнього в'язкого тертя ділянок каната між масами.

Для двомасової системи рівняння руху мають вид:

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

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

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

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

Виконаємо дослідження двомасової систему підлеглого регулювання швидкості двигуна з регулятором швидкості, налаштованим по симетричному критерію з урахуванням контура струму, налаштованим на модульний оптимум.

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

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{м}}}{dt} = \frac{1}{J'_{\text{м}}} M_{\text{пр}} + \frac{\beta}{J'_{\text{м}}} \omega_{\text{д}} - \frac{\beta}{J'_{\text{м}}} \omega_{\text{м}} - \frac{1}{J'_{\text{м}}} M_{\text{ст}}; \\ \frac{dM_{\text{пр}}}{dt} = c_{12} \omega_{\text{д}} - c_{12} \omega_{\text{м}}; \\ \frac{d\omega_{\text{д}}}{dt} = \frac{k\Phi_{\text{н}}}{J'_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J'_{\text{д}\Sigma}} M_{\text{пр}} - \frac{\beta}{J'_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J'_{\text{д}\Sigma}} \omega_{\text{м}}. \end{array} \right. \quad (3.3)$$

Розглянемо алгоритмічну схему одномасової системи, представлену в попередньому розділі, і запишемо рівняння стану системи регулювання, не враховуючи пружні властивості елементів механічної частини. Потім, доповнивши її рівняннями для двомасової системи, отримаємо систему диференціальних рівнянь стану, що враховує пружний зв'язок у системі

Рівняння стану одномасової системи:

$$\left\{ \begin{array}{l} \frac{d\omega_{\text{д}}}{dt} = \frac{k\Phi_{\text{н}}}{J_{\text{д}\Sigma}} I_{\text{д}} - \frac{1}{J_{\text{д}\Sigma}} M_{\text{ст}}; \\ \frac{dI_{\text{д}}}{dt} = -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{г}}; \\ \frac{dU_{\text{г}}}{dt} = -\frac{1}{T_{\text{г}}} U_{\text{г}} + \frac{k_{\text{г}}}{T_{\text{г}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} = -\frac{k_{\text{ш}}k_{\text{пш}}k_{\text{пт}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} \omega_{\text{д}} - \frac{k_{\text{т}}k_{\text{пт}}k_{\text{пш}}k_{\text{тп}}}{T_{\text{мн}}} I_{\text{д}} - \frac{k_{\text{н}}k_{\text{пш}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{г}} - \frac{1}{T_{\text{мн}}} U_{\text{тп}} - \frac{k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рн}} - \\ - \frac{k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рт}} + \frac{k_{\text{пн}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рш}} + \frac{k_{\text{пш}}k_{\text{пт}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{з}}; \\ \frac{dU_{\text{рн}}}{dt} = -k_{\text{ш}}k_{\text{пш}}k_{\text{пт}}k_{\text{ін}} \omega_{\text{д}} - k_{\text{т}}k_{\text{пт}}k_{\text{ін}} I_{\text{д}} - k_{\text{н}}k_{\text{ін}} U_{\text{г}} + k_{\text{ін}} U_{\text{рт}} + k_{\text{пт}}k_{\text{ін}} U_{\text{рш}} + k_{\text{пш}}k_{\text{пт}}k_{\text{ін}} U_{\text{з}}; \\ \frac{dU_{\text{рт}}}{dt} = -k_{\text{ш}}k_{\text{пш}}k_{\text{іт}} \omega_{\text{д}} - k_{\text{іт}}k_{\text{т}} I_{\text{д}} + k_{\text{іт}} U_{\text{рш}} + k_{\text{пш}}k_{\text{іт}} U_{\text{з}}; \\ \frac{dU_{\text{рш}}}{dt} = -k_{\text{ш}}k_{\text{іш}} \omega_{\text{д}} + k_{\text{іш}} U_{\text{з}}. \end{array} \right. \quad (3.4)$$

При записі системи (3.4) прийняті наступні позначення.

$U_{\Gamma}(t)$ $U_{\text{тп}}(t)$ – напруга генератора і тиристорного перетворювача;

$U_{\text{рн}}(t)$, $U_{\text{рт}}(t)$, $U_{\text{рш}}(t)$ – напруги на виході інтегральної частини ПІ регулятора напруги, струму і швидкості відповідно.

$k_{\text{пш}}$, $k_{\text{іш}}$, $k_{\text{пт}}$, $k_{\text{іт}}$, $k_{\text{пн}}$, $k_{\text{ін}}$ - коефіцієнти підсилення пропорційної і інтегральної частин регулятора швидкості току і напруги відповідно; значення указаних параметрів можуть бути отримані з виразів передатних функцій регуляторів:

$$W_{\text{рш}}(p) = \frac{T_{\text{м}} k_{\text{т}}}{8T_{\text{мш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}} \frac{1 + 4T_{\text{мш}} p}{p} = \frac{4T_{\text{мш}} T_{\text{м}} k_{\text{т}}}{8T_{\text{мш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}} + \frac{T_{\text{м}} k_{\text{т}}}{8T_{\text{мш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}} \frac{1}{p} = k_{\text{пш}} + \frac{k_{\text{іш}}}{p},$$

$$k_{\text{пш}} = \frac{4T_{\text{мш}} T_{\text{м}} k_{\text{т}}}{8T_{\text{мш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}}, \quad k_{\text{іш}} = \frac{T_{\text{м}} k_{\text{т}}}{8T_{\text{мш}}^2 k_{\text{ш}} k_{\text{д}} R_{\Sigma}},$$

$$W_{\text{рт}}(p) = \frac{k_{\text{н}} \cdot R_{\Sigma}}{2T_{\text{мт}} k_{\text{т}}} \cdot \frac{T_{\text{е}} p + 1}{p} = \frac{k_{\text{н}} R_{\Sigma} T_{\text{е}}}{2T_{\text{мт}} k_{\text{т}}} + \frac{k_{\text{н}} R_{\Sigma}}{2T_{\text{мт}} k_{\text{т}}} \frac{1}{p} = k_{\text{пт}} + \frac{k_{\text{іт}}}{p},$$

$$k_{\text{пт}} = \frac{k_{\text{н}} R_{\Sigma} T_{\text{е}}}{2T_{\text{мт}} k_{\text{т}}}, \quad k_{\text{іт}} = \frac{k_{\text{н}} R_{\Sigma}}{2T_{\text{мт}} k_{\text{т}}},$$

$$W_{\text{рн}}(p) = \frac{1}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}} \frac{T_{\text{г}} p + 1}{p} = \frac{T_{\text{г}}}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}} + \frac{1}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}} \frac{1}{p} = k_{\text{пн}} + \frac{k_{\text{ін}}}{p},$$

$$k_{\text{пн}} = \frac{T_{\text{г}}}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}}, \quad k_{\text{ін}} = \frac{1}{2T_{\text{мн}} k_{\text{г}} k_{\text{тп}} k_{\text{н}}},$$

При записі системи (3.4) також замість $T_{\text{м}}$ і $k_{\text{д}}$ записані відповідні вирази

$$T_{\text{м}} = J_{\text{д}\Sigma} \cdot \frac{R_{\text{я}\Sigma}}{(k\Phi_{\text{н}})^2}; \quad k_{\text{д}} = \frac{1}{k\Phi_{\text{н}}}.$$

Постійними часу датчиків напруги $T_{\text{дн}}$, струму $T_{\text{дт}}$ і швидкості $T_{\text{дш}}$ нехтуємо.

Об'єднаємо системи (3.3) і (3.3), одержимо наступну систему диференціальних рівнянь двомасової системи:

$$\left\{ \begin{aligned} \frac{d\omega_{\text{м}}}{dt} &= \frac{1}{J'_{\text{м}}} M_{\text{пп}} + \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{пп}}}{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{пп}} - \frac{\beta}{J'_{\text{д}\Sigma}} \omega_{\text{д}} + \frac{\beta}{J'_{\text{д}\Sigma}} \omega_{\text{м}}; \\ \frac{dI_{\text{д}}}{dt} &= -\frac{k\Phi_{\text{н}}}{R_{\Sigma}T_{\text{е}}} \omega_{\text{д}} - \frac{1}{T_{\text{е}}} I_{\text{д}} + \frac{1}{R_{\Sigma}T_{\text{е}}} U_{\text{г}}; \\ \frac{dU_{\text{г}}}{dt} &= -\frac{1}{T_{\text{г}}} U_{\text{г}} + \frac{k_{\text{г}}}{T_{\text{г}}} U_{\text{тп}}; \\ \frac{dU_{\text{тп}}}{dt} &= -\frac{k_{\text{ш}}k_{\text{пш}}k_{\text{пт}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} \omega_{\text{д}} - \frac{k_{\text{т}}k_{\text{пт}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} I_{\text{д}} - \frac{k_{\text{н}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{г}} - \frac{1}{T_{\text{мн}}} U_{\text{тп}} - \frac{k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рн}} - \\ &\quad - \frac{k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рт}} + \frac{k_{\text{пн}}k_{\text{пт}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{рш}} + \frac{k_{\text{пш}}k_{\text{пт}}k_{\text{пн}}k_{\text{тп}}}{T_{\text{мн}}} U_{\text{з}}; \\ \frac{dU_{\text{рн}}}{dt} &= -k_{\text{ш}}k_{\text{пш}}k_{\text{пт}}k_{\text{ін}} \omega_{\text{д}} - k_{\text{т}}k_{\text{пт}}k_{\text{ін}} I_{\text{д}} - k_{\text{н}}k_{\text{ін}} U_{\text{г}} + k_{\text{ін}} U_{\text{рт}} + k_{\text{пт}}k_{\text{ін}} U_{\text{рш}} + k_{\text{пш}}k_{\text{пт}}k_{\text{ін}} U_{\text{з}}; \\ \frac{dU_{\text{рт}}}{dt} &= -k_{\text{ш}}k_{\text{пш}}k_{\text{іт}} \omega_{\text{д}} - k_{\text{іт}}k_{\text{т}} I_{\text{д}} + k_{\text{іт}} U_{\text{рш}} + k_{\text{пш}}k_{\text{іт}} U_{\text{з}}; \\ \frac{dU_{\text{рш}}}{dt} &= -k_{\text{ш}}k_{\text{іш}} \omega_{\text{д}} + k_{\text{іш}} U_{\text{з}}. \end{aligned} \right.$$

Алгоритмічна схема двомасової системи показана на рис. 3.6.

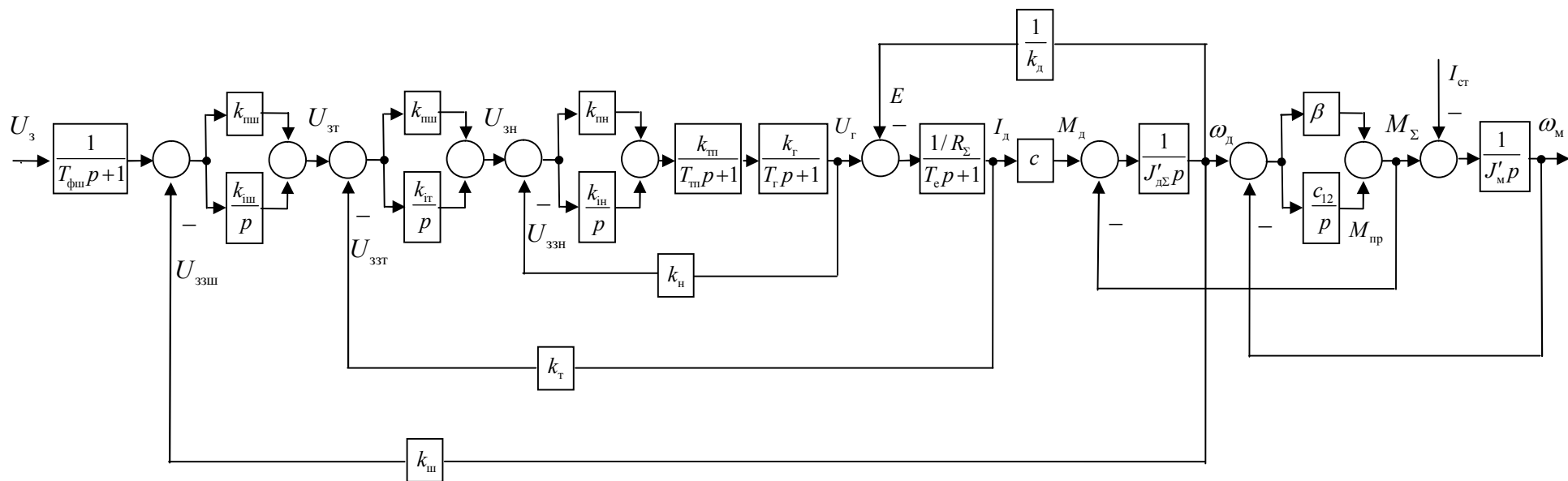


Рисунок 3.6 – Алгоритмічна схема двомасової системи

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

Введемо вектор стану двомасової системи:

$$\vec{X}(t) = [\omega_m(t), M_{\text{пр}}(t), \omega_d(t), I_d(t), U_r(t), U_{\text{тп}}(t), U_{\text{рн}}(t), U_{\text{рт}}(t), U_{\text{рш}}(t)]^T,$$

і вектор зовнішніх дій

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

Запишемо рівняння стан системи, доповнивши його рівнянням виходу системи

$$\begin{cases} \dot{\vec{X}}(t) = \mathbf{A}\vec{X}(t) + \mathbf{B}\vec{U}(t); \\ \vec{Y}(t) = \mathbf{C}\vec{X}(t), \end{cases} \quad (3.5)$$

де $\mathbf{A}$ – матриця стану;

$\mathbf{B}$ – матриця управління;

$\mathbf{C}$ – матриця виходу;

$\vec{Y}(t)$ – вектор виходу.

Матриці стану $\mathbf{A}$ і управління $\mathbf{B}$ для двомасової приведені нижче.

3.2 Розрахунок перехідних процесів двомасової системи

Визначення чисельних значень приведених моментів інерції виконуємо таким чином. У попередніх розділах були визначені наступні величини.

У попередньому розділі були визначені наступні величини: приведений до валу двигуна момент інерції механізму (завантаженого скипа) $J_m = J_b = 49250 \text{ кг} \cdot \text{м}^2$, приведений до валу двигуна момент інерції підйомного каната $J_k = 21028 \text{ кг} \cdot \text{м}^2$, момент інерції двигуна і інших елементів механічної частини електроприводу, що обертаються із швидкістю двигуна $J_{\Sigma_d} = 161584 \text{ кг} \cdot \text{м}^2$.

Сумарні приведені моменти інерції двомасової системи:

$$J'_M = J_M + \frac{J_K}{2},$$

$$J'_M = 49250 + \frac{21028}{2} = 59764 \text{ кг} \cdot \text{м}^2$$

$$J'_{d\Sigma} = J_{d\Sigma} - \frac{J_K}{2}$$

$$J'_{d\Sigma} = 161584 + \frac{21028}{2} = 172098 \text{ кг} \cdot \text{м}^2$$

Жорсткість каната визначаємо із співвідношення

$$\omega_0 = \sqrt{\frac{c_K}{m'_2}}$$

отже

$$c_K = \omega_0^2 m'_2$$

де ω_0 – частота власних коливань підйомної судини; з літератури, наприклад [29], відомо, що ω_0 знаходиться в діапазоні $(1 \div 6) \text{ с}^{-1}$. Приймаємо $\omega_0 = 3 \text{ с}^{-1}$;

m'_2 - відносна приведена маса скипа і половини каната (тобто друга маса двомасової системи):

$$m'_2 = \left(\mu_2 + \frac{\mu_k}{2} \right) \cdot m,$$

$$m'_2 = \left(0,175 + \frac{0,095}{2} \right) \cdot 56000 = 12460 \text{ кг}.$$

$$c_K = 3^2 \cdot 12460 = 112140 \text{ Н/м}$$

Приведена жорсткість підйомного каната

$$c_{12} = c_k \rho^2,$$

$$c_{12} = 112140 \cdot 2,5^2 = 700875 \text{ Нм}.$$

Приведені жорсткості ділянок каната між масами для трьохмасової системи

$$c_1 = c_2 = 2c_{12} = 2 \cdot 700875 = 1401750 \text{ Нм}.$$

Для дослідження динамічних характеристик двомасової системи використовуємо пакет прикладних програм MATLAB. Розрахунок виконуємо як за допомогою моделі, зображеної на рис.3.7, так і по векторний-матричному рівнянню (3.5).

Для зменшення інформації, що виводиться, на екран монітора виводимо перші шість змінних стану системи по задаючій і обурюючій дії. Відповідні графіки приведені на рис.3.8 -3.11. Як видно з графіків, перехідні процеси мають характер слабозатухаючих коливань.

Текст файлу, що містить початкові дані, матриці **A**, **B**, **C** приведений в додатку Д.

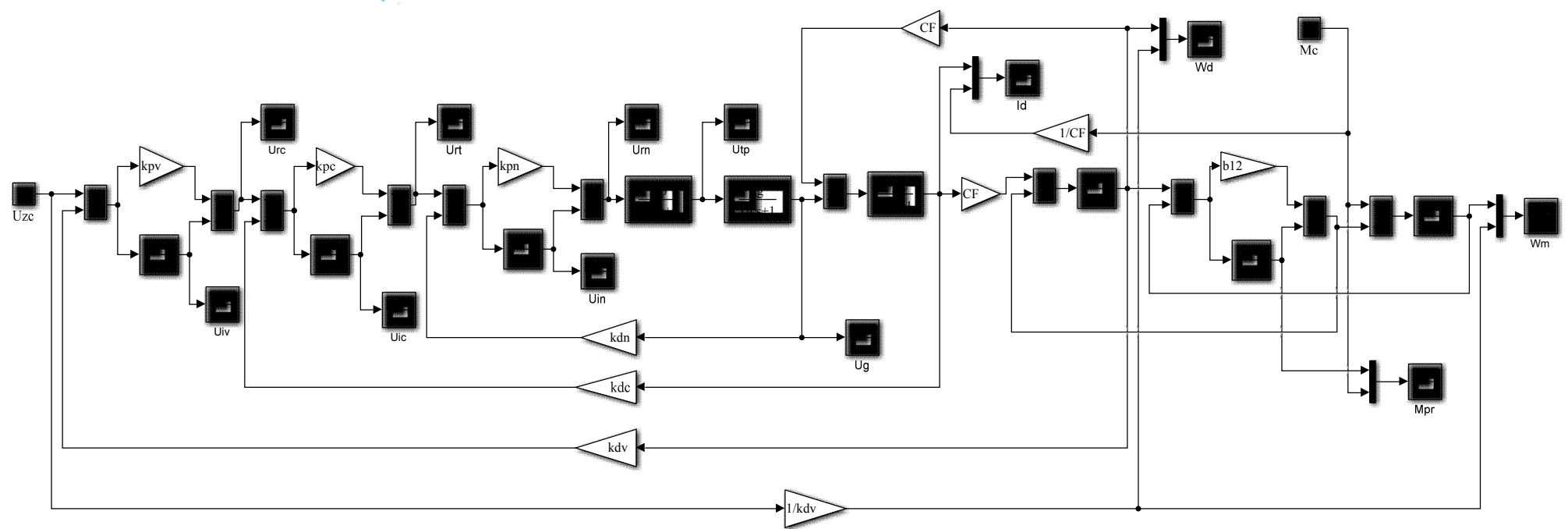


Рисунок 3.7 – Схема моделі двомасової системи

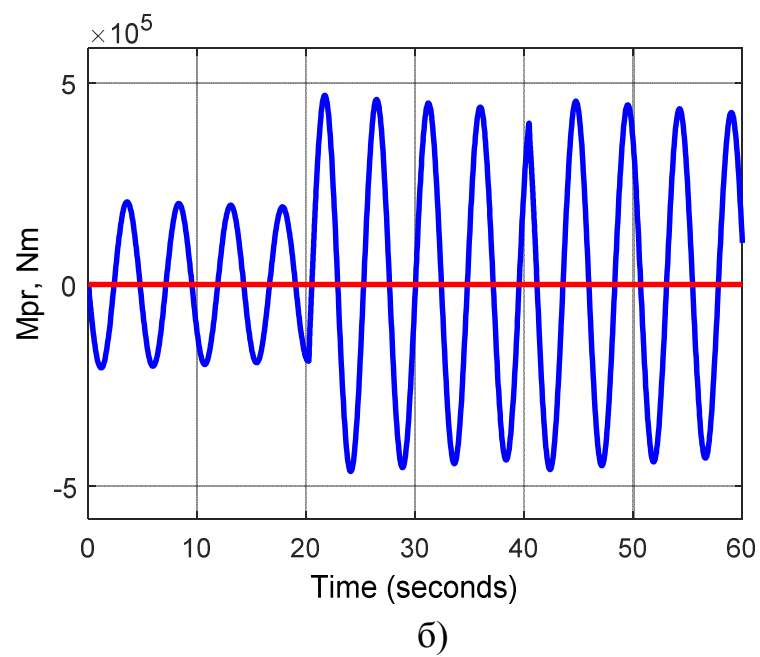
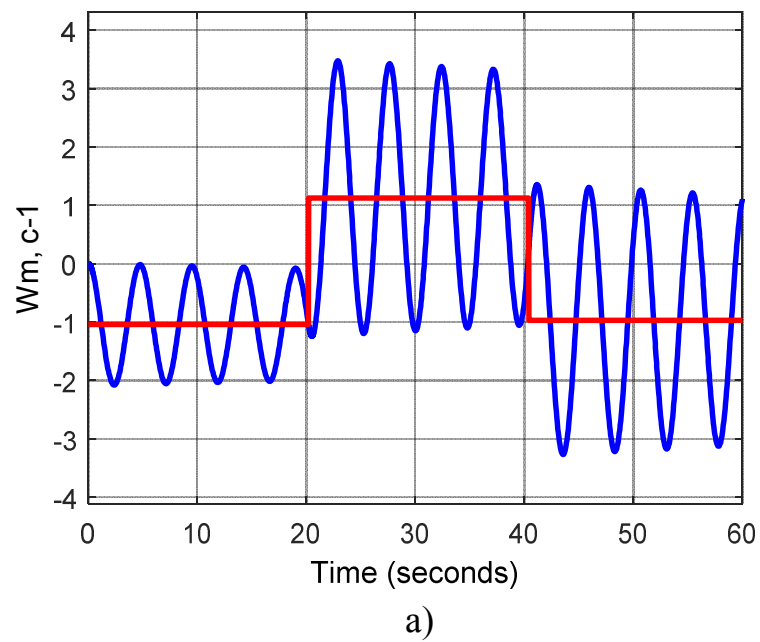


Рисунок 3.8 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по задаючій дії

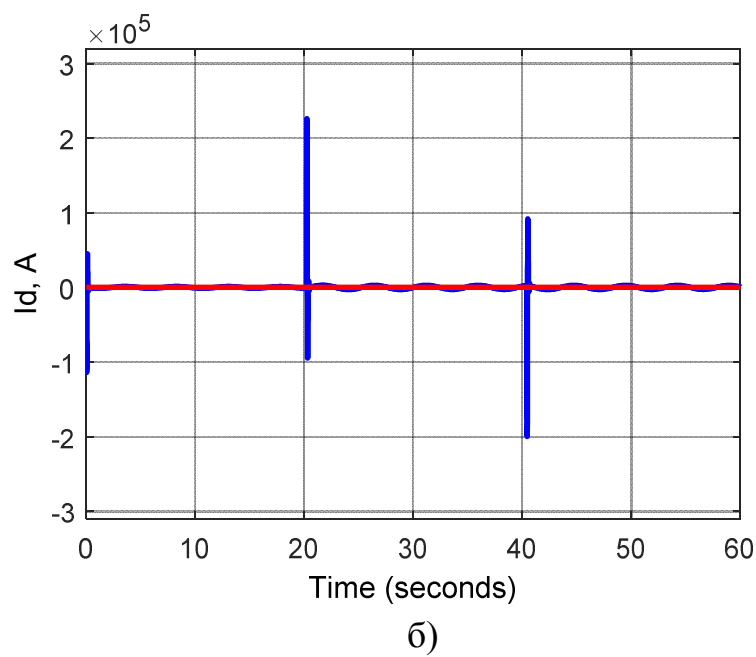
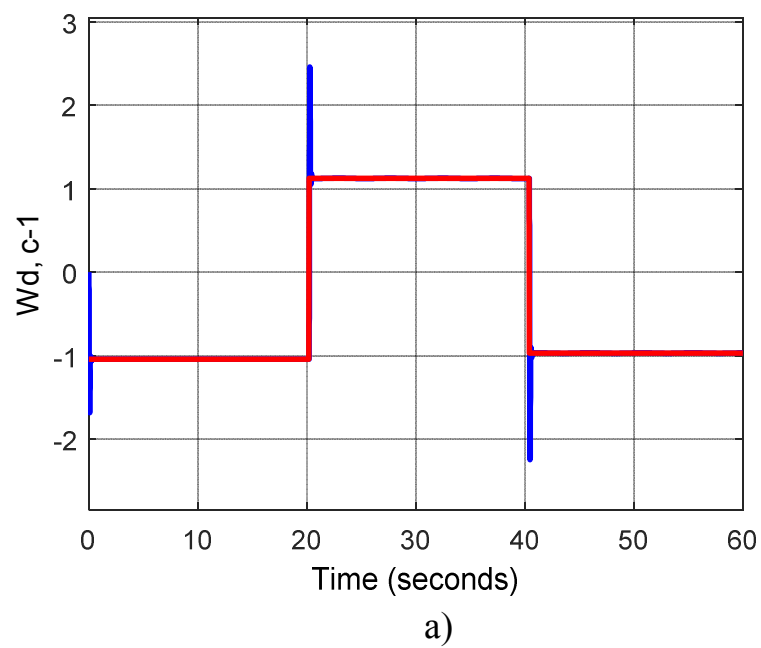
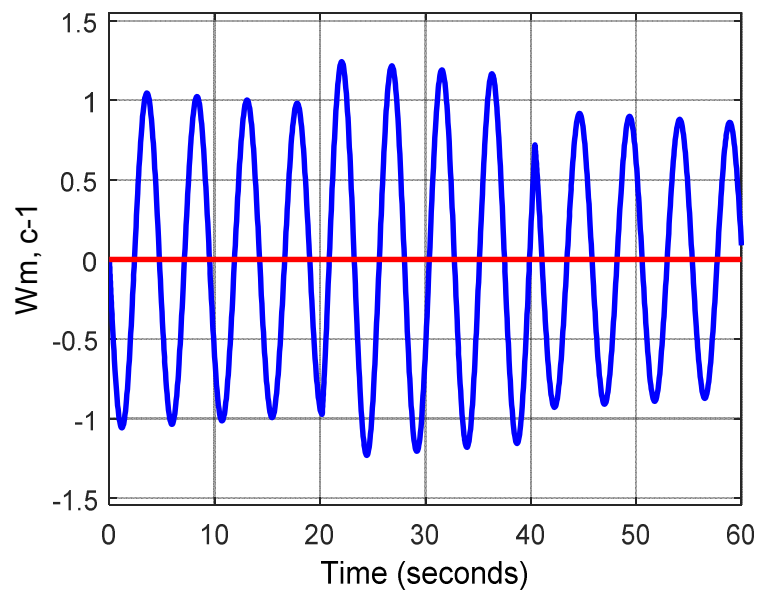
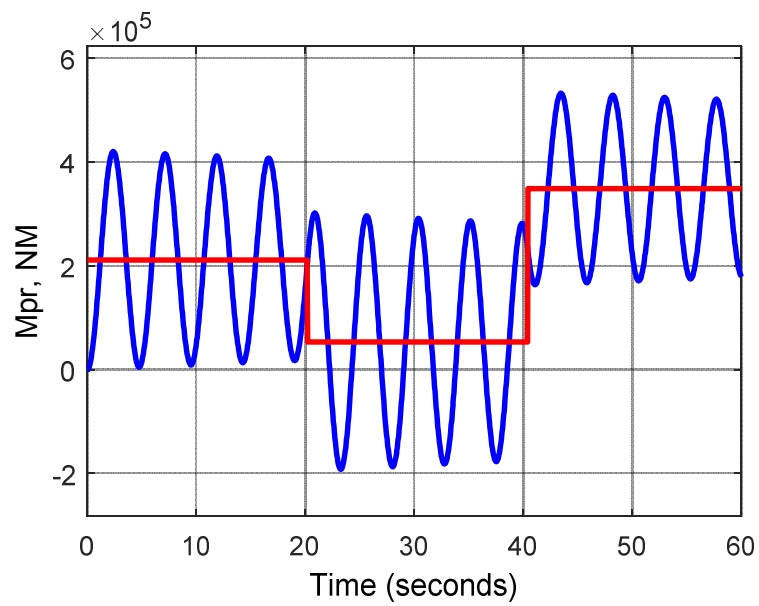


Рисунок 3.9 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по задаючій дії



a)



б)

Рисунок 3.10 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи по обурюючій дії

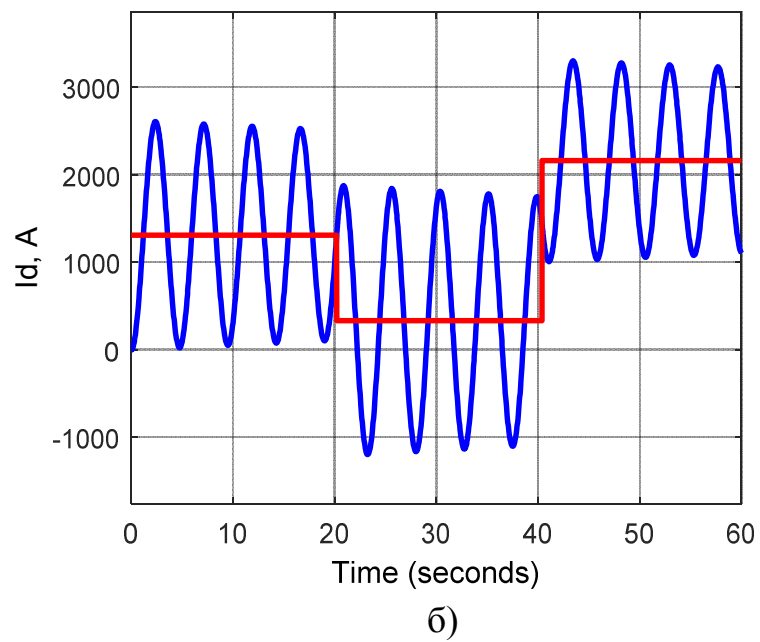
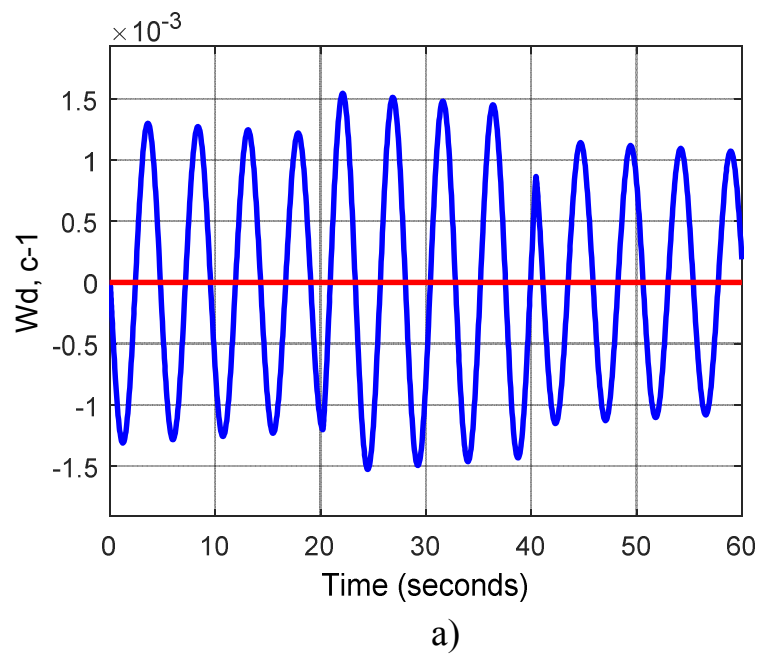


Рисунок 3.11 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи по обурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 3

1. Проведено аналіз роботи механізму підйому шахтної підйимальної установки з урахуванням пружних властивостей підйимального канату. Показано, що розрахункову схему механічної підйомної установки можна представити двома приведеними зосередженими масами сполученими між собою ідеально пружним стрижнем. Однак методика розрахунку такої схеми достатньо складна. В цілях спрощення запропоновано розглядати механічну частину підйомної установки як трьохмасову систему з представленням підйомного каната зосередженою масою, пов'язаною з двома іншими масами еквівалентними невагомими пружними зв'язками, а також як двомасову.

2. Виконано розробку математичної моделі двомасової системи. Отримано рівняння стану двомасової системи і рівняння стану двомасової системи у векторно-матричній формі, визначені елементи вектора стану, тобто змінні стану системи; визначений вектор зовнішніх дій, а також матриці стану **A** і управління **B**. Розроблена алгоритмічна схема двомасової системи.

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

РОЗДІЛ 4

СИНТЕЗ НЕЙРОМЕРЕЖЕВОЇ СИСТЕМИ УПРАВЛІННЯ

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

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

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

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

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

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

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

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

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

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

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

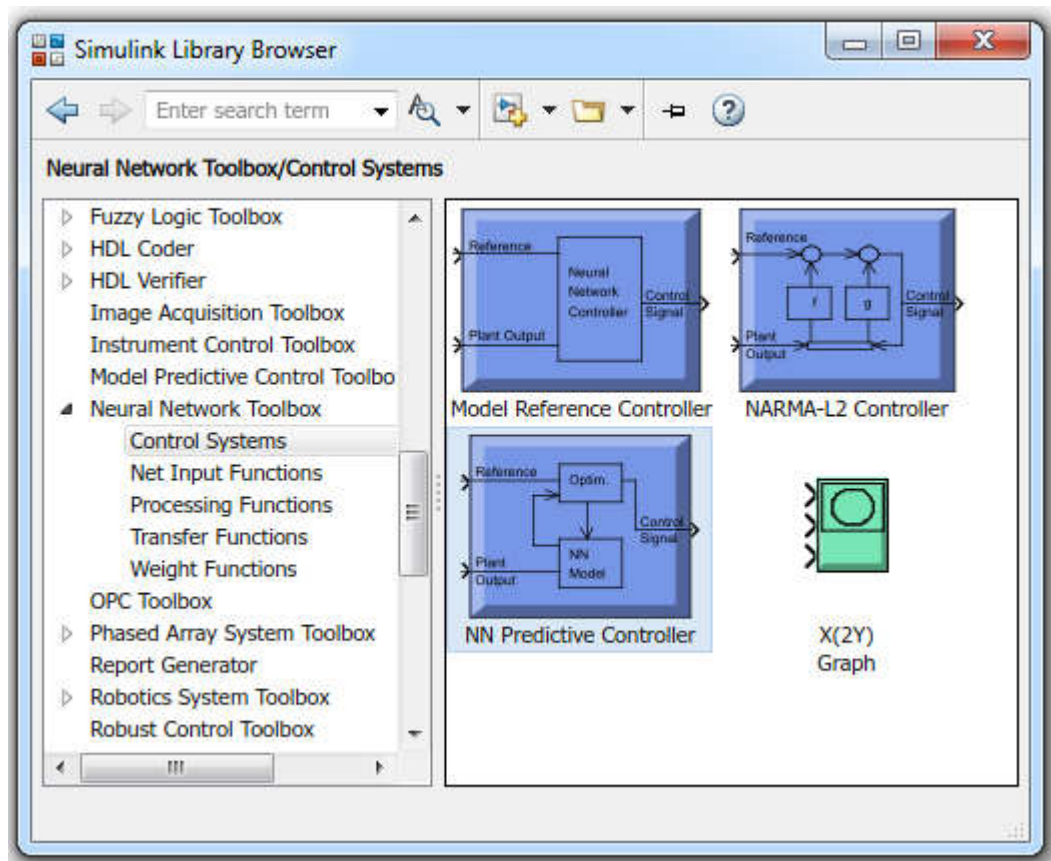


Рисунок 4.1 – Блоки Neural Network Blocksets розділу Control Systems системи SIMULINK

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

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

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

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

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

4.2 Принцип побудови регулятора з прогнозом NN Predictive Controller

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

4.2.1 Ідентифікація об'єкту управління.

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

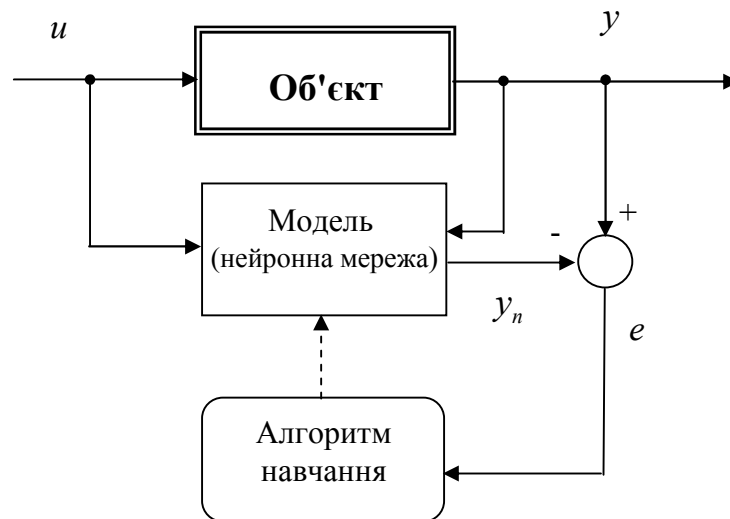


Рисунок 4.2 – Схема підсистеми ідентифікації.

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

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

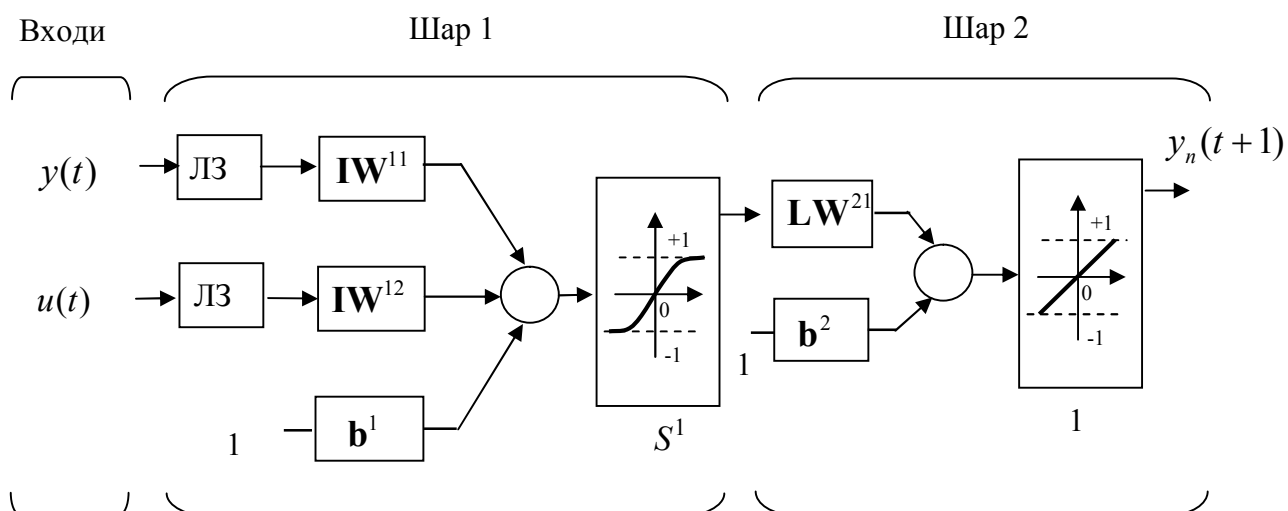


Рисунок 4.3 – Нейронна мережа регулятора керованого об'єкту

4.2.2 Схема управління з прогнозом.

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

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

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

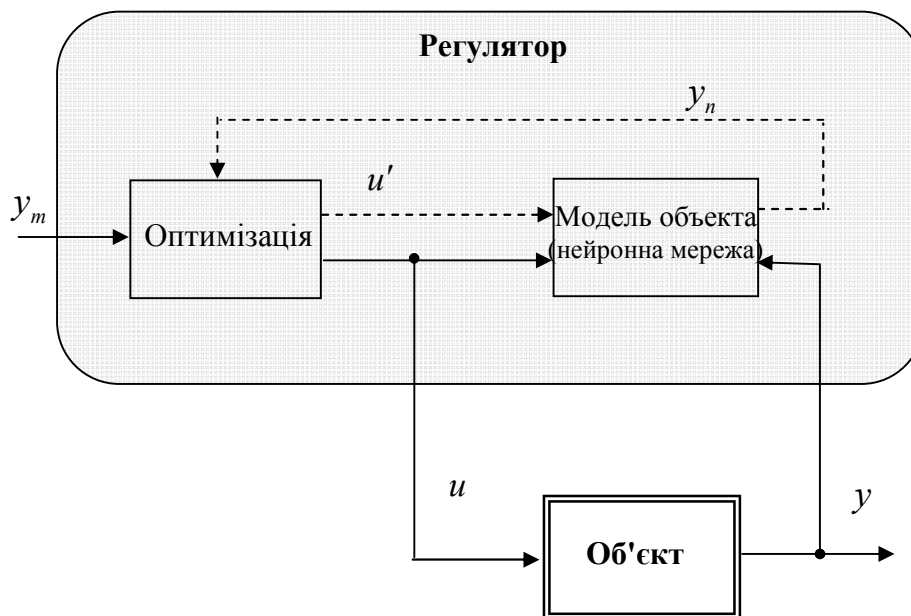


Рисунок 4.4 – Структурна схема системи з регулятором, що використовує принцип прогнозу

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

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

Функції одновимірної оптимізації:

Csrchbac – пошук із зворотним прогоном;

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

Csrchcha – метод кубічної інтерполяції Чараламбуса;

Csrchgol – метод золотого перетину;

Csrchhyb – гібридний метод бісекції і кубічної інтерполяції.

Функції для синтезу управління з прогнозом:

Calcjjdjj – обчислення функціонала якості і його градієнта;

Predopt – оптимізація регулятора з прогнозом;

Dyduvar – обчислення приватних похідних виходу по входу.

Моделі Simulink.

Predcstr – GUI – додаток для регулятора з прогнозом;

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

Допоміжні функції:

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

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

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

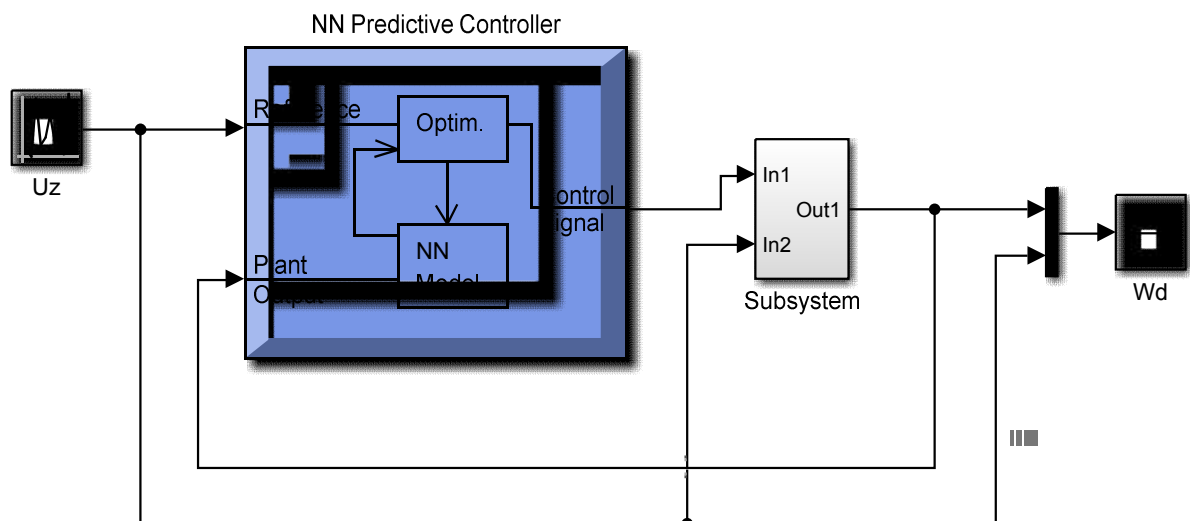


Рисунок 4.5 – Схема системи управління з нейрорегулятором

NN Predictive Controller

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

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

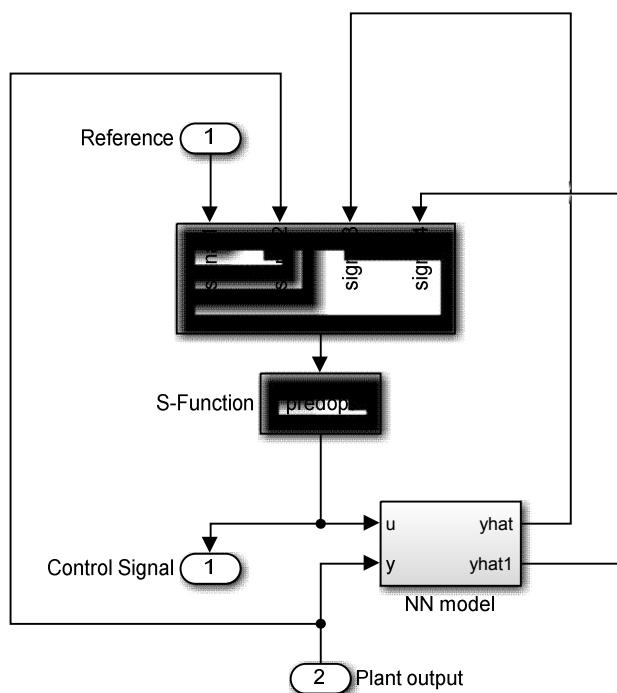


Рисунок 4.6 – Структурна схема нейрорегулятора

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

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

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

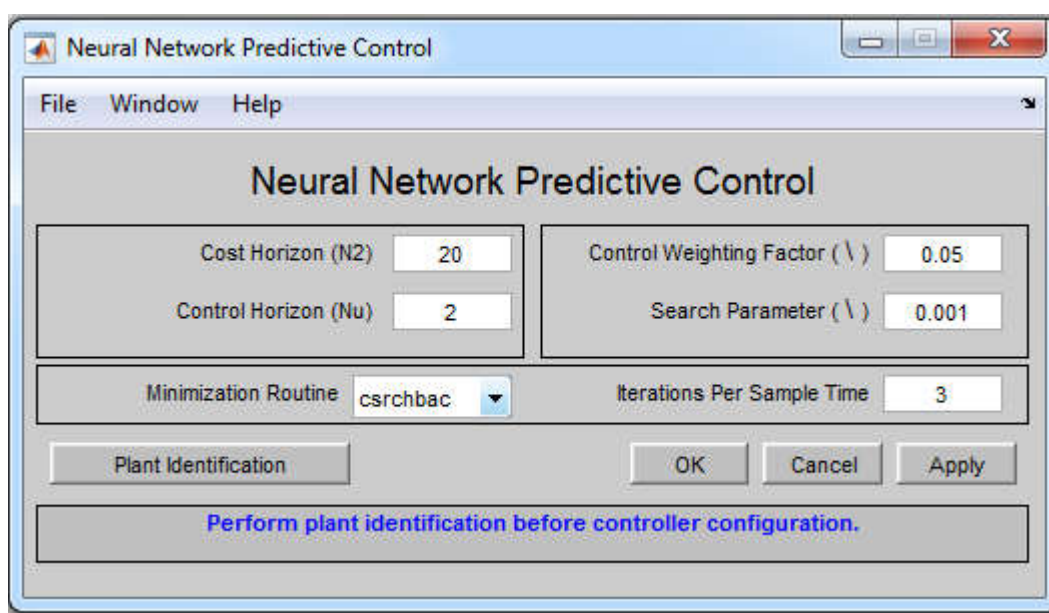


Рисунок 4.7 – Вікно завдання параметрів регулятора

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

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

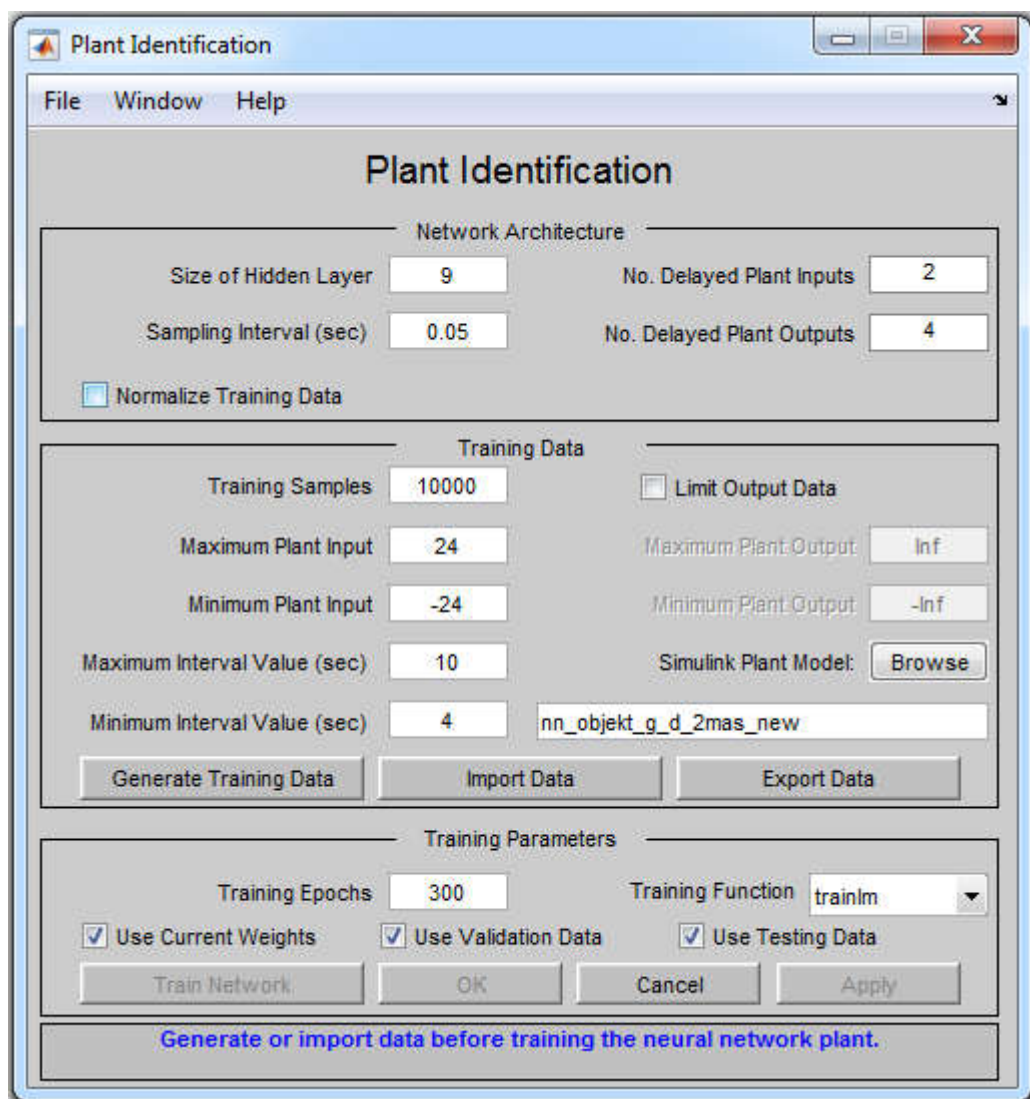


Рисунок 4.8 – Вікно ідентифікації керованої системи

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

Опис вікна Plant Identification

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Maximum Plant Output. Максимальне значення вихідного сигналу.

Minimum Plant Output. Мінімальне значення вихідного сигналу.

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

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

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

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

За допомогою кнопки **Browser** вибирається модель Simulink об'єкту управління; в даній роботі це схема моделі, показана на рис. 4.9 або рис.4.10

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

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

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

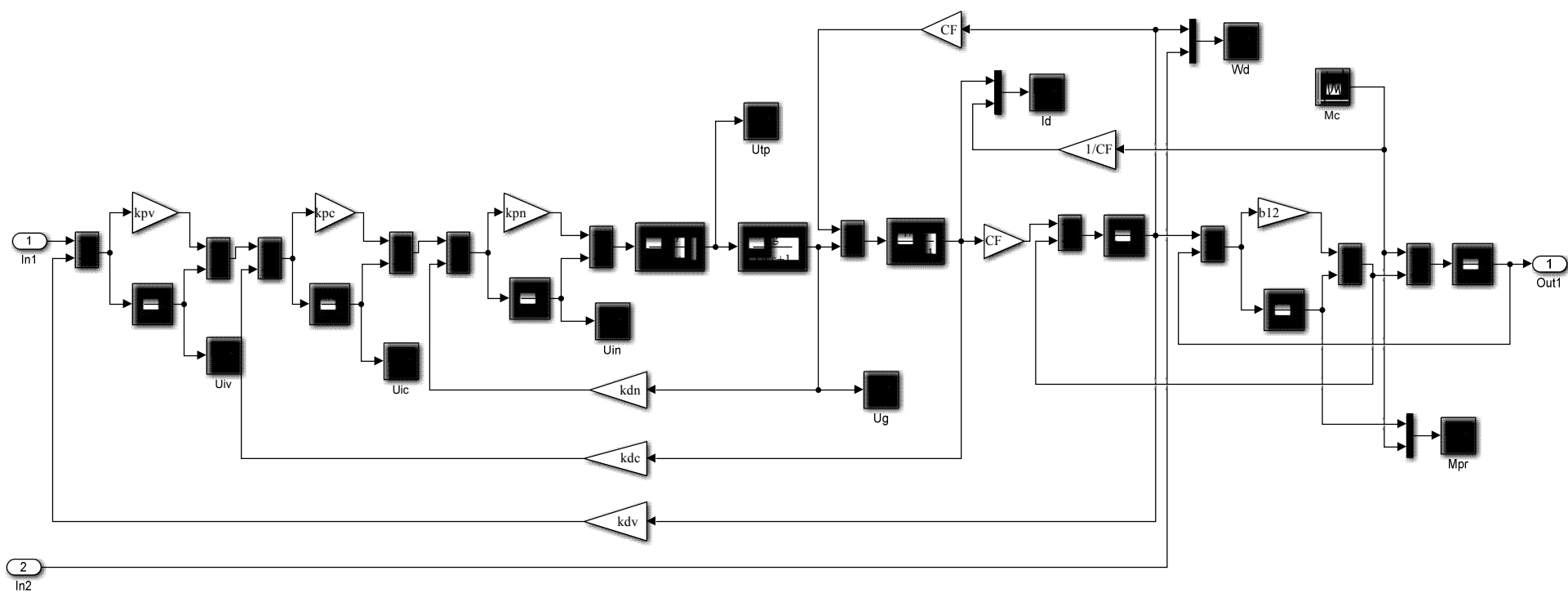


Рисунок 4.9 – Модель об'єкту управління нейрорегулятора

Секція Training Parameters (параметри навчання). Training Parameters

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

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

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

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

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

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

найменших квадратів Левенберга-Марквардта; trainbr - байєсівський регулятор.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

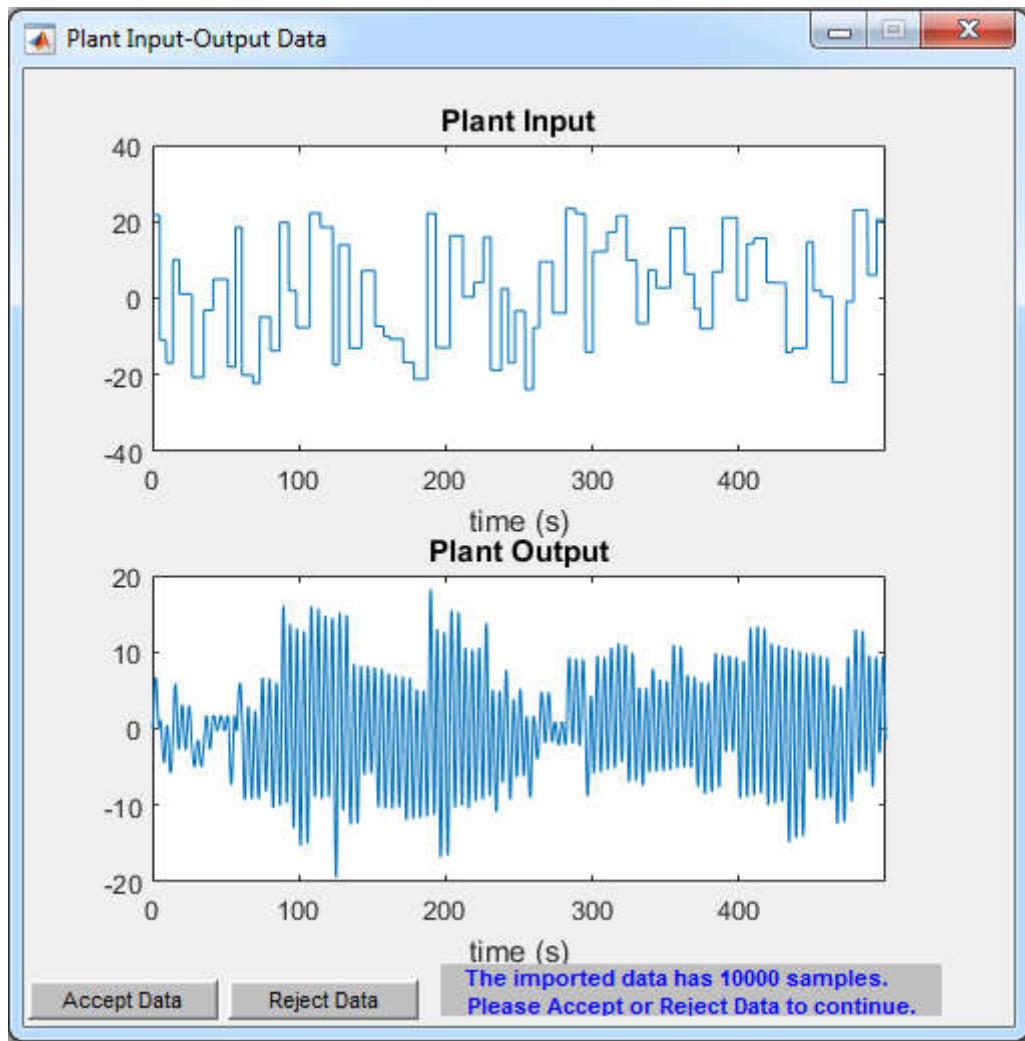


Рисунок 4.10 – Графіки вхідного і вихідного сигналів при генерації навчальної послідовності

Якщо дані прийняті, програма 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).

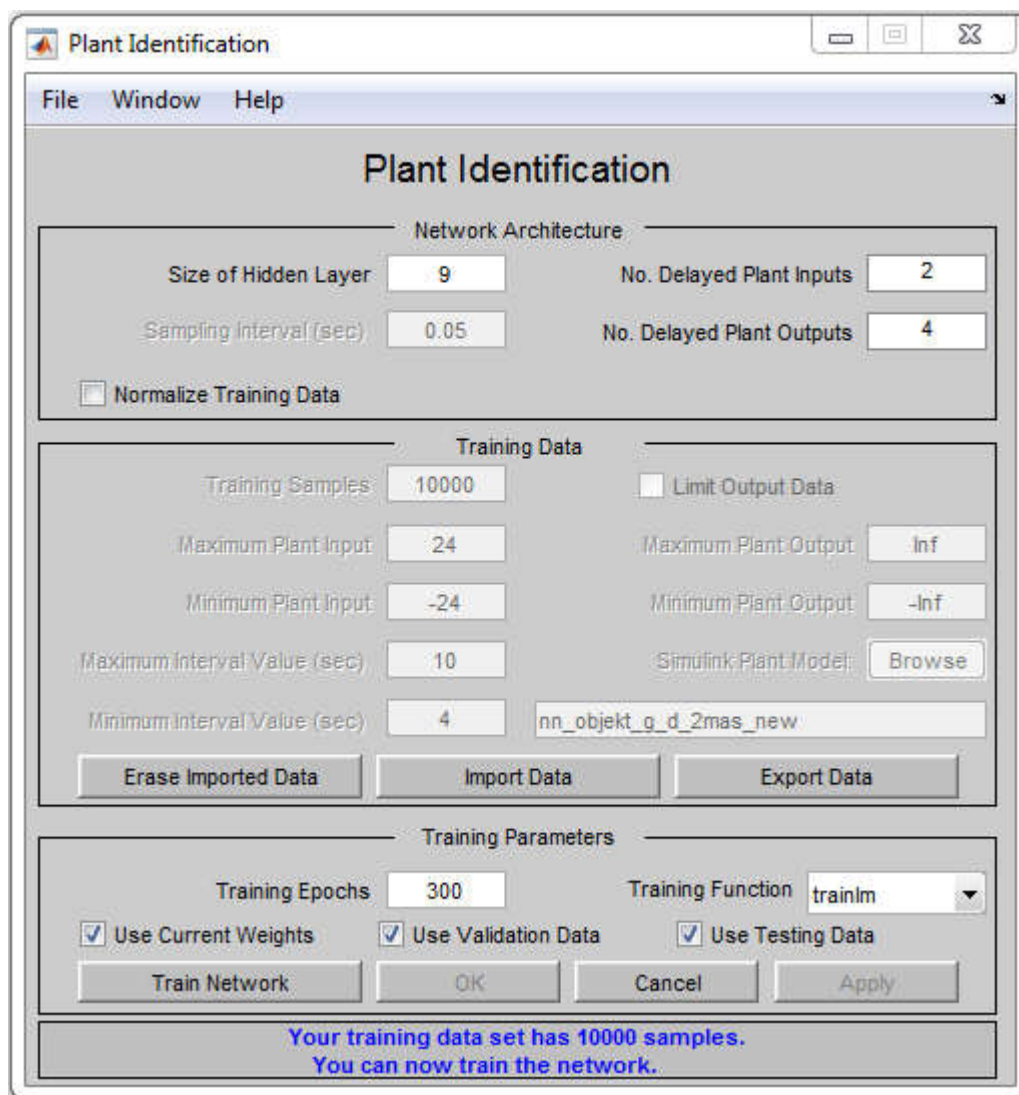


Рисунок 4.11 – Вид вікна Plant Identification після генерації навчальної послідовності

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.

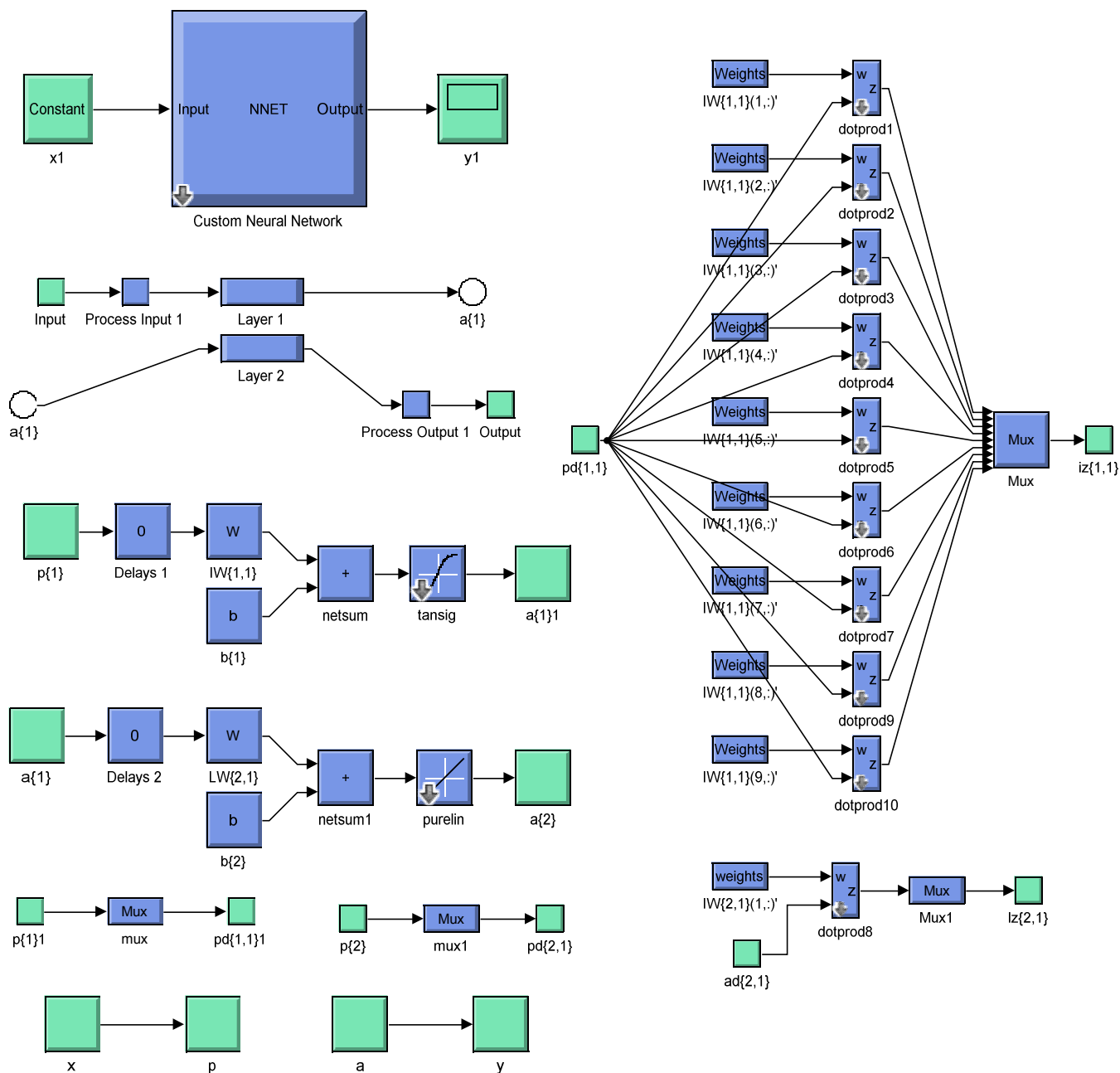


Рисунок 4.12 – Моделі елементів мережі з прямою передачею сигналу, реалізовані в Simulink

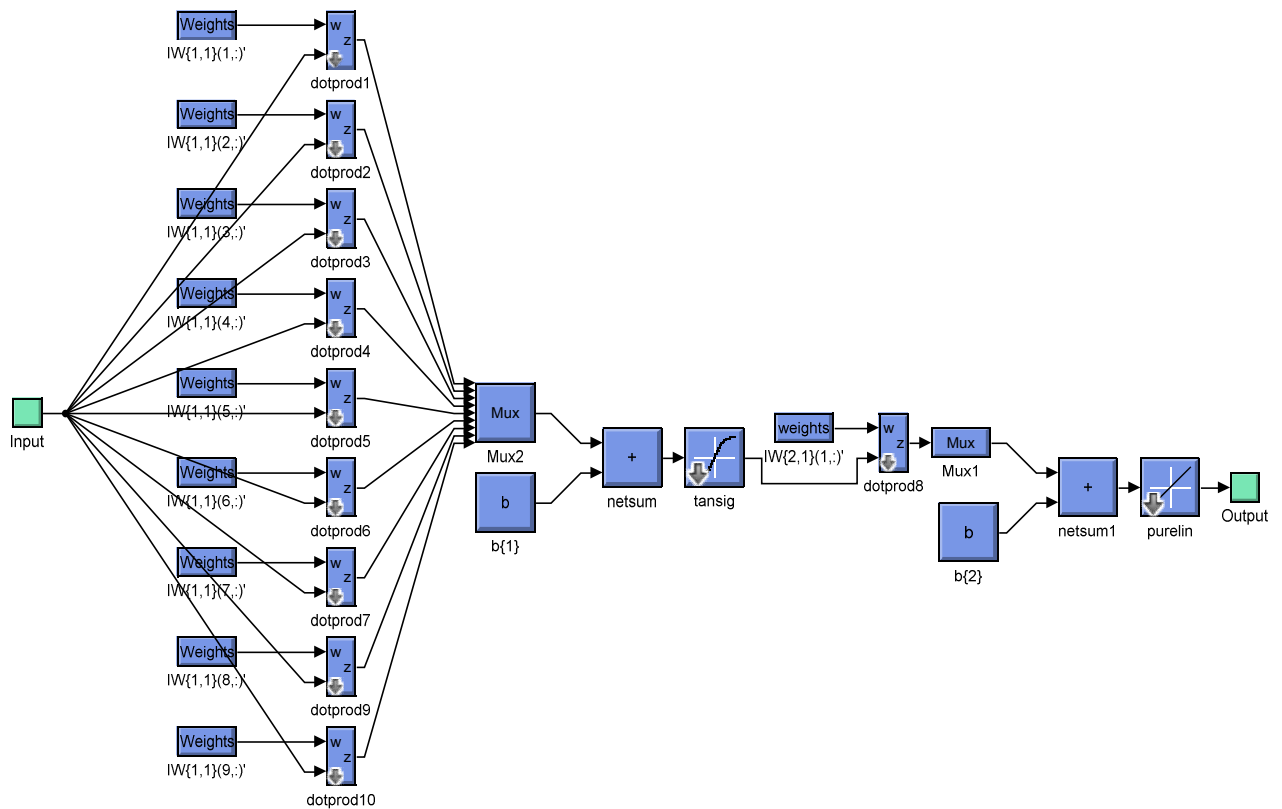


Рисунок 4.13 – Модель статичної мережі з прямою передачею сигналу, побудована в Simulink

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.

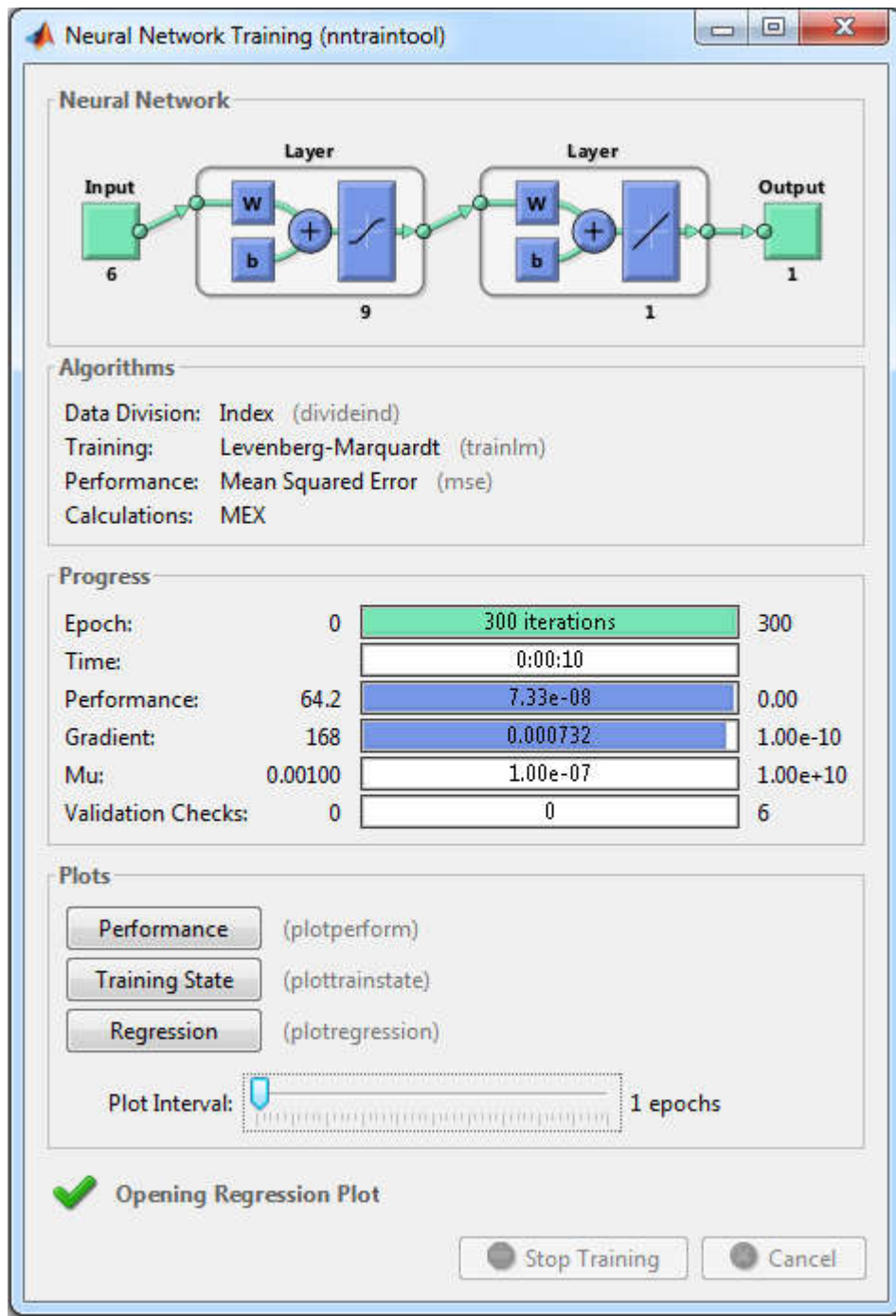


Рисунок 4.14 - Головне вікно, що відображає хід навчання та результати навчання нейронної мережі

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.

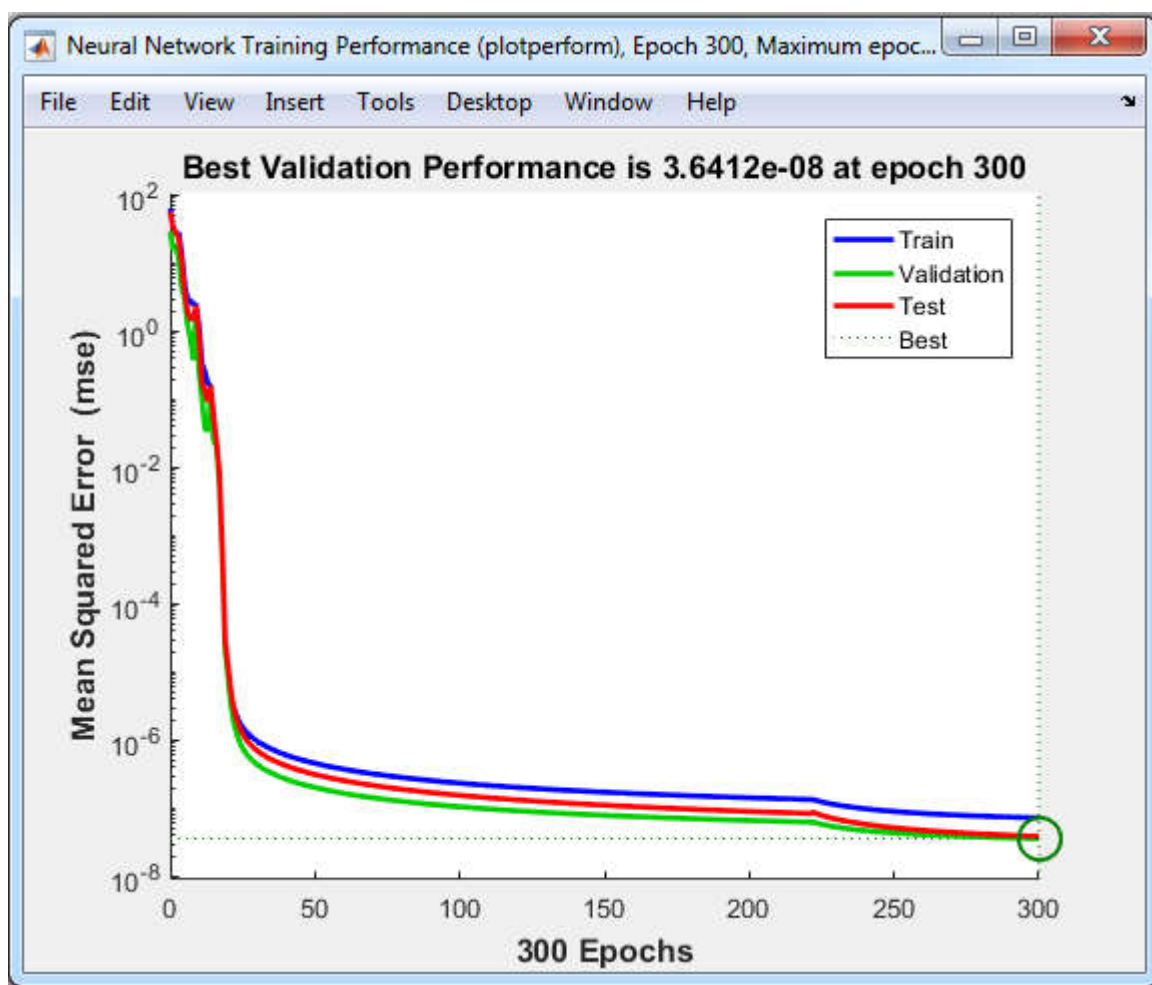


Рисунок 4.15 – Вікно контролю процесу навчання

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.

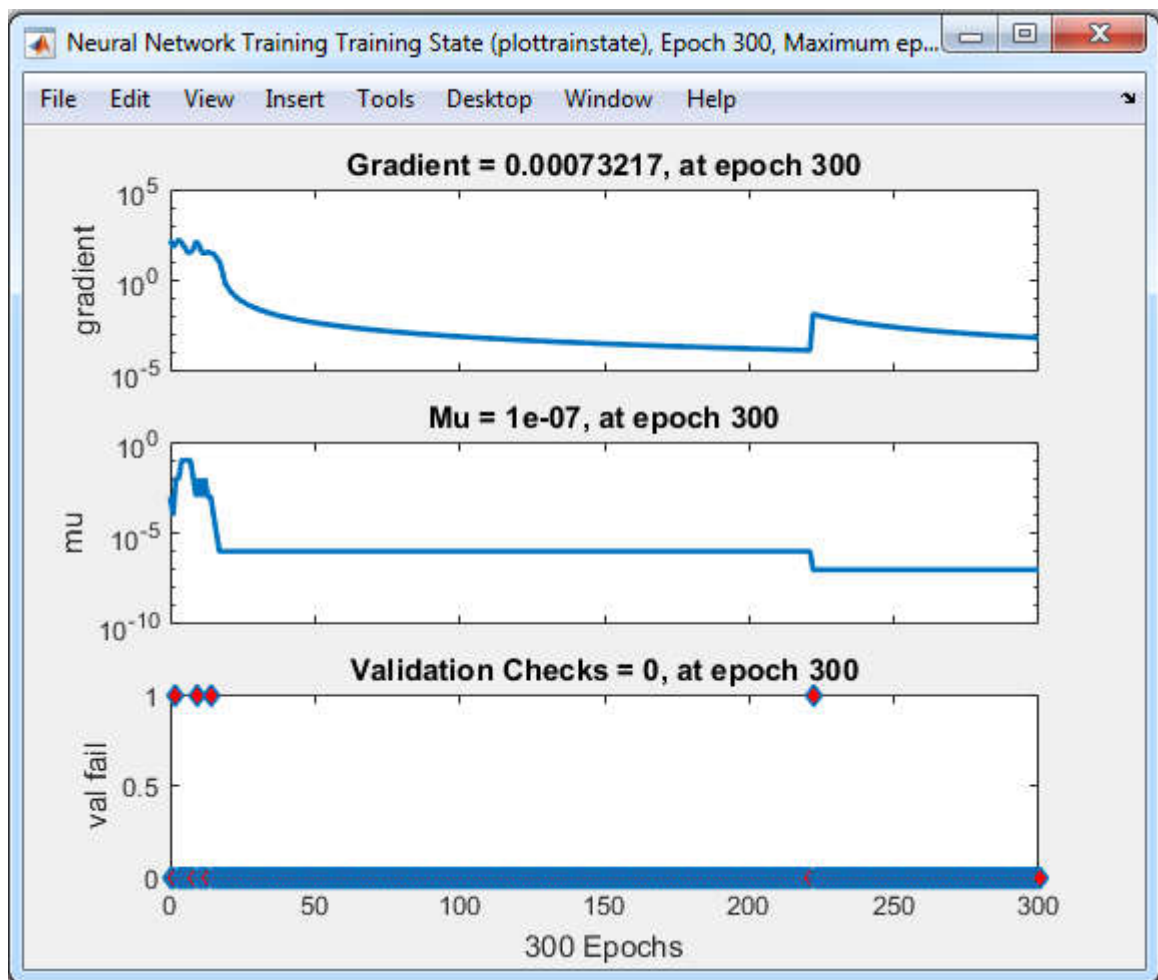


Рисунок 4.16 - Вікно Neural Network Training State

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.

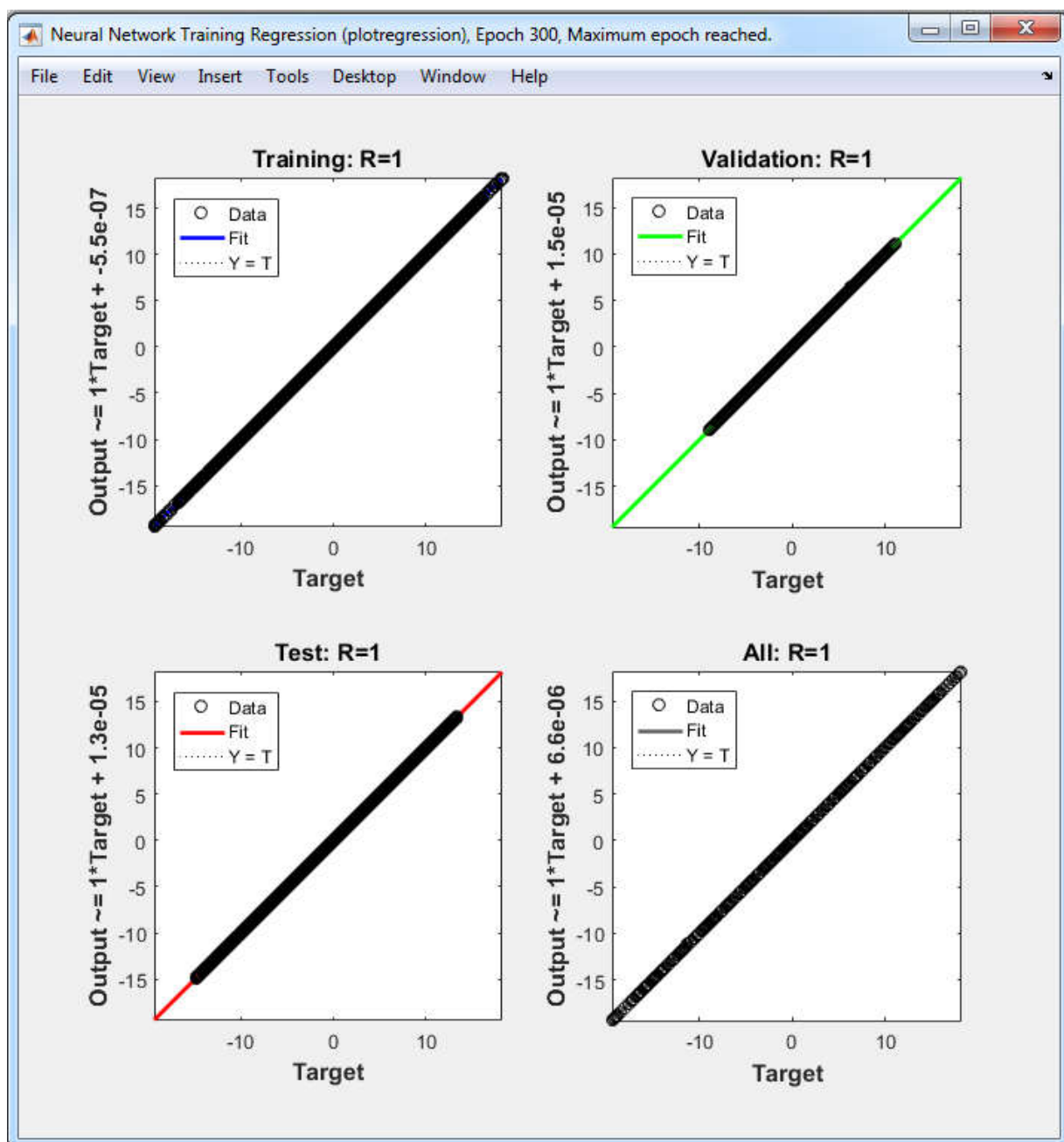


Рисунок 4.17 - Вікно Neural Network Training Regression

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.

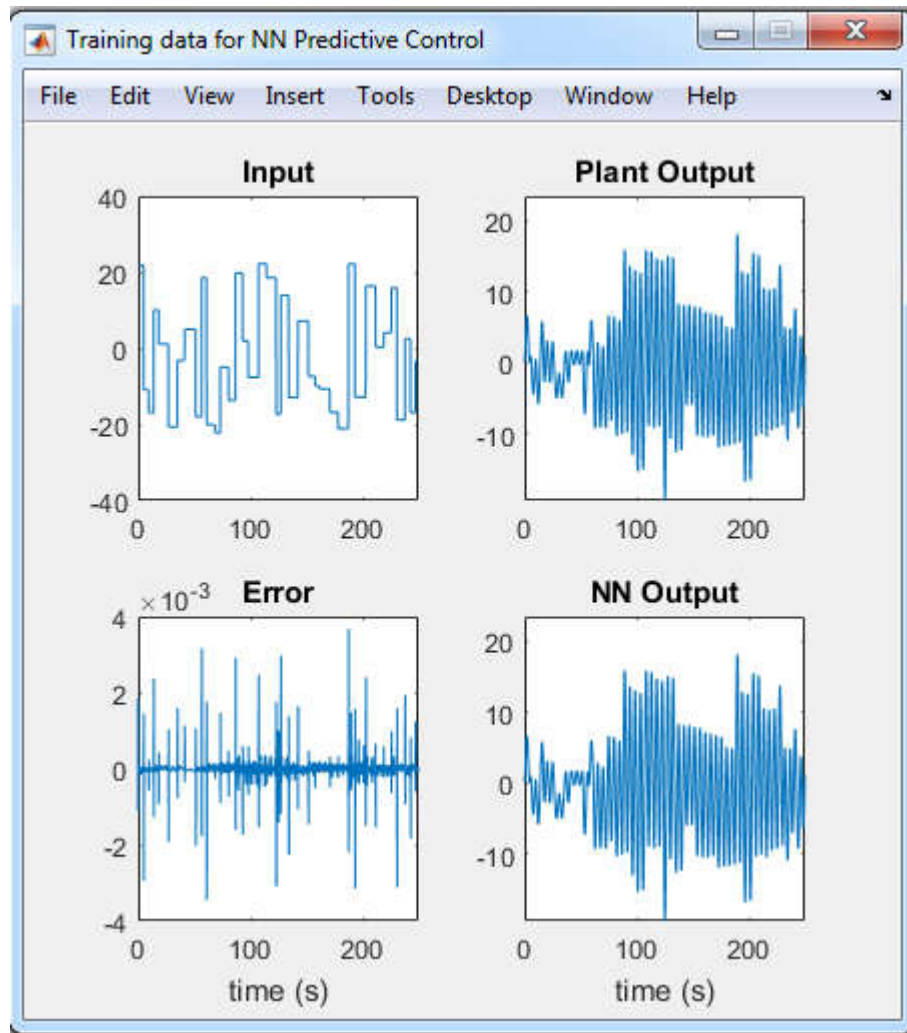


Рисунок 4.18 – Результати тренування мережі

Опис вікна Training data for NN Predictive Control

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

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

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

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

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

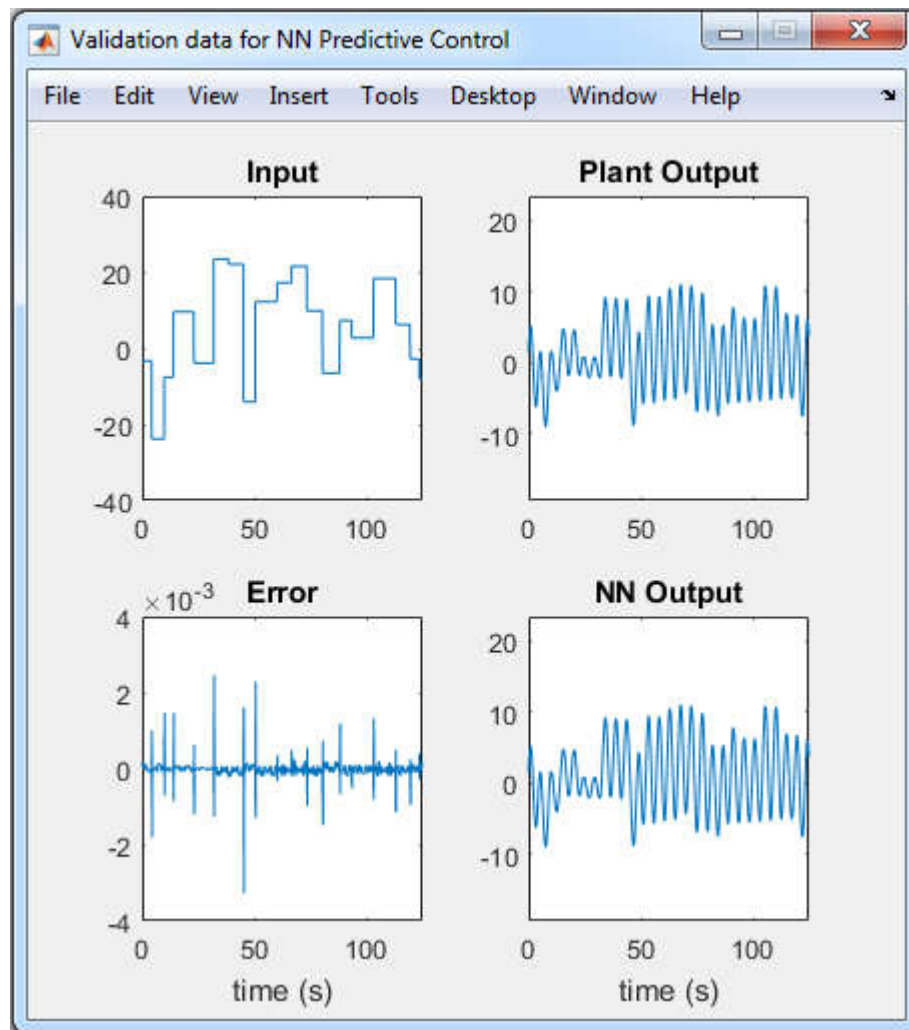


Рисунок 4.19 – Результати перевірки мережі на контрольній множині

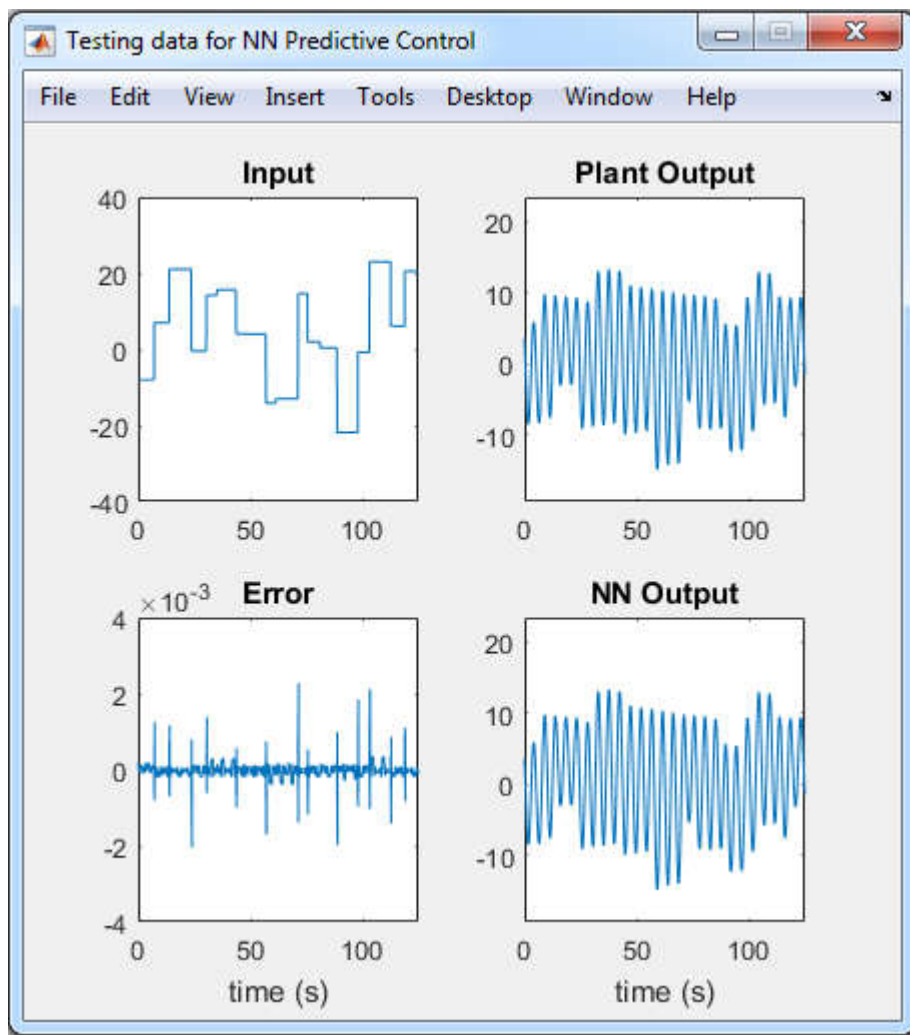


Рисунок 4.20 – Результати перевірки мережі на тестовій множині

Продовження опису вікна Plant Identification.

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

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

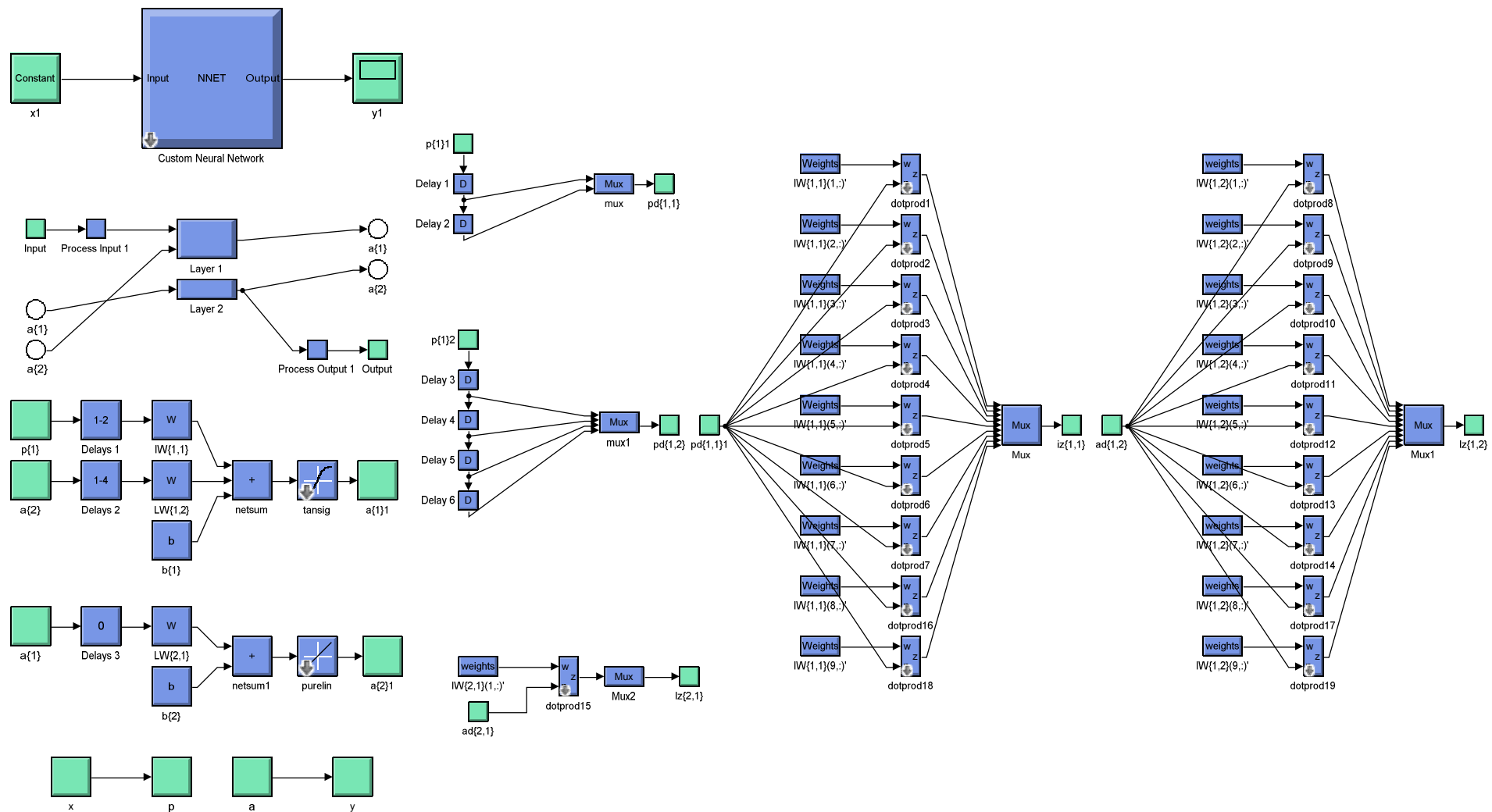


Рисунок 4.21 – Моделі елементів динамічної мережі з елементами затримок, реалізовані в Simulink

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

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

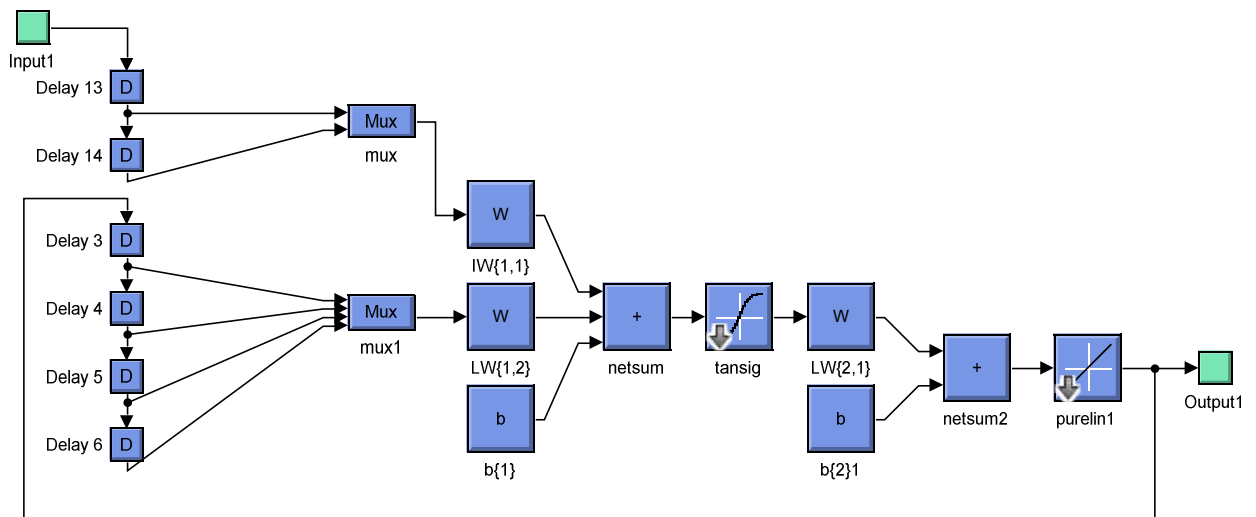


Рисунок 4.22 – Модель динамічної мережі з елементами затримок, побудованої в Simulink

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

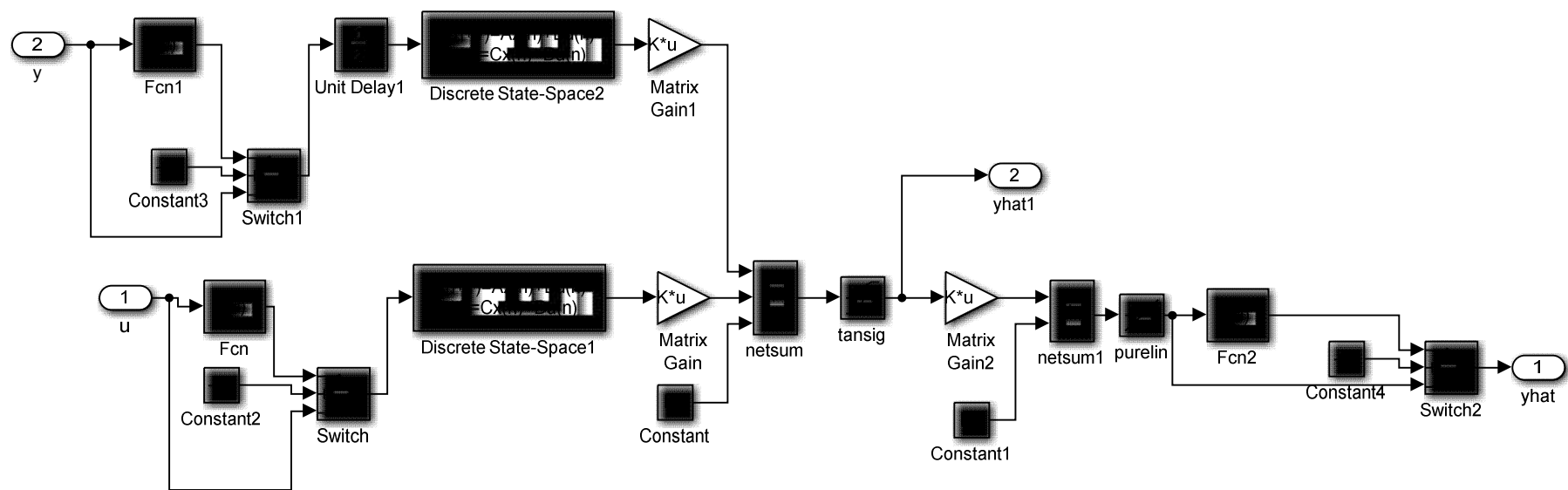


Рисунок 4.23 – Структурна схема нейронережевої моделі об'єкту регулювання

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

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

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

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

Discrete State Space 1

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

Discrete State Space 2

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

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

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

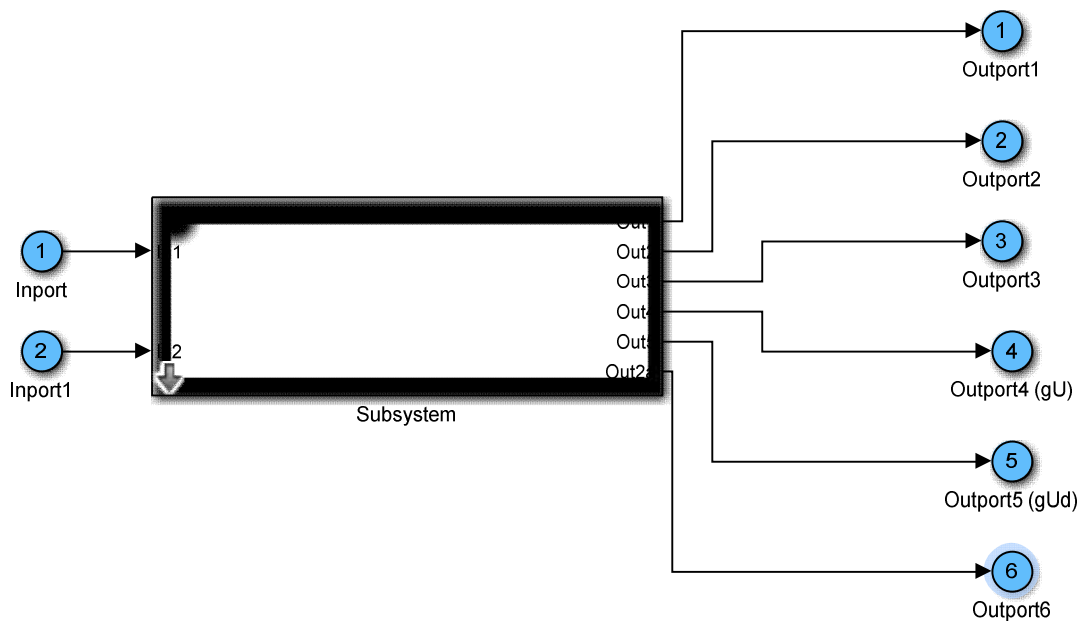


Рисунок 4.24 – Схема ptest3sim2, використовувана М-функцією predopt для прогнозу процесу в майбутньому

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

Продовження опису вікна Neural Network Predictive Controller

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

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

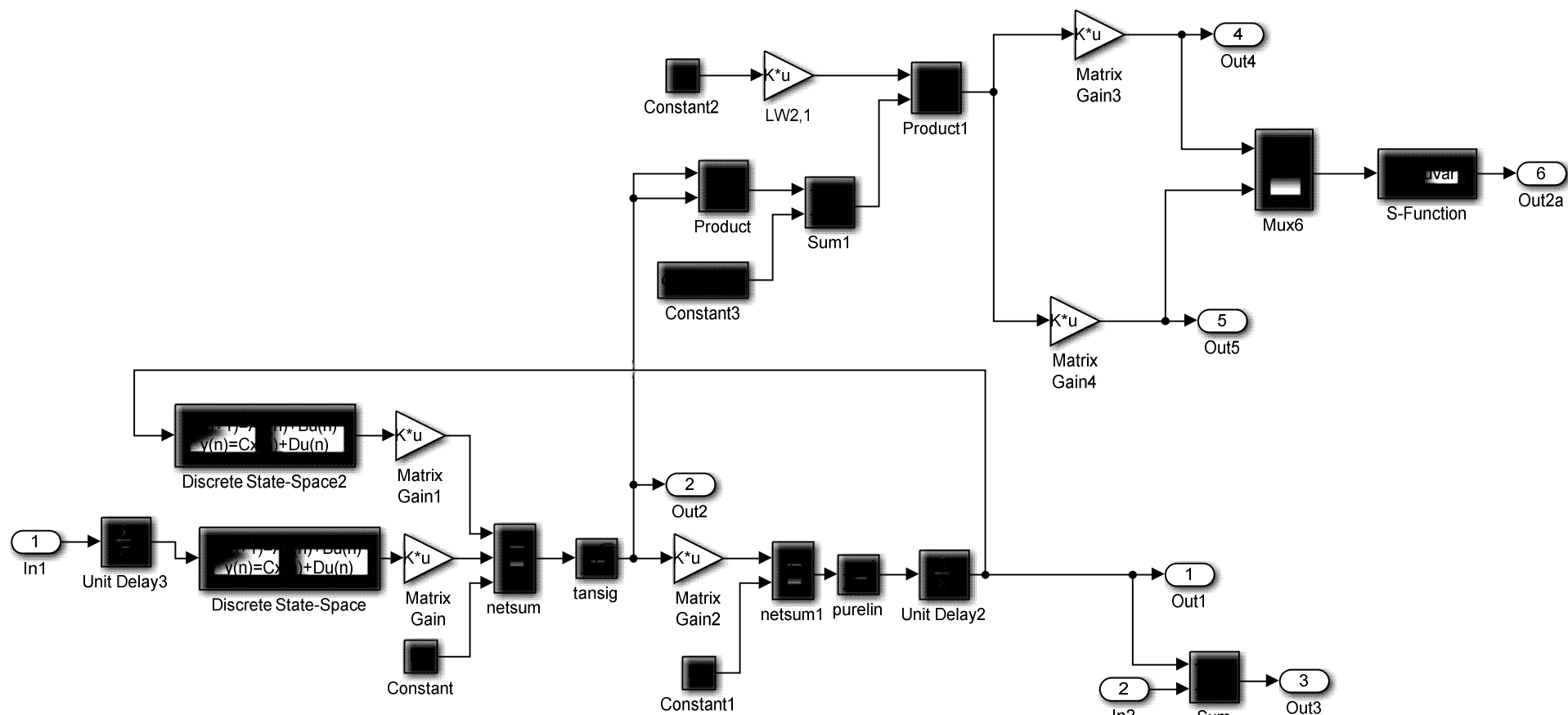


Рисунок 4.25 – Схема блоку Subsystem системи ptest3sim2

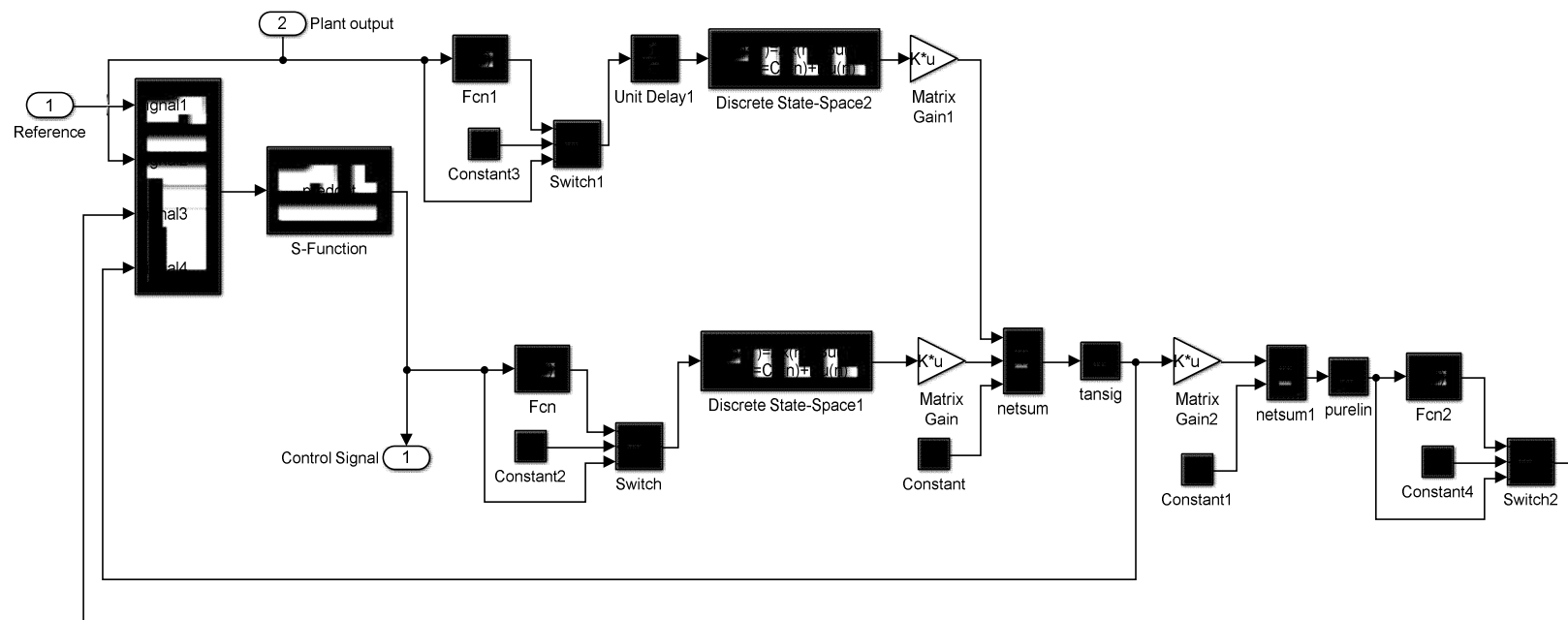


Рисунок 4.26 – Перетворена структурна схема нейрорегулятора

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

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

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

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

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

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

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

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

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

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

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

Minimization Routine. Вибір процедури одновимірного пошуку.

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

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

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

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

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

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

4.4 Вибір параметрів нейрорегулятора NN PREDICTIVE CONTROLLER

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

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

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

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

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

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

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

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

Як навчальну функцію використано **trainlm**.

4.5 Розрахунок динамічних характеристик систем з нейрорегулятором NN Predictive Controller

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

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

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

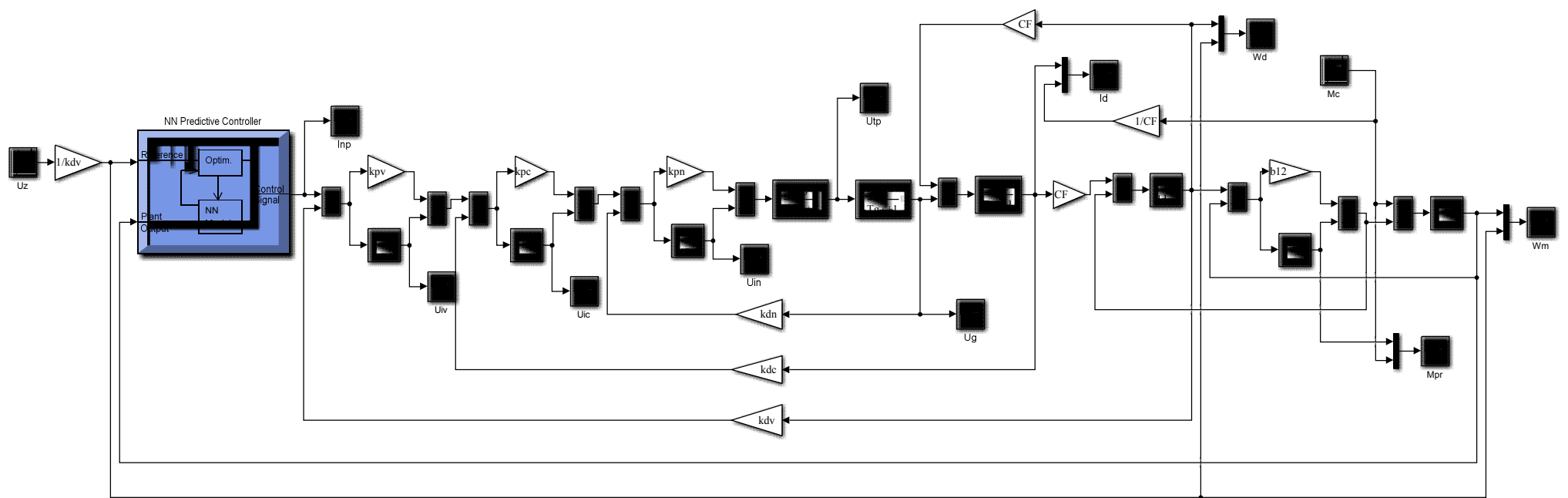


Рисунок 4.27 – Схема моделі двомасової системи з нейрорегулятором

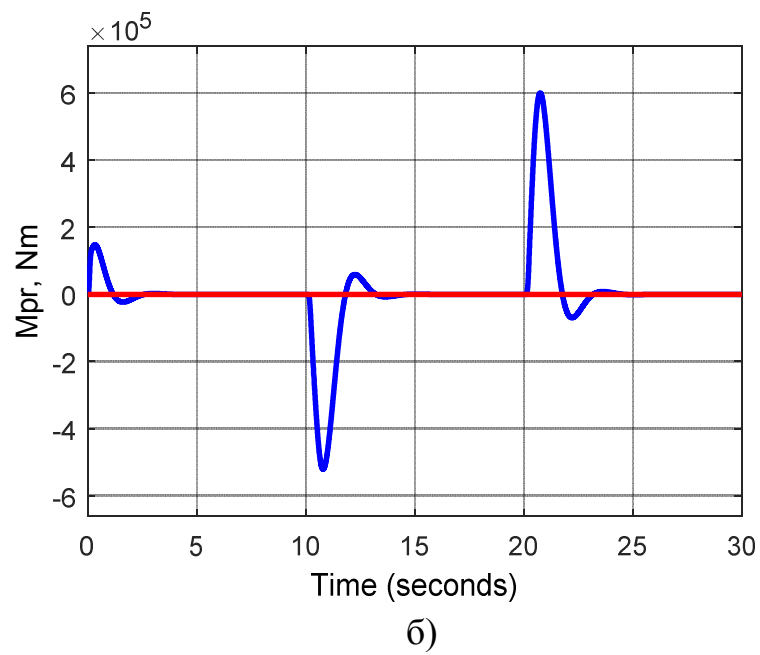
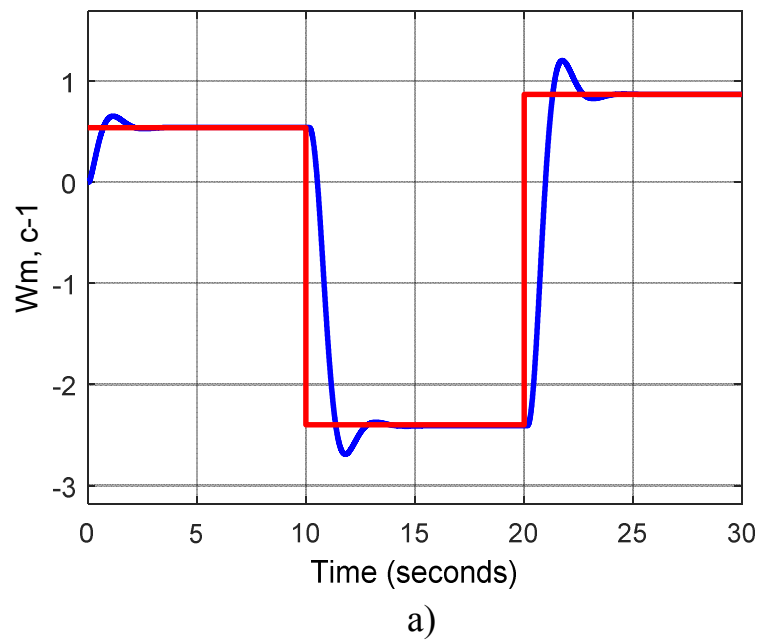


Рисунок 4.28 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{пр} = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

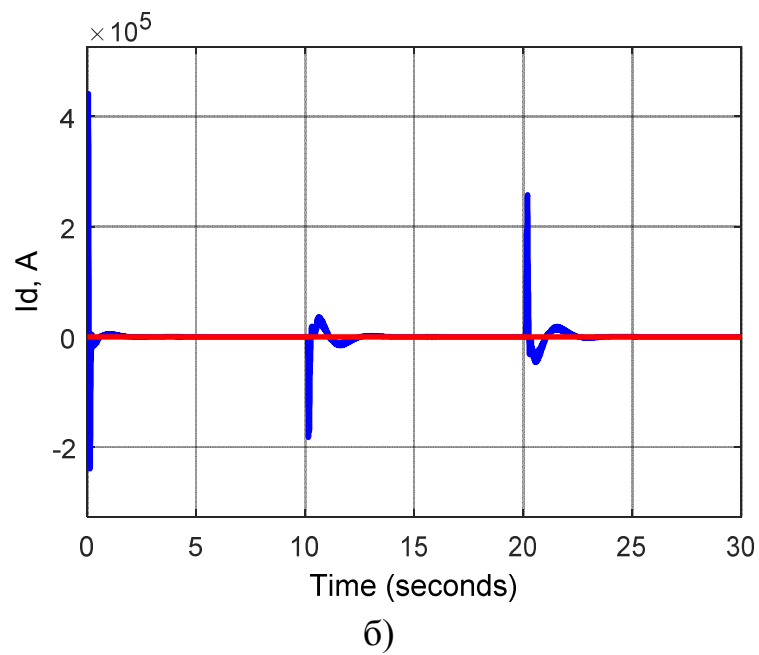
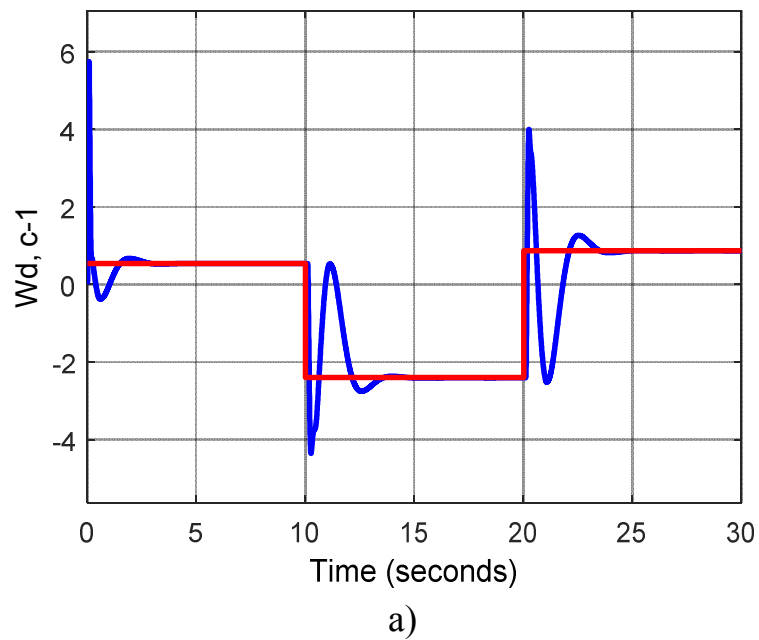


Рисунок 4.29 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи з нейрорегулятором по задаючій дії

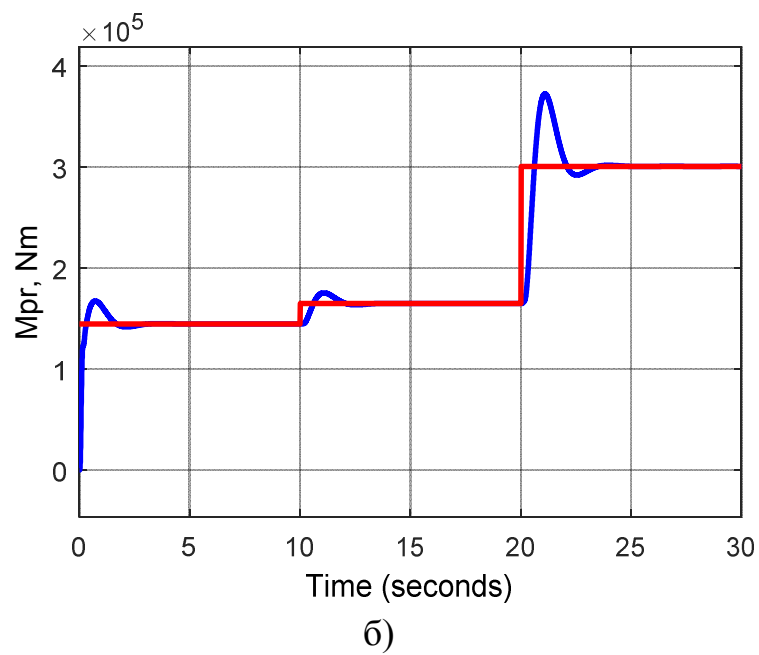
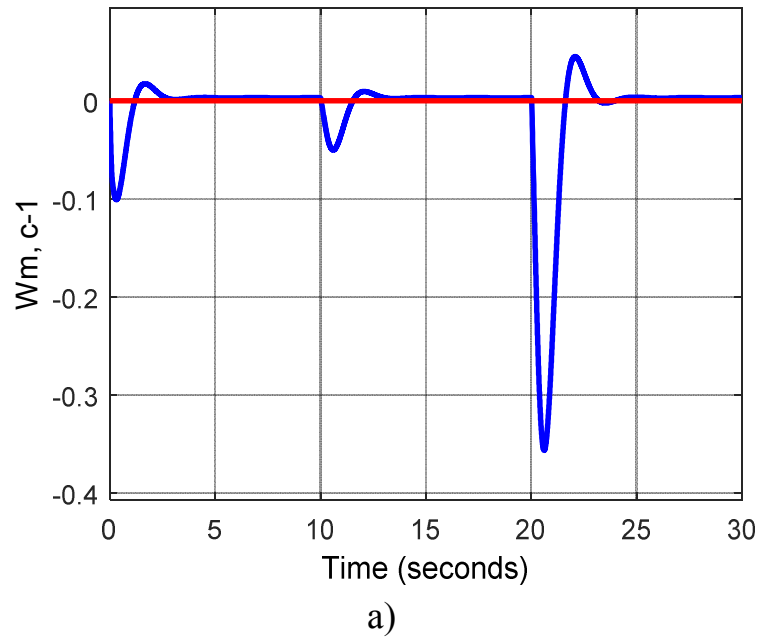


Рисунок 4.30 – Графіки перехідних процесів швидкості механізму $\omega_m = f(t)$ (а) і моменту пружності $M_{\text{пр}} = f(t)$ (б) двомасової системи з нейрорегулятором по обурюючій дії

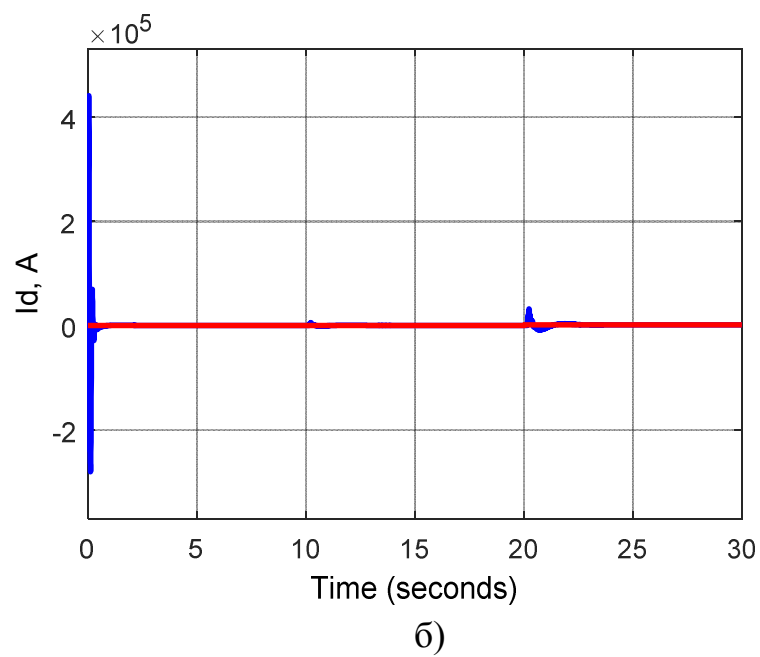
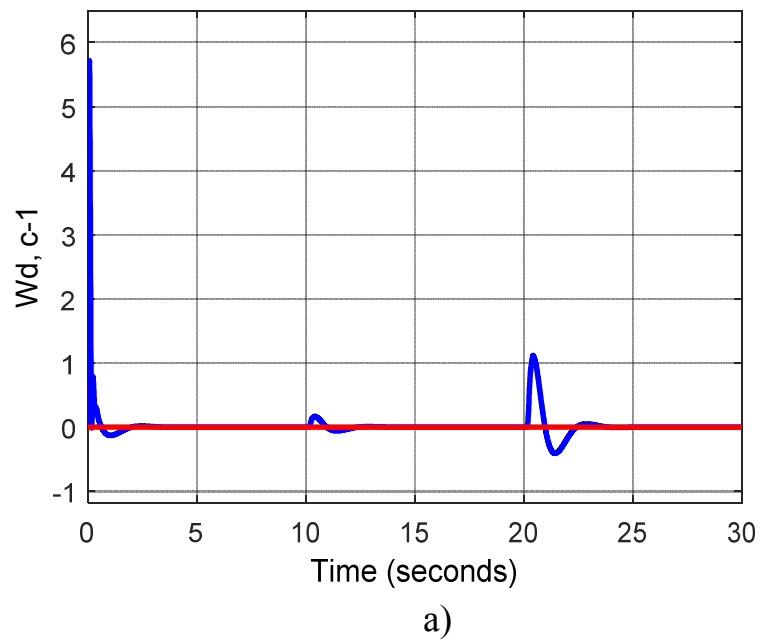


Рисунок 4.31 – Графіки перехідних процесів швидкості двигуна $\omega_d = f(t)$ (а) і струму двигуна $I_d = f(t)$ (б) двомасової системи з нейрорегулятором по обурюючій дії

ВИСНОВКИ ДО РОЗДІЛУ 4

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

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 підйомі шахтноїпвдіймальної установки 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 модель двомасової системи у вигляді двошарової нейронної мережі з прямою передачею сигналу і виконано навчання мережі з використанням алгоритму Левенберга-Марквардта. Визначено оптимальну кількість нейронів прихованого шару. Помилка навчання мережі має порядок 10^{-8} , миттєві помилки на навчальній, контрольній і тестовій множині не перевищують $4 \cdot 10^{-3}$.

5. Виконано синтез нейрорегулятора з передбаченням NN Predictive Controller. Завдяки варіації параметрів нейрорегулятора в широкому діапазоні

було визначено, які з них значно впливають на якість регулювання, а також знайдено оптимальні значення параметрів. Використання нейромережевої моделі об'єкта управління з високою ступінню ідентифікації дозволило створити нейрорегулятор, що забезпечує високі показники якості системи.

6. Проведено моделювання системи з нейромережевим регулятором. Аналіз результатів моделювання показав, що перехідні процеси змінних станів системи в системі з нейрорегулятором NN Predictive Controller у режимі запуску та навантаження мають задовільні характеристики. Система є астатичною. Перерегулювання не перевищує 10%, 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 шахтної підйимальної установки, що забезпечує високу якість регулювання.

Основні результати роботи полягають у наступному.

1. В результаті аналізу систем управління механізмом підйому шахтної підйимальної установки, 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 шахтної підйимальної установки.

2. На підставі аналізу технічних характеристик і умови роботи підйомних установок для електроприводу головного підйому вибраний без редукторний електропривод постійного струму, виконаний за системою Г–Д. Виконано вибір електродвигуна і основного електроустаткування силового кола електроприводу. Проведена перевірка двигуна по перевантажувальній здатності.

3. Проведений вибір і розрахунок системи автоматичного управління координати електроприводу. Вибрана трьох контурна система підлеглого регулювання з контуром напруги, струму і зовнішнім контуром швидкості. Виконано розрахунок параметрів силового кола електроприводу, що використовуються при розрахунку системи автоматичного управління. Проведений синтез послідовних коректуючи пристроїв контурів, з оптимізацією контурів напруги і струму по модульному критерію, а контура швидкості – по симетричному критерію. Проведено моделювання системи на ЕОМ з використанням пакету прокладних програм MATLAB.

4. Розроблена математична модель двомасової системи, що враховує пружні властивості підйимального канату. Проведено моделювання системи на

ЕОМ. Дослідження показали, що в двомасовій системі перехідні процеси основних координат мають значні коливання.

5. Для додання системі бажаних динамічних характеристик в роботі виконаний синтез нейромережевої системи управління з нейрорегулятором NN Predictive Controller, реалізованим в пакеті прикладних програм Neural Network Toolbox системи MATLAB. Розглянутий принцип побудови і функціонування нейромережевого регулятора, приведений порядок синтезу регулятора, побудовані процеси в двомасовій системі з нейрорегулятором. Аналіз результатів моделювання показав, що реакція системи на ступінчасту дію з випадковою амплітудою цілком задовільна, має коливальний характер з достатньо швидким загасанням. Система є астатичною. Перерегулювання не перевищує 8%, 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. 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.
14. 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.
15. 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.
16. Psaltis, D., Sideris, A., Yamamura, A. A Multilayered neural network controller // IEEE Control Systems Magazine. - 1988. - Vol.8. - N4. -P. 17-21.
17. Joardan M.A. Generic constraints on underspecified target trajectories // Proc. Of Int. Joint Conf. On Neural Networks (IJCNN), Washington. - 1989.- Vol.1. -N3. - P.217-225.
18. Joardan M.A., Rumelhart D.E. Forward models: Supervised learning with a distal teacher // Cognitive Science.- 1990. -Vol.16. -N5.-P.313-355.
19. Saerens M, Soquet A. Neural controller based on backpropagation algorithm // IEEE Proceedings-F. - 1991. -Vol.138. -N.I. -P.55-62.
20. 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.
21. 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.
22. Pham D.T., Liu X. Neural Network for Identification, Prediction and Control. - London: Springer - Verlag, 1997. - 238p.
23. Milito R., Padilla C.S., Padilla R.A., Cadorin D. An innovations approach to dual control // IEEE Trans. Automat. Control. - 1982. -Vol.27. -N1. -P.133-137.
24. Cichocki A., Unbehauen R. Neural networks for optimization and signal processing. – New York: John Wiley & Sons, 1998. – 52lp.

25. Garcia C.E., Prett D.M., Morari M. Model predictive control: theory and practice a survey // Automatica. – 1989. – Vol.25. – P.335-348.
26. 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.
27. 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.
28. Электрооборудование шахтных подъёмных машин /Ю.Т.Калашников, В.Е.Католиков, Г.И.Шпильберг и др., – М.: Недра, 1986. – 285 с.
29. Католиков В.Е., Динкель А.Д., Седунин А.М. Тиристорный электропривод с реверсом возбуждения двигателя рудничного подъёма. – М.: Недра, 1990.–382 с
30. Попович М. Г., Ковальчук О. В. Теорія автоматичного керування: Підручник. – 2-ге вид., перероб. і доп. – К. : Либідь, 2007. – 656 с.
31. Голюк П.Ф., Гречин Т.М. Теорія автоматичного керування: Навчальний посібник. – Л: Видавництво Львівської політехніки, 2012. – 280 с.
32. Головінський Б.Л., Шуруб Ю.В., Лисенко В.П. Теорія автоматичного управління: навч. посіб. – К.: Вид. Центр НУБіП України, 2012. – 240 с.
33. 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.
34. 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.
35. 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.

36. Євстіфєєв В. О. Теорія автоматичного керування. Частина 1. Безперервні лінійні та нелінійні системи. Навчальний посібник. – Кременчук: ПП Щербатих О. В., 2006. – 288 с.
37. Євстіфєєв В. О. Теорія автоматичного керування. Частина 2. Спеціальні системи автоматичного керування. Навчальний посібник. – Кременчук : ПП Щербатих О. В., 2007. – 224 с.
38. І.Г. Абраменко, Д. І. Абраменко Теорія автоматичного керування. – Харків: ХНАМГ, 2008. – 178 с.
39. 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
40. 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.
41. 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.
42. 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.
- 43.Гурко О.Г. Аналіз та синтез систем автоматичного управління у MATLAB: Навчальний посібник / О.Г. Гурко, І.Ф. Єрмоєнко. Харків, ХНАДУ, 2012. – 284 с.
44. Моделювання систем управління в SIMULINK : навч. посібник / В. О. Богомолів, О. Г. Гурко, В. І. Клименко, Д. М. Леонтєв, О. М. Красюк. Харків ХНАДУ, 2018. – 220 с.

45. Медведев В.В., Потемкин В.Г. Нейронные сети. MATLAB. /Под общ. Ред. к.т.н. В.Г. Потемкина. – М.: ДИАЛОГ-МИФИ, 2002.–496 с.
46. 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.
47. Дьяконов В. Математические пакеты расширения MATLAB. Специальный справочник / В. Дьяконов, В. Круглов. –СПб: Питер, 2001. –488 с.

ДОДАТОК А

БАГАТОШАРОВІ НЕЙРОННІ МЕРЕЖІ

А.1 Класифікація штучних нейронних мереж

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

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

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

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

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

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

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

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

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

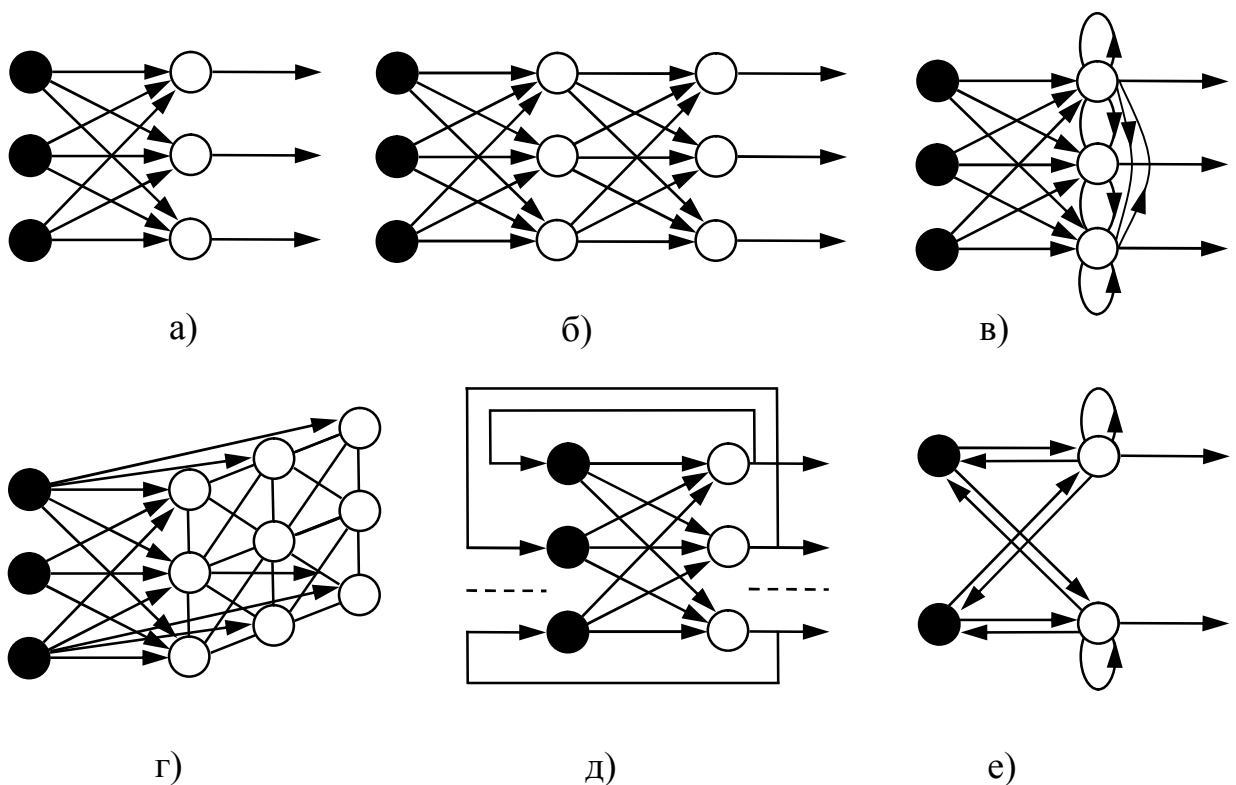


Рисунок А.1 – Структури найбільш поширених ШНМ

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

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

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

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

А.2 Базовий процесорний елемент (штучний нейрон)

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

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

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

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

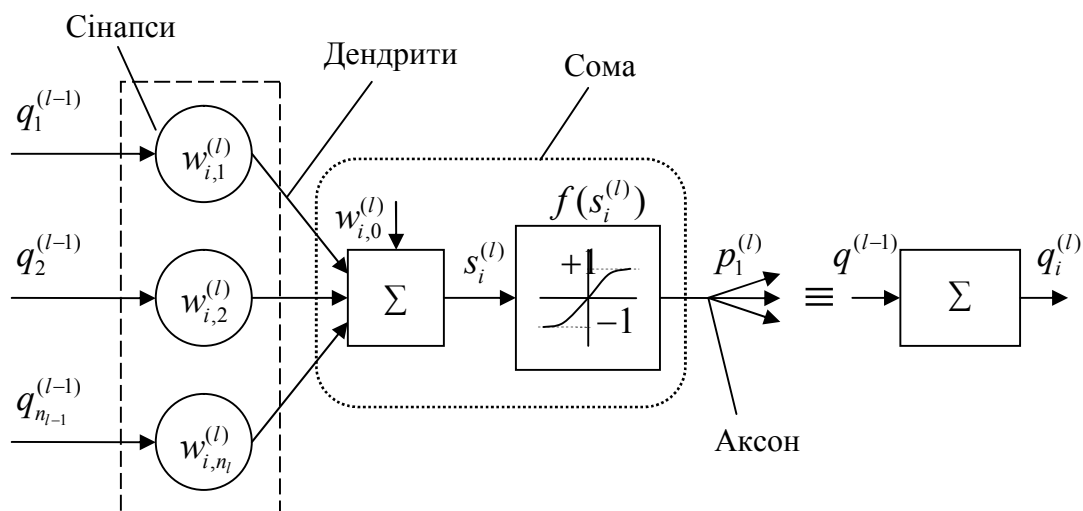


Рисунок А.2 – Схема базового процесорного елементу (штучного нейрона)

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

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

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

- експоненціальні або радіально-базисні (функції Гауса):

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

- гіперболічного тангенса:

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

- сигмоїдні:

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

- бінарні (різного визначення).

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

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

2.3 Статичні лінійні одношарові мережі.

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

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

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

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

однакового для всіх адалин,

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

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

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

видаленням останнього стовпця – вектора коефіцієнтів ініціалізуючих зв'язків

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

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

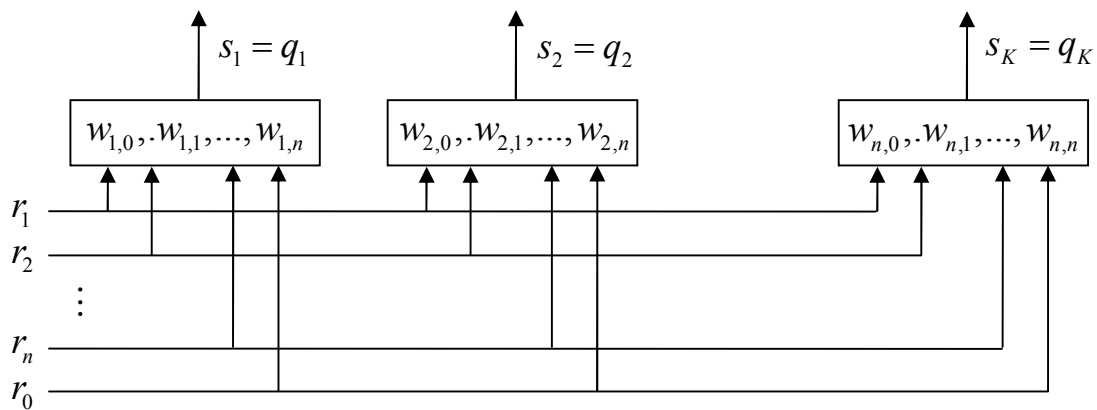


Рисунок А.3—схематичне представлення одношарової лінійної штучної нейронної мережі

А.3 Статичні багатошарові нейронні мережі

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

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

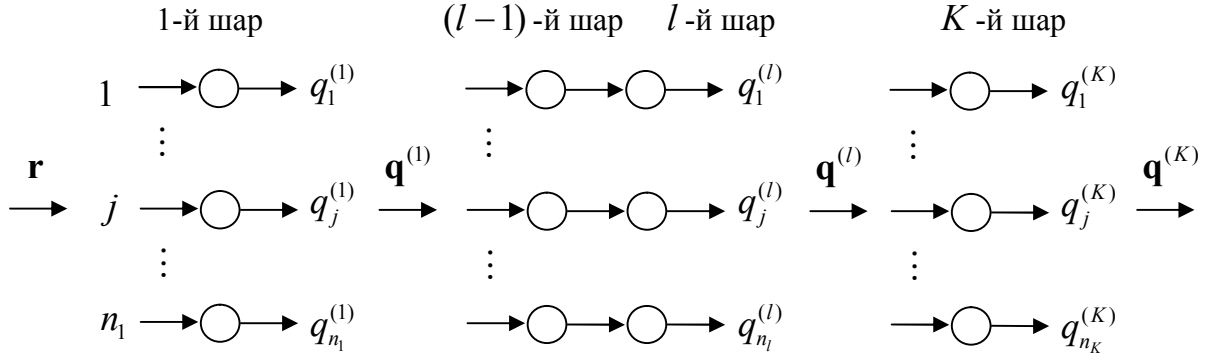


Рисунок А.4 – Структурна схема статичних БНМ

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

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

У векторній формі вихід l -го шару мережі рівний

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

де $\mathbf{w}_0^{(l)} = \text{col}(w_{1,0}^{(l)}, \dots, w_{n_l,0}^{(l)})$ – вектор вагових коефіцієнтів сигналу ініціалізації $q_0^{(l-1)}$ в шарі l , який для всіх $l = \overline{1, K}$ приймається рівним $+1$;

$\mathbf{q}_1^{(l)} = col(q_1^{(l)}, \dots, q_{n_l}^{(l)})$, $\mathbf{q}_1^{(l-1)} = col(q_1^{(l-1)}, \dots, q_{n_l}^{(l-1)})$ вектори виходів базових процесорних елементів шару l і виходів попереднього $(l-1)$ -го шару, базових елементів, що поступають на входи, шару l . Для вихідного K -го шару маємо:

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

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

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

Для елементів другого порядку дискримінантна функція визначається таким чином:

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

де $w_{i,0}^{(l)}, w_{i,j}^{(l)}, w_{j,k}^{(l)}$ ($j = \overline{1, n_{l-1}}, k \leq j$) – компоненти вектора вагових коефіцієнтів i -го базового елементу в l -му шарі мережі, а вектор входу $q^{(l-1)}$ має вигляд:

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

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

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

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

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

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

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

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

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

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

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

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

ДОДАТОК Б

ВЛАСТИВОСТІ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

Б. 1 Здібність до навчання і накопичення інформації

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

Б.2 Апроксимаційні властивості нейронних мереж

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

ДОДАТОК В

НАВЧАННЯ СТАТИЧНИХ БАГАТОШАРОВИХ НЕЙРОННИХ МЕРЕЖ

В.1 Загальна характеристика алгоритмів навчання

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

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

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

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

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

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

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

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

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

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

q_i, q_j – виходи нейронів i, j відповідно;

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

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

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

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

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

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

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

В.2 Метод і алгоритм зворотного розповсюдження помилки

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

В.2.1. Налаштування коефіцієнтів вихідного шару

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

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

де:

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

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

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

де:

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

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

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

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

Градієнт

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

де похідна:

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

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

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

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

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

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

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

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

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

а алгоритм настройки вагових коефіцієнтів K -го шару має вигляд

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

а)

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

тобто

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

де

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

б)

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

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

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

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

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

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

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

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

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

а)

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

б)

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

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

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

тоді

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

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

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

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

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

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

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

В.3 Перенавчання і узагальнення нейронних мереж

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

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

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

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

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

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

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

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

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

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

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

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

ДОДАТОК Г

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ В ЗАВДАННЯХ ІДЕНТИФІКАЦІЇ

Г.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 [13].

$$z(k+1) = \Psi[z(k), u(k), \phi(k)] ;$$

$$y(k) = \Phi[z(k), u(k), \phi(k)],$$

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.[14] 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 змінних. Наренда виділив чотири такі моделі:

– модель 1 (рис. B.1):

$$y(k+1) = \sum_{i=1}^{k_y-1} a_i y(k-i) + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (B.5)$$

– модель 2 (рис. B.2):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + \sum_{i=1}^{k_u-1} b_i u(k-i); \quad (B.6)$$

– модель 3 (рис. B.3):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1)] + g[u(k), u(k-1), \dots, u(k-n_u+1)]; \quad (B.7)$$

– модель 4 (рис. B.4):

$$y(k+1) = f[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-n_u+1)]. \quad (B.8)$$

Модель (B.4) є найбільш загальною зі всіх моделей вхід/вихід, описаних вище. При цьому модель (B.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.

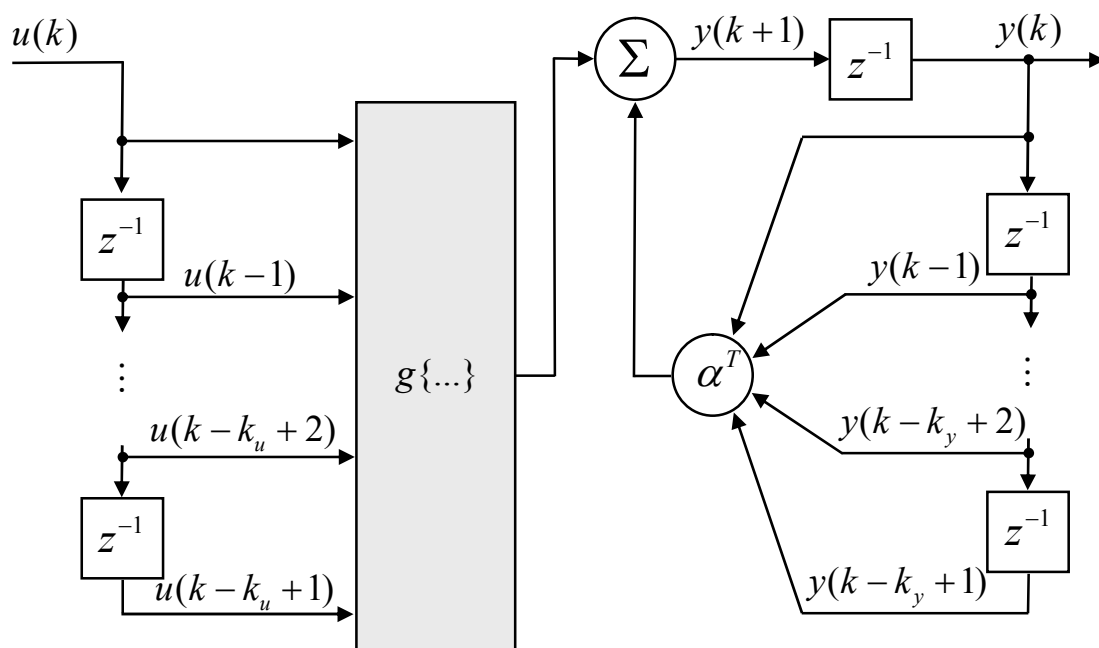


Рисунок В.1 – Модель 1. Вихід моделі нелінійно залежить від вхідних сигналів і лінійно від вихідних

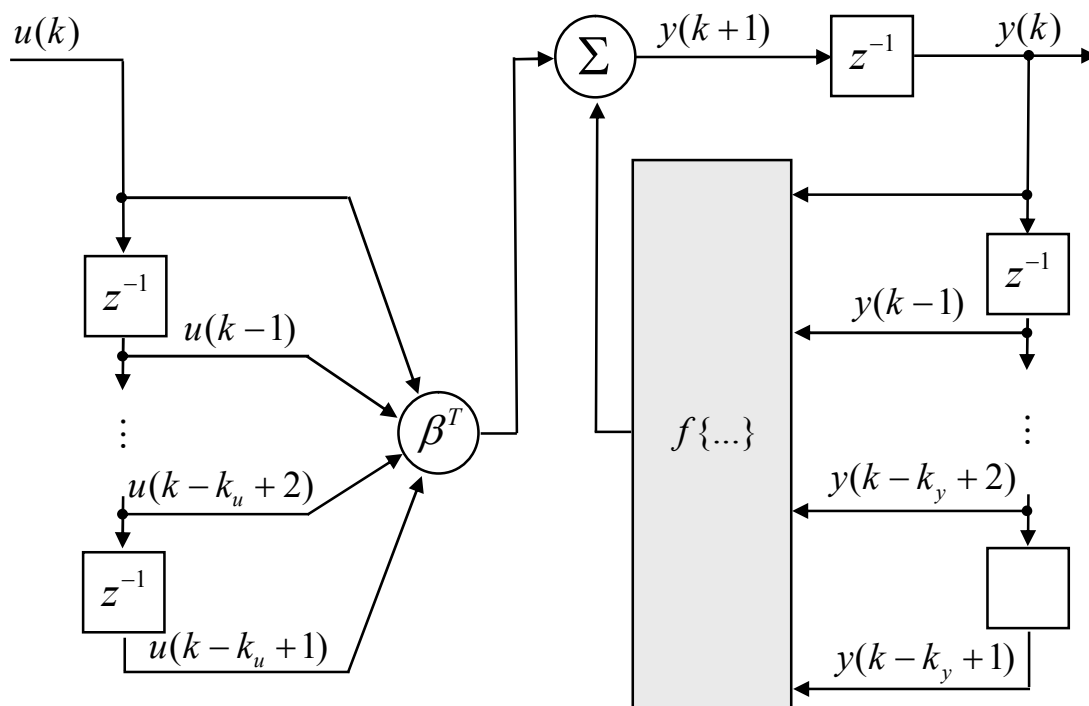


Рисунок В.2 – Модель 2. Вихід моделі лінійно залежить від вхідних сигналів і нелінійно від вихідних

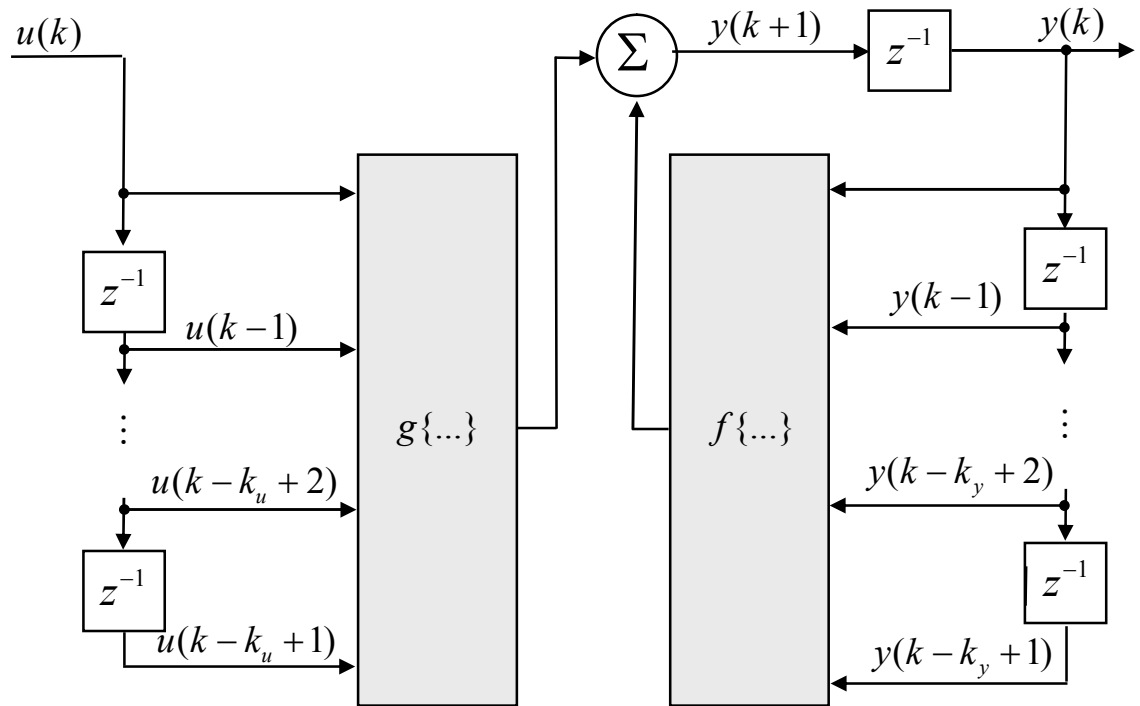


Рисунок В.3 – Модель 3. Резульгуючий сигнал є адитивною функцією нелінійних перетворень вхідних і вихідних сигналів

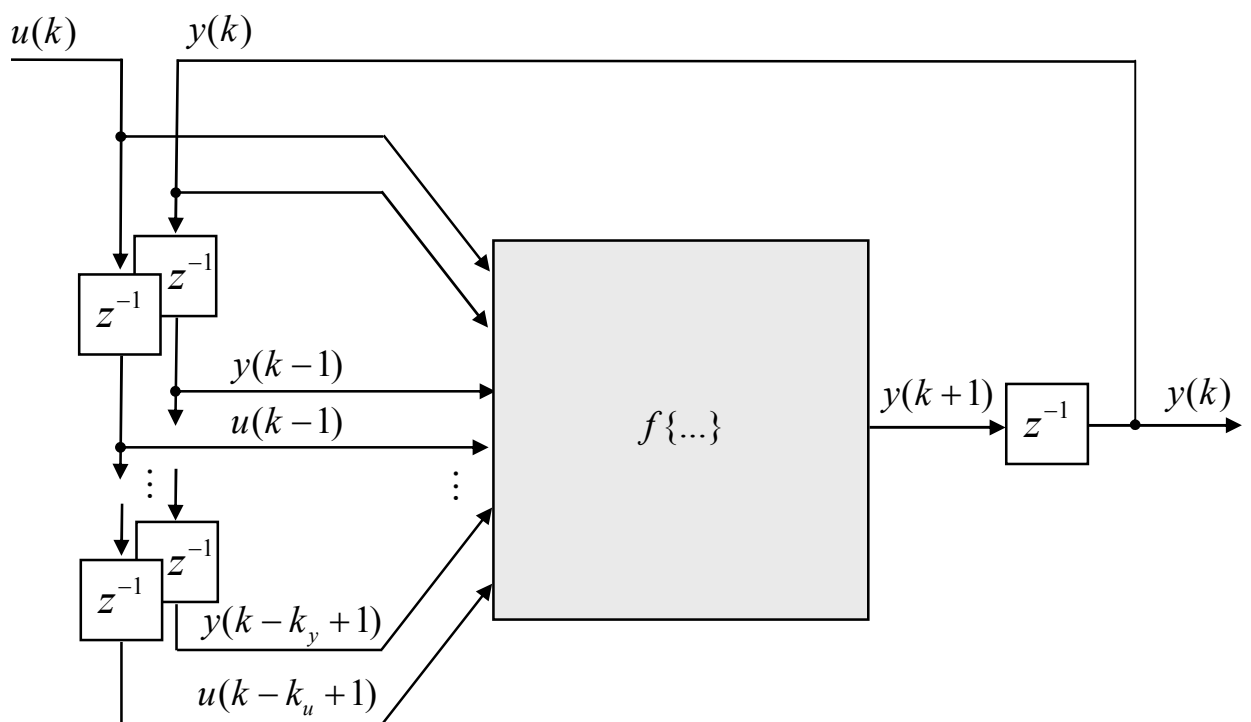


Рисунок В.4 – Модель 4. Вихід моделі нелінійно залежить як від вхідних, так і від вихідних сигналів

Г.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 [13] 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.

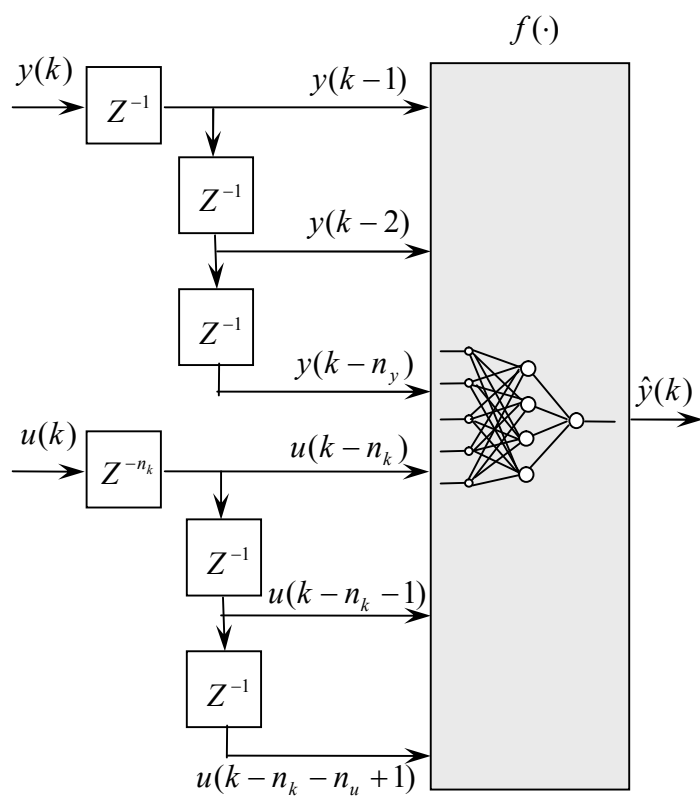


Рисунок В.5— Пряма мережева NARX модель

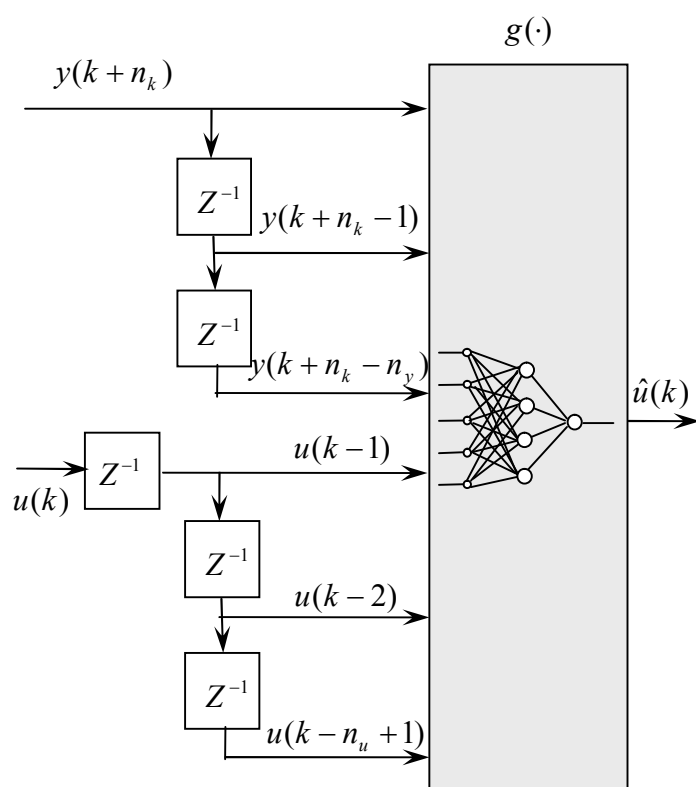


Рисунок В.6 – Інверсна мережева NARX модель

Г.3 Ідентифікація динамічних систем за допомогою нейронних мереж

Завдання теорії регулювання, що займається пошуком моделей динамічних систем на основі апріорної інформації і вимірюваних входних/вихідних сигналів системи, носить назву ідентифікація. Відомі класичні методи ідентифікації нелінійних систем застосовні тільки у поєднанні з певним типом моделі (наприклад, ряди Вальтера, моделі Хаммерштейна і Вінера), що описує певний клас нелінійних динамічних систем. У роботі під термінами «ідентифікація системи», «тренування» і «навчання» розуміється процес підстроювання вагових коефіцієнтів мережі за допомогою градієнтних методів. Метою цього процесу є по можливості точне наближення поведінки моделі до спостережуваної поведінки динамічної системи.

Принципово розрізняють пряму і інверсну модель об'єкту. На рис. В.7 представлені схеми тренування цих нейронних моделей. Для спрощення на рис. В.7 не зображені ланки затримки за часом, що формують попередні значення координат на вході нейронних мереж.

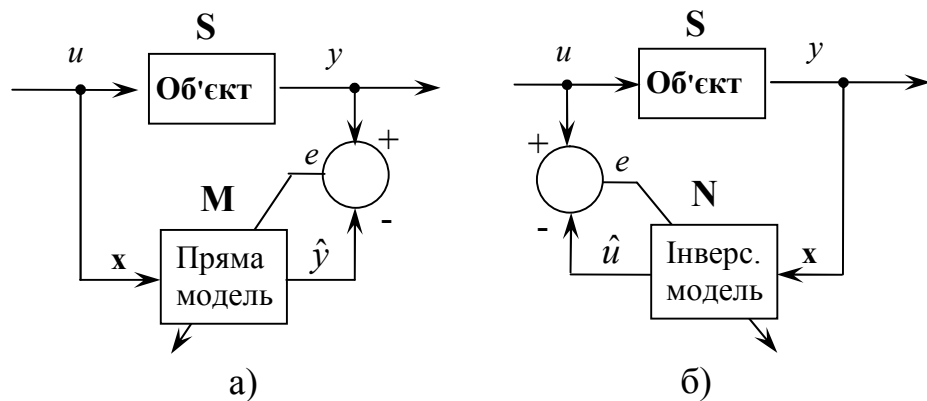


Рисунок В.7 – Схеми тренування прямої (а) і інверсної (б) моделей

Вхідний сигнал x нейронній моделі об'єкту M на рис. В.7,а ідентичний вхідному сигналу об'єкту S . Для інверсної нейромоделі N як вхідний сигнал використовується вихідний сигнал об'єкту S тоді як нейромодель повинна відновити вхідний сигнал об'єкту (див. рис. В.7,б). Поки присутня помилка між виходом нейромоделі і її зразковим сигналом, ця помилка використовується для

тренування мережі. У обох випадках ідентифікації обчислюється градієнт функціонала якості (див. таблицю В.1). Функціонал якості найчастіше при цьому квадратичний.

Таблиця В.1 – Обчислення градієнта при тренуванні нейромоделей

	Пряма модель	Інверсна модель
Бажаний вихідний сигнал	y	u
Вихідний сигнал нейромоделі	$\hat{y} = \mathbf{M}(\mathbf{x}, \mathbf{w})$	$\hat{u} = \mathbf{N}(\mathbf{x}, \mathbf{w})$
Помилка навчання	$e = y - \hat{y}$	$e = u - \hat{u}$
Градiєнт функціонала якості	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{y}}{\partial \mathbf{w}}$	$\frac{\partial J}{\partial \mathbf{w}} = \frac{\partial J}{\partial e} \frac{\partial e}{\partial \mathbf{w}} = -e \frac{\partial \hat{u}}{\partial \mathbf{w}}$

Оскільки $\frac{\partial \hat{y}}{\partial \mathbf{w}}$ і $\frac{\partial \hat{u}}{\partial \mathbf{w}}$ обчислювані, то це дозволяє вести підстроювання вагових коефіцієнтів мережі згідно розглянутим градієнтним методам. Результатом цього буде мінімізація функціонала якості. Оскільки набір тренувальних даних однозначно відображає лише поведінку модельованого об'єкту $\mathbf{S}$ у робочому діапазоні, то після завершення тренування маємо $\mathbf{M} \approx \mathbf{S}$ і $\mathbf{N} \approx \mathbf{S}^{-1}$.

Типова послідовність дій, що приводить до формування нейромоделі (ідентифікації) динамічної системи, відображена в таблиці В.2.

Таблиця В.2 – Ідентифікація за допомогою нейронних мереж

Шаг	Дія
1	Знімання тренувальних даних
2	Вибір структури мережі
3	Вибір типу моделі
4	Тренування мережі
5	Тестування нейромоделі
6	При незадов. результаті тестування, перейти до п.3 або до п.2

Оскільки тренувальні дані є носієм інформації про систему, що ідентифікується, то наявності, кількості і якості цих даних слід приділити особливу увагу. Тренувальних дані повинні мати наступні характеристики: якісна дискретизація, рівномірність дисперсії і величини варіації, рівномірність амплітуд всіх сигналів і їх середньоквадратичних значень.

При отриманні позитивного результату апроксимації, слід також перевірити здібність нейромоделі до узагальнення, тобто її здатність повторювати істотні характеристики об'єкту. Для цього тесту необхідний ще один набір даних, що не брав участь при тренуванні мережі. Цей тест дає остаточний висновок про адекватність нейромоделі.

Представлені теоретичні положення по нейронній ідентифікації, а також характеристики прямонаправлених мереж є основою успішного регулювання нелінійних динамічних систем.

ДОДАТОК Д

ФАЙЛ ВИХІДНИХ ДАНИХ

```
clear;clc;echo on;

% Система управління електроприводом
% механізму підйому шахтної підіймальної установки
% Система Г-Д, двомасова
% 2024 рік

% Вихідні дані
kg=3.34;
Tg=4.08;
ktp=22;
Tm=5e-3;
Tms=0.02;
Re=0.0254;
Te=0.036;
CF=170;
In=2624;
Mn=In*CF;
kdn=0.035;
kdc=3.66e-3;
kdv=5.44;
Jd=172098;
Jm=59764;
Js=Jd+Jm;
w0=1.5;
c12=(w0^2)*(Jd*Jm)/(Jd+Jm);
b12=0.005*c12;
%b=0;
kin=1/(2*Tm*ktp*kg*kdn);
kpn=Tg*kin;
kic=kdn*Re/(4*Tm*kdc);
kpc=Te*kic;
kiv=kdc*Js/(128*Tm^2*CF*kdv);
kpv=16*Tm*kiv;
Uz=24;

% Матриці
A=[-b12/Jm 1/Jm b12/Jm zeros(1,6);
-c12 0 c12 zeros(1,6);
b12/Jd -1/Jd -b12/Jd CF/Jd zeros(1,5);
0 0 -CF/(Re*Te) -1/Te 1/(Re*Te) 0 0 0 0];
```

```

0 0 0 0 -1/Tg kg/Tg 0 0 0;
0 0 (-kdv*kpv*kpn*kpc*ktp)/Tm -(kdc*kpc*kpn*ktp)/Tm -
(kdn*kpn*ktp)/Tm -1/Tm ktp/Tm (kpn*ktp)/Tm (kpc*kpn*ktp)/Tm;
0 0 -kdv*kpv*kpc*kin -kdc*kpc*kin -kdn*kin 0 0 kin kpc*kin;
0 0 -kdv*kpv*kic -kdc*kic zeros(1,4) kic;
0 0 -kdv*kiv zeros(1,6)]
B=[0 -1/Jm;
zeros(4,2);
(kpv*kpc*ktp*kpn)/Tm 0;
kpv*kpc*kin 0;
kpv*kic 0;
kiv 0]
C=zeros(9,9);
C(1,1)=1;C(2,2)=1;C(3,3)=1;C(4,4)=1
C(5,5)=1;C(6,6)=1;
C(7,7)=1;C(8,8)=1; C(9,9)=1;
D=zeros(9,2)

```